Classical Mechanics - Instructor's Master Document

Abudit P. Rai
Dylan R. Smith
Department of Physics and Astronomy, Michigan State University

December 2018

Basics and Week 1

Following Professor Scott Pratt's lecture notes we thought week 1 work would be good for an introduction of how to working with vectors and matrices in Python.https://web.pa.msu.edu/people/pratts/phy321/lectures/lectures.pdf Additionally, we thought it would be a good idea to show basic computational work through the classic cannonball problem seen in introductory physics 1 courses. By having heavily outlined instructions in the Jupyter Notebook that show the students how to write algorithms in Python and then implement them into a function, this should be a fairly easy week for those without any experience.

It should be noted that this Week 1 notebook should also be given to students along with the general "Python Basics" notebook provided, as that notebook gives a better introduction for programming syntax, packages, and techniques. "CM Week1," on the other hand, provides a better introduction for computational work and the implementation of physics concepts into programming.

Most quantative problems or confusions that occure during this week could probably be answered by reviewing the "Python Basics" as it is just a build off of that. However, in terms of conceptual understanding we tried to add as many instructions and examples as we could to guide students in this area.

Week 2

Week 2 work is a continuation of week 1's projectile motion, this time involving air resistance in the 2D canonball problem. We also involved a different form of information storage this time, using arrays to analyze our problem. A short exercise involving the manipulation of arrays was given along with a cell of code dubbed the "checker" that made sure students had the correct outputs. Along with this, we introduced Euler's Method to the students. This is a simple yet effective tool in teaching about computational methods to students, from which we can dive into more complex tools in the future. We provide them with code that already analyzes last week's problem with air resistance using Euler's Method and then ask them to solve the problem again using different starting conditions, along with creating graphs of the analysis.

The purpose of this week is to have students be introduced to and get fimilar with arrays, which will later be used as vectors in computations. Additionially we wanted them to use Euler's Method to compute the position and then compare it to the expected values in order show them how efficient computational methods can be, along with getting their interests for future methods and analysis. The "checker" was also provided to students this week to get them thinking about tests they can perform on their code to make sure it works properly, along with identifying where possible problems could occur. We hope to later expand on concept of testing one's code to teach students how to efficiently find problems in future programs.

We don't think it will be uncommon for there to be many students, especially those with no programing background, to find Week 2's work to be a sizable step above Week 1 and the Python Basics notebook. As this week does involve the concept of using an array to store data, the manipulation of such data types, and the first computational method that many will see. Confusion might arise from understanding of how the algorithm is updating, which is why we so heavily "hand hold" in this week's lab notebook. But as the notebooks progress we will start only giving the outline of what we want with no example code given, at most a skeleton.

Week 3

This week we will build upon what we have done the previous two weeks, and hopefully the Python Basics notebook if the instructor choose to utilize that. We look at how Euler's method is lacking in some areas and propose the usage of Verlet Velocity. This is meant to make students think about how there are different implementations of computational methods out there, hopefully sparking some curiosity as to what is possible. We also want students to start feeling comfortable modifying code to their leasure, so we insert a "momentum calculator" inside one of the functions and comment how we did it.

Hopefully by reading through this notebook the sudents gain an idea of how they can translate algorithms from words and symbols to commands in a language. It is a very algorithm heavy week, but this was inevitable and will get them thinking about how such things can be utilized. By building functions the students will also get experience with variable manipulation and proper syntax.

Week 4

Funcitons are a very important part of computational work in this day an age, which is why we thought it would be a good idea to give them practice using it this week. Analyzing harmonic oscillators would be good practice for such concepts too, as their equations give students the chance to play with variables in the function and see how movement is affected. Sympy, a simple but powerful package in Python, is used to set up equation and give experience with funcitons.

From letctures and homework, the students should already know the equations for Underdamped, critical damped, and overdamped harmonic oscillators. So the challenge this week will be creating and debugging code with Sympy in setting up graphs. Before, matplotlib.pyplot was used to graph this, but by using Sympy we hope to get students thinking about different ways in which data can be visualized.

Week 5

We wanted to give students more practice using Sympy to emphasis the usefulness of imported packages and funcitons. Since they will be diving more in-depth into potential around this time, we used the idea of analyzing potential as a funciton of position to create graphs. More questions are being asked this week to make sudents reflect on what they are doing and the value of computational tools in Physics.

Unlike before, we don't give students any examples in how to plot in 3D this week. This is because finding documentation and discussions on 3D Sympy plotting is easy to find on the internet, so we wanted to give students experience trouble shooting their own problems in coding. We encouraged students to do this in the instructions, as it'll push them to begin debugging and coding more independently.

Week 6

Around this time many of the student will be busy with midterms and projects, so we thought it would be a good idea to take things a little easier. To give a little more experience with creating their own functons again, we introduced Simpson's Rule, but only asked students to analyze the code for it. To keep things interesting, we wanted to show why numerical methods of analysis is important despite the existance of Calculus.

Questions in this week are rather simple, meant to encourage students to learn to read code and see how different variables in a funciton get manipulated. For the more experienced coders this week will be easy, but for the novices (which we assume most of the students will be) this will be good practice for code reading.

Week 7

After a few weeks of learning to make functions, use them, and some experience with numerical analysis, we want to combine these aspects through a planetary simulation. However, this time we will be using a three-body problem.

Lagrangians

This workbook highlights the simple pendulum using Lagrangian mechanics. It is presumed that this notebook would be worked through near the tail end of the semester, so knowledge about basic Python syntax and documentation is presumed. This assignment is split into essentially two parts: the first part deals with using sympy to analytically find the differential equation describing a pendulum's motion using the Euler-Lagrange equation, and the second part involves numerically solving the equation of motion using the Euler-Cromer method of integration.