

2.6.18

Dhanush Kumar A - AI25BTECH11010

September 9, 2025

Question

Find the area of the region bounded by the triangle whose vertices are $(-1, 0)$, $(1, 3)$ and $(3, 2)$.

Variables Used

Variable	Formula
A	$A = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$
B	$B = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$
C	$C = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$

Table: Variables Used

$$\text{Area of triangle ABC} = \frac{1}{2} |(\mathbf{A} - \mathbf{B}) \times (\mathbf{A} - \mathbf{C})| \quad (1)$$

$$\mathbf{A} - \mathbf{B} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \end{pmatrix} \quad (2)$$

$$\mathbf{A} - \mathbf{C} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} -4 \\ -2 \end{pmatrix} \quad (3)$$

$$(\mathbf{A} - \mathbf{B}) \times (\mathbf{A} - \mathbf{C}) = (-2)(-2) - (-3)(-4) = 4 - 12 = -8 \quad (4)$$

$$\text{Area} = \frac{1}{2} |-8| = 4 \quad (5)$$

Thus, the area of the triangle is 4.

Python code - Calculating the area of triangle

```
# Triangle Plotting Script
# Author: Dhanush (based on GVV Sharma)
# September 13, 2025
# Draw a triangle, calculate area, and save figure

import sys
import os
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

# Add parent folder of 'triangle' and 'line' to Python path
sys.path.insert(0, '/home/dhanush-kumar-a/code/CoordGeo')

# Local imports
from triangle.funcs import tri_sides, tri_mid_pt
from line.funcs import dir_vec, norm_vec, line_gen
```

Python code - Calculating the area of triangle

```
# -----  
# Triangle vertices (column vectors)  
A = np.array([-1, 0]).reshape(-1,1)  
B = np.array([1, 3]).reshape(-1,1)  
C = np.array([3, 2]).reshape(-1,1)  
m1 = dir_vec(A,B)  
m2 = dir_vec(B,C)  
m3 = dir_vec(C,A)  
  
# Area using cross product  
arvec = np.cross(m1[:,0], m3[:,0])  
area = 0.5 * LA.norm(arvec)  
print(f"Area of the triangle: {area:.3f}")
```

Python code - Plotting the triangle

```
# Generate points for triangle sides
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)

# Plot triangle sides
plt.plot(x_AB[0:], x_AB[1:], 'b', label='AB')
plt.plot(x_BC[0:], x_BC[1:], 'g', label='BC')
plt.plot(x_CA[0:], x_CA[1:], 'r', label='CA')

# Plot vertices
tri_coords = np.block([[A,B,C]])
plt.scatter(tri_coords[0:], tri_coords[1:], color='red')
```

Python code - Plotting the triangle

```
# Annotate vertices
vert_labels = ['A','B','C']
for i, txt in enumerate(vert_labels):
    plt.annotate(txt,
                  (tri_coords[0,i], tri_coords[1,i]),
                  textcoords="offset points",
                  xytext=(0,10),
                  ha='center',
                  fontsize=12, color='blue')
```

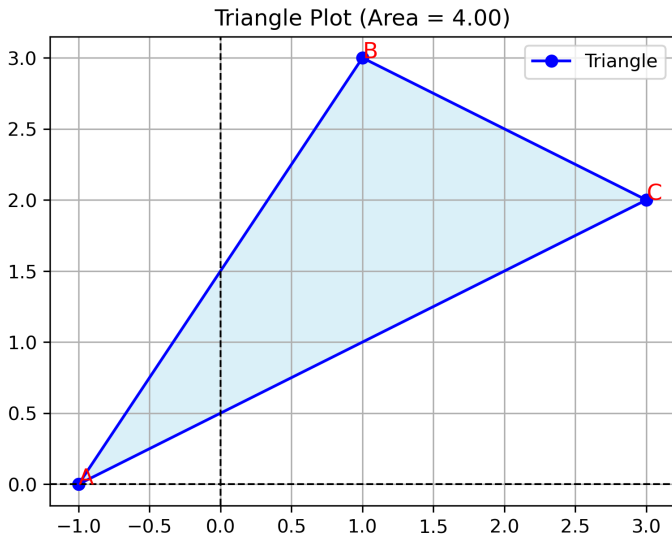

Python code - Plotting the triangle

```
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Triangle Plot')
plt.grid(True)
plt.axis('equal')
plt.legend()

# Save the figure
plt.savefig('../figs/triangle_plot.png', dpi=300)
print("Figure saved as figs/triangle_plot.png")

plt.show()
```

Plot-Using Python



C code - To calculate the area of triangle and Save points

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "/home/dhanush-kumar-a/ee1030-2025/ai25btech11010/matgeo
/1.11.12/codes/libs/matfun.h"

int main() {
    // Step 1: Create triangle vertices as 2x1 matrices
    double **A = createMat(2,1);
    double **B = createMat(2,1);
    double **C = createMat(2,1);

    A[0][0] = -1; A[1][0] = 0;
    B[0][0] = 1; B[1][0] = 3;
    C[0][0] = 3; C[1][0] = 2;
```

C code - To calculate the area of triangle and Save points

```
// Step 2: Compute vectors AB and AC using matrix subtraction
double **AB = Matsub(B, A, 2, 1);
double **AC = Matsub(C, A, 2, 1);

// Step 3: Create rotated row vector [AB_y, -AB_x] for cross
// product
double **rotAB = createMat(1,2);
rotAB[0][0] = AB[1][0]; // AB_y
rotAB[0][1] = -AB[0][0]; // -AB_x

// Step 4: Area = 0.5 * |rotAB * AC|
double **prod = Matmul(rotAB, AC, 1, 2, 1); // 1x1 matrix
double area = 0.5 * fabs(prod[0][0]);
```

C code - To calculate the area of triangle and Save points

```
// Step 5: Save results to files
```

```
FILE *fp_points = fopen("points.dat", "w");  
if(fp_points != NULL){  
    fprintf(fp_points, "Vertex\tX\tY\n");  
    fprintf(fp_points, "A\t%.2f\t%.2f\n", A[0][0], A[1][0]);  
    fprintf(fp_points, "B\t%.2f\t%.2f\n", B[0][0], B[1][0]);  
    fprintf(fp_points, "C\t%.2f\t%.2f\n", C[0][0], C[1][0]);  
    fclose(fp_points);  
}  
  
FILE *fp_area = fopen("area.dat", "w");  
if(fp_area != NULL){  
    fprintf(fp_area, "%.2f\n", area);  
    fclose(fp_area);  
}
```

C code - To calculate the area of triangle and Save points

```
// Step 6: Free memory
for(int i=0;i<2;i++){
    free(A[i]); free(B[i]); free(C[i]);
    free(AB[i]); free(AC[i]);
}
free(A); free(B); free(C);
free(AB); free(AC);
free(rotAB[0]); free(rotAB);
free(prod[0]); free(prod);

return 0;
}
```

Python code -Ploting the points using c function

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

# Step 1: Compile the C program
os.system("gcc c.c -o triangle -lm")

# Step 2: Run the compiled C program
os.system("./triangle")

# Step 3: Load points data
points_data = np.genfromtxt("points.dat", skip_header=1, dtype=
    None, encoding="utf-8")
labels = [row[0] for row in points_data]
x_vals = np.array([float(row[1]) for row in points_data])
y_vals = np.array([float(row[2]) for row in points_data])
```

Python code -Ploting the points using c function

```
# Step 4: Load area
with open("area.dat") as f:
    area = float(f.read().strip())

# Step 5: Prepare triangle coordinates
triangle_coords = [(x_vals[i], y_vals[i]) for i in range(3)]
triangle_coords.append(triangle_coords[0]) # close the triangle
tx, ty = zip(*triangle_coords)

# Step 6: Plot the triangle
fig, ax = plt.subplots(figsize=(6,6))
ax.plot(tx, ty, 'b-o', label='Triangle')
ax.fill(tx, ty, 'skyblue', alpha=0.3)
```


Python code -Ploting the points using c function

```
# Label points
for i in range(3):
    ax.text(x_vals[i], y_vals[i], f"{labels[i]}", fontsize=12,
            color='red')

# Axes formatting
ax.axhline(0, color="black", linewidth=1.0, linestyle="--")
ax.axvline(0, color="black", linewidth=1.0, linestyle="--")
ax.set_aspect("equal")
ax.grid(True)
ax.set_title(f"Triangle Plot (Area = {area:.2f})")
plt.legend()
```

Python code -Ploting the points using c function

```
# Step 7: Save and show plot
os.makedirs("../figs", exist_ok=True)
plt.savefig("../figs/triangle_plot.png", dpi=300, bbox_inches="
            tight")
plt.show()
```

Plot-Using Python and C

