

Reversible time-step adaptation for the integration of few-body systems

Tjarda C. N. Boekholt^{1*}, Timothée Vaillant² and Alexandre C. M. Correia^{2,3}

¹*Rudolf Peierls Centre for Theoretical Physics, Clarendon Laboratory, Parks Road, Oxford, OX1 3PU, United Kingdom*

²*CFisUC, Departamento de Física, Universidade de Coimbra, 3004-516 Coimbra, Portugal*

³*IMCCE, UMR8028 CNRS, Observatoire de Paris, PSL Université, 77 Av. Denfert-Rochereau, 75014 Paris, France*

19 December 2022

ABSTRACT

The time step criterion plays a crucial role in direct N -body codes. If not chosen carefully, it will cause a secular drift in the energy error. Shared, adaptive time step criteria commonly adopt the minimum pairwise time step, which suffers from discontinuities in the time evolution of the time step. This has a large impact on the functioning of time step symmetrisation algorithms. We provide new demonstrations of previous findings that a smooth and weighted average over all pairwise time steps in the N -body system, improves the level of energy conservation. Furthermore, we compare the performance of 27 different time step criteria, by considering 3 methods for weighting time steps and 9 symmetrisation methods. We present performance tests for strongly chaotic few-body systems, including unstable triples, giant planets in a resonant chain, and the current Solar System. We find that the harmonic symmetrisation methods (methods A3 and B3 in our notation) are the most robust, in the sense that the symmetrised time step remains close to the time step function. Furthermore, based on our Solar System experiment, we find that our new weighting method based on direct pairwise averaging (method W2 in our notation), is slightly preferred over the other methods.

Key words: methods: numerical

1 INTRODUCTION

N -body codes consist of two main ingredients: the integration method and the time step criterion. Our main focus is to implement and compare various time step criteria, and to measure their effectiveness in conserving energy for a wide range of initial conditions.

The simplest time step criterion is the constant one. This criterion is commonly implemented in N -body codes for planetary systems, where orbital periods are not expected to change much. Usually, the time step is chosen as some fraction of the shortest, initial, orbital period in the system (e.g. Quinn et al. 1991; Jones et al. 2001; Ito & Tanikawa 2002; Bolmont et al. 2015).

If instabilities grow during the evolution, potentially resulting in planet-planet scattering (Chambers et al. 1996), then the time step size has to be reduced in order to resolve the dynamics accurately (Chambers 1999; Petit et al. 2019; Rein et al. 2019). Varying the time step is very common in N -body codes for dense stellar systems. Here, stellar orbits are chaotic, and close binaries are formed and destroyed continuously (e.g. Heggie 1975). It is then much preferred to implement a time step adaptation method. There are various ways to do this, including: 1) assign each pair of bodies in

the current configuration, a time step according to some criterion, and then adopt the minimal value, or 2) evaluate each body individually, and determine on what time scale its orbit is changing, and then adopt the minimal value over all bodies. Commonly used examples of the first case are the pairwise free-fall and flyby time scales (e.g. Hut et al. 2007; Pelupessy et al. 2012), and for the second case we mention the criterion by Sverre Aarseth based on the acceleration and its derivatives (Aarseth 2003).

Shared time step schemes, where each body adopts the same time step size, are simple to implement, but too expensive to be used for large- N systems (Makino 1991; Trenti & Hut 2008). For example, a star in the cluster’s halo should not have to suffer the same small time step as two stars in a close encounter in the core. State of the art N -body codes for large stellar systems therefore employ an individual time step criterion (e.g. Wang et al. 2020). Synchronisation and predictions of positions and velocities play an important role, which is greatly facilitated by using block time steps (e.g. Makino et al. 2006).

If the N -body system under study allows for a constant time step criterion, then this would be greatly preferred due to its simplicity, numerical efficiency and its compatibility with symplectic integrators. If an adaptive scheme is required, this not only comes with an extra cost of evaluating the time step, but the favourable symplectic properties also tend to be lost (e.g. Dehnen 2017).

* E-mail: tjarda.boekholt@physics.ox.ac.uk

Much research has been put into making adaptive time step schemes compatible with symplectic integration methods. For an extensive review, we refer the reader to [Dehnen \(2017\)](#). The main idea however, is obtained by considering an analogy to Noether’s theorem on time invariance and energy conservation. If the level of time-reversibility of an N -body integration is increased, then correspondingly, it might be expected that the level of energy conservation increases. This idea led to the concept of time step symmetrisation methods ([Quinlan & Tremaine 1990](#); [Hut et al. 1995](#); [Dehnen 2017](#)). As a concrete example, we consider a body on an eccentric orbit somewhere between pericenter and apocenter. Based on its current position and velocity, we would calculate a time step size. This step size would be the same irrespective of whether the body was approaching pericenter or apocenter, e.g. the direction of the orbit. If we consider having taken an integration step towards pericenter, then the newly evaluated time step size will be shorter, as we are closer to pericenter. However, if we would take this new time step, but integrate back along the orbit, we would not end up on our initial condition. In other words, adaptive time step functions based on the current snapshot only, do not result in time-symmetric integrations. As a consequence, the lack of error cancellation results in a secular drift in the energy error. In order to achieve a time-symmetric integration, we require a symmetrised time step, h , which depends symmetrically on the old and new states.

However, as discussed by [Hands et al. \(2019\)](#), an adaptive and symmetrised time step is generally not sufficient. Consider a closest pair of bodies with a certain time step, T_0 . We can imagine this closest pair to be receding from each other, such that T_0 increases. At the same time, there is another pair whose time step is $T_1 > T_0$. However, this pair is approaching, and therefore T_1 is decreasing. There will be a transition point at which $T_0 = T_1$, but each pair will have a different value for the derivative. In short, if there is a transition to a new pair of bodies with the smallest time step, a discontinuity can arise in the time step function and its derivative. The discontinuities arise from adopting the minimal time step over all pairs of bodies. Instead, [Hands et al. \(2019\)](#) propose to implement a weighted average, such that small time step values have a larger weight. Such a scheme makes use of all the orbital frequencies in the N -body system instead of just one.

Our aim is to expand on the work by [Hands et al. \(2019\)](#), and to test for the positive effect of smoothness for a variety of time step criteria and N -body systems. We implement three different methods for weighting time steps, including the algorithm by [Hands et al. \(2019\)](#) and two of our own. We will also compare nine different time step symmetrisation methods, most of them based on previous literature, but complemented with three new ones. This results in a total of 27 different combinations. Furthermore, we will apply these time stepping methods to different types of astrophysical systems. This allows us to determine if there is a single, optimal time step criterion for chaotic few-body systems.

2 METHODS

We start by describing two fourth-order integration methods adopted for our simulations. Then we present the three ingredients for the time step criteria: 1) a fixed pairwise time step function, 2) three weighting methods for calculating a smooth and “global” time step, and 3) nine different time step symmetrisation methods. An overview of these ingredients is provided in Tab. 1.

2.1 Fourth-order integrators

As the benchmark integrator, we will adopt the fourth-order scheme by [McLachlan \(1995\)](#) denoted by (order 4, S, $m = 5$). This method is time-reversible and symplectic, when combined with a constant time step. The integration map consists of a symmetric sequence of kicks and drifts with a total of 5 calculations of the acceleration, as follows:

$$\phi_{\text{MCL4}}(h) = K(a_1 h) D(b_1 h) K(a_2 h) D(b_2 h) K(a_3 h) D(b_3 h) K(a_3 h) D(b_2 h) K(a_2 h) D(b_1 h) K(a_1 h). \quad (1)$$

Here h is the time step, K and D refer to the usual Kick and Drift operators (as used in the Verlet-Leapfrog integrator), and the coefficients are given by $a_1 = (14 - \sqrt{19})/108$, $a_2 = (20 - 7\sqrt{19})/108$, $a_3 = (1 - 2a_1 - 2a_2)/2$, $b_1 = 2/5$, $b_2 = -1/10$, and $b_3 = 1 - 2b_1 - 2b_2$. This sequence also includes negative time steps, but [McLachlan \(1995\)](#) demonstrates that this integrator is among the most effective fourth-order integrators constructed by a composition of kicks and drifts.

In order to test for the generality of a potential optimal time step criterion, we implement a second integrator. We adopt the method by [Chin & Chen \(2005\)](#) (following a suggestion by Walter Dehnen), which is also a fourth-order method, but which only includes positive time steps. Although this requires the calculation of the force gradient, the acceleration only needs to be calculated twice per step (together with an initial calculation at the start of the simulation). The integration map is given by:

$$\phi_{\text{CC4}}(h) = K(u_1 h) D(u_2 h) \tilde{K}(u_3 h) D(u_2 h) K(u_1 h). \quad (2)$$

The operator \tilde{K} represents the kick with the adjusted acceleration taking into account the force gradient ([Chin & Chen 2005](#); [Dehnen & Hernandez 2017](#)). The coefficients are given by $u_1 = 1/6$ and $u_2 = 1/2$ and $u_3 = 2/3$. We will abbreviate the names of the two integrators as MCL4 ([McLachlan 1995](#)) and CC4 ([Chin & Chen 2005](#)).

2.2 Time step function

For the time step function, T , we adopt the commonly used combination of the pairwise free-fall time scale and the pairwise flyby time scale (e.g. [Pelupessy et al. 2012](#); [Hands et al. 2019](#)), respectively given by:

$$T_{ij,0}^{-4} = \frac{\mu_{ij}^2}{r_{ij}^6}, \quad (3)$$

and

$$T_{ij,1}^{-2} = \frac{v_{ij}^2}{r_{ij}^2}. \quad (4)$$

Here, $r_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the separation between bodies i and j , $v_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity, and $\mu_{ij} = G(m_i + m_j)$ is the gravitational parameter. The free-fall time scale can also be interpreted as the separation divided by the circular velocity corresponding to that separation. The flyby time scale is required when the relative speeds much exceed this circular velocity.

The derivatives of the time step functions are required for some time step symmetrisation methods. They are given by

$$T_{ij,0}^{-1} \dot{T}_{ij,0} = \frac{3}{2} \frac{\mathbf{r}_{ij} \cdot \mathbf{v}_{ij}}{r_{ij}^2}, \quad (5)$$

$$T_{ij,1}^{-1} \dot{T}_{ij,1} = \frac{\mathbf{r}_{ij} \cdot \mathbf{v}_{ij}}{r_{ij}^2} - \frac{\mathbf{v}_{ij} \cdot \mathbf{a}_{ij}}{v_{ij}^2}, \quad (6)$$

The derivative of the flyby time scale includes the relative acceleration between a pair of bodies. We wish to emphasise here that this should be the difference in the total acceleration of the two bodies, i.e. $\mathbf{a}_{ij} = \mathbf{a}_i - \mathbf{a}_j$. If on the other hand, the approximation is made of only evaluating their mutual acceleration, which depending on the integration method is more convenient to calculate, then the derivative of the time step is not guaranteed to be accurate. We find that this can cause time step symmetrisation methods to fail, resulting in excessively small or large time steps.

Other time step functions could be defined, for example taking into account the accelerations. However, since some symmetrisation methods require the derivative of the time step, this would imply having to calculate the jerk (and potentially higher order derivatives), which leads to an increase of the computation time. We leave the performance comparisons with such time step criteria for elsewhere.

A common method then for calculating the next integration time step is to evaluate the free-fall and flyby time scale for all pairs of bodies in the system, and then to adopt the minimal value. This ensures the highest frequency encounters are resolved. However, this also results in discontinuities in the evolution of the time step function. An improved method is to consider a smooth combination of all pairwise time steps by introducing weights.

2.3 Weighting time steps

A general method for combining multiple time steps into a single value is the following:

$$T = \left(\frac{\sum_k T_k^{m-n}}{\sum_k T_k^{-n}} \right)^{\frac{1}{m}}, \quad (7)$$

with $m > 0$, $n > 0$ and the summation is over all the time step values under consideration (pairwise or per body as we will discuss later in this section). For example, we can set $m = 1$ resulting in

$$T = \frac{\sum_k T_k^{-n} T_k}{\sum_k T_k^{-n}} \equiv \frac{\sum_k w_k T_k}{\sum_k w_k}, \quad (8)$$

where we defined the time step weight $w_k = T_k^{-n}$. It is crucial for this weighted time step to remain close to the minimal pairwise time step in order to resolve close encounters. Another combination is ($m = 2$, $n = 10$) (Hands et al. 2019), which gives

$$T = \left(\frac{\sum_k T_k^{-8}}{\sum_k T_k^{-10}} \right)^{\frac{1}{2}}. \quad (9)$$

Alternatively, for $m = n$ we obtain

$$T = \left(\frac{\sum_k T_k^{-n}}{N_T} \right)^{-\frac{1}{n}}, \quad (10)$$

with N_T the number of time steps to be summed. As described by

Hands et al. (2019), this combination has a problematic feature. In the limit that all time steps are similar, i.e. $T_k \approx \tau$, we obtain

$$T \approx \left(\frac{N_T \tau^{-n}}{N_T} \right)^{-\frac{1}{n}} = \tau. \quad (11)$$

However, if there is a single time step which is by far the smallest, T_{\min} , then we obtain

$$T \approx \left(\frac{T_{\min}^{-n}}{N_T} \right)^{-\frac{1}{n}} = T_{\min} N_T^{\frac{1}{n}}. \quad (12)$$

Hence, for large- N systems, the time step T becomes much larger than T_{\min} , resulting in large discretisation errors. A slightly improved method is to multiply T by $N_T^{-\frac{1}{n}}$ in Eq. (10), resulting in

$$T = \left(\sum_k T_k^{-n} \right)^{-\frac{1}{n}}. \quad (13)$$

In the regime of a single smallest time step, this reduces to

$$T \approx (T_{\min}^{-n})^{-\frac{1}{n}} = T_{\min}, \quad (14)$$

while for the similar time steps case we obtain

$$T \approx (N_T \tau^{-n})^{-\frac{1}{n}} = \tau N_T^{-\frac{1}{n}}. \quad (15)$$

Hence, in this case the time step T becomes increasingly smaller than necessary for larger N systems. We note that these dependencies on N_T cancel out for combinations where $m \neq n$ in Eq. (7).

On the other hand, in the case where $m \neq n$, problems might arise with keeping the smoothed time step close to the minimum time step, for an increasing number of bodies. This can be more clearly seen by rewriting Eq. (8) as follows:

$$\frac{T}{T_{\min}} = \frac{\sum_k T_k^{1-n}}{\sum_k T_k^{-n}} = \frac{1 + \sum_{k \neq l} \left(\frac{T_k}{T_{\min}} \right)^{1-n}}{1 + \sum_{k \neq l} \left(\frac{T_k}{T_{\min}} \right)^{-n}} < f_T, \quad (16)$$

where we divided by the minimum pairwise time step, took the minimum time step outside the summation, and where $T_l = T_{\min}$. Here, f_T gives the maximum value of the fraction T/T_{\min} . We consider a certain number of bodies $N = N_0$ and a weight parameter $n = n_0$, such that Eq. (16) holds for a given value of f_T . If we now increase the number of bodies to $N = N_1 > N_0$, while assuming that the distribution of time steps is fixed, then this amounts to multiplying the summations by a factor of about $c = \left(\frac{N_1}{N_0} \right)^\alpha$, with $\alpha = 1$ or 2 depending on whether the sum is over all bodies or over all pairs. This might result in a violation of the constraint in Eq. (16). The increasing number of terms in the summation is to be compensated for by adjusting the weight parameter n from n_0 to n_1 . Considering a single term of the summation in the denominator, we require

$$\left(\frac{T_k}{T_{\min}} \right)^{-n_0} \approx \left(\frac{N_1}{N_0} \right)^\alpha \left(\frac{T_k}{T_{\min}} \right)^{-n_1}, \quad (17)$$

which can be rewritten as

$$n_1 - n_0 \propto \log \left(\frac{N_1}{N_0} \right). \quad (18)$$

Hence, in order to keep the smoothed time step close to the minimum time step, we expect the weight parameter to scale with the number of bodies as $n \propto \log N$. We confirm this result in Appendix B, and there we also demonstrate that for a particle number up to a few thousand, a weighting parameter of $n = 10$ is sufficient, consistent with the value adopted by Hands et al. (2019). We will adopt this value throughout our simulations, while also setting $m = 2$. Hence, we adopt the weighting method given by Eq. (9).

There are different ways to implement the summation, which can be over different types of time steps. In our first method, we will sum directly over all pairs of bodies, and add the free-fall and flyby time scales individually:

$$T = \eta \left(\frac{\sum_{i < j} T_{ij,0}^{-8} + T_{ij,1}^{-8}}{\sum_{i < j} T_{ij,0}^{-10} + T_{ij,1}^{-10}} \right)^{\frac{1}{2}} \quad (\text{method W1}). \quad (19)$$

Here we multiply the global time step, T , by the time step parameter, η , which allows the magnitude of the averaged time step size to be varied systematically. Alternatively, as proposed by Hands et al. (2019), we can first average the pairwise free-fall and flyby time scale according to

$$T_{ij}^{-q} \equiv T_{ij,0}^{-q} + T_{ij,1}^{-q}. \quad (20)$$

Following Hands et al. (2019), we adopt $q = 4$, which gives a good balance between the amount of smoothing and computational efficiency (note that the time step functions were introduced in powers of 2 and 4). Our second method for weighting time steps is then given by:

$$T = \eta \left(\frac{\sum_{i < j} T_{ij}^{-8}}{\sum_{i < j} T_{ij}^{-10}} \right)^{\frac{1}{2}} \quad (\text{method W2}). \quad (21)$$

with T_{ij} given by Eq. (20) with $q = 4$. Our third weighting method follows the method by Hands et al. (2019), which introduces another level of averaging. After combining the free-fall and flyby time scales as in our second method, we first calculate averaged time steps per body according to Eq. (13):

$$T_i^{-n} \equiv \sum_{j \neq i} T_{ij}^{-n}. \quad (22)$$

Although the value of n can be different than the value of q in Eq. (20), we follow the prescription by Hands et al. (2019) by setting $q = n = 4$. Subsequently, the “per body” time steps are averaged similar to the other methods:

$$T = \eta \left(\frac{\sum_i T_i^{-8}}{\sum_i T_i^{-10}} \right)^{\frac{1}{2}} \quad (\text{method W3}). \quad (23)$$

Note that here the summation is over all bodies. We will abbreviate the three weighting methods as W1, W2 and W3, as noted behind their respective equations. In our experiments, we will directly compare the performance of these three methods.

For completeness, we give the derivative of the generic expression for the summation of time steps (Eq. (7)):

$$\dot{T} = \frac{\eta}{m} \left(\frac{\sum_k T_k^{m-n}}{\sum_k T_k^{-n}} \right)^{\frac{1}{m}-1} \left((m-n) \frac{\sum_k T_k^{m-n-1} \dot{T}_k}{\sum_k T_k^{-n}} + n \frac{\sum_k T_k^{m-n} \sum_k T_k^{-n-1} \dot{T}_k}{\left(\sum_k T_k^{-n} \right)^2} \right). \quad (24)$$

2.4 Symmetrisation methods

We distinguish between the symmetrised time step, h , and the time step function, T , both of which are functions of time, t . The time step function is calculated according to the algorithms described in the previous two subsections. The symmetrised time step is the one used for the actual integration. Explicit time step symmetrisation methods assume a function $F(T)$, and a symmetric combination of $F(h_{\text{prev}})$ and $F(h_{\text{next}})$, where h_{prev} and h_{next} are the previous and next symmetrised time step, respectively. Inspired by the overview of symmetrisation methods by Dehnen (2017), we adopt the following general approaches for reversible time step adaptation:

$$(A) \quad F(T) = \frac{F(h_{\text{prev}}) + F(h_{\text{next}})}{2} \quad (\text{flip}), \quad (25)$$

$$(B) \quad \frac{dF}{dt}(T) = \frac{F(h_{\text{next}}) - F(h_{\text{prev}})}{T} \quad (\text{integration}), \quad (26)$$

$$(C) \quad \frac{dF}{dt}(T) = 2 \frac{F(h_{\text{next}}) - F(h_{\text{prev}})}{h_{\text{next}} + h_{\text{prev}}} \quad (\text{improved integration}). \quad (27)$$

Note that for method B, the right-hand side of the equation considers the change in F over a time scale T , while method C considers a time scale $(h_{\text{next}} + h_{\text{prev}})/2$. Method C (proposed by Walter Dehnen (personal communication)) is an improvement over method B, because integration errors can result in T diverging from h . For the function F , we adopt the following three functions:

$$(1) \quad F(T) = T \quad (\text{Linear}), \quad (28)$$

$$(2) \quad F(T) = \ln T \quad (\text{Logarithmic}), \quad (29)$$

$$(3) \quad F(T) = \frac{1}{T} \quad (\text{Harmonic}). \quad (30)$$

Different symmetrisation methods are obtained by considering all the permutations of the symmetrisation methods (A, B, C) and functions (1, 2, 3).

The first three symmetrisation methods are derived from method A (see Huang & Leimkuhler 1997; Holder et al. 2001; Leimkuhler & Reich 2005):

$$h_{\text{next}} = 2T - h_{\text{prev}} \quad (A1) \quad (31)$$

$$h_{\text{next}} = \frac{T^2}{h_{\text{prev}}} \quad (A2) \quad (32)$$

$$h_{\text{next}} = \frac{1}{\frac{2}{T} - \frac{1}{h_{\text{prev}}}} \quad (A3) \quad (33)$$

A second family of symmetrisation methods is derived from method B:

Abbreviation	Pairwise time step function	Weighting method	Symmetrisation method	Comment
$T_{ij,0}$	Eq. (3)			Pairwise free fall time scale
$T_{ij,1}$	Eq. (4)			Pairwise flyby time scale
T_{ij}	$T_{ij,0}, T_{ij,1}$	Eq. (20) with $q = 4$		Pairwise weighted time step
T_i	T_{ij}	Eq. (22) with $n = 4$		Weighted time step per body
W1	$T_{ij,0}, T_{ij,1}$	Eq. (19)		Weighted time step over all pairs
W2	T_{ij}	Eq. (21)		Weighted time step over all pairs
W3	T_i	Eq. (23)		Weighted time step over all bodies
A1			Eq. (31)	Linear
A2			Eq. (32)	Logarithmic
A3			Eq. (33)	Harmonic
B1			Eq. (34)	Linear, using $\tau = T$ and \dot{T}
B2			Eq. (35)	Logarithmic, using $\tau = T$ and \dot{T}
B3			Eq. (36)	Harmonic, using $\tau = T$ and \dot{T}
C1			Eq. (41)	Linear, using $\tau = (h_{\text{prev}} + h_{\text{next}})/2$ and \dot{T}
C2			Eq. (42)	Logarithmic, using $\tau = (h_{\text{prev}} + h_{\text{next}})/2$ and \dot{T}
C3			Eq. (43)	Harmonic, using $\tau = (h_{\text{prev}} + h_{\text{next}})/2$ and \dot{T}

Table 1. Overview of the time stepping methods used in this study. For methods *B* and *C*, the quantity τ refers to the time scale for the change in $F(h)$.

$$h_{\text{next}} = h_{\text{prev}} + \dot{T}T \quad (\text{B1}) \quad (34)$$

$$h_{\text{next}} = h_{\text{prev}} e^{\dot{T}T} \quad (\text{B2}) \quad (35)$$

$$h_{\text{next}} = \frac{1}{\frac{1}{h_{\text{prev}}} - \frac{\dot{T}}{T}} \quad (\text{B3}) \quad (36)$$

These methods were introduced by [Dehnen \(2017, B2\)](#) and [Hairer & Söderlind \(2005, B3\)](#), and also require the derivative of the smoothed global time step (Eq. (24)). A third family of symmetrisation methods is derived from method *C* (see [Dehnen 2017; Hairer et al. 2006](#)):

$$\dot{T} = 2 \frac{h_{\text{next}} - h_{\text{prev}}}{h_{\text{next}} + h_{\text{prev}}} \quad (37)$$

$$\frac{1}{T} \dot{T} = 2 \frac{\ln h_{\text{next}} - \ln h_{\text{prev}}}{h_{\text{next}} + h_{\text{prev}}} \quad (38)$$

$$-\frac{1}{T^2} \dot{T} = 2 \frac{\frac{1}{h_{\text{next}}} - \frac{1}{h_{\text{prev}}}}{h_{\text{next}} + h_{\text{prev}}} \quad (39)$$

Here, the linear case (Eq (37)) can easily be rewritten in terms of h_{next} . The harmonic case (Eq. (39)) can be rewritten as a second-order polynomial in h_{next} . Solving this results in two possible values for h_{next} , both of which need to be calculated and compared as \dot{T} can vary in sign as time evolves. The logarithmic case (Eq. (38)) can only be solved using some iterative root-finding algorithm, which tend to be expensive. For this particular case, we will instead consider the following relation:

$$\frac{1}{T} \dot{T} = \frac{1}{h} \dot{h} = \frac{2}{h_{\text{next}} + h_{\text{prev}}} \frac{h_{\text{next}} - h_{\text{prev}}}{\frac{h_{\text{next}} + h_{\text{prev}}}{2}} = 4 \frac{h_{\text{next}} - h_{\text{prev}}}{(h_{\text{next}} + h_{\text{prev}})^2} \quad (40)$$

This relation can also be rewritten as a second-order polynomial in h_{next} , and we will use this relation instead of Eq. (38). The third family of symmetrisation methods is then given by:

$$h_{\text{next}} = h_{\text{prev}} \frac{2 + \dot{T}}{2 - \dot{T}} \quad (\text{C1}) \quad (41)$$

$$h_{\text{next}} = \begin{cases} h_{\text{prev}}, & \text{if } \xi = 0 \text{ or } D_2 < 0. \\ -h_{\text{prev}} + \frac{2}{\xi} (1 + \sigma \sqrt{D_2}) & \text{otherwise.} \end{cases} \quad (\text{C2}) \quad (42)$$

$$h_{\text{next}} = \begin{cases} h_{\text{prev}}, & \text{if } \gamma = 0 \text{ or } D_3 < 0. \\ -\frac{h_{\text{prev}}}{2} - \frac{1}{2\gamma h_{\text{prev}}} (1 + \sigma \sqrt{D_3}) & \text{otherwise.} \end{cases} \quad (\text{C3}) \quad (43)$$

where we defined $\xi \equiv \frac{\dot{T}}{T}$, $\gamma = -\frac{\dot{T}}{2T^2}$, $D_2 = 1 - 2\xi h_{\text{prev}}$ and $D_3 = \gamma^2 h_{\text{prev}}^4 + 6\gamma h_{\text{prev}}^2 + 1$. For Eqs. (42) and (43), we have two solutions for h_{next} , one for $\sigma = +1$, the other for $\sigma = -1$. When the two solutions have a different sign, we choose the positive solution, when the solutions have the same sign, we choose the one with the minimum value of $|h_{\text{next}}|$.

These explicit symmetrisation methods are not self-starting and an initial value for h_{prev} has to be determined. In Appendix A, we demonstrate that for an eccentric binary system, the following expression is appropriate:

$$h_{\text{prev},0} = \frac{T}{1 + \frac{1}{2}\dot{T}}. \quad (44)$$

We will adopt this initial value throughout our simulations. Other initial values can be obtained based on alternative symmetrisation methods. However, the effect of a slightly different initial value is expected to be marginal as it only concerns the initialisation.

The symmetrisation methods defined above behave properly as long as the symmetrised time step, h , remains close to the time step function, T . During long term integrations of multi-body, chaotic systems however, this is not guaranteed. If $h \gg T$, energy errors will become intolerable, while if $h \ll T$, the simulation running time becomes excessive. As suggested by [Hands et al. \(2019\)](#), such instabilities can be contained by putting limits on the ratio of h/T . However, by resetting $h = T$ when a breach is detected, an irreversibility is introduced into the simulation. In order to test which time step criterion is the most robust, i.e. can keep h close to T , we

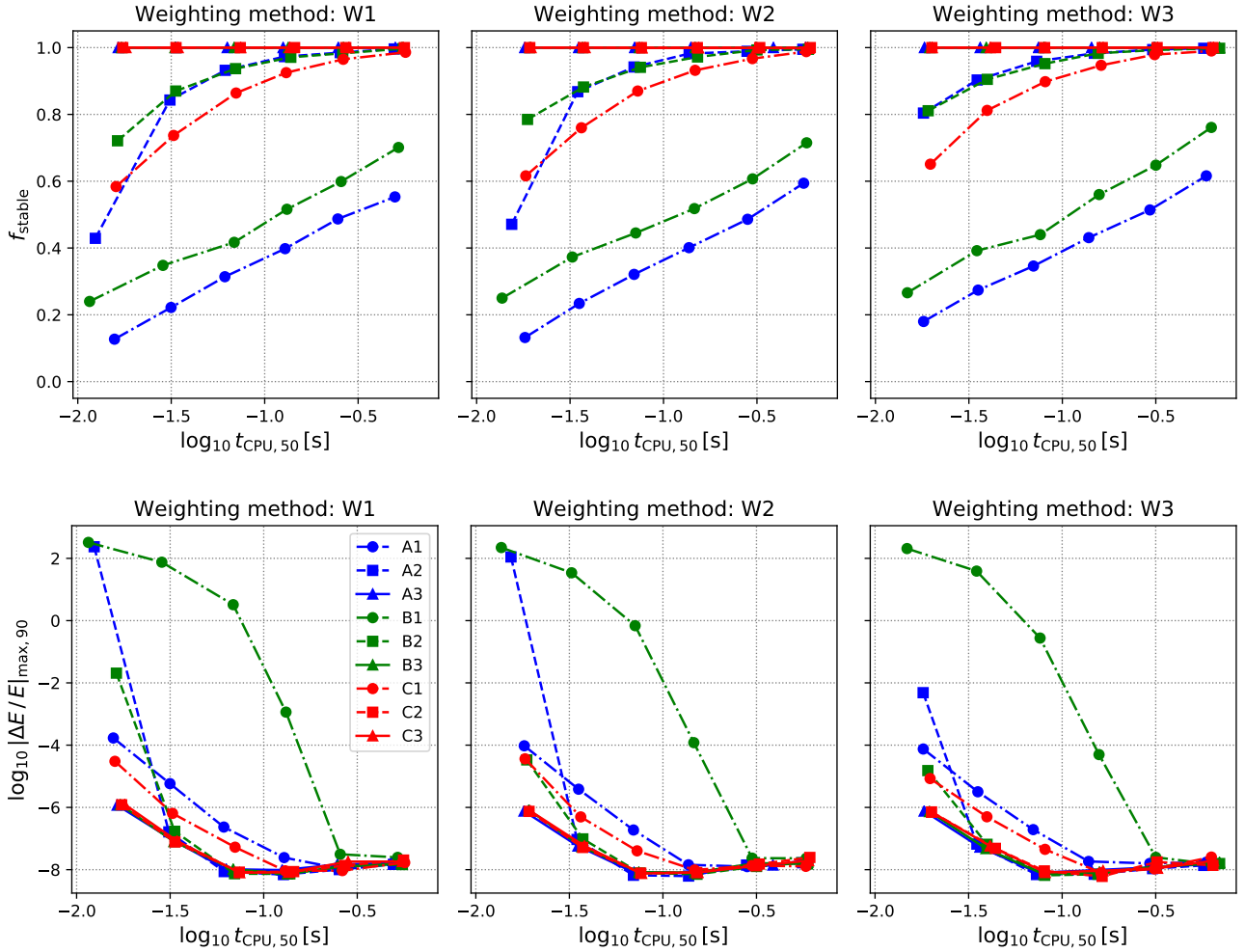


Figure 1. We generate a set of 1000 random, equal-mass triple systems drawn from a Plummer distribution. Ensembles of solutions are obtained by evolving the triples for 100 dynamical times, using different time step weighting methods, symmetrisation methods, and values for the time step parameter. For each individual simulation, we measure the CPU running time (t_{CPU}), the maximum, absolute value of the relative energy error ($|\Delta E / E|_{\text{max}}$), and the maximum and minimum ratio of the symmetrised time step and the time step function ($f = h/T$). For each ensemble of simulations, we plot the 90th-percentile of the energy error ($|\Delta E / E|_{\text{max},90}$), and the fraction of “stable” simulations (f_{stable}), for which f remained within a factor 4 from T at all times, with respect to the median CPU running time ($t_{\text{CPU},50}$).

will implement rather mild limits in which h can be a factor 128 larger or smaller than T . In practice, we have good criterion will keep h much closer to T . In this section, we have then defined three different methods for weighting time steps and nine different time step symmetrisation methods (Ai, Bi, Ci, $i = 1, 2, 3$). Therefore, there are 27 different possible combinations. In Tab. 1, we provide an overview of all time step ingredients used in this study.

3 RESULTS

In this section, we will compare the performance of the 27 permutations of the time stepping methods described in Sec. 2, for three different types of chaotic astrophysical systems. The first type consists of unstable triple stars, which will dissolve into a binary and single escaper. The second type is inspired by the Nice model, and considers unstable planetary systems. The time step size can vary drastically during planet-planet scattering. The third case considers a two million year integration of the current Solar System. By

modelling each of these three systems, we can determine if there is a single, optimal time step criterion.

3.1 Unstable triple systems

Hierarchical triple star systems consist of an inner binary system, and a third star orbiting around the center of mass of the inner binary. If the orbits are sufficiently separated, the system is stable and shows interesting long term effects, such as Lidov-Kozai cycles (Lidov 1962; Kozai 1962). However, due to a combination of stellar evolutionary and dynamical effects, it is possible for the hierarchy to change. One scenario is that the inner binary orbit expands due to mass loss from stellar winds, thereby gradually reducing the hierarchy. Once the inner and outer orbits are sufficiently close according to a stability criterion for triples (e.g. Mardling 2008), then the triple has fully destabilised and will eventually break up (for more details see Toonen et al. 2021; Hamers et al. 2021).

Due to the chaotic nature of triple systems, there is a sensitive dependence to small perturbations (e.g. Boekholt et al. 2020).

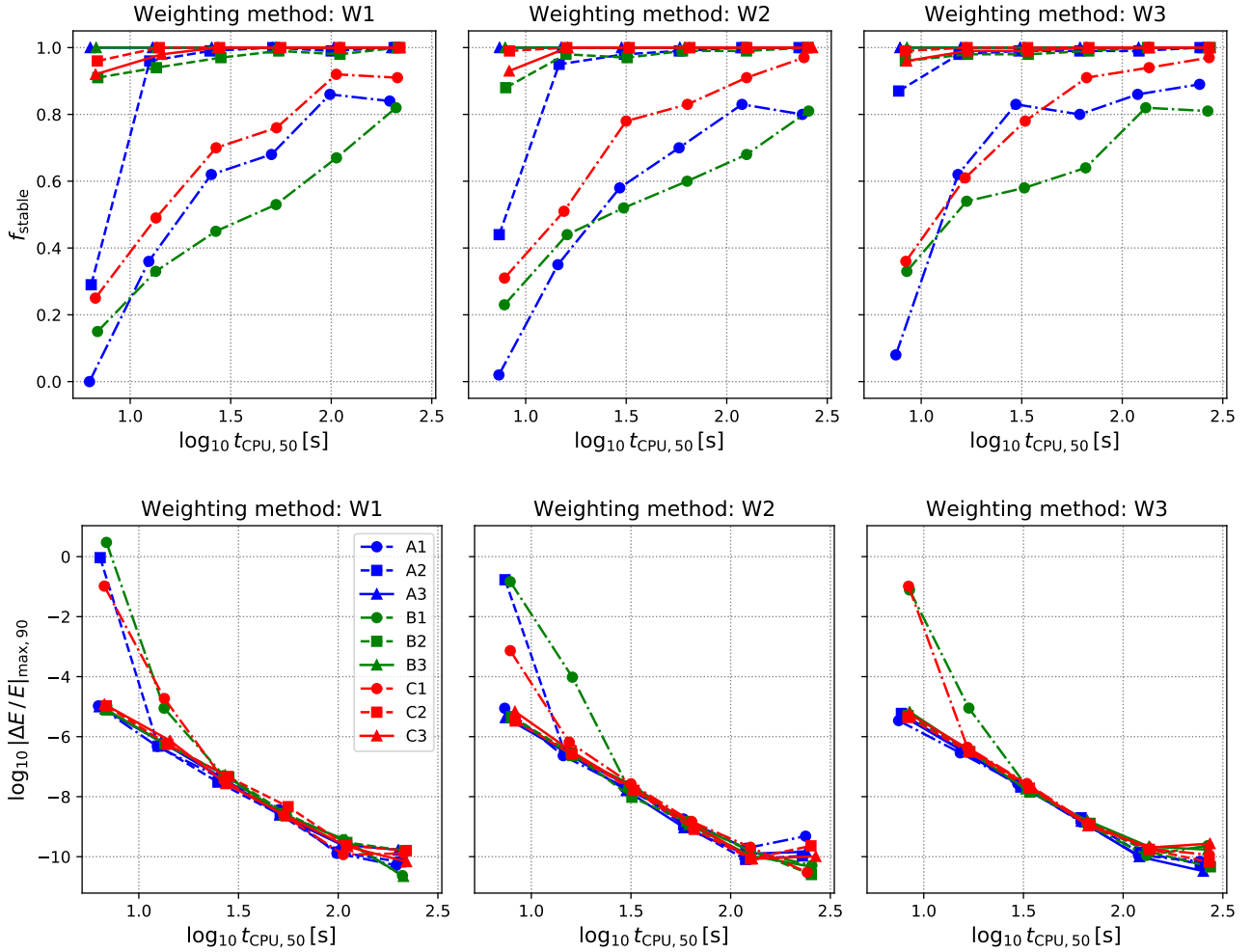


Figure 2. We generate a set of 100 random initial realisation for the Nice model, including the Sun and the four giant planets. Ensembles of solutions are obtained by evolving the initial conditions for one million years, using different time step weighting methods, symmetrisation methods, and values for the time step parameter. For each individual simulation, we measure the CPU running time (t_{CPU}), the maximum, absolute value of the relative energy error ($|\Delta E / E|_{\text{max}}$), and the maximum and minimum ratio of the symmetrised time step and the time step function ($f = h/T$). For each ensemble of simulations, we plot the 90th-percentile of the energy error ($|\Delta E / E|_{\text{max},90}$), and the fraction of “stable” simulations (f_{stable}), for which f remained within a factor 4 from T at all times, with respect to the median CPU running time ($t_{\text{CPU},50}$).

These could be caused by taking different time steps. It is therefore unfeasible to compare the performance of different time step functions for a single chaotic triple system. Instead, we generate an ensemble of 1000 random, equal-mass triple systems, drawn from a Plummer distribution (Plummer 1911). We integrate these triples for 300 N-body time units (Heggie & Mathieu 1986) (about 100 dynamical times, where a dynamical time is the average time for a star to cross the system), after which about half of the ensemble has dissolved into a permanent binary/single configuration (e.g. Boekholt & Portegies Zwart 2015). By systematically varying the time step parameter, η , we obtain ensembles of solutions with varying accuracy. The integrator is fixed to be MCL4.

In Fig. 1, we plot the results of the performance tests. First, we compare the robustness of the symmetrised time steps. We define the fraction f_{stable} as the fraction of simulations for which h remains in the interval $[T/4, 4T]$ throughout the simulation. We observe that the largest differences between the curves originate from the symmetrisation method, and that the different weighting methods produce approximately consistent results. Symmetrisation

methods A1, B1 and C1 (all based on the linear function), produce the worst results. Methods A2 and B2 become increasingly robust for smaller time step parameters (towards larger CPU times). When comparing the symmetrisation families A, B and C, we find that family C (red) tends to lay above its counterparts in families A and B. Perfect stability is obtained for the methods based on the harmonic function.

We consider a time step criterion better than another one, if for the same CPU running time, it produces a better level of energy conservation. In Fig. 1, we observe that stable symmetrised time steps indeed lead to better energy conservation. When comparing the harmonic methods (A3, B3 and C3), we find that they are competitive in performance. There is a slight advantage for A3, due to the fact that the calculation of the derivative of T is not required. We observe that the level of energy conservation stalls around 10^{-8} . This is probably due to the level of numerical precision used for

these computations (double-precision) and the spread of the resulting round-off errors.¹

3.2 Nice model: unstable planetary systems

Symplectic integrators with a constant time step are often used for modelling planetary systems. This becomes problematic when the planetary system evolves towards an instability. This potentially results in planet-planet scattering, during which very close encounters are common. The transition from the planets orbiting the star, to close planet-planet deflections, is also a great test bed for adaptive time step functions.

As an example, we adopt an initial condition based on the Nice model, which assumes a compact, resonant chain for Jupiter, Saturn, Uranus and Neptune (for more details see [Gomes et al. 2005](#); [Tsiganis et al. 2005](#)). The initial semi-major axes of Jupiter is 6 AU, Saturn is in a 3:2 resonance with Jupiter, Uranus in 3:2 with Saturn, and Neptune in 5:4 with Uranus (it is also possible to swap the initial position of Uranus and Neptune). We ignore the planetesimal disc, which would have damped the eccentricities and inclinations. This configuration is highly sensitive to small perturbations, and a slight change of the time step parameter, or changing the time step criterion, results in a different outcome. We therefore consider an ensemble of 100 realisations, where the orbital angles are chosen randomly. The systems are evolved for a million years, at which point about 80% of the systems have destabilised, resulting in at least one planet becoming unbound due to an ejection.

In Fig. 2, we plot the result of the performance test. Focusing on the stability of the symmetrised time steps, we observe again that the symmetrisation methods based on the linear function are the most unstable. The only methods which are perfectly stable in this experiment are A3 and B3. The three different weighting methods produce roughly consistent results.

Focusing on the level of energy conservation, we observe that symmetrisation methods A2, B1 and C1 produce the largest outliers at large time steps. The other methods are competitive, but again, there is a slight edge for method A3 as the derivative of the time step does not have to be calculated.

3.3 Solar System

In the previous two experiments, we found the consistent result that the symmetrisation methods A3 and B3 produce the best results. Here, we will compare these two methods further, while also varying the integration method (MCL4 and CC4). We adopt a realisation of the Solar System (from [Ito & Tanikawa 2002](#)), including the Sun, the eight planets and Pluto ($N = 10$). The Solar System is evolved up to two million years using pure Newtonian dynamics.

In Fig. 3, we plot the time evolution of the energy error and the ratio h/T . We observe that all three weighting methods produce robust results. Furthermore, the symmetrisation methods A3 and B3 produce consistent results when switching the integrator from MCL4 to CC4.

The statistical performance test is presented in Fig. 4. Here, we define the average time step size, $h_{av} = T/N_{step}$, i.e. the simulation time divided by the number of integration steps. We observe that with decreasing time steps, the ratio h/T approaches unity for

each of the solutions shown. We also observe that the integrator CC4 is faster than MCL4 for a fixed average time step size, but also less precise. In the unbiased comparison of energy error vs. CPU running time, we observe a slight advantage for the CC4 integrator. Secondly, we find a slightly better performance for the W2 weighting method, as well as for the A3 symmetrisation method. The most efficient combination is given by CC4-W2-A3.

4 CONCLUSION

Symplectic integrators with adaptive time steps are appropriate for N -body systems with close encounter episodes. The secular drift of the energy error is reduced by combining the integration method with the use of a symmetrised time step, which depends symmetrically on the old and new states. It then becomes important to have a time step, which behaves smoothly as a function of time. Our results confirm previous findings by [Hands et al. \(2019\)](#) that a weighted average of all pairwise time steps, rather than adopting the minimal pairwise value, has a stabilising effect on the energy error, but only when combined with an explicit symmetrisation method. The manual addition of bounds to the symmetrised time step can serve as a safety net. The risk of an instability in the symmetrised time step is greatly reduced by implementing a robust method. Our experiments show that the methods based on the harmonic function, in particular A3 and B3, prove to be the most robust. Our Solar System experiment also showed a slightly improved efficiency for the W2 method. In the other two experiments the differences were more subtle, hinting towards the fact that as long as there is some implementation for smoothing, that symmetrisation methods will greatly benefit from it.

Too much smoothing however, reduces the dynamic range of the adaptive time step, and can result in weighted time steps that are too large to resolve close encounters. If we define the weight of a pairwise time step as the inverse of the time step raised to the power n , then the parameter n controls the distribution of weights among the time steps. The optimal value of n scales with the number of bodies as $n \propto \log_{10} N$. For N -body systems up to a few hundred bodies, we confirm a value of $n = 10$ is sufficient.

It remains an open problem how to model chaotic N -body systems, such as dense stellar systems, without a secular drift in the energy error. Even with a symplectic integrator and symmetrised time steps, energy errors tend to drift. Error cancellation depends on the time symmetry of the integration, but also on whether the orbits themselves are time symmetric, i.e. periodic. For chaotic systems, the latter is generally not the case, so that a previous made error will generally not be undone in the future.

However, even if we were able to remove the secular drift in the energy error for chaotic N -body systems, it is not guaranteed that the solution is also accurate ([Boekholt & Portegies Zwart 2015](#), Fig. 3). Small perturbations, whether physical or numerical, grow exponentially, and saturate after only a few Lyapunov time scales (e.g. [Miller 1964](#); [Goodman et al. 1993](#); [Hut & Heggie 2002](#); [Boekholt et al. 2020](#); [Portegies Zwart et al. 2022](#)). A comparison to numerically converged solutions, as can be obtained with a code such as Brutus ([Portegies Zwart & Boekholt 2014](#); [Boekholt & Portegies Zwart 2015](#); [Boekholt et al. 2021](#)), are required to determine the statistical validity of N -body simulations with or without drifts in the energy error.

¹ Round-off errors can be reduced by using regularisation methods (e.g. [Mikkola & Tanikawa 1999](#)) or by adopting arbitrary-precision arithmetic (e.g. [Boekholt & Portegies Zwart 2015](#))

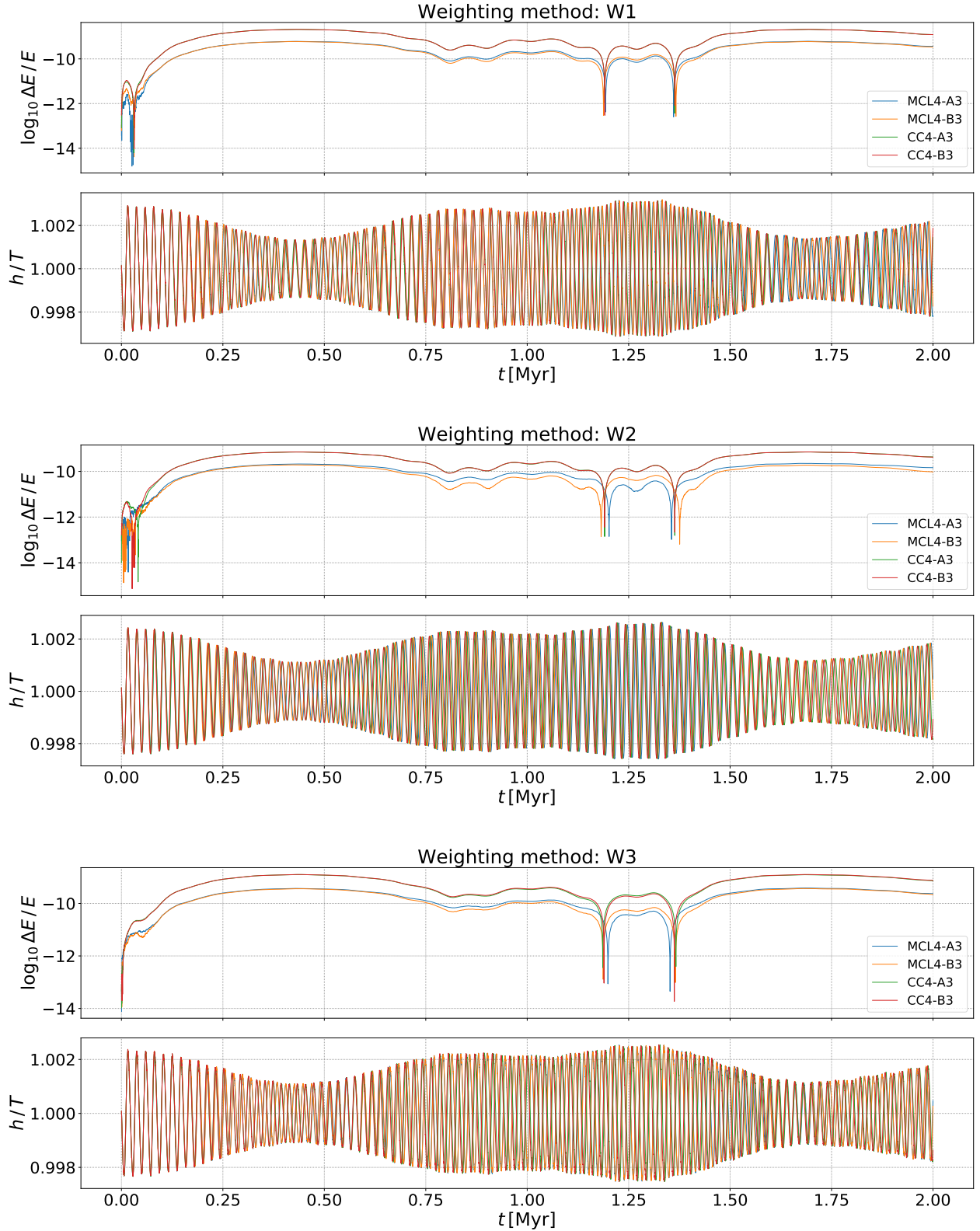


Figure 3. We evolve the Solar System (Sun, 8 planets and Pluto) for two million years. We measure the time evolution of the ratio between the symmetrised time step and the time step function, h/T , as well as the absolute value of the relative energy conservation. We plot results obtained by both the MCL4 and CC4 integrators, and we vary the time step weighting method and symmetrisation method. The time step parameter is fixed to $\eta = 2^{-6}$. Each method produces stable results, and the deviations among the different curves are very small.

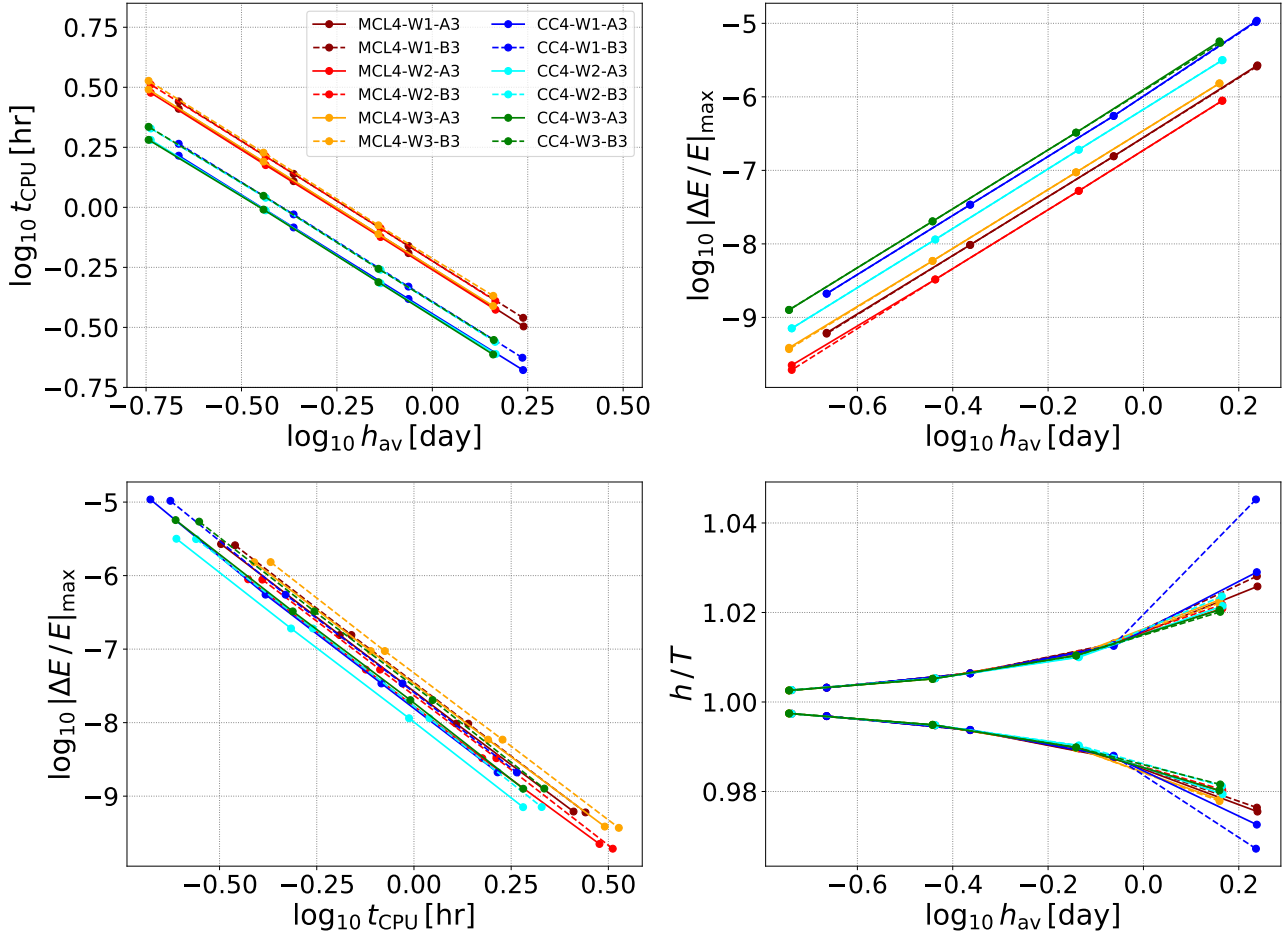


Figure 4. Performance test for the Solar System. We plot the CPU running time (top left) and maximum, relative energy error (top right) as a function of average time step size. We also plot the maximum and minimum value of the ratio h/T (bottom right). The performance comparison of energy error vs. CPU running time (bottom left) shows that the CC4-W2-A3 integration method performs the most optimal.

Acknowledgements

We are grateful to the referee Walter Dehnen for very helpful comments about this work and contributing through enlightening discussions, suggesting new ideas (in particular on the symmetrisation methods) and improving the overall presentation. The simulations were run on the Hydra computing cluster in Oxford. This project was supported by funds from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program under grant agreement No 638435 (GalNUC). This work was also supported by CFisUC (UIDB/04564/2020 and UIDP/04564/2020), GRAVITY (PTDC/FIS-AST/7002/2020), PHOBOS (POCI-01-0145-FEDER-029932), and ENGAGE SKA (POCI-01-0145-FEDER-022217), funded by COMPETE 2020 and FCT, Portugal.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

- Aarseth, S. J. 2003, *Gravitational N-Body Simulations* (Cambridge University Press)
- Boekholt, T., & Portegies Zwart, S. 2015, *Computational Astrophysics and Cosmology*, 2, 2, doi: [10.1186/s40668-014-0005-3](https://doi.org/10.1186/s40668-014-0005-3)
- Boekholt, T. C. N., Moerman, A., & Portegies Zwart, S. F. 2021, *Physical Review D*, 104, 083020, doi: [10.1103/PhysRevD.104.083020](https://doi.org/10.1103/PhysRevD.104.083020)
- Boekholt, T. C. N., Portegies Zwart, S. F., & Valtonen, M. 2020, *Monthly Notices of the Royal Astronomical Society*, 493, 3932, doi: [10.1093/mnras/staa452](https://doi.org/10.1093/mnras/staa452)
- Bolmont, E., Raymond, S. N., Leconte, J., Hersant, F., & Correia, A. C. M. 2015, *Astronomy and Astrophysics*, 583, A116, doi: [10.1051/0004-6361/201525909](https://doi.org/10.1051/0004-6361/201525909)
- Chambers, J. E. 1999, *Monthly Notices of the Royal Astronomical Society*, 304, 793, doi: [10.1046/j.1365-8711.1999.02379.x](https://doi.org/10.1046/j.1365-8711.1999.02379.x)
- Chambers, J. E., Wetherill, G. W., & Boss, A. P. 1996, *Icarus*, 119, 261, doi: [10.1006/icar.1996.0019](https://doi.org/10.1006/icar.1996.0019)
- Chin, S. A., & Chen, C. R. 2005, *Celestial Mechanics and Dynamical Astronomy*, 91, 301, doi: [10.1007/s10569-004-4622-z](https://doi.org/10.1007/s10569-004-4622-z)

Dehnen, W. 2017, Monthly Notices of the Royal Astronomical Society, 472, 1226, doi: [10.1093/mnras/stx1944](https://doi.org/10.1093/mnras/stx1944)

Dehnen, W., & Hernandez, D. M. 2017, Monthly Notices of the Royal Astronomical Society, 465, 1201, doi: [10.1093/mnras/stw2758](https://doi.org/10.1093/mnras/stw2758)

Gomes, R., Levison, H. F., Tsiganis, K., & Morbidelli, A. 2005, Nature, 435, 466, doi: [10.1038/nature03676](https://doi.org/10.1038/nature03676)

Goodman, J., Hoggie, D. C., & Hut, P. 1993, Astrophysical Journal, 415, 715

Hairer, E., Lubich, C., & Wanner, G. 2006, Springer Series in Computational Mathematics, Vol. 31, Geometric numerical integration, 2nd edn. (Springer-Verlag, Berlin), xviii+644

Hairer, E., & Söderlind, G. 2005, SIAM J. SCI. COMPUT., 26, 1838

Hamers, A. S., Perets, H. B., Thompson, T. A., & Neunteufel, P. 2021, arXiv e-prints, arXiv:2107.13620. <https://arxiv.org/abs/2107.13620>

Hands, T. O., Dehnen, W., Gratton, A., Stadel, J., & Moore, B. 2019, Monthly Notices of the Royal Astronomical Society, 490, 21, doi: [10.1093/mnras/stz1069](https://doi.org/10.1093/mnras/stz1069)

Hoggie, D. C. 1975, Mon. Not. R. Astron. Soc., 173, 729, doi: [10.1093/mnras/173.3.729](https://doi.org/10.1093/mnras/173.3.729)

Hoggie, D. C., & Mathieu, R. D. 1986, in The Use of Supercomputers in Stellar Dynamics, ed. P. Hut & S. L. W. McMillan, Vol. 267 (Springer-Verlag, Berlin Heidelberg New York), 233, doi: [10.1007/BFb0116419](https://doi.org/10.1007/BFb0116419)

Holder, T., Leimkuhler, B., & Reich, S. 2001, Applied Numerical Mathematics, 39, 367, doi: [https://doi.org/10.1016/S0168-9274\(01\)00089-7](https://doi.org/10.1016/S0168-9274(01)00089-7)

Huang, W., & Leimkuhler, B. 1997, SIAM Journal on Scientific Computing, 18, 239, doi: [10.1137/S1064827595284658](https://doi.org/10.1137/S1064827595284658)

Hut, P., & Hoggie, D. C. 2002, Journal of Statistical Physics, 109, 1017, doi: [10.1023/A:1020472526203](https://doi.org/10.1023/A:1020472526203)

Hut, P., Makino, J., & McMillan, S. 1995, Astrophysical Journal Letters, 443, L93, doi: [10.1086/187844](https://doi.org/10.1086/187844)

Hut, P., Mineshige, S., Hoggie, D. C., & Makino, J. 2007, Progress of Theoretical Physics, 118, 187, doi: [10.1143/PTP.118.187](https://doi.org/10.1143/PTP.118.187)

Ito, T., & Tanikawa, K. 2002, Monthly Notices of the Royal Astronomical Society, 336, 483, doi: [10.1046/j.1365-8711.2002.05765.x](https://doi.org/10.1046/j.1365-8711.2002.05765.x)

Jones, B. W., Sleep, P. N., & Chambers, J. E. 2001, Astronomy and Astrophysics, 366, 254, doi: [10.1051/0004-6361:20000078](https://doi.org/10.1051/0004-6361:20000078)

Kozai, Y. 1962, Astronomical Journal, 67, 591, doi: [10.1086/108790](https://doi.org/10.1086/108790)

Leimkuhler, B., & Reich, S. 2005, Simulating Hamiltonian Dynamics, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press), doi: [10.1017/CB09780511614118](https://doi.org/10.1017/CB09780511614118)

Lidov, M. L. 1962, Planetary and Space Science, 9, 719, doi: [10.1016/0032-0633\(62\)90129-0](https://doi.org/10.1016/0032-0633(62)90129-0)

Makino, J. 1991, Astrophysical Journal, 369, 200, doi: [10.1086/169751](https://doi.org/10.1086/169751)

Makino, J., Hut, P., Kaplan, M., & Saygin, H. 2006, New Astronomy, 12, 124, doi: [10.1016/j.newast.2006.06.003](https://doi.org/10.1016/j.newast.2006.06.003)

Mardling, R. A. 2008, Resonance, Chaos and Stability: The Three-Body Problem in Astrophysics, Vol. 760 (Springer-Verlag Berlin Heidelberg), 59, doi: [10.1007/978-1-4020-8431-7_3](https://doi.org/10.1007/978-1-4020-8431-7_3)

McLachlan, R. 1995, SIAM J. Sci. Comput., 16, 151

Mikkola, S., & Tanikawa, K. 1999, Monthly Notices of the Royal Astronomical Society, 310, 745, doi: [10.1046/j.1365-8711.1999.02982.x](https://doi.org/10.1046/j.1365-8711.1999.02982.x)

Miller, R. H. 1964, Astrophysical Journal, 140, 250

Pelupessy, F. I., Jänes, J., & Portegies Zwart, S. 2012, New Astronomy, 17, 711, doi: [10.1016/j.newast.2012.05.009](https://doi.org/10.1016/j.newast.2012.05.009)

Petit, A. C., Laskar, J., Boué, G., & Gastineau, M. 2019, Astronomy and Astrophysics, 628, A32, doi: [10.1051/0004-6361/201935786](https://doi.org/10.1051/0004-6361/201935786)

Plummer, H. C. 1911, Monthly Notices of the Royal Astronomical Society, 71, 460, doi: [10.1093/mnras/71.5.460](https://doi.org/10.1093/mnras/71.5.460)

Portegies Zwart, S., & Boekholt, T. 2014, Astrophysical Journal Letters, 785, L3, doi: [10.1088/2041-8205/785/1/L3](https://doi.org/10.1088/2041-8205/785/1/L3)

Portegies Zwart, S. F., Boekholt, T. C. N., Por, E. H., Hamers, A. S., & McMillan, S. L. W. 2022, Astronomy and Astrophysics, 659, A86, doi: [10.1051/0004-6361/202141789](https://doi.org/10.1051/0004-6361/202141789)

Quinlan, G. D., & Tremaine, S. 1990, Astronomical Journal, 100, 1694, doi: [10.1086/115629](https://doi.org/10.1086/115629)

Quinn, T. R., Tremaine, S., & Duncan, M. 1991, Astronomical Journal, 101, 2287, doi: [10.1086/115850](https://doi.org/10.1086/115850)

Rein, H., Hernandez, D. M., Tamayo, D., et al. 2019, Monthly Notices of the Royal Astronomical Society, 485, 5490, doi: [10.1093/mnras/stz769](https://doi.org/10.1093/mnras/stz769)

Toonen, S., Boekholt, T. C. N., & Portegies Zwart, S. 2021, arXiv e-prints, arXiv:2108.04272. <https://arxiv.org/abs/2108.04272>

Trenti, M., & Hut, P. 2008, arXiv e-prints, arXiv:0806.3950. <https://arxiv.org/abs/0806.3950>

Tsiganis, K., Gomes, R., Morbidelli, A., & Levison, H. F. 2005, Nature, 435, 459, doi: [10.1038/nature03539](https://doi.org/10.1038/nature03539)

Wang, L., Iwasawa, M., Nitadori, K., & Makino, J. 2020, Monthly Notices of the Royal Astronomical Society, 497, 536, doi: [10.1093/mnras/staa1915](https://doi.org/10.1093/mnras/staa1915)

APPENDIX A: INITIAL VALUE FOR THE PREVIOUS SYMMETRISED TIME STEP

Explicit time step symmetrisation methods keep track of the previous symmetrised time step, h_{prev} , as discussed in Sec. 2. An initial value has to be set in order to start the simulation. The simplest solution is to set $h_{\text{prev}} = T$ initially. However, this turns out not to be the best option, and may affect the potential future divergence of the symmetrised time step from the time step function. A better alternative is to make use of one of the symmetrisation methods. If we consider a function, F , of the time step, we can estimate h_{prev} as

$$F(h_{\text{prev}}) = F(T) - \frac{1}{2}T \frac{dF}{dT}. \quad (\text{A1})$$

If we then apply the harmonic function, $F(h) = 1/h$, we obtain

$$\frac{1}{h_{\text{prev}}} = \frac{1}{T} + \frac{1}{2} \frac{\dot{T}}{T} = \frac{1}{T} \left(1 + \frac{1}{2} \dot{T} \right). \quad (\text{A2})$$

Or taking the inverse, we obtain:

$$h_{\text{prev}} = \frac{T}{1 + \frac{1}{2} \dot{T}}. \quad (\text{A3})$$

Note that this expression is similar to the approximate, symmetrised time step presented by Pelupessy et al. (2012), but rather in the negative time direction. In Fig. A1, we compare the two start-up values for an eccentric binary system. We observe that when $h_{\text{prev}} = T$ and when we start away from pericenter or apocenter,

that the symmetrised time step oscillates between two limits, but is biased with respect to the time step function. If we use the improved start-up value however, we confirm the bias is removed. We implement this improved initial value throughout our simulations. Other estimates for $h_{\text{prev},0}$ can be obtained using alternative functions and/or symmetrisation methods, or by replacing the term TdF/dt by $h_{\text{prev},0}dF/dt$. However, since this only concerns the initialisation of h_{prev} , and our improved estimate presented above works well, we expect the effect of an alternative initial value to be marginal.

APPENDIX B: SMOOTHING WITH NUMBER OF BODIES

Smoothing of the time step function has a positive effect on the functionality of time step symmetrisation methods. However, too much smoothing leads to time steps which are too large. Since the weighting methods sum over all pairs of bodies (for methods W1 and W2), or over each body individually (method W3), we expect a dependence between the amount of smoothing required and the number of bodies. In Sec. 2.3, we derive that the minimum weight parameter should scale as $n \propto \log_{10} N$. The normalisation however, depends on the N-body configuration through the distribution of pairwise time step values.

In order to test the scaling, we consider equal-mass Plummer spheres (Plummer 1911) with N ranging from 4 to 4,096. Each system is evolved for 1 N-body time unit, and for a specific value of n . By varying n systematically, we measure how this changes the maximum deviation of the global time step, T , from the minimum pairwise time step, $T_{\min} = \min(T_{ij})$. By interpolation, we calculate that value of n for which the maximum deviation equals a factor $T/T_{\min} = 2$. In Fig. B1, we plot this critical value of n as a function of N for the W3 weighting method. We confirm that the scaling approximately follows the expectation. Furthermore, for a particle number up to a few thousand, we confirm that $n > 6$ is required. The choice of $n = 10$ adopted here, and also by Hands et al. (2019), thus leads to a sufficient amount of smoothing to facilitate the symmetrisation process, while avoiding too much smoothing.

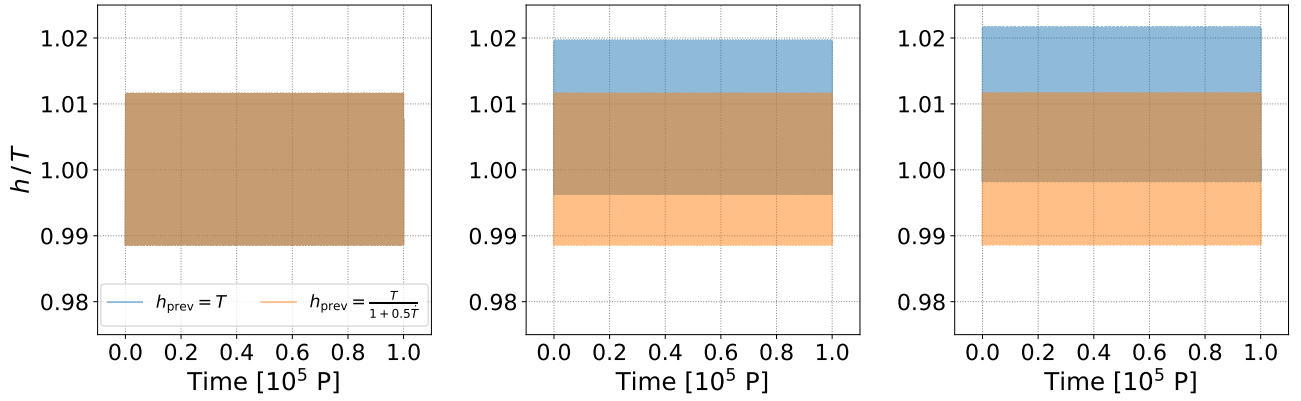


Figure A1. We evolve an equal-mass and eccentric ($e = 0.99$) binary system for 10^5 orbital periods (P). We vary the initial true anomaly per panel: 0 (left), 90 (middle) and 175 (right) degrees. The symmetrisation method used is B3 (see Tab. 1). We plot the time evolution of the ratio between the symmetrised time step, h , and the time step function, T . We compare two values for the initial previous symmetrised time step, h_{prev} (see legend). The criterion based on T and \dot{T} gives robust results in the sense that h oscillates closely around T for all three initial true anomalies.

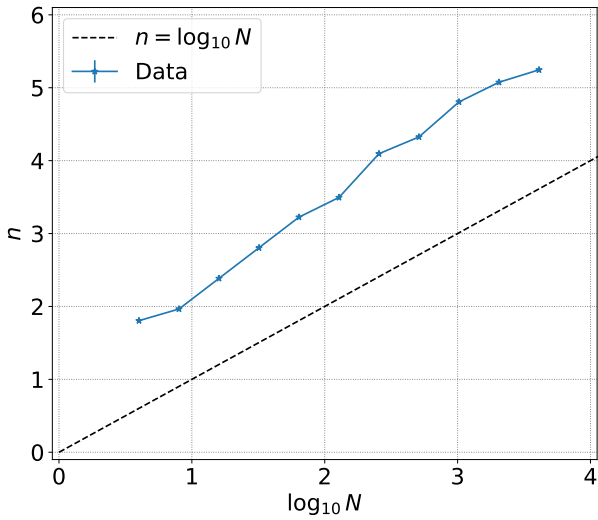


Figure B1. We consider equal-mass Plummer spheres consisting of N bodies. For each value of N we measure the value of the weight parameter, n , for which the global time step, T , is equal to twice the minimum pairwise time step, i.e. $T(n) = 2 \min\{T_{ij}\}$, using the W3 method. We confirm the expectation that approximately $n \sim \log_{10} N$. The normalisation however, depends on the N -body configuration through the distribution of time steps.