

CAAM 519: Computational Science I

Fall 2015 Overview*

Matthew G. Knepley
3021 Duncan Hall knepley@rice.edu

1 Course Description

The goal of this course is enable students to make informed choices among computational alternatives. We will illustrate the analysis of the tradeoffs involved with examples from discretization of continuum equations, solution of algebraic systems, and libraries for parallel computing. Students will develop simple numerical and performance models of the choices, and evaluate the alternatives given the scientific or engineering goals and computational environment. This course will also create a core competence with the set of basic computational tools for solving the problems of scientific computation, as well as documenting those solutions: high-level programming languages (C, Fortran, Python), program development and maintenance tools (make, debuggers, version control systems), documentation and typesetting utilities (LaTeX, Sphinx), and scientific software libraries (BLAS/LAPACK, PETSc, SLEPc). The course itself can be divided into roughly three areas of concentration.

1.1 Infrastructure

Software development practice should be as familiar to modern computational scientists as laboratory practice to physical scientists. Building code, version control, interface design, and debugging should be second nature, and form a firm foundation mathematical modeling and algorithmic experimentation. Moreover, students will learn how to communicate their advances using \LaTeX , obtain DOIs for research products, and disseminate code using hosting sites.

1.2 Process of Computation

The mathematical framework of this course focuses on the process of computation itself, rather than the object of computation as in traditional numerical analysis. Understanding why the finite element method converges, why multi-grid is efficient, or why the convergence of Newton's method is undecidable (in

*Adapted from the CAAM 420 Fall 2014 Notes of Professor Warburton

some sense) is at the core of numerical analysis. However, it is quite apart from understanding what computational operations different element methods have in common, what the scalability bottlenecks are in algebraic and geometric multigrid, and how different nonlinear solvers behave after only a few iterates, even though the source material is the same. These kinds of questions are basic to computational science, and we will address them through examples drawn from traditional numerics, namely finite difference, finite element, and finite volume discretizations of PDEs, and linear and nonlinear solvers for systems of algebraic equations.

1.3 Modeling of Computation

Just as we cannot understand the dynamical world around us without physical models, we cannot understand the operation of algorithms and computation without numerical and performance models. Modeling both the convergence and performance is key to making the crucial choice between algorithms in a scientific library and application. We will construct and analyze models for many algorithms shown in the section above. In addition, we will consider the design of interfaces in order to minimize complexity, promote extensibility, ease maintenance and debugging, and allow efficient and scalable implementation.

1.4 Texts

There are no required texts for the course, and all material will be provided online as notes and slides by the instructor. The recommended reference texts are [The C Programming Language](#) by Brian W. Kernighan and Dennis M. Ritchie and [The L^AT_EX Companion](#) by Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley, both of which will be available from the book store. In addition, [The PETSc Manual](#), available online, should prove very helpful for the assignments.

2 Objective

CAAM 519 students learn to design, implement and manage scientific computing libraries using the C programming language, leveraging their knowledge of applied mathematics, numerical analysis, and algorithmics.

3 Absence Policy

Attendance is not required, and failure to attend class will not impact a student's grade. However, much of the information for the class will be delivered in lecture, so it may prove difficult to complete assignments without attendance.

4 Special Materials

It will not be possible to complete this course without access to a computer for coding and running assignments. If a student does not currently have access, they must contact the instructor and arrange an alternative by the first week of class.

5 Office Hours

Office hours will be held in 3021 Duncan Hall from 12pm–1pm every Wednesday. The instructor will also be available by appointment if that time is unavailable or oversubscribed.

6 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](#). 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.

7 Disability Support Services

If you have a documented disability or other condition that may affect academic performance you should: 1) make sure this documentation is on file with Disability Support Services (Allen Center, Room 111 / adarice@rice.edu / x5841) to determine the accommodations you need; and 2) talk with the instructor to discuss your accommodation needs.

8 Syllabus Change Policy

This syllabus is only a guide for the course and is subject to change without advance notice.