5.8.34

M Chanakya Srinivas- EE25BTECH11036

Problem Statement

Draw the graphs of the equations

$$\mathbf{L_1} : x - y + 1 = 0, \quad \mathbf{L_2} : 3x + 2y - 12 = 0 \tag{1}$$

Determine the coordinates of the vertices of the triangle formed by these lines and the axes, using matrices and vectors only.

Step 1: Represent Lines in Matrix Form

A line $\mathbf{L}$ can be written as

$$\mathbf{n}^\top \mathbf{X} = c \tag{2}$$

where $\mathbf{n}$ is the normal vector and $\mathbf{X} = (\mathbf{x}, \mathbf{y})^\top$.

$$\mathbf{L_1} : \mathbf{n_1}^\top \mathbf{X} = -1, \quad \mathbf{n_1} = (\mathbf{1}, \mathbf{-1})^\top \tag{3}$$

$$\mathbf{L_2} : \mathbf{n_2}^\top \mathbf{X} = 12, \quad \mathbf{n_2} = (\mathbf{3}, \mathbf{2})^\top \tag{4}$$

Step 2: Intersections with Axes

Intersection of $\mathbf{L_1}$ with x-axis: $\mathbf{Y} = (\mathbf{x}, \mathbf{0})^\top$

$$\mathbf{n_1}^\top \mathbf{Y} = -1 \quad \Rightarrow \quad x = -1 \quad \Rightarrow \mathbf{A} = (-\mathbf{1}, \mathbf{0}) \tag{5}$$

Intersection of $\mathbf{L_1}$ with y-axis: $\mathbf{Y} = (\mathbf{0}, \mathbf{y})^\top$

$$\mathbf{n_1}^\top \mathbf{Y} = -1 \quad \Rightarrow \quad y = 1 \quad \Rightarrow \mathbf{B} = (\mathbf{0}, \mathbf{1}) \tag{6}$$

Intersection of $\mathbf{L_2}$ with x-axis: $\mathbf{Y} = (\mathbf{x}, \mathbf{0})^\top$

$$\mathbf{n_2}^\top \mathbf{Y} = 12 \quad \Rightarrow \quad x = 4 \quad \Rightarrow \mathbf{C} = (\mathbf{4}, \mathbf{0}) \tag{7}$$

Intersection of $\mathbf{L_2}$ with y-axis: $\mathbf{Y} = (\mathbf{0}, \mathbf{y})^\top$

$$\mathbf{n_2}^\top \mathbf{Y} = 12 \quad \Rightarrow \quad y = 6 \quad \Rightarrow \mathbf{D} = (\mathbf{0}, \mathbf{6}) \tag{8}$$

Step 3: Intersection of Lines Using Matrices

Intersection point $\mathbf{P}$ satisfies

$$\mathbf{NP} = \mathbf{C_0}, \quad \mathbf{N} = \begin{pmatrix} 1 & -1 \\ 3 & 2 \end{pmatrix}, \quad \mathbf{C_0} = \begin{pmatrix} -1 \\ 12 \end{pmatrix} \tag{9}$$

Solve using the inverse of $\mathbf{N}$:

$$\mathbf{P} = \mathbf{N}^{-1}\mathbf{C_0} = \frac{1}{5}\begin{pmatrix} 2 & 1 \\ -3 & 1 \end{pmatrix}\begin{pmatrix} -1 \\ 12 \end{pmatrix} = (\mathbf{2}, \mathbf{3}) \tag{10}$$

Step 4: Vertices of the Triangle

The triangle formed by the lines and axes has vertices

$$\mathbf{B} = (\mathbf{0}, \mathbf{1}), \quad \mathbf{C} = (\mathbf{4}, \mathbf{0}), \quad \mathbf{P} = (\mathbf{2}, \mathbf{3}) \tag{11}$$

All intersections and the triangle vertices are determined strictly using matrices and vectors.

Conclusion

The triangular region is bounded by:

$$\boxed{\mathbf{B} = (\mathbf{0}, \mathbf{1}),\ \mathbf{C} = (\mathbf{4}, \mathbf{0}),\ \mathbf{P} = (\mathbf{2}, \mathbf{3})} \tag{12}$$

This solution uses only **matrix-vector methods**, fully in **GVV-Sharma style**.

C code

```c
#include <stdio.h>

typedef struct {
    double x;
    double y;
} Point;

void get_triangle_vertices(Point* A, Point* B, Point* C) {
    A->x = -1; A->y = 0;
    B->x = 4; B->y = 0;
    C->x = 2; C->y = 3;
}
```

Python code through shared output

```
import ctypes
from ctypes import Structure, c_double
import matplotlib.pyplot as plt
import numpy as np

# Define the Point structure as in C
class Point(Structure):
    _fields_ = [(x, c_double), (y, c_double)]

# Load the shared library
lib = ctypes.CDLL('./libtriangle.so')

# Define the argument types for the C function
lib.get_triangle_vertices.argtypes = [ctypes.POINTER(Point),
    ctypes.POINTER(Point), ctypes.POINTER(Point)]
```

Python code through shared output

```
# Create Point instances to hold the vertices
A = Point()
B = Point()
C = Point()

# Call the C function to fill the points
lib.get_triangle_vertices(ctypes.byref(A), ctypes.byref(B),
    ctypes.byref(C))

# Extract points as numpy arrays for plotting
A_np = np.array([A.x, A.y])
B_np = np.array([B.x, B.y])
C_np = np.array([C.x, C.y])

# Plot setup
fig, ax = plt.subplots()
```

Python code through shared output

```
ax.set_aspect('equal')
ax.set_xlim(min(A.x, B.x, C.x) - 2, max(A.x, B.x, C.x) + 2)
ax.set_ylim(min(A.y, B.y, C.y) - 2, max(A.y, B.y, C.y) + 2)

# Draw the triangle
triangle = plt.Polygon([A_np, B_np, C_np], closed=True, color='
    skyblue', edgecolor='black', alpha=0.6, label='Triangle')
ax.add_patch(triangle)

# Draw lines
x = np.linspace(min(A.x, B.x, C.x) - 2, max(A.x, B.x, C.x) + 2,
    400)
```

Python code through shared output

```
# Line 1: x - y + 1 = 0  y = x + 1
y1 = x + 1
ax.plot(x, y1, 'r--', label='Line 1: x - y + 1 = 0')

# Line 2: 3x + 2y - 12 = 0  y = (12 - 3x)/2
y2 = (12 - 3*x)/2
ax.plot(x, y2, 'g--', label='Line 2: 3x + 2y - 12 = 0')

# X and Y axes
ax.axhline(0, color='black', linewidth=1)
ax.axvline(0, color='black', linewidth=1)
```

Python code through shared output

```
# Mark points A, B, C
points = {f'A ({A.x}, {A.y})': A_np, f'B ({B.x}, {B.y})': B_np, f
    'C ({C.x}, {C.y})': C_np}
for label, point in points.items():
    ax.plot(*point, 'ko')
    ax.text(point[0] + 0.1, point[1] + 0.1, label, fontsize=10)

# Grid and legend
ax.grid(True, linestyle='--', alpha=0.5)
ax.legend()
plt.title(Triangle formed by lines and x-axis (from C shared
    library))
plt.xlabel(x)
plt.ylabel(y)

plt.show()
```

only Python code

```
import sys
sys.path.insert(0, '/sdcard/github/matgeo/codes/CoordGeo') # your
     path

import numpy as np
import matplotlib.pyplot as plt

# local imports
from line.funcs import line_gen # your line generation function

# Given lines:
# L1: x - y + 1 = 0 => y = x + 1
# L2: 3x + 2y - 12 = 0 => y = (12 - 3x)/2
```

only Python code

```
# Find vertices of triangle formed by lines and axes:

# Intersection of L1 with x-axis (y=0): x - 0 + 1=0 => x=-1
A = np.array([-1, 0]).reshape(-1, 1)

# Intersection of L2 with x-axis (y=0): 3x - 12=0 => x=4
B = np.array([4, 0]).reshape(-1, 1)

# Intersection of L1 and L2: solve
# y = x + 1 and y = (12 - 3x)/2
# => x + 1 = (12 - 3x)/2 => 2x + 2 = 12 - 3x => 5x=10 => x=2, y=3
C = np.array([2, 3]).reshape(-1, 1)
```

only Python code

```
# Generate triangle edges using your line_gen
x_AB = line_gen(A, B)
x_BC = line_gen(B, C)
x_CA = line_gen(C, A)

# Plot original lines for reference
x_vals = np.linspace(-2, 5, 200)
y_L1 = x_vals + 1
y_L2 = (12 - 3 * x_vals) / 2

plt.plot(x_vals, y_L1, 'r--', label='$x - y + 1 = 0$')
plt.plot(x_vals, y_L2, 'g--', label='$3x + 2y - 12 = 0$')
```

only Python code

```
# Plot axes
plt.axhline(0, color='black') # x-axis
plt.axvline(0, color='black') # y-axis

# Plot triangle edges
plt.plot(x_AB[0, :], x_AB[1, :], 'b-', linewidth=2, label='
    Triangle edges')
plt.plot(x_BC[0, :], x_BC[1, :], 'b-', linewidth=2)
plt.plot(x_CA[0, :], x_CA[1, :], 'b-', linewidth=2)

# Fill triangle
plt.fill([A[0,0], B[0,0], C[0,0]], [A[1,0], B[1,0], C[1,0]], '
    skyblue', alpha=0.5)
```

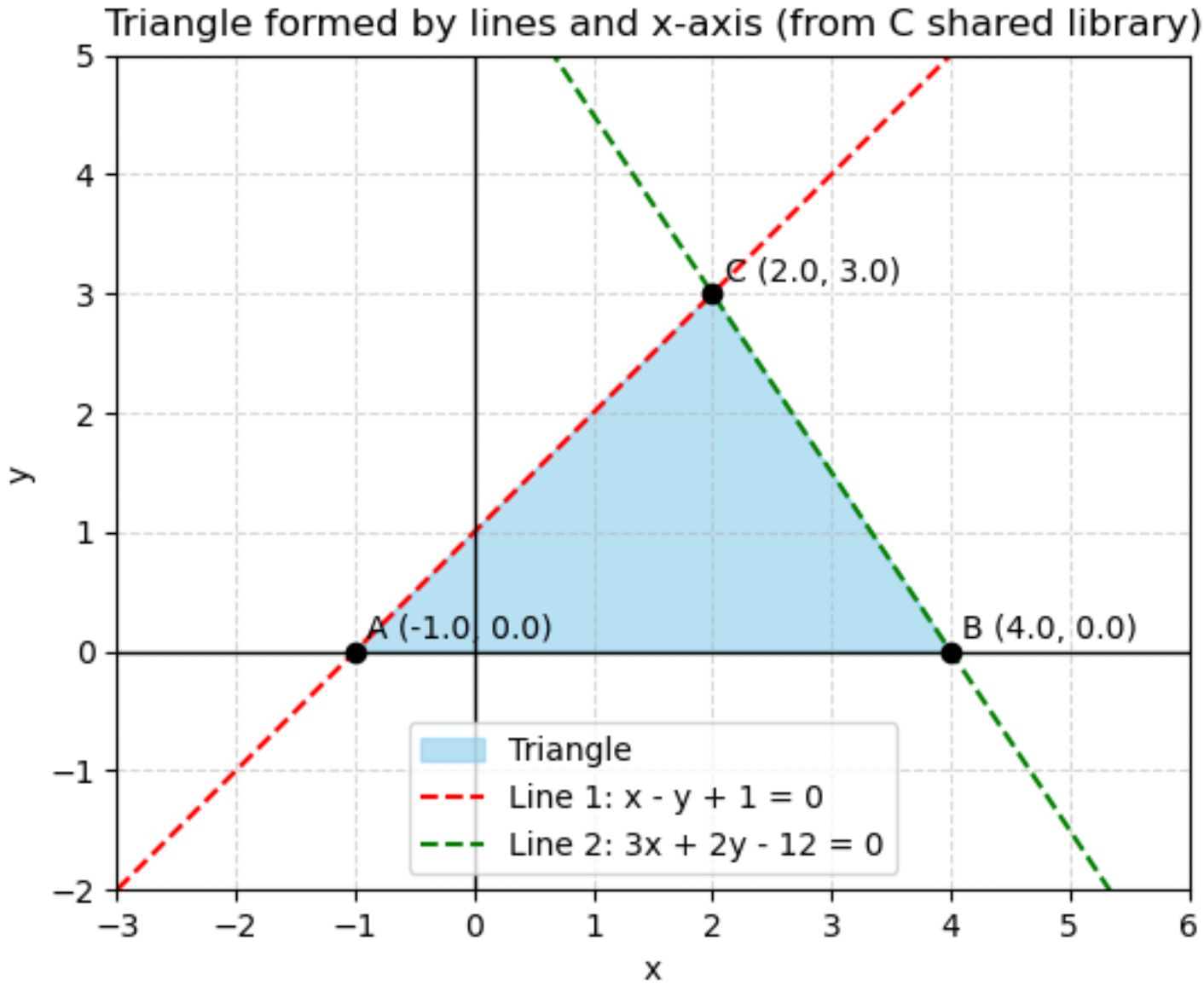
only Python code

```
# Label points
points = np.hstack((A, B, C))
labels = ['A', 'B', 'C']
for i, txt in enumerate(labels):
    plt.scatter(points[0, i], points[1, i], color='black')
    plt.annotate(f'{txt}\n({points[0, i]:.2f}, {points[1, i]:.2f
        })',
                 (points[0, i], points[1, i]),
                 textcoords=offset points,
                 xytext=(15, 5),
                 ha='center')
```

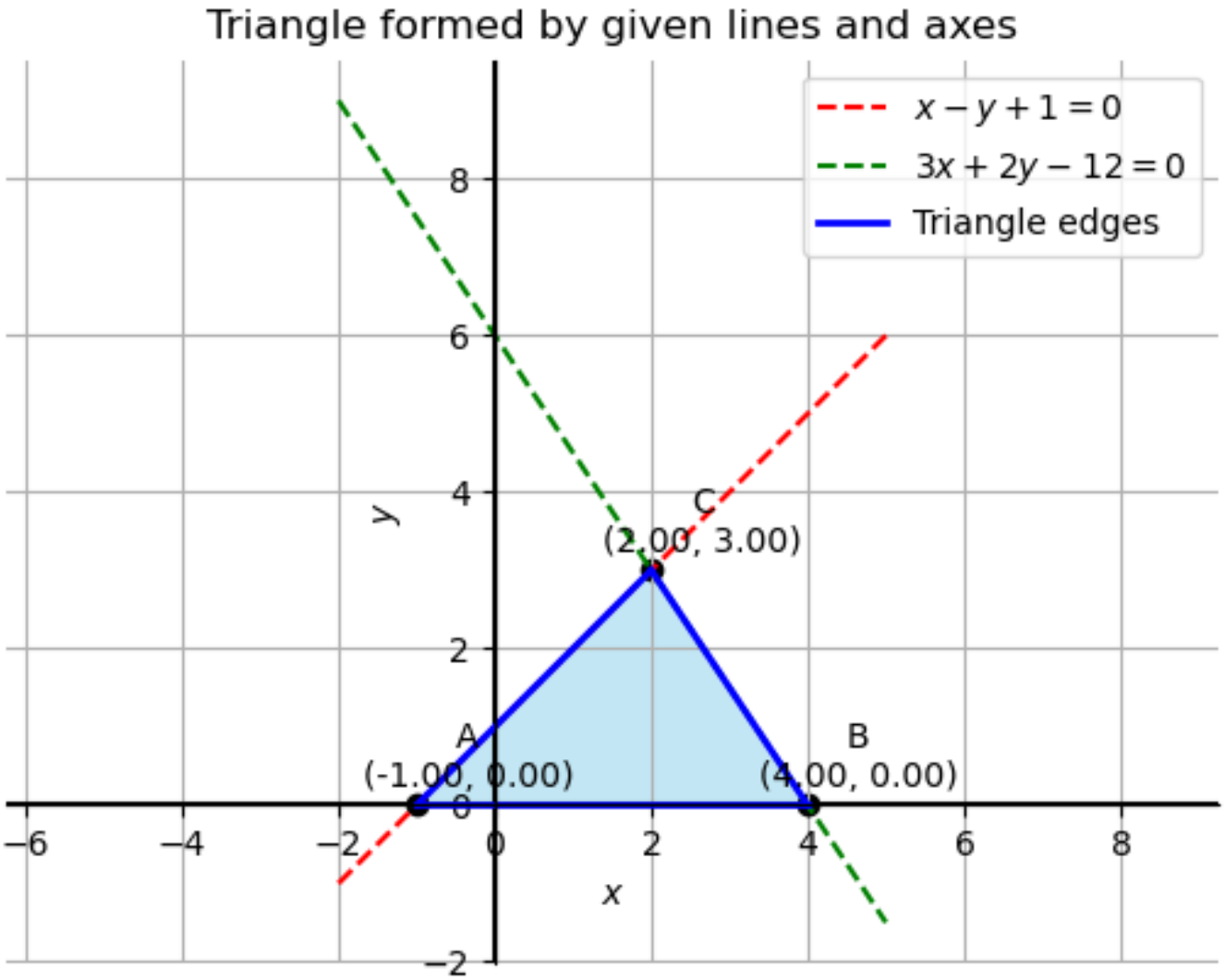
only Python code

```
# Axis formatting same as your code
ax = plt.gca()
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')

plt.grid(True)
plt.axis('equal')

plt.xlabel('$x$')
plt.ylabel('$y$')
plt.title('Triangle formed by given lines and axes')
plt.legend(loc='best')

plt.show()
```

PLOTS


Triangle formed by lines and x-axis (from C shared library)
C (2.0, 3.0)
A (-1.0, 0.0)
B (4.0, 0.0)
Triangle
Line 1: x - y + 1 = 0
Line 2: 3x + 2y - 12 = 0
x
y

PLOTS


Triangle formed by given lines and axes
$x - y + 1 = 0$
$3x + 2y - 12 = 0$
Triangle edges
C
(2.00, 3.00)
A
(-1.00, 0.00)
B
(4.00, 0.00)
x
y