Journal of Computational Physics 268 (2014) 355–362

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Leap frog integrator modifications in highly collisional particle-in-cell codes

N. Hanzlikova *, M.M. Turner

School of Physical Sciences and National Centre for Plasma Science and Technology, Dublin City University, Dublin, Ireland

ARTICLE INFO

Article history:
Received 27 March 2013
Received in revised form 6 December 2013
Accepted 10 March 2014
Available online 20 March 2014

Keywords:
Particle-in-cell
Monte Carlo
Collisions
Leap frog
Integration

ABSTRACT

Leap frog integration method is a standard, simple, fast, and accurate way to implement velocity and position integration in particle-in-cell codes. Due to the direct solution of kinetics of particles in phase space central to the particle-in-cell procedure, important information can be obtained on particle velocity distributions, and consequently on transport and heating processes. This approach is commonly associated with physical situations where collisional effects are weak, but can also be profitably applied in some highly collisional cases, such as occur in semiconductor devices and gaseous discharges at atmospheric pressure. In this paper, we show that the implementation of the leap frog integration method in these circumstances can violate some of the assumptions central to the accuracy of this scheme. Indeed, without adaptation, the method gives incorrect results. We show here how the method must be modified to deal correctly with highly collisional cases.

1. Introduction

Particle-in-cell (PIC) codes have been a popular method of modelling plasmas since the 1960s [1–4]. While their computational cost may make complicated plasma chamber geometries and complex chemistries prohibitive, they provide a direct simulation vehicle for study of plasma kinetics. By tracking samples of the particle phase space, the resultant discrete superparticles allow for direct solutions of the motions of the particles, while the interpolation of the charges onto a grid solves the potential for the system in a much more efficient fashion than direct particle–particle interactions.

Collisions began being introduced into PIC codes a decade after [5–8], mostly through the adoption of the Monte Carlo procedure. The resultant codes usually relied on low collision probabilities, consistent with low pressure plasmas, with the assumption that the collision frequency is small compared to the plasma frequency, and collisions being resolved at the end of each timestep. A modification can be introduced for higher pressure cases, where a modified collision probability is calculated, but this presents a collision limit of one collision per particle per timestep [9]. In the case of high collision rates, an alternative is provided in texts such as Hockney's [10,11], which still remain the definitive literature on the subject. In this approach, a particle subdivided into collision times leading up to the timestep boundary may undergo several collisions between each solution of the field equations. The global timestep Δt is then the interval between solutions of the field equation, while particles advances across these intervals a sequence of intermediate steps punctuated by collisions.

While attractive in its simplicity, in this work we show how subdivision method results in inaccurate estimation of the velocity and position integration over time. We present a modification to the procedure outlined in Hockney's work [10,11],

* Corresponding author.
E-mail addresses: nina.hanzlikova2@mail.dcu.ie (N. Hanzlikova), miles.turner@dcu.ie (M.M. Turner).

http://dx.doi.org/10.1016/j.jcp.2014.03.018
0021-9991/

which revises the leap frog integration method to allow for subdivision into collisional sub-timesteps without requiring alternative integration procedures or timestep size reductions previously found to be necessary [12], and so reducing the computational runtime for the simulation. In Sections 2.1 and 2.2 we outline the general procedure for implementing the equations of motion integration and collision calculation, as given in the literature [11,13]. In Section 2.3 we then outline the modification implemented in our integrator to allow for sub-timestepping, of the particles during collisions. Section 3 presents some simulation results from our implementation, examining the integration of a single particle and outlining the behaviour of our solver in the sub-timestepping limit compared to the unmodified version and the probability implementation with no sub-timesteps. An overview summary is given in Section 4.

2. Highly collisional integrator outline

2.1. Summary of the leap frog integrator

As discussed in Section 1, PIC codes consist of iterative solutions of kinetics of superparticles in phase space, coupled with the solution of the Poisson equation, to determine the potentials in the cell grid system. In this article we are mostly interested in the former part of the procedure. A simple way of implementing the velocity and position integration is using the leap frog method [11,13]

$$-\frac{q}{m}\nabla\phi_t = \frac{v_{t+\Delta t/2} - v_{t-\Delta t/2}}{\Delta t} \tag{1}$$

$$v_{t+\Delta t/2} = \frac{x_{t+\Delta t} - x_t}{\Delta t} \tag{2}$$

This straightforward method introduces a half timestep phase shift between the velocity value and the position value, effectively making the final velocity calculated and stored at the end of the timestep correspond to time $t + \Delta t/2$ or the average velocity for constant acceleration during the timestep. This is illustrated graphically in Fig. 1(a), where it can be seen that the evaluation of the position and force calculation take place on the timestep boundary while the velocity is evaluated on the mid-step. Of particular convenience is that all the position, velocity and acceleration values can be stored as single variables without resorting to excessively short timesteps to conserve the accuracy of the approximation, as would be necessary if the velocity was calculated in-step with the position, and thus require the calculation of average velocity for each timestep to replicate the accuracy of the leap frog method.

2.2. High frequency Monte Carlo collisions

The implementation of Monte Carlo collisions varies depending on the frequency of collisions as well as the type of collisions taking place. For collision cases of collision frequency up to once per timestep, the most frequently implemented technique revolves around decoupling the collision from the velocity and position integration. This is achieved by carrying out the particle advancement as in the case of no collisions and applying the collision process at the end of timestep. In this case a pseudo-random uniform number is compared to the probability of collision for the particle, P, for the timestep. The collision probability P can be calculated from the mean free path $\lambda(v)$ as given by Birdsall [14]

$$P = 1 - \exp\left(\frac{-v\Delta t}{\lambda}\right) = 1 - \exp(-\nu_c \Delta t), \quad \nu_c = \frac{v}{\lambda} \tag{3}$$

where ν_c is the collision frequency for the particle velocity. Further selectivity of collision processes can be achieved by having a probability for each different type of collision being simulated and comparing the uniform number between these to select the appropriate collision process.

This procedure however limits the simulation to a maximum of single collision within a timestep. In many problems it is desirable to have a higher collision rate, which requires breaking down of the timestep into smaller segments. This can be implemented with the probability approach by comparing a uniform pseudo-random number to the collision probability at constant sub-timestep intervals, with the Leap Frog integration taking place between each comparison. Alternatively a non-constant sub-timestep interval can be generated for each particle, giving the time until collision, as given by

$$\delta t = -\frac{\ln R}{\nu_c} \tag{4}$$

where R is the uniform pseudo-random number. In this approach the sub-timestep and timestep boundaries no longer have to align giving each particle an extra piece of data associated with it between each timestep. However this is a much more direct way of simulating collisions and at high collision frequencies can be more accurate. These techniques are discussed in further detail in Chapter 10 of Hockney et al. [11] or in the accompanying paper [10]. The schematic of the subdivision of the leap frog integration over the timestep is shown in Fig. 1(b). It is this additional timestep division that we find of interest in our work.