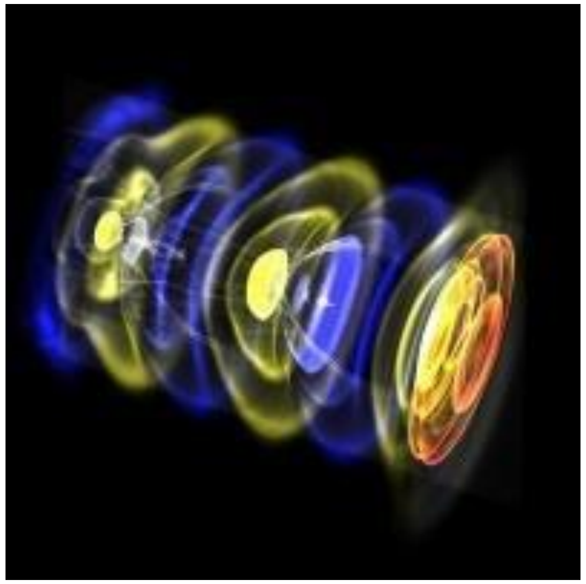
Review of spectral Maxwell solvers for electromagnetic Particle-In-Cell: algorithms and advantages

Remi Lehe
Lawrence Berkeley National Laboratory

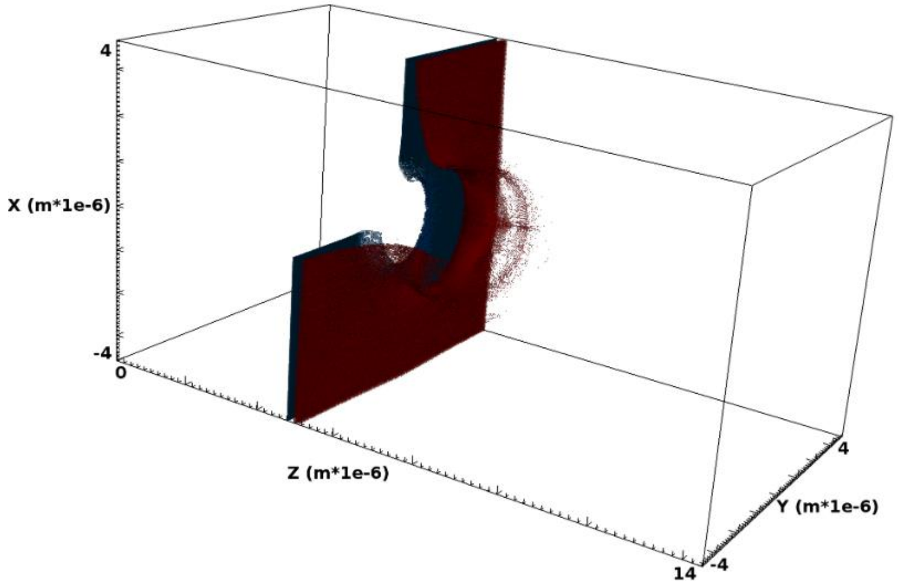
Context: simulations for laser-plasma interactions

Use high-power laser to study laser-plasma interactions

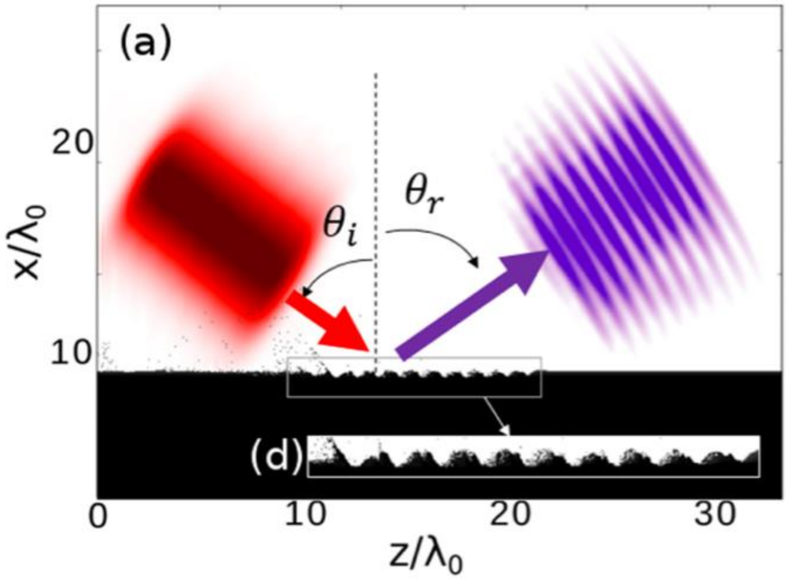
Electron acceleration



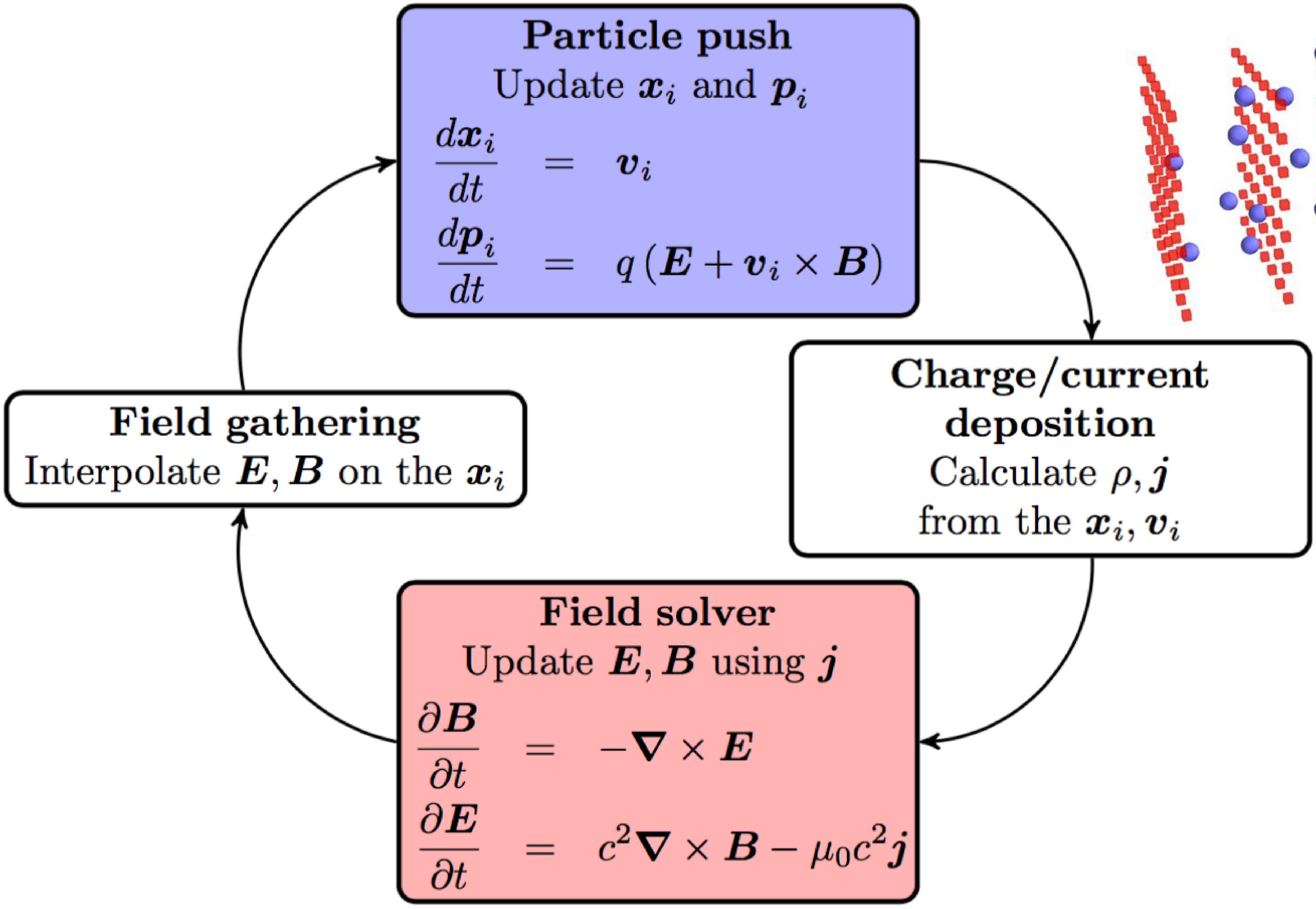
Ion acceleration

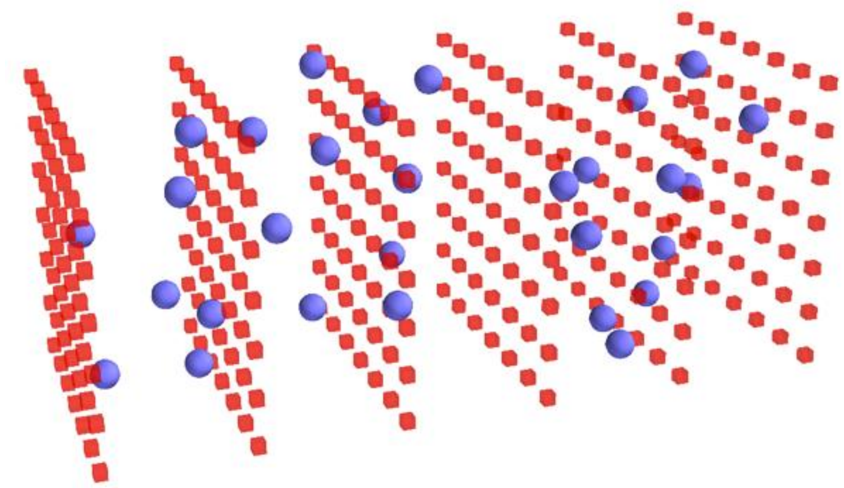


High-harmonic photon generation



Context: Electromagnetic Particle-In-Cell method





Outline

- Finite-difference vs. spectral EM-PIC
- Recent developments for spectral solvers
  - Parallelization strategy
  - Pseudo-spectral cylindrical algorithm
- Some added benefits of spectral solvers
  - Cancelation of E + v x B
  - Boosted-frame simulations and Galilean spectral scheme

Finite-difference Maxwell solver (FDTD)

Discretized Maxwell equations

$$\frac{1}{c^2}\hat{\partial}_t \boldsymbol{E} = \hat{\boldsymbol{\nabla}} \times \boldsymbol{B} - \mu_0 \boldsymbol{J}$$

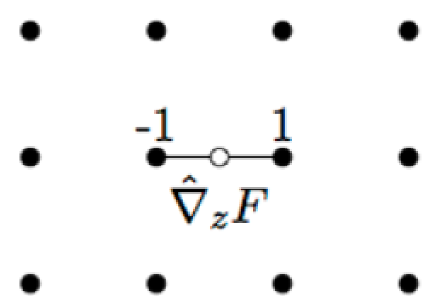
$$\hat{\partial}_t \boldsymbol{B} = -\hat{\boldsymbol{\nabla}} \times \boldsymbol{E}$$

Discretized in time
(over time steps)

$$\hat{\partial}_t \boldsymbol{B} = \frac{\boldsymbol{B}^{n+1} - \boldsymbol{B}^{n}}{\Delta t}$$

Discretized in space
(over a grid)

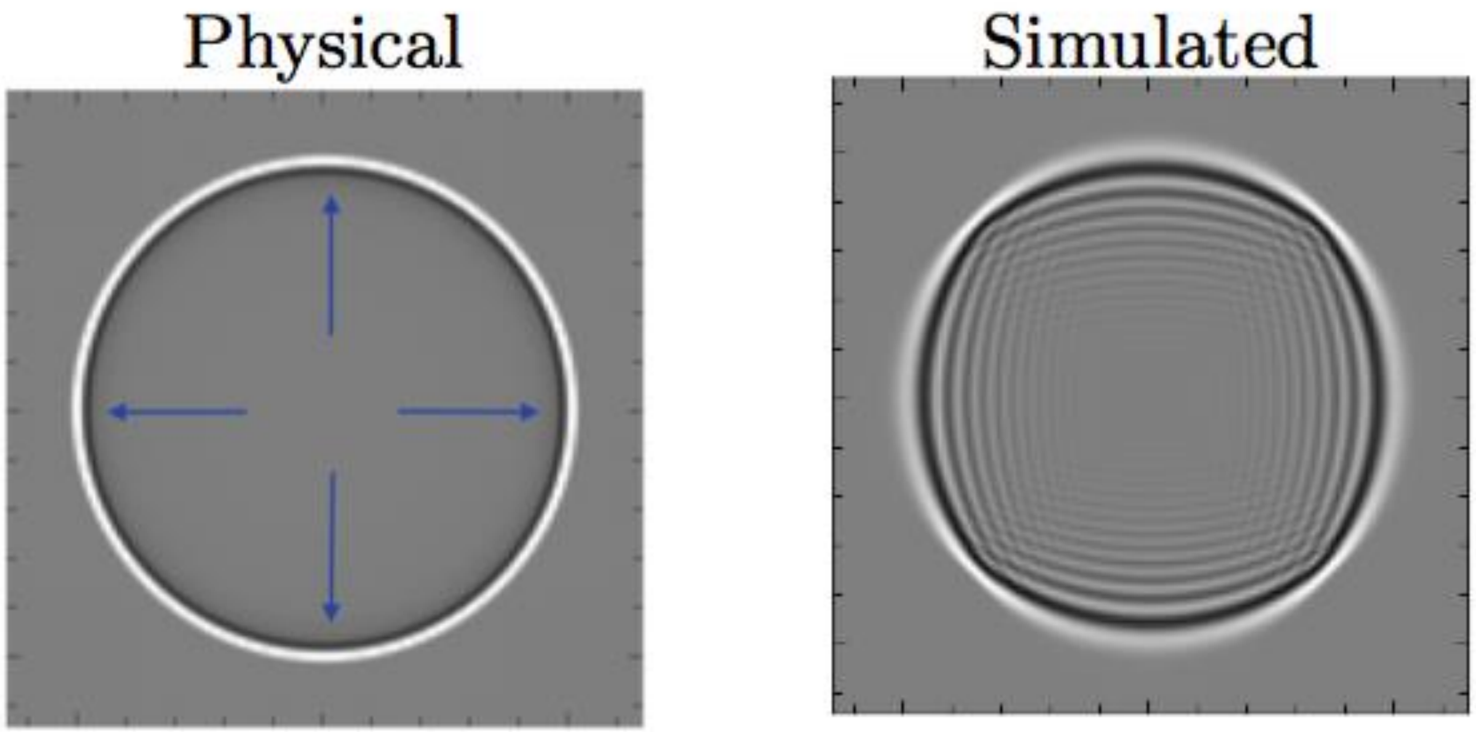
$$\hat{\partial}_x \boldsymbol{E} = \frac{\boldsymbol{E}_{i+1,j,k} - \boldsymbol{E}_{i,j,k}}{\Delta x}$$



Issue with finite-difference: Numerical dispersion

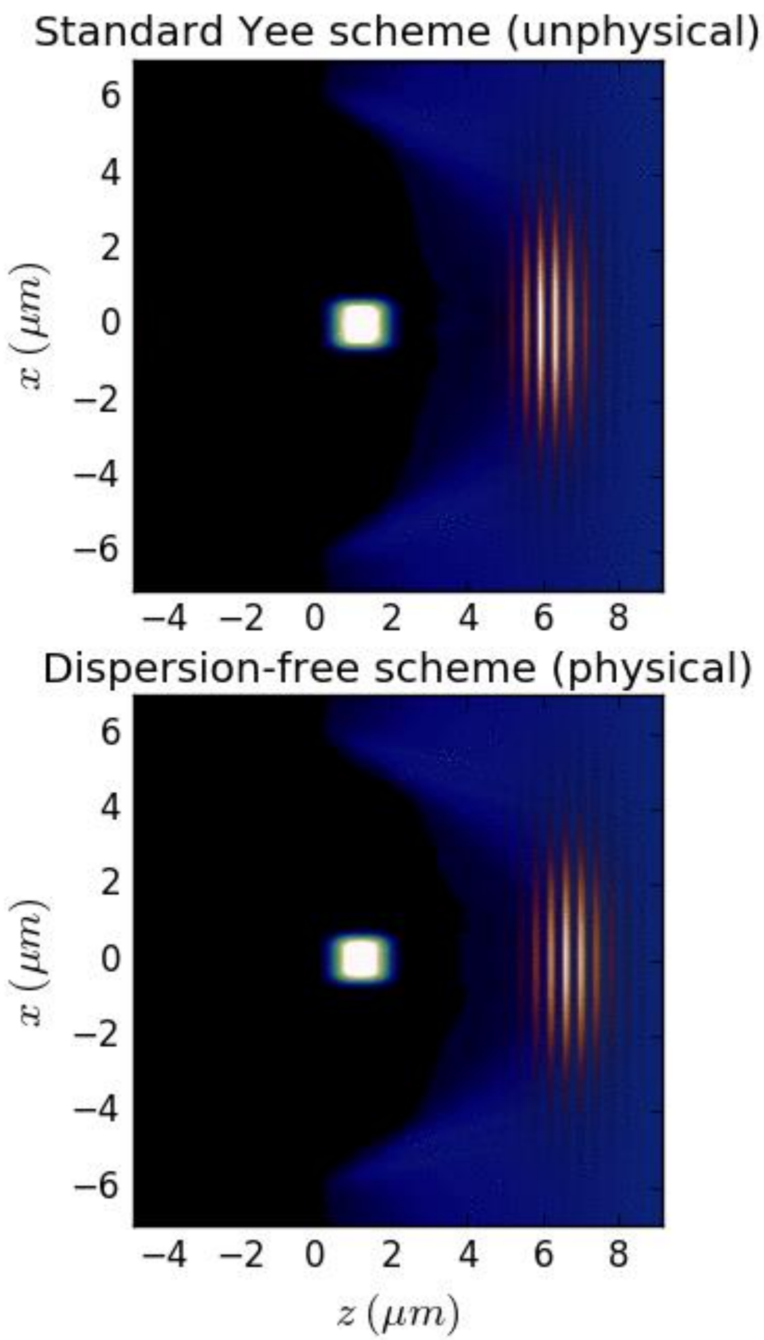
Spurious numerical dispersion

- In finite-difference algorithms, the phase velocity of electromagnetic waves is different than c, and depends on the wavelength and the propagation angle.

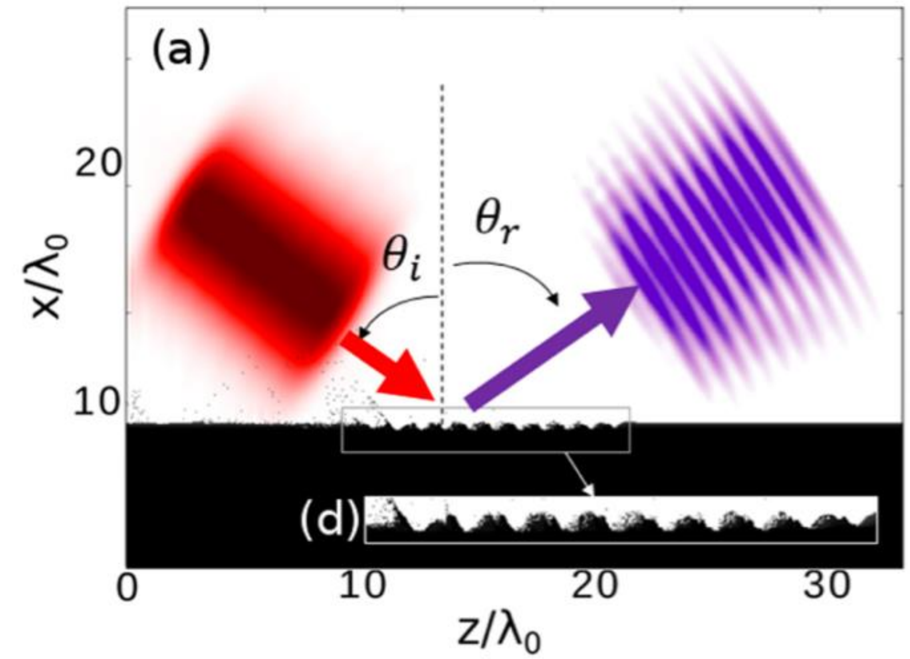


Numerical dispersion can have strong impact in simulations

Early dephasing

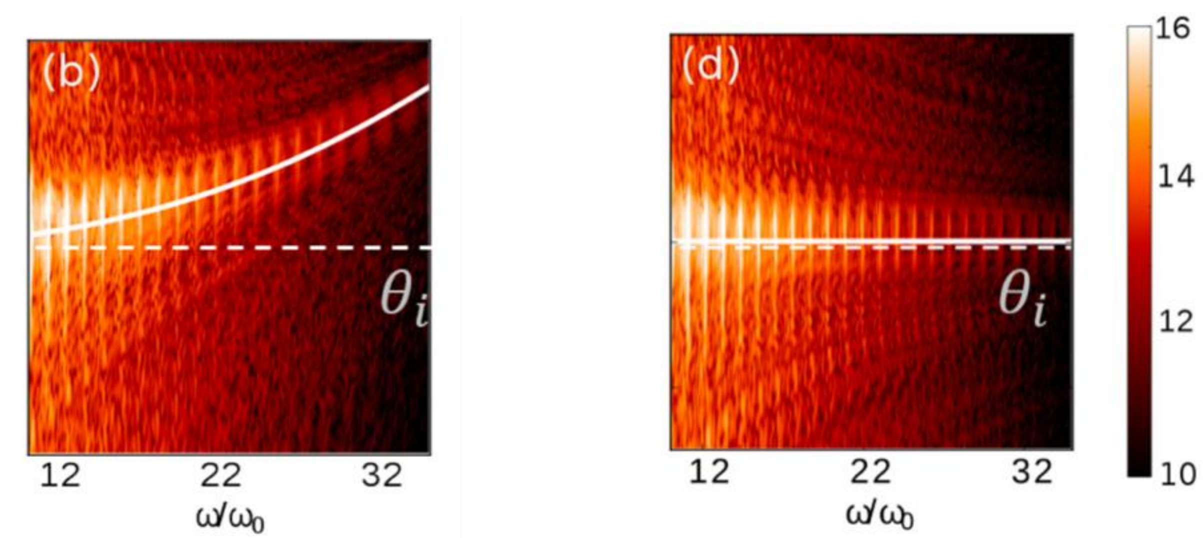


Deflection of laser-produced harmonics



Finite-difference

Dispersion-free scheme

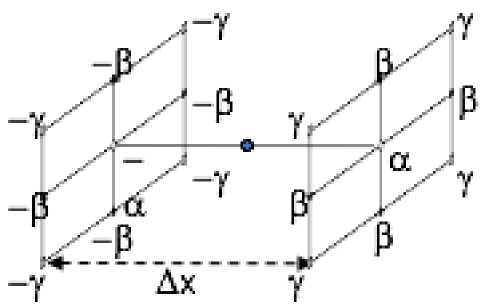


G. Blaclard et al., PRE, 2017

Some existing solutions against numerical dispersion

Extended stencil

- Can reduce numerical dispersion along a given axis.



- But has anisotropy

J. B. Cole, IEEE Trans. Microw. Theory Tech. 45 (1997)
A. Pukhov, J. Plasma Physics 61 (1999) 425
M. Karkkainen et al., Proc. ICAP, Chamonix, France (2006)
B. Cowan et al, PRST-AB 16 (2013) 041303
A. Blinne et al, ArXiv:**1710.06829** (2017)
R. Lehe et al, PRST-AB 16 (2013) 021301

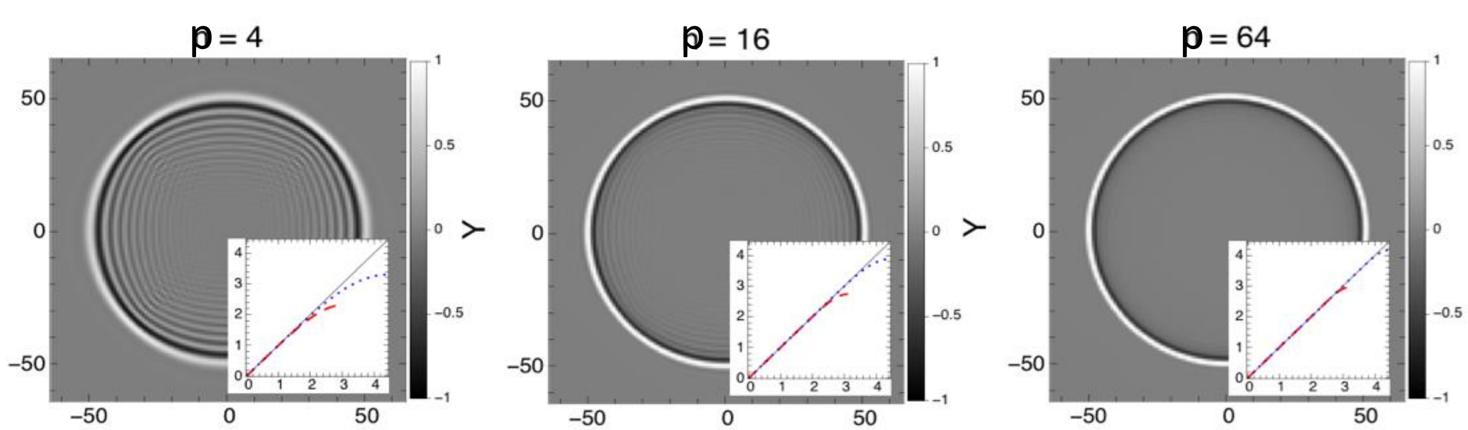
Directional splitting

- Uses collocated grid ; dispersion-free along axes
- But again, has anisotropy

e.g. R. Nuter et al, Eur. Phys. J. D 68, 177 (2014)

Spectral (PSATD)

- Dispersion and anisotropy are reduced when increasing the order p of the solver



J.L. Vay, CPC (2016) ;
H. Vincenti and J-L Vay, *CPC* 200, 147 (2016)

Finite-Difference Time Domain (FDTD)

Discretized Maxwell equations

$$\frac{1}{c^2}\hat{\partial}_t \boldsymbol{E} = \hat{\boldsymbol{\nabla}} \times \boldsymbol{B} - \mu_0 \boldsymbol{J}$$

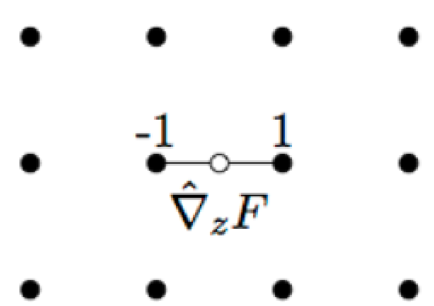
$$\hat{\partial}_t \boldsymbol{B} = -\hat{\boldsymbol{\nabla}} \times \boldsymbol{E}$$

Discretized in time
(over time steps)

$$\hat{\partial}_t \boldsymbol{B} = \frac{\boldsymbol{B}^{n+1} - \boldsymbol{B}^n}{\Delta t}$$

Discretized in space
(over a grid)

$$\hat{\partial}_x \boldsymbol{E} = \frac{\boldsymbol{E}_{i+1,j,k} - \boldsymbol{E}_{i,j,k}}{\Delta x}$$



Pseudo-Spectral Analytical Time Domain (PSATD)

Discretized Maxwell equations

$$\frac{1}{c^2}\hat{\partial}_t \boldsymbol{E} = \hat{\boldsymbol{\nabla}} \times \boldsymbol{B} - \mu_0 \boldsymbol{J}$$

$$\hat{\partial}_t \boldsymbol{B} = -\hat{\boldsymbol{\nabla}} \times \boldsymbol{E}$$

Discretized in time
(over time steps)

$$\hat{\partial}_t \boldsymbol{B} = \frac{\boldsymbol{B}^{n+1} - \boldsymbol{B}^n}{\Delta t}$$

Integrate the equations analytically in spectral space

Discretized in space
(over a grid)

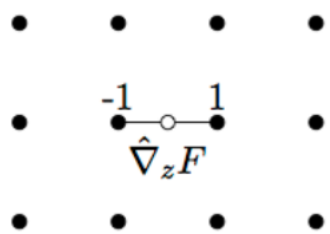
$$\hat{\partial}_x \boldsymbol{E} = \frac{\boldsymbol{E}_{i+1,j,k} - \boldsymbol{E}_{i,j,k}}{\Delta x}$$

Use very high-order approximation for spatial derivatives

Using very high-order approximation to the derivatives

Order 2 approximation

$$\hat{\partial}_x \boldsymbol{E} = \frac{\boldsymbol{E}_{i+1,j,k} - \boldsymbol{E}_{i,j,k}}{\Delta x}$$



Order p approximation

$$\hat{\partial}_x \boldsymbol{E} = \sum_{\ell=0}^{p/2-1} c_\ell \frac{\boldsymbol{E}_{i+1+\ell,j,k} - \boldsymbol{E}_{i-\ell,j,k}}{\Delta x}$$

(requires more guard cells,
in an MPI domain decomposition)

Fornberg, SIAM J. Num. Analysis, 1990

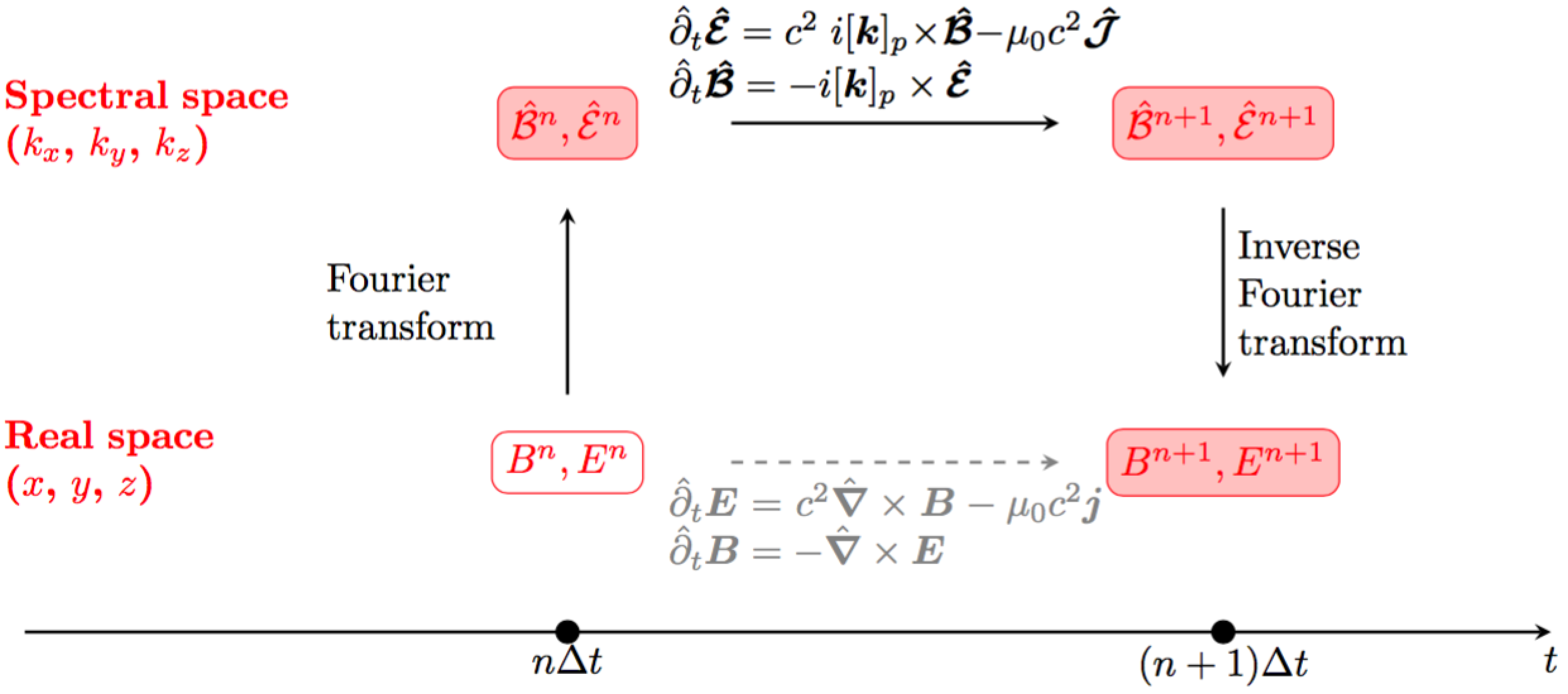
For p very large, this can be efficiently evaluated in Fourier space

$$\mathcal{F}\{\hat{\partial}_x \boldsymbol{E}\} = i[k_x]_p \mathcal{F}\{\boldsymbol{E}\}$$

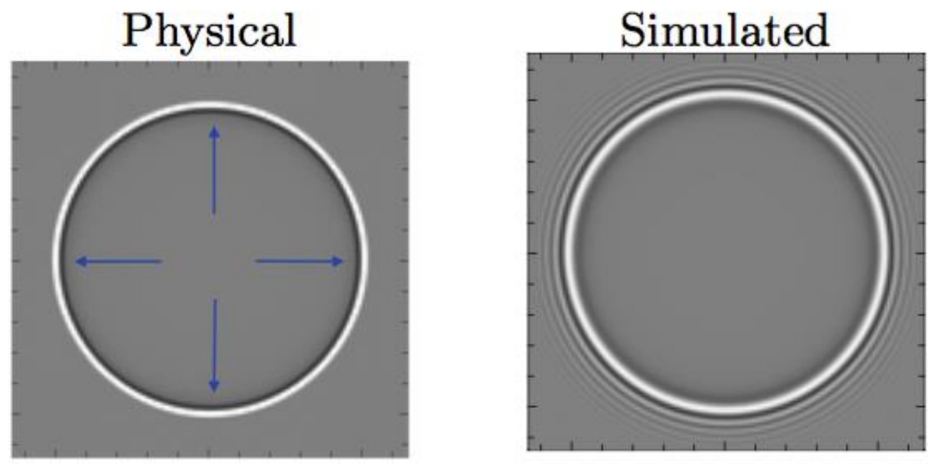
$$[k_x]_p = \sum_{\ell=0}^{p/2-1} c_\ell \frac{e^{ik_x \ell \Delta x/2} - e^{-ik_x \ell \Delta x/2}}{\Delta x}$$

$$p = \infty \rightarrow [k_x]_p = k_x$$

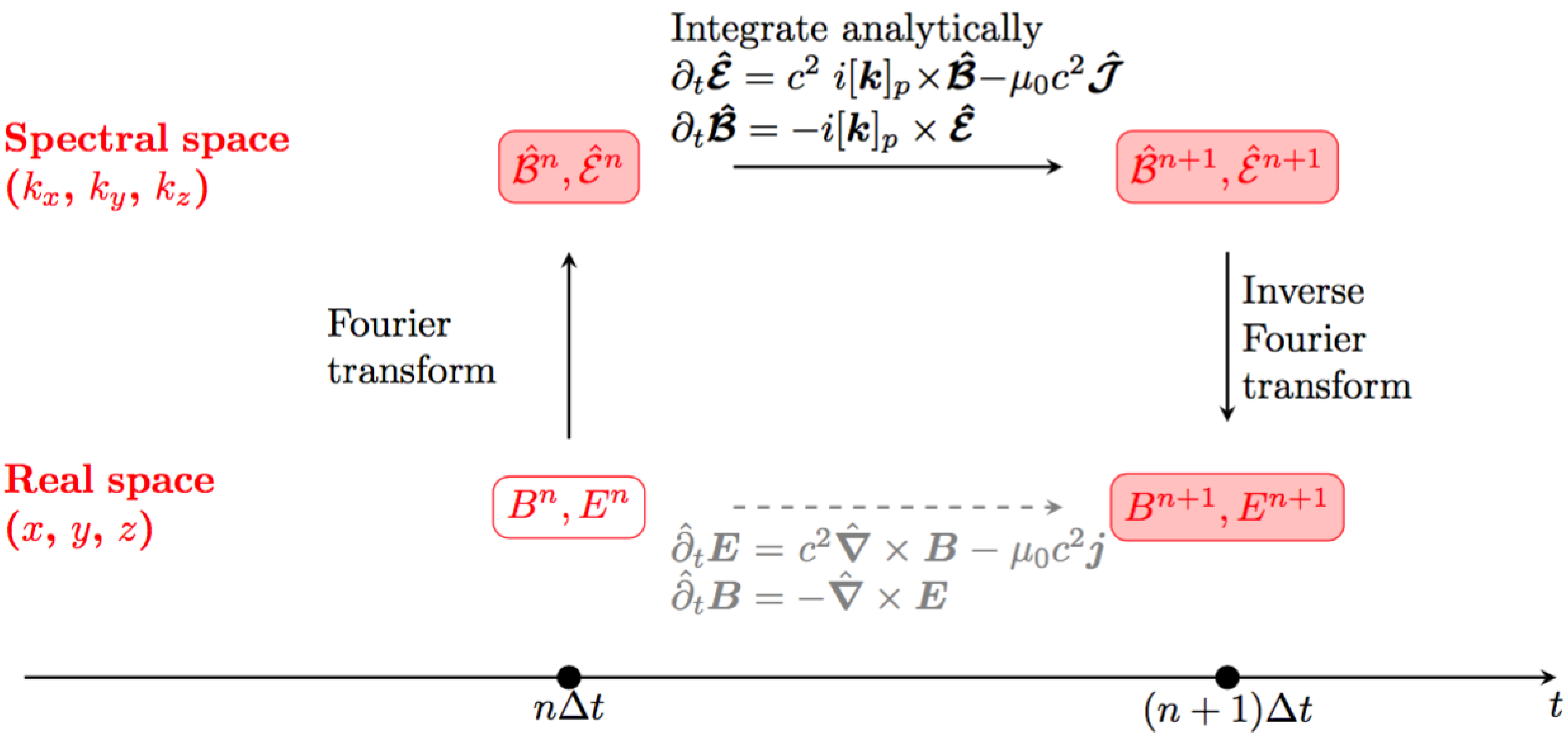
Using very high-order approximation for the derivatives



When using discretized derivatives in time:



Integrating the equations analytically

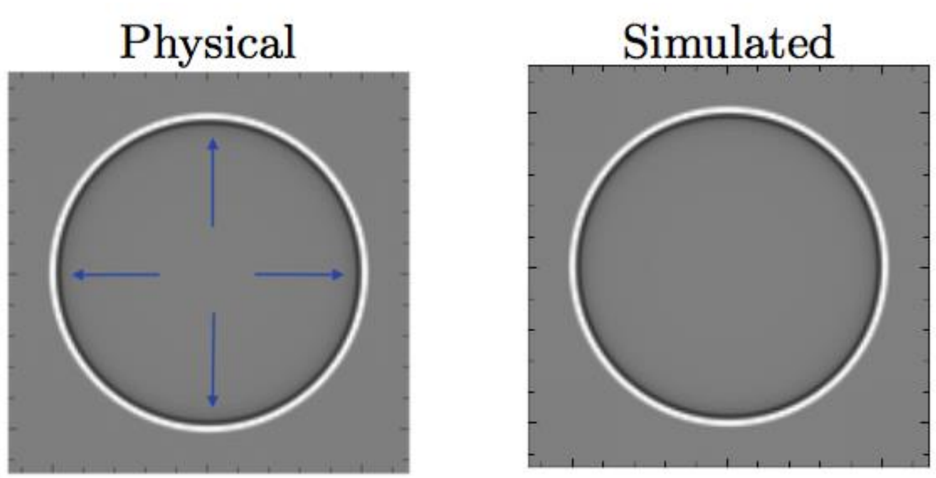


Haber et al., Proc. 6th Conf. on Num. Sim. Plasmas, 1973

$$\hat{\boldsymbol{\mathcal{E}}}^{n+1} = C\hat{\boldsymbol{\mathcal{E}}}^n + iS\frac{[\boldsymbol{k}]_p}{[k]_p}\times\hat{\boldsymbol{\mathcal{B}}}^n - \frac{S}{[k]_p}\hat{\boldsymbol{\mathcal{J}}}^{n+1/2} + (1-C)\frac{[\boldsymbol{k}]_p([\boldsymbol{k}]_p\cdot\hat{\boldsymbol{\mathcal{E}}}^n)}{[k]_p^2} + \frac{[\boldsymbol{k}]_p([\boldsymbol{k}]_p\cdot\hat{\boldsymbol{\mathcal{J}}}^{n+1/2})}{[k]_p^2}\left(\frac{S}{[k]_p} - \Delta t\right)$$

$$\hat{\boldsymbol{\mathcal{B}}}^{n+1} = C\hat{\boldsymbol{\mathcal{B}}}^n - iS\frac{[\boldsymbol{k}]_p\times\hat{\boldsymbol{\mathcal{E}}}^n}{[k_p]} + i\frac{1-C}{[k]_p^2}[\boldsymbol{k}]_p\times\hat{\boldsymbol{\mathcal{J}}}^{n+1/2} \qquad C=\cos([k]_p c\Delta t) \qquad S=\sin([k]_p c\Delta t)$$

When integrating analytically in time:



Outline

- Finite-difference vs. spectral EM-PIC

- Recent developments for spectral solvers
  - Parallelization strategy
  - Pseudo-spectral cylindrical algorithm

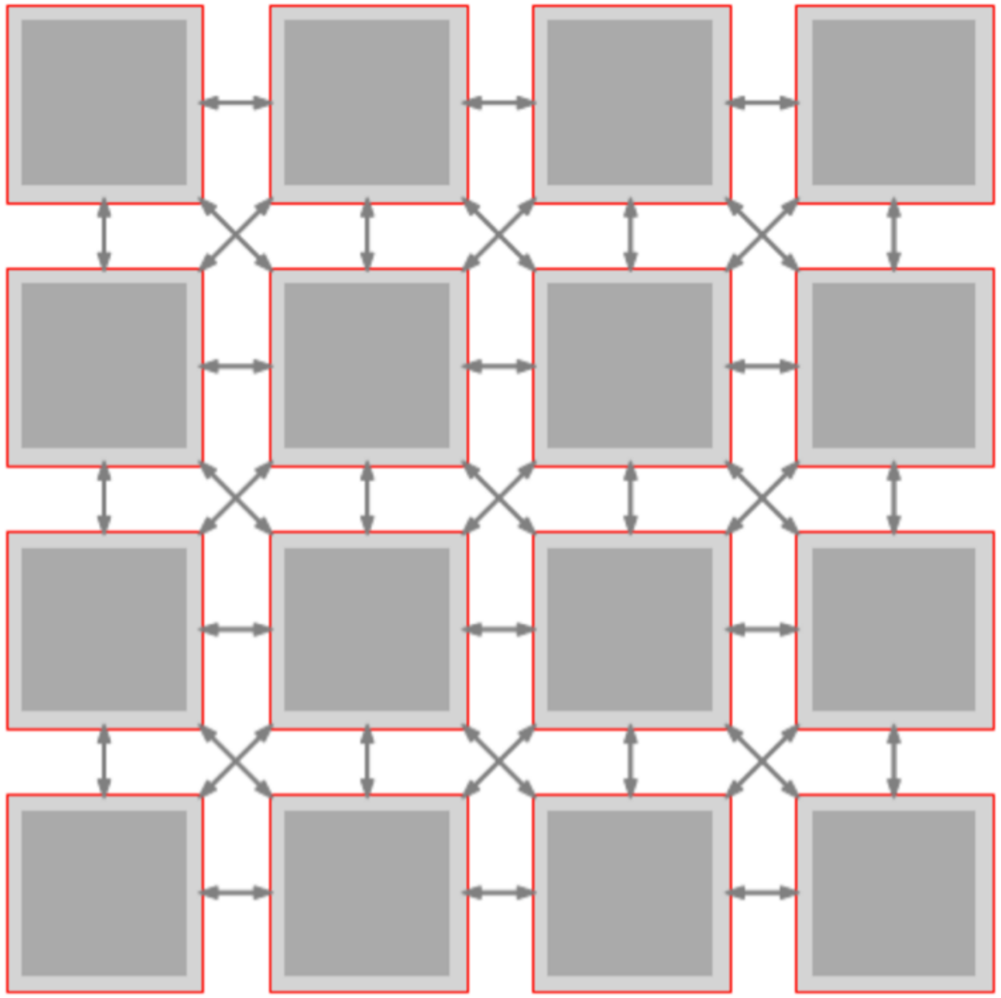
- Some added benefits of spectral solvers
  - Cancelation of E + v x B
  - Boosted-frame simulations and Galilean spectral scheme

Parallelization with distributed memory (MPI)

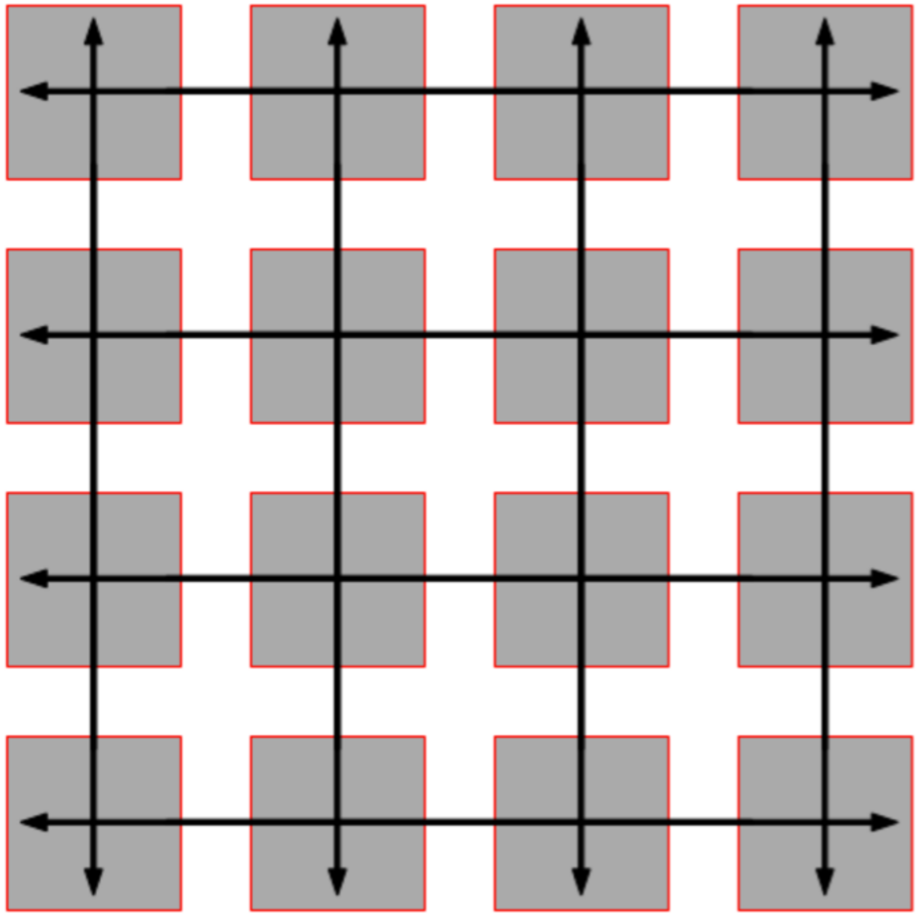
$$\frac{1}{c^2}\partial_t \boldsymbol{E} = \boldsymbol{\nabla} \times \boldsymbol{B} - \mu_0 \boldsymbol{J}$$

$$\hat{\partial}_t \boldsymbol{B} = -\boldsymbol{\nabla} \times \boldsymbol{E}$$

Finite-difference:
Exchange guard cell after field update



Spectral ("naive" implementation)
Global FFT before and after the field update in Fourier space



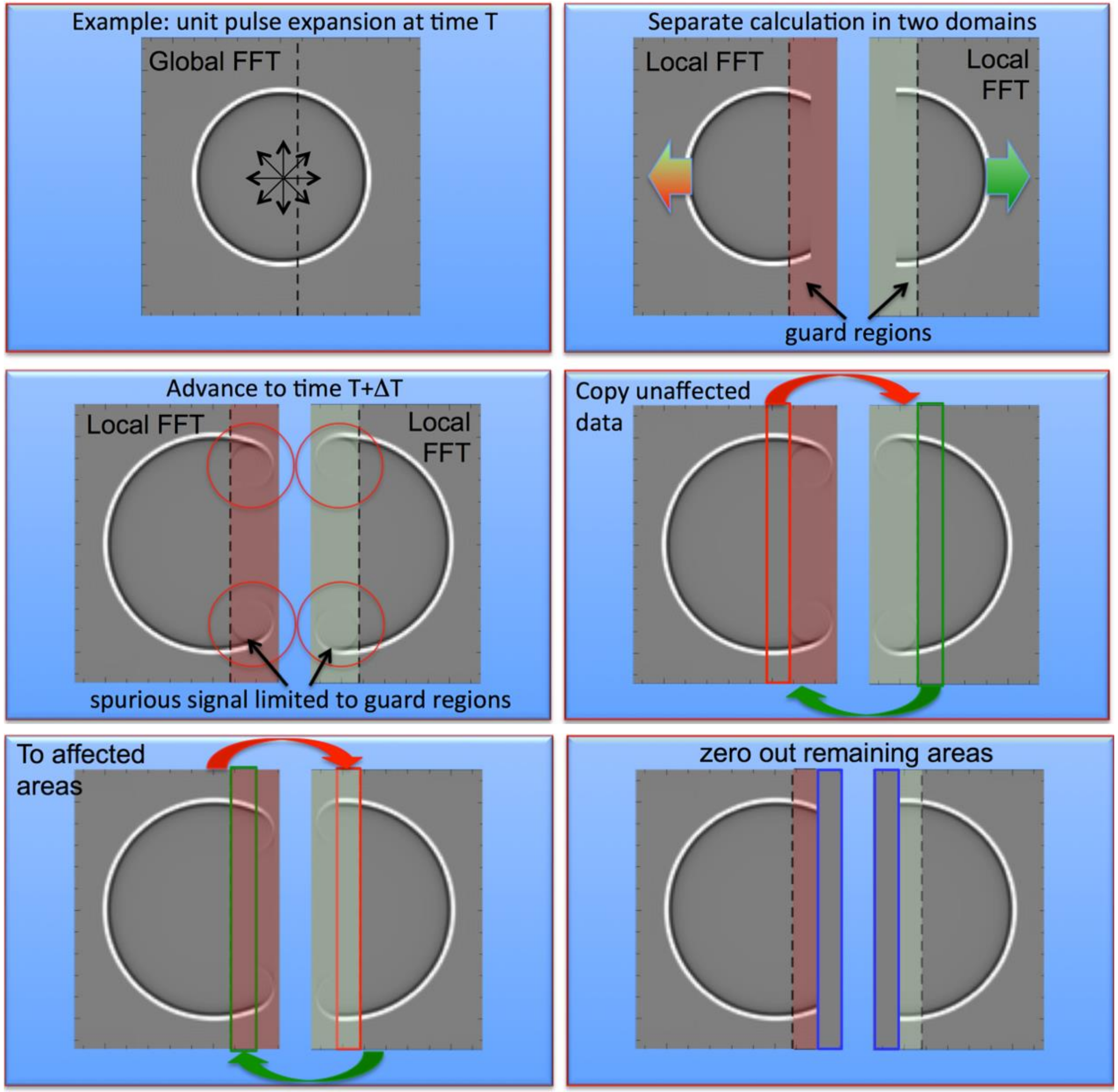
Parallelization with local FFT

Parallelization

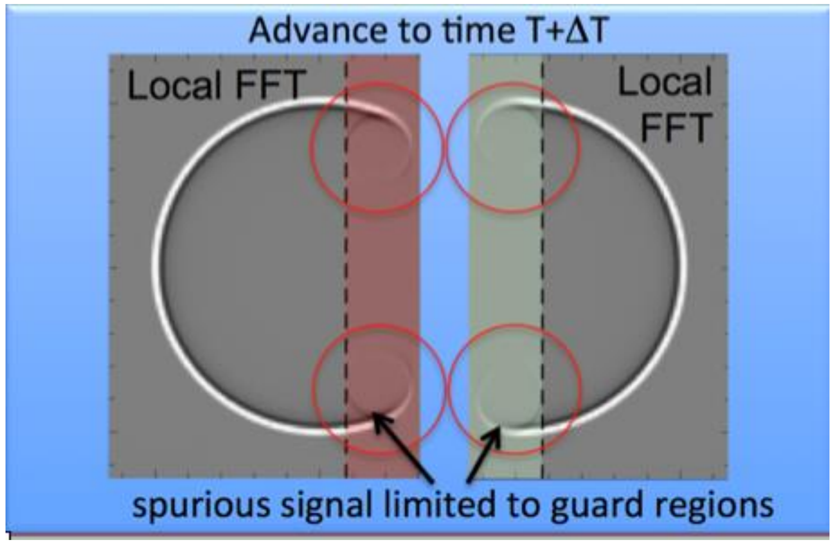
- In theory: PSATD require global FFT, which does not scale well to multiple nodes
- But for the Maxwell equations, local FFT can be used provided that there are enough guard cell
- The key idea is that the error only propagates over a finite distance over a given timestep

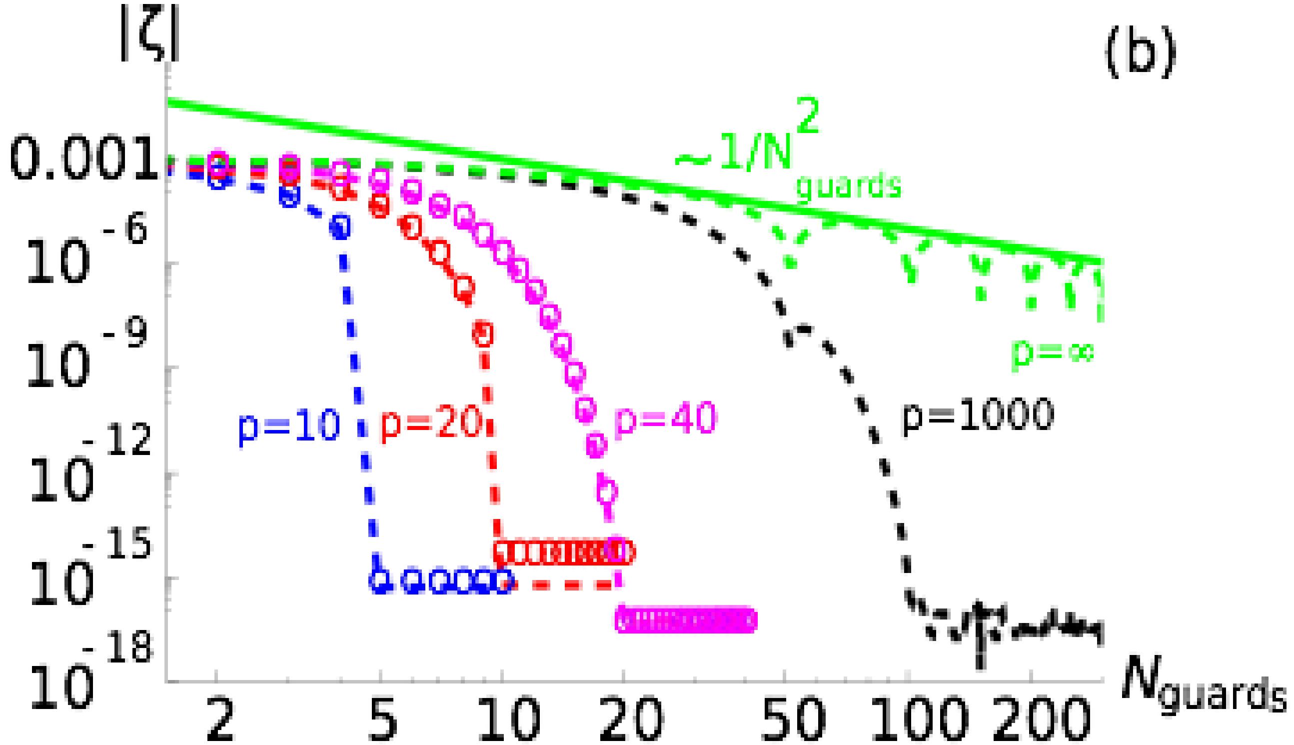
Proposed by Jean-Luc Vay

J.L. Vay, Comp. Phys. Comm. (2016)



Number of guard cell increases with the order p of the solver

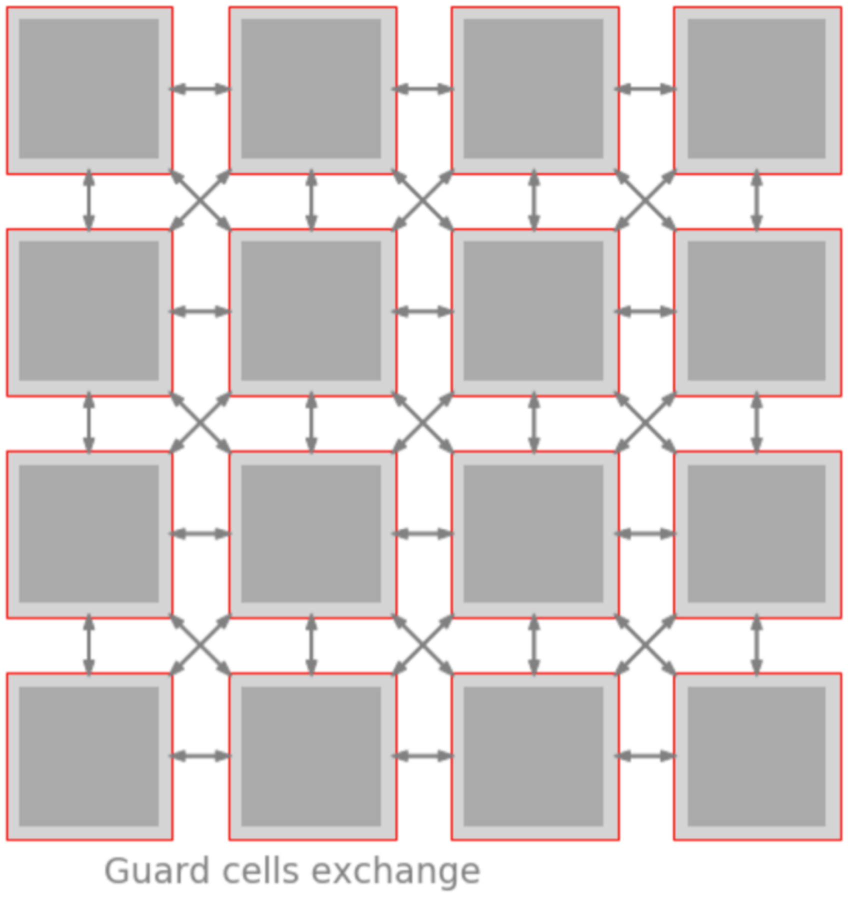




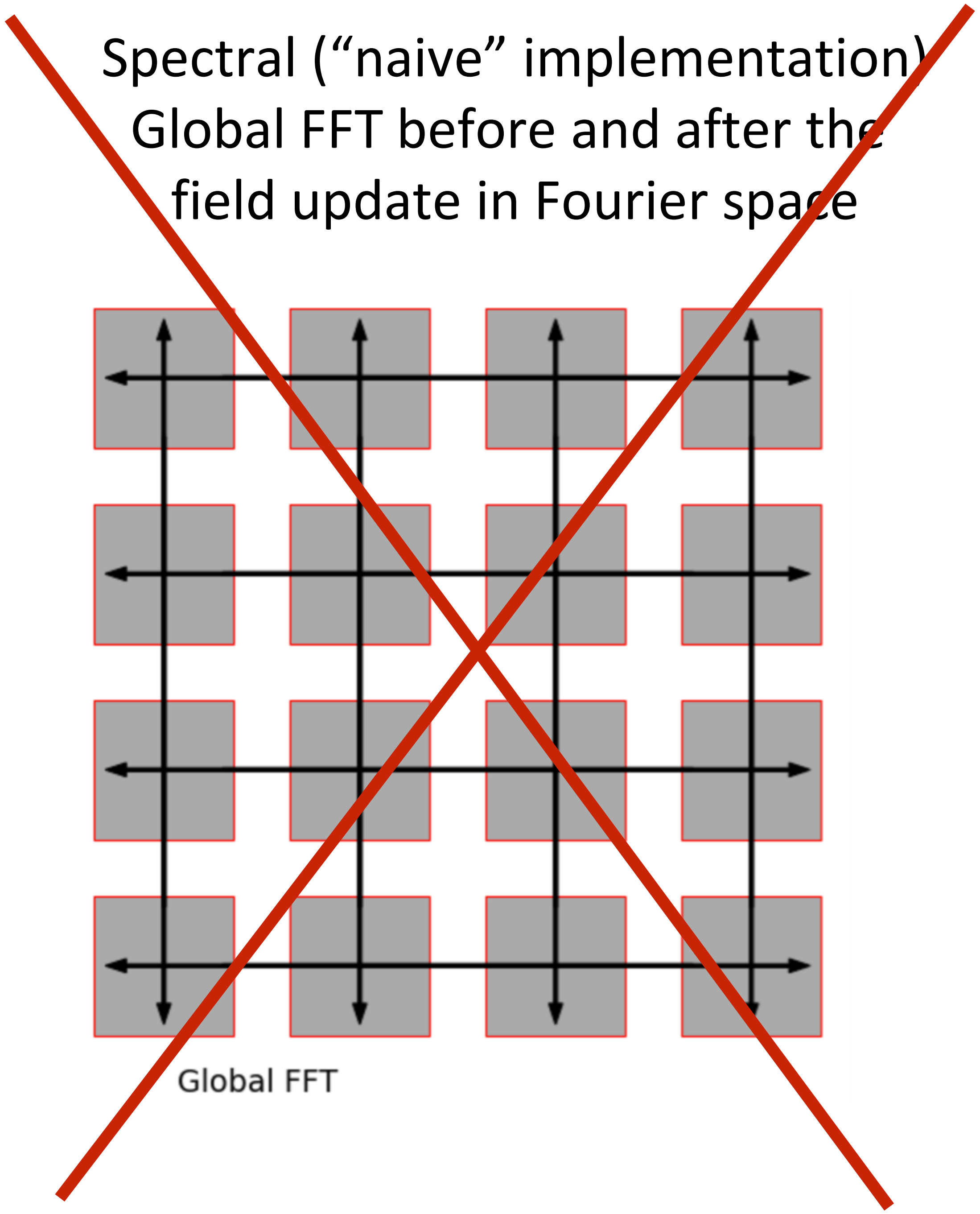
H. Vincenti and J-L Vay, *CPC* 200, 147 (2016)

Parallelization with distributed memory (MPI)

Finite-difference:
Exchange guard cell after field update

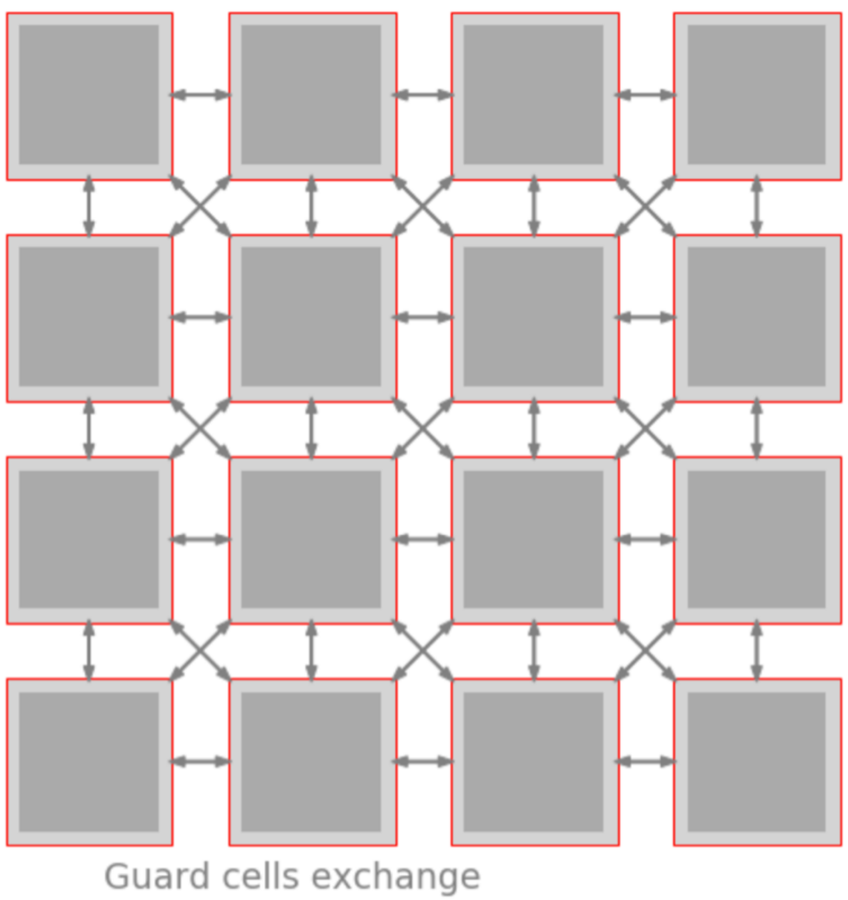


Spectral ("naive" implementation)
Global FFT before and after the field update in Fourier space

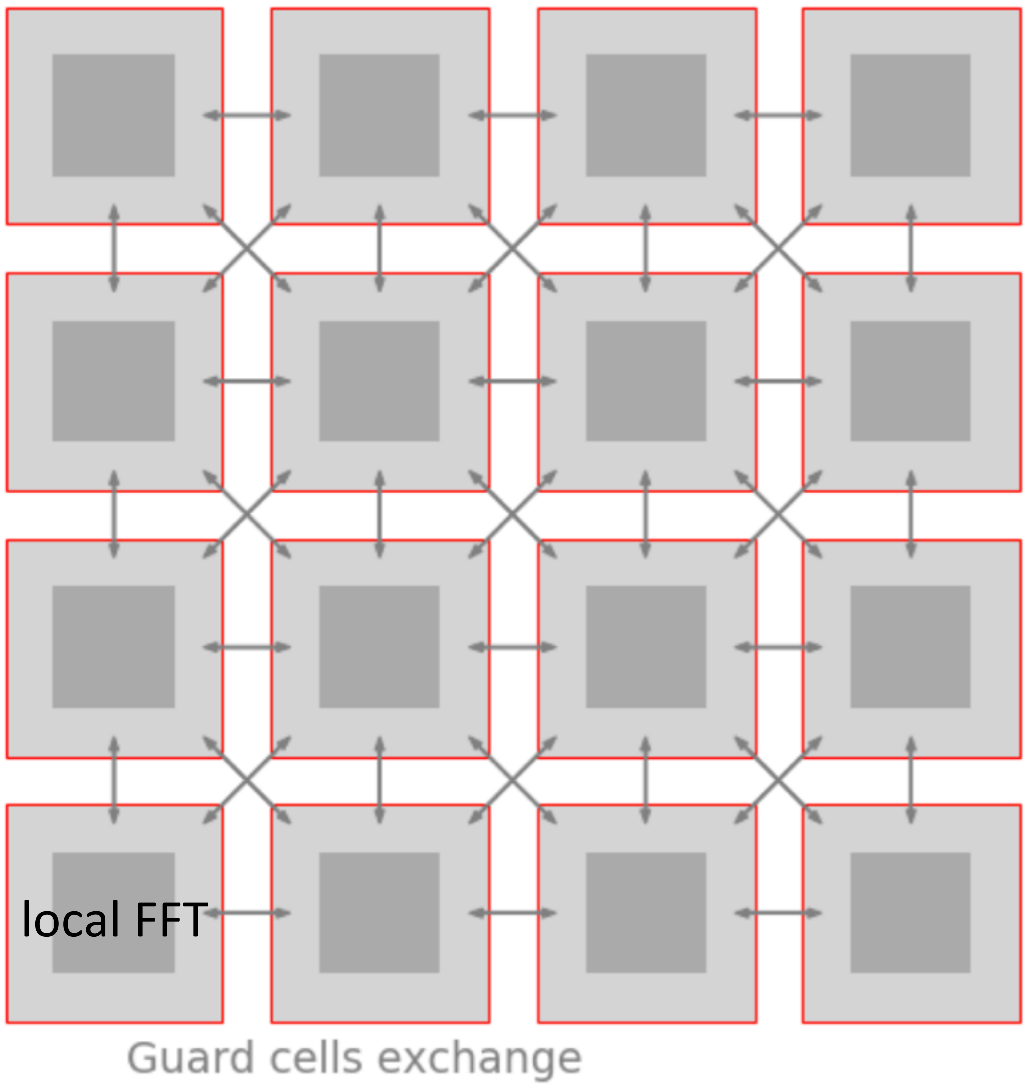


Parallelization with distributed memory (MPI)

Finite-difference:
Exchange guard cell after field update



Spectral:
Exchange (many) guard cell after field update

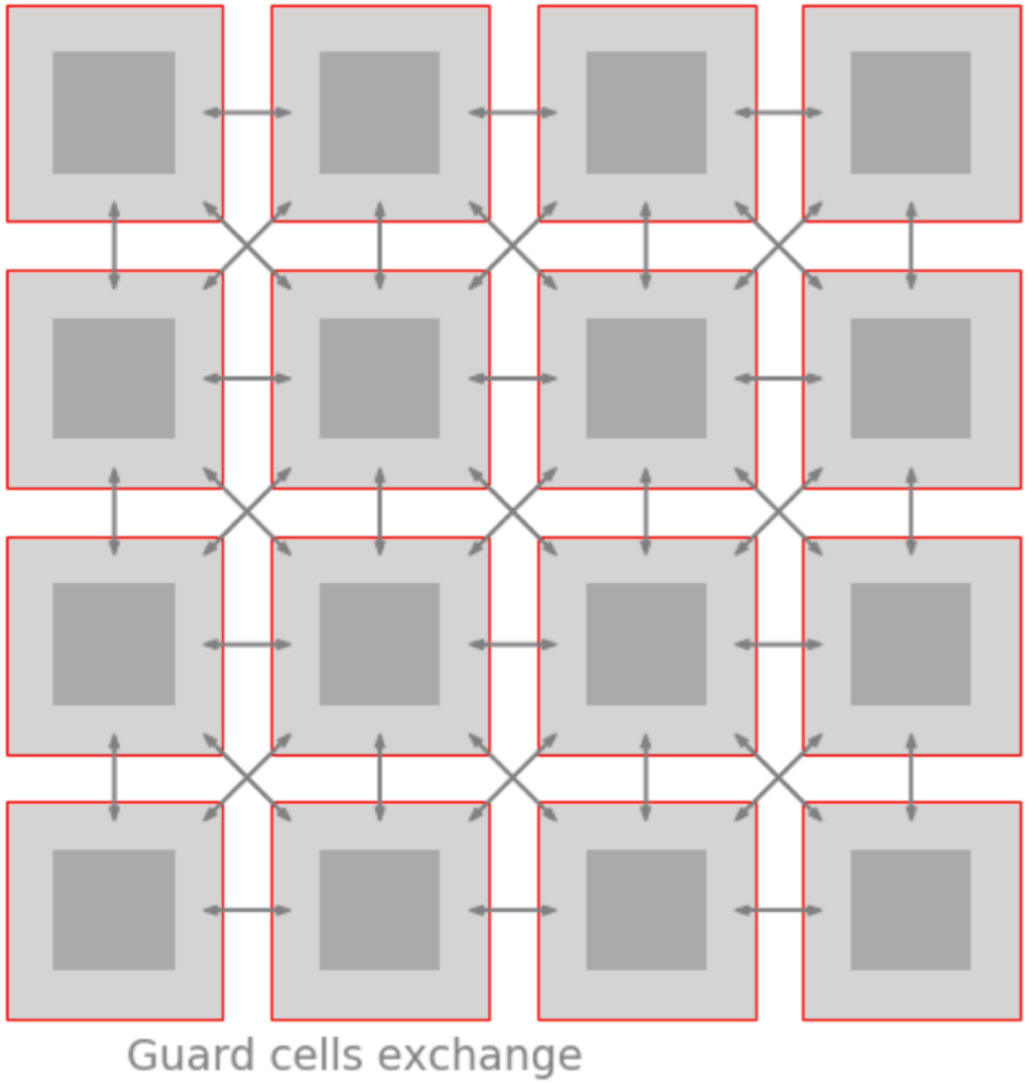


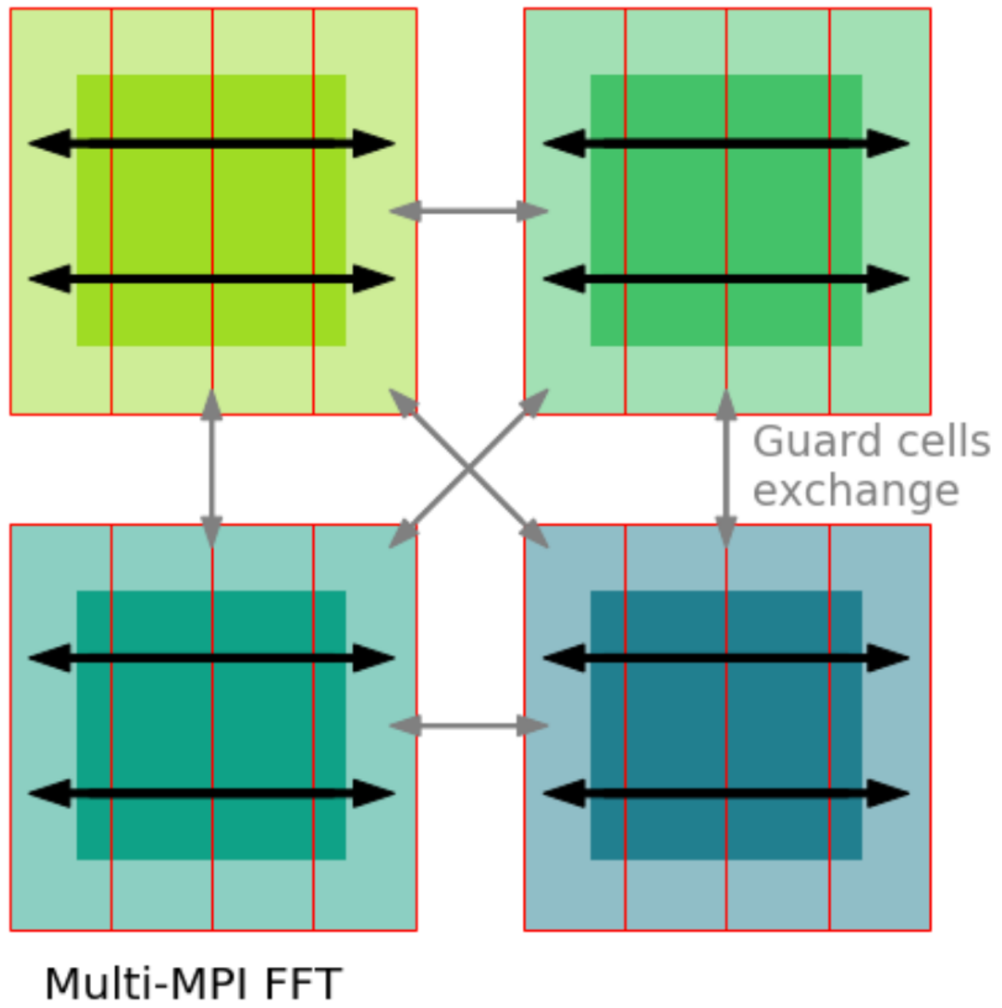
Allows good scaling to many nodes: H. Vincenti, J.-L. Vay, ArXiv: 1707.08500. (2018)

Hybrid FFT approach allows even better scaling

Spectral:
Exchange (many) guard cell after field update

Spectral:
Use global FFT within MPI groups,
and guard cell exchange between MPI groups



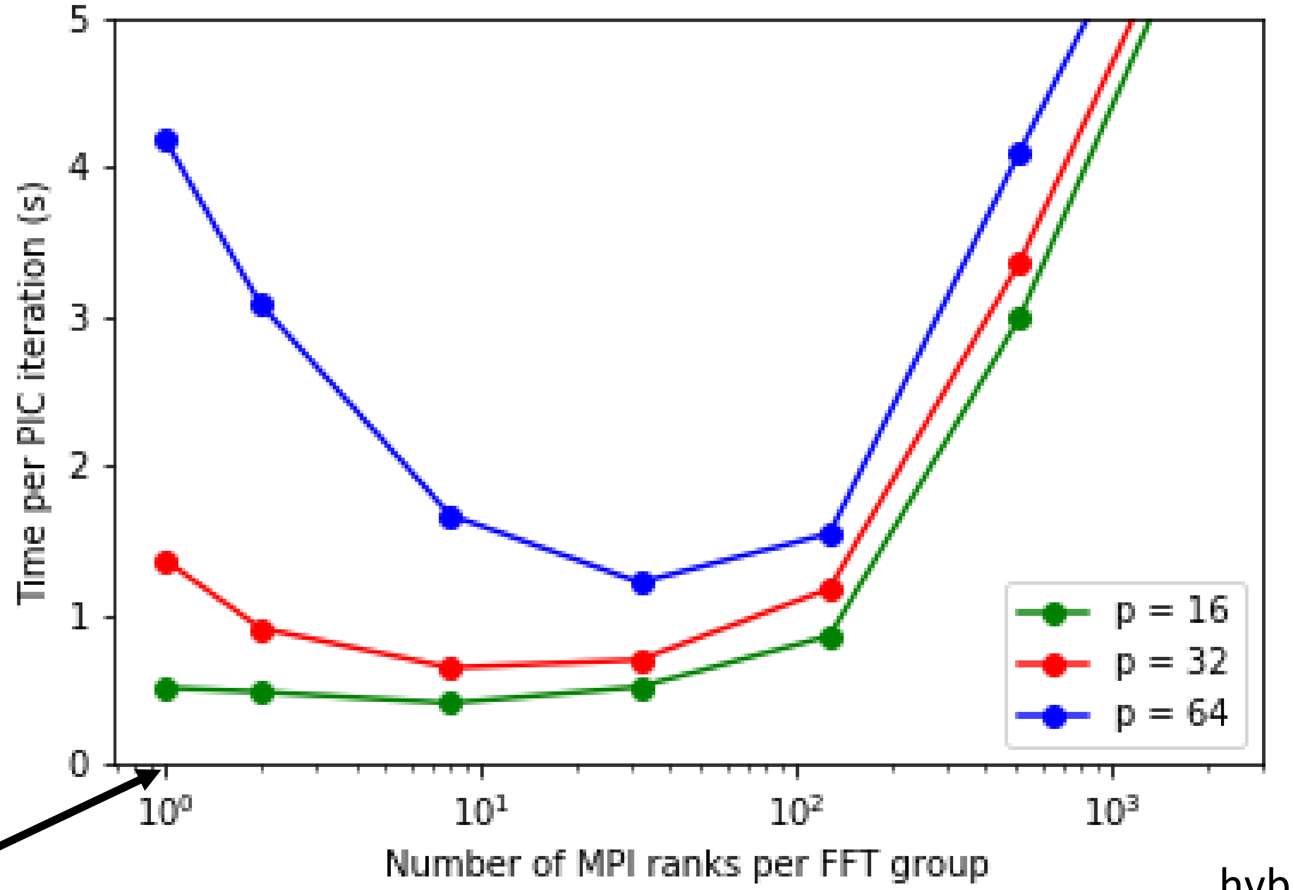


Implemented in the codes PICSAR and WarpX

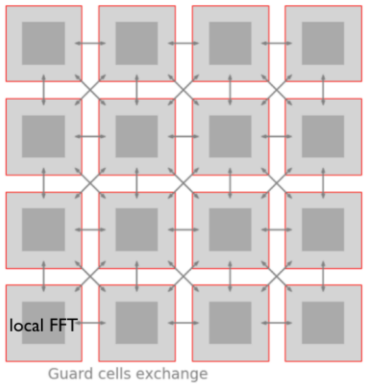
H. Kallala & H. Vincenti, in preparation (2018)

Optimal number of MPI per FFT group depends on the order p

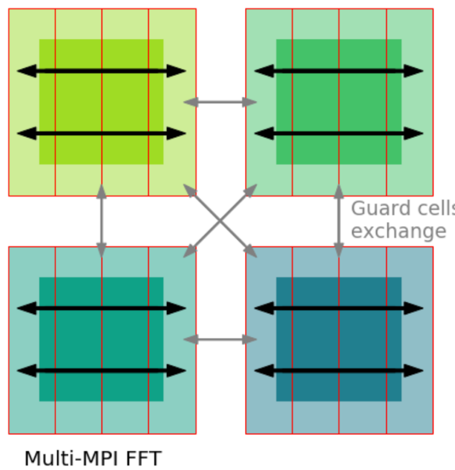
Scalings on with WarpX on 512 KNL nodes @NERSC



purely local FFTs



hybrid FFT



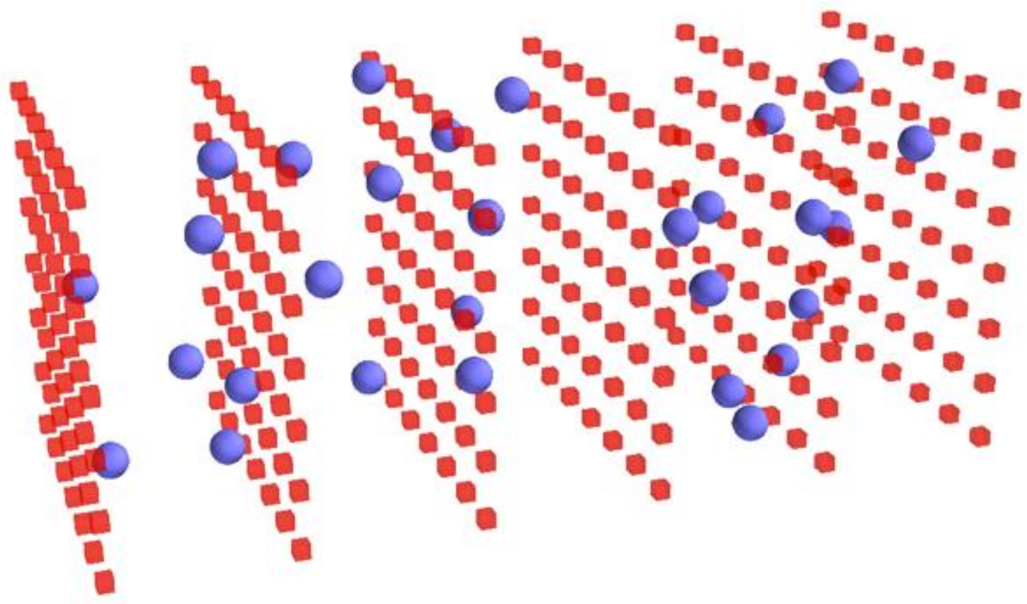
Similar results in the code PICSAR: H. Kallala & H. Vincenti, in preparation (2018)

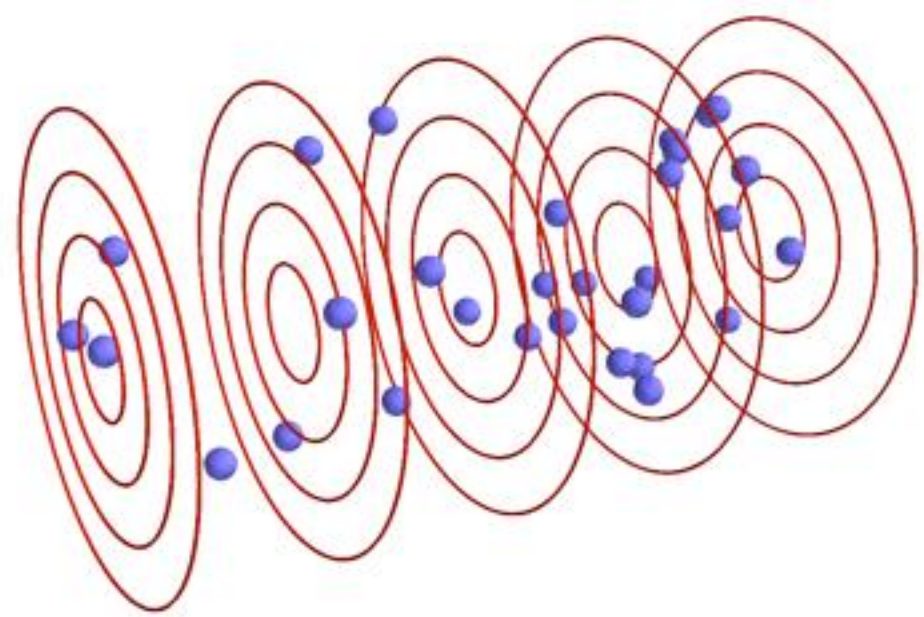
Outline

- Finite-difference vs. spectral EM-PIC
- Recent developments for spectral solvers
  - Parallelization strategy
  - Pseudo-spectral cylindrical algorithm
- Some added benefits of spectral solvers
  - Cancelation of E + v x B
  - Boosted-frame simulations and Galilean spectral scheme

Spectral cylindrical algorithm

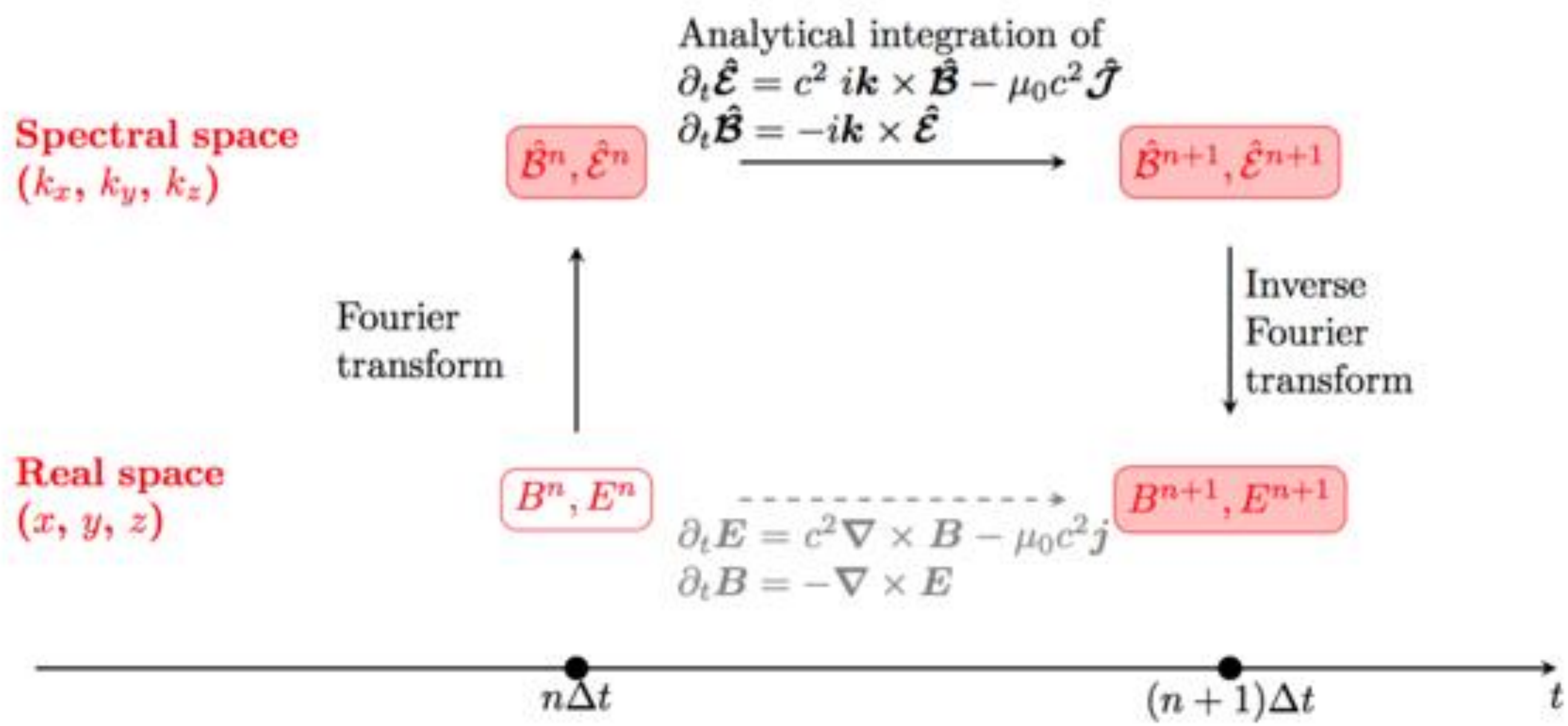
Traditional PIC codes	Cylindrical PIC codes
Use a full **3D** mesh	Use a few 2D meshes in (r, z) (one mesh per azimuthal mode)

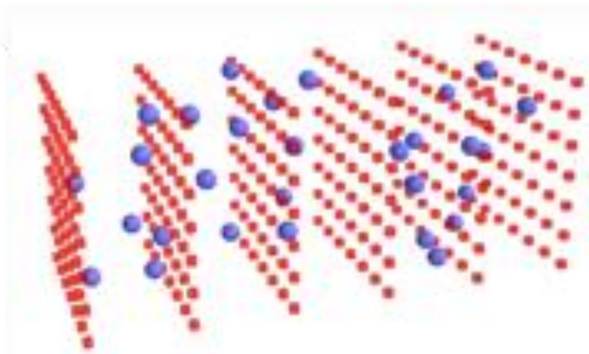




In the case of laser-wakefield acceleration:
Cylindrical codes use 2 azimuthal modes (m=0 and m=1),
which requires vastly less memory & compute power than full 3D.

Cartesian PSATD

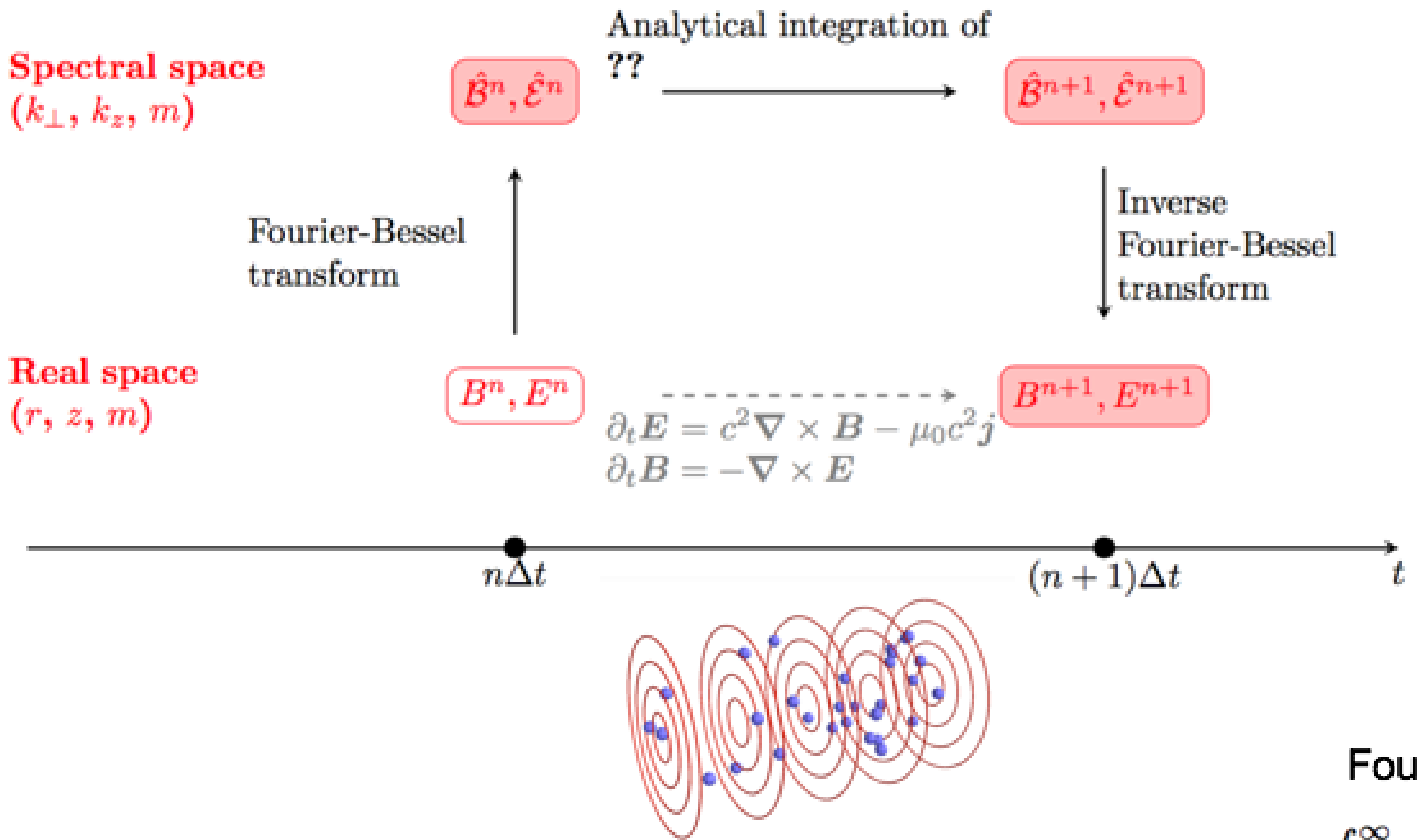




Fourier transform:

$$\hat{\mathcal{F}}_u(k_x, k_y, k_z) = \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} dz\; F_u(x, y, z) e^{-i(k_x x + k_y y + k_z z)}$$

Quasi-cylindrical PSATD



Fourier-Bessel transform:

$$\hat{\mathcal{F}}_{z,m}(k_z, k_\perp) = \int_{-\infty}^{\infty} dz \int_{-\infty}^{\infty} r\, dr\; F_{z,m}(z,r)\; J_m(k_\perp r) e^{-ik_z z}$$

$$\hat{\mathcal{F}}_{+,m}(k_z, k_\perp) = \int_{-\infty}^{\infty} dz \int_{-\infty}^{\infty} r\, dr\; \frac{F_{r,m}(z,r) - iF_{\theta,m}(z,r)}{2}\; J_{m+1}(k_\perp r) e^{-ik_z z}$$

$$\hat{\mathcal{F}}_{-,m}(k_z, k_\perp) = \int_{-\infty}^{\infty} dz \int_{-\infty}^{\infty} r\, dr\; \frac{F_{r,m}(z,r) + iF_{\theta,m}(z,r)}{2}\; J_{m-1}(k_\perp r) e^{-ik_z z}$$

Lehe et al., CPC, 2016

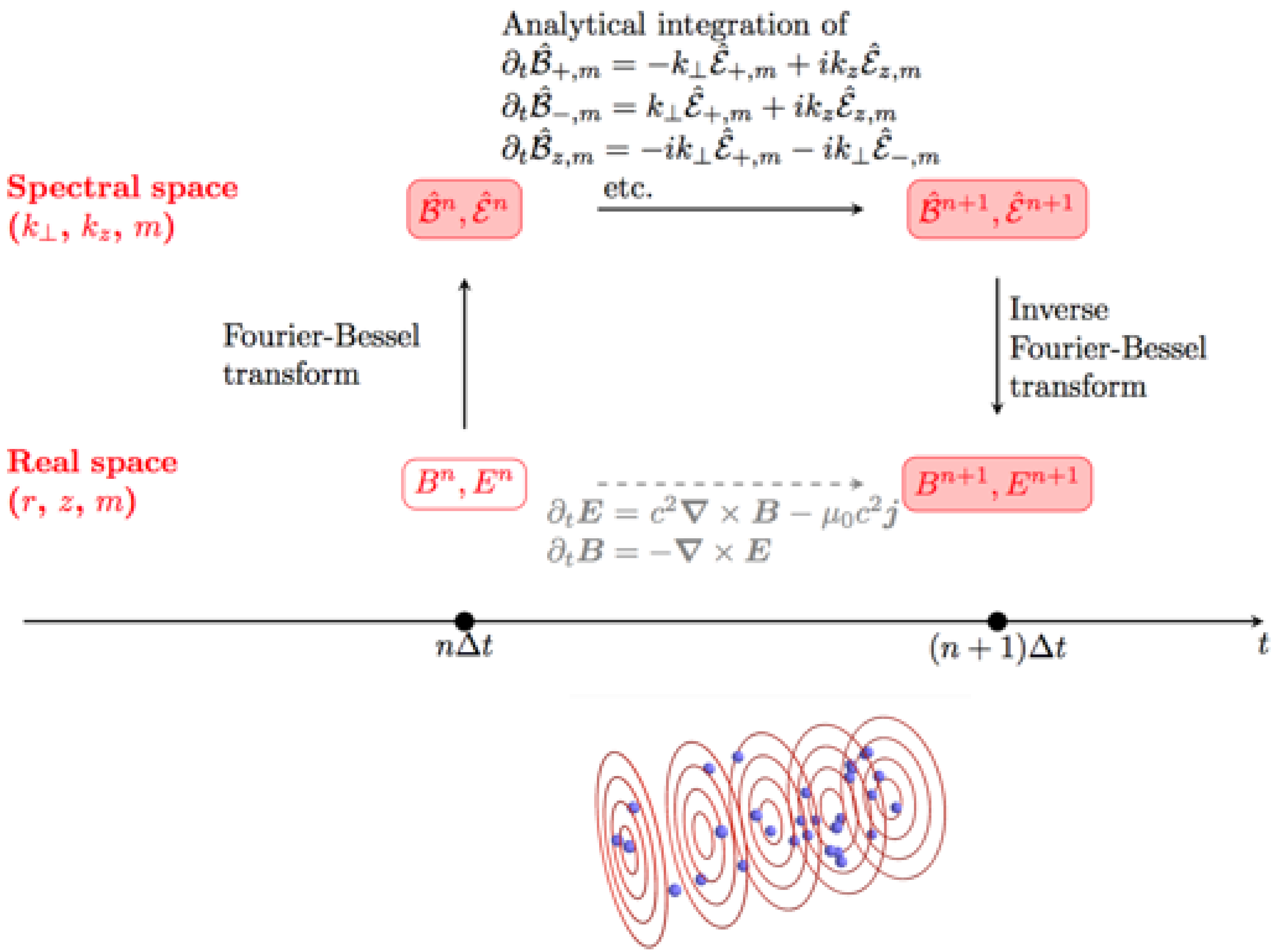
The Fourier-Bessel transformation has a simple spectral representation of the differential operators

Operator	Gradient: $\boldsymbol{F} = \boldsymbol{\nabla} S$	Curl: $\boldsymbol{F} = \boldsymbol{\nabla} \times \boldsymbol{V}$	Divergence: $F = \boldsymbol{\nabla} \cdot \boldsymbol{V}$
Spectral Cartesian representation	$\hat{\boldsymbol{\mathcal{F}}} = i\boldsymbol{k}\hat{\mathcal{S}}$ $i.e. \begin{cases} \hat{\mathcal{F}}_x = ik_x\hat{\mathcal{S}} \\ \hat{\mathcal{F}}_y = ik_y\hat{\mathcal{S}} \\ \hat{\mathcal{F}}_z = ik_z\hat{\mathcal{S}} \end{cases}$	$\hat{\boldsymbol{\mathcal{F}}} = i\boldsymbol{k} \times \hat{\boldsymbol{\mathcal{V}}}$ $i.e. \begin{cases} \hat{\mathcal{F}}_x = ik_y\hat{\mathcal{V}}_z - ik_z\hat{\mathcal{V}}_y \\ \hat{\mathcal{F}}_y = ik_z\hat{\mathcal{V}}_x - ik_x\hat{\mathcal{V}}_z \\ \hat{\mathcal{F}}_z = ik_x\hat{\mathcal{V}}_y - ik_y\hat{\mathcal{V}}_x \end{cases}$	$\hat{\mathcal{F}} = i\boldsymbol{k} \cdot \hat{\boldsymbol{\mathcal{V}}}$ $i.e.\ \hat{\mathcal{F}} = ik_x\hat{\mathcal{V}}_x + ik_y\hat{\mathcal{V}}_y + ik_z\hat{\mathcal{V}}_z$
Spectral cylindrical representation	$\begin{cases} \hat{\mathcal{F}}_{+,m} = -k_\perp\hat{\mathcal{S}}_m/2 \\ \hat{\mathcal{F}}_{-,m} = k_\perp\hat{\mathcal{S}}_m/2 \\ \hat{\mathcal{F}}_{z,m} = ik_z\hat{\mathcal{S}}_m \end{cases}$	$\begin{cases} \hat{\mathcal{F}}_{+,m} = k_z\hat{\mathcal{V}}_{+,m} - ik_\perp\hat{\mathcal{V}}_{z,m}/2 \\ \hat{\mathcal{F}}_{-,m} = -k_z\hat{\mathcal{V}}_{-,m} - ik_\perp\hat{\mathcal{V}}_{z,m}/2 \\ \hat{\mathcal{F}}_{z,m} = ik_\perp\hat{\mathcal{V}}_{+,m} + ik_\perp\hat{\mathcal{V}}_{-,m} \end{cases}$	$\hat{\mathcal{F}}_m = k_\perp(\hat{\mathcal{V}}_{+,m} - \hat{\mathcal{V}}_{-,m}) + ik_z\hat{\mathcal{V}}_{z,m}$

$$\partial_t \boldsymbol{B} = -\boldsymbol{\nabla} \times \boldsymbol{E} \quad \longrightarrow \quad \begin{aligned} \partial_t\hat{\mathcal{B}}_{+,m} &= -k_z\hat{\mathcal{E}}_{+,m} + ik_\perp\hat{\mathcal{E}}_{z,m}/2 \\ \partial_t\hat{\mathcal{B}}_{-,m} &= k_z\hat{\mathcal{E}}_{+,m} + ik_\perp\hat{\mathcal{E}}_{z,m}/2 \\ \partial_t\hat{\mathcal{B}}_{z,m} &= -ik_\perp\hat{\mathcal{E}}_{+,m} - ik_\perp\hat{\mathcal{E}}_{-,m} \end{aligned}$$

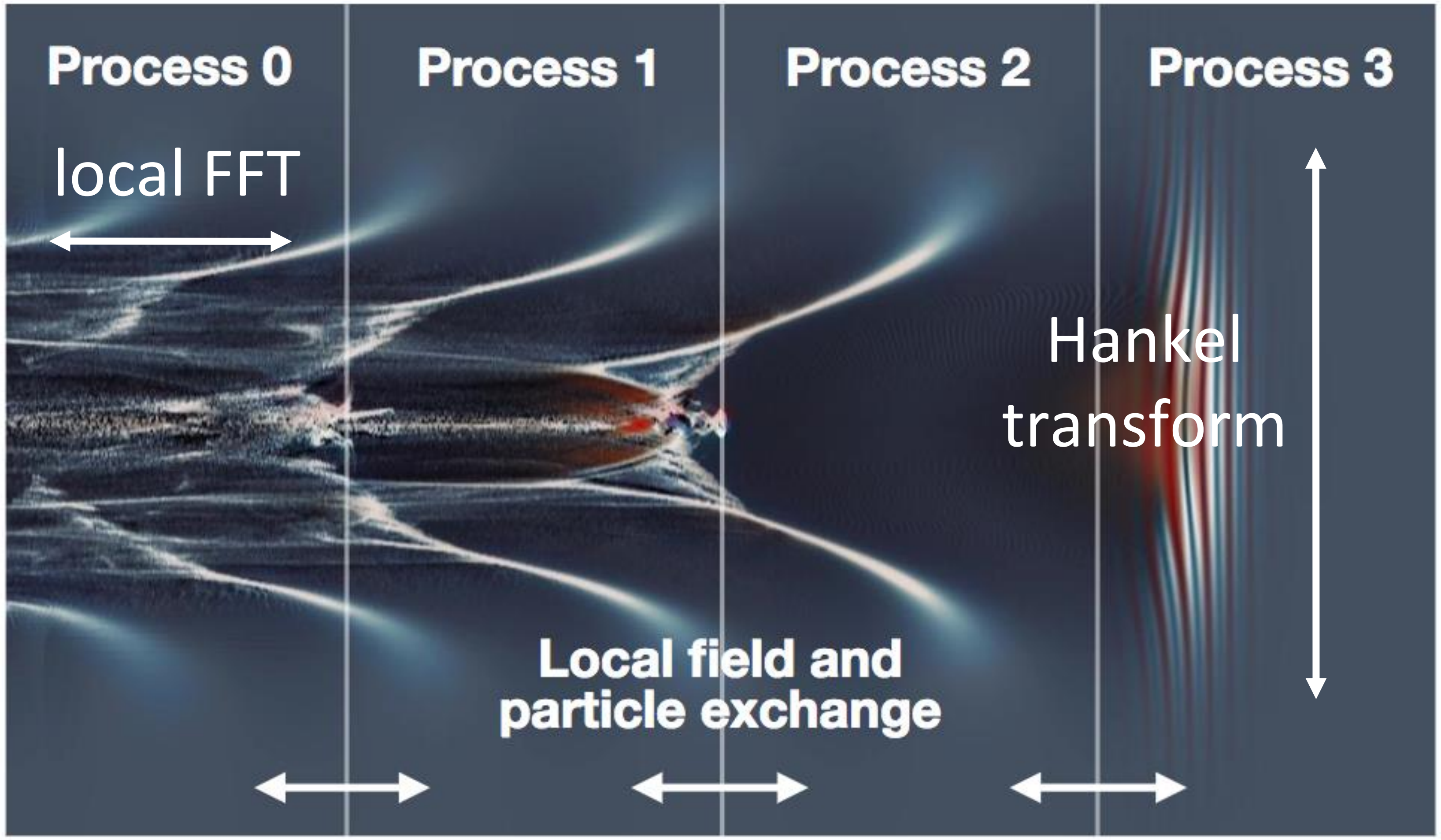
Lehe et al., PRE, 2016

Quasi-cylindrical PSATD



Parallelization

No similar MPI decomposition, with local Hankel transform…



But the Hankel transform is still parallelized across shared-memory hardware (multi-core node, GPU)

Outline

- Finite-difference vs. spectral EM-PIC

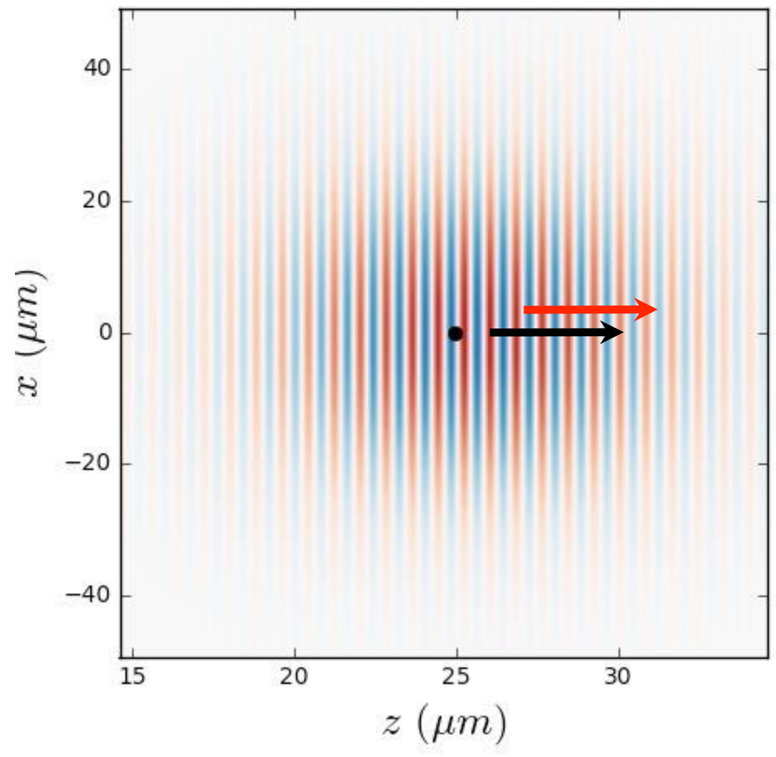
- Recent developments for spectral solvers
  - Parallelization strategy
  - Pseudo-spectral cylindrical algorithm

- Some added benefits of spectral solvers
  - Cancelation of E + v x B
  - Boosted-frame simulations and Galilean spectral scheme

Physics sometimes involves tight cancelation in E + v x B

Example

- Copropagating relativistic electron and laser pulse



- The force on the electron is

$$\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B} \sim \frac{\boldsymbol{E}}{\gamma^2}$$

Challenges for PIC

- The gathered E and B (from mesh to particles) must have the right amplitude and phase to allow this tight cancelation.

- Given the gathered E and B, the particle pusher should be able to reproduce this cancelation

$$\frac{d\boldsymbol{p}}{dt} = q\left(\boldsymbol{E} + \frac{\boldsymbol{p}}{\gamma m} \times \boldsymbol{B}\right)$$

Vay, Phys. Plasma, 2008

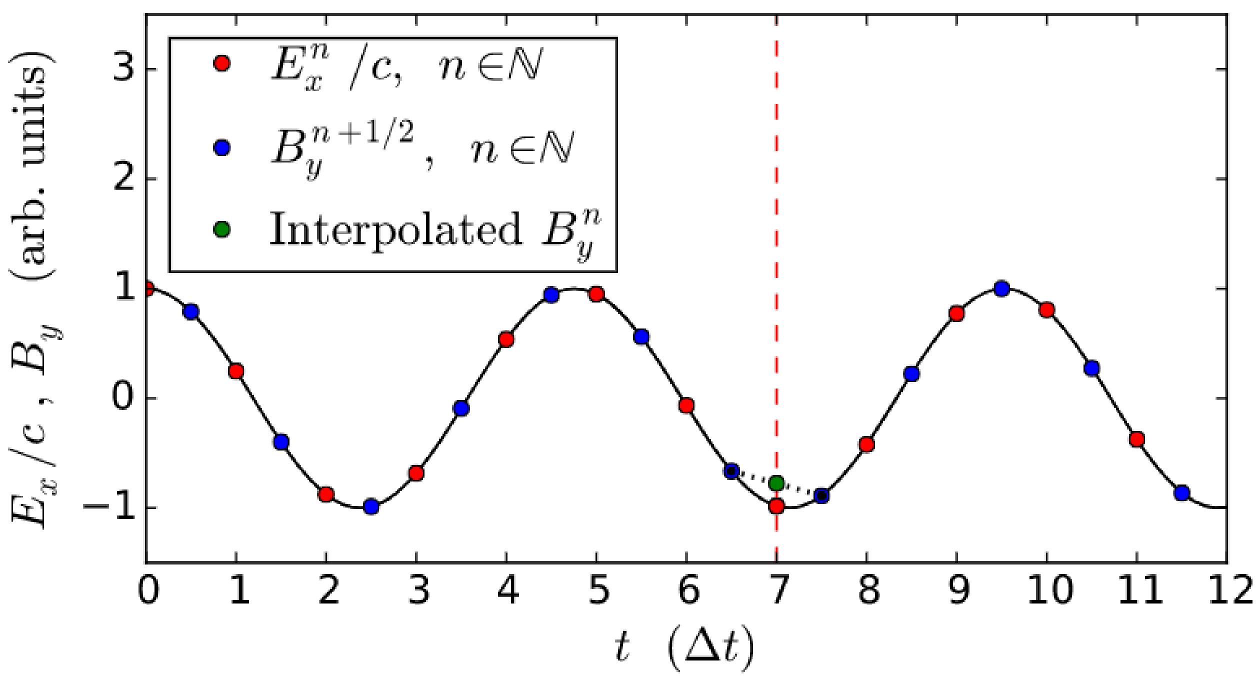
Finite-difference Time-Domain: staggering issues

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

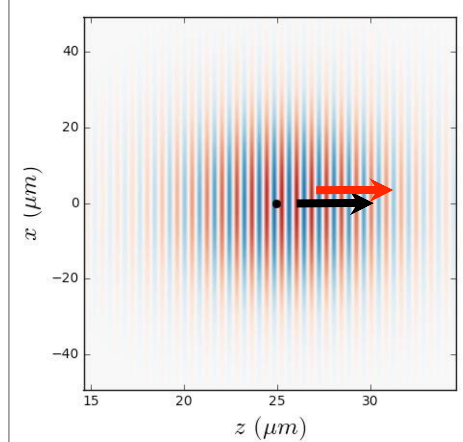
Issue

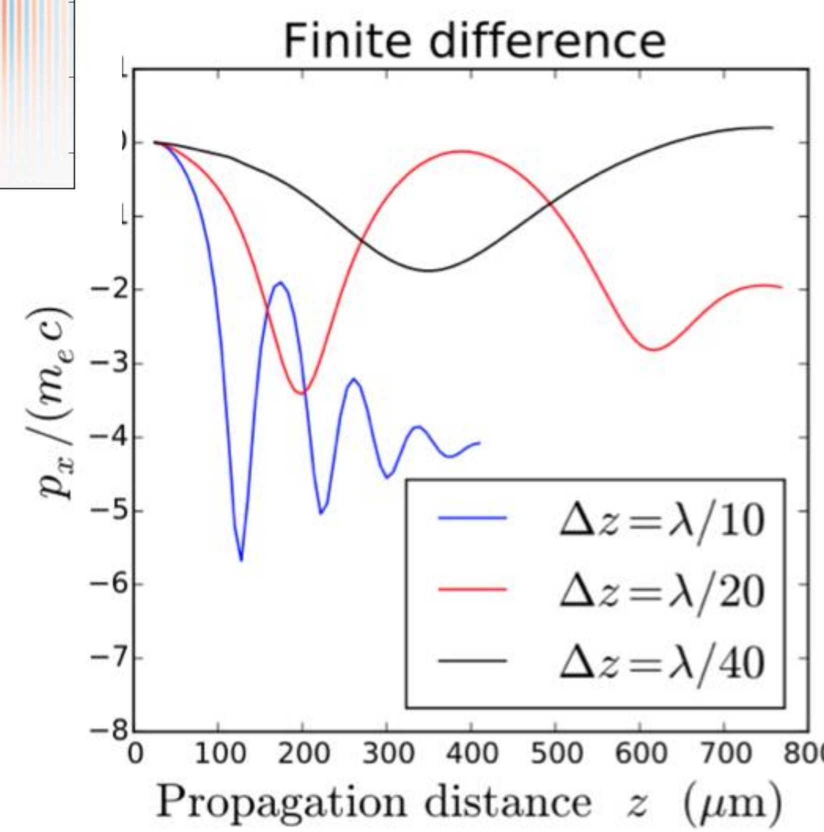
The staggering of E and B imposes that B (but not E) is averaged in time

This prevents tight cancelation.



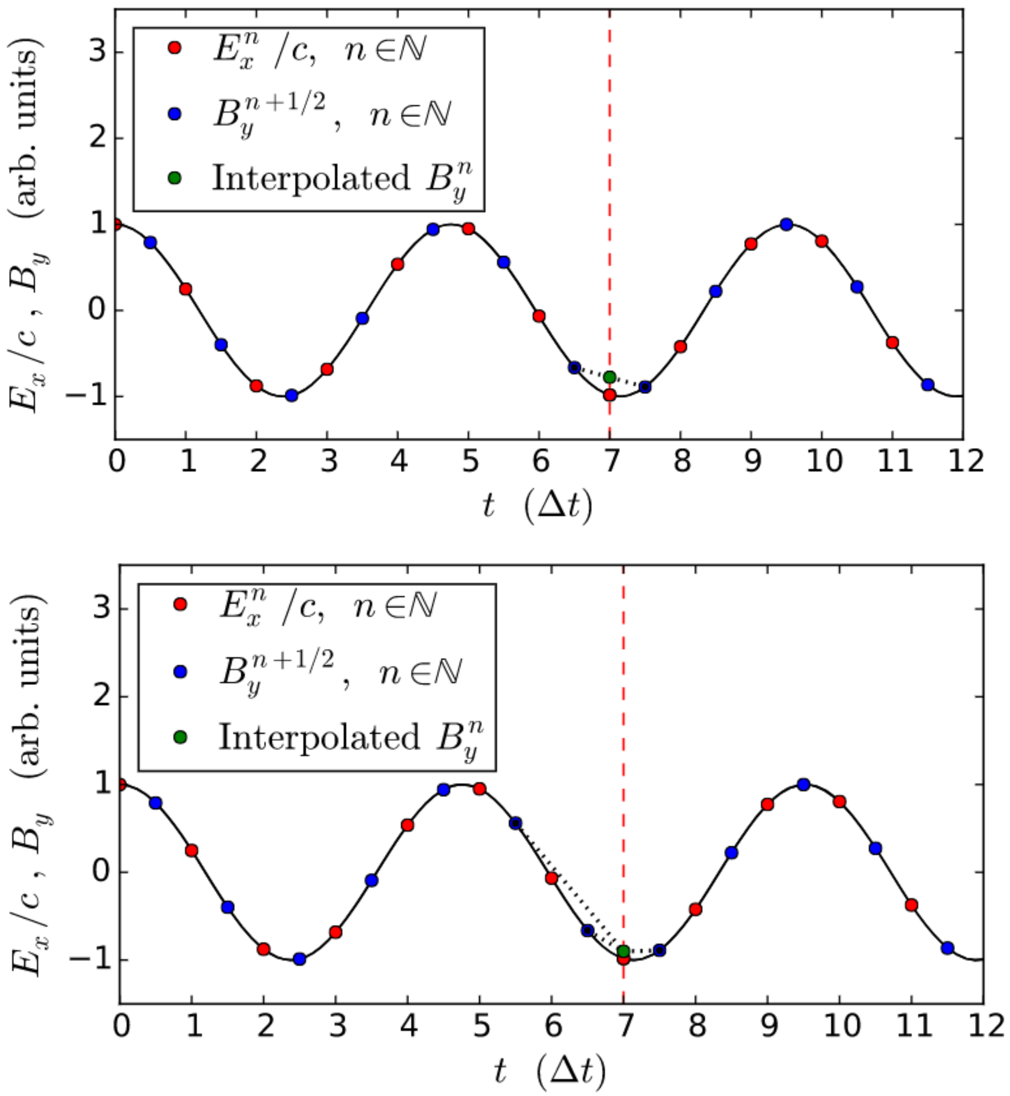
Benchmark





Some existing solutions

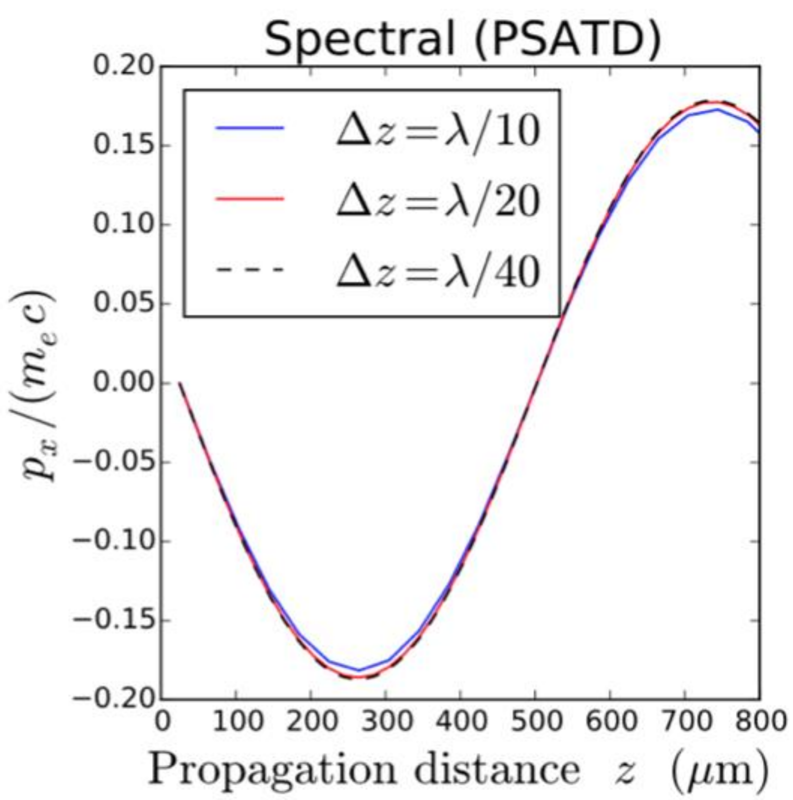
Higher-order interpolation



R. Lehe, PRSTAB, 2014

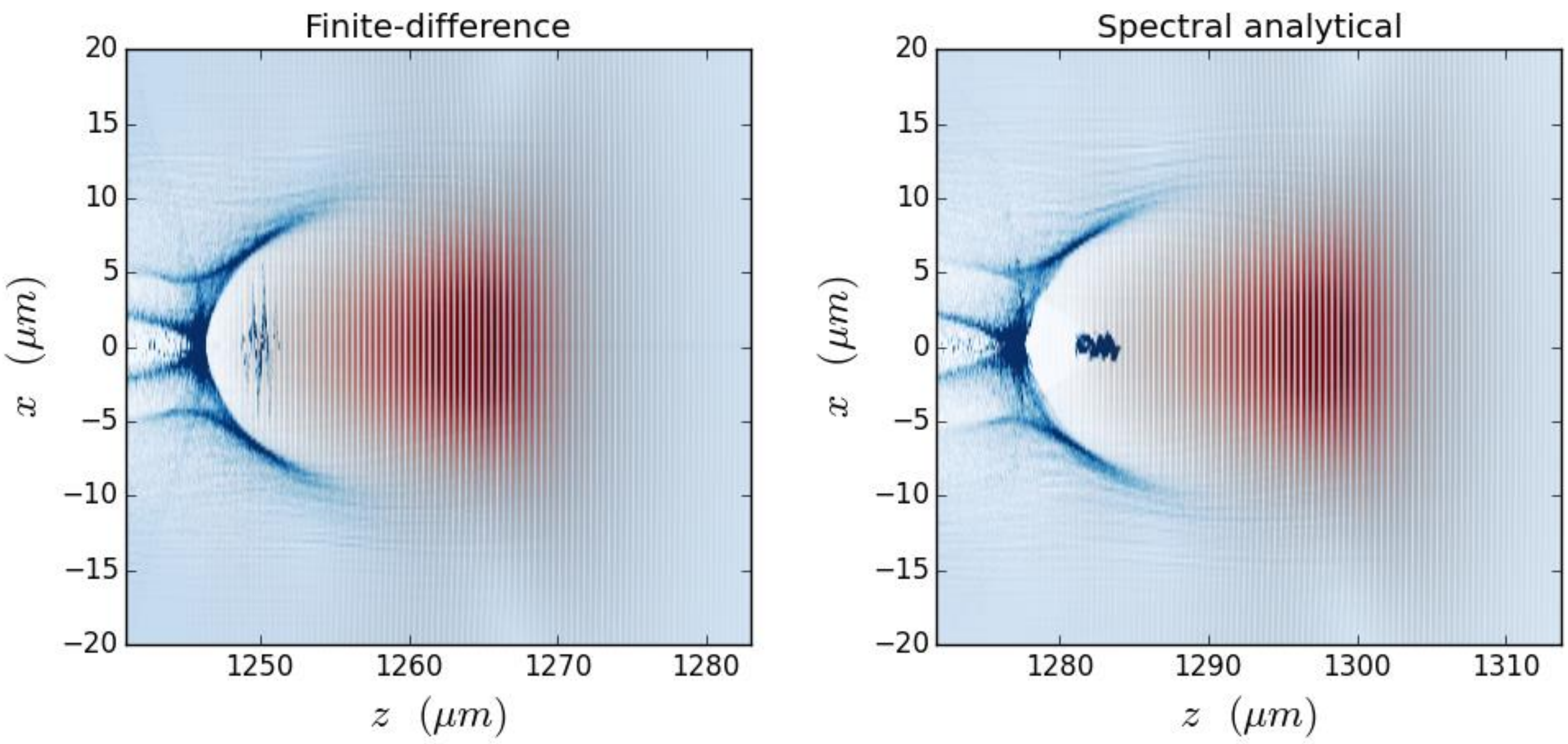
Spectral solver (PSATD)

- E and B are defined at the same time ; no need to average/interpolate B



Laser-bunch interaction

Concrete example in laser-wakefield acceleration



In the finite-difference case, the accelerated electrons feel a spurious force from the laser.

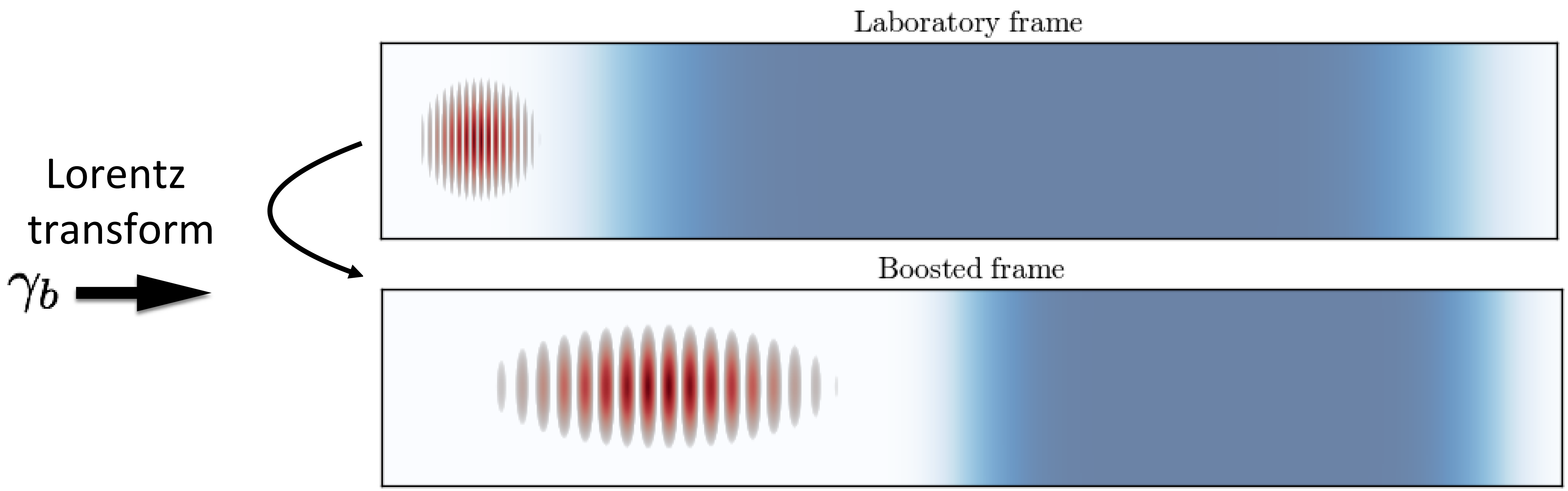
Outline

- Finite-difference vs. spectral EM-PIC

- Recent developments for spectral solvers
  - Parallelization strategy
  - Pseudo-spectral cylindrical algorithm

- Some added benefits of spectral solvers
  - Cancelation of E + v x B
  - Boosted-frame simulations and Galilean spectral scheme

Boosted-frame can speed up Particle-In-Cell simulations.



Lab-frame PIC simulation

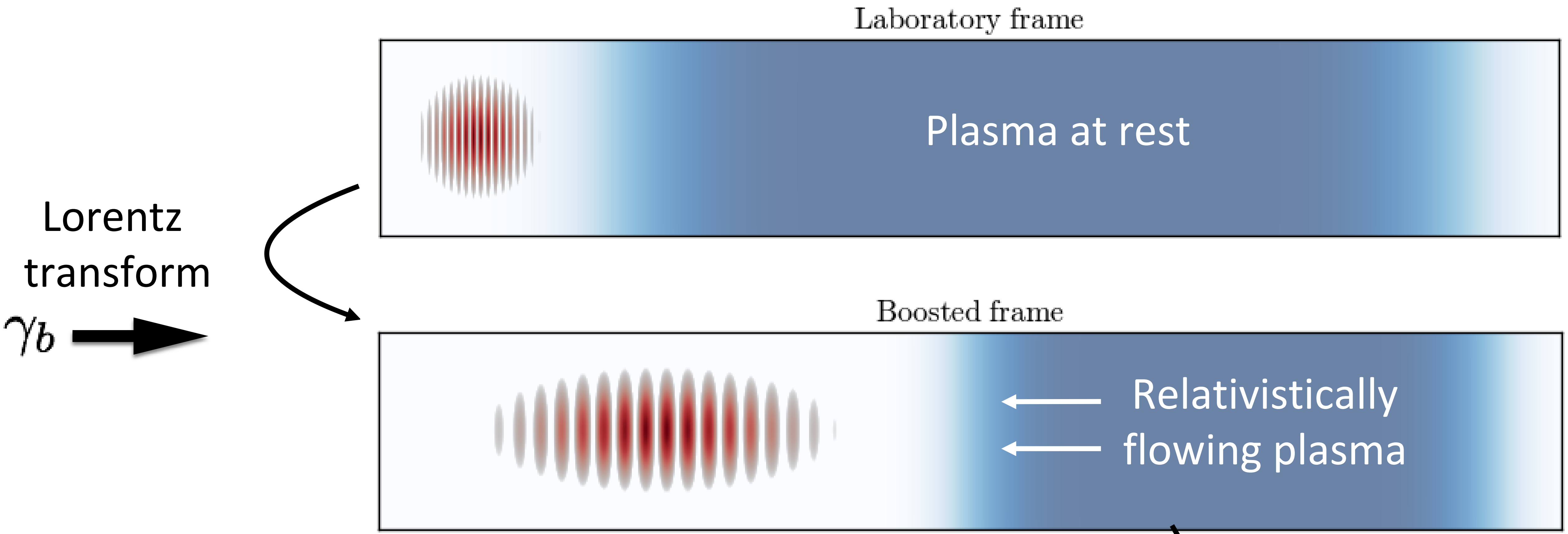
- Laser oscillations impose small grid cells and small timestep
- Propagating through the entire plasma is computationally expensive

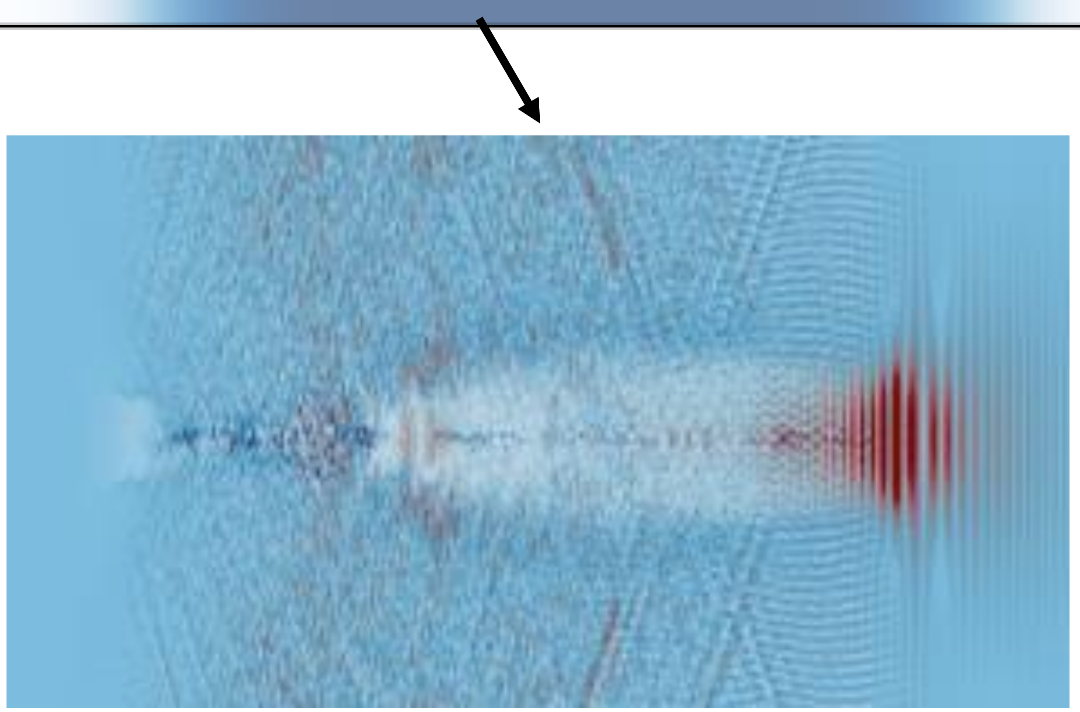
Boosted-frame PIC simulation

- Transformed laser oscillations relax constraints on grid cell and timestep.
- Computational speed up of the order $2\gamma_b^2$

Vay, PRL, 2007

Problem: the Numerical Cherenkov Instability





Problem

Relativistically flowing plasmas are numerically unstable, with the Particle-In-Cell algorithm

Existing solutions

"Magic" time step

- There is (solver-dependent) time step that minimizes the instability

J-L Vay et al, JCP, 230, 15, (2011);
B. Godfrey and J-L Vay, JCP, 267, 1-6 (2014)

Filtering of E and B

- Filter the gathered E and B in a particular way that suppresses the instability

B. Godfrey and J-L Vay, JCP, 267, 1-6 (2014);
B. Godfrey and J-L Vay, CPC, 196, 221-225 (2015)

Modified dispersion

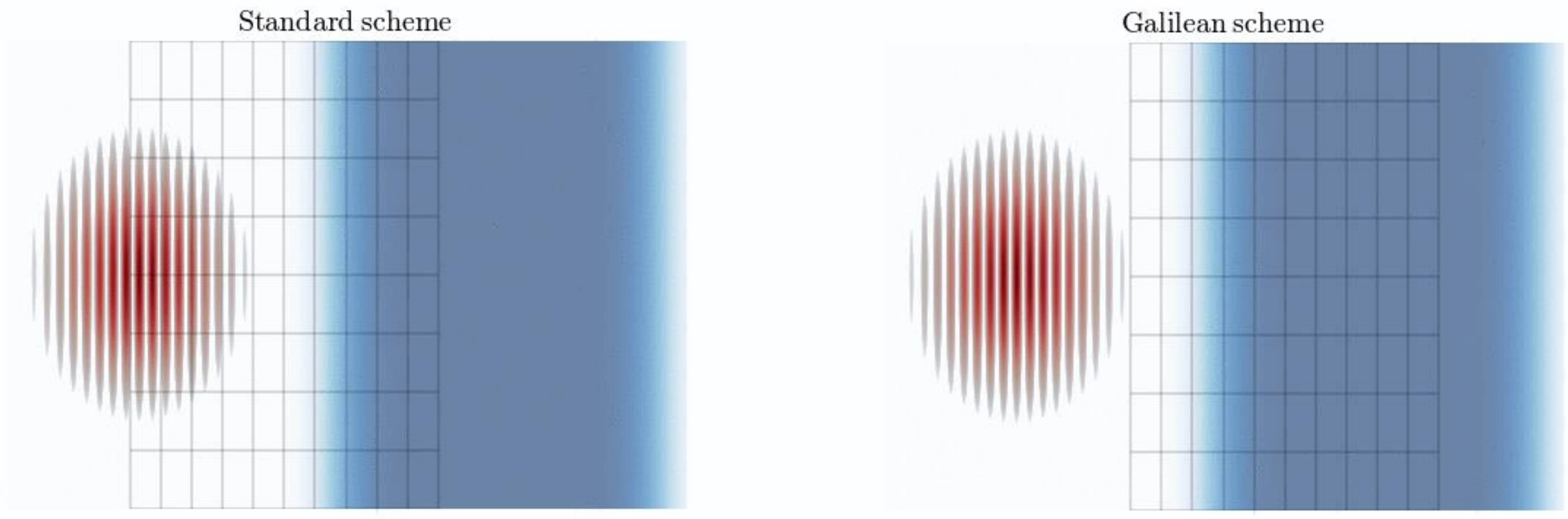
- Use small dt with spectral solver to decouple instability from physics
- Tweak the dispersion relation of the Maxwell solver at the main unstable frequency

P. Yu et al, CPC, 197, 144-152 (2015)

Galilean transformation

- Simulates physics in a grid that moves with the plasma.
- Requires PSATD solver.
  M. Kirchen et al., PoP (2016)
  R. Lehe et al., PRE (2016)

The Galilean scheme: concept

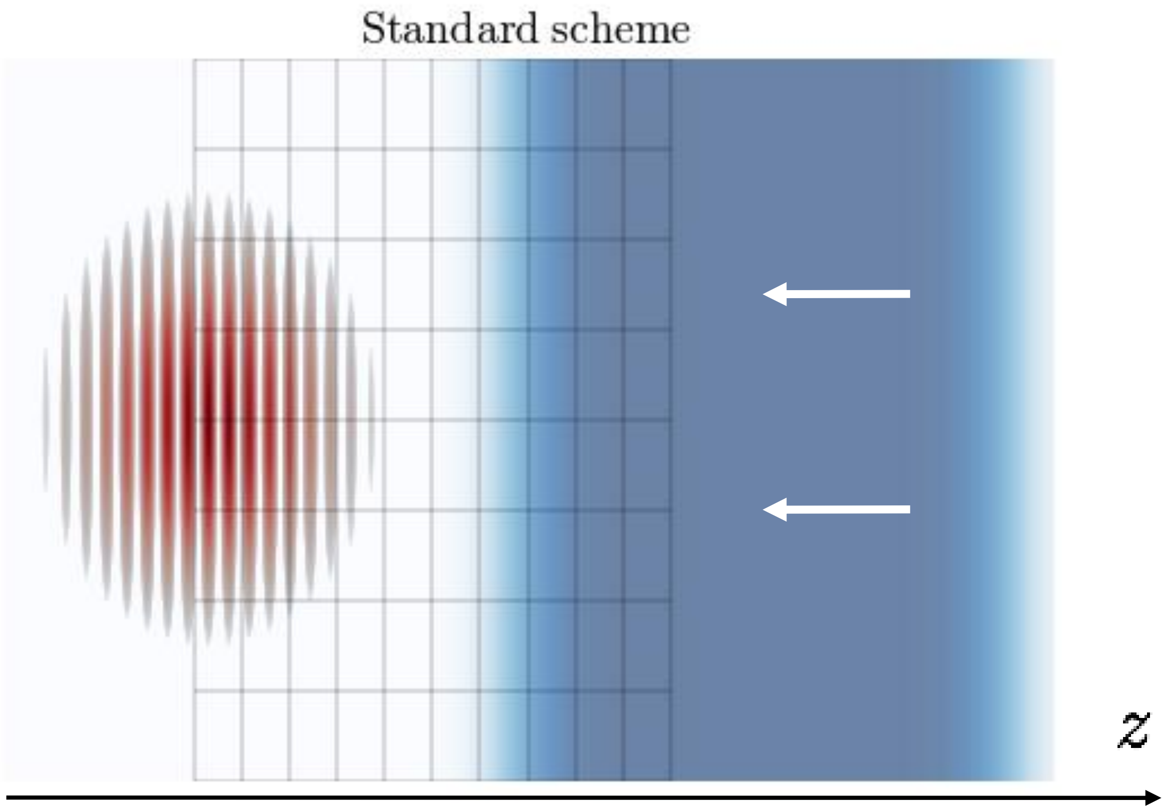


Intuition: The numerical instability arises because the bulk of the plasma flows through a fixed grid.

Solution: The grid should move along with the plasma.

Proposed by
Manuel Kirchen
(CFEL / U. Hamburg)

The Galilean scheme: mathematical formulation

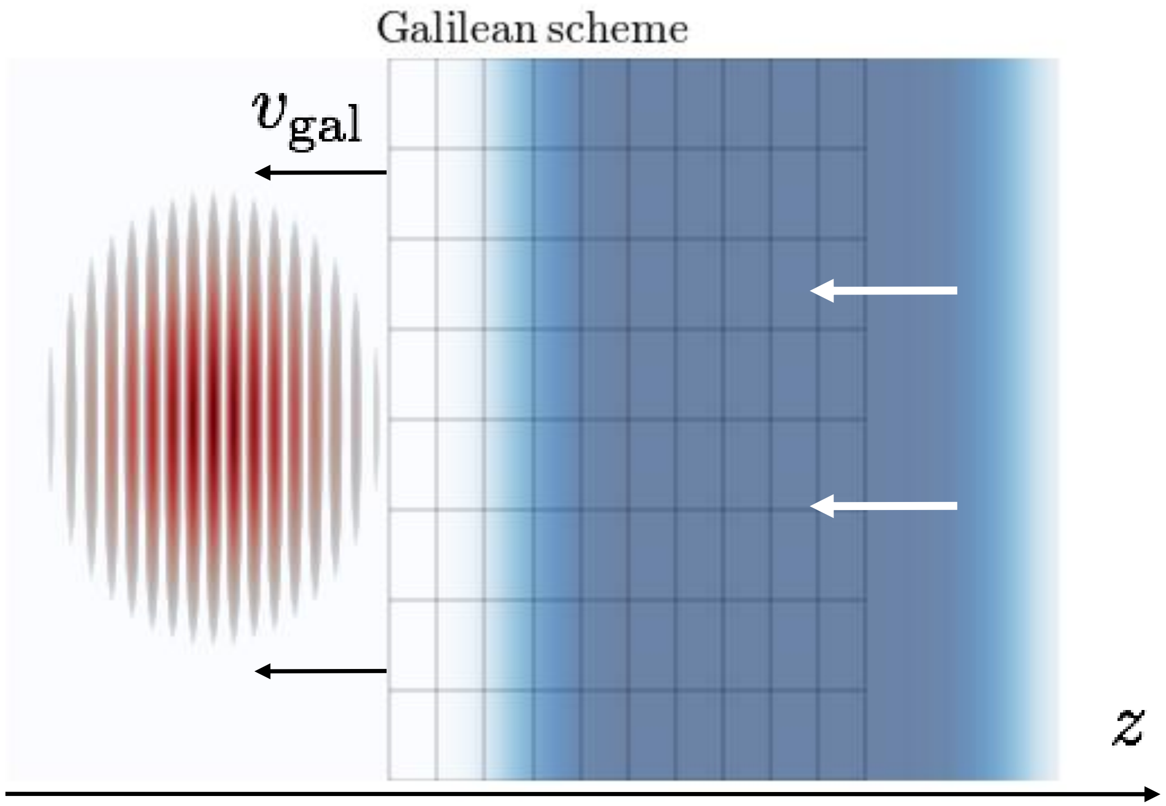


Grid = fixed values of z

Integrate on the grid

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}$$
$$\frac{1}{c^2}\frac{\partial \boldsymbol{E}}{\partial t} = \nabla \times \boldsymbol{B} - \mu_0 \boldsymbol{j}$$

assuming $\boldsymbol{j}(x, y, z, t)$ is constant over one timestep



Grid = fixed values of $z' = z - v_{\mathrm{gal}} t$

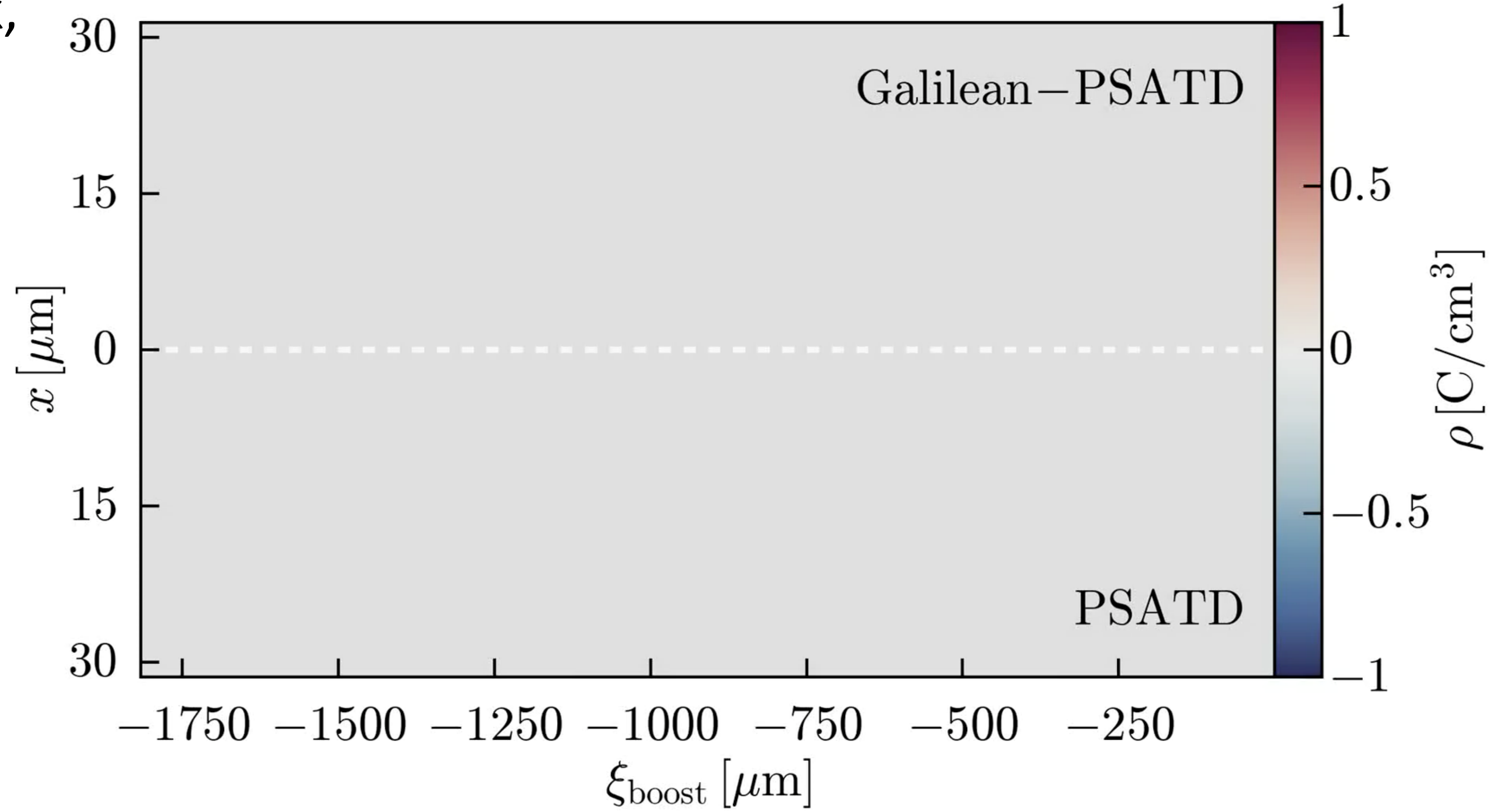
Integrate on the grid

$$\left(\frac{\partial}{\partial t} - \boldsymbol{v}_{gal} \cdot \boldsymbol{\nabla}'\right) \boldsymbol{B} = -\boldsymbol{\nabla}' \times \boldsymbol{E}$$
$$\frac{1}{c^2}\left(\frac{\partial}{\partial t} - \boldsymbol{v}_{gal} \cdot \boldsymbol{\nabla}'\right) \boldsymbol{E} = \boldsymbol{\nabla}' \times \boldsymbol{B} - \mu_0 \boldsymbol{j}$$

assuming $\boldsymbol{j}(x, y, z', t)$ is constant over one timestep

Implementation in the code FBPIC

When the Galilean equations are implemented in a PSATD framework, the instability is suppressed!



Concept & applications:
M. Kirchen et al., PoP (2016)
Algorithm & math:
R. Lehe et al., PRE (2016)

Conclusion

- EM PIC face many challenges for accurate simulations
- In many cases, spectral (PSATD) solver can mitigate those challenges, at the cost of a more intricate MPI pattern.
- But PSATD is not a universal solution, and other solutions exist.

Thank you for your attention