Daniel Mattheiss

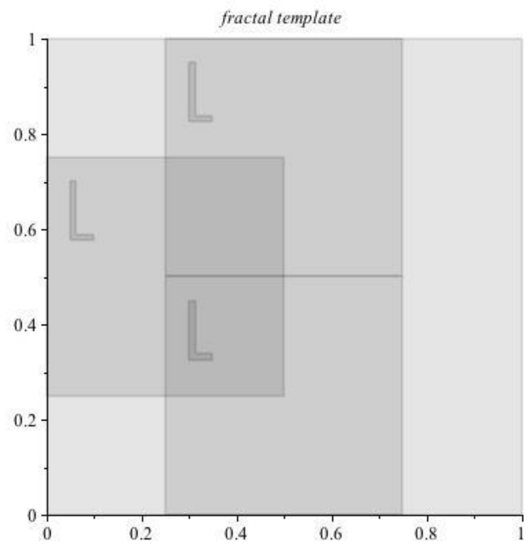
4/1/2016

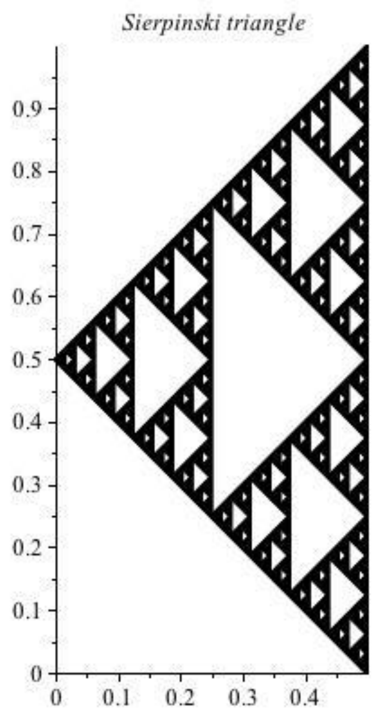
Linear Algebra 2270-2

Generating Fractals by Iterated Contraction Mappings: An Exploration of Affine Transformations

Computer-generated fractals can be created quite easily in Maple given a complete understanding of iterated contraction mappings and their uses in fractals. Thanks to late 20th century progress in technology and fractal theory, a near-infinite amount of different fractals can be generated in Maple by modifying the affine transformations that define the contraction mappings. In this report I clarify the effects of various affine transformations and explore some unique fractals, explaining clearly the steps and results for each fractal generation.

To understand how affine transformations affect fractal patterns, one must understand how fractals are generated. Given a set of transformations that maps points from one coordinate location to another, we recursively call a function 'S' which is the union of the set of transformations. 'S' is called (in an iterative fashion) until a shrinking threshold is reached, where further recursion would be unnoticeable to the human eye. From this definition, it is easier to understand the relationship between a standard affine map and the resulting fractal. For example, applying some simple translations and scalar transformations to the three-function Sierpinski's triangle results in a map and corresponding fractal:

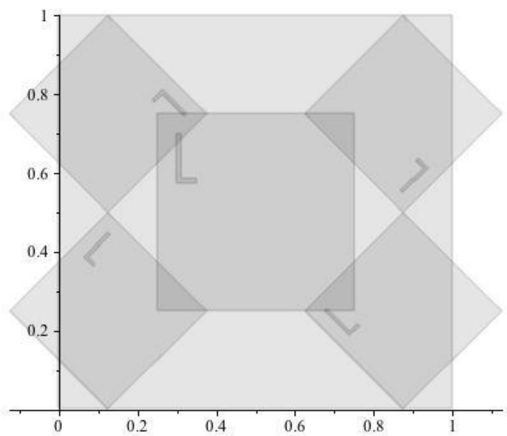




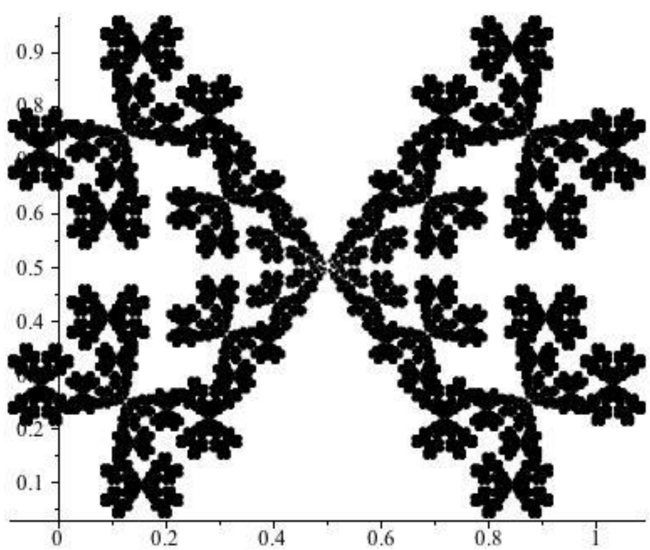
Rotating the L-boxes by 90 degress and compressing the L-boxes leads to this squashed, horizontally-oriented Sierpinski triangle in which recursive mapping can easily be seen.

Furthermore, rotational transformations can be used in the mapping functions. In my take on the cross-type example, I rotate each L-box and adjust sizes to generate an x-type fractal with a central element:

fractal template



Sierpinski triangle



I only iterate 7 times as Maple slows with further iterations. However, it is easy to note the iterative nature of this fractal by observing the overall shape and the final recursions for each leaf.

Another type of affine transformation is a shear, where L-box are transformed into a rhombus-like structure. This, combined with rotational transformations, leads to some pretty interesting results. In the following example I overlay a rotated L-box with a translated x-sheared

L-box and create a loop fractal that is common in nature (I think it looks similar to a scorpion tail).

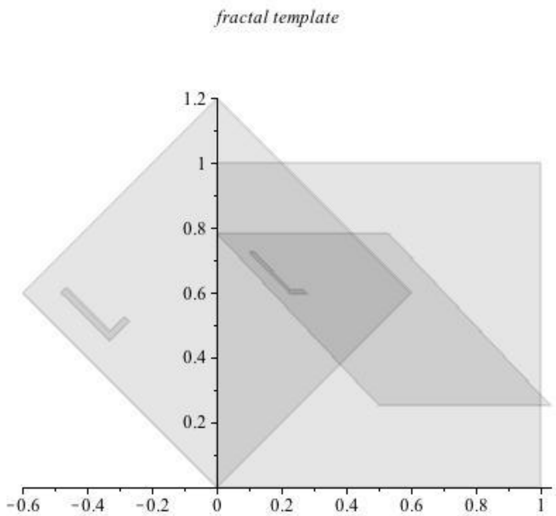


Figure 5.16 page 95 Lauwerier

These examples represent only a fraction of the possibilities available with computer-generated, iterative mapping fractals.