MatGeo Assignment 1.2.13

AI25BTECH11007

September 19, 2025

Question

Construct a triangle ABC in which

$$BC = 5\,\text{cm}, \quad \angle B = 45^\circ, \quad \text{and} \quad AC + AB = 7.5\,\text{cm}.$$

Solution

Set up points and given data

Let

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \mathbf{C} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}.$$

$$\mathbf{AC} = \mathbf{c}, \qquad \mathbf{AC} = \mathbf{b}, \qquad \mathbf{b} + \mathbf{c} = 7.5.$$

Position vector of point A

Since $\angle B = 60^\circ$, the vector $\mathbf{BA}$ has length c and direction 60° above the x-axis. Thus

$$\mathbf{A} = \mathbf{c} \begin{pmatrix} \cos 60^\circ \\ \sin 60^\circ \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{c}}{2} \\ \frac{\mathbf{c}\sqrt{3}}{2} \end{pmatrix}.$$

Expression for AC

$$\mathbf{AC} = \mathbf{C} - \mathbf{A} = \begin{pmatrix} 5 - \frac{\mathbf{c}}{2} \\ -\frac{\mathbf{c}\sqrt{3}}{2} \end{pmatrix},$$

and

$$b^2 = \left(5 - \frac{c}{2}\right)^2 + \frac{3c^2}{4}.$$

Apply $b + c = 7.5$
Since $b = 7.5 - c$, we have

$$(7.5 - c)^2 = \left(5 - \frac{c}{2}\right)^2 + \frac{3c^2}{4}.$$

Expanding and simplifying gives,

$$56.25 - 15c + c^2 = 25 - 5c + c^2,$$

$$c = 3.125.$$

Hence

$$b = 7.5 - 3.125 = 4.375.$$

Coordinates of vertices

$$\mathbf{A} = \begin{pmatrix} \frac{3.125}{2} \\ \frac{3.125\sqrt{3}}{2} \end{pmatrix} = \begin{pmatrix} 1.5625 \\ 2.7050\ldots \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}.$$

Verification

$$\mathbf{BA} \cdot \mathbf{BC} = \tfrac{5c}{2}, \qquad |\mathbf{BA}| = c, \quad |\mathbf{BC}| = 5,$$

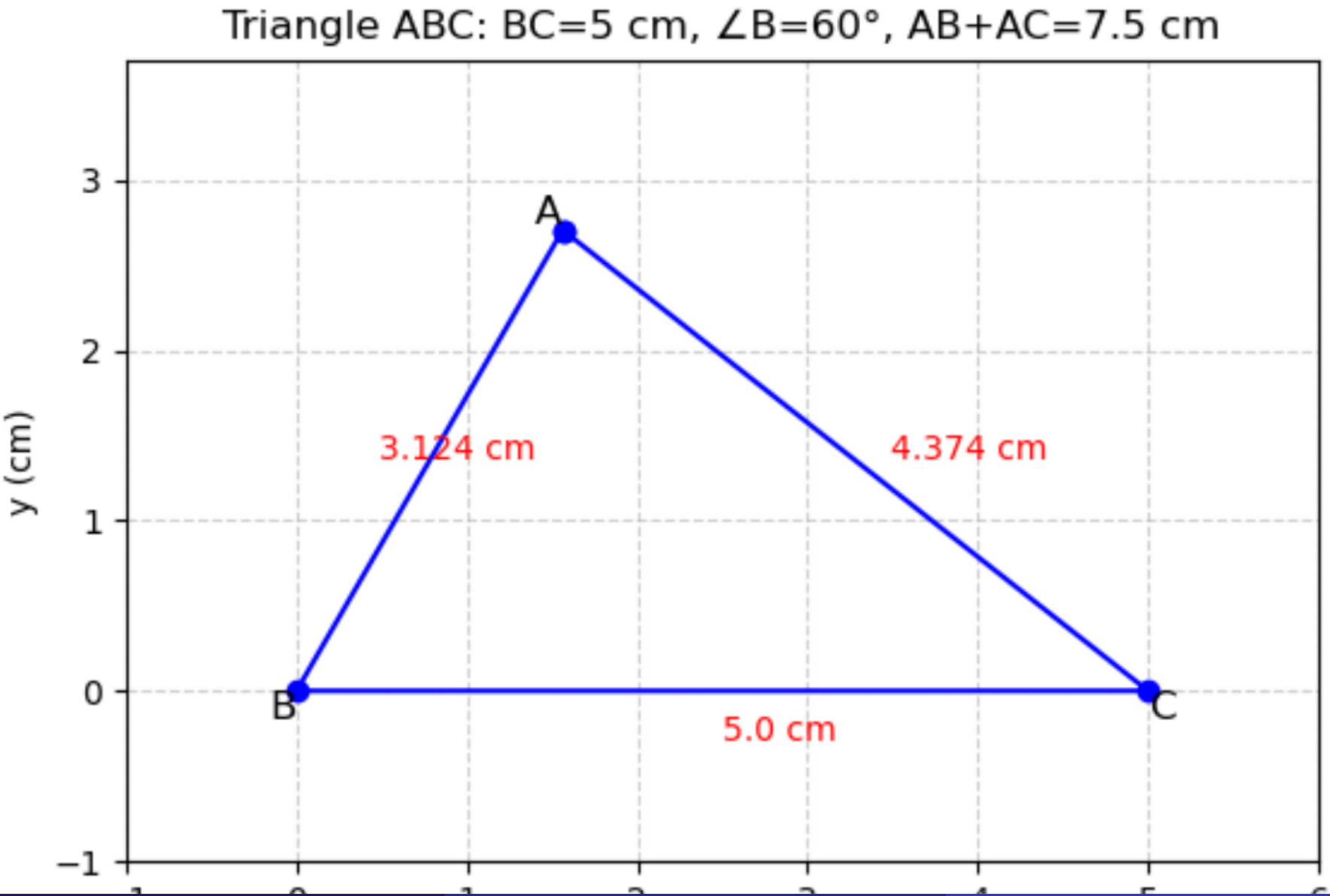
$$\cos\angle B = \frac{\frac{5c}{2}}{5c} = \tfrac{1}{2} = \cos 60^\circ.$$

Final Answer,

$$\boxed{\mathbf{A} = \begin{pmatrix} 1.5625 \\ 2.7050 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}}$$

with $AB = 3.125$ cm, $AC = 4.375$ cm, and $BC = 5$ cm.

Construction Plot



C Code: Dot Product and Magnitude

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

// Function to compute dot product of 2D vectors
double dotProduct(double A[], double B[]) {
    return A[0]*B[0] + A[1]*B[1];
}

// Function to compute magnitude of a 2D vector
double magnitude(double V[]) {
    return sqrt(V[0]*V[0] + V[1]*V[1]);
}
```

C Code: Angle Calculation and Main

```c
// Function to calculate angle (in degrees) between two vectors
double angleBetweenVectors(double A[], double B[]) {
    double dot = dotProduct(A, B);
    double magA = magnitude(A);
    double magB = magnitude(B);
    double cosTheta = dot / (magA * magB);

    // Clamp for numerical stability
    if (cosTheta > 1.0) cosTheta = 1.0;
    else if (cosTheta < -1.0) cosTheta = -1.0;

    return acos(cosTheta) * (180.0 / M_PI);
}
```

C Code

```c
int main() {
    // Coordinates of vertices
    double A[2] = {1.5625, 2.705};
    double B[2] = {0.0, 0.0};
    double C[2] = {5.0, 0.0};

    // Vectors BA and BC
    double BA[2] = {A[0] - B[0], A[1] - B[1]};
    double BC[2] = {C[0] - B[0], C[1] - B[1]};

    printf("Angle at B: %.2f degrees\n", angleBetweenVectors(BA,
        BC));
    return 0;
}
```

Python Code: Setup and Points

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

# Coordinates of vertices
A = np.array([1.5625, 2.705]) # Computed intersection
B = np.array([0, 0]) # Origin
C = np.array([5, 0]) # On x-axis
```

Python Code: Plot Triangle

```
fig, ax = plt.subplots()

# Plot the triangle edges
triangle_points = np.array([A, B, C, A])
ax.plot(triangle_points[:, 0], triangle_points[:, 1], 'b-',
    marker='o')

# Annotate vertices
ax.text(A[0], A[1], 'A', fontsize=12, ha='right', va='bottom')
ax.text(B[0], B[1], 'B', fontsize=12, ha='right', va='top')
ax.text(C[0], C[1], 'C', fontsize=12, ha='left', va='top')
```

Python Code: Final Touches and Save

```
# Formatting and labels
ax.set_aspect('equal', 'box')
ax.grid(True, linestyle='--', alpha=0.6)
ax.set_xlabel('x (cm)')
ax.set_ylabel('y (cm)')
ax.set_title('Triangle ABC: BC=5 cm, âĹăB=60Âř, AB+AC=7.5 cm')

# Axis limits with padding
padding = 1
min_x, max_x = min(A[0], B[0], C[0]) - padding, max(A[0], B[0], C
    [0]) + padding
min_y, max_y = min(A[1], B[1], C[1]) - padding, max(A[1], B[1], C
    [1]) + padding
ax.set_xlim(min_x, max_x)
ax.set_ylim(min_y, max_y)

# Save and display
plt.savefig('triangle_plot.png', dpi=300)
plt.show()
```