MATH 664-600: Computational Software for Large-Scale PDE Solvers

Lecturer:	Prof. Wolfgang Bangerth Blocker Bldg., Room 507D (979) 845 6393 `bangerth@math.tamu.edu`
Office hours:	Tue-Thu, 9:30–10:30am
Lecture:	Tuesdays + Thursdays, 11:10–12:25am Blocker Bldg., Room 164
Lab:	Wednesdays, 10:20–11:10am Blocker Bldg., Room 126

Course Outline

The course is intended to bring students to a level where they can be active users of, participants in and developers of projects that aim at the large-scale solution of partial differential equations by the finite element method, possibly using parallel and distributed computing. One of the most important lessons learned over the past decade is that due to their size such software can't be written from scratch for each new project, and must instead build on existing software libraries that handle most of what constitutes the finite element method as well as linear algebra and parallel communication. Applications then only have to implement things like bilinear forms, outer nonlinear solution loops, and linear solvers specialized to the application.

In the course, students will be taught how to build such software on top of existing libraries, in particular the deal.II adaptive finite element library as well as the PETSc parallel linear algebra library. The first part of the course will be used to review the basic mathematical concepts used in this software, such as finite element theory and iterative solvers. The rest of the course will be used for projects in which students are guided through the implementation of solvers for applications from their particular field. Part of this will be teaching the use of software engineering practices, such as the use of revision control management systems (e.g., CVS or Subversion), writing documentation, and writing tests for automated testsuites. Projects will be picked according to the field of research of a student or her interests. The goal is the development of a code that a) helps in the research of a student, and b) may be used as the starting point for future generations of students.

An outline of the topics to be covered is as follows:

- Basics of finite element implementations
- Basics of iterative solvers for large linear systems
- Basics of iterative nonlinear solvers

- Use of existing software libraries for partial differential equations and linear algebra
- Parallelization strategies for distributed computing and challenges to parallelization, such as adaptivity (depending whether there is demand for this topic)
- Collaborative software development, e.g. the use of CVS or subversion
- Software engineering practices for large-scale software, e.g. automated build and test systems, tools for documentation

This course is intended for students in the engineering disciplines as well as students of mathematics involved in research in numerics. Feedback from the involved groups would therefore be appreciated, and I will make efforts to incorporate suggestions on other topics that might be of importance to these groups.
Prerequisites

- Knowledge of the finite element method. This is covered by MATH 610, though we will probably need little of the theory developed there. An engineering equivalent of this course would definitely be sufficient. Good students of MATH 609 should also be able to read up quickly on how the finite element method works, though they may lack an understanding of why.
- Basics of iterative solvers. MATH 609 covers some of this material, and MATH 639 covers it in detail. Good students of MATH 609 should do fine, though.
- Good knowledge of C++. Students without adequate C++ programming skills will not be able to benefit from this course, and in contrast to content material, the experience needed to write programs can't be taught in a few classes at the beginning of the semester. All successful large-scale software packages are written in advanced programming languages, and students should have experience in object-oriented programming and preferrably the use of templates in C++.

Literature
To the best of my knowledge, there isn't much in terms of books on writing large-scale software for finite element simulations. What is there often treats a single software package; for example, there are books about FEMLAB and DiffPack, two commercial finite element packages. For the two packages to be used in class, there are no such books, but PETSc comes with a manual of several hundred pages, and deal.II has several thousand pages of documentation. They can be found at

- the PETSc homepage at `http://www.mcs.anl.gov/petsc/`
- The deal.II homepage at `http://www.dealii.org/`.

If anyone knows of books on the implementation of the finite element method that seem relevant to this course, I would be interested in learning about them. For the other topics, there are good books. There is a plethora of books about the theory of finite elements; for iterative solvers of linear and nonlinear systems, the following books are commonly referenced:

- Y. Saad, "Iterative Methods for Sparse Linear Systems" (an exhaustive book on the theory of iterative solvers)
- R. Barrett et al., "Templates for the Solution of Linear Systems" (a short book explaining the implementation of a variety of linear solvers)
- C. T. Kelley, "Iterative Methods for Linear and Nonlinear Equations"

The use of Subversion is explained in

- C. M. Pilato et al., "Version Control with Subversion" (this book is available online at `http://svnbook.red-bean.com/en/1.1/index.html`)

Other software design techniques such as unit testing and automated build systems are treated in many Computer Science texts, but what is needed will also be covered in class.

Webpage
Homework assignments and other course information will be posted at the course webpage

`http://www.math.tamu.edu/~wolfgang.bangerth/teaching.html`

Exams + Grading
Final course grades will be determined from the project every student gets, although projects can also be worked on in small groups. Projects will consist of writing a finite element solver for a particular equation that will generally be chosen from the area of interest of each student. The determination of the grade will be based on the following criteria:

- Sophistication of the program
- Extent of documentation of the code
- Extent of the documentation surrounding the program, i.e. description of the equation and its properties, description of the principles used in the implementation, and documentation of worked-out examples computed with the program.

As an example for projects, take a look at the deal.II tutorial programs at `http://www.dealii.org/developer/tutorial/chapter-2.step-by-step/index.html`. If students are interested, good enough projects will be published as part of the library and distributed with future versions. Of course, the authors will then be credited for their work.

Incompletes: I will consider giving an incomplete if you have successfully completed all but a small portion of the work of the course, and are prevented from completing the course by a severe, unexpected event. Simply being behind work is not a reason for an Incomplete, though; in that case you should consider dropping the course.

S/U grades: If you are registered S/U your grade will be 'S' if your letter grade is C or above, and 'U' otherwise.

Policies

Academic integrity: The usual rules of academic intregrity apply. In particular, the Aggie Honor Code "An Aggie does not lie, cheat or steal, or tolerate those who do" should be selfevident, see

`http://www.tamu.edu/aggiehonor.html`

Students may, and are encouraged to, work together and discuss homework problems with each other. However, copying work done by others is an act of scholastic dishonesty and will be persecuted to the full extent allowed by University policy.

Disabilities: If you have a disability and need special assistance, please contact me so we can make accomodations. The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accomodation of their disabilities. If you believe you have a disability requiring an accomodation, please also contact Services for Students with Disabilities, Koldus 126, 845-1637.

For other policies and other information, please read

`http://www.math.tamu.edu/teaching/operationspg.html`