

CMSE 202: Computational Modeling Tools and Techniques

Course Description

A useful definition of computational science is “the use of computers to analyze and solve scientific problems.” Over the course of this semester, we will explore various aspects of computational science building off of lessons learned in your introductory programming class (CMSE 201). The continuation of introduction to computational modeling focuses on standard methods and tools used for modeling and data analysis. Topics may include statistical analysis, symbolic math, linear algebra, simulation techniques, and data mining.

In addition, we will explore application-driven modeling of various systems, with applications to the physical, life, and social sciences, and also to engineering and mathematics. While we will learn some computer programming over this semester, the goal is utilitarian – this is a course in applied computing, rather than a course intended for beginning computer science majors!

By the end of this course, you will be able to:

1. Design and write programs to solve common problems in a variety of scientific disciplines.
2. Identify and analyze appropriate pre-existing techniques, software packages and computing libraries to use in programs to solve scientific problems.
3. Choose appropriate algorithms and data types to break down scientific problems into manageable components.
4. Use modern software development methods and tools to construct reusable programs.
5. Create computer automated workflows to manipulate, analyze and visualize datasets and use this data to evaluate scientific models.
6. Demonstrate the ability to work in diverse groups to develop scientific software.
7. Explain the results from a scientific computing problem and present it both verbally and in writing.

We will work toward the goals expressed above throughout this course using a range of activities – primarily by writing software both individually and in small groups, but also through discussion, presentations, and other types of exercises.

Topics covered

The primary topics covered in this course include:

- Finding and using existing software and tools to solve scientific models.
- Tools for collaborating with other programmers/scientists (Version Control)
- More experience in programming in Python (i.e., variables and types, functions, simple data structures (strings, lists, dictionaries))
- Moving beyond the Jupyter notebooks (Command Line Interfaces).
- Tools to be better programmers and scientists
- Plotting and data visualization
- File and dataset manipulation
- Introduction to scientific computing techniques, possibly including; signal processing, graph theory, machine learning, image analysis.

Please note that **creating models to describe and understand systems** (whether they are in the physical, life, or social sciences, or in engineering) is the driving principle of this course – everything else we teach you is in service to this goal!

Required reading materials

This class has no required book or course pack. From time to time we will direct you toward outside online resources, but the main materials will be video lectures, instructor course notes, and software.

Required materials for class

In-class programming assignments are a critical part of the learning process in this course. To that end, you are expected to bring the following every day:

- Your laptop and its power cord
- An adapter for connecting your laptop to a VGA or HDMI cable (if possible)

If you do not have a laptop, or if your laptop won't run the software that we need for class, we will have spare machines to use during class.

You are also expected to have a Slack account (<https://slack.com/>) and sign up for the CMSE Courses Slack workspace (<https://cmse-courses.slack.com/>). Details for doing so are outlined in later sections.

Course activities

Class participation: Active class participation (led both by the instructor and by students) is critical to the success of this course. As such, you are expected to attend class every week, bring the required materials (most importantly, your computer and power cord) and to actively participate in the in-class discussion.

Pre-class assignments: We will assign short assignments that are due prior to class. The purpose of these assignments is to introduce new material and give you some practice with it so that we can focus on experimentation and implementation in class. These assignments will typically consist of one or more short videos or reading assignments and related questions or problems, and will be due at 11:59 p.m. the night before class via the course's Desire2Learn page. ***Completing these assignments will be critical for your success in this course.***

In-class programming assignments: Class sessions will be held twice a week, and will be broken up into presentations, discussions, and programming activities that will allow you to immediately implement (and get instant feedback on) what you have just learned. In-class programming activities will be turned in at the end of the class session via the course's Desire2Learn page.

Homework: You will have periodic programming assignments (roughly every other week) that will provide a more in-depth exploration of the materials covered in class. These will be completed either individually, in pairs, or as a group, and will be turned in by the given deadline via the course's Desire2Learn page. In general, homework assignments will be due roughly 3 business days after the material has been discussed in class. Since these assignments are only due every other week, you can expect that they will require two week's worth of out-of-class effort, so you are encouraged to start your assignments as early as possible.

Homework assignments that are submitting late will be accepted **for up to two days beyond the due date** (i.e., 48 hours past the original deadline). **If the assignment is submitted within 24 hours of the original deadline, there is a 10% penalty.** This applies even if the assignment is 1 minute late. Similarly, **if it is submitted in the 24-48 hour window, a 20% deduction is applied.** Again, **after the 48th hour, the assignment will no longer be accepted.**

Midterm and Final: This class will have one midterm and a final exam. The midterm will likely take place on March 11 for Section 002 and March 12 for Section 001. The

final exam will either take place during the last class period of the semester or during the scheduled final exam date and time, which is May 2 from 5:45pm - 7:45pm in 315 Ernst Bessey Hall for Section 001 and April 29 from 12:45pm - 2:45pm in 310 Ernst Bessey Hall for Section 002.

Semester Project: This class will have a project that will involve synthesizing the computational modeling, data analysis, and data visualization techniques that you learned over the course of the term and presenting it (them) in writing and in an oral presentation. These will be group projects and they will likely be presented during the last week of classes. More details will be available near the middle of the semester.

Course meeting time and location

Section 001 (Dr. Yang) will meet on Tuesdays and Thursdays from 3:00-4:50 p.m. in 315 Ernst Bessey Hall.

Section 002 (Dr. Guthrey) will meet on Mondays and Wednesdays from 12:40-2:30 p.m. in 310 Ernst Bessey Hall.

Other important information

Course Website and Calendar: This course uses a Desire2Learn page for course organization, which can be found at <http://d2l.msu.edu>. Accompanying course information, including the official course calendar, can be found at this website. All assignments will be handed in via Desire2Learn. Consult the class website for instructions. We will also have course discussion pages on the course's Desire2Learn page, which can be used to discuss issues relating to the course and to ask and answer questions.

Email: At times, we will send out important course information via email. This email is sent to your MSU email address (the one that ends in "@msu.edu"). You are responsible for all information sent out to your University email account, and for checking this account on a regular (daily) basis.

Class discussion: We will be using Slack (<http://slack.com>) as our means of communicating about course content as the semester progresses. We believe that this will provide an excellent avenue to have discussions not only with course instructors,

TAs, and LAs, but also between you and your fellow classmates. In order to join the Slack team that we've created for the course you should complete the following steps:

1. Go to <https://cmse-courses.slack.com/signup>
2. Once there, sign up using your @msu.edu email address. **Important:** When you create your account, use your MSU NetID as your user name. This will make it easier for your instructors to recognize you within the Slack channels.

Once you've joined the CMSE Courses Slack team, you'll need to **add yourself to two channels**. The first channel you should add yourself to is "**cmse202-s19-help**". The second one you should add yourself to is "**cmse202-###-s19**" where "###" corresponds to the section of the course that you are enrolled in. To add yourself to these channels, click on the "Channels" header and search for the appropriate channel.

The "help" channel will be the place to go for any questions about assignments in the course or issues you're having with your computer or Python. We encourage you to help out other classmates when you can! The section-specific channel will be used by your instructor for important messages relevant only to your section of the course.

In order to ensure that Slack is a appropriately used tool that does not become overly time-consuming for the course instructors, TAs, or LAs, we have a list of rules for how we expect you to use Slack. They are:

1. Before you ask a question, be sure to check the other session channels to see if the question has already been answered.
2. The Slack team is primarily for you, the students, so help each other.
3. The TAs and LAs will monitor the channels, but will defer to the students to work through things. They will only enter a conversation if students are going down the wrong path and/or there are too few other students involved. However, you should not expect that the TAs or LAs will always be available. The TAs and LAs will spend a limited amount of time "logged in" to Slack and we ask that you be respectful of their time.
4. Slack is meant to be used to help you when you are stuck with a minor issue. If you are having major issues or trouble understanding the concept, go to office hours. Office hours are meant for more in-depth discussions of course content.
5. Course instructors will rarely check Slack, only to examine progress. While they may offer help, do not rely on it. Instructors will not respond to the same student twice within a 30 minute time interval.

6. Only in rare cases should you contact an instructor through a private channel. But, if you are struggling, feel free to use this option.
7. **Do not** post your solutions to out-of-class assignments directly into Slack unless prompted by an instructor.
8. Be courteous to everyone on Slack. Students who are being rude or who are excessively posting might be banned from posting on the course Slack channel.

Class attendance: This class is heavily based on material presented and worked on in class, and it is critical that you attend and participate fully every week! Therefore, class attendance is absolutely required. An unexcused absence will result in zero points for the day, which includes the in-class programming assignment points. *Arriving late or leaving early without prior arrangement with the instructor of your session counts as an unexcused absence.* Note that if you have a legitimate reason to miss class (such as job interviews, graduate school interviews, or medical school interviews) you must arrange this ahead of time to be excused from class. Three unexcused absences will result in the reduction of your grade by one step (e.g., from 4.0 to 3.5), with additional absences reducing your grade further at the discretion of the course instructor.

Inclusive classroom behavior: Respectful and responsible behavior is expected at all times, which includes not interrupting other students, turning your cell phone off, refraining from non-course-related use of electronic devices, and not using offensive or demeaning language in our discussions. Flagrant or repeated violations of this expectation may result in ejection from the classroom, grade-related penalties, and/or involvement of the university Ombudsperson. In particular, behaviors that could be considered discriminatory or harassing, or unwanted sexual attention, will not be tolerated and will be immediately reported to the appropriate MSU office (which may include the MSU Police Department).

In addition, MSU welcomes a full spectrum of experiences, viewpoints, and intellectual approaches because they enrich the conversation, even as they challenge us to think differently and grow. However, we believe that expressions and actions that demean individuals or groups comprise the environment for intellectual growth and undermine the social fabric on which the community is based. These demeaning behaviors are not welcome in this classroom.

Academic Honesty: Intellectual integrity is the foundation of the scientific enterprise. In all instances, you must do your own work and give proper credit to all sources that you use in your papers and oral presentations – any instance of submitting another person's work, ideas, or wording as your own counts as plagiarism. This includes failing

to cite any direct quotations in your essays, research paper, class debate, or written presentation. The MSU College of Natural Science adheres to the policies of academic honesty as specified in the General Student Regulations 1.0, Protection of Scholarship and Grades, and in the all-University statement on Integrity of Scholarship and Grades, which are included in Spartan Life: Student Handbook and Resource Guide. Students who plagiarize will receive a 0.0 in the course. In addition, University policy requires that any cheating offense, regardless of the magnitude of the infraction or punishment decided upon by the professor, be reported immediately to the dean of the student's college.

It is important to note that **plagiarism in the context of this course includes, but is not limited to, directly copying another student's solutions to in-class or homework problems; copying materials from online sources, textbooks, or other reference materials *without citing those references in your source code or documentation*, or having somebody else do your in-class work or homework on your behalf.** Any work that is done in collaboration with other students should state this explicitly, and their names as well as yours should be listed clearly. When collaborating with other students, you should still be coding/writing your own solutions to the assignments and should limit your collaboration to conceptual discussions about how one might go about solving the problems, not sharing exact solutions.

More broadly, we ask that students adhere to the Spartan Code of Honor academic pledge, as written by the Associated Students of Michigan State University (ASMSU): "As a Spartan, I will strive to uphold values of the highest ethical standard. I will practice honesty in my work, foster honesty in my peers, and take pride in knowing that honor is worth more than grades. I will carry these values beyond my time as a student at Michigan State University, continuing the endeavor to build personal integrity in all that I do."

Accommodations: If you have a university-documented learning difficulty or require other accommodations, please provide me with your VISA as soon as possible and speak with me about how I can assist you in your learning. If you do not have a VISA but have been documented with a learning difficulty or other problems for which you may still require accommodation, please contact MSU's Resource Center for People with Disabilities (355-9642) in order to acquire current documentation.

Instructor contact information

Course Instructors:

Yang Yang (Section 001)
Assistant Professor in Computational Mathematics, Science, and Engineering
1510 Engineering Building
Email: yangy5@msu.edu
Web: <https://cmse.msu.edu/directory/faculty/yang-yang>

Pierson Guthrey (Section 002)
Postdoctoral Research Associate in Computational Mathematics, Science, and Engineering
1515 Engineering Building
Email: guthreyp@msu.edu

Teaching assistant:

Mathialakan (Mathi) Thavappiragasam (Section 001 and 002)
Graduate Student in Computational Mathematics, Science and Engineering
Email: thavappi@msu.edu

Learning assistants: Jacob Crabtree (001) and Julian Liber (002)

Instructor office hours and locations

Office hours and locations are as follows, and start on January 14:

	<u>When</u>	<u>Where</u>
Yang Yang	Thur. 1:00 - 3:00pm	1510 Engineering Building (EB)
Pierson Guthrey	Tues. 1:00 - 3:00pm	2504 Engineering Building (EB)
Mathi Thavappiragasam	Wed. 10:00am - 12:00pm	2504 Engineering Building (EB)
	Fri. 3:00 - 5:00pm	2504 Engineering Building (EB)
Jacob Crabtree	Thurs. 5:00 - 7:00pm	2504 Engineering Building (EB)
Julian Liber	Wed. 6:00 - 8:00pm	2504 Engineering Building (EB)

The two sections are synchronized and teach the same topics at roughly the same time - as a result, you can go to any office hours that fit into your schedule. Any course personnel can help you with questions pertaining to the course material, including in-class and homework assignments. Technical questions, including issues with Python and/or Jupyter, should be directed first to your professor, but issues specific to JupyterHub (jupyterhub.egr.msu.edu) can be sent to support@egr.msu.edu. If you have

issues relating to class administration, including missed classes, illness, VISA issues, or school-sponsored activities please contact the instructors (Dr. Yang and Dr. Guthrey).

Grading information

There are a variety of course activities, with percentage of total grade listed. More detailed descriptions of each activity can be found elsewhere in the syllabus.

<u>Activity</u>	<u>Grade percentage</u>
Participation, attendance, in-class assignments	10
Pre-class assignments	10
Midterm	15
Final	15
Homework assignments	25
Semester projects	25
Total:	100%

Grading scale

4.0	≥ 90%
3.5	≥ 85%
3.0	≥ 80%
2.5	≥ 75%
2.0	≥ 70%
1.5	≥ 65%
1.0	≥ 60%
0.0	< 60%

Note: grades will not be curved - your grade is based on your own effort and progress, not on competition with your classmates.

Honors option: You can complete an honors option for this course. The honors option requires that you complete an additional project on a topic that is different than the group project you are already required to complete. You must receive an honors option project grade of 80% or higher and must achieve an overall grade of 3.5 in the course. If you are interested in pursuing the honors option you should meet with your course instructor within the first two weeks of the course to indicate your interest and develop a project plan.

Using this course's Python setup

Instructions for downloading Anaconda (Python 3.7.x):

1. Go to the Anaconda Download webpage: <https://www.anaconda.com/download/>
2. Select the appropriate operating system (Windows | macOS | Linux) for your computer (it may auto-detect the correct operating system).
3. Download the Python 3.7 version (64 bit recommended).
4. Follow the online documentation to install Python for your specific operating system: <https://docs.anaconda.com/anaconda/install/>

Important: When you're asked if you want to add Anaconda to your path choose **yes**.

5. Open the command line program on your computer.
 - On Windows, type CMD in the run box in the Start menu.
 - On Mac, type "terminal" in the spotlight search and run the "Terminal" application
 - On Linux, open up the "Console" application
6. Type "jupyter notebook" in the command line and hit enter.

If everything goes correctly, a browser window should open up with the Jupyter interface running. If things don't work, don't worry, we will help you get started.

Backup option if you run into issues with Python on your laptop:

In the event that you're having issues with the Python installation on your laptop, you can also try using the web-hosted JupyterHub server managed by MSU. It creates a virtual environment that allows you to run simple commands and host jupyter notebooks. If you want to try using the JupyterHub, follow the directions below. Note that there are extra steps involved that require that you upload and download you Jupyter notebooks to and from JupyterHub in order to turn them in on D2L.

Important: Even if you do not plan to use JupyterHub, you are expected to activate your MSU Engineering DECS account, just in case you need access to JupyterHub at some point.

Instructions for connecting to the engineering JupyterHub server:

Every student enrolled in this class will be given an engineering computing account. If this is your first time using your Engineering account you will need to activate the account by going to the following website:

<https://www.egr.msu.edu/decs/myaccount/?page=activate>

Enter your MSU NetID. The initial password will be your APID with an @ on the end (example: A12345678@) and then they have to set a password that meets the requirements listed on the page. Verify the password. Then agree to the terms and Activate.

Once your account is activated you can access the classroom Jupyterhub server using the following instructions:

1. Open up a web browser and go to the following URL:
<https://jupyterhub.egr.msu.edu>
2. Type your engineering login name. This will be your MSU NetID.
3. Type your engineering password.

If everything is working properly you will see the main “Files” windows in the Jupyter interface.

Instructions for getting IPython notebook files into JupyterHub:

IPython notebooks (also referred to as Jupyter notebooks) are files that end with the .ipynb extension. We will give you these files for all of your assignments, you will edit them and turn in the edited files in using the class Website.

You can download the ipynb assignment files from the class website (<http://d2l.msu.edu>). Once you have an ipynb file you can load it into JupyterHub using the “upload” button on the main “Files” tab in the Jupyter web interface. Hitting this button will cause a file browser window to open. Just navigate to your ipynb file, select it and hit the open button. Once you see your filename in the Jupyter window you can just click on that name to start using that file.

Instructions for making a copy of IPython notebooks and turning them in:

When you are finished editing your IPython notebook and are ready to turn it in you will need to download the ipynb file from the JupyterHub interface.

1. With the notebook file open in Jupyter, go to the “File” menu, select the “Download as” menu option and then choose “iPython Notebook (.ipynb)”
2. Pick a place to save the file (The desktop is a good choice).
3. Make sure you make a copy of the .ipynb file for your own records.
4. Go to the Desire 2 Learn (<http://d2l.msu.edu>) class website and upload the .ipynb file into the assignment folder.

NOTE: Video versions of these instructions are available on the CMSE YouTube channel [here](#) and [here](#).