Course Outline: Physics 392/692, Spring 2011

Professor: Greg Cook   Office: 304 Olin

Text: Numerical Recipes: The Art of Scientific Computing, 3rd Edition

By Press, Teukolsky, Vetterling, & Flannery

This syllabus is to be considered a non-binding outline of the pace of this course. The exact dates of material covered may change.

1. Numerical Errors, Differentiation, & Heaviside Calculus

    2 Classes: Jan. 13, 18

2. Interpolation & Extrapolation

    2 Classes: Jan. 20, 25

3. Integration & Gaussian Quadrature

    3 Classes: Jan. 27, Feb, 1, 3

4. Random Numbers

    2 Classes: Feb. 8, 10

5. Root Finding

    2 Classes: Feb. 15, 17

6. Integrating ODEs

    4 Classes: Feb. 22, 24, Mar. 1, 3

7. Linear Algebra & Eigensystems

    5 Classes: Mar. 15, 17, 22, 24, 29

8. Fourier Methods

    2 Classes: Mar, 31, Apr. 5

9. Partial Differential Equations

    6 Classes: Apr. 7, 12, 14, 19, 21, 26

Goals: The goals of the class are to learn how to effectively solve physicsal problems using numerical techniques. We will learn how to understand the errors inherent in numerical methods for solving problems, and to what extent we can trust the answers that we obtain. We will cover a broad class of techniquies that are frequently combined to solve real-world physical problems. Our goal is not to become expert in any of the individual techniques, but to give you a solid foundation of useful numerical methods, and the experience necessary to explore more sophisticated techniques on your own when necessary.

Programming: There is no programming prerequisite for this class, however, prior programming experience will be helpful. We will make use of `Matlab` as our primary programming tool. This is a powerful, yet easy to use problem solving environment. Instruction on progrmming will be incorporated into the lectures and in-class exercises. A useful resource is ``A Matlab Primer'' by Sigmon and Davis, the beginning of which can be found online at `http://books.google.com`. If possible, we will also incorporate an introduction to `C++` programming within the Linux environment.

Course Work: In addition to regular homework, you will complete a number of projects which will take the place of in-class tests.