1.3.7

Manohar-AI25BTECH11028

August 27,2025

Find the coordinates of the vertex A of an ABCD parallelogram whose three vertices are given as B(0,0),C(3,0), and D(0,4).

Given points,

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \mathbf{C} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \mathbf{D} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} \tag{1}$$

we can use the parallelogram property that if ABCD be a parallelogram ,

$$\mathbf{B} - \mathbf{A} = \mathbf{C} - \mathbf{D}. \tag{2}$$

$$\tag{3}$$

Theoretical Solution

In a paralleleogram,

$$\mathbf{A} = \mathbf{B} + \mathbf{D} - \mathbf{C} \tag{4}$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 0 \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} -3 \\ 4 \end{pmatrix} \tag{6}$$

Therefore,

$$\mathbf{A} = \begin{pmatrix} -3 \\ 4 \end{pmatrix} \tag{7}$$

C Code

```c
#include<stdio.h>

int main() {
    int Bx=0, By=0, Cx=3, Cy=0, Dx=0, Dy=4;
    int Ax, Ay;

    // Formula: A = B + D - C
    Ax = Bx + Dx - Cx;
    Ay = By + Dy - Cy;

    printf("Coordinates of A: (%d, %d)\n", Ax, Ay);
    return 0;
}
```

Python Code

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

# Given vertices
B = np.array([0, 0])
C = np.array([3, 0])
D = np.array([0, 4])

# Compute A = B + D - C
A = B + D - C
print("Coordinates of A:", A)
```

Python Code

```
    # Plot parallelogram
x_points = [A[0], B[0], C[0], D[0], A[0]]
y_points = [A[1], B[1], C[1], D[1], A[1]]

plt.plot(x_points, y_points, 'b-')
plt.scatter([A[0], B[0], C[0], D[0]], [A[1], B[1], C[1], D[1]],
    color='r')
```

Python Code

```python
    plt.text(A[0]-0.3, A[1]+0.2, 'A(-3,4)')
plt.text(B[0]-0.3, B[1]-0.3, 'B(0,0)')
plt.text(C[0]+0.2, C[1]-0.3, 'C(3,0)')
plt.text(D[0]-0.3, D[1]+0.3, 'D(0,4)')

plt.axis('equal')
plt.grid(True)
plt.show()
```

```python
# Save before show
plt.savefig("/storage/emulated/0/matrix/Matgeo/1.3.7/figs/
    Figure_1.png", dpi=300, bbox_inches='tight')
plt.show()
```

Plot