

Riod: A Many-Body Gravitational Simulation

Description and Features

<https://riodsim.weebly.com/>

August 30, 2021

Contents

- Riod Simulation Description
- Features
- Initial Particle Configurations
- Initial Particle Velocities
- Discussion of collision control and force softening
- Visualization and analysis tools
- All Riod details, history and features can found in the users manual at my website: <https://riodsim.weebly.com/>

The Riod Simulation Features

- Windows 64bit executable
- Multi-threaded code
- Full $N(N-1)/2$ particle pairs have forces computed for each iterative cycle. This means there are no shortcuts taken and thus getting results require patience.
- “Unlimited” numbers of particles possible (best if limited to under 100K). The largest simulations I have run are with 50,000 particles.
- Many methods of creating initial conditions, (Density profiles include uniform, shells, flattened disks, EXn, NFW, Jaffe and more.
- Many collision options including elastic, inelastic, Plummer force softening, EX3:1 force softening and other more exotic types.
- Rich, configurable data logging
- Visualization and analysis tools
- Many other features. See the manual!

Initial Particle Positions and Configurations

- In Riod terminology, an elemental particle is referred to as a standard object or SO. A collection of SO is referred to as an extended object or EO.
- Simulations can begin with just a single EO or one can make multi-EO configurations.
- The distribution of SO within and EO can be provisioned to take on many different density profiles.
- Just each EO has a density profile, the total collection of EO in multi-EO configurations can be configured to have many different density profiles.

Initial Particle Configurations (Cont.)

- Below the mass distributions of SO will be discussed and for the sake of brevity, I will use the following notation for exponential forms.

$$EXn:m = \rho(x) = \rho_0 e^{-x^{n/m}}$$

where $x=r/r_s$ and r_s is the scaling length.

- Using this notation, I will refer to a Gaussian distribution as EX2:1, for example.

Riod Simulation: Force softening details (cont.)

Let's examine some of the properties of the EXn:m density function (Note that this EXn notation is my construct to more easily talk about these functions. It is not used in any of the scientific literature that I have seen).

$$\rho(r) = \rho_0 \left[e^{-(r/r_l)^{n/m}} \right]$$

- This density profile is quickly recognized as a variant of the Einasto density profile, where r_l is the scaling length. For example, the Einasto profile is discussed with $n/m \approx 1/6$, which in my notation is $n=1, m=6$.
- When transformed to an $M(R)$ force modification for softening purposes (as in Riod with EX3:1), the force between approaching pairs of particles goes to zero as $r \rightarrow 0$. Recall this transformation is:

$$M(R) = \int_0^R \rho(r) d^3r$$

Initial Particle Configurations (Cont.)

- Distributions for SO and EO can be configured to be any of the following:
 - Exponential density profiles, Einasto-like profiles, EX3:1, EX2:1, EX3:2, EX1:1, and EX3:4.
 - Power law densities like Plummer and others.
 - Other known mass density functions used in the astrophysics literature, like the NFW and Jaffe profiles
- Creating these profiles, one can empirically determine the relationship between the virial radius and the density profile scaling length. For example, an NFW profile has a virial radius that is 19.6 times scaling length.

Initial Velocities of SO and EO

- Initial conditions of the simulation also include how the SO and EO velocities are to be determined. Of course, velocities require a speed and direction.
- First let's talk about direction. The best way to understand the direction is to begin by understanding the velocity direction is based on the particle radial direction. For simulations with a single EO, this is just the radial vector of the SO. For multi-EO simulations, it is the radial vector in the center-of-mass of the EO.
- Also understand that for multi-EO configurations, the velocity of each SO is a combined velocity for the SO and then the EO velocity.
- Both SO and EO have in common that the velocity vector relative to the radial velocity. Then from the radial component, the velocity direction is rotated from zero to 180 degrees; 90 degrees would give a direction orthogonal to the radial vector. Zero. degrees gives a direction that is in the direction of the radius.
- A second rotation out to 360 degrees, will give a rotation about the radial axis allowing a full range of directions for the initial velocity vectors.

Initial Velocities of SO and EO (cont.)

- The speeds of particles are based the idea that all EO or SO will behave initially like an orbiting object.
- For example, an object in a circular orbit about a large gravitational mass will have the following speed,

$$v = \sqrt{\frac{GM}{r}},$$

where r is the orbital distance, G is the universal gravitational constant and M is the large mass.

- We know that if the object is in an elliptical orbit with eccentricity ε , and we start it at its apoapsis or periapsis, the speed becomes:

$$v = \sqrt{\frac{GM(1-\varepsilon)}{r}}.$$

Where in this case the eccentricity can take values: $-1 \leq \varepsilon \leq 1$.

Initial Velocities of SO and EO (cont.)

- Finally, one last modification. If the large central mass is more a collection of masses, the equation becomes only dependent on the total interior mass which is designated as $M(r)$:

$$v = \sqrt{\frac{GM(r)(1-\varepsilon)}{r}}.$$

Here $M(r)$ becomes the sum of all mass inside the radius of the particle. This includes all SO and any static mass profiles that one might configure.

- Thus for SO in a single EO, $M(r)$ is for the interior mass within that EO. For multi-EO scenarios, each EO has a group speed based on the other EO positions.

Riod: The relation Between the Determination of the Iterative Time Step and the Force softening length

- The Riod simulation has an intimate relationship between the iterative time step and the force softening length.
- A solution was needed for determining an integration time step and this led organically to defining the particle size and consequently the interparticle force softening length. The following question was asked:
 - What is the period of orbit for two identical particles orbiting in a circular orbit at their two radii, “ s ”?
 - There is a known solution for this query. Using Kepler’s third law we know that $T^2 \propto a^3$; where T is the orbital period and a is the orbital semi-major axis.

If we divide the orbital period into N time slices and call that our integration time, $\Delta t = T/N$ then:

$$\Delta t^2 N^2 \propto a^3$$

Now, for circular orbits of radius s , the semi-major axis is also the radius.

Thus $s = a$.

Riod Softening Methodology and Iterative Time Steps (cont.)

Finally, filling in the missing constants, we see that for identical particles, Kepler's third law gives us a relationship between the particle size and the integration interval:

$$\Delta t = \frac{\pi}{N} \sqrt{\frac{2s^3}{Gm}}$$

There is much of interest in the above relationship.

- One can specify the particle size to get the time slice or one can specify the time slice and get a particle size. In fact, the Riod simulation gives the operator either option.
- The time slice essentially depends only on the particle density. Thus for unequal masses of the same density, this relationship holds.

After much testing, it was found that a value of $N=10,000$ gives excellent results.

Riod Collision Control

- Since Riod is not collisionless, for cases when SO pairs come close together, the code must try to retain as much of the actual physics as possible but still protect the code from unwanted numerical anomalies.
- As such, the code employs force modifications for interparticle spacings that could become problematic.
- Force modifications include force softening, elastic collisions, inelastic collisions and other collision types.
- The most commonly used force modification is force softening. Softening is used for collisions where essentially, the particles pass through each other and the force is moderated at close distances, rather than approaching infinities.
 - There are two methods of force softening available to the user, Plummer and EX3:1.
 - EX3:1 is highly localized and as such is more desirable as force softening solution.

Riod Simulation: Force softening details

- Given the relationship between the iterative time step and the particle size, it is known that Newtonian forces should be properly accounted for in our gravitational simulation down to radial sizes of twice the SO size. As such it makes perfect sense to begin force softening in separation range of twice the combined SO size.
- The Riod simulation has two methods for force-softening. Since the EX3:1 method is the most localized, it becomes a more desirable solution.
- The interior force ($F_{<}$) modification is based on the EX3:1 distribution and is given by:

$$F_{<} = F_N [1 - e^{-(r/r_l)^3}],$$

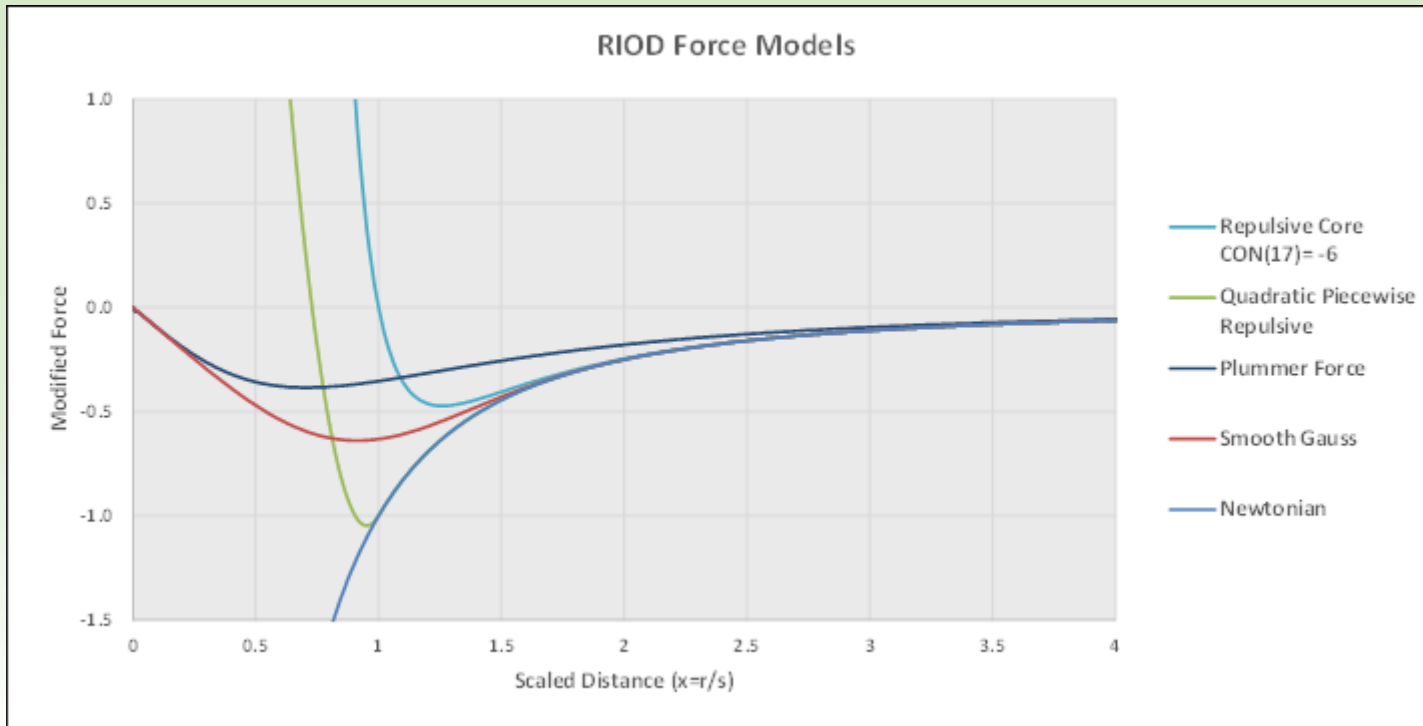
where, F_N is the standard Newtonian force and r_l is the softening length.

Note that $F_{<}/F_N = 0.999$ when $r = 1.90 r_l$; thus shutting of force softening at $r = 2r_l$, is a reasonable compromise.

Collision Control and Force Softening (cont.)

The force modifications are visualized below, with the following notes.

- Note that the EX3:1 (also called smoothed Gauss) force softening converges to the Newtonian force for $r/r_p > 1.9$.
- The Plummer softening method does not converge to the Newtonian force until 20 times farther out, or $r/r_p > 39$.
- The Lennard-Jones modification (called repulsive core) works well enough for certain powers of r and converges quickly to the Newtonian force.
- The piece-wise continuous elastic collision force modification has issues with its implementation and thus is not recommended for use.



Virial Radius Discussion

- The virial radius is an important and useful quantity to help understand these evolving gravitational systems.
- The definition below of the virial radius is directly related to the total potential energy of the system and from the virial theorem, we know that once the virial state is reached, the virial radius will be constant (inside a time averaged window).
- For my usage, the following is the definition for the virial radius, r_v :

$$\frac{1}{r_v} = \frac{2}{M_T^2} \sum \frac{m_i m_j}{|\vec{r}_j - \vec{r}_i|} = \frac{2|U|}{M_T^2 G},$$

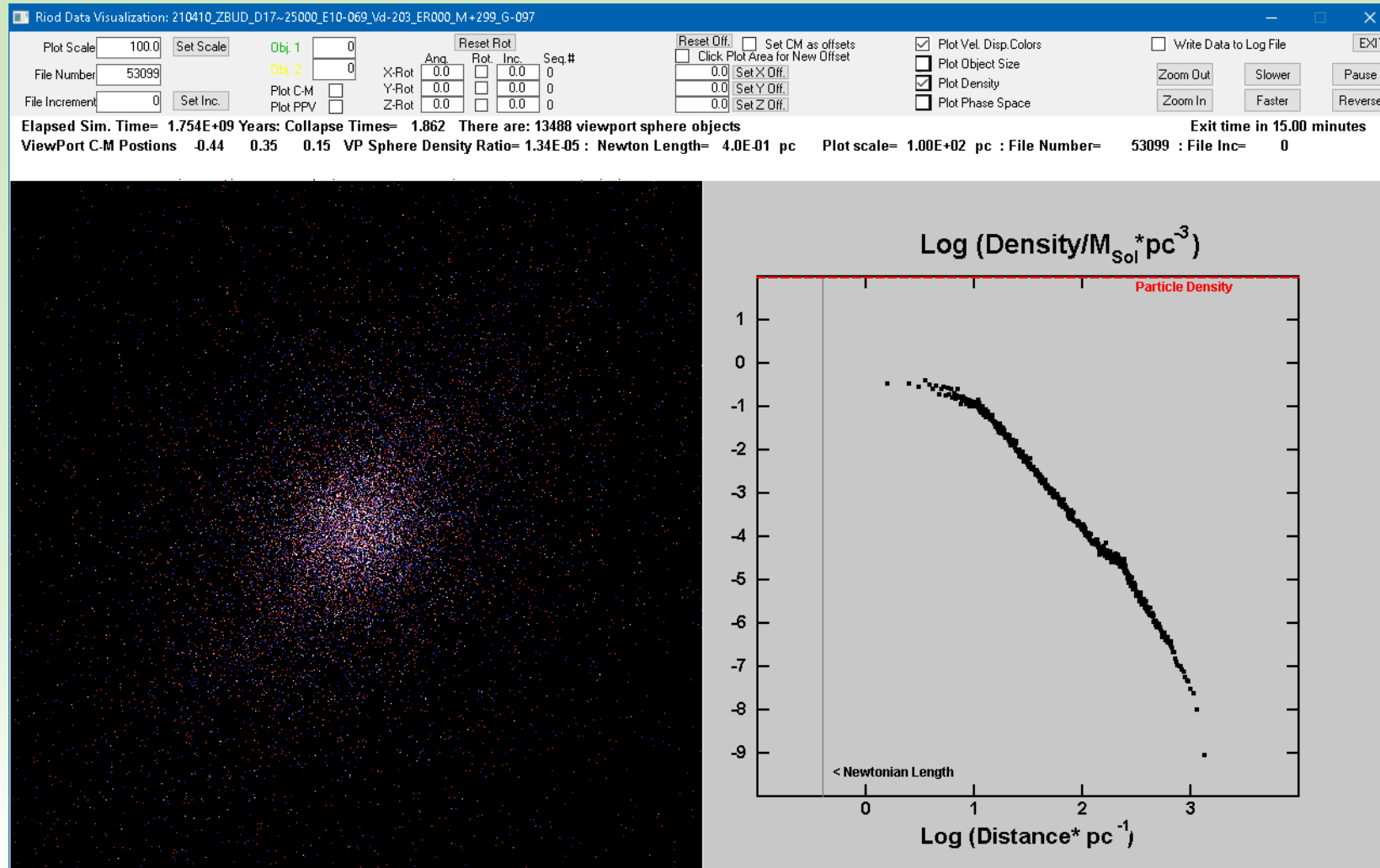
where U is the total system potential energy. These quantities are easily (but not always quickly) computed from the simulation data.

Visualization and Analysis Tools

- I have created auxiliary programs and spreadsheets to help understand and analyze the simulation results. All these tools are in a constant state of “improvement”.
 - The visualization program (Rplot.exe) is the most useful as it will “animate” the saved position data, plot a density profile, rotate about the three axis, zoom in or out, and a many other features. One can also create screen captures so that videos can be made.
 - Stats.exe will output the latest properties of all SO in the simulation. Properties include mass, size, distance out, speed, nearest neighbor and others.
 - Hist.exe will take a position file or a set of position files and create a histogram that one can extract the density profile and other properties.
 - Fits.exe will fit the $M(r)$ function from a histogram file using a set of up to three density profile functions.

Visualization and Analysis Tools (cont.)

- Here is a screen capture of the Plot.exe program which gives some idea of its capabilities:



Visualization and Analysis Tools (cont.)

- Spreadsheet templates make easy copy and paste of data to aid analysis. These tools are constantly evolving.
 - Density, virial radius evolution and phase space analysis. Includes fit profiles for fitting a model to the density profile
 - Comparison template that allows direct comparisons between different simulation runs.
 - Simulation creation tool which will help the user pick suitable choices for various simulation inputs and then plan for other variants of that simulation to augment a systematic study of similar scenarios.