ENVR 666
Numerical Methods
Fall Semester

1 General Information

Instructor:	C.T. Miller
Office:	3201 McGavran-Greenberg Hall
Office Hours:	By Appointment
Office Phone:	966-2643
E-mail:	casey_miller@unc.edu

2 Grading Bases and Student Obligations

The bases for assigning grades is described in the following table. The grading bases, and the corresponding expectations, for the course are different for undergraduate students than they are for graduate students. The significant differences are that in class exams will be given to the undergraduate students to assess their grasp of the material, and the graduate students will be required to do significant, individual work on numerical methods, which ideally will be related to their research. All students will develop a course notebook to document their efforts and learning of the course material.

Item	Undergraduate Students	Graduate Students
Mid-Term Exam	25%	—
Final Exam	25%	—
Course Notebook	50%	50%
Semester Project	—	50%

3 Course Objectives

Objectives of the course are for students to accomplish the following:

1. to understand traditional and evolving methods available for solving linear and nonlinear algebraic equations, ordinary differential equations, and partial differential equations, which are common across engineering, science, physics, and applied mathematics;

2. to learn basic scientific computing principles needed to implement computer coded solutions for the above methods;

3. to gain experience in fundamental aspects of numerical methods and applied mathematics; and

4. to learn to perform and present the results of independent work in numerical methods (graduate students only).

4 Background Required

While the focus of this course is, as the name implies, on numerical methods originating in applied mathematics, the motivation is to consider classical and evolving approaches that are important for the solution of problems that arise routinely across science and engineering. The special focus of this course is on methods that are useful in creating approximations to systems of partial differential algebraic equations, which for example arise routinely in the formulation of models based upon conservation of mass, momentum, and energy. The background that is considered essential for this course is the following:

1. mathematics through differential equations;

2. working knowledge of partial differential equations;

3. some exposure to numerical methods; and

4. computer programming skills.

The subject matter of this class requires computer literacy and a background, or willingness to learn a programming language. C++ is preferable, but other languages, including high-level languages such as Matlab, are acceptable for undergraduates. Please note though that the purpose of this course is not to explicitly teach a programming language. Students should either have this skill or be prepared to learn on their own.

Essentially all of the methods discussed in class can be adequately developed on a modern personal computer or workstation. In some cases, students may desire access to more powerful machines. If this arises, we will try to facilitate your access to an appropriate computational platform.

5 Exams

A mid-term and a final exam will be given to undergraduate students enrolled in the course. These exams will be in class, open book, open note exams intended to assess the students knowledge of the material covered and their ability to apply this material.

6 Course Notebook

Conventional style homework assignments will not be a component of this course. Instead, it is expected that each student will take the initiative to learn the material on his or her own, at a level needed to demonstrate the attainment of objectives 1 through 3, described above. Documentation and grading of this aspect of the course will require each student to develop a course notebook, and a set of referenced computer codes to accompany the course notebook.

The course notebook should be a complete account of the work performed by the student to develop and implement solution algorithms for all phases of the course: linear solvers, nonlinear solvers, ordinary differential equation solution methods, and partial differential equation solution methods. This notebook should be arranged into chapters for each major class of method and sections for each specific method implemented. For example, linear solvers should be a chapter and lower-upper decomposition would be a section. It should be noted that the course notebook *is not* a repeat of the lectures given in class. The course notebook is a document that describes the methods that the student has understood in detail, developed an algorithm to solve, coded or obtained a code to solve the method, and established that the code is error free.

The typical content for a section in the course notebook should be:

1. summary—an overview of one or two paragraphs describing the method, its uses and limitations, and the origin and features of the code developed;

2. formulation—a mathematical formulation of the method, which need not be long and should be referenced to the published literature;

3. algorithm—a concise summary of the solution algorithm used to implement the method;

4. code—discuss the location of the code (web site for ftp download or subdirectory on a usb drive enclosed with the notebook), structure of the code developed, and options and limitations;

5. validation—discuss steps taken to ensure that the code is error free; and

6. uses—briefly summarize the uses of the method.

Each such section should be on the order of five single-spaced pages. Longer write-ups will take time away from considering alternative methods, and an important objective of the course is to consider a range of methods.

The respective methods covered in the notebook are intended to represent the set of tools that the student has mastered and can apply to any given problem. It is expected that all of the major classes of topics covered in class (linear algebraic equations, nonlinear algebraic equations, ordinary differential equations, and partial differential equations) will also be covered in the student's notebook. It is also expected that the student will develop an algorithm and code for *at least* one method on their own for each of the major classes. It is

further expected that solutions obtained will be shared among students in the class and that high-quality public-domain codes will be downloaded, validated, and written up to increase the methods that a student gains experience with during the semester. The author of any portion of code should be clearly labeled and always acknowledged.

The course notebook should be prepared using a suitable text processing system. LaTeX is an elegant system for preparing documents with a significant number of mathematical symbols involved. The report document style is a good choice for the course notebook and the semester project. An extensive macro package written by the instructor will be made available, which will further aid the typesetting process.

7 Semester Project

The semester project is the vehicle used to demonstrate the attainment of course objective 4 described above, and it is required for all graduate students. The semester project should be about half of the total course effort for graduate students. A separate project topic will be chosen by each graduate student from a list of potential topics, or by agreement with the instructor. Work on the project should begin immediately and proceed throughout the semester. The status of the graduate student's progress on the project will be provided to the instructor via periodic updates.

The semester project will be submitted near the end of the semester in standard research notes format and presented orally. Computer codes, data files, and simulation output details will be submitted electronically. The LaTeX report style is a good choice for the formatting style of the semester project.

The following outline should be used for the semester project report write-up:

1. Title Page
2. Abstract
3. Notation
4. Table of Contents
5. List of Tables
6. List of Figures
7. Introduction
8. Background
9. Formulation
10. Results

11. Discussion
12. References
13. Appendices

A suitable project topic should be decided early in the semester in consultation with the instructor. This topic should be in an area of interest to the student, hopefully in an area consistent with the research goals of the student. While a wide-range of topics are possible, some typical examples are listed below in the area of methods development and evaluation:

1. development of a Krylov-deferred correction solver for DAE's;
2. development of an adaptive discontinuous Galerkin solver;
3. development of a multiscale finite element method;
4. development of a split-operator method with error control; or
5. development of a multiscale, multiphysics method.

Another class of topics is suitable as well: method applications. The emphasis is on applying methods learned in class to a problem of interest that has intrinsic merit but which has not been studied. Students are welcome to suggest possible topics or to ask for suggestions from the instructor. The success of this class of project depends heavily on the careful selection of an appropriate topic.

8 References

While no book is required for the course, many would be useful. These will be discussed throughout the semester. In the past, the class has ordered or down loaded books from the Society for Industrial and Applied Mathematics (SIAM). In addition, the instructor will distribute course notes that cover many of the topics discussed in lecture. Students should mention references that they find that are useful and request additional references for concepts that they find difficult to understand based upon the materials provided. Blackboard will be used to distribute course materials.

9 Course Outline

1. Introduction
2. Linear Equations
   (a) Direct Solvers
   (b) Banded Direct Solvers
   (c) Stationary Iterative Solvers
   (d) Krylov-Subspace Iterative Solvers
   (e) Preconditioning
3. Nonlinear Equations
   (a) Fixed-Point Iteration
   (b) Newton Iteration
   (c) Quasi Newton Methods
   (d) Strongly Convergent Methods
4. Ordinary Differential Equations
   (a) Forward Euler Method
   (b) Backward Euler Method
   (c) Runge-Kutta Methods
   (d) Error Estimation and Control
   (e) Adams Methods
   (f) Backward Difference Methods
   (g) Deferred Correction Methods
5. Finite Difference Methods
   (a) Taylor Series Formulation
   (b) Spatial Difference Methods
   (c) Temporal Difference Methods
   (d) Truncation Error Analysis
   (e) Stability Analysis
6. Algorithms
   (a) Fully Coupled
   (b) Sequential Split Operator
   (c) Strang Splitting
   (d) Alternating Split Operator

(e) Iterative Split Operator
(f) Error Estimation and Control

7. Finite Volume Methods

(a) General Formulation
(b) Upstream Weighting
(c) Lax Wendroff Method
(d) Godunov's Theorem
(e) Total Variation
(f) Slope Limiters
(g) Flux Limiters

8. Finite Element Methods

(a) Variational Approach
(b) Weighted Residual Method
(c) Bubnov-Galerkin Method
(d) Discontinuous Galerkin Methods
(e) Collocation Methods
(f) Multiscale Methods