Simulations in Statistical Physics

Course for MSc physics students

Janos Török

Department of Theoretical Physics

April 30, 2020

Algorithmically defined models

- Self-Organized Criticality
  - Bak-Tang-Wiesenfeld model
  - Forest fire model
  - Bak-Sneppen model of evolution
- Traffic models
- 1d driven systems

Agent based models

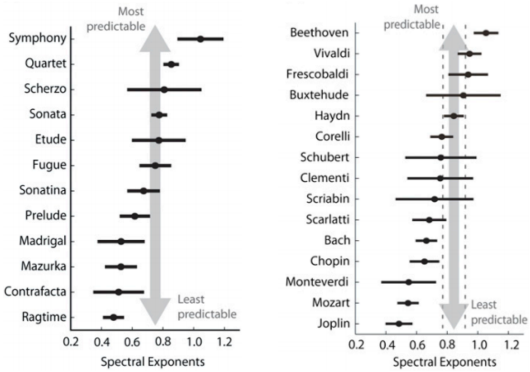
- Agents with well defined rules
- Interactions
  - Among all agents
  - Between neighbors (lattice or network)
  - Within a distance
- Complex emergent phenomena

Self-Organized Criticality

- Critical state: inflection point in the critical isotherm
- Power law functions of correlation length, relaxation time
- Control parameter, generally temperature
- Critical point as an attractor?
- Why? Power law: We see many cases
  - $1/f$ noise (music, ocean, earthquakes, flames)
  - Lack of scales (market, earthquakes)
- Underlying mechanism?

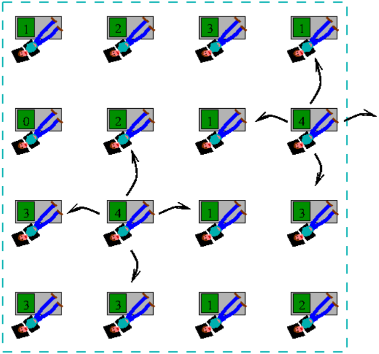
$1/f$ spectrum

- Music, paintings, fire
- We seem to like it



Bak-Tang-Wiesenfeld model

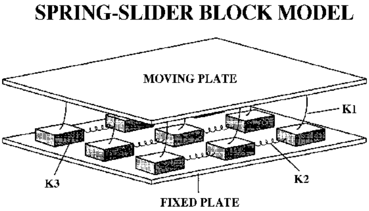
- Originally a sandpile model
- Better explained as a *Lazy Bureaucrat model*:
  - Bureaucrats are sitting in a large office in a square lattice arrangement
  - Occasionally the boss comes with a dossier and places it on a random table
  - The bureaucrats do *nothing* until they have less than 4 dossiers on their table



- Once a bureaucrat has 4 or more dossiers on its table starts to panic and distributes its dossiers to its 4 neighbors
- The ones sitting at the windows give also 1 dossier to its neighbors and throw the rest out of the window.

Bak-Tang-Wiesenfeld model

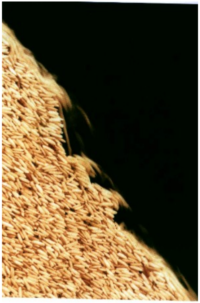
- Originally a sandpile model
- Better explained as a *Lazy Bureaucrat model*:
- Best application: Spring block model of earthquakes:
  - Masses sitting on a frictional plane in a grid are connected with springs to eachother and to the top plate



- Top plate moves slowly, increasing the stress on the top springs slowly and randomly
- If force is large enough masses move which increases the stress on the neighboring masses

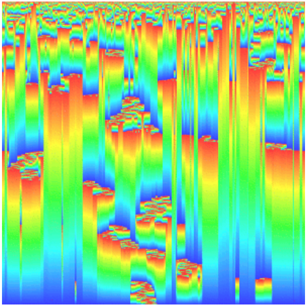
Bak-Tang-Wiesenfeld model: Results

- Avalanche size distribution is a power law $\rightarrow$ no scale
- No external parameter tuning for scale free state
- Hence the name: Self-organized criticality
- Lifetime distribution of avalanches: $1/f$ (unfortunately for sand it is $1/f^2$)



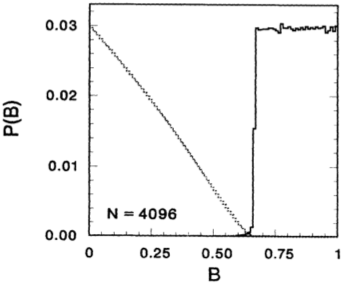
Bak-Sneppen model of evolution

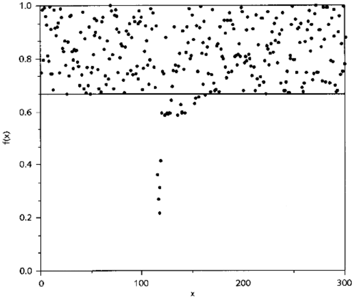
- N species all depends on two other (ring geometry)
- Each species are characterized by a single *fitness*
- In each turn the species with the lowest fitness dies out and with it its two neighbors irrespective of their fitness
- These 3 species are replaced by new ones with random fitness
- Inital and update fitness is uniform between $[0, 1]$



Bak-Sneppen model of evolution: Results

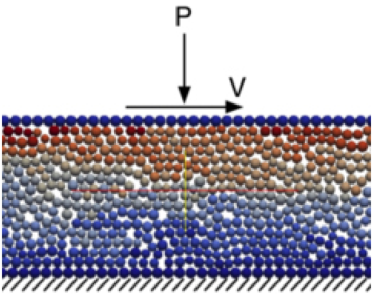
- Steady state with avalanches
- Avalanches start with a fitness $f > f_c \simeq 0.66$
- Avalanche size distribution power law
- Distance correlation power law

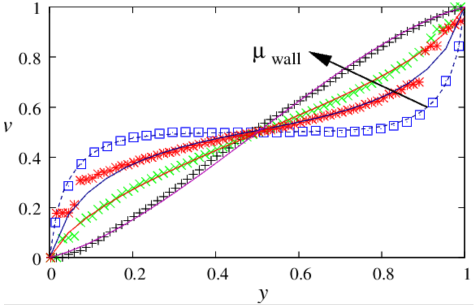




Bak-Sneppen model of evolution an application: Granular shear

- Fitness $\rightarrow$ Effective friction coefficient
- Speciment with lowest fitness dies out $\rightarrow$ block is sheared at weakest position (shear band)
- Neighbors, related species die out and replaced by new species $\rightarrow$ structure gets random aroung the shear band.

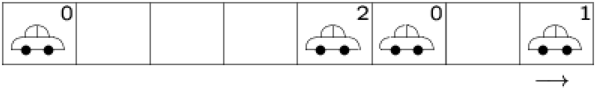




Traffic models

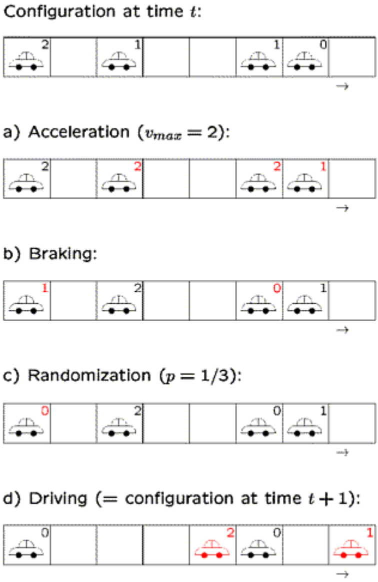
Nagel–Schreckenberg model

- Periodic 1d lattice (ring) Autobahn
- discretized in space and time
- Cars occupying a lattice moving with velocities $v = 0, 1, 2, 3, 4, 5$
- Remark, if max speed is 126 km/h, then lattice length is 7 m, a very good guess for a car in a traffic jam
- It uses parallel update: at each timestep all cars move v sites forward

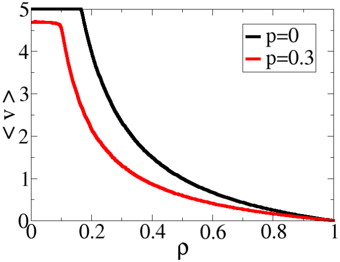


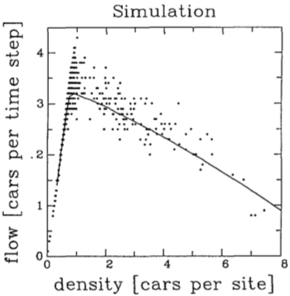
Nagel–Schreckenberg model

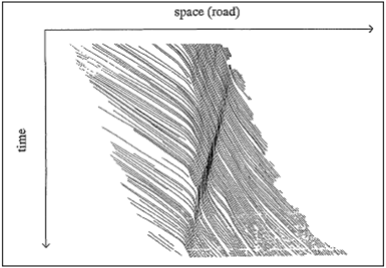
- Algorithm:
  1. **Acceleration**: All cars not at the maximum velocity increase their velocity by 1
  2. **Slowing down**: Speed is reduced to distance ahead (1 sec rule)
  3. **Randomization**: With probability p speed is reduced by 1
  4. **Car motion**: Each car moves forward the number of cells equal to their velocity.

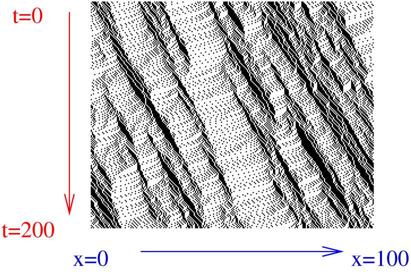


Emergence of traffic jams







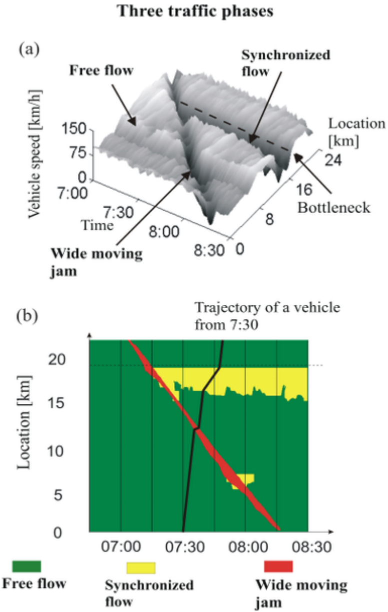


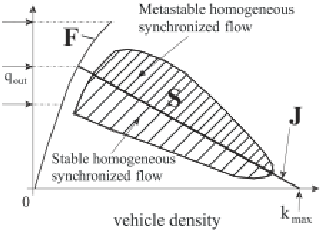
Nagel–Schreckenberg model

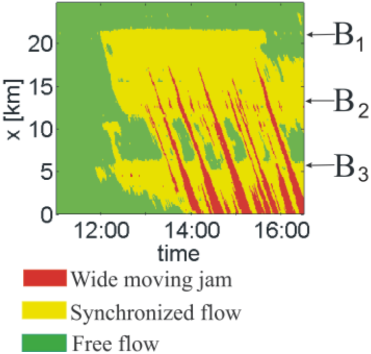
- Transition from free-flow to jammed state
- Jammed state is a phase-separated phase
- Without randomization a sharp transition
- Had been used in NRW to predict traffic jams


autobahn.NRW
Hochrechnung für Bundesautobahnen
Montag, 01.07.2013, 08:05 h
aktuelle Verkehrslage
Prognose 30 min
Prognose 60 min
Langzeit Prognose
+7
Fahrzeitprognose
Farbmodus: Standard
Baustelleninfo
Verkehrsdichten
frei
zähfließend
dicht
gestaut
Baustelle mit geringer Staugefahr
Baustelle mit hoher Staugefahr
Anschlussstelle
Sperrung der Autobahn
Vollsperrung der Anschlussstelle
Teilsperrung der Anschlussstelle
Informationen
Impressum
Version 3.6.1
0 km
15 km
Wesel
Recklinghausen
Lünen
Geldern
Kamp-Lintfort
Bottrop
Gelsenk.
Herne
Dortmund
Oberhsn.
Essen
Moers
Bochum
Mülheim
Duisburg
Witten
Kempen
Krefeld
Hagen
Velbert
Gevelsberg
Viersen
Düsseldorf
Mönchengladbach
Neuss
Wuppertal
Lüdenscheid

Three-phase traffic theory





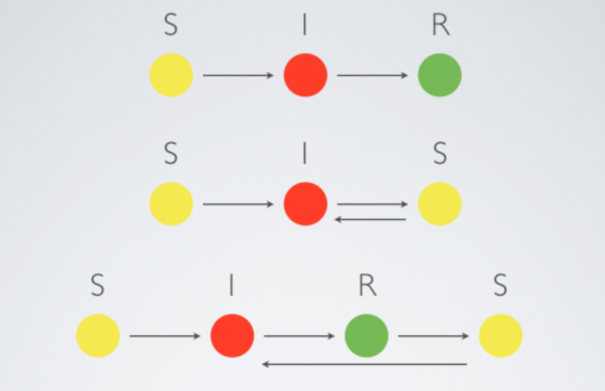


Disease spreding, SIR model

- S: susceptible (can be infected with prob. β if meets an infected)
- I: Infected (may infect susceptible, but may recover with prob. ν).
- R: Recovered (Immune to the disease)
- Other versions:
  - SI: agents do not recover (e.g. information spreading)
  - SIS: recovered people can get disease again
  - SIRS: recovered agents may become susceptible (e.g. influenza)

Disease spreding, SIR model

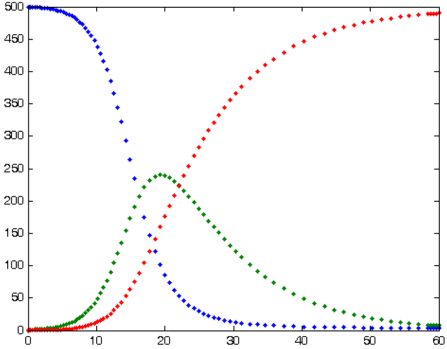
- S: susceptible
- I: Infected
- R: Recovered



SIR model, connected graph

Governing equations:

$$\dot{S} = -\beta I S$$
$$\dot{I} = \beta I S - \nu I$$
$$\dot{R} = \nu I$$



SIR model, connected graph

Governing equations:

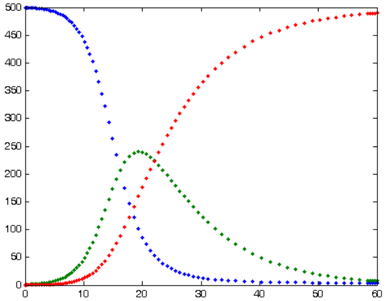
$$\dot{S} = -\beta I S$$
$$\dot{I} = \beta I S - \nu I$$
$$\dot{R} = \nu I$$

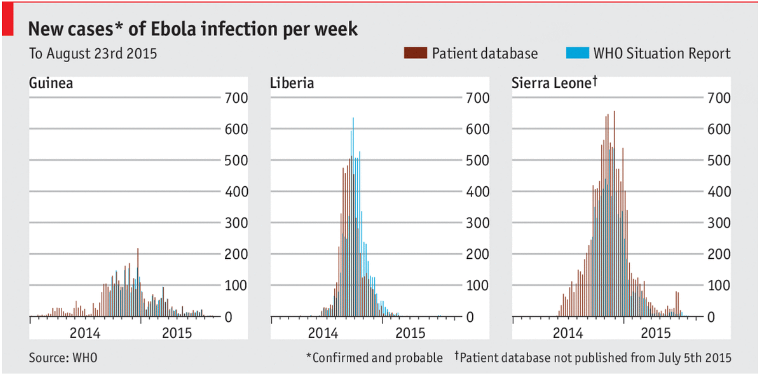
- Early stage $S \simeq 1$

$$I \simeq I_0 \exp[(\beta - \nu)t]$$

- $R = \beta/\nu$ epidemic threshold
  - $R > 1$ outbreak
  - $R < 1$ localized

SIR model vs. reality

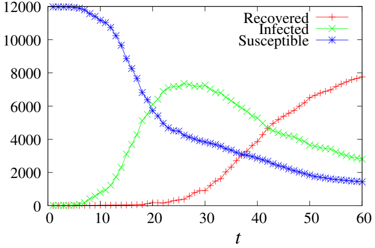


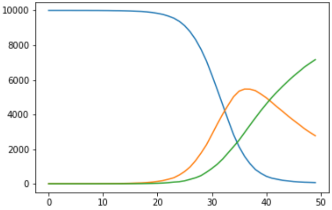


Economist.com

SIR reality vs. model

- Perfect mixing
- Everybody can meet everybody
- Covid-19, South Korea
- Susceptible approximated

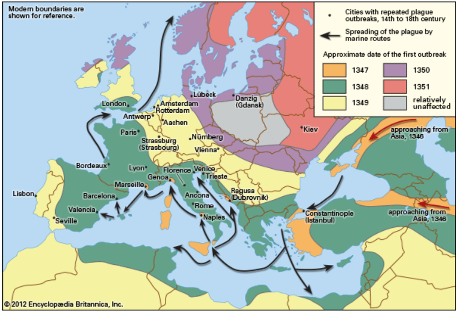


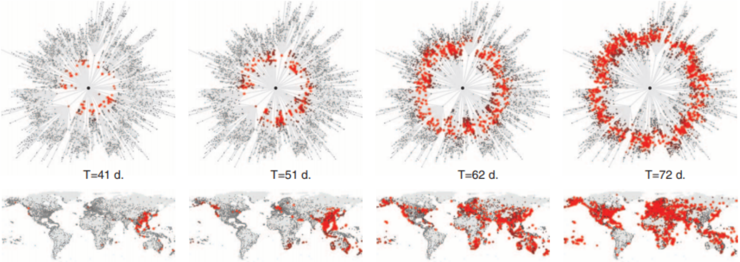


Algorithm for the SIR model

1. List of initially infected nodes is I
2. Get a random (infected) node u from the list I
3. For all neighbors w of u do 4.
4. If w is susceptible change it to infected with probability β, and enqueue it into list I
5. With probability ν change state of u to recovered otherwise put it back to I
6. If I is not empty go back to 2.

SIR on space and network





Brockmann-Helbing

SIR and vaccination

- Epidemic threshold depends on the network:

$$\lambda_c = \beta/\nu \simeq \frac{\langle k \rangle}{\langle k^2 \rangle}$$

- where k is the degree of a node
- For lattices and random networks is is finite
- For scale free networks $P(k) \sim k^{-\gamma}$, with $2 < \gamma < 3$ this is infinite. Unfortunately many networks fall in this regime, e.g. transportation networks (especially flight), inside a country $\gamma > 3$
- Vaccination: bring down λ_c to 1
  - Vaccinate hubs (doctors, shopkeepers, etc.)
  - Vaccinate large part of the society $> 50\%$

Predator prey model

- $N(t)$ number of predators
- $E(t)$ number of prey
- Model (Lotka 1925, Volterra 1926):

$$\begin{aligned} \dot{E}(t) &= \beta_E E(t) - [\mu_E N(t)]E(t) \\ \dot{E}(t) &= [\beta_N E(t)]N(t) - \mu_N N(t) \end{aligned} \tag{1}$$

- Solution $\dot{E} = \dot{N} = 0$:

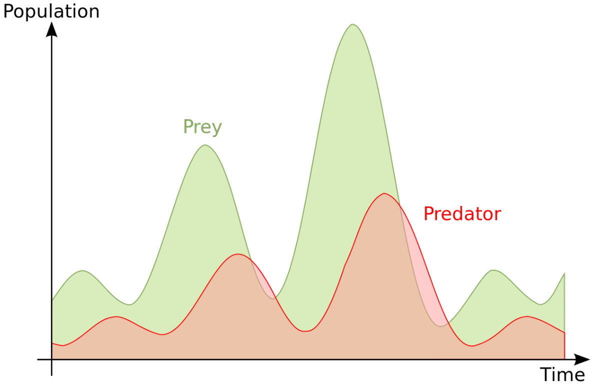
$$\begin{aligned} N = E &= 0 \\ N = \beta_E/\mu_E, &\quad E = \nu_N/\beta_N \end{aligned} \tag{2}$$

Predator prey model

- Solution $\dot{E} = \dot{N} = 0$:

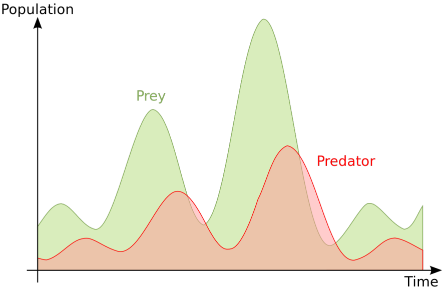
$$\begin{aligned} N = E = 0 \\ N = \beta_E/\mu_E, \quad E = \nu_N/\beta_N \end{aligned} \tag{3}$$

- Numerical solution:

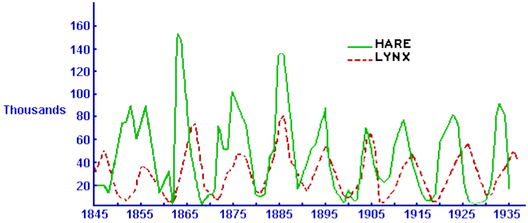


Predator prey model

- Numerical solution:



- Reality:



Other agent based models

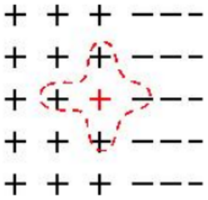
- Agents are nodes
- Interactions through links
- Any network:
  - Lattices
  - Random networks
  - Scale-free
  - Fully connected graphs
- Examples:
  - Opinion models
  - Game models

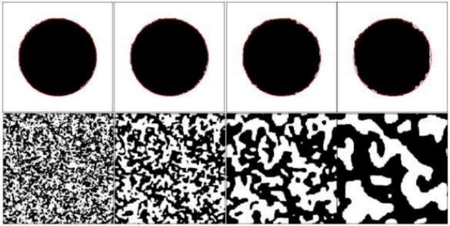
Opinion models

- Agents have opinion x_i
  - binary ± 1 (yes/no)
  - discrete (parties)
  - continuous (views)
  - vector (different aspects)
- Interaction with other agents
  - pairwise
  - global (with mean)

Ising-model at $T = 0$

- Result depends on the lattice type (surface tension)
- Phase transition
- For larger systems probability to reach order goes to zero in $d > 2$ (surface gets more important)
- Fully connected goes to order (no surface)



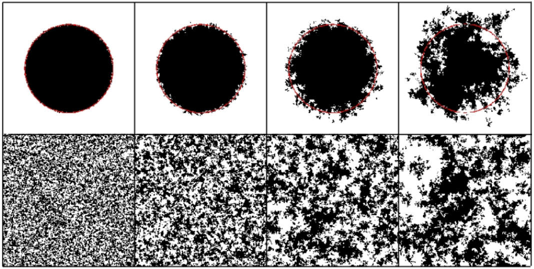


Voter model

- Agents take opinion of random neighbor

$$\begin{array}{ccccccccccc} & & 1 & & & & & & 1 & & \\ & & | & & & & & & | & & \\ 1 & - & 1 & - & 0 & \to & 1 & - & 0 & - & 0 \\ & & | & & & & & & | & & \\ & & 1 & & & & & & 1 & & \end{array}$$

- $d = 1, 2$ final state is consensus
- $d > 2$ final state is not consensus, but a finite system reaches consensus after a time $\tau(N) \sim N$



Variants

- Majority rule (with two neighbors (3 nodes) towards majority)
- Presence of zealots, i. e. agents that do not change their opinion
- Presence of contrarians
- Three opinion states with interactions only between neighboring states
- Noise (with some probability p agents change their state)
- Biased opinion in case of a tie

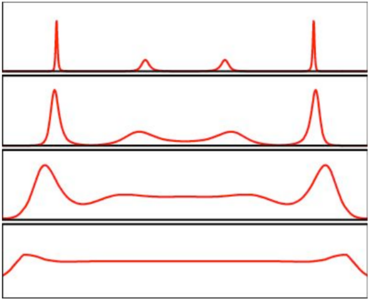
Bounded confidence model: Deffuant model

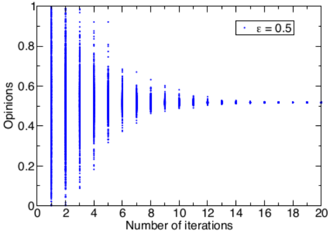
- Agents have opinion x_i
- if $|x_i(t) - x_j(t)| < \varepsilon$ then
  - $x_i(t+1) = x_i(t) - \mu[x_i(t) - x_j(t)]$
  - $x_j(t+1) = x_j(t) + \mu[x_i(t) - x_j(t)]$
- μ compromise parameter $\mu = 1/2$ complete compromise
- ε tolerance parameter
- Methods:
  - Monte-Carlo simulation
  - Master equation:

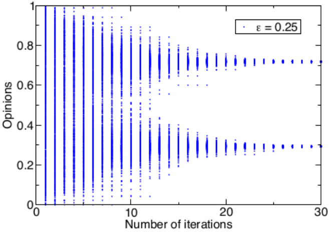
$$\frac{\partial P(x,t)}{\partial t} = \int_{|x_1 - x_2| < \varepsilon} dx_1 dx_2 P(x_1, t) P(x_2, t) \times$$

$$\times \left[\delta \left(x - \frac{x_1 + x_2}{2} \right) - \delta(x - x_1) \right]$$

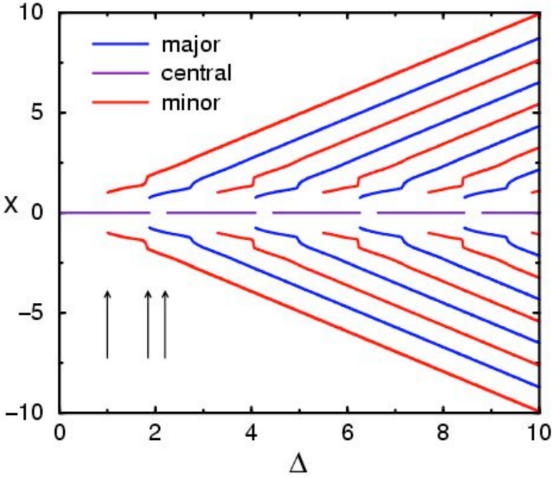
Deffuant model: Opinion groups (fully connected graph)







Deffuant model: Bifurcation diagram



$\Delta = 2/\varepsilon$, $\mu = 1/2$

Flocking Model

Flocking model

- Birds move with constant velocity (v_0)
- Align themselfs to neighbors
- Some noise due to inaccurate averaging
- Differential equation

$$\theta_i(t + \Delta t) = \langle \theta(t) \rangle_{|\mathbf{r}_i - \mathbf{r}_j| < R} + \xi$$

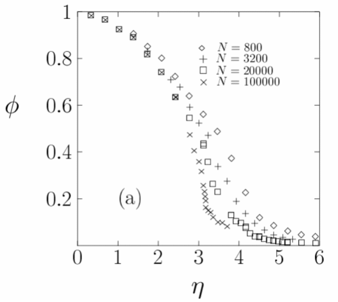
- Upgrade position:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + v_0 \mathbf{e}(\theta_i(t)) \Delta t$$

where $\mathbf{e}(\theta)$ is a unit vector in the direction of θ.

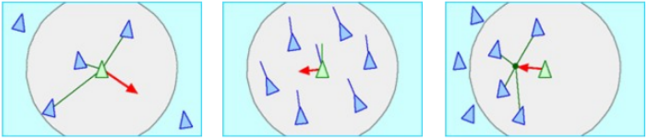
Flocking model

- Birds move with constant velocity (v_0)
- Align themselfs to neighbors
- Some noise due to inaccurate averaging
- Phase diagram 1d:



Flocking model

- Birds move with constant velocity (v_0)
- Align themselfs to neighbors
- Some noise due to inaccurate averaging
- Non-physicist model:



Separation: steer to avoid crowding local flockmates

Alignment: steer towards the average heading of local flockmates

Cohesion: steer to move toward the average position of local flockmates

http://www.red3d.com/cwr/boids/

Practice: SIR model

1. Create a network (random or lattice)
2. List of initially infected nodes is I
3. Get a random (infected) node u from the list I
4. For all neighbors w of u do 5.
5. If w is susceptible change it to infected with probability β, and enqueue it into list I
6. With probability ν change state of u to recovered otherwise put it back to I
7. If I is not empty go back to 3.
8. Check the epidemic threshold
9. Check the effect of testing: if someone is infected with probability μ it is discovered and all connected nodes go to quarantine for $1/\mu$ timesteps under which they cannot be infected nor may infect