Fall 2021: Computational Science I

Instructor: Mohammad Sarraf Joshaghani
(m.sarraf.j@rice.edu)

Module 0: Course Introduction

Outline

Lecture 1 (Aug 25, 2021)

- General course overview
- Introduction
- Installing VirtualBox

Course information

Lectures: Monday/Wednesday/Friday, 11-11:50 AM
Classroom: Online (DCH 1075 after Sep 3rd)

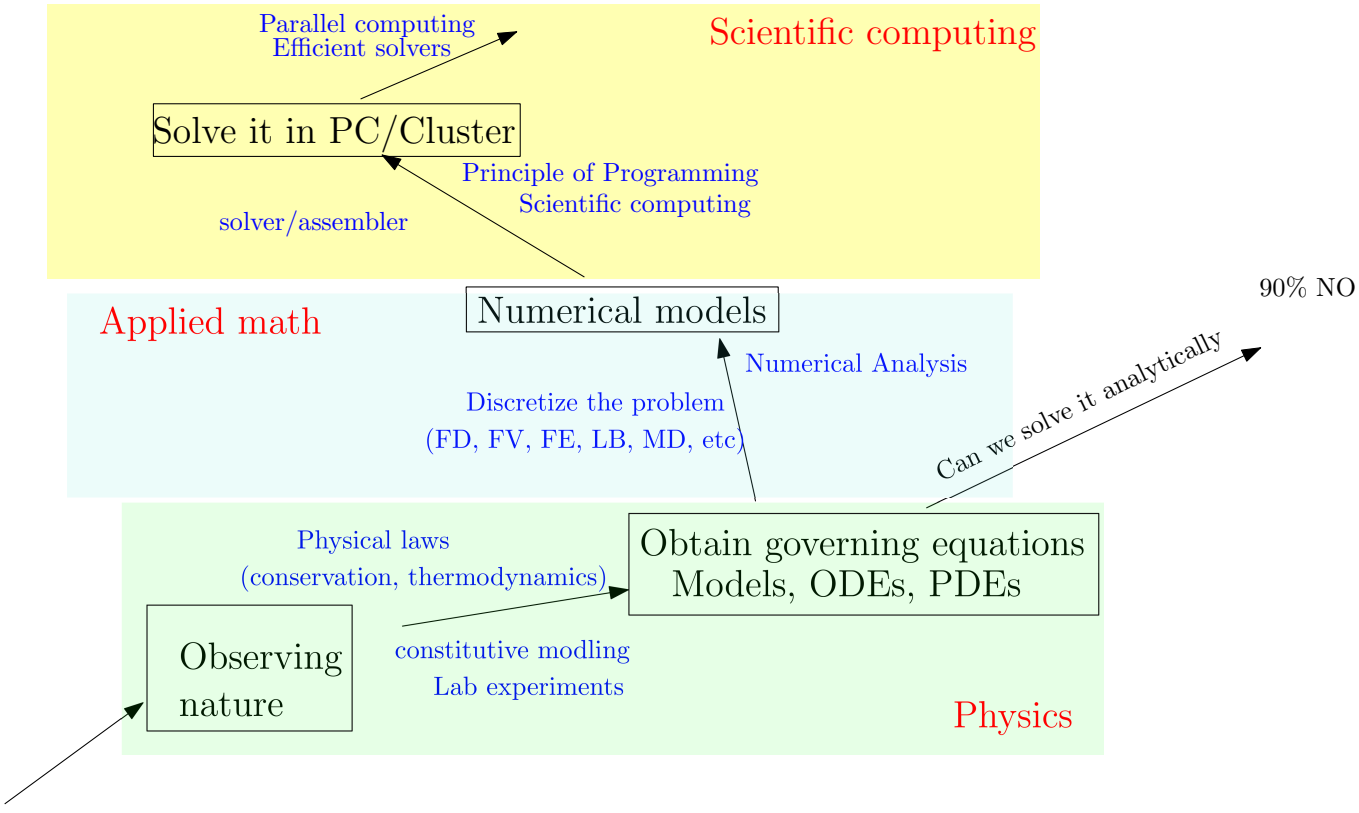
Instructor: Mohammad Sarraf Joshaghani
Email: m.sarraf.j@rice.edu

Office hours: Online (DCH 2001 after Sep 3rd)
Course Website: https://msarrafj.github.io/CAAM519-FA21/

Enrollment

- Open to graduate students
- Undergrads with "mathematical maturity" (measured by prior experience w/400+ level CAAM or equivalent) can register

Course overview



What is this course?

An introduction to:

1. Fundamental tools and techniques ubiquitous in scientific computing
2. Principles of programming for scientific tasks

Some detailed objectives

- Become comfortable with linux operating systems
- Learn on shell programming, markdown, text editors, maintaining dotfiles, etc
- Implement project organization (both software and papers/presentations) with git
- Use LaTeXfor writing papers and presentations
- 101 programming (tentative)
- Implementing mathematical concepts in C and C++
- Creating Makefiles with the GNU system
- Using the gdb debugger
- Using mathematical libraries within code

Upon completing the course, you will be able to:

- Typeset a journal paper or document software
- Manage and collaborate on a software project
- Write code to perform computational task
- Search out and learn how to use new computational tools

Logistics

- Syllabus, class website, assignments, etc.
- No required textbooks!

Tentative schedule

Weak 1-4: Basic tools

- Linux, git, LaTeX, etc.
- bash programming

Weak 5-6: Programming 101

- Basic construct (variable, loop, functions, etc),
- Data structures (arrays, sparse matrices, etc)

Weak 7-14: C and C++

- Basics of both C and C++
- Building and linking libraries

Note on schedule I will away on Nov 5 (SIAM TX-LA confernce)

Classroom experience

- Me live coding through the material
- You following along on your computer (and correcting my mistakes)

Deliverables and grading

- Evaluation based on homeworks (one assignment every 1-2 weeks)
- Typical homework: submit working code and write up
- Late policy: 25% off, up to one class period after due date, without prior permission

Honor code expectations

- Collaboration with classmates is encouraged
- Online references for coding help is encouraged
- Copying code directly is NOT allowed
- Each student must write the entirety of their code and reports independently

COVID-19 policies

- Rice pushed back the semester start day to Wednesday, August 25, 2021. For the first two weeks, the plan is for students to attend classes online to allow additional time for students to get tested and vaccinated in order to keep everyone safe.
- The online-only plan will be in effect through Friday, September 3. After that classes are F2F.
- As noted earlier in the message from Crisis Management, students - vaccinated and unvaccinated - must wear a mask in the classroom. While lecturing, vaccinated instructors can forgo wearing a mask if they maintain a distance of at least 10 feet between themselves and their students. For more information, please visit https://coronavirus.rice.edu/policies.

Introductions!

Please introduce yourself, your major, your background,
and tell us you want to get out of this class.

Poll: Linux familiarity

1. I'm familiar working with other OS (like Windows) but have not touched Linux before [$\sim$ 40%]
2. I have some basic familiarity, I can do `cp`, `ls` [$\sim$ 40%]
3. I am pretty confident working with terminal, bashrc, cluster, etc [$\sim$ 15%]
4. I have used done extensive bash programming, profiling, etc

Poll: LaTeXfamiliarity

1. I am adjusted to Office but have never written anything in TeX [$\sim$ 30%]
2. I have used it before to submit assignments. I can compile a report document in TeXStudio or other IDEs, OverLeaf, etc [$\sim$ 40%]
3. I use it regularly for my research and work [$\sim$ 20%]
4. I use it extensively and have my own preambles

Poll: Python familiarity

1. What is Python?? I hate snakes [$\sim$ 10%]
2. I have dabbled in Python, and I could probably write some for-loop and "hello world" program [$\sim$ 45%]
3. I am pretty confident with Python and I know about classes, inheritance, etc [$\sim$ 30%]
4. I use it extensively as a developer [$\sim$ 15%]

Poll: C familiarity

1. I'm familiar with programming in other languages but I've never touched C before [$\sim$ 60%]
2. I 've dabbled in C, and I could probably write some for-loop and "hello world" program [$\sim$ 20%]
3. I am pretty confident with C, makefiles and I know about pointers, memory leak, etc
4. I 've used C extensively and it is my favorite language

Poll: C++ familiarity

1. I'm familiar with programming in other languages but I've never touched C++ before [$\sim$ 55%]
2. I've dabbled in C++, and I could probably write some for-loops and a "Hello world" program
3. I'm pretty confident with C++ and I know about classes and pointers [$\sim$ 40%]
4. I've used C++ extensively, and I know about advanced features like virtual functions, templating, etc.

Poll: Version control

1. I've never used version control [$\sim$ 40%]
2. I've used some version control software but not Git
3. I've used Git occasionally [$\sim$ 30%]
4. I've used Git extensively

Poll: Parallel programming

1. I've never done any parallel programming before [$\sim 65\%$]
2. I've never used OpenMP, but I have done other types of parallel programming before
3. I've heard of OpenMP and I know some basic ideas
4. I've used OpenMP extensively

Poll: Computer access

1. I have access to a computer [$\sim$ 100%]
2. I don't have access to a computer

For next class

- Contact me if
  - you don't have access to a laptop
  - you have a disability that may affect your performance in this course
- Get set up with a Linux environment (e.g. `VirtualBox`) following instruction on the website