CMSE 201: Introduction to Computational Science
Solving problems with computers

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 and will acquire a variety of practical, fundamental computational skills. 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. Gain insight into physical, biological, and social systems through the use of computational algorithms and tools.

2. Write programs to solve common problems in a variety of disciplines.

3. Identify salient features of a system that can be codified into a model.

4. Manipulate, analyze, and visualize datasets and use this data to evaluate models.

5. Have an understanding of basic numerical methods (e.g., numerical integration, differential equations, Monte Carlo) and be able to use them to solve problems.

6. Be able to take 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:

- Creation of models (making mathematical representations of systems).
- The basics of functional programming in Python (i.e., variables and types, functions, simple data structures, strings, lists, tuples)
- Plotting and data visualization
- File and dataset manipulation
- Basic numerical techniques, possibly including statistics, linear regression, difference equations, Monte Carlo, agent-based modeling, numerical integration

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 online resources, but the main materials will be lecture notes and software.

Other required materials

In-class programming assignments are a critical part of the learning process in this course. To that end, you are expected to bring your laptop, power cord, and VGA adapter (to plug into the LCD projector) to class every day. 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.

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.

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 weekly) that will provide a more in-depth exploration of the materials covered in class. These will be pursued either individually or in pairs, 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.

Semester Projects: This class will have two projects that will involve synthesizing the computational modeling, data analysis, and data visualization techniques that you learned over the course of the term and presenting them in writing and in an oral presentation. In the first project, you will find a publicly-available dataset, come up with several questions that can be answered about that dataset, and use the analysis techniques you have learned to answer those questions and to report on the answer in written form. In the second project, you will find a model used in a subject that is of interest to you, explain what it is used for and how it works, and make connections to what has been done in class. Both parts of the project will be composed of written proposals, code, and a final written or oral presentation. More details will be available near the middle of the semester.

Course meeting time and location

Section 001 (Prof. O'Shea) will meet on Mondays and Wednesdays from 12:40-2:30 p.m. in 140 Akers Hall

Section 002 (Prof. Gao) will meet on Tuesdays and Thursdays from 12:40-2:30 p.m. in 2243 Engineering Building

Section 003 (Prof. Xie) will meet on Mondays and Wednesdays from 3:00-4:50 p.m. in 311 Ernst Bessey Hall.

Other important information

Course Website, Calendar, and discussion channel: 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.

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, graduate school, or medical school interviews) you must arrange this ahead of time to be excused from class. Three <u>unexcused</u> 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.

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).

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.

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 have their names as well as yours listed clearly.

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:

  Tong (Tony) Gao (Section 002)
  Assistant Professor of Computational Mathematics, Science and Engineering
   and Mechanical Engineering
  2459 Engineering Building
  Email: gaotong@egr.msu.edu
  Web: http://www.egr.msu.edu/~gaotong/

Brian O'Shea (Section 001)
Associate Professor of Computational Mathematics, Science and Engineering
    and Physics and Astronomy
1508F Engineering Building
Email: oshea@msu.edu
Web: http://www.pa.msu.edu/~oshea/

Yuying Xie (Section 003)
Assistant Professor of Computational Mathematics, Science and Engineering
    and Statistics
1513 Engineering Building
Email: xyy@stt.msu.edu
Web: https://cmse.msu.edu/directory/faculty/yuying-xie/

Teaching assistants:

Thomas Dixon (Sections 001 and 003)
Graduate Student in Computational Mathematics, Science and Engineering
Email: dixonth1@msu.edu

Xavier Brumwell (Section 002)
Graduate Student in Computational Mathematics, Science and Engineering
Email: brumwel1@msu.edu

Technical support:

Dr. Dirk Colbry
Director of HPC Studies, Department of Computational Mathematics, Science
    and Engineering
1516 Engineering Building
Email: colbrydi@msu.edu
Web: http://www.msu.edu/~colbrydi/

Instructor office hours and locations

Office hours and locations are as follows, and start on September 6:

	When	**Where**
Yuying Xie	Mondays, 10 a.m.-12 p.m.	1513 Engineering Building
Dirk Colbry (tech support)	Mondays, 10 a.m.-12 p.m.	1516 Engineering Building
Thomas Dixon	Tuesdays, 3-5 p.m.	CMSE Help Room - 1508A EB
Tony Gao	Wed., 9 a.m.-11 a.m.	2459 Engineering Building
Brian O'Shea	Fridays, 8:30-10:30 a.m.	1508F Engineering Building
Xavier Brumwell	Fridays, 12:00-2:00 p.m.	CMSE Help Room - 1508A EB

The three 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 to Dr. Colbry. If you have issues relating to class administration, including missed classes, illness, VISA issues, or school-sponsored activities please contact the instructor for your section (Dr. O'Shea for Section 001, Dr. Gao for Section 002, and Dr. Xie for Section 003).

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.

Activity	Grade percentage
Participation and attendance	20
Pre-class assignments:	20
In-class activities	15
Homework assignments	25
Semester projects	20
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.

Using this course's Python setup

Instructions for downloading Anaconda (Python 3.5.x):

1. Go to the Anaconda Download webpage: https://www.continuum.io/downloads
2. Use the "Jump to: Windows | OS X | Linux" to pick your operating system.
3. Download the Python 3.5 version (64 bit recommended).
4. Follow the directions at the bottom of the page to install Python on your specific operating system.
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" and hit enter in the Finder window
    *   On Linux, open up the console application
6. Type "ipython 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.

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. 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 Jupyter:

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 Jupyter 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 Jupyter 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.