

I worked with no one on this worksheet. It took about four hours.

Direct summation n-body code

First I familiarized myself with the code structure and the layout of the state vector. I then completed the code by implementing `NbodyRHS` and `NbodyRK4`. I then simulated the sun-earth system for two years, during which time the Earth's orbital radius increases slightly, going from 1.50×10^{13} cm to 1.54×10^{13} cm, an increase of 3%. Increasing the number of time-steps by a factor of four leads to a decrease in this increase by a factor of four, indicating that this is a first-order effect. In my algorithm, during any time-step the position is updated by the velocity of the previous step and the velocity is updated by the acceleration of the previous step, which means that it requires two time steps for the acceleration to propagate through to the position. This means that the force vector is likely lagging behind the radial vector, allowing gravity to do work to the initially-circular orbit. This changes the energy of the orbit, leading to a change in orbital radius.

After implementing `TotalEnergy`, we can inspect this change in energy over the course of the simulation, comparing two simulations with different time steps. The result is in Figure 1. We see that the change in energy is approximately linear with respect to time, and for a given time period the simulation with 4×10^4 time steps has its energy increase by a factor of $\frac{1}{4}$ times the energy increase in the simulation with 1×10^4 time steps. This indicates that the energy growth is a first-order in time effect. Though it is not graphed, the same is implemented with Forward Euler and the energy graph looks identical, suggesting that this energy growth is independent of the integration scheme. Again, I think the lagging of the position-update relative to the velocity-update mentioned above is a likely candidate for this energy change.

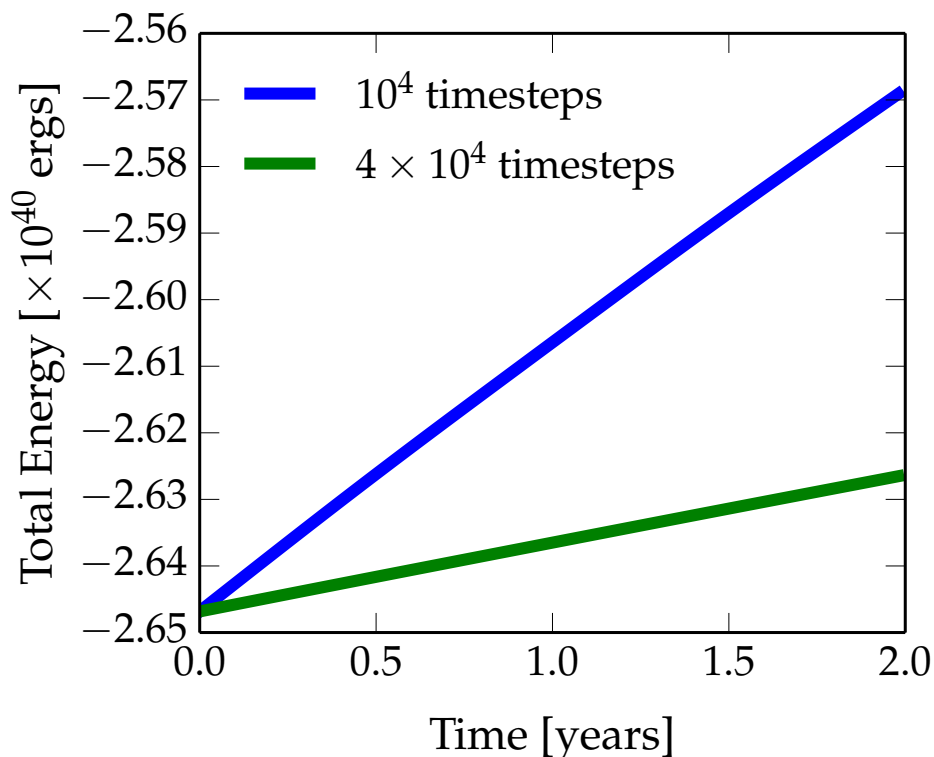


Figure 1: Total mechanical energy of the sun-earth orbital system as a function of simulated time.

I then used the same routine to evolve the system of stars with initial data in `sgrAstar.dat` for 100

years. The total energy of the system evolves as in Figure 2. As can be seen, the energy quickly shoots from negative to positive, then holds constant.

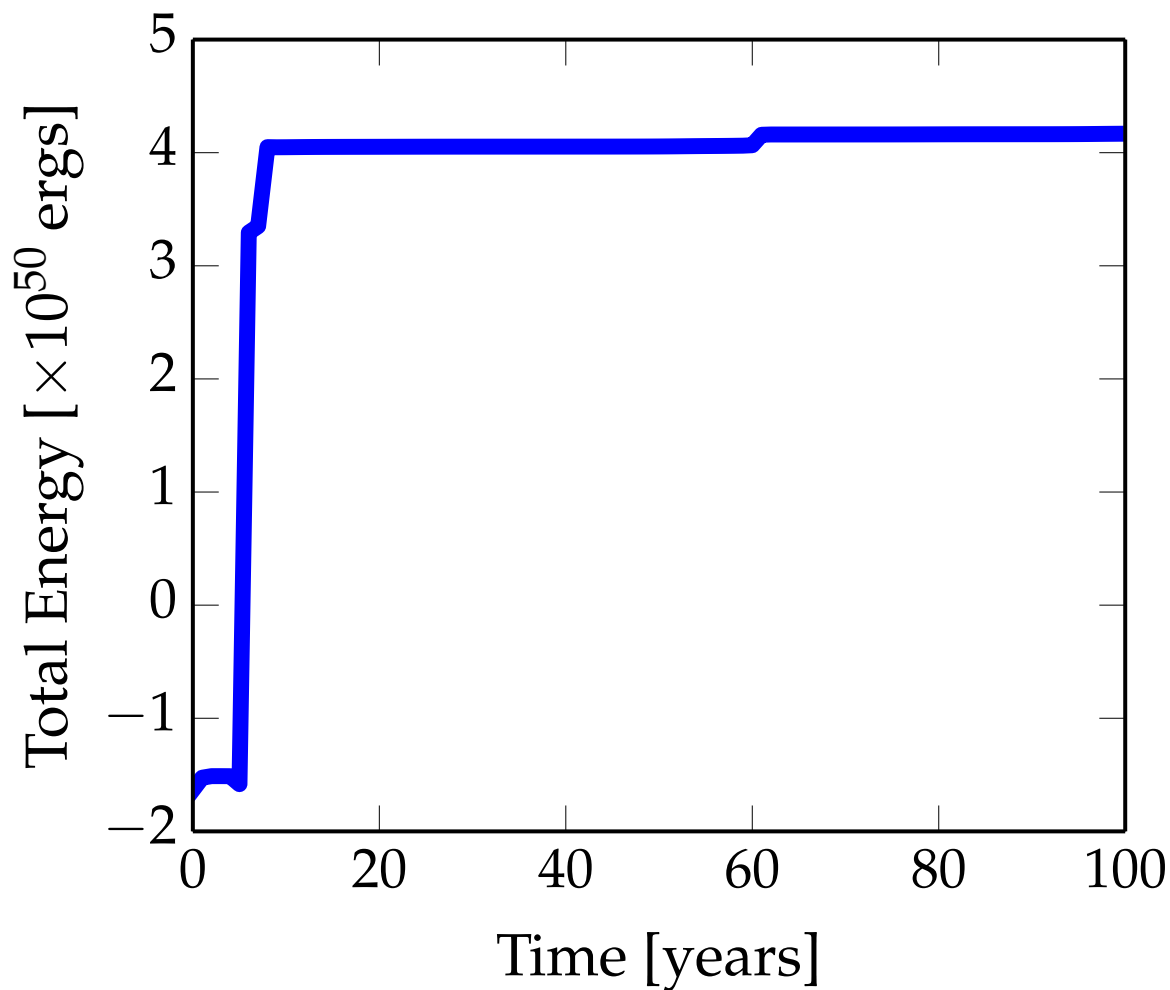


Figure 2: The total mechanical energy for the system of stars in `sgrAstar.dat` versus simulated time. 1×10^4 time steps were used.

We find that the final position of the first star in our simulation is completely different from that reported by Andrea Ghaz and Jessica Lu.