Course Outline: Physics 335/635 (Spring 2021)

Professor: Greg Cook — Office: 304 Olin
Text: Numerical Recipes: The Art of Scientific Computing
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. 28, Feb. 2 HW 1: due 2/9

2. `Interpolation & Extrapolation`

   2 Classes: Feb. 4, 9 HW 2: due 2/16

3. `Integration`

   2 Classes: Feb. 11, 16 HW 3: due 2/23

4. `Random Numbers`

   2 Classes: Feb. 18, 23 HW 4: due 3/4

5. `Root Finding`

   2 Classes: Feb. 25, Mar. 4 HW 5: due 3/11

6. `Integrating ODEs`

   3 Classes: Mar. 9, 11 HW 6: due 3/18, 16

7. `Linear Algebra & Eigensystems`

   3 Classes: Mar. 18 HW 7: due 3/25, 23, 25 HW 8: due 4/6

8. `Fourier Methods`

   2 Classes: Mar. 30, Apr. 1

9. `Partial Differential Equations`

   8 Classes: Apr. 6 HW 9: due 4/15, 8, 13, 15 HW 10: due 5/4–6, 20, 22, 27, 29

10. `Group Presentations`

    2 Classes: May 4, 6

Goals: The goals of the class are to learn fundamental numerical techniques that can be used to effectively solve physical problems. 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 techniques 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 programming will be incorporated into the lectures and in-class exercises. A useful resource is `'A Matlab Primer'` by Sigmon and Davis.

If you are already very comfortable with programming, you might want to explore scientific programming using Python. I highly recommend using the ANACONDA platform.

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

WFU Bulletin entry:

`335. Computational Physics`. (3h) An introduction to finding numerical solutions to scientific problems. Topics include understanding computational errors, differentiation, integration, interpolation, root finding, random numbers, linear systems, Fourier methods, and the solution of ODEs and PDEs. There is no computer programming prerequisite. Credit will not be given for both PHY 335 and CSC/MTH 355. P–MTH 205 or MTH 113, 121 and 251, or POI