# 4.11.34

Rushil Shanmukha Srinivas
EE25BTECH11057
Electrical Enggineering ,
IIT Hyderabad.

September 18, 2025

1. Problem
2. Solution
   - Obtaining Vertices and Finding Area
   - Plots
3. C Code
4. Python Code

# Problem Statement

**Question** : Find the area of the region bounded by the lines $3x-2y+1=0, 2x+3y-21=0$ and $x-5y+9=0$ .

# Obtaining Vertices and Finding Area

**Solution:** Given three lines are

$$\begin{pmatrix} 3 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -1 \Longrightarrow \mathbf{n}^\top \mathbf{x} = -1 \tag{3.1}$$

$$\begin{pmatrix} 2 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 21 \Longrightarrow \mathbf{m}^\top \mathbf{x} = 21 \tag{3.2}$$

$$\begin{pmatrix} 1 & -5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -9 \Longrightarrow \mathbf{p}^\top \mathbf{x} = -9 \tag{3.3}$$

The three lines form a triangle. The vertices of triangle are obtained by **Intersection of :**

$$\mathbf{n}^\top \mathbf{x} = -1 \; \textit{and} \; \mathbf{m}^\top \mathbf{x} = 21 \tag{3.4}$$

The augmented system in matrix form is

$$\left(\begin{array}{cc|c} 3 & -2 & -1 \\ 2 & 3 & 21 \end{array}\right) \xrightarrow{R_2 \longrightarrow 3R_2 - 2R_1} \left(\begin{array}{cc|c} 3 & -2 & -1 \\ 0 & 13 & 65 \end{array}\right) \tag{3.5}$$

*From the second row we get* $y = 5$ *so* $x = 3 \Longrightarrow \mathbf{A} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$

$$\mathbf{m}^\top \mathbf{x} = 21 \text{ and } \mathbf{p}^\top \mathbf{x} = -9 \tag{3.6}$$

The augmented matrix is

$$\left(\begin{array}{cc|c} 2 & 3 & 21 \\ 1 & -5 & -9 \end{array}\right) \xrightarrow{R_2 \longrightarrow 2R_2 - R_1} \left(\begin{array}{cc|c} 2 & 3 & 21 \\ 0 & -13 & -39 \end{array}\right) \tag{3.7}$$

*From the second row we get* $y = 3$ *so* $x = 6 \Longrightarrow \mathbf{B} = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$

$$\mathbf{p}^\top \mathbf{x} = -9 \text{ and } \mathbf{n}^\top \mathbf{x} = -1 \tag{3.8}$$

The augmented matrix is

$$\left(\begin{array}{cc|c} 1 & -5 & -9 \\ 3 & -2 & -1 \end{array}\right) \xrightarrow{R_2 \longrightarrow R_2 - 3R_1} \left(\begin{array}{cc|c} 1 & -5 & -9 \\ 0 & 13 & 26 \end{array}\right) \tag{3.9}$$

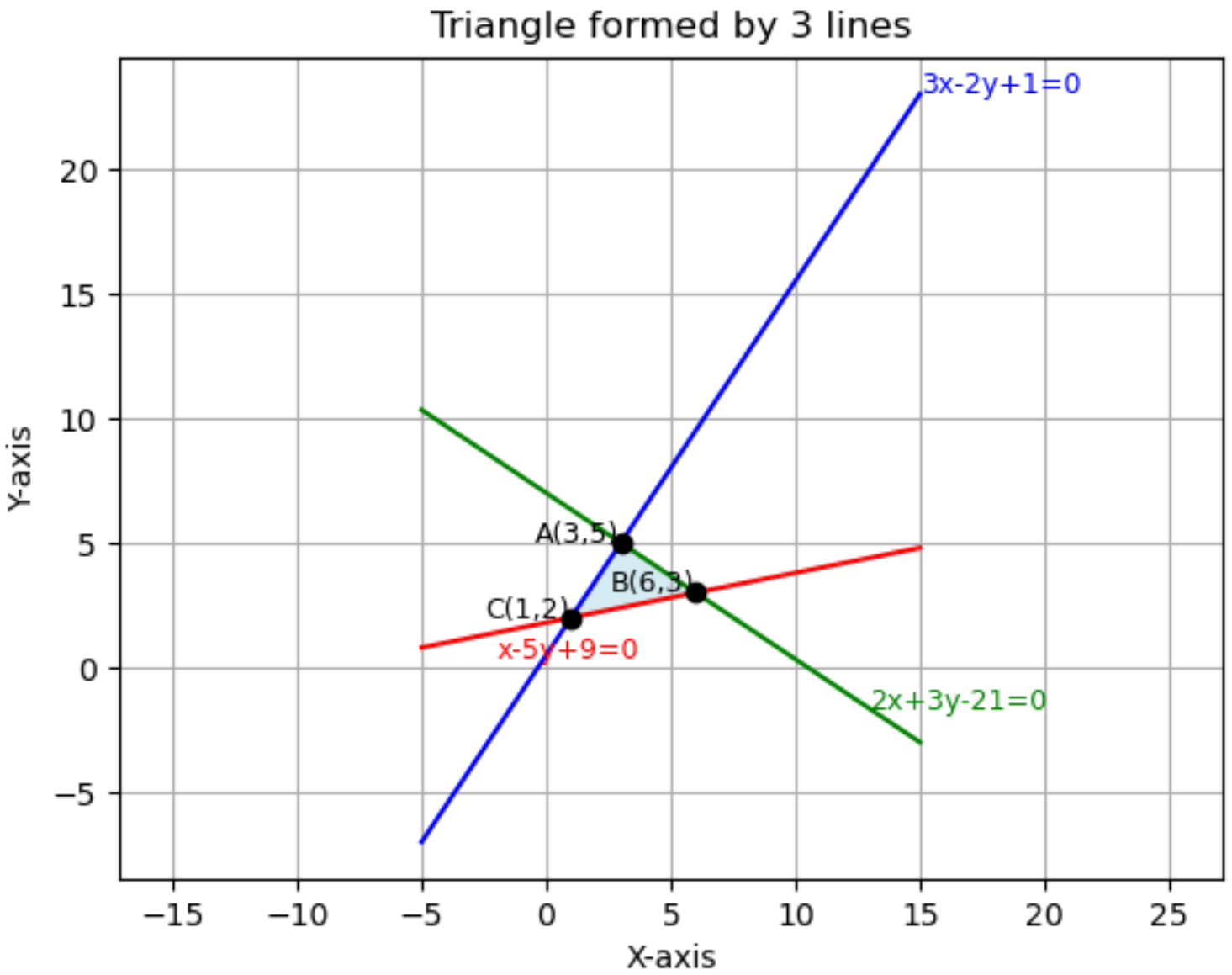
*From the second row we get* $y = 2$ *so* $x = 1 \Longrightarrow \mathbf{C} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$$\mathbf{A}-\mathbf{B}=\begin{pmatrix}-3\\2\end{pmatrix},\mathbf{A}-\mathbf{C}=\begin{pmatrix}2\\3\end{pmatrix} \tag{3.10}$$

$$\|(\mathbf{A}-\mathbf{B})\times(\mathbf{A}-\mathbf{C})\|=|\begin{vmatrix}-3 & 2\\2 & 3\end{vmatrix}|=|-9-4|=|-13|=13 \tag{3.11}$$

$$\textit{Area of the triangle}=\frac{1}{2}\|(\mathbf{A}-\mathbf{B})\times(\mathbf{A}-\mathbf{C}\|=\frac{13}{2} \tag{3.12}$$

# Plots


Triangle formed by 3 lines
3x-2y+1=0
x-5y+9=0
2x+3y-21=0
A(3,5)
B(6,3)
C(1,2)
X-axis
Y-axis
−15
−10
−5
0
5
10
15
20
25
20
15
10
5
0
−5

# C Code

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void solveIntersection(double a1,double b1,double c1,
                       double a2,double b2,double c2,
                       double *x,double *y) {
    double det = a1*b2 - a2*b1;
    *x = (b1*(-c2) - b2*(-c1)) / det;
    *y = (a2*(-c1) - a1*(-c2)) / det;}
double triangleArea() {
    double x1,y1,x2,y2,x3,y3;

    // Line1 & Line2
    solveIntersection(3,-2,1, 2,3,-21,&x1,&y1);
    // Line2 & Line3
    solveIntersection(2,3,-21, 1,-5,9,&x2,&y2);
```

```
    // Line1 & Line3
    solveIntersection(3,-2,1, 1,-5,9,&x3,&y3);

    double area = 0.5 * fabs(x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2));
    return area;
}
```

# Python : call_c.py

```
import ctypes
import os

# Path to the compiled shared object
lib_path = os.path.abspath("./libtriangle.so")

# Load the shared library
lib = ctypes.CDLL(lib_path)

# Specify return type of the triangleArea function
lib.triangleArea.restype = ctypes.c_double

# Call the C function
area = lib.triangleArea()
# Print the solution
print("The area of the triangle formed by the given lines is:", area)
```

# Python Code for Plotting

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

# Solve intersection of two lines ax + by + c = 0
def intersection(a1,b1,c1, a2,b2,c2):
    A = np.array([[a1,b1],[a2,b2]])
    B = np.array([-c1,-c2])
    x, y = np.linalg.solve(A,B)
    return x, y

# Line equations
# 1) 3x - 2y + 1 = 0
# 2) 2x + 3y - 21 = 0
# 3) x - 5y + 9 = 0

# Find vertices
A = intersection(3,-2,1, 2,3,-21)
```

```
B = intersection(2,3,−21, 1,−5,9)
C = intersection(3,−2,1, 1,−5,9)

x1,y1 = A
x2,y2 = B
x3,y3 = C
# Triangle vertices for plotting
triangle_x = [x1, x2, x3, x1]
triangle_y = [y1, y2, y3, y1]

# Plotting range
x_vals = np.linspace(−5, 15, 400)

# Line 1: 3x − 2y + 1 = 0 −> y = (3x+1)/2
y1_line = (3*x_vals + 1)/2
plt.plot(x_vals, y1_line, color="blue")
plt.text(15, (3*15+1)/2, "3x−2y+1=0", fontsize=9, color="blue") #
    moved right
```

```
# Line 2: 2x + 3y − 21 = 0 −> y = (21−2x)/3
y2_line = (21 − 2*x_vals)/3
plt.plot(x_vals, y2_line, color="green")
plt.text(13, (21−2*13)/3, "2x+3y−21=0", fontsize=9, color="green")

# Line 3: x − 5y + 9 = 0 −> y = (x+9)/5
y3_line = (x_vals + 9)/5
plt.plot(x_vals, y3_line, color="red")
plt.text(−2, ((−2+9)/5) − 1.0, "x−5y+9=0", fontsize=9, color="red")
    # moved down

# Plot triangle
plt.fill(triangle_x, triangle_y, color="lightblue", alpha=0.5)

# Mark vertices
plt.scatter([x1,x2,x3],[y1,y2,y3], color="black", zorder=5)
```

```
plt.text(x1,y1,f"A({int(round(x1))},{int(round(y1))})", fontsize=9, ha="
    right")
plt.text(x2,y2,f"B({int(round(x2))},{int(round(y2))})", fontsize=9, ha="
    right")
plt.text(x3,y3,f"C({int(round(x3))},{int(round(y3))})", fontsize=9, ha="
    right")

plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Triangle formed by 3 lines")
plt.grid(True)
plt.axis("equal")
plt.savefig("../figs/fig8.png")
plt.show()
```