Extragalactic Astronomy

Collapse of an Homogeneous Sphere - Numerical Study

Efrem Maconi
814088

Academic Year 2020/2021

Collapse of an Homogeneous Sphere - Numerical Study

1 Introduction

We present the numerical analysis of the collapse of an homogeneous sphere caused by its own gravity, the only force acting on the system. The simulations are made through the Joshua E. Barnes' *Tree Code* (for more details see `https://www.ifa.hawaii.edu/~barnes/treecode/treeguide.html`) and the results obtained are compared to the theorical predictions.

The sphere has a mass M and a radius R and it is approximated with a finite number of particles, $Nparticles$, which are equal mass ($m = M/Nparticles$). The particles are distributed such that the sphere has a constant density and this is done using the jupyter notebook *Collapse Homogeneous Sphere - Initial Condition Code (Barnerstreecode)*. The execution of that notebook give as output two files,

```
initial_conditions_file.txt
system_properties.txt,
```

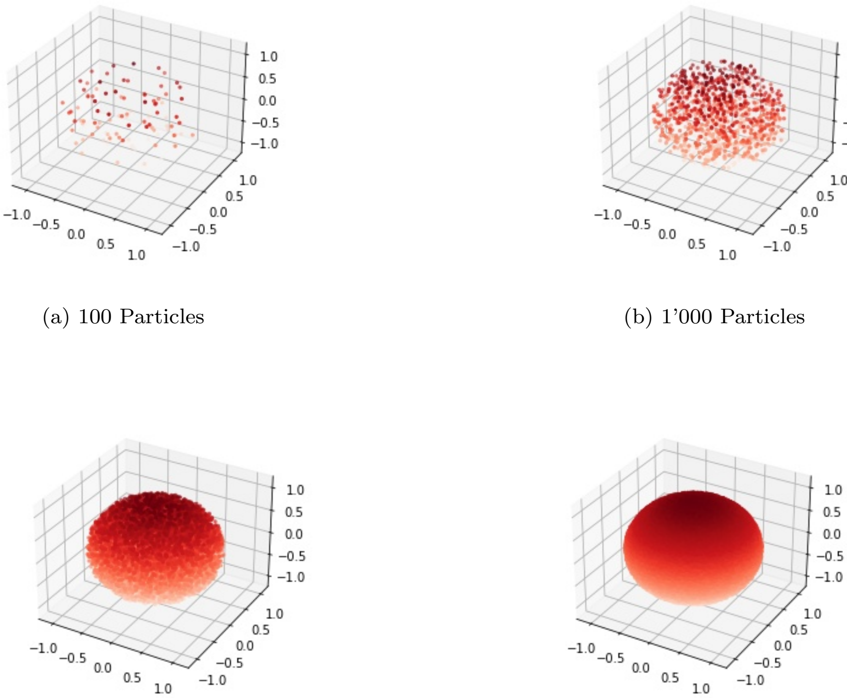
where the first one in used by the *Barnes' Tree Code* and contains the number of particles, the number of dimensions, the initial simulation time and the particles' mass, initial positions and velocities (set equal to 0 at $t = 0$); the second one is used by the jupyter notebook made for the analysis; the initial condition notebook produces then a plot of the positions of the particles, like the one in Figure 1.

The execution of the initial condition notebook has to be followed by the run of the *Barnes' Tree Code* compiled from the terminal writing, for example,

```
./treecode in=initial_conditions_file.txt out=output_data.txt dtime=value
 eps=value theta=value options=out-phi tstop=value dtout=value
 > system_description.txt
```

where it is needed to specify the parameters' values. In the *Collapse Homogeneous Sphere - Initial Condition Code (Barnerstreecode)* notebook is also present a section that gives hints on the choice of the parameters' values.

The discussion that follows contains the comparison between the theorical and the computed results and the analysis of the parameters' value choice.



(a) 100 Particles

(b) 1'000 Particles

(c) 10'000 Particles

(d) 100'000 Particles

Figure 1: Particles' initial positions

2 Choice of the numerical parameters' values

2.1 dtime

The *dtime* parameter is the integration time-step and it's convenient to use a value that has an exact representation as a floating-point number; a small value of *dtime* will result into a better precision but a greater computational cost. For the simulations presented in this relation, we choose $dtime = t_{FreeFall}/1000$.

2.2 eps

The *eps* parameter is the *Gravitational Softening Parameter*. The homogeneous sphere is approximated with a finite number of particles and the computation of its the evolution can cause unphysical scattering between them, scattering that in a real homogeneous sphere are absent; to decrease the effects of these scattering two main things can be done:

- increase the number of particles;
- introduce the so called *Gravitational Softening Parameter*, ϵ.

The scattering effects can be avoided increasing the number of particles: since the mass of the sphere is fixed and the particles are equal mass, increasing their number will result in a decreasing of their mass and, consequently, if two particle are very close one to each other the force will not be so high compared to the case of more massive particles. The increasing of the number of particles can't be done indefinitely because, though it increases the accuracy, it will also increase the computational cost; for this reason we introduce the *Gravitational Softening Parameter*, ϵ that enters in the computation of the force.

The intensity of the Newtonian force between two particles is

$$|\mathbf{F}_{ij}| = \left|G\frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|^2}\right| \tag{1}$$

and therefore it could be very high when two massive points are very close one to each other and this will result into what we have already called *unphysical scatterings*; to prevent this, we introduce ϵ and we compute the force as

$$|\mathbf{F}_{ij}| = \left|G\frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2}\right| \tag{2}$$

where we have introduced at the denominator ϵ that, if it has a tiny value, makes the force less intense when two particles are too close and it's negligible otherwise.

For ϵ we can choose a value that is proportional to the mean separation, Δ_{mean}, between the particles in the sphere

$$\epsilon = 10^{\alpha}\Delta_{mean} = 10^{\alpha}\frac{V}{N} \tag{3}$$

where N is number of particles, V the volume of the sphere and α an integer, positive or negative, number.

To find the best value or, at list, the best range of values for *eps*, we run the *Barners' Tree Code* for a sphere of fixed mass and radius, maintaining fixed all the input parameters except for *eps*; than we compute and analyse the total energy of the system, that should be conserved, and the evolution of the particles' distance respect to the center of the sphere, that should go to 0 as the time approaches the *free fall time* of the system.

The analysis that follows is done in a qualitative way and the images are obtained using the jupyter notebook *Collapse Homogeneous Sphere - Analysis Code (Barnerstreecode)*, where we analyze the output data given by the *Barnes' Tree Code*, runned from the terminal with the following string

```
./treecode in=initial_conditions_file.txt out=output_data.txt dtime=0.000111
 eps=value theta=0.1 options=out-phi tstop=0.222 dtout=0.00222
 > system_description.txt
```

where we change only the value of *eps*.

The

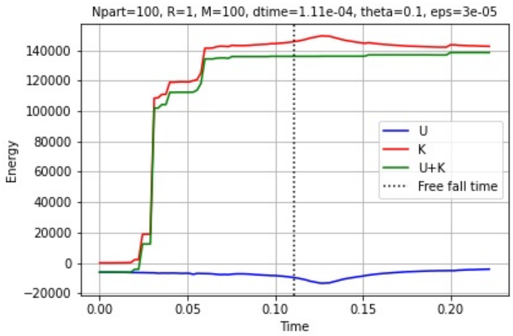
```
initial_conditions_file.txt
```

is created using the jupyter notebook *Collapse Homogeneous Sphere - Initial Condition Code (Barnerstreecode)* for 100 particles and a sphere of mass $M_{iu} = 100$ (i.e. mass in internal units) and radius $R_{iu} = 1$; the mean separation between particle is more or less $\Delta_{mean} = 0.35$.

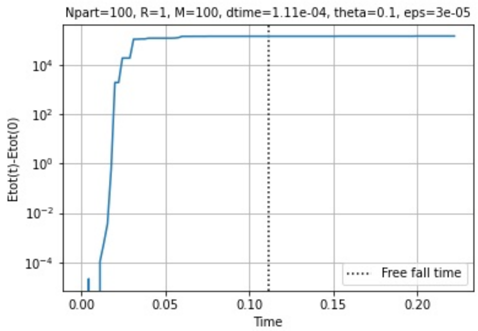
We present now, for each value of *eps*, three plots: one regards the evolution of the energies of the system over time (total potential energy per unit mass, total kinetic energy per unit mass and total energy per unit mass); the second one regards the conservation of energy: the difference between the total energy over time and the energy at time $t = 0$ is plotted; in the last one is plotted the evolution over time of $Nsamples$ (five) particles' distance respect to the center of the sphere. The samples are not chosen by random sampling, but using a particular process: first we choose the number of samples that we want to analyse and then we split sphere's radius at t=0 in $Nsamples$ intervals; for each interval we take a particle that at t=0 belongs to it and if the code cannot find any particle, it is chosen randomly. Thanks to this process we have a better sampling of particles within the radius, respect to a random selection.

- eps=3e-05:

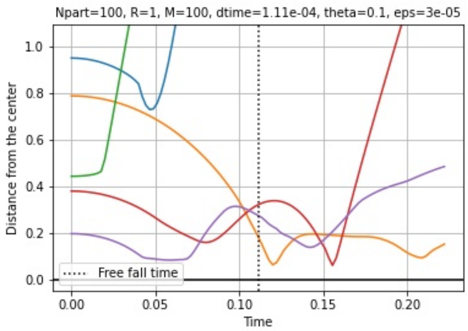
ϵ is very small respect to the mean separation between particles ($10^{-4}\Delta_{mean}$) and so the *gravitational softening* is not so effective since the force between two close particles is not too *'softed'*. This results into the non conservation of energy and in unphysical scatterings between particles as can be seen in the *Position plot* where we can even observe the ejection, and not the collapse,of some particles at $t < t_{FreeFall}$.



(a) Energies plot



(b) Conservation Energy plot

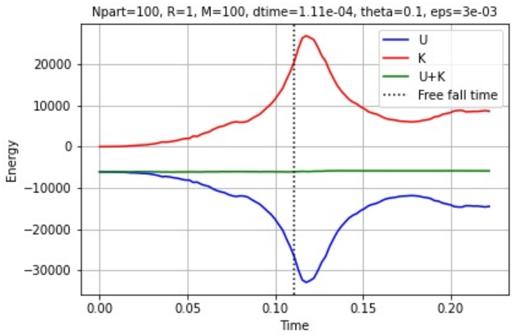


(c) Positions plot

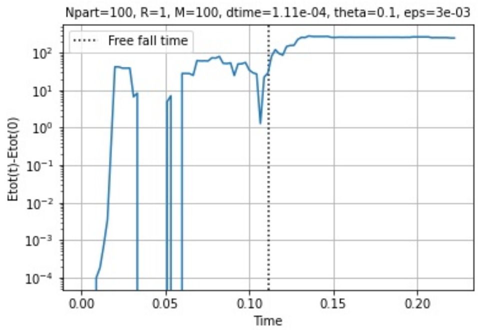
Figure 2: Plots for eps=3e-05

- eps=3e-03:

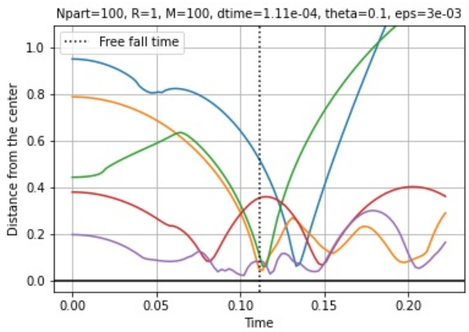
ϵ is small respect to the mean separation between particles ($10^{-2}\Delta_{mean}$) and so the *gravitational softening* is still not so effective: also here we can observe unphysical scatterings between particles, though the energy is better conserved.



(a) Energies plot



(b) Conservation Energy plot

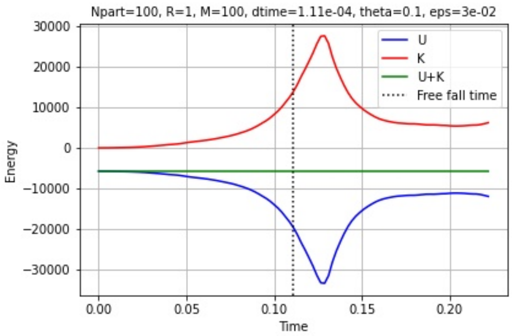


(c) Positions plot

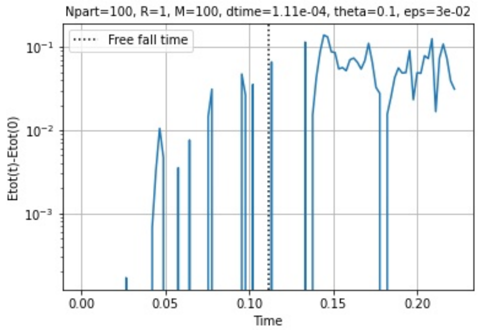
Figure 3: Plots for eps=3e-03

- eps=3e-02:

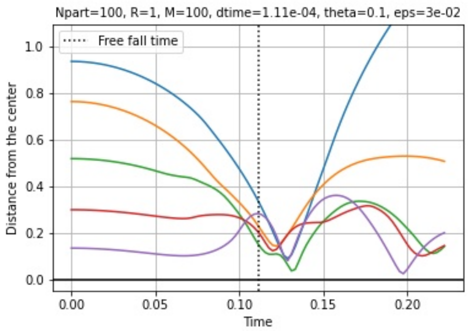
ϵ is now 1/10 of the mean separation between particles and the *gravitational softening* is now effective: the energy is way better conserved and the effects of unphysical scatterings can still be seen (in the Position plot) but with a very low impact compared to before. The minimum distance respect to the center is not reached at $t = t_{FreeFall}$ but later; this is not due to the choice of ϵ but to the number of particles used, as we will see later.



(a) Energies plot



(b) Conservation Energy plot

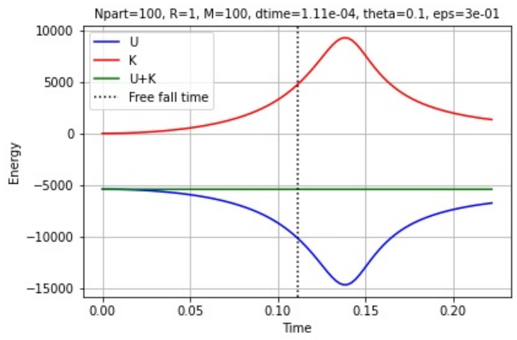


(c) Positions plot

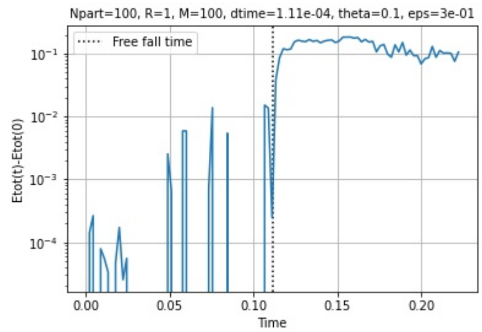
Figure 4: Plots for eps=3e-02

- eps=3e-01:

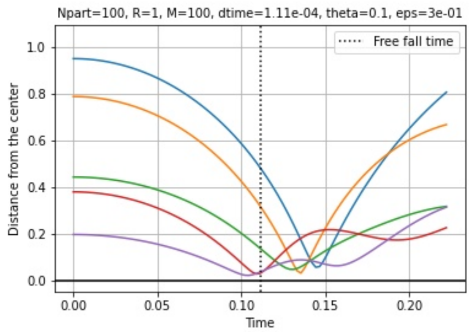
ϵ is now of the same order of the mean separation between particles and the *gravitational softening* is also in this case effective.



(a) Energies plot



(b) Conservation Energy plot

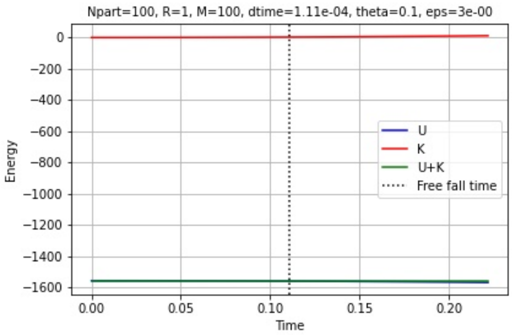


(c) Positions plot

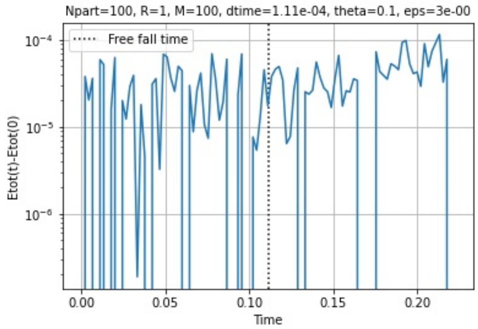
Figure 5: Plots for eps=3e-01

- eps=3e-00:

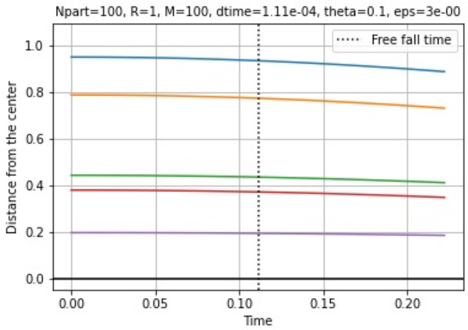
ϵ is now of 10 times greater than the mean separation between particles but now the newtonian force is too softened and the *gravitational softening parameter* has an influence also when two particles are not close one to each other; this result in a quasi-static situation, as can be seen in the *Position plot*.



(a) Energies plot



(b) Conservation Energy plot



(c) Positions plot

Figure 6: Plots for eps=3e-00

From the analysis done we think that the best value of the *gravitational softening parameter* has to be chosen in the interval

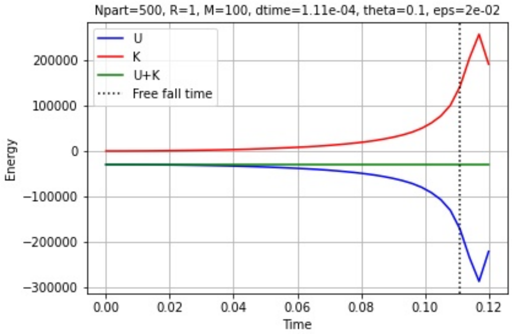
$$10^{-1}\Delta_{mean} \leq \epsilon \leq \Delta_{mean}. \tag{4}$$

2.3 theta

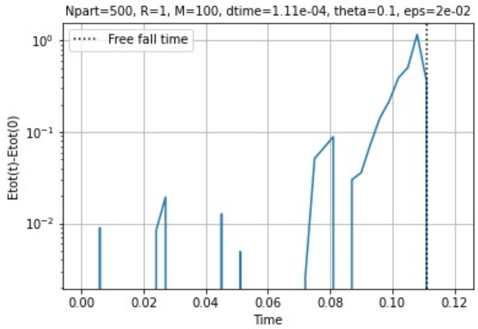
The *theta* parameter is the *opening angle* used to adjust the accuracy of the force calculation; values less than unity produce more accurate forces, albeit at greater computational expense. We executed the code for a sphere of mass $M = 100$ and a radius $R = 1$ with

```
dtime=1.11e-04   eps=2e-02
```

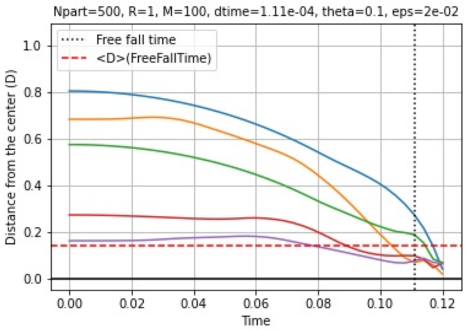
and using three different values of theta (0.1, 1, 1.6). We report the plots of the energies (potential, kinetic and total), the conservation of energy (difference between the total energy at any time and the initial total energy), the evolution over time of the distance of five sample particles and the theoretical vs. the computed path over time of a single particle.



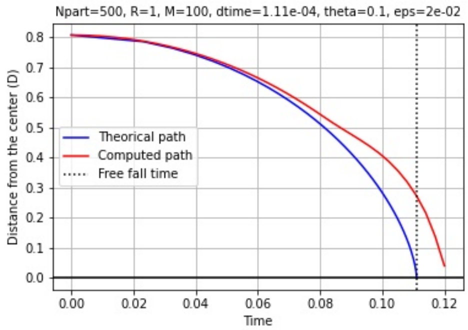
(a) Energies plot



(b) Conservation Energy plot

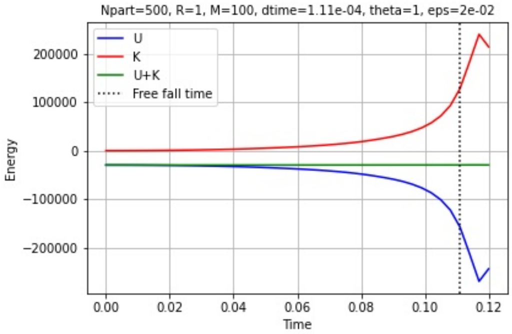


(c) Positions plot

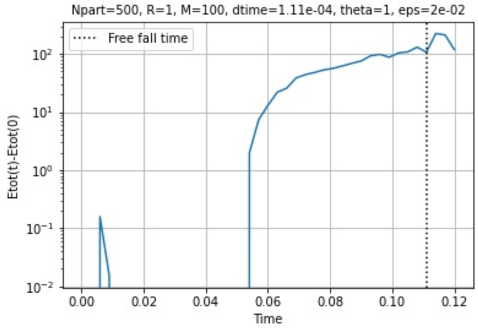


(d) Theorical vs. Computed

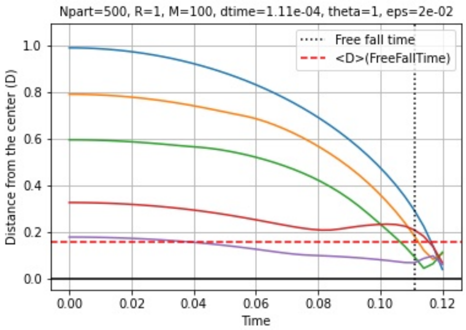
Figure 7: Plots for theta=0.1



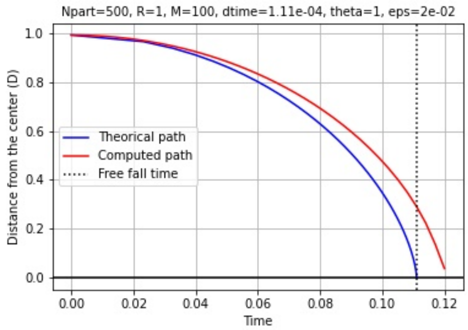
(a) Energies plot



(b) Conservation Energy plot

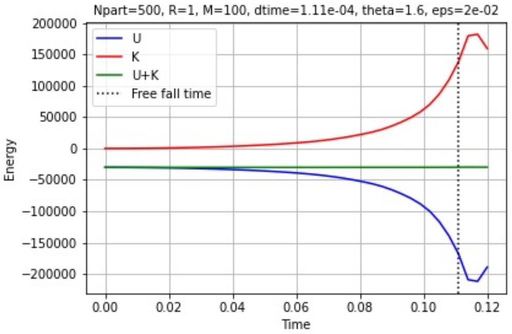


(c) Positions plot

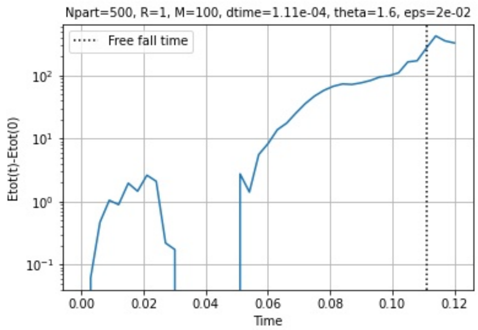


(d) Theorical vs. Computed

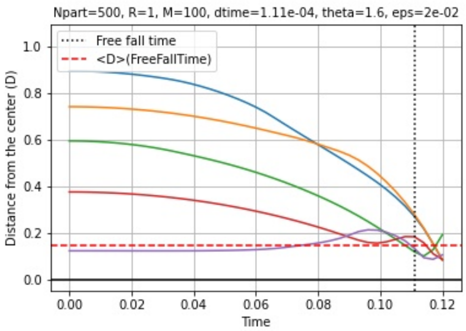
Figure 8: Plots for theta=1



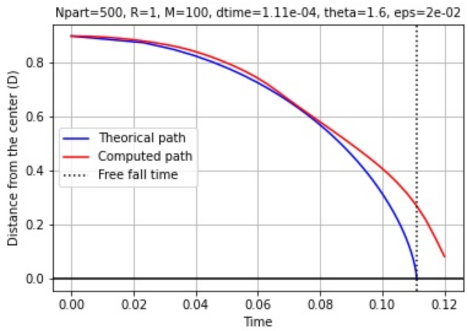
(a) Energies plot



(b) Conservation Energy plot



(c) Positions plot



(d) Theorical vs. Computed

Figure 9: Plots for theta=1.6

Analysing the plots in a qualitative way we conclude that they are very similar one to each other; it can be noted that for $theta = 0.1$ the energy is better conserved. For the simulations done, we decided to use $theta = 0.1$.

2.4 tstop

The *tstop* parameter is the time at which the N-Body integration terminates; we chosen it as a multiple of the free fall time.

2.5 dtout

The *dtout* parameter is the time interval between the saving of two complete description of the system.

3 Study of the system's evolution

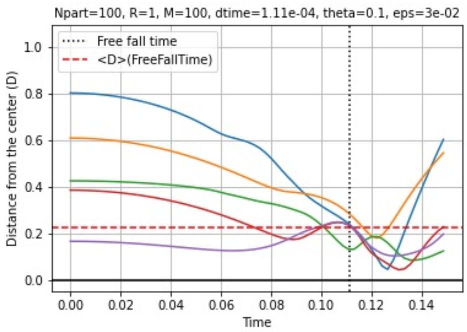
We will now study the evolution of the system comparing the results to the predicted ones; we are approximating an homogeneous sphere of fixed mass and radius with a finite number of particles; as already said in the previous section, a bigger number of particles will result into a better approximation of the sphere since the number density of particles will be greater and into a better representation of the collapse since the mass of each particle will be smaller and so the effects of unphysical scatterings will be smoothed; however, increasing the particles number will increase the computational cost, that is of the order *O(N log N)*.

We are now going to analyse the improvement in the collapse that is obtained using a greater number of particles. The plots presented are obtained through the usual jupyter notebook *Collapse Homogeneous Sphere - Analysis Code (Barnerstreecode)* that uses the output data file created by the C++ code, runned from the terminal in the same way written before.

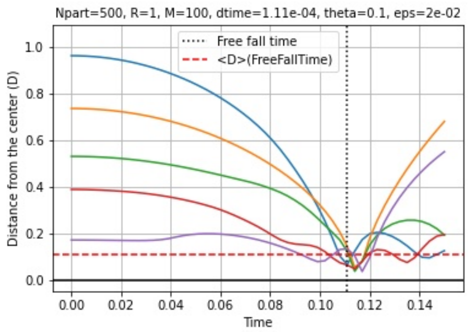
These simulations are made with the same sphere's parameters ($M = 100$, $R = 1$), with the same *dtime* (0.000111) and *theta* (0.1); for each simulation we changed the number of particles and we chose an *eps* of the order of $10^{-1}\Delta_{mean}$.

For each simulation we compute the mean distance, respect to the center, at the snapshot time that is closer to the sphere's free fall time of 100 samples (selected randomly among the particles). We know that at $t = t_{FreeFall}$ the homogeneous sphere is collapsed and so our expectation is that the mean distance just calculated decreases as the number of particle increases, since to a greater number of particles corresponds a better approximation of the problem.

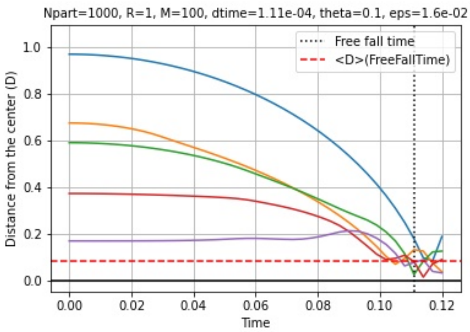
The following plots represent the evolution over time of the distance respect to the sphere's centre of five sample particles; the horizontal red dotted line is the mean position of 100 sample particles at $t = t_{FreeFall}$ (ideally it should coincide with the horizontal black bold line, that means $distance = 0$). The number of particles and the parameters' value is written on the top of each figure.



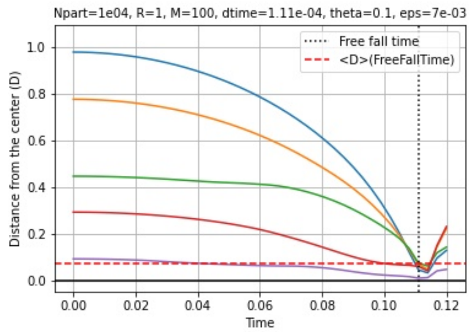
(a) 100 Particles



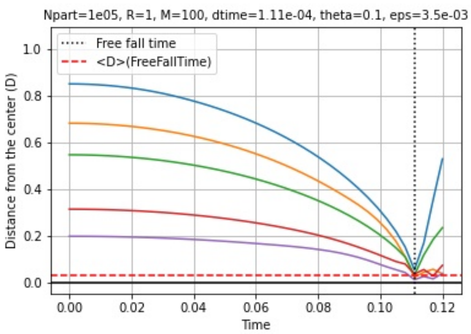
(b) 500 Particles



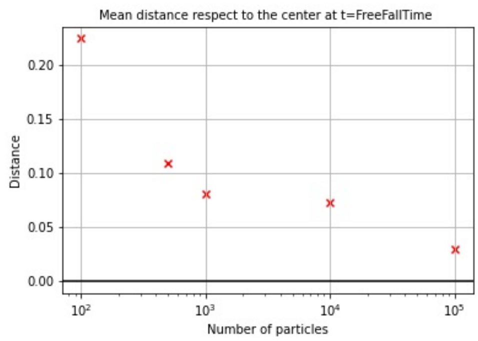
(c) 1'000 Particles



(d) 10'000 Particles



(e) 100'000 Particles



(f) Mean distance vs. Particles number

Figure 10: Distance from the center vs. Time

From the plots we can conclude that the particles' mean distance respect to the center at the free fall time decreases as the number of particles increases, as expected.

Given an homogeneous sphere that undergoes a free collapse under its own gravitational field, we can compute the evolution over time of a spherical shell's radius solving the equation

$$\frac{d^2r}{dt^2} = -G\frac{M(r_0)}{r^2} \tag{5}$$

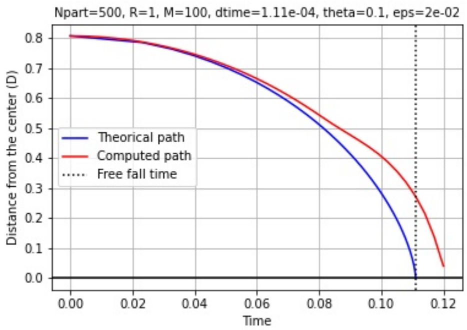
where $M(r_0)$ is the mass enclosed in the spherical shell's initial radius (since the system is free to collapse, all scales together and $M(r_0)$ do not change over time).

The integration of the differential equation will lead to

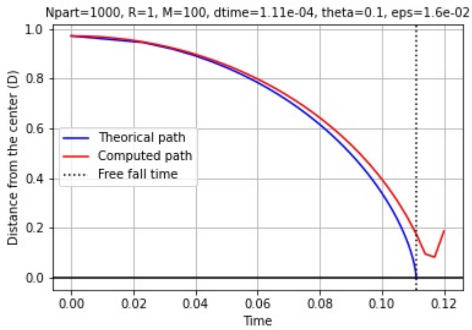
$$\sqrt{\frac{8}{3}\pi G\rho_0} \quad t = \frac{\pi}{2} + \sqrt{\frac{r}{r_0}(1-\frac{r}{r_0})} - \arcsin\sqrt{\frac{r}{r_0}} \tag{6}$$

where ρ_0 is the sphere's mean density.

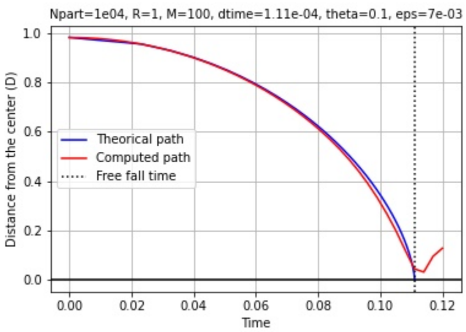
We use this equation to compute the theorical path of a particle over time; in the plots below are plotted the theorical and the computed path for a single sample: the greater the number of particles is,the greater is the accuracy, as expected.



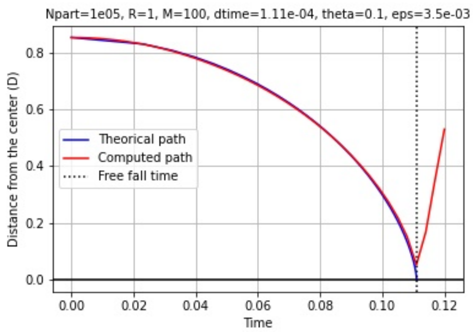
(a) 500 Particles



(b) 1'000 Particles



(c) 10'000 Particles



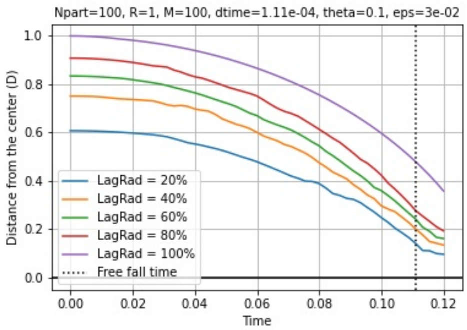
(d) 100'000 Particles

Figure 11: Theorical vs. Computed

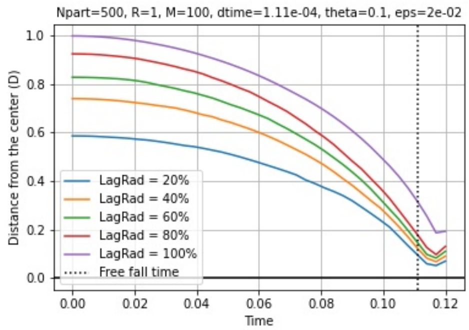
We then analyze the evolution over time of the so called *Lagrangian Radius*, which is the radius that contains a certain percentage of the mass. To compute it, at each time step we saved the distances respect to the center of all the particles, we ordered the distance array and selected the particle that corresponds to the fraction of mass desidered for the Lagrangian radius (e.g. the Lagrangian radius that contains the 25% of mass is equal to the distance from the center of the $(0.25 * Nparticles)^{th}$, since the particles are equal mass; therefore we need to select the cell whose index is equal to $(0.25 * Nparticles - 1)$ in the ordered distance array).

We decided to analyze the lagrangian radius for 100, 500 and 1000 particles because for an higher number of particles our computer takes to much time (the data and the plots are made using the usual jupyter notebook).

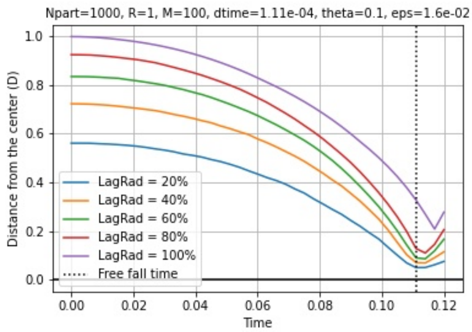
In the following plots are presented the evolution for the Lagrangian radius that embed the 20%, 40%, 60%, 80% and 100% of the mass; also here, to a greater number of particles correspond a better description of the collapse. Note also that one of the advantages of plotting the lagrangian radius is that the unphysical scatterings are not so present (compare, for example, the plot of the lagrangian radius for 100 particles with the one of the five particles' distance from the center, calculated with the same parameters).



(a) 100 Particles



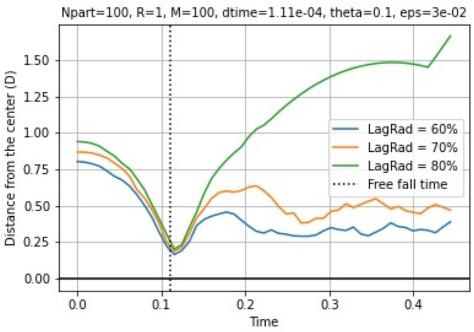
(b) 500 Particles



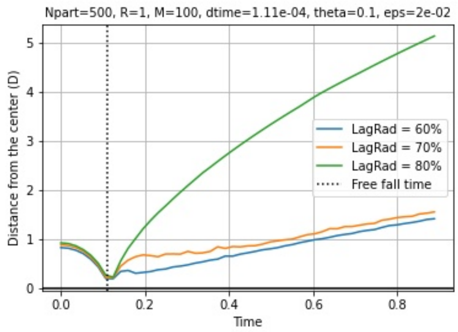
(c) 1000 Particles

Figure 12: Lagrangian Radius' evolution

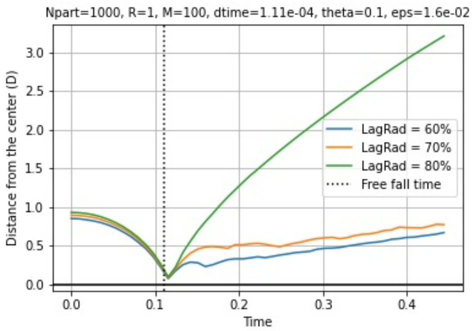
In the following plots are presented, instead, the evolution of the Lagrangian radius that embed a percentage of the mass which is greater then the 50%; the evolution, in this case, is calculate for a longer time in order to understand qualitatively what happens numerically after the collapse: as can be seen in the plots, after the free fall time, the sphere starts to expand and between the 20% and the 30% of particles are lost by the system (this can be noted analysing the different trend of the Lagrangian radius evolution).



(a) 100 Particles



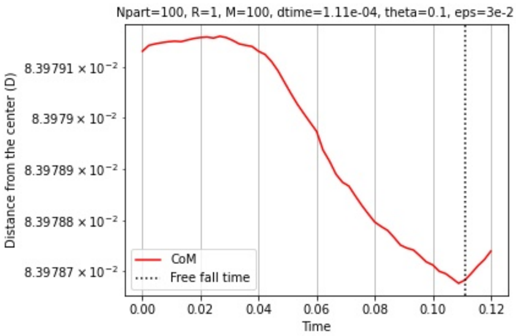
(b) 500 Particles



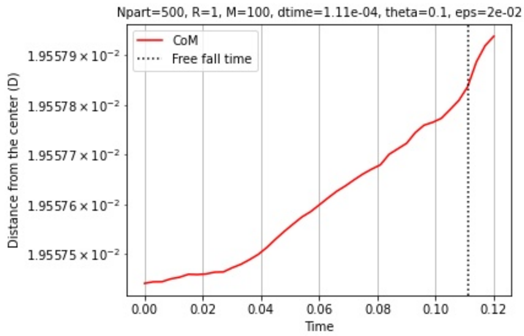
(c) 1000 Particles

Figure 13: Lagrangian Radius (long) evolution

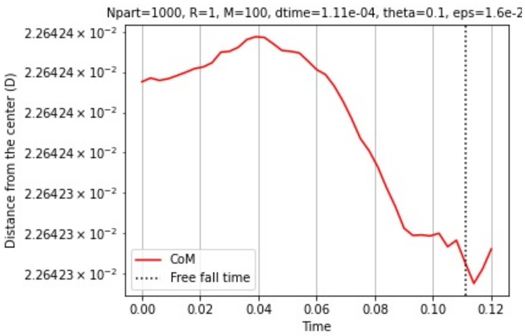
We conclude our analysis with the evolution over time of center of mass' distance respect to the centre of the sphere; in principle, since we are dealing with an homogeneous and isolated sphere, its center of mass should be and remain in $(x, y, z) = (0, 0, 0)$: the results obtained are in agreement with what we have just stated.



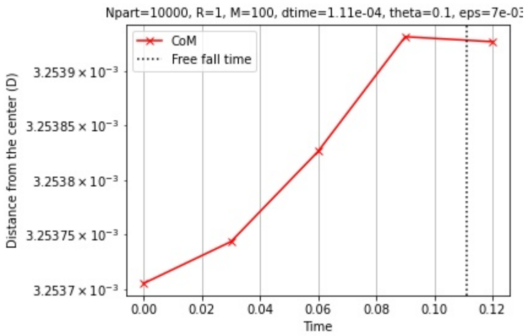
(a) 100 Particles



(b) 500 Particles



(c) 1'000 Particles



(d) 10'000 Particles

Figure 14: Center of Mass