Assignment 3: Hybrid star cluster simulation

Bieker, Jacob Natoewal, Dylan Oberhelman, Lindsey

March 2019

1 Introduction

Star clusters consist of a high number of gravitationally bound stars with a variety of stellar masses. As the dynamical time of such clusters is large ($\simeq$ 4.7 Myr for the Milky Way Galaxy) the common way to study their dynamics is to run simulations of theoretical star clusters using gravity integrators. In this project we study a theoretical star cluster in which the stars are spatially distributed in a virialized Plummer sphere and have a mass function from 0.1 - 100 M_{sun}.

For the simulation we combine the efficiency of the tree-code (TC) with the accuracy of the direct N-body code (DC). The definition of a low mass star is 0.08 - 0.5 M_{sun} and an intermediate mass star is between 0.5 - 10 M_{sun} and high mass star being everything above 10 M_{sun} [**LeBlanc**]. In our project, the stars are divided into two populations by a mass cut parameter M_{cut}, Low mass ($< M_{cut}$) stars dynamical calculations are done by a tree code integrator and for the the higher mass ($>= M_{cut}$) stars a direct code is used. The calculations of both codes are preformed using a Bridge between them. Using the literature we decided to explore mass cuts around 0.1 - 10.0M_{sun} as cuts for determining low mass and high mass stars.

In addition, we make more comparisons and explored mass cuts of 40-95M_{sun} to test the functionality of the code and our understanding of the physics.

2 Discussion 1

We split the particle set by mass because higher mass stars affect the system more than low mass stars and thus require more accuracy than low mass stars. The tree-code is less accurate but more efficient so you can use it when accuracy is not as important which is why we use it for the low mass stars.

3 Discussion 2

To calculate the dynamical time of our theoretical cluster we used Equation 2.2 from the AMUSE handbook. Inserted with a virial radius of 3 parsecs and a total system mass of $\simeq 7 \times 10^3$ M_{sun} it gives a dynamical time of $\simeq$1 Myr. For optimal outcomes the Bridge time step should be some order of magnitude smaller than 1 Myr. We use a Bridge time step of 0.01 Myr which is about two orders of magnitude smaller than the dynamical timescale. So for every 10 time steps ($\equiv$ 0.1 Myr) taken in the simulations the bridge will stop and save the data and send it back.

4 Discussion 3

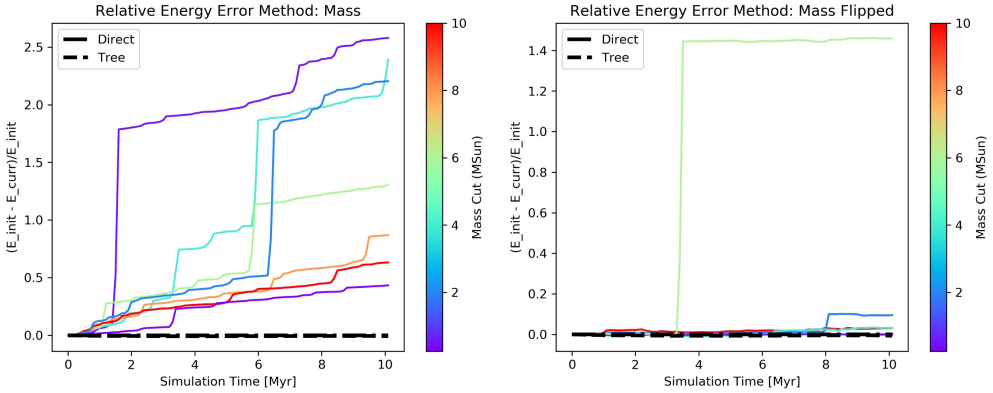

Figure 1: Relative energy error over time: Particles split according to mass. Multiple simulations for various mass cuts are shown (color coded according to the mass cut). Simulations for all particles in the direct code and the tree code are shown with black hashed and dot-hashed lines respectively.

Since we do not include stellar evolution the total energy of the system should stay constant. Figure 1 shows that the tree-code and direct code by themselves calculate the energies fairly well (with little to no error). It is only when they work together with the Bridge that the errors increase. When we used a mass cut of 10 M_{sun} the Bridge code works with very little energy error, however when use a much smaller or much larger mass cut the error increases significantly with the Bridge code especially. When the integrators are flipped and the direct code is handling the low mass stars and the tree code is handling the high mass stars the relative errors are significantly smaller. This is partly because we expect more of the stars to be in the low mass section.

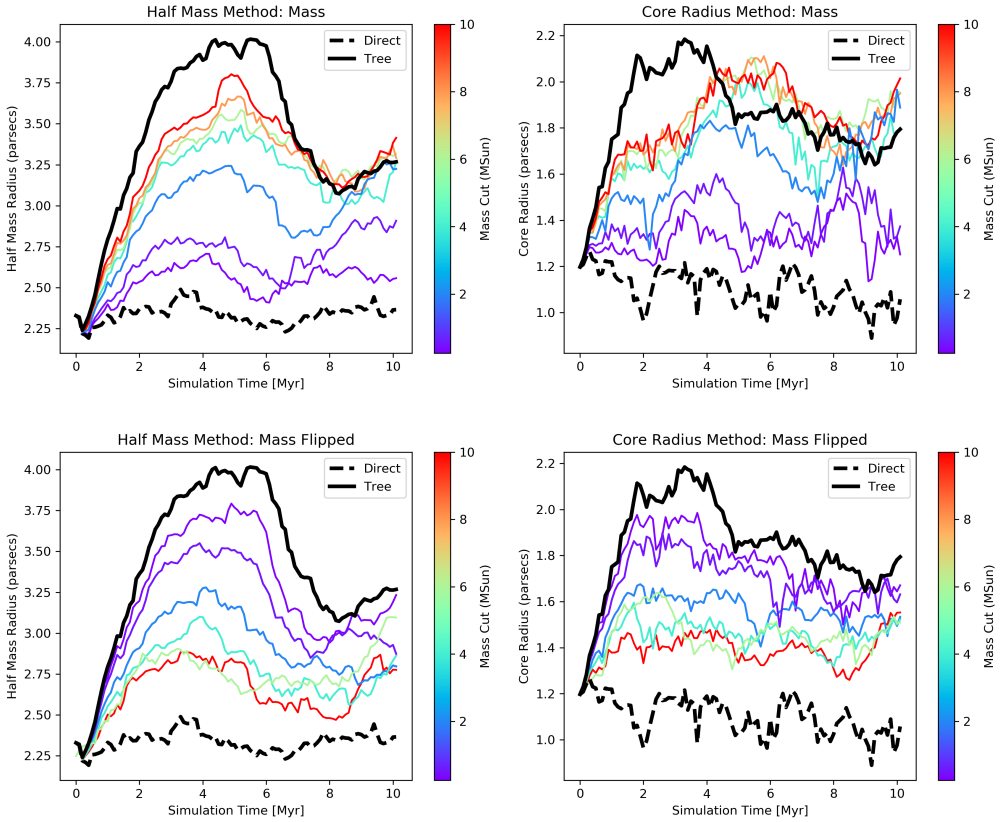

Figure 2: Half-Mass (left column) and Core (right column) radii over time. Particle set separated according to mass. Color scheme indicates mass cuts. Black solid lines: All particles in tree code. Black dashed lines: All particles in direct code. Particles with particle mass >= mass cut in direct code (top row); tree code (bottom row)

The core radius is defined according to [**King 1966**]. The half mass radius is the radius from the center that contains half the mass of the cluster. Figure 2 show that the mass cut of around $10M_{sun}$ (red line) is the ideal since it closest to the direct code result. Figures 2 also shows the half mass and core radii over time but where flipped so that the low mass stars are handled by the direct code and the high mass stars are handled by the tree code.

In Figure 3 it is shown that, when particles with particle mass above $6M_{sun}$ are put in tree code, the simulation final energy error compares to the case when particles with masses above the same mass cut are put into direct code. The simulation final energy error difference is most prominent at the mass split of

$4M_{sun}$ where if the higher mass particles are put in direct code the energy error is large and when the higher mass particles are put in TC the energy error is very small.

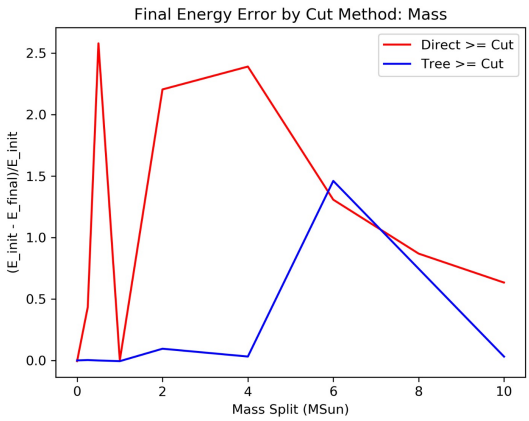

Figure 3: Energy error at simulation time = 10 Myr: Particle set split according to mass. Simulation for particles with particle masses above the mass cut put in the direct code (red) and for particles with particle mass above the mass cut put in the tree-code (blue).

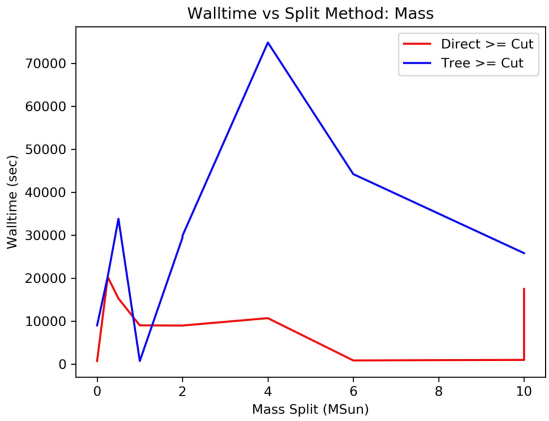

Figure 4: Wall-clock time for various mass split cuts. Particle set split according to mass. Color scheme is the same as in Figure 3

From this Figure 4 we confirm that the tree code is the fastest method however. That a mass cut of is relatively quick. The direct code does relatively well up

until about a couple thousand particles. The hybrid code is by far the slowest. This shows that the bridge adds a lot of overhead when saving the data so for short periods of time but for larger numbers of particles $\simeq$ 2000 the periods of time the overhead for the hybrid does not matter so much because the direct code takes much longer.

5 Discussion 4

Our initial assumption was that low mass stars with a tree code and high-mass stars with a direct code is advantageous. We found that when the number of stars simulated is over a couple of thousand the Bridge method of splitting the particles by a mass cut results in less error in the core radius and half mass radius than the tree code while being significantly faster than the entirely direct code evolved model. The time is takes to complete the simulation with the add time in the Bridge code becomes less then using the direct code to process all of the data and is more precise then the using the tree code to process all of the data. However we found that energy is not conserved when using this assumption the energy errors are quite large for the Bridge code results. It is important to note that this simulation does not include gas or stellar evolution and so it does not provide full physical accuracy. In sections 9 and 10 we talk discuss what we would expect were the simulation to have stellar evolution and gas.

5.0.1 Half Mass Core Radii

In Figures 2 there are fluctuations in the core radii and half mass, these variations are most likely due to just the orbit of the stars with the completely direct code result being the most accurate. Since there is no stellar evolution the fluctuations of the core radii an half mass is not realistic since none of the stars are losing mass so most likely these results are not as accurate to actual clusters in the Universe.

6 Error Propagation

7 Animation

In the evolution of dense star clusters the dominant factor is the escape of stars as a result of the cumulative effect of stellar encounters.[**King**] This could explain why in the we see some stars being ejected from the cluster. To help with the our understanding of what was happening in the star cluster over time,

we created 3D animations that could be used to easily qualitatively compare different cuts and splitting methods. A small selection is available in the Bitbucket repository. Three of the animations, titled with 'Compare Mass Cut' compare the pure direct, pure tree, and both flipped and non-flipped clusters split on a mass cut of 2.0 M_{sun}. As can be seen in the animations, the split clusters have a significant amount of stars flying out of the cluster quite quickly, many more than the pure tree code, and quite significantly more than the direct code, which has almost none. This seems to support our graphs, where the tree and hybrid gravity systems result in a much larger change to the half mass radius, core radius, and change in energy compared to the direct code.

The other animations in the Bitbucket, those titled with 'Compare Radii' look at the three different radius cuts performed, and explained below. This includes a cut on the virial radius of the initial cluster, half mass radius, and core radius. Similarly to the mass cut method animation, stars stream out from the hybrid gravity systems, but at a much lower rate than for the mass cut method.

8 Different Split Methods

In addition to splitting the particles by mass, we also attempted other methods of splitting the particles into a direct code evolved group and a tree code evolved group. For this part of the project, we split the particles into two populations based on multiples of the half mass radius, virial radius, and core radius. The reasoning for these cuts are not physical, but computational. The Bridge source code mentions that the Bridge code assumes that the two populations are well spatially separated if it is bridging two gravity codes. This comes into effect because the Bridge uses the leapfrog method to advance the individual codes. This means that for systems that are intermixed together, ignoring some of the particles closest to one another causes numerical errors to become apparent very quickly. This is why, for example, the large errors in energy only occur when cutting using the mass, but there is barely any error for the pure direct run or pure tree run.

As can be seen in Fig. 5, the relative energy error for the different radius splitting methods differs by quite a lot. While for all methods, the majority of cuts results in errors that are around 30% at most, there are some large outliers. In all cases, flipping which particles are sent to the direct code or tree code had the largest effect on the final relative energy error. When all particles farther away than the given radius were sent to the tree code, suddenly the relative error jumped up significantly, especially when the radius chosen was smaller. In contrast to the mass cut method, though, more of the radius cuts resulted in a very small (>10%) relative energy error. This is most likely because by choosing to cut radially, the two populations are more spatially separated than for the mass cuts, which is closer to the assumptions of the Bridge code.

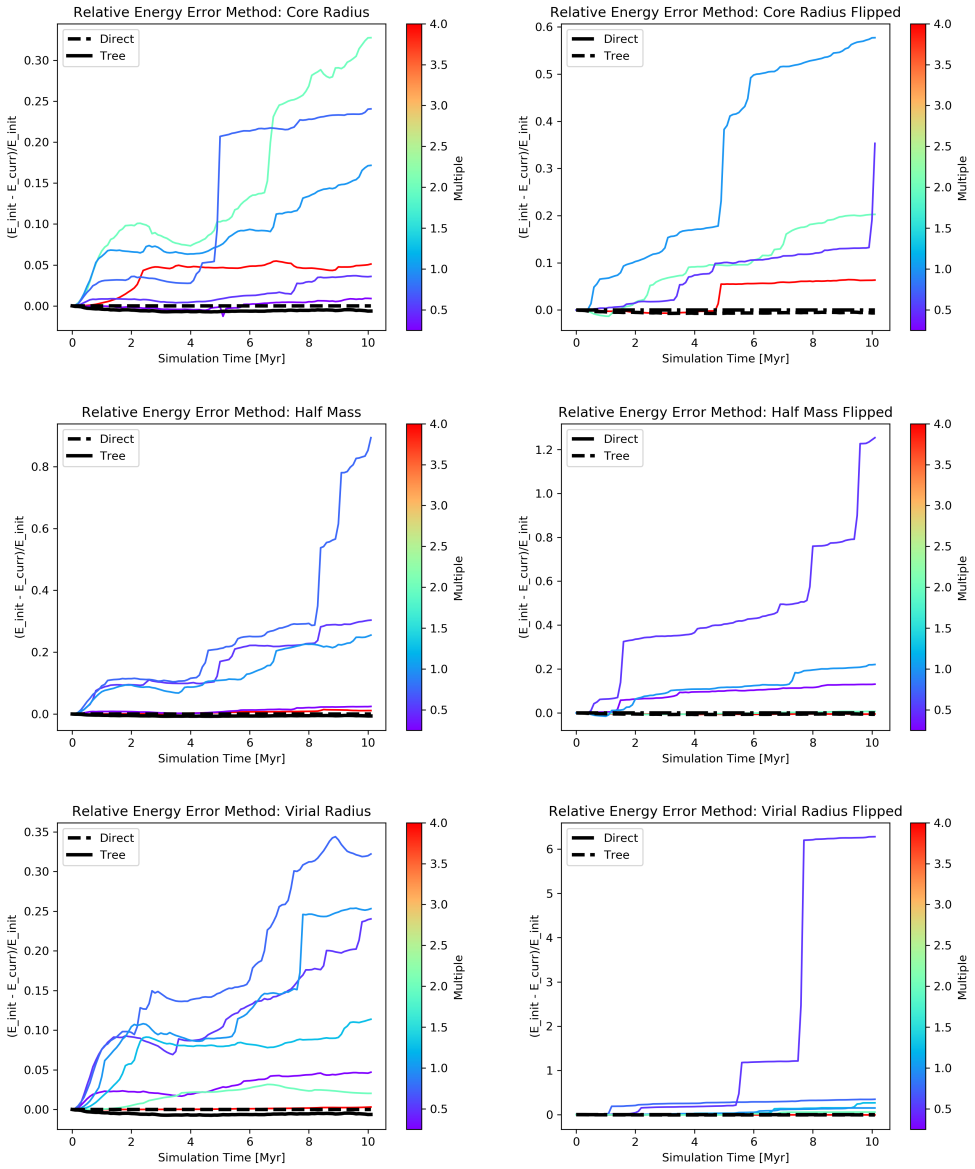

Figure 5: Relative Energy Error for various radius splitting methods. Flipped means that particles whose distance from the center of mass >= radius * multiple were evolved by the tree code, and all others by the direct code. Vice versa for the ones that do not have Flipped.

In Fig. 6, the final energy error for all four methods are compared. As can be seen in the plots, the minimum error for all methods is when only one code is

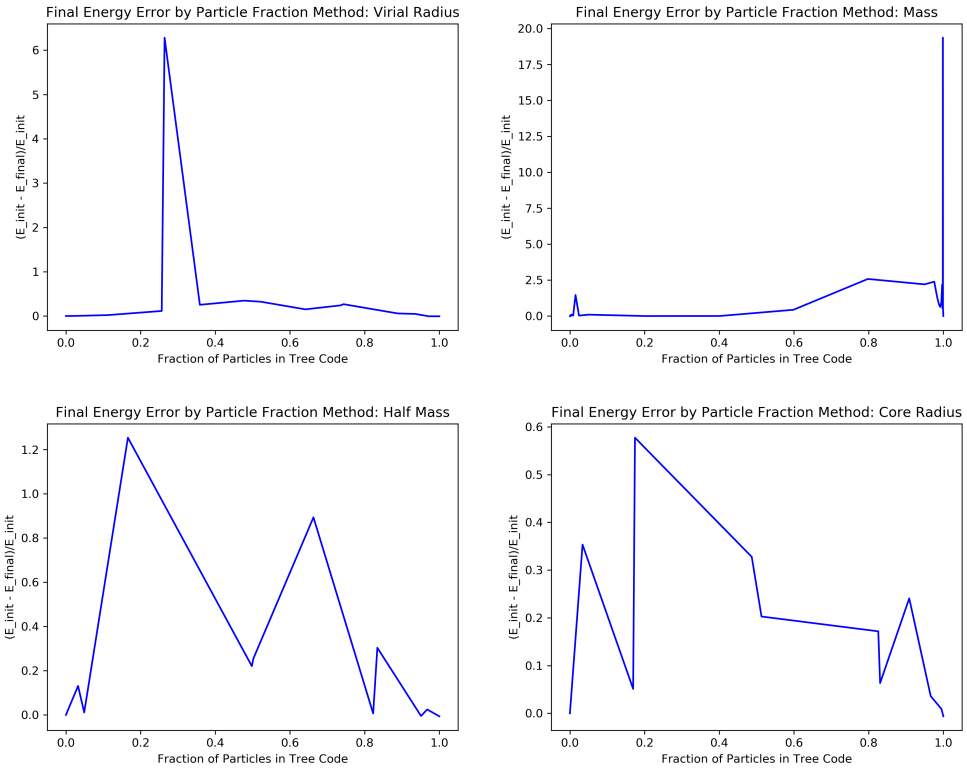

Figure 6: Final Energy Error vs the fraction of particles ran in the tree code for all four splitting methods.

used, either direct or tree code. The fraction of particles in the tree code seems to greatly affect the final energy error, with the three radial splitting methods having a peak in error at around 20% of the particles in the tree code, while the mass cut has a slowly increasing final error with the fractional amount, until nearly all the particles are in the tree code. Right before that occurs, there is a massive spike in the final error, a 2000% final energy error. While the cause is unknown, it is an intriguing difference compared to the other splitting methods, and would warrant further investigation.

Fig. 7 reveals interesting trends in the half mass radius of the different splitting methods over time. For the mass cut method, no run actually reaches the same stability as the direct code, while many different splits clump around the same results as the pure tree method. On the other hand, for the different radial methods, all have at least one splitting radius that has a similar graph as the pure direct code. For the radial splits based off the core radius and half mass of the starting cluster, the only cuts that are close are ones that are 4 times the respective radius. For the cut based off the virial radius, cuts of around 1.5, 2,

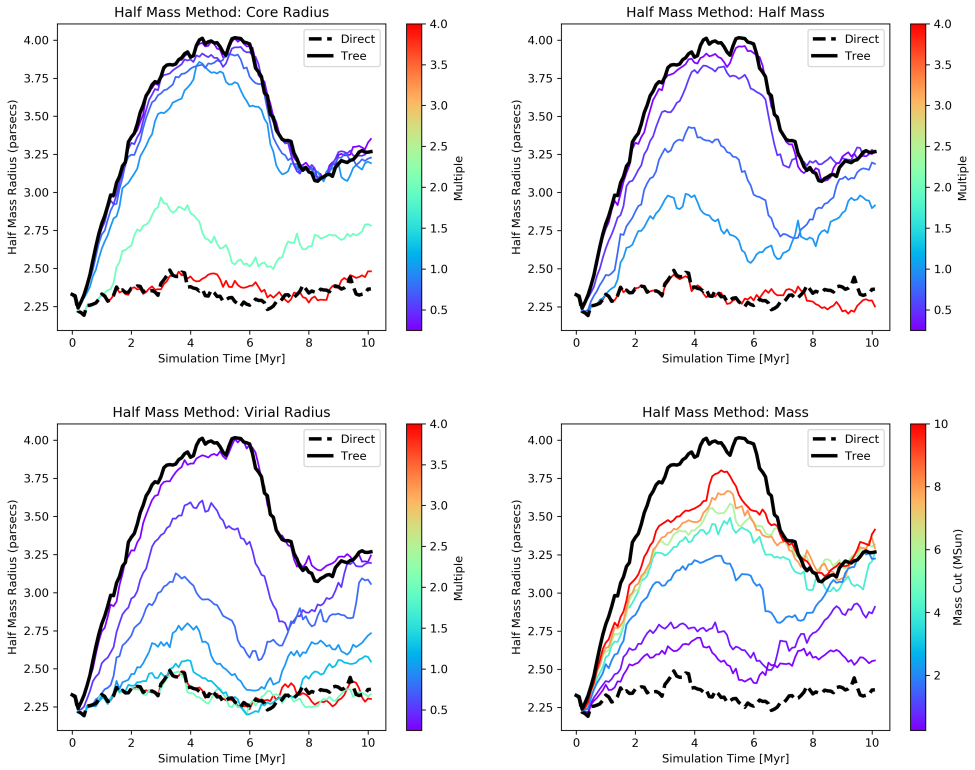

Figure 7: Half Mass Radius over time for all four splitting methods.

and 4 of the virial radius all result in similar graphs are the pure direct code. This seems to further back up the assertion that splitting on spatial distance is more effective than splitting based off the mass of the objects. Fig. 8 further backs up this assertion, generally replicating the same results as for the half mass radius.

9 Difference with Stellar Evolution

We did not add stellar evolution to this simulation. However if we had added stellar evolution we would have seen total energy change over time as the stars in the cluster lose mass and the high-mass stars die.

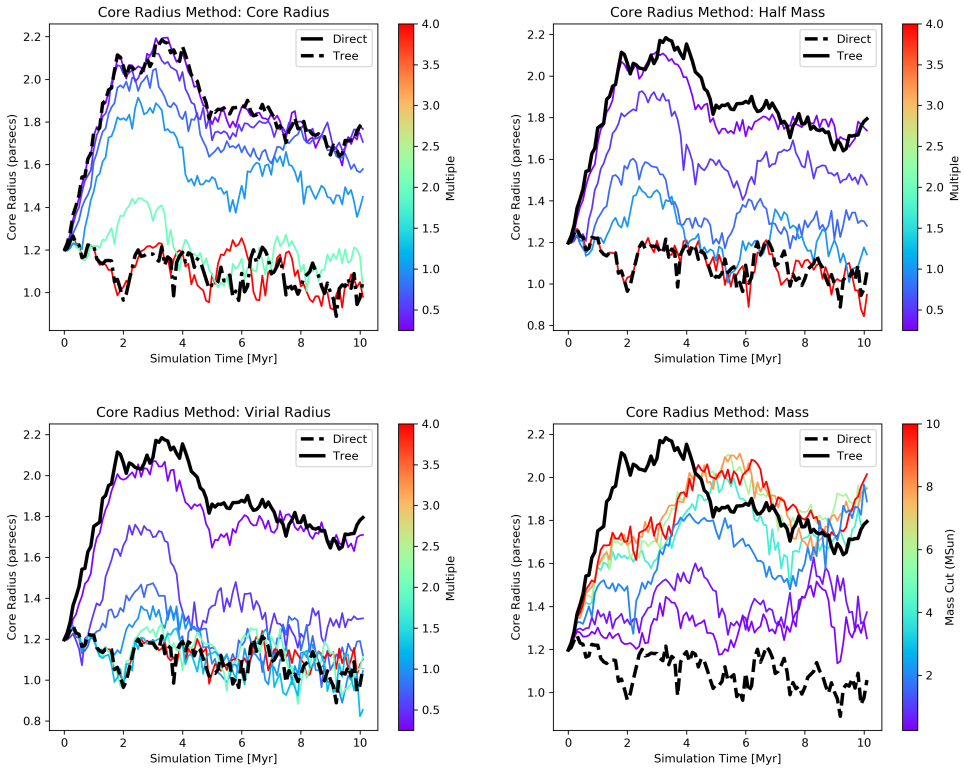

Figure 8: Core Radius over time for all four splitting methods.

10 Difference with Gas

Gas is not taken into account for our simulations. If we had inserted a gas component to the system it would have affected its evolution. Firstly because the gas can interact gravitationally with the stars in the system. This affects the motion and velocities of the stars. This would cause fluctuations in the gravitational potential energy of the cluster as a whole.

The gas also interacts thermodynamically with the stars in the cluster. As gas contracts it would expel energy in the form of heat. This changes the temperature and pressure profiles of the system which in turn affect its evolution.

Lastly, gas is a key component in star formation [**King 1958**] and over time clouds of gas can form stars. This means the number of stars in our simulated cluster would not be conserved. In addition, the formation of a star causes an

outward pressure which affects stars (and gas) in its vicinity.