1.6.15

Yoshita J - EE25BTECH11065

August 23,2025

Question

Find the value of m if the points $(5, 1)$, $(\,-2, -3)$ and $(8, 2m)$ are collinear

Theoretical Solution

Let $\mathbf{A}(5, 1)$, $\mathbf{B}(-2, -3)$, $\mathbf{C}(8, 2m)$.
Using the collinearity $(rank)$ test, form the matrix with difference vectors:

$$\begin{aligned}\begin{pmatrix}\mathbf{B}-\mathbf{A} & \mathbf{C}-\mathbf{A}\end{pmatrix} &= \begin{pmatrix}-2-5 & 8-5 \\ -3-1 & 2m-1\end{pmatrix} \\ &= \begin{pmatrix}-7 & 3 \\ -4 & 2m-1\end{pmatrix}.\end{aligned}$$

Equation

The three points are collinear $\iff$ this matrix has rank 1 $(\textit{its rows are linearly dependent})$**.**

Theoretical Solution

Using Gauss–Jordan elimination,

$$R_2 \leftarrow 7R_2 - 4R_1 \implies \begin{pmatrix} -7 & 3 \\ 0 & 14m - 19 \end{pmatrix}.$$

For rank 1, the second row must be zero:

$$14m - 19 = 0 \implies m = \tfrac{19}{14}$$

C Code - A function to find the value of m

```
#include <stdio.h>
float find_collinear_m(float Ax, float Ay, float Bx, float By,
    float Cx, float coeff_m) {
    float numerator = (By - Ay) * (Cx - Ax) + Ay * (Bx - Ax);
    float denominator = coeff_m * (Bx - Ax);
    if (denominator == 0) {
        return 0;
    }
    return numerator / denominator;
}
```

Python Code

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

try:
    c_lib = ctypes.CDLL('./code.so')
except OSError:
    print(Error: 'code.so' not found.)
    print(Please ensure you have a 'code.c' file with the find_collinear_m function)
    print(and that you have a C compiler (like gcc) installed to compile it.)
    exit()
```

Python Code

```
c_lib.find_collinear_m.argtypes = [
    ctypes.c_float, ctypes.c_float, # Ax, Ay
    ctypes.c_float, ctypes.c_float, # Bx, By
    ctypes.c_float, ctypes.c_float # Cx, coeff_m
]

c_lib.find_collinear_m.restype = ctypes.c_float


A = np.array([5.0, 1.0])
B = np.array([-2.0, -3.0])
Cx = 8.0
coeff_m = 2.0 # The y-coordinate of C is 2*m
```

Python Code

```
m_value = c_lib.find_collinear_m(
    ctypes.c_float(A[0]), # Ax
    ctypes.c_float(A[1]), # Ay
    ctypes.c_float(B[0]), # Bx
    ctypes.c_float(B[1]), # By
    ctypes.c_float(Cx), # Cx
    ctypes.c_float(coeff_m) # coeff_m
)

C = np.array([Cx, coeff_m * m_value])

print(fThe C function calculated m = {m_value:.4f})
print(fThis corresponds to point C being at ({C[0]:.2f}, {C[1]:.2f}))
print(The exact value of m is 19/14.)
```

Python Code

```python
slope = (B[1] - A[1]) / (B[0] - A[0])
intercept = A[1] - slope * A[0]

x_min = min(A[0], B[0], C[0]) - 2
x_max = max(A[0], B[0], C[0]) + 2
x_line = np.linspace(x_min, x_max, 100)
y_line = slope * x_line + intercept

plt.plot(x_line, y_line, label=f'Line through A, B, and C (m={m_value:.2f})', color='blue')

all_points = np.vstack((A, B, C)).T
plt.scatter(all_points[0, :], all_points[1, :], color='red', zorder=5)
```

Python Code

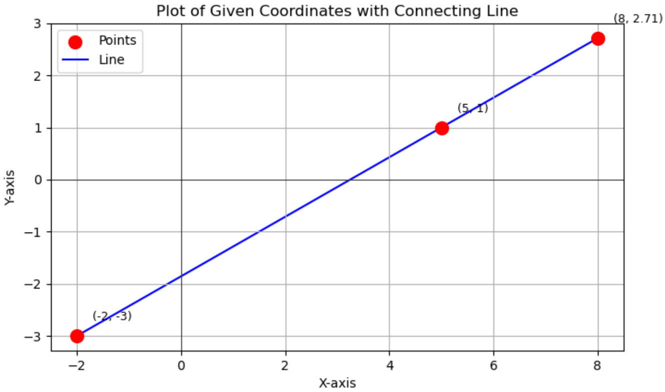
```
point_labels = [f'A ({A[0]},{A[1]})', f'B ({B[0]},{B[1]})', f'C ({C[0]:.1f},{C[1]:.2f})']
for i, txt in enumerate(point_labels):
    plt.annotate(txt,
                (all_points[0, i], all_points[1, i]),
                textcoords=offset points,
                xytext=(10, 5),
                ha='center')


plt.xlabel('$x$')
plt.ylabel('$y$')
plt.title('Visualization of Collinear Points')
plt.legend(loc='best')
plt.grid(True)
plt.axis('equal')

plt.savefig('..figs\collinear_points.png')

```

Plot


Plot of Given Coordinates with Connecting Line
● Points
Line
(8, 2.71)
(5, 1)
(-2, -3)
3
2
1
0
−1
−2
−3
Y-axis
−2
0
2
4
6
8
X-axis