

Homework 6: MAT 331

Due: 11:59pm, Tue 04/07/2020

Part of this homework is written and part is programming. You will submit everything via Blackboard. For the written part, you will upload a single pdf file called `prob1prob2.pdf` that contains solutions to Problems 1 and 2 below. You can make this pdf file in any number of ways, including using a word processor and exporting to pdf, or by scanning hand-written solutions.

For each of Problems 3,4, which involve `turtle` graphics, you will submit a `.py` described in the problem descriptions below. See the “Useful Resources” section of the course website for pointers on using `turtle` graphics. You will use python directly for this, rather than through the Jupyter/Sage notebook.

Grader for this HW (communicate with him about grading issues): Mu Zhao, `mu.zhao@stonybrook.edu`

1. We say that a subset K of Euclidean space has *fractal dimension* d if there exist some integer $n > 1$ such that K is equal to n^d disjoint (non-intersecting) copies of $\frac{1}{n}K$. Here $\frac{1}{n}K$ is obtained from K by scaling by $\frac{1}{n}$. We computed the dimension of a side \mathcal{S} of the Koch snowflake in class by using the fact that \mathcal{S} is equal to 4 copies of $\frac{1}{3}\mathcal{S}$.

The *Cantor set* \mathcal{C} is defined as the limit of the \mathcal{C}_k shown below (ignore the arrows on the right; these come from the turtle). Each \mathcal{C}_k is obtained from \mathcal{C}_{k-1} by removing the middle third of every interval, while \mathcal{C}_0 is just an interval. Compute the fractal dimension of \mathcal{C} .

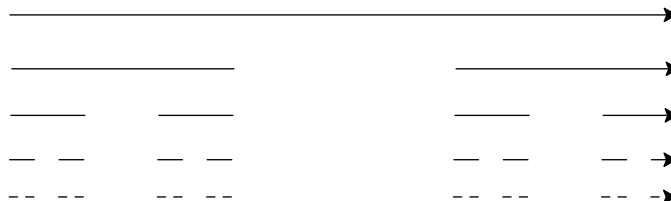


Figure 1: The iteration \mathcal{C}_k for $k = 0, 1, 2, 3, 4$. These converge to the Cantor set.

2. Compute the area enclosed by the full Koch snowflake, an approximation of which is shown below. Assume that the original equilateral triangle used to define the 0th iteration of the Koch snowflake has side-length equal to 1.

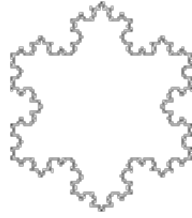


Figure 2: The full Koch snowflake.

3. Use turtle graphics to draw a spiral as below. It does not have to look exactly like the one below, but it should be a path that spirals around a point many times. Upload a file called `spiral.py`. When we run it using the `python` command it should draw the picture without any further modification.

(Hint: use a `for` or `while` loop)

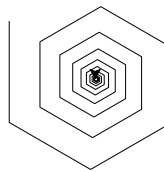


Figure 3: spiral

4. Use turtle graphics to draw the n th iteration of the Cantor set \mathcal{C}_k , defined above. The file you submit should contain a function

```
cantor(size, iter)
```

which when run draws a copy of \mathcal{C}_k of size proportional to `size` (the exact scaling does not matter), and $n = \text{iter}$. Upload a file called `cantor.py`.

(You may want to use the commands `turtle.penup()` and `turtle.pendown()`, which control whether the turtle lays a track.)