MNRAS 000, 1–5 (2022) Preprint 29 May 2022 Compiled using MNRAS LATEX style file v3.0

The N body problem - with applications to Trojan stability

Jordan W. N. Moncrieff,[1]★
[1]Department of Physics, University of Western Australia, 35 Stirling Hwy, Crawley WA 6009

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

In this paper we discuss the N body problem, and provide an overview of the methods used to numerically solve an N body system for a given initial configuration. We will present a Python implementation of a numerical N body solver, and provide a few common astrophysical applications of N body solvers in order to demonstrate the validity of our solver. Finally, we apply our solver to the study of Trojan asteroids. We will validate (through simulation) analytical results which show that the L4 and L5 Lagrange points become unstable past a critical mass ratio of the massive body to the orbiting body.

Key words: software: simulations – celestial mechanics

1 INTRODUCTION

1.1 N-body problem statement

The N-body problem asks, given N particles of known masses, initial positions, and velocities - all mutually attracting one another with a force given by Newton's law of universal gravitation - how does the system evolve over time? This problem, formally proposed by Isaac Newton in 1687, has a closed-form solution only for the case of $N = 2$; the solutions take the form of the familiar conic sections observed in planetary orbits. However, in the general $N \geq 3$ case, approximation schemes and numerical simulations must be used. In this paper, we will be discussing special classes of solutions to the N body problem, the techniques used to solve them, and their applications to astrophysics.

1.2 N-body problem history

The N body problem as we know it today was first posed by Isaac Newton in his famous 1687 book Prinicpia Mathematica. Newton had solved exactly the two body problem, which allowed him to predict the orbits of the planets around the Sun. However, he could not accurately predict the orbit of the moon about the Earth. He realised that this was due to the fact that this was a 3 body system, i.e the interaction between the moon and the sun had to be taken into account. Newton was unable to solve the problem, and for centuries no one could find a solution without making approximations. Progress was made by mathematicians such as Euler and Lagrange, who published a set of special solutions to the 3-body problem in 1767 and 1772 respectively.

By 1885 King Oscar the II of Sweden and Norway offered a prize for anyone who could make substantial progress towards a solution to the problem. Henri Poincaré won this competition by showing that there is no analytic closed form solution for the N body problem for $N \geq 3$ Bruns (1887), i.e the solution to the N body problem cannot be written in terms of a finite number of standard mathematical operations. He also showed that almost all solutions show a high sensitivity to initial conditions - foreshadowing the field of Chaos theory that emerged decades later.

The insolubility of the General N body problem lead to approximation approaches being employed to tackle the problem. Notably, the field of perturbation theory was developed to study the three body problem. This was later used by Urbain Le Verrier in 1848 to locate the previously hypothesised planet Neptune, based on irregularities in the orbit of Uranus.

These days N body problem is much more tractable, with the advent of modern computation we can simulate N body problems for vast structures such as galaxies. Indeed, the N-body problem is still an active field of research, with several new classes of solutions being found for the N body problem in recent years (e.g. Li et al. 2017; Šuvakov & Dmitrašinović 2013; Oks 2015).

1.3 Periodic solutions to the N-body problem

The first exact solutions to the three body problem were found by Euler, and are therefore known as the Euler solutions. They describe three particles of arbitrary mass that are co-linear throughout their entire lifetimes, i.e they all lie within a line in 3D space, and orbit their common centre of mass at identical angular frequencies ω as in figure 1. The relative separations between the three masses is given by the parameter $z \equiv R_{23}/R_{21}$, which can be determined uniquely as the positive real root of a quintic equation in the variable z, with coefficients given by combinations of the masses M_1, M_2, and M_3, as in Yamada & Asada (2010).

The second class of solutions of the three body problem was found by Lagrange. In this solution, three masses orbit the same plane in ellipses, but this time the three points form the vertices of an equilateral triangle at all points in time, as in figure 2.

Over two centuries after Lagrange and Euler, solutions for the N body problem continue to be found. For example Moore (1993) found a stable solution to the three body orbits, where the bodies follow a figure-8 path as in figure 6.

★ E-mail: 22968254@student.uwa.edu.au

© 2022 The Authors

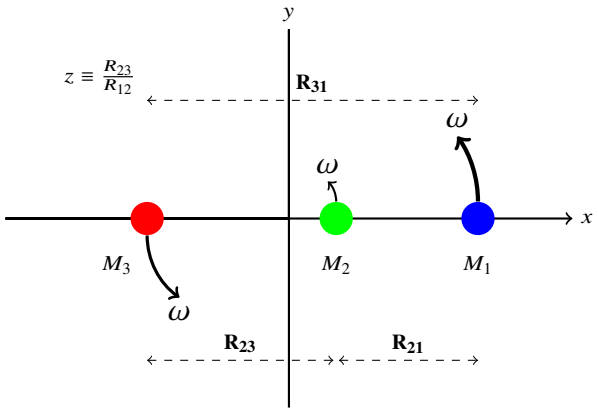


Figure 1. Euler's solution

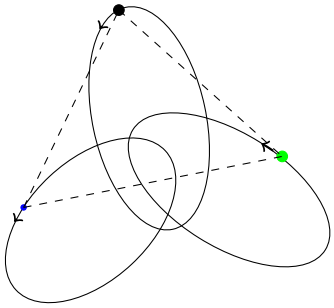


Figure 2. Illustration of Lagranges' solutions

1.4 Lagrange points

When Lagrange was studying the three body problem, he considered the case of the restricted three body problem, where one mass is much smaller relative to the other two. This models situations such as the moon, Sun, Earth system. In the restricted three body problem, there exists five equilibrium points, known as Lagrange points, depicted in figure 4. Two of these Lagrange points, L4 and L5, are stable; the other three Lagrange points are unstable Greenspan (2014). If a mass in stable equilibrium is perturbed, the system will exert a force pushing it back into its equilibrium point - stabilizing the mass. In the L4 and L5 Lagrange points, this restoring force is the Coriolis force. In contrast, any small perturbation to a point in unstable equilibrium will be amplified exponentially, ejecting the mass from the equilibrium point. Thus, any satellite placed in a L1, L2, or L3 Lagrange point will not stay for longer than perhaps a few months. This is why the James Web space telescope - located at L2 - needs to use its thrusters every few months to maintain its position in L2 Greenhouse (2016), since being exactly at the equilibrium point is impossible. Therefore, long term, the only place in the solar system that objects will remain at for an extended period of time are the L4 and L5 Lagrange points.

1.5 Trojan asteroids

Thanks to the stability of the L4 and L5 points, most planets in our solar system contain at least one "Trojan asteroid" - an asteroid that resides near a planets L4 or L5 Lagrange points. Jupiter in particular has thousands Trojans, which have probably resided in Jupiter's L4/L5 since the planet formed Marzari et al. (2002). These Lagrange points form an equilateral triangle with the Sun and Jupiter, as depicted in figure 3.

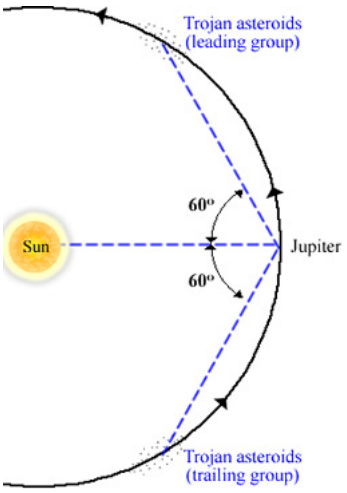


Figure 3. Trojans of Jupiter. Image credit: © Swinburne University of Technology (https://astronomy.swin.edu.au/cosmos/T/Trojan+Asteroids)

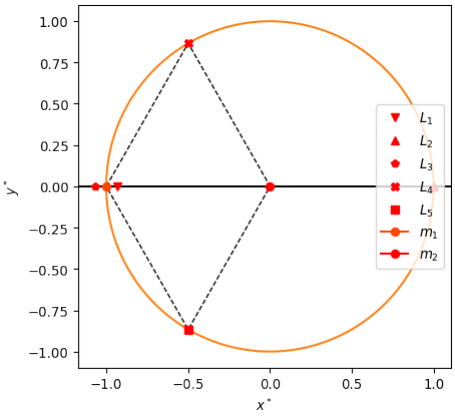


Figure 4. Lagrange points

Within this paper, we will describe a Python implementation of a N-body solver, and use this simulator to investigate the dependence of Trojan stability on the mass ratio of the Star to the orbiting planet.

2 GOVERNING EQUATIONS

Newton's law of Gravitation states that the force of attraction between two particles of mass m_1 and m_2 at a distance r_{12} from one another is given by

$$F = \frac{Gm_1m_2}{r_{12}^2}. \tag{1}$$

So from Newton's second law, $\Sigma F = ma$, the acceleration of a particle, i, due to masses $m_1, m_2, ..., m_N$ at distances $r_{i1}, r_{i2}, ..., r_{iN}$ is given by

$$\mathbf{a_i} = \sum_{j=1, j\neq i}^{N} \frac{Gm_j}{r_{ji}^2}\hat{\mathbf{r}}_{ji} \tag{2}$$

For $i = 1, 2, ..., n$ and $\mathbf{a_i} \equiv d^2\mathbf{r_i}/dt^2$. Thus we have a system of

N second order differential equations for an N particle system. This can be written as a system of $2N$ first order differential equations by the substitutions $\mathbf{v_i} = \dot{\mathbf{r}}_\mathbf{i}$ and $\dot{\mathbf{v}}_\mathbf{i} = \mathbf{a_i}$. Hence we can write it as a linear system $\dot{\mathbf{y}}(t) = \mathbf{g}(t, \mathbf{x})$, where

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_N \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{bmatrix}, \qquad \mathbf{g}(t, \mathbf{x}) = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \tag{3}$$

3 COMPUTATIONAL IMPLEMENTATION

We take a numerical approach to analysing the N body system, using the Python programming language. Our simulation implements a particle-particle based, collision-less algorithm, in order to simulate an arbitrary number of point masses in 3D space. See appendix section 8.2 for a link to the code written by the author.

3.1 leapfrog integration

To solve the governing equations from section 2 we employ the Leapfrog integration method to solve the second order differential equations. In the leapfrog method, the equations for updating position and velocity at any given time are given by

$$\begin{aligned} v_{i+1/2} &= v_i + \frac{1}{2} a_i \Delta t \\ x_{i+1} &= x_i + v_{i+\frac{1}{2}} \Delta t \\ v_{i+1} &= v_{i+\frac{1}{2}} + \frac{1}{2} a_{i+1} \Delta t \end{aligned} \tag{4}$$

Where a is the acceleration, v is the position, and x is the acceleration of any of the N particles, in any of the three coordinate axes. Note the subscripts represents the step number, and the velocity is updated twice at each time step Δt, where it "drifts" according to the previous acceleration a_i, and is than "kicked" with the updated acceleration a_{i+1}.

We decided on the Leapfrog method since it uses the same number of function evaluations (number of times acceleration is calculated) as the Euler method, while still being a second order integrator. Being second order, the error term is $O(\Delta t^2)$, compared to the much larger $O(\Delta t)$ error in the first order Euler method. There are higher order integration techniques, like the fourth order Runge-Kutta method, that could also be implemented. However, for orbital, dynamics the symplectic nature of leapfrog allows for conservation of energy - which is not conserved in other systems like Runge-Kutta, leading to the system drifting over time Quinn et al. (1997).

3.2 Computational complexity

The pairwise interaction between N particles means there are $\binom{N}{2} = \frac{1}{2}N(N-1) \sim N^2$ forces to calculate, leading to a computational time complexity $\Theta(N^2)$; meaning the computational time roughly increases with the square of the number of particles in the simulation. Given there are only $O(N)$ inputs (the N particles) gives lower bound time complexity of $\Omega(N)$, suggesting a more efficient algorithm. Indeed, there exists efficient approximation algorithms, with the current state of the art having complexity $O(Np^2)$ where p is a parameter related to the precision of the approximation scheme Reif & Tate (1993). In this paper, we make use of the particle-particle based $O(N^2)$ algorithm, for an in depth analysis of other algorithmic implementations see Blelloch & Narlikar (1997).

4 TESTING THE SIMULATOR

4.1 Conservation of energy

A very basic (but very useful) method for testing the accuracy of an N body simulator is by checking that the system conserves energy. Conservation of energy states that the total energy of a closed system is an invariant quantity, so the sum of the kinetic and potential energy of a system should not change over time. So if we calculate the kinetic energy and potential energy of all particles at each time step, their sum should not change throughout the simulation. Hence, if the total energy of the system begins changing - known as leakage - than we can no longer trust the simulation. We now show a particularly interesting case that was used to check the program conserves energy - the case of dynamical friction.

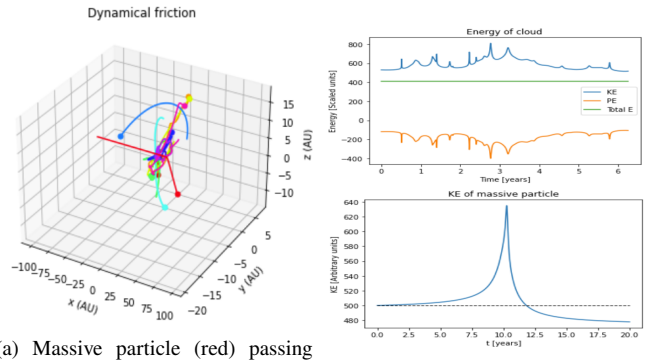
4.1.1 Dynamical friction

Dynamical friction occurs when a body travels through a region of matter, and loses kinetic energy through gravitational interaction. The reason for this drag can be seen by considering a large particle travelling through a cloud of smaller particles. As the large mass enters the cloud, its gravity pulls the particles in the cloud closer together, hence as it leaves the cloud their will be a "gravitational wake" that exerts a force on the exiting mass that is greater then the force applied as the particle was entering the cloud. Thus, the kinetic energy of the mass decreases as it escapes the gravitational effect of the cloud. This kinetic energy goes into heating the cloud.

We can simulate this effect with just a few particles. Consider the effect of a large particle passing through a dense region of smaller particles. As the particle enters the cloud, its kinetic energy will rapidly increase, as the gravitational force of the cloud accelerates the particle. The kinetic energy of the system will increase, and the close proximity of the particle will therefore decrease. As the massive particle exits the cloud, the kinetic energy will rapidly decrease due to the gravitational attraction of the cloud in its wake. In fact, the kinetic energy will decrease bellow that of its initial kinetic energy, with the lost energy being put into heating the cloud. These dynamics where recreated with the N-body simulator, and the results are displayed in figure 5. Note that despite the rapidly varying potential and kinetic energy of the system seen in figure 5b, the total energy of the system remains constant. This gives us some validation that our simulator is working properly.

4.1.2 Testing on periodic orbits

Conservation of energy is a necessary, but not sufficient condition for a correct program. Thus, in order to test the program further, we test our numerical implementation by simulating exactly solvable systems. A good test would be to simulate the figure 8 orbit. The figure 8 orbit is a periodic 3 body orbit, like the Euler and Lagrange solutions, but it also has the useful property of being a stable orbit. This is a very desirable property for numerical simulations, since any perturbation from an unstable orbit will be amplified over time and



(a) Massive particle (red) passing through cloud

(b) Energy of system

Figure 5. Dynamical friction simulation

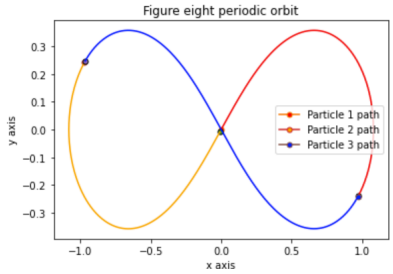


Figure 6. Figure 8 stable periodic orbits [xy projection, arbitrary spatial units]

kick the system out of equilibrium permanently, leading to a high level of instability that makes it difficult to observe numerically.

We use the following initial conditions, where each of the three particles all have a unit mass, and each of the units are in dimensionless form [1]

Initial conditions	
initial position	velocity
(0.97000436, -0.24308753, 0)	(0.4662036850, 0.4323657300, 0)
(-0.97000436, 0.24308753, 0)	(0.4662036850, 0.4323657300, 0)
(0, 0, 0)	(-0.93240737, -0.86473146, 0)

When simulating this orbit, the solution was indeed a periodic orbit as expected. See figure 6 for the xy projection of the orbit.

5 TROJAN ORBIT STABILITY

The stability of the L4 and L5 Lagrange points is apparent from the presence of Trojan asteroids that occupy regions surrounding the L4 and L5 Lagrange points of most Planets in our Solar system. Surprisingly, the stability of these points is actually dependant on the mass ratio of the large mass m_1 (in our case the Sun), to the smaller mass of the orbiting body m_2 (in our case Jupiter). It can be shown that the L4 and L5 points are unstable if (Greenhouse 2016)

[1] see appendix section 8.1 for the procedure to put equations of motion into dimensionless form.

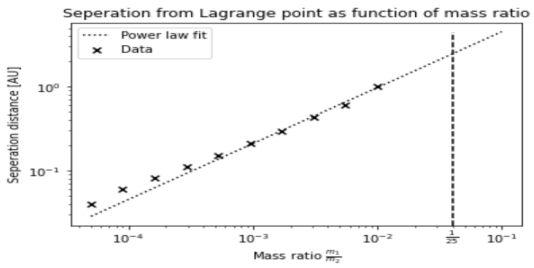


Figure 7. Max separation for low mass ratios

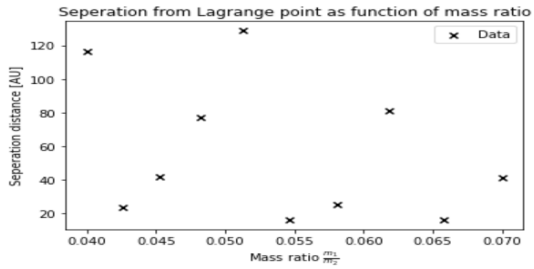


Figure 8. Max separation for high mass ratios

$$\frac{m_2}{m_1} \geq \frac{1-\sqrt{\frac{23}{27}}}{1+\sqrt{\frac{23}{27}}} \approx 1/25 \tag{5}$$

We strive to show this numerically by considering the distance a Trojan moves from its Lagrange point over ten orbits of the sun, with different mass ratios m_2/m_1.

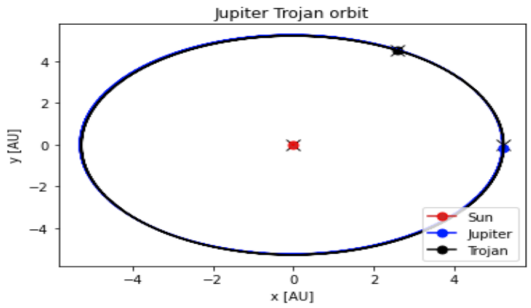
6 RESULTS AND DISCUSSION

Simulating ten orbits of a star located at the position of Jupiter, with a mass range of $5 \times 10^{-5} M_\odot$ to $\frac{1}{25} M_\odot$, we compute the maximum seperation distance between the L4 Lagrange point, and a Trojan asteroid that begins around the L4 point. we get the relationship between maximum separation between Trojan and L4/L5 Lagrange point shown in figure 7. This relationship is described quite well by a power law $d = 21AU \cdot (m_1/m_2)^{2/3}$, where d is maximum seperaion. The parameter error of the coefficient 21AU is 0.44. The coefficient does change as we change the length of time we run the simulation for, however the exponent $(2/3)$ appears to hold regardless of simulation time.

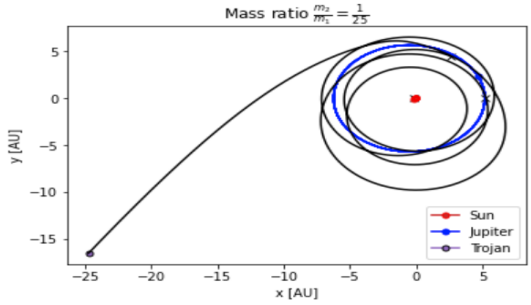
When we consider mass ranges above $\frac{1}{25} M_\odot$, this relationship disappears. As seen in figure 8, there appears to be no correlation between maximum separation distance and mass ratio. This can be understood by plotting the orbit of the three body system for different values of $\frac{m_1}{m_2}$, as shown in figure 9

Notice that for $\frac{m_2}{m_1}$ large enough, the orbit is no longer stable, and the Trojan is eventually ejected - as seen in figure 9b. When the Trojan is ejected has no obvious relationship with the separation at some fixed time, due to the chaotic nature of three body orbits (the Trojan is no longer in stable equilibrium, leading to onset of chaos). This explains why there is no relationship found in figure 8.

This agrees with the prediction given by Greenspan (2014), namely that the orbit becomes unstable with a mass ratio above 1/25. When it reaches this instability, it is eventually ejected from the system, as in



(a) Jupiter sized planet orbiting Sun ($\frac{m_2}{m_1} = 0.001$)



(b) At $\frac{m_2}{m_1} = 1/25$ system becomes unstable

Figure 9. System becomes unstable past critical threshold

9. Note, our simulation does not show that this point is exactly 1/25, it just shows that it is near 1/25. For a proof, the analytical arguments in Greenspan (2014) are required.

7 CONCLUSION

In this paper we have reviewed some aspects of the N body problem, and demonstrated a computer simulation to simulate an N body system. We tested our N body solver through one example demonstrating dynamical friction, and another example demonstrating the periodic figure eight orbit. We then used the code to simulate Trojan asteroids, and investigated the relationship between the ratio of the stationary and orbiting mass, and the distance the Trojan travels from its (L4 or L5) Lagrange point during its orbit. We used this to numerically demonstrate analytical results showing that L4/L5 Lagrange points become unstable once the mass ratio (m_2/m_1) between the orbiting and orbited star exceeds $\approx 1/25$.

8 APPENDIX

8.1 Scaling the equation of motion

In simulating equation 2, we note that the equation is plagued by extremely small and large numbers (e.g. $G \approx 6.67 \times 10^{-11}$), which do not work well with the limited precision of floating point numbers. So we strive to put equation into dimensionless form, such that the constants take on smaller values.

We introduce the scaled parameters

$$\mathbf{r}' \equiv \frac{\mathbf{r}}{L} \qquad m' \equiv \frac{m}{M} \qquad t' \equiv \frac{t}{\tau} \tag{6}$$

where L is the characteristic length scale (e.g 1AU for solar system simulations, 10kpc for galaxy simulations etc), M is the characteristic mass scale (e.g. M_{sun} for solar system), and τ is the time scale (units of time, e.g. 1 million years for galaxy formation simulation). The one free parameter is τ which we can choose to be such that it cancels the other parameters with dimensions in the problem. We find a convenient value of the parameter by plugging these values into the DE and setting the dimensionless group equal to 1.

$$\frac{d^2(\mathbf{r_i}'L)}{d(\tau t')^2} = \Sigma_{j\neq i} \frac{G(m_j' M)}{|\mathbf{r_i}'L - \mathbf{r_j}'L|^3}(\mathbf{r_i}'L - \mathbf{r_j}'L) \tag{7}$$

and hence

$$\frac{L^3}{GM\tau^3}\frac{d^2(\mathbf{r_i}')}{d(t')^2} = \Sigma_{j\neq i} \frac{m_j'}{|\mathbf{r_i}' - \mathbf{r_j}'|^3}(\mathbf{r_i}' - \mathbf{r_j}') \tag{8}$$

Identifying the dimensionless group $\frac{L^3}{GM\tau^2}$, and setting equal to one we get $\tau = \sqrt{\frac{L^3}{GM}}$. Using this time scale, and making the substitutions

$$\mathbf{r} \leftarrow \frac{\mathbf{r}}{L} \qquad m \leftarrow \frac{m}{M} \qquad t \leftarrow t\sqrt{\frac{L^3}{GM}} \tag{9}$$

we obtain the dimensionless form of the equation

$$\frac{d^2\mathbf{r_i}}{dt^2} = \Sigma_{j\neq i} \frac{m_j}{|\mathbf{r_i} - \mathbf{r_j}|^3}(\mathbf{r_i} - \mathbf{r_j}) \tag{10}$$

where the mass and length scales are to be chosen appropriately for a given problem, such that the lengths and masses are of order 1. Once the simulation is run, we can find the actual lengths, times, and masses at the end by plugging values back into the defining equations of the dimensionless variables. This reduces numerical errors.

8.2 Source code

The main code behind the simulator, aswell as code to generate all of the plots used throughout the paper (and others not seen in this paper), can be found at https://github.com/j-moncr/N-body-simulation/blob/main/Nbody-simulator.ipynb.

REFERENCES

Blelloch G., Narlikar G., 1997, Parallel Algorithms, 30, 81
Bruns H., 1887, Acta Mathematica, 11, 25
Greenhouse M., 2016, in 2016 IEEE Aerospace Conference. pp 1–11
Greenspan T., 2014
Li X., Jing Y., Liao S., 2017, arXiv preprint arXiv:1709.04775
Marzari F., Scholl H., Murray C., Lagerkvist C., 2002, Asteroids III, 1, 725
Moore C., 1993, Phys. Rev. Lett, 70, 3675
Oks E., 2015, The Astrophysical Journal, 804, 106
Quinn T., Katz N., Stadel J., Lake G., 1997, arXiv preprint astro-ph/9710043
Reif J. H., Tate S. R., 1993, in International Colloquium on Automata, Languages, and Programming. pp 162–176
Šuvakov M., Dmitrašinović V., 2013, Physical review letters, 110, 114301
Yamada K., Asada H., 2010, Physical Review D, 82, 104019

This paper has been typeset from a TEX/LATEX file prepared by the author.