1.6.18

AI25BTECH11003 - Bhavesh Gaikwad

August 26,2025

Question

Prove that points A(2,1), B(0,5) and C(-1,2) are not collinear.

Theoretical Solution

B-A$=\begin{pmatrix}0-2\\5-1\end{pmatrix}=\begin{pmatrix}-2\\4\end{pmatrix}$ $\quad C-A=\begin{pmatrix}-1-2\\2-1\end{pmatrix}=\begin{pmatrix}-3\\1\end{pmatrix}$

M $=\begin{pmatrix}B-A & C-A\end{pmatrix}=\begin{pmatrix}-2 & -3\\4 & 1\end{pmatrix}$

Row-reduce to compute the rank:

$$\begin{pmatrix}-2 & -3\\4 & 1\end{pmatrix}\xrightarrow{R_2\leftarrow R_2+2R_1}\begin{pmatrix}-2 & -3\\0 & -5\end{pmatrix}$$

The echelon form has two nonzero rows, hence rank(M)=2≠1

$$\boxed{\textit{Therefore, The points } A(2,1),\ B(0,5) \textit{ and } C(-1,2) \textit{ are not collinear}.} \tag{1}$$

Python + C Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "libs/matfun.h"
#include "libs/geofun.h"

int main(void) {
    // Points as 2x1 column vectors
    double **A = createMat(2,1);
    double **B = createMat(2,1);
    double **C = createMat(2,1);

    // Set coordinates correctly
    A[0][0] = 2.0; A[1][0] = 1.0;
    B[0][0] = 0.0; B[1][0] = 5.0;
    C[0][0] = -1.0; C[1][0] = 2.0;

    // Calculate direction vectors B-A and C-A
    double **BA = Matsub(B, A, 2, 1);
```

Python + C Code

```c
// Calculate determinant to find rank
    double det = M[0][0] * M[1][1] - M[0][1] * M[1][0];

    int rank;
    if (fabs(det) > 1e-10) { // Use fabs for absolute value
        rank = 2;
    } else {
        // Check if at least one row is non-zero
        if ((fabs(M[0][0]) > 1e-10 || fabs(M[0][1]) > 1e-10) ||
            (fabs(M[1][0]) > 1e-10 || fabs(M[1][1]) > 1e-10)) {
            rank = 1;
        } else {
            rank = 0;
        }
    }
```

Python + C Code

```c
// Output result
    if (rank == 2) {
        printf("The points A, B, and C are NOT collinear (Rank = 2)\n");
    } else if (rank == 1) {
        printf("The points A, B, and C are collinear (Rank = 1)\n");
    } else {
        printf("All points coincide (Rank = 0)\n");
    }

    // Write points to .dat file
    FILE *f = fopen("points.dat", "w");
    if (f == NULL) {
        printf("Error opening file!\n");
        return 1;
    }
```

Python + C Code

```c

    fprintf(f, "%.6f\t%.6f\n", A[0][0], A[1][0]);
    fprintf(f, "%.6f\t%.6f\n", B[0][0], B[1][0]);
    fprintf(f, "%.6f\t%.6f\n", C[0][0], C[1][0]);
    fclose(f);

    printf("Points written to points.dat\n");

    // Free memory
    freeMat(A, 2);
    freeMat(B, 2);
    freeMat(C, 2);
    freeMat(BA, 2);
    freeMat(CA, 2);
    freeMat(M, 2);

    return 0;
}
```

Python + C Code

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

# Define the points
points = [(2, 1), (0, 5), (-1, 2)]
labels = ['A(2,1)', 'B(0,5)', 'C(-1,2)']
colors = ['blue', 'orange', 'green']

plt.figure(figsize=(6, 6))

# Plot the points
for i, ((x, y), label, color) in enumerate(zip(points, labels, colors)):
    plt.scatter(x, y, color=color, s=100, zorder=5)
    plt.text(x + 0.1, y + 0.1, label, fontsize=12, fontweight='bold')
```

Python + C Code

```
# Draw dashed lines connecting the points
# Line from B to A
plt.plot([0, 2], [5, 1], '--', color='gray', alpha=0.7, linewidth=1.5)
# Line from C to A
plt.plot([-1, 2], [2, 1], '--', color='gray', alpha=0.7, linewidth=1.5)

# Set up the grid and axes
plt.grid(True, alpha=0.4, linestyle=':')
plt.axhline(0, color='black', linewidth=1)
plt.axvline(0, color='black', linewidth=1)

# Set axis limits to match the reference image
plt.xlim(-2, 4)
plt.ylim(0, 7)
```

Python + C Code

```
# Set tick marks
plt.xticks(np.arange(-2, 5, 1))
plt.yticks(np.arange(0, 8, 1))

# Add title
plt.title('Points A, B, C', fontsize=14, fontweight='bold', pad
    =20)

# Remove axis labels (x and y)
plt.xlabel('')
plt.ylabel('')

# Save the figure
plt.savefig('fig1.png', dpi=150, bbox_inches='tight')
plt.close()

print("Plot saved as fig1.png")
```

Python Code

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

# Define the coordinates directly
points = {
    'A(2,1)': (2, 1),
    'B(0,5)': (0, 5),
    'C(-1,2)': (-1, 2)
}

# Extract data for plotting
labels = list(points.keys())
xs = [coord[0] for coord in points.values()]
ys = [coord[1] for coord in points.values()]
colors = ['blue', 'orange', 'green']

plt.figure(figsize=(6, 6))
```

Python Code

```
# Plot each point with its label
for (label, (x, y)), color in zip(points.items(), colors):
    plt.scatter(x, y, color=color, s=100, zorder=5)
    plt.text(x + 0.1, y + 0.1, label, fontsize=12, fontweight='
        bold')

# Draw dashed lines connecting B to A and C to A
plt.plot([0, 2], [5, 1], '--', color='gray', alpha=0.7, linewidth
    =1.5)
plt.plot([-1, 2], [2, 1], '--', color='gray', alpha=0.7,
    linewidth=1.5)

# Grid and axes
plt.grid(True, alpha=0.4, linestyle=':')
plt.axhline(0, color='black', linewidth=1)
plt.axvline(0, color='black', linewidth=1)
```

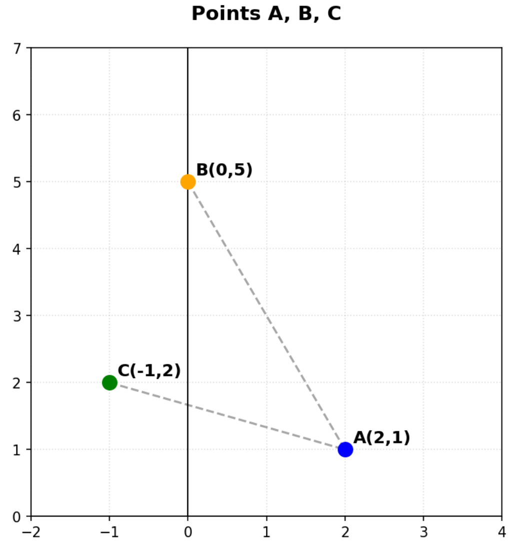
Python Code

```python
# Axis limits and ticks
plt.xlim(-2, 4)
plt.ylim(0, 7)
plt.xticks(np.arange(-2, 5, 1))
plt.yticks(np.arange(0, 8, 1))

# Title and axis labels
plt.title('Points A, B, C', fontsize=14, fontweight='bold', pad
    =20)
plt.xlabel('')
plt.ylabel('')

# Save the figure
plt.savefig('fig1.png', dpi=150, bbox_inches='tight')
plt.close()
```

Graph


Points A, B, C
B(0,5)
C(-1,2)
A(2,1)