Mem. S.A.It. Suppl. Vol. 1, 54

Memorie della
Supplementi

Tools and Techniques for N-body Simulations

P.F. Spinnato[1], S.F. Portegies Zwart[1,2], M. Fellhauer[3], G.D. van Albada[1], and P.M.A. Sloot[1]

[1] Section Computational Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands e-mail: {piero | dick | sloot}@science.uva.nl

[2] Astronomical Institute 'Anton Pannekoek', Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands e-mail: spz@astro.uva.nl

[3] Institut für Theoretische Physik und Astrophysik der Christian-Albrechts-Universität zu Kiel, Leibnizstr. 15, D-24118 Kiel, Germany e-mail: mike@astrophysik.uni-kiel.de

Abstract. Computational Astrophysics research often requires very powerful machines, and very sophisticated codes. In stellar dynamics simulations, the development of the treecode, and the realisation of the GRAPE special purpose device, have contributed dramatically to the progress of scientific research. The pseudo-particle approach, where a multipole expansion is expressed in terms of a particle distribution, allows to improve the tree-code accuracy at a limited computational cost. Moreover, it is suitable for making full use of the GRAPE. We study the error behaviour of this approach with respect to changing physical distributions. Thus we introduce improvements that reduce the errors. Furthermore we present an extension of the pseudo-particle scheme, where pseudo-particles are not fixed in space, but move following the physical particle distribution. This extension decreases the computational overhead due to pseudo-particle recomputation, and optimises the scheme for the use on GRAPE and on parallel systems.
We also study the efficiency at which a black hole spirals-in to the Galactic centre, using the direct N-body method running on the GRAPE, a tree algorithm and a particle-mesh technique. The three different techniques are in excellent agreement. The black hole infall, within the set of parameters that we consider, is not influenced by the granularity of the N-body system.

Key words. methods: N-body simulations – methods: numerical – celestial mechanics, stellar dynamics – black hole physics – Galaxy: bulge.

Send offprint requests to: P.F. Spinnato
Correspondence to: Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

1. Introduction

A relevant fraction of the Computational Astrophysics community is involved in research related with the solution of the gravitational N-body problem. In many cases, simulating an N-body system requires very sophisticated algorithms, and very high computing power. The development of the treecode (Barnes & Hut, 1986), which dramatically decreases the computational cost of the gravity force computation, constitutes a seminal progress in this field. Another milestone is the realisation of GRAPE (Makino et al., 1997), a special purpose device which performs gravity force computations at ultra high speed. The use of parallel and distributed systems has also contributed substantially to speed up N-body simulations. The development of a computational environment which efficiently integrates all the above components, would be a further remarkable improvement. In view of this goal, our research in Computational Astrophysics is dedicated to the interaction of the GRAPE with a parallel host, and the interaction of fast software applications for the N-body problem with such hybrid architecture. A substantial component of such research is the development of modified treecodes, and hybrid codes. The former are able to make full use of GRAPE by means of multipole expansions in terms of pseudo-particles (Makino, 1999). Within this framework, we present here an error analysis study of our pseudo-particle treecode. The latter are suited to perform the accurate simulation of star clusters orbiting galaxy nuclei. In this direction, we have studied the infall of an intermediate mass black hole to the Galactic centre using the direct N-body method, a tree algorithm and a particle-mesh technique with up to 2 million particles. A quantitative analysis of the results are presented in Spinnato et al. (2002).

2. The treecode and the pseudo-particle expansion

The Barnes and Hut treecode (Barnes & Hut, 1986) is one of the most popular numerical methods for particle simulation involving long range interactions. Its superior $\mathcal{O}(N \log N)$ scaling, compared with the $\mathcal{O}(N^2)$ scaling of the direct code, is obtained at the cost of a reduced accuracy. The treecode groups particles together according to a hierarchical tree, where each node of the tree is associated to a cubic cell in three-dimensional space, and a cell corresponding to a node κ consists of the eight cells corresponding to the eight child nodes of κ. The treecode attains its $\mathcal{O}(N \log N)$ scaling by computing force on a particle i from larger and larger cells as their distance from i gets bigger and bigger. In order to decide whether a cell is far enough to be accepted for such force computation, a certain Multipole Acceptability Criterion (MAC) must be provided. See Section 3 below for the MAC used in this work.

The force contribution is evaluated from a multipole expansion of the particle distribution contained into the cell. The accuracy of such evaluation depends on the highest term in the multipole expansion. Usually, expansions are truncated at the quadrupole term, leading to an accuracy in the force computed at a given moment in time in the order of 1% (Salmon & Warren, 1994), as compared to the direct $\mathcal{O}(N^2)$ algorithm. This makes treecodes unsuitable for many applications that require high numerical precision.

Research has been carried out in order to improve the treecode accuracy, by increasing the maximal multipole expansion order (McMillan & Aarseth, 1993). We aim at developing a version of the treecode that allows tunable accuracy, while limiting the impact on code performance. Such code will be developed with the goal of being run on a parallel platform that includes the GRAPE boards (Makino et al., 1997). GRAPE is a special purpose device implementing an array of fully hardwired pipelines, each one computing the gravitational interaction between two particles in a single clock cycle.

Our code stems from the pseudo-particle approach proposed by Makino (1999), and implemented by Kawai &

Makino (2001). They developed an early idea of Anderson (1992), in order to build a code able to take advantage of both the very high computing speed offered by the GRAPE, and the very high numerical precision provided by the Fast Multipole Method (FMM) (Cheng et al., 1999). FMM, similarly to the treecode, lumps particles together in order to compute force from them using a multipole expansion of the aggregate, instead of computing the contribution from each single particle. This process is hierarchically repeated in order to determine the multipole expansion of cells at higher hierarchical levels. After that, interactions among sufficiently distant cells in the same hierarchical level are computed, then such contributions are propagated downward, up to each single particle. Such cell-cell interactions, which conceptually distinguish the FMM from the treecode, let the former have a $\mathcal{O}(N)$ scaling.

When one tries to run codes using multipole expansion, such as treecode or FMM, with the GRAPE, a fundamental problem arises. GRAPE can only deal with particle-particle interactions, hence the advantage of lumping particles to obtain a single multipole expansion is wasted. GRAPE does not make use of such expansion, so that all interactions involving a multipole term must be evaluated by the host computer. The key idea of the pseudo-particle approach is to use the Anderson formulation for the multipole expansion (Anderson, 1992) which, instead of a complicate polynomial, is now given in terms of a pseudo-particle distribution. In such a way, GRAPE is also able to compute the force contribution from multipoles, since they are now expressed in terms of a particle distribution. Makino and Kawai implemented a serial pseudo-particle treecode that makes use of GRAPE (Kawai, 1999; Kawai & Makino, 2001). The algorithm they propose assigns the pseudo-particles fixed positions on a spherical surface surrounding the physical distribution, and computes the pseudo-particle masses as weighted sums of the physical particle masses. See (Makino, 1999; Kawai, 1999) for details.

Attaining high accuracy within the pseudo-particle scheme is simply a matter of increasing the number of pseudo-particles (see (Makino, 1999; Kawai, 1999) for details). of course the higher accuracy is obtained at the cost of higher compute times, and higher memory usage, that can affect the code's performance. In order to have a quantitative estimate of the error behaviour in the pseudo-particle scheme, we have carried out an error analysis study of it. We compare our results with the work of Salmon & Warren (1994). Moreover, we improve the Makino and Kawai pseudo-particle method by introducing pseudo-particle position extrapolation, in order to decrease the overhead due to pseudo-particle re-evaluation. Instead of recomputing the pseudo-particle expansion at each iteration, we extrapolate the pseudo-particle positions for a number of time-steps; in order to accomplish this, we define a pseudo-particle velocity. This leads to a substantial gain in speed-up, with acceptable accuracy deprivation. Moreover, such improved scheme is ideal for use with the GRAPE with parallel systems. In the next section, we show our error analysis, then present the moving pseudo-particle scheme.

3. Error analysis of the pseudo-particle treecode

For our error analysis purposes, we generate a particle distribution and compute, for each particle in the distribution, both the exact acceleration $\mathbf{a}$, and the acceleration given by the pseudo-particle expansion up to a given multipole $\mathbf{a}_p$. From that, the error is evaluated as:

$$\epsilon_p = \frac{|\mathbf{a} - \mathbf{a}_p|}{|\mathbf{a}|}. \tag{1}$$

The Multipole Acceptability Criterion (MAC) we adopt is the Minimal Distance (MD) criterion. According to the MD MAC

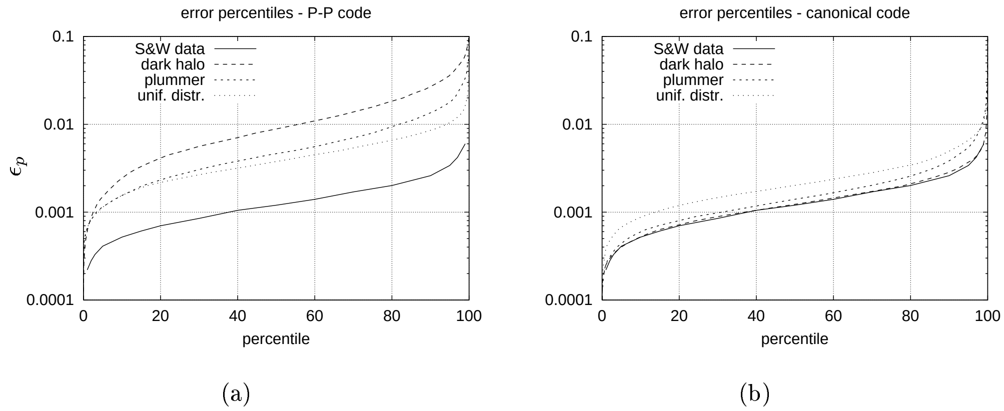

Fig. 1. Error distribution for the P-P code, and for the canonical treecode. Three different particle configurations are tested in both cases.

(see (Salmon & Warren, 1994)), a multipole expansion is accepted if

$$\frac{l}{d} < \theta \tag{2}$$

where l is the cell size, d is the minimal distance of the evaluation point from the cell, and θ is an input parameter, usually $\theta \lesssim 1$. The original MAC of Barnes & Hut (1986) differs from the MD MAC in the definition of d. Barnes and Hut define this parameter as the distance of the evaluation point from the centre of mass of the cell.

3.1. Statistical error

In order to estimate the statistical error of the pseudo-particle code, and compare it with the canonical treecode, we refer to one of the configurations in (Salmon & Warren, 1994). Namely, the 50 000 particles isolated halo data. In that experiment, 4 942 particles were chosen at random from a 50 000 particle set, and the error analysis was performed on them, using eq. (1) for the error estimation. A comparison of results using various MACs is given in fig. 11 on that paper. We operated in the same way, using the MD MAC, eq. (2), and the opening angle $\theta = 1.1$, which is the same value used by S&W. In our implementation of the pseudo-particle treecode, each non-terminal cell is associated with a pseudo-particle distribution located on the surface of the smallest sphere surrounding the cell. The pseudo-particle distribution of a parent cell is obtained recursively from the distributions of its child cells. A particle-cell interaction now becomes a set of particle-particle interactions between the particle and the pseudo-particle distribution of that cell. Multipole expansion is up to the quadrupole moment.

We noticed that the pseudo-particle expansion is very sensitive to the particle distribution. In fig. 1 the error percentile distribution for three different physical configurations is shown. We computed the relative error (eq. (1)) for the force on each particle, then obtained the percentile distribution shown in the figure. This method of analysing the error carries much more insight on the accuracy of the code, than for instance rms or maximal error. The optimal code has a flat error distribution, so that the great majority of errors have about the same value. A code with a wide spread in error values leads to a waste of compute time, since increasing the accuracy in order to reduce large errors results in unnecessary refinement of the force computations whose error was already small. See (Salmon & Warren, 1994) for a more extensive discussion.

The configurations that we used are a dark halo model such as the reference model in (Salmon & Warren, 1994), a Plummer model, and an uniform distribution. All models include 4096 particles. Fig. 1(a) contains data from the pseudo-particle code, data in panel (b) are from a canonical treecode, i.e. a treecode computing the multipole expansion in the usual way (see (McMillan & Aarseth, 1993)). It is clear how broad is the quantitative variation among the three cases in the error distribution for the pseudo-particle code, and how larger are the errors in this case with respect to the S&W data. The canonical treecode is less sensitive to the particle spatial distribution, and errors are worse for the uniform distribution, which, conversely, is the best case for the pseudo-particle code. Data for the canonical treecode running a dark halo distribution can be hardly seen because they overlap with the S&W data, as can be expected, since both refer to the same physical distribution, and forces are computed using the same scheme.

We conclude from this that the pseudo-particle expansion accuracy suffers remarkably from non uniform distribution of particles. In order to have cells with a particle distribution as much homogeneous as possible, we shifted the centre of the treecode root cell with respect to the physical distribution centre. Our results, compared with S&W results, are in fig. 2. Errors are now much smaller, having decreased by about one order of magnitude, even though the percentage of large errors is still high for non uniform distributions. In order to limit the amount of large errors, the opening angle θ must be reduced. Research is on course to improve further the pseudo-particle expansion accuracy for non uniform distributions, in order to obtain a flatter error distribution.

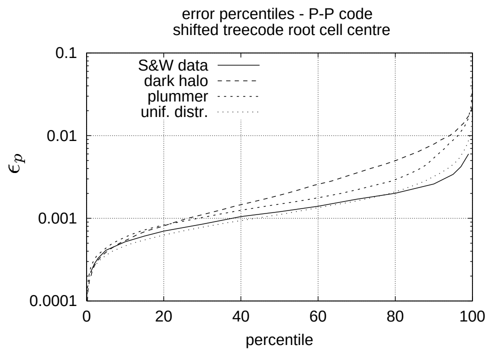

Fig. 2. Error distribution for the improved pseudo-particle code. A fat tail of large errors is still present, implying that a smaller value for the accuracy parameter θ has to be chosen.

3.2. Moving pseudo-particle scheme

The evaluation of the pseudo-particle masses is computationally relevant, especially when a high multipole order is required (see (Kawai, 1999; Makino, 1999)). Moreover, when the GRAPE hardware is used, the recomputed pseudo-particle data must be reloaded at each iteration. This introduces an high overhead, limiting the convenience of the pseudo-particle approach. We propose a scheme that does not require a re-evaluation of the pseudo-particle masses at each iteration. Instead, we assign a velocity to the pseudo-particles, and let them move according to such velocity. In such a way pseudo-particle data must be recomputed less frequently, and when GRAPE is used, no reload is necessary, since GRAPE contains the hardware needed to extrapolate particle positions. Velocities must be assigned in such a

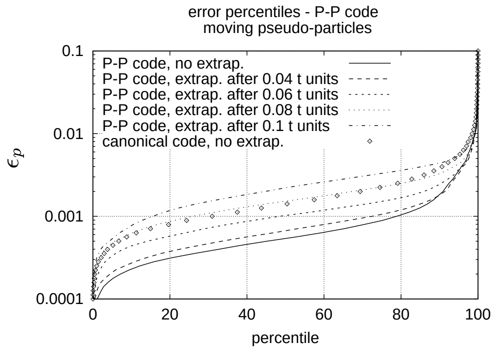

Fig. 3. Error distribution for the moving pseudo-particle code. The error profile at various extrapolation times is compared with a canonical treecode error profile. We set $\theta = 0.8$ in the P-P case, in order to limit large error values. For the canonical code is $\theta = 1.1$.

way to make the pseudo-particles motion reproduce faithfully the real system motion. We compute the pseudo-particle momenta adapting the formula used to compute the pseudo-particle mass (see (Kawai, 1999; Makino, 1999)):

$$\mathbf{p}_k = \frac{1}{K} \sum_{j_1}^{N} \mathbf{p}_j \sum_{l_0}^{P} \left(\frac{r_j}{r_a}\right)^l (2l+1)\; P_l(\cos\gamma_{jk}) \tag{3}$$

Here $\mathbf{p}_k$ is the momentum of pseudo-particle k, K is the total number of pseudo-particles, N is the number of real particles from which the pseudo-particle expansion is computed, $\mathbf{p}_j$ is the momentum of the real particle j, P is the maximal order of the multipole expansion, r_j is the norm of the position vector of particle j, r_a is the radius of the pseudo-particle sphere, P_l is the modified Legendre polynomial of order l, and finally γ_{jk} is the angle between the position vectors of particles j and k.

In order to carry out a statistical error analysis of the moving pseudo-particle scheme, we used a plummer model configuration with 4096 particles. The pseudo-particle expansion is computed only at the first iteration, after that pseudo-particle positions are extrapolated using the velocities obtained according to eq. (3).

Fig. 3 shows the error percentiles for the configuration as a function of time. Initially the extrapolated pseudo-particles reproduce the real particle dynamical configuration with an accuracy comparable with the static pseudo-particle expansion (the error profile after 0.04 time units is very close to the initial profile). Subsequently, accuracy worsens, and after 0.1 time units errors are about fourfold larger. The canonical treecode data are also shown for comparison. In order to reduce the amount of large errors, we reduced the opening angle of the pseudo-particle code to $\theta = 0.8$, whereas for the canonical code is $\theta = 1.1$. We can see that the moving pseudo-particle scheme is able to represent the dynamical evolution of the real system with satisfying accuracy up to 0.08 time units. This is not a large value for typical treecode time steps, but is very large compared to higher precision codes. For the Plummer model configuration that we used in this analysis, we have $\min(|\mathbf{v}|/|\mathbf{a}|) \simeq 0.04$, whereas the average (individual) time step in a direct code simulation is $\simeq 1.5 \cdot 10^{-4}$, at least in the early stage of evolution. This result is then promising for the use of the moving pseudo-particle scheme with high precision codes. Research is on course on the error analysis of the moving pseudo-particle scheme with higher multipole expansion, alongside performance evaluation of the scheme.

4. Methods and model for the black hole infall to the Galactic centre

4.1. Theorical aspects of the black hole infall

We study the efficiency at which a black hole or dense star cluster spirals-in to the Galactic centre. This process takes place on a dynamical friction time scale. Dynamical friction affects a mass moving in a background sea of lower mass objects. A practical expression for the strength of the drag force on a point particle with mass M_{BH} is (Binney & Tremaine 1987, p. 424):

$$\frac{d\mathbf{v}_{BH}}{dt} = -4\pi G^2 \ln\Lambda \rho M_{BH} \frac{\mathbf{v}_{BH}}{v_{BH}^3}\chi. \quad (4)$$

where

$$\chi = \mathrm{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \quad \text{and} \quad X = \frac{v_{BH}}{\sqrt{2}\,\sigma} \ ,$$

σ being the Maxwellian velocity dispersion, and ρ is the background stellar density.

The classical value of Λ is (Binney & Tremaine 1987, p. 423)

$$\Lambda = \frac{b_{max} v_{typ}^2}{G(M_{BH}+m)}. \quad (5)$$

Here b_{max} is the largest possible impact parameter for an encounter between the massive point particle and a member of the background population, v_{typ} is the typical speed of the objects in the background population, and m is the mass of each of the background stars. Equation (5) can then be generalised to

$$\Lambda = \frac{b_{max}}{b_{min}}\,. \quad (6)$$

Here b_{min} is the distance below which an encountering particle is captured, instead of being scattered by the massive object. It is somewhat smaller than the 90° turn-around distance.

McMillan & Portegies Zwart (2002) obtained an analytic expression for the distance $r(t)$ of the BH to the galactic centre, with the assumptions that the BH's orbits are nearly circular, and the mass profile of the Galaxy is given by a power law:

$$M(R) = AR^{\alpha}\,. \quad (7)$$

They obtained:

$$r(t)=R_0\left[1-\frac{\alpha(\alpha+3)}{\alpha+1}\sqrt{\frac{G}{AR_0^{\alpha+3}}}\,\chi\, M_{BH}\, \ln\Lambda\, t\right]^{\frac{2}{3+\alpha}} . \quad (8)$$

We take R_0 equal to the half-mass radius R_{hm} (see Section 4.5). The best fit of equation (8) on the simulation data gives the value of $\ln\Lambda$ for that simulation.

4.2. Direct method

For our direct N-body calculations we used the `kira` integrator module of the Starlab software environment[1] (Portegies Zwart et al., 2001). Conceived and written as an independent alternative to Aarseth's NBODY4 and NBODY5 (Aarseth, 1985, 1999), the workhorses of collisional N-body calculations for the past 25 years, `kira` is a high-order predictor-corrector scheme designed for simulations of collisional stellar systems. This integrator incorporates a Hermite integration scheme (Makino & Aarseth, 1992) and a block time step scheduler (McMillan, 1986) that allows homogeneous treatment of all objects in the system.

While `kira` is designed to operate efficiently on general-purpose computers, it achieves by far its greatest speed when combined with GRAPE-6 special purpose hardware (Makino et al., 2002)[2]. For the work presented here we performed simulations with the GRAPE-6 system at the University of Tokyo with up to 80 000 particles.

4.3. Treecode

Generals on the treecode have been presented in Section 2. Our treecode simulations were initially performed with both a code written by Jun Makino (Makino, 1991), and with GADGET (Springel et al., 2001). In GADGET each particle is assigned an individual time-step, and at each iteration only those particles having an update time below a certain time are selected for force evaluation. This criterion was originally introduced in the direct N-body code (*cf.* Aarseth 1999). This code is parallelized using MPI (Message Passing Interface Forum, 1997). In the parallel version, the geometrical domain is partitioned, and each processor hosts the particles located in the domain partition assigned to it.

[1] See: `http://manybody.org`
[2] See: `http://www.astrogrape.org`

The computation of forces on the selected i-particles is performed by scattering the particle data to remote processors. Then partial forces from the particles hosted by the remote processors are computed locally. Finally, calculated forces are received back by the i-particle host, and added up resulting in the total force on the i-particles.

4.4. Particle-mesh code

To perform calculations using several millions of particles we use a particle-mesh (PM) code named SUPERBOX (Fellhauer et al., 2000). With the particle-mesh technique densities are derived on Cartesian grids. Using a fast Fourier transform algorithm these densities are converted into a grid-based potential. Forces acting on the particles are calculated using these grid-based potentials, making the code nearly collisionless. The current implementation completely neglects two-body relaxation causing it to retain only a small amount of grid-based relaxation (Fellhauer et al., 2000).

The adopted implementation incorporates some differences to standard PM-codes. State of the art PM-codes are using a cloud-in-cell (CIC) scheme to assign the masses of the particles to the grid-cells; which means that the mass of a particle i is split up into neighbouring cells according to its distance to the centre of the cell. Forces are then calculated by adding up the same fractions of the forces from all cells to particle i. In contrast our code uses the "old-fashioned" nearest-grid-point scheme, where the total mass of the particle is assigned to the grid-cell the particle is located in. Forces acting on the particle are then calculated only from the forces acting on this particular cell. To achieve similar precision as CIC, we use space derivatives up to the second order.

To achieve high resolution at the places of interest, we incorporate for every simulated object (e.g. each galaxy and/or star cluster or disc, bulge and halo) two levels of sub-grids which stay focused on the objects of interest while they are moving through the simulated area. This provides higher resolution only where it is necessary. Our PM-code is parallelized using MPI.

4.5. Initial condition

We generate the initial mass distribution according to the power law given by equation (7), with $\alpha = 1.2$, which reproduces the mass distribution in the centre of the Galaxy, according to Mezger et al. (1999). The scale factor is $A = 4.25 \cdot 10^6 \mathrm{M}_\odot$, corresponding to 0.44 in the N-body standard units (Heggie & Mathieu, 1985), which are reported in Table 1. We use the standard units hereafter, unless other units are explicitly reported. The distributions that we generate are truncated at $R = 1.7 = 13.6\,\mathrm{pc}$, with a total mass within this radius $M_{tot} = 1$. The particles have equal mass m. Particles are assigned Maxwellian velocities, then the system is virialised to dynamical equilibrium.

The BH particle is placed at the half-mass radius $R_{hm} \simeq 0.87$ with a circular orbit velocity, and its mass is $M_{BH} = 0.000528$. The background particles number varies from 16 000 to 2 million. The low particle number simulations are performed with the PP code, the intermediate and high number simulations with the treecode and the PM code. This allows us to span a large range in particle number, so that the influence of granularity in the BH motion towards the Galaxy centre can be studied.

In contrast to the other models, we choose physical units for the PM code simulations. The conversion factors from physical units to N-body units are shown in Table 1, where l.u. denotes the unit length in N-body units, m.u. the unit mass, and t.u. the unit time.

5. Black hole infall - Results

We will now study the dependence of our results on the number of particles N in Section 5.1, and compare the various N-

Table 1. Conversion table between the N-body units used in our work, and physical units. The N-body units are such that $G = 1$, $M_{tot} = 1$, and $E_{tot} = -0.25$.

$$
\begin{aligned}
G &= 1 \text{ v.u.}^2 \text{l.u./m.u.} \\
1 \text{ l.u.} &= 8 \text{ pc} \\
1 \text{ m.u.} &= 1.18 \cdot 10^8 \text{ M}_\odot \\
1 \text{ t.u.} &= \sqrt{\frac{1 \text{ l.u.}^3}{G \cdot 1 \text{ m.u.}}} = 0.031 \text{ Myr}
\end{aligned}
$$

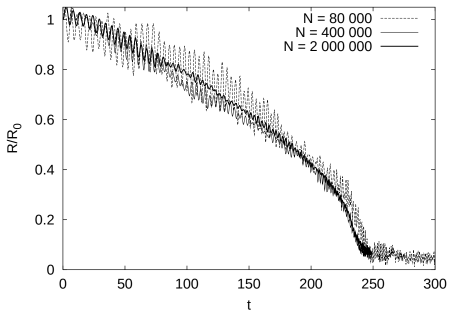

Fig. 4. Time evolution of the radial distance of the black hole to the Galactic centre. Time is measured in N-body units. The various curves (identified in the top right corner) present results obtained with the treecode. One N-body time unit corresponds to about 0.031 Myr. The distance of the black hole to the Galactic centre (Y-axis) is given in terms of its initial distance. In these simulations is $\epsilon = 0.003735 \simeq 0.03$ pc and $M_{BH} = 0.000528$.

body methods with identical initial realisations in Section 5.2. We continue by studying the effect of softening/cell size (Section 5.3) on the black hole infall to the centre of the Galaxy.

5.1. Dependence of the infall on N

In Fig. 4 we show the evolution of the BH distance from the centre of mass

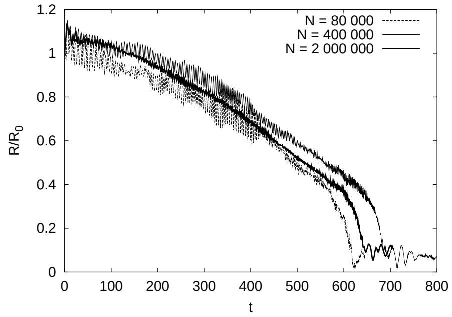

Fig. 5. Same as Fig. 4 above, but for PM code simulations. The intermediate grid cell size is here $l = 0.69$ pc, and $M_{BH} = 0.000528$.

of the system for three treecode simulations. Values are measured in dimensionless N-body units. N varies from 80 000 to 2 million, with $\epsilon = 0.003735$ in dimensionless N-body units, corresponding to about 0.03 pc. In Fig. 5 we present a similar figure from PM code simulations. Here is $N \in \{80\,000, 400\,000, 2\,000\,000\}$, with 32 cells per dimension, resulting in a cell size of about 0.69 pc.

Figs 4 and 5 show that increasing N results in a much smoother motion of the BH in its infall towards the centre of the Galaxy. The BH infall rate is not much affected by a change in N. Accordingly, we can conclude that the infall in each of the two sets above is consistent.

5.2. Comparison of the codes

In this section we compare the results obtained from the various codes, to check their consistency. The comparison of the PM results with the two other codes results is particularly critical, since the PM code computes forces using a different mathematical approach, i.e. a grid based force derivation vs a direct particle-particle computation for the particle-particle (PP) code, or particle-multipole computation for the treecode. A consequence of this is a different parameter to tune the accuracy of the simulation, namely the cell size l for the

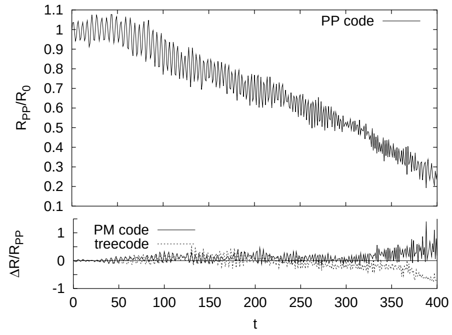

Fig. 6. Top panel shows a black hole infall simulated by the PP code, with $N = 80\,000$, $M_{BH} = 0.000528$ and $\epsilon = 0.02988$. Bottom panel shows a comparison of the PP results with treecode and PM results. $\Delta R = (R - R_{PP})$.

PM code, and the softening length ϵ for the other two codes. We will study here how these two parameters influence the black hole infall.

In Fig. 6 we show the time evolution of the galactocentric BH distance R simulated by the PP code, accompanied by a plot of the time evolution of $\Delta R/R_{PP}$ for treecode and PM simulations, where $\Delta R = (R - R_{PP})$. The relative difference $\Delta R/R_{PP}$ remains on average small for a large fraction of the infall. The final discrepancy is mostly due to the small values of the quantities at that point, which are likely to amplify relative differences. The BH infall is predicted with very good consistency among the codes.

5.3. Measurement of $\ln\Lambda$

The efficiency of the infall of the BH to the Galactic centre depends on the value of the Coulomb logarithm $\ln\Lambda$, defined in eq. (5). In turn, $\ln\Lambda$ is affected by the softening parameter ϵ used in the treecode and PP code, and the cell length l in the PM code.

The influence of the softening parameter on the BH dynamics has been studied by running a number of simulations with the three codes. In Table 2 we report the value of $\ln\Lambda$ obtained from our simulations. For the PP code and treecode simulations, we increase ϵ from 0 to $0.1195 \simeq 0.96\,$pc. For the PM code, we increase l from 0.00934 to 0.0859. In all cases is $M_{BH} = 0.528 \cdot 10^{-3} \simeq 62\,300\,\mathrm{M}_\odot$.

Table 2. $\ln\Lambda$ versus ϵ from PP code, treecode, and PM code runs. For the PP code and treecode runs is $N = 80\,000$. For the PM code runs is $N = 2\,$million.The reference value for the accuracy parameter is $\epsilon_0 = 0.003735$.

ϵ/ϵ_0	PP code	treecode
0	6.6	
0.01	6.0	
0.1	5.3	5.7
1	4.8	4.7
2	3.5	4.1
8	2.8	3.0
16	1.8	2.0
32		1.6

l/ϵ_0	PM code
2.5	4.1
5	3.8
10	3.0
23	2.2

For the PP code and the treecode, we selected $N = 80\,000$ as a suitable value. The relaxation time is for this value of N $t_x \simeq 0.1N/\ln N \cdot R/v_{typ} \simeq 2000$, about one order of magnitude larger than the typical BH infall time, so that the system is collisionless, and we can confidently use the treecode to simulate it. With this choice for N, the BH mass is $M_{BH}/m \simeq 42.3$. We did not increase ϵ above $\simeq 0.12$, since at this point ϵ is already much bigger than b_{min}, and the infall time is now close to t_x.

For the PM code simulations, we used $N = 2\,$million in order to have enough particles to fill all the cells, even for the simulations with a small l. Since a PM simulation gives incorrect results for $N_c \ll 1$, we did not decrease l below 2.5 since in such case we have already $N_c \simeq 0.1$.

The decrease of the value of $\ln\Lambda$ as ϵ or l increases is clear from Table 2. In (Spinnato et al., 2002) we focus on the relation between Λ and ϵ, and provide a fitting formula for $\ln\Lambda(\epsilon)$. From the PP value for $\epsilon = 0$, we can conclude that $\ln\Lambda = 6.6$ in the Galactic centre.

6. Conclusions

We presented an error analysis of both the pseudo-particle tree-code, and the moving pseudo-particle scheme that we propose in order to reduce the compute time, and optimise the use of such method with the GRAPE Special Purpose Device. We showed the error behaviour of this method, and compared it with previous work on the Barnes-Hut tree-code. We introduced a simple technical modification in the code, that leads to an error decrease. We also showed how the error of our moving pseudo-particle scheme remains well within acceptable values for a large time interval, making it suitable for the use with high accuracy tree-codes.

We simulated the evolution of a massive particle in a sea of lighter particles in a self gravitating system. The main goal of this work is to obtain an accurate value of the Coulomb logarithm ($\ln\Lambda$) (see (Spinnato et al., 2002)). This helps us to understand the dynamics of the Galactic bulge and the rate at which intermediate mass black holes sink to the Galactic centre. We ran both N-body particle-particle (PP) simulations, softened treecode simulations, and particle-mesh (PM) simulations. The comparative simulations are performed for 80 000 particles, and all result in a consistent BH infall. We performed simulations with up to 2 million particles using a treecode and a PM code. The obtained BH infall does not depend on the number of particles. Apparently, 80 000 particles is already enough to elimitate any granularity for our choice of initial conditions. The results of the treecode, at the low N-limit, are in excellent agreement with the PP simulations. We conclude that $\ln\Lambda = 6.6$ in the Galactic centre.

Acknowledgements. The authors acknowledge Tokyo University and Jun Makino for the access to the GRAPE-6 system, ASCI (Dutch Advanced School for Computing and Imaging) for the access to the DAS2 supercomputer, Piet Hut and the Starlab collaboration for the Starlab software, and Volker Springel and the GADGET team for the treecode we used, Steve McMillan for stimulating discussions, NOVA (Dutch Research School for Astronomy) for grant V-37 for MF visit to Amsterdam. PFS acknowledges the support of KNAW (Royal Netherlands Academy of Arts and Sciences) under grant 95-BTN-15, and NWO (Netherlands Organisation for Scientific Research) under Spinoza grant 08-0 to E.P.J. van den Heuvel.

References

Aarseth S. J., 1985, in Brackbill J., Cohen B., eds, Multiple Time Scales. Academic Press, Orlando, p. 378

Aarseth S. J., 1999, PASP, 111, 1333

Anderson C. R., 1992, SIAM J. of Scientific and Statistical Computing, 13, 923

Barnes J. E., Hut P., 1986, Nature, 324, 446

Binney J., Tremaine S., 1987, Galactic Dynamics. Princeton University Press, Princeton, New Jersey

Cheng H., Greengard L., Rokhlin V., 1999, J. Comput. Phys., 155, 468

Fellhauer M., Kroupa P., Baumgardt H., Bien R., Boily C., Spurzem R., Wassmer N., 2000, NewA, 5, 305

Heggie D., Mathieu R., 1985, in McMillan S. L. W., Hut P., eds, The Use of Supercomputers in Stellar Dynamics. Springer Verlag, Berlin, pp 233–235

Kawai A., 1999, PhD thesis, University of Tokyo

Kawai A., Makino J., 2001, ApJ, 550, L143

McMillan S. L. W., 1986, in McMillan S. L. W., Hut P., eds, The Use of Supercomputers in Stellar Dynamics. Springer Verlag, Berlin, p. 156

McMillan S. L. W., Aarseth S. J., 1993, ApJ, 414, 200

McMillan S. L. W., Portegies Zwart S. F., 2002, submitted to ApJ
Makino J., 1991, PASJ, 43, 621
Makino J., 1999, J. Comput. Phys., 151, 910
Makino J., Aarseth S. J., 1992, PASJ, 44, 141
Makino J., Fukushige T., Namura K., 2002, in preparation
Makino J., Taiji M., Ebisuzaki T., Sugimoto D., 1997, ApJ, 480, 432
Message Passing Interface Forum 1997, MPI-2: Extensions to the Message-Passing Interface. University of Tennessee, Knoxville, Tennessee
Mezger P., Zylka R., Philipp S., Launhardt R., 1999, A&A, 348, 457
Portegies Zwart S. F., McMillan S. L. W., Hut P., Makino J., 2001, MNRAS, 321, 199
Salmon J. K., Warren M. S., 1994, J. Comput. Phys., 111, 136
Spinnato P. F., Fellahuer M., Portegies Zwart S. F., 2002, MNRAS, in preparation
Springel V., Yoshida N., White S. D. M., 2001, NewA, 6, 79