Georgia Institute of Technology

School of Physics

Project Proposal
PHYS 3266

Simulating Orbital Perturbations and Inferring their Sources

Written By:
-Joshua Brandt-
-Paul Vollrath-
-Chloe Fair-

Date: February 22, 2022

1 Research Topic & Question

Question: Given a deviation from a planetary Keplerian orbital trajectory, can we determine properties of an intervening planet?
The goal of our project is to create a computational simulation of a method that was used to predict the location and properties of undiscovered planets - which ultimatley led to the predictions and discovieries and Neptune and Pluto, and continue to create wonder about more potential planets in the Solar System. Our goal is to take in data on the orbits of a set of planets and use and N-body simulator to guess where a planet would need to be to create observed orbital perturbations.

2 Simulation Description

We are analysing our system by way of an N-Body program, which is a tool that allows for simulations of multiple bodies interacting according to physical laws and specified boundary conditions. In this case, we are using an N-Body simulator to model the orbits of planets around the Sun in our Solar System.

In general, you cannot solve analytically the motion of "N bodies" interacting under gravity. The fundamental structure of the N-Body program is that of an interative process, which uses the laws of gravitation at a certain moment in time, and interpolates the current state to the next iteration. Using this method, we can determine the orbital trajectory of a system of planets interacting under gravity.

We will begin by taking in data on the orbital trajectories of some system of planets. Then, we will (at first, without much evidence) propose a potential planet that might cause observed orbital perturbations, run the N-Body simulator on the system, and determine how well the planet's orbits fit the data. Using this comparison, we can propose deviations in the proposed planet's properties, making it more massive or further away, and re-run the simulation. We see this as an automatic process, a system of educated "guess and check" until we find agreement between our simulation and the data, and can conclude what type of object could cause observed orbits.

3 Required Input Physics

The backbone of our project is Newton's Law of Gravity. In particular, to determine the net force acting on each object, we evaluate

$$\vec{F} = GM \sum_j \frac{m_j}{r_j^2} \hat{r_j}$$

In our current code, we assume the acceleration of each body is constant over the time step Δt, so we begin each iteration by updating its acceleration $\vec{a}$ and computing a $\vec{a}\Delta t$ to add to the current velocity $\vec{v}$. This is a reasonable assumption to make given the small time step, and relatively large astronomical distances, as well as the fact that while in modeling the solar system the separation between planets may vary wildly, the primary contributor to the net force on each object is the Sun, whose distance to an individual object does not vary noticeably over a small time interval. We then compute $\Delta\vec{r} = \vec{v}\Delta t$ and add it to the object's current position. This way, to first order, we keep track of the approximate motion under gravity.

Another mathematical problem we will need to address is how to quantify how similar two orbits are. If we want to determine how much more likley one proposed planet is than another, we need to measure how different the orbits caused by those two are different from the data. As of now, we haven't given this much consideration.

4 Computational Methods

Our current computational methods are rudimentary in execution. We plan to integrate the FTCS method (Ch. 9.3) or other differential equation methods that would be applicable. Since an N-body problem has not only no closed form function, but also is largley chaotic, the integration methods we covered in class will not be appropriate to solving our problem. Our main goal is to find a method that leverages the most information to give accurate results of where planets move under gravitational influence, a process governed by a second order differential equation.

The current implementation of our code already includes a variety of computational methods that make running the simulations seamless. Each body that participates in a simulation is established in a JSON file containing information about the object, which our code reads in and converts to a "Body" Python object, whose attributes include mass, position, velocity etc, and methods include updating position, or finding acceleration. We automatically fill-in information we can pull from Astropy for accuracy and ease. Another module contains relevant physical equations, and yet another has the code to run the N-body simulator, which calculates the net force on each body, looping over subsequent iterations until some stopping condition is reached. Simulation parameters like the stopping condition and time-steps are passed through another JSON file which our code parsses before initalizing the N-body module.

An important method to be efficient in our project will be deciding what new planets to propose. Given that one planet does not give an answer close to the data, how should we decide to make the next proposition of what planet to inject next? There is probably a computational method hidden in this question, but so far we have not come up with a solution.

5 Simulation Set-up

6 Quantities to Inspect