Natural Phenomena Modeling and Visualization

What are natural objects?

- E.g. (Deussen et al. [Realistic modeling and rendering of plant ecosystems](#).)

Natural objects…

- Often merged to dynamic processes

Natural objects: terrain

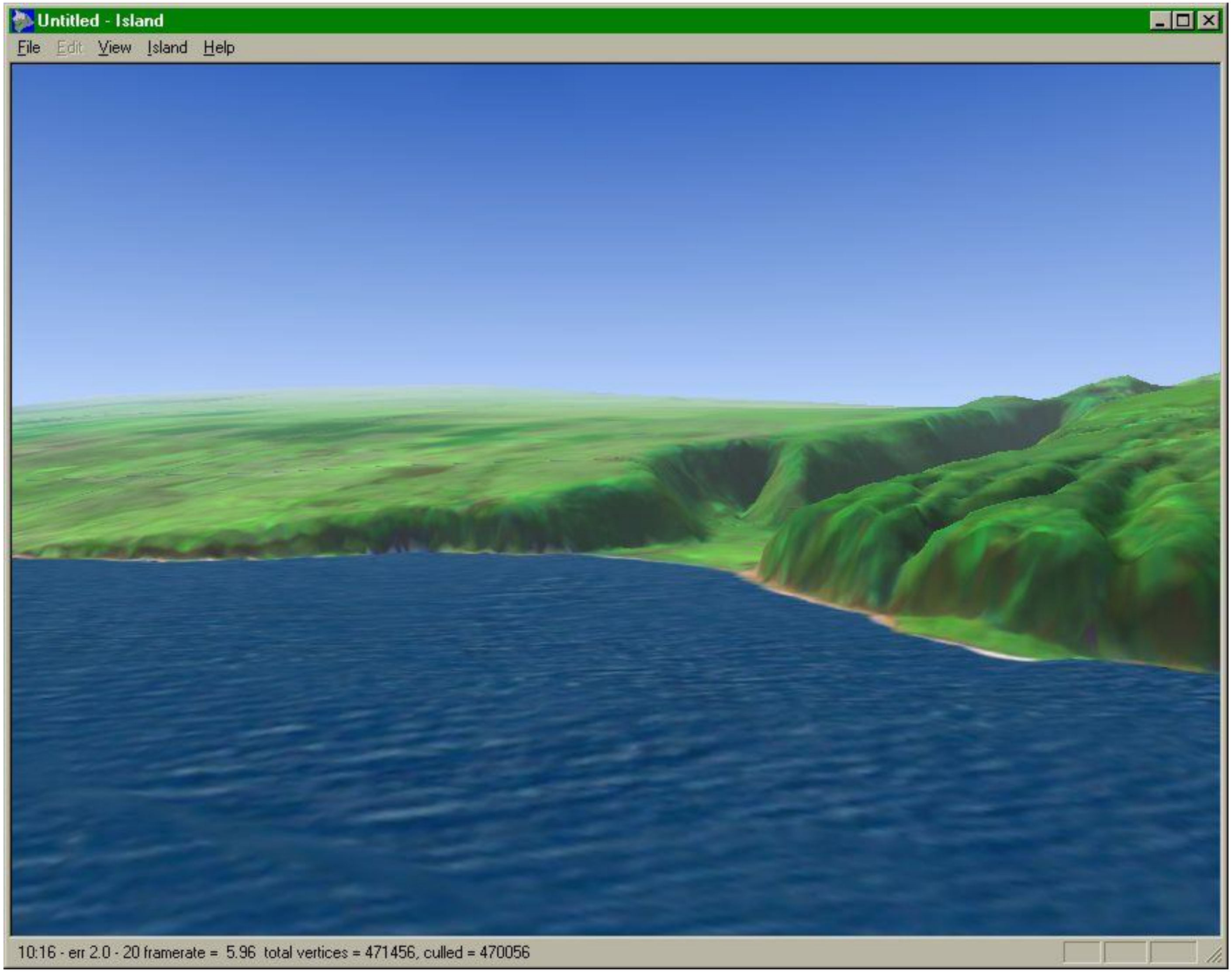
What do we want to model ?

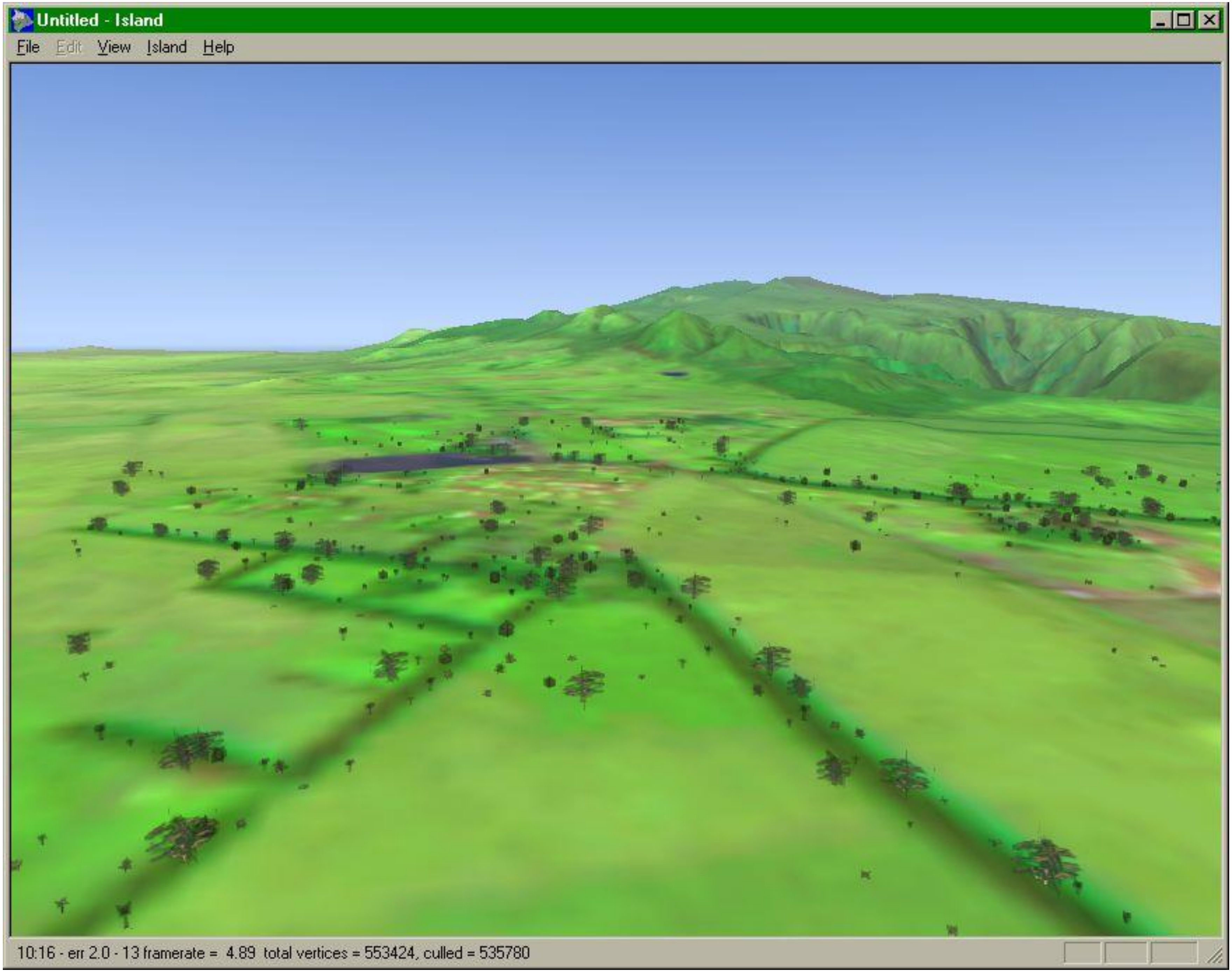
What we can see .....

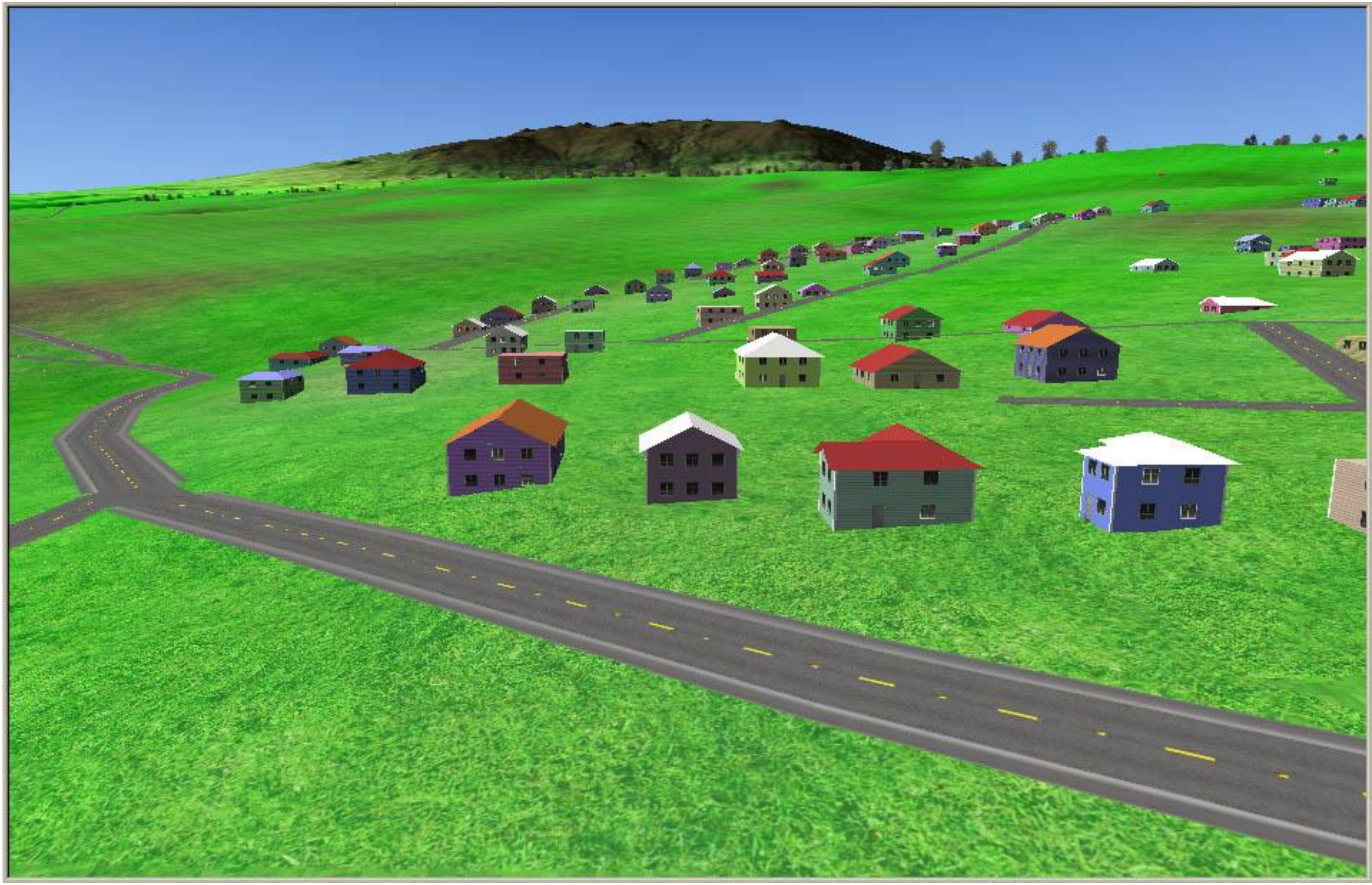
- Terrain
- Plants
- Water surface
- Clouds
- Smoke

Terrain modeling

Old examples from: **`www.vterrain.org`**




Untitled - Island
File Edit View Island Help
10:16 - err 2.0 - 13 framerate = 4.89 total vertices = 553424, culled = 535780



How to model complex natural objects?

- Parametric surfaces?
- Implicit surfaces?

- We have to find a new way

Fractals… have a long history

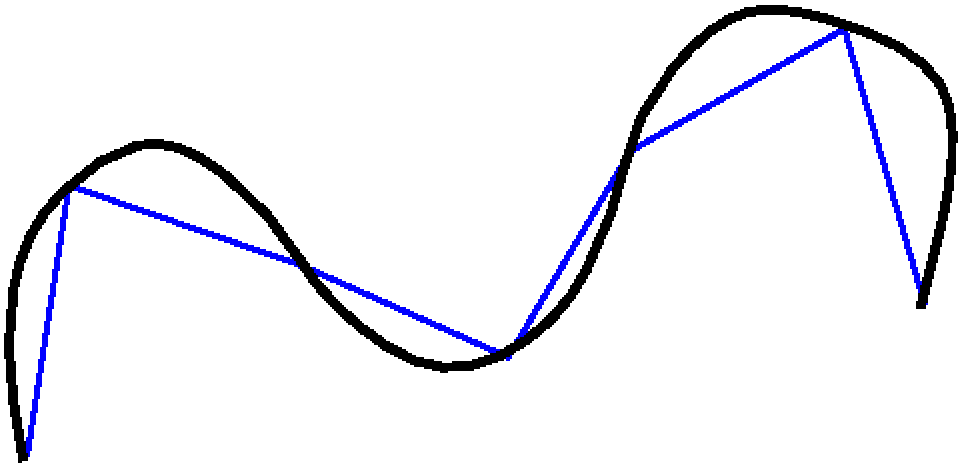
•XVII Gallileo (anti - fractals)

•XIX / XX Koch, Sierpiński, ...

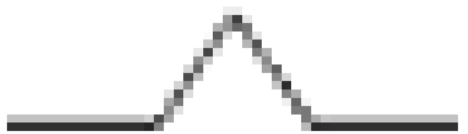
•Second half of XX c. - Benoit Mandelbrot

Length measurement

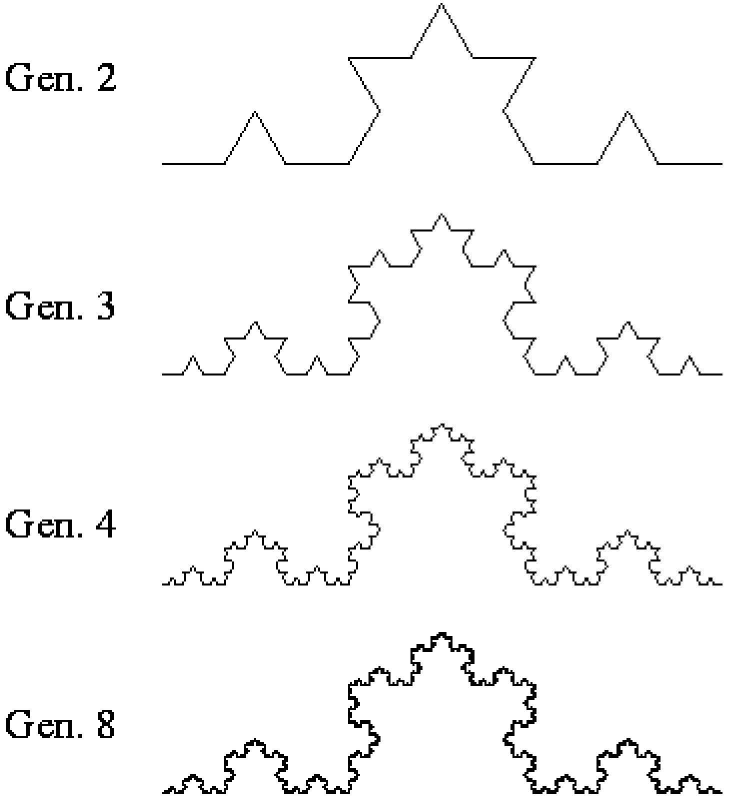
Measuring the lenght of a smooth curve

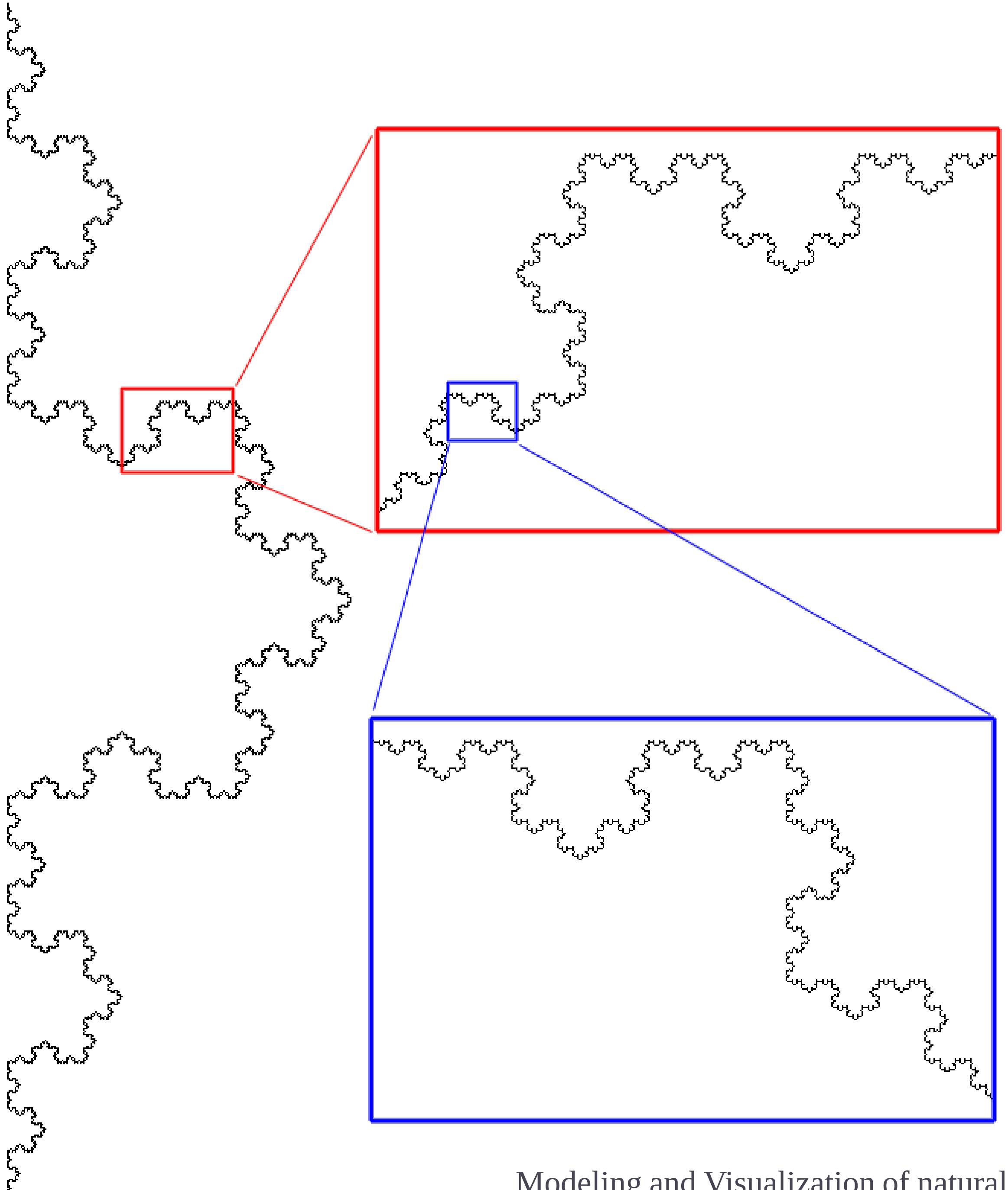


Mathematical constructions



Von Koch curve

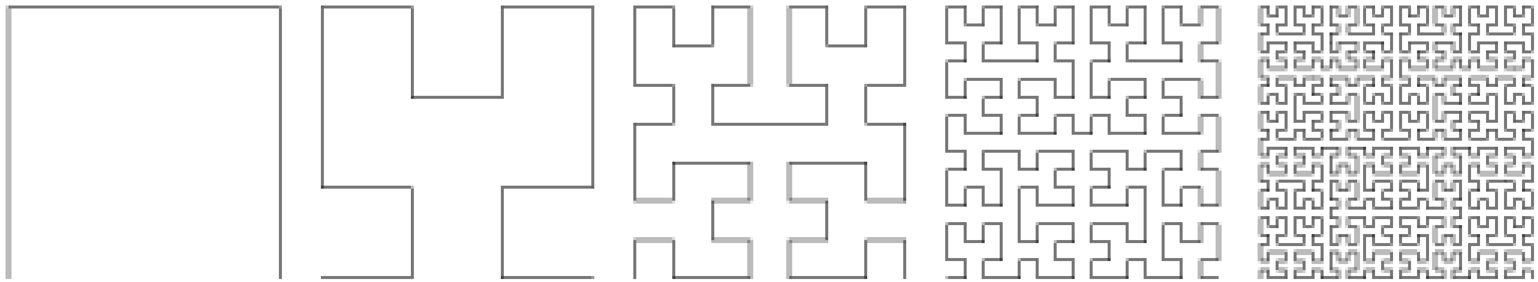




Self-similarity

- precise – reaches infinity
- Von Koch curve is a good example

Mathematical constructions



Hilbert curve

Definition of fractal

A curve or geometric figure, each part of which has the same statistical character as the whole. Fractals are useful in modeling structures (such as eroded coastlines or snowflakes) in which similar patterns recur at progressively smaller scales, and in describing partly random or chaotic phenomena such as crystal growth,

Fractal Dimension

Based on von Koch example

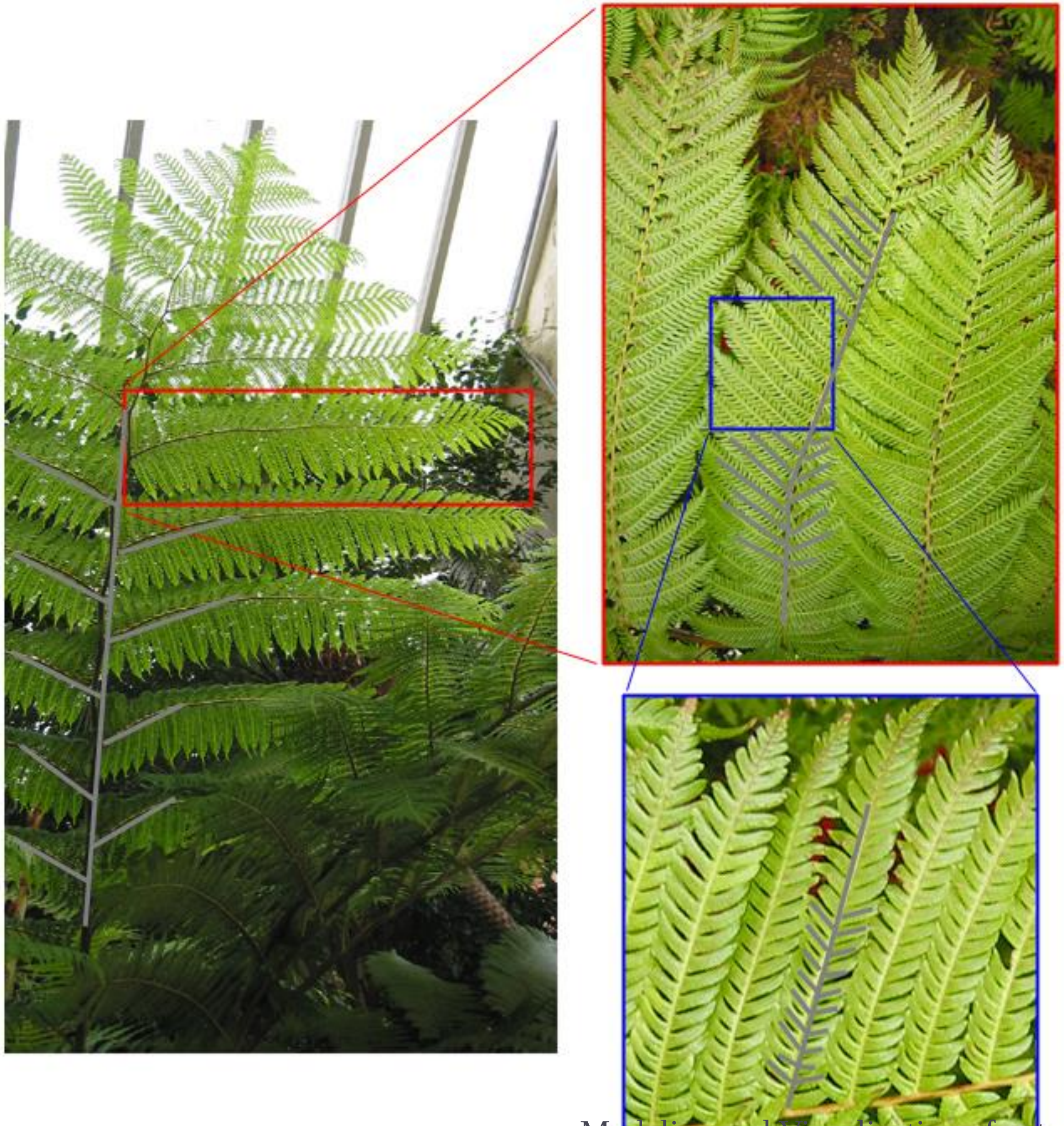
$$-\frac{\log N(\varepsilon)}{\log \varepsilon} = \frac{\log 4}{\log 3} = 1.26$$

Lenght measurement

B. Mandelbrot: How long is the coast of Britain?
Science, 155:636-638, 1967

Example of the coastline paradox. If the coastline of Great Britain is measured using units 100 km (62 mi) long, then the length of the coastline is approximately 2,800 km (1,700 mi). With 50 km (31 mi) units, the total length is approximately 3,400 km (2,100 mi), approximately 600 km (370 mi) longer. (Wikipedia)

Self-similarity in real world



- Limited when compared to mathematical constructions to only a few scales
- Obviously approximated
- Self-similarity among plants is an effect of their growth process

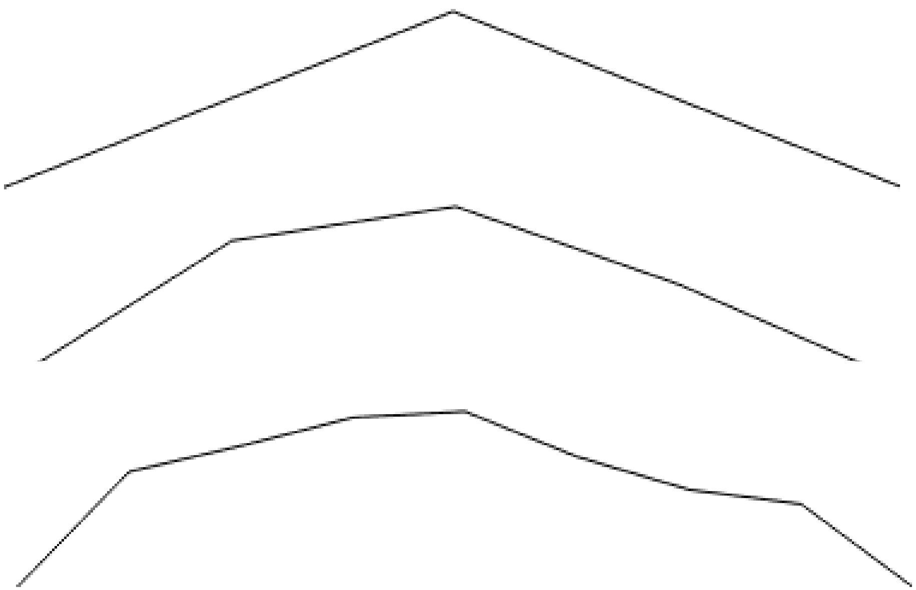
Fractal food

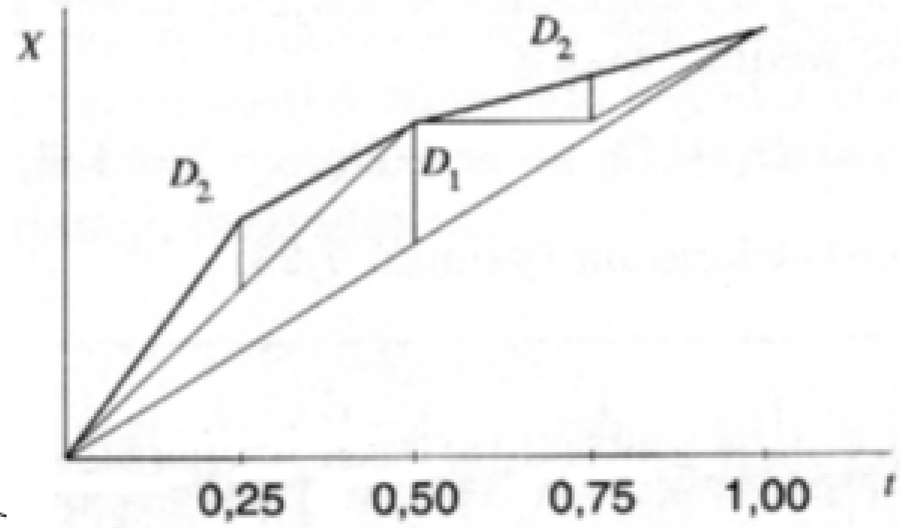
http://www.fourmilab.ch/images/Romanesco/

Fractal terrains

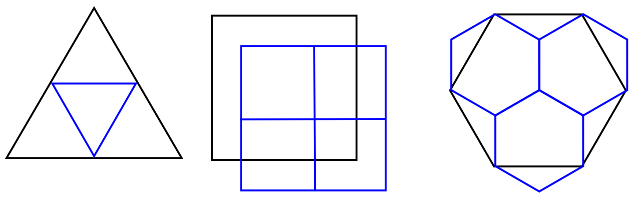
We can start from Brown's motion

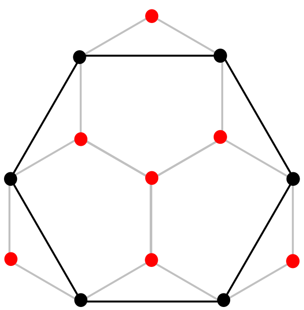
Fractal terrains



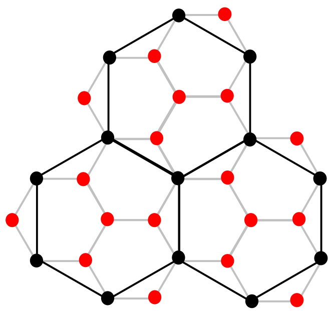


Fractal terrain: 2D mesh

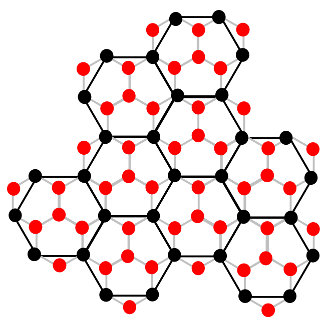




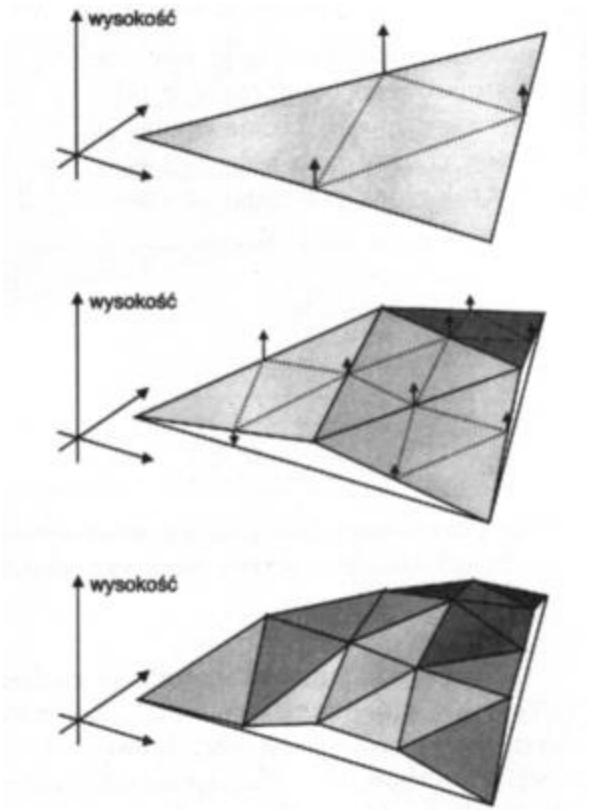
Krok I



Krok II

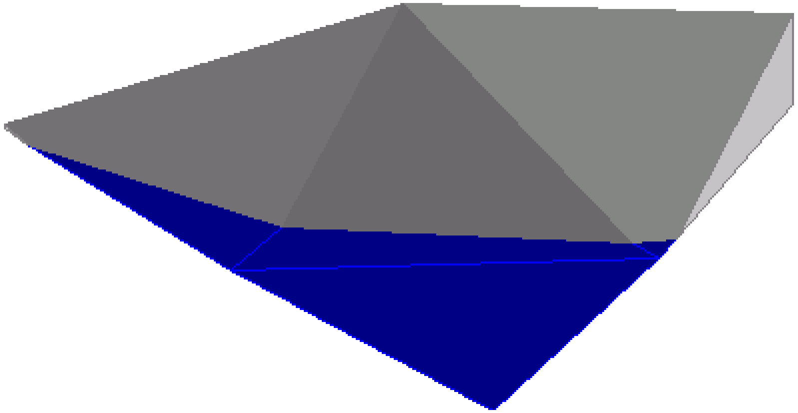


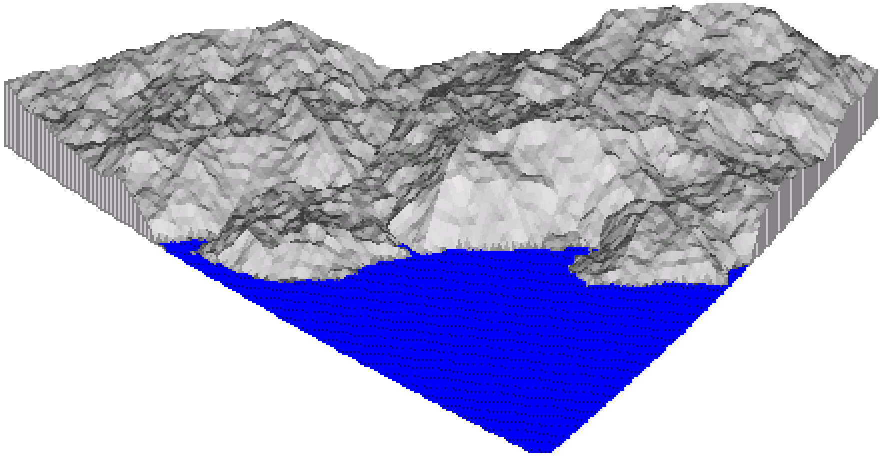
Krok III



Fractal terrain (historic example)

Fractal Landscapes 3.0





What do we search for?

- Terragen, 4.x

Generation and terrain edition in WebGL

- http://callumprentice.github.io/apps/webgl_terrain/
- http://www.webgl.com/2012/03/webgl-demo-terrain-generation-diamond-square-algorithm-and-three-js/
- http://codeflow.org/entries/2011/nov/10/webgl-gpu-landscaping-and-erosion/#demo

Terrain generation in Three.js

- Use ”three.js landscape generator” and you’ll find a lot
- https://github.com/IceCreamYou/THREE.Terrain
- https://github.com/jbouny/terrain-generator
- https://threejs.org/examples/webgl_terrain_dynamic.html
- Cesium.js – library for Earth maps

Affine transformation

$$x` = ax + by + c$$

$$y` = dx + ey + f$$

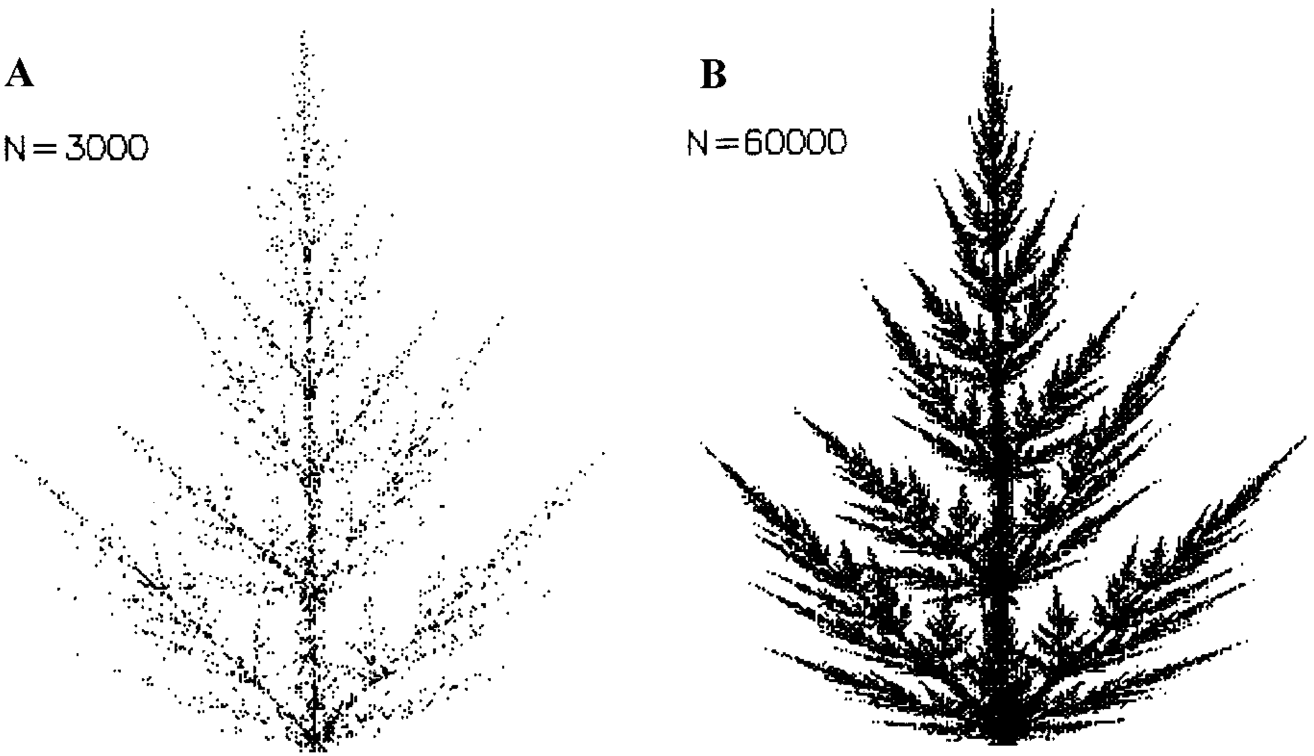
Any figure can be transformed according to affine transformation and:

- Translated
- Scaled
- Sheared
- Rotated
- ….

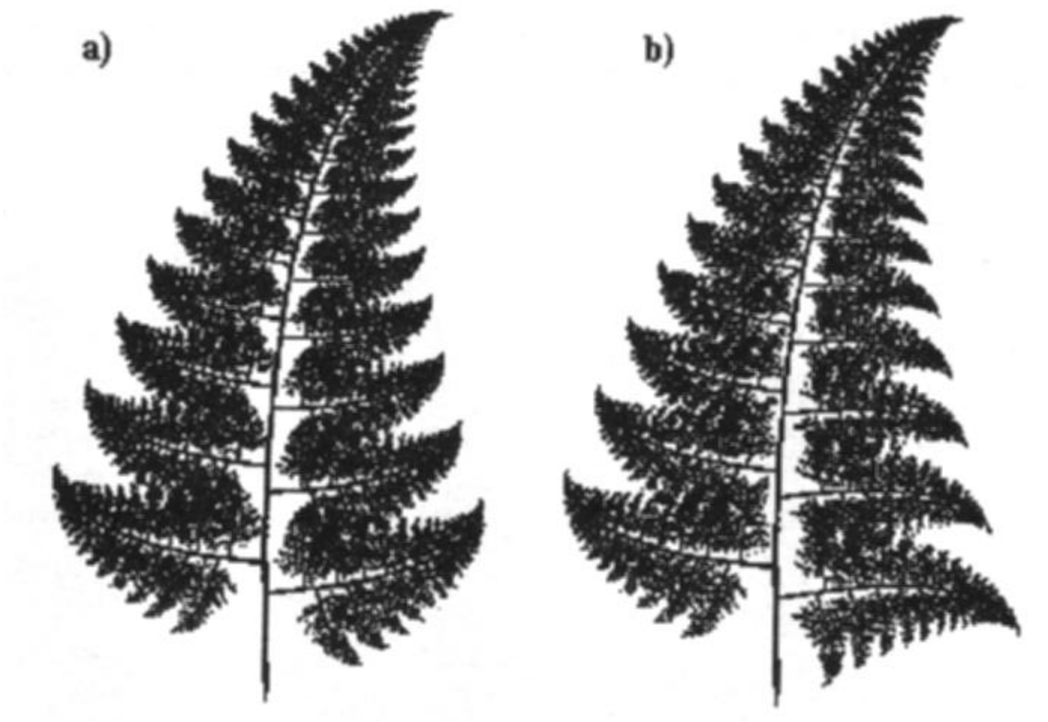
Example: a tree

Transformation	***a***	***b***	***c***	***d***	***e***	***f***
1	-0,67	-0,02	0	-0,18	0,81	10,0
2	0,4	0,4	0	-0,1	0,4	0
3	-0,4	-0,4	0	-0,1	0,4	0
4	-0,1	0	0	0,44	0,44	-2

We generate a series of points (x_0, y_0), (x_1, y_1), (x_2, y_2), ... , (x_n, y_n), ...
Point (x_0, y_0) jest has arbitrary coordinates on a plane.
We choose any of the transformations on random and use it to make a point (x_1, y_1) from (x_0, y_0).
We repeat the procedure.


A
N=3000
B
N=60000

Barnsley fern – look up in the web



a) With mirror reflection; b) without

Elements of L-systems

Introduction

The growth process in the living world can be represented in the form of fractal structures.

Examples include plants, lung channels, and blood vessels.

There are various similarities during growth because growth is associated with the repetition of a simple process. Such a process can be presented with sufficient accuracy as a set of processing rules.

Lindenmayer systems

Lindenmayer systems or L-systems are a type of symbolic dynamic systems with added interpretation of the geometric evolution of the system.

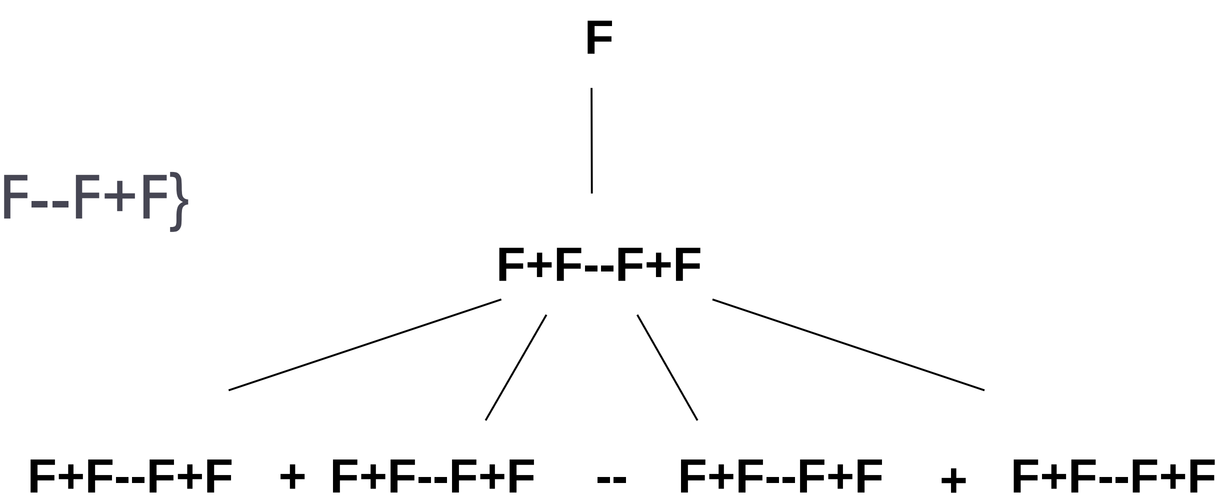
They were introduced by Aristid Lindenmayer in 1968 and used to model biological growth.

Grammar

- Grammar $G = \langle N, T, P, Z \rangle$ where:
  - N – Non-terminal symbols,
  - T – Terminal symbols,
  - P – Productions $\alpha \rightarrow \beta$
  - $Z \in N$ – initial symbol
- where:
  - $P \subseteq (N \cup T)^{+} \times (N \cup T)^{*}$
  - $P = \{ \alpha \rightarrow \beta \mid \alpha \in (N \cup T)+, \beta \in (N \cup T)^{*} \}$

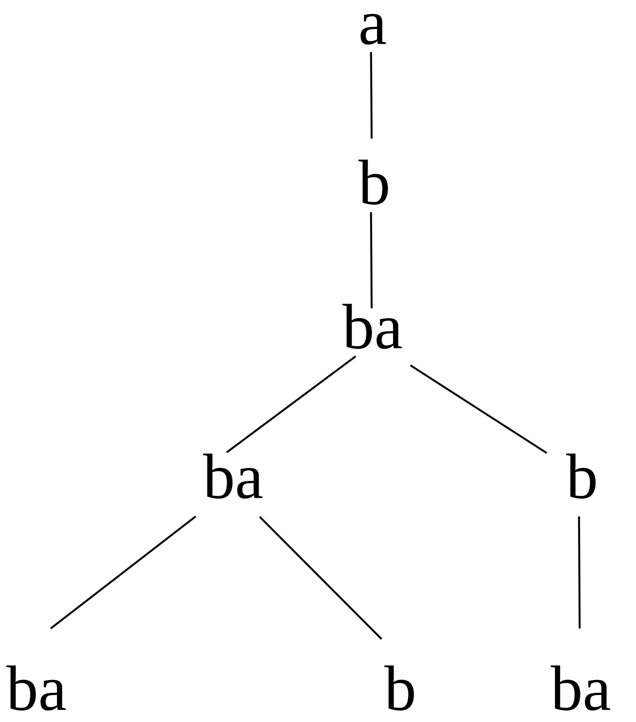
Grammars and L-systems

- On the contrary to Chomsky grammars we perform productions in parallel.
- Example:
  - N = {F}
  - T = {+,–}
  - P = {F→F+F--F+F}
  - Z = F



Types of L-systems

- Deterministic and Contextless systems (*DOL-systems*)
- *Fibonacci L-system*
  - $N = \{a,b\}$
  - $P = \{p_1,p_2\}$
    - $p_1: a \rightarrow b$
    - $p_2: b \rightarrow ba$
  - $Z = a$



DOL-systems cd.

- Fibonacci sequence in terms of L-systems
  - $F_0 = F_1 = 1$;
  - $F_{n+2} = F_{n+1} + F_n$ for $n \geq 0$
- Common in real world.

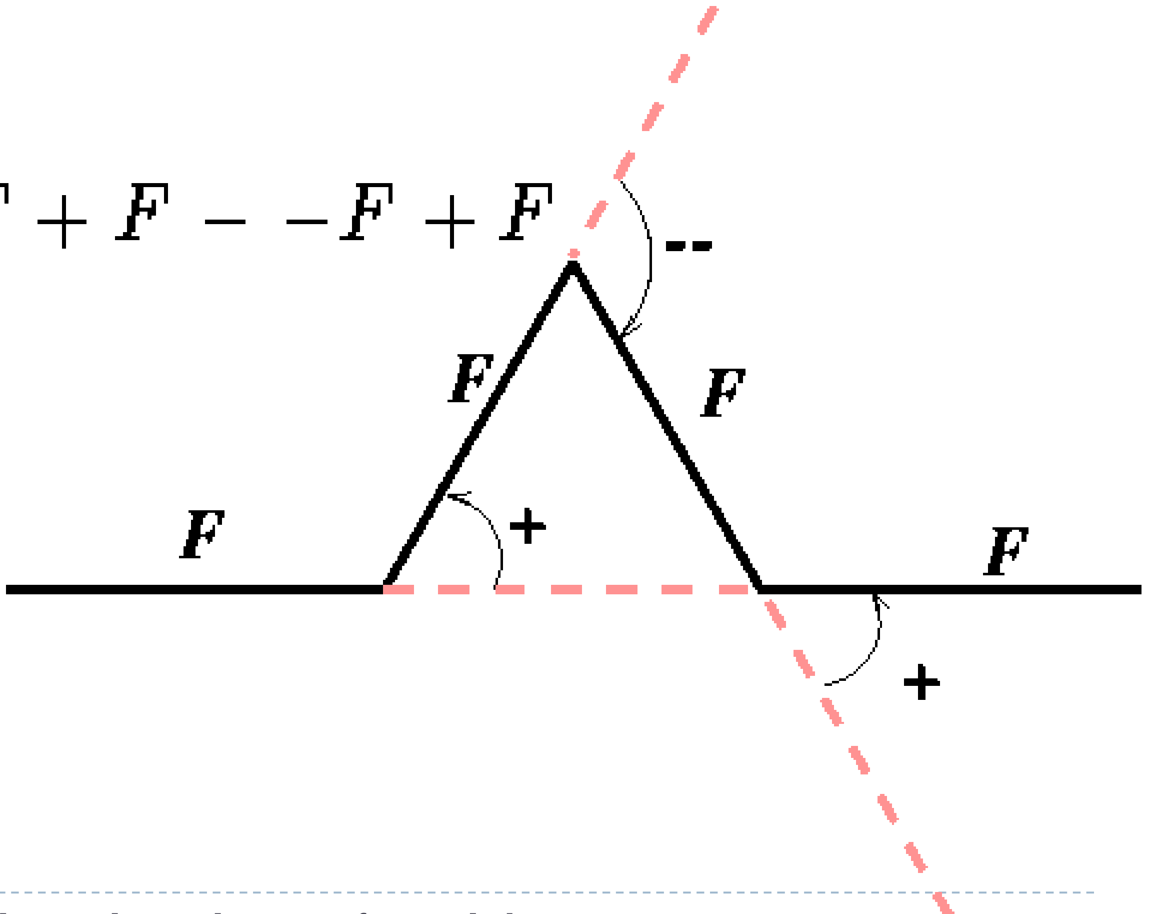
g0:	*a*
g1:	*b*
g2:	*ba*
g3:	*bab*
g4:	*babba*
g5:	*babbabab*
g6:	*babbababbabba*
g7:	*babbababbabbababbabab*

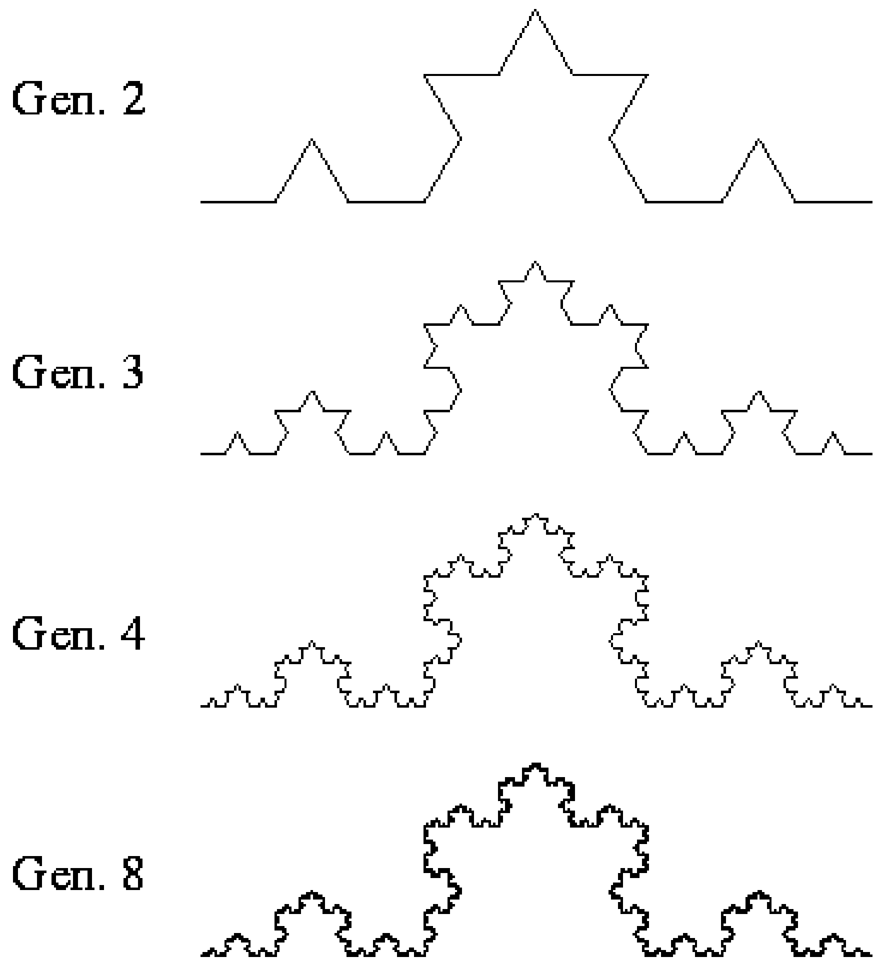
von Koch curve

$$V = \{F, +, -\}$$

$$\omega = F$$

$$p_1 : \quad F \longrightarrow F + F - - F + F$$

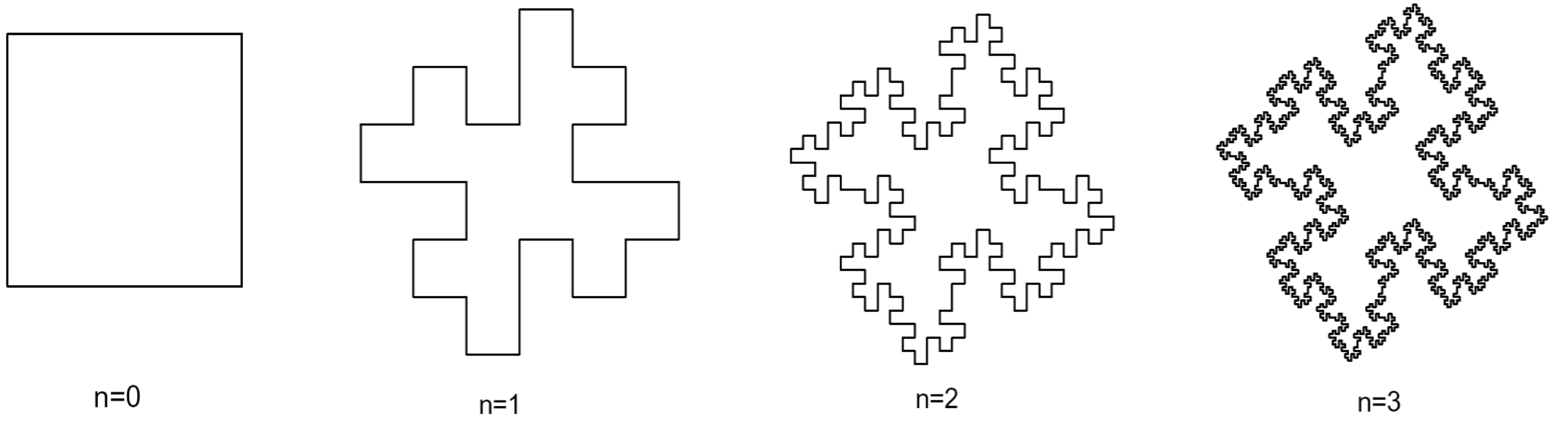



Gen. 2
Gen. 3
Gen. 4
Gen. 8

von Koch Island

Z: F + F + F + F

P1: F → F - F + F + FF - F - F + F



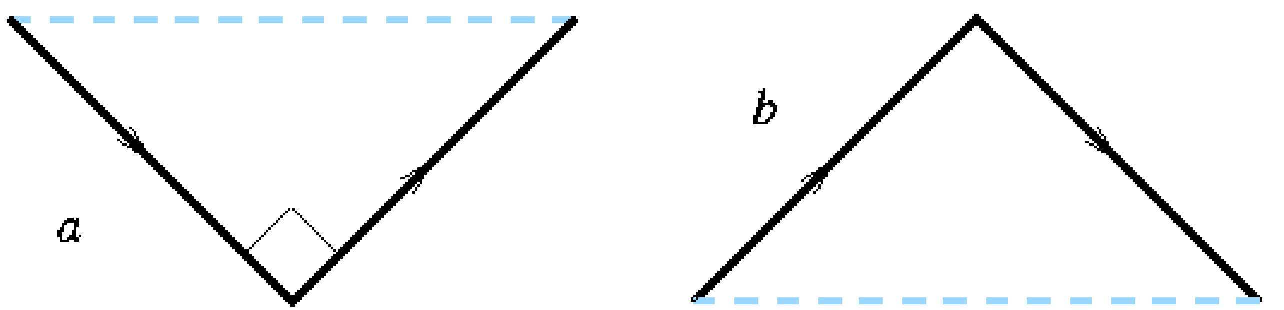
Dragon's curve

$$V = \{a, b\}$$

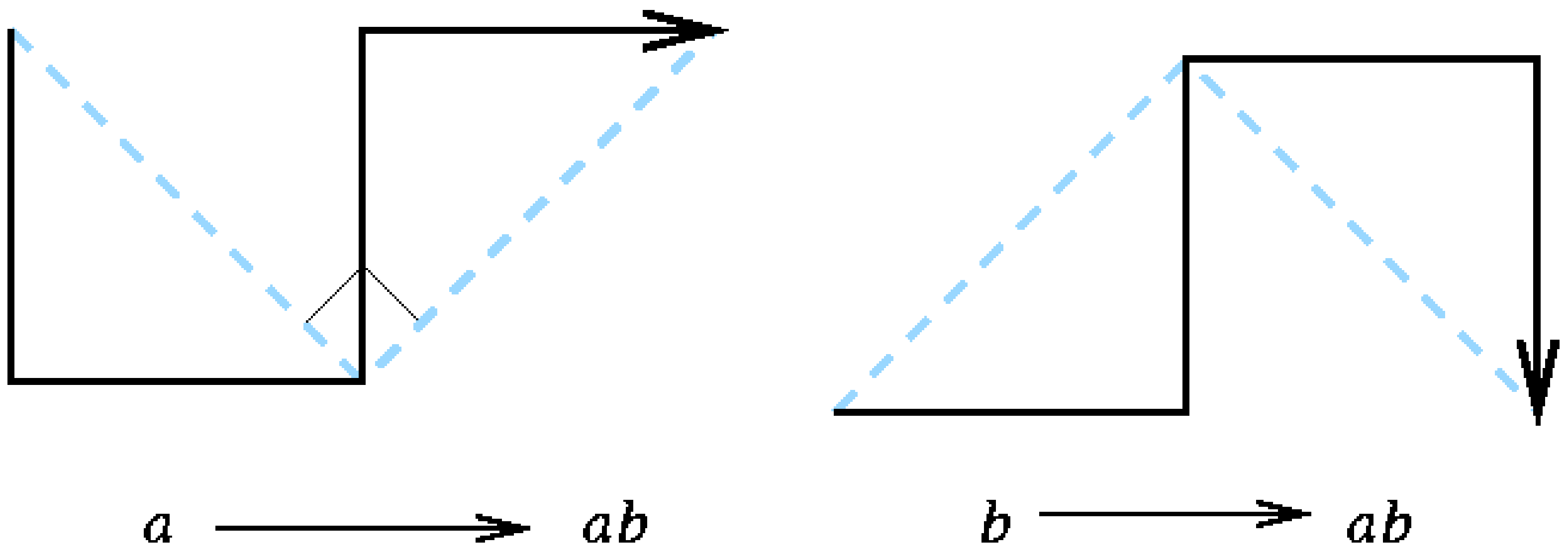
$$\omega = a$$

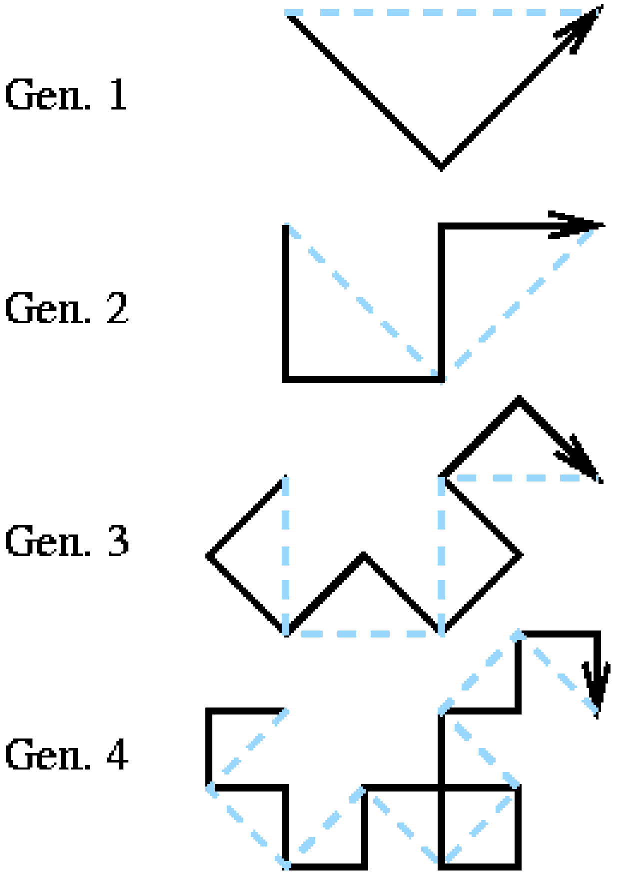
$$p_1 : \quad a \longrightarrow ab$$

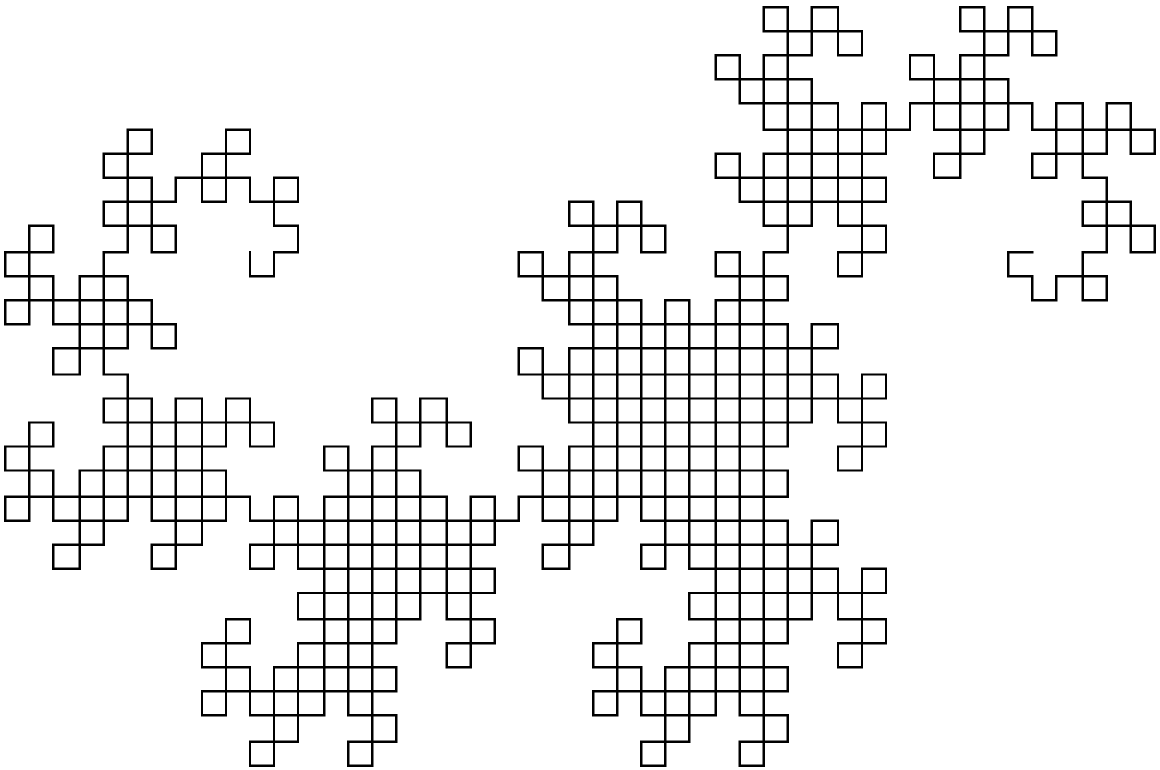
$$p_2 : \quad b \longrightarrow ab$$

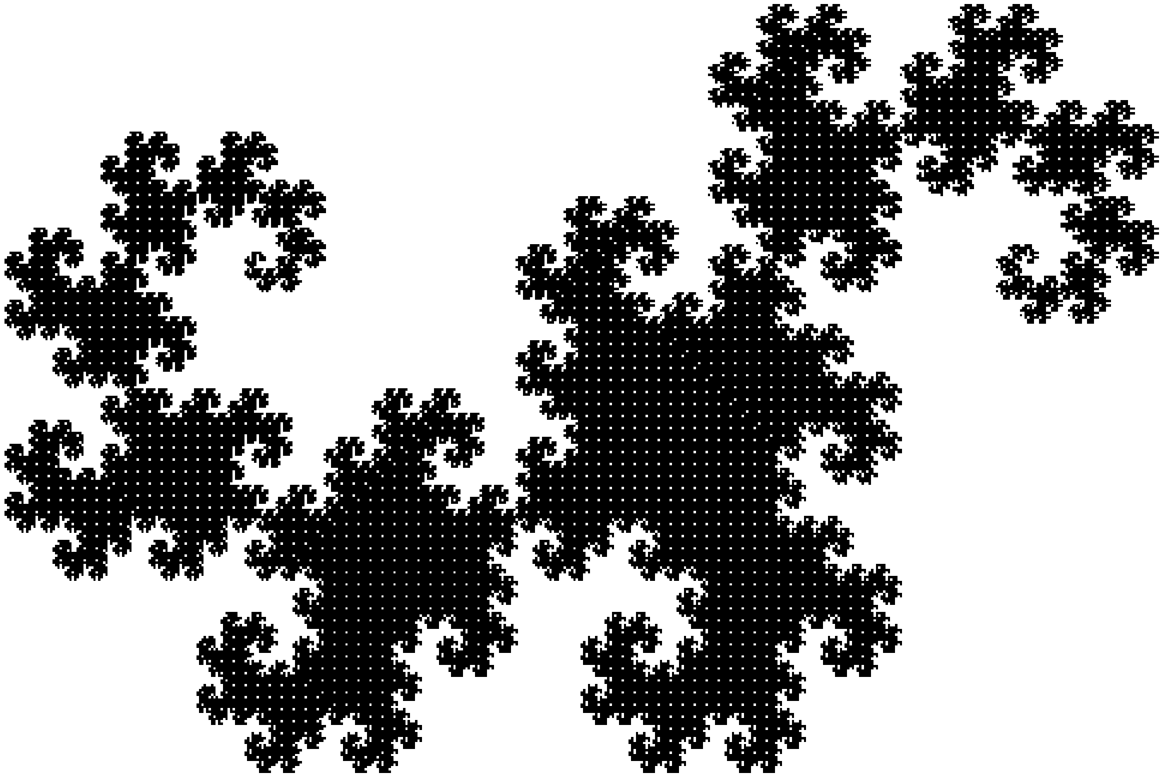


Dragon's curve, c.d.



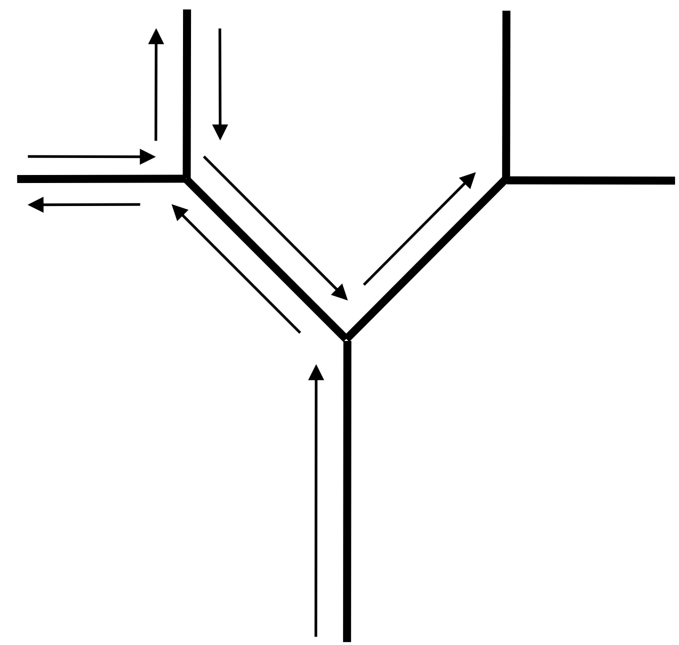

Gen. 1
Gen. 2
Gen. 3
Gen. 4





Branches

- Two new symbols:
  - **[** - put the current state on the stack.
  - **]** – pop the state from the stack.

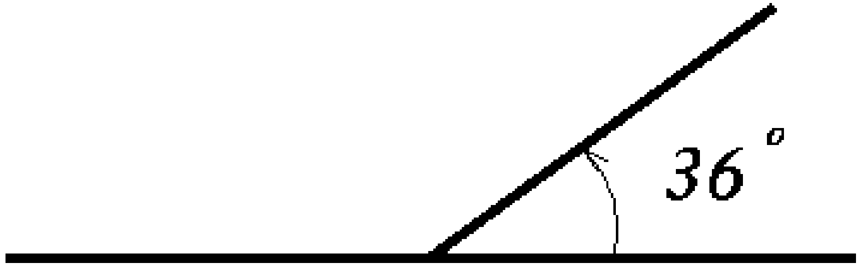


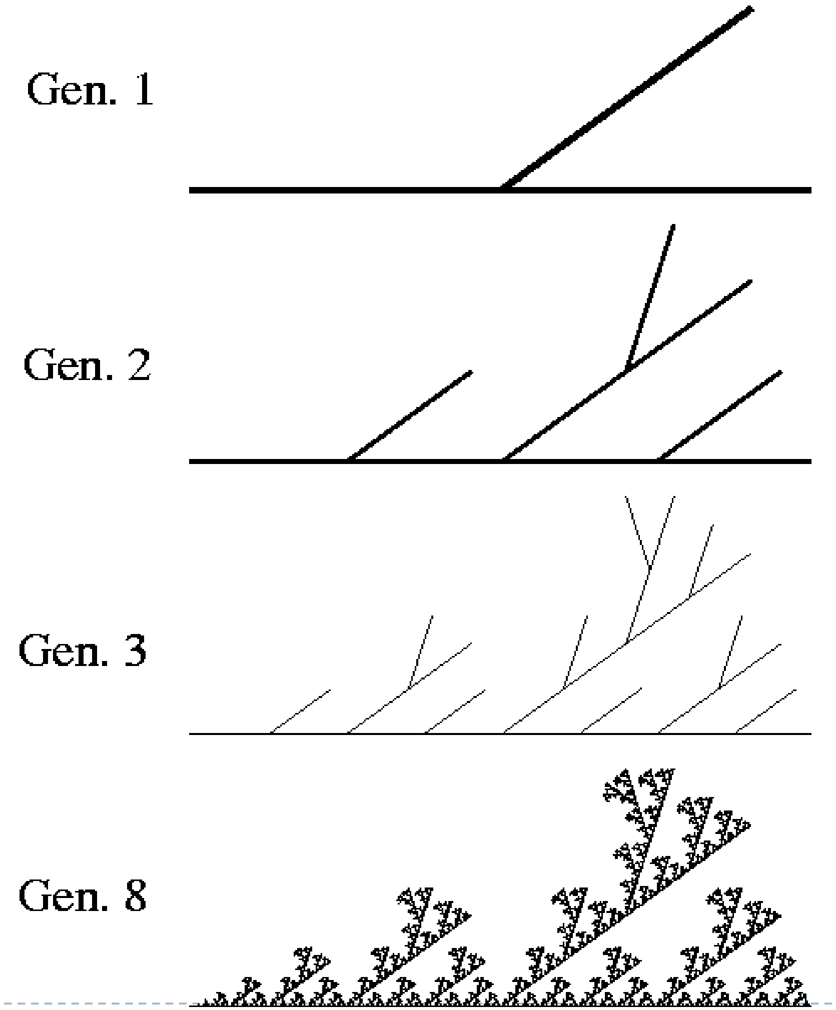
F+F+F----f+F----f+…

vs.

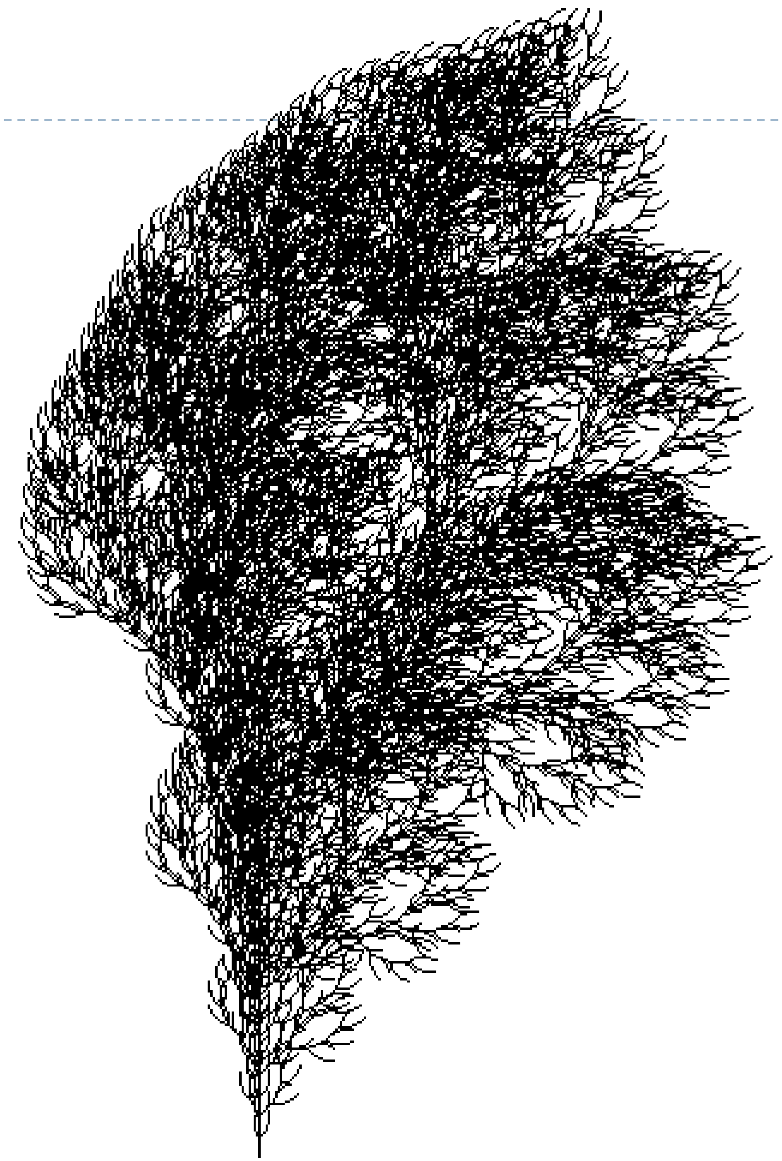
F[+F[+F][-F]][-F[+F][-F]]

Angle 10 *(one tenth of 360 degrees)*
Axiom F
F=F[+F]F

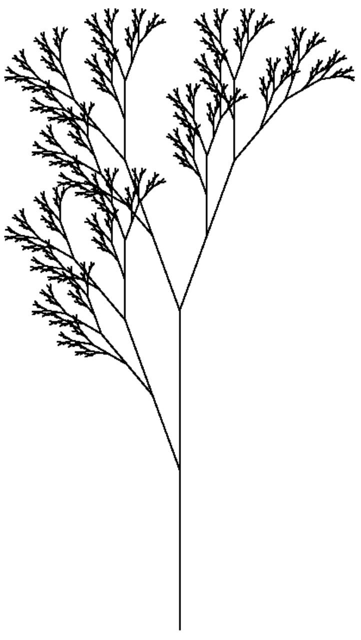



Gen. 1
Gen. 2
Gen. 3
Gen. 8

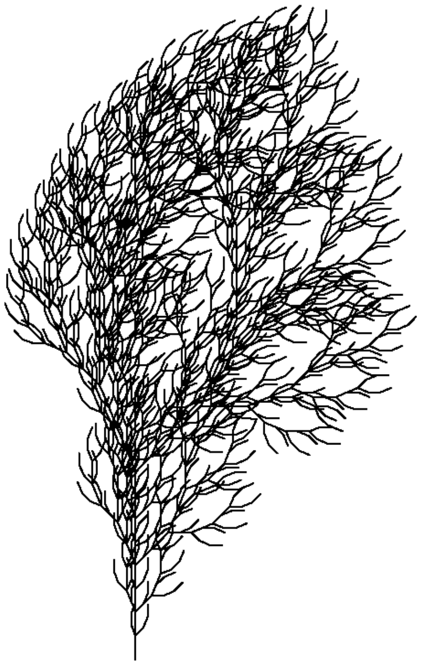
Angle 16
Axiom ++++F
F=FF-[-F+F+F]+[+F-F-F]



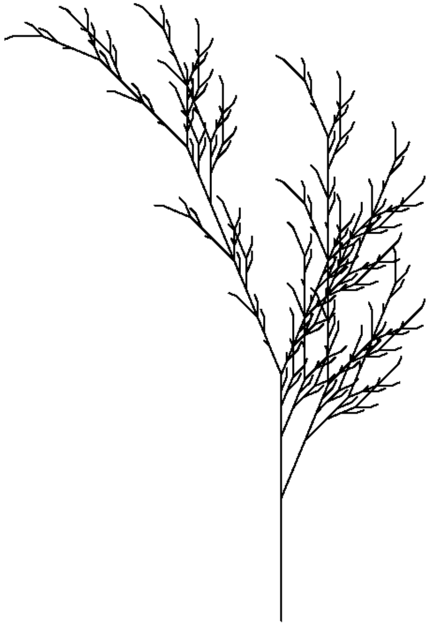
Sample tree-like structures



n=7, δ =20
X
X=F[+X]F[-X]+X
F=FF



n=4, δ =22.5
F
F=FF-[-F+F+F]+[+F-F-F]



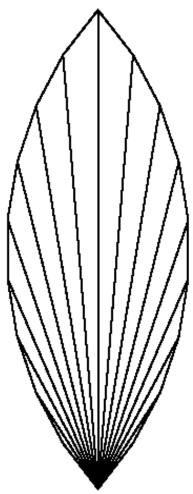
n=5, δ =22.5
X
X=F-[[X]+X]+F[+FX]-X
F=FF

Leaves

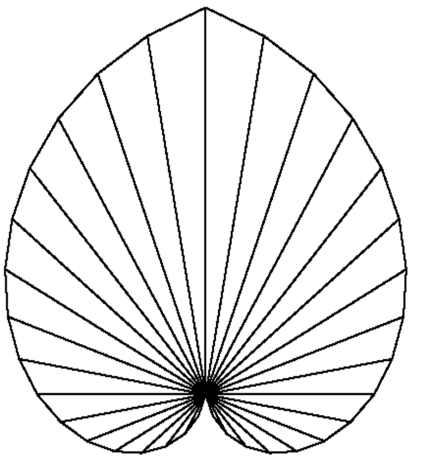
- New symbols
  - {
  - .
  - }

- We are not going to analyze them.

Leaves - examples



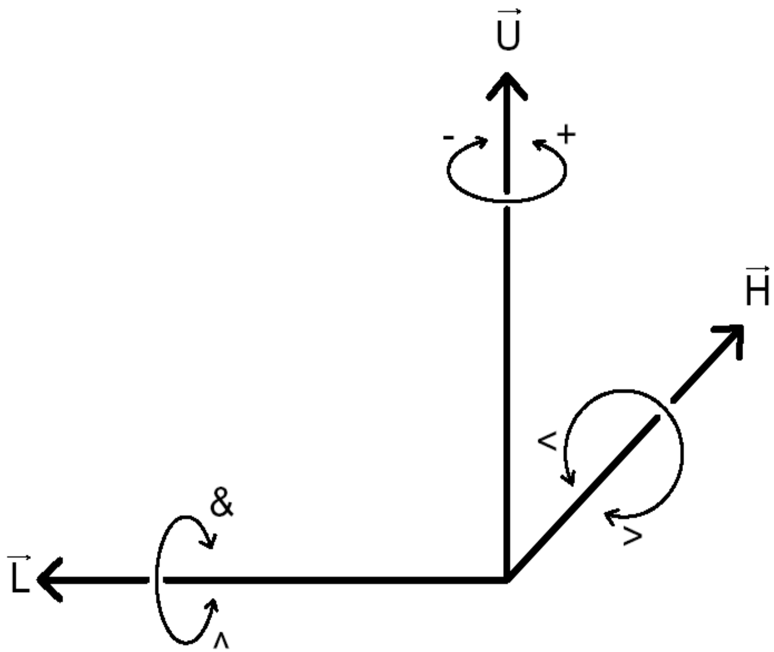
n=9, δ =5
{.a|a|a|}
|=f++|
a=---------



n=18, δ =10
[a][b]
a=[+a{.].c.}
b=[-b{.].c.}
c=gc

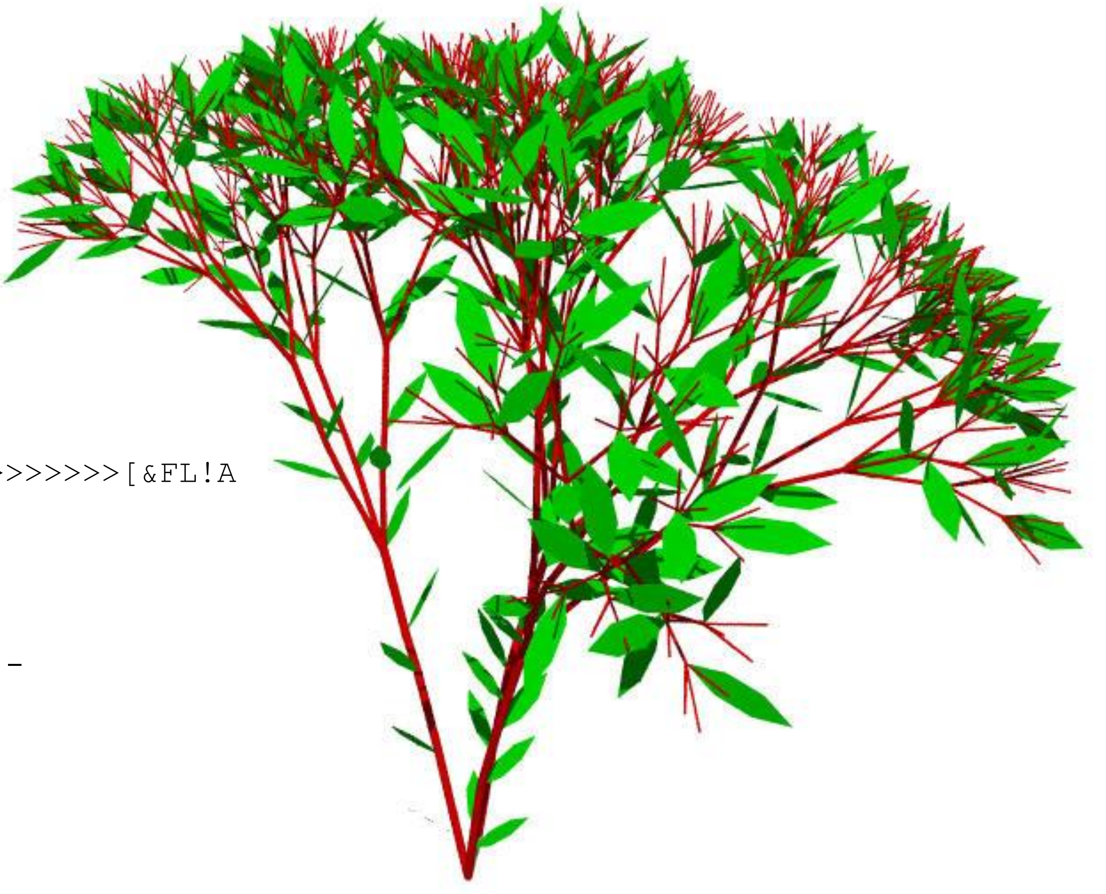
Turtle in 3D

- **+** turn left
- **-** turn right
- **&** pitch down
- **^** pitch up
- **<** roll left
- **>** roll right
- **|** turn back (180 degrees around U)

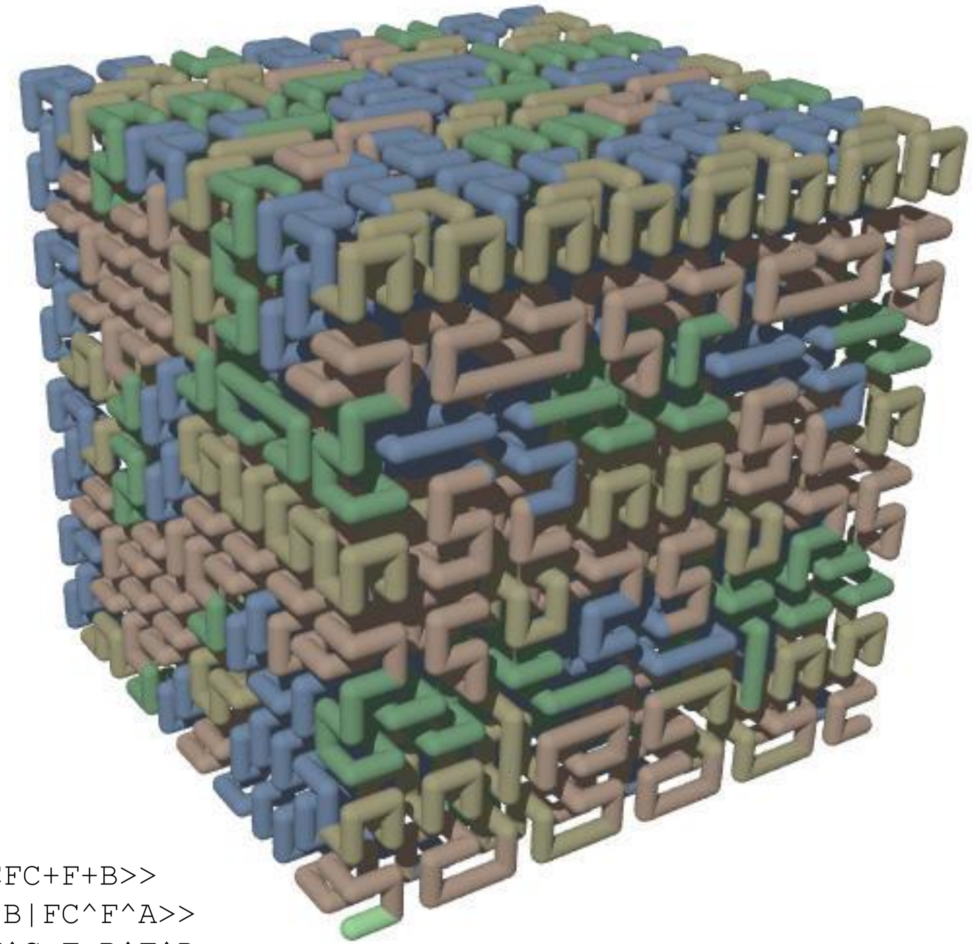


Examples

```
@i=6
@a=22.5
@l=2
@t=0.4
=c(#ff0000)A
A=[&FL!A]>>>>>[&FL!A]>>>>>>>[&FL!A
]
F=S>>>>>F
S=~(5)FL
L=[c(#00ff00)^^{.-
f(0.6)+f(0.6)+f(0.6)-|-
f(0.6)+f(0.6)+f(0.6)}]
@@
```



Examples

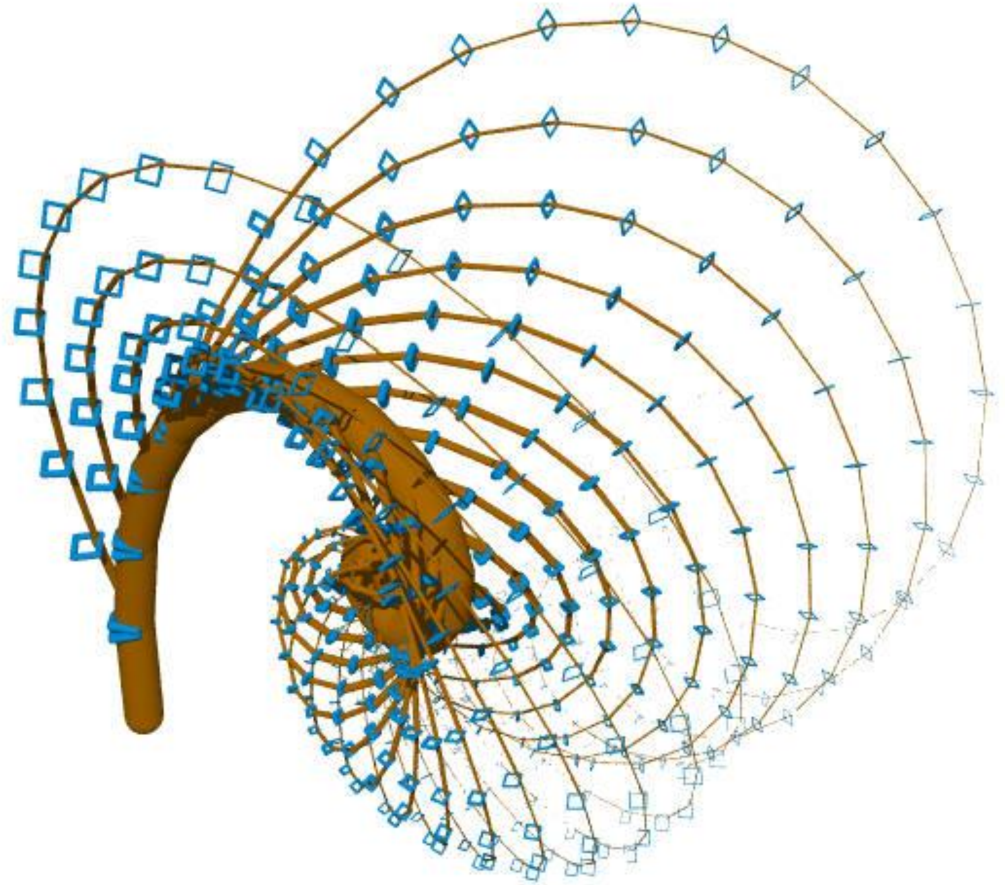


```
@i=4
@a=90
@l=1
@t=0.5
=A
A=c(#42844B)B-F+CFC+F-D&F^D-F+&&CFC+F+B>>
B=c(#9E755C)A&F^CFB^F^D^^-F-D^|F^B|FC^F^A>>
C=c(#416094)|D^|F^B-F+C^F^A&&FA&F^C+F+B^F^D>>
D=c(#8B844E)|CFB-F+B|FA&F^A&&FB-F+B|FC>>
@@
```

Examples

```
@i=38
@a=10
@l=1.5
@t=0.5
=>(110)PPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
P=S>(20)gg'
S=[&(90)[A]+(20)^(90)gg'&(90)[B][D]]
D=c(#914A3E)^FD
A=[^gAc(#828354){.].
B=B^.g.}
@@
```

Examples

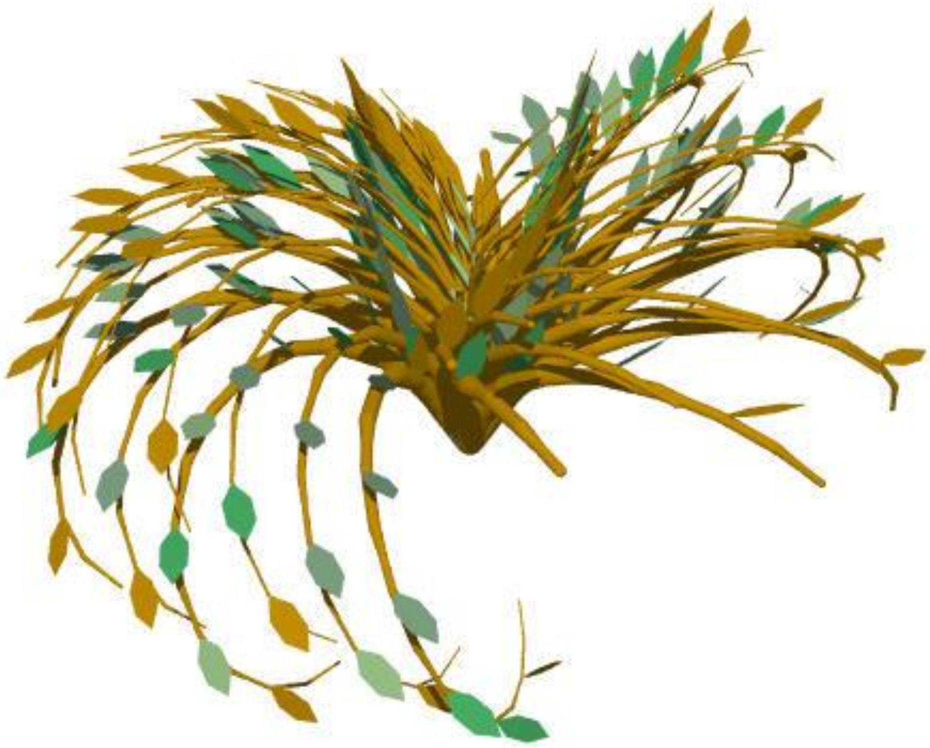


```
@i=20
@a=10
@l=3
@t=1
=a
a=Fs+;'a
s=[::!!!!&&[b]^^^^[b]]
b=mF!+mF+;'b
m=[c(#0066FF)&(90)f(0.05)+(135)F(0.2)+(90)F(0.2)+(90)F(0.2)+(90)F(0.2)]
@@
```

Examples

```
@i=11
@a=30
@l=3
@t=1
=&&&^D
D=AB!'Dc(#864740)F(0.5)O
B=[c---'''!:D]
A=&+^FLA
L=[---c(#6C8938){.-f+f+f-|-f+f+f}]
O=[c(#FF0000)C!iw>>w>>w>>w>>w>>w]
w=[&{.-f+f|-f+f}]
i=;i
C=(0.3)c(#9900FF)
C=(0.3)c(#CCFF00)
C=(0.3)c(#FF3399)
@@
```

Examples



```
@i=10
@a=10
@l=1
@t=0.8
=C
C=[c(#CC6600)H]f!>C
H=[^^^G]>(180)[^^^G]>>>H
G=FF!^+FF^+;'[&&&L]G
L=[X{--f++f++f--|--
f++f++f}]
X=(0.33)c(#508954)
X=(0.33)c(#73BF59)
X=(0.33)c(#169C2E)
@@
```

bop02

```
7
22.5
80
c(12)A
A=[&FL!A]>>>>>[&FL!A]>>>
>>>>[&FL!A]
F=S>>>>>F
S=FL
L=[c(8)^^{-f+f+f-|-
f+f+f}]
@
```

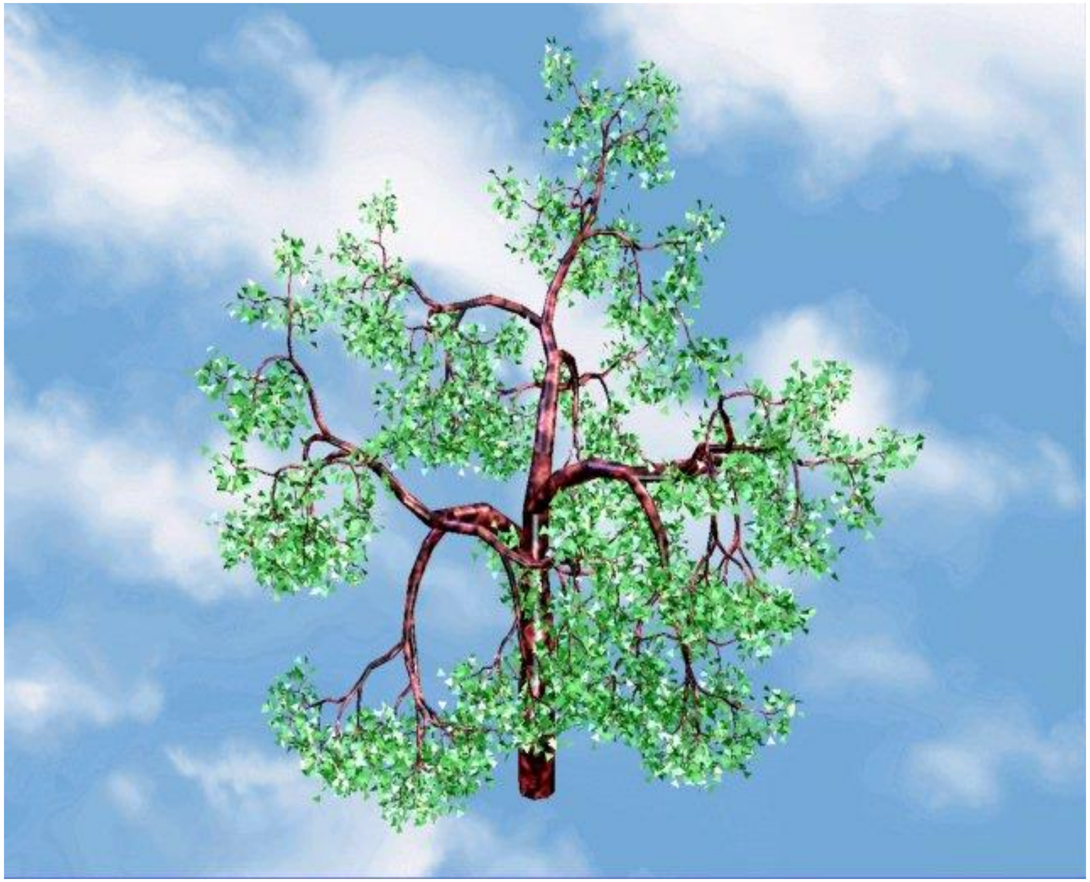
fern

```
20 # recursion
20 # angle
15 # thickness
# axioms
c(12)b>(60)b>(60)b>(60)b>(60)b>(60)b
# rules
b=[&(30)A]
A=~(7)$t(.1)F[+(40)C][-
(40)C]!(.95)~(7)t(.1)FA
C=~(10)$tF[+(60)L][-(60)L]C
L=[~(15)c(4){-f+f+f-|-f+f}]
F='(1.3)F'(.77)
f='(1.3)f'(.77)
@
```

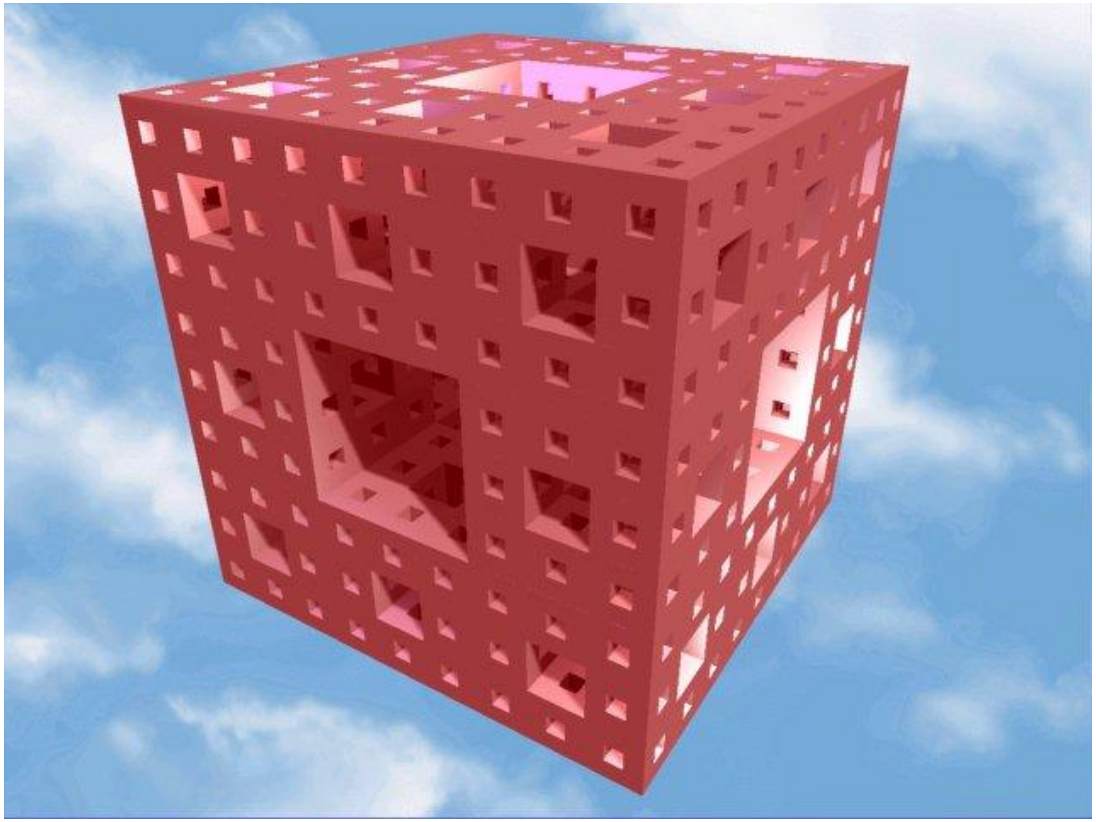
tree10

```
15
20
50
c(12)T
T=CCA
A=CBD>(94)CBD>(132)BD
B=[&CDCD$A]
D=[g(50)Lg(50)Lg(50)Lg(50)Lg(50)Lg(50)L]
C=!(.95)~(5)tF
F='(1.25)F'(.8)
L=[~f(200)c(8){+(30)f(200)-(120)f(200)-(120)f(200)}]
f=z
z=_
@
```

tree10 + gravity

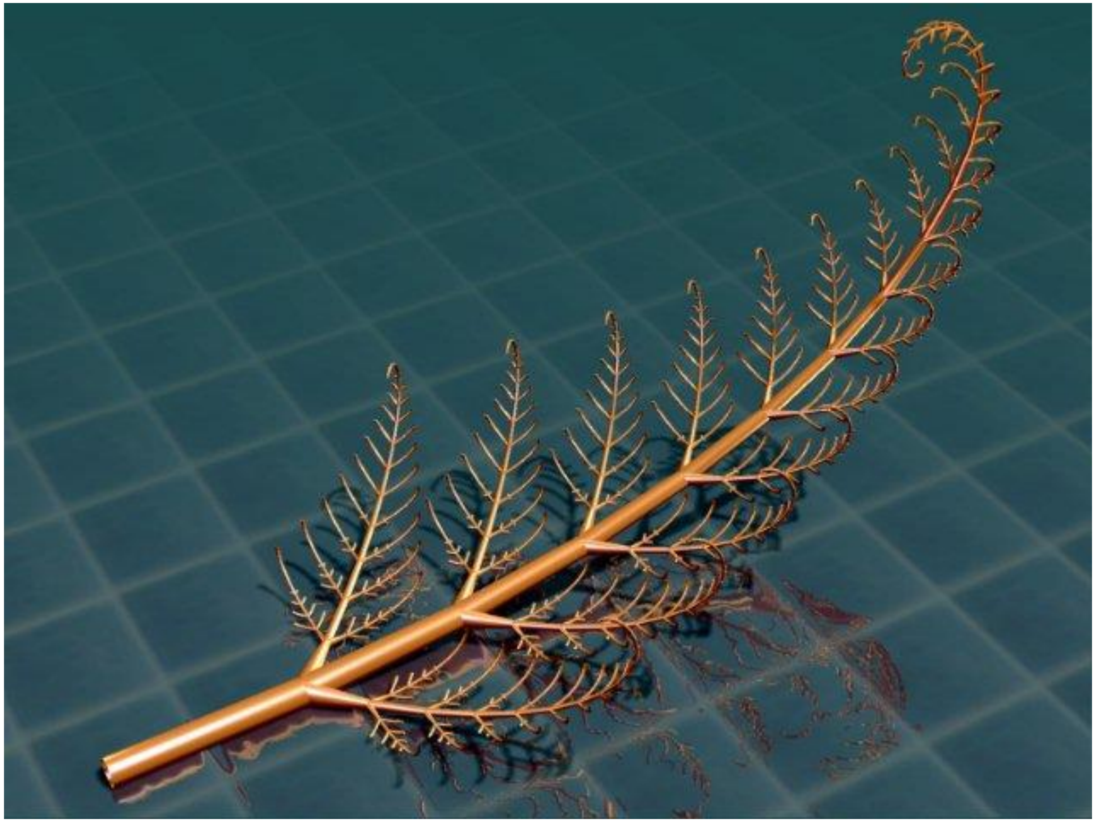


menger



```
4
90
142.857
F
F=["(.333333)[-f+&f^B]]f
B=[FFF|z|+zFF|z|+zFF|z|+zF]^f&[FfF|z|+zfF|z|+zfF]^f&[FFF|z|+zFF|z|+zFF|z|+zF]
@
```

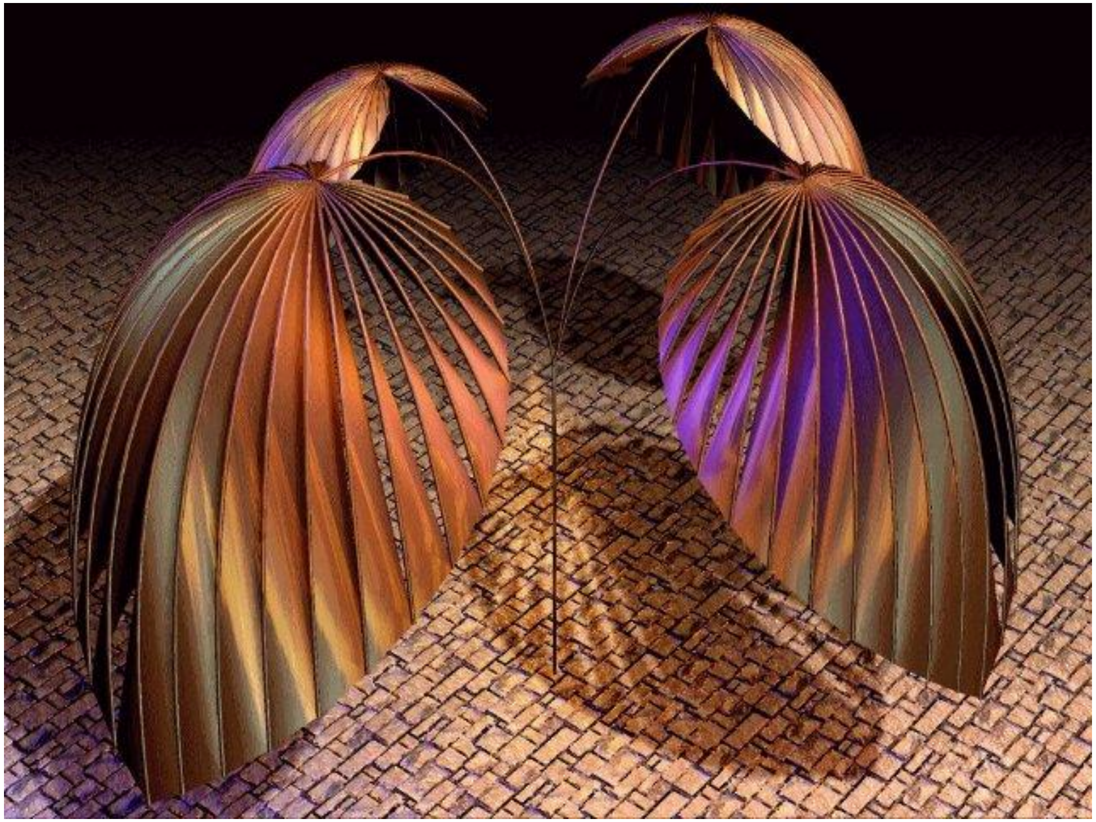
bop08



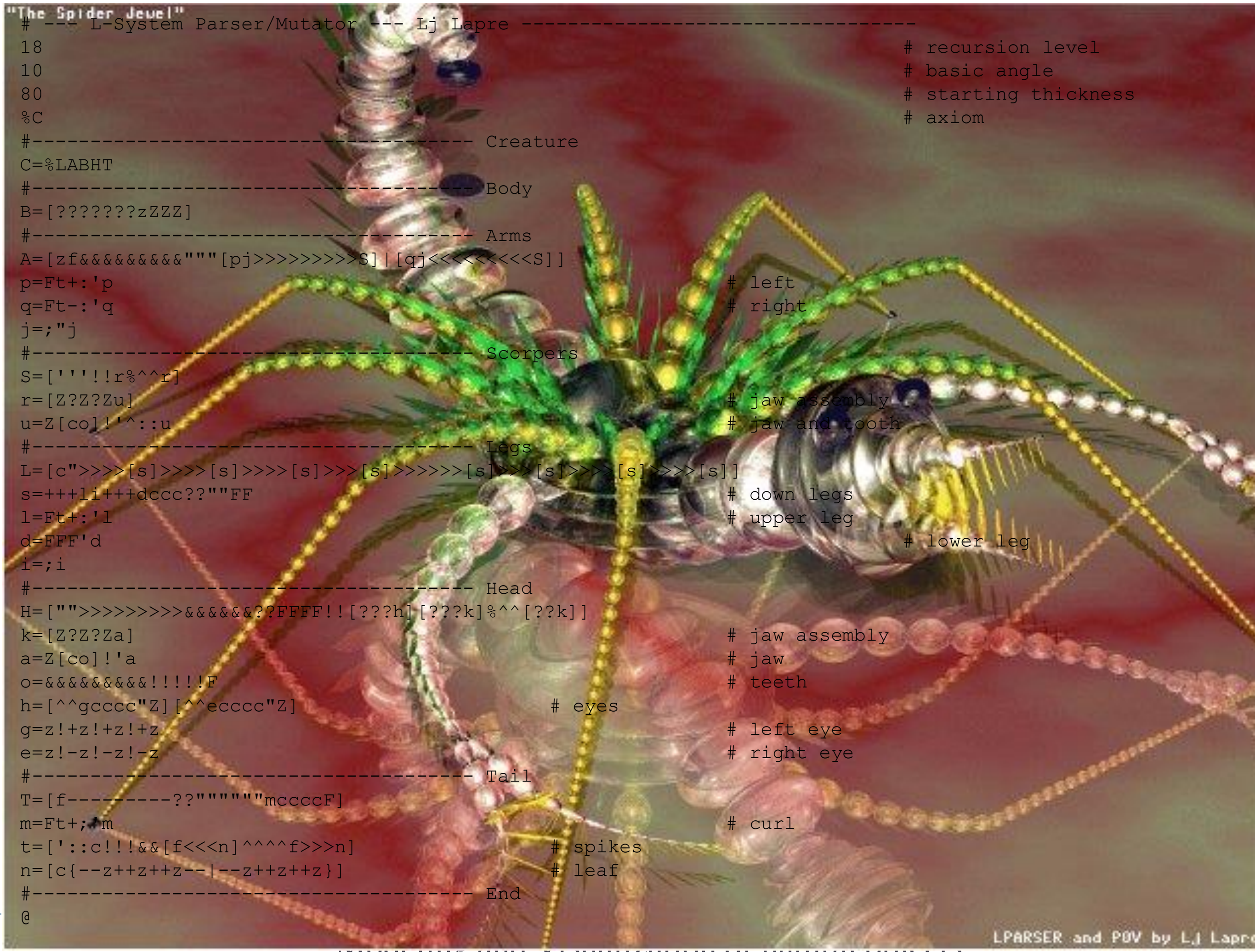
```
29
5
10
#
&(90)+(90)a
#
a=F[+(45)l][-(45)l]^;a
#         apical delay
l=j
j=h
h=s
s=d
d=x
x=a
#
F='(1.17)F'(.855)
#internode elongation rate
```

leaves

```
18        # recursion depth
10        # angle
10        # thickness as % of
length
#
#P
                    # one
leave
[|FFFFFFFFFF]P>(90)'P>(90)'P
>(90)'P   # plant
#
P=[&(10)G[ccA][ccB][a][b]]
G=tFtFtFtFtFtFtFtFtFtFtFtFtF
#
A=[+A{.].C.}
B=[-B{.].C.}
C=tfC
#
a=[+a]d
b=[-b]d
d=tFd
@
```



spider



"The Spider Jewel"
--- L-System Parser/Mutator --- Lj Lapre ----------------------------------
18 # recursion level
10 # basic angle
80 # starting thickness
%C # axiom
#--------------------------------------- Creature
C=%LABHT
#--------------------------------------- Body
B=[???????zZZZ]
#--------------------------------------- Arms
A=[zf&&&&&&&&&""""[pj>>>>>>>>>S]|[qj<<<<<<<<<S]]
p=Ft+:'p # left
q=Ft-:'q # right
j=;"j
#--------------------------------------- Scorpers
S=['''!!r%^^r]
r=[Z?Z?Zu] # jaw assembly
u=Z[co]!'^::u # jaw and tooth
#--------------------------------------- Legs
L=[c">>>>[s]>>>>[s]>>>>[s]>>>[s]>>>>>>[s]>>>[s]>>>>[s]>>>>[s]]
s=+++li+++dccc??""FF # down legs
l=Ft+:'l # upper leg
d=FFF'd # lower leg
i=;i
#--------------------------------------- Head
H=[""">>>>>>>>>&&&&&&??FFFF!![???h][???k]%^^[??k]]
k=[Z?Z?Za] # jaw assembly
a=Z[co]!'a # jaw
o=&&&&&&&&&!!!!!F # teeth
h=[^^gcccc"Z][^^ecccc"Z] # eyes
g=z!+z!+z!+z # left eye
e=z!-z!-z!-z # right eye
#--------------------------------------- Tail
T=[f---------??"""""""mccccF]
m=Ft+;'m # curl
t=['::c!!!&&[f<<<n]^^^^f>>>n] # spikes
n=[c{--z++z++z--|--z++z++z}] # leaf
#--------------------------------------- End
@
LPARSER and POV by Lj Lapre

Example: ecosystem modeling

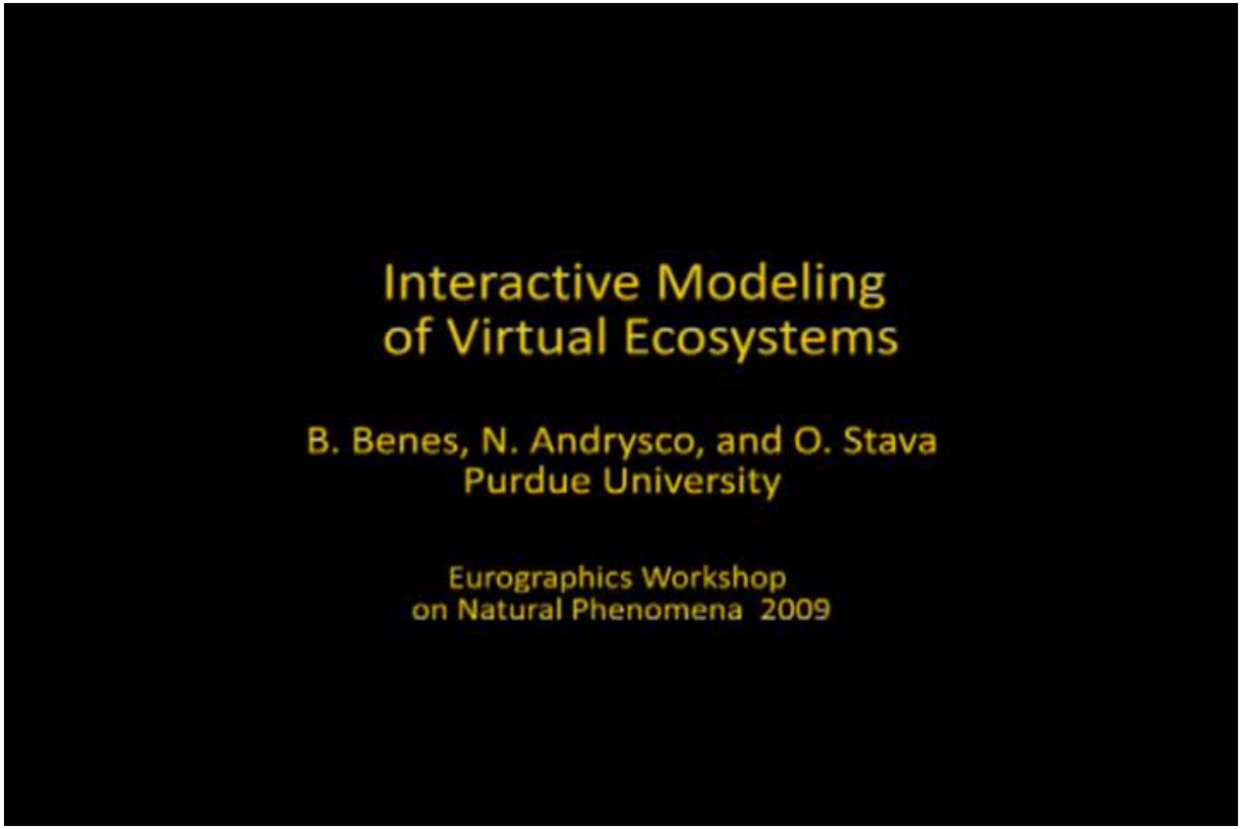
Drawbacks of L-systems

- Difficulties in controlling the shape of the plants. Changes in productions can lead to unpredictable results.
- Difficulties in adjusting plants to constrains and environment conditions.

- The above causes a search of new methods.

Bedrich Benes

- Interactive Modeling of Virtual Ecosystems, 2009



Xfrog model

- X Window Finite Recursive Object Generator
- Greenworks
- Maya

www.greenworks.de


(c) Bernd Lintermann 1995


(c) Bernd Lintermann 1995

Trees in Blender?

- Sapling add-on

Trees in WebGL?

- www.snappytree.com

- L-systems in WebGL, L-systems in Three.js give us a number of links

Bibliography

- P. Prusinkiewicz, A. Lindenmayer, *The Algorithmic Beauty of Plants*
- Lparser http://home.wanadoo.nl/laurens.lapre/