Simulations in Statistical Physics

Course for MSc physics students

Janos Török

Department of Theoretical Physics

December 9, 2014

Exams

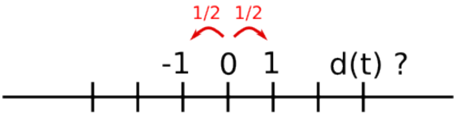
All exam starts at 9 am.

- 2014.12.18.
- 2015.01.05.
- 2015.01.07.
- 2015.01.12.
- 2015.01.15.
- 2015.01.19.
- 2015.01.21.
- 2015.01.26.

Spreading on networks

- Diffusion
- Random walk
- Disease USA UK

Random Walk on Random Networks



- Master egyenlet:

$$\frac{\partial n(i)}{\partial t} = \frac{1}{2}[n(i-1) - 2n(i) + n(i+1)]$$

$$\frac{\partial n(x)}{\partial t} = D\Delta n(x)$$

- Dicrete:

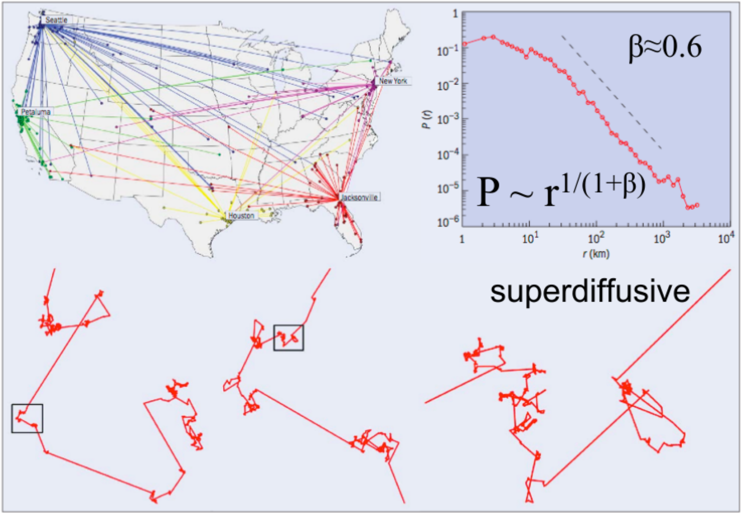
$$\frac{\partial n_i}{\partial t} = \sum_j D_{ij} n_j$$

- What is D_{ij}?

Random Walk on Random Networks

- Discrete Laplace operator D_{ij}
- 1d: $$\begin{pmatrix} -2 & 1 & & & & 0 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & \ddots & & \\ & & \ddots & \ddots & 1 & \\ & & & 1 & -2 & 1 \\ 0 & & & & 1 & -2 \end{pmatrix}$$
- 2d: $$\begin{pmatrix} -4 & 1 & 0\cdots & 1 & & 0 \\ 1 & -4 & 1 & & \ddots & \\ 0 & 1 & -4 & \ddots & & 1 \\ 1 & & \ddots & \ddots & 1 & \\ & \ddots & & 1 & -4 & 1 \\ 0 & & 1 & & 1 & -4 \end{pmatrix}$$
- General: adjacency matrix: $D_{ij} = A_{ij} - k_j \delta_{ij}$

Random Walk on Random Networks



Random Walk on Random Networks

- Rate equation n_k probability of finding the walker an a site with k edges:

$$\frac{\partial n_k}{\partial t} = -rn_k + k\sum_{k'} P(k'|k)\frac{r}{k'}n_{k'}$$

- Uncorrelated random network:

$$P(k'|k) = \frac{k'}{\langle k\rangle}P_{k'}$$

- New equation:

$$\frac{\partial n_k}{\partial t} = -rn_k + r\frac{k}{\langle k\rangle}\sum_{k'} P(k')n_{k'}$$

- Solution:

$$n_k = \frac{k}{\langle k\rangle N}$$

- Random walkers gather on high connectivity nodes

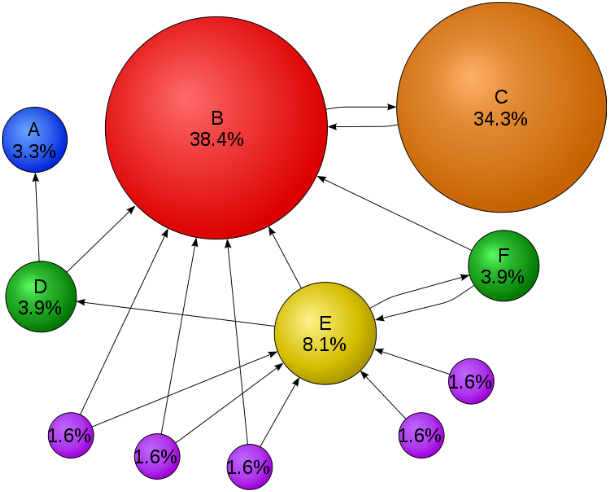
Page rank

- Do what surfers do
- Random walk on pages, but sometimes (probability q) a new (random) restart
- Dumping factor $d = 1 - q$ (general choice $d = 0.85$).
- Self-consistent, equation:

$$P_R(i) = \frac{q}{N} - (1 - q) \sum_j A_{ij} \frac{P_R(j)}{k_{\text{out},j}}$$

- Solution: iteration
- Result: Favours sites which are linked by many (reliable sources)

Page rank example

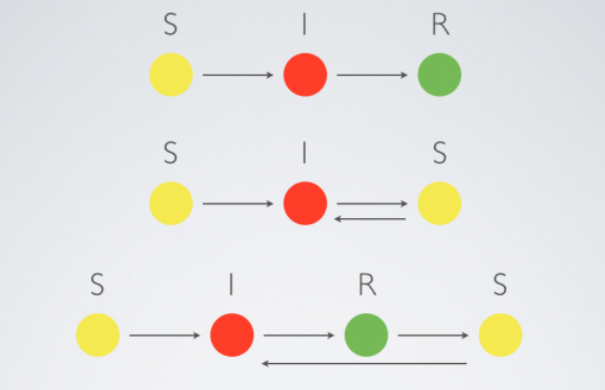


Disease spreding, SIR model

- 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

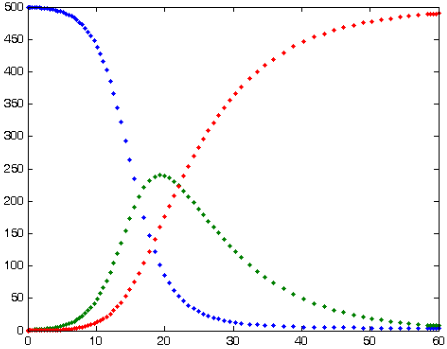
- S: susceptible
- I: Infected
- R: Recovered



SIR model, connected graph

Governing equations:

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



SIR model, connected graph

Governing equations:

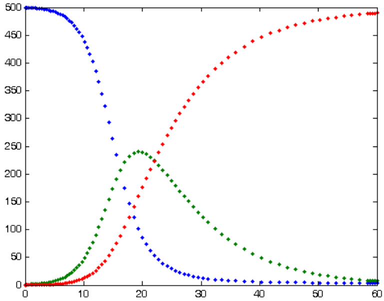
$$\dot{S} = -\beta I S$$
$$\dot{I} = \beta I S - \nu I$$
$$\dot{R} = \nu I$$

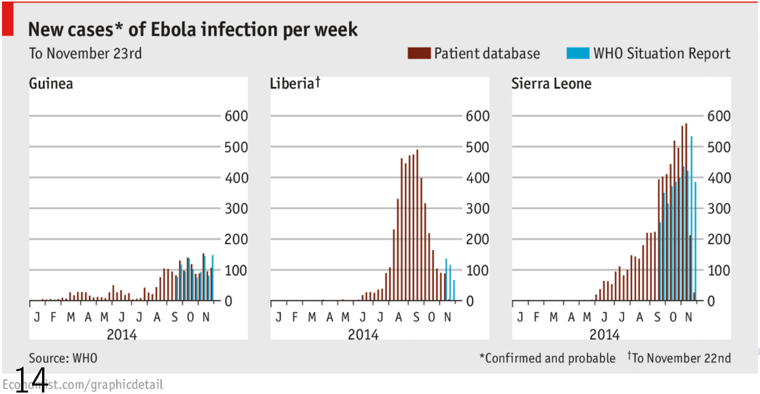
- Early stage $S \simeq 1$

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

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

SIR model vs. reality





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.

Bit coding algorithm for the SIR model

- Ensemble average: each bit is a different instance
- Choose a link l which is between nodes n_i and n_j
- r is a random number with bits 1 of probability β (choose $\beta = 2^{-n}$ or similar)
- Passing disease: $p = [s(n_i)|s(n_j)]\&r$
- Change states: $s(n_i)| = p$ and $s(n_j)| = p$
- A slightly different implementation than previous

Other agent based models

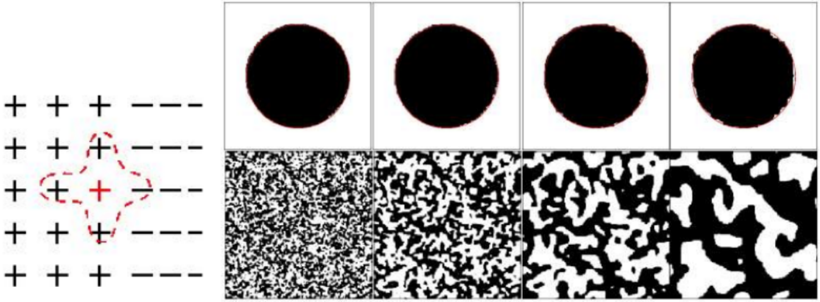
- Agents are nodes
- Interactions through links
- Any network:
  - Lattices
  - Random networks
  - Scale-free
  - Fully connected graphs
- Examples:
  - Opinion models
  - Game models

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)

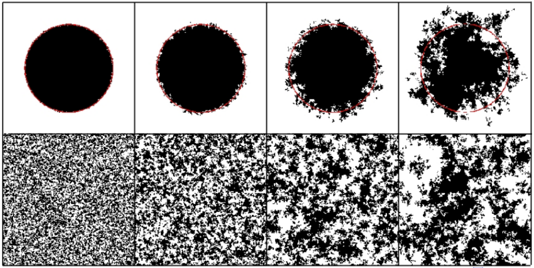


Voter model

- Agents take opinion of random neighbor

$$
\begin{array}{ccccccccccc}
 & & 1 & & & & & & 1 & & \\
 & & | & & & & & & | & & \\
1 & - & 1 & \color{red}{-} & \color{red}{0} & \rightarrow & 1 & - & \color{red}{0} & - & 0 \\
 & & | & & & & & & | & & \\
 & & 1 & & & & & & 1 & &
\end{array}
$$

- $d = 1, 2$ 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

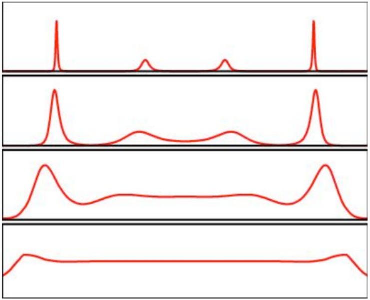
Bounded confidence model: Deffuant model

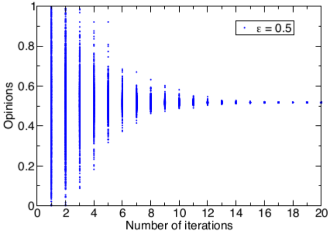
- Agents have opinion x_i
- if $|x_i(t) - x_j(t)| < \varepsilon$ then
  - $x_i(t+1) = x_i(t) - \mu[x_i(t) - x_j(t)]$
  - $x_j(t+1) = x_j(t) + \mu[x_i(t) - x_j(t)]$
- μ compromise parameter $\mu = 1/2$ complete compromise
- ε tolerance parameter
- Methods:
  - Monte-Carlo simulation
  - Master equation:

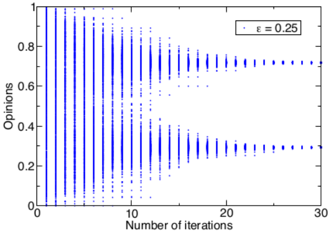
$$\frac{\partial P(x,t)}{\partial t} = \int_{|x_1 - x_2| < \varepsilon} dx_1 dx_2 P(x_1,t) P(x_2,t) \times$$

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

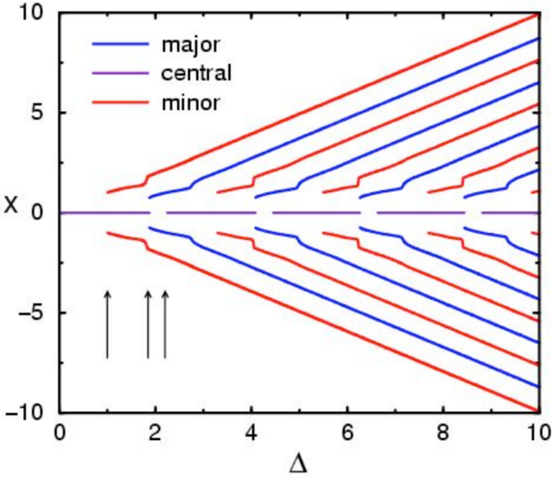
Deffuant model: Opinion groups (fully connected graph)







Deffuant model: Bifurcation diagram

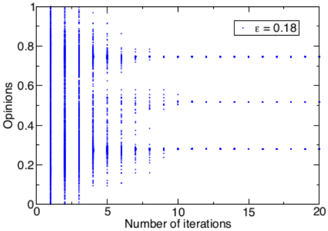


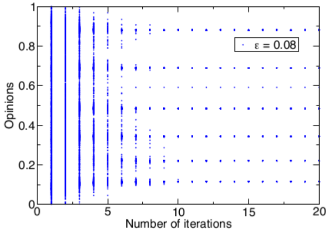
$\Delta = 2/\varepsilon$, $\mu = 1/2$

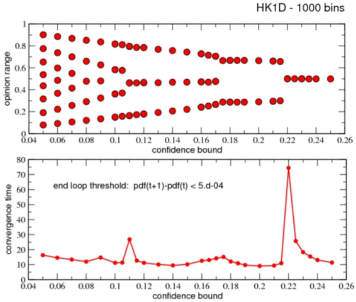
Global: Hegselmann-Krause model

- Choose node i
- Test for all neighbors, which have opinion within the tolerance level
- Average their opinion
- Adapt to it
- Similar behavior

Hegselmann-Krause model





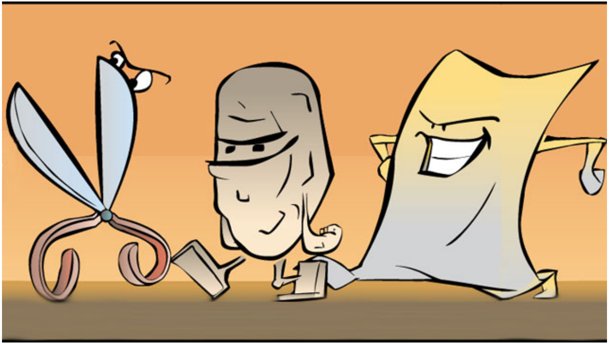


Game models:

- Rock-paper-scissors
- Prisoner's dilemma
- Chicken, hawk-dove game

Rock-paper-scissors

- No winning strategy on (truly) random opponent
- E.g bacterian and antibiotics in mice
- Grass-rabbit-fox
- Popular in games



Prisoner's Dilemma

- Each player with a preferred strategy that collectively results in an inferior outcome
- Dominating strategy regardless of the opponent's action
- Nash equilibrium, from which no individual player benefits from deviating

	Cooperate	Defect
Cooperate	4, 4	1, 5
Defect	5, 1	2, 2

Prisoner's Dilemma

- One game $\rightarrow$ defect
- Fixed number of games $\rightarrow$ defect
- Large pool of players (movie):
  - If other codes are known, it can be derived
  - If pool is diverse the best strategy is tit for tat (start with cooperation)
  - In general:
    - Nice (do not defect before opponent does)
    - Retaliating (punish!)
    - Forgiving (Yes!)
    - Non-envious (do not want to gain more than your neighbor)

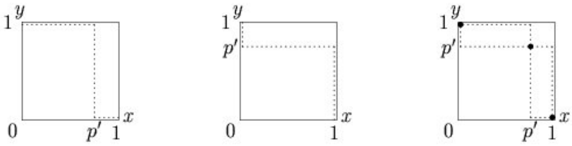
Chicken game, Hawk-Dove game

Chicken game, Hawk-Dove game

- No preferred strategy
- The best strategy is to anti-coordinate with your opponent

	Cooperate	Defect
Cooperate	0, 0	-1, 2
Defect	2, -1	-5, -5

- Example: Cold war
- Solution: anti-correlated pure strategy
- Probabilistic (play Hawk with p')



Chicken game, Hawk-Dove game difference to Prisoner's dilemma

	Cooperate	Defect
Cooperate	Reward	S, T
Defect	T, S	Punish

Hawk-Dove

	C	D
C	0, 0	-1, +1
D	+1, -1	-10,-10

Prisoner's dilemma

	C	D
C	3, 3	0, 5
D	5, 0	1, 1

- Prisoner's dilemma:
  Temptation(T)>Reward(R)>Punish(P)>Sucker(S)
- Chicken game:
  Temptation(T)>Reward(R)>Sucker(S)>Punish(P)

Prisoner's dilemma