Advanced modeling tools for laser-plasma accelerators (LPAs) 1/3

Carlo Benedetti
LBNL, Berkeley, CA, USA
(with contributions from R. Lehe, J.-L. Vay, T. Mehrling)

Work supported by Office of Science, Office of HEP, US DOE Contract DE-AC02-05CH11231

Course overview

- 3 lectures: Monday, Tuesday, Thursday

- Topics:

  - [1] The Particle-In-Cell (PIC) method as a tool to study laser-plasma interaction in LPAs;

  - [2] Limits/challenges of conventional PIC codes;

  - [3] Tools to speed-up the modeling of LPAs (Lorentz-boosted frame, quasi-static approximation, Fourier-mode decomposition, ponderomotive guiding center description, etc.);

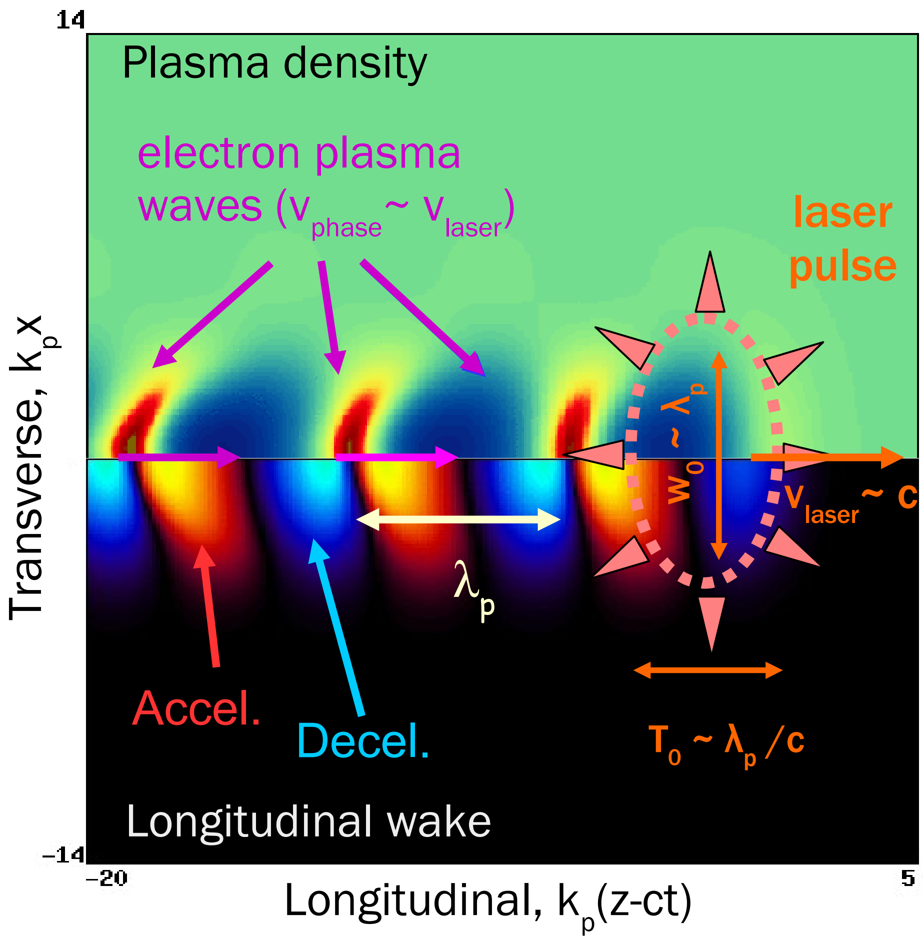
Overview of lecture 1

- Basic physics of laser-plasma accelerators (LPAs);
- The Vlasov-Maxwell (V-M) equations system;
- The PIC approach to solve V-M equations system:
  - Numerical particles;
  - The PIC loop;
  - Force interpolation and current deposition;
  - Pushing “numerical” particles;
  - Solving Maxwell's equations on a grid;

LPA as compact accelerators

Short and intense laser propagating in a plasma:
- short → $T_0 = L_0/c \sim \lambda_p/c$ (tens of fs)
- intense → $a_0 = eA_0/mc^2 \sim 1$ (λ_0=0.8 um, $I_0 > 10^{18}$ W/cm^2)

Plasma wavelength:
$\lambda_p \sim n_0^{-1/2} \approx$ 10-100 μm,
for $n_0 \approx 10^{19}$-10^{17}cm^{-3}



▲ = ponderomotive force:

$F_p \sim -\text{grad}[I_{laser}]$

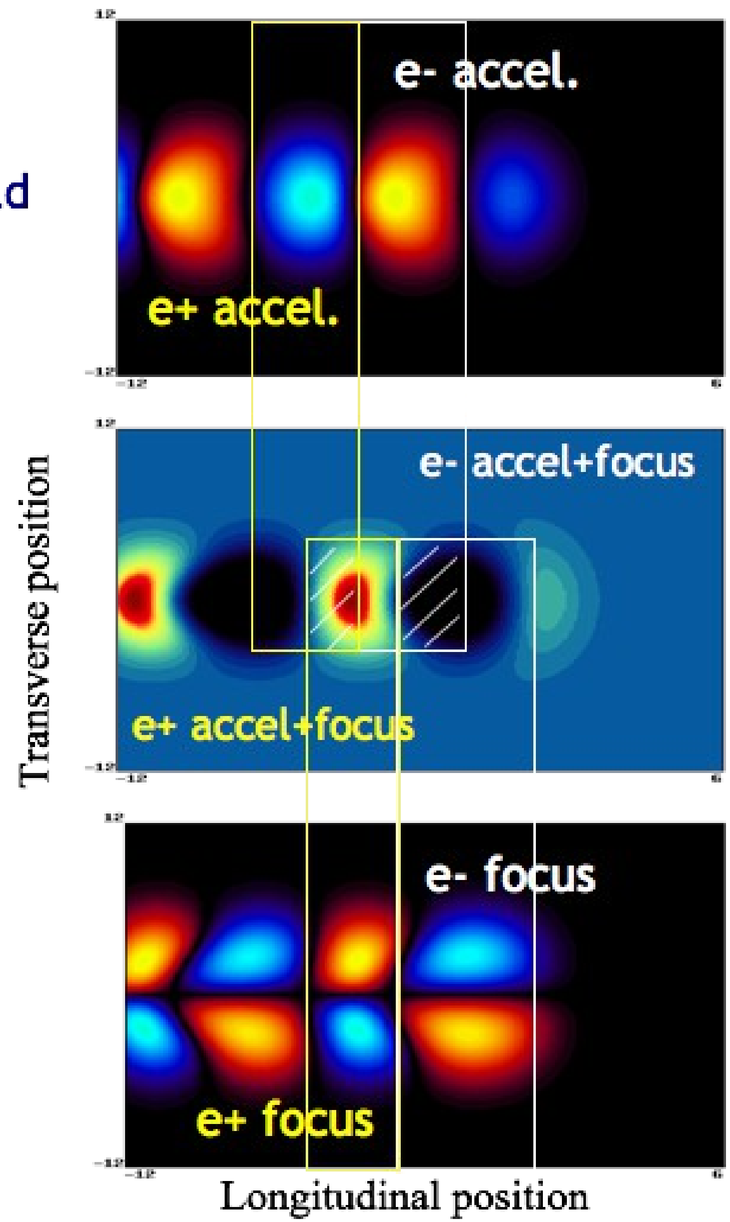
→ F_p displaces electrons (but not the ions) creating charge separation from which EM fields arise

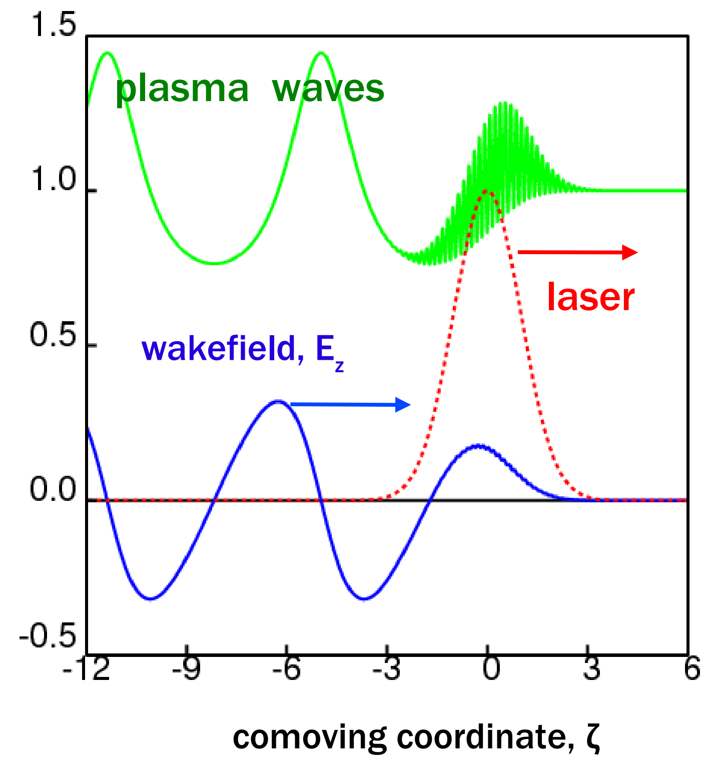
LPAs produce 1-100 GV/m accelerating gradients + confining forces

Accelerating field

Plasma density

Focusing field





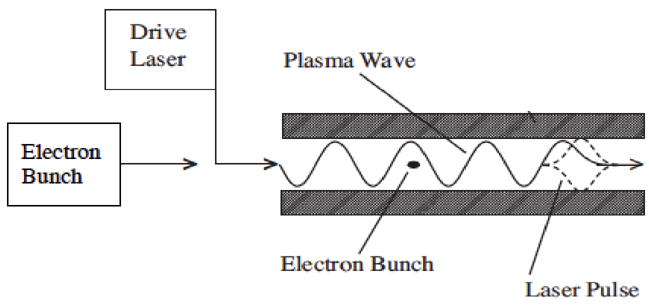
$E_z \sim mc\omega_p/e \sim 100\ [V/m] \times (n_0[cm^{-3}])^{1/2}$

e.g., $n_0 \sim 10^{17}\ cm^{-3}$, $a_0 \sim 1 \rightarrow E_z \sim 30$ GV/m, ~ 10^2-10^3 larger than conventional RF accelerators

Electron bunches to be accelerated in an LPA can be obtained from background plasma

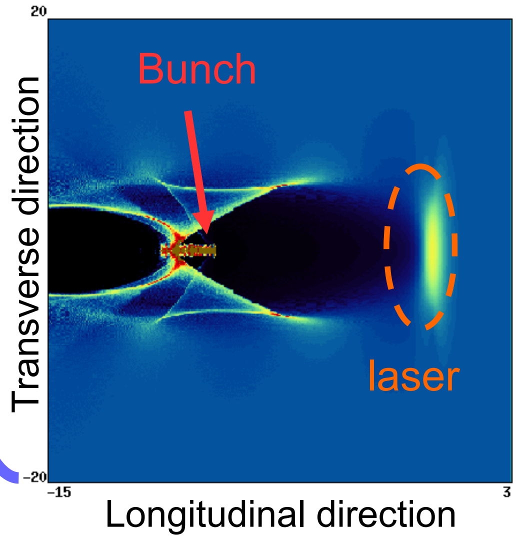
Electron bunch to be accelerated

→ external injection (bunch from a conventional accelerator)



Requires:
- short (~ fs) bunch generation
- precise bunch-laser synchronization

→ trapping of background plasma electrons



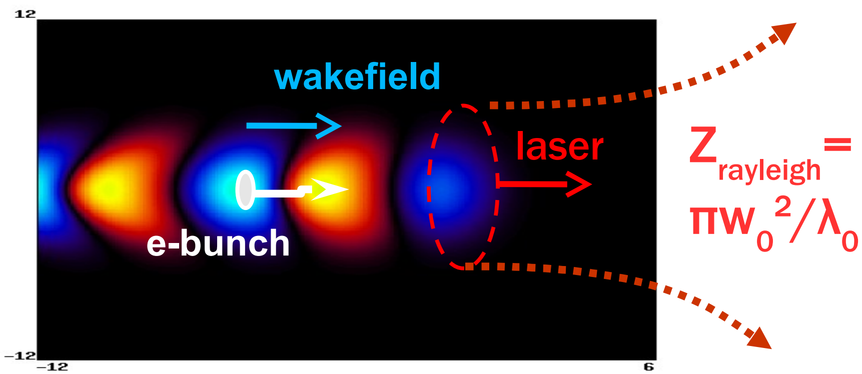
* self-injection (requires high-intensity, high plasma density) → limited control

* controlled injection → use laser(s) and/or tailored plasma to manipulate the plasma wave properties and capture background electrons

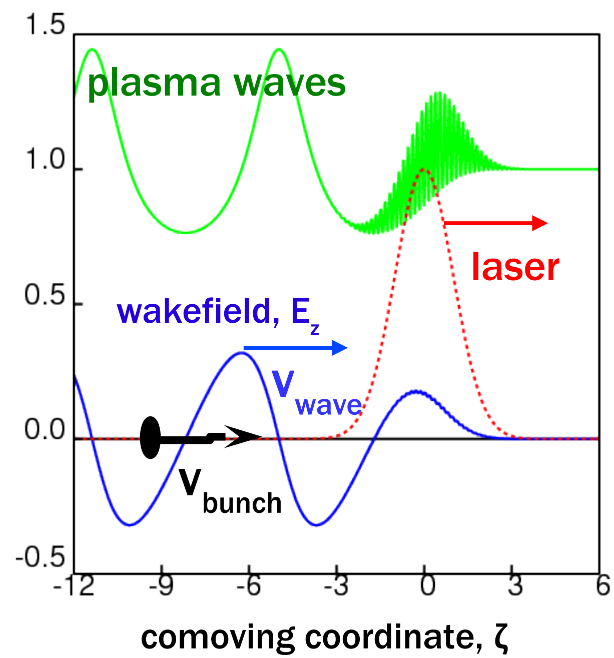
- laser-triggered injection (e.g., colliding pulse)
- ionization-induced injection
- density gradient injection

Limits to energy gain in a (single stage) LPA

- laser diffraction (~ Rayleigh range)
  → mitigated by guiding:
  plasma channel and/or
  self-focusing/self-guiding



- beam-wave dephasing:
  $\beta_{bunch} \approx 1$, $\beta_{wave} \approx 1 - \lambda_0^2/(2\lambda_p^2)$
  → slippage $L_d \approx (\lambda_p/4)/(\beta_{bunch} - \beta_{wave}) \sim n_0^{-3/2}$
  → mitigated by longitudinal density tailoring

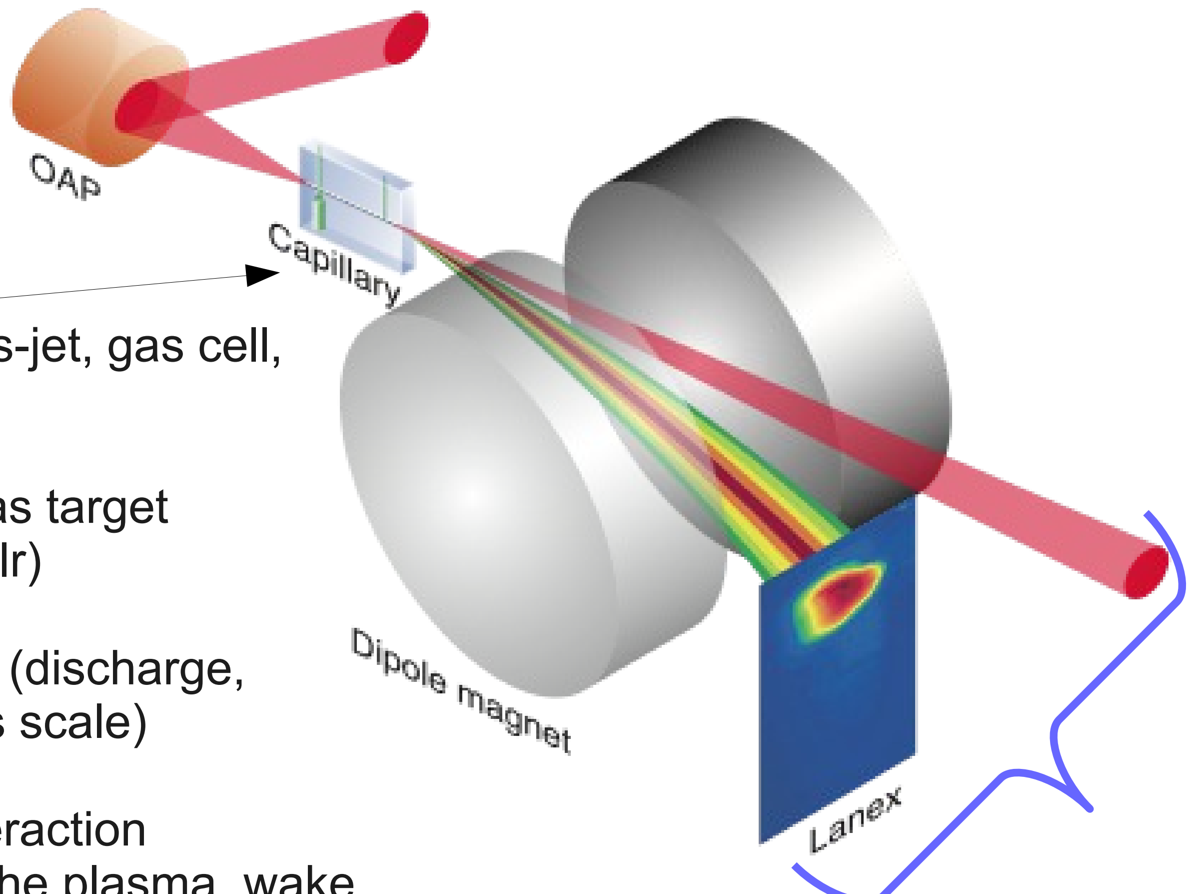


- laser energy depletion → energy loss into plasma wave excitation, $L_{pd} \sim n_0^{-3/2}$
  ($L_{pd} \approx 1$ cm for $n_0 = 10^{18}$ cm^{-3})

Interaction length $\sim n_0^{-3/2}$ Acc. gradient $\sim n_0^{1/2}$	→ Energy gain $\sim n_0^{-1}$

Schematic of a “typical” LPA experiment + modeling needs

Laser pulse [“known”]



Plasma target (gas-jet, gas cell, capillary, etc..):

- Gas dynamics (gas target formation; ~ms scalr)

- Plasma formation (discharge, MHD; 1 ns - 100 ns scale)

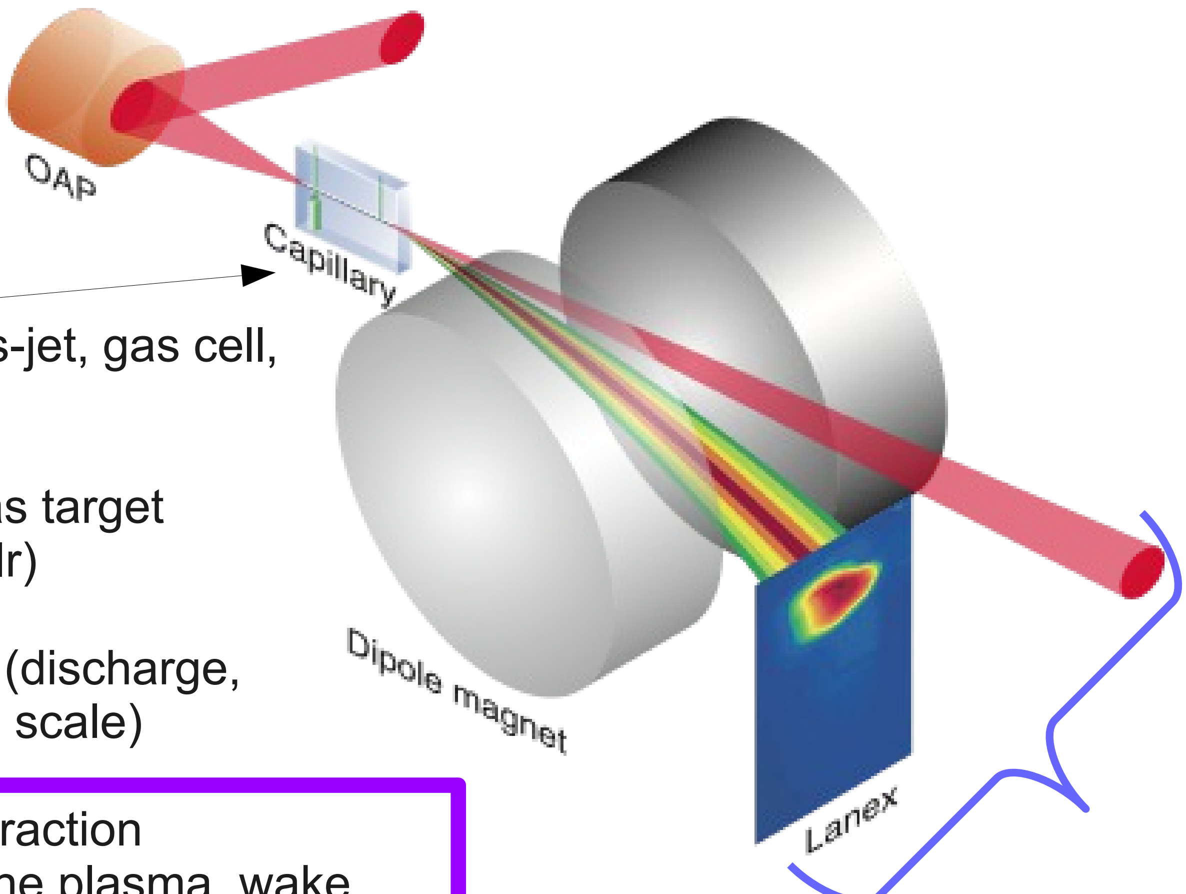
- Laser-plasma interaction (laser evolution in the plasma, wake formation and evolution, [self-]injection, bunch dynamics; ~fs → ~ps scale)

Diagnostics:
- laser (e.g., laser mode, spectrum, etc.)
- bunch (charge, spectrum, divergence, etc.)
- radiation (betatron, etc.)

Bunch transport (transport optics, etc.)

Schematic of a "typical" LPA experiment + modeling needs

Laser pulse ["known"]



Plasma target (gas-jet, gas cell, capillary, etc..):

- Gas dynamics (gas target formation; ~ms scalr)

- Plasma formation (discharge, MHD; 1 ns - 100 ns scale)

- Laser-plasma interaction (laser evolution in the plasma, wake formation and evolution, [self-]injection, bunch dynamics; ~fs → ~ps scale)

← Computationally expensive part!

Diagnostics:
- laser (e.g., laser mode, spectrum, etc.)
- bunch (charge, spectrum, divergence, etc.)
- radiation (betatron, etc.)

Bunch transport (transport optics, etc.)

Laser-plasma interaction physics in LPAs described via Maxwell-Vlasov equations

- **Statistical*** description for the plasma in the 6D (**r**,**p**) phase-space
→ phase-space distribution function $f_s(\mathbf{r},\mathbf{p},t)d\mathbf{r}d\mathbf{p}$ = # particles (s=electron, ion) located between **r** and **r**+d**r** with a momentum between **p** and **p**+d**p** at time t

- Evolution of the distribution → **Vlasov** equation (collisionless plasma)

$$\frac{\partial f_s}{\partial t} + \mathrm{v} \cdot \frac{\partial f_s}{\partial \mathrm{r}} + q_s \left(\mathbf{E} + \frac{v}{c} \times \mathbf{B} \right) \cdot \frac{\partial f_s}{\partial \mathbf{p}} = 0$$

} Plasma dynamics

- Evolution of the fields **E**(**r**, t), **B**(**r**, t) → **Maxwell** equations

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E} \quad \frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{J}$$

} Laser + Wakefield dynamics

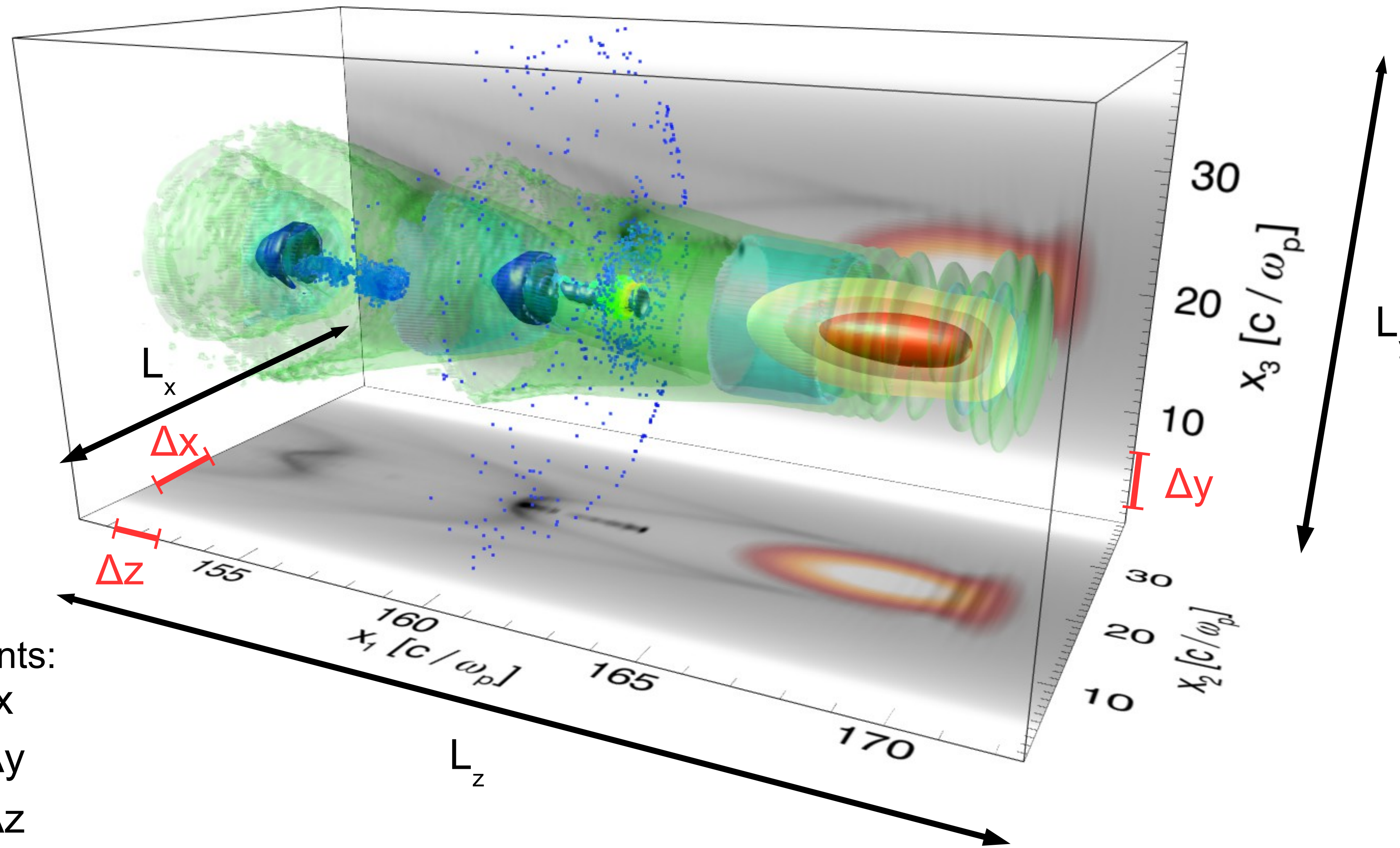
- Coupling between Vlasov ↔ Maxwell

$$\mathbf{J} = \sum_s q_s \int \mathbf{v} f_s(\mathbf{p}, \mathbf{r}, t) d\mathbf{p}$$

$n(\mathbf{r},t)=\int f_s(\mathbf{r},\mathbf{p},t)d^3p$

Numerical solution of M-V equations requires using grids (spatial, phase-space) to represent physical quantities

- E.g., 3D grid for density



Grid points:

$N_x \approx L_x/\Delta x$

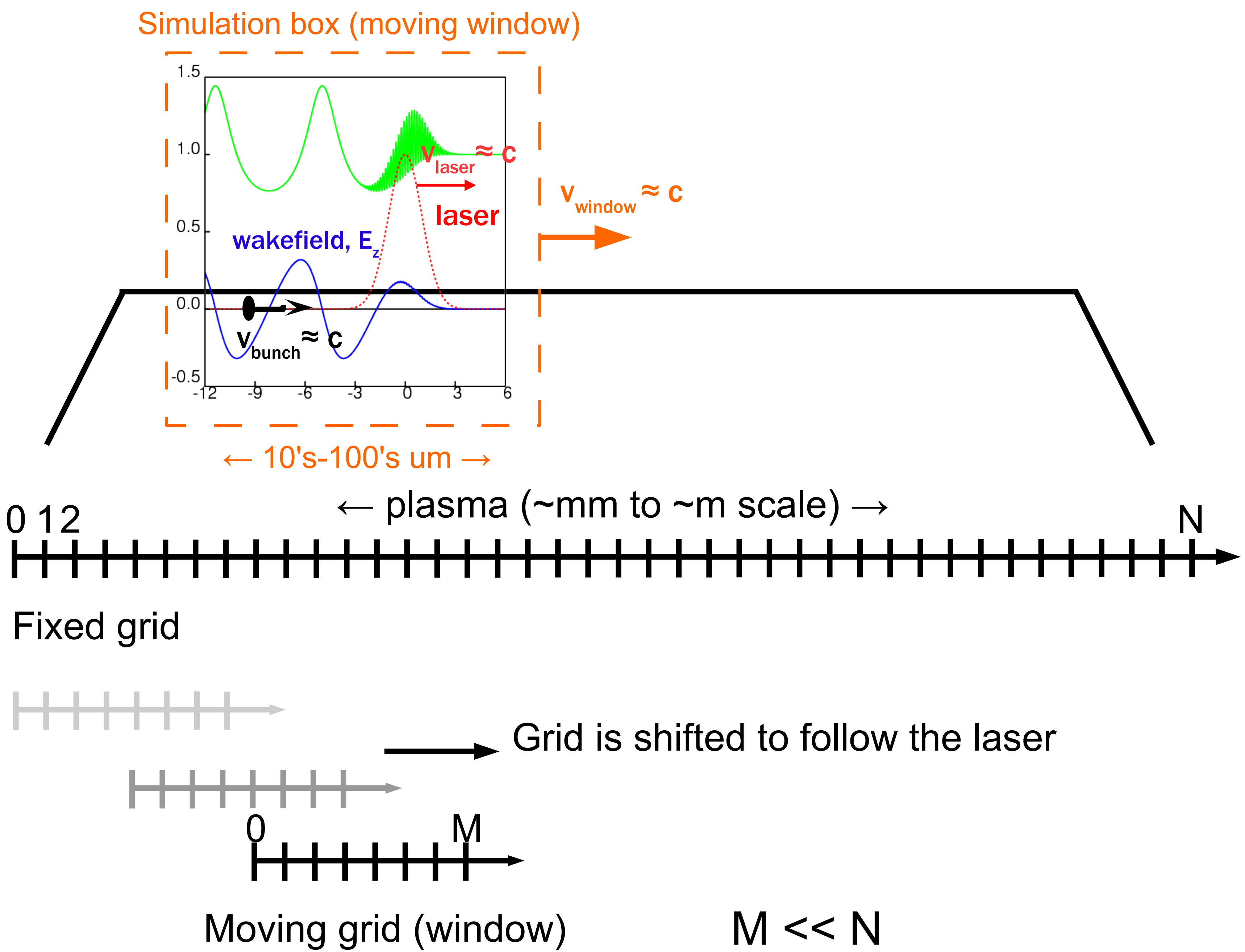
$N_y \approx L_y/\Delta y$

$N_z \approx L_z/\Delta z$

$\rightarrow N_{3D} = N_x * N_y * N_z$

*OSIRIS simulation

Use of a moving computational box greatly reduces memory requirements for LPA simulations



Numerical solution of the Maxwell-Vlasov equations: direct solution

- Direct numerical solution of MV equations is **unfeasible**

E(**r**), B(**r**), J(**r**), ... → discretized on a 3D spatial grid

f_s(**r**, **p**, t) → discretized on a 6D = 3D (space) x 3D (momentum) phase-space grid

Ex: Plasma: $n_0 \sim 10^{18}$ cm^{-3} → $\lambda_p \sim 30$ um
Laser: $\lambda_0 \sim 1$ um, $L_0 \sim 10$ um, $w_0 \sim 30$ um

3D spatial grid:
$L_x \sim L_y \sim L_z \sim 100$ um [a few plasma lengths]
$\Delta x \sim \Delta y \sim \lambda_p/60$ [transverse]
$\Delta z \sim \lambda_0/30$ [longitudinal]
$N_x \sim N_y \sim 200$, $N_z \sim 3000$

3D momentum grid:
$L_{px} \sim L_{py} \sim 10$ mc [transverse]
$L_{pz} \sim 2000$ mc [longitudinal]
$\Delta px \sim \Delta py \sim \Delta pz \sim mc/10$ [transverse]
$N_{px} \sim N_{py} \sim 100$, $N_{pz} \sim 20000$

$N_{3D} = N_x * N_y * N_z \sim 1.2\text{x}10^8$ points
$N_{6D} = N_{3D} * N_{px} * N_{py} * N_{pz} \sim 2\text{x}10^{16}$ points

→ Representing 1 double precision quantity (f_s) on a grid with N_{6D} points requires >**200 PBytes** of memory ==> **UNFEASIBLE**!!!

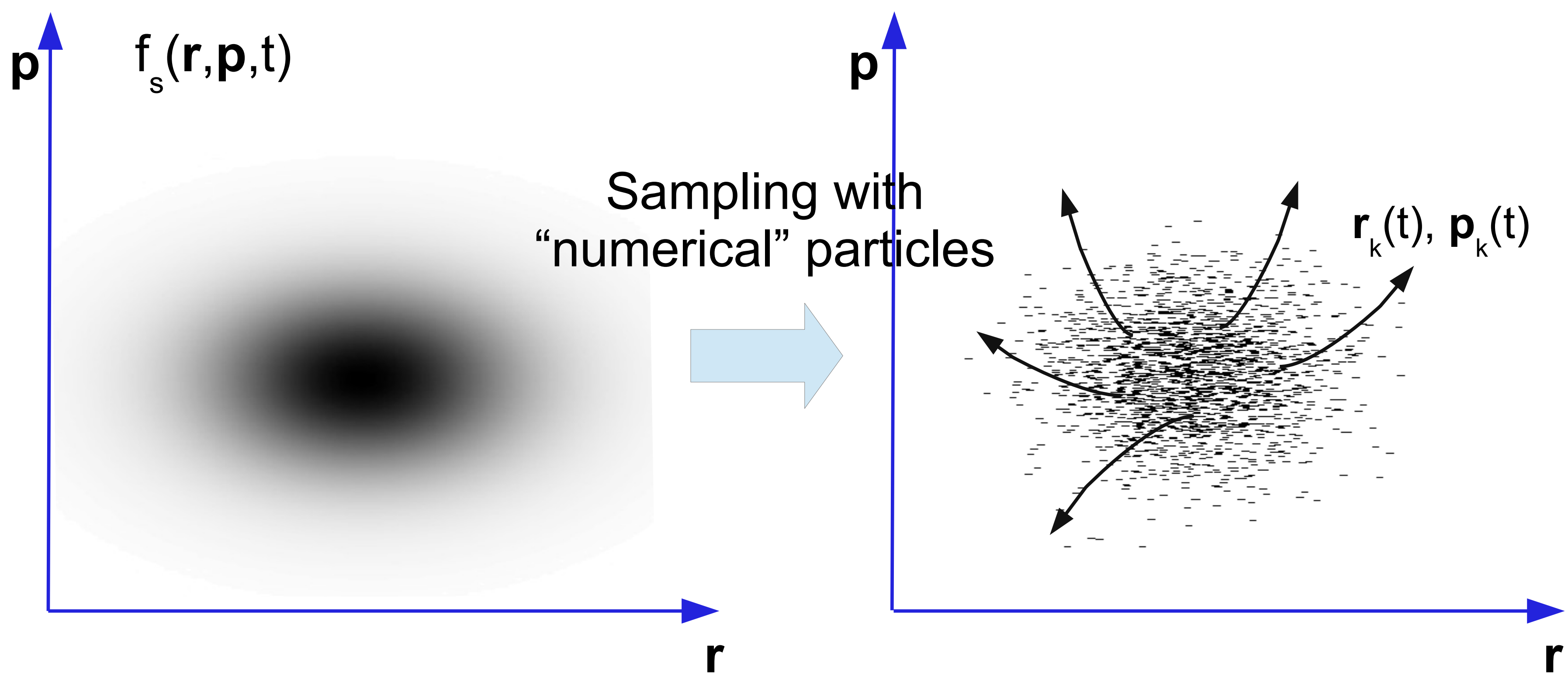
Numerical solution of the Maxwell-Vlasov equations: particle method (PIC)/1

- Vlasov equation solved using a **particle method** (+ 3D spatial grid for the fields)

$$f_s(\mathbf{r},\mathbf{p},t) = (1/N_s)\sum_k g[\mathbf{r}-\mathbf{r}_k(t)]\delta[\mathbf{p}-\mathbf{p}_k(t)]$$

Particle "shape"
(finite spatial extent)

$g \rightarrow$ "compact support" function $\int g(\mathbf{r})d\mathbf{r}=1$
$\delta \rightarrow$ Dirac function
$N_s \rightarrow$ # "numerical" particles
$\mathbf{r}_k(t)$, $\mathbf{p}_k(t) \rightarrow$ phase-space orbit of the k-th "numerical" particle (Vlasov characteristic)



Numerical solution of the Maxwell-Vlasov equations: particle method (PIC)/2

- Equation for the characteristics of the Vlasov equation

$$\frac{\partial f_s}{\partial t} + \mathrm{v}\cdot\frac{\partial f_s}{\partial \mathrm{r}} + q_s\left(\mathbf{E} + \frac{v}{c}\times\mathbf{B}\right)\cdot\frac{\partial f_s}{\partial \mathrm{p}} = 0$$

$$f_s(\mathbf{r},\mathbf{p},t) = (1/N_s)\sum_k g[\mathbf{r}-\mathbf{r}_k(t)]\delta[\mathbf{p}-\mathbf{p}_k(t)]$$

→

$$d\mathbf{r}_k/dt=\mathbf{v}_k=\mathbf{p}_k/m\gamma_k$$

$$d\mathbf{p}_k/dt= q_s[\mathbf{E}_k+(\mathbf{v}_k/c)\mathbf{B}_k]$$

where $\mathbf{E}_k = \int\mathbf{E}(r)g(\mathbf{r}-\mathbf{r}_k)d\mathbf{r}$,

$\mathbf{B}_k = \int\mathbf{B}(r)g(\mathbf{r}-\mathbf{r}_k)d\mathbf{r}$

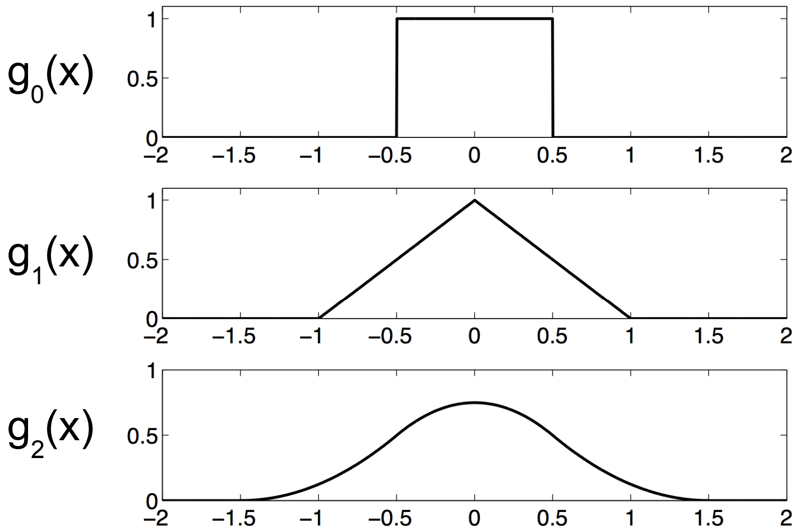
==> evolution of f_s described via the motion of a “swarm” of numerical particles

- Expressing the current density using numerical particles

$$\mathbf{J} = \sum_s q_s\int \mathbf{v} f_s(\mathbf{p},\mathbf{r},t)d\mathbf{p} \longrightarrow \mathbf{J}=\sum_s(q_s/N_s)\sum_{k=0,\ Ns}\mathbf{v}_k g[\mathbf{r}-\mathbf{r}_k(t)]$$

Numerical solution of the Maxwell-Vlasov equations: particle method (PIC)/3

- Example of particle shapes: $g(\mathbf{r})=g_x(x)g_y(y)g_z(z)$



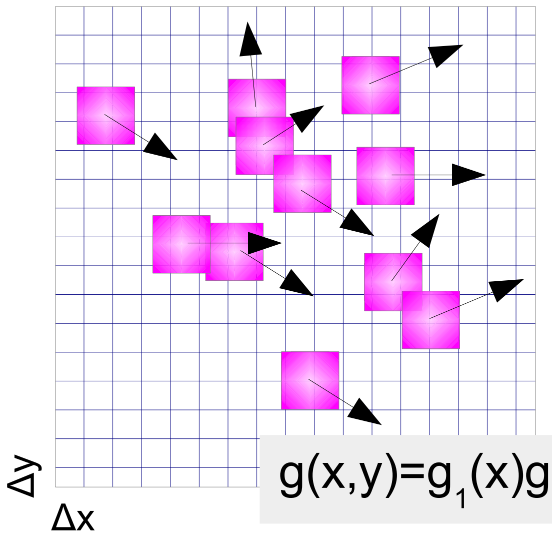
[Lapenta]

Spatial extension = Δx

Spatial extension = 2Δx

Spatial extension = 3Δx

Numerical particles on the spatial grid
("clouds" of charge)



$g(x,y)=g_1(x)g_1(y)$

→ describes interaction particles ↔ grid
→ finite spatial extension (to limit number of calculations)
→ type of shape controls noise in simulation (higher order reduces noise)

Memory requirements for the solution of Maxwell-Vlasov equations using the PIC technique

Plasma: $n_0 \sim 10^{18}$ cm$^{-3} \rightarrow \lambda_p \sim 30$ um

Laser: $\lambda_0 \sim 1$ um, $L_0 \sim 10$ um, $w_0 \sim 30$ um

3D spatial grid:

$L_x \sim L_y \sim L_z \sim 100$ um [a few plasma lengths]

$\Delta x \sim \Delta y \sim \lambda_p/60$ [transverse]

$\Delta z \sim \lambda_0/30$ [longitudinal]

$N_x \sim N_y \sim 200$, $N_z \sim 3000$

$N_{3D} = N_x * N_y * N_z \sim 1.2 \times 10^8$ points

Particles:

$N_{ppc} = 1\text{-}100$

$N_{tot} = N_{3D} * N_{ppc} \sim 10^8\text{-}10^{10}$ particles

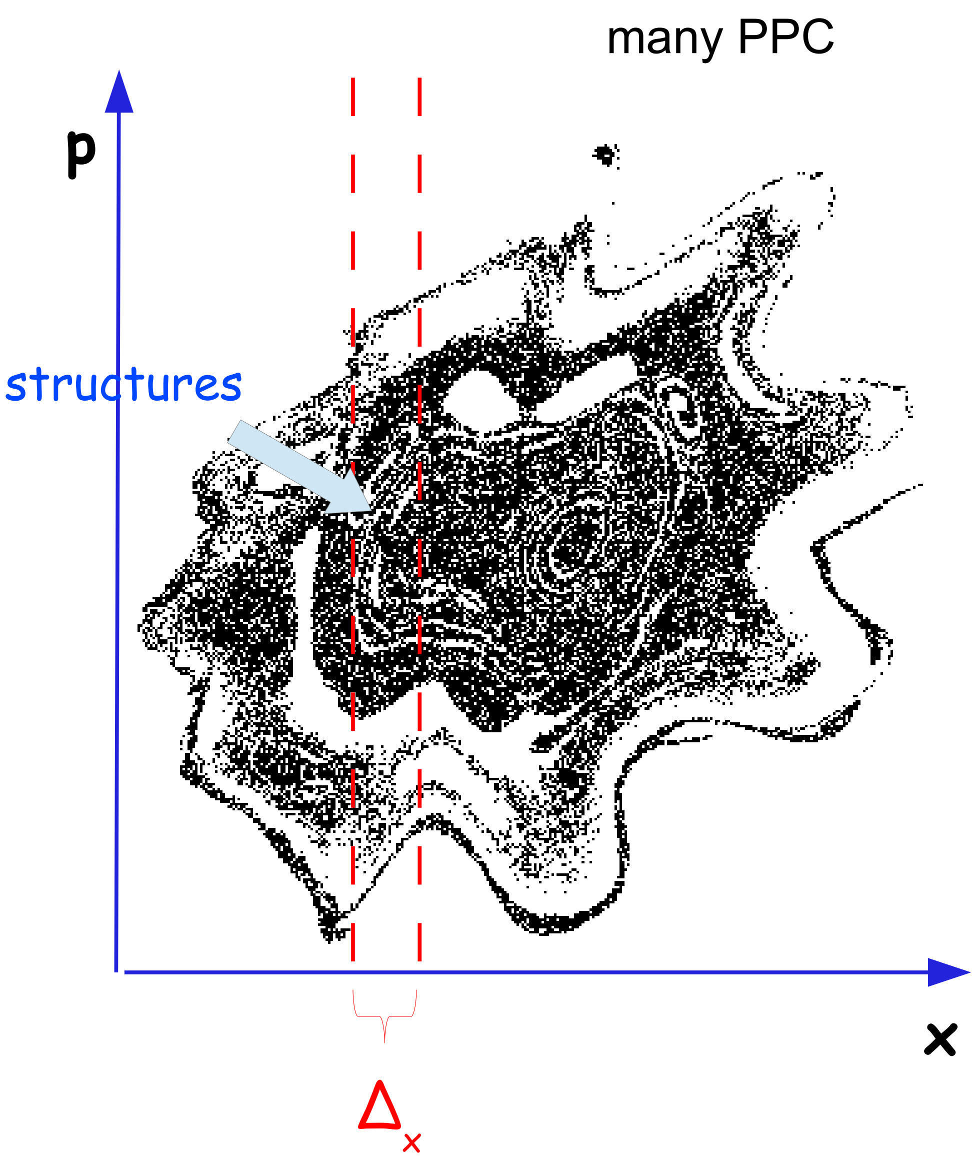
Grid $\rightarrow$ (9 fields) x (8 bytes) x N_{3D} ~ 7 GBytes

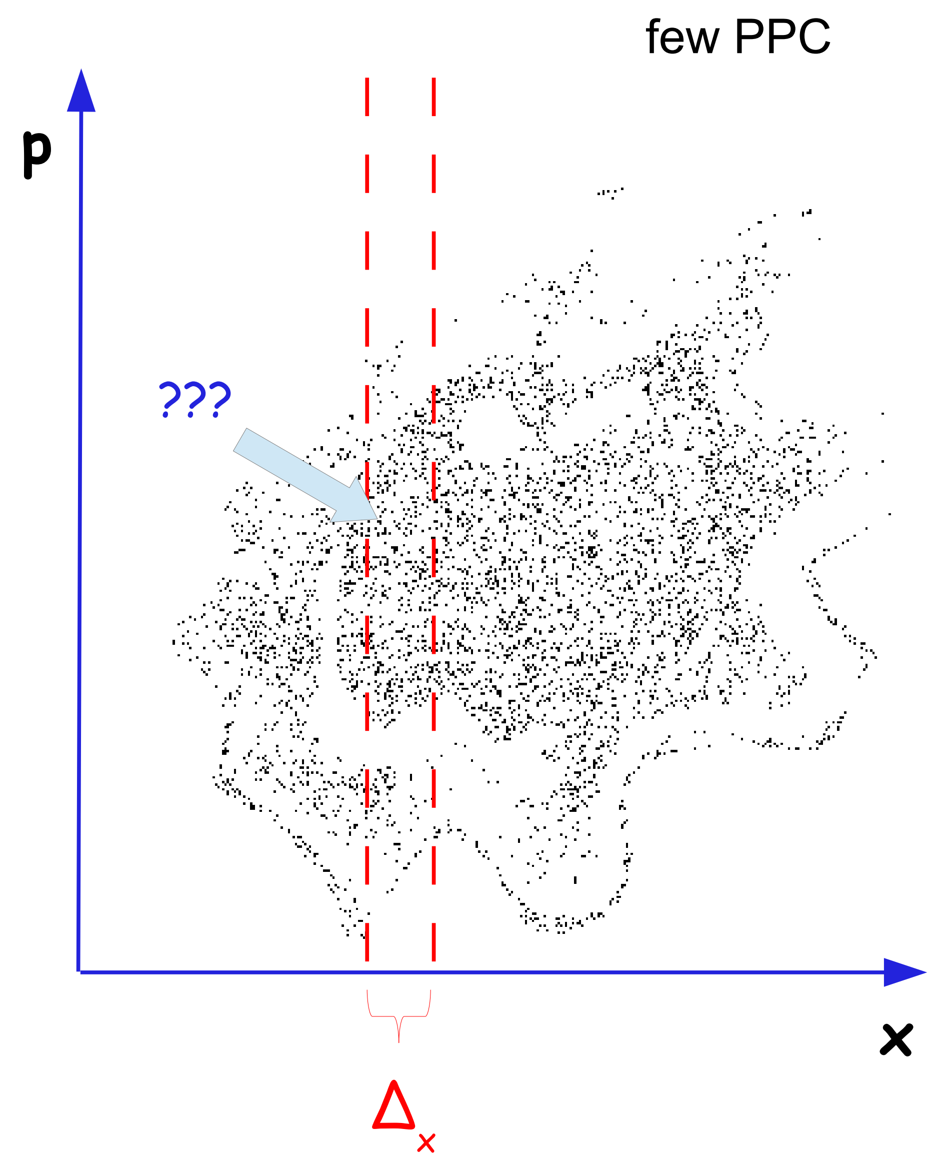
Particles $\rightarrow$ (6 coordinates) x (8 bytes) x N_{tot} ~ (5-500) GBytes

Memory requirements OK!!!

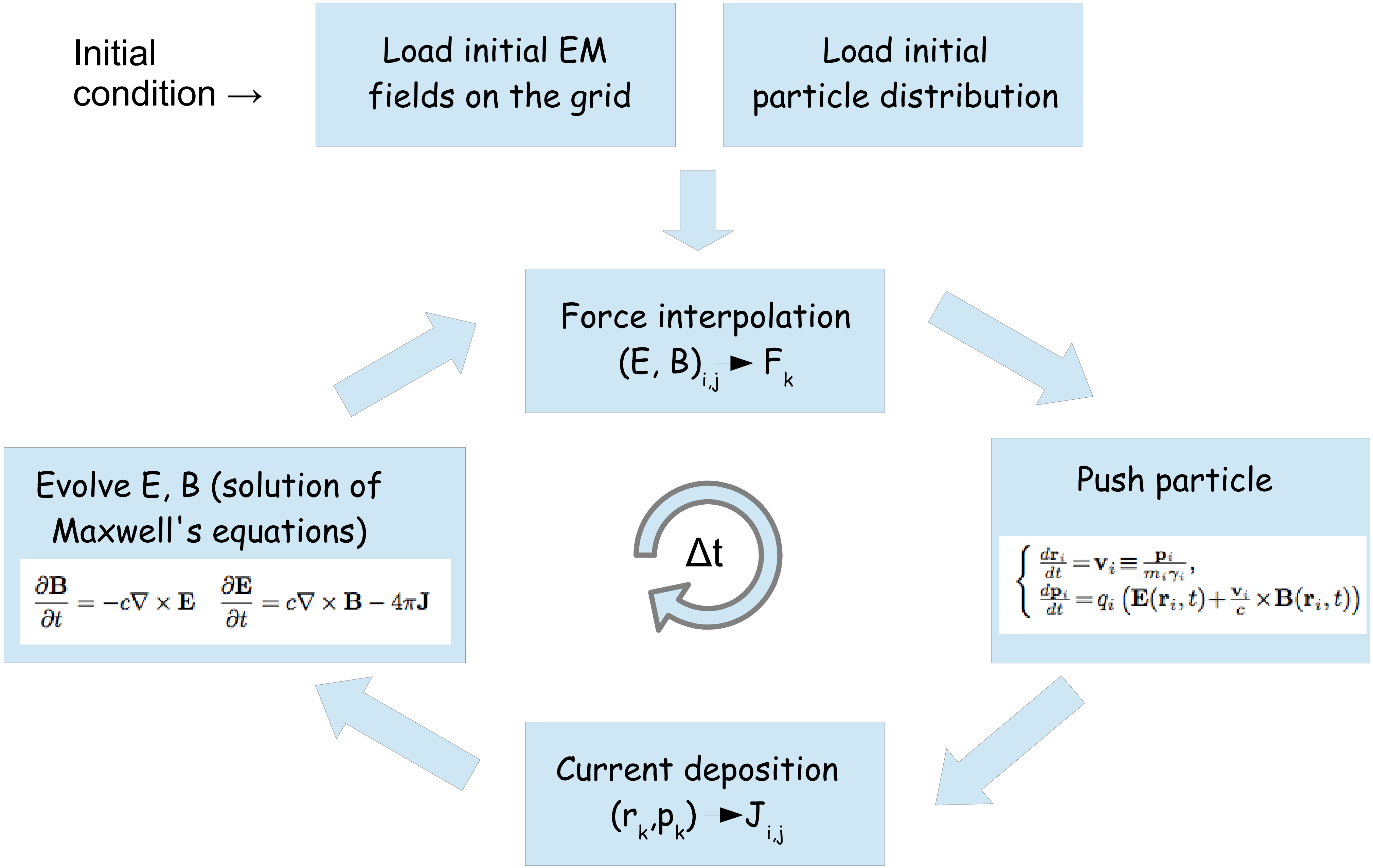
Edison @ NERSC (10^5 CPUs) = 360 TBytes, 2.6 Pflops/s

Resolution in momentum space depends on number of “numerical” particles per cell

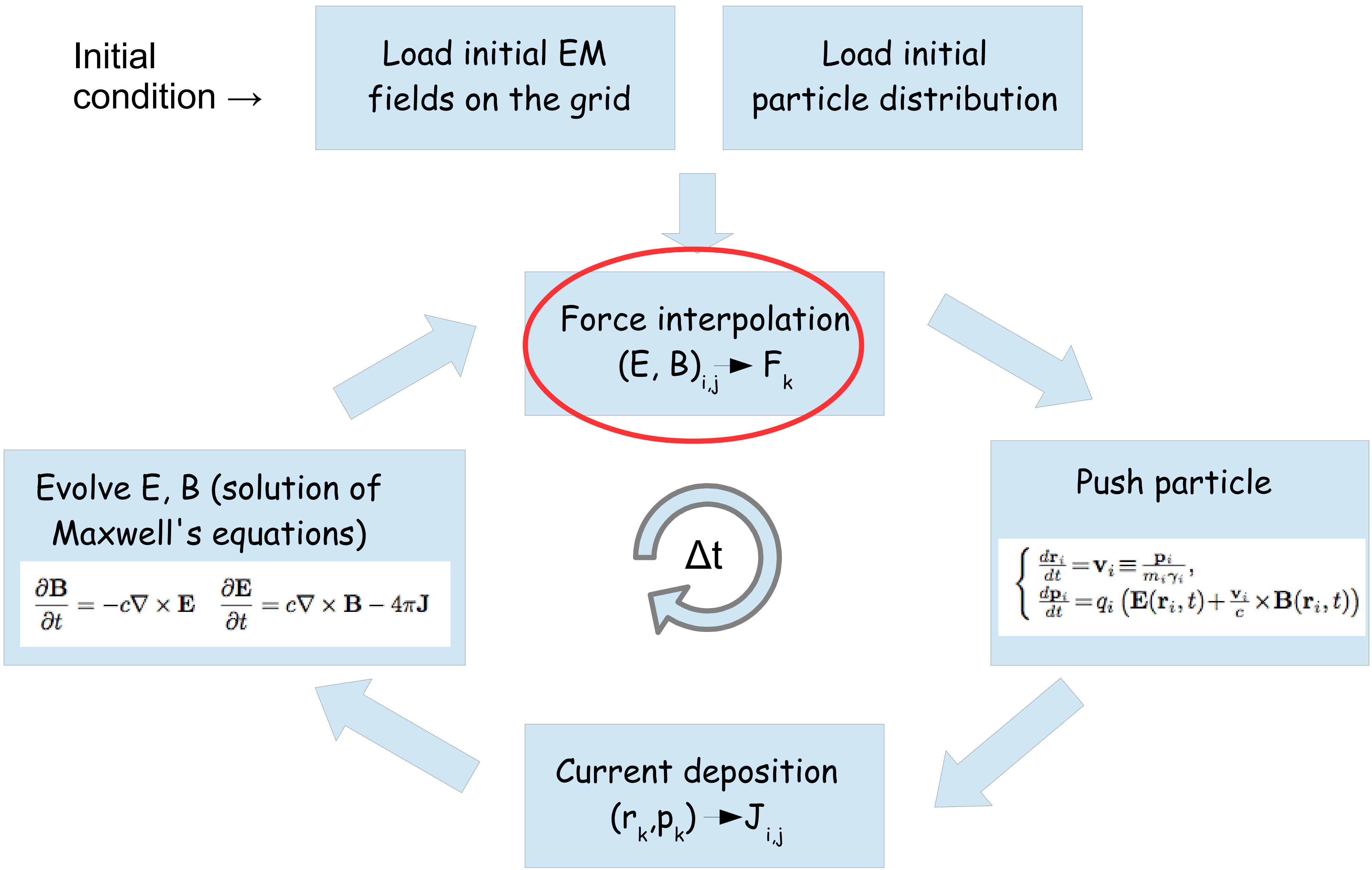




The PIC loop: self-consistent solution of Maxwell-Vlasov equations



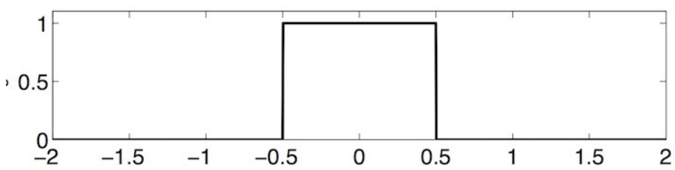
The PIC loop: self-consistent solution of Maxwell-Vlasov equations

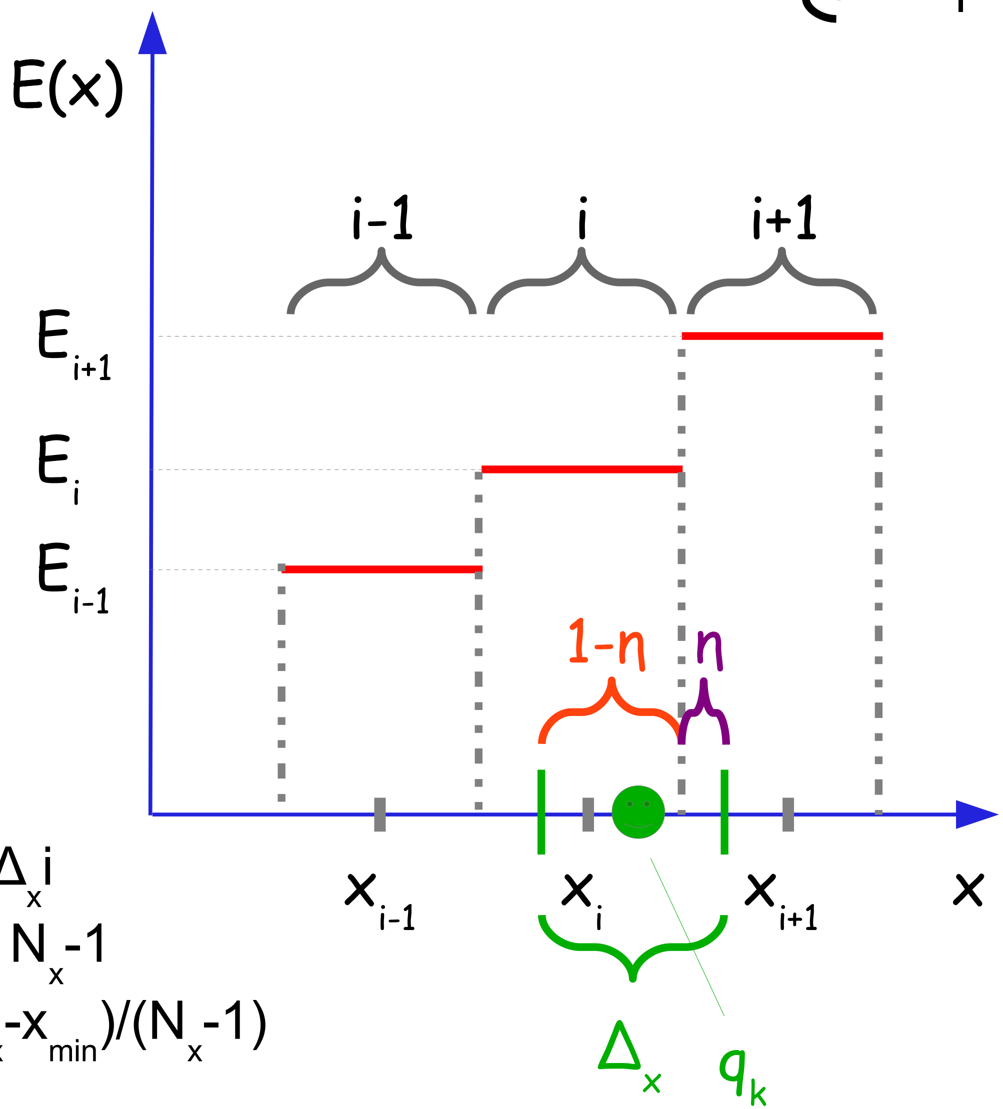


Force interpolation: grid → particle [1D]

$$E_k = \int E(x) g(x - q_k) dx$$

$$g_0(x) = \begin{cases} 1/\Delta_x & |x| < \Delta_x/2 \\ 0 & |x| > \Delta_x/2 \end{cases}$$





$\eta = (q_k - x_i)/\Delta_x$

$\langle E \rangle (x = q_k) = (1-\eta) E_i + \eta E_{i+1}$

if $q_k = x_i \rightarrow \langle E \rangle = E_i$

if $q_k = x_{i+1/2} \rightarrow \langle E \rangle = (E_i + E_{i+1})/2$

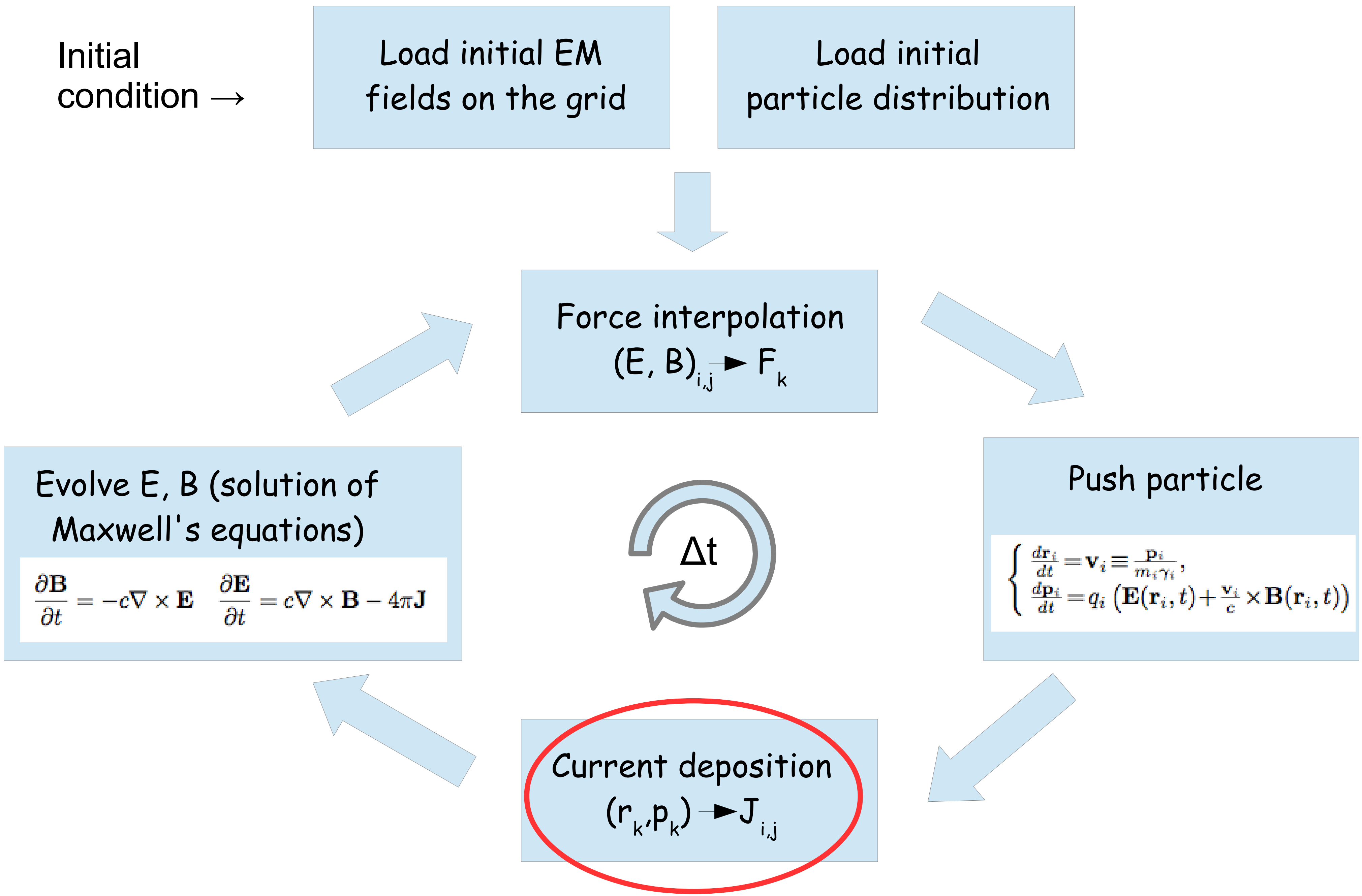
if $q_k = x_{i+1} \rightarrow \langle E \rangle = E_{i+1}$

$x_i = x_{min} + \Delta_x i$

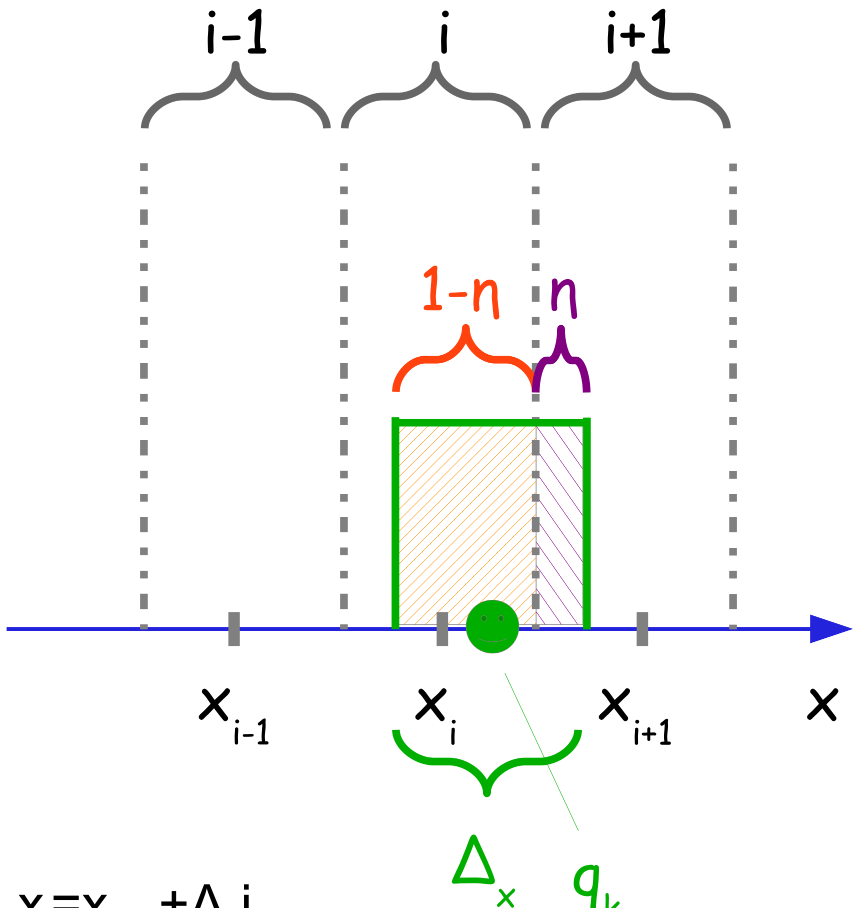
$i = 0, 1, .., N_x - 1$

$\Delta_x = (x_{max} - x_{min})/(N_x - 1)$

The PIC loop: self-consistent solution of Maxwell-Vlasov equations



Current deposition: particle → grid [1D]



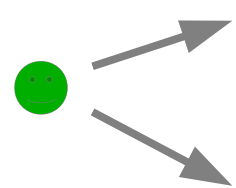
$$J_i = \frac{1}{\Delta_x}\int_{x_i-\Delta_x/2}^{x_i+\Delta_x/2} J(x)dx$$

$$g_0(x)=\begin{cases} 1/\Delta x & |x|<\Delta_x/2 \\ 0 & |x|>\Delta_x/2 \end{cases}$$

$$\eta=(q_k-x_i)/\Delta_x$$

==> Charge distributed between the grid points i and i+1

$x_i=x_{min}+\Delta_x i$
$i=0,1,.., N_x-1$
$\Delta_x=(x_{max}-x_{min})/(N_x-1)$

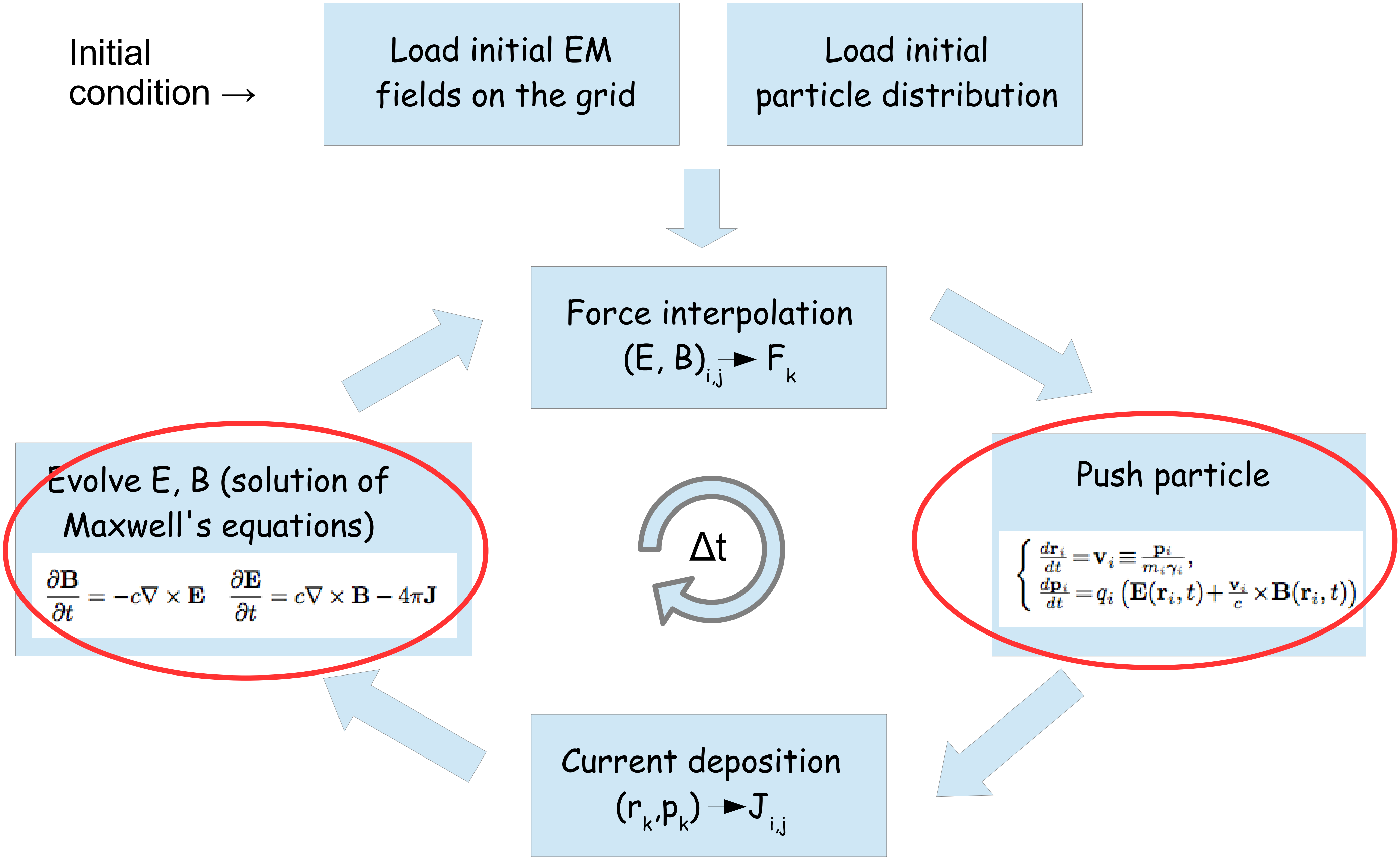


J_i += $(1-\eta)*(q_s u_k/m\gamma_k)*(1/\Delta_x)*(1/N_s)$

J_{i+1} += $\eta*(q_s u_k/m\gamma_k)*(1/\Delta_x)*(1/N_s)$

N.B. Using the same scheme to perform force interpolation and current deposition gives no self-force on the particle.

The PIC loop: self-consistent solution of Maxwell-Vlasov equations



Major criteria to chose algorithms in a PIC code

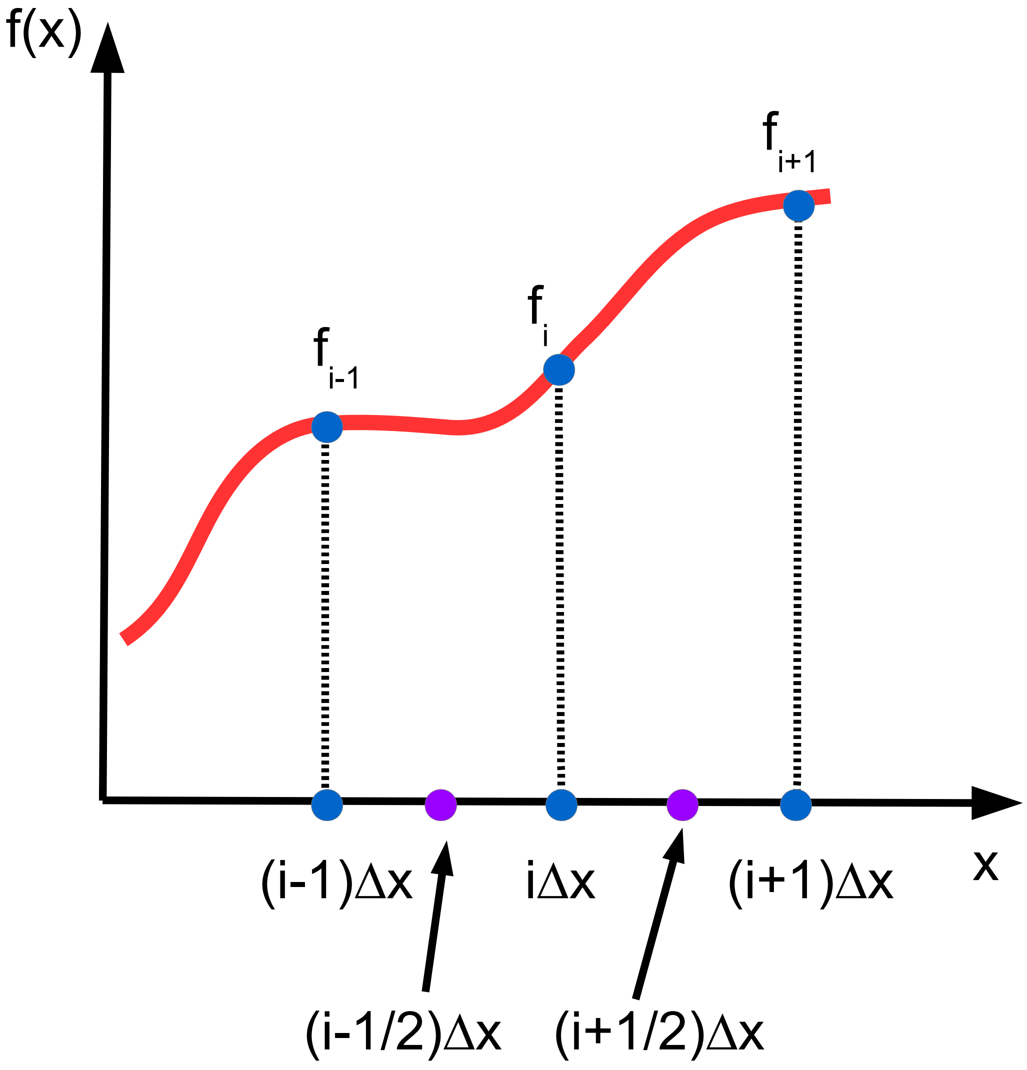
Integration of Maxwell's equations and particle's equations of motion requires solving PDEs and ODEs → discretized numerical solution

Properties of numerical schemes:

- **Convergence** → the numerical solution goes to the analytical one if Δ_x, Δ_y, Δ_z and Δ_t go to zero.
- **Accuracy** → scaling of the truncation error with Δ_x, Δ_y, Δ_z and Δ_t.
- **Stability** → if total errors (truncation + round-off) grows in time then the scheme is unstable.
- **Efficiency** → computational cost of the algorithm.
- **Dissipation** → dissipation of some physical quantity due to truncation error.
- **Conservation** → deviation of the conservation law caused by the truncation error.

Discretization of (spatial and temporal) derivatives

$x \rightarrow$ space or time variable
$\Delta x \rightarrow$ discretization step
$f(x) \rightarrow$ some function of x



Derivatives of f(x) [using Taylor's expansion]:

1st order:

- $df/dx|_i=(f_{i+1}-f_i)/\Delta x + O(\Delta x)$
- $df/dx|_i=(f_i-f_{i-1})/\Delta x + O(\Delta x)$

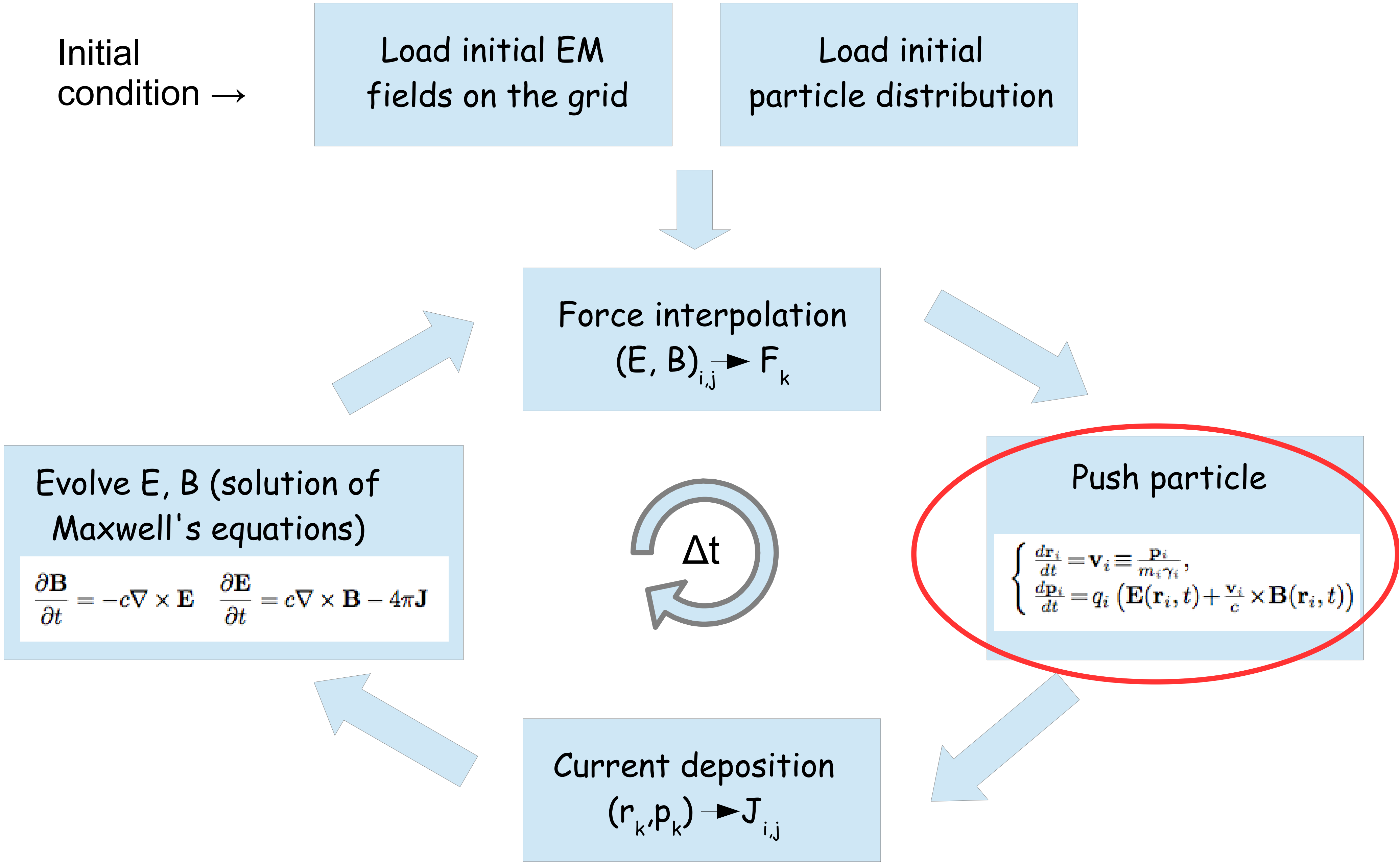
2nd order:

- $df/dx|_i=(f_{i+1}-f_{i-1})/(2\Delta x) + O(\Delta x^2)$
- $df/dx|_{i+1/2}=(f_{i+1}-f_i)/\Delta x + O(\Delta x^2)$
- $df/dx|_{i-1/2}=(f_i-f_{i-1})/\Delta x + O(\Delta x^2)$

$\rightarrow$ centering easy way to construct 2nd order scheme

$\rightarrow$ time integration of an ODE requires at least a 2nd order scheme in order to provide meaningful results

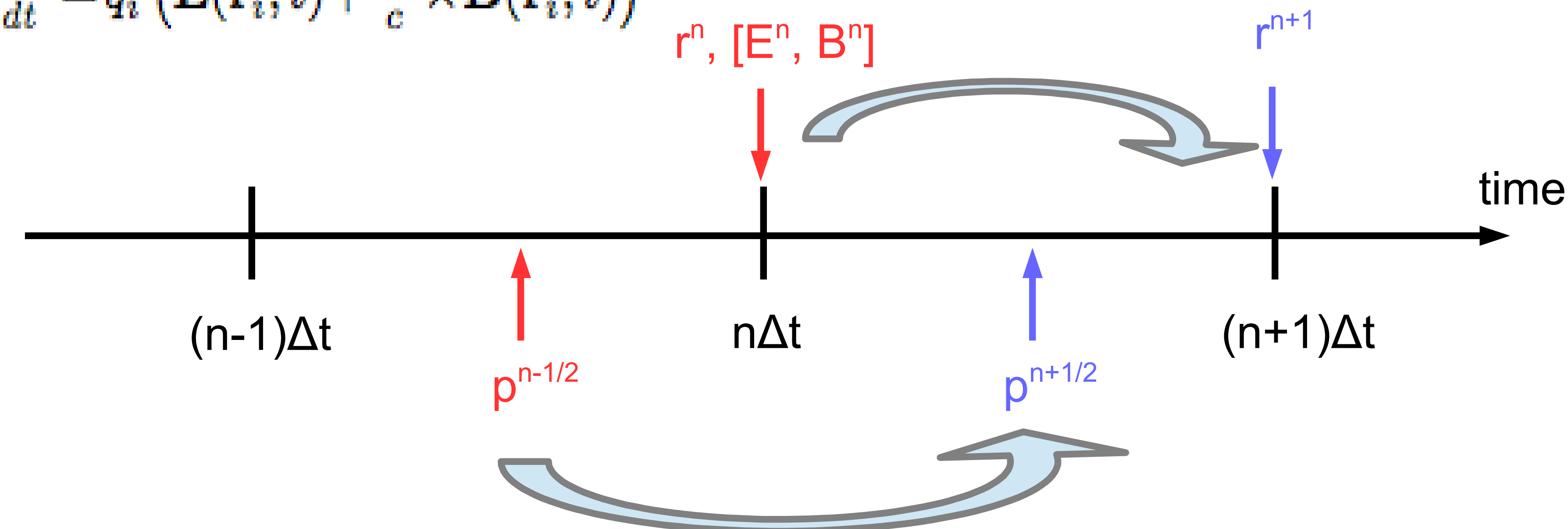
The PIC loop: self-consistent solution of Maxwell-Vlasov equations



Particle pusher (2nd order "leapfrog" scheme)

$$\begin{cases} \frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \equiv \frac{\mathbf{p}_i}{m_i \gamma_i}, \\ \frac{d\mathbf{p}_i}{dt} = q_i \left(\mathbf{E}(\mathbf{r}_i, t) + \frac{\mathbf{v}_i}{c} \times \mathbf{B}(\mathbf{r}_i, t) \right) \end{cases}$$

Position and momentum are **staggered in time** → 2nd order accurate scheme!



Implicit equation!

$(d\mathbf{p}/dt)^n \rightarrow (\mathbf{p}^{n+1/2} - \mathbf{p}^{n-1/2})/\Delta t$

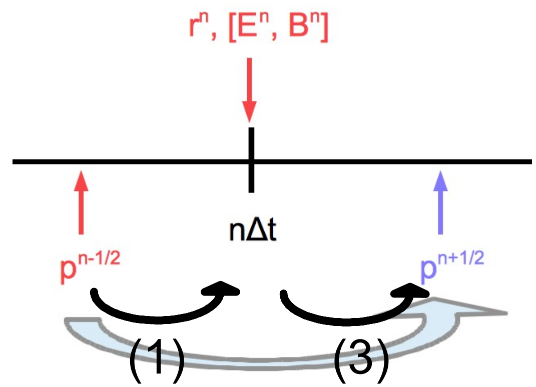
$\mathbf{v}^n/c \rightarrow (\mathbf{p}^{n+1/2} + \mathbf{p}^{n-1/2})/(2mc\gamma^n)$

$$\frac{\mathbf{p}^{n+1/2} - \mathbf{p}^{n-1/2}}{\Delta t} = q \left(\mathbf{E}^n + \frac{\mathbf{p}^{n+1/2} + \mathbf{p}^{n-1/2}}{2mc\,\gamma^n} \times \mathbf{B}^n \right)$$

$$(d\mathbf{r}/dt)^{n+1/2} \rightarrow \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\Delta t} = \mathbf{v}^{n+1/2} = \mathbf{p}^{n+1/2}/(m\,\gamma^{n+1/2})$$

Solution of momentum equation with Boris scheme (explicit)

Boris scheme (2nd order, time reversible) separates the contributions of electric and magnetic fields in the motion of the particle



→ (1) momentum change due to **E** (1/2 kick)

$$\mathbf{p}^{n-1/2} \rightarrow \mathbf{p}^{-} = \mathbf{p}^{n-1/2} + q\,\mathbf{E}^{n}\,(\Delta t/2)$$

→ (2) rotation of $\mathbf{p}^{-}$ due to **B** (particle energy does not change)

$$\gamma^{n} = [1+(\mathbf{p}^{-}/mc)^{2}\,]^{1/2} \qquad \mathbf{t} = q\Delta t\mathbf{B}^{n}/2mc\gamma^{n} \qquad \mathbf{s} = 2\mathbf{t}/(1+|\mathbf{t}|^{2})$$

$$\mathbf{p}' = \mathbf{p}^{-} + \mathbf{p}^{-} \times \mathbf{t}$$

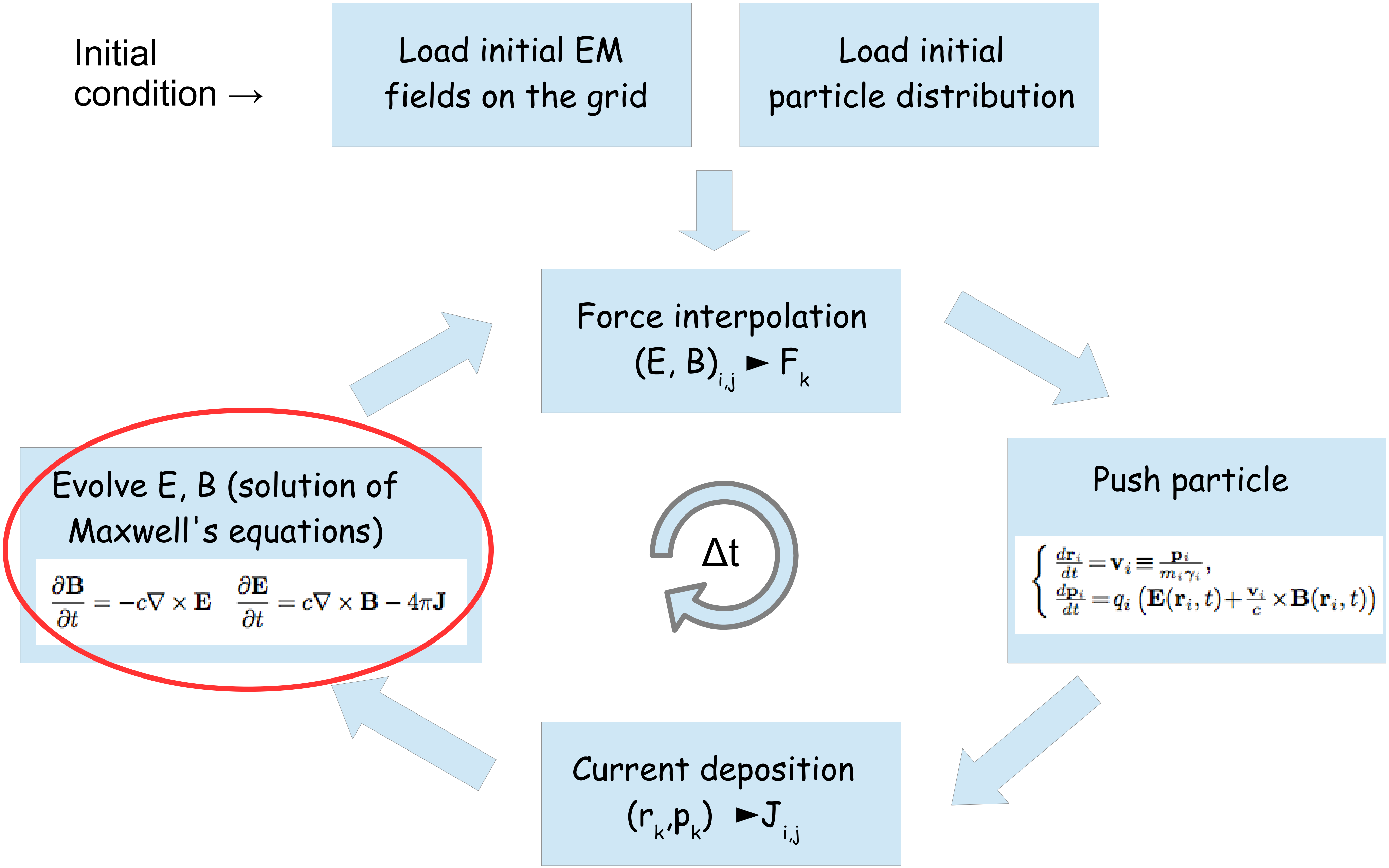
$$\mathbf{p}^{+} = \mathbf{p}^{-} + \mathbf{p}' \times \mathbf{s}$$

$\mathbf{p}^{-} \rightarrow \mathbf{p}^{+}$: rotation around $\mathbf{B}^{n}$ by an angle $\arctan[q\Delta t\mathbf{B}^{n}/2mc\gamma^{n}]$

→ (3) momentum change due to **E** (1/2 kick)

$$\mathbf{p}^{+} \rightarrow \mathbf{p}^{n+1/2} = \mathbf{p}^{+} + q\,\mathbf{E}^{n}\,(\Delta t/2)$$

The PIC loop: self-consistent solution of Maxwell-Vlasov equations



Field solver (2nd order finite-difference time-domain "Yee" scheme): time discretization

$$\begin{cases} \dfrac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{J} \\[2ex] \dfrac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E} \end{cases}$$

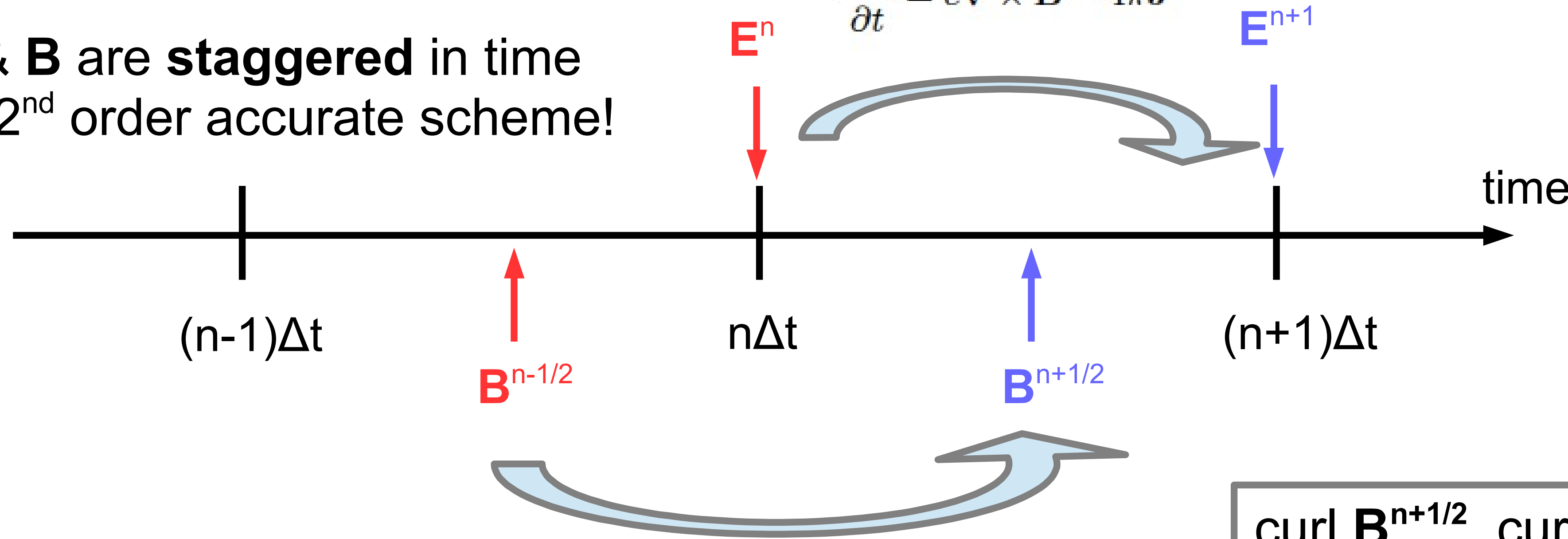
$\mathbf{E}^{n+1} = \mathbf{E}^{n} + \Delta t\,[c\Delta \mathrm{x}\mathbf{B}^{n+1/2} - 4\pi \mathbf{J}^{n+1/2}]$

$(\partial \mathbf{E}/\partial t)^{n+1/2} \rightarrow (\mathbf{E}^{n+1} - \mathbf{E}^{n})/\Delta t$

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{J}$$

E & **B** are **staggered** in time

→ 2nd order accurate scheme!



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

$(\partial \mathbf{B}/\partial t)^{n} \rightarrow (\mathbf{B}^{n+1/2} - \mathbf{B}^{n-1/2})/\Delta t$

curl $\mathbf{B}^{n+1/2}$, curl $\mathbf{E}^{n}$ → computed numerically on the grid

To push particle we need:

$\mathbf{B}^{n} = (\mathbf{B}^{n-1/2} + \mathbf{B}^{n+1/2})/2$ ← $\mathbf{B}^{n+1/2} = \mathbf{B}^{n-1/2} - c\,\Delta t\,\Delta \mathrm{x}\mathbf{E}^{n}$

Field solver (2nd order finite-difference time-domain "Yee" scheme): space discretization

Rewriting Maxwell's equation in 1D ($E=E_x$, $B=B_y$) and in vacuum (**J**=0) [$c\Delta t=\Delta\tau$]

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} \qquad\qquad \frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}$$

Time discretization (2nd order)

$$(E^{n+1} - E^{n})/\Delta\tau = -\partial B/\partial z|^{n+1/2} \qquad\qquad (B^{n+1/2} - B^{n-1/2})/\Delta\tau = -\partial E/\partial z|^{n}$$

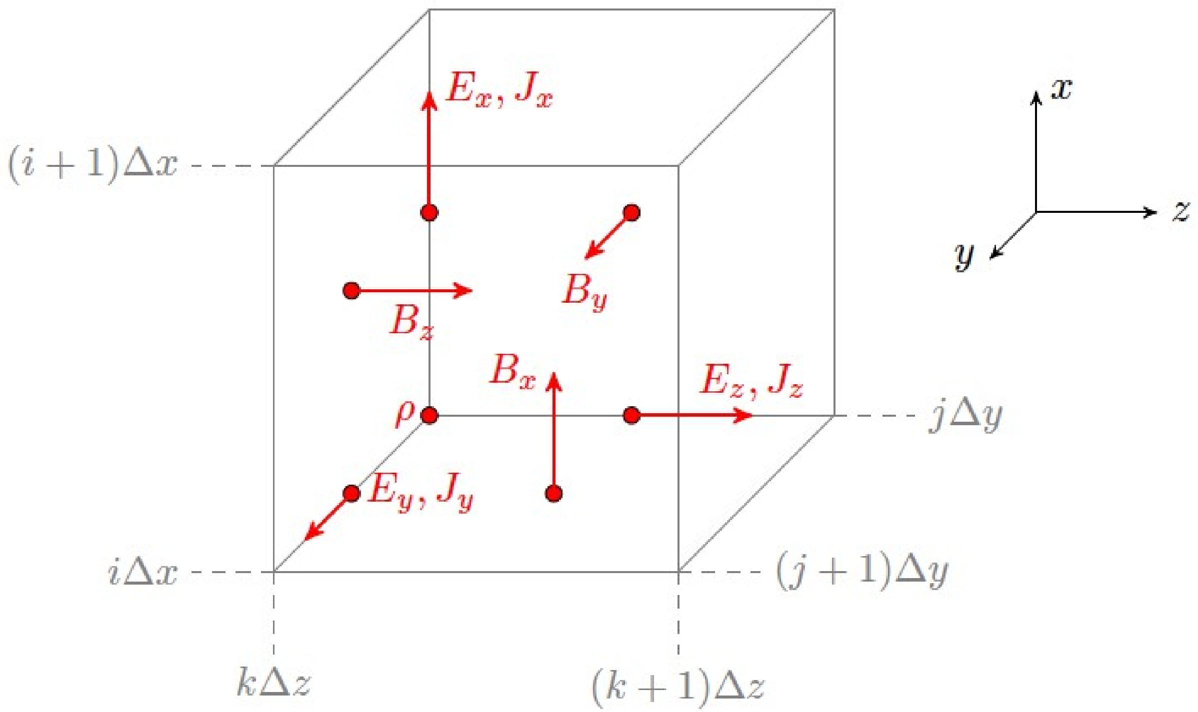
Space discretization (2nd order)

$$(E^{n+1}_{k} - E^{n}_{k})/\Delta\tau = -(B^{n+1/2}_{k+1/2} - B^{n+1/2}_{k-1/2})/\Delta z$$

$$(B^{n+1/2}_{k+1/2} - B^{n-1/2}_{k+1/2})/\Delta\tau = -(E^{n}_{k+1} - E^{n}_{k})/\Delta z$$

E & **B** are **staggered** in space
→ 2nd order accurate scheme!

Field solver (Yee) in 3D: exploits spatial and temporal staggering of fields to obtain 2nd order accurate scheme/1



=> Different components of the different fields are **staggered**, so that all derivatives in the Maxwell equations are centered

Field	Position in space and time				Notation
	x	y	z	t	
E_x	$(i+\frac{1}{2})\Delta x$	$j\Delta y$	$k\Delta z$	$n\Delta t$	${E_x}^n_{i+\frac{1}{2},j,k}$
E_y	$i\Delta x$	$(j+\frac{1}{2})\Delta y$	$k\Delta z$	$n\Delta t$	${E_y}^n_{i,j+\frac{1}{2},k}$
E_z	$i\Delta x$	$j\Delta y$	$(k+\frac{1}{2})\Delta z$	$n\Delta t$	${E_z}^n_{i,j,k+\frac{1}{2}}$
B_x	$i\Delta x$	$(j+\frac{1}{2})\Delta y$	$(k+\frac{1}{2})\Delta z$	$(n+\frac{1}{2})\Delta t$	${B_x}^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}$
B_y	$(i+\frac{1}{2})\Delta x$	$j\Delta y$	$(k+\frac{1}{2})\Delta z$	$(n+\frac{1}{2})\Delta t$	${B_y}^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}$
B_z	$(i+\frac{1}{2})\Delta x$	$(j+\frac{1}{2})\Delta y$	$k\Delta z$	$(n+\frac{1}{2})\Delta t$	${B_z}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}$

Field solver (Yee) in 3D: exploits spatial and temporal staggering of fields to obtain 2nd order accurate scheme/2

Maxwell-Ampère

$$\partial_t E_x|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} = c^2 \partial_y B_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} - c^2 \partial_z B_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} - \mu_0 c^2 j_x{}^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k}$$

$$\partial_t E_y|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k} = c^2 \partial_z B_x|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k} - c^2 \partial_x B_z|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k} - \mu_0 c^2 j_y{}^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k}$$

$$\partial_t E_z|^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} = c^2 \partial_x B_y|^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} - c^2 \partial_y B_x|^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} - \mu_0 c^2 j_z{}^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}}$$

Maxwell-Faraday

$$\partial_t B_x|^{n}_{i,j+\frac{1}{2},k+\frac{1}{2}} = -\partial_y E_z|^{n}_{i,j+\frac{1}{2},k+\frac{1}{2}} + \partial_z E_y|^{n}_{i,j+\frac{1}{2},k+\frac{1}{2}}$$

$$\partial_t B_y|^{n}_{i+\frac{1}{2},j,k+\frac{1}{2}} = -\partial_z E_x|^{n}_{i+\frac{1}{2},j,k+\frac{1}{2}} + \partial_x E_z|^{n}_{i+\frac{1}{2},j,k+\frac{1}{2}}$$

$$\partial_t B_z|^{n}_{i+\frac{1}{2},j+\frac{1}{2},k} = -\partial_x E_y|^{n}_{i+\frac{1}{2},j+\frac{1}{2},k} + \partial_y E_x|^{n}_{i+\frac{1}{2},j+\frac{1}{2},k}$$

$$\partial_t F|^{n'}_{i',j',k'} \equiv \frac{F^{n'+\frac{1}{2}}_{i',j',k'} - F^{n'-\frac{1}{2}}_{i',j',k'}}{\Delta t} \qquad \partial_x F|^{n'}_{i',j',k'} \equiv \frac{F^{n'}_{i'+\frac{1}{2},j',k'} - F^{n'}_{i'-\frac{1}{2},j',k'}}{\Delta x}$$

$$\partial_y F|^{n'}_{i',j',k'} \equiv \frac{F^{n'}_{i',j'+\frac{1}{2},k'} - F^{n'}_{i',j'-\frac{1}{2},k'}}{\Delta y} \qquad \partial_z F|^{n'}_{i',j',k'} \equiv \frac{F^{n'}_{i',j',k'+\frac{1}{2}} - F^{n'}_{i',j',k'-\frac{1}{2}}}{\Delta z}$$

What happens to div B = 0 and div E=4πρ equations?

- The (discretized) div **B** = 0 and div **E**=4πρ equations must be satisfied for t=0 (consistent initial condition)

- If div **B** = 0 is satisfied for t=0, then it remains satisfied at all times as long as B is evolved with the Faraday equation. This remains true when equations are discretized in space and time (provided that **div curl = 0**)

- If div **E**=4πρ is satisfied for t=0 then it remains satisfied at all times if continuity equation (div **J** + ∂ρ/∂t=0) holds

- Unfortunately, using direct charge and current deposition (i.e., J and ρ from numerical particles via shape-functions), the discretized version of the continuity equation is **not satisfied** (div **E**≠4πρ):

  - At each step correct **E,** namely **E'**=**E**-grad[δϕ], so that div **E'**=4πρ → Δ(δϕ) = div **E** - 4πρ [Boris correction];

  - Construct **J** in such a way cont. equation is automatically satisfied [Esirkepov, 2001];

References

LPA physics:
- E. Esarey, C. B. Schroeder, and W. P. Leemans, Rev. Mod. Phys. 81, 1229 (2009)

PIC method:
- C. K. Birdsall and A. B. Langdon, Plasma Physics Via Computer Simulation (Adam-Hilger, 1991)
- J.-L. Vay and R. Lehe, Rev. Accl. Sci. Tech. 09, 165 (2016)
- Haugboelle *et al.*, Physics of Plasmas 20, 062904 (2013)
- T. Esirkepov, Computer Physics Communications, 135(2), 144 (2001)