Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

August 27, 2025

Problem Statement

Find the coordinates of the point R on the line segment joining P(1,3) and Q(2,5) such that $\mathbf{PR} = \frac{3}{5}\mathbf{PQ}$

Description of Variables used

Input variable	Value
P	$\begin{pmatrix} 1 \\ 3 \end{pmatrix}$
Q	$\begin{pmatrix} 2 \\ 5 \end{pmatrix}$
$\frac{PR}{PQ}$	$\frac{3}{5}$

Table

Theoretical Solution

Let the position vectors be

$$\mathbf{P} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \qquad \mathbf{Q} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}.$$

If $\mathbf{R}$ is the position vector of R, then

$$\mathbf{R} - \mathbf{P} = \frac{3}{5}(\mathbf{Q} - \mathbf{P}) \implies \mathbf{R} = \mathbf{P} + \frac{3}{5}(\mathbf{Q} - \mathbf{P}).$$

So,

$$\mathbf{R} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \frac{3}{5}\left(\begin{pmatrix} 2 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 \\ 3 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \frac{3}{5}\begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Hence,

$$\mathbf{R} = \begin{pmatrix} 1 + \frac{3}{5} \\ 3 + \frac{6}{5} \end{pmatrix} = \begin{pmatrix} \frac{8}{5} \\ \frac{21}{5} \end{pmatrix}.$$

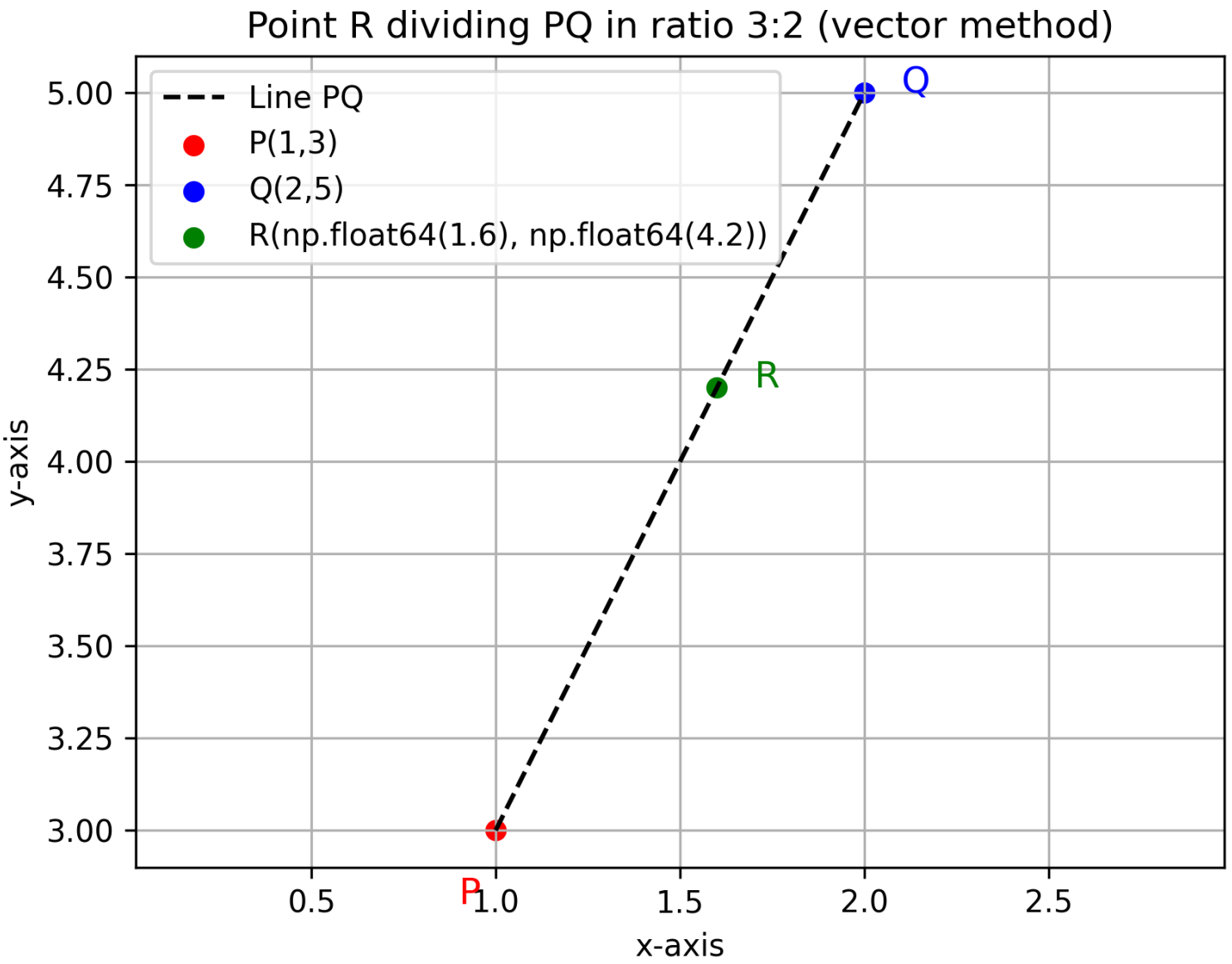
Theoretical Solution

Therefore, the required point is

$$\boxed{\mathbf{R} = \begin{pmatrix} \frac{8}{5} \\ \frac{21}{5} \end{pmatrix}}$$

which indeed satisfies $\mathbf{R} - \mathbf{P} = \frac{3}{5}(\mathbf{Q} - \mathbf{P})$.

Plot


Point R dividing PQ in ratio 3:2 (vector method)
Line PQ
P(1,3)
Q(2,5)
R(np.float64(1.6), np.float64(4.2))
P
Q
R
y-axis
x-axis
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
0.5
1.0
1.5
2.0
2.5

Code - C

The code to find the coordinates of point R is

```
#include <stdio.h>

void point_on_segment2d(const double P[2], const double Q[2], double
    lambda, double R[2]) {
    R[0] = P[0] + lambda * (Q[0] - P[0]);
    R[1] = P[1] + lambda * (Q[1] - P[1]);
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

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

# Load the shared library (Linux name shown here)
lib = ctypes.CDLL("./librpoint.so")

# Tell ctypes the C function signature
lib.point_on_segment2d.argtypes = [
    ctypes.POINTER(ctypes.c_double), # P
    ctypes.POINTER(ctypes.c_double), # Q
    ctypes.c_double, # lambda
    ctypes.POINTER(ctypes.c_double) # R
]
lib.point_on_segment2d.restype = None
```

Code - Python(with shared C code)

```
# Data
P = np.array([1.0, 3.0], dtype=np.float64)
Q = np.array([2.0, 5.0], dtype=np.float64)
lam = 3.0/5.0
R = np.zeros(2, dtype=np.float64)

# Call C
lib.point_on_segment2d(
    P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    lam,
    R.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
)

print("R-from-C:", tuple(R)) # Expect (1.6, 4.2)
```

Code - Python(with shared C code)

```
# Plot
plt.plot([P[0], Q[0]], [P[1], Q[1]], 'k--', label="PQ")
plt.scatter(*P, color='red', label="P")
plt.scatter(*Q, color='blue', label="Q")
plt.scatter(*R, color='green', label="R")
plt.legend()
plt.title("R=P+(3/5)(Q-P)")

# Save first, then show
plt.savefig("/sdcard/ee1030-2025/ai25btech11032/Matgeo/1.4.2/figs/
    PQ_R_plotnew.png", dpi=300)
plt.show()
```

Code - Python only

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

# Define vectors
P = np.array([1, 3])
Q = np.array([2, 5])



lam = 3/5
R = P + lam * (Q - P)
```

Code - Python only

```
# Plot PQ Line
plt.plot([P[0], Q[0]], [P[1], Q[1]], 'k--', label="Line PQ")

# Plot points
plt.scatter(*P, color='red', label="P(1,3)")
plt.scatter(*Q, color='blue', label="Q(2,5)")
plt.scatter(*R, color='green', label=f'R{tuple(R)}")

# Annotate points
plt.text(P[0]-0.1, P[1]-0.2, "P", fontsize=12, color='red')
plt.text(Q[0]+0.1, Q[1], "Q", fontsize=12, color='blue')
plt.text(R[0]+0.1, R[1], "R", fontsize=12, color='green')
```

Code - Python only

```
# Style
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Point R dividing PQ in ratio 3:2 (vector method)")
plt.legend()
plt.grid(True)
plt.axis("equal")

# Save plot to file
plt.savefig("PQ_R_plot.png", dpi=300) # Saves in current folder
# plt.savefig("PQ_R_plot.pdf") # Alternative format

plt.show()
```