Matgeo Presentation

K. Akshay Teja
AI24BTECH11002
IIT Hyderabad

November 5, 2024

Outline

1. Problem
2. Solution
   - Variable Description
   - Collinearity Condition
   - Matrix and Row Reduction
3. Conclusion
4. Codes
   - Generating points on line using C
   - Plotting the figure using Python

Problem Statement

Show that the points

$$P = \begin{pmatrix} -2 \\ 3 \\ 5 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad R = \begin{pmatrix} 7 \\ 0 \\ -1 \end{pmatrix} \tag{1.1}$$

are collinear.

Collinearity Condition

The points P, Q, and R are collinear if the vectors $Q - P$ and $R - P$ are linearly dependent. This is equivalent to the rank condition:

$$\text{rank}\begin{pmatrix} Q - P & R - P \end{pmatrix} = 1 \tag{2.1}$$

Vector Calculation

Calculate the vectors $Q - P$ and $R - P$:

$$Q - P = \begin{pmatrix} 1-(-2) \\ 2-3 \\ 3-5 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ -2 \end{pmatrix}, \quad R - P = \begin{pmatrix} 7-(-2) \\ 0-3 \\ -1-5 \end{pmatrix} = \begin{pmatrix} 9 \\ -3 \\ -6 \end{pmatrix} \tag{2.2}$$

Matrix Row Reduction

Perform row reduction on the matrix:

$$\begin{pmatrix} 3 & -1 & -2 \\ 9 & -3 & -6 \end{pmatrix} \xrightarrow{R_2 \leftarrow R_2 - 3R_1} \begin{pmatrix} 3 & -1 & -2 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.3}$$

Since the matrix has a rank of 1, the points P, Q, and R are collinear.

Conclusion

Since the matrix formed by $Q - P$ and $R - P$ has a rank of 1, the points P, Q, and R are collinear.

Visualization

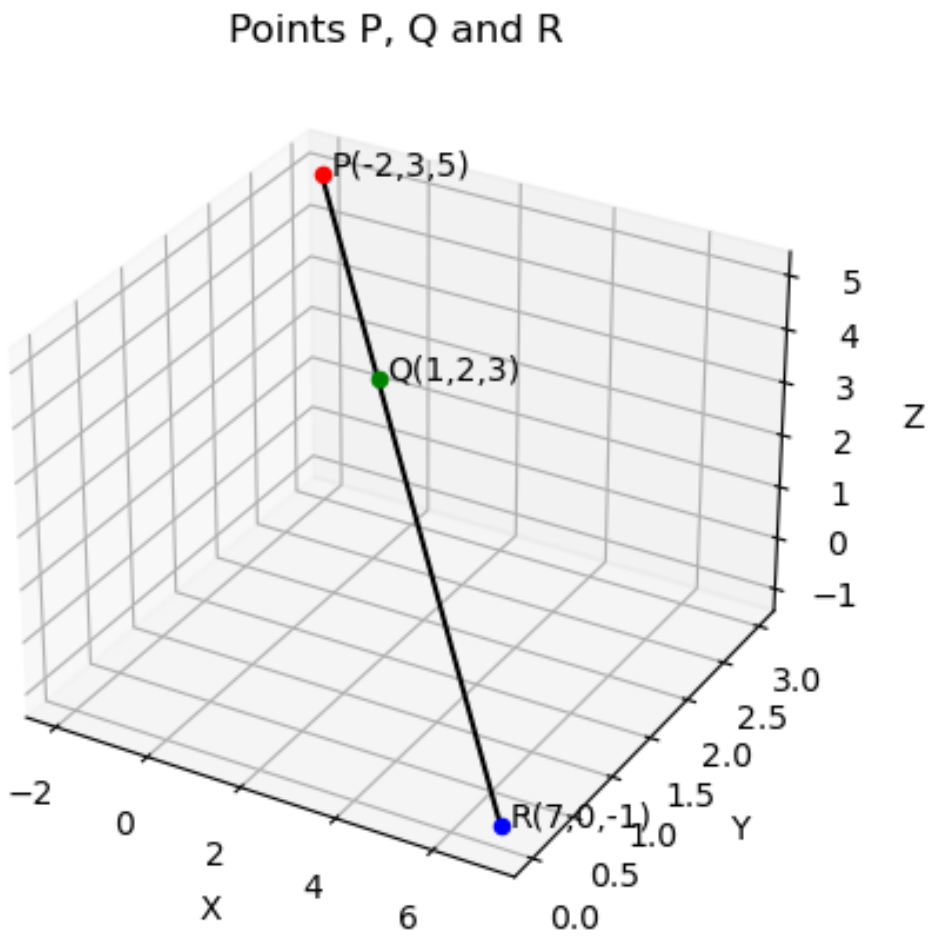

Figure: Plot of Points P, Q, and R

Generating Points on Line using C I

```c
#include<stdio.h>

int main(){
        double P[]={-2,3,5},Q[]={1,2,3},R[]={7,0,-1};

        printf("P %.2lf %.2lf %.2lf\n",P[0],P[1],P[2]);
        printf("Q %.2lf %.2lf %.2lf\n",Q[0],Q[1],Q[2]);
        printf("R %.2lf %.2lf %.2lf\n",R[0],R[1],R[2]);
        int numberOfValues=100;

        double
        ↪  x_values[numberOfValues],y_values[numberOfValues],z_values[numberO

        for(int i=0;i<numberOfValues;i++){
                double t=(double)i/numberOfValues;
                x_values[i]=P[0]+t*(R[0]-P[0]);
                y_values[i]=P[1]+t*(R[1]-P[1]);
                z_values[i]=P[2]+t*(R[2]-P[2]);
        }
        for (int i = 0; i < numberOfValues; i++) {
```

Generating Points on Line using C II

```
20          printf("%.2f %.2f %.2f\n", x_values[i], y_values[i], z_values[i]);
21      }
22
23      return 0;
24  }
```

Plotting the Figure using Python I

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from mpl_toolkits.mplot3d import Axes3D
4  import subprocess
5
6  result = subprocess.run(['./code'],stdout = subprocess.PIPE,text=True)
7  output = result.stdout.strip().split('\n')
8
9  P = np.fromstring(output[0].replace('P ',''),sep=' ')
10 Q = np.fromstring(output[1].replace('Q ',''),sep=' ')
11 R = np.fromstring(output[2].replace('R ',''),sep=' ')
12
13 store=np.genfromtxt(output[3:],delimiter='')
14 x_values,y_values,z_values = store.T
15
16 fig = plt.figure()
17 ax = fig.add_subplot(111, projection='3d')
18
19 ax.scatter(*P, color='r', label='P')
20 ax.scatter(*Q, color='g', label='Q')
```

Plotting the Figure using Python II

```
21 ax.scatter(*R, color='b', label='R')
22
23 ax.text(P[0]+0.2,P[1],P[2],'P(-2,3,5)',color='black', ha='left')
24 ax.text(Q[0]+0.2,Q[1],Q[2],'Q(1,2,3)',color='black', ha='left')
25 ax.text(R[0]+0.2,R[1],R[2],'R(7,0,-1)',color='black', ha='left')
26
27 ax.plot(x_values,y_values,z_values,color='k',label='Line through P,Q,R')
28
29 ax.set_xlabel('X')
30 ax.set_ylabel('Y')
31 ax.set_zlabel('Z')
32
33 plt.title('Points P, Q and R')
34 plt.grid(True)
35 plt.savefig('/home/akshay-teja-kondi/gvv/Assignment3/fig/fig.png')
```