Exploring the dynamics of galaxy-like systems through Verlet integration.

MANUEL ALMAGRO RIVAS

Physics degree. Universidad de Granada.

M. Almagro: malmriv@correo.ugr.es

Compiled June 20, 2020

The velocity Verlet algorithm is applied to a 2D galactic model based on the Milky Way. A simulation developed in Fortran is carried out and the evolution of certain magnitudes (energy conservation, radial mass distribution, etc.) are monitored. The physical fitness of this algorithm is verified. We discuss certain weaknesses regarding the rotation of the galaxy, and suggest improvements that can be made towards a more physically accurate representation of a galaxy.

1. INTRODUCTION

N-body simulations continue to be an open problem in computational physics. There are many algorithms available, but each one of them is based on a different set of assumptions and physical models of the forces at hand. In this work, I will explore how to apply the velocity Verlet algorithm (a Newtonian approximation to the N-body problem) to the simulation of a galaxy-like system. The mathematical foundation of the algorithm will be discussed, as well as certain specific details about the implementation of the simulation. Certain magnitudes of the system (such as energy and momentum conservation, its rotational curve, mass distribution, number of collisions, central black hole mass, etc.) will be monitored and discussed. Finally, the results will be discussed, arriving to a solid evaluation of the model's adequacy. The most generic form of the code elaborated for this research will be made publicly available.

2. THE VELOCITY VERLET ALGORITHM.

A. Motivation for an adequate integrator.

Newtonian gravitational interaction is defined by the well-known differential equation:

$$m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} = -G m_i \sum_{j \neq i} \frac{m_j \left(\boldsymbol{r}_i - \boldsymbol{r}_j \right)}{\left| \boldsymbol{r}_i - \boldsymbol{r}_j \right|^3} \tag{1}$$

Where $\boldsymbol{r_i}(t)$ is the position vector of the i-th body, m_i is its mass and G is the Newtonian gravitational constant. For a system of N bodies, there will be N equations which can be solved numerically. Therefore, a good method is required in order to obtain the trajectories of the bodies in the system. There exist different algorithms that allow this, but some of them suffer from unavoidable deficiencies. In one extreme of the simplicity spectrum lies Euler integration, which suffers from a notorious energy-accumulation problem (especially in oscillatory systems). In the opposite extreme we find the well-known fourth-order Runge-Kutta method, which describes well the evolution of the system, but requires four evaluations of the force per step. Fortunately, there are methods which lie in some sort of middle ground that satisfy our needs of computational simplicity and physical fitness. Namely, the leapfrog algorithm(s) and Verlet integration. We will make us of the latter. Verlet integration was developed in the 1960-70 decade by Loup Verlet in a series of articles concerning molecular dynamics. Although it underwent different stages regarding complexity, including naive approximations (see Verlet, 1967)[1], some more sophisticated variants fulfill our requirements sufficiently well.

B. Mathematical derivation of the velocity Verlet algorithm.

If a system evolves in time steps of duration h, positions and velocities after a step can be obtained by computing the Taylor expansion of both $r(t+h)$ and $v(t+h)$ up to 2^{nd} order. Since velocity is also explicitly computed (instead of simply applying the definition of derivative to the known positions) this type of Verlet integration is known as velocity Verlet integration. This expansion results in:

$$\boldsymbol{r}(t+h) = \boldsymbol{r}(t) + h\frac{d\boldsymbol{r}(t)}{dt} + \frac{h^2}{2}\frac{d^2\boldsymbol{r}(t)}{dt^2} + \mathcal{O}\left(h^3\right) \tag{2}$$

$$\boldsymbol{v}(t+h) = \boldsymbol{v}(t) + h\frac{d\boldsymbol{v}(t)}{dt} + \frac{h^2}{2}\frac{d^2\boldsymbol{v}(t)}{dt^2} + \mathcal{O}\left(h^3\right) \tag{3}$$

Identifying the velocity and acceleration in Eq. 2 and 3:

$$\boldsymbol{r}(t+h) \approx \boldsymbol{r}(t) + h\boldsymbol{v}(t) + \frac{h^2}{2}\boldsymbol{a}(t) \tag{4}$$

$$\boldsymbol{v}(t+h) \approx \boldsymbol{v}(t) + \frac{h}{2}[\boldsymbol{a}(t) + \boldsymbol{a}(t+h)] \quad \textbf{(5)}$$

These equations can be used to compute the position and velocities for every step (the force is given by Eq. 1), and both the total energy of the system and the computational cost of the resulting simulation fit our needs. It will be shown that both energy and momentum are conserved even if elastic collisions are implemented. This is remarkable, given the sudden change in velocity implied by frontal collisions. Finally, although this algorithm is not fit for following individual particles' trajectories, it is good enough if all we want is to study the dynamics of a system as a whole.

C. Implementation.

The Fortran code used in the present work is referenced in Section 6. The algorithm can be optimally implemented like in the following pseudocode:

Algorithm 1. Velocity Verlet.

1: **procedure** VERLET
2: $N \leftarrow 1000$ ▷ set no. of steps
3: maxtime $\leftarrow 4$ ▷ set simulation time
4: $h \leftarrow$ maxtime$/N$
5: $t \leftarrow 0$
6: **read** $r_i(t), v_i(t)$
7: **evaluate** $a_i(t)$ ▷ Eq. 1
8: **while** (t $\leq$ maxtime) **do**
9: **evaluate** $r_i(t+h)$ ▷ Eq. 4
10: $w_i = v_i(t) + \frac{h}{2}a_i(t+h)$ ▷ auxiliary term
11: **evaluate** $a_i(t+h)$ ▷ w/ positions from line 9
12: $v_i(t+h) = w_i + \frac{h}{2}a_i(t+h)$
13: $t \leftarrow t+h$
14: **save data** ▷ store in files as needed
15: **end procedure**

Where the indices refer to the i-th body. For small systems (a pendulum, a small stellar system, a cluster of a few tens of atoms, etc.) the auxiliary variable w_i can be skipped by storing all of the variables into matrices instead of 1D-arrays. Therefore, all information from past iterations can be accessed directly by applying Eq. 5. If the system is expected to contain many bodies (like in this case), the aforementioned pseudocode results in the best strategy, memory wise.

C.1. Some notes on time complexity.

Evaluating accelerations implies evaluating their interaction with each body in the system, so this will become an $\mathcal{O}(n^2)$ problem. Certain techniques, like implementing a neighbour list (or Verlet list), can make the execution time shorter. However, the problem remains an $\mathcal{O}(n^2)$ problem. Some authors (Zhenhua et al., 2004)[2] have suggested ways of lowering time-complexity down to $\mathcal{O}(n \log n)$. Nevertheless, those techniques lay out of the scope of this work; some of their originality steams from the idea of exploiting features found only in older hardware (especially mechanical drives) that no longer continues to be a part of high-performance equipment. Implementing Zhenhua et al.'s modified version of a Verlet list could potentially be interesting to researchers whose available computing system still uses mechanical drives instead of more modern solid-state drives. A standard Verlet list[3] will be sufficient for our purposes, with the exception that the black hole will appear in every list in the system due to its orchestrating role in the simulation.

C.2. Implementing elastic collisions.

There are some non-trivial aspects about implementing a collision algorithm that I feel are relevant to those who wish to replicate the present work. Physically, there is only one obvious condition for two bodies to bounce off of each other: the distance between their centers of mass being equal or less than the sum of their radii, assuming round (2D) or spherical (3D) bodies. Mathematically:

$$|\boldsymbol{r}_i - \boldsymbol{r}_j| \leq R_i + R_j \quad \textbf{(6)}$$

This seems sufficient, but the results are not satisfactory. The bodies make contact with each other, but it happens very frequently for certain velocity regimes that the inter-body distance does not change rapidly enough after a collision. If by the next iteration two bodies that have already collided are not sufficiently separated, the only imposed condition so far could trigger the program into computing their new velocities, thus redirecting the bodies against each other. This results in a "sticky behaviour", for lack of a better name. Its manifestation is clear: bodies start to gather into undesired clusters with an occasional burst due to particles that do collide "successfully".

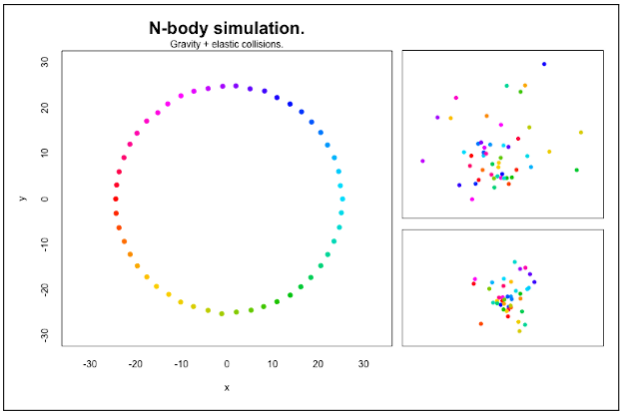

Fig. 1. Cluster formation due to a badly designed collision algorithm. **Left**: initial configuration (almost completely symmetric). **Top-right**: correct collision algorithm after 4.0 s. **Bottom-right**: incomplete collision algorithm with the aforementioned "sticky behaviour" after 4.0 s.

A possible solution (and its geometric intuition) is given here. Suppose two bodies, i and j. If i is getting closer to j, then $\boldsymbol{v}_i - \boldsymbol{v}_j$ must *roughly* be pointing towards j. (Quantitatively, this means that before a collision has taken place the absolute angle between $\boldsymbol{v}_i - \boldsymbol{v}_j$ and $\boldsymbol{r}_i - \boldsymbol{r}_j$ must be lower than 90^o). Once they collide, $\boldsymbol{r}'_i - \boldsymbol{r}'_j$ remains relatively unchanged, but $\boldsymbol{v}'_i - \boldsymbol{v}'_j$ must have changed in such a way that the new absolute angle between $r'_i - r'_j$ and $v'_i - v'_j$ is now greater than 90^o. This fuzzy intuition can be rendered mathematically like this:

$$\textbf{angle}\left(\boldsymbol{v}'_i - \boldsymbol{v}'_j, \boldsymbol{r}'_i - \boldsymbol{r}'_j\right) \geq 90^o \Rightarrow \langle \boldsymbol{v}'_i - \boldsymbol{v}'_j, \boldsymbol{r}'_i - \boldsymbol{r}'_j \rangle \leq 0 \quad \textbf{(7)}$$

(Where $\langle , \rangle$ refers to vector dot product). There does not seem to exist a simpler requirement that results in energy-conserving collisions. Merely considering changes in the angle between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ makes the algorithm fail either for frontal or side-to-side

collisions. Considering relative positions of each pair of particles seems necessary to set a strong condition that allows for any type of collision at all.

3. DEFINING OUR SYSTEM.

Our galaxy-like system, which is loosely based on the Milky Way, will include a central black hole around which a number of smaller bodies (which can be seen as solar systems) will rotate. These bodies can elastically collide with each other, and they will be absorbed by the central black hole. When this happens, a new body fitting the average parameters of the rest of the bodies (mass, radius and velocity) will be generated.

A. Normalisation.

Simulating a system of the scale of a galaxy means that we will need to work with values differing by tens of orders of magnitude, even if we use adequate units (like kiloparsecs and solar masses). Therefore, it is best to first define a system to normalise the numbers we will be working with. The resulting magnitudes will be expressed in terms of *normalised distance units* (ndu), *normalised time units* (ntu) and *normalised mass units* (nmu). An immediate choice for distance and mass could be:

$$\boldsymbol{r}' = \frac{\boldsymbol{r}}{c} \tag{8}$$

$$m' = \frac{m}{M_{cbh}} \tag{9}$$

Where c is the distance from the center of the galaxy to some vantage point, like our own Solar System, and M_{cbh} is the mass of the central black hole of our galaxy. Therefore, time is fixed:

$$t' = \sqrt{\frac{GM_{cbh}}{c^3}}t \tag{10}$$

The value of c has been estimated in previous works. In this study, we will use the value provided by Abuter et al. (2019)[5] of $c \equiv R_{SgrA*-Sun} = (8.178 \pm 0.022)$ kpc. The same paper offers an accurate estimation of the mass of the central black hole in our galaxy, Sagittarius A* (or Sgr A*) of $M_{cbh} = (4.154 \pm 0.014) \cdot 10^6 M_\odot$. Furthermore, we will also need to gather data regarding the radius of the Milky Way (R). In this work we will use the value given by Li (2016)[4] of $R = 25$ kpc, which appears to be the standard cutoff radius used in numerous works.

As for the bodies orbiting our simulated black hole, it will suffice to assign each body a mass (m) and effective radius (r_{eff}), which will be used for collision computations. (Even though these bodies are meant to represent solar systems, we will assume that the mass in a solar system is mainly due to its star, disregarding planetary contributions). Online catalogs, like the TAPVizieR catalog[6] from the University of Strasbourg, provide a good source of raw data. The data collected by Grieves et al. (2018)[7] comprise a sufficiently diverse mix of stars for the purposes of this study. A sample of $N = 2343$ stars (those with complete, well-formatted data) returns an average mass of $m = (1.09 \pm 0.13)\ M_\odot$ and an average radius of $R = (1.3 \pm 0.4)\ R_\odot$. Nevertheless, it soon became obvious that this radius (in the order of magnitude of 10^{-20} ndu) was too small for any collision to take place. Therefore, we performed a simulation where the galaxy was left to evolve for a period of time equivalent to the current age of the Sun, with the effective radius parameter varying each time. The results appear in Table 1, and the representation of the data contained there can be seen in Figure 2. Since stellar collision seems to be a rare event, setting $r_{eff} = 1 \cdot 10^{-3}$ (corresponding to the lower number of collisions in Table 1) seems like a sensible choice, but this is entirely arbitrary. The normalisation values used in the simulation are summarised in Table 2, and the rest of the parameters that describe the system are contained in Table 3.

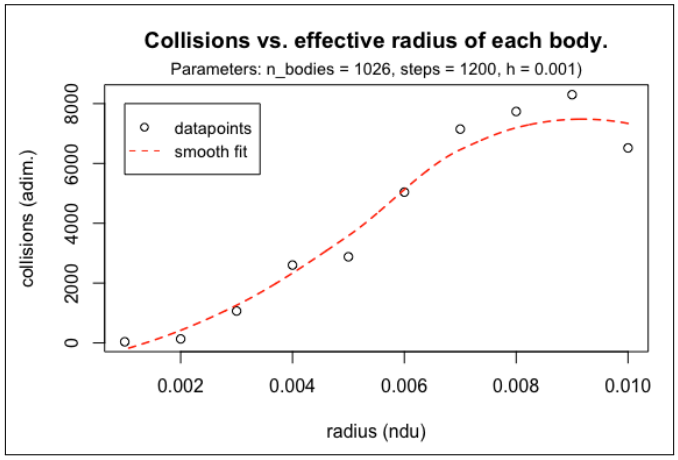

Fig. 2. Number of collisions vs. r_{eff}. The Gaussian-like shape is due to individual bodies achieving escape velocity. A high number of relatively big bodies contained in a small area results in some bodies being ejected far away from the galactic center, thus no longer participating in the simulation.

Table 1. Results of the simulation (effective radii in normalised distance units).

$r_{eff}(ndu)$	collisions
0.001	37
0.002	133
0.003	1066
0.004	2602
0.005	2879
0.006	5041
0.007	7148
0.008	7737
0.009	8298
0.01	6518

Table 2. Normalisation values.

M_{cbh}	c
$(4.154 \pm 0.014) \cdot 10^6 M_\odot$	(8.178 ± 0.022) kpc

The resulting escalated units properly allow us to simulate a system of galactic proportions. A normalised time unit (1 ntu) is approximately equal to $1.707 \cdot 10^{17}$ s, our roughly 1.2 times the current age of the sun. In this span of time, gravitational interaction (which propagates at the same speed as electromagnetic fields) has more than enough time to cover the whole

Table 3. Galaxy parameters.

	R	m	r_{eff}
Raw	25 kpc	$(1.09 \pm 0.13)\ M_\odot$	-
Normalised	3.057 ndu	$(2.6 \pm 0.3) \cdot 10^{-7}$ nmu	$1 \cdot 10^{-3}$ ndu

galaxy from side to side. Therefore, our Verlet list will need to be defined by an arbitrary radius. (In this work, we will choose $r_{Verlet} = 0.4$ ndu, which after some test runs was shown to return physically accurate results with a considerable decrease in run time).

B. Remaining parameters.

Only three more parameters need to be fixed, as shown in Algorithm 1. These are the number of bodies (n_{bodies}), the total number of steps (N) and the total time to simulate (t_{max}). The value of h needs to be sufficiently small for the Verlet approximation to remain valid, but the timespan to be simulated needs to be long enough that we can study the evolution of the system over time. We selected $N = 10^4$ with $t_{max} = 20.0$ ntu, which results in a time step of $h = 2 \cdot 10^{-3}$ ntu. The total number of bodies is only constrained by the total amount of RAM available. Considering that we will be mainly interested in the dynamics of the system, we will need a bare minimum of seven arrays: two for position, two for velocity, two for acceleration and one for mass. (In practice, we will also account for simulated time, energy conservation, mass absorption by the central black hole, mass radial distribution, and some other magnitudes). These vectors will eventually be stored in data files, each one of them containing $N \times n_{bodies}$ entries in double precision (8 bytes/entry). Keeping in mind that it will eventually be necessary to load these files onto the RAM of a computer in order to perform numerical analysis, it is possible to achieve a rough estimation of how many bodies a certain computer can handle. Calling the total memory we want to allocate for the simulation M, we have that:

$$n_{bodies} = \frac{M}{7N \cdot 8\ \ bytes} \tag{11}$$

The computer used in this simulation has a total of 4 GB of RAM memory available. Given that the operating system and the relevant software will be using some of that memory, we can safely allocate 1.5 GB for the numerical processing of the data generated by the simulation. According to Equation 11, this results in $n_{bodies} \approx 2800$ bodies. Since this is the theoretical maximum number of bodies this computer could realistically handle, we will select a lower value of $n_{bodies} = 1500$. The parameters of the simulation are summarised in Table 4.

Table 4. Simulation parameters.

N	n_{bodies}	t_{max}	h	r_{Verlet}
10^4	1500	20 ntu	$2{\cdot}10^{-3} ntu$	0.4 ndu

C. Generating an initial state.

We will use the values and parameters specified in Tables 2, 3 and 4 to generate a data file containing the initial state of the system, and then let the simulation run. This file will define each individual body, specifying spatial coordinates, velocity, mass and effective radius. The process of generating a list of bodies with an assigned mass, position, velocity and effective radius can be best performed in a statistical analysis environment like R. The link to the code is available en Section 6. A list of 1500 bodies was generated with their masses being sampled from a Gaussian distribution, $m \sim N(1.09, 0.13)$ nmu. The effective radius r_{eff}, which will be used in estimating collisions, is also sampled from a Gaussian distribution $N(10^{-3}, \frac{10^{-3}}{3})$. Their positions were uniformly distributed through the circle of radius $R = 3.057$ ndu. The most immediate approach of just generating random radii and angles does not work, even if these magnitudes are uniformly distributed in adequate domains. This happens because the arch separating two bodies is proportional to their radii, so most masses accumulate towards the center of the galaxy. Therefore, our approach was following the same technique used in some Monte Carlo simulations. Generating $\frac{4N}{\pi}$ bodies in the square of side $2R$ ensures that approximately N bodies will lay inside the inscribed circle of radius R. Therefore, we do not end with the precise number of bodies we intended to generate, but it is possible to simply trim the list of coordinates down to n_{bodies} when it is longer than we need. A true uniform distribution of bodies inside a circle is therefore obtained. Velocities were assigned in such a way that each body would roughly move in a circular orbit around the central black hole (Newtonian approximation). In practice, the trajectories become chaotic very soon due to gravitational pull from nearby masses, but this approach ensures that we will get some kind of circular motion instead of random motion.

D. Objectives.

The magnitudes we will monitor and study are:

1. Energy conservation.
2. Angular momentum conservation.
3. Moment of inertia.
4. Central black hole mass gain.
5. Radial distribution of mass.
6. Inter-particle collisions.
7. Rotation curve.

Our task will be to assess if the simulation results match our expectations, as well as explaining (if necessary) the possible deviation of our model's behaviour from that of a real galaxy.

4. RESULTS AND DISCUSSION.

The simulation took 6.5 hours of wall-clock time in a domestic computer, equivalent to 13 hours of CPU time in a processor with two cores operating simultaneously. The resulting files could be loaded by the computer with no memory overflow happening, as we expected from our previous estimation in Eq. 11. The simulation was animated afterwards and made available online (see Section 6). Watching the animation is advised, since it provides a valuable resource for understanding the results obtained from here on.

A. General description.

The galaxy undergoes an evolution with several distinct stages that will leave an imprint in most of the studied magnitudes. From $t = 0$ to $t \approx 2$ ntu, the galaxy suffers an expansion, probably because the velocities assigned to each body were greater than the corresponding orbital velocities. Then, from $t \approx 2$ to $t \approx 5$, the stars located in the more external parts of the galaxy start precipitating towards the central black hole, resulting in an overall compression of the galaxy. Most of the absorptions into the black hole take place at this moment. At $t \approx 6$, the stars that did not get absorbed by the black hole suffer a gravitational pull that allows them to get away from the center of the galaxy, and the whole system starts losing its initial configuration. Some stars are expelled in this process, while others just remain in orbit. By this point, the galaxy has reached a somewhat stable state. In the following subsections it will become clear that detailed numerical analysis can unveil some interesting subtleties.

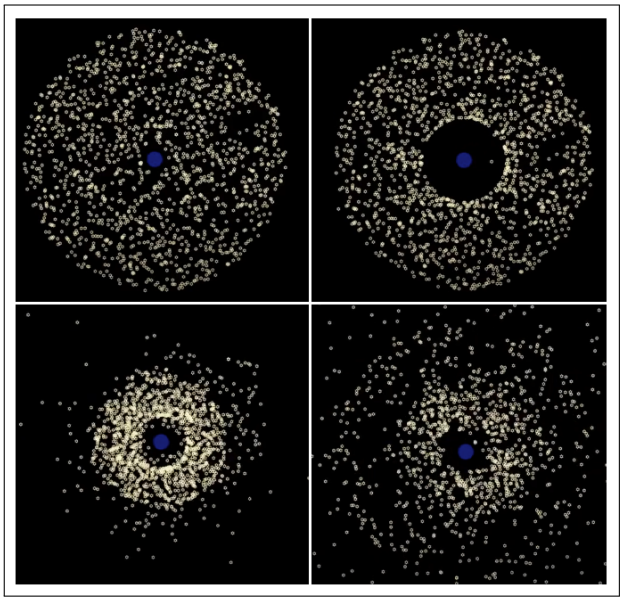

Fig. 3. Evolution of the galaxy. **Top left:** initial configuration. **Top right:** expansion. **Bottom left:** contraction. **Bottom right:** stability.

B. Energy conservation.

Energy is largely preserved in the simulation. As expected, Verlet integration performs well when it comes to preserving the energy of a system, even with add-ons like collision computation (where velocities change very rapidly) and new body generation (which artificially introduces new bodies). In Figure 4 we can see, in blue, the period of expansion mentioned earlier. Kinetic energy soars, while potential energy drops due to most stars getting further away from the main source of gravitational potential (the black hole). At $t \approx 5$ ntu, the period of contraction begins and most stars gather in the vicinity of the black hole. Most absorptions occur in this span of time, which has a noticeable effect in the total energy of the system. Since every swallowed star is replaced by a new one, the total mass of the system increases very rapidly. This results in an apparent violation of energy conservation, but can be explained by claiming that this galaxy is an open system: new matter is incorporated in the form of new stars. Even with mass being added, the total energy of the system expressed as a linear function of time results in $E(t) = (2.768 \pm 0.008) \cdot 10^{-6} t$, with $R^2 = 0.9566$. The galaxy enters a stability period with a less ordered configuration. Since most collisions have taken place already, the rate at which the mass of the galaxy changes decreases, and the total energy remains largely unaltered. There is an upwards trend, of course, due to a star being occasionally swallowed by the black hole.

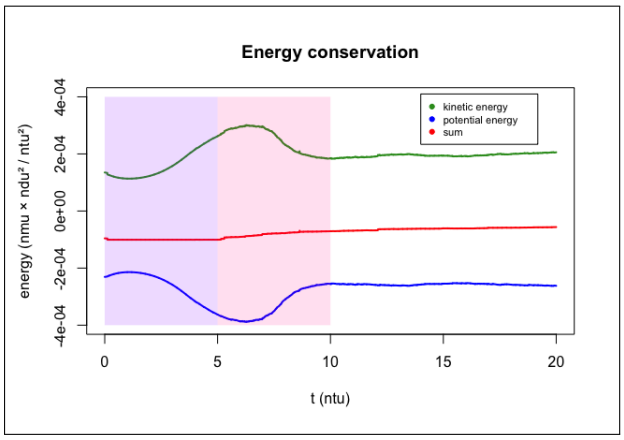

Fig. 4. Energy conservation. The approximate period of expansion is highlighted in blue, while the period of contraction is highlighted in light red.

C. Angular momentum conservation.

Angular momentum is well-preserved, although the influence of star-replacement is more noticeable.

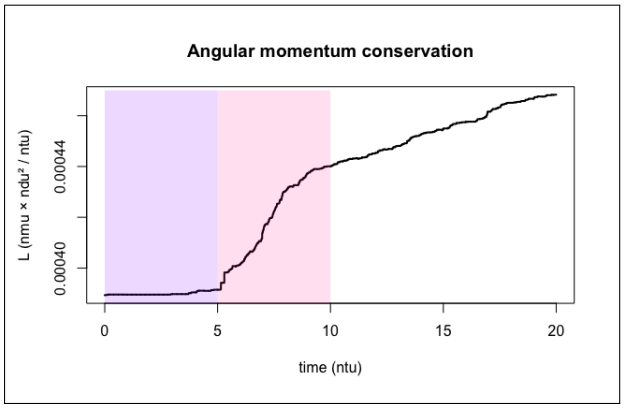

Fig. 5. Angular momentum conservation. The approximate period of expansion is highlighted in blue, while the period of contraction is highlighted in light red.

Angular momentum is more affected by star replacement than energy seems to be. It can be seen in Figure 5 that **L** is almost constant during the expansion phase. Nevertheless, as soon as stars are incorporated to the black hole (and new stars are generated) angular momentum increases rapidly. After the contraction phase has ended, angular momentum becomes more stable and, like energy, increases at a slower rate.

D. Moment of inertia.

The central black hole did not partake in the computation of the moment of inertia. It is always located at $\vec{r} = 0$, and therefore

its contribution is always null. The contraction period does not seem to affect inertia noticeably. This could seem like a wrong result, but one ought to keep in mind that a galaxy is not a system where each particle is strongly linked to its neighbours. For a solid body, we would expect inertia to drop if the overall radius is made smaller, but gravitational attraction does not satisfy this requirement. Each body is only linked to its closest neighbours, both because of the Verlet list implementation, and also *de facto* due to gravity being an $1/r^2$ central force. It can be seen that the moment of inertia increases as stars disperse far from the galactic center, which is a foreseeable result.

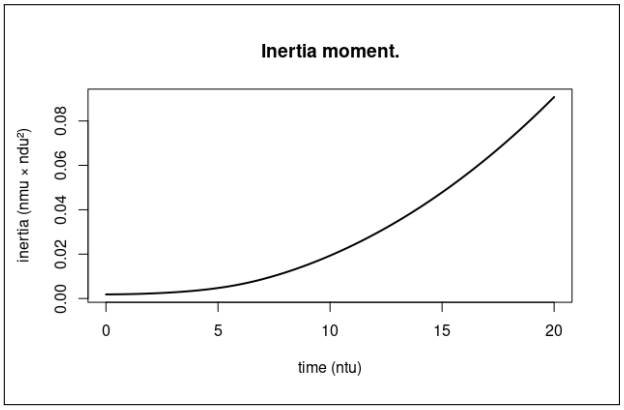

Fig. 6. Evolution of the moment of inertia.

E. Black hole mass gain.

In total, 420 stars were swallowed by the black hole. Star absorption is a rare event in the expansion period, where most stars near the black hole are moving further from it. Most of the stars were swallowed during the subsequent contraction period. Afterwards, during the stability phase, black hole absorption becomes a rarer event (see Fig. 8). It can be seen that there is a direct correlation between the mass contained in the black hole and both angular momentum conservation (see Fig. 5) and, less noticeably, total energy (see Fig. 4).

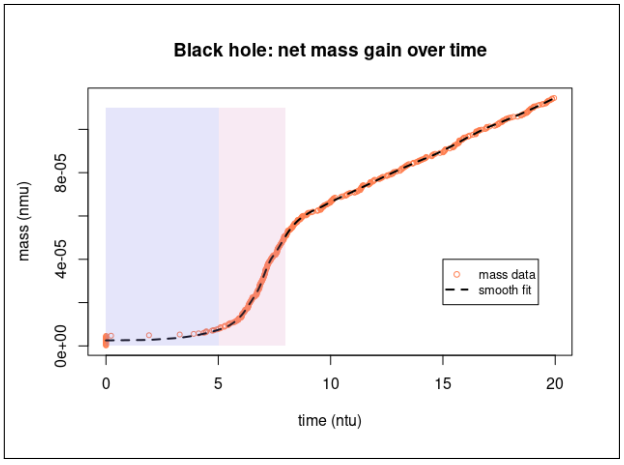

Fig. 7. Evolution of the black hole's mass.

Computing the discrete derivative of dM_{cbh} with respect to time, dM_{cbh}/dt, returns a somewhat noisy set of points, but the decreasing tendency is evident when the results are plotted:

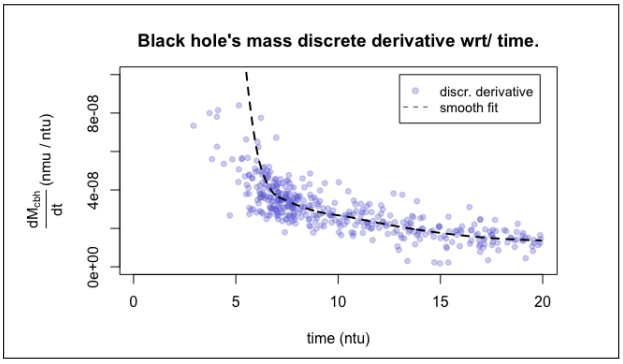

Fig. 8. dM_{cbh}/dt vs. time

The graph shows the rate at which the black hole's mass changes as a function of time. There are two layers of information in Figure 8: both mass gain rate and absorption frequency are shown.

1. The decreasing tendency is evident to the eye. This means that the black hole starts gaining mass very quickly during the contraction phase. After that, the influx of mass becomes more stable.

2. Since star absorptions are not uniformly distributed events, we can infer from the point density in the graph that most absorptions happen during the contraction phase mentioned earlier. Before $t \approx 5$ ntu, barely any stars fall into the black hole.

F. Radial distribution of mass.

Radial distribution of mass can be adequately visualised via a *Mass vs. distance* graph for a single iteration of the simulation. If one wants to visualise how this magnitude evolves over time, it becomes necessary to use a color-coded graph. (See Section 6 for the relevant code). For each iteration, a script subdivides the system in 40 concentric rings centered around the black hole. The total mass of the bodies located in each ring is computed. Each ring is mapped to an entry in a 40×1 column matrix, and each column matrix corresponds to a different iteration. These column matrices are appended together into a 40×10^4 matrix, and each entry is assigned a color whose luminosity is proportional to its value. Therefore, lighter values imply higher density of mass. The resulting graph can be seen in Fig. 9.

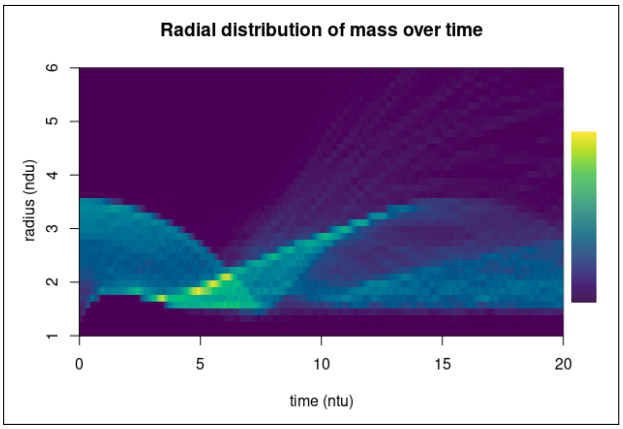

Fig. 9. Radial distribution of mass.

Both the expansion and the contraction period are clearly visible in this graph, as well as the stable phase where some of the stars have strayed away from the galaxy. Figure 9 provides an additional layer of information that could not be inferred from previous graphs. We can clearly see that a secondary, less noticeable expansion period happens at around $t \approx 10$ ntu. This indicates that our choice for t_{max} as well as some other parameters found in Table 4 should be chosen differently if a more detailed study of the stable phase was to take place.

G. Inter-particle collisions.

A total of 5317 star-to-star collisions took place throughout the simulation. Figure 10 shows the evolution of inter-collision time. Collisions took place regularly (every $\sim 2 \cdot 10^{-3}$ ntu) at first, but this regularity was disrupted at the beginning of the contraction period due to a higher radial density of mass (see Figures 10 and 9).

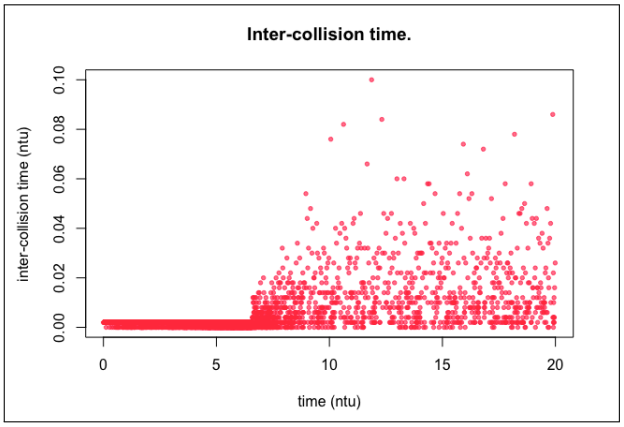

Fig. 10. Inter-collision time as the galaxy evolves.

H. Rotation curve.

A rotation curve is a graph showing orbital speeds of observed bodies vs. radial distance from their respective galactic center. The discrepancy between observed rotation curves and predicted curves based on the amount of matter observable is one of the main bits of evidence pointing towards the existence of dark matter.

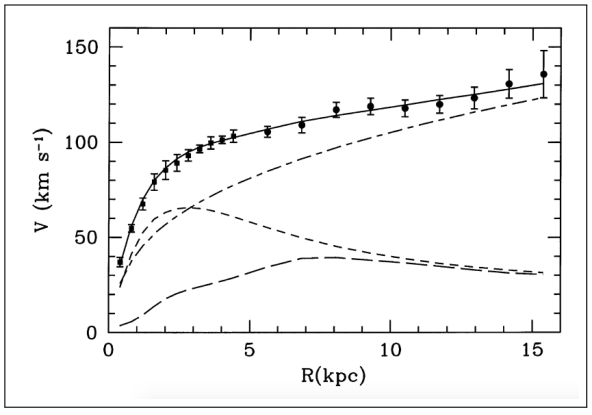

Fig. 11. Rotation curve of M33 and predicted curves for observed mass. **Source**: *The extended rotation curve and the dark matter halo of M33*, by Corbelli and Salucci (2000)[8].

There is a wealth of studies that show the profile of these curves for real galaxies, as well as the best-fit prediction with the available data. See Figure 11 for an example by Corbelli and Salucci (2000)[8]. In Fig. 11 one can see that the rotation curve of a real galaxy shows, at least, a constant velocity after a certain orbital distance. Nevertheless, the predictions by Corbelli et al. (2000) show a peak around the central halo, and then a decrease. Dark matter would, ideally, explain this difference completely. There are certain mass distribution profiles (like the Navarro-Frenk-White profile[9]) that explain observed rotation curves and also require the existence of a different type of matter. Our

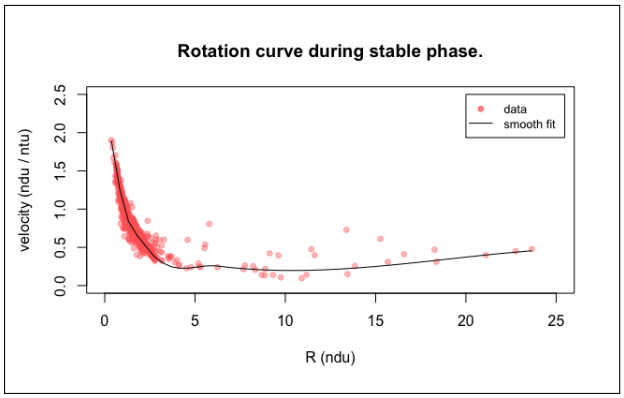

Fig. 12. Rotation curve for our simulated galaxy during the stable phase.

galaxy shows (see Fig 12) a curve corresponding to a galaxy with no dark matter. This result is not surprising. There was no consideration whatsoever regarding different types of matter, nor we ensured that an adequate mass distribution profile was satisfied in our simulation. Our result is not compatible with most observations of real galaxies. *Differential velocity*, as it is called, would cause the centre of a galaxy to rotate much faster than the external stars, thus not allowing for the formation of large-scale structures. Some models, like the Density Wave Theory suggested by Lin and Shu (1964)[10] allow the existence of dark matter as well as the existence of spiral structures. This is not a sign of failure in our simulation: it was never our purpose to account for the existence of exotic types of matter. In fact, we obtained the expected velocity profile for a galaxy where only ordinary matter is to be found. A large halo would need to be introduced and stabilised by means of constraining the motion of its components so that we could obtain a flat or up-going rotation curve.

5. CONCLUSIONS.

1. The numerical analysis shed light over some features that were not expected. The radial distribution of mass (Figure 9) reveals that the galaxy evolves in a way reminiscent of an underdamped oscillator, reaching equilibrium after a few expansion-contraction cycles. After the first cycle, a secondary expansion can be observed. Therefore, we can conclude that our initial choice of parameters (Table 4), specifically the combination of t_{max} and N, was not sufficient if our purpose was to study a truly stable galaxy. A longer simulation should be prepared so that the system can evolve towards a stable configuration. The expected succession of damped expansion-contraction cycles would

give way to a truly stable configuration where most stars do follow stable orbits.

2. As for the Verlet algorithm, our analysis seems to confirm that both energy and angular momentum are preserved even when some additional features are at work.

   (a) Collisions result in a sudden change in velocity, although this does not seem to affect continuity in kinetic energy.

   (b) Star replacement adds mass to the system whenever the central black hole swallows a star. Both angular momentum and total energy seem to be affected by the addition of new mass. Fortunately, the results do not fall far from the expected increase that would take place in an open system with mass flowing in.

   Therefore, our implementation works fairly well, since these magnitudes are well conserved (as can be seen in Figures 4 and 5). We consider that this simulation succeeded in preserving those magnitudes that physical laws require to remain constant, such as total energy and angular momentum.

3. A more detailed study should be done regarding the minimum extent of the Verlet list's radius, and its relationship with other parameters like mass density and galaxy size. This parameter is key in improving the overall performance of the simulation, and a lower value could improve the speed at which similar simulations are carried out.

4. Incorporating a mass density profile that accounts for dark matter could potentially be interesting for future research.

6. ONLINE RESOURCES.

The code and scripts written for this simulation are available in the author's GitHub page. Technical details about how these files are to be used together can be found in the Read Me page in the previous link. The animation derived from this work can also be viewed online. Any inquiries can be sent to the author's email address, which can be found in the first page.

7. ACKNOWLEDGEMENTS.

I would like to thank the University of Granada for allowing me to use the PROTEUS Computing Service from the "Carlos I" Institute of Theoretical and Computational Physics. I would also like to thank my mother, who supplied a constant flow of caffeinated beverages during the long hours it took to debug the Fortran code, as well as Coqui, our dog, whose company I greatly appreciated.

8. REFERENCES.

1. Verlet, L. (1967). *Computer "Experiments" on Classical Fluids: Thermodynamical Properties of Lennard-Jones Molecules.* **Physical Review**, 159(1), 98-103. doi: 10.1103/physrev.159.98

2. Yao, Zhenhua Wang, Jian-Sheng Liu, G.R. Cheng, Min. (2004). *Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method.* **Computer Physics Communications**. 161. 27-35. doi: 10.1016/j.cpc.2004.04.004

3. Rapaport, D. (2004). *The Art of Molecular Dynamics Simulation* (2nd ed., pp. 54-58). Cambridge, UK: C**ambridge University Press.**

4. Li, E. (2016). *Modelling mass distribution of the Milky Way galaxy using Gaia billion-star map*. Retrieved 14 March 2020, from **arXiv.org**. Link: https://arxiv.org/pdf/1612.07781

5. Abuter, R., Amorim, A., Bauböck, M., Berger, J., Bonnet, H., Brandner, W. et al. (2019). *A geometric distance measurement to the Galactic center black hole with 0.3% uncertainty*. **Astronomy Astrophysics**, 625, L10. doi: 10.1051/0004-6361/201935656

6. TAPVizieR Catalog from Université de Strasbourg: https://vizier.u-strasbg.fr/viz-bin/VizieR

7. Grieves, N., Ge, J., Thomas, N., Willis, K., Ma, B., Lorenzo-Oliveira, D., Queiroz, A., Ghezzi, L., Chiappini, C., Anders, F., Dutra-Ferreira, L., Porto de Mello, G., Santiago, B., da Costa, L., Ogando, R., del Peloso, E., Tan, J., Schneider, D., Pepper, J., Stassun, K., Zhao, B., Bizyaev, D. and Pan, K., 2018. *Chemo-kinematics of the Milky Way from the SDSS-III MARVELS survey.* **Monthly Notices of the Royal Astronomical Society**, 481(3), pp.3244-3265. doi: 10.1093/mnras/sty2431

8. Corbelli, E., Salucci, P. (2000). *The extended rotation curve and the dark matter halo of M33.* **Monthly Notices Of The Royal Astronomical Society**, 311(2), 441-447. doi: 10.1046/j.1365-8711.2000.03075.x

9. Navarro, J., Frenk, C., White, S. (1996). *The Structure of Cold Dark Matter Halos*. **The Astrophysical Journal**, 462, 563. doi: 10.1086/177173

10. Lin, C., Shu, F. (1964). *On the Spiral Structure of Disk Galaxies*. **The Astrophysical Journal**, 140, 646. doi: 10.1086/147955