2.4.23

J.NAVYASRI- EE25BTECH11028

september 2025

Question

Do the points $(3, 2)$, $(-2, -3)$, and $(2, 3)$ form a triangle? If so, name the type of triangle formed.

Theoretical solution

Given points,

$$A = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad B = \begin{pmatrix} -2 \\ -3 \end{pmatrix}, \quad C = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \tag{1}$$

Form the matrix:

$$M = \begin{pmatrix} 3 & 2 & 1 \\ -2 & -3 & 1 \\ 2 & 3 & 1 \end{pmatrix} \tag{2}$$

Apply row operations:

$$R_2 \leftarrow R_2 + 2R_1, \quad R_3 \leftarrow 3R_3 - 2R_1 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 4 & 1 & 3 \\ 0 & 5 & 1 \end{pmatrix} \tag{3}$$

$$R_2 \leftarrow 3R_2 - 4R_1 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 0 & -5 & 5 \\ 0 & 5 & 1 \end{pmatrix} \tag{4}$$

Theoretical solution

$$R_3 \leftarrow R_3 + R_2 \quad \Rightarrow \quad \begin{pmatrix} 3 & 2 & 1 \\ 0 & -5 & 5 \\ 0 & 0 & 6 \end{pmatrix} \tag{5}$$

Since all three rows are nonzero:

$$\text{rank}(M) = 3 \tag{6}$$

$\Rightarrow$ Points are not collinear, so they form a triangle.

$$\overrightarrow{AB} = B - A = \begin{pmatrix} -5 \\ -5 \end{pmatrix}, \quad \overrightarrow{AC} = C - A = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{7}$$

Theoretical solution

$$\overrightarrow{AB} \cdot \overrightarrow{AC} = (-5)(-1) + (-5)(1) = 0 \tag{8}$$

$$\Rightarrow \overrightarrow{AB} \perp \overrightarrow{AC}$$

So, the triangle is right-angled at

$$A = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \tag{9}$$

The given points form a triangle (rank = 3). (10)

The triangle is right-angled at $A = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$. (11)

Python Code

```
    import matplotlib.pyplot as plt

# Define the coordinates of the points
A = (3, 2)
B = (-2, -3)
C = (2, 3)
```

Python Code

```
    # Plot lines connecting the points
plt.plot([B[0], A[0]], [B[1], A[1]], 'b-') # Line from B to A
plt.plot([B[0], C[0]], [B[1], C[1]], 'b-') # Line from B to C
plt.plot([A[0], C[0]], [A[1], C[1]], 'b-') # Line from A to C

# Plot the points themselves
plt.plot(A[0], A[1], 'ko') # Point A
plt.plot(B[0], B[1], 'ko') # Point B
plt.plot(C[0], C[1], 'ko') # Point C
```

Python Code

```
# Add labels near the points
plt.text(A[0] + 0.1, A[1], 'A(3,2)')
plt.text(B[0] - 1.5, B[1], 'B(-2,-3)')
plt.text(C[0] - 1, C[1], 'C(2,3)')

# Axes labels
plt.xlabel('x')
plt.ylabel('y')

# Grid and central axes
plt.grid(True)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)

# Title and show plot
plt.title('Graph of Points A, B, C')
plt.show()
```

C Code

```c
#include <stdio.h>
int main() {
    int x1=3,y1=2, x2=-2,y2=-3, x3=2,y3=3;

    int det = x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2);

    if(det==0)
        printf("The points are collinear. No triangle formed.\n")
            ;
    else
        printf("The points form a triangle.\n");

    return 0;
}
```

C Code

```c
#include <stdio.h>
int main() {
    int x1=3,y1=2, x2=-2,y2=-3, x3=2,y3=3;

    int ABx=x2-x1, ABy=y2-y1;
    int ACx=x3-x1, ACy=y3-y1;
    int BCx=x3-x2, BCy=y3-y2;

    if(ABx*ACx + ABy*ACy == 0)
        printf("The triangle is right-angled at A(3,2).\n");
    else if(ABx*BCx + ABy*BCy == 0)
        printf("The triangle is right-angled at B(-2,-3).\n");
    else if(ACx*BCx + ACy*BCy == 0)
        printf("The triangle is right-angled at C(2,3).\n");

    return 0;
}
```

C Code

```c
#include <stdio.h>
int main() {
    printf("Final Conclusion -> Triangle formed, right-angled at
        A(3,2).\n");
    return 0;
}
```

Python and C Code

```
# PART 1: Input & Setup
x1, y1 = 3, 2
x2, y2 = -2, -3
x3, y3 = 2, 3
print("Points: A(3,2), B(-2,-3), C(2,3)")
```

Python and C Code

```
# PART 2: Check Collinearity
x1,y1,x2,y2,x3,y3 = 3,2,-2,-3,2,3
det = x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)
print("Collinear" if det==0 else "Not Collinear")
```

Python and C Code

```
# PART 3: Check Right Angle
x1,y1,x2,y2,x3,y3 = 3,2,-2,-3,2,3
AB, AC, BC = (x2-x1,y2-y1), (x3-x1,y3-y1), (x3-x2,y3-y2)

if AB[0]*AC[0]+AB[1]*AC[1]==0: print("Right angle at A")
elif AB[0]*BC[0]+AB[1]*BC[1]==0: print("Right angle at B")
elif AC[0]*BC[0]+AC[1]*BC[1]==0: print("Right angle at C")
else: print("No right angle")
```

Python and C Code

```
import matplotlib.pyplot as plt

# PART 4: Final Conclusion + Plot
x1,y1,x2,y2,x3,y3 = 3,2,-2,-3,2,3
X,Y = [x1,x2,x3,x1],[y1,y2,y3,y1]
plt.plot(X,Y,'bo-'); plt.grid(True); plt.gca().set_aspect('equal'
    )
plt.text(x1+0.1,y1+0.1,"A"); plt.text(x2+0.1,y2+0.1,"B"); plt.
    text(x3+0.1,y3+0.1,"C")
plt.show()
```

Graphical Representation:

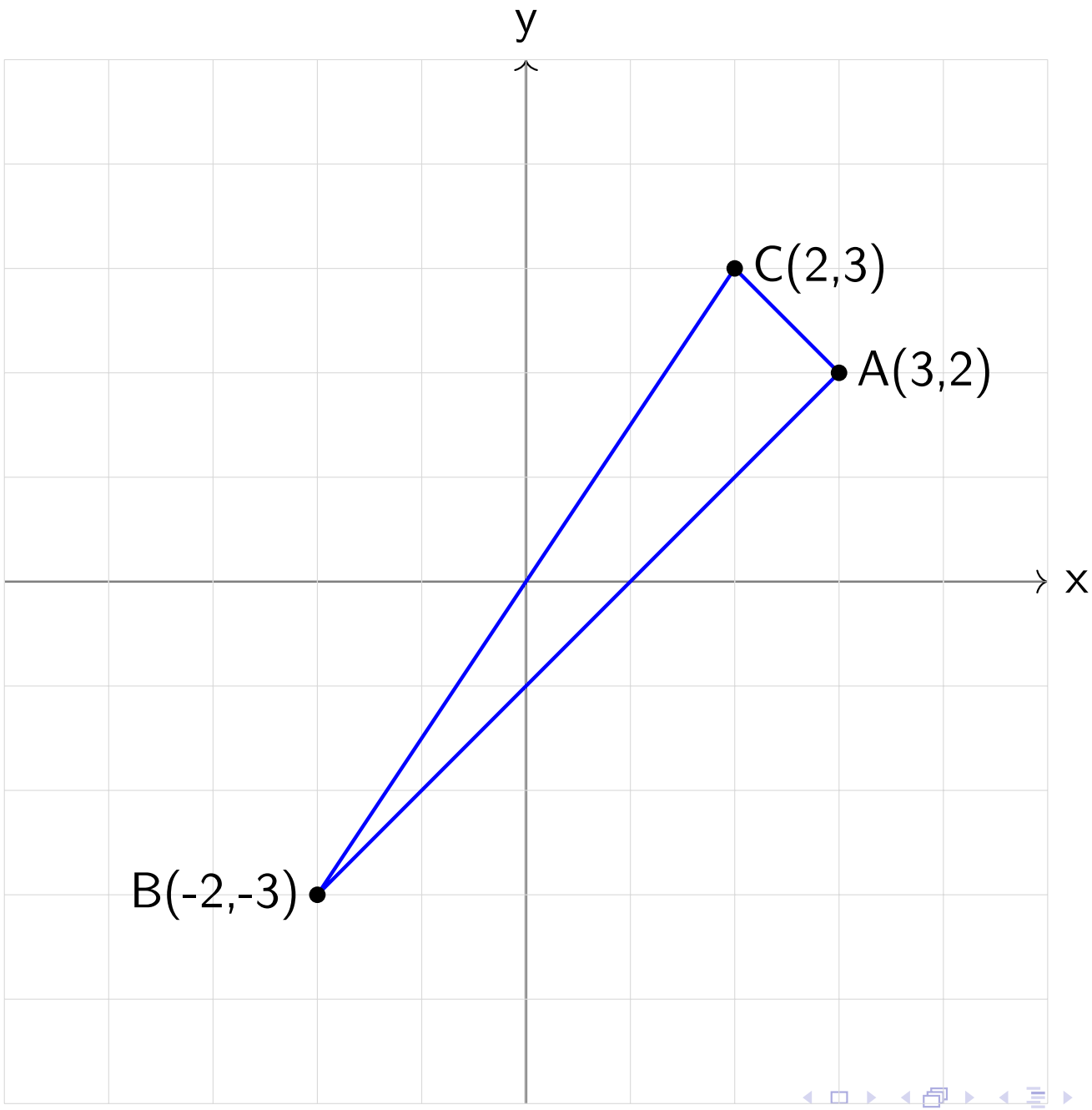

Fig. 0