Agent Based Models

Sarah Olson

Department of Mathematical Sciences

Worcester Polytechnic Institute

WPI

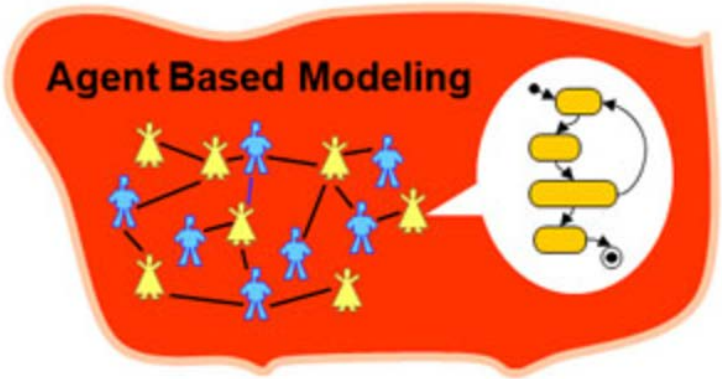
2018 Mathematics Institute for Secondary Teaching

Math Modeling

- Why Model??
  - To understand, To predict, To control
- Models provide a conceptual framework
- Synthesize/summarize large quantities of data
- Provide insight and test hypotheses
- Agent Based Models: easy to understand implement
- Mathematics – provides a precise language with well defined rules (variables, parameters,…)

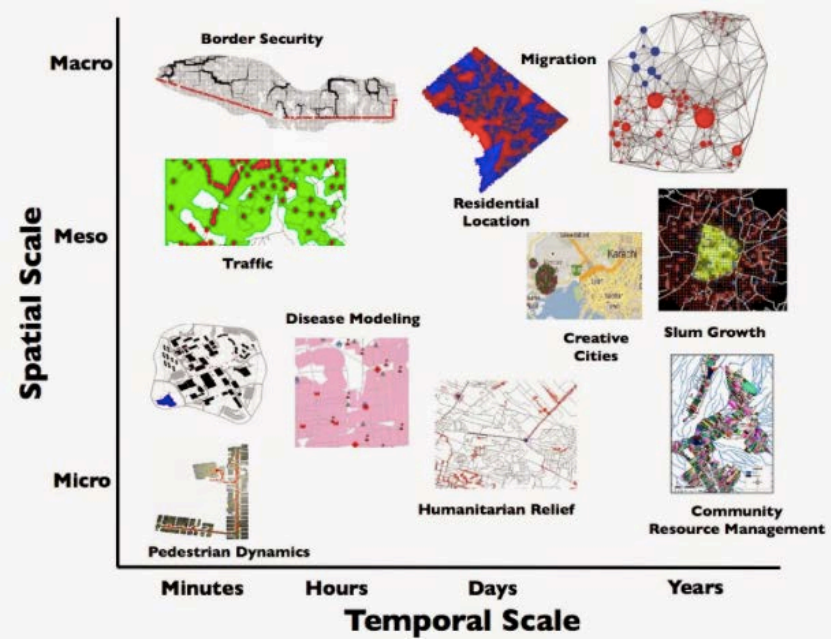
Agent Based Models

- Also known as:
  - Cellular Automaton
  - (Biased) Random Walks
- Basic idea: Rule based
- "Agents" have different states and rules govern whether they change states and/or move, rules for interactions of agents.
- Interesting emergent phenomena and interactions can occur!

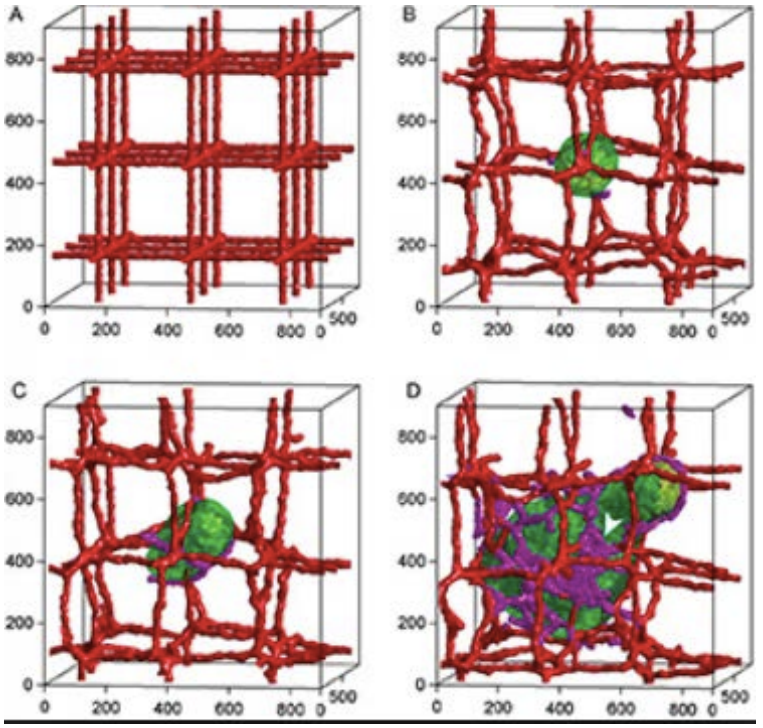


http://introtopolicyinformatics.wikispaces.asu.edu/Agent+Based+Modeling

Agent Based Models - Applications



https://www.gisagents.org/2015/01/



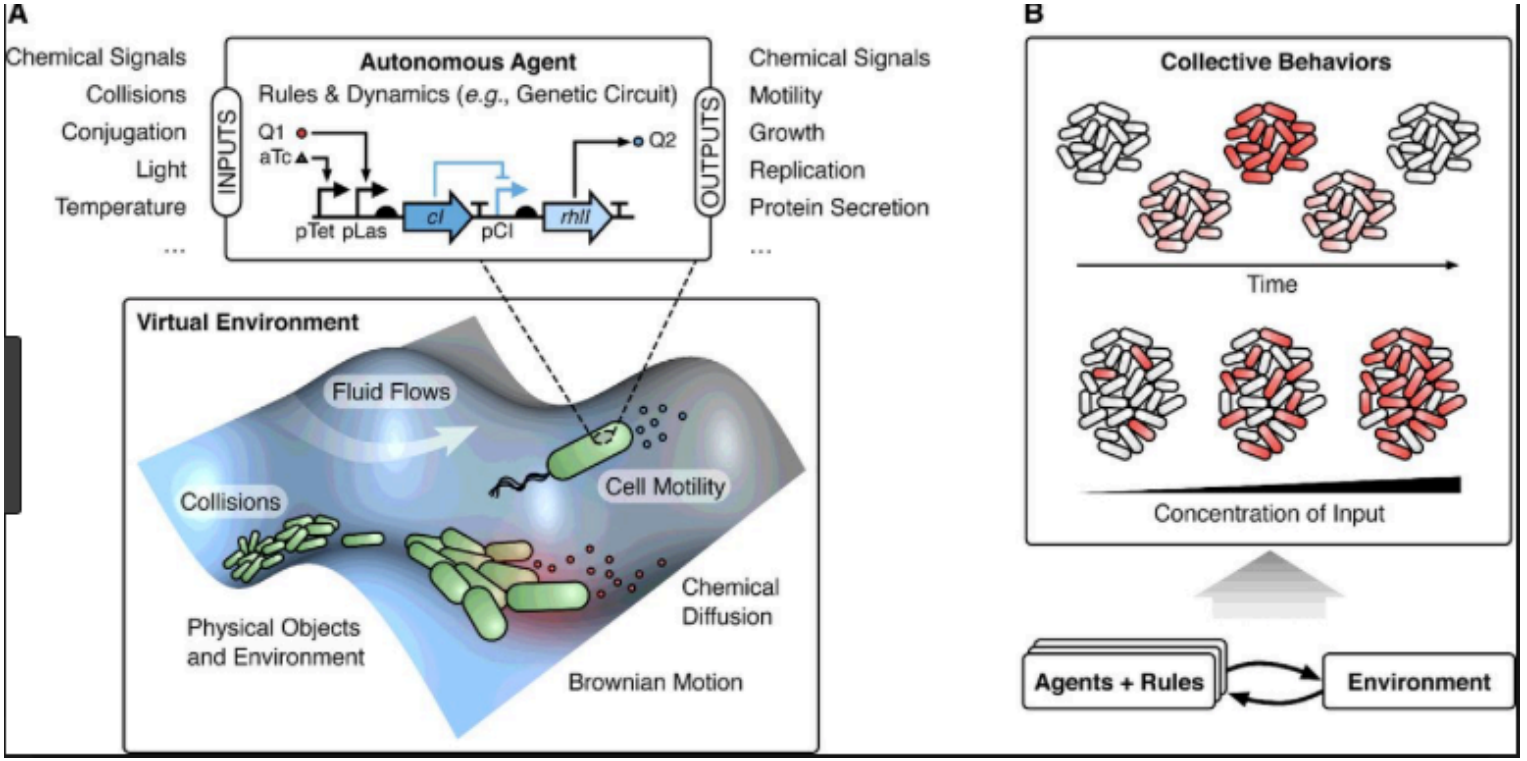
The Biology of Interacting Things: The Intuitive Power of Agent-Based Models

Biomedical applications of ABMs are taking off.

By Alexander Gelfand

pdf *print* *leave a comment*

Agent Based Models - Applications

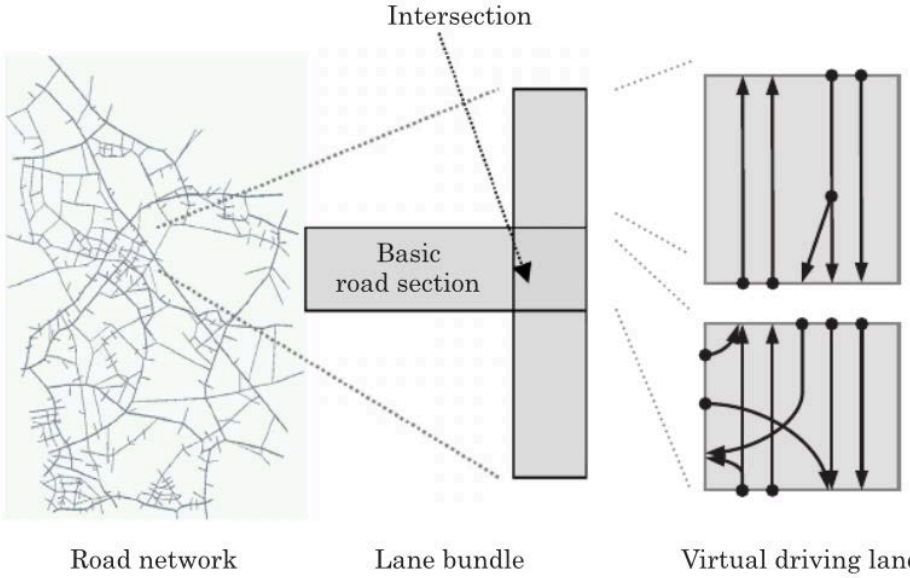


Agent-based modelling in synthetic biology

Thomas E. Gorochowski

Essays In Biochemistry
Nov 30, 2016,
60(4)
325-336;

Applications



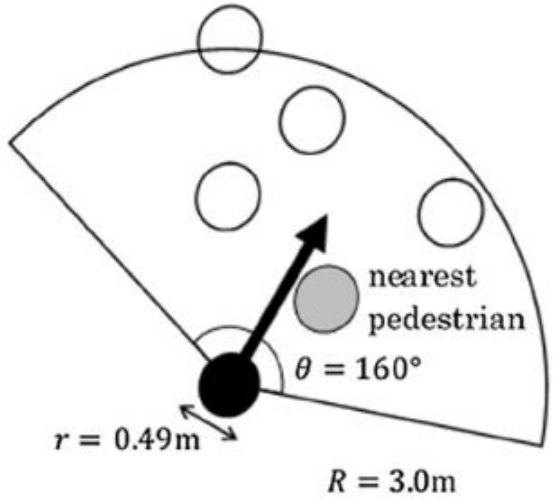
Contents lists available at ScienceDirect

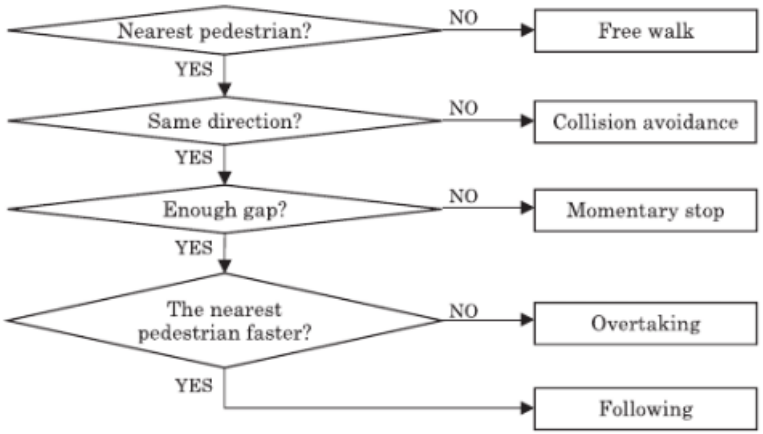
Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Agent-based simulation framework for mixed traffic of cars, pedestrians and trams☆

Hideki Fujii*, Hideaki Uchida, Shinobu Yoshimura





Agent Based Models

- Why are they so popular?
  - Complex systems are handled easily
  - Autonomous Agents (not centrally governed)
  - Diverse and Heterogeneous Agents
  - Agents can adapt
- Complex social processes and a system can be built from the bottom up
- Can be analyzed mathematically

Agent Based Models

- Computational Power
- CPU: just a few cores with huge cache memory that can handle a few threads (or processes) at a time
- GPU: hundred of cores that can handle thousands of threads (or processes) simultaneously

CPU versus GPU

Graphical processing unit –> the brains

GPU

Central processing unit –> the brains

CPU

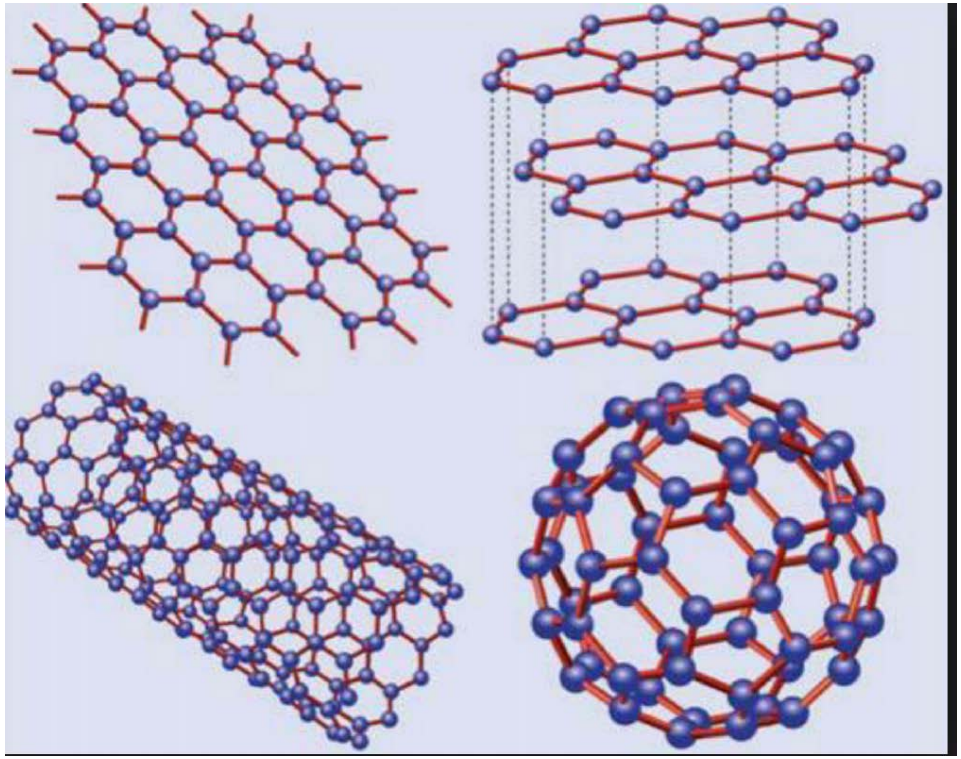
Table 2: Time to Generate Approximations (Without GPU, With GPU) sec

	1e6	1e7
n=5	(0.187122, 0.010357)	(1.722311, 0.048325)
n=10	(0.316912, 0.013135)	(3.170940, 0.082713)
n=15	(0.472206, 0.016734)	(4.725572, 0.117412)
n=20	(0.644042, 0.020177)	(6.274727, 0.151933)

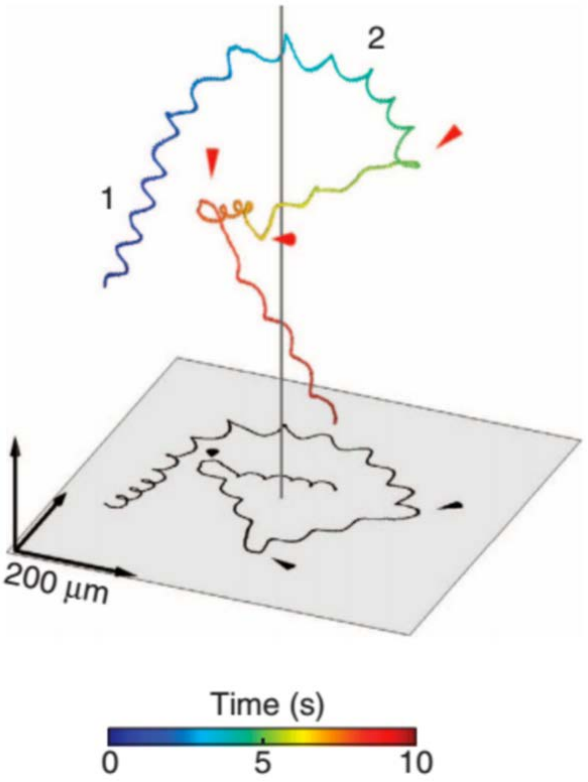
https://www.overclock3d.net/reviews/gpu_displays/gainward_gtx_titan/2

Models – movement

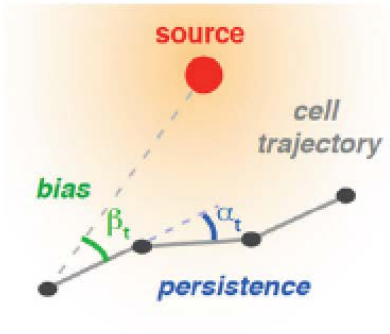
- Lattice

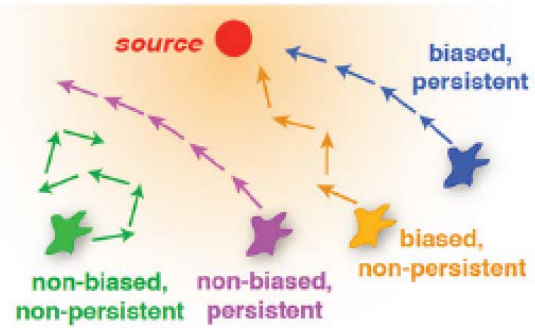


- Off-Lattice



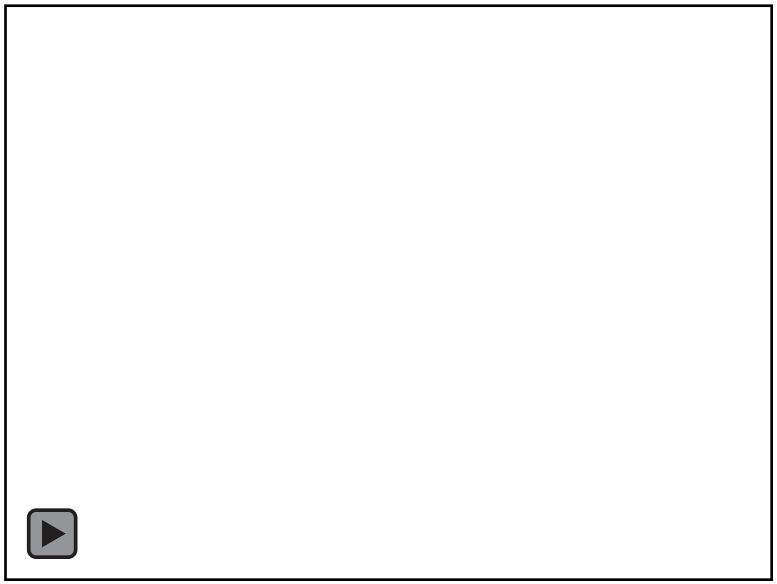
Movement





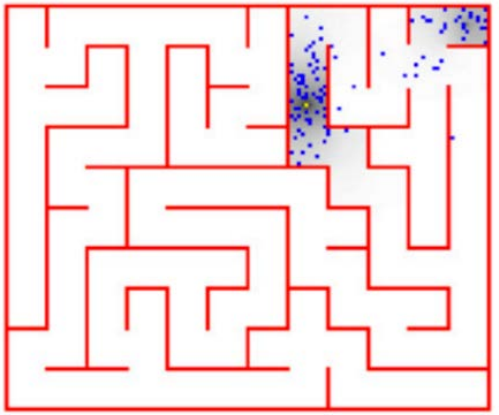
- 1-d, 2-d, 3-d?
- Rules for movement? Randomness?
- Probability / expected value
- Vectors
  - Position: $\mathbf{X}=\langle x(t),y(t)\rangle$
  - Movement: $\mathbf{X}_{new}=\mathbf{X}_{old}+t\mathbf{V}$

    where **V** is the velocity, which depends on?



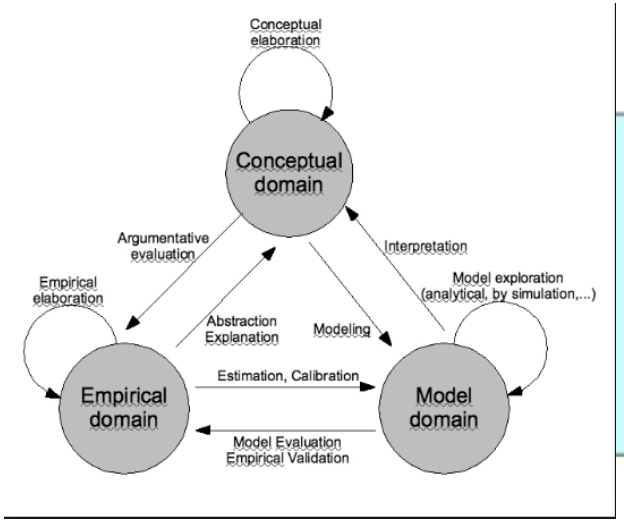
Movement

- Distributions
  - Poisson, von Mises, wrapped normal, wrapped Cauchy, circular distributions, normal, …
- Biased motion
  - Bias time, location, angle, movement
  - Environmental conditions
    - Crowding, fluid flow, chemicals, food, ….

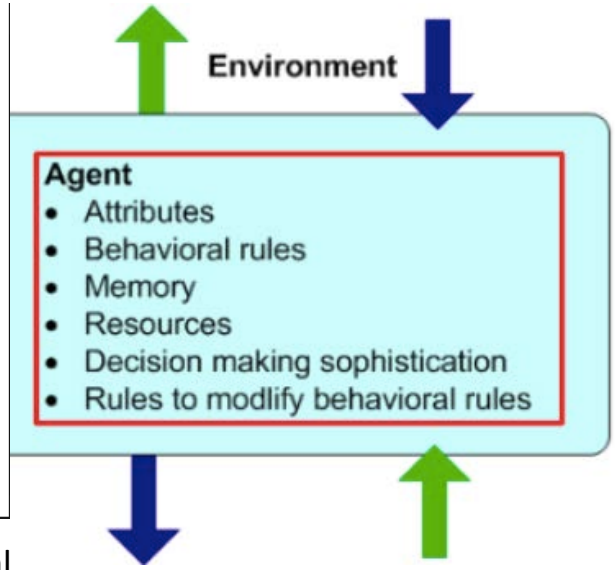


Agent Based Models

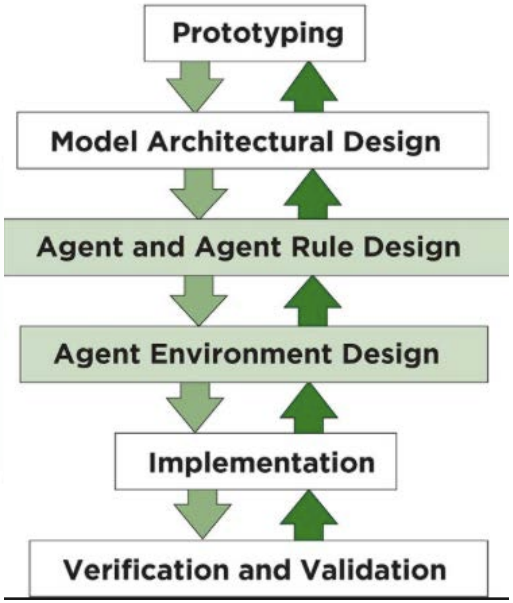
- What goes into an agent based model?



http://jasss.soc.surrey.ac.uk/13/1/3.html



https://www.smartcaptoolbox.com/?page_id=36



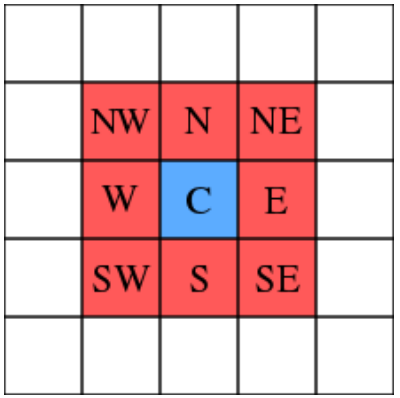
https://www.fhwa.dot.gov/advancedresearch/pubs/11036/index.cfm

Agent Based Models

- Back to the basics…
- Cellular Automaton
  - On a grid
  - Discrete set of time steps
  - Agents fixed at each grid point
  - Iterate through time

John Conway's Game of Life

- Developed in 1970
- Zero-player game: determined by initial conditions and rules of the "agents" determine the evolution
- Agents are on a 2-d grid in one of two states:
  - Alive or Dead
  - Moore neighborhood to determine state changes



Rules for Game of Life

1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by over-population.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

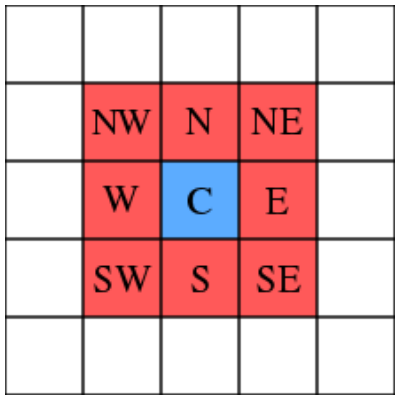
Game of Life

- State space: $\Sigma = \{0, 1\}$ where 0=dead, 1=alive
- No movement of agents
- At t=0, initialize every point on a 2-d grid with a 0 or 1
- Transition rule for state changes at each time iteration:
  - if $s_{i,j}^t = 1$

$$s_{i,j}^{(t+1)} = \begin{cases} 1 & \text{either 2 or 3 neighbors} = 1 \\ 0 & \text{otherwise} \end{cases}$$

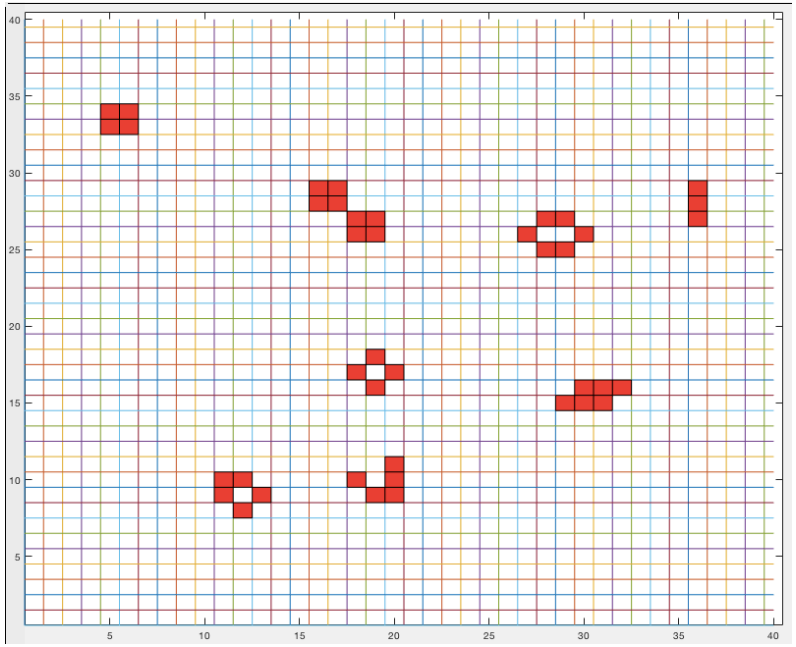
  - if $s_{i,j}^t = 0$

$$s_{i,j}^{(t+1)} = \begin{cases} 1 & \text{exactly 3 neighbors} = 1 \\ 0 & \text{otherwise} \end{cases}$$



Game of Life

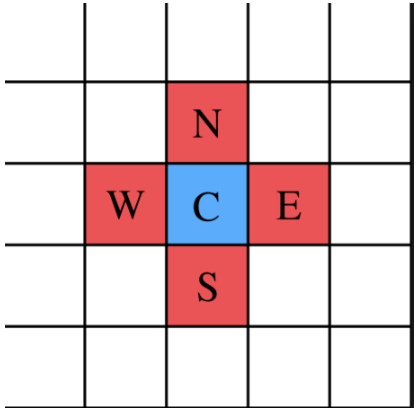
- Code



Patterns in Game of Life

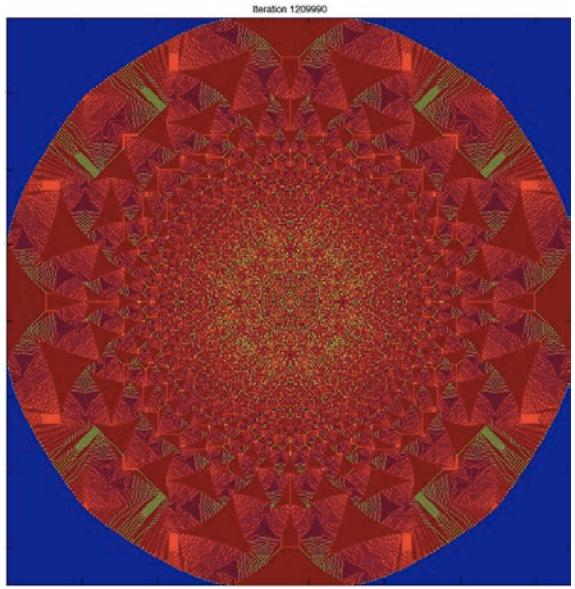
- Still lifes
- Oscillators
- Spaceships
- Wikipedia
- Simple rules can lead to:
  - Interesting math to formalize and analyze
  - Fun patterns!

Abelian Sandpile



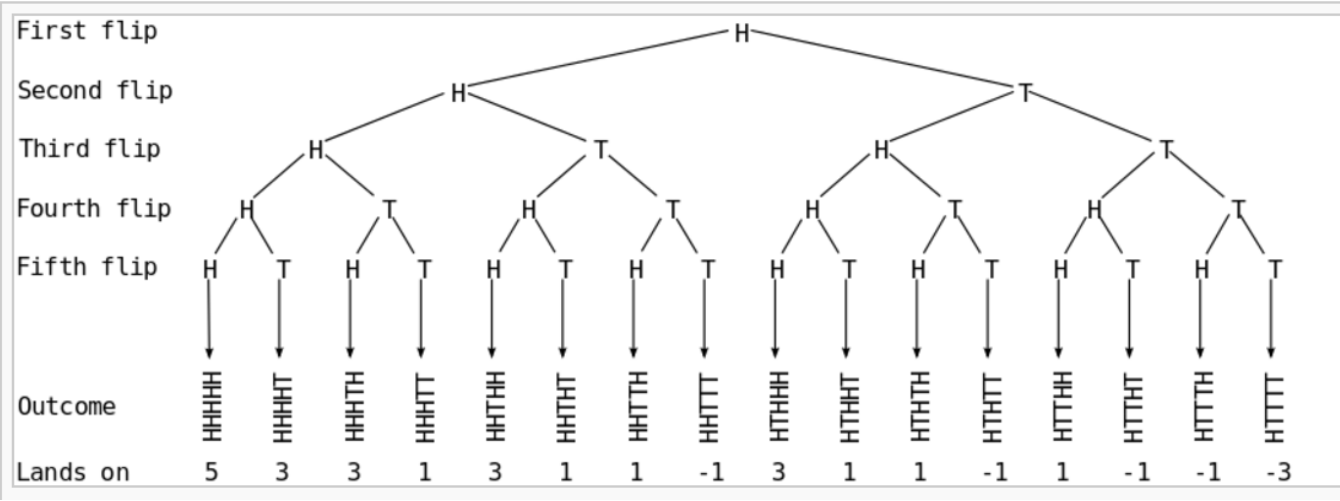
https://commons.wikimedia.org/wiki/File:Von_neumann_neighborhood_with_cardinal_directions.svg

- 2-d Cellular Automaton
- Begin with a certain number of grains of sands at the origin at time zero
- At each lattice point:
  - If # of grains >3 then distribute one grain to each of four neighbors
  - Stop at steady state



1-d Agent Based Model

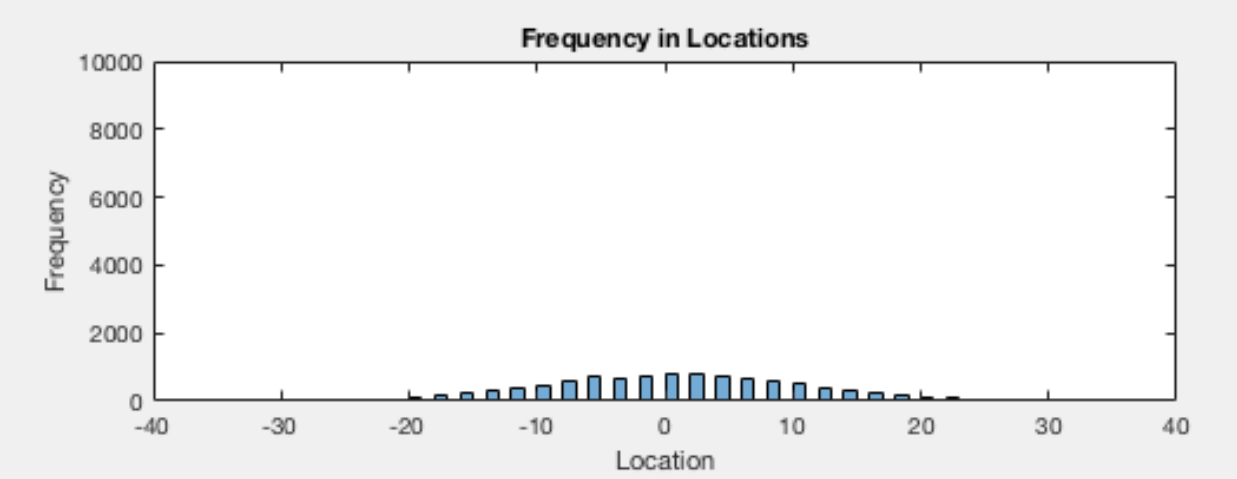
- Starting simple: a single agent in a single state that moves on a 1-d grid or line.
- At each iteration, the agent flips a coin. Move left 1 box if T and move right 1 box if H



https://en.wikipedia.org/wiki/Random_walk

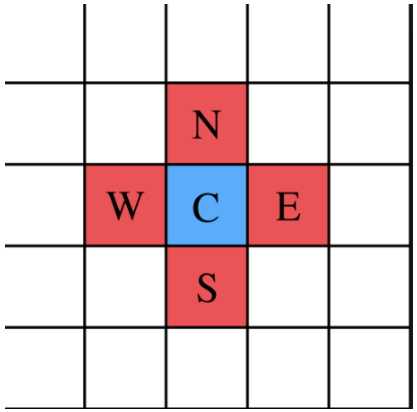
1-d Agent Based Model

- Code
- Expected Value and Law of Large Numbers
- Gambler's ruin or Gambler's fallacy
- Random Process with independent trials



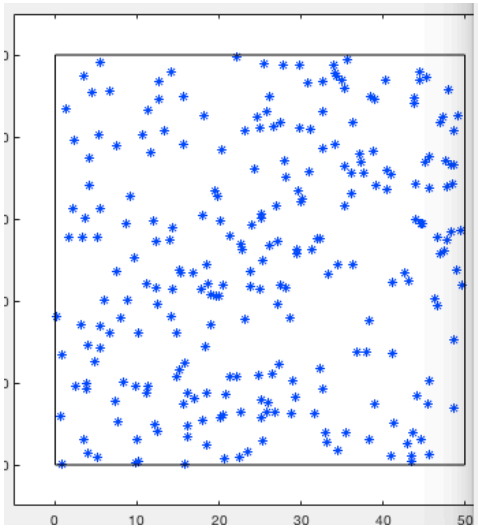
2-d Agent Based Model

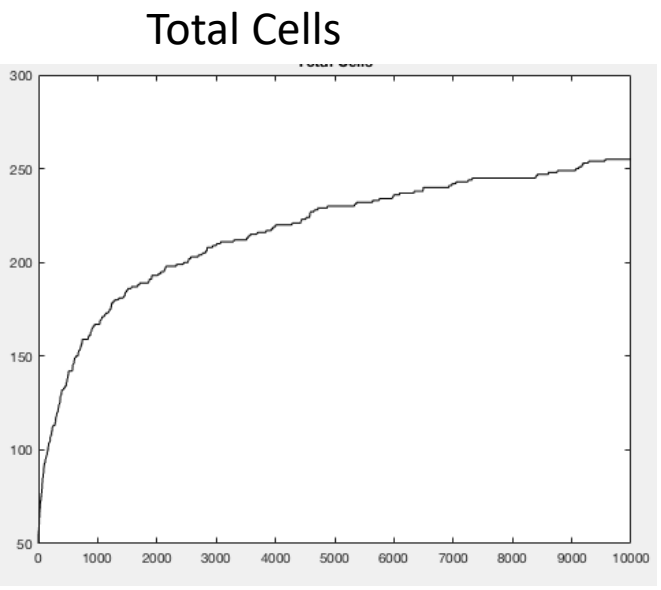
- On a 2-d grid
- Von Neumann neighborhood
- Code



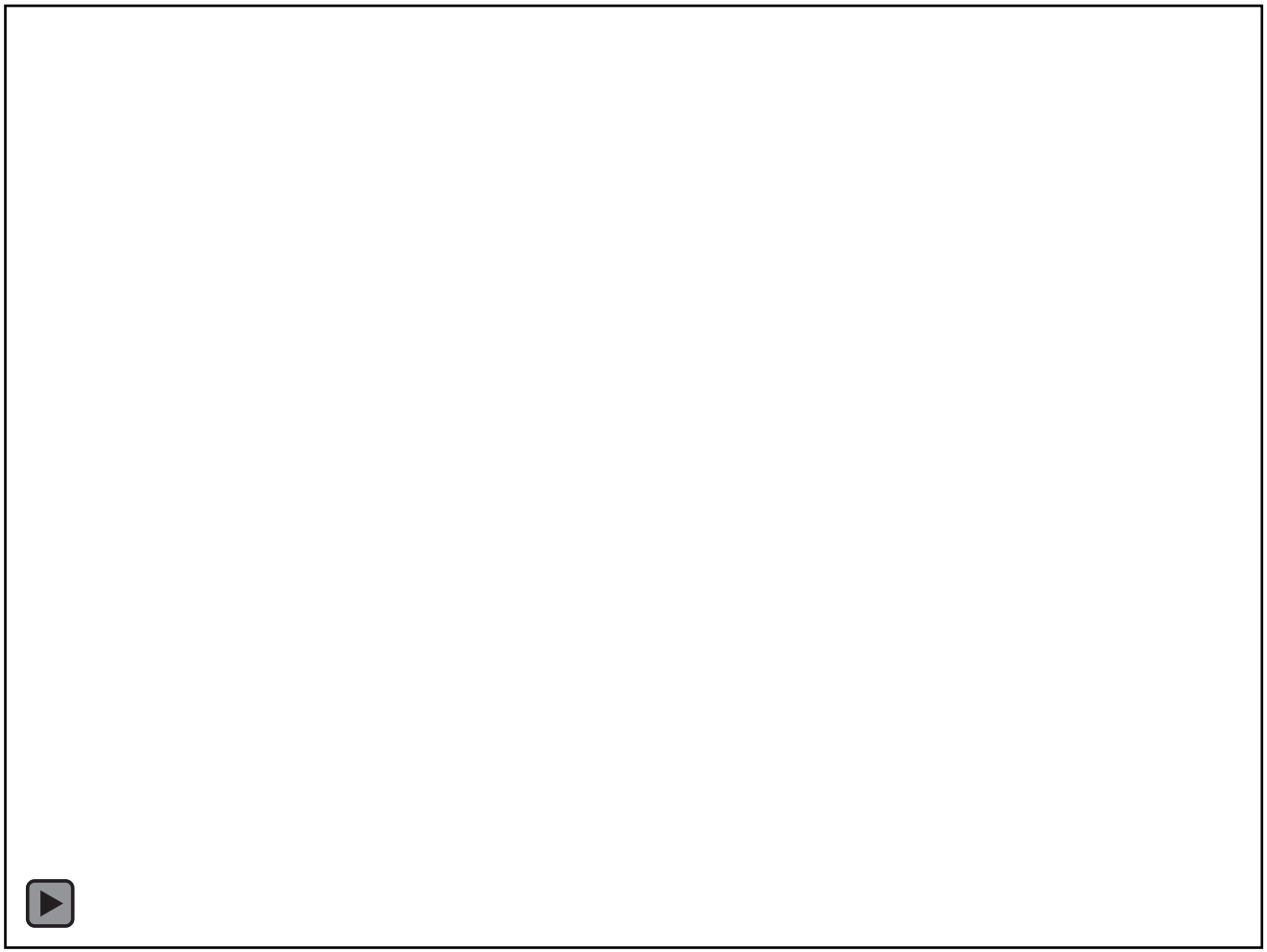
Cell Proliferation

- Cells in an experiment can reproduce and move
- Move freely, confined to stay within the domain
- Code





Cells Absorbing Particles

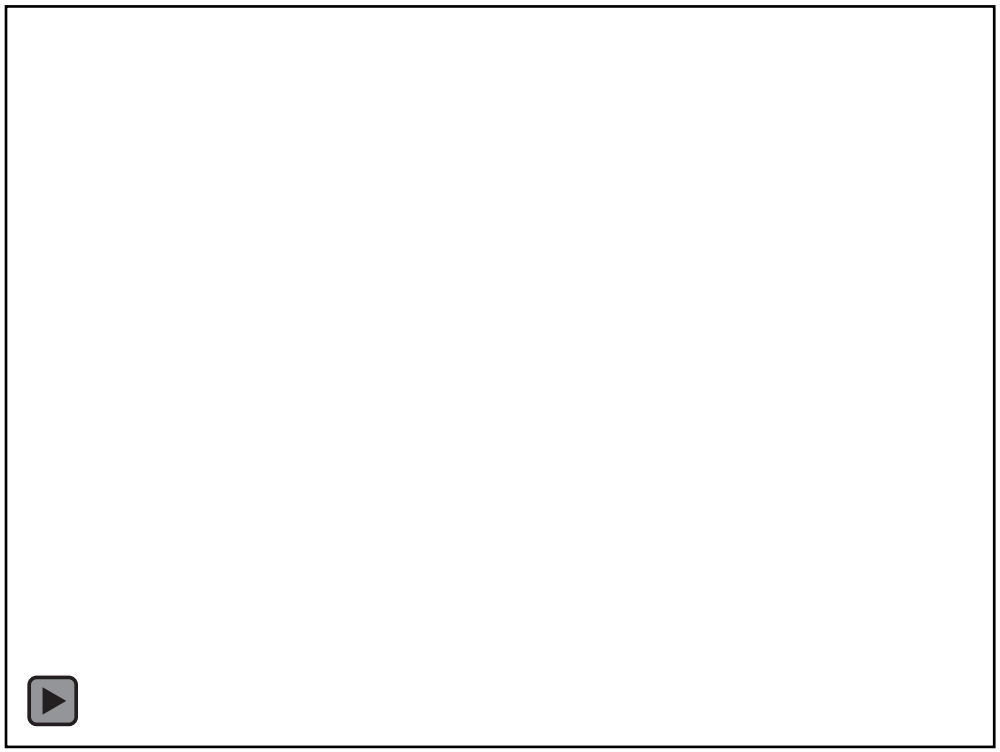


Spread of an epidemic

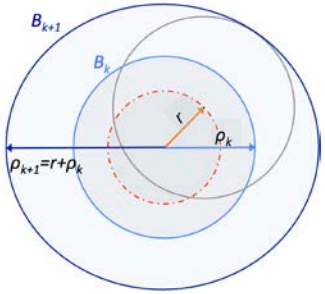
- e.g. Flu or Ebola virus
- Start with healthy population and introduce one sick person into the population
- Probability: of getting sick, recovering, dying
- Rules: Get sick if come into contact with other sick individuals

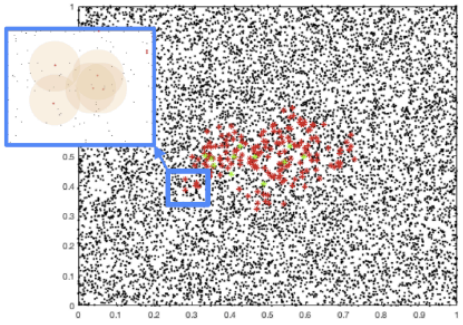
SIR Model

● Susceptible ● Infected ● Recovered

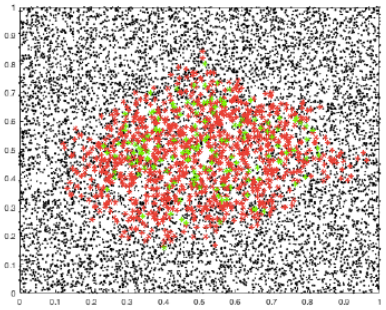


SIR Model

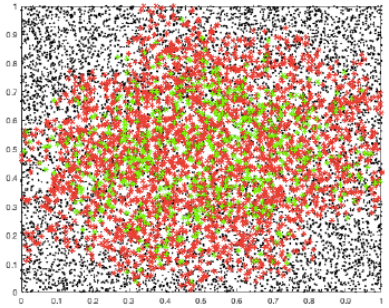




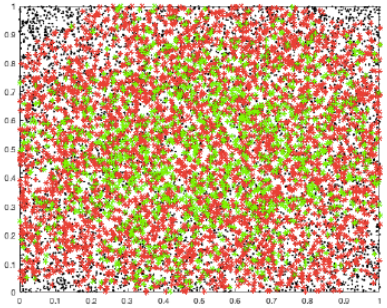
$t = 10$



$t = 20$



$t = 30$



$t = 40$

For $s_k^t = \mathcal{S}$:

$$f(s_k^t) = \begin{cases} \mathcal{I}_1 & : \text{if } \mathbf{x}_k^t \in B_{\mathcal{S},\mathcal{I}_1}^t \text{ and } K < X, \\ & \quad \text{where } X \sim \text{Uniform}[0,1] \\ \mathcal{S} & : \text{otherwise} \end{cases}$$

For $s_k^t = \mathcal{I}_j, \exists j = 1, 2, ..., T_I$:

$$f(s_k^t) = \begin{cases} \mathcal{I}_{j+1} & : \text{if } 1 \le j < T_I \\ \mathcal{R}_1 & : \text{if } j = T_I \end{cases}$$

For $s_k^t = \mathcal{R}_m, \exists m = 1, 2, ..., T_R$:

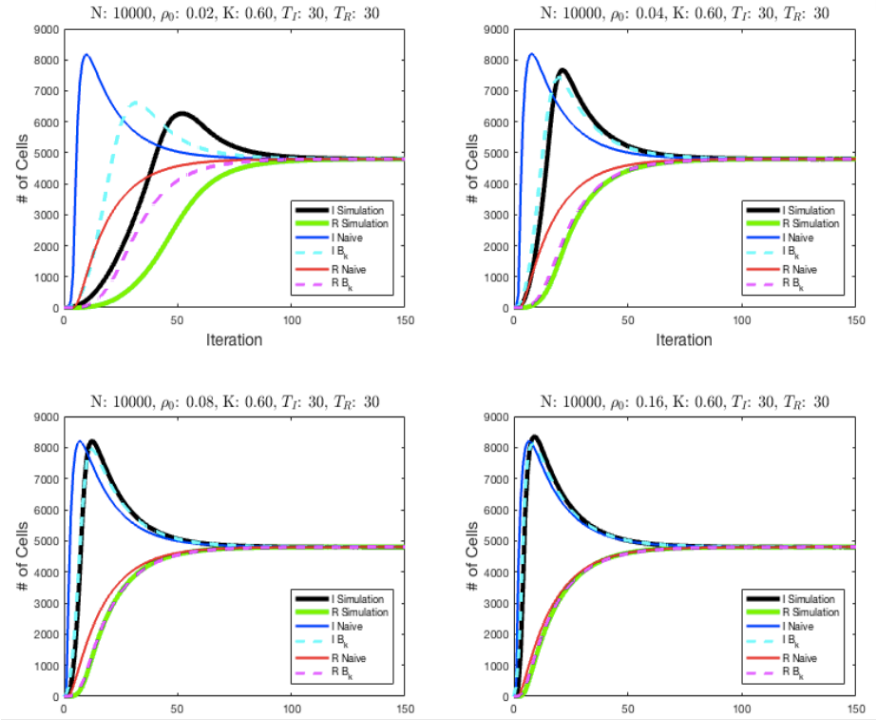
$$f(s_k^t) = \begin{cases} \mathcal{R}_{m+1} & : \text{if } 1 \le m < T_R \\ \mathcal{S} & : \text{if } m = T_R \end{cases}$$

T_I=total infection time, T_R=*total recovery time,* K=contact tolerance

Analyzing Steady States

- $\mathcal{U} \in \Sigma$ be a state
- U_t: the number of cells in state $\mathcal{U}$ at iteration t
- $B_{\mathcal{U}}$: the $\mathcal{U}$ transition neighborhood
- $W(v \to \mathcal{U})$: probability cell v transitions to state $\mathcal{U}$

$$\begin{aligned}\mathbb{E}(U_{t+1}) &= \sum_{v \in \mathcal{X}} \mathbb{P}(v_{t+1} \in \mathcal{U}) \\ &= \sum_{\mathcal{V} \in \Sigma} \sum_{v \in \mathcal{V}} \mathbb{P}(v_{t+1} \in \mathcal{U}) \\ &= \sum_{\mathcal{V} \in \Sigma} \sum_{v \in \mathcal{V}} \mathbb{P}(\ell_{v_t} \in B_{\mathcal{U}}) W(v_t \to \mathcal{U})\end{aligned}$$



For our rules in the CA model, we get:

$$\tilde{I}_{t+1} = (N - \tilde{I}_t - \tilde{R}_t)\left\{1 - \left(1 - \frac{\mu(\mathcal{N})}{\mu\left(B^t_{\mathcal{S},\mathcal{I}}\right)}\right)^{\tilde{I}_t}\right\}\frac{\mu\left(B^t_{\mathcal{S},\mathcal{I}}\right)}{\mu(\Omega)}(1-K) + \left\{1 - \frac{1}{T_I}\right\}\tilde{I}_t =: \tilde{H}(\tilde{I}_t, \tilde{R}_t)$$

$$\tilde{R}_{t+1} = \frac{1}{T_I}\tilde{I}_t + \left\{1 - \frac{1}{T_R}\right\}\tilde{R}_t =: \tilde{G}(\tilde{I}_t, \tilde{R}_t)$$

SIR Models

- Alternate approaches to SIR Models
- Differential equations – could analyze steady states…

$$\begin{aligned} \frac{dS}{dt} &= -\beta \frac{SI}{N} \\ \frac{dI}{dt} &= \beta \frac{SI}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I \\ N &= S + I + R \end{aligned}$$

Analysis and Comparison to Other Models

- Can be used to understand collective motion
- For what rules do we observe dynamic patterns? Wave propagation?
- Why use these models?

AB or CA Models

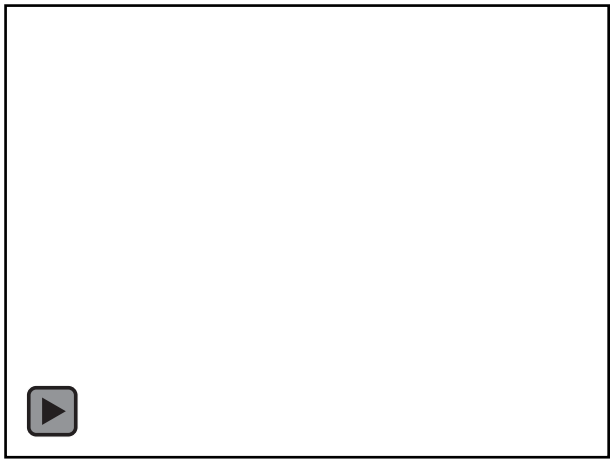
Pros	Cons
Local solution	Needs to run sufficiently many times
Decrease step size increases time complexity, not memory	Expensive to find global solution
Very easy to code (even with strange geometries)	Difficult to analyze parameter effects
Relatively easy to couple with other equations	

PDE Models

Pros	Cons
Many established solution methods	Analytic solution difficult with strange geometries, B.C.
Closed form solution with simple geometries	Numerical Scheme solves full solution (expensive with memory)
Easier to find: MFPT over entire region, Parameter Sensitivity, Parameter Estimation	Difficult to make numerical scheme stable/accurate with strange geometries or BCs
Rigorously formulate applications	Very difficult to couple with other equations

Collective Motion

- Deriving a Global Recurrence Rule and analyzing stability allows us to determine behavior in a stochastic system of agents
- Other analysis…
  - How long does it reach an agent or a group of agents to reach a given location?
  - Application of MFPT – Mean First Passage Time



Mean First Passage Time in 1D

- Line from 0 to 1 and divide the line into N segments $0, \delta, 2\delta, ..., N\delta$
- At each iteration x has probability $\ell(x)$ moving left and $r(x)$ moving right

$$\text{Note that } \forall x \in [0,1], \ell(x) + r(x) = 1.$$

- We then have a distribution of random variables X_k denoting locations of x at iteration k.
- $T(x)$: mean time for x to escape through the boundary $x = 0$ or $x = 1$.
- τ: iteration size (i.e. time step)

$$\begin{aligned} T(x) &= \mathbb{E}\{t : X_0 = x\} \\ &= \mathbb{E}\{t : X_1 = x - \delta, X_0 = x\} + \mathbb{E}\{t : X_1 = x + \delta, X_0 = x\} \end{aligned}$$

MFPT 1D

Let Y_k be the random variables defined by $Y_k = X_{k+1}$. Note:

- $T_Y(y) = T(x) - \tau$.
- Distribution of $\{X_k\}$ is same as $\{Y_k\}$

$$\begin{aligned} T(x) &= \mathbb{E}\{t : Y_0 = x - \delta\} + \mathbb{E}\{t : Y_0 = x + \delta\} + \tau \\ &= \ell(x)T(x-\delta) + r(x)T(x+\delta) + \tau \end{aligned}$$

If we assume $\delta \ll 1$ we can expand $T(x-\delta)$ and $T(x+\delta)$ by Taylor Series expansions.

$$\begin{aligned} T(x) = \tau &+ \ell(x)\Big(T(x) - \delta T'(x) + \frac{\delta^2}{2}T''(x)\Big) \\ &+ r(x)\Big(T(x) + \delta T'(x) + \frac{\delta^2}{2}T''(x)\Big) + \mathcal{O}(\delta^3) \end{aligned}$$

$$\boxed{-\tau = \Big(r(x) - \ell(x)\Big)\delta T'(x) + \Big(r(x) + \ell(x)\Big)\frac{\delta^2}{2}T''(x)}$$

If probability to move left and right is ½ and escaping through x=0 and x=1:

$$\begin{cases} \frac{-2\tau}{\delta^2} = T''(x) & : x \in (0,1) \\ T(x) = 0 & : x \in \{0,1\} \end{cases}$$

Escape Time or MFPT on Unit Disk

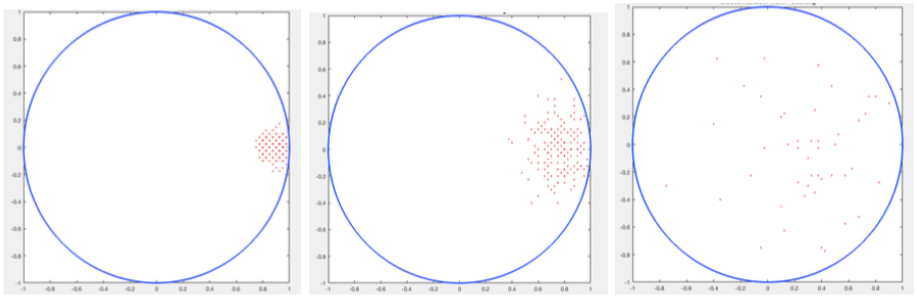
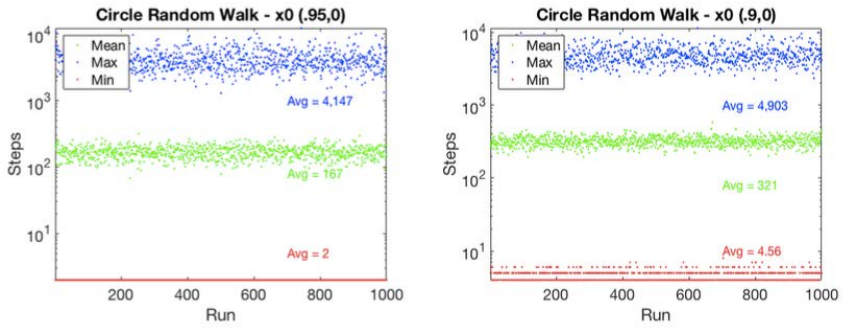


Fig. 5.3: Simulation of random walk on disk after 10, 100, 1000 iterations for MFPT estimation

$$\begin{cases} \nabla^2 U = \frac{-4\tau}{\delta^2} & : \mathbf{x} \in B(0,1) \\ U = 0 & : \mathbf{x} \in \partial B(0,1) \end{cases}$$



x_0	(.95,0)	(.9,0)
Analysis	156	304
Simulation	167	321

Summary

- Agent Based Models
  - Useful!
  - Involves probability, statistics, vectors, …
  - Could lead to interesting math analysis
  - Computationally intensive!
  - Model development and rules depends on application
- Free coding platforms
  - Python and Octave
  - NetLogo