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 (BHTree) and for the the higher mass ($>= M_{cut}$) stars a direct code (ph4) is used. This is exactly opposite for all figures indicated with "Flipped" in the figure title. 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

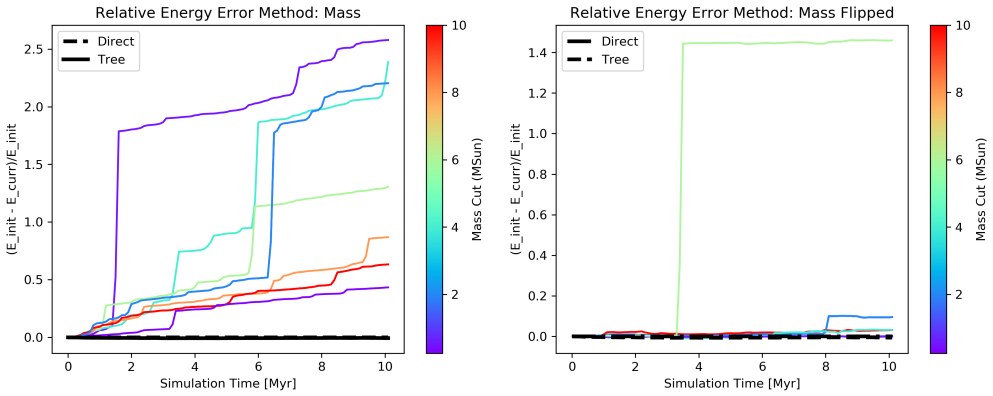

Figure 1: Relative energy error over time: Particles split according to mass. Multiple simulations for various mass cuts are shown (color coded according to the mass cut). Simulations for all particles in the direct code and the tree code are shown with black solid and dashed lines respectively (opposite for the Flipped case; right).

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 for almost all mass cuts. This is partly because we expect more of the stars to be in the low mass section.

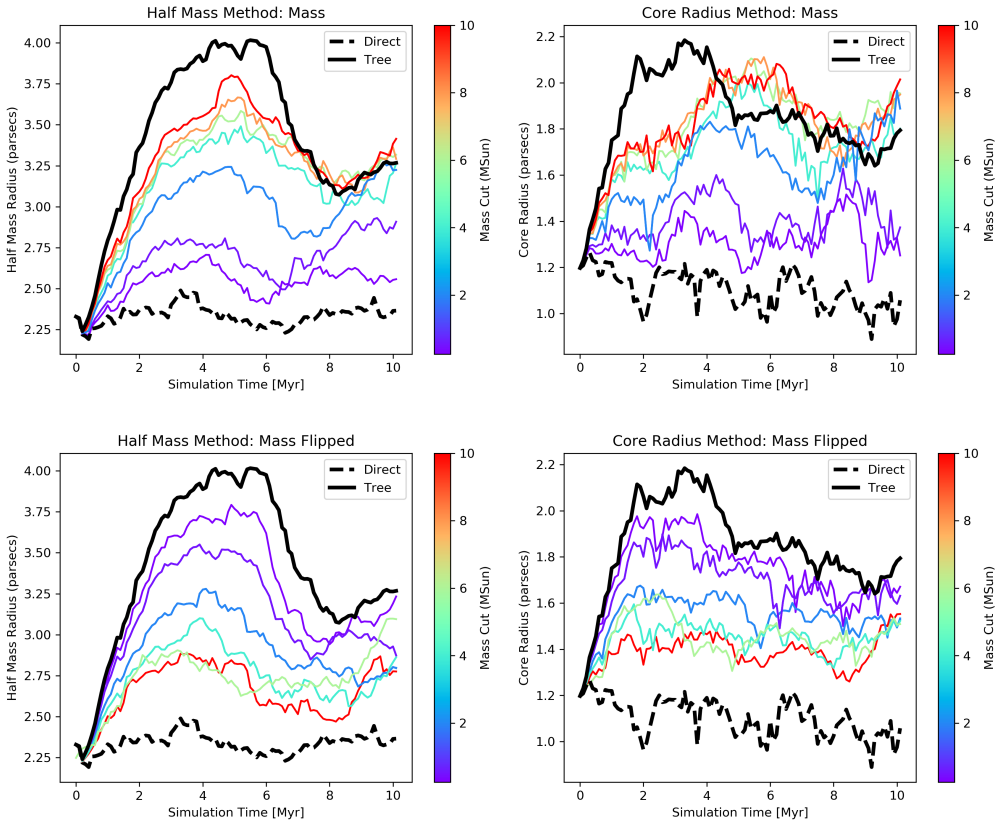

Figure 2: Half-Mass (left column) and Core (right column) radii over time. Particle set separated according to mass. Top: high mass particles inserted in direct code. Bottom: high mass particles inserted in tree-code (Flipped). Color scheme indicates mass cuts. Black solid line: All particles inserted in tree-code. Black dashed line: All particles inserted in tree-code

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.

4.1 Half-Mass radii (left column)

The "Half Mass" figures show the half-mass radius of the system simulated for various mass cuts and span 10 Myr. Both figures show a 1.75 parsec half-mass radius increase for the tree-code only simulation (black solid line) between 0-4 Myr. This decreases between 6-8 Myr. Both figures also show relatively little fluctuations for the direct code only simulation (black dashed line). For the top figure: higher mass cuts lead to a half-mass radius evolution similar to that of the tree-code simulation and lower mass cuts lead to the evolution to become more like the direct code only simulation. For the bottom figure (the Flipped case) this is opposite.

4.2 Core radii (right column)

The "Core Radius" figures show the core radius of the system simulated for various mass cuts and span 10 Myr. Both figures show a 0.9 parsec core radius increase between 0-2 Myr for the tree-code only simulation (black solid line). This decreases for the rest of the simulation. For the top figure: higher mass cuts lead to the core radius evolution to increase with heavy fluctuations. Some high mass cut simulations exceed the tree-code only simulation. For the bottom figure: higher mass cuts lead to the core radius evolution to resemble the direct code only simulation while lower mass cuts show a core radius more similar to the tree-code only simulation.

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 tree-code the energy error is very small. There is also a strong fluctuation in the energy error for the regular case (red) at around a mass split of $1M_{sun}$.

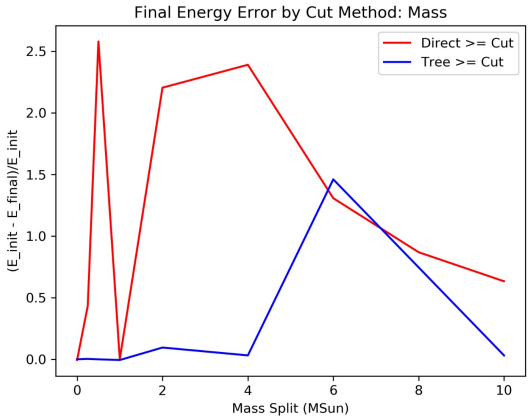

Figure 3: Energy error at simulation time = 10 Myr: Particle set split according to mass. Simulation for particles with particle masses above the mass cut put in the direct code (red) and for particles with particle mass above the mass cut put in the tree-code (blue).

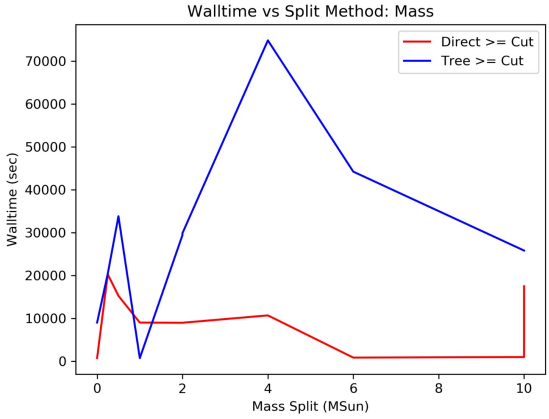

Figure 4: Wall-clock time for various mass split cuts. Particle set split according to mass. Color scheme is the same as in Figure 3

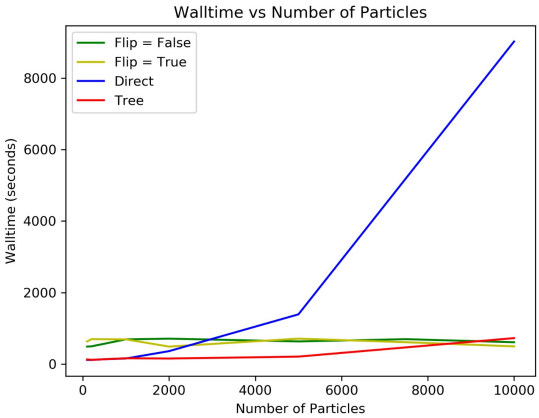

Figure 5: Wall-clock time as a function of the number of particles. Mass split is set to 5 M_{sun}.

From Figure 4 we see that the wall-clock time is largest when particles with masses above mass cuts of 1-10 M_{sun} are put into tree-code. Higher mass cuts (as can be seen in the utmost right part of Figure 4) lead to the tree-code wall-clock time declining and the direct code wall-clock time rising rapidly. In Fig. 5, we can see the N^2 behaviour of the only direct code, while at the same time seeing how the combined gravity and purely tree code runs have a much smaller increase in run time as the number of particles increases. This validates that the major reason for using the combined gravity instead of the purely direct code does occur, at least for large number of particles. For small amounts of particles, the overhead of the Bridge, such as the synchronization, seems to dominate the run time, but for larger amounts, over a few thousand particles, the combined gravity starts to compare favorably to the pure direct code.

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 (see

Figure 1). 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 Radius

In Figure 2 (left column) there are fluctuations in the half-mass radius simulations. We assume that the direct code results are the most accurate as the half-mass radius fluctuations should be small for 10 Myr. As the tree-code is less accurate for these particle numbers (compared to direct code) it can be seen from both left column figures in Figure 2 that when more particles are inserted into the tree-code, the half-mass radius evolution is increasingly less accurate between 0-8 Myr. This discrepancy between the tree-code only and direct code only half- mass radius simulations decreases after 8 Myr but does not disappear entirely.

5.0.2 Core Radius

In Figure 2 (light column) there are fluctuations in the half-mass radius simulations. We assume that the direct code results are the most accurate as the core radius fluctuations should be small for 10 Myr. From both right column figures in Figure 2 it can be seen that the simulation accuracy increases (i.e. becomes more like the direct code only simulation) when more particles are inserted into the direct code.

5.1 Error Propagation

Figure 3 shows the final energy error (= the energy error at the end of the simulation) for various mass cuts. For the regular case (red) it can be seen that the energy error f for mass cuts between 0-4 M_{sun}. This means that, for that region, the final energy of the of the system is is about double that of the initial system configuration. It should be noted that this final energy error is the result of an energy error increase (see Figure 1) and a higher resolution would provide more insight to the behavior of the final energy error as a function of mass cut. This is also the reason we see a single increase before (and decrease after) the peak of the Flipped final energy error (blue) at 6 M_{sun}. Figure 1 shows this blue peak in Figure 3 is most likely an anomaly rather than a physical outcome.

5.2 Calculation times

Figure 4 shows the wall-clock time (total time it took to run the simulation) for various mass cuts. It clearly shows that while similar for low mass cuts, the

regular case wall-clock time (red) is very much shorter than the Flipped case wall-clock time (blue). This indicates the direct code is faster than the tree-code between mass cuts 0-10 M_{sun}. For systems with high particle numbers we expect the tree-code to be more efficient. Our results show otherwise; the direct code seems more efficient.

5.3 Overall

The results show that the most accurate (smallest energy errors) and least efficient (largest wall-clock time) simulations of the theoretical cluster were produced when most particles were inserted into tree-code. This is unexpected as for particle number N=10000 the tree-code should be more efficient than the direct code `(Lecture L2_gravity_in_AMUSE)`. We decided to investigate further as to what the cause for this could be.

6 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.

7 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. 6, 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.

In Fig. 7, 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 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. 8 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, 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. 9 further backs up this assertion, generally replicating the same results as for the half

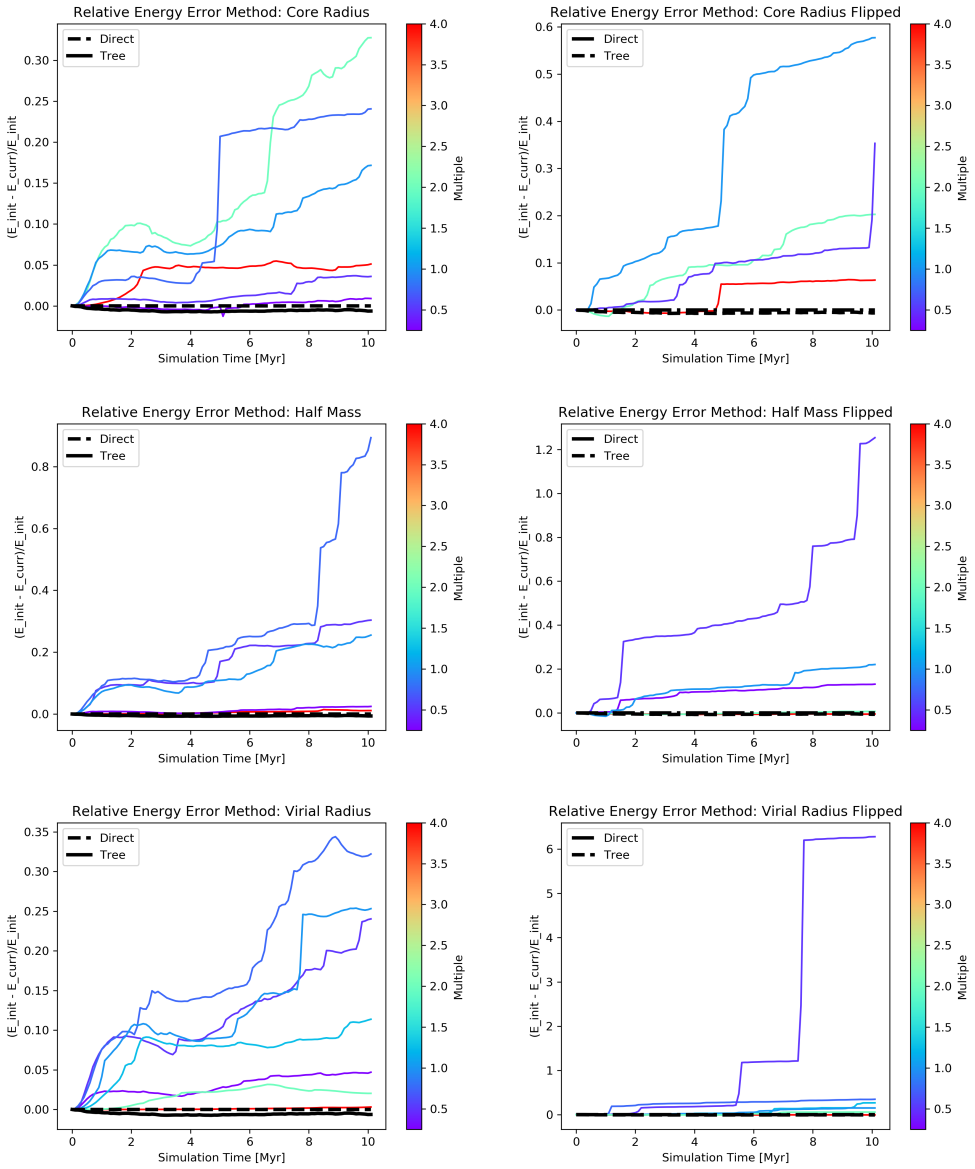

Figure 6: Relative Energy Error for various radius splitting methods. Flipped means that particles whose distance from the center of mass >= radius * multiple were evolved by the tree code, and all others by the direct code. Vice versa for the ones that do not have Flipped.

mass radius.

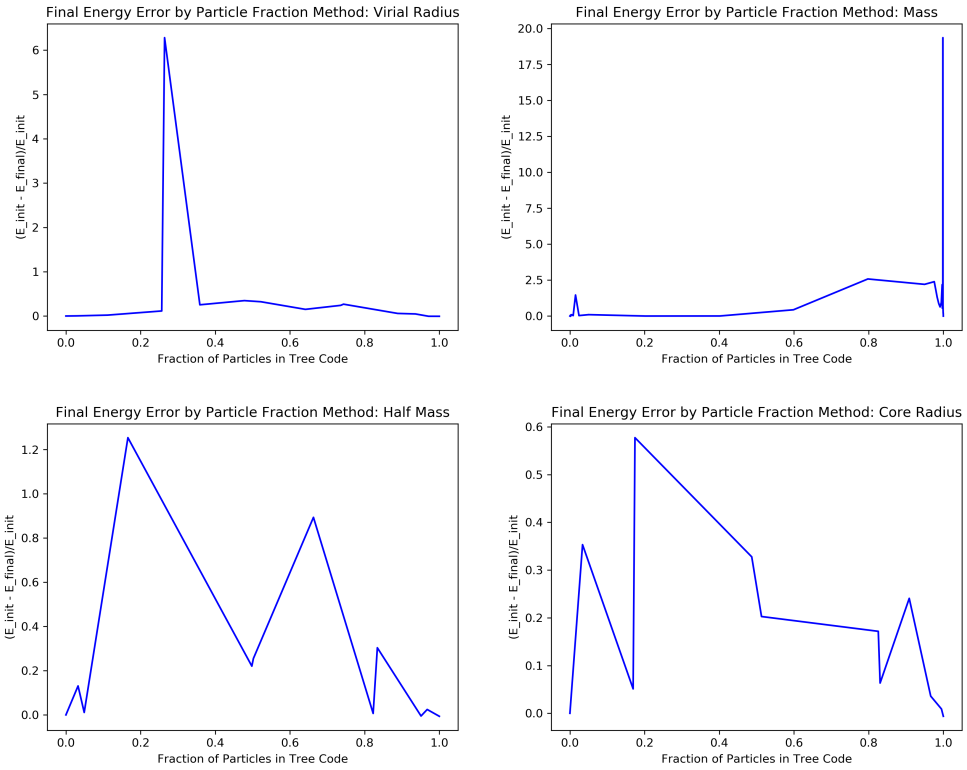

Figure 7: Final Energy Error vs the fraction of particles ran in the tree code for all four splitting methods.

From Fig. 9 we can compare the core radius evolution as calculated by the four splitting methods. There are many similarities found in the simulations for which the particle set was split according to a radius (core, half-mass, virial). As with the half-mass radius, the mass split method produces no core radius evolution similar to the direct code only core radius evolution.

8 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.

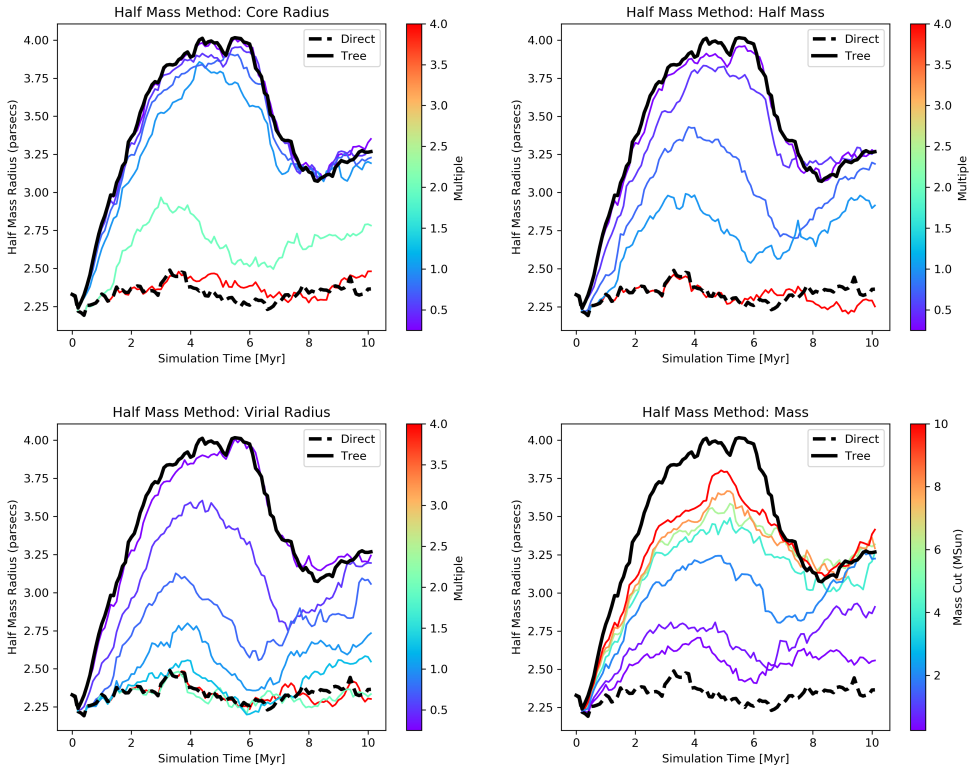

Figure 8: Half Mass Radius over time for all four splitting methods.

9 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

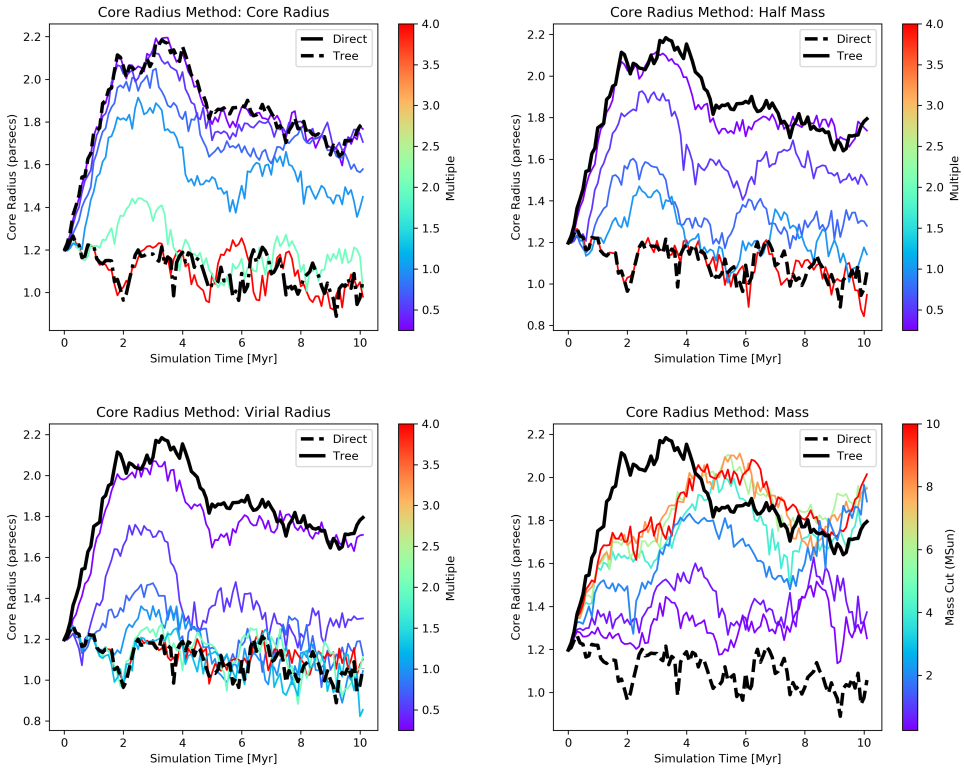

Figure 9: Core Radius over time for all four splitting methods.

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

10 Conclusion

From our experiments, it seems that splitting the particles based on the mass can give decent results. Issues include the loss of energy, and AMUSE's Bridge is not designed for intermixed gravitational systems, resulting in some instability as seen in the animations. Other methods of splitting the particles into two populations seem like they are more accurate, as they maintain more spatial separation between the two populations than the mass splitting method, while still being much faster than the direct nbody code. This suggests that splitting the population of stars into two groups, one for the more accurate direct code and one for the tree code can be a valid option for large numbers of particles, but only for certain mass cuts or radial cuts that minimize the energy error.

References

[LeBlanc] Francis L., 2010, Wiley

[King 1958] King I., 1958,

[King 1966] King I., 1966,