Kinetic Plasma Simulation: Particle In Cell Method

Giovanni Lapenta Center for mathematical Plasma Astrophysics, KU Leuven, Belgium

The classical or relativistic description of the natural world is based on describing the interaction of elements of matter via force fields. The example that will guide the discussion is that of a plasma composed of charged particles but the discussion would be similar and easily replicated for the case of gravitational forces. In the case of a plasma, the system is composed by charged particles (for example negative electrons and positive ions) interacting via electric and magnetic fields.

If we identify each particle with a label p and their charge with q_p, position with $\mathbf{x}_p$, position with $\mathbf{v}_p$, the force acting on the particles is the combination of the electric and magnetic (Lorentz) force:

$$\mathbf{F}_p = q_p\mathbf{E}(\mathbf{x}_p) + \mathbf{v}_p \times \mathbf{B}_p(\mathbf{x}_p) \qquad (1)$$

The force acting on the particles is computed from the electric and magnetic fields evaluated at the particle position.

The electric and magnetic fields are themselves created by the particles in the system and by additional sources outside the system (for example magnets around the plasma or external electrodes). The fields are computed by solving the Maxwell's equations:

$$\begin{aligned} \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} &= 0 & \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \end{aligned} \qquad (2)$$

1 Types of interacting systems

A key point in the derivation of the particle in cell method is the consideration of how the sources in the Maxwell's equations ought to be computed. In principle since the system is made of a collection of particles of infinitesimal size, the sources of the Maxwell's equations are distributions of contributions one for each particle.

Figure 1 summarises visually the situation. Let us consider a system made by a collection of particles, each carrying a charge situated in a box with the side of the Debye length, λ_D (the box is 3D but is depicted as 2D for convenience). We choose the Debye length because a basic property of plasmas is to shield the effects of localized charges over distances exceeding the Debye length. Of course the shielding is exponential and the effect is not totally cancelled over one Debye length, but such a length provides a conventional reasonable choice for the interaction range. The electric field in each point of the box is computed by the superposition of the contribution of each particle.

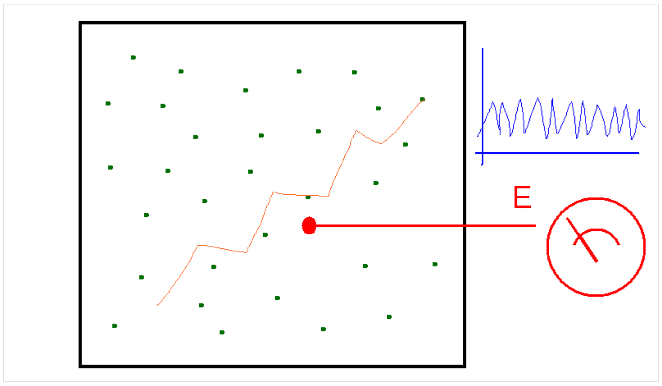

Figure 1: *A strongly coupled system.*

Let us conduct an ideal thought experiment based on using a experimental device able to detect the local electric field in one spatial position. Such an experimental device exists, but the determination of the local electric field remains a difficult but not impossible task. At any rate we try to conduct a thought experiment where in no step any law of physics is violated but where the difficulties of experimental work are eliminated.

If we consider the configuration in Fig. 1, we note that within the domain there are few particles and the measurement obtained by our fantastic electric field meter would be very jumpy. The particles in the box move constantly, interacting with each other and agitated by their thermal motion. As a particle passes by the detector, the measurement detects a jump up and when a particle moves away it detects a jump down. On average at any given time very few particles are near the detector and their specific positions are key in determining the value measured. The effect of a given particle on the electric field at the location of measurement decays very rapidly with the distance and only when the particle is nearby the effect is strong.

The same effect is detected by each of the particles in the system. The electric field each particle feels is a the sum of the contributions of all others but only when another particle passes by the electric field would register a jump: in common term this event is called a collision. The particle trajectories would then be affected by a series of close encounters registered as jumps in the trajectory.

The system described goes in the language of kinetic theory as a *strongly coupled system*, a system where the evolution is determined by the close encounters and by the relative configuration of any two pairs of particles. The condition just described is characterised by the presence of few particles in the box: $N_D = n\lambda_D^3$ is small.

The opposite situation is that of a weakly coupled system. The corresponding configuration is described in Fig. 2.

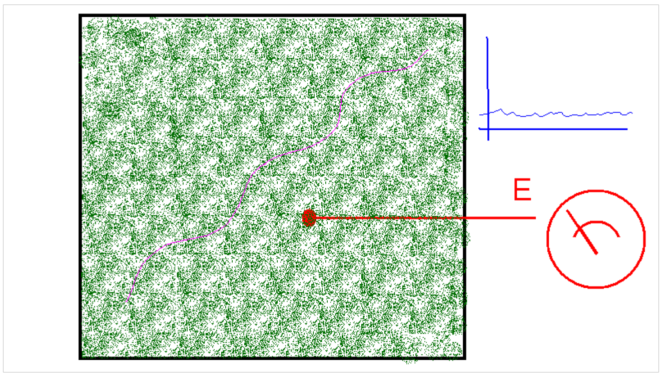

Figure 2: *A weakly coupled system.*

Now the system is characterised by being composed by an extremely large number of particles. In any given point, the number of particles contributing to the electric field is very large. Regardless of the particle motion, the field is given by the superposition of many contributions. As a consequence, by simple averaging of the effects of all the particles contributing to the measurement, the measurement is smooth and does not jump in time. Similarly the trajectory of a particle is at any time affected by a large number of other particles. The trajectory is smooth and witjout jumps. These systems are called *weakly coupled*. If in the strongly coupled system, the characteristic feature was the presence of a succession of collisions, in the weakly coupled system, the characteristic feature is the mean field produced by the superposition of contributions from a large number of particles.

2 Description of interacting systems

The discriminant factor in the previous discussion was the number of particles present in the box under consideration. If we choose the conventional box with side equal to the Debye length, the number of particles present is

$$N_D = n\lambda_D^3 \tag{3}$$

where n is the plasma density.

A system is considered weakly coupled when N_D is large and strongly coupled when N_D is small.

This concept can be further elaborated by considering the energies of the particles in the system. The particles in the box are distributed in a non-uniform, random way, but on average, the volume associated with each particle is simply the volume of the box, λ_D^3, divided by the number of particles in the box, N_D. This volume, $V_p = n^{-1}$, can be used to determine the average interparticle distance, $a = V_p^{1/3} \equiv n^{-1/3}$. This relation provides an average statistical distance. The particles are distributed randomly and their distances are also random, but on average the interparticle distance is a.

The electrostatic potential energy between two particles with separation a is

$$E_{pot} = \frac{q^2}{4\pi\epsilon_0 a} \tag{4}$$

where we have assumed equal charge q for the two particles. Conversely, from statistical physics, the kinetic energy of the particles can be computed to be of the order of

$$E_{th} = kT \tag{5}$$

where k is the Boltzmann constant.

A useful measure of the plasma coupling is given by the so-called *plasma coupling parameter*, Λ, defined as:

$$\Lambda = \frac{E_{th}}{E_{pot}} = \frac{4\pi\epsilon_0 akT}{q^2} \tag{6}$$

Recalling the definition of Debye length ($\lambda_D = (\epsilon_0 kT/ne^2)^{1/2}$) and the value of a obtained above, it follows that:

$$\Lambda = \frac{4\pi\epsilon_0 kT}{q^2 n^{1/3}} \equiv 4\pi N_D^{2/3} \tag{7}$$

The plasma parameter gives a new physical meaning to the number of particles per Debye cube. When many particles are present in the Debye cube the thermal energy far exceed the potential energy, making the trajectory of each particle little influenced by the interactions with the other particles: this is the condition outlined above for the weakly coupled systems. Conversely, when the coupling parameter is small, the potential energy dominates and the trajectories are strongly affected by the near neighbour interactions: this is the condition typical of strongly coupled systems.

3 Computer simulation

A computer simulation of a system of interacting particles can be conducted in principle by simply following each particle in the system. The so-called *particle-particle (PP)* approach describes the motion of N particles by evolving the equations of Newton for each of the N particles taking as a force acting on the particle the combined effect of all the other particles in the system.

The evolution is discretized in many temporal steps Δt, each chosen so that the particles move only a small distance, and after each move the force is recomputed and a new move is made for all the particles. If we identify the particle position and velocity as, respectively, $\mathbf{x}_p$ and $\mathbf{v}_p$, the equations of motion can be written as:

$$\begin{aligned} \mathbf{x}_p^{new} &= \mathbf{x}_p^{old} + \Delta t \mathbf{v}_p^{old} \\ \mathbf{v}_p^{new} &= \mathbf{v}_p^{old} + \Delta t \mathbf{F}_p \end{aligned} \tag{8}$$

The main cost of the effort is the computation of the force which requires to sum over all the particles in the system,

$$\mathbf{F}_p = \sum_{p'} \mathbf{F}_{pp'} \tag{9}$$

where $F_{pp'}$ is the interaction force between two particles p and p'. For example in the case of the electrostatic force,

$$\mathbf{F}_{pp'} = \frac{q_p q_{p'}}{4\pi\epsilon_0 |\mathbf{x}_p - \mathbf{x}_{p'}|^2} \cdot \frac{\mathbf{x}_p - \mathbf{x}_{p'}}{|\mathbf{x}_p - \mathbf{x}_{p'}|} \tag{10}$$

where in practice all forces are computed with the old values of the particle positions available at a given time. Once the force is computed the new velocities can be computed. Then the new positions can be computed and the cycle can be repeated indefinitely.

For each particle, the number of terms to sum to compute the force is $N-1$, and considering that there are N particles, but that each pair needs to be computed only once, the total number of force computations is $N(N-1)/2$.

For strongly coupled systems, where the number of particles per Debye cube is small, the PP approach is feasible and forms the basis of the very successful molecular dynamics method used in condensed matter and in biomolecular studies. We refer the reader to a specific text on molecular dynamics to investigate the approach more in depth [FS02]. The approach is also used in the study of gravitational interactions, for example in the cosmological studies of the formation and distribution of galaxies. In that case, specifically the dark matter is studied with a PP approach. The PP approach can be made more efficient by using the *Barnes-Hut* or *tree algorithm* [BH86] that can reduce the cost (but not without loss of information) to $O(N \log N)$.

Even with the reduced cost of the tree algorithm, PP methods cannot be practical for weakly coupled systems where the number of particles is very large. As the number of particles increases, the cost scales quadratically (or as $N \log N$) and makes the computational effort unmanageable. In that case, one cannot simply describe every particles in the system and a method must be devised to reduce the description to just a statistical sample of the particles. This is the approach described in the next section.

4 Finite size particles

The key idea behind the simulation of weakly coupled systems is to use as building block of the model not single particles but rather collective clouds of them: each computational particle (referred to sometimes as superparticle) represents a group of particles and can be visualised as a small piece of phase space. The concept is visualized in Fig. 3.

The fundamental advantage of the finite-size particle approach is that the computational particles,

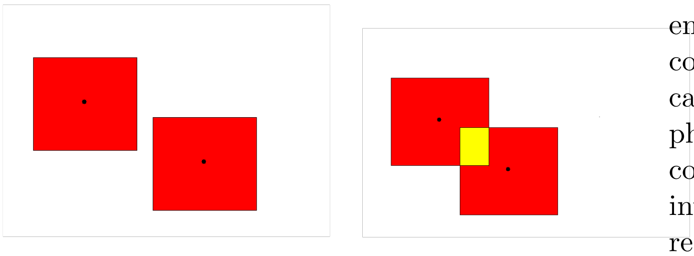

Figure 3: *Finite size particle.*

being of finite size, interact more weakly than point particles. When two point particles interact, for example via coulombian force, the repulsive or attractive force grows as the particles approach, reaching a singularity at zero separation. Finite size particles instead, behave as point particles until their respective surfaces start to overlap. Once overlap occurs the overlap area is neutralized, not contributing to the force between the particles. At zero distances when the particles fully overlap (assuming here that all particles have the same surface) the force become zero. Figure 4 shows the force between two spherical charged particles as a function of their distance. At large distances the force is identical to the Coulomb force, but as the distance becomes smaller than the particle diameter, the overlap occurs and the force starts to become weaker than the corresponding Coulomb force, until it becomes zero at zero separation.

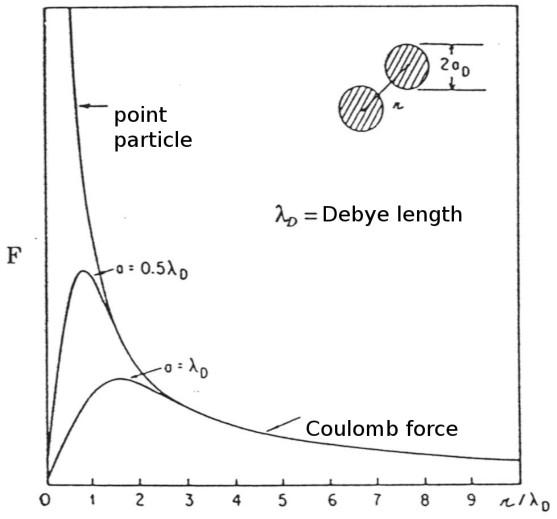

Figure 4: *Interactions between finite size particles. Reproduced from [Daw83].*

The use of finite-size computational particles allows to reduce the interaction among particles. Recalling the definition of plasma parameter, the use of finite-size particles results in reducing the potential energy for the same kinetic energy. The beneficial consequence is that the correct plasma parameter can be achieved by using fewer particles than in the physical system. The conclusion is that the correct coupling parameter is achieved by fewer particles interacting more weakly. The realistic condition is recovered.

5 Particle in Cell Method

The idea of the particle in cell (PIC, also referred to as particle-mesh, PM) method is summarised in Fig. 5. The system is represented by a small number of finite-size particles all interacting via the correct potential at distances beyond the overlap distance, but correcting the effect of fewer particles at small distances by the reduced interaction potential.

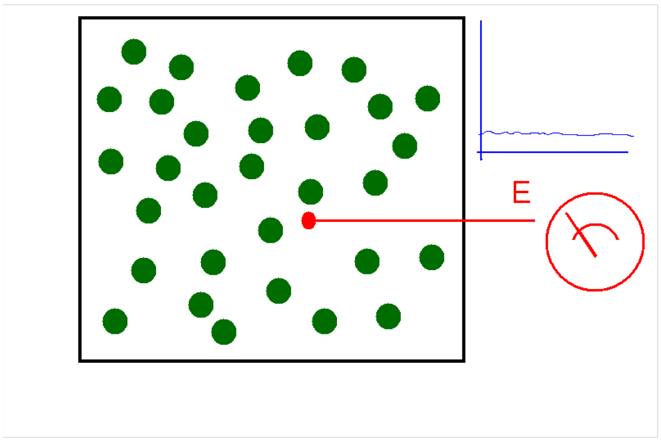

Figure 5: *A system of finite size particles.*

The end result is that the electric field fluctuations in the system are correctly smooth as they should be in a weakly coupled system. The reason now is not that at any time a very large number of particles average each other but rather that the effect of the few particles close to the measure point is weak.

Similarly the trajectory of particles are smooth as in the real system but not because each particle is surrounded by a very large number of near neighbours. Rather the few near neighbours produce weak interactions.

The collective effect is still correct as the long range interaction is unmodified and reproduces correctly the physical system.

6 Mathematical Derivation of the PIC method

We consider there the procedure for deriving the PIC method. Two classic textbooks [HE81, BL04] and a review paper [Daw83] report a heuristic derivation based on the physical properties of a plasma. We consider here a different approach aimed at making a clear mathematical link between the mathematical model of the plasma and its numerical solution. To make the derivation as easy as possible, while retaining all its fundamental steps we consider the following 1D electrostatic and classical plasma. The extension to 3D electromagnetic plasmas is no more difficult but clouded by the more complicated notation.

The phase space distribution function $f_s(x, v, t)$ for a given species s (electrons or ions), defined as the number density per unit element of the phase space (or the probability of finding a particle in a dx and dv around a certain phase space point (x, v)), is governed by the Vlasov equation:

$$\frac{\partial f_s}{\partial t} + v\frac{\partial f_s}{\partial x} + \frac{q_s E}{m_s}\frac{\partial f_s}{\partial v} = 0 \qquad (11)$$

where q_s and m_s are the charge and mass of the species, respectively.

The electric field in the electrostatic limit is described by the Poisson's equation for the scalar potential:

$$\epsilon_0 \frac{\partial^2 \varphi}{\partial x^2} = -\rho \qquad (12)$$

where the net charge density is computed from the distribution functions as:

$$\rho(x, t) = \sum_s q_s \int f_s(x, v, t) dv \qquad (13)$$

6.1 Numerical Approach

The PIC method can be regarded as a *finite element approach* but with finite elements that are themselves moving and overlapping. The mathematical formulation of the PIC method is obtained by assuming that the distribution function of each species is given by the superposition of several elements (called computational particles or superparticles):

$$f_s(x, v, t) = \sum_p f_p(x, v, t) \qquad (14)$$

Each element represents a large number of physical particles that are near each other in the phase space. For this reason, the choice of the elements is made in order to be at the same time physically meaningful (i.e. to represent a bunch of particles near each other) and mathematically convenient (i.e. it allows the derivation of a manageable set of equations).

The PIC method is based upon assigning to each computational particle a specific functional form for its distribution, a functional form with a number of free parameters whose time evolution will determine the numerical solution of the Vlasov equation. The choice is usually made to have two free parameters in the functional shape for each spatial dimension. The free parameters will acquire the physical meaning of position and velocity of the computational particle. The functional dependence is further assumed to be the tensor product of the shape in each direction of the phase space:

$$f_p(x, v, t) = N_p S_x(x - x_p(t)) S_v(v - v_p(t)) \qquad (15)$$

where S_x and S_v are the *shape functions* for the computational particles and N_p is the number of physical particles that are present in the element of phase space represented by the computational particle.

A number of properties of the shape functions come from their definition:

1. The support of the shape functions is compact, to describe a small portion of phase space, (i.e. it is zero outside a small range).

2. Their integral is unitary:

   $$\int_{-\infty}^{\infty} S_\xi(\xi - \xi_p) d\xi = 1 \qquad (16)$$

   where ξ stands for any coordinate of phase space.

3. While not strictly necessary, Occam's razor suggests to choose symmetric shapes:

   $$S_\xi(\xi - \xi_p) = S_\xi(\xi_p - \xi) \qquad (17)$$

While these definitions still leave very broad freedom in choosing the shape functions, traditionally the choices actually used in practice are very few.

6.2 Selection of the particle shape

The standard PIC method is essentially determined by the choice of S_v, the shape in the velocity direction as a Dirac's delta:

$$S_v(v - v_p) = \delta(v - v_p) \qquad (18)$$

This choice has the fundamental advantage that if all particles within the element of phase space described by one computational particle have the same speed, they remain closer in phase space during the subsequent evolution.

The original PIC methods developed in the 50's were based on using a Dirac's delta also as the shape function in space. But now for the spatial shape functions, all commonly used PIC methods are based on the use of the so-called b-splines. The b-spline functions are a series of consecutively higher order functions obtained from each other by integration. The first b-spline is the flat-top function $b_0(\xi)$ defined as:

$$b_0(\xi) = \begin{cases} 1 & if \ |\xi| < 1/2 \\ 0 & otherwise \end{cases} \qquad (19)$$

The subsequent b-splines, b_l, are obtained by successive integration via the following generating formula:

$$b_l(\xi) = \int_{-\infty}^{\infty} d\xi' b_0(\xi - \xi') b_{l-1}(\xi') \qquad (20)$$

Figure 6 shows the first three b-splines.

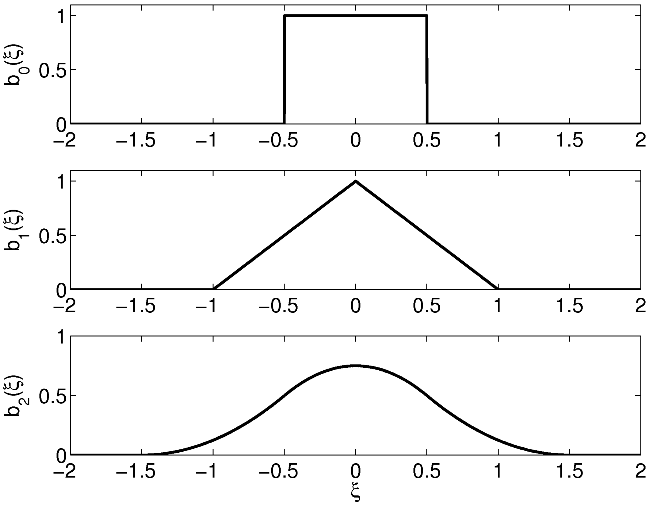

Figure 6: *First three b-spline functions.*

Based on the b-splines, the spatial shape function of PIC methods is chosen as:

$$S_x(x - x_p) = \frac{1}{\Delta_p} b_l \left(\frac{x - x_p}{\Delta_p} \right) \qquad (21)$$

where Δ_p is the scale-length of the support of the computational particles (i.e. its size). A few PIC codes use splines of order 1 but the vast majority uses b-splines of order 0, a choice referred to as cloud in cell because the particle is a uniform square cloud in phase space with infinitesimal span in the velocity direction and a finite size in space.

7 Derivation of the equations of motion

To derive the evolution equations for the free parameters x_p and v_p, we require that the first moments of the Vlasov equation to be exactly satisfied by the functional forms chosen for the elements. This procedure require some explanations:

1. The Vlasov equation is formally linear in f_s and the equation satisfied by each element is still the same Vlasov equation. The linear superposition of the elements gives the total distribution function and if each element satisfies the Vlasov equation, the superposition does too. A caveat, the electric field really depends on f_s making the Vlasov equation non-linear. As a consequence the electric field used in each Vlasov equation for each element must be the total electric field due to all elements, the same entering the complete Vlasov equation for f_s:

$$\frac{\partial f_p}{\partial t} + v \frac{\partial f_p}{\partial x} + \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} = 0 \qquad (22)$$

2. The arbitrary functional form chosen for the elements does not satisfy exactly the Vlasov equation. The usual procedure of the finite element method is to require that the moments of the equations be satisfied.

We indicate the integration over the spatial and velocity domain by the symbol $< \ldots > \equiv \int dx \int dv$.

7.1 Moment 0

The zeroth order moment ($< Vlasov >$) gives:

$$\frac{\partial < f_p >}{\partial t} + \left\langle v \frac{\partial f_p}{\partial x} \right\rangle + \left\langle \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \qquad (23)$$

where we used the interchangeability of the integration in $dxdv$ and of the derivation over time. The second and third term are zero, as:

$$\int \frac{\partial f_p}{\partial x} dx = f_p(x = +\infty) - f_p(x = -\infty) = 0$$

where the last equality follows from the compact support of f_p, assumed in the definition of the elements. A similar calculation holds for the term with the derivative over v. Recalling that $< f_p > = N_p$, it follows:

$$\frac{dN_p}{dt} = 0 \qquad (24)$$

The application of the first zeroth order moment leads to the establishment of the conservation of the number of physical particles per computational particle.

7.2 Moment 1_x

The application of the first order moment in x, ($< x \cdot Vlasov >$) gives:

$$\frac{\partial < f_p x >}{\partial t} + \left\langle vx \frac{\partial f_p}{\partial x} \right\rangle + \left\langle x \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \quad (25)$$

The last term is still zero by virtue of integration over v, the other terms, instead, are new. The first term is:

$$< f_p x >= N_p \int S_v(v - v_p) dv \int x S(x - x_p) dx$$

where the first integral is 1 by definition of S_v as a function of unitary integral and the second expresses the first order moment of S_x. Recalling the assumption of symmetry of S_x, that moment equals x_p:

$$< f_p x >= N_p x_p$$

The third term requires the integration of:

$$\int v dv \int x \frac{\partial f_p}{\partial x} dx =$$

$$\int v \left[f_p(x = +\infty) - f_p(x = -\infty) \right] x dv - \int v f dx dv =$$

$$- < f_p v >$$

where integration by part has been used. The integral can be computed as above, reversing the roles of x and v:

$$< f_p v >= N_p \int v S_v(v - v_p) dv \int S(x - x_p) dx = N_p v_p$$

using the parity of S_v. The end result of applying the first order moment in x is:

$$\frac{dx_p}{dt} = v_p \quad (26)$$

7.3 Moment 1_v

The application of the first order moment in v, ($< v \cdot Vlasov >$) gives:

$$\frac{\partial < f_p v >}{\partial t} + \left\langle v^2 \frac{\partial f_p}{\partial x} \right\rangle + \left\langle v \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \quad (27)$$

The second term is still zero by virtue of integration over x, as in the case of the zeroth order moment. The first term has already been computed above. The remaining term must be computed:

$$\int \frac{q_s E}{m_s} dx \int v \frac{\partial f_p}{\partial v} dv = - \int \frac{q_s E}{m_s} dx \int f_s dv = \left\langle \frac{q_s E}{m_s} f_s \right\rangle$$

using again integration by part and the finite support of the elements.

The remaining integral defines a new important quantity, the average electric field acting on a computational particle, E_p:

$$\left\langle \frac{q_s E}{m_s} f_s \right\rangle = -N_p \frac{q_s}{m_s} E_p$$

where the electric field on a computational particle is:

$$E_p = \int S_v(v - v_p) dv \int S_x(x - x_p) E(x) dx \quad (28)$$

Recalling the property of S_v, the formula for E_p simplifies to:

$$E_p = \int S_x(x - x_p) E(x) dx \quad (29)$$

The first order moment in v gives the final equation:

$$\frac{dv_p}{dt} = \frac{q_s}{m_s} E_p \quad (30)$$

7.4 Equations of motion for the computational particles

The equations above give the following complete set of evolution equations for the parameters defining the functional dependence of the distribution within each element:

$$\begin{aligned} \frac{dN_p}{dt} &= 0 \\ \frac{dx_p}{dt} &= v_p \\ \frac{dv_p}{dt} &= \frac{q_s}{m_s} E_p \end{aligned} \quad (31)$$

It is a crucial advantage of the PIC method that its evolution equations resemble the same Newton equation as followed by the regular physical particles. The key difference is that the field is computed as the average over the particles based on the definition of E_p.

Naturally, the electric field is itself given by Maxwell's equations which in turn need the charge density (and for complete models also the current density). The particle in cell approach described above provides immediately the charge density as the integral over the velocity variable of the distribution function:

$$\rho_s(x, t) = q_s \sum_p \int f_p(x, v, t) dv \quad (32)$$

Using the functional form for the distribution function of each computational element, the charge density becomes:

$$\rho_s(x,t) = \sum_p q_s N_p S_x(x-x_p) \tag{33}$$

The set of equations above provide a closed description for the Vlasov equation. Once accompanied by an algorithm to solve Maxwell's equations the full Vlasov-Maxwell system can be solved.

8 Field Equations

The solution of the field equations can be done with a wide variety of methods. The majority of the existing PIC methods relies on finite difference or finite volume, a choice we follow here to provide an example of the interfacing with the numerical solution of the Poisson and Vlasov equations.

Assuming the finite volume approach, a grid of equal cells of size Δx is introduced with cell centres x_i and cell vertices $x_{i+1/2}$. The scalar potential is discretized by introducing the cell-averaged values φ_i. The discrete form of the field equation is obtained by replacing the Laplacian operator (i.e. the simple second derivative in 1D) with a corresponding discretized operator.

In the simplest form, the Poisson's equation can be discretized in 1D using the classic three point formula:

$$\epsilon_0 \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{\Delta x^2} = -\rho_i \tag{34}$$

where the densities ρ_i are similarly defined as average over the cells:

$$\rho_i = \frac{1}{x_{i+1/2} - x_{i-1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x)dx \tag{35}$$

A most convenient formulation of the density averaged over each cell can be obtained recalling the definition of the b-spline of order 0

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x)dx = \int_{-\infty}^{\infty} b_0\left(\frac{x-x_i}{\Delta x}\right)\rho(x)dx \tag{36}$$

and recalling the expression of the density:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x)dx = \sum_p \int_{-\infty}^{\infty} b_0\left(\frac{x-x_i}{\Delta x}\right) S(x-x_p)dx \tag{37}$$

The standard nomenclature of the PIC method defines the *interpolation function* as:

$$W(x_i - x_p) = \int S_x(x-x_p) b_0\left(\frac{x-x_i}{\Delta x}\right) \tag{38}$$

It is crucial to remember the distinction between the shape function and the interpolation function. The interpolation function is the convolution of the shape function with the top hat function of span equal to the cell. The usefulness of the interpolation functions is that they allow a direct computation of the cell density without the need for integration. Defining the average cell density as, $\rho_i = \int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x)dx/\Delta x$, it follows that:

$$\rho_i = \sum_p \frac{q_p}{\Delta x} W(x_i - x_p) \tag{39}$$

where $q_p = q_s N_p$.

From the definition of the shape functions based on the b-spline of order l, it follows that if the shape function $S_x = \frac{1}{\Delta_p} b_l\left(\frac{x-x_p}{\Delta_p}\right)$ a very simple expression can be derived when the particle size equals the cell size, $\Delta p = \Delta x$:

$$W(x_i - x_p) = b_{l+1}\left(\frac{x_i - x_p}{\Delta_p}\right) \tag{40}$$

that follows trivially from the generating definition of the b-splines.

The solution of the Poisson equation can be conducted with the Thomas algorithm given appropriate boundary conditions. Once the solution is obtained, the potential is known in each cell, but in the form of the discrete values of the cell-averaged potentials φ_i. To compute the fields acting on the particles, the field is needed in the continuum. A procedure is needed to reconstruct it.

First, the electric field is computed in the cell centres from the discrete potentials as:

$$E_i = -\frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} \tag{41}$$

where centred difference are used. Then the continuum electric field is reconstructed using the assumption that the field is constant in each cell and equal to its cell-averaged value

$$E(x) = \sum_i E_i b_0\left(\frac{x-x_i}{\Delta x}\right) \tag{42}$$

From the definition of E_p it follows that:

$$E_p = \sum_i E_i \int b_0\left(\frac{x-x_i}{\Delta x}\right) S_x(x-x_p) \tag{43}$$

and recalling the definition of interpolation function,

$$E_p = \sum_i E_i W(x_i - x_p) \tag{44}$$

9 Discretization of the equations of motion

The equations of motion derived in paragraph 1.3.4 are simple ordinary differential equations with the same form as the regular Newton equations. Of course, in the literature there are many algorithms to achieve the goal of solving the Newton equations. For the PIC algorithm a efficient choice is to use simple schemes: given the very large number of particles used (billions are now common in published works), the use of complex schemes may result in prohibitively long simulations. However, if more advanced schemes allow one to use large time steps, the additional cost per time step may be compensated by taking longer time steps.

The simplest algorithm and by far the most used in the so-called *leap-frog algorithm* based on staggering the time levels of the velocity and position by half time step: $x_p(t = n\Delta t) \equiv x_p^n$ and $v_p(t = (n+1/2)\Delta t) \equiv v_p^{n+1/2}$. The advancement of position from time level n to time level $n+1$ uses the velocity at mid-point $v_p^{n+1/2}$, and similarly the advancement of the velocity from time level $n-1/2$ to $n+1/2$ uses the mid point position x_p^n. This stepping of velocity over position and position over velocity recalled some of the early users of the children's game bearing also the name leap-frog (see Fig. 7).

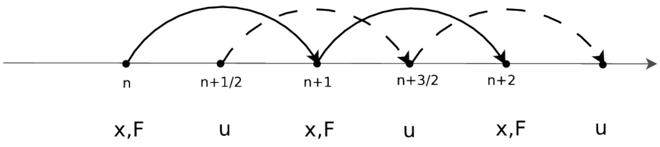

Figure 7: *Visual representation of the leap-frog algorithm.*

The scheme is summarised by:

$$\begin{aligned} x_p^{n+1} &= x_p^n + \Delta t v_p^{n+1/2} \\ v_p^{n+3/2} &= v_p^{n+1/2} + \Delta t \frac{q_s}{m_s} E_p(x_p^{n+1}) \end{aligned} \tag{45}$$

where E_p is computed solving the Poisson equation from the particle positions given at time level n.

Note that technically the leap-frog algorithm is second order accurate, when instead the regular explicit Euler-scheme is only first order. Nevertheless, the two differ in practice only for the fact that the velocity is staggered by half time step. This staggering is achieved by moving the initial velocity of the first time cycle by half a time step using an explicit method:

$$v_p^{1/2} = v_p^0 + \Delta t \frac{q_s}{m_s} E_p(x_p^0)$$

10 Recapitulation

Collecting the steps gathered so far, the PIC algorithm is summarised by the series of operations depicted in Fig. 8.

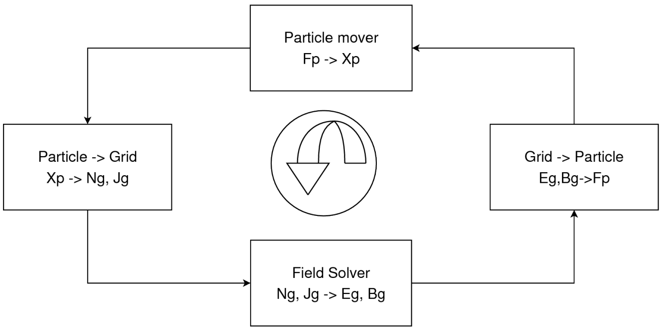

Figure 8: *Summary of a computational cycle of the PIC method.*

10.1 Algorithm of the PIC method, electrostatic case in 1D

i The plasma is described by a number of computational particles having position x_p, velocity v_p and each representing a fixed number N_p of physical particles.

ii The equations of motion for the particles are advanced by one time step using,

$$x_p^{n+1} = x_p^n + \Delta t v_p^{n+1/2}$$

$$v_p^{n+3/2} = v_p^{n+1/2} + \Delta t \frac{q_s}{m_s} E_p^{n+1}$$

using the particle electric field from the previous time step.

iii The charge densities are computed in each cell using:

$$\rho_i = \sum_p \frac{q_p}{\Delta x} W(x_i - x_p)$$

iv The Poisson equation is solved:

$$\epsilon_0 \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{\Delta x^2} = -\rho_i$$

and the electric field E_i in each cell is computed:

$$E_i = -\frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x}$$

v From the field known in the cells, the field acting on the particles is computed as

$$E_p^{n+1} = \sum_i E_i W(x_i - x_p^{n+1})$$

which is used in the next cycle

vi The cycle restarts.

The algorithm above is implemented in the MATLAB code provided (see Lapenta's web site). The b-spline of order 0 is used for the shape functions and consequently of order 1 for the interpolation function.

References

[BH86] J. Barnes and P. Hut. A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324:4446–449, 1986.

[BL04] C.K. Birdsall and A.B. Langdon. *Plasma Physics Via Computer Simulation*. CRC Press, Boca Raton, 2004.

[Daw83] J. M. Dawson. Particle simulation of plasmas. *Rev. Mod. Phys.*, 55:403–447, 1983.

[FS02] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, San Diego, 2002.

[HE81] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York, 1981.

[Lap12] Giovanni Lapenta. Particle simulations of space weather. *Journal of Computational Physics*, 231(3):795–821, 2012.

[LBR06] G. Lapenta, J. U. Brackbill, and P. Ricci. Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas. *Physics of Plasmas*, 13:5904, May 2006.