

# Models of Social Dynamics

## An Introductory Module

Paul E. Smaldino

University of California, Merced

### Unit 1:

# Introduction



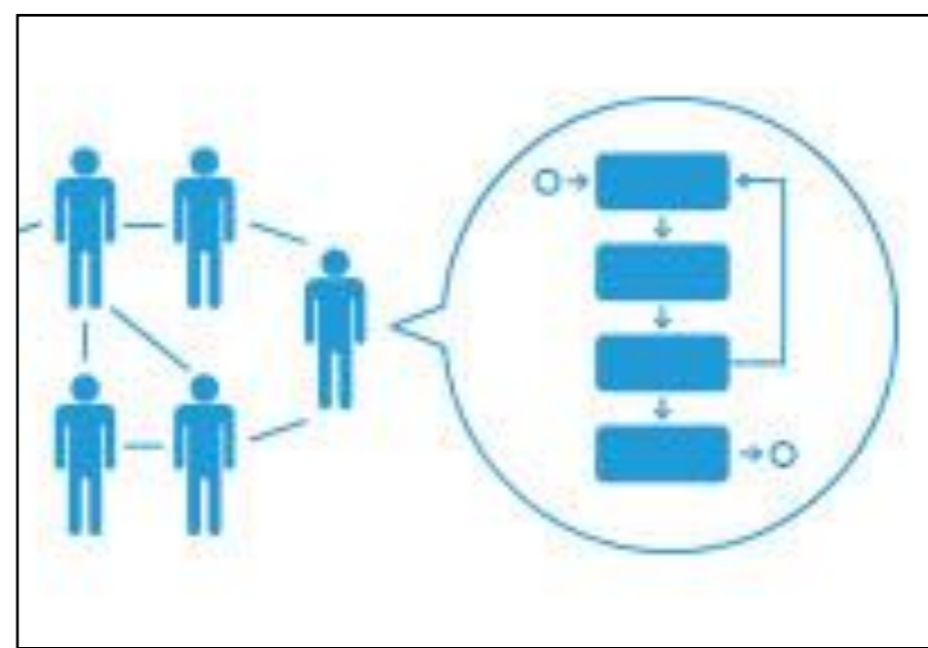


# Topics covered

- Contagion
- Opinions and polarization
- Cooperation
- Coordination and norms
- Sociopolitical cycles

# Models

Models are structures (abstract or physical) that can potentially represent real-world phenomena.



# Formal models

- A simplified version of a system with a specification of parts and relationships between them
- A logical engine for turning assumptions into conclusions

The image displays several mathematical formulas related to probability theory and statistics, specifically focusing on the properties of the likelihood function and the Fisher information matrix. The formulas are written in a stylized, handwritten-like font on a dark background.

Key formulas visible include:

- The derivative of the log-likelihood function with respect to a parameter  $\theta$ :  $\frac{\partial}{\partial \theta} \ln f_{a, \sigma^2}(\xi_1) = \frac{(\xi_1 - a)}{\sigma^2} f_{a, \sigma^2}(\xi_1)$ .
- The expectation of the score function:  $\int_{\mathcal{R}_n} T(x) \cdot \frac{\partial}{\partial \theta} f(x, \theta) dx = M \left( T(\xi) \cdot \frac{\partial}{\partial \theta} \ln L(\xi, \theta) \right)$ .
- The Fisher information matrix:  $\int_{\mathcal{R}_n} T(x) \cdot \left( \frac{\partial}{\partial \theta} \ln L(x, \theta) \right) \cdot f(x, \theta) dx = \int_{\mathcal{R}_n} T(x) \cdot \left( \frac{\frac{\partial}{\partial \theta} f(x, \theta)}{f(x, \theta)} \right) \cdot f(x, \theta) dx$ .
- The Cramér-Rao lower bound:  $\frac{\partial}{\partial \theta} \ln T(\xi) = \frac{\partial}{\partial \theta} \int_{\mathcal{R}_n} T(x) f(x, \theta) dx = \int_{\mathcal{R}_n} \frac{\partial}{\partial \theta} T(x) f(x, \theta) dx$ .

# What can models do for us?

Prediction



Precision



Tractability

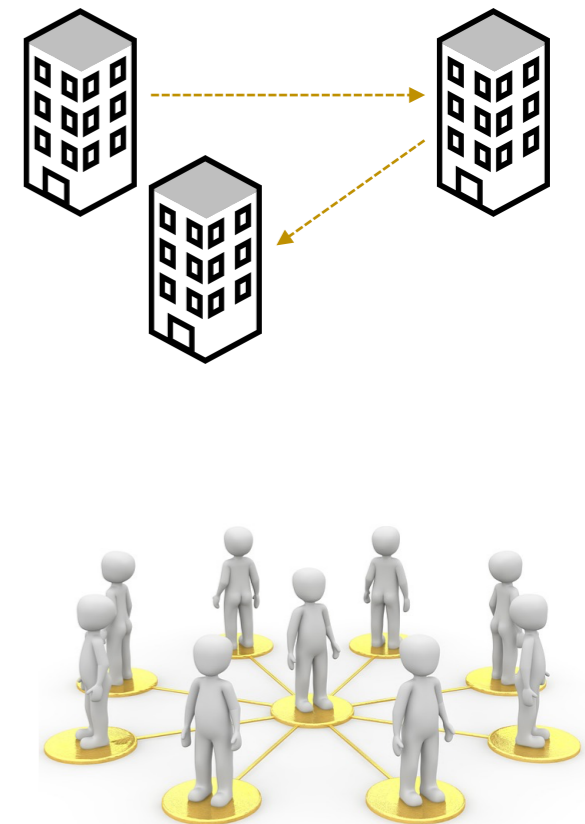


Mental models

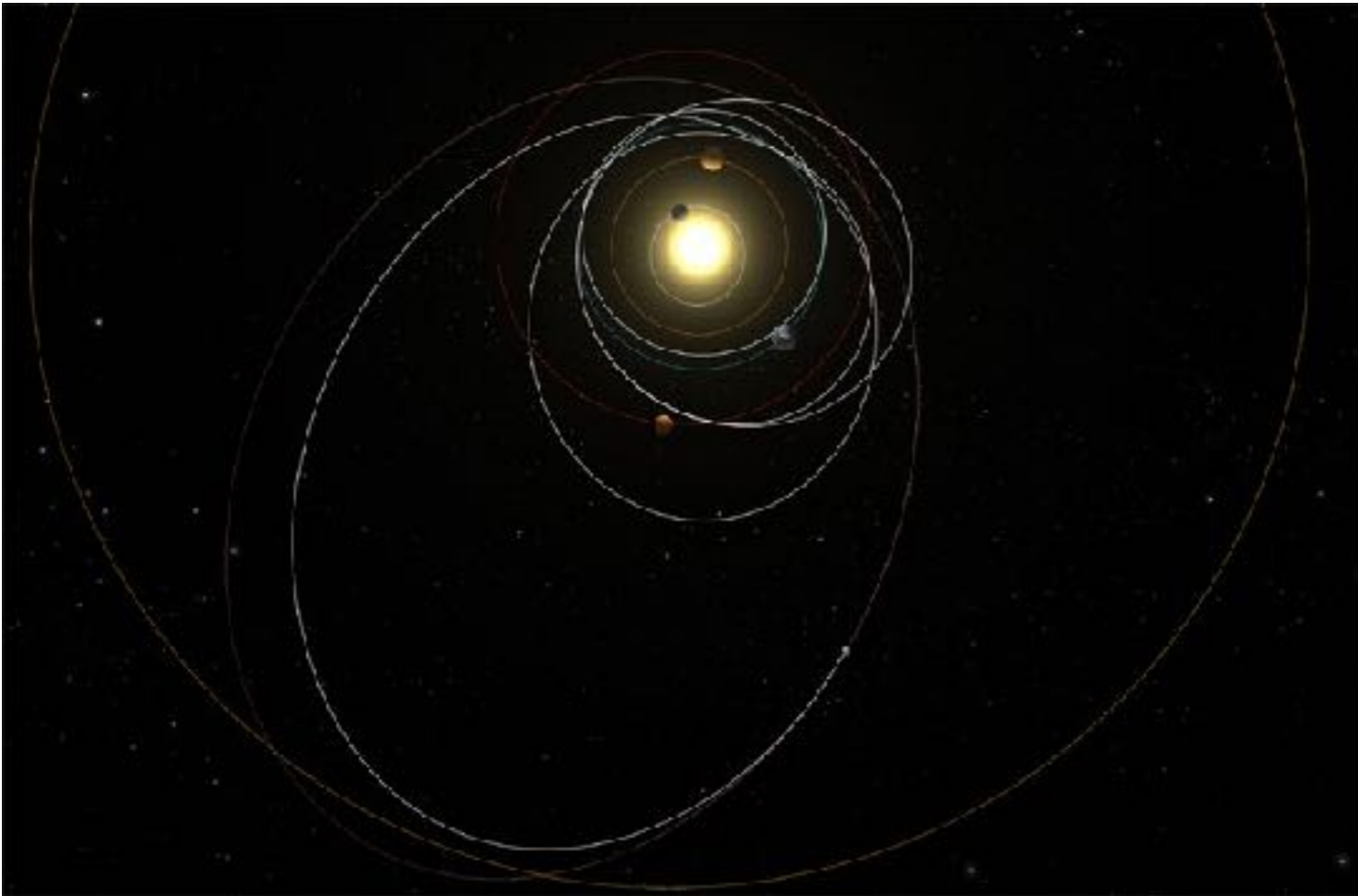


# Hypothesis formation and the articulation of parts

- Science: We want to explain some behavior of some system
- A system can be decomposed into **parts** and **interactions** between those parts
- No single best decomposition for a system. Depends on your question.



# Newton's Model of Gravity



Brahe



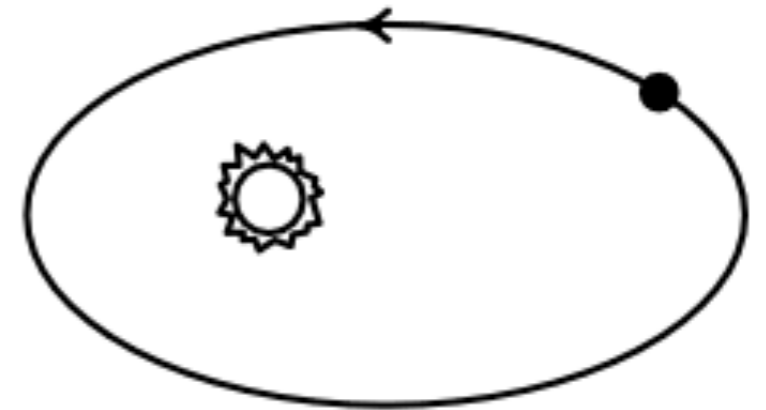
Kepler



Newton

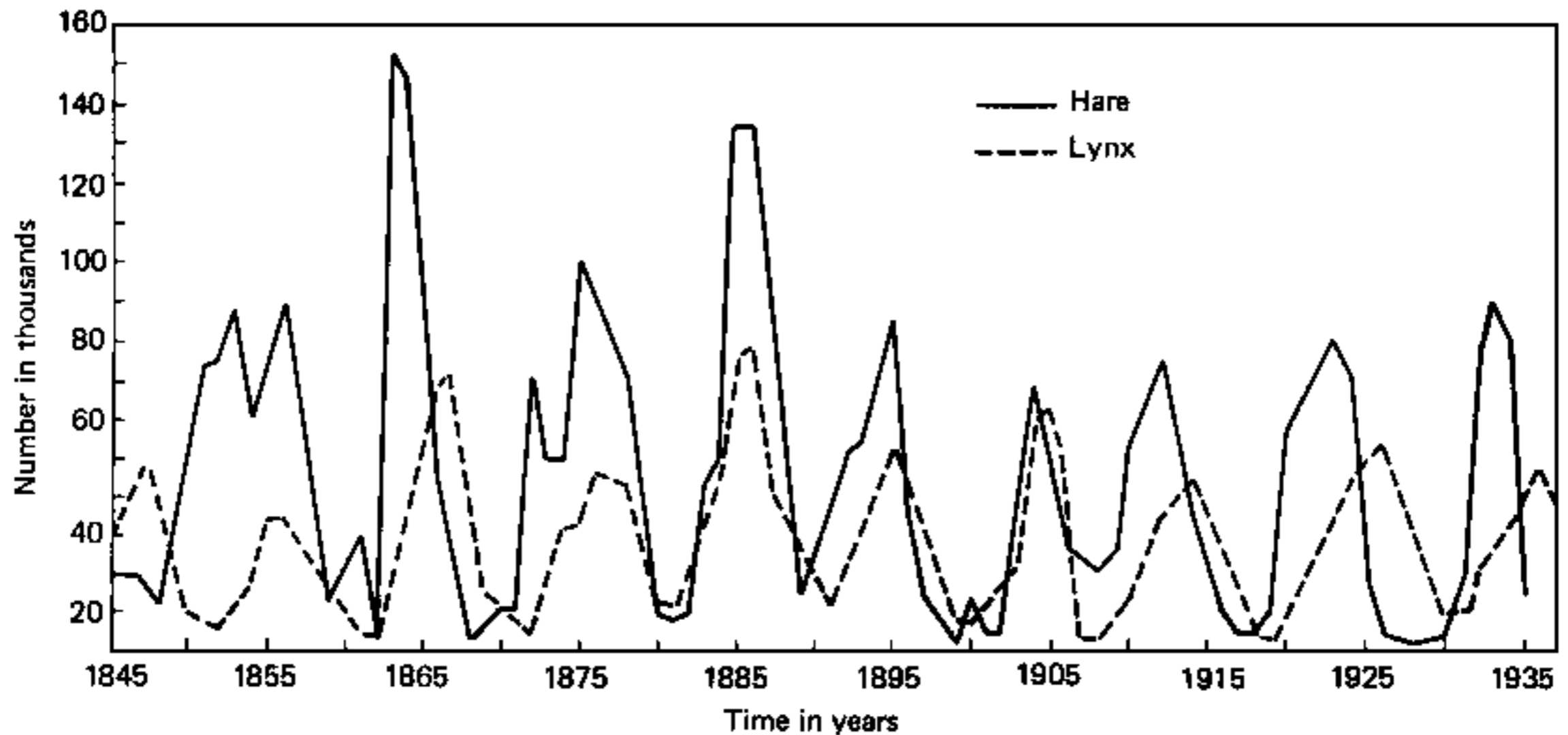
# Newton's Model of Gravity

- Assumptions: Inverse square law + Newton's 2nd law of motion
- Others had proposed inverse square law (Hooke, Wren, Halley). Newton was first to test consequences of this rule.
- Showed that the orbit of a particle acted upon by a force would be elliptical (1687)

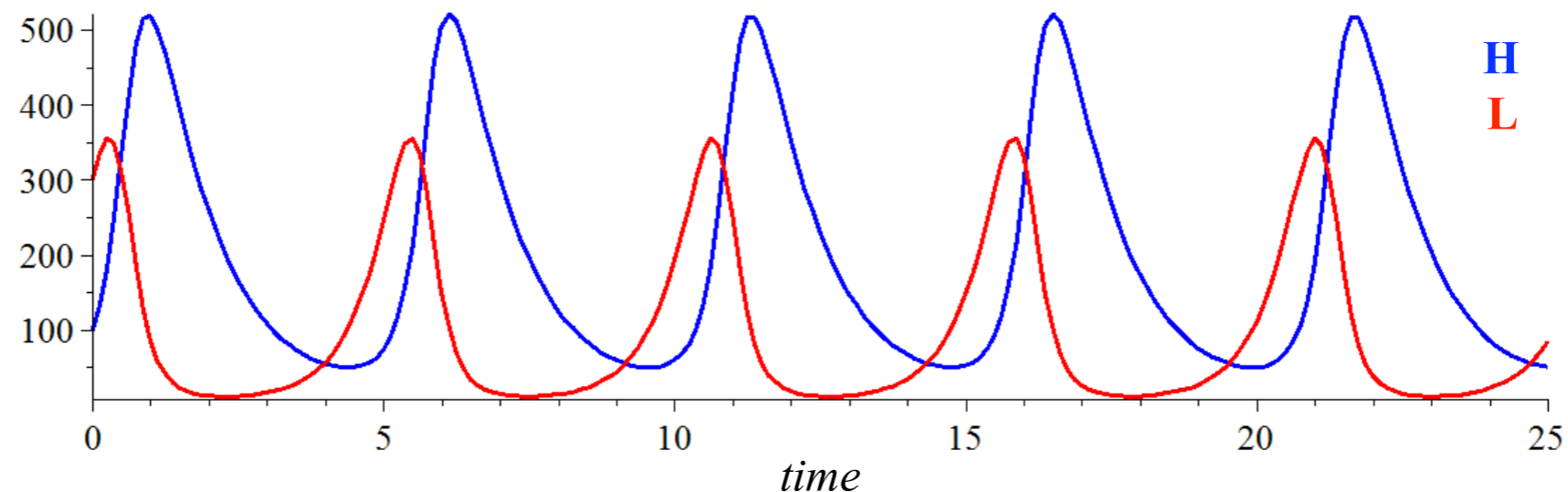
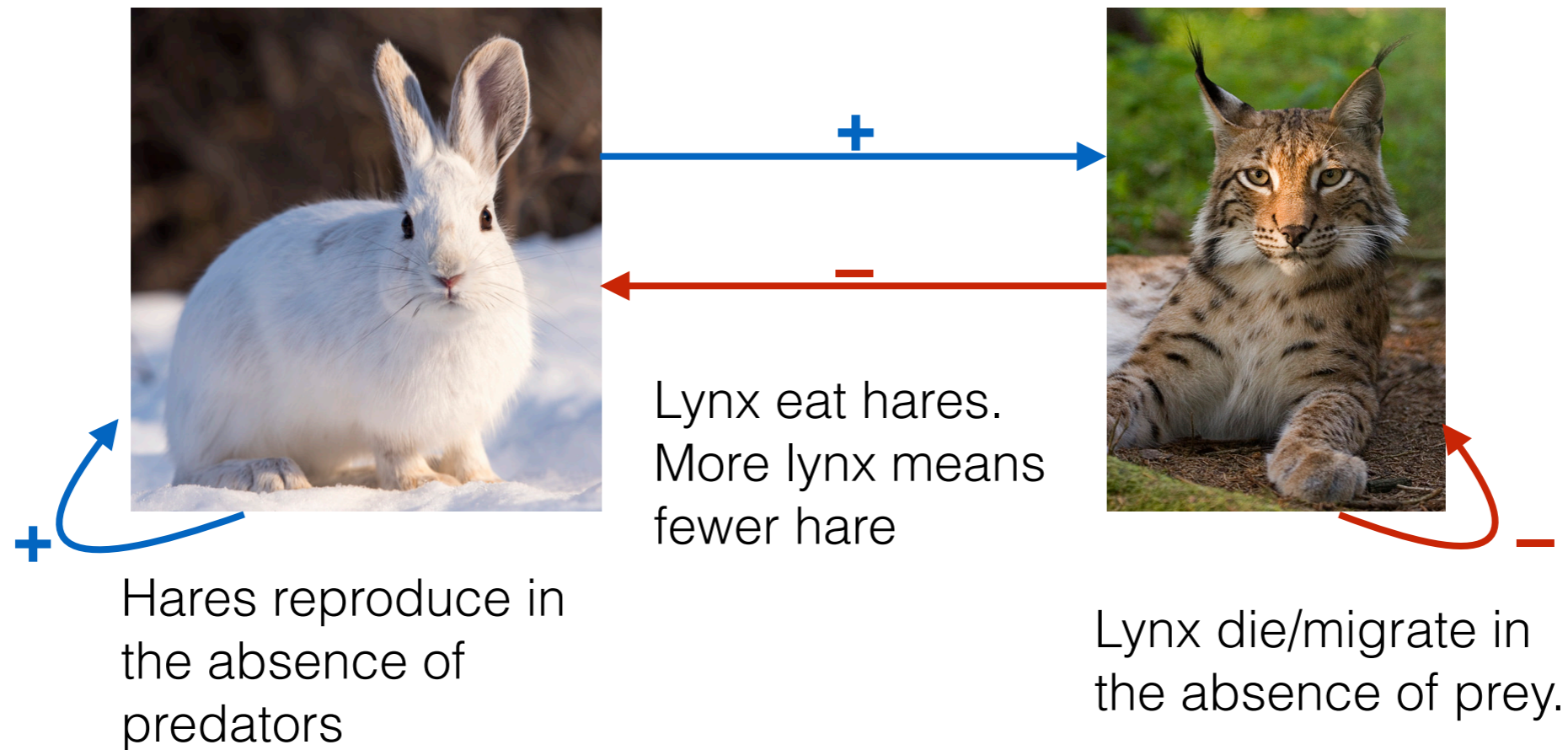


$$\vec{F} = G \frac{m_1 m_2}{|r_{12}|^2} \hat{r}$$

# Lotka-Volterra model of predator-prey interactions



# Lotka-Volterra model of predator-prey interactions

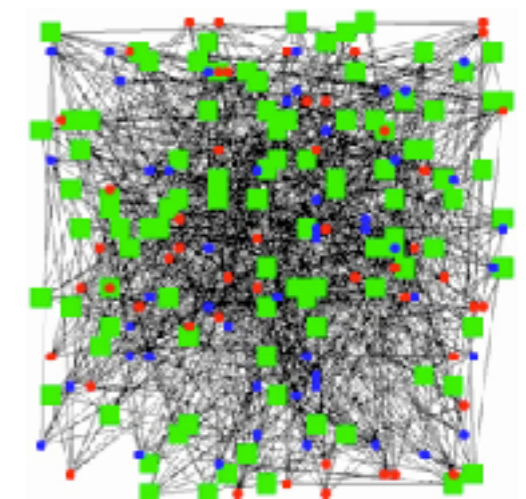
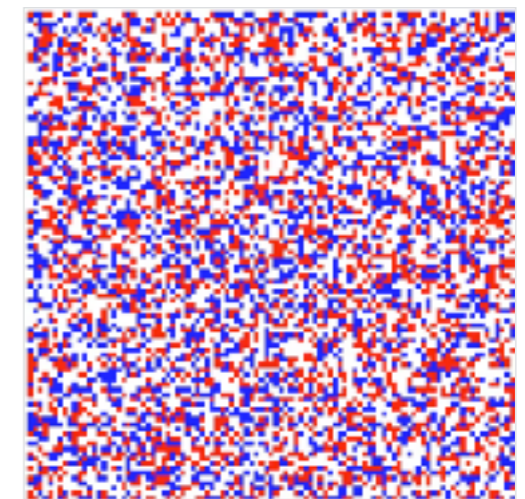
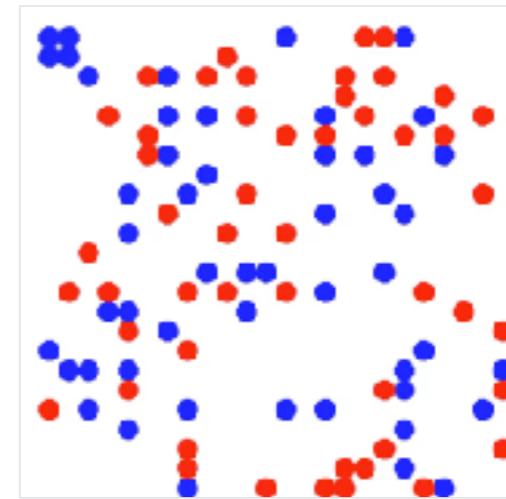


# Disadvantages of mathematical modeling

- **Limited realism.** Models assume a lot of homogeneity to make them tractable. Most social populations have important sources of heterogeneity.
- **Limited intuition.** For those whose deep math training, equations often provide limited insight.
- **High bar to entry.** Lack of mathematical training can prevent access to valuable models.

# Agent-based models

- A type of **formal model** in which individuals (agents) are simulated as explicit computational entities
- Costs: Analytical tractability, easy parameter exploration, a certain kind of elegance
- Benefits: Can account for greater complexity, heterogeneity, and structure (such as spatial or network structure). Can help us to understand emergent phenomena. Lower bar to entry.
- Tradeoff relative to research questions being asked

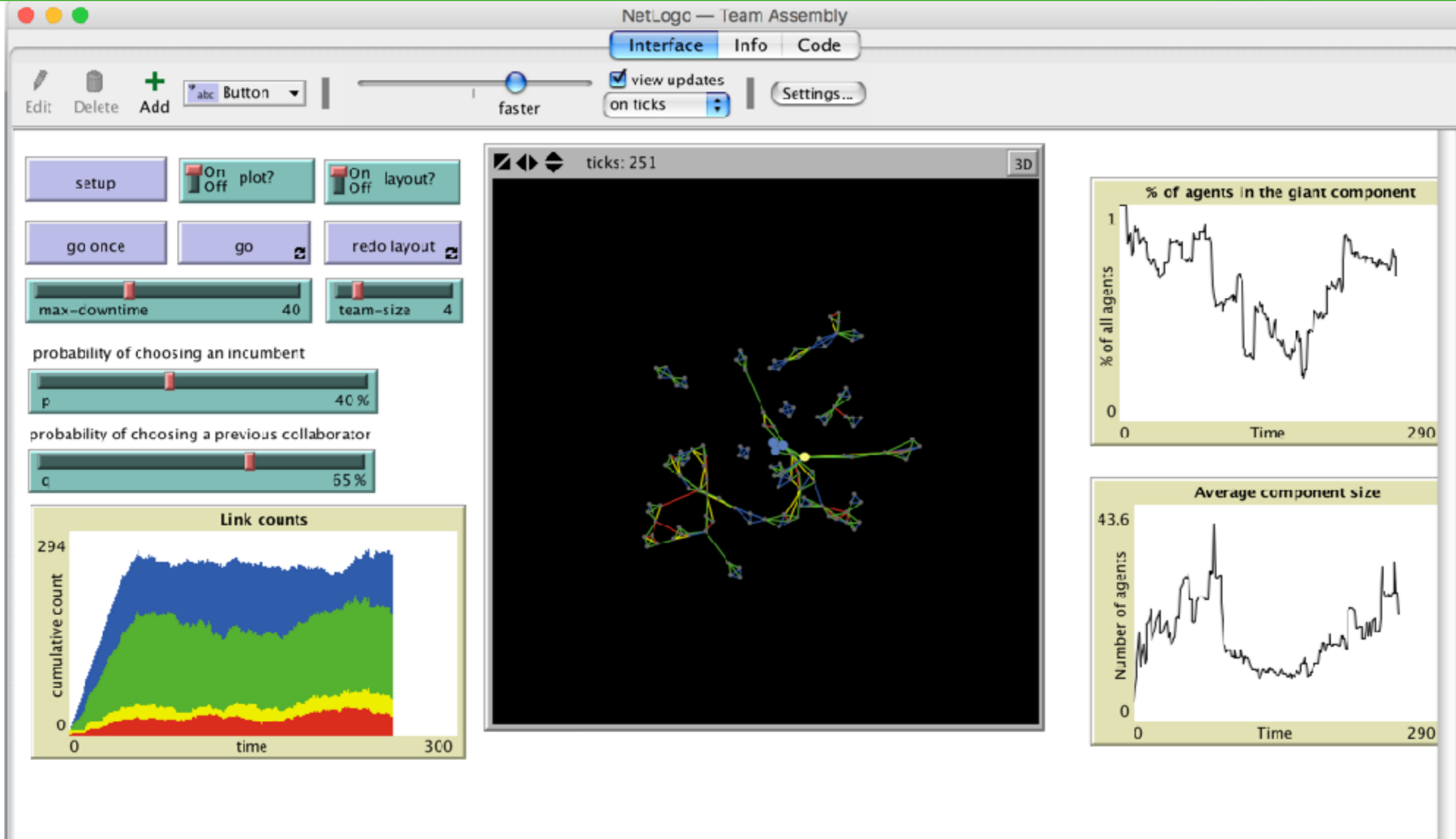


Both mathematical and agent-based  
models are valuable!

# How to code an agent-based model

```
each: function(e, t, n) {
  var r, i = 0,
      o = e.length,
      a = M(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r === !1) break;
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
      for (; o > i; i++)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    } else
      for (i in e)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    return e;
  },
  trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e)
  } : function(e) {
    return null == e ? "" : (e + "").replace(C, "")
  },
  makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n
  },
  isArray: function(e, t, n) {
    var r;
    if (t) {
      if (n) return n.call(t, e, n);
      for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n) : n : 0; r > n; n++)
        if (n in t && t[n] === e) return n;
    }
  }
```

# NetLogo

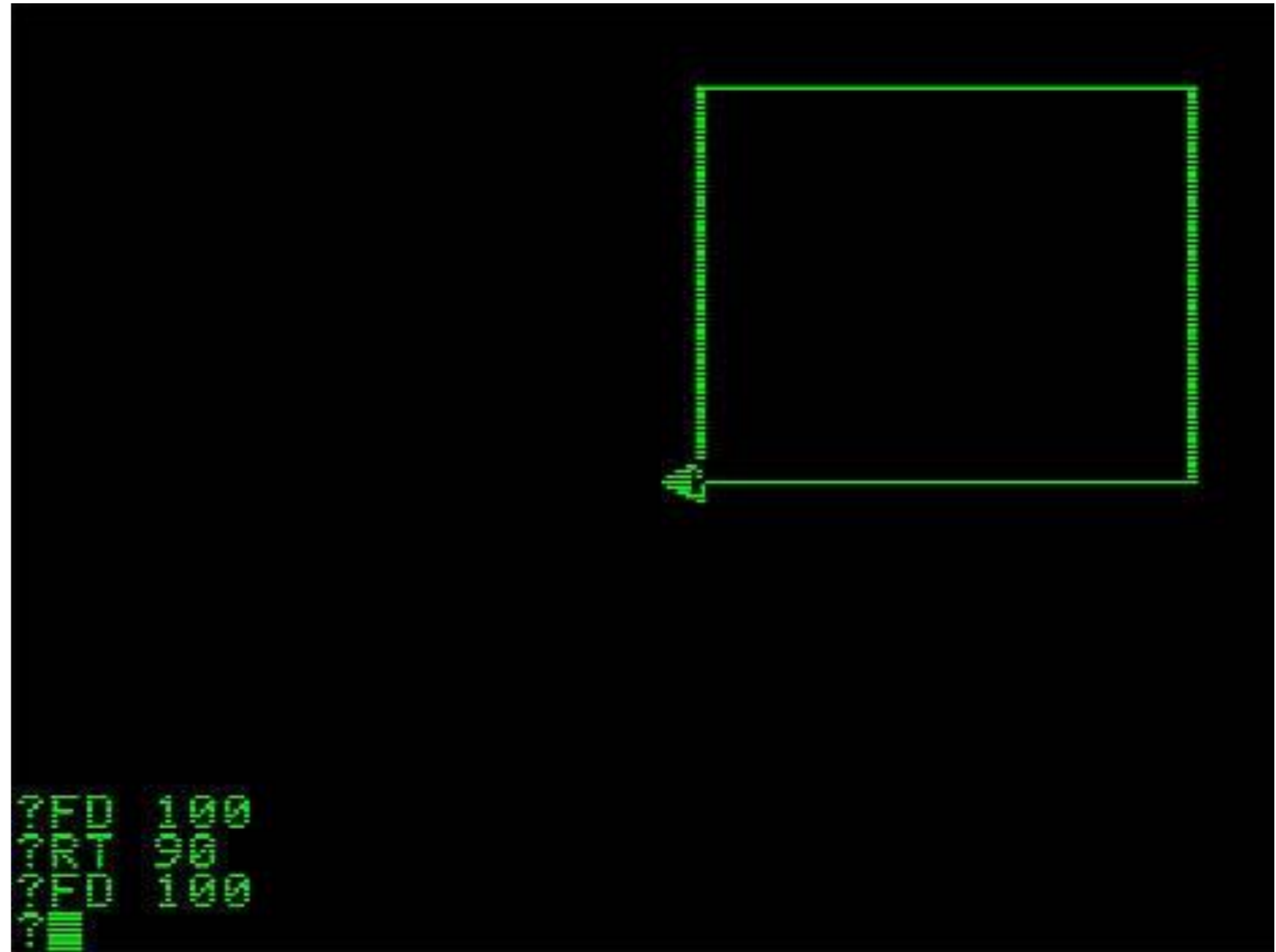


<https://ccl.northwestern.edu/netlogo/>

# Origins



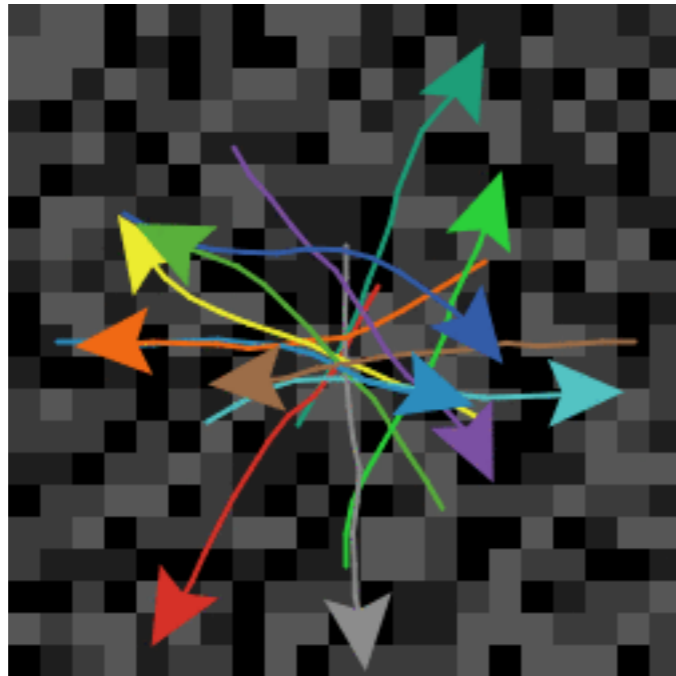
Grey Walter and his tortoise, 1953



Logo: Feurzeig, Papert, & Solomon (1967)

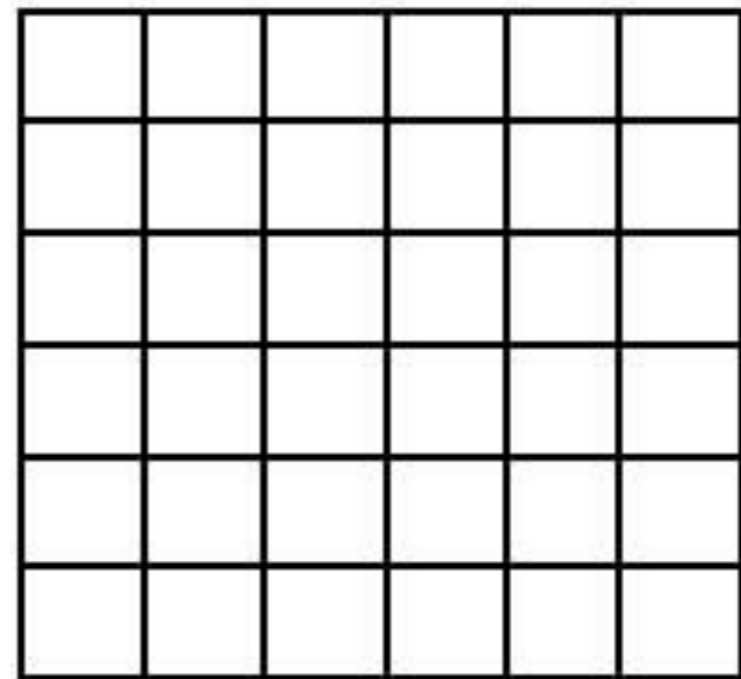
# NetLogo Components

## Turtles



- mobile
- can die and reproduce
- can be networked
- can occupy patches

## Patches



- stationary
- one per location

# NetLogo Components

## ASKING

```
to move
  ask turtles [ ;;move
    left random turning-angle
    right random turning-angle
    fd speed
  ]
end
```

## BEHAVIORSPACE

The screenshot shows the 'Experiment' window in NetLogo's BehaviorSpace. The 'Experiment name' is 'experiment'. Under 'Vary variables as follows (note brackets and quotation marks):', there is a list of variables: ["run-turtles" 100 500 1000], ["transmissibility" 0.3 0.6], ["recovery-rate" 0], ["init-infected" 5], and ["speed" 0.5 1]. Below this, there is a section for 'Repetitions' set to 10, with a checkbox for 'Run combinations in sequential order' which is checked. The 'Measure runs using these reporters:' section contains 'count turtles'. At the bottom, there are fields for 'Setup commands' (containing 'setup') and 'Go commands' (containing 'go'). There are also checkboxes for 'Stop condition' and 'Final commands', and a 'Time limit' field set to 0.

Experiment name: experiment

Vary variables as follows (note brackets and quotation marks):

- ["run-turtles" 100 500 1000]
- ["transmissibility" 0.3 0.6]
- ["recovery-rate" 0]
- ["init-infected" 5]
- ["speed" 0.5 1]

Enter list values in a list, for example: ["my slider" 1 2 7 8] or specify start, increment, and end, for example: ["my slider" 10 1 10] (note additional brackets to go from 0.1 and end to 10.0). You may also vary min, pcolor, min pcolor, max pcolor, min pcolor, random seed.

Repetitions: 10  
run each combination this many times

☒ Run combinations in sequential order  
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' var values will be sequential order: 1, 1, 2, 2, 3, 3 alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

- count turtles

one reporter per line; you may not split a reporter across multiple lines

☐ Measure runs at every step  
if unchecked, runs are measured only when they are over

Setup commands: setup

Go commands: go

☐ Stop condition: the run stops if this reporter becomes true

☐ Final commands: run at the end of each run

Time limit: 0  
stop after this many steps (0 = no limit)

Cancel OK

a simple simulation

**CODE:** movement.nlogo

# Outline of the course

- Unit 1: Introduction
- Unit 2: Contagion
- Unit 3: Opinions and polarization
- Unit 4: Cooperation
- Unit 5: Coordination and norms
- Unit 6: Sociopolitical cycles
- Unit 7: Coda

The 3 Cs:  
Communication  
Cooperation  
Coordination