5.2.4

EE25BTECH11001 - Aarush Dilawri

October 3, 2025

Problem

Question:
Solve the system of equations

$$6x - 3y + 10 = 0 \tag{1}$$

$$2x - y + 9 = 0 \tag{2}$$

Solution Setup

The equation of line:

$$\mathbf{n}^\top \mathbf{x} = c \tag{3}$$

Line L:

$$\begin{pmatrix} 6 & -3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -10 \tag{4}$$

Line K:

$$\begin{pmatrix} 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -9 \tag{5}$$

Matrix Form

These can be combined and written in matrix form:

$$\begin{pmatrix} 6 & -3 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -10 \\ -9 \end{pmatrix} \tag{6}$$

The following augmented matrix can be solved by Gaussian elimination:

$$\left(\begin{array}{cc|c} 6 & -3 & -10 \\ 2 & -1 & -9 \end{array}\right) \xleftrightarrow{R_2 \leftarrow 3R_2 - R_1} \left(\begin{array}{cc|c} 6 & -3 & -10 \\ 0 & 0 & -17 \end{array}\right) \tag{7}$$

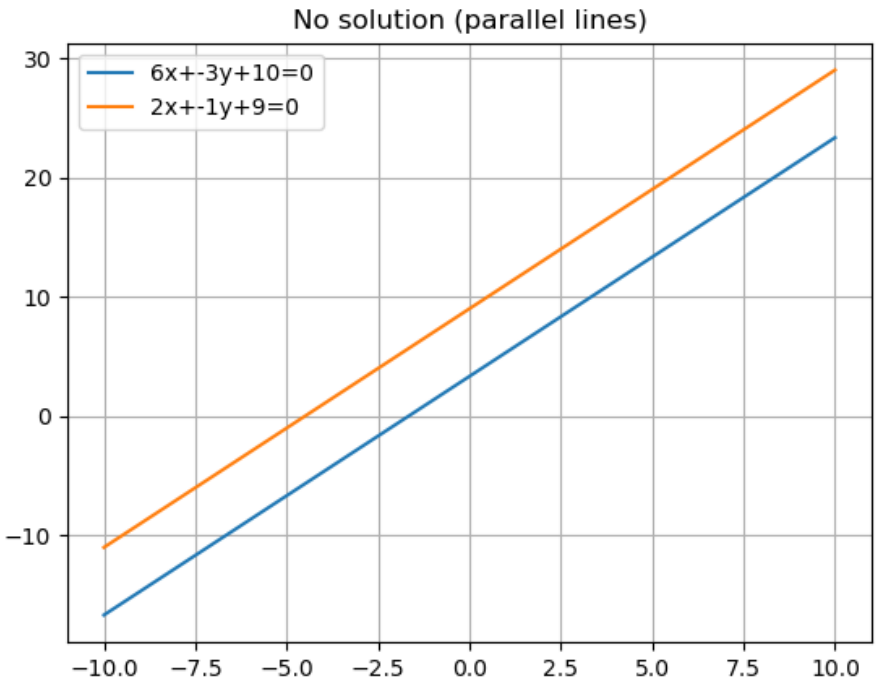
Rank and Solution

The rank of coefficient matrix is 1 whereas the rank of the Augmented matrix is 0. Thus, by Rouche–Capelli theorem:

$$\text{If } \operatorname{rank}(\mathbf{A}) \neq \operatorname{rank}([\mathbf{A} \mid b]) \quad \text{, then the system has no solution.} \tag{8}$$

Hence, there are no solutions to this system of equations.

Figure


No solution (parallel lines)
6x+-3y+10=0
2x+-1y+9=0
30
20
10
0
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

C Code (code.c)

```c
#include <stdio.h>
1, double b1, double c1,
                    double a2, double b2, double c2,
                    double *x, double *y) {
    double det = a1*b2 - a2*b1;
    if(det != 0) {
        *x = (b1*c2 - b2*c1) / det;
        *y = (c1*a2 - c2*a1) / det;
        return 1; // unique solution
    }
    // det = 0, check proportionality
    if(a1*b2 == a2*b1 && a1*c2 == a2*c1 && b1*c2 == b2*c1) {
        return 2; // infinite solutions
    }
    return 0; // no solution
}
```

Python Code (code.py)

```
# code.py
import matplotlib.pyplot as plt
import numpy as np

# Example equations: 6x − 3y + 10 = 0, 2x − y + 9 = 0
a1, b1, c1 = 6, −3, 10
a2, b2, c2 = 2, −1, 9

det = a1*b2 − a2*b1
xs = np.linspace(−10, 10, 400)

if det != 0:
    x = (b1*c2 − b2*c1) / det
    y = (c1*a2 − c2*a1) / det
    res = 1
```

Python Code (code.py)

```
elif (a1*b2 == a2*b1 and a1*c2 == a2*c1 and b1*c2 == b2*c1):
    res = 2
else:
    res = 0

# Plot lines
y1 = -(a1*xs + c1)/b1
y2 = -(a2*xs + c2)/b2
plt.plot(xs, y1, label=f"{a1}x+{b1}y+{c1}=0")
plt.plot(xs, y2, label=f"{a2}x+{b2}y+{c2}=0")
```

Python Code (code.py)

```
if res == 1:
    plt.scatter([x],[y], color="red", zorder=5, label=f"Unique ({x:.2f},{y:.2
        f})")
elif res == 2:
    plt.title("Infinite solutions (same line)")
else:
    plt.title("No solution (parallel lines)")

plt.legend()
plt.grid(True)
plt.show()
```

Python Code (nativecode.py)

```
# nativecode.py
import ctypes
import matplotlib.pyplot as plt
import numpy as np

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

lib.solve_linear.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.
    c_double,
                             ctypes.c_double, ctypes.c_double, ctypes.
                                 c_double,
                             ctypes.POINTER(ctypes.c_double), ctypes.
                                 POINTER(ctypes.c_double)]
lib.solve_linear.restype = ctypes.c_int
```

Python Code (nativecode.py)

```
# Example equations: 6x − 3y + 10 = 0, 2x − y + 9 = 0
a1, b1, c1 = 6, −3, 10
a2, b2, c2 = 2, −1, 9
x = ctypes.c_double()
y = ctypes.c_double()

res = lib.solve_linear(a1,b1,c1,a2,b2,c2,ctypes.byref(x),ctypes.byref(y))

# Plot
xs = np.linspace(−10, 10, 400)
y1 = −(a1∗xs + c1)/b1
y2 = −(a2∗xs + c2)/b2

plt.plot(xs, y1, label=f"{a1}x+{b1}y+{c1}=0")
plt.plot(xs, y2, label=f"{a2}x+{b2}y+{c2}=0")
```

Python Code (nativecode.py)

```
if res == 1:
    plt.scatter([x.value],[y.value], color="red", zorder=5,
                label=f"Unique ({x.value:.2f},{y.value:.2f})")
elif res == 2:
    plt.title("Infinite solutions (same line)")
else:
    plt.title("No solution (parallel lines)")

plt.legend()
plt.grid(True)
plt.show()
```