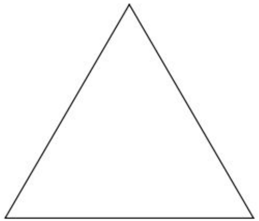
Graphical modeling using L-systems

Rewriting systems

- Axiom

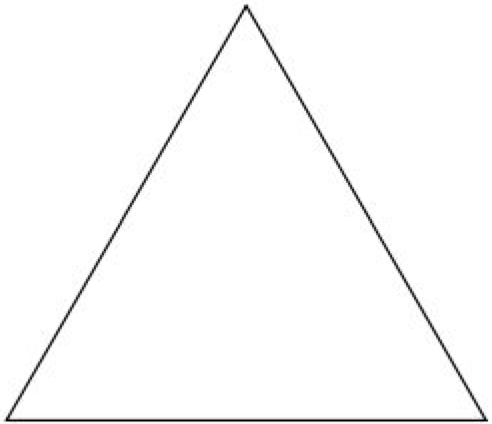


initiator

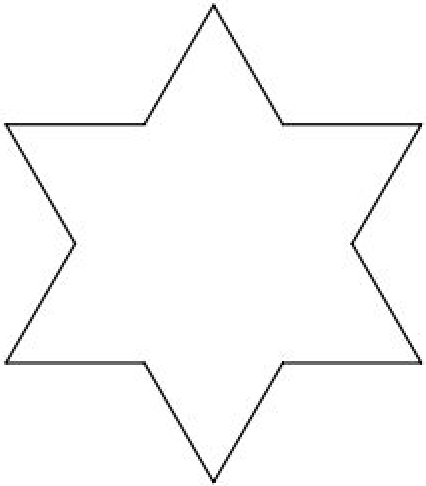
- Productions

generator

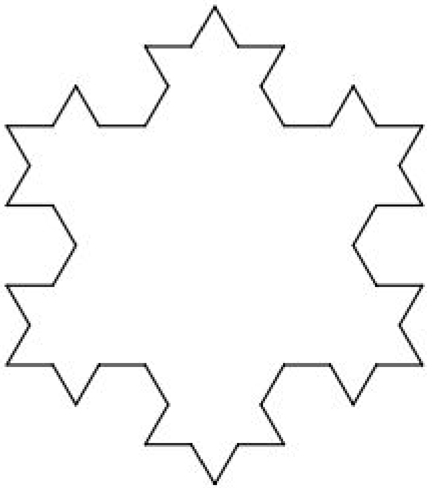
0 iterations



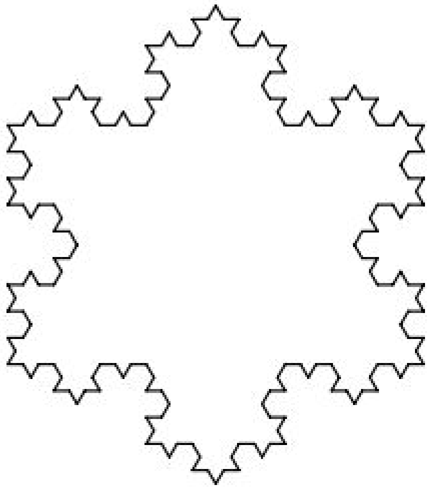
1 iteration



2 iterations



3 iterations

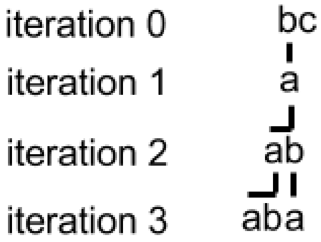


L-system representation

- An alphabet *V* (eg. {a, b, c})
- An axiom ω (eq. "ababc")
- One or more productions *P* (eg. {a→b, b→ba})
- Terminals and non-terminals

An example

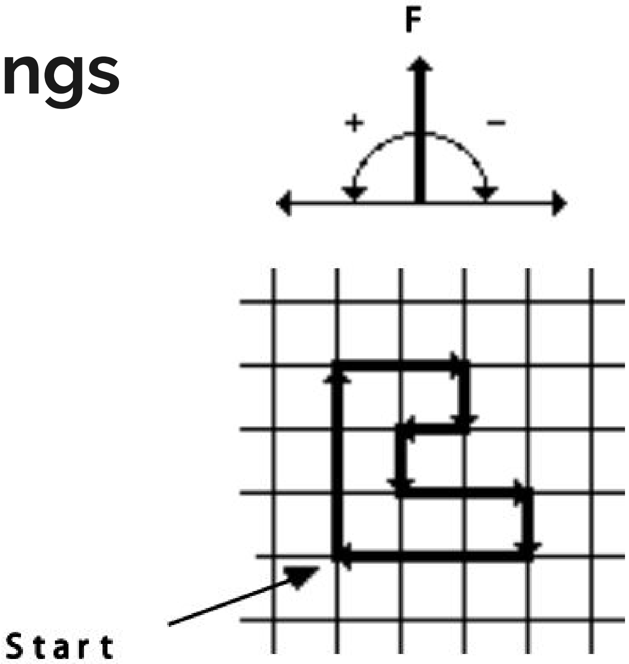
Alphabet V = {a, b, c}
Axiom ω = “bc”
Productions P = {a→ab, b→a}



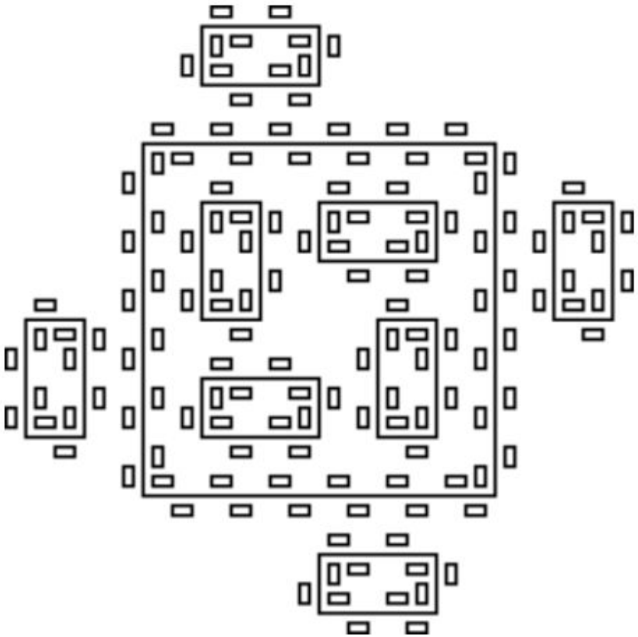
Turtle interpretation of strings

Alphabet V = {F, f, +, -}

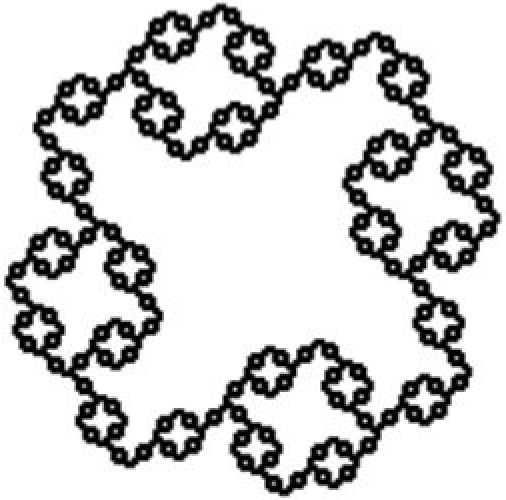
F: Move forward a length d and draw a line
f: Move forward a length d without drawing a line
+: Rotate to the left by an angle δ
-: Rotate to the right by an angle δ



FFF-FF-F-F+F+FF-F-FFF



$n = 2, \ \delta = 90^\circ$
F+F+F+F
F → F+f-FF+F+FF+Ff+FF-f+FF-F-FF-Ff-FFF
f → ffffff



$n = 4, \ \delta = 90°$

F-F-F-F

$F \rightarrow$ FF-F-F-F-F-F+F

Moving on to three dimensions

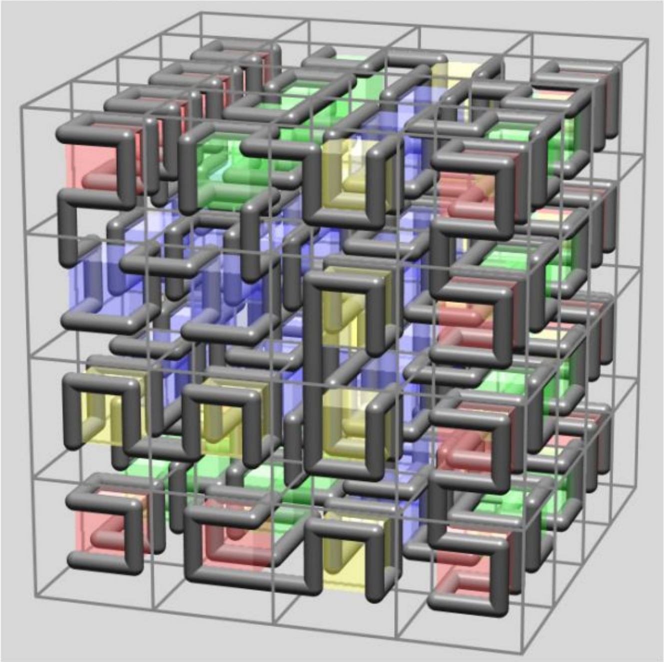
Alphabet V = {F, f, +, -, &, ^, \, /, |}

+: Turn left using rotation matrix $R_U(\delta)$
-: Turn right using rotation matrix $R_U(-\delta)$
&: Pitch down using rotation matrix $R_L(\delta)$
^: Pitch up using rotation matrix $R_L(-\delta)$
\: Roll left using rotation matrix $R_H(\delta)$
/: Roll right using rotation matrix $R_H(-\delta)$
|: Turn around using rotation matrix $R_U(180°)$

$$\mathbf{R_U}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R_L}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

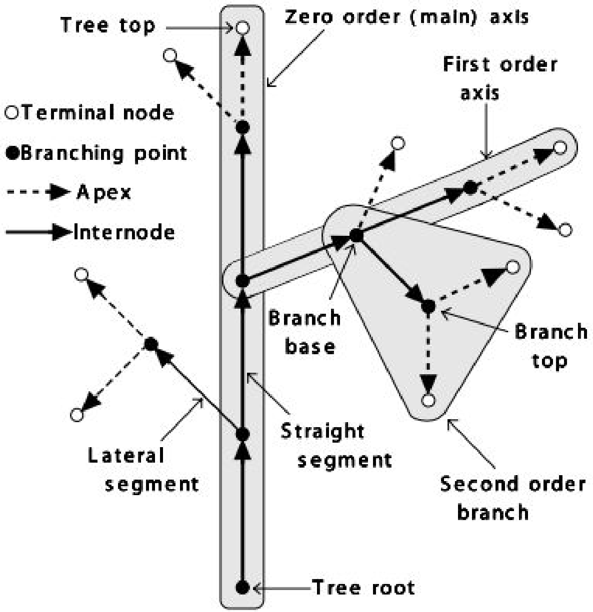
$$\mathbf{R_H}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$



```
n=2, δ=90°
A
A → B-F+CFC+F-D&F∧D-F+&&CFC+F+B//
B → A&F∧CFB∧F∧D∧∧-F-D∧|F∧B|FC∧F∧A//
C → |D∧|F∧B-F+C∧F∧A&&FA&F∧C+F+B∧F∧D//
D → |CFB-F+B|FA&F∧A&&FB-F+B|FC//
```

Branching structures

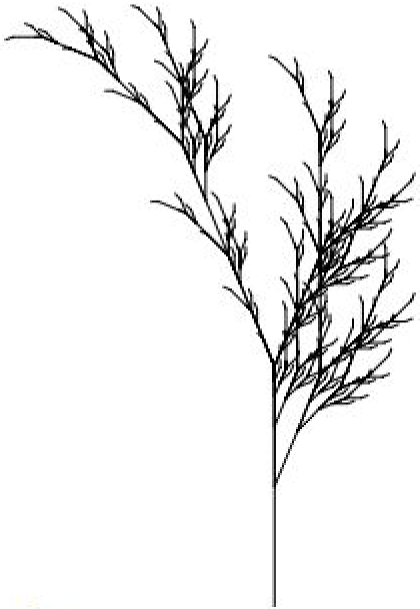
- A dominating behaviour in the plant kingdom
- Axial trees



Branching in turtle representation

To more letters to our alphabet: [and], *V* = {F, f, +, -, &, ^, \, /, |, [,]}

[: Push the state of the turtle on a stack
]: Pop a state of the turtle from the stack



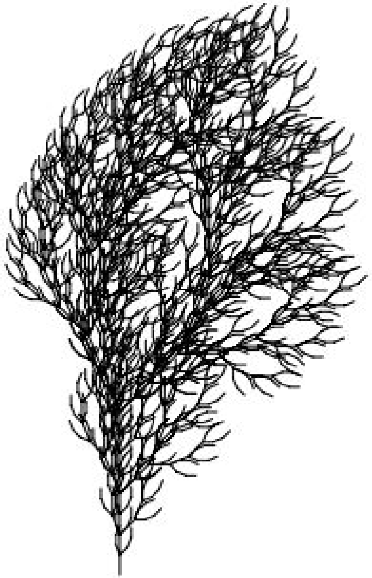
f

n=5, δ=22.5°

X

X→F-[[X]+X]+F[+FX]-X

F→FF

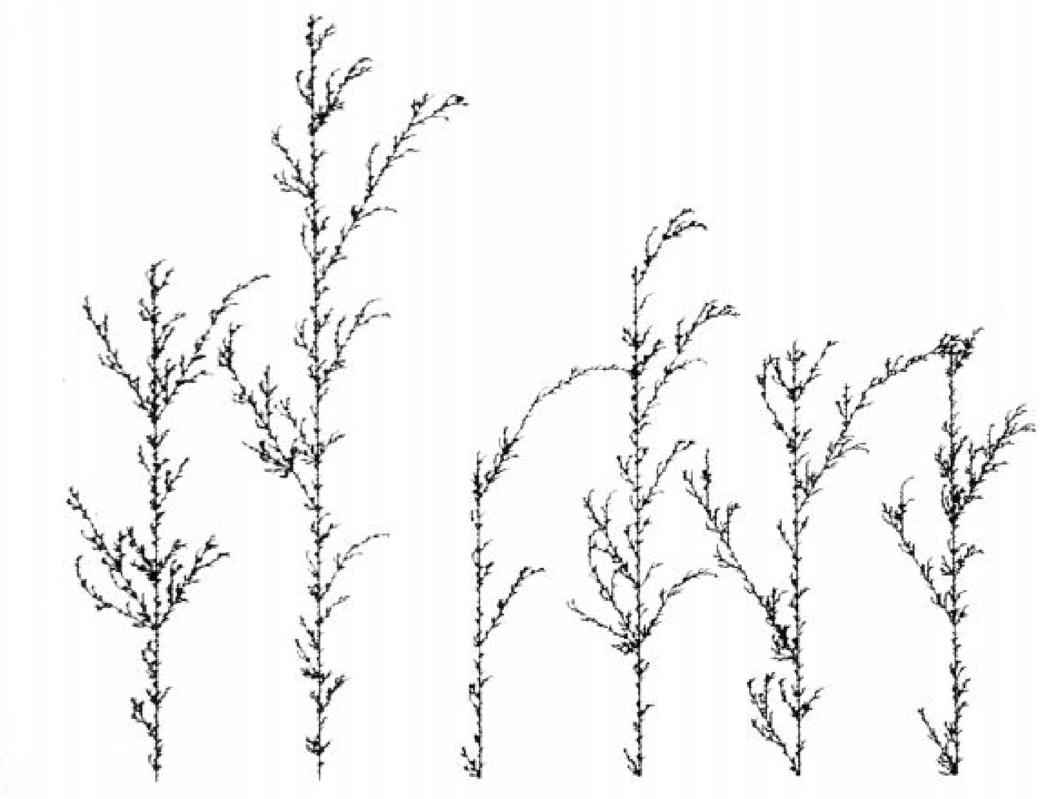


c
n=4, δ=22.5°
F
F→FF-[-F+F+F]+
[+F-F-F]

Stochastic L-systems

- Running a non-stochastic L-system will always produce the same geometry
- Add a probability to the selection of a production
- Can model different specimens of the same plant species

$$
\begin{aligned}
\omega &: \quad F \\
p_1 &: \quad F \xrightarrow{.33} F[+F]F[-F]F \\
p_2 &: \quad F \xrightarrow{.33} F[+F]F \\
p_3 &: \quad F \xrightarrow{.34} F[-F]F
\end{aligned}
$$



Context-sensitive L-systems

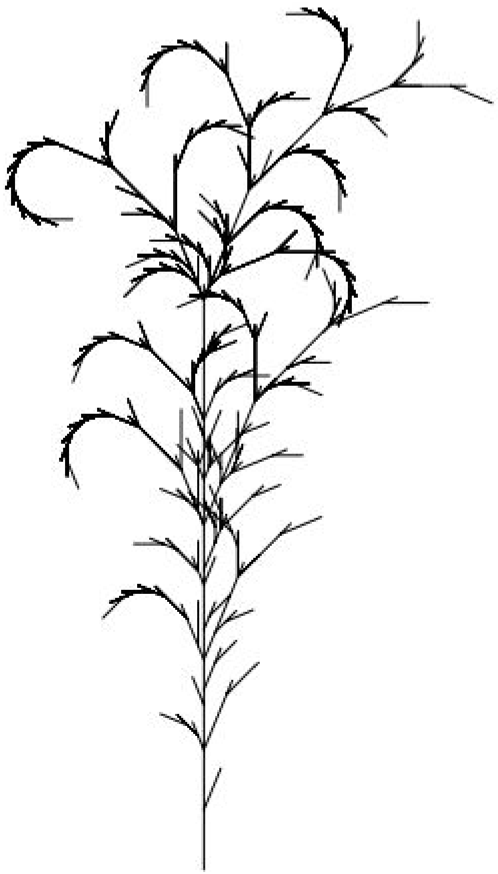
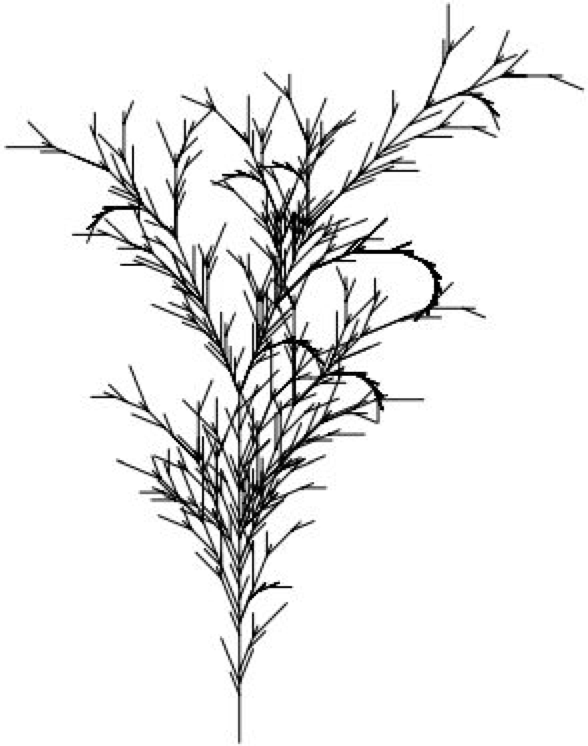
- Up until now we have only seen context-free grammars
- Useful in simulating interactions between plant parts

1L-systems

$$a_l < a \rightarrow \chi \text{ or } a > a_r \rightarrow \chi.$$

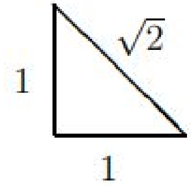
2L-systems

$$a_l < a > a_r \rightarrow \chi.$$

Parametric L-systems

- Can change the length d and angle δ throughout the generation of a string
- Parameters are stored with the letters
- Creates a continuous system



Turtle interpretation of parametric L-systems

F(a): Move forward a length a and draw a line
f(a): Move forward a length a without drawing a line
+(a): Turn using rotation matrix $R_U(a)$
&(a): Pitch using rotation matrix $R_L(a)$
/(a): Roll using rotation matrix $R_H(a)$

$$\begin{array}{lll} \#\text{define} & c & 1 \\ \#\text{define} & p & 0.3 \\ \#\text{define} & q & c - p \\ \#\text{define} & h & (p * q) \wedge 0.5 \end{array}$$

$$\begin{array}{ll} \omega : & F(1) \\ p_1 : & F(x) \rightarrow F(x*p) + F(x*h) - -F(x*h) + F(x*q) \end{array}$$