



UvA-DARE (Digital Academic Repository)

Hybrid Systems for N-body Simulations

Spinnato, P.F.

Publication date
2003

[Link to publication](#)

Citation for published version (APA):

Spinnato, P. F. (2003). *Hybrid Systems for N-body Simulations*. Eigen Beheer.

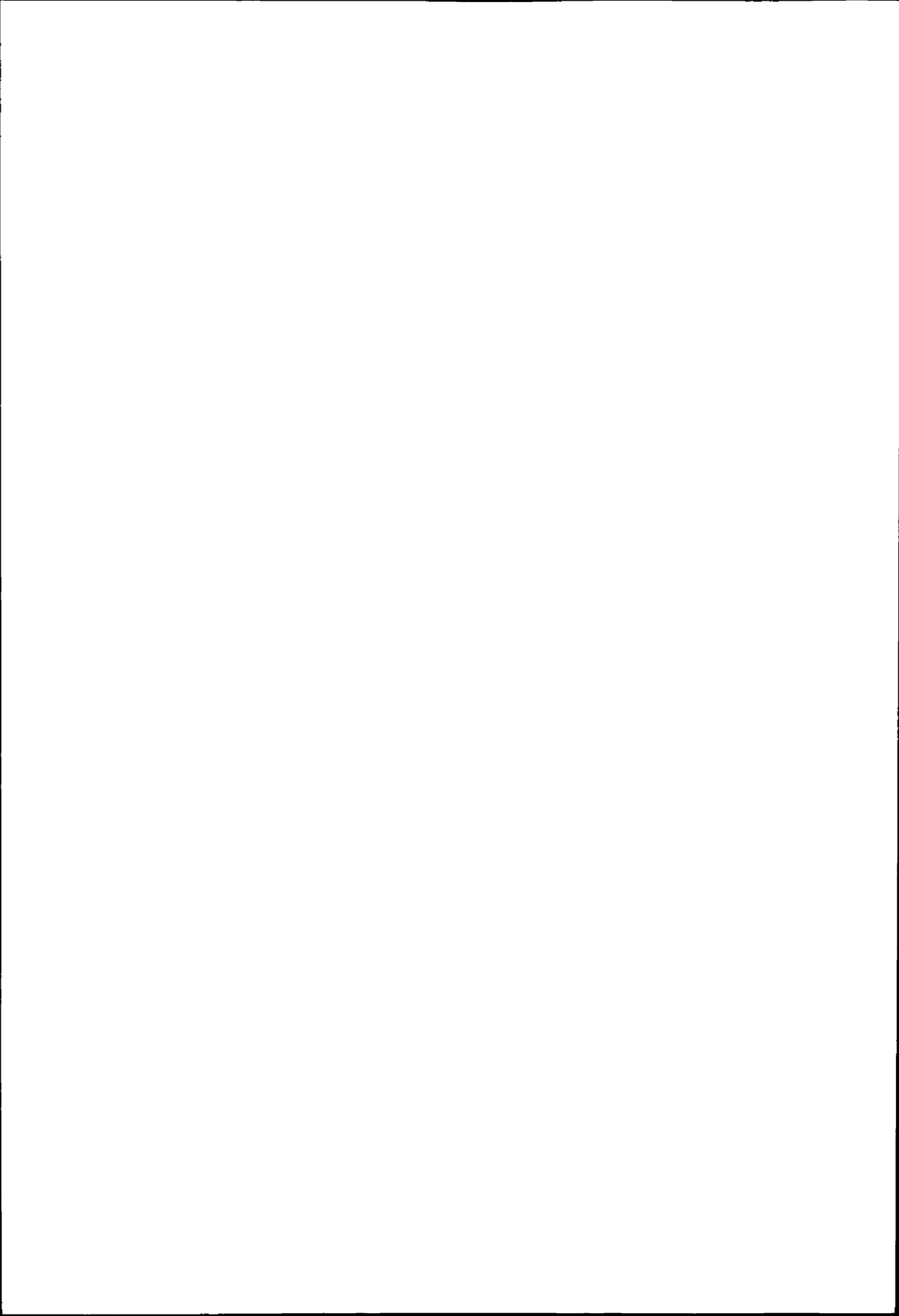
General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Part III
Conclusions



In this thesis we studied the use of hybrid architectures as a tool to accelerate the numerical solution of the N -body problem. We can define the N -body problem as the challenge to understand the motion of N point-like particles, subject to their mutual interactions. This definition is inspired by the one given on page 14 of the profound and delightful book on the gravitational million body problem, written by Douglas Heggie and Piet Hut (2003). In this thesis, we also focus on the gravitational N -body problem because of its highly demanding computational requirements, that make it interesting and stimulating for a computational scientist to examine the computational characteristics, and explore the possible avenues to speed-up the problem's numerical solution.

The analytic insolubility of the gravitational N -body problem for $N > 2$ led to the development of numerical techniques to study it, as discussed in section 1.2. The need to retain the full $\mathcal{O}(N^2)$ direct particle-particle scheme for the simulation of collisional systems, i.e. systems requiring high computational accuracy (see section 1.2), led in turn to the development of the GRAPE, specialised hardware to accelerate the computation of the gravitational force interactions, introduced in section 1.3.

The gravitational force computational kernel of the $\mathcal{O}(N^2)$ scheme is so demanding in terms of hardware power, that its execution on the one Tflop/s GRAPE-4 can be driven by a host workstation which is about 10^4 times slower (see section 1.2, fig. 1.5, and our performance analysis work presented in chapter 2).

Our research aims at exploring the possibilities of using the GRAPE to boost N -body simulations other than those of astronomical collisional systems, which is the native domain of application of the GRAPE. For the simulation of collisionless systems, fast and more sophisticated methods like the treecode, the FMM (Fast Multipole Method), or the PM (Particle-Mesh), have been developed, as described in section 1.2. They reduce the computational complexity of the N -body problem to $\mathcal{O}(N \log N)$, and even $\mathcal{O}(N)$, trading higher speed for lower accuracy.

Using the GRAPE with these methods is not straightforward, since the particle-particle force computation, while still relevant, is no longer the most computationally expensive task of the application. In this case, the relatively high computational load of the GRAPE-host

could make the host activity the system bottleneck.

The use of Performance Modelling techniques allows us to study the interplay of the general purpose host, the special purpose device, and the application executed on the hybrid architecture. Chapter 3 describes our performance models of hybrid architectures which include GRAPE boards to accelerate the execution of direct particle-particle codes and treecodes. We show that a distributed hybrid architecture running a treecode optimised for the GRAPE has the potential to outperform a serial-host GRAPE4-like monolithic system running a direct code, even for simulations requiring high computational accuracy.

This supports our idea of hybrid architectures as an effective tool to speed up the execution of applications characterised by a heavy and small computational kernel, which is amenable to hardware implementation, but also including auxiliary relevant computational requirements. Further support for the hybrid architecture paradigm comes from the fact that the GRAPE-6 architecture (see section 1.3) is essentially an instance of this class of computer systems.

In order to efficiently use the hybrid architecture, the software application that is run on it needs to be fine-tuned. A fine-tuning of the treecode to run optimally on a platform which includes the GRAPE is the pseudo-particle scheme (see chapter 4). In this formulation, the gravitational potential of the particle aggregates created by the treecode domain partition, is converted from multipole expansion to a pseudo-particle distribution. In this way, the GRAPE is also able to compute the force due to particle aggregates, allowing for an optimal execution of the treecode on the hybrid machine.

In chapter 4 we described the pseudo-particle approach, and presented the accuracy improvement that we developed. We showed that the original pseudo-particle formulation (Makino, 1999; Kawai & Makino, 2001) is less accurate than the canonical multipole-based treecode, especially in the case of highly inhomogeneous matter distributions. We introduced an improvement, consisting in an extra particle added to the pseudo-particle set, located at the centre of mass of the real particle distribution. In such a way the pseudo-particles approximate inhomogeneous real particle distributions more closely, with a significant accuracy benefit.

We also developed a temporal expansion scheme for the pseudo-particle approach, where we define a pseudo-particle velocity. In this way, we do not re-compute the pseudo-particles at each time step, but we let them move, following the real particle evolution. This not only reduces the overhead from calculating the pseudo-particle, but also optimises the communication with the GRAPE. The GRAPE needs to know the pseudo-particle expansion of particle aggregates, which must be communicated by the host at each time step. With our scheme, pseudo-particle expansions can be retained in the GRAPE memory for a number of time steps, and evolved locally by the GRAPE extrapolation pipeline (see section 1.3 and fig. 1.4), because pseudo-particles now have a velocity assigned to them.

In chapter 5 we presented an actual example of N -body simulative study. Namely, we carried out a comparative study of the infall of a massive black hole towards the Galactic centre, in order to measure the Coulomb logarithm. The Coulomb logarithm is the parameter that quantifies the efficiency of a massive body slow-down due to its gravitational interaction

with a background of lighter stars. If the body is orbiting a centre of gravity, as in our case, its deceleration results in an inspiraling trajectory.

We used a direct particle-particle code, a treecode, and a particle-mesh code for our simulations, in order to understand how particle granularity and code inaccuracy influence the measure of the Coulomb logarithm. The direct code is accurate, but limited in the number of particles because of its $\mathcal{O}(N^2)$ scaling. The other codes are less accurate, but allow for higher numbers of particles, hence low particle granularity, which gives a smoother representation of the gravity field. Our measure of the Coulomb logarithm appears to be independent of the number of particles, at least for the range of physical parameters chosen. The parameter that quantifies the numerical inaccuracy influences the value of the Coulomb logarithm. The logarithm argument is inversely proportional to the inaccuracy parameter.

Future work

The work presented in chapter 5 is the first step towards the study of the infall of a star cluster, i.e. an extended object. As described in section 1.7, phenomena of this kind, consisting of the interaction of a collisional and a collisionless system, are not straightforward to simulate, either with a direct code or an approximate method. Our solution is to devise a hybrid code, where the star cluster collisional “phase” is simulated by the direct code, and the Galactic centre collisionless “phase” is simulated by the treecode.

This hybrid code makes high demands of the host computing platform, which must provide both high computational power for the gravitational force evaluation, and comparably high power for the other tasks of the software, otherwise the treecode execution would become the system bottleneck. A hybrid architecture of the kind discussed in this thesis would be an ideal solution for this problem.

Our group is planning to realise a hybrid system including GRAPE-6 boards, configured to optimally run the hybrid code. The computational asymmetry inherent in the coexistence of a direct code “phase” and a treecode “phase” calls for a corresponding asymmetry in the hardware architecture. The direct code requires a large computational power for the gravitational force kernel evaluation, and has few requirements for the other tasks. A single host having a number of GRAPE boards connected to it would fulfill these needs. On the other hand, the treecode has a relatively lower need for the force evaluation part, but needs more power for the other general purpose tasks. A homogeneous distributed configuration with each node of a parallel host connected to the same number of GRAPEs would be more appropriate in this case.

The ideal architecture to run the hybrid code would then be an amalgam of the two platforms. One node should be connected to a higher number of GRAPEs with respect to the other nodes. A performance model will be very useful to configure this system, and fine-tune it to obtain the best performance when running the hybrid code. Our work on performance modelling that was presented in part I will be the basis of this model.

We have presented an example of the potential of hybrid architectures to solve specific,

highly demanding computational problems. We focussed on hybrid architectures for the gravitational N -body problem, but other fields of Computational Science can profit from this architectural paradigm. The use of special hardware like the APE series (Tripiccione, 1999) developed for Quantum Chromo-Dynamics simulations, or the MDM machine (Narumi *et al.*, 1999, 2000), the sibling of GRAPE used for Molecular Dynamics simulations and mentioned in section 1.3.3, can be enhanced by a hybrid architecture approach. We believe that hybrid architectures can be the platform of choice for a sizable number of computational scientists. With this thesis we hope to contribute to a move in this direction.