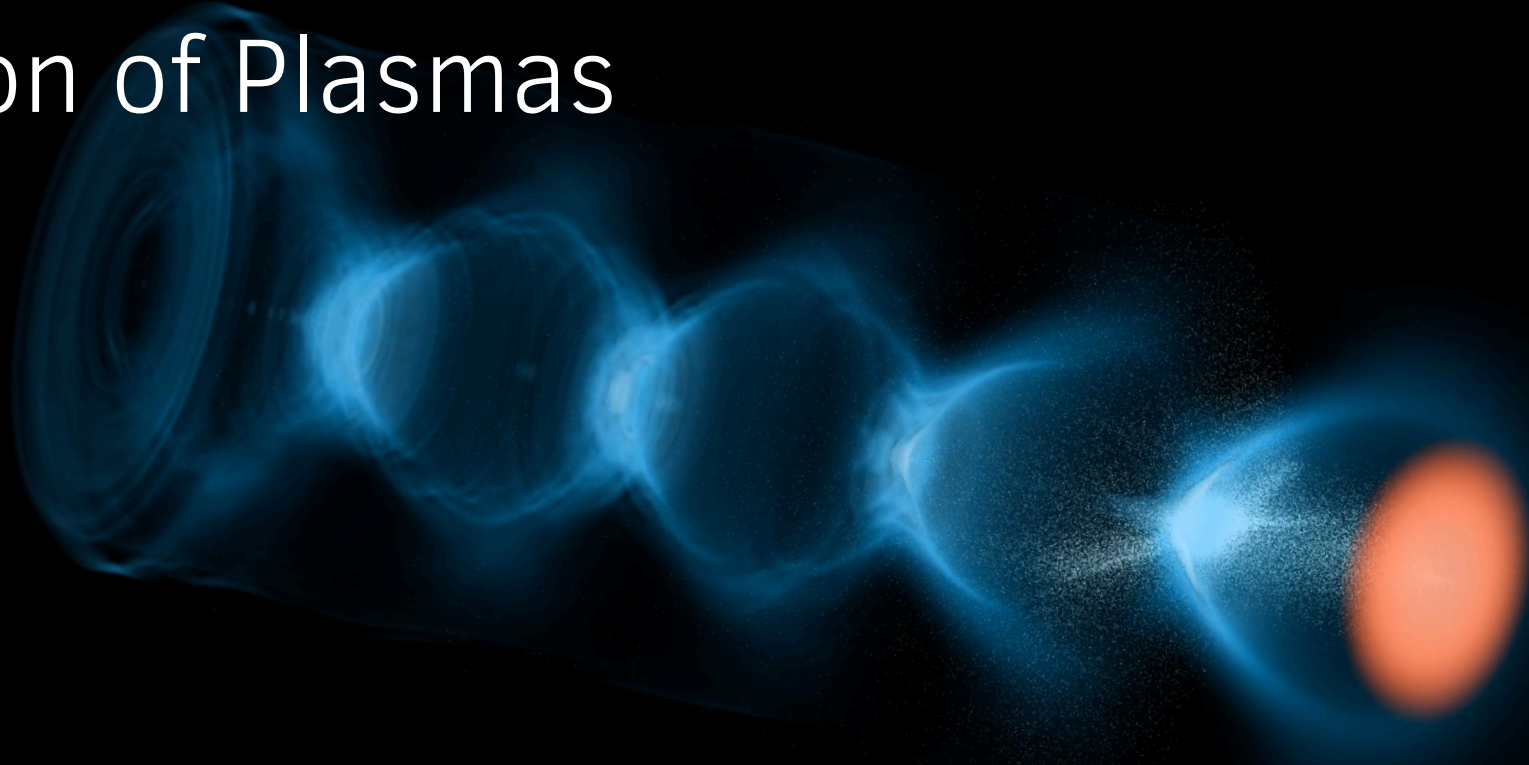


# The Particle-In-Cell (PIC) Simulation of Plasmas



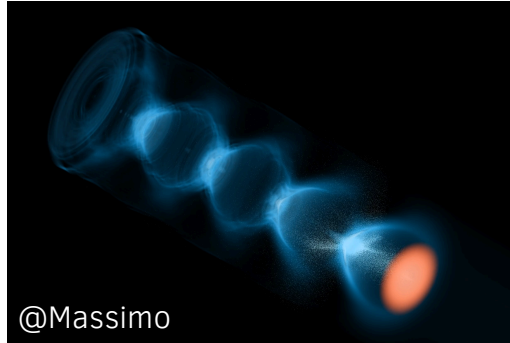
Mickael Grech, LULI

SCIPAC • Atelier Calcul • IJClab, Orsay • October 16-18, 2024

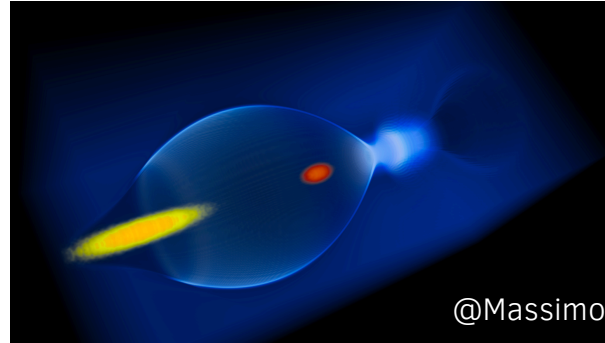
# The Particle-In-Cell (PIC) simulation of plasmas

from Laboratory Plasmas ...

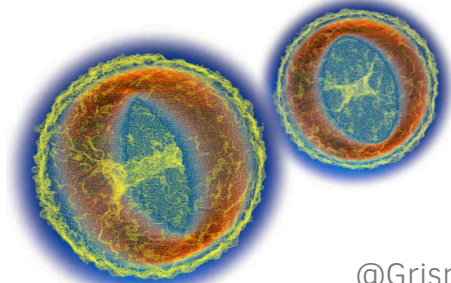
Laser Wakefield Acceleration



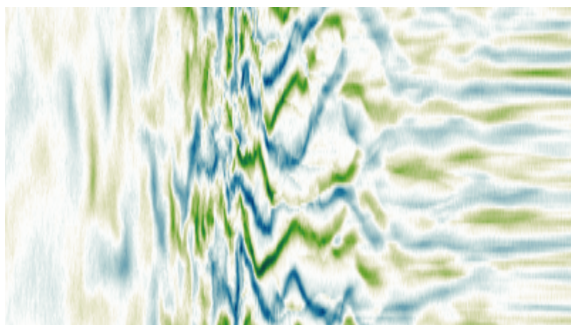
Particle-driven Wakefield Acceleration



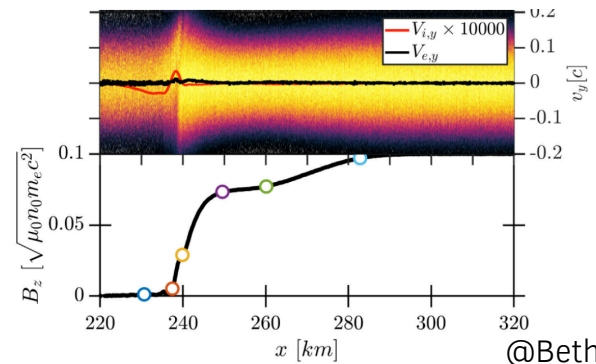
Beam-beam collision



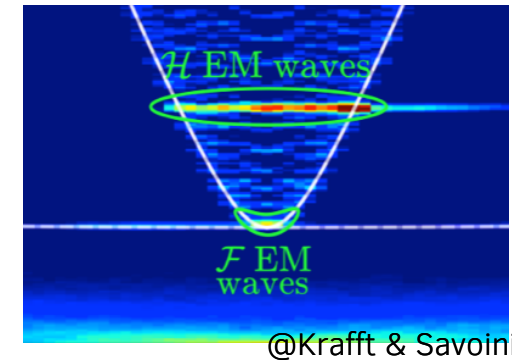
Collisionless shocks & Cosmic Rays



Comet boundaries



Solar radio-burst



... to Space & Astrophysical Plasmas

# The Vlasov-Maxwell description

Vlasov Eq - Species of the plasma

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$

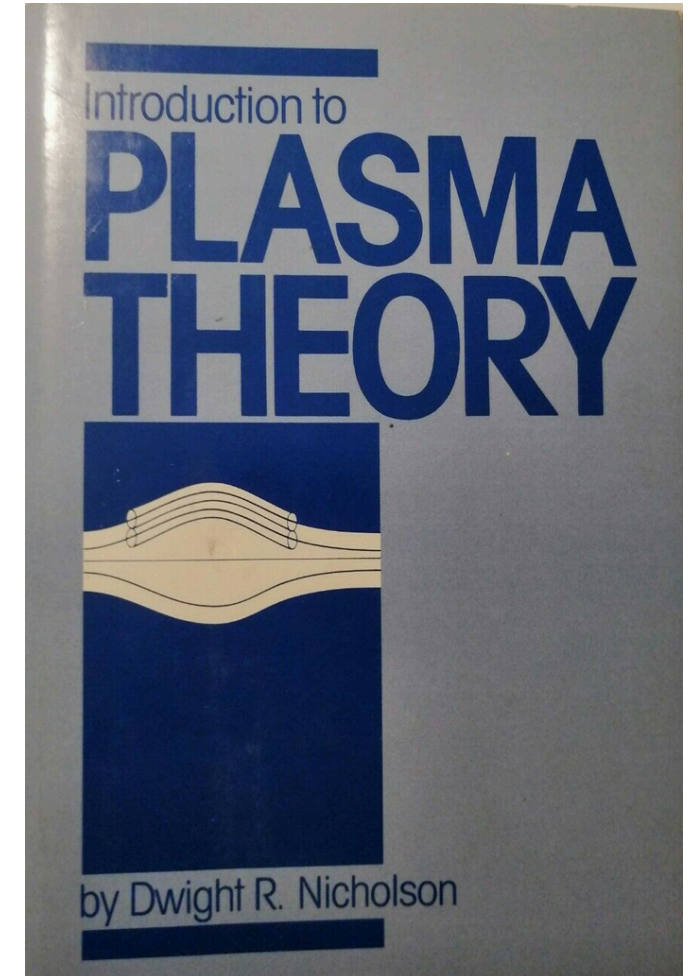
$$\mathbf{F}_L = q_s \left( \mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right)$$
$$\rho(t, \mathbf{x}) = \int d\mathbf{p} f_s(t, \mathbf{x}, \mathbf{p})$$
$$\mathbf{J}(t, \mathbf{x}) = q_s \int d\mathbf{p} \frac{\mathbf{p}}{m_s \gamma} f_s(t, \mathbf{x}, \mathbf{p})$$

Maxwell Eqs - Electromagnetic Fields

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \partial_t \mathbf{E} = -\frac{1}{\epsilon_0} \mathbf{J} + c^2 \nabla \times \mathbf{B}$$

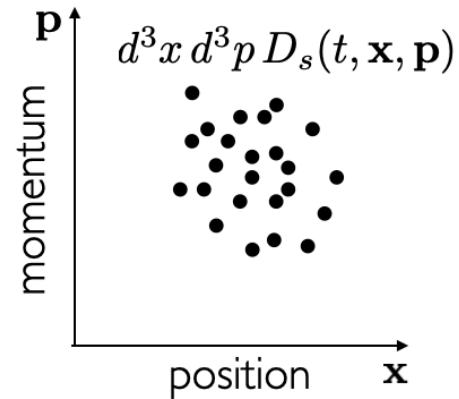
$$\nabla \cdot \mathbf{B} = 0 \quad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Introduction to plasma theory,  
D. R. Nicholson (1983)



# Vlasov equation in a nutshell

Starting point : Klimontovich's exact picture



If the exact state of the system is known at a time  $t_0$

$$D_s(t_0, \mathbf{x}, \mathbf{p}) = \sum_p \delta(\mathbf{x} - \mathbf{x}_p(t_0)) \delta(\mathbf{p} - \mathbf{p}_p(t_0))$$

the evolution of the system at later times is known exactly and satisfies the

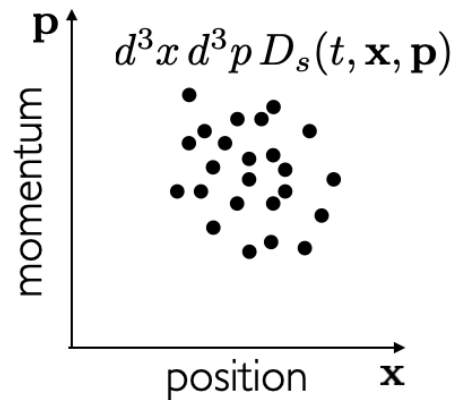
Klimontovich equation:

$$\partial_t D_s + \mathbf{p} \cdot \nabla D_s + q_s (\mathbf{E}_{\text{tot}} + \mathbf{v} \times \mathbf{B}_{\text{tot}}) \cdot \nabla_p D_s = 0.$$



# Vlasov equation in a nutshell

Starting point : Klimontovich's exact picture



If the exact state of the system is known at a time  $t_0$

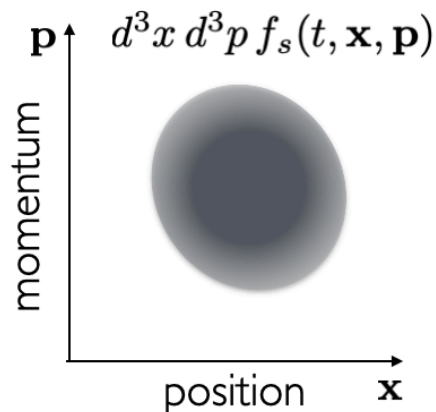
$$D_s(t_0, \mathbf{x}, \mathbf{p}) = \sum_p \delta(\mathbf{x} - \mathbf{x}_p(t_0)) \delta(\mathbf{p} - \mathbf{p}_p(t_0))$$

the evolution of the system at later times is known exactly and satisfies the

Klimontovich equation:

$$\partial_t D_s + \mathbf{p} \cdot \nabla D_s + q_s (\mathbf{E}_{\text{tot}} + \mathbf{v} \times \mathbf{B}_{\text{tot}}) \cdot \nabla_p D_s = 0.$$

Ensemble averaging : towards the plasma kinetic equation and Vlasov equation



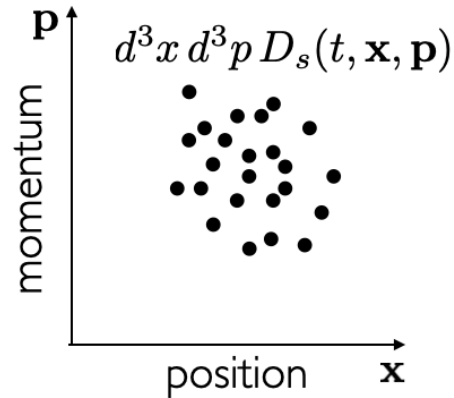
$\overbrace{D_s(\mathbf{x}, \mathbf{v}, t)}^{\text{total}}$	$=$	$\overbrace{f_s(\mathbf{x}, \mathbf{v}, t)}^{\text{smooth, average}}$	$+$	$\overbrace{\delta D_s(\mathbf{x}, \mathbf{v}, t)}^{\text{microscopic fluctuations}}$
$\mathbf{E}_{\text{tot}}(\mathbf{x}, t)$	$=$	$\mathbf{E}(\mathbf{x}, t)$	$+$	$\delta \mathbf{E}(\mathbf{x}, t)$
$\mathbf{B}_{\text{tot}}(\mathbf{x}, t)$	$=$	$\mathbf{B}(\mathbf{x}, t)$	$+$	$\delta \mathbf{B}(\mathbf{x}, t)$

Plugging this in Klimontovich equation and ensemble averaging leads:

$$\partial_t f_s + \mathbf{p} \cdot \nabla f_s + q_s (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_p f_s = -q_s \langle (\delta \mathbf{E} + \mathbf{v} \times \delta \mathbf{B}) \delta f_s \rangle.$$

# Vlasov equation in a nutshell

Starting point : Klimontovich's exact picture



If the exact state of the system is known at a time  $t_0$

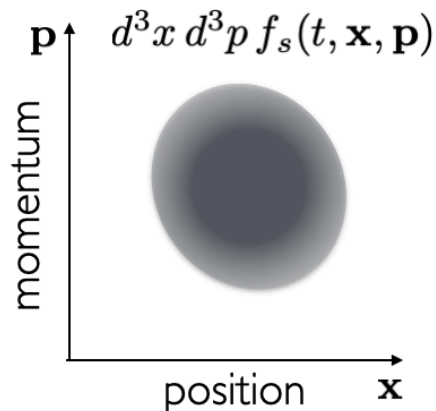
$$D_s(t_0, \mathbf{x}, \mathbf{p}) = \sum_p \delta(\mathbf{x} - \mathbf{x}_p(t_0)) \delta(\mathbf{p} - \mathbf{p}_p(t_0))$$

the evolution of the system at later times is known exactly and satisfies the

Klimontovich equation:

$$\partial_t D_s + \mathbf{p} \cdot \nabla D_s + q_s (\mathbf{E}_{\text{tot}} + \mathbf{v} \times \mathbf{B}_{\text{tot}}) \cdot \nabla_p D_s = 0.$$

Ensemble averaging : towards the plasma kinetic equation and Vlasov equation



$\overbrace{D_s(\mathbf{x}, \mathbf{v}, t)}^{\text{total}}$	$=$	$\overbrace{f_s(\mathbf{x}, \mathbf{v}, t)}^{\text{smooth, average}}$	$+$	$\overbrace{\delta D_s(\mathbf{x}, \mathbf{v}, t)}^{\text{microscopic fluctuations}}$
$\mathbf{E}_{\text{tot}}(\mathbf{x}, t)$	$=$	$\mathbf{E}(\mathbf{x}, t)$	$+$	$\delta \mathbf{E}(\mathbf{x}, t)$
$\mathbf{B}_{\text{tot}}(\mathbf{x}, t)$	$=$	$\mathbf{B}(\mathbf{x}, t)$	$+$	$\delta \mathbf{B}(\mathbf{x}, t)$

Plugging this in Klimontovich equation and ensemble averaging leads:

$$\partial_t f_s + \mathbf{p} \cdot \nabla f_s + q_s (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_p f_s = -q_s \langle (\delta \mathbf{E} + \mathbf{v} \times \delta \mathbf{B}) \delta f_s \rangle.$$

**collective behavior**

**microscopic/collisions**

# Solving Vlasov-Maxwell with the PIC method

Vlasov Eq - Species of the plasma

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$

$$\mathbf{F}_L = q_s \left( \mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right)$$

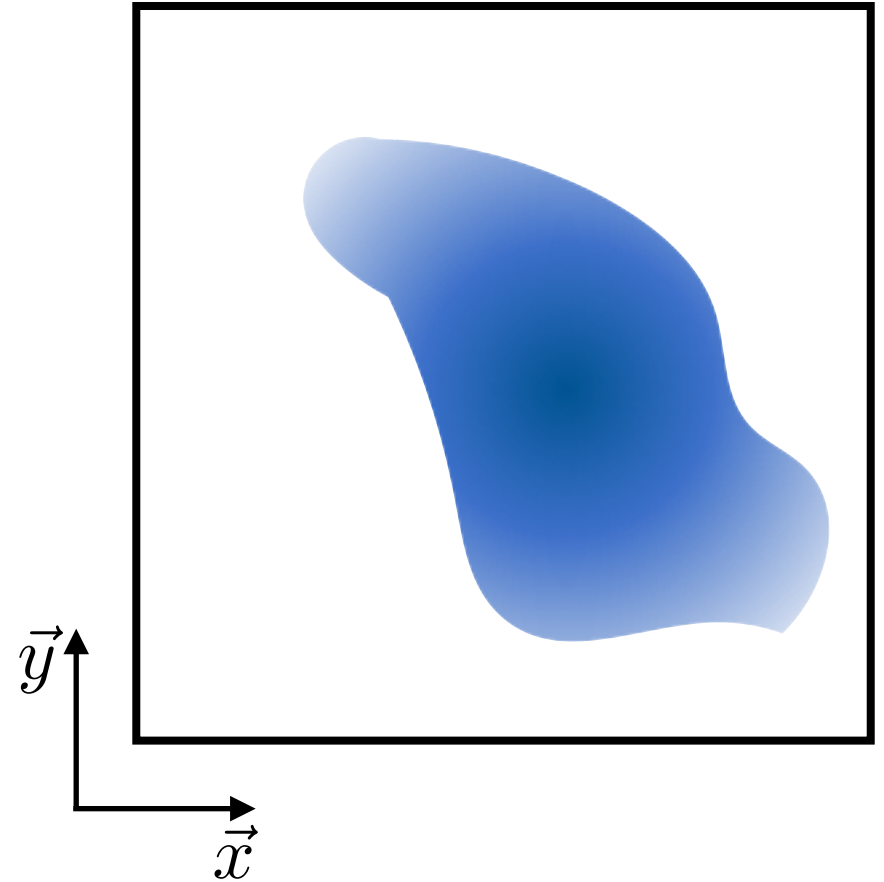
$$\rho(t, \mathbf{x}) = \int d\mathbf{p} f_s(t, \mathbf{x}, \mathbf{p})$$

$$\mathbf{J}(t, \mathbf{x}) = q_s \int d\mathbf{p} \frac{\mathbf{p}}{m_s \gamma} f_s(t, \mathbf{x}, \mathbf{p})$$

Maxwell Eqs - Electromagnetic Fields

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \partial_t \mathbf{E} = -\frac{1}{\epsilon_0} \mathbf{J} + c^2 \nabla \times \mathbf{B}$$

$$\nabla \cdot \mathbf{B} = 0 \quad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$



# Solving Vlasov-Maxwell with the PIC method

Vlasov Eq - Species of the plasma

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$

$$\mathbf{F}_L = q_s \left( \mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right)$$

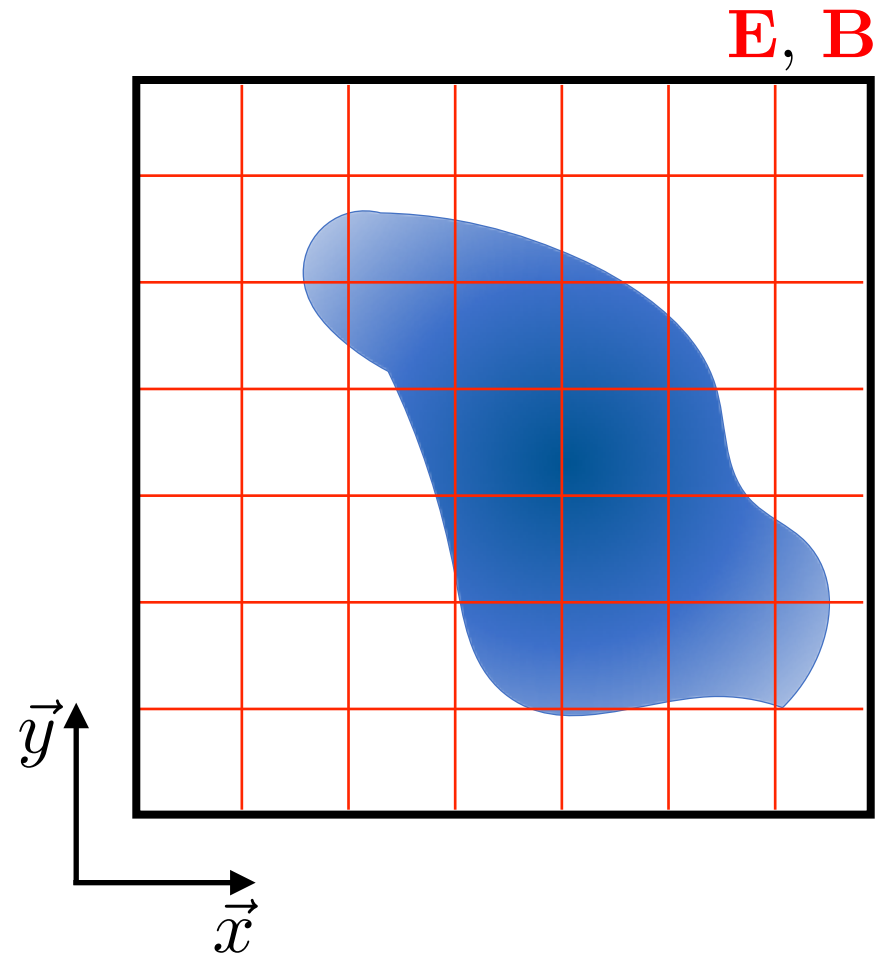
$$\rho(t, \mathbf{x}) = \int d\mathbf{p} f_s(t, \mathbf{x}, \mathbf{p})$$

$$\mathbf{J}(t, \mathbf{x}) = q_s \int d\mathbf{p} \frac{\mathbf{p}}{m_s \gamma} f_s(t, \mathbf{x}, \mathbf{p})$$

Maxwell Eqs - Electromagnetic Fields

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \partial_t \mathbf{E} = -\frac{1}{\epsilon_0} \mathbf{J} + c^2 \nabla \times \mathbf{B}$$

$$\nabla \cdot \mathbf{B} = 0 \quad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$





# Solving Vlasov-Maxwell with the PIC method

Vlasov Eq - Species of the plasma

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$

$$\mathbf{F}_L = q_s \left( \mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right)$$

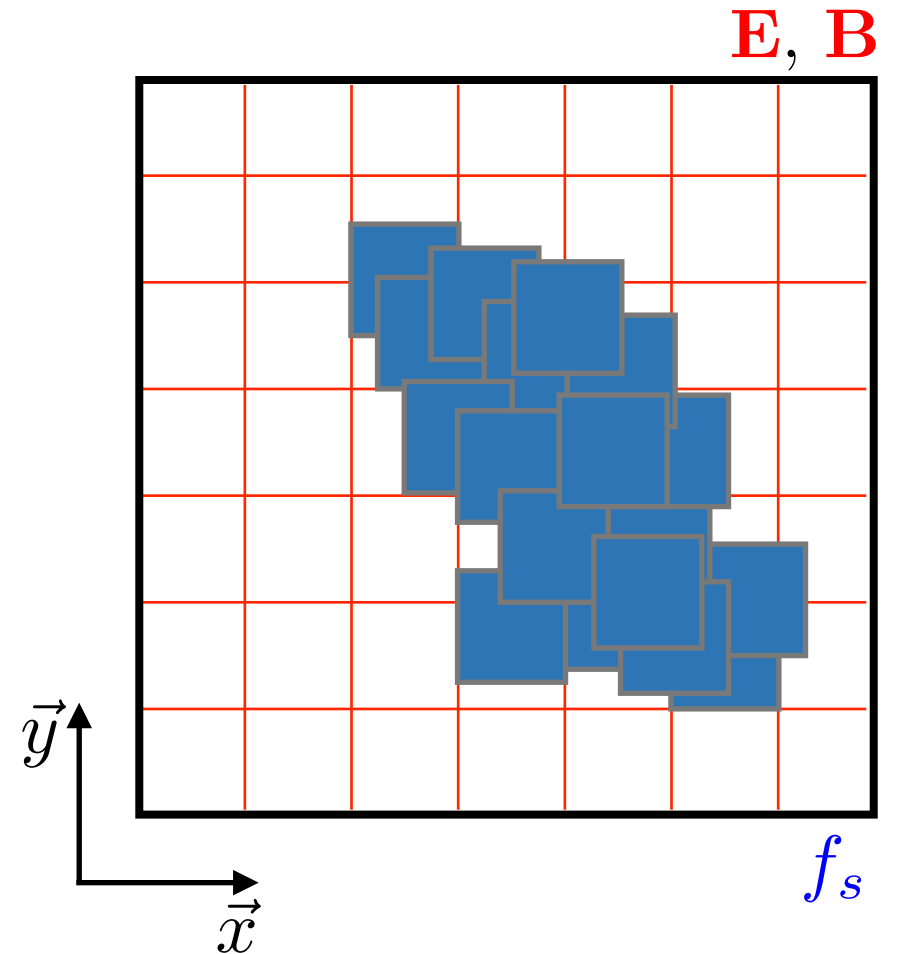
$$\rho(t, \mathbf{x}) = \int d\mathbf{p} f_s(t, \mathbf{x}, \mathbf{p})$$

$$\mathbf{J}(t, \mathbf{x}) = q_s \int d\mathbf{p} \frac{\mathbf{p}}{m_s \gamma} f_s(t, \mathbf{x}, \mathbf{p})$$

Maxwell Eqs - Electromagnetic Fields

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \partial_t \mathbf{E} = -\frac{1}{\epsilon_0} \mathbf{J} + c^2 \nabla \times \mathbf{B}$$

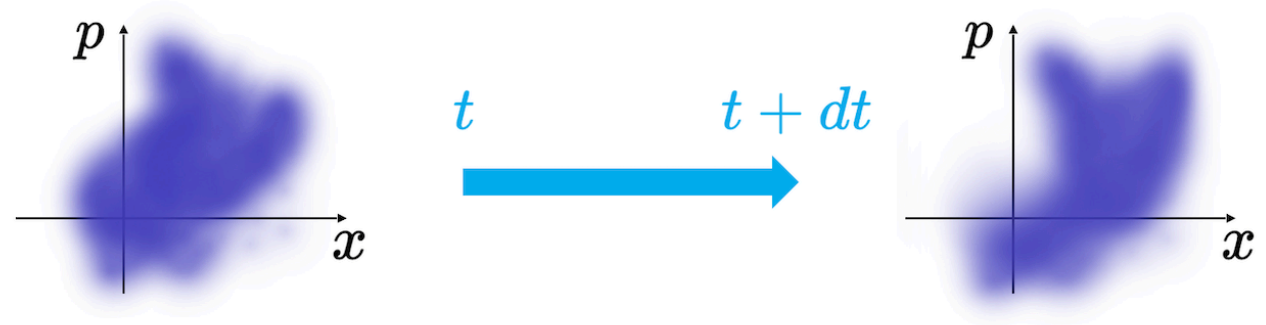
$$\nabla \cdot \mathbf{B} = 0 \quad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$



# The PIC ansatz

We want to solve a discretized version of Vlasov equation:

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$



# The PIC ansatz

We want to solve a discretized version of Vlasov equation:

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$



This is done using approximating the distribution function at all times as a sum of [quasi/macro/pseudo-particles](#):

$$f_s(t, \mathbf{x}, \mathbf{p}) \stackrel{!}{=} \sum_{p=1}^N w_p S(\mathbf{x} - \mathbf{x}_p(t)) \delta(\mathbf{p} - \mathbf{p}_p(t))$$

numerical weight      shape-function

# The PIC ansatz

We want to solve a discretized version of Vlasov equation:

$$\partial_t f_s + \frac{\mathbf{p}}{m_s \gamma} \cdot \nabla f_s + \mathbf{F}_L \cdot \nabla_p f_s = 0$$



This is done using approximating the distribution function at all times as a sum of **quasi/macro/pseudo-particles**:

$$f_s(t, \mathbf{x}, \mathbf{p}) \stackrel{!}{=} \sum_{p=1}^N w_p S(\mathbf{x} - \mathbf{x}_p(t)) \delta(\mathbf{p} - \mathbf{p}_p(t))$$

numerical weight      shape-function

Solving Vlasov then reduces to solving the (relativistic) **equations of motion** of the macro-particles:

$$\partial_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p) \quad \text{with} \quad (\mathbf{E}, \mathbf{B})_p \equiv \int d\mathbf{x} (\mathbf{E}, \mathbf{B})(\mathbf{x}) S(\mathbf{x} - \mathbf{x}_p)$$



# The PIC loop: (1) Field interpolation

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

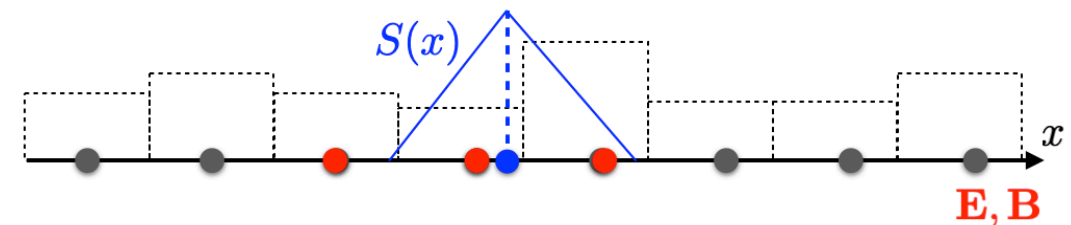
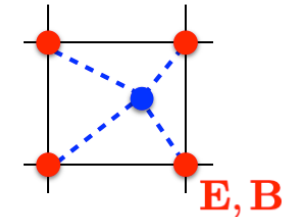
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

$$(\mathbf{E}, \mathbf{B})_p \equiv \int d\mathbf{x} (\mathbf{E}, \mathbf{B})(\mathbf{x}) S(\mathbf{x} - \mathbf{x}_p)$$



# The PIC loop: (2) Particle pusher

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

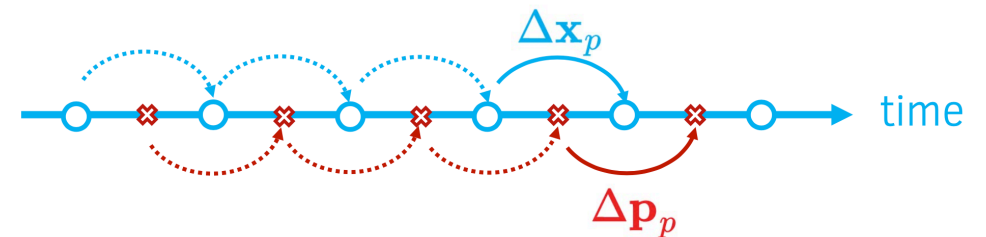
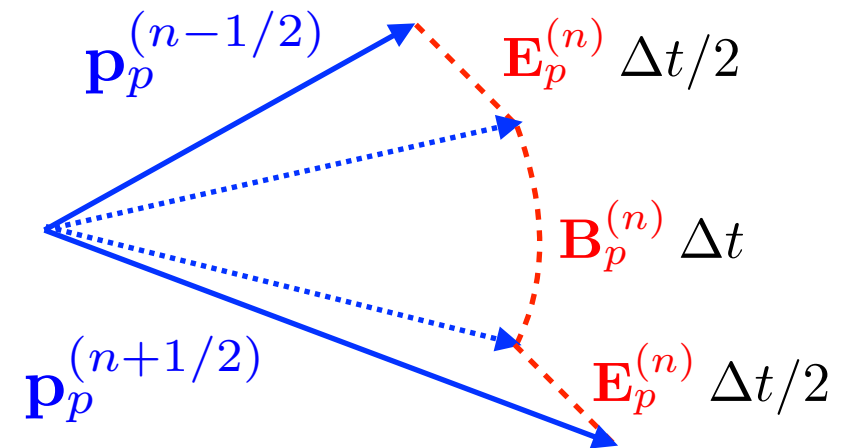
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

the Boris (leap-frog) Pusher



# The PIC loop: (3) Current deposition

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

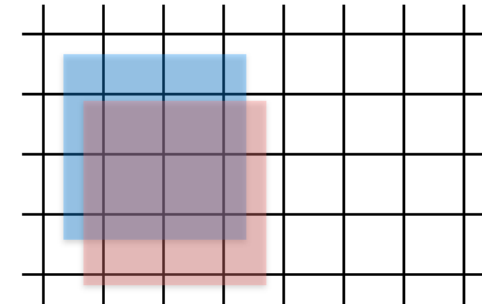
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

the Esirkepov method for  
charge-conserving current deposition



$$(J_{x,p})_{i+\frac{1}{2},j}^{(n+\frac{1}{2})} = (J_{x,p})_{i-\frac{1}{2},j}^{(n+\frac{1}{2})} + q_s w_p \frac{\Delta x}{\Delta t} (W_x)_{i+\frac{1}{2},j}^{(n+\frac{1}{2})}$$
$$(J_{y,p})_{i,j+\frac{1}{2}}^{(n+\frac{1}{2})} = (J_{y,p})_{i,j-\frac{1}{2}}^{(n+\frac{1}{2})} + q_s w_p \frac{\Delta y}{\Delta t} (W_y)_{j,i+\frac{1}{2}}^{(n+\frac{1}{2})}$$

Esirkepov, *Comp. Phys. Comm.* **135**, 144 (2001)

# The PIC loop: (4) Maxwell solver

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

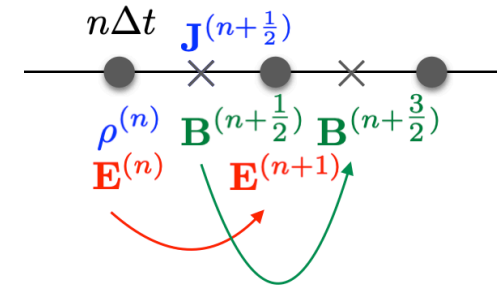
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

the Finite-Domain Time-Difference  
Maxwell solver



$$\left( \frac{dE_y}{dt} \right)_i^{(n)} = -(J_y)_i^{(n)} + \frac{(B_z)_{i+1/2}^{(n+1/2)} - (B_z)_{i-1/2}^{(n+1/2)}}{\Delta x}$$

Taflove, *Computation Electrodynamics* (2005)  
Nuter et al., *EPJD* **68**, 1 (2014)



# The PIC loop: (4) Maxwell solver

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

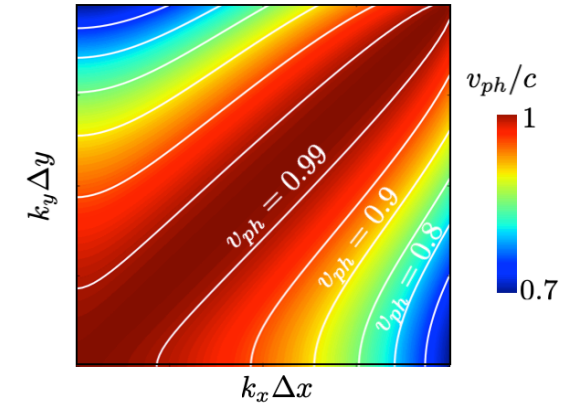
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

the Finite-Domain Time-Difference  
Maxwell solver



Taflove, *Computation Electrodynamics* (2005)  
Nuter et al., *EPJD* **68**, 1 (2014)

# The PIC loop: (4) Maxwell solver

Field Interpolation

$$[\mathbf{E}, \mathbf{B}] \rightarrow [\mathbf{E}_p, \mathbf{B}_p]$$

Maxwell Solver

$$\partial_t \mathbf{E} = -\mathbf{J} + \nabla \times \mathbf{B}$$

$$\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

Particle Pusher

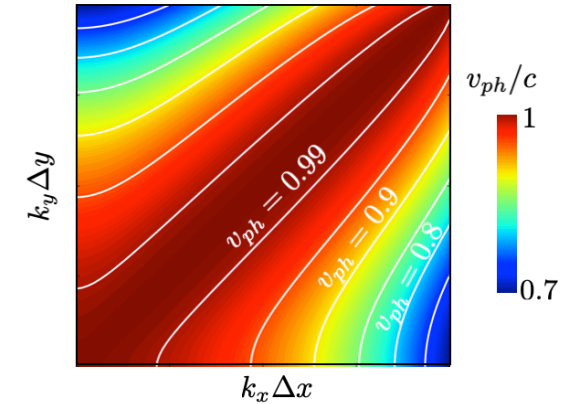
$$d_t \mathbf{p}_p = q_s (\mathbf{E}_p + \mathbf{v} \times \mathbf{B}_p)$$

$$d_t \mathbf{x}_p = \mathbf{p}_p / (m_s \gamma)$$

Current Projection

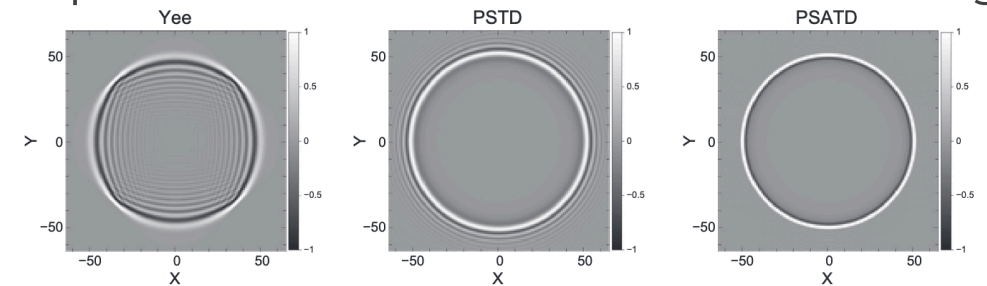
$$[\mathbf{x}_p, \mathbf{p}_p] \rightarrow [\rho_s, \mathbf{J}_s]$$

the Finite-Domain Time-Difference  
Maxwell solver



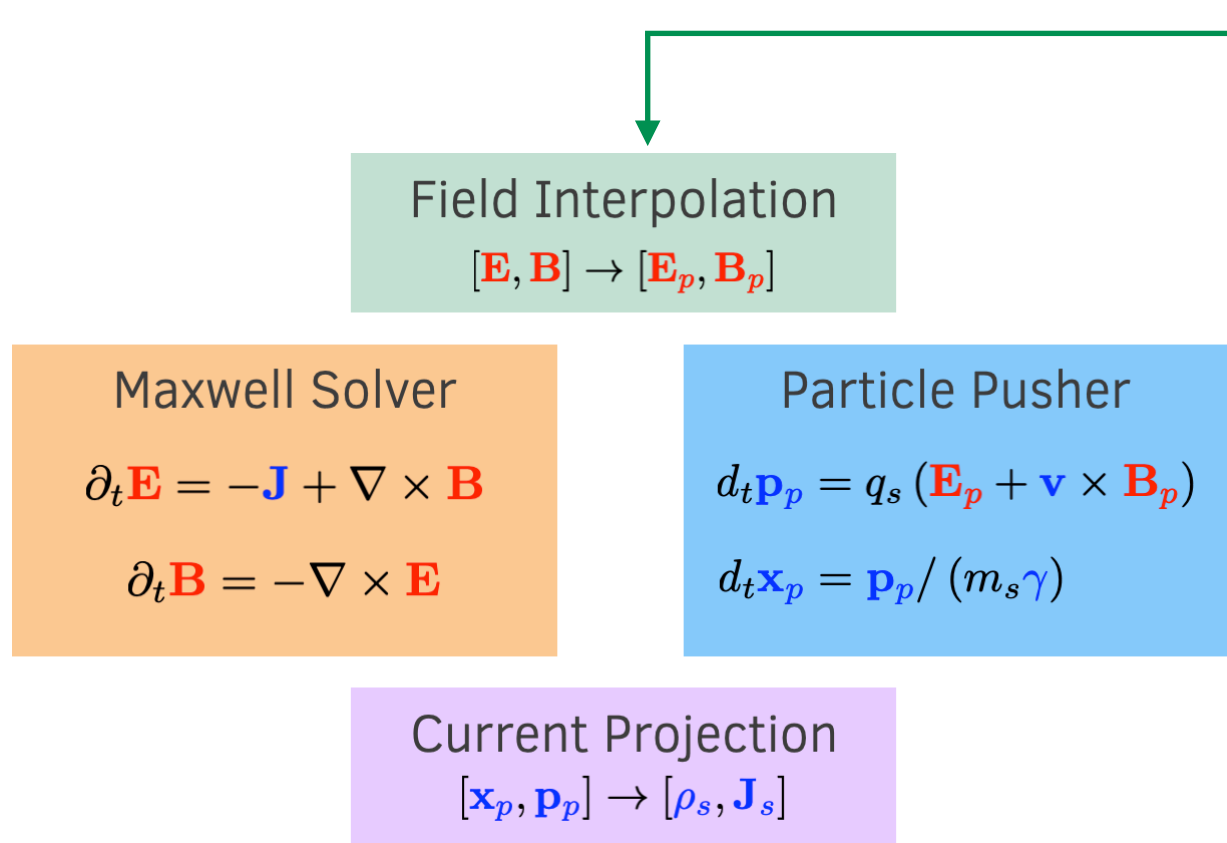
Taflove, *Computation Electrodynamics* (2005)  
Nuter et al., *EPJD* **68**, 1 (2014)

Spectral methods are also interesting



Vay et al., *J. Comp. Phys.* **243**, 260 (2013)

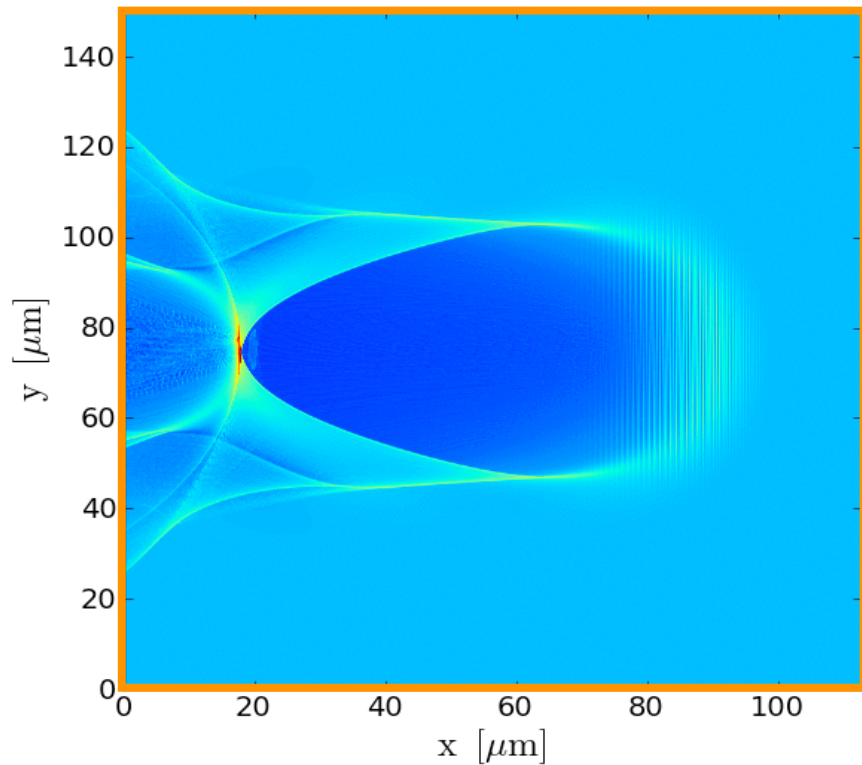
# Around the PIC loop: Initialization, Boundary Conditions, Parallelization



**Initialization:** entering the PIC loop

- initialize your macro-particles
- solve Poisson equation(s)
- add divergence-free fields

## My Simulation (LWFA)



### Initialization: entering the PIC loop

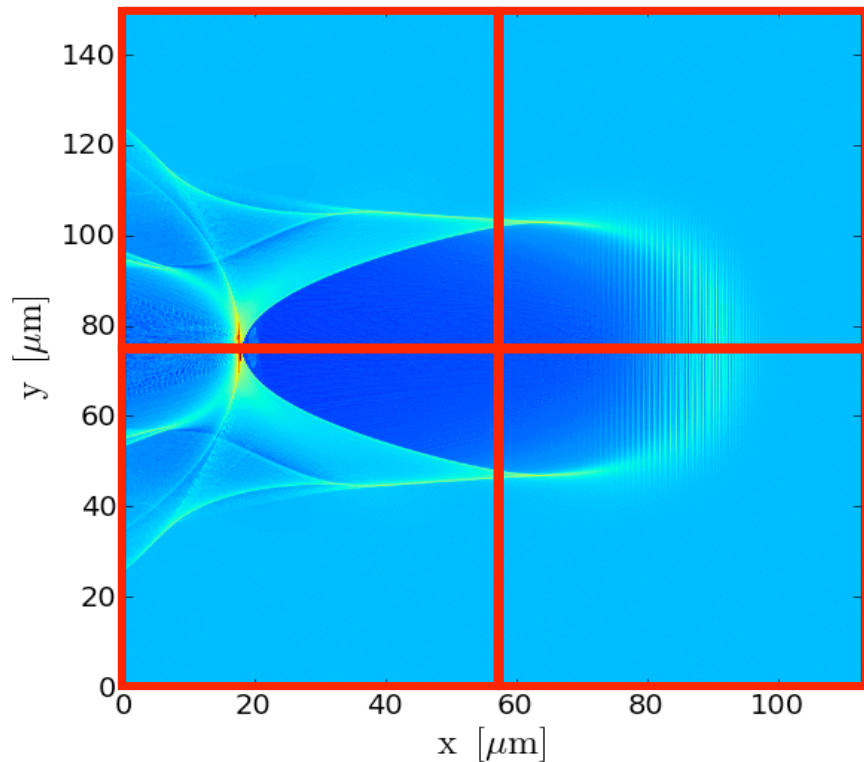
- initialize your macro-particles
- solve Poisson equation(s)
- add divergence-free fields

### Boundary conditions:

- on electromagnetic fields (reflecting, injecting, absorbing, etc)
- on macro-particles (reflecting, injecting, thermalizing, etc)



## My Simulation (LWFA)



### Initialization: entering the PIC loop

- initialize your macro-particles
- solve Poisson equation(s)
- add divergence-free fields

### Boundary conditions:

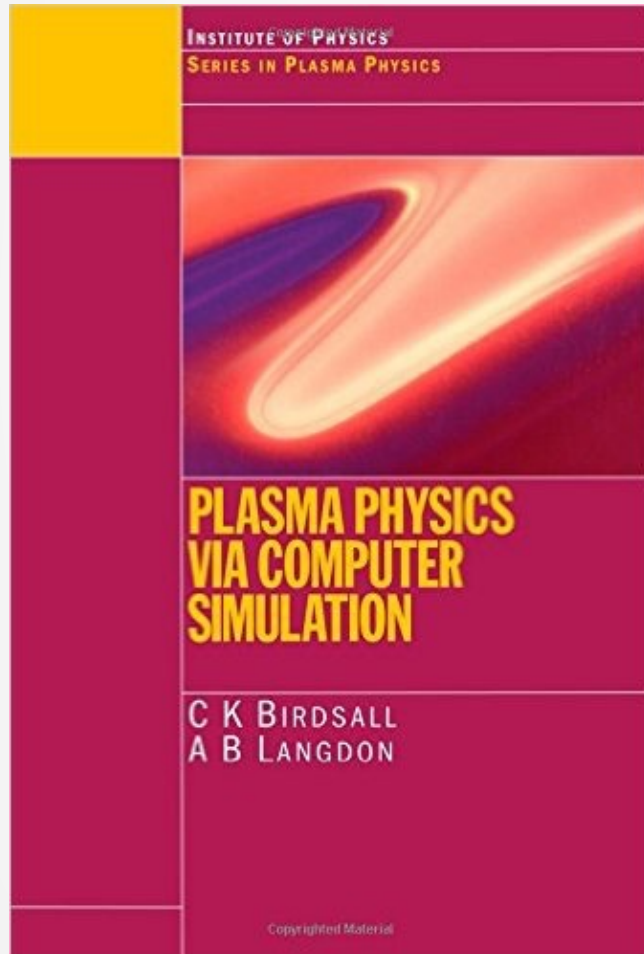
- on electromagnetic fields (reflecting, injecting, absorbing, etc)
- on macro-particles (reflecting, injecting, thermalizing, etc)

### Parallelization:

- domain decomposition
- (dynamic) load balancing
- vectorization & much more

# References to go beyond this introduction

Plasma Physics via Comp. Sim.  
Birdsall & Langdon



Series of online lectures by Paolo Ricci  
EPFL lectures, available on youtube



- Plasma time scales
- The simulation approaches
- Numerical solution of the Vlasov equation: the PIC method

The Smilei PIC code website  
Extensive documentation & tutorials

**Smilei** Overview Understand Use

### Quasi-particles

The *Particle-In-Cell* method owes its name to the discretization of the distribution function  $f_s$ , as a sum of  $N_s$  quasi-particles (also referred to as *super-particles* or *macro-particles*):

$$f_s(t, \mathbf{x}, \mathbf{p}) = \sum_{p=1}^{N_s} \frac{w_p}{V_c} S(\mathbf{x} - \mathbf{x}_p(t)) \delta(\mathbf{p} - \mathbf{p}_p(t)), \quad (5)$$

where  $w_p$  is a quasi-particle weight,  $\mathbf{x}_p$  is its position,  $\mathbf{p}_p$  is its momentum,  $V_c$  is the hypervolume of the cell,  $S$  is the shape-function of all quasi-particles, and  $\delta$  is the Dirac distribution.

In PIC codes, Vlasov's equation (1) is integrated along the continuous trajectories of these quasi-particles, while Maxwell's equations (3) are solved on a discrete spatial grid, the spaces between consecutive grid points being referred to as *cells*. Injecting the discrete distribution function of Eq. (5) in Vlasov's equation (1), multiplying the result by  $\mathbf{p}$  and integrating over all  $\mathbf{p}$  and over the volume of the quasi-particles, leads to the relativistic equations of motion of individual quasi-particles:

$$\frac{d\mathbf{x}_p}{dt} = \frac{\mathbf{u}_p}{\gamma_p}$$
$$\frac{d\mathbf{u}_p}{dt} = r_s \left( \mathbf{E}_p + \frac{\mathbf{u}_p}{\gamma_p} \times \mathbf{B}_p \right),$$

where  $r_s = q_s/m_s$  is the charge-over-mass ratio (for species  $s$ ),  $\mathbf{u}_p = \mathbf{p}_p/m_s$  is the reduced momentum and  $\gamma_p = \sqrt{1 + \mathbf{u}_p^2}$  is the Lorentz factor.

### Time and space discretization

Maxwell's equations are solved here using the *Finite Difference Time Domain (FDTD)* approach as well as refined methods based on this algorithm. In these methods, the electromagnetic fields are discretized onto

**tutorials** PIC basics Performances Advanced

### Physical configuration

Download the two input files `weibel_1d.py` and `two_stream_1d.py`.

In both simulations, a plasma with density  $n_0$  is initialized ( $n_0 = 1$ ). This makes code units equal to plasma units, i.e. times are normalized to the inverse of the electron plasma frequency  $\omega_{pe} = \sqrt{e^2 n_0 / (\epsilon_0 m_e)}$ , distances to the electron skin-depth  $c/\omega_{pe}$ , etc...

Ions are frozen during the whole simulation and just provide a neutralizing background. Two electron species are initialized with density  $n_0/2$  and a mean velocity  $\pm v_0$ .

### Check input file and run the simulation

The first step is to check that your *input files* are correct. To do so, you will run (locally) **Smilei** in test mode:

```
./smilei_test weibel_1d.py
./smilei_test two_stream_1d.py
```

If your simulation *input files* are correct, you can run the simulations. Before going to the analysis, check your *logs*.

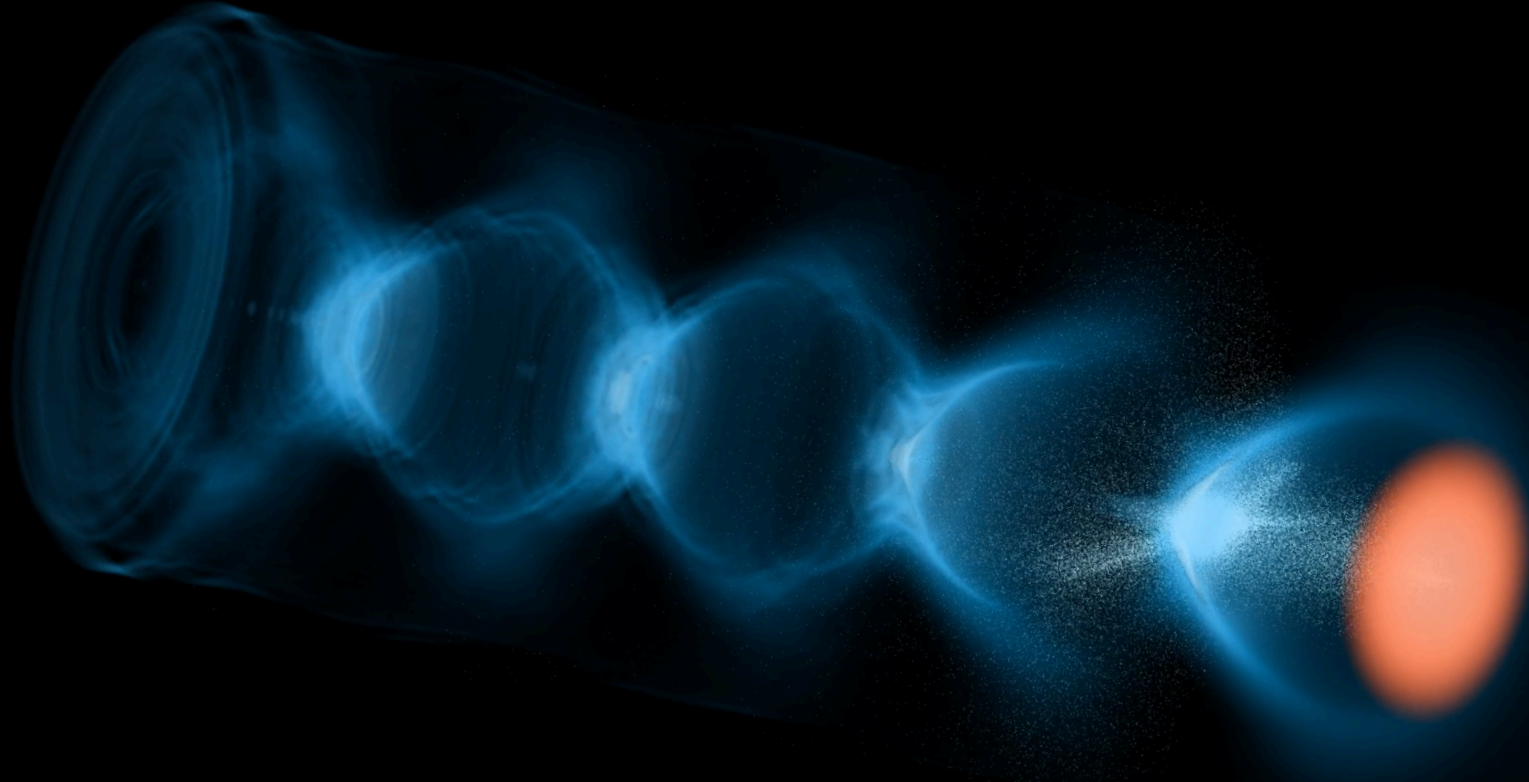
### Weibel instability: analysis

In an **ipython** terminal, open the simulation:

```
S = happi.Open('/path/to/your/simulation/weibel_1d')
```

The **streak** function of **happi** can plot any 1D diagnostic as a function of time. Let's look at the time evolution of the total the current density  $J_z$  and the magnetic field  $B_y$ :

Thanks for your attention!



[mickael.grech@polytechnique.edu](mailto:mickael.grech@polytechnique.edu)