Last weekend I came upon an interesting article about generating random terrain using simple fractals. If you’ve played games like Minecraft, you’re already familiar with the results of these techniques: surprisingly natural landscapes that are significantly and interestingly different almost every time.

The article points out that fractal-based terrain is simple to implement and scales well because fractals are self-similar. A self-similar design is one which, if you isolate and magnify a small part, it will look like the larger whole. The Mandelbrot Set is a famous fractal that you’ve probably seen before. It looks something like a large circle with smaller circular polyps growing out of the perimeter. If you zoom in on any of those polyps, you will eventually find an exact image of the larger set. In short: each small part contains the whole.

I wonder to what extent the principle of self-similarity applies to stories? What kind of output would we get from a fractal-based story generation algorithm?

The article is dedicated to the Diamond Square Algorithm. It’s a little too complicated to reproduce in this blog post, but the introduction eases you in by way of a simpler predecessor: the Midpoint Displacement Algorithm. Imagine a straight line. Now, take the midpoint of that line and move it a random amount upward or downward. Repeat that process for the left and right halves of the line: find the midpoint and move it randomly up or down. Keep dividing the line segments in half and repeating this process, and eventually you get something that looks like the silhouette of a mountain range off in the distance. This basic principle can be extended to 3 dimensions to get a 3D landscape.

The mountain range silhouette produced by this algorithm is self-similar. If you zoom in on any one part, it will have the same basic properties as the whole. They won’t look exactly the same because of the random movements, but essentially each peak of the mountain contains an entire mountain range in itself. Now, how is this relevant to stories?

Firstly, these self-similar algorithms are very simple to implement. Obviously, no story generation system that hopes to cover a wide variety of human narratives is ever going to be simple, but we might be able to push the complexity down into the low-level realization of the story while keeping the high-level design simple. Imagine using midpoint displacement to create the conflict arc of a story—natural rises and falls in intensity as the story progresses.
The Midpoint Displacement Algorithm can be seeded with values to define its basic shape. In other words, you could start with a slanted line or a curve rather than a straight one. Freytag’s Triangle seems like a good place to start for a fractal-based intensity arc.

More importantly, fractals scale well. Novels are made up of chapters, chapters of scenes, etc. A fractal-based story generator could maintain a single theme or basic structural element throughout the story at large and each constituent part (with enough random variation to avoid stagnation).

More complicated fractal generation methods, like the Perlin noise heightmaps used in Minecraft, allow you to sample any part of the space at any time. Chunks of a Minecraft map are generated dynamically as you explore them, but they are guaranteed to fit together seamlessly no matter what order you visit them in. Imagine creating one single, unified story for the entire community of an MMORPG. This task would be daunting for even an army of human writers, but if the story generation was guided by a fractal, perhaps a truly massive story could be generated that had a single theme, while still maintaining the property that each local element was consistent with its neighbors and fit seamlessly into the whole.

Fractals are useful for terrain generation because they produce realistic looking landscapes. Their usefulness to story generation will hinge on whether or not they can produce natural and realistic storyscapes. As story generation research continues to progress, I will be interested to see what role fractals have to play.