

Digital growth

Nature-inspired procedural generation

Complex patterns emerge from simple rules

Modern understanding of nature (for this presentation) rely on four works:

The Chemical Basis of Morphogenesis, Alan Turing (1952)

Mathematical models for cellular interaction in development, Astrid Lindenmayer (1968)

The Fractal Geometry of Nature, Benoît B. Mandelbrot (1982)

The Algorithmic Beauty of Plants, Przemysław Prusinkiewicz and Aristid Lindenmayer (2004)

Goals of this presentation

Goals of this presentation

Presented algorithms fits in two categories:

Goals of this presentation

Presented algorithms fits in two categories:

- Physics models (Reaction-Diffusion and Diffusion-Limited Aggregation)

Goals of this presentation

Presented algorithms fits in two categories:

- Physics models (Reaction-Diffusion and Diffusion-Limited Aggregation)
- Biological growth models (L-System and Space Colonization)

Goals of this presentation

Presented algorithms fits in two categories:

- Physics models (Reaction-Diffusion and Diffusion-Limited Aggregation)
- Biological growth models (L-System and Space Colonization)

What I'll show:

Goals of this presentation

Presented algorithms fits in two categories:

- Physics models (Reaction-Diffusion and Diffusion-Limited Aggregation)
- Biological growth models (L-System and Space Colonization)

What I'll show:

- Brief outlines of each method

Goals of this presentation

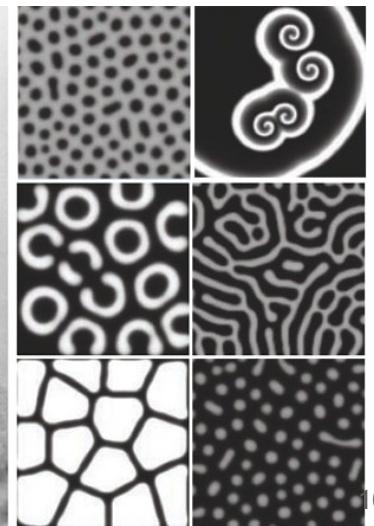
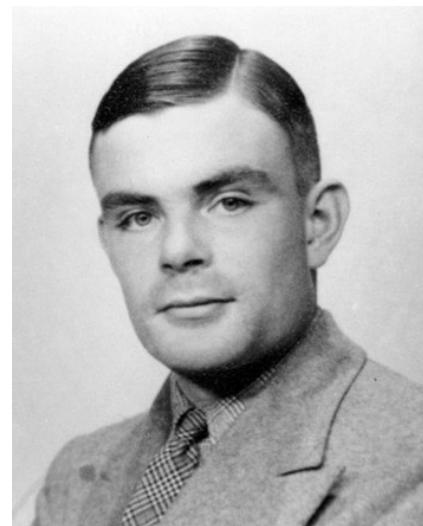
Presented algorithms fits in two categories:

- Physics models (Reaction-Diffusion and Diffusion-Limited Aggregation)
- Biological growth models (L-System and Space Colonization)

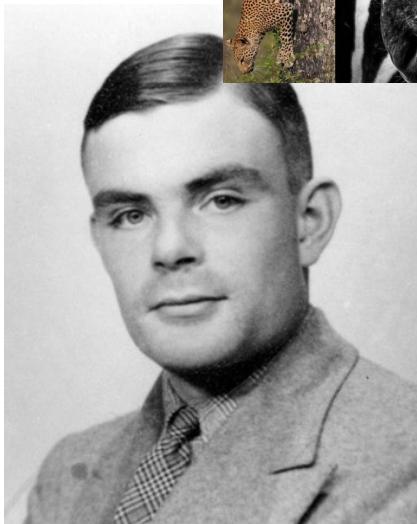
What I'll show:

- Brief outlines of each method
- Implementation ease

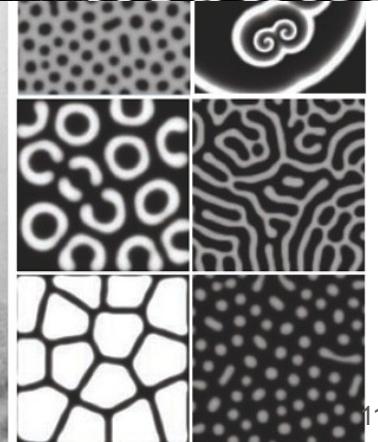
Reaction-Diffusion for Turing Patterns



Reaction-Diffusion for Turing Patterns

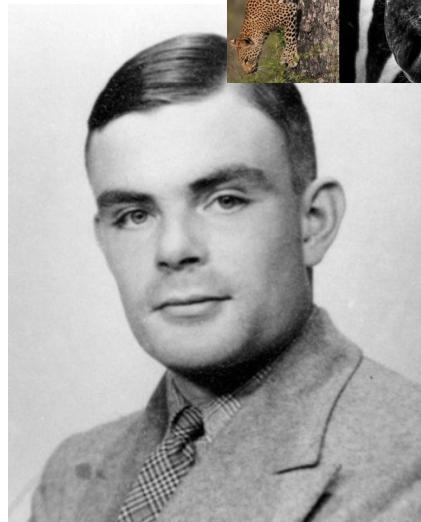


© Polymaths lab

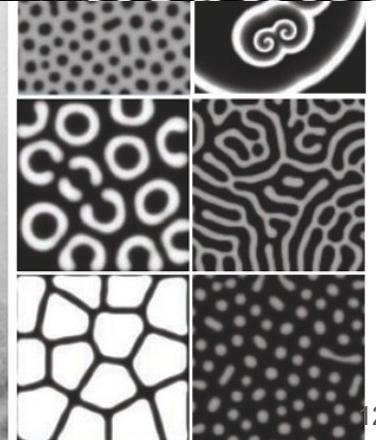


Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment



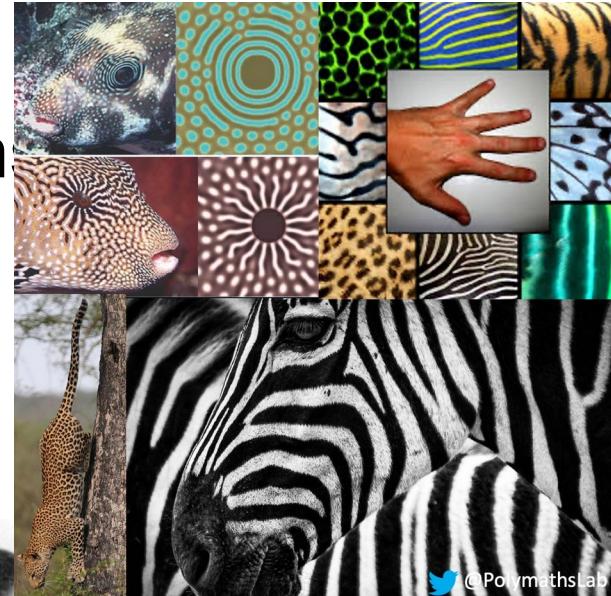
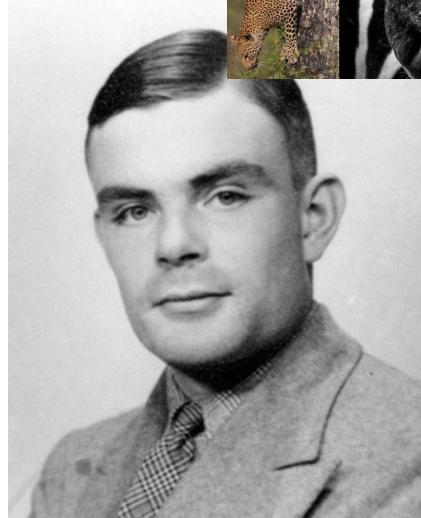
 C Polymaths lab



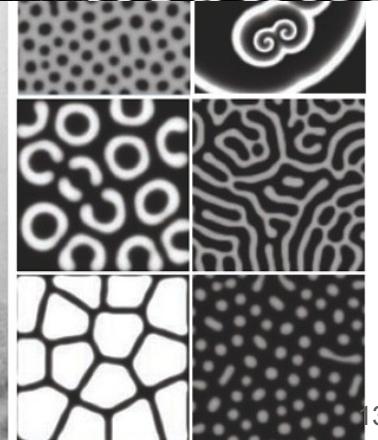
Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse



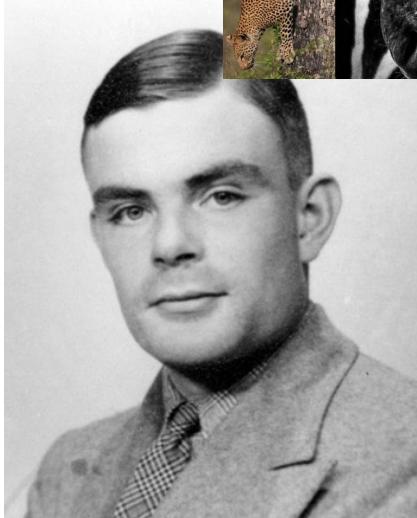
Twitter logo and text "Polymaths lab"



Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other

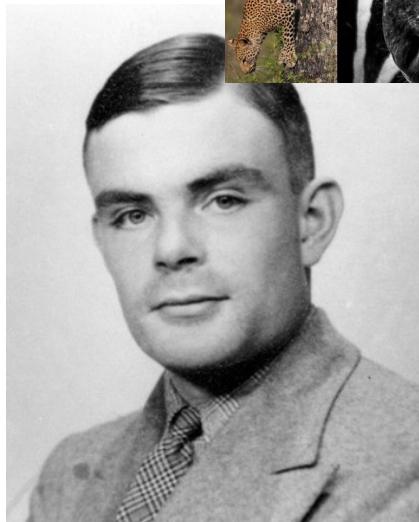


© Polymaths lab

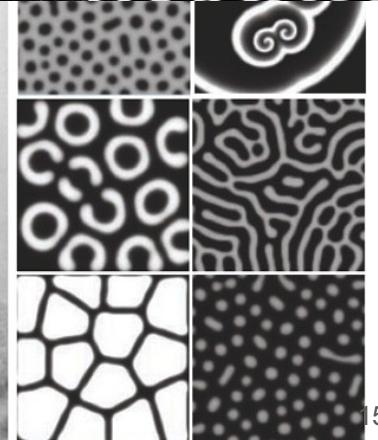
Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other
3. That's all



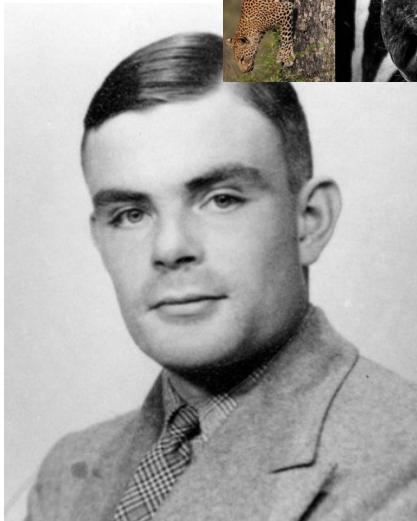
© Polymaths lab



Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other
3. That's all



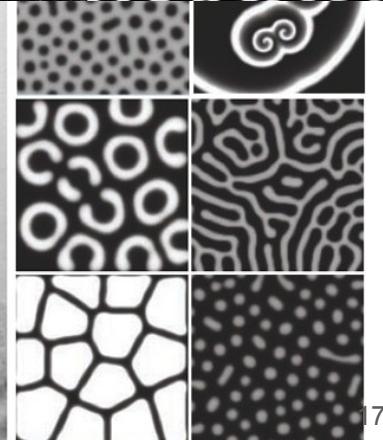
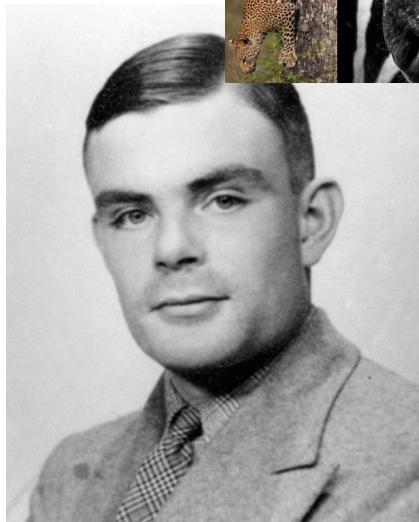
© Polymaths lab

Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other
3. That's all

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u, v),$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v)$$



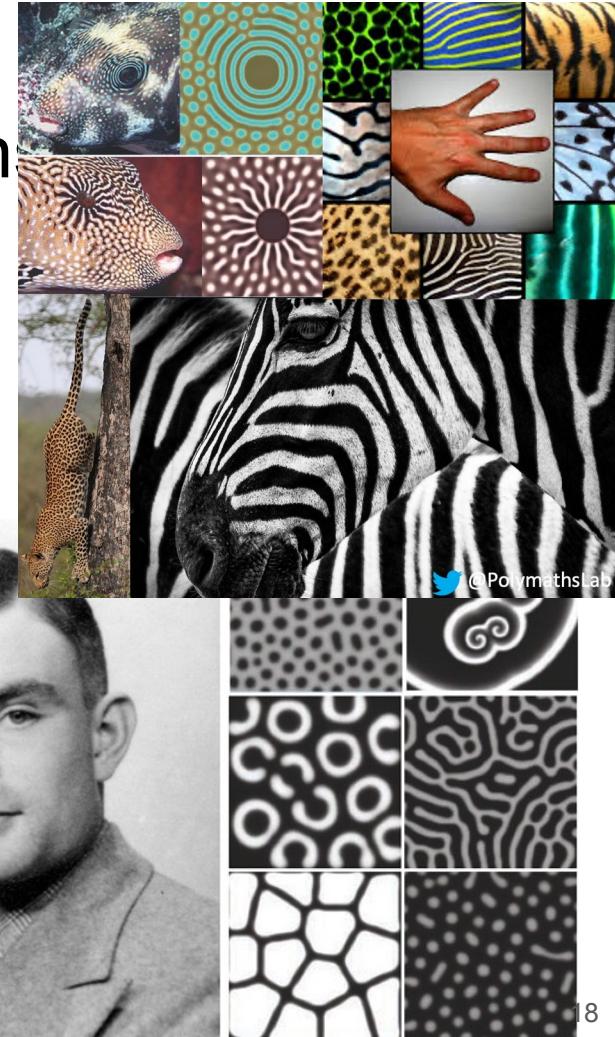
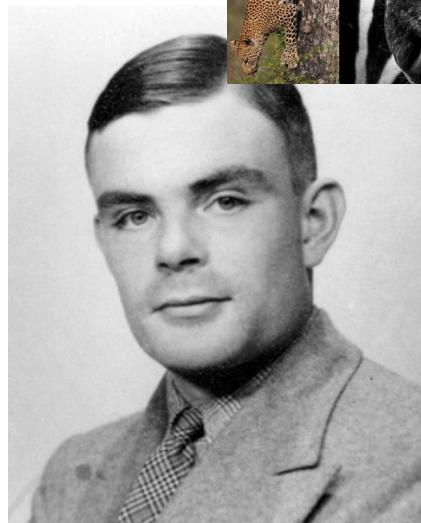
Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other
3. That's all

Diffusion

$$\frac{\partial u}{\partial t} = \boxed{D_u \nabla^2 u} + f(u, v),$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v)$$

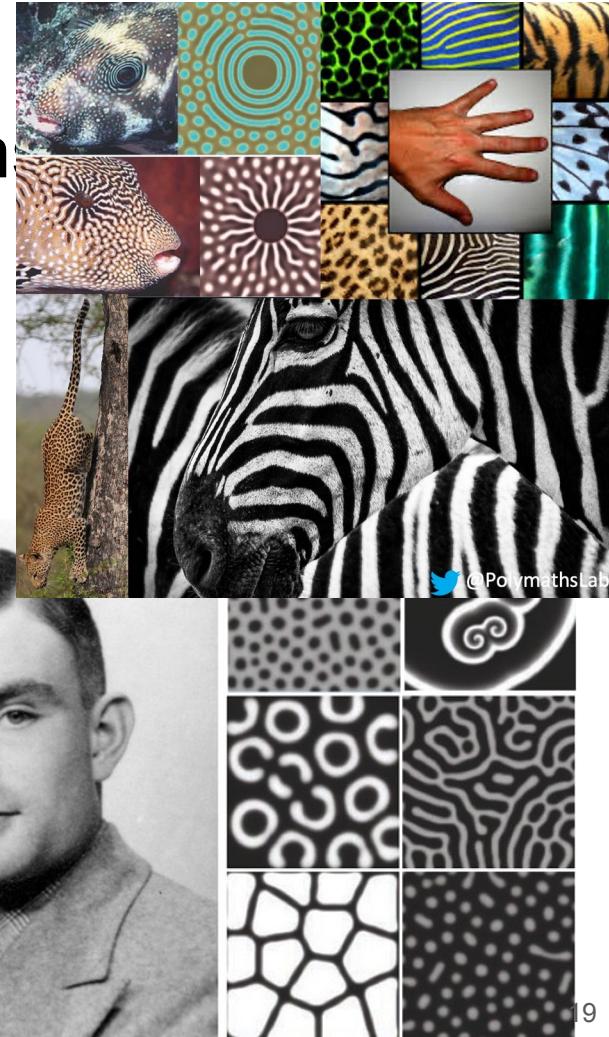
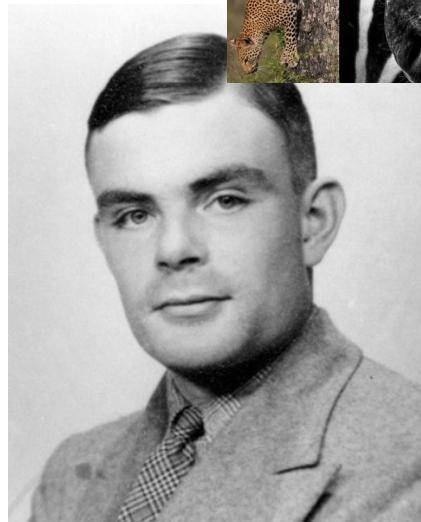


Reaction-Diffusion for Turing Patterns

2 components (u and v) are evolving in an environment

1. They diffuse
2. They react to each other
3. That's all

$$\begin{array}{c} \text{Diffusion} \quad \text{Reaction} \\ \frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u, v), \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v) \end{array}$$



Reaction-Diffusion for Turing Patterns

Gray-Scott model:

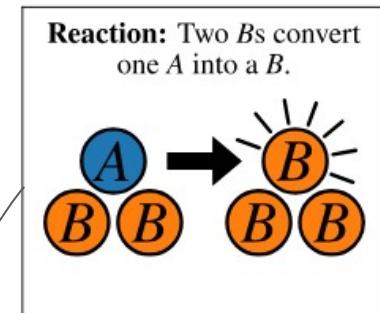
$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F \cdot (1 - u),$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (K + F)v$$

Reaction-Diffusion for Turing Patterns

Gray-Scott model:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F \cdot (1 - u),$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (K + F)v$$

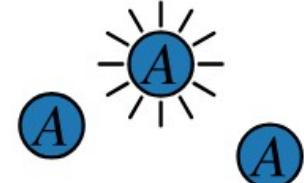


Reaction-Diffusion for Turing Patterns

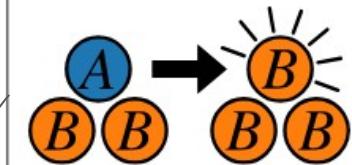
Gray-Scott model:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + [F \cdot (1 - u)],$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (K + F)v$$

Chemical A is added at a given feed rate.



Reaction: Two Bs convert one A into a B.

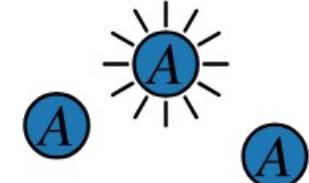


Reaction-Diffusion for Turing Patterns

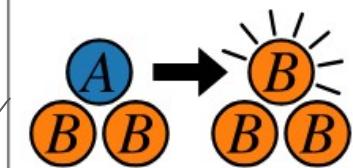
Gray-Scott model:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F \cdot (1 - u),$$
$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (K + F)v$$

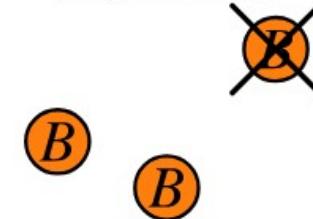
Chemical A is added at a given feed rate.



Reaction: Two Bs convert one A into a B.

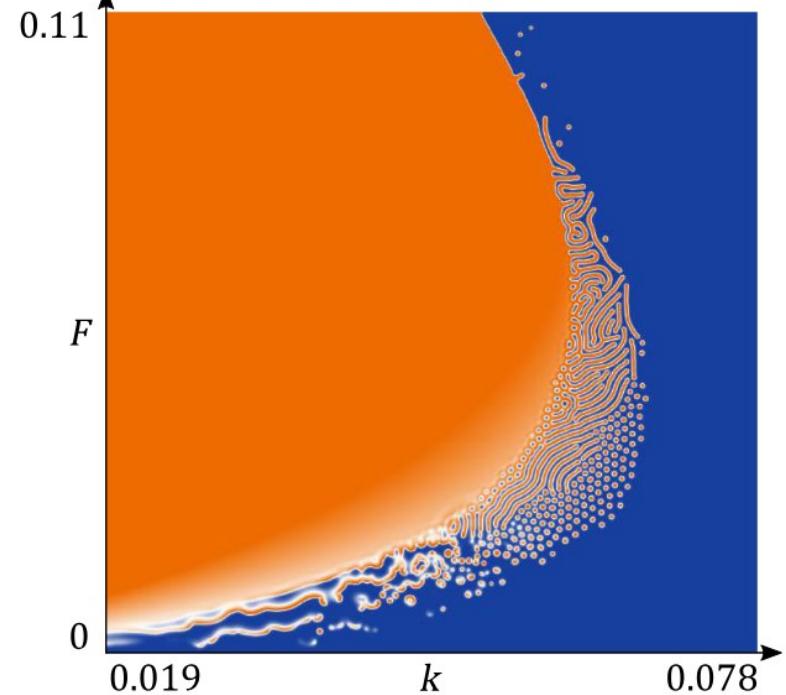


Chemical B is removed at a given kill rate.



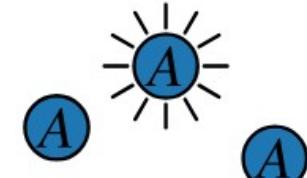
Reaction-Diffusion for Turing Patterns

Grav-Scott model:

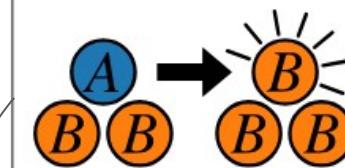


$$\begin{aligned} &+ [F \cdot (1 - u)], \\ &- (K + F)v \end{aligned}$$

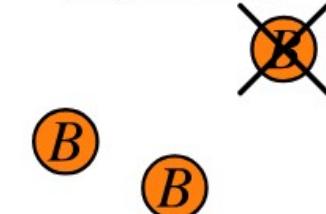
Chemical A is added at a given feed rate.



Reaction: Two Bs convert one A into a B.

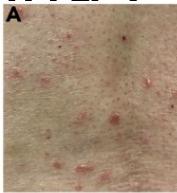
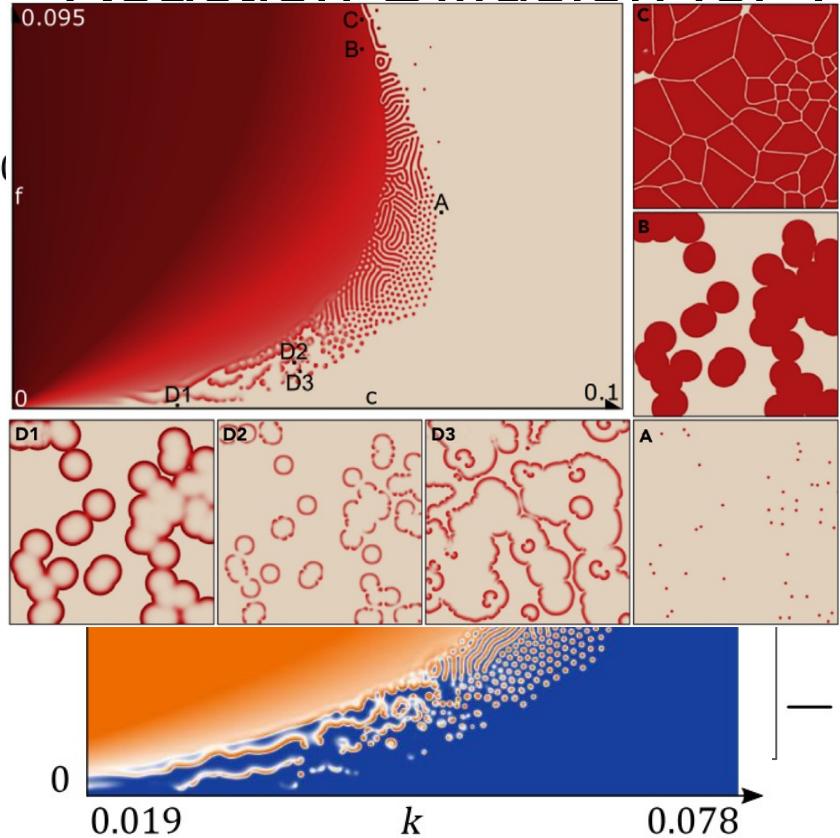


Chemical B is removed at a given kill rate.

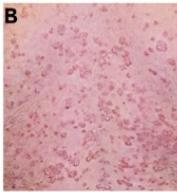


Reaction-Diffusion for Turing Patterns

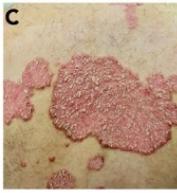
Reaction-Diffusion for Turing Patterns



A. Papular. Small, millimeter-size, scattered, erythematous papules. This pattern also represents the earliest lesion of psoriasis and may evolve into patterns B-D



B. Small plaque/nummular. Round or oval plaques with smooth borders and the diameter less than 30 mm, distributed all over the body. The surface of the lesion does not show appreciable patterning, such as fissuring or faceting. Adjacent, growing nummular lesions merge into polygonal structures.



C. Large plaque. Scaly, erythematous plaques larger than the nummular plaques of Pattern B. The plaques have irregular, polycyclic contour. Fissuring and faceting of the surface is often present.

F

$$-(K + F)v$$



D.Circinate.
D1. Annular: Ring-shaped plaques with central clearing. Individual rings do not intersect but merge into a larger annular structure.



D2. Rosettes. Rings with discontinuous boundary.



D3. Reniform. kidney form, oval with a notch (arrow).

Chemical A is added at a given feed rate.



Reaction-Diffusion for Turing Patterns

Short demo: <https://editor.p5js.org/marchartley/sketches/q-kafkilA>

Reaction-Diffusion for Turing Patterns

Short demo: <https://editor.p5js.org/marchartley/sketches/q-kafkilA>



Reaction-Diffusion for Turing Patterns

Short demo: <https://editor.p5js.org/marchartley/sketches/q-kafkilA>



Reaction-Diffusion for Turing Patterns

Short demo: <https://editor.p5js.org/marchartley/sketches/q-kafkilA>



Diffusion-Limited Aggregation

Diffusion-Limited Aggregation



Diffusion-Limited Aggregation



Diffusion-Limited Aggregation



Diffusion-Limited Aggregation

1. Particles move in a brownian motion



Diffusion-Limited Aggregation

1. Particles move in a brownian motion
2. A particle sticks to a surface imperfection and free



Diffusion-Limited Aggregation

1. Particles move in a brownian motion
2. A particle sticks to a surface imperfection and freeze
3. Any other particle touching it also freeze



Diffusion-Limited Aggregation

1. Particles move in a brownian motion
2. A particle sticks to a surface imperfection and freeze
3. Any other particle touching it also freeze
4. And so on



Diffusion-Limited Aggregation

Short demo: <https://editor.p5js.org/marchartley/sketches/ZwPy8xGOp>

Diffusion-Limited Aggregation

Short demo: <https://editor.p5js.org/marchartley/sketches/ZwPy8xGOp>



Diffusion-Limited Aggregation

Short demo: <https://editor.p5js.org/marchartley/sketches/ZwPy8xGOp>





gregation

<https://org/marchartley/sketches/ZwPy8xGOp>



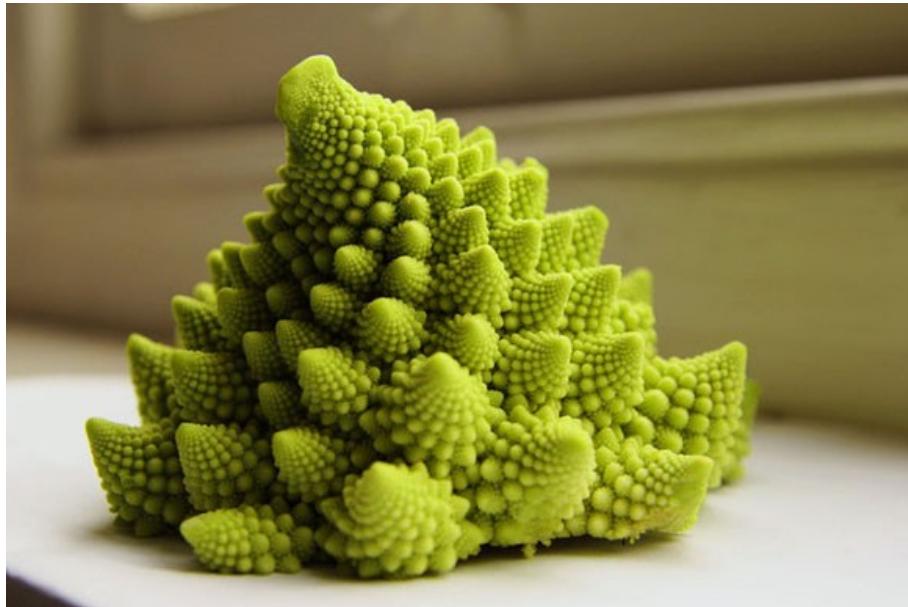


gregation

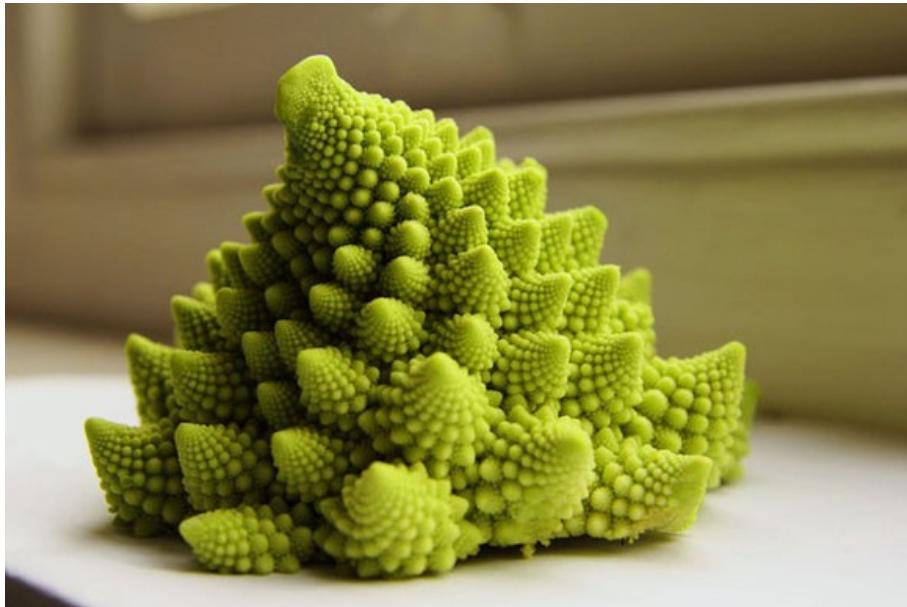
<http://org/marchartley/>



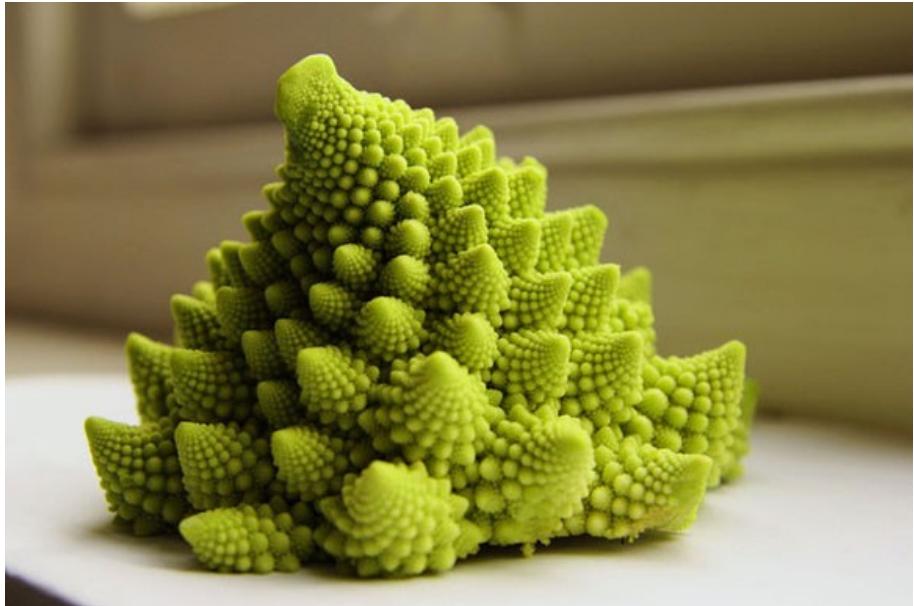
Nature is fractal



Nature is fractal



Nature is fractal



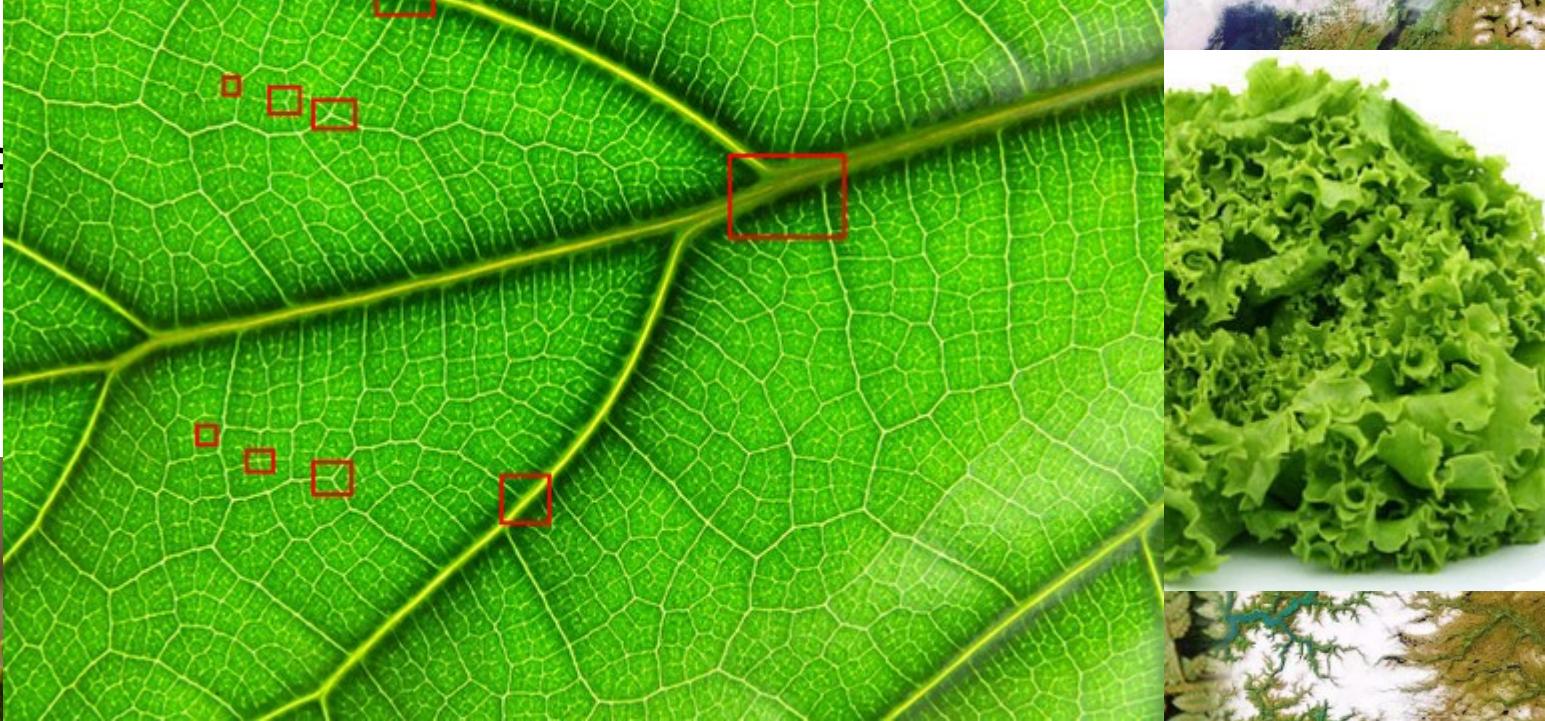
Nature is fractal



Nature is fractal



Nature



L-System, the language of nature

Rewriting system for modeling plant growth with recursivity

Composed of:

- Alphabet (letters to use),
- Axiom (starting point),
- Production rules (how to rewrite each letter)

Usually, the final string is used to command a Turtle generator



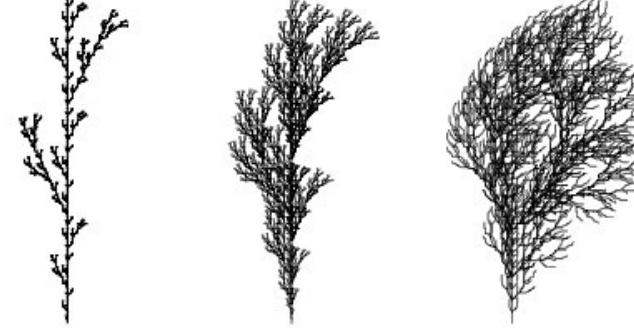
L-System, the language of nature

Rewriting system for modeling plant growth via L-Systems

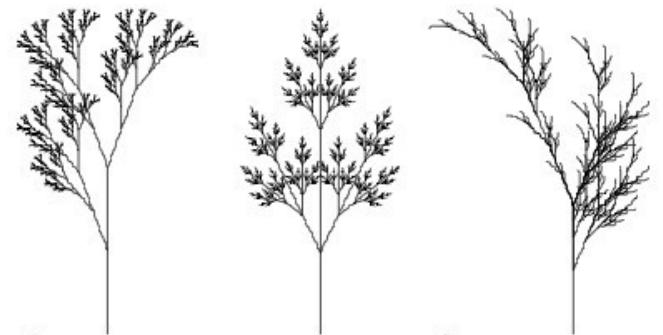
Composed of:

- Alphabet (letters to use),
- Axiom (starting point),
- Production rules (how to rewrite each letter)

Usually, the final string is used to command a computer to draw the structure.



a	b	c
$n=5, \delta=25.7^\circ$ F $F \rightarrow F [+F] F [-F] F$	$n=5, \delta=20^\circ$ F $F \rightarrow F [+F] F [-F] [F]$	$n=4, \delta=22.5^\circ$ F $F \rightarrow FF - [-F+F+F] + [+F-F-F]$



d	e	f
$n=7, \delta=20^\circ$ X $X \rightarrow F [+X] F [-X] + X$ $F \rightarrow FF$	$n=7, \delta=25.7^\circ$ X $X \rightarrow F [+X] [-X] FX$ $F \rightarrow FF$	$n=5, \delta=22.5^\circ$ X $X \rightarrow F - [[X] + X] + F [+FX] - X$ $F \rightarrow FF$

Figure 1.24: Examples of plant-like structures generated by bracketed OL-systems. L-systems (a), (b) and (c) are edge-rewriting, while (d), (e) and (f) are node-rewriting.

L-System, the language of nature

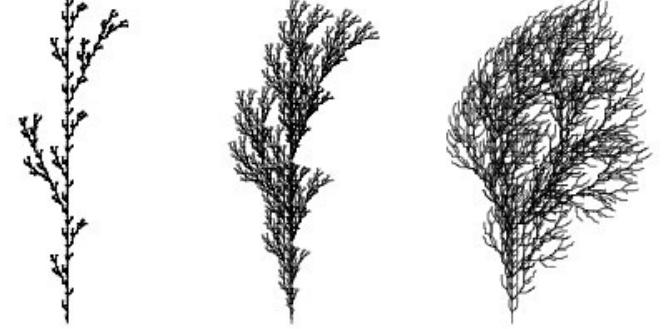
Rewriting system for modeling plant growth via L-Systems

Composed of:

- Alphabet (letters to use),
- Axiom (starting point),



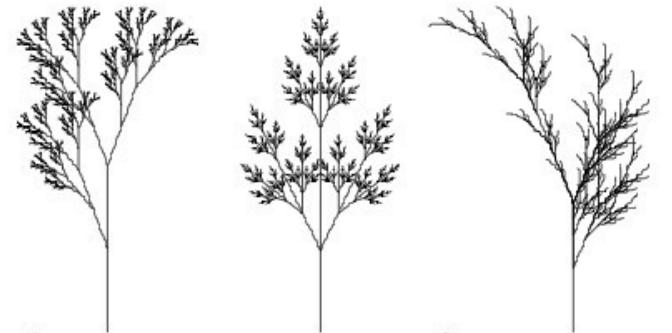
which let
and :



a
 $n=5, \delta=25.7^\circ$
F
 $F \rightarrow F [+F] F [-F] F$

b
 $n=5, \delta=20^\circ$
F
 $F \rightarrow F [+F] F [-F] [F]$

c
 $n=4, \delta=22.5^\circ$
F
 $F \rightarrow FF - [-F+F+F]+[+F-F-F]$



d
 $n=7, \delta=20^\circ$
X
 $X \rightarrow F [+X] F [-X] +X$
 $F \rightarrow FF$

e
 $n=7, \delta=25.7^\circ$
X
 $X \rightarrow F [+X] [-X] FX$
 $F \rightarrow FF$

f
 $n=5, \delta=22.5^\circ$
X
 $X \rightarrow F - [[X] + X] + F [+FX] - X$
 $F \rightarrow FF$

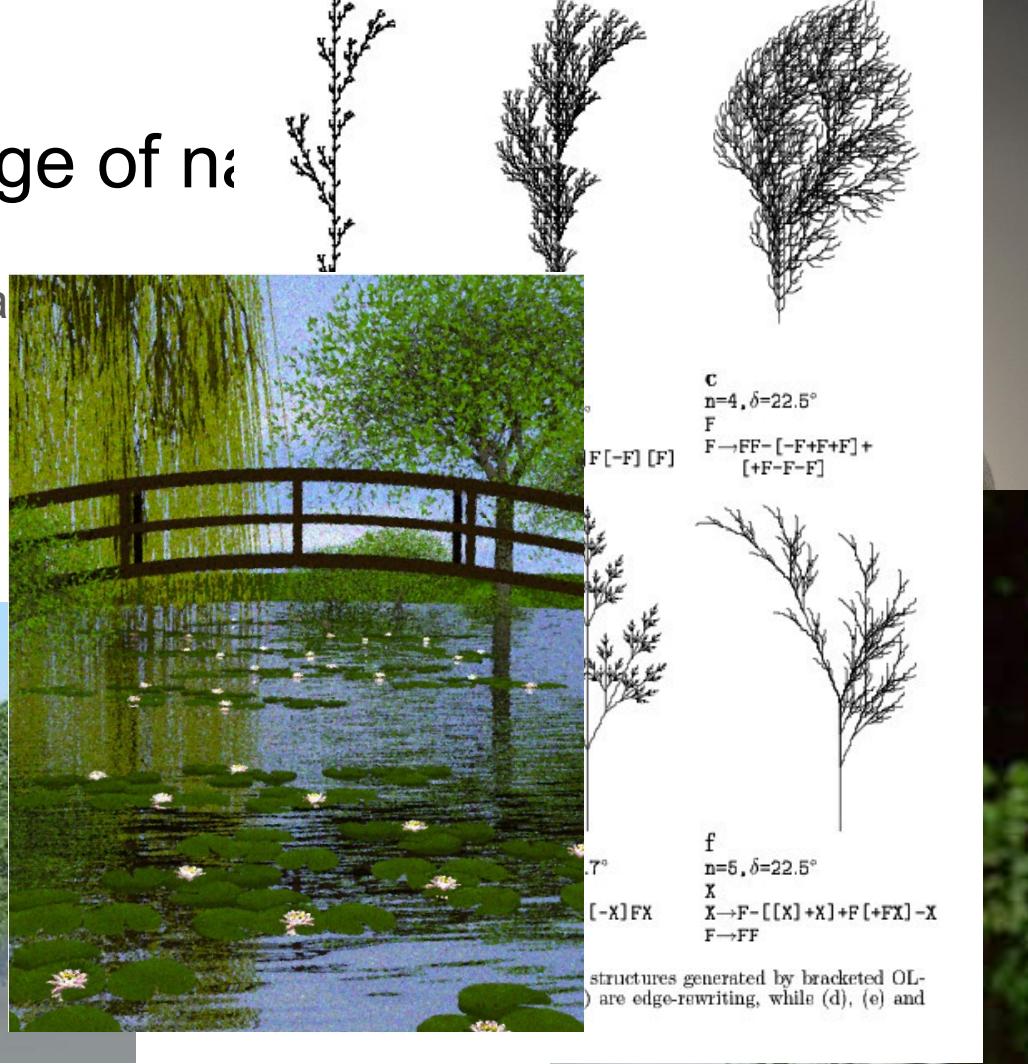
Figure 1.24: Examples of plant-like structures generated by bracketed OL-systems. L-systems (a), (b) and (c) are edge-rewriting, while (d), (e) and (f) are node-rewriting.

L-System, the language of nature

Rewriting system for modeling plants

Composed of:

- Alphabet (letters to use),
- Axiom (starting point),

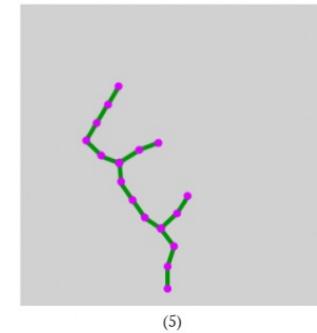
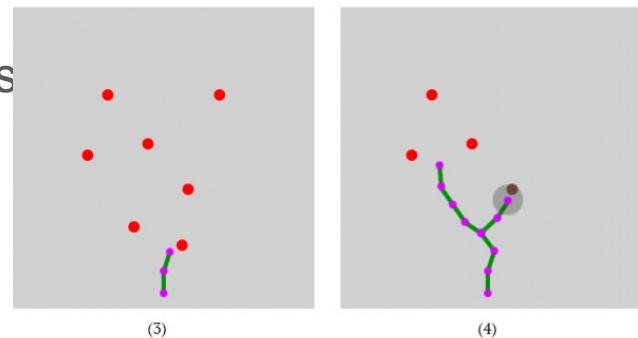
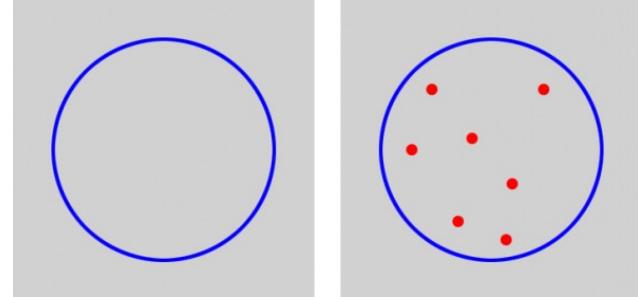


L-System, the language of nature

Short demo: <https://editor.p5js.org/marchartley/sketches/UkGvoRwob>

Space Colonization

1. Define a space that will contain the tree
2. Randomly add leaves in it
3. Grow branches (or trunk) in direction of the leaves
(with a radius of influence)
4. Remove very close leaves
5. Voilà



Space Colonization

Short demo: https://editor.p5js.org/marchartley/sketches/_g3xOJuUy

Space Colonization

Short demo: https://editor.p5js.org/marchartley/sketches/_g3xOJuUy



Space Colonization

Short demo: https://editor.p5js.org/marchartley/sketches/_g3xOJuUy



Space Colonization

Short demo: https://editor.p5js.org/marchartley/sketches/_g3xOJuUy

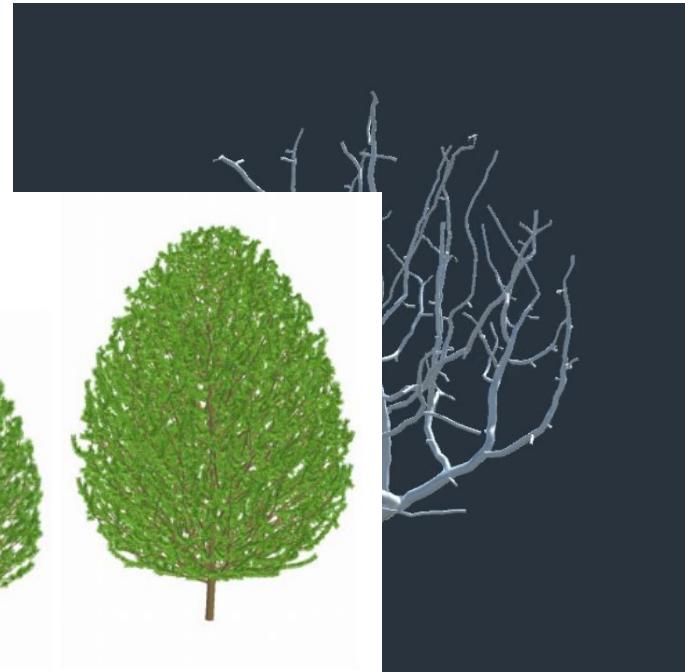
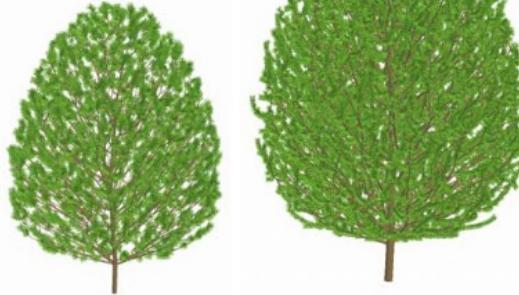


Figure 3.25: Four trees representing nine trees of different ages.

Conclusion

Conclusion

Conclusion

- 4 algorithms presented: Reaction-Diffusion, Diffusion-Limited Aggregation, L-System and Space Colonization

Conclusion

- 4 algorithms presented: Reaction-Diffusion, Diffusion-Limited Aggregation, L-System and Space Colonization
- Procedure comes from a physical process or biological process

Conclusion

- 4 algorithms presented: Reaction-Diffusion, Diffusion-Limited Aggregation, L-System and Space Colonization
- Procedure comes from a physical process or biological process
- Very easy to implement

Conclusion

- 4 algorithms presented: Reaction-Diffusion, Diffusion-Limited Aggregation, L-System and Space Colonization
- Procedure comes from a physical process or biological process
- Very easy to implement
- +/- Easy control on the generation process