Utrecht University

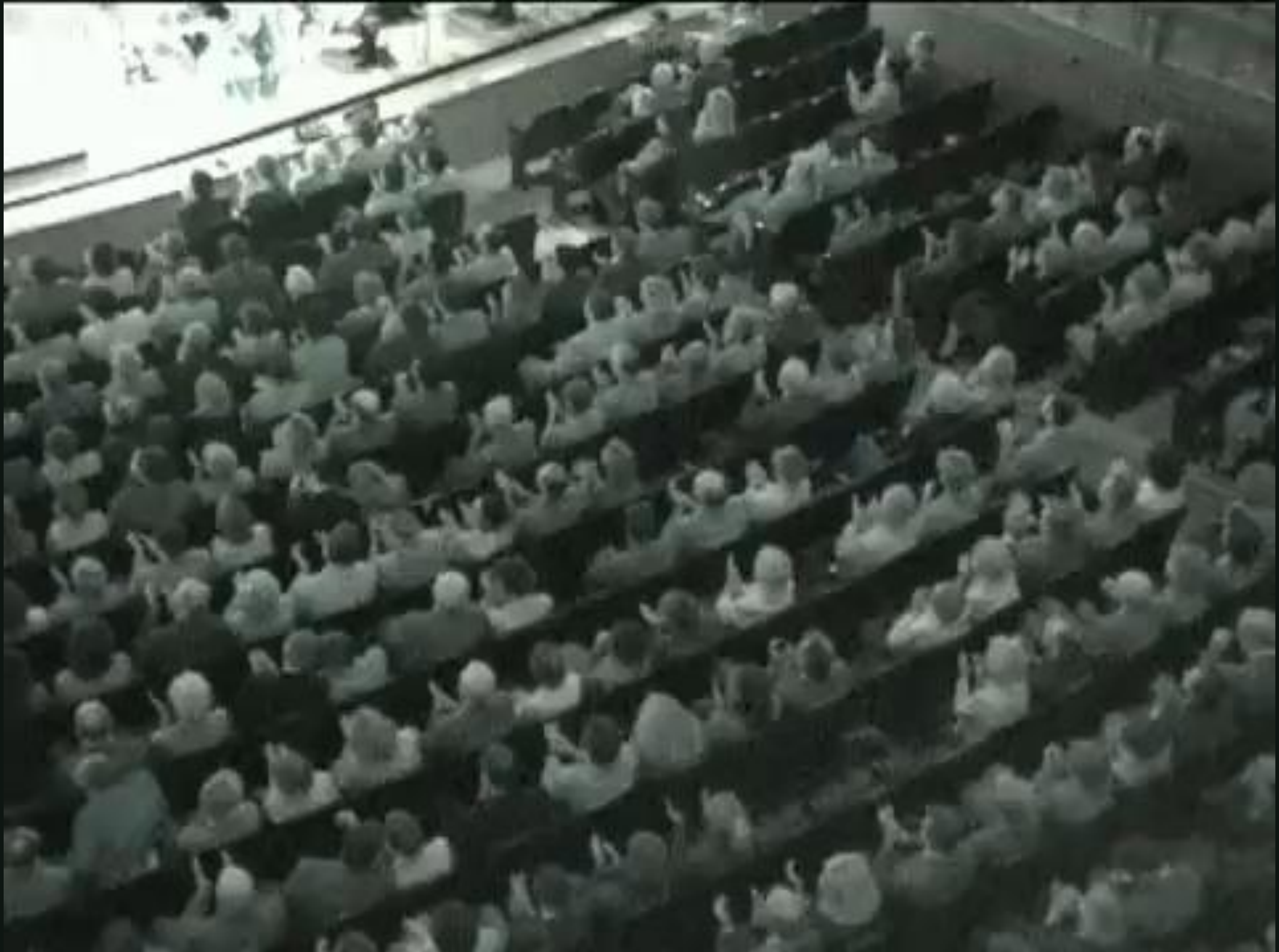
SUMMER SCHOOL - INTRODUCTION TO COMPLEX SYSTEMS

# *Feedbacks & Emergence*

**Dr. Robbin Bastiaansen**
r.bastiaansen@uu.nl

**Slides available at:**
**bastiaansen.github.io/talks/CCSS_SS_2026_Emergence.pdf**

# Part I
# Examples
# & Definition



# Some definitions

A **complex system** is a (typically large) collection of (typically similar) elements that are connected to each other by 'signals'.

'signals' could be e.g. money, energy, agreements, hormones, information, tweets, infection, friendship

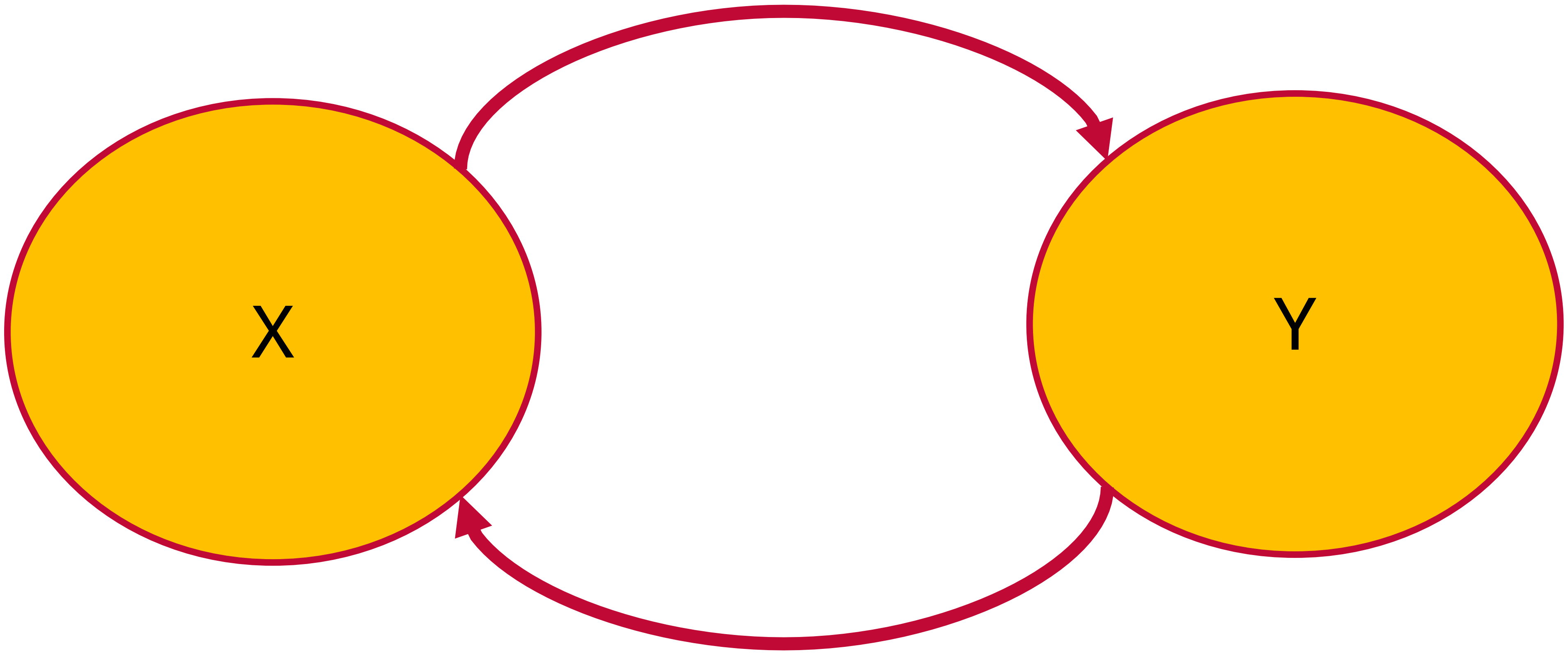
**Emergence**

Indivuals perceive signals from other individuals. Then, a sufficiently summed strength of these signals causes individuals to act in certain ways that produce additional signal, leading to emergent collective behaviour.

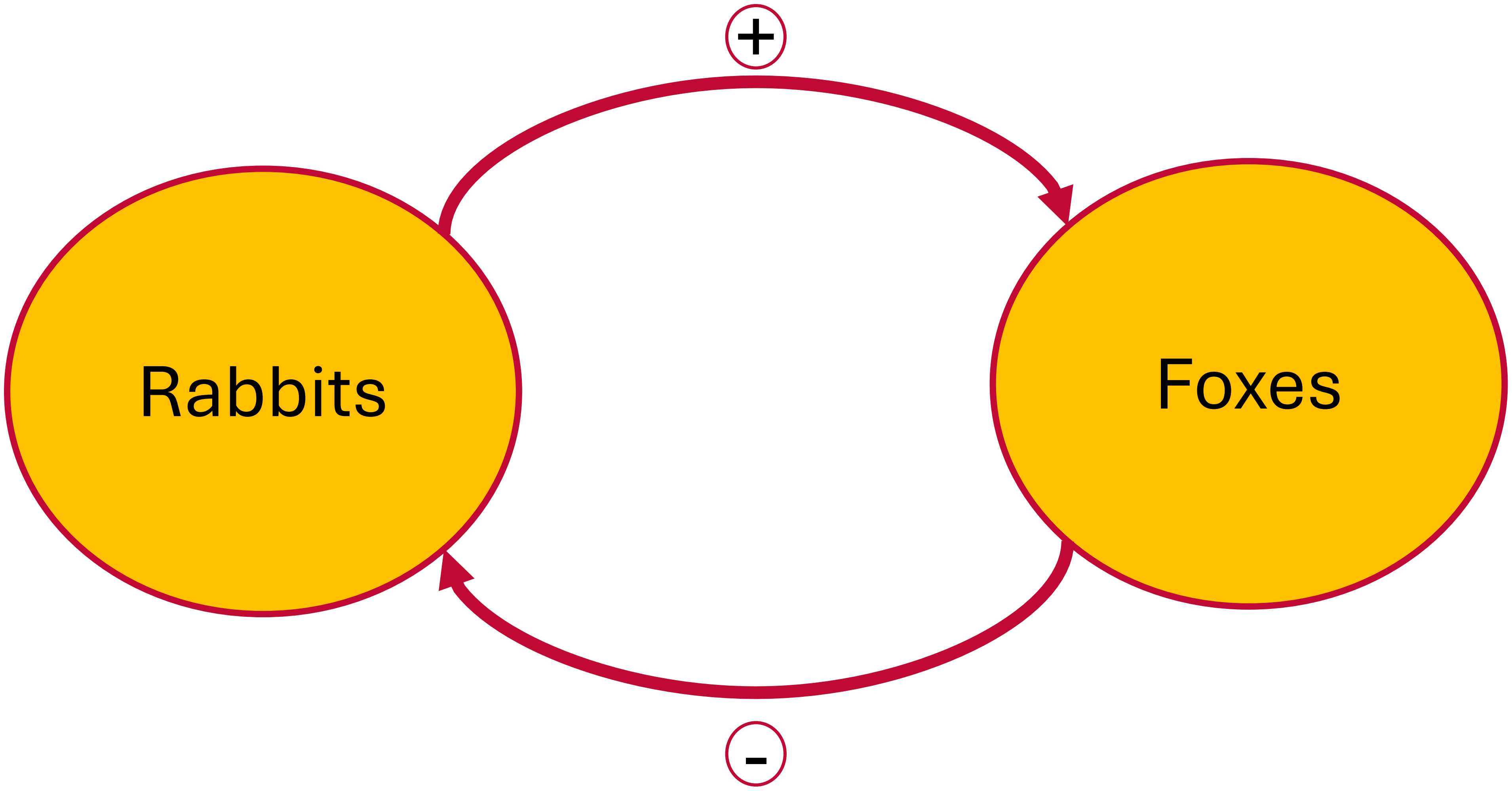
Crucially, NO central control. It is self-organised!

# *Part II*
# *Feedbacks*

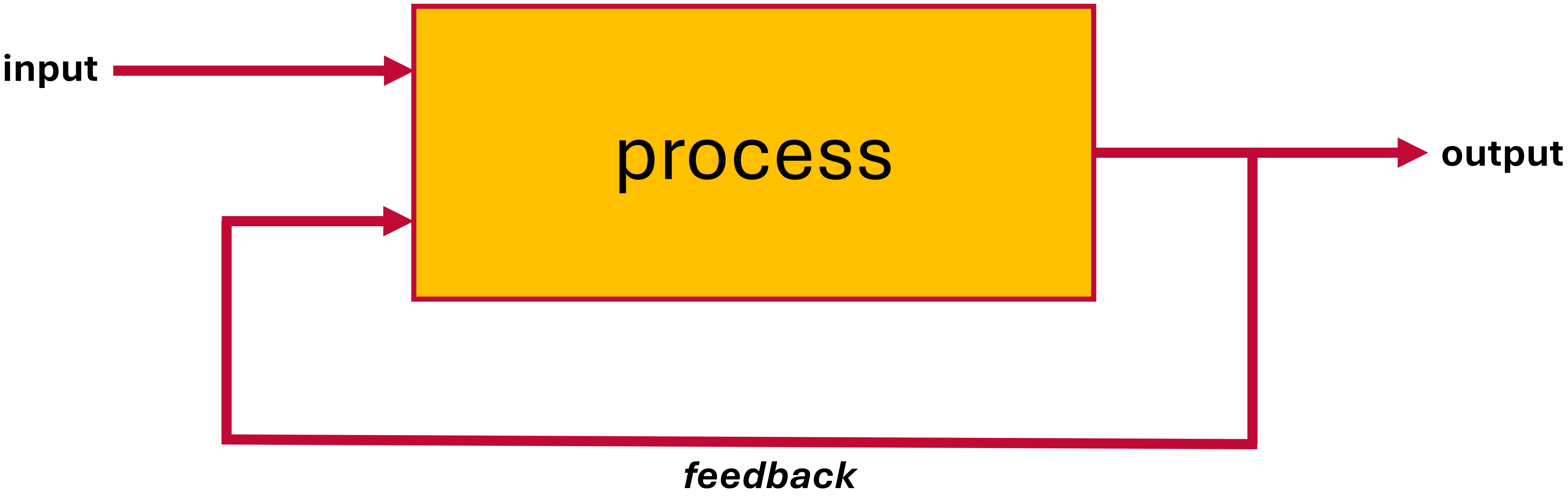
# Feedback loops



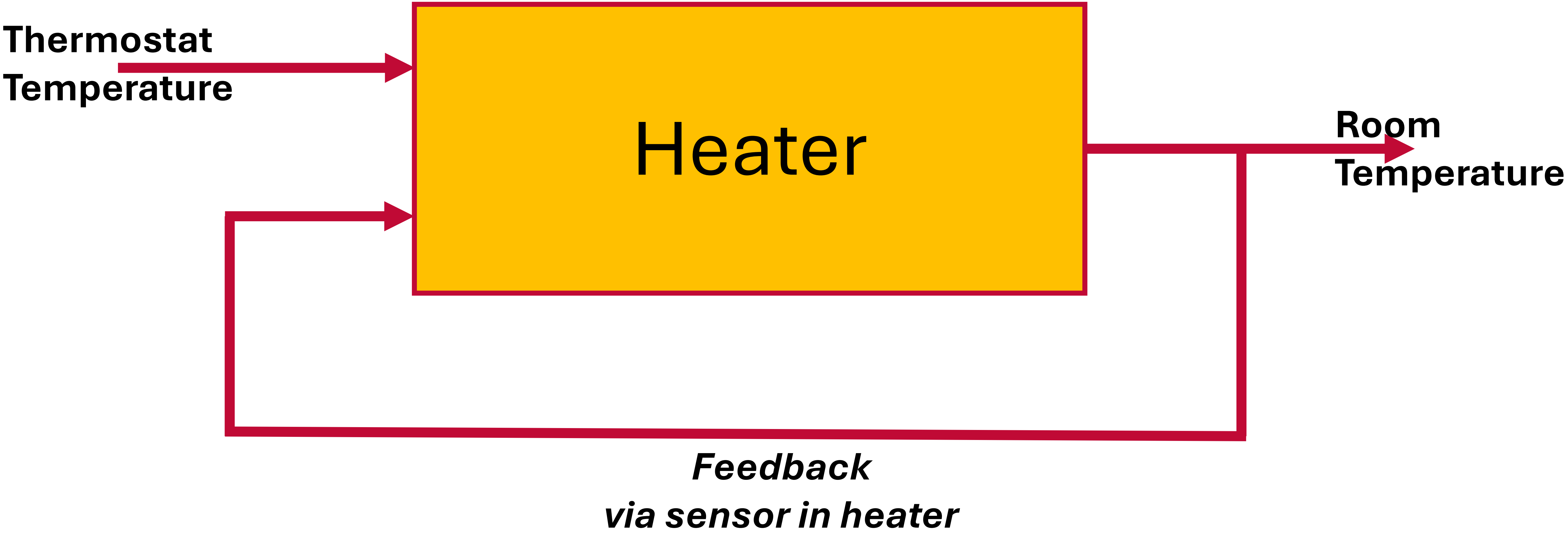
# Feedback loops – Example predator-prey system



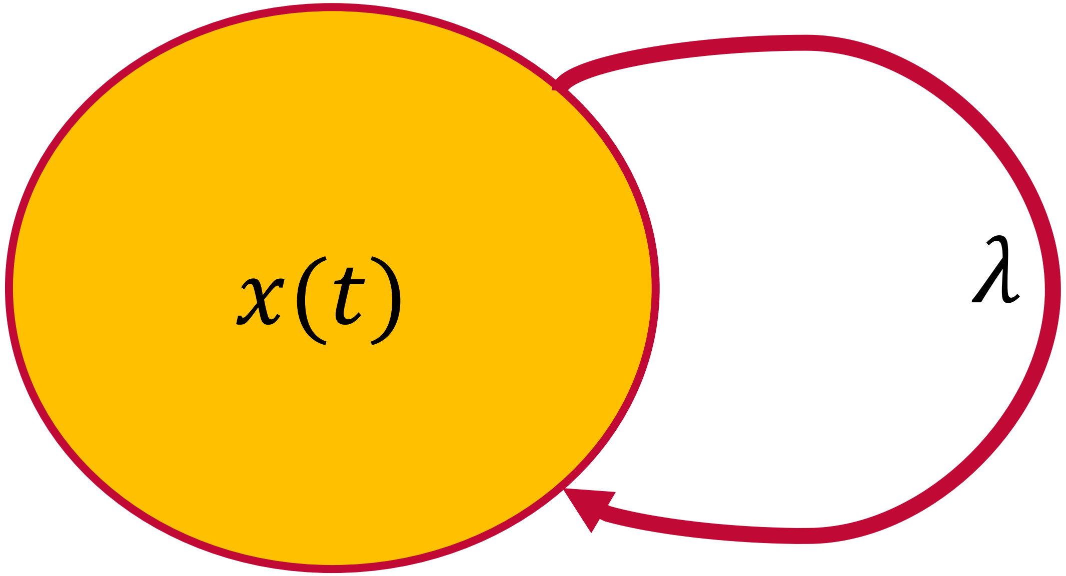
# Feedback loops – classic control theory view



# Feedback loops – classic – example



# Dynamics




Generalises with multiple components:

$$\frac{d\vec{x}}{dt} = A\,\vec{x}$$

and eigenvalues of matrix $A$

$$\frac{dx}{dt}(t) = \lambda\, x(t)$$

$$x(t) = e^{\lambda\, t}\, x_0$$

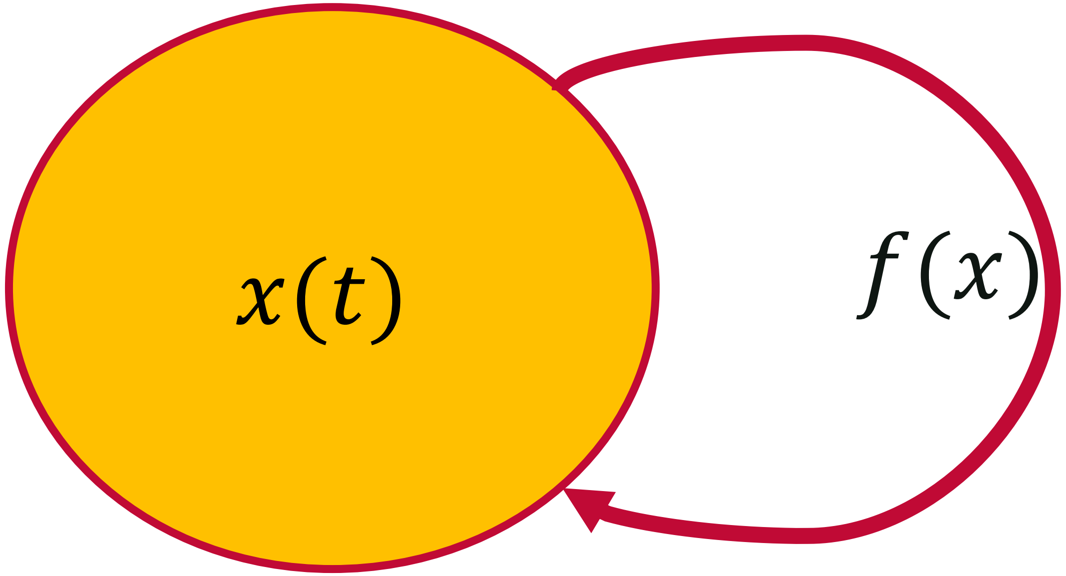
$Re\,\lambda < 0 : x(t) \to 0$

**negative** feedback

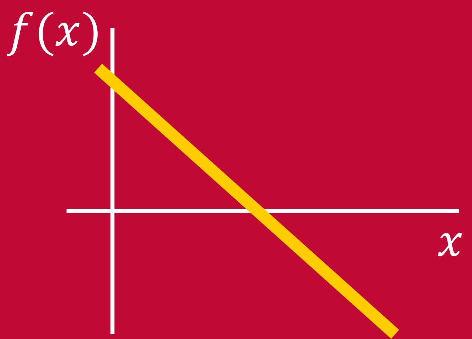
$Re\,\lambda > 0 : x(t) \to \infty$

**positive** feedback

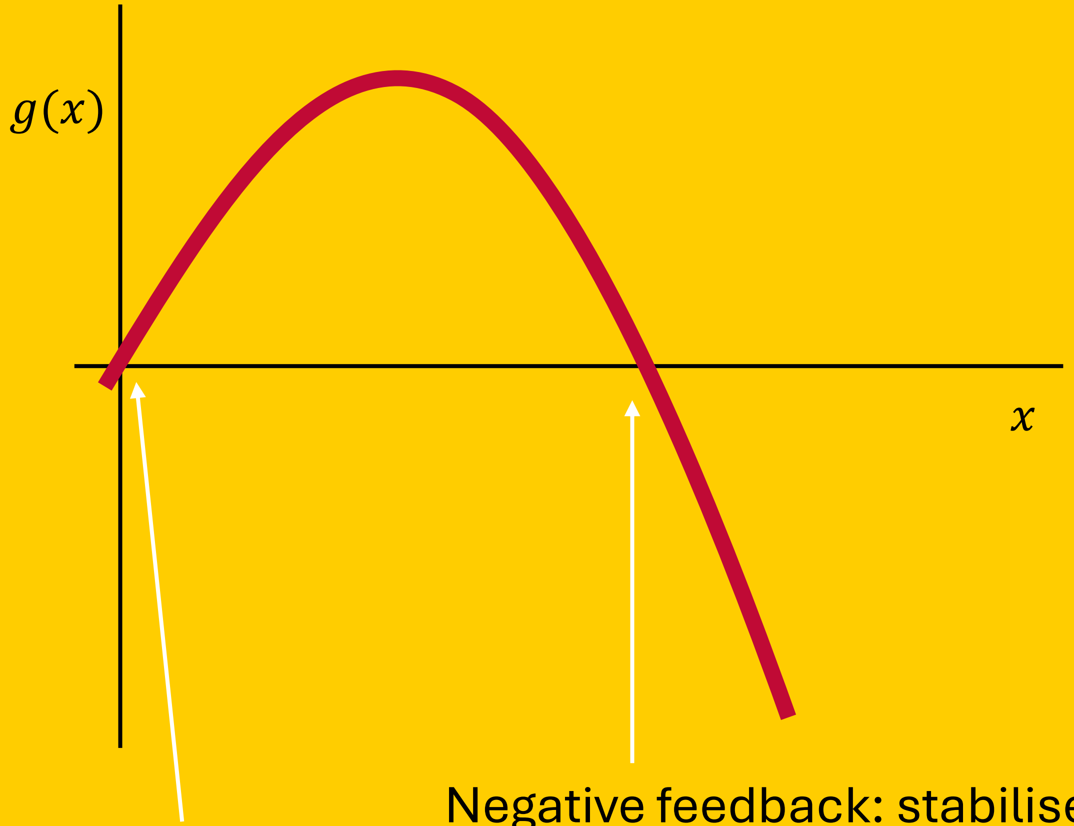
# Nonlinear feedbacks




Example:
$f(x) = 1 - x$




$$\frac{dx}{dt}(t) = f(x)x =: g(x)$$




Negative feedback: stabilises

Positive feedback: destabilises

# Part III
# Mean Field Approximations

# Connecting micro to macro

**MODEL**

state $s_j(t)$

$0$ : no vegetation

$1$ : vegetation

**Windblown**

$$\mathbb{P}(0 \to 1) = \epsilon$$

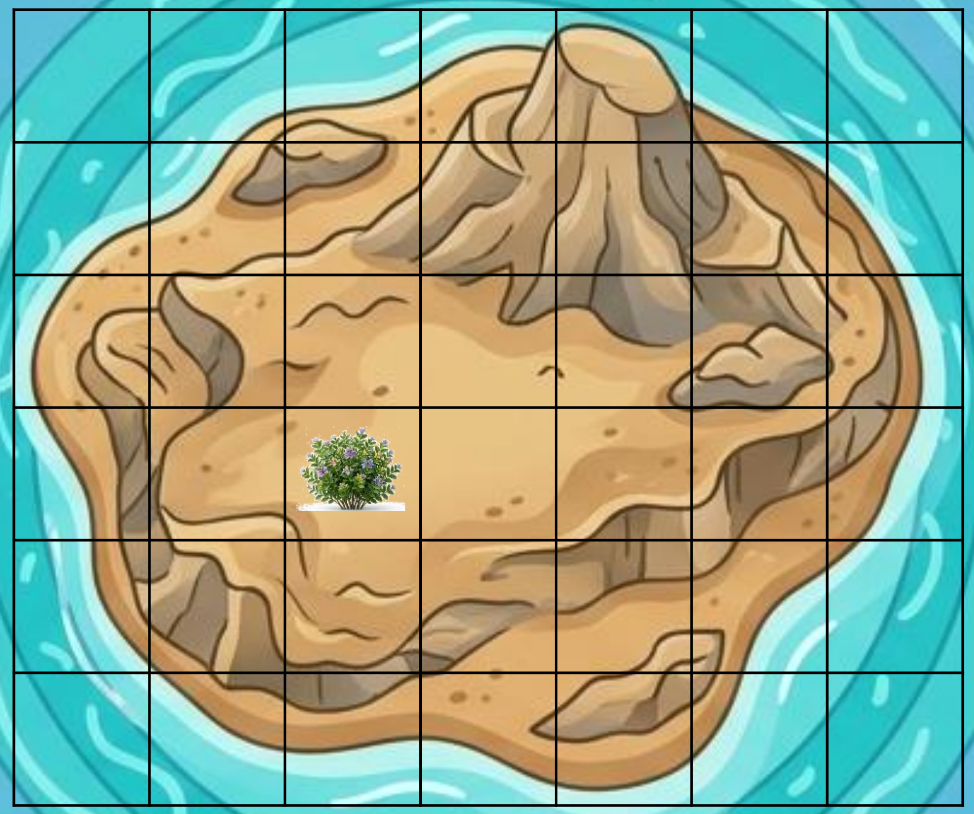
**Reproduction**

$$\mathbb{P}(0 \to 1) = \alpha\,\rho^2$$

with $\rho$ density around site

**Mortality**

$$\mathbb{P}(1 \to 0) = \delta$$



Implemented at bastiaansen.github.io/CCSS_CA.html

# Mean-field approximation

**MODEL**

state $s_j(t)$

$0$ : no vegetation

$1$ : vegetation

**Windblown**

$$\mathbb{P}(0 \to 1) = \epsilon$$

**Reproduction**

$$\mathbb{P}(0 \to 1) = \alpha\, \rho^2$$

with $\rho$ density around site

**Mortality**

$$\mathbb{P}(1 \to 0) = \delta$$

**Mean field approximation**

Let $S(t) \coloneqq \frac{1}{N}\sum_{j=1}^{N} s_j(t)$ and let $x(t) \coloneqq \mathbb{E}[S(t)]$

Then:

$$x(t+1) = \mathbb{E}[S(t+1)]$$
$$= \mathbb{E}[s(t) = 1] \cdot \mathbb{P}(1 \to 1) + \mathbb{E}[s(t) = 0] \cdot \mathbb{P}(0 \to 1)$$
$$= x(t) \cdot (1-\delta) + \big(1 - x(t)\big) \cdot (\epsilon + \alpha \mathbb{E}[\rho^2])$$
$$= x(t) \cdot (1-\delta) + \big(1 - x(t)\big) \cdot (\epsilon + \alpha\, x^2)$$

In which we assume spatial mixing, i.e. $\mathbb{E}[\rho^2] = x^2$

So: $\boldsymbol{\frac{dx}{dt} = \epsilon - (\epsilon + \delta)x + \alpha x^2 - \alpha x^3}$

Implemented at bastiaansen.github.io/CCSS_CA.html

# Analyis of mean-field equation

$$\frac{dx}{dt} = \epsilon - (\epsilon + \delta)x + \alpha x^2 - \alpha x^3$$

Using dimensional analysis this reduces to

$$\frac{dy}{dt} = \nu \; - y + y^2 - r\, y^3$$

with

$$\nu := \frac{\epsilon\alpha}{(\epsilon+\delta)^2}, r := \frac{\epsilon+\delta}{\alpha}, \mathrm{y} := \frac{\mathrm{x}}{\mathrm{r}}$$

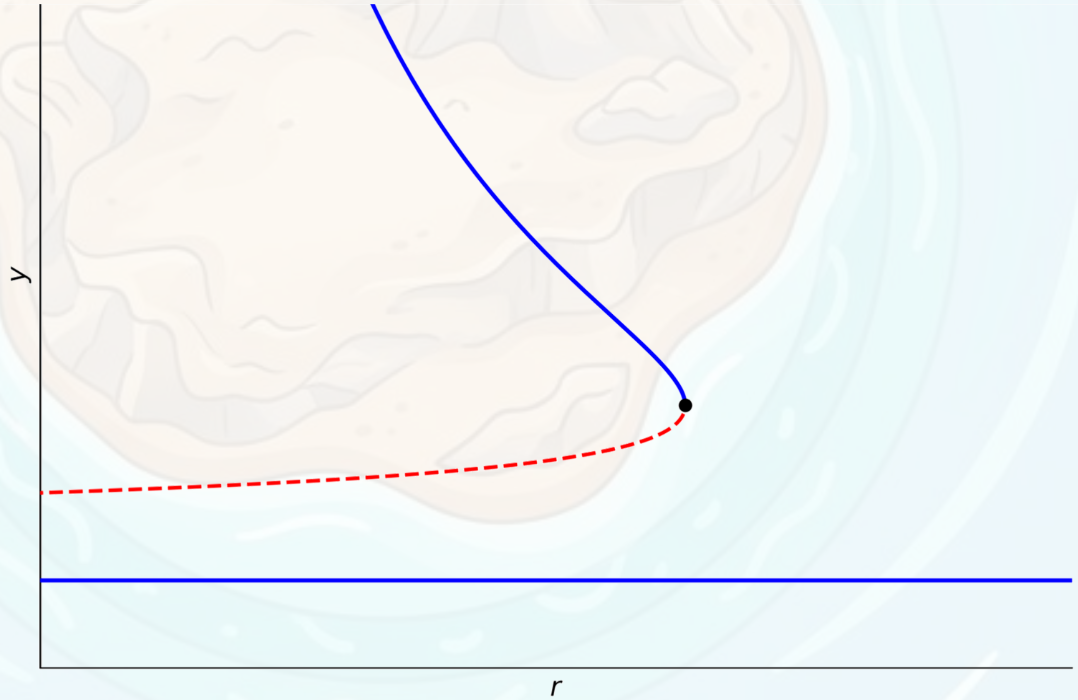
We take $\nu = 0$ ($\epsilon = 0$) and study

$$\frac{dy}{dt} = -y + y^2 - r\, y^3$$

$$\frac{dy}{dt} = -y + y^2 - r\, y^3$$

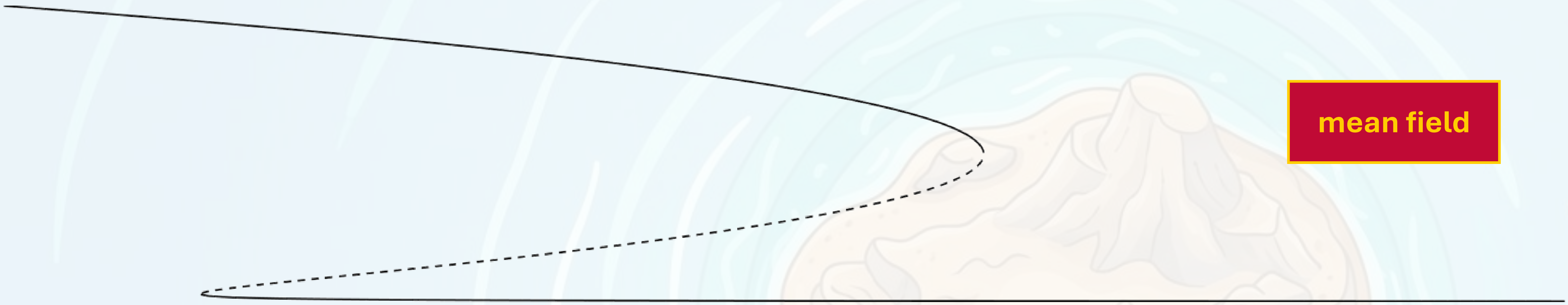
Fixed points:

$$y_1 = 0, y_{2,3} = \frac{1 \pm \sqrt{1-4r}}{2r}$$

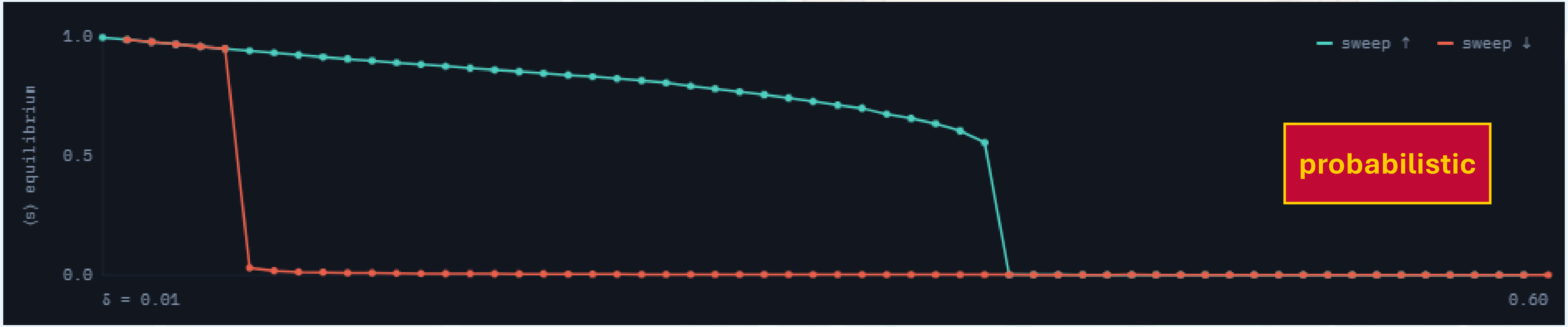


# Compare to simulation

**Parameters:**
$\epsilon = 0.001, \alpha \coloneqq 1.50$



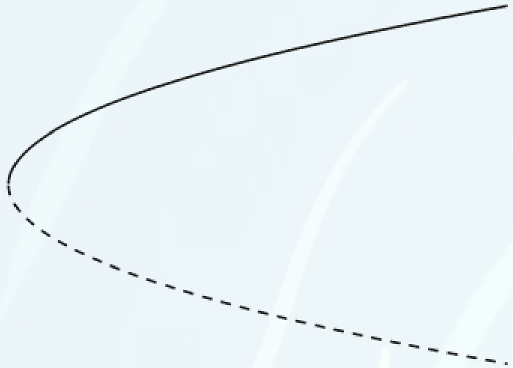
mean field



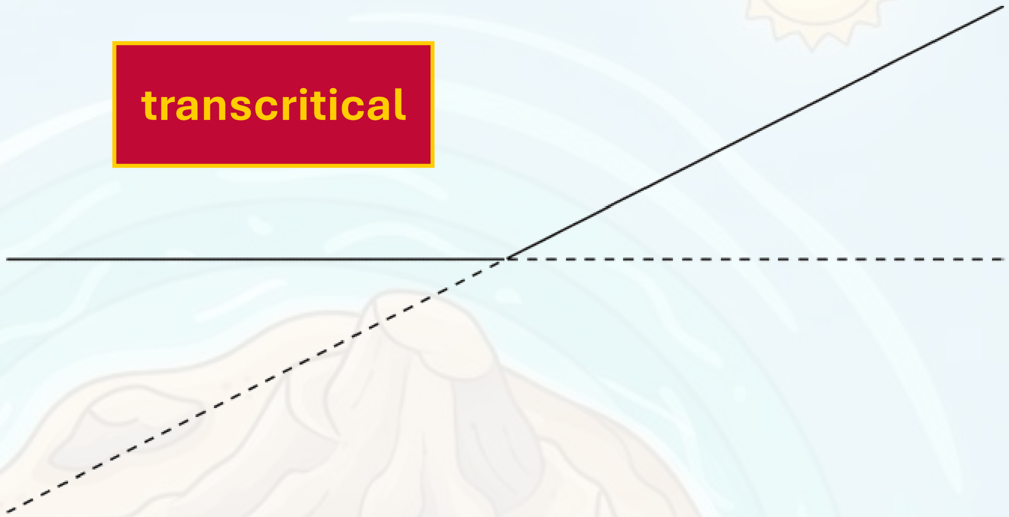

probabilistic

# Other bifurcations

fold

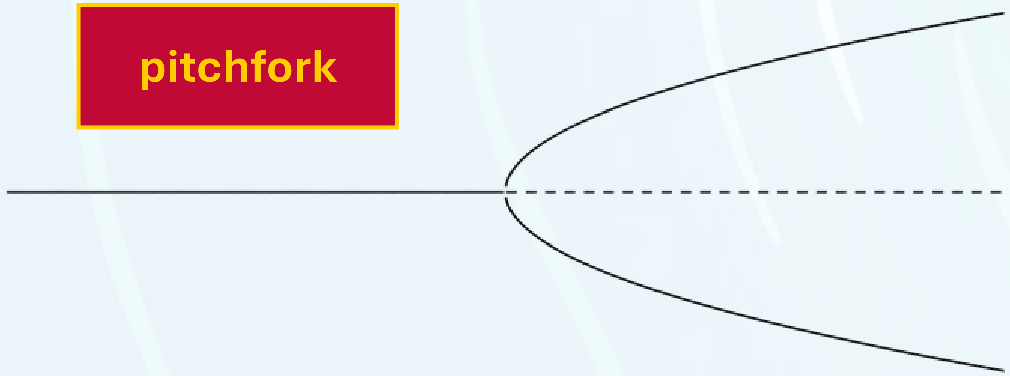


transcritical

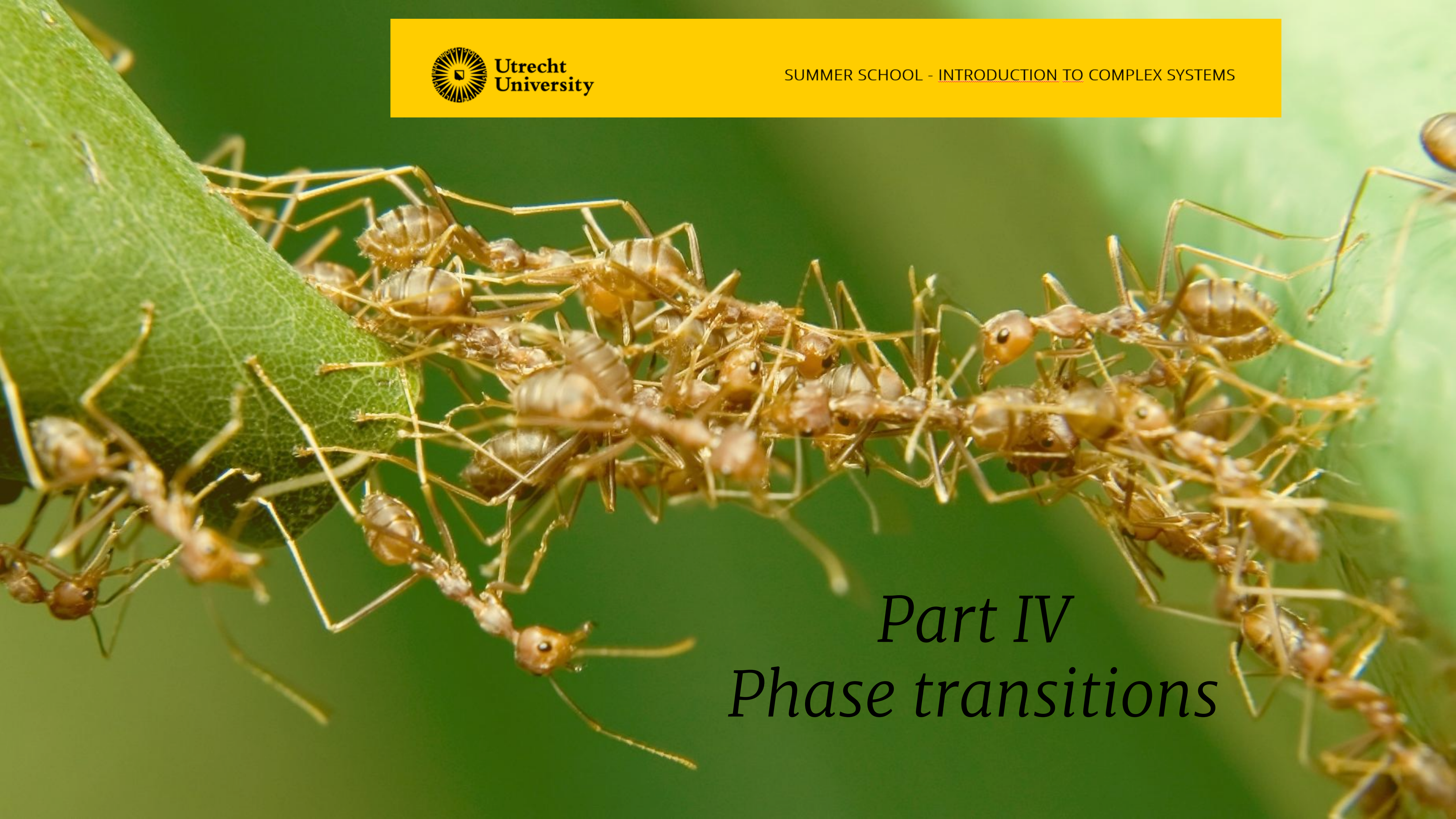


Key property is oscillations

pitchfork



Hopf



# *Part IV*
# *Phase transitions*

# Simplified Ising Model

## MODEL

state $s_j(t)$
$1$ : spin up
$-1$ : spin down

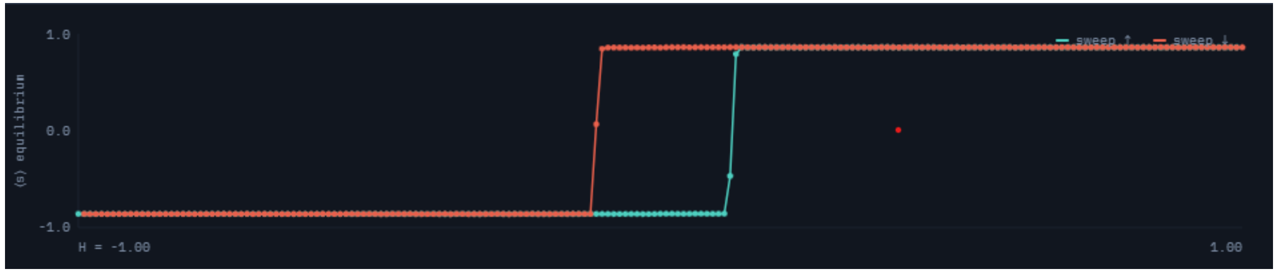
**Update rule**
- with probability $P$: flip cell
- Otherwise:
    - count spin of neighbours
    - add bias $H$
    - If sum is $\geq 0$: cell will be +1
    - If sum is $< 0$: cell will be -1

**Varying $P$**
**$(H = 0)$**

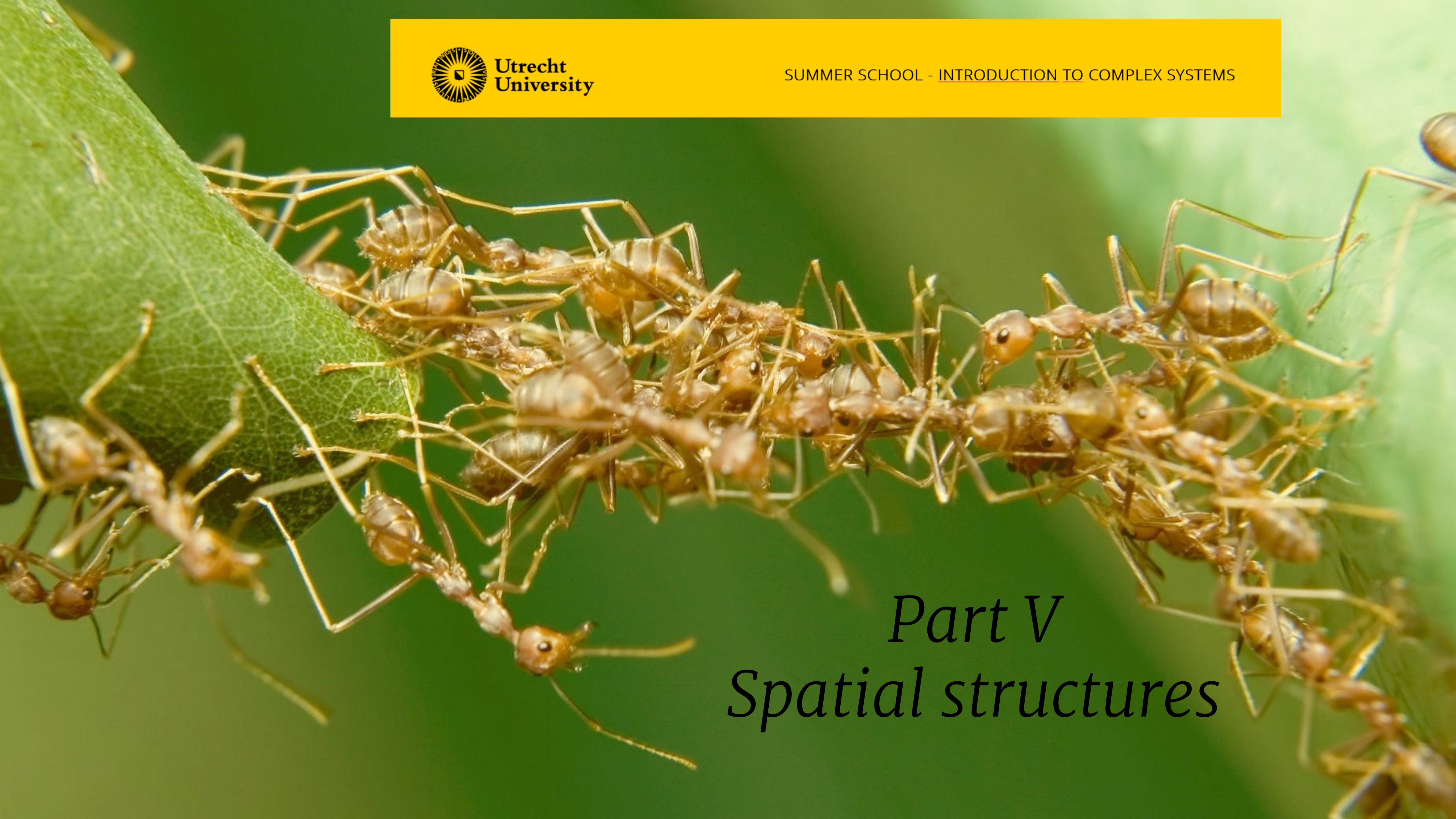
**second**-order phase transition


**Varying H**
**$(P = 0,14)$**

**first-order** phase transition




**Implemented at https://bastiaansen.github.io/CCSS_CA.html**



# Part V
# Spatial structures

# Conway's Game of Life

**MODEL**

state $s_j(t)$

$1$ : ON

$0$ : OFF

**Update rule**

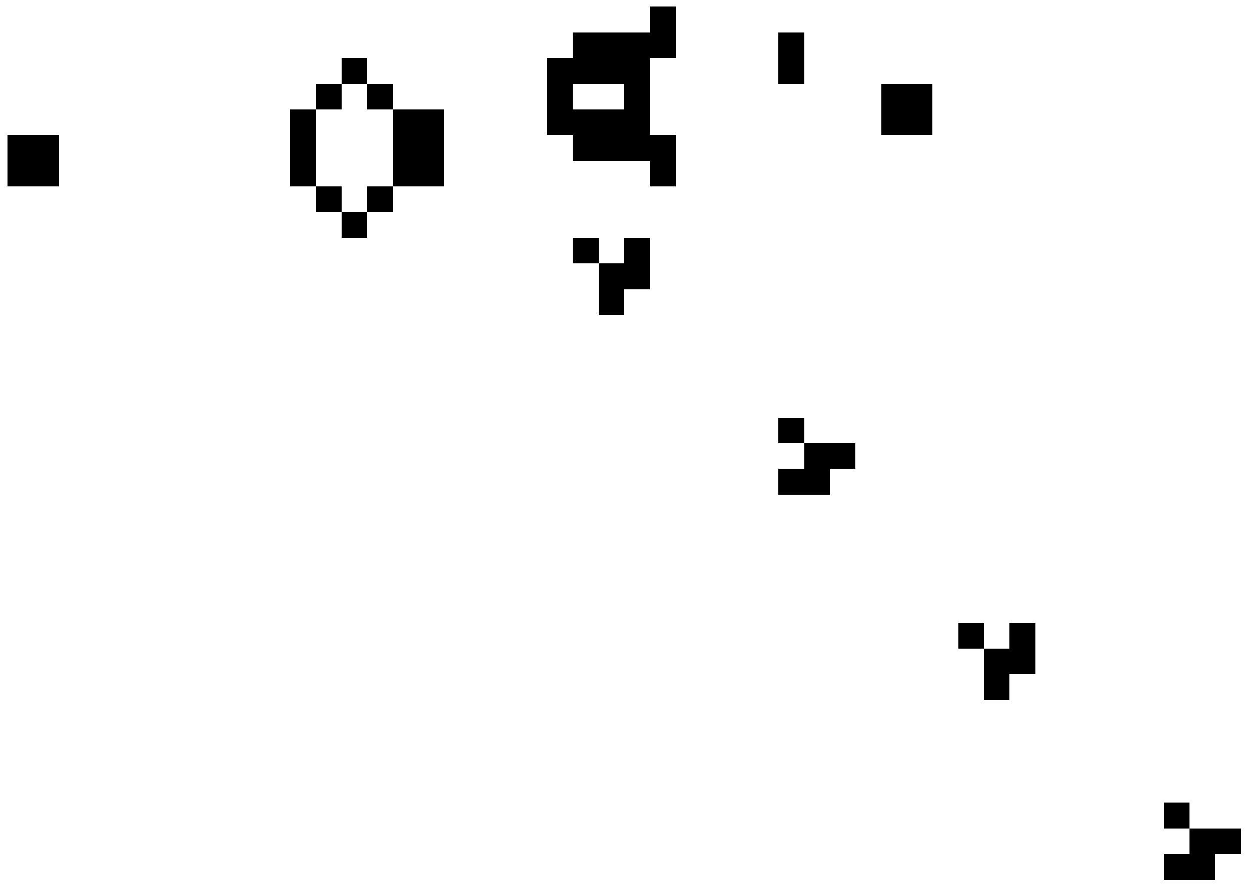
ON → OFF:

- if $<$ 2 ON neighbours
- if $>$ 3 ON neighbours

OFF → ON:

- if 3 ON neighbours

*(otherwise cell does not change)*



**Implemented at bastiaansen.github.io/CCSS_CA.html**

# Turing patterning

**MODEL**

state $s_j(t)$

1 : ALIVE

0 : DEAD

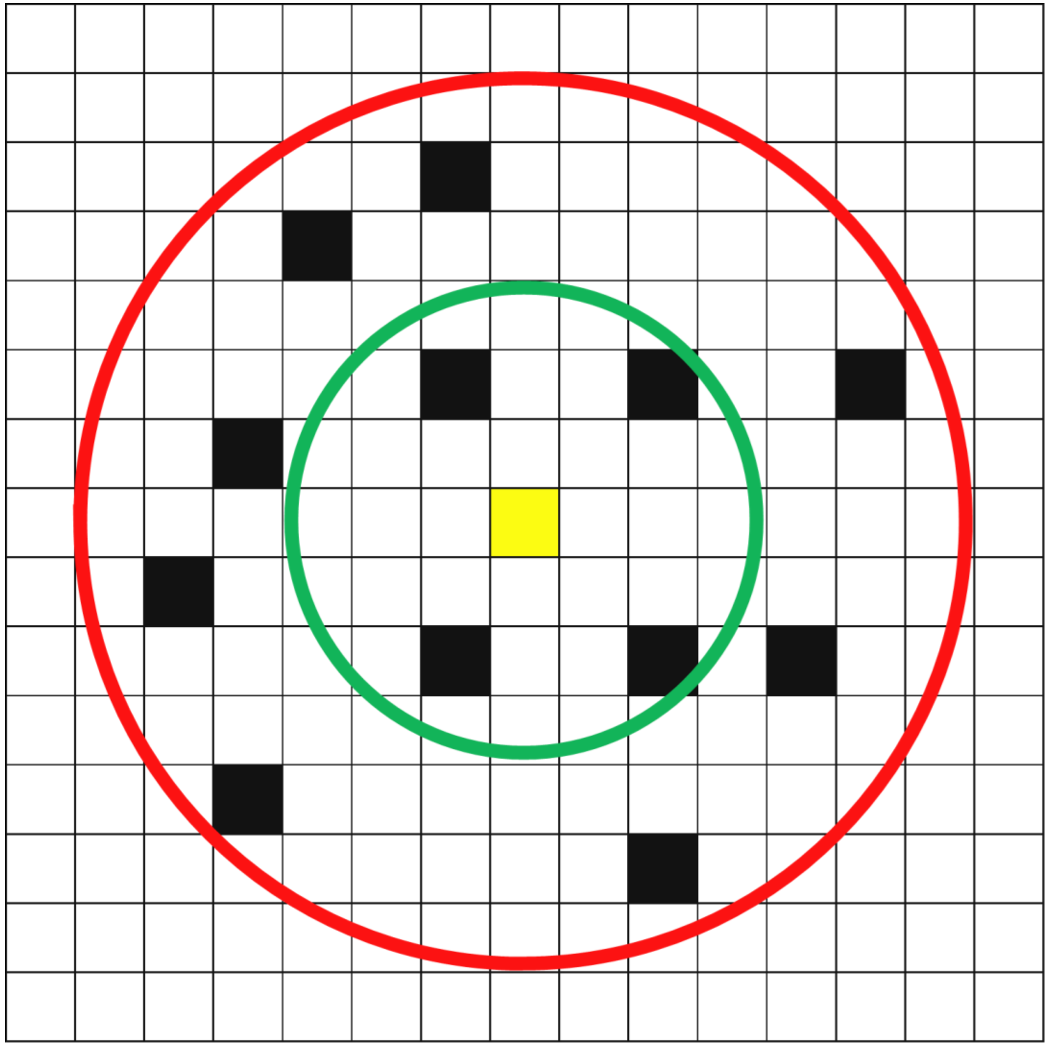
**Update rule**

Per cell, compute

$$S := \sum_{\mathcal{N}_a} w_a \frac{s_j}{|\mathcal{N}_a|} - \sum_{\mathcal{N}_i} w_i \frac{s_j}{|\mathcal{N}_i|}$$

If $S > 0$: cell becomes ALIVE

If $S \leq 0$: cell becomes DEAD



**scale-dependent feedback**

**Implemented at bastiaansen.github.io/CCSS_CA.html**

# Turing patterning

**MODEL**

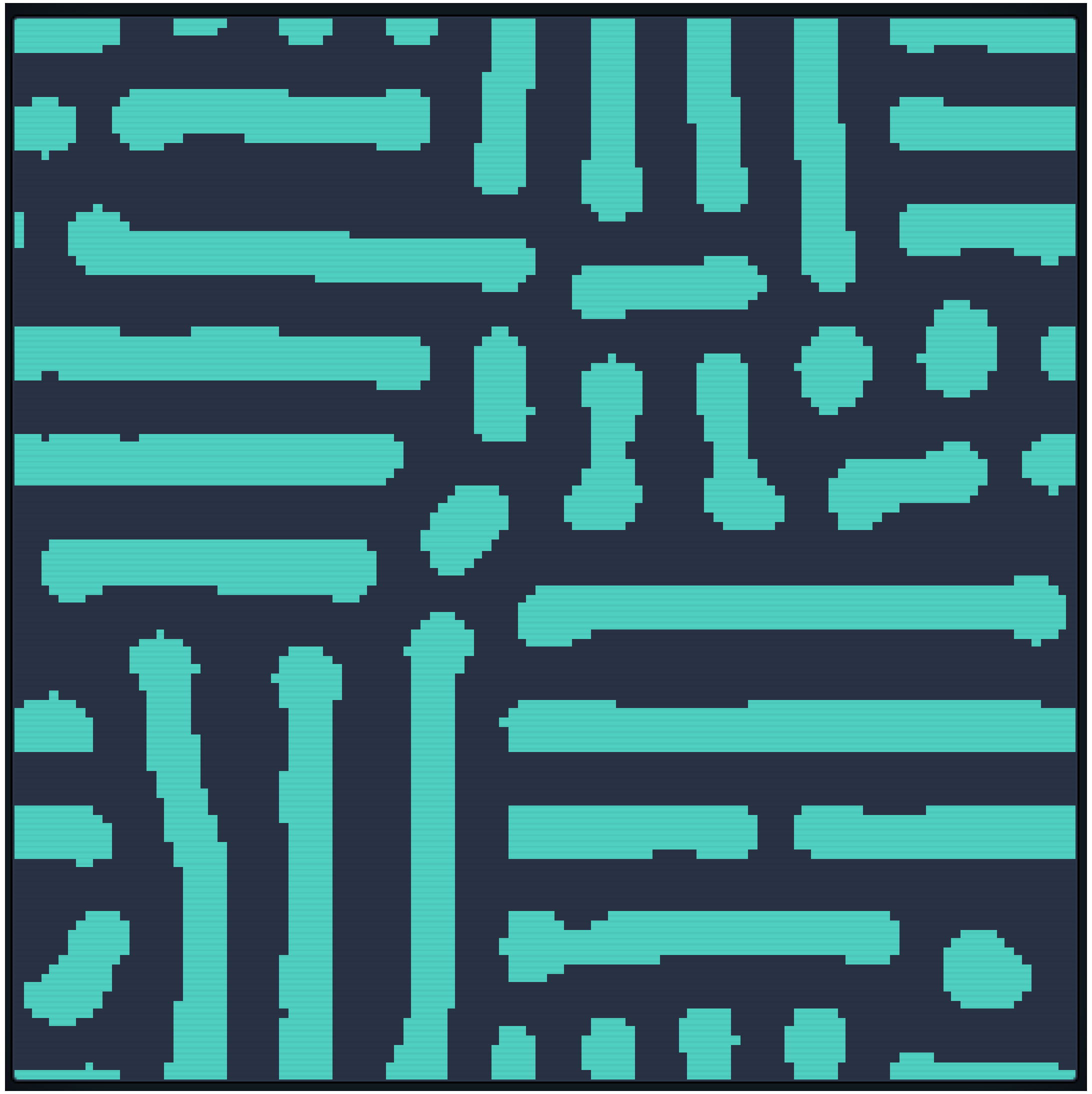
state $s_j(t)$
1 : ALIVE
0 : DEAD

**Update rule**

Per cell, compute

$$S := \sum_{\mathcal{N}_a} w_a \frac{s_j}{|\mathcal{N}_a|} - \sum_{\mathcal{N}_i} w_i \frac{s_j}{|\mathcal{N}_i|}$$

If $S > 0$: cell becomes ALIVE
If $S \leq 0$: cell becomes DEAD



Parameters: $r_a = 3, w_a = 1{,}15, r_i = 6, w_i = 1.31$

**Implemented at bastiaansen.github.io/CCSS_CA.html**

# Island model (only local interaction)

**MODEL**

state $s_j(t)$

$0$ : no vegetation

$1$ : vegetation

**Windblown**

$$\mathbb{P}(0 \to 1) = \epsilon$$

**Reproduction**

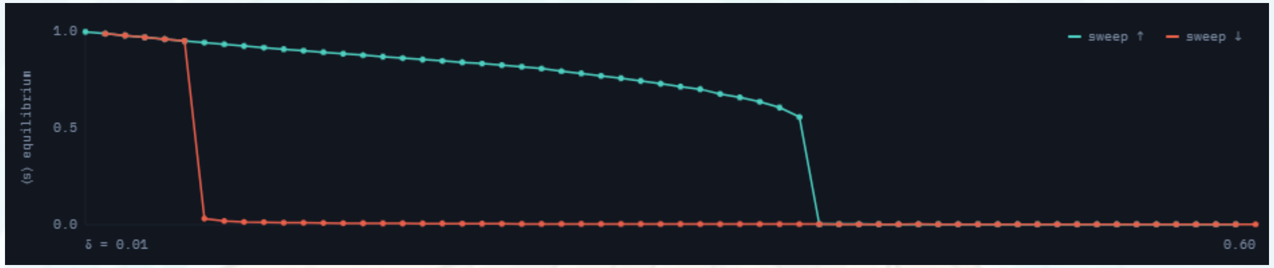
$$\mathbb{P}(0 \to 1) = \alpha\, \rho^2$$

with $\rho$ density around site

**Mortality**

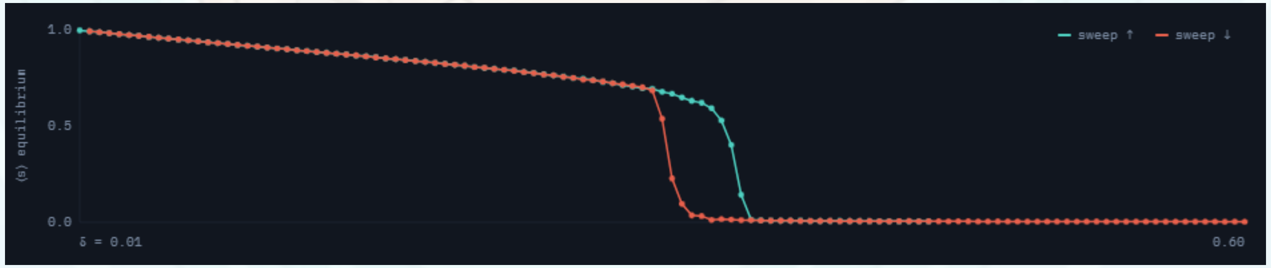
$$\mathbb{P}(1 \to 0) = \delta$$

**Global interaction**

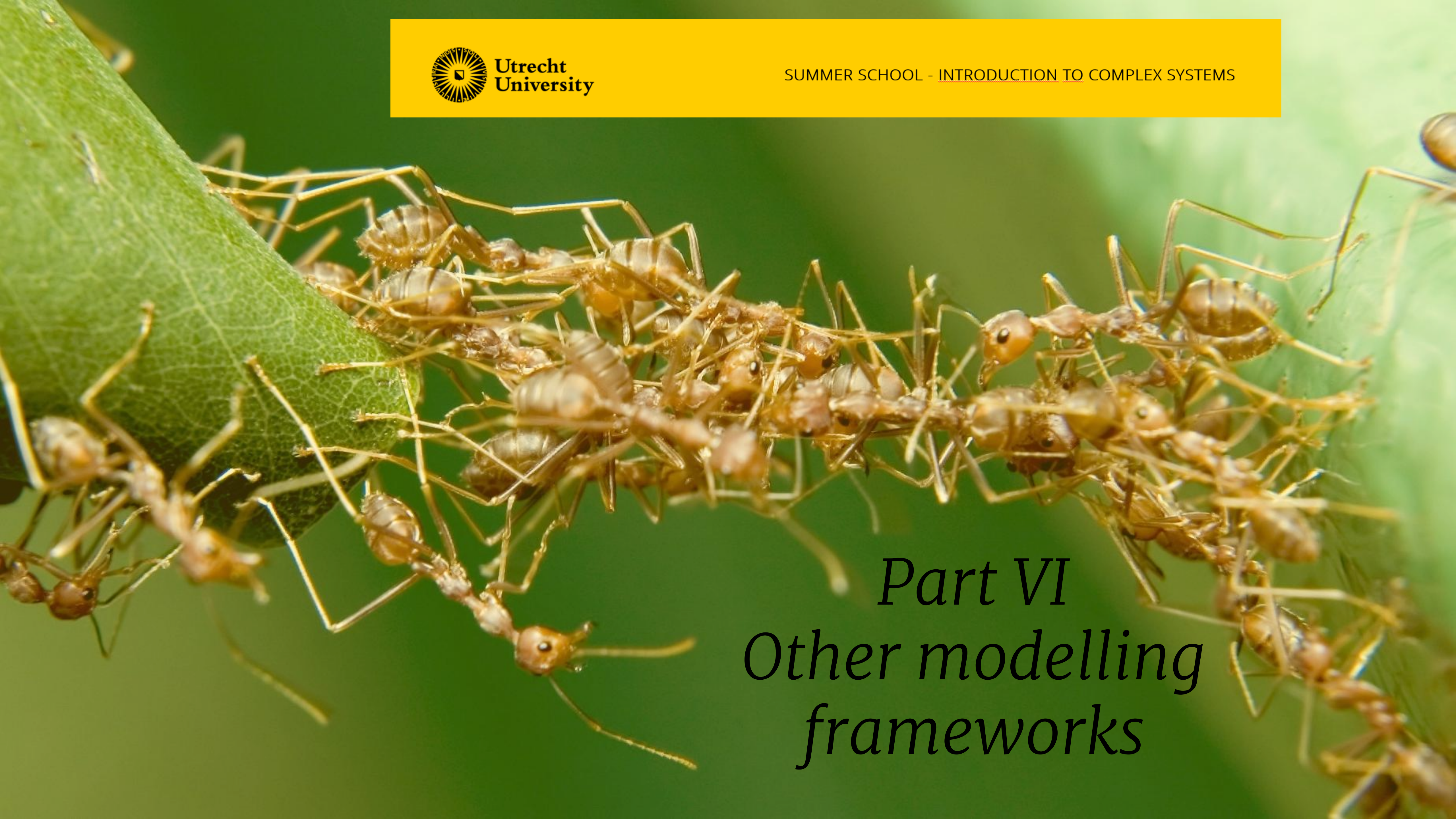



Parameters: $\epsilon = 0.001, \alpha = 1.50, p = 1$ (use $\delta = 0.33$ to see differences)

**Only local interaction**




Implemented at bastiaansen.github.io/CCSS_CA.html



# Part VI
# Other modelling frameworks

# Agent-based / indivual-based model (ABM/IBM)

**MODEL** (from agents.jl)

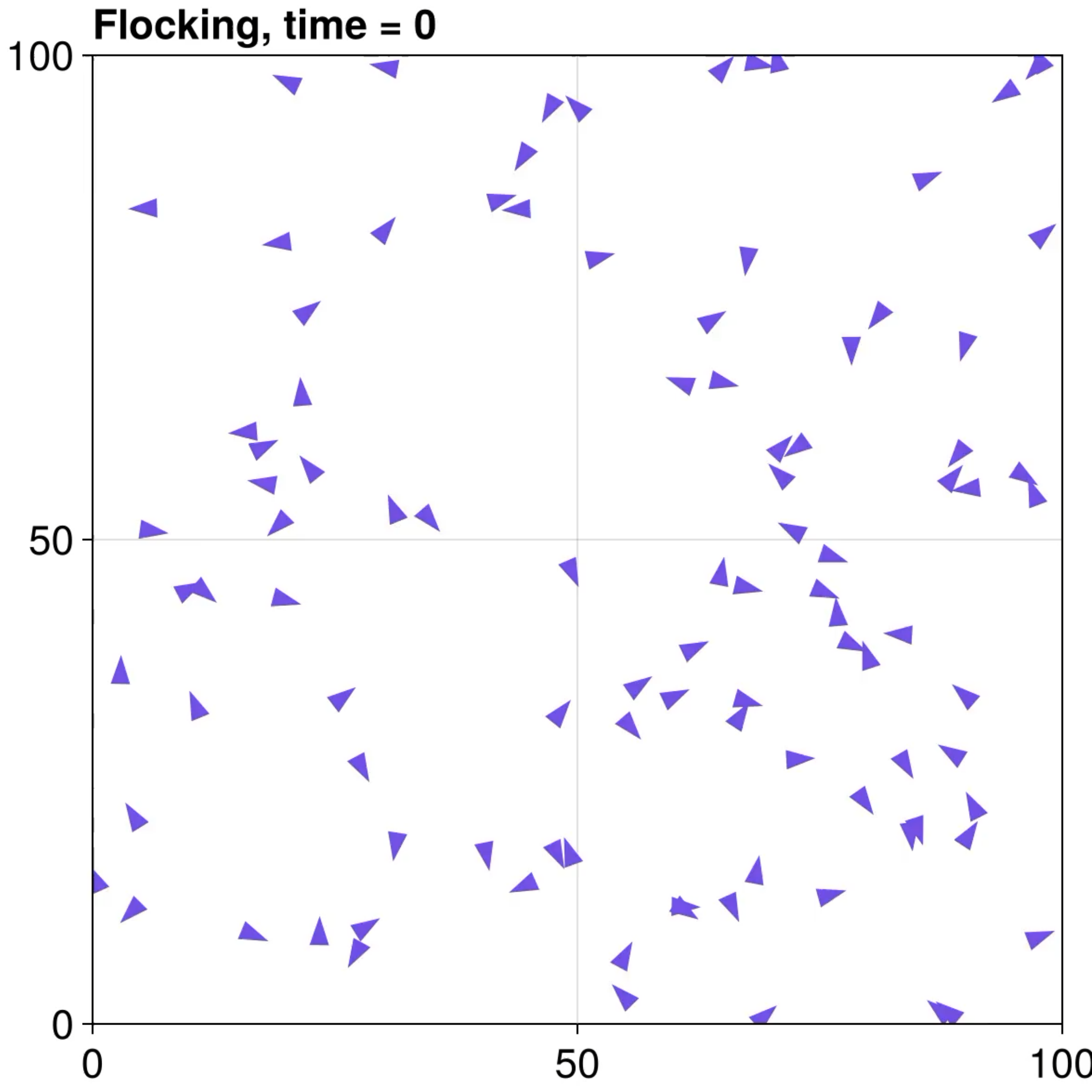
Each agent has location and velocity

**Cohesion**
- Move towards others

**Separation**
- Keep minimal distance to others

**Alignment**
- Match velocity of others

See https://juliadynamics.github.io/Agents.jl/stable/examples/flock/



# Partial differential equations

**MODEL**

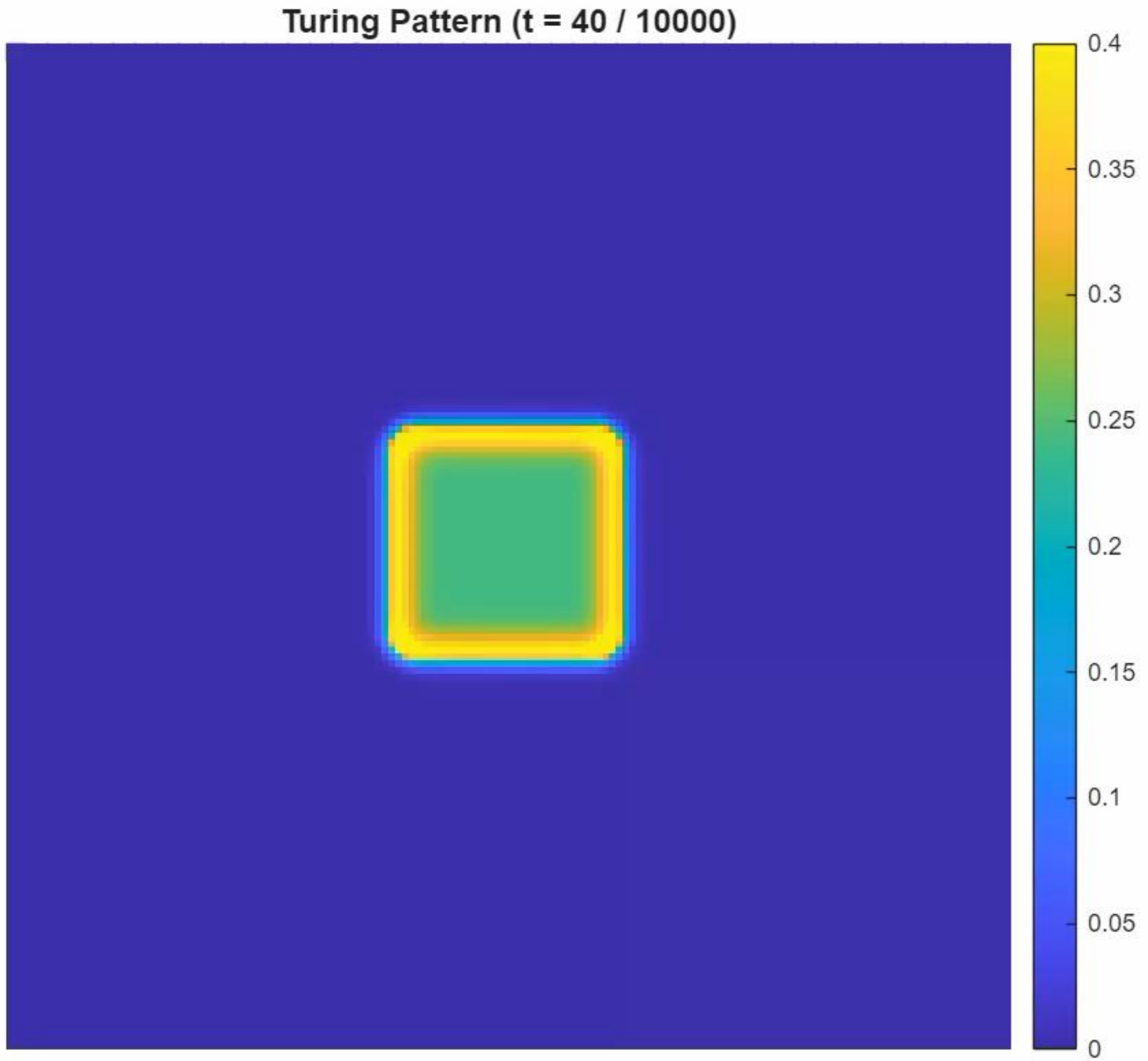
**Gray-Scott model**

$$\frac{\partial U}{\partial t} = D_u \nabla^2 U - UV^2 + F(1 - U)$$

$$\frac{\partial V}{\partial t} = D_v \nabla^2 V + UV^2 - (F + k)V$$

$$D_u = 0.16, D_v = 0.08,$$
$$F = 0.04, k = 0.06$$

Can be derived as spatially-explicit mean-field approximation




Simulation by MATLAB code by Gemini Flash 3.6

# *Part VII*
# *Ok – so what?*

# Micro and macro

- Which micro rules lead to observed macro behaviour?
  Examples:
  - How does flocking work?
  - Why are there spatial patterns?

- What macro behaviour emerges from observed micro dynamics?
  Examples:
  - What effects do pheromone deposits have in an ant colony?
  - How does algorithmic trading influence financial markets?

# How to study

- Numerical simulations
  - Cellular Automata (CA)
  - Agent-Based Models (ABMs / IBMs)
  - Ordinary or Partial Differential Equations (ODEs & PDEs)

- Analytic results only sometimes available

- Often: system-specific (data) analysis is required


Utrecht University
SUMMER SCHOOL - INTRODUCTION TO COMPLEX SYSTEMS
Feedbacks
& Emergence

# Summary

**Feedbacks**

negative/stabilising & positive/destabilising

**Mean-field approximations**

bifurcations/transitions of macro dynamics

**Spatial structures**

deviations from mean-field approximation

emergence of spatial structures

**CA code at bastiaansen.github.io/CCSS_CA.html**
**Slides available at bastiaansen.github.io/talks/CCSS_SS_2026_Emergence.pdf**

**Dr. Robbin Bastiaansen**
r.bastiaansen@uu.nl