PHYS 305
Computational Physics II
Winter 2022

Instructor:	Prof. S. McMillan (Disque 815, x2709)
Time and place:	Tu 3:30-4:50, F 1:30-2:50, Disque 704
	or zoom: `https://drexel.zoom.us/my/slm23`
Office hour:	F 3:00-4:00, Disque 815, or by appointment
e-mail:	`slm23 (at) drexel.edu`
URL:	`http://www.physics.drexel.edu/~steve/Courses/Comp_Phys/Physics-305`

Course Overview

This is the second in a series of hands-on "computational labs" designed to complement the traditional "lecture/lab/recitation" Physics instructional sequence. It is intended to be taken after the Contemporary Physics (PHYS 113, 114, 115) or the Fundamentals of Physics (PHYS 101, 102, 201) sequences. Other prerequisites are MATH 210 (Ordinary Differential Equations) and PHYS 324 (Topics in Mathematical Physics), or their equivalents. It will be assumed that the student is familiar with the Python programming language; the course will be conducted primarily in Python. Familiarity with a more efficient lower-level language, such as C or C++, may be helpful for some applications. Students will be introduced to basic scientific programming techniques and problem-solving strategies using examples and case studies drawn from a variety of sources. Case studies will include the solution of systems of simple differential equations, investigation of the properties of chaotic systems, many-body systems in astrophysics and/or molecular dynamics, and the time-independent Schrödinger equation.

Course Outline

A detailed course outline is posted on the course web page, and will form the day-to-day teaching framework for the lectures and in-class work. It will grow as the course progresses. A short preliminary list of topics follows. This list will also likely evolve during the course in response to student suggestions and interests.

- Review of Python, numpy, and matplotlib
- Numerical differentiation and integration
- Overview of ODEs
  - reduction to standard form
  - initial-value problems
  - boundary-value problems
- Integration algorithms for initial-value problems
  - Runge-Kutta methods
  - explicit and implicit methods

- applications and refinements
- predictor-corrector schemes
- the Leapfrog integrator
- symplectic methods

- Fourier analysis of data
- Application: nonlinear systems and chaos
  - damped and driven harmonic motion
  - time sequences, phase portraits, and Poincare sections
  - periodic, quasiperiodic, and chaotic motion
  - self-similar structure
  - attractors and basins of attraction
- Application: N-body dynamics in astrophysics and molecular dynamics
  - direct-summation techniques
  - treecodes
  - the few-body problem
- Algorithms for boundary-value problems
  - solving algebraic equations
  - shooting
  - relaxation
- Application: the one-dimensional Schrödinger equation

Textbook

There is no assigned text book for this course. The course web site and links therein will be the primary source. Details on much of the computational material can be found in the text *Numerical Recipes in C*, by W. Press, S. Teukolsky, W. Vetterling, and B. Flannery (1992, Cambridge University Press). Free online versions are available—see the links on the course web site.

Evaluation

Grading will be based on roughly 6-7 homeworks. There will be no mid-term and no formal final examination. An approximate grading scheme will be made available before the week 7 withdrawal deadline.

Homeworks will be distributed via Learn and the course web page, and should be turned in via Learn in the form of a single PDF document. Homeworks will involve writing programs to solve problems, printing out the solutions, and usually plotting graphs of the results. To ensure that all of your work is clearly presented, your solution should begin with a cover page containing the homework number and your name, and a brief description of the material that follows. Your solution to each problem should begin by presenting all of the written answers and numerical solutions requested, followed by graphs, tables, programs, etc., in the order listed on the cover page. All answers should be clearly marked, and all graphs and program listings should be clearly labeled.

If everything in your homework works, I won't need copies of your actual python scripts. However, in case of problems, or if you need debugging help, I will need the scripts themselves, transmitted as additional attachments to the material uploaded to Learn. Please ensure that any scripts you send are complete and run on your computer (or on `newton`, the Physics Department server). A fragment from a Jupyter notebook won't do! I won't spend time trying to recreate a partial script.

Learning Outcomes

On completing this course, students should be able to

- explain the essential mathematical and numerical differences between initial-value and boundary-value problems in computational physics
- write a short program to numerically integrate a function
- present a basic numerical scheme to solve an initial-value problem for an ordinary differential equation
- present a basic numerical scheme to solve an boundary-value problem for an ordinary differential equation
- write a program to solve and analyze the differential equations describing a chaotic nonlinear system
- write a program to solve the coupled equations of motion of a classical N-body system
- write a simple program to solve the time-independent Schrödinger equation in one dimension

Drexel Learning Priorities

- Information Literacy – Possess the skills and knowledge to access, evaluate and use information effectively, competently and creatively.
- Technology Use – Make appropriate use of technologies to communicate, collaborate, solve problems, make decisions, and conduct research, as well as foster creativity and life-long learning.
- Professional Practice – Apply knowledge and skills gained from a program of study to the achievement of goals in a work, clinical, or other professional setting.

Academic Policies

Discussion is strongly encouraged when solving problems, but the work you turn in must be your own. If a friend describes how to solve a problem without specifically writing down any equations, then you may use that information in your own words in your own solution. However, if you directly copy someone else's work, you are committing plagiarism.

You may not copy anyone else's exam, homework, or program. All of these actions are considered cheating and will be dealt with in the following manner. The first infraction will result in a zero for all parties involved. The second infraction will result in an "F" for the course and a report to the Office of Academic Affairs.

Please refer to Drexel's academic dishonesty policy:

http://drexel.edu/provost/policies/academic_dishonesty/

and disability and course-drop policies:

http://drexel.edu/oed/disabilityResources/overview/
http://www.drexel.edu/provost/policies/course_drop/