The Particle-in-Cell (PIC) Method

Kai Germaschewski

Space Science Center & Department of Physics
University of New Hampshire

July 29, 2013

Contributors: W. Fox, H. Ruhl
Overview

1. Kinetic description of plasmas
   - Vlasov-Maxwell description of plasmas

2. The FDTD scheme for solving Maxwell’s equations
   - Maxwell’s equations
   - FDTD

3. Interaction strength
   - Coupling strength
   - Finite-size particles

4. Derivation of the PIC method
   - Numerical Approach
   - Derivation of the equations of motion
   - Time integration

5. Plasma oscillations
A plasma is an ionized gas, where the charged particles (ions and electrons) interact via electromagnetic forces. The equations of motion for a particle $p$ are:

\[
\frac{dx_p}{dt} = v \\
\frac{mv_p}{dt} = F_p
\]

The force $F_p$ on the particle is just the Lorentz force:

\[
F_p = q_p(E(x_p) + v_p \times B(x_p))
\]
Finding the electromagnetic fields

Here, the electromagnetic fields need to be known at every particle’s position. For, e.g., the electrostatic electric field, Coulomb’s Law could be used to find the electric field by summing up the contributions from every other particle:

\[
\mathbf{E}(\mathbf{x}_p) = \sum_{p' \neq p} \frac{1}{4\pi\epsilon_0} \frac{q_p'}{r_{qq'}^2} \hat{\mathbf{r}}_{qq'}
\]
To simulate a plasma, one could just implement an ODE solver that integrates the equations of motion for all particles in the system, and obtains the forces needed from Coulomb’s / Biot-Savart’s Laws. In fact for certain systems, in particular in the electrostatic case, this is actually done, and called *molecular dynamics*. However, for typical systems that we call plasmas, this is computationally not feasible at all due to two main reasons:

- Plasmas consist of a huge number of particles, e.g., there are roughly $10^{30}$ particles in the magnetosphere.
- Using Coloumb’s Law creates an algorithm that scales as $O(N^2)$, where $N$ is the number of particles.
Vlasov-Maxwell description of plasmas

Vlasov equation

\[ \partial_t f_s + \mathbf{v} \cdot \nabla_x f_s + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f_s = \left( \frac{\partial f_s}{\partial t} \right)_{\text{coll}} \]

The electromagnetic fields are obtained from Maxwell’s equations:

\[ \nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \nabla \cdot \mathbf{B} = 0 \]

\[ \frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \frac{\mathbf{j}}{\epsilon_0} \quad \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \]
Finding the source quantities in Maxwell’s equations

Charge density and current density are obtained by integrating the distribution function over velocity space:

\[
\rho = \sum_s q_s \int f(x, v, t) \, d^3v \\
\mathbf{j} = \sum_s q_s \int \mathbf{v} f(x, v, t) \, d^3v
\]

and satisfy the \textit{continuity equation}

\[
\partial_t \rho + \nabla \cdot \mathbf{j} = 0
\]
Too many equations?

\[ \nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \quad \nabla \cdot \mathbf{B} = 0 \]

\[ \frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \frac{j}{\varepsilon_0} \quad \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \]

**Which equations should we use?**

If the divergence equations are satisfied initially (say, at time \( t = 0 \)), then they will remain satisfied at all times:

\[ \partial_t (\nabla \cdot \mathbf{B}) = \nabla \cdot \partial_t \mathbf{B} = \nabla \cdot (-\nabla \times \mathbf{E}) = 0 \]

**Exercise:** What happens for \( \nabla \cdot \mathbf{E} \)?
Discretizing Maxwell’s equations

To be able to solve partial differential equations (PDEs) on a computer, they generally need to be discretized in time and space. One method to do this is to introduce a uniform grid, so, e.g., the function $E(x, y, z, t)$ is represented by its values on the discrete grid:

$$E_{i,j,k}^n = E(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$$
Spatial derivatives are then approximated by finite differences (slopes), and a time integration scheme is used to step the solution from an initial condition at \( t = 0 \) forward in time.

Most numerical schemes try to collocate all degrees of freedom \( (\mathbf{E}, \mathbf{B}) \) on the same spatial locations. It is also highly desirable for the numerical scheme to satisfy discrete equivalents of conserved quantities, though, and for Maxwell’s equation we saw some of those earlier, in particular the fact that the divergence of the fields maintain their initial value. Since there are no magnetic monopoles in reality, it’d be rather nice to not have magnetic monopoles in your simulations, either.
The FDTD method satisfies discrete equivalents of the two divergence conditions in Maxwell’s equation, as well as a discrete Poynting theorem (in particular, energy is conserved if \( \mathbf{j} \cdot \mathbf{E} = 0 \). Unfortunately, it comes at a cost (or at least an inconvenience): The FDTD scheme is staggered in time and space.
On the so-called *Yee grid*, the electric fields live on edges of the grid cell, and the magnetic fields live on the faces.

\[
\begin{align*}
E_x,i+1/2,j,k, & \quad B_x,i,j+1/2,k+1/2 \\
E_y,i,j+1/2,k, & \quad B_y,i+1/2,j,k+1/2 \\
E_z,i,j,k+1/2, & \quad B_z,i+1/2,j+1/2,k
\end{align*}
\]
As one example, the z-component Faraday’s Law

\[
\frac{\partial B_z}{\partial t} = -\partial_x E_y + \partial_y E_x
\]

is discretized as

\[
\frac{B_{z,i+1/2,j+1/2,k}^{n+1} - B_{z,i+1/2,j+1/2,k}^{n}}{\Delta t} = \frac{E_{y,i+1,j+1/2,k}^{n+1/2} - E_{y,i,j+1/2,k}^{n+1/2}}{\Delta x} + \frac{E_{x,i+1/2,j+1,k}^{n+1/2} - E_{x,i+1/2,j,k}^{n+1/2}}{\Delta y}
\]
The divergence of $\nabla \cdot \mathbf{B}$ lives naturally in the cell center:

$$(\nabla \cdot \mathbf{B})_{i+1/2,j+1/2,k+1/2} = \frac{B_{x,i+1,j+1/2,k+1/2} - B_{x,i,j+1/2,k+1/2}}{\Delta x} + \frac{B_{x,i+1/2,j+1,k+1/2} - B_{x,i+1/2,j,k+1/2}}{\Delta y} + \frac{B_{x,i+1/2,j+1/2,k+1} - B_{x,i+1/2,j+1/2,k}}{\Delta z}$$

**Exercise:** Assuming that everything is invariant in $z$-direction, calculate $(\nabla \cdot \mathbf{B})^{n+1/2}_{i+1/2,j+1/2,k+1/2}$ as a function of $(\nabla \cdot \mathbf{B})^n_{i+1/2,j+1/2,k+1/2}$ and $\mathbf{E}^{n+1/2}$. 
An important point to consider in the description of plasmas is the type of interaction between a particle and the rest of the system. A test charge introduced into a plasma causes other charged particles to adjust their position and effectively shield the charge on a length scale called the Debye length $\lambda_D$:

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T}{nq^2}}$$

This material is based on [Lapenta].
Strongly coupled systems

Let’s look at just a $\lambda_D$-sized box of charged particles:
Weakly coupled systems

If the number of particles in the Debye sphere is large, however, the situation looks rather different:

When we talk about plasmas, we normally mean a weakly interacting system dominated by collective behavior, a system with many particles in the Debye sphere, or equivalently, a system dominated by kinetic energy ($E_{\text{kin}} \gg E_{\text{pot}}$).
The quasi-particles that are used to represent the distribution function correspond to a large number of actual particles. But they are also given a finite size, typically related to the grid spacing of the mesh where the electromagnetic field equations are solved.
Finite-size particles interact (about) the same as point particles as long as they do not overlap. However, once they overlap, the forces on the overlapping pieces cancel.
Vlasov-Maxwell

Vlasov equation for species $s$

$$\partial_t f_s + \mathbf{v} \cdot \nabla_x f_s + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f_s = \left( \frac{\partial f_s}{\partial t} \right)_{\text{coll}}$$

together with Maxwell’s equations

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$
$$\nabla \cdot \mathbf{B} = 0$$

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \frac{\mathbf{j}}{\varepsilon_0}$$
$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

Charge density and current density are obtained by integrating the distribution function over velocity space:

$$\rho = \sum_s q_s \int f(x, v, t) \, d^3v$$
$$\mathbf{j} = \sum_s q_s \int \mathbf{v} f(x, v, t) \, d^3v$$
We now approximate the phase space distribution function as a sum of quasi-particles:

\[ f_s(x, v, t) = \sum_p f_p(x, v, t) \]

Each quasi-particle represents a large number of particles that are close to each other in phase space. Instead of delta-functions which would represent a single particle, we now use shape function in configuration and velocity space to represent the distribution function belonging to the quasi-particle \( p \):

\[ f_p(x, v, t) = N_p S_x(x - x_p(t)) S_v(v - v_p(t)) \]
It makes sense to demand that the shape functions can be factorized as, e.g.,

\[ S_x(x - x_p(t)) = S_x(x - x_p(t)) \cdot S_y(y - y_p(t)) \cdot S_z(z - z_p(t)) \]

We further demand certain properties of the shape functions:
- The support of the shape function is compact.
- The shape function is normalized. \[ \int_{-\infty}^{\infty} S_\xi(\xi - \xi_p) \, d\xi = 1 \]
- Symmetry. \[ S_\xi(\xi - \xi_p) = S_\xi(\xi_p - \xi) \]
The standard PIC method chooses the velocity space shape functions to be delta-functions, so that the spatial shape function remains constant in time:

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

For the spatial shape function, the original PIC method used delta-functions, too, but nowadays, *b-splines* are now commonly used.
The first spline $b_0$ is defined as follows:

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

The other b-splines are obtained by convoluting the previous spline with $b_0$:

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

The first 3 b-splines are shown here:
The actual spatial shape function is then chosen as

\[ S_x(x - x_p) = \frac{1}{\Delta x} b_l \left( \frac{x - x_p}{\Delta x} \right) \]

where \( \Delta x \) is the scale length for the size of the computational particle.
Derivation of the equations of motion

So far, each quasi-particle has a position and velocity, $x_p$ and $v_p$, whose evolution in time has not yet been specified. We obtain evolution equations by taking moments of the Vlasov equation. For simplicity, the following is written for the 1-d electrostatic case, but generalization to 3-d is straightforward. The Vlasov equation for each quasi-particle distribution function looks like

$$\frac{\partial f_p}{\partial t} + v \frac{\partial f_p}{\partial x} + \frac{q_s E}{m_s} \frac{\partial f_p}{\partial v} = 0.$$ 

where the electric field is still determined by all particles together, though.
The Vlasov equation will not be satisfied for the prescribed particle shapes, however, we demand that the first moments are satisfied. We obtain moments by integrating over configuration and velocity space:

\[ \langle \ldots \rangle \equiv \int dx \int dv \ldots \]
Taking the 0th order moment of the Vlasov equation, it follows that

\[ \frac{dN_p}{dt} = 0 \]

That is, the number of physical particles represented by a quasi-particle remains constant in time.
Derivation of the equations of motion

Taking the moment of the Vlasov equation multiplied by $x$, we find

$$\frac{dx_p}{dt} = v_p$$

This is the same equation of motion that we had for a single real particle.
Finally, we take the moment of the Vlaosv equation multiplied by $v$ and find

$$\frac{dv_p}{dt} = \frac{q_s}{m_s} E_p$$

where the electric field acting on the quasi particle is averaged over space according to the particle’s shape function:

$$E_p = \int S_x(x - x_p)E_x$$
Derivation of the equations of motion

\[
\begin{align*}
\frac{dN_p}{dt} &= 0 \\
\frac{dx_p}{dt} &= v_p \\
\frac{dv_p}{dt} &= \frac{q_s}{m_s} E_p
\end{align*}
\]

In conclusion, the particle in cell method actually solves the usual Newton’s equations of motions for quasi-particles as we had for the actual particles, though the electric field is averaged due to the finite size of the quasi-particles.
Derivation of the equations of motion

The electric field is calculated on the grid and then assumed to be constant within each cell, so it is given as

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

From the definition of $E_p$ we get

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

where the weight function $W$ is essentially just the next higher order b-spline

$$W(x_i - x_p) = b_{l+1} \left( \frac{x_i - x_p}{\Delta x} \right)$$

where $l$ is the order of the b-spline used in the particle shape function b-spline.
The integration of the particle is staggered in time (leap-frog), too. We start with particle velocities at $v_n^p$ and particle positions at $x_{p}^{n+1/2}$. The positions $x_{p}^{n+1/2}$ are used to interpolate the fields to the particle position at time $n+1/2$ and calculate the Lorentz force $F_{p}^{n+1/2}$. The Lorentz force is then used to update the particle velocity according to

$$m \frac{d v_p}{dt} = F_p \quad \Rightarrow \quad m \frac{v_{p}^{n+1} - v_{p}^{n}}{\Delta t} = F_{p}^{n+1/2}.$$ 

Then, the new particle positions can be found using $v_{p}^{n+1}$:

$$\frac{d x_p}{dt} = v_{p} \quad \Rightarrow \quad \frac{x_{p}^{n+3/2} - x_{p}^{n+1/2}}{\Delta t} = v_{p}^{n+1}.$$
The particle positions $x^{n+1/2}$ and $x^{n+3/2}$ are used to calculate the charge densities on the grid at those times. From the charge densities we then find the current densities $j^{n+1}$ to satisfy discretely the charge continuity equation

$$\frac{d\rho}{dt} = \nabla \cdot j \quad \implies \quad \frac{\rho_{i,j,k}^{n+3/2} - \rho_{i,j,k}^{n+1/2}}{\Delta t} = (\nabla \cdot j)^{n+1}$$
The whole cycle showing the interaction between fields and particles is summarized in the following diagram.
We skipped the details of numerically solving Newton’s 2nd Law with the Lorentz force, which contains the particle velocity, too. One common option is to use the pusher developed by Boris [1970]:

We split the Lorentz force into electric and magnetic parts:

\[
F_p = F_{elec,p} + F_{magn,p} = q_s E_p + q_s v_p \times B_p
\]

and then split the update as follows:

\[
m_p \frac{v^- - v^n}{\Delta t/2} = q_s E_p^{n+1/2} \quad \Rightarrow \quad v^- = v^n + \frac{q_p}{m_p} E_p^{n+1/2}
\]

\[
m_p \frac{v^+ - v^-}{\Delta t} = q_s \frac{v^- + v^-}{2} \times B^{n+1/2} \quad \Rightarrow \quad v^+ = \text{rotation of } v^- \text{ around } B^{n+1/2}
\]

\[
m_p \frac{v^{n+1} - v^+}{\Delta t/2} = q_s E_p^{n+1/2} \quad \Rightarrow \quad v^{n+1} = v^+ + \frac{q_p}{m_p} E_p^{n+1/2}
\]
Courant-Friedrichs-Lewy conditions:

\[ c\Delta t < \Delta x \]
\[ \omega_{pe}\Delta t < 2 \]

are needed for avoiding numerical instability.
Plasma oscillations

Single-fluid picture:

\[ m \frac{\partial v_1}{\partial t} = qE_1 \quad \nabla \times B_1 = j_1 + \epsilon_0 \frac{\partial E}{\partial t} \]

\[ \implies \omega^2 = \omega_{pe}^2 = \frac{n_0 q^2}{\epsilon_0 m} \]

With background flow:

\[ \omega = k \cdot v_0 \pm \omega_{pe} \]