Trigonometric Functions

Drawing a Circle Using Trigonometric Functions

When I wanted to create a smoother and more precise circle, I decided to generate it using trigonometric functions. This approach allowed me to calculate the circle's points directly, ensuring a seamless curve.

I started by defining the circle's radius, center, and the number of points I wanted along its perimeter. By using 1,000 points, I ensured that the circle would appear smooth and continuous. To generate these points, I relied on trigonometric functions—cosine for the x-coordinates and sine for the y-coordinates. I evenly spaced the angles between 0 and 2π2\pi2π, creating a complete revolution around the center.

After calculating the coordinates, I plotted the circle using MATLAB's `plot` function. To maintain its proper shape, I set the axis scaling to be equal, ensuring the circle didn't appear distorted. The result was a clean, smooth circle that looked natural and precise.

This method showcased the power of trigonometric functions for generating geometric shapes.

```
% Drawing a Smooth Circle Using Trigonometric Functions
% I want to create a smoother circle by generating points with trigonometric functions.

% Define the number of points, radius, and center
n = 1000; % Number of points
r = 2; % Circle radius
c = [3 3]; % Circle center [x, y]

% Generate the circle coordinates
t = linspace(0, 2 * pi, n); % Angle values
x = c(1) + r * cos(t); % x-coordinates
y = c(2) + r * sin(t); % y-coordinates

% Plot the circle
figure;
plot(x, y, 'r', 'LineWidth', 2); % Draw the circle as a line
axis equal; % Ensure the aspect ratio is correct
title('Circle Using Trigonometric Functions');
```