UROP 2019/20

Project: Numerically Simulating New Exoplanet Systems

Mentor: Dr. Juliette Becker

My limitations

I do want to point out (again) that I will be traveling a decent amount, and so our in-person meetings will be limited to when I am at UM. My current travel schedule for this term is:
9/23-9/30 UM
10/1-10/18 Caltech, Georgia Tech, Missouri
10/19 - 11/8 UM
11/11 - 12/1 Caltech, Seattle
12/2 - 12/20 UM

We can meet virtually and I will be able to continue to comment on your lab notebooks (see later in this document for more details on that), but PLEASE think about whether this schedule is OK with you! For many people, it is better to be able to meet your mentor in person frequently, and while I will be happy to meet up in person whenever I am at UM, my travel schedule may not work for everyone!

I would like to meet with everyone before I leave for my October travel, ideally (so, sometime the week of Sept 23 - 27th), so that we can get the software set up and answer any other initial questions.

Time commitment

The current suggested hours for this project are 6-8 / week. If you would like to do more than this (for work study reasons), let me know, and I can try to see if the UROP program will allow an upgrade in hours. The majority of your time will be spent working independently. We will have one one on one meeting per week, which can be either in person or via Skype (or some other video conferencing tool - I prefer Google Hangouts) as needed. We will also have group meetings with everyone working on TESS data periodically: once everyone officially signs on for the project, we will schedule the first of these meetings at a time that works for everyone.

Activity	**Time**	**Format**
One on one meeting	We will choose a weekly time that generally works, and reschedule as needed (between 30 min - 1 hour per week)	In person (in West Hall) when possible, online when either of us cannot make it in person
Group meetings (combined with John	We will choose a time during the week that works for	Group meetings with ~5 students, me, and John, on

Monnier's students)	everyone, and hold a meeting every ~2 weeks (between 1 and 2 hours per week in weeks we hold the meeting)	varying topics.
Self-directed work on project	Between 3 - 7 hours a week.	You can choose where and when you want to work - just make sure to keep track of your progress in your GoogleDoc lab notebook!

Keeping track of your work
One really important part of research is keeping track of what you try, including what fails and what succeeds. In many wet lab settings (such as biology or chemistry labs), you might do this in a paper lab notebook. For this project, I will ask you to keep a virtual lab notebook in Google Doc form (and, importantly, share the document with me!). Each day you work on the project, make an entry in your document describing what you did and what the results were. Some days, the entry might look like "September 5: trying to install python, but am having trouble. I keep getting the following error message: [copy&paste of the error message]", while other days you might put a plot you made in the document and describe what you did to make the plot.

This document is intended to both be helpful to you (you will be amazed how useful it is to be able to go back and see what you did earlier in the project!), but also allow me to give you real-time feedback and suggestions via comments in the GoogleDoc. If you are on work study, this also provides an easy documentation of your hours and when you worked!

Projects within our group
We will have multiple students working towards similar goals (but not exactly the same goal). You will all start out learning Python, GitHub, and how to make plots, so you may find it useful to work together and share your code with each other. Or, you may find that you would rather work alone. I only ask that you all share contact information with each other - beyond that, it's up to you if you would like to work with others or not.

Hosting code
I will ask you to use GitHub to store your code (you'll need to sign up for an account - GitHub is free, so sign up for the free student account). I suggest installing the GitHub desktop client, choosing one folder on your computer to contain all your UROP materials, and syncing that folder to GitHub. You can choose whether you would rather make your code public or private to start with - I would suggest private at first, as you work on getting the basics working. I will ask you to add me as a collaborator on your folder, which means I will be able to add documents

and make changes to your code as needed. We will talk more about exactly how to use GitHub once we start working together!

Research Tasks

Since a large part of this UROP project is coding, our standard workflow will be me giving you a coding task, with varying amounts of direction (at the start, I will give you a Jupyter notebook with parts you should fill in, and later on I will give you no template code at all!). I will host these instructions (and you should host your work on them) on GitHub.

Requirements for the project

Since this is a coding project, you will need to have either your own computer, or access to a computer that you can run Python and Jupyter on. Let me know if this is going to be a problem, and we can figure something out!

Project Steps

Here is an approximate ordered list of the things you will do during this UROP. You might want to start part way down the list or skip some bits, depending on how much you already, and you may not get to the end of the list by the time the UROP ends - any of which are fine! I'll give you more details on each when we get there.

1. Install needed software (the GitHub desktop client, Python, Jupyter, and Pythn packages including pandas, numpy, scipy, matplotlib, rebound - many of those should come pre-installed) on your computer
2. Learn the basics of Python (in the Jupyter environment) - make an array, plot, write to a file, etc
3. In Python, read in data and make a plot
4. Learn how N-body simulations work
5. Make your own N-body simulation, and make some plots (this may take a while)
6. Switch over to Rebound and learn to use it (this may take a while too)
7. Simulate some existing systems, making plots / movies / etc
8. Learn to handle output from large number of simulations
9. Apply your knowledge to a real exoplanet system! (Details of this stage TBD).