Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 26, 2025

Problem Statement

Problem 5.3.26 For what value of k, does the system of linear equations

$$2x + 3y = 7, \qquad (k-1)x + (k+2)y = 3k \tag{1.1}$$

have an infinite number of solutions?

Description of Variables used

Symbol	Description	Value/Expression
x, y	Unknown variables	Real numbers
k	Parameter in system	To be determined
$\mathbf{x}$	Unknown vector	$\begin{pmatrix} x \\ y \end{pmatrix}$
A	Coefficient matrix	$\begin{pmatrix} 2 & 3 \\ k-1 & k+2 \end{pmatrix}$
$\mathbf{b}$	RHS vector	$\begin{pmatrix} 7 \\ 3k \end{pmatrix}$
$[A\|b]$	Augmented matrix	$\begin{pmatrix} 2 & 3 & 7 \\ k-1 & k+2 & 3k \end{pmatrix}$

Table

Theoretical Solution

$$\begin{pmatrix} 2 & 3 \\ k-1 & k+2 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 7 \\ 3k \end{pmatrix}, \quad \text{where } \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.1}$$

$$\text{The augmented matrix is } \begin{pmatrix} 2 & 3 & 7 \\ k-1 & k+2 & 3k \end{pmatrix}. \tag{2.2}$$

$$R_2 \to R_2 - \tfrac{k-1}{2} R_1 \tag{2.3}$$

$$\begin{pmatrix} 2 & 3 & 7 \\ k-1 & k+2 & 3k \end{pmatrix} \to \begin{pmatrix} 2 & 3 & 7 \\ 0 & (k+2) - \frac{3}{2}(k-1) & 3k - \frac{7}{2}(k-1) \end{pmatrix}. \tag{2.4}$$

Theoretical Solution

$$= \begin{pmatrix} 2 & 3 & 7 \\ 0 & \frac{-k+7}{2} & \frac{-k+7}{2} \end{pmatrix}. \tag{2.5}$$

$$\text{For infinite solutions: } \operatorname{rank}(A) = \operatorname{rank}([A|b]) < 2. \tag{2.6}$$

$$\tfrac{-k+7}{2} = 0 \quad \Longrightarrow \quad k = 7. \tag{2.7}$$

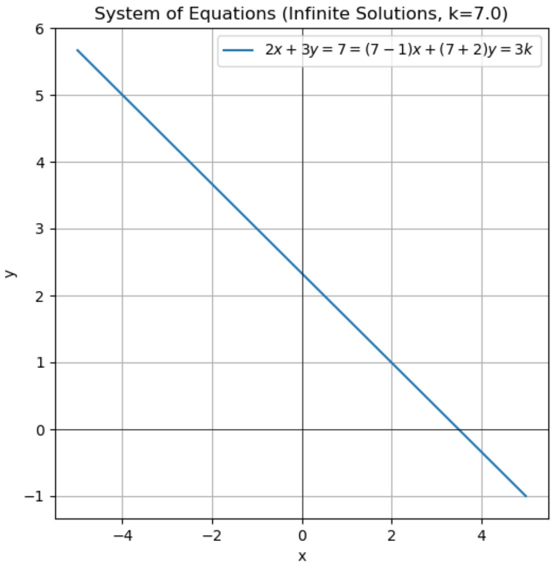
$$\text{When } k = 7, \quad \begin{pmatrix} 2 & 3 & 7 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.8}$$

$$\operatorname{rank}(A) = \operatorname{rank}([A|b]) = 1 < 2. \tag{2.9}$$

Theoretical Solution

$$\therefore \quad \text{The system has infinitely many solutions when } \boxed{k = 7}. \tag{2.10}$$

Plot


System of Equations (Infinite Solutions, k=7.0)
$2x + 3y = 7 = (7 - 1)x + (7 + 2)y = 3k$
y
x
6
5
4
3
2
1
0
−1
−4
−2
0
2
4

Code - C

```c
#include <stdio.h>

// Perform one step of row reduction on a 2x3 augmented matrix
void row_reduce(double A[2][3]) {
    if (A[0][0] != 0) {
        double factor = A[1][0] / A[0][0];
        for (int j = 0; j < 3; j++) {
            A[1][j] = A[1][j] - factor * A[0][j];
        }
    }
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

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

# Load the C shared library
lib = ctypes.CDLL("./row_reduction.so")
lib.row_reduce.argtypes = [
    np.ctypeslib.ndpointer(dtype=np.float64, ndim=2, flags="
        C_CONTIGUOUS")
]

def row_reduce(matrix):
    A = np.array(matrix, dtype=np.float64, order='C')
    lib.row_reduce(A)
    return A
```

Code - Python(with shared C code)

```
# Step 1: Solve for k using SymPy
k = sp.symbols('k')

# Row reduction condition: (-k+7)/2 = 0
expr = (-k + 7) / 2
solution = sp.solve(sp.Eq(expr, 0), k)
k_val = float(solution[0]) # numeric value for plotting

print(f"Calculated k = {k_val}")

# Step 2: Build augmented matrix with that k
A = np.array([
    [2, 3, 7],
    [k_val-1, k_val+2, 3*k_val]
], dtype=np.float64)
```

Code - Python(with shared C code)

```
print("\nOriginal-Augmented-Matrix:")
print(A)

# Step 3: Row reduction via C
reduced = row_reduce(A.copy())
print("\nRow-Reduced-Matrix:")
print(reduced)

# Step 4: Plot only the single line
x_vals = np.linspace(-5, 5, 400)

# General second equation: (k-1)x+(k+2)y=3k or y = (3k-(k-1)x)/(k
    +2)
y = (3*k_val - (k_val-1)*x_vals) / (k_val+2)
```

Code - Python(with shared C code)

```
plt.figure(figsize=(6,6))
plt.plot(x_vals, y, label=rf"$2x+3y=7 = ({int(k_val)}-1)x+({int(k_val)
    }+2)y=3k$")
plt.xlabel("x")
plt.ylabel("y")
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.legend()
plt.grid(True)
plt.title(f"System of Equations (Infinite Solutions, k={k_val})")
plt.savefig("infsols.png")
plt.show()
```

Code - Python only

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp

# Step 1: Solve for k using SymPy
k = sp.symbols('k')
expr = (-k + 7) / 2 # from row reduction condition
solution = sp.solve(sp.Eq(expr, 0), k)
k_val = float(solution[0])
print(f"Calculated k = {k_val}")

# Step 2: Build augmented matrix with NumPy
def augmented_matrix(k):
    return np.array([
        [2, 3, 7],
        [k-1, k+2, 3*k]
    ], dtype=float)
```

Code - Python only

```
A = augmented_matrix(k_val)
print("\nOriginal Augmented Matrix:")
print(A)

# Step 3: Row reduction in NumPy
def row_reduce(A):
    A = A.astype(float).copy()
    if A[0,0] != 0:
        factor = A[1,0] / A[0,0]
        A[1,:] = A[1,:] - factor * A[0,:]
    return A

R = row_reduce(A)
print("\nRow Reduced Matrix:")
print(R)
```

Code - Python only

```python
# Step 4: Plot the single line
x_vals = np.linspace(-5, 5, 400)
# From (k-1)x+(k+2)y=3k or y = (3k-(k-1)x)/(k+2)
y_vals = (3*k_val - (k_val-1)*x_vals) / (k_val+2)

plt.figure(figsize=(6,6))
plt.plot(x_vals, y_vals, label=rf"$2x+3y=7 = ({int(k_val)}-1)x+({int(
    k_val)}+2)y=3k$")
plt.xlabel("x")
plt.ylabel("y")
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.legend()
plt.grid(True)
plt.title(f"System of Equations (Infinite Solutions, k={k_val})")
plt.savefig("pyinfsols.png")
plt.show()
```