Complex Systems 530: Computer Modeling of Complex Systems

Lecture 1: Introduction and overview - 1/9/20

Welcome!

- Today
  - Class logistics & introductions
  - Why model? How to choose what kind of model to use?
  - Agent-based models (ABMs)
  - ABM exploration

Getting to know you

- Math background?
- Coding background?
- Modeling background?
- Previous complex systems courses?

Getting to know you

- Name
- Department, stage/year
(e.g. Magical studies, master’s student, 1st year)
- Math & coding background/level of comfort
- Why you’re interested in this course
- Anything else you’d like me to know (optional)

Course Info

- Instructor: Prof. Marisa Eisenberg
- Office Hours - Tuesdays 2-4 pm or by appointment
- Course website & syllabus
  https://epimath.github.io/cscs-530-materials/
  (Go through this together & talk about the plan for the class)
- Course Structure
  - Homework/Labs & Project
- Course Goals & Philosophy

Course overview

- Focus of course: how to design, build, simulate, visualize, analyze, document, and compare agent-based models

- Along the way, we will also explore:
  - A bit of philosophy of modeling/complex systems
  - Intro/intermediate programming
  - Some probability, statistics
  - Cellular automata
  - Some network (graph) theory & data analysis
  - A bit of collective behavior, pattern formation, & game theory
  - Models in social, biological, and physical systems

Course overview

- We will not focus on equation-based models (e.g. ODE’s, PDE’s)—these are also super useful! But there are a lot of existing courses covering these topics (e.g. CSCS 511)
  - (That said, if math is your jam, we will have some of that too!)
  - If you are curious whether your question is better addressed with ABMs vs. other kinds of models, we will discuss some & can talk more as we go

Goals for the course

- Learn about models and model-building—how and why to build models for a given problem
- Become a sophisticated consumer of articles involving ABM and gain a solid grounding in some of the key models in the field
- Develop familiarity and competencies in two programming languages, NetLogo and Python
- Leave this class with an ABM you have designed, built, and analyzed that will (hopefully) be useful to you in your future research!

Languages for Coding

- Example code and HW files will be mostly in NetLogo & Python (3?)
  - See website for more information on how to install
  - Anaconda
- However, for much of the class you can code in whatever language is comfy for you (e.g. R, MATLAB) (within reason…)

Prior Background

- A background in complex systems is not required, but hopefully you have a strong interest!
- Some basic programming and basic math/stats background is **strongly** encouraged.

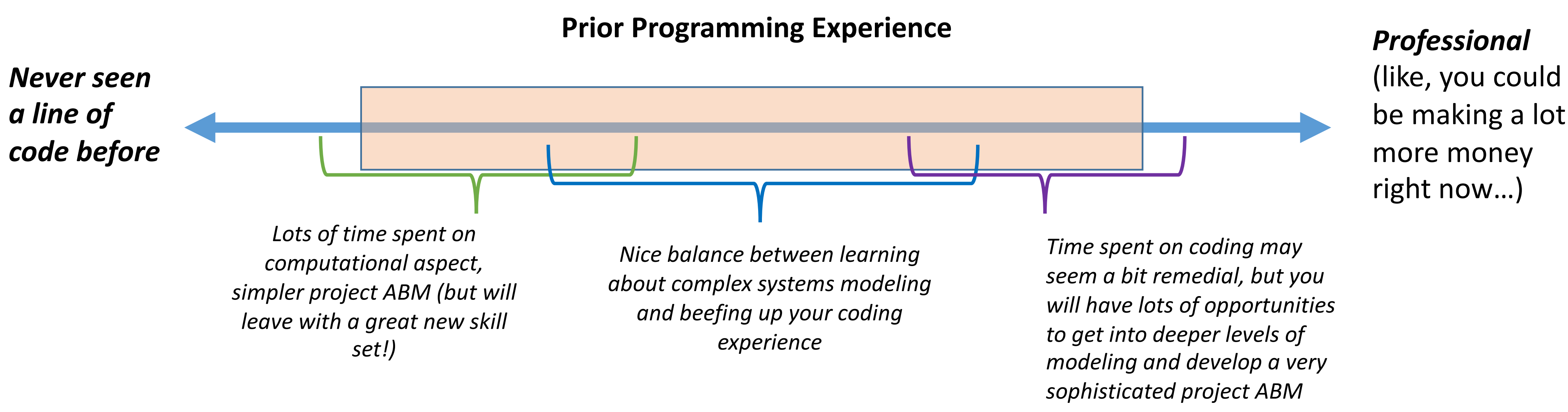


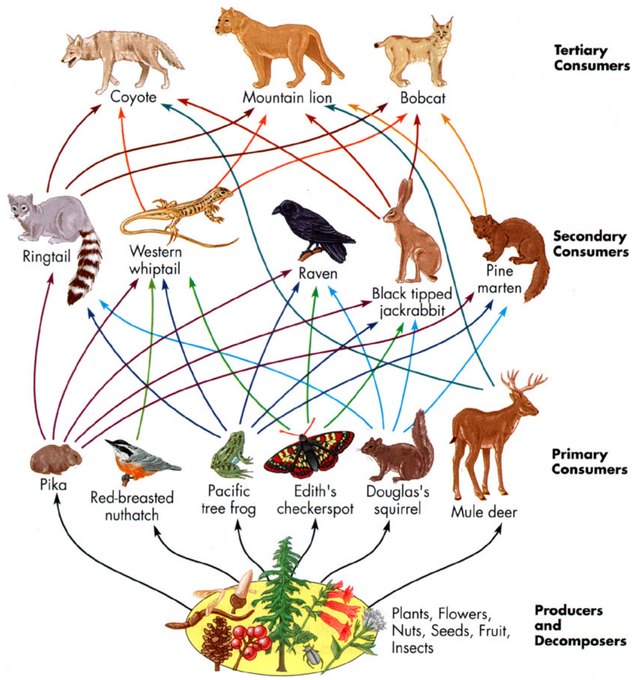
diagram borrowed from Lynette Shaw!

Laptops

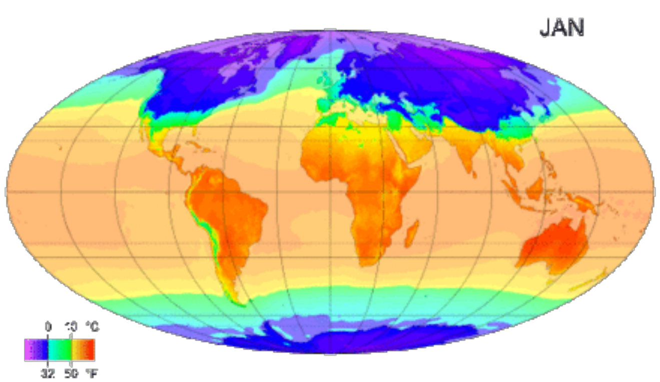
- **A laptop is required for this class**
- If you don’t have access to a personal laptop, we can make arrangements through LSA to get you one for the semester

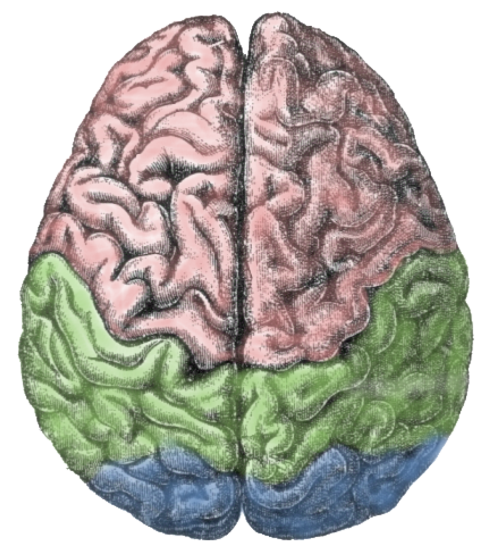
Introduction

What is a complex system?



© Daniel Biber, Germany, Shortlist, Professional, Natural World & Wildlife, 2018 Sony World Photography Awards



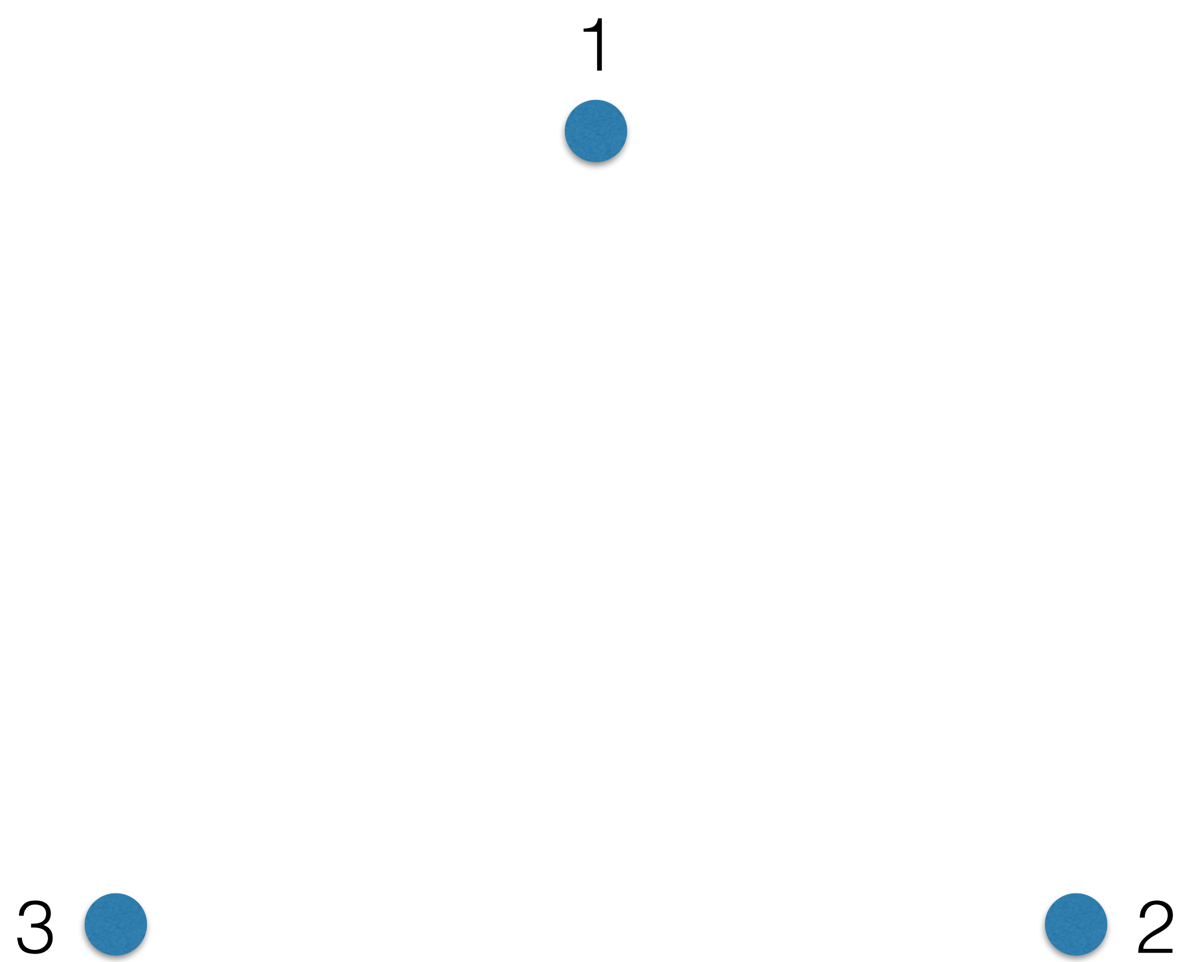


What is a complex system?

- **A collection of components that interact, often in nonlinear ways**

- Sayama: “Complex systems are networks made of a number of components that interact with each other, typically in a nonlinear fashion. Complex systems may arise and evolve through self-organization, such that they are neither completely regular nor completely random, permitting the development of **emergent behavior at macroscopic scales**.”

Interactive Example!



Interactive example

- 1) Mark a point at random within the triangle (can really be anywhere but this is simpler)

- 2) Randomly pick a vertex

- 3) Mark your next point: the point halfway between your current point and the vertex you chose

- Repeat steps 2) and 3)

Interactive example

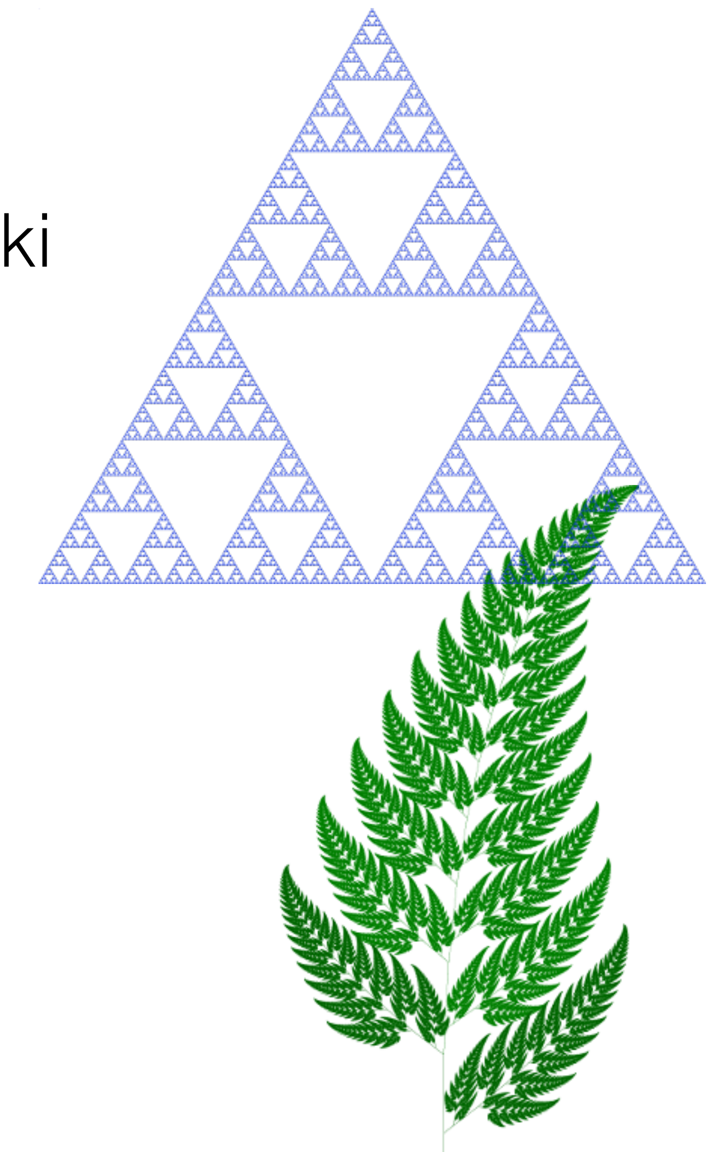
- What do you think will happen?
- **Try it out!** Everyone plot 10-15 points on your transparency starting at a random point.
  - Use the ruler to measure carefully!
- Once you’re done, we’ll combine results

What did we find?

- Discuss
- Might be tough to see, let’s code this process and see what happens with a lot more points!
- (Run code here)

The chaos game!

- Generates a fractal! This is called the **Sierpiński gasket** or Sierpiński triangle
- Fractals are objects that exhibit self-similarity (ferns, trees, snowflakes, rivers, & more)
- Simple rules can generate surprising emergent patterns!



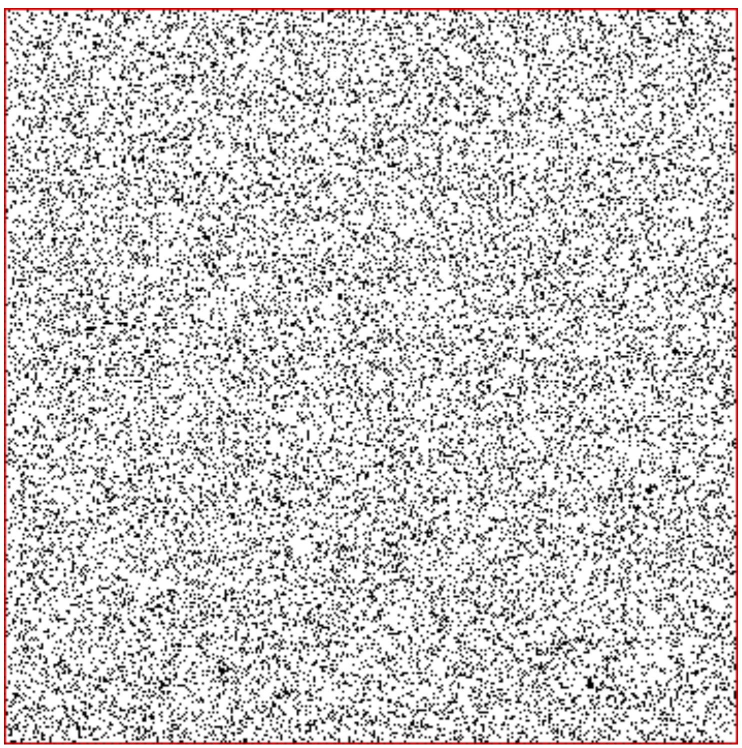
Interactive game is from: http://math.bu.edu/DYSYS/chaos-game/node4.html

The chaos game

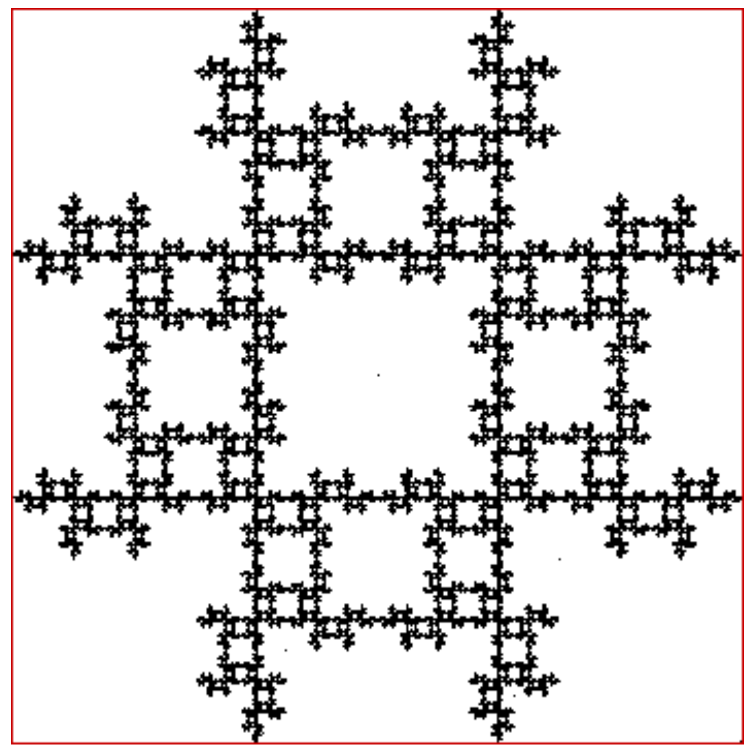
- The Sierpinski gasket is an attractor for this process
- Why does this work? A little intuition—think about transformations—discuss
- We will do more with this later
- Does it work with other shapes?

The chaos game

Try it with a square?



Use a square but don't choose the same vertex twice



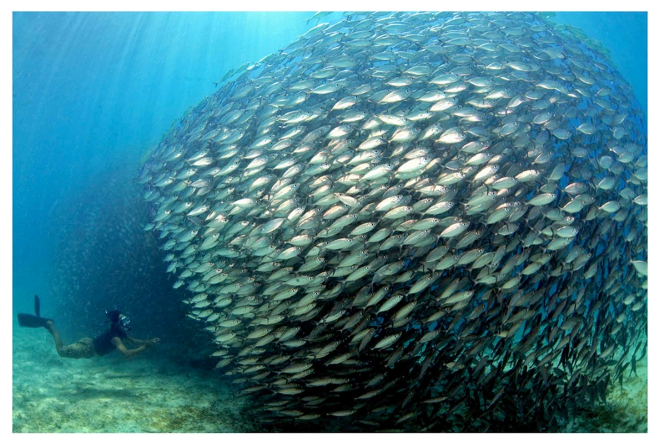
Many other variations!

Complex Systems

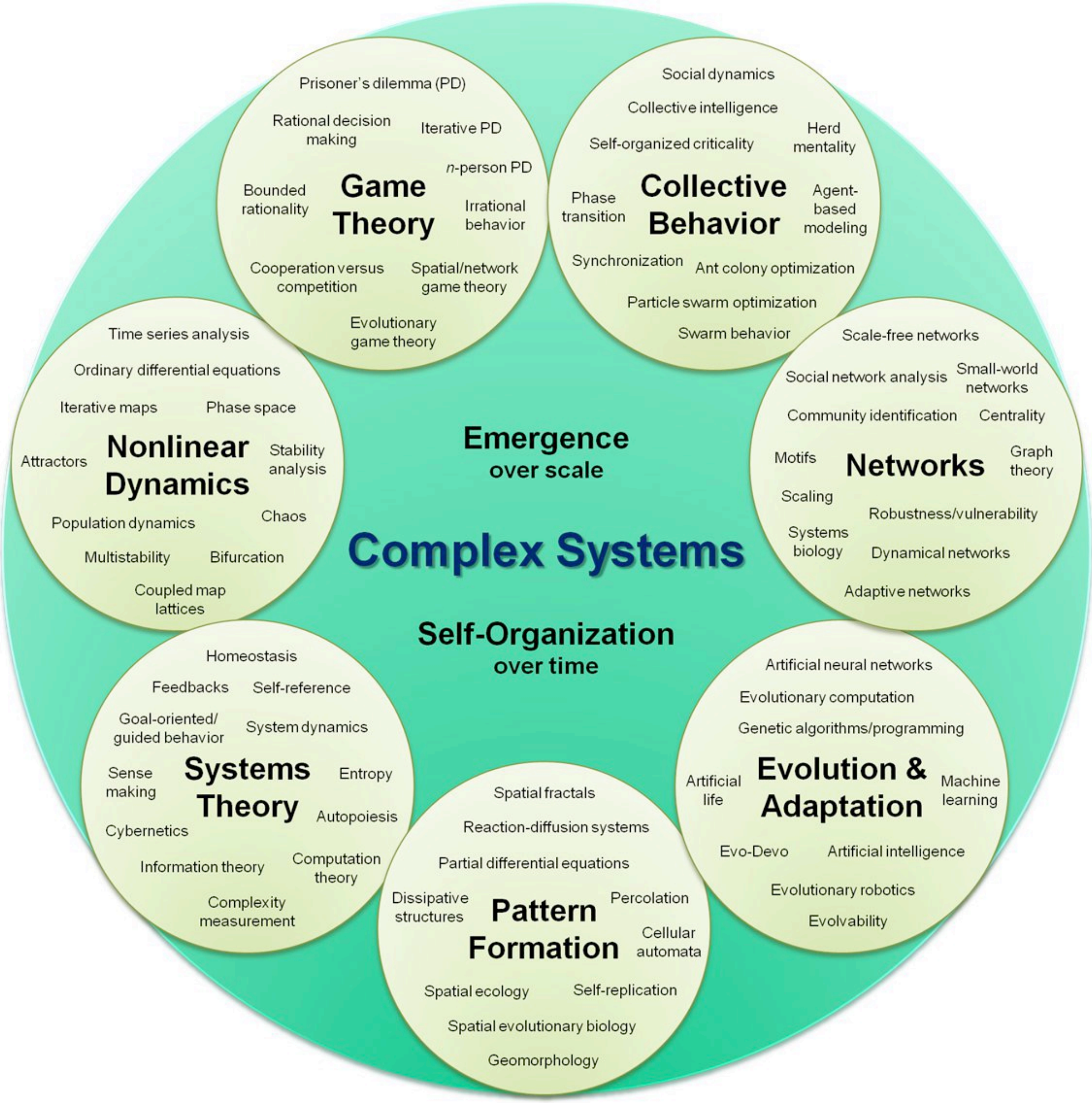
- Emergent behavior
- Self-organization
- Adaptive interactions
- “Fat-Tail” Behavior
- Chaos
- Nonlinearity, tipping points, etc.

Example: flocking!

- Flock of geese/school of fish
    - Forms a large, organized pattern
    - But no ‘group mind’ or leader
    - Birds follow local rules
    - Result is emergent, organized behavior



Complex Systems Science



Introduction to the Modeling and Analysis of Complex Systems by Hiroki Sayama

What is a model?

- In a few words, what does a model mean to you?

What is a model?

Reality

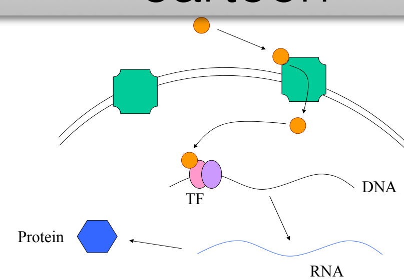
↕

Abstraction of Reality

Physical
e.g. miniature, animal model

Diagram
e.g. schematic, map, cartoon

Math/computational
e.g. equations, code



$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

```
for i in range(1,10):
    print(i)
```

Models are abstractions

- They are **simplifications**—and so they are **wrong**!
- But they are also often **useful**
- Abstraction is both their strength & weakness
- The trick is, how much to simplify? What details to keep, which ones to remove?
- Even “simple” systems can be complex

Why model?

- Discuss

Why model?

- You already model all the time! All of us build mental models, pictures, etc. of the world and the systems we're interested in.

- Really, we are asking: why build explicit models, where we lay out the mechanisms, assumptions, parameters, etc. as equations/code?

    - If we make our models explicit, we can test, interrogate, explore them—can make sure they are self-consistent, explore more complex scenarios, etc.

Why model?

- Understand mechanisms, causality
- Help to guide & test control, treatment, intervention strategies,
- Test alternative scenarios (counterfactuals)
- Forecast/predict

Why model?

From Epstein 2008:

- Explain
- Guide data collection
- Illuminate core dynamics
- Suggest dynamical analogies
- Discover new questions
- Promote a scientific habit of mind
- Bound (bracket) outcomes to plausible ranges
- Illuminate core uncertainties
- Offer crisis options in near-real time
- Illuminate core dynamics
- Demonstrate tradeoffs / suggest efficiencies
- Challenge the robustness of prevailing theories
- Expose prevailing wisdom as incompatible with data
- Train practitioners
- Discipline the policy dialogue
- Educate the general public
- Reveal the apparently simple (complex) to be complex (simple)

Why model?

Known knowns	Known unknowns
Unknown knowns	Unknown unknowns

Why model?

Known knowns	Known unknowns
Unknown knowns	Unknown unknowns

Words of caution

- How to know if you have the right mechanism?

- How much do our assumptions and simplifications affect our outcome?

- May not be possible to, for example, predict certain things from the data/understanding that we have! (e.g. early epidemic curve)

- *Importance of understanding/testing assumptions, uncertainty quantification, model comparison, etc.*