Matgeo Presentation - Problem 4.7.60

ee25btech11063 - Vejith

September 7, 2025

Question

Reduce the equation $\sqrt{3}$x+y-8=0 into normal form.Find the values of p and ω.

Solution

Given line equation is

$$\sqrt{3}x + y - 8 = 0 \tag{0.1}$$

which can be written as

$$\mathbf{n}^T\mathbf{x} = c \tag{0.2}$$

$$\implies \begin{pmatrix} \sqrt{3} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 8 \tag{0.3}$$

$$\mathbf{n} = \begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ and c} = 8 \tag{0.4}$$

Length (norm) of **n** is given as

$$\|\mathbf{n}\| = \sqrt{\mathbf{n}^T\mathbf{n}} = 2. \tag{0.5}$$

The unit normal is given by

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|} = \begin{pmatrix} \sqrt{3}/2 \\ 1/2 \end{pmatrix} \tag{0.6}$$

Solution

Divide the line equation by $\|\mathbf{n}\|$ to get the normal form

$$\implies \mathbf{n}^T\mathbf{x} = 4. \tag{0.7}$$

$$\implies \begin{pmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 4. \tag{0.8}$$

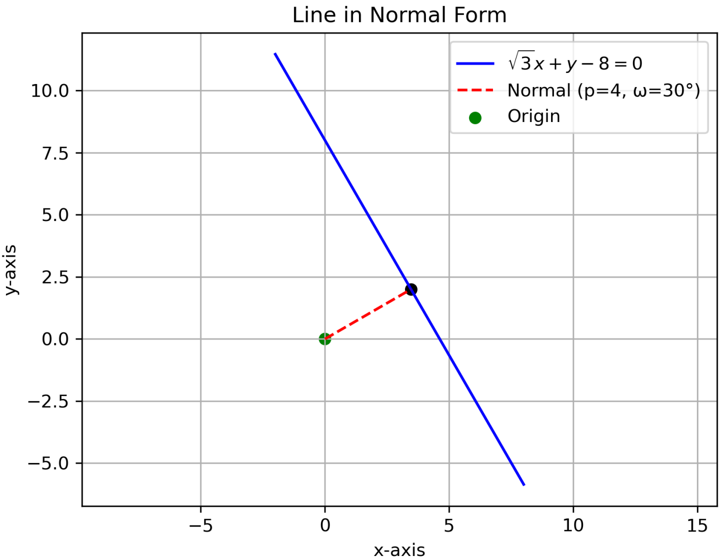
The standard form of line in normal form is given by

$$\begin{pmatrix} \cos\omega & \sin\omega \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathrm{p}. \tag{0.9}$$

On comparing equations (8) and (9) we get

$$\mathrm{p} = 4 \text{ and } \omega = \frac{\pi}{6} \tag{0.10}$$

Plot


Line in Normal Form
$\sqrt{3}x + y - 8 = 0$
Normal (p=4, ω=30°)
Origin
y-axis
x-axis
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−5
0
5
10
15

C Code: triangle.c

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

int main() {
    FILE *fp;
    fp = fopen("norm.dat", "w");
    if (fp == NULL) {
        printf("Error␣opening␣file!\n");
        return 1;
    }

    // Line equation: sqrt(3)x + y - 8 = 0
    // Normal vector n = [sqrt(3), 1]
    double n[2] = {sqrt(3), 1.0};

    // Compute norm of n
    double norm = sqrt(n[0]*n[0] + n[1]*n[1]);

    // Unit normal (n cap)
    double n_cap[2];
    n_cap[0] = n[0] / norm;
    n_cap[1] = n[1] / norm;

    // Compute p = constant / norm
    double p = 8.0 / norm;

    // Compute angle omega = atan2(sin, cos)
    double omega = atan2(n_cap[1], n_cap[0]); // in radians

    // Write results into file
    fprintf(fp, "Normal␣vector␣n␣=␣[%.4f,␣%.4f]\n", n[0], n[1]);
    fprintf(fp, "Norm␣of␣n␣=␣%.4f\n", norm);
    fprintf(fp, "Unit␣normal␣n_cap␣=␣[%.4f,␣%.4f]\n", n_cap[0], n_cap[1]);
```

C Code: triangle.c

```
 fprintf(fp, "Normal␣form:␣(%.4f)x␣+␣(%.4f)y␣=␣%.4f\n", n_cap[0], n_cap[1], p);
 fprintf(fp, "p␣=␣%.4f\n", p);
    fprintf(fp, "omega␣(radians)␣=␣%.4f\n", omega);
    fprintf(fp, "omega␣(degrees)␣=␣%.2f\n", omega * 180.0 / M_PI);

    fclose(fp);

    printf("Results␣written␣to␣norm.dat\n");
    return 0;
}
```

Python: plot.py

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

# Line: sqrt(3)x + y - 8 = 0
# => y = -sqrt(3)x + 8
x = np.linspace(-2, 8, 400)
y = -np.sqrt(3) * x + 8

# Plot line
plt.plot(x, y, 'b', label=r'$\sqrt{3}x␣+␣y␣-␣8␣=␣0$')

# Plot normal vector at the foot of perpendicular (p=4, omega=30)
p = 4
omega = np.pi / 6 # 30 degrees
# Point on the line at perpendicular distance p from origin
x0 = p * np.cos(omega)
y0 = p * np.sin(omega)

# Draw perpendicular from origin
plt.plot([0, x0], [0, y0], 'r--', label='Normal␣(p=4,␣=30)')
plt.scatter([x0], [y0], color='k') # mark foot of perpendicular
plt.scatter([0], [0], color='g', label='Origin')

# Labels, legend, grid
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Line␣in␣Normal␣Form")
plt.legend()
plt.grid(True)
plt.axis("equal")
# Save figure
plt.savefig("line_normal_form.png", dpi=300)
plt.close()
```