PHYSICS 177: **Computational Physics**

INSTRUCTOR:	Prof. Flip Tanedo	LECTURE:	TR	5:10 – 6:30pm
CONTACT:	`flip.tanedo@ucr.edu`	ROOM:	Physics	2104
OFFICE:	Physics 3054	FINAL:	Exam/interview	To be confirmed
OFFICE HOUR:	By appointment			

TEXTBOOK (required): *Computational Physics* (ISBN 978-1480145511) by Mark Newman
COURSE WEBPAGE: https://tanedo.github.io/Physics177-2018/
Lecture notes, homework, and additional reading will be posted there.

Official Course Description

4 UNITS[1], LECTURE 3, LABORATORY 3, Prerequisite(s): Prerequisite(s): CS 010 or CS 012 or CS 030; one of the following: PHYS 002C with a grade of B- or better, PHYS 040E with a grade of C- or better, PHYS 041C with a grade of C- or better; or consent of instructor.

Covers computer applications for solving problems in physical sciences. Addresses symbolic manipulation languages such as Mathematica, mathematical operations, plotting, and symbolic and numerical techniques in calculus. Includes numerical methods such as histogramming, the Monte-Carlo method for modeling experiments, statistical analysis, curve fitting, and numerical algorithms.

Remark: we will *not* be using the computer labs. You will work on your own computers using modern tools to collaborate digitally.

Unofficial (Effective) Course Description

If you are taking this class, you already have a solid grasp of 'pen and paper' physics. Modern physics, however, often requires tools beyond 'pen and paper' in two ways:

1. Large data sets may require sophisticated processing and analysis.
2. Intractable calculations, for example nonlinear systems or high-dimensional phase spaces, may require brute force numerical evaluation.

This is a crash course in using computers to do physics. It is neither a computer science course on algorithms nor a data-based lab course, but it is somewhere in between.

Programming Language

All examples and assignments will be in **Python** in the specific context of a **Jupyter** (iPython) notebook. We will use **GitHub** to submit code and **Gitter** to communicate outside of class.

[1]UCR Senate Regulation 760 defines one unit as 3 hours of course work per week. Each week, this class has 3 hours of lecture, 1 hour of discussion, and expects 8 hours of your own time.

Students are expected to be familiar with Python and to familiarize themselves with Jupyter at the beginning of the course: you are free to use any resources or references for writing code[2], but we will not spend much time on reviewing Python or debugging individual code. Instead, we will focus on understanding and implementing algorithms.

Evaluation and course policies

You will be evaluated on the following criteria:

- Weekly homework assignments. (35%)
- Brief in-class assessments. (20%)
- A mid-term oral presentation. (10%)
- Final Exam: (35%)
  - In-class project presentation: Saturday, June 9 at 8:00am.
    or
  - One-on-one interviews during finals week.

Homework assignments may include extra credit components. Presentations and interviews are designed test that you understand and wrote your own code. Final exam arrangements will be decided collectively midway through the quarter.

Physics Hackathon: depending outside variables, we may do a small Physics Hackathon this year. Participation is voluntary but will be counted as extra credit.

Policies

Part of your grade is be based on in-class participation. You responsible for the material that you miss if you are absent. For advance notice and valid emergencies, we can arrange to make up in-class work.

I suggest that you collaborate with other students on assignments. You are responsible for writing up your own homework independently and submitting your own code to GitHub.

You must abide by the UCR academic integrity policies.

Topics

The main topic of this course are: numerical approximations and errors, basic algorithms for integration and differential equations, and random numbers in computational physics. If time permits, we may explore more advanced topics such as machine learning.

[2] This includes online forums like Stack Exchange and other students in the class.