1.8.19

EE25BTECH11002 - Achat Parth Kalpesh

September 14,2025

Question

If $\mathbf{Q}(0,1)$ is equidistant from $\mathbf{P}(5,-3)$ and $\mathbf{R}(x,6)$, find the values of x. Also find the distances $\mathbf{QR}$ and $\mathbf{PR}$.

Theoretical Solution

Let the vectors for the given points **P**, **Q** and **R** be

$$\mathbf{P} = \begin{pmatrix} 5 \\ -3 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} x \\ 6 \end{pmatrix} \tag{1}$$

It is given that **Q** is equidistant from **P** and **R**.

$$\|\mathbf{P} - \mathbf{Q}\| = \|\mathbf{R} - \mathbf{Q}\| \tag{2}$$

Squaring both sides,

$$\|\mathbf{P} - \mathbf{Q}\|^2 = \|\mathbf{R} - \mathbf{Q}\|^2 \tag{3}$$

The squared norm of a vector $\mathbf{v}$ is given by $\mathbf{v}^\top \mathbf{v}$.

$$(\mathbf{P} - \mathbf{Q})^\top (\mathbf{P} - \mathbf{Q}) = (\mathbf{R} - \mathbf{Q})^\top (\mathbf{R} - \mathbf{Q}) \tag{4}$$

Theoretical Solution

$$\left(\mathbf{P}^\top - \mathbf{Q}^\top\right)(\mathbf{P} - \mathbf{Q}) = \left(\mathbf{R}^\top - \mathbf{Q}^\top\right)(\mathbf{R} - \mathbf{Q}) \tag{5}$$

$$\mathbf{P}^\top\mathbf{P} - 2\mathbf{Q}^\top\mathbf{P} = \mathbf{R}^\top\mathbf{R} - 2\mathbf{R}^\top\mathbf{Q} \tag{6}$$

$$\begin{pmatrix} 5 & -3 \end{pmatrix}\begin{pmatrix} 5 \\ -3 \end{pmatrix} - 2\begin{pmatrix} 0 & 1 \end{pmatrix}\begin{pmatrix} 5 \\ -3 \end{pmatrix} = \begin{pmatrix} x & 6 \end{pmatrix}\begin{pmatrix} x \\ 6 \end{pmatrix} - 2\begin{pmatrix} x & 6 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{7}$$

$$25 + 9 + 6 = x^2 + 36 - 12 \tag{8}$$

$$x^2 = 16 \tag{9}$$

$$\implies x = \pm 4 \tag{10}$$

Theoretical Solution

Therefore, the two possible vectors for **R** are:

$$\mathbf{R}_1 = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \tag{11}$$

$$\mathbf{R}_2 = \begin{pmatrix} -4 \\ 6 \end{pmatrix} \tag{12}$$

$$\|\mathbf{Q} - \mathbf{R}\| = \|\mathbf{P} - \mathbf{Q}\| = \sqrt{5^2 + (-4)^2} = \sqrt{41} \approx 6.40 \tag{13}$$

Theoretical Solution

- For $\mathbf{R}_1 = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$:

$$\|\mathbf{R_1} - \mathbf{P}\| = \left\| \begin{pmatrix} 4-5 \\ 6-(-3) \end{pmatrix} \right\| = \left\| \begin{pmatrix} -1 \\ 9 \end{pmatrix} \right\| \tag{14}$$

$$= \sqrt{(-1)^2 + 9^2} = \sqrt{82} \approx 9.06 \tag{15}$$

- For $\mathbf{R}_2 = \begin{pmatrix} -4 \\ 6 \end{pmatrix}$:

$$\|\mathbf{R}_2 - \mathbf{P}\| = \left\| \begin{pmatrix} -4-5 \\ 6-(-3) \end{pmatrix} \right\| = \left\| \begin{pmatrix} -9 \\ 9 \end{pmatrix} \right\| \tag{16}$$

$$= \sqrt{(-9)^2 + 9^2} = \sqrt{162} = 9\sqrt{2} \approx 12.73 \tag{17}$$

C code

```c
#include <stdio.h>
#include <math.h>
void formula(double *P, double *Q, double *R){
    double sum1 = 0, sum2 = 0;
    for (int i=0; i<2; i++) {
        sum1 += pow(P[i] - Q[i], 2);
        sum2 += pow(R[i] - Q[i], 2);
    }
    if (sum1 == sum2) {
        printf("Q is equidistant from P and R.");
    }
}
```

Python Code

```
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as LA
import ctypes
import os

# --- Part 1: Mathematical Solution ---
Q_coords = (0, 1)
P_coords = (5, -3)
# From calculation, x = 4 or x = -4
R1_coords = (4, 6)
R2_coords = (-4, 6)

# Calculate distances using numpy
P = np.array(P_coords)
Q = np.array(Q_coords)
R1 = np.array(R1_coords)
R2 = np.array(R2_coords)
```

Python Code

```python
dist_QR1 = LA.norm(R1 - Q)
dist_PR1 = LA.norm(R1 - P)
dist_PR2 = LA.norm(R2 - P)

print(f"Distance QR: {dist_QR1:.2f}")
print(f"Distance PR1 (for x=4): {dist_PR1:.2f}")
print(f"Distance PR2 (for x=-4): {dist_PR2:.2f}\n")

# --- Part 2: C Code Integration using ctypes ---
try:
    # Load the shared library
    geom_lib = ctypes.CDLL('./formula.so')

    # Define the function signature
    c_double_p = ctypes.POINTER(ctypes.c_double)
    geom_lib.formula.argtypes = [c_double_p, c_double_p,
        c_double_p]
    geom_lib.formula.restype = None
```

Python Code

```python
    # Prepare data for C function
    P_c = (ctypes.c_double * 2)(*P)
    Q_c = (ctypes.c_double * 2)(*Q)
    R1_c = (ctypes.c_double * 2)(*R1)
    R2_c = (ctypes.c_double * 2)(*R2)

    # Call the C function for both solutions
    print("Calling C function with R1(4, 6):")
    geom_lib.formula(P_c, Q_c, R1_c)
    print("\nCalling C function with R2(-4, 6):")
    geom_lib.formula(P_c, Q_c, R2_c)

except (OSError, AttributeError) as e:
    print(f"Error loading or using the C library: {e}")
```

Python Code

```python
# --- Part 3: Plotting ---
Q = Q.reshape(-1, 1)
P = P.reshape(-1, 1)
R1 = R1.reshape(-1, 1)
R2 = R2.reshape(-1, 1)

r = LA.norm(P - Q) # Radius is distance QP

# Helper function to generate circle points
def circ_gen(O, r):
    len = 100
    theta = np.linspace(0, 2*np.pi, len)
    x_circ = np.zeros((2, len))
    x_circ[0, :] = r * np.cos(theta)
    x_circ[1, :] = r * np.sin(theta)
    x_circ = (x_circ.T + O.T).T
    return x_circ
```

Python Code

```
    x_circ = circ_gen(Q, r)
x_line = np.linspace(-10, 10, 100)
y_line = np.full_like(x_line, 6)

plt.figure(figsize=(8, 8))
plt.plot(x_circ[0, :], x_circ[1, :], label='Circle centered at Q'
    )
plt.plot(x_line, y_line, label='Line y = 6')

# Plot lines connecting points
plt.plot([P[0,0], R1[0,0]], [P[1,0], R1[1,0]], 'g--', label='Line
     $PR_1$')
plt.plot([P[0,0], R2[0,0]], [P[1,0], R2[1,0]], 'm--', label='Line
     $PR_2$')
```

Python Code

```
# Plot and label the points
points = np.hstack((P, Q, R1, R2))
plt.scatter(points[0, :], points[1, :], s=50, color='red', zorder
    =5)
point_labels = ['P(5,-3)', 'Q(0,1)', 'R1(4,6)', 'R2(-4,6)']

for label, (x, y) in zip(point_labels, points.T):
    plt.annotate(label, (x, y), textcoords="offset points",
    xytext=(0,10), ha='center')

# Plot formatting
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Visual Representation of the Solution")
plt.legend()
plt.grid(True)
plt.axis('equal')
plt.show()
```

Python Plot

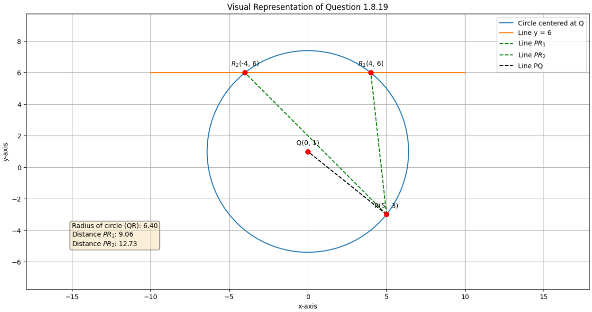

Figure: Visual representation of the solution. The points R_1 and R_2 are the intersections of the circle centered at Q and the line $y = 6$.