1.7.4

Hemanth Reddy-AI25BTECH11018

September 8,2025

Question

Using vectors, prove that the points(2,-1,3), (3,-5,1),and(-1,11,9) are collinear.

Theoretical Solution

Solution:

Let $\mathbf{A}\begin{pmatrix}2\\-1\\3\end{pmatrix}$ $\mathbf{B}\begin{pmatrix}3\\-5\\1\end{pmatrix}$ $\mathbf{C}\begin{pmatrix}-1\\11\\9\end{pmatrix}$ be vectors

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

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

$$\text{Let}\ \ (\mathbf{B}-\mathbf{A} \quad \mathbf{C}-\mathbf{A}) = \mathbf{D} \tag{2}$$

$$\text{rank}\mathbf{D} = \text{rank}\mathbf{D}^T \tag{3}$$

Theoretical Solution

$$\mathbf{D}^T = \begin{pmatrix} 1 & -4 & -2 \\ -3 & 12 & 6 \end{pmatrix} \tag{4}$$

$$R_2 = R_2 + 3R_2 \tag{5}$$

$$\mathbf{D}^T = \begin{pmatrix} 1 & -4 & -2 \\ 0 & 0 & 0 \end{pmatrix} \tag{6}$$

which has rank 1.So we can conclude that the given points are collinear.

C Code

```c
#include <stdio.h>

// Function to compute cross product of two vectors in 3D
void crossProduct(int v1[3], int v2[3], int cross[3]) {
    cross[0] = v1[1]*v2[2] - v1[2]*v2[1];
    cross[1] = v1[2]*v2[0] - v1[0]*v2[2];
    cross[2] = v1[0]*v2[1] - v1[1]*v2[0];
}

// Function to check if three points are collinear in 3D
int areCollinear(int A[3], int B[3], int C[3]) {
    int AB[3], AC[3], cross[3];
    for (int i = 0; i < 3; i++) {
        AB[i] = B[i] - A[i];
        AC[i] = C[i] - A[i];
    }
```

C Code

```
    crossProduct(AB, AC, cross);
    return (cross[0] == 0 && cross[1] == 0 && cross[2] == 0);
}

int main() {
    int A[3] = {2, -1, 3};
    int B[3] = {3, -5, 1};
    int C[3] = {-1, 11, 9};
    if (areCollinear(A, B, C))
        printf("The points are collinear.\n");
    else
        printf("The points are not collinear.\n");
    return 0;
}
```

Python Code

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

# Points
A = np.array([2, -1, 3])
B = np.array([3, -5, 1])
C = np.array([-1, 11, 9])

# Check collinearity (print for reference)
AB = B - A
AC = C - A

print("Vector AB:", AB)
print("Vector AC:", AC)
```

Python Code

```
# Verify if AC is a scalar multiple of AB
if np.allclose(AC, (AC[0]/AB[0]) * AB):
    print("Points are collinear")
else:
    print("Points are not collinear")

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

# Plot points
ax.scatter(*A, color='red', label='A(2, -1, 3)')
ax.scatter(*B, color='green', label='B(3, -5, 1)')
ax.scatter(*C, color='blue', label='C(-1, 11, 9)')

# Plot line through A in direction of AB (which also passes
    through B and C)
```

Python Code

```
t = np.linspace(-2, 2, 100)
line = A[:, None] + np.outer(AB, t)

ax.plot(line[0], line[1], line[2], 'k--', label='Line through A,
    B, and C')

# Labels and legend
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
ax.set_title('Collinear Points in 3D')

# Save figure as PNG
plt.savefig("collinear_points.png")

# Show plot
plt.show()
```

Plot

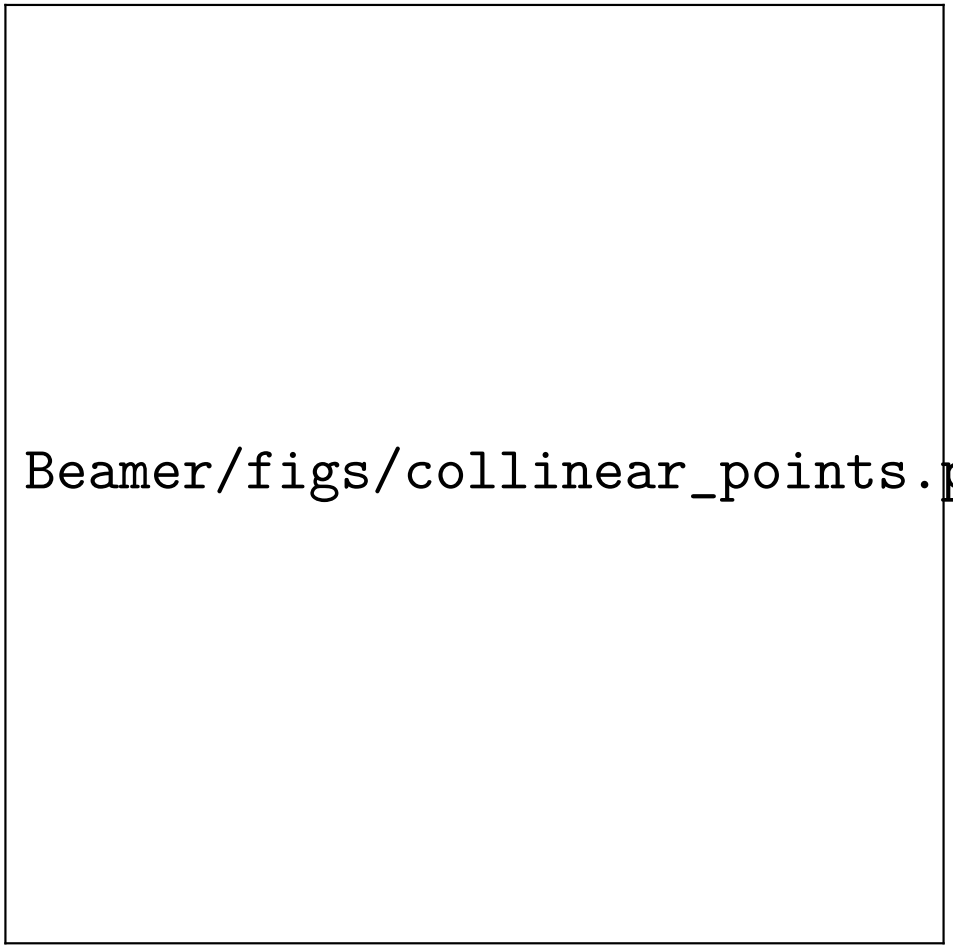

Figure: