

# CSCS 530

## Computer Modeling of Complex Systems

Rick Riolo <rrriolo@umich.edu>

Office hours: after class, by appt (email)

CSCS Computer Lab – 4477 Randall

Start classes in 120 West Hall (LSA Lab)

Mid/Late-January (?) => 321 West Hall

Later in term, discussions in CSCS commons:  
317A West Hall

### **Today:**

- Brief introduction
  - Complex Adaptive Systems
  - Modeling, Agent-Based Modeling
- Course goals
- Administrivia: emails, logins, etc

## **Complex [Adaptive] Systems**

- Molecules (Percolation; DLA, Self-Org)
- Biological (ecologies, brains, IS, cells)
- Social (families, groups, populations)
- Economies, markets, firms, consumers, ...
- Political, cultural systems (cities, states,...)

## **Micro-level characteristics**

- Medium number of “agents”:  $10^1 - 10^7$
- Agents:
  - usually simple; limited mem, capabilities
  - diverse: state (knowledge), capabilities, interactions, goals
  - adaptive (capabilities, goals, relations)
- Interaction patterns are key: who interacts with whom, how often, when, how?
- Interactions are (mostly) local, non-random: spatial / structured (eg, social) topologies
- Mix of cooperative/competitive interactions; Positive/Negative feedbacks emerge.

## Macro-level characteristics of CAS

(*what we want to understand!*)

- Aggregate features and patterns: GNP, market prices, system structure, population size, fraction infected; dynamics *and* effects of parameters
- Hard to predict *detailed* dynamics
- Path dependent phenomena:
  - sensitivity to initial conditions
  - accumulation of historical accidents
  - multiple “equilibria”
- Large scale structure/dynamics emerge: “self organized criticality;” cycles, basins of attraction; bifurcation patterns; nonlinear dynamics, e.g., feedbacks  $\rightarrow$  “tipping” phenomena.
- Perpetual novelty: “transient behavior” is important; systems never reach equilibrium!
- New (emergent) structures: cells, organisms; tribes, firms, nation states.

## How to Model CAS??

Models:

- Description of **part of** the world.  
abstraction, simplification; generalization
- Capture structure/dynamics that are **of interest to modeler.**

Key questions:

- What to include?
- What to leave out?

Answering those: **art and craft of modeling!**

**KISS** (Keep It Simple, Stupid)

- Easier for you to understand
- Easier to analyze, display results
- Easier to describe and explain to others!

Computer models:

great temptation to include too much!

**Another key question:** *What is the purpose of the model?*

- Predictive:

Strongest claims; clear policy implications; most difficult to obtain (CAS–impossible?)

Note role of abstraction in “predictions:“

- There will be cycles (vs equil vs chaos)
- There will be 12-year cycles
- There will be this exact time series...

- Explanatory:

Explain observed patterns, but not make detailed predictions. Example: evolutionary models; anthropological models; ...

- Exploratory:

Develop intuitions. Understand processes. Study alternative hypotheses. Determine implications of assumptions. Examples: Schelling’s segregation model; Axelrod’s Cultural Diversity model; SOC models; ...

## Modeling languages

- Text description (informal)
- Mathematical (formal)  
Equations (static, dynamic); logic stmts.  
Precise, robust (over assumptions).  
Must be “simple” for closed form solutions.  
Numerical methods for analysis/estimation.  
(e.g., the system dynamics approach to “equation-based models” EBM).  
Difficulties from:
  - heterogenous, non-continuous space
  - heterogenous agents
  - nonlinear mechanisms (if-then rules)
  - adaptive agents; creating new agents.
- Computational Models  
-> ABM, IBM, “simulations”  
Formal; precise. (Standard formalisms?)  
Text description of ABM often messy!  
“Solve” by running the program!

## Agent Based Models

- Basic components are “agents” with their own state and capabilities.  
(Contention over “agency” in some fields)
- Dynamics defined in terms of simple, locally driven mechanisms and interactions.
- Observe the complex micro- and macro-behavior (dynamics) which result.

Example: Schelling’s Segregation model.

- Agents: households of two types (B,W), located in 2D discrete space.
- Dynamics:
  - simple decision rule on whether to move (local neighborhood composition)
  - simple movement rule
- We observe: segregation patterns, neighborhood composition, etc

## **EBM: equation based modeling**

- Fundamental entities are observable, often aggregate variables (GNP, #infected)
- Dynamics: Relate variables by eqns (ODE, PDE)
- Often based on mean-field approach (ignore/approximate any underlying details)
- “Top-down” approach (macro econ, epidemiology, ecology, system dynamics, ...)

## **ABM: agent (individual) based modeling**

- “Bottom-up” approach
- Basic entities are “agents”
- Dynamics: behaviors of agents, including interactions with each other and the “world;” adaptive changes, learning.
- Generally not include aggregate variables in model dynamics.
- Observe “emergence” of aggregate level dynamics and structure (GNP cycles, trade patterns, nation alliance formation/dissolution).

## EBM vs ABM vs XXX

- World -> Observations -> “Theory” -> Conceptual Model(s) -> Implementation(s) -> Observations
  
- Distinguish conceptual model from implementations of the model.
  
- Some models are mixtures: some processes driven by equations, some by micro-level mechanisms
  
- Key feature: ABM == Bottom-Up  
Basic dynamics represented at micro level.

## CSCS 530 Course Goals

- Introduce issues/concepts involved in modeling CAS using ABM approach.
- Emphasis on exploratory / explanatory models of CAS
- Modeling from design to publication:
  - what to model (given your goals) – *model at the conceptual level*
  - implementation – *computer program level*
  - verification / validation
  - *experiment* design and management
  - writing “modeling” papers
- Implementation: be able to modify/implement a simple CAS model (e.g., Schelling, Bak, Axelrod, El Farol) in a day or two.
  - Small project(s): Repast-java
    - \* Term project: any language you choose (recommend: Repast-Java)

- The Art and Craft of Modeling:  
One learns this by example and practice.  
Read, think about, *discuss* a variety of papers / models.  
Design, implement, *document* your models, computational experiments and results.
- Computer programming
  - **Learn by doing** – its the only way
  - Try things – beyond the assignments
  - Spend time in lab – ask questions, brainstorm, discuss, help.
  - *Learn to read manuals., search web!*
  - Learn to *document* your *programs, testing and experiments!!*

There is not time to cover:

- How to generate random numbers.
- Survey of algorithms for event scheduling
- Statistical analysis of results
- Repast / Java in great depth
- “Systems dynamics” models.

## **Classwork and grades** (Probably...*It is a lot!!*)

- 10% – computer project #1 (group!)
- 20% – computer project #2
- 20% – computer project #3
- 20% – class discussion–student led!
- 30% – term project: presentation, paper

Communication by email , www

- announcements via email (ctools)
- updated syllabus on line (RSN...)

*[cscs.umich.edu/education/CSCS-courses/cscs530](http://cscs.umich.edu/education/CSCS-courses/cscs530)*

CSCS Computer Lab: 4477 Randall (321 W.Hall)

Weekdays: ~10am - ?;

AskCSCS office to let you in if needed.

Weekends: if someone with key is in–knock!

CSCS students – can get key!

Machines are up, accessible 24x7.

Access via ssh only!

(Windows: Putty, Hummingbird Exceed)

## General plan:

- Wed. 10th: Demo some models. Simple linux, emacs, &etc. Copy/run demo(s).  
***Read “Things to Read” (syllabus Jan 5)***
- Fri 12th – Start fast intro to Java
- Wed 17th– More intro to Java; start Repast?
- Next weeks: Quickly learn some Java/Repast leading up to Projects.
- Read/discuss variety of CAS models

Quick survey: Java? C++? C? Unix user?  
OO Programming? RePast? NetLogo? Other?

## Books &etc:

- Class handouts (examples, overheads)
- Most model papers will be online
- Papers/books in CSCS lab/commons.

Programming books – personal preferences vary...

- Many java books (O’reilly’s, Eckel, ...);  
Lab has some – see bookshelves.  
Books you like? Tell us all–ctools
- RePast documentation online.
- Java docs online (Tutorials, API,Books)

Send me an email **\*By Sunday\*** with:

- Name, email, unickname
- enrolled (or will be)? Department (grad,undergrad)
- Your CAS modeling interests  
Your goals for this course.
- Used u/linux? Java? C? Other langauges?
- Used stat packages, mathematica, matlab?

We will create a CSCS account and send you email about it.

- Change pw as soon as possible.
- Follow pointers to CSCS online doc pages, poke around a bit.
- Go through the “Intro to Unix” short tutorial. Be able to:
  - login/out, in lab, remotely if need be
  - move around directories, files
  - edit files (emacs)
  - other basic unix commands

*Don't be shy:* ask me, Sarah, Mike, others in lab for help!

cscs.admins@umich.edu