Complex Systems 530: Computer Modeling of Complex Systems

Lecture 2: Intro to agent-based models
1/14/20

Readings!

- Read
  - “Why Model?”, Epstein 2008
  - Sayama, Chp. 1-2
  - “More is Different,” Anderson 1987
  - Wilensky, Chp. 0-1
- Check-in: Everyone has NetLogo & Python installed?

How to choose a modeling framework?

- Differential equations, stochastic population-based models, agent-based models, many others!
- Deterministic vs. stochastic?
- Discrete vs. continuous?
- Population-based or individual-based?
- Spatial/non-spatial?
- Different frameworks will have different analytical and computational tractability, interpretability, and assumptions

How to choose a modeling framework?

- Often can model the same process with many different frameworks (ABMs, ODEs, Markov models, etc.)
- Can also sometimes implement the same or equivalent model in different frameworks
- Discuss for:
    - Infectious disease epidemics
    - Population growth (e.g. birth/death processes)
    - Swarming/flocking (e.g. murmuration, fish schooling)

How to choose a modeling framework?

- Depends on the problem/question of interest!
- ABMs often particularly advantageous for questions where individual heterogeneity is key (e.g. spatial position of individuals, individuals with varying properties, etc.)
- May also just be more illustrative/clear/interpretable in one framework or another even if equivalent

How to choose a modeling framework?

- What if the modeling framework you choose affects your results?
- More generally, how to decide how realistic/ simplified to make your model?
- Model comparison & inference robustness assessment—more on this later

Koopman JS. Infection transmission science and models. Japanese journal of infectious diseases. 2005 Dec 1;58(6):S.
Pollock KH. Inference robustness vs. criterion robustness: an example. The American Statistician. 1978 Nov 1;32(4):133-6.

Top-down vs. bottom-up

- **Top-down**: start with the understanding of the larger system, then break-down or decompose into smaller subsystems

- **Bottom-up**: start with micro-level processes and build up to emergent behaviors at the macro level

Top-down vs. bottom-up

- Related to Hayek's (1973) consideration of two different Greek conceptions of "order":
  - **Taxis**: An arranged, top-down order. A "made" or "designed" order, purposefully built and imposed by a part onto the greater whole.
  - **Cosmos**: A grown, bottom-up order. An order that arises spontaneously and unintentionally from the interaction of parts within a whole.

Top-down vs. bottom-up

- Many models & modeling frameworks can be built/ thought of from either perspective
- Advantages/disadvantages of each?
- What about in the context of complex systems?
- Top-down/bottom-up often conflated with simple models vs. detailed ones, but not quite true

Agent-based models (ABMs)

Agent-based models have 3 main components

- **Agents** - independent “agents” move, interact, explore environment, etc.
- **Environment** - agents exist in a non-agent environment (can be static or dynamic)
- **Rules/interactions** - to govern agent behavior, how they interact with the environment, etc.

Motivating example: video games!

ABM Advantages

- Can handle situations where population cannot be viewed as aggregates
  - Heterogeneity
- Often a natural description of system - think in terms of individuals & their decisions/actions
  - Builds from micro (individual process) to macro (overall emergent behavior)
- Flexible, can account for more complexity/detail

ABM Disadvantages

- Often harder to develop, document, and validate
- Fewer analytical tools for understanding dynamics, parameter estimation from data, etc.
- Flexibility towards complexity/complicatedness is both a strength & weakness
    - Can be tempting to make highly complicated & realistic—can make it difficult to know what's going on! Be careful to keep it 'as simple as possible (but not simpler)'

ABM Dynamics

- Hard to build general mathematical theory of ABM dynamics
  - Not always so easy to classify
  - Not necessarily equilibrium values to calculate
  - Phase plane ideas may not be helpful b/c of spatial aspect, etc.

ABM Dynamics

- Wide range of possibilities
- Stable constant steady states
- Repeating patterns (oscillation)
- Organized but non-repeating structure/patterns
- Disorder
- All sorts of things!

We will explore

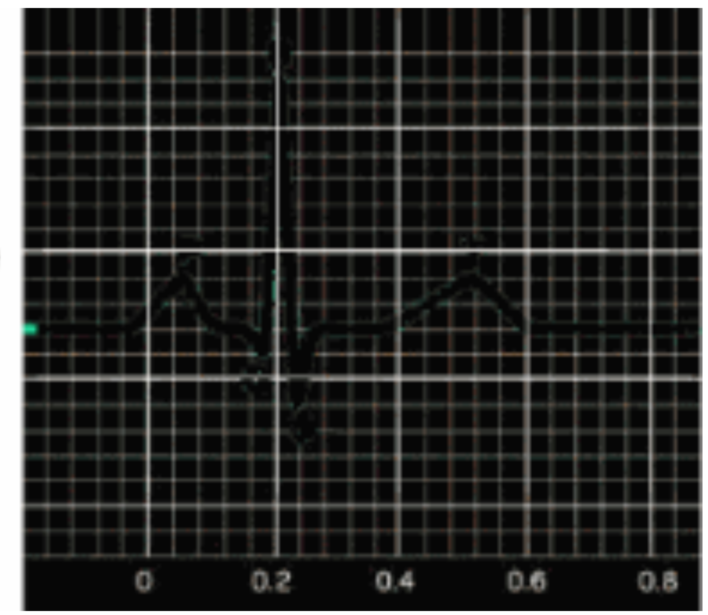
- Designing, building, and simulating ABMS
- Parameter exploration & sampling - exploring the model behavior
- Interpreting results
- Documenting ABMs!
- Alternative models & inference robustness - how do the inferences/predictions/outcomes/explanatory power of our model change as we add realism?

Let’s try out an ABM!

- Forest fire model: https://ncase.me/simulating/model/
- What do you notice?
- Any interesting behaviors as you adjust things? You can also add other agent types into the mix!
- What happens if we increase tree growth to 5% and start with all trees?
- Modeling by analogy: what else could this model (or similar) be used for?

Cardiac dynamics & heart muscle tissue

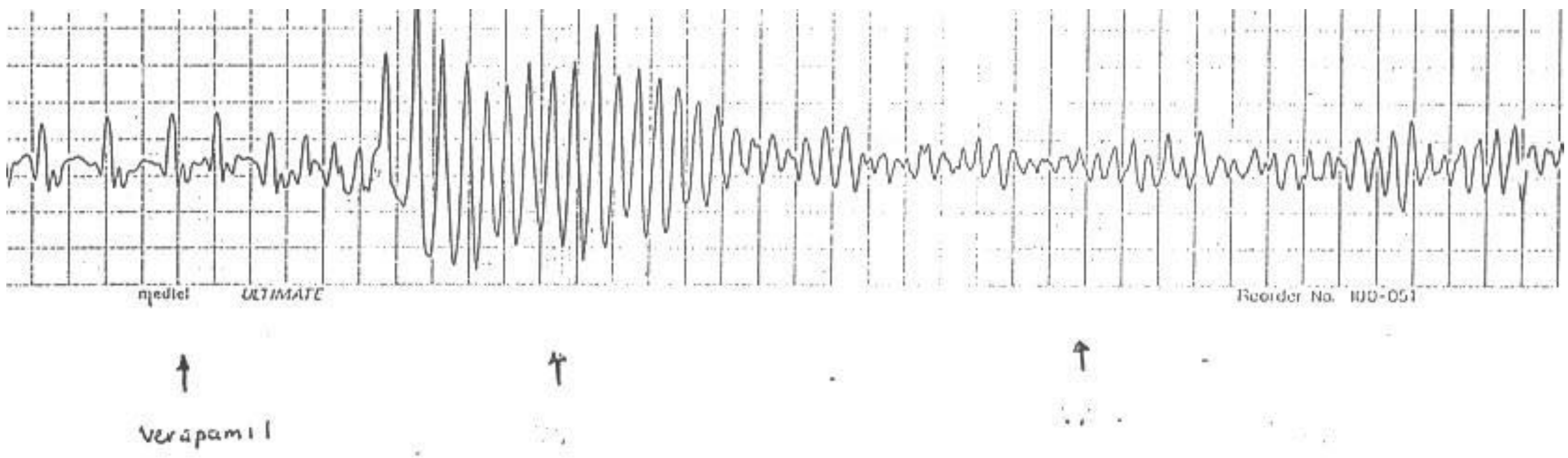
- Normal rhythm: depolarization wave contracts atrium and ventricle in a regular rhythm, originating at sinoatrial node & traveling to atrioventricular node



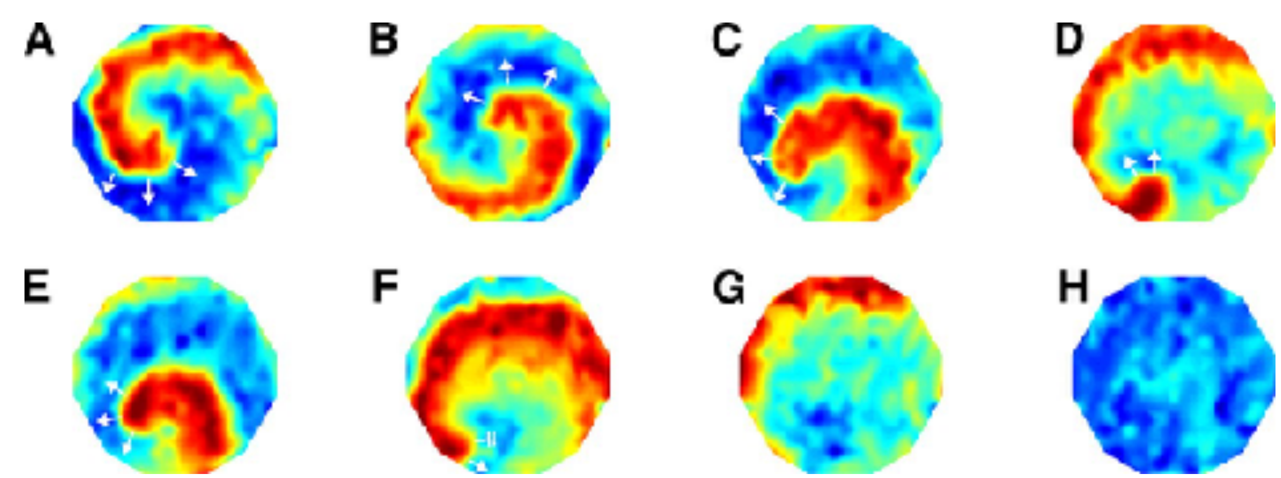
http://www.scholarpedia.org/article/Cardiac_arrhythmia

- However, heart can undergo bifurcation to other behaviors—arrhythmias

Cardiac arrhythmias

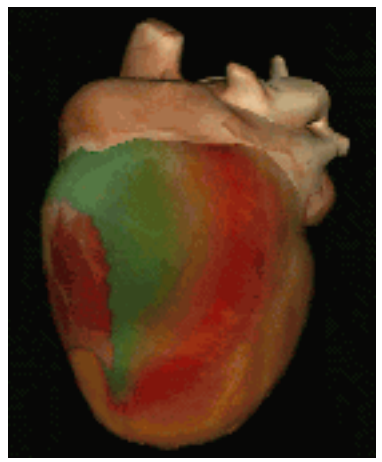


- Tachycardia—>ventricular flutter—>ventricular fibrillation
- These bifurcations can often be understood in terms of spiral waves, period doubling bifurcations, and other spatiotemporal dynamics

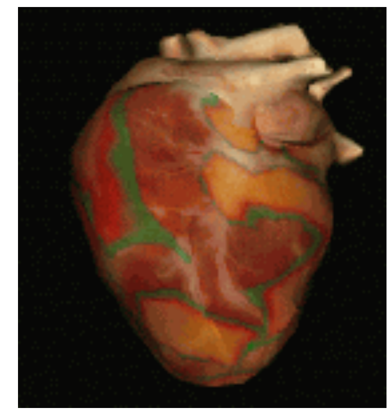


http://ajpheart.physiology.org/content/294/1/H58.figures-only

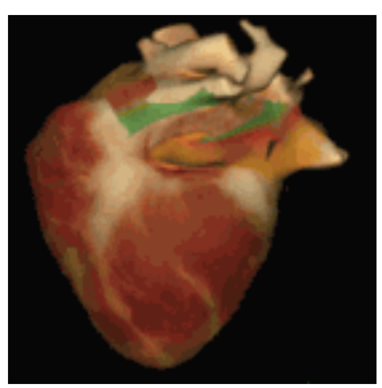
Spiral/scroll waves in heart arrhythmias



Ventricular tachycardia



Ventricular fibrillation



Atrial flutter

Atrial fibrillation

http://www.scholarpedia.org/article/Cardiac_arrhythmia

Now let's make one from scratch

- https://ncase.me/sim/?s=blank
- Simple voting model!
- 2 parties/candidates/options - everyone starts with some random initial preference
- They tally the planned votes of their neighbors and if more than half of neighbors are voting the other way, they switch
- What happens?

Wilensky, U. (1998). NetLogo Voting model. http://ccl.northwestern.edu/netlogo/models/Voting. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Voting model

- Things to try:
  - Award ties differently (e.g. switch if exactly half of neighbors vote the other way)
  - Different initial percentages of each party
- Try with 3 parties
  - What happens? Same patterns?

More emoji ABMs

- https://ncase.me/sim/
- Explore!

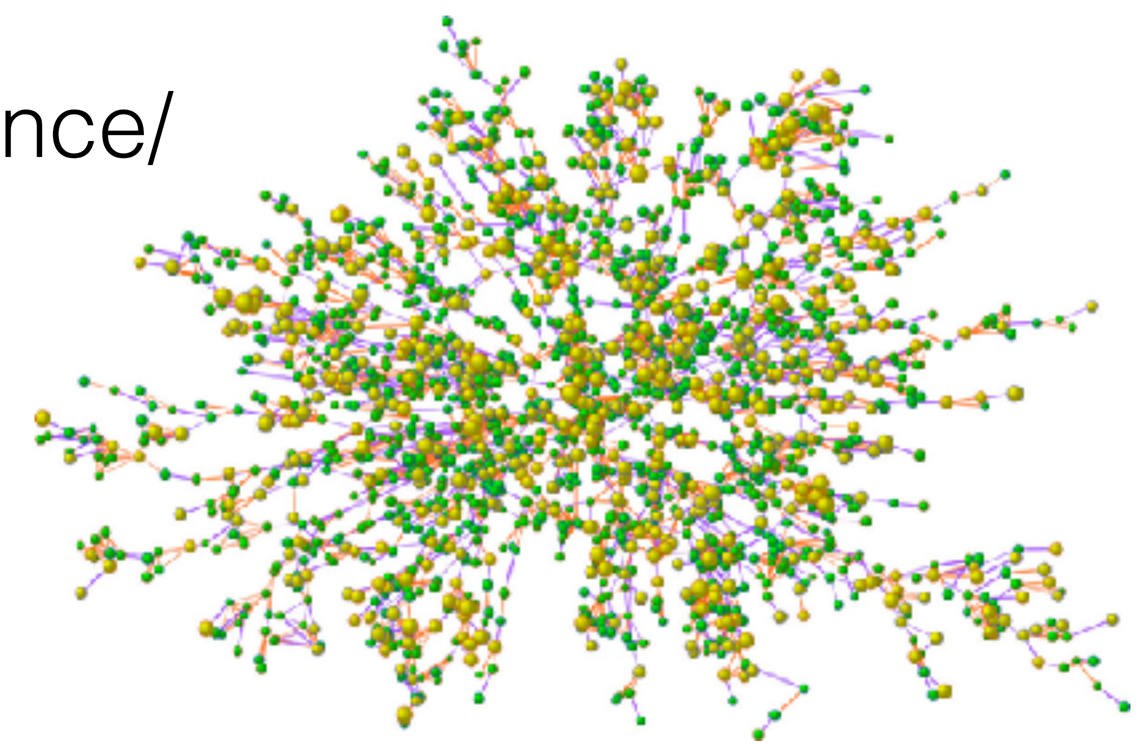
Examples & features of ABMs

Interactions

- Infectious disease models - account for different mixing patterns, school, work, etc.

- Evolution & invasion of strategies, ideas, memes, species

- Models of markets, voting, violence/ unrest, etc.



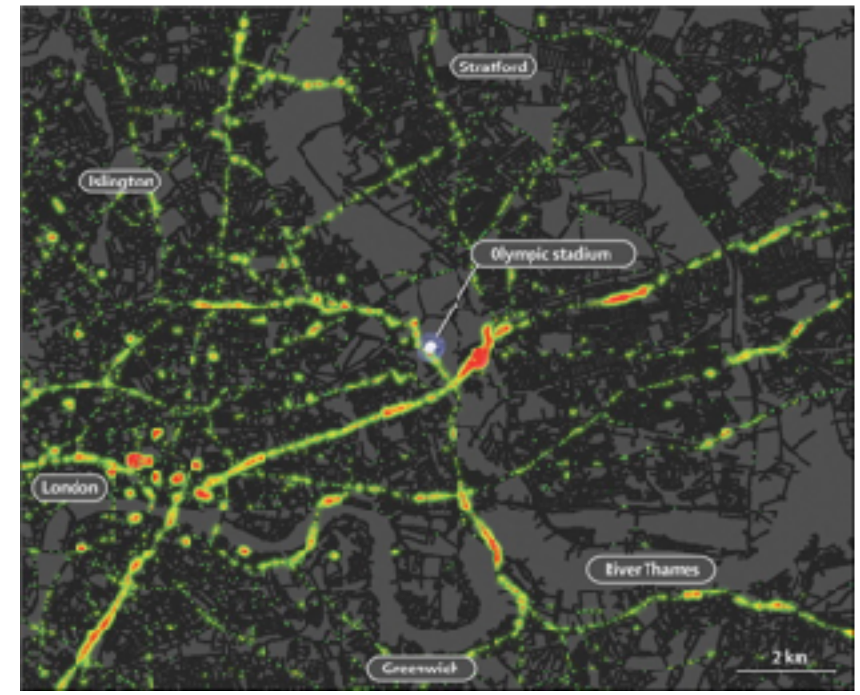
- Many dynamic social network models are ABMs too

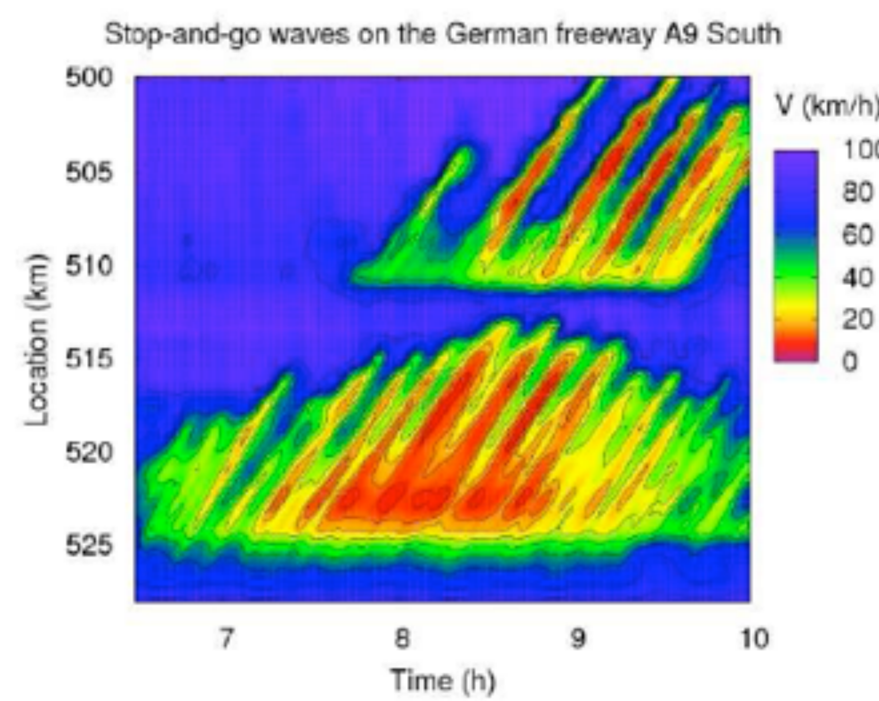
Examples

- Netlogo virus on a network model
- Sugarscape - classic model of interactions between individuals and the environment in a society with spatially distributed resources

Flows & movement

- Swarming, crowd behavior, building evacuation
- Traffic flow - simple rules & shockwaves
- Walking, commuting, migrations, etc.
  - E.g., How does neighborhood structure & distribution of resources affect health?

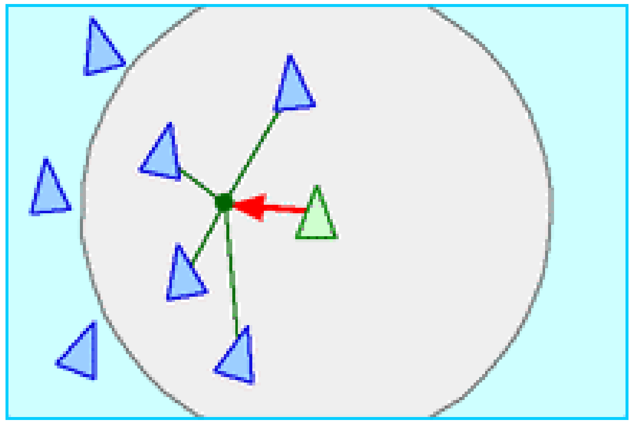




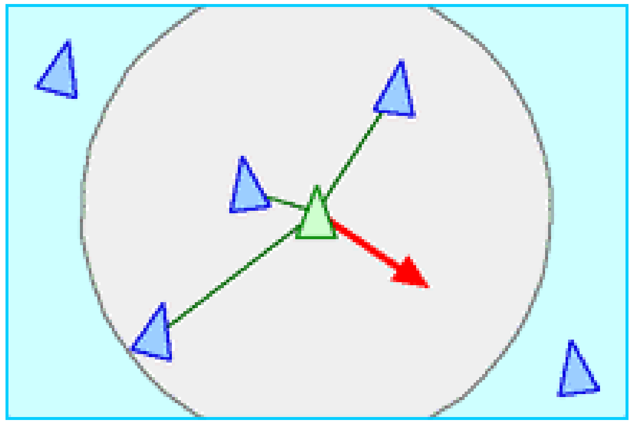
http://www.youtube.com/watch?v=GUkjC-69vaw

http://www.youtube.com/watch?feature=player_embedded&v=PRRJM-Xsg-U

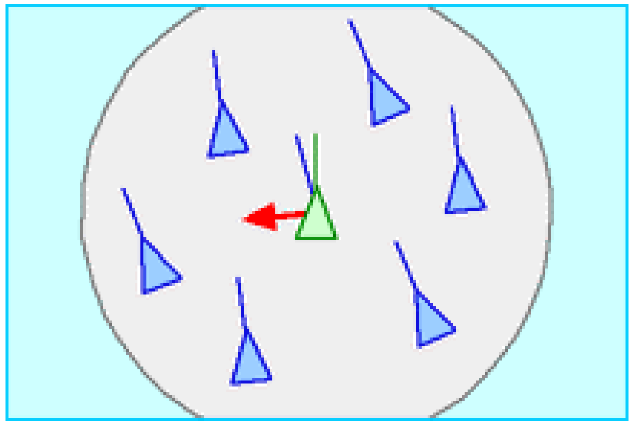
Boids



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



Separation:
Steer to avoid crowding local flockmates



Alignment:
Steer towards the average heading of local flockmates

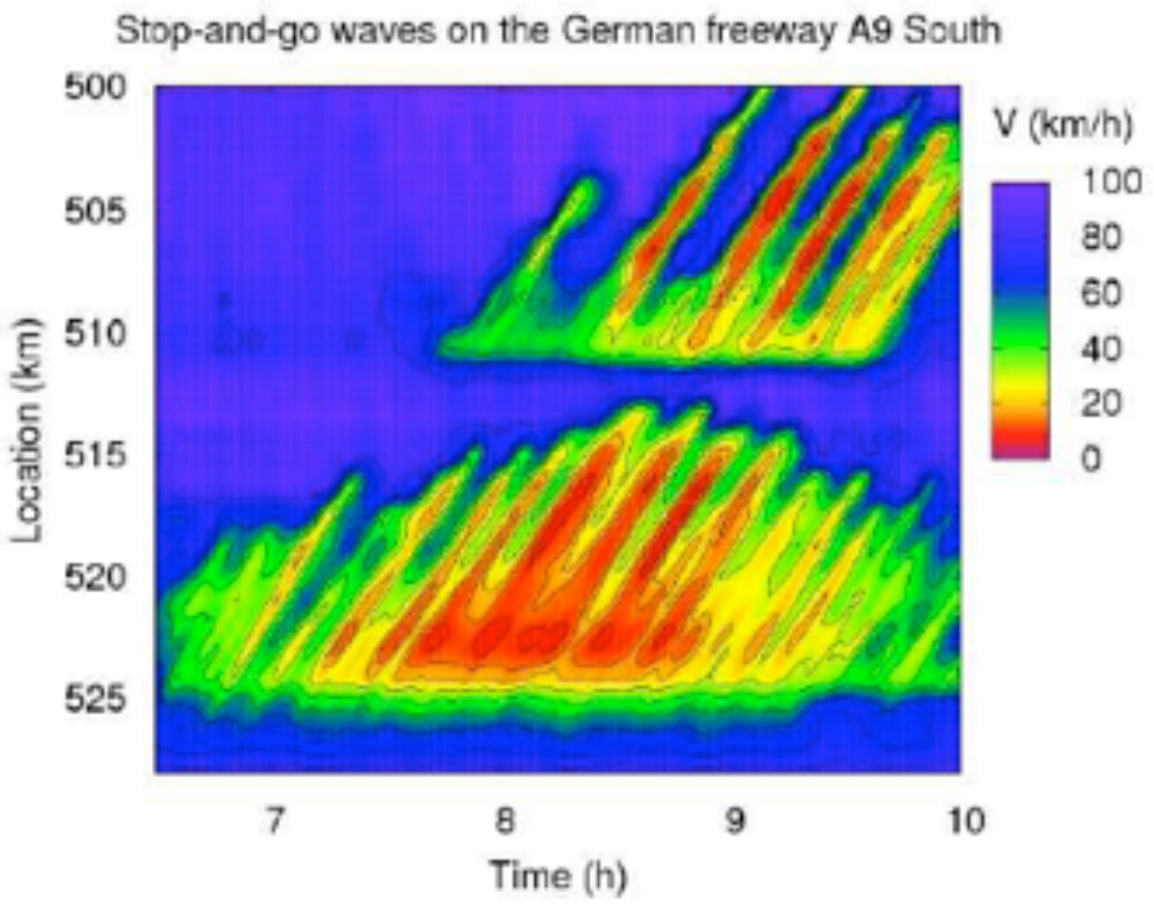
Source: “Boids” by Craig Reynolds, http://www.red3d.com/cwr/boids/

Boids/flocking models

- Leap example
- Netlogo Flocking model

Traffic Flow

- https://traffic-simulation.de
- Netlogo traffic model



Building ABMs: lots of choices!

- What types of agents? How many? How distributed? What properties do they need?
- What type of environment? Grid? GIS? 1D/2D/3D?
- Major issues/interactions/processes to consider? (movement? agent interactions? etc.)

Building ABMs

- Major Design Decision:
  - Who will be interacting with whom and with what part of the environment?
- Relevant terminology:
  - **Agent's local environment**: local area around the agent
  - **Neighbors**: subset of agents with whom an agent interacts
  - **Neighborhood**: social or spatial areas in the "world" containing an agent's neighbors
  - **Topology**: how agents are connected to one another within the system

Building ABMs

- Agents, Environment, Rules/interactions
- Useful to start by laying out each of these
- Diagrams or tables often useful
- Reproducibility

PARTE Framework

- Properties, Actions, Rules, Time, Environment
- Agents are defined by their properties, actions, and rules

https://www.ncbi.nlm.nih.gov/books/NBK305917/

Features of Agents

- **Discrete, self-contained** - clear boundaries between agents (i.e. they are each distinct elements)
- **Properties or states** - possesses a set of attributes that can change via interactions and which drive its behavior
- **Autonomous** - typically functions independently, takes in local information and executes behaviors based on it
- **Social** - usually interactions of agents with one another influences their behaviors

Features of Agents

- May also be:
  - **Adaptive** - rules that modify behaviors based on accumulated experience (i.e. some form of learning)
  - **Goal-directed** - outcomes can be judged against whether they achieve a goal and future behaviors adjusted accordingly
  - **Heterogeneous** - agents within a model can vary on all previously described dimensions

Agent design is usually broken into two main pieces:

- Agent Properties or Attributes
  - Defines the state of the agent
  - Can be either static or dynamic
  - Captured via “<agent>-owned” variables (NetLogo) or via entries in a data structure (e.g. list, array, dictionary) or class attributes (Python)

Agent design is usually broken into two main pieces:

- Agent Actions or Methods
    - How state and interaction information translates into the behavior of the agent
    - Also used to update attributes
    - Captured via agent executed procedures (NetLogo) or via functions on agent-related data structures or class methods (Python)

Agent design

- Rules of the ABM and its agents govern how and when these properties (attributes) and actions (methods) occur

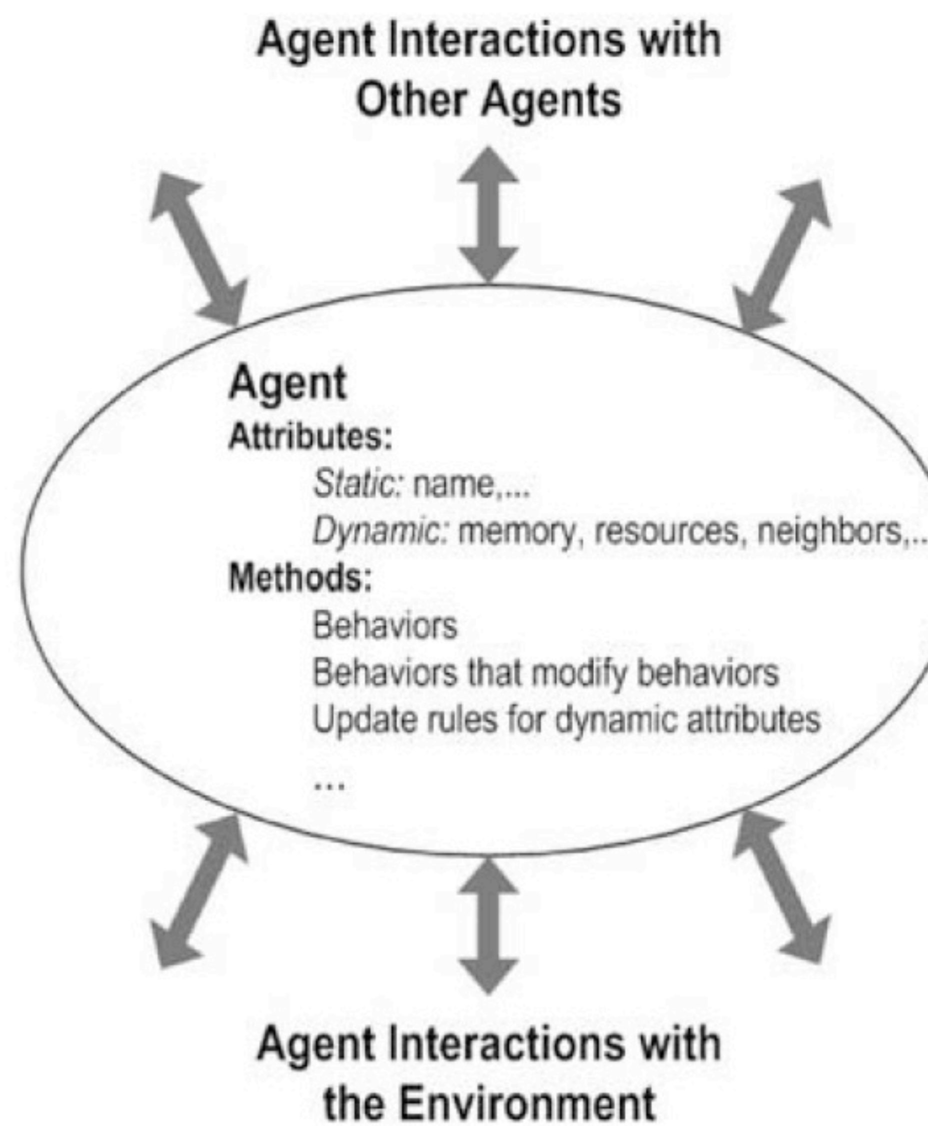
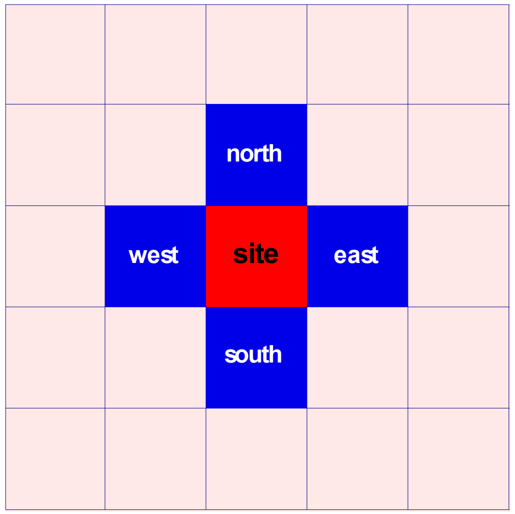


Figure 2 A typical agent.

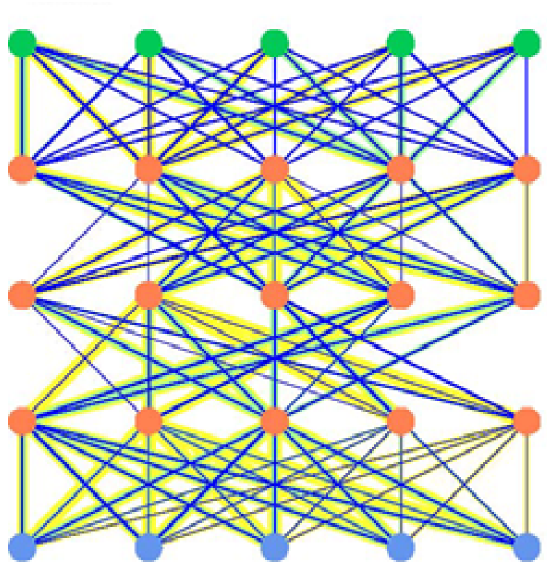
(Macal and North, 2010)

Environments

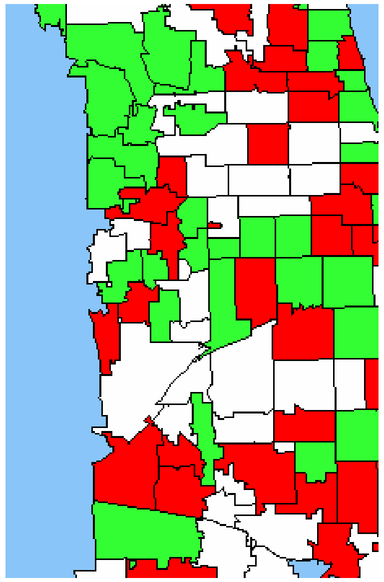
Euclidean Space: *2D, 3D*



Grid: *von Neumann neighborhood*



Network

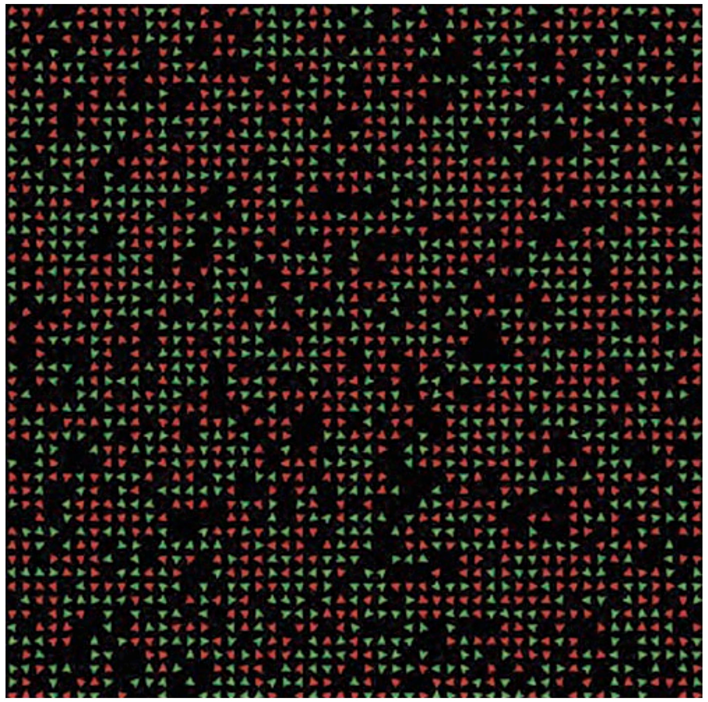
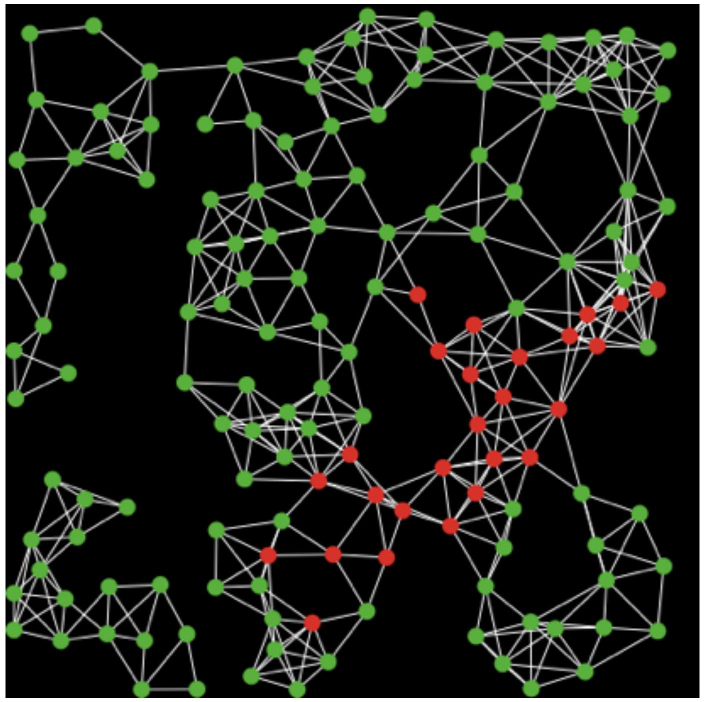
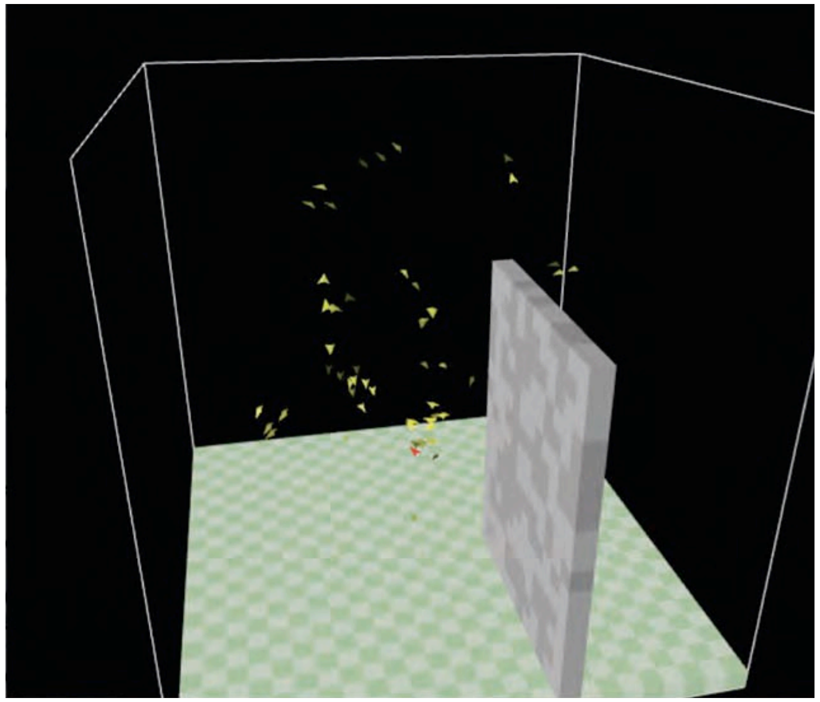
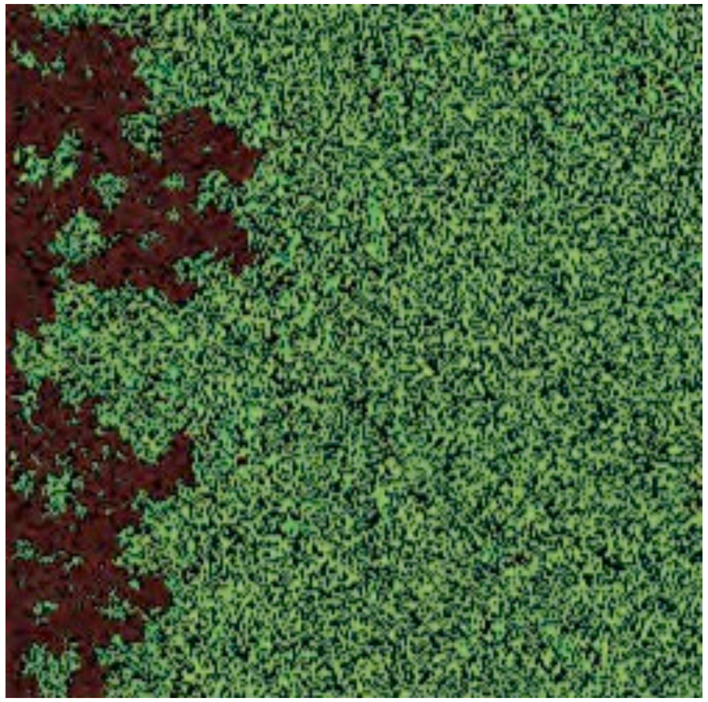
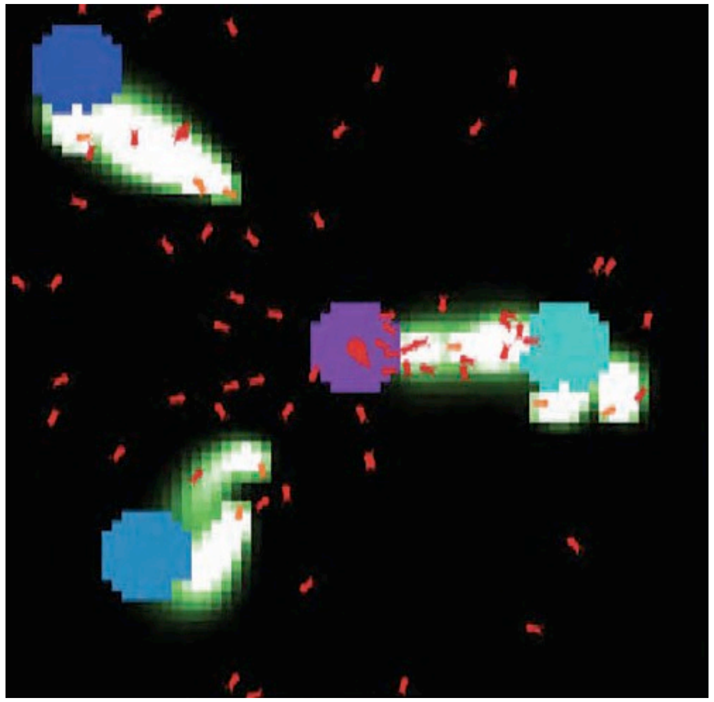


GIS: Geographic Information System

Introduction to Agent-based Modeling and Simulation

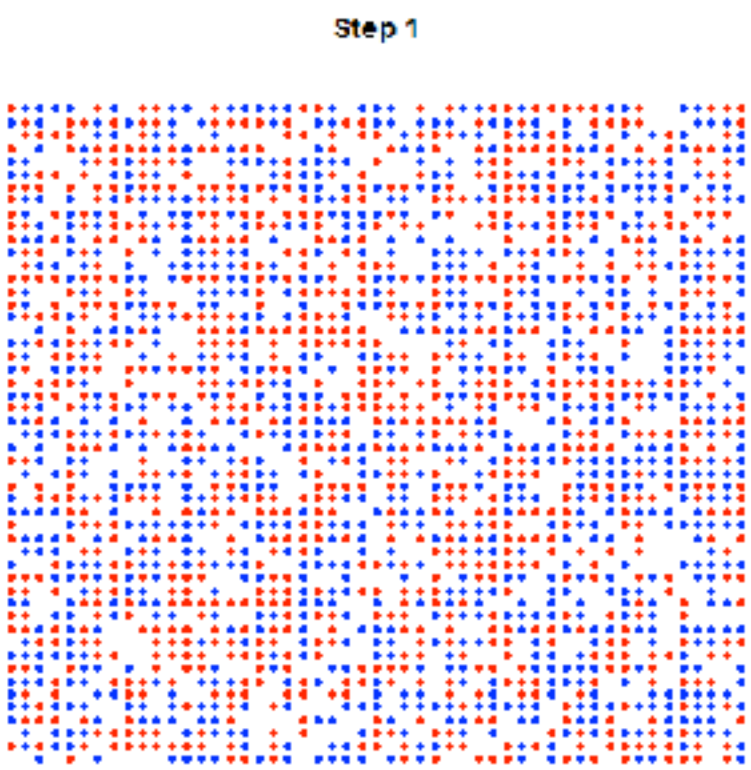
Charles M. Macal and Michael J. North

Environments

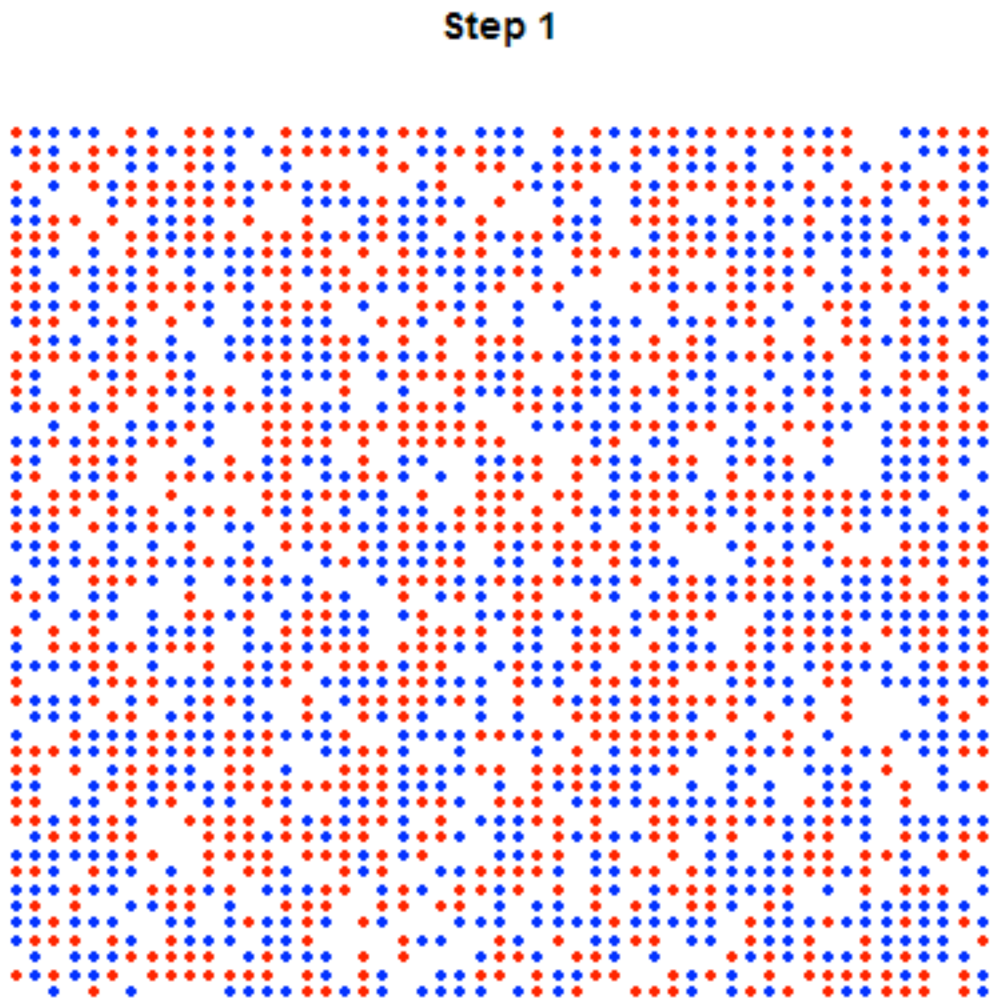







Example: Schelling model

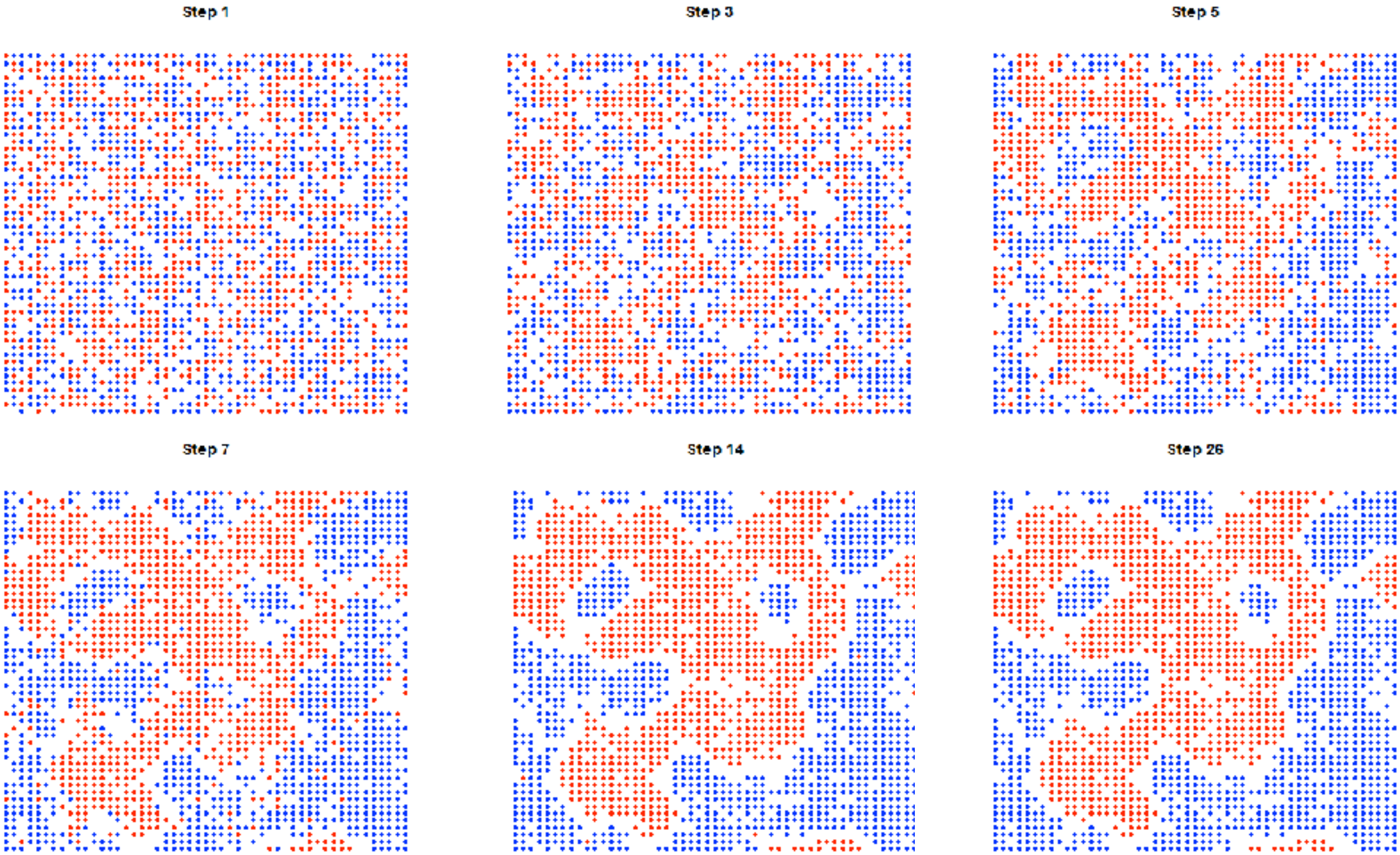
- Thomas Schelling developed to understand neighborhood segregation
- Basic rules
  - Two kinds of agents (red/blue), initially placed randomly
  - Each wants at least X% of neighbors to be the same type as them
  - If not, move to a new location



Example: Schelling Model



Example: Schelling Model



Example: Schelling Model

- What processes need to happen?
  - Each individual needs to evaluate its neighborhood, decide if happy
  - Each individual needs to move if unhappy

Example: Schelling Model

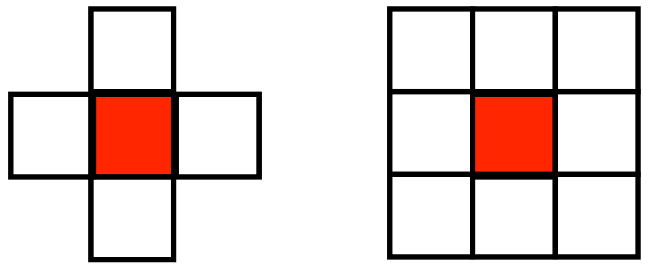
- **Agents**: people
  - **Properties**
    - Red or Blue
    - Fraction of neighbors they want to be same
    - Location
    - Happiness Level

Example: Schelling Model

- **Actions**: move or stay
- **Rules**
  - How to decide to move/stay?
  - What order to evaluate agent movements?
- **Time**: usually discrete steps
  - Continuous time?
- **Environment**: grid (size? boundaries?)

Example: Schelling Model

- What order to do rules in?
- Where to move them if unhappy?
- Count diagonal neighbors? What about grid edges?



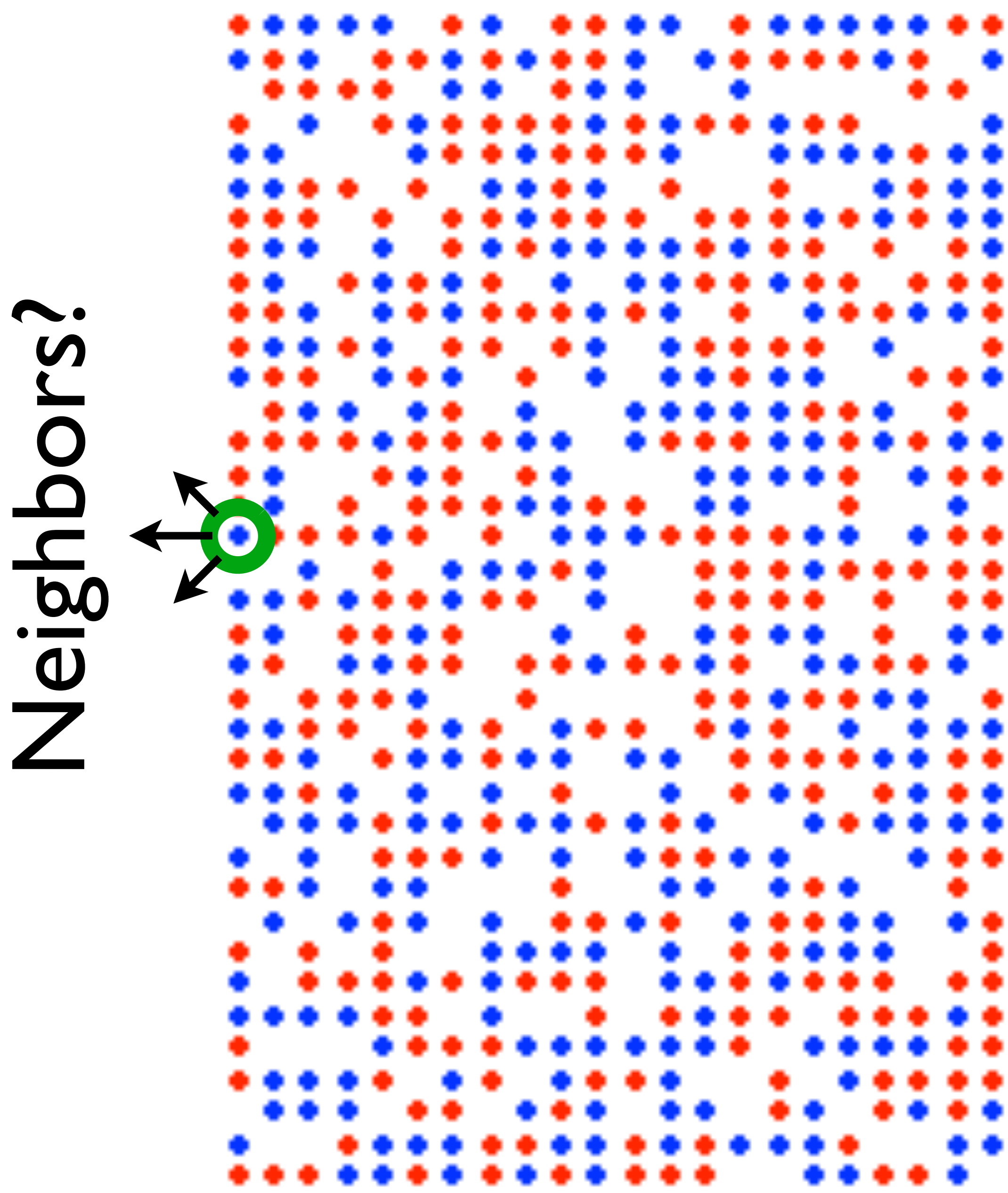
Example: Schelling Model

- For each individual (choose randomly)
  - 1) Evaluate happiness
  - 2) Relocate if needed

- 1) For each individual evaluate happiness
- 2) Randomly relocate individuals as needed

Example: Schelling Model

- Environment: gridded
  - Size?
  - Borders? What to do about neighbors on edges - should it wrap or..?



Example: Schelling Model

- These choices can affect the outcome!
- Need to think carefully about assumptions and potentially do a lot of sensitivity analyses to determine if the observed behavior is robust to these different decisions about how to build the model
- We will explore this more on Thursday!

Common Issues with ABMs (and models in general)

- May seem easy to set up
- Allows for a lot of details to be added
- Frequently used by beginning modelers for these reasons
- But can be very subtle!
- Details of implementation can completely change behavior

- "You should assume that, no matter how carefully you have designed and built your simulation, it will contain bugs (code that does something different to what you wanted and expected)." (Gilbert 2007)

- "Achieving internal validity is harder than it might seem. The problem is knowing whether an unexpected result is a reflection of a mistake in the programming, or a surprising consequence of the model itself. … Confirming that the model was correctly programmed was substantially more work than programming the model in the first place." (Axelrod 1997)

Common Issues with ABMs (and models in general)

- Rule order - SIR model
    - Recover vs transmit first?
- Grid method (e.g. climate models)
- Floating point arithmetic
- Boundary issues (wrap boundary, closed edge, or..?)

Common Issues with ABMs (and models in general)

- ‘Emergent’ behavior can be a strength & weakness
  - Interaction of simple rules can lead to complex, organized behavior
  - But this extends to often arbitrary rules of how you set the model up, e.g. how you make the grid for the model, or order in which you implement rules

Common Issues with ABMs (and models in general)

- Importance of documenting, verifying, and validating code
- Replicability of ABMs and models in general is key to testing these issues—allows authors and others to change assumptions, underlying setup, etc.
- ODD (Overview, Design concepts, Details) Protocol - https://www.comses.net/resources/standards/

Parameters & Sensitivity Analysis

- Sensitivity analysis essential
- Parameter values are almost certainly wrong
- Sensitivity of behavior to parameters can mean predictions or results are less certain
- Simple approach - often know a reasonable range for parameters

For next time…

- Read
  - “Why Model?”, Epstein 2008
  - Sayama, Chp. 1-2
  - “More is Different,” Anderson 1987
  - Wilensky, Chp. 0-1
  - **PARTE framework reading by Ross Hammond (on website)**
- Do
  - Download and Install NetLogo and Python (e.g. Anaconda)