Question

Problem

If the line

$$\frac{x}{a} + \frac{y}{b} = 1$$

passes through the points $(2, -3)$ and $(4, -5)$, then find (a, b).

Solution (Step 1)

For $(2, -3)$:

$$\frac{2}{a} - \frac{3}{b} = 1$$

For $(4, -5)$:

$$\frac{4}{a} - \frac{5}{b} = 1$$

Let

$$u = \frac{1}{a}, \quad v = \frac{1}{b}.$$

Solution (Step 2)

System becomes:

$$2u - 3v = 1, \qquad 4u - 5v = 1$$

Matrix form:

$$\begin{bmatrix} 2 & -3 \\ 4 & -5 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Compute:

$$A^{-1} = \frac{1}{2} \begin{bmatrix} -5 & 3 \\ -4 & 2 \end{bmatrix}.$$

Solution (Step 3)

$$\begin{bmatrix} u \\ v \end{bmatrix} = A^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -2 \\ -2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

Thus,

$$a = -1, \quad b = -1.$$

Equation of line:

$$-x - y = 1.$$

C Code (Part 1)

```c
#include <stdio.h>

int main() {
    double A[2][2] = {{2, -3}, {4, -5}};
    double B[2] = {1, 1};
    double det, u, v, a, b;

    // Determinant of A
    det = A[0][0]*A[1][1] - A[0][1]*A[1][0];
```

C Code (Part 2)

```
    if(det == 0) {
        printf("No unique solution.\n");
        return 0;
    }

    // Cramer's Rule
    u = (B[0]*A[1][1] - B[1]*A[0][1]) / det;
    v = (A[0][0]*B[1] - A[1][0]*B[0]) / det;

    a = 1.0 / u;
    b = 1.0 / v;

    printf("Solution: a = %.2f, b = %.2f\n", a, b);
    return 0;
}
```

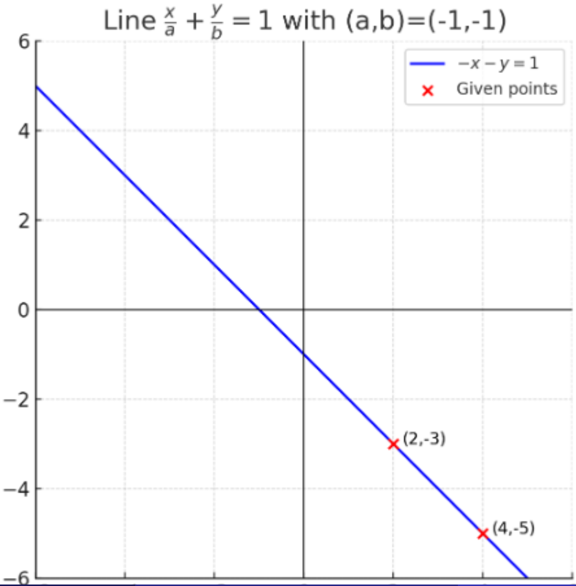
Python Code (Part 1)

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

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

# Define the function signature for points
lib.points.argtypes = [
    ctypes.c_float, # x_0
    ctypes.c_float, # y_0
    ctypes.c_float, # x_end
    ctypes.c_float, # h
    np.ctypeslib.ndpointer(dtype=np.float32, ndim=1),
    np.ctypeslib.ndpointer(dtype=np.float32, ndim=1),
    ctypes.c_int # steps
]
```

Python Code (Part 2)

```python
# Parameters for simulation
x_0, y_0 = 0.0, 2.0
x_end, step_size = 1.0, 0.001
steps = int((x_end - x_0) / step_size) + 1

x_points = np.zeros(steps, dtype=np.float32)
y_points = np.zeros(steps, dtype=np.float32)

# Call the points function
lib.points(x_0, y_0, x_end, step_size,
           x_points, y_points, steps)

# Theoretical solution (C = -2)
def theoretical_solution(x):
    return (-x + 4 - 2*np.exp(x))

x_theory = np.linspace(x_0, x_end, 1000)
y_theory = theoretical_solution(x_theory)
```

Plot



Line $\frac{x}{a} + \frac{y}{b} = 1$ with (a,b)=(-1,-1)
$-x - y = 1$
Given points
(2,-3)
(4,-5)
6
4
2
0
−2
−4
−6