1 Assignment Four

2 Aspects of the Project Done Well

Our group reviewed the Quasists project and overall the code ran without errors and they completed the tasks of the assignment. Our group was able to run the code with the help of the and reproduce the plots in the report. We did not include all the plots from all the different mass cuts they included. But the select ones that we did recreate can be found in test_results folder. In addition, the format of the code is very readable and well structured. They also added stellar evolution to the simulation which was not required by the project but makes the simulation more realistic. The program also prints updates (percent completed) on the progress it has made which is nice because it does run for a long time. It would be better if they gave a estimation of the time remaining instead of a percent completed.

3 Suggested Changes

Splitting out the simulation code from the plotting code. This is to make each function focused on a single task, making it easier to maintain the code and understand.

3.1 Report

It might make it easier to understand the results if they were on the same plot. For example if combined the relative energy plots for a the different mass cuts might make it easier to compare them.

Address in the report and try reversing the which particles are integrated by the tree code and which particles are integrated by the direct code.

Stellar evolution is included in the code cluster.py but the results from the implementation are not discussed in the report. What did you find out. How does stellar evolution being on or off affect the results?

It also might make the report more readable to have the plots embedded in the text that talks about them instead of having them all at the end of the of the paper.

4 Changes Implemented

Changed **import matplotlib.pyplot as plt**, **import time as whatever**, and **import numpy as np** to **import matplotlib.pyplot**, **import time**, and **im-**

port numpy so that it is clear what package is being used.

Add setting random seed for reproducability otherwise the exact distribution of masses and locations changes every run, even with the same mass function and all other variables, resulting in not exactly the same initial conditions.

Made docstrings from the comments on the functions. This makes each function contain the comments describing what they do and the inputs, making each function more self-contained. Also removed commented out code, and variables that were assigned but not used, such as the heavy and light particles at the beginning of the while loop in the hybrid gravity function.

Since the code also simulates stellar evolution, it is much slower than just the nbody simulation. To speed up the running of the code in general, we added **the number of workers per code as a command line argument**. This allows for much faster runtimes if there are sufficient CPU cores for the code to run on.

Added exceptions for invalid options in the majority of the code. With exceptions being raised, the code now gives slightly more informative error messages.

We fixed the section of the code that produces an animation of the cluster by making it it's own function **create_animation()**.To get it running, we had to change the animation code so that it updates the positions of the points every frame, instead of clearing and replotting the points. We also changed the output to MP4 to make it work, and added axis labels. It doesn't show the animation but creates animation.mp4 in the test_results folder.

We added testing in the form of nosetests. The tests cover mostly the cluster generating function, focusing on verifying that the generated sets of particles fulfill the input parameters, as well as checking faulty input to make sure the correct error codes are returned. There are also tests for the half mass radius calculation and the dynamical timescale functions as well. These tests are designed to verify the functionality in the code. No tests were created for the output from the hybrid_gravity functon, as the tests would take a long time to finish, and numerical differences in the simulations could mean that the tests sometimes pass and sometimes do not.