4.11.39

Vivek K Kumar - EE25BTECH11062

October 2, 2025

Question

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

Variables used

(1)

Point	Value
$\mathbf{n_1}$	$\begin{pmatrix} 3 \\ -2 \end{pmatrix}$
$\mathbf{n_2}$	$\begin{pmatrix} 2 \\ 3 \end{pmatrix}$
$\mathbf{n_3}$	$\begin{pmatrix} 1 \\ -5 \end{pmatrix}$
$\mathbf{c_1}$	-1
$\mathbf{c_2}$	21
$\mathbf{c_3}$	-9

Table: Variables used

Solution

The given lines can be represented as

$$\mathbf{n_1}^\top \mathbf{x} = c_1 \tag{2}$$

$$\mathbf{n_2}^\top \mathbf{x} = c_2 \tag{3}$$

$$\mathbf{n_3}^\top \mathbf{x} = c_3 \tag{4}$$

Let the points of intersections of the given lines be represented as $\mathbf{A}, \mathbf{B}, \mathbf{C}$

$$\begin{pmatrix} \mathbf{n_1} & \mathbf{n_2} \end{pmatrix}^\top \mathbf{A} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} \mathbf{n_2} & \mathbf{n_3} \end{pmatrix}^\top \mathbf{B} = \begin{pmatrix} c_2 \\ c_3 \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} \mathbf{n_3} & \mathbf{n_1} \end{pmatrix}^\top \mathbf{C} = \begin{pmatrix} c_3 \\ c_1 \end{pmatrix} \tag{7}$$

Solution

The area of the triangle can be then represented as

$$\frac{1}{2}\left\|\mathbf{A}-\mathbf{B}\right\|\left\|\mathbf{C}-\mathbf{B}\right\|\sqrt{1-\left(\frac{\mathbf{n_2}^{\top}\mathbf{n_3}}{\left\|\mathbf{n_2}\right\|\left\|\mathbf{n_3}\right\|}\right)^2} \tag{8}$$

Solving for $\mathbf{A}, \mathbf{B}, \mathbf{C}$

$$\begin{pmatrix} 3 & -2 \\ 2 & 3 \end{pmatrix}\mathbf{A} = \begin{pmatrix} -1 \\ 21 \end{pmatrix} \tag{9}$$

$$\implies \left(\begin{array}{cc|c} 3 & -2 & -1 \\ 2 & 3 & 21 \end{array}\right) \xleftrightarrow{R_2 \leftarrow R_2 - 2/3R_1} \left(\begin{array}{cc|c} 3 & -2 & -1 \\ 0 & 13/3 & 65/3 \end{array}\right) \tag{10}$$

$$\mathbf{A} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} \tag{11}$$

Solution

$$\begin{pmatrix} 2 & 3 \\ 1 & -5 \end{pmatrix} \mathbf{B} = \begin{pmatrix} 21 \\ -9 \end{pmatrix} \tag{12}$$

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

$$\mathbf{B} = \begin{pmatrix} 6 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 1 & -5 \\ 3 & -2 \end{pmatrix} \mathbf{C} = \begin{pmatrix} -9 \\ -1 \end{pmatrix} \tag{14}$$

$$\implies \left(\begin{array}{cc|c} 1 & -5 & -9 \\ 3 & -2 & -1 \end{array}\right) \xleftrightarrow{R_2 \leftarrow R_2 - 3R_1} \left(\begin{array}{cc|c} 1 & -5 & -9 \\ 0 & 13 & 26 \end{array}\right) \tag{15}$$

$$\mathbf{C} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{16}$$

Solution

Area of the triangle from the previous equations is

$$\frac{1}{2}\left\|\begin{pmatrix}-3 & 2\end{pmatrix}^{\top}\right\|\left\|\begin{pmatrix}-5 & -1\end{pmatrix}^{\top}\right\|\sqrt{1-\left(\frac{-13}{13\sqrt{2}}\right)^{2}}=\frac{13}{2} \tag{17}$$

Python - Importing libraries and checking system

```
import sys
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import math

from libs.line.funcs import *
from libs.triangle.funcs import *
from libs.conics.funcs import circ_gen

import subprocess
import shlex

print('Using termux?(y/n)')
y = input()
```

Python - Writing required points and direction vectors

```
m1 = np.array([2, 3]).reshape(-1,1)
m2 = np.array([-3, 2]).reshape(-1,1)
m3 = np.array([5, 1]).reshape(-1,1)
A = np.array([3, 5]).reshape(-1,1)
B = np.array([6, 3]).reshape(-1,1)
C = np.array([1, 2]).reshape(-1,1)
```

Python - Generating points and plotting

```
p_l1 = line_gen(A-1.5*m1, A+1.5*m1)
p_l2 = line_gen(B-1.5*m2, B+1.5*m2)
p_l3 = line_gen(C-1.5*m3, C+1.5*m3)

fig = plt.figure()
ax = fig.add_subplot(111)

ax.plot(p_l1[0, :], p_l1[1, :], label = '3x-2y+1=0')
ax.plot(p_l2[0, :], p_l2[1, :], label = '2x+3y-21=0')
ax.plot(p_l3[0, :], p_l3[1, :], label = 'x-5y+9=0')
```

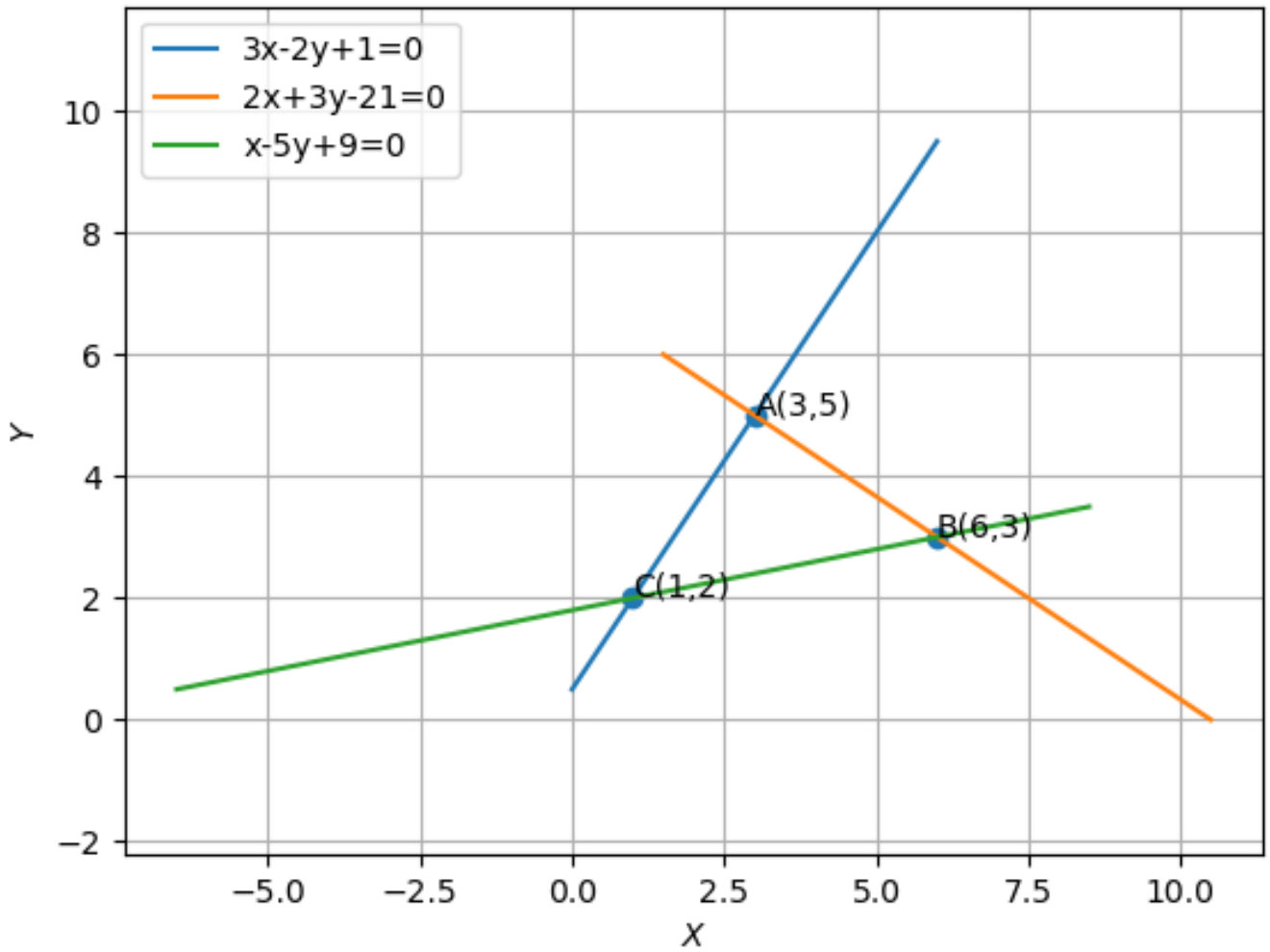
Python - Labelling points

```
pts = np.block([A, B, C])
labels = ['A(3,5)', 'B(6,3)', 'C(1,2)']
ax.scatter(pts[0, :], pts[1, :])
for i, txt in enumerate(labels):
        ax.text(pts[0, i], pts[1, i], s=txt)

ax.set_xlabel('$X$')
ax.set_ylabel('$Y$')
ax.legend(loc='best')
ax.grid(True)
ax.axis('equal')
```

Python - Saving figure and opening it

```
fig.savefig('../figs/fig.png')
print('Saved figure to ../figs/fig.png')

if(y == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig.png'))
else:
    subprocess.run(["open", "../figs/fig.png"])
```

Plot-Using only Python



C Code (0) - Importing libraries

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "libs/matfun.h"
#include "libs/geofun.h"
```

C Code (1) - Function to Generate Points on a Line

```c
void point_gen(FILE *p_file, double **A, double **B, int rows,
    int cols, int npts){
    for(int i = 0; i <= npts; i++){
     double **output = Matadd(A, Matscale(Matsub(B, A, rows, cols
         ), rows, cols, (double)i/npts), rows, cols);
     fprintf(p_file, "%lf, %lf\n", output[0][0], output[1][0]);
     freeMat(output, rows);
    }
}
```

C Code (2) - Function to write points b/w given points to a file

```c
void write_points(double x1, double y1, double x2, double y2,
    double x3, double y3, int npts){
    int m = 2;
    int n = 1;

    double **A = createMat(m, n);
    double **B = createMat(m, n);
    double **C = createMat(m, n);

    B[0][0] = x2;
    B[1][0] = y2;
```

C Code (2) - Function to write points b/w given 2 points to a file

```
A[0][0] = x1;
A[1][0] = y1;
C[0][0] = x3;
C[1][0] = y3;
double **L1_1 = Matsub(A, Matscale(Matsub(B, A, m, n), m, n,
    -1.5), m, n);
double **L1_2 = Matsub(A, Matscale(Matsub(B, A, m, n), m, n,
    1.5), m, n);
double **L2_1 = Matsub(B, Matscale(Matsub(C, B, m, n), m, n,
    -1.5), m, n);
double **L2_2 = Matsub(B, Matscale(Matsub(C, B, m, n), m, n,
    1.5), m, n);
double **L3_1 = Matsub(C, Matscale(Matsub(C, A, m, n), m, n,
    -1.5), m, n);
double **L3_2 = Matsub(C, Matscale(Matsub(C, A, m, n), m, n,
    1.5), m, n);
```

C Code (2) - Function to write points b/w given 2 points to a file

```
    FILE *p_file;
    p_file = fopen("plot.dat", "w");

    if(p_file == NULL)
        printf("Error opening one of the data files\n");
    point_gen(p_file, L1_1, L1_2, m, n, npts);
    point_gen(p_file, L2_1, L2_2, m, n, npts);
    point_gen(p_file, L3_1, L3_2, m, n, npts);
```

C Code (2) - Function to write points b/w 2 points to a file

```
    freeMat(A, m);
    freeMat(B, m);
    freeMat(C, m);
    freeMat(L1_1, m);
    freeMat(L1_2, m);
    freeMat(L2_1, m);
    freeMat(L2_2, m);
    freeMat(L3_1, m);
    freeMat(L3_2, m);

    fclose(p_file);
}
```

Python Code (0) - Importing libraries and checking system

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
import os
import sys
import subprocess
import math

print('Using termux? (y/n)')
termux = input()
```

Python Code (1) - Using Shared Object

```
lib_path = os.path.join(os.path.dirname(__file__), 'plot.so')
my_lib = ctypes.CDLL(lib_path)

my_lib.write_points.argtypes = [ctypes.c_double, ctypes.c_double,
     ctypes.c_double, ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_int]
my_lib.write_points.restype = None
A = np.array([3, 5]).reshape(-1, 1)
B = np.array([6, 3]).reshape(-1, 1)
C = np.array([1, 2]).reshape(-1, 1)
npts = 20000
```

Python Code (2) - Loading points and plotting them

```
fig = plt.figure()
ax = fig.add_subplot(111)
labels = ['3x-2y+1=0', '2x+3y-21=0', 'x-5y+9=0']
point_labels = ['A(3, 5)', 'B(6, 3)', 'C(1,2)']
pts = np.block([A, B, C])

for i,label in enumerate(labels):
    points = np.loadtxt('plot.dat', delimiter = ',', usecols
        =(0,1))[i*(npts+1):(i+1)*(npts+1)]
    ax.plot(points[:, 0], points[:, 1], label = label)
    ax.text(pts[0, i], pts[1, i], s=point_labels[i])
```

Python Code (3) - Labelling plot

```
        ax.set_xlabel('$X$')
        ax.set_ylabel('$Y$')
        ax.legend(loc='best')
        ax.grid()
        ax.axis('equal')
```

Python Code (4) - Saving and displaying plot

```
fig.savefig('../figs/fig2.png')
print('Saved figure to ../figs/fig2.png')

if(termux == 'y'):
    subprocess.run(shlex.split('termux-open ../figs/fig2.png'))
else:
    subprocess.run(["open", "../figs/fig2.png"])
```

Plot-Using Both C and Python