3.2.1

Namaswi-EE25BTECH11060

september 10,2025

Question

Draw a triangle ABC in which AB=4cm,BC=6cm and AC=9cm.

solution

According to given data lets assume,

$$A = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad B = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \qquad C = \begin{pmatrix} x \\ y \end{pmatrix}$$

solution

$$||C - A|| = 9 \tag{1}$$

$$C^\top C = 81 \tag{2}$$

$$||C - B|| = 6 \tag{3}$$

$$(C - B)^\top (C - B) = 36 \tag{4}$$

$$C^\top C - 2B^\top C + B^\top B = 36 \tag{5}$$

$$as, B^\top B = 16 \; ; C^\top C = 81 \tag{6}$$

$$2B^\top C = 61 \tag{7}$$

$$\begin{pmatrix} 8 & 0 \end{pmatrix} C = 61 \tag{8}$$

$$Augmented\ Matrix \implies \left(\begin{array}{cc|c} 8 & 0 & 61 \end{array}\right) \tag{9}$$

$$\implies \left(\begin{array}{cc|c} 1 & 0 & 61/8 \end{array}\right) \tag{10}$$

$$x = 61/8 \tag{11}$$

Solution

$$as, C^\top C = 81 \tag{12}$$

$$x^2 + y^2 = 81 \tag{13}$$

$$y = \sqrt{\frac{1463}{64}} \tag{14}$$

$$C = \begin{pmatrix} 7.625 \\ \pm 4.781 \end{pmatrix} \tag{15}$$

Refer fig

C Code

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

int main() {
    // Given distances (according to your solution)
    double AC = 9.0, BC = 6.0;
    double AB = 4.0;

    // Fix A and B
    double Ax = 0.0, Ay = 0.0;
    double Bx = 4.0, By = 0.0;

    // Step 1: Find x using the relation (from your derivation)
    // 2 * B^T * C = AC^2 + B^T B - BC^2
    double x = (AC*AC + (Bx*Bx + By*By) - BC*BC) / (2 * Bx);
```

C Code

```
    // Step 2: Use AC^2 = x^2 + y^2 to solve for y
    double y_square = AC*AC - x*x;

    if (y_square < 0) {
        printf("No real solution (triangle inequality violated).\
            n");
        return 0;
    }

    double y1 = sqrt(y_square);
    double y2 = -sqrt(y_square);
```

C Code

```
  // Print results
    printf("Coordinates of A: (%.3f, %.3f)\n", Ax, Ay);
    printf("Coordinates of B: (%.3f, %.3f)\n", Bx, By);
    printf("Possible coordinates of C:\n");
    printf("C1 = (%.3f, %.3f)\n", x, y1);
    printf("C2 = (%.3f, %.3f)\n", x, y2);

    return 0;
}
```

Python Code

```
 import matplotlib.pyplot as plt

# Define vertices
A = (0, 0)
B = (4, 0)
C = (7.625, 4.781)

# Collect coordinates (close the triangle by repeating A at the
    end)
x_coords = [A[0], B[0], C[0], A[0]]
y_coords = [A[1], B[1], C[1], A[1]]

# Plot the triangle
plt.figure(figsize=(6,6))
plt.plot(x_coords, y_coords, 'b-', linewidth=2) # Triangle edges
plt.fill(x_coords, y_coords, 'skyblue', alpha=0.3) # Fill inside
```

Python Code

```
 # Plot vertices
plt.scatter(*A, color='red', s=60)
plt.scatter(*B, color='green', s=60)
plt.scatter(*C, color='purple', s=60)

# Label points
plt.text(A[0]-0.3, A[1]-0.3, 'A', fontsize=12, fontweight='bold')
plt.text(B[0]+0.2, B[1]-0.3, 'B', fontsize=12, fontweight='bold')
plt.text(C[0]+0.2, C[1]+0.2, 'C', fontsize=12, fontweight='bold')

# Formatting
plt.axhline(0, color='gray', linewidth=0.5)
plt.axvline(0, color='gray', linewidth=0.5)
plt.gca().set_aspect('equal', adjustable='box')
plt.title("Triangle ABC")
plt.grid(True)
plt.show()
```

C and Python Code

```
 import ctypes
import os

# Load shared object
lib = ctypes.CDLL(os.path.abspath("./triangle.so"))

# Define argument and return types
lib.find_triangle.argtypes = [ctypes.c_double, ctypes.c_double,
    ctypes.c_double,
                              ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double),
                              ctypes.POINTER(ctypes.c_double)]
```

C and Python Code

```
 # Function wrapper
def find_triangle(AB, AC, BC):
    x = ctypes.c_double()
    y1 = ctypes.c_double()
    y2 = ctypes.c_double()

    lib.find_triangle(AB, AC, BC,
                      ctypes.byref(x), ctypes.byref(y1), ctypes.
                          byref(y2))

    return x.value, y1.value, y2.value
```

C and Python Code

```
# Example usage
if __name__ == "__main__":
    AB, AC, BC = 4.0, 9.0, 6.0 # Example
    x, y1, y2 = find_triangle(AB, AC, BC)

    print(f"A = (0,0)")
    print(f"B = ({AB},0)")
    print(f"C1 = ({x:.3f},{y1:.3f})")
    print(f"C2 = ({x:.3f},{y2:.3f})")
```

Plot