Scientific Programming with C

Math 1800: Special Topics
Spring 2025 Semester
Syllabus

`https://people.sc.fsu.edu/~jburkardt/classes/sc_2025/syllabus.pdf`

Description: This course gives you a start in scientific computing. We will learn how to design, implement, and test computational programs on a personal laptop, for eventual execution on a remote high performance computing system. You will pick up skills needed for working in a national laboratory, a research center, or an industrial development organization. For our course, we will learn:

- The **unix** operating system;
- Scripting with the **bash** shell;
- Elementary **C** programming;
- Compilation and execution;
- Arrays: Creating and using vectors and matrices;
- File input and output;
- Graphics now or later;
- Reading and writing digital images;
- Minimization with Nelder-Mead;
- Creating and managing coordinate grids;
- Finite difference approximations to derivatives;
- Solution techniques for small, medium and large linear systems;
- The **umfpack()** sparse matrix library;
- Solving the Poisson equation in 1D and 2D;
- Solving a partial differential equation with a spiral solution;
- Animate the spiral solutions using **Imagemagick()**;
- Timing, performance monitoring, and optimization;
- **OpenMP** for parallel execution;
- Remote access with **ssh** and file transfer with **sftp**;
- Remote job execution with **slurm**;

Prerequisite: I will try to give you a start on every topic that we consider. It would help if you knew some kind of programming, and had some idea of numerical optimization, linear algebra, and finite differences, but I will try to present enough background so that any diligent student can master the work. You should be able to find access to **unix** on your laptop. For a Mac, this is simply done through the `Terminal` command. For Windows 11 PC's, there is a way to get to **unix** from your desktop environment; I am Windows-ignorant however, so you may need a classmate to show you how;

Grading: There will be no exams. There are several computing projects that we will work on. Your grade will be based on attendance, class participation, and proper completion of the computer projects.

Text: Rouben Rostamian, Programming Projects in C for Students of Engineering, Science, and Mathematics, SIAM, 2014, ISBN: 978-1-611973-49-5

Computer software Along with **unix**, you will need to have access to a **C** compiler (I recommend **gcc**, the Gnu C compiler). You should have an editor, or better a code editor, or best an integrated development environment. You will eventually need to install the packages **netpbm**, **umfpack** and **Imagemagick**. You will need to work with **ssh** and **sftp**, but I expect these will automatically be available as part of **unix**.

Pitt maintains the **h2p** high performance computing system. I expect that we will use that system to demonstrate our remote computing skills.

The Pittsburgh Supercomputing Center has many resources for scientific computing. We may get some presentations and exercises from them as well.

Getting Help: The Pitt IT Help Desk may be able to assist you if you have trouble on your laptop. Send email to

helpdesk@pitt.edu

or check their web page at

https://www.technology.pitt.edu/247-it-help-desk

Office Hours: Office hours will be 10:00 to 10:50pm, Monday and Wednesday. My office is room 618, Thackeray Hall. My university email is `jvb25@pitt.edu`.

Disability Resource Services: If you have a disability for which you are or may be requesting an accommodation, you are encouraged to contact both your instructor and the Office of Disability Resources and Services, 216 William Pitt Union (412) 624-7890 as early as possible in the term.

Academic Integrity: Cheating and plagiarism will not be tolerated. Students suspected of violating the University of Pittsburgh Policy on Academic Integrity will incur a minimum sanction of a zero score for the work in question. Additional sanctions may be imposed, depending on the severity of the infraction.