Research paper

Non-intrusive hierarchical coupling strategies for multi-scale simulations in gravitational dynamics

Simon Portegies Zwart [a,*], Inti Pelupessy [b], Carmen Martínez-Barbosa [c], Arjen van Elteren [a], Steve McMillan [d]

[a] *Leiden Observatory, Leiden University, PO Box 9513, 2300 RA, Leiden, The Netherlands*
[b] *Netherlands eScience Center, Amsterdam*
[c] *Deltares Software Center unit, Delft, The Netherlands*
[d] *Department of Physics and Astronomy, Drexel University, Disque Hall, 32 S 32nd St., Philadelphia, PA 19104, USA*

A R T I C L E I N F O

A B S T R A C T

Hierarchical code coupling strategies make it possible to combine the results of individual numerical solvers into a self-consistent symplectic solution. We explore the possibility of allowing such a coupling strategy to be non-intrusive. In that case, the underlying numerical implementation is not affected by the coupling itself, but its functionality is carried over in the interface. This method is efficient for solving the equations of motion for a self-gravitating system over a wide range of scales. We adopt a dedicated integrator for solving each particular part of the problem and combine the results to a self-consistent solution. In particular, we explore the possibilities of combining the evolution of one or more microscopic systems that are embedded in a macroscopic system. The here presented generalizations of BRIDGE include higher-order coupling strategies (from the classic 2nd order up to 10th-order), but we also demonstrate how multiple bridges can be nested and how additional processes can be introduced at the bridge time-step to enrich the physics, for example by incorporating dissipative processesor. Such augmentation allows for including additional processes in a classic Newtonian *N*-body integrator without alterations to the underlying code. These additional processes include for example the Yarkovsky effect, dynamical friction or relativistic dynamics. Some of these processes operate on all particles whereas others apply only to a subset.

The presented method is non-intrusive in the sense that the underlying methods remain operational without changes to the code (apart from adding the get- and set-functions to enable the bridge operator). As a result, the fundamental integrators continue to operate with their internal time step and preserve their local optimizations and parallelism. Multiple bridges can be nested and coupled hierarchically, allowing for the construction of a complex environment of multiple nested augmented bridges. While the coupling topology may become rather complicated, we introduce the hierarchical coupling language (HCL), a meta language in which complex bridge topologies can be described. The meta language is meant for stimulating the discussion on even more complex hierarchies in which the bridge operators are introduced as patterns

* Corresponding author.
  E-mail address: spz@strw.leidenuniv.nl (S. Portegies Zwart).

We present example applications for several of these cases and discuss the conditions under which these integrators can be applied. Typical applications range over 10 orders of magnitude in temporal and spatial scales when we apply the method to simulating planetary systems (au spatial and year-temporal scale) in a star cluster that orbits in the Galaxy (100 kpc-spatial and 10 Gyr-temporal scale).

1. Introduction

Scientific progress is mediated by performing computer simulations of laboratory experiments or observed phenomena. This is generally done by executing complex instructions on digital computers that are inspired by a Turing machine [1]. The results of these simulations are subsequently interpreted with similar trust as if they were produced in a lab experiment or from observations [2]. The exponential growth in computer power [3] and improved expressiveness in computer languages [4] enables researchers to perform ever more complex calculations. At the same time, multi-messenger observations and the gradual increase in the resolution of lab experiments require simulations to include more details, wider scales and a broader palette of phenomena and therefore they become more complex.

Simulation research is slowly progressing into a multi-scale regime. Once the range in scales exceeds three or four orders of magnitude, we refer to them as multi-scale simulations [5]. The macroscopic scale in multi-scale simulations tend to progress slowly compared to the microscopic scale and a large number of operations needed in the latter tends to introduce round-off and convergence errors.

In those cases, one of the extremes, typically the microscopic scale, is addressed differently than the rest of the system. Speed-up and improved convergence is then often realized by incorporating (semi)analytic solutions or approximations. As a consequence, the underlying code becomes specific for the particular problem: The scale transition tends to be hard-coded for each problem-dependent topology.

Problem specific implementations often lead to the boundless growth of the source code due to adaptations made upon reuse. In such dinosource [6], the newly added functionality does not increase functionality because deprecated code is not removed but left to decay [7].

Dinosource can be prevented if the individual components of the multi-scale hybridization would be non-intrusive. This allows extensions of the original functionality to be added without affecting any other parts of the code.

If multi-scale problems can be effectively addressed by non-intrusive code-coupling strategies, it may also be possible to expand the range of the domain-specific code by coupling it with codes from a completely different domain. This strategy mediates performing multi-physics simulations without interfering with the direct functionality of the individual solvers. Such non-intrusive domain-specific coupling will not only support multi-scale simulation environments but also multi-physics.

Non-intrusiveness code-coupling allows code development independently from any of the other solvers. This strict separation of functionality makes it possible to develop a dedicated implementation on one domain at some scale without having to worry about any of the other scales or domains. It makes code confined, clean, readable and maintainable. Code that solves for the same physics on the same scale but written for a different architecture may be developed in parallel and independently of each other.

One of the advantages of this Duplo-approach [6] is support for additional functionality in the form of extensions and expansions. In the former, we envision additional functionality inside the coupling method that adds functionality to the data-driven domain. Examples include boundary conditions or run time variations to the coordinate systems. The latter case includes additions to the underlying physics that operates on the same temporal or spatial scale.

There is no particular reason why code coupling should be limited to two scales or domains, but it is possible to build a cascade of solvers. With such hybridization, one can extend the coupling over many orders of magnitude with a cascade of dedicated solvers each addressing a limited range of scales.

Another attractive aspect is the possibility of hybridizing coupling strategies in a complex hierarchy of codes. If the implementation supports controlling across-scale and across-disciplinary interactions it becomes possible to tune the scale or nature of the interactions depending on the problem.

If truly non-intrusive, the coupling topology may be allowed to change at run time. If, for example, the bottleneck in the simulation shifts, because some criterion is satisfied or a local situation emerges, some codes may be replaced or terminated whereas others may be initiated. This allows for flexible run-time behavior, at the cost of transparency for supercomputer job-scheduling brokers [8,9].

In the non-intrusive multi-scale and multi-physics coupling strategy presented here, all these requirements are satisfied. The coupling strategy is non-intrusive and it can be controlled at run time, expanded, extended, hybridized and hierarchically nested.

In this paper, we describe the method (see Section 2), its adaptations and extensions (see Section 3) and we provide a more practical notation in the hierarchical coupling language (HLC) in Section 4. Eventually, in Section 5 we provide validation and examples. But first we review the problem in Section 2.1 from the point of view of gravitational dynamics.
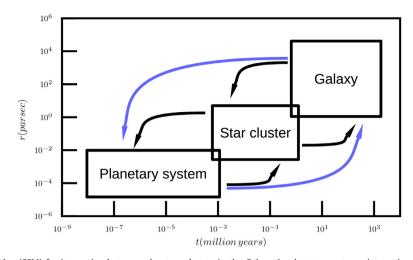
Fig. 1. Scale Separation Map (SSM) for interaction between planets and stars in the Galaxy. In planetary systems, interactions take place on a timescale of days to 1000 years and on spatial scales ranging from 0.3 to about 100 au. For the Solar system, these scales should be associated with the orbit of the planet Mercury on the small scales and the Kuiper belt on the longest scales. The stars in a clusters interact on time scales ranging from 10^3 to 10^6 years and typical distance scales range from 10^4 au to about 1 parsec. Galactic scales range in billion years and 30 kilo-parsec. The arrows indicate the interaction between neighboring systems, whereas the blue arrows show the cross-scale interactions. The slight overlap of the various domains illustrate the sometimes fuzzy devision between these, and the demand for introducing cross-boundary domain solutions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2. Non-intrusive coupling strategy for gravitational dynamics

2.1. Scale separation in gravitational dynamics

Gravitational dynamics provides an excellent starting point because the computational complexity of the problem demands novel software. Newton's [10] equations of motion for $N > 2$ self-gravitating mass-points is one of the oldest and most outstanding problems in astrophysics [11]. The lack of a practical analytic solution together with the intrinsic chaotic behavior of the system demands numerical integration to extremely high precision and accuracy [12]. The unfavorable scaling of the compute time ($\mathcal{O}(N^2)$ for a direct force-evaluation scheme) requires enormous resources even for a relatively small problem. To overcome these limitations a wide variety of algorithms have been designed in which computer time can be reduced by trading accuracy for speed. In this way, families of algorithms are dedicated (and considered suitable by the community) for addressing specific problems in astrophysics.

One of these problems includes the evolution of a star cluster in orbit around the Galactic center. This problem is too expensive in terms of computer time to be integrated with an accurate method in which the forces are evaluated directly, whereas a hierarchical method such as a tree-code [13] is insufficiently precise.

This lead Michiko Fujii and co-workers in 2007 to design a strategy in which both methods could be hybridized [14]. Their method, called BRIDGE, is based on a second-order extension of the mixed-variable symplectic scheme developed in the context of long term integrations of planetary systems [15].

The classic implementation is rather rigid but combines a direct 4th-order predictor-corrector Hermite scheme [16,17] with a 2nd-order tree-code. The coupling method itself is 2nd order. The coupling requires the specification of a rigid cross-over time step. The length of this step is chosen to minimize the error produced in the interface. Properly choosing this step size requires some knowledge of the system and assumes that it does not change much with time. This hybridization allows the integration of a small but dense star cluster to be carried out with high accuracy together with a large number of field stars using low accuracy. A similar strategy was later employed for studying merging supermassive black holes in galactic nuclei [18] and to simulate galaxy mergers [19]. We present a generalization of this BRIDGE method to higher-order and to include dissipative forces but most importantly to facilitate non-intrusive coupling.

The range in scales of the problem we have in mind is illustrated in a scale separation map (SSM) [20]. In Fig. 1, we present the SSM for planetary systems (bottom left) in star-clusters (middle) which are a part of the Galaxy (top right).

The evolution of planetary systems in a star cluster which again is orbiting the Galaxy provides an excellent example because all aspects of the non-intrusive hierarchical coupling strategy can be debated in this context. Planetary systems tend to be composed of one star (or maybe two) which is orbited by some tens of planets, hundreds of moons and millions of minor bodies. They are commonly resolved using a symplectic method in which energy is preserved on a secular time scale while conserving the orbital phase [21]. Due to the generally tight orbits of some of the planets or moons, integration time steps can be as small as days or hours.

The majority of planetary systems are born in clustered environments with thousands of members. The dynamical time-scales of these systems range from a few thousand to a few million years. Star clusters are commonly integrated using the Hermite scheme, which is not symplectic but 4th or 6th order. Tight planets may experience thousands of integration steps within a single integration step of a star in the cluster. A non-intrusive coupling strategy may be beneficial in this case because it leaves both the symplectic and the direct Hermite codes unaltered.

The star cluster itself orbits a galaxy. The latter is composed of hundreds of billion stars and the mean orbital time scale is of the order of a few hundred million years. The large number of stars in a galaxy demands a fast integration method with a scaling much better than N^2 and time steps thousands of times larger than those employed in the star clusters. Galaxies, therefore, tend to be integrated using a hierarchical method [13,22] or a self-consistent mean-field [23,24] or fast multi-pole [25] method.

The spatial and temporal separation, illustrated in Fig. 1 shows that these systems tend to be separated in temporal and spatial scales. In the figure, we introduce a little overlap to indicate that there may be a mesoscale, but in most applications, the various domains are well separated.

There is no one-size-fits-all *N*-body solution for the entire domain. Integrating a galaxy with methods generally used in planetary dynamics will not give you much progress, whereas the integration of a planetary system using one of the methods designed for galactic dynamics will not give you a (community accepted) interpret-able solution. The common practice is that planets and star clusters are generally ignored in galactic dynamics, and in studies on planetary dynamics the Galaxy and star clusters are neglected.

This poses no problem so long as planets stay close to their host star and stars stay in their clusters. In those cases, one could solve the entire problem as a number of individual disconnected problems. However, a planet may escape its host star to become a cluster member or even part of the Galaxy at large, or galactic stars may be captures by a star cluster and interact with a local planetary system. In this way, particles can change domain at run time. This is illustrated with the arrows in Fig. 1. This exchange of objects, and therefore of information between the microscopic, the mesoscopic and the macroscopic system demands for a method in which each of these systems is resolved to sufficient accuracy and precision.

The strict separation of the particular solvers for the sub-domains while realizing the coupling non-intrusively has several advantages. First of all, it allows the independent development and running of each of the scale-domains. It also allows the incorporation of additional processes to some particles but ignores them in others. For the smallest scale, one may desire to include non-Newtonian forces, such as the Yarkovsky effect [26] or general relativity [27,28], whereas for the largest scale these effects are irrelevant. On the other hand, the largest-scale may include dynamical friction [29] or Milgromian dynamics [26,30] which are irrelevant for the microscopic scales.

An addition advantage of the strict separation of scales, it is possible to operate each system in its own frame of reference and with a separate coordinate system. The microscopic and the macroscopic systems could both operate using a typical lengh-scale and time scale of order unity. This enables multi-scale simulations that extend beyond the dynamic range limited by the IEEE double-precision standard.

2.2. Second-order non-intrusive coupling

In the classic scheme, two different gravitational solvers are combined to bridge the wide range of scales. The microscopic scale in [14] was a cluster of $\mathcal{O}(10^3)$ stars in a volume of $\mathcal{O}(1)$ pc, whereas the macroscopic scale was a galactic nucleus of $\mathcal{O}(10^8)$ stars in a $\mathcal{O}(100)$ pc radius sphere. The relaxation time-scales cover more than 5 orders of magnitude. In this example, the star cluster was integrated using a 4-th order Hermite predictor-corrector direct-summation scheme. The interactions between stars in the galaxy, and between the cluster and the galactic stars were resolved using a hierarchical tree-code. Here we briefly discuss the classic coupling method, as a preparation for the discussion on its high-order generalization and the augmented bridge (see Section 3).

The BRIDGE integrator can be formulated from a Hamiltonian splitting argument, in a way similar to the derivation of symplectic integrators used in planetary dynamics. The Hamiltonian of a *N*-body system with sub-systems *A* and *B* under gravitational interaction is given by the expression:

$$H = \sum_{i \in A \cup B}^{N} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i < j \in A \cup B}^{N} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||} . \tag{1}$$

Here *G* is Newton's constant, m_i, $\mathbf{r}_i$ and $\mathbf{p}_i$ are the mass, and vectors for the position and momentum of particle *i*.

The systems *A* and *B* may represent a star cluster and its parent Galaxy, respectively. Following [14], the Hamiltonian shown in Eq. 1 can be separated in the following way:

$$H = H_{A+B} + H_{\text{int}} = H_A + H_B + H_{\text{int}}, \tag{2}$$

where:

$$H_A = \sum_{i \in A}^{N_A} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i \neq j \in A}^{N_A} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||} , \tag{3a}$$

$$H_B = \sum_{i \in B}^{N_B} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i \neq j \in B}^{N_B} \frac{Gm_i m_j}{\|\mathbf{r}_i - \mathbf{r}_j\|} , \tag{3b}$$

$$H_{\text{int}} = -\sum_{\substack{i \in A, \\ j \in B}}^{N_{A+B}} \frac{Gm_i m_j}{\|\mathbf{r}_i - \mathbf{r}_j\|} . \tag{3c}$$

In the last equations we do not require to explicitly state $j < i$ because i and j are from separate sets. The time evolution of the whole system can be written, for a second order approximation, as follows:

$$e^{\tau H} \simeq e^{\frac{\tau}{2} H_{A+B}} e^{\tau H_{\text{int}}} e^{\frac{\tau}{2} H_{A+B}} \quad \text{or}$$
$$e^{\tau H} \simeq e^{\frac{\tau}{2} H_{\text{int}}} e^{\tau H_{A+B}} e^{\frac{\tau}{2} H_{\text{int}}} \tag{4}$$

Here τ corresponds to the time step in which the two methods are coupled.

By defining we write the drift and kick operators as:

$$D(\tau) \equiv \prod_k^Q e^{\tau H_{A+B}} \quad \text{and} \tag{5a}$$

$$K(\tau) \equiv \prod_{k \neq l}^Q e^{\tau H_{\text{int}}} , \tag{5b}$$

we then rewrite Eq. 4 as

$$\mathcal{B}_2(\tau) = D\left(\frac{\tau}{2}\right) K(\tau) D\left(\frac{\tau}{2}\right), \quad \text{or}$$
$$\mathcal{B}_2(\tau) = K\left(\frac{\tau}{2}\right) D(\tau) K\left(\frac{\tau}{2}\right), \tag{6}$$

respectively, where we wrote the bridge operator as $\mathcal{B}_2(\tau)$, indicating that it is second order and dependent on the bridge time step τ.

Since H_{int} depends on the positions only, the operator $e^{\tau H_{\text{int}}}$ represents a pure momentum kick. During this process, the velocity of elements in the microscopic system is updated with the external force generated by the macroscopic system. To prevent confusion in terms of micro and macro, we tend to refer to the more astronomical application in which the microscopic system could be referring to the stars in a cluster whereas the macroscopic system refers to the cluster as part of the Galaxy. In Fig. 1 we presented this view schematically in a scale separation map. The velocities of the stars in the galaxy are also updated after computing the acceleration due to their self-gravity using the tree code.

Since H_A and H_B are completely independent, the evolution operator $e^{\tau H_{A+B}} \equiv e^{\tau H_A} e^{\tau H_B}$ consists of the separate evolution of the two subsystems. A full time-step in BRIDGE then consists of

 i) mutually kicking the sub-systems A and B for $\tau/2$,
 ii) evolving the two sub-systems A and B in isolation for τ using suitable codes together with an update of their positions, and
 iii) mutually kicking the sub-systems A and B for another $\tau/2$.

The strict separation of the operator in Eq. 4 allows us to solve both parts separately but combine the result to a self-consistent solution of the whole combined system. This enables us to split any compound solver to be separated into fundamental parts that can be solved individually and subsequently combined. This again allows us to write efficient and confined solvers for each of these individual parts. Even though depending on the adopted integrator each of the coupled codes scales as $\mathcal{O}(N^2)$ or $\mathcal{O}(N \log(N))$, the coupling itself scales with $\mathcal{O}(N)$.

The modularization achieved enables a more efficient calculation of the evolution of the joined system under the condition that the time step of the macroscopic system (the interaction term H_{int}) exceeds that of any of the microscopic time steps (of H_A and H_B systems). This is the case, but not exclusively so, if the spatial and temporal scales of the microscopic system are well separated from the macroscopic system. Once this condition is met, it is possible to integrate the microscopic and the macroscopic systems with different integrators geared towards their respective requirements. To this point, the method we described is not different from the classic bridge method [14].

It is not difficult to find a counterexample where the BRIDGE integrator degenerates: take e.g. a star cluster where the stars are assigned to system A and B at random. In this case, the formal splitting is still valid, but the bridge time step τ reduces to the global minimum time step. Note that in the approach above, the coupling strategy is defined manually at the beginning of the simulation and therefore the coupling remains static throughout the time evolution of the system. If a merger of the two clusters occurs during the simulation, the BRIDGE scheme evolves into the degenerate state. The degeneration of the solver can be mitigated by allowing the coupling to be time-dependent and variable and if we allow the two systems to exchange particles at such intervals. A similar strategy was adopted by [18] but then coded directly in C.

2.3. High-order non-intrusive coupling

The classic BRIDGE scheme may experience numerical difficulties when the spatial and/or temporal scales of an interaction of two or more sub-systems become comparable. This may in part be resolved by increasing the order of the coupling strategy. Increasing the bridge-order also allows us to use higher-order dedicated integrators without losing precision in the bridge step. We present a generalization of the classic BRIDGE scheme to an arbitrary number of systems and high orders. We begin by assuming a system of particles, $S = \bigcup_k S_k$, composed by a number Q of sub-systems S_k. In this case the total Hamiltonian of the system,

$$H = \sum_{i \in S}^{N} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i \neq j \in S}^{N} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||} , \tag{7}$$

can be split such that we obtain:

$$H = \sum_{k}^{Q} H_{S_k} + \sum_{k \neq l}^{Q} H_{S_k S_l}^{\text{int}} . \tag{8}$$

The terms in Eq. 8 are given by the following relations:

$$H_{S_k} = \sum_{i \in S_k}^{N} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i \neq j \in S_k}^{N} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||} , \tag{9a}$$

$$H_{S_k S_l}^{\text{int}} = \sum_{\substack{i \in S_k, \\ j \in S_l}}^{N} -\frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||} . \tag{9b}$$

Based on this splitting, a general, multi sub-system, second-order time-evolution operator can be constructed, as we demonstrated in Section 2.2.

Similarly to the classic BRIDGE, operators $e^{\tau H_{S_k}}$ independently evolves each of the sub-systems S_k in isolation. Operators $e^{\tau H_{S_k S_l}^{\text{int}}}$ represents the pure momentum kicks due to the interaction between sub-systems S_k and S_l. In the case of the Hamiltonian in eq. 8, the forces due to the interaction terms do not depend on velocities, therefore, the operators $e^{\tau H_{S_k S_l}^{\text{int}}}$ are commutative. We note, however, that commutability is not possible for velocity-dependent forces and therefore, a special treatment is required (see Section 3.3 and for an example Section 5.4). Each of the operators $e^{\tau H_{S_k}}$ and $e^{\tau H_{S_k S_l}^{\text{int}}}$ can be associated to different solvers running concurrently.

A high-order BRIDGE scheme can be constructed in a similar way as in a symplectic integrator. Eq. 6 can be extended to a higher order by composition of $D(\tau)$ (see Eq. 5a) and $K(\tau)$ (see Eq. 5b) operators [31]. For a 4th symmetric composition with 4 stages, the high-order BRIDGE takes the form (see also Eq. 53):

$$\mathcal{B}_\triangle(\tau) = \mathcal{D}(\sqcap_\prime \tau)\mathcal{K}(\sqcap_\prime \tau)\mathcal{D}(\sqcap_\infty \tau)\mathcal{K}(\sqsubseteq_\infty \tau)\mathcal{D}(\sqcap_\in \tau)\mathcal{K}(\sqsubseteq_\infty \tau)\mathcal{D}(\sqcap_\infty \tau)\mathcal{K}(\sqsubseteq_\prime \tau)\mathcal{D}(\sqcap_\prime \tau) , \tag{10}$$

for convenience we list the coefficients for u_i and v_i for a selection of choices in Table 1 of Section 6, but for a more complete overview we refer to [32] and [31]. For a sixth order symmetric composition, $D(\tau/2)K(\tau)D(\tau/2)$ leads to (see also Eq. 56)

$$\mathcal{B}(\tau) = D(w_0/2\tau)K(w_0\tau)D((w_0 + w_1)/2\tau)\ldots$$
$$\ldots D((w_{s-1} + w_s)/2\tau)K(w_s\tau)D(w_s/2\tau), \tag{11}$$

with coefficients w_i given in Table 1 (see also [33], their Eqs. 11–17). We present a complete listing of the bridge equations in Eq. 52–Eq. 59. The self-adjoined methods associated to Eqs. 10 and 11 are also possible. The formulation above provides a fully symplectic time evolution if the codes being bridged are symplectic as well.

Using this scheme, integrators of different orders can be coupled to construct a high-order scheme by matching the order of the BRIDGE to be used during the coupling. For example, when coupling a sixth-order to a fourth-order method, it is probably not appropriate to choose $\mathcal{B}_2(\tau)$, but rather $\mathcal{B}_4(\tau)$ or $\mathcal{B}_6(\tau)$ to have a convergent compound-method of fourth or sixth order. If the sixth order bridge $\mathcal{B}_6(\tau)$ would be chosen, the compound method would still be fourth-order (constrained by the fourth-order sub-integrator). In such a hierarchical coupling, while formally the overall order of convergence of the compound solver is limited by the lowest order, locally sub-systems being evolved with higher-order are still integrated at this higher order. This can be advantageous if, for example, the subsystem dominates in the overall error.

The disadvantage of a higher-order scheme is the requirement for each of the sub-integrators to be time-reversible. Tree-codes are not intrinsically time-symmetric and it is hard to make them time symmetric [34], but for shared time-step higher order schemes this does not pose a severe limitation (see for example in Section 5.3).

Table 1

Coefficeints for the higher-order symplectic bridge implementations, see [31,32].

S2M2		S4M5	
u_0	1/2		
v_0	1		

S4M4		S4M5	
u_0	$(642 + 471^{1/2})/3924$	$(14 - 19^{0.5})/108$	
u_1	$121(12 - 471^{1/2})/3924$	$(20 - 7 \cdot 19^{0.5})/108$	
u_2	$1 - 2(u_0 + u_1)$	$1/2 - (u_0 + u_1)$	
v_0	6/11	2/5	
v_1	$1/2 - v_0$	$-1/10$	
v_2		$1 - 2(v_0 + v_1)$	

S4M6	
u_0	0.0792036964311957
u_1	0.353172906049774
u_2	−0.0420650803577195
u_3	$1. - 2(u_0 + u_1 + u_2)$
v_0	0.209515106613362
v_1	−0.143851773179818
v_2	$0.5 - v_0 - v_1$

S6M11		S6M13
w_0	0.213755839458782545555518066964857	0.138619308540516952458080013042625
w_1	0.183293814074257139113859744251217	0.133465628510747604070468588322209
w_2	0.176928194730989437948988117099929	0.130705310114492251905427555785015
w_3	-0.443290826811702158496228296626258	0.129618937569070347725053666537091
w_4	0.117285604328659353854035856691364	-0.350003248939208965161708309113233
w_5	0.504054748438027364048327817142393	0.118055306530023871702734389540493
w_6		0.399077515348715874599887955200653

S8M21		S10M35
w_0	0.1064772898455003182393196785489631	0.0787957225216864192639076793377684
w_1	0.108374086458357263974334105915463	0.313096103415108527764812471926471
w_2	0.353378210526543424195345413240801	0.0279183832350780661095202732752997
w_3	-0.233414140231650821987802811283194	-0.229592841593907094151213396796553
w_4	-0.244452667915288412694621714132163	0.130962061077164863174656859279611
w_5	0.113178484357556331470095251559937	-0.269733405654510714344609732224114
w_6	0.118929056250003500626929722839513	0.0749733431558914356661371056414109
w_7	0.126039123218259881403056702683657	0.111993423999810204889575080736405
w_8	0.125817187361760418043923916415877	0.366133449546226751193148123531505
w_9	0.116991350192176421807228814335331	-0.399105630136035897878629810583403
w_{10}	-0.3826359601264366535094467074404041	0.103087398527471077313580277001372
w_{11}		0.411430873955890237820704118976082
w_{12}		-0.00486663605831352617621956593099771
w_{13}		-0.392033353708639906448081936426101
w_{14}		0.0519425029624496470371829040159761
w_{15}		0.0506650907599244963358743441568662
w_{16}		0.0496743706397298790545688002794613
w_{17}		0.0493177357595945379176800083393381

3. Adaptations to bridge

Sometimes coupling existing methods require additional non-canonical operations inside the bridge operator. This happens when the coordinate system changes at run time or if one desires to adopt specific boundary conditions, which may change at run time.

3.1. Introducing boundary conditions or an expanding coordinate system

Changing the coordinate system and invoking periodic boundary conditions at run-time are common in cosmological simulations to mimic the expansion of the Universe while limiting the computational domain. Instead of incorporating these in the physics solver directly, we may want to implement them in the operator. The same procedure is then ported from the *N*-body code to the bridge, for example by introducing periodic boundary conditions in the particle's coordinate system

$$\dot{\mathbf{r}} = \mathbf{v}, \tag{12}$$

$$\dot{\mathbf{v}} = \left(\frac{\ddot{a}}{a} + \mathcal{H}\right)\mathbf{r} + \mathbf{g}, \tag{13}$$

Here $\mathcal{H}$ could be zero or it could represent the Hubble flow $\mathcal{H} = 0.5\frac{H_0^2\Omega_m}{s^3}$, where the scale factor s, Hubble constant H_0 and matter density Ω_m have their usual meaning, and $\mathbf{g}$ is Newtonian gravity. A cosmological bridge can then be constructed by including the terms $\mathbf{r}\ddot{a}/a$ in the kick operator. One argument for adopting BRIDGE to include such boundary conditions or phase-space alterations is the convenience of using an unaltered integrator for the drift operator. The performance is not an issue here because the cosmological terms are not expensive to calculate and the drift operation scales linearly with the number of particles in the simulation.

3.2. BRIDGE In rotating reference frames

Another class of problems require solving on a non-inertial frame of reference, but, for example, in a rotating frame of reference. We encountered such a situation when integrating the equations of motion of a star cluster in orbit around the Galactic center. In that case, we are interested in the stimulated evaporation of a star cluster due to the non-radial structure in the Galactic potential. This potential is implemented as a semi-analytic background with a non-radial structure in the form of spiral arms and a bar. The bar and spiral arms in the model rotates as rigid bodies with a particular pattern speed [35]. As a consequence, the potential associated with the various components in the Galaxy model depend on time. Instead of having the Galaxy rotate, we may opt for implementing the rotation in the coupling pattern. This results in better energy conservation for the same time-steps size. [36].

We formulate a BRIDGE for a rotating frame of reference, such that the interactions between the stellar systems and the terms in the equations of motion arising from the non-inertial terms are bridged (The latter is convenient because it allows the integral to be formulated for an inertial frame of reference without adjustments).

The rotating BRIDGE is derived by considering a particle of mass m located in a frame that rotates around the z-axis with constant angular speed Ω. The Hamiltonian of this particle is then

$$H = \frac{||\mathbf{p}||^2}{2m} + U_{\text{gen}}(\mathbf{r}, \mathbf{p}), \tag{14a}$$

$$H = \frac{||\mathbf{p}||^2}{2m} + U_{\text{ext}}(\mathbf{r}) - (\Omega \times \mathbf{r}) \cdot \mathbf{p} - \frac{1}{2}m||\Omega \times \mathbf{r}||^2. \tag{14b}$$

Here $\mathbf{r}$ and $\mathbf{p}$ are the position and momentum vectors of the particle in the rotating frame. The term $U_{\text{ext}}(\mathbf{r})$ is the potential energy due to an external force, which depends only on the position of the particle. For example, $U_{\text{ext}}(\mathbf{r})$ represents the galactic potential. The last two terms in Eq. 14b correspond to a potential energy which accounts for the centrifugal and Coriolis forces. The energy associated to the centrifugal and Coriolis forces together with $U_{\text{ext}}(\mathbf{r})$, represents the total generalized potential energy of the particle, $U_{\text{gen}}(\mathbf{r}, \mathbf{p})$.

Here $U_{\text{gen}}(\mathbf{r}, \mathbf{p})$ depends on the momentum of the particle. Therefore, it is not possible to split the above Hamiltonian to obtain the drift and kick operators as we demonstrated in Eqs. 5a and 5b. We construct a rotating BRIDGE integrator by splitting the equations of motion of a particle in such a way that it satisfies Eq. 14a.

There are two ways to construct a rotating BRIDGE. One of them we call canonical and the other the non-canonical [see also 37]. In Section 3.2.1 and 3.2.2 we explain these approaches in more detail.

3.2.1. Canonical approximation

In this approach, the equations of motion of a particle moving in a rotating frame, are defined in terms of the canonical coordinates ($\mathbf{Q}$, $\mathbf{P}$). The canonical momentum ($\mathbf{P}$) is defined as

$$\mathbf{P} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{Q}}} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}} = \mathbf{p} + m(\Omega \times \mathbf{r}). \tag{15}$$

Here $\mathcal{L}$ is the Lagrangian $\mathcal{L} = \dot{\mathbf{Q}}\mathbf{P} - H$. The canonical momentum can be interpreted as the velocity of the particle seen in the inertial reference frame which is coaxial to the rotating frame of reference. By using Eq. 15 we can obtain the canonical momenta in Cartesian components:

$$P_x = p_x - m\Omega y,$$
$$P_y = p_y + m\Omega x,$$
$$P_z = p_z. \tag{16}$$

The equations of motion of a particle that satisfies Eq. 14a can then be written in terms of the canonical moment as follows:

$$\dot{x} = p_x/m + \Omega y; \qquad \dot{p}_x = F_x + \Omega P_y,$$
$$\dot{y} = p_y/m - \Omega x; \qquad \dot{p}_y = F_y - \Omega P_x,$$
$$\dot{z} = p_z/m; \qquad \dot{p}_z = F_z. \tag{17}$$

Here $\mathbf{F}$ is the external force associated to $U_{ext}(\mathbf{r})$. We proceed to split Eqs. 17 to build the kick and drift operators. The set of equations that represent the kick operator $K(\tau)$ is

$$
\begin{aligned}
\dot{x} &= \dot{y} = \dot{z} = 0, \\
\dot{p}_x &= F_x + \Omega P_y, \\
\dot{p}_y &= F_y - \Omega P_x, \\
\dot{p}_z &= F_z.
\end{aligned}
\tag{18}
$$

The solution of these equations is :

$$
\begin{aligned}
v_x(t + \tau) = {} & \left[v_x(t) - \left(\frac{a_y + \Omega^2 y}{\Omega} \right) \right] \cos(\Omega\tau) \\
& + \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{\Omega} \right) \right] \sin(\Omega\tau) \\
& + \frac{a_y + \Omega^2 y}{\Omega},
\end{aligned}
\tag{19a}
$$

$$
\begin{aligned}
v_y(t + \tau) = {} & -\left[v_x(t) - \left(\frac{a_y + \Omega^2 y}{\Omega} \right) \right] \sin(\Omega\tau) \\
& + \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{\Omega} \right) \right] \cos(\Omega\tau) \\
& - \frac{a_x + \Omega^2 x}{\Omega},
\end{aligned}
\tag{19b}
$$

$$
v_z(t + \tau) = v_z(t) + a_z \tau.
\tag{19c}
$$

Here a_x, a_y and a_z are the acceleration on a particle in the various cartesian coordinates, x, y and z, respectively. The drift operator $D(\tau)$ in the canonical approximation is represented by the following set of equations:

$$
\begin{aligned}
\dot{p}_x &= \dot{p}_y = \dot{p}_z = 0, \\
\dot{x} &= p_x/m + \Omega y, \\
\dot{y} &= p_y/m - \Omega x, \\
\dot{z} &= p_z/m.
\end{aligned}
\tag{20}
$$

The solution of these equations is

$$
\begin{aligned}
x(t + \tau) = {} & \left[x(t) - \frac{v_y}{\Omega} \right] \cos(\Omega\tau) \\
& + \left[y(t) + \frac{v_x}{\Omega} \right] \sin(\Omega\tau) + \frac{v_y}{\Omega},
\end{aligned}
\tag{21a}
$$

$$
\begin{aligned}
y(t + \tau) = {} & -\left[x(t) - \frac{v_y}{\Omega} \right] \sin(\Omega\tau) \\
& + \left[y(t) + \frac{v_x}{\Omega} \right] \cos(\Omega\tau) - \frac{v_x}{\Omega},
\end{aligned}
\tag{21b}
$$

$$
z(t + \tau) = z(t) + v_z \tau.
\tag{21c}
$$

The canonical formulation has two advantages: it generates a stable algorithm and this approximation is symplectic (see Fig. 2). However, for systems with interacting particles, it is convenient to have a drift operator that is independent of Ω. This is not the case for the canonical formulation (although this can be remedied by further splitting the operator).

3.2.2. Non-canonical approximation

In the non-canonical approximation, the motion of a particle is defined in terms of its position and velocity coordinates $(\mathbf{r}, \mathbf{v})$. Given the generalized force $\mathbf{F}_{gen} = m\mathbf{a} - m\Omega \times (\Omega \times \mathbf{r}) - 2m(\Omega \times \mathbf{v})$, the equations of motion of a particle in a rotating frame can be written as:

$$
\begin{aligned}
\dot{x} &= v_x; & \dot{v}_x &= a_x + \Omega^2 x + 2\Omega v_y, \\
\dot{y} &= v_y; & \dot{v}_y &= a_y + \Omega^2 y - 2\Omega v_x, \\
\dot{z} &= v_z; & \dot{v}_z &= a_z.
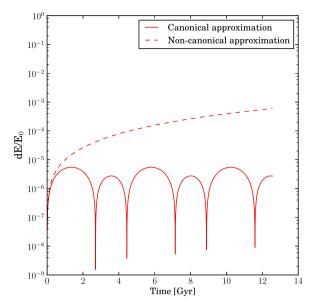\end{aligned}
\tag{22}
$$

Fig. 2. Energy error as a function of time for a particle in a rotating frame. Here we use a second order rotating BRIDGE with a time step of 1 Myr.

We split Eqs. 22 to build the kick and drift operators. The set of equations that represent the kick operator $K(\tau)$ is the following:

$$\dot{x} = \dot{y} = \dot{z} = 0,$$
$$\dot{v}_x = a_x + \Omega^2 x + 2\Omega v_y,$$
$$\dot{v}_y = a_y + \Omega^2 y - 2\Omega v_x,$$
$$\dot{v}_z = a_z. \tag{23}$$

The solution of these equations give expressions for the kick velocity of the particle in the three Cartesian coordinates at time $t + \tau$:

$$v_x(t + \tau) = \left[v_x(t) - \left(\frac{a_y + \Omega^2 y}{2\Omega} \right) \right] \cos(2\Omega\tau)$$
$$+ \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{2\Omega} \right) \right] \sin(2\Omega\tau)$$
$$+ \frac{a_y + \Omega^2 y}{2\Omega}, \tag{24a}$$

$$v_y(t + \tau) = -\left[v_x(t) - \left(\frac{a_y + \Omega^2 y}{2\Omega} \right) \right] \sin(2\Omega\tau)$$
$$+ \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{2\Omega} \right) \right] \cos(2\Omega\tau)$$
$$- \frac{a_x + \Omega^2 x}{2\Omega}, \tag{24b}$$

$$v_z(t + \tau) = v_z(t) + a_z\tau. \tag{24c}$$

Here the vector **a** corresponds to the acceleration of the particle due to the external galactic potential $U_{ext}(\mathbf{r})$.
The drift operator $D(\tau)$ on the other hand, is represented by the following set of equations:

$$\dot{v}_x = \dot{v}_y = \dot{v}_z = 0,$$
$$\dot{x} = v_x,$$
$$\dot{y} = v_y,$$
$$\dot{z} = v_z. \tag{25}$$

The solution of these equations for the kick velocity of the particle in the three Cartesian coordinates is simpler than in the canonical case (Eqs. 24)

$$x(t + \tau) = x(t) + v_x(t + \tau/2)\tau,$$ (26a)

$$y(t + \tau) = y(t) + v_y(t + \tau/2)\tau,$$ (26b)

$$z(t + \tau) = z(t) + v_z(t + \tau/2)\tau.$$ (26c)

By using the non-canonical approximation, a second order rotating BRIDGE can be constructed as follows

$$\mathcal{B}_2^r(\tau) = K(\tau/2) \cdot D(\tau) \cdot K(\tau/2),$$ (27)

where the operators $K(\tau)$ and $D(\tau)$ are described by Eqs. 24 (see Eq. 19 for the canonical case) and 26 (see also Eq. 21) respectively. Consequently, every $\tau/2$ a star receives a velocity kick due to the external galactic potential and the position of the star is updated every τ.

The rotating BRIDGE can be generalized to a system of self interacting particles. The Hamiltonian of a stellar system A which is located in a frame that rotates around the z-axis with constant angular speed Ω is given by:

$$H = H_A + H_{int},$$ (28)

where

$$H_A = \sum_{i \in A}^{N_A} \frac{||\mathbf{p}_i||^2}{2m_i} - \sum_{i \neq j \in A}^{N_A} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||},$$

$$H_{int} = \sum_{i \in A}^{N_A} \left[U_{ext}(\mathbf{r}_i) - (\Omega \times \mathbf{r}_i) \cdot \mathbf{p}_i - \frac{1}{2} m_i ||\Omega \times \mathbf{r}_i||^2 \right].$$ (29)

The temporal evolution of the system in a second order approximation is given by Eq. 27, which can be written as a kick-drift-kick operation:

$$e^{H_{int}\tau/2} e^{H_A\tau} e^{H_{int}\tau/2}.$$ (30)

Here the term $e^{H_{int}\tau}$ represents the kick operator $K(\tau)$ while the term $e^{H_A\tau}$ corresponds to the drift operator $D(\tau)$. For a system of self-interacting particles the drift operator is given by

$$\mathbf{x}(t + \tau) = \mathbf{x}(t) + \mathbf{v}(t + \tau/2)\tau,$$ (31a)

$$\mathbf{v}'(t + \tau/2) = \mathbf{v}(t + \tau/2) + \mathbf{a}\tau/2.$$ (31b)

The evolution of system A during a rotating BRIDGE time step τ is composed of the following steps:

i) At $\tau/2$ the system receives a velocity kick due to the external potential of its parent galaxy (Eqs. 24). This velocity is referred to as $\mathbf{v}(t + \tau/2)$.

ii) The positions of the stars are updated for a time step τ (Eq. 31a). In addition, the velocities of the stars are updated once more after evolving system A by means of direct N-body integration (Eq. 31b).

iii) The system receives a velocity kick for another $\tau/2$ (Eq. 24). This kick is computed by using the previous velocity $\mathbf{v}'(t + \tau/2)$. Note that the only difference between the rotating and classical BRIDGE is in the kick operator. In particular, the code evolution operator $e^{H_A\tau}$ does not have to be changed.

The above procedure can also be applied to a more generalized case in which there are several self-gravitating systems.

The precision of the rotating BRIDGE can be improved by applying the drift and kick operators in accordance with Eqs. 10 and 11 as we discussed in Section 2.3. A higher-order rotating bridge can be constructed similarly. In Section 5.5 we present an example of a rotating bridge.

In Fig. 2 we present the energy error resulting from integrating a test particle in a background potential of a Milky Way-like galaxy with two spiral arms. One calculation is performed with the canonical approximation Section 3.2.1 and the other with the non-canonical approximation Section 3.2.2. The former (canonical case) is shown to behave symplectic in terms of energy conservation, whereas the non-canonical case is not symplectic.

3.3. BRIDGE Expansions for a particular subset

In some cases, one would like to apply a certain force or operation to a specific subset of particles. For example, in the Yarkovsky-O'Keefe-Radzievskii-Paddack [YORP for short, 38] effect and the diurnal Yarkovsky effect [39] only the lightest pebbles are affected, whereas planets and other major bodies are not. One could solve this by integrating pebbles with a separate integrator in which a direct N-body code is bridged with the YORP calculation and another N-body code for

the planets. This, however, makes it harder to deal with inter-code operations, such as close encounters between planets and asteroids, tidal effects of collisions. With two separate *N*-body codes, detecting a collision can then be implemented in the bridge step, but this is generally longer than the internal integration time scale of any of the bridged sub-codes. This limitation can be overcome by reducing the bridge time step τ, but only at the cost of spending more time in the bridge, which is generally expensive compared to the optimized *N*-body codes (See also the discussion in chapter 4 of [40]). A better solution is to augment a subset of the bridged particles with the additional physics, whereas the other subset is not affected.

Here we demonstrate how to construct such an integrator. We provide the example in which a small subset of particles is affected by post-Newtonian dynamics, whereas the majority of the particles is integrated under the Newtonian approximation.

In order to extend BRIDGE to include post-Newtonian (PN) corrections a special treatment is needed to handle the velocity dependency in the PN terms. Here we adopt the recipes developed in [41], where an auxiliary velocity, $\mathbf{w}_i(t)$ with $\mathbf{w}_i(t = 0) = \mathbf{v}_i(t = 0)$, is introduced to make the time evolution operators separable and therefore allowing the use of the leapfrog algorithm for implementing the PN corrections.

In BRIDGE, the operators $e^{\tau H_{S_k}}$ and $e^{\tau H^{\text{int}}_{S_k S_l}}$ in Eqs. 5a and 5b can be associated with different solvers for each of the sub-systems in a simulation: For example, if the *j*-th sub-system requires PN corrections whereas interactions within the other sub-system can be treated classically. In that case, only the operator $e^{\tau H_{S_k}}|_{k=j}$ in Eq. 5a requires modifications, in the form of a simple substitution of a PN solver to the regular Newtonian solver for this particular sub-system *j*.

In the more complex case where several interacting sub-systems require PN corrections, both operators $D(\tau)$ and $K(\tau)$ in Eqs. 5a and 5b have to be modified accordingly. In the following, we will assume that from a total number Q of sub-systems being bridged, a number Q^N are "Newtonian", and a number Q^E require PN corrections so that $Q = Q^N + Q^E$. We define S^N as the set of "Newtonian" sub-systems and S^E as the set of "post-Newtonian" sub-systems, so that $S = S^N \bigcup S^E$. In this way, operator $D(\tau)$ in Eq. 5a can be extended into the following expression

$$\tilde{D}(\tau) = \prod_{k \in S^N}^{Q^N} e^{\tau H_{S_k^N}} \prod_{k \in S^E}^{Q^E} e^{\tau H_{S_k^E}}. \tag{32}$$

Here $e^{\tau H_{S_k^N}}$ represents a Newtonian solver for the *k*-th sub-system in S^N and $e^{\tau H_{S_k^E}}$ represents a PN solver for the *k*-th sub-system in S^E. With a PN solver we indicate a code that evolves particles under the total acceleration $\mathbf{a} = \mathbf{a}^N + \mathbf{a}^E$, rather than just $\mathbf{a}^E$ as might be implied by our notation.

The operator $K(\tau)$ in Eq. 5b can be extended as follows. We first define some auxiliary kick operators,

$$K^{N \leftrightarrow N}(\tau) = \prod_{k \neq l}^{Q^N} e^{\tau H^{\text{int}}_{S_k^N S_l^N}}, \tag{33}$$

$$K^{N \leftrightarrow E}(\tau) = \prod_{k \in S^N}^{Q^N} \prod_{l \in S^E}^{Q^E} e^{\tau H^{\text{int}}_{S_k^N S_l^E}}, \tag{34}$$

$$K^{E \leftrightarrow E}(\tau) = \prod_{k \neq l}^{Q^E} e^{\tau H^{\text{int}}_{S_k^E S_l^E}}, \tag{35}$$

from which follows that the extended kick operator for BRIDGE with PN corrections can be written as (see also Eq. 52):

$$\begin{aligned} \tilde{K}(\tau) &= K^{N \leftrightarrow N}(\tau/2) \cdot K^{N \leftrightarrow E}(\tau/2) \cdot K^{E \leftrightarrow E}(\tau) \cdot K^{E \leftrightarrow N}(\tau/2) \cdot K^{N \leftrightarrow N}(\tau/2) \\ &= K^{E \leftrightarrow E}(\tau/2) \cdot K^{E \leftrightarrow N}(\tau/2) \cdot K^{N \leftrightarrow N}(\tau) \cdot K^{N \leftrightarrow E}(\tau/2) \cdot K^{E \leftrightarrow E}(\tau/2). \end{aligned} \tag{36}$$

Eq. 33 represents the Newtonian kick due to the interaction between "Newtonian" sub-systems, and is identical to the original definition in Eq. 5b. Eq. 34 represents the kick due to the interaction between "Newtonian" and "post-Newtonian" sub-systems. In this particular case, a choice has to be made on whether or not PN corrections should be included. Such a decision could be based, for example on the distance between the two interacting subsystems or their masses. Eq. 35 represents the PN kick due to the interaction between "post-Newtonian" sub-systems. Finally, Eq. 36 represents the extended kick operator to be used in BRIDGE with PN corrections.

4. Extension of the construction strategies of BRIDGE

4.1. Extending notation and terminology: The hierarchical coupling language

To accommodate an abstraction to our understanding and discussion on bridge topologies, we introduce a meta description language of the coupling patterns using BRIDGE. We, therefore, introduce a notation which helps us to think in more

abstract terms about the various coupling strategies. For each of the possible adaptations, we suggest a notation, examples are given in Section 5.2–Section 5.6.

In the classic implementation, Fujii et al. [14] constructed a bridge between a direct *N*-body code and a tree-code. In our nomenclature, we called these codes *S* for the microscopic system and *G* for the macroscopic system. We indicate the order of the integrator using an integer subscript, 2 for 2nd order, 4 for 4th order, etc. The bi-directional coupling strategy using a 2nd order scheme for the coupling, discussed in Section 2.2, is then written as

$$[S_4 \underset{2}{\longleftrightarrow} G_2]. \tag{37}$$

A second-order bridge is appropriate here because the lowest order in the hierarchy is also second order. Using a 4th bridge puts specific constraints on the choice of integrators, but will lead to better energy conservation. In future notation, we tend to omit the order of the bridge when it is the same as the lowest of the two coupled codes. An example of the classic bridge is given in Section 5.2.

Higher accuracy would be acquired in the previous calculations by adopting an individual time-step direct *N*-body scheme for the global system. This would motivate the use of a fourth-order bridge

$$[S_4 \longleftrightarrow G_4], \tag{38}$$

or even 6th order for the subsystem

$$[S_6 \longleftrightarrow G_4]. \tag{39}$$

Note that here both bridges are implicitly assumed to be of 4th order. In Section 5.3 we present an example of the higher-order bridge coupling.

One could replace the numerical integration by semi-analytic a potential.

$$[S_4 \underset{2}{\longleftarrow} G]. \tag{40}$$

Here the global system has no order because it uses a semi-analytic potential on which particles float, but they are integrated using the 2nd-order bridge. We only presented the bridge with an arrow in one direction, indicating the single direction of the hierarchical coupling: the subsystem is affected by the global system, but not vice versa. The solution is subsequently not self-consistent, but the calculation will be fast.

4.2. Augmented coupling strategies

The effect of the global system can still be taken into account by including a correction term to the bridge. This additional term comes in the form of a function *f*, which depends on the characteristics of the subsystem and the global system. We write the one-directional bridge with a one-directional support function as

$$[S \underset{f}{\longleftarrow} G] \tag{41}$$

Here the function $f \equiv f(S, G)$ below the left-pointed arrow represents a function that operates on the subsystem (*S*) with additional information about the global system (*G*). It could, for example, calculate the effect of dynamical friction on the cluster *S* due to its evolution in the Galaxy *G*. In that case, the function *f* would contain the dynamical term exerted on the orbit of a star cluster due to its interaction with the Galactic field stars. The function *f(S, G)* is rather general and can include terms for post-Newtonian correction, the Yarkovsky effect or gas drag. We present an example of the mono-directional bridge coupling in Section 5.4.

A special case of an augmented bridge is the rotating bridge. In that case, the function *f* describes the consequences of a rotating frame of reference, as we discussed in Section 3.2. An example is provided in Section 5.5.

4.3. Compound hierarchical coupling strategies

Instead of adding an interacting function, a new bridge could be declared to generate a compounded bridge. We could add another sub-system to the already present system. The potential advantage of a cascade of subsystems is illustrated by studying the dynamics of planetary systems of stars in a cluster that is part of a galaxy. Maybe there are even multiple clusters that interact as part of the same galaxy. For clarity, we indicate the planetary integrator with (P, for picoscopic system), the star cluster with (S) and the galaxy with (G). One way to address such a hierarchical coupling strategy can be written as

$$[[P_6 \longleftrightarrow S_4] \longleftrightarrow G_2]]. \tag{42}$$

In this example, the planets (P) and stars (S) integrators are coupled with a 4th-order scheme, whereas the stars in the cluster couple to the galaxy using a 2nd-order scheme. To allow the Galactic potential to affect the star cluster but not vice versa one could opt for the more efficient, but inconsistent coupling

$$[[P_6 \longleftrightarrow S_4] \longleftarrow G_2]]. \tag{43}$$

This scheme can be further expanded hierarchically, for example by adding a lunar system (M, for minimicroscopic) to one of the planets.

$$[M \longleftarrow [P \longleftrightarrow [S \longleftarrow G]]]. \tag{44}$$

In this case, the moons (M) are affected by the planets (P), the stars (S) and the Galaxy (G), but the Galaxy is unaware of the moons, planets or stars. Depending on the requirements for accuracy and speed, there are several ways in which such a hierarchical compound system can be constructed, the adopted topology depends on the underlying scientific question. The topology could be democratic

$$[[M \longleftarrow P] \longleftrightarrow [S \longleftarrow G]], \tag{45}$$

or have a reversed hierarchy

$$[[[M \longleftarrow P] \longleftrightarrow S] \longleftarrow G]. \tag{46}$$

The various hierarchies and direction of the coupling makes the technique versatile and flexible. In particular if the implementation allows changes to the topology at run-time.

4.4. Bridging of a specific subset

The notation can be further expanded for compound systems in which one part (S) is affected in a different way than a subset (S'),

$$[S \longleftarrow G, S' \underset{f}{\longleftarrow} G]. \tag{47}$$

In this case, $S \subset S'$, but this is not necessarily relevant, in which case one could write:

$$[(S - S') \longleftarrow G, S' \underset{f}{\longleftarrow} G]. \tag{48}$$

In Section 3.3 we presented the technical implementation and illustrated it for the use for including post-Newtonian corrections in a purely Newtonian N-body solver.

The possibility of declaring specific subsets, combined with hierarchical couplings, higher-order, and augmentations make for a powerful non-intrusive tool in which complicated simulation environments can be constructed. In Section 5.6 we present an example of such a compound hierarchical implementation for simulating planetary systems in star clusters.

5. Implementation, validation, verification and demonstration

5.1. Implementation in the Astrophysical Multipurpose Software Environment

The wide variety of Bridge coupling strategies can be implemented in several ways. Rigid implementations have been introduced to simulate planetary systems [42], star clusters [43] and galactic nuclei [18]. We illustrate our implementation as we have realized in the Astrophysical Multipurpose Software Environment (Amuse) [40,44,45] and in Omuse [46].

Amuse is a software environment for astrophysical simulations written in multiple languages but the fundamental structure is based on Python [see 47]. Amuse presents a wide variety of astrophysical codes using homogeneous interfaces, simplifying their use. Omuse is the equivalent of Amuse but then for simulating seas and oceans on the surface of a celestial body.

The Bridge method has turned into an essential part for the proper operation of Amuse and Omuse which enables us to couple with other codes from different domains. Amuse includes a number of hydrodynamics and gravitational dynamics codes, for Omuse the coupling is similar.

For the Bridge integrators here, it is important to note that the gravitational dynamics codes in Amuse provide convenient implementations of the evolution operator $e^{\tau H}$ in the form of an `evolve_model` method on the interface. This can be combined with simple force evaluations (by using another component code or implemented on the interface level) to provide $e^{\tau H_{\text{int}}}$, the interaction operators. The integrators presented above can be quickly formulated using ready-made 'building blocks,' much in the Duplo philosophy advocated in [48]. It supports the selection of the appropriate integrator from a wide variety of implementations to balance the needs for the precision, accuracy, and performance or by focusing on some specific characteristics of a particular implementation.

In the following sections we present several examples of bridged schemes as they may appear in astrophysics. We complement these tests with validation.
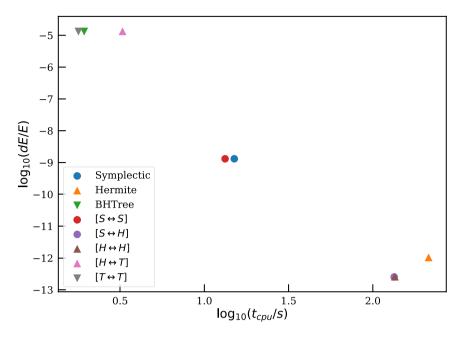
Fig. 3. Performance in terms of wall-clock time (x-axis) and energy conservation (y-axis) for a selection of bridges and codes. In these cases, it is not the bridge that gives rise to the energy error, but the codes that are coupled. The codes we coupled included the 2nd-order symplectic integrator `Huayno` (indicated as Symplectic and the letter S in the legend), the 4th-order Hermite predictor-corrector integrator (indicated with Hermite and H) and the Barnes-Hut tree code (BHTree and T). The bridged schemes adopt two integrators in a bi-directional coupling (see Section 4.1). The lowest order integrator determines the overall energy error. The initial conditions for the calculations ware a star cluster of 1000 stars and a total mass of 600 $M_{\odot}$ in a 10 pc Plummer sphere orbiting in a 10^{11} $M_{\odot}$ life Galaxy composed of 10^4 equal-mass stars distributed in a standard bulge-disk-halo structure with a characteristic radius of 10 kpc generated using the `Galaxia` [50] software in AMUSE. The globular cluster was put in a circular orbit at a distance of 5 kpc from the Galactic center. We adopted a bridge time-step of 0.1 Myr and integration lasted for 100 Myr for each of the simulations.

5.2. Example 1: The classic bridge: $[S_4 \longleftrightarrow G_2]$

```
(1) code1=Hermite()
(2) code1.particles.add_particles(cluster)
(3) code2=BHTree()
(4) code2.particles.add_particles(galaxy)
(5) sys=Bridge(timestep=0.1 | units.Myr)
(6) sys.add_system(code1, (code2,))
(7) sys.add_system(code2, (code1,))
(8) sys.evolve_model(10 | units.Myr)
```

To illustrate the bi-directional bridge, in snippet 1 we show the usage of the BRIDGE integrator through the AMUSE framework. In this example, we present the steps needed to evolve a star cluster (contained in the cluster particle set[1]) in its parent galaxy (the galaxy set). The initial realization can be constructed within AMUSE using e.g. a Plummer sphere [49] model. In line number (1) we initialize the *N*-body integrator to calculate the internal evolution of the star cluster. In this case, we use the Hermite integrator. In line (2) We send the particle data to the *N*-body code. (3) and (4) similarly a code appropriate for the galaxy model (in this case a tree-code) is started and initialized. (5) We instantiate the BRIDGE integrator, setting a time step for the coupling timescale. We couple the cluster code and the galaxy into BRIDGE in lines (6) and (7). The method `add _system` has two arguments: the main *N*-body realization (called the *system*) and a set with *interaction partners*. The *interaction partners* indicate which systems will kick the *system*. Therefore, in line (6) the galaxy will kick the particles in the cluster code. In line (7) the particles in the cluster code will kick the galaxy. In this way, we ensure that both cluster and galaxy are evolved self-consistently. Finally, in line (8) the compound system is evolved for a particular time frame.

The coupling strategy provides a symplectic time evolution. We demonstrate this in Fig. 3, were we measure the energy conservation of several integrations using BRIDGE. This example is for illustrative purposes, but it is apparent that in all cases

---

[1] A "Particle set" is AMUSE nomenclature, indicating the collection of individual objects that are subject to the same force law.
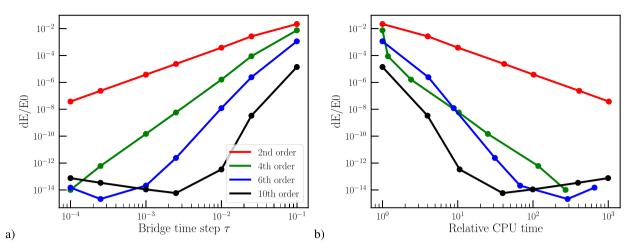
Fig. 4. a) Energy error for integrating two binaries in orbit around each other using a bridge. The lines give the relative energy error of the integration as a function of the bridge time step τ in dimension-less *N*-body units. From top to bottom, the lines red, green, blue and black give the relative energy error for a second, fourth, sixth and tenth order Bridge coupling (see also the legend in Fig. 5). b) CPU time for the same calculations as in the other panel. The CPU time is normalized to the lowest accuracy calculations, because for the adopted size of the system the overhead of the bridging method is large compared to the actual time spent in the integrators. To prevent clutter, we present the legend in the inset in the left-hand panel of Fig. 5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the bridged methods requires less computer time at the same energy conservation. When the underlying bridged systems become progressively more complicated and expensive in terms of computer time, the relative speed increase of the bridge improves whereas the energy conserving characteristics remains roughly the same. This is just a small non-optimal example of the working of a bridge. In this case, it indicates that the speed does not necessary affect the energy conservation. The relative efficiency in terms of performance and accuracy depends in various factors such as the topology of the system, the selected codes and the tuning parameters in each of the individual codes. It goes too far for this paper to explore those parameters, as they should be tuned for each individual problem separately. Regretfully, we have to ready solution yet for the optimal choices of these parameters.

The various measurements presented in Fig. 3 indicate that the error in the energy is dominated by the lowest-order methods adopted in the bridge. The energy errors introduced in the high-order integrator of the microscopic system is negligible compared to the error introduced by the low-order integrator used for the macroscopic system. Also the error in the energy introduce by the bridge method itself is negligible. The choice of parameters does not emphasize the speedup introduce by the splitting method, simply because most of the work in these cases is done in the macroscopic system. Further fine-tuning will allow the user to make a trade-off between speed and accuracy, depending on the specific requirements of the problem.

The formulation is not limited by two sub-systems, but can be composed of an arbitrary number of subsystems. Multiple sub-systems can be integrated using different specialized solvers and bridged either using a Hamiltonian splitting technique (as we discuss in section 2.3) or by applying the above splitting scheme recursively [this was done in 51]. Besides, multiple microscopic systems may be nested hierarchically, in which case the macroscopic system can interact with all or only with a subset of the microscopic systems.

5.3. Example 2: Higher order bridge: $[S_6 \longleftrightarrow G_4]$, $[S_8 \longleftrightarrow G_6]$, etc.

In this example, we calculate the evolution of a stable hierarchical quadruple system consisting of two binary stars that orbit each other. The total system comprises of 4 equal-mass bodies [for a total mass of 1 in *N*-body units, 52], and the orbits are co-planar with the two binary orbits having a semi-major axis $a = 1/8$ and moderate eccentricity ($\epsilon = 0.5$). Each binary is set-up in an orbit with $a = 1$. and $\epsilon = 0.5$. Each binary is integrated with separate codes and their interaction is bridged using the 2nd-order Bridge, a fourth-order, sixth-order, and a tenth-order Bridge. The binaries themselves are evolved using a Kepler solver, which calculates the evolution of the subsystems to machine precision. As a consequence, the measured error is caused by the Bridge at least down to a relative energy error of $dE/E \sim 10^{-15}$.

In Fig. 4, we present the relative energy error as a function of the bridge time step τ for a series of identical experiments. The setup is a quadruple star system composed of two binaries that orbit each other. The two binaries are integrated using a Kepler solver that, apart from round off, does not produce any errors. The Bridge schemes give the expected error behavior, appropriate for their respective order (Note that the sixth and tenth order integrators show saturation around the machine precision, as expected. Improved precision beyond the common ~ 16 decimal places and $\sim 10^{-12}$ energy conservation per
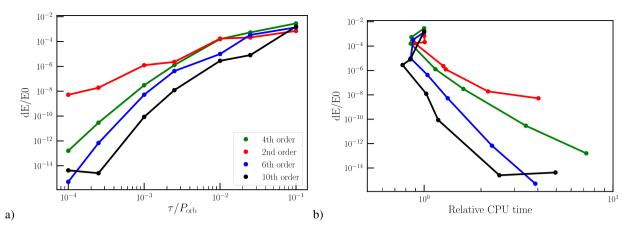
Fig. 5. a) Energy error for two orbiting clusters clusters of 64 particles. The two clusters are initialized with a King model [55]. Plotted is the energy error as a function of the BRIDGE time step τ (given as a fraction of the orbital period of the two clusters). The order of the BRIDGE is given by the colors, red, green, blue and black for $\mathcal{B}_2(\tau)$, $\mathcal{B}_4(\tau)$, $\mathcal{B}_6(\tau)$ and $\mathcal{B}_{10}(\tau)$ order BRIDGE operators. b) Energy error as a function of the relative CPU times for the same calculations. As in Fig. 4 (panel b), the CPU time is normalized to the lowest accuracy calculations, because for the adopted size of the system the overhead of the bridging method is large compared to the actual time spent in the integrators. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

step is hard and requires special treatment of the force evaluation and time-stepping, as is advocated in [53].). We performed the same series of experiments using a leapfrog [54] integrator (not show in Fig. 4), which at small time steps ($\lesssim 10^{-3}$) produces energy errors ~ 3 orders of magnitude higher than the Kepler solver: the behavior of the BRIDGE is washed out by the errors produced in the leapfrog integration of the subsystems. This test demonstrates that the BRIDGE scheme can be applied effectively to couple codes at higher-order.

In Fig. 5 we show the energy error for a similar test where we put two star clusters in orbit around each other. The models consist of King models with 64 particles, put on circular orbits with a separation of 8 *N*-body scale-length units. To speed-up the calculation, we introduce a softening length of 0.2 *N*-body scale-length units [56]. We integrate for half an orbital period (which is 284 *N*-body time units). The two models are integrated using a high precision sixth order method which are bridged using either $\mathcal{B}_2(\tau)$ (red), $\mathcal{B}_4(\tau)$ (green), $\mathcal{B}_6(\tau)$ (blue) and $\mathcal{B}_{10}(\tau)$ (black). For the 10th-order integrators it is hard to establish the order from the figure because the energy error decreases so steeply that we quickly run in the round-off error of the computer. Overall, however, the energy error shows the expected behavior Fig. 5. These high-order methods would benefit from using extended or arbitrary-precision arithmetic, such as is realized in the calclations in [53]. It appears, however, that for most applications (in astrophysics) double precision-arithmetic is sufficient [57].

Eventually, it is up to the reader to decide if spending more computer time is worth the effort on reducing the energy error. The right-hand panels in Figs. 4 and 5 we present the relative time spend in the various integrators. For the few particles in the examples, the higher-order integrators turn out to be considerably more expensive than the lower-order methods. When the number of particles (and therefore the number of operations) both sub-systems increases the relative overhead of adopting a higher-order integrator drops. In practice, it is probably the lowest order in the individual integrators that determines the choice for the order of the bridge integrator, rather than the computational cost.

5.4. Example 3: Augmented bridge to include a frictional force: $[S_4 \underset{f}{\leftarrow} G]$

In Section 3.3 we explained how supplementary forces can be included in the bridge by adding an extra term to the kick operator. In this way solving constructing an *N*-body solver that includes the YORP effect, tidal evolution or post-Newtonian dynamics becomes relatively straightforward. Here we demonstrate how such an augmentation to the BRIDGE operator can be used to add dynamical friction two independent solvers.

A star cluster in the Galactic center will feel a frictional force due to the accumulation of stars in its wake [58]. [59] used a direct integration technique to study the time scale on which the Arches star cluster sinks from its birth location at about 30 pc from the Galactic center towards the middle of the Milky Way. It turns out that the time scale depends quite sensitively on the Coulomb parameter $\ln \Lambda$ [60]. In their Fig.6 they present the results of several calculations for the distance to the Galactic center as a function of time for a star cluster of 10^6 M$_\odot$. In Fig. 6 we repeat their calculation but using an external function included in the BRIDGE step, which then becomes
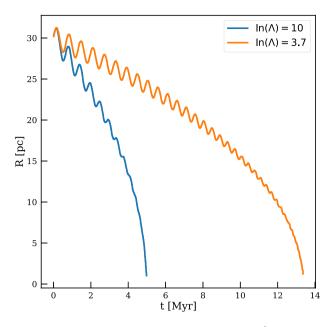
$$[S_4 \underset{f}{\leftarrow} G]. \tag{49}$$

Fig. 6. Time evolution of Galactocentric distance R of a star cluster having an initial mass $M = 10^6$ M$_\odot$. The rightmost curve represents a model with $\ln(\Lambda) = 3.7$; the leftmost curve has $\ln(\Lambda) = 10$. The sinusoidal variations along the curves result from the slightly eccentric orbits we adopted.

The equation to solve in $f(S, G)$ is Eq. 7 of [59].

```
(1) code1=Hermite()
(2) code1.particles.add_particles(cluster)
(3) code2=DynamicalFriction(Coulomb_logarithm=3.7)
(4) sys=Bridge(timestep=0.1 | units.Myr)
(5) sys.add_system(code1)
(6) sys.add_code(code2)
(7) sys.evolve_model( 10 | units.Myr )
```

In this example, we included the dynamical friction as an additional drag force in the kick of the bridging step. Alternatively, we could include a term in the drift-part of the bridge step.

5.5. Example 4: Rotating bridge: $[S_4 \overset{r}{\leftarrow} G_2]$

To evaluate the accuracy of the Rotating BRIDGE, we show in Fig. 7 the maximum fractional energy error as a function of bridge time step of a single star moving in different galactic potentials representing the Milky Way (axisymmetric, with bar, and including spiral arms). Although the microscopic system contains only a single star, integration of the subsystem is performed with a 4th order scheme. A consequence of only integrating a single particle warrants that the energy errors presented in Fig. 7 are solely caused by the bridge.

The axisymmetric potential was modeled by taking into account the parameters of [61]. The central bar was modeled with a Ferrers potential [62] and the spiral arms were modeled as perturbations of the axisymmetric Galactic potential following the tight winding approximation [63]. Both bar and spiral arms rotate as rigid bodies with different pattern speeds. For further details on the Galactic model, we refer the reader to [64].

We also show two different stellar motions. In the left panels, a single star moves through the Galaxy in a nearly circular orbit with an eccentricity of $\epsilon = 0.01$. In the right panels, the star moves in an eccentric orbit with $\epsilon = 0.5$. We computed both orbits for ten orbital periods. The circular and eccentric orbits have orbital periods corresponding to 224 and 471 Myr respectively.

We observe that the fourth and sixth-order methods have better energy conservation compared to the second-order rotating BRIDGE. In addition, the energy error is smaller for circular than for eccentric orbits for a given Galaxy model and BRIDGE time step. This is expected because the external tidal field is rather constant in a circular orbit, whereas for $e > 0$ it varies. In particular, in eccentric orbits, a high-order rotating BRIDGE can result in satisfactory energy conservation at small
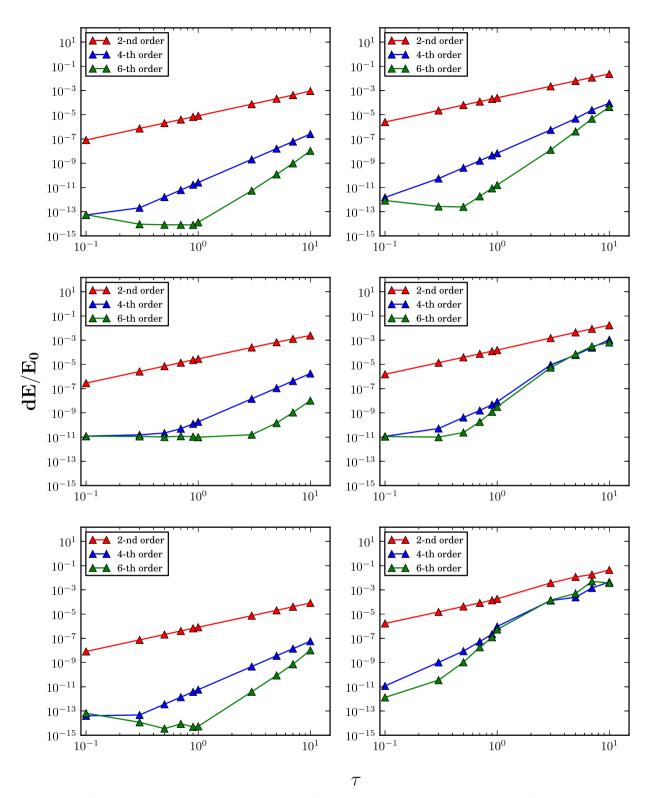
Fig. 7. Maximum fractional energy error as a function of Bridge time step of a star moving in a nearly circular orbit ($\epsilon = 0.01$, left panels) or in an eccentric orbit ($\epsilon = 0.5$, right panels). Presented is the Jacobi energy error, because this is the quantity conserved in rotating frame. The Milky Way is represented by: *Top:* A pure axisymmetric potential. *Middle:* Axisymmetric + bar potential. *Bottom:* Axisymmetric + spiral arms potential. The red lines correspond to the normal second order rotating BRIDGE. The blue and green lines correspond to the rotating BRIDGE in a fourth and sixth order. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

time steps ($dE/E_0 < 10^{-7}$ at $\tau \leq 1$ Myr), whereas 10^{-3} is probably sufficient [65]. The high-order rotating BRIDGE is suitable for computing the stellar motion in the adopted semi-analytic galactic background-potential.

5.6. Example 5: Compound hierarchical bridge: The NEMESIS integrator $[P_8 \longleftrightarrow [S_4 \leftarrow G_2]]$

BRIDGE operators can become rather elaborate, in particular if they are hierarchically nested and some operate on different subsets than others. The NEMESIS strategy,[2] a module in AMUSE, follows such a strategy.

It is composed of hierarchically nested BRIDGE systems in which the wide separation in scales allows us to separate the gravitational forces between the different levels of the hierarchy. The hierarchy is separated in a macroscopic system (the parent) and the microscopic system (the children). A parent can have multiple children, but each child has only one parent. The structure is hierarchical in the sense that a child can be a parent with multiple children of its own. We designed such a compounded hierarchical bridge for integrating multiple planetary systems in star clusters [66], but it is also used for integrating multiple star clusters in a Galaxy.

In NEMESIS, the entire star cluster is separated into one global structure and any number of sub-structures, which are coupled together with a cascade of BRIDGE patterns. The global structure is composed of individual stars and single planets, and integration can be realized with a direct *N*-body solver. Substructures are treated individually and, depending on their characteristics, integrated using a symplectic *N*-body solver.

Each planetary system can now be integrated with its own dedicated solver, and this is also the case for the star cluster. These integrators are dynamically created and deleted at run time, depending on whether or not a subsystem forms or dissolves.

If a planet escapes from its host star, it is picked up by the global *N*-body integrator whereas the other planets in that same system continue to be integrated with the dedicated solver. If a planetary system completely dissolves into individual unbound components, each star and planet in the subsystem is incorporated into the global *N*-body code and the subsystem integrator is terminated. If a new bound subsystem appears, a new (symplectic) *N*-body code to handle the local dynamics is started and incorporated using a BRIDGE to the global cluster code.

Asteroids can be taken into account by incorporating an additional BRIDGE operator with a test-particle integrator, much in the same way as was introduced in [67]. In NEMESIS, stars feel the gravitational force of all other stars and planets. Planets around a particular star feel the force of each other and of free-floating planets and the other stars in the cluster, but not the force of the planets in orbit around other stars. Asteroids, if included in the calculation, feel the force of all stars and free-floating planets in the cluster and the local planets, but not the force of other asteroids or planets around other stars.

NEMESIS is realized by incorporating four extensions to BRIDGE, these include:

- *topology* NEMESIS supports two types of systems, (1) the parent that provides the frame of reference and (2) children, each of which is a subsystem of its parent. Each child is represented as a single particle in the parent system.
- *creation/destruction* A child is created for every close encounter between two individual particles in the parent. If a single particle or a child encounters a child, it is absorbed by the larger child. If a child system contains a single particle, that particle becomes part of the parent and the particular integrator is stopped.
- *exchange* Particles can be exchanged between systems. A particle that moves too far from a child will be removed from that system and transferred to the parent. A particle in the parent that approaches a child system will be incorporated into the child and removed from the parent.
- *accounting* A database is maintained to keep track of all the particles in the parent system and those in the child systems. This database also keeps track of the running codes.

It is rather complicated to express the NEMESIS module in simple pseudo-code, but our notation is sufficiently expressive to show the fundamental structure of this compound hierarchical bridge. Here we write planets that belong to star S^i as $P(S^i)$. These planets feel the force of all the other planets in orbit around star *i*, but not the forces from the planets around other stars $S - S^i$. We first define the compound system

$$C(P_8, S_4) \equiv [P(S^i)_8 \longleftrightarrow S_4^i, (P(S - S^i))_8 \longleftrightarrow (S - S^i)_4]], \tag{50}$$

which we may want to abbreviate to $C(P_8, S_4) \equiv [P(S_4^i)_8, P(S_4 - S_4^i)_8]$. We can subsequently construct a classic bridge from the compound system $C(P_8, S_4)$ with the galaxy G_2 as usual.

$$[C(P_8, S_4) \leftarrow G_2]. \tag{51}$$

With this strategy, we create a complex topography of interacting codes. In its simplest form, NEMESIS requires 3 different gravity solvers:

- One code for each of the microscopic systems,
- One code to integrate the macroscopic system,
- One code to calculate all forces and coordinates the communication between the other codes.

---

[2] The NEMESIS module is named after the first application during development in which we studied a hypothetical companion to the Sun.
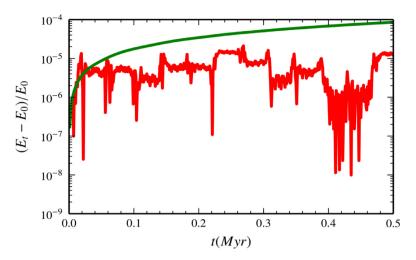
Fig. 8. Total energy error as a function of time for a validation simulation consisting of a wide binary of which one star is orbited by 5 planets (see [66]). The energy error of the Nemesis method (in red) is compared with the results obtained using a single 4th-order Hermite code for all particles (smooth green curve). The time evolution of the energy error is more erratic in the Nemesis method because of the close interactions of the orbiting star. The overall error, however, remains rather constant over a long timescale, whereas for the Hermite method the energy error is smoother but gradually grows with time, characteristic for a non-symplectic integrator. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In a more elaborate setup, children themselves could be subdivided into children to accommodate moons, etc. These sub-children could be integrated with another method. In principle each child and each parent can have its dedicated integration method, depending on the local requirements. It is even possible to built-up multiple Nemesis modules hierarchically to make an even more complicated compound.

5.6.1. Testing Nemesis

To illustrate the working of Nemesis we present the results of a calculation in which we integrated the multiple hierarchical configuration for simulating a star cluster with planetary systems, but without a galactic background. This is only the $C(P_8, S_4)$ component of the compound solver.

We test the setups by integrating an isolated system of five planets in orbit around a 1 $M_\odot$ star and a perturbing star in a wide orbit. The planetary system is generated using the oligarchic growth model [68] for a 1 $M_\odot$ star with a 400 au disk of 0.1 $M_\odot$. The outer star of $1 M_\odot$ has a semi-major axis of 1500 au with 0.5 eccentricity and an inclination of 90°.

The simulations were performed using Nemesisand integrating all objects in a single N-body code. The Nemesis method was constructed using one code for the planetary system and one code for the center of mass of the planetary system and the orbiting secondary star. For both integrators in Nemesis, we adopted the eighth order symplectic integrator in Huayno [69]. This system was integrated using a nemesis time step of $dt_\text{Nemesis} = 100$ yr. The calculation with Nemesis method took about ten minutes on a 3.6 GHz core i7-4790 based workstation: the Hermite scheme (green curve) took 18 hours on the same machine. The resulting evolution of the energy error is presented in In Fig. 8.

The energy error in the Hermite (smooth green curve) grows monotonically, which is the typical response for a non-symplectic integrator, such as the adopted Hermite scheme. The evolution of the energy error in the hybrid integrator does not grow on a secular timescale. The evolution of the energy error is rather erratic with sharp peaks to low values as well as high values but stays stable overall. The secular growth of Nemesis is much smaller than the single Hermite integrator.

6. Conclusions

We introduce a coupling strategy allowing numerical integrators to co-operate and interact without the requirement of making changes to the underlying codes. We show how this non-intrusive method is symplectic, it can be expanded to higher (even) orders, and how it can be augmented to include other terms in the operator. The scheme is based on the fundamental work of [14], which by introducing Bridge provides a powerful basis for building complex integrators while keeping the underlying codes dedicated, small and clean.

The symplectic behavior of Bridge is preserved when symplectic codes are coupled. When non-symplectic codes are coupled the symplectic quality of the compound solver is lost also. The consequential symplectic growth in the integration error can be compensated in part by adopting a higher-order integrator for the coupling strategy. The same strategy can be used when the individual coupled codes exhibit higher-order behavior.

We demonstrated how the scheme can be extended to higher-order, as well as to multiple interacting systems and deeper hierarchies of nested subsystems. Hierarchically coupled bridges provide a high-order symplectic coupling between two or more subsystems. The formulation is independent of the actual implementation of the evolution operators.

An additional advantage of the non-intrusive bridge presented here is the possibility to introduce variations in the spatial or temporal coordinate system and by additional forces in the interface. This allows us to introduce additional physical processes, such as post-Newtonian terms or dynamical friction without changes to the underlying codes. The main advantage of such a procedure is its ability to define a general high-level coupling strategy, which again allows underlying solvers to remain small and simple.

We tested the method on self-gravitating astrophysical systems in which we coupled Newtonian force evaluators via a (symplectic and non-symplectic) direct *N*-body and hierarchical *N*-body (tree) codes. We confirm that energy conservation has the right order for the second (Eq. 52), fourth (Eq. 53), sixth (Eq. 56), eighth (Eq. 58) and tenth (Eq. 59) order coupling strategies, up to machine precision ($dE/E \simeq 10^{-15}$), if the limitations regarding the component integrators are satisfied. Within Amuse up to tenth order methods are available. The method is hierarchical and the same expansion can be repeated to construct more complicated bridge structure.

We also present an implementation for integration in a rotating frame of reference. The formulation follows a similar splitting argument, where a formally symplectic scheme follows if we use canonical coordinates. A non-canonical formulation is somewhat easier to combine with self-interacting systems. The difference with the classic Bridge is that the kick operator affecting the interactions between the different subsystems must be adapted. Note that the evolution between kicks has not changed form, and the same integrator can be used for both operations. In the non-canonical case the resulting integrator is not formally symplectic (and this manifests itself as a net drift in energy), but for practical applications (e.g. a stellar cluster in a galaxy potential) the resulting drift is, due to the relatively low number of orbits, negligible in particular if combined with a higher-order integration scheme.

We finally demonstrate how multiple Bridge operators can be combined to construct a hierarchical compound system in which a subset of particles selectively feel the force of some other subset. This selective hierarchical nesting of bridge operators preserves the symplectic quality of the individual integrators. We call this method Nemesis and it is used to study planetary systems in star clusters.

The operators are implemented in the public Amuse and Omuse frameworks. Together with a wide variety of different numerical integrators, Amuse provides the ready-made building blocks for combining them to a tailored application to study astrophysical phenomena.

Declaration of Competing Interest

No interest

Acknowledgements

Appendix A

Overview of the coefficients used for high-order symplectic integration [31,32] used in Section 2.3.

The various integrators are indicated with S#M# (or equivalently $\mathcal{B}_S^M$). Here the first # indicates the splitting strategy; the symmetric composition S and the number of evaluations per bridge step M. The equations associated with the integration constants listed in Table 1.

$$\mathcal{B}_2 = K(u_0\tau)D(v_0\tau)K(u_0\tau) \tag{52}$$

$$\mathcal{B}_4^4 = K(u_0\tau)D(v_0\tau)K(u_1\tau)D(v_1\tau)K(u_2\tau)D(v_1\tau)K(u_1\tau)D(v_0\tau)K(u_0\tau) \tag{53}$$

$$\mathcal{B}_4^5 = K(u_0\tau)D(v_0\tau)K(u_1\tau)D(v_1\tau)K(u_2\tau)D(v_2\tau)K(u_2\tau)D(v_1\tau)K(u_1\tau)D(v_0\tau)K(u_0\tau) \tag{54}$$

$$\mathcal{B}_4^6 = K(u_0\tau)D(v_0\tau)K(u_1\tau)D(v_1\tau)K(u_2\tau)D(v_2\tau)K(u_3\tau)D(v_2\tau)K(u_2\tau)D(v_1\tau) \\ K(u_1\tau)D(v_0\tau)K(u_0\tau) \tag{55}$$

$$\mathcal{B}_6^{11} = K(w_0\tau/2)D(w_0\tau)K((w_0+w_1)\tau/2)D(w_1\tau)K((w_1+w_2)\tau/2)D(w_2\tau) \\ K((w_2+w_3)\tau/2)D(w_3\tau)K((w_3+w_4)\tau/2)D(w_4\tau)K((w_4+w_5)\tau/2)D(w_5\tau) \\ K((w_5+w_4)\tau/2)D(w_4\tau)K((w_4+w_3)\tau/2)D(w_3\tau)K((w_3+w_2)\tau/2)D(w_2\tau) \\ K((w_2+w_1)\tau/2)D(w_1\tau)K((w_1+w_0)\tau/2)D(w_0\tau)K((w_0)\tau/2) \tag{56}$$

$$\begin{aligned}
\mathcal{B}_6^{13} = & K(w_0\tau/2)D(w_0\tau)K((w_0+w_1)\tau/2)D(w_1\tau)K((w_1+w_2)\tau/2)D(w_2\tau) \\
& K((w_2+w_3)\tau/2)D(w_3\tau)K((w_3+w_4)\tau/2)D(w_4\tau)K((w_4+w_5)\tau/2)D(w_5\tau) \\
& K((w_5+w_6)\tau/2)D(w_6\tau)K((w_6+w_5)\tau/2)D(w_5\tau)K((w_5+w_4)\tau/2)D(w_4\tau) \\
& K((w_4+w_3)\tau/2)D(w_3\tau)K((w_3+w_2)\tau/2)D(w_2\tau)K((w_2+w_1)\tau/2)D(w_1\tau) \\
& K((w_1+w_0)\tau/2)D(w_0\tau)K(w_0\tau/2)
\end{aligned}$$
(57)

$$\begin{aligned}
\mathcal{B}_8^{21} = & K(w_0\tau/2)D(w_0\tau)K((w_0+w_1)\tau/2)D(w_1\tau)K((w_1+w_2)\tau/2)D(w_2\tau) \\
& K((w_2+w_3)\tau/2)D(w_3\tau)K((w_3+w_4)\tau/2)D(w_4\tau)K((w_4+w_5)\tau/2)D(w_5\tau) \\
& K((w_5+w_6)\tau/2)D(w_6\tau)K((w_6+w_7)\tau/2)D(w_7\tau)K((w_7+w_8)\tau/2)D(w_8\tau) \\
& K((w_8+w_9)\tau/2)D(w_9\tau)K((w_9+w_{10})\tau/2)D(w_{10}\tau)K((w_{10}+w_9)\tau/2)D(w_9\tau) \\
& K((w_9+w_8)\tau/2)D(w_8\tau)K((w_8+w_7)\tau/2)D(w_7\tau)K((w_7+w_6)\tau/2)D(w_6\tau) \\
& K((w_6+w_5)\tau/2)D(w_5\tau)K((w_5+w_4)\tau/2)D(w_4\tau)K((w_4+w_3)\tau/2)D(w_3\tau) \\
& K((w_3+w_2)\tau/2)D(w_2\tau)K((w_2+w_1)\tau/2)D(w_1\tau)K((w_1+w_0)\tau/2)D(w_0\tau) \\
& K(w_0\tau/2)
\end{aligned}$$
(58)

$$\begin{aligned}
\mathcal{B}_{10}^{35} = & K(w_0\tau/2)D(w_0\tau)K((w_0+w_1)\tau/2)D(w_1\tau)K((w_1+w_2)\tau/2)D(w_2\tau) \\
& K((w_2+w_3)\tau/2)D(w_3\tau)K((w_3+w_4)\tau/2)D(w_4\tau)K((w_4+w_5)\tau/2)D(w_5\tau) \\
& K((w_5+w_6)\tau/2)D(w_6\tau)K((w_6+w_7)\tau/2)D(w_7\tau)K((w_7+w_8)\tau/2)D(w_8\tau) \\
& K((w_8+w_9)\tau/2)D(w_9\tau)K((w_9+w_{10})\tau/2)D(w_{10}\tau)K((w_{10}+w_{11})\tau/2)D(w_{11}\tau) \\
& K((w_{11}+w_{12})\tau/2)D(w_{12}\tau)K((w_{12}+w_{13})\tau/2)D(w_{13}\tau)K((w_{13}+w_{14})\tau/2)D(w_{14}\tau) \\
& K((w_{14}+w_{15})\tau/2)D(w_{15}\tau)K((w_{15}+w_{16})\tau/2)D(w_{16}\tau)K((w_{16}+w_{17})\tau/2)D(w_{17}\tau) \\
& K((w_{15}+w_{16})\tau/2)D(w_{16}\tau)K((w_{14}+w_{15})\tau/2)D(w_{15}\tau)K((w_{13}+w_{14})\tau/2)D(w_{14}\tau) \\
& K((w_{12}+w_{13})\tau/2)D(w_{13}\tau)K((w_{11}+w_{12})\tau/2)D(w_{12}\tau)K((w_{10}+w_{11})\tau/2)D(w_{11}\tau) \\
& K((w_{10}+w_9)\tau/2)D(w_9\tau)K((w_9+w_8)\tau/2)D(w_8\tau)K((w_8+w_7)\tau/2)D(w_7\tau) \\
& K((w_7+w_6)\tau/2)D(w_6\tau)K((w_6+w_5)\tau/2)D(w_5\tau)K((w_5+w_4)\tau/2)D(w_4\tau) \\
& K((w_4+w_3)\tau/2)D(w_3\tau)K((w_3+w_2)\tau/2)D(w_2\tau)K((w_2+w_1)\tau/2)D(w_1\tau) \\
& K((w_1+w_0)\tau/2)D(w_0\tau)K(w_0\tau/2)
\end{aligned}$$
(59)

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cnsns.2020.105240 .

References

[1] Turing AM. On computable numbers, with an application to the entscheidungsproblem. Proc London Math Soc 1936;2(42):230–65.
[2] Oberkampf WL, Roy CJ. Verification and validation in scientific computing. 1st edition. New York, NY, USA: Cambridge University Press; 2010.
[3] Moore GE. Cramming more components onto integrated circuits. Electronics 1965;38(8–12).
[4] Felleisen M. On the expressive power of programming languages. Sci Comput Program 1991;17(1):35–75.
[5] Hoekstra AG, Zwart SP, Coveney PV. Multiscale modelling, simulation and computing: from the desktop to the exascale. Philos Trans R Soc A: Math Phys Eng Sci 2019;377(2142):20180355.
[6] Manser CJ, et al. A planetesimal orbiting within the debris disc around a white dwarf star. Science 2019;364(6435):66–9.
[7] Fowler M. Refactoring: improving the design of existing code. Boston, MA, USA: Addison-Wesley; 1999.
[8] Hoekstra AG, Portegies Zwart S, Bubak M, Sloot P. Towards distributed petascale computing. Petascale Computing: Algorithms and Applications. Bader DA, editor. Chapman & Hall/CRC computational science series; 2008. 565pp. (ISBN: 9781584889090, ISBN 10: 1584889098), pp. 147–164
[9] Groen D, Portegies Zwart S, Ishiyama T, Makino J. High-performance gravitational n-body simulations on a planet-wide-distributed supercomputer. Computational Science and Discov 2011;4(1):015001.
[10] Newton I. Philosophiae Naturalis Principia Mathematica 1687;1.
[11] Valtonen M, Karttunen H. The three-Body problem. Cambridge University Press; 2006.
[12] Portegies Zwart SF, Boekholt TC. Numerical verification of the microscopic time reversibility of newton's equations of motion: fighting exponential divergence. Commun Nonlinear Sci Numer Simul 2018;61:160–6.
[13] Barnes J, Hut P. A hierarchical o(nlogn) force-calculation algorithm. Nat 1986;324:446–9.
[14] Fujii M, Iwasawa M, Funato Y, Makino J. BRIDGE: a direct-tree hybrid n-body algorithm for fully self-consistent simulations of star clusters and their parent galaxies. Publ Astr Soc Jpn 2007;59:1095.
[15] Wisdom J, Holman M. Symplectic maps for the n-body problem. AJ 1991;102:1528–38.
[16] Makino J, Aarseth SJ. On a hermite integrator with Ahmad-Cohen scheme for gravitational many-body problems. Publ Astr Soc Jpn 1992;44:141–51.
[17] Mikkola S, Aarseth SJ. An efficient integration method for binaries in n-body simulations. New Astron 1998;3:309–20.
[18] Iwasawa M, Portegies Zwart S, Makino J. GPU-enabled particle-particle particle-tree scheme for simulating dense stellar cluster system. Comput Astrophys Cosmol 2015;2:6.
[19] Bédorf J, Portegies Zwart S. The effect of many minor mergers on the size growth of compact quiescent galaxies. MNRAS 2013;431:767–80.
[20] Borgdorff J, Lorenz E, Hoekstra AG, Falcone JL, Chopard B. A principled approach to distributed multiscale computing, from formalization to execution. e-Science Workshops. IEEE Computer Society; 2011. Pp.97–104

[21] Murray C.D., Dermott S.F.. Solar system dynamics. 1999. Murray, C. D., Dermott, S. F.
[22] Bédorf J, et al. 24.77 pflops on a gravitational tree-code to simulate the milky way galaxy with 18600 gpus. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '14. Piscataway, NJ, USA: IEEE Press; 2014. Pp. 54–65
[23] Bender M, Heenen PH, Reinhard PG. Self-consistent mean-field models for nuclear structure. Rev Mod Phys 2003;75(1):121–80.
[24] Meiron Y, Li B, Holley-Bockelmann K, Spurzem R. Expansion techniques for collisionless stellar dynamical simulations. ApJ 2014;792:98.
[25] Dehnen W. A fast multipole method for stellar dynamics. Comput Astrophys Cosmol 2014;1:1.
[26] Milgrom M. A modification of the newtonian dynamics as a possible alternative to the hidden mass hypothesis. ApJ 1983;270:365–70.
[27] Will CM. Incorporating post-newtonian effects in n-body dynamics. Phys Rev D 2014;89(4):044043.
[28] Duez MD, Zlochower Y. Numerical relativity of compact binaries in the 21st century. Rep Prog Phys 2019;82(1):016902.
[29] Spitzer LJ, Hart MH. Random gravitational encounters and the evolution of spherical systems. ii. models. ApJ 1971;166:483.
[30] Suriano SS, Li ZY, Krasnopolsky R, Suzuki TK, Shang H. The formation of rings and gaps in wind-launching non-ideal MHD discs: three-dimensional simulations. MNRAS 2019;484:107–24.
[31] Hairer E, Lubich C, Wanner G. Geometric numerical integration. Springer Verlag; 2006. Vol. 31
[32] Yoshida H. Construction of higher order symplectic integrators. Phys Lett A 1990;150:262–8.
[33] Sofroniou M, Spaletta G. Derivation of symmetric composition constants for symmetric integrators. Optim Methods Softw 2005;20(4–5):597–613.
[34] Dehnen W. A hierarchical $\mathcal{O}(n)$ force calculation algorithm. J Comput Phys 2002;179:27–42.
[35] Minchev I, Famaey B. A new mechanism for radial migration in galactic disks: spiral-bar resonance overlap. ApJ 2010;722:112–21.
[36] Martínez-Barbosa C, et al. The evolution of the sun's birth cluster and the search for the solar siblings with gaia. MNRAS 2016;457:1062–75.
[37] Pfenniger D, Friedli D. Computational issues connected with 3d n-body simulations. aap 1993;270:561–72.
[38] Rubincam DP. Radiative spin-up and spin-down of small asteroids. Icarus 2000;148:2–11.
[39] Vokrouhlický D., Bottke W.F., Chesley S.R., Scheeres D.J., Statler T.S.. The yarkovsky and YORP effects. 2015. Michel P, DeMeo F.E., Bottke W.F., 509–531.
[40] Portegies Zwart S, McMillan S. Astrophysical Recipes; The art of AMUSE 2018.
[41] Hellström C, Mikkola S. Explicit algorithmic regularization in the few-body problem for velocity-dependent perturbations. Celestial Mech Dyn Astron 2010;106:143–56.
[42] Iwasawa M, Oshino S, Fujii MS, Hori Y. PENTACLE: Parallelized particle-particle particle-tree code for planet formation. Publ Astr Soc Jpn 2017;69(5):81.
[43] Hozumi S, Iwasawa M, Nitadori K. A mean-field approach to simulating the merging of collisionless stellar systems using a particle-based method. ApJ 2019;875(1):20.
[44] Portegies Zwart S, McMillan SLW, van Elteren E, Pelupessy I, de Vries N. Multi-physics simulations using a hierarchical interchangeable software interface. Comput Phys Commun 2013;183:456–68.
[45] Pelupessy FI, et al. The astrophysical multipurpose software environment. A&A 2013;557:A84.
[46] Pelupessy I, et al. The oceanographic multipurpose software environment. Geosci Model Dev Discuss 2016;2016:1–36.
[47] Portegies Zwart S, Bédorf J. Using GPUs to enable simulation with computational gravitational dynamics in astrophysics. Computer 2015;48(11):50–8.
[48] Portegies Zwart S. Computational astrophysics for the future. Science 2018;361(6406):979–80.
[49] Plummer HC. On the problem of distribution in globular star clusters. MNRAS 1911;71:460–70.
[50] Sharma S, Bland-Hawthorn J, Johnston KV, Binney J. Galaxia: a code to generate a synthetic survey of the milky way. Astrophysics Source Code Library; 2011.
[51] Pelupessy FI, Portegies Zwart S. The evolution of embedded star clusters. MNRAS 2012;420:1503–17.
[52] Heggie DC, Mathieu RD. Standardised units and time scales. The Use of Supercomputers in Stellar Dynamics. Hut P, McMillan SLW, editors. Lecture Notes in Physics, Berlin Springer Verlag; 1986. Vol. 267, p. 233
[53] Boekholt T, Portegies Zwart S. On the reliability of n-body simulations. Comput Astrophys Cosmol 2015;2:2.
[54] Verlet L. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Phys Rev 1967;159(1):98–103.
[55] King IR. The structure of star clusters. iii. some simple dvriamical models. AJ 1966;71:64–75.
[56] Barnes JE. Gravitational softening as a smoothing operation. MNRAS 2012;425(2):1104–20.
[57] Portegies Zwart SF, Boekholt TCN. Numerical verification of the microscopic time reversibility of newton's equations of motion: fighting exponential divergence. Commun Nonlinear Sci NumerSimul 2018;61:160–6.
[58] Spitzer L. Dynamical evolution of globular clusters. Princeton, NJ: Princeton University Press; 1987. 1987, 191 p.
[59] McMillan SLW, Portegies Zwart SF. The fate of star clusters near the galactic center. i. analytic considerations. ApJ 2003;596:314–22.
[60] Spinnato PF, Fellhauer M, Portegies Zwart SF. The efficiency of the spiral-in of a black hole to the galactic centre. MNRAS 2003;344:22–32.
[61] Allen C, Santillan A. An improved model of the galactic mass distribution for orbit computations. Rev Mex Astron Astrofis 1991;22:255–63.
[62] Ferrers NM. Pure Appl Math 1877;14:1.
[63] Antoja T, et al. Constraints on the galactic bar from the hercules stream as traced with RAVE across the galaxy. A&A 2014;563:A60.
[64] Martínez-Barbosa CA, Brown AGA, Portegies Zwart S. Radial migration of the sun in the milky way: a statistical study. MNRAS 2015;446:823–41.
[65] Portegies Zwart S, Boekholt T. On the minimal accuracy required for simulating self-gravitating systems by means of direct n-body methods. ApJL 2014;785:L3.
[66] van Elteren A, Portegies Zwart S, Pelupessy I, Cai MX, McMillan SLW. Survivability of planetary systems in young and dense star clusters. A&A 2019;624:A120.
[67] Hamers AS, Portegies Zwart SF, Merritt D. Relativistic dynamics of stars near a supermassive black hole. MNRAS 2014;443:355–87.
[68] Kokubo E, Ida S. Oligarchic growth of protoplanets. Icarus 1998;131:171–8.
[69] Pelupessy FI, Jänes J, Portegies Zwart S. N-Body integrators with individual time steps from hierarchical splitting. New Astron 2012;17:711–19.
[70] Portegies Zwart S. AMUSE: astrophysical multipurpose software environment. Astrophysics Source Code Library; 2011.
[71] Portegies Zwart S., et al. AMUSE: the astrophysical multipurpose software environment. 2018b.
[72] Bovy J. galpy: galactic dynamics package. Astrophysics Source Code Library; 2014.
[73] SLW M. Modeling the formation and evolution of dense stellar systems. AAS/Division of Dynamical Astronomy Meeting. AAS/Division of Dynamical Astronomy Meeting; 2014. Vol. 45, p. 303.01
[74] Hunter JD. Matplotlib: a 2d graphics environment. Comput Sci Eng 2007;9:90–5.
[75] Oliphant TE. A guide to NumPy. USA: Trelgol Publishing; 2006. Vol. 1
[76] van Rossum G. Extending and embedding the python interpreter. Report CS-R9527; 1995.