L-systems

I.Kolingerová

1. L-systems - introduction
2. d0L-system
3. Fractals and graphical interpretation of strings
4. Bracketed L-systems and plants models
5. Open L-systems
6. Plants simulation

Literature:

- Francis S.Hill Jr.: Computer Graphics, Macmillan Publishing Company, New York, 1990
- H.A. Lauwerier, J.A. Kaandrop: Fractals (Mathematics, Programming and Applications), TR CS-R8762, Centre for Mathematics and Computer Science, Amsterdam, The Nietherlands, 1980
- G.Ochoa: An Introduction into Lindenmayer Systems, http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html

P.Prusinkiewicz, M. Hammel, R. Mech: Visual Models of Morphogenesis: A Guided Tour, http://www.cpsc.ucalgary.ca/Research/bmv/vmm/title.html

L-systems - introduction

- Mathematical formalism – A.Lindenmayer, the biologist, 1968
- Applications also in computer graphics, namely for fractal generation, realistic modelling of plants, streams, sea shells
- Central idea: replace simple object parts by more complex ones using rewriting rules, replacement can be recursive
- Best studied – rewriting systems for strings – Chomski work on formal grammards, 1957 => big interest in these systems => formal languagues

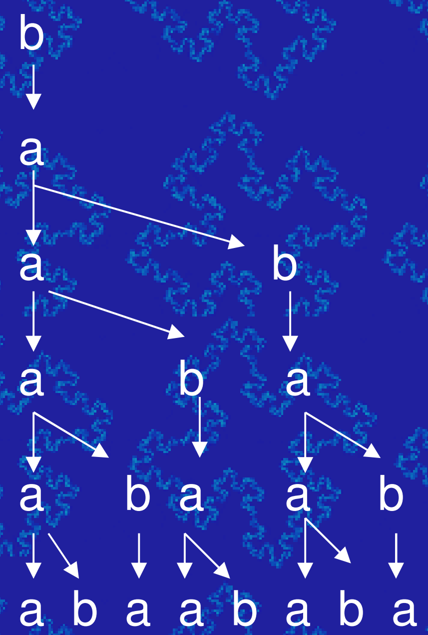
- Lindenmayer – a new type of rewriting system - L-system
- Difference to Chomski – rewriting rules at Ch. applied sequentially while in L-system in parallel – all characters in a word are replaced at the same time
- The difference reflects the biological motivation of L-systems – cell division in multicell organisms where many divisions run at the same time
- L-system: K=<G,W,P>, where G-a set of symbols, W-a set of starting symbols (axioms), P-rewriting (production) rules: A->B, A from G,B from G*

d0L-system

- The simplest L-system - deterministic (i.e., P cannot contain 2 rules with the same left side), contextless (i.e., a symbol is rewriten in the same way in any context)
- Ex.: strings from 2 characters a,b (they can be repeated in the string), for each character a rewriting rule, starts from the axiom

a->ab

b->a



Fractals and graphical interpretation of strings

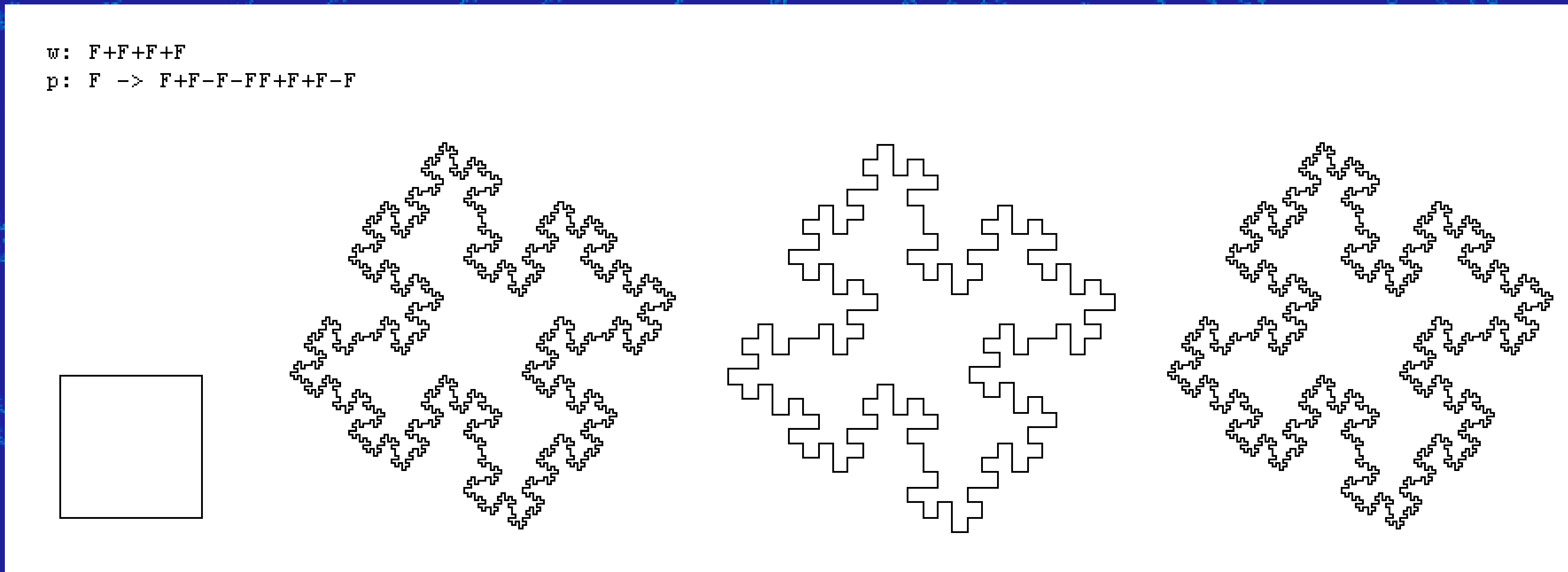
- L-systems originally understood as a mathematical theory of development, without geometrical aspects
- Then several geometrical interpretations for fractals and plant models proposed
- Many finite approximations of fractals can be understood as sequences of line segments
- Graphical interpretation of strings done using turtle graphics
- Can be done also in 3D

- Turtle status: (x,y,a), where (x,y) is the turtle position, a- turtle's view direction
- Given: step size d, angle increment b

- Commands:
  - F – d steps forward, turtle status is changed to (x+d cos a, y + d sin a, a), draw a line segment
  - f – d steps forward, turtle status is changed to (x+d cos a, y + d sin a, a), move
  - + turn left by angle b, the new status is (x,y,a+b)
  - - turn right by angle b, the new status is (x,y,a-b)
  - Other symbols are ignored by the turtle
- => mapping of strings to drawings, interpretation of strings generated by L-systems

Ex.: approximation of Koch island
w: F+F+F+F
p: F-> F+F-F-FF+F-F

It corresponds to the derivations of the length n=0,1,2,..., b=90°

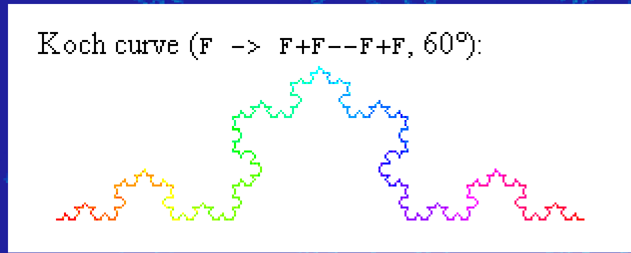


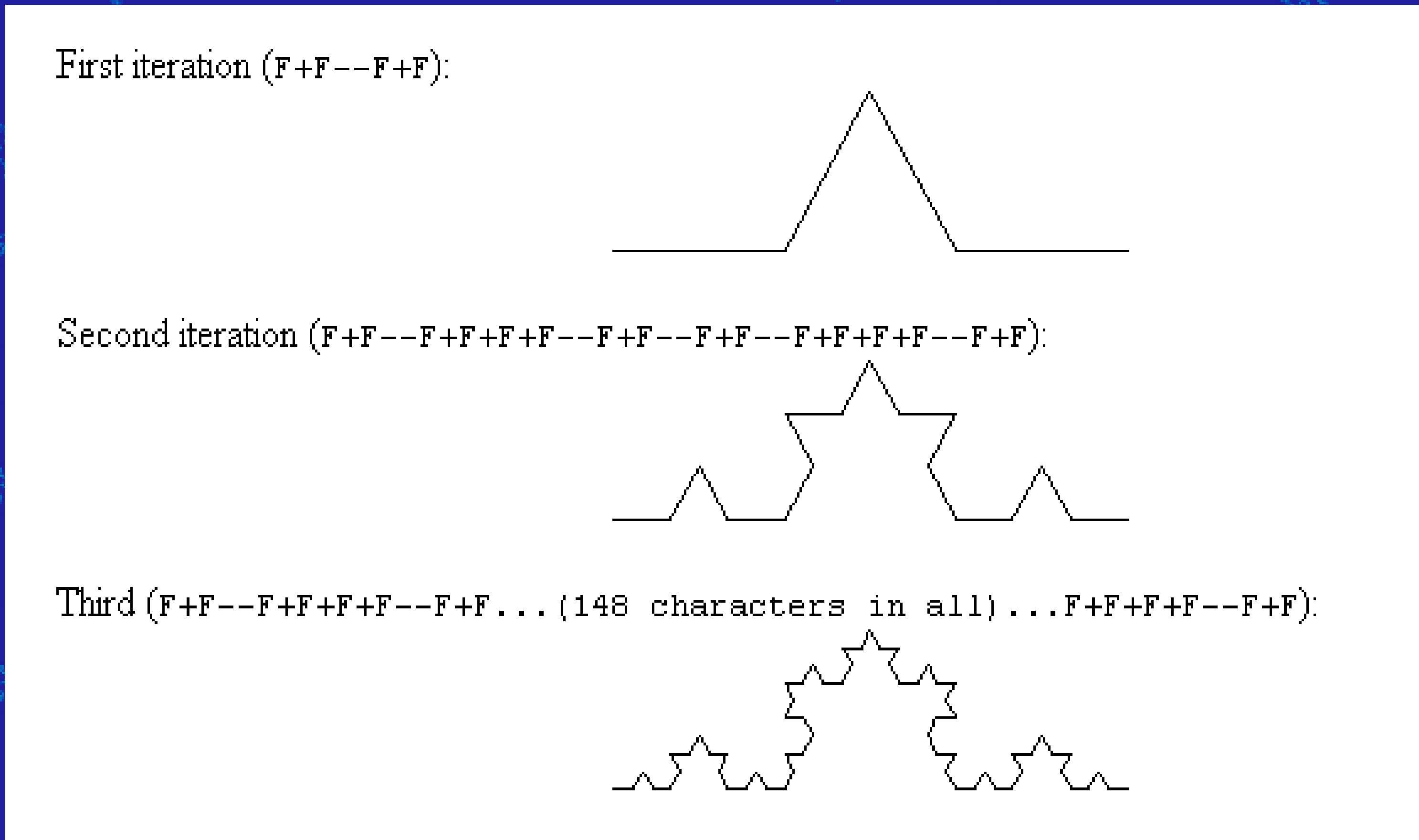
Ex.: approximation of the Koch curve

w: F

p: F-> F+F--F+F

60°



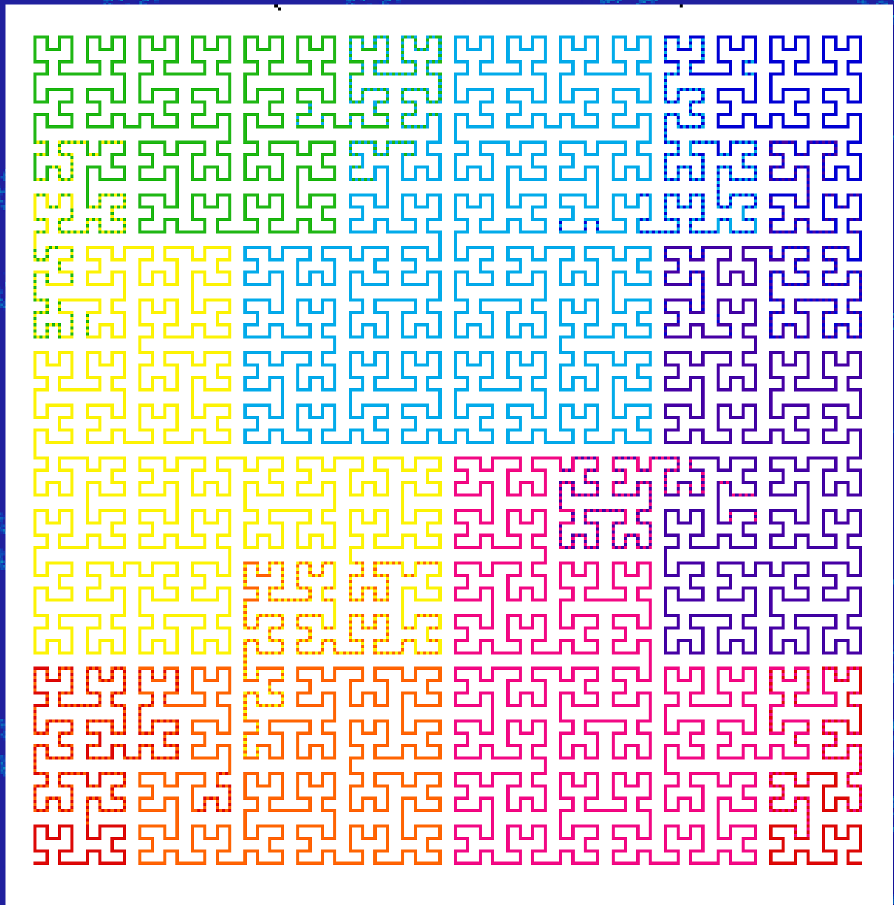


Ex.: approximation of the Hilbert curve

w: L,R

p1: L-> +RF-LFL-FR+

p2: R->-LF+RFR+FL-



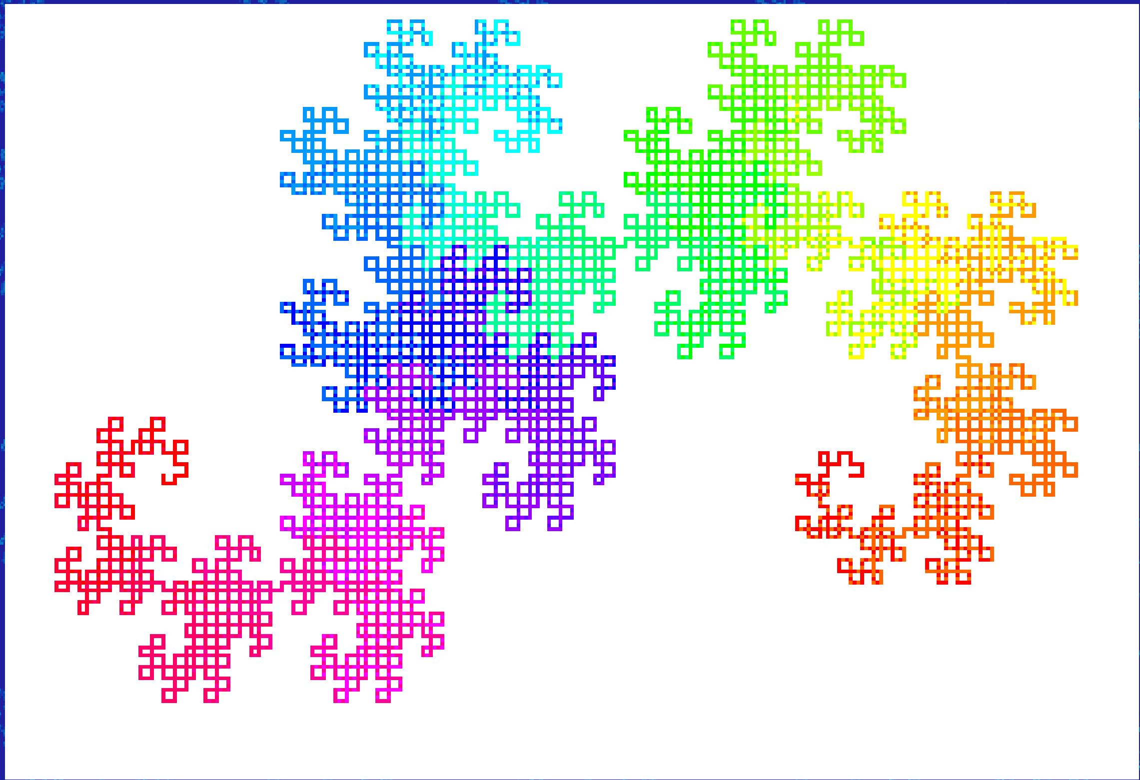
Coloring: as the drawing proceeds (red-amber-yellow-green-blue-purple-red)

Ex.: Dragon curve

w: X,Y

p1: X-> X+YF+

p2: Y->-FX-Y



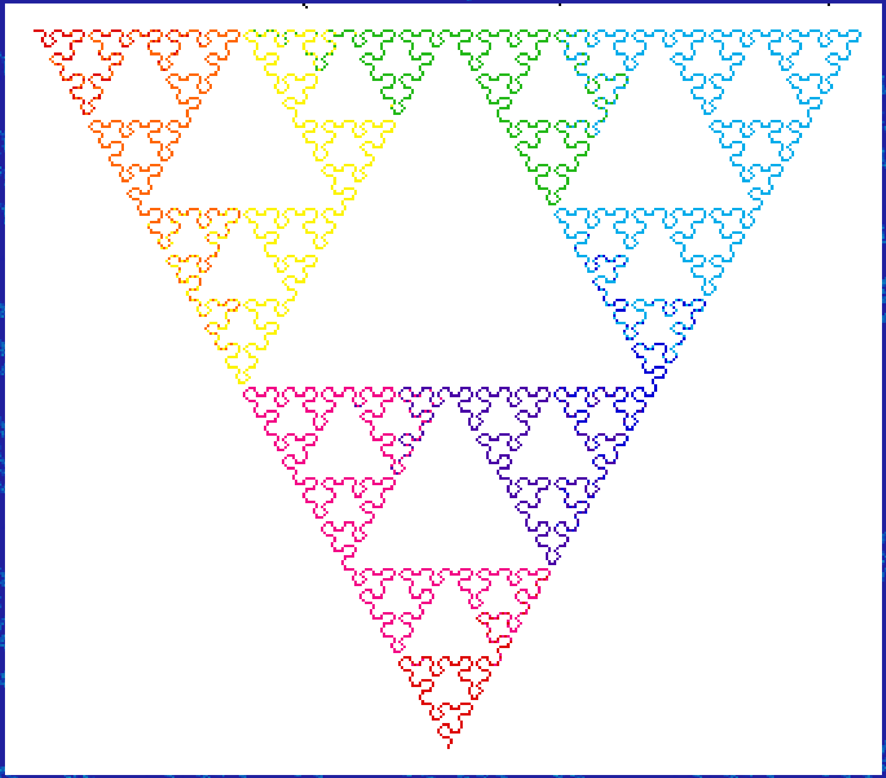
Ex.: Sierpinski triangle

w: X,Y

p1: X-> YF+XF+Y

p2: Y->XF-YF-X

60°



- Sometimes the turtle status given as: (x,y,**H**,**U**,**L**), where (x,y) is the turtle position, vectors **H** - heading – forward direction, **U**- up – upper direction, rectangular to the shell, **L**- left- on which side the turtle has left feet

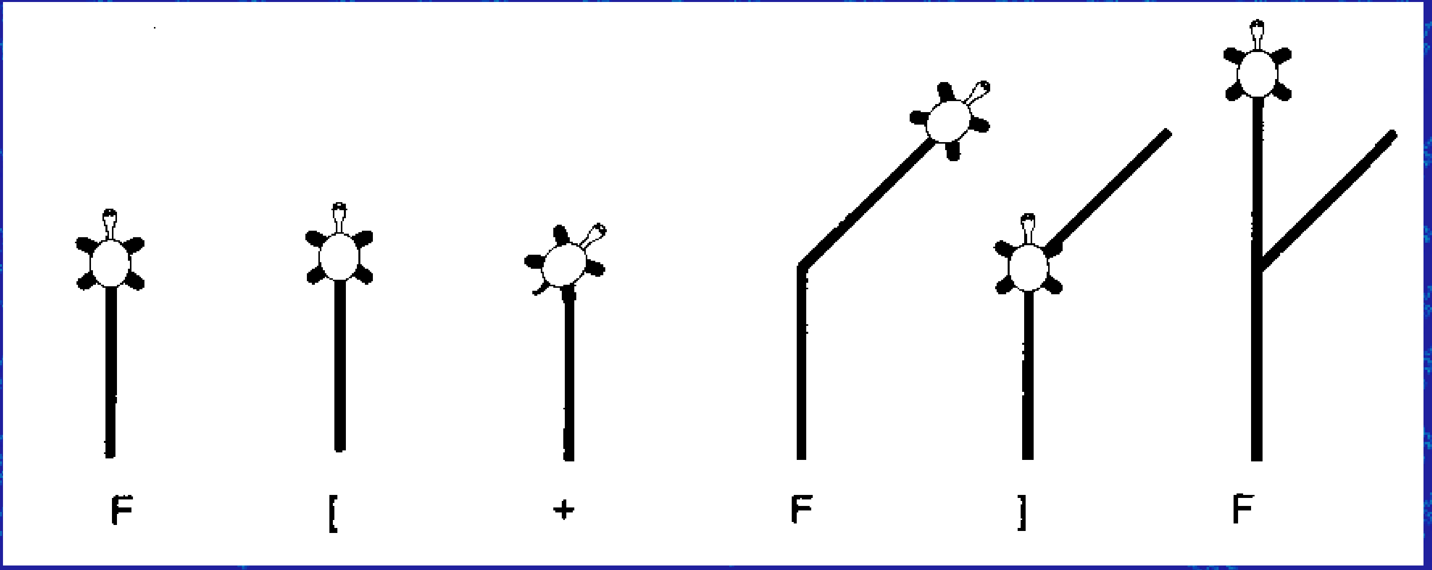
Bracketed L-systems and plant models

- Turtle interpretation of the string – a sequence of mutually connected line segments =>

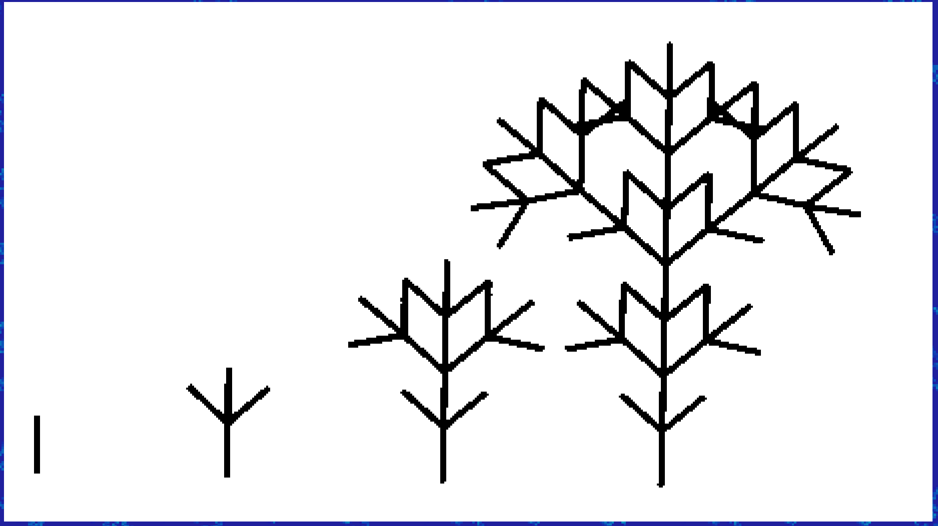
  a continuous drawing
- Branching can also be represented, bushes then contain strings with brackets (a bracket – an independent object part, a branch, which goes to the right or left)
- Other commands:
  - [- store the current turtle status to the stack
  -] - pick up the status from the stack and make it the current turtle status

Ex. F[+F]F, where +/- is 60° rotation right/left

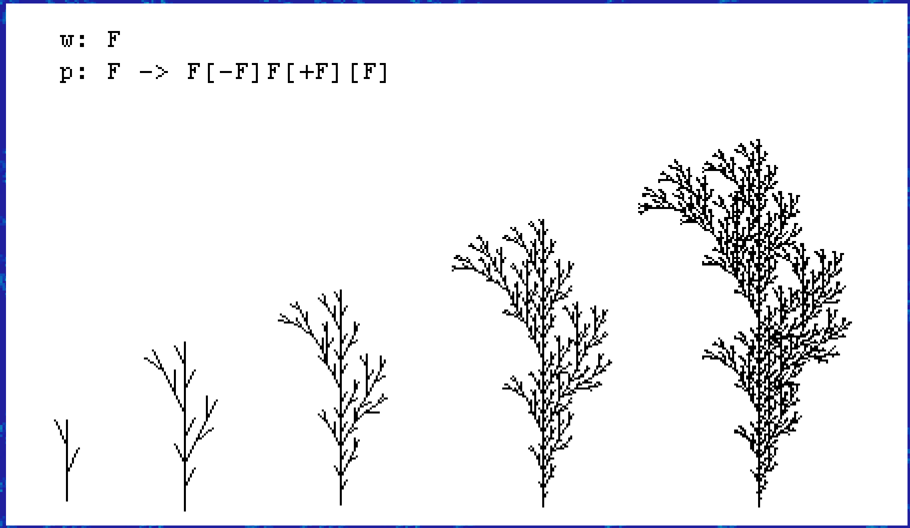
The turtle goes forward (F), stores its status ([),
turns right and goes forward,
reads the status from the stack(]), jumps back
and continues in the original direction.



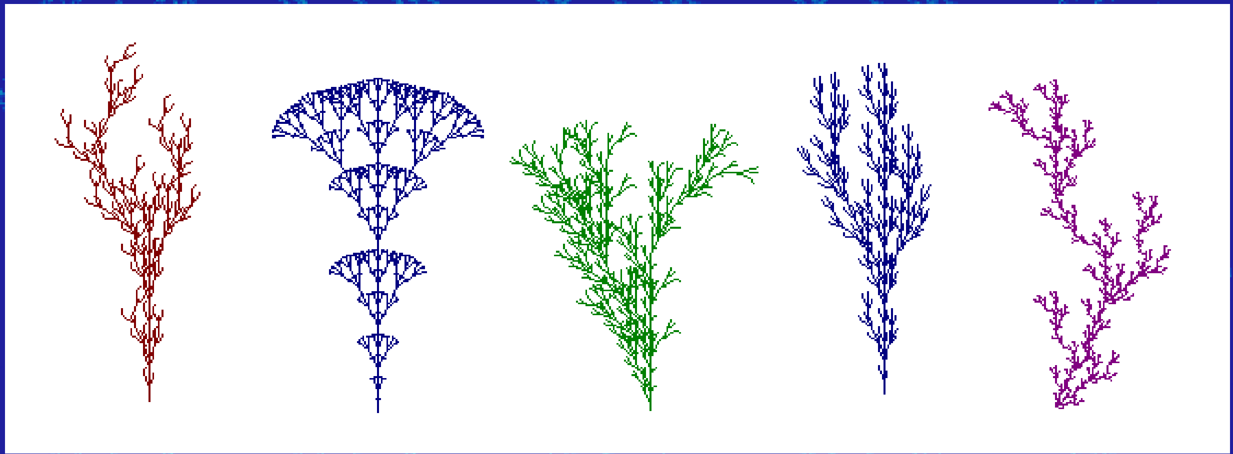
Ex. G={F,+,-,[,]}, W=F, P={F->F[+F][-F]F}



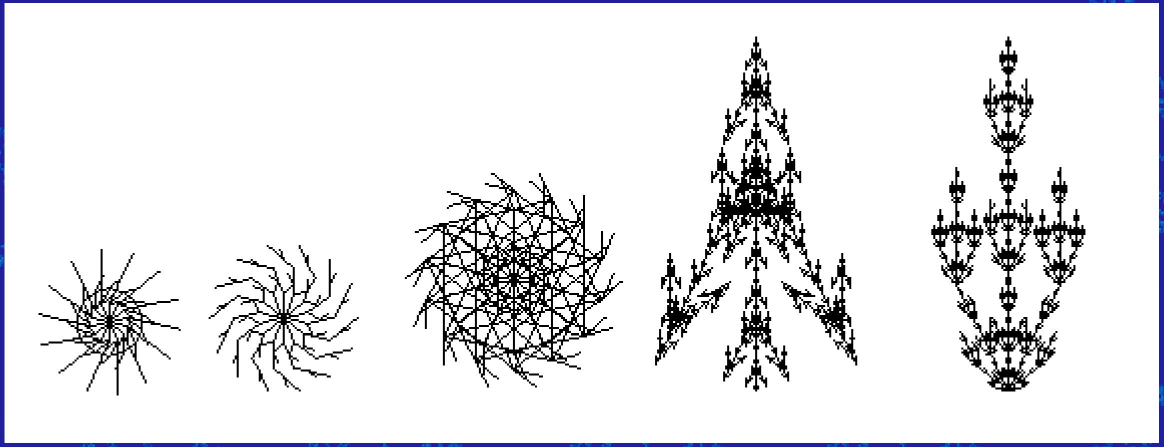
Ex.:
w: F
p: F-> F[-F]F[+F][F]



Ex. With the use of genetic algorithms with genotypes inspired by L-systems:



Ex. With the use of genetic algorithms, too, with genotypes inspired by L-systems, with a preference of bilateral symmetry:

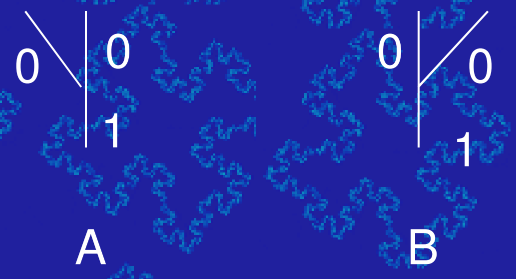


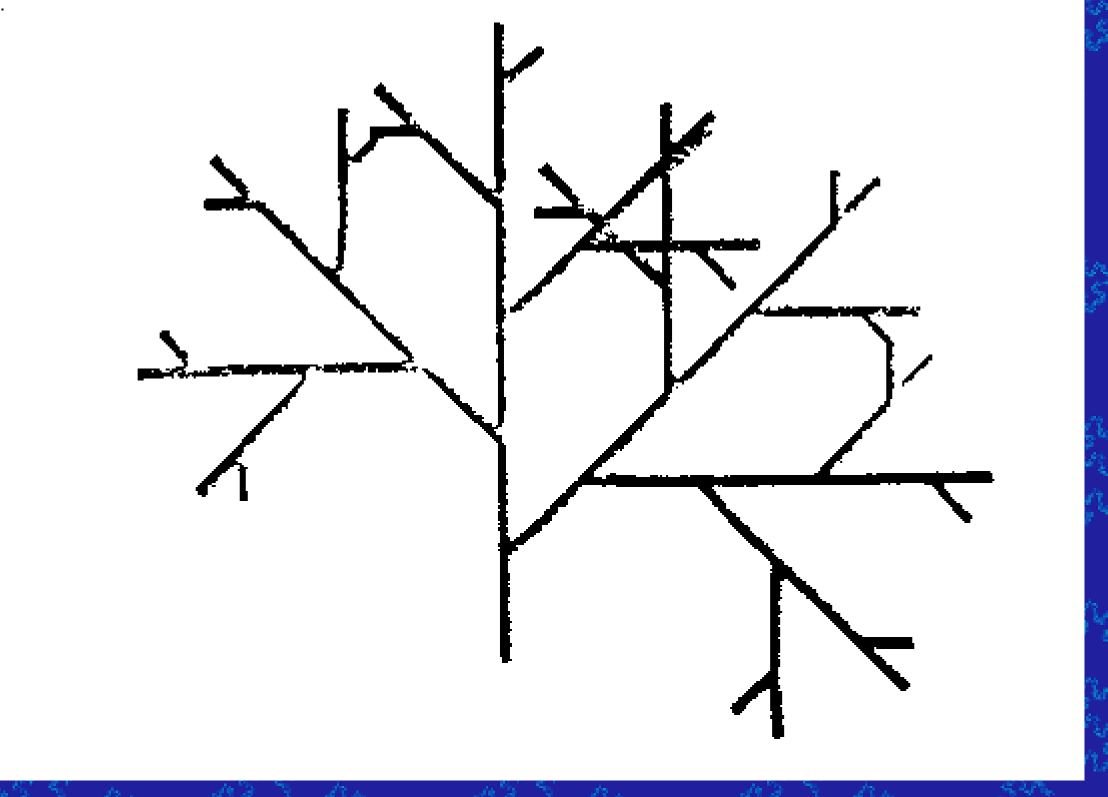
Monopodic branching (1 branch)

- KM=<G,WM,PM>
- G={0,1,[,]}, WM=0, PM={0->1[0]0,1->1,[->[,]->]}

iteration	generated string
1	0
2	1[0]0
3	1[1[0]0]1[0]0
4	1[1[1[0]0]1[0]0]1[1[0]0]1[0]0

Visualization, e.g. : the string 1[0]0 visualized once by A and once by B:





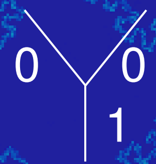
Visualization of the string, for monopodic branching, level 5

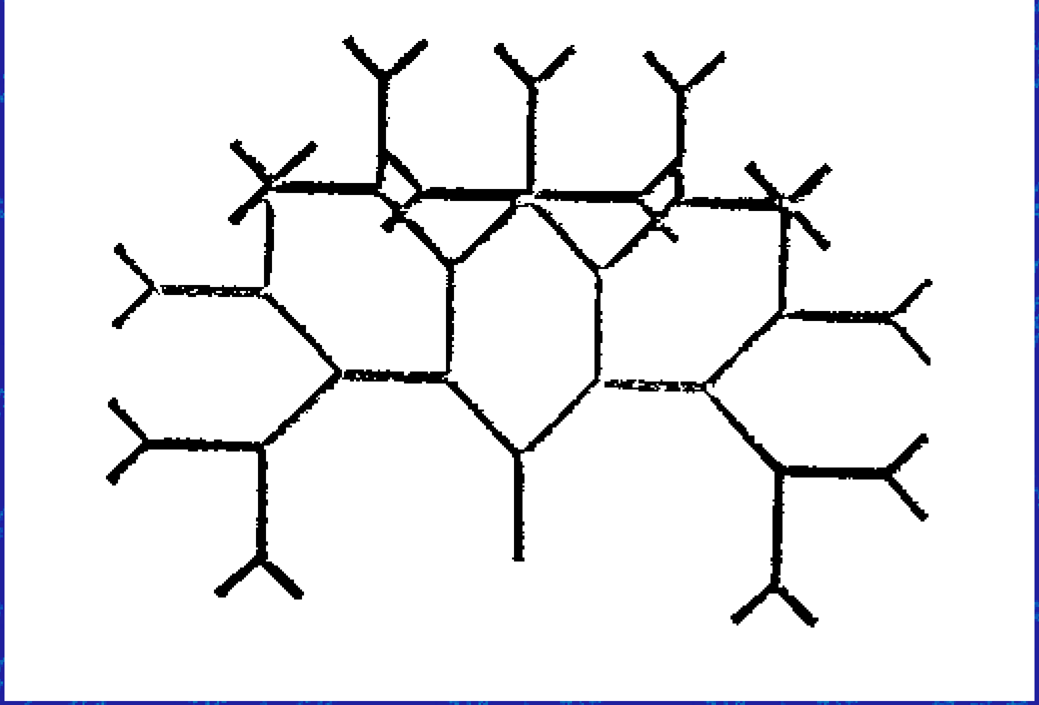
Dichotomic branching (2 branches)

- KD=<G,WD,PD>
- G-{0,1,[,]}, WD=0, PD={0->1[0][0],1->1,[->[,]->]}

iteration	generated string
1	0
2	1[0][0]
3	1[1[0][0]][1[0][0]]
4	1[1[1[0][0]][1[0][0]]][1[1[0][0]][1[0][0]]

Visualization, e.g.: the string 1[0][0] visualized by the shape:





Visualization of the string, for dichotomic branching, level 5

Open L-systems

- Open non-deterministic context parametric L-systems, designed mainly for simulation of growth of synthetic plant models
- Possibility to propagate biological signals from roots to leaves and back
- Possibility to interact with the neighbourhood (in both directions) –
  - in – information for the rewriting process about collision detection with obstacles, amount of light, soil acidity, insects presence,...,
  - out – information for the neighbourhood about plant space spread, amount of produced chemicals (e.g., CO_2)...

Open L-systems

- 3 necessary generalizations - stochastic, context, parametric systems

1. Stochastic L-system

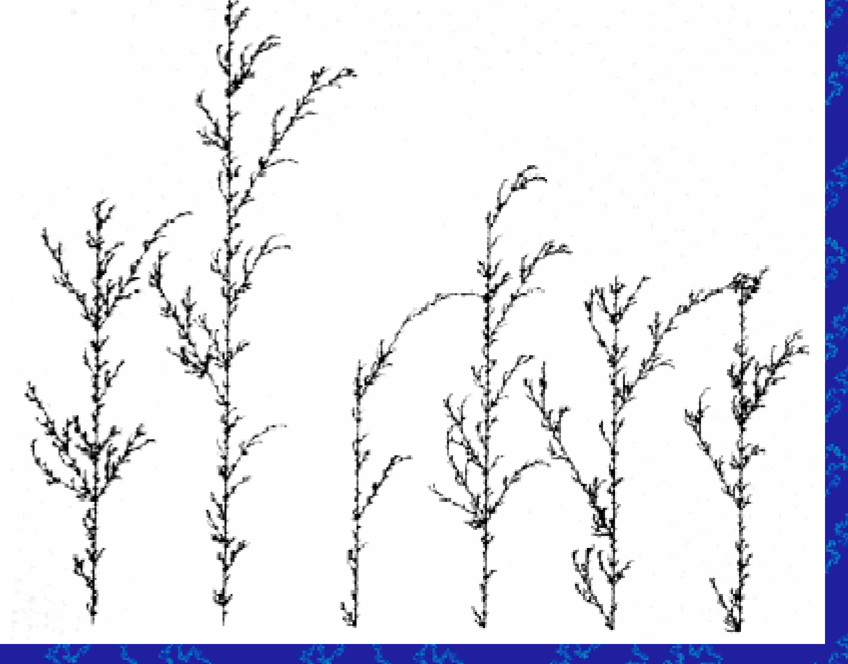
- More rules with the same left side allowed

  A->B: prob

  - the rule used with probability prob, the sum of all probabilities for the rules with the same left side =1

Ex.: random bushes

- w: F
- p: F-> F[+F]F[-F]F : 0.33
- p: F-> F[+F]F : 0.33
- p: F-> F[-F]F : 0.34



2. Context L-system

- For symbol rewriting, its context is cosidered

  lc<A>rc->B

  lc, rc - strings – left and right contexts

  A replaced by B only in the given context

Ex.: XY<A>CDE->AAA

A is rewritten as

... ZAXYACDEF...=> ...ZAXYAAACDEF ...

3. Parametric L-system

- It works with the so-called modules = characters/letters with added parameters
- A modul: $A(x_1,...,x_m)$, the parameter set must be finite, can be empty
- Formal parameters are replaced by real parameters from R, AL exressions allowed
- K=<G,FP,W,P>, where FP – the set of formal parameters
- Rules: id: lc<pred>rc:cond->succ:prob

  where rule number, lc,rc-context, pred - predecessor,

  succ - successor (the right side of the rule),

  cond – logical expression, values 0/1,

  prob - probability

Ex. Rules of stochast. param. context. L-system

w:A(1)B(3)A(5)
p1: A(x)->A(x+1):0.4
p2: A(x)->B(x-1):0.6
p3: A(x)<B(y)>A(z):y<4->B(x+z)[A(y)]

1.Derivation of this L-system is, e.g. :

A(1)B(3)A(5)=>A(2)B(6)[A(3)]B(4)

Interpretation: a turtle reads the modules and values of their parameters, interprets geometrically; parameter – line thickness, step, rotation angle,....

Open L-system

- Parametric context stochastic L-system enlarged by communication moduls
  $?E(x_1,\ldots,x_m)$
- Communication modul – information transfer
- Before rewiting to get values of the parameters, a message is sent to the neighbourhood, parameters are set, then normal rewriting as for parametric L-systems

Ex. Rules limit the growth in the y direction to the distance =2

w:F(0,0)A?E(0)

p1: A >?E(y): y<2->F(x,y+1)A

p2: A >?E(y): y≥2->ε

Module F(x,y) – draw a line segment from the last point to (x,y)

ε: Erase A from the module sequence

Derivation: F(0,0)A?E(0)=>F(0,0)F(0,1)A?E(1) =>
F(0,0)F(0,1)F(0,2)A?E(2) =>
F(0,0)F(0,1)F(0,2)?E(3)

Plants simulation

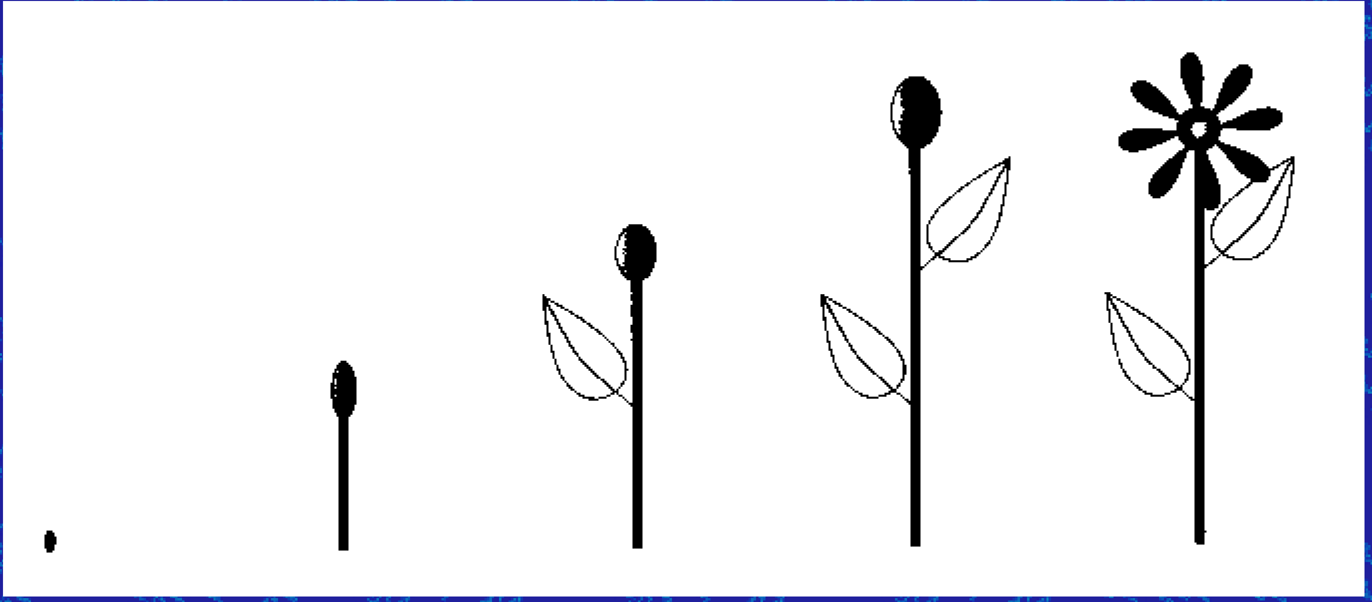
- L-system – so far, best worked out formal theory of synthetic plants
- It usually generates in the 1st phase only a plant skeleton, in the 2nd phase more exact and detailed representation, e.g., using Bezier surfaces or NURBs.

- Ex.: a model of the growing plant, using a simple set of rules; to a complete model definition, geometrical description, dependence on the parameters, …, should be added
  - w:A(0)
  - p1: A(x) : x=0->FA(x+1)
  - p2: F<A(x) : x=1->[-L]FA(x+1)
  - p3: F<A(x) : x=2->[+L]FA(x+1)
  - p4: A(x) : x=3->B
- Turtle commands:
  - F-const. step in **H direction**
  - A(x) – top bud, x – its age
  - B - bloom
  - L - leaf
  - p1 wakes up the top, p2 and p3 cause its growth and then leaves to the left and right are generated

Derivation of this system:

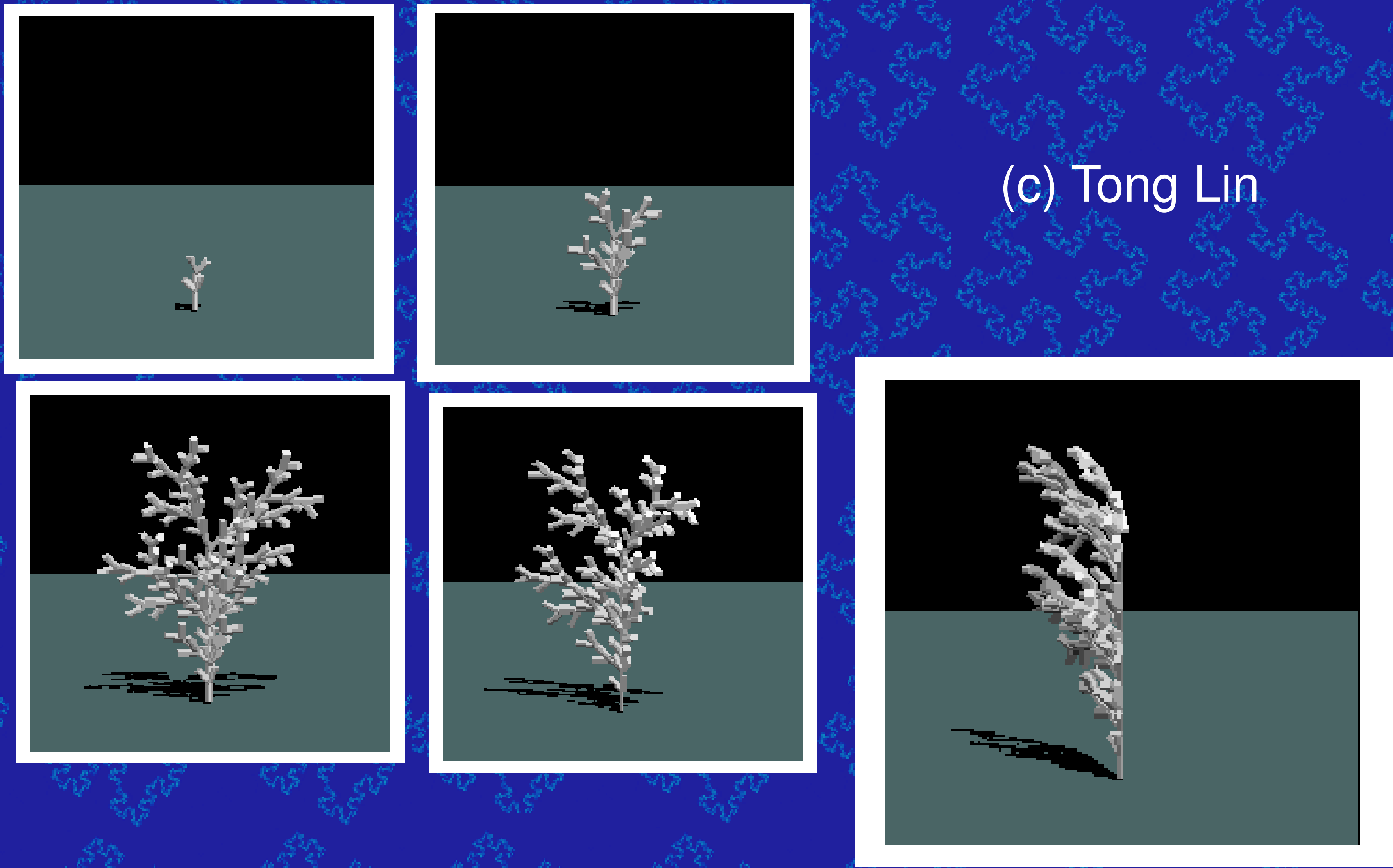
A(0)=> FA(1)=>F[-L]FA(2)=>F[-L]F[+L]FA(3)
=>F[-L]F[+L]FB

Resulting plant:

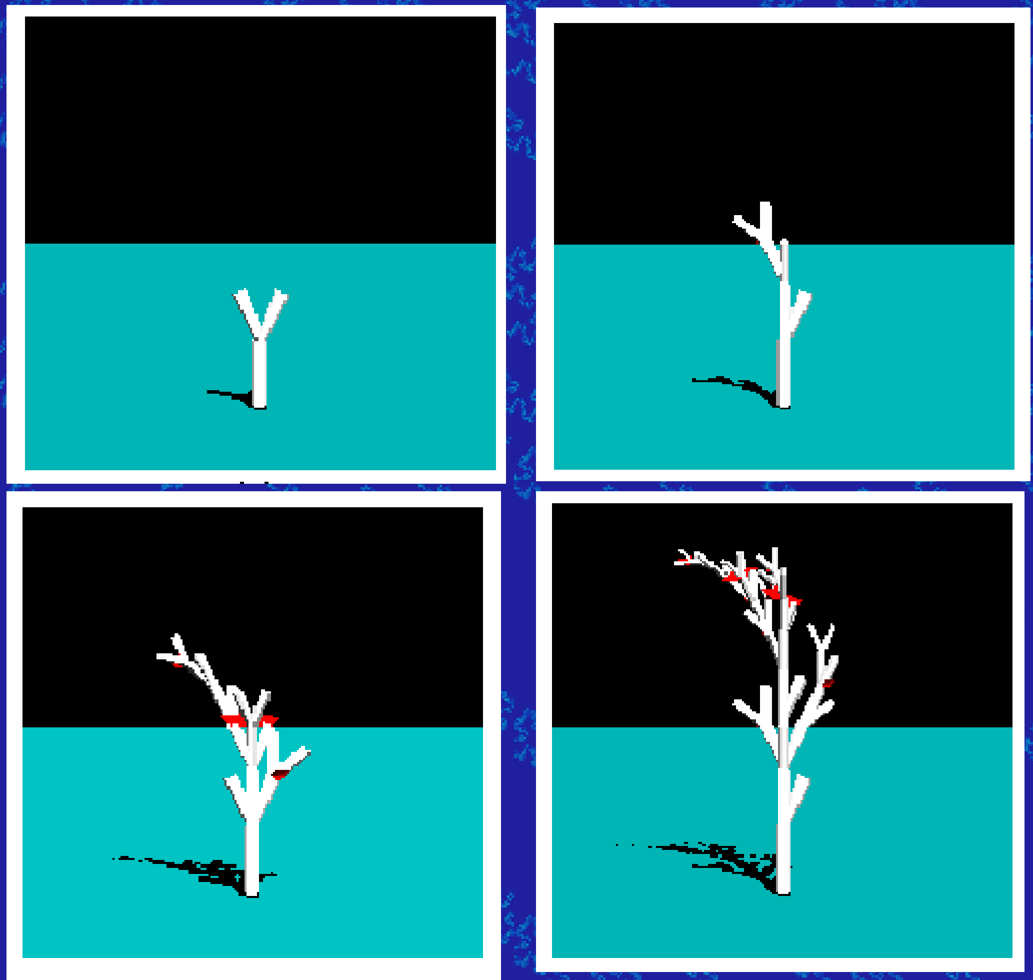


Simple L-system

(c) Tong Lin



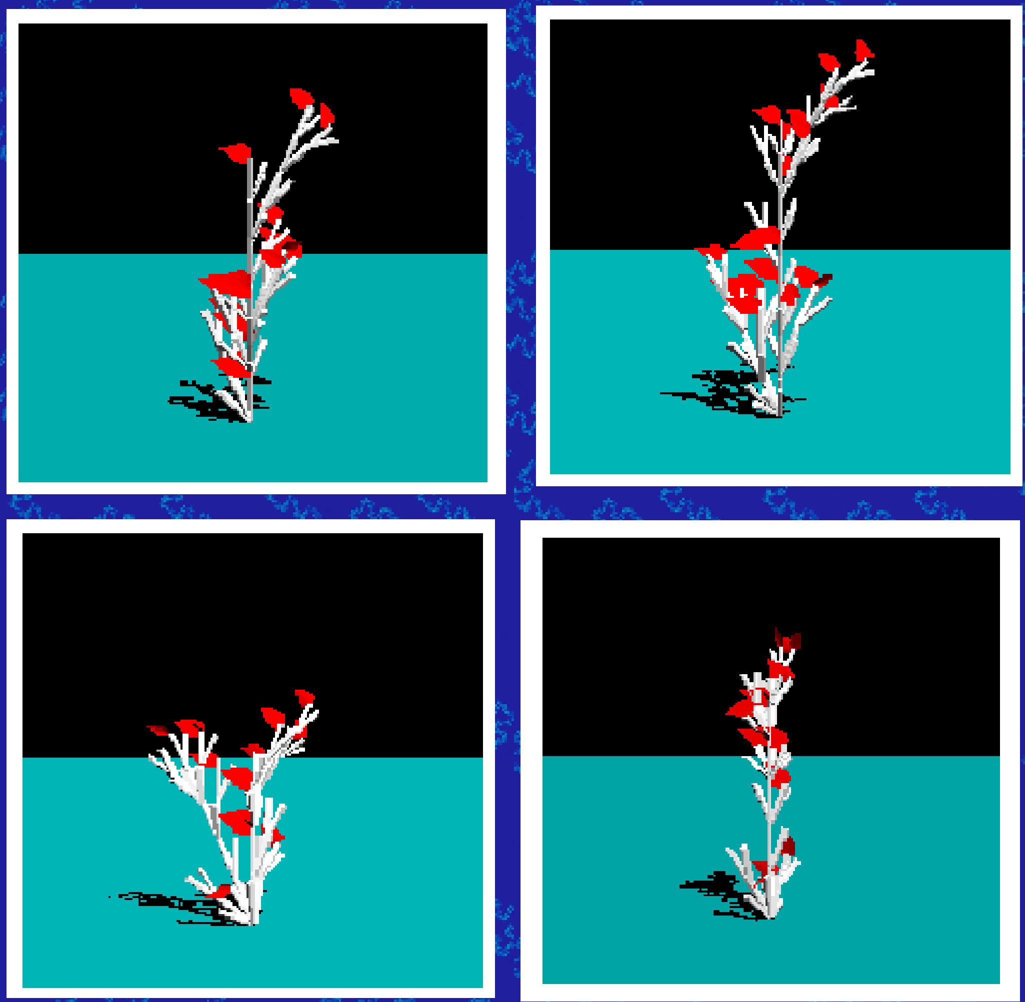
Context L-system



(c) Tong Lin

Simulation of the influence of nutrients stream by interactions of neighbouring plant parts

Stochastic L-system



(c) Tong Lin

4 different plants from the same L-system

Not willing to program it yourself?

Try Fractint: https://www.fractint.org/