Turtle Tkinter Matplotlib Python Imaging Library Pyglet Python Numpy Pandas Python Database

Rhodonea Curves and Maurer Rose in Python

Last Updated : 30 Jun, 2021

In this article, we will create a Rhodonea Curve and Maurer Rose pattern in Python! Before we proceed to look at what exactly is a *rhodonea curve* or *maurer rose* we need to get the basic structure of our program ready!

Basic Structure of the Program –

Before we move on to learn anything about Rhodonea curves or Maurer rose patterns, we first need to get the basic structure of our program ready. So that when we refactor our code we will only modify a function while the rest of the program will be the same. So here’s the basic structure of the program –

Python3

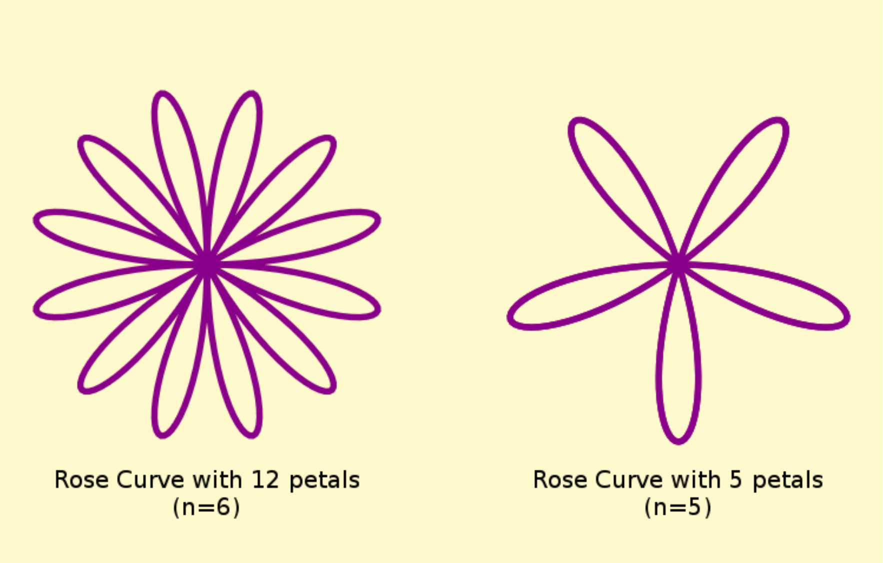
```
from math import sin
from math import cos
from math import radians

import pygame

# The width and height of our program
(width, height) = (800, 600)

# Setup the window and init the screen surface
screen = pygame.display.set_mode((width, height))

pygame.display.set_caption('Rose Curve in Python !')

# The background color of the screen
screen.fill((250, 250, 205)) # lemonChiffon color

# Our function for drawing the rose pattern
drawPattern()
```

```
# Flip the drawn canvas with the newly created canvas (double-buffering)
# Basically we're refreshing the screen surface after drawing
pygame.display.flip()

# Our Main Loop
while True :

    # Poll the events in the event queue
    for event in pygame.event.get() :

        # if the user closed the window
        if event.type == pygame.QUIT :

            # deactivate pygame and quit the program
            pygame.quit()
            quit()

        # Draws the surface object to the screen.
        pygame.display.update()
```

If you run the code, you will get an error that drawPattern is not defined. We'll define it later. But for now just let it be and let's move on to understand what are **Rhodonea curves**!

What is a Rhodonea Curve?

Rhodonea Curve (also known as the *rose* in mathematics) is basically a curve which can be expressed in the form $r = cos(n\theta)$where $n$is an integer defining the number of petals ($2n$if $n$is even and $n$when odd)
Confused ? See this image comparing a rose with 12 petals versus a rose with 5 petals –



Rose Curve with 12 petals
(n=6)

Rose Curve with 5 petals
(n=5)

Note that the coordinates in the equation above are in polar form. But since in computer graphics we express pixel locations in cartesian form rather than polar form, the curve has to re-described as a set of all points (x, y)each in parametric equation form as shown below:

$x = rcos(\theta), y = rsin(\theta)$

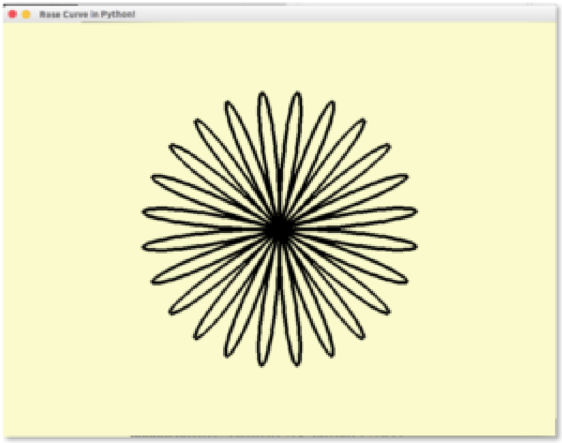
Getting our hands dirty with Rose Curves

The above was just the theory and in case you didn't understand the concept completely you will after you *see it in action*. Just add the following in line 17 (right above where we invoked drawPattern):

Python3

```python
# Draws a rose with n petals and of radius size about `size`
def drawRhodoneaCurve(n, size):
    points =[]
    for i in range(0, 361):
        # The equation of a rhodonea curve
        r = size * sin(radians(n * i))

        # Converting to cartesian co-ordinates
        x = r * cos(radians(i))
        y = r * sin(radians(i))

        list.append(points, (width / 2 + x, height / 2 + y))
    # Draws a set of line segments connected by set of vertices points
    # Also don't close the path and draw it black and set the width to 5
    pygame.draw.lines(screen, (0, 0, 0), False, points, 5)

def drawPattern():
    # Try changing these values to what you want
    drawRhodoneaCurve(12, 200)
```

This will produce the following result –



Explanation

- We calculate the vertices (each significant point on the rose which are later connected by *edges* which are simply line-segments) using the formula we discussed in the above section
- We store a list called points which has all the vertices of the rose and later this list of vertices is feeded to pygame.draw.lines which draws a sequence of continuous straight lines and here we use it to draw polygonal chain (since a rose is just a polygonal chain)
- Note that we are using radians method here. This is because the angles are in degree format (we could have directly used radians but range expects only integers, also for maurer curves we'll anyways have to use degrees so why not start from now) – hopefully we didn't have to do any conversion ourselves as Python3 provides built-in functions degrees and radians.

- Also note that we are shifting the coordinates by (width/2, height/2). This is because by default PyGame has its coordinates in Top-Left corner. But we want it in the center so we shift it by half the screen width and height. We could have translated the whole coordinate system but it's easier this way!

A note about making the program more interactive:
The program currently isn't that interactive but you can make it interactive by perhaps increasing the number of petals by a fractal amount every frame and depending on how fast/slow you increment you can get pretty awesome results. Making the program interactive is out of scope (since that'd' significantly increase the LOC count and perhaps also increase the complexity of the program and violate the basic structure that we created in the beginning) but I experimented a bit and here's the result of first creating a variable n and then incrementing it by 0.1 at every frame!

Now What is a Maurer Rose?

Now that we know about Rhodonea Curves let's proceed to Maurer Rose:-
A Maurer rose is a rhodonea curve of the form $r = cos(n\theta)$consisting of 361 lines connected by 361 points – each of the form $(sin(nk), k)$for every $k$in the set $\{0, d, 2d, 3d, ..., 360d\}$, where $d$is an integer.
Varying values of $d$can create varying types of roses even if they all have the same number of petals (meaning same value of n)
Now the cartesian coordinates will be of the form:-$x = rcos(k), y = rsin(k)$

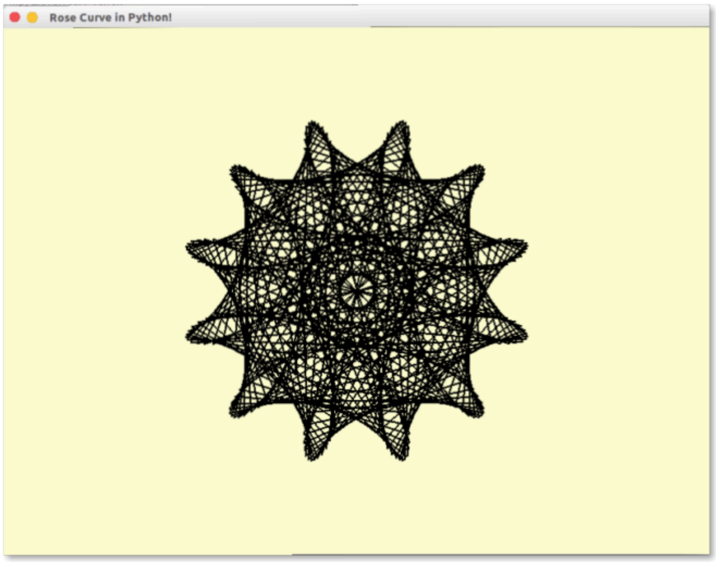
Getting our hands dirty with Maurer Roses

So to make something out of what we just learned, we over-write the previous drawPattern function with the new one:-

Python3

```
# Draws a maurer rose with value n and d it's size about `size`
def drawMaurerRose(n, d, size):
    points =[]
    for i in range(0, 361):
        # The equation of a maurer rose
        k = i * d
        r = size * sin(radians(n * k))

        # Converting to cartesian co-ordinates
        x = r * cos(radians(k))
        y = r * sin(radians(k))

        list.append(points, (width / 2 + x, height / 2 + y))

    # Draws a set of line segments connected by set of vertices points
    # Also don't close the path and draw it black and set the width to 5
    pygame.draw.lines(screen, (0, 0, 0), False, points, 5)

def drawPattern():
    # Try changing these values to what you want
    drawMaurerRose(6, 79, 200)
```

And now when you run the program you should get the following result –



Explanation:

Only the highlighted lines have been changed rest everything is same –

- The function name has been changed to drawMaurerRose from drawRhodoneaCurve and now expects an extra parameter d!
- We now introduce a new variable k which actually comes from the formula. It's value at each iteration is equal to i*d, so for d=1 the function is exactly the same as drawRhodoneaCurve!!
- We now use k instead of i and it should be noted that the line width has been changed to 2 from 5 (otherwise some lines would appear as one because of their high thickness).
- And the invoke statement has ofcourse been changed as well, it now calls the newly created drawMaurerRose function with n=6 and d=71. The values were copied from WikiPedia since random value of d can sometimes produce weird looking roses!

References:

https://en.wikipedia.org/wiki/Rose_(mathematics)
https://www.geeksforgeeks.org/degrees-and-radians-in-python/
https://en.wikipedia.org/wiki/Maurer_rose
http://mathworld.wolfram.com/MaurerRose.html

B youn... 3

Previous Article Next Article

Snowfall display using Pygame in Python Creating start Menu in Pygame

Similar Reads

How to make Wind Rose and Polar Bar Charts in Plotly - Python?

A Plotly is a Python library that is used to design graphs, especially interactive graphs. It can plot various graphs and charts like histogram, barplot, boxplot,...

2 min read

Make filled polygons between two curves in Python using Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays a...

1 min read

Make filled polygons between two horizontal curves in Python using…

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays a...

2 min read

Plotting polar curves in Python

A point in polar co-ordinates is represented as (r, theta). Here, r is its distance from the origin and theta is the angle at which r has to be measured from origin....

4 min read

How to plot Andrews curves using Pandas in Python?

Andrews curves are used for visualizing high-dimensional data by mapping each observation onto a function. It preserves means, distance, and variances. It is...

2 min read

Important differences between Python 2.x and Python 3.x with examples

In this article, we will see some important differences between Python 2.x and Python 3.x with the help of some examples. Differences between Python 2.x an...

5 min read

Creating and updating PowerPoint Presentations in Python using python -…

python-pptx is library used to create/edit a PowerPoint (.pptx) files. This won't work on MS office 2003 and previous versions. We can add shapes, paragraphs,...

4 min read

Learn DSA with Python | Python Data Structures and Algorithms

This tutorial is a beginner-friendly guide for learning data structures and algorithms using Python. In this article, we will discuss the in-built data...

15+ min read

Setting up ROS with Python 3 and Python OpenCV

Setting up a Robot Operating System (ROS) with Python 3 and OpenCV can be a powerful combination for robotics development, enabling you to leverage ROS's...

3 min read

Reading Python File-Like Objects from C | Python

Writing C extension code that consumes data from any Python file-like object (e.g., normal files, StringIO objects, etc.). read() method has to be repeatedly...

3 min read

Article Tags : Python Technical Scripter Python-PyGame Technical Scripter 2019

Practice Tags : python

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design
High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation
Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects
Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos
DSA
Python
Java
C++
Web Development
Data Science
CS Subjects