Matgeo Presentation - Problem 5.2.24

ee25btech11021 - Dhanush sagar

September 30, 2025

Problem Statement

Solve the following system of linear equations

$px + qy = p - q$

$qx - py = p + q$

solution

Given

$$px + qy = p - q \tag{0.1}$$

$$qx - py = p + q \tag{0.2}$$

The matrix equation for a line is defined as

$$\mathbf{n}^\top \mathbf{x} = c \tag{0.3}$$

where $\mathbf{n}$ is the coefficient vector and $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$

Writing the two lines in matrix form:

$$\begin{pmatrix} p & q \end{pmatrix} \mathbf{x} = p - q \tag{0.4}$$

$$\begin{pmatrix} q & -p \end{pmatrix} \mathbf{x} = p + q \tag{0.5}$$

Combine into a single system:

$$\begin{pmatrix} p & q \\ q & -p \end{pmatrix} \mathbf{x} = \begin{pmatrix} p - q \\ p + q \end{pmatrix} \tag{0.6}$$

solution

Observe that the right-hand side vector can itself be written as the coefficient matrix multiplied by a simple vector:

$$\begin{pmatrix} p & q \\ q & -p \end{pmatrix} \mathbf{x} = \begin{pmatrix} p-q \\ p+q \end{pmatrix} \tag{0.7}$$

$$= \begin{pmatrix} p & q \\ q & -p \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{6}$$

Since the same coefficient matrix appears on both sides, and it is invertible whenever $p^2 + q^2 \neq 0$, we may cancel it to obtain

$$\mathbf{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{0.8}$$

Hence the solution is:

$$x = 1, \quad y = -1 \tag{0.9}$$

C Source Code:generate system.c

```
#include <stdio.h>
// Function to generate coefficient matrix A and RHS vector b
// System:  px + qy = p - q
//          qx - py = p + q
void generate_system(double p, double q, double *A, double *b)
   A[0] = p;    A[1] = q;
    A[2] = q;    A[3] = -p;
b[0] = p - q;
    b[1] = p + q;
}
int main() {
    double p = 2, q = 3;
    double A[4], b[2];
    generate_system(p, q, A, b);
    printf("Matrix A = [[%lf, %lf], [%lf, %lf]]\n", A[0], A[1
    printf("Vector b = [%lf, %lf]\n", b[0], b[1]);
    return 0;}
```

Python Script:solve system.py

```
import ctypes
import numpy as np

# Load shared C library
lib = ctypes.CDLL("./generate_system.so")
lib.generate_system.argtypes = [ctypes.c_double, ctypes.c_dou
                                ctypes.POINTER(ctypes.c_double
                                ctypes.POINTER(ctypes.c_double

def generate_system(p, q):
    A = (ctypes.c_double * 4)()
    b = (ctypes.c_double * 2)()
    lib.generate_system(p, q, A, b)
    A_np = np.array([[A[0], A[1]], [A[2], A[3]]])
    b_np = np.array([b[0], b[1]])
    return A_np, b_np
```

Python Script:solve system..py

```
# Example usage
p, q = 2, 3
A, b = generate_system(p, q)

print("Given system:")
print(f"{p}x + {q}y = {p-q}")
print(f"{q}x - {p}y = {p+q}\n")

print("Matrix form:")
print("A =", A)
print("b =", b, "\n")

# Solve using normal equations: (A^T A)x = A^T b
lhs = A.T @ A
rhs = A.T @ b
x = np.linalg.solve(lhs, rhs)
print("Solution vector x =", x)
```

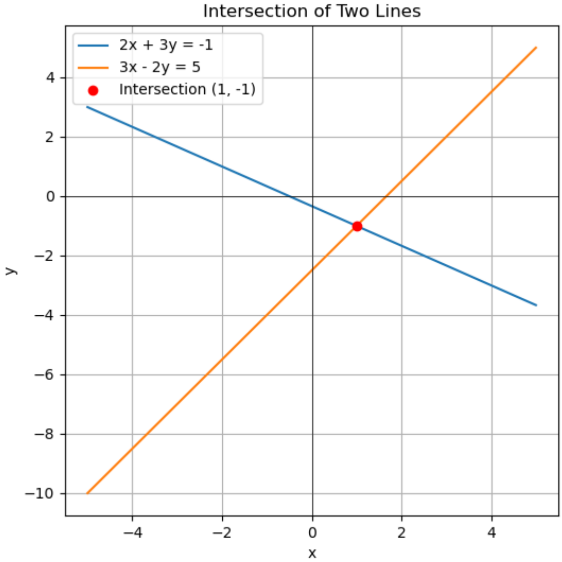
Python Script: plot system.py

```
import numpy as np
import matplotlib.pyplot as plt
p, q = 2, 3
x_sol, y_sol = 1, -1  # always same solution
x_vals = np.linspace(-5, 5, 400)
# Line 1: px + qy = p - q → y = (p - q - p*x)/q
y1 = (p - q - p*x_vals) / q
# Line 2: qx - py = p + q → y = (q*x - (p+q))/p
y2 = (q*x_vals - (p + q)) / p
plt.figure(figsize=(6,6))
plt.plot(x_vals, y1, label=f"{p}x + {q}y = {p-q}")
plt.plot(x_vals, y2, label=f"{q}x - {p}y = {p+q}")
```

Python Script: plot system.py

```
# Intersection point
plt.plot(x_sol, y_sol, 'ro', label="Intersection (1, -1)")
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.title("Intersection of Two Lines")
plt.show()
```

Result Plot



Intersection of Two Lines
2x + 3y = -1
3x - 2y = 5
Intersection (1, -1)
4
2
0
−2
−4
−6
−8
−10
y
−4
−2
0
2
4
x