1.4.26

Vector Section Formula

EE25BTECH11010 - Arsh Dhoke

Question

The position vector of the point which divides the join of points $2\mathbf{a} - 3\mathbf{b}$ and $\mathbf{a} + \mathbf{b}$ in the ratio $3 : 1$ is ____________.

Theoretical Solution

$$\mathbf{P} = 2\mathbf{a} - 3\mathbf{b}, \tag{1}$$

$$\mathbf{Q} = \mathbf{a} + \mathbf{b}. \tag{2}$$

Now, the matrix form for **Q** and **P** is:

$$\begin{pmatrix} \mathbf{Q} & \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{a} & \mathbf{b} \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 1 & -3 \end{pmatrix}. \tag{3}$$

Equation

Using section formula, the point $\mathbf{R}$ dividing $\mathbf{Q} - \mathbf{P}$ in ratio $3:1$ is:

$$\mathbf{R} = \frac{3\mathbf{Q} + 1\mathbf{P}}{3+1} \tag{4}$$

$$\mathbf{R} = \frac{1}{4} \cdot \begin{pmatrix} \mathbf{Q} & \mathbf{P} \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} \tag{5}$$

$$\mathbf{R} = \frac{1}{4} \cdot \begin{pmatrix} \mathbf{a} & \mathbf{b} \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} \tag{6}$$

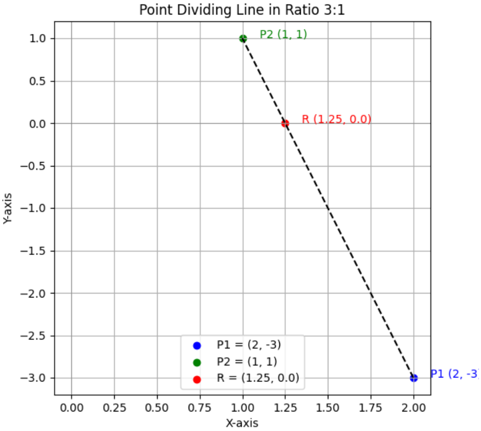
Calculation Steps

$$\mathbf{R} = \frac{1}{4} \cdot \begin{pmatrix} \mathbf{a} & \mathbf{b} \end{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix} \tag{7}$$

$$\mathbf{R} = \frac{1}{4} \cdot \begin{pmatrix} 5\mathbf{a} \end{pmatrix} \tag{8}$$

Let $\mathbf{a} = 1$ and $\mathbf{b} = 1$.

Plot


Point Dividing Line in Ratio 3:1
P2 (1, 1)
R (1.25, 0.0)
P1 (2, -3
P1 = (2, -3)
P2 = (1, 1)
R = (1.25, 0.0)
Y-axis
X-axis
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

C Code - Section formula

```c
#include <stdio.h>

void sectionFormula(int m, int n, float a, float b, float *x) {
    *x = (m * b + n * a) / (float)(m + n);
}
```

Python Code

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

# Define values for a and b
a = 1 # Example value
b = 0 # Example value

# Define points as NumPy arrays
P = np.array([2*a, -3*b]) # Point P
Q = np.array([a, b]) # Point Q

# Ratio m:n
m = 3
n = 1

# Section formula for internal division
R = (m * Q + n * P) / (m + n)

# Print the result
```

Python Code

```python

# Plotting
plt.figure(figsize=(6, 6))
plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)

# Plot P, Q, and R
plt.scatter(*P, color='blue', label='P (2a, -3b)')
plt.scatter(*Q, color='green', label='Q (a, b)')
plt.scatter(*R, color='red', label=f'R (ratio {m}:{n})')

# Draw line between P and Q
plt.plot([P[0], Q[0]], [P[1], Q[1]], color='gray', linestyle='--')

# Annotate points
plt.text(P[0]+0.2, P[1]+0.2, 'P')
plt.text(Q[0]+0.2, Q[1]+0.2, 'Q')
plt.text(R[0]+0.2, R[1]+0.2, 'R')
```

Python Code

```python

plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Section Formula Visualization')
plt.legend()
plt.grid(True)
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo/1.4.26/figs/q1.png")
plt.show()
```

Python+ C Code

```python

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

# Load the shared library
lib = ctypes.CDLL("./libsection_int.so")

# Define argument and return types
lib.sectionFormula.argtypes = [
    ctypes.c_int, ctypes.c_int,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double),
    ctypes.POINTER(ctypes.c_double)
]
lib.sectionFormula.restype = None

# Values for a and b
a = 1
```

Python+ C Code

```python

# Points P and Q
P = (ctypes.c_double * 2)(2 * a, -3 * b)
Q = (ctypes.c_double * 2)(a, b)
R = (ctypes.c_double * 2)(0.0, 0.0)

# Ratio m:n
m, n = 3, 1

# Call the C function
lib.sectionFormula(m, n, P, Q, R)

# Convert to NumPy arrays for plotting
P_np = np.array([P[0], P[1]])
Q_np = np.array([Q[0], Q[1]])
R_np = np.array([R[0], R[1]])

print("Position vector of the point R (from C):", R_np)
```

Python+ C Code

```
# Plotting
plt.figure(figsize=(6, 6))
plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)

# Plot P, Q, and R
plt.scatter(*P_np, color='blue', label='P (2a, -3b)')
plt.scatter(*Q_np, color='green', label='Q (a, b)')
plt.scatter(*R_np, color='red', label=f'R (ratio {m}:{n})')

# Draw line between P and Q
plt.plot([P_np[0], Q_np[0]], [P_np[1], Q_np[1]], color='gray',
    linestyle='--')

# Annotate points
plt.text(P_np[0]+0.2, P_np[1]+0.2, 'P')
plt.text(Q_np[0]+0.2, Q_np[1]+0.2, 'Q')
plt.text(R_np[0]+0.2, R_np[1]+0.2, 'R')
```

Python+ C Code

```
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Section Formula Visualization (Using C & Python)')
plt.legend()
plt.grid(True)
plt.axis('equal')

# Save the plot
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /1.4.26/figs/q1.png")

# Show plot
plt.show()
```