1.10.2

RAVULA SHASHANK REDDY - EE25BTECH11047

September 14, 2025

Question

Check whether the points $(7, 10)$, $(-2, 5)$, $(3, 4)$ form an isosceles right triangle.

Equation

The condition for two sides to be perpendicular :

$$\mathbf{n_1^T n_2} = 0$$

Theoretical Solution

Given:

$$\mathbf{A} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} \tag{1}$$

$$\mathbf{B} = \begin{pmatrix} -2 \\ 5 \end{pmatrix} \tag{2}$$

$$\mathbf{C} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \tag{3}$$

Side vectors:

$$\mathbf{A} - \mathbf{B} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} - \begin{pmatrix} -2 \\ 5 \end{pmatrix} = \begin{pmatrix} 9 \\ 5 \end{pmatrix} \tag{4}$$

$$\mathbf{A} - \mathbf{C} = \begin{pmatrix} 7 \\ 10 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \tag{5}$$

$$\mathbf{B} - \mathbf{C} = \begin{pmatrix} -2 \\ 5 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \end{pmatrix} \tag{6}$$

Theoretical Solution

Isosceles check:

1. Altitude from **A**

$$\mathbf{D} = \frac{\mathbf{B}+\mathbf{C}}{2} = \frac{1}{2}\begin{pmatrix} -2+3 \\ 5+4 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{9}{2} \end{pmatrix}. \tag{7}$$

$$\mathbf{A}-\mathbf{D} = \begin{pmatrix} \frac{13}{2} \\ \frac{11}{2} \end{pmatrix}, \qquad \mathbf{B}-\mathbf{C} = \begin{pmatrix} -5 \\ 1 \end{pmatrix}. \tag{8}$$

$$(\mathbf{A}-\mathbf{D})^T(\mathbf{B}-\mathbf{C}) = \begin{pmatrix} \frac{13}{2} & \frac{11}{2} \end{pmatrix}\begin{pmatrix} -5 \\ 1 \end{pmatrix} = -\frac{65}{2} + \frac{11}{2} \neq 0. \tag{9}$$

2. Altitude from **B**

$$\mathbf{E} = \frac{\mathbf{C}+\mathbf{A}}{2} = \begin{pmatrix} 5 \\ 7 \end{pmatrix}. \tag{10}$$

$$\mathbf{B}-\mathbf{E} = \begin{pmatrix} -7 \\ -2 \end{pmatrix}, \qquad \mathbf{C}-\mathbf{A} = \begin{pmatrix} -4 \\ -6 \end{pmatrix}. \tag{11}$$

$$(\mathbf{B}-\mathbf{E})^T(\mathbf{C}-\mathbf{A}) = \begin{pmatrix} -7 & -2 \end{pmatrix}\begin{pmatrix} -4 \\ -6 \end{pmatrix} = 28+12 = 40 \neq 0. \tag{12}$$

3.Altitude from $\mathbf{C}$

$$\mathbf{F} = \frac{\mathbf{A}+\mathbf{B}}{2} = \begin{pmatrix} \frac{5}{2} \\ \frac{15}{2} \end{pmatrix}. \tag{13}$$

$$\mathbf{C}-\mathbf{F} = \begin{pmatrix} \frac{1}{2} \\ -\frac{7}{2} \end{pmatrix}, \qquad \mathbf{A}-\mathbf{B} = \begin{pmatrix} 9 \\ 5 \end{pmatrix}. \tag{14}$$

$$(\mathbf{C}-\mathbf{F})^T(\mathbf{A}-\mathbf{B}) = \begin{pmatrix} \frac{1}{2} & -\frac{7}{2} \end{pmatrix}\begin{pmatrix} 9 \\ 5 \end{pmatrix} = \frac{9}{2} - \frac{35}{2} = -13 \neq 0. \tag{15}$$

Hence it is not isosceles triangle.

Theoretical Solution

Right angle Check:

For a right angle, the dot product of two sides must be zero.

$$(\mathbf{A}-\mathbf{B})^T(\mathbf{A}-\mathbf{C}) = (9)(4)+(5)(6) = 66 \neq 0 \tag{16}$$

$$(\mathbf{A}-\mathbf{B})^T(\mathbf{B}-\mathbf{C}) = (9)(-5)+(5)(1) = -40 \neq 0 \tag{17}$$

$$(\mathbf{A}-\mathbf{C})^T(\mathbf{B}-\mathbf{C}) = (4)(-5)+(6)(1) = -14 \neq 0 \tag{18}$$

Hence, the given points forms neither an isosceles nor a right-angled triangle.

C Code

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

int main() {
    // Coordinates of the points
    int Ax = 7, Ay = 10;
    int Bx = -2, By = 5;
    int Cx = 3, Cy = 4;

    // Squared lengths of sides
    int AB2 = (Ax - Bx) * (Ax - Bx) + (Ay - By) * (Ay - By);
    int AC2 = (Ax - Cx) * (Ax - Cx) + (Ay - Cy) * (Ay - Cy);
    int BC2 = (Bx - Cx) * (Bx - Cx) + (By - Cy) * (By - Cy);

    // Dot products (for right angle check)
    int dot_AB_AC = (Ax - Bx) * (Ax - Cx) + (Ay - By) * (Ay - Cy)
        ;
```

C Code

```
int dot_AB_BC = (Ax - Bx) * (Bx - Cx) + (Ay - By) * (By - Cy)
    ;
int dot_AC_BC = (Ax - Cx) * (Bx - Cx) + (Ay - Cy) * (By - Cy)
    ;

printf(Squared side lengths:\n);
printf(AB^2 = %d\n, AB2);
printf(AC^2 = %d\n, AC2);
printf(BC^2 = %d\n, BC2);

printf(\nDot products:\n);
printf((A-B)(A-C) = %d\n, dot_AB_AC);
printf((A-B)(B-C) = %d\n, dot_AB_BC);
printf((A-C)(B-C) = %d\n, dot_AC_BC);
```

C Code

```
    // Check isosceles right triangle
    if ((AB2 == AC2 && dot_AB_AC == 0) ||
        (AB2 == BC2 && dot_AB_BC == 0) ||
        (AC2 == BC2 && dot_AC_BC == 0)) {
        printf(\nThe points form an ISOSCELES RIGHT triangle.\n);
    } else {
        printf(\nThe points DO NOT form an isosceles right
            triangle.\n);
    }

    return 0;
}
```

Python Direct Code

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

# local imports
from libs.line.funcs import line_gen
from libs.triangle.funcs import *

# Coordinates of the points
A = np.array(([7, 10])).reshape(-1, 1)
B = np.array(([-2, 5])).reshape(-1, 1)
C = np.array(([3, 4])).reshape(-1, 1)

# Squared lengths of sides
AB2 = np.sum((A - B) ** 2)
AC2 = np.sum((A - C) ** 2)
BC2 = np.sum((B - C) ** 2)
```

Python Direct Code

```python
# Dot products
dot_AB_AC = np.dot((A - B).T, (A - C))[0, 0]
dot_AB_BC = np.dot((A - B).T, (B - C))[0, 0]
dot_AC_BC = np.dot((A - C).T, (B - C))[0, 0]

print(Squared side lengths:)
print(fAB^2 = {AB2})
print(fAC^2 = {AC2})
print(fBC^2 = {BC2})

print(\nDot products:)
print(f(A-B)(A-C) = {dot_AB_AC})
print(f(A-B)(B-C) = {dot_AB_BC})
print(f(A-C)(B-C) = {dot_AC_BC})
```

Python Direct Code

```
# Check isosceles right triangle
if ((AB2 == AC2 and dot_AB_AC == 0) or
    (AB2 == BC2 and dot_AB_BC == 0) or
    (AC2 == BC2 and dot_AC_BC == 0)):

    result = ISOSCELES RIGHT triangle
else:
    result = NOT an isosceles right triangle

print(f\nThe points form {result}.)
```

Python Direct Code

```python
# ---- Plotting ----
x_AB = line_gen(A, B)
x_BC = line_gen(B, C)
x_CA = line_gen(C, A)

plt.plot(x_AB[0, :], x_AB[1, :], 'b')
plt.plot(x_BC[0, :], x_BC[1, :], 'b')
plt.plot(x_CA[0, :], x_CA[1, :], 'b')

# Mark points
tri_coords(A, B, C, ['A(7,10)', 'B(-2,5)', 'C(3,4)'])

# Title
plt.title(fTriangle ABC: {result})
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.show()
```

Python Shared Output

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

# Define C int type
c_int = ctypes.c_int

def main():
    # Coordinates of the points (numpy arrays)
    A = np.array([c_int(7).value, c_int(10).value])
    B = np.array([c_int(-2).value, c_int(5).value])
    C = np.array([c_int(3).value, c_int(4).value])

    # Squared lengths of sides
    AB2 = c_int(np.sum((A - B) ** 2))
    AC2 = c_int(np.sum((A - C) ** 2))
    BC2 = c_int(np.sum((B - C) ** 2))
```

Python Shared Output

```
    # Dot products (for right angle check)
    dot_AB_AC = c_int(np.dot(A - B, A - C))
    dot_AB_BC = c_int(np.dot(A - B, B - C))
    dot_AC_BC = c_int(np.dot(A - C, B - C))

    print(Squared side lengths:)
    print(fAB^2 = {AB2.value})
    print(fAC^2 = {AC2.value})
    print(fBC^2 = {BC2.value})

    print(\nDot products:)
    print(f(A-B)(A-C) = {dot_AB_AC.value})
    print(f(A-B)(B-C) = {dot_AB_BC.value})
    print(f(A-C)(B-C) = {dot_AC_BC.value})

    # Check isosceles right triangle
    if ((AB2.value == AC2.value and dot_AB_AC.value == 0) or
        (AB2.value == BC2.value and dot_AB_BC.value == 0) or
```

Python Shared Output

```
        (AC2.value == BC2.value and dot_AC_BC.value == 0)):
        result = ISOSCELES RIGHT triangle
    else:
        result = NOT an isosceles right triangle
    print(f\nThe points form {result}.)

    # ---- Plotting ----
    fig, ax = plt.subplots()
    # Draw triangle edges
    triangle = np.array([A, B, C, A]) # closed loop
    ax.plot(triangle[:, 0], triangle[:, 1], 'b-o', linewidth=2)

    # Annotate points
    ax.text(A[0]+0.2, A[1]+0.2, A(7,10), color=red)
    ax.text(B[0]+0.2, B[1]+0.2, B(-2,5), color=red)
    ax.text(C[0]+0.2, C[1]+0.2, C(3,4), color=red)
```

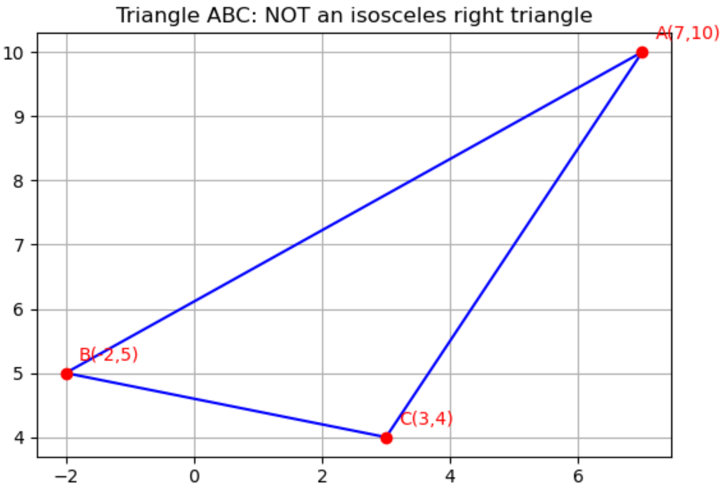
Python Shared Output

```python
    # Title
    ax.set_title(fTriangle ABC: {result})
    ax.set_aspect('equal', adjustable='box')
    ax.grid(True)

    plt.show()

if __name__ == __main__:
    main()s
```

Plot



Triangle ABC: NOT an isosceles right triangle
A(7,10)
B(-2,5)
C(3,4)
10
9
8
7
6
5
4
−2
0
2
4
6