Updated on February 3, 2021

Math 3607: Exam 1 (Written)

Due: 11:59PM, Friday, February 5, 2021

1 Spiral Triangle to Spiral Polygon [25 points]

Recall that the following script generates spirals using equilateral triangles as shown in the figure below. Note the introduction of a new variable `d_rot`, which is accountable for the rotation of the innermost triangle.

```
m = 21; d_angle = 4.5; d_rot = 90;
th = linspace(0, 360, 4) + d_rot;
V = [cosd(th);
     sind(th)];
C = colormap(hsv(m));
s = sind(150 - abs(d_angle))/sind(30);
R = [cosd(d_angle) -sind(d_angle);
     sind(d_angle) cosd(d_angle)];
hold off
for i = 1:m
    if i > 1
        V = s*R*V;
    end
    plot(V(1,:), V(2,:), 'Color', C(i,:))
    hold on
end
set(gcf, 'Color', 'w')
axis equal, axis off
```

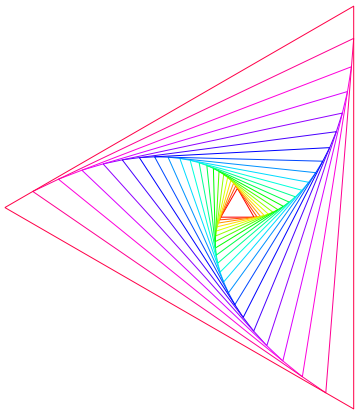

Figure 1: A spiral triangle with $m = 21$ and $\theta = 4.5°$.

(a) Modify and generalize the script so that it generates spirals using m regular n-gons for any $n \geq 3$. Then turn the script into a function m-file `spiralgon.m`.

```
function V = spiralgon(n, m, d_angle, d_rot)
% SPRIALGON plots spiraling regular n-gons
% input:   n = the number of vertices
%          m = the number of regular n-gons
%          d_angle = the degree angle between successive n-gons
%          (can be positive or negative)
%          d_rot = the degree angle by which the innermost n-gon is
   rotated
% output:  V = the vertices of the outermost n-gon
....
```

(b) Run the statements below to generate some aesthetic shapes.

```
clf
subplot(2, 2, 1), spiralgon(3, 41, 4.5, -90);
subplot(2, 2, 2), spiralgon(4, 37, -2.5, 45);
subplot(2, 2, 3), spiralgon(5, 61, 3, -90);
subplot(2, 2, 4), spiralgon(8, 91, -4, 22.5);
```

2 Doorkeepers [25 points]

Let $n \in \mathbb{N}$. There are n doors and n doorkeepers, which are indexed by integers from 1 to n. All doors are initially closed. Then the doorkeepers, one after another, do the following:

- Doorkeeper 1 visits all doors and opens them.
- Doorkeeper 2 visits all doors indexed by even numbers and closed them.
- Doorkeeper 3 visits all doors indexed by multiples of 3 and opens the closed ones and closes the open ones.
- In general, Doorkeeper j visits all doors indexed by multiples of j and opens the closed ones and closes the open ones.

Your mission, should you choose to accept it, is to write a MATLAB script which, given n, generates a row vector of integers corresponding to the doors that are open at the end.

Extra Credit [5 points]

Find a mathematical *formula* which identifies all open doors at the end. Write clearly your justification. Then write a single MATLAB statement implementing your formula. Confirm that the statement yields the same result as the script you wrote above.

Table 1: 0 is for "closed" and 1 for "open". For $n = 6$, only two doors (1 and 4) are left open at the end.

$n = 6$	Door 1	Door 2	Door 3	Door 4	Door 5	Door 6
initial	0	0	0	0	0	0
after Doorkeeper 1	1	1	1	1	1	1
after Doorkeeper 2	1	0	1	0	1	0
after Doorkeeper 3	1	0	0	0	1	1
after Doorkeeper 4	1	0	0	1	1	1
after Doorkeeper 5	1	0	0	1	0	1
after Doorkeeper 6	1	0	0	1	0	0