4.11.9

Sai Sreevallabh - EE25BTECH11031

October 1, 2025

Question

Find the area of the triangle $\triangle ABC$ bounded by the lines $4x - y + 5 = 0$, $x + y - 5 = 0$ and $x - 4y + 5 = 0$.

Theoretical Solution

Given lines can be written as:

$$\begin{pmatrix} 4 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -5 \tag{1}$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 5 \tag{2}$$

$$\begin{pmatrix} 1 & -4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -5 \tag{3}$$

Theoretical Solution

Solving equations (1) and (2) to get the point of intersection **A**:

$$\begin{pmatrix} 1 & 1 \\ 4 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 5 \\ -5 \end{pmatrix} \tag{4}$$

Making the Augmented Matrix and converting to echelon form

$$\left(\begin{array}{cc|c} 1 & 1 & 5 \\ 4 & -1 & -5 \end{array}\right) \xleftrightarrow{R_2 \to R_2 - 4R_1} \left(\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & -5 & -25 \end{array}\right) \tag{5}$$

We get

$$\mathbf{A} = \begin{pmatrix} 0 \\ 5 \end{pmatrix} \tag{6}$$

Theoretical Solution

Solving equations (2) and (3) to get the point of intersection **B**

$$\left(\begin{array}{cc|c} 1 & 1 & 5 \\ 1 & -4 & -5 \end{array}\right) \xleftrightarrow{R_2 \to R_2 - R_1} \left(\begin{array}{cc|c} 1 & 1 & 5 \\ 0 & -5 & -10 \end{array}\right) \tag{7}$$

We get

$$\mathbf{B} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \tag{8}$$

Theoretical Solution

Solving equations (1) and (3) to get the point of intersection **C**

$$\left(\begin{array}{cc|c} 1 & -4 & -5 \\ 4 & -1 & 5 \end{array}\right) \xleftrightarrow{R_2 \to R_2 - 4R_1} \left(\begin{array}{cc|c} 1 & -4 & -5 \\ 0 & 15 & 15 \end{array}\right) \tag{9}$$

We get

$$\mathbf{C} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{10}$$

Theoretical Solution

The vertices of the triangle are

$$\mathbf{A} = \begin{pmatrix} 0 \\ 5 \end{pmatrix} , \mathbf{B} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} , \mathbf{C} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{11}$$

Now

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} 3 \\ -3 \end{pmatrix} \text{ and } \mathbf{C} - \mathbf{A} = \begin{pmatrix} -1 \\ -4 \end{pmatrix} \tag{12}$$

Theoretical Solution

Area of the triangle $\triangle ABC$ is given by

$$\frac{1}{2}\left\|(\mathbf{B}-\mathbf{A})\times(\mathbf{C}-\mathbf{A})\right\| \tag{13}$$

$$= \frac{1}{2}\left\|\begin{pmatrix}3\\-3\end{pmatrix}\times\begin{pmatrix}-1\\-4\end{pmatrix}\right\| \tag{14}$$

$$= \frac{15}{2} \tag{15}$$

Hence,

$$ar\left(\triangle ABC\right) = \frac{15}{2} \tag{16}$$

C Code - Finding x-coordinate of Point of Intersection

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

double POI_x(double a1, double b1, double c1, double a2, double
    b2, double c2){

    double x = (b1*c2-b2*c1)/(a1*b2-a2*b1);

    return x;
}
```

C Code - Finding y-coordinate of Point of Intersection

```c
double POI_y(double a1, double b1, double c1, double a2, double
    b2, double c2){

    double y = (a2*c1-a1*c2)/(a1*b2-a2*b1);

    return y;
}
```

Python Code - Using Shared Object

```
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import ctypes

c_lib=ctypes.CDLL('./code.so')

c_lib.POI_x.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_double, ctypes.c_double,ctypes.c_double]

c_lib.POI_y.argtypes = [ctypes.c_double, ctypes.c_double, ctypes.
    c_double, ctypes.c_double, ctypes.c_double,ctypes.c_double]

c_lib.POI_x.restype = ctypes.c_double
c_lib.POI_y.restype = ctypes.c_double
```

Python Code - Using Shared Object

```
a1, b1, c1 = 4.0, -1.0, 5.0
a2, b2, c2 = 1.0, 1.0, -5.0
a3, b3, c3 = 1.0, -4.0, 5.0

x1 = c_lib.POI_x(a1, b1, c1, a2, b2, c2)
y1 = c_lib.POI_y(a1, b1, c1, a2, b2, c2)

x2 = c_lib.POI_x(a2, b2, c2, a3, b3, c3)
y2 = c_lib.POI_y(a2, b2, c2, a3, b3, c3)

x3 = c_lib.POI_x(a3, b3, c3, a1, b1, c1)
y3 = c_lib.POI_y(a3, b3, c3, a1, b1, c1)
```

Python Code - Using Shared Object

```
A = np.array([x1, y1])
B = np.array([x2, y2])
C = np.array([x3, y3])

plt.plot([A[0],B[0]],[A[1],B[1]], label = "$4x-y+5=0$")
plt.plot([C[0],B[0]],[C[1],B[1]], label = "$x+y-5=0$")
plt.plot([A[0],C[0]],[A[1],C[1]], label = "$x-4y+5=0$")

A = A.reshape(-1,1)
B = B.reshape(-1,1)
C = C.reshape(-1,1)
```

Python Code - Using Shared Object

```
tri_coords = np.block([[A,B,C]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['A','B','C']

for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i
        ]:.0f})',
                 (tri_coords[0,i], tri_coords[1,i]),
                 textcoords="offset points",
                 xytext=(20,5),
                 ha='center')
```

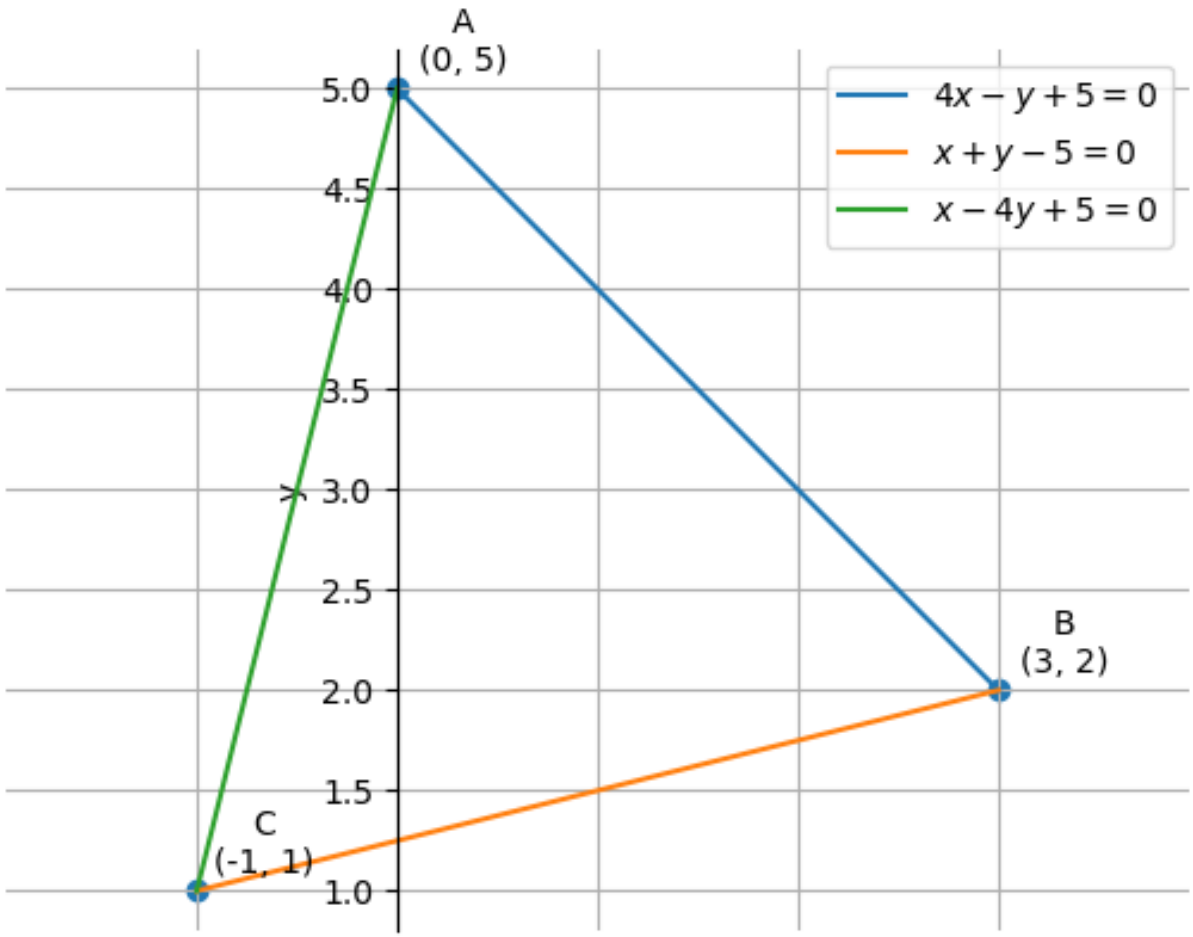
Python Code - Using Shared Object

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')

plt.savefig("../Figs/plot(py+C).png")

plt.show()
```

Plot-Using Both C and Python



Python Code

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

a1, b1, c1 = 4.0, -1.0, 5.0
a2, b2, c2 = 1.0, 1.0, -5.0
a3, b3, c3 = 1.0, -4.0, 5.0

x1 = (b1*c2-b2*c1)/(a1*b2-a2*b1)
y1 = (a2*c1-a1*c2)/(a1*b2-a2*b1)

x2 = (b2*c3-b3*c2)/(a2*b3-a3*b2)
y2 = (a3*c2-a2*c3)/(a2*b3-a3*b2)

x3 = (b1*c3-b3*c1)/(a1*b3-a3*b1)
y3 = (a3*c1-a1*c3)/(a1*b3-a3*b1)
```

Python Code

```
A = np.array([x1, y1])
B = np.array([x2, y2])
C = np.array([x3, y3])

plt.plot([A[0],B[0]],[A[1],B[1]], 'r', label = "$4x-y+5=0$")
plt.plot([C[0],B[0]],[C[1],B[1]], 'g', label = "$x+y-5=0$")
plt.plot([A[0],C[0]],[A[1],C[1]], 'b', label = "$x-4y+5=0$")

A = A.reshape(-1,1)
B = B.reshape(-1,1)
C = C.reshape(-1,1)
```

Python Code

```
tri_coords = np.block([[A,B,C]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['A','B','C']
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i
        ]:.0f})',
                 (tri_coords[0,i], tri_coords[1,i]),
                 textcoords="offset points",
                 xytext=(20,5),
                 ha='center')
```

Python Code

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['left'].set_position('zero')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')

plt.savefig("../Figs/plot(py).png")
plt.show()
```

Plot-Using Python only