Dynamical evolution of massive black holes in galactic-scale N-body simulations – introducing the regularized tree code ‘rVINE’

Simon J. Karl,1⋆ Sverre J. Aarseth,2 Thorsten Naab,3 Martin G. Haehnelt1 and Rainer Spurzem4,5,6

1Institute of Astronomy and Kavli Institute for Cosmology, Madingley Road, Cambridge CB3 0HA, UK
2Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK
3Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str. 1, D-85741 Garching bei München, Germany
4National Astronomical Observatories of China, Chinese Academy of Sciences, 20A Datun Rd., Chaoyang District, 100012 Beijing, China
5Astronomisches Rechen-Institut, Zentrum für Astronomie, University of Heidelberg, Mönchhofstrasse 12-14, D-69120 Heidelberg, Germany
6Kavli Institute for Astronomy & Astrophysics, Yi He Yuan Lu 5, Hai Dian Qu, 100871 Beijing, China

Accepted 2015 June 29. Received 2015 May 13; in original form 2014 October 20

ABSTRACT
We present a hybrid code combining the OpenMP-parallel tree code VINE with an algorithmic chain regularization scheme. The new code, called ‘rVINE’, aims to significantly improve the accuracy of close encounters of massive bodies with supermassive black holes (SMBHs) in galaxy-scale numerical simulations. We demonstrate the capabilities of the code by studying two test problems, the sinking of a single massive black hole to the centre of a gas-free galaxy due to dynamical friction and the hardening of an SMBH binary due to close stellar encounters. We show that results obtained with rVINE compare well with NBODY7 for problems with particle numbers that can be simulated with NBODY7. In particular, in both NBODY7 and rVINE we find a clear N-dependence of the binary hardening rate, a low binary eccentricity and moderate eccentricity evolution, as well as the conversion of the galaxy’s inner density profile from a cusp to a core via the ejection of stars at high velocity. The much larger number of particles that can be handled by rVINE will open up exciting opportunities to model stellar dynamics close to SMBHs much more accurately in a realistic galactic context. This will help to remedy the inherent limitations of commonly used tree solvers to follow the correct dynamical evolution of black holes in galaxy-scale simulations.

Key words: black hole physics – methods: numerical – stars: kinematics and dynamics – galaxies: evolution – galaxies: nuclei.

1 INTRODUCTION
The presence of supermassive black holes (SMBHs; e.g. Rees 1984) with masses of $10^6 M_{\odot} < M_{BH} < 10^{10} M_{\odot}$ hosted in the central regions of virtually all massive spheroids in the nearby Universe, including the Galactic bulge, is now firmly established (Richstone et al. 1998; Kormendy & Ho 2013). SMBHs must have also been present at much earlier phases of our Universe, powering active galactic nuclei (AGNs) and quasars from only a few hundred million years after the Big Bang throughout cosmic history (e.g. Lynden-Bell 1969; Fan et al. 2001; Civano et al. 2011; Mortlock et al. 2011).

We now think of SMBHs as integral components of galactic nuclei, possibly playing a decisive role in shaping the structure and morphology, as well as the gas and thus stellar content of massive galaxies. The Λ cold dark matter paradigm for structure formation together with a number of surprisingly tight relations between the SMBH masses and fundamental properties of the galactic bulges hosting them – e.g. the spheroid luminosity (Kormendy & Richstone 1995) and mass (Magorrian et al. 1998; Häring & Rix 2004), and the stellar velocity dispersion (Gebhardt et al. 2000; Ferrarese & Merritt 2000; Tremaine et al. 2002) – suggests co-evolution of the hierarchically growing galaxies and their central black holes (e.g. Lynden-Bell 1969; Silk & Rees 1998; Kauffmann & Haehnelt 2000).

At present, variants of particle-based smoothed particle hydrodynamics (SPH; e.g. Wadsley, Stadel & Quinn 2004; Springel 2005), grid-based adaptive mesh refinement (e.g. Kravtsov, Klypin & Khokhlov 1997; Teyssier 2002) codes, or moving mesh codes (Springel 2010), are the methods of choice for numerical simulations of cosmological galaxy formation, exploiting the high

* E-mail: skarl@ast.cam.ac.uk
dynamic range and spatial flexibility in resolution. The gravity solvers in these codes either employ a particle-mesh scheme or are tree based and assume the simulated system to be collisionless, with two-body relaxation time-scales exceeding the age of the system. In 'tree' algorithms (Barnes & Hut 1986, see also McMillan & Aarseth 1993 for a collisional tree code) the gravitational force on a single particle from a distant group of particles is approximated by a multipole expansion about the group’s centre of mass, and the N-body particles represent massive tracer particles that sample the underlying smooth gravitational potentials. To reduce the graininess of the potential, gravitational forces are ‘softened’ on small spatial scales and the softening length $\varepsilon$ is the natural resolution limit of the code. Unfortunately, this also means that close two-body encounters with a massive body such as an SMBH can – by construction – not be computed accurately. An alternative, much more cost-intensive approach of calculating the gravitational forces in an astrophysical system is the direct summation of each particle’s gravity on every other particle in the system. Combined with high-order integrators, this method is a very accurate way to calculate the gravitational forces and is widely used to simulate collisional N-body systems (e.g. Aarseth 1999, 2003b; Hut et al. 2010).

Owing to the inherent limitations in the numerical methods, studies of stellar dynamics in the vicinity of SMBHs to date could only probe single separate aspects of the full problem. On the one hand, direct N-body simulations on SMBH binary dynamics in the centres of isolated galaxies or merger remnants (e.g. Ebisuzaki, Makino & Okumura 1991; Milosavljević & Merritt 2001, 2003; Berczik et al. 2006; Merritt, Mikkola & Szell 2007; Khan, Just & Merritt 2011; Preto et al. 2011; Khan et al. 2012) use idealized initial conditions to represent the inner parts of galaxies in which the massive binaries are then embedded and evolved. Due to the steep scaling of the required computing time with particle number, of order $O(N^2)$, these studies are still rather limited in the particle number, hindering a self-consistent treatment of full galactic environments. On the other hand, simulations of galaxy mergers or cosmological simulations of structure formation including SMBHs (e.g. Springel, Di Matteo & Hernquist 2005; Di Matteo et al. 2008; Booth & Schaye 2009; Johansson, Naab & Burkert 2009; Choi et al. 2012; Martizzi, Teyssier & Moore 2012; Sijacki et al. 2015) have difficulties to capture the dynamics of SMBHs and their surrounding stars below the resolution limit. This leads to uncertainties, e.g. in the dynamical friction time-scales for the SMBHs, and affects the density and velocity profiles of the stellar background through interactions with the SMBHs, and the hardening and merging time-scales of close SMBH pairs. The latter is generally assumed ‘a priori’ to happen fast in these simulations, much reducing the accuracy and predictive power of such simulations. Hence, there is still substantial uncertainty in the current understanding of the dynamical evolution of SMBHs and their surrounding star clusters in realistic cosmological settings, which directly feeds back into uncertainties in our understanding of how SMBH singles or multiples influence the structure of galaxies.

The main goal of this paper is to help remedy these shortcomings by combining the best parts of the two numerical approaches: a regularization method to efficiently and accurately compute the dynamics close to the black holes and a fast tree code to treat the global galactic dynamics. This goes in line with the development of similar recent hybrid codes (as discussed in the next section) and a software interface designed to efficiently combine different stand-alone code architectures (Portegies Zwart et al. 2009, 2013). With the new algorithm, we will be in a position to better take into account the relevant dynamical processes regarding the interaction between SMBHs and the stars in their environments and other (SM)BHs, in principal without limitations on the spatial and temporal resolution down to scales where other types of physical phenomena become important, e.g. gravitational wave induced coalescence of SMBH binaries or the tidal disruption of low-angular-momentum stars (e.g. Pretorius 2005; Lodato, King & Pringle 2009). This might be an important next step towards investigating the dynamical co-evolution of SMBHs and their host galaxy nuclei in a self-consistent manner in galaxy-scale or cosmological simulations.

In this paper, we present the details of our hybrid N-body code which will help us to focus on the role of gravitational dynamics in the interplay between SMBHs and realistic representations of the central regions of their host galaxies. The paper is structured as follows. In Section 2, we describe the details and structure of the new hybrid code ‘VINE’ and test its performance in Section 4 after we have described the numerical set-up in Section 3. First tests on the code in comparison with the direct N-body code NBODY7 and the tree code VINE are presented in Section 5. We discuss our results in Section 6 and, finally, summarize and draw our conclusions in Section 7.

## 2 A NEW REGULARIZED TREE CODE

In this section, we present the structure of the regularized tree code called ‘VINE’. A number of currently available N-body codes employ regularization techniques intended for the integration of strong gravitational interactions but are primarily developed for integrations of collisional systems such as star clusters or the dense central regions of galactic nuclei (see Aarseth 2003a, 2007; Harfst et al. 2008; Gaburov, Harfst & Portegies Zwart 2009; Nitadori & Aarseth 2012; Berczik, Spurzem & Wang 2013; Wang et al. 2015). In addition, there are two recent hybrid codes combining tree and N-body codes. The IRBIDE code (Fujii et al. 2007) uses a simple fixed time-step oct-tree, while the BONSAI code (Bédorf, Gaburov & Portegies Zwart 2012) is an oct-tree run entirely on graphics processing units (GPUs). Both hybrids use a symplectic mapping method (Wisdom & Holman 1991) to couple the tree to a direct N-body algorithm with a fourth-order Hermite integrator.

In VINE, we compute the evolution of a subsystem of particles near the black hole by means of a regularization method, while regions of the galaxy further out with long relaxation times which are basically unaffected by the presence of the black hole, are integrated by a collisionless tree code. To this end, we combine two published algorithms: a version of the tree/SPH code VINE (Wetzstein et al. 2009; Nelson, Wetzstein & Naab 2009) and the algorithmic regularization (AR) chain method (Mikkola & Merritt 2006, see also Hellström & Mikkola 2010 for a general discussion). The latter was kindly provided as a stand-alone code by Seppo Mikkola.

VINE is an OpenMP-parallelized tree/SPH code employing a binary tree algorithm and an individual hierarchical block time-step scheme (Wetzstein et al. 2009; Nelson et al. 2009).1 The AR-chain method is an efficient and extremely accurate method to study close dynamical few-body encounters and is capable of handling even (repeated) two-body collisions (Mikkola & Tanikawa 1999a, b; Preto & Tremaine 1999; Mikkola & Aarseth 2002). This is achieved by

1 Note that in the present paper we will discuss new developments done in the parts of VINE related with the leapfrog integrator, an individual time-step scheme and no smoothed particle hydrodynamics.
effectively removing any singular behaviour in the equations of motion by a time transformation in the Hamiltonian of the regularized subsystem. The coordinates and the (original) time and their respective ‘momenta’ are integrated using a simple leapfrog method. In addition, a Bulirsch–Stoer extrapolation method (Gragg 1965; Bulirsch & Stoer 1966) is applied to guarantee high accuracy, as well as a chain concept of smallest interparticle vectors to reduce round-off errors (Mikkola & Aarseth 1990, 1993). In our present version of the AR-chain, chain particles are sorted according to their gravitational forces, not according to their distance. It also includes a method to handle velocity-dependent forces, which allows us, in principle, to treat additional viscous and relativistic terms in the regularized force calculations (Mikkola & Merritt 2006). The new code, however, is purely Newtonian at the present stage.

There exist other regularization schemes which are comparable to the AR-chain in accuracy and speed, e.g. the KS-wheelspoke (Zare 1974; Aarseth 2007) and the KS-chain method (Mikkola & Aarseth 1993). Due to limitations of these methods in the context of stellar dynamics around one or several SMBHs we decided to use the AR-chain as our principal regularization algorithm. For example, large mass ratios may lead to a loss in numerical accuracy for the less massive bodies in a KS-chain, whereas the wheelspoke has difficulties in treating multiple heavy bodies on an equal footing.

The gravitational forces for the majority of the particles are computed with VINE’s fast binary tree scheme without regularization, using a pre-defined spline or Plummer softening, while particles near a designated massive particle (SMBH) become members of a compact subsystem which is integrated in the AR-chain in its centre-of-mass reference frame.

Chain integration starts if any particle comes closer to the SMBH than \( r_{\text{infl}}^BH < r_{\text{chain}, 0} \), where \( r_{\text{chain}, 0} \) defines the initial chain size and is an input parameter which has to be chosen at the start of a simulation as described in the following.

The intended purpose for including particles in the chain integration is twofold. First, we want to accurately follow close orbits near the black holes, i.e. within a fair fraction of the SMBHs’ influence radii, and secondly, we need to overcome the limitations posed by the gravitational softening, i.e. for encounters within \( \sim \) a few times \( \epsilon \equiv \max (\epsilon^{BH}, \epsilon^\star) \), where \( \epsilon^{BH} \) and \( \epsilon^\star \) denote the gravitational softening lengths of the stellar particles and SMBHs, respectively. Hence, we determine the initial chain radius,

\[
r_{\text{chain}, 0} = \max (\alpha \cdot r_{\text{infl}}, \beta \cdot \epsilon),
\]

at the start of the chain, where \( \alpha \) and \( \beta \) are input parameters and \( r_{\text{infl}} \) is the gravitational influence radius of the SMBH. For practical purposes, we define \( r_{\text{infl}} = \min (r_{\text{infl}}^BM, r_{\text{infl}}^\star) \) using two commonly used proxies for the gravitational influence radius of the SMBH, being (1) the radius enclosing twice the mass of the SMBH, \( r_{\text{infl}}^BM = r(<2\,M_{BH}) \), and (2) the radius within which the gravitational force of the SMBH dominates over the self-gravity of the stellar background, \( r_{\text{infl}}^\star = GM_{BH}/\sigma^2 \). Here, \( M_{BH} \) is the mass of the black hole and \( G \) the gravitational constant. The velocity dispersion \( \sigma \) is determined by averaging over the nearest 50 particles outside the chain. We generally find good results for \( 1 \leq \alpha \leq 1.5 \) and \( 1 \leq \beta \leq 2 \).

The chain’s centre of mass is treated as a massive particle in the tree,\(^3\) i.e. it is included in the tree force calculations and advanced in time within the tree code. The chain particle is advanced in time on the smallest tree time-step and we formally set the gravitational softening of the chain particle to zero in the tree code. Tree particles that become members of the chain are converted into ‘ghost’ particles with no further advancement in the tree. This is done by assigning a very small, but finite-sized mass in the tree code data structure, rendering their contribution to the gravitational forces negligible. Furthermore, ghost particles are not allowed to determine the size of the time-step in the tree. If the chain is active, the member particles within the chain are advanced using the AR-chain integration, every time particles in the tree code on the smallest time-step level are being advanced.

The equations of motion of the chain members include external forces exerted by a set of nearby tree particles we call ‘perturbers’, which are identified via a tidal criterion,

\[
r_{j, \text{CoM}} < \left( \frac{2}{\gamma_{\text{crit}} M_{\text{chain}}} \right)^{\frac{1}{3}} \times r_{\text{crit}}, \tag{2}
\]

where \( m_j \) is the mass of perturber \( j \), \( M_{\text{chain}} \) is the total mass in the chain and \( \gamma_{\text{crit}} \) a dimensionless parameter which defines the relative tidal perturbation on the chain. We typically set \( r_{\text{crit}} = \min (r_{\text{chain}}, r_{\text{chain}, 0}) \) in the simulations presented here. For better accuracy, the perturber positions relative to the centre-of-mass particle are predicted (to first order) to the current time at each force calculation in the AR-chain. The perturber forces have to be predicted many times during the numerical substep cycles of the Bulirsch–Stoer extrapolations. To improve the overall performance of the chain part of the code we, therefore, have implemented parallel routines for the predictions of the perturbers and the computation of the perturber forces.

Likewise, the force contributions of individual (‘resolved’) chain particles are added to the gravitational forces for the perturber particles during the force updates in the tree. The gravitational force on the chain particle is also corrected by resolving the chain particles. If the chain particle is active, we first calculate the gravitational force exerted on it by particles within the tree code, but subtract the tree forces (i.e. direct, softened \( N \)-body interactions) from the perturber particles again. Then the (direct) gravitational force from the perturbers on each individual chain particle is calculated, and added up as a correction to the chain particle’s force according to

\[
\vec{a}_{\text{CoM}, \text{corr}} = \frac{1}{M_{\text{chain}}} \sum_{i=1}^{N_{\text{chain}}} m_{\text{chain}, i} \vec{a}_{\text{chain}, i}.
\]

After advancing the chain one full time-step its membership is updated. Perturber particles are added to the chain if they come closer than the ‘chain radius’, i.e.

\[
r_{j, \text{CoM}} < r_{\text{chain}}, \tag{4}
\]

where we define the chain radius as the largest distance of any chain member relative to the chain centre of mass in the last chain step.

\(^2\) However, the singularities do formally remain in the transformed equations of motion – unlike in schemes based on Kustaanheimo & Stiefel (1965; KS) regularization, which applies a time and a coordinate transformation.

\(^3\) Henceforth, we will call this particle simply the ‘chain particle’, denoting the chain’s centre-of-mass particle that is advanced in the tree code; not to be confused with a single ‘particle in the chain’, which we will equally call a ‘chain member’ from now on.

\[\text{MNRAS 452, 2337–2352 (2015)}\]
Figure 1. Projected trajectories of regularized particles in the chain around a supermassive black hole. The trajectories are shown in a time interval \( \Delta t \approx 1 \), which corresponds to \( \sim \) one crossing time of the subsystem shown.

Figure 2. Illustration of the different integration regions near a regularized massive particle in the hybrid AR-regularized tree code. The centre-of-mass reference body, consisting of the members within the regularized subsystem \( r < r_{\text{chain}} \), is surrounded by a swarm of nearby particles ('perturbers') which are considered as external force terms in the chain force calculations \( r < r_{\text{pert}} \) where the critical radius \( r_{\text{pert}} \) depends on a tidal criterion and, themselves, experience direct \( N \)-body forces from the resolved chain (see text). Further out, we indicate the regime where the direct integration of particles (open circles) switches to the multipole approximations, depending on the acceptance criteria in the tree code (indicated by the groups of particles in the large star symbols).

Particles are removed from the AR-chain if they recede far enough from the chain’s centre of mass

\[
r_{j, \text{CoM}} > Y \times r_{\text{chain}},0 \equiv r_{\text{escape}}
\]

and move radially away from the chain centre. We typically choose \( Y = 1.5 \). Upon removal of a chain member, its mass, position and velocities in the global (i.e. tree) reference frame are restored. If the number of chain members would fall below \( N_{\text{chain}} < 2 \) after an escape the chain is terminated and the remaining two chain members are restored in the tree. Likewise, the chain is terminated in the (unlikely) case that the designated massive particle is removed from the chain – unless the chain membership includes several massive particles, in which case one of the remaining SMBHs is chosen to be the new centre-of-mass particle in the tree code. A representative example of near-Keplerian, perturbed orbits of particles in the chain is illustrated in Fig. 1.

The regularization of particle orbits near a designated (massive) particle, leads to different integration regions for different particles in the code, as illustrated in Fig. 2. With increasing distance to the chain’s centre of mass the particles are either

(i) regularized members of the chain (within the dashed circle in Fig. 2), feeling external gravitational forces from the perturbers (filled circles) only,

(ii) perturber particles which feel the gravitational forces of the individual chain members, or

(iii) tree particles, whose gravitational forces on to the chain particle may either be calculated via direct summation (open circles) or approximated by a multipole expansion of a group of particles (indicated by star symbols), depending on the values for the acceptance criteria chosen in the tree.

For cost reasons, we restrict the total number of chain members and perturbers. We have tested up to values of \( N_{\text{ch}}^{\text{max}} = 250 \) and \( N_{\text{pert}}^{\text{max}} = 5000 \) without any loss in the stability of the code.

In short, a typical time-step in the hybrid code proceeds as follows.

1. Determine the next time-step and active particles for the tree.
2. If the chain is not active, or, the system does not evolve on the smallest time-step, continue with step (6).
3. Integrate the members of the chain using the AR-chain method. Treat the external forces exerted by nearby particles ('perturbers') as perturbations to the members’ gravitational forces.
4. Check for absorption by and escape from the chain. Perform a search to identify the perturbers via a tidal criterion in regular time intervals (see equation 2).
5. Terminate the chain if \( N_{\text{chain}} < 2 \).
(6) Perform a regular leapfrog time-step in the tree code in the ‘drift-kick-drift’ scheme:

(a) Update tree particle positions at the half time-step.

(b) Compute gravitational forces for the active tree particles. If the chain particle or any perturber is active include force corrections due to the gravitational forces between the perturber particles and the resolved members of the chain.

(c) Update tree particle positions and velocities at the full time-step.

(7) Update tree particle time-steps in the individual time-step scheme and update the tree structure if necessary.

(8) If the chain is not active check for particles near the SMBH that fulfill the conditions to begin chain regularization.

3 NUMERICAL SET-UP

In this section, we shortly describe the numerical set-up and the different numerical models we will introduce in the following sections. In particular, we will detail the way the code works by discussing an example simulation and testing its performance for a number of different code parameters in Section 4. In Section 5, we test the code against a number of other currently available N-body codes, such as the vine and gadget-3 tree codes and the NBODY7 direct summation code. All of the vine, gadget, or gadget-3 simulations presented in this paper were run on the COSMOS cluster at DAMTP, Cambridge. For the NBODY7 simulations, we used single nodes on the Wilkes cluster at the High Performance Computing (HPC) Service of the University of Cambridge, consisting of a Dell PowerEdge T620 server à 12 Intel Xeon E5-2670 CPUs plus two NVIDIA K20 GPUs.

As our principal numerical model, we use a non-rotating Hernquist (1990) sphere to represent the galactic nucleus. The Hernquist density profile follows a \( \rho(r) \propto r^{-3} \) power law at small radii \( r \ll r_0 \) while converging quickly, with \( \rho(r) \propto r^{-4} \) to a finite mass at large radii \( r \gg r_0 \). All models are set up with unit total mass \( M_{\text{tot}} = 1 \) and scale radius \( r_0 = 1 \), and the gravitational constant \( G \) is set to unity in all codes used throughout this paper. Note that, for convenience, we will use a system of code units, with a small spherical (dE50) galaxy or a galactic nucleus of mass \( M_{\text{gal}} = 10^{10} M_\odot \) and scalelength \( r_0 = 1 \) kpc yielding time units and velocity units of \( \sim 4.7 \) Myr and \( \sim 207 \) km s\(^{-1} \), respectively. In code units the half-mass radius of the Hernquist sphere \( r_{1/2} = (1 + \sqrt{2}) r_0 \approx 2.41 \) with a half-mass dynamical time of \( t_{\text{dyn,h}} = 27 \). In the different simulations considered here the galaxy is realized with total particle numbers in the range \( 10^5 \sim 10^6 \) of equal-mass stellar particles \( (m_\ast \propto N^{-1}) \) for vine and rVine.4

Note that, throughout the text, added suffixes will state the actual particle numbers of the simulations.

Table 1. Parameters for the different N-body calculations.

| Simulation | \( N_\ast \) | \( M_{\text{tot}} \) | \( r_0 \) | \( m_\ast/M \) | \( \epsilon_\ast \) | \( N_{\text{BH}} \) | \( \epsilon_{\text{BH}} \) | \( r_{\text{BH}} \) | \( M_{\text{BH}}/M_{\text{tot}} \) | \( \epsilon_{\text{BH}} \) |
|------------|----------|----------------|----------|----------------|--------|----------------|----------------|----------------|----------------|----------------|
| A_Nbody7   | 100k     | 1.0            | 1.0      | \( 10^{-5} \) | –      | 1              | 2.14           | 0.46           | 10^{-3}        | –              |
| A_VineE1/A_GadgetE1 | 100k     | 1.0            | 1.0      | \( 10^{-5} \) | 0.02   | 1              | 2.14           | 0.46           | 10^{-3}        | 0.1            |
| A_VineE2/A_GadgetE2 | 100k     | 1.0            | 1.0      | \( 10^{-5} \) | 0.02   | 1              | 2.14           | 0.46           | 10^{-3}        | 0.02           |
| A_rVine    | 100k     | 1.0            | 1.0      | \( 10^{-5} \) | 0.02   | 1              | 2.14           | 0.46           | 10^{-3}        | –              |
| B_Nbody7   | 10k–100k | 1.0            | 1.0      | \( 10^{-4} \sim 10^{-5} \) | –      | 2              | 0.10           | 0.28/0         | 5 \times 10^{-3} | –              |
| B_Vine     | 10k–1M   | 1.0            | 1.0      | \( 10^{-4} \sim 10^{-6} \) | 0.01   | 2              | 0.10           | 0.28/0         | 5 \times 10^{-3} | 0.01           |
| B_rVine    | 10k–1M   | 1.0            | 1.0      | \( 10^{-4} \sim 10^{-6} \) | 0.01   | 2              | 0.10           | 0.28/0         | 5 \times 10^{-3} | 0.01           |

4 All quantities are given in code units.

4 Note that we plan to employ much higher particle numbers in rVine, well above what we have used here for the comparison tests, in future (see Section 6).
Figure 3. Projections of the initial stellar surface densities (in code units) for a realization of model A_rVine_100k. The orbital evolution of an SMBH, placed on a circular orbit at the half-mass radius, is shown as the black solid line and symbols for the first orbital period.

Table 2. Parameters for the rVINE calculations.

| Simulation  | $\alpha$ | $\beta$ | $\gamma$ | $\gamma_{\text{crit}}$ |
|-------------|----------|---------|----------|---------------------|
| A_rVine     | 1.0      | 1.0     | 1.5      | $10^{-4}$           |
| B_rVine     | 0.1–0.3  | 1.0–4.0 | 1.5      | $10^{-4}/10^{-5}$   |

*aNote that, throughout the text, added suffixes will state the actual particle numbers of the simulations.*

Figure 4. Time evolution of the number of members in the chain and the number of perturbers in a realization of model A_rVine_100k, with one SMBH initially set on a circular orbit (see Section 5.1). Average numbers are given as text and indicated by the dashed lines.

Overall, a fraction of 6.9 per cent of the $10^5$ particles, was subject to integration in the chain during this run. Chain integration was active for a total of 42 per cent of the total run time, $t_{\text{max}} = 800$. At the early stages of the simulation, the chain is used only intermittently to treat the occasional strong stellar encounters near the SMBH in the low-density environment. At later phases, when the SMBH is near to the centre, the chain is used more intensively, with the chain being active continuously for $\sim 285$ time units. Stellar particles are typically included repeatedly in the chain with an average number of $\sim 8$ recurrences.

Fig. 5 shows the corresponding time evolution of some characteristic radii from the simulation shown in Fig. 4. The escape radius (black dashed line), given by equation (5), serves as an effective upper bound for the chain radius (red solid line). For times $t \lesssim 300$ the minimum disturber distance, $r_{\text{p, min}}$ (green solid line) is generally well above the escape radius, before some perturbers may come closer to a chain that has only a small, but increasing, number of members while the SMBH is sinking to denser central regions ($300 \lesssim t \lesssim 520$). Once the SMBH is close to the centre of the Hernquist sphere ($t \gtrsim 550$) and the number of particles in the chain has increased by a factor of 10, $r_{\text{chain}}$ and $r_{\text{p, min}}$ both oscillate around $r_{\text{escape}}$, which naturally arises when a number of particles is situated close to the conditions for both absorption and escape being satisfied at a certain time. In this case, in rVINE we prioritize the absorption of near perturbers over a (delayed) escape of a chain member, until the chain radius has grown by 5 per cent over the nominal escape radius. A noticeable oscillation around the escape radius can then occur if a series of subsequent absorptions of perturbers near the SMBH takes place.
Figure 5. Time evolution of characteristic radii in a realization of model A_rVine_100k, with one SMBH set on a circular orbit (see Section 5.1): the radius of the chain (red solid line), the distance of the nearest perturber to the chain centre of mass (green solid line) and the radius of escaping chain particles (black dashed line).

In Figs 6 and 7, we investigate the performance characteristics of rvine using a set of simulations of our basic Hernquist model with two SMBHs, each having a mass of \( M_{\text{BH}} = 5 \times 10^{-3} M_{\text{tot}} \). One of the SMBHs is initially set on a circular orbit close to the centre (\( x_{\text{BH1}} = \pm 0.1 \) and \( v_{\text{y, BH1}} \approx 0.28 \)), the other one is at rest at the origin (models B_Vine and B_rVine; see also Section 5.2). All runs shown in Figs 6 and 7 were performed on a single node (eight CPUs) of Cosmos2, except for a few comparison runs using NBODY7 (see Fig. 6, upper panel) which were performed using eight CPUs plus acceleration from two NVIDIA K20 GPUs on the Wilkes cluster.

The three panels in Fig. 6 show the wall clock time required to evolve the simulation to \( t = 1 \) (upper panel), as well as the average number of particles in the chain (middle panel) and the average number of perturber particles (bottom panel) as a function of the initial chain radius \( r_{\text{chain,0}} \) and particle number \( N \). Interestingly, the run time does not seem to depend strongly on the choice of the initial chain radius; it typically changes by a factor of \( \sim \) a few, and at most by a factor of \( \sim 8 \), when varying \( r_{\text{chain,0}} \) up to a factor of 3. In addition, the scaling with \( N \) is still relatively shallow for all initial chain radii as the computing time is dominated by the tree. For comparison, we show the scaling obtained with NBODY7 using the same initial conditions for \( N \leq 2 \times 10^5 \) (black dashed line). Due to the fact that we can use the additional acceleration of the two NVIDIA K20 GPUs (plus some contribution from running on a different system) the total computing time is actually a factor of \( \sim 5 \) to \( \sim 10 \) times lower compared to rvine at the lowest particle numbers. However, the steeper scaling of NBODY7 \( (T_{\text{wall}} \propto N^{1.9}) \) yields comparable computing times already for \( N \gtrsim 2 \times 10^5 \). Extrapolating this scaling would give clear advantages to rvine for \( N \gtrsim 10^6 \) particles in that particular case.

The average number of particles in the chain shows a roughly linear scaling with total particle number for \( N > 2 \times 10^4 \) while the average number of perturbers has a shallower \( N \)-dependence. The latter can be understood by considering the fact that the region of the perturber particles actually becomes smaller with increasing \( N \), since the mass of the perturbers scales as \( \propto N^{-1} \) (see equation 2).
of $\tau_* \leq 0.3$, the scaling is much steeper for larger values of $r_{\text{chain,0}}$, i.e. when a larger fraction of tree code particles are integrated in the more accurate chain. For the simulations with $r_{\text{chain},0} = 0.02$ and $r_{\text{chain},0} = 0.03$ the relative energy error scales roughly as $\propto \tau_*^2$.

5 COMPARISON WITH ANALYTICAL ESTIMATES AND OTHER CODES

5.1 Dynamical friction of an SMBH in a Hernquist sphere

As a first test of VINE, we investigate the orbital evolution of a massive particle due to dynamical friction in a spherical non-rotating Hernquist sphere (model A), comparing results from different simulations using NBODY7, VINE, VINE and GADGET-3. The massive particle of mass $M_{\text{BH}} = 10^{-3} M_\odot$ is set on an initially circular, corotating orbit at the half-mass radius ($r_{\text{BH}}(t = 0) = 2.41$) (see also Section 3).

5.1.1 Dynamical friction theory

For a meaningful assessment of the different codes, we compare the simulated SMBH trajectories with theoretical expectations from dynamical friction theory (Chandrasekhar 1943; Binney & Tremaine 2008). Assuming a locally isotropic velocity distribution function, a homogeneous stellar density $\rho$, as well as a sufficiently large velocity of the SMBH, $v_{\text{BH}}$, relative to the stellar background, the deceleration of the SMBH due to dynamical friction may be given by the standard formula (cf. Binney & Tremaine 2008, equation 8.5)

$$\dot{a}_{\text{DF}} = -4\pi G^2 \frac{M_{\text{BH}}}{v_{\text{BH}}^3} \rho(v < v_{\text{BH}}) \ln \Lambda \, v_{\text{BH}},$$

where $\ln \Lambda$ is the Coulomb logarithm. Only the mass density of stars moving slower than the SMBH, $\rho(v < v_{\text{BH}})$, contributes to the dynamical friction. With gravitational softening there is a limit on the maximum gravitational force that may be exerted in any star-SMBH encounter through an effective minimum impact parameter $b_{\text{min}}$ (see e.g. White 1976; Just et al. 2011),

$$b_{\text{min}} = 1.5 \cdot \max(\epsilon_*, \epsilon_{\text{BH}}).$$

Taking this softening effect into account, we write the Coulomb logarithm as

$$\ln \Lambda = \ln \left( \frac{b_{\text{max}}}{\sqrt{b_{\text{max}}^2 + b_{\text{00}}}^2} \right),$$

where $b_{\text{max}}$ is the maximum impact parameter and $b_{\text{00}}$ is the impact parameter for a 90° scattering event of the incident star. The choice for the latter two parameters is often rather arbitrary, with $b_{\text{00}}$ depending on the typical velocity $v_{\text{typ}}$ of the stars. The maximum impact parameter $b_{\text{max}}$ is often taken proportional to the orbital radius of the SMBH. Following Just & Peñarrubia (2005), we identify $b_{\text{max}}$ with the local scalelength,

$$b_{\text{max}} = \frac{\rho}{|\nabla \rho|} = \frac{r_{\text{BH}}}{3 - \eta}, \quad \eta \leq 2,$$

where the last equation is true for the family of $\eta$-models (Dehnen 1993; Tremaine et al. 1994) if $\eta \leq 2$. Hence, in the case of the Hernquist profile ($\eta = 2$) we obtain that the maximum

Figure 7. Performance characteristics for the VINE code as a function of varying initial chain radii for the model used in Fig. 6. Shown are the code run time (top panel) and the relative energy error (bottom panel) versus the accuracy parameter of the tree code time-step criteria, $\tau_*$. All runs were evolved for one code time unit ($\Delta t = 1$).

Hence, if the number of particles in the chain were to scale strictly linearly with $N$, or in the limit of the SMBH dominating the total mass of the chain, e.g. for a high SMBH-to-star mass ratio and small $r_{\text{chain,0}}$, we would expect $\langle N_{\text{pert}} \rangle$ to be largely independent of $N$. The observed shallow increase of $\langle N_{\text{pert}} \rangle$ with $N$ is likely to be caused by the non-trivial non-linear scaling of $\langle N_{\text{chain}} \rangle$ observed in the middle panel. Both $\langle N_{\text{chain}} \rangle$ and $\langle N_{\text{pert}} \rangle$ show a scaling going roughly as $N \propto r_{\text{chain,0}}^2$, as expected for the central parts of the Hernquist profile, where $M(r) \propto r^2$ (see Hernquist 1990, their equation 3).

The upper panel of Fig. 7 shows the total run time of the simulations as a function of the accuracy parameter for the tree code time-step criteria, $\tau_*$. In principle, one is free to choose different values for the different time-step criteria used in VINE (see Section 3 and equations 10–12 in Wetzstein et al. 2009), but we decided to adopt one, identical value of $\tau_*$ for all criteria and fixed the particle number at $N = 10^5$. The time-step size increases linearly with $r_*$ leading to an overall decrease in the run time for all values of $r_{\text{chain,0}}$, albeit with some non-negligible scatter. Within the scatter, the total wall time scales roughly as $T_{\text{wall}} \propto r_{\text{chain,0}} \epsilon_*$ for fixed $\epsilon_*$. In the lower panel of Fig. 7, we show the energy conservation of the code for a given time-step accuracy. To avoid spurious energy errors upon start-up, we have measured the energy errors in the interval from $t = 4$ to $t = 5$. The calculations become generally less accurate for larger time-step sizes as expected. However, especially for values

MNRAS 452, 2337–2352 (2015)
impact parameter exactly equals the orbital radius of the SMBH, i.e. \( b_{\text{max}} = 2r_H \). In addition the 90\(^\circ\) deflection parameter is given by

\[
b_{90} = \frac{GM_{BH}}{v_{Tp}} \approx \frac{GM_{BH}}{2v^2 + v_{BH}^2}.
\]  

(10)

Using equations (6)–(10), we evolve the orbit of an SMBH, initially placed on a circular orbit at the half-mass radius, in the analytic Hernquist potential including the additional drag forces due to dynamical friction with a leapfrog integrator. The density of stars with velocities smaller than the SMBH is represented as

\[
\rho(v < v_{\text{BH}}) = \kappa \cdot \rho_s(v < v_{\text{BH}}),
\]

(11)

where \( \rho_s(v < v_{\text{BH}}) \) denotes a locally isotropic Maxwellian velocity distribution given by

\[
\rho_s(v < v_{\text{BH}}) = \rho(r) \times \left[ \text{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right],
\]

(12)

where \( X = v_{\text{BH}}/\sqrt{2} \sigma \) with dispersion \( \sigma \), and \( \text{erf}(X) \) is the error function (Binney & Tremaine 2008). We use \( \kappa \) as a free parameter to account for the fact that the velocity distribution of the Hernquist model does not follow a simple Maxwellian distribution, as often used as an approximation in dynamical friction calculations. Hence, the locally isotropic Maxwellian velocity distribution corresponds to \( \kappa = 1.0 \).

5.1.2 Results

The upper panels of Fig. 8 show the evolution of the radial decay (left) and the velocity (right) of the SMBH with time for model A_rVine_100k. _NBODY 7_ is very well suited to follow the dynamical friction of the heavy body. We gauge run A_rVine_100k (blue line) against theoretical orbits for different values of \( \kappa = (0.5, 0.75, 1.0) \) shown as black lines and use it subsequently as a reference for the other simulation codes. Initially, the orbital evolution shows quite closely the analytic prediction with \( \kappa = 0.75 \) (black dashed line). At smaller central distances \( (r < 0.6r_{\text{infl}}) \) the dynamical friction seems to act even more efficiently in the full N-body run. This leads to a slightly faster orbital evolution such that the SMBH reaches the centre (defined as \( r < r(M < 2M_{\text{BH}}) \), where dynamical friction ceases to be efficient) within \( t < 500 \) time units.

In the second row of Fig. 8, we compare the efficiency of dynamical friction on the SMBH in model A_rVine_100k (red line) and model A_rVine_100k (blue line). We find that the early orbital evolution \( (r > 0.5r_{\text{infl}}) \) in model A_rVine_100k is nearly indistinguishable from the one in model A_rVine_100k such that the SMBH reaches the centre on a very similar time-scale, only about \( \sim 8 \) per cent longer than in model A_rVine_100k, in very good agreement with the _NBODY 7_ result.

On the other hand, owing to their inherent resolution limitations, we expect the tree codes to show a significantly slower radial decay depending on the adopted gravitational softening, via equation (7). If the SMBH is treated as a quasi collisionless particle with a large gravitational softening length \( \epsilon_{\text{BH}} = 0.1 \), third row), it decays to only about half its original distance within the time span \( (t \sim 960) \) shown in Fig. 8 in both model A_Vine_E1_100k (green line) and model A_Vine-E1_100k (pale blue line). The total time to reach the centre is \( t > 1350 \) time units for both codes, significantly longer (by a factor of \( \geq 2.7 \)) than with _NBODY 7_. This can be understood as a result of the reduced frictional force for near encounters. Both codes, however, follow the analytical prediction with \( \kappa_s = 0.75 \) if we adopt the minimal impact parameter \( b_{\text{min}} = 1.5 \cdot \epsilon_{\text{BH}} \) from equation (7). Setting \( \epsilon_{\text{BH}} = \epsilon_s = 0.02 \) (bottom row) yields an enhancement in the near encounters, but the decay time to the centre is still about 50 per cent (70 per cent) longer than in the _NBODY 7_ case for model A_Vine-E1_100k (A_Gadget-E1_100k). Hence, even this drastic reduction in the gravitational softening of the SMBH does not significantly improve the accuracy of the dynamical friction time-scale.

While all codes capture the effects of dynamical friction as expected, if we consider softened gravitational force terms, this highlights the importance of an accurate treatment of close encounters for the correct description of dynamical friction in N-body tree codes. From our analysis, we conclude that the AR-regularized tree code removes the limitations due to the gravitational softening of the SMBH imposed on present-day tree codes by including nearby particles in the chain.

5.2 Evolution of an SMBH binary at the centre of a Hernquist sphere

Another crucial test for _VINE_ is to investigate the hardening of an SMBH binary, compared to results obtained with _NBODY 7_ and _VINE_. For our model B, we choose an initial set-up similar to the one presented in Merritt et al. (2007) and Vasiliev, Antonini & Merritt (2014). In particular, we set up two SMBHs at the centre of a non-rotating Hernquist sphere with masses \( M_{\text{BH}} = 5 \times 10^{-3} M_{\text{tot}} \) and one of the SMBHs on a circular orbit \( (v_{\text{BH}} = 0.1, v_{\text{infl}} = \sim 0.28) \) and the other SMBH at rest at the origin. For models B_Vine and B_rVine, we use gravitational softening lengths \( \epsilon_s = 0.01 = \epsilon_{\text{BH}} \) for the stellar and SMBH particles in the tree code. Particles in the chain are not softened. The initial distance is chosen slightly larger than the influence radii of the SMBHs, which typically increase in the B_rVine and B_Nbody7 runs from \( r_{\text{infl}} \sim 0.1 \) to \( r_{\text{infl}} \sim 0.16 \) according to a decrease in the central density during the simulations. In the simulations using model B_rVine, initially, only the corotating SMBH is regularized, but the second SMBH is captured into the chain quickly, for \( t < 1.5 \) in all runs. In the B_rVine runs, we use an AR-chain with a chain radius \( r_{\text{chain}} = 1.25 \times 10^{-3} \), while we test different values of the initial chain radius for the AR-chain in _VINE_ which we set to a size comparable to the hard binary semimajor axis, \( a_{\text{chain}} \sim \text{a few} \times a_{\text{hard}} \).

The hard binary semimajor axis is given as

\[
a_{\text{hard}} = \frac{1}{M_{\text{BH1}} + M_{\text{BH2}}} \frac{r_{\text{infl}}^{2M}}{4},
\]

(13)

with \( M_{\text{BH}} = (1/M_{\text{BH1}} + 1/M_{\text{BH2}})^{-1} \) being the reduced mass. Initially, \( a_{\text{hard}} = r_{\text{infl}}^{2M}/16 \approx 6.25 \times 10^{-3} \).

In Fig. 9, we show the time evolution of the SMBH binary hardness \( (1/a, \text{left-hand panels}) \) and eccentricity (right-hand panels) for the first \( t = 100 \) time units in the simulations of model B with \( N = 10^5 \) particles. If we set a large initial chain radius \( r_{\text{chain}, 0} = 0.02 \) (red solid lines), \( r_{\text{chain}, 0} = 0.03 \) (red dashed lines) or \( r_{\text{chain}, 0} = 0.04 \) (red dot–dashed lines), model B_rVine_100k agrees well with model B_rVine_100k (blue lines) showing an efficient hardening with a nearly constant hardening rate \( \frac{d}{dt} (1/a) \) for \( t > 10 \), and a rather mild

\footnote{Note that in Vasiliev et al. (2014) both SMBHs are set on corotating orbits, while here we set one of the SMBHs at rest at the origin. We tested that this different set-up gives similar results. Furthermore, in their simulations, the initial velocities are set to \( |v| = 0.31 \), which is about \( \sim 10 \) per cent higher than the circular velocity.}

\footnote{Note that employing equally large chain radii would not be feasible in _NBODY 7_ without major modifications to the code.}
Figure 8. Comparison of the distance to the centre (left-hand panels) and velocity (right-hand panels) of a single SMBH, initially set on a circular orbit at the half-mass radius of a Hernquist sphere, for different code architectures. From top to bottom, we show the direct summation code \textsc{nbody7} (blue; model A\_Nbody7\_100k), the AR-chain regularized tree code \textsc{rVine} (red; model A\_rVine\_100k) and the \textsc{VINE} and \textsc{GADGET-3} tree codes (green and pale blue lines) with varying softening lengths, $\epsilon_{BH} = 0.1$ (models A\_Vine-E1\_100k and A\_Gadget-E1\_100k) and $\epsilon_{BH} = 0.02 = \epsilon_\ast$ (models A\_Vine-E2\_100k and A\_Gadget-E2\_100k). Theoretical expectations from dynamical friction theory are given as black lines for different parameters of $\kappa_s$ ($\kappa_s$ parametrizes the deviation from a locally isotropic Maxwellian velocity distribution, see Section 5.1). For comparison, we show the results for the \textsc{nbody7} run also in the other panels. \textsc{rVine} very well recovers the expected orbital evolution by effectively removing the limitations imposed by gravitational softening. Results shown are averages over several realizations of the initial conditions, as described in the text.
The evolution of binary parameters of an SMBH binary at the centre of a non-rotating Hernquist sphere (model B). Shown are binary hardness ($1/a$) and eccentricity ($e$) as a function of time. Different colours indicate simulations using different codes, i.e. $\text{nbody7}$ (blue), $\text{vine}$ (green), and $\text{rVine}$ (red lines). Different line styles for model $\text{B}_{rVine}$ denote different runs with varying initial chain radii ($r_{\text{chain}, 0} = 0.01$: dotted, 0.02: solid, 0.03: dashed, and, 0.04: dot-dashed). The results shown are averages over several realizations of the initial conditions, as described in the text.

For $\text{vine}$ and $\text{rVine}$, we investigate simulations with particle numbers ranging from $N = 2 \times 10^3$ to $10^9$ and between $N = 2 \times 10^5$ and $10^7$ for $\text{nbody7}$. In the initial stages of the simulation ($t \lesssim 5$), when the loss-cone is full and dynamical friction is still efficient, the binary parameters evolve qualitatively similar in all three codes, with a steep rise in $1/a$ and a quite broad range of moderate eccentricities $0.1 \lesssim e \lesssim 0.7$. However, thereafter models $\text{B}_{\text{Vine}}$ (bottom row) quickly evolve to high eccentricities while the binary semimajor axis typically stalls at $a \lesssim 0.5 a_{\text{hard}}$. In models $\text{B}_{\text{nbody7}}$ (top row) and $\text{B}_{r\text{Vine}}$ (middle row), on the other hand, a hard binary with $a \leq a_{\text{hard}}$ forms quickly within $t \approx 2.5$–5. Models $\text{B}_{\text{nbody7}}$ and $\text{B}_{r\text{Vine}}$ again show an overall much more efficient binary hardening and eccentricity evolution than in $\text{B}_{\text{Vine}}$. The binary semimajor axis and eccentricity reach final values of $1/a \lesssim (1100–2600)$ and $e \lesssim (0.2–0.7)$ in the $\text{B}_{\text{nbody7}}$ and $\text{B}_{r\text{Vine}}$ runs, respectively, depending on $N$. Both in models $\text{B}_{\text{nbody7}}$ and $\text{B}_{r\text{Vine}}$, the binary hardening decreases with increasing $N$. Since this is due to the lower efficiency of collisional loss-cone refilling for larger $N$, however, the decrease is more pronounced in model $\text{B}_{\text{nbody7}}$. As expected, in model $\text{B}_{\text{Vine}}$ we again have a very weak evolution of the binary hardening ($1/a \lesssim 400$) but high final eccentricities ($0.8 \lesssim e \lesssim 0.95$) with a very weak dependence on $N$.

In Fig. 11, we show the effect the ejection of low-angular-momentum stars at the bottom of the potential has on the structure of the galactic nucleus. Shown are radial profiles for the density (upper panel), the radial velocity dispersion (middle panel) and the velocity dispersion anisotropy parameter, $\beta_r = 1 - \frac{\sigma_{\phi}^2 + \sigma_{\theta}^2}{2 \sigma_r^2}$ (bottom panel), for the models shown in Fig. 9 at the end of the simulations. Stars are ejected from the galaxy centre by the hardening binary on orbits with high radial velocities in the $\text{B}_{r\text{Vine}}$ (red) and $\text{B}_{\text{nbody7}}$ (blue) runs. This leads to a large increase in the radial velocity dispersion at galactocentric radii with $r \geq 10 r_0$ for these simulations (upper panel), together with a depression in the central density profile within $r \leq r_{\text{tail}}$ and some added mass in the outskirts of the galaxy (middle panel). The central density profile is converted from an initial Hernquist profile with inner slope of $r^{-1}$ (dashed line) to a profile with $r^{-0.5}$ in the $\text{B}_{r\text{Vine}}$ and $\text{B}_{\text{nbody7}}$ runs. The increase in the radial velocity dispersion is not seen in the tangential velocity dispersions such that the anisotropy profile is strongly radially biased for $r \gtrsim 10 r_0$. We verified that this is caused only by particles escaping the system after being ejected in interactions with the central binary in both codes. For $\text{B}_{\text{Vine}}$, the radial velocity dispersion is on average to be more effective in removing mass from the centre than $\text{nbody7}$. Given that the SMBH binary hardens by roughly the same amount in both codes, this might be due to the fact that the central density is replenished more efficiently by the high rate of collisional loss-cone refilling with stars originating from larger radii in the $\text{B}_{\text{nbody7}}$ runs.

We further analyse the properties of the high radial velocity stars in the realization using model $\text{B}_{r\text{Vine}}$ shown in Fig. 11 by...
Figure 10. Time evolution of binary hardness ($1/a$, left-hand panels) and eccentricity (right-hand panels) as a function of particle number $N$ for the three different codes in model B: NBODY7 (top row), rVINE (middle row) and VINE (bottom row). Shown are simulations with particle numbers increasing from $N = 20k$ to $N = 1M$ from top to bottom, with colours indicated in the legend. The initial chain radius is set to $r_{\text{chain},0} = 0.02$ in the simulations of model B_rVine. The results shown are averages over several realizations of the initial conditions, as described in the text.
examining the radial velocities of stars escaping the chain in Fig. 12. Not all of the particles interacting with the SMBH binary leave the chain on high radial velocity orbits: the majority of the escapers has $v_{\text{rad}} < 1$ at all times in the simulation. However, for $t \lesssim 10$ there is an enhanced interaction rate of stars in the initially full loss-cone with the hardening SMBH binary leading to a significant population of escapers with radial velocities $1.5 < v_{\text{rad}} < 2.5$, comparable to the expected kick velocities in a slingshot interaction of a field star with a massive binary with hardness $1/a \sim 1000$. Furthermore, we find a small population of high-velocity outliers ($2 < v_{\text{rad}} < 6.5$) – much higher than the expected kick velocity – which provide a clear observational signpost of the hard SMBH binary at the centre.

6 DISCUSSION

Due to the favourable $O(N \log(N))$ scaling of the tree code we should be able to employ (much) higher particle numbers than in the test calculations presented in this paper in future applications of the new code. However, as a caveat, we note that the time spent for the AR-chain calculations scales steeper with particle number than the time spent for the tree calculations, mostly due to the
costly extrapolations of the Bulirsch–Stoer method and the repeated predictions and force evaluations of the perturbers. For the simulations performed for this paper, execution of the chain part of the code has only taken a moderate fraction of the total CPU time (typically below 5 per cent with a maximum of ∼20 per cent), and with some optimization it should be possible to further increase the speed of the code. There will, however, be some critical particle number, \( N_{\text{crit}} \), at which the AR-chain will become the dominant contributor to the total computing costs even if only a small fraction of the total particle number is actually integrated in the chain. It is not within the scope of this paper to investigate this in detail and we leave this to future work where we will use our new hybrid code for full-scale galaxy simulations.

Throughout Section 5.2, we have found a qualitatively similar evolution of the hardening binary both in \texttt{NBody7} and \texttt{rVine} for chain sizes of \( \sim \) a few times the hard binary distance \( d_{\text{hard}} \) (equation 13). The agreement is particularly good for the highest particle numbers studied here (\( N \gtrsim 80k \)) where spurious relaxation effects become less important in the direct-\( N \)-body code. Similarly, for lower particle numbers (\( N \lesssim 50k \)), \texttt{rVine} shows a shallower \( N \)-dependence of the hardening rate since the tree code better reproduces the collisionless galactic stellar dynamics at distances far from the SMBHs. Hence, the hybrid code seems to catch the relevant dynamical interactions of a real galaxy better at lower \( N \) for our set of parameters adopted for the chain.

However, we also note here that the high hardening rates we find in Section 5.2 in \texttt{NBody7} simulations are somewhat larger than those found in a number of recent studies of spherical and axisymmetric models using similar techniques and initial conditions (see e.g. Khan et al. 2013; Vasiliev et al. 2014). In particular, the final binary hardness for our B_\text{NB} model is on average about a factor of \( \sim 2.5 \) higher for comparable particle numbers than the spherical models studied in Vasiliev et al. (2014). Several reasons could be responsible for this difference including slightly different choices in the initial positions and velocities of the SMBHs (see Section 5.2), differences in the integration techniques (parameters for the AR-chain, settings for the gravitational softening and the integration accuracy, etc.), or some slight inaccuracies in the scaling of the \( N \)-body systems.

In Section 5.1, we have found that in the tree codes \texttt{Vine} and \texttt{Gadget-3} – even with the most conservative choice of the SMBH softening length – the dynamical friction time-scales for the SMBH to sink to the galactic centre differ by more than 50 per cent from the ones in \texttt{NBody7} and \texttt{rVine}. Hence are present-day cosmological and galaxy merger simulations suffering from significant (unavoidable) uncertainties in the SMBH orbital time-scales? Strictly speaking they do, but probably, in most cases SMBHs are rarely found orbiting ‘naked’ in their host galaxies, but are instead embedded in stellar and gaseous cores or cusps that are then the prime subjects to dynamical friction in galaxy interactions. However, it has been shown that ‘naked’ BHs might be quite commonly formed after the disruption of galactic nuclei in gas-rich minor mergers (Callegari et al. 2009; Van Wassenhove et al. 2014), making accurate estimates of the dynamical friction time-scales of the ‘naked’ BHs necessary in these cases (see also Tremmel et al. 2015).

Of similar importance for the galaxy formation community is to get better estimates for SMBH binary coalescence time-scales in order to make robust predictions with respect to the dynamical evolution of SMBHs in their host galaxies. For example, studying the exciting possible formation of systems with multiple SMBHs at high redshift due to the high merger rate of galaxies relies crucially on (1) an accurate description of dynamical friction in order to correctly quantify the populations of binary, triple, etc. SMBHs being present at a given time, and (2) accurate orbits in order to reliably calculate the final outcome of the strong multibody interactions between the SMBHs in the galactic centres (see e.g. Blecha et al. 2011; Kulkarni & Loeb 2012). Obtaining accurate coalescence time-scales for binary SMBHs in gas-rich galaxy mergers is also essential for accurate estimates of the likelihood of recoiling SMBHs escaping from the rapidly steepening central potential of the merger remnants (e.g. Sijacki, Springel & Haehnelt 2011). This should be particularly relevant for large-scale cosmological simulations like, e.g. the recent EAGLE and Illustris (Vogelsberger et al. 2014; Schaye et al. 2015) simulations that assume fast coalescence of two SMBHs once their distance falls below the resolution limit (see e.g. Sijacki et al. 2015, for details).

7 CONCLUSIONS

In this paper, we have presented a hybrid code combining an OpenMP-parallel binary tree code (\texttt{Vine}) with an algorithmic chain regularization scheme, and report on first tests with the new code called \texttt{rVine}.\footnote{\texttt{rVine} is available to anyone interested upon request to the authors.}

We have shown that, using the AR-regularized tree code, we can significantly improve the numerical accuracy in the calculation of the gravitational interactions of SMBHs with their close environment. By comparison with the collisionless code \texttt{NBody7}, we have verified that we have overcome some of the fundamental limitations imposed by the gravitational softening of the SMBHs, as it is used in traditional tree codes. As a consequence, we are now able to follow the orbital evolution of SMBHs much more accurately in more realistic, galaxy-scale settings. We have shown that with the new hybrid code we obtain both significantly improved estimates for dynamical friction time-scales of single SMBHs sinking to the galactic centres and for the time evolution of hard SMBH binaries. In particular, using \texttt{rVine}, we find a clear \( N \)-dependence of the binary hardening rate, a low binary eccentricity along with a moderate eccentricity evolution, as well as the conversion of the galaxy’s inner density profile from a cusp to a core via the ejection of low-angular-momentum stars on orbits with high radial velocity, similar to the results obtained with \texttt{NBody7} here and in previous work.

Due to the modular design of \texttt{rVine}, the AR-chain part with its hybrid interface should be easily portable to other codes used for simulations of galaxy formation. It will likewise be straightforward to incorporate additional physics into \texttt{rVine}, e.g. formulations for hydrodynamics and additional subgrid models of (1) star formation and stellar feedback, and (2) black hole accretion and feedback, or, the addition of post-Newtonian terms in the AR-chain. Note also that in the present paper we have restricted ourselves to the case of regularizing the dynamics of one single subsystem only. The next step here is to extend the present code to allow for multiple chains in order to handle the regularization of several distant SMBHs at once.

Important problems that will benefit from the accurate dynamical modelling of the evolution of (binary) SMBHs are predictions with regard to SMBH coalescence rates and their associated gravitational wave background (e.g. Haehnelt 1994), the population of SMBHs and AGNs (either living at their host galaxies centres or being displaced from the central regions by three-body encounters or gravitational wave recoils), the acceleration of hyper-velocity stars...
This work was supported by the Science & Technology Facilities Council (STFC; grant number ST/J001538/1) and the STFC DiRAC project dp021. We would very much like to thank Seppo Mikkola for providing the AR-chain routines. SJK is grateful to the hospitality shown by the Max-Planck-Institut für Astrophysik in Garching and would like to thank E. Vasiliev for helpful comments on the manuscript. TN acknowledges support by the DFG cluster of excellence ‘Origin and Structure of the Universe’. We acknowledge support by Chinese Academy of Sciences through the Silk Road Project at NAOC (RS, SJA and SJK), through the Chinese Academy of Sciences Visiting Professorship for Senior International Scientists, Grant Number 2009S1 – 5 (RS), and through the ‘Qianren’ special foreign experts program of China. The main numerical part of this work was undertaken using the DiRAC Shared Memory Processing system at the University of Cambridge, operated by the COSMOS Project at the Department of Applied Mathematics and Theoretical Physics on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). This equipment was funded by BIS National E-infrastructure capital grant ST/J005673/1, STFC capital grant ST/H008586/1, and STFC DiRAC Operations grant ST/K00333X/1. DiRAC is part of the National E-infrastructure. The NBODY7 part of this work was realized on the Wilkes GPU cluster at the HPC Service of the University of Cambridge.
