

## ASSESSING THE HIERARCHICAL HAMILTONIAN SPLITTING INTEGRATOR FOR COLLISIONLESS $N$ -BODY SIMULATIONS

G. Aguilar-Argüello, O. Valenzuela, H. Velázquez, J. C. Clemente, and J. A. Trelles

Instituto de Astronomía, Universidad Nacional Autónoma de México, México, CDMX, México.

*Received September 2 2021; accepted March 18 2022*

### ABSTRACT

The large dynamic range in some astrophysical  $N$ -body problems led to the use of adaptive multi-time-steps; however, the search for optimal strategies is still challenging. We numerically quantify the performance of the hierarchical Hamiltonian Splitting (HHS) integrator for collisionless simulations using a direct summation code. We compare HHS with the constant global time-step leapfrog integrator, and with the adaptive one (AKDK). We find that HHS is approximately reversible, whereas AKDK is not. Therefore, it is possible to find a combination of parameters such that the energy drift is considerably milder for HHS, resulting in a better performance. We conclude that HHS is an attractive alternative to AKDK, and it is certainly advantageous for direct summation and P3M codes. Also, we find advantages with GADGET4 (Tree/FMM) HHS implementation that are worth exploring further.

### RESUMEN

El gran intervalo dinámico en algunos problemas astrofísicos de  $N$ -cuerpos ha llevado al uso de pasos de tiempo múltiples adaptivos, sin embargo, la búsqueda de estrategias óptimas es aún un reto. Estudiamos numéricamente el integrador Hierarchical Hamiltonian Splitting (HHS) utilizando un código de suma directa y lo comparamos con el rendimiento de leapfrog con paso global constante y su versión multi-paso adaptivo (AKDK). Encontramos que HHS es aproximadamente reversible, mientras que AKDK no. Por lo que es posible encontrar una combinación de parámetros tales que el cambio de energía es considerablemente menor para HHS, resultando en una mayor eficiencia. Concluimos que HHS es una alternativa competitiva con ventaja para códigos de suma directa y P3M. También, hallamos ventajas para la implementación de HHS en GADGET4 (Árbol/FMM) que merecen ser investigadas más.

*Key Words:* celestial mechanics — galaxies: kinematics and dynamics — gravitation — large-scale structure of Universe — methods: numerical — software: simulations

### 1. INTRODUCTION

Historically, fully self-consistent realistic astrophysical  $N$ -body simulations are a challenging problem (Aarseth 1971; Efstathiou et al. 1985; Stadel 2001; Springel et al. 2001; Dehnen & Read 2011; Klypin 2018). On the purely gravitational case, direct summation  $N$ -body codes suffer from a bottleneck due to the computational cost of force calculation on a particle-by-particle basis, which scales as the square of particle number ( $N_p^2$ ). For this reason, they are primarily used for simulating dense stellar

environments or planetary systems. Such limitations triggered the development of sophisticated approximate hybrid collisionless methods, like the TreePM (Tree-Particle Mesh)/P3M (Particle Particle- Particle Mesh)/P3T (Particle Particle- Particle Tree) codes (Xu 1995; Bode et al. 2000; Bagla 2002; Bode & Ostriker 2003) where the short-range component of the force is carried out either by expensive/accurate direct summation or tree force solvers (Couchman 1991; Oshino et al. 2011; Habib et al. 2013). Alternatively, AMR (Adaptive Mesh refine-

ment) methods are used to compute the large dynamical range of the gravitational force (Villumsen 1989; Jessop et al. 1994; Kravtsov et al. 1997; Teyssier 2002), seeking a balance between accuracy and computational efficiency.  $N$ -body simulations require both a fast way to calculate the accelerations and an accurate and efficient integration method to evolve particles in time. The second-order leapfrog symplectic integrator (Verlet 1967) is the most widely used in collisionless  $N$ -body simulations (e.g. Klypin 2018; Angulo & Hahn 2021). It is strictly symplectic when a global-constant time-step is adopted; however, this is not suitable for addressing problems with a large dynamical range that are currently studied with modern codes. In the quest to improve efficiency, it is necessary to adopt multiple or adaptive time-steps. The general problem of geometric/symplectic (preserving phase space volume), time-symmetric (recover initial conditions after changing  $dt$  for  $-dt$ ) and reversible integrators (recover initial conditions after changing the sign of velocities) has been addressed in the field of differential equations for dynamical systems (Hairer et al. 2002). In such work, they point out the differences between adaptive global time-steps or multi-time-steps (several rates of evolution for different parts of the system) and discuss the constraints required for the integration method and the time-step selection function to preserve the mapping properties. In astronomy, the influential study of Quinn et al. (1997) discusses different operator-based leapfrog implementations using time step blocks plus a time-step selection function in the KDK/DKD leapfrog integrator for massive  $N$ -body simulations. They point out that particle migration across time-step blocks may break up the symmetry and sometimes involves backward integration, which can be difficult to reconcile with a dissipative component like gas. Current collisionless simulations codes commonly use the KDK leapfrog implementation with adaptive time-steps (e.g. Quinn et al. 1997; Springel 2005; Dehnen & Read 2011; Klypin 2018), which we call hereafter AKDK. Recently, Dehnen (2017) discusses conditions where the Hairer et al. (2002) analysis for time-symmetric integrators can be extended to discrete time-stepping. They conclude that there is no general solution. Many of the proposed integrators truly preserve the symmetries; however, the specific time-step selection function should also respect symmetrization. In several cases, the computational overhead makes the proposal impractical.

In this paper, we explore and quantify the Hierarchical Hamiltonian Splitting (HHS) strategy pro-

posed by Pelupessy et al. (2012), which is, as leapfrog, a second-order scheme. This integrator was tested for small number of bodies or collisional simulations, delivering good energy conservation and momentum conservation at machine accuracy. However, no analysis of time-symmetry or reversibility was presented. Some modern  $N$ -body codes like AREPO (Weinberger et al. 2020) and GADGET4 (Springel et al. 2021) have implemented versions of HHS with some differences with respect to the original proposal, although they do not give details of the performance or other properties that allow the comparison with the commonly used integrators. In this work, we extend the discussion of HHS in the context of collisionless  $N$ -body simulations, by numerically investigating the time-symmetry and velocity reversibility. We also test some time-step selection functions to explore the potential advantages. In all cases we compare with the global constant time-step leapfrog integrator and the adaptive one (AKDK) in order to assess the conditions under which HHS is a competitive alternative. As we discussed above, in some modern P3M codes running in hybrid architectures, the most expensive calculation is the short-range direct summation force integration, in some cases processed inside GPUs (Habib et al. 2016). Motivated by that, we implemented HHS in a direct summation code running in GPUs and complement that with additional tests with the Tree/FMM code GADGET4.

The rest of the paper has been organized as follows: § 2 summarizes the main properties of the integrators used here to carry out their comparison while § 3 introduces the time-step selection functions. § 4 shows accuracy tests performed with emphasis on the cases on an isolated halo and a minor merger. § 5, § 6 and § 7 contain the results of these tests taking into account the effect of time-step functions, performance and long-term stability, respectively. In § 8, we quantify reversibility and time-symmetry for the different codes. Finally, a discussion and the main conclusions are given in § 9.

## 2. INTEGRATORS

We implemented three different integrators in a direct summation code dubbed as  $\text{NP}_{\text{splitt}}$  (Aguilar-Argüello et al. in prep.), the leapfrog, Adaptive-KDK (AKDK) and the Hierarchical Hamiltonian Splitting integrators (HHS). Below, we describe each integrator. It is common to express integrators as a composition of operators using the Hamiltonian splitting technique in potential (*Kick*) and kinetic energy (*Drift*), although there are other possibilities (Oshino et al. 2011).

### 2.1. Leapfrog

The leapfrog integrator is a second-order widely used integrator. As mentioned previously, this integrator is strictly symplectic only when a global-constant time-step is adopted. Symplectic integrators are designed to numerically preserve the integrals of motion and the phase-space volume of the simulated system.

In the leapfrog method, the evolution of the gravitational system can be written as a sequence of *Kick* (advance of velocities) and *Drift* (advance of positions) operators (e.g. Channell 1993; Quinn et al. 1997), defined by:

$$\begin{aligned} K(dt) : \mathbf{v}(t_n + dt) &= \mathbf{v}(t_n) + dt \mathbf{a}(t_n), \\ D(dt) : \mathbf{x}(t_n + dt) &= \mathbf{x}(t_n) + dt \mathbf{v}(t_n), \end{aligned} \quad (1)$$

where  $\mathbf{x}$ ,  $\mathbf{v}$  and  $\mathbf{a}$  are the position, velocity and acceleration of a particle, respectively, and  $dt$  is the time step. In this paper, we use the operator sequence called KDK leapfrog (also known as *velocity Verlet*, Swope et al. 1982):

$$KDK : K(dt/2) D(dt) K(dt/2), \quad (2)$$

where we consider that the evolution is for one time step, i.e. from  $t_n$  to  $t_n + dt$ . Through this paper, we will refer to KDK leapfrog with a global-constant time-step as the Leapfrog integrator.

### 2.2. AKDK

Contemporary codes have extensively used KDK (equation 2) combined with a block time-step scheme (Hayli 1967; Sellwood 1985; Hut & McMillan 1986; Hernquist & Katz 1989; Makino 1991), frequently using rungs which are power of two:  $dt_r = dt_0 2^{(-r)}$  and different assigning time-step functions, most frequently an acceleration based one (Springel 2005). We will use it as a reference integration scheme, but it should be noted that it is not symplectic (Hairer et al. 2002) and that the block-step is a multi time-step scheme.

### 2.3. Hierarchical Hamiltonian Splitting

The hierarchical Hamiltonian Splitting (HHS) method is a second-order integrator that uses individual time steps of the particles (Pelupessy et al. 2012) through recursively splitting the Hamiltonian. It accurately preserves linear and angular momentum and has a good energy conservation.

This integrator consists of adaptively and recursively splitting the Hamiltonian as a function of the current time step,  $dt$ , so that the so called *Slow* system (hereinafter  $S$ ) contains all the particles with a

time step larger than  $dt$ , and the so called *Fast* system (hereinafter  $F$ ) contains all the particles with a time step smaller than  $dt$ . Thus, the splitting is as follows:

$$\begin{aligned} H_S &= T_S + V_{SS} + V_{SF}, \\ H_F &= T_F + V_{FF}, \end{aligned} \quad (3)$$

where:

$$\begin{aligned} T_X &\equiv \sum_{i \in X} \frac{p_i^2}{2m_i}, \\ V_{XX} &\equiv -G \sum_{i \in X} \sum_{j \in X, j > i} \frac{m_i m_j}{|r_i - r_j|}, \\ V_{XY} &\equiv -G \sum_{i \in X} \sum_{j \in Y} \frac{m_i m_j}{|r_i - r_j|}, \end{aligned} \quad (4)$$

are, respectively, the kinetic and potential energies, and  $V_{SF}$  is the potential energy of the interactions between  $S$  and  $F$  particles. The previous splitting scheme is known as HOLD (since it “holds”  $V_{SF}$  for evaluation at the slow time-step, Pelupessy et al. 2012).

The  $S$  system is solved using the DKD scheme (also known as *position Verlet*, Tuckerman et al. 1990), which consists of *drifts* of the particles in this system (due to  $T_S$ ) and *kicks* on the particles of both systems (due to  $V_{SS} + V_{SF}$ ). For the  $F$  system, the same procedure as for the original system is applied but using a halved time-step. Hence, the splitting is applied recursively to the  $F$  system with time-step  $dt/2^r$ . The recursion ends when the system  $F$  (of the rung  $r$ ) has no particles. At the end of the current integration step, the new time-step of a particle is calculated. In this scheme, a particle can change its time step to higher (lower) value if its current integration time is synchronized with the higher (lower) rung.

It is well known that the *Kick* and *Drift* operators are symplectic. However, using multiple or adaptive time-steps may not preserve such properties in a general way (Hairer et al. 2002). Therefore we need to investigate the behaviour of HHS.

## 3. TIME-STEP SELECTION FUNCTION

Besides the formulation of integrators with individual time steps based on symmetric operators, the choice for each particle time step is made through the so called time-step selection function. There is no unique choice; arguably the most commonly used time-step function in contemporary collisionless  $N$ -body codes (e.g. GADGET, Springel 2005) is based on the acceleration as:

$$\tau_i = \eta_{\text{accel}} \sqrt{\frac{\epsilon}{a_i}}, \quad (5)$$

where  $a_i$  is the acceleration acting on the particle  $i$ , giving the code the possibility of adapting to high/low accelerations, and  $\epsilon$  is the force Plummer softening<sup>1</sup>. Improvements have been recently discussed by using a dynamical time proxy (Zemp et al. 2007) and a tidal force time scale (Dehnen & Read 2011; Grudić & Hopkins 2020), establishing a balance between short and long time steps, which may translate into higher efficiency. Extensive comparisons of AKDK with both choices have been discussed in Zemp et al. (2007) and Grudić & Hopkins (2020).

In our study, in an attempt to preserve the energy stability of the HHS integrator while allowing adaptive multi time-steps, and following Pelupessy et al. (2012), we use an approximated time-symmetrized time-step extrapolation criterion for each particle. To obtain such a time-step criterion, we start from the implicit criterion (Hut et al. 1995):

$$\tau_{\text{sym}} = \frac{1}{2} [\tau(t) + \tau(t + \tau_{\text{sym}})], \quad (6)$$

where  $\tau$  is a time-step function (non-symmetrized), and  $\tau_{\text{sym}}$  is the symmetrized time-step function of  $\tau$ . To a first-order perturbative expansion:

$$\tau(t + \tau_{\text{sym}}) \approx \tau(t) + \frac{d\tau}{dt} \tau_{\text{sym}}; \quad (7)$$

hence, from equation 6:

$$\tau_{\text{sym}} \approx \tau(t) + \frac{1}{2} \frac{d\tau}{dt} \tau_{\text{sym}}, \quad (8)$$

so that the time-step we will use is given by (Pelupessy et al. 2012)

$$\tau_i = \min_j \left[ \frac{\tau_{ij}}{\left(1 - \frac{1}{2} \frac{d\tau_{ij}}{dt}\right)} \right]. \quad (9)$$

It is important to state that the minimization indicated above is across the so called *Slow* particles. For a time-step proportional to the inter-particle free-fall time:

$$\begin{aligned} \tau_{ij} &= \eta_{\text{FF}} \sqrt{\frac{r_{ij}^3}{G(m_i + m_j)}}, \\ \frac{d\tau_{ij}}{dt} &= \frac{3\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{2r_{ij}^2} \tau_{ij}. \end{aligned} \quad (10)$$

<sup>1</sup>We will adopt the softening as twice the average inter-particle distance at the minimum radius where the density profile is not dominated by Poisson fluctuations, as it is usually adopted in collisionless simulations.

The former option is a two-body-based proxy for the dynamical-time-motivated step function suggested by Zemp et al. (2007).

For completeness with Pelupessy et al. (2012), for a time-step proportional to the inter-particle fly-by time (typically used in collisional problems):

$$\begin{aligned} \tau_{ij} &= \eta_{\text{FB}} \frac{r_{ij}}{v_{ij}}, \\ \frac{d\tau_{ij}}{dt} &= \frac{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{r_{ij}^2} \tau_{ij} \left( 1 + \frac{G(m_i + m_j)}{v_{ij}^2 r_{ij}} \right). \end{aligned} \quad (11)$$

We will quantify the efficiency of such time-step functions. However, the high acceleration derivatives in the case of collisional problems may require going beyond the first order in the perturbative expansion.

Along the paper, we will mostly use the approximated symmetric free-fall time-step for HHS (defined by equations 9 and 10), and only in a few tests we will use the minimum of this and the approximated symmetric fly-by time-step (equations 9 and 11). In § 5, we will present a comparison with the GADGET4 implementation of HHS (Springel et al. 2021); this code uses a time-step function similar to equation 5 but the accuracy parameter,  $\eta_{\text{accel}}$ , is included inside the square root. For AKDK, we will use the standard time-step function given by equation 5.

Table 1 summarizes the combinations of integrators and time-step selection functions used through this work.

#### 4. ACCURACY TESTS

In this section, we present the test results of the HHS algorithm in terms of accuracy by simulating an isolated halo and sinking satellites, and compare them with the global-constant time-step leapfrog and the adaptive one, AKDK. To proceed with the comparison we implemented the different integrators in a direct summation  $N$ -body code (Aguilar-Argüello et al. in prep.). All the experiments were run in a single GPU. As a sanity check, we performed binary system tests (not reported here) and the results are consistent with those reported in other works (e.g. Dehnen & Read 2011; Pelupessy et al. 2012; Springel 2005).

##### 4.1. Isolated Cuspy Halo

We adopted as a reference model an equal particle mass, isolated halo following the NFW cuspy density profile predicted by collisionless dark matter cosmological simulations (Navarro et al. 1997). The large density range and the corresponding different dynamical times make it a suitable system for an

TABLE 1  
COMBINATIONS OF INTEGRATORS AND TIME-STEP FUNCTIONS USED ALONG THIS PAPER

Integrator name	Integration scheme	Time-step scheme	Time-step selection function	Reference
Leapfrog	KDK	Global-constant		Verlet (1967)
AKDK	KDK	BLOCK	eq. 5	e.g. Quinn et al. (1997); Springel (2005) Dehnen & Read (2011); Klypin (2018)
HHS	HHS	HOLD	eqs. 9, 10	Pelupessy et al. (2012)
nsHHS	HHS	HOLD	eqs. 9, 10 (without $d\tau_{ij}/dt$ )	This work
sAKDK	KDK	BLOCK	eqs. 9, 10	This work

adaptive time-step code. Such tests depend on resolution to actually capture the benefit of individual time-steps as compared with a global-constant time-step scheme. We will use as a reference time scale the dynamical time<sup>2</sup> at the NFW characteristic radius ( $r_s$ ), since it has been used to study the stability of the halo in other works (e.g. Klypin et al. 2015).

For the integration of our fiducial model, we adopted  $G = 1$  (gravitational constant),  $M_{\text{vir}} = 1$  (virial mass) and  $r_s = 1$  (scale length, also called characteristic radius), as model units. We will use these model units through the paper.

To investigate and quantify differences in accuracy and performance between the integrators first, we followed a fiducial halo sampled with  $10^5$  particles for 40 dynamical times at  $r_s, t_{\text{dyn}}$ .

Because our implementations of AKDK and HHS have different time-step function, a meaningful comparison is to assume an energy conservation threshold, which implies using distinct accuracy parameters for both integrators. For the first test, we considered a  $10^{-7}$  threshold and accuracy parameters  $\eta_{\text{FF}} = 0.003$  and  $\eta_{\text{accel}} = 0.01$  for HHS and AKDK, respectively, both constrained to 6 time-step rungs. Figure 1 shows the result of these tests. The upper left panel shows the energy error. HHS (black) stays very close to Leapfrog (red) during the first  $20 t_{\text{dyn}}$ , afterwards it shows a small drift. AKDK (blue) drifts almost linearly and after  $20 t_{\text{dyn}}$  it slightly flattens. Because the main computational overhead of HHS over AKDK comes from building and updating the time-step hierarchy in HHS, we decided to explore experiments where we delayed such an update, and denoted them as HHS-sTSS. We observed that such an action results in important savings in com-

putational time (yellow line). The energy accuracy test is lower but acceptable for a collisionless simulation, and it is faster; in addition linear and angular momentum are preserved to machine precision.

Regarding the conservation of other dynamical quantities like linear and angular momentum or the system barycenter the situation is different (see Figure 1). Leapfrog and HHS preserve almost at machine precision the linear and angular momentum (see bottom panels), whereas AKDK presents a smaller accuracy, although it has a slope until  $10 t_{\text{dyn}}$ , afterwards it flattens. Interestingly, HHS with a delay in updating the time-step hierarchy (HHS-sTSS, yellow) is almost indistinguishable from HHS and leapfrog. The upper right panel shows the accuracy in preserving the halo centroid. Once again, leapfrog and both HHS versions accurately keep the centroid, while the AKDK accuracy is degraded two orders of magnitude. The wall-clock time for leapfrog, HHS and HHS-sTSS experiments was, respectively, 1.7, 0.8 and 0.6 times the corresponding for AKDK. As a complement, we performed tests using larger time steps reaching lower energy accuracy, the general results are the same. Finally, we emphasize that all integrators accurately preserve the density profile, as we can see in Figure 2; with obvious dependence on the number of particles, we decided to show the test with two million particles in order to minimize discreteness effects.

#### 4.2. Minor Merger: Sinking Satellites

Satellite accretion onto larger galaxies is an astrophysical problem commonly simulated by both isolated and cosmological  $N$ -body simulations (Miller et al. 2020; Arca-Sedda & Capuzzo-Dolcetta 2016). We simulated a satellite, represented by a softened and massive particle, falling into a spherical system, represented by collisionless softened particles, for seven dynamical times. The spherical system

<sup>2</sup>A dynamical time, also called crossing time, is the time taken for a typical particle to cross the system. In this paper, a dynamical time is defined as  $t_{\text{dyn}}(r) = [r^3 / GM(r)]^{1/2}$ , where  $r$  and  $M(r)$  are the radius and mass, respectively.

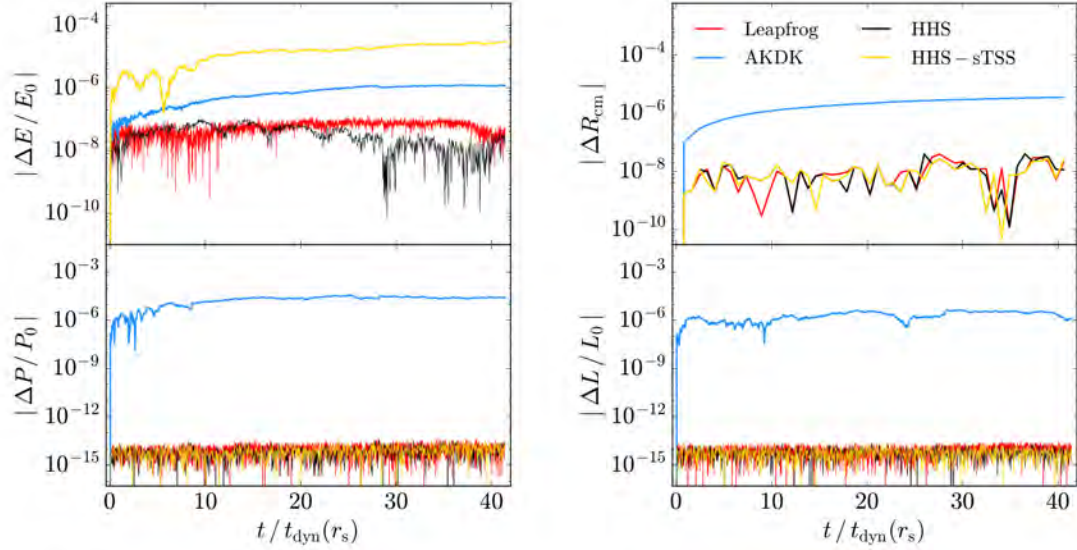


Fig. 1. Error in conserved quantities for an isolated NFW halo with  $N = 10^5$  particles, simulated up to 40 dynamical times (at scale radius,  $r_s$ ). Shown is the energy error (upper left panel), change in center of mass position (upper right panel), error in linear (lower left panel) and angular momentum (lower right panel) for the three integrators: Leapfrog (red), AKDK (blue) and HHS (black). Also, it is shown the HHS with a delay in the time-step hierarchy update (HHS-sTSS, yellow) version. The color figure can be viewed online.

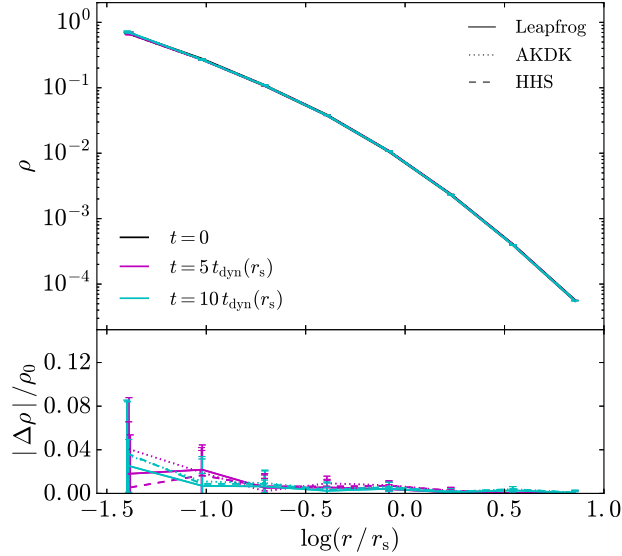


Fig. 2. Density profile, at different times, of the isolated NFW halo with  $2 \times 10^6$  particles simulated up to 10 dynamical times for the three integrators: Leapfrog (solid lines), AKDK (dotted lines) and HHS (dashed lines). The three integrators accurately preserve the density profile. The color figure can be viewed online.

consists of  $N = 10^5$  equal mass particles spatially distributed according to the NFW density profile (Navarro et al. 1997). We adopted the same units

as the previous fiducial isolated halo case. We used Plummer softening and we chose a softening parameter  $\epsilon = 0.026$  (in model units). The satellite's initial separation from the center of the spherical system is  $R_{\text{sat}} = 2.6$ , and its mass is  $m_{\text{sat}} = 0.01$ , which is  $\approx 1300$  times bigger than the mass of one collisionless particle.

This test is particularly useful because the sinking process involves orbital angular momentum and energy transfer into the host system. We tracked energy, linear and angular momentum conservation as well as the host center of mass behavior (see Figure 3). Leapfrog (red) stays flat. HHS (black) starts to jump at one dynamical time, afterwards it stays flat. AKDK (blue) shows a lower accuracy in energy conservation and presents a systematic energy growth. As in the case of the isolated halo, the HHS energy drift slope is considerably flatter than the corresponding to AKDK. Linear and angular momentum are less accurate for AKDK by several orders of magnitude, while system barycenter behaves essentially the same for all three integrators. These results indicate that HHS is an excellent alternative for dynamical friction studies.

The evolution of the satellite radial position shows differences below the one percent level (see Figure 4). We conclude that, for a reduced number of dynamical times, the three integrators can provide an accurate description of the sinking process.

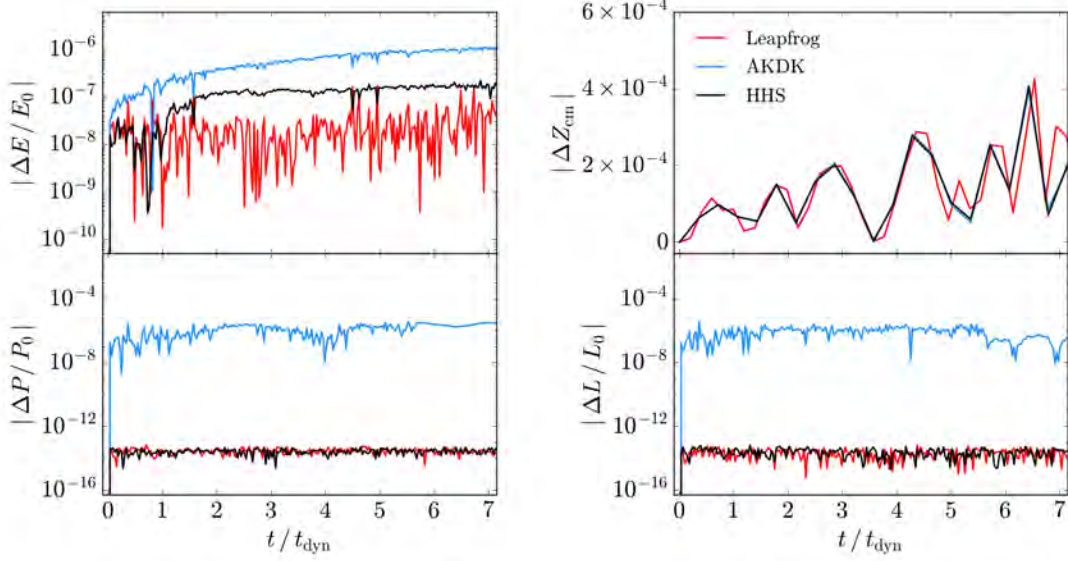


Fig. 3. Error in conserved quantities for a NFW halo with  $N = 10^5$  particles plus a satellite simulated up to 7 dynamical times. Shown is the energy error (upper left panel), change in center of mass position (upper right panel), error in linear (lower left panel) and angular momentum (lower right panel) for the three different integrators: Leapfrog (red), AKDK (blue) and HHS (black). The color figure can be viewed online.

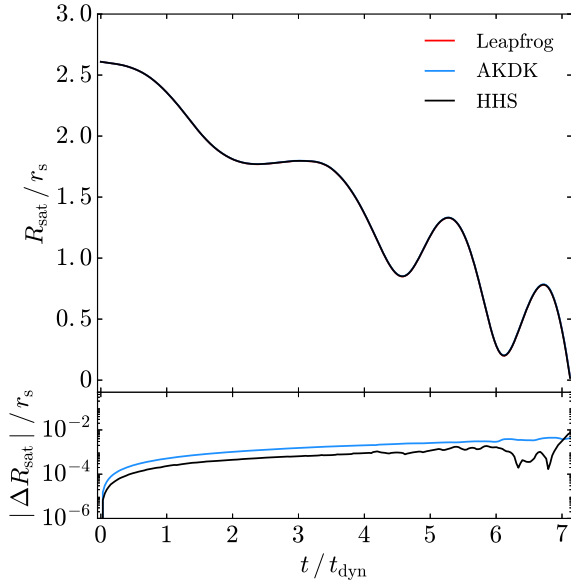


Fig. 4. Evolution of the satellite radial position (upper panel) for the sinking satellite test and for the three integrators: leapfrog (red), AKDK (blue) and HHS (black). When comparing with leapfrog (lower panel), the differences are below 1% between integrators. The color figure can be viewed online.

## 5. TIME-STEP SELECTION FUNCTION TESTS

As it has already been discussed some hierarchical/adaptive time-step integrators, like HHS and

AKDK, include a time-step selection function; such a prescription may help restoring the integrator symmetry. We may question if the approximated symmetric time-step selection function based on particle pairs given by equation 9 is only useful for a particular kind of simulation or code, and if we lose all the convenient properties of HHS, observed at this point, when the problem is not tractable by a direct summation code. To quantify such an effect we evolved the fiducial isolated halo switching the time-step selection function and we additionally performed test with the Tree/FMM code GADGET4.

### 5.1. Direct Summation Code

First, we consider our direct summation code implementation, simulating the fiducial isolated halo considering the HHS and AKDK integrators, as it can be seen in the upper panel of Figure 5. We take as a reference the AKDK integrator with an accuracy parameter  $\eta_{\text{accel}} = 0.08$  (fiducial, solid blue line). The dotted black line shows HHS (with  $\eta_{\text{FF}} = 0.055$ ), which is almost twice faster but slightly less accurate, and the dashed black line shows a case of HHS (with  $\eta_{\text{FF}} = 0.015$ ) remarkably more accurate but 10% slower than the fiducial AKDK. We include another AKDK case (dashed blue) with a smaller accuracy parameter  $\eta_{\text{accel}} = 0.049$ , which is 20% less accurate than HHS (dashed black line) but 10% slower, and also is 20% slower and an order of magnitude more accurate than the fiducial AKDK

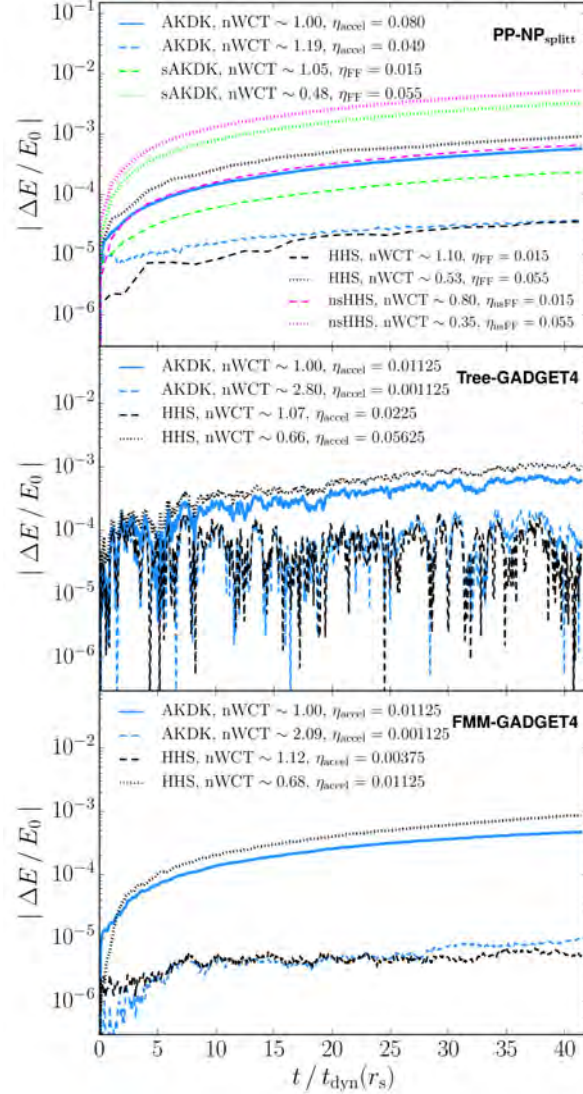


Fig. 5. Effect of time-step selection function and different acceleration codes. nWCT corresponds to the wall-clock time of each test normalized to the fiducial AKDK (solid blue line). Upper panel shows tests with our direct summation code. Solid blue line is the fiducial AKDK test and dotted black line is HHS, almost 50% faster. Dashed lines correspond to experiments with smaller accuracy parameter but reaching a limit where HHS is faster and more accurate than AKDK. Magenta dotted/dashed lines show the same HHS tests but neglecting the symmetrizing derivative term in equation 9; even going with smaller  $\eta_{\text{nsFF}}$  still may be competitive with AKDK. Green dotted/dashed lines show AKDK test with the symmetrized time-step selection function (equations 9 and 10) using the same parameters as HHS cases (black dotted/dashed lines, respectively). Middle panel shows experiments with GADGET4 using the Tree version. Results are consistent with the direct summation code. Solid blue line is the fiducial AKDK, dotted black line is HHS 7% slower and slightly less accurate. Dashed lines are AKDK and HHS, with almost flat behaviour, slightly better for HHS; however, AKDK is almost three times slower. Lower panel shows the equivalent tests but now for FMM GADGET4. As before, the AKDK for the flat case (dashed blue) is slightly worse in energy accuracy and almost twice slower as compared with HHS (dashed black). We conclude that, even for different time-step functions, there is still a regime where HHS is more efficient. The color figure can be viewed online.

case (solid blue). This means that considering even smaller values for the parameter  $\eta$  will not make AKDK more efficient than HHS. Next, we performed

some changes in the time-step selection function, we removed the derivative term from equation 9 which represents the symmetrizing correction for the HHS

integrator, we dubbed such tests nsHHS and they appear as the magenta lines in Figure 5. The energy drift is larger as compared with HHS but it is still acceptable for collisionless simulations, and it is faster. For completeness, we performed tests using equation 9 in AKDK (dubbed as sAKDK, green lines) with the same parameters ( $\eta_{\text{FF}}$ , rungs, etc.) as in the HHS tests. The energy drift is larger for the sAKDK test than the one corresponding to HHS; however, sAKDK is faster. Note that it is possible to match HHS accuracy by lowering the  $\eta_{\text{FF}}$  parameter in sAKDK tests, but it is slower (we do not show it because it does not add new information). So far, our tests suggest that HHS may benefit hybrid codes that use direct summation force calculation as part of their algorithm. For example, in the P3M technique (Hockney & Eastwood 1998) the most expensive part corresponds to direct summation, which runs in accelerators like GPUs (Habib et al. 2013, 2016; Cheng et al. 2020). Even using different time-step selection function, HHS may still obtain considerable performance. We analyze such a situation in the following sections, as well as the HHS performance in codes using approximated force computations (e.g. a tree code).

### 5.2. Tree Code: GADGET4

Recently, the 4th version of the publicly available code GADGET has implemented the HHS in the so called hierarchical gravity mode (Springel et al. 2021). The time-step selection function is similar to our equation 5 but the accuracy parameter  $\eta_{\text{accel}}$  is inside the square root. Changing the step function is a sensible choice because our equation 9, based on pair interactions, is not practical for very large numbers of particles. Also, in the GADGET4 HHS implementation, instead of using the DKD representation (as in our implementation), the authors adopted the KDK one (for further details we refer to Springel et al. 2021), in a similar way as Zhu (2017). Hence, this allow us to explore the case of an approximated gravitational acceleration code, such as a tree code (e.g. Barnes & Hut 1986), with different time-step selection function. For that purpose, we perform some tests using AKDK and HHS with the Tree version of GADGET4. As in the case of direct summation tests (§ 5.1), we simulated the fiducial isolated halo in such a way that we can directly compare with the direct summation code tests, including a non-symmetrized time-step function. The middle panel of Figure 5 shows the results for the Tree code version of GADGET4. For the corresponding fiducial AKDK (solid blue line), GADGET4 opens three time-step rungs,

and for its HHS implementation (dotted black line), it opens only two time-step rungs. Both integrators show an energy drift similar to  $10^{-3}$ , however, HHS (dotted black line) is 66% faster, but less accurate. Motivated by this result, we ran a new HHS test decreasing the  $\eta_{\text{accel}}$  parameter (dashed black line); the energy is quite flat; therefore, energy conservation after  $40 t_{\text{dyn}}$  is almost an order of magnitude better than the fiducial AKDK (solid blue line), but it is 7% slower. If we decrease  $\eta_{\text{accel}}$  for the AKDK integrator in order to match energy conservation (dashed blue line), the wall-clock time is considerably larger, by 50%. Although this is a particular example model, it is consistent with our previous tests. HHS is more stable and for medium and long-term integrations may be more efficient than AKDK, regardless of not using the approximated symmetric free-fall particle pairs time-step function (equations 9 and 10).

### 5.3. Fast Multipole Method Code: GADGET4

Gadget4 has implemented a different gravity solver based on the Fast Multipole Method expansion (FMM, e.g. Dehnen 2000, 2002). We ran the same fiducial isolated halo, as in the previous section, adopting the FMM scheme truncating the expansion at the quadrupole term. Results are presented in the lower panel of Figure 5. The blue solid line shows the fiducial case using AKDK ( $\eta_{\text{accel}} = 0.01125$ ), the corresponding HHS case ( $\eta_{\text{accel}} = 0.01125$ , dotted black line) is relatively faster (66%); however, it is less accurate. We experimented lowering the  $\eta_{\text{accel}}$  parameter for HHS, and the energy evolution is flat (dashed black line). We also decreased  $\eta_{\text{accel}}$  for AKDK (dashed blue line); the energy accuracy is 10% worse but the wall-clock time is almost 100% larger; therefore, there is no point in trying smaller  $\eta_{\text{accel}}$  values. The experiments with the GADGET4 FMM version also confirm the conclusions from our direct summation tests.

## 6. PERFORMANCE

Although the adaptive nature of the HHS and AKDK integrators may imply a higher efficiency compared with the global-constant time-step leapfrog, the benefit of taking adaptive time-steps is evident only when the dynamical range is large. In comparison with AKDK, HHS has a computational overhead due to the recursive splitting of the Hamiltonian needed to build the time-step hierarchy.

We decided to make a short exploration of the simulation parameters (accuracy, rung number, and minimum time-step) with our direct summation code. We present the results in Figure 6, which

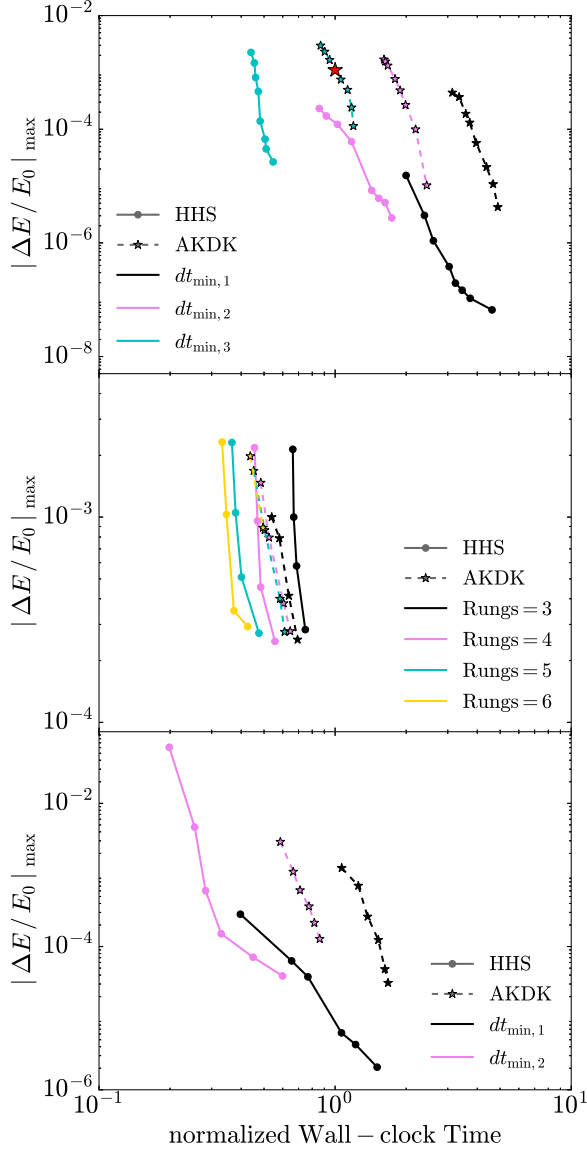


Fig. 6. Energy conservation as a function of wall-clock time (normalized with respect to the red star case in upper panel) for the fiducial isolated halo (upper and middle panels) and for a cuspy halo including four live cuspy satellites (bottom panel), for the HHS (solid circles) and AKDK (stars) integrators. In the upper panel, we used a Plummer softening  $\epsilon = 0.007$ , and we varied the minimum time step ( $dt_{\min,1} = 5.4 \times 10^{-3}$ ,  $dt_{\min,2} = 1.1 \times 10^{-2}$  and  $dt_{\min,3} = 2.2 \times 10^{-2}$ ) but fixing the number of rungs (6 and 5 for HHS and AKDK, respectively). In the middle panel, we used  $\epsilon = 0.01$ , and we varied the number of rungs, but fixed the minimum time step ( $dt_{\min} = 4.3 \times 10^{-2}$ ). For the bottom panel, we used  $\epsilon = 0.004$ , and we varied the minimum time step ( $dt_{\min,1} = 6.4 \times 10^{-3}$  and  $dt_{\min,2} = 1.3 \times 10^{-2}$ ) but fixing the number of rungs (6 and 3 for HHS and AKDK, respectively). The color figure can be viewed online

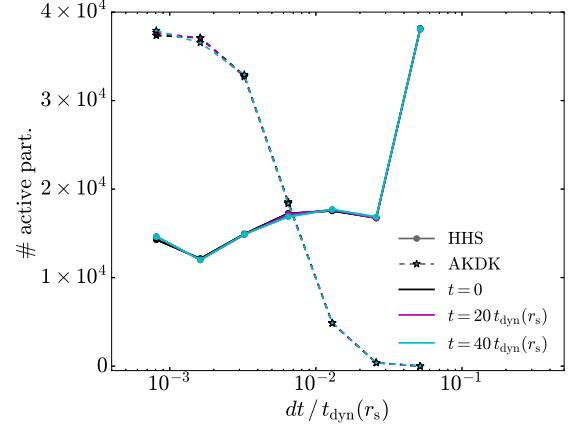


Fig. 7. Particle distribution across time-step rungs for different selection functions: acceleration criteria (equation 5, stars) and the approximated symmetric free-fall (equations 9 and 10, solid circles), for the fiducial isolated halo using the same parameters as in Figure 1. Note that the acceleration criterion has more particles with small time step and that the free-fall criterion is almost the opposite, with potential consequences for the performance. The color figure can be viewed online.

shows a pragmatic diagnostic of the integrator performance: energy conservation vs. wall-clock time. The upper and middle panels correspond to the fiducial isolated halo. In the upper panel we fixed the number of time-step rungs and we varied the minimum time-step; whereas in the middle panel, we fixed the minimum time-step (same for both integrators) and we varied the number of time-step rungs. The lower panel corresponds to the cuspy halo including four live cuspy satellites, a common situation in cosmological simulations that requires a large dynamical range. In many cases HHS outperforms AKDK. We may wonder what the reason is given the extra computations related with the splitting process. One possible suspect is the individual time-step distribution. To investigate that, we built the histogram of time steps for particles for the initial conditions and at later times (see Figure 7). For HHS (solid circles) it is clear that only a moderate fraction of particles are found in the deepest time-step rung. For AKDK the distribution is almost the opposite, there is a peak of particles in the three deepest rungs, which translates into many more time-steps than HHS. The difference in performance is partly due to the time-step selection function, in agreement with Zemp et al. (2007); Grudić & Hopkins (2020). Although tweaking parameters we may obtain differences in performance and accuracy, we noticed that for a defined energy conservation

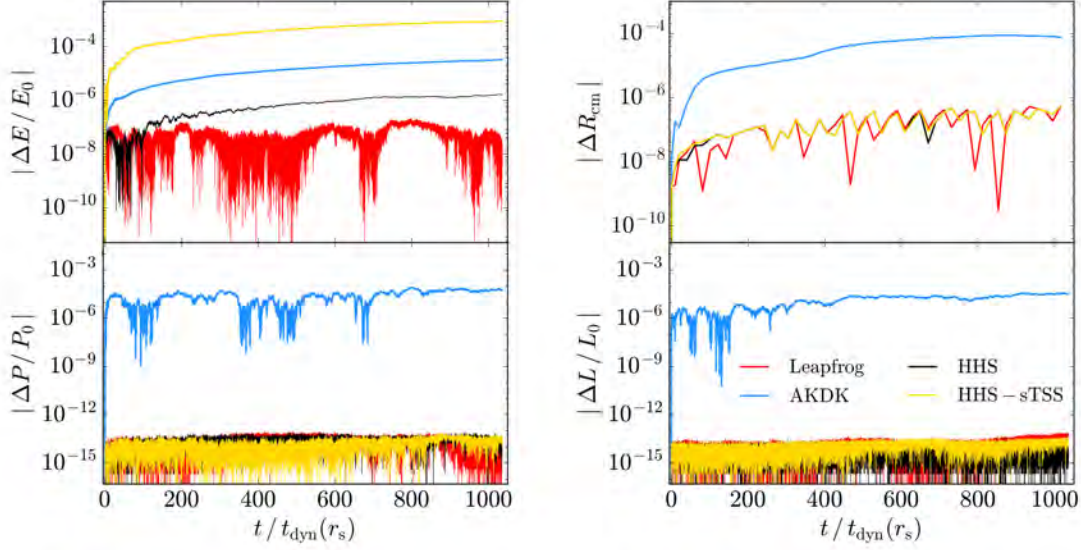


Fig. 8. As in Figure 1, but following the fiducial isolated halo up to  $10^3$  dynamical times (at scale radius). The color figure can be viewed online.

threshold in many cases HHS outperforms AKDK, because the energy growth is smaller for HHS than for the standard AKDK implementation, allowing HHS to use larger steps. Nevertheless, it is important to mention that if all the particles are similarly distributed in the time-step runs for both integrators the computational overhead of HHS starts to play a more important role. This may happen for very low resolution runs where the dynamical range is artificially shortened. This is in agreement with the middle panel of the figure, showing that only when using more than three step runs HHS is faster than AKDK. The time-step histogram presented in Figure 7 is a handy tool to assess the situation.

Our conclusions are consistent with recent studies that published successful results using HHS with GADGET4 and reaching an extremely large dynamical range with multi-million particle numbers (Wang et al. 2020).

## 7. LONG-TERM STABILITY

At this point we have compared the accuracy and performance of the integrators for some dynamical times. A natural question arises, whether the HHS advantages are relevant for realistic long-term integration. Dark matter halos survive around 30 – 200 dynamical times in cosmological simulations depending on the merger/accretion history (Klypin et al. 2015). As before, we investigated the stability of en-

ergy, linear and angular momentum and density centroid for our fiducial isolated halo model, this time for hundreds of dynamical times (see Figure 8). Energy conservation of HHS (black) is quite close to the global-constant time-step leapfrog (red) behavior during the first 20 – 40 dynamical times (consistent with our previous tests). However, after that time it starts to slowly drift which seems to get slower at the end of the simulation ( $\approx 700 t_{\text{dyn}}$ ). AKDK (blue) quickly drifts to a considerably larger energy error and keeps systematically growing. The yellow curve represents the HHS-sTSS version that delays the time-step hierarchy updating. As we observed before, it behaves as AKDK but with a smaller accuracy, although it is faster ( $\approx 40\%$ ). For the linear and angular momentum all integrators preserve them. However, while HHS preserves them almost at machine precision, AKDK preservation is almost 8 orders of magnitude worse. A similar disparity is obtained by following the halo centroid. Seeking a cause of this difference in accuracy we tracked the circular velocity in experiments with a smaller number of particles ( $N = 2000$ ) evolving the system for 400 dynamical times at  $2.1 r_s$ , where the circular velocity peaks. Figure 9 shows the circular velocity profile at several times. We found some differences, but they are all inside 10%. We conclude that, at the level of high accuracy, we do not expect important differences between integrators, and performance is the most relevant difference. As for as  $N$ -body simu-

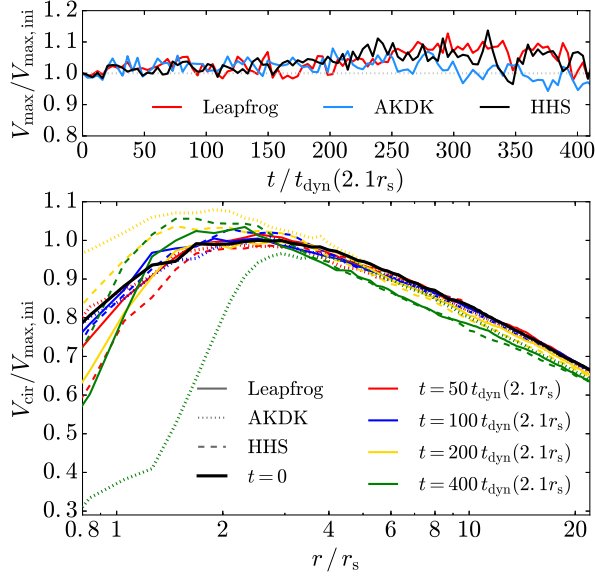


Fig. 9. Circular velocity curves for the fiducial isolated halo with 2000 particles following the system for long integration times. There is no systematic difference between different integrators. The energy conservation level is  $\approx 10^{-5}$ . The maximum circular velocity is scattered inside 10% at different moments. The color figure can be viewed online.

lations reaching a larger dynamical range, the smallest structures may live for a larger number of dynamical times; in this context HHS may offer a more stable option.

Recent studies regarding long-term  $N$ -body evolution (Hernandez et al. 2020), suggest that long-term integrations may be unstable to small changes in initial conditions realizations. This may be particularly critical for highly non-linear situations like the three-body problem. We generated a small ensemble of realizations for the fiducial isolated halo and for the sinking satellite problem. Results are shown in Figure 10 only for Leapfrog (red) and HHS (black). Indeed, some scatter is found; however, overall the results are robust.

## 8. REVERSIBILITY AND TIME SYMMETRY

The above numerical experiments show that there are certain parameter combinations where HHS is more accurate than AKDK or, alternatively, it is faster for a given energy accuracy. Because both integrators have different parameters it is natural to ask if there is a true advantage of HHS or if it is a misleading result, dependent on our implementation, accuracy or even particle number.

As we discussed in the introduction and in agreement with Hairer et al. (2003) we performed time symmetry and velocity reversibility tests using both HHS and AKDK integrators; we used leapfrog with a global-constant time-step as a reference case.

We chose the sinking satellite system as the test bed because the satellite orbit allows easily to track the system response in configuration space as well as in energy. At three different moments, termed BW1, BW2 and BW3, we reversed the velocity signs for all particles, and for one case instead of velocities we reversed the time sign; after that we continued the integration. Figure 11 shows the global result of forward (FW, solid lines) and backward (BW) evolution (i.e. inverting the sign of velocities) of the satellite distance to the halo center for all integrators, either for high (left panels) or low (right panels) energy conservation accuracy. For high energy conservation accuracy ( $\approx 10^{-8}$ , left panels), the differences in configuration space are in general small, which is consistent with Hernandez & Bertschinger (2018), where they discuss that AKDK preserves quantities like angular and linear momentum (see also Figure 3). However, there are still some differences as we can see in the lower panel. The satellite distance change with respect to its own forward evolution (lower panels) is a good diagnostic of reversibility. Clearly, the position difference in HHS (black) regarding the symplectic Leapfrog (red, global constant step) is below the simulation resolution determined by the softening (horizontal grey line), while for AKDK (blue) the calculated position difference is larger but close to the simulation softening. At a less accurate energy conservation but more common in collisionless simulations ( $10^{-3}$ , right panels), differences between forward and velocity reversal integration in Leapfrog (global constant step) and HHS are still below the simulation softening. However, AKDK has notable differences that are well above the simulation resolution and they are even detected in configuration space. We also tracked the fractional energy change (Figure 12). The first outstanding fact is the truly reversible behavior of the global-constant time-step leapfrog (upper panel); almost every peak and valley is reproduced after the velocity reversal. The middle panel shows the high accuracy tests for AKDK (blue lines) and HHS (black lines), and the bottom panel the corresponding low accuracy tests. For AKDK, there is a roughly systematic growth for both forward integration (blue solid opaque line) and also after reversing velocities, suggesting that AKDK is non-reversible. Instead, HHS is almost flat, suggesting that it is approximately reversible. We also ran a

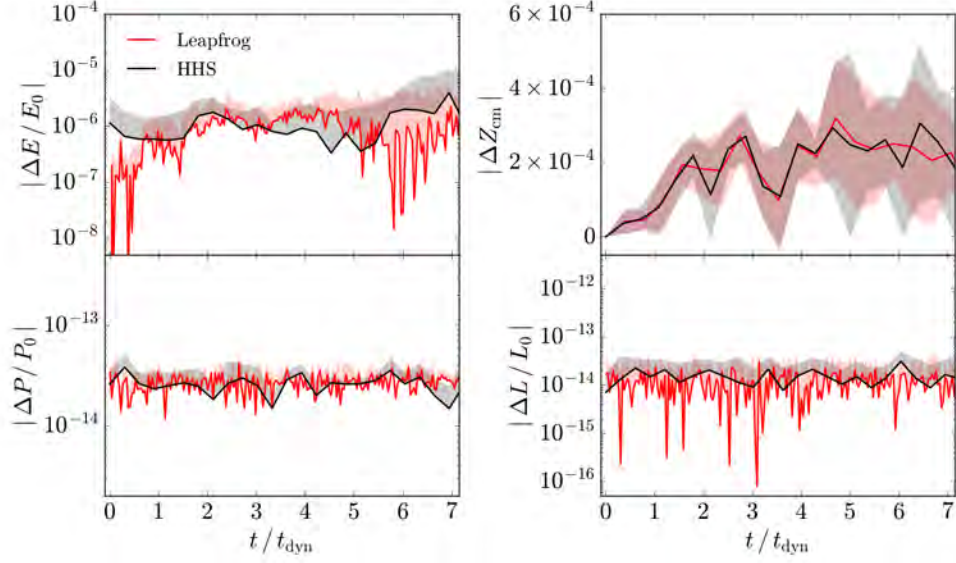


Fig. 10. Error in conserved quantities for a NFW halo with  $N = 10^5$  particles plus a satellite simulated up to 7 dynamical times. Shown is the energy error (upper left panel), the change in center of mass position (upper right panel), the error in linear (lower left panel) and angular momentum (lower right panel) for the Leapfrog (red) and HHS (black) integrators. Shaded portions represent one sigma standard deviation propagated from different realizations ran under different random seeds. The color figure can be viewed online.

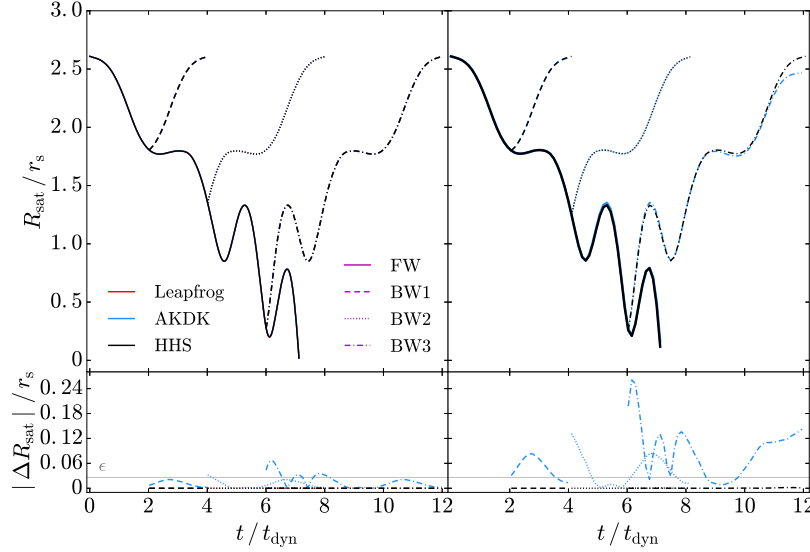


Fig. 11. Reversibility test. For the sinking satellite experiment we reversed the velocity sign at different moments, indicated by BW1 (dashed), BW2(dotted) and BW3 (dash-dot), and continued the integration. Solid lines represents the forward integration. Purple line labels only indicate the line-style corresponding to the start of each FW/BW integration. Upper panels show the normalized satellite distance as a function of time for the truly symplectic leapfrog (red lines), the adaptive one AKDK (blue lines) and HHS (black lines) integrators. Lower panels show the difference in position regarding the corresponding forward solution (solid lines). For high energy conservation accuracy ( $\approx 10^{-8}$ , left panels), there are some differences between integrators. However, they are inside (Leapfrog and HHS) or very close (AKDK) to the simulation softening (horizontal grey line). For a lower energy conservation accuracy ( $\approx 10^{-3}$ , right panels), still HHS and Leapfrog position differences are inside the simulation softening, whereas the AKDK case is above the simulation resolution. The color figure can be viewed online.

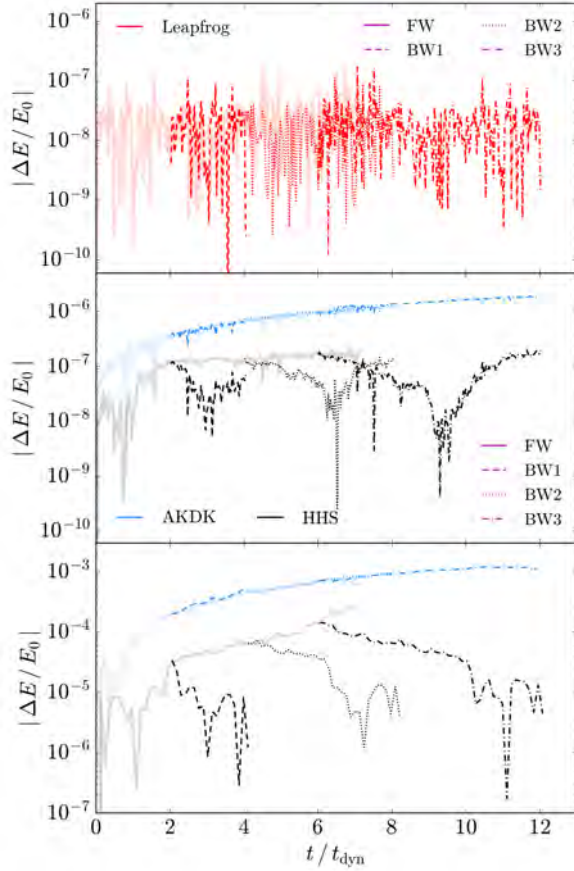


Fig. 12. Fractional energy error before (solid opaque lines) and after reversibility (dashed, dotted and dash-dot lines for BW1, BW2 and BW3 moments, respectively), for the same test of Figure 11. Top panel shows the global-constant time-step leapfrog, middle panel shows the higher accuracy in energy conservation tests, and the bottom panel the lower accuracy tests. Leapfrog is truly reversible. AKDK (blue lines) has a systematic growth of energy, suggesting that it is non-reversible. HHS (black lines) is almost flat with a small drift of energy after  $\approx 2 t_{\text{dyn}}$  ( $4.5 t_{\text{dyn}}$ ) for the high (low) accuracy test of the backward integrations, indicating that it is approximately reversible. Note that the purple line labels only indicate the line-style corresponding to the start of each FW/BW integration. The color figure can be viewed online.

test using HHS where we changed the time variable sign. In this case, the energy presents a systematic growth, showing that HHS is not time symmetric. However, for the sake of clarity, it was not included. We performed similar tests for the fiducial isolated halo and obtained similar results. The small slope showed in the energy drift suggests that HHS is approximately reversible. Furthermore the energy drift

is small compared to AKDK, which is non-reversible and non time-symmetric. This is in agreement with Hairer et al. (2002) who showed that the symplectic Störmer-Verlet-LeapFrog (AKDK) performs poorly for an adaptive time-step integrator, unless the time-step assigning function is properly selected with a possible computational cost.

Figure 5 shows that even GADGET4 with the selection function given by equation 5 and its implementation of HHS allows, under certain conditions, for a faster or more accurate integration than AKDK. Figure 13 investigates the reversibility of HHS and AKDK in the Tree (upper panel) and FMM (lower panel) implementations using GADGET4. The forward (FW) evolution of both integrators is shown with solid opaque lines, and the backward (BW) evolution after changing the velocity sign, at  $40 t_{\text{dyn}}$ , is shown with dashed lines. For the Tree gravity solver using HHS, the energy drift for both forward and backward evolution is similar, particularly during the first  $10 t_{\text{dyn}}$ . However, although encouraging, we do not have enough evidence to claim that HHS is reversible. The time-step function adopted by GADGET4 may be the reason it is slightly different to our tests with a direct summation code. On the other hand, the reversed integration of AKDK shows a discontinuous change in its slope, indicating that it is not reversible. We conclude that Tree-HHS is more stable than Tree-AKDK. For the FMM gravity solver the test is not conclusive, because it does not show a significantly smaller slope in energy drift using the HHS scheme (Hierarchical Gravity).

## 9. DISCUSSION AND CONCLUSIONS

Using a GPU direct summation  $N$ -body code, we tested and characterized the Hierarchical Hamiltonian Splitting (HHS) integrator proposed by Pelupessy et al. (2012), but we focused on collisionless simulations. As a reference we compared with the global-constant time-step symplectic Leapfrog integrator and the widely used Adaptive one (AKDK). We also complemented our study using the HHS implementation in GADGET4 (dubbed as hierarchical gravity, Springel et al. 2021), which uses a different time-step selection function and approximate force solvers (Tree and FMM).

As recently discussed (e.g. Dehnen 2017), there is no general solution for a symplectic adaptive multi time-step integrator, although there are several proposals (e.g. Huang & Leimkuhler 1997; Hairer 1997; Calvo et al. 1998; Hardy et al. 1999; Farr & Bertschinger 2007). The problem is not exclu-

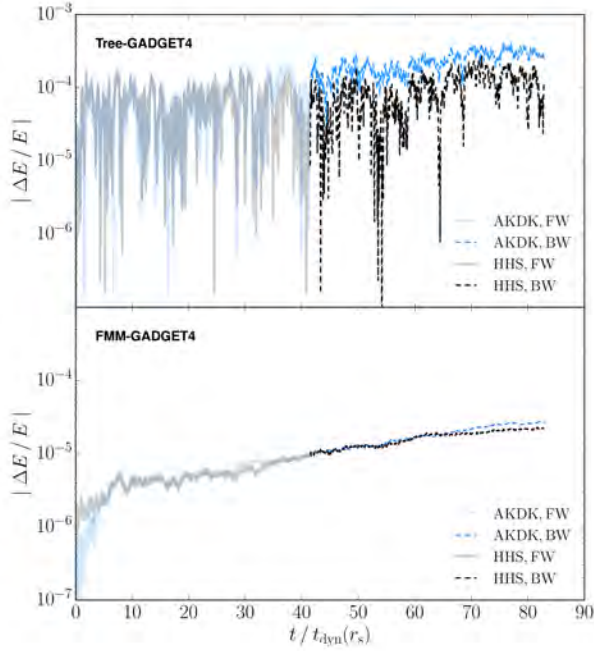


Fig. 13. Reversibility test for GADGET4. We followed the fiducial isolated halo using the GADGET4 code (shown with solid opaque lines) and we reversed the sign of particle velocities at around  $40 t_{\text{dyn}}$  (dashed lines). Upper panel (Tree-GADGET4): for HHS with a non-symmetrized step function, remarkably up to  $60 t_{\text{dyn}}$  the energy drift stays flat, then it slowly starts to grow. For AKDK the reversed integration just continues the forward systematic growth. Lower panel: For FMM with HHS and AKDK using a non-symmetrized step function, the energy drift grows systematically for both integrators. We can see that Tree-GADGET4 with HHS and the non-symmetrized step function could be approximately reversible or at least more stable. However, more studies are required. The color figure can be viewed online.

sive of astrophysics; the field of differential equations of dynamical systems has extensively reviewed the subject (Hairer et al. 2002). In particular, it is important to say that adaptive and multi-step techniques are not identical but they are related. Hernandez & Bertschinger (2018) address the important case of adaptive time steps, inspired by Hairer et al. (2002). They discuss what is called the “backwards error analysis” in order to explore commonly used integrators in astrophysics, concluding that symplecticity, time symmetry and reversibility are not the same and they are not always a guarantee of energy conservation. Nevertheless, the KAM (Kolmogorov-Arnold-Moser) theorem assures stability for symplectic (geometrical) and reversible in-

tegrators (Hairer et al. 2002) in contrast to non-reversible ones, like AKDK with the standard time-step selection function. The case of multi-step integration, requires extra conditions like the so called impulse (splitting) or averaging (less often force evaluation) techniques (Hairer et al. 2002). Similar problems have been discussed in the molecular dynamics field, and the r-RESPA method (Tuckerman et al. 1992) that is a splitting technique has been applied to the long-short range splitting case or multiple mass segregation with considerable advantage in performance. An important feature is that such approaches to the multi-step problem resulted in a symmetric or reversible method, not in a symplectic one. The major concrete benefit has been an improvement in the performance at a given energy accuracy. A similar situation applies to HHS, the case we study in this paper. HHS is a composition of *Fast* and *Slow* operators (Pelupessy et al. 2012) and based in our numerical experiments it is approximately reversible, which explains the performance advantage shown in § 6. The discussion in Hairer et al. (2002) and Hernandez & Bertschinger (2018) shows that the reversibility depends on the symmetry under a velocity sign change of hamiltonian and the corresponding time-step assigning function, and not on the accuracy of gravity calculation or particle number. The particular case of equation 10 is an approximate 1st order version of the one proposed by Hut et al. (1995) that depends only on the module of particle pairs relative velocities; therefore it is reversible. A future interesting avenue inspired by the r-RESPA method presented by Tuckerman et al. (1992) is the extension already discussed by Portegies Zwart et al. (2020). We experimented with the strategy motivated by the multi-step averaging technique (Hairer et al. 2002), updating the time-step hierarchy only at a given number of global time-steps (dubbed as HHS-sTSS, Figures 1 and 8). There is a gain in CPU time; however, the energy drift increases suggesting that further investigation is required.

Our results are summarized below:

1. Based on reversibility and time symmetry tests we concluded that HHS is not time symmetric but it is approximately reversible; it is also more stable than AKDK for a given energy accuracy. Although the exact correspondence between forward and backwards integration lasts only for few dynamical times, even for ten dynamical times the energy drift growth is small. In contrast, the AKDK energy drift grows systematically with time, clearly showing that it is non-reversible using the commonly used time-

step selection function. Based in our tests HHS reversibility explains the advantage in performance regarding AKDK. Based on the Hairer et al. (2002) discussion and the used time-step selection functions that depend only on the velocity module, such properties are independent of accuracy in gravity calculation and of particle number.

2. Our findings with the direct summation code may be also relevant for the high accuracy and costly section (PP) of codes using the P3M technique, as it has been also shown in molecular dynamics studies (Plimpton et al. 1997).
3. In agreement with our direct summation code tests, changing the time-step selection function for a non-symmetrized one, it is possible to find a combination of parameters in GADGET4 (using both the Tree and FMM code version) where HHS is more efficient than AKDK. We found approximate reversibility for the Tree Gadget4 test. However, for FMM Gadget4 our tests were inconclusive.
4. The population of particle histogram across the time-step rungs is useful to find a convenient parameter combination for the integrators.

GAA thanks V. Springel for nicely clarifying GADGET4 information. GAA and JCC thank Leobardo Ithehua Rico for the fruitful technical support and discussion. OV and GAA acknowledge support from UNAM PAPIIT Grant IN112518, IG101620 and IG101222. GAA and JAT thank support from CONACyT graduate fellowships 455336 and 825451, respectively. HV acknowledges support from IN101918 PAPIIT-UNAM Grant and from a Megaproyecto DGTIC-LANCAD for access to the Supercomputer MIZTLI. This research was partially supported through computational and human resources provided by the LAMOD UNAM project through the clusters Atocatl and Tochtli. LAMOD is a collaborative effort between the IA, ICN and IQ institutes at UNAM, and it acknowledges benefit from the MX28274 agreement with IBM. We thank CEDIA computing center for access to the DGX A100 cluster through an agreement with CUDI.

#### REFERENCES

- Aarseth, S. J. 1971, *Ap&SS*, 14, 20, <https://doi.org/10.1007/BF00649191>
- Angulo, R. E. & Hahn, O. 2021, arXiv e-prints, arXiv:2112.05165, <https://doi.org/10.48550/arxiv.2112.05165>
- Arca-Sedda, M. & Capuzzo-Dolcetta, R. 2016, *MNRAS*, 461, 4335, <https://doi.org/10.1093/mnras/stw1647>
- Bagla, J. S. 2002, *JApA*, 23, 185, <https://doi.org/10.1007/BF02702282>
- Barnes, J. & Hut, P. 1986, *Natur*, 324, 446, <https://doi.org/10.1038/324446a0>
- Bode, P. & Ostriker, J. P. 2003, *ApJS*, 145, 1, <https://doi.org/10.1086/345538>
- Bode, P., Ostriker, J. P., & Xu, G. 2000, *ApJS*, 128, 561, <https://doi.org/10.1086/313398>
- Calvo, M., López-Marcos, M., & Sanz-Serna, J. 1998, *Applied Numerical Mathematics*, 28, 1
- Channell, P. J. 1993, in *Proceedings of Symplectic Integrators Workshop*, Waterloo Ontario, Canada
- Cheng, S., Yu, H.-R., Inman, D., et al. 2020, in *Accepted for SCALE 2020*, colocated as part of the proceedings of CCGRID 2020; 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), <https://doi.org/10.1109/CCGrid49817.2020.00-22>, arXiv: 2003.03931
- Couchman, H. M. P. 1991, *ApJ*, 368, 23, <https://doi.org/10.1086/185939>
- Dehnen, W. 2000, *ApJ*, 536, 39, <https://doi.org/10.1086/312724>
- . 2002, *JCoPh*, 179, 27, <https://doi.org/10.1006/jcph.2002.7026>
- . 2017, *MNRAS*, 472, 1226, <https://doi.org/10.1093/mnras/stx1944>
- Dehnen, W. & Read, J. I. 2011, *EPJP*, 126, 55, <https://doi.org/10.1140/epjp/i2011-11055-3>
- Efstathiou, G., Davis, M., White, S. D. M., & Frenk, C. S. 1985, *ApJS*, 57, 241, <https://doi.org/10.1086/191003>
- Farr, W. M. & Bertschinger, E. 2007, *ApJ*, 663, 1420, <https://doi.org/10.1086/518641>
- Grudić, M. Y. & Hopkins, P. F. 2020, *MNRAS*, 495, 4306, <https://doi.org/10.1093/mnras/staa1453>
- Habib, S., Morozov, V., Frontera, N., et al. 2013, *SC'13 Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis*, 6, <https://doi.org/10.1145/2503210.2504566>
- Habib, S., Pope, A., Finkel, H., et al. 2016, *New A*, 42, 49, <https://doi.org/10.1016/j.newast.2015.06.003>
- Hairer, E. 1997, *ApNM*, 25, 219, [https://doi.org/10.1016/s0168-9274\(97\)00061-5](https://doi.org/10.1016/s0168-9274(97)00061-5)
- Hairer, E., Lubich, C., & Wanner, G. 2002, *Springer Series in Computational Mathematics*, 12, 399
- . 2003, *AcNum*, 12, 399, <https://doi.org/10.1017/s0962492902000144>
- Hardy, D. J., Okunbor, D. I., & Skeel, R. D. 1999, *ApNM*, 29, 19, [https://doi.org/10.1016/s0168-9274\(98\)00031-2](https://doi.org/10.1016/s0168-9274(98)00031-2)
- Hayli, A. 1967, *NASSP* 153, *The N-Body Gravitational Problem and the Simulation of Galactic Clusters*, 315
- Hernandez, D. M. & Bertschinger, E. 2018, *MNRAS*, 475, 5570, <https://doi.org/10.1093/mnras/sty184>

- Hernandez, D. M., Hadden, S., & Makino, J. 2020, *MNRAS*, 493, 1913, <https://doi.org/10.1093/mnras/staa388>
- Hernquist, L. & Katz, N. 1989, *ApJS*, 70, 419, <https://doi.org/10.1086/191344>
- Hockney, R. W. & Eastwood, J. W. 1998, *ISC Workshops*
- Huang, W. & Leimkuhler, B. 1997, *SIAM Journal on Scientific Computing*, 18, 239, <https://doi.org/10.1137/s1064827595284658>
- Hut, P. & McMillan, S. L. W. 1986, *The Use of Supercomputers in Stellar Dynamics*, (New York, NY: Springer), 267, <https://doi.org/10.1007/BFb0116387>
- Hut, P., Makino, J., & McMillan, S. 1995, *ApJ*, 443, 93, <https://doi.org/10.1086/187844>
- Jessop, C., Duncan, M., & Chau, W. Y. 1994, *JCoPh*, 115, 339, <https://doi.org/10.1006/jcph.1994.1200>
- Klypin, A. 2018, *Cosmological N-Body Simulations*, (World Scientific Series in Astrophysics), 27
- Klypin, A., Prada, F., Yepes, G., Heß, S., & Gottlöber, S. 2015, *MNRAS*, 447, 3693, <https://doi.org/10.1093/mnras/stu2685>
- Kravtsov, A. V., Klypin, A. A., & Khokhlov, A. M. 1997, *ApJS*, 111, 73, <https://doi.org/10.1086/313015>
- Makino, J. 1991, *ApJ*, 369, 200, <https://doi.org/10.1086/169751>
- Miller, T. B., van den Bosch, F. C., Green, S. B., & Ogiya, G. 2020, *MNRAS*, 495, 4496, <https://doi.org/10.1093/mnras/staa1450>
- Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, *ApJ*, 490, 493, <https://doi.org/10.1086/304888>
- Oshino, S., Funato, Y., & Makino, J. 2011, *PASJ*, 63, 881, <https://doi.org/10.1093/pasj/63.4.881>
- Pelupessy, F. I., Jänes, J., & Portegies Zwart, S. 2012, *New A*, 17, 711, <https://doi.org/10.1016/j.newast.2012.05.009>
- Plimpton, S. J., Pollock, R., & Stevens, M. J. 1997, *PPSC*
- Portegies Zwart, S., Pelupessy, I., Martínez-Barbosa, C., van Elteren, A., & McMillan, S. 2020, *CNSNS*, 85, 105240, <https://doi.org/10.1016/j.cnsns.2020.105240>
- Quinn, T., Katz, N., Stadel, J., & Lake, G. 1997, *arXiv: astro-ph/9710043*
- Sellwood, J. A. 1985, *MNRAS*, 217, 127, <https://doi.org/10.1093/mnras/217.1.127>
- Springel, V. 2005, *MNRAS*, 364, 1105, <https://doi.org/10.1111/j.1365-2966.2005.09655.x>
- Springel, V., Pakmor, R., Zier, O., & Reinecke, M. 2021, *MNRAS*, 506, 2871, <https://doi.org/10.1093/mnras/stab1855>
- Springel, V., Yoshida, N., & White, S. D. M. 2001, *New A*, 6, 79, [https://doi.org/10.1016/s1384-1076\(01\)00042-2](https://doi.org/10.1016/s1384-1076(01)00042-2)
- Stadel, J. G. 2001, *Cosmological N-body simulations and their analysis*, PhD Thesis, University of Washington
- Swope, W. C., Andersen, H. C., Berens, P. H., & Wilson, K. R. 1982, *JChPh*, 76, 637, <https://doi.org/10.1063/1.442716>
- Teyssier, R. 2002, *A&A*, 385, 337, <https://doi.org/10.1051/0004-6361:20011817>
- Tuckerman, M., Berne, B. J., & Martyna, G. J. 1992, *JChPh*, 97, 1990, <https://doi.org/10.1063/1.463137>
- Tuckerman, M. E., Martyna, G. J., & Berne, B. J. 1990, *JChPh*, 93, 1287, <https://doi.org/10.1063/1.459140>
- Verlet, L. 1967, *PhRv*, 159, 98, <https://doi.org/10.1103/PhysRev.159.98>
- Villumsen, J. V. 1989, *ApJS*, 71, 407, <https://doi.org/10.1086/191380>
- Wang, J., Bose, S., Frenk, C. S., et al. 2020, *Natur*, 585, 39, <https://doi.org/10.1038/s41586-020-2642-9>
- Weinberger, R., Springel, V., & Pakmor, R. 2020, *ApJS*, 248, 32, <https://doi.org/10.3847/1538-4365/ab908c>
- Xu, G. 1995, *ApJS*, 98, 355, <https://doi.org/10.1086/192166>
- Zemp, M., Stadel, J., Moore, B., & Carollo, C. M. 2007, *MNRAS*, 376, 273, <https://doi.org/10.1111/j.1365-2966.2007.11427.x>
- Zhu, Q. 2017, *arXiv: 1712.10116*, <https://doi.org/10.48550/arxiv.1712.10116>