4.3.32

Nipun Dasari - EE25BTECH11042

September 18, 2025

Question

Find the slope of a line which cuts off intercepts of equal length on the axes is. Solve using matrices.

Theoretical Solution

Consider normal form of a line:

$$\mathbf{n}^T\mathbf{x} = c, \text{where } \mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \tag{1}$$

Given that equal intercepts are cut off we get 2 cases:
On substituting the intercepts in place of $\mathbf{x}$:
Case 1: The intercepts are equal $(b = a)$

$$\implies \mathbf{n}^T \begin{pmatrix} a \\ 0 \end{pmatrix} = 1 \implies an_1 = 1 \tag{2}$$

$$\implies \mathbf{n}^T \begin{pmatrix} 0 \\ a \end{pmatrix} = 1 \implies an_2 = 1 \tag{3}$$

$$\tag{4}$$

Theoretical Solution

from (2)

$$n_1 = \frac{1}{a} \tag{5}$$

from (3)

$$n_2 = \frac{1}{a} \therefore \mathbf{n} = \begin{pmatrix} \frac{1}{a} \\ \frac{1}{a} \end{pmatrix} \tag{6}$$

A simpler direction vector would be

$$\mathbf{n}' = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{7}$$

Case 2: The intercepts are negatives of each other $(-b = a)$

$$\implies \mathbf{n}^T \begin{pmatrix} a \\ 0 \end{pmatrix} = 1 \implies an_1 = 1 \tag{8}$$

$$\implies \mathbf{n}^T \begin{pmatrix} 0 \\ -a \end{pmatrix} = 1 \implies -an_2 = 1 \tag{9}$$

(10)

(5)

$$n_1 = \frac{1}{a} \tag{11}$$

(6)

$$n_2 = \frac{-1}{a} \tag{12}$$

$$\therefore \mathbf{n} = \begin{pmatrix} \frac{1}{a} \\ \frac{-1}{a} \end{pmatrix} \tag{13}$$

Theoretical Solution

A simpler direction vector would be

$$\mathbf{n}' = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{14}$$

The direction vector is given (in general) by:

$$\mathbf{n}' = \begin{pmatrix} 1 \\ m \end{pmatrix} \text{ where } m \text{ is slope of given line} \tag{15}$$

On comparing with the obtained direction vectors

$$\therefore m = \pm 1 \tag{16}$$

C Code

```c
#include <stdio.h>
void calculate_slopes(double intercept_a, double* output_slopes) {
    // A line requires non-zero intercepts. If a is zero, we can't calculate a slope.
    if (intercept_a == 0.0) {
        output_slopes[0] = 0.0; // Or some error value like NAN
        output_slopes[1] = 0.0;
        return;
    }
    double slope1 = intercept_a / (-intercept_a);
    double slope2 = (-intercept_a) / (-intercept_a);
    // Fill the output array with the two calculated slopes
```

Python Code using shared output

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

# --- Step 1: Load the shared library ---

lib = ctypes.CDLL('./4.3.32.so')


# --- Step 2: Define the C function
    signature ---
calculate_slopes_func = lib.
    calculate_slopes
calculate_slopes_func.argtypes = [
ctypes.c_double,
np.ctypeslib.ndpointer(dtype=np.double,
    ndim=1, flags='C_CONTIGUOUS')
]
calculate_slopes_func.restype = None
```

Python Code using shared output

```
# --- Step 3: Prepare data and call the C function ---
intercept_a = 4.0
output_slopes = np.zeros(2, dtype=np.double)
calculate_slopes_func(intercept_a, output_slopes)

print(fC function called with intercept a = {intercept_a})
print(fCalculated slopes returned: {output_slopes[0]} and {output_slopes[1]})

# --- Step 4: Plot the results with highlighted axes ---

fig, ax = plt.subplots(figsize=(8, 8))

# Define x values for plotting the lines
x = np.linspace(-6, 6, 400)
```

Python Code using shared output

```
# --- Line 1 (Slope = -1) ---
slope1 = output_slopes[0]
y_intercept1 = intercept_a
y1 = slope1 * x + y_intercept1
ax.plot(x, y1, 'r-', label=f'Line 1: y = {slope1}x
    + {y_intercept1:.0f}')
ax.plot([intercept_a, 0], [0, y_intercept1], 'ro',
    markersize=8)

# --- Line 2 (Slope = 1) ---
slope2 = output_slopes[1]
y_intercept2 = -intercept_a
y2 = slope2 * x + y_intercept2
ax.plot(x, y2, 'b-', label=f'Line 2: y = {slope2}x
    - {y_intercept2:.0f}')
ax.plot([intercept_a, 0], [0, y_intercept2], 'bo',
    markersize=8)
```

Python Code using shared output

```
# --- Highlighting the Coordinate Axes ---
# Remove the default box-like plot frame (
    spines)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

# Move the bottom and left spines to the
    center (0,0)
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')

# Make the new axes bold
ax.spines['bottom'].set_linewidth(1.5)
ax.spines['left'].set_linewidth(1.5)
```

Python Code using shared output

```
# Add arrows to the end of the new axes
ax.plot(1, 0, >k, transform=ax.get_yaxis_transform
    (), clip_on=False)
ax.plot(0, 1, ^k, transform=ax.get_xaxis_transform
    (), clip_on=False)
# --- End of Highlighting Section ---

# --- Plot Styling ---
ax.set_title('Lines with Intercepts of Equal
    Length', fontsize=16)
# Add axis labels at the end of the arrows
ax.set_xlabel('X-axis', fontsize=12, loc='right')
ax.set_ylabel('Y-axis', fontsize=12, loc='top',
    rotation=0)
```

Python Code using shared outut

```
                ax.set_aspect('equal', adjustable='box')
        ax.grid(True, linestyle=':')
        ax.legend()
        ax.set_xlim(-6, 6)
        ax.set_ylim(-6, 6)

        plt.savefig('slope_plot_highlighted.png')

        plt.show()
```

Plot by python using shared output from c

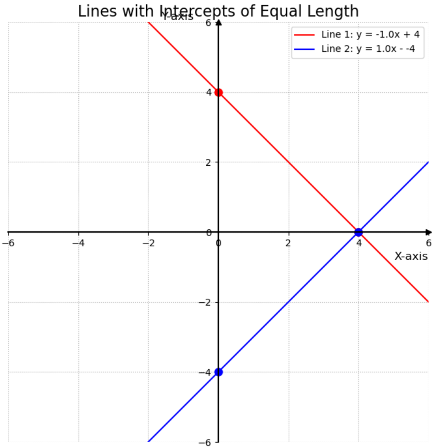

Figure: *