Journal of Computational Physics 355 (2018) 582–596

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Tree code for collision detection of large numbers of particles applied to the Breit–Wheeler process

O. Jansen *, E. d'Humières, X. Ribeyre, S. Jequier, V.T. Tikhonchuk

Univ. Bordeaux/CNRS/CEA, Centre Lasers Intenses et Applications, UMR 5107, Talence 33405, France

ARTICLE INFO

Article history:
Received 1 August 2016
Received in revised form 25 September 2017
Accepted 15 November 2017
Available online 21 November 2017

Keywords:
Tree code
Collision detection
QED
Breit–Wheeler process
Pair creation
Astrophysics

ABSTRACT

Collision detection of a large number N of particles can be challenging. Directly testing N particles for collisions among each other leads to N^2 queries. Especially in scenarios, where fast, densely packed particles interact, challenges arise for classical methods like Particle-in-Cell or Monte-Carlo. Modern collision detection methods utilising bounding volume hierarchies are suitable to overcome these challenges and allow a detailed analysis of the interaction of large number of particles. This approach is applied to the analysis of the collision of two photon beams leading to the creation of electron–positron pairs.

1. Introduction

Modelling a large number of particles often is a challenge in physics. Many-body problems are well known in astronomy, plasma physics, solid state physics and other disciplines. In astrophysics a common way to overcome the challenge of simulating a many-body problem, like the movement of stars of one galaxy under each others gravitational force, is to use the Barnes–Hut (BH) method [1]. In a BH simulation space is partitioned in an hierarchic octree structure. The tree branches grow towards successive smaller volumes of space in such way as to include at maximum one particle (star) in each leaf node, while still covering the entirety of the simulation domain. Only particles, that are in partitions close to each other, interact. Other octree-based codes exist [2], [3], that deal with the interaction of particles in a mesh-less simulation codes, usually focusing on Monte-Carlo algorithms for computing collisions.

In plasma physics Particle-in-Cell (PIC) codes [4][5] are regularly used. There, the simulated domain is structured in a mesh, with fields being defined on the nodes of that grid, while particles, combined in mono-energetic clouds called *macro-particles* (MPs), can move freely in space. Vlasov- and Fokker–Planck codes as well are quite common in plasma physics. Tree codes like the BH in plasma physics are not very common, but exist [6].

For many problems in solid-state physics Monte-Carlo (MC) [7] and Molecular Dynamics (MD) [8][9] simulations are well suited.

Many of these simulations can require a lot of computational power. Dealing with collision detection in systems with many particles even further increases the burden on computational performance. In order to test an ensemble of N particles on all possible collisions, one needs N^2 queries. This can easily become a computational effort too large to handle, which

* Corresponding author.
E-mail address: oliver.jansen@celia.u-bordeaux.fr (O. Jansen).

https://doi.org/10.1016/j.jcp.2017.11.021
0021-9991/

calls for different approaches and approximations in simulations. In PIC codes collisions usually are handled statistically and in a probabilistic fashion using MC methods or binary collision models [10]. In BH and MD simulations collisions usually are not treated directly, but via the potentials generated by the particles.

In other fields direct collision detections of moderate numbers of particles are very common. In airport short-term conflict alert systems compute possible collisions between aircrafts in order to avoid them. In video games and computer-generated imagery (CGI) over the years more and more objects interact in the same simulation environment with increasingly higher details. In the cases of airport traffic control and video games, performance is of utmost importance, since all calculations have to be done in real time. However, precision can not be allowed to suffer from the high demand on performance. In fact, modern methods are even more precise than their predecessors.

In the following we present a method, that combines some of the before mentioned techniques into one powerful and versatile tool for computing collisions of a very large number of particles. The main difference to already successfully published octree-methods lies in the partitioning of the phase-space. By not just taking the distance between particles into account for partitioning, but also their energy, we are able to significantly reduce the number of necessary collision tests.

We present results from simulations, computed on a single workstation without parallelisation, including collisions of about 10^6 particles in a single particle picture, up to 10^9 particles in a slightly restrained way and arbitrary number of particles in a statistical approach. These results are obtained by using the scheme called Bounding Volume Tree Hierarchy Simulation for Interactions in Large Ensembles (TrI LEns), which adaptively partitions the phase space in bounding volumes for efficiently pruning the problem of collision detection. Our main motivation is to develop a method, that allows an accurate single-particle description of collisions while dealing with a very large number of particles. It is important for us to obtain a tool, that can be used in different regimes with arbitrary large numbers of particles, even though the form of representation of particles and collision might change between regimes. This way we are able to investigate a transition from one regime to another and compare results from analytical predictions, simulation results with detailed descriptions of individual particles and statistical approximations.

The presented method is designed to be used in a stand-alone tree code, as presented in this paper, or as a module for other types of simulation (PIC for example) in order to help with the challenging task of collision detection. Our focus is on the concept of using so-called bounding volume hierarchies to improve binary collision tests.

In recent years, the Breit–Wheeler (BW) process [11] of the binary collisions of photons with creation of electron–positron pairs has become of particular interest in the fundamental research and astrophysics. The BW process is the inverse process to the annihilation of an electron and a positron into two photons and it is therefore, the most basic process for the creation of matter from light. It is believed to be an important aspect in gamma ray burst [12], active galactic nuclei, black holes and other large-scale explosive phenomena [13] in the universe. The BW process also is responsible for the TeV cutoff in the photon energy spectrum of extra-galactic sources [14].

An experiment at the Stanford Linear Accelerator Center [15][16] was able to detect pairs created by the collision of photons, however, was unable to reach the regime of the linear BW process, in which exactly two photons create one electron and one positron. Ribeyre et al. proposed a novel experiment [17] in order to investigate the BW process in laboratory conditions with a significant number of particles being created and detected with a minimum of noise sources. The actual collision of photons in this scheme takes place in a small volume in vacuum, far away from any sources of trident or Bethe–Heitler [21] pair creation. This experimental set-up could be implemented in soon upcoming facilities like Apollon [22] or ELI-NP [23]. A more detailed analysis of the specific implementation of the experiment will be handled elsewhere. Even though, the dynamics of two photons in the BW process can easily be calculated, a large number of photons colliding in a small volume can become difficult to handle. In the following, we present a way of how this simulation code can be applied to investigations of the BW process. For the BW process the total cross-section, that defines the probability of two photons colliding, is given [24][25] by

$$\sigma_{\gamma\gamma} = \frac{\pi}{2} r_e^2 (1-\beta^2) \left[-2\beta(2-\beta^2) + (3-\beta^4) \ln \frac{1+\beta}{1-\beta} \right], \tag{1}$$

with $\beta = \sqrt{1-1/s}$ and $s = E_{\gamma_1} E_{\gamma_2} (1-\cos\Phi_B)/(2m_e^2 c^4)$, where the E_{γ_i} are the photon energies, c is the speed of light, m_e is the rest mass of the electron, Φ_B is the angle under which collision occurs and r_e is the classical electron radius. For reason of energy conservation, the BW process has the strict threshold of

$$E_{\gamma_1} E_{\gamma_2} \geq \frac{m_e^2 c^4}{\frac{1}{2}(1-\cos\Phi_B)}. \tag{2}$$

The denominator in the right hand side of equation (2) is caused by the centre-of-mass (CoM) movement of the photon pair and the electron–positron pair in case the two photons do not perfectly counter-propagate ($\Phi_B \neq 180°$). In such a case, the CoM movement of the photons has to be carried over to the electron–positron pair. In the CoM frame the two created particles have the momenta

$$p'_{e,p} = \pm \sqrt{\frac{1}{4}\left[\left(\frac{E_{Total}}{c}\right)^2 - (\vec{p}_{Total})^2\right] - (m_e c)^2}, \tag{3}$$