Semester project instructions

PHY-905-005
Computational Astrophysics and Astrostatistics
Spring 2023

Overall plan: The goals of the semester project in this course are: (1) for you to *broaden and/or deepen* your understanding of the numerical techniques you have learned in this class, (2) to apply this knowledge to your own research or another topic of your choice in astrophysics or physics through software that you have written to either **analyze a dataset or model a physical situation**; and (3) to share this knowledge with the rest of the class via a presentation and code demonstration. You will demonstrate achievement of these goals through several deliverables, as detailed below. Note that most deadlines are given as dates, with the time of the deadline being 11:59 p.m. that day unless otherwise specified.

In terms of project size and complexity, this is meant to be doable in the roughly six week time frame and take roughly as much time as two homework assignments (roughly 20 total hours of work). To that end, it may be very difficult to get something working that directly ties to your research – in that case, you should choose a project that is a simplified, constrained, and/or idealized version of what you'd actually need for your research!

Also, please note that I expect you to put all of your code, analysis scripts, data files, and so on in the directory `source_code_and_other_files` in this Git repository, and regularly commit changes to the repository and push them to GitHub (in other words, use this as the working directory for your project). This serves three key functions: (i) it allows you to keep track of your code changes, and revert those changes as needed; (ii) it provides a critical backup of your project and data in case of hardware failure, and (iii) it allows me to keep up on your progress, and follow up as necessary. So, please commit and push your changes early and often! Note also that your final code must (1) be properly formatted and commented according to the class coding standard, and (2) must pass a `Pylint` check with no significant warnings or errors.

Project components and deadlines, in order of due date:

1. **Project proposal, due before class on Thursday 3/24/2023**. Write a brief ($\simeq 1-2$ paragraph) proposal describing the project that you would like to pursue. Briefly explain (i) the scientific motivation for the project, (ii) how it relates to your research and/or your scientific interests, (iii) the numerical methods that you will learn about and implement for this project and how this goes beyond what we've learned in class in either breadth or depth, and (iv) your intended deliverables in terms of data analysis products or model outputs. Turn this in via a text or markdown file in the `proposal` subdirectory in this repository. In class the next day we'll have a brief roundtable for everybody to explain their proposed project, and I will also follow up with written feedback (and may ask you to change your proposal).

2. **Project update #1, due Wednesday 4/5/2023**. The goal for this update is for you to have done the necessary background reading and started implementation of the code you need for your project, but not necessarily have a fully-working code (though what you do have should be in `source_code_and_other_files` and committed to the repository). The written update go in the subdirectory `update_1` and should include an update on your proposed project that details (i) the numerical methods you're using and (if relevant) the sources you've used to learn about them, (ii) your progress thus far, and (iii) a discussion of any unexpected challenges you've run into and the steps you've taken to deal with these challenges. Note: if your project is going poorly this is a good time to re-evaluate your project and revise your plans, which may include reducing its scope! We will also have a brief roundtable in class the following day for everybody to give a quick project update, highlighting numerical methods and challenges.

3. **Project update #2, due Monday 4/17/2023**. At this point, your code should be close to complete in terms of implementation of its core functionality, and you should be able to verify that it behaves as expected using simple datasets (if you've created an analysis tool) or a simple simulation setup (if you've written a model of some sort). Note that these simple setups may be useful as tests! As

with the previous update, your code should be in the `source_code_and_other_files` directory and your written update goes in the subdirectory `update_2`. The written update should describe (i) your progress thus far, (ii) what the tool/model is intended to do and why you believe that it is behaving as expected based on your simpl dataset/model setup, and (iii) what remains to be done before your final submission. **By this point, your code should also adhere to the class coding standard and pass checks with `Pylint`.** We'll have a brief set of updates in class the next day.

4. **Project presentations and code demonstrations, due 10:00 a.m. on Wednesday 5/3/2023 (the beginning of the course's final exam session)**. Your final presentation can use any of the standard presentation file formats (Powerpoint, Keynote, or PDF), and should be put in the `presentation_demo` subdirectory. Presentations will be a total of 15 minutes long, including 10 minutes for your talk and 5 minutes for questions and discussion. Make sure that your presentation includes the scientific justification for your project and your goals (i.e., what your code is supposed to accomplish), the numerical methods used (including a brief explanation of how the new methods you have learned about work!), the results that you obtained, and any lessons that you learned about the methods or implementation that you think would be interesting to your colleagues. If the project is an idealized, reduced, etc. version of something you'd do for your research, please also explain that! You should also be prepared to give a brief demonstration of your code, if necessary. Your presentation will be evaluated on the clarity of your visual aids as well as the quality, clarity, and delivery of your presentation. Please note that talk slides are due via GitHub immediately before we start presentation.

5. **Project final code deadline, due Friday 5/5/2023**. This is the final deadline for your code, all of which should go in the `final_code` subdirectory.

   Your code should: (1) be properly formatted and commented according to the class coding standard, and (2) must pass a `Pylint` check with no significant warnings or errors. Comments at the top of the main source file must describe how to run the code, including a listing of any additional Python packages that must be installed (on top of a baseline Anaconda distribution running the most recent version of Python 3) in order to get your code to work. Your code will be assessed on your implementation of the chosen algorithms, the results that it produces, code clarity/modularity/comments, and whether it runs on my Mac laptop and on the MSU supercomputer when using the aforementioned Anaconda distribution. In addition to your code, please include any data files that are required, as well as scripts used to generate plots or do additional analysis. (If the datasets are larger than $\simeq$ 10 megabytes, please talk to me before committing them to the repository - there are other, better solutions!)

NOTE: I strongly encourage you to come and talk to me at all stages of this project – I'm happy to brainstorm ideas, give suggestions, and look at paper drafts and your code. Please use my office hours, and if you can't make it to those contact me via MatterMost to schedule an appointment at another time!