Last month’s blog introduced the idea of the chaos game, which is a simple and compact computing method for generating deterministic fractals using a Monte Carlo-like random method. The central ingredients of the chaos game are an affine transformation of the plane

\[ \left[ \begin{array}{c} x\_i \\ y\_i \end{array} \right] = \left[ \begin{array}{cc} a^{(p)} & b^{(p)} \\ c^{(p)} & d^{(p)} \end{array} \right] \left[ \begin{array}{c} x\_{i-1} \\ y\_{i-1} \end{array} \right] + \left[ \begin{array}{c} f^{(p)} \\ g^{(p)} \end{array} \right] \; ,\]

which expresses the current point (labeled by the index i) in terms of the previous point (labeled by the index i-1) in terms of known, fixed parameters $$\{a^{(p), b^{(p), c^{(p), d^{(p), f^{(p), g^{(p)}$$. The index $$p$$ is an integer label for the various affine transformations available and is chosen at random at each step, according to some pre-defined probability distribution. For the Sierpinski triangle discussed in the previous post, $$p$$ can take on the values (1,2,3) each with probability 1/3.

It is convenient to package all the information into a table with each row corresponding to a one of the affine transformations. For the Sierpinski triangle, the table is given by

<

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cumulative Probability | a | b | c | d | f | g |
| 0.33 | 0.5 | 0 | 0 | 0.5 | 1 | 1 |
| 0.66 | 0.5 | 0 | 0 | 0.5 | 1 | 50 |
| 1.00 | 0.5 | 0 | 0 | 0.5 | 50 | 50 |

>

, which is directly translated into the numpy array

<

sierpinski\_triangle = np.array([[0.33,0.5,0,0,0.5, 1, 1],

[0.66,0.5,0,0,0.5, 1,50],

[1.00,0.5,0,0,0.5,50,50]])

>

used for simulation (details in the previous post).

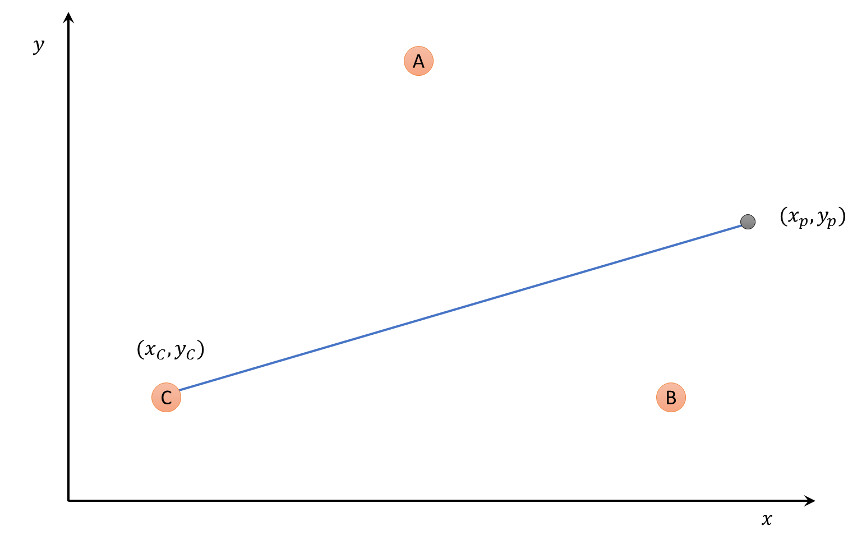
The above material, which is a quick recap of last month’s entry, sets the stage for this post, which focuses on understanding how the structures emerge from such a simple set of rules.

A good starting point for that understanding is the Numberphile video on the chaos game

<iframe width="560" height="315" src="https://www.youtube.com/embed/kbKtFN71Lfs" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

The presenter, a mathematician by the name of Ben Sparks, starts by showing how the chaos game for the Sierpinski triangle is played by examining single iterations performed by hand. He performs a single iteration by taking the initial point, rolling a six-sided die (yes die is the correct word for the singular – like Sparks that grammatical rule confuses me as well) to randomly pick one of the three vertices, connecting the point to the chosen vertex with a straight line, and then establishing the new point at a distance halfway along that line.

To map this procedure to a mathematical algorithm, start with the situation pictured below.



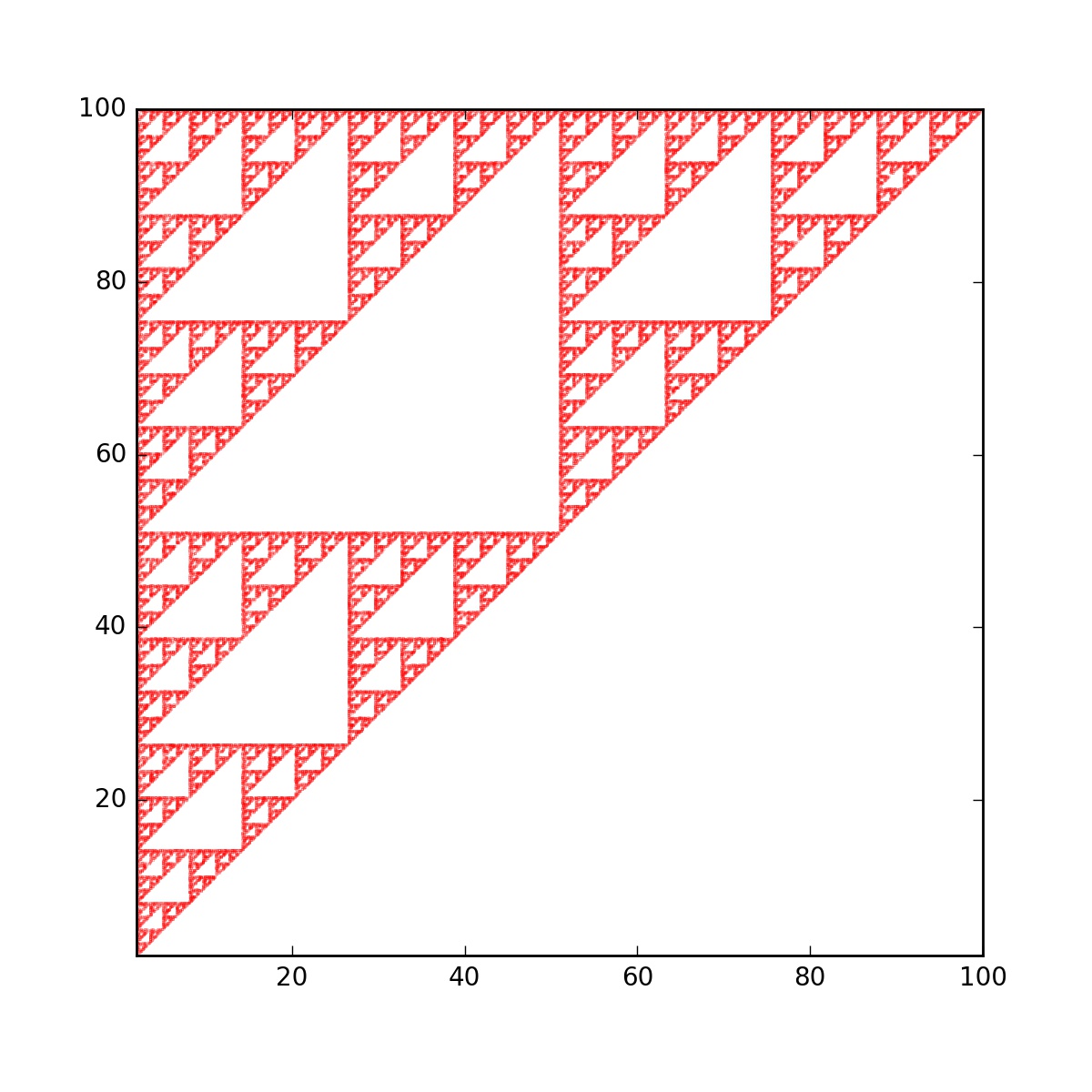
Assuming that vertex C is chosen by random, the line joining the current point to the vertex is given by

\[ \begin{array}{c}\left[ x \\ y\right]\end{array} = \begin{array}{c}\left[ x\_p - x\_C \\ y\_p - y\_C\right]\end{array} + \lambda \begin{array}{c}\left[ x\_p - x\_C \\ y\_p - y\_C\right]\end{array} \; , \]

where the parameter $$\lambda$$ ranges from 0 to 1. The point halfway from the initial point to the vertex C has coordinates given by substituting $$\lambda = 1/2$$ to yield

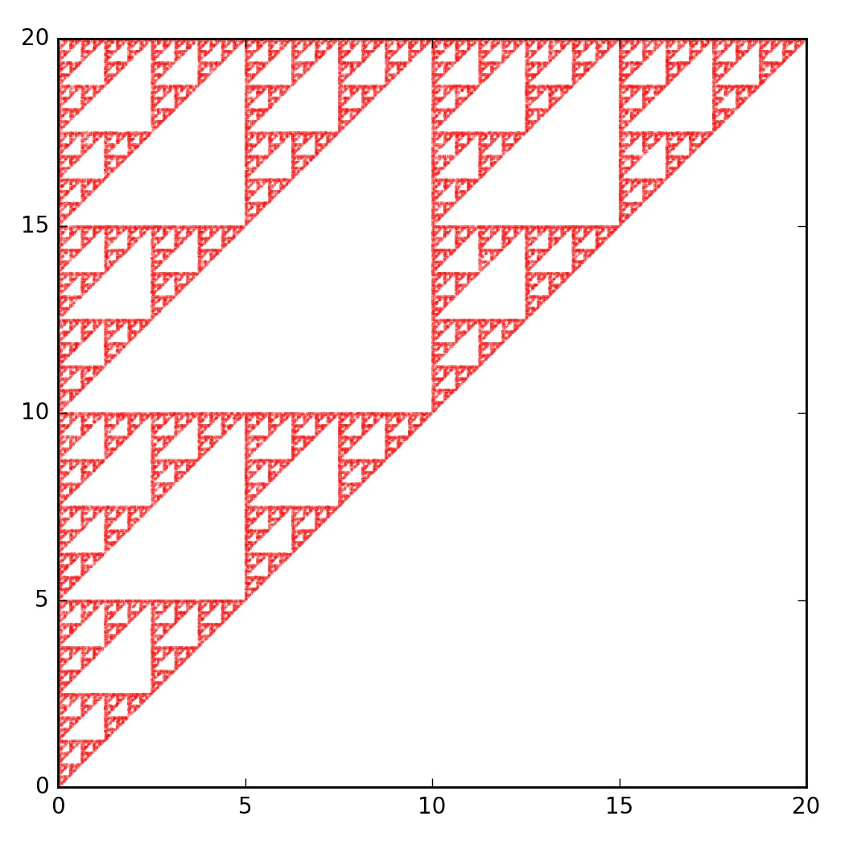
\[ \begin{array}{c}\left[ x\_{1/2} \\ y\_{1/2}\right]\end{array} = \frac{1}{2} \begin{array}{c}\left[ x\_p \\ y\_p\right]\end{array} + \frac{1}{2} \begin{array}{c}\left[ x\_C \\ y\_C\right]\end{array} \; . \]

By comparing this formal transformation with the rows of the table, it is obvious that the vertices of the triangle used in the chaos game are: A - (2,2), B - (2,100), C - (100,100). This is nicely confirmed by putting the axes back on the figures and foregoing the aesthetic transformation used last month.



The limits of the figure are [2,100] for the x-axis and y-axis and the triangle fits perfectly into the upper left half.

Further confirmation is obtained by changing the values of $$f$$ and $$g$$ for all three affine transformations so that the vertices are now located at: A - (0,0), B - (0,20), C - (20,20). Rerunning the chaos game yields the following fractal



Here the limits on the plot are [0,20] for both the x- and y-axes.

With a firm understanding of what the parameters $$f$$ and $$g$$ do, next turn attention to the matrix-portion of the transformation. Referring to the half-way point equation above, the first term on the right-hand side can be recast as

\[ \frac{1}{2} \begin{array}{c}\left[ x\_p \\ y\_p\right]\end{array} = \begin{array}{cc}\left[ 1/2 & 0 \\ 0 & 1/2 \right]\end{array} \begin{array}{c}\left[ x\_p \\ y\_p\right]\end{array} \; ,\]

thus explaining what the parameters $$a$$, $$b$$, $$c$$, and $$d$$ describe. For example, to get to point a third of the way along the line connecting the current point to the particular vertex, the table should be changed so that $$a = 1/3 = d$$ while keeping $$b=0=c$$.

Finally, the cumulative probabilities can be modified as well to create additional realizations of the game.

Next post will explore some of the various new fractals that can be obtained by fiddling with the matrix component of the affine transformation and the cumulative probabilities as a way of investigating the stability of the game to variations.