Direction and Normal Vectors

EE25BTECH11008 - Anirudh M Abhilash

September 30, 2025

Problem Statement

Find the direction and normal vectors of the line

$$y = 3x$$

Solution

The equation of the line is

$$y = mx + c \tag{1}$$

Comparing with $y = 3x$,

$$m = 3, \quad c = 0 \tag{2}$$

The vector form of the line is

$$\mathbf{x} = \mathbf{h} + \kappa\mathbf{m} \tag{3}$$

Solution (cont..)

Since the line passes through the origin,

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4}$$

Thus,

$$\mathbf{x} = \kappa \begin{pmatrix} 1 \\ 3 \end{pmatrix} \tag{5}$$

By Comparison, the direction vector is

$$\mathbf{m} = \begin{pmatrix} 1 \\ m \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \tag{6}$$

Solution (cont..)

For normal vector $\mathbf{n}$,

$$\mathbf{n}^\top \mathbf{m} = 0 \tag{7}$$

By Solving, the normal vector is

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \end{pmatrix} \tag{8}$$

$$\boxed{\mathbf{m} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \mathbf{n} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}}$$

Python Code (Plotting Line and Vectors)

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

m = 3
c = 0
mvec = np.array([1, m])
nvec = np.array([-1*m, 1])

x_vals = np.linspace(-5, 5, 100)
y_vals = m * x_vals + c

plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
```

Python Code (cont..)

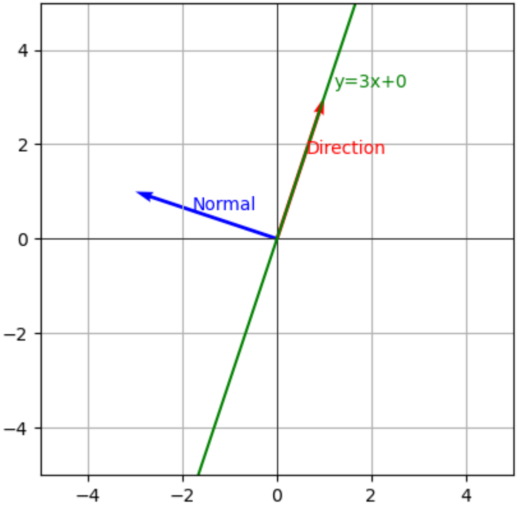
```
plt.plot(x_vals, y_vals, color="green")
plt.text(1+0.2, m*1 + c + 0.2, f"y={m}x+{c}", color="green")

plt.quiver(0, 0, mvec[0], mvec[1], angles='xy', scale_units='xy',
    scale=1, color='red')
plt.text(mvec[0]*0.6, mvec[1]*0.6, "Direction", color="red")

plt.quiver(0, 0, nvec[0], nvec[1], angles='xy', scale_units='xy',
    scale=1, color='blue')
plt.text(nvec[0]*0.6, nvec[1]*0.6, "Normal", color="blue")

plt.xlim(-5, 5)
plt.ylim(-5, 5)
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

Plot



C Code (Computations)

```
#include <stdio.h>

void line_vectors(double slope, double* mvec, double* nvec) {
    mvec[0] = 1.0;
    mvec[1] = slope;
    nvec[0] = -slope;
    nvec[1] = 1.0;
}
```

Python Code (Calling C)

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

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

lib.line_vectors.argtypes = [ctypes.c_double,
                             ctypes.POINTER(ctypes.c_double),
                             ctypes.POINTER(ctypes.c_double)]
```

Python Code (cont..)

```
def get_vectors(m, c):
    mvec = (ctypes.c_double * 2)()
    nvec = (ctypes.c_double * 2)()
    lib.line_vectors(m, mvec, nvec)
    return np.array([mvec[0], mvec[1]]), np.array([nvec[0], nvec
        [1]])

m = 3
c = 0

mvec, nvec = get_vectors(m, c)
print("Direction vector m =", mvec)
print("Normal vector n =", nvec)
```

Python Code (cont..)

```
x_vals = np.linspace(-5, 5, 100)
y_vals = m * x_vals + c

plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)

plt.plot(x_vals, y_vals, color="green")
plt.text(1+0.2, m*1 + c + 0.2, f"y={m}x+{c}", color="green")
```

Python Code (cont..)

```
plt.quiver(0, 0, mvec[0], mvec[1], angles='xy', scale_units='xy',
    scale=1, color='red')
plt.text(mvec[0]*0.6, mvec[1]*0.6, "Direction", color="red")

plt.quiver(0, 0, nvec[0], nvec[1], angles='xy', scale_units='xy',
    scale=1, color='blue')
plt.text(nvec[0]*0.6, nvec[1]*0.6, "Normal", color="blue")

plt.xlim(-5, 5)
plt.ylim(-5, 5)
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```