Centre for Planetary Science
Summer Research Summary

Loic Nassif-Lachapelle
Supervisor: Dr. Alan Jackson

August 2018

Contents

1 Introduction

My work with Dr. Jackson over the summer of 2018 involved numerically analyzing the collisions of planetesimals and embryos within the formation of a planetary system, with the sun as the primary, of around the first 50 million years of its evolution. We've explored a few different N-body integrators such as GENGA and MERCURY.

Terrestrial planet formation is believed to have developed through multiple stages (Chambers (2013)). N-body analysis of planetary evolution focuses on the late stage of this planet formation process. Such late stage is characterized by a few dozen embryos and planetesimals colliding with each other. Such a process is inherently stochastic (Chambers (2013), Kaib & Cowan (2015)), and thus a statistical approach to the analysis of N-body simulation results lends itself as a good tool.

Most of N-body simulations assume perfect mergers mostly due to the computational limitations, which is indeed the case for MERCURY and GENGA. However, taking into account a wider variety of possibilities, such as Hit and Run collisions and fragmentation would improve the overall accuracy of N-body outputs. (Leinhardt & Stewart (2012)) have developed a way to analyze and correct perfect merger N-body simulation outputs to take into account non-perfect merger collision scenarios. This scheme as modified by Dr. Jackson and his PhD student Travis S.J. Gabriel (Gabriel & al. (2018)) which is the scheme I explore in section 4 and 5.

Looking at the relationship between a planet's accretion history and its final mass and position is important for gauging the probability of the formation of bodies that would lead to Earth-Moon impact scenarios (Kaib & Cowan (2015)). As such, in section 5 I attempt to create an algorithm that visualizes MERCURY outputs and shows a planet's collision history as a tree diagram.

2 Running GENGA and MERCURY Simulations

(Chambers (2013)) runs eight simulations. Each simulation is comprised of 14 embryos and 140 planetesimals. The embryos and the planetesimal share an equal split of the total initial mass of 28 Mars-mass or about $3M_{\oplus}$. In contrast, (Kaib & Cowan (2015)) runs three ensembles of 50 simulations. Each such simulation is comprised of 100 embryos and 1000 planetesimals. Just as (Chambers (2013)), the total mass of the disk is split evenly between embryos and planetesimals. In this case with a total disk mass of $5M_{\oplus}$. Both papers use a bulk density of $3g/cm^3$. Due to the finite time-step nature of integrators, computations of objects near the sun become vastly inaccurate. As such, both papers set a distance from the sun as to where an object is considered of have "hit" the sun. For (Chambers (2013)), such a limit is 0.2AU, while for (Kaib & Cowan (2015)), the distance is 0.1AU.

We've considered a collision with the sun at a radius of 0.01AU. We've used an initial size of 15 embryos and 150 planetesimals. We've used a total mass of $5M_{\oplus}$ divided in two between embryos and planetesimals. The initial position of the embryos were implemented by spacing them out over equal intervals, similarly to (Kaib & Cowan (2015)). We've ran about 89 different simulations each up to 50 million years. The mass distribution of the embryos were divided up in two difference ways, each way being roughly half the number of simulations. The first way is to separate the masses evenly among all the embryos. The second way, used in both (Kaib & Cowan (2015)) and

(Chambers (2013)), is to withhold a surface density proportional to $a^{-3/2}$ where a is the semi-major axis, and fix the masses accordingly.

3 Data Analysis on Simulation Outputs

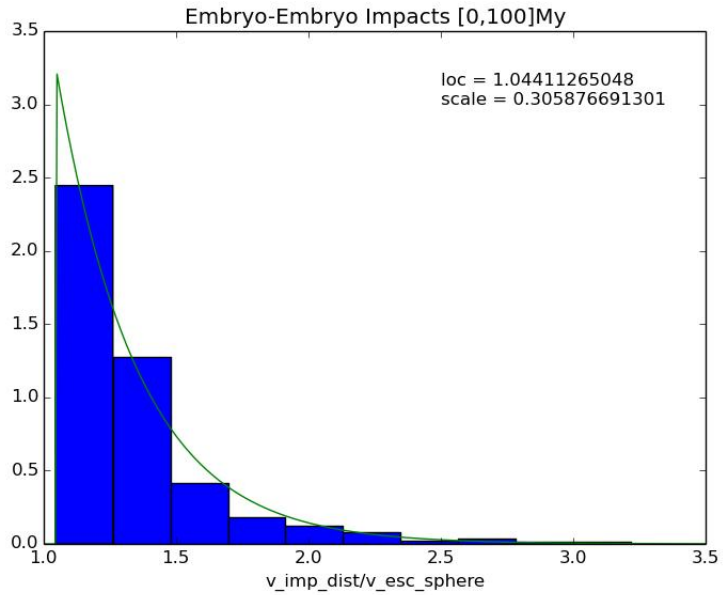

Figure 1: Embryo impacts follow a decreasing exponential curve.

Including my own simulations, there were simulations done by a previous student that I've used to perform analysis on, while my other simulations were running. The plots and discussion in this section, as well as in the subsequent sections are all from this set of data, which comprised of over 89 simulations running for 50 million years. Some of the simulations contained Jupiter, while other did not. In either case, the general behaviour of the ratio of impact velocity and escape velocity follows previous results such as in (Leinhardt & Stewart (2012)). Figure 1 demonstrates the impact velocity over the escape velocity of all impacts of a given simulation over the simulation time. The escape velocity is computed by assuming that the collision yields a perfect sphere of the same density as the target and impactor and the masses combined, from which we can compute the radius of this new object. An alternative to computing the radius that way is to compute the distance between the impactor and the target on impact and use that as the radius to compute the escape velocity. We've noticed, especially for computations involved in section 4 and section 5, that the second method yields more accurate results.

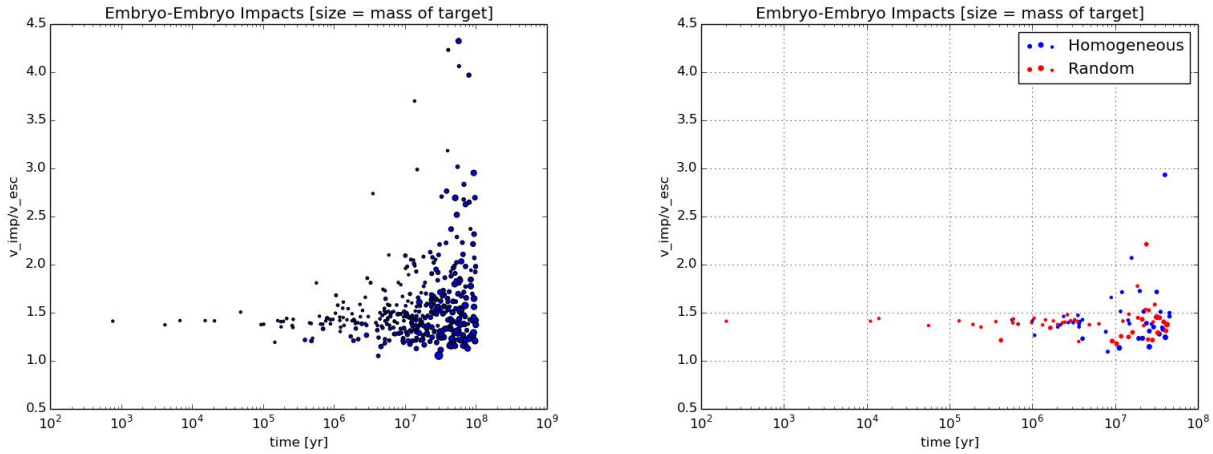

Figure 2: On the right side, the distribution of impact velocity over the escape velocity of collisions in simulations without Jupiter. Homogeneous refers to equally spaced initial positions for both the planetesimals and the embryos, while random is a random initial position. There seems to be little to no differences between the two initial position configurations. On the left hand side, the same distribution is shown but for simulations that include Jupiter. For both types of simulations, impacts that have more massive targets tend to occur much later in the simulation time as well as producing much lower velocity ratios.

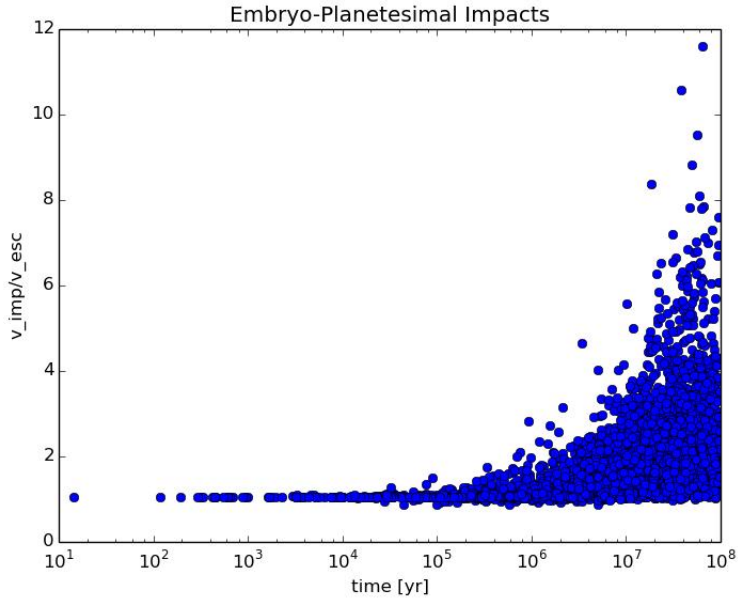

Figure 3: As time passes, the ratio of impact velocity over the escape velocity increases due to the increase in size of embryos as they accrete more mass.

4 Hit and Run Collisions

When running the mass-correction algorithm from (Gabriel & al. (2018)) to take into account Hit-and-Run collisions, we've noticed that about 25% of the collisions are Hit-and-Run. This is rather surprising considering that (Chambers (2013)) states they've found that around 42% of collisions are Hit-and-Run, while (Kokubo & Genda (2010)) have their number at 49%.

It seems that about 54% of Hit-and-Run collisions between two embryos is followed up by a subsequent collision of the same two bodies (Chambers (2013)). With that in mind, we've explored the idea of running the mass-correction algorithm assuming that all Hit-and-Run collisions would occur again and see the effect it would have. For simplicity, the second collision is considered to be a perfect merging. This means that instead of just taking the mass of the largest remnant as the final mass of the collision, we add the mass of the runaway object and add it to the largest remnant mass. When running the (Gabriel & al. (2018)) algorithm, I've decided to sample the impact angle from a random sin distribution ranging from 0 to $\pi/2$. This means that every application of the algorithm has some randomness associated with it. To mitigate this, for both the genealogy tree in Section 5 and the analysis in this section, I have performed the simulation multiple times. In Figure 4, the difference between taking into account a second subsequent collision is simply an increase in the corrected mass. Figure 5 demonstrates the same conclusion, where "lvl" corresponds to the number of collisions the target has already been through. Note that in Figure 5, the reason for the linear progression of the levels is due to the fact that the mass-correction algorithm is applied by assuming a perfect merging mass of the target and the impactor.

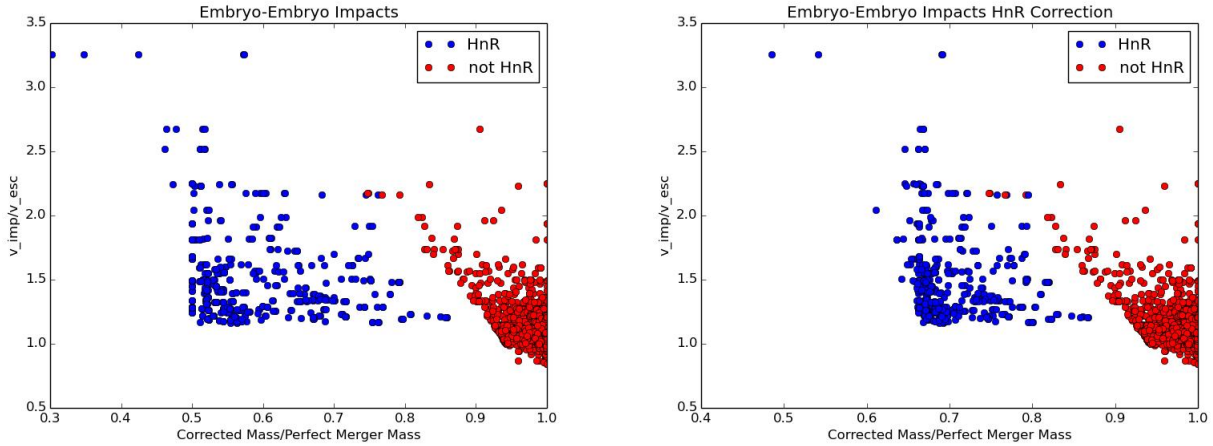

Figure 4: Multiple runs of the mass correction algorithms applied on the same data set. The left plot assumes no subsequent collision by the runaway mass if it was a hit and run collision, while the right plot does assume a perfect merger impact after the initial collision.

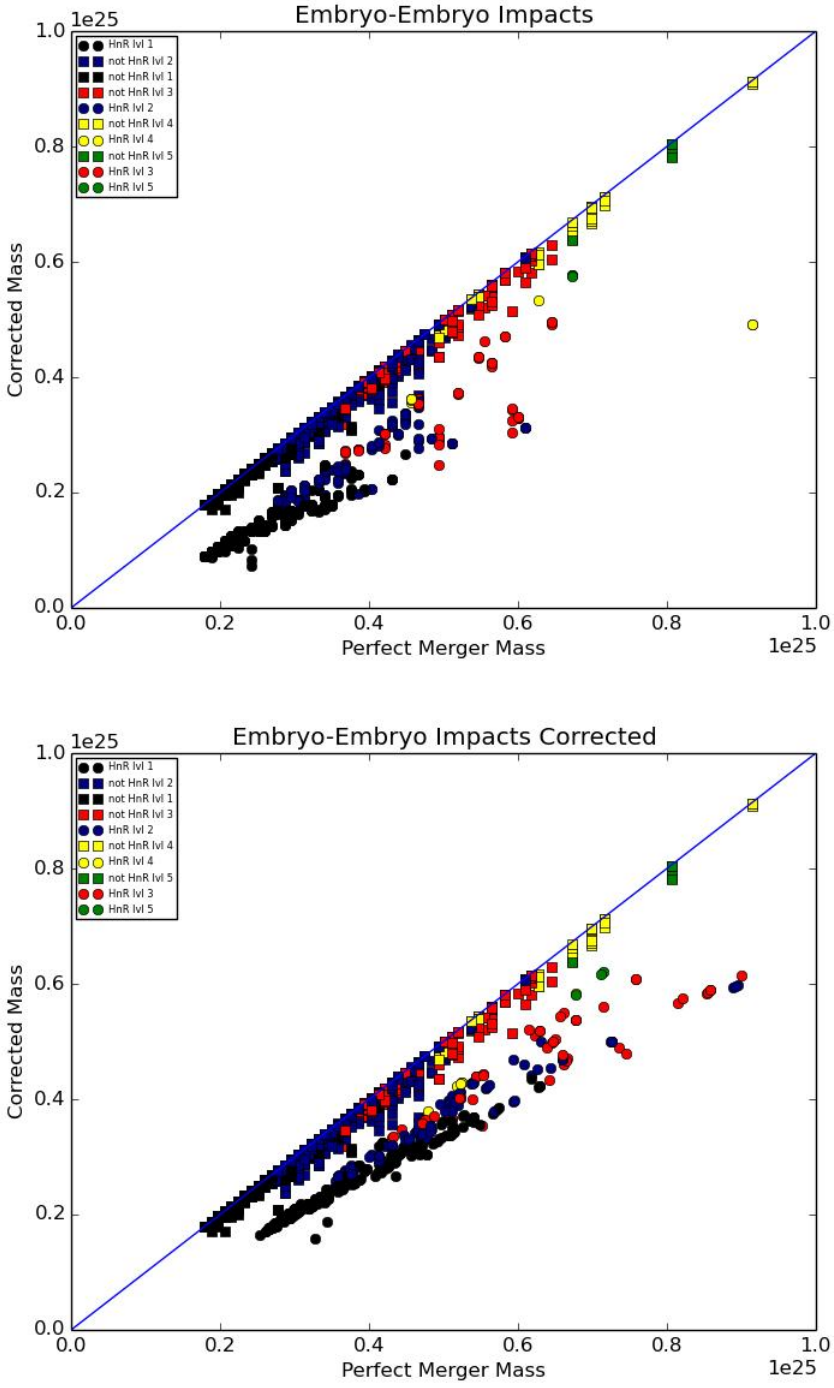

Figure 5: The "lvl" are simply the target's numbered collision so far. Thus a "lvl 2" indicates that the target is the result of a previous collision.

5 Genealogy Collision Tree

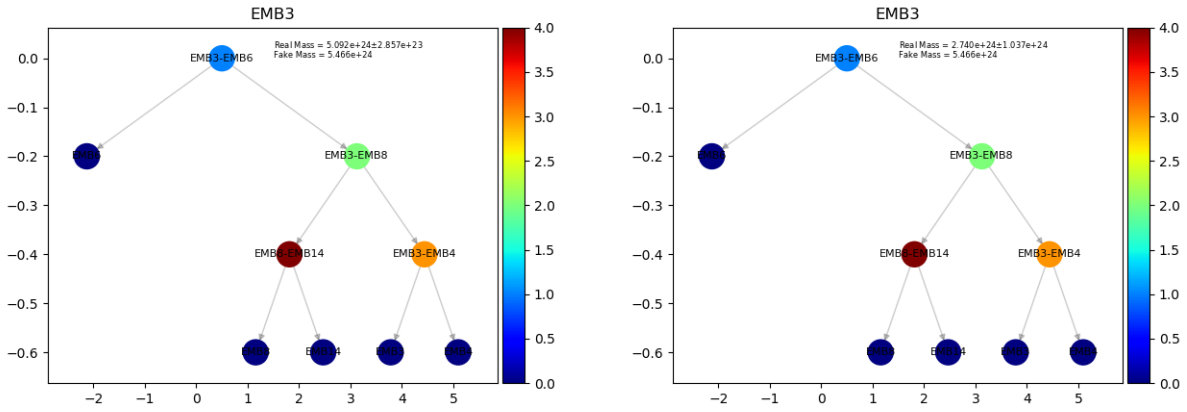

Figure 6: Example of collision history. The left graph computes a corrected mass for the final collision assuming perfect merging for all previous collisions, while the right plot computes the corrected mass given that previous collisions were also corrected, those results are labeled as "Real Mass", while "Fake Mass" is simply assuming perfect merger for all collisions.

There are a multitude of information necessary to develop a collision tree such as the one in Figure 6. The most obvious piece of info needed is the collision history of a particular embryo. Thankfully, the `.emb` files provide this information, as well as the mass of the target and the mass of the impactor. The process of the actual software is then relatively straightforward, although not trivial to implement. A binary search tree data structure was the most obvious structure to use for my algorithm. To visualize the binary search tree, I've used the Python graph package `networkx`, a free package to visualize graph structures.

An important quantity to compute when looking at the collision history is the relative velocity,

$$v_{rel} = \frac{\sqrt{v_{imp}^2 - v_{esc}^2 + v_{esc,sphere}^2}}{v_{esc}} \tag{5.1}$$

where $v_{esc,sphere}$ is computed using the total mass of the impactor and the target, as well as a merging of the two bodies assumed to produce a perfectly spherical body of the same density, from which the radius can be computed. This requires the additional files of `.colls` and `.dat`. The relative velocity is colour coded in Figure 6.

There are two ways to apply the mass-correction algorithm discussed in (Gabriel & al. (2018)). The first one is to only correct the current collision based on the incorrect simulation masses of the impactor and the target. The other, and more interesting, method is to correct the current collision based on previous already corrected masses of the impactor and the target. A few challenges arise during implementation, as one must be careful between using the simulation computed velocity within the mass-correction velocity as opposed to the newly corrected velocity. Furthermore, the implementation requires the application of a reverse breadth-first search algorithm to be able to map out the tree in inverse order. This propagating mass correction is applied to a data set of collisions representing an evolution of 50 million years with the inclusion of Jupiter, Figure 7 is

the result of this. Note that the relation between the perfect merger mass and the corrected mass decreases as we move up the collision tree, as expected. This is not the case for Figure 5, which assumes all previous collisions to be perfect mergers. Thus this retroactive correction in mass is actually quite important when studying later collisions within a simulation.

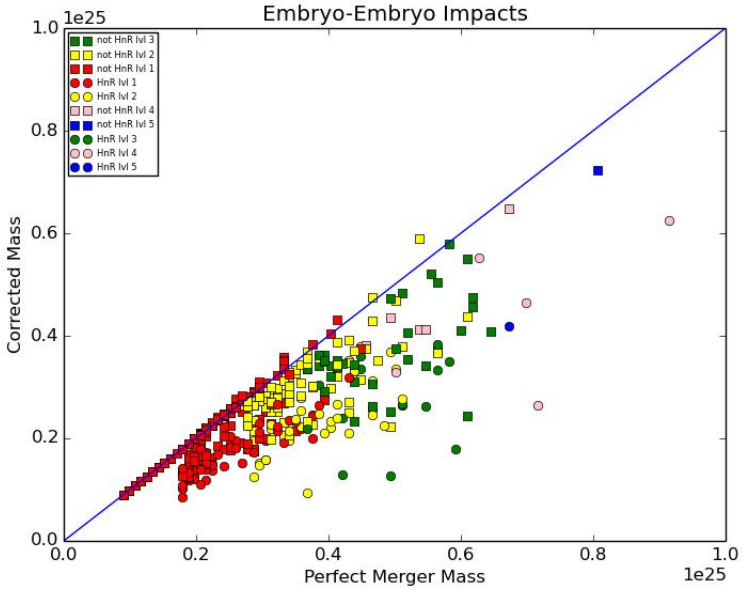

Figure 7: Later collisions are clearly impacted heavier by the mass correction as an accumulation of ejected mass ignored by the simulation impacts the final mass of the embryos.

References

Chambers J., 2013, Icarus, 224, 43

Gabriel T. S., al. 2018, draft

Kaib N. A., Cowan N. B., 2015, Icarus, 252, 161

Kokubo E., Genda H., 2010, The Astrophysical Journal Letters, 714, L21

Leinhardt Z. M., Stewart S. T., 2012, The Astrophysical Journal, 745:79, 27