Mathematics 5620

Introduction to Numerical Analysis II

Spring 2004 (MTWF 8:35-9:25, JWB 333)

Instructor: Grady Wright **Phone:** 581-8649
Office: JWB 126 **E-mail:** wright@math.utah.edu
Office Hours: MTWF 9:35-10:35, or by appointment

Text:

- Atkinson, An Introduction to Numerical Analysis, 2nd ed., John Wiley and Sons, 1987
  We will use this text for studying numerical linear algebra (Ch. 7, 8, and 9).
- Iserles, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, 1996.

Website: http://www.math.utah.edu/~wright/courses/5620

Course Description: This is the second half of a yearlong introductory course on numerical analysis at the 5000 level. The topics covered this semester include numerical linear algebra (solving systems of linear equations and solving eigenvalue problems), numerical methods for solving ordinary differential equations (ODEs) (both initial-value and boundary-value problems), and numerical methods for solving partial differential equations (PDEs) (primarily finite difference methods, but, if time permits, we may also study spectral and finite element methods).

Prerequisites: No previous experience in numerical analysis is necessary. However, knowledge of the following topics is required:

- Single and multivariable calculus
- Linear algebra
- Computer programming

Homework: There are weekly homework assignments consisting of three to four problems. Homework is handed out in class (and posted on the class website) on Wednesday and due the following Wednesday in class. The problems involve a mix of theory and computing. Regarding the latter, please read the text below the **Programming** heading.

Your submitted homework should show all necessary work you used to solve the problems; mathematical statements should be complete (or nearly complete) sentences; and the reasoning and logic underlying all arguments should be clearly spelled out. Please see the document "How to Present your Work" on the course web page for tips on meeting these requirements. **Failure to adhere to the above requirements may result in a loss of points.**

Teamwork: Teamwork is part of the real world and therefore permitted (and encouraged) for all homework assignments (but not exams). Keep in mind that the purpose of teamwork is to enhance the learning effect, not to decrease the workload. It is up to each team member to prevent abuse. Please observe the following rules:

- **No teams with more than three members!**
- If you work in a team then clearly state so and hand in just one set of answers. If you collaborate on some problems but not on others then **clearly state so** on your turned-in solution.

- In general, you can consult literature and people, but **you have to acknowledge all help so obtained** (except for the textbook or myself).

Homework solutions (taken from selected students solutions) are posted on the course website once the grading has concluded.

Grading Policy: The final grade for the course is on based homework assignments, a midterm exam, and a final project. The breakdown for the course grade is as follows:

- Homework: 60%
- Midterm exam (probably at the beginning of March): 20%
- Final Project: 20%

Your lowest homework score is discarded. Note that unlike last semester, no portion of your midterm score can be discarded.

As mentioned above, homework assignments are due on Wednesday **in class**. Late homework assignments are accepted up to three days beyond their due date. However, a 10% penalty is applied for each day they are late. This means that if an assignment is turned in on Saturday, then 30% is automatically deducted. Late homework assignments need to be put into my box (located JWB 228) or, if you have the means, e-mailed to me. **Please indicate the date and time you put the homework in my box.**

Programming: All programming assignments involving numerical computations are to be done in MATLAB. An important part of numerical analysis today is the use of commercial or public domain software packages for solving particular problems. In order to gain exposure to this side of numerical analysis, students need to practice using such packages. MATLAB offers the perfect opportunity for such practice; it is one of the most dominant commercial computing environments.

If you have not used MATLAB previously, help resources are available on the course website and from the instructor. MATLAB is available in the Undergraduate and Graduate student mathematics computing labs (as well as various other locations on campus). For more information on these labs, including hours of operation and location, go to http://www.math.utah.edu/ugrad/lab/.

Problems involving symbolic computations can be done using *Maple* or *Mathematica*. The former is available in the Undergraduate and Graduate student mathematics computing labs.

Final Project: Instead of a final exam, you will prepare and present a final project to the class. The purpose of the final project is to investigate a problem in numerical analysis that interests you. For example, if you are a mathematician, you may be interested in comparing (in detail) the accuracy, stability and convergence of some different numerical methods we discuss, or perhaps developing your own technique. If you are a computer scientist, you may choose to implement (optimally of course) a software package for some complicated algorithm we discuss. If you are an engineer (or applied mathematician), you may be interested in applying some numerical method we discuss to some physical problem (for example, solving some ordinary or partial differential equations). The focus of your investigation should be primarily on numerical methods we discuss this semester, although exceptions can be made if a persuasive argument is made.

You must work in groups of two or three on the project.

The final project will consist of a proposal for the investigation you wish to undertake (due April 5), a final **typewritten** report (5 to 30 pages), and a final presentation (15 to 20 minutes). More information about the project will be given as the semester proceeds. This does not mean that you should procrastinate until then; start thinking about a possible project now!

Important Dates:

- **Jan. 19** – Martin Luther King Jr. day, i.e. **No Class**
- **Jan. 21** – Last day to drop term and first session classes (and get your money back)
- **Jan. 26** – Last day to add classes and to elect CR/NC or audit for term and first session
- **Mar. 5** – Last day to withdraw from term length classes
- **Mar. 15 – 19** – Spring break, i.e. **No class**
- **Apr. 28** – Classes end
- **May 5** – Final project presentations, 8:00-10:00am

Material to Cover: We will cover the following material (in roughly this order):

(Legend: K.A. = Atkinson, A.I. = Iserles)

1. Linear Algebra (K.A. Chapter 7)
    a. Vector spaces, Matrices, and Linear systems (K.A. 7.1)
    b. Special matrices and properties
    c. Common matrix structures
    d. Norms (K.A. 7.3)
2. Numerical Solution of Linear Systems (K.A. Chapter 8)
    a. Direct Methods
        i. Gaussian elimination (K.A. 8.1-8.2)
        ii. Cholesky method (K.A. 8.3)
        iii. Error analysis (K.A. 8.4)
        iv. Residual correction (K.A. 8.5)
    b. Iterative Methods
        i. Jacobi (K.A. 8.6)
        ii. Gauss-Seidel (K.A. 8.6)
        iii. Successive Over-Relaxation (K.A. 8.7)
        iv. Conjugate Gradient method (K.A. 8.9)
    c. QR factorization (K.A. 9.3)
3. Computing Matrix Eigenvalues (K.A. Chapter 9)
    a. Review properties of eigenvalues
    b. Gerschgorin theorem (K.A. 9.1)
    c. Power method and Rayleigh Quotient (K.A. 9.2)
    d. Jacobi's method
    e. QR Factorization (K.A. 9.3)
    f. Householder reflections (K.A. 9.3)
    g. Givens' rotations
    h. QR method (K.A. 9.4-9.5)
    i. Inverse iteration (K.A. 9.6)
4. Singular Value Decomposition
    a. Least squares solutions of overdetermined linear systems (K.A. 9.7)
5. Ordinary Differential Equations (ODEs; K.A. Chapter 6 and A.I Chapters 1-6)
    a. Introduction (K.A. 6.1)
    b. Initial Value Problems

        i. Taylor's method (A.I 1.3)
        ii. Linear multistep methods
            1. Euler (A.I. 1.2, K.A. 6.2)
            2. Adams (A.I. 2.1, K.A. 6.7)
            3. Backward differentiation formulae (A.I. 2.3)
            4. Accuracy (A.I. 2.2, K.A. 6.8)
            5. Stability (A.I. 2.2, 4.1-4.4, K.A. 6.8)
            6. Dahlquist stability barriers (A.I. 4.4
            7. Predictor corrector strategies (K.A. 6.6)
        iii. Runge-Kutta methods (A.I. 3.1-3.4, K.A. 6.10, 4.3)
        iv. Extrapolation methods
    c. Boundary Value Problems
        i. Shooting method (K.A. 6.11)
        ii. Finite differenced-based linear system methods
6. Partial Differential Equations (PDEs)
    a. Introduction
        i. Character of solution
        ii. Well posedness
    b. Wave-type (Hyperbolic) Equations (A.I. Ch. 14)
        i. Finite difference methods
        ii. Method-of-lines (MOL)
        iii. Stability and convergence
            1. CFL condition
            2. von Neumann stability analysis
            3. Lax equivalence theorem
        iv. Pseudospectral methods (time permitting)
    c. Diffusive (parabolic) equations (A.I. Ch. 13)
        i. Crank-Nicolson method
        ii. ADI method
    d. Equilibrium (elliptic) equations
        i. Finite difference schemes (A.I. Ch. 7)
        ii. Iterative methods (A.I. Ch. 10)
        iii. Fast Poisson solvers (A.I. Ch. 12)
        iv. Multi-grid methods (time permitting) (A.I. Ch. 11)
        v. Finite-element methods (time permitting) (A.I. 8.3)

Other good reading:

- Trefethen and Bau, Numerical linear algebra (SIAM)
- Golub and Van Loan, Matrix Computations (Johns Hopkins University Press)
- Demmel, Applied Numerical Linear Algebra (SIAM)
- Fornberg, A Practical Guide to Pseudospectral methods (Cambridge University Press)

There are 10 kinds of people in this world; those who understand binary and those who don't. –William Woessner