PHYS486-FINAL: The N-Body Problem

Nicholas Cemenenkoff

June 10, 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 seem 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 script `orbit.py` 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 certain 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, and how can this be confirmed?

2 Numerical Solution

At its core, the numerical solution to this problem involves solving for the instantaneous accelerations for a given a set of position data. The accelerations on body m_i felt by all other masses for a given set of positions is given by

$$\boldsymbol{a}_{ij} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{Gm_j\left(\boldsymbol{r}_j - \boldsymbol{r}_i\right)}{\left\|\boldsymbol{r}_j - \boldsymbol{r}_i\right\|^3}.$$

There are several stepper methods that calculate the instantaneous accelerations $\boldsymbol{a}_{ij}$ for every body, but they do so in different ways. The employed methods are:

1. Euler,
2. Euler-Cromer (EC),
3. 2nd Order Runge-Kutta (RK2),
4. 4th Order Runge-Kutta (RK4),
5. Velocity Verlet (VV),

6. Position Verlet (PV),

7. Velocity Extended Forest-Ruth-like (VEFRL), and

8. Position Extended Forest-Ruth-like (PEFRL).

The Euler method is the only method on the list that does not conserve energy. Its non-energy-conserving nature 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. As expected, the most advanced methods (RK4, VEFRL, and PEFRL) are the slowest algorithms. We've found these methods can often be up to 10 times as slow as the Euler method for varying initial conditions and $2 \leq N \leq 10$. To see which method is most likely the most robust for large N, the methods were timed for the average number of iterations per second given $N = 10$, $t_f = 20\,\text{years}$, and $dt = 2\,\text{hours}$.

method	iterations/second
Euler	1440.19
EC	1454.08
RK2	728.48
RK4	365.23
VV	728.95
PV	1451.20
VEFRL	293.02
PEFRL	365.26

Given these data, we conclude that PV is the best algorithm for exploring large systems because it is symplectic, energy-conserving, and also nearly just as fast as the Euler or Euler-Cromer methods. Interestingly, leapfrogging methods that propagate the *position* first rather than the velocity are faster. This may have to do with the fact that calculating velocity values requires an extra $\boldsymbol{a}_{ij}$ function call, thus slowing down the iteration considerably.

The Verlet and Extended Forest-Ruth-like methods not only conserve energy, but are also symplectic (geometry 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 n.

Once the $\boldsymbol{r}$ and $\boldsymbol{v}$ arrays are computed, potential energy data is derived from $\boldsymbol{r}$, and 3D momentum

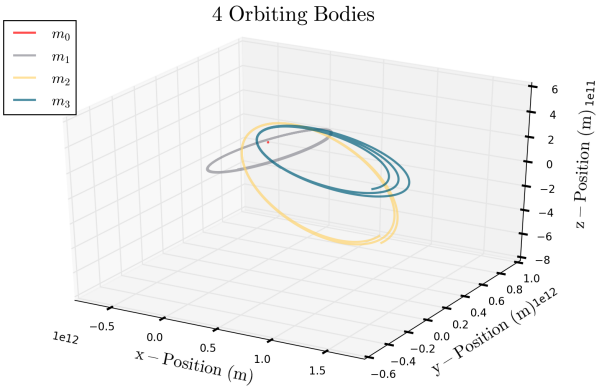

Figure 1: caption

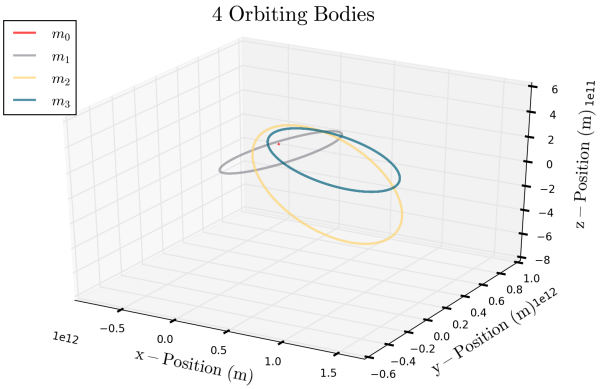

Figure 2: caption

and kinetic energy data are derived from $\boldsymbol{v}$ and 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 (IC1) involves point masses with the masses of the sun, Mercury, Venus, and Earth, and the other (IC2) 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 then a sun-like point mass of 2^{30} kg for the more chaotic second set of initial conditions.

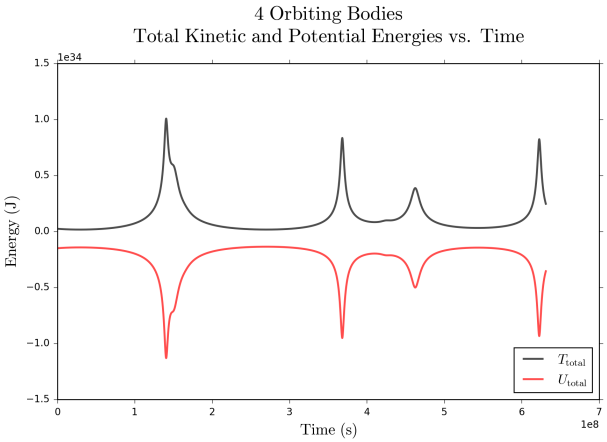

Figure 3: IC1 Euler

3 Physical Analysis

3.1 Analysis of Four Bodies

The primary way of analyzing the physical accuracy of this simulation is via the energy of the system, and secondarily through phase space analysis. Kinetic energies for each body are calculated via the square of the velocity at each time step in each direction via $\frac{1}{2}m_i\boldsymbol{v}_i^2$. The total system potential energy is calculated at each time step via

$$U = -\sum_{1\le i\le j\le n} \frac{Gm_im_j}{\|\boldsymbol{r}_j - \boldsymbol{r}_i\|},$$

so if the bodies 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, thus corroborating the numerical calculation of *both* $\boldsymbol{r}$ and $\boldsymbol{v}$. As shown in the figures to the right, 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. Note additionally in Figure 8 that each body's kinetic energy has a periodic nature.

Next, we employ PV and explore phase space for IC1. Notice next (via the phase space figures) that in both the momentum and kinetic energy spaces, m_1 follows a cyclic trajectory repeatedly, indicating

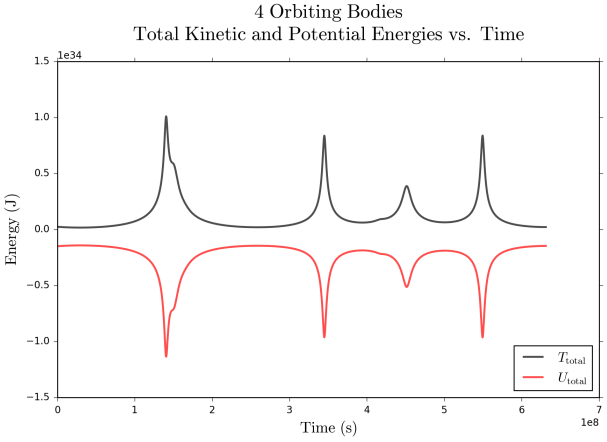

Figure 4: IC1 velocity Verlet

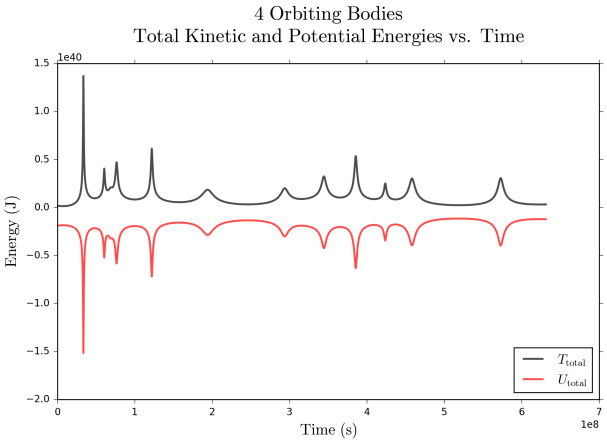

Figure 5: IC2 Euler

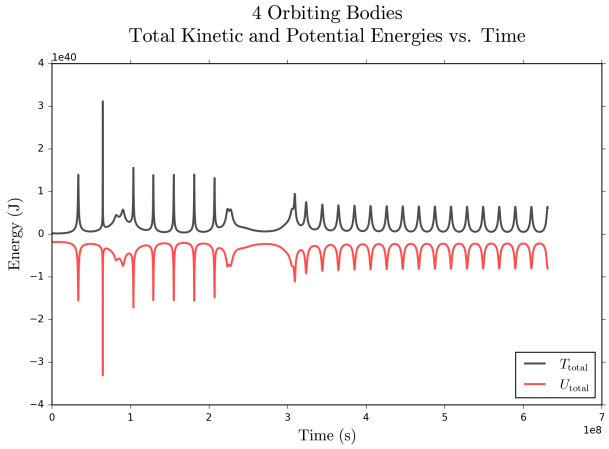

Figure 6: IC2 velocity Verlet

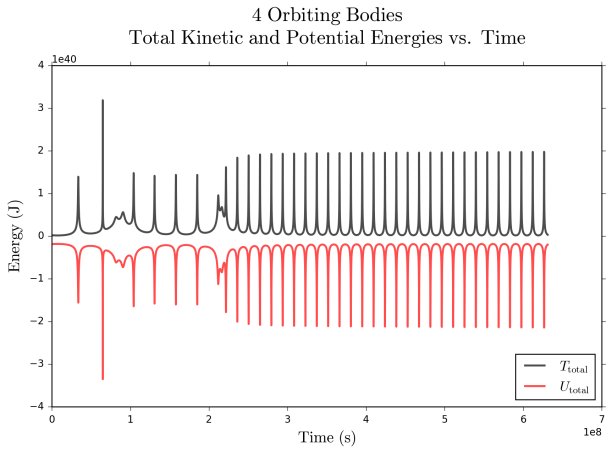

Figure 7: IC2 PEFRL

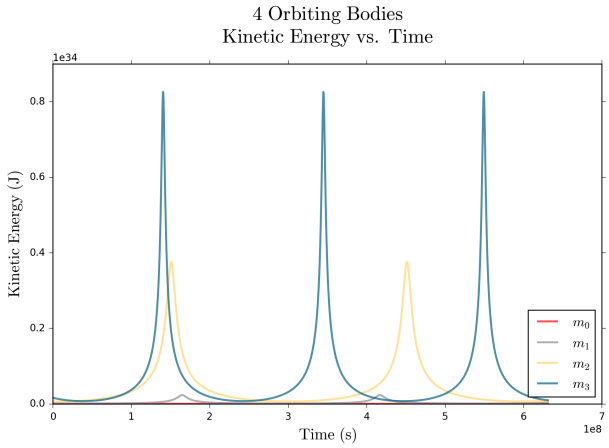

Figure 8: IC1 PEFRL-calculated kinetic energies for each body

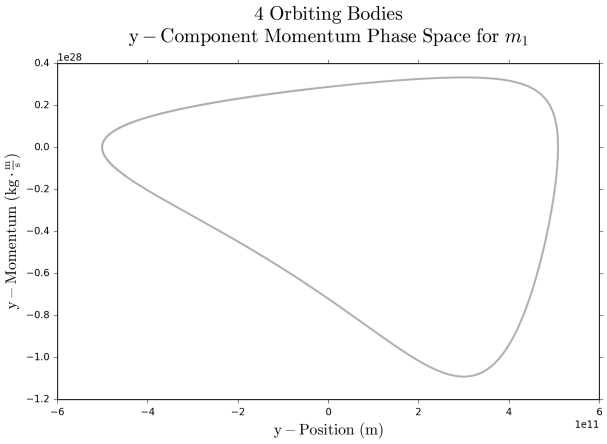

Figure 9: IC1 PV-calculated momentum phase space for m_1

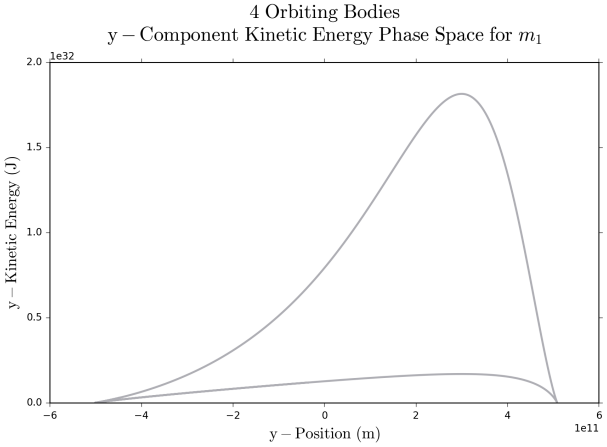

Figure 10: IC1 PV-calculated kinetic energy phase space for m_1

momentum and kinetic energy (in addition to total energy) are conserved.

Since IC2 is far more chaotic than IC1 and doesn't seem to have a stable, repeated trajectory, we employ the state-of-the-art PEFRL algorithm when exploring phase space for m_1. Note in the following figures that there is still a cyclic nature in the plots, but the cycle tends to drift, indicating that energy and momentum are conserved as m_1 translates through space. If we could somehow put ourselves in the frame of m_1, the associated phase space plots would be similar to those for m_1 in IC1 and continually retrace themselves. All of these observations taken together heavily corroborate the physical accuracy of our numerical implementation: both energy and momentum are conserved.

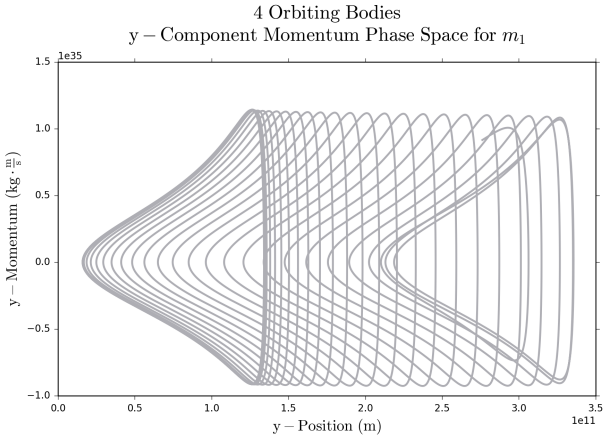

Figure 11: IC2 PEFRL-calculated momentum phase space for m_1

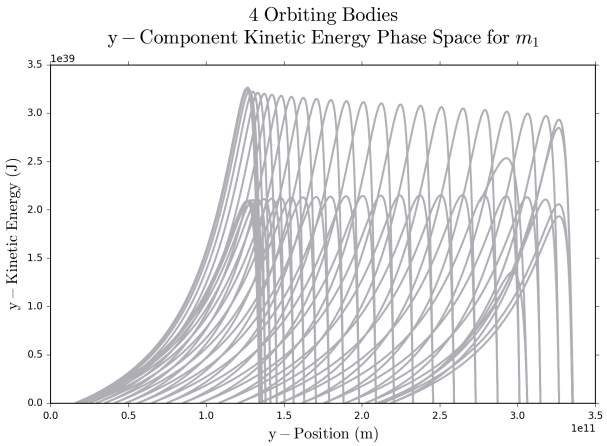

Figure 12: IC2 PEFRL-calculated kinetic energy phase space for m_1

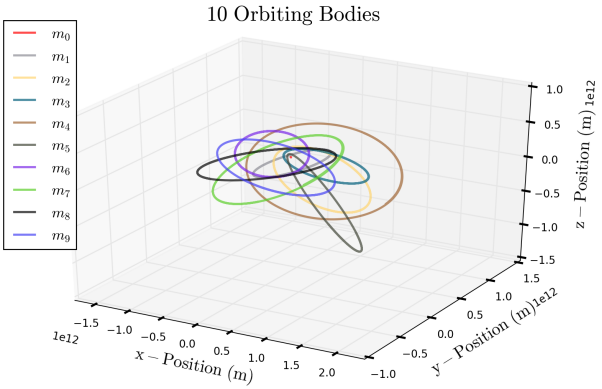

Figure 13: IC1 PV-calculated orbital trajectories for t_f = 20 years

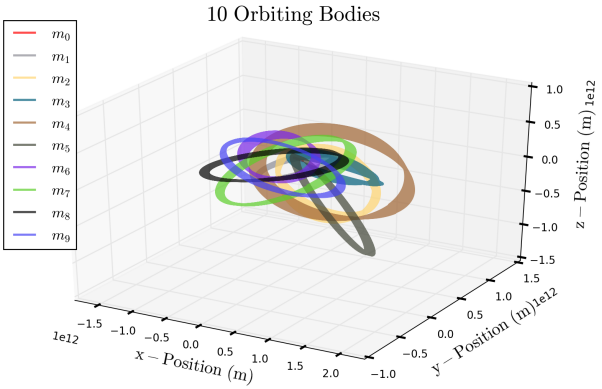

Figure 14: IC1 PV-calculated orbital trajectories for t_f = 500 years

3.2 Ten Bodies

Here we start to show the powerful general nature of the simulation, and also peculiar properties that are characteristic of its 3D nature. In Figures 13 and 14, we show a system of $N = 10$ bodies from the stable IC1 set, and investigate the orbital trajectories when allowed to propagate forward through time for $t_f = 20$ years versus $t_f = 500$ years, each with $dt = 2$ hours via the PV algorithm (symplectic and energy-conserving).

Despite the symplectic nature of the PV stepper, the system still seems to "drift" or "smear" over time. Perhaps if given a long enough of a period, this drifting would also be periodic, but it might take a few days or even weeks of continuous run time to confirm such a result. For IC1 simulations with $N = 3$ or $N = 4$, this smearing effect isn't apparent. The author believes this smearing effect has to due with some sort of instability that is introduced when the number of bodies starts to get large. Note also that this smearing effect is exhibited in phase space plots

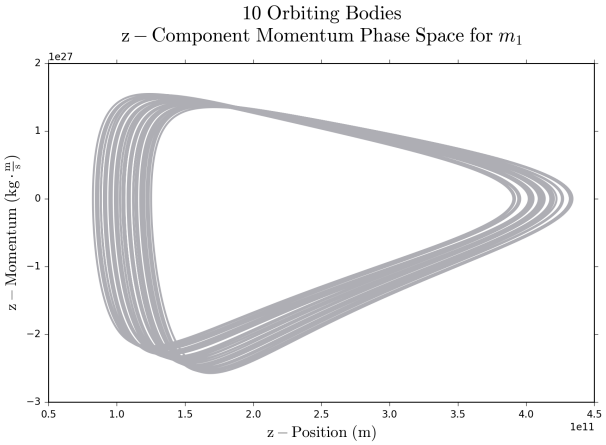

Figure 15: IC1 PV-calculated momentum phase space for m_1

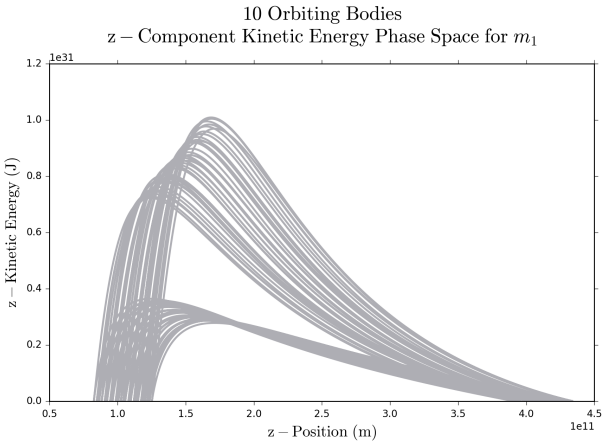

Figure 16: IC1 PV-calculated kinetic energy space for m_1

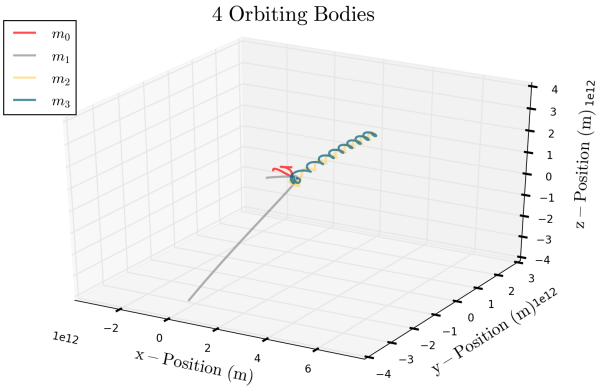

Figure 17: IC2 $dt = 15$ hours

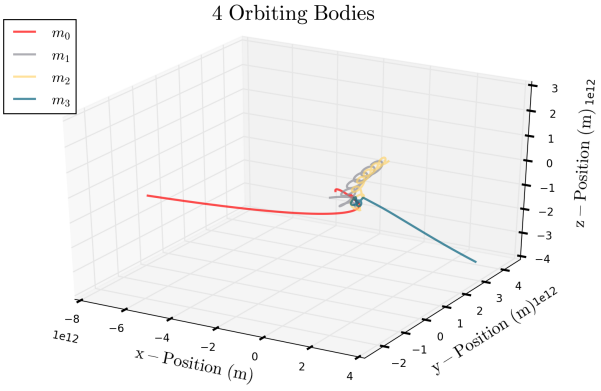

Figure 18: IC2 $dt = 7$ hours

for m_1. Each cycle seems to contain the same amount of area, so a possible explanation is that energy is indeed conserved, but the entire system's center of mass is following what may or may not be a periodic trajectory. Either way, at least artistically, the orbital trajectory plots are vaguely reminiscent of some sort of ringed planet like Saturn.

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 difficult to cross-check with other simulations. At first 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, thus providing 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 converge to a single physical system trajectory. We use the more chaotic IC2 for our investigation. Starting with only 11700 steps ($dt = 15$ hours) yields trajectories that just shoot outward from the center with minimal orbiting, but with 175000 steps the results look far more intuitively realistic. We can also see that at lower time steps, the difference between overall results gets increasingly smaller. This qualitatively shows that the numerical solution converges to a single solution as the time step is reduced. Note that for stable orbital trajectories as in IC1, a time step of $dt = 15$ hours produces essentially the same orbital paths as $dt = 1$ hour.

Additionally, there is further proof that there is 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

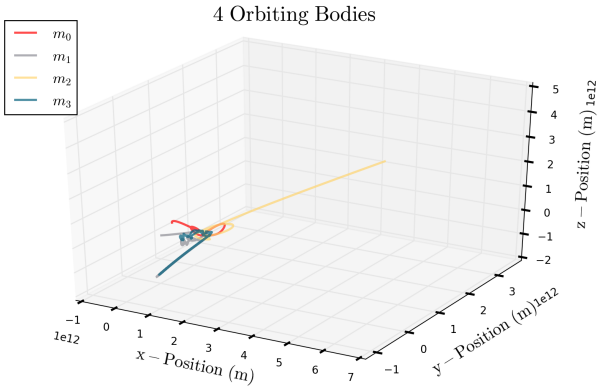

Figure 19: IC2 $dt = 4$ hours

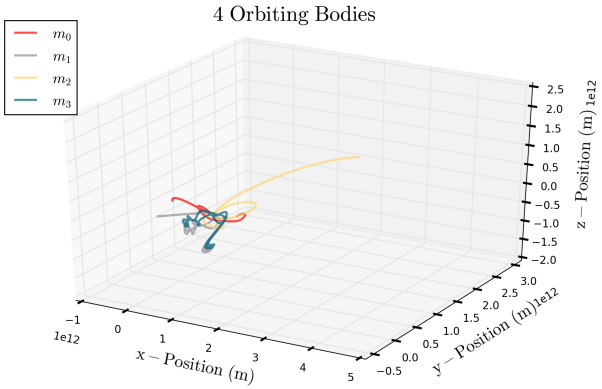

Figure 20: IC2 $dt = 2$ hours

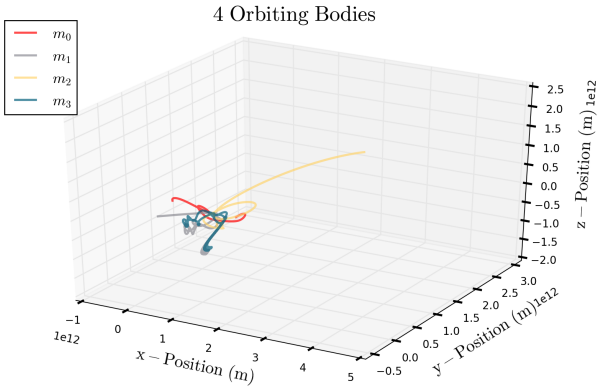

Figure 21: IC2 $dt = 1$ hour

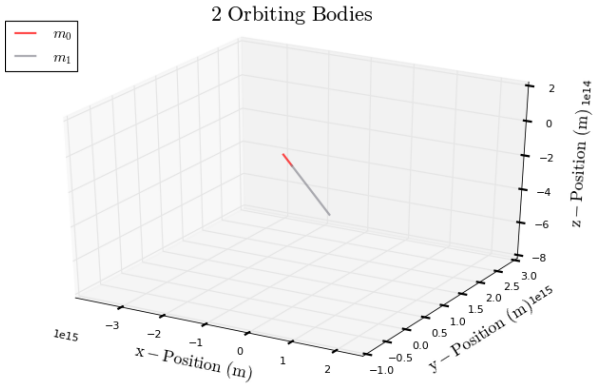

Figure 22: Two bodies with unequal masses initialized at distinct positions with zero initial velocities only travel along the line that connects them.

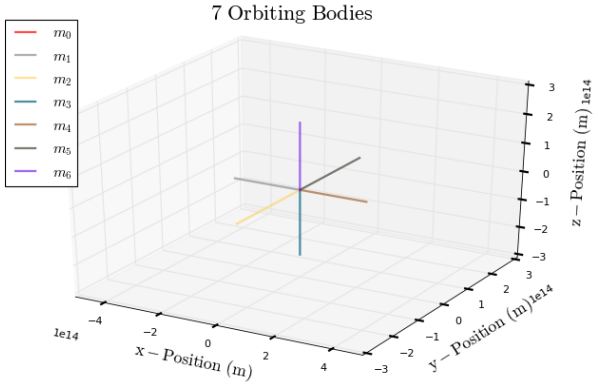

Figure 23: There is a central body with the mass of the sun, and then six symmetrically arranged bodies with the mass of Mercury. All bodies are given zero initial velocities. As might be expected, because the perfect symmetry of the initial conditions, the resulting orbital trajectories are also perfectly symmetrical.

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 $(0, 0, \pm 1)$, $(0, \pm 1, 0)$, and $(0, 0, \pm 1)$, 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. Given the investigation into both the speed and accuracy of each numerical integration method, PV is found to be the most appropriate exploratory method, with PEFRL providing the most appropriate means for investigating systems in higher resolution. This simulation is powerful in that it truly may accommodate N bodies if given the appropriate amount of initial conditions, thus it lays a foundation for further N-body simulation research that might entail automated generation of initial conditions, or perhaps importing initial conditions from modern stellar surveys. Lastly, the author encourages the reader again to explore the associated Python script `orbit.py` to independently investigate a new N-body system. Who knows what mysterious patterns you may find?