Question

Problem

If the line

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

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

Solution

Substituting the points:

$$\frac{2}{a} - \frac{3}{b} = 1, \tag{1}$$
$$\frac{4}{a} - \frac{5}{b} = 1. \tag{2}$$

Let

$$u = \frac{1}{a}, \quad v = \frac{1}{b}. \tag{3}$$

Then the system becomes

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

Solution

Start with the augmented matrix:

$$\left[\begin{array}{cc|c} 2 & -3 & 1 \\ 4 & -5 & 1 \end{array}\right]$$

Row operations:

$$R_2 \to R_2 - 2R_1, \tag{5}$$
$$R_1 \to R_1 + 3R_2, \tag{6}$$
$$R_1 \to \tfrac{1}{2}R_1. \tag{7}$$

Final reduced form:

$$\left[\begin{array}{cc|c} 1 & 0 & -1 \\ 0 & 1 & -1 \end{array}\right]$$

Final Answer

Thus,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}. \tag{8}$$

Back-substitute:

$$u = \frac{1}{a} = -1 \;\Rightarrow\; a = -1, \tag{9}$$
$$v = \frac{1}{b} = -1 \;\Rightarrow\; b = -1. \tag{10}$$

$$\boxed{(a, b) = (-1, -1)} \tag{11}$$

Hence, the line is

$$x + y = -1. \tag{12}$$

C Code (Part 1)

```
#include <stdio.h>
int main() {
    // Augmented matrix for the system:
    // 2u - 3v = 1
    // 4u - 5v = 1
    double A[2][3] = {
        {2, -3, 1},
        {4, -5, 1}
    };
    // Step 1: R2 -> R2 - 2R1
    A[1][0] = A[1][0] - 2*A[0][0];
    A[1][1] = A[1][1] - 2*A[0][1];
    A[1][2] = A[1][2] - 2*A[0][2];
```

C Code (Part 2)

```c
    // Step 2: R1 -> R1 + 3R2
    A[0][0] = A[0][0] + 3*A[1][0];
    A[0][1] = A[0][1] + 3*A[1][1];
    A[0][2] = A[0][2] + 3*A[1][2];
    // Step 3: R1 -> R1 / 2
    A[0][0] /= 2;
    A[0][1] /= 2;
    A[0][2] /= 2;
    // Extract solution
    double u = A[0][2];
    double v = A[1][2];
    double a = 1.0 / u;
    double b = 1.0 / v;
    printf("u = %lf, v = %lf\n", u, v);
    printf("a = %lf, b = %lf\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("code.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))
```

Python Code (Part 3)

```python
# Generate theory curve
x_theory = np.linspace(x_0, x_end, 1000)
y_theory = theoretical_solution(x_theory)

# Plot results
plt.plot(x_points, y_points, 'ro-',
        markersize=2, linewidth=4, label="sim")
plt.plot(x_theory, y_theory, 'b-',
        linewidth=2, label="theory")

plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True, linestyle="--")
plt.show()
```

Plot of the Line