2.6.25

Varun-ai25btech11016

September 1, 2025

Question

Find the area of a triangle formed by the points $A(5, 2)$, $B(4, 7)$ and $C(7, -4)$.

Theoretical Solution

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} -1 \\ 5 \end{pmatrix} \tag{1}$$

$$\mathbf{C} - \mathbf{A} = \begin{pmatrix} 2 \\ -6 \end{pmatrix} \tag{2}$$

$$\begin{aligned} \text{Area of the triangle ABC} &= \tfrac{1}{2}\left|\left|(\mathbf{B} - \mathbf{A}) \times (\mathbf{C} - \mathbf{A})\right|\right| \\ &= \tfrac{1}{2}\left|\left| \begin{pmatrix} -1 \\ 5 \end{pmatrix} \times \begin{pmatrix} 2 \\ -6 \end{pmatrix} \right|\right| \\ &= 2 \end{aligned} \tag{3}$$

Conclusion

Therefore,

The area of triangle ABC is 2

C Code

```c
#include <math.h>

// Function to compute triangle area using cross product
double triangle_area(double x1, double y1, double x2, double y2,
    double x3, double y3) {
    // Vectors B-A and C-A
    double ux = x2 - x1;
    double uy = y2 - y1;
    double vx = x3 - x1;
    double vy = y3 - y1;

    // Cross product (2D)
    double cross = ux * vy - uy * vx;

    // Area is half magnitude of cross product
    return 0.5 * fabs(cross);}
```

C plus Python code

```
import ctypes
import matplotlib.pyplot as plt

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

# Define the argument and return types
lib.triangle_area.argtypes = [ctypes.c_double, ctypes.c_double,
                              ctypes.c_double, ctypes.c_double,
                              ctypes.c_double, ctypes.c_double]
lib.triangle_area.restype = ctypes.c_double

# Triangle vertices
A = (5, 2)
B = (4, 7)
C = (7, -4)
```

C plus Python code

```
# Call the C function
area = lib.triangle_area(A[0], A[1], B[0], B[1], C[0], C[1])
print(Area of triangle ABC =, area)

# ---- Plotting ----
x_vals = [A[0], B[0], C[0], A[0]]
y_vals = [A[1], B[1], C[1], A[1]]

plt.plot(x_vals, y_vals, 'b-', linewidth=2, label=Triangle ABC)
plt.scatter([A[0], B[0], C[0]], [A[1], B[1], C[1]], color='red')

# Annotate points
plt.text(A[0], A[1],  A+str(A))
plt.text(B[0], B[1],  B+str(B))
plt.text(C[0], C[1],  C+str(C))
```

C plus Python code

```
plt.xlabel(X-axis)
plt.ylabel(Y-axis)
plt.title(fTriangle ABC (Area = {area}))
plt.grid(True)
plt.axis(equal)
plt.legend()
plt.savefig(/sdcard/Matrix/ee1030-2025/ai25btech11016/Matgeo
    /2.6.25/figs/2.6.25.png)
plt.show()
```

Python code

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

# Given points
A = np.array([5, 2])
B = np.array([4, 7])
C = np.array([7, -4])

# Compute area using determinant formula
area = 0.5 * abs(A[0]*(B[1]-C[1]) + B[0]*(C[1]-A[1]) + C[0]*(A
    [1]-B[1]))
print(Area of triangle:, area)
```

Python code

```
# Plot triangle
plt.figure(figsize=(6,6))
plt.plot([A[0], B[0], C[0], A[0]], [A[1], B[1], C[1], A[1]], 'b-'
    , linewidth=2)
plt.fill([A[0], B[0], C[0]], [A[1], B[1], C[1]], color='skyblue',
     alpha=0.4)

# Mark points
plt.scatter(*A, color='r')
plt.text(A[0]+0.2, A[1], A(5,2), fontsize=10)
plt.scatter(*B, color='r')
plt.text(B[0]+0.2, B[1], B(4,7), fontsize=10)
plt.scatter(*C, color='r')
plt.text(C[0]+0.2, C[1], C(7,-4), fontsize=10)
```

Python code

```
# Axis setup
plt.axhline(0, color='gray', linewidth=0.5)
plt.axvline(0, color='gray', linewidth=0.5)
plt.gca().set_aspect(equal)
plt.title(fTriangle ABC, Area = {area})
plt.grid(True)
plt.savefig(/sdcard/Matrix/ee1030-2025/ai25btech11016/Matgeo
    /2.6.25/figs/2.6.25.png)
plt.show()
```

Plot