4.2.15

BEERAM MADHURI - EE25BTECH11012

September 2025

Three distinct points A, B and C are given in the 2-dimensional coordinate plane such that the ratio of the distance of any one of them from the point $(1, 0)$ to the distance from the point $(-1, 0)$ is equal to $\frac{1}{3}$. Then the circumcentre of the triangle ABC is at the point:

1. $\left(\frac{5}{4}, 0\right)$
2. $\left(\frac{5}{2}, 0\right)$
3. $\left(\frac{5}{3}, 0\right)$
4. $(0, 0)$

let $\mathbf{F_1}$, $\mathbf{F_2}$ be the vectors such that:

Point	Vector
$\mathbf{F_1}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$\mathbf{F_2}$	$\begin{pmatrix} -1 \\ 0 \end{pmatrix}$

Table: Variables used

finding the Circumcenter of the triangle formed by A,B and C

Let P $\begin{pmatrix} x \\ y \end{pmatrix}$ be any point in the plane of A,B,C.

given,

$$\frac{\|\mathbf{PF_1}\|}{\|\mathbf{PF_2}\|} = \frac{1}{3} \tag{1}$$

$$\frac{\sqrt{(\mathbf{P} - \mathbf{F_1})^\top(\mathbf{P} - \mathbf{F_1})}}{\sqrt{(\mathbf{P} - \mathbf{F_2})^\top(\mathbf{P} - \mathbf{F_2})}} = \frac{1}{3} \tag{2}$$

By Substituting Values and Simplifying:

$$2x^2 + 2y^2 - 5x + 2 = 0 \tag{3}$$

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \frac{5}{4} \\ 0 \end{pmatrix} \right\| = \sqrt{\frac{21}{4}} \tag{4}$$

On comparing the equation with general form:-

$$||\mathbf{P} - \mathbf{C}|| = r \qquad (5)$$

$$\text{where , } \mathbf{P} = \text{any point on the circle} \qquad (6)$$

$$\mathbf{C} = \text{Center of circle} \qquad (7)$$

$$r = \text{radius of circle} \qquad (8)$$

$$\text{Center of circle} = \left(\frac{5}{4}, 0\right) \qquad (9)$$

Hence, the circumcenter of the triangle is $\left(\frac{5}{4}, 0\right)$

Python Code

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

# --- 1. Define the geometric elements ---

# The two fixed points from the problem
f1 = np.array([1, 0])
f2 = np.array([-1, 0])
```

Python Code

```python
# The locus of points P such that dist(P, F1) / dist(P, F2) = 1/3
    is a circle.
# By solving the distance equation, we find the circle's
    properties:
# 9 * ((x-1)^2 + y^2) = (x+1)^2 + y^2
# 8x^2 - 20x + 8y^2 + 8 = 0
# x^2 - (5/2)x + y^2 + 1 = 0
# (x - 5/4)^2 + y^2 = (3/4)^2
# The center of this circle is the circumcenter of triangle ABC.
circumcenter = np.array([5/4, 0])
radius = 3/4
```

Python Code

```
theta = np.linspace(0, 2 * np.pi, 200)
circle_x = circumcenter[0] + radius * np.cos(theta)
circle_y = circumcenter[1] + radius * np.sin(theta)
angles = np.array([np.pi/4, 5*np.pi/6, 3*np.pi/2])
A = circumcenter + radius * np.array([np.cos(angles[0]), np.sin(
    angles[0])])
B = circumcenter + radius * np.array([np.cos(angles[1]), np.sin(
    angles[1])])
C = circumcenter + radius * np.array([np.cos(angles[2]), np.sin(
    angles[2])])
triangle_points = np.array([A, B, C, A]) # Add A at the end to
    close the triangle
```

Python Code

```python
# --- 3. Create the plot ---

# Set up the plot style and figure
plt.style.use('seaborn-v0_8-whitegrid')
fig, ax = plt.subplots(figsize=(10, 8))

# Plot the circumcircle
ax.plot(circle_x, circle_y, 'b-', label='Locus of A, B, C (
    Circumcircle)')
```

Python Code

```python
# Plot the example triangle ABC
ax.plot(triangle_points[:, 0], triangle_points[:, 1], 'g--',
    marker='o', markersize=8, label='Example Triangle ABC')
# Plot the two fixed points
ax.plot(f1[0], f1[1], 'ro', markersize=10, label='Fixed Point F1
    (1, 0)')
ax.plot(f2[0], f2[1], 'mo', markersize=10, label='Fixed Point F2
    (-1, 0)')
# Plot and highlight the circumcenter
ax.plot(circumcenter[0], circumcenter[1], 'k*', markersize=15,
    label=f'Circumcenter ({circumcenter[0]}, {circumcenter[1]})')
```

Python Code

```python
# --- 4. Add labels and formatting ---

# Annotate the points on the graph
ax.text(A[0], A[1] + 0.1, 'A', fontsize=14, color='green')
ax.text(B[0] - 0.15, B[1], 'B', fontsize=14, color='green')
ax.text(C[0], C[1] - 0.15, 'C', fontsize=14, color='green')
ax.text(circumcenter[0], circumcenter[1] + 0.05, 'Circumcenter',
    fontsize=12, ha='center')
```

Python Code

```python
# Set plot title and axis labels
ax.set_title('Solution for Finding the Circumcenter', fontsize
    =16)
ax.set_xlabel('X-axis', fontsize=12)
ax.set_ylabel('Y-axis', fontsize=12)

# Ensure the aspect ratio is equal to prevent the circle from
    looking like an ellipse
ax.set_aspect('equal', adjustable='box')
ax.axhline(0, color='black', linewidth=0.5)
ax.axvline(0, color='black', linewidth=0.5)
```

Python Code

```python
# Add legend and grid
ax.legend(loc='upper right')
ax.grid(True)

# Set axis limits for a clean view
ax.set_xlim(-2, 3)
ax.set_ylim(-1.5, 1.5)

# Display the final plot
plt.show()
```

C Code

```c
#include <stdio.h>
#include <math.h>

struct Point {
    double x;
    double y;
};
struct Point findCircumcenter(struct Point p1, struct Point p2,
    double ratio) {
    struct Point center;
    double k = ratio;
    double k_squared = k * k;
```

C Code

```c
// Formula for the center of the Circle of Apollonius
// Center (h, k) = ( (x1 - k^2*x2) / (1 - k^2), (y1 - k^2*y2) /
    (1 - k^2) )
center.x = (p1.x - k_squared * p2.x) / (1 - k_squared);
center.y = (p1.y - k_squared * p2.y) / (1 - k_squared);
return center;
}

int main() {
// Define the two fixed points from the problem
struct Point p1 = {1.0, 0.0}; // Point (1, 0)
struct Point p2 = {-1.0, 0.0}; // Point (-1, 0)
```

C Code

```
// Define the given ratio
double ratio = 1.0 / 3.0;

// Calculate the circumcenter
struct Point circumcenter = findCircumcenter(p1, p2, ratio);
printf("The circumcenter of the triangle ABC is at the point:
    (%.2f, %.2f)\n",
circumcenter.x, circumcenter.y);
printf("In fraction form, this is (5/4, 0).\n");
return 0;
}
```

Python and C Code

```python
import ctypes

# Define a structure equivalent to C struct Point
class Point(ctypes.Structure):
    _fields_ = [("x", ctypes.c_double),
                ("y", ctypes.c_double)]

def find_circumcenter(p1: Point, p2: Point, ratio: float) ->
    Point:
    center = Point()
    k = ratio
    k_squared = k * k
```

```python
    # Apply the same formula as in the C code
    center.x = (p1.x - k_squared * p2.x) / (1 - k_squared)
    center.y = (p1.y - k_squared * p2.y) / (1 - k_squared)
    return center
if __name__ == "__main__":
    # Define the two points
    p1 = Point(1.0, 0.0)
    p2 = Point(-1.0, 0.0)
```

Python and C Code

```
# Define the ratio
ratio = 1.0 / 3.0

# Compute the circumcenter
circumcenter = find_circumcenter(p1, p2, ratio)

# Print the result
print(f"The circumcenter of the triangle ABC is at the point:
        ({circumcenter.x:.2f}, {circumcenter.y:.2f})")
print("In fraction form, this is (5/4, 0).")
```
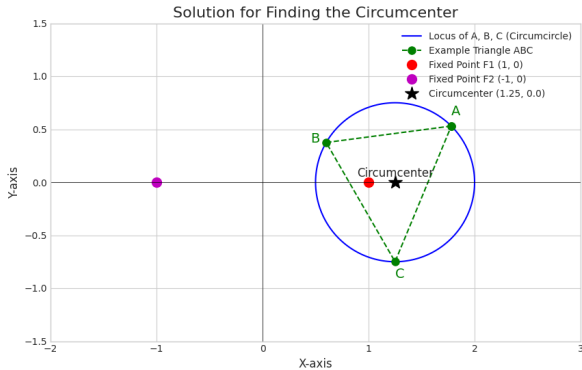
Figure: Plot