Artificial Life
Summer 2025

Examples for Pattern Formation in Biological Systems

Master Computer Science [MA-INF 4201]
Mon 14c.t. – 15:45, HSZ, HS-2

Dr. Nils Goerke, Autonomous Intelligent Systems,
Department of Computer Science, University of Bonn

Pattern Formation

- Cellular Automata (CA)
- Langton's Ant
- Langton's Loop
- *Self Organizing Criticality (SOC)*
- Lindenmayer systems
- Iterated functions
- Differential equations
- Population dynamics

Pattern Formation

- Iterated functions
- Linear and exponential growth
- Fibonacci sequence
- Logistic growth
- Predator-prey system
- Lotka-Volterra equations
- Activator-inhibitor equations
- Reaction-diffusion systems
- Plant morphogenesis, phylotaxis
- Golden section, Golden angle
- Self similarity

Iterated Functions

Iterated functions are closely related to sequences.

Sequence of values:

$$\{\ x(1),\quad x(2),\quad x(3),\quad x(4),\quad x(5),\quad \dots\ \}$$

Iterated Functions

Iterated functions are closely related to sequences.

Sequence of values:

$$\{\ x(1),\quad x(2),\quad x(3),\quad x(4),\quad x(5),\quad \ldots\ \}$$

Sequences can be generated using recursive defined or iterated functions:

$$x(i+1) = f(\ x(i)\ )$$

Iterated Functions

Iterated functions are closely related to sequences.
Sequence of values:

$$\{\ x(1),\quad x(2),\quad x(3),\quad x(4),\quad x(5),\quad \ldots\ \}$$

Sequences can be generated using recursive defined or iterated functions:

$$x(i+1) = f\,(\ x(i)\ )$$

Iterated Functions

Iterated functions are closely related to sequences.
Sequence of values:

$$\{\ x(1),\quad x(2),\quad x(3),\quad x(4),\quad x(5),\quad \dots\ \}$$

Sequences can be generated using recursive defined or iterated functions:

$$x(i+1) = f\,(\ x(i)\ )$$

In iterated functions the next value $x(i+1)$ is a function of the preceding value $x(i)$.

Iterated functions need an initial value $x(0)$.

Iterated Functions, examples

Examples for iterated functions:

$\boldsymbol{x(i+1) = x(i) + 2}$ with $\boldsymbol{x(0)}=3$

$\{ 3, 5, 7, 9, 11, \ldots. \}$

$\boldsymbol{x(i+1) = a\, x(i)}$ with $\boldsymbol{x(0)}=100.0$, $\boldsymbol{a}=0.9$

$\{ 100.00, 90.00, 81.00, 72.90, 65.61, 59.049, \ldots \}$

$\boldsymbol{x(i+1) = (1-\varepsilon)\, x(i)}$ with $\boldsymbol{x(0)}>0$, and $0 < \boldsymbol{\varepsilon} << 1$

$\boldsymbol{x(i+1) = (1+\varepsilon)\, x(i)}$ *with* $\boldsymbol{x(0)}>0$, *and* $0 < \boldsymbol{\varepsilon} << 1$

Iterated Functions, examples 2

The definition of iterated functions can contain logic functions as well:

Iterated Functions, examples 2

The definition of iterated functions can contain logic functions as well:

$$x(i+1) = \begin{cases} \frac{1}{2}\, x(i) & \text{if } x(i) \text{ is even} \\ 3\, x(i) + 1 & \text{if } x(i) \text{ is odd} \end{cases}$$

Iterated Functions, examples 2

The definition of iterated functions can contain logic functions as well:

$$x(i+1) = \begin{cases} \frac{1}{2}\, x(i) & \text{if } x(i) \text{ is even} \\ 3\, x(i) + 1 & \text{if } x(i) \text{ is odd} \end{cases}$$

Can you predict the final value of *x(i)* when the starting value *x(0)* is given ?

Iterated Functions, examples 2

The definition of iterated functions can contain logic functions as well:

$$x(i+1) = \begin{cases} \tfrac{1}{2}\, x(i) & \text{if } x(i) \text{ is even} \\ 3\, x(i) + 1 & \textit{if } x(i) \textit{ is odd} \end{cases}$$

Can you predict the final value of *x(i)* when the starting value *x(0)* is given ?

Play this iterated function as a two person game:
Start with x(0) and iterate the function by turns:
You lose, if ***<u>you</u>*** *end up with x(i)=1.*

See: Collatz Conjecture; https://en.wikipedia.org/wiki/Collatz_conjecture

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called “**Look-and-Say Sequence**”:

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called "**Look-and-Say Sequence**":

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) : 1

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called “**Look-and-Say Sequence**”:

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) : 1 : “one 1”

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called “**Look-and-Say Sequence**”:

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) : 1 : “one 1” resulting in “11”

L&S (1) : 11

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called "**Look-and-Say Sequence**":

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) :	1	: "one 1" resulting in "11"
L&S (1) :	11	: "two 1s" resulting in "21"
L&S (2) :	21	:

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called “**Look-and-Say Sequence**”:

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) :	1	: “one 1” resulting in “11”
L&S (1) :	11	: “two 1s” resulting in “21”
L&S (2) :	21	: “one 2” , “one 1” resulting in “1211”
L&S (3) :	1211	:

Iterated functions, examples 3

Another really nice, and interesting iterated sequence is defined via a mixture of math and language.

It is called "**Look-and-Say Sequence**":

Look at the element(i) of the sequence and **say** what you see; this yields the next element (i+1).
In detail: speak out the amount of identical ciphers of a kind, followed by the numerical value of the cipher.

L&S (0) :	1	: "one 1" resulting in "11"
L&S (1) :	11	: "two 1s" resulting in "21"
L&S (2) :	21	: "one 2" , "one 1" resulting in "1211"
L&S (3) :	1211	: "one 1" , "one 2" , "two 1s"
L&S (4) :	111221	: "three 1s" , "two 2s" , "one 1"
L&S (5) :	312211	: "one 3" , "one 1" , "two 2s" , "two 1s"
L&S (6) :	13112221	: "one 1" , "one 3", …. …. ….

Iterated functions, examples 3

The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.

The sequence, starting with “d=3” is named, **Conway sequence**.

The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).

The starting number “d”, will stay at the end end of the sequence
“ d , 1d , 111d , 311d , 13211d , 111312211d , … “

The sequence will grow indefinitely, exept for “22, 22, 22, …. ”.

Starting with numbers “0,1,2,3” no larger number than 3 will appear.

Denoting L_i as the length of the sequence, the limit of the ratio of two subsequent lengths exists and is given by:

$$\lim_{i \to \infty} \frac{L_{i+1}}{L_i} = \lambda \approx 1.30357726903\ldots$$ with average growth of 30%.

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.
- The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.
- The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).
- The starting number “d”, will stay at the end end of the sequence
  “ d , 1d , 111d , 311d , 13211d , 111312211d , … “

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.
- The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).
- The starting number “d”, will stay at the end end of the sequence “ d , 1d , 111d , 311d , 13211d , 111312211d , … “
- The sequence will grow indefinitely, exept for “22, 22, 22, …. ”.

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.
- The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).
- The starting number “d”, will stay at the end end of the sequence “ d , 1d , 111d , 311d , 13211d , 111312211d , … “
- The sequence will grow indefinitely, exept for “22, 22, 22, …. ”.
- Starting with numbers “0,1,2,3” no larger number than 3 will appear.

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, examples 3

- The look-and-say sequence was introduced and analyzed by the mathematician **John Horton Conway**.
- The sequence, starting with “d=3” is named, **Conway sequence**.
- The idea behind this sequence is identical to the Run-length encoding (**RLE**) lossless data compression algorithm (1967, 1983).
- The starting number “d”, will stay at the end end of the sequence “ d , 1d , 111d , 311d , 13211d , 111312211d , … “
- The sequence will grow indefinitely, exept for “22, 22, 22, …. ”.
- Starting with numbers “0,1,2,3” no larger number than 3 will appear.
- Denoting L_i as the length of the sequence, the limit of the ratio of two subsequent lengths exists and is given by:

$$\lim_{i \to \infty} \frac{L_{i+1}}{L_i} = \lambda \approx 1.30357726903\ldots$$

with average growth of 30%.

Parts taken from: https://en.wikipedia.org/wiki/Look-and-say_sequence

Iterated functions, Typical Behavior

From dynamical system theory, the following classes of behavior including some sub-classes have been identified (so far).

- divergence
- convergence
- periodic behavior
- deterministic chaos

Directly related are the different types of attractors:
- infinity
- fixpoint, stable and unstable
- limit cycles, stable and unstable
- strange attractors, often with fractal dimension

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (1 - x(i))$ $\quad 0.0 < x(i) < 1.0$, and $0.0 < a < 4.0$

Almost independent of the starting value $x(0)$ this function shows all known classes of behavior, adjustable via the parameter **a**.

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (1 - x(i))$ $0.0 < x(i) < 1.0$, and $0.0 < a < 4.0$

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

0.0 < **a** < 1.0 : decay to 0.0

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (\, 1 - x(i)\,)$ $0.0 < x(i) < 1.0$, and $0.0 < a < 4.0$

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

0.0 < **a** < 1.0 : decay to 0.0
1.0 < **a** < 3.0 : convergence to a fixpoint x* (depening on a)

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (1 - x(i))$ 0.0 < x(i) < 1.0, and 0.0 < a < 4.0

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

0.0 < **a** < 1.0 : decay to 0.0
1.0 < **a** < 3.0 : convergence to a fixpoint x* (depening on a)
3.0 < **a** < 3.44949. periodic oscillation between 2 values

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (1 - x(i))$ 0.0 < x(i) < 1.0, and 0.0 < a < 4.0

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

0.0 < **a** < 1.0 :	decay to 0.0
1.0 < **a** < 3.0 :	convergence to a fixpoint x* (depening on a)
3.0 < **a** < 3.44949.	periodic oscillation between 2 values
3.44949 < **a** <	periodic oscillation between 4 values
.	periodic oscillation between 8 values
.	periodic oscillation between 16 values

Iterated functions, Logistic Map

An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (\, 1 - x(i)\,)$ $0.0 < x(i) < 1.0$, and $0.0 < a < 4.0$

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

0.0 < **a** < 1.0 :	decay to 0.0
1.0 < **a** < 3.0 :	convergence to a fixpoint x* (depening on a)
3.0 < **a** < 3.44949.	periodic oscillation between 2 values
3.44949 < **a** <	periodic oscillation between 4 values
.	periodic oscillation between 8 values
.	periodic oscillation between 16 values

3.56995 < **a** < 4.0	onset to deterministic chaos
a = 4.0	deterministic chaos
4.0 < **a**	divergence

Iterated functions, Logistic Map

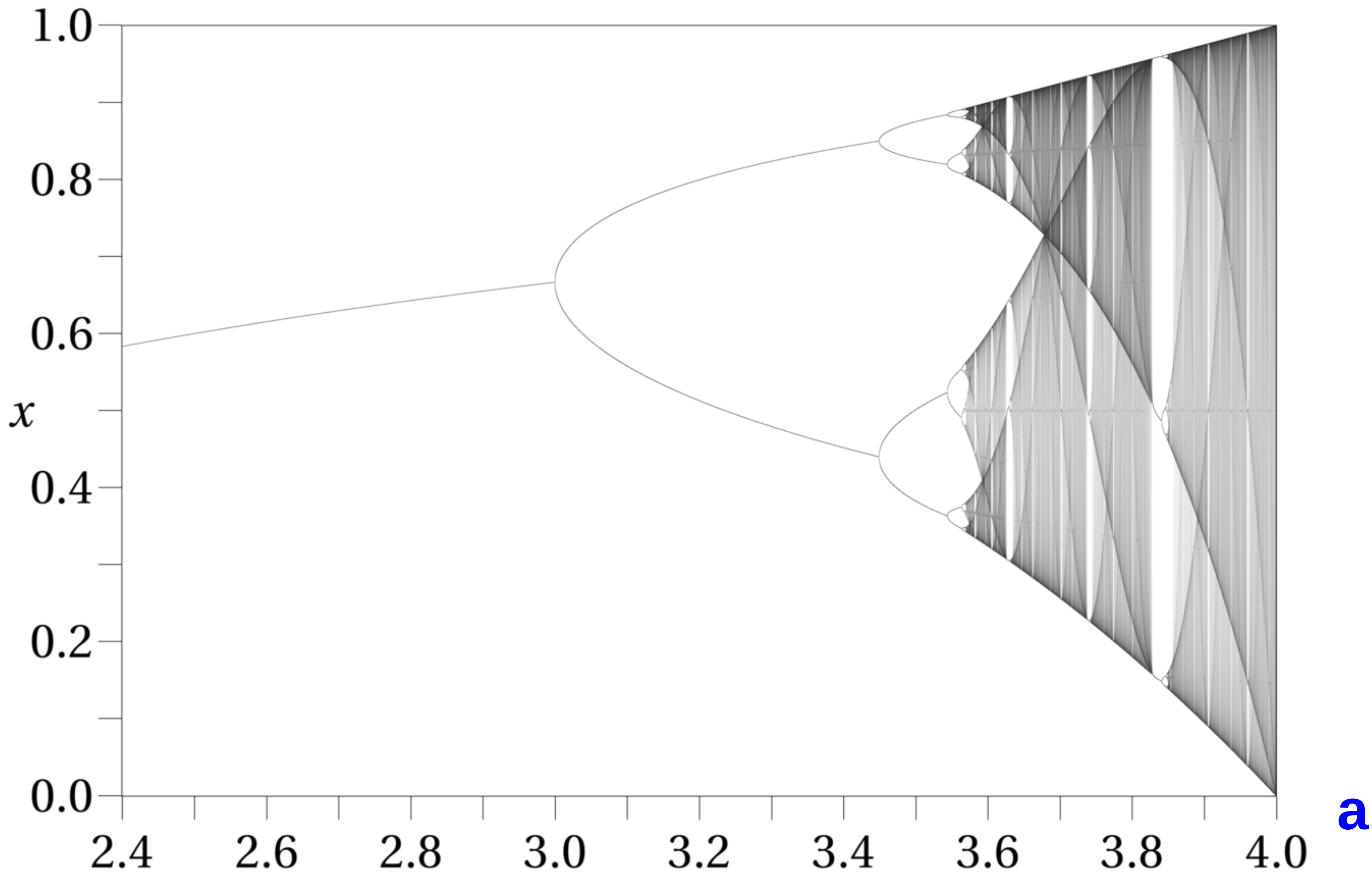
An extremely fascinating iterated function is the logistic map:

$x(i+1) = a\, x(i)\, (1 - x(i))$ $\quad 0.0 < x(i) < 1.0$, and $0.0 < a < 4.0$

Almost independent of the starting value x(0) this function shows all known classes of behavior, adjustable via the parameter **a**.

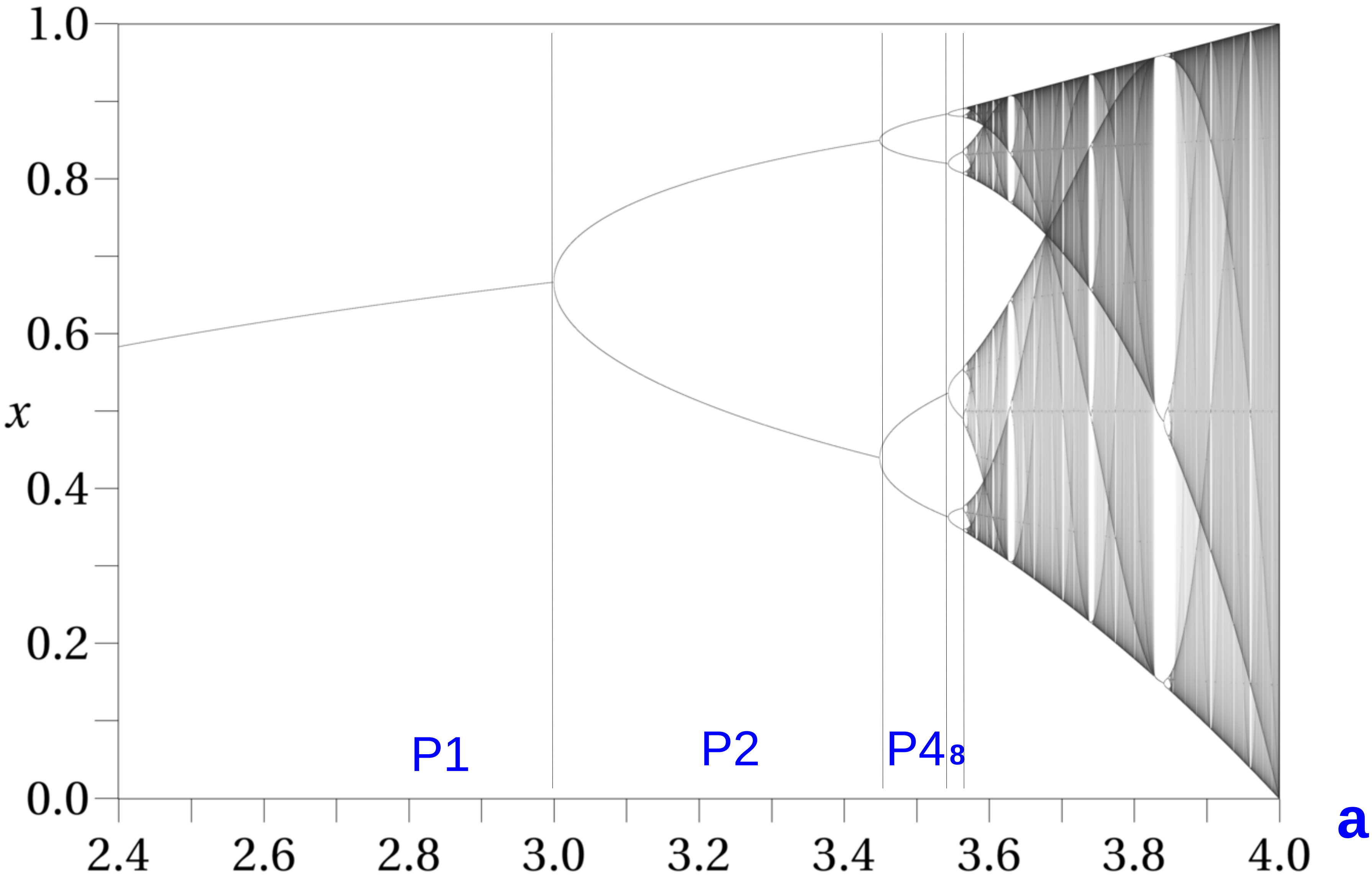
0.0 < **a** < 1.0 :	decay to 0.0
1.0 < **a** < 3.0 :	convergence to a fixpoint x* (depending on a)
3.0 < **a** < 3.44949.	periodic oscillation between 2 values
3.44949 < **a** < ...	periodic oscillation between 4 values
.	periodic oscillation between 8 values
.	periodic oscillation between 16 values
3.56995 < **a** < 4.0	onset to deterministic chaos
a = 4.0	deterministic chaos
4.0 < **a**	divergence

Iterated functions, Logistic Map



Picture from: https://en.wikipedia.org/wiki/Logistic_map

Iterated functions, Logistic Map



Picture adapted from from: https://en.wikipedia.org/wiki/Logistic_map

Iterated functions, Typical Behavior

Iterated functions can show a wide variety of different behaviors.

Depending on kind and structure of the iterated functions, some of these behaviors can be described and explicitly explained:
e.g.:
- linear growth
- exponential decay
- exponential growth
- logistic growth
- periodic behavior
- deterministic chaos (in part)

But be aware, that some of the possible behaviors are still too complex to be easily described by todays vocabulary.

Linear Growth

Implementing linear growth of a population of individuals is easy with iterated functions:

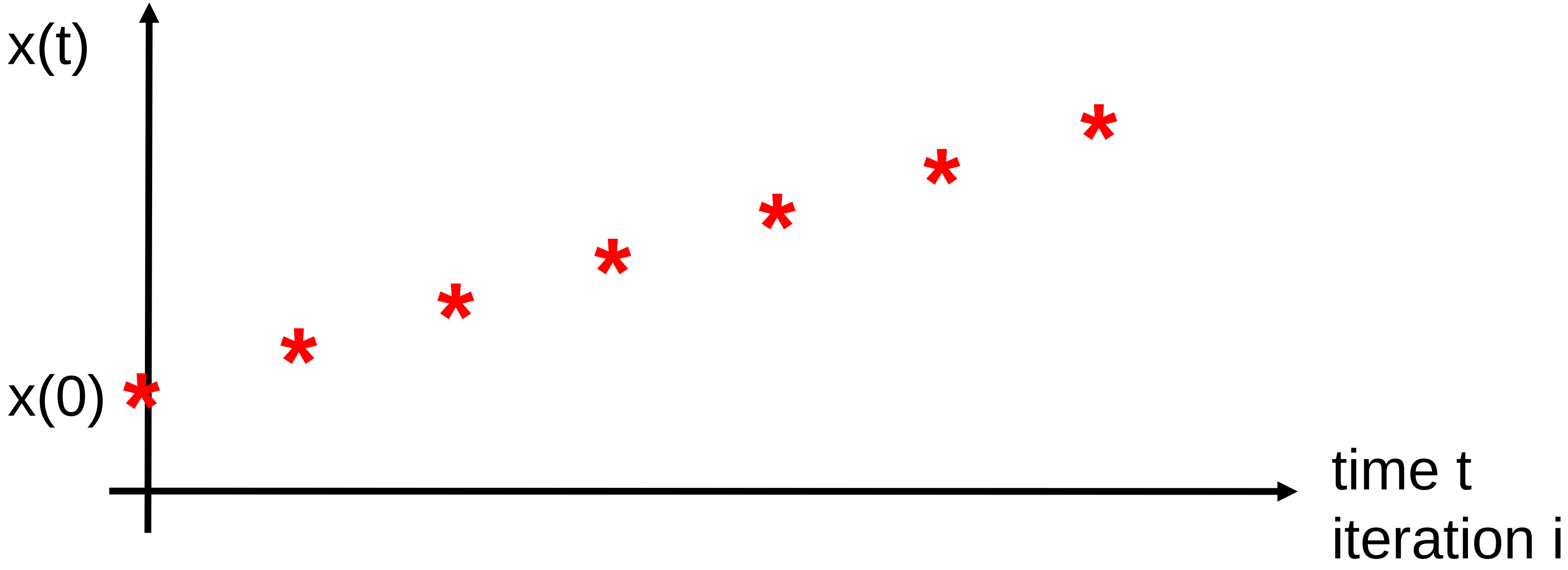
$$x(i+1) = x(i) + c \qquad \text{with } x(0)>0, \text{ and } c>0$$

Linear Growth

Linear Growth

Implementing linear growth of a population of individuals is easy with iterated functions:

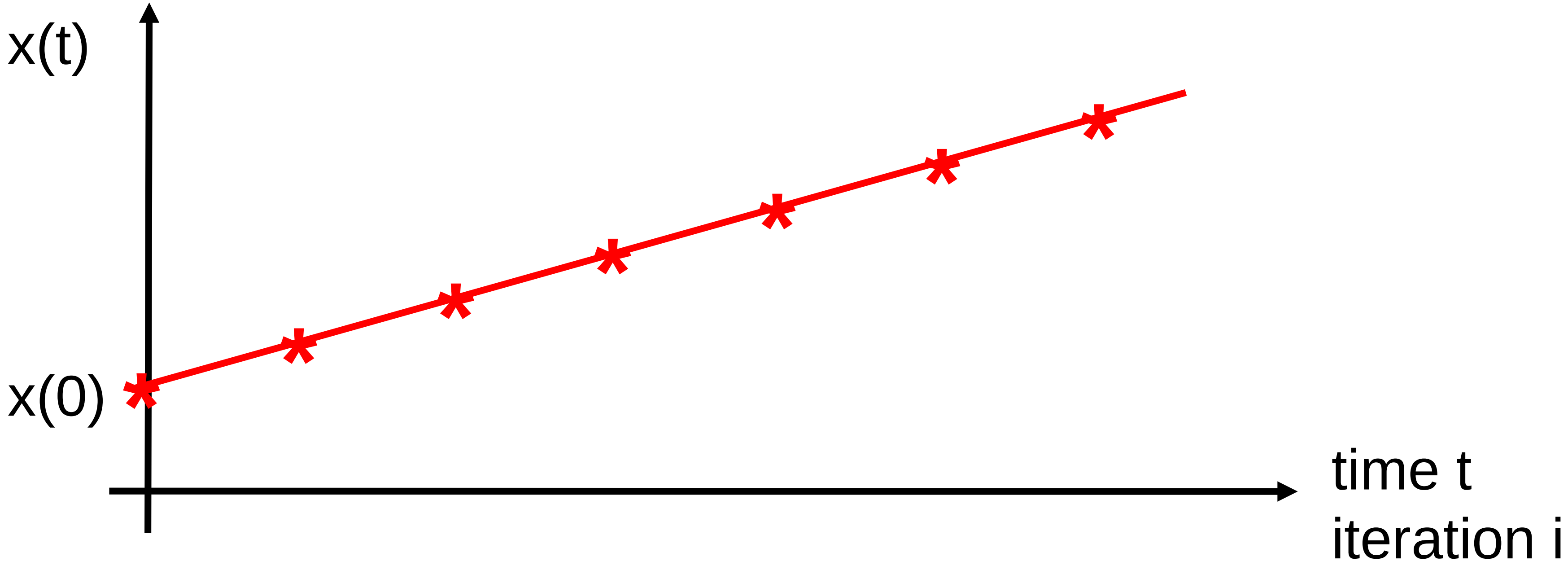
$$x(i+1) = x(i) + c \qquad \text{with } x(0)>0, \text{ and } c>0$$



Linear Growth

Implementing linear growth of a population of individuals is easy with iterated functions:

$$x(i+1) = x(i) + c \qquad \text{with } x(0)>0, \text{ and } c>0$$



Exponential Growth

To obtain exponential growth, the increase is proportional to the number of individuals (*birthrate*).

$$
\begin{aligned}
x(i+1) &= x(i) + r\,x(i) && \text{with } x(0)>0 \\
&= (1+r)\,x(i) && \text{with } b=1+r=1+\varepsilon \\
&= b\,x(i)
\end{aligned}
$$

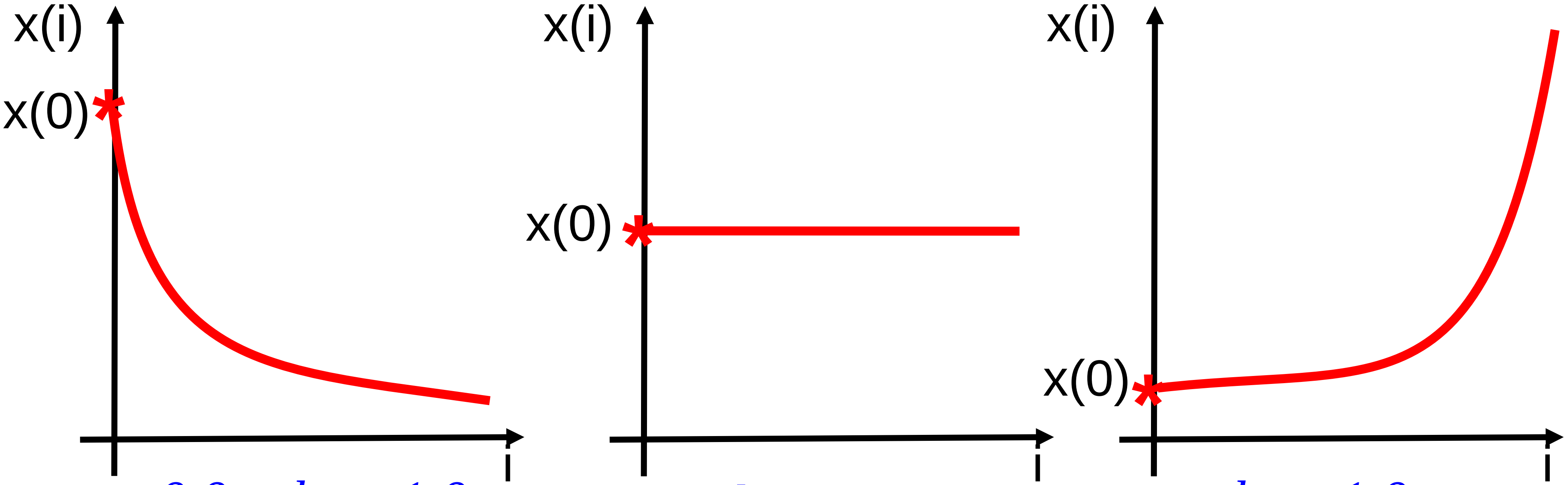
Depending on *b* a different behavior results:

$0.0 < b < 1.0$	exponential decay
$b = 1.0$	constant, stable population
$b > 1.0$	exponential growth

Exponential Growth

To obtain exponential growth, the increase is proportional to the number of individuals (*birthrate*).

$$x(i+1) = b\, x(i)$$



$0.0 < b < 1.0$ $b = 1.0$ $b > 1.0$

Fibonacci Sequence

In 1202 AD, Leonardo of Pisa (also known as Fibonacci), has mentioned a recursive sequence that has been inspired by a growing population
(number of pairs of rabbits, unbounded reproduction).

$$x(i+1) = x(i) + x(i-1)$$

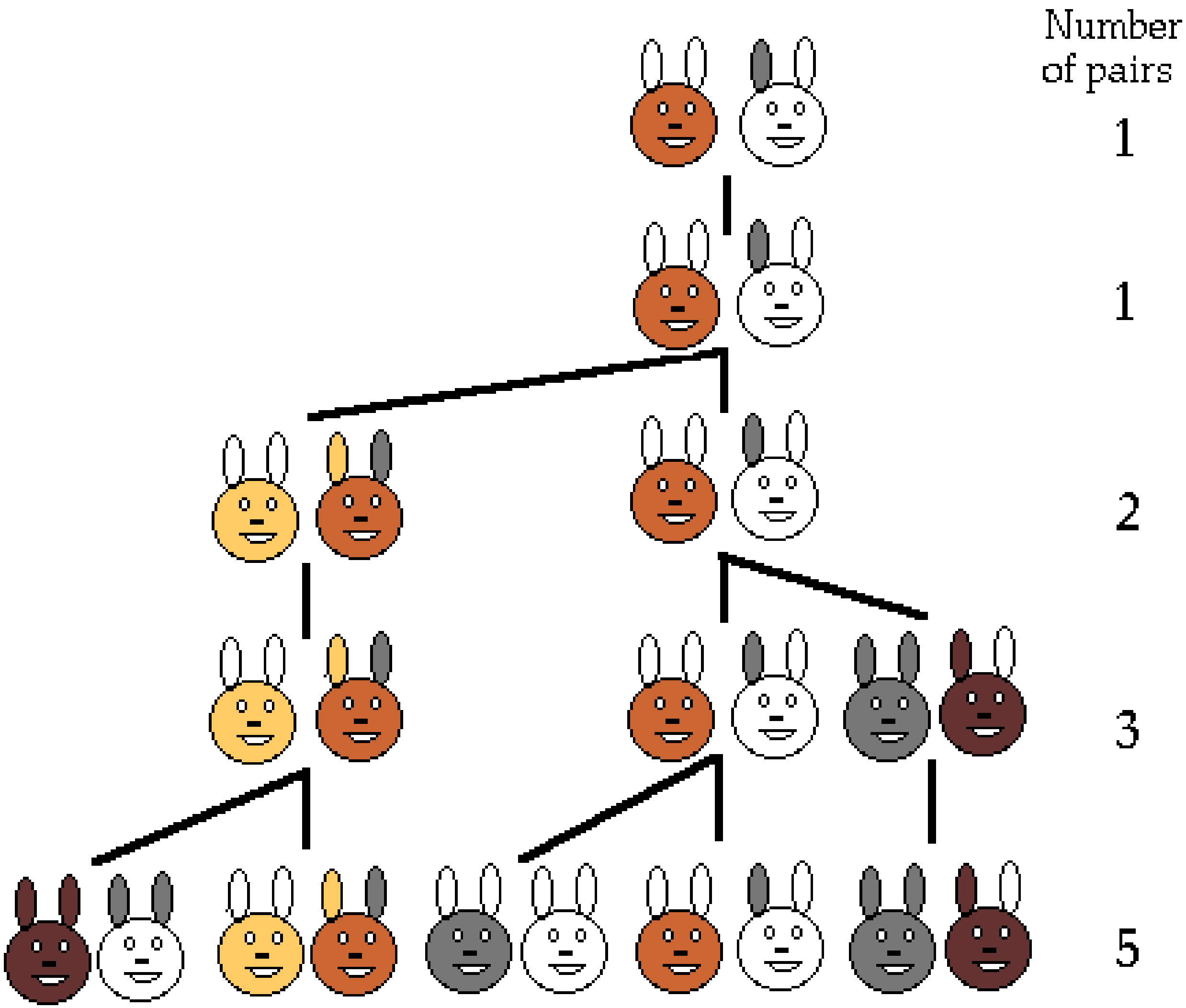
$$x(0)=0,\ \ x(1)=1$$

$$\{\, 0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ \dots \,\}$$

The sequence (although named after Fibonacci) has been reported to be known before that time, e.g. 200 BC, by Indian mathematicians.

From Wikipedia, (9.6.2009) http://en.wikipedia.org/wiki/Fibonacci_number

Fibonacci Sequence



Suppose a newly-born pair of rabbits, one male, one female, are put in a field.

Rabbits are able to mate at the age of one month so that at the end of its second month a female can produce another pair of rabbits.

Suppose that our rabbits never die and that the female always produces one new pair (one male, one female) every month from the second month on.

From Ron Knott's web pages on Mathematics, Fibonacci Numbers and Nature ,9.6.2009
http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html

Fibonacci Sequence

There is a fascinating connection between the Fibonacci Sequence and the **Golden Ratio**.
The limit of the quotient of two consecutive Fibonacci numbers is the very interesting number **ρ:**

$$\lim_{i \to \infty} \frac{x(i)}{x(i+1)} = \rho \approx \mathbf{0.618033988\ldots}$$

and the inverse **φ = 1/ρ** *is interesting as well*

$$\lim_{i \to \infty} \frac{x(i+1)}{x(i)} = \phi \approx \mathbf{1.618033988\ldots}$$

Golden Ratio

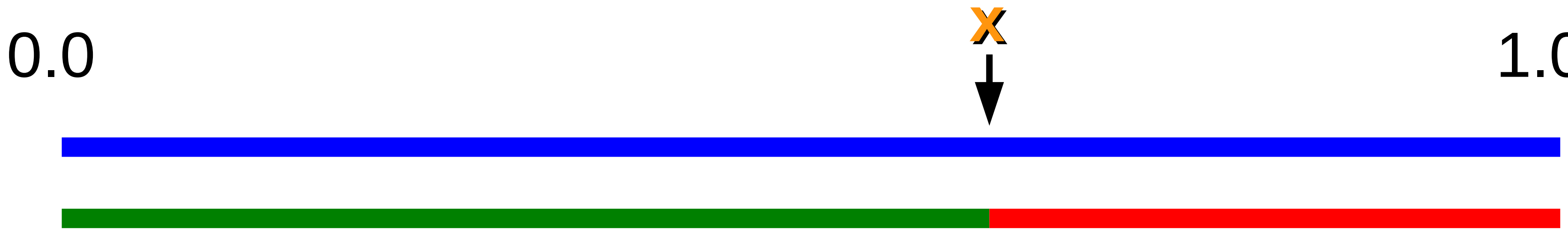
The **Golden Ratio** divides a given **interval** in a very special way into two pieces:

Golden Ratio

The **Golden Ratio** divides a given **interval** in a very special way into two pieces:

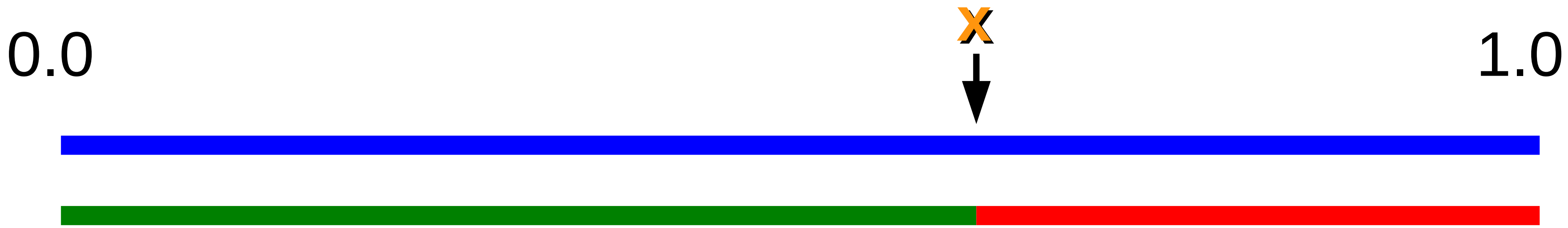
Golden Ratio

The **Golden Ratio** divides a given **interval** in a very special way into two pieces:



Golden Ratio

The **Golden Ratio** divides a given **interval** in a very special way into two pieces:



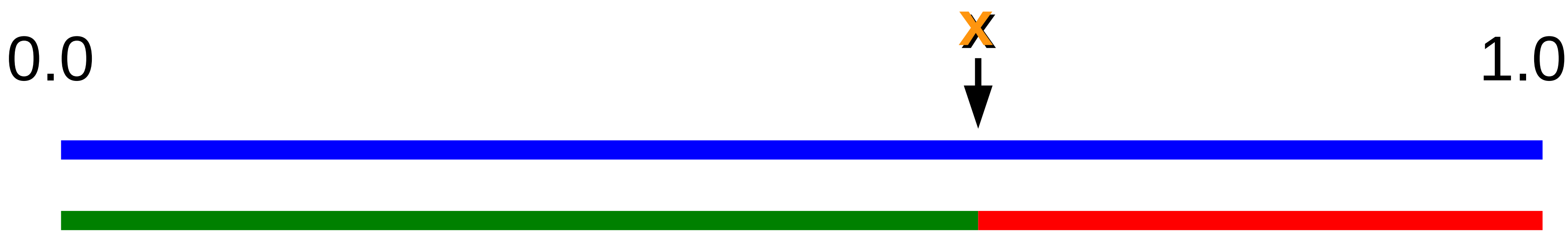
the **complete** part is to the **longer** part

as

the **longer** part is to the **shorter** part

Golden Ratio

The **Golden Ratio** divides a given **interval** in a very special way into two pieces:

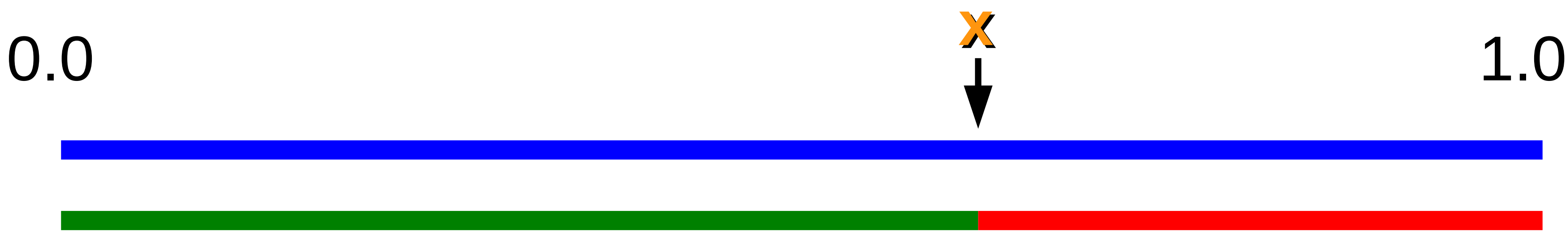


the **complete** part is to the **longer** part as the **longer** part is to the **shorter** part

$$1 / x \quad = \quad x / (1-x)$$

Golden Ratio

The **Golden Ratio** divides a given **interval** in a very special way into two pieces:



the **complete** part is to the **longer** part

as

the **longer** part is to the **shorter** part

$$1 / x = x / (1-x)$$

$$1 * (1-x) = x * x$$

$$(1-x) = x^2$$

$$\boxed{x^2 + x - 1 = 0}$$

Golden Ratio

Solving the quadratic equation yields two solutions:

$$x^2 + x - 1 = 0$$

Golden Ratio

Solving the quadratic equation yields two solutions:

$$x^2 + x - 1 = 0$$

$$-x_1 = \varphi \approx 1.618033988...$$

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618033988...$$

$$x_2 = \rho \approx 0.618033988...$$

Golden Ratio

Solving the quadratic equation yields two solutions:

$$x^2 + x - 1 = 0$$

$$-x_1 = \varphi \approx 1.618033988...$$

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618033988...$$

$$x_2 = \rho \approx 0.618033988...$$

φ and ρ are called the **Golden Ratio**,
they have some interesting algebraic properties:

Golden Ratio

Solving the quadratic equation yields two solutions:

$$x^2 + x - 1 = 0$$

$$-x_1 = \varphi \approx 1.618033988... \qquad \phi = \frac{1+\sqrt{5}}{2} \approx 1.618033988...$$

$$x_2 = \rho \approx 0.618033988...$$

φ and ρ are called the **Golden Ratio**,
they have some interesting algebraic properties:

$$\varphi - \rho = 1 \qquad 1/\varphi = \varphi - 1 \qquad \varphi^2 = 1 + \varphi$$

$$\varphi * \rho = 1 \qquad 1/\rho = \rho + 1 \qquad \rho^2 = 1 - \rho$$

Golden Ratio

Solving the quadratic equation yields two solutions:

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618033988...$$

φ and ρ are called the **Golden Ratio**,
they have some interesting algebraic properties:

Logistic Growth

Unbounded reproduction is not realistic.
Several circumstances exist (e.g. restricted resources) that yield to a maximal, bounded growth rate.

Logistic Growth

Unbounded reproduction is not realistic.
Several circumstances exist (e.g. restricted resources) that yield to a maximal, bounded growth rate.

$x(i+1) = x(i) + g\,x(i)$ with a growth rate g

The growth rate g shall now reflect restricted resources.
The available resources are those resources that have been left over by the population $x(i)$.

Thus, g is modeled as a growth rate a, that is bounded by

$g = a\,(M - x(i))$ with *M* maximum of the resources.

Logistic Growth

Thus a modified growth formula can be used:

$$x(i+1) = x(i) + g\ x(i) \qquad \text{with } g = a(M - x(i))$$

$$= x(i) + a\,(M - x(i))\,x(i)$$

This iterated function implements a differential equation to model realistic growth of populations.
It was published by P.-F. Verhulst (1838) :

$$\frac{dP}{dt} = P\left(1 - P\right)$$

From Wikipedia, (9.6.2009)
http://en.wikipedia.org/wiki/Logistic_function

Logistic Growth

Iterating the equation with a small initial value of $x(0)$

$$x(i+1) = x(i) + a(M-x(i))\, x(i)$$

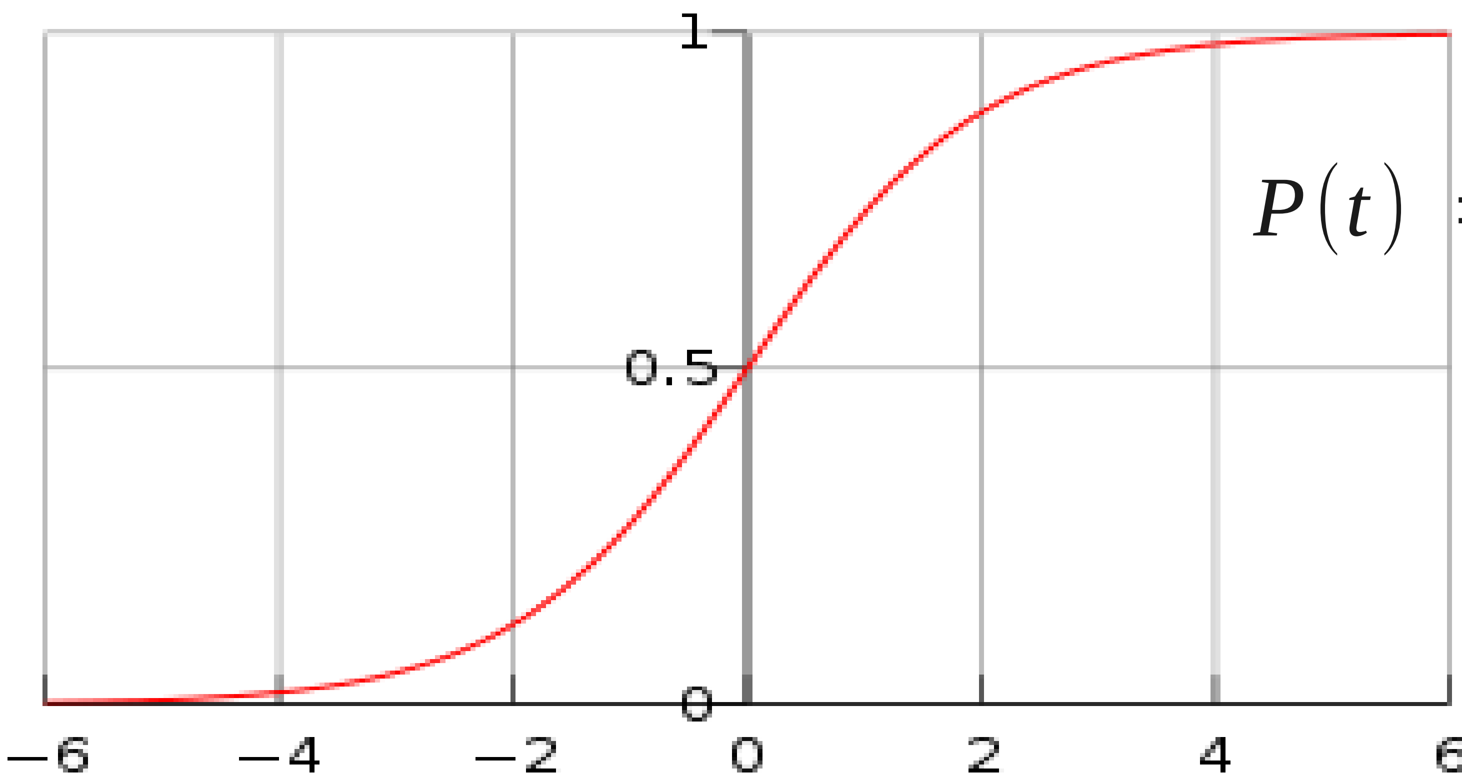
The population grows, the number $x(i)$ of individuals within the population, starts to increase with almost exponential growth. In this phase, the increase is getting larger from step to step.

This growth continues until $x(i)$ reaches a value of $x(j)=½\, M$. After that, the increase is getting smaller and smaller, until $x(i)$ converges to the maximal possible value of $x(i) = M$.

Logistic Growth

Solving the Verhulst differential equation yields the logistic function with its typical sigmoid shape.

$$\frac{dP}{dt} = P(1-P)$$



$$P(t) = \frac{1}{1+e^{-t}}$$

From Wikipedia, (9.6.2009): http://en.wikipedia.org/wiki/Logistic_function

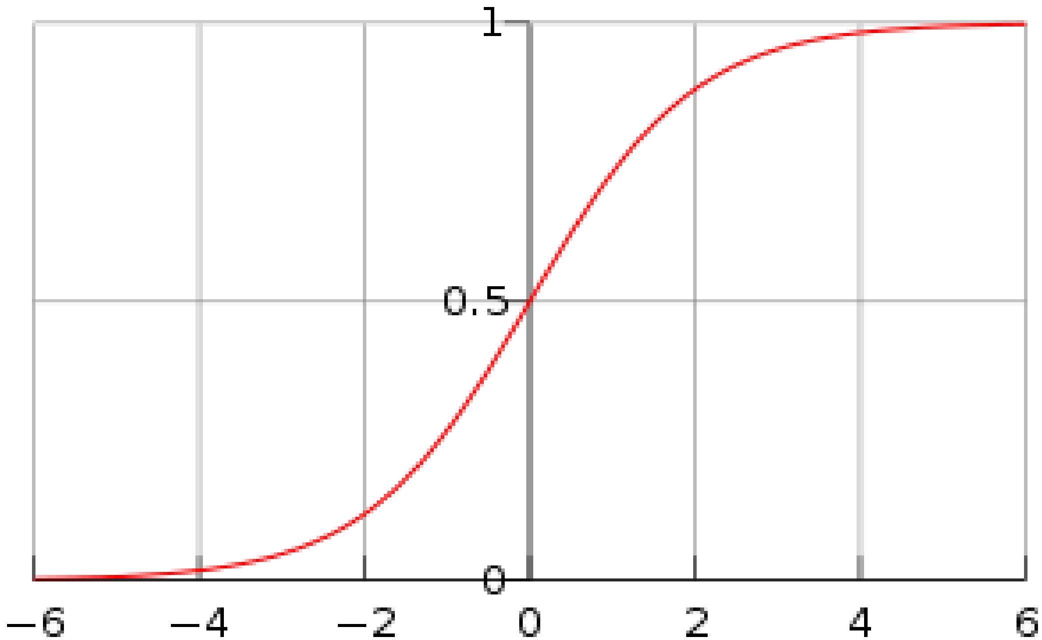
Logistic Growth

Solving the Verhulst differential equation yields the logistic function with its typical sigmoid shape.

$$\frac{dP}{dt} = P\left(1-P\right)$$

$$P\left(t\right) = \frac{1}{1+\mathrm{e}^{-t}}$$

In the field of neural networks one can find this function denoted as: Sigmoid function or Fermi function.



From Wikipedia, (9.6.2009)
http://en.wikipedia.org/wiki/Logistic_function

Logistic Map / Logistic Parabola

A special version of logistic growth is the function called:
Logistic Map or **Logistic Parabola**

$$x(i+1) = a\, x(i)\, (\, 1 - x(i)\,) \qquad \text{with } 0.0 < a < 4.0$$

The logistic map is famous within the community researching on nonlinear dynamics and chaos. It shows a wide variety of nonlinear effects known to be related with deterministic chaos;

- period doubling route to chaos
- self similarity
- positive Ljapunov exponents
- strange attractor
- fractal dimension
- Hopf bifurcations, Pitchfork bifurcations

Predator-Prey System

Consider a situation with more than one species that interact with each other:

Predator-Prey System

Consider a situation with more than one species that interact with each other:

Predator and Prey

one species (prey) is chased by the other species (predator).

Predator-Prey System

Consider a situation with more than one species that interact with each other:

Predator and Prey

one species (prey) is chased by the other species (predator).

Both, predator and prey are growing,
and both are interacting.
The dynamics can be modeled by the following equations:

$$N(i+1) = N(i) + aN(i) - b\,N(i)P(i)$$

$$P(i+1) = P(i) + cP(i)N(i) - d\,P(i)$$

Predator-Prey System

$$N(i+1) = N(i) + aN(i) - b\,N(i)P(i) \qquad \textit{prey}$$

$$P(i+1) = P(i) + cP(i)N(i) - d\,P(i) \qquad \textit{predator}$$

$N(i)$ is the number of prey individuals:
$N(i)$ is increasing with the growth factor a *(birth)*
and is decreasing due to limited resources $bN(i)$
and due to the number of predators $P(i)$ *(hunt).*

$P(i)$ his the number of predators:
$P(i)$ is increasing proportional to the number of predators $P(i)$ *(birth)* and the number of prey $N(i)$ *(food)*,
and is decreasing due to limited resources defined by d.

Lotka-Volterra Equations

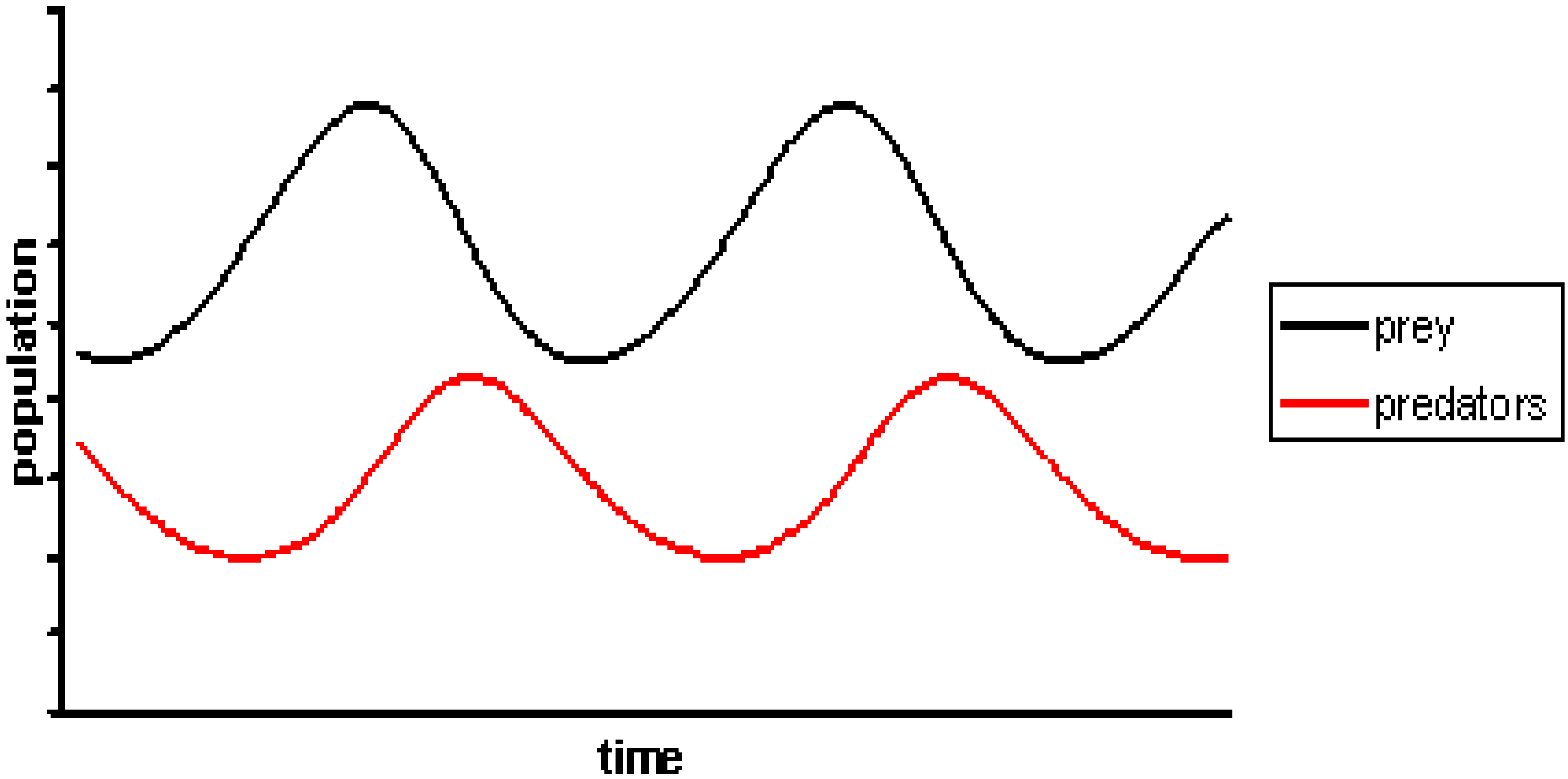
The prey-predator equations written as system of two coupled differential equations:
Lotka-Volterra (1925, 1926)

$$\partial N/\partial t = aN - bPN = N(a - bP) \quad \text{(prey)}$$
$$\partial P/\partial t = cPN - dP = P(cN - d) \quad \text{(predator)}$$

Several observations from real animal data indicate that the Lotka-Volterra model is describing population dynamics sufficiently:
Hudson-Bay Company, 1845 – 1935, lynx, hares.

Lotka-Volterra Equations

The prey-predator equations from Lotka and Volterra establish a periodic oscillation over time.



From Wikipedia, (9.6.2009)http://en.wikipedia.org/wiki/Lotka-Volterra_equation

Activator-Inhibitor Equations

The prey-predator equations are an example of a complete family of differential equations called:

Activator – Inhibitor Systems

Typical **Activator – Inhibitor** systems have two components that interact with each other.

One component is called the **Activator**, making the system grow. The other component is called the **Inhibitor**, which is the component that restricts the system and prevents the growth from getting exponential.

Activator-Inhibitor Equations

$$X(i+1) = X(i) + aX(i) - bY(i) \qquad \textit{activator}$$

$$Y(i+1) = Y(i) + cX(i) - dY(i) \qquad \textit{inhibitor}$$

Activator-Inhibitor Equations

$$X(i+1) = X(i) + aX(i) - bY(i) \quad \text{activator}$$
$$Y(i+1) = Y(i) + cX(i) - dY(i) \quad \text{inhibitor}$$

The ***Activator X*** enforces it's own growth,
and enforces the growth of the inhibitor.

$$X(i+1) = \ldots + aX(i) \ldots \qquad Y(i+1) = \ldots + cX(i) \ldots$$

Activator-Inhibitor Equations

$$X(i+1) = X(i) \quad +aX(i) \quad - bY(i) \qquad \textit{activator}$$

$$Y(i+1) = Y(i) \quad +cX(i) \quad - dY(i) \qquad \textit{inhibitor}$$

The ***Activator*** ***X*** enforces it's own growth,
and enforces the growth of the inhibitor.

$$X(i+1) = \ldots + aX(i) \ldots \qquad Y(i+1) = \ldots +cX(i) \ldots$$

The ***Inhibitor*** ***Y*** constrains it's own growth,
and constrains the growth of the activator.

$$X(i+1) = \ldots - bY(i) \ldots \qquad Y(i+1) = \ldots - dY(i) \ldots$$

Activator-Inhibitor Equations

Activator-Inhibitor characteristics can be found in a lot of systems from biology, physics, medicine and techniques.

Activator-Inhibitor Equations

Activator-Inhibitor characteristics can be found in a lot of systems from biology, physics, medicine and techniques:

Morphogenesis
plants, animals

Textures
sea shells, tiger stripes (Alan Turing 1952)

Generation of rhythmic patterns
heartbeat, locomotion

Limb control, muscles
antagonistic pairs, flexor-extensor

Population dynamics
predator-prey, birthrate-deathrate

Activator-Inhibitor Equations

Activator-Inhibitor characteristics can be found in a lot of systems from biology, physics, medicine and techniques:

Activator-Inhibitor Equations

Activator-Inhibitor characteristics can be found in a lot of systems from biology, physics, medicine and techniques:

Very often the Activator-Inhibitor principle is combined with a **spatial component**.

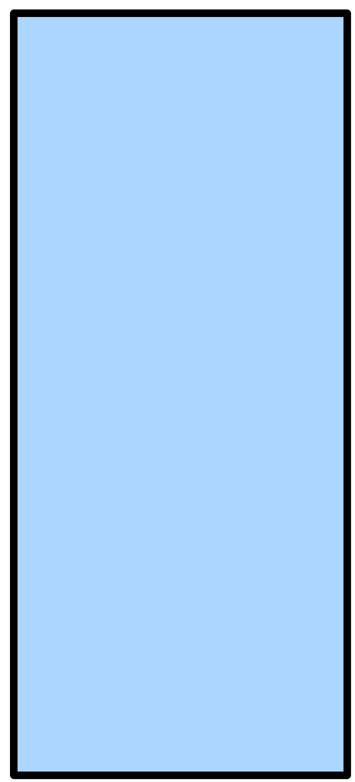
Not only the conditions at one spatial position are considered, but also the conditions from the **direct neighborhood** are taken into account via a diffusion process.

Reaction-Diffusion System

The differential equations of a **Reaction**-**Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.

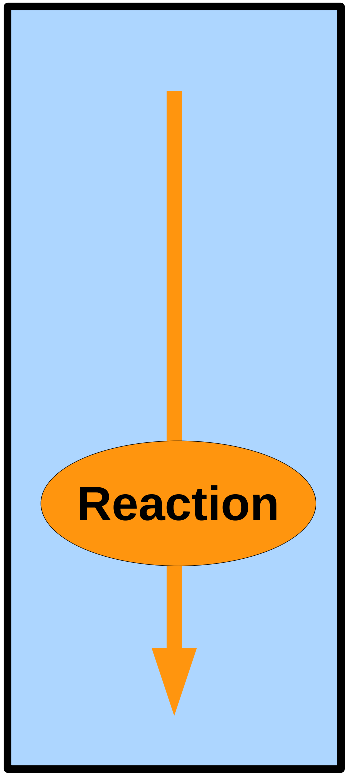
Reaction-Diffusion System

The differential equations of a **Reaction-Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



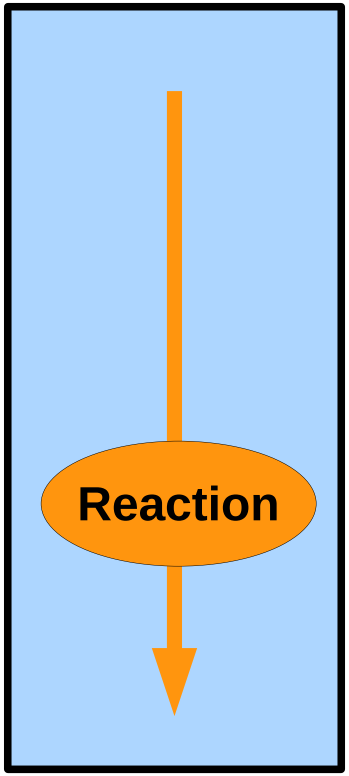
Reaction-Diffusion System

The differential equations of a **Reaction-Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



Reaction-Diffusion System

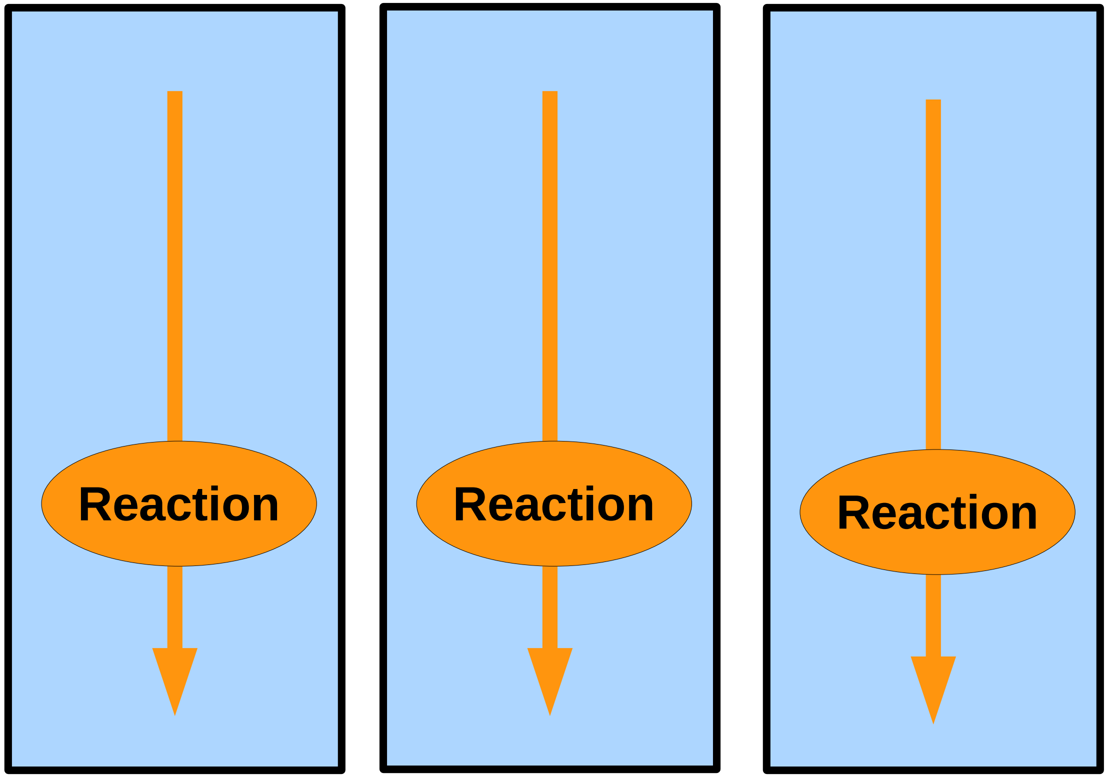
The differential equations of a **Reaction-Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



$$dV/dt = f_V(V)$$

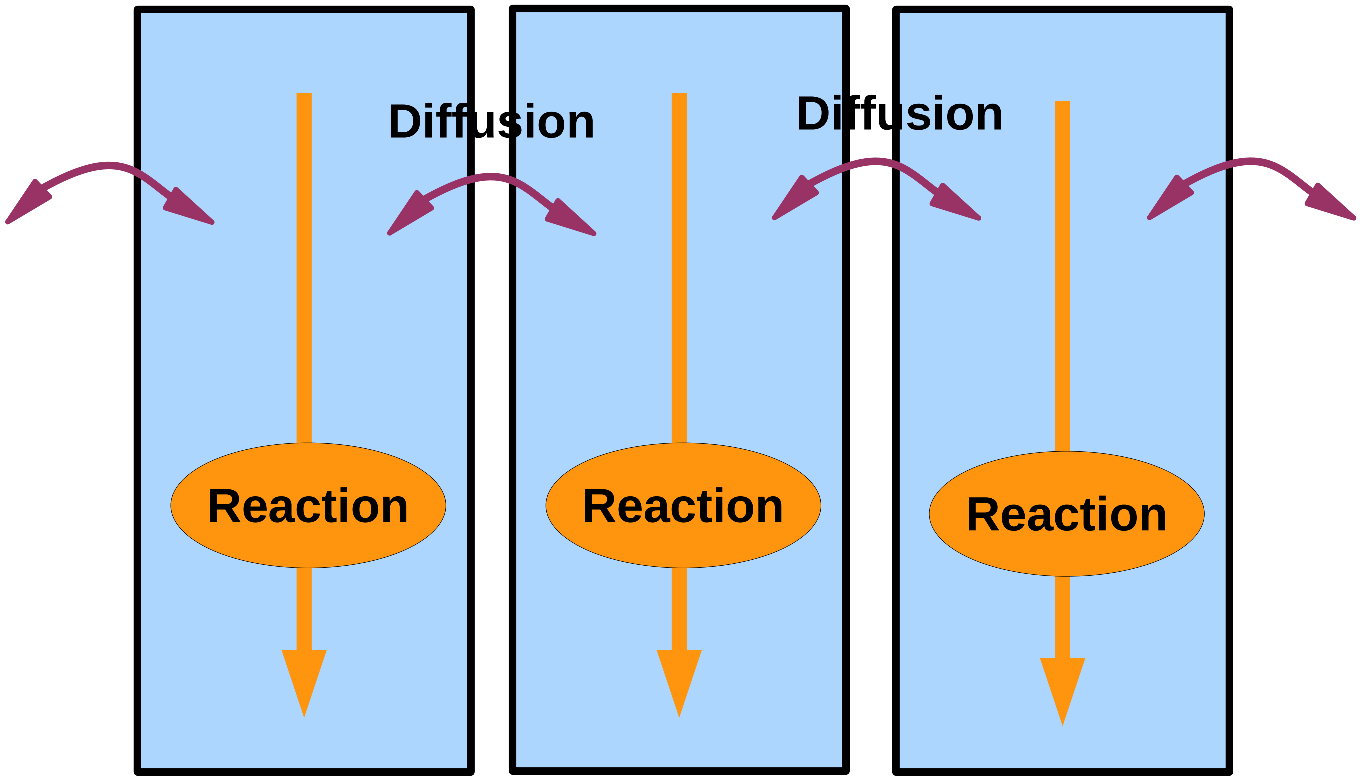
Reaction-Diffusion System

The differential equations of a **Reaction-Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



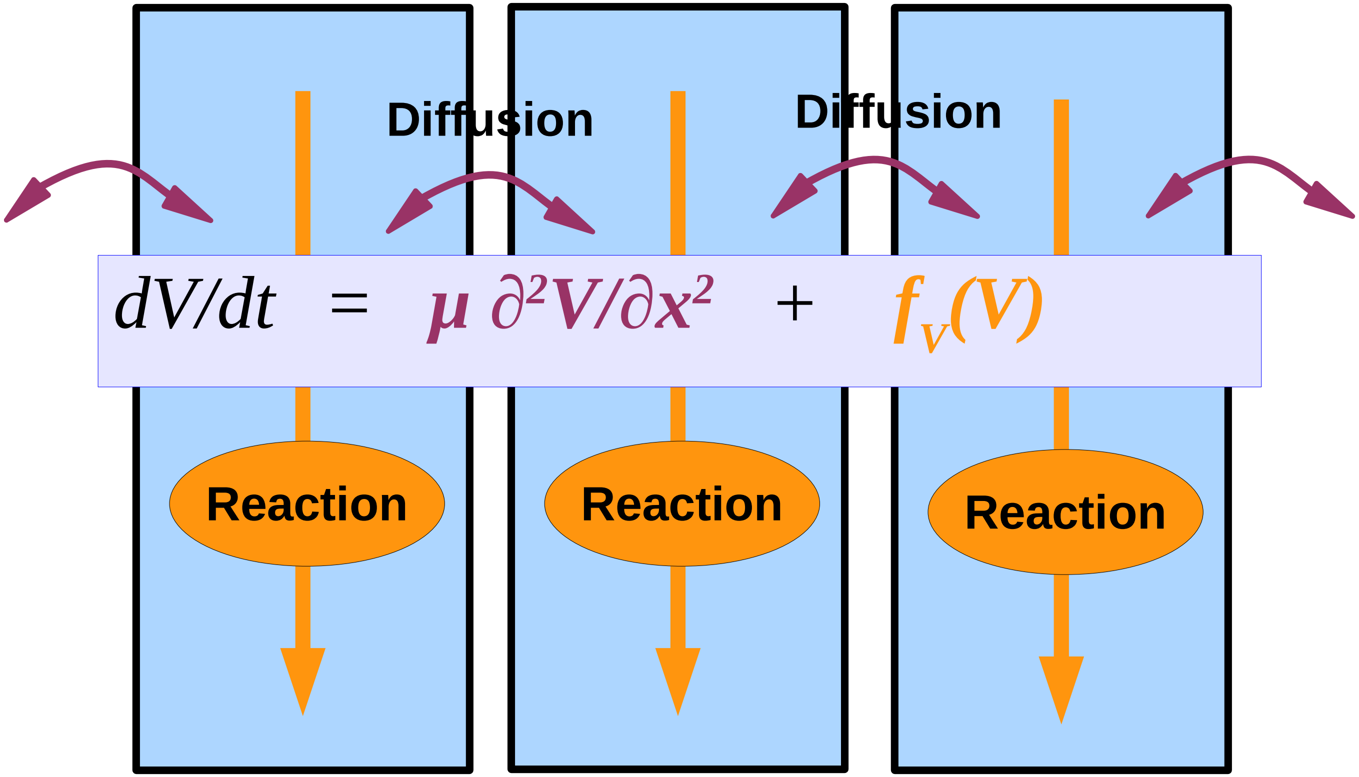
Reaction-Diffusion System

The differential equations of a **Reaction**-**Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



Reaction-Diffusion System

The differential equations of a **Reaction-Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.



$$dV/dt = \mu\, \partial^2 V/\partial x^2 + f_V(V)$$

Reaction-Diffusion System

The differential equations of a **Reaction**-**Diffusion** system try to model the local influence from the neighborhood via a diffusion processes and the activity within one compartment using the reaction terms.

$$dV/dt = \mu\, \partial^2 V/\partial x^2 + f_V(V,W)$$

$$dW/dt = \sigma\, \partial^2 W/\partial x^2 + f_W(V,W)$$

Example for a 1-dim Reaction-Diffusion system
with the diffusion terms: $\mu\, \partial^2 V/\partial x^2$ and $\sigma\, \partial^2 W/\partial x^2$
and the reaction terms: $f_V(V,W)$ and $f_W(V,W)$

Reaction-Diffusion System

Cellular automata can be seen as discretized versions of Reaction-Diffusion systems.

Reaction-Diffusion System

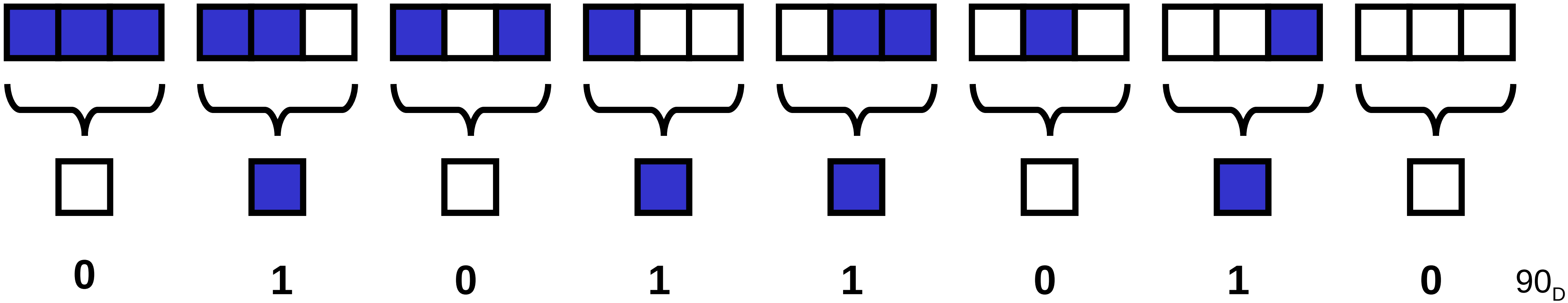
Cellular automata can be seen as discretized versions of Reaction-Diffusion systems.

The rule of a cellular automaton takes the state of the direct neighborhood into account, thus the diffusion part of a Reaction-Diffusion system can be easily implemented; the reaction term is straight forward.

Reaction-Diffusion System

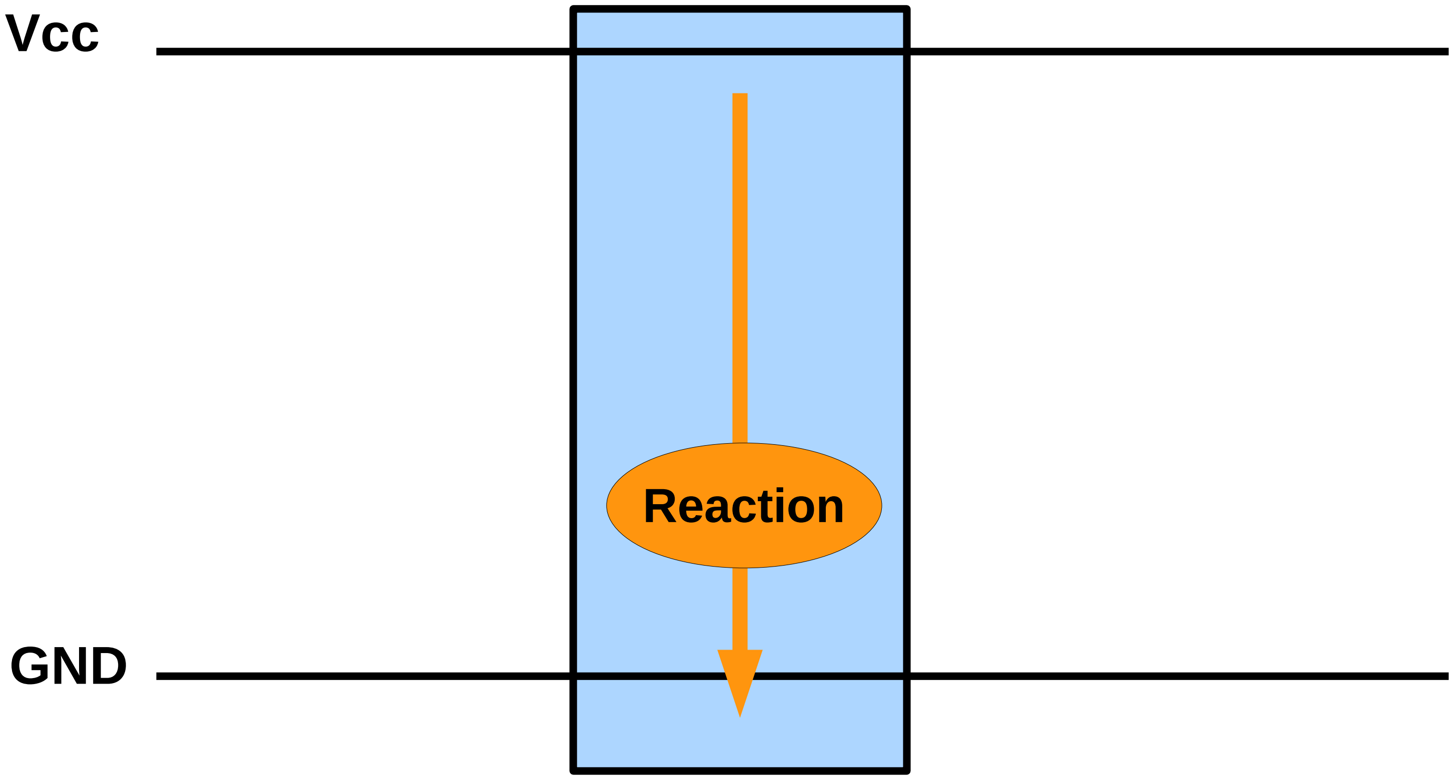
Cellular automata can be seen as discretized versions of Reaction-Diffusion systems.

The rule of a cellular automaton takes the state of the direct neighborhood into account, thus the diffusion part of a Reaction-Diffusion system can be easily implemented; the reaction term is straight forward.



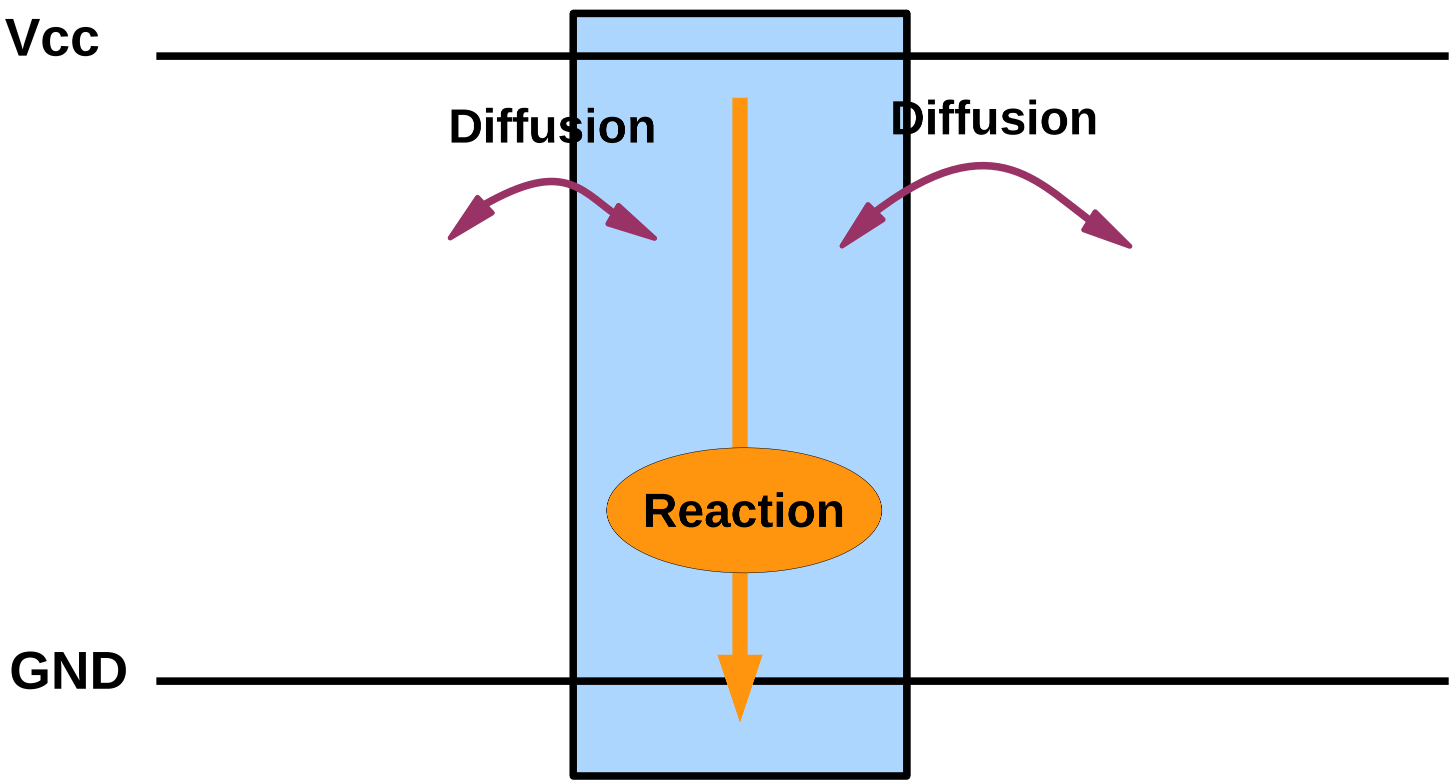
Reaction-Diffusion System

A real implementation of an **Activator-Inhibitor** **Reaction-Diffusion** system is a field (1 or 2-dim) of nonlinear electrical oscillators.



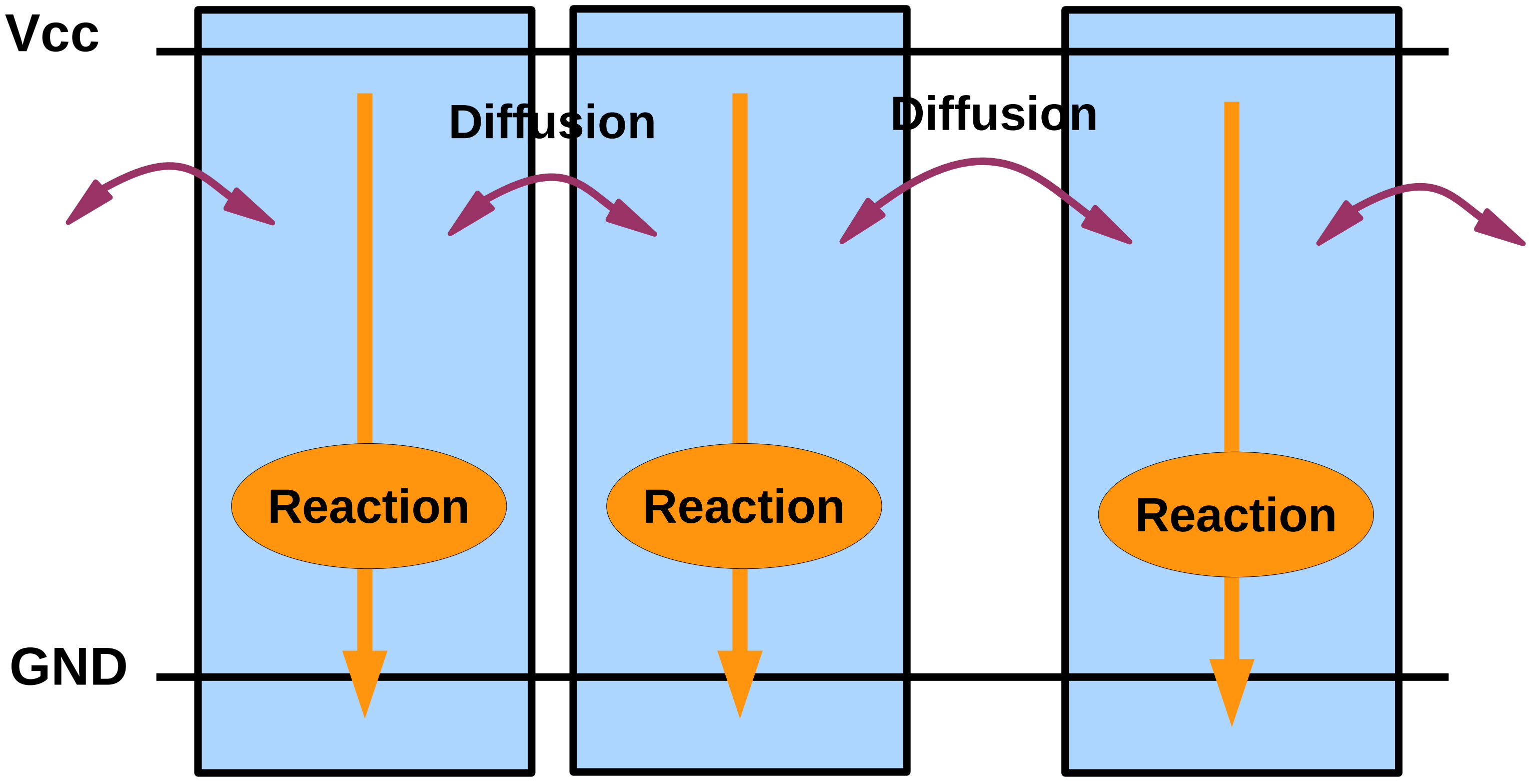
Reaction-Diffusion System

A real implementation of an **Activator-Inhibitor** **Reaction-Diffusion** system is a field (1 or 2-dim) of nonlinear electrical oscillators.



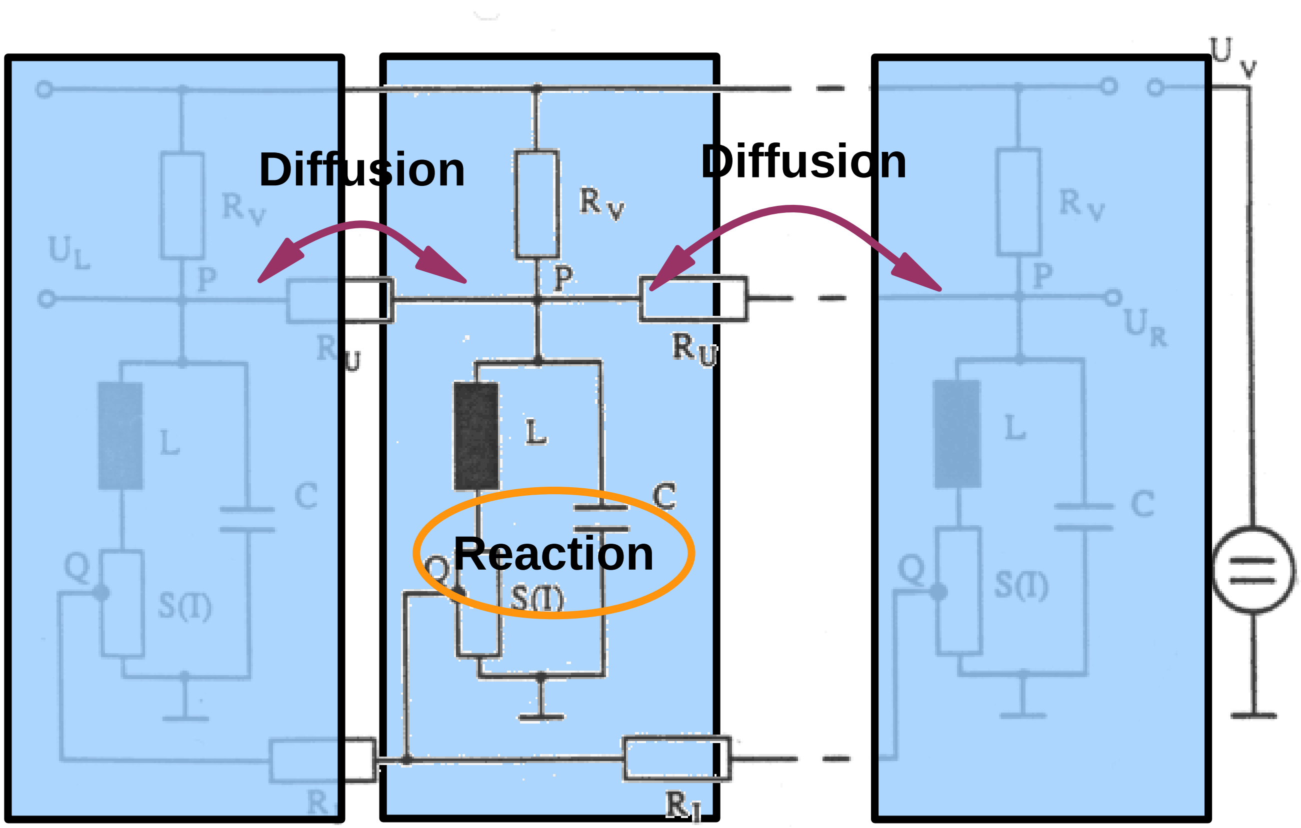
Reaction-Diffusion System

A real implementation of an **Activator-Inhibitor** Reaction-Diffusion system is a field (1 or 2-dim) of nonlinear electrical oscillators.



Reaction-Diffusion System

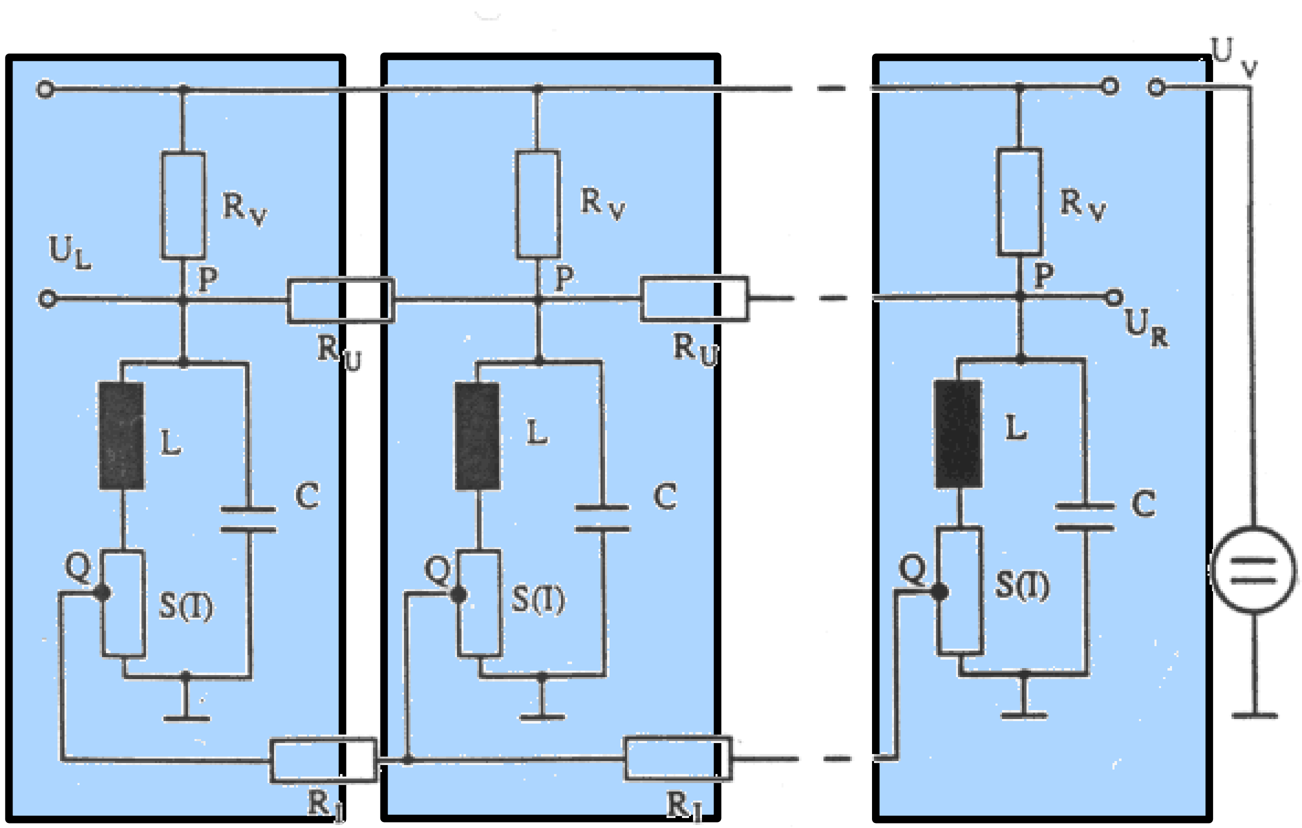
A real implementation of an **Activator-Inhibitor** Reaction-**Diffusion** system is a field (1 or 2-dim) of nonlinear electrical oscillators.



http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction-Diffusion System

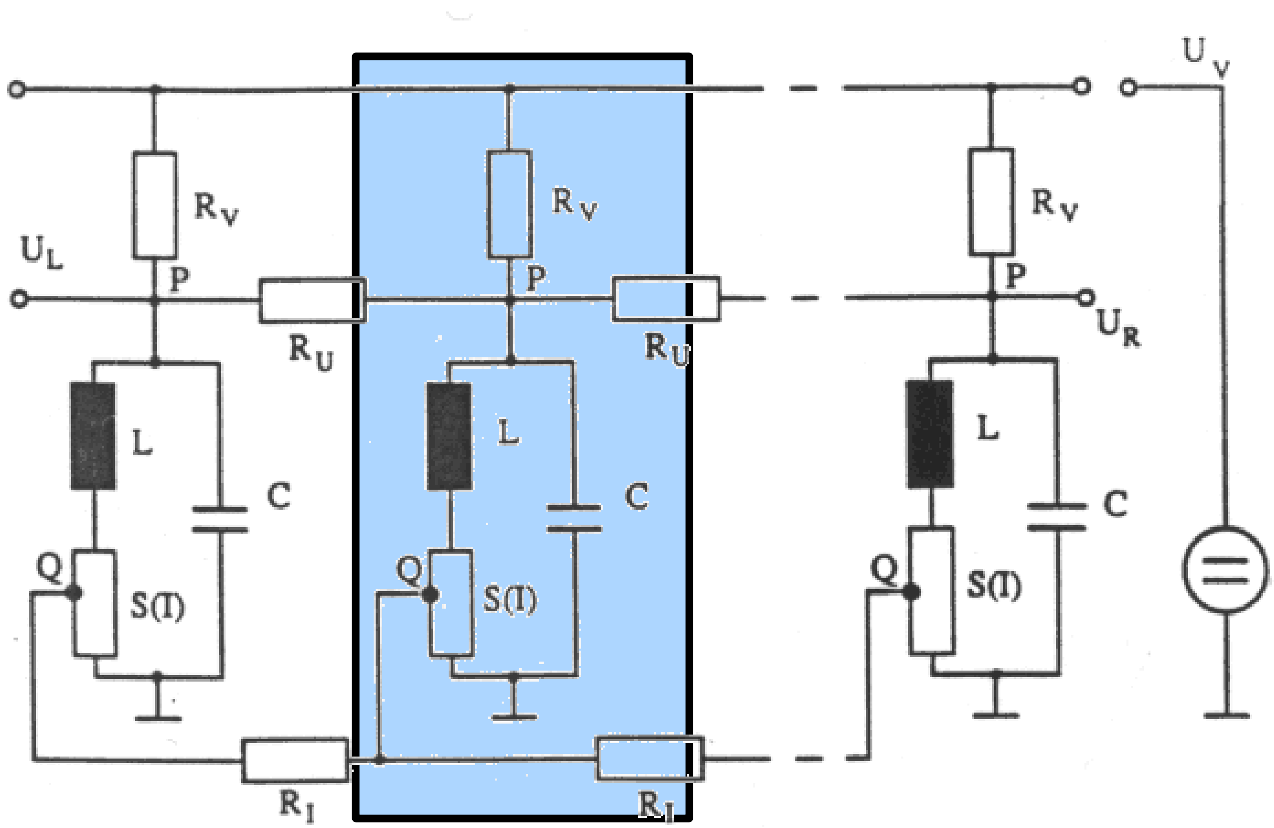
A real implementation of an **Activator-Inhibitor** **Reaction-Diffusion** system is a field (1 or 2-dim) of nonlinear electrical oscillators.

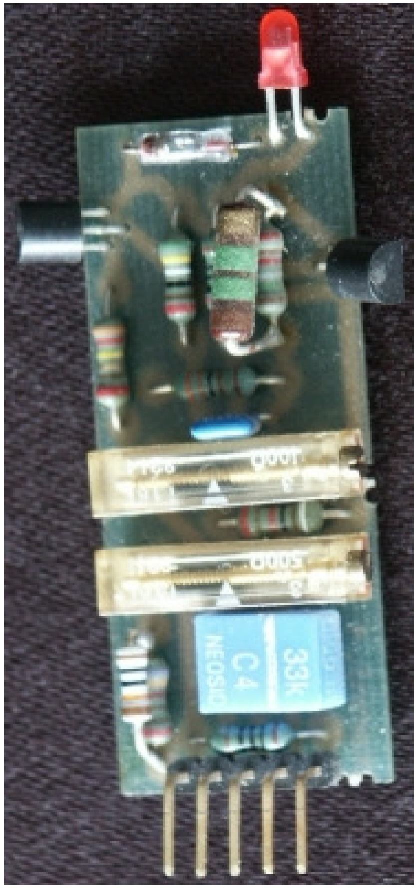


http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction-Diffusion System

A real implementation of an **Activator-Inhibitor** Reaction-Diffusion system is a field (1 or 2-dim) of nonlinear electrical oscillators.

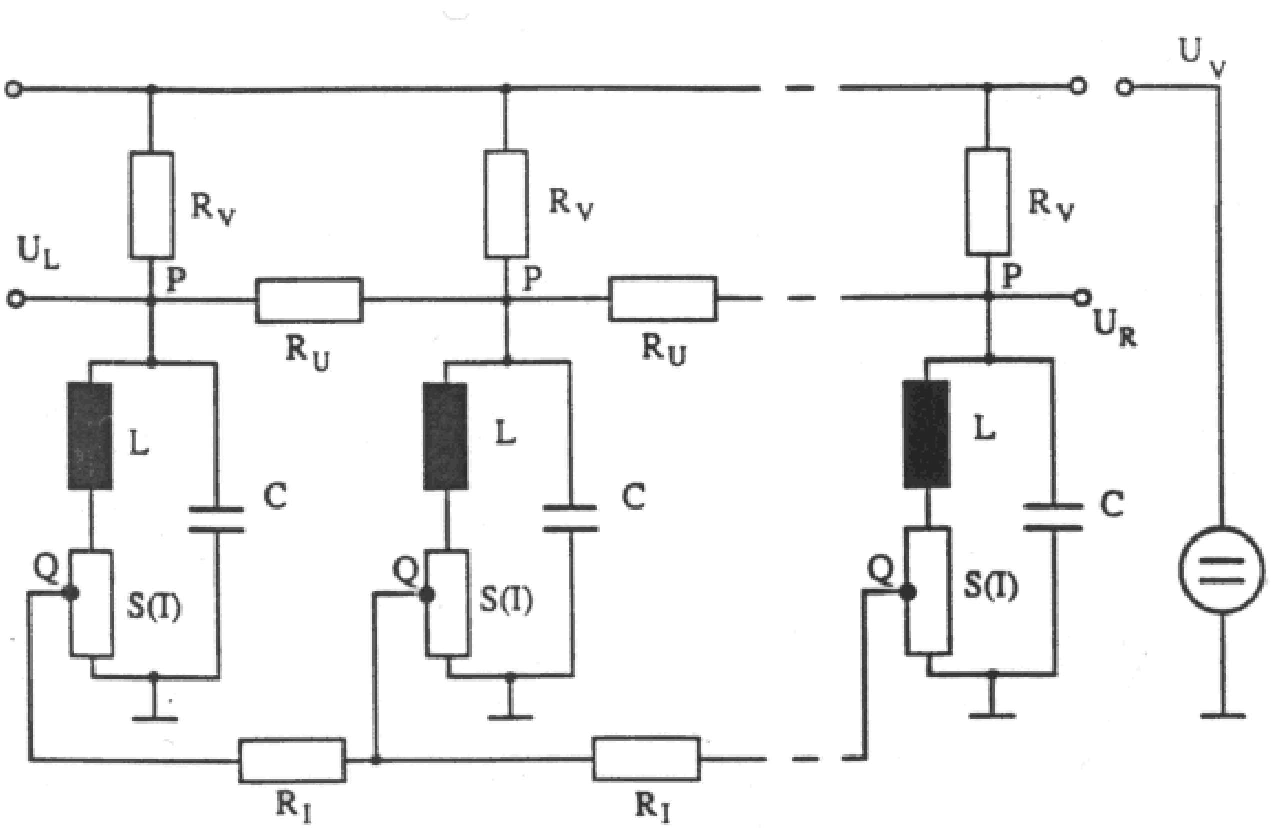




http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

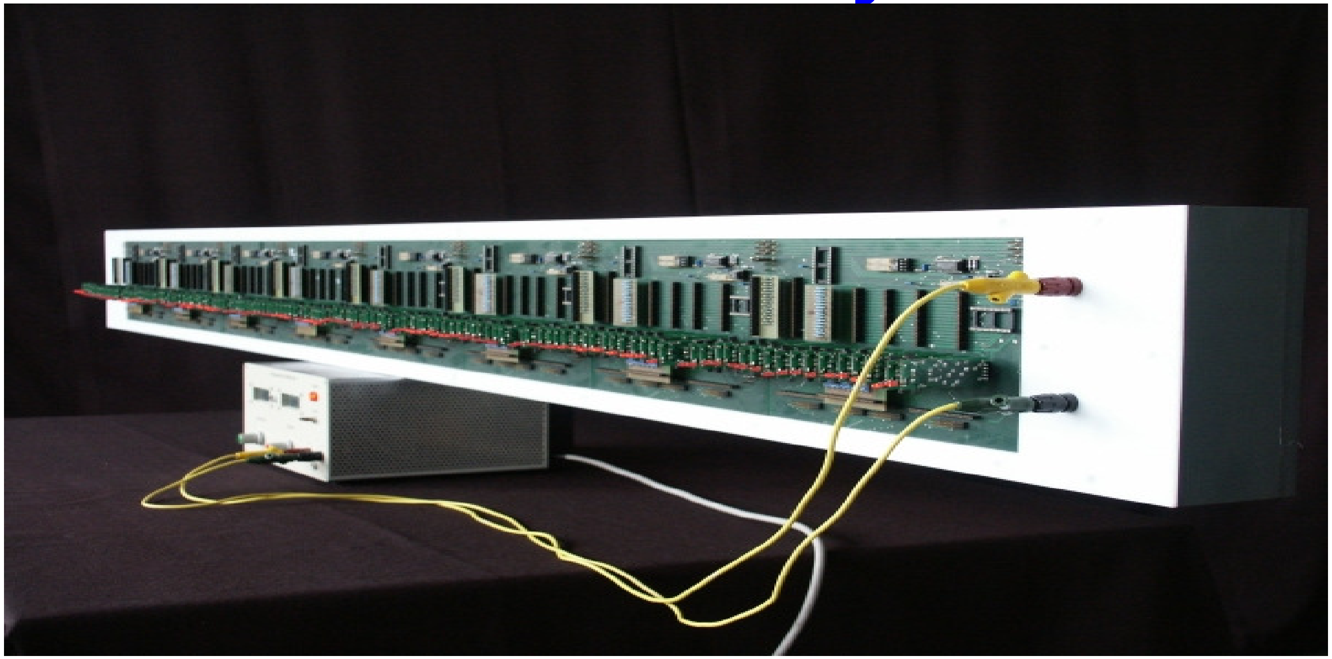
Reaction-Diffusion System

A real implementation of an **Activator-Inhibitor** Reaction-Diffusion system is a field (1 or 2-dim) of nonlinear electrical oscillators.



http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

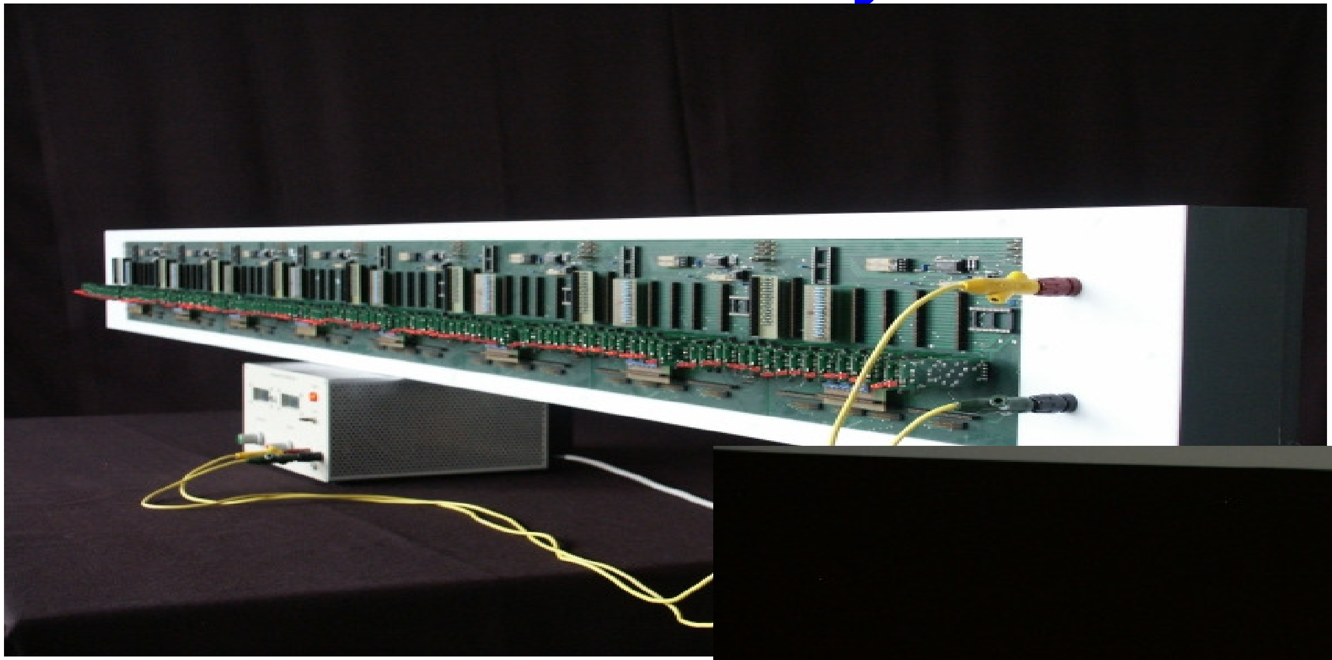
Reaction-Diffusion System



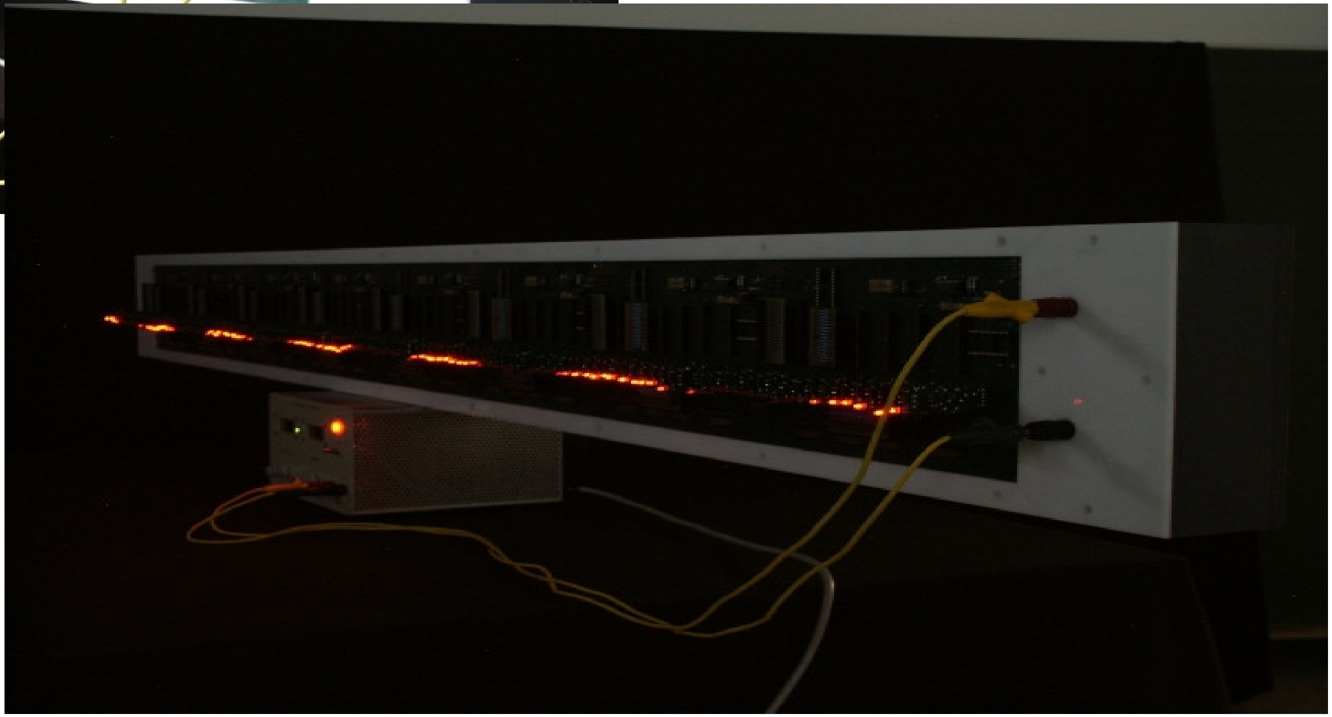
128 oscillators are connected to a one-dimensional field.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction-Diffusion System



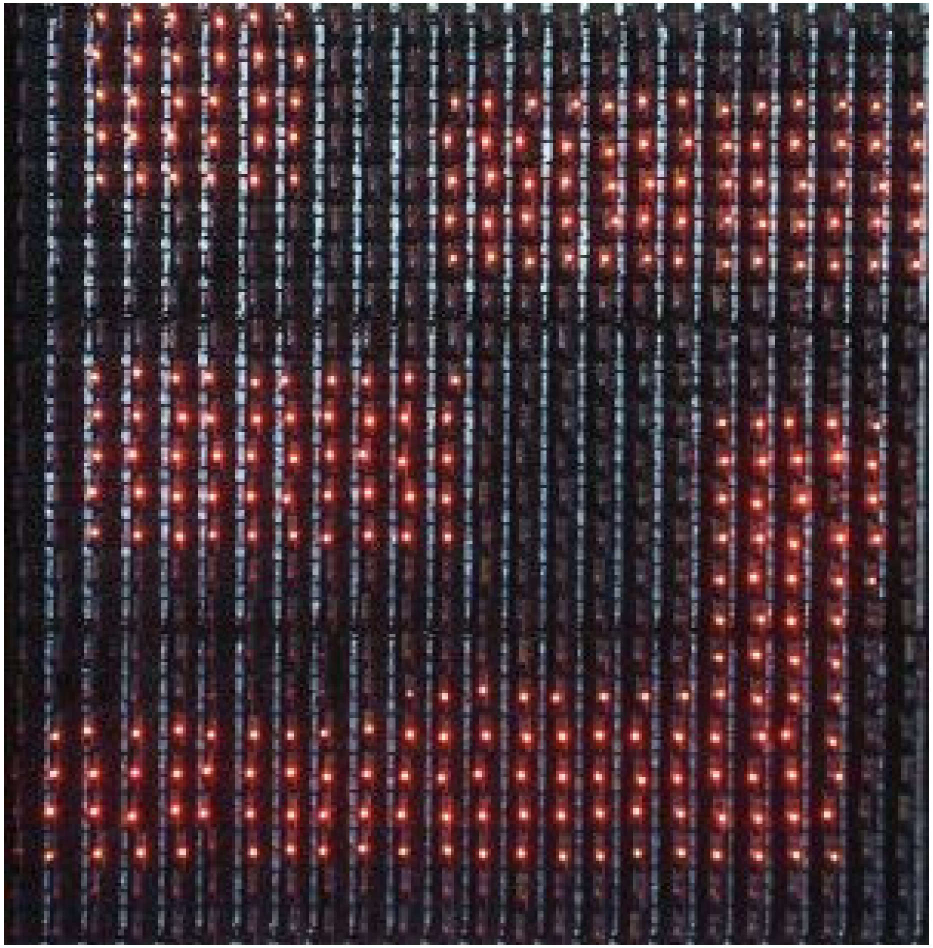
128 oscillators are connected to a one-dimensional field.

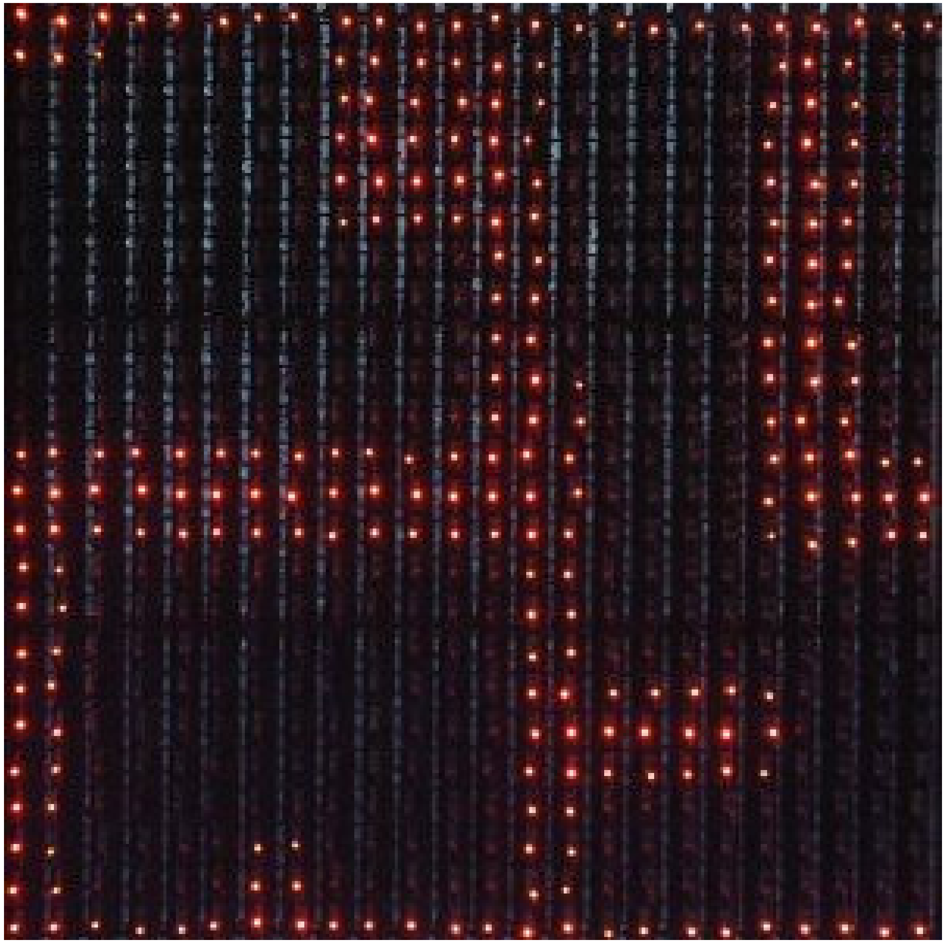


Spatial structures emerge as a result of coupling.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction-Diffusion System

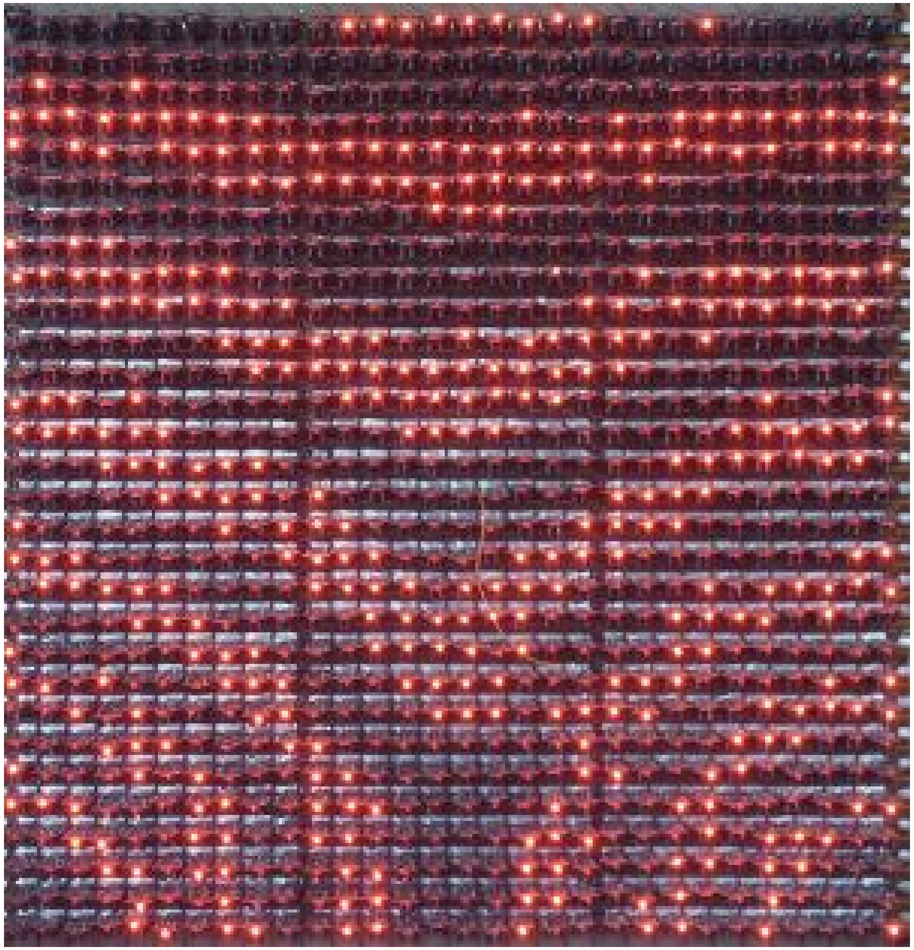


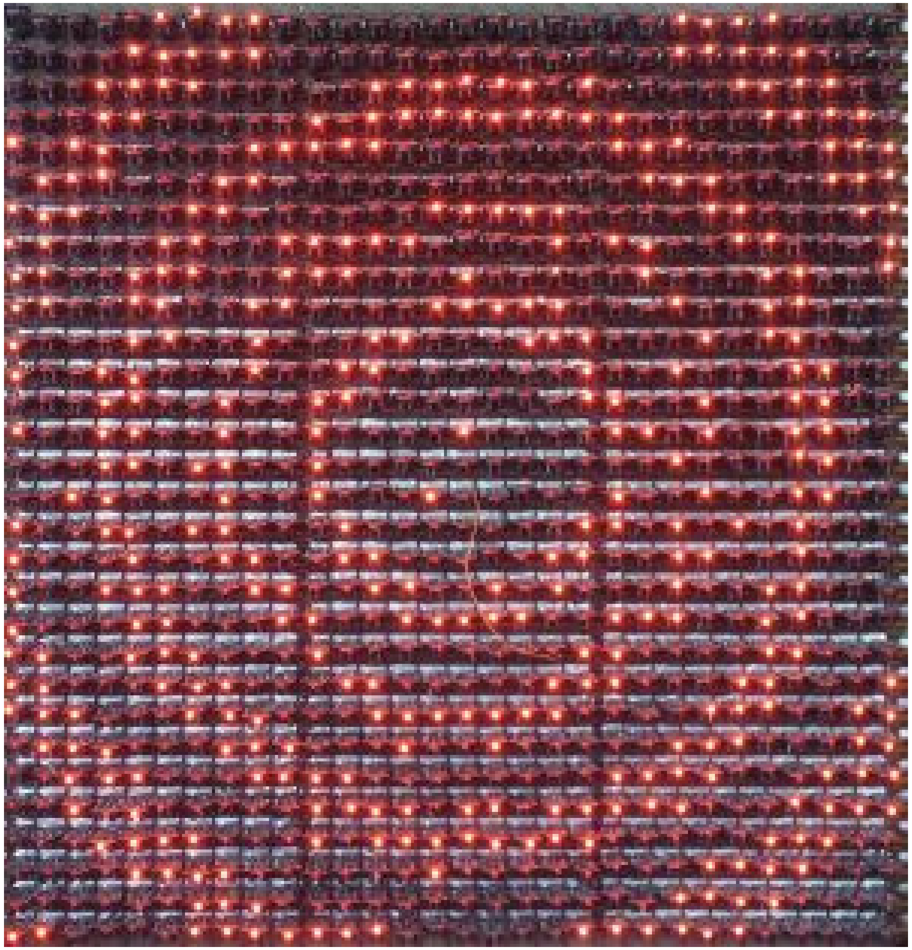


Spatial structures emerge as a result of coupling.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction-Diffusion System

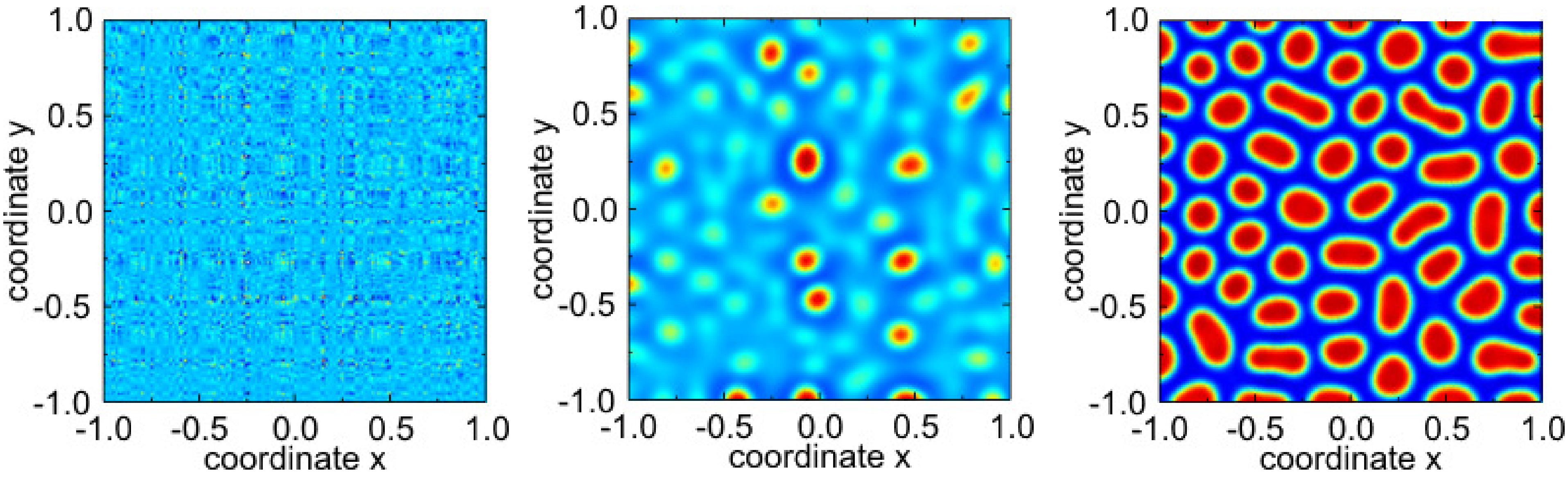




Spatial structures emerge as a result of coupling.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/ElNw-de.html

Reaction – Diffusion System

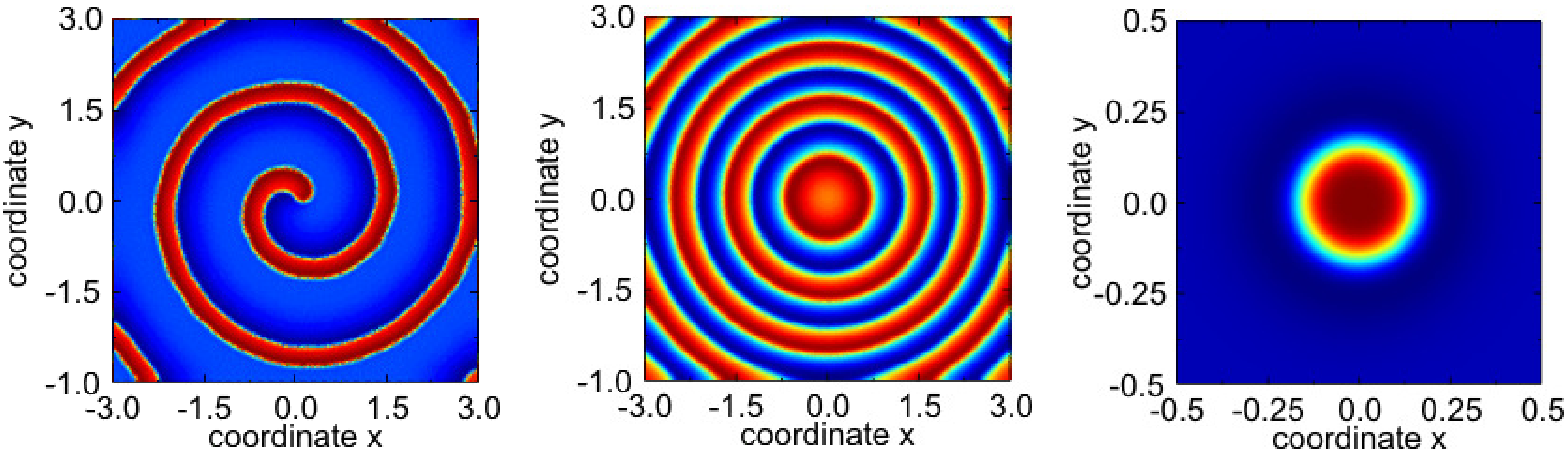


Formation of a hexagonal pattern from noisy initial conditions in the two-component reaction-diffusion system of Fitzhugh–Nagumo type:

$$\partial_t u = d_u^2 \nabla^2 u + f(u) - \sigma v,$$
$$\tau \partial_t v = d_v^2 \nabla^2 v + u - v$$

https://en.wikipedia.org/wiki/Reaction%E2%80%93diffusion_system

Reaction – Diffusion System

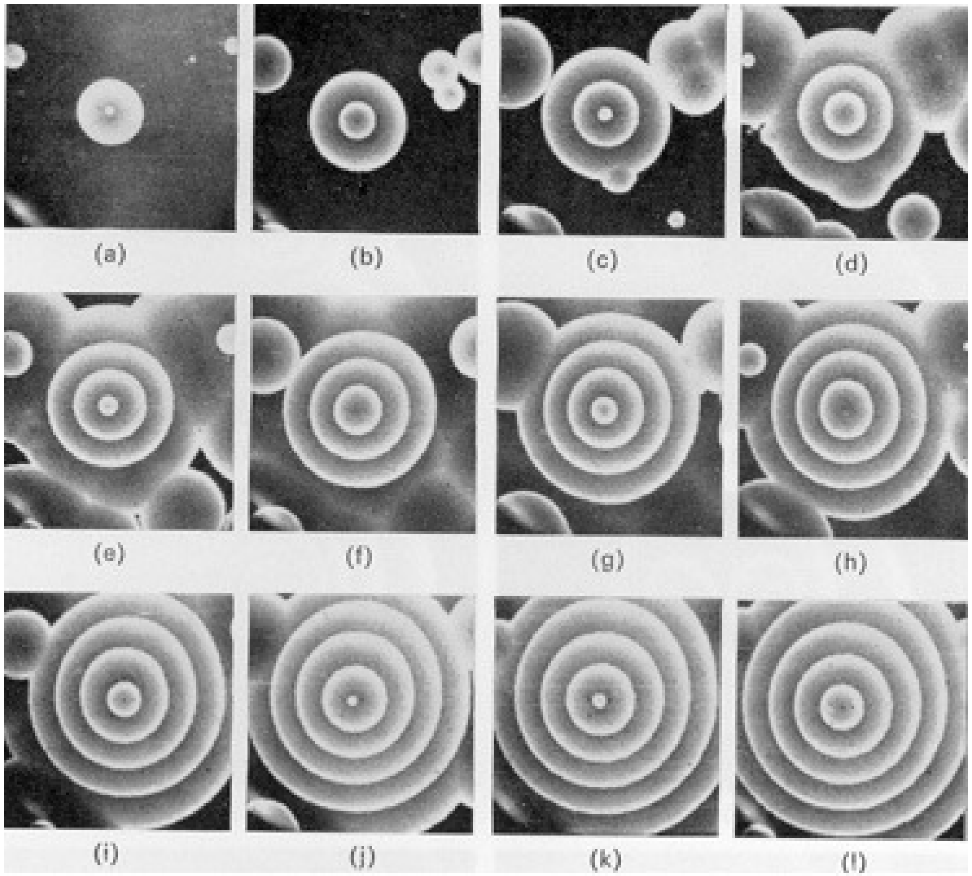


Other patterns in the two-component reaction-diffusion system of Fitzhugh–Nagumo type:

https://en.wikipedia.org/wiki/Reaction%E2%80%93diffusion_system

Belousov-Zhabotinsky-Reaction

Example for an Activator-Inhibitor Reaction-Diffusion System from chemistry.



$$HBrO_3 + HBrO_2 \rightarrow 2BrO_2^{\bullet} + H_2O$$

$$H^+ + BrO_2^{\bullet} + Fe(phen)_3^{2+} \rightarrow Fe(phen)_3^{3+} + HBrO_2$$

$$HBrO_2 + H^+ + Br^- \rightarrow 2HOBr$$

The Belousov-Zhabotinsky reaction yields a spatio-temporal moving pattern.

http://www.scholarpedia.org/article/Belousov-Zhabotinsky_reaction

Belousov-Zhabotinsky-Reaction

http://www.uni-muenster.de/Physik.AP/Purwins/DE/BZ-Reaktion-de.tml

Patterns in Mixtures of Granulated Material

A mixture of granulated material with different density consisting of Salt (white) and Poppy Seed (black) can establish a spatial structure.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/Streifen_im_Salz-Mohn-Gemisch-de.html

Patterns in Mixtures of Granulated Material

A mixture of granulated material with different density consisting of Salt (white) and Poppy Seed (black) can establish a spatial structure.

http://www.uni-muenster.de/Physik.AP/Purwins/DE/Streifen_im_Salz-Mohn-Gemisch-de.html

Patterns in Mixtures of Granulated Material

A mixture of granulated material with different density consisting of Salt (white) and Poppy Seed (black) can establish a spatial structure.

The nonlinear combination of different densities, the movement and the friction causes the effect.

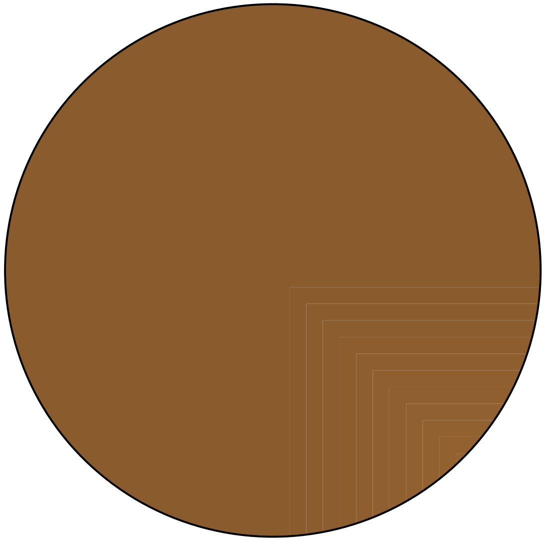
http://www.uni-muenster.de/Physik.AP/Purwins/DE/Streifen_im_Salz-Mohn-Gemisch-de.html

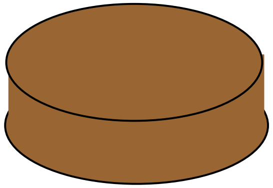
Growth Patterns

Growth patterns of plants are often determined by activator-inhibitor reaction-diffusion systems.

Growth Patterns

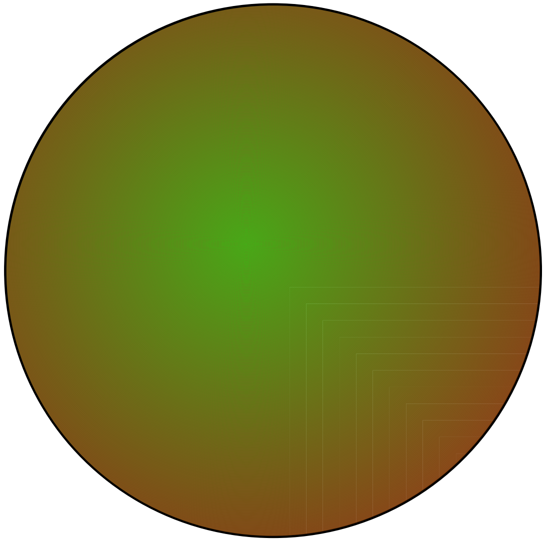
Growth patterns of plants are often determined by activator-inhibitor reaction-diffusion systems.

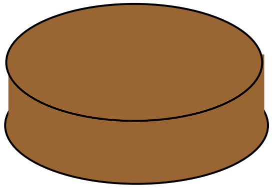




Growth Patterns

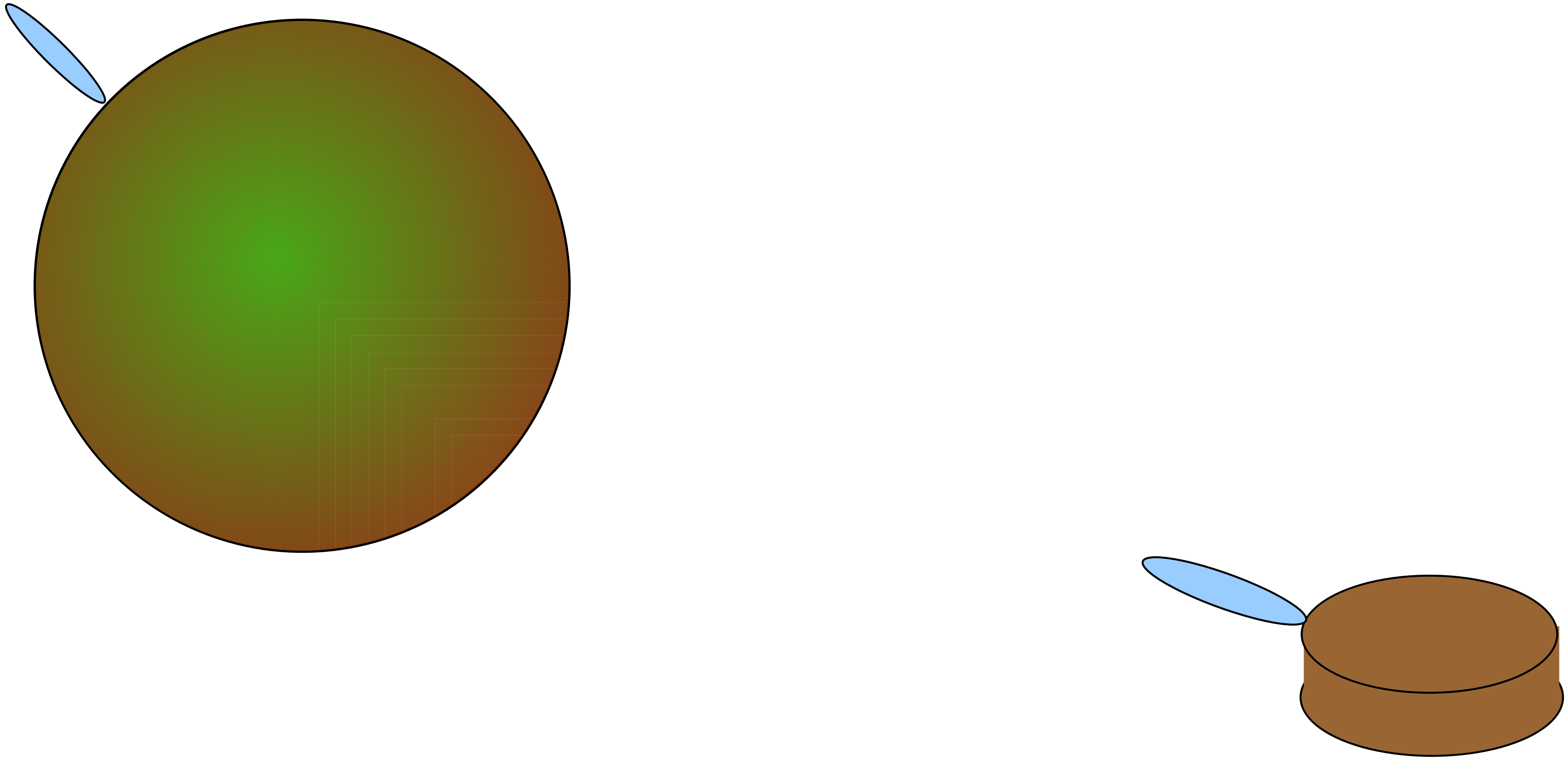
The growing plant produces resources (activator)





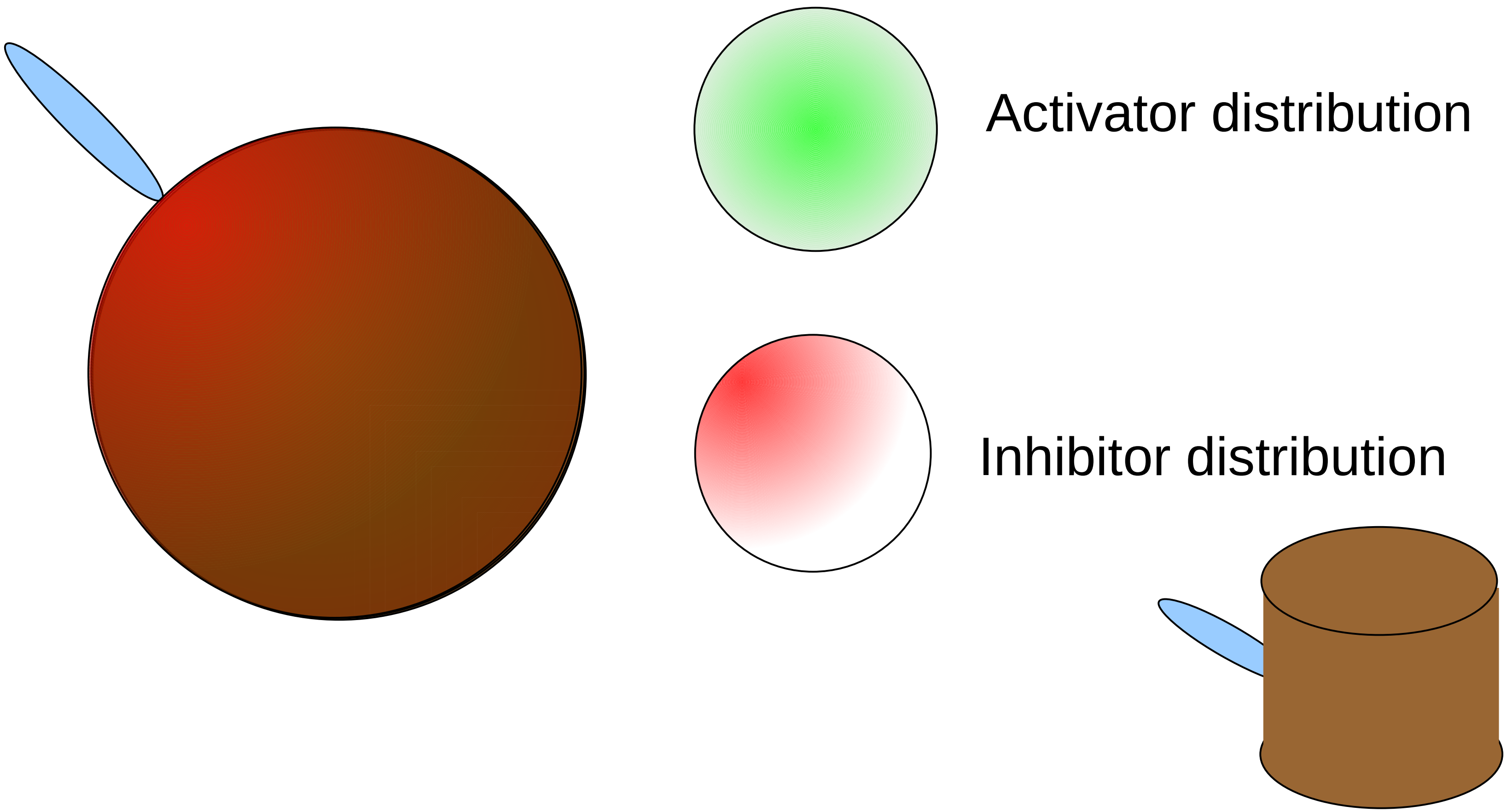
Growth Patterns

As soon as enough resources (activator) are available,
a leaf is growing in that position.



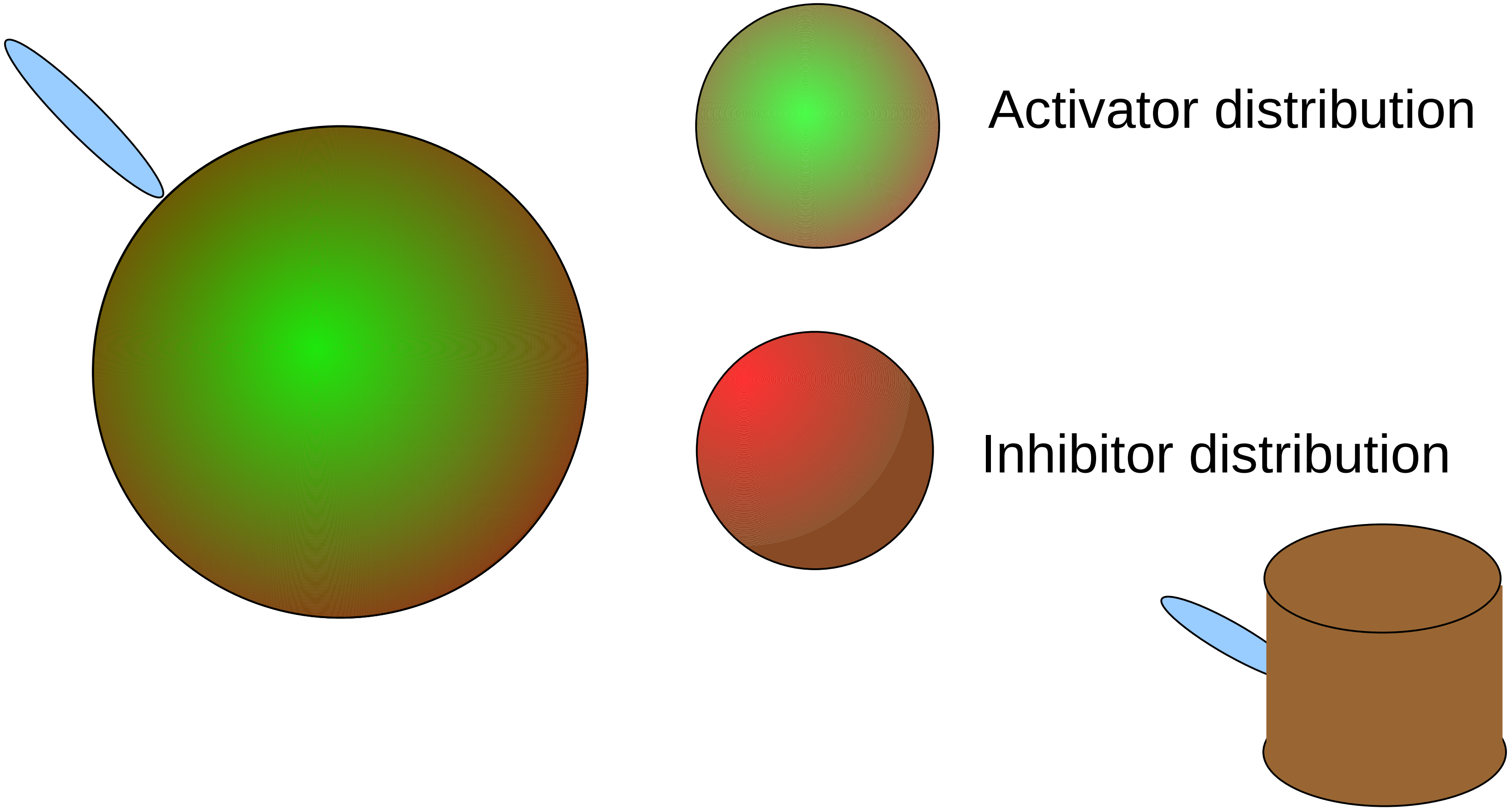
Growth Patterns

With the plant growing, the concentration of activator and inhibitor is decaying.



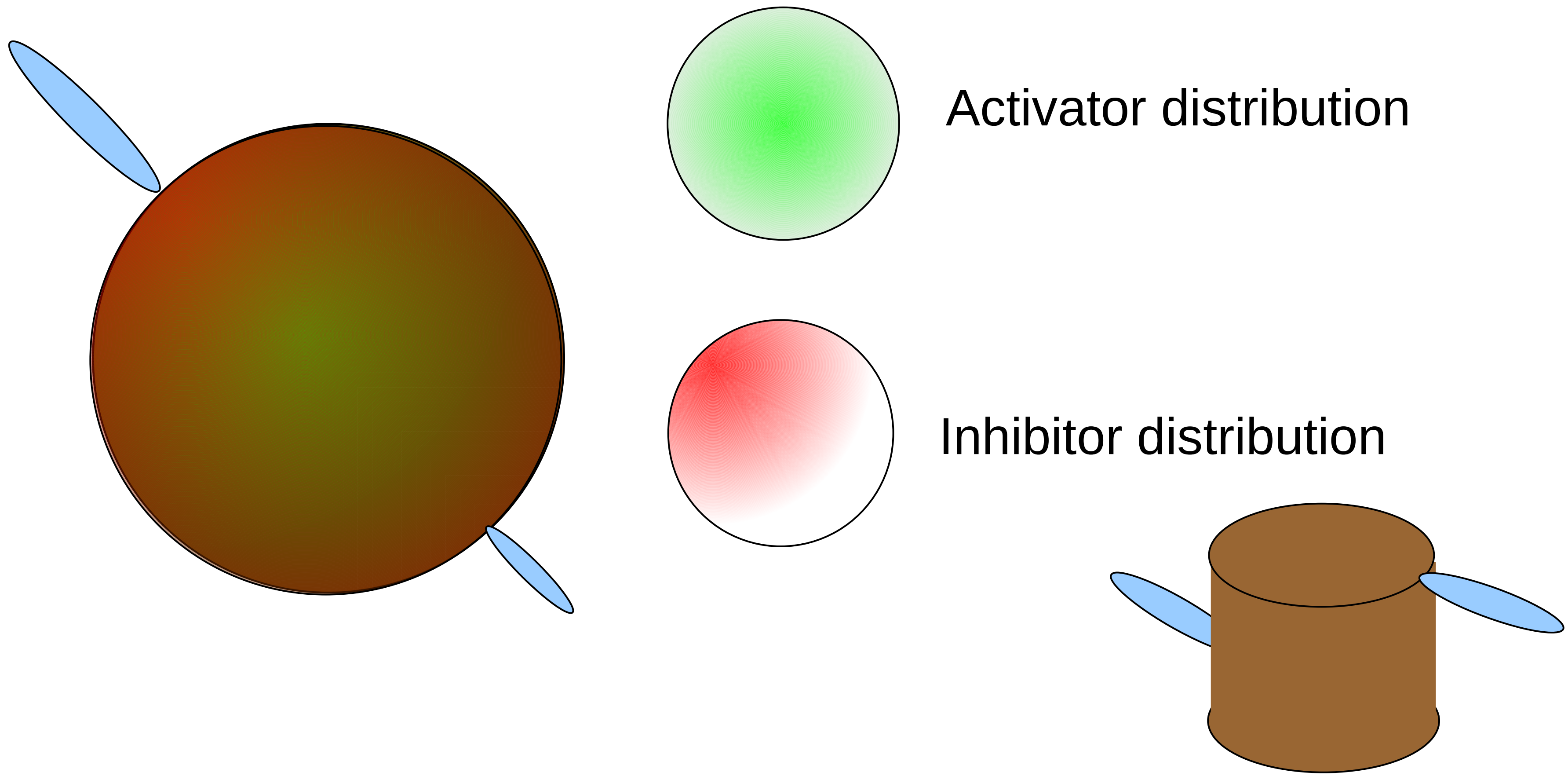
Growth Patterns

After a while, enough resources are available again to produce a new leaf.



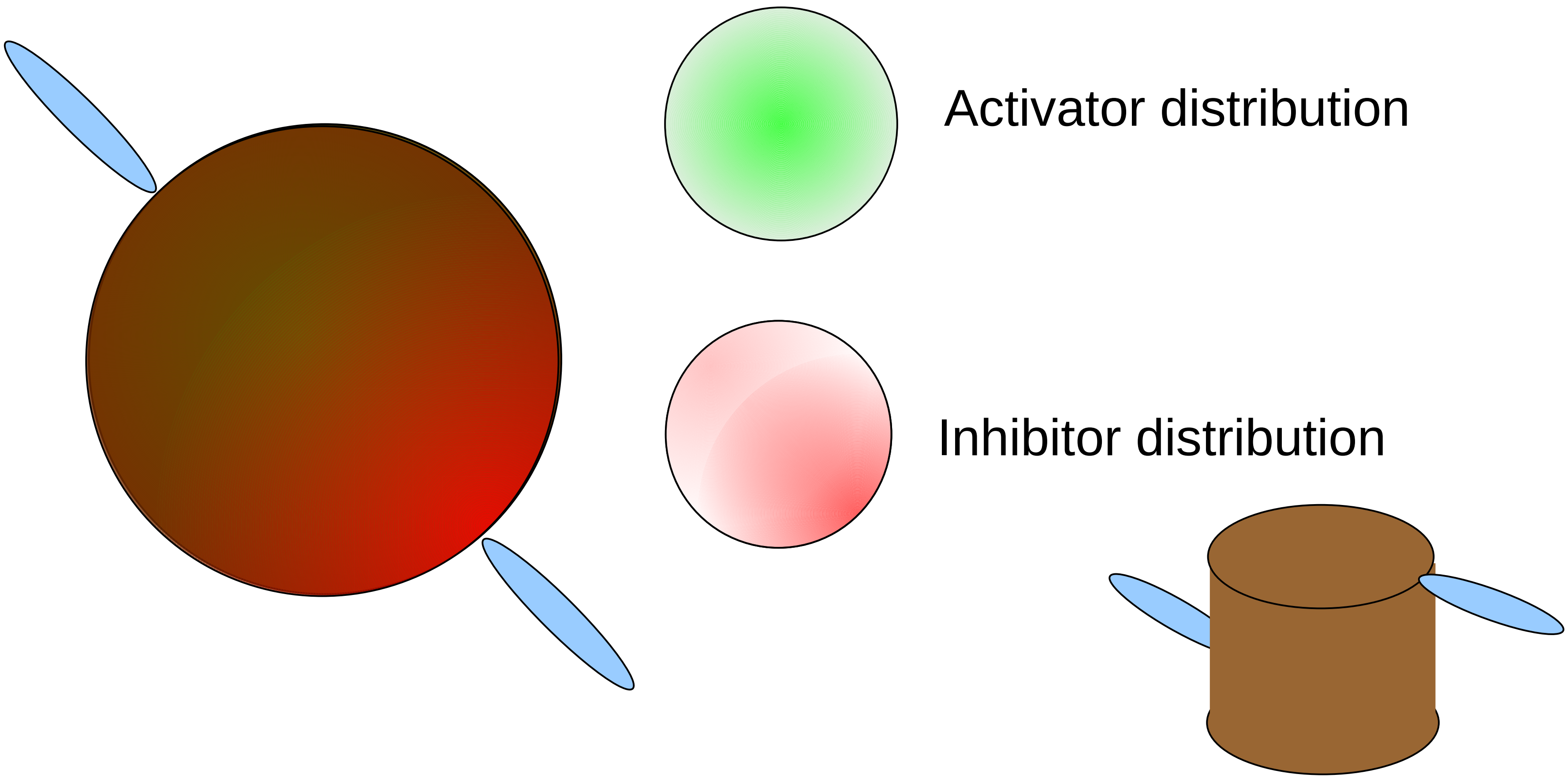
Growth Patterns

But the inhibitor is still present; thus the position of the next leaf will be a far away from the fist one.



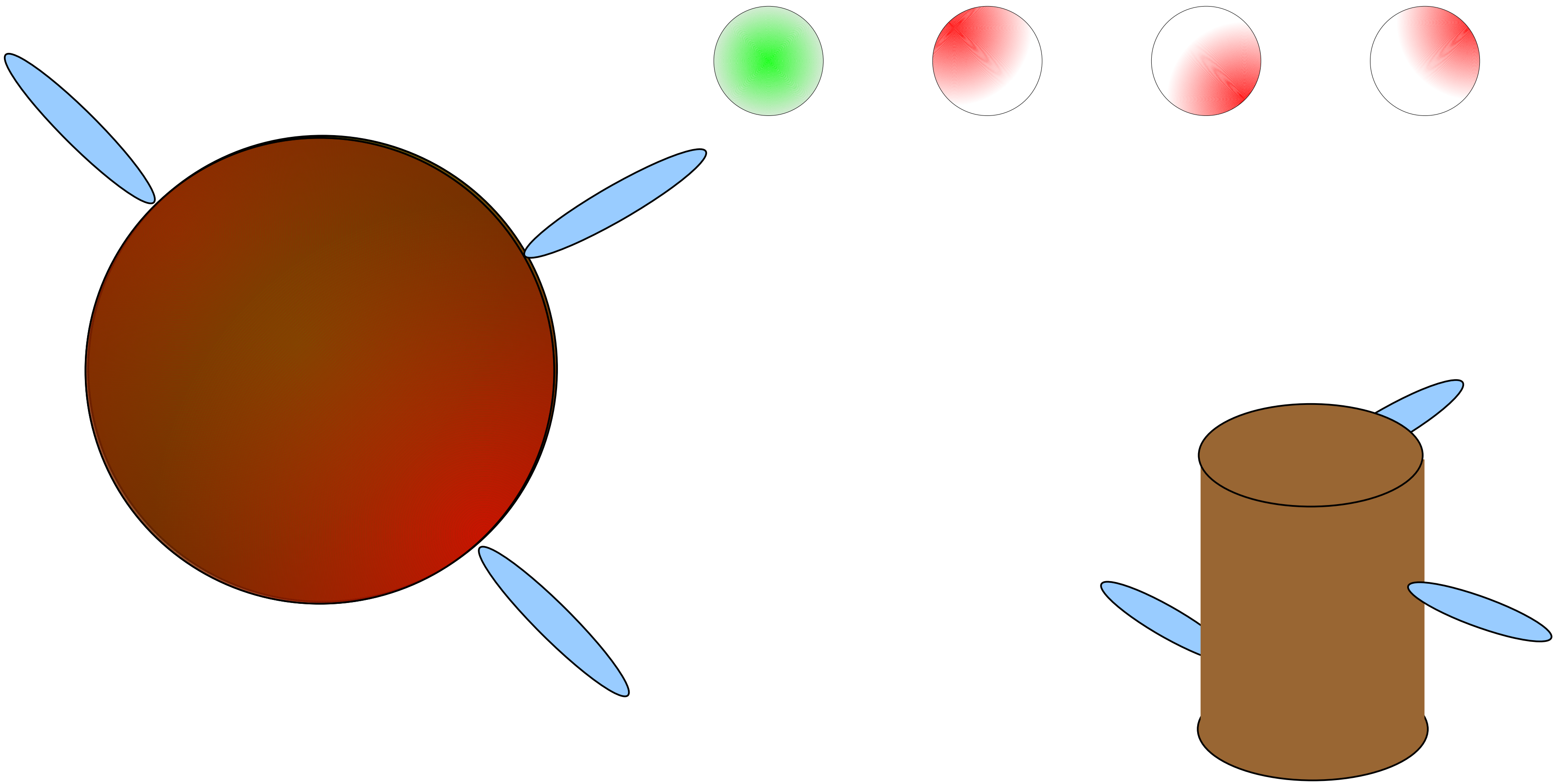
Growth Patterns

Again, the growing leaf will produce inhibitor.



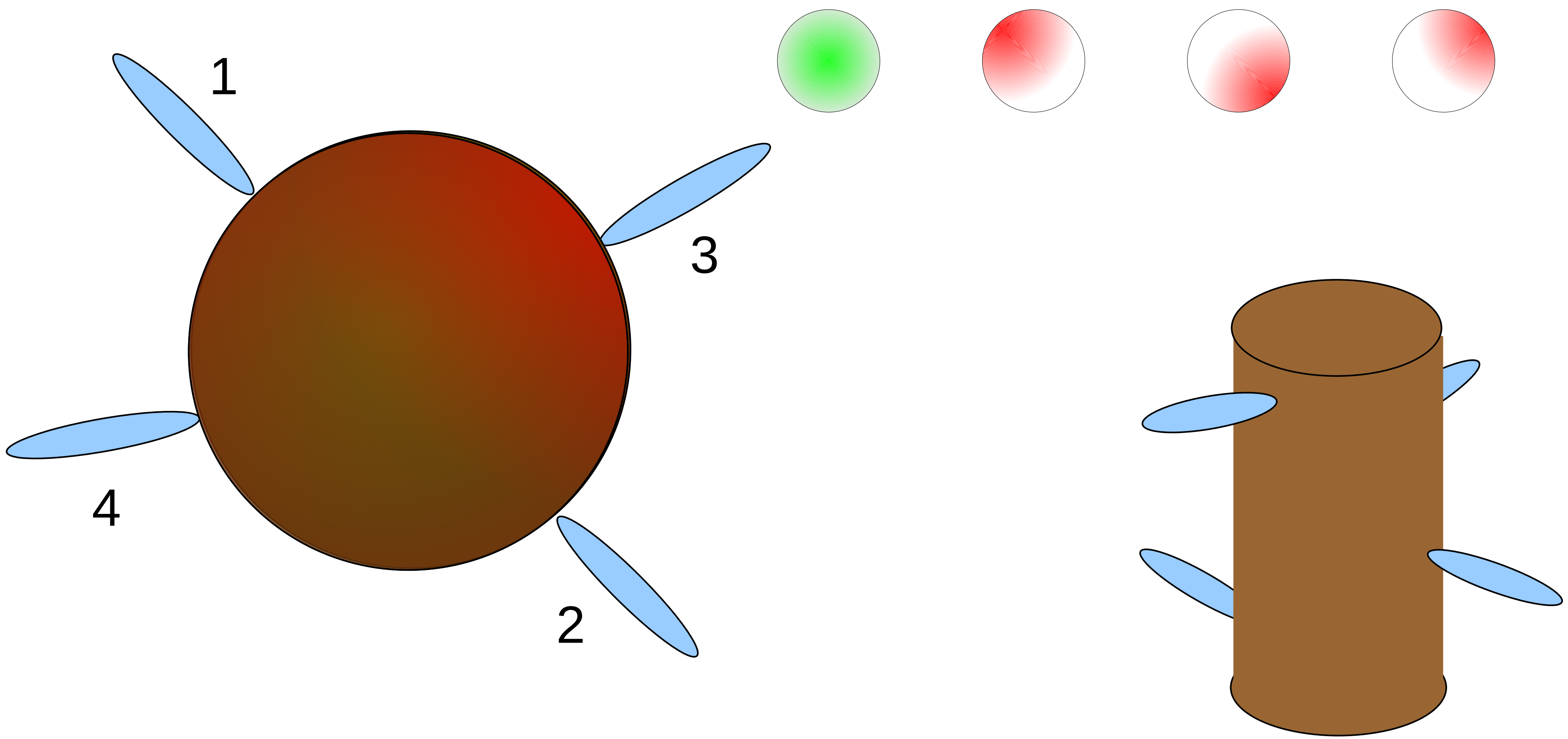
Growth Patterns

The next leaf to grow is influenced by the inhibitor of both existing leaves, leading to a position in between.



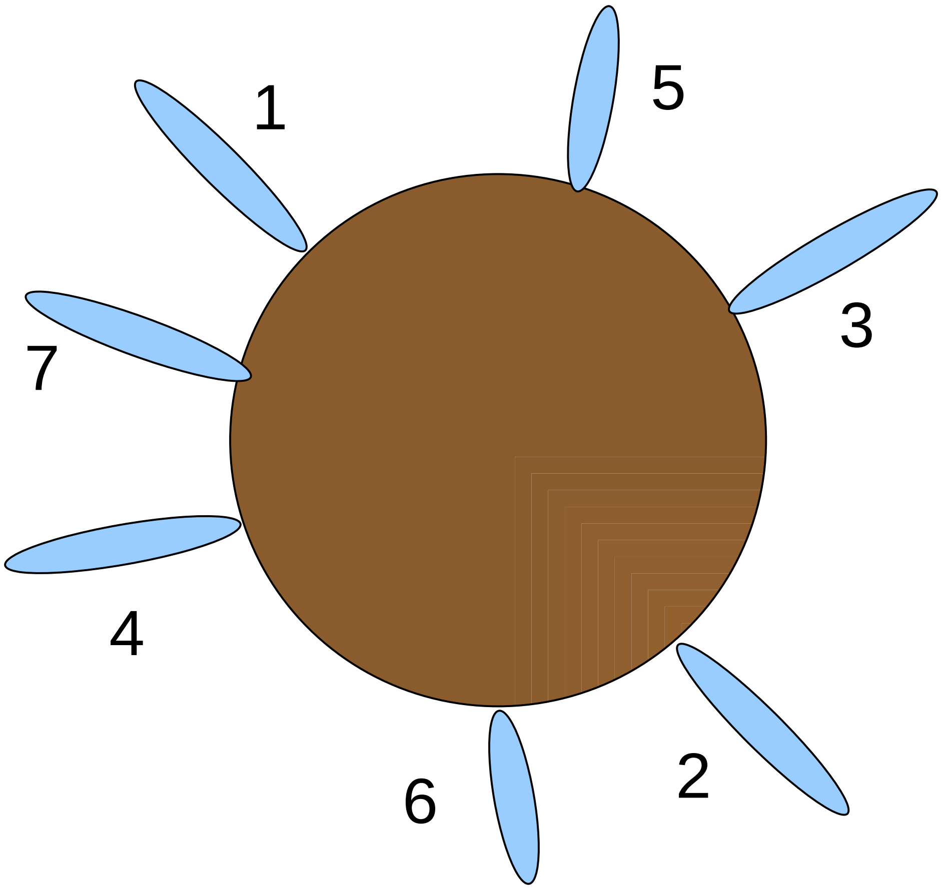
Growth Patterns

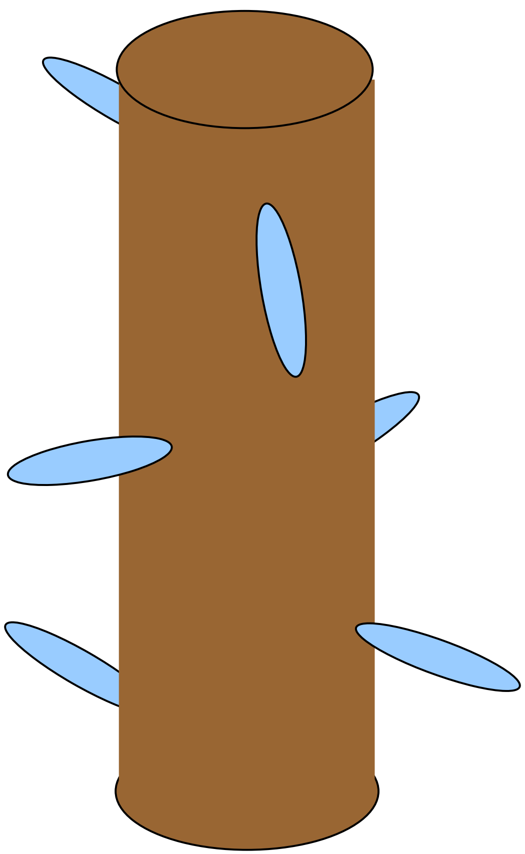
The next leaf to grow is influenced by the inhibitor of all the existing leaves, leading to a positions in between.



Growth Patterns

After a while, the angle between subsequent leafs has become almost the **Golden Angle** of **137.51°** .





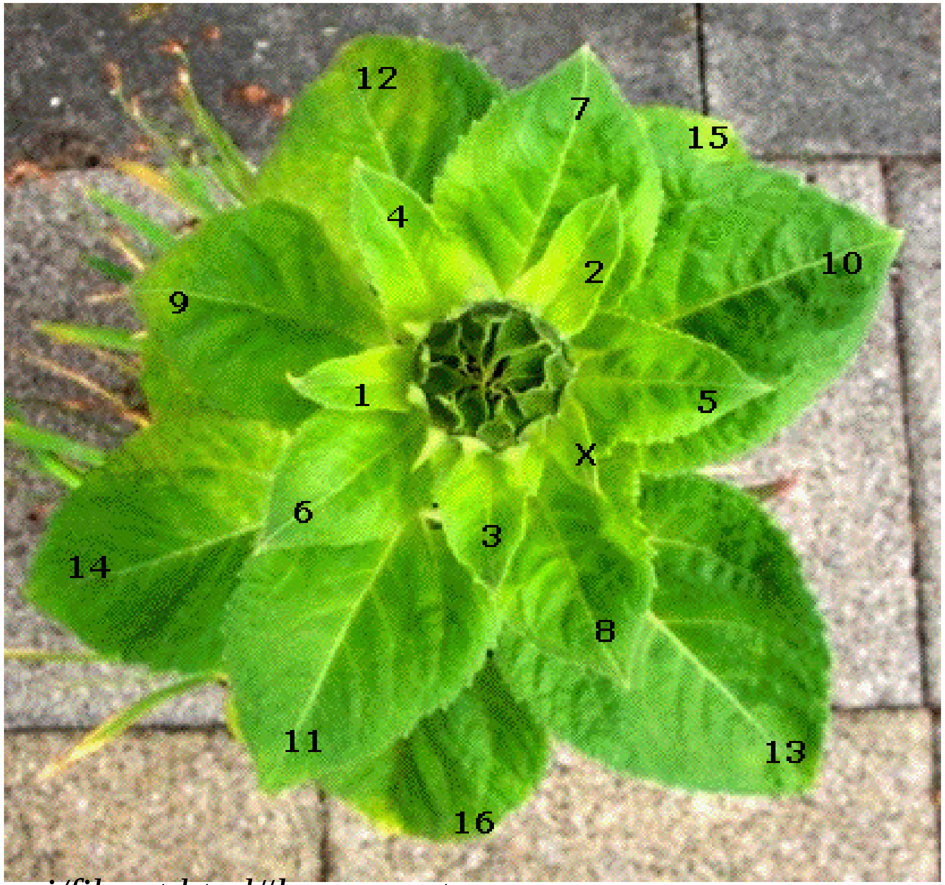
Growth Patterns

Growth patterns of plants are often determined by activator-inhibitor reaction-diffusion systems.

http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibnat.html#leavesperturn

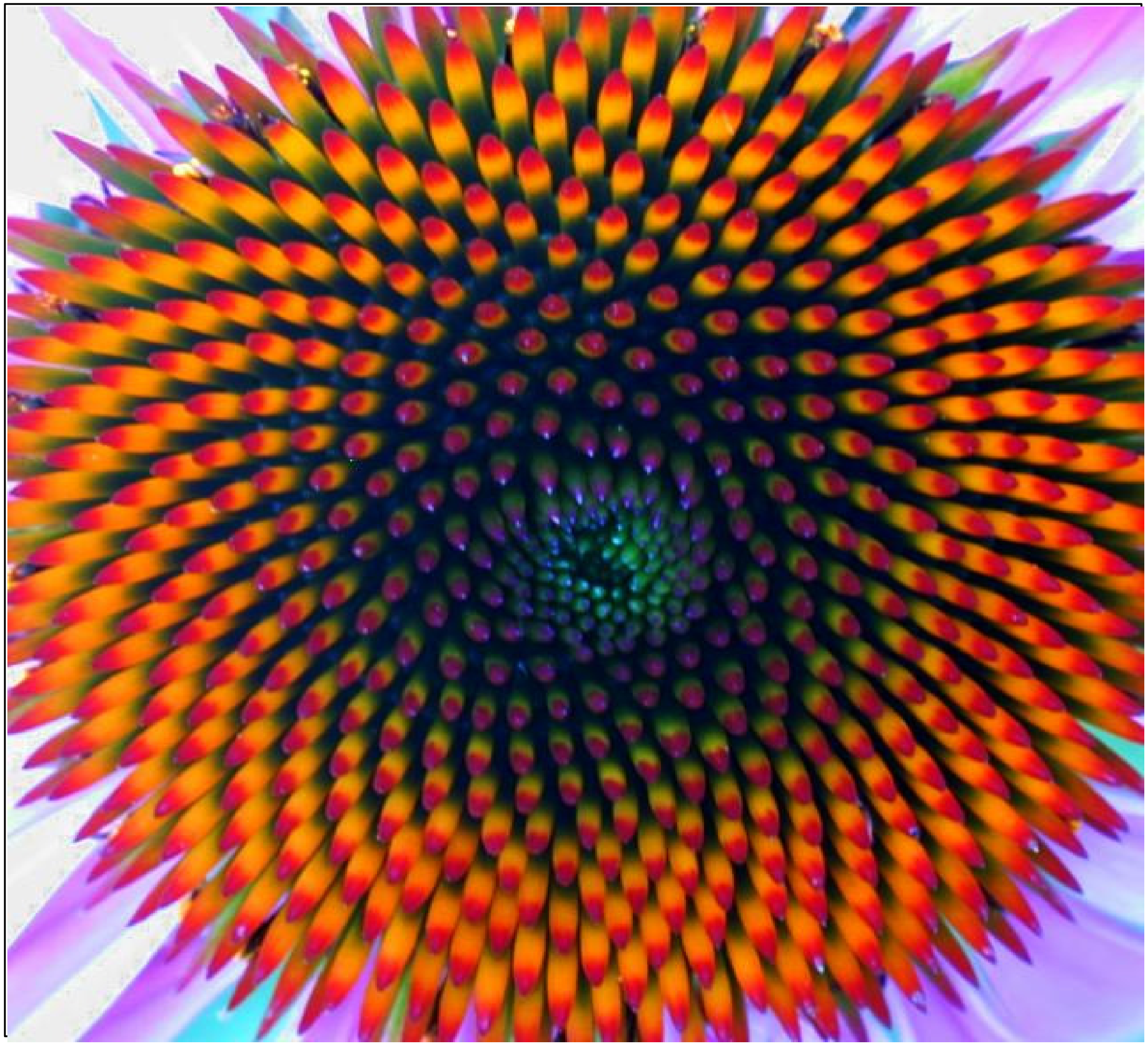
Growth Patterns

Growth patterns of plants are often determined by activator-inhibitor reaction-diffusion systems.



http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibnat.html#leavesperturn

Fibonacci Spirals in Plant Morphogenesis

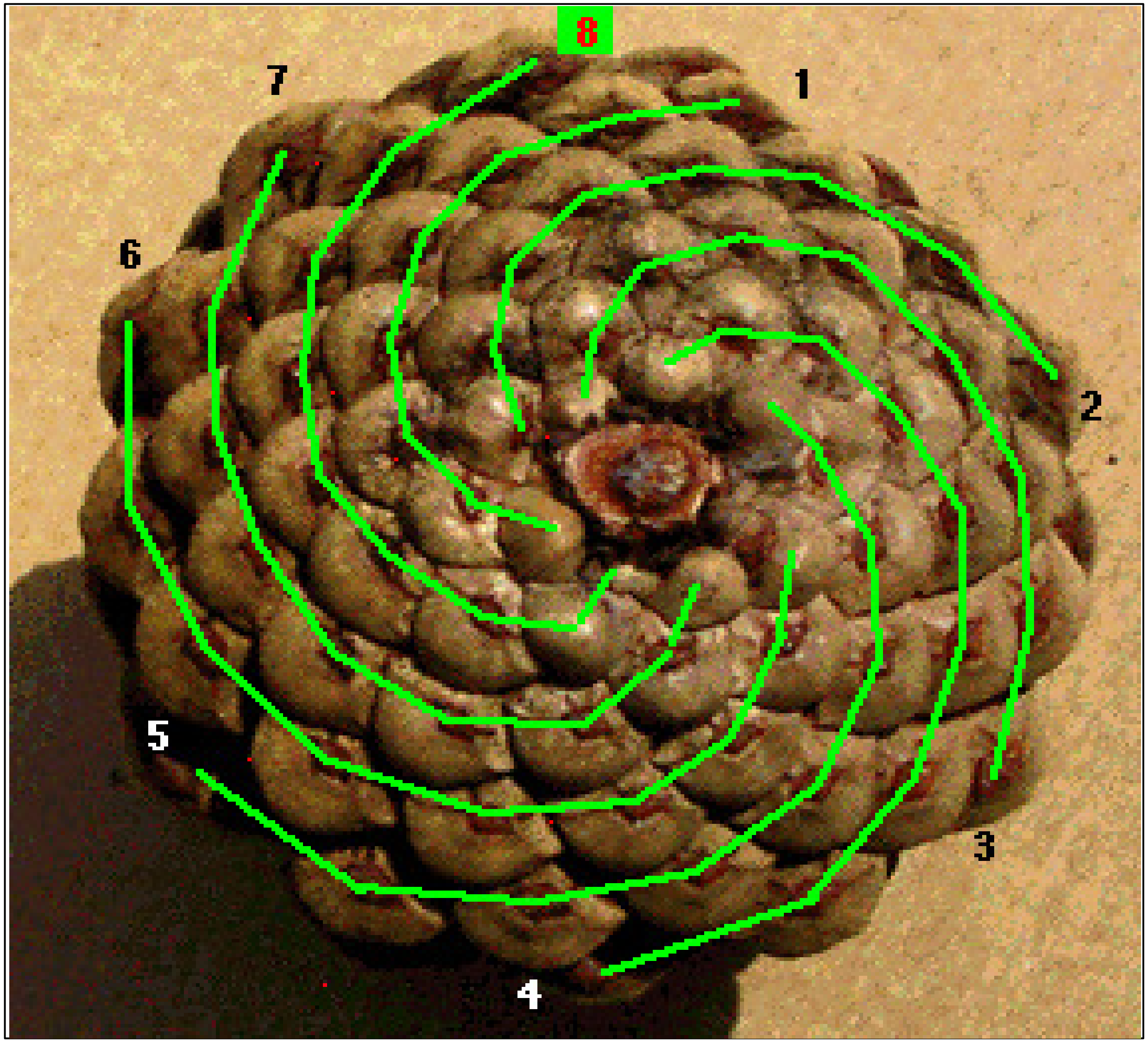


http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

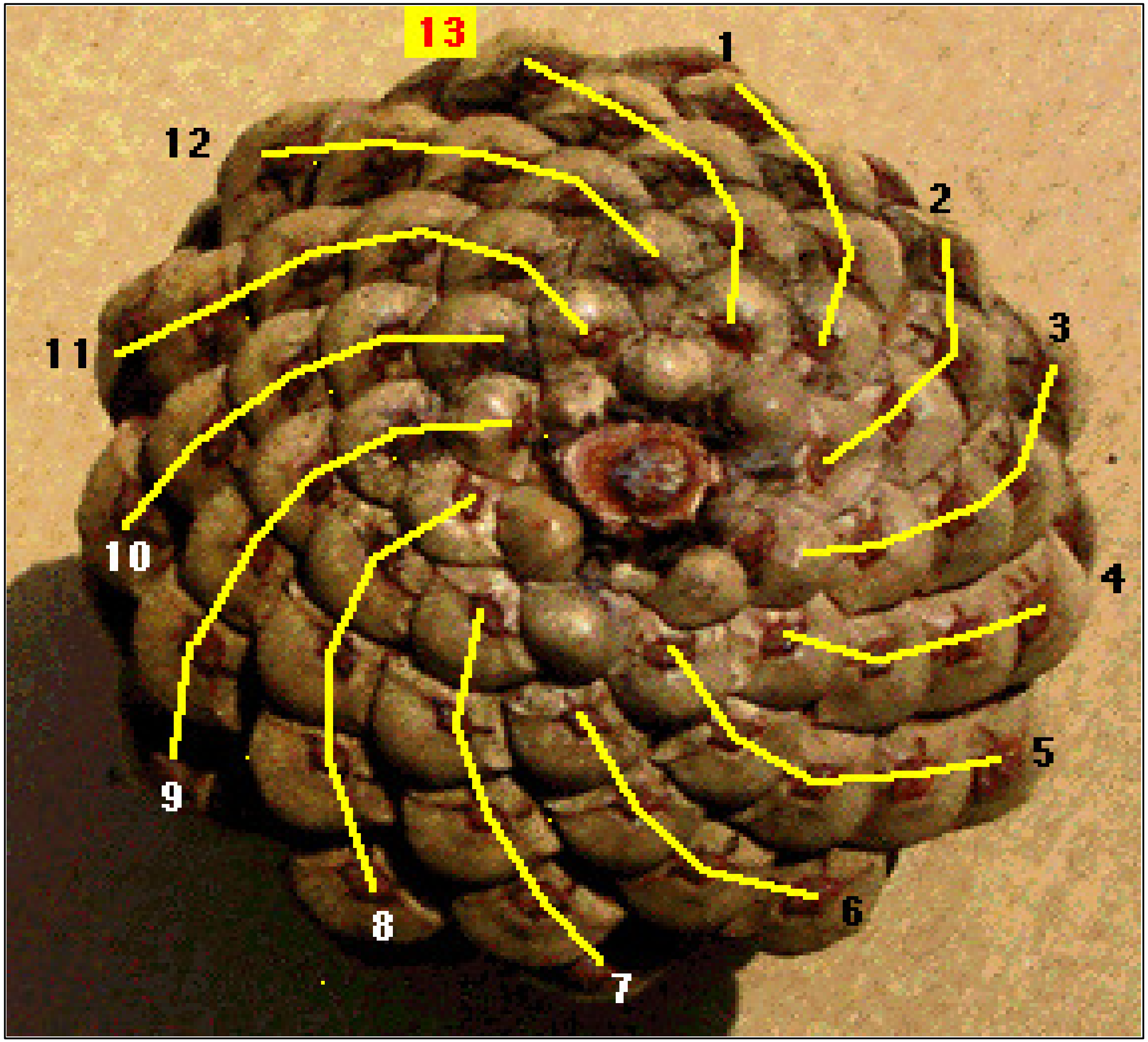


http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis



http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

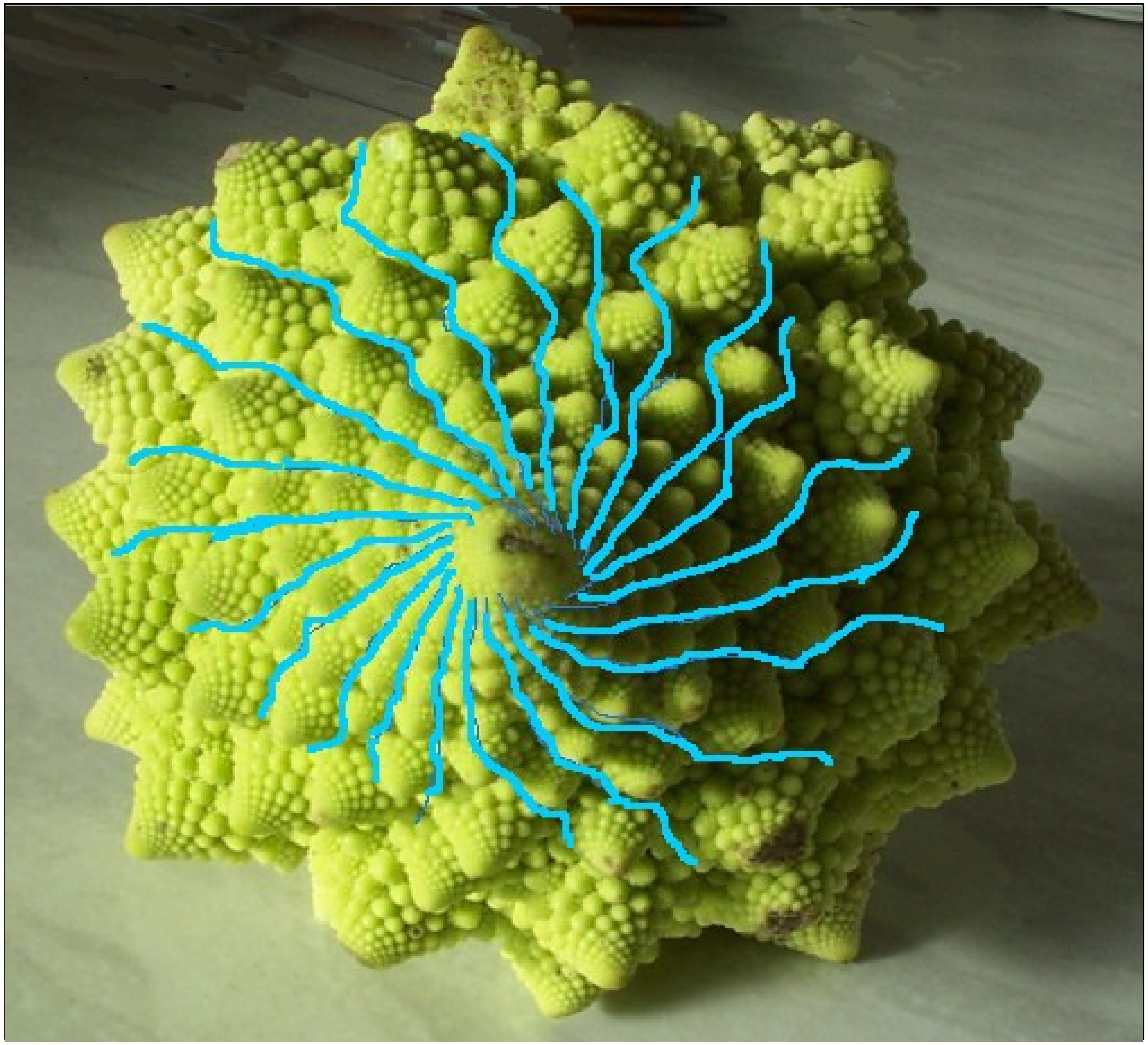
The number of spirals that can be found in either direction are typically consecutive Fibonacci numbers.

From Ron Knott's web pages on Mathematics, Fibonacci Numbers and Nature , 9.6.2009
http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

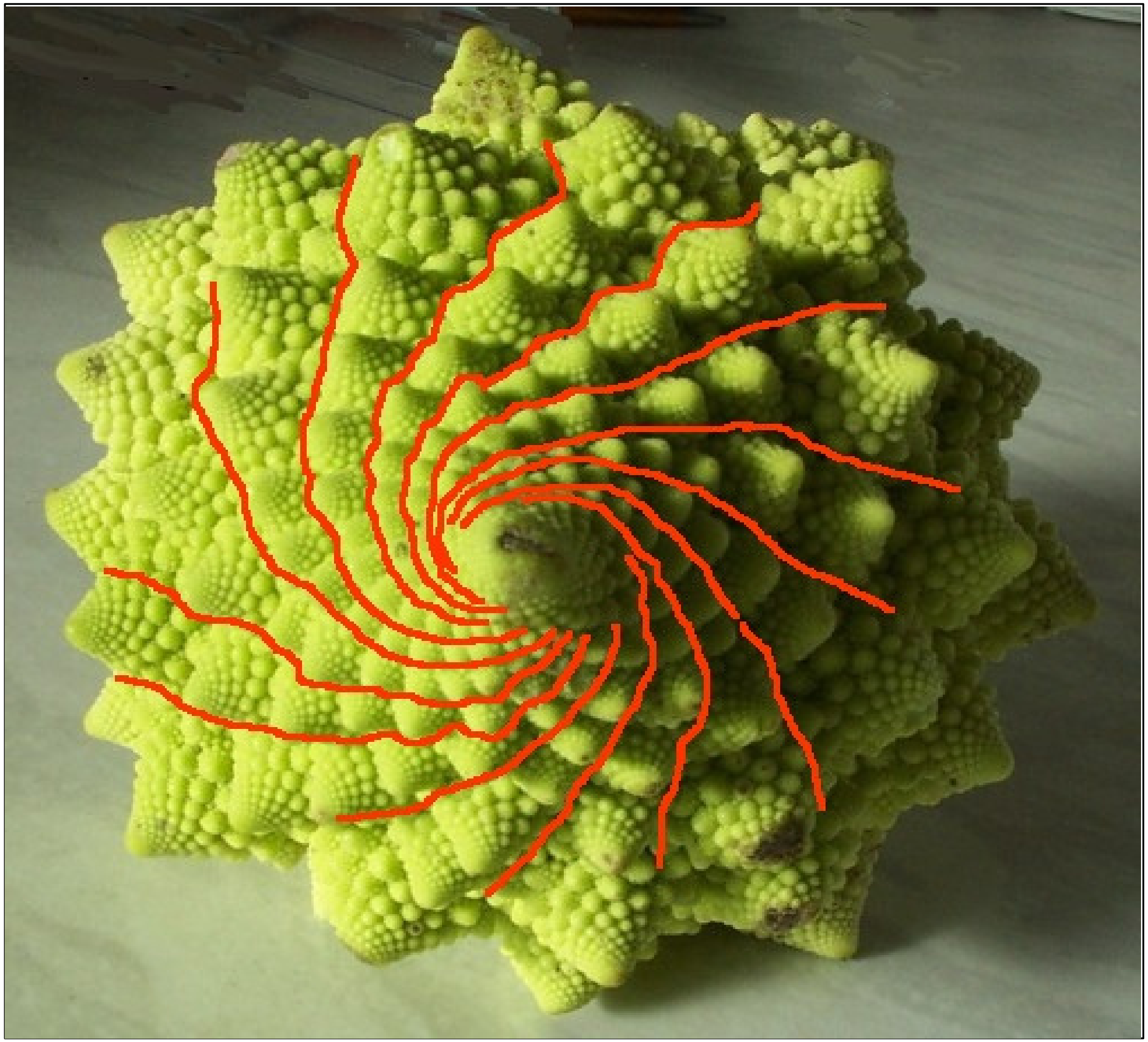
http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis



http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis

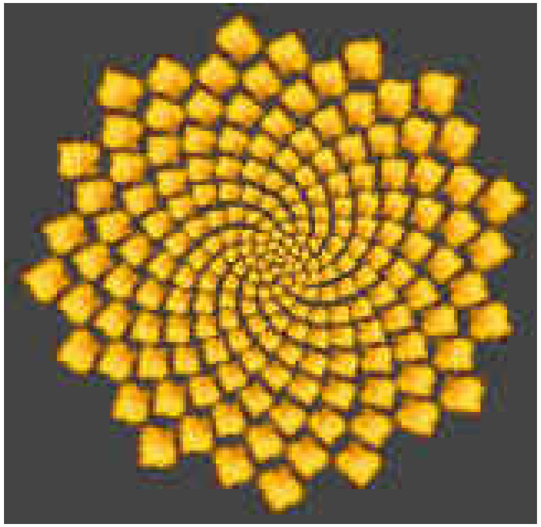


http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

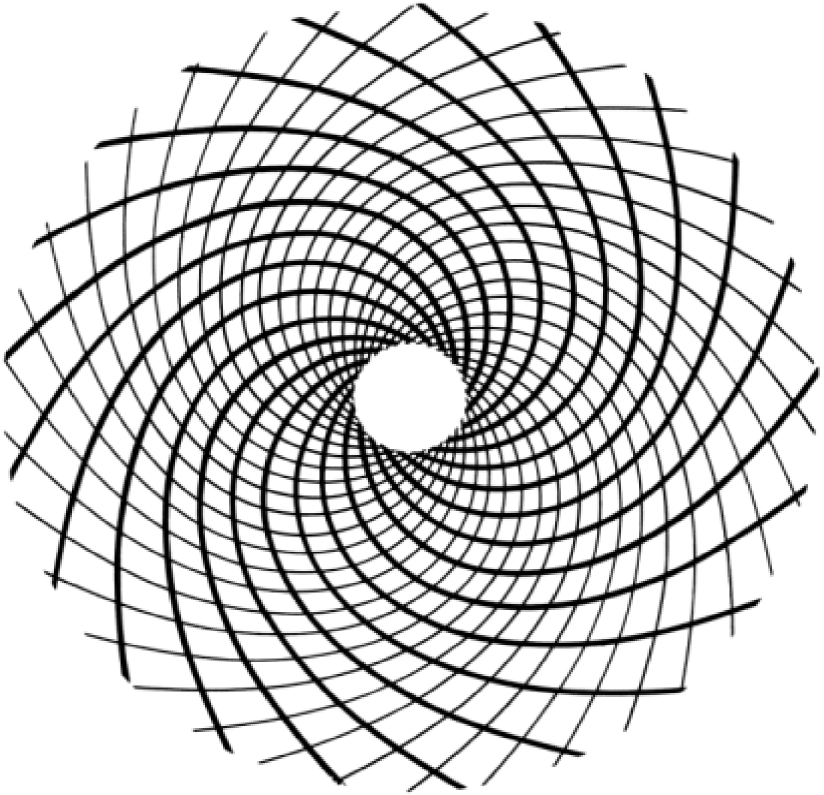
Fibonacci Spirals in Plant Morphogenesis

http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#pinecones

Fibonacci Spirals in Plant Morphogenesis



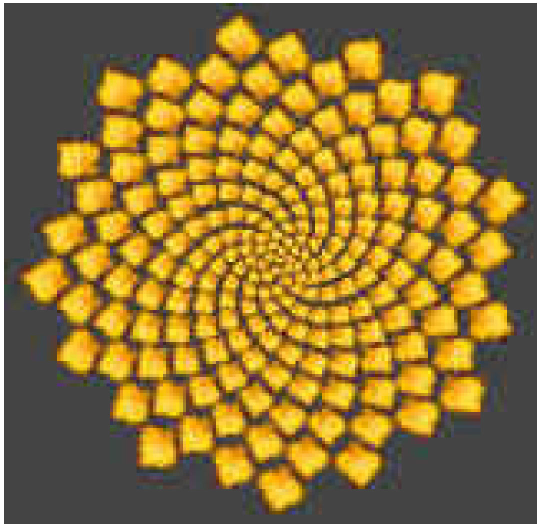
Double Spiral Pattern in the Sunflower
(Helianthus Annuus)

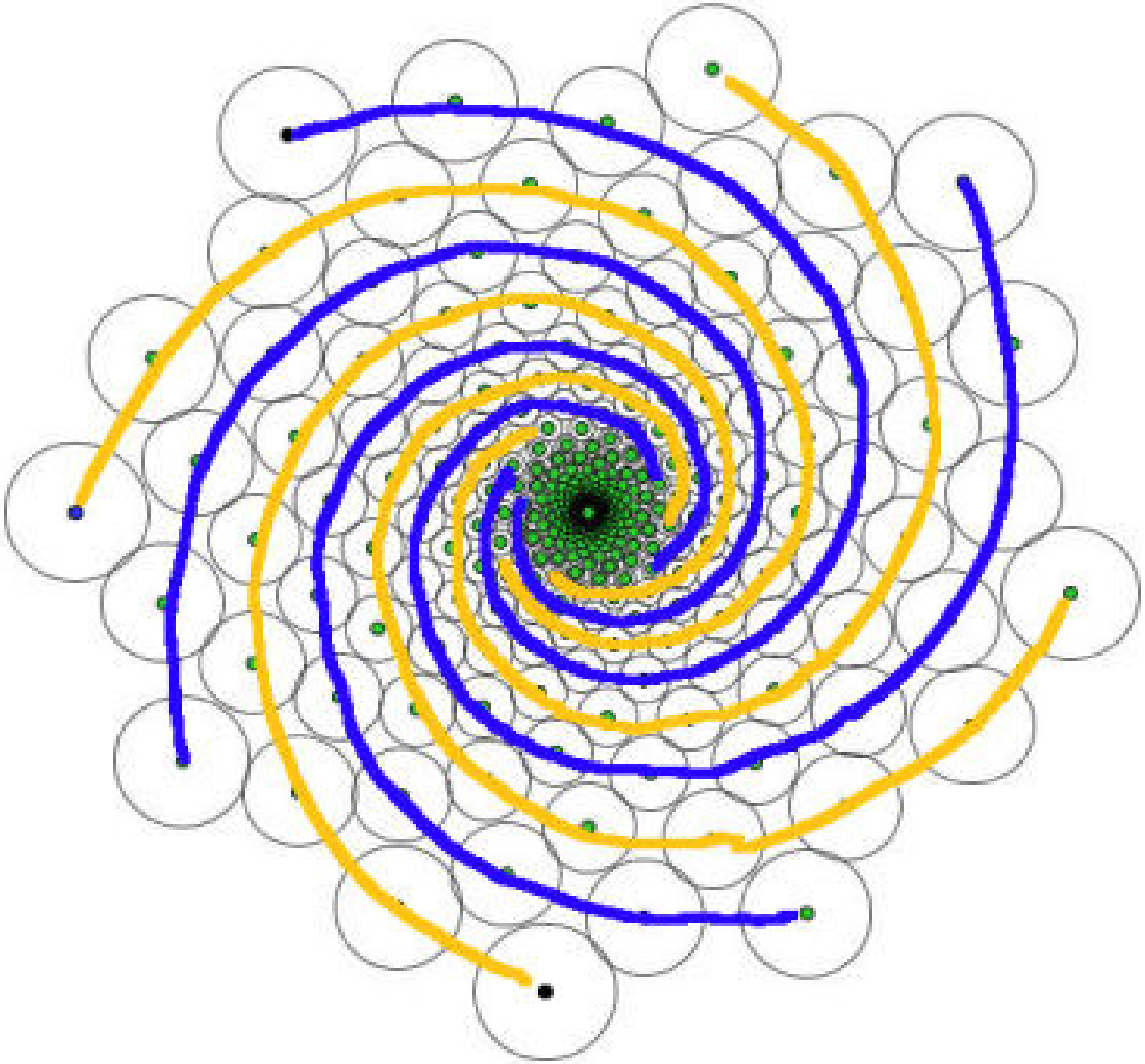


21 Clockwise Spirals and 34 Anticlockwise Spirals
(21 and 34 are Successive Fibonacci Numbers)

$$34 \div 21 \approx \phi$$

Fibonacci Spirals in Plant Morphogenesis





Self Similarity

Some structures that can be found in natural systems and in mathematical systems show a fascinating property.
The overall structure repeats itself:
A fragment of the overall system looks like the total system itself.

This fascinating property is called:

Self Similarity

Self Similarity

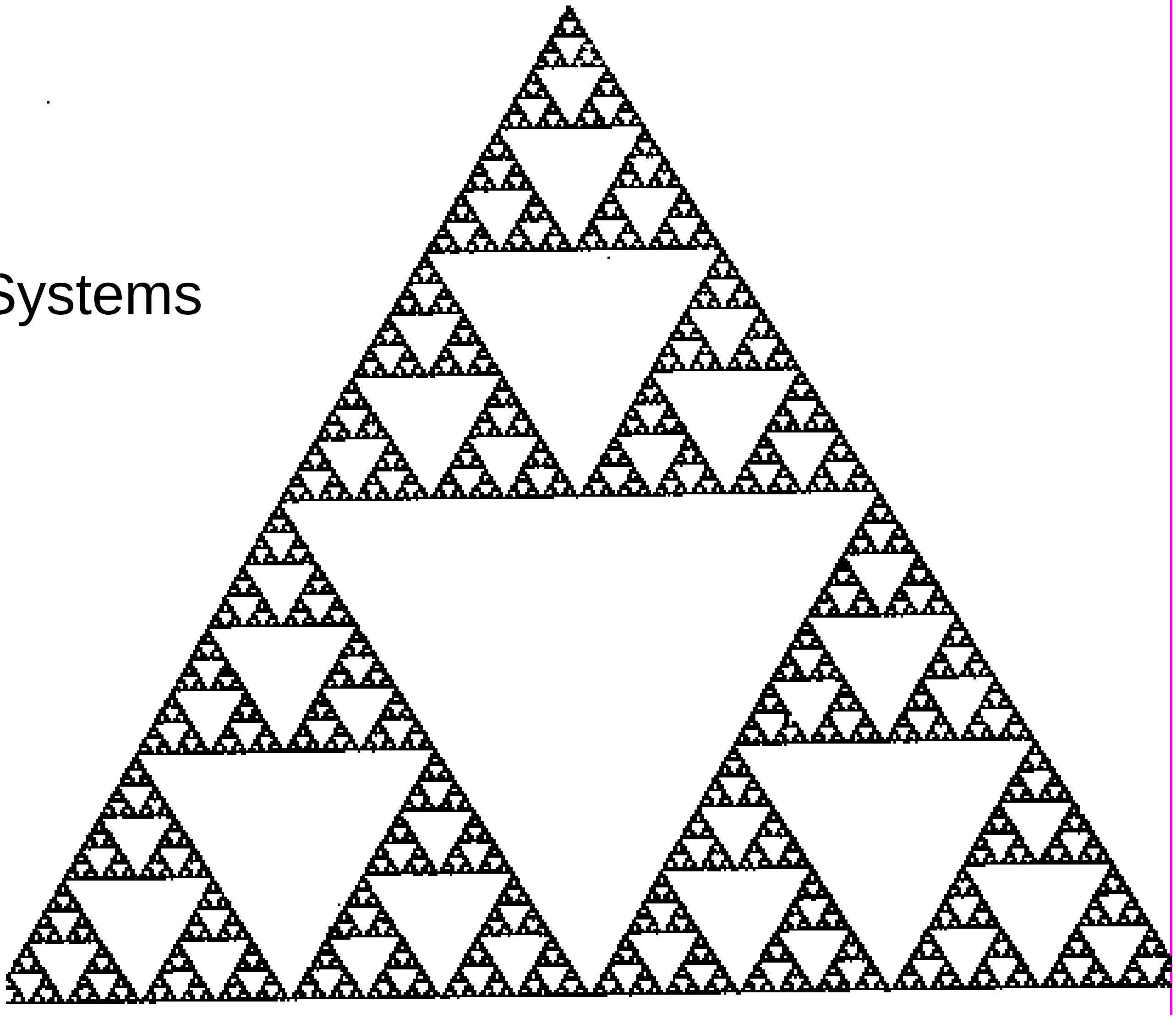
Self Similarity can be found in:

- **Plant Morphogenesis**
- **Lindenmayer Systems**
- **Cellular Automata**
- **Nonlinear Dynamical Systems**
  Deterministic Chaos
- **Iterated Functions**

Self Similarity

Self Similarity can be found in:

- Plant Morphogenesis
- Lindenmayer Systems
- **Cellular Automata**
- Nonlinear Dynamical Systems
  Deterministic Chaos
- Iterated Functions



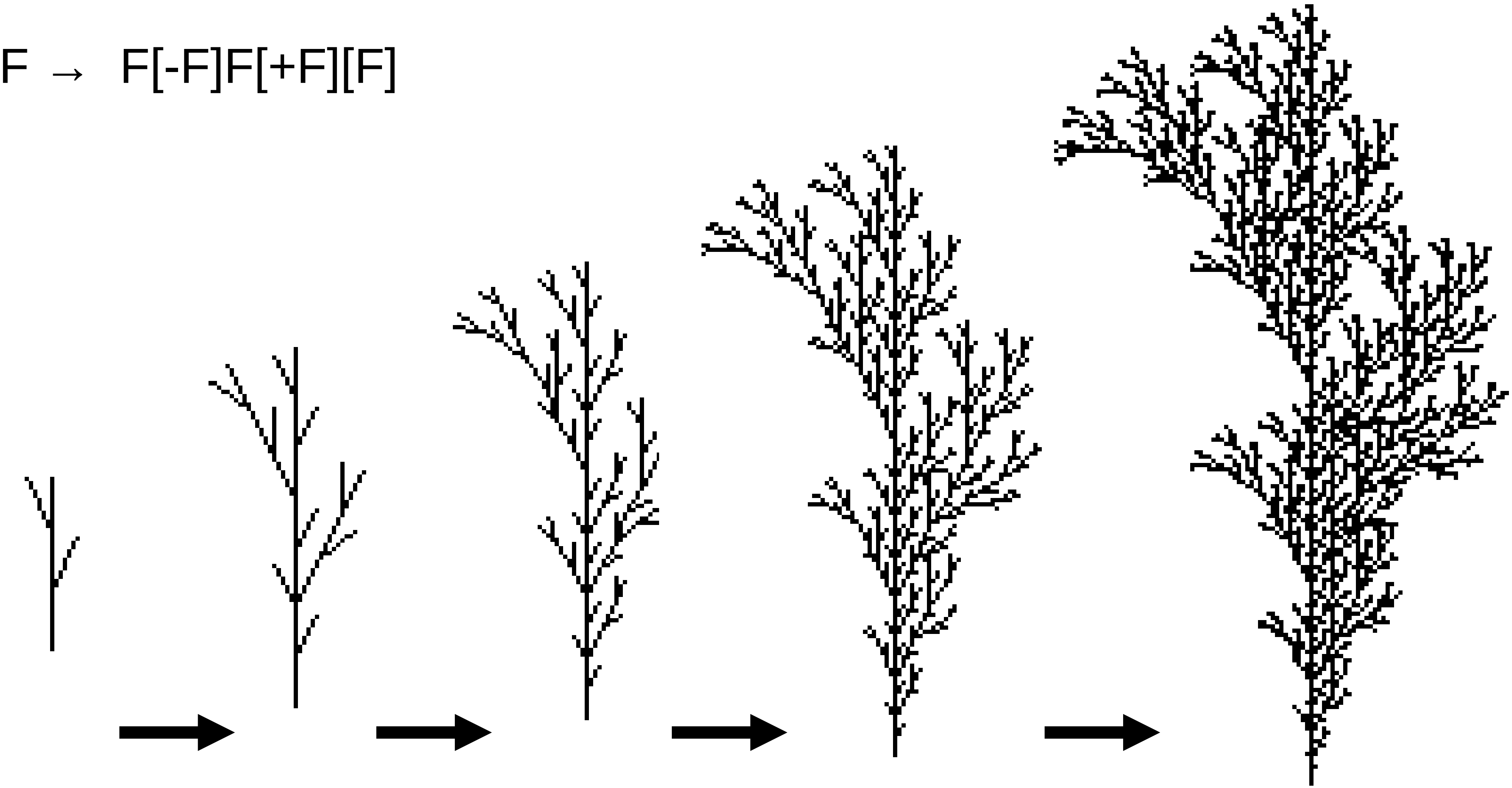
Self Similarity

Self Similarity can be found in:

- Plant Morphogenesis
- **Lindenmayer Systems**
- Cellular Automata
- Nonlinear Dynamical Systems
  Deterministic Chaos
- Iterated Functions

Lindenmayer Systems: Examples

F → F[-F]F[+F][F]

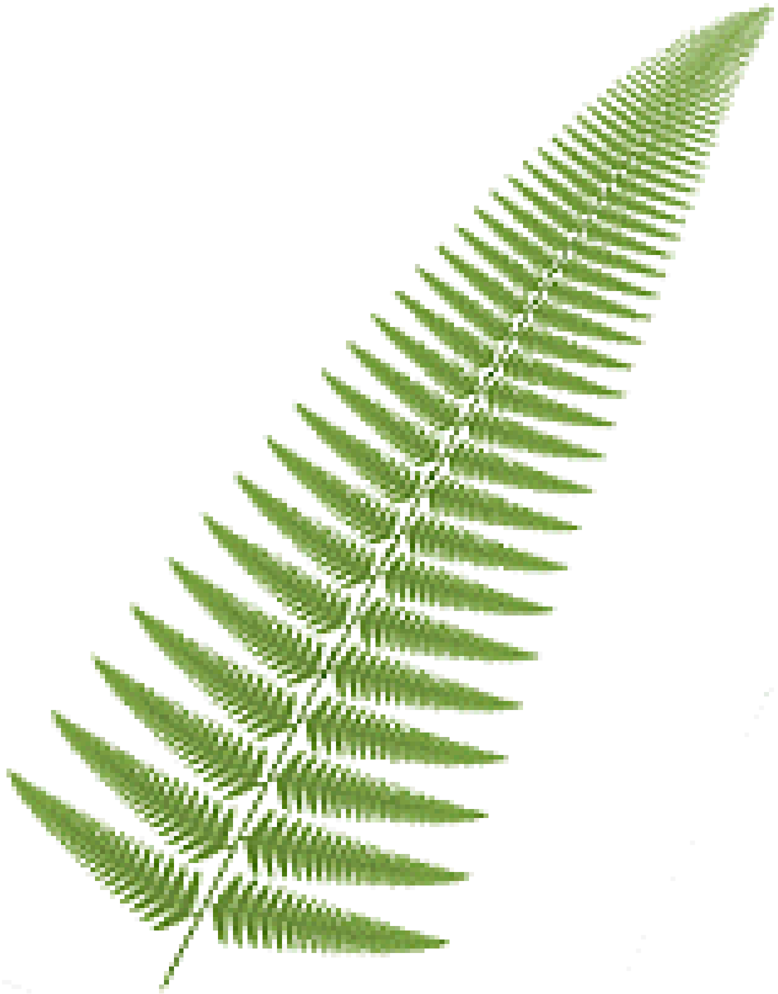


From:*http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html*

Self Similarity

Self Similarity can be found in:

- Plant Morphogenesis
- **Lindenmayer Systems**
- Cellular Automata
- Nonlinear Dynamical Systems
  Deterministic Chaos
- Iterated Functions



Self Similarity

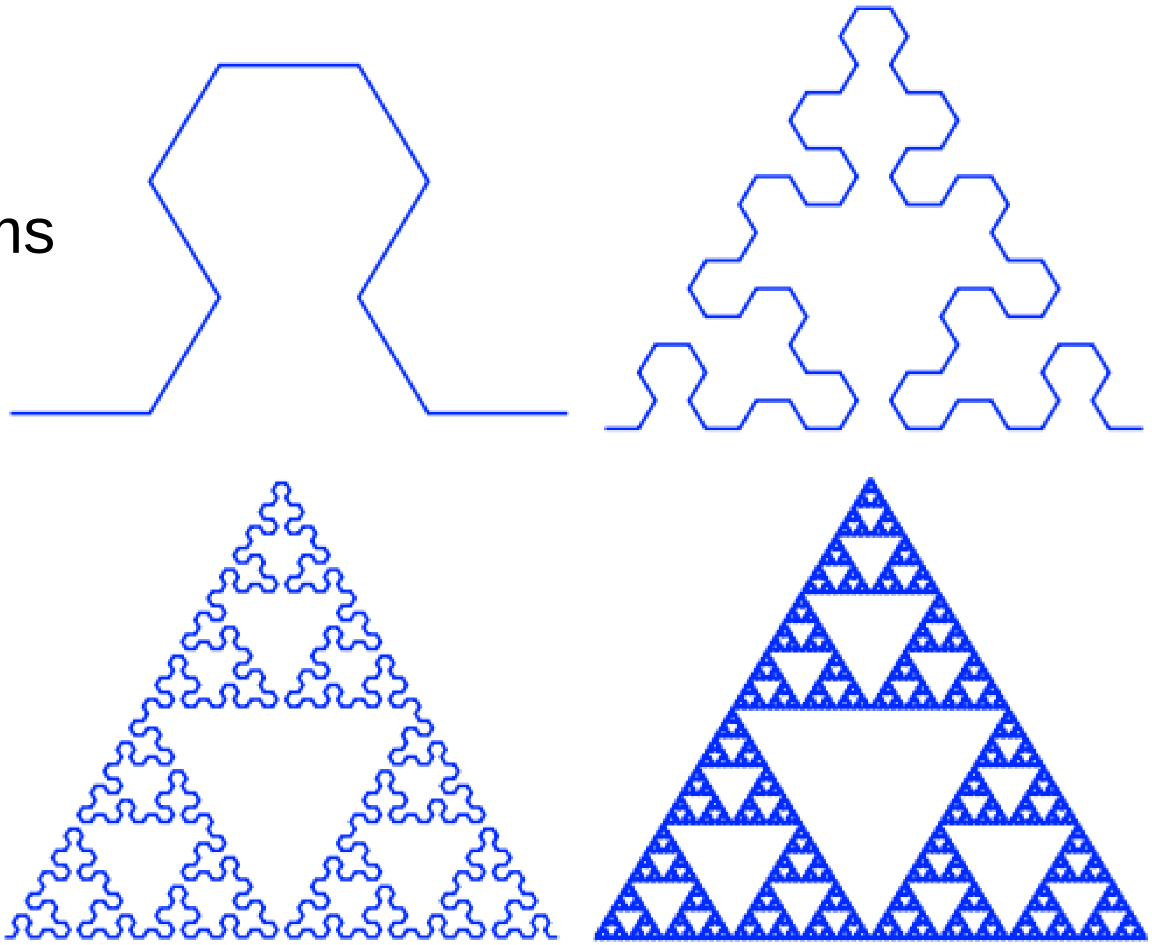
Self Similarity can be found in:

- Plant Morphogenesis
- **Lindenmayer Systems**
- Cellular Automata
- Nonlinear Dynamical Systems
  Deterministic Chaos
- Iterated Functions

Self Similarity

Self Similarity can be found in:

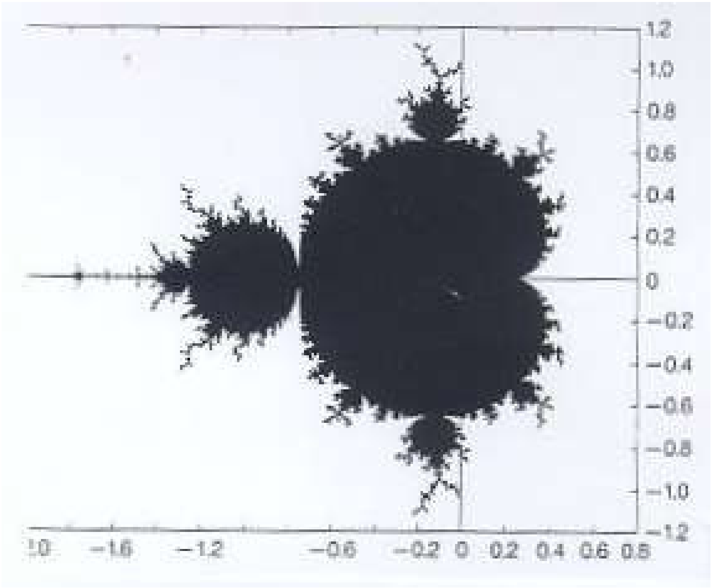
- Plant Morphogenesis
- **Lindenmayer Systems**
- Cellular Automata
- Nonlinear Dynamical Systems
  Deterministic Chaos
- Iterated Functions

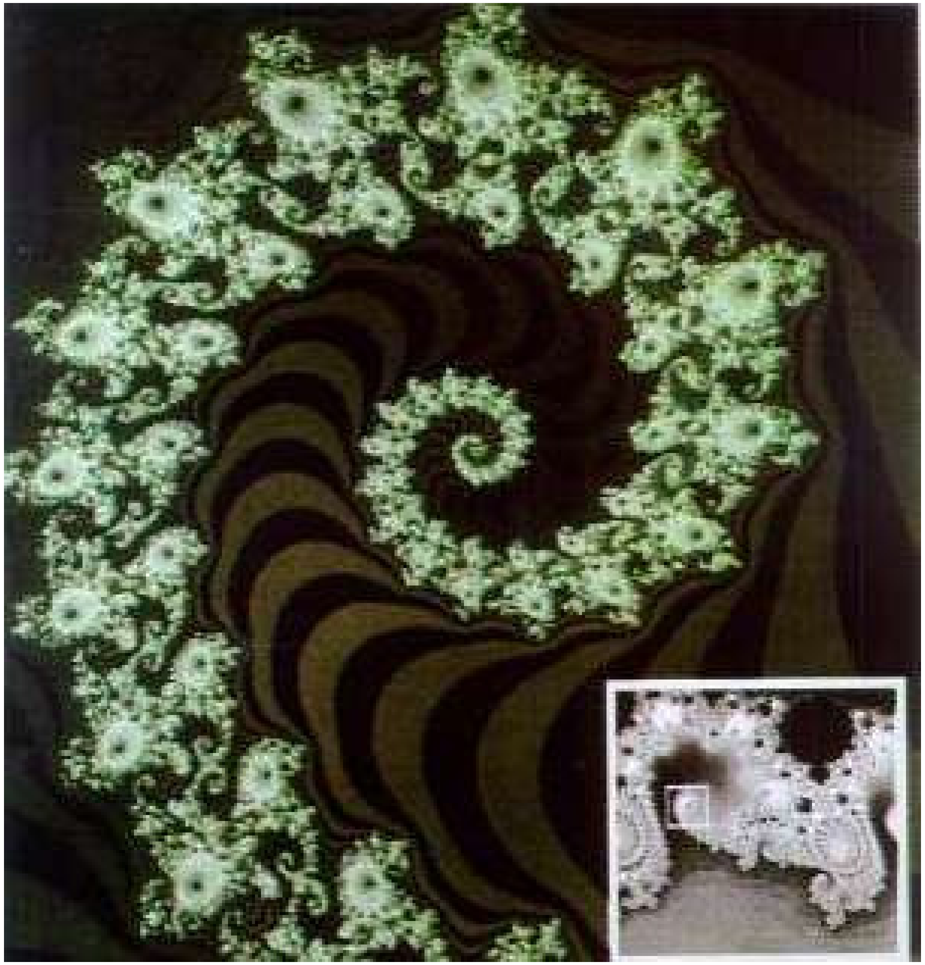


Self Similarity

Self Similarity can be found in:

- Plant Morphogenesis
- Lindenmayer Systems
- Cellular Automata
- Nonlinear Dynamical Systems
  Deterministic Chaos
- **Iterated Functions**





Self Similarity

Self Similarity can be found in:

- **Plant Morphogenesis**
- **Lindenmayer Systems**
- **Cellular Automata**
- **Nonlinear Dynamical Systems**
  Deterministic Chaos
- **Iterated Functions**
-

Self Similarity in Plant Morphogenesis

Brassica oleracea (Romanescu)
© Biopix.dk: N Sloth

Self Similarity in Plant Morphogenesis

Brassica oleracea (Romanescu)
© Biopix.dk: N Sloth

Self Similarity in Plant Morphogenesis

Brassica oleracea (Romanescu)
© Biopix.dk: N Sloth

Pattern Formation

- Iterated functions
- Linear and exponential growth
- Fibonacci sequence
- Logistic growth
- Predator-prey system
- Lotka-Volterra equations
- Activator-inhibitor equations
- Reaction-diffusion systems
- Plant morphogenesis, phylotaxis
- Golden section, Golden angle
- Self similarity

Some important dates

Wed 14.5.25: **Dies Academicus** , special talks, no regular teaching

Thu 29.5.25 : **Ascension Day** , no lectures, no exercises, ...

Sun 8.6. - 9.6.25 : **Pentecost, Whitsun, Pfingsten**, Public holiday
Tue 10.6.25 – Fri 13.6.25 : Excursion week, no lectures, exercises, …

Thu 19.6.25 : **Feast of Corpus Christi**, no lectures, exercises, ...

Artificial Life
Summer 2025

Examples for Pattern Formation in Biological Systems

Master Computer Science [MA-INF 4201]
Mon 14c.t. – 15:45, HSZ, HS-2

Dr. Nils Goerke, Autonomous Intelligent Systems,
Department of Computer Science, University of Bonn

Artificial Life
Summer 2025

Examples for Pattern Formation in Biological Systems

Thank you for your kind attention

Dr. Nils Goerke, Autonomous Intelligent Systems,
Department of Computer Science, University of Bonn