Simulating the asteroid belt and other ring distributions using N-body tree codes

Steven Shockley

December 2022

Abstract

This paper describes the use and results of an N-body Barnes-Hut algorithm tree code for the simulation of the solar system's asteroid belt as well as a test of Jupiter's ability to clear its own orbit given a similar belt structure at its orbital radius. The solar system is created using the planet's semi-major axes and eccentricities from Mercury to Jupiter, and the asteroid belt approximated using a torus. The belt placed at Jupiter was approximated using the same structure at Jupiter's orbital radius. In the case of limited particles and realistic particle masses, the belt proved exceptionally stable, and increasing each particle's mass to approximate the belt's total mass produces a similar result. An attempt to increase the number density of particles failed, though computational time proved efficient. Finally, Jupiter proved its ability to clear its orbit, though the timescales of the simulation weren't long enough to fully evolve to our solar system's current state. Improvements in parallelization and initial conditions could lead to more realistic simulations and further improvements.

1 Simulation

The base code used in this project was taken from Homework 6. The RK4 method was removed in favor of exclusively using the leapfrog integrator. Both a simulator and a generator for object initial conditions were written, `nbody_it` and `generate_nbody` respectively. The command line arguments are mostly the same as in that case but also mostly deprecated, though still functional. The main argument now used is the `-F` flag (uppercase) which accepts an "options" file of the format shown in `./opts/default`. Below is a description of the options file.

- All scope declarations, in which settings are posted, begin with an `@` symbol and must be all upper case. Scopes include
  - `SIM`: simulator parameters, such as step-size, steps, modes, etc.
  - `IO`: input/output options.
  - `GEN`: generator methods and settings.
- Any global line without a leading `@` symbol will be ignored.
- Scopes must be enclosed by braces and not contain any spaces.
- Within the `SIM` scope you may declare options delimited by a colon, with no internal spaces, including
  - `softening`: the softening parameter, double.
  - `stepsize`: the step-size, double.
  - `steps`: the total number of steps to run for, integer.
  - `bh_mode`: whether to use Barnes-Hut mode. Either 0 or 1.
- For the `IO` scope:
  - `file_in`: a space delimited csv file in the Homework 6 format used to start the simulator, if not using generator scope.
  - `file_out`: deprecated.
  - `dir_out`: output directory for csv's named after the step. Must end in a trailing forward slash.
  - `console`: whether to print to console. Either 0 or 1.
  - `out_freq`: output frequency. On every step divisible by this frequency, a file will be printing (or to console).
- In the `GEN` scope:
  - `mode`: which mode to generate in. Can be '2body' or 'sphere', with the number of points in the sphere supplied by the command-line argument `-g [int]`.

  - `mass`: which mass distribution to use. 'uniform' will generate a constant mass.

In addition, the generator executable takes a file input using `-f` of a script file using the same rules and format as the options file, but with one addition and different keywords. The `-o` argument will print the results to a space delimited csv provided.

- With these new declarations, parentheses may be included between the scope and the opening brace that has an integer value of points.
- Comments may be made by preceding with two dashes.
- Units are as follows:
  - Distance: AU
  - Velocity: 30 km/s
  - Mass: $M_\odot$
  - Time: $\frac{1}{4\pi^2}$ years
  - Angles: radians
- In the `GROUP` scope:
  - The number of points listed must match the provided points in the parentheses.
  - Each list item must be preceded by a colon delimited declaration of `polar` or `cart`.
  - `polar` takes a point of the format: mass, radius, θ, ϕ, velocity, θ_v, ϕ_v. No spaces.
  - `cart` is in the format mass, x, y, z, v_x, v_y, v_z. No spaces.
- In the `RING` scope:
  - `r_inner`: the inner radius of the torus.
  - `width`: the cross-sectional width of the torus.
  - `e_max`: the maximum eccentricity of generated objects.
  - `height`: the b/a ratio of the cross-sectional ellipse.

  - `m_avg`: the average mass of a particle.
  - `m_sig`: the variance of the mass.
  - `m_orb`: the mass of the central object(s).
  - `GEN`: a keyword to signify the end of the scope to generate the ring.

Given that the simulator implements the Barnes-Hut algorithm, there is also a θ_{crit}. This is defined in the `bhmode_nbody.c` source file. By default, it is 0.05 (a common value) and if it is modified the program will ask to confirm before running that it is intended to be overwritten.

2 Results

Using the group declaration for the Sun and the first five planets and the ring to generate an asteroid using the following script:

```
@GROUP(6){
    -- polar:m, r, theta, phi, v, t_v, p_v
    -- cart:m, x, y, z, v_x, v_y, v_z
    polar:1.000000,0.000000,0.0,0.0000,0.000000,0.0000,0.0000
    polar:1.659e-7,3.870e-1,0.0,1.5708,1.580000,1.5708,1.5708
    polar:2.448e-6,7.233e-1,0.0,1.5708,1.167000,1.5708,1.5708
    polar:3.002e-6,1.000000,0.0,1.5708,1.000000,1.5708,1.5708
    polar:3.228e-7,1.524000,0.0,1.5708,8.033e-1,1.5708,1.5708
    polar:9.542e-4,5.204000,0.0,1.5708,4.367e-1,1.5708,1.5708
}

@RING(1000){
    -- options: r_inner, width, e_max, height (b/a for ellipse),
    -- m_avg, m_sig, m_orb
    r_inner:2.1
    width:1.1
    e_max:0.05
    height:0.6
    m_avg:1.202e-15
    m_sig:5.0e-16
    m_orb:1.0
    GEN
}
```

In addition, the most commonly used settings were provided in this script:

```
Simulator parameters.
@SIM{
    softening:0.0001
    stepsize:0.005
    bh_mode:1
    steps:5000
}
I/O parameters.
@IO{
    file_in:./data/gen_test/out.csv
    file_out:none
    dir_out:./data/test/
    console:1
    out_freq:10
}
```

The solar system with a 1,000 point asteroid belt was made. Running it with a 0.005 step-size for 5,000 steps with an output frequency of 10 and softening parameter of 0.0001, a stable asteroid belt was achieved.

In terms of initial conditions, these proved successful in creating a stable environment that, in terms of visuals, appears to approximate the solar system well for simple generation methods. However, given the rough number density of about $n = 2.0 \times 10^{-16}\text{m}^{-3}$, there is an average initial separation of $r = 1.7 \times 10^{8}\text{m}$, meaning that two asteroids will provide an initial acceleration of only $a = 9.0 \times 10^{-12}\text{m/s}^2$. Though cumulative, this separation would produce only minuscule changes in velocity. For a proper interaction, such as a 1% change in an asteroid's velocity, the acceleration would have to be roughly 600m/s^2. This can be achieved an encounter of 21m - given the initial separation, this is exceedingly rare in this setup.

To increase interactions, there are two options: increase the average particle's mass to increase acceleration induced, or increase the number density of particles. The first case is simple, where one must simply increase the mass. This is done by a factor of 10^6 in the test case. The second choice is more complex given the computational demands of an increase in particles. However, given the mutable nature of θ_{crit}, it was increased to 1.0 to facilitate the simulation of 10,000 particles, at which point the issue was less running the simulation but instead plotting it, among other issues.

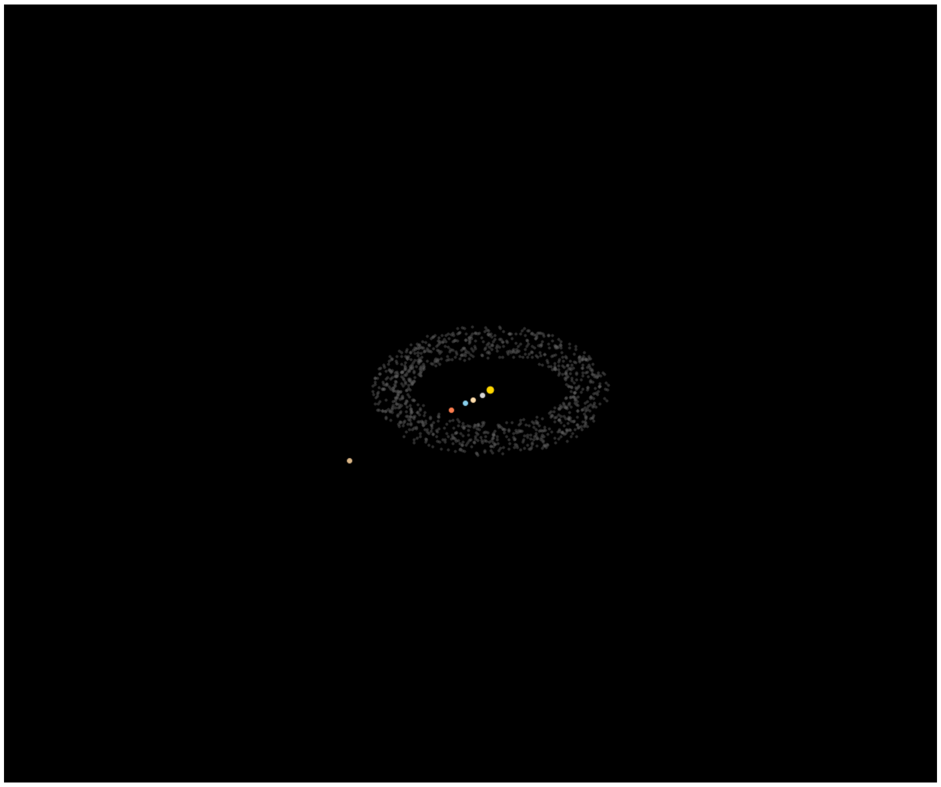

Figure 1: The stable 1,000 point asteroid belt.

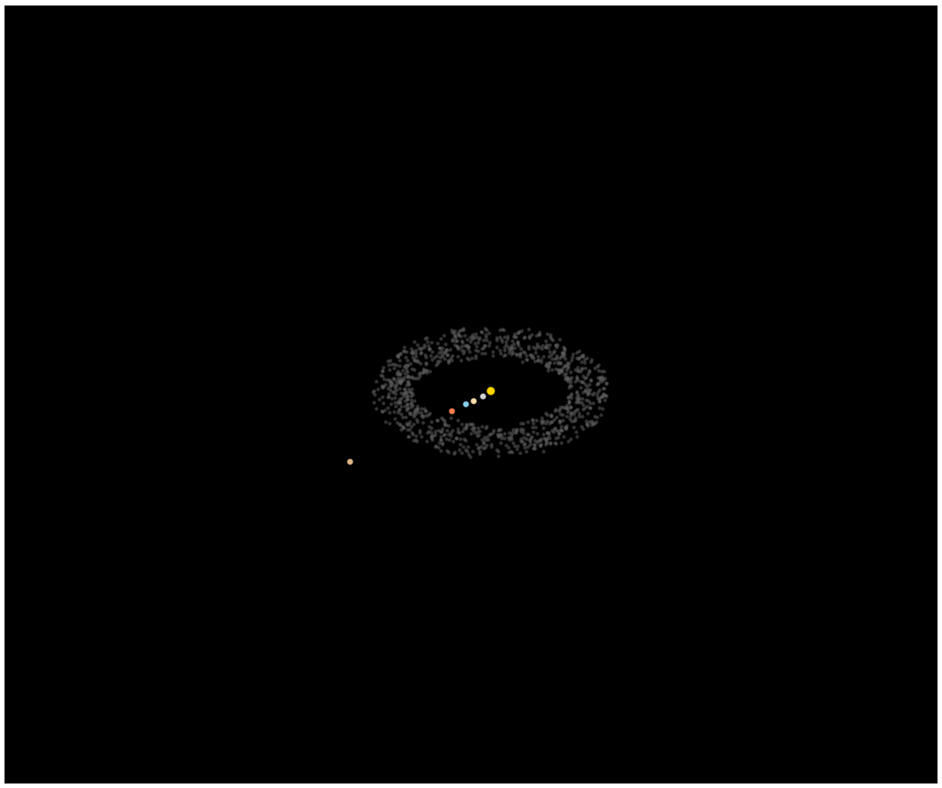

Figure 2: The stable 1,000 point asteroid belt with increased particle mass.

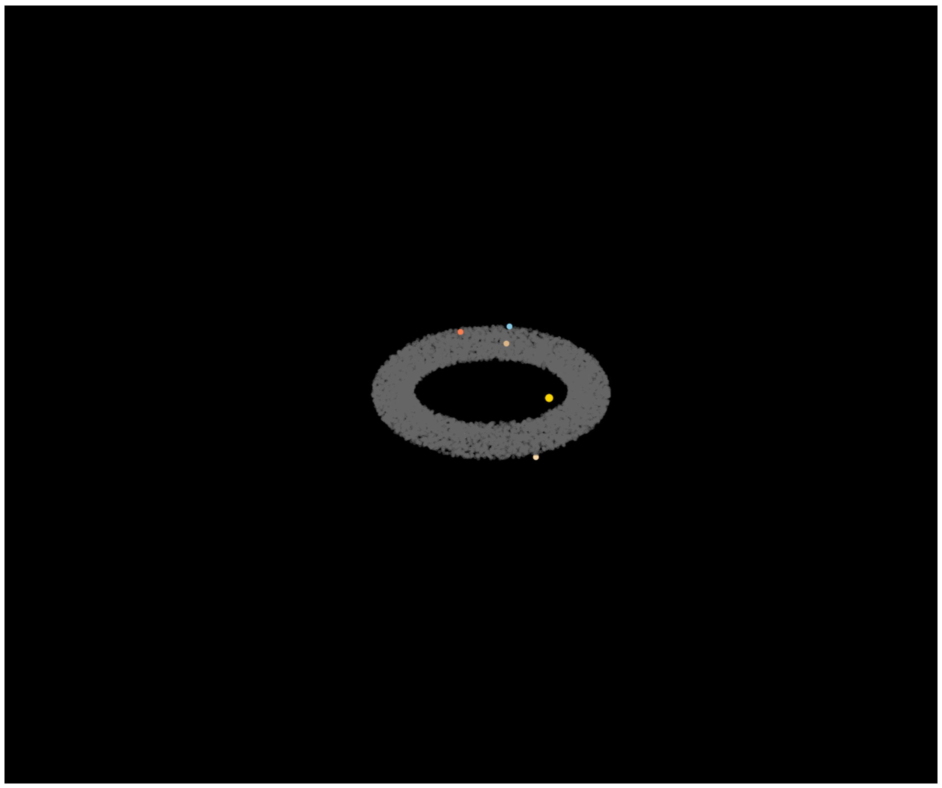

Figure 3: The unstable 10,000 point asteroid belt.

As seen in Figure 3, it is immediately clear there is an issue in the simulation. After this frame, the asteroid belt is torn apart, with planets drifting outwards and the Sun itself ascending. The initial conditions file seems to be valid, though the simulation itself is not, leaving the exact cause unknown, but likely related to the lack of softening parameter in this run leading to an explosion in force of some asteroids as well as the high approximation of calculations. More analysis was conducted on the three cases, but due to the obvious invalidity of the third case, it is excluded from further consideration.

In terms of the analytical properties of these simulations, we have plotted the maximum and minimum orbital radii of objects in the asteroid belt, the total angular momentum of the various objects of the system as well as specific angular momentum, and the location of the system's barycenter. Since the high density case was seen to be unstable already by visual inspection, it is excluded from further analysis.

In the two plots of Figure 4, the expected aphelion and perihelion from the initial conditions of eccentricity and orbital radius are plotted alongside the maximum and minimum orbital radii of asteroids. In the first lightweight case, it is seen that some asteroids exceed both the upper and lower limits expected, indicating interaction between asteroids or with the planets of the solar system despite the lack of dramatic interactions within the belt. In the second case, the plot looks similar, though no asteroid manages to exceed the aphelion while some pass the perihelion.

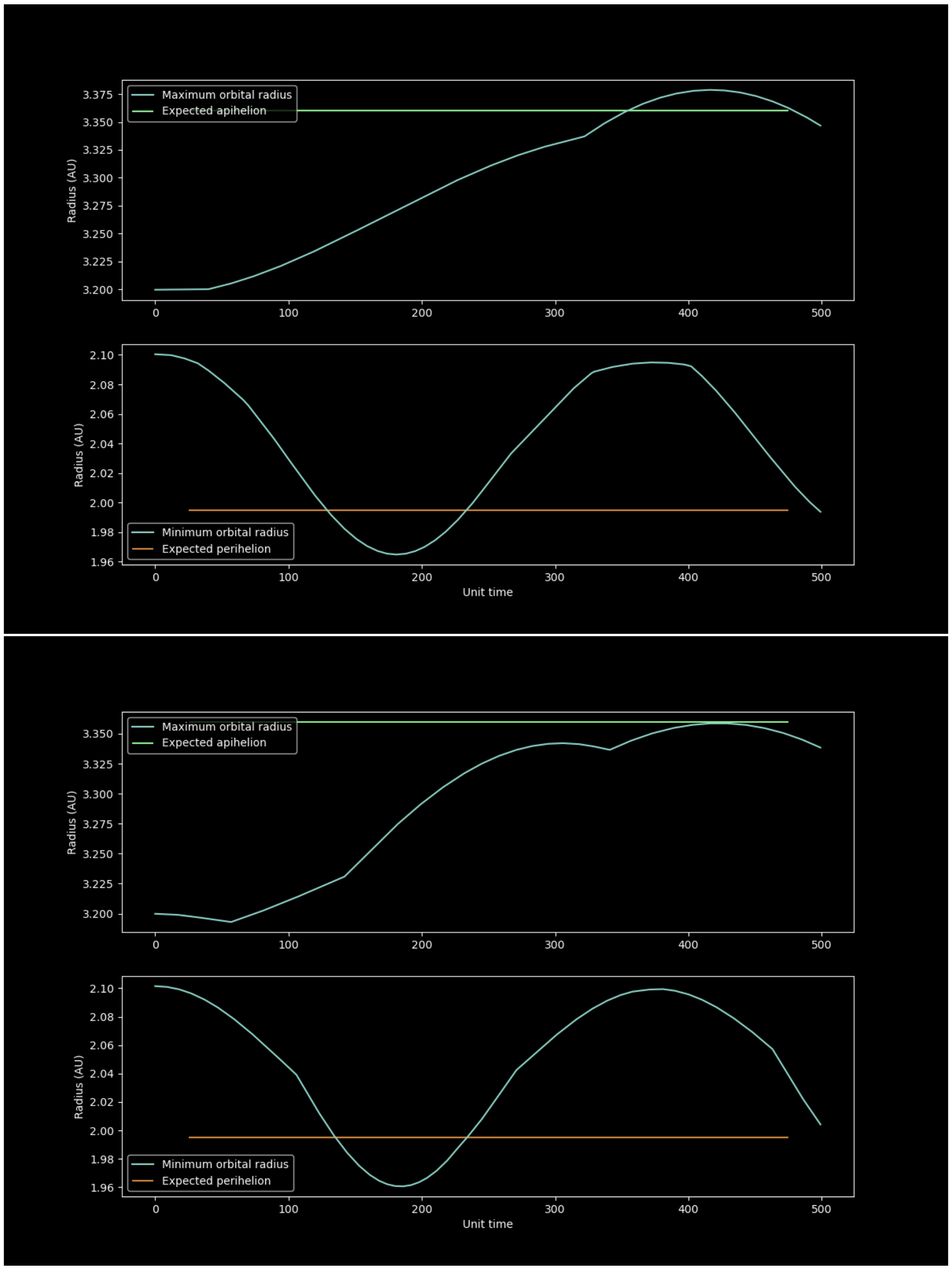

Figure 4: The orbital radius plots of the first (top) and high mass (bottom) cases.

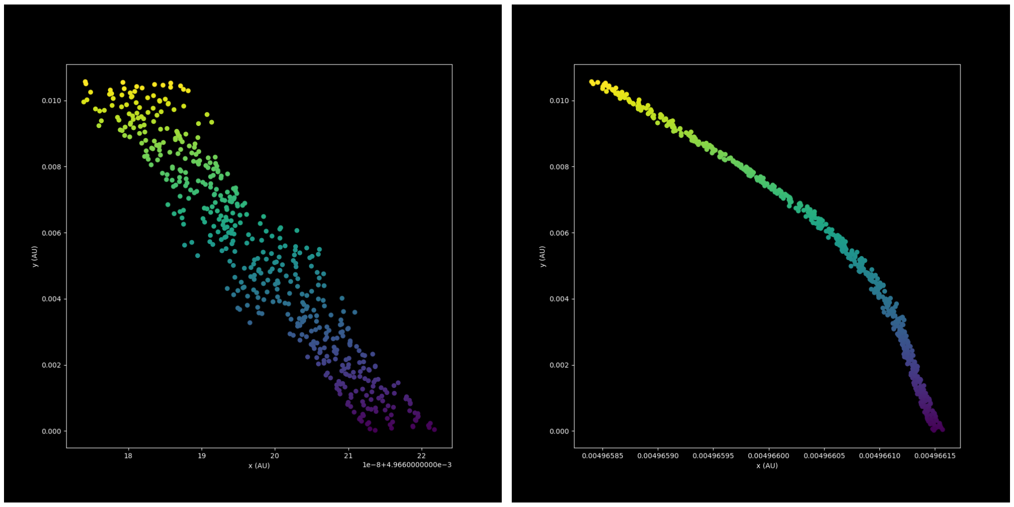

Figure 5: The precession of the system's barycenter in the first (left) and high mass (right) cases. Purple indicates the start of the simulation, and yellow the end.

Figure 5 demonstrates the precession of the systems' barycenters. Since the initial conditions included the velocities of all the Sun's satellites but the Sun was defined as a non-moving center, each system has a net momentum. This is seen in both cases as it moves across the x-y plane. The high mass case reduces the scatter of the precession, as well as adding in a curve that seems to indicate a change in momentum of the system that is unexplained.

Figures 6 and 7 end up showing a similar behavior to each other at different magnitudes. In the angular momentum (L) plot, with the y-scale logarithmic and the zero point shows by the white line, Jupiter dominates the system's L with the Sun quickly gaining much on its own due to its mass and movement that begins after the simulation starts. The other planets and asteroid belt are seen to oscillate closer to the zero point, and the downturn of Jupiter and the Sun's momentums near the end of the plot suggest they might have some oscillatory behavior as well if the simulation was set to run for longer.

The L/m plot shows that the asteroid belt dominates the specific angular momentum. This is not surprise as L/m for the belt was calculated as a sum of point masses, so their value would accumulate much more than any other single body in the solar system's. Once again, it demonstrates oscillatory behavior as the L plot did.

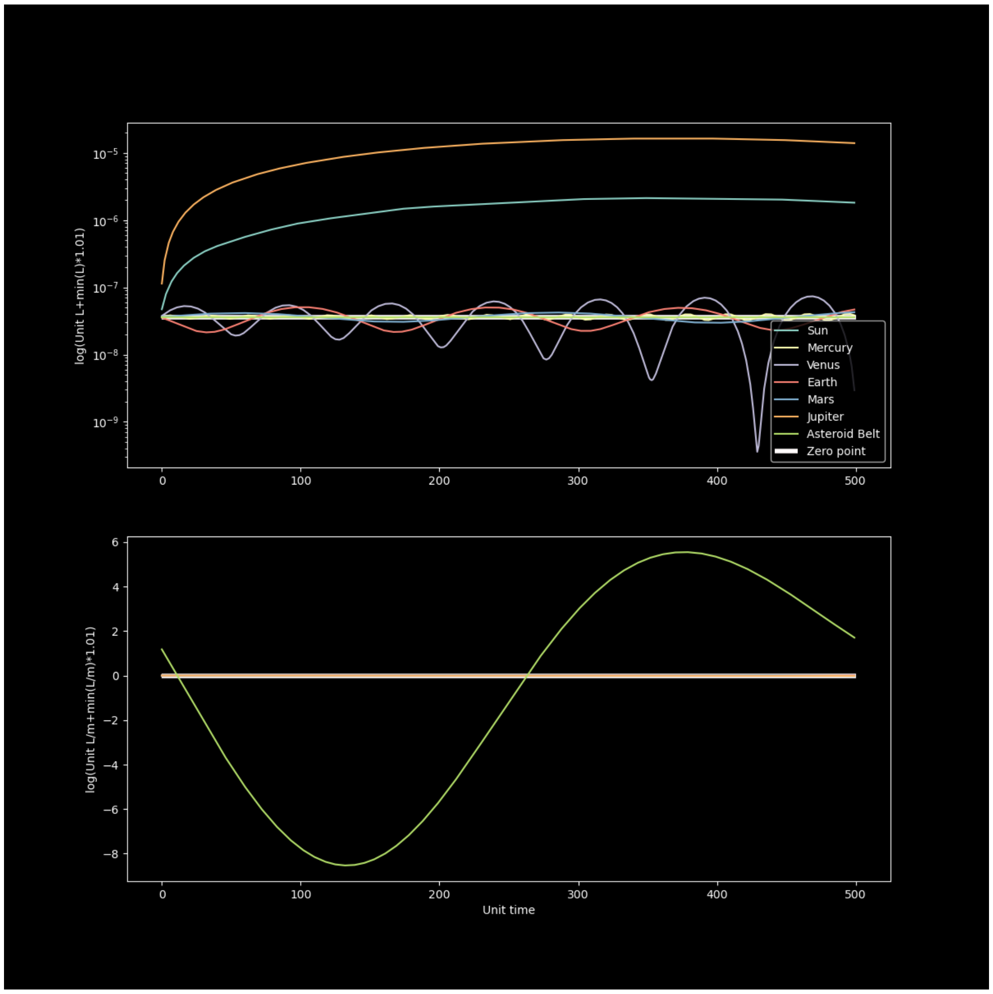

Figure 6: The angular momentum and specific angular momentum of the system in the first case.

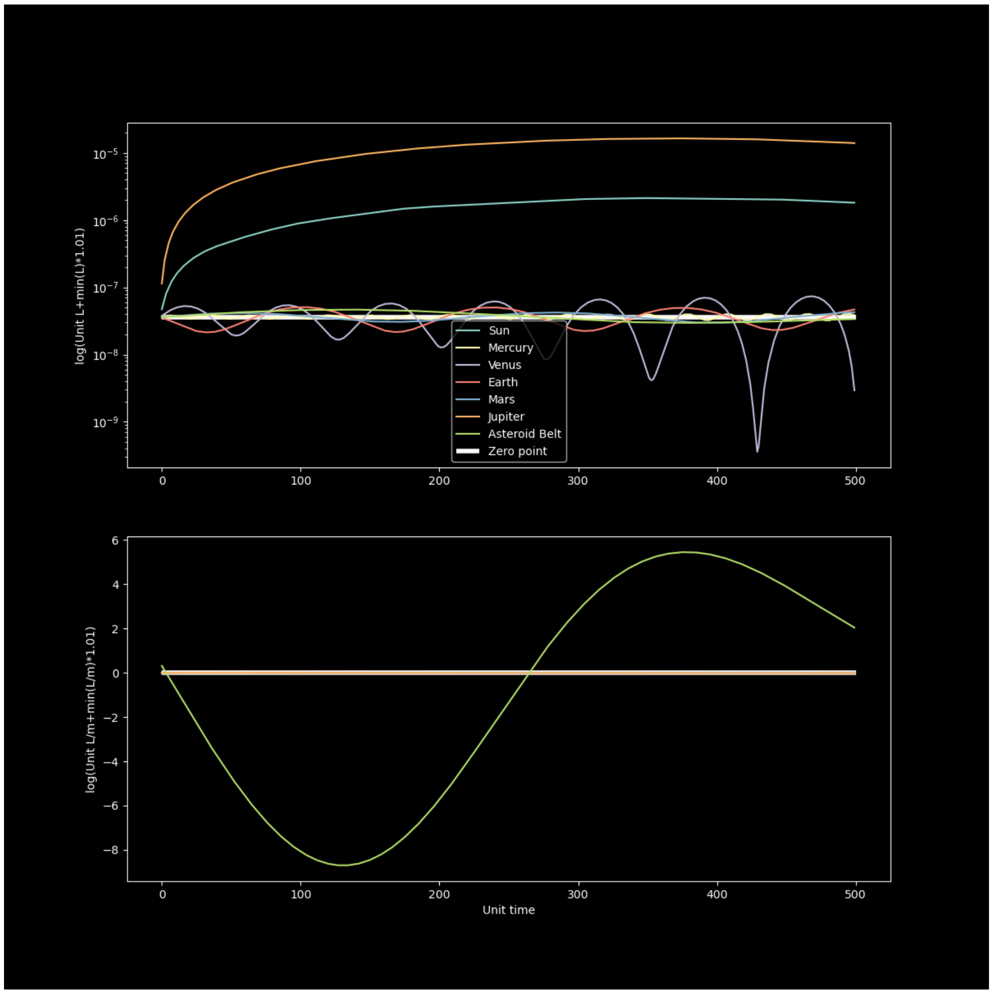

Figure 7: The angular momentum and specific angular momentum of the system in the high mass case.

As the original plan for this code was to reconstruct shepherd moons in Saturn's rings, one last case was tested (though not analytically analyzed) in which Jupiter found itself within a second asteroid belt. This was run to see how the planet would, if it all, clear its own orbit.

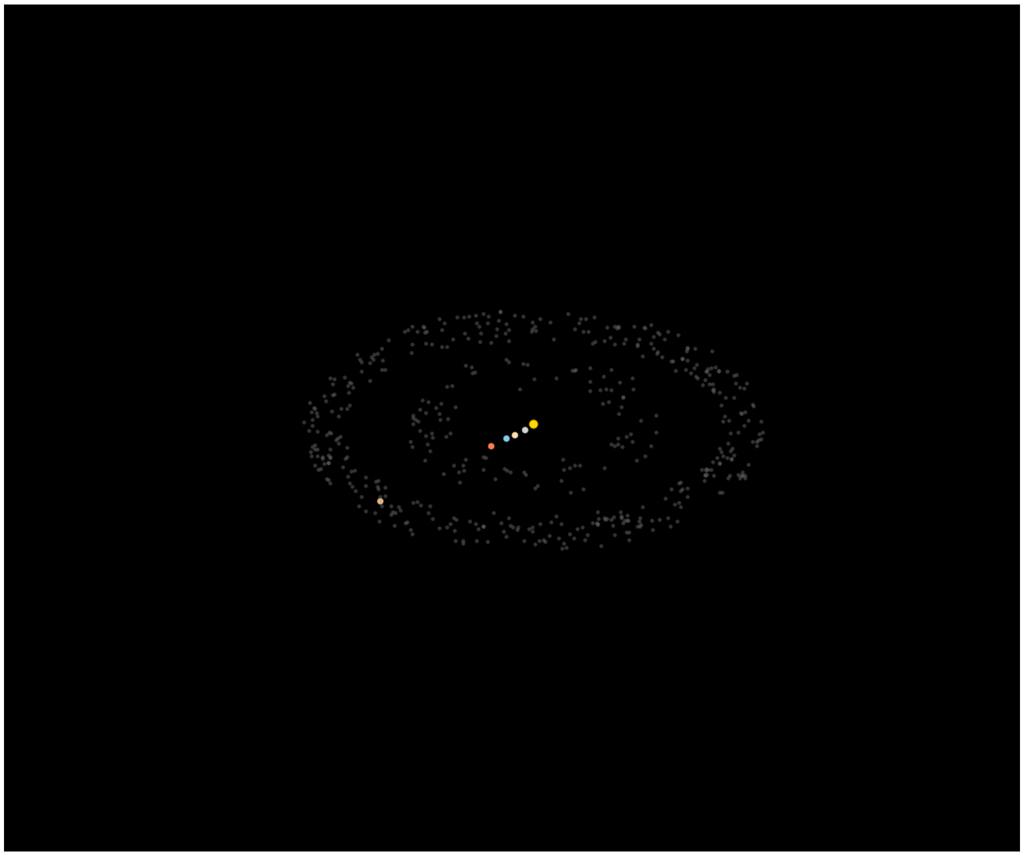

Figure 8: The beginning of the Jupiter belt system.

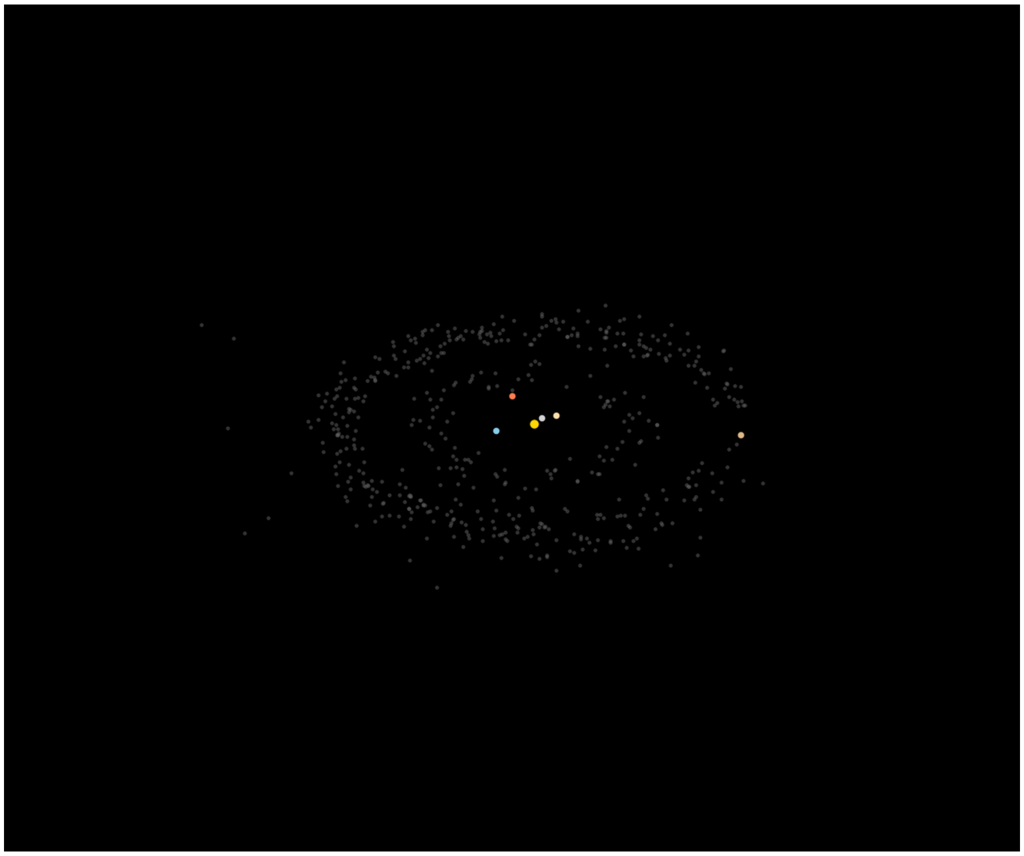

Figure 9: The Jupiter belt system after 4,000 years of simulation.

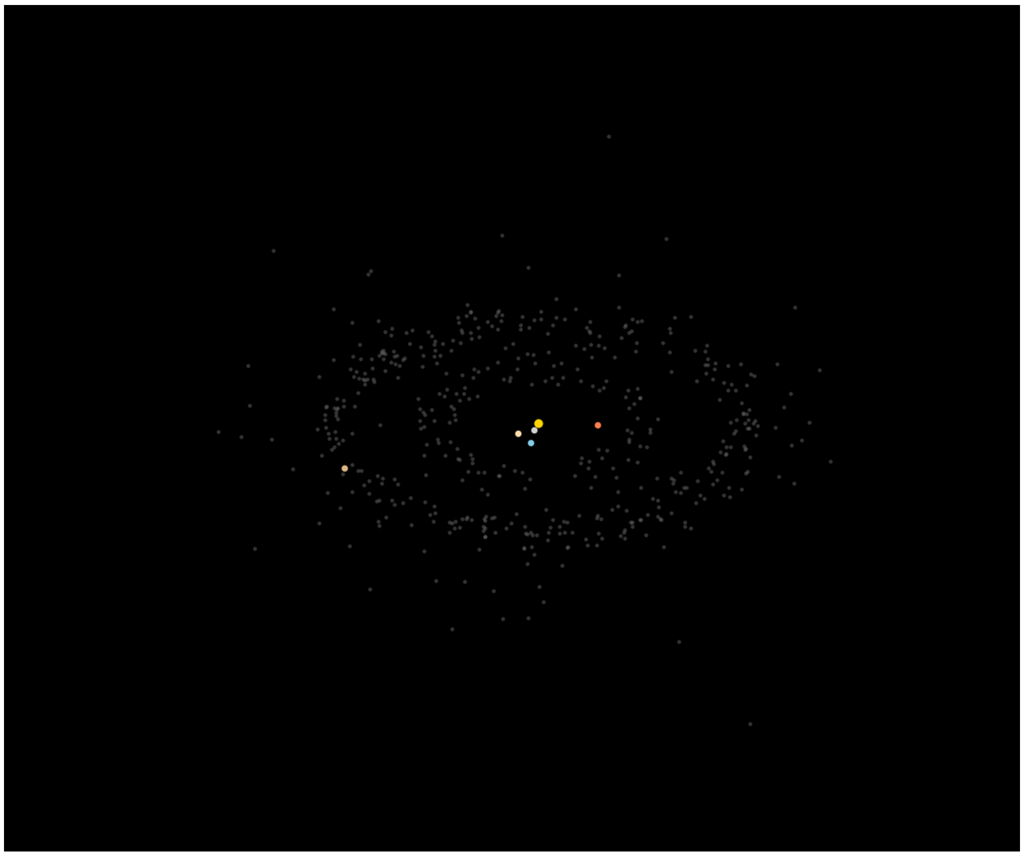
Figure 10: The final evolution of the Jupiter belt system after 44,000 years of simulation from the beginning.

Figure 8 shows this system at its beginning with Jupiter within a second asteroid belt. By Figure 9, 4,000 years after the beginning of the simulation, it appears that Jupiter has begun to create a gap in the belt around itself and the both belts have began to be less well defined and spread across a wider region. Finally, by Figure 10, the gap around Jupiter is less well defined, but the spread of the two belts is much greater than before. As this is the final simulation, further states are only speculation, but it seems that given a much longer time period, Jupiter would indeed clear its own orbit and disrupt the inner (real) asteroid belt in the process.

3 Discussion and Conclusions

The Barnes-Hut tree code took time and effort to implement, but appears to have worked in the end as the final solutions were stable (except for the one high density case, where the critical angle may have been relaxed too dramatically) and fairly accurate. An energy plot of the system revealed that the total value varied little, though it was not as stable over time as in the previous homework, which may be due to the Barnes-Hut method not conserving energy perfectly.

This code could be further optimized by adding in parallelization, allowing for even more complex and efficient simulations. However, by the end, the issue was not the simulation itself but the time to plot the simulation. In all cases, the matplotlib library was used, and it appears that the scatter plot method is too inefficient for data of 1,000 points or more to be feasible, especially when animated with several hundred frames. Each 500 frame plot of 1,006 points would take roughly two hours to complete, and the 80 frame high-density plot of 10,006 points took closer to five hours. In comparison, the simulation was running much more efficiently than the previous homework, with a speed of (at $\theta_{crit} = 0.05$) roughly two steps per second, meaning that a 500 frame simulation (where each frame was after 10 steps) would take only roughly 40 minutes.

The Barnes-Hut codes seems to work well for simulating high numbers of particles, though the approximation angle must be selected carefully for results to be feasible and stable. This further proves that an N-body simulation can be used to simulate complex systems such as our solar system with high accuracy and that there exist methods for approximating the simulation in order to highly increase its efficiency.

For some final details of the system, the time step used was at first $h = 0.005$ and later $h = 0.01$ to double the time of the simulation. Either time step produced a stable orbit for the asteroids with reasonable behavior, and perhaps even higher time steps could be used though they were not tested. For these simulations, the first used a softening parameter of $\eta = 0.0001$, but this was removed in later simulations as it was apparent that on the scales of the planets and asteroids, there were rarely interactions that came even close to testing the limits of the force calculation minus an exceedingly rare example of asteroids passing Jupiter that realistically should have been torn apart by or collided with the planet. Lastly, there were no specific time calculations for the simulation, but anecdotally, it did indeed seem to scale as $O(N) = N\log(N)$ as the number of particles seemed to only increase the time of calculation by a roughly linear amount, and definitely well below the scale of $O(N) = N^2$.