Smilei) Workshop 2022

Particle-In-Cell (PIC) basics

Guillaume Bouchard
CNRS | IN2P3 | Laboratoire Leprince Ringuet – March 2022

Smilei... What for...

The goal is to **simulate a plasma** dominated by **collective effects** and neglecting Particle-Particle interactions :

1. Charged particles (a lot...);
2. Collisionless;
3. Collective/Mean fields.

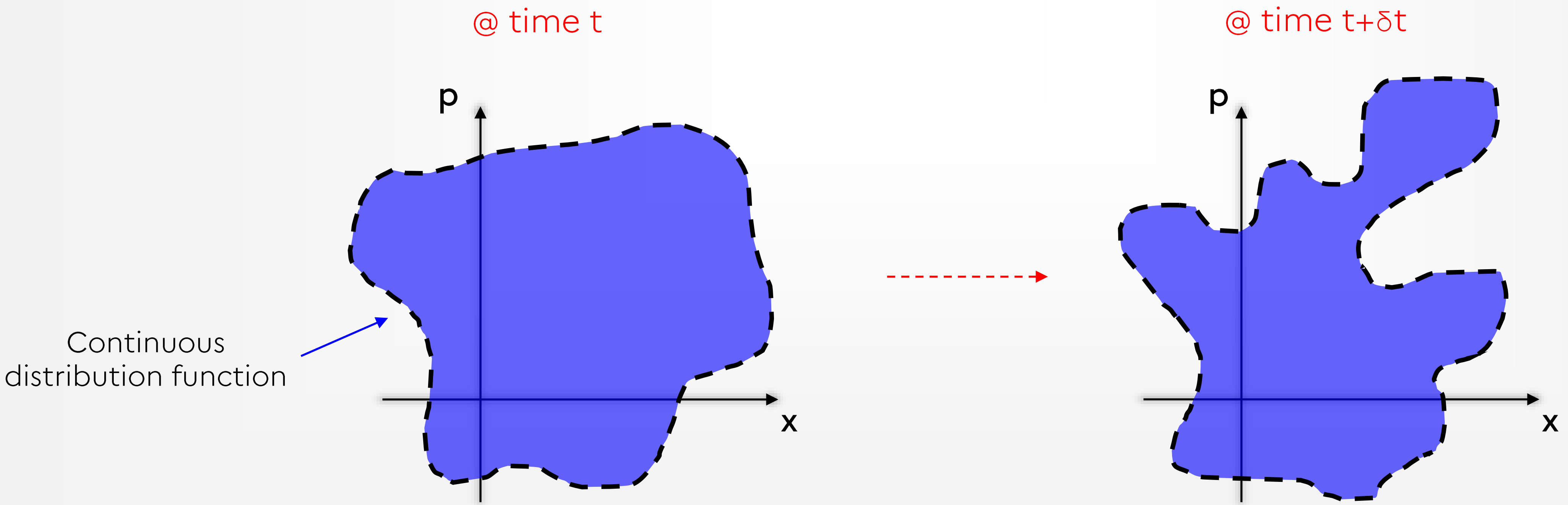
Describe the evolution of the system in time ≡ Describe the evolution of the distribution function – **Vlasov Equation** :

$$\frac{\partial f_s}{\partial t} + \frac{\partial f_s}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} + \frac{\partial f_s}{\partial \mathbf{p}} \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = 0$$

Vlasov equation

Describe the evolution of the system in time ≡ Describe the evolution of the distribution function – **Vlasov Equation** :

$$\frac{\partial f_s}{\partial t} + \frac{\partial f_s}{\partial \mathbf{x}} \frac{\mathbf{p}}{m_s \gamma} + \frac{\partial f_s}{\partial \mathbf{p}} (\mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B}) = 0$$



Outline

I. Macro-particles and Maxwell-Vlasov system

II. Units

III. The PIC loop

   i. Numerical integration of Maxwell's equations

   ii. Numerical integration of macro-particles dynamics

   iii. Fields gathering on macro-particles

   iv. Current deposition on the grid

IV. Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundary conditions

Smilei) Workshop 2022

I. Macro-particles and Maxwell-Vlasov system

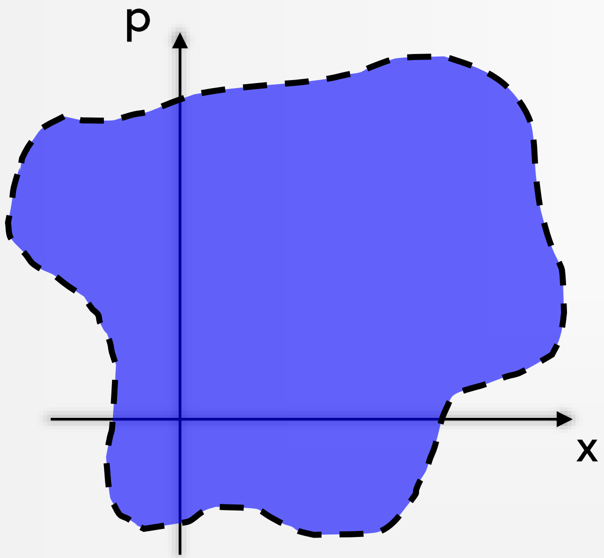
Macro-particles

Vlasov Equation :

Partial differential equation (PDE) in
(3 (**x**) + 3 (**p**) = 6 dimensions) x N_s

$$\frac{\partial f_s}{\partial t} + \frac{\partial f_s}{\partial \mathbf{x}} \frac{\mathbf{p}}{m_s \gamma} + \frac{\partial f_s}{\partial \mathbf{p}} \left(\mathbf{E} + \frac{\mathbf{p}}{m_s \gamma} \times \mathbf{B} \right) = 0$$

Particle-in-Cell ansatz :



Shape function

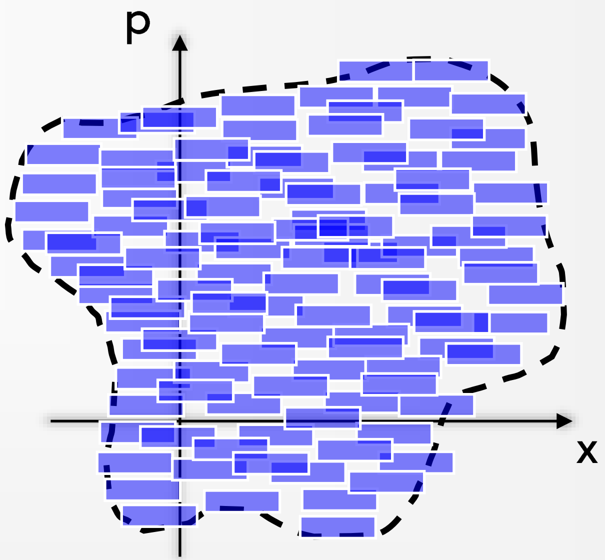
$$f_s(\mathbf{x}, \mathbf{p}, t) = \sum_{p=1}^{N} w_p S(\mathbf{x} - \mathbf{x}_p(t)) \delta(\mathbf{p} - \mathbf{p}_p(t))$$

weight

Dirac-distribution

$$f_s(\mathbf{x}, \mathbf{p}, t) = \sum_{p=1}^{N}$$

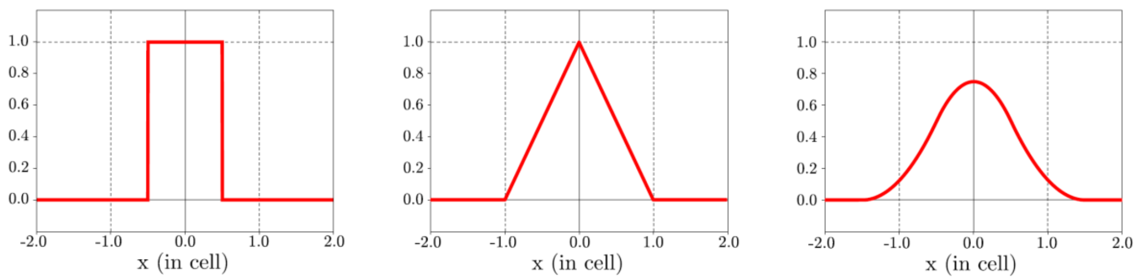
a "macro-particle"



Macro-particles shape function

$$f_s(\mathbf{x},\mathbf{p},t) = \sum_{p=1}^{N} \underbrace{w_p}_{\text{weight}} \overbrace{S(\mathbf{x}-\mathbf{x}_p(t))}^{\text{Shape function}} \underbrace{\delta(\mathbf{p}-\mathbf{p}_p(t))}_{\text{Dirac-distribution}}$$

Shape functions



(a) Ordre 0 : « Top Hat » (b) Ordre 1 : « Triangle » (c) Ordre N : « Spline »

Macro-particles motion

Shape function

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

weight

Dirac-distribution

$$\frac{\partial f_s}{\partial t} + \frac{\partial f_s}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} + \frac{\partial f_s}{\partial \mathbf{p}}\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = 0$$

We can show that injecting the macro-particle ansatz in the Vlasov equation leads to resolve [Derouillat *et al*, 2018]:

$$\mathrm{d}_t\mathbf{p}_p = \mathbf{F}_p \longrightarrow \mathrm{d}_t\mathbf{p}_p = q_s(\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p)$$

$$\mathrm{d}_t\mathbf{x}_p = \mathbf{p}_p/m_p\gamma_p$$

Maxwell's equations

Charge see E.M. fields → Charge move in space with E.M. field → E.M. fields change due to the particle motion

We have to solve Maxwell's equations in order to update fields and resolve macro-particles dynamics :

$$(a)\ \text{Maxwell-Gauss :}\quad \nabla \cdot \boldsymbol{E} = \frac{\varrho}{\varepsilon_0} \qquad (c)\ \text{Maxwell-Faraday :}\quad \nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

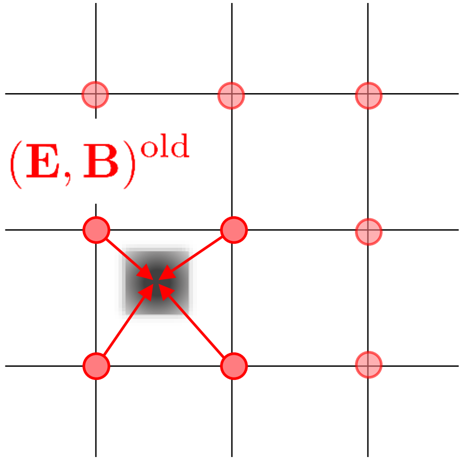
$$(b)\ \text{Maxwell-Thomson :}\quad \nabla \cdot \boldsymbol{B} = 0 \qquad (d)\ \text{Maxwell-Ampère :}\quad \nabla \times \boldsymbol{B} = \mu_0 \mathcal{J} + \mu_0 \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

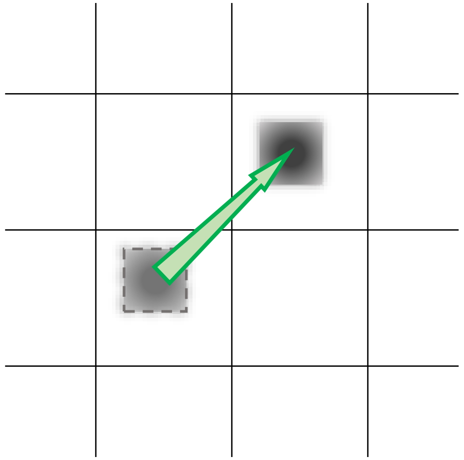
With :

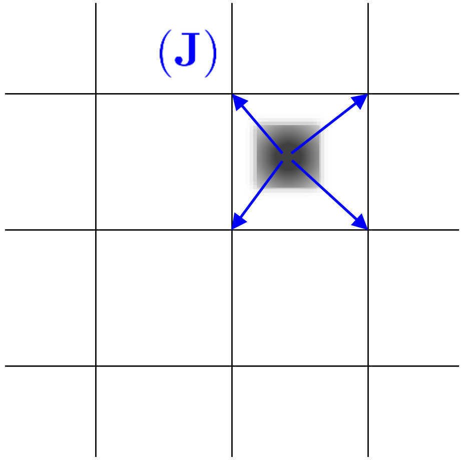
$$\varrho = \sum_s q_s \int_{-\infty}^{+\infty} f_s(\mathbf{x}, \mathbf{p}, t)\, \mathrm{d}\mathbf{p} \qquad \mathcal{J} = \sum_s q_s \int_{-\infty}^{+\infty} \frac{\mathbf{p}}{m_s \gamma} f_s(\mathbf{x}, \mathbf{p}, t)\, \mathrm{d}\mathbf{p}$$

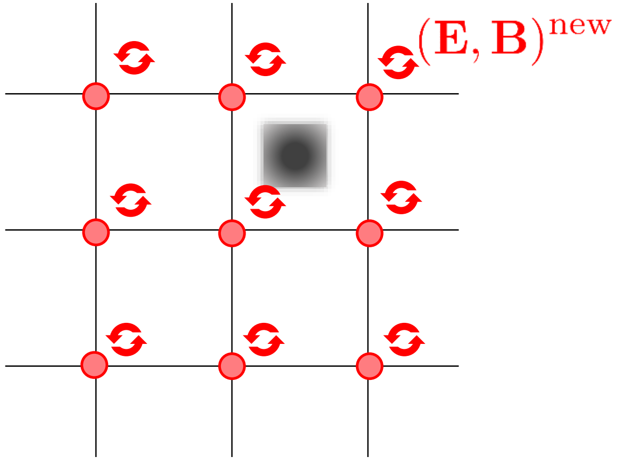
Macro-particles and Maxwell-Vlasov system

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD









$$(\mathbf{E}_g,\mathbf{B}_g) \to (\mathbf{E}_p,\mathbf{B}_p)$$

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

$$\partial_t \mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

$$(\mathbf{J}_p) \to (\mathbf{J}_g)$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial B}{\partial t}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0\varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

Smilei) Workshop 2022

II. Units

Units

Natural units for electrodynamics :

Units of velocity	c
Units of charge	e
Units of mass	m_e
Units of momentum	$m_e\,c$
Units of energy, temperature	$m_e c^2$

If we fix a unit for time ω_r^{-1} :

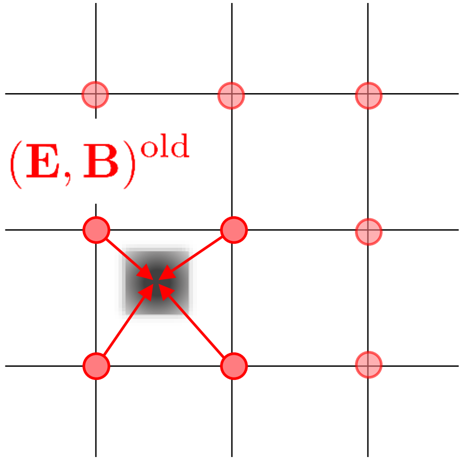
Units of length	c/ω_r
Units of number density	$n_r = \epsilon_0\, m_e\, \omega_r^2/e^2$
Units of current density	$e\, c\, n_r$
Units of pressure	$m_e\, c^2\, n_r$
Units of electric field	$m_e\, c\, \omega_r/e$
Units of magnetic field	$m_e\, \omega_r/e$
Units of Poynting flux	$m_e\, c^3\, n_r/2$

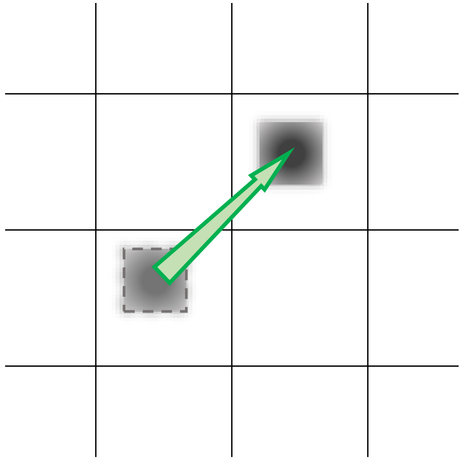
ω_r^{-1} is not defined *a priori* and acts as a scaling factor except for some feature like ionization or collision module for which it has to be defined in SI units.

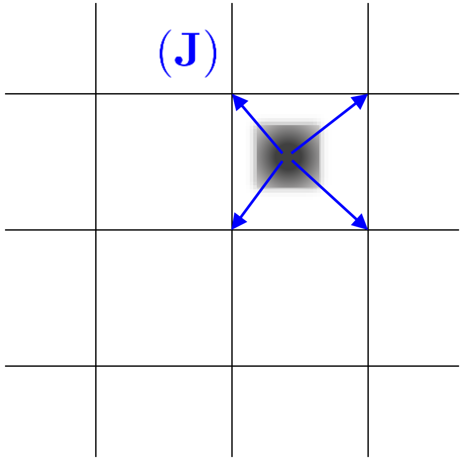
When considering the irradiation of a plasma by a laser with angular frequency ω_0 it can be convenient to use $\omega_r = \omega_0$

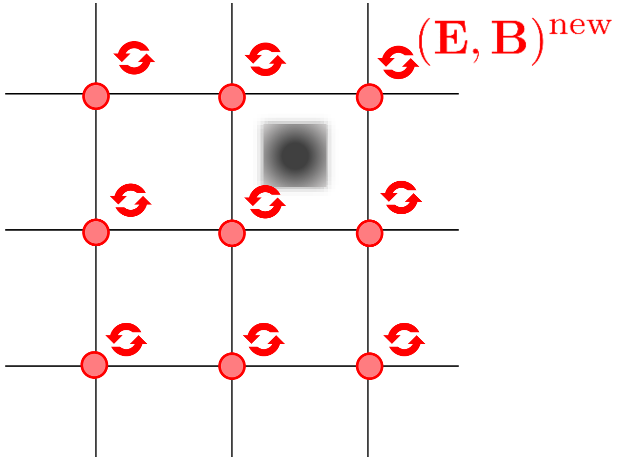
Maxwell's equations integration (Maxwell Solver)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD









$$(\mathbf{E}_g,\mathbf{B}_g)\rightarrow(\mathbf{E}_p,\mathbf{B}_p)$$

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

$$\partial_t\mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

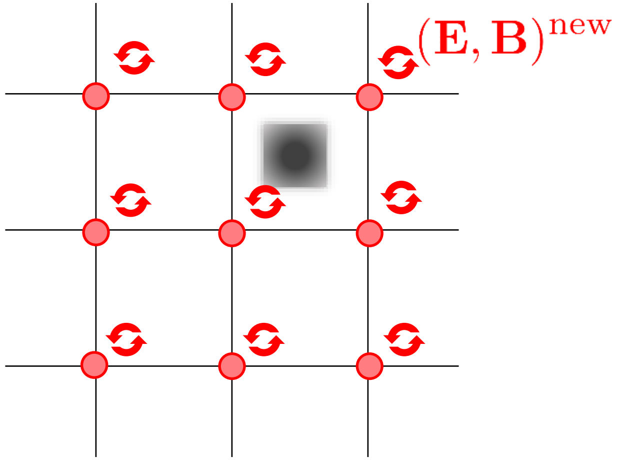
$$(\mathbf{J}_p)\rightarrow(\mathbf{J}_g)$$

$$\nabla\times\boldsymbol{E} = -\frac{\partial B}{\partial t}$$

$$\nabla\times\boldsymbol{B} = \mu_0\boldsymbol{\mathcal{J}}+\mu_0\varepsilon_0\frac{\partial\boldsymbol{E}}{\partial t}$$

Maxwell's equations integration (Maxwell Solver)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD



$$(\mathbf{E}_g,\mathbf{B}_g) \rightarrow (\mathbf{E}_p,\mathbf{B}_p)$$

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

$$\partial_t \mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

$$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0 \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

Smilei) Workshop 2022

III. The PIC loop

Maxwell's equations integration (Maxwell Solver)

Maxwell's equations integration (Maxwell Solver)

We have to solve 2 Maxwell's equations to allow fields to evolve in time :

$$\nabla \times \mathbf{B} - \mu_0 \boldsymbol{\mathcal{J}} = \mu_0 \varepsilon_0 \, \partial_t \, \mathbf{E}$$

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

One popular method to solve these coupled partial equations is to use Finite-Difference Time-Domain (FDTD) where differentials operators are approximated by finite differences :

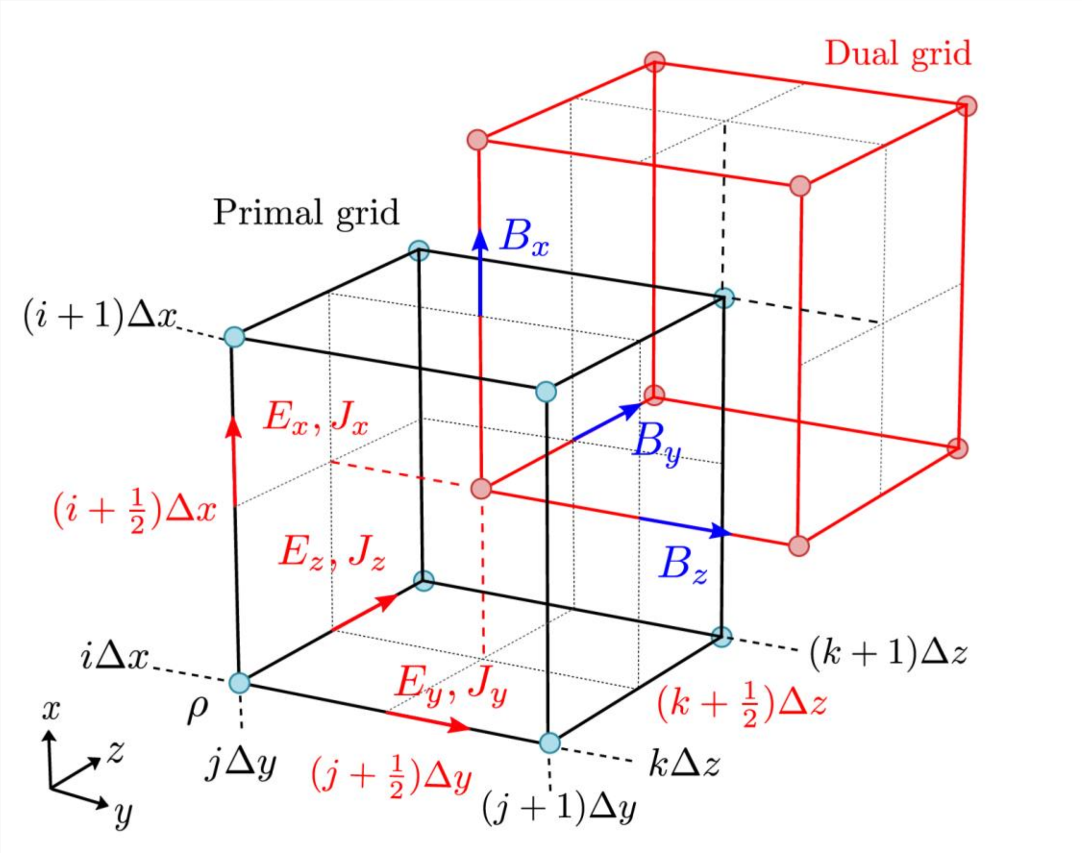
$$\partial_i \mathbf{F} \rightarrow D_i \mathbf{F} = \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta_i} \qquad \partial_t \mathbf{F} \rightarrow D_t \mathbf{F} = \frac{\mathbf{F}^{n+1/2} - \mathbf{F}^{n-1/2}}{\Delta_t}$$

To ensure consistency and stability the difference is centered with the RHS :

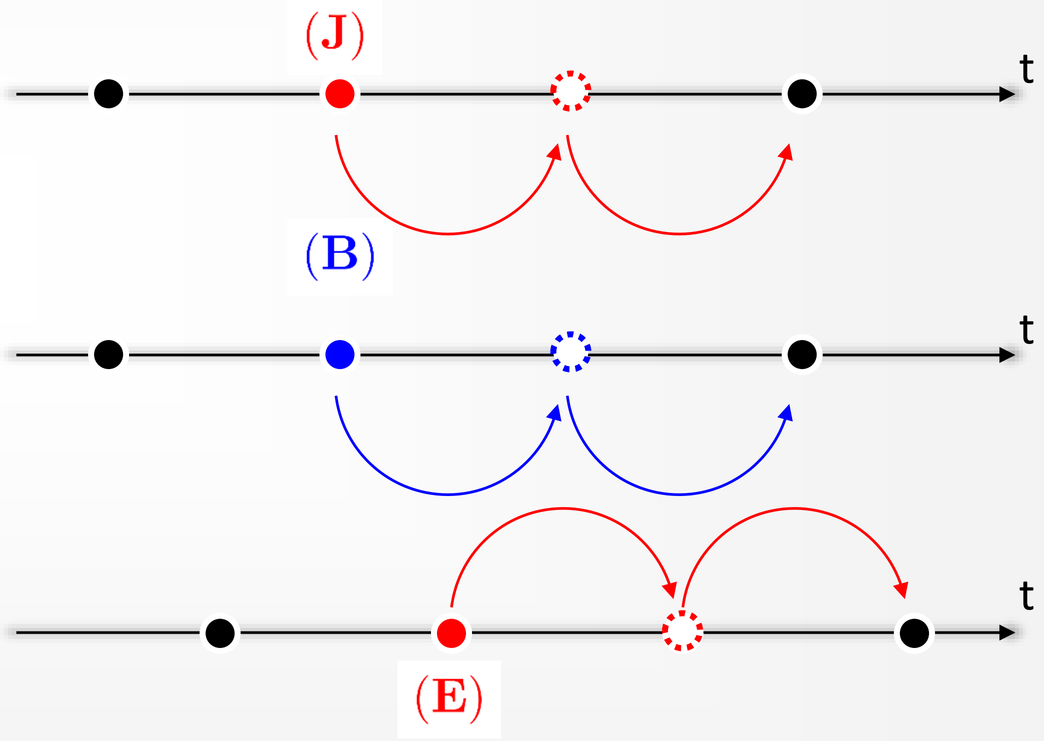
$$\partial_i \mathbf{F} = \mathbf{G} \rightarrow \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta_i} = \mathbf{G}_i \qquad \partial_t \mathbf{F} = \mathbf{G} \rightarrow \frac{\mathbf{F}^{n+1/2} - \mathbf{F}^{n-1/2}}{\Delta_t} = \mathbf{G}_n$$

Maxwell's equations integration (Maxwell Solver)

Finally in 3D, we can represent a cell (Yee cell) like :

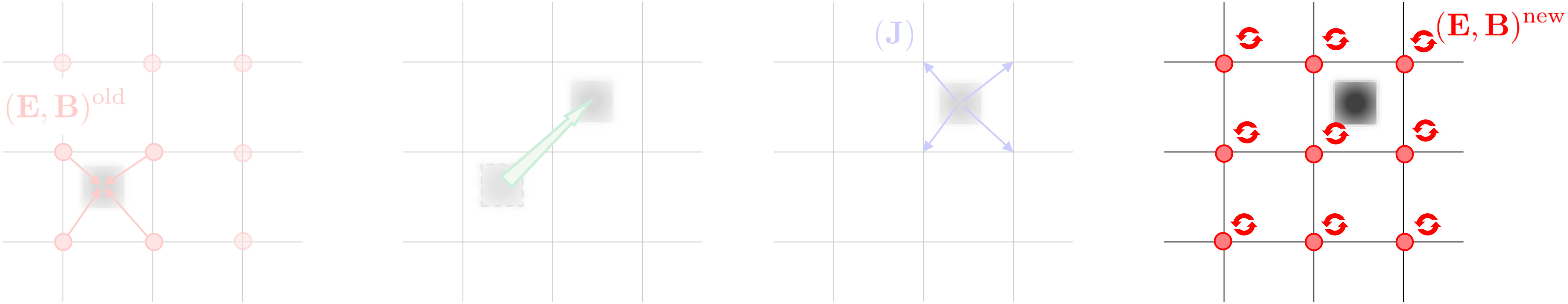


The grid of the discretized time is like :



PIC loop : Numerical integration of macro-particles' dynamics (Pusher)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD



$(\mathbf{E}_g, \mathbf{B}_g) \rightarrow (\mathbf{E}_p, \mathbf{B}_p)$

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

$\partial_t \mathbf{x}_p = \mathbf{p}_p/\gamma_p$

$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0 \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

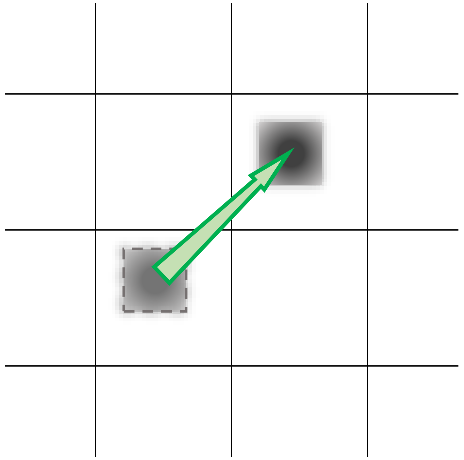
PIC loop : Numerical integration of macro-particles’ dynamics (Pusher)

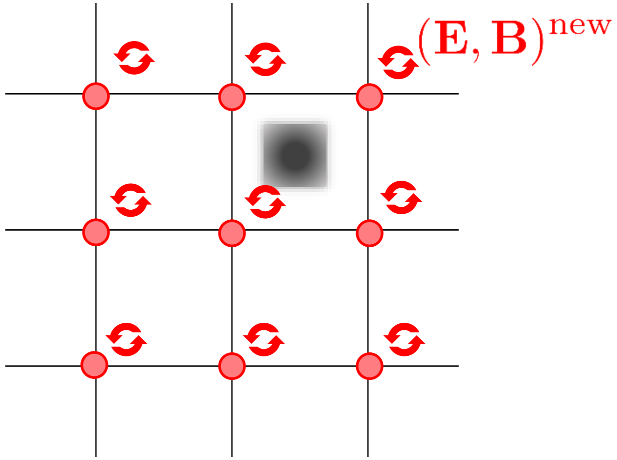
Gathering E.M. Fields on each particle’s position

Numerical integration of macro-particles’ dynamics equation using Lorentz’s force

Deposition of current density (and charge eventually)

Numerical integration of Maxwell’s equation with FDTD





$$(\mathbf{E}_g, \mathbf{B}_g) \rightarrow (\mathbf{E}_p, \mathbf{B}_p)$$

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

$$\partial_t \mathbf{x}_p = \mathbf{p}_p / \gamma_p$$

$$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial B}{\partial t}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0 \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

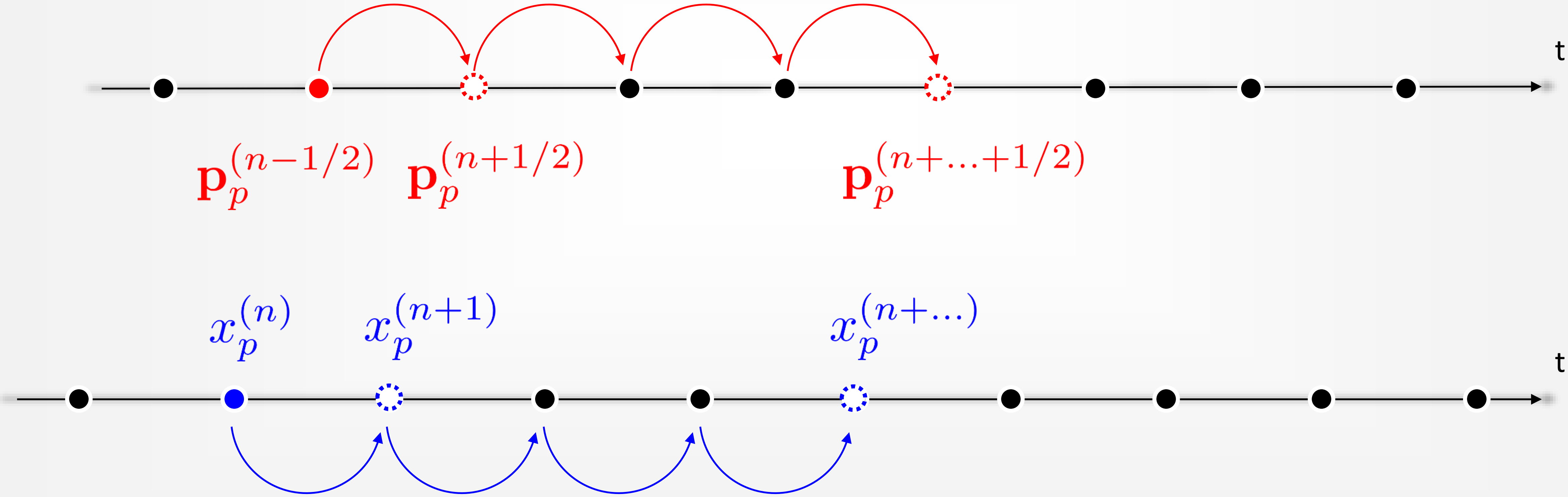
Smilei) Workshop 2022

III. The PIC loop

Numerical integration of macro-particles' dynamics (Pusher)

PIC loop : Numerical integration of macro-particles' dynamics (Pusher)

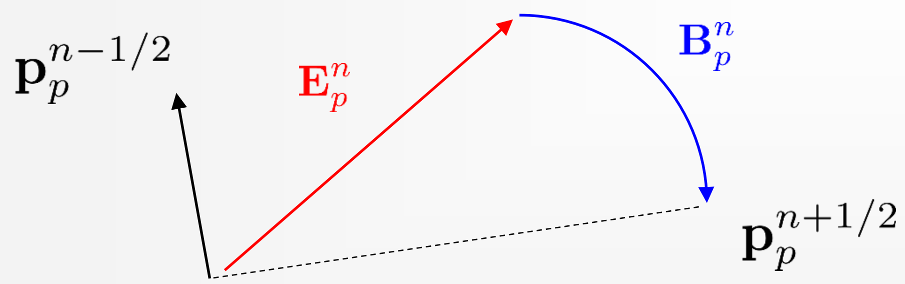
$$\mathrm{d}_t \mathbf{p}_p = q_s(\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p) \qquad \mathrm{d}_t \mathbf{x}_p = \mathbf{p}_p / m_p \gamma_p$$



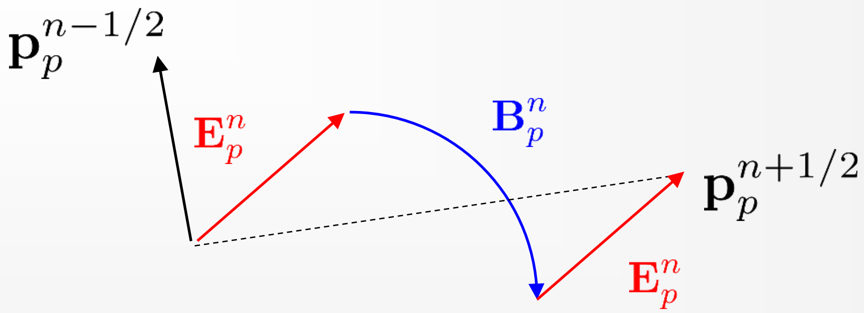
PIC loop : Numerical integration of macro-particles' dynamics (Pusher)

$$\mathrm{d}_t \mathbf{p}_p = q_s(\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p) \xrightarrow[\text{discretization}]{\text{time's}} \frac{\mathbf{p}_p^{n+1/2} - \mathbf{p}_p^{n-1/2}}{\Delta_t} = q_s\left(\mathbf{E}_p^n + \frac{\mathbf{p}_p^{n+1/2} + \mathbf{p}_p^{n-1/2}}{2\gamma^n} \times \mathbf{B}_p^n\right)$$

"Naïve" Pusher



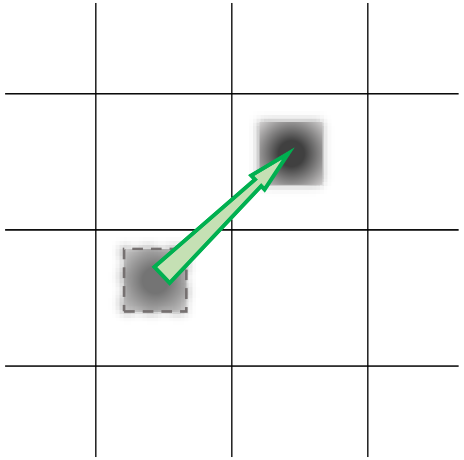
Boris Pusher



PIC loop : Gathering E.M. Fields on each particle's position (Interpolator)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD

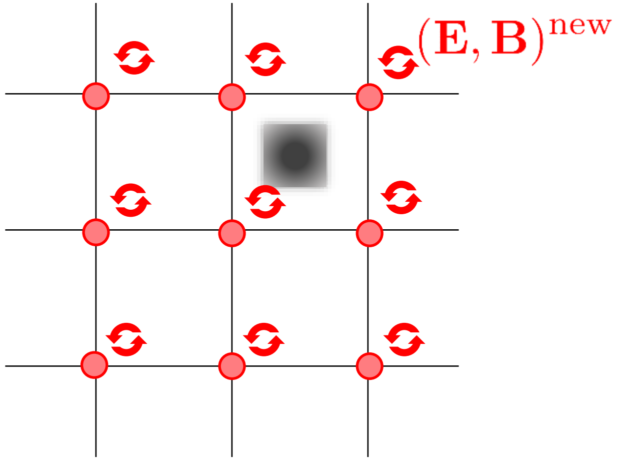
$(\mathbf{E}_g, \mathbf{B}_g) \rightarrow (\mathbf{E}_p, \mathbf{B}_p)$



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

$$\partial_t \mathbf{x}_p = \mathbf{p}_p / \gamma_p$$

$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$

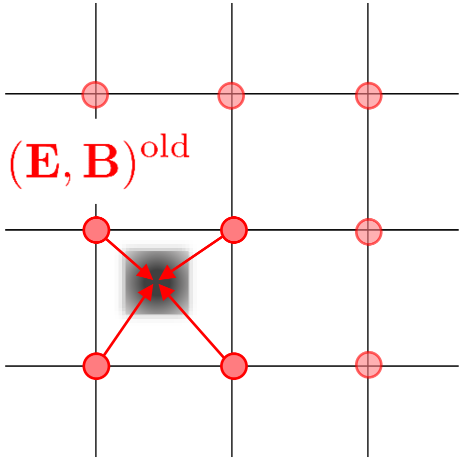


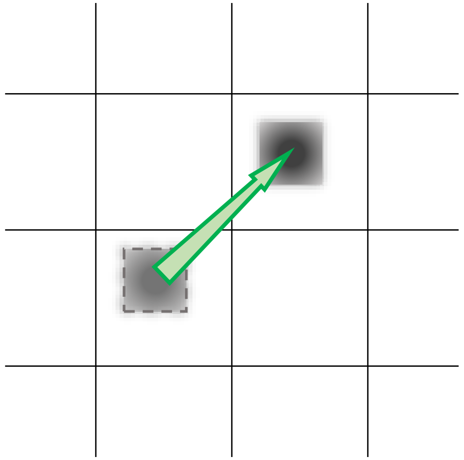
$$\nabla \times \boldsymbol{E} = -\frac{\partial B}{\partial t}$$

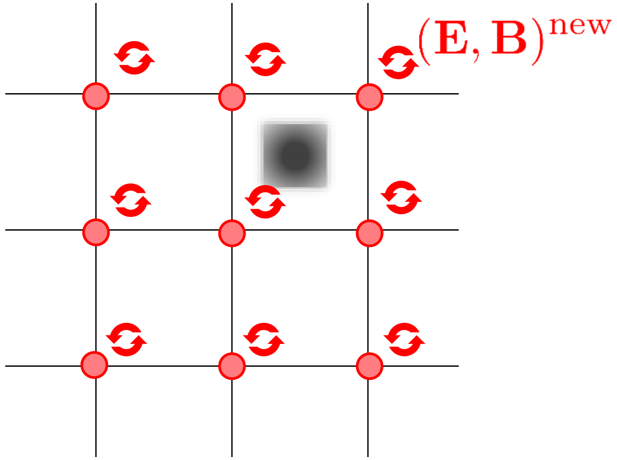
$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0 \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

PIC loop : Gathering E.M. Fields on each particle's position (Interpolator)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD







$$(\mathbf{E}_g,\mathbf{B}_g) \rightarrow (\mathbf{E}_p,\mathbf{B}_p)$$

$$\partial_t \mathbf{p}_p = q_s(\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p)$$
$$\partial_t \mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

$$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial B}{\partial t}$$
$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0\varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

Smilei) Workshop 2022

III. The PIC loop

Gathering E.M. Fields on each particle's position (Interpolator)

PIC loop : Gathering E.M. Fields on each particle's position (Interpolator)

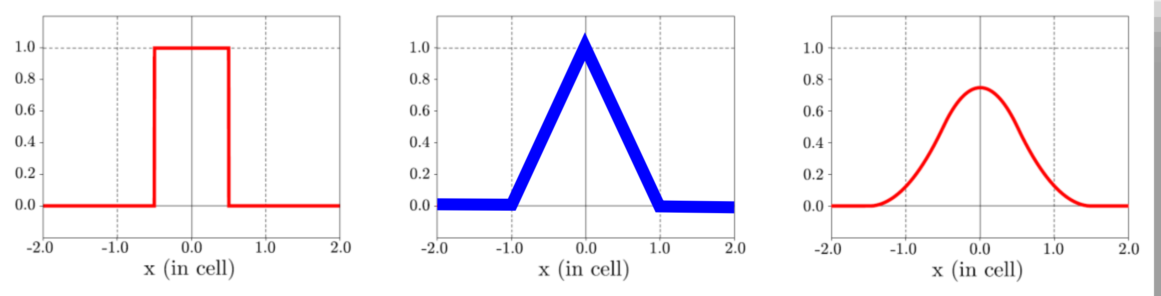
Shape function

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

weight

Dirac-distribution

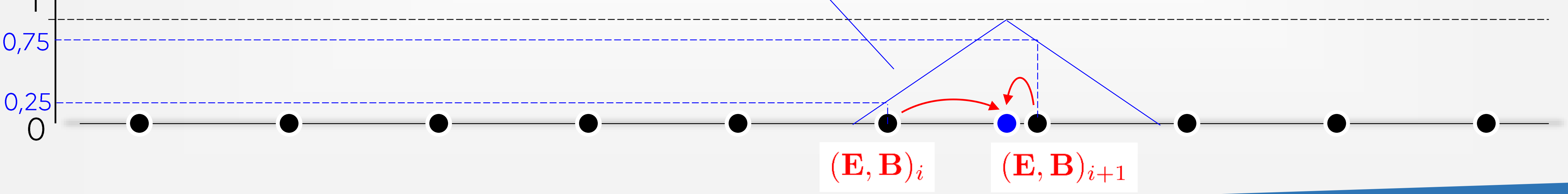
Shape functions



(a) Ordre 0 : « Top Hat » (b) Ordre 1 : « Triangle » (c) Ordre N : « Spline »

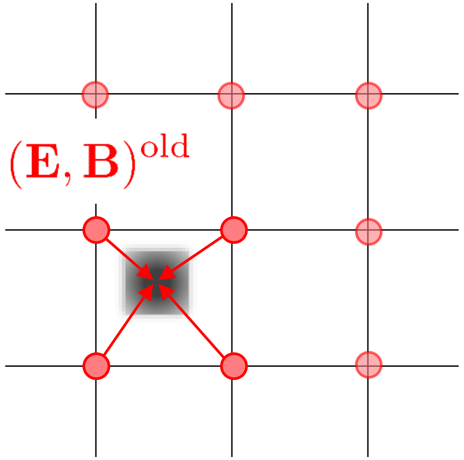
Interpolation on macro-particle's position $(\mathbf{E},\mathbf{B})_p \equiv \int (\mathbf{E},\mathbf{B})(\mathbf{x})S(\mathbf{x}-\mathbf{x}_p)\mathrm{d}\mathbf{x}$

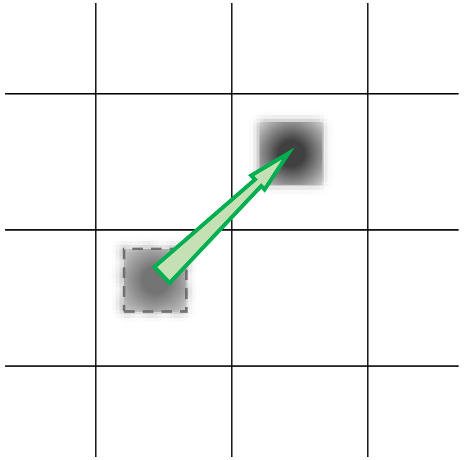
$$S(x) = \left(1 - \left|\frac{x}{\Delta_x}\right|\right)$$

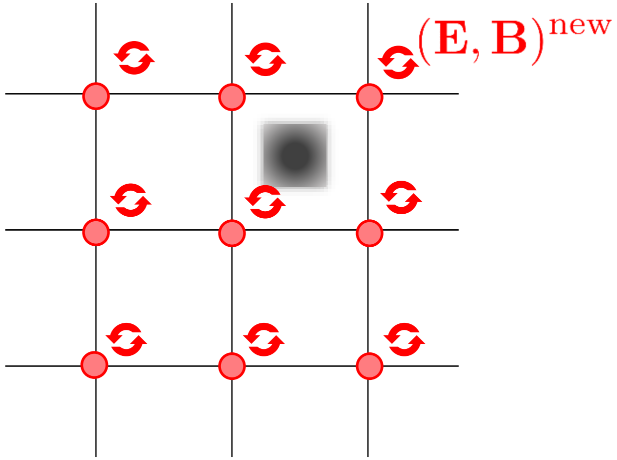


PIC loop : Deposition of current's species (Projector)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD







$$(\mathbf{E}_g,\mathbf{B}_g) \rightarrow (\mathbf{E}_p,\mathbf{B}_p)$$

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

$$\partial_t \mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

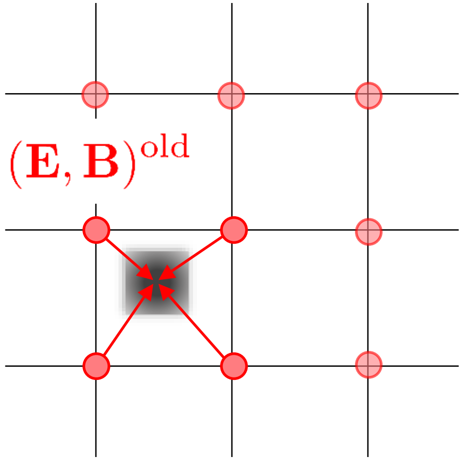
$$(\mathbf{J}_p) \rightarrow (\mathbf{J}_g)$$

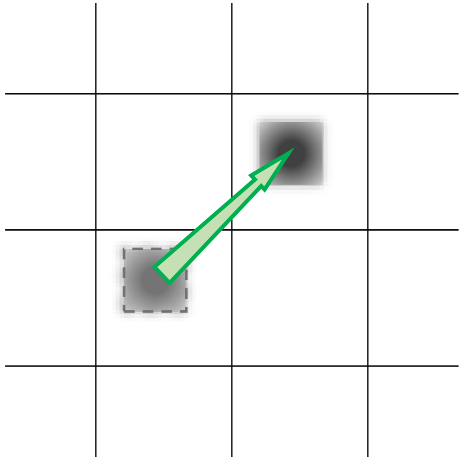
$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

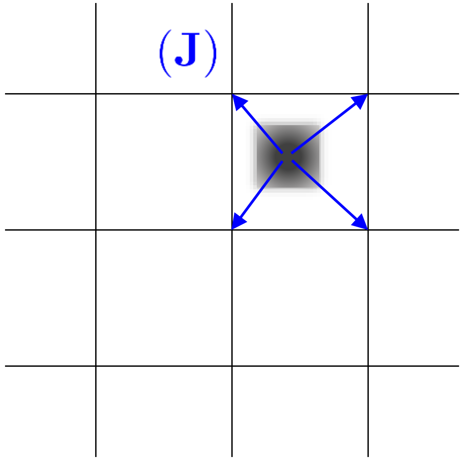
$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{\mathcal{J}} + \mu_0\varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t}$$

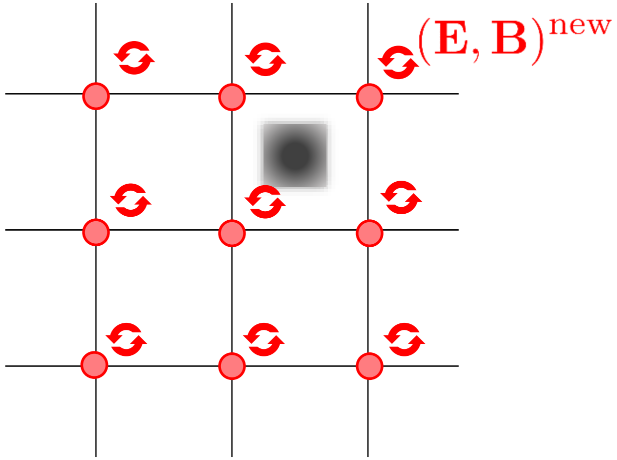
PIC loop : Deposition of current's species (Projector)

Gathering E.M. Fields on each particle's position	Numerical integration of macro-particles' dynamics equation using Lorentz's force	Deposition of current density (and charge eventually)	Numerical integration of Maxwell's equation with FDTD









$$(\mathbf{E}_g,\mathbf{B}_g)\rightarrow(\mathbf{E}_p,\mathbf{B}_p)$$

$$\partial_t\mathbf{p}_p = q_s(\mathbf{E}_p+\mathbf{v}_p\times\mathbf{B}_p)$$
$$\partial_t\mathbf{x}_p = \mathbf{p}_p/\gamma_p$$

$$(\mathbf{J}_p)\rightarrow(\mathbf{J}_g)$$

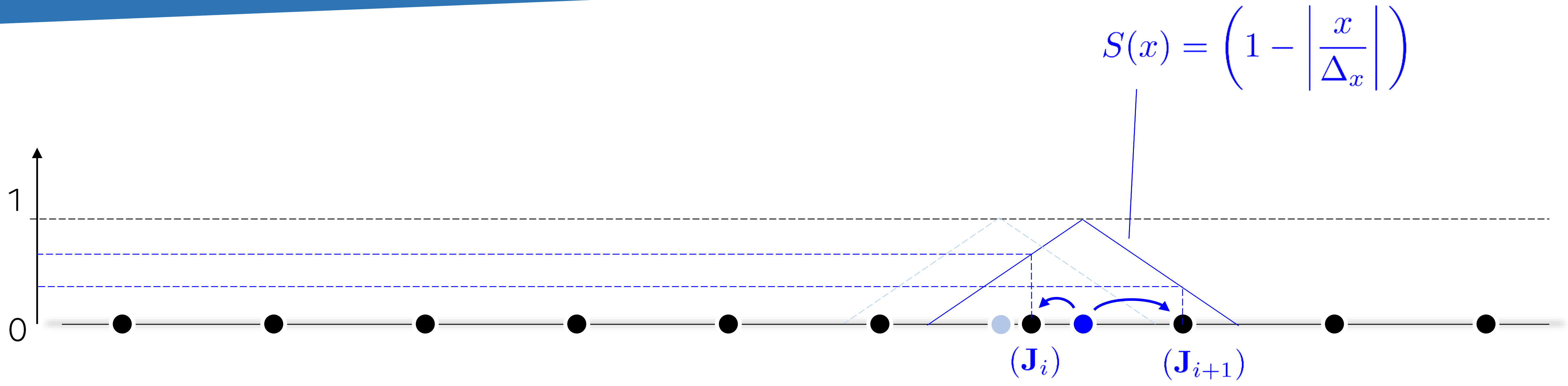
$$\nabla\times\boldsymbol{E} = -\frac{\partial B}{\partial t}$$
$$\nabla\times\boldsymbol{B} = \mu_0\boldsymbol{\mathcal{J}}+\mu_0\varepsilon_0\frac{\partial\boldsymbol{E}}{\partial t}$$

Smilei) Workshop 2022

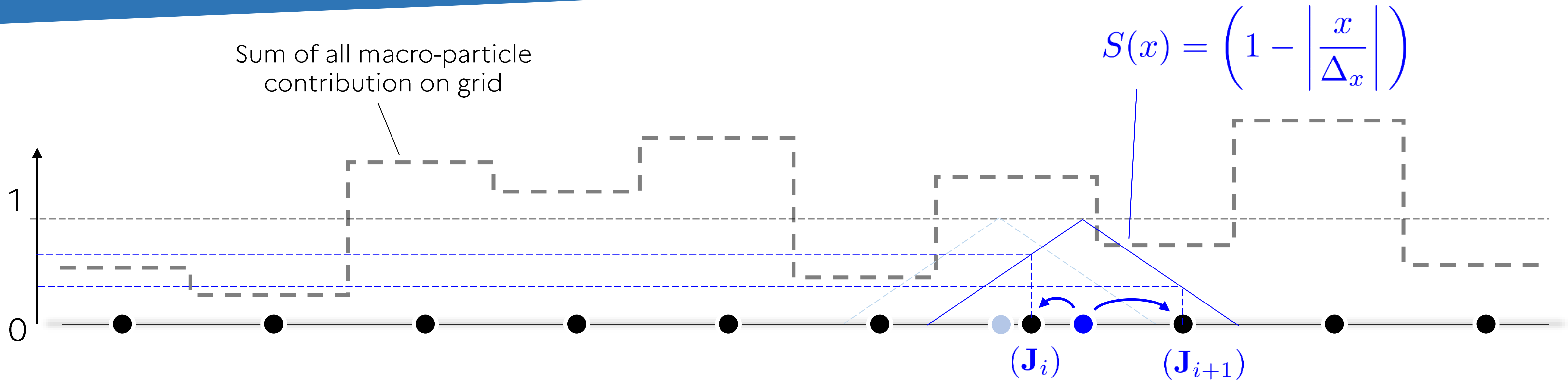
III. The PIC loop

Deposition of current's species (Projector)

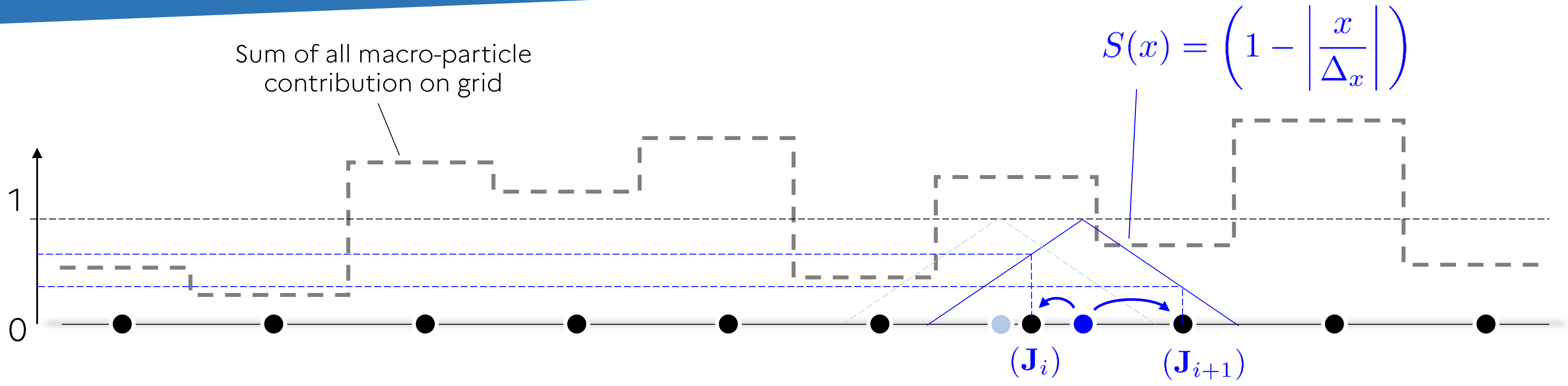
PIC loop : Deposition of current's species (Projector)



PIC loop : Deposition of current's species (Projector)

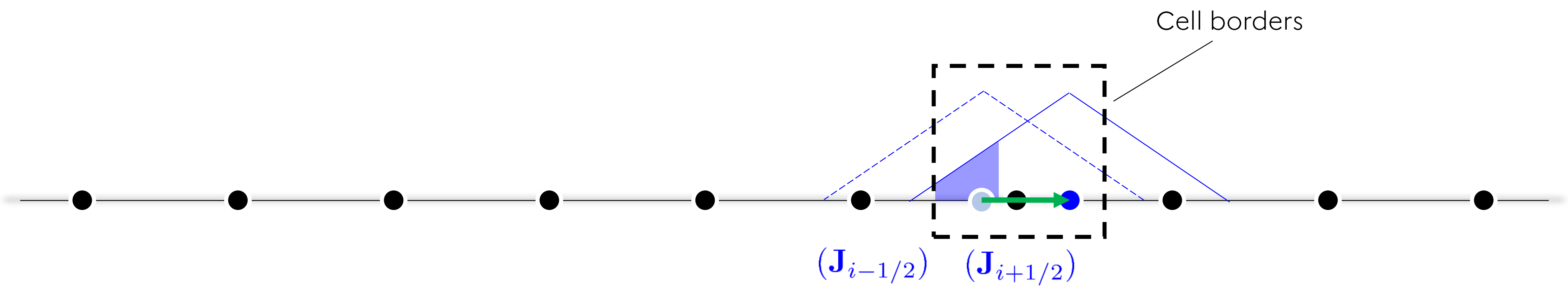


PIC loop : Deposition of current's species (Projector)



Unfortunately, this is not charge conserving

PIC loop : Deposition of current's species (Projector)



Not a simple deposition, because we have to ensure **charge conservation** solving $\nabla \cdot \mathbf{J} = \partial_t \rho$ [Villasenor & Buneman, 1992] ; [Esirkepov, 2001]

$$\frac{(J_x)^{(n+1/2)}_{i+1/2,j} - (J_x)^{(n+1/2)}_{i-1/2,j}}{\Delta_x} = \frac{q_s w_p}{\Delta_t} (W_x)^{(n+1/2)}_{i+1/2,j}$$

Workshop 2022

IV. Some numerical considerations CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\frac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$
2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;
3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;
4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.
5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\frac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$

2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;

3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;

4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.

5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\frac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$

2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;

3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;

4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.

5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\frac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$

2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;

3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;

4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.

5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\dfrac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$

2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;

3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;

4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.

5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Some numerical considerations: CFL, numerical heating, numerical Cherenkov, boundaries conditions

Some numerical limitations and constraints :

1. **Numerical heating** due to space discretization and poor Debye length $\lambda_D = \sqrt{\frac{\varepsilon_0 \mathrm{k_B} T_e}{e^2 n_e^2}}$ resolution
   Constraint: $\Delta x < 3\lambda_D$
2. **Courant-Friedrich-Levy limit**: The information can't propagate through more than on cell during one timestep. For one dimensional simulation $c\Delta t < \Delta x$;
3. **Numerical dispersion**: Generally, phase velocity of wave in vacuum is neither the same for all frequency nor isotropic. Often $v_\phi \leq c$;
4. **Numerical Cherenkov Radiation**: If particles velocities are above numerical phase velocity of wave it produce radiation.
5. **Boundary conditions**: Due to finite grid and finite stencil, open boundary conditions are not perfect (for all frequency and angle) and spurious reflection can deteriorate physics of interest.

Smilei) Workshop 2022

References

References

Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics
S.D. Gedney

Plasma Physics via Computer Simulation
C.K. Birdsall & A. B. Langdon

Computational Electrodynamics
A. Taflove

Numerical Recipies
W.H. Press *et al.*

Acknowledgement

Thank you for your attention

Funding agencies

GdR APPEL
Groupement de Recherche Accélérateurs Plasma PompÉs par Lasers

PLAS@PAR Plasmas à Paris

P2IO Physique des 2 Infinis et des Origines

PALM Laboratoire d'Excellence Physique : Atomes Lumière Matière

Physique des Infinis Une Initiative Sorbonne Université

AIDAS AI Data Analytics and Scalable Simulations

Contributing labs & institutions

MAISON DE LA SIMULATION

irap astrophysique & planétologie

cnrs

l'X ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS

iris

intel

CELIA

cea

SORBONNE UNIVERSITÉ

université PARIS-SACLAY

Smilei) Workshop 2022

EXTRA

Smilei) Workshop 2022

a. Step forward for Maxwell equation FDTD implementation

Simulation box: Space grids and time discretization

Naturally, Maxwell equation become (for Bz as example) :

$$\frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n-1/2}_{i+1/2,j+1/2}}{\Delta_t} = \frac{(E_y)^{n}_{i+1,j+1/2} - (E_y)^{n}_{i,j+1/2}}{\Delta_x} + \frac{(E_x)^{n}_{i+1/2,j} - (E_x)^{n}_{i+1/2,j+1}}{\Delta_y}$$

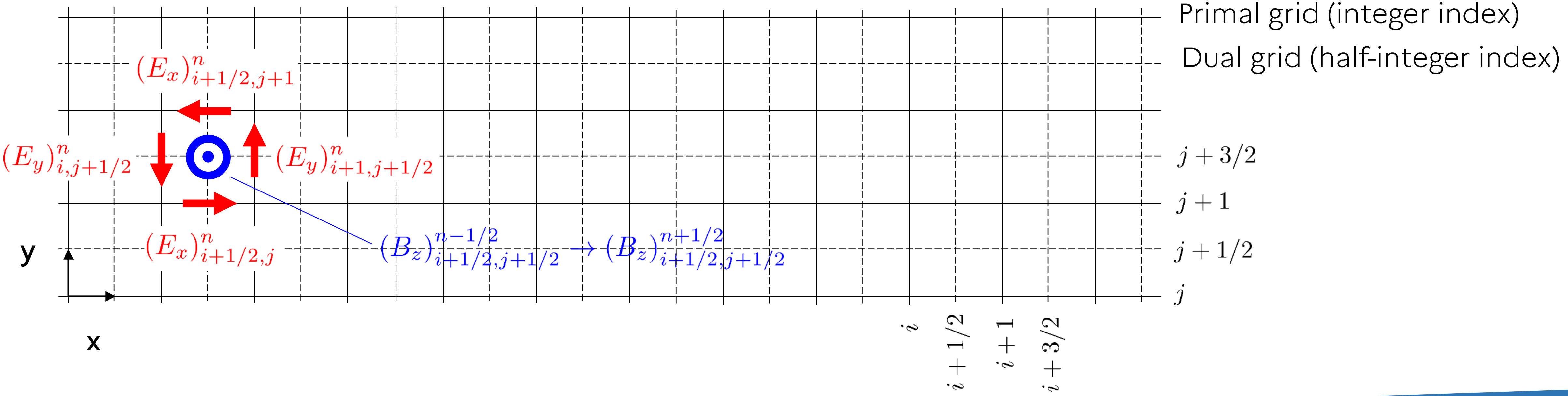
$$\frac{(E_y)^{n+1}_{i,j+1/2} - (E_y)^{n}_{i,j+1/2}}{\Delta_t} = -(J_y)^{n+1/2}_{i,j+1/2} - \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i-1/2,j+1/2}}{\Delta_x}$$

$$\frac{(E_x)^{n+1}_{i+1/2,j} - (E_x)^{n}_{i+1/2,j}}{\Delta_t} = -(J_x)^{n+1/2}_{i+1/2,j} + \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i+1/2,j-1/2}}{\Delta_y}$$

Simulation box: Space grids and time discretization

$$\frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n-1/2}_{i+1/2,j+1/2}}{\Delta_t} = \frac{(E_y)^n_{i+1,j+1/2} - (E_y)^n_{i,j+1/2}}{\Delta_x} + \frac{(E_x)^n_{i+1/2,j} - (E_x)^n_{i+1/2,j+1}}{\Delta_y}$$

Graphically you can see :

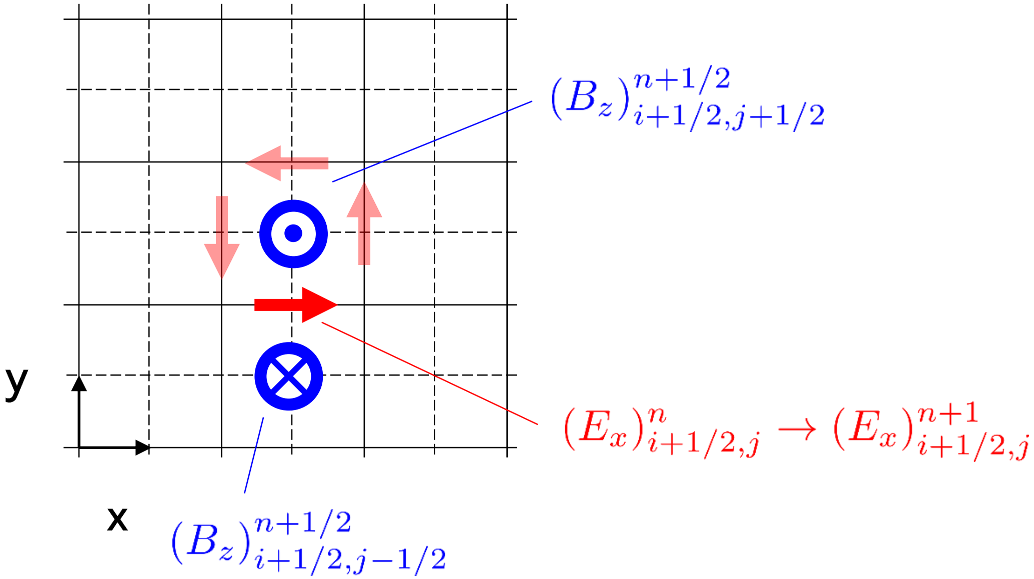


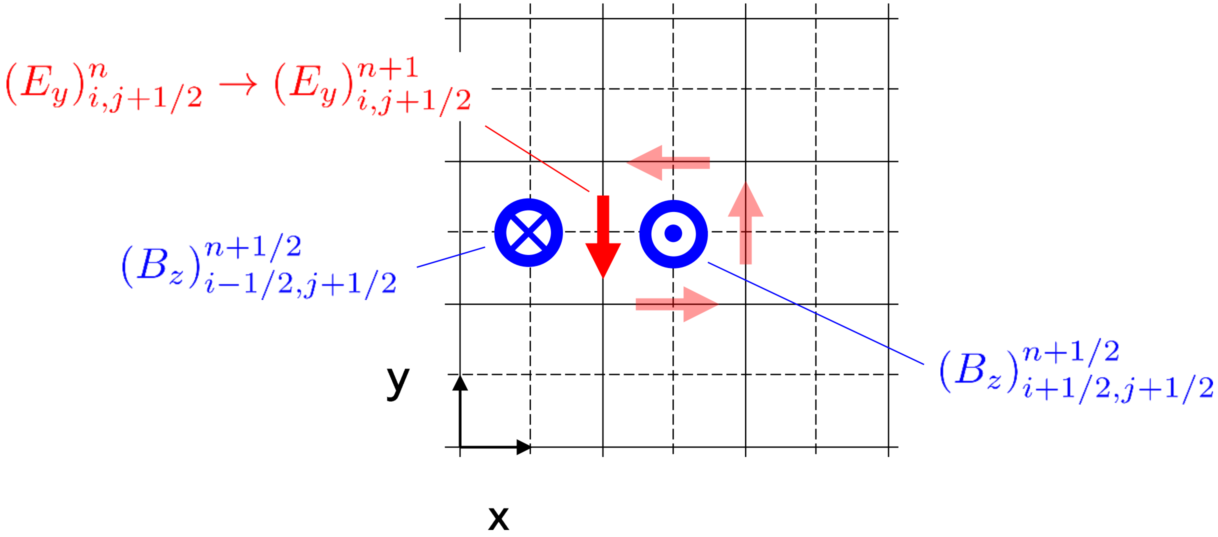
Simulation box: Space grids and time discretization

$$\frac{(E_y)^{n+1}_{i,j+1/2} - (E_y)^{n}_{i,j+1/2}}{\Delta_t} = -(J_y)^{n+1/2}_{i,j+1/2} - \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i-1/2,j+1/2}}{\Delta_x}$$

$$\frac{(E_x)^{n+1}_{i+1/2,j} - (E_x)^{n}_{i+1/2,j}}{\Delta_t} = -(J_x)^{n+1/2}_{i+1/2,j} + \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i+1/2,j-1/2}}{\Delta_y}$$

Graphically you can see :



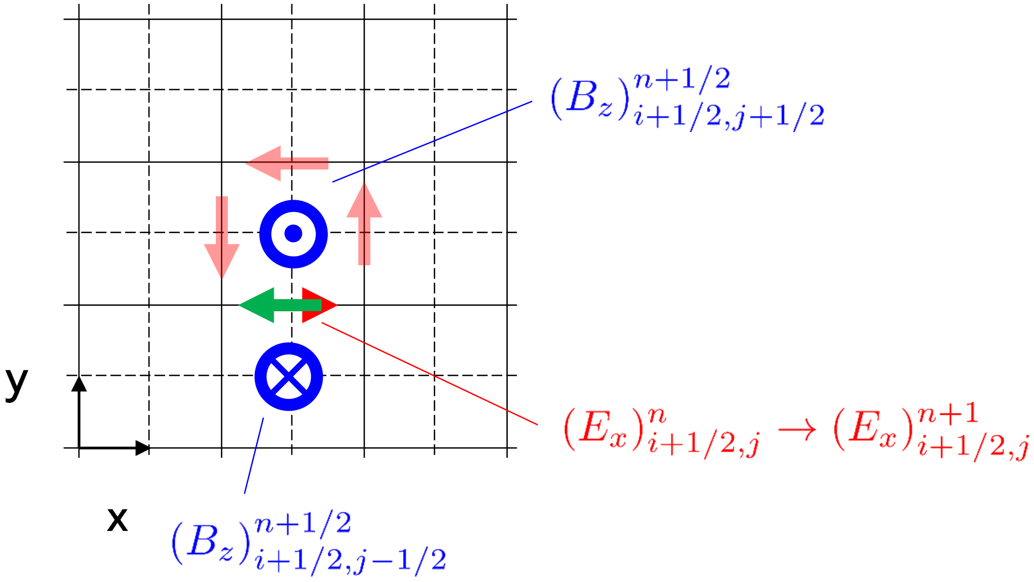


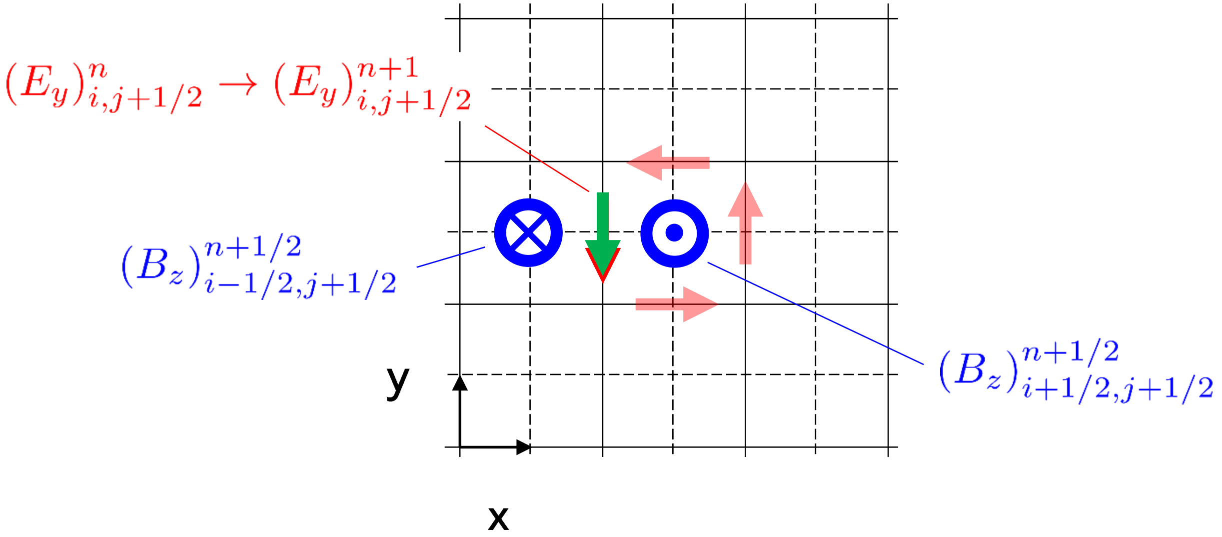
Simulation box: Space grids and time discretization

$$\frac{(E_y)^{n+1}_{i,j+1/2} - (E_y)^{n}_{i,j+1/2}}{\Delta_t} = -(J_y)^{n+1/2}_{i,j+1/2} - \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i-1/2,j+1/2}}{\Delta_x}$$

$$\frac{(E_x)^{n+1}_{i+1/2,j} - (E_x)^{n}_{i+1/2,j}}{\Delta_t} = -(J_x)^{n+1/2}_{i+1/2,j} + \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i+1/2,j-1/2}}{\Delta_y}$$

Graphically you can see :





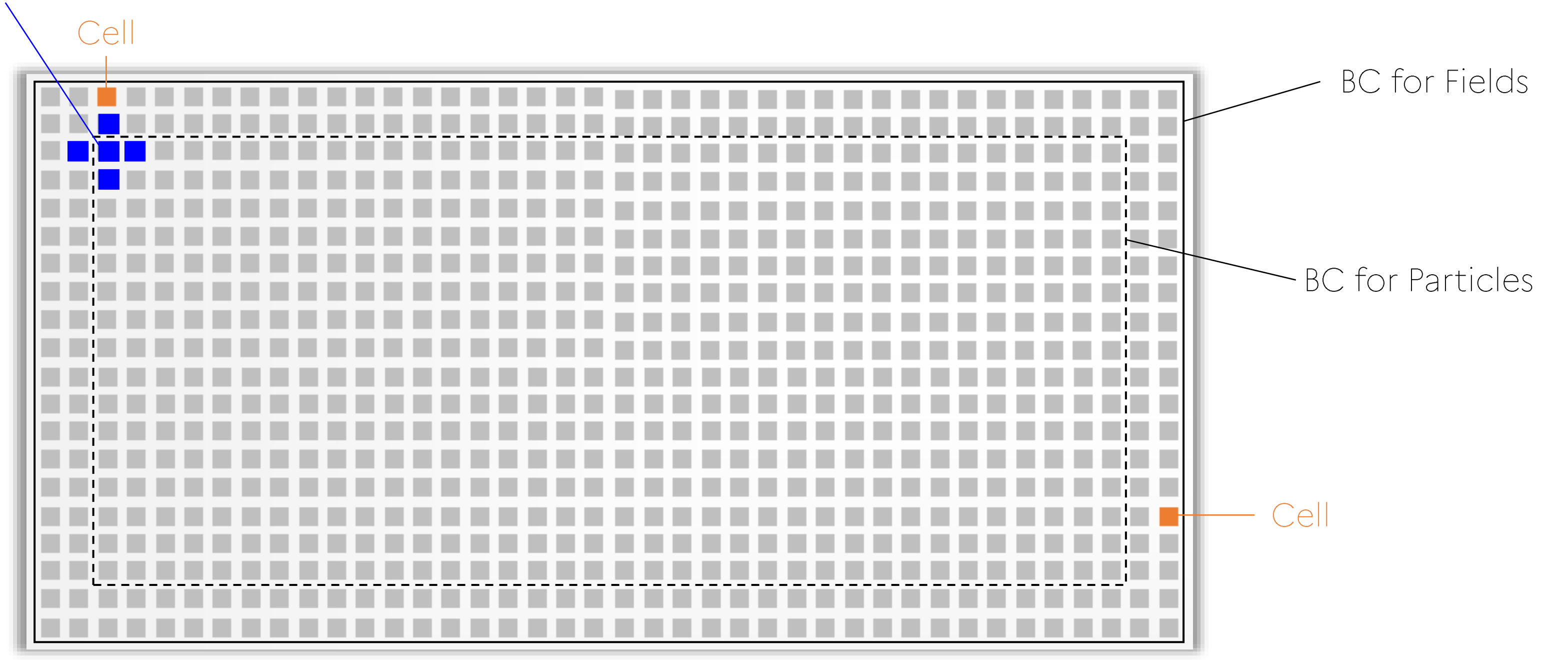
Smilei) Workshop 2022

b. Pro for FDTD method which is easy to parallelize

Simulation box: Space grids and time discretization

The locality in space and time of the FDTD method help to speed-up the calculation :

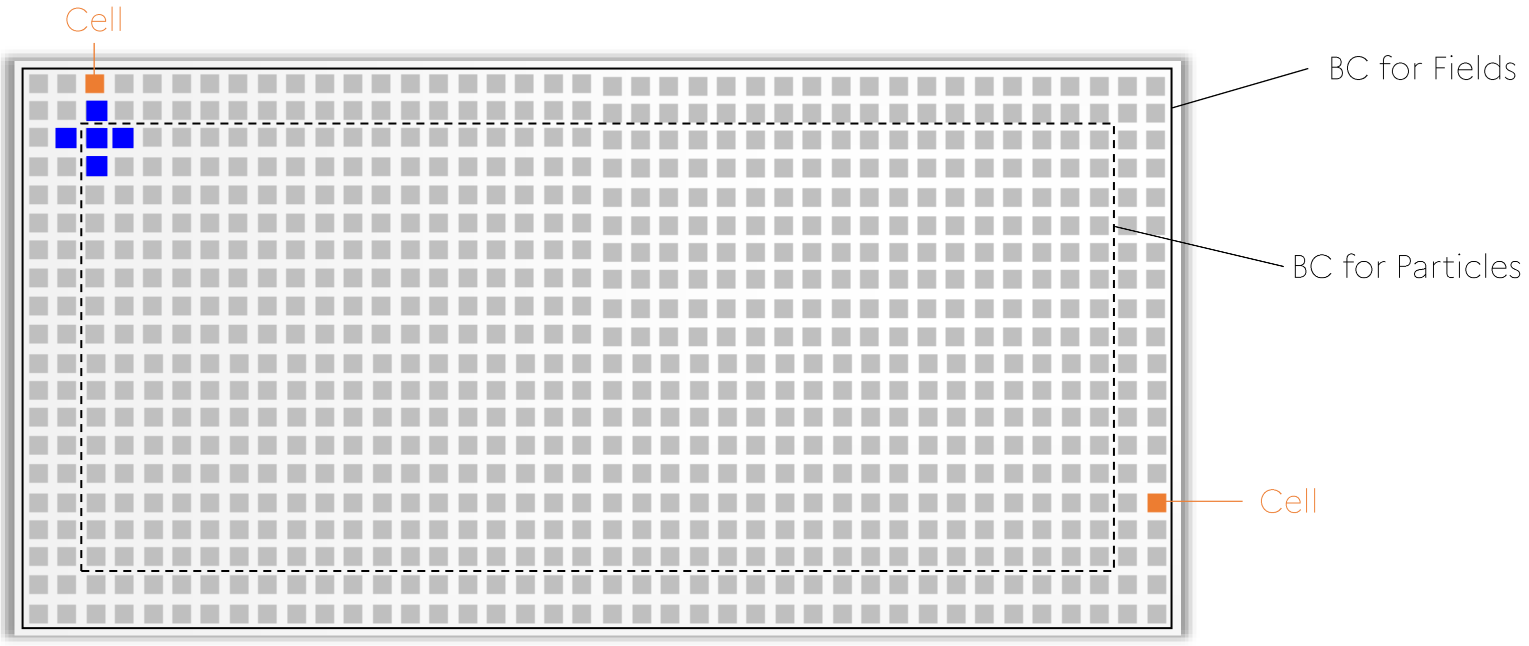
$$(B_z)^{n-1/2}_{i+1/2,j+1/2} \to (B_z)^{n+1/2}_{i+1/2,j+1/2}$$



Simulation domain

Simulation box: Space grids and time discretization

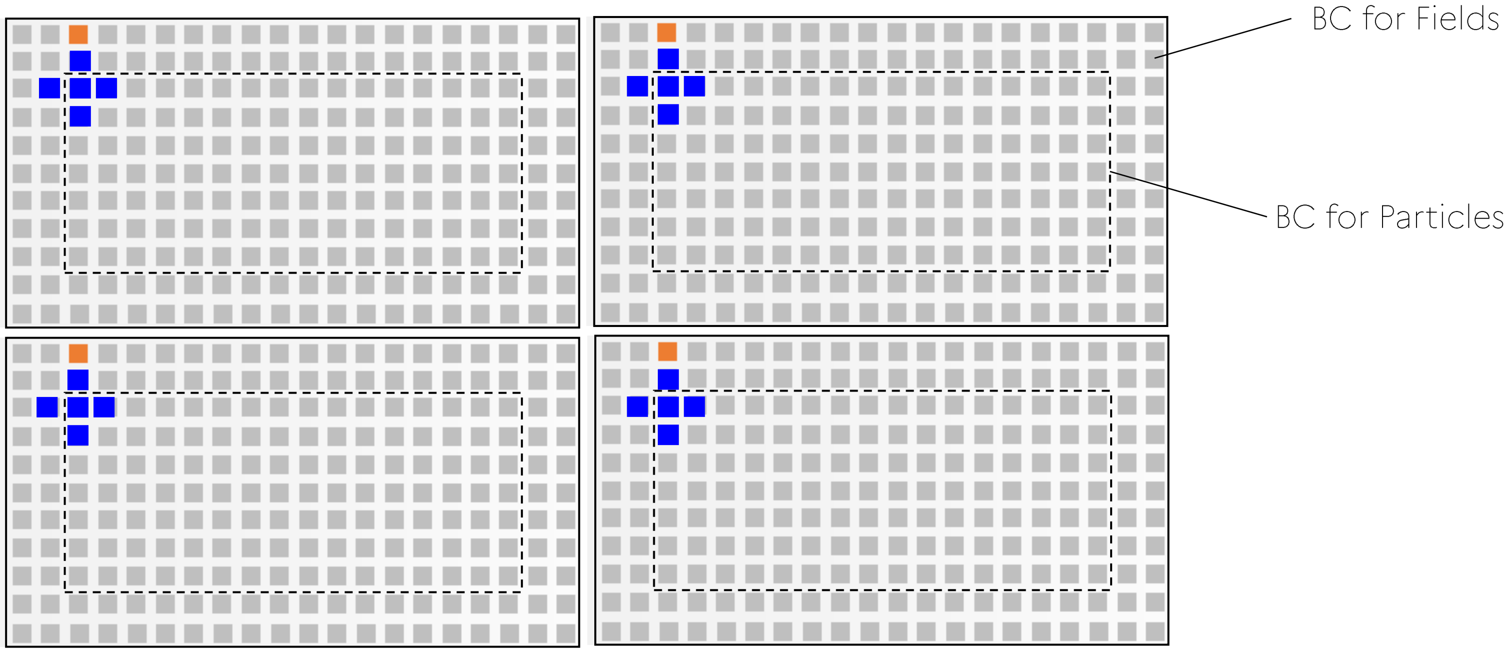
The locality in space and time of the FDTD method help to speed-up the calculation :



Simulation domain

Simulation box: Space grids and time discretization

The locality in space and time of the FDTD method help to speed-up the calculation :



Simulation domain

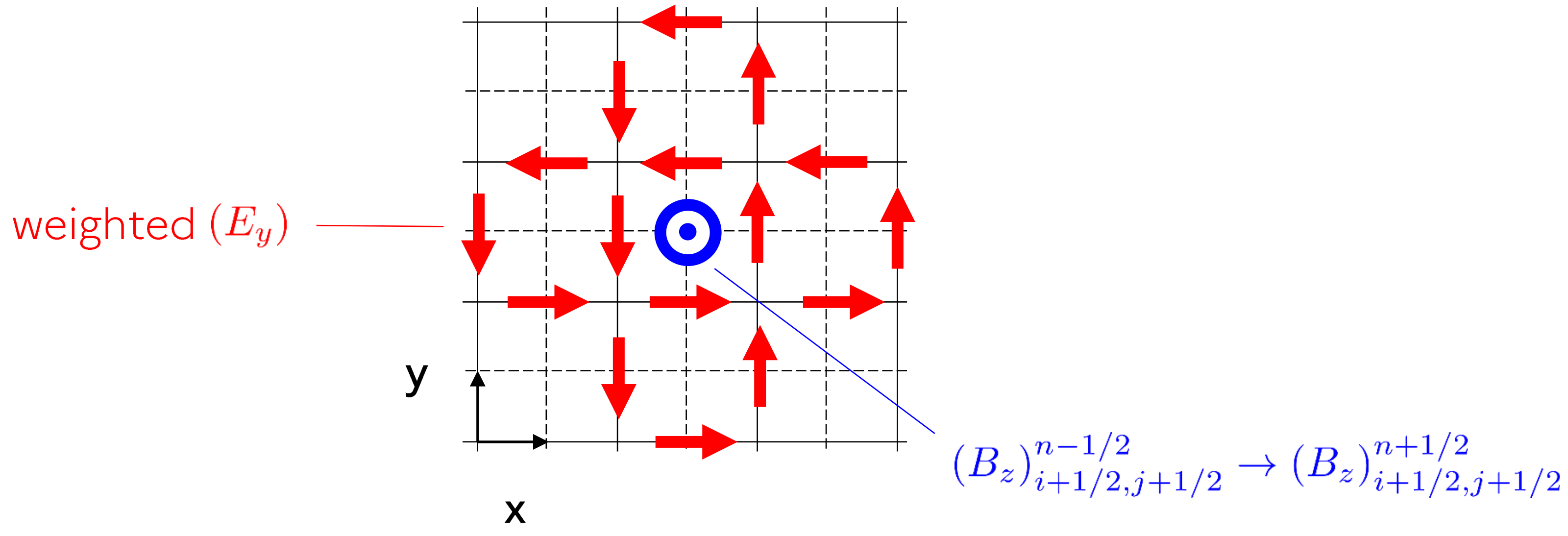
Smilei) Workshop 2022

c. Fancy way to discretize curl operator

Simulation box: Space grids and time discretization

We can extend stencil in order to reduce some numerical drawbacks. E.g. for **Maxwell-Faraday** in 2D:

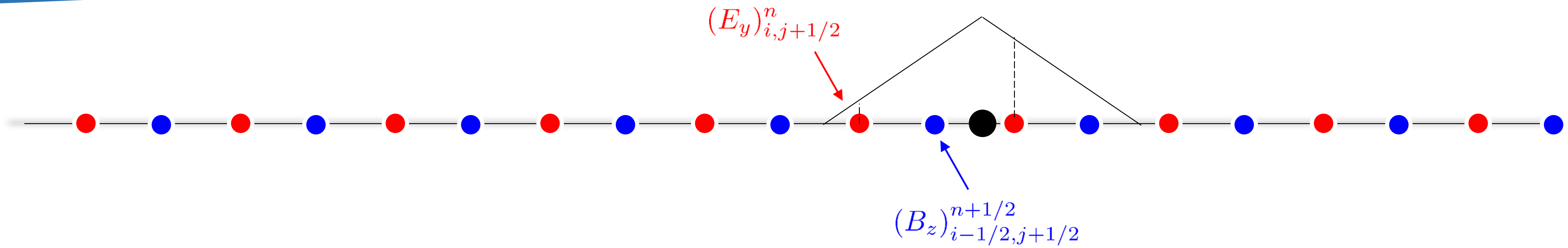
$$\nabla \times \mathbf{E} = -\partial_t \mathbf{B}$$



Smilei) Workshop 2022

d. Note on the interpolator

PIC loop
Interpolator



$$\frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n-1/2}_{i+1/2,j+1/2}}{\Delta_t} = \frac{(E_y)^n_{i+1,j+1/2} - (E_y)^n_{i,j+1/2}}{\Delta_x} + \frac{(E_x)^n_{i+1/2,j} - (E_x)^n_{i+1/2,j+1}}{\Delta_y}$$

$$\frac{(E_y)^{n+1}_{i,j+1/2} - (E_y)^n_{i,j+1/2}}{\Delta_t} = -(J_y)^{n+1/2}_{i,j+1/2} - \frac{(B_z)^{n+1/2}_{i+1/2,j+1/2} - (B_z)^{n+1/2}_{i-1/2,j+1/2}}{\Delta_x}$$

Time interpolation : $\mathbf{B}^{(n)} = \dfrac{\mathbf{B}^{(n-1/2)} + \mathbf{B}^{(n+1/2)}}{2}$

Smilei) Workshop 2022

e. Discussion about CFL conditions

Courant-Friedrich Limit for explicit FDTD

Von Neumann study : $F = F_0 \exp\Big(\jmath\,(\omega t - \boldsymbol{k}\cdot\boldsymbol{r})\Big) \rightarrow (F)^n_{i,j} = F_0 \exp\Big(\jmath\,(\omega\; n\Delta t - k_x\; i\Delta x - k_y\; j\Delta y)\Big)$

$$\frac{\sin^2(\omega\Delta t/2)}{\Delta t^2} = \frac{\sin^2(k_x\Delta x/2)}{\Delta x^2}\Big[\alpha_x + 2\beta_{xy}\cos(k_y\Delta y) + \delta_x\Big(1 + 2\cos(k_x\Delta x)\Big)\Big]$$
$$+ \frac{\sin^2(k_y\Delta y/2)}{\Delta y^2}\Big[\alpha_y + 2\beta_{yx}\cos(k_x\Delta x) + \delta_y\Big(1 + 2\cos(k_y\Delta y)\Big)\Big]$$

Continum limit : $\omega^2 = k_x^2(\alpha_x + 2\beta_{xy} + 3\delta_x) + k_y^2(\alpha_y + 2\beta_{yx} + 3\delta_y)$

$$\alpha_x + 2\beta_{xy} + 3\delta_x = 1$$
$$\alpha_y + 2\beta_{yx} + 3\delta_y = 1$$

CFL :

$$\sin^2\left(\omega\frac{\Delta t}{2}\right) \leq 1$$

Courant-Friedrich Limit for explicit FDTD

CFL :

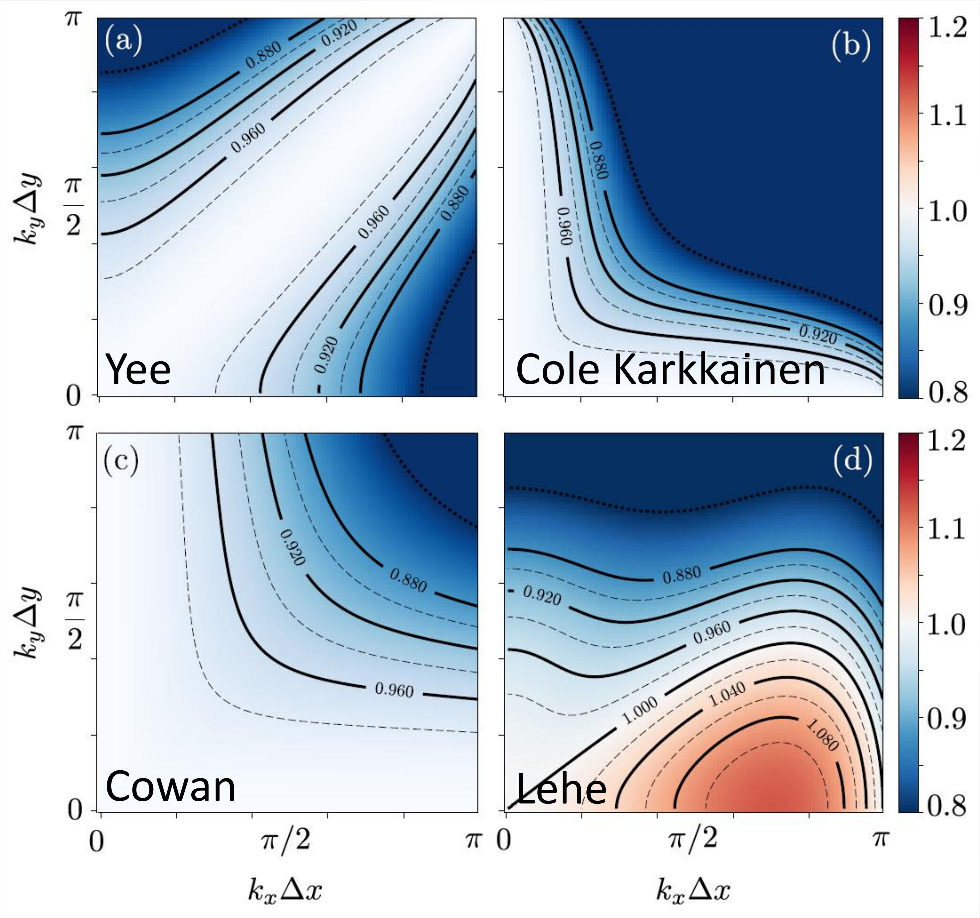
$$\sin^2\left(\omega\frac{\Delta t}{2}\right) \leq 1$$

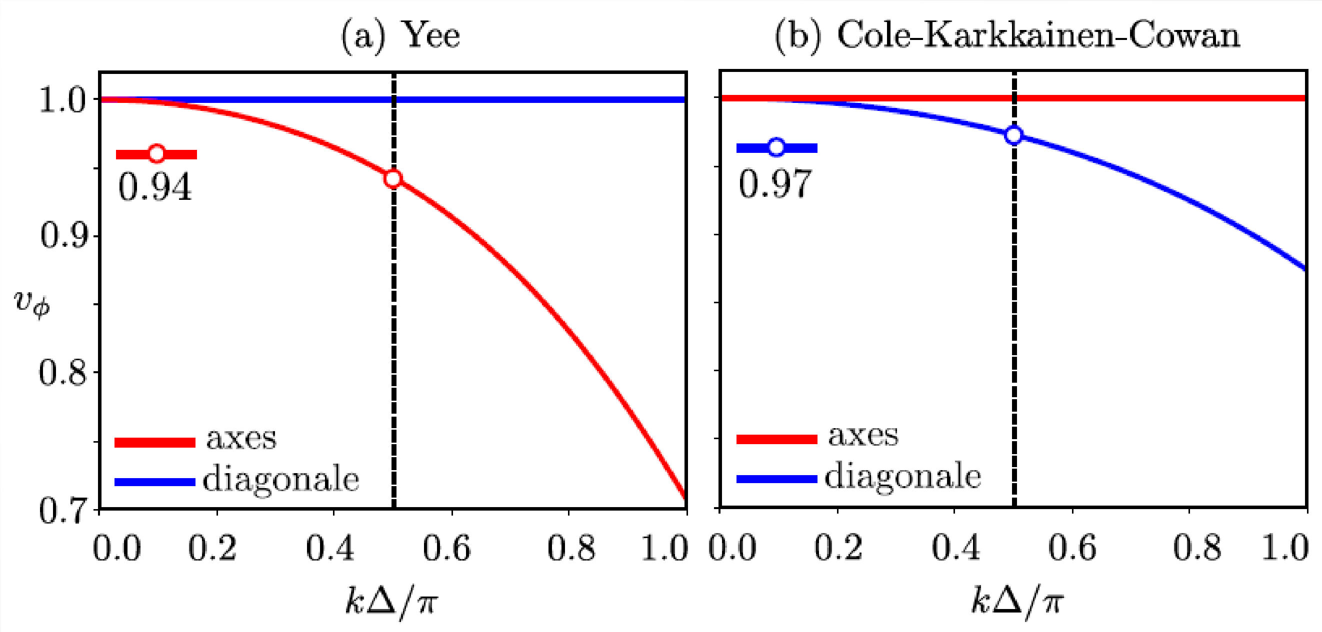
$$\Delta t \leq \mathrm{MIN}\left(\frac{\Delta x}{\sqrt{1-4\delta_x}}, \frac{\Delta y}{\sqrt{1-4\delta_y}}, \sqrt{\frac{1-4\beta_{xy}-4\delta_x}{\Delta x^2}+\frac{1-4\beta_{yx}-4\delta_y}{\Delta y^2}}^{-1}\right)$$

Smilei) Workshop 2022

f. Numerical dispersion

Dispersion of Maxwell Solver



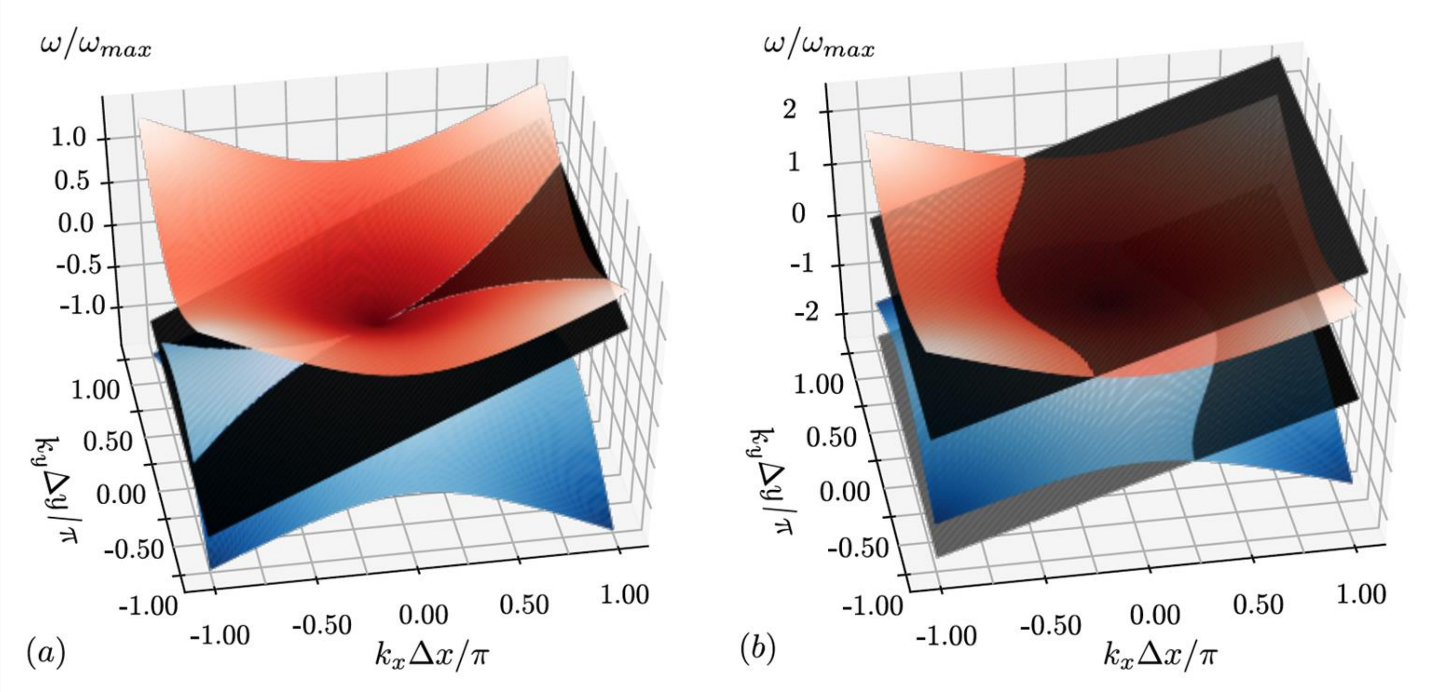


Smilei) Workshop 2022

g. Numerical cherenkov

Numerical Cherenkov

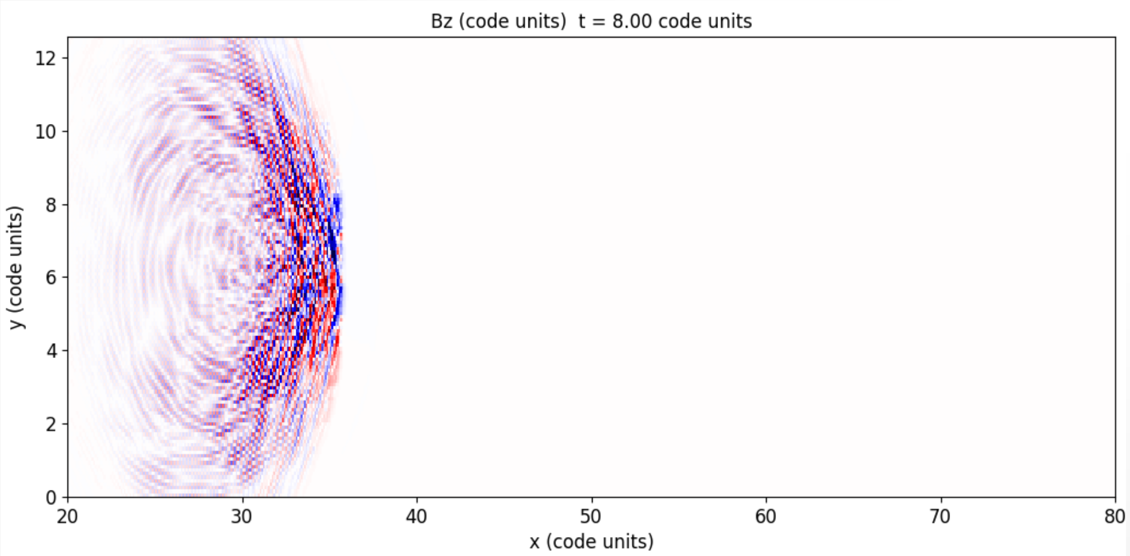
Numerical Cherenkov Radiation + Instabilities

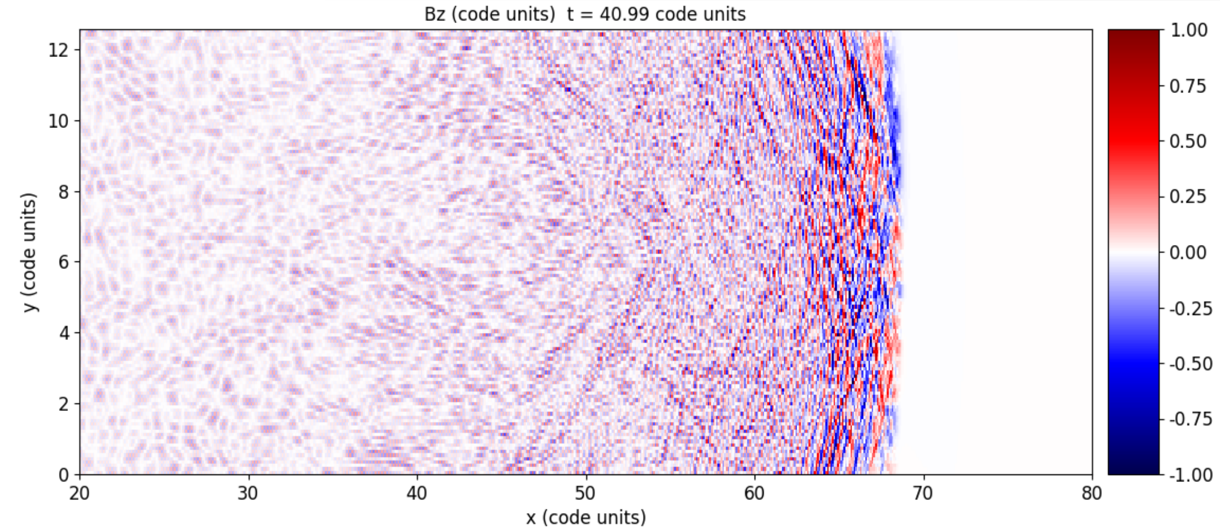


Numerical Cherenkov

Numerical Cherenkov Radiation

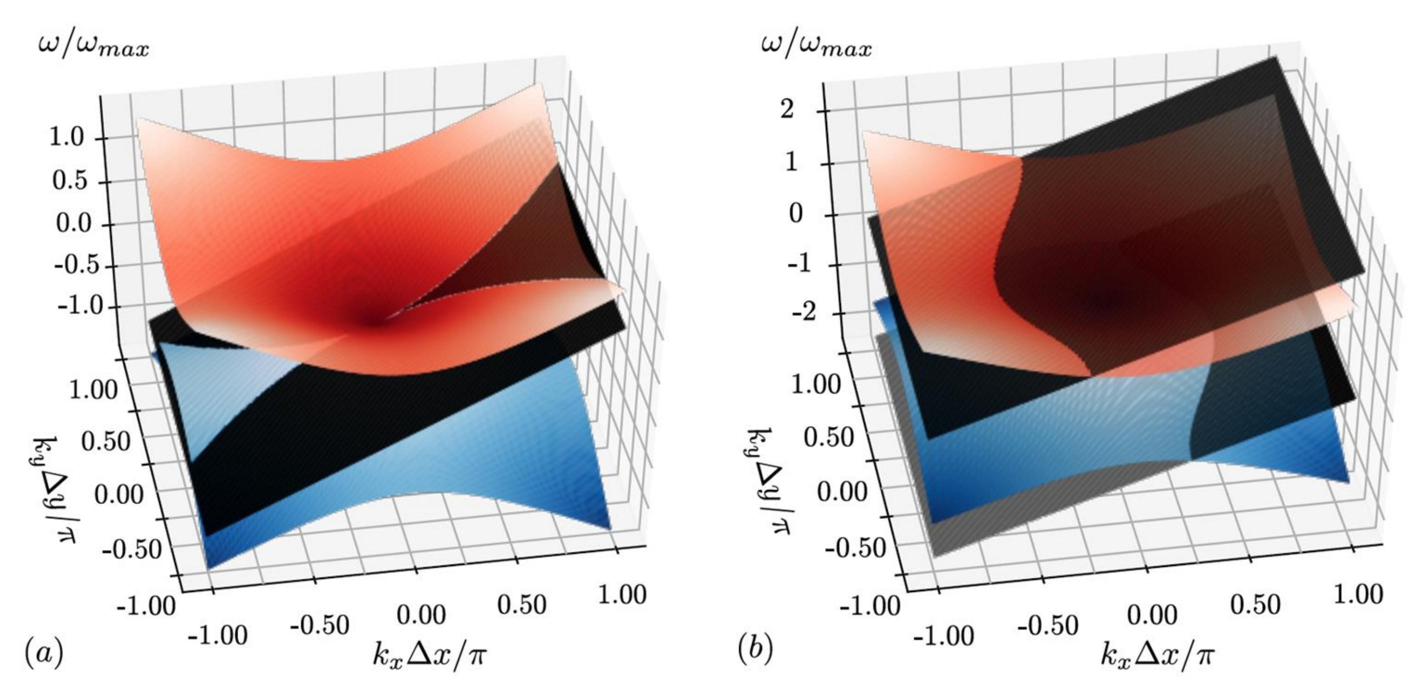
A relativistic bunch of e- and e+ with v_x*=0.98*

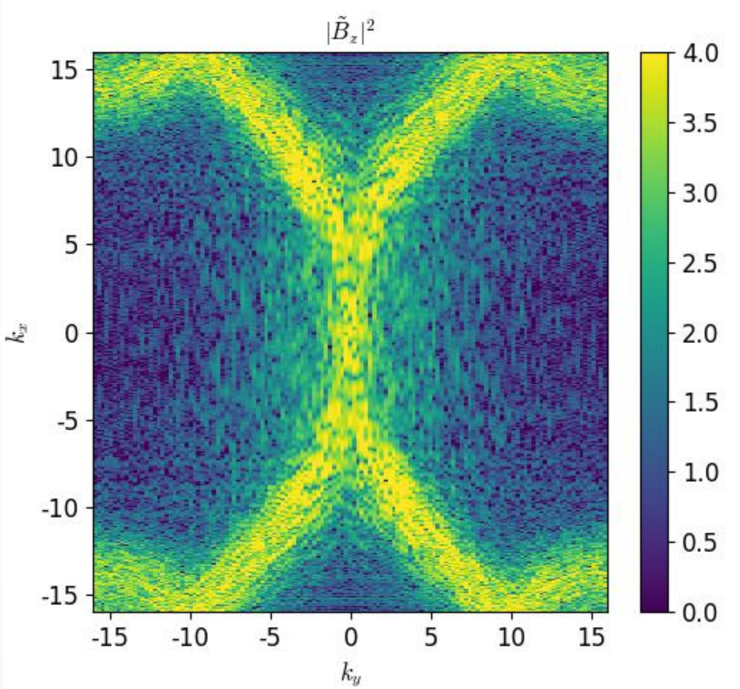




Numerical Cherenkov

Numerical Cherenkov Radiation + Instabilities (explanation with Fourier space)

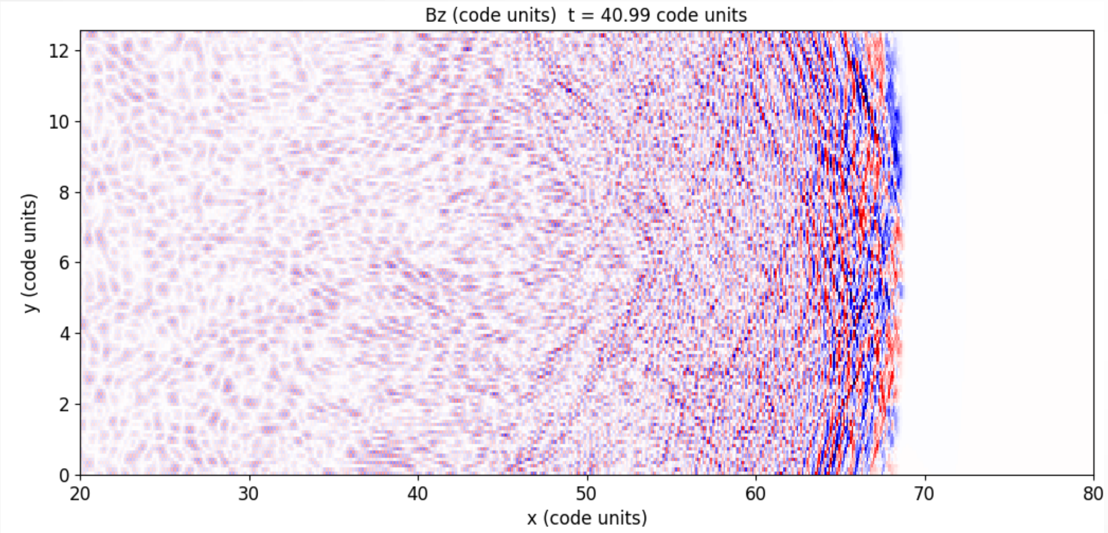




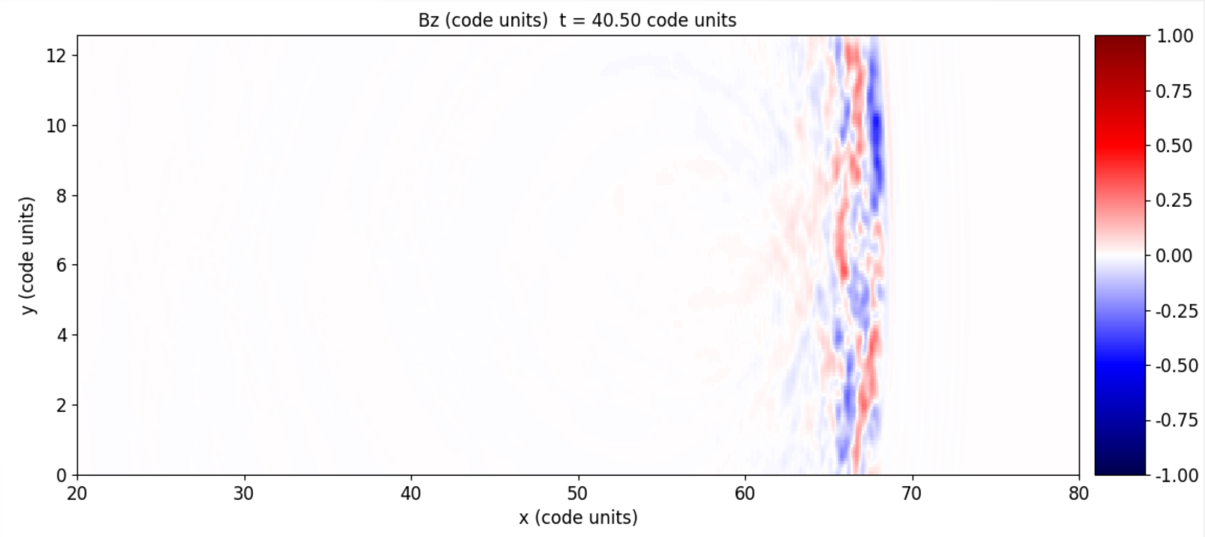
Fourier Space

Numerical Cherenkov

Numerical Cherenkov Radiation



with Numerical Radiation



without Numerical Radiation

Numerical Cherenkov

Numerical Cherenkov Radiation