BINARY SOLUTIONS

MAT 275
Polar to Cartesian Conversion

contact@thebinarysolutionsllc.com

Introduction

This script aims to perform the conversion from polar coordinates to Cartesian coordinates. It generates Cartesian coordinates (x, y) based on given polar angles (theta) and a constant radius (r).

Problem Statement

The objective is to create a MATLAB script that calculates Cartesian coordinates from polar coordinates using the provided angles and a constant radius.

Solution Steps

- **Define Polar Coordinates:**
  - Create a row vector theta representing different angles in radians

    $$(‘0, pi/4, pi/2, 3 * pi/4, pi, 5 * pi/4’).$$

  - Set an initial radius value ‘r’ to ‘2’.
- **Calculate Cartesian Coordinates:**
  - Use the polar-to-Cartesian conversion formulas:

    $$x = r * \cos(theta)$$

    $$y = r * \sin(theta)$$

    to obtain the corresponding x and y values.

contact@thebinarysolutionsllc.com

Solution Steps

- **Calculate Radius from Cartesian Coordinates:**
  - Calculate the radius using the formula:

$$radius = sqrt(x.\text{^}2 + y.\text{^}2)$$

to verify the correctness of the Cartesian coordinates.

Pseudo Code

1. Clear the console and workspace
2. Define the array of angles
   - theta = [0, pi/4, pi/2, 3*pi/4, pi, 5*pi/4]
3. Set the initial radius value
   - r = 2
4. For each angle in theta do:
   - 4.1 Calculate the x-coordinate:
     - x = r * cos(angle)
   - 4.2 Calculate the y-coordinate:
     - y = r * sin(angle)
   - 4.3 Calculate the radius:
     - radius = sqrt(x^2 + y^2)
   - 4.4 Output the value of radius
5. End loop.

contact@thebinarysolutionsllc.com

MATLAB Code

```
clc

clear all

theta=[0 pi/4 pi/2 3*pi/4 pi 5*pi/4]; % angles row vector

r=2;    % radius initial value

x=r*cos(theta); % defining x

y=r*sin(theta); % defining y

radius=sqrt(x.^2+y.^2)  % calculating radius from x and y
```

Code Explanation

- **clc; clear all**
  - Clears the command window.
  - Clears all variables from the MATLAB workspace.
- **Theta**
  - Defines a row vector containing angles [0, pi/4, pi/2, 3*pi/4, pi, 5*pi/4].
- **r**

  Sets the initial value for the radius to 2.
- **x**

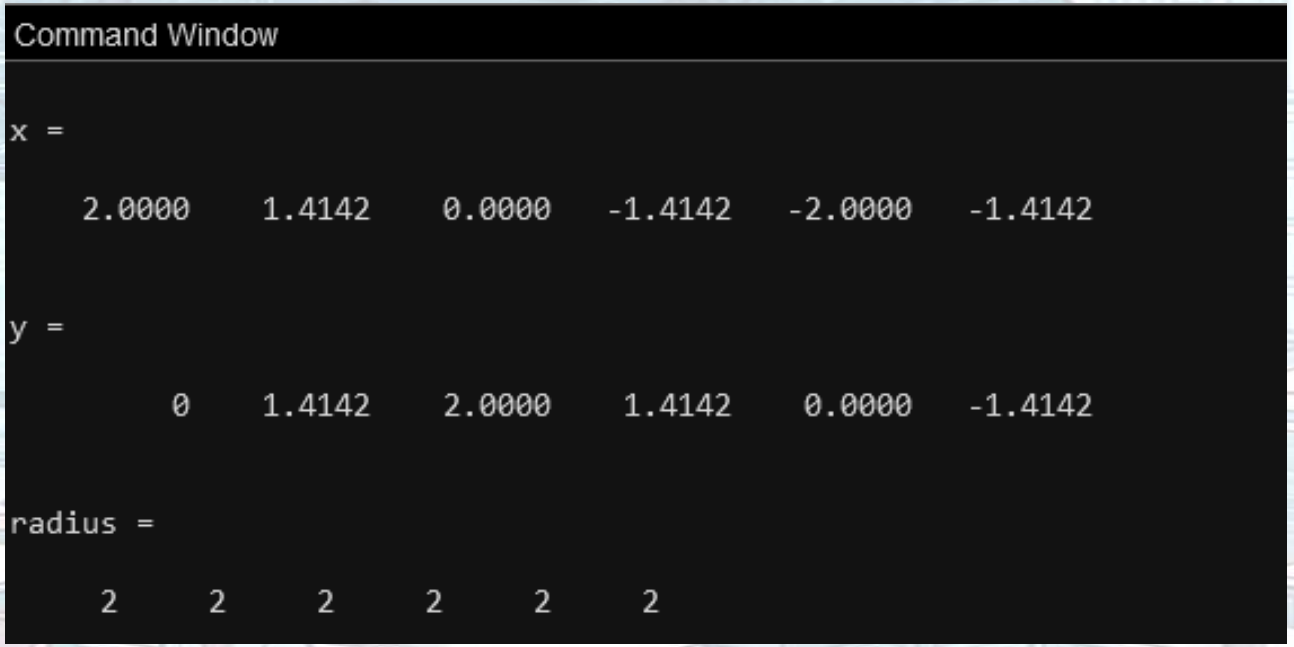
  Calculates the x-coordinates based on the formula r * cos(theta).
- **y**

  Calculates the y-coordinates based on the formula r * sin(theta).
- **radius**

  Calculates the radius using the formula sqrt(x.^2 + y.^2), which is the Euclidean distance from the origin (0,0) to each point (x, y).

Final Answer

The final output is the radius vector containing the calculated ‘radius’ values from Cartesian coordinates.



contact@thebinarysolutionsllc.com

Additional Comments/Tips

The script is concise and efficiently achieves the goal of polar-to-Cartesian conversion.

Conclusion

The script successfully converts polar coordinates to Cartesian coordinates and verifies the result by calculating the radius.