PHYS486-FINAL: The N-Body Problem

Nicholas Cemenenkoff

June 9, 2018

Abstract

We construct a 3D model that simulates N gravitationally interacting point masses. The model travels through time via several different numerical integration methods ranging from very basic to state of the art techniques. To clarify, at each time step, each body m_i calculates the forces it feels from the other $N-1$ bodies in mutual orbit, so the computational demand of this problem is on the order of N^2. Because no analytical solution exists for this problem, we calibrate our model to represent a system very close to our solar system, but with initial conditions chosen to produce non-coplanar orbits. The results explored in this paper are mysterious and beautiful, and the orbital trajectories can take on many twisting paths with far greater variety than simply circles, ellipses, or hyperbolas. For systems with only four or five bodies, the orbits (especially when calculated with leapfrog integration methods) are extremely stable and retrace the same trajectories over and over again. For systems with nine or ten bodies, the orbits seems relatively stable, but if propagated far enough through time, show a sort of smearing effect representative of the changing center of mass of the system or perhaps entrance into a chaotic regime. Because the N-body problem is quite computationally expensive, to ensure reasonable computation times, $N = 10$ is the maximum number of bodies explored, but if given the appropriate amount of initial conditions, the code can truly accommodate N bodies. We encourage readers to open the actual Python document associated with this paper to then explore their own N-body simulations.

1 Problem Statement

Imagine a vast empty three dimensional space. Initialize some point masses of varying size at different locations and with different initial velocities. Accounting only for the gravitational force between the bodies (and not allowing for collisions), what sort of orbital trajectories will result? This problem is classically known as the N-body problem, and was first seriously studied by Sir Isaac Newton. In this paper, we implement a numerical solution to the problem and explore the following questions:

- What are the core characteristics of the N-body problem?
- As the number of bodies increases, does orbital stability seem more or less likely?
- Which factors seem to play a critical role in establishing stable orbits?
- How can the accuracy of the simulation be checked?
- Which numerical stepping methods produce the most accurate orbital trajectories?
- Which stepping methods are the fastest, which methods are the slowest, which are symplectic, and which conserve energy?

2 Numerical Solution

At its core, the numerical solution to this problem involves solving for the instantaneous accelerations for a given set of position data for a set of point masses given by

$$\boldsymbol{a}_{ij} = \frac{Gm_i\left(\boldsymbol{r}_j - \boldsymbol{r}_i\right)}{\left\|\boldsymbol{r}_j - \boldsymbol{r}_i\right\|^3}.$$

There are several stepper methods that calculate the instantaneous accelerations for every body, but they do so in different ways. The employed methods are:

1. Euler
2. Euler-Cromer
3. 2nd Order Runge-Kutta
4. 4th Order Runge-Kutta
5. Velocity Verlet

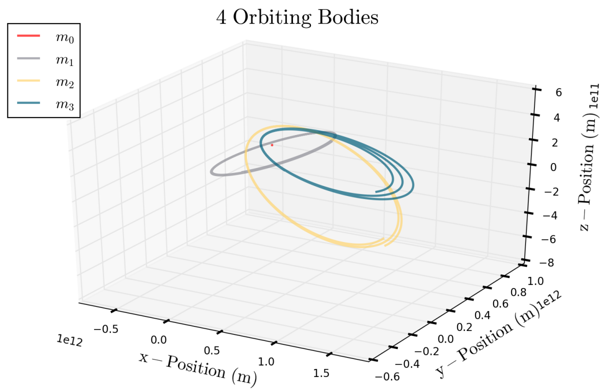

Figure 1: caption

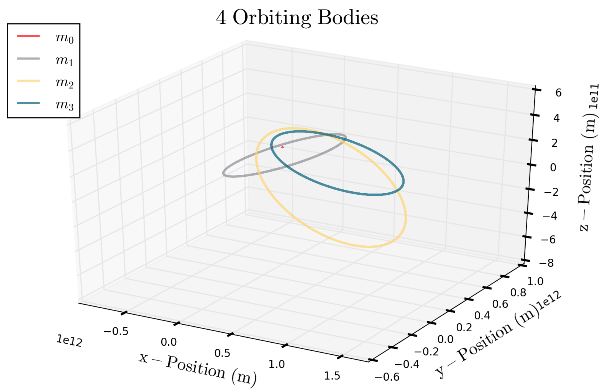

Figure 2: caption

6. Position Verlet
7. Velocity Extended Forest-Ruth-like
8. Position Extended Forest-Ruth-like

The Euler method is the only method on the list that does not conserve energy. This can be seen in Figure 1. The orbital trajectories for four orbiting bodies spiral inward with this method type, but are repeating ellipses for all other methods.

The Verlet and Extended Forest-Ruth-like methods not only conserve energy, but are also symplectic (area preserving). Interestingly, with $N = 4$, all of these methods produce generally similar results for initial conditions that are somewhat close to our own solar system (stable ellipses), but wild, twisty, *differing* results for more chaotic initial conditions (still with $N = 4$). We spare the reader the details of each specific algorithmic implementation, but note importantly that they all functionally serve the same purpose as to produce $\boldsymbol{r}\,[i+1]$ and $\boldsymbol{v}\,[i+1]$ given $\boldsymbol{r}\,[i]$, $\boldsymbol{v}\,[i]$, a callable acceleration function based on $\boldsymbol{a}_{ij}$, and a finite number of steps.

Once the $\boldsymbol{r}$ and $\boldsymbol{v}$ arrays are computed, potential energy data is derived from $\boldsymbol{r}$, and 3D momentum and kinetic energy data are derived from $\boldsymbol{v}$ and $\boldsymbol{v}^2$ respectively. In the exploration that follows, $N = 4$ for two separate sets of initial conditions, run from $t = 0$ to $t = 100\,$years with a time step of $dt = 2\,$hours. The first set of initial conditions involves point masses with the masses of the sun, Mercury, Venus, and Earth, and the other involves four bodies all close to the mass of the sun in what the author claims could represent the emergence of two binary star systems.

Lastly, one body in the system is chosen for purposes of exploring phase space. In our specific runs, phase space plots are generated for the Mercury-like point mass for stable initial conditions, and a sun-like point mass of $2^{30}\,$kg for the more chaotic second set of initial conditions.

3 Physical Analysis

The primary way of analyzing the physical accuracy of this simulation is via the energy of the system. Kinetic energies are calculated via the square of the velocity at each time step in each direction, and the potential energies are calculated via

$$U = -\sum_{1\leq i\leq j\leq n} \frac{G m_i m_j}{\|\boldsymbol{r}_j - \boldsymbol{r}_i\|},$$

so if they behave in a way such that the total system kinetic energy T is maximized when U is minimized, and vice versa, the simulation is at least following the physical law of conservation of energy, and corroborating the numerical calculation of *both* $\boldsymbol{r}$ and $\boldsymbol{v}$. As shown in the figures below, U and T always seem to cancel out "by eye" to a common constant, so we can have faith in the model. For IC1, the height and depth of energy peaks seem to be the same across stepping methods. For the more chaotic IC2, however, there is variability in the peak and depth height on the total energy plot. We can currently only speculate as to what governs the peak density in the total energy plot, but note that there are more energy peaks for stepping methods considered scientifically more robust.

4 Error Analysis

Analysis of error is difficult for the simulation as no analytical solution exists for this problem. Additionally, initial conditions that produce stable (but nontrivial) 3D trajectories are difficult to come upon by guessing and checking, and they are even more

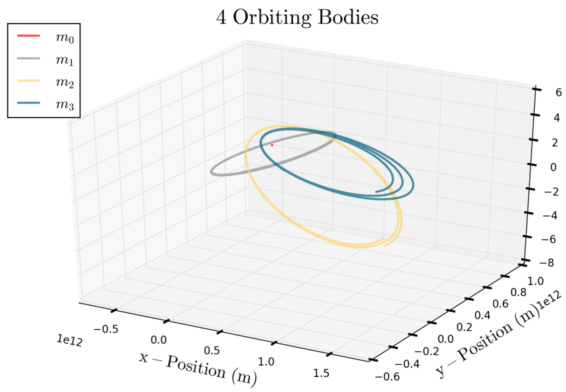

Figure 3: caption

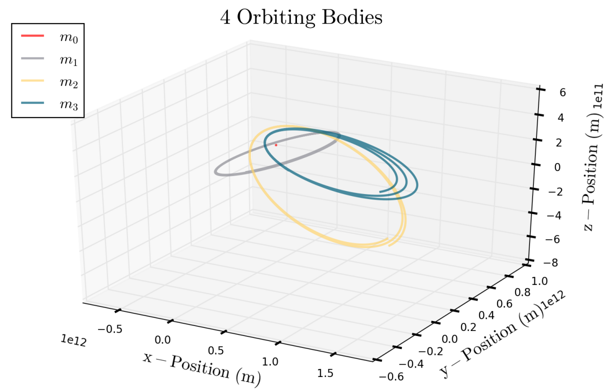

Figure 4: caption

difficult to cross-check with other simulations. Initially, it was thought to compare results with the industry standard Python N-body simulator known as Rebound, but the package is limited to $2D$ orbits, and is thus provides an ineffective comparison. Instead, we are able to show that as the time step dt is reduced and the number of steps taken by the system increased, the bodies get closer to real physical trajectories. As an example, starting with only 20 steps ($dt \approx 9000\,\text{years}$) yields trajectories that just shoot outward from the center, but with 175000 steps the results look far more intuitively realistic. Additionally, there is further proof that there are insignificantly small error in the simulation in that the system behaves exactly as expected for unnatural, but mathematically symmetrical initial conditions. If two bodies are initialized in 3D space within the simulation (at distinct positions) and given zero initial velocity, they only oscillate back and forth on the line connecting the two bodies. In a separate test where there is a large massive object at the origin and six smaller objects symmetrically positioned around it at $(0, 0, \pm1)$, $(0, \pm1, 0)$, and $(0, 0, \pm1)$, and every massive body starts with zero initial velocity, the system again oscillates back and forth in a predictable manner: the massive body stays still while the smaller bodies oscillate back and forth along the coordinate axes.

5 Conclusions

Concluding, this numerical implementation accurately provides a means of investigating the centuries-old N-body problem in three dimensions. Although initial conditions for stable systems are difficult to discover, we are able to create nearly stable orbits by using values close to those that we find in our solar system. In the same sense, having one mass in the set of massive bodies a few orders of magnitude larger that the rest seems to provide stability to the system. Unstable systems are easier to produce, and are found especially when the bodies in the simulation are all close to the same mass. For stable systems, all stepping implementations tend to agree on the numerical results, with only the Euler method producing clearly erroneous results due to the fact that it does not conserve energy. For unstable systems, the stepping methods predict significantly different system behavior, but it is noted that the most advanced ones (RK4, VEFRL, and PEFRL) all qualitatively produce the same orbital trajectories, albeit with slight variations. This simulation is powerful in that it truly may accommodate N bodies if given the appropriate amount of initial conditions, thus laying a foundation for further N-body simulation research that might entail automated generation of initial conditions, or importing them from modern stellar surveys.