Mathematical Modelling and Simulation

# Agent-Based Modelling

## Classification and Case Studies

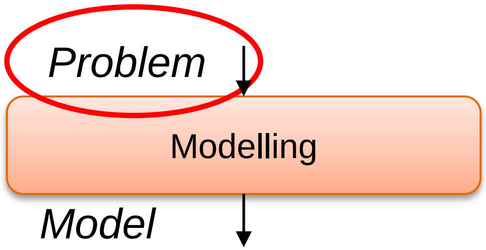
# Lecture Goal

Mathematical Modelling and Simulation

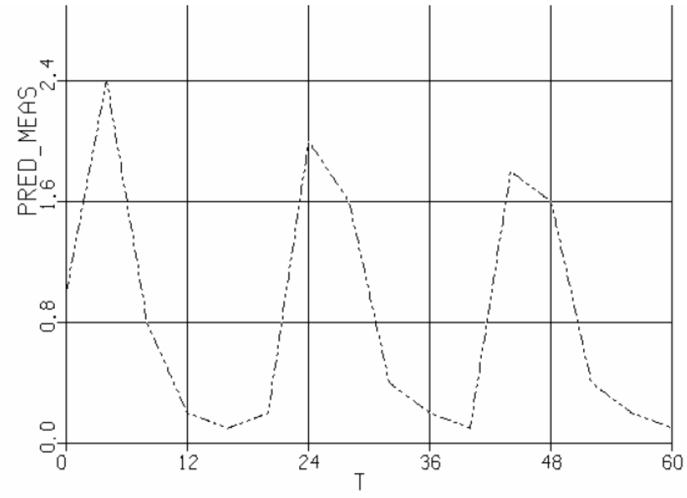
- Get some idea about, how an agent-based model may „look like“
- Get some idea about, how diverse agent-based modelling is
- Classifications of ABMs – clean-up this mess...
- Tips and Tricks

# Case Study 1: Predator Prey Model

Mathematical Modelling and Simulation




Dynamics: Predator eats Prey
Predator / Prey births, deaths




Environment: isolated

Measurement: Predator Population

5 Years = 60 months, quarterly

Problem: When is a reasonable time to use chemical pesticides to reduce number of predators?

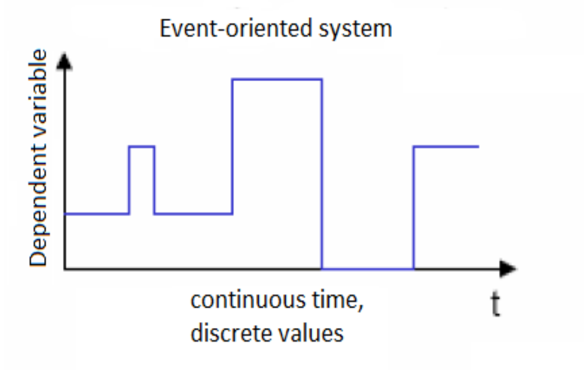
# Case Study 1: Predator Prey Model

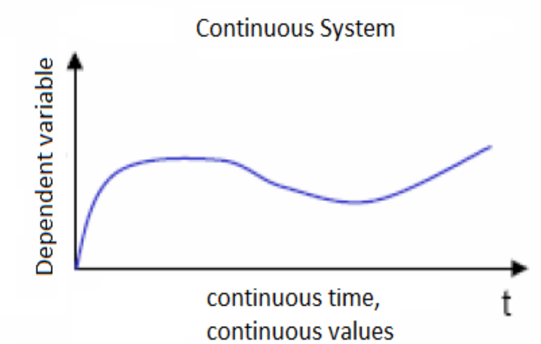


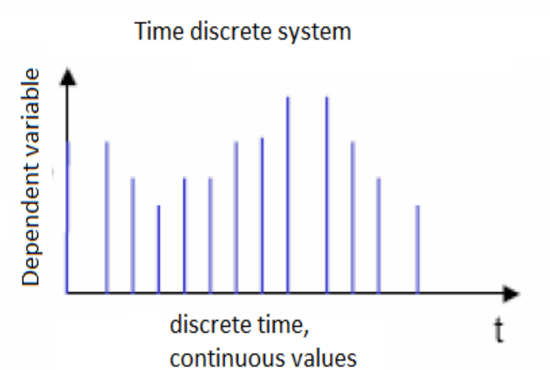
*Problem*

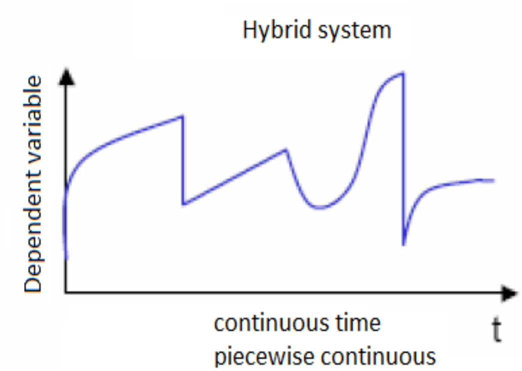
Modelling

*Model*

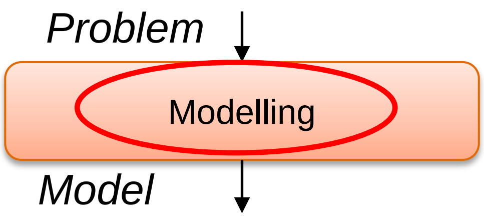


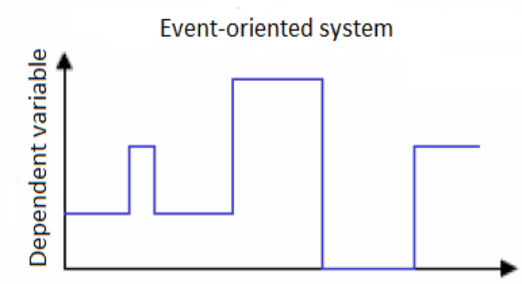



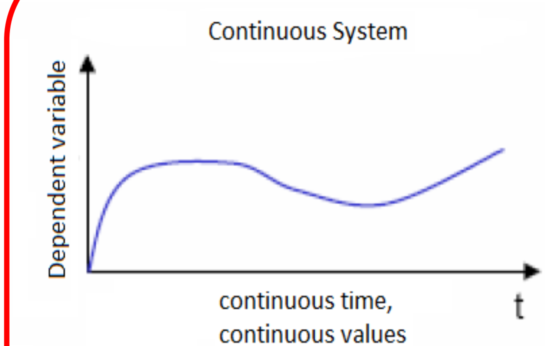



# Case Study 1: Predator Prey Model




**Approach 1 (see 1st lecture)**

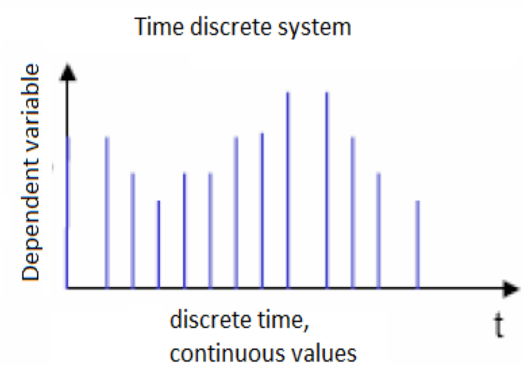


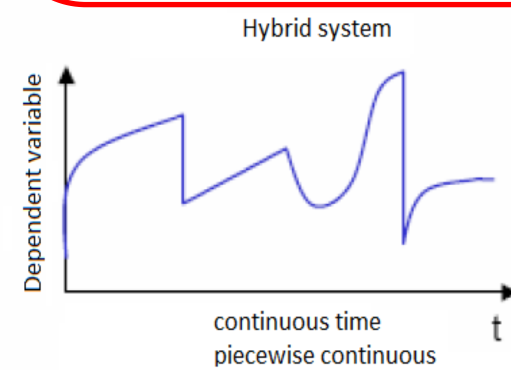

$Y(t)$ ... Prey

$X(t)$ ... Predators

Separation –
Isolated environment

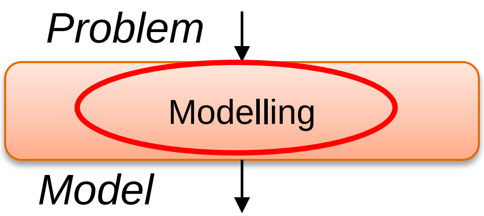
Choice -
2 variables = 2 states

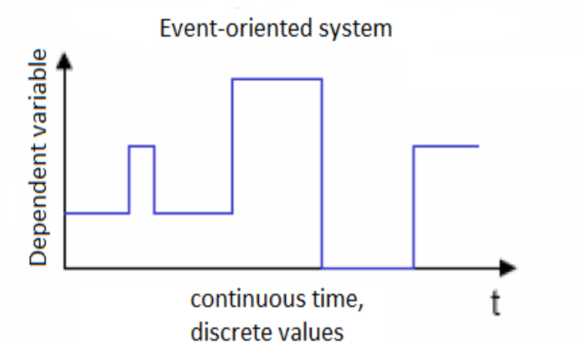


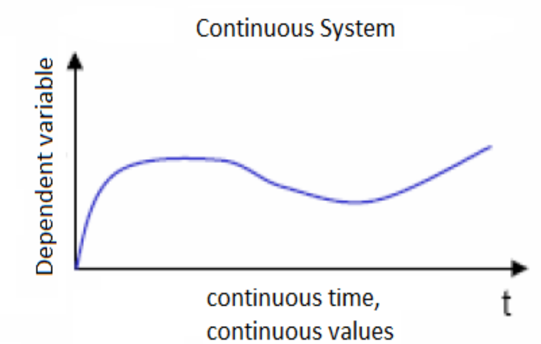




# Case Study 1: Predator Prey Model



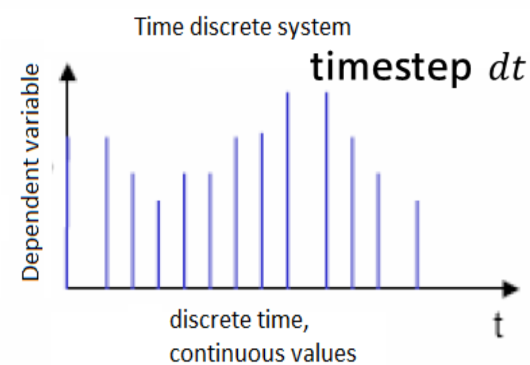


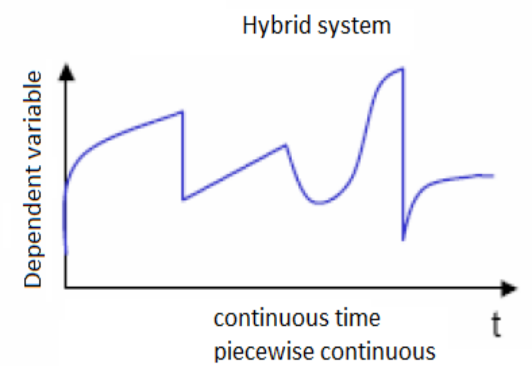




Separation –
Isolated environment
(~ rectangular grid)

Choice -
$Y(t) = \#\{y_i(t)\}$ prey agents
$X(t) = \#\{x_i(t)\}$ predator agents




$y_i(t)$ ... **Prey animal**
$x_i(t)$ ... **Predator animal**

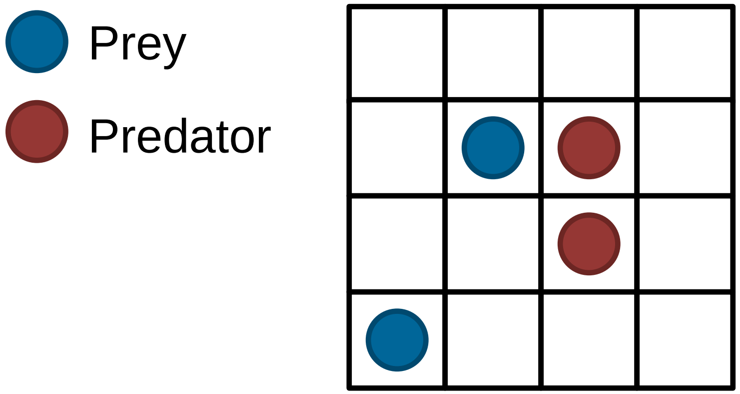



# Case Study 1: Predator Prey Model

- <u>Initialisation:</u>
  $Y(0) = Y_0$ prey agents and $X(0) = X_0$ predator agents distributed uniformly on a rectangular grid with $M \cdot N > Y_0 + X_0$ cells
- <u>Time Step Dynamics:</u>
  A time step is split into two phases:
  1. Movement
  2. Population Dynamics

# Case Study 1: Predator Prey Model

- <u>Time Step Dynamics:</u>
  A time step is split into two phases:
  1. Movement
  2. Population Dynamics

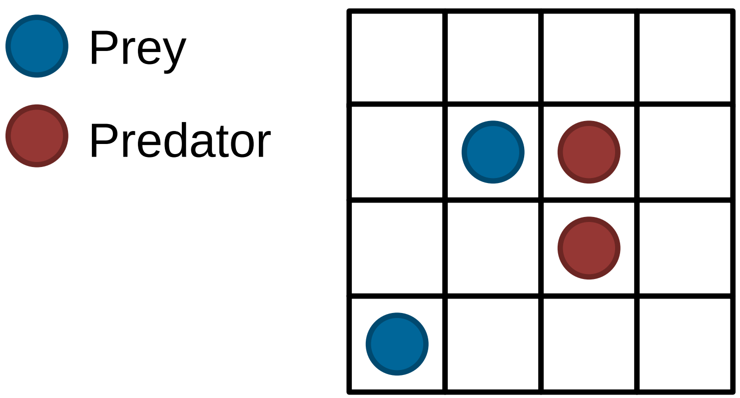



- Every agent moves in a randomly picked neighbour cell (Moore neighbourhood)

# Case Study 1: Predator Prey Model



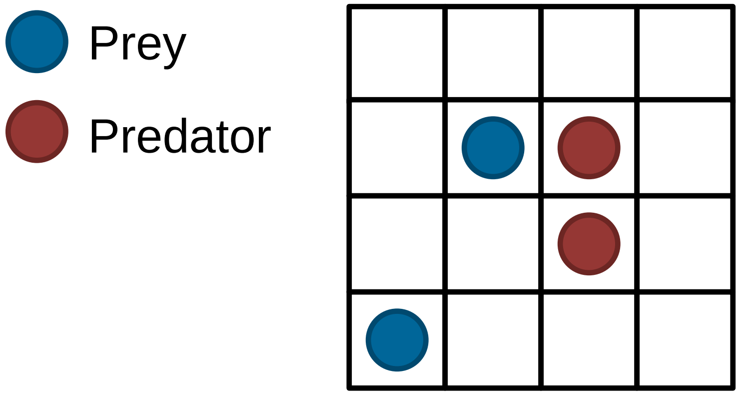
- <u>Time Step Dynamics:</u>
  A time step is split into two phases:
  1. Movement
  2. Population Dynamics




- Every agent moves in a randomly picked neighbour cell (Moore neighbourhood)
- Iterate in random order, periodic boundary conditions

# Case Study 1: Predator Prey Model

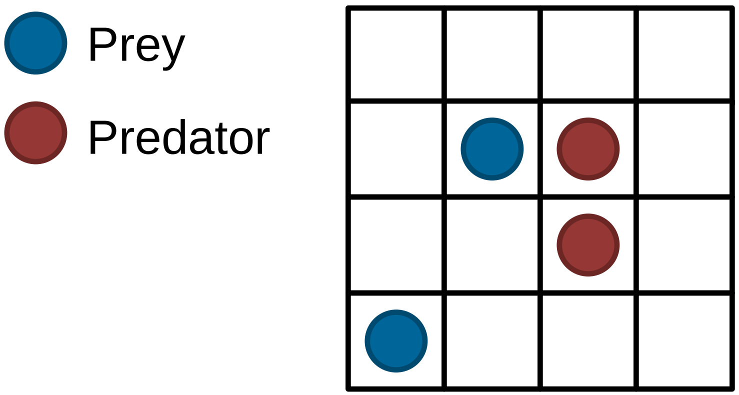
- <u>Time Step Dynamics:</u>
  A time step is split into two phases:
  1. Movement
  2. Population Dynamics




- Every agent moves in a randomly picked neighbour cell (Moore neighbourhood)
- Iterate in random order, periodic boundary conditions

# Case Study 1: Predator Prey Model

- <u>Time Step Dynamics:</u>
  A time step is split into two phases:
  1. Movement
  2. Population Dynamics



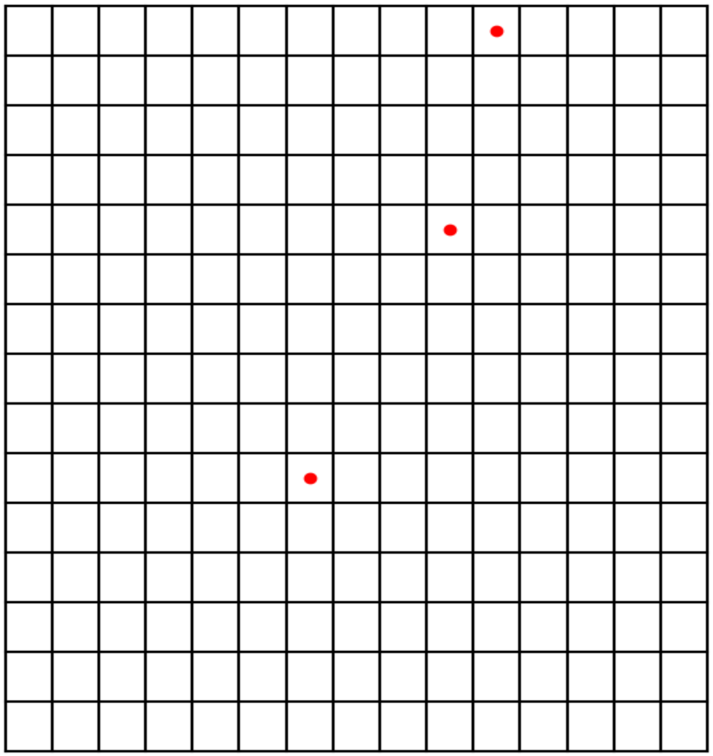

Every time-step agents are iterated in random order:

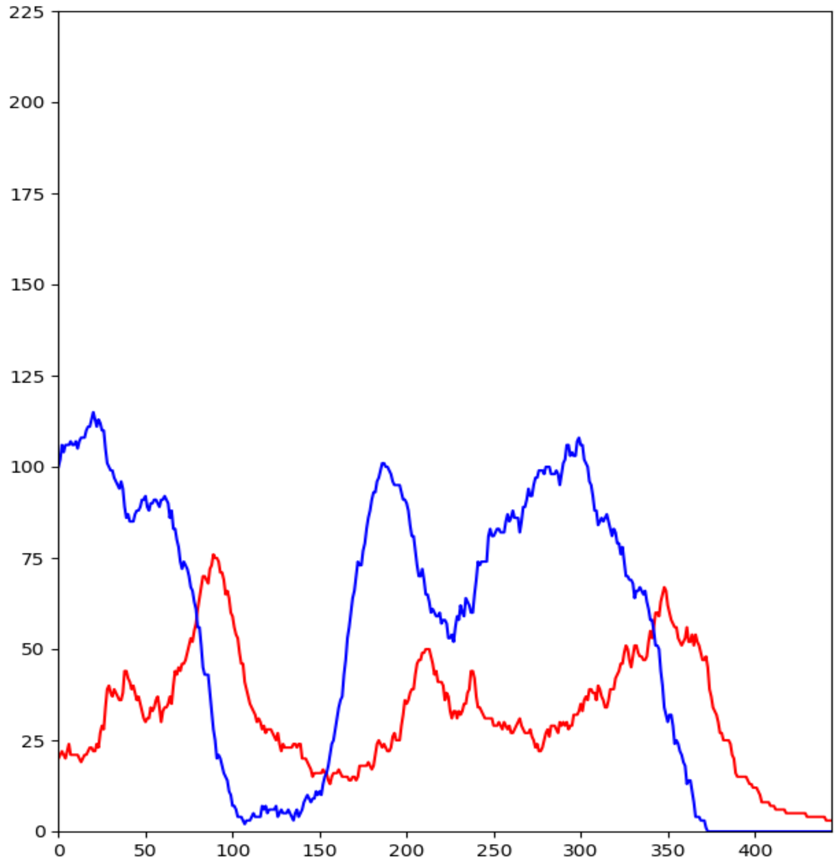
**Predator:**
- Every predator dies with probability $\alpha$
- If prey is around (Moore), the predator successfully catches one of it with probability $\beta$ and „replaces“ it by one offspring

**Prey:**
- If possible, every prey produces an offspring in one randomly picked neighbour cell with probability $\gamma$

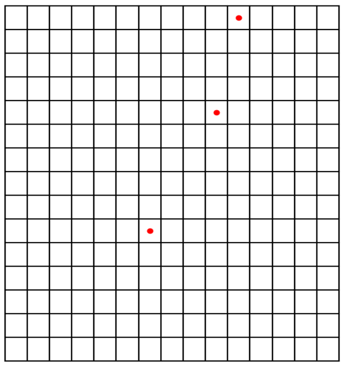
# Case Study 1: Predator Prey Model

Mathematical Modelling and Simulation

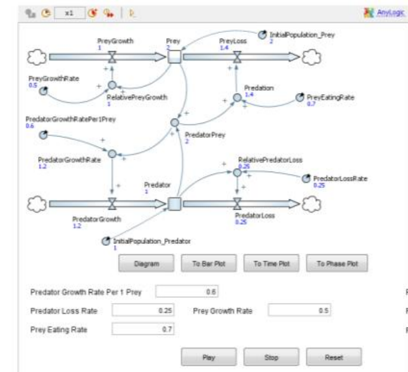


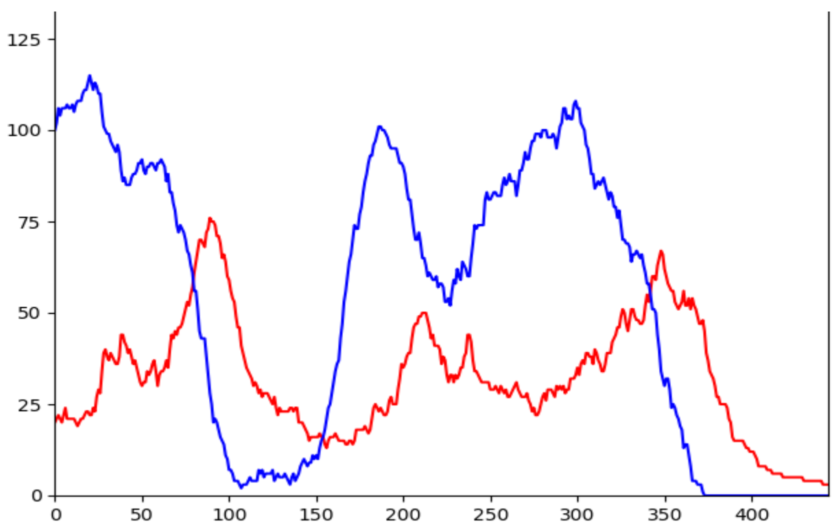


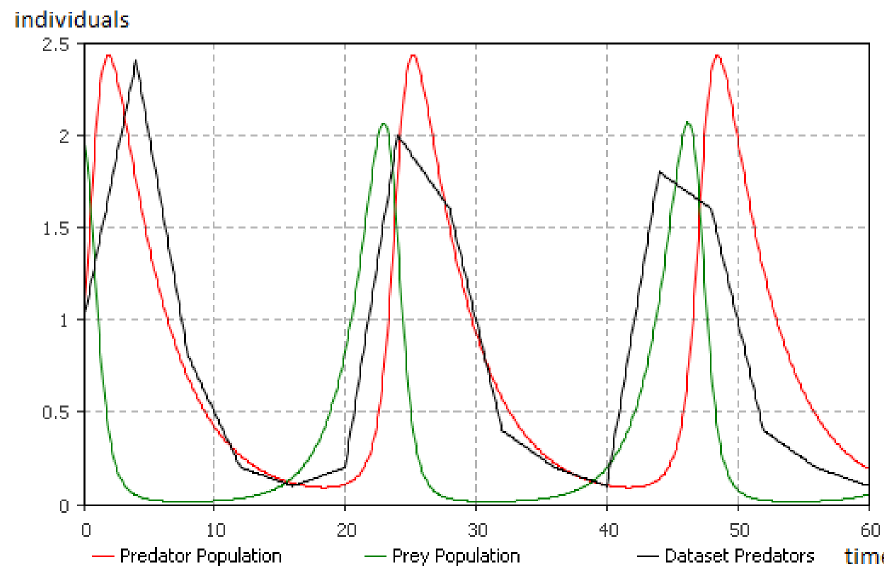
# Case Study 1: Predator Prey Model



Compare ABM model results with SD model results

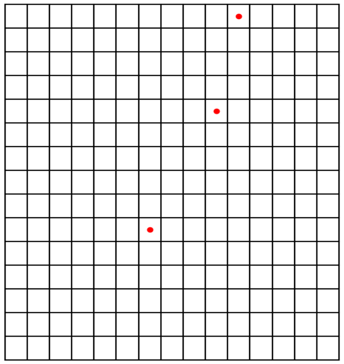




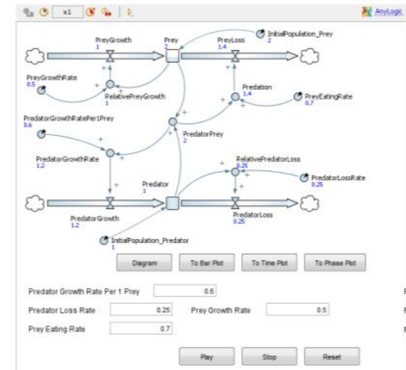



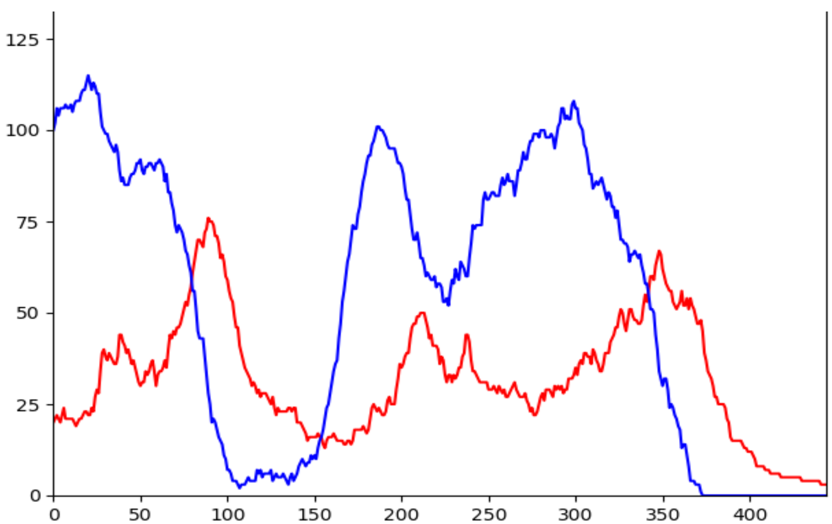
# Case Study 1: Predator Prey Model

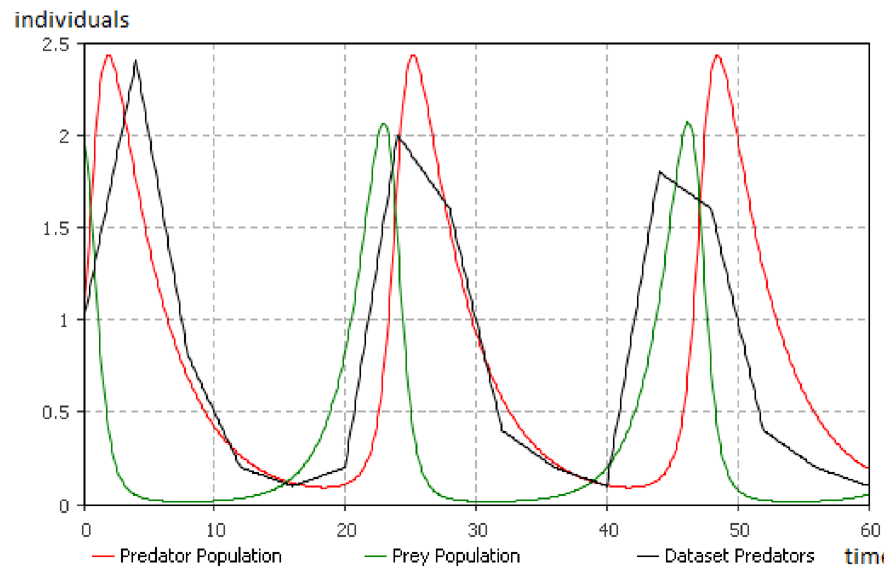




- Fuzzy (randomness)
- Dying out
- Scale

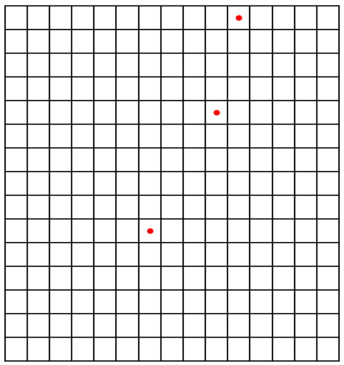




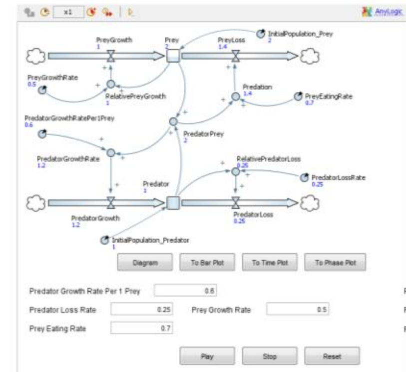



# Case Study 1: Predator Prey Model

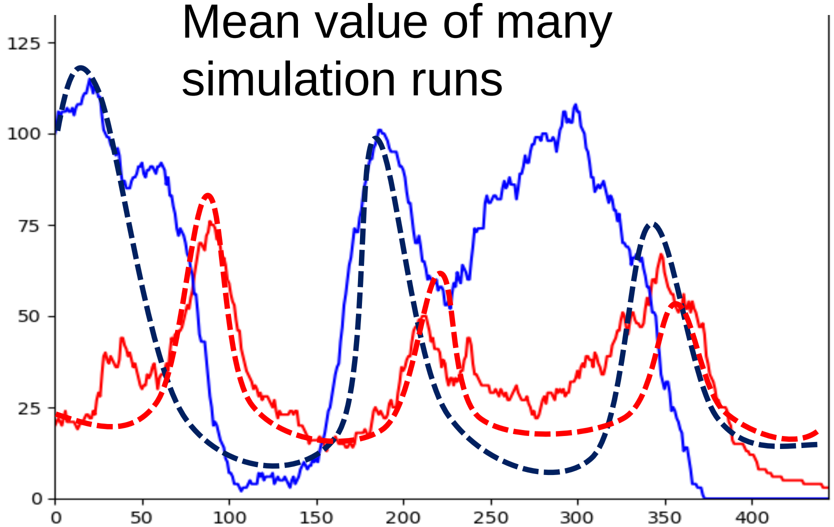
Mathematical Modelling and Simulation

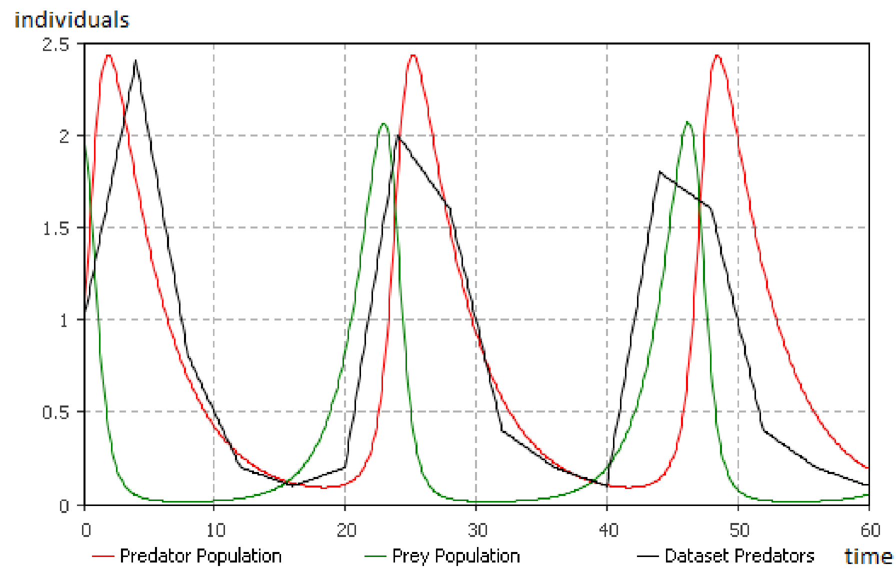


- Fuzzy (randomness)
- Dying out
- Scale



Mean value of many simulation runs



# Case Study 1: Lessons Learned



**Lesson 1:** Be careful, in which sequence/order agents are addressed to perform actions. Don‘t unintentionally favour some!

# Case Study 1: Lessons Learned

Mathematical Modelling and Simulation

**Lesson 1:** Be careful, in which sequence/order agents are addressed to perform actions. Don‘t unintentionally favour some!

**Lesson 2:** Be careful, when implementing movement on a grid. Don‘t occupy spots twice!

# Case Study 1: Lessons Learned

Mathematical Modelling and Simulation

**Lesson 1:** Be careful, in which sequence/order agents are addressed to perform actions. Don‘t unintentially favour some!

**Lesson 2:** Be careful, when implementing movement on a grid. Don‘t occupy spots twice!

**Lesson 3:** Never judge only based on only one simulation result, if randomness is involved!

# Case Study 2: Boids Flock Model

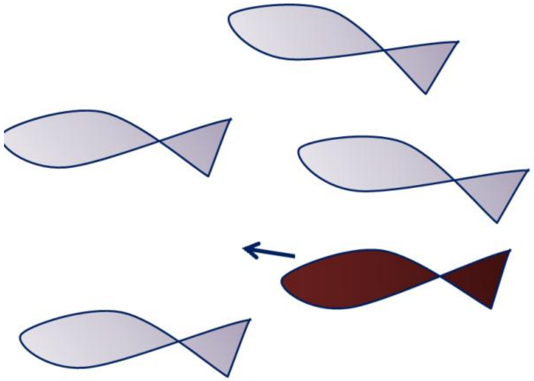
- Model for simulation of (bird) flocking behaviour
- Craig Reynolds in 1986

- Three simple rules on individual level lead to complex behaviour of the crowd

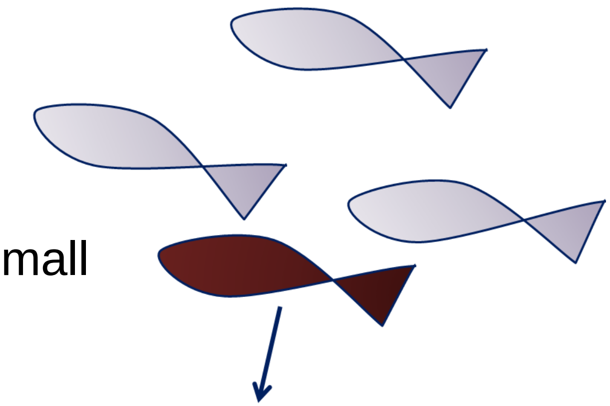
https://www.youtube.com/watch?v=QOGCSBh3kmM

# Case Study 2: Boids Flock Model

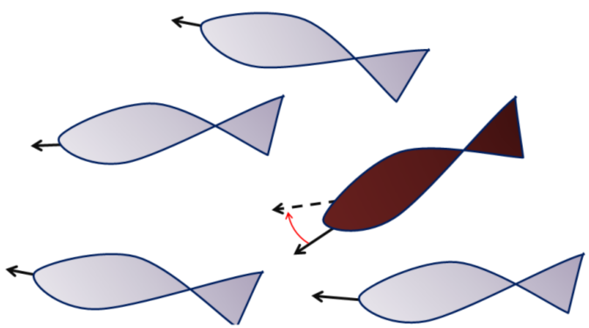
Mathematical Modelling and Simulation



Each agent tends towards the centre of its neighbours

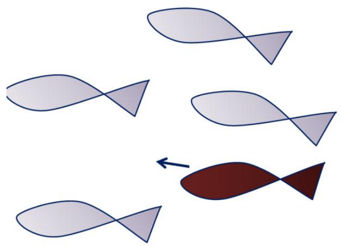


Keep a distance that is neither too far nor too small

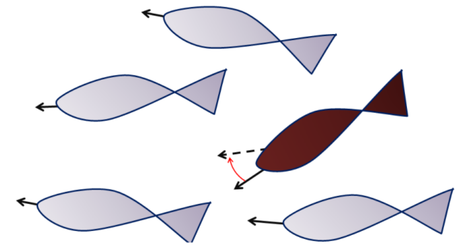


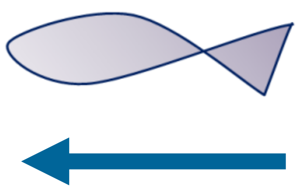
Swim in the same direction as your neighbours

# Case Study 2: Boids Flock Model



- $a_k$ current position of agent $k$
- $v_k$ current velocity of agent $k$

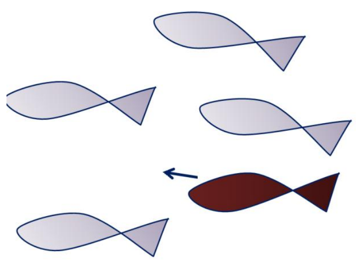




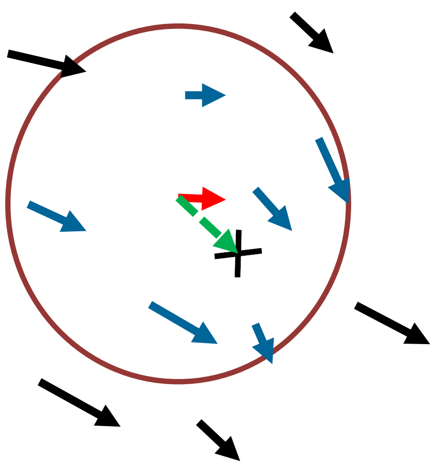

- $a_k$ current position of agent $k$
- $v_k$ current velocity of agent $k$

# Case Study 2: Boids Flock Model

Mathematical Modelling and Simulation



Each agent tends towards the centre of its neighbours



Let $a_k$ be the position of agent $k$ and let

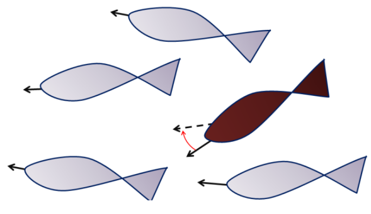
$I := \{k \neq i : ||a_k - a_i|| < Or\}$

for and observation radius $Or$.

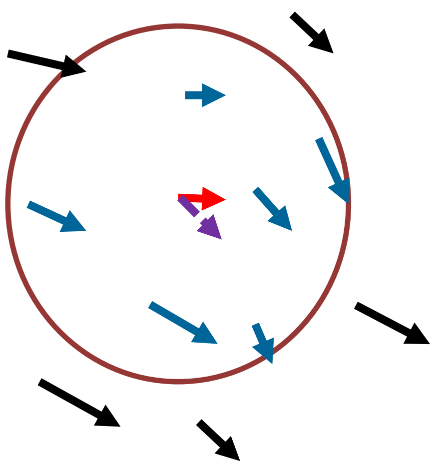
Then:

$$d_1^k = \frac{1}{|I|}\sum_{i \in I} a_i - a_k$$

# Case Study 2: Boids Flock Model



Swim in the same direction as your neighbours

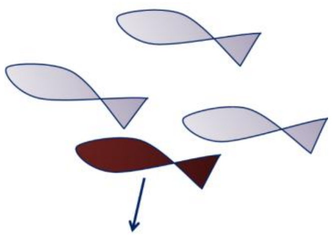


Let $v_k$ be the velocity of agent $k$ and let
$I := \{k \neq i : \left\| a_k - a_i \right\| < Or\}$
for and observation radius $Or$.
Then:

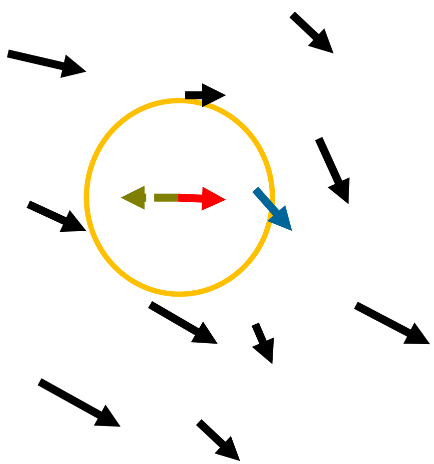
$$d_2^k = \frac{1}{|I|} \sum_{i \in I} v_i$$

# Case Study 2: Boids Flock Model

Mathematical Modelling and Simulation



Keep a distance that is neither too far nor too small

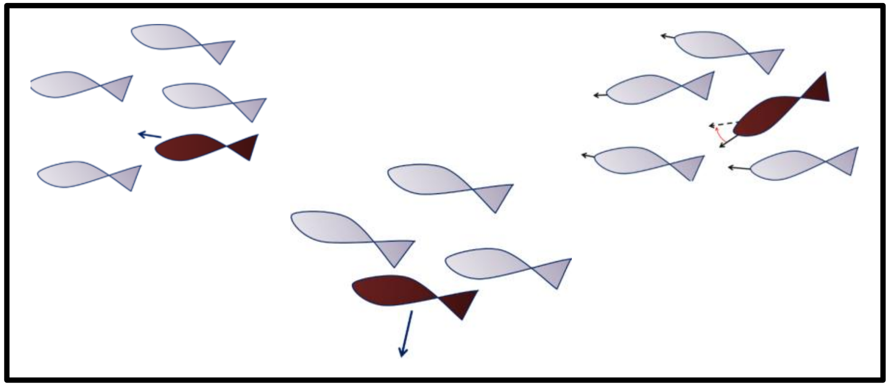


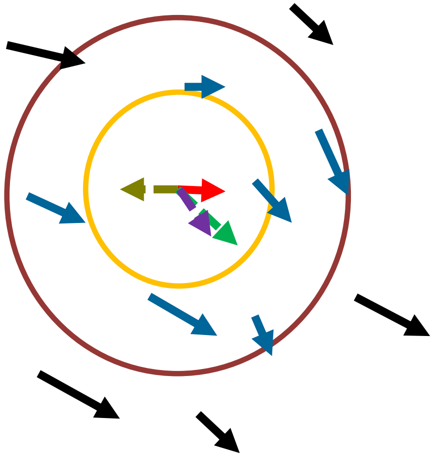
Let $a_k$ be the position of agent $k$ and let
$J := \{k \neq i : \left\|a_k - a_i\right\| < Cr\}$
for and collision radius $Cr$.
Then:

$$d_3^k = a_k - \frac{1}{|J|} \sum_{i \in J} a_i$$

# Case Study 2: Boids Flock Model

Mathematical Modelling and Simulation





Update velocity

$$\tilde{v}_i = \alpha_0 v_i + \alpha_1 d_1^i + \alpha_2 d_2^i + \alpha_3 d_3^i$$

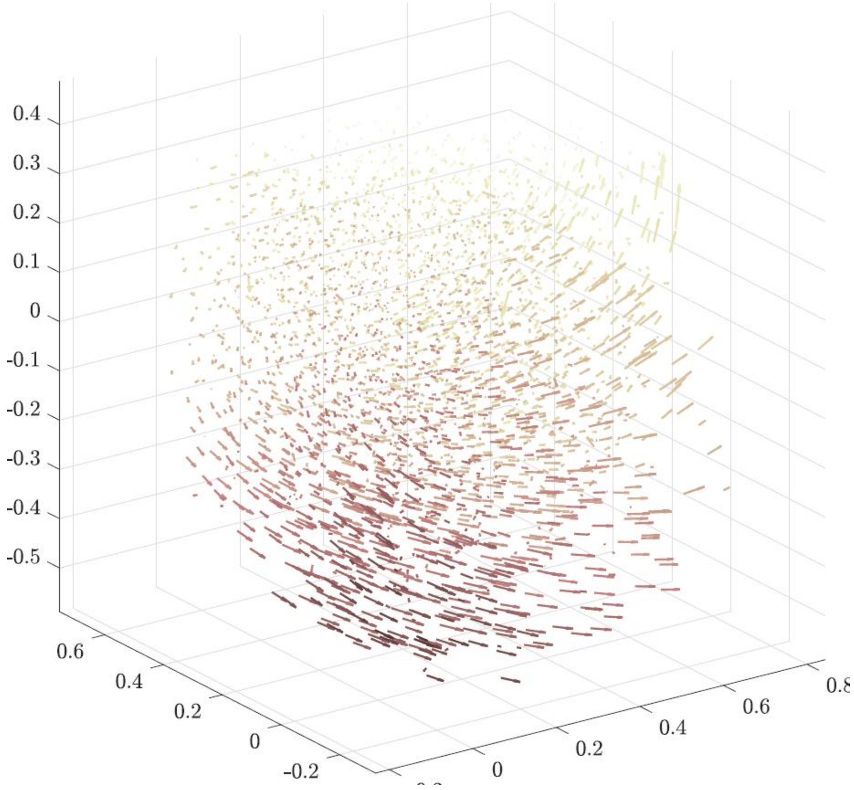
Update position

$$\tilde{x}_i = x_i + \tilde{v}_i$$

# Case Study 2: Boids Flock Model

Mathematical Modelling and Simulation

- Boids model is the most picturesque example for emergence in ABMs
- It is a good test case for agent-based simulators (high computation performance required)



# Case Study 2: Lessons Learned



**Lesson 4: ABMs  are often computationally expensive! Think about, how to optimize your code performance**

# Case Study 2: Lessons Learned



**Lesson 4: ABMs  are often computationally expensive! Think about, how to optimize your code performance**

**Lesson 5: A fully reproducible model description is difficult and can be long and cconfusing. Think about using a standartised protocol for it.**

# Case Study 2: Lessons Learned

Mathematical Modelling and Simulation

- Example: ODD Protocol by Volker Grimm et.al.
- Standartised documentation of agent-based models

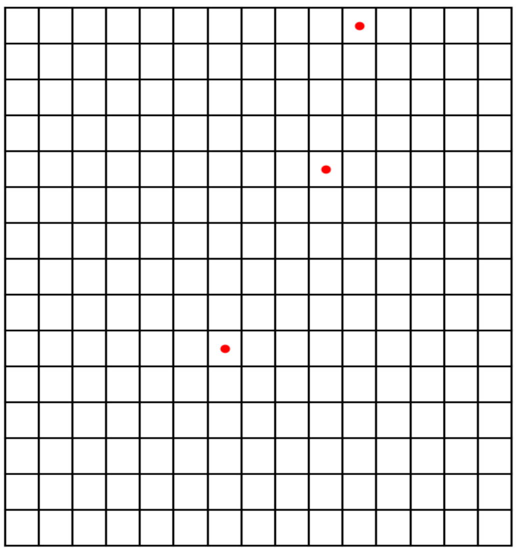
| | |
|---|---|
| **Overview** | **Purpose** |
| | **State variables and scales** |
| | **Process overview and scheduling** |
| **Design concepts** | **Design concepts** |
| **Details** | **Initialization** |
| | **Input** |
| | **Submodels** |

**Fig. 1 – The seven elements of the ODD protocol, which can be grouped into the three blocks: Overview, Design concepts, and Details.**

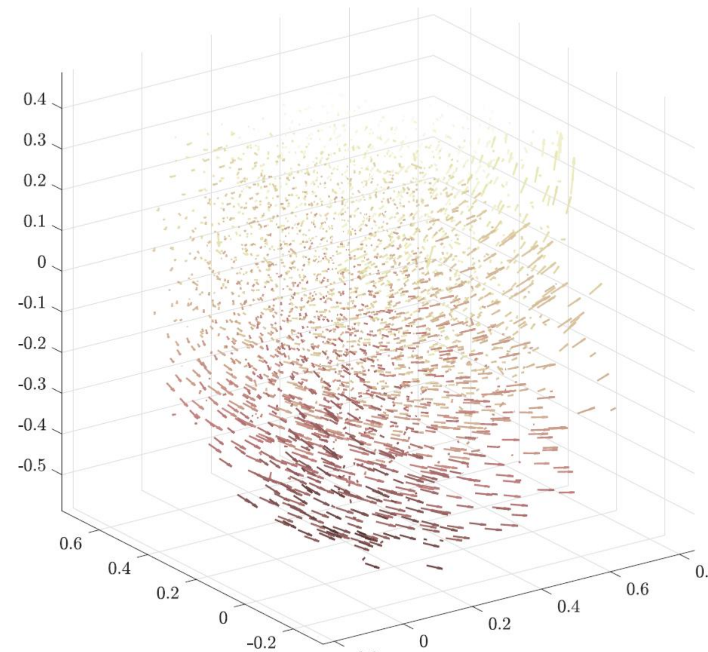
Grimm, Volker, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, u. a. „A Standard Protocol for Describing Individual-Based and Agent-Based Models“. *Ecological Modelling* 198, Nr. 1–2 (September 2006): 115–26. https://doi.org/10.1016/j.ecolmodel.2006.04.023.

# Classification of ABMs (1)

Differences?



vs.



# Classification of ABMs (1)

Mathematical Modelling and Simulation

| Classification 1 | |
|---|---|
| with respect to <u>modelling purpose</u> (i.e. the research question) | |
| ABMs for <u>qualitative</u> investigation | ABMs for <u>quantitative</u> investigation |
| • (On purpose) very abstract<br>• Usually very complex model behaviour<br>• Hardly any parameters identified with real data | • Rather simple agent interactions<br>• A lot of data involved for model parametrisation and validation<br>• Usually less famous |

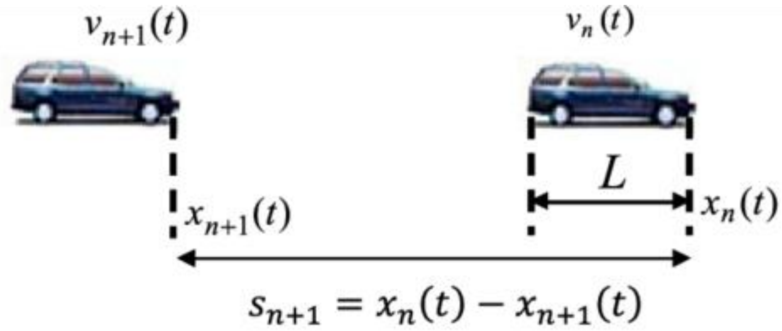
# Classification of ABMs (2)

Mathematical Modelling and Simulation

| Classification 2 | | | |
|---|---|---|---|
| with respect to agent <u>**environment**</u> | | | |
| <u>**spatial**</u> environment | | <u>**abstract**</u> environment | |
| <u>**lattice**</u> | <u>**continuous**</u> | <u>**network**</u> | ... |
| • Sometimes equivalent to a CA<br>• Different forms of grids<br>• 1D – 3D | • Often uses distance-metrics for agent interaction<br>• Surprisingly, often easier to handle than lattice models | • Contacts between agents modelled as edges of a network | |

# Case Study 3: Gipps 's Car Following Model

Mathematical Modelling and Simulation

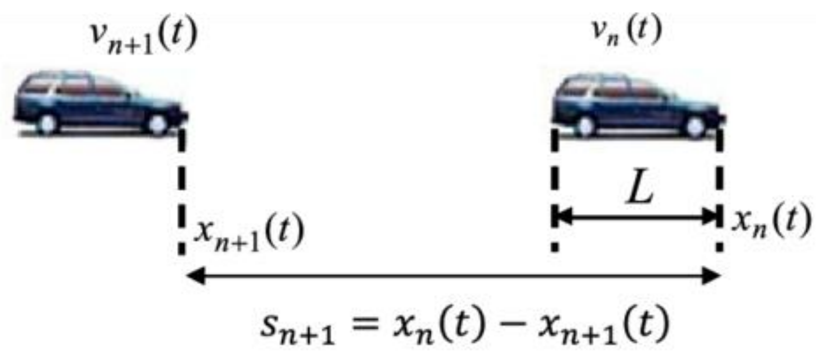



- Each car in a one-lane road is represented by an agent
- Each agent $i$ has a certain length $L_i$, position $x_i(t)$ and velocity $v_i(t)$
- Velocity update is based on a differential equation that includes the distance to and velocity of the car in front

# Case Study 3: Gipps 's Car Following Model

Mathematical Modelling and Simulation

- Each car in a one-lane road is represented by an agent
- Each agent $i$ has a certain length $L_i$, position $x_i(t)$ and velocity $v_i(t)$
- Velocity update is based on a differential equation that includes the distance to and velocity of the car in front



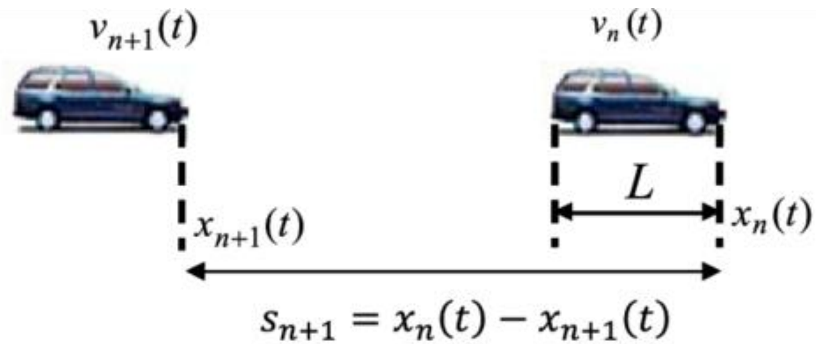

$$\dot{v}_{i+1}(t+\tau) = A \cdot \frac{v_i(t) - v_{i+1}(t)}{x_i(t) - x_{i+1}(t) - L_i}$$

$A$ ... acceleration constant
$\tau$ ... reaction time

# Case Study 3: Gipps ‘s Car Following Model



- Each car in a one-lane road is represented by an agent
- Each agent $i$ has a certain length $L_i$, position $x_i(t)$ and velocity $v_i(t)$
- Velocity update is based on a differential equation that includes the distance to and velocity of the car in front



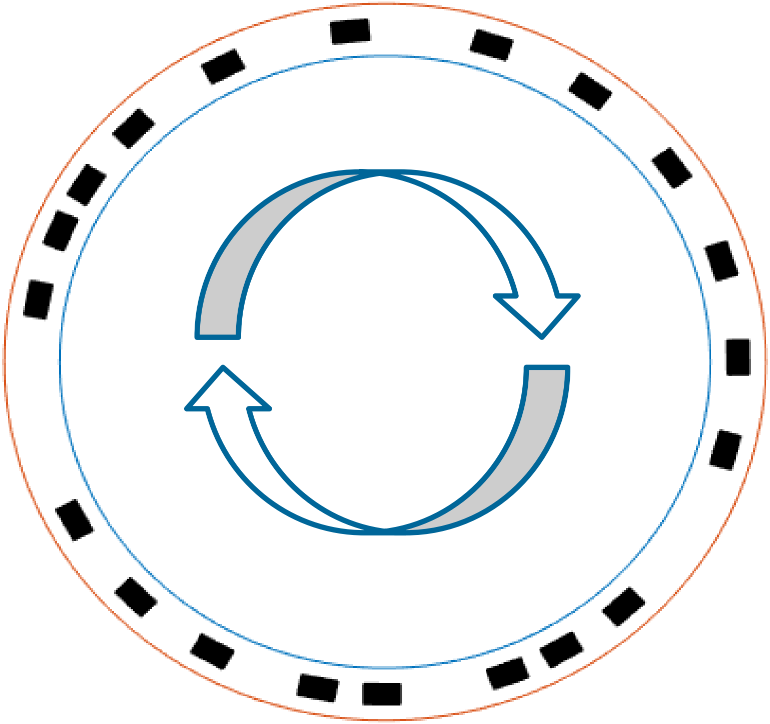

$$\dot{v}_{i+1}(t+\tau) = A \cdot \frac{v_i(t) - v_{i+1}(t)}{x_i(t) - x_{i+1}(t) - L_i}$$

$A$ ... acceleration constant
$\tau$ ... reaction time

Some additional parameters:
lingering, maximum velocity, maximum acceleration, maximum break force, length of the road

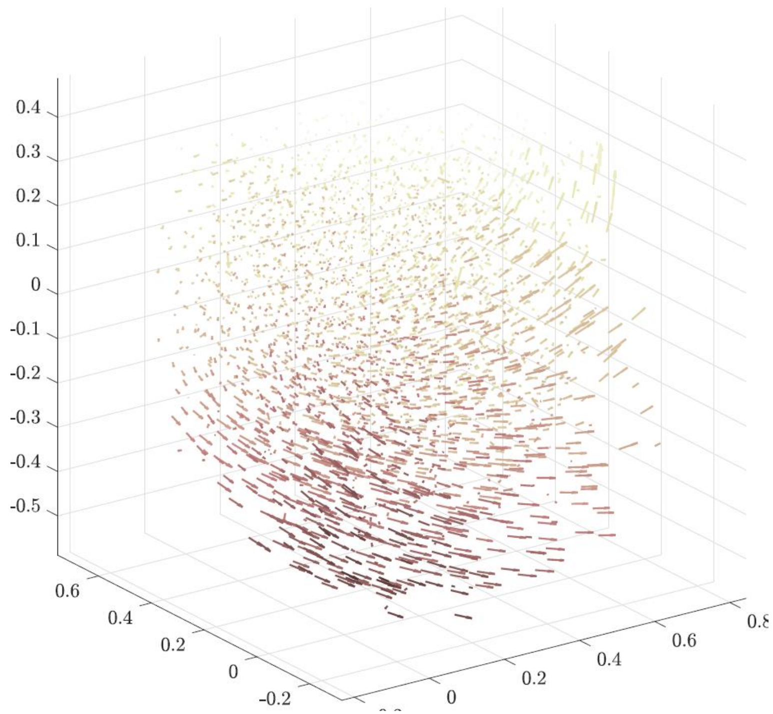
# Case Study 3: Gipps ‘s Car Following Model

- Gipps model poses the base for most modern models for traffic flow
- Alternative approaches: Nagel-Schreckenberg Model, Burgers equation
- Extensions to: multiple lanes, junctions, traffic lights, ...

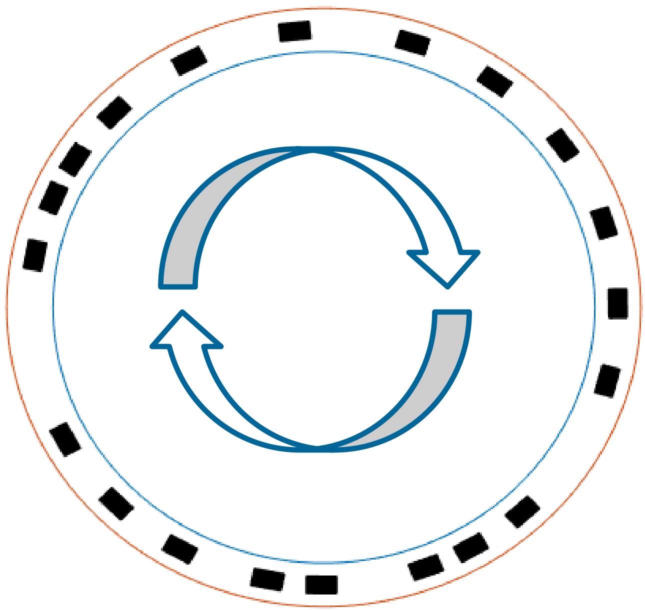


# Case Study 3: Gipps 's Car Following Model

Differences?




vs.



# Classification of ABMs (3)

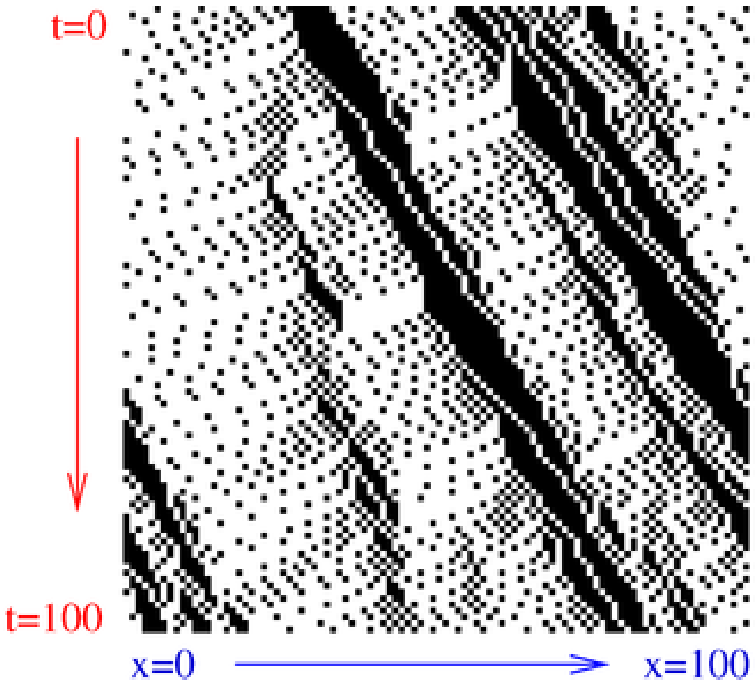
Mathematical Modelling and Simulation

<table>
<tr><th colspan="3">Classification 3</th></tr>
<tr><td colspan="3">with respect to <u>time update</u></td></tr>
<tr><td colspan="2"><u>time continuous</u></td><td><u>time discrete</u></td></tr>
<tr><td><u>differential equation</u></td><td><u>event-based</u></td><td><u>time steps</u></td></tr>
<tr><td>• Usually used for systems with physical laws</td><td>• Often used for scheduling problems</td><td>• Most common update strategy.<br>• Needs special care with events happening at the same time</td></tr>
</table>

# Case Study 4: Nagel Schreckenberg Model

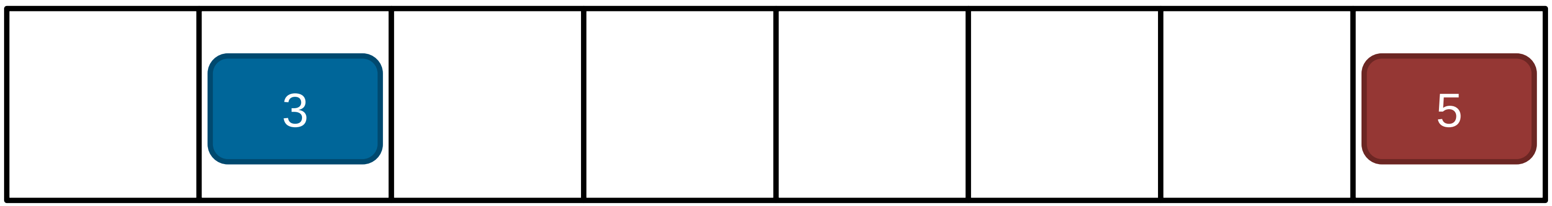
Mathematical Modelling and Simulation

- 1992, Kai Nagel and Michael Schreckenberg
- Same Purpose as Gipps Model
- Discrete 1D Grid insread of continuous road
- One car per grid point



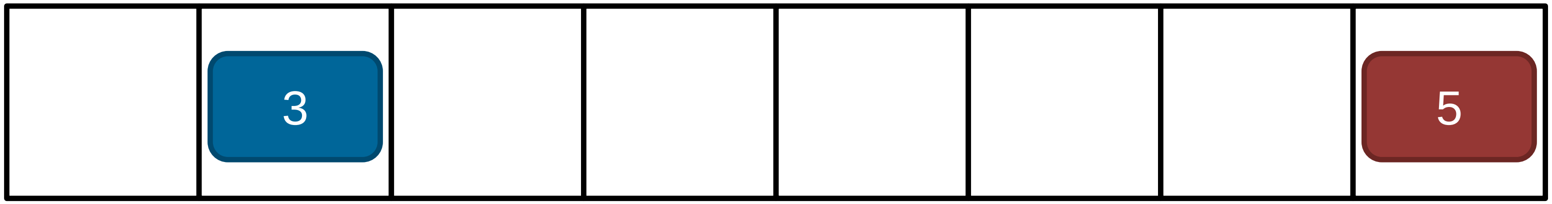
# Case Study 4: Nagel Schreckenberg Model






- Agents enter the model from the left (at the left-most cell)
- Each agent has a certain velocity (natural number)
- Model is updated with equidistant time-steps

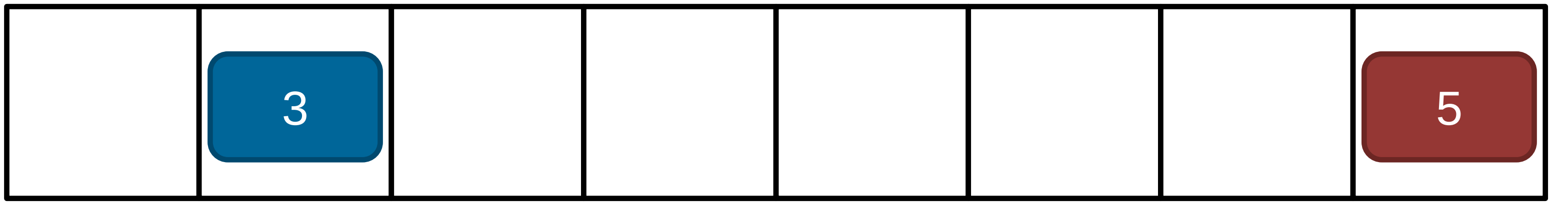
# Case Study 4: Nagel Schreckenberg Model




Each time-step, each agent...

- Updates its velocity according to the car in front (if any)
- Drives that many cells to the right

# Case Study 4: Nagel Schreckenberg Model

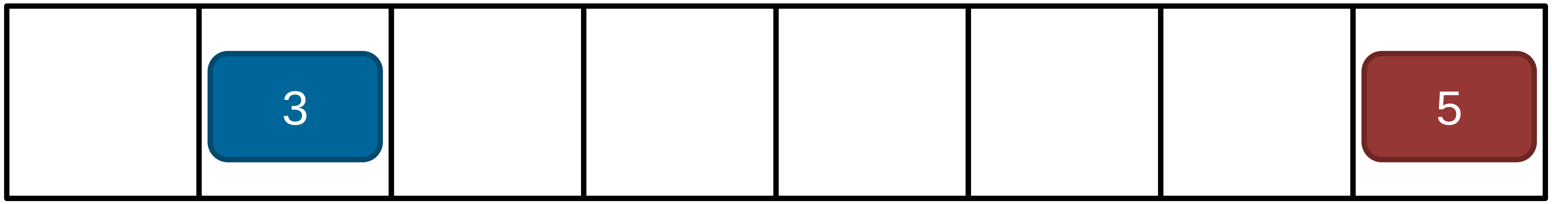



Each time-step, each agent…

- Updates its velocity according to the car in front (if any)
  1. Increases its velocity $v$ by one: $v \leftarrow v + 1$
  2. Checks how many cells to the right are empty (say $q$)
  3. If $v > q$, then $v \leftarrow q$
  4. With a certain probability: $v \leftarrow \max(v - 1{,}0)$

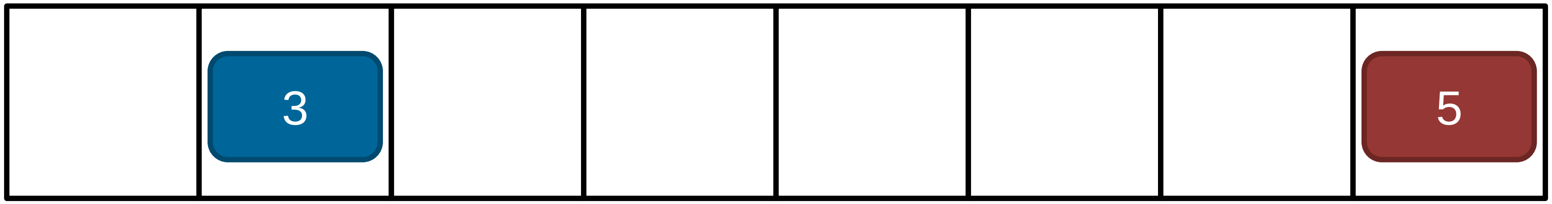
# Case Study 4: Nagel Schreckenberg Model






Each time-step, each agent…
- Updates its velocity according to the car in front (if any)
- Drives that many cells to the right
  1. agent advances $v$ cells

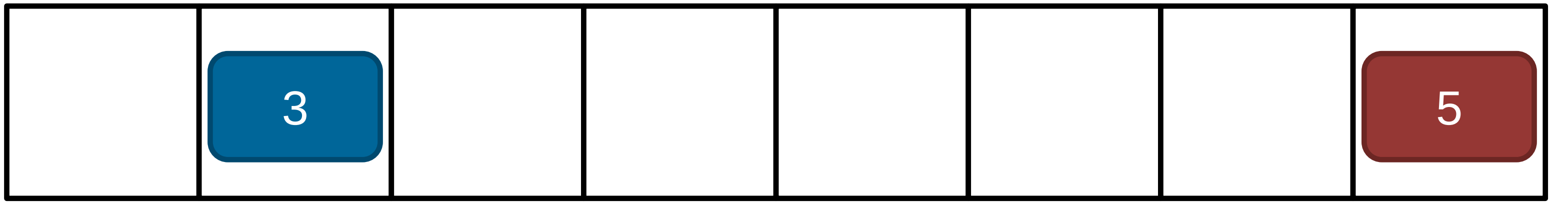
# Case Study 4: Nagel Schreckenberg Model



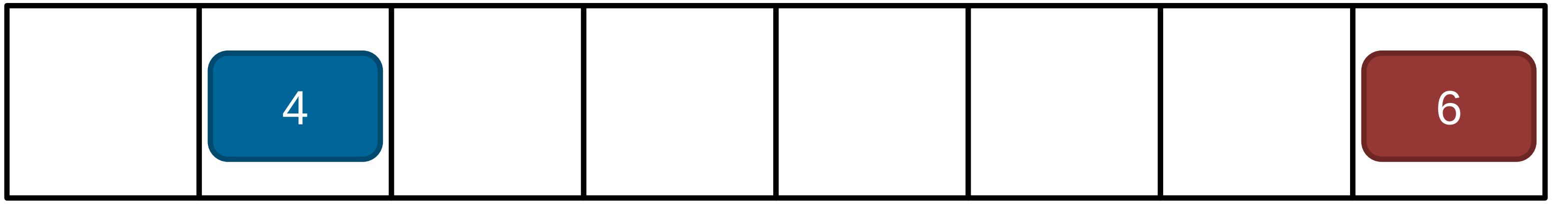

Increases its velocity $v$ by one:

$v \leftarrow v + 1$

# Case Study 4: Nagel Schreckenberg Model

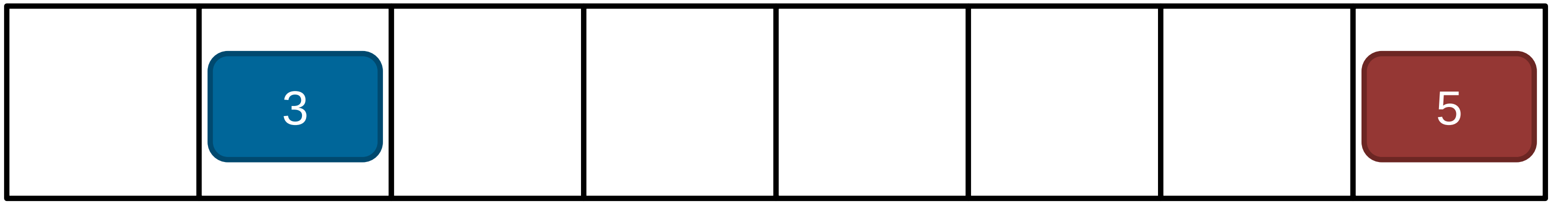



Increases its velocity $v$ by one:
$v \leftarrow v + 1$

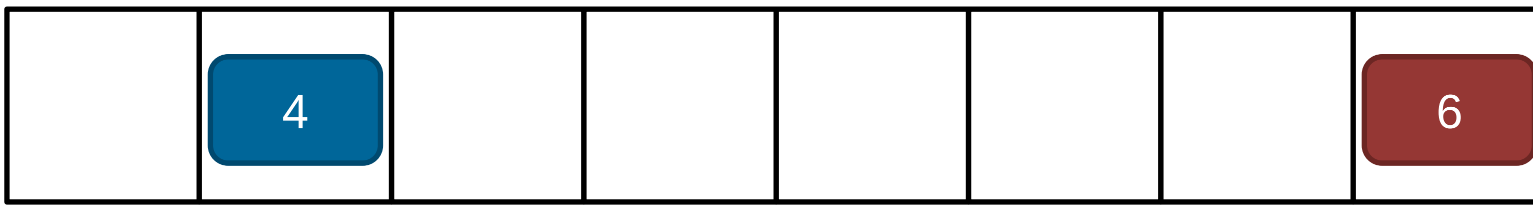



Checks how many cells to the right are empty (say $q$)

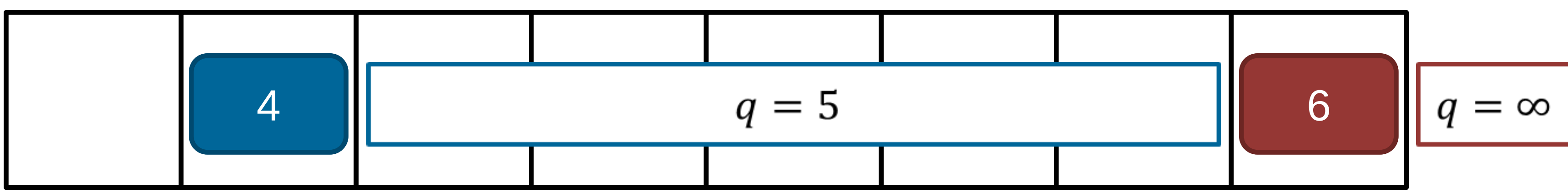
# Case Study 4: Nagel Schreckenberg Model



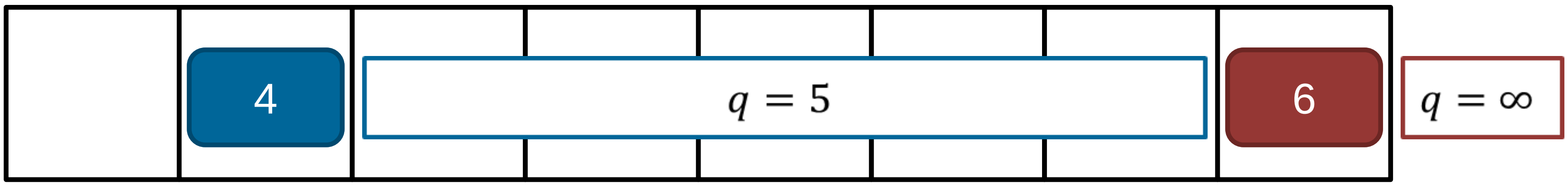

Increases its velocity $v$ by one:
$v \leftarrow v + 1$




Checks how many cells to the right are empty (say $q$)



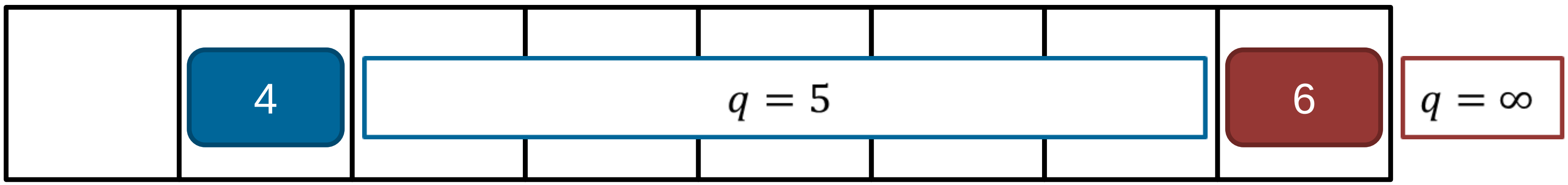
# Case Study 4: Nagel Schreckenberg Model



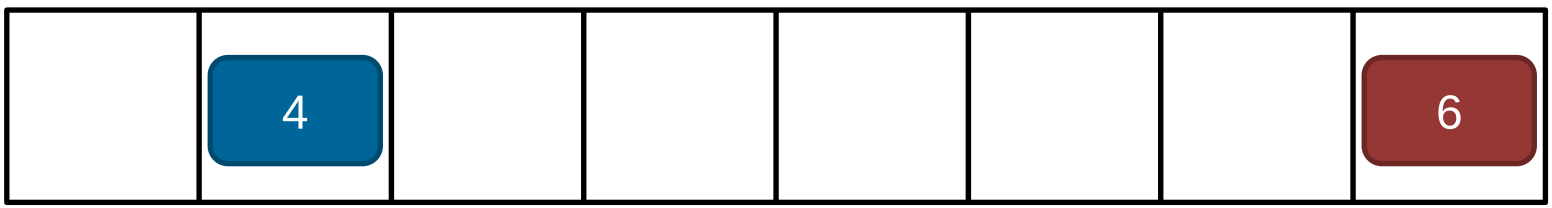

If $v > q$, then $v \leftarrow q$

# Case Study 4: Nagel Schreckenberg Model





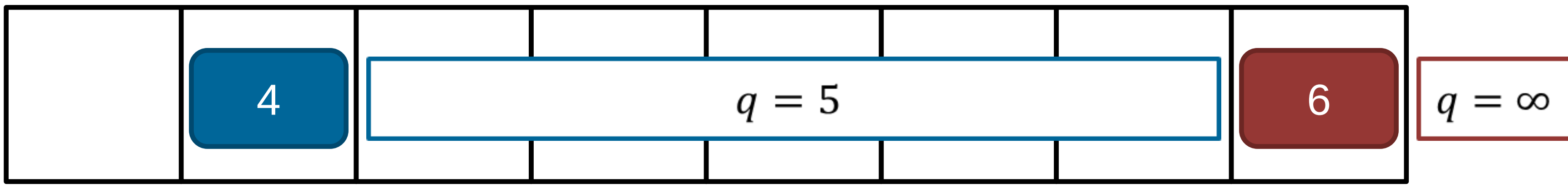

If $v > q$, then $v \leftarrow q$



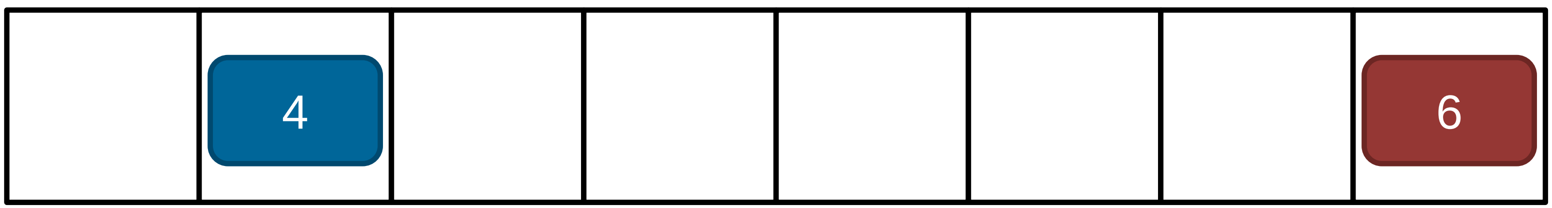

With a certain probability: $v \leftarrow \max(v - 1{,}0)$

# Case Study 4: Nagel Schreckenberg Model

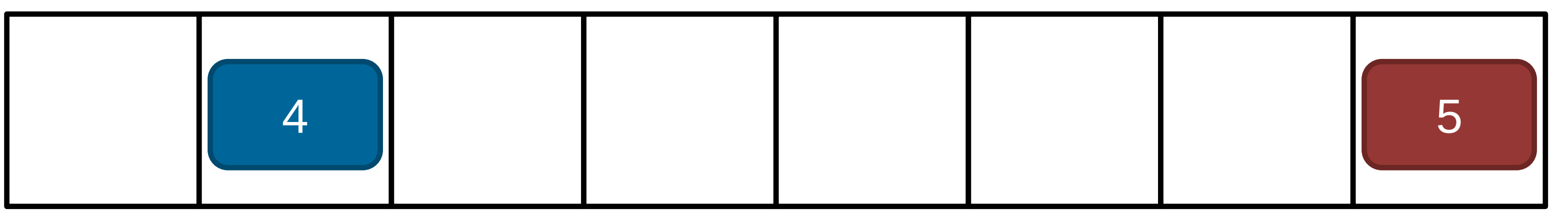





If $v > q$, then $v \leftarrow q$

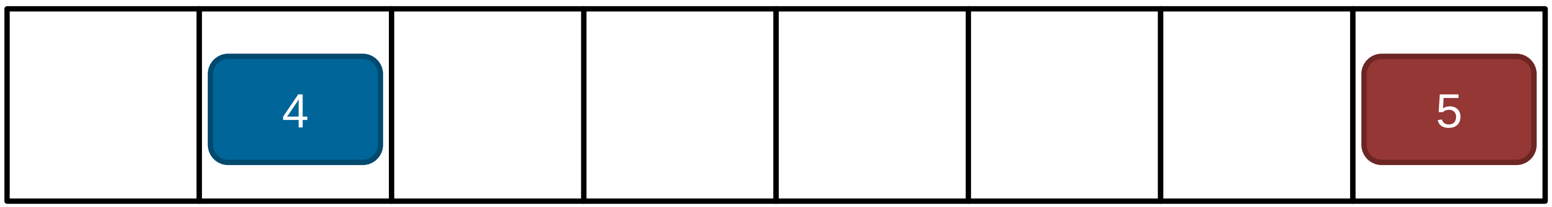



With a certain probability: $v \leftarrow \max(v - 1, 0)$

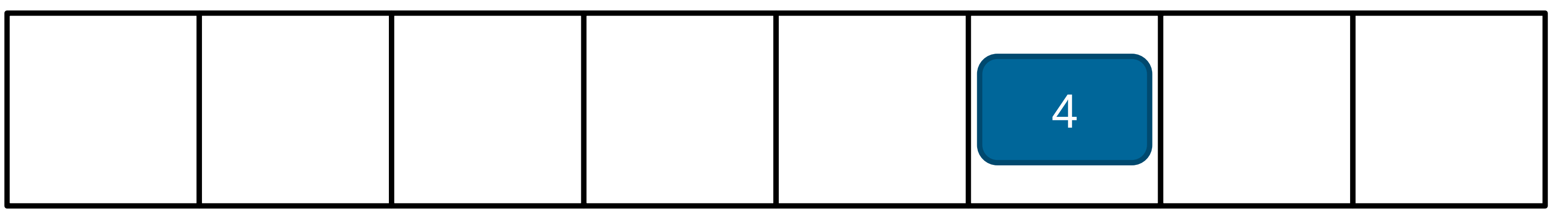


# Case Study 4: Nagel Schreckenberg Model



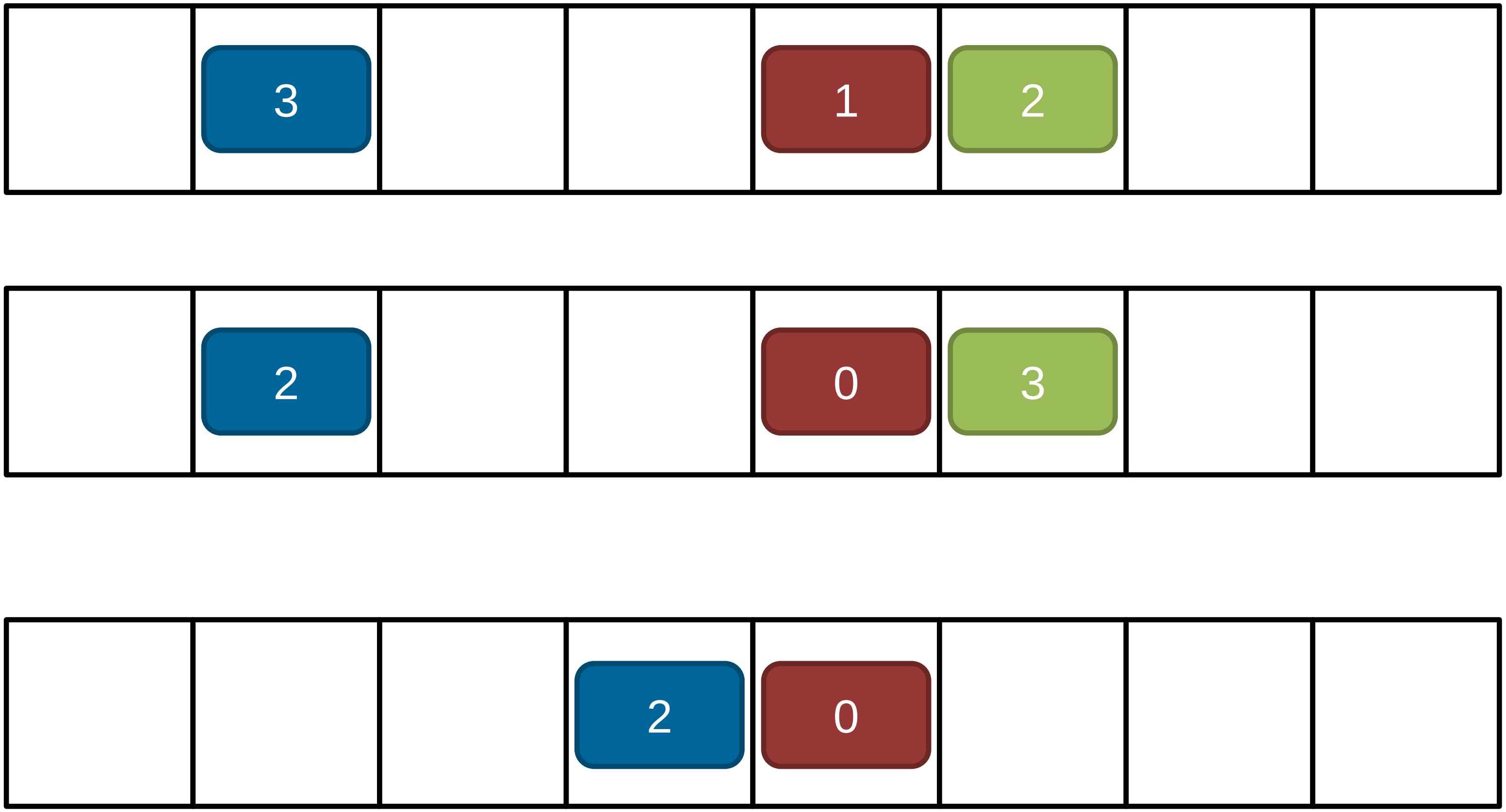



agent advances $v$ cells



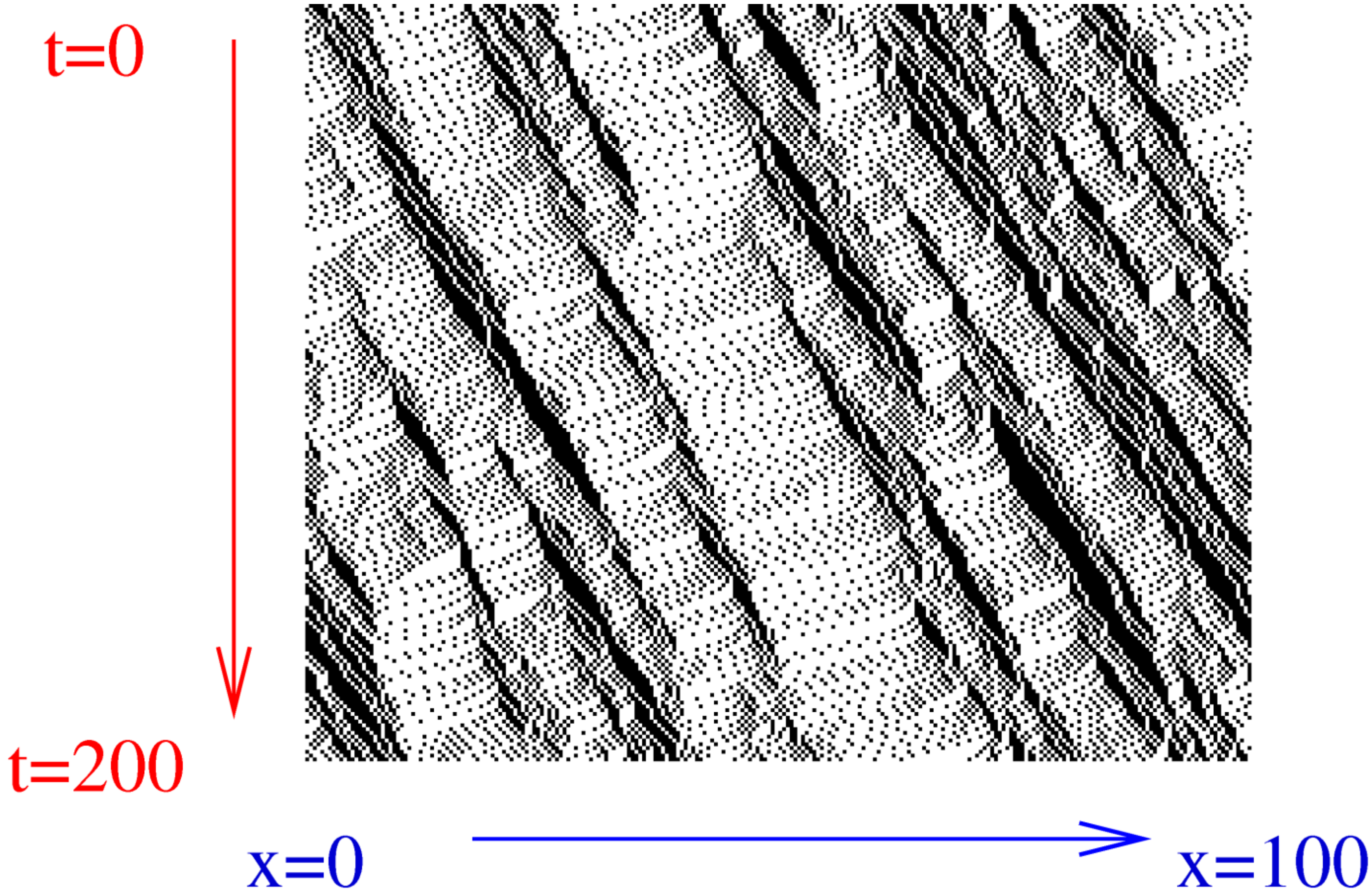
# Case Study 4: Nagel Schreckenberg Model





# Case Study 4: Nagel Schreckenberg Model

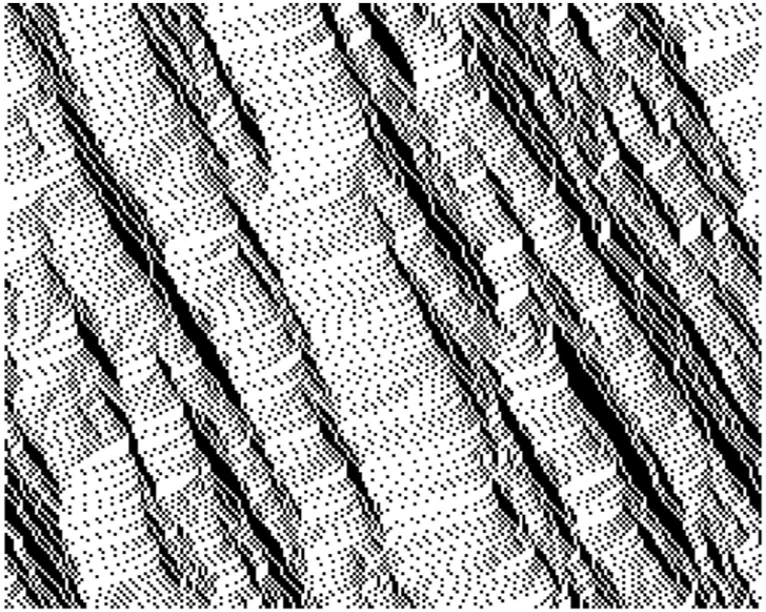




# Case Study 4: Nagel Schreckenberg Model

Mathematical Modelling and Simulation

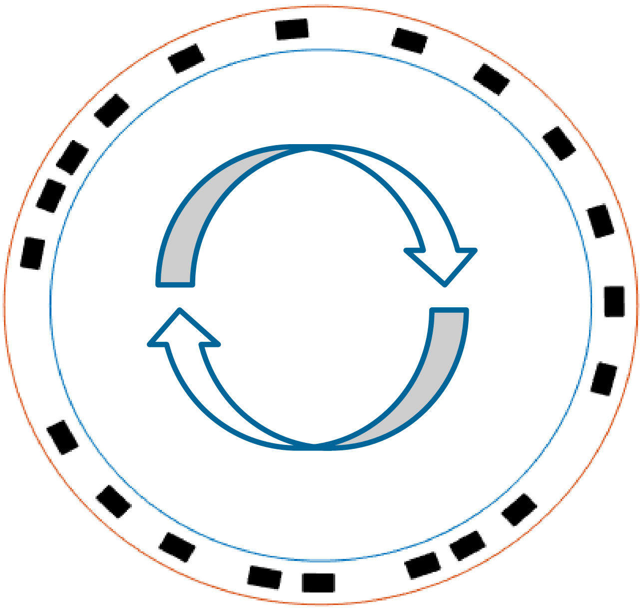
- Model usually described as a cellular automaton
- Model extendable to multiple lanes
- Either torodorial boundary conditions or new generation of cars every time-step



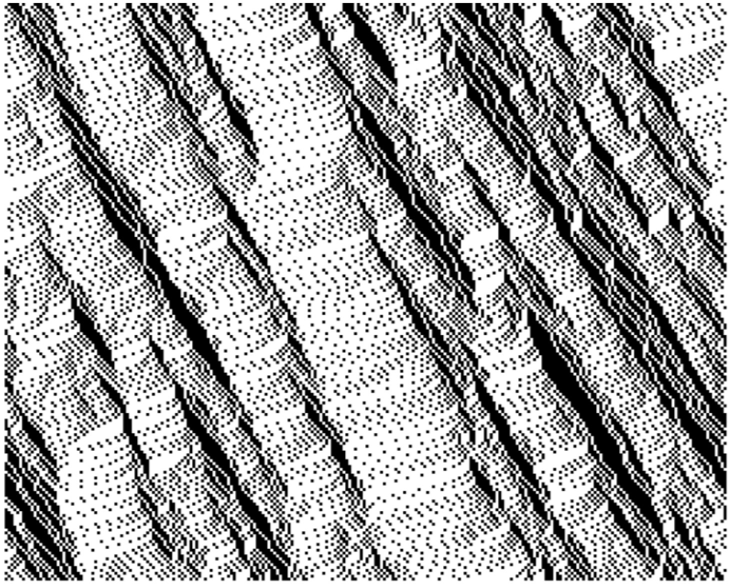
# Case Study 4: Nagel Schreckenberg Model

Mathematical Modelling and Simulation

Differences?
(apart from known)

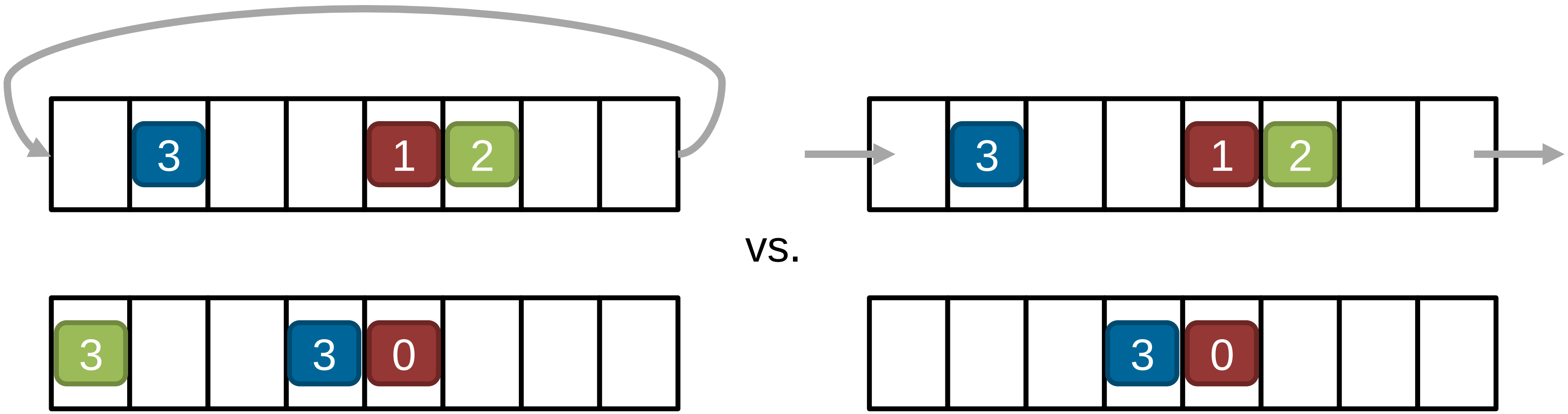


vs.



# Case Study 4: Nagel Schreckenberg Model

Differences?



# Classification of ABMs (4)

Mathematical Modelling and Simulation

| Classification 4 | |
|---|---|
| with respect to agent population | |
| **population static** | **population dynamic** |
| • agents only generated at the beginning of the simulation<br>• system variables only change due to change of agent states | • agents are (can be) generated on run-time<br>• system variables can also change due to change of number of agents<br>• usually more difficult to deal with due to space allocation of vectors |

# Case Study 4: Lessons Learned



**Lesson 6: Careful when implementing population dynamic agent based models: Removal and adding of elements to a list is usually expensive.**

**Consider, recreating the list every time instead of adding and removal!**

# Case Study 4: Lessons Learned



**Lesson 6: Careful when implementing population dynamic agent based models: Removal and adding of elements to a list is usually expensive.**

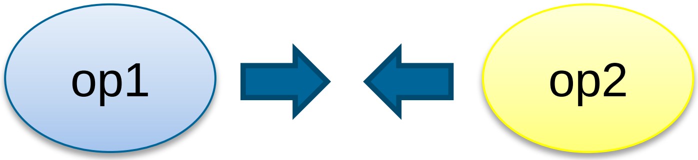
**Consider, recreating the list every time instead of adding and removal!**

**Lesson 7: Be aware, that agents need to be updated simultaneously when using models with time-steps!**

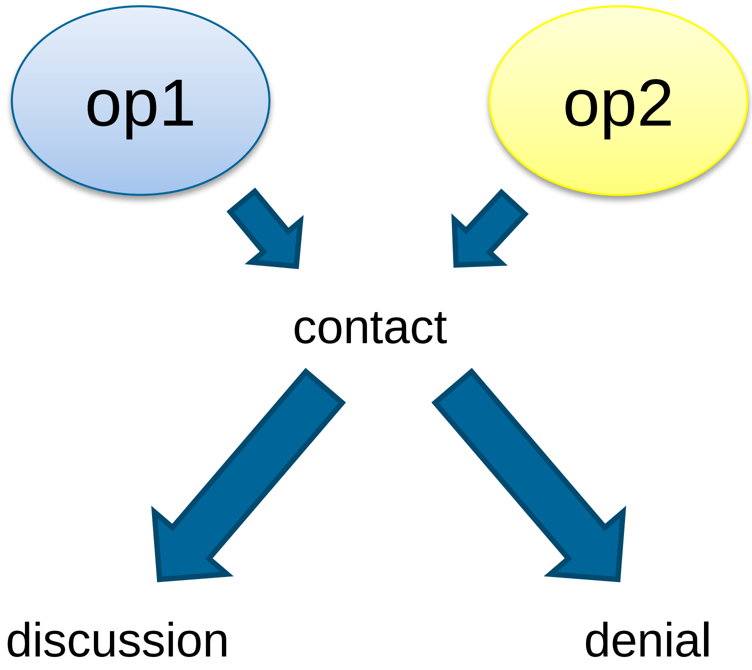
# Case Study 5: Opinion Dynamics

Mathematical Modelling and Simulation

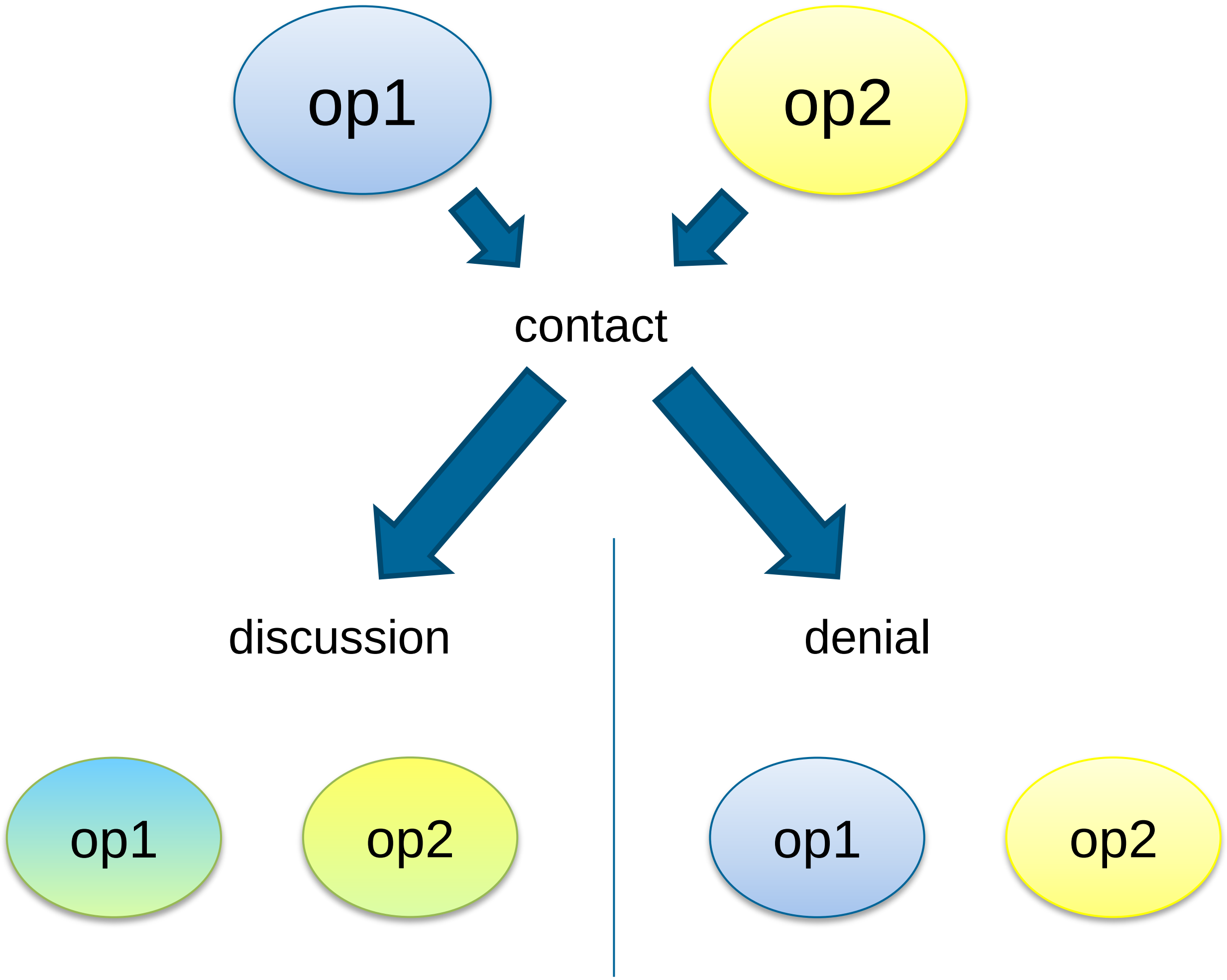
- Specific model from Guillaume Deffuant 2000 (basic concepts much older)
- Simple model that depicts spread and development of different opinions



# Case Study 5: Opinion Dynamics



# Case Study 5: Opinion Dynamics

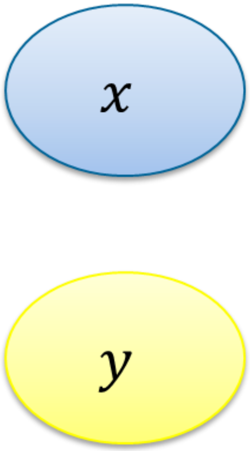


# Case Study 5: Opinion Dynamics



- $N$ agents are initialised
- Every agent is assigned an opinion $x \in [-1,1]$
- Every time step,
  - Split the population into two random but equivalent halves
  - pick $N/2$ random partners from both, say with opinions $x$ and $y$.
  - If $|x - y| < \tau$, the two start discussing and
    $$x \leftarrow x + \mu(y - x)$$
    $$y \leftarrow y - \mu(y - x)$$
  - Otherwise
    $$x \leftarrow x$$
    $$y \leftarrow y$$

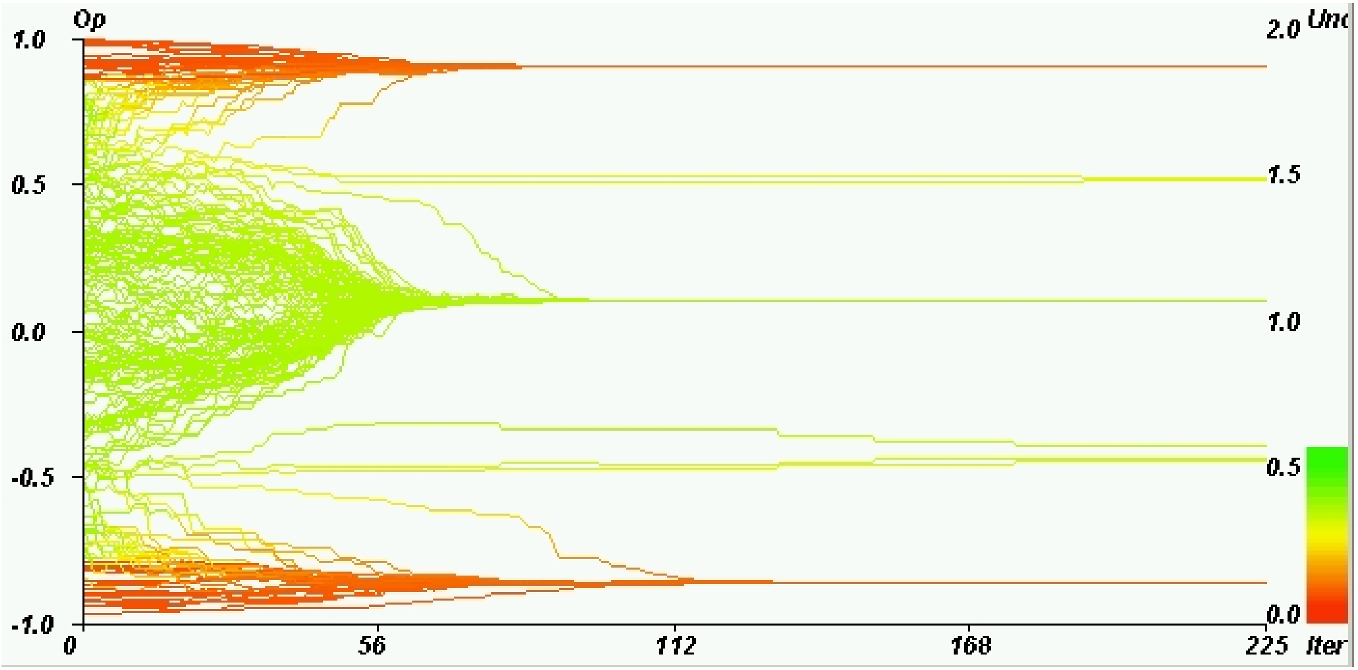
$$\tau \in [0,2]$$
$$\mu \in \left[0, \frac{1}{2}\right]$$



# Case Study 5: Opinion Dynamics

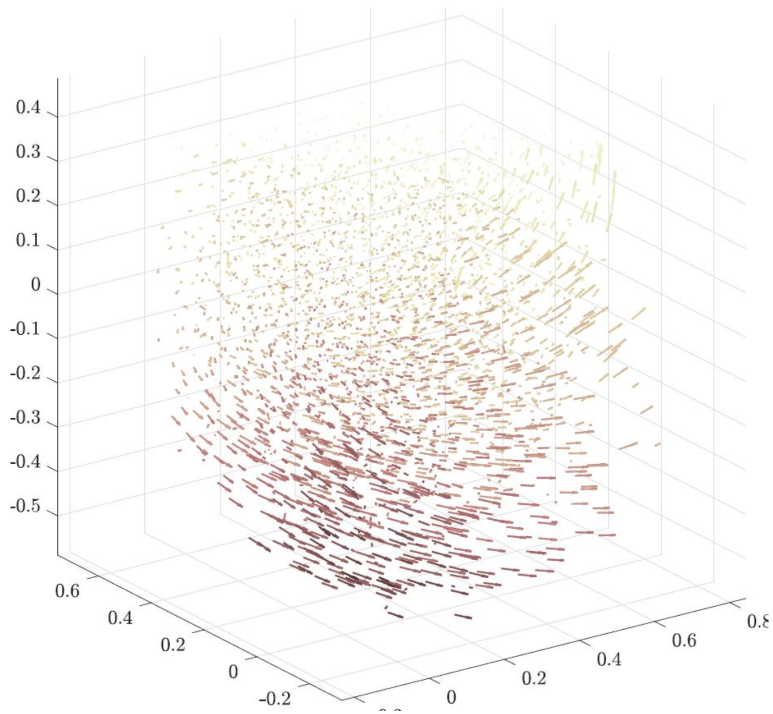


- Picturesque model to show, how communities with different opinions develop (e.g. Political parties,...)

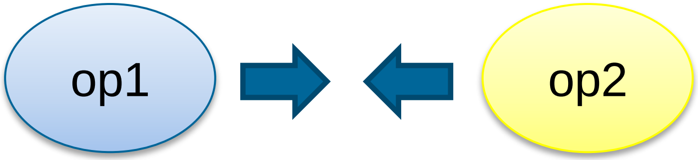


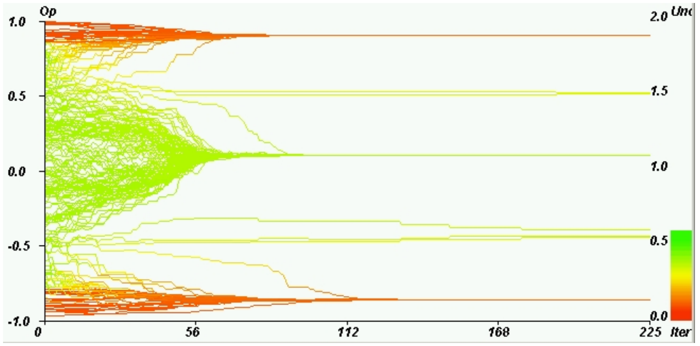
# Case Study 5: Opinion Dynamics

any new differences?



vs.



# Classification of ABMs (5)

TU WIEN Mathematical Modelling and Simulation

| Classification 5 | | |
|---|---|---|
| **with respect to randomness (stochasticity)** | | |
| **stochastic** | | **deterministic** |
| **initial-value stochastic** | **update stochastic** | |
| • Initial setting (of agents) is determined using random numbers | • Update rules user random numbers | • The outcome of the model is uniquely defined by its initial condition |

# Summary

When developing an agent-based model particularly care for ...

- ... order of actions / simultaneous events
- ... correct movement of agents if using a gridded environment
- ... correct result interpretation (randomness , quantitative/qualitative, ...)
- ... code performance
- ... reproducible documentation