The handle http://hdl.handle.net/1887/38874 holds various files of this Leiden University dissertation.

**Author:** Martinez-Barbosa, Carmen Adriana  
**Title:** Tracing the journey of the sun and the solar siblings through the Milky Way  
**Issue Date:** 2016-04-13
Abstract

The BRIDGE integrator [Fujii et al., 2007] presents an efficient method to calculate the combined evolution of a small system embedded in larger systems, the typical application being the collisional dynamics of star clusters in the large scale environment of their parent galaxies. Here, we present generalizations of the principles of BRIDGE to a wider set of applications: we generalize the second order coupling of BRIDGE to higher order and we present a formulation for a rotating frame of reference. We also discuss the formulation of BRIDGE for cases where part of the system evolves
in the post-Newtonian regime. We present example applications for these cases and discuss the conditions under which our integrators can be applied.

**keywords:** Stellar dynamics; Methods: $N$-body simulations; Methods: numerical

### 2.1. Introduction

Over the past five decades astrophysical simulations have been an indispensable tool to understand large observational datasets and gain knowledge about the formation and evolution of astrophysical systems. The demand for such simulations has increased accordingly and, in close relationship with the development of computer hardware, it is likely to continue to grow in the coming years due to the recent introduction of multi-core architectures such as Graphic Processing Units (GPUs) as general purpose performance boosters.

The main bottleneck in gravitational $N$-body codes is often attributed to the amount of computational work needed to accurately compute the orbits of the particles in the system on a star-by-star basis. For the direct summation method, which is the simplest and more accurate method for calculating forces, the computational cost scales as $O(N^2)$. A way to alleviate the computational requirements for $N$-body simulations emerged with the development of tree-based algorithms which approximates the forces by means of a multipole expansion, where interactions with a subset of distant particles are only included via multipole terms. This in practice reduces the computational costs to $O(N \log(N))$ at the expenses of a more complex algorithmic implementation and a decreased precision in the force calculation. In addition, the development of a momentum-conserving fast multi-pole method (FMM) lowered the computational complexity of $N$ -body simulations to $O(N \log(\log(N))) \approx O(N)$ [Dehnen, 2014]. Each one of these methods have its own range of applicability. For example, direct summation methods are more appropriate for simulations of collisional systems such as planetary systems, star-clusters, galactic nuclei and black-hole (BH) dynamics. Tree- and FMM-based codes are better suitable for studies of collisionless systems such as galactic dynamics and cosmological
2.1 Introduction

simulations. In addition to this, with the introduction of parallel architectures, GRAvity PipE (GRAPE) dedicated boards, and more recently, GPUs, a big boost (typically one to two orders of magnitude) in computational speed was accomplished and today’s simulation codes have greatly improved their nominal dynamic range and numerical precision [see e.g. Portegies Zwart & Bédorf, 2014; Belleman et al., 2014].

On the other hand, gravitational \( N \)-body simulations require both, a fast way to calculate the forces and an accurate integration method to evolve the particles in time. Tree- and FMM-based codes typically adopt a simple leapfrog scheme, whereas in direct summation codes, symplectic schemes (for long term integrations of planetary systems) and high-order Hermite schemes (for integrations of star-clusters or galactic nuclei with BHs) have been systematically adopted. Symplectic schemes with hierarchical splitting have been recently developed and they could be an alternative to non-symplectic Hermite schemes. Other techniques such as the use of an individual or block-time-step algorithm [Makino et al., 2006], Ahmad-Cohen neighbour scheme [Ahmad & Cohen, 1973] and more exotic hybrids [Wisdom & Holman, 1991; McMillan & Aarseth, 1993; Pelupessy et al., 2012] reduce the computational costs associated with the force calculation. The block-time-step scheme is specially helpful in this sense, because in this case only a subset or block of particles need to be evolved per time-step, which in practice reduces the total calculation cost considerably in simulations which cover a wide range in time-scales. Nevertheless, an additional layer of efficiency can be gained by tailoring the integration method to the specific problem at hand. For example, a hybrid method named BRIDGE and firstly introduced by Fujii et al. [2007], was developed as a way to combine two different gravitational solvers. BRIDGE was developed to study the evolution of a star-cluster orbiting a parent galaxy.

The BRIDGE method is based on an second order extension of the mixed variable symplectic (MVS) scheme developed in the context of long term integrations of planetary systems. In its classic version [Fujii et al., 2007], BRIDGE couples a highly accurate direct code with a fast tree-code in a single compound solver, making the co-evolution of collisionless and collisional systems self-consistent. This BRIDGE method is quite powerful, both for the elegant principle it embodies as well as for the efficiency it allows, combining different
specialized solvers without the necessity to modify each one of the solvers individually. In Saitoh & Makino [2010]; Pelupessy et al. [2013] the BRIDGE method has been extended for coupling a gravitational solver and a SPH-based hydrodynamical solver, using a fully asynchronous time-stepping scheme, where the gravity and hydro solvers are allowed to use different time-step sizes to optimize performance.

The aim of this chapter is to present different generalizations of the classical BRIDGE method, which allows for the coupling between an arbitrary number of specialized solvers. In Sect. 2.2 we present a brief review of the classical BRIDGE method. We also introduce generalizations of this integrator, such as a high-order BRIDGE, a BRIDGE which includes post-Newtonian corrections and a BRIDGE to be used in rotating coordinates. In Sect. 2.3 we show the validation and applications of each generalized BRIDGE integrator. We discuss the advantages and limitations in Sect 2.4 and conclude in Sect. 2.4.

2.2. Method

2.2.1. The Classic Bridge

In the classic BRIDGE scheme a hybrid integrator combining two different gravitational solvers is constructed in order to study the evolution of a system comprised of two interacting systems, the canonical example being a star-cluster orbiting a parent galaxy. For this example the cluster is integrated using a 4-th Hermite direct summation code. Interactions between stars in the galaxy, and between the cluster and the galactic stars are resolved using a hierarchical tree-code. In this section we briefly revise the BRIDGE method, as a preparation for the discussion on its high-order generalization and extensions.

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:

$$H = \sum_{i \in A \cup B} \frac{||p_i||^2}{2m_i} - \sum_{i \neq j \in A \cup B} \frac{Gm_i m_j}{||r_i - r_j||}.$$  (2.1)
2.2 Method

The systems $A$ and $B$ may represent a star cluster and its parent Galaxy respectively. Following Fujii et al. [2007], the Hamiltonian shown in Eq. 2.1 can be separated in the following way:

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

(2.2)

where:

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

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

$$H_{\text{int}} = -\sum_{i \in A, j \in B} \frac{Gm_i m_j}{||\mathbf{r}_i - \mathbf{r}_j||}$$  

(2.3)

The time evolution of the whole system can be written, for a second order approximation, as follows:

$$e^{\tau H} = e^{\frac{\tau}{2} H_{A+B}} e^{\tau H_{\text{int}}} e^{\frac{\tau}{2} H_{A+B}}$$  

or

$$e^{\tau H} = e^{\frac{\tau}{2} H_{\text{int}}} e^{\tau H_{A+B}} e^{\frac{\tau}{2} H_{\text{int}}}.$$

(2.4)

The operator $e^{\tau H_{\text{int}}}$ represents pure momentum kicks, since $H_{\text{int}}$ only depends on the positions. During this process, the velocity of the stars in the cluster are updated due to the external force generated by the galaxy. The velocity of the stars in the galaxy are also updated after computing the acceleration due to their self-gravity by means of a tree code.

Since $H_A$ and $H_B$ are completely independent the evolution operator $e^{\tau H_{A+B}} = e^{\tau H_A} e^{\tau H_B}$ consists of the separate evolution of the the two subsystems. For the example of a cluster in a galaxy, a direct code is used to evolve accurately the stellar cluster while a tree code is used in parallel to follow the evolution of the galaxy system. A full time-step in BRIDGE integrator then consists of i) mutually kicking the sub-systems $A$ and $B$ for $\frac{\tau}{2}$, ii) evolving the two sub-systems $A$ and $B$ for $\tau$, and then iii) mutually kicking the sub-systems $A$ and $B$ again for $\frac{\tau}{2}$.
and $B$ in isolation for $\tau$ using suitable codes together with an update of their positions, and iii) mutually kicking the sub-systems $A$ and $B$ for another $\frac{\tau}{2}$.

In the classical BRIDGE scheme a fully self-consistent treatment of the whole system is achieved using Eq. 2.4. The bridge integrator can allow a more efficient calculation of the evolution of the joined system under the following conditions: the first requirement (which is a necessary requirement) is that the timestep allowed by the interaction terms $H_{int}$ is longer than one or both of the internal timesteps of the $H_A$ and $H_B$ systems (this can happen, but not exclusively so, if the two subsystems are spatially and temporally well separated). Secondly, and this is optional, it may be that the two systems are evolving in a different regime, such that different integrators, geared towards their respective dynamics, can be used. For the cluster-galaxy example both conditions contribute: the internal dynamical timescale of a cluster is much shorter than the interaction timescale of the cluster-galaxy interactions, and the cluster is governed by collisional dynamics while the galaxy experiences collisionless dynamics.

Thus the coupling of codes in BRIDGE works well when the interacting sub-systems stays relatively well separated in spatial and/or temporal scales during a simulation. It is not difficult to find a counter example 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 timestep required reduces to the global minimum timestep. Note that in the approach above, the coupling strategy is defined manually at the beginning of a 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 case and one has to monitor the simulation carefully. In principle, the use of a self-adaptive time-stepping in BRIDGE would alleviate this issue. However, ultimately, the problem would still exist since the coupling strategy still remains static throughout the simulation, therefore, demanding the use of unnecessarily small time-steps for the coupling. A dynamic approach where the system is merged and/or split as necessary would provide a more robust solution in this case. A similar method like this was adopted by Iwasawa et al. [2015], but then coded directly in C.
2.2 Method

The above formulation provides fully symplectic time evolution. However, in practice, each of the codes being bridged may not be symplectic, in which case the compound solver is not symplectic. Note that the above formulation is not limited to only two sub-systems. Indeed, multiple sub-systems being integrated by different specialized solvers can be bridged either by a Hamiltonian splitting technique (see section 2.2.2) or by applying the above split scheme recursively [Pelupessy et al., 2012].

2.2.2. A High-Order Bridge

For systems in which the spatial and/or temporal scale of interaction of two or more sub-systems are not well separated, the BRIDGE scheme may have numerical difficulties. Additionally, the coupling of two integrators of high order (fourth and higher order are often used in the context of cluster problems) still occurs with a second order method, which ultimately determines the global order of the compound method. In order to address these issues, it is necessary to increase the order of the BRIDGE scheme. We present here a generalization of the classical BRIDGE to a high-order coupling scheme. First, we show how BRIDGE can be extended to a higher order, also taking into account multiple subsystems. Then we describe how the order of the coupling scheme can be decided based on the integrators being bridged.

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,

\[
H = \sum_{i \in S} \frac{||p_i||^2}{2m_i} - \sum_{i \neq j \in S} \frac{Gm_im_j}{||r_i - r_j||}, \tag{2.5}
\]

can be split such that we obtain:

\[
H = \sum_k^Q H_{S_k} + \sum_{k \neq l}^Q H_{S_kS_l}^{int}. \tag{2.6}
\]

The terms in Eq. 2.6 are given by the following relations:
Chapter 2 : High-order hybrid N-body methods for compound systems

\[ H_{S_k} = \sum_{i \in S_k} \frac{\|p_i\|^2}{2m_i} - \sum_{i \neq j \in S_k} \frac{Gm_im_j}{\|r_i - r_j\|} \]

\[ H_{S_kS_l}^{\text{int}} = \sum_{i \in S_k, j \in S_l} -\frac{Gm_im_j}{\|r_i - r_j\|} \tag{2.7} \]

Based on this splitting, a general, multi sub-system, second order time evolution operator can be constructed as follows:

\[ \text{Bridge}_2(\tau) = \prod_{k} e^{\frac{\tau}{2} H_{S_k}} \prod_{k \neq l} e^{\tau H_{S_kS_l}^{\text{int}}} \prod_{k} e^{\frac{\tau}{2} H_{S_k}} \text{ or } \]

\[ \text{Bridge}_2(\tau) = \prod_{k \neq l} e^{\frac{\tau}{2} H_{S_kS_l}^{\text{int}}} \prod_{k} e^{\tau H_{S_k}} \prod_{k \neq l} e^{\frac{\tau}{2} H_{S_kS_l}^{\text{int}}} . \tag{2.8} \]

Similarly to the classical Bridge, operators \( e^{\tau H_{S_k}} \) independently evolves each of the sub-systems \( S_k \) in isolation. Operators \( e^{\tau H_{S_kS_l}^{\text{int}}} \) 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. 2.6, the forces due to the interaction terms do not depend on velocities, therefore, the operators \( e^{\tau H_{S_kS_l}^{\text{int}}} \) commutate amongst themselves. We note, however, that commutability is not possible for velocity dependent forces and therefore, a special treatment is required (see section 2.2.4). Since each of the operators \( e^{\tau H_{S_k}} \) and \( e^{\tau H_{S_kS_l}^{\text{int}}} \) can, in principle, be associated to different solvers running concurrently, a highly efficient parallel implementation can be achieved for the generalized Bridge scheme in eq. 2.8.

A high-order Bridge scheme can be constructed in a similar way as in a symplectic integrator. By defining the drift and kick operators as:

\[ D(\tau) = \prod_{k} e^{\tau H_{S_k}} , \tag{2.9} \]

and

\[ K(\tau) = \prod_{k \neq l} e^{\tau H_{S_kS_l}^{\text{int}}} , \tag{2.10} \]
2.2 Method

eq 2.8 can be extended to a higher order by composition of $D(\tau)$ and $K(\tau)$ operators [Heirer et al., 2006]. For a 4th symmetric composition with 4 stages, the high-order Bridge takes the form:

$$\text{Bridge}_4(\tau) = D(u_0 \tau)K(u_0 \tau)D(u_1 \tau)K(v_1 \tau)$$

$$D(u_2 \tau)K(v_1 \tau)D(u_1 \tau)K(v_0 \tau)D(u_0 \tau),$$

(2.11)

[e.g. Yoshida, 1990; Heirer et al., 2006, also for the coefficients $u_i, v_i$]. For a sixth and higher order symmetric compositions of the symmetric method, $D(\tau/2)K(\tau)D(\tau/2)$ leads to

$$\text{Bridge}_6(\tau) = D(w_0/2 \tau)K(w_0 \tau)D((w_0 + w_1)/2 \tau) ...$$

$$... D((w_{s-1} + w_s)/2 \tau)K(w_s \tau)D(w_s/2 \tau),$$

(2.12)

with coefficients $w_i$ given in Sofroniou & Spaletta [2005] and Heirer et al. [2006]. We notice that the self-adjoint methods associated to eqs. 2.11 and 2.12 are also equally possible. The formulation above provides a fully symplectic time evolution if the codes being bridged are symplectic as well.

Using this scheme, codes/integrators of different orders can be coupled in order to construct a formally 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 appropriate to choose $\text{Bridge}_4(\tau)$ in order to have a fourth order convergent compound method. If on the other hand $\text{Bridge}_6(\tau)$ would have been chosen, the method would be still be only fourth order (constrained by the fourth order sub-integrator). On the other hand, the logic of selecting an appropriate Bridge order may not always be straightforward: for example: if a second, fourth and sixth order were coupled in the classical Bridge ($\text{Bridge}_2(\tau)$), resulting in an second order convergent compound method. This, however, can be improved by adopting the notion of a hierarchical coupling strategy: the fourth and sixth order codes being bridged with $\text{Bridge}_4(\tau)$, and then this compound method being bridged with the remaining second order code using $\text{Bridge}_2(\tau)$. In this hierarchical coupling strategy, while formally the overall order of convergence of the compound solver is still second order, locally, sub-systems being evolved with the more precise codes are still being coupled at higher order than the rest of the system,
and this can be advantageous if this subsystem dominates in the overall error or requires higher precision, for example because of the requirement to resolve close encounters. We note then, that, while the methods presented here are highly flexible, the choice of integrators is highly problem dependent.

### 2.2.3. Bridge in rotating frames

It is often expedient to consider a system in a non-inertial, rotating frame of reference. An example of this being the evolution of one or more (potentially interacting) stellar system in an analytic galaxy potential, with contributions from a galactic halo component, stellar disc, central bar and galactic spiral arms. In general, the bar and spiral arms of galaxies rotate as rigid bodies with certain pattern speed [Minchev & Famaey, 2010]. This means that the potential associated with these galactic components will depend on time in an inertial frame. In this case choosing a frame corotating with either bar and/or spiral arms will make the potential contribution of these time independent and integrating in this frame will allow for larger timesteps and/or better energy conservation behavior. For these cases we can 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 since it, as we will see, it allows the use of ‘normal’ integrators, formulated for an inertial frame, without changes).

First, we consider a particle of mass $m$ located in a frame that rotates around the $z$-axis with constant angular speed $\Omega$. The Hamiltonian of this particle is given by the relation:

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

$$H = \frac{||p||^2}{2m} + U_{\text{ext}}(r) - (\Omega \times r) \cdot p - \frac{1}{2}m||\Omega \times r||^2. \quad (2.13)$$

Here $r$ and $p$ are the position and momentum vectors of the particle in the rotating frame. The term $U_{\text{ext}}(r)$ is the potential energy due to an external force, which depends only on the position of the particle. In particular, $U_{\text{ext}}(r)$ represents the galactic potential. The last two terms in Eq. 2.13 correspond to a potential energy which accounts for the centrifugal and Coriolis forces. The
energy associated to the centrifugal and Coriolis forces together with \( U_{\text{ext}}(r) \), represents the total generalized potential energy of the particle, \( U_{\text{gen}}(r, p) \).

Note that \( U_{\text{gen}}(r, p) \) 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 in Eqs. 2.9 and 2.10. Thus, the way to construct a rotating Bridge integrator is by splitting the equations of motion of a test particle that satisfies Eq. 2.13.

There are two approaches to generate a rotating Bridge namely the canonical and non-canonical approximations [see also e.g. Pfenniger & Friedli, 1993]. In Sects. 2.2.3 and 2.2.3 we explain these approaches in more detail. We also generalize the rotating Bridge integrator to be used in systems of self-interacting particles.

**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 \((Q, P)\). The canonical momentum \((P)\) is defined as

\[
P = \frac{\partial \mathcal{L}}{\partial \dot{Q}} = \frac{\partial \mathcal{L}}{\partial \dot{r}} = p + m(\Omega \times r). \tag{2.14}
\]

Here \( \mathcal{L} \) is the Lagrangian \( \mathcal{L} = \dot{Q}P - H \). The canonical momentum can be interpreted as the velocity of the particle seen in the inertial reference frame which is coaxial to the rotating one. By using Eq. 2.14 we can obtain the canonical momenta in cartesian components:

\[
P_x = p_x - m\Omega y \\
P_y = p_y + m\Omega x \\
P_z = p_z. \tag{2.15}
\]

The equations of motion of a particle that satisfies Eq. 2.13 can then be written in terms of the canonical moment as follows:
Here $F$ is the external force associated to $U_{\text{ext}}(r)$. We proceed to split Eqs. 2.16 to build the kick and drift operators. The set of equations that represent the kick operator $K(\tau)$ is the following:

\begin{align*}
\dot{x} &= \dot{y} = \dot{z} = 0, \\
p_x &= F_x + \Omega P_y, \\
p_y &= F_y - \Omega P_x, \\
p_z &= F_z.
\end{align*}

The solution of these equations is:

\begin{align*}
v_x(t + \tau) &= v_x(t) - \left(\frac{a_y + \Omega^2 y}{\Omega}\right) \cos(\Omega \tau) \\
&\quad + \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{\Omega}\right)\right] \sin(\Omega \tau) \\
&\quad + \frac{a_y + \Omega^2 y}{\Omega}, \quad (2.18a) \\
v_y(t + \tau) &= -\left[v_x(t) - \left(\frac{a_y + \Omega^2 y}{\Omega}\right)\right] \sin(\Omega \tau) \\
&\quad + \left[v_y(t) + \left(\frac{a_x + \Omega^2 x}{\Omega}\right)\right] \cos(\Omega \tau) \\
&\quad - \frac{a_x + \Omega^2 x}{\Omega}, \quad (2.18b) \\
v_z(t + \tau) &= v_z(t) + a_z \tau. \quad (2.18c)
\end{align*}

Here $\tau$ corresponds to the rotating BRIDGE timestep.

The drift operator $D(\tau)$ in the canonical approximation is represented by the following set of equations:
2.2 Method

\[ \dot{p}_x = \dot{p}_y = \dot{p}_z = 0, \]
\[ \dot{x} = p_x/m + \Omega y, \]
\[ \dot{y} = p_y/m - \Omega x, \]
\[ \dot{z} = p_z/m. \]  \hspace{1cm} (2.19)

The solution of these equations is

\[ x(t + \tau) = \left[ x(t) - \frac{v_y}{\Omega} \right] \cos (\Omega \tau) \]
\[ + \left[ y(t) + \frac{v_x}{\Omega} \right] \sin (\Omega \tau) + \frac{v_y}{\Omega}, \]  \hspace{1cm} (2.20a)
\[ y(t + \tau) = -\left[ x(t) - \frac{v_x}{\Omega} \right] \sin (\Omega \tau) \]
\[ + \left[ y(t) + \frac{v_x}{\Omega} \right] \cos (\Omega \tau) - \frac{v_x}{\Omega}, \]  \hspace{1cm} (2.20b)
\[ z(t + \tau) = z(t) + v_z \tau. \]  \hspace{1cm} (2.20c)

The canonical formulation has two advantages: it generates a stable algorithm and this approximation is symplectic (see Fig. 2.1). However, for systems with interacting particles we will see that it is convenient to have a drift operator that is independent of \( \Omega \), which is not the case for the canonical formulation (although this can be remedied by introducing a further splitting).

Non-canonical approximation

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

\[ \dot{x} = v_x; \quad \dot{v}_x = a_x + \Omega^2 x + 2\Omega v_y, \]
\[ \dot{y} = v_y; \quad \dot{v}_y = a_y + \Omega^2 y - 2\Omega v_x, \]
\[ \dot{z} = v_z; \quad \dot{v}_z = a_z. \]  \hspace{1cm} (2.21)
Chapter 2: High-order hybrid N-body methods for compound systems

Figure 2.1: Energy error as a function of time of a particle in a rotating frame. Here we use a second order rotating Bridge with a timestep of 1 Myr.

We split Eqs. 2.21 to build the kick and drift operators. The set of equations that represent the kick operator $K(\tau)$ is the following:

\[
\begin{align*}
\dot{x} &= \dot{y} = \dot{z} = 0, \\
v_{x} &= a_{x} + \Omega^{2}x + 2\Omega v_{y}, \\
v_{y} &= a_{y} + \Omega^{2}y - 2\Omega v_{x}, \\
v_{z} &= a_{z}.
\end{align*}
\] (2.22)

The solution of these equations is
2.2 Method

\[ v_x(t + \tau) = \left[ v_x(t) - \left( \frac{a_y + \Omega^2 y}{2\Omega} \right) \right] \cos(2\Omega \tau) \]
\[ + \left[ v_y(t) + \left( \frac{a_x + \Omega^2 x}{2\Omega} \right) \right] \sin(2\Omega \tau) \]
\[ + \frac{a_y + \Omega^2 y}{2\Omega} , \quad (2.23a) \]
\[ v_y(t + \tau) = \left[ v_x(t) - \left( \frac{a_y + \Omega^2 y}{2\Omega} \right) \right] \sin(2\Omega \tau) \]
\[ + \left[ v_y(t) + \left( \frac{a_x + \Omega^2 x}{2\Omega} \right) \right] \cos(2\Omega \tau) \]
\[ - \frac{a_x + \Omega^2 x}{2\Omega} , \quad (2.23b) \]
\[ v_z(t + \tau) = v_z(t) + a_z \tau . \quad (2.23c) \]

Here the vector \( \mathbf{a} \) corresponds to the acceleration of the particle due to the external galactic potential \( U_{\text{ext}}(\mathbf{r}) \), and \( \tau \) is the Rotating BRIDGE timestep.

The drift operator \( D(\tau) \) on the other hand, is represented by the following set of equations:

\[ \dot{v}_x = \dot{v}_y = \dot{v}_z = 0 , \]
\[ \dot{x} = v_x , \]
\[ \dot{y} = v_y , \]
\[ \dot{z} = v_z . \quad (2.24) \]

The solution of these equations is

\[ x(t + \tau) = x(t) + v_x(t + \tau) \tau , \quad (2.25a) \]
\[ y(t + \tau) = y(t) + v_y(t + \tau) \tau , \quad (2.25b) \]
\[ z(t + \tau) = z(t) + v_z(t + \tau) \tau . \quad (2.25c) \]

By using the non-canonical approximation, a second order rotating BRIDGE can be constructed in the following way:
Chapter 2: High-order hybrid N-body methods for compound systems

\[ \text{RBridge}_2(\tau) = K(\tau/2) \cdot D(\tau) \cdot K(\tau/2), \quad (2.26) \]

where the operators \( K(\tau) \) and \( D(\tau) \) are described by Eqs. 2.23 and 2.25 respectively. Consequently, every \( \tau/2 \) a star receives a velocity kick due to the external galactic potential and every \( \tau \) the position of the star is updated.

The Rotating Bridge can be easily 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 \( \Omega \) is given by:

\[ H = H_A + H_{\text{int}}, \quad (2.27) \]

where

\[ H_A = \sum_{i \in A} \frac{||p_i||^2}{2m_i} - \sum_{i \neq j \in A} \frac{Gm_i m_j}{||r_i - r_j||}, \]

\[ H_{\text{int}} = \sum_{i \in A} \left[ U_{\text{ext}}(r_i) - (\Omega \times r_i) \cdot p_i - \frac{1}{2} m_i ||\Omega \times r_i||^2 \right]. \quad (2.28) \]

The temporal evolution of the system in a second order approximation is given by Eq. 2.26, which can also be written as:

\[ e^{H_{\text{int}} \tau} e^{H_A \tau} e^{H_{\text{int}} \tau}. \quad (2.29) \]

Here the term \( e^{H_{\text{int}} \tau} \) 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:

\[ x(t + \tau) = x(t) + v(t + \tau/2) \tau, \quad (2.30) \]

\[ v'(t + \tau/2) = v(t + \tau/2) + a_{cc} \tau/2. \quad (2.31) \]

Thus, the evolution of system \( A \) during a rotating Bridge timestep \( \tau \) is given by the following steps:

i) At \( \tau/2 \) the system receives a velocity kick due to the external potential of its parent galaxy (Eqs. 2.23). This velocity is referred to as \( v(t + \tau/2) \).
ii) The positions of the stars are updated for a time step $\tau$ (Eq. 2.30). Additionally, the velocities of the stars are updated once more after evolving system $A$ through a direct N-body code (Eq. 2.31).

iii) The system receives a velocity kick for another $\tau/2$ (Eqs. 2.23). This kick is computed by using the previous velocity $v'(t + \tau/2)$. Note that the only difference between the rotating and classical BRIDGE is in the kick operator. In particular, the code evolution operator $e^{HA\tau}$ 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 increased by applying the drift and kick operators in accordance with Eqs. 2.11 and 2.12. In this case a higher order rotating BRIDGE can be generated.

2.2.4. Bridge with post-Newtonian Corrections

In order to extend BRIDGE to include post-Newtonian (PN) corrections a special treatment is needed so that the velocity dependency in the PN terms can be handled correctly. Here we adopt the recipe developed in Hellström & Mikkola [2010], where an auxiliary velocity, $w_i(t)$ with $w_i(t = 0) = v_i(t = 0)$, is introduced in order to make the time evolution operators separable, thus allowing the use of an explicit leapfrog-like algorithm for implement the PN corrections.

In BRIDGE, the operators $e^{\tau H_{Sk}}$ and $e^{\tau H_{int_i} S_l}$ in eqs. 2.9 and 2.10 can be associated to different solvers, which evolve the different sub-systems in a simulation. If, for example, the $j$-th sub-system requires PN corrections and we assume that interactions with others sub-systems can be treated without PN corrections, then only the operator $e^{\tau H_{Sk}}|_{k=j}$ in eq. 2.9 needs to be modified, which consists of a simple substitution of a regular Newtonian solver by a PN one for this particular $j$-th sub-system.

In the more complex case where several interacting sub-systems require PN corrections, both operators $D(\tau)$ and $K(\tau)$ in eqs. 2.9 and 2.10 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
Chapter 2: High-order hybrid N-body methods for compound systems

A number $Q^{PN}$ require PN corrections, so that $Q = Q^N + Q^{PN}$. Moreover, we define $S^N$ as the set of “Newtonian” sub-systems and $S^{PN}$ as the set of “post-Newtonian” sub-systems, so that $S = S^N \cup S^{PN}$. In this way, operator $D(\tau)$ in eq. 2.9 can be extended into the following expression

$$\tilde{D}(\tau) = \prod_{k \in S^N} e^{\tau H_{SN}^k} \prod_{k \in S^{PN}} e^{\tau H_{SPN}^k}. \quad (2.32)$$

Here $e^{\tau H_{SN}^k}$ represents a Newtonian solver for the $k$-th sub-system in $S^N$ and $e^{\tau H_{SPN}^k}$ represents a PN solver for the $k$-th sub-system in $S^{PN}$. We clarify here that by PN solver we mean a code that evolve the particles under the total acceleration $a = a^N + a^{PN}$, not just $a^{PN}$ as might be implied by our notation.

The operator $K(\tau)$ in eq. 2.10 can be extended as follows. We first define some auxiliary kick operators,

$$K^{N\leftrightarrow N}(\tau) = \prod_{k \neq l} e^{\tau H_{SN}^{int} S^N_{k} S^N_{l}}, \quad (2.33)$$

$$K^{N\leftrightarrow PN}(\tau) = \prod_{k \in S^N} \prod_{l \in S^{PN}} e^{\tau H_{SN}^{int} S^{PN}_{k} S^{PN}_{l}}, \quad (2.34)$$

$$K^{PN\leftrightarrow PN}(\tau) = \prod_{k \neq l} e^{\tau H_{SPN}^{int} S^{PN}_{k} S^{PN}_{l}}, \quad (2.35)$$

from which follows that the extended kick operator for BRIDGE with PN corrections can be written as:

$$\tilde{K}(\tau) = K^{N\leftrightarrow N}(\tau/2) \cdot K^{N\leftrightarrow PN}(\tau/2) \cdot K^{PN\leftrightarrow PN}(\tau) \cdot K^{N\leftrightarrow N}(\tau/2) \cdot K^{N\leftrightarrow PN}(\tau/2)$$

$$= K^{PN\leftrightarrow PN}(\tau/2) \cdot K^{PN\leftrightarrow PN}(\tau/2) \cdot K^{N\leftrightarrow N}(\tau) \cdot K^{N\leftrightarrow PN}(\tau/2) \cdot K^{PN\leftrightarrow PN}(\tau/2). \quad (2.36)$$

Eq. 2.33 represents the Newtonian kick due to the interaction between “Newtonian” sub-systems, and is identical to the original definition in eq. 2.10.
Eq. 2.34 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 PN corrections should be included or not. Such decision could be based, for example on the distance between the two interacting sub-systems. Eq. 2.35 represents the PN kick due to the interaction between “post-Newtonian” sub-systems. Finally, eq. 2.36 represents the extended kick operator to be used in BRIDGE with PN corrections.

2.3. Tests and Applications

2.3.1. Implementation

The above new BRIDGE integrators can be implemented in a number of different ways. For the tests here we will use the Astrophysical Multipurpose Software Environment (AMUSE) [Pelupessy et al., 2013; Portegies Zwart et al., 2013]. AMUSE is a python software environment for astrophysical simulations. AMUSE presents a wide variety of astrophysical codes using homogeneous interfaces, simplifying their use. AMUSE contains codes from different domains, amongst which a number of gravitational dynamics codes.

For the BRIDGE integrators here, it is important to note that the gravitational dynamics codes in AMUSE provide convenient implementations of the evolution operator $e^{tH}$ in the form of an evolve_model method on the interface. This can then be combined with simple force evaluations (evaluated by other component codes or implemented on the interface level) to provide $e^{tH_{int}}$, the interaction type operators. Thus the integrators presented above can be quickly formulated using ready made ‘building blocks,’ selecting the appropriate integrators from the ones available, balancing the needs for the precision in the calculations with the performance or special features of a given code.

To illustrate this, in Fig. 2.2 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) in its parent galaxy (the galaxy set). The initial realization can be constructed within AMUSE using e.g. a Plummer sphere model. Other realizations are implemented in AMUSE such as king profiles [King, 1966] and fractal distributions [Goodwin
Chapter 2: High-order hybrid N-body methods for compound systems

(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 )

Figure 2.2: Example usage of the Bridge integrator through the Amuse framework.

& Whitworth, 2004]. At (1) we initialize the N-body integrator to calculate the internal evolution of the star cluster. In this case, we use the Hermite integrator [Hut et al., 1995]. For a list of gravity codes implemented in Amuse, we refer the reader to Pelupessy et al. [2013]. (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 timestep for the coupling timescale. For a Bridge in rotating coordinates, the initialization is made by typing: sys=Rotating_Bridge(). (6) and (7) We couple the gravity code and the galaxy into the Bridge integrator. The method add_system has two arguments: 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 gravity code. In line (7) the particles in the gravity code will kick the galaxy. In this way we ensure that both cluster and parent galaxy will be evolved self-consistently. (8) We evolve the compound system for a set amount of time.

2.3.2. High Order Bridge

To evaluate the high-order formulation of bridge and show that we indeed can construct high-order compound methods this way, we first calculate the evolution of a very simple system, namely a stable hierarchical quadruple system consisting of two binary systems orbiting each other. The total system comprises of 4 equal mass bodies [for a total mass of 1 in N-body units, Heggie
2.3 Tests and Applications

Figure 2.3: Left: Energy error for quadruple system test. Drawn lines: second, fourth and sixth order bridge coupling results (left to right, red green and blue) and Kepler solver subsystem. Dotted lines: same with leapfrog subsystem. Right: CPU time for the quadrupole test. Note the CPU time is indicative of the scaling only, because for such a small problem the overhead of the bridging method is large.

& Mathieu, 1986], and the orbits are coplanar with the two binary orbits having a semi-major axis $a = 1/8$ and moderate eccentricity ($\epsilon = 0.5$). The two
Chapter 2: High-order hybrid N-body methods for compound systems

![Graph showing energy error and CPU time vs. fraction of orbital period and CPU time respectively.]

**Figure 2.4:** Left: Energy error for orbiting King clusters test. Plotted is the energy error as function of the BRIDGE timestep (given as fraction of the orbital period). Green: integration using BRIDGE$_2(\tau)$, red: results for BRIDGE$_4(\tau)$ Right: CPU times. Note the CPU time is indicative of the scaling only, because for such a small problem the overhead of the bridging method is large.

binaries in turn are set-up in an orbit with $a = 1.$ and $\epsilon = 0.5$. The two binary systems are integrated in separate codes, and their interaction is bridged using
the standard leapfrog bridge, and a fourth order and sixth order method. The binaries can be evolved using a Kepler solver, which calculates the evolution of the subsystems exact to machine precision, or a leapfrog integrator, which is second order.

If we investigate the results for the bridged Kepler solvers in figure 2.3, where we have plotted the relative energy error as function of the bridge timestep, we see that the Bridge schemes show the expected error behavior, appropriate for their respective order (note that the sixth order integrator shows saturation at machine precision as expected). If we change the integrator of the subsystems to a leapfrog integrator (dotted lines), at small time steps ($<10^{-3}$) the energy error is $\sim 3$ orders of magnitude higher with respect to the integration with the Kepler solver. This energy error is caused by the integration of the subsystems, which show the expected second order behavior (we keep the ratio of the timestep in the subsystem to the bridge timestep constant). This tests shows that the high-order Bridge schemes can be effectively applied in cases where the compound integrators allow for the extra precision.

Improved precision beyond the common $\sim 16$ decimal places and $\sim 10^{-12}$ energy conservation per step is hard, and requires special treatment of the force evaluation and timestepping. Recently, Boekholt & Portegies Zwart [2015] designed a gravitational $N$-body method in which the precision can be tuned to arbitrary precision.

In figure 2.4 we show the energy error behavior for a similar test where we put two 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 length units. 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, and bridged using either $\text{Bridge}_2(\tau)$ or $\text{Bridge}_4(\tau)$. As we can see in figure 2.4 the energy error shows the expected order behavior in the two cases.

### 2.3.3. Rotating bridge

To evaluate the accuracy of the Rotating Bridge, we show in Fig. 2.5 the maximum fractional energy error as a function of bridge timestep of a star moving in different galactic potentials representing the Milky Way. In this plot, the Galaxy is represented by: a pure axisymmetric potential (top panel);
an axisymmetric + bar potential (middle panel); and an axisymmetric+ spiral arms potential (bottom panel).

The axisymmetric potential was modelled by taking into account the parameters of Allen & Santillán [1991]. The central bar was modelled with a Ferrers
potential [Ferrers, 1877] and the spiral arms were modelled as perturbations of
the axisymmetric Galactic potential following the tight winding approximation
[Antoja et al., 2011]. Both bar and spiral arms rotate as rigid bodies with dif-
fferent pattern speeds. For further details on the Galactic model, we refer the
reader to Martínez-Barbosa et al. [2015]. Contrary to Fig. 2.3, in this example
we use physical units, i.e. the Bridge timestep is in Myr.

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

By analyzing the figure, we can observe that the fourth and sixth order
integrators generate a better energy conservation compared to the normal (sec-
ond order) rotating Bridge scheme. Additionally, note that the energy error
is smaller in circular than in more eccentric orbits for a given Galaxy model
and Bridge timestep. This is expected, because the external tidal field is
constant in circular orbits. However, note that even in eccentric orbits, a high-
order rotating Bridge can allow a good energy conservation at small timesteps
($dE/E_0 < 10^{-7}$ at $\tau \leq 1$ Myr). We computed the energy error of a star in the
eccentric orbit for 100 orbital periods (more than a Hubble time) and for each of
the Galaxy models explained above. We found that the fractional energy error
is of the order of $10^{-7}$ when using $\tau \leq 1$ Myr. Therefore, the high-order Ro-
tating Bridge is a suitable scheme to compute the stellar motion in analytical
Galaxy models.

In Fig. 2.6 we show the evolution of a star cluster in the Milky Way by using
a high-order rotating Bridge. The cluster is modelled with a Plummer sphere
of 1700 stars. The Galaxy is modelled with an analytic prescription that con-
tains an axisymmetric potential (bulge, disk and a dark matter halo) together
with a bar and spiral arms. We evolved the star cluster during 4.6 Gyr. We
use the Huayno code [Pelupessy et al., 2012] to resolve the gravitational effects
among the stars and a sixth order rotating Bridge to couple the N-body code
with the Galaxy model. Given that we use the Amuse framework to perform
the simulation, we added the SeBa population synthesis code [Portegies Zwart
& Verbunt, 1996; Toonen et al., 2012] to model stellar evolution effects in the
Chapter 2: High-order hybrid N-body methods for compound systems

Figure 2.6: Evolution of a star cluster in the Milky Way using a rotating Bridge integrator to a sixth order. Left: Distribution of the stars (yellow points) in the Galactic disk after 4.6 Gyr of evolution. The white line corresponds to the orbit of the cluster before dissolution. The pink dashed lines represent the present-day potential of the spiral arms. Middle: Bound mass of the star cluster as a function of time. Right: Lagrange radii of the star cluster as a function of time.

cluster. The Bridge timestep used was such that the maximum energy error in the simulation was of the order of $10^{-7}$.

During 4.6 Gyr of evolution, the star cluster moves in an orbit with $\epsilon = 0.1$, its pericenter being at 8.6 kpc with respect to the Galactic center (e.g. solid white line, left panel Fig. 2.6). During the first 0.1 Gyr of evolution, the star cluster loses mass due to stellar evolution (see middle panel Fig. 2.6). After $\sim 0.1$ Gyr, two-body relaxation effects and the external tidal field of the Galaxy are the main mechanisms that dissolve the star cluster. Both stellar evolution and two-body relaxation effects make the cluster expands (see right panel Fig. 2.6). This phenomenon is called expansion phase [Gieles et al., 2011] and it has been observed by several authors previously [i.e Madrid et al., 2012, and references therein]. After $\sim 0.6$ Gyr, the effect of the tidal field becomes more important. This produces a gradual evaporation of the star cluster (see right panel Fig. 2.6) which contributes with its completely disruption in the Galaxy at $\sim 1.5$ Gyr (middle panel Fig. 2.6). After 4.6 Gyr, the stars are spread in the second and third quadrants of the Galaxy, with galacto-centric distances of around 8 kpc. (see left panel Fig. 2.6).
2.4. Discussion and Conclusions

The classical Bridge scheme [Fujii et al., 2007] provides an elegant and convenient integrator under the conditions discussed in section 2.2.1, namely that there are multiple subsystems where there is a separation in timescales of the interactions, and/or there are different regimes of gravitational dynamics at play.

We have shown that this coupling can be extended to higher orders, as well as multiple interacting systems and deeper hierarchies of subsystems. We name this suspension-bridge. Suspension-bridge provides a high-order hierarchical symplectic coupling between two or more subsystems. The formulation is relatively independent of the actual implementation of the needed evolution operators and we have implemented them within the AMUSE framework. Here the different numerical integrators available provide ready-made building blocks to combine in different type of integrators that can be tailored towards the specific application and precision required. We tested the method on gravitational systems in which we coupled Newtonian gravitational dynamics with Newtonian forces via a direct N-body codes, confirming the energy conservation has the right order for the second, fourth and sixth order coupling strategies, up to machine precision \( (dE/E \approx 10^{-15}) \), if the requirements on the component integrators are met. Within AMUSE up to tenth order methods are available. The method is hierarchical and the same expansion can be repeated to construct more complicated suspension schemes.

We have also presented a variant of Bridge for 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 only difference with the classical Bridge is that the kick operator effecting the interactions between the different subsystems must be adapted. Note that the evolution in between the kicks has not changed form (and can use the same integrators as before). In the non-canonical case the resulting integrator is not formally symplectic (and this manifests itself as a small drift in energy), but for practical applications (e.g. a stellar cluster in a galaxy potential) the resulting drift is neglectable (due to the relative low number of orbits) especially
Chapter 2: High-order hybrid N-body methods for compound systems

if combined with a higher order integration scheme.

Acknowledgements

We thank Tjarda Boekholt, Adrian Hamers and Arjen van Elteren for enriching discussions. This work was supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (IAP P7/08 CHARM) and by the Netherlands Research Council NWO (grants #643.200.503, #639.073.803 and #614.061.608) and by the Netherlands Research School for Astronomy (NOVA). The numerical computations were carried out using the Little Green Machine at Leiden University.