Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 16, 2025

Problem Statement

Problem 4.12.46 :
Find the values of θ and p, if the equation

$$x\cos\theta + y\sin\theta = p \tag{1.1}$$

is the normal form of the line

$$\sqrt{3}x + y + 2 = 0. \tag{1.2}$$

Description of Variables used

Quantity	**Value**
Normal vector $\mathbf{n}$	$\begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix}$
Constant c	2

Table

Theoretical Solution

The line can be expressed as

$$\mathbf{n}^T \mathbf{u} = -c. \tag{2.1}$$

The length of the normal is

$$\|\mathbf{n}\| = \sqrt{(\sqrt{3})^2 + 1^2} = 2. \tag{2.2}$$

Thus, the unit normal becomes

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|} = \begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix}. \tag{2.3}$$

Theoretical Solution

Dividing (3) by $\|\mathbf{n}\|$ gives the normal form:

$$\hat{\mathbf{n}}^T\mathbf{u} = \frac{-c}{\|\mathbf{n}\|} = -1. \tag{2.4}$$

Comparing with the standard normal form

$$x\cos\theta + y\sin\theta = p, \tag{2.5}$$

we identify

$$\cos\theta = \frac{\sqrt{3}}{2}, \quad \sin\theta = \frac{1}{2}. \tag{2.6}$$

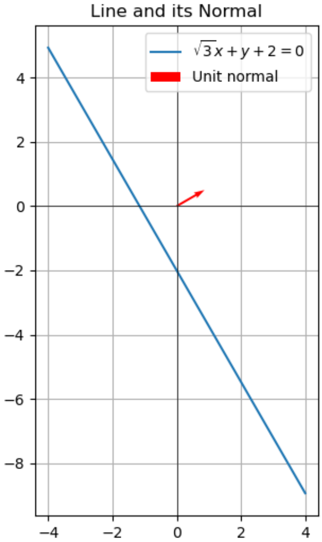
Hence,

$$\theta = \frac{\pi}{6}, \quad p = -1. \tag{2.7}$$

Final Answer:

$$\boxed{\theta = \frac{\pi}{6}, \quad p = -1.} \tag{2.8}$$

Plot



Code - C

```
#include <stdio.h>
#include <math.h>

// Compute norm of a 2D vector
double norm(double *vec) {
    return sqrt(vec[0]*vec[0] + vec[1]*vec[1]);
}

// Normalize a 2D vector
void normalize(double *vec, double *out) {
    double n = norm(vec);
    out[0] = vec[0]/n;
    out[1] = vec[1]/n;
}
```

Code - C

```c
// Compute p = -c / norm(n)
double compute_p(double *vec, double c) {
    double n = norm(vec);
    return -c / n;
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

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

# Load shared library
lib = ctypes.CDLL("./liblineutils.so")

# Define argument/return types
lib.norm.argtypes = [ctypes.POINTER(ctypes.c_double)]
lib.norm.restype = ctypes.c_double

lib.normalize.argtypes = [ctypes.POINTER(ctypes.c_double), ctypes.
    POINTER(ctypes.c_double)]
lib.normalize.restype = None
```

Code - Python(with shared C code)

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

# Input data
n = np.array([np.sqrt(3), 1.0], dtype=np.double)
c = 2.0

# Allocate output for unit normal
unit_n = np.zeros(2, dtype=np.double)

# Convert numpy array to C pointer
n_ptr = n.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
unit_ptr = unit_n.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
```

Code - Python(with shared C code)

```
# Call C functions
norm_val = lib.norm(n_ptr)
lib.normalize(n_ptr, unit_ptr)
p_val = lib.compute_p(n_ptr, c)

print("||n|| =", norm_val)
print("Unit normal =", unit_n)
print("p =", p_val)

# −−−−−−−−−−−−−−−−−− Plotting
    −−−−−−−−−−−−−−−−−−
# Line: sqrt(3)x + y + 2 = 0 => y = −sqrt(3)x − 2
x_vals = np.linspace(−4, 4, 200)
y_vals = −np.sqrt(3)*x_vals − 2
```

Code - Python(with shared C code)

```
#Call C functions
norm_val = lib.norm(n_ptr)
lib.normalize(n_ptr, unit_ptr)
p_val = lib.compute_p(n_ptr, c)

print("||n|| =", norm_val)
print("Unit normal =", unit_n)
print("p =", p_val)

# −−−−−−−−−−−−−−−−−−−−− Plotting
    −−−−−−−−−−−−−−−−−−−−−
# Line: sqrt(3)x + y + 2 = 0 => y = −sqrt(3)x − 2
x_vals = np.linspace(−4, 4, 200)
y_vals = −np.sqrt(3)*x_vals − 2
# Normal vector (scaled for plotting)
origin = np.array([0,0])
normal_line = np.vstack([origin, unit_n*2]) # scale for visibility
```

Code - Python(with shared C code)

```
plt.figure(figsize=(6,6))
plt.plot(x_vals, y_vals, label=r"$\sqrt{3}x+y+2=0$")
plt.quiver(0,0, unit_n[0], unit_n[1], angles='xy', scale_units='xy', scale=1,
    color='red', label="Unit normal")

plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.legend()
plt.gca().set_aspect("equal")
plt.title("Line and its Normal")
plt.grid(True)
plt.savefig("normalform.png")
plt.show()
```

Code - Python only

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

# Input
n = np.array([np.sqrt(3.0), 1.0])
c = 2.0

# Compute norm, unit normal, and p
norm_n = np.linalg.norm(n)
unit_n = n / norm_n
p = -c / norm_n
theta = np.arctan2(unit_n[1], unit_n[0])
```

Code - Python only

```
# Print results
print("n =", n)
print("c =", c)
print("||n|| =", norm_n)
print("unit normal =", unit_n)
print("p =", p)
print("theta =", theta, "rad (~", theta*180/np.pi, "degrees)")

# Plot the line: sqrt(3)x + y + 2 = 0 -> y = -sqrt(3)x - 2
x_vals = np.linspace(-4, 4, 400)
y_vals = -np.sqrt(3.0) * x_vals - 2

plt.figure(figsize=(6,6))
plt.plot(x_vals, y_vals, label="Line: sqrt(3)x + y + 2 = 0")
```

Code - Python only

```
# Plot unit normal arrow from origin
plt.quiver(0, 0, unit_n[0], unit_n[1],
           angles="xy", scale_units="xy", scale=1,
           color="red", label="Unit normal")

# Axes and labels
plt.axhline(0, color="black", linewidth=0.5)
plt.axvline(0, color="black", linewidth=0.5)
plt.gca().set_aspect("equal")
plt.xlim(-4, 4)
plt.ylim(-6, 4)
plt.legend()
plt.title("Line and its Unit Normal")
plt.grid(True)
plt.savefig("normalform_new.png")
plt.show()
```