Presentation

Sri Sathwik Desaboina
AI24BTECH11007

November 6, 2024

1. Problem
2. Solution
   - Usage of variables
   - Parametric form
   - Matrix equation
   - Verification
3. Codes
   - C code to verify the Area
   - Plotting the figure using Python
   - Plot

Problem Statement

Show that the point $\begin{pmatrix} x \\ y \end{pmatrix}$ given by $x = \frac{2at}{1+t^2}$ and $y = \frac{a(1-t^2)}{1+t^2}$ lies on a circle for all real values of t such that $-1 \leq t \leq 1$, where a is any given real number.

Usage of variables

S.No	variables used	description
1	t	a variable which takes the real values in the range $(-1, 1)$
2	a	it is a fixed real number
3	$\mathbf{A}(\mathbf{t})$	it is a transformation matrix of parameter t
4	$\mathbf{v}(\mathbf{t})$	it represent the parameter t and allows to define x and y
5	$\mathbf{p}(\mathbf{t})$	a point with coordinates x and y.

Parametric form

Given x and y in the parametric form,

$$x = \frac{2at}{1+t^2}, \tag{3.1}$$

$$y = \frac{a(1-t^2)}{1+t^2} \tag{3.2}$$

Let $\mathbf{p(t)}$ be equal to,

$$\mathbf{p}(t) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{2at}{1+t^2} \\ \frac{a(1-t^2)}{1+t^2} \end{pmatrix}. \tag{3.3}$$

Matrix equation

The transformation matrix $\mathbf{A}(\mathbf{t})$ and $\mathbf{v}(\mathbf{t})$ with parameter t are,

$$\implies \mathbf{A}(t) = \begin{pmatrix} \frac{2a}{1+t^2} & 0 \\ 0 & \frac{a(1-t^2)}{1+t^2} \end{pmatrix}, \tag{3.4}$$

$$\implies \mathbf{v}(\mathbf{t}) = \begin{pmatrix} t \\ 1 \end{pmatrix}, \tag{3.5}$$

$$\mathbf{p}(\mathbf{t}) = \mathbf{A}(\mathbf{t})\mathbf{v}(\mathbf{t}), \tag{3.6}$$

$$\implies \mathbf{p}(t) = \begin{pmatrix} \frac{2a}{1+t^2} & 0 \\ 0 & \frac{a(1-t^2)}{1+t^2} \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}, \tag{3.7}$$

$$\tag{3.8}$$

Verification

Now, if we check the value of ,

$$\mathbf{p}(t)^{\top}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\mathbf{p}(t) \tag{3.9}$$

We get,

$$\mathbf{p}(t)^{\top}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\mathbf{p}(t) = a^2 \tag{3.10}$$

$\implies$ We proved that the given points lie on a circle $x^2 + y^2 = a^2$.
Since we have the values of t in $(-1, 1)$, the y-coordinate of the points is always positive.
We get a semi-circle with those points.

C code to generate points I

```c
#include <stdio.h>

int main() {
    // Declare a pointer to the file
    FILE *file;

    // Open the file points.dat for writing (will create it if it doesn't exist)
    file = fopen("points.dat", "w");

    // Check if the file was opened successfully
    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;  // Return 1 if there was an error
    }

    // Write the origin point O to the file (x=0, y=0)
    fprintf(file, "0.00000 0.00000\n");

    // Define the number of points and the range of the parameter t
```

C code to generate points II

```c
    int num_points = 100;  // Number of points to generate
    double t_start = -1.0, t_end = 1.0;  // Range of t (from -1 to 1)
    double t_increment = (t_end - t_start) / (num_points - 1);  // Step size for t

    // Loop to calculate points and write them to the file
    for (int i = 0; i < num_points; i++) {
        // Calculate t value for the current point
        double t = t_start + i * t_increment;

        // Parametric equations to calculate x and y
        double x = (2 * t) / (1 + t * t);  // Equation for x
        double y = (1 - t * t) / (1 + t * t);  // Equation for y

        // Write the calculated point (x, y) to the file with 5 decimal precision
        fprintf(file, "%.5f %.5f\n", x, y);
    }

    // Close the file
```

C code to generate points III

```c
    fclose(file);

    // Print success message
    printf("Data written to points.dat successfully.\n");

    return 0;
}



```

Plotting the figure using Python I

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

# Load the points from the file (now space-separated)
data = np.loadtxt("points.dat", delimiter=" ")

# Extract the center point O from the data file (first point)
center_x = data[0, 0]  # Should be 0.0
center_y = data[0, 1]  # Should be 0.0

# Separate the circle points and transformed points
circle_points = data[1:101]  # Assuming first 100 points are from the circle
transformed_points = data[101:]  # Remaining points are transformed points

# Create a plot with minimized dimensions
plt.figure(figsize=(5, 4))  # Changed dimensions to 5x4 inches

# Plot the circle points
```

Plotting the figure using Python II

```python
plt.plot(circle_points[:, 0], circle_points[:, 1], label='Circle Points', color='blue', linestyle='-', marker='o', markersize=4)

# Plot the transformed points (without a label)
plt.plot(transformed_points[:, 0], transformed_points[:, 1], color='red', linestyle='-', marker='o', markersize=4)

# Set the plot title and labels
plt.title('Circle Points')  # Removed transformed points from the title
plt.xlabel('x')
plt.ylabel('y')
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid()
plt.axis('equal')
plt.legend()

# Save the plot as a PNG file
plt.savefig("plot.png", dpi=300)  # Save with 300 dpi for better quality

```

Plotting the figure using Python III

```python
# Optional: Show the plot
plt.show()
```

Plot

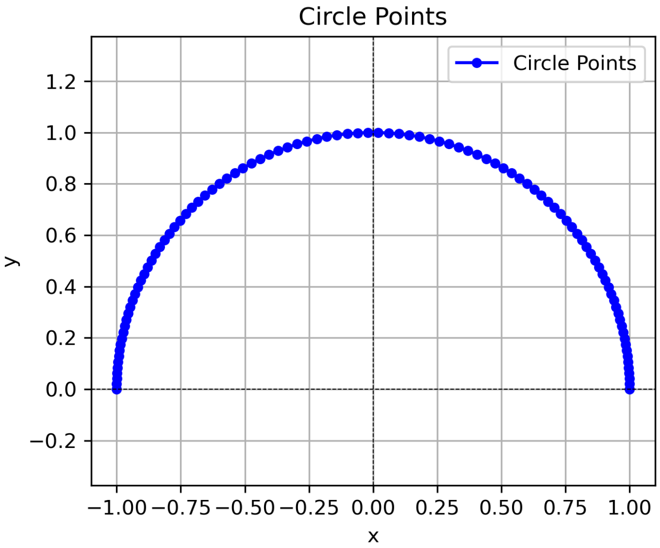

Figure: Circle Points