Particle-in-cell plasma simulation with OmpSs-2

Rodrigo Arias Mallo

July 2, 2019

Outline

Introduction

What exactly is a plasma?
Talk about 2D only.
Introduce species.
Plasma must be neutral.

Introduction

A plasma is an almost neutral gas of charged and neutral particles which exhibits collective behavior.

Simulation can provide insight in the behavior of plasma without expensive laboratory equipment.

Vlaslov equation

The Vlaslov equation describes the evolution of a plasma, with long range iteration between particles.

$$\frac{\partial f_j}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f_j}{\partial \boldsymbol{x}} + \frac{q_j}{m_j} \left(\boldsymbol{E} + \frac{\boldsymbol{v} \times \boldsymbol{B}}{c} \right) \cdot \frac{\partial f_j}{\partial \boldsymbol{v}} = 0 \tag{1}$$

Where $f_j(\boldsymbol{x}, \boldsymbol{v}, t)$ is the distribution function of the specie j, and the fields electric $\boldsymbol{E}$ and magnetic $\boldsymbol{B}$ are determined by Maxwell equations.

Maxwell equations

The evolution of the fields is governed by the Maxwell equations

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

Where ρ is the charge density distribution, $\boldsymbol{J}$ the current field and ϵ_0 and μ_0 are constants.

Electrostatic approximation

In our model, we will only have a **strong fixed magnetic field B_0** and the Maxwell equations can be simplified.

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon_0} \qquad\qquad \nabla \times \boldsymbol{E} = 0 \tag{3}$$

Electrostatic approximation

In our model, we will only have a **strong fixed magnetic field $\boldsymbol{B}_0$** and the Maxwell equations can be simplified.

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon_0} \qquad\qquad \nabla \times \boldsymbol{E} = 0 \tag{3}$$

As the electric field is irrotational, we can introduce a scalar field ϕ, the electric potential, and rewrite $\boldsymbol{E}$ as a gradient $\boldsymbol{E} = -\nabla\phi$. Finally, if we solve for ϕ we obtain the Poisson equation:

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}, \tag{4}$$

This approximation is called **electrostatic** as opposed to the **electromagnetic** case, when $\boldsymbol{B}$ varies with time.

The particle-in-cell method

Solving the Vlaslov equation directly is very **computationally expensive**. The current solution is to introduce a spatial grid where the Maxwell equations are solved.

The distribution function f_j is then modeled by macro-particles which interact with the grid by an interpolation method.

This method is known as **particle-in-cell** (PIC).

The particle-in-cell method

The space of length $\boldsymbol{L}$ is discretized into grid points with a separation $\Delta\boldsymbol{x}$.

The scalar fields ρ and ϕ are now matrices of size $\boldsymbol{G} = (G_x, G_y)$

The vector field $\boldsymbol{E}$ is decomposed into E_x and E_y, which are now matrices of the same size.

The particle-in-cell method

The PIC method has four main steps

- **Charge accumulation**: The charge density ρ is interpolated in the grid from the particle positions.

The particle-in-cell method

The PIC method has four main steps

- **Charge accumulation**: The charge density ρ is interpolated in the grid from the particle positions.
- **Solve field equation**: From the charge density the electric potential is obtained ϕ and then the electric field $\boldsymbol{E}$.

The particle-in-cell method

The PIC method has four main steps

- **Charge accumulation**: The charge density ρ is interpolated in the grid from the particle positions.
- **Solve field equation**: From the charge density the electric potential is obtained ϕ and then the electric field $\boldsymbol{E}$.
- **Interpolation of electric field**: The electric field is interpolated back to the particle positions.

The particle-in-cell method

The PIC method has four main steps

- **Charge accumulation**: The charge density ρ is interpolated in the grid from the particle positions.
- **Solve field equation**: From the charge density the electric potential is obtained ϕ and then the electric field $\boldsymbol{E}$.
- **Interpolation of electric field**: The electric field is interpolated back to the particle positions.
- **Particle motion**: The force is computed from the electric field at the particle position and the particle is moved accordingly.

Charge accumulation

At each grid point g at $\boldsymbol{x}$, the charge of each particle p at $\boldsymbol{x}_p$ is accumulated using an interpolation function W.

$$\rho(\boldsymbol{x}) = \sum_p q\,W(\boldsymbol{x} - \boldsymbol{x}_p) + \rho_0 \tag{5}$$

Using linear interpolation, we can define W for two dimensions as

$$W(\boldsymbol{x}) = \begin{cases} \left(1 - \dfrac{|x|}{\Delta x}\right)\left(1 - \dfrac{|y|}{\Delta y}\right) & \text{if } -\Delta\boldsymbol{x} < \boldsymbol{x} < \Delta\boldsymbol{x} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Charge accumulation

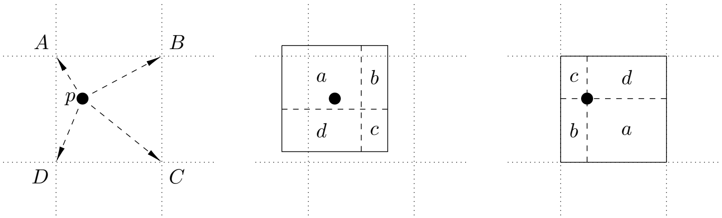

Figure: Interpolation of particle p charge into the four grid points A to D.

Field equations

First the electric potential ϕ must be computed from ρ, and then the electric field $\boldsymbol{E}$.

Several methods are available for solving the Poisson equation:

- **Iterative methods**: Jacobi, Gauss-Seidel, Successive Over Relaxation (SOR), Chebyshev...
- **Matrix methods** Thomas Tridiagonal algorithm, Conjugate-Gradient, LU, Incomplete Decomposition...
- **Spectral methods** A family of methods that use the fast Fourier transform (FFT).

We will only focus on two methods: The LU decomposition used as debug, and MFT, a spectral method with complexity $O(N_g \log N_g)$.

LU solver

The Poisson equation is transformed in a linear system of N_g equations (one for each grid point).

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0} \quad \rightarrow \quad A \frac{\phi}{\Delta x^2 \Delta y^2} = -\frac{\rho}{\epsilon_0} \quad \rightarrow \quad Ax = b \qquad (7)$$

The coefficient matrix A has non-zero coefficients only at $a_{ii} = 4$ and $a_{ij} = -1$ with $j \in \{i+1, i-1, i+N_x, i-N_x\} \mod N_x$, for all $0 \leq i \leq Ng$.

The decomposition $A = LU$ is used to form two systems of equations $Ux = u$ and $Ly = b$, with a computational cost of $O(N_g^3)$ (but only at the beginning), which are solved in each iteration with cost $O(N_g^2)$.

MFT solver

Let $g =$

$$g \xrightarrow{\text{FFT}} \hat{g} \xrightarrow{\hat{G}} \hat{\phi} \xrightarrow{\text{IFFT}} \phi$$

Sequential simulator

A sequential version of the simulator was produced to test the model with a visualization module to see the fields in real-time.

Both simulations in 1D and 2D are supported.

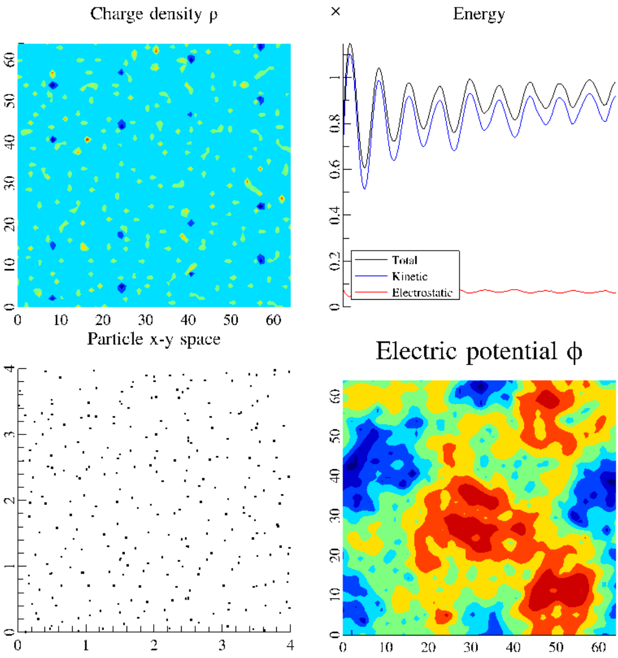

Figure: Example run in 2D of the simulator in debug mode.

Test: Conservation of energy

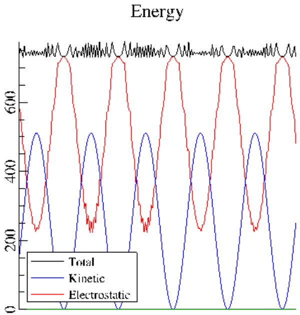

Figure: Energy conservation in two particle test as shown in the simulator.

Parallelization

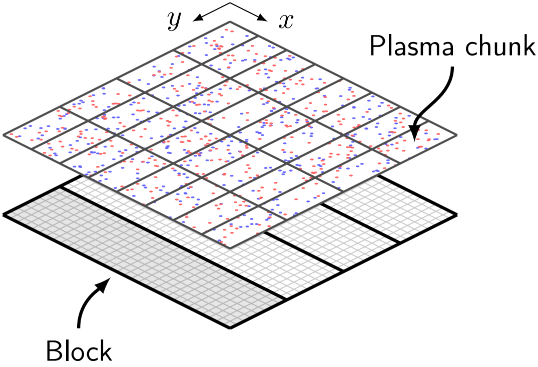

Figure: Domain decomposition: The plasma is divided into chunks in both directions and the fields into blocks in the Y dimension only

Multithread FFT

The FFTW library has a very bad performance with threads.

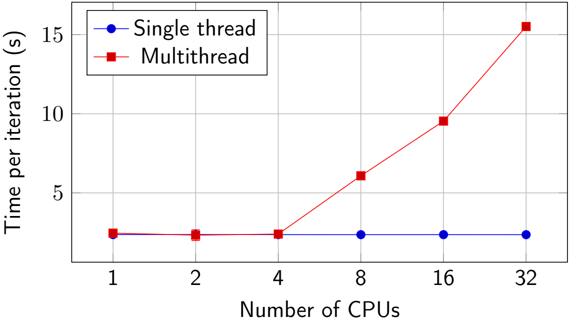

Figure: The number of CPUs is increased with only one process: the solver cannot scale and the time per iteration increases. Configuration used: $N_p = 5 \times 10^5$, $N_g = 8192 \times 8192$.

Strong scaling with $N_g = 2048^2$

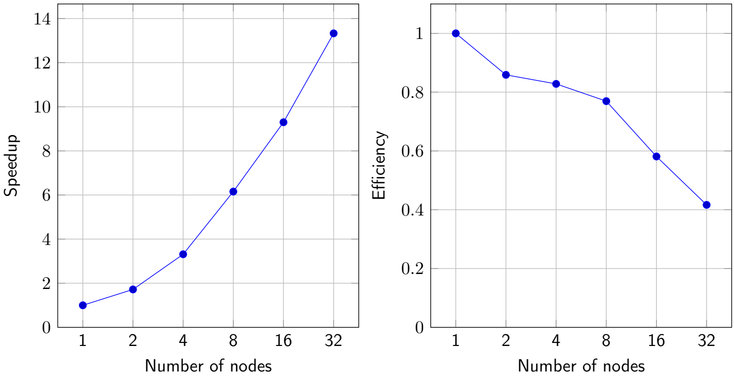

Figure: Strong scaling with configuration: $N_p = 1 \times 10^8$, $N_g = 2048^2$, $N_c = 128$, one process per node, using each 48 cores.

End

Thanks for your attention.