ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D PHYS

Joseph El-Forzli

Barnes-Hut simulation for N-body problem

CSP semester project

Swiss Federal Institute of Technology (ETH) Zurich

June 09, 2023

Abstract

Analysis of a N-body system, as needed in molecular dynamic or astrophysics, often requires the use of simulation. Different algorithms are designed to describe mechanics of those systems, and optimising their performances can be challenging. In this report, we propose a study about the Barnes-Hut algorithm starting with a detailed explanation of its operation followed by an analysis of its performance. The save in computation time ($\mathcal{O}(N^2) \rightarrow \mathcal{O}(N \log N)$) and the error induced by the algorithm are measured and lead to a confirmation of its great usefulness.

Contents

1 Introduction

In physics, the study of a system is often reduced to the study of the interaction between a tremendous number of bodies. Usually, obtain an exact description of the system (all trajectories and momentum over time) is totally hopeless by the mean of a direct mathematical approach. It is in this context, that simulation takes a prominent place allowing the extraction of global behaviours and even important quantities. The N-body problem is a textbook case of this situation where it is impossible to obtain an explicit analytic solution. It consists of the study of N bodies interacting with each other according to a radial force (Van der Waals' force, gravity,..), describing systems from molecular dynamic to galaxy formation.

Since the dawn of computers, scientists have been interested in simulating the N-body system [1]. First trials uses a "trivial" algorithm with a complexity of $\mathcal{O}(N^2)$. At that time that wasn't a serious issue and a lot of results were obtained. But then, with the development of computing power, more ambitious studies have been undertaken, and the need for more powerful algorithm has grown.

Different solutions have been proposed, like the addition of a cut-off radius or the linked cell method. Now, one of the most widely used technique is called the Barnes-Hut algorithm or the Tree method [2]. It consists of an approximation of the long range interaction between clusters of bodies using a recursive division of the space. In principle it should reduce the scaling from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ giving a huge improvement in computation time.

In this work we propose a detailed description of the trivial and Barnes-Hut algorithm in the context of a galaxy simulation. Several performance analysis have been conducted like a verification of the scaling and an error analysis. Energy conservation and Virial's theorem have also been tested.

2 Methods

The N-body problem consists in the solving of the second-order differential equation :

$$\ddot{X} = f(X) \tag{1}$$

where X contains all bodies' position and f computes all acceleration. Knowing initial conditions of the system (X_0 and $\dot{X}_0$) it is possible to compute position at all wanted time using an integration process (like Euler, Verlet,.. methods).

2.1 Initialisation

To initialise the system corresponding to a galaxy, N particles are placed randomly in a flattened sphere, taking care to space them far enough apart. A massive particle is placed at the middle of the system, it helps for the stabilisation and it can be seen as a central black hole. Initial velocity is calculated using :

$$v(r) = \sqrt{\frac{GM(r)}{r}} \tag{2}$$

where $M(r)$ is the total mass contained in a cylinder of radius r and r is the distance of the considered body in the X-Y plan. This velocity is oriented such to create a rotational motion of the galaxy. A Gaussian noise is also added in order to perturbate the system.

2.2 Integration process

The Verlet integration process [3] is used to evolve the system. It has been chosen for its important qualities such as good stability, preservation of symplectic form (energy conservation) and high order of convergence (usually 4th order).

Knowing X_t and $X_{t-\Delta t}$ we obtain $X_{t+\Delta t}$ using the following scheme:

$$X_{t+\Delta t} = 2X_t - X_{t-\Delta t} + \frac{1}{2}\Delta t^2 f(X_t) \tag{3}$$

All that remains is to calculate f.

2.3 Determine acceleration

The major part of complexity in the N-body simulation emerges from the computation of f which is very resource-intensive. In our case, f should take into account all gravitational interactions between each bodies.

2.3.1 Regularisation of gravity

The exact force acting on a body at a position $\vec{r_i}$ is :

$$\vec{F_i} = \sum_{j \neq i} G\frac{m_i m_j}{r_{ij}^2}\vec{u}_{ij} \tag{4}$$

where G is the gravitational constant, m_i is the mass of the ith body, r_{ij} is the distance between the ith and the jth body and $\vec{u}_{ij}$ is a unit vector pointing from the ith body to the jth body. To obtain the acceleration, Newton's second law $\vec{a}_i = \frac{\vec{F}_i}{m_i}$ is obviously used.

The singular behaviour of the gravitational force can make the integration process unstable. A lot of studies have been conducted trying to solve this problem called "regularisation of the gravitational potential" [4]. In our case, we simply limit the force at short range introducing a new parameter such that if $r_{ij} < r_{min}$:

$$\vec{F}_{ij} = G\frac{m_i m_j}{r_{min}^2}\vec{u}_{ij} \tag{5}$$

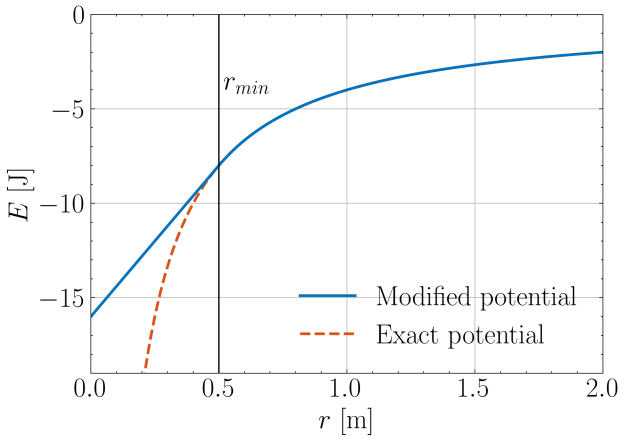

Figure 1: Regularised gravitational potential energy.

2.3.2 Trivial approach

As its name suggests, the Trivial approach consist to compute the exact force acting on each body giving a complexity of $\mathcal{O}(N^2)$. It also uses the Newton's third law ($\vec{F}_{ij} = -\vec{F}_{ji}$) to reduce the scaling to $\mathcal{O}(0.5N(N-1)) \approx \mathcal{O}(N^2)$, which is insufficient to makes this method really useful for large N.

2.3.3 Barnes-Hut method

This method tries to reduce the computational time of f by creating virtual bodies corresponding to distant clusters. At each time step it create an Octree structure [5] following the scheme:

1. Create a first cell containing all bodies.
2. If there is more than one particle, divide into sub-cell. Redo it while each individual particle doesn't have a corresponding cell. It will create a hierarchical graph structure.
3. For each node, create a virtual body at the center mass of all belonging bodies.

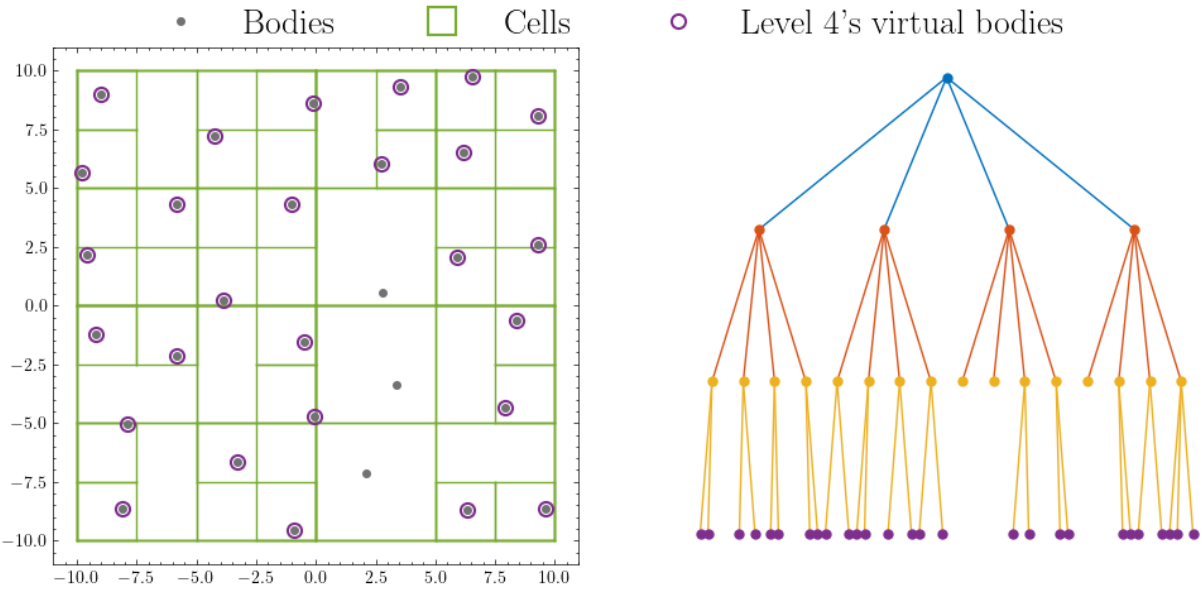

Figure 2: Space division of the system and its corresponding Quadtree (equivalent of an Octree in 2D) structure for $N = 30$.

It is now possible to calculate an approximation of the force acting on a considered body following this scheme :

1. Consider the first node.
2. - If the distance d between the considered body and the virtual body of the actual node respect the condition $\frac{\texttt{Cell's size}}{d} < \theta$ where `Cell's size` is the size of the longest side of the cell, take the interaction with the virtual body.
   - Else, consider each sub node and redo this step.

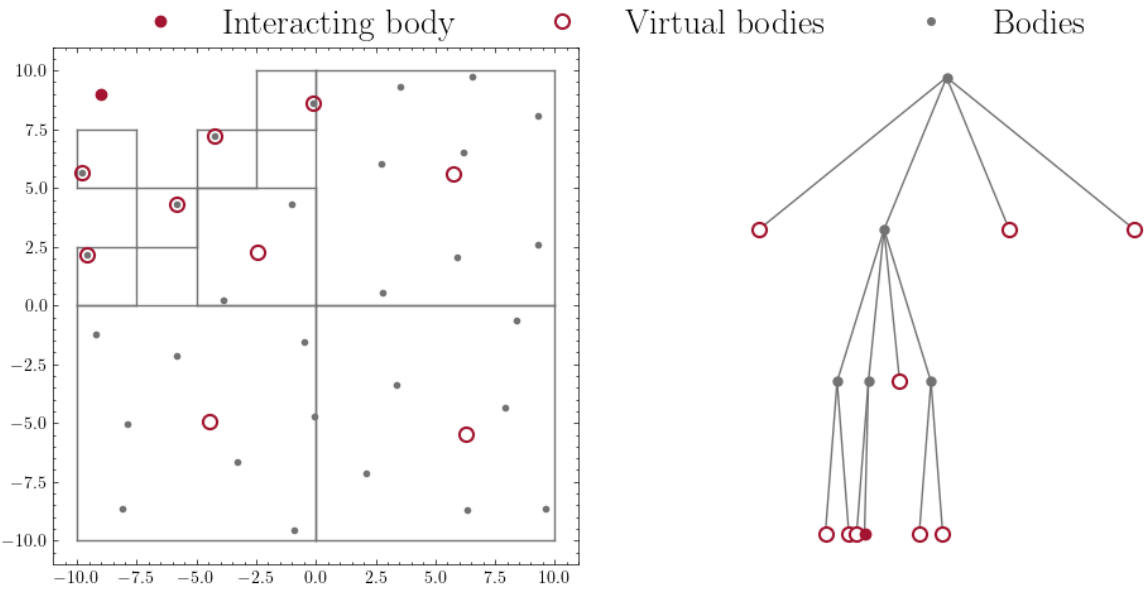

Figure 3: Virtual bodies in interaction with the considered body for $N = 30$ and $\theta = 0.7$.

This procedure will automatically sort from all interactions those for which more precision is needed; it will approximate long range interactions without neglect them as Linked Cell method would have done. The introduced θ parameter controls the level of approximation. As $\theta \to 0$, this method becomes closer to the trivial one, until it equals.

The illustration shown Figure 3 presents an example of the potential gain in computation time: only 9 virtual bodies enters in interaction with the considered body versus 29 in the trivial case.

2.4 Implementation and set-up

To see all details of implementation, please look at the `readMe.txt` and `main.py`, a python file attached to this report.

To compute simulation we have constructed a python version to evaluate performance and perform low N simulation running on our personal PC (`main.py` and `timePerformance.py`). For more ambitious simulation we have coded a CPP optimised and parallelized version (`barnes-hut.cpp`) running on a multi-core server. To visualise the results we use `matplotlib` for low N simulation and an homemade script in `Processing` (`https://processing.org/`) for high N simulation.

3 Results

All parameters have been tuned by hand to obtain good performance. You can find in the attached file `video` videos of different trials.

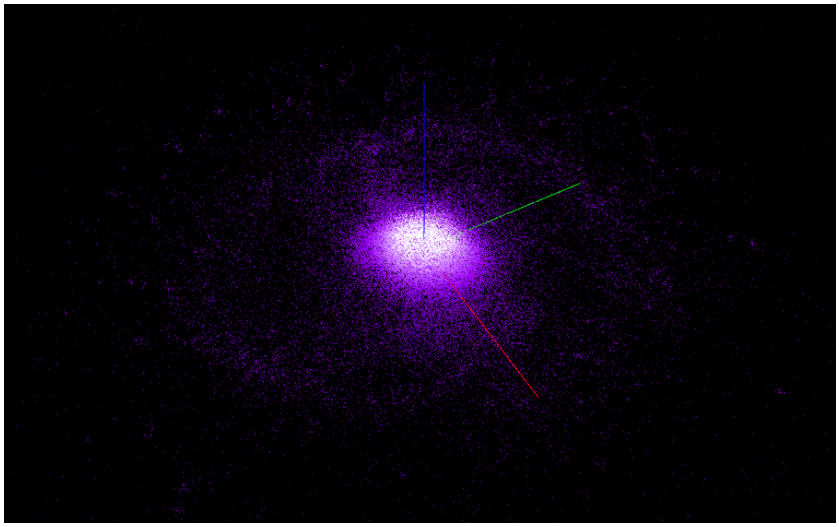

Figure 4: Image from a simulation with $N = 100000$ using the optimised CPP code.

After having checked by eye the physical behaviour of the system, an indicator that everything work well can be the verification of energy conservation and Virial's theorem. Figure 5 shows the evolution of these quantities over time for a system of $N = 50$ particles.

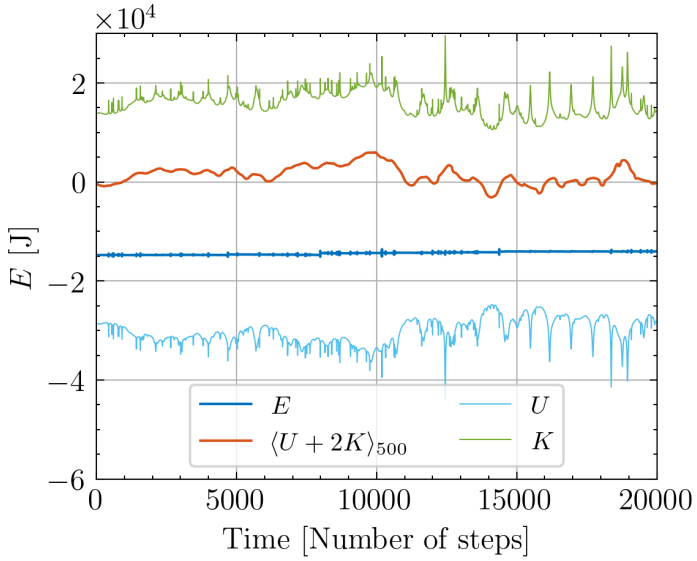

Figure 5: Energy and Virial evolution during the simulation with $N = 50$, $\theta = 0.7$ and $\Delta t = 2 \cdot 10^{-4}$.

To confirm the scaling of the Barnes-Hut algorithm, time of computation for different N have been measured and compared with that taken by the trivial method.

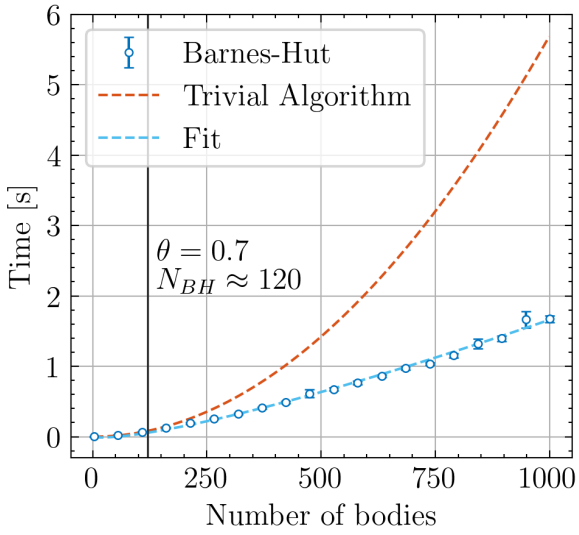

Figure 6: Time taken by one step in function of the number of bodies simulated with $\theta = 0.7$

A clear gap in the performance is observed, but this is done at the cost of approximations. It is important to measure the error induced by the employed method and its dependence in θ.

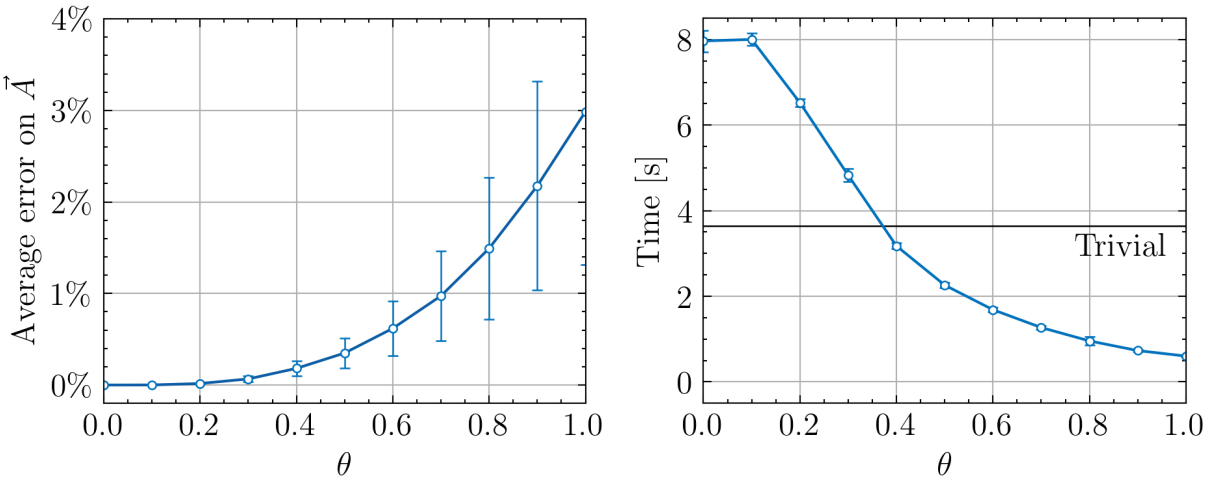

Figure 7: Average error on acceleration and time taken by one step in function of θ for $N = 800$.

Also, because of the complexity of Barnes-Hut algorithm's implementation, a lower performance is expected for low N. There should be a number of body N_{BH}, where Barnes-Hut approach begins to perform better than the trivial method.

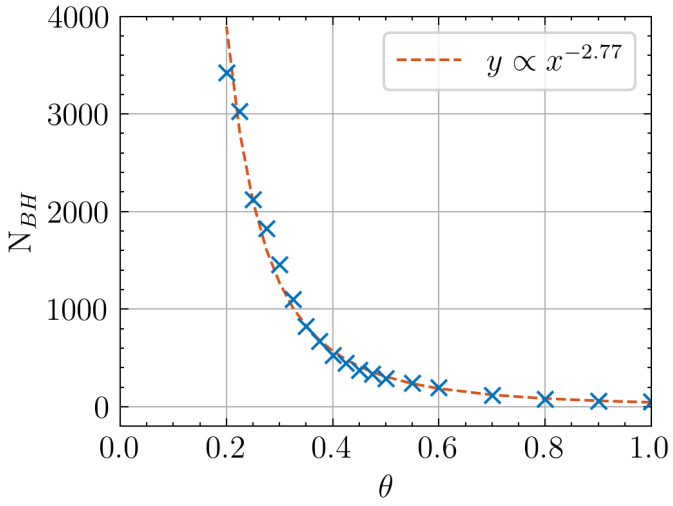

Figure 8: N_{BH} in function of θ.

4 Discussion

Simulations conducted with big N gives visually satisfactory behaviour, but deeper analysis of the result, like the computation of correlation function, is necessary to extract physical quantities.

The computation of exact energy needs a lot of resources ($\mathcal{O}(N^2)$), that is why it was conducted on a system of only $N = 50$. Its variation is shown Figure 5 and exhibits small discrete jumps. But after several trials, a net dependence of these variation in Δt is observed which suggest the implication of the integration process and not the Barnes-Hut method. The Virial's theorem [6], which affirms that a self gravitating system at equilibrium obeys:

$$2\langle K \rangle = -\langle U \rangle \tag{6}$$

seems to be verified, despite a small surplus of kinetic energy.

By measuring the time of computation for one step in function of N, the scaling in $\mathcal{O}(N \log N)$ is perfectly verified and illustrates the great interest of the used method; we estimate that our simulation of $N = 100000$ particles takes 200 times less time than the Trivial method. This huge gain makes highly ambitious simulations possible.

A fine control of error is often needed in order to make serious simulation. Figure 7 by showing the error on acceleration and the taken time by step in function of θ illustrates the trade-off between these two quantities. For all our simulations we

took $\theta \approx 0.7$ to keep good performance and to limit the error to 1%. Moreover, the N-body problem is highly chaotic [7], making virtually impossible to obtain long time individual trajectory by simulation. That is why slight error on acceleration is acceptable while a global behaviour is obtained.

Finally, the use of Barnes-Hut algorithm may seem over complicated for low N simulation. Figure 8 indicates the minimal number of body in function of θ for which Barnes-Hut method starts to become more efficient. As expected, when precise simulation is needed, N_{BH} grows up to become infinite at $\theta = 0$ (exact acceleration). But we were quite surprise to see how low N_{BH} is for the used θ ($N_{BH}(\theta = 0.7) = 120$).

5 Conclusion and Outlook

The implementation of Barnes-Hut algorithm offers the opportunity to work an a beautiful recursive algorithm using hierarchical graph structure. Thought different analysis it has shown all its interest in the computing of the N-body problem. In particular with its complexity of $\mathcal{O}(N \log N)$, with the possibility to have a fine control of error using the θ parameters and with its conservation of energy. Nowadays, this algorithm is largely used in conjunction with other method for simulation of large scale structure of the universe [8].

However, during this work some areas for amelioration have been devised : during the division process of the Octree generation, resources [9] take as point of division the geometric center of the cell while we have used the centre of mass (not on illustration for visibility). This, in principle, should limit the number of level in the Octree structure and with a new definition of cell's size shouldn't increase error (see the annexes 6). Giving a slight improvement in performance. But new studies need to be conducted in order to confirm this intuition.

6 Annexes

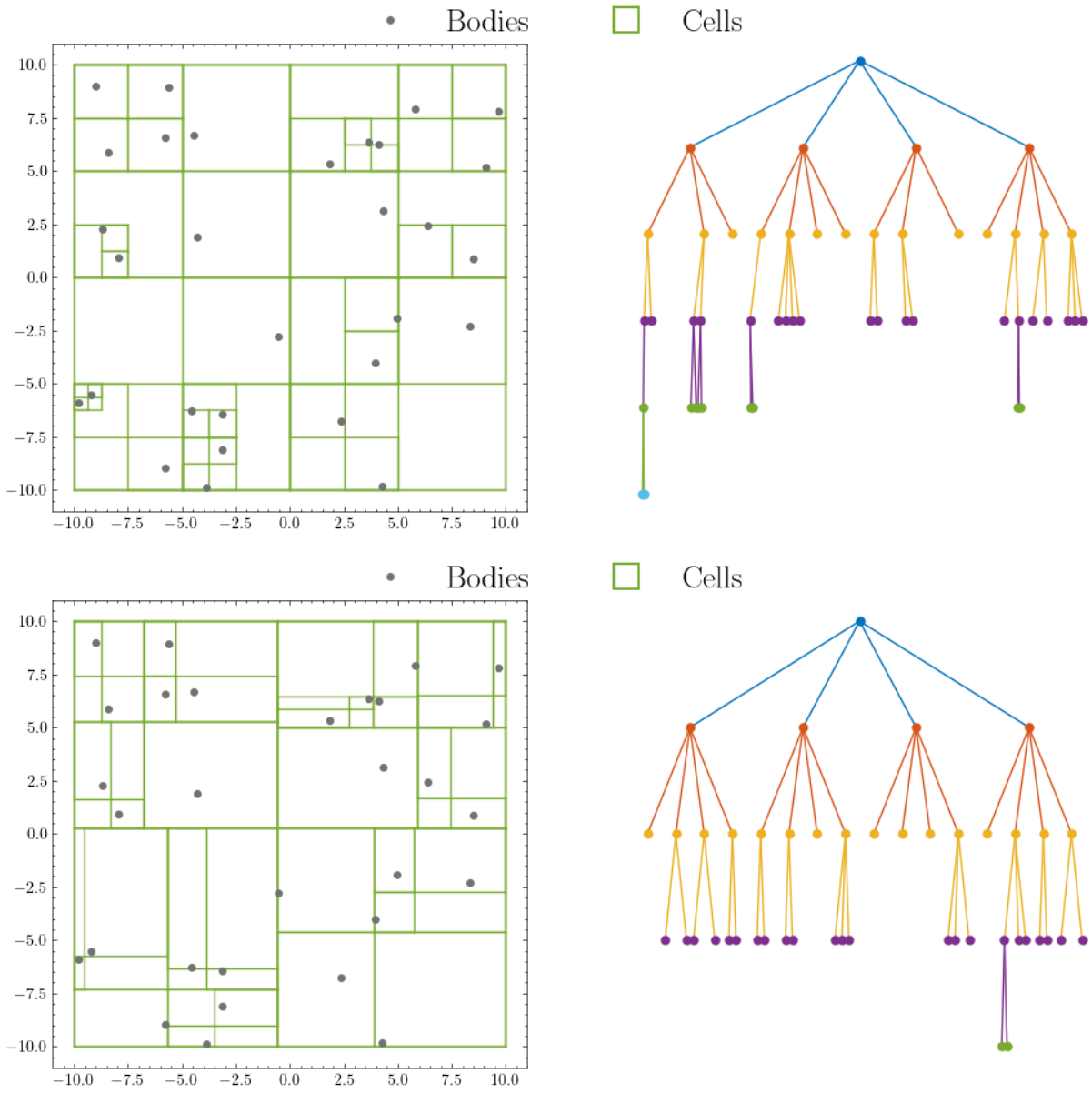

Figure 9: Space division of the system and its corresponding Quadtree structure for $N = 30$, division at the geometrical middle and at the center of mass.

The impact of the division location during the Octree generation process is illustrated Figure 9. In principle, a division at the center of mass should reduce the size of the Octree thanks to a better distribution of bodies in cells. But it create cells of unpredictable size; we probably need a new definition of cell's size during the calculation of acceleration.

References

[1] S. J. Aarseth and F. Hoyle. Dynamical Evolution of Clusters of Galaxies, I. *Monthly Notices of the Royal Astronomical Society*, 126(3):223–255, 06 1963.

[2] Josh Barnes and Piet Hut. A hierarchical o(n log n) force-calculation algorithm. *Nature*, 324(6096):446â449, 1986.

[3] Apr 2023.

[4] S. Mikkola. A brief history of regularisation. *Proceedings of the International Astronomical Union*, 3(S246):218â227, 2007.

[5] Octree structure. `https://en.wikipedia.org/wiki/Octree`, Jan 2023.

[6] Virial theorem. `https://en.wikipedia.org/wiki/Virial_theorem`, Apr 2023.

[7] Henri Poincare and Bruce D. Popp. *The three-body problem and the equations of Dynamics poincare s foundational work on dynamical systems theory.* Springer International Publishing, 2018.

[8] Tomoaki Ishiyama, Francisco Prada, Anatoly A Klypin, Manodeep Sinha, R Benton Metcalf, Eric Jullo, Bruno Altieri, Sofía A Cora, Darren Croton, Sylvain de La Torre, et al. The uchuu simulations: Data release 1 and dark matter halo concentrations. *Monthly Notices of the Royal Astronomical Society*, 506(3):4210–4231, 2021.

[9] Christian Swinehart. The barnes-hut algorithm. `http://arborjs.org/docs/barnes-hut`.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Title of work:

Barnes-Hut simulation for N-body problem

Thesis type and date:

CSP semester project, June 09, 2023

Supervision:

Student:

Name:	Joseph El-Forzli
E-mail:	jelforzli@student.ethz.ch
Legi-Nr.:	19-816-735
Semester:	2

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

`http://www.ethz.ch/faculty/exams/plagiarism/confirmation_en.pdf`

Zurich, 9. 6. 2023: