Texas A&M University
Departmental Request for a New Course
Undergraduate + Graduate + Professional
- Submit original form and attach a course syllabus.

1. This request is submitted by the Department of Mathematics

2. Course prefix, number and complete title of course: MATH 676 Finite Element Methods in Scientific Computing

3. Course description (not more than 50 words): Basic finite element methods; structure of finite element codes; assembling linear systems of equations and algorithmic aspects; linear iterative solvers; adaptive mesh refinement; vector-valued and mixed problems; nonlinear problems; visualization; parallelization aspects. Additional topics may be chosen by instructor.

4. Prerequisite(s): MATH 610, ENGR finite element class or MATH 417/420 plus instructor approval. Knowledge of C++

   Cross-listed with:

   Cross-listed courses require the signature of both department heads.

5. Is this a variable credit course? □ Yes ☑ No  If yes, from _____ to _____.

6. Is this a repeatable course? □ Yes ☑ No □ Yes ☑ No  If yes, this course may be taken _____ times. Will the course be repeated within the same semester/term?

7. Has this course been taught as a 289/489/689? ☑ Yes □ No  If yes, how many times? 1  Indicate the number of students enrolled for each academic period it was taught. 06A-11

8. This course will be:
   a. required for students enrolled in the following degree program(s) (e.g., B.A. in history)

   b. an elective for students enrolled in the following degree program(s) (e.g., M.S., Ph.D. in geography)

   M.S., Ph.D. in Mathematics

9. If other departments are teaching or are responsible for related subject matter, the course must be coordinated with these departments. Attach approval letters.

10. Prefix | Course # | Title (excluding punctuation)
       | MATH | 676 | FEM IN SCIENTIFIC COMPUT

       | Lect. | Lab | SCH | Subject Matter Content Code | Admin. Unit | Acad. Year | FICE Code | Level |
       | 3 | 0 | 0 | 0 | 3 | 0 | 2 | 7 | 0 | 3 | 0 | 3 | 0 | 0 | 1 | 1 | 8 | 7 | 5 | 0 | 9 | - | 1 | 0 | 0 | 0 | 3 | 6 | 3 | 2 |

Approval recommended by:

Head of Department  Al Boggess  Date: 11/3/07
Chair, College Review Committee  Date: 11/29/07

Head of Department (if cross-listed course)  Date: 11/29/07
Dean of College

Submitted to Coordinating Board by:
Dean of College
Date

Director of Academic Support Services
Date
Effective Date

Questions regarding this form should be directed to Sandra Williams at 845-8836.
OAR/AS – 04/07

1 of 5 B17
MATH 6XX
Finite element methods in scientific computing

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

Course Outline
The course is intended to give students a perspective on the practical aspects of the finite element method, in particular on typical finite element software is structured, on algorithmic details of how to efficiently implement it, and the pre- and postprocessing steps necessary in the scientific computing workflow.
The course will use a large Open Source finite element library and other open source tools for the entire workflow. The library, deal.II (see http://www.dealii.org) is developed mainly at Texas A&M University in collaboration with other researchers worldwide. It is used as the standard research and teaching tool for numerical computing in the Department of Mathematics. One of the most important lessons learned over the past decade is that due to their size modern numerical 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.
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 an application related to their research project or interests. Part of this will be teaching the use of software engineering practices, such as the use of revision control management systems (e.g. Subversion), writing documentation, and writing tests for automated test suites. 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. It may also be converted into a tutorial for deal.II.
An outline of the topics to be covered is as follows:

- Basics of finite element methods
- Structure of finite element codes
- Assembling linear systems of equations and algorithmic aspects

I
• Iterative solvers for large linear systems
• Adaptive mesh refinement
• Vector-valued and mixed problems
• Nonlinear problems
• Postprocessing, visualization, and parallelization aspects
• Collaborative software development, e.g. the use of subversion
• Software engineering practices for large-scale software, e.g. automated build and test systems, tools for documentation
• Additional topics may be chosen based on student interests.

This course is intended for students of mathematics involved in research in numerical methods as well as students in the engineering disciplines.

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 preferably 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 packages to be used in class, there are no such books, but deal.II has several thousand pages of documentation. It can be found at the deal.II homepage at http://www.dealii.org/. 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 (see for example the step-21, step-24, and step-25 tutorial programs). Of course, the authors will then be credited for their work.

Incomplete: 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 integrity 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