

Particle In Cell Methods

With Application to Simulations in Space
Weather

Giovanni Lapenta

*Centrum voor Plasma Astrofysica
Katholieke Universiteit Leuven*

Contents

1	Physical Derivation of the PIC method	<i>page</i> 1
1.1	Types of interacting systems	2
1.2	Description of interacting systems	5
1.3	Computer simulation	6
1.4	Finite size particles	8
1.5	Particle in Cell Method	9
2	Mathematical Derivation of the PIC method	12
2.1	Numerical Approach	13
2.2	Selection of the particle shape	14
2.3	Derivation of the equations of motion	16
	2.3.1 Moment 0	17
	2.3.2 Moment 1_x	18
	2.3.3 Moment 1_v	19
	2.3.4 Equations of motion for the computational particles	20
2.4	Field Equations	21
2.5	Discretization of the equations of motion	23
2.6	Recapitulation	25
3	Practical Considerations in using the PIC method	28
3.1	Stability	28
3.2	Conservation Laws	29
3.3	Diagnostics	29

3.4	Normalisation and non-dimensional units	30
3.5	Initial Loading	32
3.6	Boundary Conditions	33
3.7	Exercises	33
4	Explicit Particle Simulations of Space Weather	35
4.1	Derivation of the Particle Method	37
4.1.1	Discretisation of the Distribution Function	39
4.1.2	Selection of the particle shapes	41
4.1.3	Derivation of the equations of motion	43
4.1.4	Equations of motion for the computational particles	46
4.2	Explicit Temporal Discretisation of the Particle Methods	48
4.2.1	Stability of Explicit Time Differencing	50
4.3	Space-Time Discretisation of the Field Equations	51
4.3.1	Space Resolution and Finite Grid Instability	55
4.3.2	Time Discretisation of Maxwell's equations	57
4.4	Recapitulation of the explicit particle methods	61
5	Implicit Particle Methods for Space Weather	63
5.1	Fully Implicit Particle Methods	67
5.2	Semi Implicit Methods	71
5.2.1	Field Solver	75
5.2.2	Particle mover	76
5.2.3	Stability of the semi-implicit methods	77
5.3	Adaptive Gridding	79
5.4	Practical use of particle simulation in space weather	82

Contents

v

5.4.1	Scaling of massively parallel implicit PIC	82
5.4.2	Computational Costs of Particle Modeling in Space Weather	83
<i>Appendix 1</i>	MATLAB program for 1D electrostatic PIC	88
<i>Bibliography</i>		90

1

Physical Derivation of the PIC method

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:

$$\mathbf{F}_p = q_p \mathbf{E}(\mathbf{x}_p) + \mathbf{v}_p \times \mathbf{B}_p(\mathbf{x}_p) \quad (1.1)$$

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:

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} & \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \cdot \mathbf{B} &= 0 & \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}
\end{aligned} \tag{1.2}$$

1.1 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 1.1 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.

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.

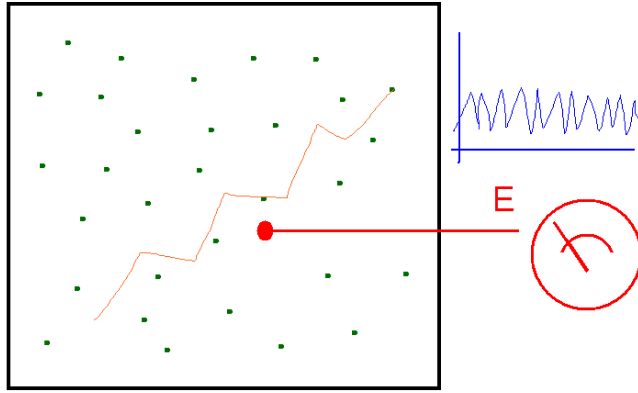


Fig. 1.1. A strongly coupled system.

If we consider the configuration in Fig. 1.1, 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. 1.2.

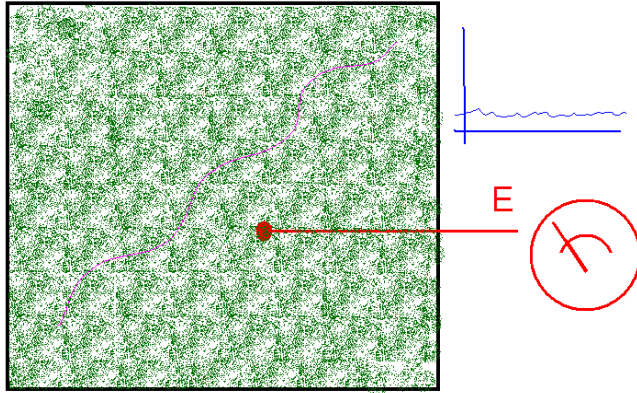


Fig. 1.2. A weakly coupled system.

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 par-

ticles. The trajectory is smooth and without 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.

1.2 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

$$N_D = n\lambda_D^3 \quad (1.3)$$

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

$$E_{pot} = \frac{q^2}{4\pi\epsilon_0 a} \quad (1.4)$$

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

$$E_{th} = kT \quad (1.5)$$

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:

$$\Lambda = \frac{E_{th}}{E_{pot}} = \frac{4\pi\epsilon_0 a kT}{q^2} \quad (1.6)$$

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:

$$\Lambda = \frac{4\pi\epsilon_0 kT}{q^2 n^{1/3}} \equiv 4\pi N_D^{2/3} \quad (1.7)$$

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.

1.3 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:

$$\begin{aligned}\mathbf{x}_p^{new} &= \mathbf{x}_p^{old} + \Delta t \mathbf{v}_p^{old} \\ \mathbf{v}_p^{new} &= \mathbf{v}_p^{old} + \Delta t \mathbf{F}_p\end{aligned}\tag{1.8}$$

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

$$\mathbf{F}_p = \sum_{p'} \mathbf{F}_{pp'}\tag{1.9}$$

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

$$\mathbf{F}_{pp'} = \frac{q_p q_{p'}}{4\pi\epsilon_0 |\mathbf{x}_p - \mathbf{x}_{p'}|^2} \cdot \frac{\mathbf{x}_p - \mathbf{x}_{p'}}{|\mathbf{x}_p - \mathbf{x}_{p'}|}\tag{1.10}$$

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.

1.4 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. 1.3.

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

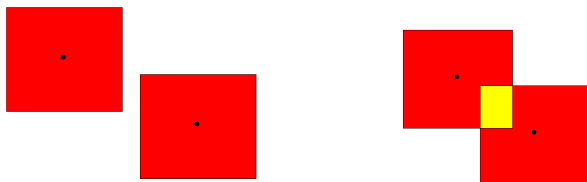


Fig. 1.3. Finite size particle.

cles fully overlap (assuming here that all particles have the same surface) the force become zero. Figure 1.4 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.

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.

1.5 Particle in Cell Method

The idea of the particle in cell (PIC, also referred to as particle-mesh, PM) method is summarised in Fig. 1.5. The system is represented by a small number of finite-size particles all interacting via the correct potential at distances beyond the

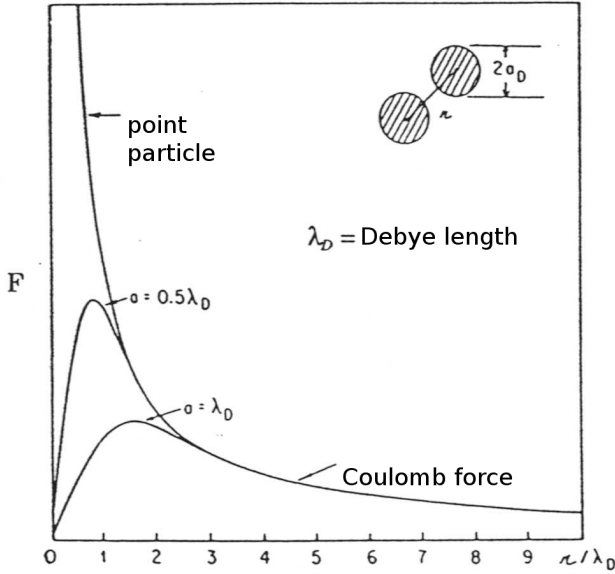


Fig. 1.4. Interactions between finite size particles.

overlap distance, but correcting the effect of fewer particles at small distances by the reduced interaction potential.

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.

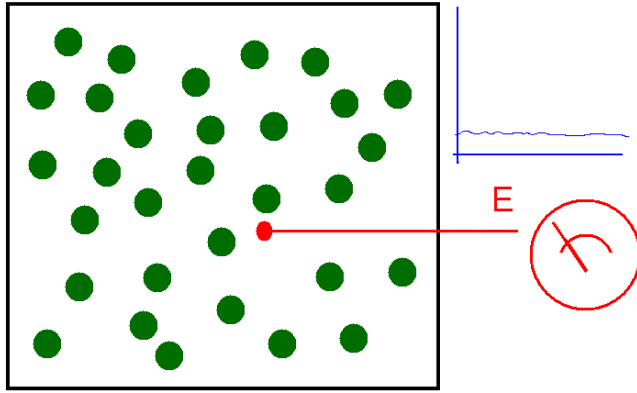


Fig. 1.5. A system of finite size particles.

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

2

Mathematical Derivation of the PIC method

We consider there the procedure for deriving the PIC method. Two classic textbooks [HE81, BL04a] 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:

$$\frac{\partial f_s}{\partial t} + v \frac{\partial f_s}{\partial x} + \frac{q_s E}{m_s} \frac{\partial f_s}{\partial v} = 0 \quad (2.1)$$

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:

$$\epsilon_0 \frac{\partial^2 \varphi}{\partial x^2} = -\rho \quad (2.2)$$

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

$$\rho(x, t) = \sum_s q_s \int f_s(x, v, t) dv \quad (2.3)$$

2.1 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):

$$f_s(x, v, t) = \sum_p f_p(x, v, t) \quad (2.4)$$

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:

$$f_p(x, v, t) = N_p S_x(x - x_p(t)) S_v(v - v_p(t)) \quad (2.5)$$

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:

- (i) 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).
- (ii) Their integral is unitary:

$$\int_{-\infty}^{\infty} S_\xi(\xi - \xi_p) d\xi = 1 \quad (2.6)$$

where ξ stands for any coordinate of phase space.

- (iii) While not strictly necessary, Occam's razor suggests to choose symmetric shapes:

$$S_\xi(\xi - \xi_p) = S_\xi(\xi_p - \xi) \quad (2.7)$$

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

2.2 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:

$$S_v(v - v_p) = \delta(v - v_p) \quad (2.8)$$

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:

$$b_0(\xi) = \begin{cases} 1 & \text{if } |\xi| < 1/2 \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

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

$$b_l(\xi) = \int_{-\infty}^{\infty} d\xi' b_0(\xi - \xi') b_{l-1}(\xi') \quad (2.10)$$

Figure 2.1 shows the first three b-splines.

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

$$S_x(x - x_p) = \frac{1}{\Delta_p} b_l \left(\frac{x - x_p}{\Delta_p} \right) \quad (2.11)$$

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.

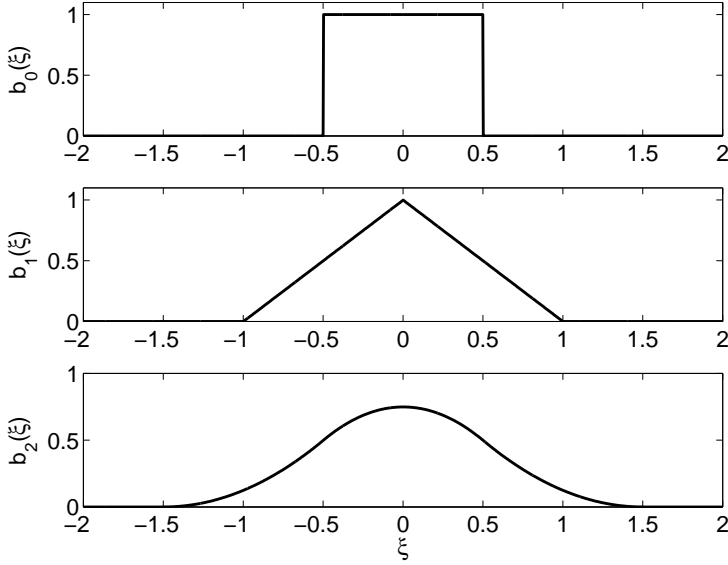


Fig. 2.1. First three b-spline functions.

2.3 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:

- (i) 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 :

$$\frac{\partial f_p}{\partial t} + v \frac{\partial f_p}{\partial x} + \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} = 0 \quad (2.12)$$

- (ii) 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 $\langle \dots \rangle \equiv \int dx \int dv$.

2.3.1 Moment 0

The zeroth order moment ($\langle Vlasov \rangle$) gives:

$$\frac{\partial \langle f_p \rangle}{\partial t} + \left\langle v \frac{\partial f_p}{\partial x} \right\rangle + \left\langle \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \quad (2.13)$$

where we used the interchangeability of the integration in $dx dv$ 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 $\langle f_p \rangle = N_p$, it follows:

$$\frac{dN_p}{dt} = 0 \quad (2.14)$$

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

2.3.2 Moment 1_x

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

$$\frac{\partial \langle f_p x \rangle}{\partial t} + \left\langle vx \frac{\partial f_p}{\partial x} \right\rangle + \left\langle x \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \quad (2.15)$$

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

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

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 :

$$\langle f_p x \rangle = N_p x_p$$

The third term requires the integration of:

$$\begin{aligned} & \int v dv \int x \frac{\partial f_p}{\partial x} dx = \\ & \int v [f_p(x = +\infty) - f_p(x = -\infty)] x dv - \int v f dx dv = \\ & - \langle f_p v \rangle \end{aligned}$$

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

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

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

$$\frac{dx_p}{dt} = v_p \quad (2.16)$$

2.3.3 Moment 1_v

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

$$\frac{\partial \langle f_p v \rangle}{\partial t} + \left\langle v^2 \frac{\partial f_p}{\partial x} \right\rangle + \left\langle v \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} \right\rangle = 0 \quad (2.17)$$

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:

$$E_p = \int S_v(v - v_p) dv \int S_x(x - x_p) E(x) dx \quad (2.18)$$

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

$$E_p = \int S_x(x - x_p) E(x) dx \quad (2.19)$$

The first order moment in v gives the final equation:

$$\frac{dv_p}{dt} = \frac{q_s}{m_s} E_p \quad (2.20)$$

2.3.4 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:

$$\begin{aligned}\frac{dN_p}{dt} &= 0 \\ \frac{dx_p}{dt} &= v_p \\ \frac{dv_p}{dt} &= \frac{q_s}{m_s} E_p\end{aligned}\tag{2.21}$$

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:

$$\rho_s(x, t) = q_s \sum_p \int f_p(x, v, t) dv\tag{2.22}$$

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

$$\rho_s(x, t) = \sum_p q_s N_p S_x(x - x_p)\tag{2.23}$$

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.

2.4 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:

$$\epsilon_0 \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{\Delta x^2} = -\rho_i \quad (2.24)$$

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

$$\rho_i = \frac{1}{x_{i+1/2} - x_{i-1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x) dx \quad (2.25)$$

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

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x) dx = \int_{-\infty}^{\infty} b_0 \left(\frac{x - x_i}{\Delta x} \right) \rho(x) dx \quad (2.26)$$

and recalling the expression of the density:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x) dx = \sum_p \int_{-\infty}^{\infty} b_0 \left(\frac{x - x_i}{\Delta x} \right) S(x - x_p) dx \quad (2.27)$$

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

$$W(x_i - x_p) = \int S_x(x - x_p) b_0 \left(\frac{x - x_i}{\Delta x} \right) \quad (2.28)$$

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:

$$\rho_i = \sum_p \frac{q_p}{\Delta x} W(x_i - x_p) \quad (2.29)$$

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, $\Delta_p = \Delta x$:

$$W(x_i - x_p) = b_{l+1} \left(\frac{x_i - x_p}{\Delta_p} \right) \quad (2.30)$$

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:

$$E_i = -\frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} \quad (2.31)$$

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

$$E(x) = \sum_i E_i b_0 \left(\frac{x - x_i}{\Delta x} \right) \quad (2.32)$$

From the definition of E_p it follows that:

$$E_p = \sum_i E_i \int b_0 \left(\frac{x - x_i}{\Delta x} \right) S_x(x - x_p) \quad (2.33)$$

and recalling the definition of interpolation function,

$$E_p = \sum_i E_i W(x_i - x_p) \quad (2.34)$$

2.5 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. 2.2).

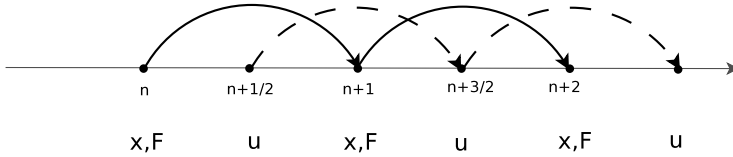


Fig. 2.2. Visual representation of the leap-frog algorithm.

The scheme is summarised by:

$$\begin{aligned} x_p^{n+1} &= x_p^n + \Delta t v_p^{n+1/2} \\ v_p^{n+3/2} &= v_p^{n+1/2} + \Delta t \frac{q_s}{m_s} E_p(x_p^{n+1}) \end{aligned} \tag{2.35}$$

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 veloc-

ity of the first time cycle by half a time step using an explicit method:

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

2.6 Recapitulation

Collecting the steps gathered so far, the PIC algorithm is summarised by the series of operations depicted in Fig. 2.3.

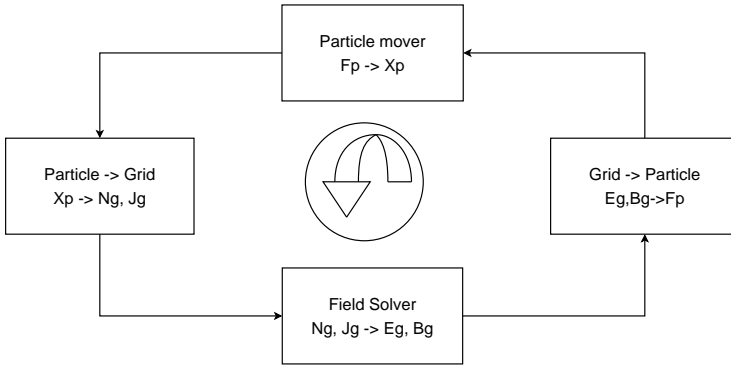


Fig. 2.3. Summary of a computational cycle of the PIC method.

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. The b-spline of order 0 is used for the shape func-

tions and consequently of order 1 for the interpolation function.

3

Practical Considerations in using the PIC method

We consider here some of the most immediate issues related to the use of PIC methods.

3.1 Stability

The first thing to consider in running a PIC code is a proper choice of the time step and of the grid spacing. The problem has been studied theoretically in depth and a precise understanding has been reached. We refer the reader to the two main textbooks on the topic for the derivations [BL04a, HE81]. Below we just summarize the results of the analysis that provide a straightforward prescription that must be followed in every occasion. The time step needs to resolve both light-wave propagation and Langmuir wave propagation:

$$c\Delta t < \Delta x \tag{3.1}$$

$$\omega_{pe}\Delta t < 2 \tag{3.2}$$

regardless to their relevance to the scale of interest. The grid spacing needs to resolve the electron Debye length to avoid the so-called finite grid instability due to the aliasing of different Fourier modes:

$$\Delta x < \varsigma\lambda_{De} \tag{3.3}$$

where ς is a constant of order 1 whose exact value depends on the choice of the interpolation and assignment functions. For the cloud in cell (CIC) choice, the value is $\varsigma \approx \pi$. In summary the standard PIC method need to resolve the finest scales everywhere. For this reason a more advanced approach called *implicit moment method* has been designed to remove this constraint, allowing the user to resolve just the scales of interest [LBR06a].

3.2 Conservation Laws

The PIC method described in the previous chapter conserves momentum to machine precision. However, energy is not preserved exactly. The fundamental reason is that in practice the PIC method uses many particles per cell: there are infinite particle configurations resulting in the same value of the quantities projected to the grid. This degree of freedom is what causes the finite grid instability and the lack of exact energy conservation.

For explicit PIC codes, based on the algorithm described above, the energy needs to be monitored and the time step and grid spacing need to be chosen sufficiently small as to conserve energy: satisfying the stability constraints is not enough. In practice usually the time step has to be less than 1/10 of its stability limit and the grid spacing less than 1/3 for energy to be conserved sufficiently. But the actual performance is problem dependent.

The implicit moment method solves this problem also, by enforcing a much better (but still not exact) energy conservation.

3.3 Diagnostics

The great advantage of the PIC method is that it provides quantities very similar to those provided in a actual plasma

30 *Practical Considerations in using the PIC method*

experiments: the user has both information on the distribution of the plasma particles and of the fields. Below we describe some of the most used diagnostics.

Typical *particle diagnostics* are:

- phase space plots (x_p vs v_p)
- more complex phase space density plots where the phase space is discretized in a grid of cells and the number of particles in each cell is counted to give the distribution function
- Velocity distribution functions (e.g. histogram of particle counts in energy bins)

Typical *field diagnostics* are:

- plots of the fields (e.g. E , φ , ρ , ...) versus space and/or time
- FFT modes of the field quantities to measure growth rates of specific modes.

Furthermore, information can be derived from integral quantities. Examples are: the *total kinetic energy*:

$$E_k = \frac{1}{2} \sum_p m_p v_p^2 \quad (3.4)$$

the *total momentum*:

$$P = \sum_p m_p v_p \quad (3.5)$$

the *total energy of the field*:

$$E_E = \Delta x \sum_c \frac{\epsilon_0 E^2}{2} \equiv \Delta x \sum_c \frac{\rho \varphi}{2} \quad (3.6)$$

3.4 Normalisation and non-dimensional units

It is traditional in theoretical physics to use non-dimensional units to conduct the derivations. This practice is even more

useful in computational physics as it can be used to make sure that the numbers remain close to order one, leading to a maximum accuracy of the finite-digits operations of computers.

Many methods to non-dimensionalize can be used. One example is suggested here. Let us consider space and time first. To normalise together space and time, one plasma frequency (e.g. that of the electrons can be used) and the speed of light are used:

$$t' = \omega_{pe} t$$

$$\ell' = \frac{\omega_{pe} \ell}{c}$$

resulting in times counted in periods of electron plasma waves and lengths in electron inertial lengths $d_e = c/\omega_{pe}$.

The charge (or mass) can be normalised in units of charge (mass) in the unit cube: $Q' = Q/n_0 e d_e^3$ ($M' = M/n_0 m_e d_e^3$). All other quantities can be similarly derived from the unit length, time, mass and charge of the normalised system.

In practice, the normalisation above corresponds in simplifying the expressions of the Vlasov-Maxwell system and of its discretization in the PIC method by choosing a number of quantities as 1. In particular, choosing the charge, mass and reference density of the electrons unitary, $e = 1$, $m_e = 1$, $n_e = 1$, leads to a unitary electron plasma frequency. If further, $\epsilon_0 = 1$ and $\mu_0 = 1$ (in SI or $c = 1$ in cgs), the other normalisations of velocity, charge and mass follow.

Note that once these fundamental quantities are chose as unitary, all others must follow correctly. For example, if $m_e = 1$, then $m_i = 1836$ for hydrogen and similar for all other derived quantities.

3.5 Initial Loading

The PIC algorithm described in the previous chapter provides a deterministic marching order where the positions and velocities of the computational particles and the fields in the cells are advanced according to precise rules. These rules require an initialisation based on an initial condition.

Given the initial state of the system a prescribed number of computational particles need to be generated so that they represent accurately the distribution function of the initial system. The operation of creating the initial particles is referred to as *particle loading*. Two fundamental approaches are common: the random start and the quiet start.

Random starts are obtained using MonteCarlo methods for distribution sampling. The simplest case is the use of the function RAND of Matlab (or similar in other languages) to generate a uniform distribution. Similarly simple is the generation of a Maxwellian distribution using the Matlab function RANDN. In some languages a similar function is not available but algorithms are easily available in the literature or on the web to sample numbers from any distribution function. Usually the initial distribution is the product of separate distributions in each direction of the six dimensional phase space.

Quiet starts are an attempt to reduce the noise and indeterminism of random starts. The phase space is covered by a pattern of computational particles. Their statistical weights (N_p) or their distribution in space (in a non-uniform pattern) is chosen to reproduce the required distribution. For example, a sinusoidal distribution can be reproduced by putting more particles in the peaks and fewer in the valleys. Rigorous mathematical techniques have been developed to do this. However, the use of quiet starts is not common. For three reasons. First, the approach is rather more complicated formally than the random start. Second, the use of a

regular array of particles can lead to correlation among the particles that have no equivalence in reality. Third, the noise is only reduced briefly, and later in time the discrete nature of the particle representation reasserts itself reintroducing a noise. Nevertheless, in some instances quiet starts can be considered.

3.6 Boundary Conditions

Two sets of boundary conditions are needed: for the fields and for the particles.

The field boundary conditions are the same as in any method solving the Maxwell equations. The particle boundary conditions can be chosen to best represent a physical system. Typical examples are: reflection at a boundary, periodic (where a particle exists one boundary and re-enters the opposite boundary with the same speed), removal to represent a wall.

Injection is also a possibility, to impose a prescribed inflow of plasma. Injection is a similar process to particle injection and it also can rely on deterministic methods (akin to quiet starts) or much more commonly on statistical methods (akin to random starts).

3.7 Exercises

As a demo session, we suggest the following exercises:

- (i) Start from the skeleton MATLAB code provided and add the following diagnostics:
 - History of kinetic and potential energy
 - History of the total momentum
 - History of the thermal energy (using the cov function of MATLAB)
 - Fourier modes at a given time
 - History of selected Fourier modes

34 *Practical Considerations in using the PIC method*

- Distribution of the particle velocities at a given time
 - Phase space scatter plot of the particles
 - Plot of the potential and electric field at a given time
 - Space-time plot of the potential and electric field
 - Phase-space density plot
- (ii) Consider a uniform maxwellian plasma and try to exceed, separately, $\omega_{pe}\Delta t < 2$ and $\lambda_{De}\Delta x < \pi$. What happens to the energy and the particle scatter plot in phase space?
- (iii) Consider the same plasma and add a initial sinusoidal perturbation of the particle position. What happens to the energy, and to the Fourier mode with the same wavelength as the initial perturbation? What happens to the phase space? What is the process developing?
- (iv) Consider next two beams with a small thermal spread and two opposite velocities. The systems undergoes the two-stream instability. Its dispersion relation is:

$$D(k, \omega) = 1 - \frac{\omega_{pe}^2}{2} \left[\frac{1}{(\omega - kv_0)^2} + \frac{1}{(\omega + kv_0)^2} \right]$$

- Design a simulation box with length equal to the wavelength of maximum growth ($k = k_M$). Run the simulation and consider the evolution of the Fourier modes. What happens to the mode supposed to be the fastest? And what to the Fourier mode with half the wavelength (i.e. $k = 2k_M$)?
- Double the simulation box. What happens if you excite the mode $k = k_M$, compared with the mode $k = k_M/2$?
- Let a simulation run for a long time, until the energy saturates. What happens to the velocity distribution as time progresses?

4

Explicit Particle Simulations of Space Weather

According to the definition used by the U.S. National Space Weather Strategic Plan [spa], space weather refers to conditions on the sun and in the solar wind, magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of space-borne and ground-based technological systems and can endanger human life or health. Adverse conditions in the space environment can cause disruption of satellite operations, communications, navigation, and electric power distribution grids, leading to a variety of socioeconomic losses. The topic of space weather is becoming ever more important for its societal impact and great national and international efforts are being devoted to it [BD07].

Simulating space weather means literally simulating the magnetized plasma that permeates the whole solar system. A daunting task. Daunting because plasma physics includes a wide variety of space and temporal scales. A plasma is made of a collection of ionised atoms (nuclei), and electrons interacting via electric and magnetic fields. Near the Sun gravity plays a role too. At the core of the space weather simulation issue stands the vast gap between the electron and ion scales. An electron is 1836 times lighter than its nucleus in an hydrogen atom. This simple physical fact creates a great challenge in modeling the small electron scales

embedded into solar system sized problems. For example, Fig. 4.1 shows the typical electron and ion scales observed in satellite crossing of the Earth magnetosphere (the magnetotail to be specific) [Kal98]. The scales are presented in the form of a hourglass with the top part presenting the macroscopic scales of evolution of the Earth magnetotail that during quiescent times can be 10,000 km thick but at interesting times becomes as thick as the ion inertial length, d_i or 1000 km. The ion scales are smaller and the electron scales much smaller, down to 100 m corresponding to typical electron Debye lengths. Similarly in time, the overall evolution of space weather events requires simulating hours, if not days, but the ion responses are on the scale of seconds to milliseconds and the electrons respond on scales of multiples of microseconds.

Simulating days of evolution of the plasma in the solar system with a time step of microseconds and with a resolution of 100 m is simply not possible and it will not be possible for a long time to come. The present paper reviews the efforts to overcome this problem and allow a simulation of space weather events at the particle level, retaining the full kinetic and electromagnetic treatment of the plasma.

Traditionally, the impossibility to resolve all scales has led to the use of fluid (MHD or more advanced models such as Hall-MHD and two-fluid) or hybrid methods where the electrons are described as a fluid while the ions are kinetic [Lip02]. However, such reduced models lose the ability to capture the dissipation processes at the electron scales. Important examples for space weather are the regions of large energy releases such as shocks and reconnection regions where small scale processes at the ion and electron particle scale are key in determining the production of high energy particles (such as the solar energetic particles in the cosmic rays [BD07]) and in allowing macroscopic changes in the magnetic field configuration (e.g. magnetic reconnection during solar flares

and coronal mass ejections [Asc05] or in the Earth environment during magnetic storms and substorms [Kal98]).

We review in the present chapter and the next the full kinetic models based on the particle in cell (PIC) simulation technique. Chapter 4 expands to the general case the derivation of the mathematical foundations of the PIC method presented above, in a manner that helps identify the strengths and weaknesses for the application to space weather. The focus next go on the explicit time discretisation of PIC methods and highlights the severe limitations of this approach when applied to space weather. Additional information is provided about the technical aspects of applying the explicit technique and developing codes based on it.

Chapter 5 introduces a different approach, that of implicit time differencing in PIC methods, an approach particularly suitable to space weather applications. The approach is not new but it is reviewed here especially in light of its more recent space weather applications. For this reason, we review in Section 6 the general idea of implicit PIC and provide in the subsequent sections its two possible implementations: fully implicit, requiring a solution of a set of non-linearly coupled equations and semi-implicit where the coupling is linearised and made more manageable. An introduction is given of the recent developments in fully implicit PIC. This area is in its infancy with just a few initial works done but it is perhaps the most promising area of research in this field and it has been included despite the lack of a strong previous literature. Section 8 reviews the much more mature field of semi-implicit PIC methods.

4.1 Derivation of the Particle Method

As a key step in the study of the more advanced applications of PIC methods, we lay first the mathematical groundwork for the formulation of the particle in cell method start-

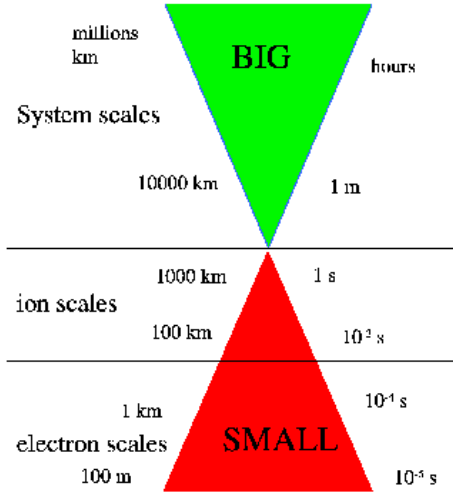


Fig. 4.1. Visual representation of the typical scales observed in the Earth environment (conditions common in the magnetotail).

ing from the mathematical continuous problem, the Vlasov-Maxwell system. The assumption is made in the derivations that classical physics applies, but the extension to special relativity is straight-forward. The classic textbooks [HE88, BL04b] report a heuristic derivation based on the physical properties of a plasma. Instead, we consider here a different approach aimed at making a clear mathematical link between the continuous Vlasov-Maxwell description of the plasma and its numerical solution.

The phase space distribution function $f_s(\mathbf{x}, \mathbf{v}, t)$ 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 $d\mathbf{x}$ and $d\mathbf{v}$ around a certain phase

space point (\mathbf{x}, \mathbf{v}) , is governed by the Vlasov equation:

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0 \quad (4.1)$$

where q_s and m_s are the charge and mass of the species, respectively. The derivatives are written in the standard vector notation in the 3-dimensional configuration space \mathbf{x} and 3-dimensional velocity space \mathbf{v} , forming together the classical phase space.

The electric and magnetic fields are given by the two curl Maxwell's equations,

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (4.2)$$

$$\nabla \times \mathbf{B} = \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \mathbf{j}$$

supplemented by the two divergence equations:

$$\epsilon_0 \nabla \cdot \mathbf{E} = \rho \quad (4.3)$$

$$\nabla \cdot \mathbf{B} = 0$$

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

$$\rho(\mathbf{x}, t) = \sum_s q_s \int_{\mathbb{V}} f_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \quad (4.4)$$

and the current density as:

$$\mathbf{j}(\mathbf{x}, t) = \sum_s q_s \int_{\mathbb{V}} \mathbf{v} f_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \quad (4.5)$$

where \mathbb{V} is the velocity space.

4.1.1 Discretisation of the Distribution Function

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):

$$f_s(\mathbf{x}, \mathbf{v}, t) = \sum_p f_p(\mathbf{x}, \mathbf{v}, t) \quad (4.6)$$

Each element represents a large number of physical particles that are near each other in 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. to allow 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:

$$f_p(\mathbf{x}, \mathbf{v}, t) = N_p S_x(\mathbf{x} - \mathbf{x}_p(t)) S_v(\mathbf{v} - \mathbf{v}_p(t)) \quad (4.7)$$

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:

- (i) 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).

- (ii) Their integral over the domain is unitary:

$$\int_{V_\xi} S_\xi(\xi - \xi_p) d\xi = 1 \quad (4.8)$$

where ξ stands for any coordinate or any velocity direction.

- (iii) While not strictly necessary, Occam's razor suggests to choose symmetric shapes:

$$S_\xi(\xi - \xi_p) = S_\xi(\xi_p - \xi) \quad (4.9)$$

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

4.1.2 Selection of the particle shapes

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

$$S_{\mathbf{v}}(\mathbf{v} - \mathbf{v}_p) = \delta(v_x - v_{xp})\delta(v_y - v_{yp})\delta(v_z - v_{zp}) \quad (4.10)$$

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:

$$b_0(\xi) = \begin{cases} 1 & \text{if } |\xi| < 1/2 \\ 0 & \text{otherwise} \end{cases} \quad (4.11)$$

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

$$b_l(\xi) = \int_{-\infty}^{\infty} d\xi' b_0(\xi - \xi') b_{l-1}(\xi') \quad (4.12)$$

Figure 4.2 shows the first three b-splines.

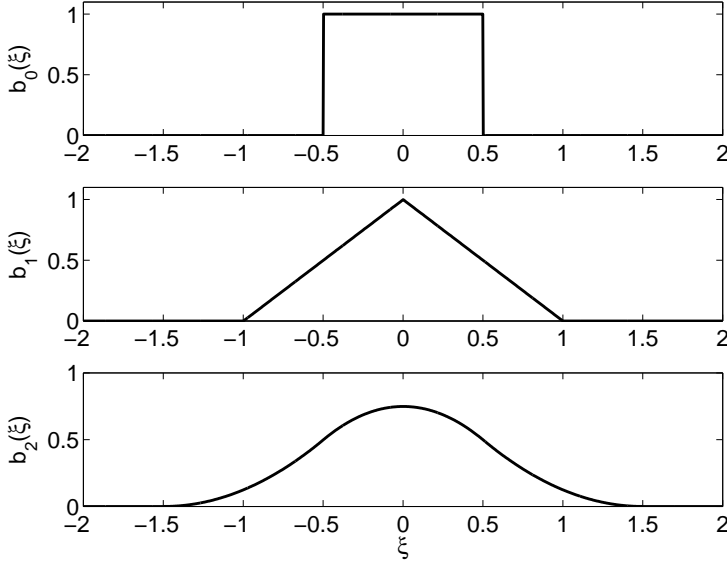


Fig. 4.2. First three b-spline functions.

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

$$S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) = \frac{1}{\Delta x_p \Delta y_p \Delta z_p} b_l \left(\frac{x - x_p}{\Delta x_p} \right) b_l \left(\frac{y - y_p}{\Delta y_p} \right) b_l \left(\frac{z - z_p}{\Delta z_p} \right) \quad (4.13)$$

where Δx_p , Δy_p and Δz_p are the lengths of the support of the computational particles (i.e. its size) in each spatial dimension. 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 space with infinitesimal span in the velocity directions.

4.1.3 Derivation of the equations of motion

To derive the evolution equations for the free parameters \mathbf{x}_p and \mathbf{v}_p , we require the first moments of the Vlasov equation to be exactly satisfied by the functional forms chosen for the elements. This procedure requires some explanation. First, 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 fields really depends on f_s making the Vlasov equation non-linear. As a consequence the fields used in each Vlasov equation for each element must be the total fields due to all elements, the same entering the complete Vlasov equation for f_s :

$$\frac{\partial f_p}{\partial t} + \mathbf{v} \cdot \frac{\partial f_p}{\partial \mathbf{x}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_p}{\partial \mathbf{v}} = 0 \quad (4.14)$$

Second, 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 equation be satisfied, instead. The moments of the Vlasov equation are obtained multiplying by powers of \mathbf{x} and \mathbf{v} and integrating. We indicate the integration over the spatial and velocity domain by the symbol $\langle \dots \rangle \equiv \int d\mathbf{x} \int d\mathbf{v}$.

4.1.3.1 Moment 0

The zeroth order moment ($\langle Vlasov \rangle$) gives:

$$\frac{\partial \langle f_p \rangle}{\partial t} + \left\langle \mathbf{v} \cdot \frac{\partial f_p}{\partial \mathbf{x}} \right\rangle + \left\langle \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_p}{\partial \mathbf{v}} \right\rangle = 0 \quad (4.15)$$

where we used the interchangeability of the integration in $dx dv$ and of the derivation over time. The second and third term are zero because f_p has finite support and vanishes at infinity in the coordinate space and in velocity. Note that the third term vanishes only because all terms of $\mathbf{v} \times \mathbf{B}$ do not include the same component of \mathbf{v} as the partial derivative they multiply. Recalling that $\langle f_p \rangle = N_p$, it follows:

$$\frac{dN_p}{dt} = 0 \quad (4.16)$$

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

4.1.3.2 Three first order spatial moments

The application of the first order moment in \mathbf{x} , ($\langle \mathbf{x} \cdot \mathbf{V} \rangle$), that implies the use of vector notation to treat simultaneously three equations for each spatial coordinate, gives:

$$\frac{\partial \langle f_p \mathbf{x} \rangle}{\partial t} + \left\langle \mathbf{x} \mathbf{v} \cdot \frac{\partial f_p}{\partial \mathbf{x}} \right\rangle + \left\langle \mathbf{x} \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_p}{\partial \mathbf{v}} \right\rangle = 0 \quad (4.17)$$

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

$$\langle f_p \mathbf{x} \rangle = N_p \int S_v (\mathbf{v} - \mathbf{v}_p) d\mathbf{v} \int \mathbf{x} S(\mathbf{x} - \mathbf{x}_p) d\mathbf{x} \quad (4.18)$$

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 \mathbf{x}_p :

$$\langle f_p \mathbf{x} \rangle = N_p \mathbf{x}_p \quad (4.19)$$

The middle term of eq.(4.17) requires the integration of:

$$\int \int \mathbf{x} \left(v_x \frac{\partial f_p}{\partial x} + v_y \frac{\partial f_p}{\partial y} + v_z \frac{\partial f_p}{\partial z} \right) d\mathbf{x} d\mathbf{v} \quad (4.20)$$

When integrated all terms vanish, except that with the derivative in the same direction as the moment. For example in the x direction, only the first term remains,

$$\int \int x v_x \frac{\partial f_p}{\partial x} d\mathbf{x} d\mathbf{v} = - \int \int v_x f_p d\mathbf{x} d\mathbf{v} = -N_p \langle v_x \rangle \quad (4.21)$$

where the finite support of f_p , integration by part and the symmetry of S_v have been used. Using vector notation, the middle term of eq.(4.17) leads to

$$\int \int \mathbf{x} \left(v_x \frac{\partial f_p}{\partial x} + v_y \frac{\partial f_p}{\partial y} + v_z \frac{\partial f_p}{\partial z} \right) d\mathbf{x} d\mathbf{v} = -N_p \mathbf{v}_p \quad (4.22)$$

The end result of applying the first order moment in \mathbf{x} is:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \quad (4.23)$$

4.1.3.3 Three first order velocity moments

The application of the first order moment in \mathbf{v} , ($\langle \mathbf{v} \cdot \nabla f_p \rangle$) gives:

$$\frac{\partial \langle f_p \mathbf{v} \rangle}{\partial t} + \left\langle \mathbf{v} \mathbf{v} \cdot \frac{\partial f_p}{\partial \mathbf{x}} \right\rangle + \left\langle \mathbf{v} \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_p}{\partial \mathbf{v}} \right\rangle = 0 \quad (4.24)$$

The first term is computed in analogy to the first term in the spatial moments. The second term is still zero by virtue of integration over \mathbf{x} . The remaining term can be computed as above, recalling that all terms of $\mathbf{v} \times \mathbf{B}$ do not include the same component of \mathbf{v} as the partial derivative they multiply:

$$\int \int \mathbf{v} \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_p}{\partial \mathbf{v}} d\mathbf{x} d\mathbf{v} = - \int \int \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) f_p d\mathbf{x} d\mathbf{v} \quad (4.25)$$

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

The resulting integral defines two new important quantities, the average electric field and magnetic field acting on a

computational particle:

$$\int \int \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) f_p d\mathbf{x} d\mathbf{v} = -N_p \frac{q_s}{m_s} (\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p) \quad (4.26)$$

where the electric field on a computational particle is:

$$\mathbf{E}_p = \int S_{\mathbf{v}}(\mathbf{v} - \mathbf{v}_p) d\mathbf{v} \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{E}(\mathbf{x}) d\mathbf{x} \quad (4.27)$$

Recalling the property of $S_{\mathbf{v}}$, the formula for \mathbf{E}_p simplifies to:

$$\mathbf{E}_p = \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{E}(\mathbf{x}) d\mathbf{x} \quad (4.28)$$

For the magnetic field,

$$\begin{aligned} & \int \int \frac{q_s}{m_s} \mathbf{v} \times \mathbf{B} f_p d\mathbf{x} d\mathbf{v} = \\ & \frac{q_s}{m_s} \int \mathbf{v} S_{\mathbf{v}}(\mathbf{v} - \mathbf{v}_p) d\mathbf{v} \times \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{B}(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (4.29)$$

resulting in:

$$\mathbf{B}_p = \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{B}(\mathbf{x}) d\mathbf{x} \quad (4.30)$$

The first order moment in \mathbf{v} gives the final equation:

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_s}{m_s} (\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p) \quad (4.31)$$

4.1.4 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:

$$\begin{aligned}
\frac{dN_p}{dt} &= 0 \\
\frac{d\mathbf{x}_p}{dt} &= \mathbf{v}_p \\
\frac{d\mathbf{v}_p}{dt} &= \frac{q_s}{m_s}(\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p)
\end{aligned} \tag{4.32}$$

It is a crucial advantage of the PIC method that its evolution equations resemble the same Newton equations as followed by the regular physical particles. The key difference is that the field is computed as average over the shape function based on the definition of \mathbf{E}_p and \mathbf{B}_p .

Naturally, the electric field is itself given by Maxwell's equations which in turn need the charge density and the current density. The particle in cell approach provides them immediately as integrals of the distribution function over the velocity variable. Using the functional form for the distribution function of each computational element, the charge and current densities become:

$$\begin{aligned}
\rho_s(\mathbf{x}, t) &= \sum_p q_s N_p S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \\
\mathbf{j}_s(\mathbf{x}, t) &= \sum_p q_s N_p \mathbf{v}_p S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p)
\end{aligned} \tag{4.33}$$

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.

In summary, the particle method, as derived above, is a finite-element-like discretisation of phase space. The continuum distribution function is replaced by a discrete mathematical representation provided by the superposition of moving fixed-shape elements. The pitfalls of this representation

have been studied in depth in the past. A full discussion of this aspect is beyond the scope of the present review. We refer the reader to [BL04b, HE88] for additional comments. In extreme summary, a element of phase space initially described correctly by the shape functions chosen for the discretisation would be distorted by non-uniform electric and magnetic fields. Instead, the computational particles have fixed shape and neglect this effect. This approximation introduces of course an error that can be interpreted as a modification of the phase space. It would be misleading to confuse this error with numerical diffusion, the distortion of phase space does not lead to any spatial propagation of information, as the Liouville theorem insures the conservation of the phase-space volume of each element, a process that is still correctly described by the discretised system. Alternatives to PIC have been developed to remove this error by introducing additional degrees of freedom for each computational element (superparticles) [CLD⁺96, BH98]. The additional degrees of freedom can describe the particle shape and represent its distortion in phase space. However, the additional complexity of such methods have so far prevented their widespread use.

4.2 Explicit Temporal Discretisation of the Particle Methods

The equations of motion for the computational particles 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 [JH06].

The simplest algorithm and by far the most used is the so-called *leap-frog algorithm* based on staggering the time levels of the velocity and position by half time step: $\mathbf{x}_p(t = n\Delta t) \equiv \mathbf{x}_p^n$ and $\mathbf{v}_p(t = (n + 1/2)\Delta t) \equiv \mathbf{v}_p^{n+1/2}$. The advancement of position from time level n to time level $n + 1$ uses the velocity at mid-point $\mathbf{v}_p^{n+1/2}$, 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 of position over velocity gives the method its name for its resemblance to the children's game bearing the same name (see Fig. 4.3).

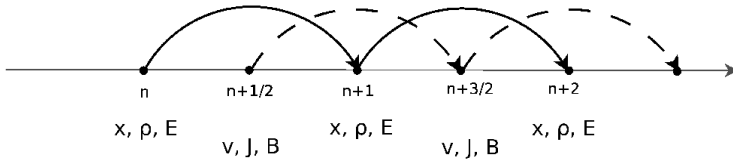


Fig. 4.3. Visual representation of the leap-frog algorithm. The time discretisation is staggered, with the electric field, charge density and particle positions at integer times and the magnetic field, current and particle velocity at half time steps.

The scheme is summarized by:

$$\begin{aligned} \frac{\mathbf{x}_p^{n+1} - \mathbf{x}_p^n}{\Delta t} &= \mathbf{v}_p^{n+1/2} \\ \frac{\mathbf{v}_p^{n+1/2} - \mathbf{v}_p^{n-1/2}}{\Delta t} &= \frac{q_s}{m_s} \mathbf{E}_p(\mathbf{x}_p^n) + \frac{q_s}{m_s} \left(\frac{\mathbf{v}_p^{n+1/2} + \mathbf{v}_p^{n-1/2}}{2} \right) \times \mathbf{B}_p(\mathbf{x}_p^n) \end{aligned} \quad (4.34)$$

where the use of \mathbf{E}_p and \mathbf{B}_p implies knowing the solution of Maxwell's equations given the particle positions at time level n . The first equation is clearly explicit, the second is triv-

ially formulated explicitly as a roto-translation of the vector $\mathbf{v}_p^{n+1/2}$ [BL04b].

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:

$$\frac{\mathbf{v}_p^{1/2} - \mathbf{v}_p^0}{\Delta t/2} = \frac{q_s}{m_s} \mathbf{E}_p(x_p^0) + \frac{q_s}{m_s} \left(\frac{\mathbf{v}_p^{1/2} + \mathbf{v}_p^0}{2} \right) \times \mathbf{B}_p(x_p^0) \quad (4.35)$$

4.2.1 Stability of Explicit Time Differencing

The discretisation of the equations of motion using the leap-frog algorithm (or other similar explicit time differencing) introduces a stability constraint. As usual stability can be studied with the Von Neumann analysis [MM95, HE88, BL04b]: the equations are linearized and their stability is studied by Fourier analysis. Note that the leap-frog algorithm is implicit in the Lorentz force term, but explicit on the electric force. As a consequence only the latter is of primary concern and stability can be studied in the limit of zero magnetic field. Assuming a linear (harmonic) dependence of the electric field,

$$\frac{q_s}{m_s} \mathbf{E}_p(\mathbf{x}_p) = -\Omega^2 \mathbf{x}_p \quad (4.36)$$

the Von Neuman analysis searches for solutions of the form $\mathbf{x}_p^n = \widetilde{\mathbf{x}}_p e^{i\omega_N t}$ and $\mathbf{v}_p^n = \widetilde{\mathbf{v}}_p e^{i\omega_N t}$ with the numerical response of the system given by ω_N . Simple algebraic manipulations [HE88, BL04b] lead to:

$$\left(\frac{\Omega \Delta t}{2} \right)^2 = \sin^2 \left(\frac{\omega_N \Delta t}{2} \right) \quad (4.37)$$

Notoriously the \sin function of real arguments has values only in the $[-1, 1]$ interval. As a consequence, for any values of $\Omega\Delta t > 2$, the Von Neuman analysis leads to a complex numerical oscillation frequency, ω_N . Indeed being the problem real to begin with, two complex conjugate solutions arise. The numerical solution becomes unstable, with a numerical growth rate that has no physical equivalence. In practice, the particles heat unboundedly and quickly and the simulation fails within a few time steps.

The first cardinal rule of explicit plasma simulation is that the condition $\Omega\Delta t > 2$ should never be violated, for any typical Ω in the system. Indeed, practice advises the use of a considerably smaller time step of order $\Omega\Delta t = .1$, to avoid numerical heating [BL04b].

The temporal stability constraint produces a first great challenge to the application of particle methods in space weather simulation (but also in most other applied plasma physics problems). As noted above, the range of time scales typical of space weather events is tremendous. What the stability constraint imposes is to choose a time step that is able to resolve with a cadence of about 1/10 the fastest frequency in the system. For space weather applications (see Fig.4.1) this is commonly the electron plasma frequency, at least 5-7 orders of magnitude smaller than the typical scales of evolution of space weather phenomena.

4.3 Space-Time Discretisation of the Field Equations

So far the attention has focused on the particle discretisation of the Vlasov part of the Vlasov-Maxwell system. To complete the simulation method, the Maxwell's equations need to be discretised in space and time.

A wide variety of methods have been developed to do so. Essentially all methods for general electromagnetics can be coupled with particle methods [HE88, BL04b, GVF05]. Nearly

every particle code uses (vastly or slightly) different methods from any other particle code. Below, some general considerations are made, applicable to most if not all of the methods in the literature.

The Maxwell's equations require two inputs, the sources ρ and \mathbf{j} that come from the information carried by the particles. In turn, the outputs are the electric and magnetic fields (or, equivalently the vector and scalar potential). The main concern here is the coupling of Maxwell's equations with the particles and its consequences.

We assume that the fields are known at grid points, not necessarily the same for both fields and are defined as point values or averages over control volumes: $\mathbf{E}_g, \mathbf{B}_g$ where the index g labels the discrete points (possibly different for electric and magnetic field). Similarly, we assume that the chosen Maxwell scheme needs the sources at (possibly different) discrete grid points defined as averages over control volumes V_g defining each discrete point:

$$\rho_g = \sum_s \frac{q_s}{V_g} \int_{\mathbb{V}} \int_{V_g} f d\mathbf{v} d\mathbf{x} \quad (4.38)$$

$$\mathbf{j}_g = \sum_s \frac{q_s}{V_g} \int_{\mathbb{V}} \int_{V_g} \mathbf{v} f d\mathbf{v} d\mathbf{x}$$

Recalling eq.(4.6-4.7), the computation of the sources for Maxwell's equations require computing a 3-dimensional integral of the particle shape over the control volume of each grid point:

$$\rho_g = \sum_p \frac{1}{V_g} \int_{V_g} S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) d\mathbf{x} \quad (4.39)$$

$$\mathbf{j}_g = \sum_p \mathbf{v}_p \frac{1}{V_g} \int_{V_g} S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) d\mathbf{x}$$

where the integrations in velocity space were done easily re-

calling the definition of S_v . The integral in space is straightforward but can involve complicated geometric operations if one allows general shapes for the control volumes. For this reason, the vast majority of PIC methods use an astute choice of the particle shapes and of the control volumes to simplify this effort. If the particle shapes are chosen as in eq. (4.13) and the grid is chosen uniform, Cartesian and with cell sizes in each dimension equal to the particle sizes, a simple and elegant reformulation of the integration step can be obtained recalling property (4.12) of b-splines. Each dimension can be done separately thanks to the choice of Cartesian uniform grid and of the shape of particles as products of shapes in each direction. The integral needed in each direction can be reformulated as

$$\int_{\Delta x_g} S_x(x - x_p) = \int_{-\infty}^{\infty} S_x(x - x_p) b_0 \left(\frac{x - x_g}{\Delta x_g} \right) \quad (4.40)$$

where Δx_g is the interval in x relative to control volume V_g . Choosing, $\Delta x_g = \Delta x_p$ (called simply Δx), recalling the definition of S_x , eq. (4.13) and using eq. (4.12):

$$\int_{-\infty}^{\infty} S_x(x - x_p) b_0 \left(\frac{x - x_g}{\Delta x} \right) = b_{l+1} \left(\frac{x_g - x_p}{\Delta x} \right) \quad (4.41)$$

Collecting the integral in each direction, the so-called interpolation function can be defined:

$$W(\mathbf{x}_g - \mathbf{x}_p) = b_{l+1} \left(\frac{x_g - x_p}{\Delta x} \right) b_{l+1} \left(\frac{y_g - y_p}{\Delta y} \right) b_{l+1} \left(\frac{z_g - z_p}{\Delta z} \right) \quad (4.42)$$

The interpolation functions are a direct consequence of the choice made for the shape functions. Using the b-splines proves a powerful choice as it allows to write the interpolation functions just as b-splines one order higher. The calculation of the source for Maxwell's equations becomes then just the sum of a number of function evaluations without requiring geometrically complex integrals:

$$\begin{aligned}\rho_g &= \frac{1}{V_g} \sum_p q_p W(\mathbf{x}_g - \mathbf{x}_p) \\ \mathbf{j}_g &= \frac{1}{V_g} \sum_p q_p \mathbf{v}_p W(\mathbf{x}_g - \mathbf{x}_p)\end{aligned}\tag{4.43}$$

An analogous set of operations can be carried out for connecting the output of the Maxwell's equations to the particle equations of motion. The needed quantities are the electric and magnetic fields derived above in eq. (4.27,4.30). The definitions for the particle fields require continuum fields. These can be obtained once again using interpolation:

$$\begin{aligned}\mathbf{E}(\mathbf{x}) &= \sum_g \mathbf{E}_g S_{\mathbf{E}}(\mathbf{x} - \mathbf{x}_g) \\ \mathbf{B}(\mathbf{x}) &= \sum_g \mathbf{B}_g S_{\mathbf{B}}(\mathbf{x} - \mathbf{x}_g)\end{aligned}\tag{4.44}$$

where $S_{\mathbf{E}}(\mathbf{x} - \mathbf{x}_g)$ and $S_{\mathbf{B}}(\mathbf{x} - \mathbf{x}_g)$ are the fields interpolation functions. Upon substitution in the definition of the particle fields,

$$\begin{aligned}\mathbf{E}_p &= \sum_g \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{E}_g S_{\mathbf{E}}(\mathbf{x} - \mathbf{x}_g) d\mathbf{x} \\ \mathbf{B}_p &= \sum_g \int S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) \mathbf{B}_g S_{\mathbf{B}}(\mathbf{x} - \mathbf{x}_g) d\mathbf{x}\end{aligned}\tag{4.45}$$

The geometrical complexity of the integrals can be avoided choosing the interpolation functions for the electric and magnetic fields as b-splines of order 0:

$$S_{\mathbf{E},\mathbf{B}}(\mathbf{x} - \mathbf{x}_g) = b_0 \left(\frac{x_g - x_p}{\Delta_x} \right) b_0 \left(\frac{y_g - y_p}{\Delta_y} \right) b_0 \left(\frac{z_g - z_p}{\Delta_z} \right)\tag{4.46}$$

leading to

$$\begin{aligned} \mathbf{E}_p &= \sum_g \mathbf{E}_g W(\mathbf{x}_g - \mathbf{x}_p) d\mathbf{x} \\ \mathbf{B}_p &= \sum_g \mathbf{B}_g W(\mathbf{x}_g - \mathbf{x}_p) d\mathbf{x} \end{aligned} \tag{4.47}$$

with the same interpolation function used for the sources.

While the vast majority of schemes follows the choice of b-splines and uniform grids, more advanced schemes have also been considered. The cell positions can be chosen differently for different fields and different sources. Staggered schemes and Yee lattices are commonly used [SB91, TH05]. Also different order of interpolations can be used for different quantities [VB92]. A more radical solution is to consider non-uniform or even unstructured grids (see Sect. 5.3). In this case the interpolation cannot use the b-spline properties and the convenient definition of interpolation function. Nevertheless, the integral definitions above still hold and one can base the computation of sources and fields using directly the integral definition.

4.3.1 Space Resolution and Finite Grid Instability

The introduction of a grid to compute the field characterizes the particle methods for plasma physics as particle in cell (PIC) methods. The particles move in a continuum space, but their information is projected to grid points using integration over control volumes. This operation reduces the information, collapsing the continuum particle shapes to discrete contributions in a set of points (the grid values). The loss of information allows one numerical instability to creep in: the finite grid instability.

The mathematical study of the finite grid instability is complex, requiring a careful and complete analysis of all

computational steps with the Laplace transformation of time and the Fourier transformation of space. The reader interested in the details is referred to the textbooks on the subject [BL04b, HE88] and to the original work cited therein. The summary of the analysis is that to avoid the finite grid instability in explicit particles methods, the grid spacing must be chosen to satisfy the constraint

$$\Delta x / \lambda_{De} < \varsigma \quad (4.48)$$

where λ_{De} is the Debye length and the parameter ς is of order one and depends on the details of the implementation. For the widely used choice of shape functions as b-spline of order 0 and consequently interpolation functions of order 1, the scheme referred to as cloud in cell (CIC), the literature reports, $\varsigma \approx \pi$.

The practical consequence of the finite grid instability is a tremendous numerical heating of the plasma characterized by an alternatively positive-negative variation of the electric field accompanied by a correlated zig-zag perturbation of the phase-space. Within a few cycles, the energy reaches the bounds of overflow in the machine representation of numbers. The finite grid instability must be avoided at all costs, requiring its stability constraint to be respected everywhere in the domain for all directions.

The consequences of this stability constraint for space weather simulations are devastating. The spatial scale to resolve is the Debye length in the highest density region of the domain where the Debye length is smallest. If using uniform grids, this constraint requires to use cell sizes several orders of magnitude smaller than the scales of interest. Note that this consideration has two sides.

First, often the processes of interest do not reach down to the Debye scale. For example most reconnection events in space are dominated by processes that are on the electron inertial length, $d_e = c/\omega_{pe}$. This scale is still two orders of

magnitude larger than the Debye length. Having to resolve the Debye length purely for numerical stability reasons implies a waste of fully 2 orders of magnitude in each direction, or 10^6 , a million times. If the time scale constraint encountered before hits the explicit approach once, the spatial stability constraint hits it three times, once per spatial dimension. The combined effect of these two constraint makes it imperative to look beyond explicit particle methods.

Second, often space plasmas present regions of high density in contact with regions of lower density, resulting in a large range of Debye lengths. To avoid the finite grid instability, uniform grids need to be chosen to resolve the smallest Debye length in the system. Adaptive schemes can be devised to resolve in each region only the local Debye length, resulting in considerable savings [FS08]. However, even in this case, often the local scales of interest are much larger than the Debye length and the local grid resolution would be more desirably chosen according to other consideration than the need to resolve the Debye length.

4.3.2 Time Discretisation of Maxwell's equations

The most common discretisation in space locates the electric and magnetic fields and the particle charges and currents at different locations. The Yee lattice in Fig. 4.4 finds by far the widest application but alternative approaches are widespread. Many approaches are based on finite differences [BL04b, HE88] or finite volume [SB91]. The electric field and currents are vertex-centered while the magnetic field and the density are cell-centered. Finite element approaches have also been considered [JH06].

Even when the sources are assumed to be known from the explicit particle approach (i.e. are taken from the previous time step), still the Maxwell's equations are a linear coupled system for the fields. Whatever the scheme followed, the

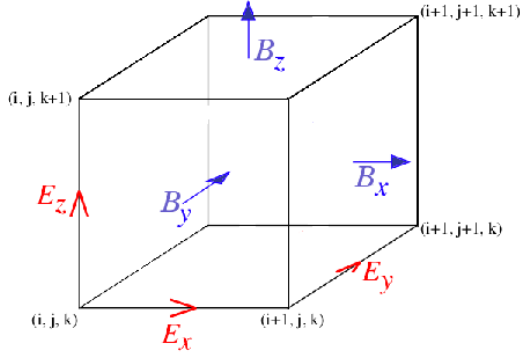


Fig. 4.4. Location of the electromagnetic fields in a typical PIC method based on the Yee lattice discretisation. The current density \mathbf{j}_g is collocated with the electric field. The explicit PIC does not directly use the charge density, except for divergence cleaning (see text), and for that the charge density ρ_g and the correction potential $\delta\varphi_g$ are located on the vertices (i, j, k) .

fields on the discrete grid points will be obtained by numerically discretising the Maxwell's equations that are linear in the fields:

$$\mathbb{M} \begin{pmatrix} \mathbf{E}_g \\ \mathbf{B}_g \end{pmatrix} = \begin{pmatrix} \rho_g \\ \mathbf{j}_g \end{pmatrix} \quad (4.49)$$

where the matrix \mathbb{M} summarizes the linear coupling of the electric and magnetic fields. In the simplest approach the Maxwell's equations are discretised explicitly. The leap-frog algorithm is used again (see Fig. 4.3). The current and the magnetic field are computed at the half time steps: $\mathbf{B}^{n+1/2}$, $\mathbf{j}^{n+1/2}$ and the electric field and the density are computed at the integer time step: \mathbf{E}^n , ρ^n . A time centered marching algorithm follows simply. The magnetic field is computed from the $\nabla \times \mathbf{E}$ Maxwell's equation:

$$\mathbf{B}^{n+1/2} = \mathbf{B}^{n-1/2} - \Delta t \nabla \times \mathbf{E}^n \quad (4.50)$$

and the electric field is computed over a time interval half-

shifted:

$$\mathbf{E}^{n+1} = \mathbf{E}^n + \frac{\Delta t}{\mu_0 \epsilon_0} \left(\nabla \times \mathbf{B}^{n+1/2} - \mu_0 \mathbf{j}^{n+1/2} \right) \quad (4.51)$$

The same simple marching order used for the particles can be used also for the fields resulting in a exceeding simple approach. However, the use of an explicit scheme within the field solver introduces an additional stability constraint: the Courant–Friedrichs–Lewy (CFL) [MM95] condition on the propagation of electromagnetic waves: $c\Delta t < \Delta x$.

This limitation is intrinsic to the Maxwell's equations and can be removed by discretising implicitly just the linear part of the equations. The coupling with the plasma can still be treated explicitly, i.e. the sources are kept frozen to their value at the beginning of the time step, but the coupling between the two curl Maxwell's equations is kept implicitly. Again many methods can be applied. In practice (for historical more than practical reasons), the method of Ref. [MN71] based on the potential formulation is used in most codes where only the field solve is done implicitly but the particle part is still explicit. Instead, the second-order implicit formulation of the Maxwell's equations based on the electric field derived in Ref [RLB02b] is used in codes where also the particles are treated implicitly.

It is worth insisting on the fact that removing the CFL condition on the light waves has no impact on the other two constraints that remain in place as long as the particle part of the algorithm and its coupling with Maxwell's equations is done explicitly.

So far the attention has focused on the two curl Maxwell's equations. In the continuum, the other two divergence conditions are automatically satisfied at all times provided that they are satisfied at the initial time and requiring that de-

tailed charge conservation is satisfied:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot \mathbf{j} \quad (4.52)$$

Once discretised, however, this convenient truth is no longer true. Most schemes enforce automatically the divergence condition on \mathbf{B} . In fluid modeling, especially applied to space weather, great attention needs to go into enforcing the condition that $\nabla \cdot \mathbf{B} = 0$. In PIC methods, instead, this condition is usually trivially satisfied by the judicious choice of a discretisation scheme where $\nabla \cdot \nabla \times \mathbf{E} = 0$ is satisfied. This is usually the case, except in some methods applied to non-uniform grids. In these rare cases, the vast literature developed for MHD can be of help [T00, DKK⁺02]. Instead, a much more serious problem for PIC methods comes from the other divergence conditions, $\epsilon_0 \nabla \cdot \mathbf{E} = \rho$. This condition is normally not satisfied easily.

Two types of approaches are commonly followed.

First, the so-called charge conserving schemes can be used. The interpolation of charge and density is chosen in a way to conserve locally charge ensuring that the discretised version of eq. (4.52) is valid [BL04b]. The approach has two serious drawbacks: to enforce this property one is forced to sacrifice the conservation of momentum (see below) and reportedly the simulations display an increased level of noise requiring many more particles per cell (see Ref. [BL04b] at page 360 and Ref. [MN71]). The advantage is that no elliptic solver is needed as both divergence equations are automatically enforced. This feature leads to excellent parallel scaling when implemented on massively parallel computers [BAY⁺09].

Second, divergence cleaning approaches can be used by simply correcting the electric field for its electrostatic part to enforce Poisson's equation:

$$\nabla^2 \delta \varphi = \nabla \cdot \mathbf{E} - \rho / \epsilon_0 \quad (4.53)$$

where the corrected field is $\mathbf{E}' = \mathbf{E} - \nabla\delta\varphi$. Clearly this achieves the goal by adding a correction. But at the cost of solving an elliptic problem, a cpu-intensive operation. To circumvent this difficulty inexact versions have been proposed [Lan92, MOS⁺00] where the elliptic equation is not solved to convergence but is in fact in a sense converged over multiple time steps, reducing the computational burden at the cost of allowing a controlled but non-zero error in the divergence equation.

4.4 Recapitulation of the application of explicit particle methods to space weather modeling

Collecting the steps gathered so far, the PIC algorithm is summarized by the series of operations depicted in Fig. 4.5.

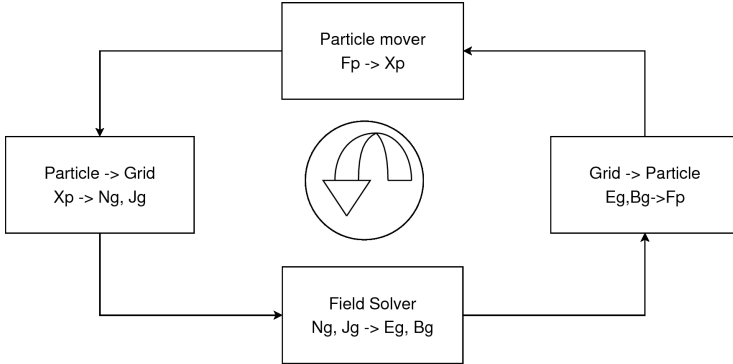


Fig. 4.5. Summary of a computational cycle of the explicit PIC method.

The loop is followed step by step without iterations. Starting for example from the top, the particle equations of motion discretised with the leap-frog algorithm are advanced one step from $\mathbf{x}_p^n, \mathbf{v}_p^{n-1/2}$ to $\mathbf{x}_p^{n+1}, \mathbf{v}_p^{n+1/2}$, the new particle information is projected to the grid to obtain the density and

current density, sources of Maxwell's equations. The latter are then solved for the new fields $\mathbf{E}^{n+1}, \mathbf{B}^{n+1/2}$, given the old fields $\mathbf{E}^n, \mathbf{B}^{n-1/2}$. This operation requires only the sources just computed. Once the new fields are obtained, the particle forces can be computed by interpolation from the grid and the cycle can restart.

The simplicity of the approach explains its wide success. However, as noted above the approach is plagued by two stability constraint coming from the explicit particle discretisation, $\omega_{pe}\Delta < 2$ and $\Delta x/\lambda_{De} < \varsigma$. Additionally, the condition $c\Delta t < \Delta x$ should also be considered when the field solver is also explicit.

The key complexity of a plasma comes from the non-linear coupling of particles and fields. In the real world, the four blocks displayed in Fig. 4.5 are constantly coupled. Instead, the explicit approach freezes temporarily one to apply the other: the particles move for one time step in frozen fields and the Maxwell's equations advance the fields in one time step based on frozen sources. Neither are true physically. This is the key limitation of the explicit approach, especially in multiple scale system such as is typical for space weather problems.

To help the more interested reader familiarize with the explicit approach, Appendix 1 reports a simple but complete implementation of the explicit algorithm above in a MATLAB code for explicit electrostatic 1D systems.

5

Implicit Particle Methods for Space Weather

To avoid the prohibitively large number of time steps and the exceedingly high resolution needed to run realistic problems with explicit particle methods, implicit methods have been considered for several decades [Mas81, Den81, BF82, LCF83].

The starting point of implicit particle methods replaces the explicit time differencing of the equations of motion with an implicit scheme. For example, the so-called θ scheme can be used [VB92, VB95]:

$$\begin{aligned}\mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \mathbf{v}_p^{n+1/2} \Delta t \\ \mathbf{v}_p^{n+1} &= \mathbf{v}_p^n + \frac{q_s \Delta t}{m_s} \left(\mathbf{E}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}) + \mathbf{v}_p^{n+1/2} \times \mathbf{B}_p^n(\mathbf{x}_p^{n+1/2}) \right)\end{aligned}\tag{5.1}$$

where the flexibility of defining a decentering parameter θ is used to vary the properties of the scheme. The quantities at time level $n + \theta$ are computed as weighted averages between the old and new level, $\Psi^{n+\theta} = \Psi^n(1 - \theta) + \Psi^{n+1}\theta$. Other implicit schemes derived from the leap-frog algorithm have been proposed also [CLHP89, Fri90].

The equations (5.1) are implicit for two reasons.

First, to advance the position the new velocity is needed and to advance the velocity the new position is needed (be-

cause the fields need to be evaluated at the mid point between the new and old position). This implicitness is local to each particle and can be handled easily. In fact it can be eliminated with a clever decomposition of the velocity. The velocity equation is more conveniently rewritten as:

$$\mathbf{v}_p^{n+1/2} = \hat{\mathbf{v}}_p + \beta_s \hat{\mathbf{E}}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}) \quad (5.2)$$

where $\beta_s = q_p \Delta t / 2m_p$ (independent of the particle weight and unique to a given species). For convenience, we have introduced hatted quantities obtained by explicit transformation of quantities known from the previous computational cycle:

$$\begin{aligned} \hat{\mathbf{v}}_p &= \boldsymbol{\alpha}_s^n \cdot \mathbf{v}_p^n \\ \hat{\mathbf{E}}_s^{n+\theta} &= \boldsymbol{\alpha}_s^n \cdot \mathbf{E}_s^{n+\theta} \end{aligned} \quad (5.3)$$

The transformation tensor operators $\boldsymbol{\alpha}_s^n$ are defined as:

$$\boldsymbol{\alpha}_s^n = \frac{1}{1 + (\beta_s B^n)^2} (\mathbf{I} - \beta_s \mathbf{I} \times \mathbf{B}^n + \beta_s^2 \mathbf{B}^n \mathbf{B}^n) \quad (5.4)$$

and represent a scaling and rotation of the velocity vector. The tensor operator $\boldsymbol{\alpha}_s^n$ is the same for all computational particles of the same species s .

Second, and more importantly, eqs. (5.1) are implicit because the new fields are needed also. If the electric and magnetic fields were given, the equations of motion could be solved simply. The challenge of the implicit method is that the electric and magnetic fields are not given. Unlike the explicit method, the particle equations of motion need the new fields that are not known yet. The implicit method reintroduces the coupling of the particle equations with the field equations. Implicit methods solve the Maxwell's equations also implicitly in time. The semi-discrete, continuous space, temporal discretisation to Maxwell's equations can be

written as:

$$\begin{aligned}
 \nabla \times \mathbf{E}^{n+\theta} + \frac{1}{c} \frac{\mathbf{B}^{n+1} - \mathbf{B}^n}{\Delta t} &= 0 \\
 \nabla \times \mathbf{B}^{n+\theta} - \frac{1}{c} \frac{\mathbf{E}^{n+1} - \mathbf{E}^n}{\Delta t} &= \frac{4\pi}{c} \mathbf{j}^{n+\frac{1}{2}} \\
 \nabla \cdot \mathbf{E}^{n+\theta} &= 4\pi \rho^{n+\theta} \\
 \nabla \cdot \mathbf{B}^n &= \nabla \cdot \mathbf{B}^{n+1} = 0,
 \end{aligned} \tag{5.5}$$

The interpolation formulas derived above provide the expression for the fields in the equations of motion and for the sources in Maxwell's equations. The coupling of the set of equations is evident.

The parameter $\theta \in [1/2, 1]$ is chosen in order to adjust the numerical dispersion relation for electromagnetic waves (for $\theta < 1/2$, the algorithm is shown to be unstable [BF82]). We note that for $\theta = 1/2$ the scheme is second-order accurate in Δt ; for $1/2 < \theta \leq 1$ the scheme is first-order accurate. The θ parameter can be adjusted to improve energy conservation [Fri90]

The same Von Neumann stability analysis repeated for the implicit equations above leads to a very different solution,

$$\left(\frac{\Omega \Delta t}{2} \right)^2 = \tan^2 \left(\frac{\omega_N \Delta t}{2} \right) \tag{5.6}$$

The \tan function has values in the whole real continuum, allowing the use of any time step Δt . The stability constraint is eliminated and in principle the time step can be selected based only on the physics of interest.

All stability constraints are eliminated and replaced by the sole constraint of selecting the desired accuracy. For space weather simulations this is a great advantage. The scales of interest are often on the electron inertial length,

nearly two orders of magnitude larger than the Debye length that needs to be resolved by the explicit method. Similarly, time can be set to resolve the electron trajectories rather than the fastest waves. Several orders of magnitude can be gained in avoiding unnecessary spatial and temporal resolution. Alone this vastly overwhelms the significant additional cost per computational cycle of the implicit method over the explicit method. As shown below for each case considered, the additional cost of one computational cycle can be a factor of 3 to 10 higher for implicit method (depending whether fully implicit or semi-implicit). But the savings are instead of several orders of magnitude. In the example presented in Sect. 5.4.2 the gain of the implicit method is a staggering 10^6 , overwhelming any additional cost per cycle.

But what happens then to the scales that are not resolved? The question was immediately felt as the implicit particle methods were developed and a considerable body of literature is available, reviewed in [BL04b, BC85]. In summary, the scales that are not resolved remain present but in a distorted fashion. The frequency of all unresolved waves saturates towards the maximum resolved frequency, the Nyquist frequency of $1/\Delta t$. For a choice of a decentering parameter larger than $\theta > 1/2$, the unresolved waves are further damped, at a rate exceeding the natural Landau damping. Physically this allows the energy dissipation towards the unresolved scales, allowing to capture the gross contribution of such scales to the energy balance. A complete analysis of the fate of all plasma waves when the time step is not resolving them is described in Ref. [BF82].

The fundamental problem to address in developing an implicit PIC method is the coupling between the equations of motion and the field equations for the presence of the time advanced electric field (but not magnetic field, that is used from the previous cycle, as no instability is introduced) in the equations of motion and for the appearance [due to the

presence of \mathbf{x}_p and \mathbf{v}_p in eq. (4.33)] of the particle properties in the sources of the Maxwell equations. In both cases the coupling is implicit, so that the new particle properties need to be known before the fields can be computed and likewise the new fields need to be available before the new particle properties can be computed.

In recent years, the direct solution of the coupled system for fields and particles became within reach of modern computational methods and computer resources [KCL05], but traditionally, alternative methods were developed to avoid dealing directly with the full system of coupled non-linear equations.

Below, we consider first the more modern fully implicit approach, and consider the more traditional semi-implicit approach next.

5.1 Fully Implicit Particle Methods

The implicit method requires the coupled solution of the implicit equations of motion (5.1), where the new fields at the advanced time level are required, and of the implicit field equations (5.5), where the new particle properties are needed to compute the sources. The coupling is non-linear, just as the original Vlasov-Maxwell system is also non-linear: the force on the particles is not given but rather depends non-linearly on the fields that in turn depend on the particles. Modern computational science has developed a powerful tool for solving such systems of non-linear equations: the preconditioned Jacobian-Free-Newton-Krylov (JFNK) approach.

In extreme summary and referring the reader to the relevant textbooks on the subject [Kel95], given a non-linear set of equations, $\mathbf{f}(\mathbf{w}) = 0$ for the set of unknowns \mathbf{w} , the JFNK method is based on solving iteratively the residual equation with the Newton method without having to form the Jacobian matrix. The new iteration $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \delta\mathbf{w}$ is ob-

tained by computing the displacement $\delta\mathbf{w}$ using the Newton method [Kel95], i.e. using a Taylor series expansion truncated at first order (to ensure linearity of the resulting equation for $\delta\mathbf{w}$). To avoid forming and storing the Jacobian matrix needed for the Taylor series expansion, the equation for $\delta\mathbf{w}$ is approximated numerically by replacing the derivatives with differences:

$$\mathbf{f}(\mathbf{w}^{(k)}) + \frac{\mathbf{f}(\mathbf{w}^{(k)} + \epsilon\delta\mathbf{w}) - \mathbf{f}(\mathbf{w}^{(k)})}{\epsilon} = 0 \quad (5.7)$$

While formally non-linear, eq. (5.7) is linearized by choosing ϵ small enough to ensure the accurate replacement of the first order derivative with the difference in the Taylor expansion. In practice, ϵ is chosen according to the machine precision [Kel95]. The Jacobian eq. (5.7) is expressed only in terms of residual function evaluations and there is no need to form or store the matrix. To solve eq. (5.7), preconditioned Krylov-based schemes are used. Here, we use the GMRES algorithm since the Jacobian matrix can be non symmetric [Saa03].

In the present problem, the unknowns are in principle the whole set formed by the particle positions and velocities at the advanced time level: \mathbf{x}_p^{n+1} , \mathbf{v}_p^{n+1} and the fields at the advanced level: \mathbf{E}_g^{n+1} and \mathbf{B}_g^{n+1} . While viable, this choice leads in practice to the need to handle the application of the JFNK methods to an extremely large array. The typical particle simulation nowadays employs billions of particle and millions of grid points. To reduce the load on the computer memory, a new technique, the subsidiarity of the particle equations, has been introduced [KCL05].

The starting point is the realization that the particle equations are not directly coupled with each other. Each set of 6 equations for a particle do not require the direct application of the information for the other particles except via the fields. So each set of 6 equations can be solved directly

given the fields. Within the JFNK iteration scheme, the latest guess (Newton iteration k) is indeed available. Therefore the Newton method does not need to consider the particles as part of the unknown. Given a guess of the fields, the particle properties at the advanced time level are immediately computable. This is the core property of particle subsidiarity: the JFNK method considers as unknowns only the fields and leaves the particle computation as a user-supplied rule that is part of the residual evaluation for the fields.

In practice, the approach sets the following unknown vector

$$\mathbf{w} = \begin{pmatrix} \mathbf{E}_g^{n+1} \\ \mathbf{B}_g^{n+1} \end{pmatrix} \quad (5.8)$$

and the corresponding residual equation is formed by eq. (5.5). The particle equations become simply a rule on how to compute the sources $\rho_g^{n+\theta}$ and $\mathbf{j}_g^{n+\frac{1}{2}}$. The computational cycle of the explicit scheme is replaced by the interaction between the JFNK solver and the residual equation, as outlined in Fig. 5.1.

The JFNK method requires the user to provide the rule that produces the residual given a value of the unknown vector. The implicit particle method needs to procure such residual. This is done in steps:

- (i) The NK vector \mathbf{w} is converted back into separate electric and magnetic fields: $\mathbf{E}_g^{n+1}, \mathbf{B}_g^{n+1}$.
- (ii) These fields are interpolated to the particles: $\mathbf{E}_p^{n+1}, \mathbf{B}_p^{n+1}$.
- (iii) The mover advances the particles in these fields provided by JFNK.
- (iv) Once the particles are moved, the field sources can be computed: $\rho_g^{n+\theta}$ and $\mathbf{j}_g^{n+\frac{1}{2}}$.
- (v) With the sources, the field equations provide the resid-

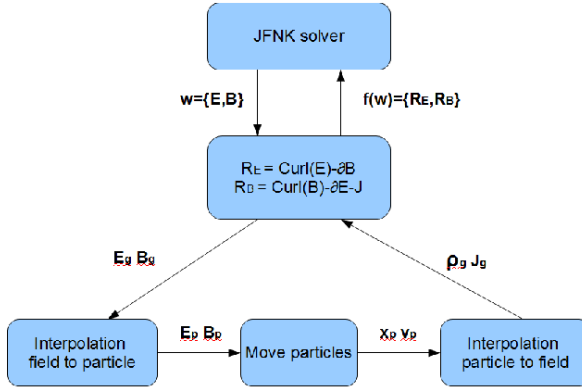


Fig. 5.1. Outline of the interaction of particle mover, field residual computation and JFNK solver in a fully implicit particle method using particle subsidiarity. The equations are indicated in a summarized incomplete notation just for reference.

uals, $\mathbf{r}_{\mathbf{E}_g^{n+1}}, \mathbf{r}_{\mathbf{B}_g^{n+1}}$, needed by JFNK in response to its provided guess for the fields.

- (vi) The NK residual vector is formed: $\mathbf{r}_w = \left(\mathbf{r}_{\mathbf{E}_g^{n+1}}, \mathbf{r}_{\mathbf{B}_g^{n+1}} \right)$

The operations to be conducted are still of the same order of complexity as if the JFNK was directly working also on the particle equations, however, the memory requirements for storing the intermediate Krylov vectors for the JFNK solver are greatly reduced.

The use of the JFNK makes the solution of the system of non-linear equations extremely simple for the user. Powerful libraries, ranging from the simple tool provided in the textbooks [Kel95] to more sophisticated parallel libraries such

as PETSc [BBG⁺09, BBE⁺08, BGMS97] remove the need for the user to code the JFNK method. All that is needed is to provide the simple procedures for computing the residuals. The approach has the advantage of minimizing the personal time of the programmers who develop the software. However, the computational cost of one compute cycle to bring the JFNK to completion is several times that of the simple explicit method described above. This additional cost is vastly overwhelmed by the gains in grid spacing and time step allowed by the removal of the stability constraint.

5.2 Semi Implicit Methods

The full implicit formulation described above has become tractable numerically only in recent years thanks to the progress in the JFNK approach. Previously, the task was considered beyond reach and methods were developed to simplify the non-linear coupling and reduce it to a more tractable linear problem. The semi-implicit approach still preserves many of the good properties of stability of the fully implicit method.

The research in the field has organized itself into two lines: the implicit moment method [BF82] and the direct implicit method [LCF83]. While started from significantly different points of view, over the years the two approaches have converged towards a more common perspective and mathematical framework. The idea is to linearize the coupling between particle and fields and solve it as a coupled linear system instead of the true non-linear system it is. This approach is common in scientific computing and is referred to as semi-implicit. For space weather applications, so far, the implicit moment method has found the majority of applications, while the direct implicit methods is primarily used in inertial fusion research. Below we present in general the semi-implicit method, focusing on the moment implicit approach as the main example but pointing out the differences

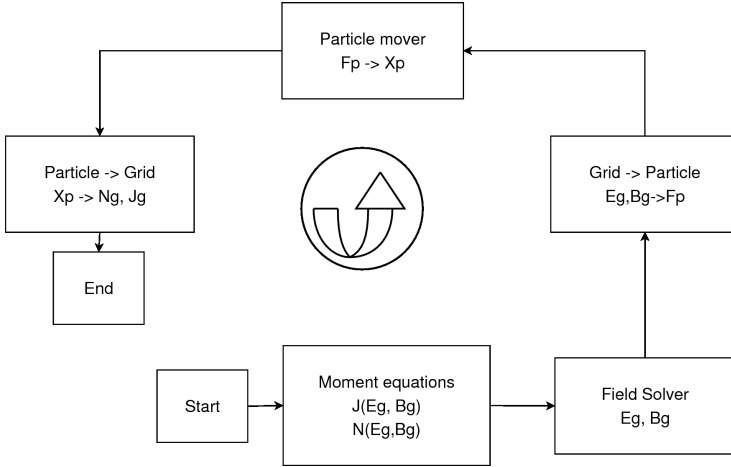


Fig. 5.2. Blocks needed in a cycle of the semi-implicit PIC scheme.

between implicit moment and direct implicit when appropriate.

The semi-implicit method reduces the number of equations that must be solved self-consistently to a set composed by the field equations and a simplified plasma response model that approximates the response of the particles to the field changes over the time step considered. The solution of this set of equations implicitly, and the subsequent explicit solution of the particle equations of motion in the resulting fields, is stable and accurate.

The idea is summarized in Fig. 5.2 where the logic of the sequence of operations involved in the semi-implicit method is shown. The coupling due to the implicit discretisation of both field and particle equations is broken by approximating the sources of the field equations using the moment equations instead of the particle equations directly. Once the field equations are solved within this approximation, the rest of the steps can be completed directly without iterations: with

the new fields, the particle equations of motion can be solved and the new current and density can be computed for the next computational cycle.

The formulation used here is described in Refs. [BF82, VB92, LBR06b, LCF83]. The key step is to derive a suitable set of moment equations that can approximate the response of the plasma particles to the fields over a computational cycle. The approach is based on a series expansion of the interpolation function appearing in the expression for the field sources, eq. (4.33). The expansion is done with respect to the particle position, choosing the center of the expansion as the particle positions at the start of the computational cycle:

$$W(\mathbf{x} - \mathbf{x}_p^{n+1}) = W(\mathbf{x} - \mathbf{x}_p^n) + (\mathbf{x} - \mathbf{x}_p^n) \cdot \nabla W(\mathbf{x} - \mathbf{x}_p^n) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_p^n)(\mathbf{x} - \mathbf{x}_p^n) : \nabla \nabla W(\mathbf{x} - \mathbf{x}_p^n) + \dots \quad (5.9)$$

where a tensor notation is used. The main difference between the implicit moment and direct implicit method is in the choice of the center of the series expansion. In the implicit moment method, the particle mover uses the θ -scheme and it is natural to center the expansion on the old particle position \mathbf{x}_p^n . The direct implicit method uses instead one in a number of different movers based on stabilizing the leap-frog algorithm with implicit variants [Fri90]. In this case the expansion is done around a guess of the new advanced position [LCF83].

The expansion (5.9) can be used to compute the field sources directly using particle information only from the previous computational cycle and removing the need to iterate the particle and field equations. The details of the simple but tedious algebraic manipulations are provided in Ref. [VB92],

the final answer being:

$$\begin{aligned}\rho_s^{n+1} &= \rho_s^n - \Delta t \nabla \cdot \mathbf{J}_s^{n+1/2} \\ \mathbf{J}_s^{n+1/2} &= \hat{\mathbf{J}}_s - \frac{\Delta t}{2} \boldsymbol{\mu}_s \cdot \mathbf{E}_\theta - \frac{\Delta t}{2} \nabla \cdot \hat{\boldsymbol{\Pi}}_s\end{aligned}\quad (5.10)$$

where the following expressions were defined:

$$\begin{aligned}\hat{\mathbf{J}}_s &= \sum_p q_p \hat{\mathbf{v}}_p W(\mathbf{x} - \mathbf{x}_p^n) \\ \hat{\boldsymbol{\Pi}}_s &= \sum_p q_p \hat{\mathbf{v}}_p \hat{\mathbf{v}}_p W(\mathbf{x} - \mathbf{x}_p^n)\end{aligned}\quad (5.11)$$

with the obvious meaning, respectively, of current and pressure tensor based on the transformed hatted velocities. An effective dielectric tensor is defined to express the feedback of the electric field on the plasma current and density:

$$\boldsymbol{\mu}_s^n = -\frac{q_s \rho_s^n}{m_s} \boldsymbol{\alpha}_s^n \quad (5.12)$$

The expression (5.10) for the sources of the Maxwell's equations provide a direct and explicit closure of Maxwell's equations. When eq. (5.10) is inserted in eq. (5.5), the Maxwell's equations can be solved without further coupling with the particle equations. The procedure followed distilled from the equations of motion of the particles an average response that can be used to represent the plasma response within one time step. This is the key property of the semi-implicit method and allows it to retain the once-through approach typical of explicit methods and eliminates the need for expensive iteration procedures that would require to move the particles multiple times per each computational cycle.

In the moment implicit method, the plasma response takes the form presented above and represents a closure of the fluid equations using the particle information to express the pressure tensor. In the direct implicit method, a similar approach is used that relies on the mathematical linearisa-

tion of the response rather than on the use of the moment method [BC85].

5.2.1 Field Solver

Different versions of the semi-implicit PIC method have relied on different solution procedures for the Maxwell equations. In CELESTE1D a potential formulation was applied [VB92], while early versions of CELESTE3D were based on a time-staggered solution of the two divergence Maxwell's equations and of a second order formulation of the two curl Maxwell's equations [RLB02b]. More recently [RLB02b] a consistent second order formulation has been presented and is summarized here.

In the continuum, the two curl Maxwell's equations give a solution with the property of satisfying also the two divergence equations at all times, provided that appropriate initial and boundary conditions are used. Based on this property [RLB02b], the following second order equation is obtained from the two time-discretised (but continuous in space) curl equations in eq. (5.5):

$$\begin{aligned} & (c\theta\Delta t)^2 \left[-\nabla^2 \mathbf{E}^{n+\theta} - \nabla \nabla \cdot (\boldsymbol{\mu}^n \cdot \mathbf{E}^{n+\theta}) \right] + \boldsymbol{\epsilon}^n \cdot \mathbf{E}^{n+\theta} \\ & = \mathbf{E}^n + (c\theta\Delta t) \left(\nabla \times \mathbf{B}^n - \frac{4\pi}{c} \hat{\mathbf{J}}^n \right) - (c\theta\Delta t)^2 \nabla 4\pi \hat{\rho}^n \end{aligned} \quad (5.13)$$

where $\boldsymbol{\mu}^n = \sum_s \boldsymbol{\mu}_s^n$ and $\boldsymbol{\epsilon}^n = \mathbf{I} + \boldsymbol{\mu}^n$, being \mathbf{I} the identity tensor.

As shown in Ref. [RLB02b], the second order formulation for the electric field needs to be coupled with a divergence cleaning step to ensure that

$$\epsilon_0 \nabla \cdot \mathbf{E}^n = \rho^n \quad (5.14)$$

holds for the initial field of each time step.

The boundary conditions for the second order formulation for \mathbf{E} can be derived from the natural boundary conditions

expressed in both \mathbf{E} and \mathbf{B} using theorems of classical electrodynamics [RLB02b]. The magnetic field is computed directly once the new electric field is computed, as:

$$\frac{\mathbf{B}^{n+1} - \mathbf{B}^n}{\Delta t} + \nabla \times \mathbf{E}^{n+\theta} \quad (5.15)$$

Using the temporal second-order temporal formulation above, the field equations are further discretised in space (see e.g. Ref. [RLB02b, LBR06b] for details). The discretised equations and their boundary conditions form a non-symmetric linear system that is solved using GMRES [Saa03]. For the divergence cleaning equation, CG (or FFT on a uniform grids) [Saa03] can be used since the discretised equation leads to a symmetric matrix.

5.2.2 Particle mover

Once the fields are computed from the semi-implicit moment method, the particles can be advanced. The set of equations of motions (5.1) have an inner coupling among themselves due to the appearance of the new velocity in the equation for the position and of the new position in the equation for the velocity. This coupling is purely local at the level of each particle and does not involve coupling with other particles. Instead of simply solving the 6 coupled equations of motions considering position and velocity as unknowns, the formal linearity of the explicit quadrature of the velocity provided in eq.(5.2) can be used. Substituting the first of the Newton eqs. (5.1) in the second, the nonlinear (vectorial) function expressing the discretised Newton equations is:

$$\mathbf{f}(\mathbf{x}_p^{n+1}) = \mathbf{x}_p^{n+1} - \mathbf{x}_p^n - \Delta t \hat{\mathbf{v}}_p - \Delta t \beta_s \hat{\mathbf{E}}_p^{n+\theta}(\mathbf{x}_p^{n+1/2}) \quad (5.16)$$

The three residual equations above have only three unknowns, the components of \mathbf{x}_p^{n+1} , all other quantities being known.

The equations are non-linear due to the spatial dependence of the electric and magnetic fields.

Two methods have been used in the literature. First, a predictor-corrector (PC) approach can be used to solve the non-linear system of equations of motions [VB95]. Alternatively, less commonly, the Newton method [Kel95] can be used to solve the set of equations of motion to convergence. While the JFNK method can be used for this task also, the simplicity of the 3 coupled equations allows the use of the full-fledged Newton method.

The use of the Newton-based mover ensure solution of the equations of motion to convergence of the non-linear iteration, when instead the PC mover uses typically a fixed number of iterations allowing for an uncontrolled error. This feature endows the Newton-based mover with an improved energy conservation in the overall scheme [LBR06b]. Both movers can be extended to the relativistic case [NTZL07].

5.2.3 Stability of the semi-implicit methods

The stability properties of the semi-implicit method described above have been studied extensively in the past [BF82]. The semi-implicit particle mover removes the need to resolve the electron plasma frequency, and the implicit formulation of the field equations removes the need to resolve the speed of light.

The time step constraints are replaced by an accuracy limit arising from the derivation of the fluid moment equations using the series expansion in eq. (5.9). This limit restricts the mean particle motion to one grid cell per time step [BF82], i.e.

$$v_{th,e}\Delta t/\Delta x < 1, \quad (5.17)$$

The finite grid instability limit for the explicit method, $\Delta x <$

$\varsigma\lambda_{De}$ is replaced by [BF82]

$$\Delta x < \varepsilon^{-1} v_{th,e} \Delta t, \quad (5.18)$$

that allows large cell sizes to be used when large time steps are taken.

This is a key feature of the semi-implicit method: time and space scales can be chosen freely according to the desired accuracy, but the ratio of spatial and length scales is not free. Rather it must stay within the bounds just outlined:

$$\varepsilon < v_{th,e} \Delta t / \Delta x < 1, \quad (5.19)$$

The upper limit is generally the main concern. The lower limit, ε , is in practical cases usually a very small number that can often be approximated by zero. The result is that the conditions can be satisfied by selecting large grid spacings and large time steps. In principle, the lower limit would prevent the choice of using large grid spacings with small time step, but as noted in practice this limitation is weak to non-existent and the time step can be chosen as small as desired. However, the condition $v_{th,e} \Delta t / \Delta x < 1$ should never be violated.

The gain afforded by the relaxation of the stability limits is two-fold.

First, the time step can far exceed the explicit limit. In typical space weather problems, the electron plasma frequency is far smaller than the time scales of interest and its accurate resolution is not needed. The processes developing at the sub- Δt scale are averaged and their energy is damped by a numerically-enhanced Landau damping. In other approaches, such as the gyrokinetic or hybrid approach [Lip02], such processes are completely removed and the energy channel towards them is interrupted, removing for example the possibility to exchange energy between sub- Δt fluctuations and particles. In the semi-implicit approach, instead, the sub- Δt scales remain active and the energy channel remains

open. This is a crucial feature to retain in a full kinetic approach. Furthermore, when additional resolution of the smallest scales is needed, the semi-implicit method can access the same accuracy of the explicit method simply using a smaller time step and grid spacing. This feature is not accessible to reduced model, e.g. gyroaveraged methods, that remove the small scales entirely.

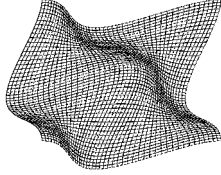
Second, the cell sizes can far exceed the Debye length. Often the scales of interest are much larger than the Debye length. The ability to retain a full kinetic treatment without the need to resolve the Debye length results in a much reduced cost for the semi-implicit PIC method.

5.3 Adaptive Gridding

As noted above the particle in cell method is essentially based on the assumption that the particles have a finite size, and that finite size is fixed and equal to the cell size. The approach is therefore naturally suited to uniform grids. The extension to non-uniform, adaptive grids is not straightforward. Nevertheless there are noteworthy attempts to expand the PIC method to non-uniform grids. We review below some of the most notable approaches of interest to space weather problems. As a matter of order, we organize the topic here into three groups.

First, we consider methods where the grid is non-uniform but still retain a regular structure for the connectivity that allow one to map the grid to a uniform logical grid. This is the case when all the nodes have equal connectivity, each node being in contact with 2 others in 1D, 4 in 2D and 6 in 3D. An example of such an arrangement is presented in 2D in Fig. 5.3. This approach can be defined as moving mesh adaptation (MMA) as it can be imagined as the end result of a process that distorts the grid by attracting more points in the regions of interest and away from regions where the

a) Physical domain



b) Logical domain

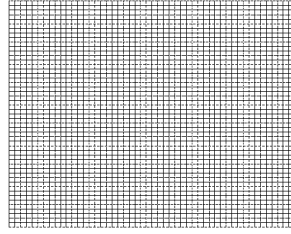


Fig. 5.3. Conceptual representation of moving grid adaptation. The grid in the physical domain x on the left can be mapped to a uniform grid in the logical coordinates ξ .

system is smooth an uneventful. This approach has been followed for example within the Celeste implicit moment method and is reviewed in [Bra93, Bra91, Lap08].

Second, adaptive mesh refinement (AMR) [BO84] can be used for the field part of the PIC method. Main examples of this approach are [VCF⁺04, FS08]. The latter approach was developed especially for space weather application and successful investigations have been reported demonstrating the ability to resolve with the AMR approach the localized regions of reconnection embedded in larger systems [FS08].

Lastly, unstructured grids can be used [JH06] starting from finite element methods to discretise the fields.

All these approaches need to deal with the fact that as the grid spacing changes locally, the particles and the cells cannot always have the same size. As noted above the derivation presented is based on such an equality to provide a simple efficient interpolation between particles and cells. The first two approaches have relaxed the requirement that the particle size remain constant and have based the interpolation on the local grid spacing. The particles have the local size of the cells they are embedded in. For the first approach

this is achieved by assuming that the particle shape function is constant in the logical space ξ rather than in the physical space x . If the shape functions are chosen in the logical space, the interpolation functions can still be computed with the b-spline chain rule and the same formulas of uniform grids can still be used. In the second approach, typically a patch-based AMR approach is used and the particles have the size of the cells in the patch where they reside and interpolation can proceed in each patch as if it was a uniform grid.

This simplification of the interpolation step on adaptive PIC has a serious consequence: the exact derivation breaks down, the changing particle shape introduces new terms proportional to the temporal derivative of the shape function. The derivation of the PIC method outlined in Set. 4.1 assumed a fixed particle shape. All the integrations have been done using such fixed shapes. When this process is followed, a simple analysis proves that momentum is conserved exactly when the same interpolation function is used for all quantities [HE88, GVF05] and provided that the solution of the Maxwell's equations also conserve momentum (i.e. the numerical Green's function reflects the properties of anti-symmetry of the exact continuum Green's function).

When the shape functions change in time the mathematical derivation of the PIC method is modified. In particular, the derivation of the equations of motion require additional terms related to the temporal variation of the shape function. However, in practice these additional terms are neglected and the resulting schemes do not conserve momentum. A side effect is that particles experience a self-force, due to their traversing regions of different sizes [VCM⁺02, CN10]. For the moving mesh approach, this error is controlled by using smooth grids where the spacing changes progressively. Being the neglected terms proportional to the changes in the shape function, when the changes are small

the error is small. In the AMR approach, however, the change in particle shape is sudden at the interface between patches and leads to severe errors that need to be corrected with appropriately devised methods [CN10, VCM⁺02].

5.4 Practical use of particle simulation in space weather

The application of particle simulations to space problems is extremely wide. The reader is advised to seek other reviews. As an example the published proceedings of the series of the International School/Symposium on Space Simulation (e.g. Ref. [BDS03]) or of the conference series ASTRONUM (see e.g. [PACZ09]) provide recent up to date overviews. We focus here, instead, on the critical aspect of space weather: the coupling of large and small scales. The focus will be on the use of implicit particle simulation as it has the better promise of being able to actually model space weather events at the kinetic level. Conversely, the explicit approach is more suitable to zero in on some detailed physics, a topic of interest to space weather but better described as fundamental space science and beyond the scope of the present review.

We report first on the latest developments on the implementation of the implicit methods in codes focusing especially on the latest results relative to iPIC3D, a massively parallel implicit moment PIC code [MLRu10]. Next, we consider the issue of computational costs in the particle simulation of space weather problems with multiple-scale coupling.

5.4.1 Scaling of massively parallel implicit PIC

Particle modeling has led to the development of many codes. For explicit modeling, the simplicity of the algorithm and the relative simplicity of its parallelization has led to the development of many codes based on significantly different imple-

5.4 *Practical use of particle simulation in space weather*

mentations with different particle movers and field discretisation. Implicit codes are much fewer. Essentially there are only two families of codes. The direct implicit method led to the development of the code AVANTI [HL87], later developed into a commercial code, LSP aimed primarily at inertial fusion problems [WROC01]. The implicit moment method has been applied primarily in space problems. Even the names are space-oriented: First came VENUS [BF82], followed by CELESTE [VB92, LBR06b] whose main improvement over VENUS is the use of modern preconditioned Krylov solvers for the field equations. CELESTE3D is available open source [cel]. Parsek2D [MCB⁺09] and iPIC3D [MLRu10] have brought the implicit moment method to massively parallel computers.

Figure 5.4 shows a scaling study conducted on NASA's Pleiades using up to 16,384 processors. The study uses one realistic problem, the study of reconnection in a 2D simulation box, using 32x32 cells per processor, each with 4 species having 196 particles per cell. The scaling is done by increasing the number of processors at a constant load per processor, thereby increasing the system size, all else (time step and number of cycles) remaining the same.

As can be noted, the measured scaling is only 20% lower than the ideal scaling when going from 32 to 16,384 processors. The details of the parallel implementation allowing such performances has been reported in [MLRu10].

5.4.2 *Computational Costs of Particle Modeling in Space Weather*

The key difference between explicit and implicit methods for their application to space weather is the different selection criteria for grid spacing and time step. The explicit method must satisfy the stability constraints while the im-

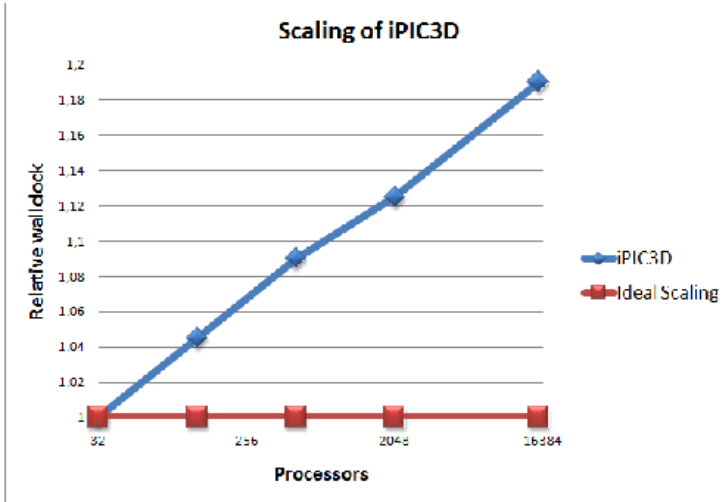


Fig. 5.4. Weak scaling study conducted on NASA's Pleiades supercomputer using the implicit moment code iPIC3D. In red, the ideal scaling is shown where the total CPU time remains constant as the number of processors and the system size are increased keeping the load per processor constant. In blue, the actual observed scaling is shown. Work in collaboration with Stefano Markidis.

PLICIT method need only to resolve the scales of interest. An example is the best illustration of the concept.

Considering again the case of the Earth magnetic tail whose scales are illustrated in Fig. 4.1, Table 5.1 shows the typical choices forced upon the explicit method by the stability constraints. The grid spacing must be chosen equal to the smallest Debye length in the system and the time step must resolve the electron plasma frequency.

The implicit method, instead, can select the accuracy desired and Table 5.1 shows some typical choices. The grid spacing is selected to resolve the scales of reconnection. The smallest scale of interest is the electron skin depth that corresponds to the dissipation layer where field lines decouple

from the electrons, break and reconnect. The choice is then $\Delta x/d_e = 1$ (justified by the consideration that the reconnection region has a much larger electron skin depth than the reference value, see Ref. [LMD⁺10] for details), that corresponds to $\Delta x/\lambda_{De} = 32$. This gain increases by another order of magnitude in other colder regions, such as in the solar wind. The time step is constrained by $\Delta t < \Delta x/v_{the}$ or $\omega_{pe}\Delta t < \Delta x/\lambda_{De} = 100$. To resolve time well, a significantly smaller time step is typically used: $\omega_{pi}\Delta t = .1$. When those choices are made, a gain of order $1.4 \cdot 10^6$ is obtained in CPU time. For reference, if the same simulation is done on the same computer with the same number of processors, for an implicit run that took one day, the corresponding explicit run would require approximately 4,000 years. The gain can be partially offset by the higher cost per computational time. Typically semi-implicit codes require about three times the cost per cycle. In that case, still a thousand years would be required for the corresponding explicit run. The gain achieved in lower temperature regions, common for example in the solar wind, are even more staggering, by several more orders of magnitude. Given these numbers, it is clear that the range of application of implicit PIC is vastly larger than explicit PIC.

The tremendous gain comes at the cost of loosing the information on the scales not resolved. Those scales are not eliminated but rather the waves developing there are distorted to the Nyquist frequency and damped. This leaves the channel open and allows the dissipations to develop. This is exactly what is needed for space weather applications. Not the detail of the most detailed scale but the ability to resolve only the sales of the most important processes.

For the demonstration of the application of the model above to space weather, we refer the reader to the published work done with Celeste3D [LB96, LB97, LB00, LB02, LBD03, RLB02a, RBDL04b, RLB04, RBDL04a, Lap03, RLB03, LBR06b, BGH⁺05,

	Explicit	Implicit	Gain
Δx	$\lambda_{De} = 250\text{m}$	$d_e = 8\text{Km}$	32
Δy	$\lambda_{De} = 250\text{m}$	$d_e = 8\text{Km}$	32
Δz	$\lambda_{De} = 250\text{m}$	$d_e = 8\text{Km}$	32
Δt	$\Delta t = 0.1\omega_{pe}^{-1} = 2.5 \cdot 10^{-6}\text{s}$	$\Delta t = 0.1\omega_{pi}^{-1} = 10^{-4}\text{s}$	$\sqrt{1836}$
Total			$1.4 \cdot 10^6$

Table 5.1. *Paradigmatic choice of parameters for a magnetospheric simulation with explicit and implicit PIC. Spatial and temporal scales are shown in each row. The last column shows the gain in time step and grid spacing for each dimension of space-time. The last row summarizes the total gain.*

CL05, NTZL07, IL07, LKK⁺06, LK07, WL08c, WL08a, WL08b, WLDE08, Lap09] and iPIC3D [MLRu10, LMD⁺10] at the scales of interest outlined in Table 5.1. In particular, detailed comparisons of the application of extremely refined explicit simulations and coarser implicit simulations have proven the ability of the latter to resolve correctly the relevant dissipation scales in processes of reconnection [RBDL04b] and in other space physics problems.

Acknowledgments

The author is grateful to Stefano Markidis, the main developer of Parsek2D and iPIC3D, to Gianni Coppa for his clever implementation of 1D electrostatic PIC in MATLAB (see Appendix 1) as well as for the fruitful discussions on the mathematical foundations of the PIC method. The author is also grateful to Jerry Brackbill for the many years of collaboration and stimulating exchanges of ideas on the implicit PIC

*5.4 Practical use of particle simulation in space weather*⁸⁷

method. The present work is supported by the Onderzoeksfonds KU Leuven (Research Fund KU Leuven) and by the European Commission's Seventh Framework Programme under the grant agreement no. 218816 (SOTERIA project, www.soteria-space.eu). The simulations were conducted on the resources of the NASA Advanced Supercomputing Division (NAS), of the NASA Center for Computational Sciences Division (NCCS) and of the Vlaams Supercomputer Centrum (VSC) at the Katholieke Universiteit Leuven.

Appendix 1

MATLAB program for 1D electrostatic PIC

```

clear all
close all

L=2*pi;
DT=.5;
NT=200;NTOUT=25;
NG=32;

N=1000;
WP=1;
QM=-1;
V0=0.2;
VT=0.0;

XP1=1;
V1=0.0;
mode=1;

Q=WP^2/(QM*N/L);
rho_back=-Q*N/L;
dx=L/NG;

% initial loading for the 2 Stream instability

xp=linspace(0,L-L/N,N)';
vp=VT*randn(N,1);
pm=[1:N]';pm=1-2*mod(pm,2);
vp=vp+pm.*V0;

% Perturbation

vp=vp+V1*sin(2*pi*xp/L*mode);
xp=xp+XP1*(L/N)*sin(2*pi*xp/L*mode);

p=1:N;p=[p p];
un=ones(NG-1,1);
Poisson=spdiags([un -2*un un],[-1 0 1],NG-1,NG-1);

% Main computational cycle

for it=1:NT

    % update xp

    xp=xp+vp*DT;

    % apply bc on the particle positions

```

```
out=(xp<0); xp(out)=xp(out)+L;
out=(xp>=L);xp(out)=xp(out)-L;

% projection p->g
g1=floor(xp/dx-.5)+1;g=[g1;g1+1];
frazl=1-abs(xp/dx-g1+.5);fraz=[frazl;1-frazl];

% apply bc on the projection
out=(g<1);g(out)=g(out)+NG;
out=(g>NG);g(out)=g(out)-NG;

mat=sparse(p,g,fraz,N,NG);
rho=full((Q/dx)*sum(mat))'+rho_back;

% computing fields
Phi=Poisson\(-rho(1:NG-1)*dx^2);Phi=[Phi;0];
Eg=([Phi(NG); Phi(1:NG-1)]-[Phi(2:NG);Phi(1)])/(2*dx);

% projection q->p and update of vp
vp=vp+mat*QM*Eg*DT;

end
```

Bibliography

- [Asc05] M. J. Aschwanden. *Physics of the Solar Corona. An Introduction with Problems and Solutions (2nd edition)*. December 2005.
- [BAY⁺09] K. J. Bowers, B. J. Albright, L. Yin, W. Daughton, V. Roytershteyn, B. Bergen, and T. J. T. Kwan. Advances in petascale kinetic plasma simulation with VPIC and Roadrunner. *Journal of Physics Conference Series*, 180(1):012055–+, July 2009.
- [BBE⁺08] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.0.0, Argonne National Laboratory, 2008.
- [BBG⁺09] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2009. <http://www.mcs.anl.gov/petsc>.
- [BC85] J. U. Brackbill and B. I. Cohen, editors. *Multiple Time Scales*, chapter Simulation of Low-Frequency, Electromagnetic Phenomena in Plasmas, (J.U. Brackbill, D.W. Forslund), page 271. Academic Press, Orlando, 1985.
- [BD07] V. Bothmer and I. A. Daglis. *Space Weather – Physics and Effects*. Praxis Publishing, 2007.
- [BDS03] J. Büchner, C. Dum, and M. Scholer, editors. *Space Plasma Simulation*, volume 615 of *Lecture Notes in Physics*, Berlin Springer Verlag, 2003.
- [BF82] J.U. Brackbill and D.W. Forslund. Simulation of low frequency, electromagnetic phenomena in plasmas. *J. Computat. Phys.*, 46:271, 1982.

- [BGH⁺05] J. Birn, K. Galsgaard, M. Hesse, M. Hoshino, J. Huba, G. Lapenta, P. L. Pritchett, K. Schindler, L. Yin, J. Buchner, T. Neukirch, and E. R. Priest. Forced magnetic reconnection. *Geophys. Res. Lett.*, 32:L06105, 2005.
- [BGMS97] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
- [BH86] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:4446–449, 1986.
- [BH98] William B. Bateson and Dennis W. Hewett. Grid and particle hydrodynamics:: Beyond hydrodynamics via fluid element particle-in-cell. *Journal of Computational Physics*, 144(2):358–378, 1998.
- [BL04a] C.K. Birdsall and A.B. Langdon. *Plasma Physics Via Computer Simulation*. CRC Press, Boca Raton, 2004.
- [BL04b] C.K. Birdsall and A.B. Langdon. *Plasma Physics Via Computer Simulation*. Taylor & Francis, London, 2004.
- [BO84] M. J. Berger and J. Olinger. The AMR technique. *J. Computat. Phys.*, 53:484–512, 1984.
- [Bra91] J.U. Brackbill. An adaptive grid with directional control for toroidal plasma simulation. In *Proc. 14th International Conference on the Numerical Simulation of Plasmas, American Physical Society, Annapolis MD, 1991*, 1991.
- [Bra93] J. U. Brackbill. An adaptive grid with directional control. *J. Computat. Phys.*, 108:38, 1993.
- [cel] <http://code.google.com/p/celeste/>.
- [CL05] E. Camporeale and G. Lapenta. Model of bifurcated current sheets in the Earth’s magnetotail: Equilibrium and stability. *Journal of Geophysical Research (Space Physics)*, 110:7206–+, July 2005.
- [CLD⁺96] G. G. M. Coppa, G. Lapenta, G. Dellapiana, F. Donato, and V. Riccardo. Blob method for kinetic plasma simulation with variable-size particles. *Journal of Computational Physics*, 127(2):268–284, 1996.
- [CLHP89] B. I. Cohen, A. B. Langdon, D. W. Hewett, and R. J. Proccassini. Performance and Optimization of Direct Implicit Particle Simulation. *Journal of Computational Physics*, 81:151–+, March 1989.
- [CN10] P. Colella and P. C. Norgaard. Controlling self-force errors at refinement boundaries for AMR-PIC. *Journal of Computational Physics*, 229:947–957, February 2010.
- [Daw83] J. M. Dawson. Particle simulation of plasmas. *Rev. Mod.*

- Phys.*, 55:403–447, 1983.
- [Den81] J. Denavit. Time-filtering particle stimulations with $\omega_{pe}\Delta t \gg 1$. *J. Computat. Phys.*, 42:337, 1981.
- [DKK⁺02] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic Divergence Cleaning for the MHD Equations. *Journal of Computational Physics*, 175:645–673, January 2002.
- [Fri90] A. Friedman. A Second-Order Implicit Particle Mover with Adjustable Damping. *Journal of Computational Physics*, 90:292–+, October 1990.
- [FS02] D. Frenkel and B. Smit. *Understanding Molecular Simulation*. Academic Press, San Diego, 2002.
- [FS08] K. Fujimoto and R. D. Sydora. Electromagnetic particle-in-cell simulations on magnetic reconnection with adaptive mesh refinement. *Computer Physics Communications*, 178:915–923, June 2008.
- [GVF05] Yu. N. Grigoryev, Vitali Andreevich Vshivkov, and Mikhail Petrovich Fedoruk. *Numerical particle-in-cell methods: theory and applications*. VSP BV, AH Zeist, 2005.
- [HE81] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York, 1981.
- [HE88] R.W. Hockney and J.W. Eastwood. *Computer simulation using particles*. Taylor & Francis, 1988.
- [HL87] D. W. Hewett and A. B. Langdon. Electromagnetic direct implicit plasma simulation. *Journal of Computational Physics*, 72:121–155, September 1987.
- [IL07] M. E. Innocenti and G. Lapenta. Momentum creation by drift instabilities in space and laboratory plasmas. *Plasma Physics and Controlled Fusion*, 49:521–+, December 2007.
- [JH06] G. B. Jacobs and J. S. Hesthaven. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 214:96–121, May 2006.
- [Kal98] May-Britt Kallenrode. *Space Physics*. Springer, 1998.
- [KCL05] H. Kim, L. Chacón, and G. Lapenta. Fully implicit particle-in-cell algorithm. *Bull. Am. Phys. Soc.*, 50:2913, 2005.
- [Kel95] C. T. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, Philadelphia, 1995.
- [Lan92] A. B. Langdon. On enforcing Gauss’ law in electromagnetic particle-in-cell codes. *Computer Physics Communications*, 70:447–450, July 1992.
- [Lap03] G. Lapenta. A new paradigm for 3D Collisionless Magnetic Reconnection. *Space Sci. Rev.*, 107:167–174, April 2003.
- [Lap08] G. Lapenta. Adaptive Multi-Dimensional Particle In Cell. *ArXiv e-prints*, June 2008.

- [Lap09] G. Lapenta. Large-scale momentum exchange by microinstabilities: a process happening in laboratory and space plasmas. *Phys. Scripta*, 80(3):035507–+, September 2009.
- [LB96] G. Lapenta and J.U. Brackbill. Contact discontinuities in collisionless plasmas: A comparison of hybrid and kinetic simulations. *Geophys. Res. Lett.*, 23:1713, 1996.
- [LB97] G. Lapenta and J. U. Brackbill. A kinetic theory for the drift-kink instability. *jgr*, 102:27099–27108, December 1997.
- [LB00] G. Lapenta and J.U. Brackbill. 3d reconnection due to oblique modes: a simulation of harris current sheets. *Non-linear Processes Geophys.*, 7:151, 2000.
- [LB02] G. Lapenta and J.U. Brackbill. Nonlinear evolution of the lower hybrid drift instability: Current sheet thinning and kinking. *Phys. Plasmas*, 9(5):1544–1554, May 2002.
- [LBD03] G. Lapenta, J.U. Brackbill, and W.S. Daughton. The unexpected role of the lower hybrid instability in magnetic reconnection in three dimensions. *Phys. Plasmas*, 10(5):1577–1587, May 2003.
- [LBR06a] 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.
- [LBR06b] G. Lapenta, J. U. Brackbill, and P. Ricci. Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas. *Phys. Plasmas*, 13:055904, 2006.
- [LCF83] A.B. Langdon, BI Cohen, and A Friedman. Direct implicit large time-step particle simulation of plasmas. *J. Computat. Phys.*, 51:107–138, 1983.
- [Lip02] Alexander S. S. Lipatov. *The Hybrid Multiscale Simulation Technology*. Springer, Berlin, 2002.
- [LK07] G. Lapenta and J. King. Study of current intensification by compression in the Earth magnetotail. *Journal of Geophysical Research (Space Physics)*, 112:12204–+, December 2007.
- [LKK⁺06] G. Lapenta, D. Krauss-Varban, H. Karimabadi, J. D. Huba, L. I. Rudakov, and P. Ricci. Kinetic simulations of x-line expansion in 3D reconnection. *grl*, 33:10102–+, May 2006.
- [LMD⁺10] G. Lapenta, S. Markidis, A. Divin, M. Goldman, and D. Newman. Scales of guide field reconnection at the hydrogen mass ratio. *Physics of Plasmas*, 17(8):082106, 2010.
- [Mas81] R.J. Mason. Implicit moment particle simulation of plasmas. *J. Computat. Phys.*, 41:233, 1981.
- [MCB⁺09] S. Markidis, E. Camporeale, D. Burgess, Rizwan-Uddin, and G. Lapenta. Parsek2D: An Implicit Parallel Particle-in-Cell Code. In N. V. Pogorelov, E. Audit, P. Colella, & G. P. Zank, editor, *Astronomical Society of the Pacific Conference Series*, volume 406 of *Astronomical Society of the Pacific Conference*

- Series*, pages 237–+, April 2009.
- [MLRu10] Stefano Markidis, Giovanni Lapenta, and Rizwan-uddin. Multi-scale simulations of plasma with ipic3d. *Mathematics and Computers in Simulation*, 80(7):1509–1519, 2010.
- [MM95] K.W Morton and D.F. Mayers. *Iterative methods for linear and nonlinear equations*. SIAM, Philadelphia, 1995.
- [MN71] R. L. Morse and C. W. Nielson. Numerical Simulation of the Weibel Instability in One and Two Dimensions. *Physics of Fluids*, 14:830–840, April 1971.
- [MOS⁺00] C.-D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, and U. Voß. Divergence Correction Techniques for Maxwell Solvers Based on a Hyperbolic Model. *Journal of Computational Physics*, 161:484–511, July 2000.
- [NTZL07] K. Noguchi, C. Tronci, G. Zuccaro, and G. Lapenta. Formulation of the relativistic moment implicit particle-in-cell method. *Physics of Plasmas*, 14(4):042308–+, April 2007.
- [PACZ09] N. V. Pogorelov, E. Audit, P. Colella, and G. P. Zank, editors. *Numerical Modeling of Space Plasma Flows: ASTRONUM-2008*, volume 406 of *Astronomical Society of the Pacific Conference Series*, April 2009.
- [RBDL04a] P. Ricci, J. U. Brackbill, W. Daughton, and G. Lapenta. Influence of the lower hybrid drift instability on the onset of magnetic reconnection. *Physics of Plasmas*, 11:4489–4500, September 2004.
- [RBDL04b] P. Ricci, J.U. Brackbill, W. Daughton, and G. Lapenta. Collisionless magnetic reconnection in the presence of a guide field. *Phys. Plasmas*, 11(8):4102–4114, 2004.
- [RLB02a] P. Ricci, G. Lapenta, and J.U. Brackbill. Gem challenge: Implicit kinetic simulations with the physical mass ratio. *Geophys. Res. Lett.*, 29:10.1029/2002GL015314, 2002.
- [RLB02b] P. Ricci, G. Lapenta, and J.U. Brackbill. A simplified implicit maxwell solver. *J. Computat. Phys.*, 183:117–141, 2002.
- [RLB03] P. Ricci, G. Lapenta, and J.U. Brackbill. Electron acceleration and heating in collisionless magnetic reconnection. *Phys. Plasmas*, 10(9):3554–3560, 2003.
- [RLB04] P. Ricci, G. Lapenta, and J.U. Brackbill. Structure of the magnetotail current: Kinetic simulation and comparison with satellite observations. *Geophys. Res. Lett.*, 31:L06801, doi:10.1029/2003GL019207, 2004.
- [Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.
- [SB91] D. Sulsky and J. U. Brackbill. A numerical method for suspension flow. *J. Computat. Phys.*, 96:339, 1991.
- [spa] <http://www.ofcm.gov/nswp-sp/text/a-cover.htm>.
- [Tó0] Gábor Tóth. The $\nabla \cdot \mathbf{B} = 0$ constraint in shock-capturing

- magnetohydrodynamics codes. *J. Comput. Phys.*, 161(2):605–652, 2000.
- [TH05] A. Tafove and Susan C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House Publishers, Norwood, 2005.
- [VB92] H. X. Vu and J. U. Brackbill. Celest1d: An implicit, fully-kinetic model for low-frequency, electromagnetic plasma simulation. *Comput. Phys. Comm.*, 69:253, 1992.
- [VB95] H.X. Vu and J.U. Brackbill. Accurate numerical solution of charged particle motion in a magnetic field. *J. Computat. Phys.*, 116:384, 1995.
- [VCF⁺04] J.-L. Vay, P. Colella, A. Friedman, D. P. Grote, P. McCorquodale, and D. B. Serafini. Implementations of mesh refinement schemes for Particle-In-Cell plasma simulations. *Computer Physics Communications*, 164:297–305, December 2004.
- [VCM⁺02] J.-L. Vay, P. Colella, P. McCorquodale, B. van Straalen, A. Friedman, and D. P. Grote. Mesh refinement for particle-in-cell plasma simulations: Applications to and benefits for heavy ion fusion. *Laser and Particle Beams*, 20:569–575, October 2002.
- [WL08a] W. Wan and G. Lapenta. Electron Self-Reinforcing Process of Magnetic Reconnection. *Physical Review Letters*, 101(1):015001–+, July 2008.
- [WL08b] W. Wan and G. Lapenta. Evolutions of non-steady-state magnetic reconnection. *Physics of Plasmas*, 15(10):102302–+, October 2008.
- [WL08c] W. Wan and G. Lapenta. Micro-Macro Coupling in Plasma Self-Organization Processes during Island Coalescence. *Physical Review Letters*, 100(3):035004–+, January 2008.
- [WLDE08] W. Wan, G. Lapenta, G. L. Delzanno, and J. Egedal. Electron acceleration during guide field magnetic reconnection. *Physics of Plasmas*, 15(3):032903–+, March 2008.
- [WROC01] D. R. Welch, D. V. Rose, B. V. Oliver, and R. E. Clark. Simulation techniques for heavy ion fusion chamber transport. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 464(1-3):134–139, 2001.