Course Syllabus

Jump to Today

Course Syllabus

Please click on the syllabus icon below to access the full downloadable and printable syllabus for the course.

Downloadable Syllabus (https://canvas.umt.edu/courses/17268/files/1827692?wrap=1) (https://canvas.umt.edu/courses/17268/files/1827692/download?download_frd=1)

Course Objective

In this course we are trying to capture a small piece of our world by encoding it with computational structures. This is what is meant by the course's title; ***simulation modeling.*** Our ***model*** is a computer program that represents some natural phenomena. The ***simulation*** is running the model to investigate a question or test a hypothesis. The mathematical language of physics is readily encoded and will be a focus, but the course will also take on mathematical systems from other domains such as biology, chemistry, and planetary science. The utility of simulation modeling is apparent to scientists and engineers that wish to explore processes, reproduce patterns, or make predictions. This year, for the first time, we'll be considering the relation between simulation modeling and ***reinforcement learning,*** a new field where agents maximize rewards in a dynamic environment. Our central concern with reinforcement learning will be how the dynamic environment can be represented with computational models, and if it is good enough to allow for the so-called sim to real transition to create effective robots. There is some **evidence (https://www.youtube.com/watch?v=20gS20EFDe8&ab_channel=RoboticSystemsLab%3ALeggedRoboticsatETHZ%C3%BCrich)** that it is.

Student Outcomes

After taking this course, students will be able to:

- write numerically efficient Python programs of intermediate complexity,
- construct algorithms that capture the physical characteristics of a system,
- solve ordinary differential equations with computers,
- verify and validate computer simulations,
- manipulate the fundamental data structures of numerical computing; arrays,

- represent and understand datasets graphically, and
- relate what is taught in class to platforms for reinforcement learning and gain insight into how well sim-to-real transitions might work.

Course Logistics

Most classes will have the same structure, which follows.

Board work Each class begins by having randomly selected students go to the board and demonstrate their work. I will randomly select students ***with replacement***, meaning that the probability of going to the board is the same for each student, every time. This will take about 10 minutes and will involve 2-3 students. Each student will be graded based on a rubric appearing on the course Canvas. The presentations are to be progress reports on the problems, projects, or reports that are currently being worked on. Students should report on code they have written to aid in the simulation they are performing. The class will be encouraged to suggest improvements or call out good ideas they will use themselves. *Because of this requirement, students that can not* ***consistently*** *spend at least one hour outside of class preparing for the next class are discouraged from taking this course.*

Lecture After board work, I will lecture for about 20-30 minutes. Beyond background theory, I will focus on code examples that are similar to the assigned work.

Group problem solving Either mixed through the lecture, or at the end of lecture, groups of 2 students will have time to work on the current questions or projects. There will be about 30 minutes of time in each class dedicated to this. Groups will be set by the instructor and changed through the course.

Wrap up While students are working, I will walk around and ask students questions about what is going well, and what isn't. Based on responses, I may make some comments at the end of class to address common sources of confusion. This varies in length, but is generally less than 10 minutes.

Optimize! While students or the teacher are presenting, other students can call out approaches to coding that might be optimized. You'll be learning that numpy provides many opportunities for optimization. A student identifying a significant optimization will receive 5 percentage points of extra-credit on a question set they scored less than 100% on. No more that 100% will be awarded as a final grade for question sets.

Learning Resources

Textbooks

An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition) 3rd Edition (https://canvas.umt.edu/courses/17268/files/1796037?wrap=1) (https://canvas.umt.edu/courses/17268/files/1796037/dov download_frd=1)

by **Harvey Gould** (https://www.amazon.com/Harvey-Gould/e/B001H6U9DO/ref=dp_byline_cont_book_1) (Author), **Jan Tobochnik** (https://www.amazon.com/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Jan+Tobochnik&text=Jan+Tobochnik&sort=relevancerank&search-alias=books) (Author), **Wolfgang Christian** (https://www.amazon.com/s/ref=dp_byline_sr_book_3?ie=UTF8&field-author=Wolfgang+Christian&text=Wolfgang+Christian&sort=relevancerank&search-alias=books) (Author)

Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control 2nd Edition (https://canvas.umt.edu/courses/17268/files/1796038?wrap=1) (https://canvas.umt.edu/courses/17268/files/1796038/dov download_frd=1)

by **Steven L. Brunton** (https://www.amazon.com/Steven-L-Brunton/e/B08HJV14C2/ref=dp_byline_cont_book_1) (Author), **J. Nathan Kutz** (https://www.amazon.com/J-Nathan-Kutz/e/B00E0MUKHM/ref=dp_byline_cont_book_2) (Author)

Online Resources

I find even a laptop to be more powerful than the free instances available online, but if you're really strapped for computing resources **Google Colab** (https://colab.research.google.com/) is probably all you need to get through this class. Use a Chromebook? Sure!

Course Software

You'll need Python, numpy, scipy, and matplot lib at a minimum. To protect your system's Python from corruption, it's advisable to either use **Conda** (https://anaconda.org/anaconda/conda), or containerize an operating system. For containerization I've been using **Fedora Silverblue** (https://fedoraproject.org/atomic-desktops/silverblue/) atomic OS as a base operating system and **distrobox** (https://github.com/89luca89/distrobox) managed containers. This is working well for me!

The reinforcement learning environment I recommend is **Open AI's Gymnasium** (https://gymnasium.farama.org/index.html). This is because it appears to work on multiple platforms and be well documented.

I am aware of others. The most exciting is **Genesis** (https://github.com/Genesis-Embodied-AI/Genesis), which appears to be a complete redesign/rewrite of the simulation platform. Nvidia's environment has received a lot of use in publications related to RL, but requires Nvidia hardware to work. There is **Isaacgym** (https://docs.robotsfan.com/isaacgym/) and **Omniverse** (https://developer.nvidia.com/isaac/sim), both of which are exciting.

I won't make any requirements about the platform you chose to peruse, only the recommendation that the best chance of success is with Open AI's Gymnasium.

Canvas

You've arrived! Look here for announcements that provide lecture materials, clarifications to questions that are asked, additional material, and up-to-date grades.

Grading

Scale

The standard scale for assigning grades based on percentage of points obtained follows. Students achieving the numerical scores above are guaranteed the associated letter grade. However, if average performance is low, I may decide to assign a higher letter grade for a lower score; e.g. a B+ for a numerical score of 84. Students taking the course pass/no pass are required to earn a grade of D (63%) or better in order to pass.

Letter Grade	Percentage	GPA
A	94–100%	4.0
A−	90–94%	3.7
B+	87–90%	3.3
B	83–87%	3.0
B−	80–83%	2.7
C+	77–80%	2.3
C	73–77%	2.0
C−	70–73%	1.7
D+	67–70%	1.3
D	63–67%	1.0
D−	60–63%	0.7
F	0–60%	0.0

Categories and weights

The following assessments will be used and weighted according to the values in the table to determine final grades.

Component	Description	Weight UG/G
Problems	These correspond to the problems in the Gould book, which will be assigned on a regular basis. Assessment will be based on student presentations at the beginning of class.	20%
Projects	At the end of each chapter in the Gould text, there are several projects. These are more involved and leverage code developed in the problems. I will assign 3 of these through the semester and each is worth 15% (10% for graduate students) of the total course grade. Formal	45%/30%

	write-ups in the form of a Jupyter notebook will be submitted and graded according to a rubric.	
Exams	A full class day will be taken for students to address a problem of my choosing from the book. The test will be to see how far students can get on the problem in a short window of time, and without having help from other students. A rubric will be used to grade these.	20%
Reports	After each project students will work in teams of 4 to research how processes are implemented in a gymnasium, or platform for reinforcement learning. A final 'report' will be studying the chapter on reinforcement learning from Brunton's book. Rubric for reports.	15%
Graduate Project	For graduate students only, a project of similar complexity and depth to the projects done in this class. To be presented at the time and place of the final examination for this course.	0%/15%

Attendance

More than most other classes, attendance in this class is critical for the sake of collaboration and learning. I prefer a more mature approach to this, and just let students know they are expected to come to class. That said, if you do not inform me of an absence ***prior*** to class, I will award a grade of 0 on any in-class presentations you are randomly selected for.

Schedule

Our focus will be on four chapters of Gould, selected for their relation to gymnasium type reinforcement learning environments.

- Chapter 3, Simulating Particle Motion
- Chapter 5, Few-Body Problems: The Motion of Planets
- Chapter 8, The Dynamics of Many-Particle Systems
- Chapter 17, Visualization and Rigid Body Dynamics (*time permitting, no project, focus on quaternions*)

We will complete each unit with a project. Inserted between these units, we'll be looking at gymnasiums to create 'reports'. After completing Chapter 17 we will consider Brunton's chapter on RL.

Collaboration and Generative AI

Generative AI represents a remarkable frontier in technology, holding transformative potential for expanding knowledge and enhancing learning. Yet, there is confusion on the part of students about what acceptable use of generative AI is.

In this class, rather than making rules, I've added two features that will make the abuse of generative AI problematic for your grade. They are:

1. The in-class presentations demand that you know the details of your solution.
2. 20% of the course grade is in-class, monitored examinations. No AI can be used on these.

Each student is expected to write the code that satisfies the assignment and passes tests for correctness. Doing so without using generative AI should ensure a passing grade. Generative AI has the potential to help you learn better, but following some suggestions will help you realize that potential.

- Resist the urge to use it immediately, see how far you can get on your own. Be patient with yourself. Struggling with a problem for an hour is time well spent.
- Begin interacting with AI by asking it conceptual questions like "what is the value of using a dictionary in Python", or "what is the Runge-Kutta algorithm".
- If you're still stuck, try asking the AI extremely specific questions related to a single line of code.
- In all cases, avoid general questions that equate to 'how do I do this assignment". Generative AI may get the answer right, but you won't learn anything from that.

University Policies

Student Conduct Code and Community Standards

In cases of academic dishonesty, I will seek out the maximum allowable penalty. If you have questions about which behaviors are acceptable, especially regarding use of code found on the internet or shared by your peers, please ask me.

The **Student Conduct Code (https://www.umt.edu/campus-life/community-standards/)** at the University of Montana embodies and promotes honesty, integrity, accountability, rights, and responsibilities associated with constructive citizenship in our academic community. This Code describes expected standards of behavior for all students, including academic conduct and general conduct, and it outlines students' rights, responsibilities, and the campus processes for adjudicating alleged violations.

ODE Accommodations

Students with disabilities may request reasonable modifications by contacting me. The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and Office for Disability Equity. “Reasonable” means the University permits no fundamental alterations of academic standards or retroactive modifications. While I'm delighted to assist with disabilities, students seeking accommodation for a disability must notify me ***in advance*** of any assessment requiring modification.

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 or **ode@umontana.edu (mailto:dss@umontana.edu)** , or visit **Office of Disability Equity website (http://www.umt.edu/disability)** for more information.

Course Summary:

Date	Details	Due
Tue Jan 21, 2025	**Problem 1.1 (https://canvas.umt.edu/courses/17268/assignments/235387)**	due by 12:30pm
Thu Jan 23, 2025	**Box Ball (https://canvas.umt.edu/courses/17268/assignments/239172)**	due by 9:30am
Thu Jan 30, 2025	**ODE Integrators (https://canvas.umt.edu/courses/17268/assignments/239174)**	due by 9:30am
Tue Feb 4, 2025	**Coffee Filter (https://canvas.umt.edu/courses/17268/assignments/241380)**	due by 9:30am
Tue Feb 11, 2025	**Complete a Ballistics Object (mostly) (https://canvas.umt.edu/courses/17268/assignments/242881)**	due by 9:30am

Date	Details	Due
Thu Feb 13, 2025	Bullet Trajectory (trial project) (https://canvas.umt.edu/courses/17268/assignments/242500)	due by 9:30am
Thu Feb 20, 2025	The N-Body Problem (https://canvas.umt.edu/courses/17268/assignments/243979)	due by 11:59pm
Thu Feb 27, 2025	ODE Limitations (https://canvas.umt.edu/courses/17268/assignments/244880)	due by 9:30am
Tue Mar 11, 2025	Project 1: The Helium Atom (https://canvas.umt.edu/courses/17268/assignments/245902)	due by 9:30am
	Exam I (https://canvas.umt.edu/courses/17268/assignments/245903)	due by 11am
Tue Mar 25, 2025	Mujoco 1: Time integration (https://canvas.umt.edu/courses/17268/assignments/245905)	due by 9:30am
Thu Apr 3, 2025	Lennard-Jones and Periodic Boundaries (https://canvas.umt.edu/courses/17268/assignments/250785)	due by 10:50am
Tue Apr 8, 2025	Problems 8.3 and 8.4 (https://canvas.umt.edu/courses/17268/assignments/251296)	due by 10:30am
Thu Apr 17, 2025	Applications of Many Particle Systems Code (https://canvas.umt.edu/courses/17268/assignments/252130)	due by 9:30am

Date	Details	Due
Thu May 1, 2025	Exam II (https://canvas.umt.edu/courses/17268/assignments/259405)	due by 11am
Thu May 8, 2025	Mujoco 2: Gymnasium (https://canvas.umt.edu/courses/17268/assignments/258366)	due by 8am
Fri May 9, 2025	Project 2: Granular Materials (https://canvas.umt.edu/courses/17268/assignments/258359)	due by 11:59pm
	Bullet Trajectory ODE dydt function (https://canvas.umt.edu/courses/17268/assignments/242879)	
	Syllabus Quiz (https://canvas.umt.edu/courses/17268/assignments/208207)	