Physics 301
Computational Physics
Fall 2020
Syllabus

Prof: Aldo H. Romero Office: White Hall 207

Class: Tuesday and Thrusday from 10:00-11:15 am Online, Zoom link
https://wvu.zoom.us/j/95472621454?pwd=Y1MwbE1sSDRMS0RXUXBkS0Vwa3oyUT09
Recorded lectures will be available and I will send the link to your email after each class.

Email: alromero@mail.wvu.edu (easiest way to contact me)
Office Hours: Tuesday/Thursday 9:30am-10:00am, 11:30am – 12:30pm
Chat hours: To de defined by you guys!. Maybe some time in the afternoon.

Text: *Computational Physics: Problem Solving with Python*, 3rd edition, by Landau, Páez, and Bordeianu. Free Enlarged eTextbook Python Third Edition) Each student must have his/her own copy of this book.
Notebooks: For the intro to Python, you need to clone the git:
https://github.com/romerogroup/intro_python
and for the intro to Machine Learning, you need to clone:
https://github.com/romerogroup/quickintro_machinelearning

Introduction: This course is designed to introduce you to modern computational methods for physics data analysis and modeling. If you had some computational experience, then you will learn some numerical techniques, otherwise, we will not assume any familiarity with any computer language. The course is intended for upper-level undergraduates. The course was designed for physics students but should also be appropriate for students in mathematics and other physical sciences. Some programming experience will be useful but is not required. The course uses Python and looks at a variety of common problems and Scientific Python libraries. The aim of the course is to teach students how to solve problems and what to look for when searching for tools, rather than to teach all the details of specific tools. Common data manipulation and numerical analysis techniques will be investigated, with a strong focus on visualizing the results.

Grading:

- Weekly problem (or 10 days, depending on topic) set: 45%.

  EACH ONE OF THIS WEEKLY problem should have the following sections:
  (i) Set up of the problem

(ii) demonstration that your solution is correct (by comparing with an analytic prediction or by using provided data or checking with respect to a provided solution
(iii) Discussion of the results. Here you need to address why it fails or how you can improve it.

This should be single repository in GitHub. For each homework you will get a link where you will upload your code, your report and any data that you will use to execute your implementation. Github will record the time of upload and I will be able to see if the homework was hand it in on time - please contact me if you need an extension ***before*** a deadline is passed.

I will use the package Café by Liénardi et al to correct your implementation and compare between the different implementation of your peers. I advise to work alone on the solution of the Homeworks!. The University Rules, including the Student Code of Conduct, are applicable and should be followed in this class. Any violation will be dealt with on an individual basis according to the severity of the misconduct. For example, any material based heavily on outside sources should be attributed in a code comment or a similar manor.

- Individual Term project:
    - Oral presentation: 15%
    - Writeup: 20%
    - Advance written report 5%
- Random Quizzes before class: 15%

Once in a while, you will receive a quiz one day before the class and it is your duty to submit it at midnight before the class. The idea is to allow you to read the topics in advance. The purpose is not to evaluate your knowledge but to test if you are reading the material beforehand. This will allow me to get deeper in the class.

Technology use during/for class

The class will be interactive; registered students have access to HPC computing resources (spruce.hpc.wvu.edu). Students are expected to either use the computers in White Hall 103 (please send me an email if you need it) or their personal devices. Most lectures are designed as notebook that students can follow along with during the class.

Attendance Policy

You are expected to attend each class if you can, and watch the recording if you are not available. All materials should be turned in promptly before the set deadlines - please contact me if you need an extension *before* a deadline is passed.

Conduct

The University Rules, including the Student Code of Conduct, are applicable and should be followed in this class. Any violation will be dealt with on an individual basis according to the severity of the misconduct. For example, any material based heavily on outside sources should be attributed in a code comment or a similar manor.

SCHEDULE (in parenthesis the book sections are given. If it says 2.4, it means all subsections of section 2.4)

Week	**Tuesday**	**Thursday**
1		Introduction to class/Git/shell
2	Introduction to Python (1.4)	Numpy, Matplotlib (1.5)
3	Scipy, Error Accumulation (2.4, 3.1)	Pandas (review: https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html)
4	Random Numbers/MonteCarlo (Chapter 4)/**Project selection**	MonteCarlo (Chapter 4)/Integration (5.7-5.15)
5	Integration/Monte Carlo integration (5.7-5.15,5.17)	Differentiation/ Higher order rules/ error assessment (5.1-5.6)
6	Vectorization/Numpy arrays/Linalg package (chapter 6)	Linear Algebra/Matrix computing (chapter 6)
7	Histograms/ Probability Distributions	Minimizing and fitting (chapter7)
8	Confidence Intervals / Markov Chain Monte Carlo	Markov Chain Monte Carlo Computational Performance
9	Parallel computing (chapter 10)	Intro to ODE (chapter 8)
10	Solving ODE/Fourier series (chapter 12) / **Project written report progress**	Fourier Transforms (chapter 12)
11	Signal Filtering (chapter 12)	Wavelets (chapter 13)
12	Molecular Dynamics (chapter 18)	Fractals/GUI (chapter 16)
13	PDE (chapter 19)	PDE (chapter 19)
14	Introduction to Machine Learning	Machine Learning
15	**Presentation Final Project**	**Presentation Final Project**