# The REBOUND Package Report

[0] REBOUND: “Because even planets need a little push sometimes”
Roach Giancarlo - May 18, 2025

[1] REBOUND is an open-source N-body simulation package designed to model the gravitational interactions between particles, such as planets, stars, or asteroids, over time. It allows researchers to simulate complex orbital dynamics with high precision using various integrators like IAS15, WHFast, and others. The package is particularly useful for studying celestial mechanics, planetary system formation, orbital resonances, and long-term stability. Its python interface makes it accessible and flexible for customizing physical forces, initial conditions, and output analysis.

[2] I picked this package after failing to successfully get AMUSE to work on my desktop computer. This package is C at its core but it has full Python API and is rather beginner-friendly as this semester was my first time learning Python. It also has built-in matplotlib support which is a tool we have been using this entire semester which also helped when trying to find example codes in Python.

[3, 3.5] REBOUND was released in 2012 and developed by Hanno Rein at the Institute for Advanced Study at Princeton.  It is being maintained by Mr.Rein and other developers. REBOUND has a very long line of gravitational N-body simulation codes, prior to its release NBODY6 or Aarseth was very famous for its N-body simulations and is widely used for stellar cluster research. Other notable codes include GADGET which is utilized for cosmological simulations. After REBOUND REBOUNDx was released which is an extension of its former but added additional physics like tides and general relativity. REBOUND isn't the sole competitor in the field, other similar codes that solve similar tasks include PKDGRAV and none other than AMUSE.

REBOUND’s version file was labeled (4.4.8)

[4] Regarding the installation it was very easy compared to AMUSE. Since I was using python and pip.REBOUND has a python wrapper it made the entire process very simple. I downloaded Ubuntu and did the following. I also installed jupyterlab matplotlib and then imported REBOUND:
Python3 -m venv rebound-env # Created the environment
Source rebound-env/bin/activate
Pip install rebound #Installation of REBOUND via pip
Pip install jupyterlab matplotlib
Import rebound

[5] As stated before can be installed via pip install command

[6] Code is open source and available on github

[7] REBOUND is utilized in other popular codes such as REBOUNDx and AMUSE. REBOUNDx is widely used in planetary dynamics research and was built as an extension of REBOUND. AMUSE while being a python framework that integrates multiple astrophysics codes for stellar evolution etc. REBOUND is used as a gravity solver within AMUSE, furthermore allowing REBOUND to be used in multiple physics simulations with other codes.

[8] It is mostly utilized in python scripts for automated simulations along with integration with other tools. Moreover, it can also be used in Jupyter Notebooks along with command-line.

[9] Here is a figure from my example where I simulated a planet orbiting a star using an integrator(ias15) which tracks the planet's position over time and plotted it over a 2d space.

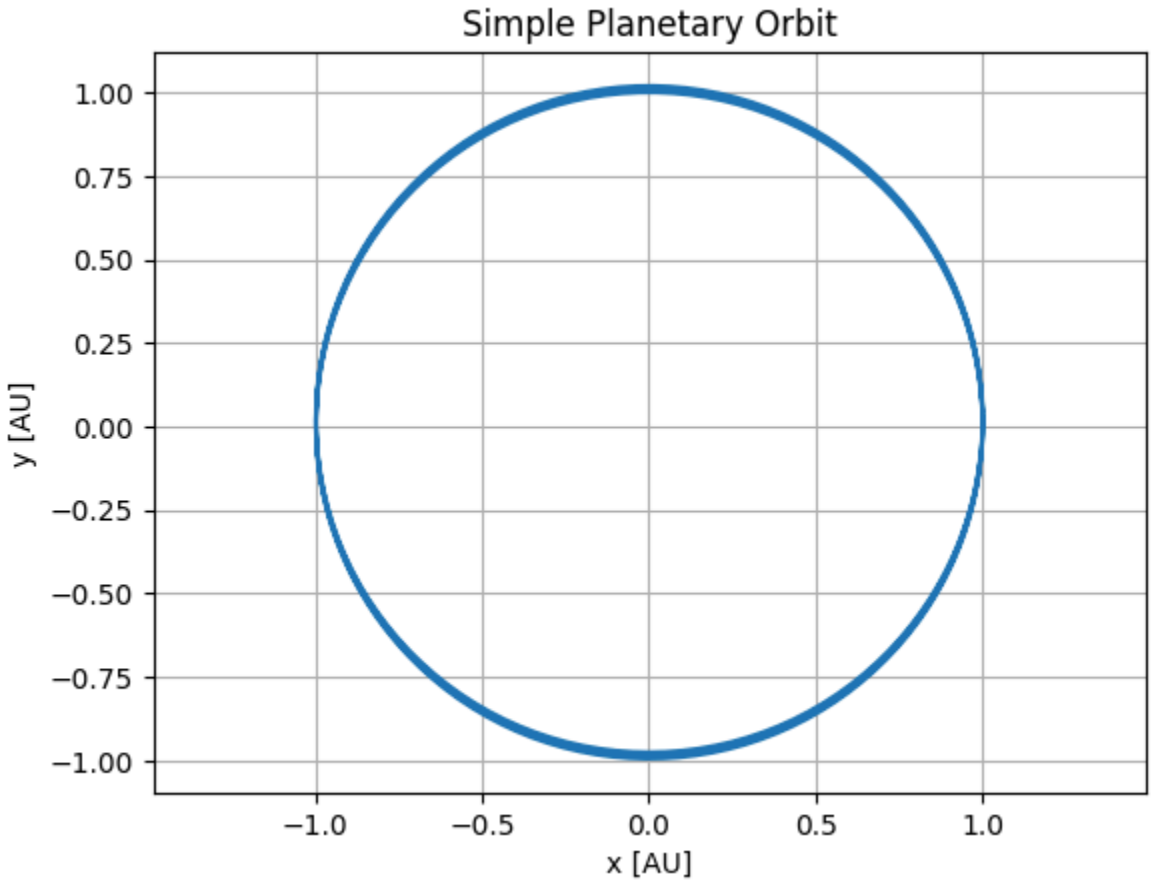


Fig 1.1

[10] While REBOUND itself does not have built-in plotting for figures.You are on your own but, REBOUND was designed to work seamlessly with Python plotting libraries such as matplotlib.
[11] In these two figures (2.1, 2.2) below you can see the before and after of particle spatial distribution on a planetary disk like Saturn. This simulation allows you to study clumping of these particles along with examining the angular momentum as they travel and form self-gravity wakes

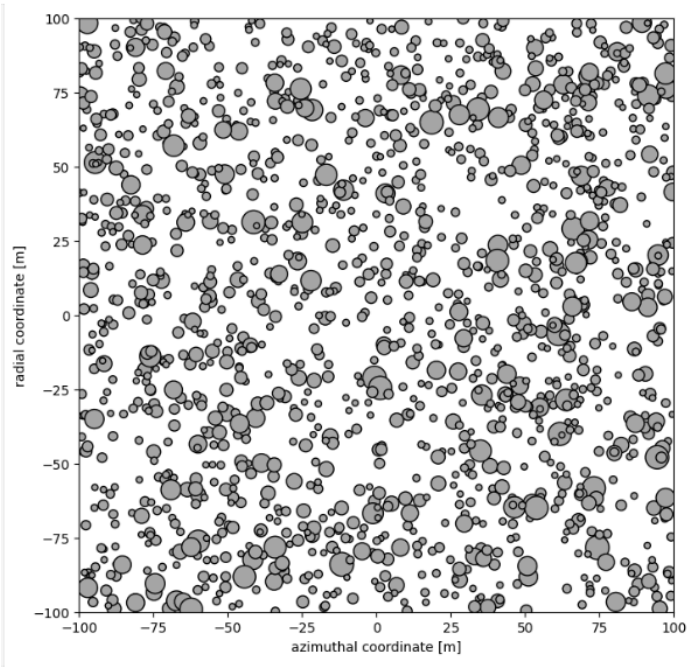


Fig 2.1 (Before)

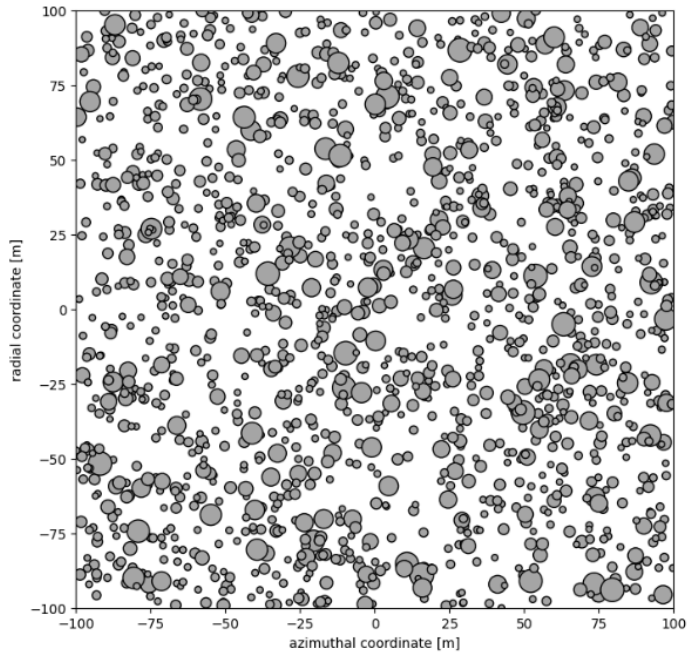


Fig 2.2(After)

[12] REBOUND is not purely python, the core is written in C to handle the physics aspect but it uses a python interface that wraps the C code. No Fortran is used in the package.

[13] The input is mostly generated from scratch as the software was designed to build simulations from the ground up and setting your own parameters. REBOUND does not directly import datasets like .csv files but, you can utilize pandas and numpy to read datasets and convert values using the sim.add() command.

[14] Regarding the output, you can produce snapshots and In-memory access.

[15] If you download REBOUND from the Github repository you can run benchmarks, regression tests and unit tests. These tools ensure numeral results are consistent across versions along with helping compare performance between integrators, collision algorithms

[16] I can be confident that the code produces reliable results by running the benchmarks and tests stated before as a cross-reference to my results from jupyterlab notebook

[17] The main packages REBOUND uses are numpy, cython, and matplotlib. I found this out by running the following command: pip show rebound

[18] REBOUND provides extensive documentation. From many online API documents it also has released many Github README/examples along with in-code docstrings.

[19,20] References

- REBOUND preferred citation: https://ui.adsabs.harvard.edu/abs/2012A%26A...537A.128R/abstract
- IAS15 : https://ui.adsabs.harvard.edu/abs/2015MNRAS.446.1424R/abstract
- WHFAST : https://academic.oup.com/mnras/article/452/1/376/1748797?login=false
- AMUSE: https://ui.adsabs.harvard.edu/abs/2013CoPhC.184..456P/abstract
- REBOUNDx: https://ui.adsabs.harvard.edu/abs/2020MNRAS.491.2885T/abstract

[21] ADS citations for ASCL based code:

- https://ui.adsabs.harvard.edu/abs/2022MNRAS.511.1848C/abstract
- https://ui.adsabs.harvard.edu/abs/2021ascl.soft04031B/abstract

[22] This course provided me with the knowledge and tools necessary to get me through this final project.

[23] I did not have any prior experience with this software package and I did not work with anyone from this course on this project. All 25 prompts have been answered.