[0] Rebound: When galaxies collide - By Ben Ackeman - 5/17/2025

[1] For this project we used the Rebound package. The package integrates the notion of particles under the influence of gravity. These “particles” can be anything from stars to dust.

[2] We selected this package because our original plan was to use the pNbody package, but due to errors in the download process we needed to pivot. We wanted to still do a pNbody type of simulation so we searched and found rebound.

[3] Rebound was made in 2011 making it 14 years old. After rebound the contributors also made Reboundx for adding migration forces, GR effects and spin to Rebound simulations. For this project I have Version 4.4.8 installed.

[4] The package is still maintained and updated, but it is an open source package so the contributors are many and varied but the creators do still appear to be actively updating the code. I cannot find instructions on how to contribute to the code, but I assume it's done via forking and pushing the git repository.

[5,6] It is extremely easy to install, all you need to do is pip install and it works perfectly.

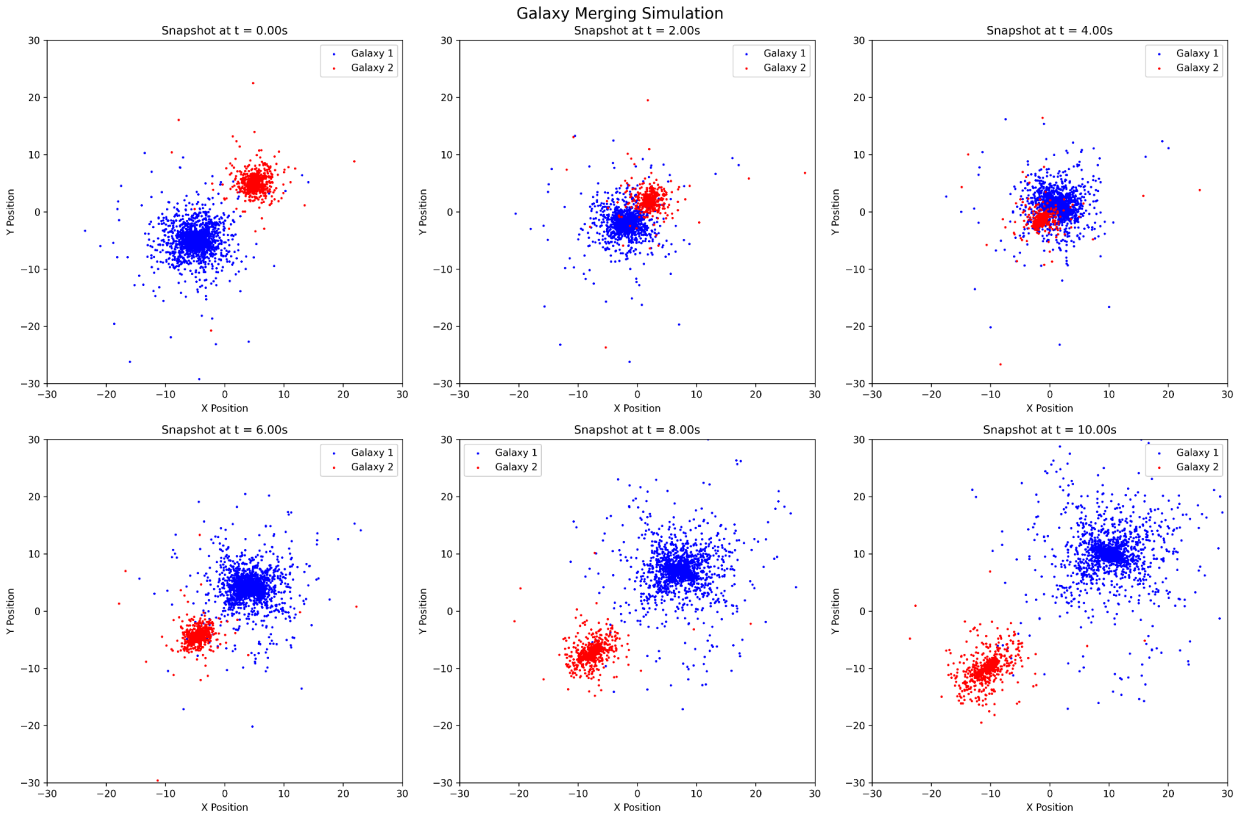
[7] The source code is available for download in the git repository for the package.

[8] Rebound is used in Posidonius: N-Body simulator for planetary and/or binary systems and ASSIST: Solar system test particles trajectories integrator. Cited below

[9] The code can be used in python, python notebooks, c, and even through a link on the web listed in the download instructions section of the read me page. You can only run premade examples built with webassembly and C

[10,12] The code is in a Jupyter notebook. In this note book the Rebound library is used to run an N-Body gravitational simulation by making two clusters of particles arranged in a plummer sphere. The simulation is integrated over time and records particle position and energies at each time step. Then the spheres and their interactions are plotted over the course of 6 frames.

Rebound can build its own 3-D Open GL Visualization, but for this figure matplotlib was used.



This figure shows a series of snapshots from a simulation of galaxies merging over 10 units of time. Each subplot displays the form of the two galaxies (Galaxy 1 in blue and Galaxy 2 in red) at different times. The galaxies initially start separated, move toward each other, and then interact dynamically, showing tidal distortions and mixing as they merge.

[13] The Package was built in both Python and C.

[14,15]The package does not need much to run effectively. It only needs parameters to work, things like particle masses, coordinates, the integrator to be used and its settings. The package can have multiple types of outputs ranging from a few outputs, datasets, or 3-D simulation figures

[16,17]On the Git you can find a host of different unit tests given by the creators for many of the main functions of Rebound. There is also a specific test_units.py among them. We rely on these tests to make sure the code is working and accurate. Also the simulation showed the tidal distortions of the galaxy one might expect in this context.

[18] In the Git there is a requirements document that discusses what the code needs which includes only Numpy and Matplotlib. However, upon looking through the source code I found there is also a very heavy requirement on Python's math package.

[19] The Rebound package provides not only in depth explanation and examples of all of Rebound's capabilities but also has links to youtube video tutorials on their website. In the git, there is also a host of different larger and in depth example projects that use various aspects of Rebound.

[20] If you were to use this code in a paper they would like you to use the code

```
sim = rebound.Simulation()
-your setup-
sim.cite()
```

To generate the materials needed to cite the package, or the Rebound citation bot

[21]

1. Rebound git:https://github.com/hannorein/rebound/tree/main
2. Pypi:https://pypi.org/project/rebound/
3. Rebound Website: https://rebound.readthedocs.io/en/latest/
4. ASCL: https://ascl.net/code/search/rebound

[22]

1. In, The dynamical orbital stability of the proposed 2+1+1 hierarchical eclipsing binary systems, Rebound is used to model a star cluster's dynamics. Rebounds integrators are used to accurately model long term orbital evolution and interactions of stars, allowing us to view cluster stability and energy exchanges in the system.
2. In, A highly resonant Neptunian region: A systematic search for two-body and three-body mean-motion resonances, Rebound is used to model the orbital evolution of small system bodies under gravitational perturbations. The code integrates the complexities over time and incorporates the forces of gravity between the planets to analyze long term stability.

[23] Aside from package specific functions, Phys265 completely prepared me to use this code.

[24I collaborated with Pete Micciche and neither of us have any prior experience with the Rebound package.