Astr 596: Project 2 Research Memo

Nodoka Masamune

September 2025

1 Executive Summary

For this project we simulated N body dynamics using a variety of ODE integration and approximation methods. From these simulations we were then able to track and plot the energies of these systems, as well as changes in the relative energy error and virial ratio as the simulation progressed. We then used these plots to compare the accuracy of these various integration methods to each other. For our smaller two- and three- body simulations we also graphed the trajectories of these objects. Finally, for my extension project I created two animations of cluster evolution.

The four integration methods we used were Euler's method, the Runge-Kutta 2 (RK2), the Runge-Kutta 4 (RK4), and the leapfrog integration method. We began by simulating the Earth's orbit around the Sun, a two body problem. Then, we used Earth and Jupiter's orbit around the Sun to simulate a three body problem.

After validating that these systems were working as expected, we moved onto larger systems. We began by randomly generating locations within a given radius of 100 AU for ten stars with identical masses of one solar mass and zero initial velocities. Their dynamics were calculated using the RK4 and leapfrog integration methods, which we had determined were the two most accurate integration methods via the two body simulation. This was determined by their close accuracy when calculating orbits, as well as their relatively small relative energy errors.

Finally, we used the canonical Kroupa Initial Mass Function (IMF) and Plummer sphere model to generate a more realistic stellar cluster. The IMF is a power law that determines how many stars form at each mass. The Plummer model is a model of density in spherical stellar clusters. We were able to use these two simple laws to generate a population of N stars, with masses determined by the IMF and initial positions and velocities determined by the Plummer model.

We used the leapfrog integration method to model this stellar cluster's dynamics. Plots of various energies, the relative energy error, and virial ratio vs. time were then plotted. Finally, several plots graphing the changes in location of each body.

Finally, for my project extension I created two animations showing the movement of the stars in the stellar cluster. The first animation shows the stellar dynamics of a physically realistic star cluster generated by the IMF and Plummer equations, while the second animation shows the calculated motions of a stellar population generated using randomization. The details for why are covered in the extension section of this report.

We found that the two best methods for integration were the RK4 and leapfrog integration methods. The Euler method showed clear deterioration in the 2 body simulations where it calculated Earth's orbit would spiral away from the Sun in a decade. This is also obvious in its plot of relative energy error vs time, which grew incredibly quickly over the same time frame. The RK2, while better, also showed a visible wobble in its Earth orbit, and also had large errors. Again, this is over the course of ten years, which is barely a blink of an eye in astronomical terms. In comparison, the RK4 had incredible accuracy with its calculated orbit and a very small increase in relative energy error. The leapfrog method also had an accurate orbit, but more importantly, it showed perfect conservation of energy over time. For these reasons, only the RK4 and leapfrog methods were used in simulations after the 2 body simulation.

2 Methodology

Methodology: approach, algorithms, and numerical methods

The four integration methods we were working with were the Euler's method, RK2, RK4, and the leapfrog method. Euler's method takes on the form:

$$\vec{y}_{n+1} = \vec{y}_n + \triangle t \cdot \vec{f}(t_n, y_n) \tag{1}$$

It is a first order integration method, and the simplest one we will be testing. For every step, we calculate the acceleration, and then use it to calculate the velocity and position in that moment. As we see later, errors increase very quickly.

The RK2 is a second order midpoint method that takes the form:

$$\vec{k}_1 = \vec{f}(t_n, y_n) \tag{2}$$

$$\vec{k}_2 = \vec{f}(t_n + \frac{\triangle t}{2}, y_n + \frac{\triangle t}{2}\vec{k}_1) \tag{3}$$

$$\vec{y}_{n+1} = \vec{y}_n + \triangle t \cdot \vec{k}_2 \tag{4}$$

The RK4 is a similar 2nd order method, but has much higher accuracy over time. Its basic form is:

$$\vec{k}_1 = \vec{f}(t_n, y_n) \tag{5}$$

$$\vec{k}_2 = \vec{f}(t_n + \frac{\triangle t}{2}, y_n + \frac{\triangle t}{2}\vec{k}_1) \tag{6}$$

$$\vec{k}_3 = \vec{f}(t_n + \frac{\triangle t}{2}, y_n + \frac{\triangle t}{2}\vec{k}_2) \tag{7}$$

$$veck_4 = \vec{f}(t_n + \triangle t, y_n + \triangle t \cdot \vec{k}_3) \tag{8}$$

$$\vec{y}_{n+1} = \vec{y}_n + \frac{\triangle t}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \tag{9}$$

The final integration method we used is the symplectic (energy-conserving) second order leapfrog method, where velocities are calculated in 'half steps,' while position is calculated in 'full steps':

$$\vec{v}_{n+\frac{1}{2}} = \vec{v}_n + \frac{\triangle t}{2} \cdot \vec{a}(\vec{r}_n) \tag{10}$$

$$\vec{r}_{n+1} = \vec{r}_n + \triangle t \cdot \vec{v}_{n+\frac{1}{2}} \tag{11}$$

$$\vec{v}_{n+1} = \vec{v}_{n+\frac{1}{2}} + \frac{\triangle t}{2} \cdot \vec{a}(\vec{r}_{n+1}) \tag{12}$$

We first began by testing only the Euler method integration package we wrote on the two body system of the Earth and Sun. After validating that it produces the expected 'spiral out' trajectory over a course of ten years, as well as the expected energy drifts, we moved onto creating the three other integration packages, which were subsequently written into a single class package. This was done to simplify its use and to increase efficiency; as all four methods had been written to use the same inputs and structure, combining all four into the same package ensured that code did not have to be repeated unnecessarily. The other three systems were similarly tested with the Earth and Sun.

From these results we decided to proceed with the three body simulation of Earth, Jupiter, and the Sun using only the RK4 and leapfrog methods. By using three bodies, we can ensure that our integration package can properly calculate the total gravitational force an object feels from two or more different objects.

After confirming that our integrators were able to handle adding different forces into one resultant force, we moved onto randomly generating small groups of stars confined to a sphere of radius 100 AU. We wrote a small package that randomly generates initial positions for N number of bodies within this sphere, with pre-set masses of $1M_\odot$ and initial velocities, $\vec{v}_0 = 0$. The dynamics of this 'stellar cluster' was then simulated by using both the RK4 and leapfrog methods. This was to ensure that our integrators worked in both two- and three- dimensions.

Due to the leapfrog method's better performance over larger timescales the following simulations used only the leapfrog method.

After confirming that our integrators worked in three dimensions and showing that the leapfrog method is more accurate over larger frames of time, we began simulating physically accurate stellar clusters. These stellar clusters were generated using the Kroupa Initial Mass Function (IMF) and the Plummer sphere model.

The IMF determines the distribution of masses. We simulated a two segment IMF, which takes on the form:

$$\xi_1(m) = k_\xi k_i m^{-\alpha_i}, m \in [m_{min}, m_{max}] \tag{13}$$

$$\alpha_1 = 1.3, (m_{min} \leq \frac{m}{M_\odot} < 0.5) \tag{14}$$

$$\alpha_2 = 2.3, (0.5 \leq \frac{m}{M_\odot} < min_{max}) \tag{15}$$

The minimum mass, $m_{min} = 0.08M_{\odot}$, was chosen as this is the hydrogen burning limit; the maximum mass, $m_{max} = 150M_{\odot}$, was chosen as this is the approximate upper stellar mass limit.

By setting a number, N, we could generate an array of masses that were physically realistic.

The Plummer model is a mathematical model of density in spherical stellar clusters. We used it to set initial positions and velocities for each star, given its mass. The Plummer density is given by the equation:

$$\rho(r) = \frac{3M}{4\pi a^3}(1 + \frac{r^2}{a^2})^{\frac{-5}{2}} \tag{16}$$

Then, the mass enclosed within a certain radius, r is:

$$M(< r) = M\frac{r^3}{(r^2 + a^2)^{(3/2)}} \tag{17}$$

By creating packages with the IMF and Plummer sphere model encoded, we were able to generate physically realistic stellar clusters to then simulate with our leapfrog integrator.

Finally, I attempted to vectorize my leapfrog function; unfortunately, due to the way I had defined my calculations I found this very difficult. I had already written most of my function to use numpy arrays rather than using loops, but I did this in a way that still required some level of loops in my implementation, which I was unable to get rid of. This meant that I struggled to and was ultimately not able to vectorize my function in any meaningful way.

3 Results

The results of all the Earth-Sun system simulation calculated by all four methods were plotted (see figure 1) to compare their accuracy over time. From these plots we can see that the Euler method performed worst, with visible errors in its projected trajectory and significantly larger errors compared to the other three. The RK2, while having a much smaller relative error in energy in the same time frame, also produced a visibly wobbly orbit. On the other hand, both the RK4 and leapfrog methods showed errors smaller by several orders of magnitude. All other methods had significantly smaller virial ratio changes compared to the Euler method as well.

The results of the three body Earth-Sun-Jupiter simulation can be seen in FIGURE ????. As we can see, both RK4 and leapfrog had comparable results. Neither calculated disastrous trajectories for the Earth or Jupiter, and we can also see that Earth's orbit is slightly distorted from Jupiter's gravitational influence.

We then simulated the evolution of a 10 body system generated by our random position simulator using the RK4 and leapfrog methods; the results can be seen in figure 2. As seen in the figures, the RK4 method showed significantly more energy drift than the leapfrog method. While the RK4's energy error predictably drifts higher, the leapfrog method should show an oscillating error bound to a relatively small range.

A stellar cluster of 100 stars was generated using the IMF and Plummer models, which was then simulated for a hundred years using the leapfrog integration method. The results of this can be seen in figure 3. As you can see, there is a slight drift in the relative energy error, which is unexpected. This could be due to an issue in calculating total energy that did not arise until larger numbers of objects were simulated for larger durations. My attempts at finding this error were unsuccessful; however, I suspect the issue comes from the gravitational potential energy calculations, as the graph shows a straight line which is also unexpected.

I also plotted cluster 'snapshots,' which are 3D plots of the stars' location at various points of the simulation. These times were at the very beginning, a quarter of the way through, halfway through, three quarters of the way through, and at the very end. Due to the relatively small timescale (100 years), the relatively small number of stars (100), and the size of the sphere set by the scaling factor (100 AU for systems with N ¡ 1000 stars), the relative movements of the stars compared to the size of the cluster, is quite small and hard to discern; however, close inspection does show some movement. These plots can be seen in figure 4.

4 Validation

Validation was necessary at several key points of this project. Initial validation was done by graphing and comparing the four different integration methods during the two body simulation, where I showed that the Euler method was the worst approximation, while both the RK4 and leapfrog methods had the least amount of relative error growth. Again, this can be seen in figure 1.

Continued plotting of the changes in energy, error, and the virial ratio over time for the numerous simulations were used to confirm that the leapfrog method was superior in long-term accuracy compared to the RK4. This was expected as the leapfrog method conserved energy while the RK4 did not.

Validation for the IMF and Plummer method implementation was done via several steps. First, I confirmed that the generated masses fit the distributions across the two mass segments predicted by the alpha values and midpoint

First I confirmed that two mass segments contained the same proportion of stars as predicted by the probabilities set by the alpha values and breakpoint mass. This was done by calculating what proportion of the generated masses were above and below the set breakpoint mass, and comparing it to the proportion values calculated by the model.

Afterwards, a histogram of masses was generated (see figure 5). We can see that the histogram changes slopes at the breakpoint mass, with the initial segment having a less steep slope. This matches expectations set by the IMF.

The initial segment's slope should be

$$-/alpha = -1.3$$

and the second slope should be

$$-/alpha = -2.3$$

/.

The Plummer implementation was then validated by graphing the radial density profile of the generated initial positions. This can be seen in figure 6. This density profile matches published Plummer radial density profiles, confirming that it has been encoded properly.

5 Extension

For my extension, I created two animated gifs showing cluster evolution as time passed. I chose this extension for two reasons: the first is that I wanted to gain more experience with plotting in python, which I see as a weakness of mine. The second is that I found the final results of the project really fascinating, and I was interested in actually seeing how these objects interacted with each other. I really enjoyed being able to see the trajectories of Earth and Jupiter around the sun in the earlier simulations, and wanted to be able to see the movements in the larger stellar clusters I was later simulating.

The first animation shows the movement of the stellar cluster produced by the IMF and Plummer model; however, I faced strict restrictions with both the number of stars and duration I could simulate due to computational limits. This made it so that I actually saw very little movement in my animation. Since the scaling factor that determines the location of my objects was 100 AU for systems with less than a thousand bodies and 1000 AU for systems with more, stellar motions were relatively small compared to the scale of the system. This was doubly true for the timescales I was able to compute; my computational power maxxed out at around a hundred years. This all resulted in a relatively lackluster animation, so I decided to create a second one.

This second animation showed the cluster evolution of an artificially massive cluster. I did this by implementing my random generator function; since it generates stars with an initial mass of $1M_{\odot}$, this meant that the masses were, on average, larger than the ones generated in the IMF, which meant that their gravitational influences were also larger. I then created an animation of a simulation with the same number of stars and same duration, which showed actual movement.

6 Conclusions

Through the implementation of the IMF and Plummer functions I gained a far deeper understanding of what these models actually represented in the real world. While the math was intimidating, my understanding of it at the end was incredibly satisfying and being able to see it validated was also gratifying.

This process also taught me a lot about numpy array slicing and indexing; this was my first time working with more complicated array shapes, so by having to both assign and refer to specific indices in arrays of shape N X 3 for the final leapfrog simulations was initially very difficult for me.

Ultimately we were able to show that the leapfrog integration method was the best integration approximation for longer timescales. This is because it conserves energy, so while individual point-by-point approximations may be more accurate through the RK4 method, which is higher order, in systems with conserved energy and over long enough durations, the leapfrog method proved superior. Simulations calculated by the leapfrog method showed far smaller energy errors in comparison to other methods.

7 Appendix A: Figures

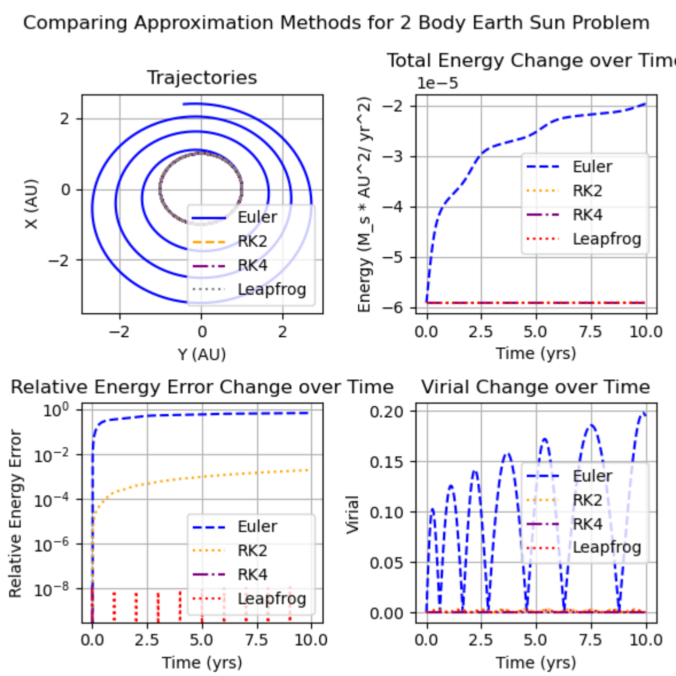


Figure 1: Four plots comparing various outputs of the 2 body earth Sun simulation for the Euler method, RK2, RK4, and leapfrog integration methods. The first plot shows projected trajectories. The second shows total energy change over time. The third shows the relative energy error. The fourth shows the virial ratio over time.

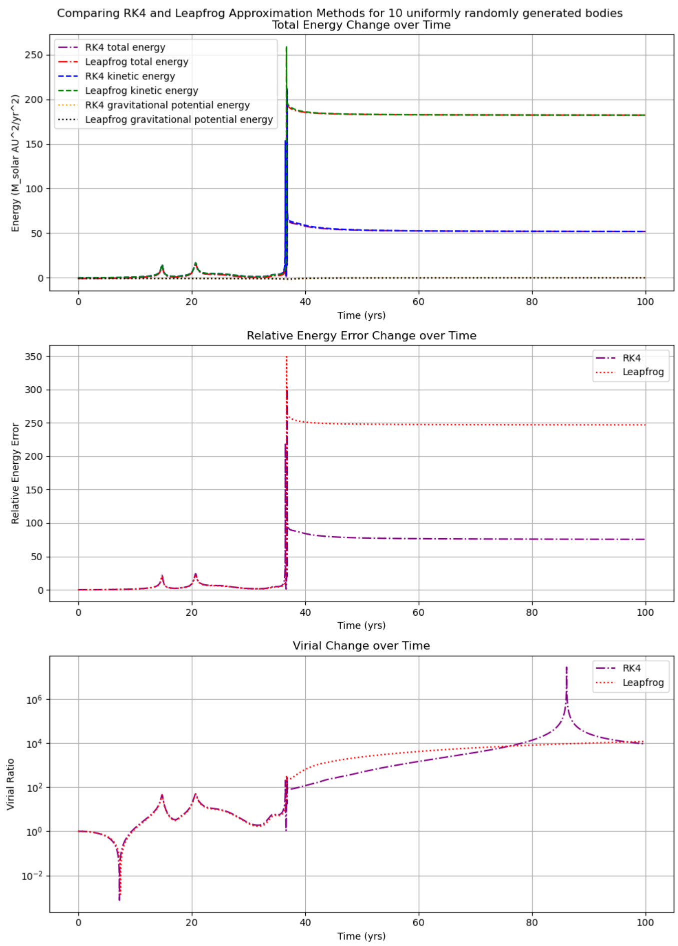


Figure 2: Three plots comparing various outputs from the 10 body simulation calculated by the RK4 and leapfrog integration methods.

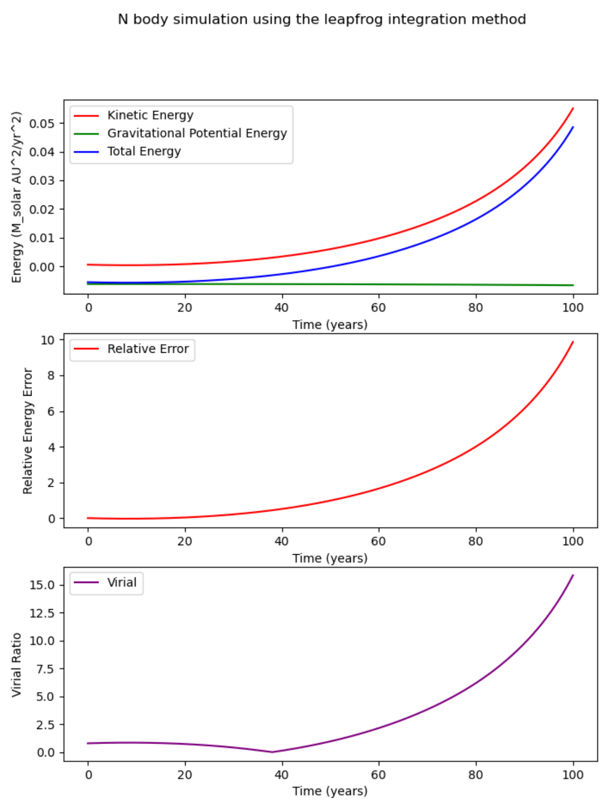
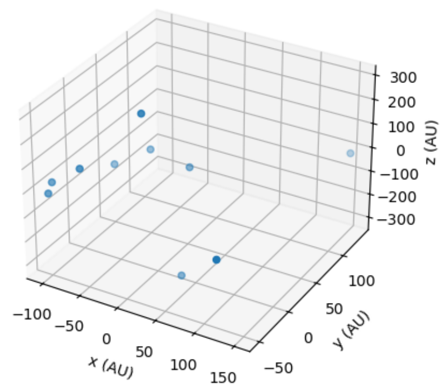
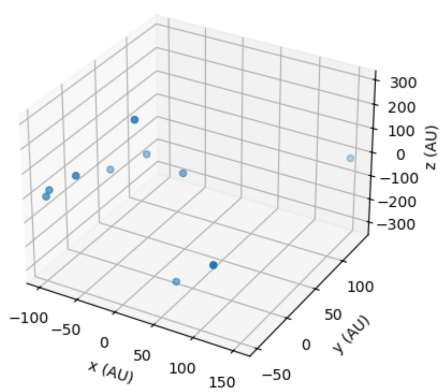


Figure 3: Three plots showing the various outputs from the N=100 simulation calculated by the leapfrog integration method.

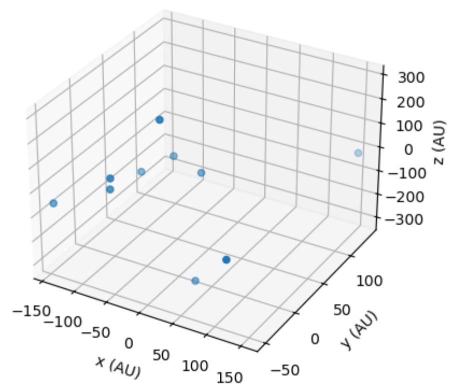
Cluster at t/duration = 0



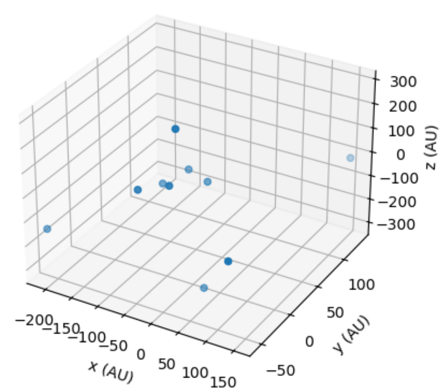
Cluster at t/duration = 0.25



Cluster at t/duration = 0.5



Cluster at t/duration = 0.75



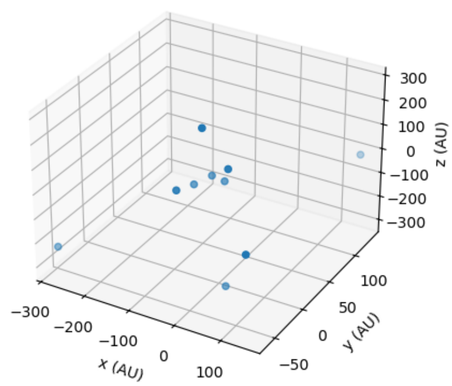


Figure 4: Five plots showing cluster evolution at different snapshots in time.

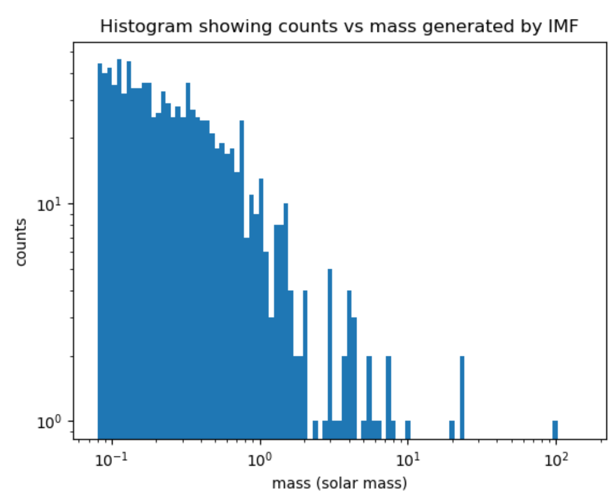


Figure 5: Histogram of generated masses used to validate IMF implementation.

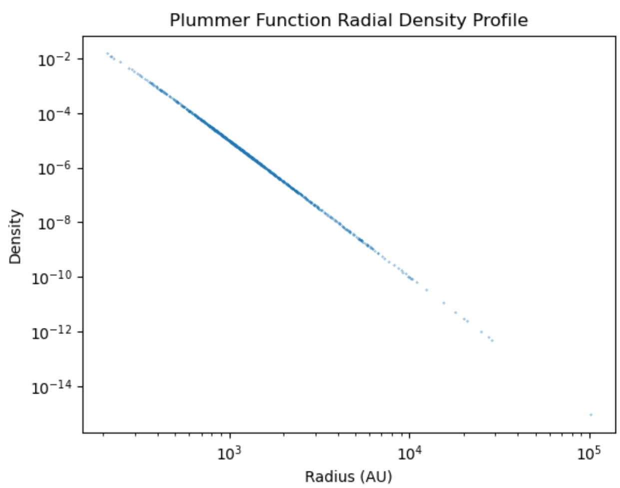


Figure 6: Radial density profile of generated stellar positions, used to validate the plummer function implementation.