MatGeo Presentation - Problem 1.6.14

EE25BTECH11064 - Yojit Manral

Question

Prove or disprove: The straight line $5x + 4y = 0$ passes through the point of intersection of the straight lines $x + 2y - 10 = 0$ and $2x + y + 5 = 0$.

Solution

→ Let

$$\mathbf{n_1}^T\mathbf{x} = c_1 \qquad \mathbf{n_2}^T\mathbf{x} = c_2 \qquad \mathbf{n_3}^T\mathbf{x} = c_3$$

$$\mathbf{n_1} = \begin{pmatrix}5\\4\end{pmatrix} \qquad c_1 = 0 \tag{0.1}$$

$$\mathbf{n_2} = \begin{pmatrix}1\\2\end{pmatrix} \qquad c_2 = 10 \tag{0.2}$$

$$\mathbf{n_3} = \begin{pmatrix}2\\1\end{pmatrix} \qquad c_3 = -5 \tag{0.3}$$

→ The three lines are concurrent if there exists some $\mathbf{x}$ such that

$$\begin{pmatrix}\mathbf{n_1}^T\\\mathbf{n_2}^T\\\mathbf{n_3}^T\end{pmatrix}\mathbf{x} = \begin{pmatrix}c_1\\c_2\\c_3\end{pmatrix} \tag{0.4}$$

$$\mathbf{A}\quad\mathbf{x} = \quad\mathbf{B} \tag{0.5}$$

Solution

$\rightarrow$ A unique $\mathbf{x}$ exists iff the matrix $\mathbf{A}$ is a full-rank matrix (i.e., rank 2 for a system of 2 variables).

$$\mathbf{A} = \begin{pmatrix} 5 & 4 \\ 1 & 2 \\ 2 & 1 \end{pmatrix} \xrightarrow[R_2 \leftrightarrow R_2 - R_1]{R_1 \leftrightarrow (1/5) R_1} \begin{pmatrix} 1 & 4/5 \\ 0 & 6/5 \\ 2 & 1 \end{pmatrix} \tag{0.6}$$

$$\xrightarrow[R_2 \leftrightarrow (5/6) R_2]{R_3 \leftrightarrow R_3 - 2R_1} \begin{pmatrix} 1 & 4/5 \\ 0 & 1 \\ 0 & -3/5 \end{pmatrix} \tag{0.7}$$

$$\xrightarrow[R_1 \leftrightarrow R_1 - (4/5) R_2]{R_3 \leftrightarrow R_3 + (3/5) R_2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{0.8}$$

Only 2 nonzero rows $\longrightarrow$ Rank of $\mathbf{A}$ is 2 $\longrightarrow$ The 3 lines are concurrent

Solution

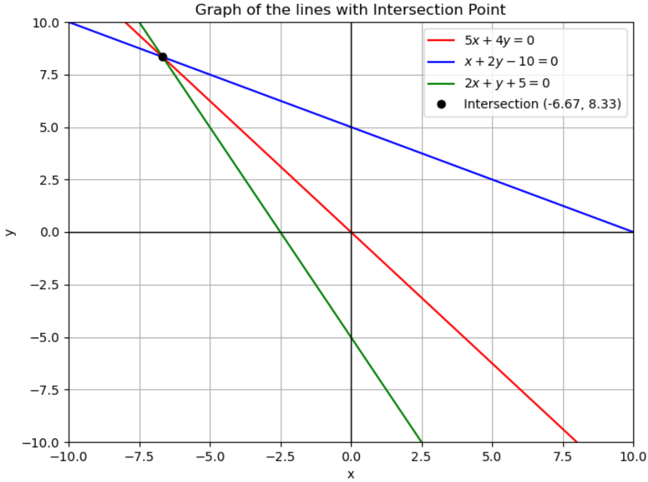

Figure: Plot of the three lines

File: points.c

```
#include <stdio.h>

int main() {
  FILE *fp;

  // -------------------
  // Question 4.13.68
  // -------------------


  fp = fopen("points.dat", "w");
  fprintf(fp, "%d,%d,%d\n", 5, 4, 0); // n1 c1
  fprintf(fp, "%d,%d,%d\n", 1, 2, 10); // n2 c2
  fprintf(fp, "%d,%d,%d\n", 2, 1, -5); // n3 c3
  fclose(fp);
  return 0;
}
```

File: call_c.py

```python
import subprocess

# Compile the C program
subprocess.run(["gcc", "points.c", "-o", "points"])

# Run the compiled C program
result = subprocess.run(["./points"], capture_output=True, text=True)

# Print the output from the C program
print(result.stdout)
```

File: plot.py

```
import numpy as np
import matplotlib.pyplot as plt
from sympy import symbols, Eq, solve

# Define the symbols
x, y = symbols('x y')

# Define the equations
eq1 = Eq(5*x + 4*y, 0) # 5x + 4y = 0
eq2 = Eq(x + 2*y - 10, 0) # x + 2y - 10 = 0

# Solve the system of two equations
solution = solve((eq1, eq2), (x, y))

# Extract the intersection point
intersection_x = solution[x]
intersection_y = solution[y]

# Define the lines
def line1(x):
    return (-5*x) / 4

def line2(x):
    return (10 - x) / 2

def line3(x):
    return (-2*x - 5)

# Generate x values
x_vals = np.linspace(-10, 10, 400)
```

File: plot.py

```
# Plot the lines
plt.figure(figsize=(8, 6))
plt.plot(x_vals, line1(x_vals), label=r'$5x␣+␣4y␣=␣0$', color='r')
plt.plot(x_vals, line2(x_vals), label=r'$x␣+␣2y␣-␣10␣=␣0$', color='b')
plt.plot(x_vals, line3(x_vals), label=r'$2x␣+␣y␣+␣5␣=␣0$', color='g')

# Mark the point of intersection
plt.plot(intersection_x, intersection_y, 'ko', label=f'Intersection␣({intersection_x:.2f},␣{intersection_y
       :.2f})')

# Set plot limits and labels
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black',linewidth=1)
plt.axvline(0, color='black',linewidth=1)
plt.grid(True)

# Labeling
plt.title('Graph␣of␣the␣lines␣with␣Intersection␣Point')
plt.xlabel('x')
plt.ylabel('y')

# Adding a legend
plt.legend()

# Show plot
plt.show()
```