Vamshi Jandhyala

Cozy Circles in Regular Polygons

April 2025

A nice puzzle by Xavier Durawa for the week of 4/20.

Puzzle

Start with a regular triangle. At the midpoint of each side draw a circle on the inside of the triangle, tangent to the side. Each circle is the same size and its radius is maximised so that the interior circles touch each other. What is the ratio of the area of the three interior circles to the area of the triangle? How about a square? How about a regular n-gon?

Solution

Let the regular polygon have n sides of length s, and let r be the common radius of the circles. Consider two adjacent sides; let C be their common vertex, F and D the points of tangency of the circles with those two sides, and G and H the corresponding circle centres. The internal angle at C is

$$\angle C \;=\; \frac{(n-2)\pi}{n}.$$

The length $|FD|$ is

$$|FD| \;=\; 2\left(\tfrac{s}{2}\right)\cos\left(\frac{\pi-\frac{(n-2)\pi}{n}}{2}\right) \;=\; s\cos\left(\tfrac{\pi}{n}\right).$$

The angles $\angle GFD = \angle HDF$ are

$$\angle GFD \;=\; \frac{\pi}{2}-\frac{\pi}{n} \;=\; \frac{\pi(n-2)}{2n}.$$

Since $|GF| = |DH| = r$ and $|GH| = 2r$, the projection condition $|FD| = 2r\cos(\angle GFD) + |GH|$ gives

$$s\cos\left(\tfrac{\pi}{n}\right) \;=\; 2r\left(\cos\left(\tfrac{\pi(n-2)}{2n}\right)+1\right),$$

so

$$r \;=\; \frac{s\cos(\pi/n)}{2\,(\sin(\pi/n)+1)}.$$

The area of an n-gon of side s is $(ns^2/4)\cot(\pi/n)$, and the total area of the n circles is

$$n\pi\left(\frac{s\cos(\pi/n)}{2\,(\sin(\pi/n)+1)}\right)^2.$$

The required ratio is therefore

$$\boxed{\frac{\pi}{2}\cdot\frac{\sin(2\pi/n)}{(\sin(\pi/n)+1)^2}.}$$

Triangle $(n=3)$.

$$\frac{\pi}{2}\cdot\frac{\sqrt{3}/2}{(1+\sqrt{3}/2)^2} = \frac{\pi\sqrt{3}}{7+4\sqrt{3}}.$$

Square $(n=4)$.

$$\frac{\pi}{2}\cdot\frac{1}{(1+1/\sqrt{2})^2} = \frac{\pi}{3+2\sqrt{2}}.$$

Python code

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon, Circle
import math

def draw_regular_polygon_with_circles(n,
        polygon_edge_color='black',
        polygon_fill_color='none',
        circle_color='blue',
        show_figure=True):
    fig = plt.figure(figsize=(8, 8))
    ax = fig.add_subplot(111)

    # Circumradius for unit-side polygon
    R = 1 / (2 * math.sin(math.pi / n))

    # Polygon vertices
    vertices = [(R * math.cos(2 * math.pi * i / n),
                 R * math.sin(2 * math.pi * i / n))
                for i in range(n)]

    polygon = Polygon(vertices, closed=True,
                      edgecolor=polygon_edge_color,
                      facecolor=polygon_fill_color)
    ax.add_patch(polygon)

    circle_radius = math.cos(math.pi / n) / (2 * (1 + math.sin(math.pi / n)))

    for i in range(n):
        j = (i + 1) % n
        midpoint_x = (vertices[i][0] + vertices[j][0]) / 2
        midpoint_y = (vertices[i][1] + vertices[j][1]) / 2
        dx = vertices[j][0] - vertices[i][0]
```

```
        dy = vertices[j][1] - vertices[i][1]
        normal_x, normal_y = -dy, dx
        nlen = math.sqrt(normal_x**2 + normal_y**2)
        normal_x /= nlen
        normal_y /= nlen
        # Ensure normal points inward
        if normal_x * midpoint_x + normal_y * midpoint_y > 0:
            normal_x, normal_y = -normal_x, -normal_y

        cx = midpoint_x + normal_x * circle_radius
        cy = midpoint_y + normal_y * circle_radius
        ax.add_patch(Circle((cx, cy), circle_radius,
                            fill=False, edgecolor=circle_color))

    ax.set_aspect('equal')
    margin = R * 1.2
    ax.set_xlim(-margin, margin)
    ax.set_ylim(-margin, margin)
    ax.set_axis_off()
    if show_figure:
        plt.tight_layout()
        plt.show()
    return fig
```