LAB REPORT N-BODY SIMULATIONS

Astrophysical Simulation Course

This document reviews key concepts for simulating star clusters using N-body methods, focusing on potential modeling, setting initial conditions, and understanding star cluster dynamics, particularly the effects of gas expulsion. The information is drawn from lecture materials and audio transcripts dated April, 2025.

Introduction

A central concept in simulating gravitational interactions is the potential. **Simulations employ two main types: fixed (analytical) potentials and live (N-body) potentials**. **A fixed potential is defined by an unchanging analytical function**, making it computationally efficient as particle accelerations can be calculated analytically. This approach is suitable when the simulated object's mass is negligible compared to the system's dominant mass, such as a dwarf galaxy orbiting a massive cluster, because the object's movement doesn't significantly alter the overall potential. However, it is inaccurate for systems where components have comparable masses. In contrast, **a live potential is constructed directly from the distribution of the simulated particles themselves**. It dynamically adapts as the particles move and redistribute, reflecting the changing gravitational landscape. Although more computationally intensive, **this method is essential for accurately simulating systems with interacting components of similar mass**, like merging galaxies, or where the self-gravity of the particles, as in star clusters, is significant. The live potential arises from the sum of the individual potentials of all particles, resulting in a potentially complex structure that evolves throughout the simulation.

N-body systems, by definition, consist of numerous individual particles ("bodies") that interact gravitationally. Running an N-body simulation involves tracking these mutual gravitational interactions over time. Consequently, N-body simulations inherently utilize a live potential, as the collective gravity of all particles dynamically shapes the potential well in which they move.

Setting up realistic and stable **initial conditions is critical for N-body simulations**. Simply placing particles at rest will lead to gravitational collapse. **Achieving stability requires carefully assigning both initial positions and velocities to the particles**. For instance, creating a stable dark matter halo necessitates setting particle velocities correctly to maintain equilibrium. Random number generators are commonly used tools for assigning these initial positions and velocities, often drawing numbers between 0 and 1. Using a specific seed for the generator ensures reproducibility. Positions can be assigned using various methods, such as distributing particles uniformly within a cube, placing

them on a spherical shell using random angular coordinates, or creating a spherical distribution with varying density by making the shell radius itself a random variable.

Assigning initial velocities is equally important for stability. Instead of a single velocity value, particles at a given position typically receive velocities drawn from a distribution function specific to the system model, like a Plummer sphere. **This velocity distribution, often expressed relative to the local escape velocity, ensures that particles remain bound**; velocities exceeding the escape velocity would lead to particle loss and system instability. The process involves sampling velocities from this distribution using random numbers, often by accepting random points under the distribution function's curve. For models like the Plummer sphere, the velocity field is typically isotropic, meaning velocities are distributed uniformly in all directions, which is also achieved using random number techniques similar to positioning.

The Plummer sphere is a widely used analytical model representing the density distribution of star clusters. It is defined by its total mass (Mpl) and a scale radius (rpl), featuring a central density that decreases with radius. The model provides an equation for the mass enclosed within a given radius. By inverting this equation to express radius as a function of the fractional enclosed mass and then assigning random numbers (0 to 1) to represent this fraction for each particle, one can generate a particle distribution that accurately reflects the Plummer density profile. A useful diagnostic for verifying the setup is the half-mass radius (r_half), the radius containing half the total mass, which for a Plummer sphere is approximately 1.3 times the scale radius (rpl). This serves as a sanity check on the initial conditions. The Plummer model also has a known analytical velocity distribution function, crucial for assigning initial velocities that ensure the system's stability.

Testing the stability of the created N-body model is essential. **A stable system, like a correctly initialized Plummer sphere, maintains its overall density distribution over time,** even as individual particles move on their orbits. **Stability is often assessed using Lagrangian radii – the radii enclosing fixed percentages of the total mass** (e.g., 10%, 50%, 90%). Tracking these radii over the simulation duration reveals whether the system is stable, collapsing, or expanding. Simulations should run for a sufficient duration, typically several **crossing times (t_cross), which is the characteristic time for a particle to traverse the system**. Accurate simulation requires appropriate time stepping (dt), small enough to resolve the fastest dynamical processes (e.g., at least 50 steps per t_cross). Numerical stability also benefits from a **softening length (l_soft), which prevents unrealistically strong forces during close particle encounters by smoothing gravity at small scales**, often set near the mean interparticle spacing. Last but not least, **Leapfrog integration is a common numerical method for updating particle positions and velocities reliably over these time steps**.

Finally, these simulation techniques are applied to understand phenomena like star cluster "infant mortality." Stars typically form within clusters embedded in molecular

gas. However, many young clusters are observed to be destroyed shortly after formation. This destruction is often attributed to the removal of the primordial gas, frequently driven by feedback effects like supernova explosions from the first massive stars. The severity of this disruption depends primarily on the initial gas fraction (fgas) – the proportion of the cluster's mass in gas – and the gas removal timescale (t_gone) relative to the cluster's crossing time. **Higher gas fractions and more rapid (instantaneous) removal lead to greater disruption**. Simulating this involves adding a potential representing the gas to the N-body system and then modeling its decay over time (t_gone) starting after a delay (t_SN). As the gas potential weakens, stars that were gravitationally bound by the combined mass of stars and gas may become unbound and escape, potentially leading to the cluster's complete dissolution. Tracking Lagrangian radii during such simulations demonstrates the expansion or collapse resulting from gas expulsion and helps quantify cluster survival prospects.

Glossary of Key Terms

- Fixed Potential: An analytical description of a gravitational potential that remains constant throughout a simulation, independent of the movement or distribution of particles within it.

- Live Potential: A gravitational potential in a simulation that is calculated based on the current positions and masses of the particles, and therefore can change and adapt as the simulation evolves.

- N-body System: A simulation method used to model the gravitational interactions between a large number of particles or bodies.

- Random Number Generator: A computational tool that produces a sequence of numbers that appear random, typically between 0 and 1.

- Seed (Random Number Generator): An initial value provided to a random number generator that determines the specific sequence of random numbers produced. Using the same seed allows for the reproduction of the same sequence.

- Uniform Sampling: A method of generating random numbers or selecting points such that each value or location within a given range or space has an equal probability of being chosen.

- Spherical Shell: A hollow sphere-like distribution of particles at a specific radius from a central point.

- Plummer Sphere: A commonly used analytical model for the density distribution of spherical stellar systems, characterized by two parameters: total mass and scale radius.

- Enclosed Mass (M(<r)): The total mass contained within a sphere of radius 'r' from the center of a system.

- Half Mass Radius (r50): The radius within which half of the total mass of a system is contained.

- Escape Velocity (Vesc): The minimum velocity a particle needs at a given location to escape the gravitational pull of a system and never return.

- Distribution Function (for Plummer Sphere): A mathematical function that describes the distribution of velocities of particles within a Plummer sphere at a given radius, relative to the escape velocity.

- Isotropic Velocity Field: A velocity field where the velocities of particles are distributed uniformly in all directions at any given point.

- Lagrangian Radii (r20, r50, r80): Radii that enclose a specific fixed percentage of the total mass of a system (e.g., 20%, 50%, 80%). Tracking their evolution indicates the stability and structural changes of the system.

- Crossing Time (tcross): The typical time it takes for a particle to traverse a characteristic scale radius of a system. It is a measure of the dynamical timescale.

- Softening Length (lsoft): A parameter used in N-body simulations to prevent unphysically strong gravitational forces between particles that come very close to each other. It effectively "softens" the gravitational interaction at small distances.

- Leapfrog Integration Scheme: A numerical integration method commonly used in N-body simulations that updates velocities and positions in alternating half steps, which can improve stability and accuracy.

- Infant Mortality (Star Clusters): The phenomenon where young star clusters are destroyed soon after formation, often attributed to the rapid expulsion of natal gas.

- Gas Fraction (fgas): The ratio of the mass in gas to the total mass (gas + stars) within a system.

- Supernova Time (t_SN): The time in a simulation when the first supernova explosion occurs, triggering the potential removal of gas.

- Gone Time (tgone): The timescale over which gas is removed from a star cluster after a supernova event, often measured in units of crossing times.

Explicit HyperLinks

Main Repository:

https://github.com/ccanunez/Simulation-Course-Project-2

Setting up Plummer Sphere:

https://github.com/ccanunez/Simulation-Course-Project-2/blob/7a33251 8fd7a2d6 97dfacd6 400eb43c1f91af401/Setting%20up%20Plummer%20Sphere.py

Stability of Plummer Sphere:

https://github.com/ccanunez/Simulation-Course-Project-2/blob/f9680e2245d978e65a103c527a76fe9f5303d8e3/Stability%20of%20Plummer%20Sphere.py

Embedded Star Cluster:

https://github.com/ccanunez/Simulation-Course-Project-2/blob/be65af5782162fdcb8f6490c0afe0bf2cb47 25d0/Embedded%20Star%20Clusters.py

Part 1

A computational model representing a Plummer sphere was developed. The simulation parameters were set with a total mass Mpl =1000M⊙ , a Plummer radius rpl =1.0 pc, and Npart =1000 particles. Particle positions were generated in units of parsecs according to the Plummer density profile. The half-mass radius was calculated from this particle distribution and verified against the theoretical expectation (rhalf ≈1.3rpl). Subsequently, initial velocities were assigned to each particle in units of km/s, sampled from the velocity distribution function appropriate for a stable Plummer sphere, which is dependent on the local escape velocity (Vesc). A plot of the total velocity magnitude (Vtot) versus radial position was generated for all particles, and the theoretical escape velocity curve Vesc (r) was overlaid to confirm that the assigned velocities were physically realistic (i.e., Vtot <Vesc).

Procedure 1.1.1

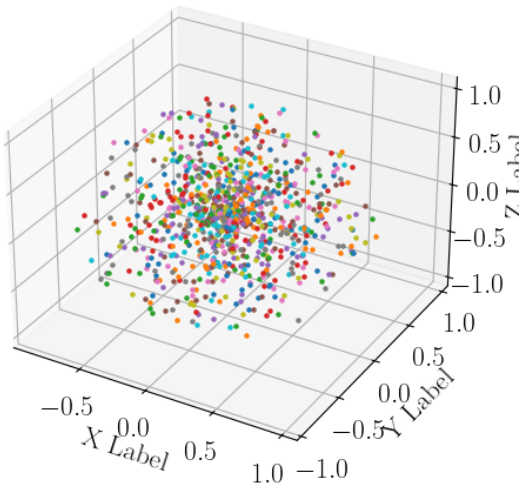
A. Plummer Sphere Model Setup:

A numerical N-body representation of a Plummer sphere was initialized using custom code. The model was configured with a total mass Mpl =1000M⊙ and a characteristic Plummer radius rpl =1.0 pc, comprising Npart =1000 equal-mass particles. The spatial coordinates (x, y, z) for each particle were determined in parsecs by sampling from the Plummer density profile, $\rho(r) \propto (1+(r/rpl\ )2)-5/2$. To validate the positional setup, the radius containing half the total mass (rhalf) was computed directly from the generated particle positions and compared to the analytically expected value for a Plummer sphere, rhalf ≈1.305rpl .

The python code developed can be seen in the following repository:

Setting up Plummer Sphere.py > Part 1.1.1

Python Outputs 1.1.1



Plummer sphere random distribution in the whole space.

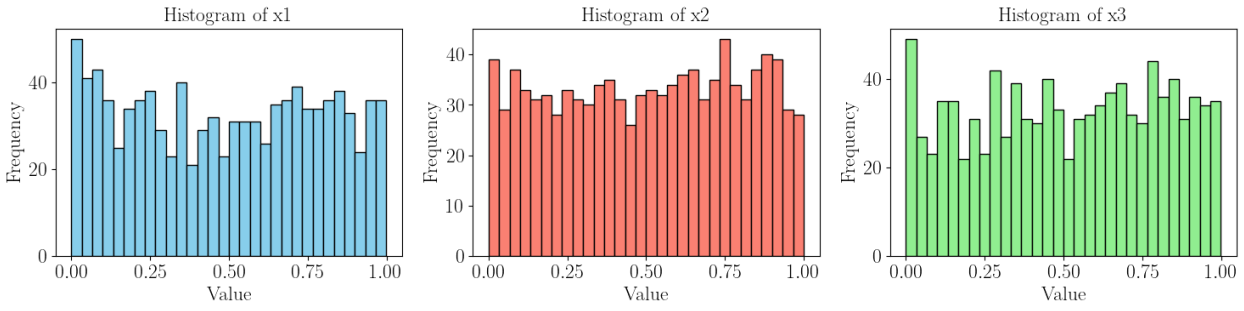
This code simulates the spatial distribution of particles in a **Plummer sphere**, a common model used in astrophysics to describe the spherically symmetric, isotropic distribution of stars in a globular cluster or galaxy. The simulation is performed using random sampling and visualized in 3D and through histograms of the random variables.

It can be seen a major distribution at the center of the globular cluster. This is what we would expect to a real observation of the cosmos, since the grate gravitational pull concentrates great part of the mass at the center. Those point (representing galaxies) which are at the shell of this spherical distribution can be interpreted as outliers.

Notes:

1. This implementation does not strictly follow the radial distribution of a Plummer sphere (which requires sampling from a specific cumulative distribution function). Instead, it uses uniform sampling in radius and angle for demonstration purposes.

2. The script is useful as a first approximation or visualization tool but would need adjustment for precise N-body initial conditions.

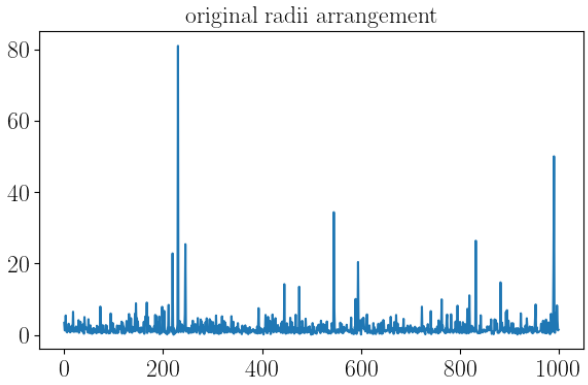


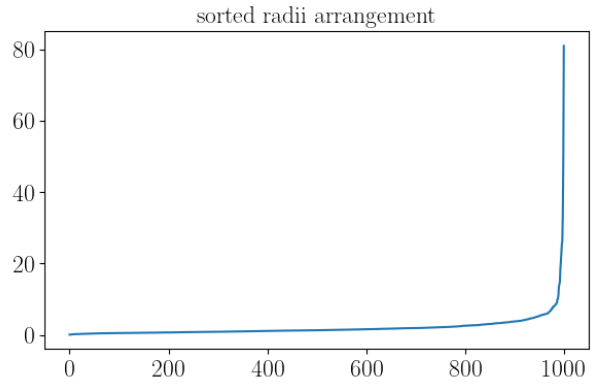
The more stars we compute, the more similar the histograms tend to be in each plot.

Python Outputs 1.1.2

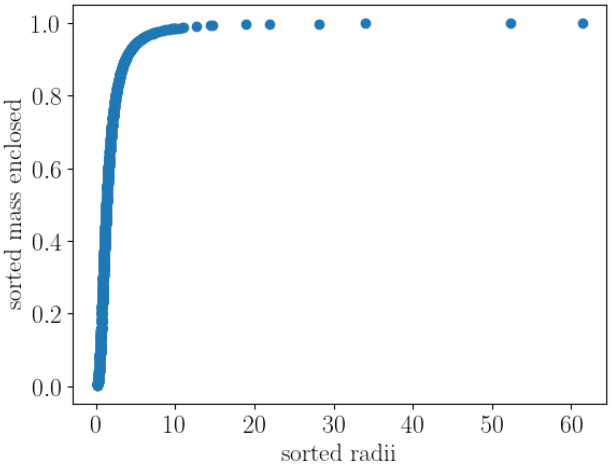
The radial relation we can encounter is the randomness of the sample viewed in the first plot. However, if the sample radii array is sorted then we can observe and study the half mass radius. Note that since each point has 1 solar mass the half of the total mass (1000) will be 500.

After running the computational python script we can conclude that the radius containing half the total mass enclosed is 1,3 as expected for the Plummer distribution.





Given that we are observing a distribution with a greater concentration of mass at its center, we can observe a proportional relationship between the mass enclosed (1 solar mass for each point) and the radius. At a low radius scale, the mass enclosed tends to increase more steeply, but as the radius approaches the entire cluster, the mass enclosed remains constant.



Procedure 1.1.2

B. Velocity Initialization and Verification:

Following the spatial distribution setup, initial velocities were assigned to the particles in units of km/s. Particle velocity magnitudes were drawn from the probability distribution function f(V) corresponding to the Plummer potential, ensuring $0 \leq V < V_{esc}(r)$, where $V_{esc}(r)$ is the escape velocity at radius r. Velocity vectors were oriented isotropically. To verify the assigned kinematics, the total velocity magnitude ($V_{tot} = V_{x2} + V_{y2} + V_{z2}$) of each particle was plotted against its radial distance from the sphere's center. The theoretical escape velocity profile, $V_{esc}(r) = 2|\Phi(r)|$, where $\Phi(r)$ is the Plummer potential, was superimposed on the plot to visually confirm that all particles possessed velocities below the local escape speed, consistent with a bound system.

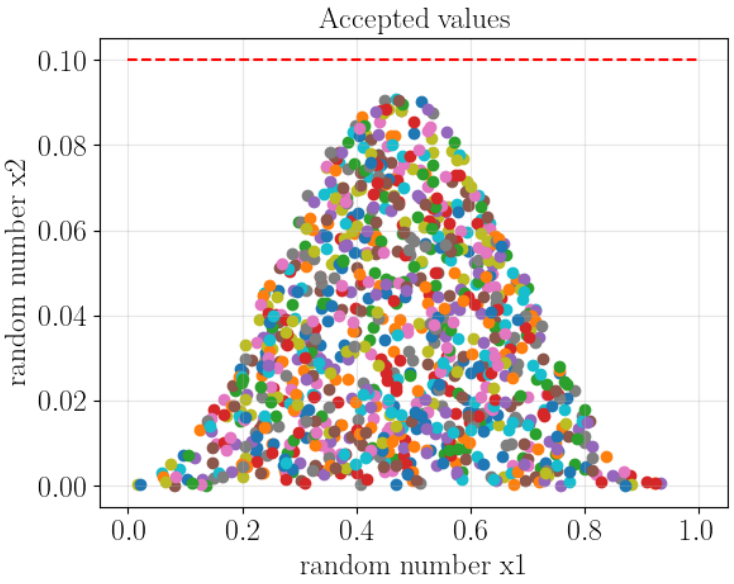
For the previous part see

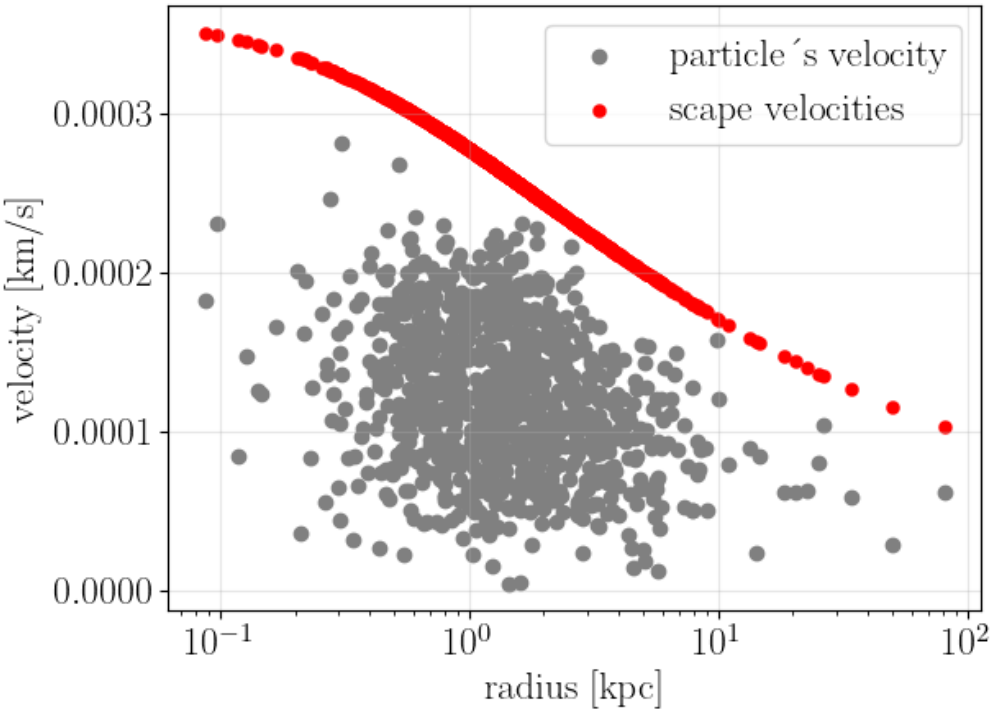
Setting up Plummer Sphere.py > Part 1.1.2

Initial velocities were assigned to the particles within the previously generated Plummer sphere model ($M_{pl} = 1000 M_{\odot}$, $r_{pl} = 1.0$ pc, $N_{part} = 1000$) using a rejection sampling algorithm. This method was employed to ensure the particle kinematics were consistent with the theoretical velocity distribution function for a stable Plummer potential. For each particle at a given radius r, the local escape velocity, $V_{esc}(r)$, was calculated using the standard gravitational constant G(from astropy.constants) and the Plummer potential derived from M_{pl} and r_{pl}. The rejection sampling involved generating pairs of uniform random numbers (x_1, x_2 between 0 and 1). A velocity magnitude was accepted if the condition $0.1 x_2 < x_{12} (1 - x_{12})^{3.5}$ was met, where x_1 represents the ratio of the particle's velocity to the local escape velocity (V/V_{esc}) and the function $x_{12} (1 - x_{12})^{3.5}$ is proportional to the Plummer velocity distribution function. The accepted x_1 value determined the particle's total velocity magnitude via $V_{tot} = x_1 V_{esc}(r)$. Velocity components (V_x, V_y, V_z) were then determined by assuming an isotropic distribution, assigning a random direction in 3D space. Finally, the assigned total velocities (V_{tot}) were plotted against particle radius r (logarithmic scale) and compared against the calculated escape velocity curve $V_{esc}(r)$ to visually verify that all particles were gravitationally bound ($V_{tot} < V_{esc}(r)$).

Python Outputs 1.2

This section of the code generates velocity vectors for particles within a Plummer sphere and compares their total velocities to the local escape velocity, using a rejection sampling method based on the theoretical velocity distribution function of the model.





It is evident that none of the simulated particles surpass the escape velocity, indicating that this system is bound.

Part 2

Procedure 2.1 & 2.2 & 2.3

To assess the stability of the generated Plummer sphere model, an N-body simulation was performed. A system of internal code units was adopted where G=1, the characteristic length Lsim =1.0 pc, and the characteristic mass Msim =1.0M⊙ . The corresponding derived units for time (Tsim in Myr) and velocity (Vsim in m/s) were calculated. The initial particle positions (pc) and velocities (km/s) from Part 1 were scaled into these dimensionless simulation units. The simulation was run for a total duration equivalent to 10 dynamical crossing times (tcross) of the initial Plummer sphere. This duration was calculated in Myr, converted to simulation time units (Tsim), and used as the simulation end time (tend). The integration timestep (dt) was chosen to ensure at least 50 steps per crossing time (dt≤tcross /50) for numerical accuracy. A gravitational softening length (l_soft or ϵ) was implemented to prevent numerical singularities during close encounters, with its value set to approximate the mean interparticle spacing estimated from the number of particles and the previously determined half-mass radius (rhalf). The simulation was executed using these parameters, and the stability was evaluated by tracking the evolution of various Lagrangian radii (radii enclosing fixed mass fractions, e.g., 10%, 50%, 90%) over the simulation time using custom analysis scripts.

For the previous part see

Stability of Plummer Sphere.py > Part 2.1 & 2.2 & 2.3

Regarding the constants and units used it is essential to consider Tsim=14.9 Myrs and Vsim=65.8 m/s while the number of bodies simulated were N=1000. Using the right conversion of this units we can employ the equation for the crossing time such as $t = \left(\frac{3\pi}{32}\right)^{-3/2}\left(\frac{r_{pl}^3}{GM_{pl}}\right)^{1/2}$ while the approximate interparticle spacing was given by the volume $V = \frac{4}{3}\pi r_{1/2}^2$ were the softening corresponds to $s = \left(\frac{V}{N/2}\right)^{1/3}$.

As it was mentioned before, the repository where the python code can be found is already with the explanation of what the code does.

Python Output 2.2

We do not have to worry about the length nor the mass since we are considering values equal to simulated units. Nevertheless, we do have to compute the conversion of units for the velocity in order to gain the right units onto the nbody.py code.

Python Output 2.2 & 2.3

Regarding to the python code employed we can obtain the following results:

```
the crossing time is 0.19784190099458493 in Tsim
and 10 crossing times equals 1.9784190099458492 in Tsim
therefore 10 crossing times in Tsim are 29.5 Myr
the softening is 0.24500594292472744
```

This values were copied and pasted with all the decimal places shown above i.e., we do not want to approximate in order to have accuracy.

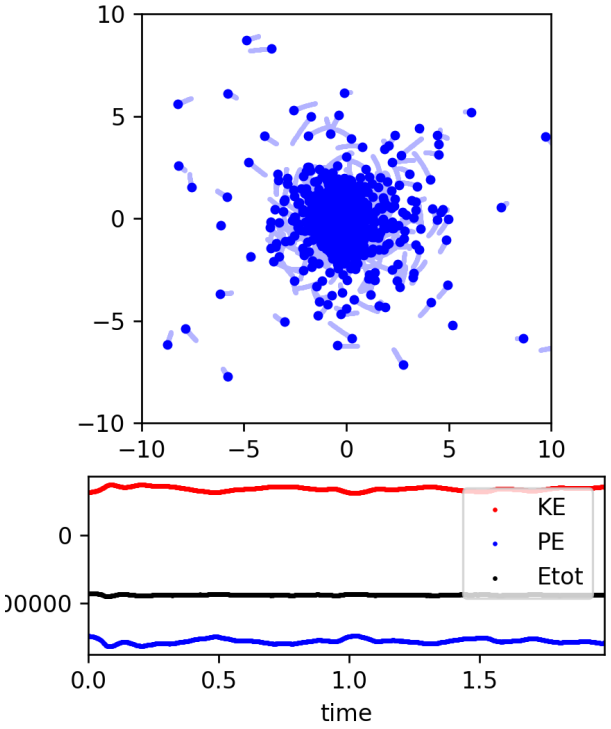
Python Output 2.4

This part of the code does not generate any graphs. However, it does create a table where it lies the positions (x,y,z), the mass (m) and the velocity componentes (vx,vy,vz) all described by its single column for each one. The length of the table should be the total N bodies simulated (N=1000). This table with be read by the nbody.py code in order to simulate the live potential.

Some comments we have to keep in mind are those were the simulation “explodes”. This fact means that the initial conditions were not quite accurate nor good for the simulations. Probably the problem relays in the exportation of velocities onto the data file containing all simulation parameters. However, if the table happens to be fine and right then the problem is absolutely in the simulation parameters created for the nbody.py code. In other words, could be Part 2.2 & 2.3 the key of the problem. It could be possible that the crossing time, the end of the simulation time, the time stepper or the softening length may be causing the potential to be unstable.

What we are looking for is a steady potential with minimum perturbation over time. The same graph that shows the evolution of the simulation also describes the behave of the kinetic, potential and mechanic energy over time. This plot should evolve line a straight line (ideal but not quite right) i.e., we do not want to see bumps or periods where the energy (non of them) has a strange behavior. The following figure should do the trick:

N-Body Simulation output

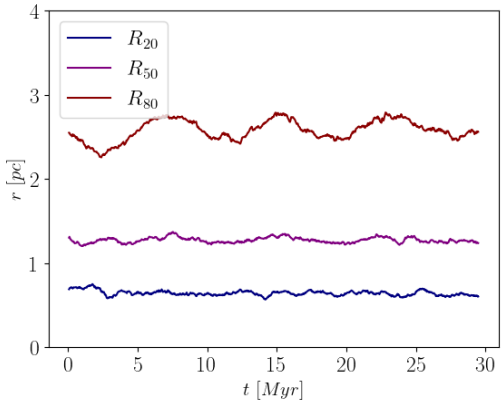


Position evolution over time and display of energy behaviour.

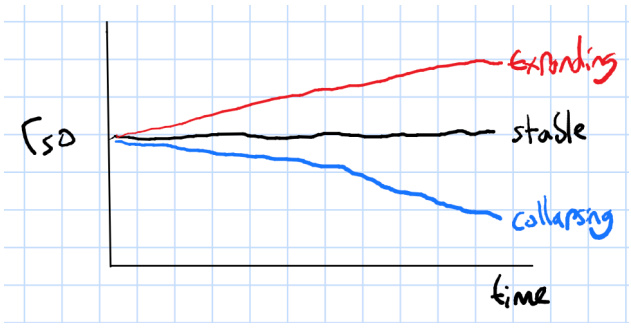
Even though it seems to be randomly moving each line described for every particle is bounded to the overall system. We might see a little dance of the particles over time they remain *still* for are purposes.

Looking into the Lagrangian radii we can see stability of all three radius over time. With this plot we can be certain of the veracity of the simulation where all radius are described as stable. We consider a stable radii where the evolution of inner and outer regions behave the same. Thus, the model is not expanding nor collapsing into itself.

On the other hand, the time evolution that we use to simulate the live potential (in this case the representation of a Cluster: Plummer Sphere) is given by the crossing time. A slightly out of equilibrium model will tend to oscillate (expand or collapse) on timescales. As a result of this we depend on the crossing time; this is why we use 10 crossing times to fully relax the potential.



Lagrangian radii evolution over time



Example of different situation where the simulation is inadequate in exception of the black line (middle one)

The reason why we are plotting 3 Lagrangian radius and not only one relays on the option of a collapsing or expanding region that we are not aware of. It could be possible that the center and outer region is stable over time, but the inner region could be collapsing on itself. Therefore, we ensure that the most representative radius are stable to describe an overall bound system.

Part 3

This section of the project focuses on the dynamic evolution of embedded star clusters in response to gas expulsion. This process can lead to the **premature disruption of young clusters, commonly known as "infant mortality".** Star formation typically occurs in clusters surrounded by dense molecular gas, with gas fractions reaching up to 90–95% of the total initial mass. Since the gravitational potential is predominantly dominated by this gas, its removal can have a significant impact on the cluster's stability. The feedback from the first massive stars, particularly supernovae, is believed to drive this gas expulsion. For low-mass clusters (approximately 1000 $M_\odot$), **even a single supernova event may be sufficient to expel all the gas, potentially unbinding the cluster.**

Two key parameters influence the outcome of this process: the initial gas fraction (fgas) and the timescale over which the gas is removed (tgone). A higher gas fraction increases the gravitational impact of gas loss, while the removal timescale determines how abruptly the potential changes. Instantaneous removal (tgone ≈ 0) represents the most disruptive case, while slower removal allows the stellar system more time to adjust, potentially preserving part of the cluster. The expulsion typically commences after a delay time (t*SN), often set at five crossing times (t*SN = 5 t_cross), to simulate the onset of supernova activity.

To model this numerically, the gas is represented as a Plummer sphere with an analytically derived gravitational potential. The acceleration due to the gas is computed and added to the accelerations from the stellar component in the N-body simulation. Instead of simulating gas particles directly, the model maintains the number of star particles constant while reducing their individual masses to account for the gas fraction. Subsequently, the gas mass is linearly decreased after t_SN over a timescale tgone. These modifications are implemented within a leapfrog integration scheme, where the gas acceleration is evaluated both initially and at every time step.

Lagrangian radii plots are used to assess the effects of gas expulsion on cluster survival. These plots track the evolution of radii enclosing fixed percentages of the system's total mass, such as 20%, 50%, and 80%. A rapid expansion of all radii indicates complete disruption, while stable inner radii with expanding outer layers suggest partial loss of stars. A nearly constant set of radii (as seen above on Part 2.4) implies that the cluster remains largely unaffected.

The main tasks in this section include deriving the gas Plummer sphere acceleration components, modifying the simulation code to include gas effects, testing the stability of clusters with varying gas fractions, implementing time-dependent gas removal, and analyzing survival outcomes using Lagrangian radii plots. **These simulations enable a detailed investigation into the parameter space where embedded clusters either survive or are destroyed due to gas expulsion.**

Procedure 3.1

This part of the project relays on the *comment* code lines that are not visible at first. Thus, we will need to *uncomment* them in order to include the gas fraction onto the Plummer's new potential. Mathematically we are transforming the equation $\phi(r) = -\frac{GM_p}{(r^2 + r_p^2)^{1/2}}$ into $\phi(r) = -\frac{GM_p f_{gas}}{(r^2 + r_p^2)^{1/2}}$ where the acceleration given by the first derivative of the potential $a(r) = -\frac{d\phi}{dr}$ is now equal to $a(r) = -\frac{GM_p f_{gas}}{(r^2 + r_p^2)^{3/2}} r \Longrightarrow -\frac{GM_p f_{gas}}{(1 + \frac{r^2}{r_p})^{3/2}} \frac{r}{r_p^3}$ which is the expression used in the python **nbody.py** code.

Note that there are 4 sections of the code that needs to be uncommented.

Important considerations of the use of the python code given above is to know that the *output.dat* file will have the header—**# x y z id snap**—but it will be **necessary to eliminate the first character** denote by "#". Otherwise, the code will interpret this character not as an index numerical one but part of the columns we will work on. This is a real boomer when we compute the Lagrangian radii script without making this change.

Procedure 3.2 & 3.3

If the simulation goes wrong we would observing an odd behave for the total energy graph display while the simulation is running. This would mean something is not quite accurate relating the gas potential parameters.

Initially there were a big perturbation on both energies and then the particles start to change position drastically. This can be observe in the following plot (see next page). Nevertheless, the Lagrangian radii tell us about the stability of the Plummer Sphere and, consequently, we can observe a nice bound system for all three radius after a certain time evolution.

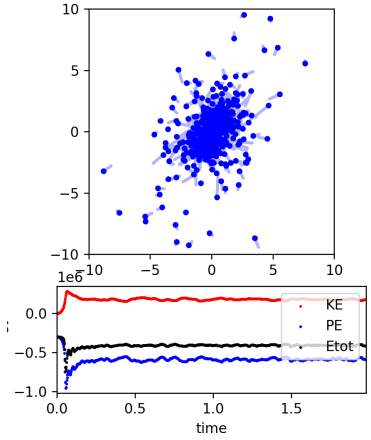
Even though all plots look alike there is a slightly difference between all of them. Primarily, this indicates that all three configurations enables stability over time i.e., creates a bound system. This effect is observable after around 5 Myrs. Naturally, this is what we would analyze from a physical observable cluster. Since no astronomical object is *created* spontaneously and reaches stability, we would expect a proper time evolution for this configuration to reach and enable a bound system.

On the other hand, after this 5 Myrs the Lagrangian radii of the cluster is approximately constant over time i.e., for the next 25 Myrs. This simulated cluster has a normal distribution for particles (not referring to Gaussian distribution but a *expected* one) for all three simulations: no gas fraction, 50% and 95% of gas fraction.

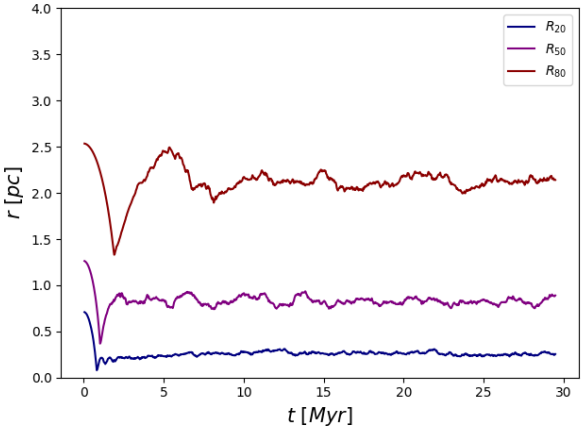
The new changes in the code can be viewed in the following repository:

Embedded Star Cluster.py > Part 3.2 & 3.3

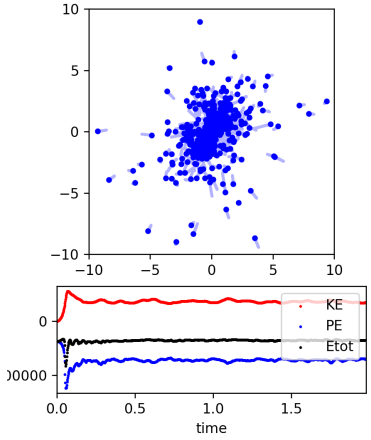
Python Output 3.2 & 3.3



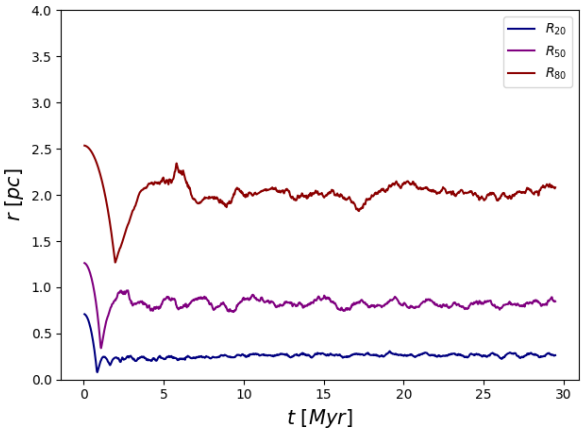
No gas considered



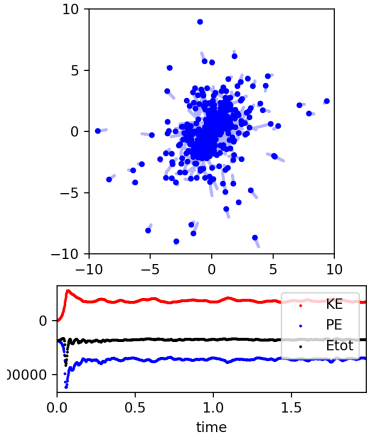
No gas considered



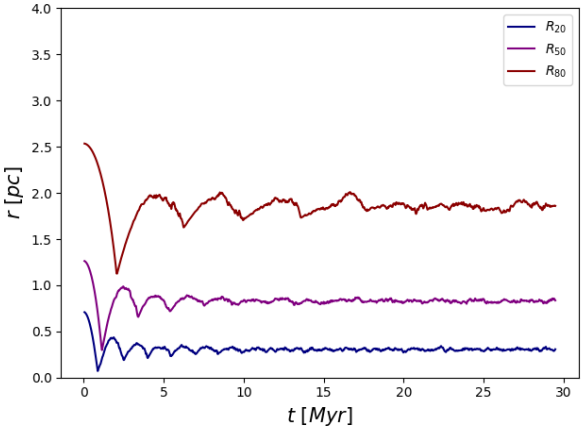
50% gas fraction



50% gas fraction



95% gas fraction



95% gas fraction

Procedure 3.4

In this part of the project we will analyze the removal of gas after 5 crossing times. We should expect a linearly decrease after this evolution.

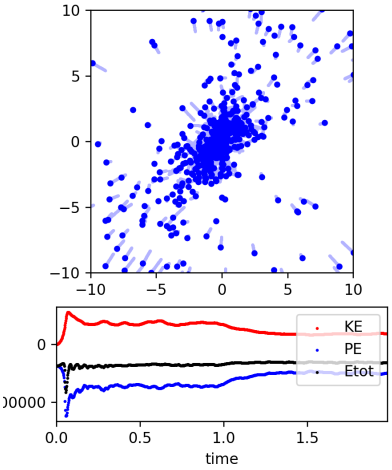
As it was explained before, the removal of gas in this cluster could be explained due to a supernova explosion of some random particle. We do not worry about which particle causes the explosion but to the overall effect that will cause instability and an unbound system.

- Infant Mortality in Star Clusters: Young star clusters are susceptible to dissolution due to rapid gas expulsion, a phenomenon known as infant mortality.
- Gas Expulsion Simulation: The simulation models gas expulsion by representing gas as a Plummer sphere and incorporating its gravitational influence on stellar dynamics.
- Factors Affecting Cluster Survival: The survival of a star cluster after gas expulsion depends on the initial gas fraction and the timescale over which the gas is removed.
- Lagrangian Radii Plot Interpretation: Expanding radii indicate disruption, stable inner radii and expanding outer radii suggest partial survival, and minimal change signifies stability.
- Simulation Goal: Determine conditions for embedded cluster survival or destruction due to gas expulsion.
- Simulation Tasks: Derive analytical acceleration, modify N-body code, test cluster stability, implement gas removal, and analyze system response using Lagrangian radii plots.

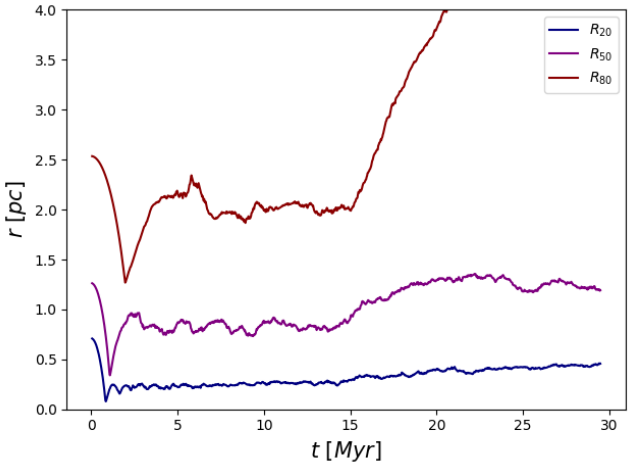
The python code involved can be seen in the following repository:

Embedded Star Cluster.py > Part 3.4

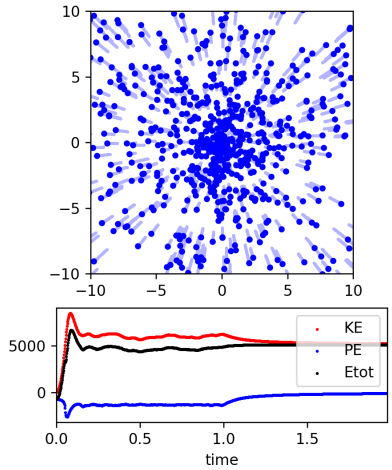
Python Output 3.4



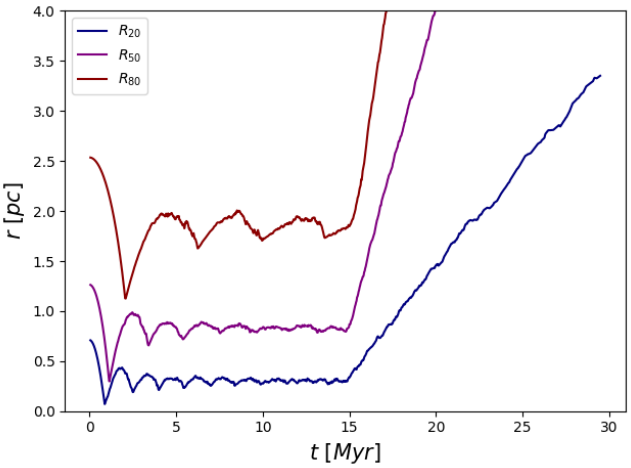
50% gas fraction unbound



50% gas fraction unbound



95% gas fraction unbound

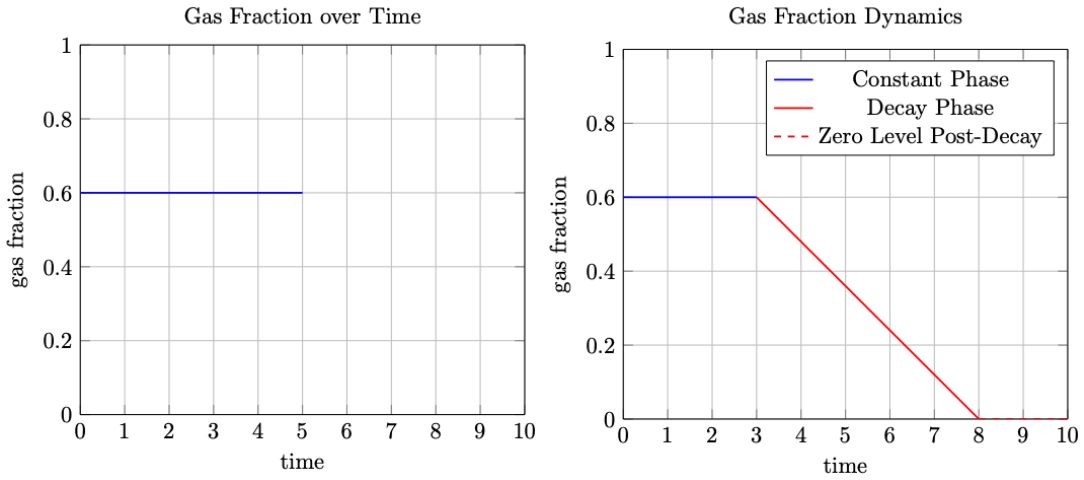


95% gas fraction unbound

It can be observe that the relation were the cluster loosed more mass contained by the gas fraction, the caos of instability disrupts the system more severely. As we have mentioned before, there was no parameter included that advocates the mass nor the characteristic of the star that performed the explosion (supernova). The same supernova was considered for both configurations; thus, the fraction of gas enclosed on the cluster was extremely important and, as we can see, after 15 Myrs the slope for the 50% and 20% of Lagrangian radii are completely different in both scenarios. In other words, the loss of a greater fraction of the cluster causes more instability after a supernova.

Essentially we are looking at a gas fraction over time with the following behavior. Is set constant (in this case to 60%) and then it suddenly stops. Nevertheless, we want to simulate a linear decay after a certain crossing times (5 crossing times).

Interpretation of the gas fraction behavior over time



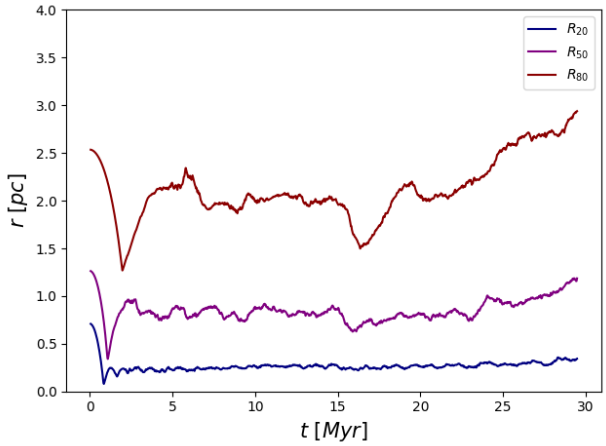
Please do not take seriously the numbers used in this example. They are just used for explication matters.
In the last page we saw the effects of the first plot (left one) and now we attend to study the overall behave of the second one (right one).

The red line labeled "Decay Phase" represents the gradual loose of gas over time. Naturally, we would expect more of this kind of phenomena over the observation of the cosmos. There is no such star able to extract all of the gas enclosed in the cluster but rather have many supernovas over time that decreases the amount of gas in the cluster. We may not see an exact linear behave for the gas fraction that's been pushed out of the system but is a fine approximation. Nevertheless, the following python code could be edited to whom want to analyze a different decay phase.

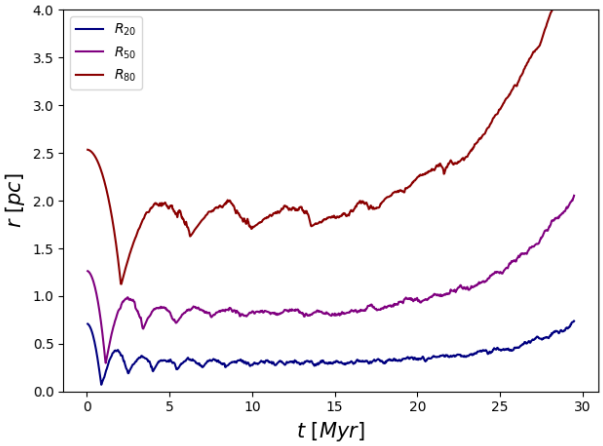
Embedded Star Cluster.py > Part 3.5

We will introduce a new variable call t_{gone} which is going to be $t_{gone} = t - t_{SN}$ where t is the total time simulated or the time where the gas fraction equals zero. Then, t_SN is going to be the time where the supernova happens or the moment where the cluster begins to lose gas. This will lead us to create a function for the gas fraction over time that represents the linear decay such as $f_g = -\frac{t - t_{SN}}{t_{gone}} + 1$ where f_g represents the gas fraction.

Graphs behavior for liner decay



50% linear decay of gas lost



95% linear decay of gas lost

All the configurations (initial conditions) remained unchanged for all simulations conducted on this project except for the one that involves the gas fraction on the cluster. Consequently, the time for the first explosion (spontaneously and with a constant rate) occurs at the same moment, indicating that the perturbation on the cluster commences at 15 Myrs.

At 15 Myrs, a change in the Lagrangian radii can be observed in both graphs presented above. The graph with a 50% gas fraction demonstrates stabilization for R_20 and R_50, which represent 20% and 50% of the mass enclosed within the cluster, respectively. Based on this analysis, we can infer a steady system for these minor radii. However, R_80 gradually decreases over time.

A similar phenomenon occurs when considering a 95% gas fraction. At the end of the simulation, three Lagrangian radii are lost, but R_20 and R_50 maintain a steady system for 5-10 Myrs after the supernova. It is evident that R_80 undergoes a drastic change, representing the outer regions of the cluster. Consequently, particles that were bound at very high distances from the cluster center are completely lost shortly after the supernova.