Problem 1.4.25

EE25BTECH11009 – Anshu Kumar Ram

August 30, 2025

Question

Find the position vector of a point R which divides the line joining two points P and Q whose position vectors are $2\vec{a}+\vec{b}$ and $\vec{a}-3\vec{b}$ externally in the ratio $1:2$.

Solution Step 1

Step 1: Represent points in coordinates

$$\vec{P} = 2\vec{a} + \vec{b} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{1}$$

$$\vec{Q} = \vec{a} - 3\vec{b} = \begin{pmatrix} 1 \\ -3 \end{pmatrix} \tag{2}$$

Solution Step 2

Step 2: Apply section formula (external division)

$$\vec{R} = \frac{1 \cdot \vec{Q} - 2 \cdot \vec{P}}{1 - 2} \tag{3}$$

$$= \frac{1}{-1}\left(\begin{pmatrix} 1 \\ -3 \end{pmatrix} - 2\begin{pmatrix} 2 \\ 1 \end{pmatrix}\right) \tag{4}$$

$$= -\begin{pmatrix} -3 \\ -5 \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} 3 \\ 5 \end{pmatrix} \tag{6}$$

$$\vec{R} = 3\vec{a} + 5\vec{b} \tag{7}$$

Graphical Representation

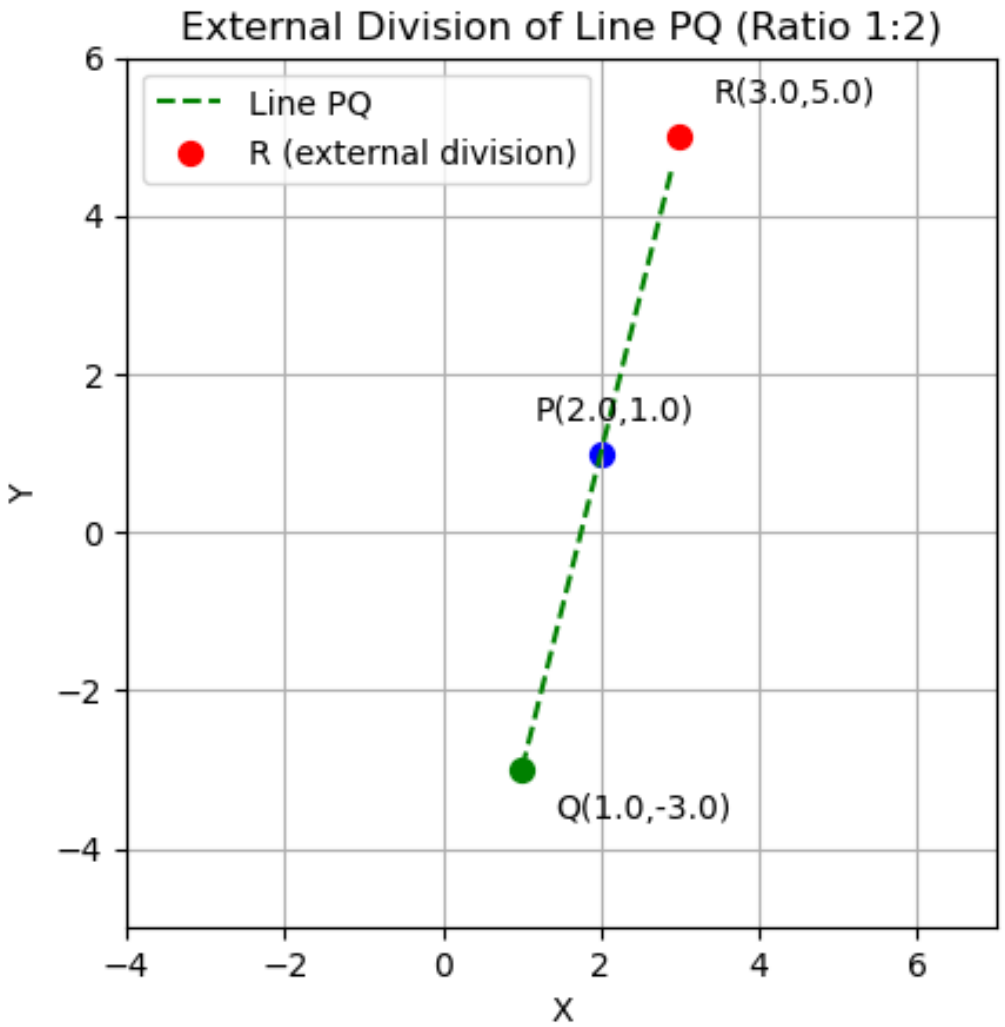

Figure: Graph for Question 2

C Code: section() Function

```c
void section(double* P, double* Q, double* R, int m) {
    for (int i = 0; i < m; i++) {
        R[i] = (Q[i] - 2 * P[i]) / (1 - 2);
    }
}
```

C Code: line_gen() Function

```c
void line_gen(double* X, double* Y, const double* A, const double* B, int n, int m) {
    double temp[2];
    for (int i = 0; i < 2; i++) {
        temp[i] = (B[i] - A[i]) / (double)n;
    }
    for (int i = 0; i <= n; i++) {
        X[i] = A[0] + temp[0] * i;
        Y[i] = A[1] + temp[1] * i;
    }
}
```

Python + C: Load Library

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

handc = ctypes.CDLL("./func.so")

# section function
handc.section.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int
]
handc.section.restype = None

# line_gen function
handc.line_gen.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int,
    ctypes.c_int
]
handc.line_gen.restype = None
```

Python + C: Compute & Plot

```
m = 2
a = np.array([1,0], dtype=np.float64)
b = np.array([0,1], dtype=np.float64)
P = 2*a + b
Q = a - 3*b
R = np.zeros(m, dtype=np.float64)

handc.section(P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
              Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
              R.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
              m)

n = 20
X_l = np.zeros(n, dtype=np.float64)
Y_l = np.zeros(n, dtype=np.float64)
handc.line_gen(X_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
               Y_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
               Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
               R.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
               n, m)

plt.plot(X_l, Y_l, "g--", label="Line PQ")
plt.scatter(P[0], P[1], color="blue", s=50)
plt.scatter(Q[0], Q[1], color="green", s=50)
plt.scatter(R[0], R[1], color="red", s=50, label="R")
plt.show()
```

Pure Python: Functions & Setup

```python
import sys
sys.path.insert(0, '/home/anshu-ram/matgeo/codes/CoordGeo')
import numpy as np
import matplotlib.pyplot as plt

from line.funcs import *
from triangle.funcs import *
from conics.funcs import circ_gen

def section_point(P, Q, m, n, external=True):
    if external:
        return (m*Q - n*P)/(m-n)
    else:
        return (m*Q + n*P)/(m+n)
```

Pure Python: Compute & Plot

```
a = np.array([1,0]).reshape(-1,1)
b = np.array([0,1]).reshape(-1,1)
P = 2*a + b
Q = a - 3*b
R = section_point(P, Q, 1, 2, external=True)

x_PQ = line_gen_num(P, Q, 20)
x_PR = line_gen_num(P, R, 20)
x_QR = line_gen_num(Q, R, 20)

plt.plot(x_PQ[0,:], x_PQ[1,:], "g--", label="Line PQ")
plt.plot(x_PR[0,:], x_PR[1,:], "r--", label="Line PR")
plt.plot(x_QR[0,:], x_QR[1,:], "b--", label="Line QR")
tri_coords = np.hstack((P,Q,R))
plt.scatter(tri_coords[0,:], tri_coords[1,:])
plt.show()
```