1.6.3

Shriyansh Chawda-EE25BTECH11052 August 23, 2025

Determine if the points (1, 5), (2, 3) and (-2, -11) are collinear.

Solution

Points **A**, **B**, **C** are defined to be collinear if

$$\text{rank}(\mathbf{B} - \mathbf{A} \quad \mathbf{C} - \mathbf{A}) = 1 \qquad (1.1.9.1)$$

Let $\mathbf{A} = (1, 5)$, $\mathbf{B} = (2, 3)$, $\mathbf{C} = (-2, -11)$. From this, the collinearity matrix can be expressed as

$$\left(\begin{array}{cc} \mathbf{B} - \mathbf{A} & \mathbf{C} - \mathbf{A} \end{array} \right) = \begin{pmatrix} 1 & -3 \\ -2 & -16 \end{pmatrix} \xrightarrow{R_2 \leftarrow R_2 + 2R_1} \begin{pmatrix} 1 & -3 \\ 0 & -22 \end{pmatrix}$$

which is a rank 2 matrix. Using (1.1.9.1), the above-mentioned property, we conclude that the given points are **not** collinear.

C Code - Collinearity Formula

```c
#include <math.h>

// Return 1 if collinear, 0 otherwise
int check_collinearity(double x1, double y1,
double x2, double y2,
double x3, double y3,
double tol)
{
        double det = (x2 - x1) * (y3 - y1) -
            (y2 - y1) * (x3 - x1);
        return fabs(det) <= tol ? 1 : 0;
}
```

Python Code

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


lib = ctypes.CDLL('./libcollinearity.so')
lib.check_collinearity.argtypes =
    [ctypes.c_double, ctypes.c_double,
ctypes.c_double, ctypes.c_double,
ctypes.c_double, ctypes.c_double,
ctypes.c_double]
lib.check_collinearity.restype = ctypes.c_int

def check_collinearity(A, B, C, tol=1e-9):
return lib.check_collinearity(A[0], A[1], B[0],
    B[1], C[0], C[1], tol)
```

Python Code

```python
A = (1,5)
B = (2,3)
C = (-2,-11)

is_collinear = check_collinearity(A, B, C)
print("Collinear" if is_collinear else "Not
    collinear")


plt.figure(figsize=(8, 8))


x_line = np.linspace(min(A[0], C[0]) - 1,
    max(A[0], C[0]) + 1, 100)
slope = (C[1] - A[1]) / (C[0] - A[0])
y_line = A[1] + slope * (x_line - A[0])
plt.plot(x_line, y_line, 'k--', label="Line
    through A & C")
```

Python Code

```python
plt.plot(A[0], A[1], 'ro', label='A')
plt.plot(B[0], B[1], 'go', label='B')
plt.plot(C[0], C[1], 'bo', label='C')

plt.text(A[0]+0.2, A[1], f"A{A}", fontsize=12)
plt.text(B[0]+0.2, B[1], f"B{B}", fontsize=12)
plt.text(C[0]+0.2, C[1], f"C{C}", fontsize=12)

mp.use("TkAgg")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Collinearity Check')
plt.legend()
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
```

Python Code

```
plt.savefig(
"/home/shriyasnh/Desktop/matgeo/1.6.3/codes",
dpi=300,
bbox_inches='tight'
)
plt.show()
```
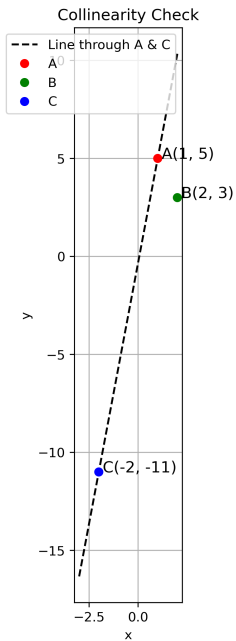
Figure: