Earth/Climate 410: Earth System Modeling

Fall Semester 2023
Lecture: CSRB 2230, TuTh 9:30AM-11:30PM, In-Person
Prof. Daniel Welling (he/him, dwelling@umich.edu, CSRB 1424D)

Course Description & Learning Outcomes

The Earth is a complicated system with different time and length scales, different physical processes and characteristics, and intricate interactions between parts. The ability to create and employ *models* of the system - either as individual parts or as a combined coupled system - is a critical component of scientific studies. Because of the complexity and diversity of the system, a broad range of approaches is possible, from the empirical to the first-principles-based.

EARTH / CLIMATE 410 introduces and explores different methods for modeling parts of the Earth system. A variety of methods will be introduced, including numerical differential equation solvers, probabilistic methods to model the spread of stochastic phenomena, and much more. An emphasis on model development is made, including code testing maintenance as well as use of the Git version control system. Models will be implemented in modern Python. Students will address scientific questions using models developed in this course.

Learning Outcomes: By the end of the semester, all students should have the ability to:

1. **Design, implement, and maintain software for simulating physical processes that occur in the Earth's atmosphere.** Code *design* is the process through which software strategy is chosen and mapped out, including what equations to solve, handling of inputs and outputs, and allowing the user to configure the parameter space. Code *implementation* is the act of turning the design plan into software reality, including verification (e.g., unit testing) and debugging. Code *maintenance* is the process of adopting modern software engineering standards, including maintaining proper documentation, use of active software repositories (e.g., git), and following a *reusable code* philosophy. All three of these steps will be demonstrated and practiced by students in this course.

2. **Design & Employ models to answer scientific questions about the Earth system.** Designing and employing computer models as scientific tools is not merely the act of finding a model and executing software. Rather, it is the process of constructing testable hypotheses, designing model-based experiments to test each hypothesis, configuring and executing software to perform the experiments, and analyzing and visualizing the resulting output to come to quantitative conclusions. This includes identifying or deriving the mathematical relationships that can represent the system to be explored and constructing a robust algorithm for solving them. Part of this process includes understanding the assumptions built in to the models used and reflecting on their limitations.

3. **Communicate results and conclusions drawn from models in a clear and unambiguous manner.** Science breakthroughs are nothing if not communicated across the broader scientific community. There are several ways to communicate the results drawn from models: clear scientific writing; the creation of well-labeled figures and diagrams; and spoken presentation. Throughout this course, students will develop their ability to communicate their results through clear, creative visualizations and well-written reports.

Assignments & Grading

Students will be graded via *homework assignments*, *lab reports*, and a *final project.* There will be no exams.

Homework: Homework assignments will be short problem sets or tasks given on a weekly basis. They will

Assignment Type	Due Date	Weight
Homework (∼1 per week)	*Before* the subsequent lecture period	20%
Lab Reports (6 total)	Mondays before 5pm as assigned	60%
Final Project	*Before* Wednesday, Dec. 6th, 11:59 PM	20%

range from mathematical problem sets to code-related tasks. The goals of these assignments are to reinforce concepts shared in lecture, expand thinking on course material, ensure key coding milestones are being achieved, and increase participation. In some cases, we may work on homework sets in class together or in small groups. Results will always be submitted to Canvas and are due *before* the next lecture after they are assigned (e.g., if homework is assigned during the Tuesday lecture, it must be submitted to Canvas *before* the start of the Thursday lecture.)

Labs & Lab Reports: Labs are longer assignments with deeper coding requirements. For each lab, a set of science questions will be asked of the students. Students will develop and deploy computer models to address the questions. Labs are integrated into the lecture environment: we will introduce core concepts, algorithms, and strategies for each lab as we progress. Some time from lecture periods will be reserved for working on labs to help everyone get started effectively. An approximate schedule for the labs is given in the table below; lab due dates will be set during the semester but will typically be due on Monday after the last week they appear in the schedule.

Lab submissions consist of a working code base submitted via GitHub as well as a full lab report. Code must be fully documented and commented. The report must include,

- An introduction that outlines the problem in your own words and the science questions to be answered.
- A methodology section that covers your approach and, as necessary, a description of how to run the code to reproduce all figures presented.
- A results section that includes well-labeled figures, a full text description of each, and the quantitative conclusions drawn from them.
- A discussion of the results, including limitations of the approach and their impact on the conclusions, possible next steps, and other considerations.

The grading rubric for labs can be found on Canvas

You are encouraged to discuss the labs with other students. This includes the collaboration and development of algorithms and debugging scripts. However, you are individually responsible for producing your own technical report for each lab that is not a verbatim copy of any other student's work (this includes figures, code, write-up text, etc.). In particular, students should individually compose the discussions, comments, and responses to questions in the lab reports.

Final Project: A project will take the place of a final exam. For the final project, each student will need to design their own lab project for this class. Final submission must include:

- The lab assignment document that describes the problem, and scientific questions to be answered, a description of the physical processes of the system, and a description of the prescribed numerical method to be employed.
- Any datasets or other resources required to complete the lab.
- The lab solution, including fully commented and working code and the lab report.

Final projects are due *before* 5pm Eastern on the last day of classes.

Late Policy: Homework, Labs, and Final projects have hard-set due dates and times. Each assignment will be marked down 10% for each day late. If you are unable to complete an assignment, please contact me as soon as possible to develop a timeline for submission.

If you have a personal concern about completing coursework due to illness or other issues, please email me (dwelling@umich.edu) as soon as you can, and we will work towards a solution.

Programming Environment and Expectations

This course is programming intensive. Students will be expected to develop code in Python both *independently* and *in small groups*. This means that students will need to invest time in preparing their programming environment (i.e., setting up the Python interpreter, selecting an editor, installing required packages, etc.) and learning the intricacies of the language. Be prepared to take on these tasks early, but also expect help in learning how to set things up and

I give students a lot of latitude in terms of defining their own work flow. That said, there are a few hard requirements. Students are required to use Python 3.x and a limited subset of packages. Software must be stored on the GitHub software repository website. Details on each item are given below.

Previous Coding Experience: While this is not an introductory software course, previous coding experience is not required. We will work to build our source code together in a way that expands everyone's software knowledge. That said, those with zero prior experience will need to put in extra effort at the outset to make sure they are able to keep up throughout the semester. There will be ample opportunity to meet with me and the Instructional Assistant to ask coding questions and fix bugs.

Python & Coding Environment: All software must be written in Python 3.x using *common* scientific packages, including Numpy, Matplotlib, and Scipy. For the purposes of emphasizing core Python concepts, we will *not* use the Pandas package. Deviating from lower-level courses, software development will not make use of Jupyter Notebooks. While students are welcome to use the platform and setup they most strongly prefer, we will work from a configuration available to all students with CAEN lab access: Windows 11, Anaconda Python, and Microsoft's VS Code. VS Code has many helpful plugins to assist with coding, and I encourage the use of many of these. **Use of AI-assisted coding is not allowed. It will be considered a plagiarism infraction in violation of the Honor Code.** For information on how to launch these tools, see the help document in the Canvas course site.

Git & GitHub: Successful code developers are those who can use version control software, especially the current industry standard, *Git*. Even basic knowledge and use of Git (or similar, such as SVN or CVS) demonstrates adherence to basic code development standards and an ability to work collaboratively with a group in an open-source arena. Indeed, including "Git" on a resume' is more often a requirement for a job than an added bonus.

To this end, this course will require students to maintain their own software repositories on GitHub. Students will be expected to learn and employ basic Git commands, such as `commit`, `push`, `pull`, and `status`. Several exercises will require students to fork another student's repository, branch, change, and submit a pull request. All homework, labs, and final project submissions that have a software element *must* have an associated repository on Git.

On Pandas and Jupyter Notebooks: Jupyter Notebooks are a popular Python environment that combines rich text editing with a code execution environment. They are a powerful way to introduce Python and create an educational coding experience. However, they have important shortcomings, and in this course, we will run into those issues quickly. For that reason, **we will not be using Jupyter Notebooks in this class.**

Pandas is a popular Python package for data science applications. It is widely used and powerful. Because of this, it often becomes an entry point for new Python users, creating groups of people who cannot use basic Python functionality in sometimes embarrassing fashion. Further, it introduces esoteric commands and behavior that confuse new users when not using Pandas. For these reasons, **we will not be using the Pandas package in this class. Submitting work that uses either Jupyter Notebooks or Pandas will result in a grade of zero for that assignment.**

Course Policies

CoE Honor Code: All class activities fall under the UM CoE Honor Code. Review these policies thoroughly.

Communication: Information about the course will be shared through Canvas announcements. Students are expected to monitor the Canvas page regularly. Questions about course material should be primarily asked through the Canvas discussion board. Unless your question involves a personal issue, please ask your question on the forum so that everyone can benefit from the response. For personal questions or requests, please email me and I will make every effort to respond to emails within a day. **When emailing, please always include the course number in the subject.**

Course Recordings: Course lectures will be audio/video recorded and made available to other students in this course. As part of your participation in this course, you may be recorded. If you do not wish to be recorded, please contact me the first week of class (or as soon as you enroll in the course, whichever is latest) to discuss alternative arrangements.

Attendance: Earth/Climate/Space 410 is given as an *in person* course. While attendance is not mandatory, student success is critically correlated with *regular*, *active* attendance.

Technology Requirements & Required Materials: All required software is provided to students via CAEN lab computers. Students are expected to use Canvas receive assignments, grades, and other communications. Each of the homework assignments can be creatively solved using a multitude of external software packages obtainable from the web. However, one of the goals of these assignments is to teach you to code, not to teach you how to use other's code. **Assignments should be completed using only the software libraries we utilize during the lecture**. In other words, if it doesn't run on a standard CAEN computer, it will not get graded.

Required Materials: All software reviewed in this course is open source and freely available.

Recommended Texts: No texts are required. A comprehensive list of suggested books and other references will be shared in the course's Canvas website.

Mental Health and Well-Being: Students may experience stressors that can impact both their academic experience and their personal well-being. These may include academic pressures and challenges associated with relationships, mental health, alcohol or other drugs, identities, finances, etc. If you are experiencing concerns, seeking help is a courageous thing to do for yourself and those who care about you. If the source of your stressors is academic, please contact me so that we can find solutions together. For personal concerns, U-M offers a variety of resources, many which are listed on the Resources for Student Well-being webpage https://wellbeing.studentlife.umich.edu/resources-list.

Disability Statement: The University of Michigan is committed to providing equal opportunity for participation in all classes, programs, services and activities. Requests for accommodations by persons with disabilities may be made by contacting the Services for Students with Disabilities (SSD) Office located at G664 Haven Hall. The SSD phone number is 734-763-3000. Once your eligibility for accommodation has been determined you will be issued a verified individual services accommodation (VISA) form. Please present this form to me at the beginning of the term, or at least two weeks prior to the need for the accommodation (quizzes, labs or project).

Sexual Misconduct: Title IX prohibits discrimination on the basis of sex, which includes sexual misconduct - including harassment, domestic and dating violence, sexual assault, and stalking. We understand that sexual violence can undermine students' academic success and we encourage anyone dealing with sexual misconduct to talk to someone about their experience, so they can get the support they need. Confidential support and academic advocacy can be found with the Sexual Assault Prevention and Awareness Center (SAPAC) on their 24-hour crisis line, 734-936-3333 and at sapac.umich.edu. Alleged violations can be non-confidentially reported to the ECRT.

Covid-19 Statement: For the safety of all students, faculty, and staff on campus, it is important for each of us to be mindful of safety measures that have been put in place for our protection. By returning to campus, you have acknowledged your responsibility for protecting the collective health of our community. Your participation in this course on an in-person basis is conditional upon your adherence to all safety measures mandated by the State of Michigan and the University, including maintaining physical distancing of six feet from others, and properly wearing a face covering in class. Other applicable safety measures may be described in the Wolverine Culture of Care and the University's Face Covering Policy for COVID-19.

Lecture Schedule (Subject to Change)

Week	Date	Lecture/Example Topic	Lab Topic
1	29 Aug	Course Introduction; Git Basics	Lab 1:
	31 Aug	Basic Python; Newton's Law of Cooling	Forest Fires
2	5 Sept	Simple model of forest fires / disease spread	Lab 1
	7 Sept	Numerical integration of ODEs	cont'd
3	12 Sept	Advanced ODE methods	Lab 2:
	14 Sept	Deterministic chaos, limit cycles	Predator/Prey Model
4	19 Sept	Deterministic chaos, limit cycles cont'd.	Lab 2
	21 Sept		cont'd
5	26 Sept	Climate models	Lab 3:
	28 Sept	Energy balance models	Energy Balance
6	3 Oct	Linear algebra	Lab 3
	5 Oct	Finite difference approximations	cont'd
7	10 Oct	1-D energy balance model	Lab 4:
	12 Oct	Snowball Earth	Snowball Earth 1
8	17 Oct	**Fall Break**	Lab 4
	19 Oct	Catch-up Day	cont'd
9	24 Oct	Snowball Earth, cont'd	Lab 5:
	26 Oct	Climate tipping points	Snowball Earth 2
10	31 Oct	Vegetation & impact on climate	Lab 5
	2 Nov		cont'd
11	7 Nov	Finite differencing & diffusion	Lab 6:
	9 Nov		Ground Water Pollution
12	14 Nov	Diffusion models, cont'd	Lab 6
	16 Nov	Ground water flow	cont'd
13	21 Nov	Catch up day	
	23 Nov	**Thanksgiving Break**	
14	28 Nov	Final Projects	
	30 Nov		
15	5 Dec	Final Projects	