Name: Kamdem Louis Mozart

Course: Probabilistic simulation

practical assignement

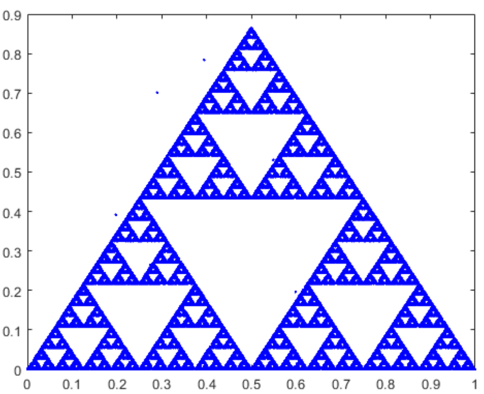
Date: December 26, 2022

Task4:

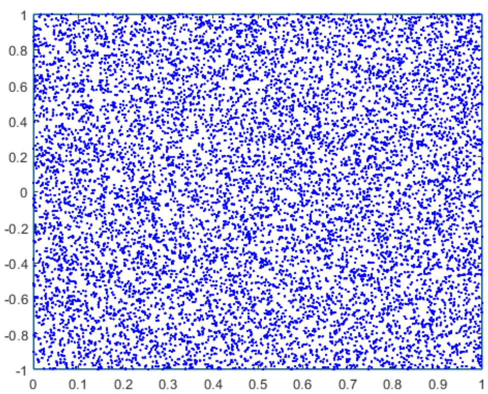
I ***Sierpinski type processes:***

(i) Study the same iteration as in the Sierpinski triangle, but several (4, 5 or 6) regular corner points. Explain the difference in the outcome.

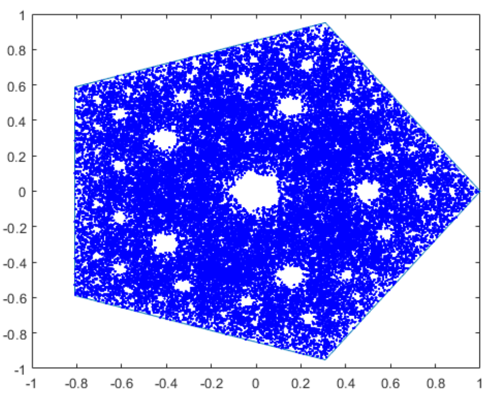
Writing and running the same algorithm of the Sierpinski triangle with several regular corners yields to the plots below:



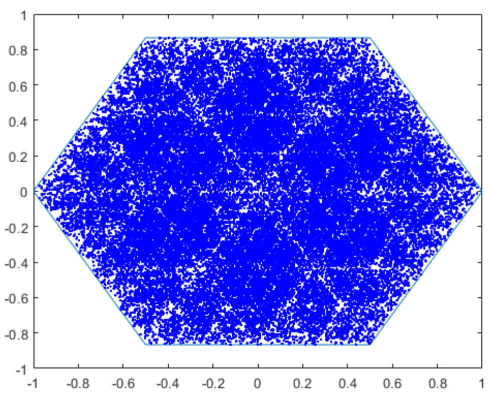
(a) Iteration with 3 regular corners



(b) Iteration with 4 regular corners



(c) Iteration with 5 regular corners



(d) Iteration with 6 regular corners

We observe from the generated plots above that with 4 regular corners, there is no fractal images which appear and the square is just filled with points where as with 3 regular corners we can clearly identify fractal images of equilateral triangles inside the initial triangle with opposites direction as well as for 5 regular corners where we can identify fractal images of pentagones inside the initial pentagone. For the iterations with 6 regular corners, although it's not very clear, we can identify fractal images with a diamond shape filled with points. However, it's possible with the case of 4 regular corners to get a fractal image with and additional condition (see this page) such a selected vertex can't be selected at the next iteration. Taking into account that condition, we got the fractal image below

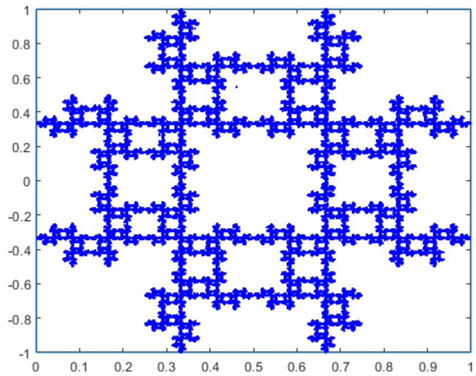


Figure 2: A fractal image using 4 regular corners

(ii) Study the outcome of the Sierpinski triangle iteration, if the probabilities of selecting the corner points are non-uniform, and the step size from the current point to the selected corner a fixed fraction $0 < s < 1$, other than $s = 0.5$.

Using $s = 2/3$ and given: the probability of selecting the first vertice is 0.4 for the second vertice is 0.35 and for the third vetice 0.25. The Sierpinski triangle iteration is now given the output below:

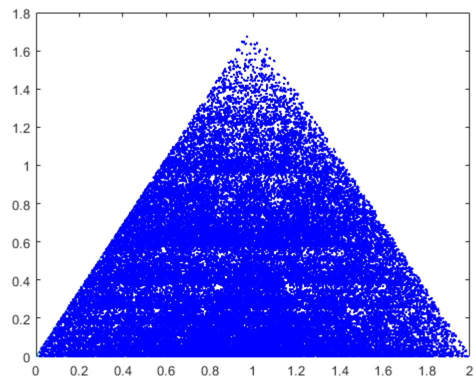


Figure 3: Sierpinski triangle with $s \neq 0.5$ and non uniform selections of vertices

As we can see, no fractal image is coming this time just a filled up regular triangle.

II ***Barnsley fern type figures:***

(1) Study what is the role of the 4 different transformations $f1, f2, f3, f4$ used.

The transformations f_1, f_2, f_3 and f_4 are mappings with fifferent roles Let's consider the following image which is a stem generated by the Barnsley fern algorithm:

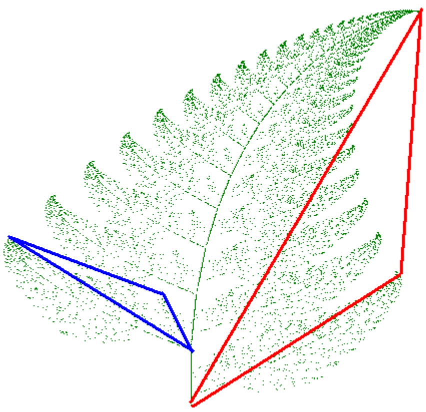


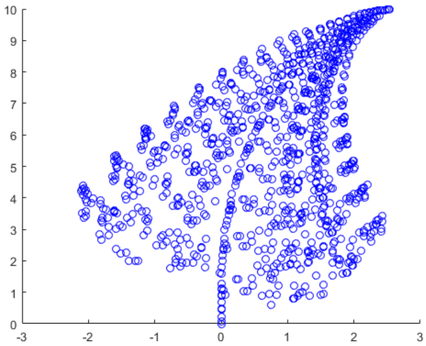
Figure 4: Image from this page

- The mapping f_1 which is chosen with a probability of 0.01 at eah iteration maps any points to another point in the first line segment located at the base of the stem.
- The mapping f_2 which is chosen with a probability of 0.85 at eah iteration maps any points located in the big red triangle (drawn on figure (4)) of the stem to the smaller one (in blue).
- The mapping f_3 which is chosen with a probability of 0.07 at eah iteration maps any point in the blue triangle to a point inside the corresponding flipped triangle along the stem
- The mapping f_4 also choose with a probability of 0.07 is doing the same job as f_3 but this time, without flipping the triangle.

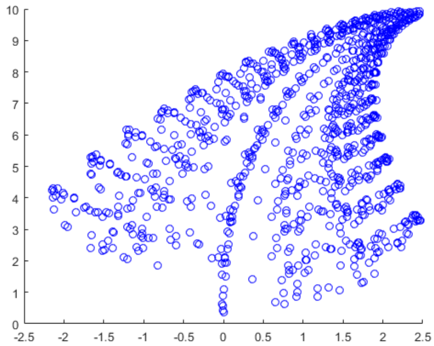
Running the Barnsley fern algorithm

(2) Study the sensitivity of the simulation outcome with respect to randomized noise in the matrix components of added in the transformations, as well as in the probabilities of selecting them. Use relative Gaussian/uniform noise of size, e.g., $1\%, 5\%$etc

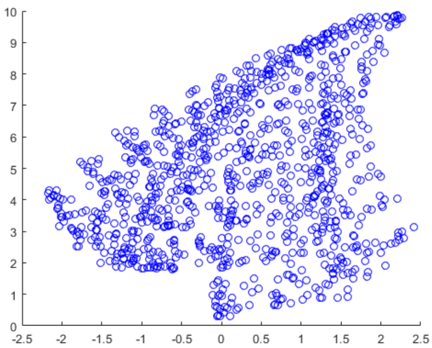
To achieve this, I added some noises(Gaussian and Uniform) to the added matrices and to the probability of choosing the functions f_1, f_2, f_3 and f_4. For the case of gaussian noise, here are the results:



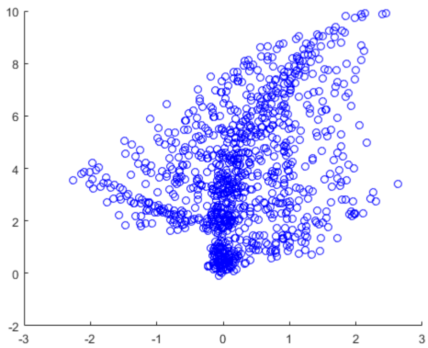
(a) Gaussian noise level = 0



(b) Gaussian noise level =1%

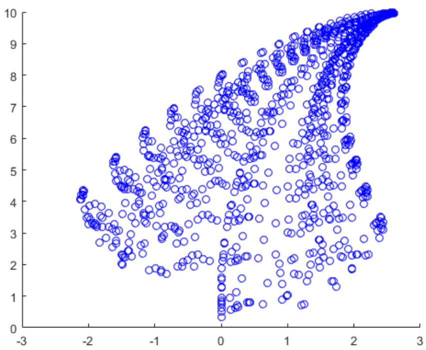


(c) Gaussian noise level =5%

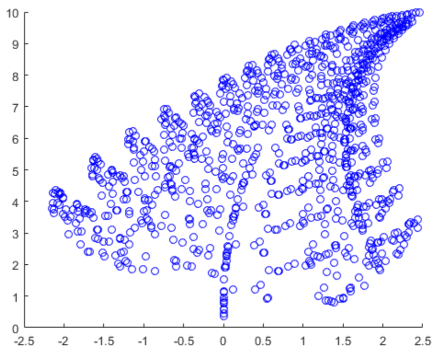


(d) Gaussian noise level =10%

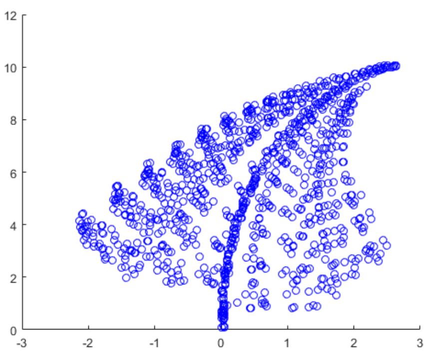
And when adding Uniform noise, here are the results



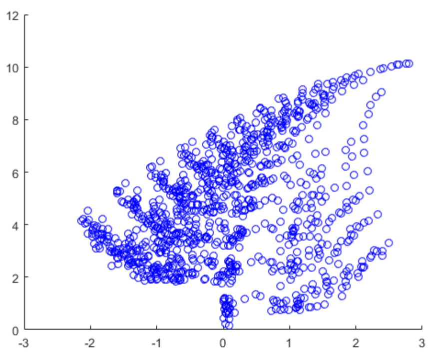
(a) Uniform noise level = 0



(b) Uniform noise level =1%



(c) Uniform noise level =5%



(d) Uniform noise level =10%

We can see that the simulation is more sensitive to gaussian noise than to uniform noise. In fact, We can still well recognize the shape of the stem when we add uniform noise at a level of 5% but not when it is a gaussian noise. Also, with the Gaussian noise of 10%, we observe that all the data point tend to converge to the initial chosen point instead of converging to and horizon point.