CSE585: Digital Image Processing II
Computer Project 5
Fractal Generation Using Iterated Function Systems

Aishwarya Mallampati, Wushuang Bai, Mrunmayi Ekbote,

Date:04/23/2020

1 Objectives

The main objectives of this project are:

- Learn to generate fractal images using iterated function systems(IFS).
- Observe the effect of changing probabilities and the number of iterations on the final fractal images generated.

2 Methods

2.1 Iterated Function Systems

2.1.1 Theory and Algorithms

Iterated function systems (IFSs) are a method of constructing fractals; the resulting fractals are often self-similar. IFS fractals, as they are normally called, can be of any number of dimensions, but are commonly computed and drawn in 2D. The fractal is made up of the union of several copies of itself, each copy being transformed by a function. The functions are normally contractive, which means they bring points closer together and make shapes smaller. Hence, the shape of an IFS fractal is made up of several possibly-overlapping smaller copies of itself, each of which is also made up of copies of itself, ad infinitum. This is the source of its self-similar fractal nature.

An affine transformation $w : R^2- > R^2$ from two dimensional space R^2 into itself is defined by

$$w\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}a_{00} & a_{01}\\a_{10} & a_{11}\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}t_0\\t_1\end{bmatrix} \tag{2.1}$$

where $a'_{ij}s$ and t'_is are real constants. The affine transformation is defined by six real numbers. An affine transformation is called contractive if $s < 1$ in the eq2.2

$$||w(\vec{x},\vec{y})|| \leq s.||\vec{x}-\vec{y}|| \tag{2.2}$$

A two dimensional IFS consists of a set of N affine transformations - $\{w_1, w_2, w_3, ..., w_N\}$ and a set of probabilities $\{p_1, p_2, p_3, ..., p_N\}$ where each $p_i > 0$ and $(p_1+p_2+p_3+...+p_N = 1)$ For an IFS code, there is a unique associated geometrical object, denoted by A which is a subset of R^2, called the attractor of the IFS. There is also a unique associated measure

denoted by μ. The structure of A is controlled by the affine maps $\{w_1, w_2, w_3, ..., w_N\}$ in the IFS code. That is, the 6.N numbers in the affine maps fix the geometry of the underlying model and will in turn determine the geometry of associated images. The measure μ is governed by the probabilities $\{p_1, p_2, p_3, ..., p_N\}$ in the IFS code. It is this information that provides rendering information for images.
By fixing a seed point and viewing window, the provided affine transformations are iterated random number of times to generate a deterministic image as result which is the attractor of the transformations.

2.1.2 Matlab

All the code used for this problem is placed in main.m file. The following are the matlab files used in this project:

- main.m: The following parameters are fixed in this file:
  - The size of the fractal image is set to 1024 x 1024.
  - User is allowed to choose the probability set and the number of iterations to run the algorithm.
  - Displays the final fractal image generated after all the number of iterations.
- render_IFS.m: The following steps are performed in this file using the parameters passed to it:
  - Initialize an image of size 1024 x 1024.
  - Initialize a starting point(x,y).
  - Runs the algorithm for the specified number of times.
  - In each iteration, generate a random number.
  - Based on random number, choose the affine transform to be applied.
  - Apply the transform and get the transformed values.
  - Check if the X and Y values are within the max and min range. Calculate which pixel corresponds to this iteration and set that pixel to 1.
  - The updated vector(x,y) will be the seed point for next iteration.
- affine_transform.m: Applies affine transform based on passed parameters.

Note: Run main.m file to get the results

The flowcharts for this project is displayed in Figure1

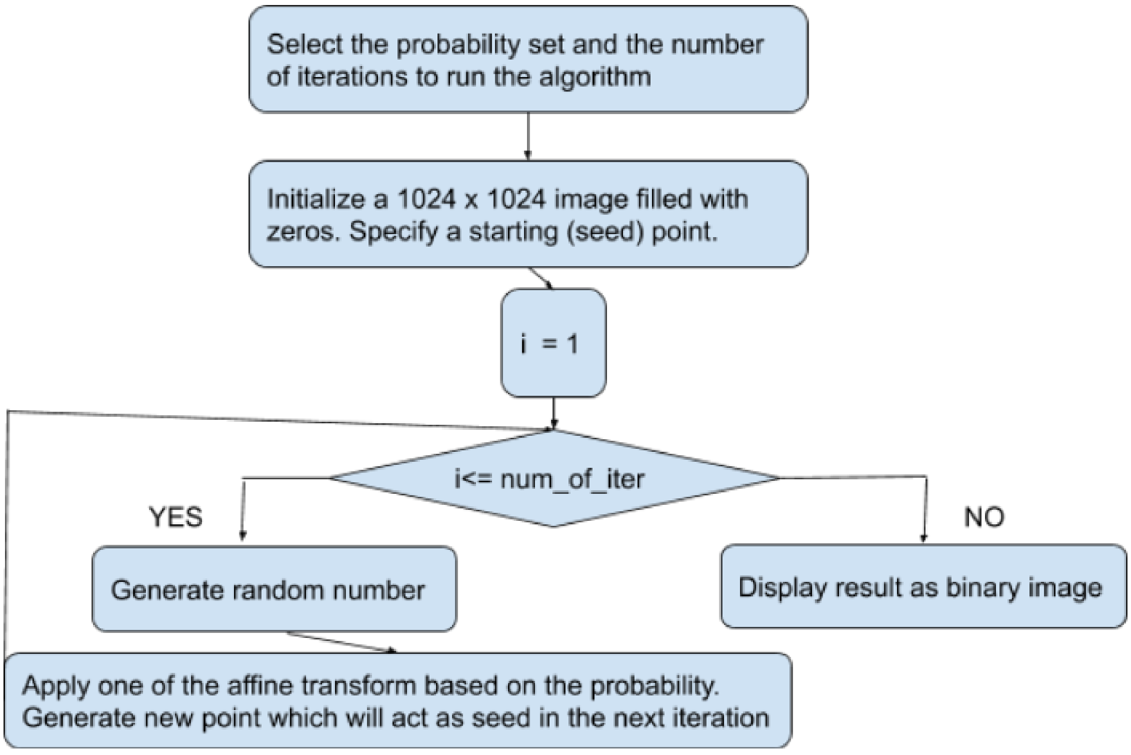
Select the probability set and the number of iterations to run the algorithm
Initialize a 1024 x 1024 image filled with zeros. Specify a starting (seed) point.
i = 1
i<= num_of_iter
YES
NO
Generate random number
Display result as binary image
Apply one of the affine transform based on the probability. Generate new point which will act as seed in the next iteration

Figure 1: Flowchart :Iterated Function Systems

3 Results

In order to generate fractal images using iterative functional systems, the following affine transforms are used:

W	a_{00}	**a_{01}**	**a_{10}**	a_{11}	t_{0}	t_{1}
1	**0**	0	0	0.16	0	0
2	**0.2**	-0.26	0.23	0.22	0	1.6
3	**-0.15**	0.28	0.26	0.24	0	0.44
4	**0.85**	0.04	-0.04	0.85	0	1.6

Table 1: AFFINE TRANSFORMS

Different probability sets are used to generate the results which are outlined in the following subsections.

3.1 ProbabilitySet1: [0.2, 0.35, 0.35, 0.1]

Figures2 - 9 displays the results for various iteration in the increasing order. The probability set corresponding to these results is [0.2, 0.35, 0.35, 0.1]. If can be seen that with increasing number of iterations, a better fractal is achieved.

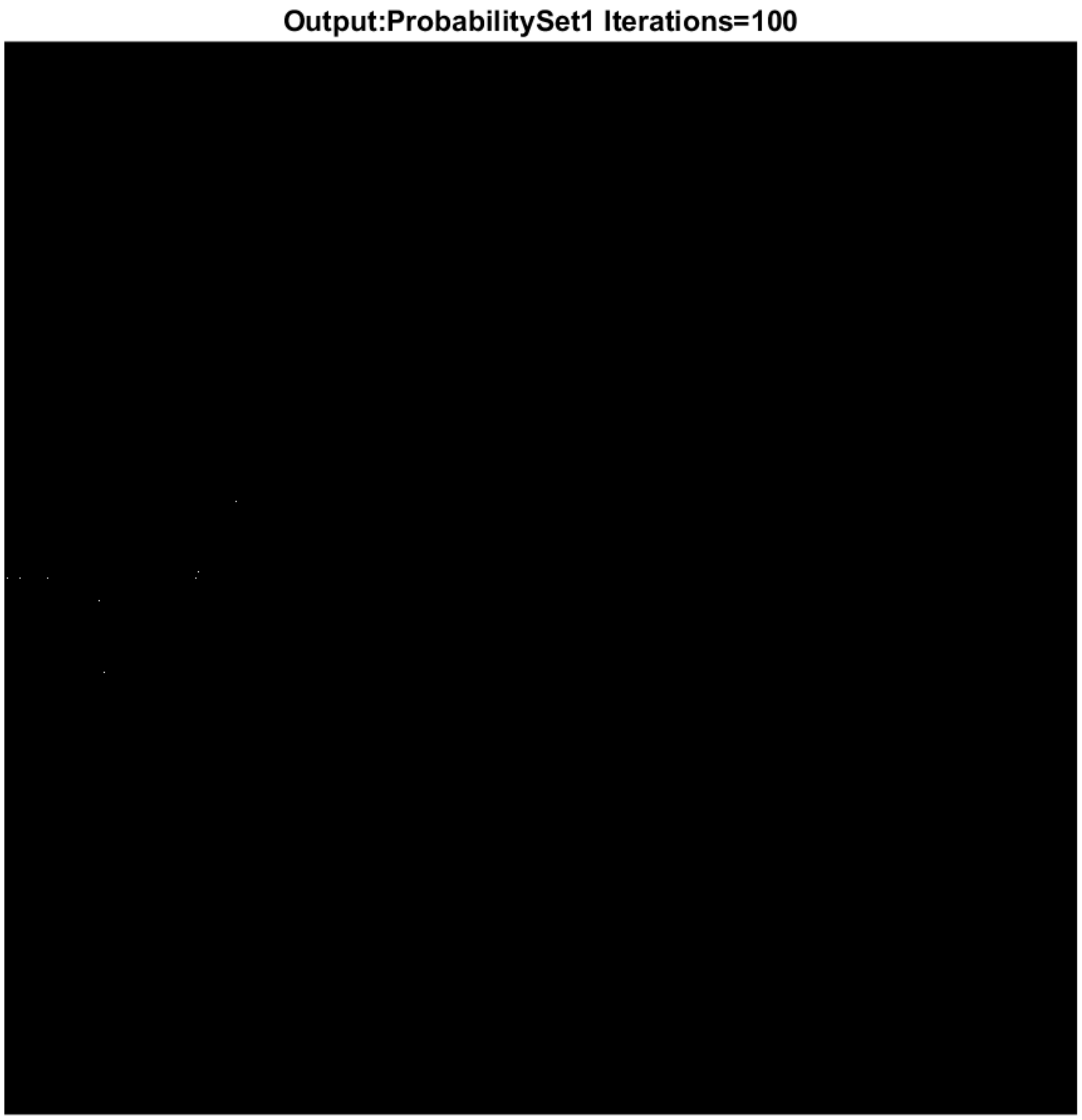

Figure 2: Fractal image obtained after applying IFS with ProbabilitySet1 for 100 iterations

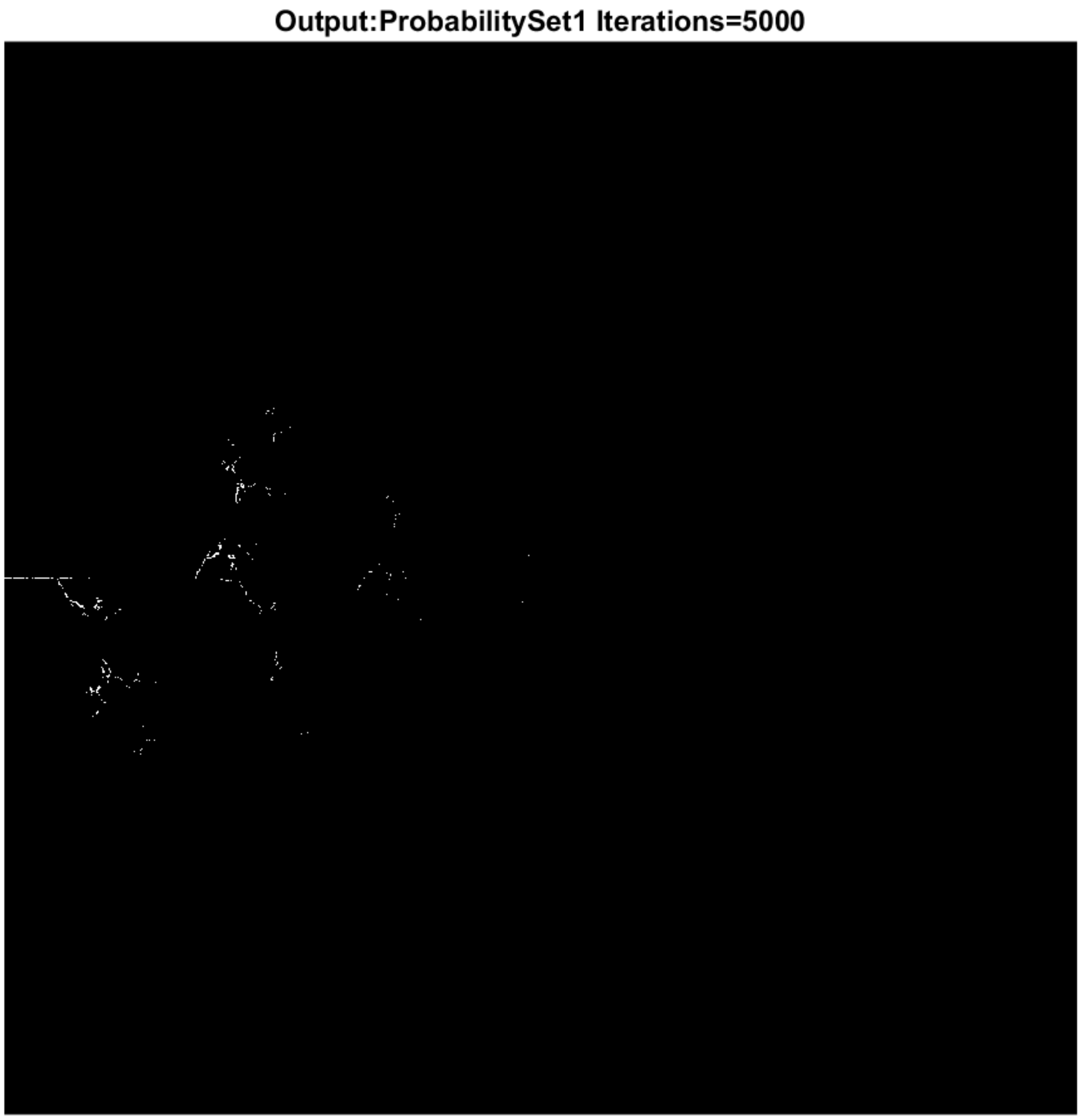

Figure 3: Fractal image obtained after applying IFS with ProbabilitySet1 for 5000 iterations

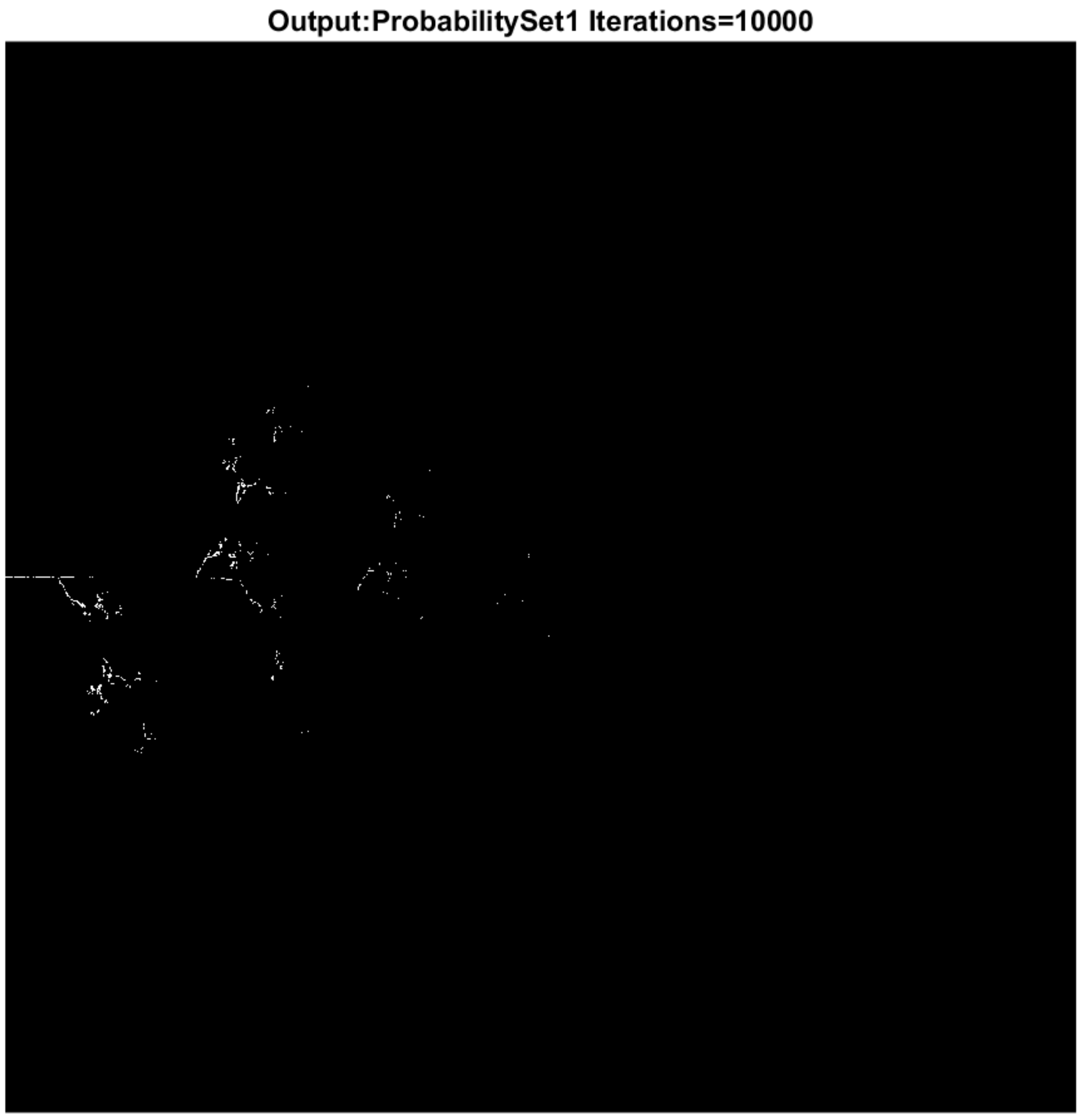

Figure 4: Fractal image obtained after applying IFS with ProbabilitySet1 for 10000 iterations

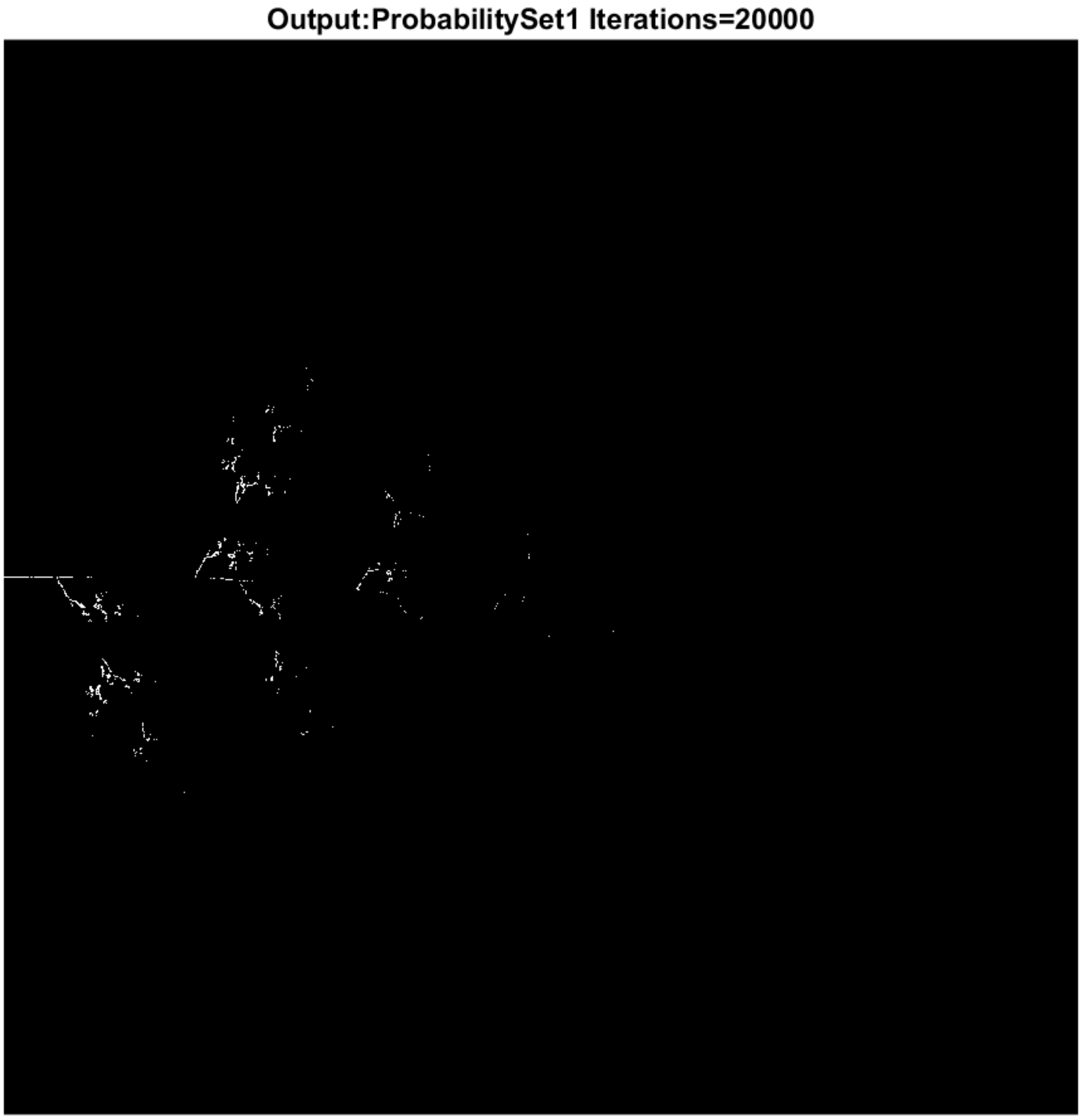

Figure 5: Fractal image obtained after applying IFS with ProbabilitySet1 for 20000 iterations

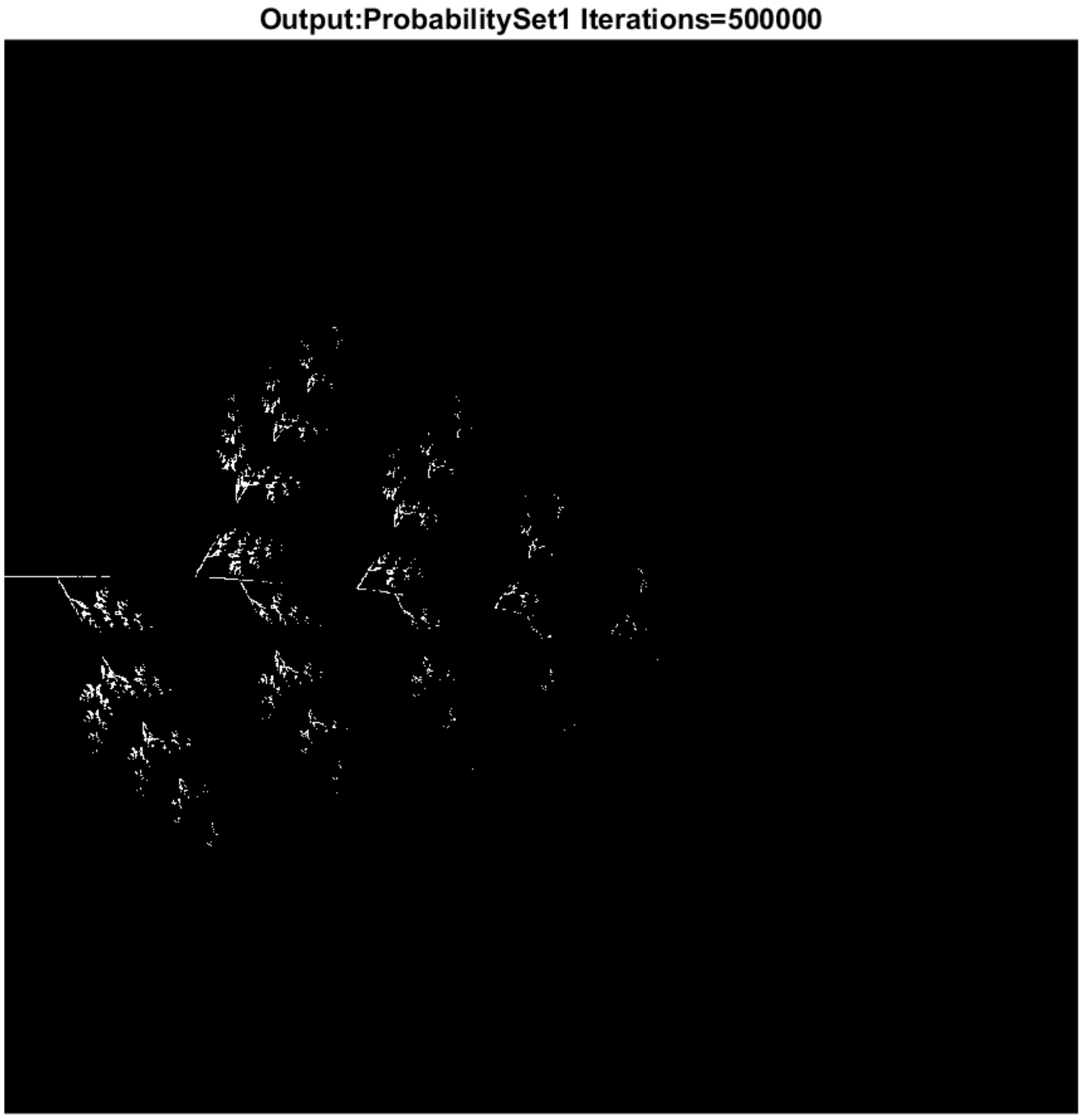

Figure 6: Fractal image obtained after applying IFS with ProbabilitySet1 for 500000 iterations

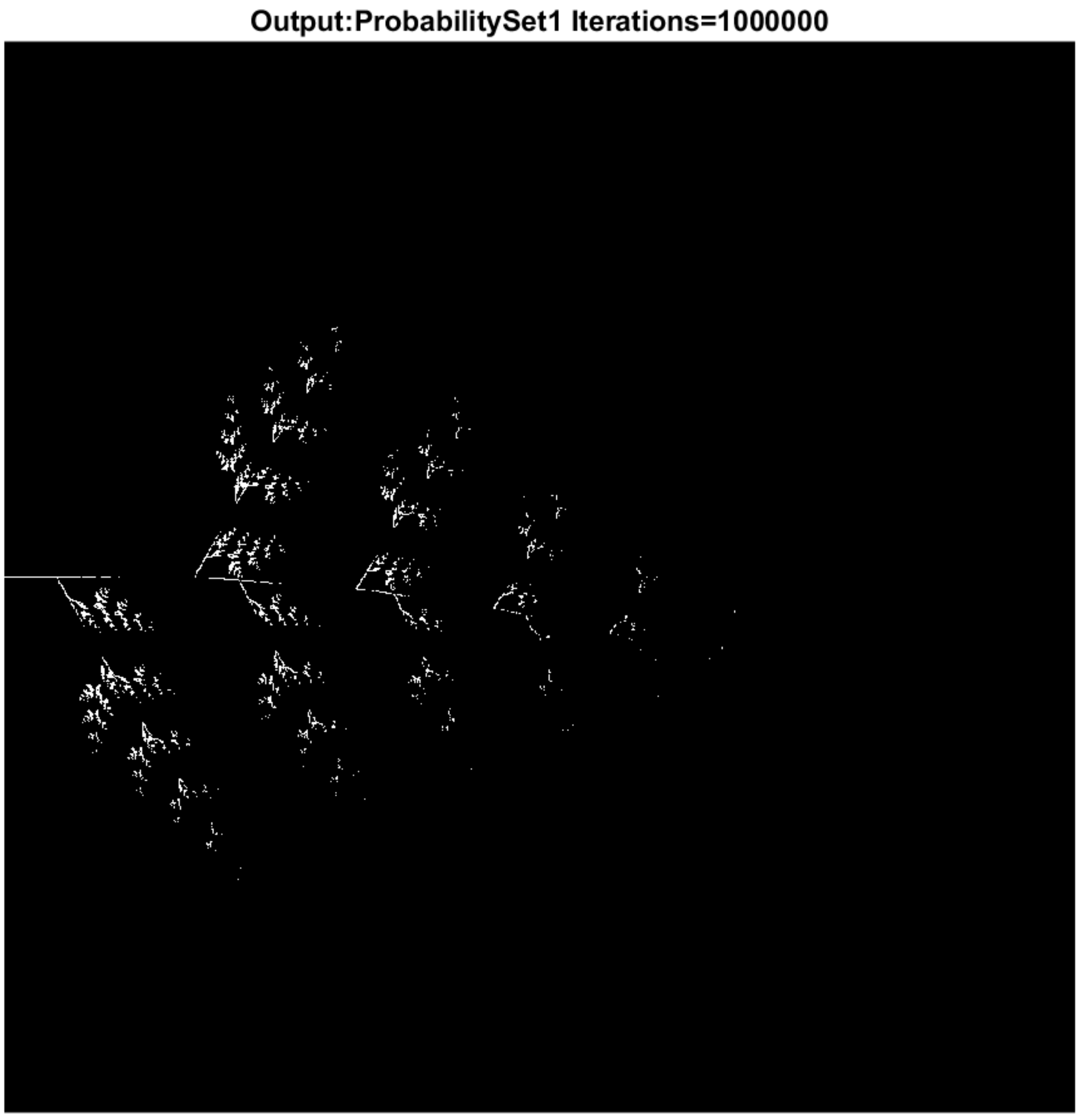

Figure 7: Fractal image obtained after applying IFS with ProbabilitySet1 for $10^6 iterations$

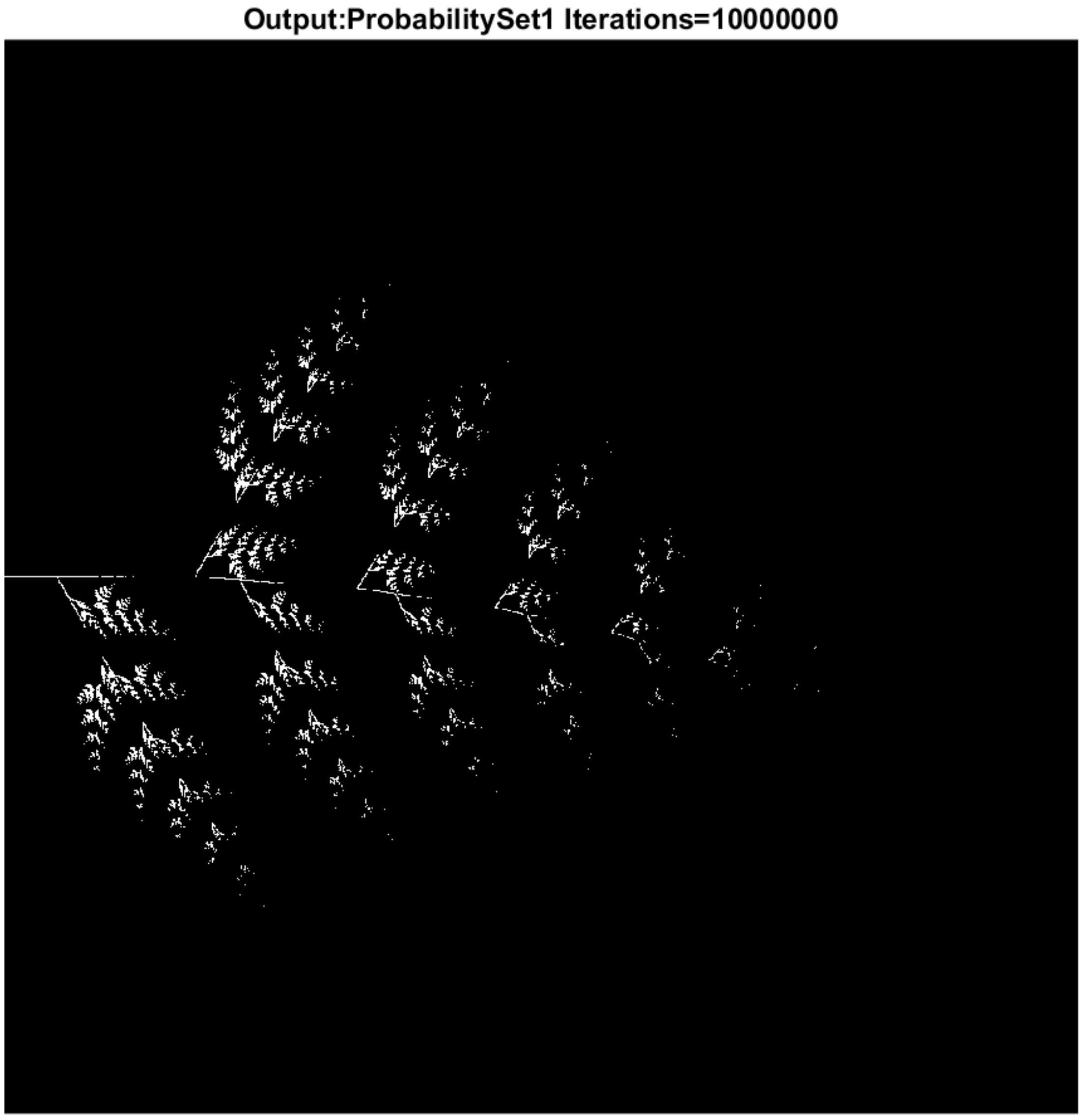

Figure 8: Fractal image obtained after applying IFS with ProbabilitySet1 for $10^7 iterations$

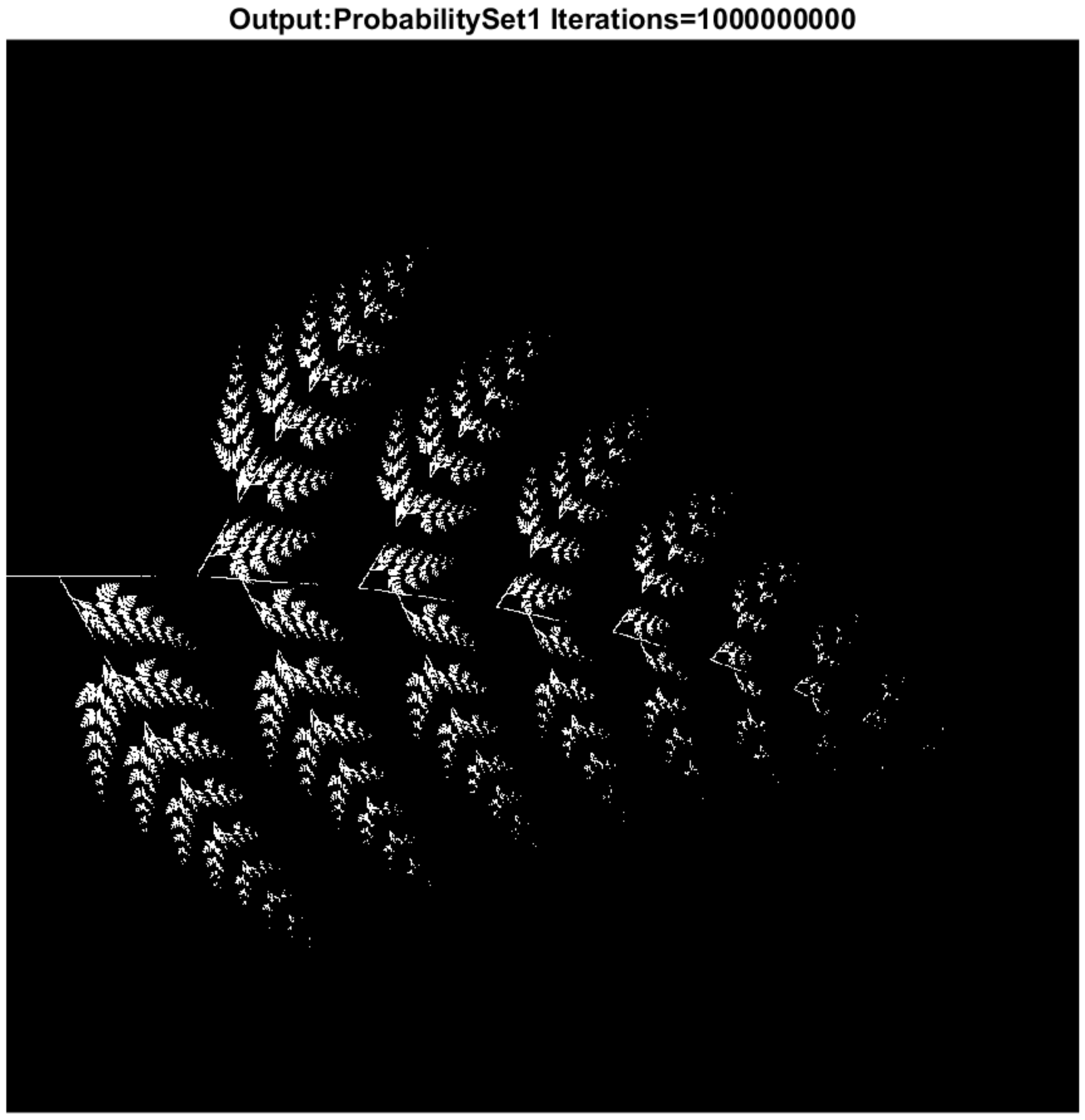

Figure 9: Fractal image obtained after applying IFS with ProbabilitySet1 for $10^9 iterations$

3.2 ProbabilitySet2: [0, 0.35, 0.35, 0.3]

Figures10 - 11 displays the results for various iteration in the increasing order. The probability set corresponding to these results is [0, 0.35, 0.35, 0.3].

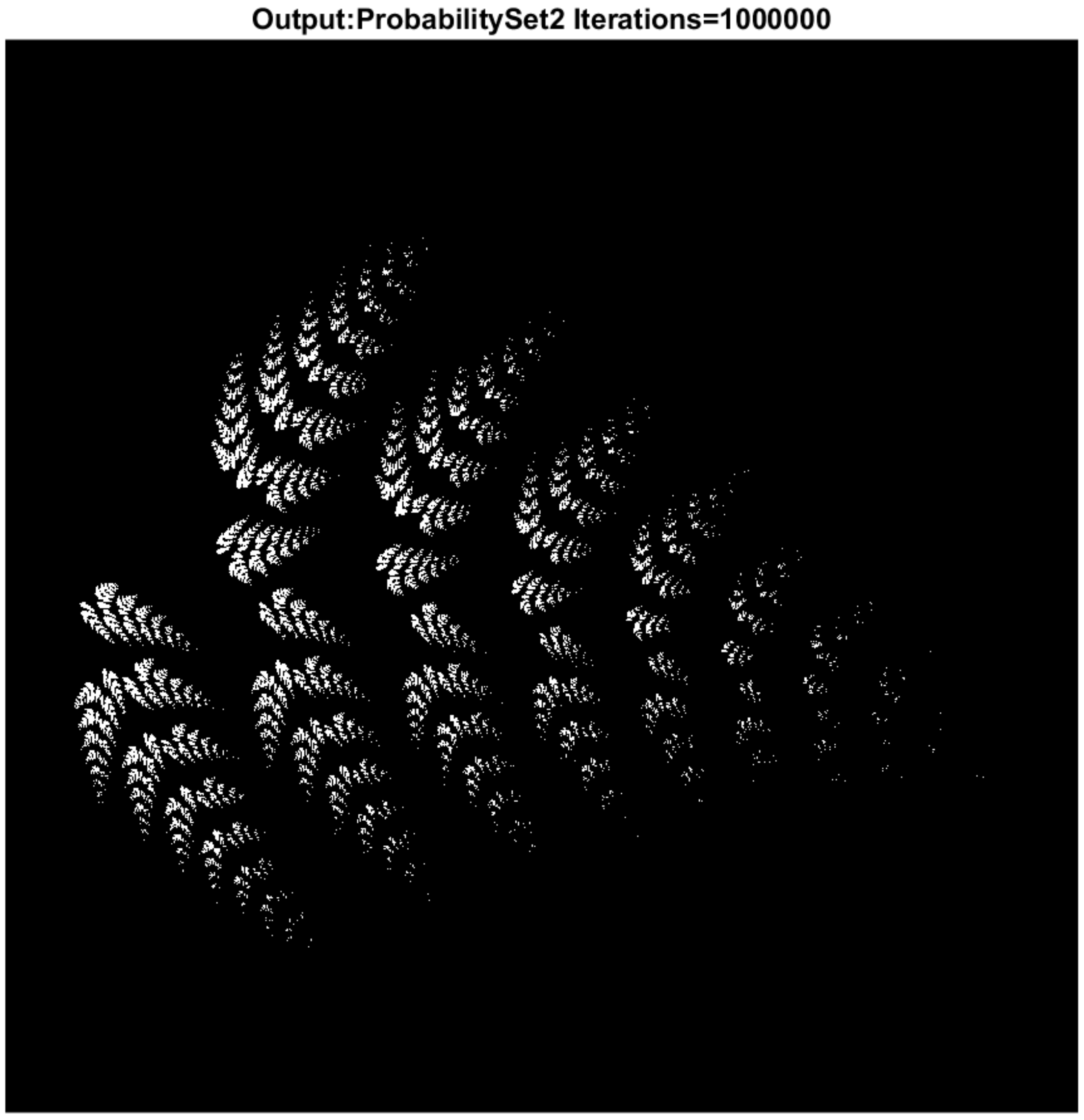

Figure 10: Fractal image obtained after applying IFS with ProbabilitySet2 for $10^6 iterations$

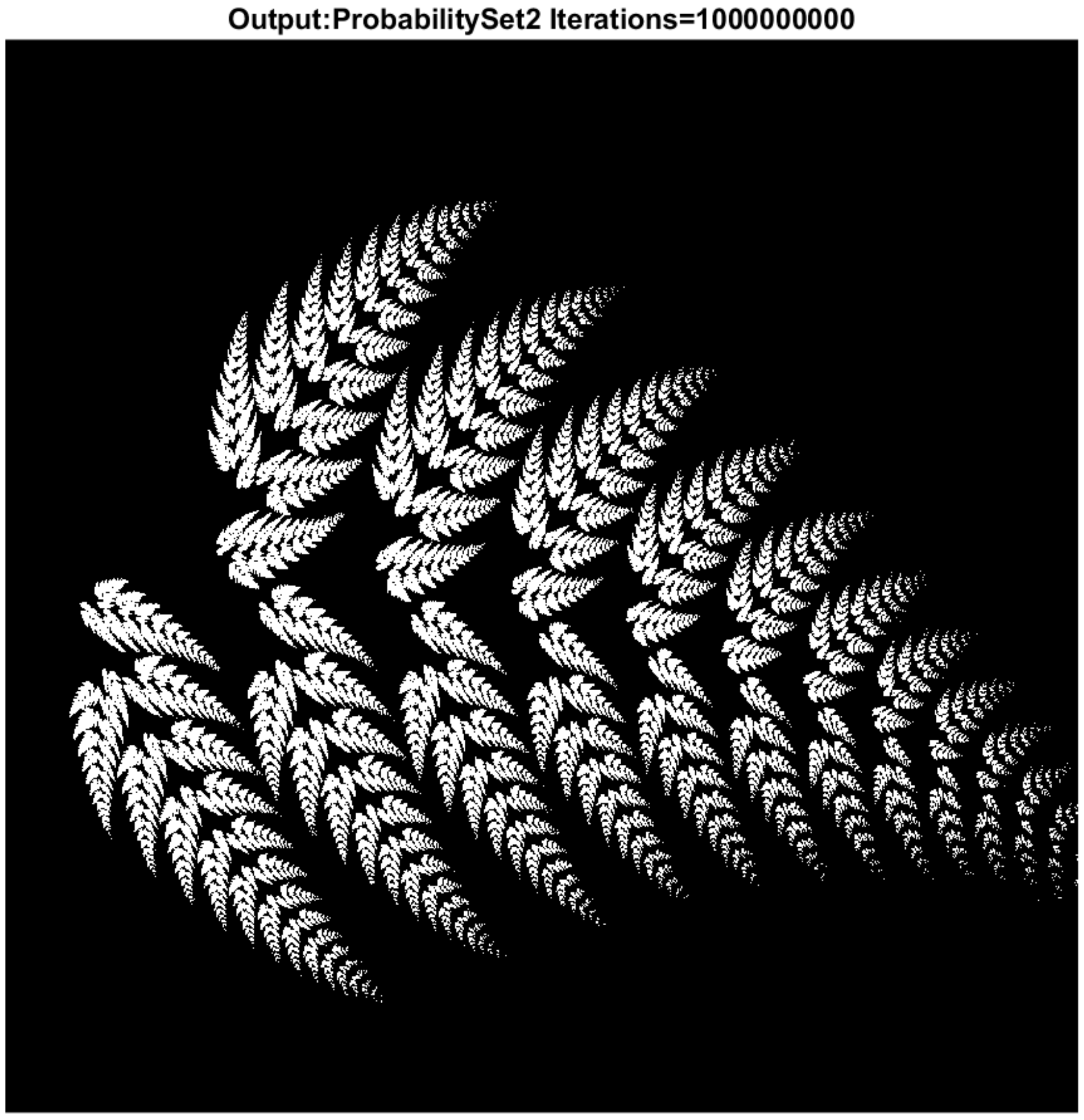

Figure 11: Fractal image obtained after applying IFS with ProbabilitySet2 for $10^9 iterations$

3.3 ProbabilitySet3: [0.4, 0, 0.35, 0.25]

Figures12 - 13 displays the results for various iteration in the increasing order. The probability set corresponding to these results is [0.4, 0, 0.35, 0.25].

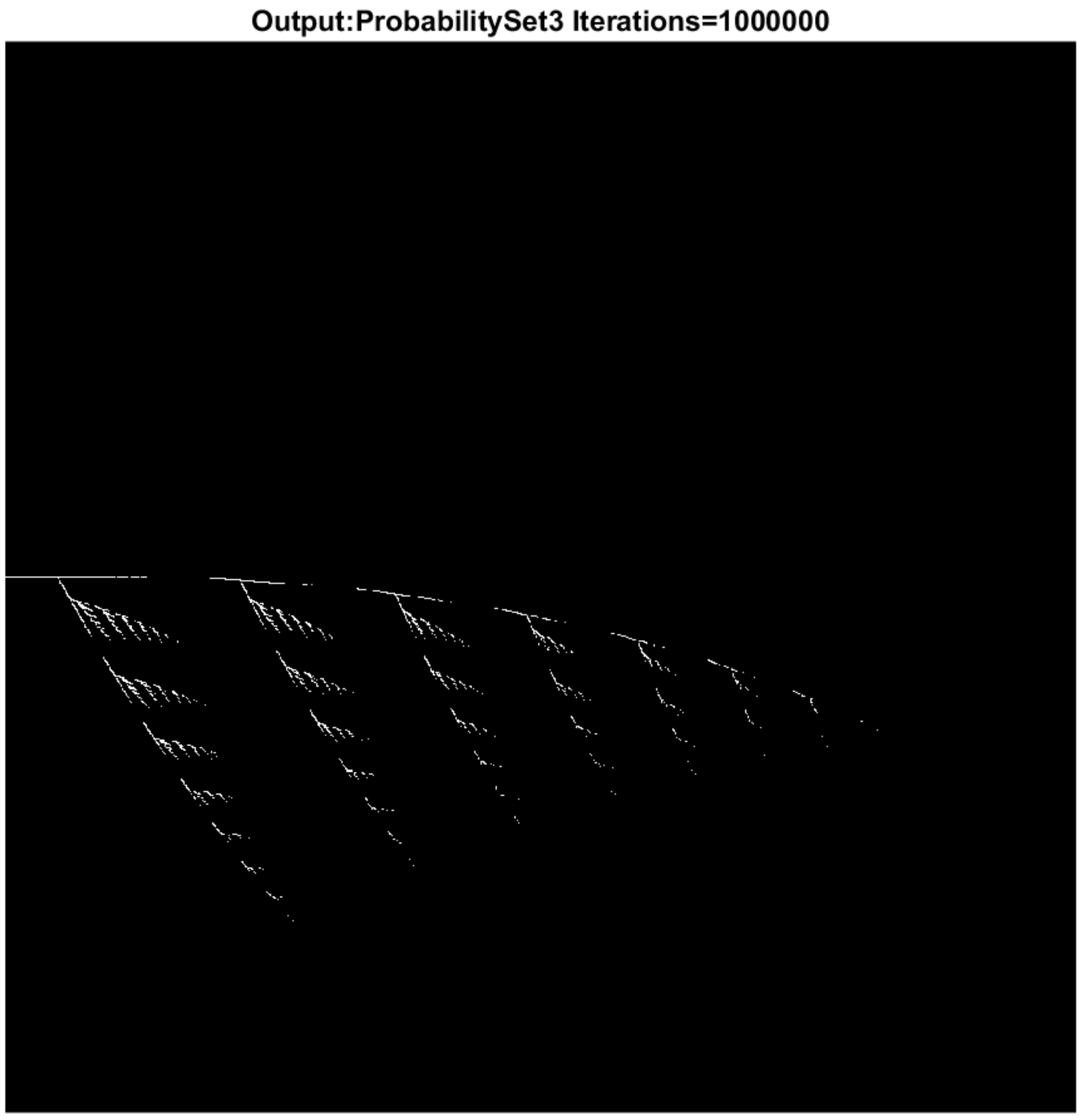

Figure 12: Fractal image obtained after applying IFS with ProbabilitySet3 for $10^6 iterations$

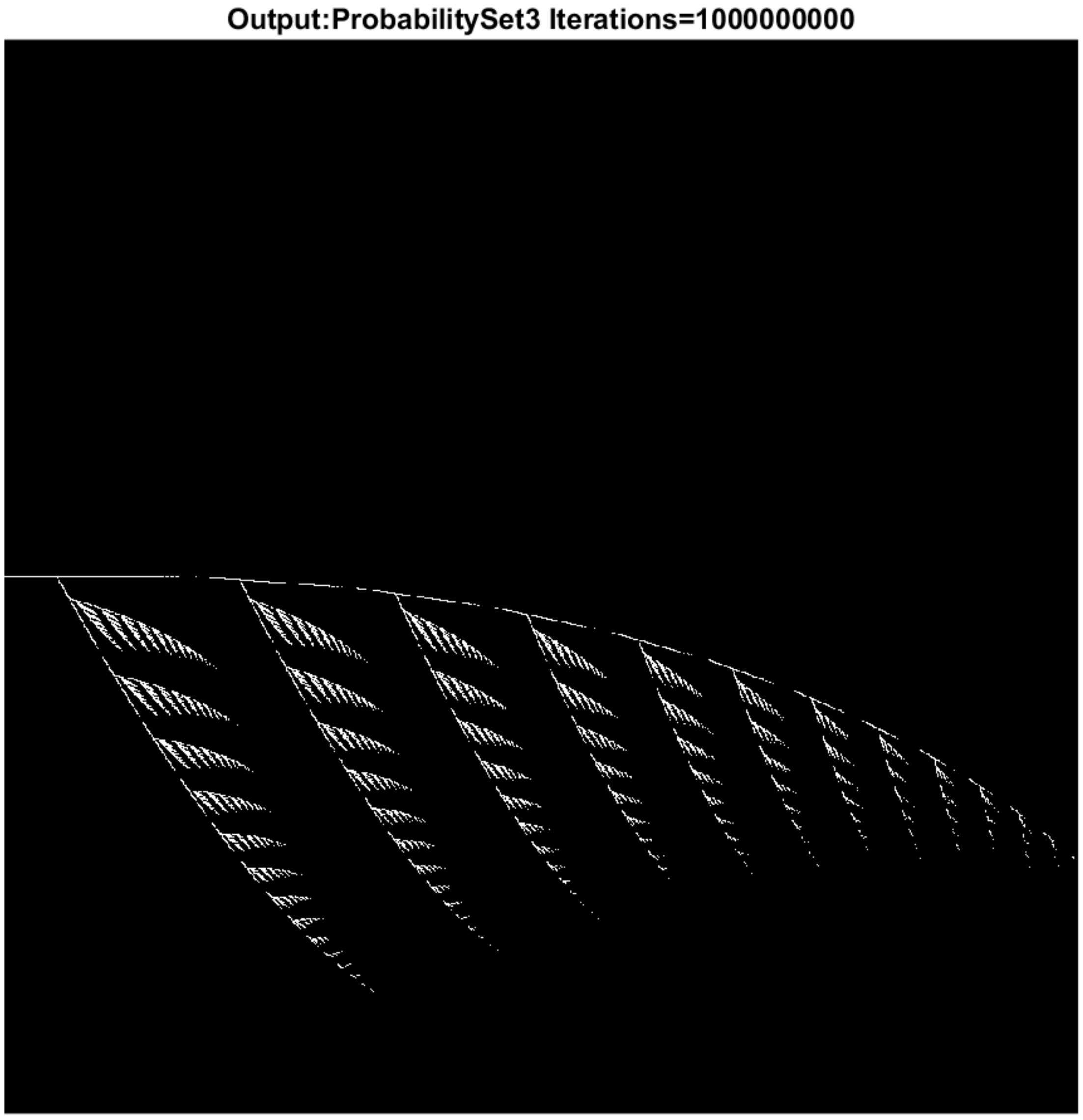

Figure 13: Fractal image obtained after applying IFS with ProbabilitySet3 for $10^9 iterations$

3.4 ProbabilitySet4: [0.2,0.35,0,0.1]

Figures14 display the results for the probability set[0.2,0.35,0,0.1] for $10^6 iterations$.

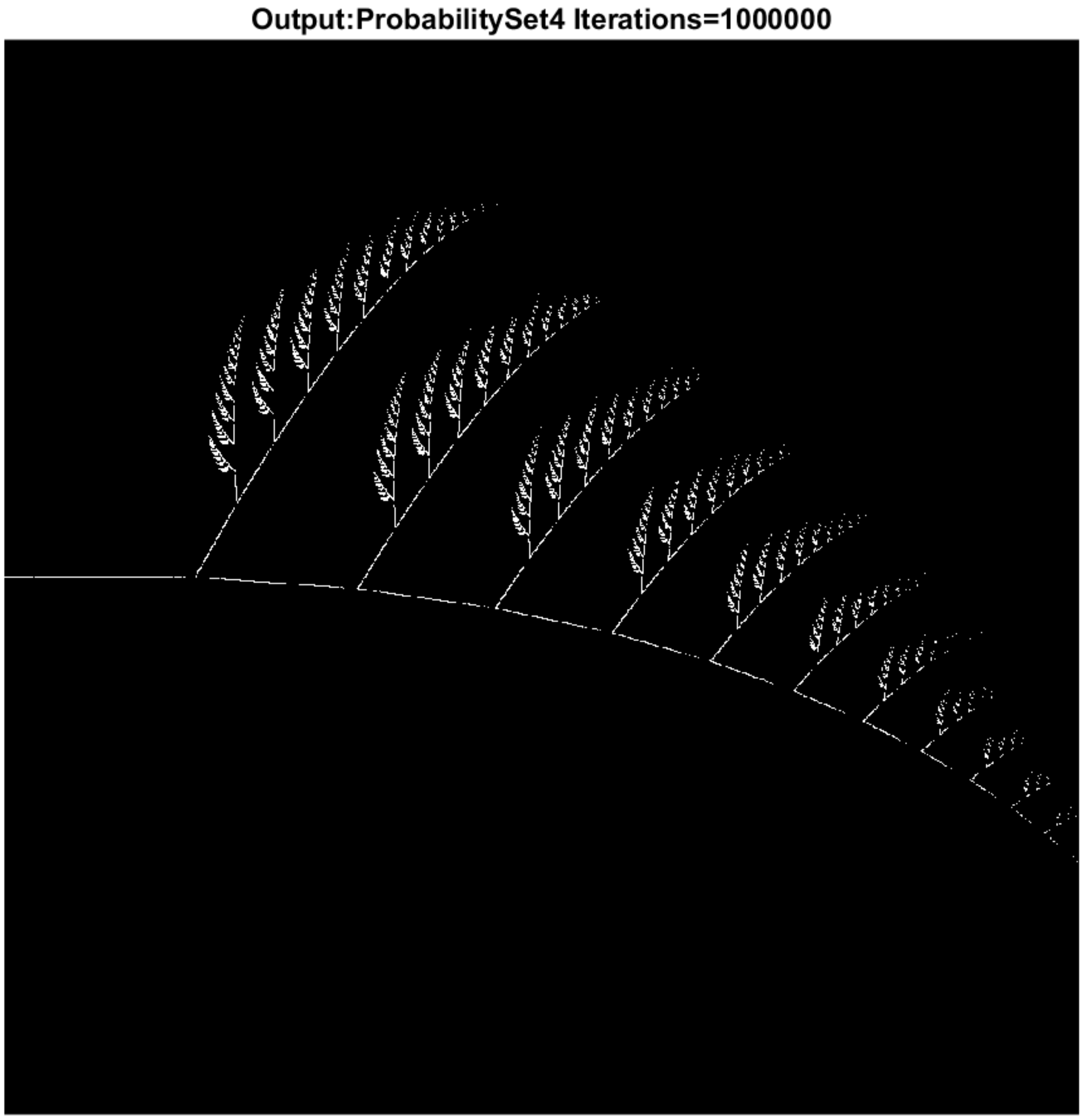

Figure 14: Fractal image obtained after applying IFS with ProbabilitySet5 for $10^6 iterations$

3.5 ProbabilitySet5: [0.2, 0.35, 0, 0.45]

Figures15 display the results for the probability set[0.2, 0.35, 0, 0.45] for $10^6 iterations$.

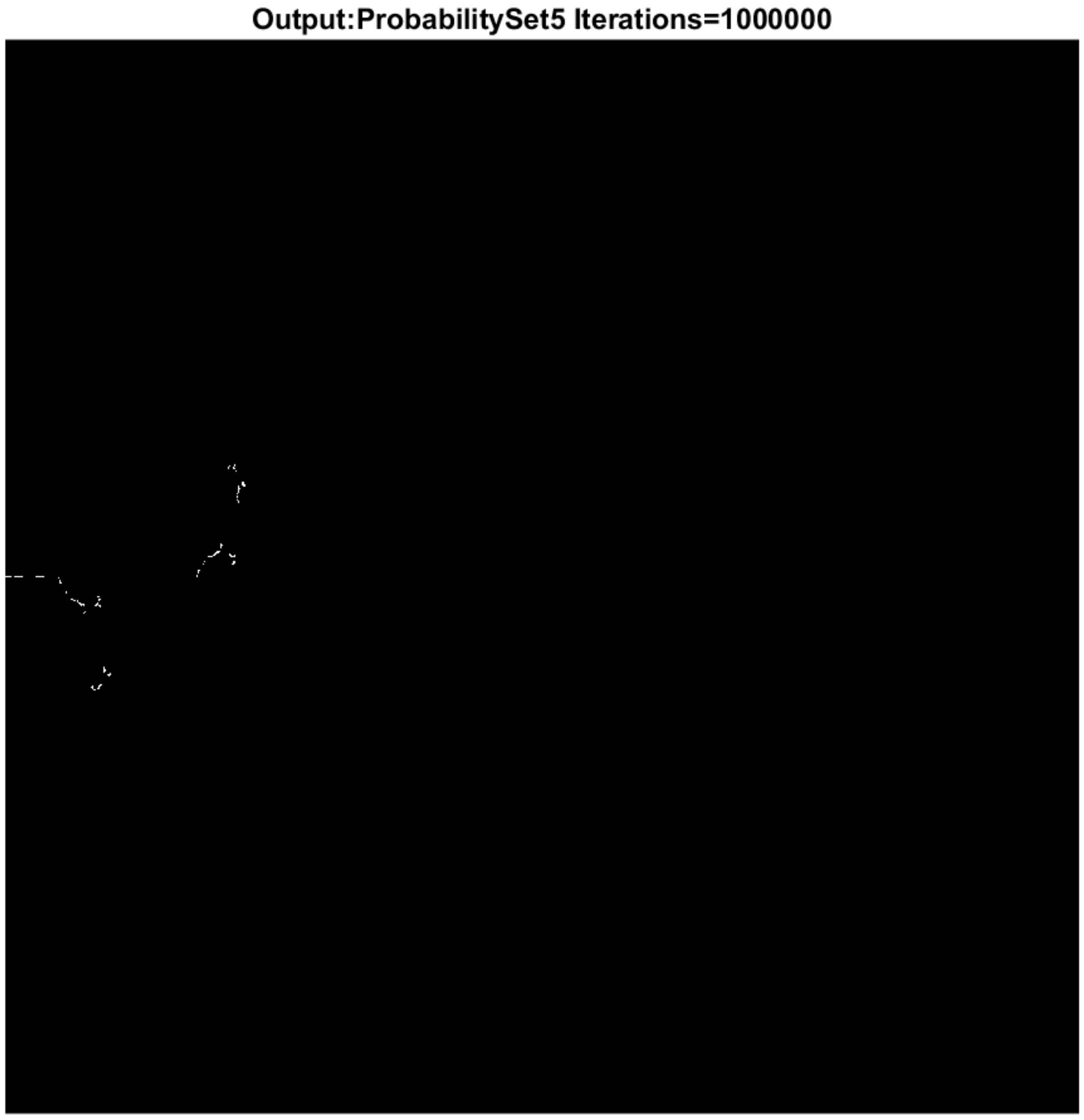

Figure 15: Fractal image obtained after applying IFS with ProbabilitySet5 for $10^6 iterations$

4 Conclusion

IFS is useful for generating fractal images, like ferns. The smallest geometric structure in the complete fractal image is determined by the affine transforms. The probability set decides the overall shape of the fractal image. The time by the algorithm to generate the required fractal depends on the probability set. So, the probability set needs to be chosen carefully such that the program runs for minimum amount of time. The time taken by program also increases as the number of iterations is increased.