CAAM 519: Computational Science I

Fall 2019

Instructor:	Joey Huchette	**Time:**	MWF 11:00 – 11:50
Email:	joehuchette@rice.edu	**Place:**	Duncan Hall 1042

Course Description: This course is an introduction to basic computational tools and techniques in scientific computing. We will cover core technologies (Linux, git, LaTeX, etc.), as well as basic programming principles relevant for research and practice in applied mathematics. The course will use a number of popular programming languages for numeric computing; however, it should not be viewed as an introductory programming class in any one of them. Students will be introduced to C, C++, Julia, Python, and R. The course is intended to teach *fundamentals* (how do I design a software package? How do I reuse existing best-of-class libraries for critical tasks?), rather than particulars of any one language, software library, or application area.

Desired Outcomes: After the course, the student should be able to independently perform an "end-to-end" computational task. For example, the student should be able to write a technical paper describing a research idea, and then develop, manage, and collaborate on a sophisticated software project to validate the methods developed in that paper. In the likely event that this course did not cover the particular technology best suited for the computational task, the student should leave with the ability to seek out that best tool, and learn how to apply it to their task.

Prerequisites: Informally, "mathematical maturity": i.e. a familiarity and comfort with higher-level mathematical thought. Prior programming experience is a plus, but not necessary. Formally, the students must be graduate students, or receive a special registration exemption from the instructor. Special registrations will be granted to students who have previously taken courses in CAAM (or equivalent) at the 400 level or above.

Course Website: `https://www.caam.rice.edu/~caam519/`

Office Hours: Thursday 1–2pm, or by appointment.

Main References: Lecture notes, posted on the course website, will be the primary reading materials for the course. This course has no required textbooks. If you're interested in textbooks or reference materials that may be useful, the following are good starting points, and should be available from the library:

- Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley, *LaTeX Companion: Tools and Techniques for Computer Type Setting, 2nd Edition*, Addison Wesley Professional, 2004.
- Brian Kernighan and Dennis Ritchie, *The C Programming Language, 2nd Edition*, Prentice Hall, 1988.
- Bjarne Stroustrup, *The C++ Programming Language, 2nd Edition*, Addison Wesley, 2013.

Technology: Access to a computer is essential for this course. If you do not have ready access to a computer, please contact the instructor by the end of the first week of classes.

Grading Policy: Students will be graded on the basis of four coding projects, each roughly equivalent to a problem set in terms of complexity and required effort. You are encouraged to discuss these assignments with other students, but the final write-up (including computer code) must be entirely your own work. Likewise, you are encouraged to use online resources to help with technical questions (e.g. searching StackOverflow to diagnose and fix a stack overflow in your code). However, directly copying code–from online sources or elsewhere–is strictly forbidden.

Late assignments will be penalized 25%, and will only be accepted up to one class period after the original due date without prior arrangement.

Tentative Course Outline: Note that this is subject to change (potentially substantially), depending on interest and rate of progress!

- Weeks 1-3 (8/26 – 9/13): Foundational tools (Linux, git, LaTeX, Markdown)
  - NO CLASS on 9/2 (Labor Day)
- Weeks 4-6 (9/16 – 10/11): Programming 101 (in Julia)
- Week 7 (10/14 – 10/18): Case study: Software for mathematical optimization
- Weeks 8-12 (10/21 – 11/15): C and C++
  - NO CLASS on 10/14 (midterm recess)
  - NO CLASS on 10/21 or 10/25 (conference travel)
- Week 13 (11/18 – 11/22): Data wrangling and visualization in R
- Week 14 (11/25 – 11/29): TensorFlow in Python
  - NO CLASS on 11/29 (Thanksgiving)
- Week 15 (12/2 – 12/6): TBD (e.g. overflow, parallelism, student suggestions)

Disability Policy: If you have a documented disability that may affect academic performance, you should: 1) make sure this documentation is on file with Disability Resource Center (visit Allen Center, Room 111 or email adarice@rice.edu) to determine the accommodations you need; and 2) meet with me to discuss your accommodation needs.

Rice Honor Code: In this course, all students will be held to the standards of the Rice Honor Code, a code that you pledged to honor when you matriculated at this institution. If you are unfamiliar with the details of this code and how it is administered, you should consult the Honor System Handbook at `http://honor.rice.edu/honor-system-handbook/`. This handbook outlines the University's expectations for the integrity of your academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process.