

## 1.6.9

Vaishnavi - EE25BTECH11059

August 29, 2025

# Question

if three points  $\begin{pmatrix} h \\ 0 \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} 0 \\ k \end{pmatrix}$  lie on a line, show that

$$\frac{a}{h} + \frac{b}{k} = 1$$

Point	Name
$(h, 0)$	Point A
$(0, k)$	Point B
$(a, b)$	Point C

Table: Variables Used

# Solutions

If the rank of the Collinearity matrix is 1, then the points are collinear  
The Collinearity matrix is given by

$$(\mathbf{C} - \mathbf{A} \quad \mathbf{B} - \mathbf{A})^T = \begin{pmatrix} a-h & b \\ -h & k \end{pmatrix} \quad (1)$$

$$\xleftrightarrow{R_1 \rightarrow \frac{R_1}{a-h}} \begin{pmatrix} 1 & \frac{b}{a-h} \\ -h & k \end{pmatrix} \quad (2)$$

$$\xleftrightarrow{R_2 \rightarrow \frac{R_2}{-h}} \begin{pmatrix} 1 & \frac{b}{a-h} \\ 1 & \frac{-k}{h} \end{pmatrix} \quad (3)$$

$$\xleftrightarrow{R_1 \rightarrow R_1 - R_2} \begin{pmatrix} 0 & \frac{b}{a-h} + \frac{k}{h} \\ 1 & \frac{-k}{h} \end{pmatrix} \quad (4)$$

# Solution

since the rank of matrix=1

$$\frac{b}{a-h} + \frac{k}{h} = 0$$
$$\implies bh + ka - kh = 0 \text{ (dividing the eq with } kh) \implies \frac{a}{h} + \frac{b}{k} = 1$$

# Graph

Refer to Fig.

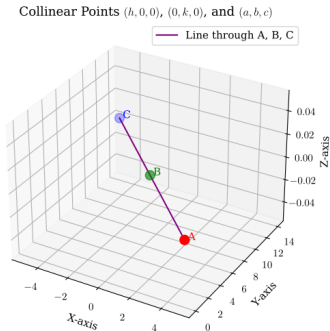


Figure:

```
# Plotting points A(1, -2, -8), B(5, 0, -2), and C(11, 3, 7)

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Define the points as numpy arrays
A = np.array([1, -2, -8])
B = np.array([5, 0, -2])
C = np.array([11, 3, 7])
```

# Python Code

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
```

```
# Use LaTeX-compatible font settings
```

```
matplotlib.use('pgf')
plt.rcParams.update({
    text.usetex: True,
    font.family: 'modern',
    font.size: 11,
})
```

```
# Define points A and B
```

```
h, k = 5, 7
A = np.array([h, 0, 0])
B = np.array([0, k, 0])
```

```
# Compute a third point C that lies on the line AB using
```



# Python Code

```
# 3D Plot
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

# Plot the points
ax.scatter(points[:,0], points[:,1], points[:,2], color=['red', 'green', 'blue'], s=100)
ax.plot(points[:,0], points[:,1], points[:,2], color='purple', label='Line through A, B, C')

# Annotate the points
ax.text(*A, 'A', color='red')
ax.text(*B, 'B', color='green')
ax.text(*C, 'C', color='blue')
```

```
# Axis labels
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title(r'Collinear Points  $(h,0,0)$ ,  $(0,k,0)$ , and  $(a,b,c)$ 
            ')
ax.legend()
ax.grid(True)

# Save the figure
fig.savefig('collinear_3d_plot.png')
```

# C Code

```
#include <stdio.h>
#include <stdbool.h>

// Function to check collinearity using the matrix method
bool check_collinearity_matrix(int h, int a, int b, int k) {
    if (h == 0 || k == 0) {
        printf(Invalid input: h and k must be non-zero.\n);
        return false;
    }

    // Step 1: Construct the matrix from vector differences
    int row1_col1 = a - h;
    int row1_col2 = b;
    int row2_col1 = -h;
    int row2_col2 = k;

    printf(Collinearity matrix before row operations:\n);
    printf([ %d\t%d ]\n, row1_col1, row1_col2);
    printf([ %d\t%d ]\n, row2_col1, row2_col2);
```

```
printf("\nAfter row operation R1 = R1 - R2:\n");
printf([ %d\t%d ]\n, new_r1_col1, new_r1_col2);
printf([ %d\t%d ]\n, row2_col1, row2_col2);

// Step 3: Check the condition (a/h + b/k == 1) without
           floating point
int lhs = a * k + b * h;
int rhs = h * k;

printf("\nChecking condition: (a/h + b/k == 1)\n");
printf(Computed: (%d * %d + %d * %d) = %d\n, a, k, b, h, lhs)
;
printf(Expected: (%d * %d) = %d\n, h, k, rhs);
```

```
printf(Checking collinearity for points:\n);  
printf(A = (%d, 0), B = (%d, %d), C = (0, %d)\n\n, h, a, b, k  
    );  
  
if (check_collinearity_matrix(h, a, b, k)) {  
    printf(\nPoints are collinear (Matrix rank = 1 and a/h +  
        b/k = 1).\n);  
} else {  
    printf(\n Points are NOT collinear (Condition fails).\n);  
}  
  
return 0;  
}
```

```
import subprocess

# Compile the C program
subprocess.run([gcc, points.c, -o, points])

# Run the compiled C program
result = subprocess.run([./points], capture_output=True, text=True)

# Print the output from the C program (solution)
print(result.stdout)
```