MatGeo Presentation - Problem 3.2.5

EE25BTECH11064 - Yojit Manral

Question

Draw a triangle ABC in which BC $= 6$ cm, CA $= 5$ cm and AB $= 4$ cm.

Solution

$\rightarrow$ Let

$$a = \|\mathbf{C} - \mathbf{B}\| = 6cm \tag{0.1}$$
$$b = \|\mathbf{A} - \mathbf{C}\| = 5cm \tag{0.2}$$
$$c = \|\mathbf{B} - \mathbf{A}\| = 4cm \tag{0.3}$$

$\rightarrow$ By using cosine law in $\triangle$ABC, we get

$$\cos B = \frac{a^2 + c^2 - b^2}{2ac} \tag{0.4}$$
$$\implies \cos B = \frac{6^2 + 4^2 - 5^2}{2 \times 6 \times 4} \tag{0.5}$$
$$\implies \cos B = \frac{9}{16} \tag{0.6}$$
$$\implies \angle B = \cos^{-1}\left(\frac{9}{16}\right) \approx 55^\circ \tag{0.7}$$

Solution

→ The coordinates of $\triangle ABC$ can then be expressed as

$$\mathbf{A} = c \begin{pmatrix} \cos B \\ \sin B \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

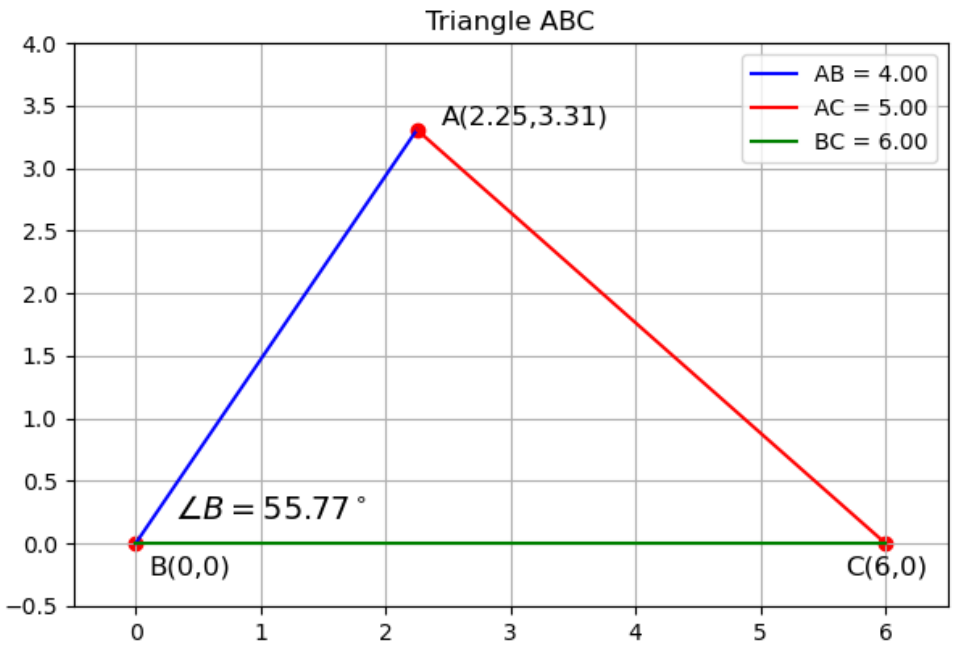

Figure: Plot of $\triangle ABC$

File: points.c

```
#include <stdio.h>

int main() {
  FILE *fp;

  // -------------------
  // Question 3.2.5
  // -------------------


  fp = fopen("points.dat", "w");
  fprintf(fp, "%d,%d,%d\n", 2.25, 3.31, 0); // A
  fprintf(fp, "%d,%d,%d\n", 0, 0, 0); // B
  fprintf(fp, "%d,%d,%d\n", 6, 0, 0); // C
  fclose(fp);
  return 0;
}
```

File: call_c.py

```
import subprocess

# Compile the C program
subprocess.run(["gcc", "points.c", "-o", "points"])

# Run the compiled C program
result = subprocess.run(["./points"], capture_output=True, text=True)

# Print the output from the C program
print(result.stdout)
```

File: plot.py

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

# Given coordinates
B = np.array([0, 0])
C = np.array([6, 0])
A_x = 4 * np.cos(np.deg2rad(55.77)) # A_x = 4 * cos(55.77 degrees)
A_y = 4 * np.sin(np.deg2rad(55.77)) # A_y = 4 * sin(55.77 degrees)
A = np.array([A_x, A_y])

# Calculate side lengths
AB = np.linalg.norm(A - B)
BC = np.linalg.norm(C - B)
AC = np.linalg.norm(A - C)

# Calculate angle B using the law of cosines
cos_angle_B = (AB**2 + BC**2 - AC**2) / (2 * AB * BC)
angle_B_rad = np.arccos(cos_angle_B) # In radians
angle_B_deg = np.rad2deg(angle_B_rad) # Convert to degrees

# Create the plot
plt.figure(figsize=(7, 5))

# Plot the triangle
plt.plot([B[0], A[0]], [B[1], A[1]], 'b-', label=f'AB = {AB:.2f}')
plt.plot([A[0], C[0]], [A[1], C[1]], 'r-', label=f'AC = {AC:.2f}')
plt.plot([C[0], B[0]], [C[1], B[1]], 'g-', label=f'BC = {BC:.2f}')
```

File: plot.py

```
# Mark the vertices
plt.scatter([B[0], A[0], C[0]], [B[1], A[1], C[1]], color='red')
plt.text(B[0]+0.1, B[1]-0.1, 'B(0,0)', fontsize=12, ha='left', va = 'top')
plt.text(A[0]+0.2, A[1], f'A({A_x:.2f},{A_y:.2f})', fontsize=12, ha='left', va = 'bottom')
plt.text(C[0], C[1]-0.1, 'C(6,0)', fontsize=12, ha='center', va='top')

# Label the angle at B and its value
plt.text(B[0] + 0.3, B[1] + 0.2, r'$\angle␣B␣=␣{:.2f}^\circ$'.format(angle_B_deg), fontsize=14, ha='left')

# Set plot properties
plt.gca().set_aspect('equal', adjustable='box')
plt.xlim(-0.5, 6.5)
plt.ylim(-0.5, 4)

# Title and grid
plt.title('Triangle␣ABC')
plt.grid(True)

# Show the plot
plt.legend()
plt.show()
```