Computational Astrophysics 2013
Assignment 4: Hydrodynamics

Bernie Lau
Leiden University

Luuk Visser
Leiden University and
Delft University of Technology

April 1, 2013

Abstract

For the fourth assignment we have to work with a hydrodynamics code. A stars behaves like liquid spheres when interacting with another star, and those we treat with hydrodynamics. Gravity is obviously also important, but for this experiment we adopt the built in gravity solver in the hydrodynamics code. We will evolve so called Plummer spheres and also smash two Plummer spheres. Animations of the simulations are also available.

1 Introduction

For this assignment we need to work with the hydrodynamics codes within the AMUSE framework. We will use the internal gravity code within the hydrodynamic solvers to do our simulations. In the next section we will show how the code can be executed using the command line. In the third section we will handle all sub-assignments and their respective questions.

A few notes on the directories and files in the final package:

- All plots required for the assignment can be found in hydrodynamics/plots
- There are animations of the simulations in the folder hydrodynamics/movies which we have created just for fun.
- There is a file called 'generate_examples.py' that can be run which includes the examples above.
- ffmpeg is required to create animations with 'animations.py'. The only version that has been tested to work is version 1.2.
- The hdf5 files in the directory 'bodies' contain different information from the hdf5 files in 'hydroresults'. The former contains snapshots of particle classes, while the latter contains data to create plots/animations.

2 Commands to execute the code

The information in this section can also be found in the README file.

2.1 Evolve plummer models

One can run the hydrodynamics simulations of the plummer models, using the following commands.

```
% mpiexec amuse evolve.py -N 4000 -n 100 -t 1.0 -H evolveN4000n100t1.hdf5
% mpiexec amuse evolve.py -N 2000 -n 300 -t 0.5 -H evolveN2000n300t0.5.hdf5 -B bodyN2000n300.h
% mpiexec amuse evolve.py -N 1000 -n 100 -t 0.5 -B bodyN1000n100.hdf5
```

This will generate the following bodies in the default directory 'bodies/':

```
bodies/bodyN2000n300.hdf5
bodies/bodyN1000n100.hdf5
```

And the following hydroresults in the default directory 'hydroresults/':

```
hydroresults/evolveN4000n100t1.hdf5
hydroresults/evolveN2000n300t0.5.hdf5
```

2.2 Smash plummer models

One can do the smashing of two plummer models using the following commands.

```
% mpiexec amuse evolve.py -n 200 -t 1 -H smash_vx20.hdf5 \
    -p bodies/bodyN1000n100.hdf5 -q bodies/bodyN1000n100.hdf5 --vx 20 --vy 0 --vz 0

% mpiexec amuse evolve.py -n 200 -t 1 -H smash_vx76.hdf5 \
    -p bodies/bodyN1000n100.hdf5 -q bodies/bodyN1000n100.hdf5 --vx 76 --vy 0 --vz 0
```

This will generate the following hydroresults in the default directory 'hydroresults/'

```
hydroresults/smash_vx20.hdf5
hydroresults/smash_vx76.hdf5
```

2.3 Create animations

One can create the animations in a mp4 file, with the following commands.

```
% mpiexec amuse animation.py -f hydroresults/evolveN2000n300t0.5.hdf5 -r 2
% mpiexec amuse animation.py -f hydroresults/smash_vx20.hdf5 -r 4
% mpiexec amuse animation.py -f hydroresults/smash_vx76.hdf5 -r 8
```

This will generate the following animations in the directory 'movies':

```
evolveN2000n300t0.5r2.0.mp4
smash_vx20r4.0.mp4
smash_vx76r8.0.mp4
```

In the final package, we included six movies of different simulations.

3 Answers to the questions

In this section we treat the three questions/subassignment.

3.1 Generate Plummer gas spheres

3.1.1 Assignment

A stars behaves like liquid spheres when interacting with another star, and those we treat with hydrodynamics. Gravity is obviously also important, but for this experiment we adopt the built in gravity solver in the hydrodynamics codes (that is easier, and sufficient for our current assignment). Write a script that generate a single plummer distributions of gaseous (SPH) particles, with mass m and characteristic size r. In the script *hydrosimple.py* in the directory *example/syllabus* you can see how to do this. Adopt an adiabatic equation of state for the gas and make sure to turn self gravity on. Run your first experiments with as few particles as possible, just to get the code working and get some feeling for the code. Is your model stable with $N = 10$ particles, and what about for $N = 100$ and $N = 1000$? Plot the wall-clock time for running the simulation for one time-step as a function of N. And how does the energy error in your integration vary with N.

Source: Assignment 4 from webpage Computation Astrophysics 2013 [1].

3.1.2 Answer and result

The models are stable for 100, 1000 particles and probably above as well, but not for 10 particles.

The runtime as function of the number of particles can be seen in Figure 1.

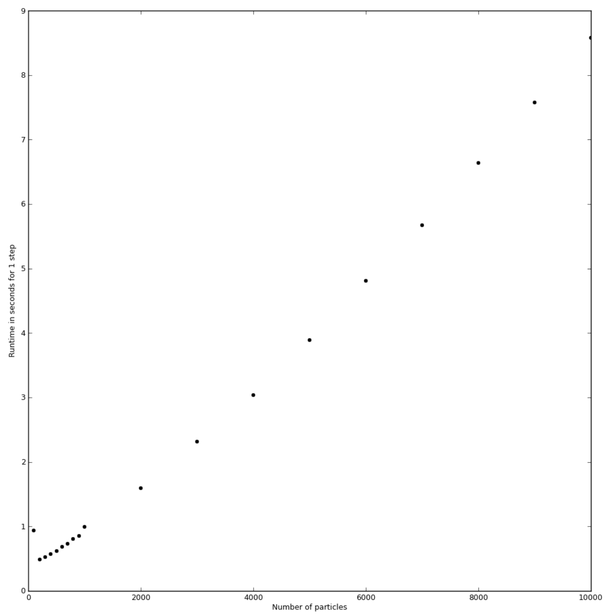

Figure 1: Runtime as function of number of particles N.

The error as function of the number of particles can be seen in Figure 2.

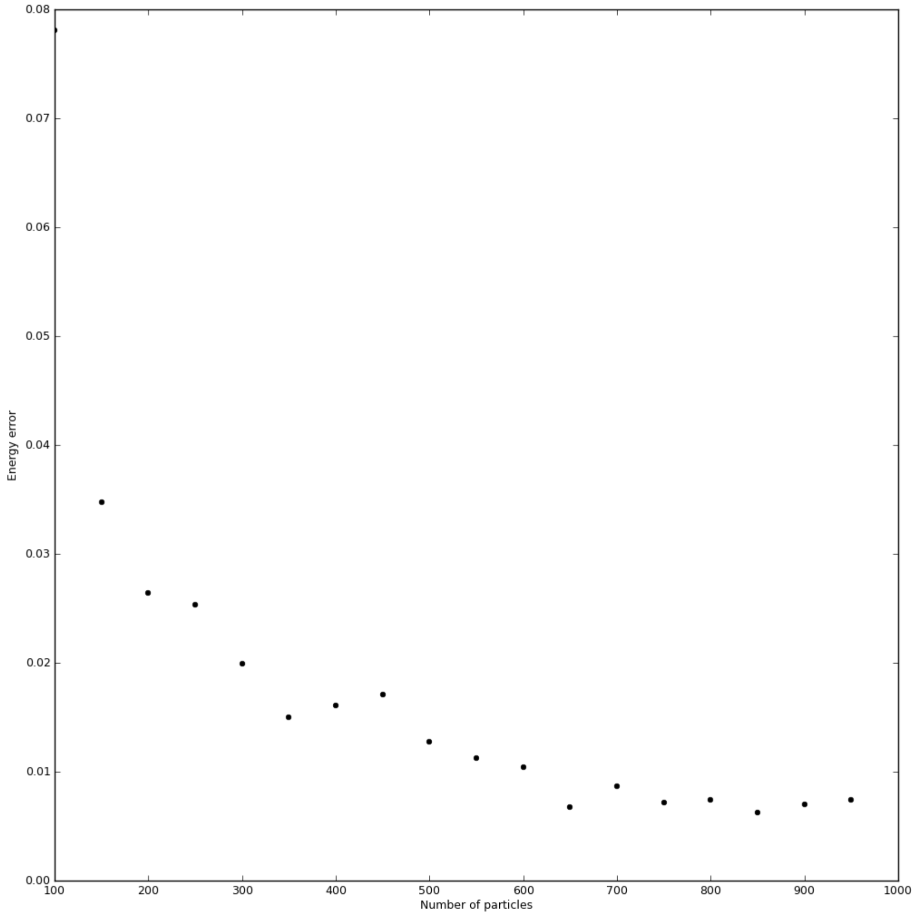

Figure 2: Integration error as function of number of particles N.

3.2 Run until equilibrium and convergence

3.2.1 Assignment

After you have acquired a stable initial configuration, increase the number of SPH particles for a more robust result. Plot as a function of time the angular momentum, and plot the initial and final radial density profile. Calculate the kinetic and potential energies and compare them with the initial value. Instead of plotting a large number of density profiles in one frame, one often refers to Lagrangian radii; those are the distance from the center of mass which contain $x\%$ of the initial mass. In AMUSE there is a standard routine for calculating Lagrangian radii. Use this function to plot, as a function of time the 10%, 25%, 50% and 75% Lagrangian radius of your gaseous sphere. You have a converged model if your resulting density profile becomes independent of N. You can test this by making a plot, printing the numbers but the best way is by performing a Kolmogorov-Smirnoff test. The KS test (for short) returns a probability that one model is a coincidental random representation of your other model. It is a handy test to remember.

Source: Assignment 4 from webpage Computation Astrophysics 2013 [1].

3.2.2 Answer and result

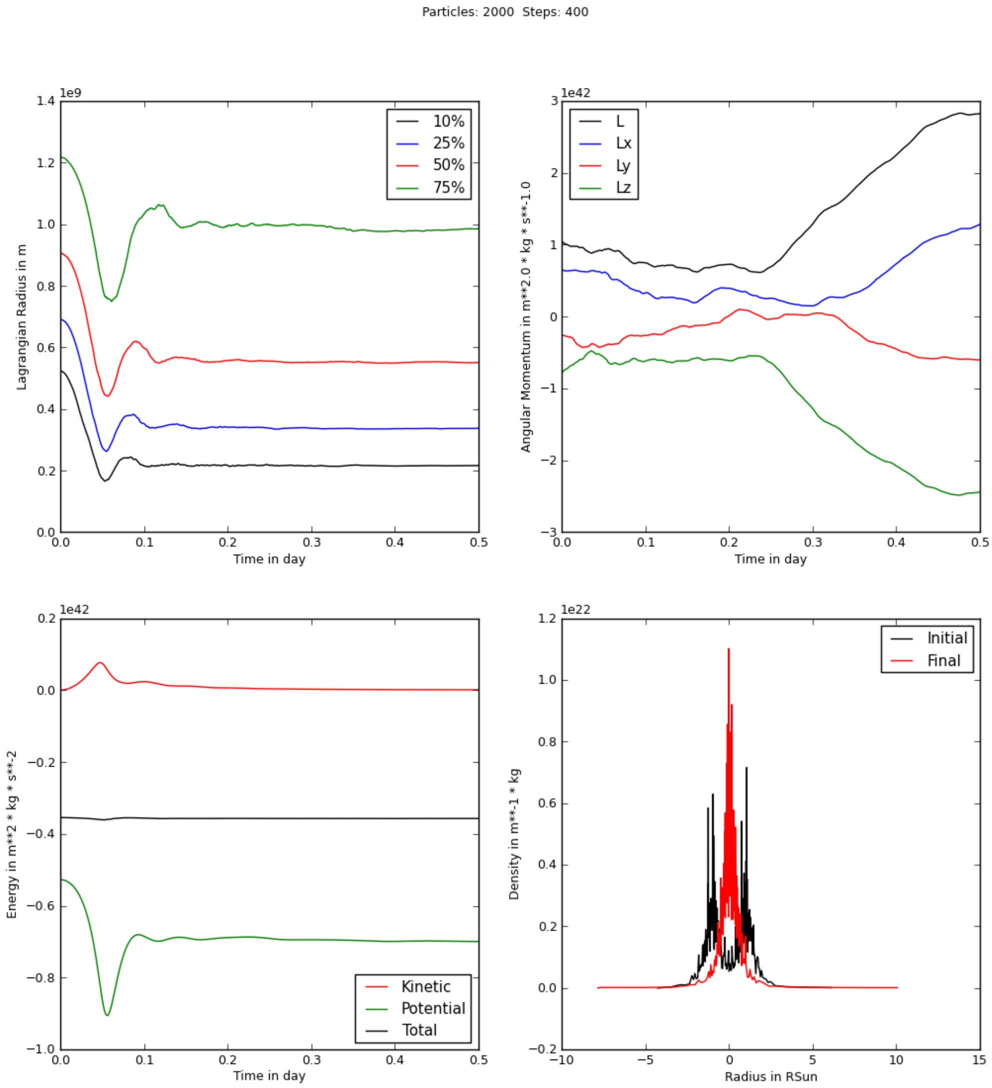

Figure 3: Evolution of a Plummer sphere with with 2000 particles, 400 time steps and no initial velocity.

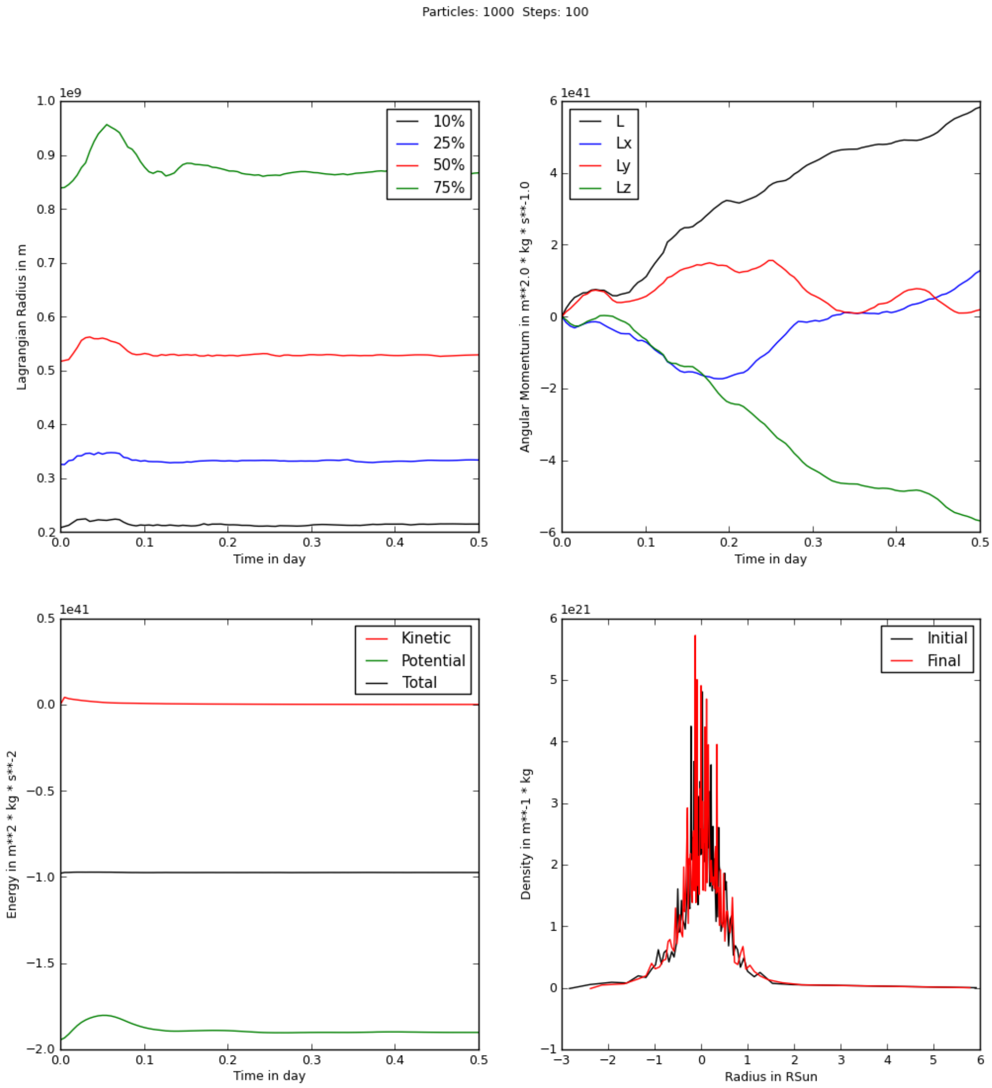

Figure 4: Evolution of a Plummer sphere with 1000 particles and 100 time steps.

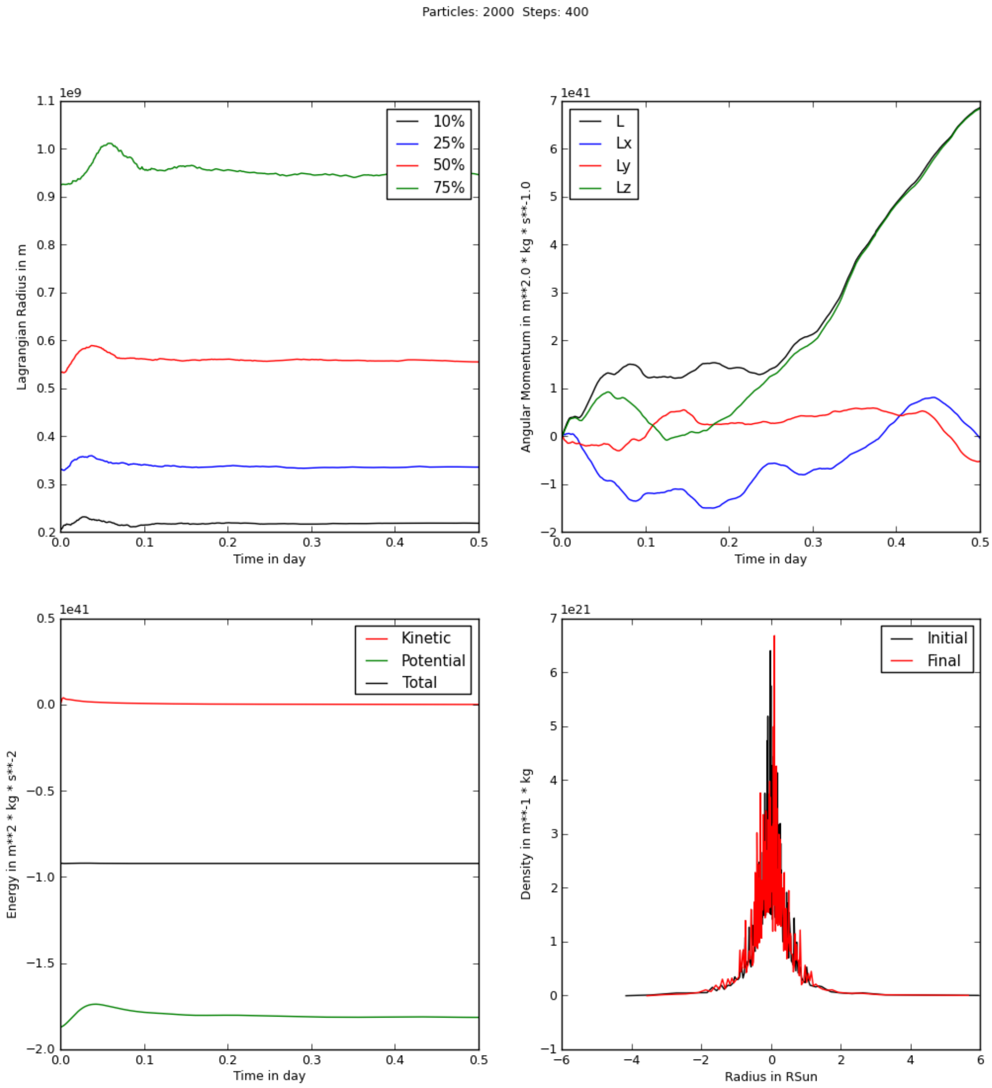

Figure 5: Evolution of a Plummer sphere with 2000 particles and 300 time steps.

3.3 Smash them up

3.3.1 Assignment

New generate a second Plummer sphere (or make a copy of your first one), and smash them into each other with radial velocity v from a distance $2r$. At first instance adopt $v = 0\ km/s$, but increase it until you obliterate the two gaseous bodies. Of course, then there is no equilibrium model, and therefore you will have to stop the code if you become impatient. Plot as a function of v the Lagrangian radii of the merger product (simulated until it is in equilibrium and for the converged solutions, if possible). Can you predict (or postdict) what happens if $v^2 \simeq 2Gm/r$?

Source: Assignment 4 from webpage Computation Astrophysics 2013 [1].

3.3.2 Answer and result

The smashing of two plummers with the escape velocity $v^2 = 2GM/r$ which is equal about to 76 Solar radii/per day does not seem to obliterate the two spheres.

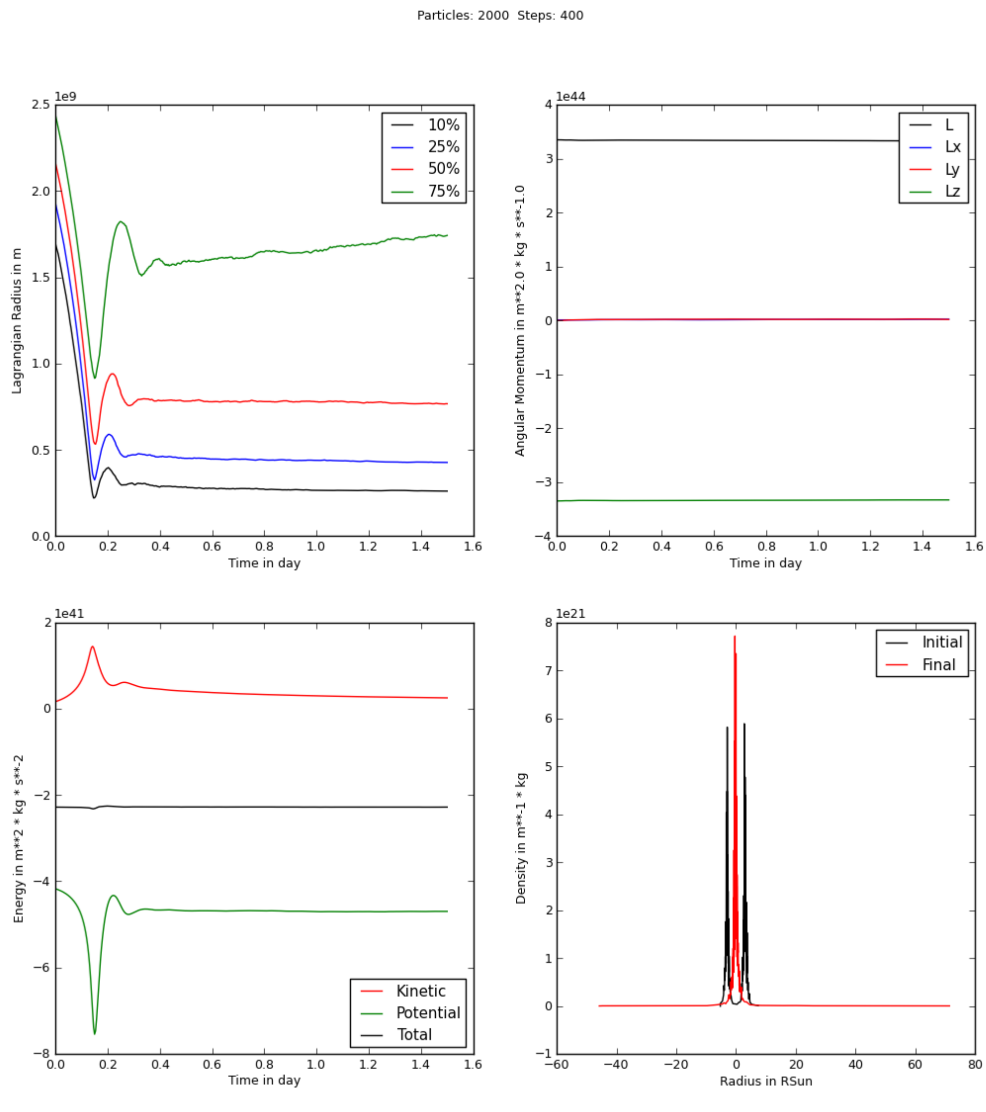

Figure 6: Smash of two Plummer spheres with vx=10 and xy=5 (both in Solar radii/per day).

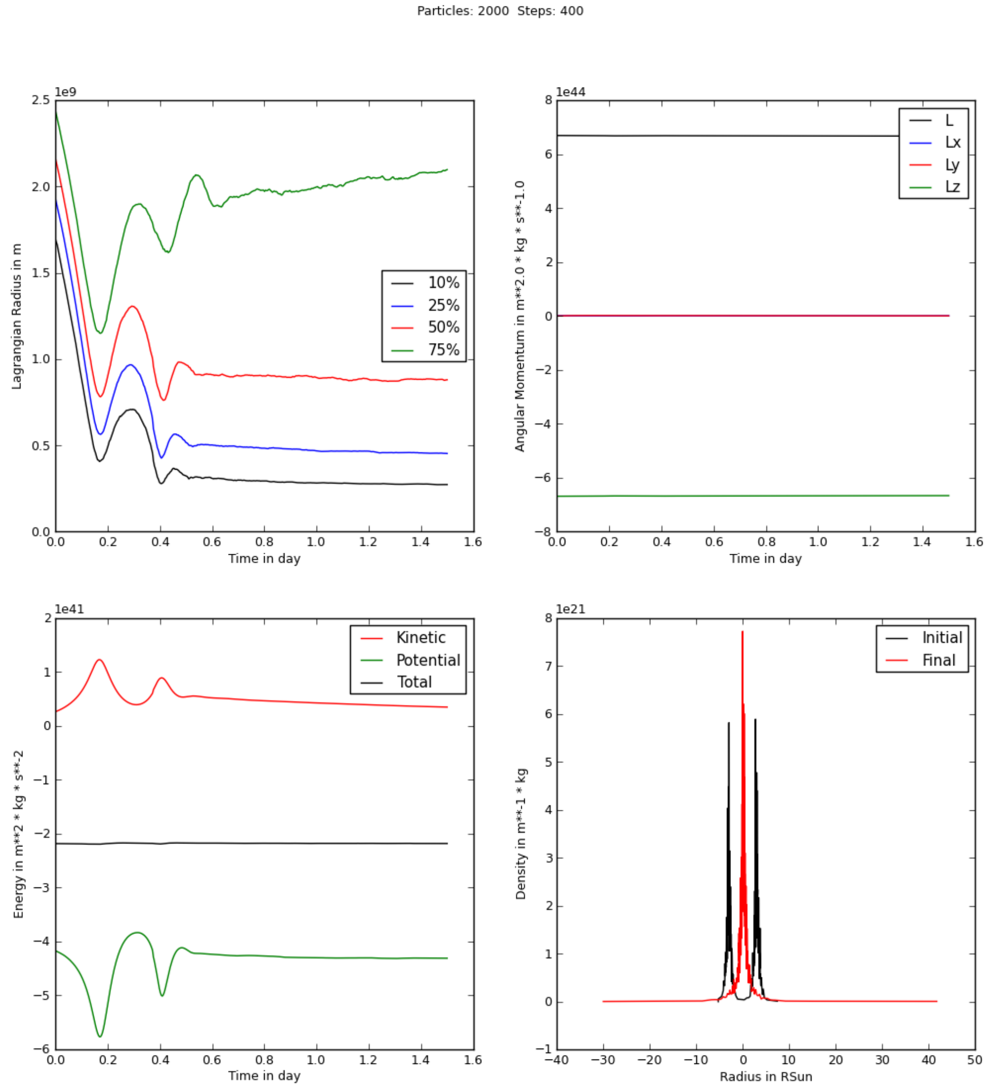

Figure 7: Smash of two Plummer spheres with vx=10 and xy=10 (both in Solar radii/per day).

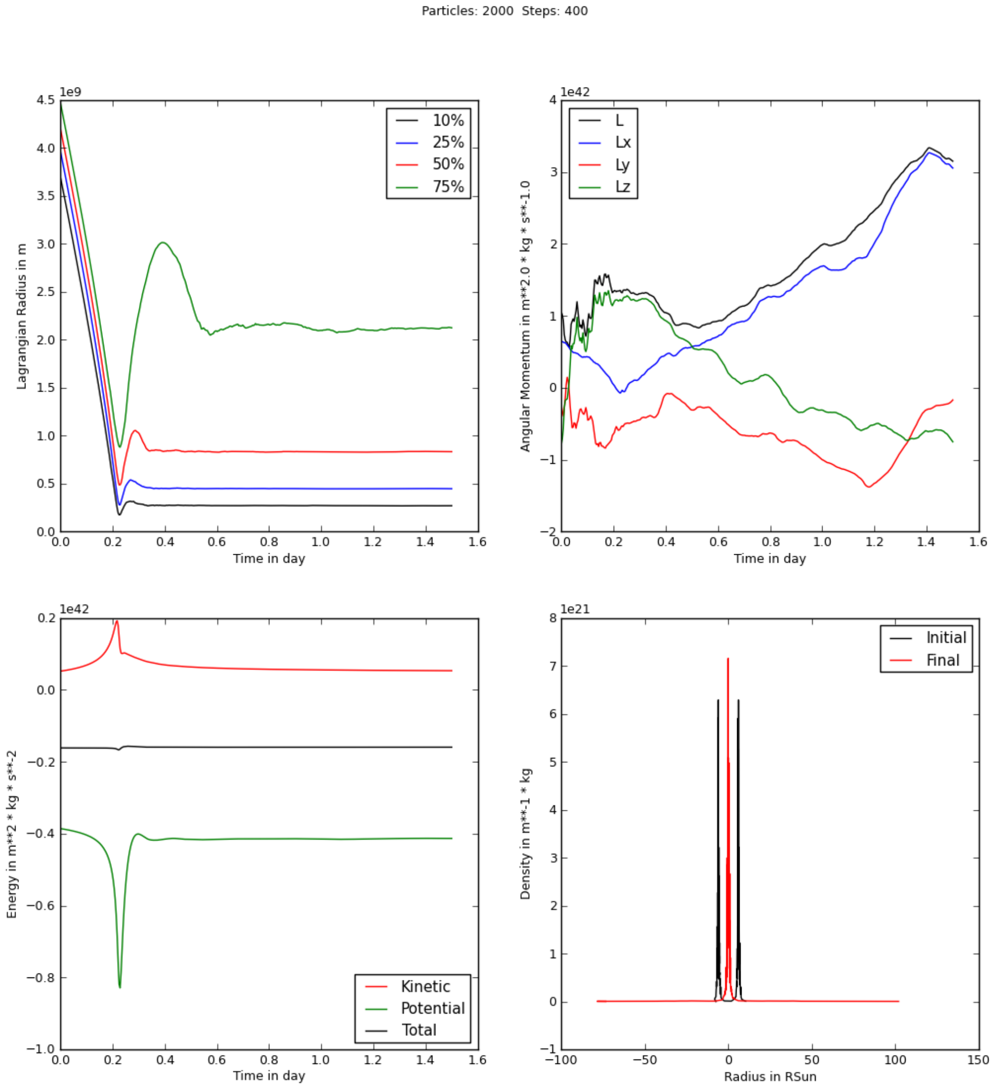

Figure 8: Smash of two Plummer spheres with vx=20 and xy=0 (both in Solar radii/per day).

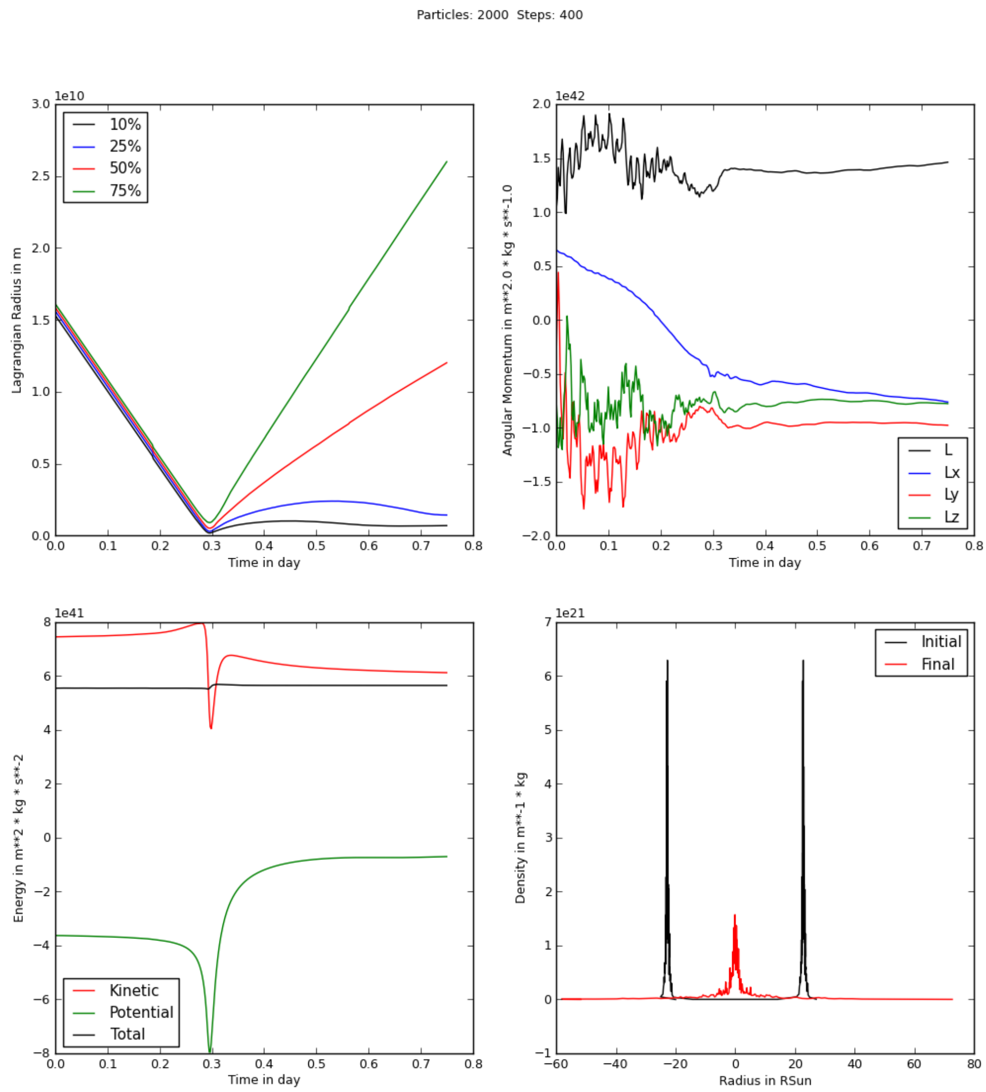

Figure 9: Smash of two Plummer spheres with vx=76 (escape velocity) and xy=0 (both in Solar radii/per day).

4 Conclusion

Using AMUSE and the available hydrodynamics code, we can simulate the hydrodynamics and gravity of Plummer spheres. The error will decrease with more particles, but the runtime will also increase about linearly with the number of particles.

We can evolve the the Plummer sphere over time, and also smash multiple Plummer spheres to see what that will do with the integrity.

References

[1] CA 2013 document, Assignments 4, 2013-02-26, http://castle.strw.leidenuniv.nl/ca2013