Iterated Function Systems

CS 418

Interactive Computer Graphics

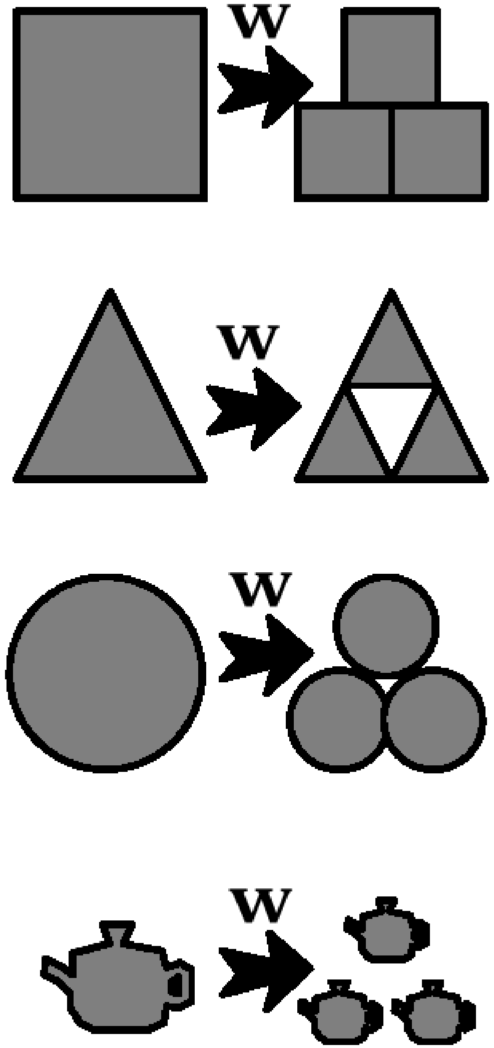
John C. Hart

Iterated Function System

- An IFS is a set of N contractive affine transformations $\{w_i\}$
- Example: Three tranformations that scale by ½ and translate halfway to the vertices of an equilateral triangle
- Hutchinson operator

$$\mathbf{w}(X) = \cup w_i(X)$$

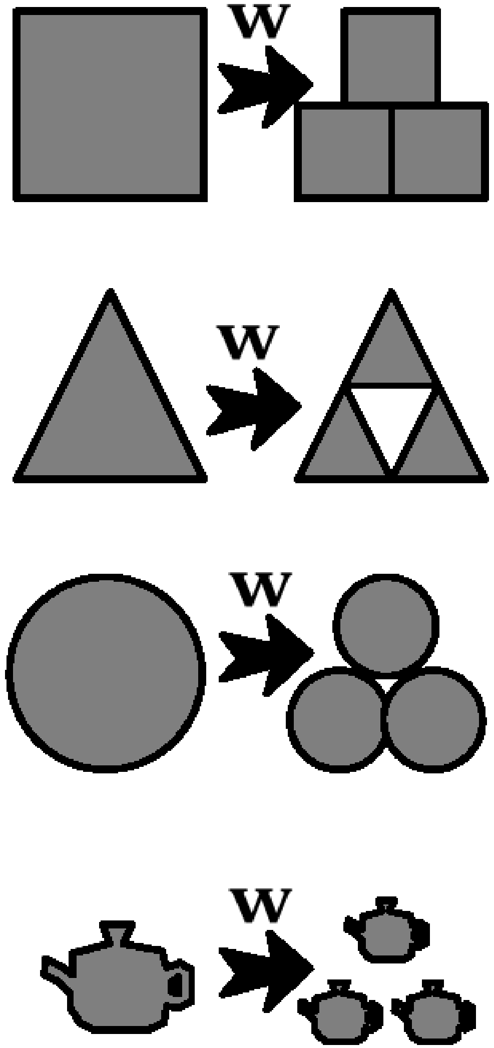
- Hutchinson operator applies the IFS to a shape to generate a new shape



Attractor

- The IFS describes a unique set called an attractor *A*
- *A* is invariant

$$A = \mathbf{w}(A)$$



- *A* is attractive

$$A = \lim_{n\to\infty} \mathrm{w}^{\mathrm{o}n}(B)$$

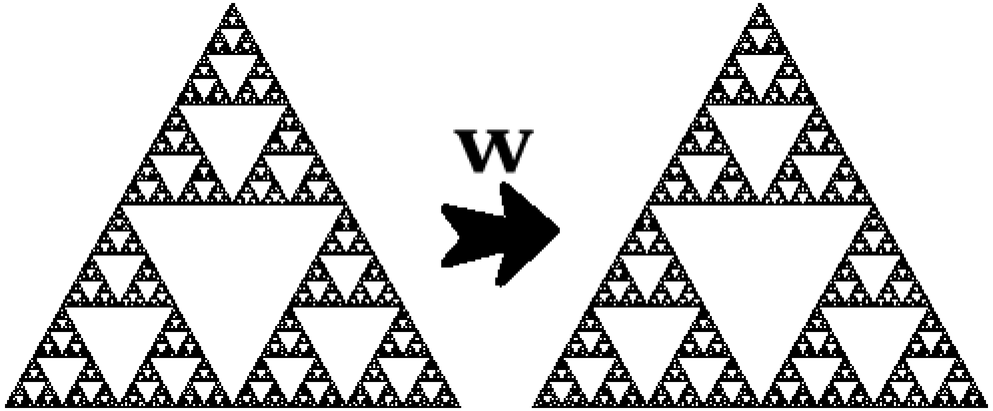
for any B*

* non-empty, compact

Attractor

- The IFS describes a unique set called an attractor *A*
- *A* is invariant

$$A = \mathbf{w}(A)$$



- *A* is attractive

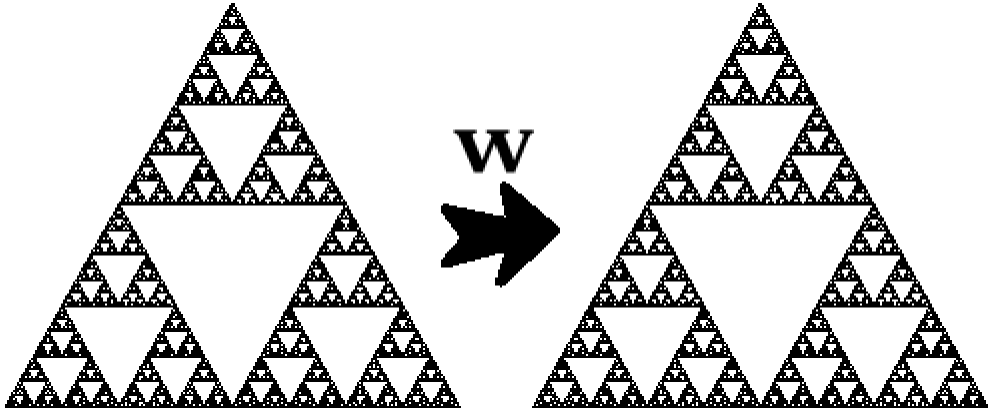
$$A = \lim_{n \to \infty} \mathrm{w}^{\mathrm{o}n}(B)$$

for any B*

* non-empty, compact

Attractor

- The IFS describes a unique set called an attractor *A*



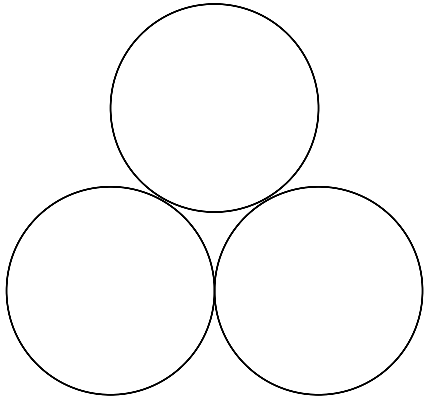
- *A* is invariant

$$A = \mathbf{w}(A)$$

- *A* is attractive

$$A = \lim_{n\rightarrow\infty} \mathrm{w}^{\mathrm{o}n}(B)$$

for any B*

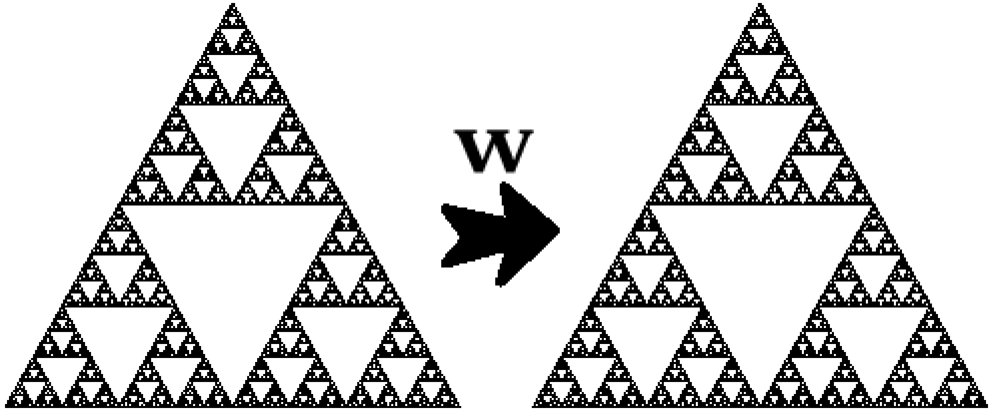


* non-empty, compact

Attractor

- The IFS describes a unique set called an attractor *A*
- *A* is invariant

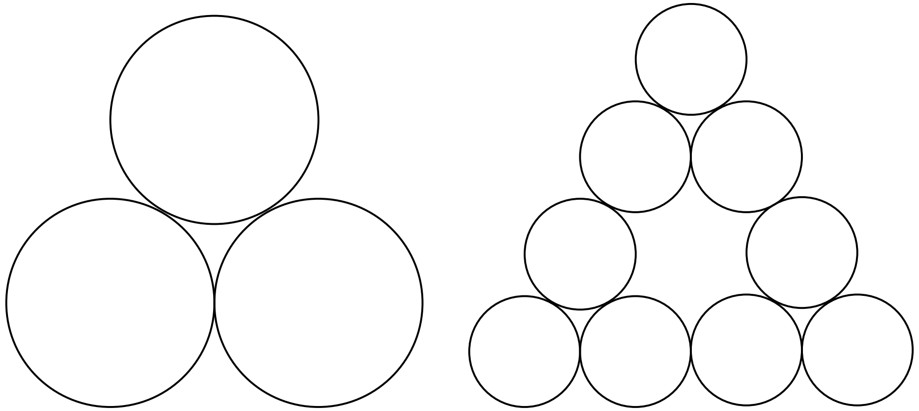
$$A = \mathbf{w}(A)$$



- *A* is attractive

$$A = \lim_{n\rightarrow\infty} \mathrm{w}^{\mathrm{o}n}(B)$$

for any B*

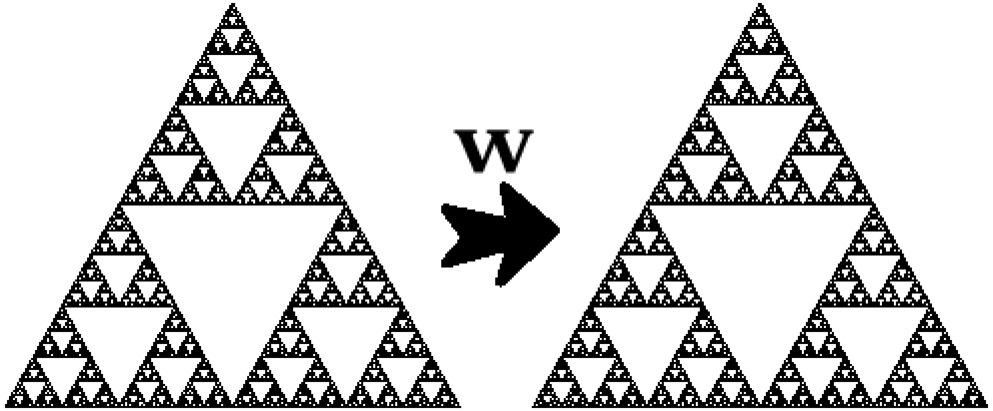


* non-empty, compact

Attractor

- The IFS describes a unique set called an attractor *A*
- *A* is invariant

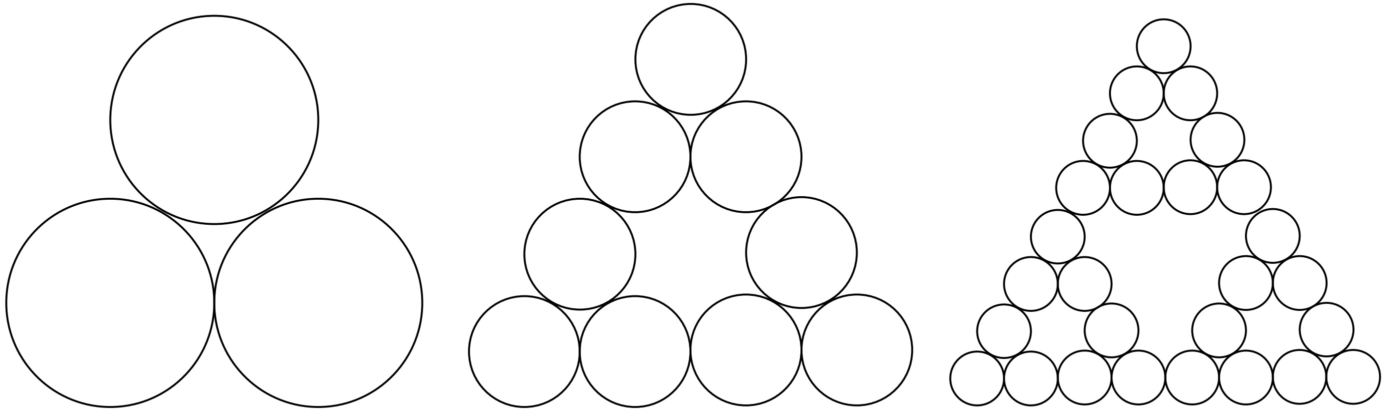
$$A = \mathbf{w}(A)$$



- *A* is attractive

$$A = \lim_{n \rightarrow \infty} \mathrm{w}^{\circ n}(B)$$

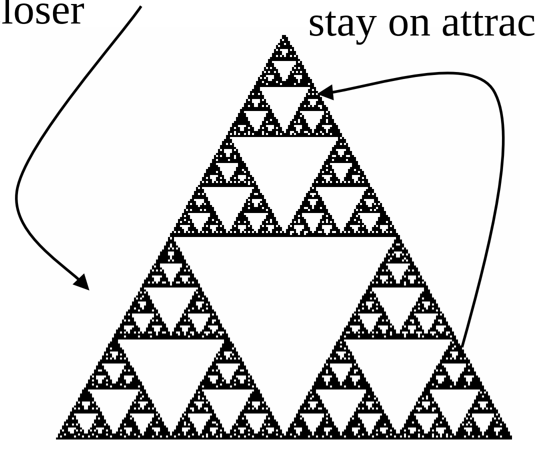
for any B*



* non-empty, compact

Chaos Game

- Iterating random IFS maps on a point on the attractor yields more points on the attractor
- These points plot the attractor
- Can start with any point
  - Will converge quickly to points “on” (extremely near) the attractor
  - Throw away the first 10 points
- Distribution skewed
  - Choose maps with probability inversely proportional to their contractivity

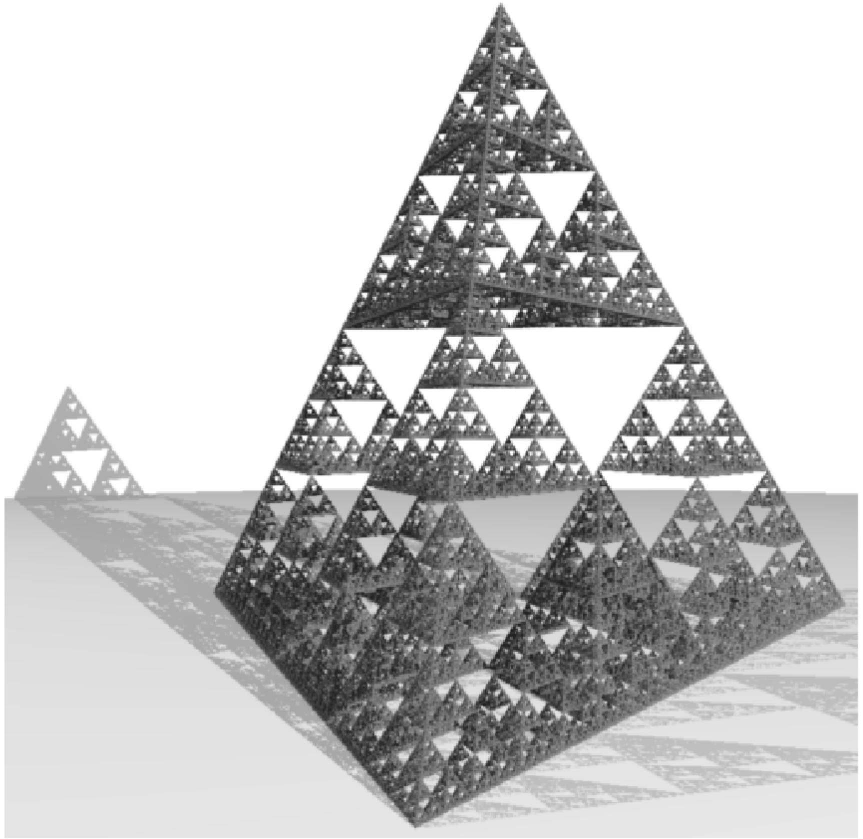


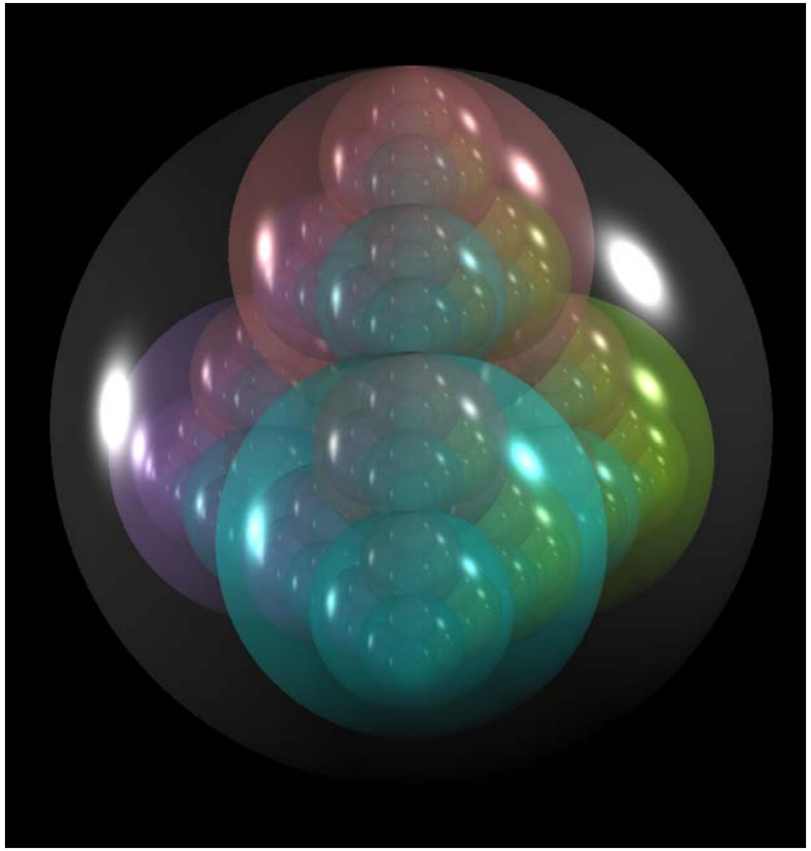
Points off attractor get closer

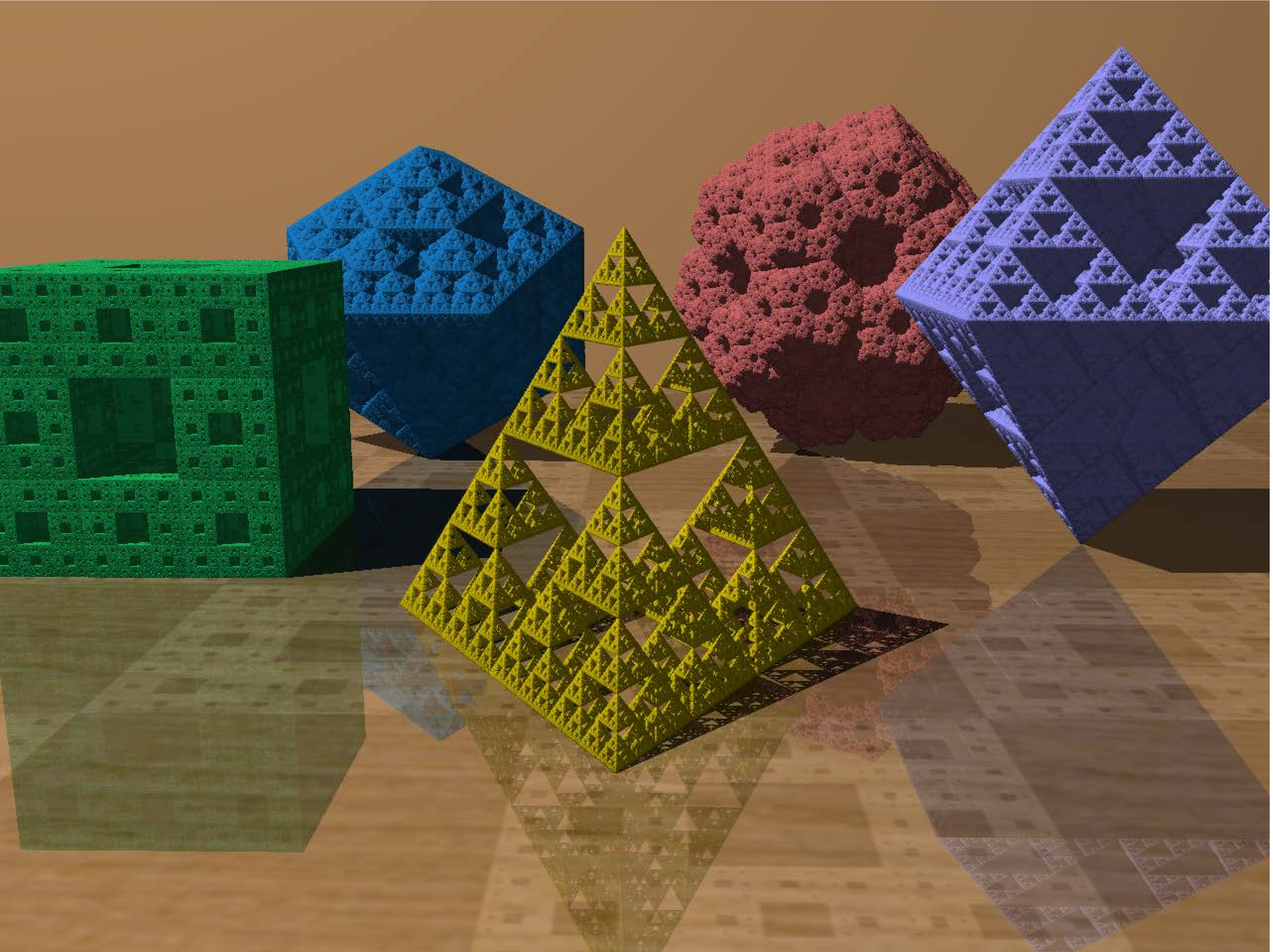
Points on attractor stay on attractor

```
x = (0,0)
for (int i = 0; i < 10; i++)
    x = w[randint(N)]*x
while (true)
    plot(x = w[randint(N)]*x)
```

3-D IFS

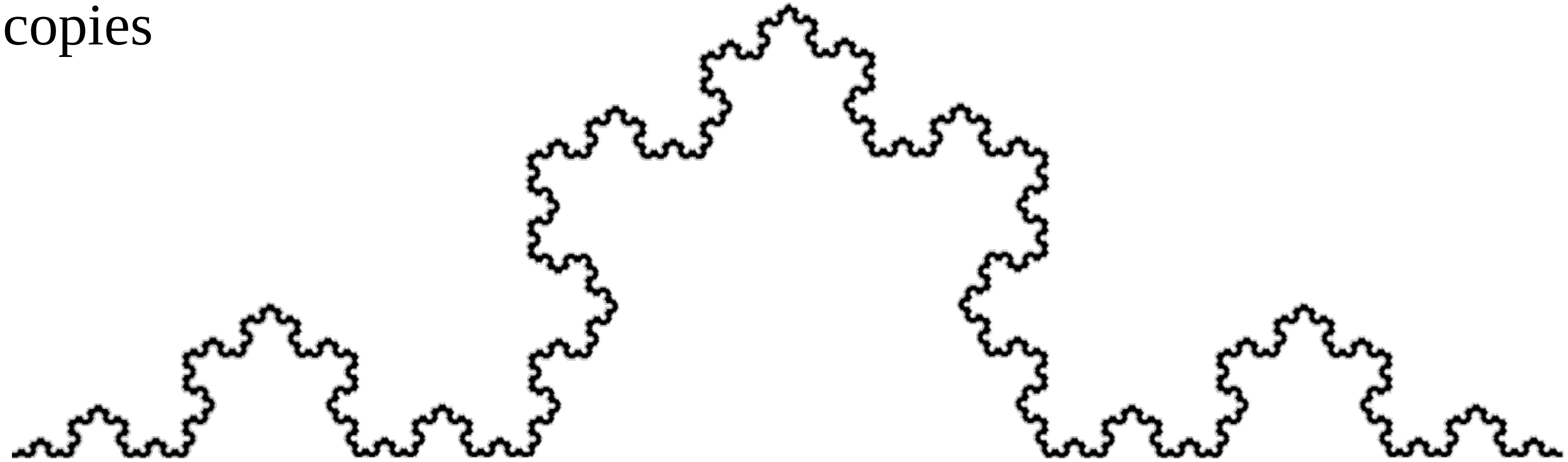






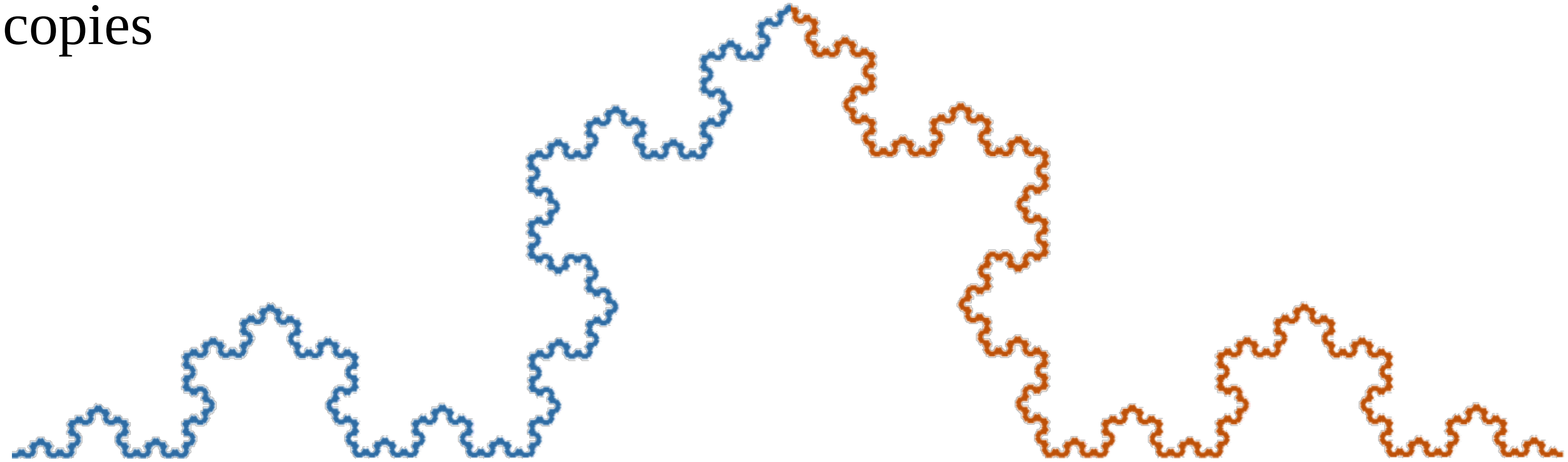
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



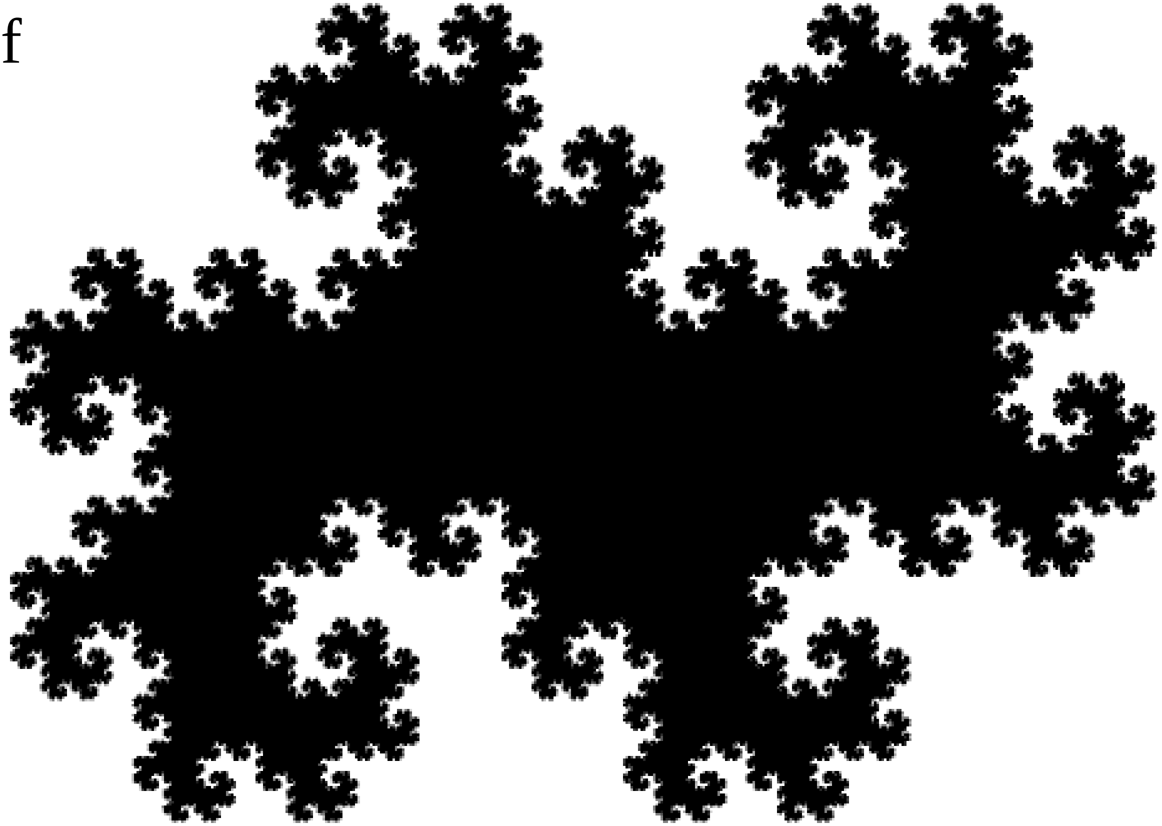
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



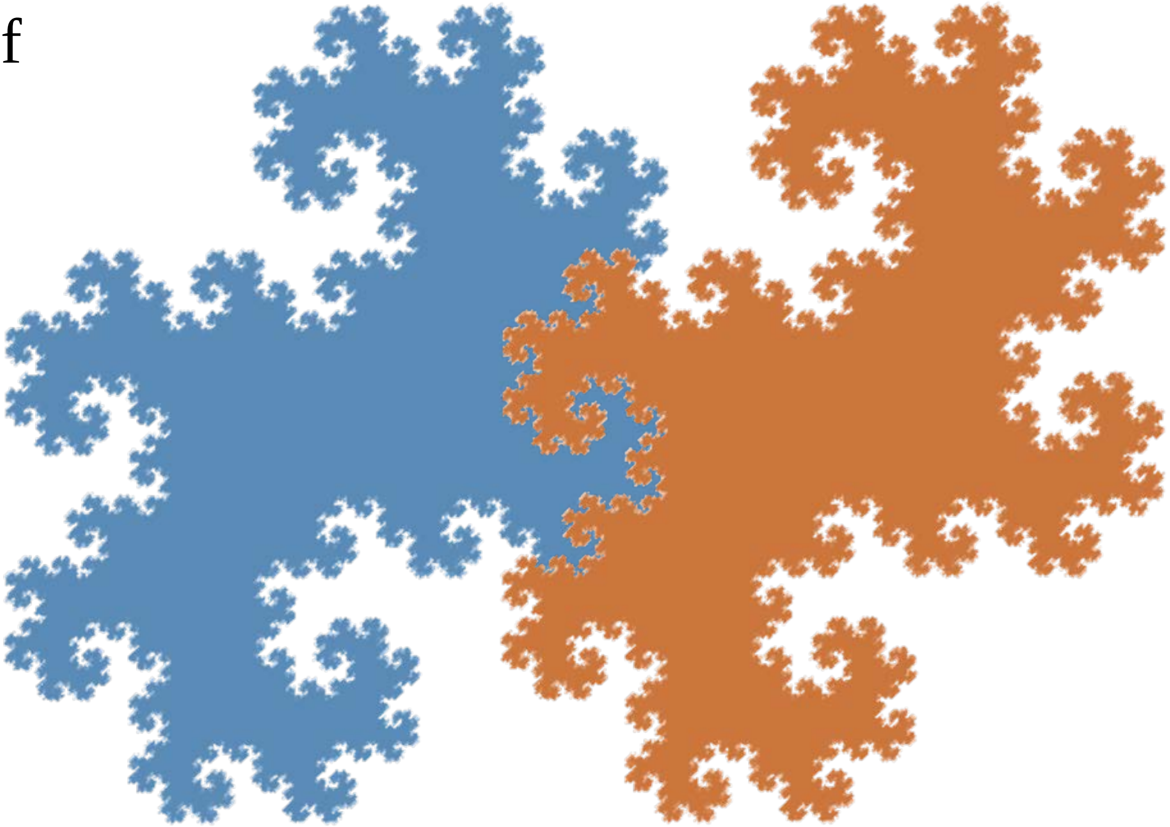
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



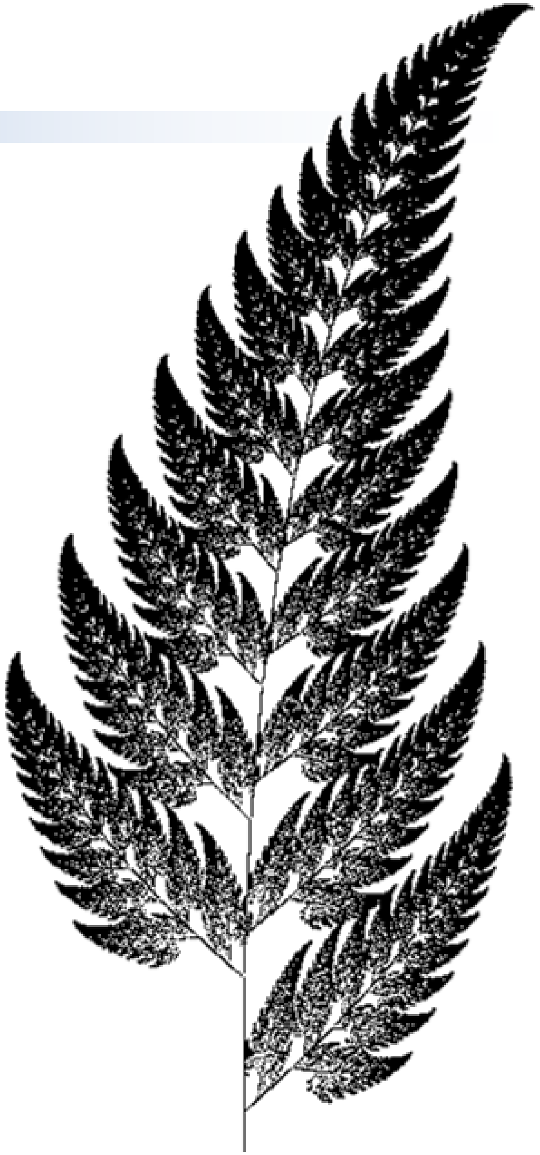
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



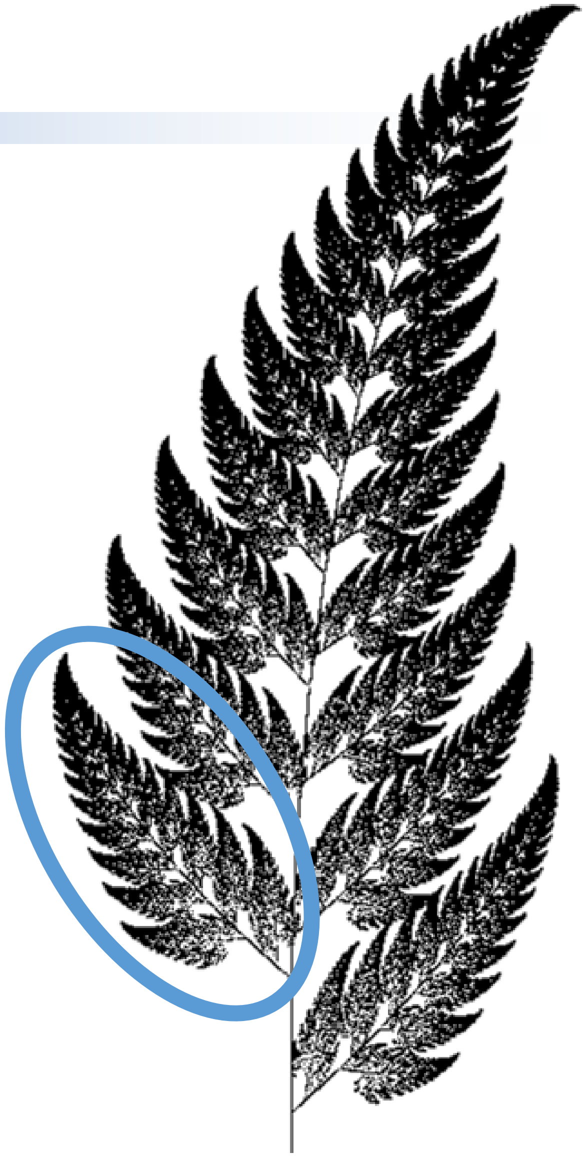
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



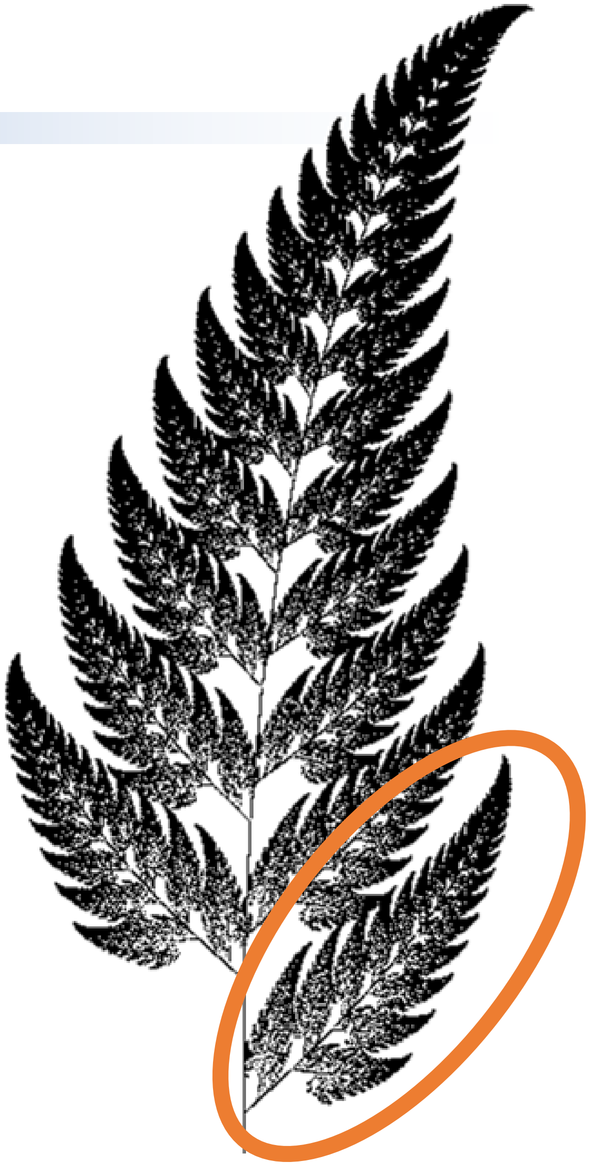
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



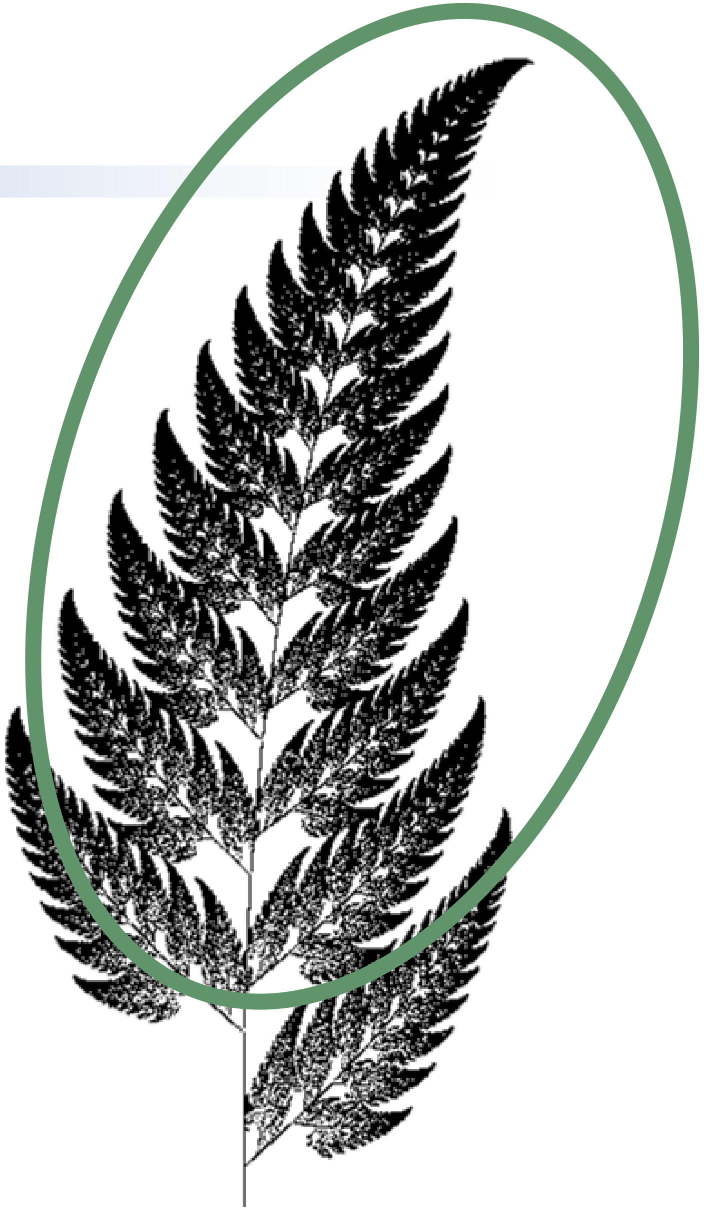
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



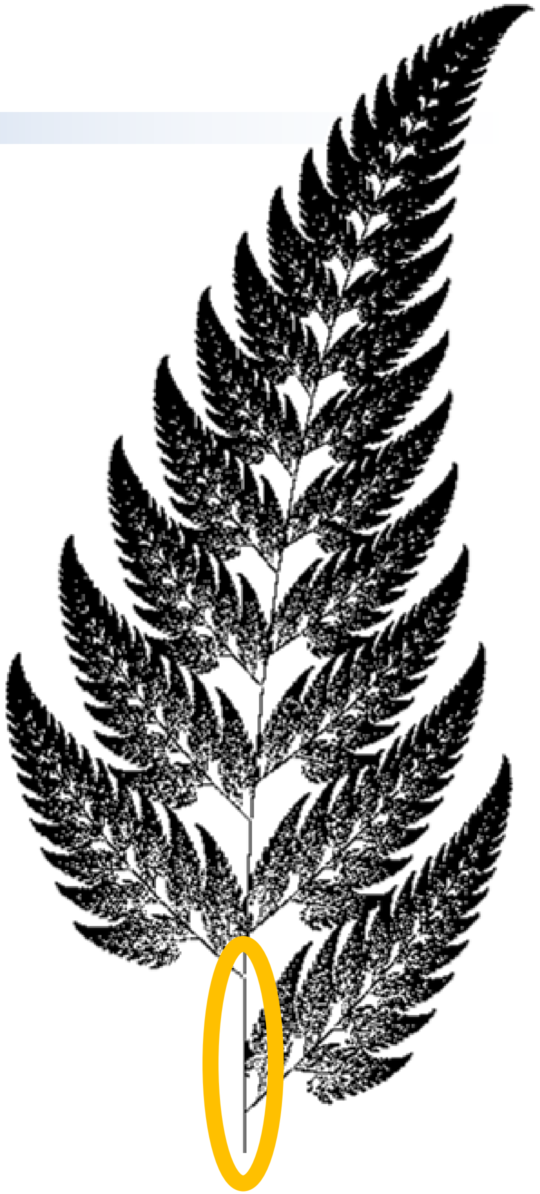
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



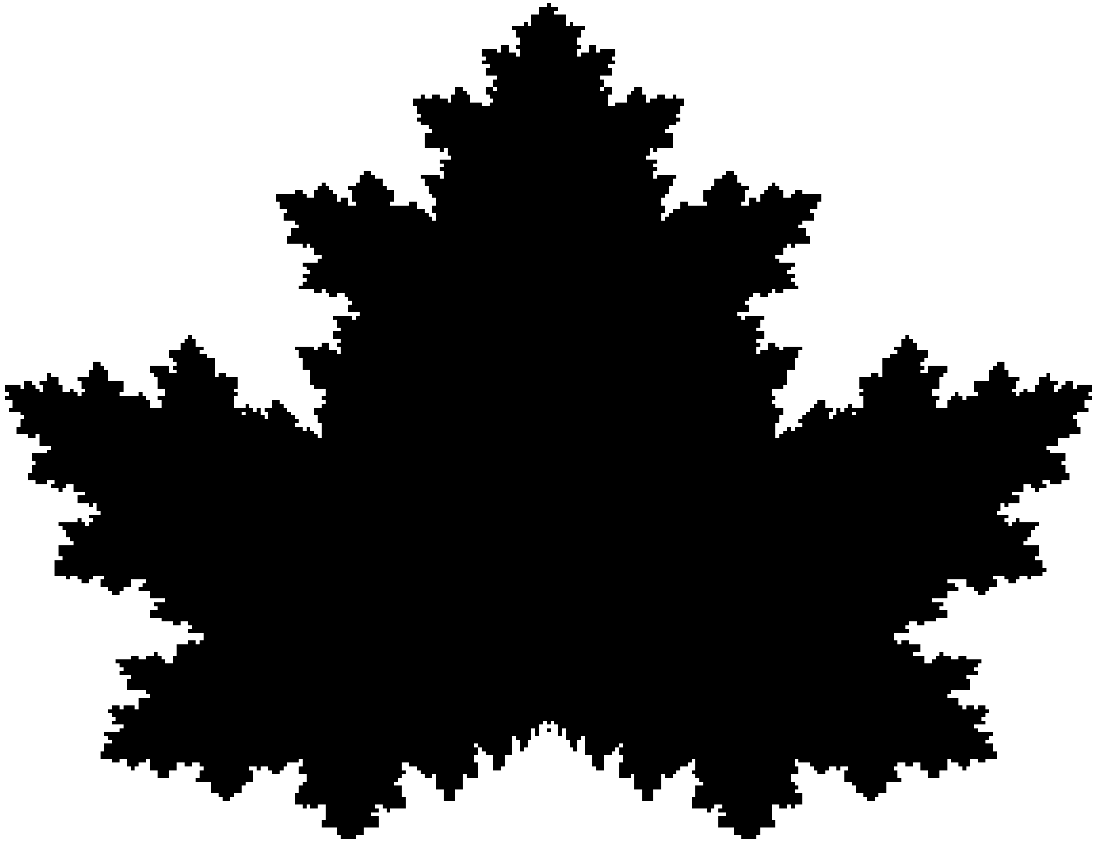
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



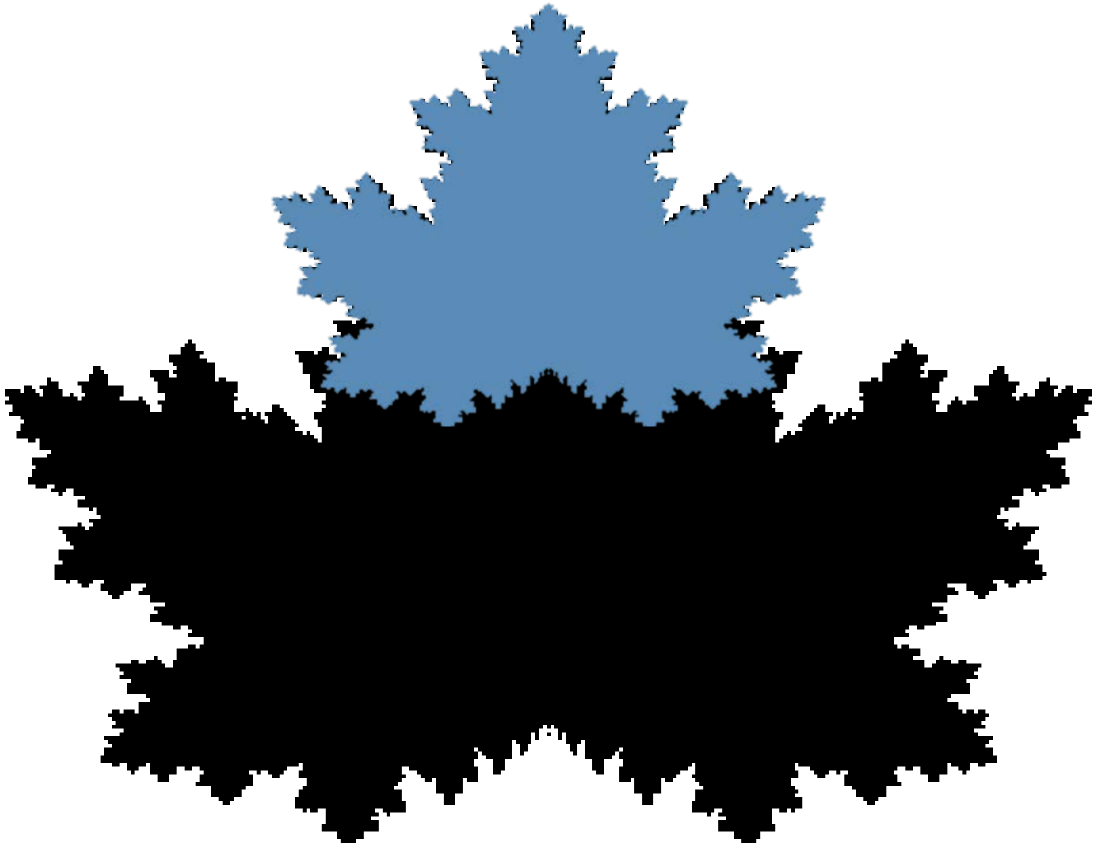
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



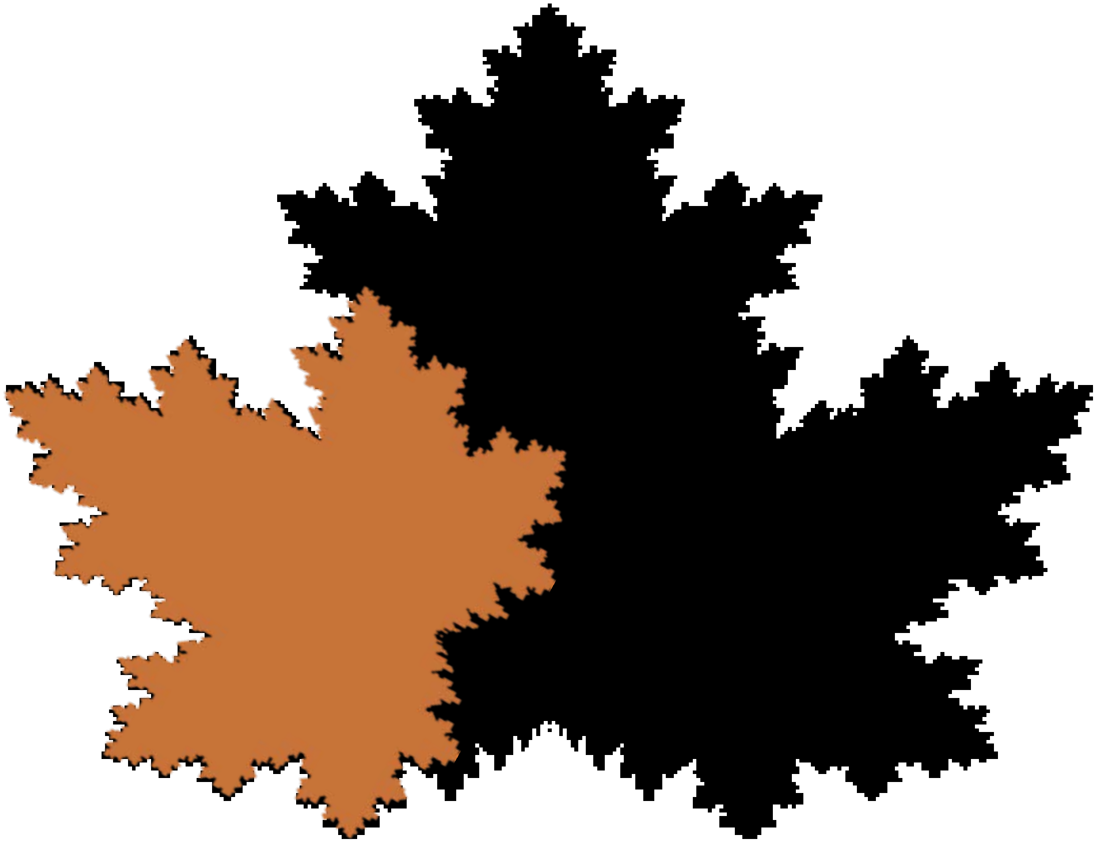
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



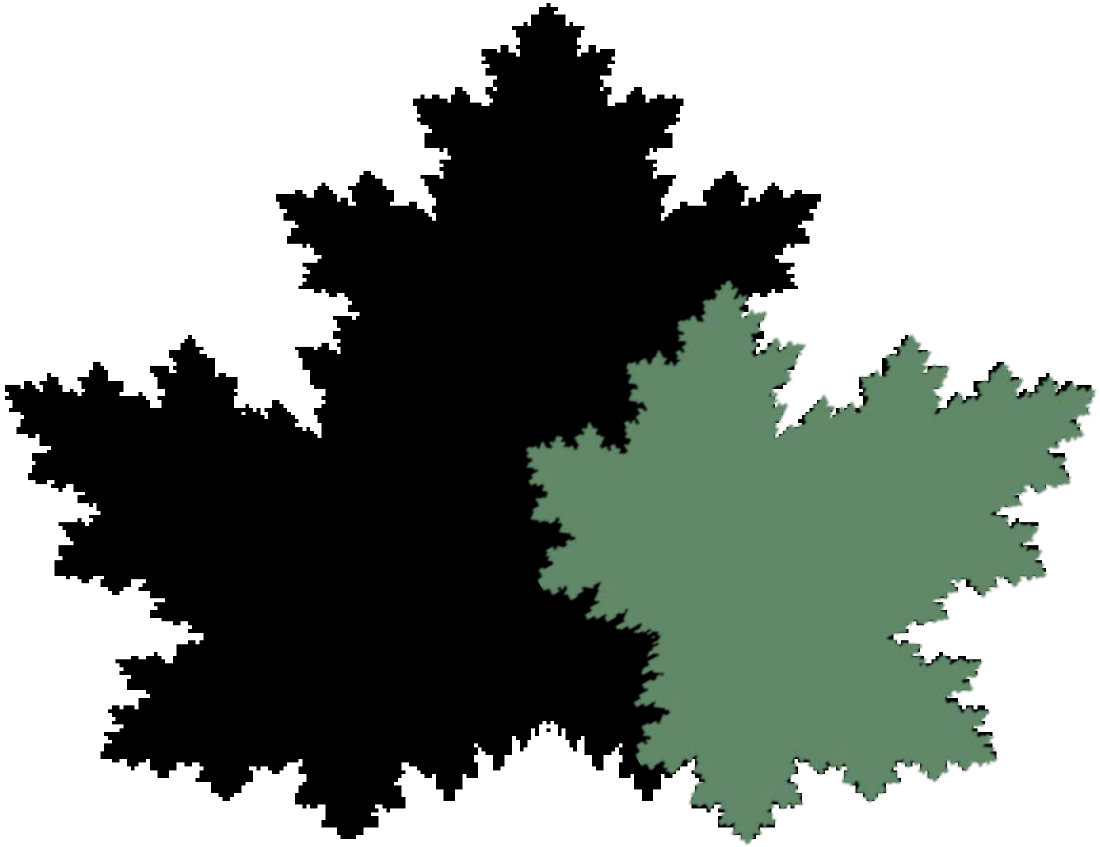
Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies



Designing Shapes

- IFS models a shape out of smaller copies of itself
- Attractorlets: $A_i = w_i(A)$
- The transformations of the IFS take the shape to each of the smaller copies