Matgeo Presentation - Problem 2.7.33

ee25btech11021 - Dhanush sagar

September 15, 2025

Problem Statement

Find the equation of the line passing through (1, 2) and making angle 30°with y-axis.

solution

Given point,

$$\mathbf{A} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{0.1}$$

and the line makes an angle of 30° with the y-axis.

The slope of the line is reciprocal of $\tan 30^\circ$:

$$m = \frac{1}{\tan 30^\circ} \tag{0.2}$$

Evaluating, we get:

$$m = \sqrt{3} \tag{0.3}$$

The direction vector of the line is $\begin{pmatrix} 1 \\ m \end{pmatrix}$, hence the normal vector is:

$$\mathbf{n} = \begin{pmatrix} \sqrt{3} \\ -1 \end{pmatrix} \tag{0.4}$$

solution

Equation of the line is given by :

$$\mathbf{n}^\top \mathbf{x} = \mathbf{n}^\top \mathbf{A} \tag{0.5}$$

Substituting the values of **n** and **A**:

$$\begin{pmatrix} \sqrt{3} & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} \sqrt{3} & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{0.6}$$

Evaluating the RHS gives:

$$\begin{pmatrix} \sqrt{3} & -1 \end{pmatrix} \mathbf{x} = \sqrt{3} - 2 \tag{0.7}$$

C Source Code:line solver.c

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

// Function: line through A at 30° with y-axis
// Inputs: Ax, Ay
// Outputs: nx, ny, c in n^T x = c
void line_equation(double Ax, double Ay, double *nx,
\double *ny, double *c) {
    double angle = 30.0 * M_PI / 180.0;
    double m = 1.0 / tan(angle);   // slope
    *nx = m;        // √3
    *ny = -1.0;
    *c  = (*nx)*Ax + (*ny)*Ay;   // n^T A
}
```

Python Script:call line.py

```
import numpy as np
import ctypes
# ----------------------------
# (1) NumPy derivation
# ----------------------------
A = np.array([[1], [2]])   # column vector
print("(0.1) A =\n", A, "\n")
m = 1 / np.tan(np.deg2rad(30))
print("(0.2) m = 1/tan(30°)")
print("(0.3) m =", np.sqrt(3), "\n")n = np.array([[np.sqrt(3)]
print("(0.4) n =\n", n, "\n")
lhs = n.T
rhs = lhs @ A
print("(0.5) n^T x = n^T A\n")
print("(0.6)", lhs, " x = ", lhs, A, "\n")
print("(0.7)", lhs, " x = ", rhs, "\n")
```

Python Script:call line.py

```
# ----------------------------
# (2) Call C shared library
# ----------------------------
lib = ctypes.CDLL("./libline.so")
lib.line_equation.argtypes=[ctypes.c_double, ctypes.c_double,
                            ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double),
                            ctypes.POINTER(ctypes.c_double)]
lib.line_equation.restype = None
nx = ctypes.c_double()
ny = ctypes.c_double()
c  = ctypes.c_double()
lib.line_equation(1.0, 2.0, ctypes.byref(nx), ctypes.byref(ny
print("\n--- From C code ---")
print(f"n = ({nx.value:.3f}, {ny.value:.3f})")
print(f"Equation: {nx.value:.3f}*x1 + {ny.value:.3f}*x2 = {c.
```

Python Script: plot line.py

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
# (1) Call C shared library to get equation
# ------------------------------
lib = ctypes.CDLL("./libline.so")
lib.line_equation.argtypes = [ctypes.c_double, ctypes.c_double
                              ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double)]
lib.line_equation.restype = None
nx = ctypes.c_double()
ny = ctypes.c_double()
c  = ctypes.c_double()
# Pass point A(1,2)
lib.line_equation(1.0, 2.0, ctypes.byref(nx), ctypes.byref(ny)
, ctypes.byref(c))
```

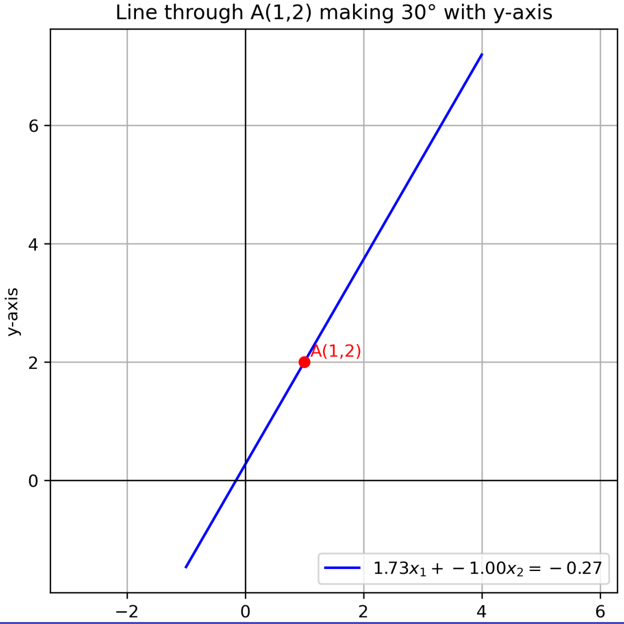
Python Script: plot line.py

```
# -----------------------------
# (2) Generate line points
# -----------------------------
x_vals = np.linspace(-1, 4, 200)
y_vals = (c.value - nx.value * x_vals) / ny.value
# -----------------------------
# (3) Plot line + point A
# -----------------------------
plt.figure(figsize=(6,6))
# Line
plt.plot(x_vals, y_vals, ’b-’, label=fr"${nx.value:.2f}x_1 +
# Point A
A = np.array([1, 2])
plt.scatter(A[0], A[1], color="red", zorder=5)
plt.text(A[0]+0.1, A[1]+0.1, "A(1,2)", color="red")
# Axes formatting
```

Python Script: plot line.py

```
plt.axhline(0, color="black", linewidth=0.8)
plt.axvline(0, color="black", linewidth=0.8)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Line through A(1,2) making 30° with y-axis")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.savefig("line_plot.png", dpi=300, bbox_inches="tight")
plt.show()
```

Result Plot


Line through A(1,2) making 30° with y-axis
y-axis
A(1,2)
$1.73x_1 + -1.00x_2 = -0.27$
−2
0
2
4
6
0
2
4
6