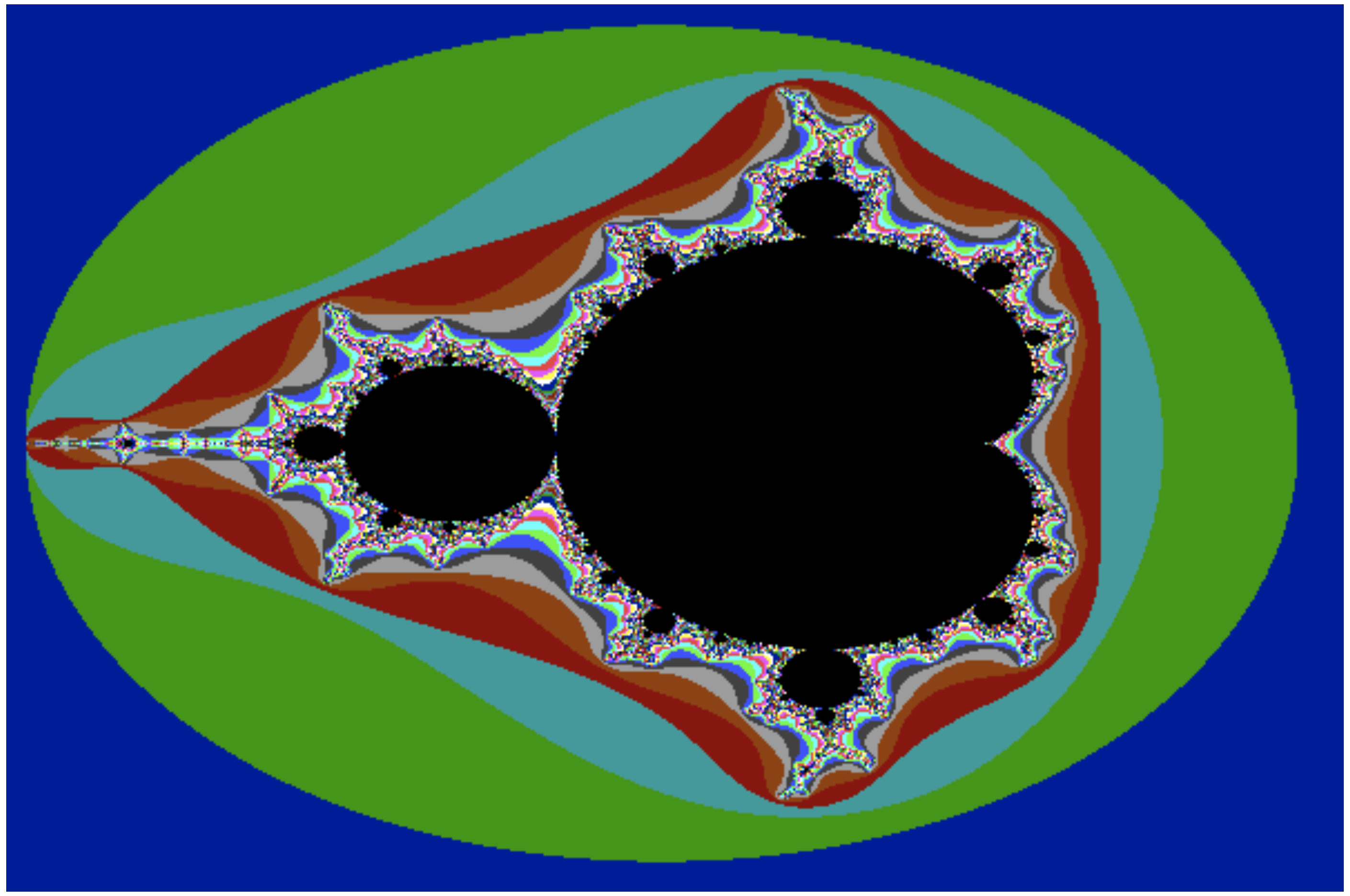
Fractals

Creating complex and interesting shapes from code

Most famous fractal: Mandelbrot set

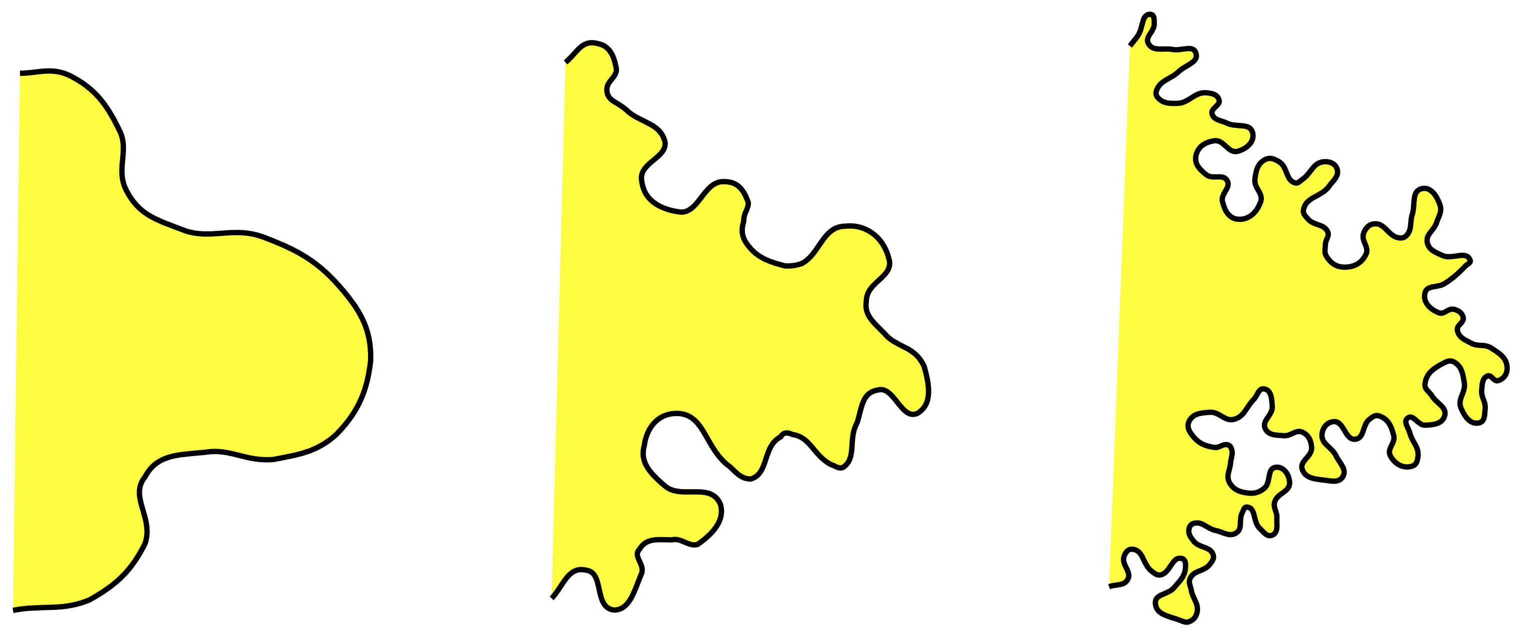


What is it, more than a pretty image?

Natural objects have fractal features

Classic example: Coastline

Shape and length varies with resolution



Classic example 2: Bracken

Self-similar, variable scale

Fractals in computer graphics

Fractals are shapes with:

- self-similarity
- infinite resolution

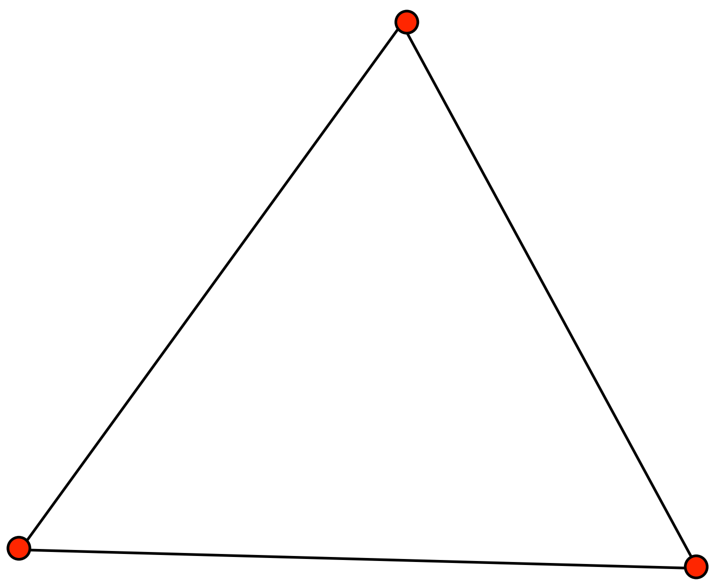
Used for modelling such shapes

Classification of fractals

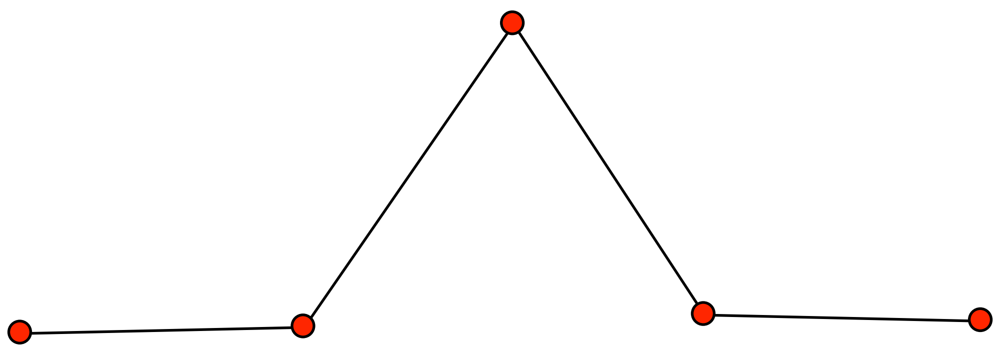
- geometrical recursive construction
- stochastic fractals
- mathematical formulas (in the complex plane)

Geometric construction of self-similar fractals

Example: Koch curve

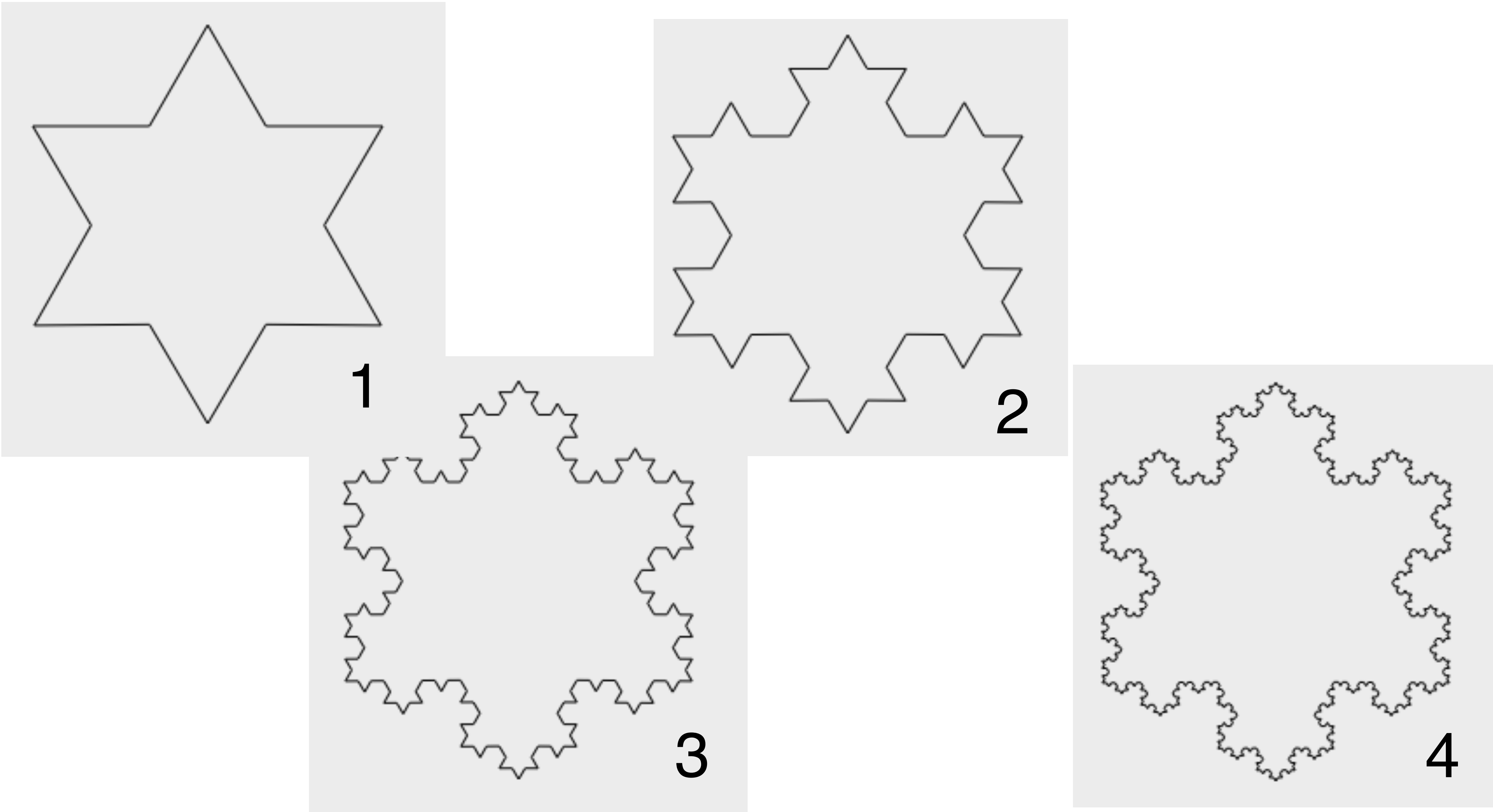


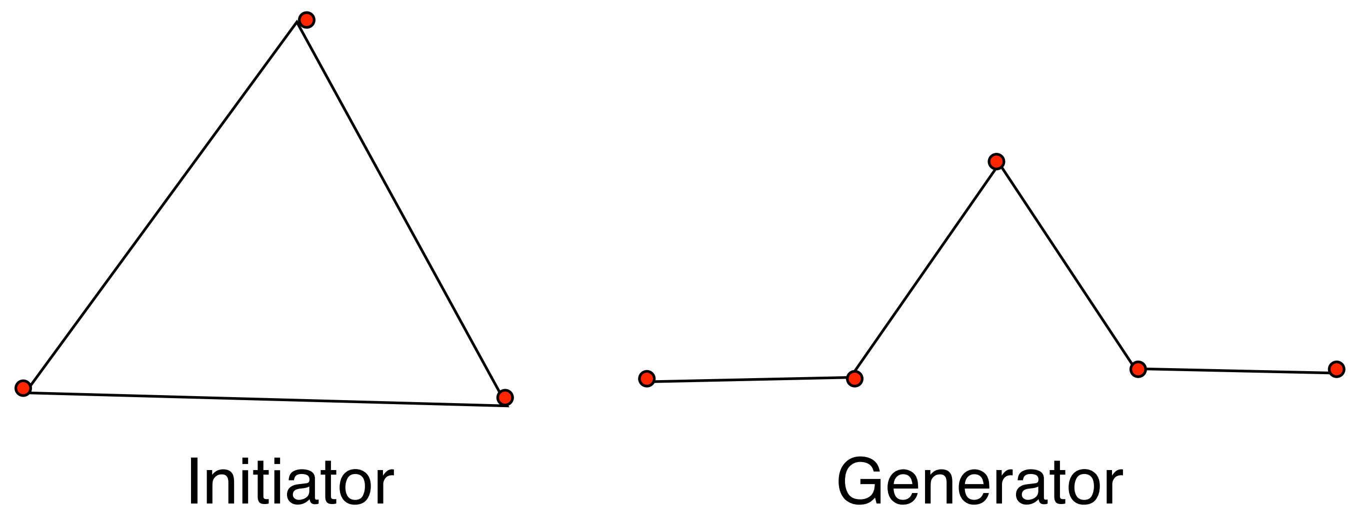
Initiator



Generator

Resulting Koch curves





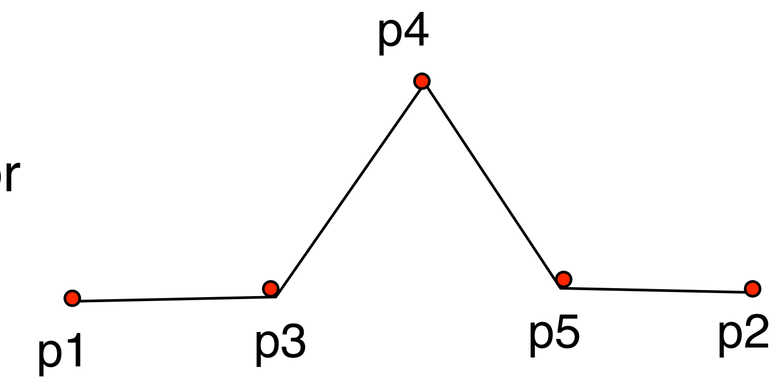
Recursive function

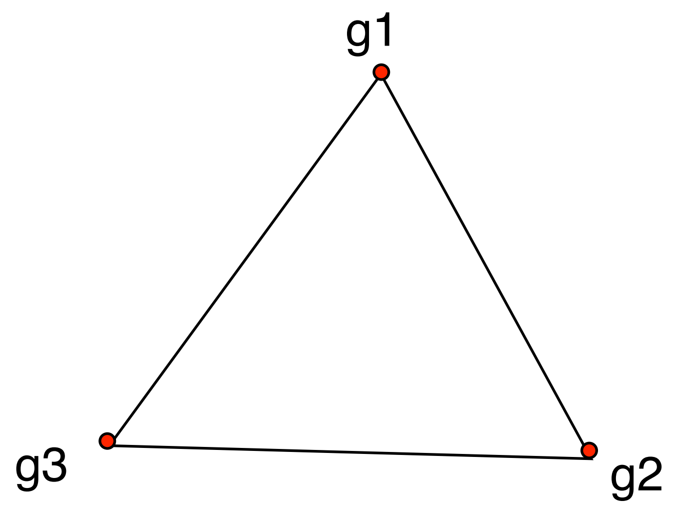
Pass all parts to next level

Replace part with the generator, *scaled to same length*

Stop at desired recursion depth or when sections are small enough (e.g. 1 pixel long)

```
procedure DrawKoch(p1, p2, depth)

if depth >= maxDepth then

    MoveTo(p1)
    LineTo(p2)
    return

else

    calculate p3, p4, p5 as the three points inside the generator

    DrawKoch(p1, p3, depth+1)
    DrawKoch(p3, p4, depth+1)
    DrawKoch(p4, p5, depth+1)
    DrawKoch(p5, p2, depth+1)


main procedure:

Choose three generator points, g1, g2, g3

DrawKoch(g1, g2, 0)
DrawKoch(g2, g3, 0)
DrawKoch(g3, g1, 0)
```





Fractal dimension

A measure of how rough or fragmented the shape is

Definition:

$$ns^D = 1$$

n = number of subparts

s = scaling

D = fractal dimension

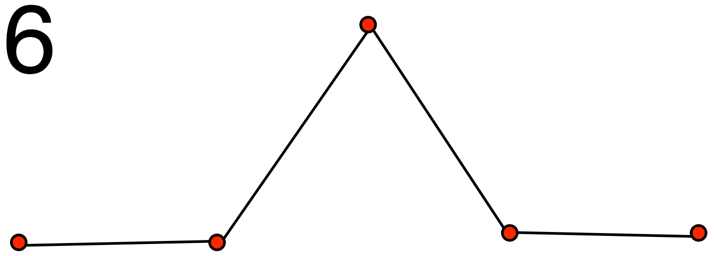
Solves to $D = \ln(n) / \ln(1/s)$

Fractal dimension example: Koch curve

$n = 4$

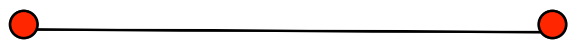
$s = 1/3$

$D = \ln 4 / \ln 3 = 1.26$

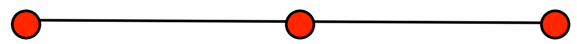


Fractal dimension example: Splitting a line

=>



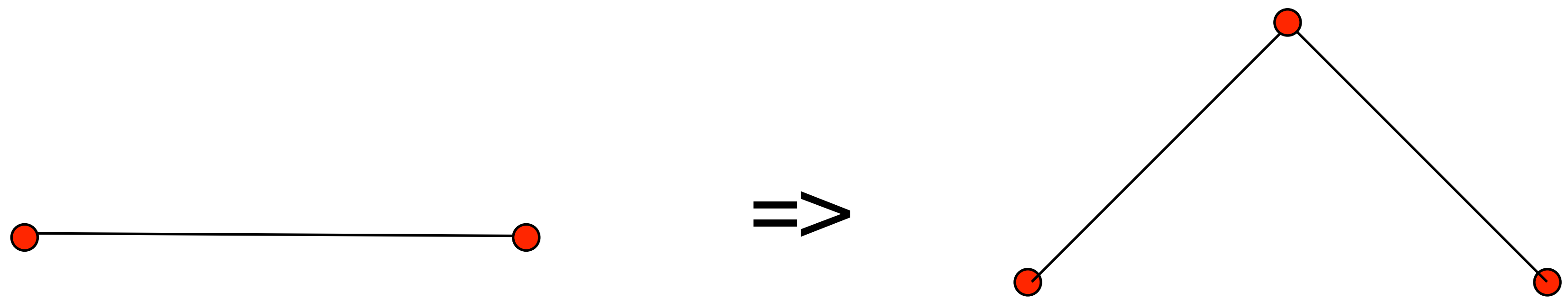
$n = 2$



$s = 1/2$

$D = \ln 2 / \ln 2 = 1$

Fractal dimension example: Splitting a line and moving midpoint



$$n = 2$$

$$s = 1/\sqrt{2}$$

$$D = \ln 2 / \ln \sqrt{2} = 2$$

Fractal dimension:

In 2D:

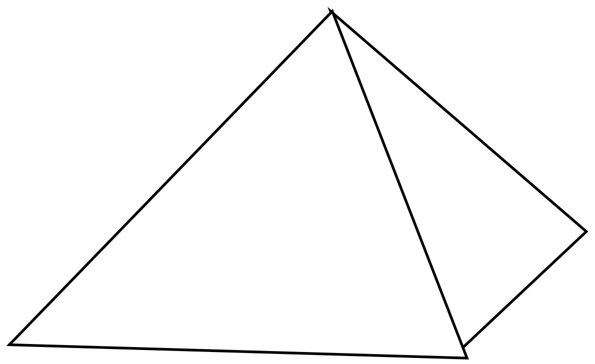
1 to 2: Well-behaved fractal curve

>2: Self-intersecting, area-covering

Split line: D = 1 minimum, no fractal
Koch: D = 1.26, moderate fractal
Moved midpoint: D = 2, maximum

Geometric construction of self-similar fractals in 3D

Example

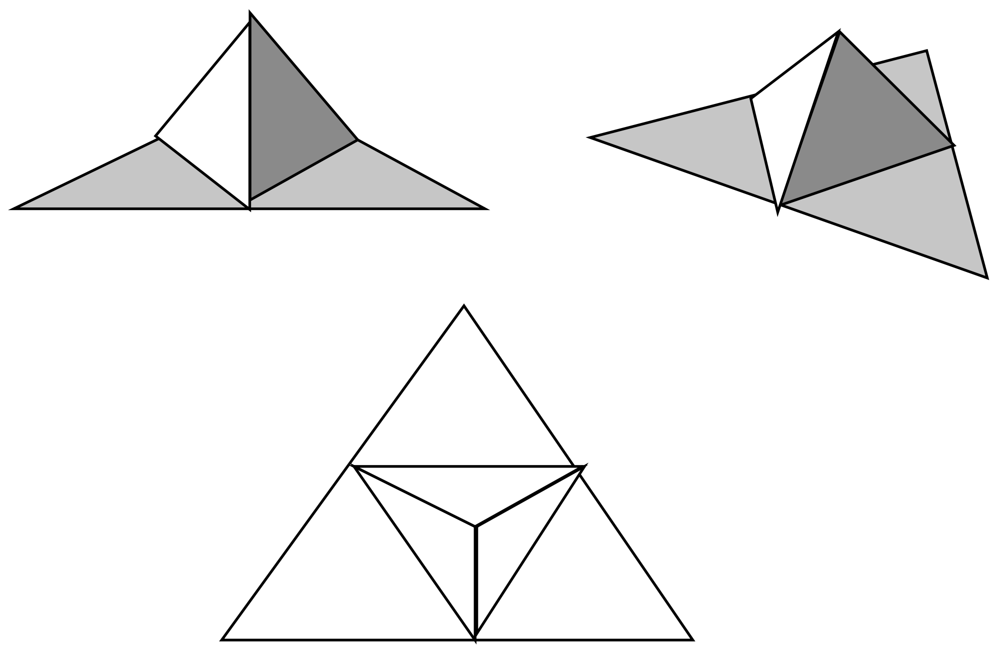


Initiator

$n = 6$

$s = 1/2$

$D = \ln 6 / \ln 2 = 2.58$



Generator

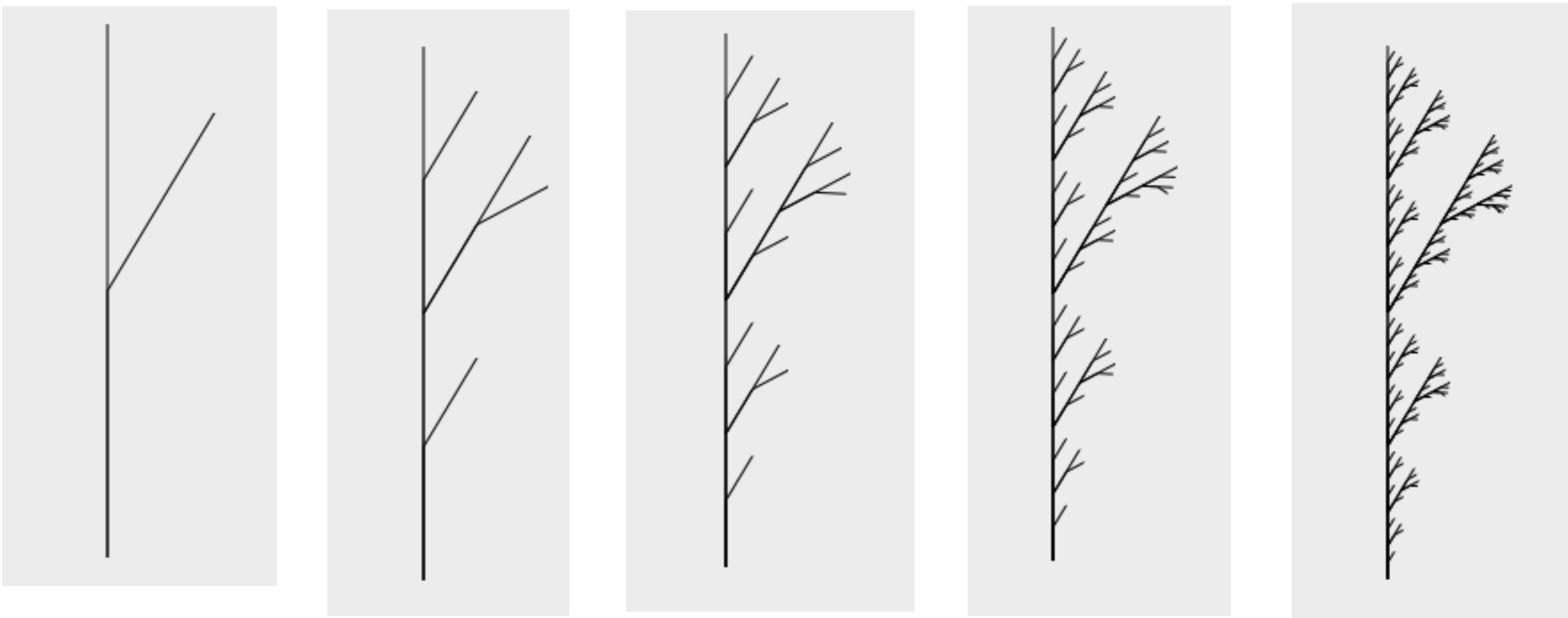
Interpretation of fractal dimension:

In 3D:

2 to 3: Well-behaved fractal surface

>3: Self-intersecting, volume-covering

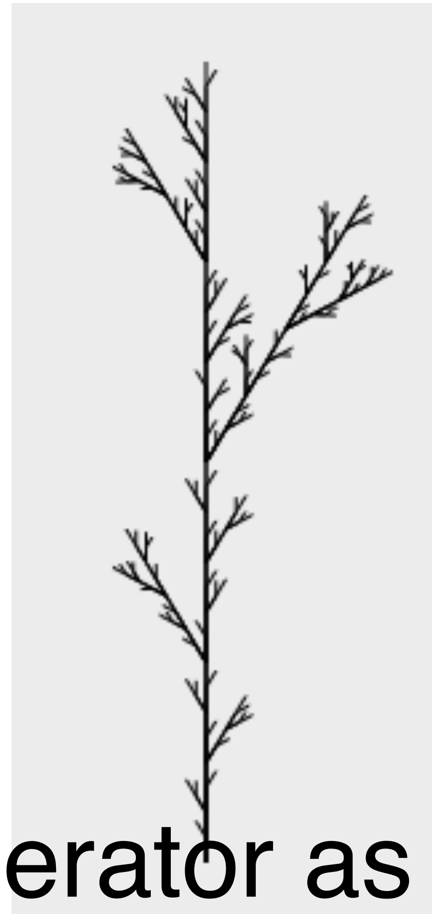
Example: Generation of plants



Promising, but too self-similar!

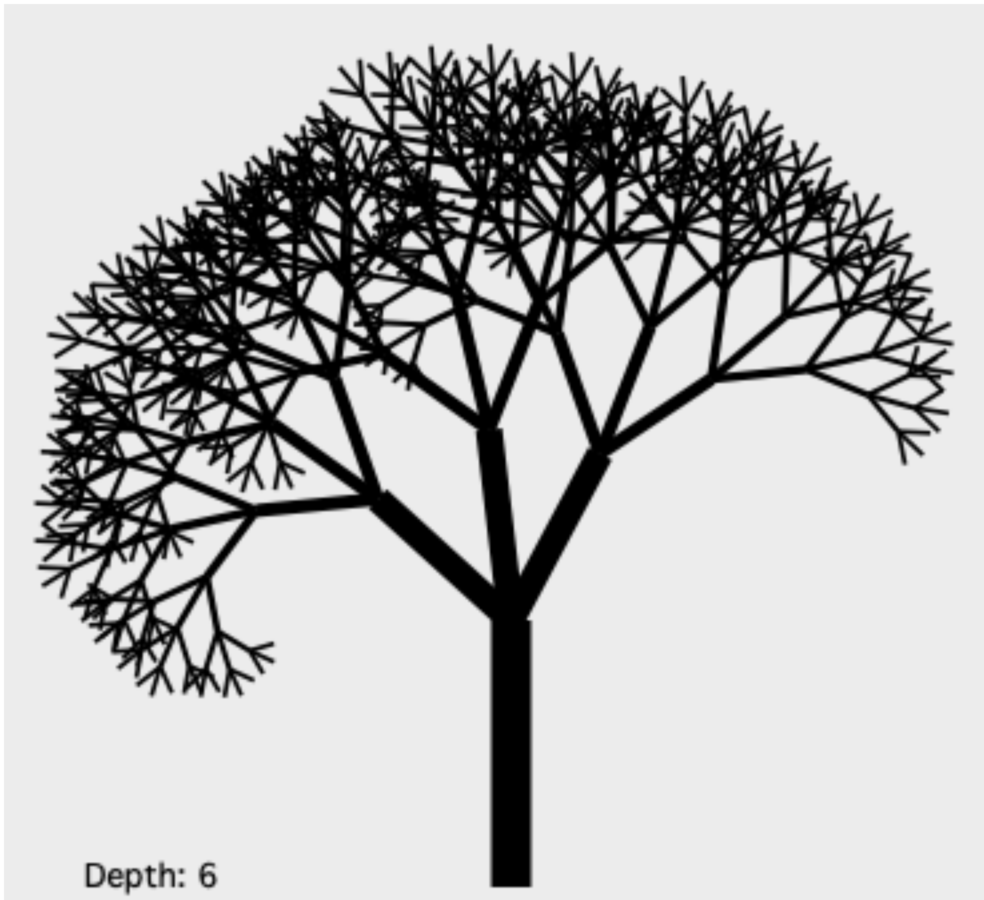
Statistically self-similar fractals

Random variation of generator



Same branch generator as before, with some randomness!

Example: Generation of plants #2



Related methods:

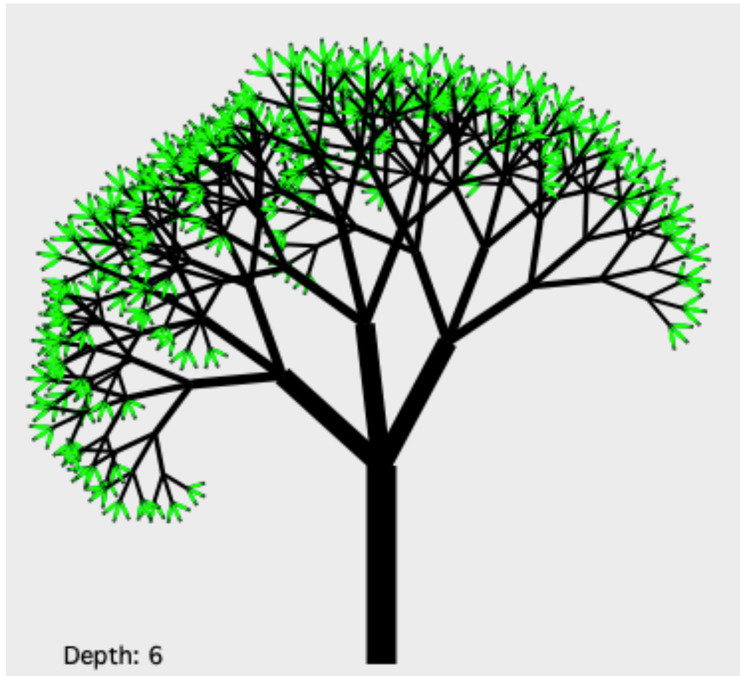
Shape grammars and procedural methods

No unlimited resolution

Different rules at different levels

Example: Tree with leaves: replace last iteration with leaf generator

"graftals"



Self-squaring fractals

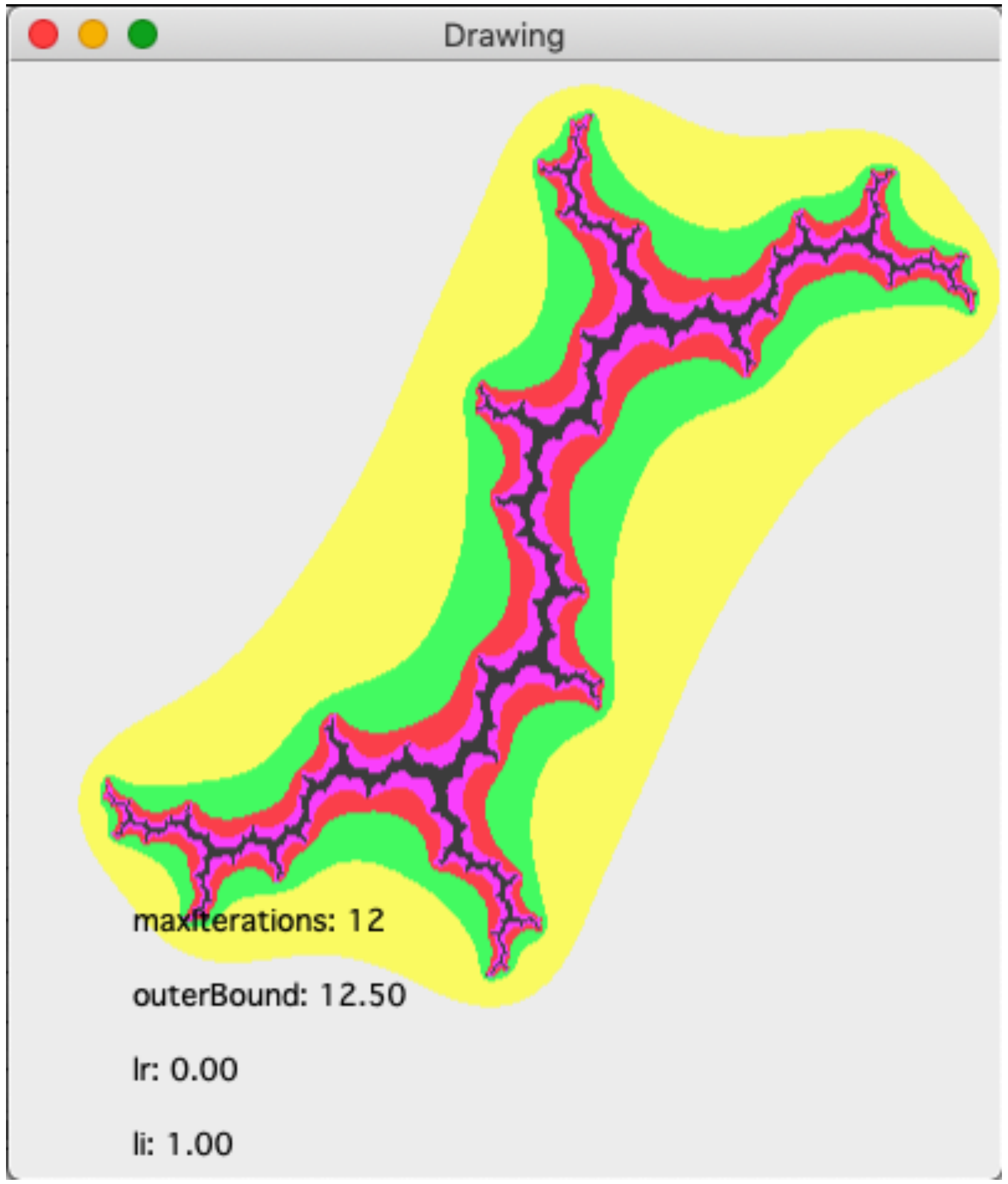
Based on simple functions in complex space

Insert complex numbers (points) into a function

Apply function recursively, and analyze the behaviour.

- Diverge?
- Converge?
- Chaotic?

Converge or chaotic: Does it keep within some limit in a number of iterations?



Self-squaring fractals

The Julia set

$$z_{k+1} = z_k^2 + \lambda$$

Julia set for $\lambda = (0, 1) = 0 + j$

The Julia set - Implementation

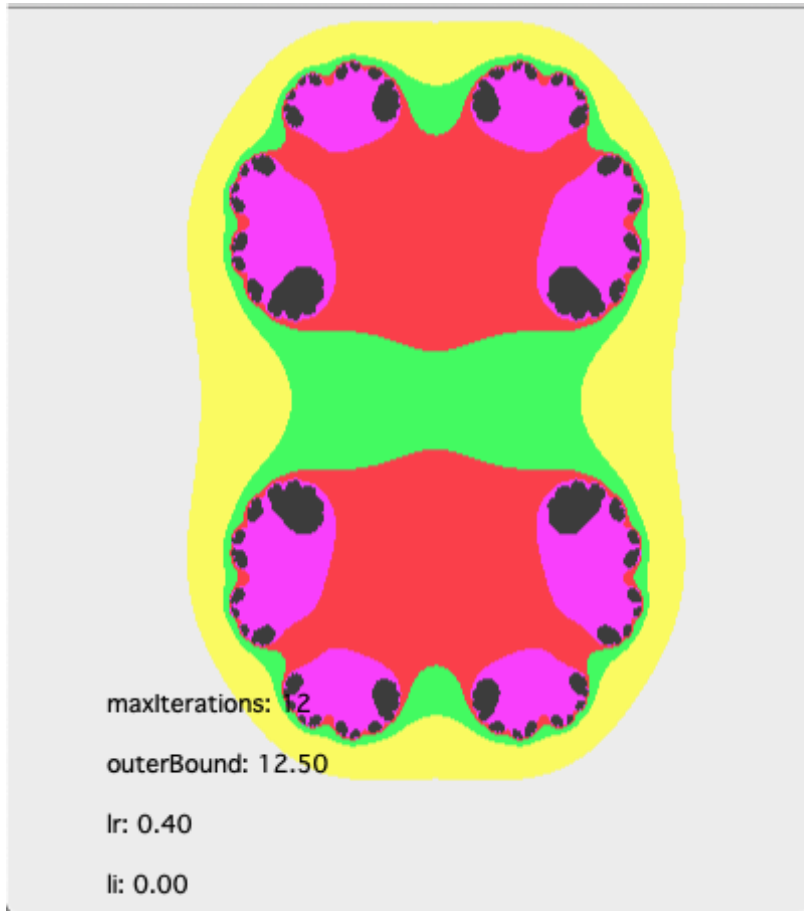
```
for y = miny to maxy
   for x = minx to maxx
      (zr, zi) = scaling of (x,y)

      for i = 0 to maxiterations
         z = z² + λ
         if |z| > R then Leave

      Draw pixel (x,y) (different colors for different i)
```

maxiterations ≈ 15 enough for decent result.
$R^2 \approx 10$

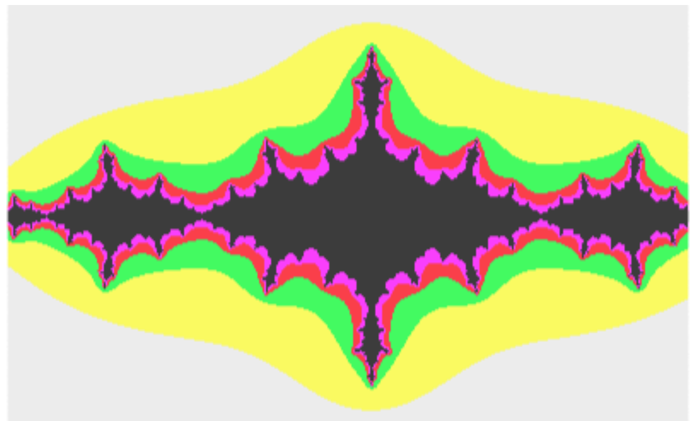
Other Julia sets

$$z_{k+1} = z_k^2 + \lambda$$

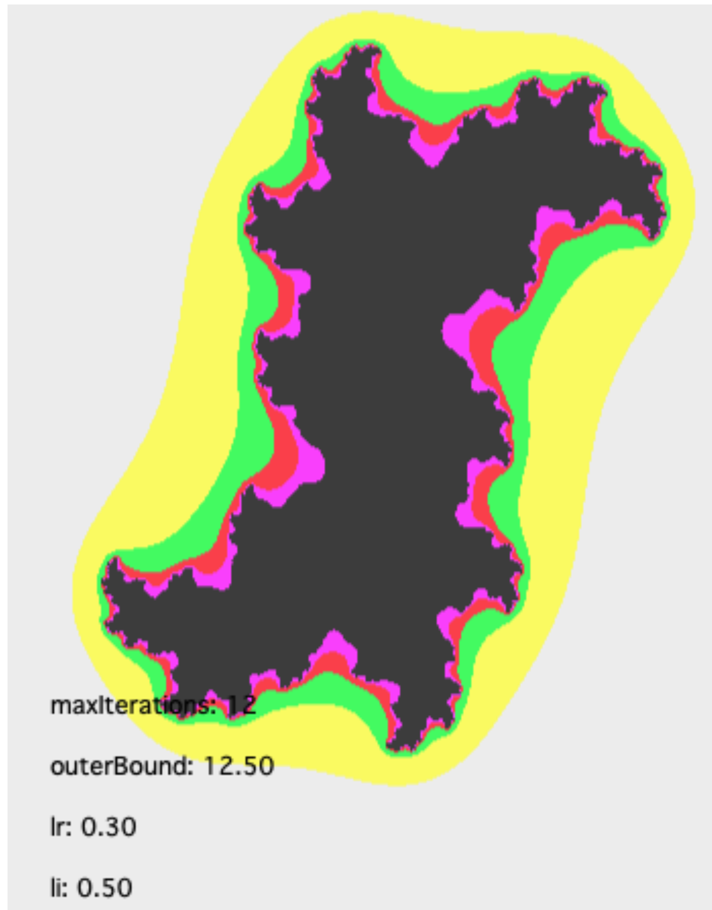
Other λ values



$\lambda = (0.4, 0)$



$\lambda = (-1.3, 0)$

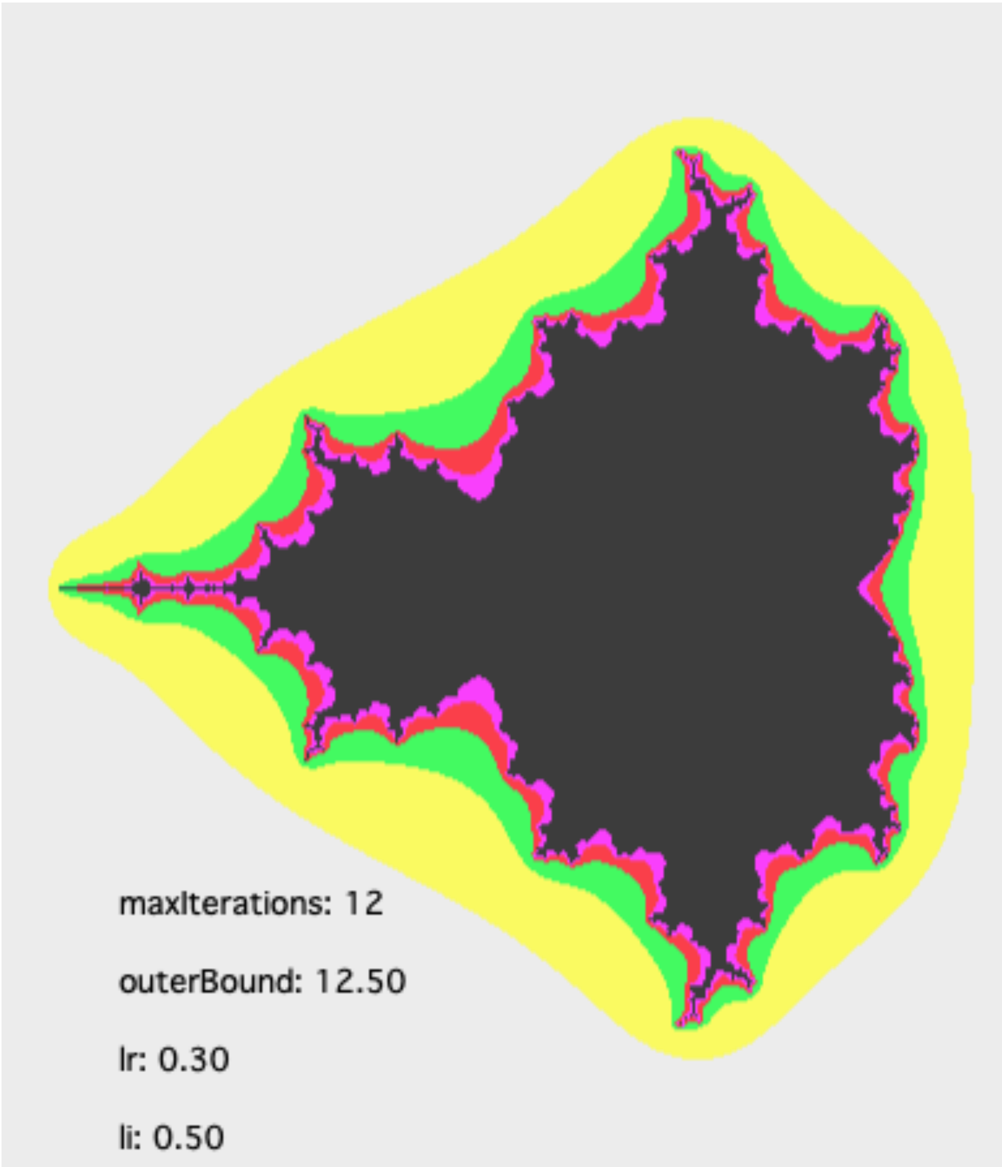


$\lambda = (0.3, 0.5)$

Self-squaring fractals

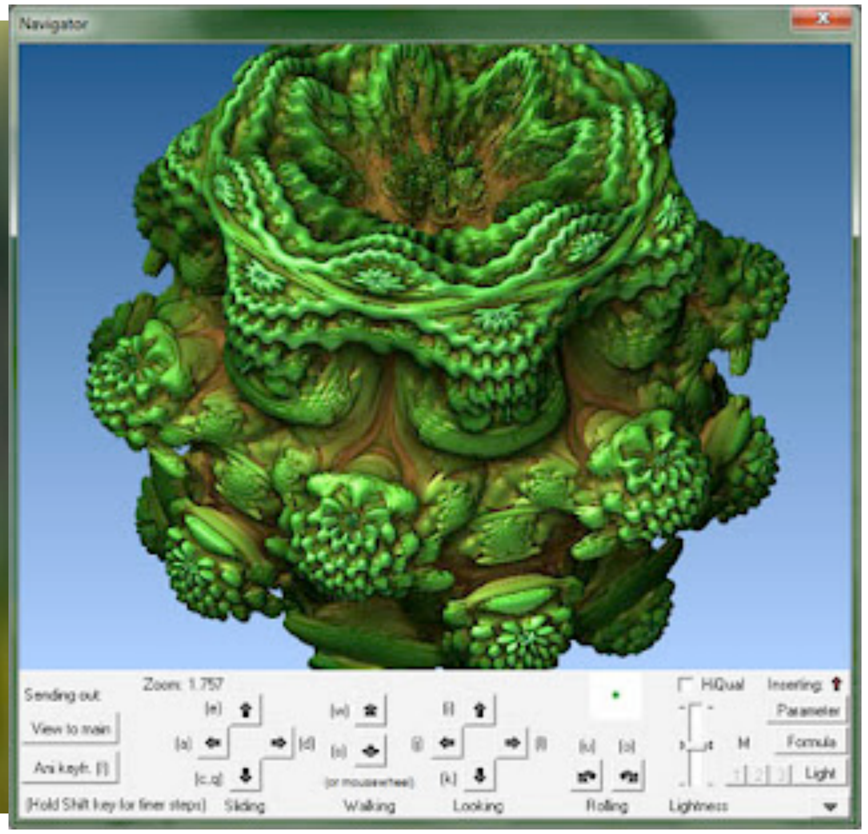
The Mandelbrot

$$z_{k+1} = z_k^2 + z_0$$



3D fractals

Mandelbulb. Based on polar coordinates rather than complex numbers.

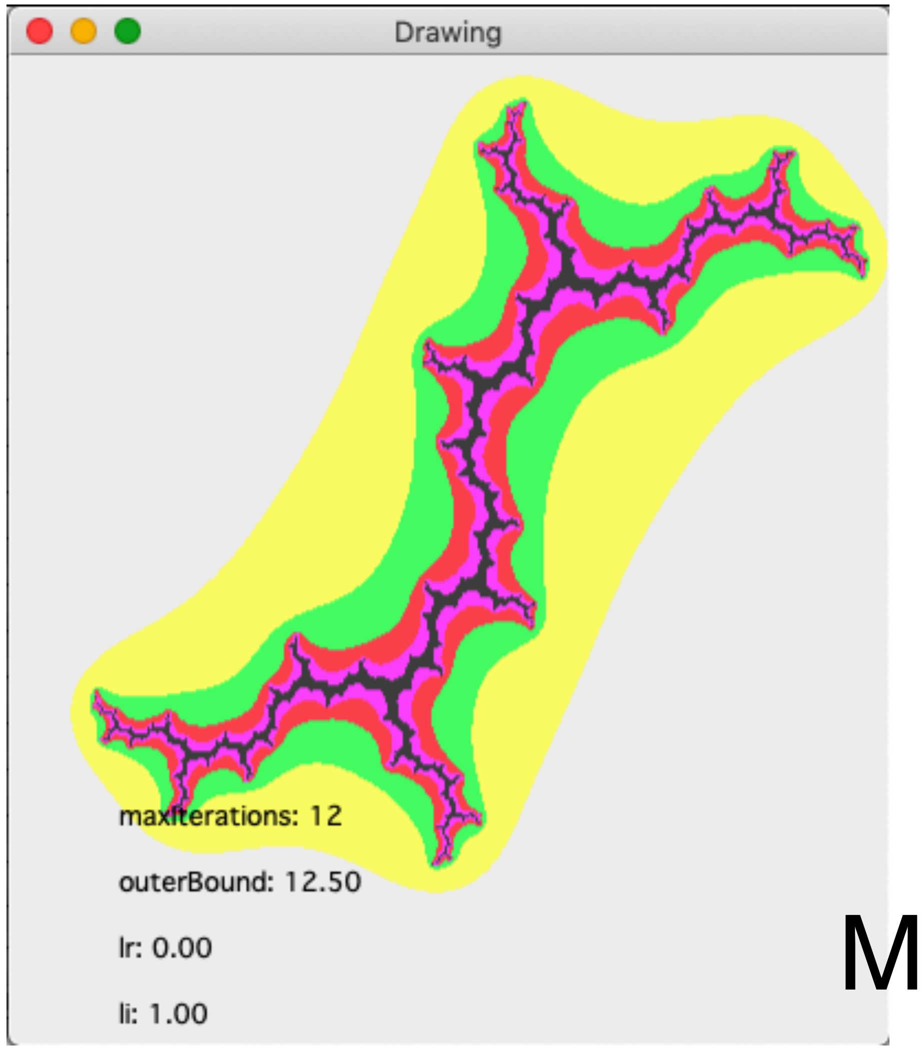


Mandelbulb

Several different variations. Amazing surrealistic scenes! Some potentially useful - but you will rather adapt yourself to the fractal than the fractal to you needs.

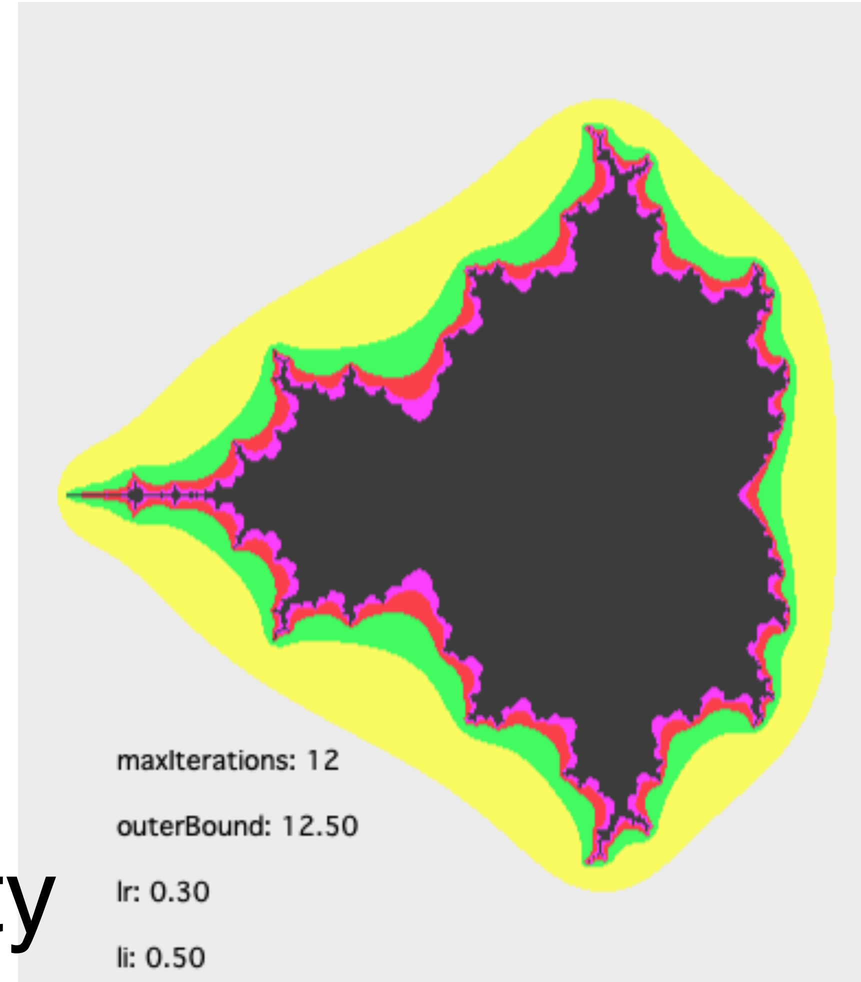
Many other 3D fractals exist.

Self-squaring fractals



- Beautiful
- Non-predictable
- Limited usability

Mathematical curiosity



Fractals, summary

1) Geometrically constructed fractals

Very useful for generating many kinds of natural objects

Allows design of complex models with arbitrary resolution

2) Self-squaring fractals (and other adventures in the complex plane)

Questionable practical usability

Hard to do planned designing