2.4.26

Josyula G S Avaneesh - EE25BTECH11030

september 16,2025

Check whether (5,-2), (6,4) and (7,-2) are the vertices of an isosceles triangle.

For the points **ABC** to represent an isosceles triangle:
Property: The perpendicular bisector of a side passes through the opposite vertex.

Given details:

$$\mathbf{A} = \begin{pmatrix} 5 \\ -2 \end{pmatrix} \ \mathbf{B} = \begin{pmatrix} 6 \\ 4 \end{pmatrix} \ \mathbf{C} = \begin{pmatrix} 7 \\ -2 \end{pmatrix} \tag{1}$$

Midpoint of side **AC**:

$$\mathbf{M} = \frac{\mathbf{A} + \mathbf{C}}{2} = \frac{\begin{pmatrix} 5 \\ -2 \end{pmatrix} + \begin{pmatrix} 7 \\ -2 \end{pmatrix}}{2} = \begin{pmatrix} 6 \\ -2 \end{pmatrix} \tag{2}$$

Direction vector of side **AC**:

$$\mathbf{C} - \mathbf{A} = \begin{pmatrix} 7 \\ -2 \end{pmatrix} - \begin{pmatrix} 5 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \tag{3}$$

Vector from midpoint **M** to **B**:

$$\mathbf{B} - \mathbf{M} = \begin{pmatrix} 6 \\ 4 \end{pmatrix} - \begin{pmatrix} 6 \\ -2 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix} \tag{4}$$

$$(\mathbf{C} - \mathbf{A})^{\top}(\mathbf{B} - \mathbf{M}) = \begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 6 \end{pmatrix} = 2(0) + 0(6) = 0 \qquad (5)$$

B lies on the perpendicular bisector of side **AC**.

$$\therefore \mathbf{AB} = \mathbf{BC} \implies \triangle ABC \text{ is isosceles.}$$

C Code (1) - Function to store the points

```c
#include <stdio.h>

void isosceles_triangle(double *points) {
    double coords[6] = {5,-2, 6,4, 7,-2,};

    for (int i = 0; i < 6; i++) {
        points[i] = coords[i];
    }
}
```

Python Code - Using Shared Object

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

# Load the shared object
triangle_lib = ctypes.CDLL("./triangle.so")

# Define function return type
triangle_lib.isosceles_triangle.argtypes = [np.ctypeslib.
    ndpointer(dtype=np.double, ndim=1, flags="C")]
```

Python Code - Using Shared Object

```python
# Create numpy array to hold 6 values (x,y for 3 points)
points = np.zeros(6, dtype=np.double)

# Call C function to fill points
triangle_lib.isosceles_triangle(points)

# Reshape into (3,2)
points = points.reshape((3,2))

# Close the triangle (repeat first point)
points = np.vstack([points, points[0]])
```
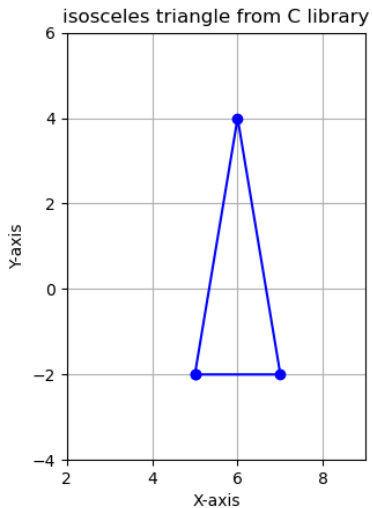
```python
# Plot isosceles triangle
plt.plot(points[:,0], points[:,1], "bo-")
plt.xlim([2,9])
plt.ylim([-4,6])
plt.title("isosceles triangle from C library")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.gca().set_aspect("equal")
plt.grid(True)
plt.savefig('figs/triangle.png')
plt.show()
```

Plot-Using Both C and Python

Python Code

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

# Define the vertices of the triangle
points = np.array([
    [5, -2], # A
    [6, 4], # B
    [7, -2], # C
])
```

Python Code

```python
# Close the triangle (repeat first point)
points = np.vstack([points, points[0]])

# Plot
plt.plot(points[:,0], points[:,1], "bo-", linewidth=2)
plt.xlim([2,9])
plt.ylim([-4,6])
plt.title("isosceles triangle")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.gca().set_aspect("equal")
plt.grid(True)
plt.savefig('figs/triangle2.png')
plt.show()
```

Plot-Using only Python