Syllabus – CSCI477/577: Simulation and Modelling

Spring 2023

Meeting time and location

Lecture: TR 10:30-12:00, SS 362

Instructor information

Instructor: Doug Brinkerhoff (feel free to use my first name)

Office: Social Science 403

E-mail: doug.brinkerhoff@mso.umt.edu

Office Hours: Office hours will be determined based on availability poll on day 1!

Course Requirements

Formal prerequisites: CSCI152 (Intro CS), M172 (Calculus II)

Very helpful: M221 (Linear algebra), CSCI232 (Data structures), STAT341 (Probability and statistics)

Resources

Website: Course material/submissions/grades are available via Moodle.

Textbook: Gould, Tobochnik, and Christian: 'An Introduction to Computer Simulation Methods'. Order a hard copy if you wish, or find a PDF copy here. Note that this textbook is in Java: to use the listings, you'll need to translate them into python (which is a useful exercise).

Textbook: McElreath: 'Statistical Rethinking'. Selected chapters, pdf copy found here.

Course Objectives

The purpose of this course is to introduce you to the principles of representing physical and biological systems in terms of mathematics and to use computers to approximate them. Additionally, we will emphasize the calibration of these models to data using Bayesian inference. Indeed, this is, explicitly or implicitly, the primary task of science generally, and I hope that this course will give you a toolbox that you can use to formulate models of systems that you may be interested in for your research or future work. In pursuing this goal, you will also become familiar with some useful tools for visualization, code optimization, and general python trickery. Specific learning objectives are as follows:

- Understand cellular automata models of complex systems
- Solve systems of ordinary (and perhaps partial) differential equations on a computer.
- Formulate hierarchical probabilistic models of physical and ecological systems.
- Use sampling techniques such as Markov Chain Monte Carlo to perform model calibration.
- Use vectorized operations (e.g. Numpy) and just-in-time compilation (e.g. Numba) to write performant code.
- Actually understand what you learned in your calculus and linear algebra classes.
- Graphically represent data and model results in multiple ways.

Below is a list of topics I expect to cover, in rough chronological order and subject to change. Please consult Moodle for an up-to-date schedule.

- Ballistics
- Population dynamics
- Orbital systems
- Molecular dynamics
- Monte Carlo simulation
- MCMC
- Cellular automata
- Chaos and fractals
- Ecological models
- Wildlife models

Course structure

This course will be a mixture of lectures and group-oriented in-class coding exercises. In particular, I will usually spend a bit of time at the start of classes motivating a problem. After this, you will work collaboratively to try and develop a solution to a problem, typically posed via Jupyter notebook. After we have completed an in-class assignment, I will ask a random group to present their results and methods to the rest of the class. *Please see moodle for an up-to-date schedule. This is the first time that I've taught this course, and it will be a work in progress!*

Assessment

The final grade in this course is computed by dividing the number of points earned by the number of points that were possible to earn. Note that I will distribute a detailed grading rubric along with each assignment. Points may be earned by completing tasks in the following categories in accordance with said rubric (note that grade percentages are given as undergraduate/graduate):

In-class assignments: 50%/40% The primary activity of this course will be the collaborative development of python code for simulation. These will be submitted as a group, and will include a self-assessement of group member contributions. Groups will be assigned by me and will be changed a few times over the course of the semester. In-class assignments will also be submitted as a group. They will be graded based on correctness of the results (40% of points for correct implementation of the method), correctness of approach (40% of points for usage of the correct method), and clarity of presentation (20% of points for code and explanations that are well-commented, make sense, are free of mechanical errors, etc.).

Homework Assignments: 50%/40% Homework assignments will involve written responses to conceptual questions or step-by-step explanations of the application of critical algorithms. These questions are designed to discover the limits of your knowledge and creativity. To the latter end, some of these questions are quite difficult. Homeworks will be graded based on correctness of result (10% of points awarded for getting the correct answer), correctness of approach (50% of points awarded for utilizing a feasible method in the attempt to solve the problem, even if mistakes are made), and clarity (40% of points awarded for clear language, formatting, and code commenting).

Graduate Project: 0%/20% As the graduate increment for this co-convened course, graduate students will be responsible for undertaking an advanced project that utilizes computer simulation. The deliverable for this project will be a paper, comparable in length and quality to a short scientific manuscript, as well as a 10 minute presentation to be given in the last few days of class. Progress towards completion of this project will be assessed via series of milestones, including (week 5) the identification of a topic and (week 10) an outline of the manuscript and working example code.

Computers

A laptop will be a requirement for the course. On the first day of class, I will assist you in setting up a machine with requisite libraries using the package manager conda, but you may set up your machine differently if you prefer.

Late Work

Each assignment will have a due date and time given in Moodle. Assignments must be submitted by that time, and late work will not be accepted. In the event of a technical issue with Moodle, you may E-mail me with your assignment: a lack of Moodle access is not grounds for an extension. I will grant extensions only under extenuating circumstances. The best way to ensure that you do not encounter issues with late work is to complete and turn in assignments early (not early in the sense of doing everything in the first week of class, but rather just turn things in a few days in advance of the due date).

Academic Dishonesty

I take an extremely dim view of academic dishonesty (described in the UM Student Code of Conduct). On the first incident, you will be awarded zero points on the associated assignment. On the second incident, your case will be referred to the Office of Community Standards and you may receive a zero for the course. Rather than cheat, I encourage you to seek help from me. You will be pleasantly surprised by how much you (and how quickly) you can legitimately understand a topic with a bit of careful conversation. Note that I do not forbid the use of generative AI, but I do think that it is foolish (and likely ineffective) to use uncritically. *I look forward to working with students to develop ways in which such tools can improve their understanding of the course material.*

Accessibility

The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and the Office for Disability Equity (ODE). If you anticipate or experience barriers based on disability, please contact the ODE at: (406) 243-2243, ode@umontana.edu, or visit www.umt.edu/disability for more information. Retroactive accommodation requests will not be honored, so please, do not delay. As your instructor, I will work with you and the ODE to implement an effective accommodation, and you are welcome to contact me privately if you wish.