1.5.14

Harsha-EE25BTECH11026

August 21,2025

Question

Points P and Q trisect the line segment joining the points A $(-2, 0)$ and B$(0, 8)$ such that P is nearer to A. Find the coordinates of points P and Q.

Theoretical Solution

Let the vectors $\mathbf{P}$ and $\mathbf{Q}$ be

$$\mathbf{P} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} , \mathbf{Q} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{1}$$

Given the points,

$$\mathbf{A} = \begin{pmatrix} -2 \\ 0 \end{pmatrix} \mathbf{B} = \begin{pmatrix} 0 \\ 8 \end{pmatrix} \tag{2}$$

we can use the internal division formula to find the points $\mathbf{P}$ and $\mathbf{Q}$.

Equation

Internal division formula for a vector R which divides the line formed by vectors A and B in the ratio m:n is given by

$$\mathbf{R} = \frac{m\mathbf{B} + n\mathbf{A}}{m + n} \tag{3}$$

Theoretical Solution

To find vector **P**, as it is near the point A, it divides the line formed by line A and B in ratio 1:2.
Therefore,

$$\mathbf{P} = \frac{2 \times \begin{pmatrix} -2 \\ 0 \end{pmatrix} + 1 \times \begin{pmatrix} 0 \\ 8 \end{pmatrix}}{1+2} \tag{4}$$

$$\mathbf{P} = \begin{pmatrix} \frac{-4}{3} \\ \frac{8}{3} \end{pmatrix} \tag{5}$$

Theoretical Solution

To find vector $\mathbf{Q}$, as it is near the point B, it divides the line formed by line A and B in ratio 2:1.
Therefore,

$$\mathbf{Q} = \frac{1 \times \begin{pmatrix} -2 \\ 0 \end{pmatrix} + 2 \times \begin{pmatrix} 0 \\ 8 \end{pmatrix}}{2+1} \tag{6}$$

$$\mathbf{Q} = \begin{pmatrix} \frac{-2}{3} \\ \frac{16}{3} \end{pmatrix} \tag{7}$$

C Code - Internal division formula

```
\#include <stdio.h>

void find\_section\_point(double \;x1, double \;y1, double \;x2,
    double y2, double m, double n, double* x, double* y) {
    *x = (m * x2 + n * x1) / (m + n);
    *y = (m * y2 + n * y1) / (m + n);
```

Python Code

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

lib = ctypes.CDLL('./libintdiv_formula.so')
lib.find_section_point.argtypes = [ctypes.c_double, ctypes.
    c_double, ctypes.c_double,
                                ctypes.c_double, ctypes.c_double,
                                     ctypes.c_double,
                                ctypes.POINTER(ctypes.c_double),
                                    ctypes.POINTER(ctypes.
                                    c_double)]
lib.find_section_point.restype = None
```

Python Code

```python
def find_section_point(x1, y1, x2, y2, m, n):
    x = ctypes.c_double()
    y = ctypes.c_double()
    lib.find_section_point(x1, y1, x2, y2, m, n, ctypes.byref(x),
         ctypes.byref(y))
    return (x.value, y.value)
```

Python Code

```python
# Given points
A = (-2,0)
B = (0,8)

# Find P such that AP:PB=1:2
P = find_section_point(A[0], A[1], B[0], B[1], 1, 2)
# Find Q such that AQ:QB=2:1
Q = find_section_point(A[0], A[1], B[0], B[1], 2, 1)

# Format results
P_formatted = (round(P[0], 2), round(P[1], 2))
Q_formatted = (round(Q[0], 2), round(Q[1], 2))

print(f"P: {P_formatted}")
print(f"Q: {Q_formatted}")
```

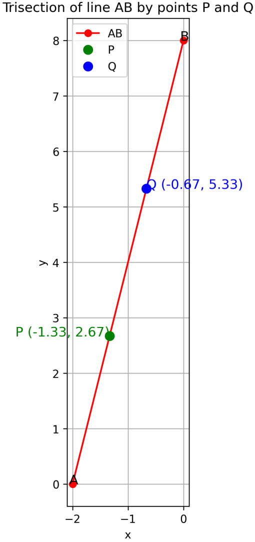
Python Code

```python
# Plotting
plt.figure(figsize=(8, 8))

# Line AB
plt.plot([A[0], B[0]], [A[1], B[1]], 'ro-', label='AB')

# Points P and Q
plt.plot(*P_formatted, 'go', label='P', markersize=8) # green
plt.plot(*Q_formatted, 'bo', label='Q', markersize=8) # blue

# Labels
plt.text(A[0]+0.1, A[1], 'A', fontsize=12, ha='right')
plt.text(B[0]+0.1, B[1], 'B', fontsize=12, ha='right')
plt.text(*P_formatted, f'P {P_formatted}', fontsize=12, ha='right
    ', color='green')
plt.text(*Q_formatted, f'Q {Q_formatted}', fontsize=12, ha='left'
    , color='blue')
```

Python Code

```python
mp.use("TkAgg")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Trisection of line AB by points P and Q')
plt.legend()
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')

# Save before show
plt.savefig("/home/user/Matrix/Matgeo_assignments/1.5.14/figs/Figure_1.png", dpi=300, bbox_inches='tight')
plt.show()
```

Plot


Trisection of line AB by points P and Q
AB
P
Q
B
A
Q (-0.67, 5.33)
P (-1.33, 2.67)
y
x
0
1
2
3
4
5
6
7
8
−2
−1
0