*Star clusters and black holes in galaxies across cosmic time*
*Proceedings IAU Symposium No. 312, 2014*
*Y. Meiron, S. Li, F.-K. Liu & R. Spurzem, eds.*

doi:10.1017/S1743921315007954

# GraviDy: a modular, GPU-based, direct-summation *N*-body code

**Cristián Maureira-Fredes and Pau Amaro-Seoane**

Max-Planck-Institut für Gravitationsphysik (Albert Einstein Institut Potsdam-Golm, Germany
email: cristian.maureira.fredes@aei.mpg.de

**Abstract.** The direct-summation of $N$ gravitational forces is a complex problem for which there is no analytical solution. Dense stellar systems such as galactic nuclei and stellar clusters are the loci of different interesting problems. In this work we present a new GPU, direct-summation $N$-body integrator written from scratch and based on the Hermite scheme. The first release of the code consists of the Hermite integrator for a system of $N$ bodies with softening. We find an acceleration factor of about $\approx 90$ of the GPU version in a single node as compared to the Serial-Single-CPU one. We additionally investigate the impact of using softening in the dynamics of a dense cluster. We study how it affects the two body relaxation, as compared with another code, NBODY6, which uses KS regularization, so as to understand the role of softening in the evolution of the system. This initial release is the first step towards more and more realistic scenarios, starting for a proper treatment for binary evolution, close encounters and the role of a massive black hole.

**Keywords.** $N$-body, Hermite integrator, GPU

## 1. Introduction

The dynamical evolution of a dense stellar system such as e.g. a globular cluster or a galactic nucleus has been addressed extensively by a number of authors. For Newtonian systems compounded by more than two stars we must rely on numerical approaches in the general case which provide us with solutions which are more or less accurate. A very well known example is the family of direct-summation $N$-body integrators of Aarseth for dense stellar systems (see Aarseth 1999; Spurzem 1999; Aarseth 2003† or also KIRA Portegies Zwart *et al.* 2001‡). The progress in both software and hardware has put us now in a position in which we start to get closer and closer to simulate realistic systems. There has been recently an effort at porting existing codes to graphics processing units (GPUs), like e.g. the work of Portegies Zwart *et al.* (2007); Hamada & Iitaka (2007); Belleman *et al.* (2008) on single nodes or using large GPU clusters Berczik *et al.* (2011); Nitadori & Aarseth (2012); Capuzzo-Dolcetta *et al.* (2013).

## 2. Integrator

To numerically integrate the system of equations we adopt the widely known 4th-order Hermite integrator (H4 henceforth) presented in (Makino & Aarseth 1992; and see also Aarseth 1999, 2003), which is a scheme based on a predictor-corrector scenario, in other words, the extrapolation and interpolation of the equations of motion. An advantage of the choice for H4 is that we can use the family of Aarseth's codes as a test for our implementation.

† All versions of the code are publicly available at the URL
`http://www.ast.cam.ac.uk/ sverre/web/pages/nbody.htm`
‡ `http://www.sns.ias.edu/ starlab/`

## 3. Code structure

We present here the structure of our new $N$-body code developed purely in C/C++, using CUDA/MPI/OpenMP for parallelization. One of the main concerns of GRAVIDY is to treat our $N$-body code as a piece of software, being written using an "Iterative and incremental development", which is a software development methodology similar to the development APOD cycle (CUDA Best Practices, NVIDIA 2012; Assess, Parallelize, Optimize and Deploy). Another key point in the development of GRAVIDY is its legibility. We have focused on making it easy to read and modify by other users or potential future developers without compromising the computational performance of the algorithm. The code organization was thought to be highly modular, in other words, every critical process of the integrator is represented by a separated class or chain of functions. Because maintainability is one of our design goals, its documentation is also a critical factor (using Doxygen) to explain the role of every function.

## 4. Testing the code

We performed a number of tests to measure the performance and the accuracy of GRAVIDY using different amount of particles, which agreed with the previous work.

## 5. On-going and future work

We are currently working on the implementation of a binary treatment which is based on the time-symmetric Hermite 4th order proposed by Kokubo *et al.* (1998) and implemented by Konstantinidis & Kokkotas (2010). On the other hand, we are testing the adaptation of an Hermite 6th order scheme, as a alternative for the current 4th order. And finally we are working on the multi-GPU implementation of GRAVIDY. A detailed version of this work is under development and will be published soon.

### References

Aarseth, S. J. 1999, *The Publications of the Astronomical Society of the Pacific*, 111, 1333-1346.
Spurzem, R. 1999, *Journal of Computational and Applied Mathematics*, 109, 407-432.
Aarseth, S. J. 2003, ISBN 0521432723.
Portegies Zwart, S. F., McMillan, S. L. W., Hut, P., & Makino, J. 2001, *MNRAS*, 321, 199-226.
Portegies Zwart, S. F., Belleman, R. G., & Geldof, P. M. 2007, *New A*, 12, 641-650.
Hamada, T. & Iitaka, T. 2007, *New A*.
Belleman, R. G., Bédorf, J., &Portegies Zwart, S. F. 2008, *New A*, 13, 103-112.
Berczik, P., Nitadori, K., Zhong, S., Spurzem, R., Hamada, T., Wang, X., Berentzen, I., Veles, A., & Ge, W. 2011, *International conference on High Performance Computing*, 8-18.
Nitadori, K. & Aarseth, S. J. 2012, *MNRAS*, 424, 545-552.
Capuzzo-Dolcetta, R., Spera, M., & Punzo, D. 2013, *Journal of Computational Physics*, 236, 580-593.
Kokubo, E., Yoshinaga, K., & Makino, J. 1998, *MNRAS*, 297, 1067-1072.
Konstantinidis, S. & Kokkotas, K. D. 2010, *A&A*, 522, A70.
Makino, J. & Aarseth, S. J. 1992, *PASJ*, 44, 141-151.

https://doi.org/10.1017/S1743921315007954 Published online by Cambridge University Press