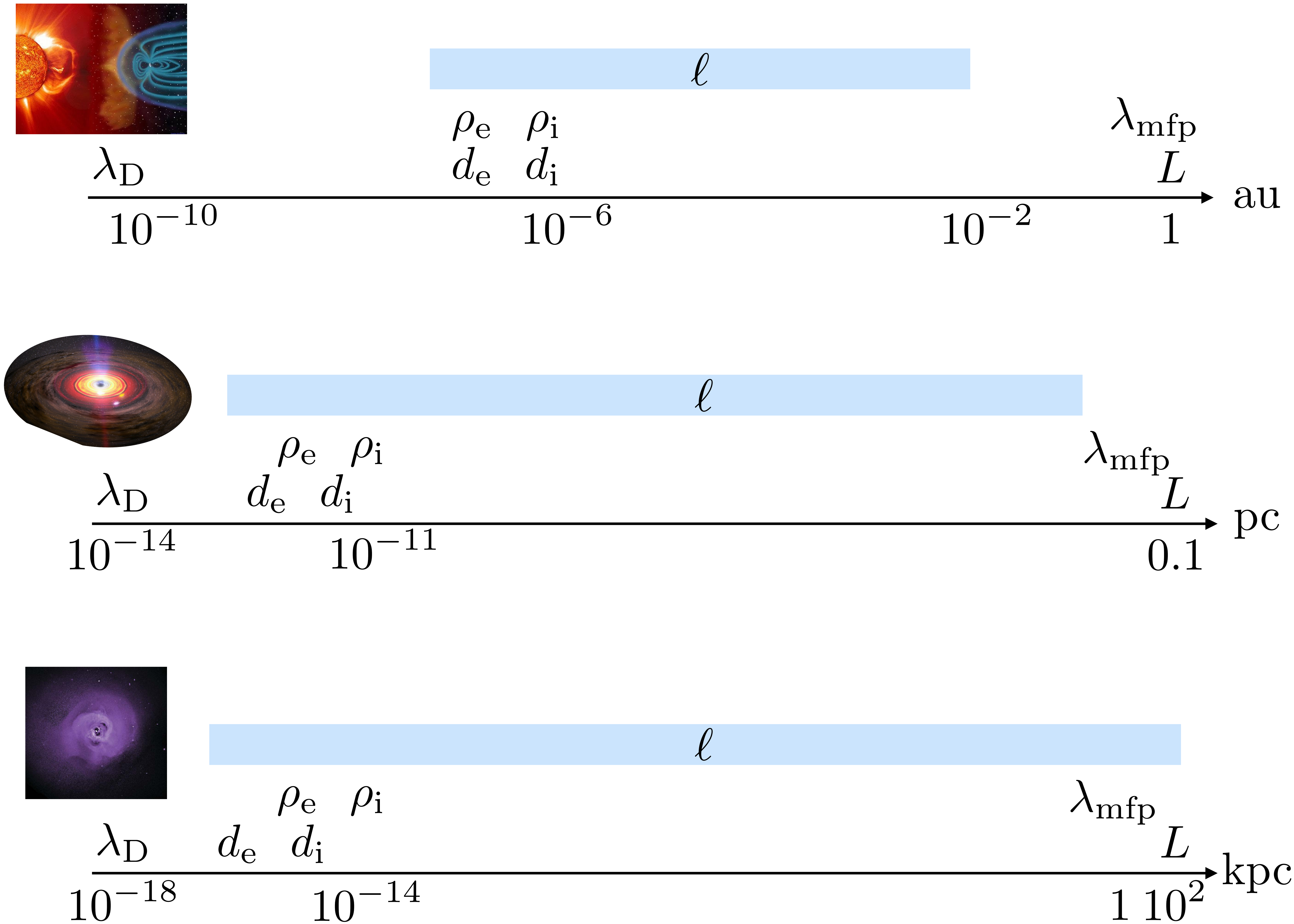
an extremely abridged and biased introduction to

Numerical Methods for Plasma Kinetics

Matthew Kunz

Princeton University



ℓ
ρ_e ρ_i
λ_mfp
λ_D
d_e d_i
L
au
10^{-10} 10^{-6} 10^{-2} 1
ℓ
ρ_e ρ_i
λ_mfp
λ_D
d_e d_i
L
pc
10^{-14} 10^{-11} 0.1
ℓ
ρ_e ρ_i
λ_mfp
λ_D
d_e d_i
L
kpc
10^{-18} 10^{-14} 1 10^{2}

Vlasov-(Landau)-Maxwell kinetics	everything \ {  }
Hybrid kinetics	$\lambda_{\rm D},\ m_{\rm e}/m_{\rm i} \to 0$ fluid electrons, kinetic ions
Gyrokinetics	$\omega/\Omega \ll 1 \quad k_\parallel L \sim k_\perp \rho \sim 1$ and small fluctuations
Kinetic MHD	$k\rho \sim \omega/\Omega \ll 1,\ \mathrm{Ma} \sim 1$

Landau fluid	(magneto)fluid equations + closure mimicking collisionless damping
Braginskii-MHD	fluid equations + anisotropic transport due to magnetization
Hall-MHD	MHD + finite skin depth
MHD	so 20th century…

First, a brief review of where Vlasov comes from…

(due to Klimontovich)

$$F_\alpha(\boldsymbol{r},\boldsymbol{v},t) = \sum_{i=1}^{N_\alpha} \delta(\boldsymbol{r} - \boldsymbol{R}_{\alpha i}(t))\delta(\boldsymbol{v} - \boldsymbol{V}_{\alpha i}(t))$$

positions of particles of species α

velocities of particles of species α

$$\lim_{\mathrm{d}\boldsymbol{r}\mathrm{d}\boldsymbol{v}\to 0} \int \mathrm{d}\boldsymbol{r}\mathrm{d}\boldsymbol{v}\, F_\alpha(\boldsymbol{r},\boldsymbol{v},t)$$ is either 1 or 0

if you know $\boldsymbol{R}_{\alpha i}(0)$ and $\boldsymbol{V}_{\alpha i}(0)$, and can solve

$$\frac{\mathrm{d}\boldsymbol{R}_{\alpha i}}{\mathrm{d}t} = \boldsymbol{V}_{\alpha i} \qquad \frac{\mathrm{d}\boldsymbol{V}_{\alpha i}}{\mathrm{d}t} = \frac{q_\alpha}{m_\alpha}\left[\boldsymbol{E}_\mathrm{m}(\boldsymbol{R}_{\alpha i},t) + \frac{1}{c}\boldsymbol{V}_{\alpha i}\times\boldsymbol{B}_\mathrm{m}(\boldsymbol{R}_{\alpha i},t)\right]$$

then you know everything. Done.

“Microphysical” fields computed from Maxwell’s equations

$$\nabla \cdot \boldsymbol{B}_{\mathrm{m}} = 0$$

$$\nabla \cdot \boldsymbol{E}_{\mathrm{m}} = 4\pi \sum_{\alpha} q_{\alpha} \int \mathrm{d}\boldsymbol{v}\, F_{\alpha}(\boldsymbol{r}, \boldsymbol{v}, t)$$

$$\nabla \times \boldsymbol{B}_{\mathrm{m}} = \frac{1}{c}\frac{\partial \boldsymbol{E}_{\mathrm{m}}}{\partial t} + \frac{4\pi}{c} \sum_{\alpha} q_{\alpha} \int \mathrm{d}\boldsymbol{v}\, \boldsymbol{v} F_{\alpha}(\boldsymbol{r}, \boldsymbol{v}, t)$$

$$\nabla \times \boldsymbol{E}_{\mathrm{m}} = -\frac{1}{c}\frac{\partial \boldsymbol{B}_{\mathrm{m}}}{\partial t}$$

Rather than evolve $\boldsymbol{R}_{\alpha i}$ and $\boldsymbol{V}_{\alpha i}$, solve

$$\left[\frac{\partial}{\partial t} + \boldsymbol{v} \cdot \nabla + \frac{q_{\alpha}}{m_{\alpha}}\left(\boldsymbol{E}_{\mathrm{m}} + \frac{1}{c}\boldsymbol{v} \times \boldsymbol{B}_{\mathrm{m}}\right) \cdot \frac{\partial}{\partial \boldsymbol{v}}\right] F_{\alpha}(\boldsymbol{r}, \boldsymbol{v}, t) = 0$$

“Klimontovich equation”

The Klimontovich equation is equivalent to phase-space conservation, but it is NOT a statistical equation.
It *looks* like the Vlasov equation, but it is completely different!

With proper initial conditions,
it is *deterministic*, not *probabilistic* .

This makes it cumbersome… but it *does* form the basis of particle-in-cell (PIC) methods and statistical plasma kinetics.

Let’s see the latter…

Ensemble averaging over all microscopic realizations of the macroscopic plasma (which is equivalent to a coarse-graining procedure by ergodicity),

$$\left[\frac{\partial}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla} + \frac{q_\alpha}{m_\alpha}\left(\boldsymbol{E} + \frac{1}{c}\boldsymbol{v}\times\boldsymbol{B}\right)\cdot\frac{\partial}{\partial \boldsymbol{v}}\right] f_\alpha(\boldsymbol{r},\boldsymbol{v},t) =$$

$$-\frac{q_\alpha}{m_\alpha}\left\langle\left(\delta\boldsymbol{E} + \frac{1}{c}\boldsymbol{v}\times\delta\boldsymbol{B}\right)\cdot\frac{\partial F_\alpha}{\partial \boldsymbol{v}}\right\rangle$$

LHS = Vlasov equation

RHS = collisions due to discrete nature of particles

$\sim \Lambda^{-1} \doteq (n\lambda_{\mathrm{D}}^3)^{-1} \ll 1$ the LHS

this is probabilistic (even more so once the RHS is simplified) solve

$$\left[\frac{\partial}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla} + \frac{q_\alpha}{m_\alpha}\left(\boldsymbol{E} + \frac{1}{c}\boldsymbol{v}\times\boldsymbol{B}\right)\cdot\frac{\partial}{\partial \boldsymbol{v}}\right] f_\alpha(\boldsymbol{r},\boldsymbol{v},t) = \left(\frac{\partial f_\alpha}{\partial t}\right)_{\mathrm{coll}}$$

in 6D phase space (“Eulerian”)

or

solve

$$\frac{\mathrm{d}\boldsymbol{R}_{\alpha i}}{\mathrm{d}t} = \boldsymbol{V}_{\alpha i} \qquad \frac{\mathrm{d}\boldsymbol{V}_{\alpha i}}{\mathrm{d}t} = \frac{q_\alpha}{m_\alpha}\left(\boldsymbol{E}_\mathrm{m} + \frac{1}{c}\boldsymbol{V}_{\alpha i}\times\boldsymbol{B}_\mathrm{m}\right)$$

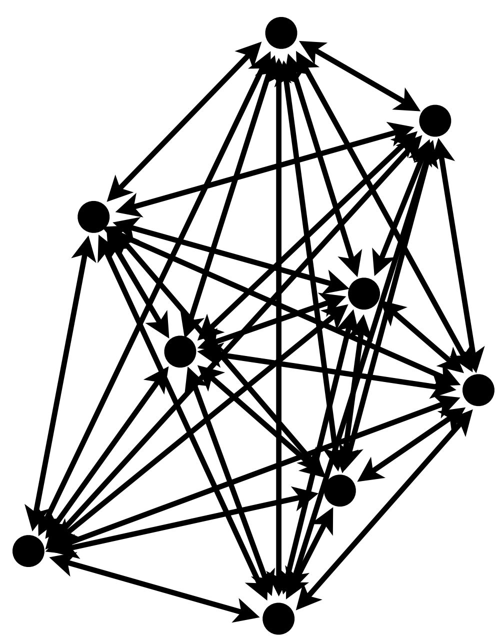
for a finite number of (macro)particles (“Lagrangian”)

(f = const on these characteristics)

In the Lagrangian case, you really don't want to do particle pairing for ~10^{10} particles per Debye cloud!

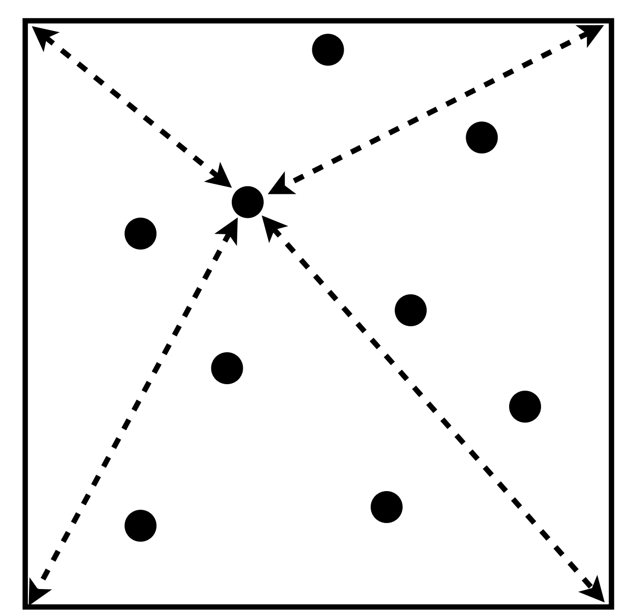
concept of (macro)particles communicating with one another electromagnetically via a grid; reduction in # of pairings

particle-particle



number of pairs: $\dfrac{N(N-1)}{2} \propto N^2$

particle-mesh (PIC)



$\propto N$

Lagrangian (Klimontovich/PIC)

- Only 3D grid needed for real space; Monte-Carlo sampling of velocity space; means that parallelization is easy and usually gives good scaling
- Easy to write
- “Unlimited” dynamical range for particle velocities; no boundary conditions on **v**

- Difficult to include explicit collisions; usually not even implemented
- Limited phase-space density resolution
- Errors from finite-size particles (smoothing)
- Load balancing issues

- √N noise! Need lots of particles to capture phase mixing, collisionless damping, and small-amplitudes fluctuations properly
- Things can go unpredictably wrong

Eulerian
(Vlasov-Landau)

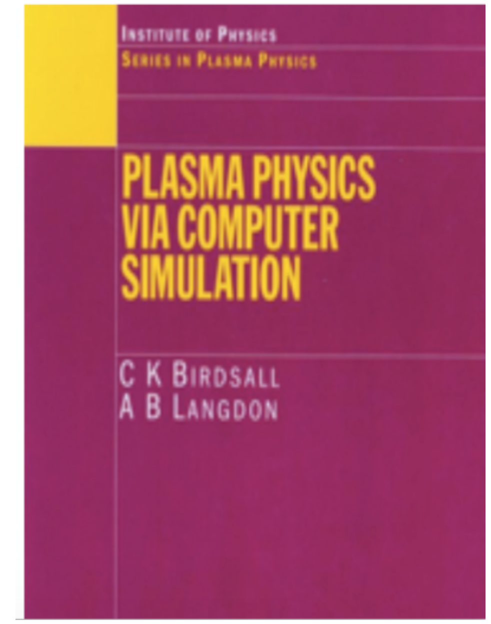
- No noise
- Good control over dissipation; easier to include collisions
- No issues if plasma very inhomogeneous

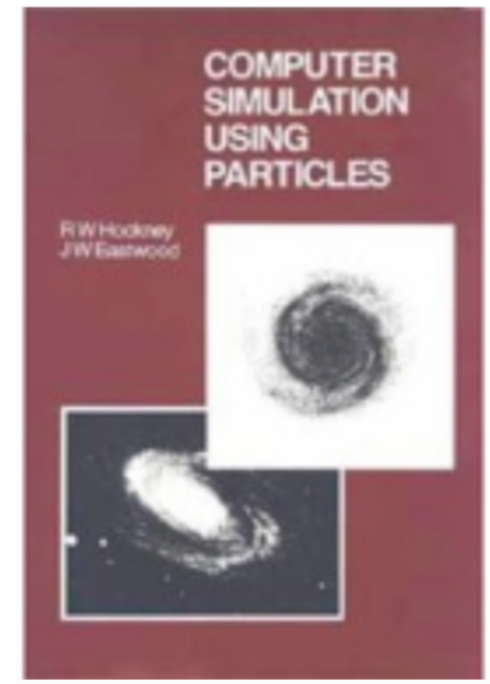
- 6D grid -> extremely expensive; often results in poor velocity-space resolution
- Difficult to parallelize efficiently

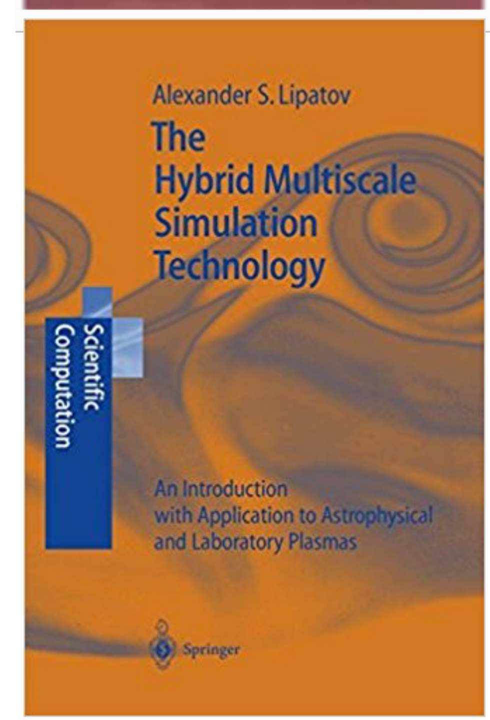
- Velocity space isn't (easily) adaptable, …

PIC Simulations: Some History

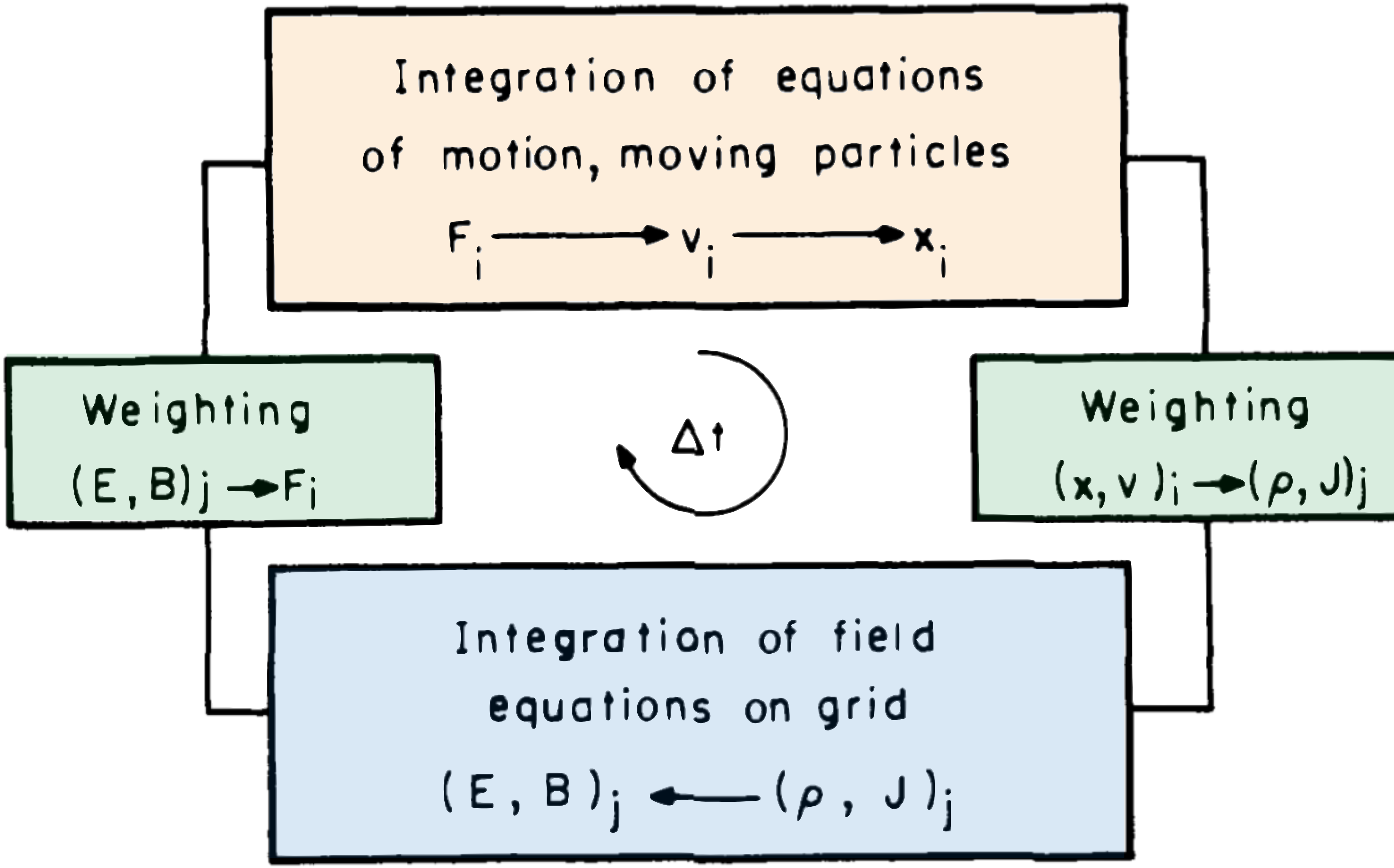
- Dawson's sheet model (1962): 1000 sheets in 1D; started late 1950s at Princeton, later @ UCLA
- Hockney, Buneman (1965): introduced grids and direct Poisson solve
- Finite-size particles and PIC (Dawson et al. 1968; Birdsall et al. 1968)
- Short-wavelength and high-frequency particle noise minimized via charge sharing and smoothing schemes; noise studied by fluctuation-dissipation theorem (Klimontovich 1967; Langdon 1979; Birdsall & Langdon 1983; Krommes 1993 for GK PIC)
- 1980s-90s 3D electromagnetic PIC booms; "PIC bibles" 1988 and 1990







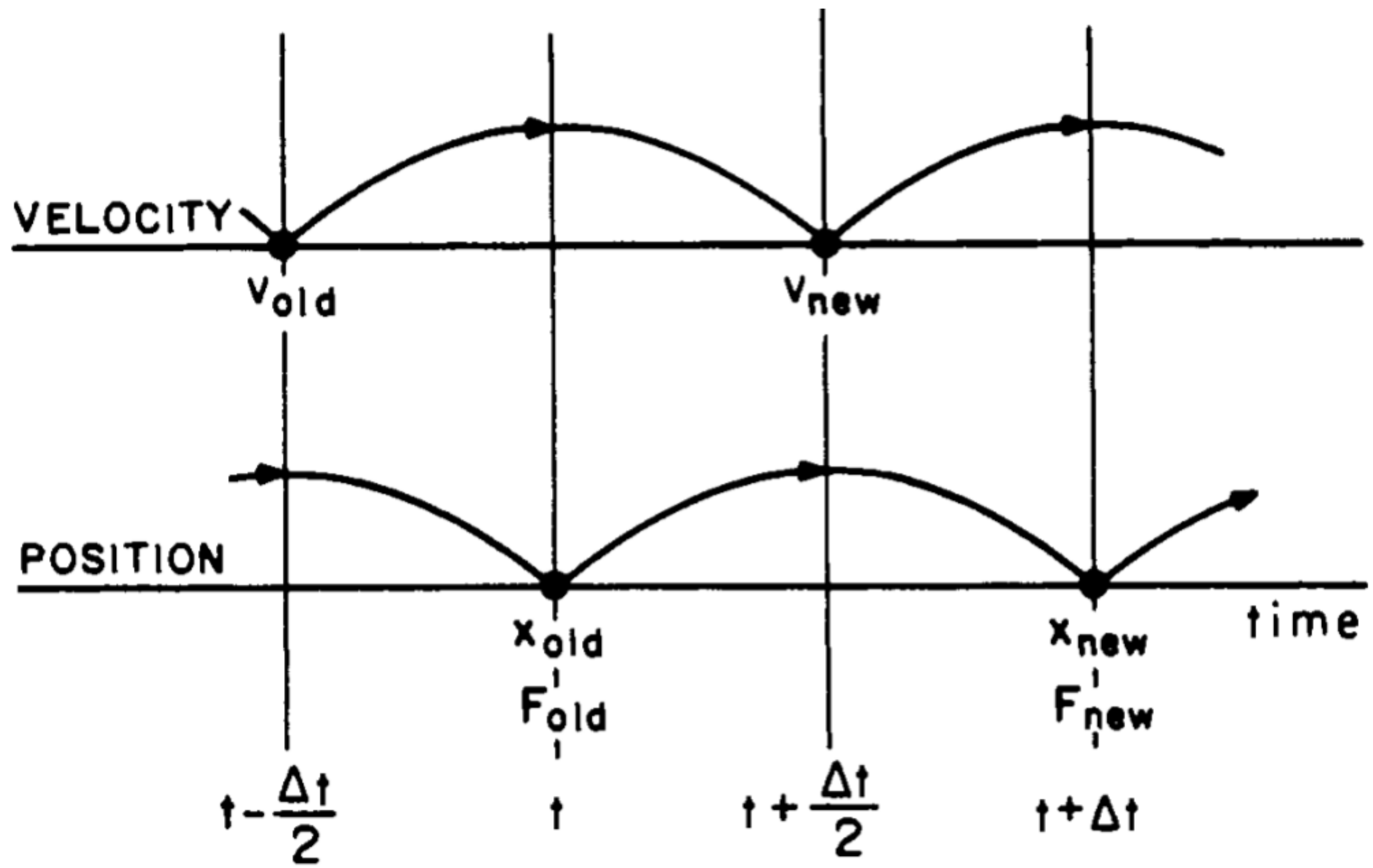
general idea:



note: sometimes fields are subcycled to reduce cost, but great care must be taken to avoid instability

Step 1: Push Particles

leapfrog algorithm



2nd-order accurate

time-reversible

symplectic

?

$$\frac{\boldsymbol{V}_i^{n+1/2} - \boldsymbol{V}_i^{n-1/2}}{\Delta t} = \boldsymbol{E}_i^n(\boldsymbol{R}_i^n) + \boldsymbol{V}_i^n \times \boldsymbol{B}^n(\boldsymbol{R}_i^n)$$

$$\frac{\boldsymbol{R}_i^{n+1} - \boldsymbol{R}_i^n}{\Delta t} = \boldsymbol{V}_i^{n+1/2}$$

Step 1: Push Particles

Crank-Nicholson (Buneman 1967): $\boldsymbol{V}_i^n = \dfrac{\boldsymbol{V}_i^{n+1/2} + \boldsymbol{V}_i^{n-1/2}}{2}$

$$\Longrightarrow \frac{\boldsymbol{V}_i^{n+1/2} - \boldsymbol{V}_i^{n-1/2}}{\Delta t} = \boldsymbol{E}_i^n(\boldsymbol{R}_i^n) + \frac{\boldsymbol{V}_i^{n+1/2} + \boldsymbol{V}_i^{n-1/2}}{2} \times \boldsymbol{B}^n(\boldsymbol{R}_i^n)$$

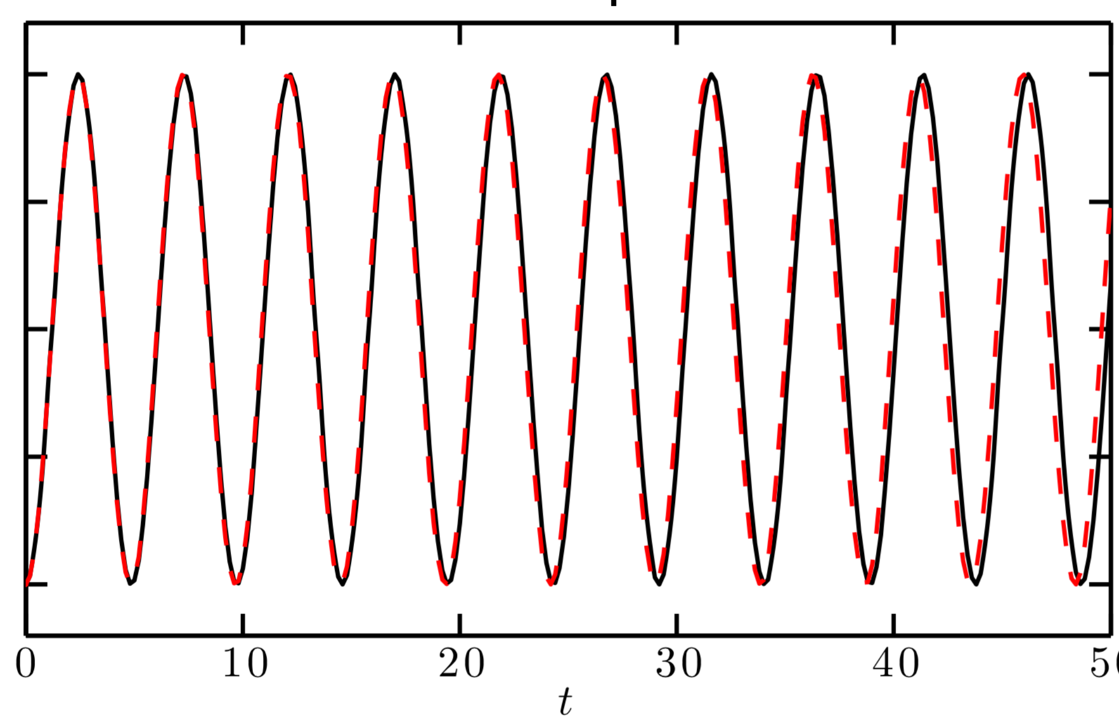
Boris (1970) algorithm (time-reversible, conserves energy):

$$\boldsymbol{V}_i^- = \boldsymbol{V}_i^{n-1/2} + \boldsymbol{E}_i^n(\boldsymbol{R}_i^n)\frac{\Delta t}{2}$$

$$\boldsymbol{V}_i^+ = \boldsymbol{V}_i^- + \left(\frac{\boldsymbol{V}_i^+ + \boldsymbol{V}_i^-}{2}\right) \times \boldsymbol{B}^n(\boldsymbol{R}_i^n)\Delta t$$

$$\boldsymbol{V}_i^{n+1/2} = \boldsymbol{V}_i^+ + \boldsymbol{E}_i^n(\boldsymbol{R}_i^n)\frac{\Delta t}{2}$$

makes small phase error



can overstep gyromotion without stability issues (just accuracy issues…)

Step 2: Deposit Particles to Grid

simulation particles are not delta functions in real space;
they represent large number of physical particles:
"macroparticles" or "Lagrangian markers"

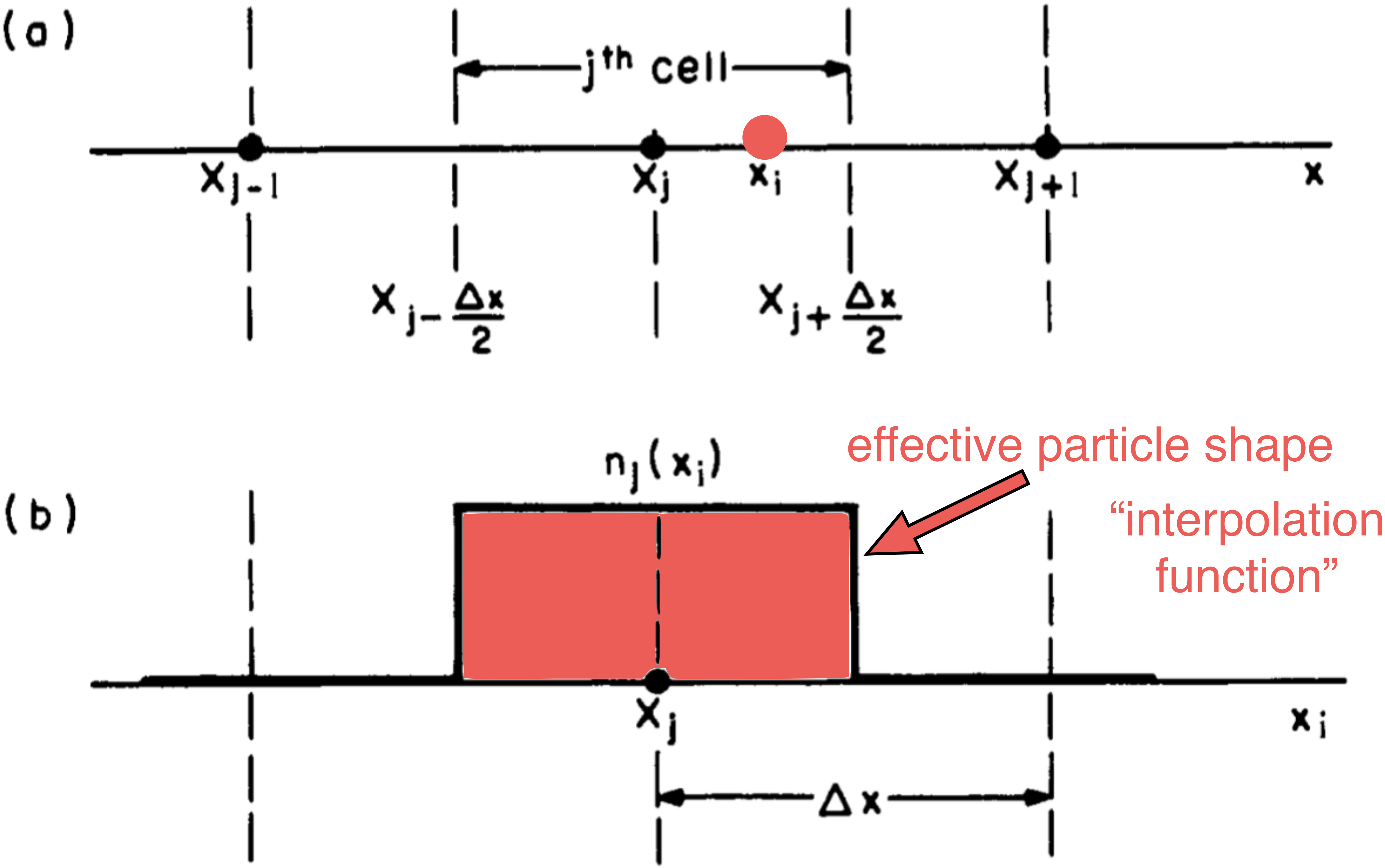
$$n_\alpha(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{v}\, F_\alpha = \sum_{i=1}^{N_\alpha} \delta(\boldsymbol{r} - \boldsymbol{R}_{\alpha i}) \rightarrow \sum_{i=1}^{N_\alpha} S(\boldsymbol{r} - \boldsymbol{R}_{\alpha i})$$

$$n_\alpha(\boldsymbol{r})\boldsymbol{u}_\alpha(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{v}\, \boldsymbol{v} F_\alpha = \sum_{i=1}^{N_\alpha} \boldsymbol{V}_{\alpha i}\, \delta(\boldsymbol{r} - \boldsymbol{R}_{\alpha i}) \rightarrow \sum_{i=1}^{N_\alpha} \boldsymbol{V}_{\alpha i}\, S(\boldsymbol{r} - \boldsymbol{R}_{\alpha i})$$

"shape function"

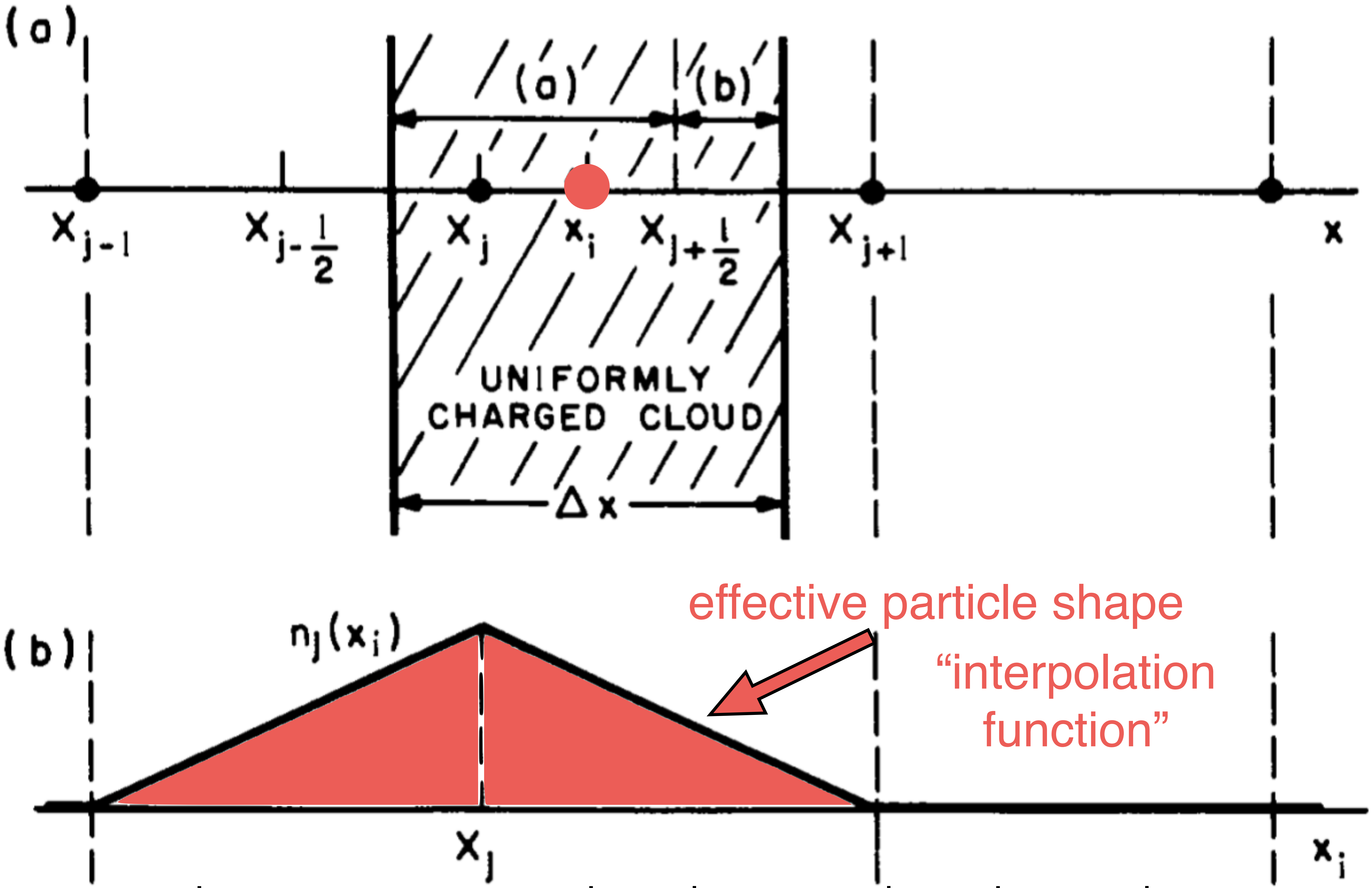
dictates how much phase-space density
is assigned to a given grid cell

0th-order particle weighting (nearest neighbor)



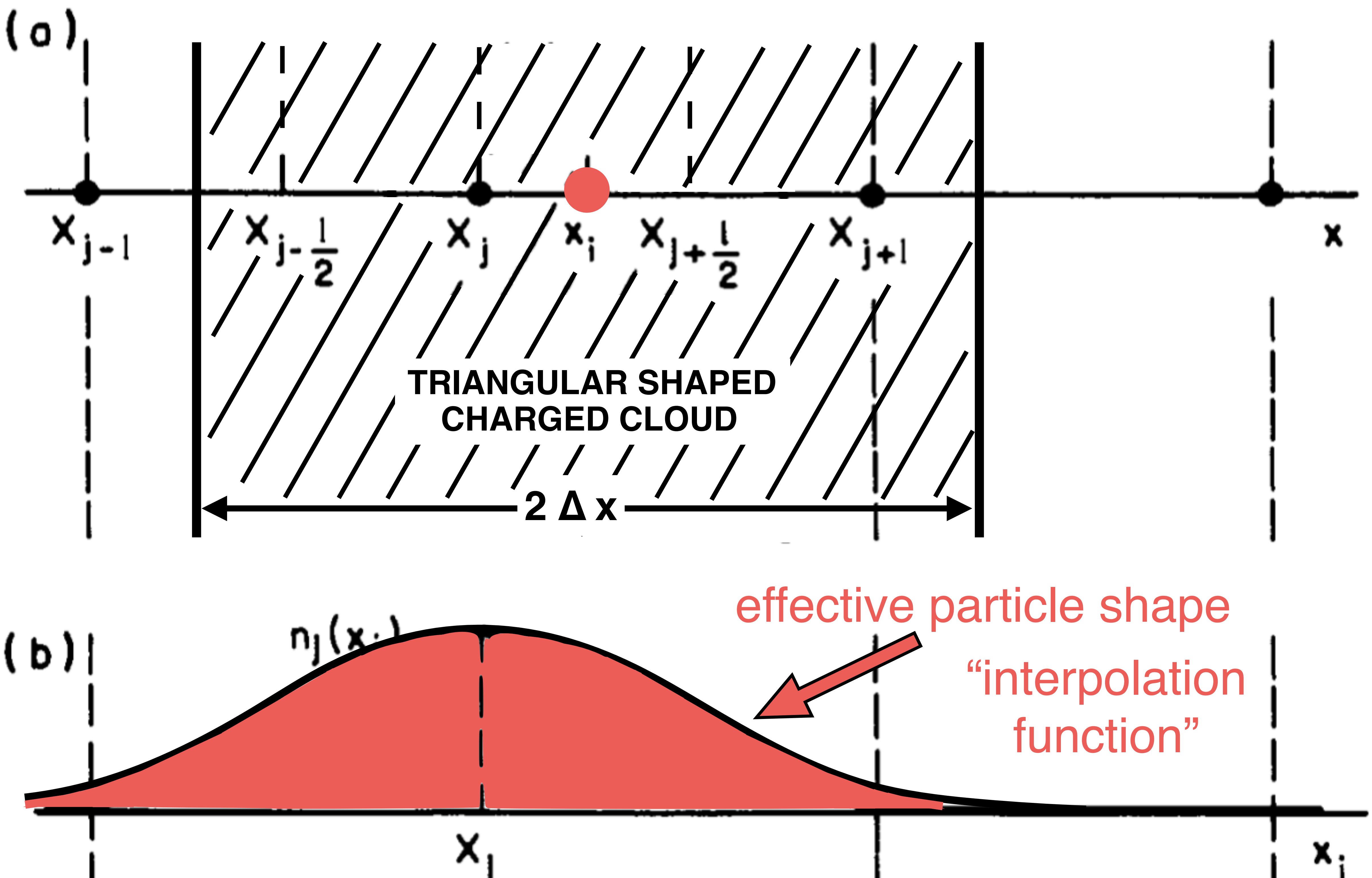
assigned to whatever cell contains particle
(bad: discontinuous forces)

1st-order particle weighting (cloud-in-cell; CIC)



assignment proportional to overlapping volume
(ok: continuous forces, discontinuous derivatives)

2nd-order particle weighting (triangular shaped cloud; TSC)



assignment proportional to overlapping volume
(good: continuous forces and first derivatives)

in principle, higher-order shape functions can be used, which result in better spatial filtering of high-frequency components; but these require a larger stencil, which means many more accesses of memory

> 2nd-order deposition rarely used

instead, spatial filtering performed to smooth moments

spectral code: trivially done in *k* space

grid code: done by “digital filtering” (Hamming 77)

replace ϕ_j with $\dfrac{W\phi_{j-1} + \phi_j + W\phi_{j+1}}{1 + 2W}$

cannot be done in place; Birdsall & Langdon, App C

Coulomb force between finite-size particles

Finite-size particles considerably reduce Coulomb interactions

inter-particle forces inside a cell are underestimated; collisions must be re-introduced for controlled dissipation (rarely done)

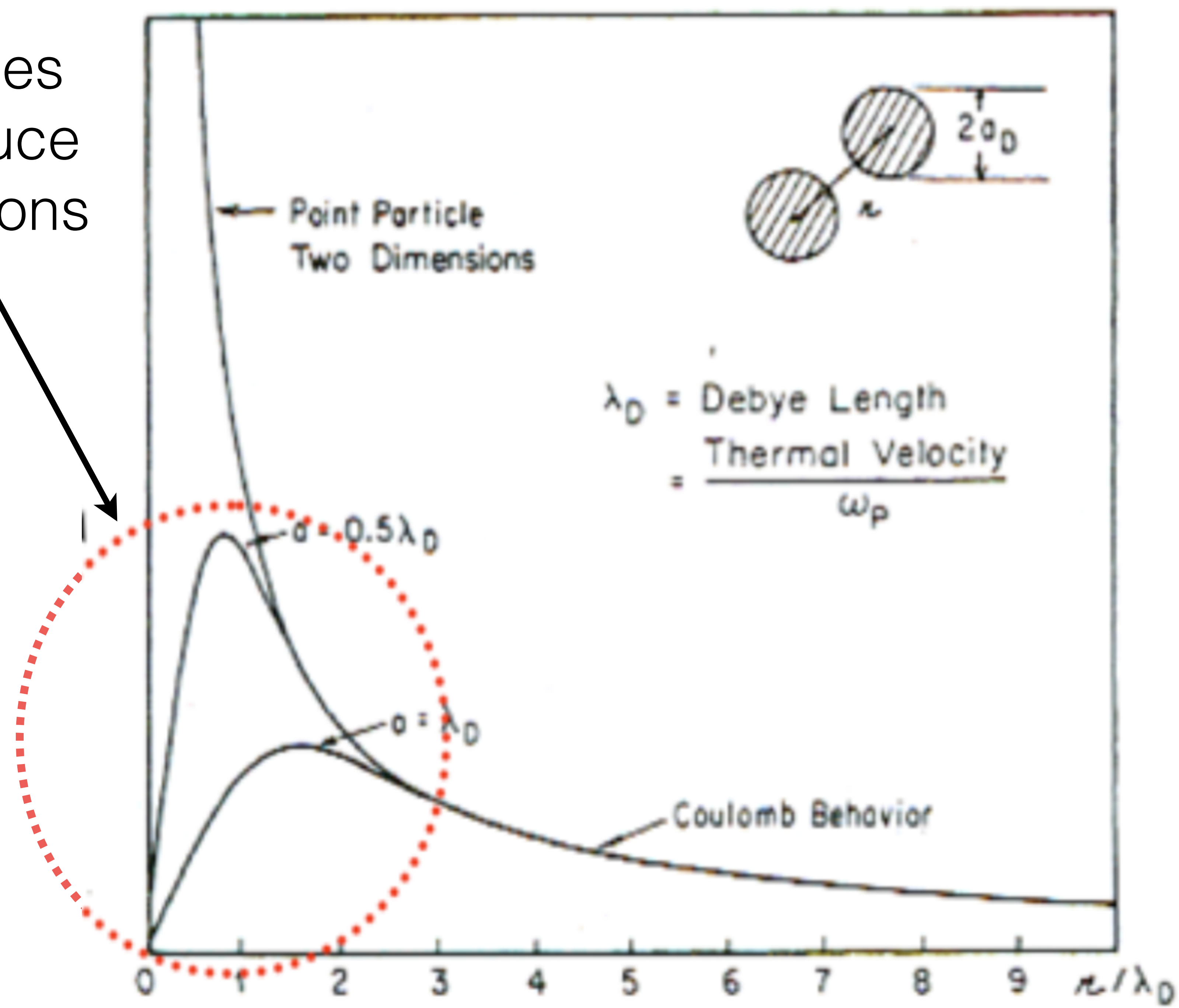
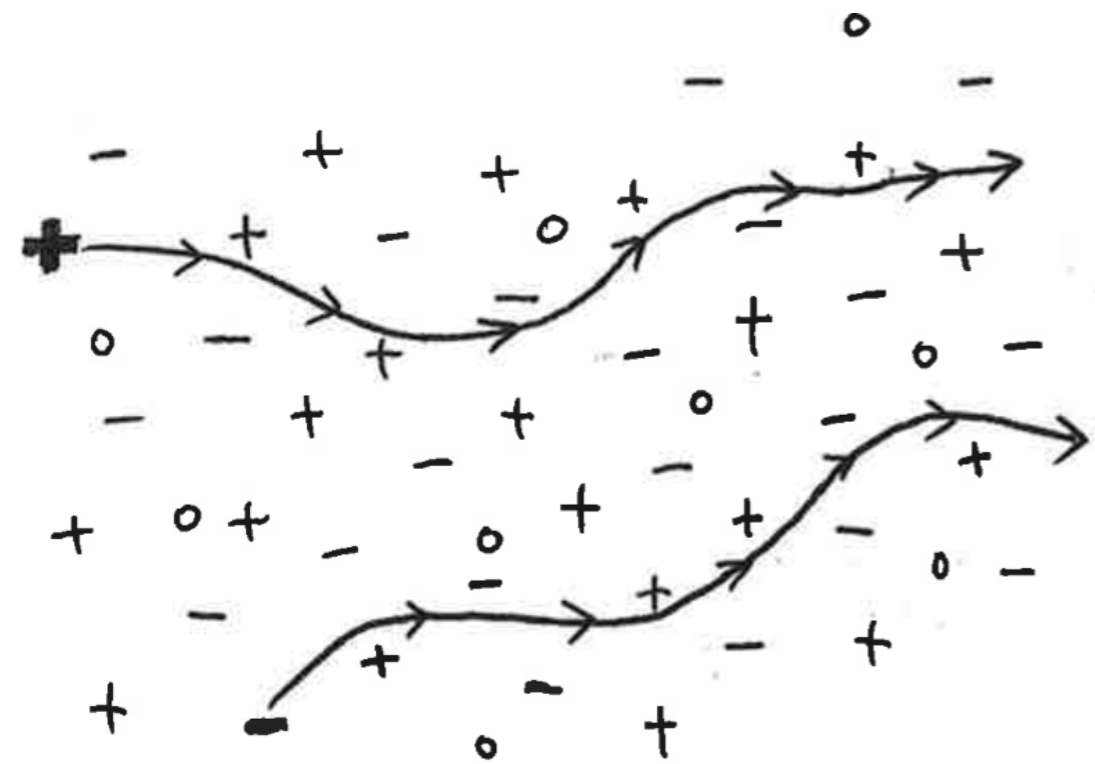


FIG. 2. Force law between finite-size particles in two dimensions for various sized particles. A Gaussian-shaped charge-density profile was used.

key idea:

smooth particle trajectories
are not due to small-angle
deflections mediated by a Debye
sphere of many charged particles



rather, relatively few near neighbors
produce weak interactions due to
overlapping shape functions

fewer particles acting
more weakly ~ $\Lambda \gg 1$

Step 3: Update Fields

evolution equations:

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

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\boldsymbol{\nabla} \times \boldsymbol{B} - 4\pi \sum_{\alpha} q_{\alpha} \sum_{i=1}^{N_{\alpha}} \boldsymbol{V}_{\alpha i}\, S(\boldsymbol{r} - \boldsymbol{R}_{\alpha i})$$

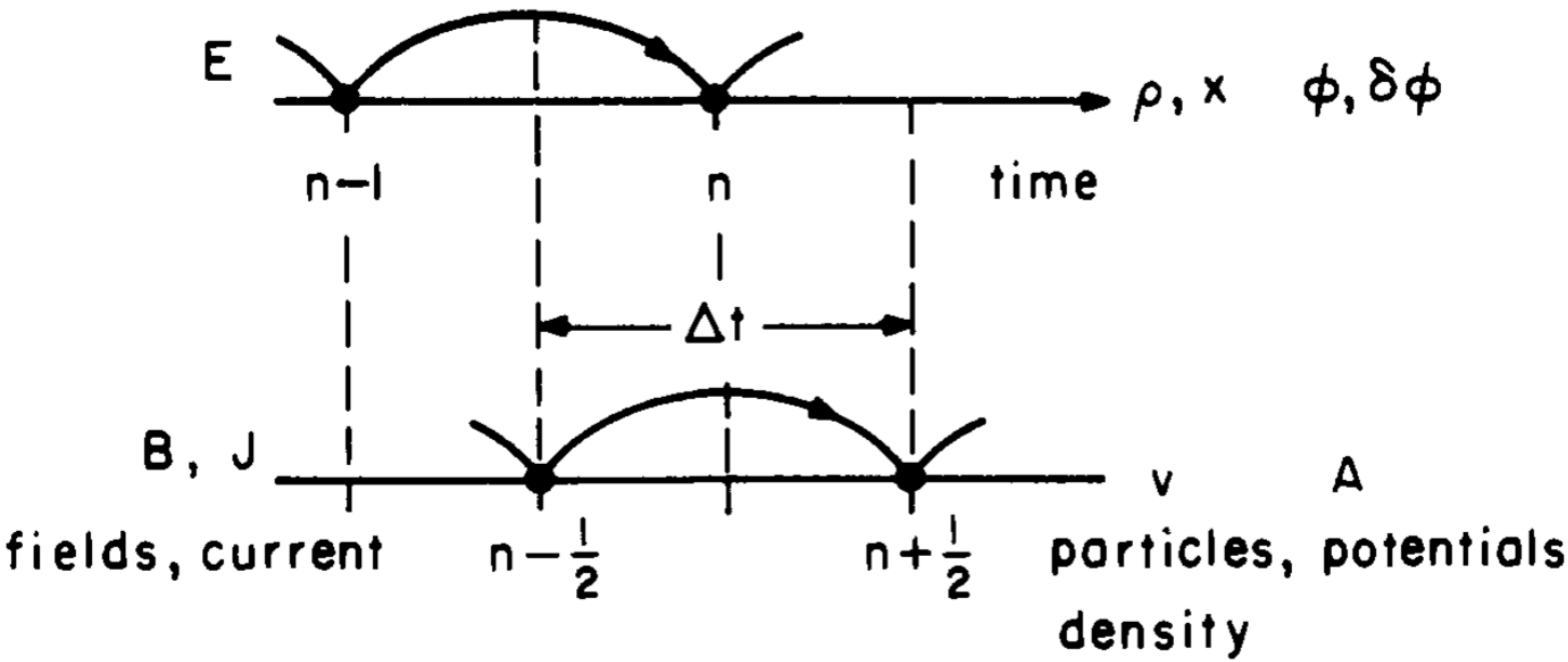
constraints:

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{E} = 4\pi \sum_{\alpha} q_{\alpha} \sum_{i=1}^{N_{\alpha}} S(\boldsymbol{r} - \boldsymbol{R}_{\alpha i})$$

broken by truncation error if you're not careful!

symmetry of Maxwell's equations suggests leapfrog:

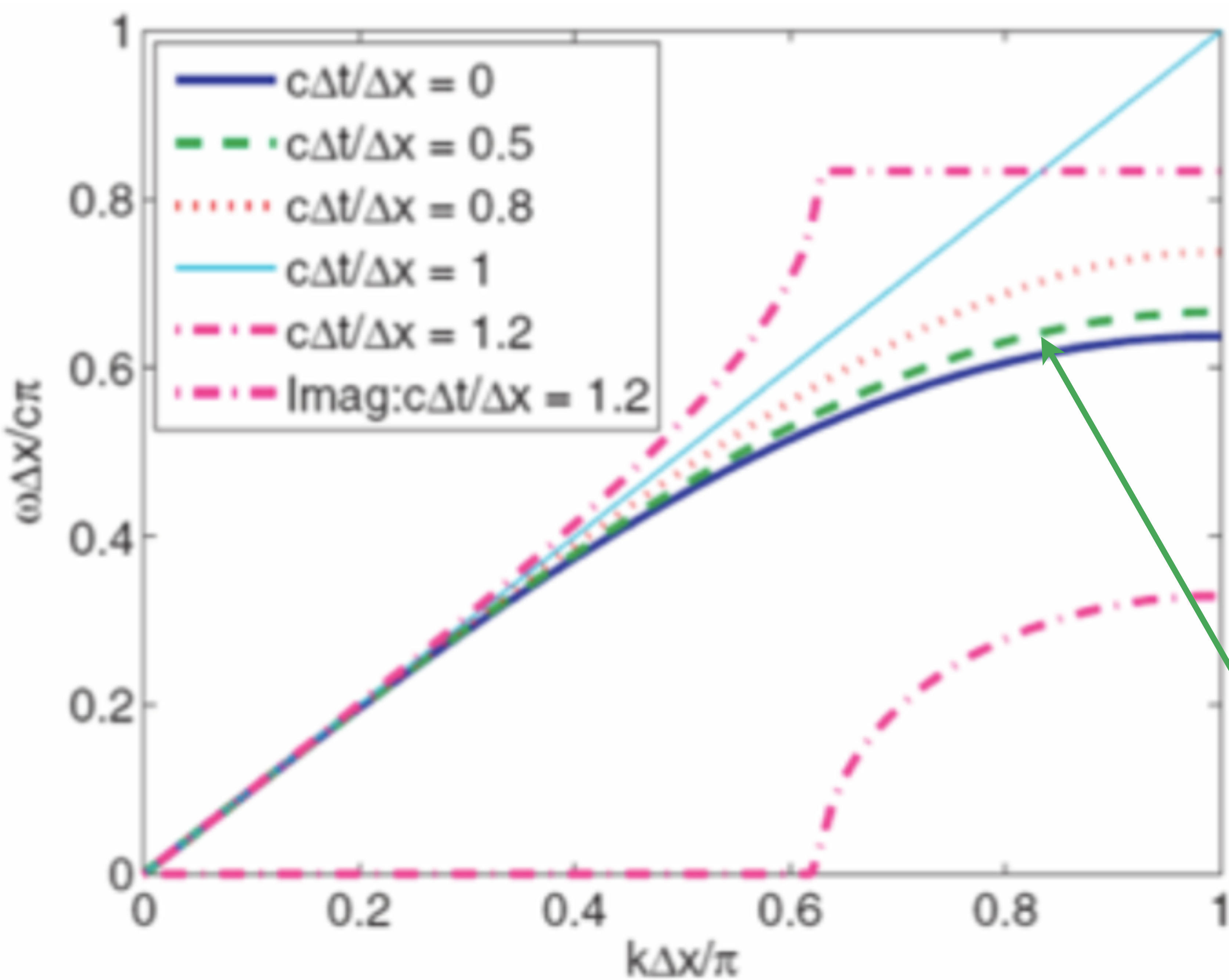


$$\frac{\boldsymbol{E}^n - \boldsymbol{E}^{n-1}}{\Delta t} = \boldsymbol{\nabla} \times \boldsymbol{B}^{n-1/2} - \boldsymbol{J}^{n-1/2}$$

$$\frac{\boldsymbol{B}^{n+1/2} - \boldsymbol{B}^{n-1/2}}{\Delta t} = -\boldsymbol{\nabla} \times \boldsymbol{E}^n$$

Stability Condition

$$\Delta t \leq \frac{1}{c}\left(\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}+\frac{1}{(\Delta z)^2}\right)^{-1/2}$$



exact

unstable!

stable, but exhibits phase error at small wavelengths; causes numerical Cerenkov radiation

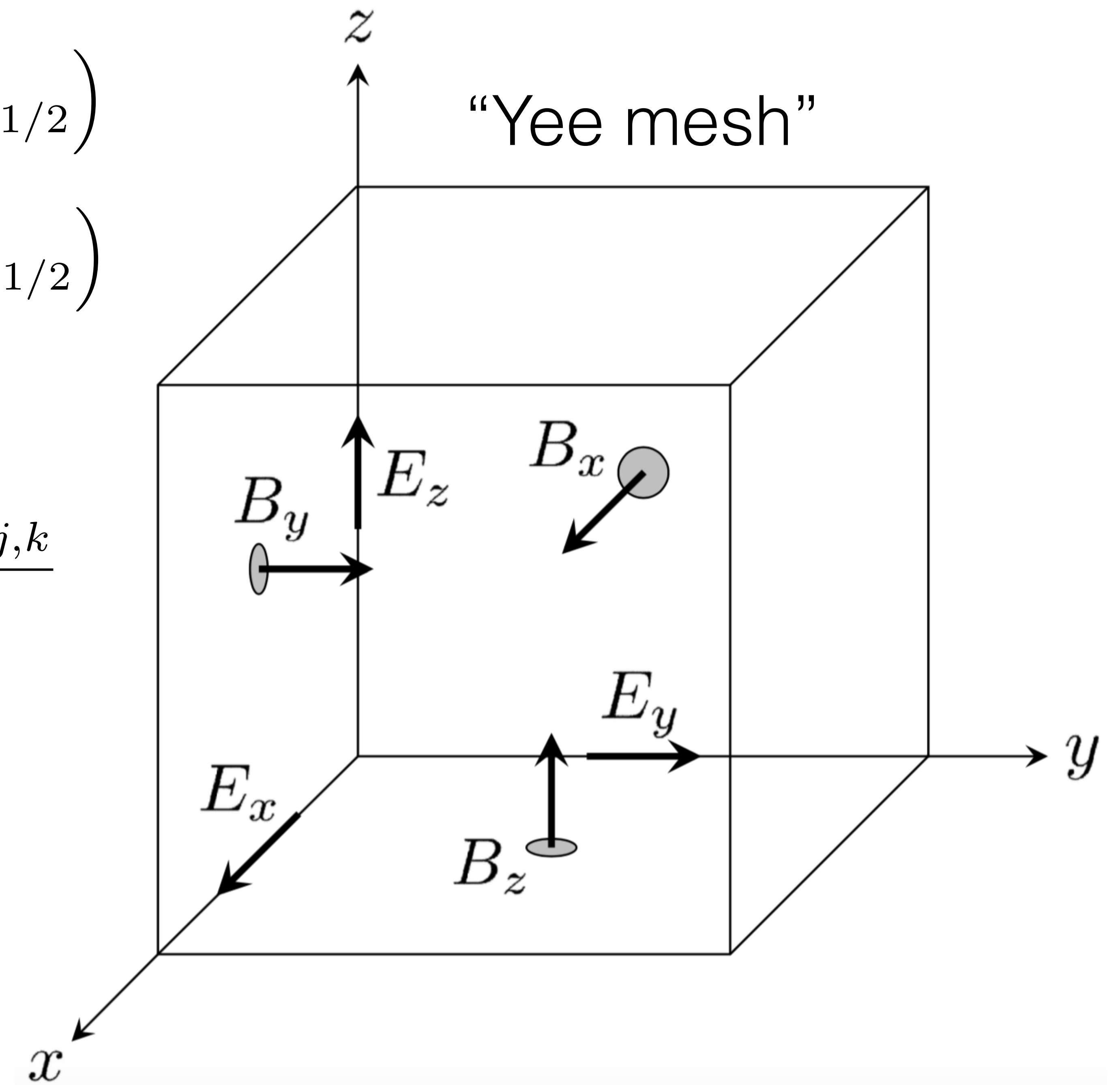
but how should we enforce field constraints?

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0$$

$$\begin{aligned} B^{n+1/2}_{z;i,j,k-1/2} &= B^{n-1/2}_{z;i,j,k-1/2} \\ &- \frac{\Delta t}{\Delta x}\left(E^n_{y;i+1/2,j,k-1/2} - E^n_{y;i-1/2,j,k-1/2}\right) \\ &+ \frac{\Delta t}{\Delta y}\left(E^n_{x;i,j+1/2,k-1/2} - E^n_{x;i,j-1/2,k-1/2}\right) \end{aligned}$$

$$\begin{aligned} (\boldsymbol{\nabla} \cdot \boldsymbol{B})^{n+1/2}_{i,j,k} &= \frac{B^{n+1/2}_{x;i+1/2,j,k} - B^{n+1/2}_{x;i-1/2,j,k}}{\Delta x} \\ &+ \frac{B^{n+1/2}_{y;i,j+1/2,k} - B^{n+1/2}_{y;i,j-1/2,k}}{\Delta y} \\ &+ \frac{B^{n+1/2}_{z;i,j,k+1/2} - B^{n+1/2}_{z;i,j,k-1/2}}{\Delta z} = 0 \end{aligned}$$

“Yee mesh”



but how should we enforce field constraints?

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0$$

NB: (at least) some hybrid-kinetic codes don't do this. Instead, they either use:

(a) vector potential $\boldsymbol{B} = \boldsymbol{\nabla} \times \boldsymbol{A}$, as long as $\boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \times$ vanishes identically … but it's a bit cumbersome and there's gauge freedom…;

(b) divergence cleaning… but it can get expensive and is ideologically disturbing; or

(c) ignorance, which is inexpensive but can cost you in unknown ways.

but how should we enforce field constraints?

$$\boldsymbol{\nabla} \cdot \boldsymbol{E} = 4\pi \sum_{\alpha} q_\alpha \sum_{i=1}^{N_\alpha} S(\boldsymbol{r} - \boldsymbol{R}_{\alpha i})$$

total charge is conserved, but not necessarily that deposited on grid

options:

(a) correct the E field *ex post facto* to satisfy Poisson (expensive, but might be easiest on non-orthogonal grids);

Section 15-6 of Birdsall & Langdon (also Boris 1970):

$$\text{correction } \boldsymbol{E}' = \boldsymbol{E} - \boldsymbol{\nabla}\delta\varphi \text{ such that } \boldsymbol{\nabla} \cdot \boldsymbol{E}' = 4\pi\rho_q$$

$$\Longrightarrow \boldsymbol{\nabla} \cdot (\boldsymbol{E} - \boldsymbol{\nabla}\delta\varphi) = 4\pi\rho_q \Longrightarrow \nabla^2\delta\varphi = \boldsymbol{\nabla} \cdot \boldsymbol{E} - 4\pi\rho_q$$

but how should we enforce field constraints?

$$\boldsymbol{\nabla}\cdot\boldsymbol{E} = 4\pi\sum_{\alpha} q_\alpha \sum_{i=1}^{N_\alpha} S(\boldsymbol{r}-\boldsymbol{R}_{\alpha i})$$

total charge is conserved, but not necessarily that deposited on grid

options:

(a) correct the E field *ex post facto* to satisfy Poisson (expensive, but might be easiest on non-orthogonal grids);

(b) hyperbolic divergence cleaning (Marder 1987); or

(c) charge-conserving deposition.

charge-conserving deposition

$$\frac{\partial \rho_q}{\partial t} = -\boldsymbol{\nabla} \cdot \boldsymbol{j}$$

If currents are handled with care,
Poisson equation is just an initial condition

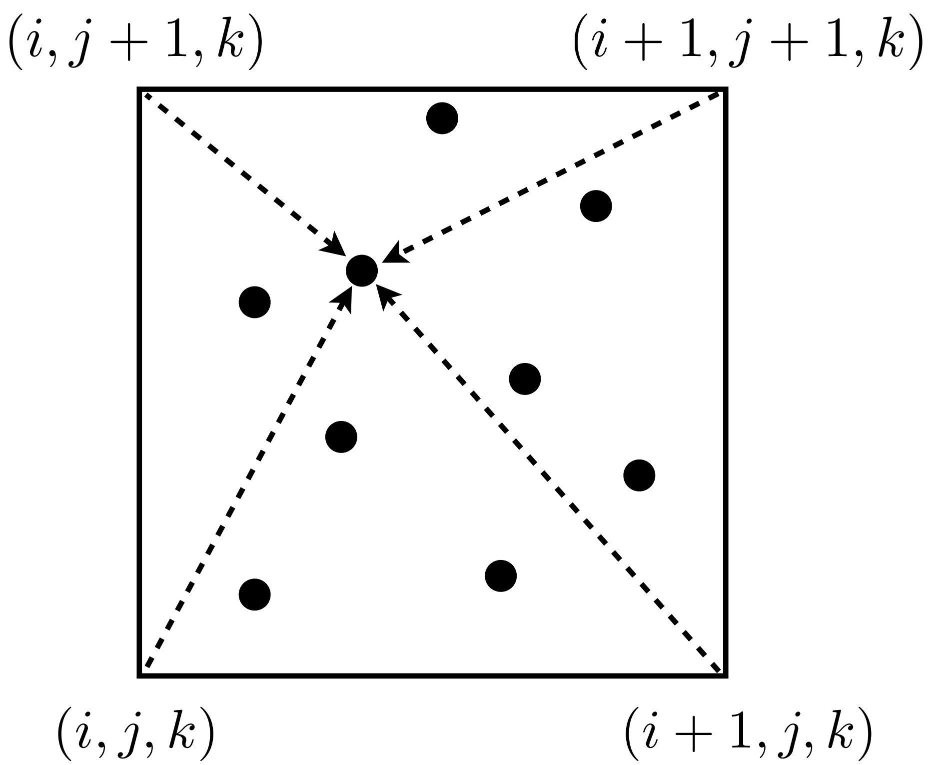
(the same way that div-B is just an initial condition
if electric fields are handled with care)

instead of volume weighting, count the
“volume current” through appropriate faces
(Villasenor & Buneman 1992)

can take as much time as particle mover (sometimes more)

Step 4: Interpolate Grid to Particles

interpolation to/from grid must be done in same way,
or else you get self-force



$$\boldsymbol{E}(\boldsymbol{R}_p) = \sum_{i,j,k} \boldsymbol{E}(\boldsymbol{r}_{i,j,k})\, S(\boldsymbol{r}_{i,j,k} - \boldsymbol{R}_p)$$

$$\boldsymbol{B}(\boldsymbol{R}_p) = \sum_{i,j,k} \boldsymbol{B}(\boldsymbol{r}_{i,j,k})\, S(\boldsymbol{r}_{i,j,k} - \boldsymbol{R}_p)$$

$$\sum_{i,j,k} \boldsymbol{E}(\boldsymbol{r}_{i,j,k}) \cdot \boldsymbol{B}(\boldsymbol{r}_{i,j,k})\, S(\boldsymbol{r}_{i,j,k} - \boldsymbol{R}_p) \stackrel{?}{=} \boldsymbol{E}(\boldsymbol{R}_p) \cdot \boldsymbol{B}(\boldsymbol{R}_p)$$

better be…

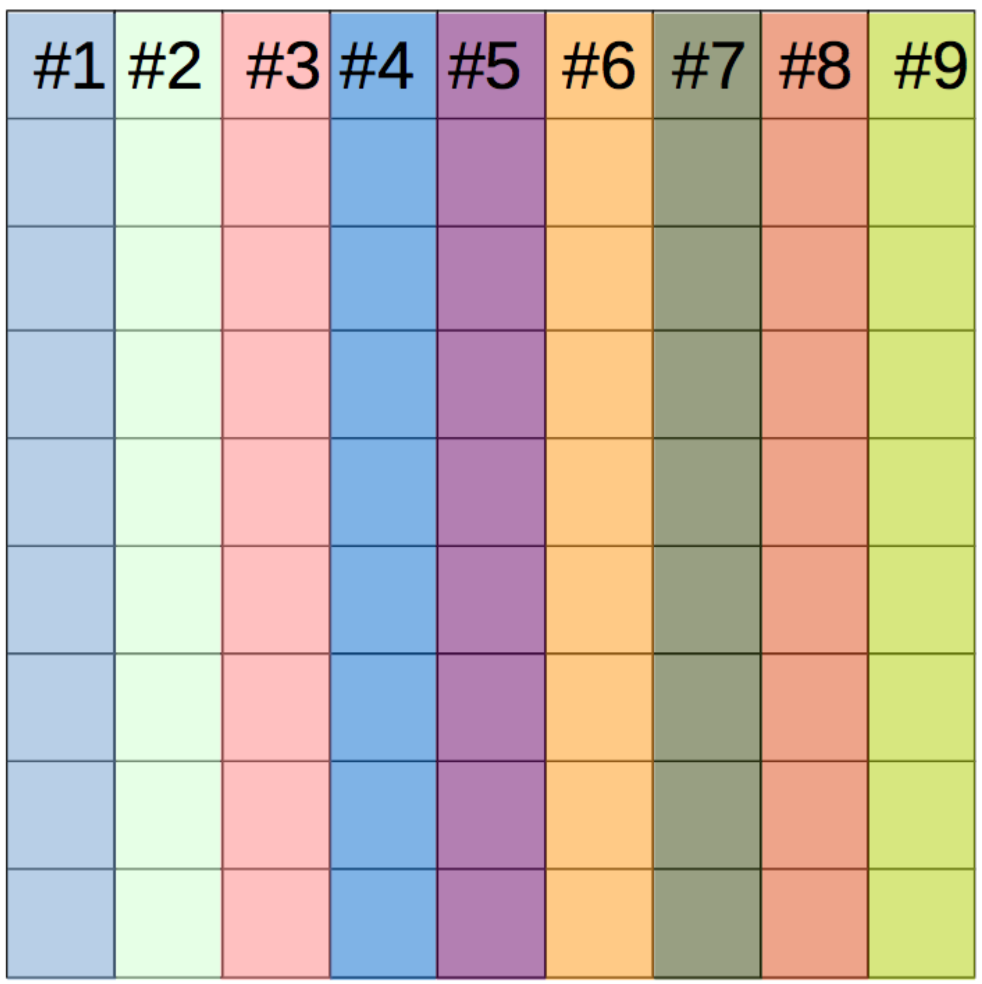
Parallelization: Domain decomposition

PIC code are really demanding in computing resources => **Need to parallelize the code!**

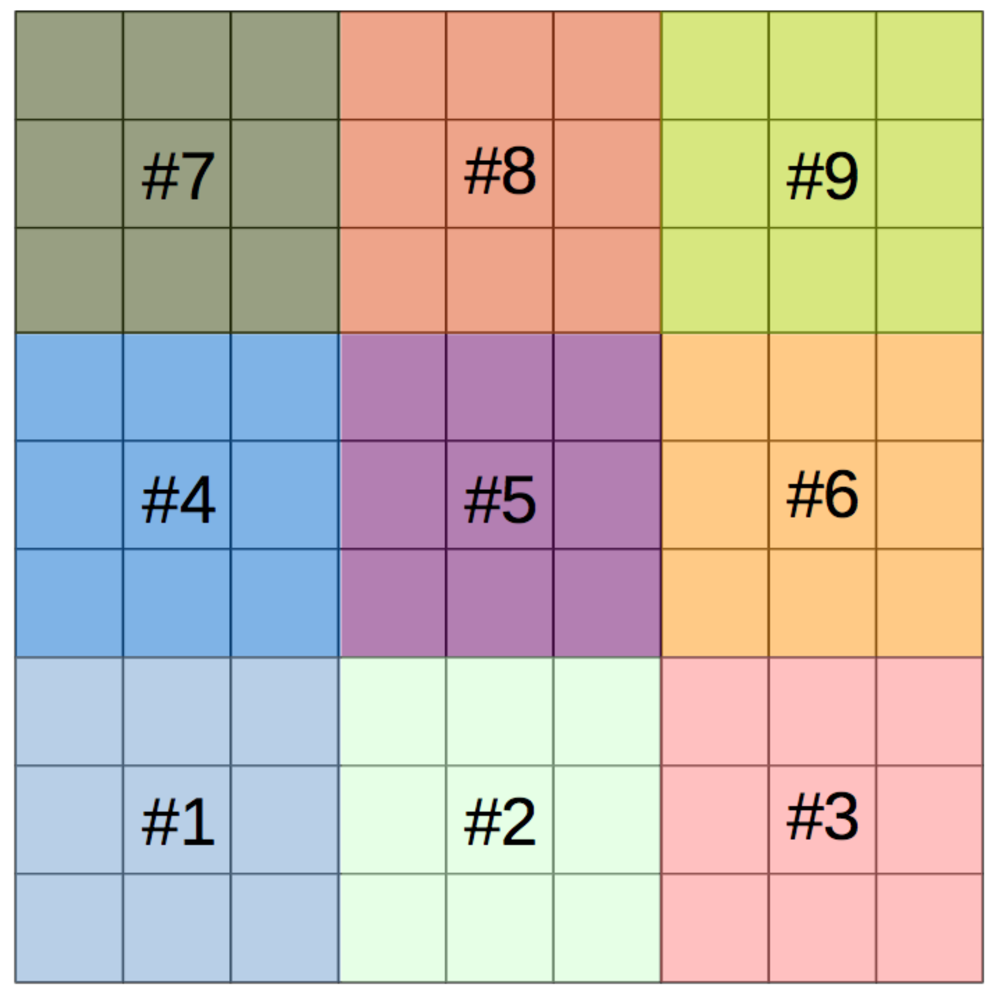
A common practice is to use the **Message Passing Interface (MPI)** library and the **domain decomposition technique.**

Example: Consider a 2D mesh 9x9 cells and 9 CPUs.

1D decomposition

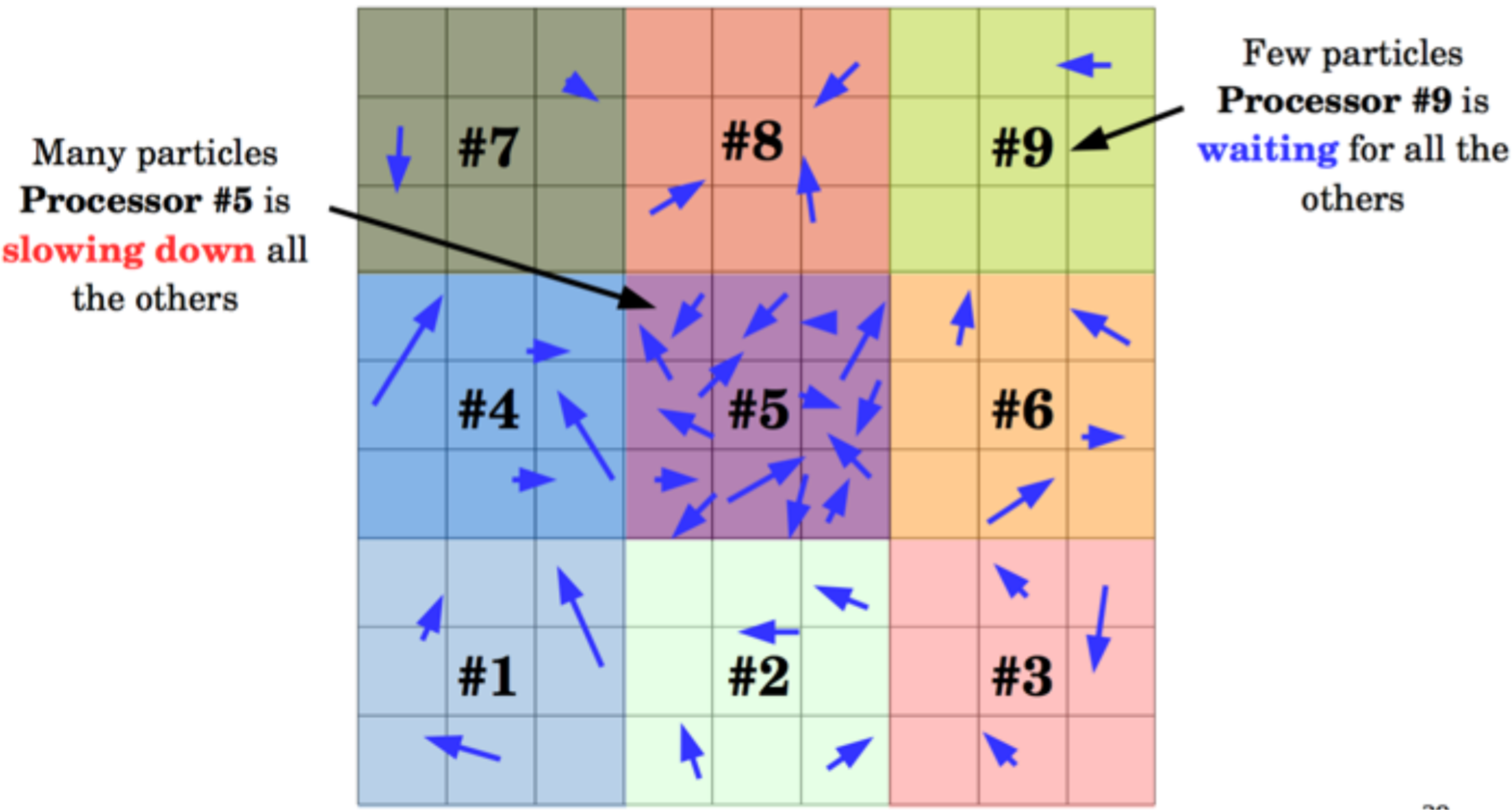


2D decomposition



slide from Benoît Cerutti

load balancing sometimes required



must think about how particle list is stored in memory

slide from Benoît Cerutti

https://tristan-mp.wikispaces.com

Anatoly Spitkovsky

2D/3D cartesian EM PIC code
various BCs; moving window
charge-conservative current deposition
fully parallelized domain decomposition

tristan-mp

- Wiki Home
- Recent Changes
- Pages and Files
- Members

Search

Home
User's Manual
- Code Features
- Input Structure
- User file
- Example: Weibel instability
- Example: Collisionless shock
- Compilation
- Running

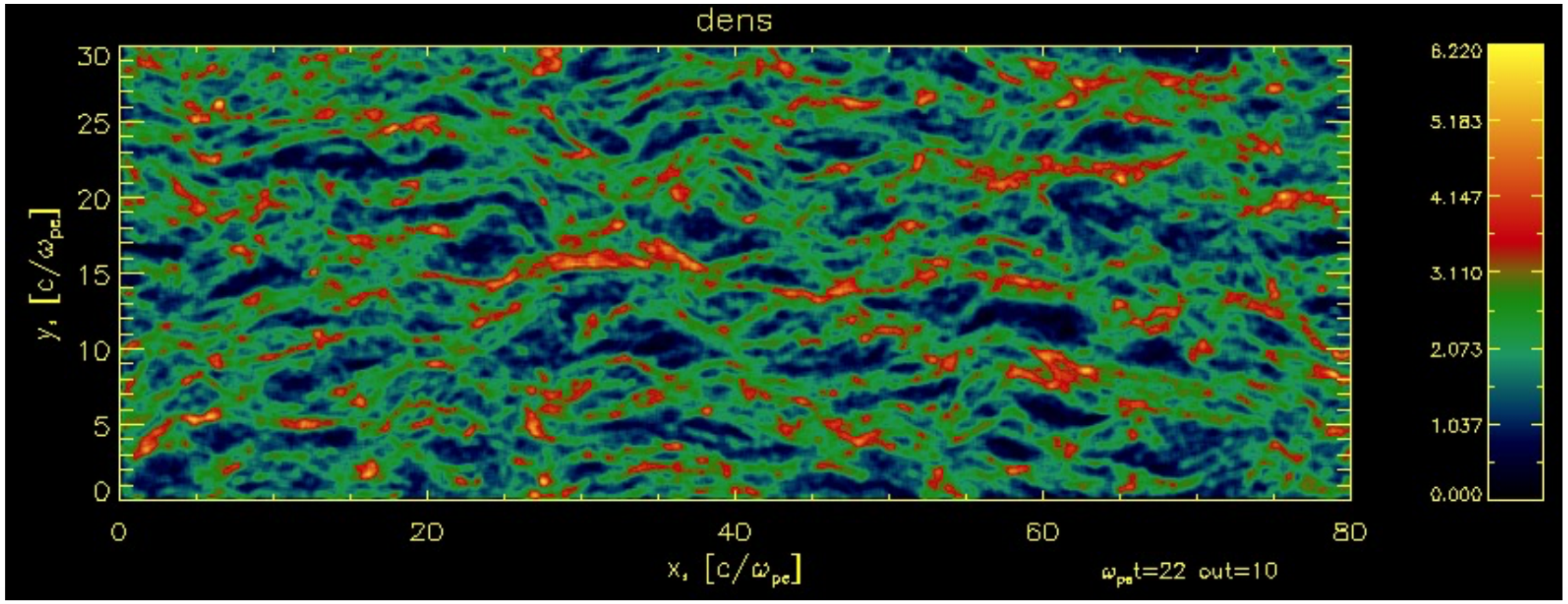
Developer's Manual

Weibel

Edit | 0 | 5 | ...

Weibel instability run

If you use one of the example input files provided with the current distribution of the code (input.weibel in the SampleInputs directory), you will generate the weibel instability. For this run, two (electron-positron) plasmas are distributed uniformly across the box, and are given a flow velocity in the x direction corresponding to a relativistic gamma factor of 15. One of the plasmas is flowing to the right, the other one is flowing to the left. After some iterations, a magnetic field should start to grow from noise. The total plasma density and the Bz component of the magnetic field should look like this (at time t=22 wpe-1):



http://benoit.cerutti.free.fr/Zeltron/

includes radiative losses
spherical Yee mesh (Cerutti et al. 2015, 2016)
non-Euclidean metric (Philippov et al. 2015)

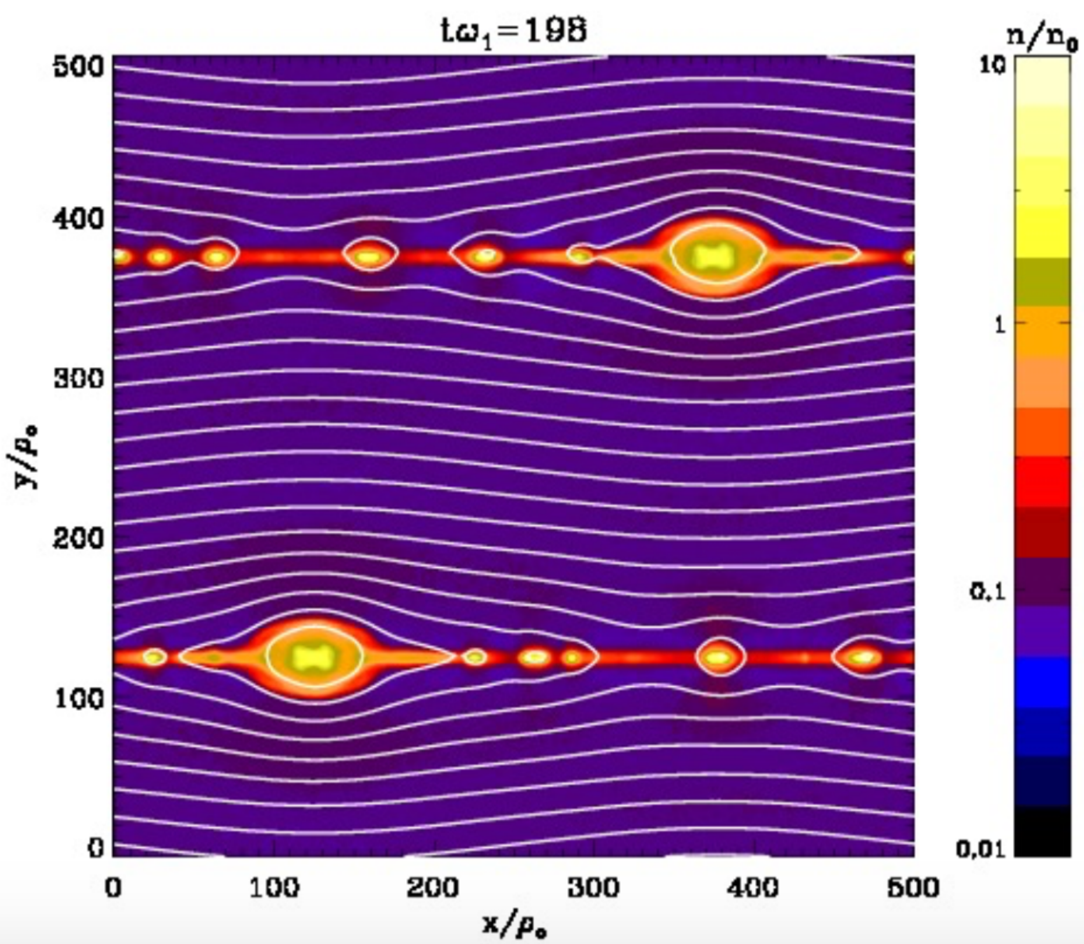
Benoît Cerutti

The Zeltron code

- Home
- Features
- Download
- Quickstart
- User guide
- Gallery
- Publications
- Developers
- Contact

Gallery

This page presents a few applications of the Zeltron code to astrophysical problems.



2D relativistic reconnection simulation

Electron-positron plasma density of a 2D simulation of relativistic reconnection with synchrotron radiation reaction force. White contours show the magnetic field lines. This simulation was performed to explain flashes of energetic gamma rays from the Crab Nebula (see Cerutti et al. 2013).

Play movie ▶

a menu

XOOPIC (2D RPIC, free unix version, Mac and Windows are paid through Tech-X);

VORPAL (1,2,3D RPIC, hybrid, sold by Tech-X)

TRISTAN (public serial version), 3D RPIC (also have 2D), becoming public now

OSIRIS (UCLA) 3D RPIC, mainly used for plasma accelerator research

Apar-T, Zeltron.

PIC-on-GPU — open source

LSP -- commercial PIC and hybrid code, used at national labs

VLPL -- laser-plasma code (Pukhov ~2000)

Reconnection research code (UMD, UDelaware)

Every national lab has PIC codes. (VPIC at Los Alamos)

All are tuned for different problems, and sometimes use different formulations (e.g. vector potential vs fields, etc). Direct comparison is rarely done.

Hybrid kinetics

often, c, λ_{D}, ω_{pe}, d_{e}, ρ_{e}, Ω_{e} is too much
and not even affordable without severe sacrifice

$$\cancel{\frac{\partial \boldsymbol{E}}{\partial t}} = c\boldsymbol{\nabla}\times\boldsymbol{B} - 4\pi\sum_{\alpha} q_\alpha \sum_{i=1}^{N_\alpha} \boldsymbol{V}_{\alpha i}\, S(\boldsymbol{r}-\boldsymbol{R}_{\alpha i})$$

$$\cancel{\boldsymbol{\nabla}\cdot\boldsymbol{E}} = 4\pi\sum_{\alpha} q_\alpha \sum_{i=1}^{N_\alpha} S(\boldsymbol{r}-\boldsymbol{R}_{\alpha i})$$

$$\cancel{m_{\mathrm{e}} n_{\mathrm{e}} \frac{\mathrm{d}\boldsymbol{u}_{\mathrm{e}}}{\mathrm{d}t}} = -\boldsymbol{\nabla}\cdot\mathsf{P}_{\mathrm{e}} - e n_{\mathrm{e}}\left(\boldsymbol{E} + \frac{1}{c}\boldsymbol{u}_{\mathrm{e}}\times\boldsymbol{B}\right) + \int \mathrm{d}\boldsymbol{v}\, m_{\mathrm{e}}\boldsymbol{v}\left(\frac{\partial f_{\mathrm{e}}}{\partial t}\right)_{\mathrm{coll}}$$

must assume something;
usually isotropic and:
isothermal or barotropic

must assume something;
usually Ohm's law:
$$\frac{m_{\mathrm{e}} n_{\mathrm{e}}}{\tau_{\mathrm{ei}}}(\boldsymbol{u}_{\mathrm{i}} - \boldsymbol{u}_{\mathrm{e}}) = e n_{\mathrm{e}}\, \eta \boldsymbol{j}$$

Hybrid kinetics

$$\frac{\partial f_\mathrm{i}}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} f_\mathrm{i} + \frac{Ze}{m_\mathrm{i}} \left(\boldsymbol{E} + \frac{1}{c} \boldsymbol{v} \times \boldsymbol{B} \right) \cdot \frac{\partial f_\mathrm{i}}{\partial \boldsymbol{v}} = \left(\frac{\partial f_\mathrm{i}}{\partial t} \right)_\mathrm{coll}$$

$$\boldsymbol{E} = -\frac{1}{e n_\mathrm{e}} \boldsymbol{\nabla} p_\mathrm{e} - \frac{1}{c} \boldsymbol{u}_\mathrm{e} \times \boldsymbol{B} + \eta \boldsymbol{j}$$

$$= \boldsymbol{u}_\mathrm{i} - \frac{\boldsymbol{j}}{Zen_\mathrm{i}}$$

$$= Zn_\mathrm{i}$$

$$\boldsymbol{j} = \frac{c}{4\pi} \boldsymbol{\nabla} \times \boldsymbol{B}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -c \boldsymbol{\nabla} \times \boldsymbol{E} \qquad \boldsymbol{\nabla} \cdot \boldsymbol{B} = 0$$

difficulty: E is now a quasi-neutrality constraint

Byers et al. 1978; Harned 1982; Hewett & Nielson 1978

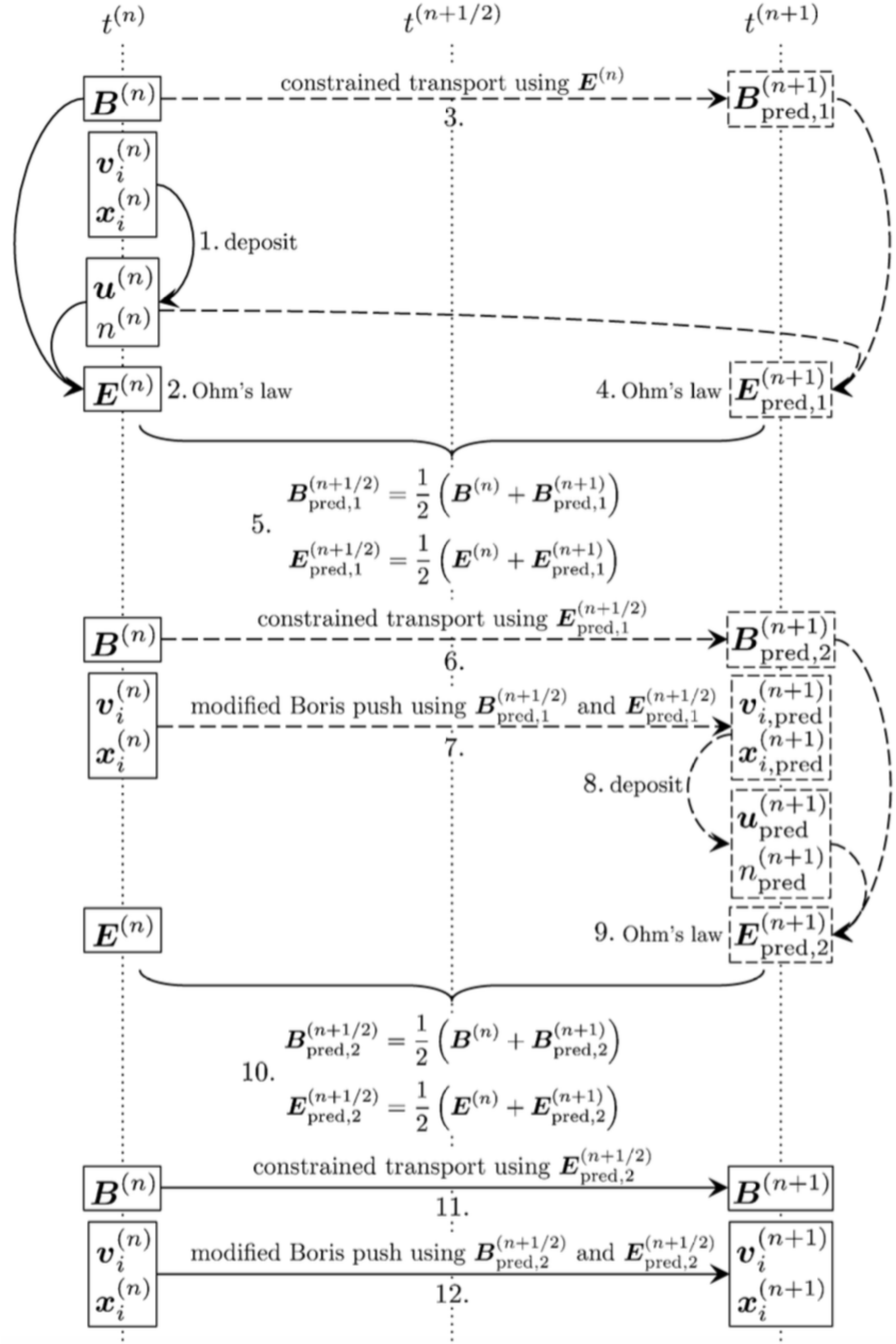
usually some kind of predictor-corrector method is used
(or worse)

Lipatov book

Winske et al. (2003) review

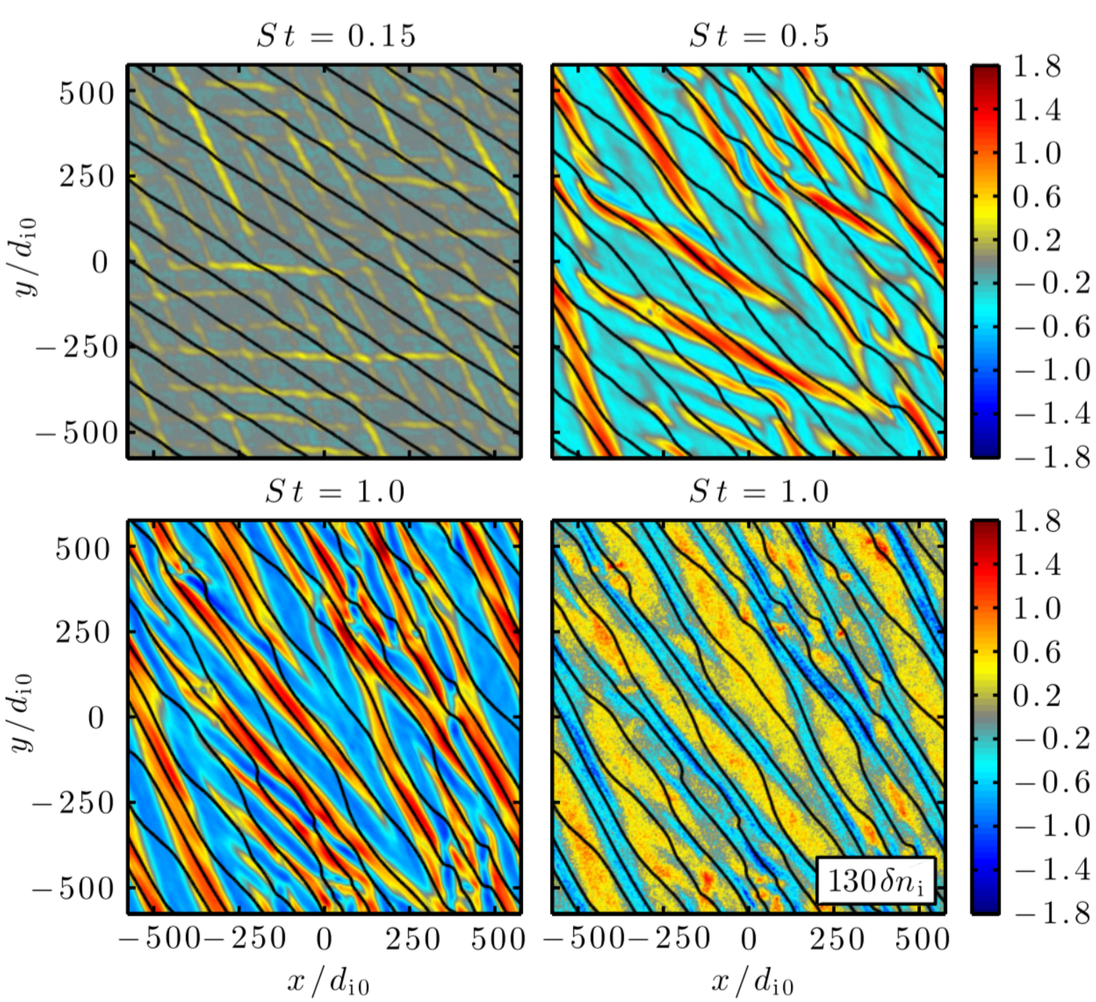
Kunz et al. (2014) Pegasus paper

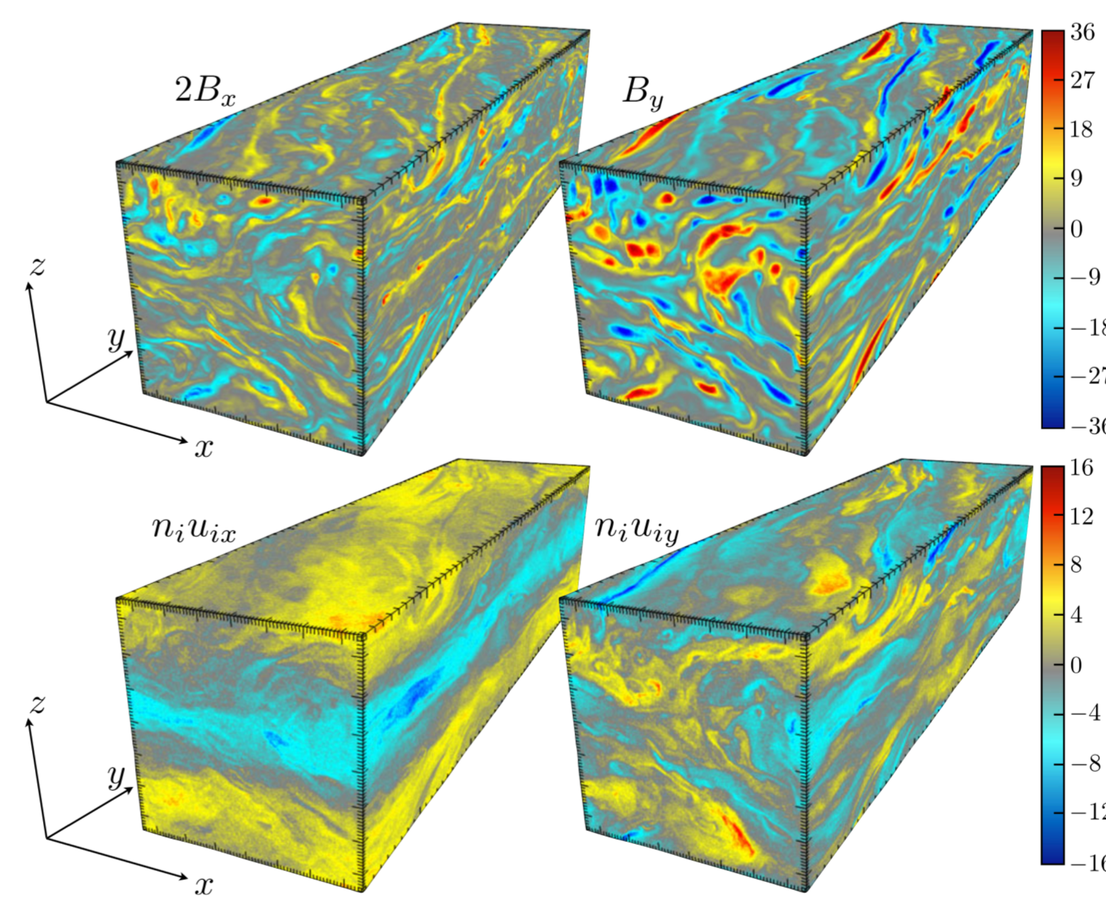
can solve either by continuum or PIC methods



application to:

- **non-relativistic shocks** (Gargate & Spitkovsky 2011, Caprioli & Spitkovsky 2014)
- **firehose/mirror instabilities**
  (Hellinger, Matteini, Trávnácek; Kunz, Stone & Schekochihin 2014)
- **kinetic magnetorotational instability** (Kunz, Stone & Quataert 2017)
- **Alfvén-wave propagation in high-ß plasma** (Squire, Kunz, Quataert, Schekochihin)
- **solar-wind turbulence**
  (Califano, Cerri, Fraci, Hellinger, Matteini, Servidio, Valentini, …)
- **reconnection** (Burgess, Cerri & Califano 2017, …)
- **near-Earth space physics** (Karimabadi, Lin, Quest, Omidi, Swift, Winske, …)

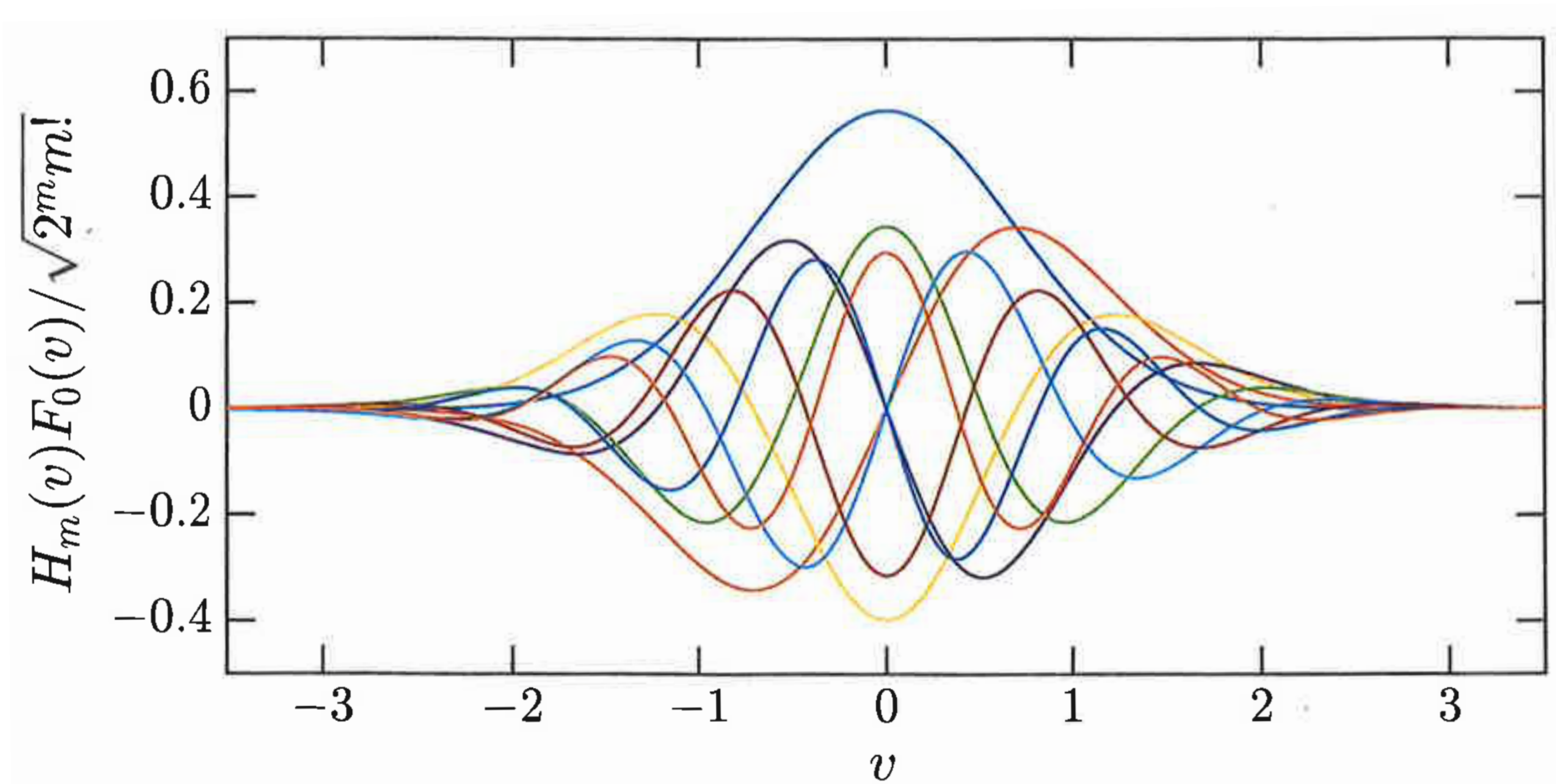




spectral representations of velocity space

$$H_m(v_\parallel) \doteq (-1)^m \, \mathrm{e}^{v^2} \frac{\mathrm{d}^m}{\mathrm{d}v^m} \mathrm{e}^{-v^2} \qquad \int \mathrm{d}v_\parallel \, \frac{H_m(v_\parallel) H_{m'}(v_\parallel)}{2^m m!} F_0(v_\parallel) = \delta_{mm'}$$

$$\delta f(v_\parallel) = \sum_{m=0}^{\infty} \frac{H_m(v_\parallel) F_0(v_\parallel)}{\sqrt{2^m m!}} \, \delta f_m, \quad \delta f_m = \int \mathrm{d}v_\parallel \, \frac{H_m(v_\parallel)}{\sqrt{2^m m!}} \, \delta f(v_\parallel)$$



spectral representations of velocity space

$$\frac{\partial \delta f_0}{\partial t} + \frac{\partial}{\partial z}\frac{\delta f_1}{\sqrt{2}} = \mathrm{source}$$

$$\frac{\partial \delta f_1}{\partial t} + \frac{\partial}{\partial z}\left(\delta f_2 + \frac{1+\alpha}{\sqrt{2}}\delta f_0\right) = \mathrm{source}$$

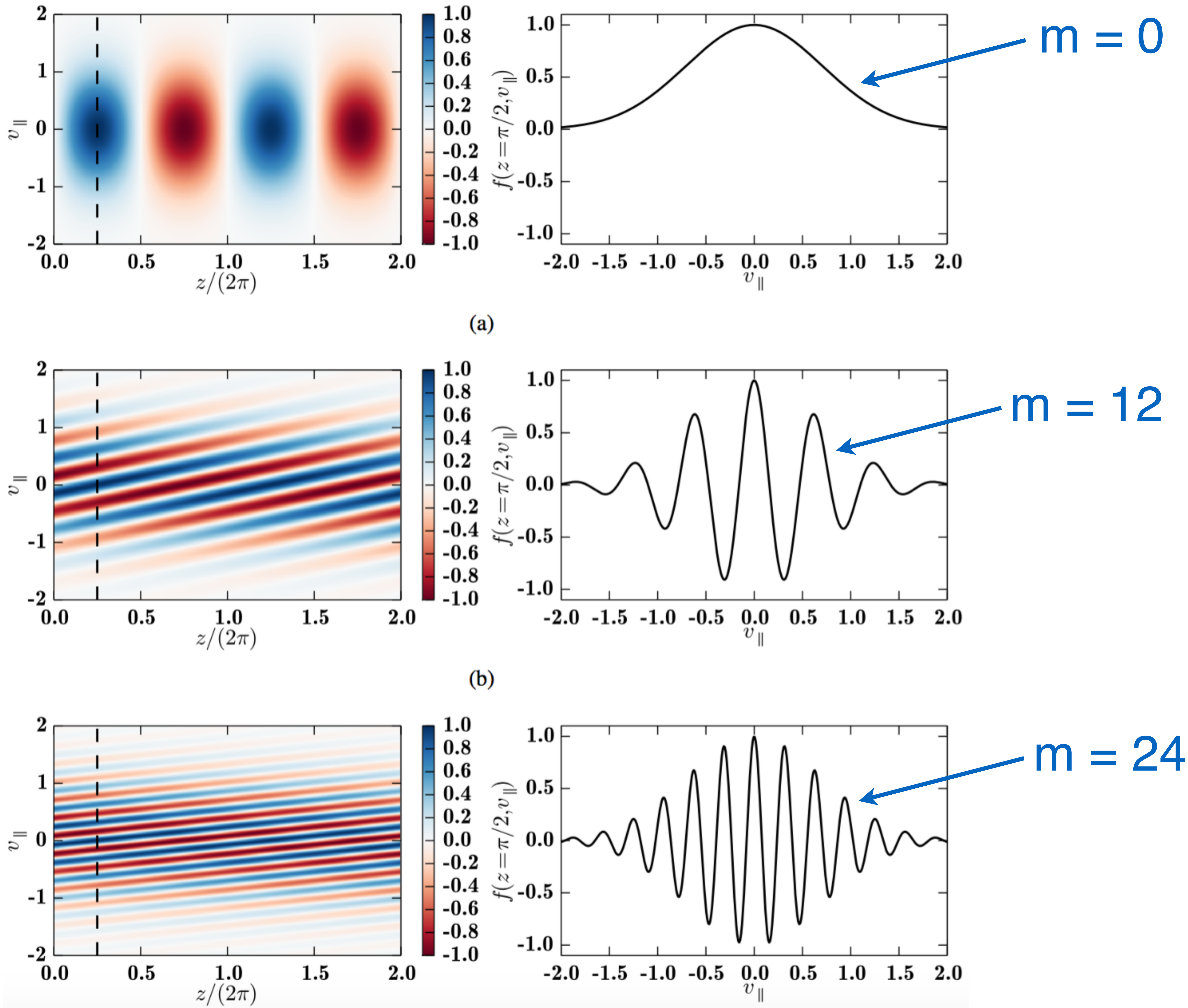
$$\vdots$$

$$\frac{\partial \delta f_m}{\partial t} + \frac{\partial}{\partial z}\underbrace{\left(\sqrt{\frac{m+1}{2}}\delta f_{m+1} + \sqrt{\frac{m}{2}}\delta f_{m-1}\right)}_{\text{phase mixing!}} = -\nu m \delta f_m, \quad m \geq 2$$

see Kanekar, Schekochihin, Dorland & Loureiro (2014) for more

SpectroGK: J. Parker & Dellar (2015); J. Parker thesis (arXiv:1603.04727)

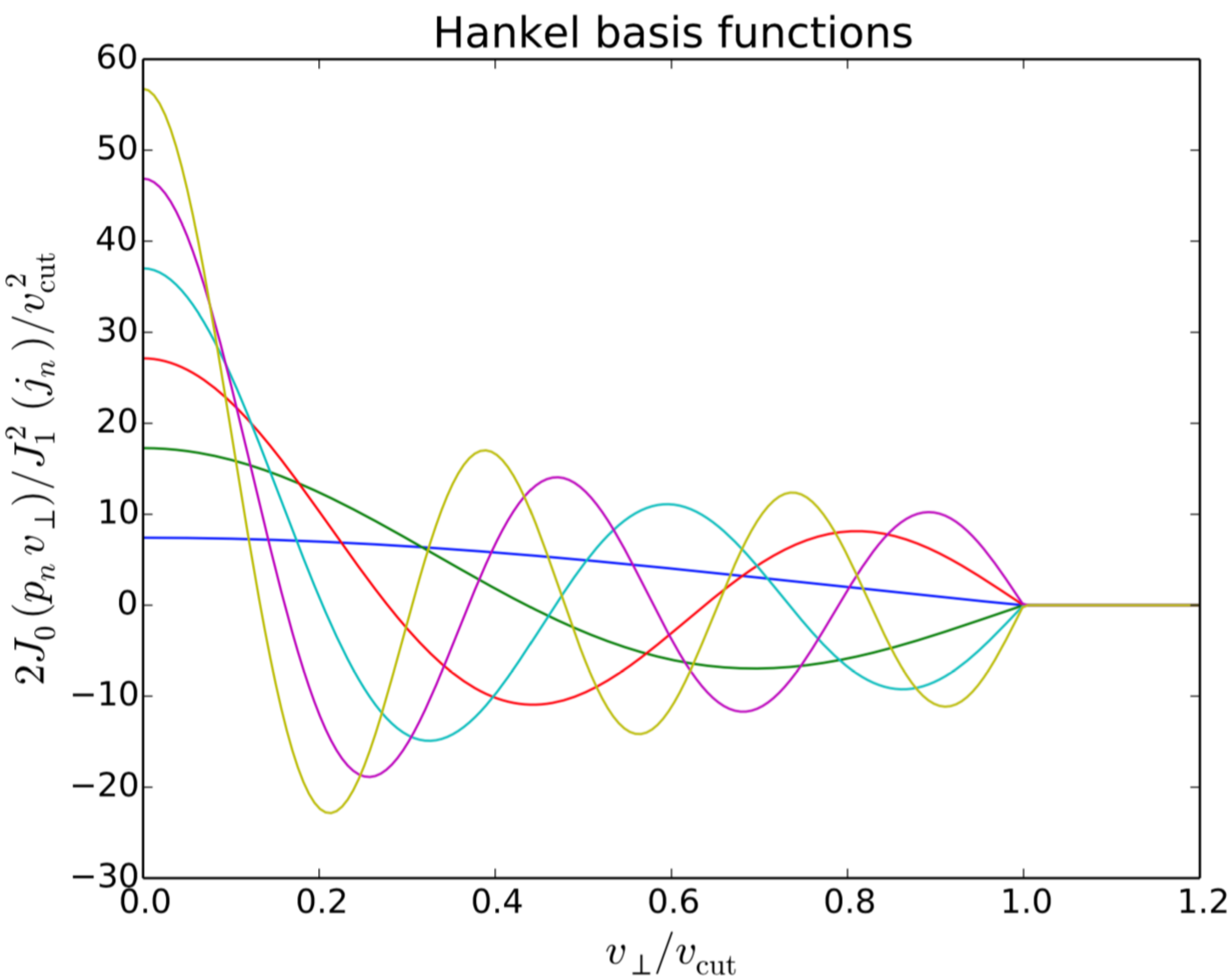
Viriato: Loureiro, Dorland et al. (2016)



J. Parker thesis (arXiv:1603.04727)

spectral representations of velocity space

can also spectral-ize perpendicular velocity space using Hankel transform:



$$\delta f(v_\perp) = \int_0^\infty \mathrm{d}p\, p J_0(p v_\perp) \delta f(p)$$

$$\int_0^\infty \mathrm{d}v_\perp v_\perp\, p J_0(p v_\perp) J_0(q v_\perp) = \delta(p-q)$$

Plunk, Dorland, Parker, Tatsuno,…

nonlinear perpendicular phase mixing

$$\frac{\partial \bar{g}_{\boldsymbol{k}m}(p)}{\partial t} + i v_{\mathrm{th}i} k_\parallel \left(\sqrt{\frac{m+1}{2}} \bar{g}_{\boldsymbol{k},m+1}(p) + \sqrt{\frac{m}{2}} \bar{g}_{\boldsymbol{k},m-1}(p) \right)$$
$$+ i \frac{q_s}{\sqrt{m_s T_s}} \frac{k_\parallel}{\sqrt{2}} \delta_{m1} \frac{1}{2\pi} \exp\left(-(\rho_i k_\perp)^2/2\right) I_0\left((\rho_i k_\perp)^2/2\right) \varphi_{\boldsymbol{k}} + \int_0^\infty \mathrm{d}v_\perp\, v_\perp J_0(p v_\perp) \left\{ \langle \varphi \rangle_{\boldsymbol{R}_i}, g_m(\boldsymbol{R}_i, v_\perp) \right\}_{\boldsymbol{k}} = 0,$$

Future for Vlasov-Maxwell and PIC codes

- Exploit modern computing architectures:
  leverage long vector lengths, port to Xeon Phi and KNL
- Speed up algorithms for particle deposition and interpolation
- Improve data locality: array of structures vs structure of arrays
- Build efficient non-orthogonal meshes
- Implement AMR for velocity-space grid
- Discontinuous Galerkin methods (Hammett, Hakim, TenBarge, Juno)
- Improved electron physics in Hybrid (must enforce quasi-neutrality…)

Some gyrokinetic codes of interest

PIC:

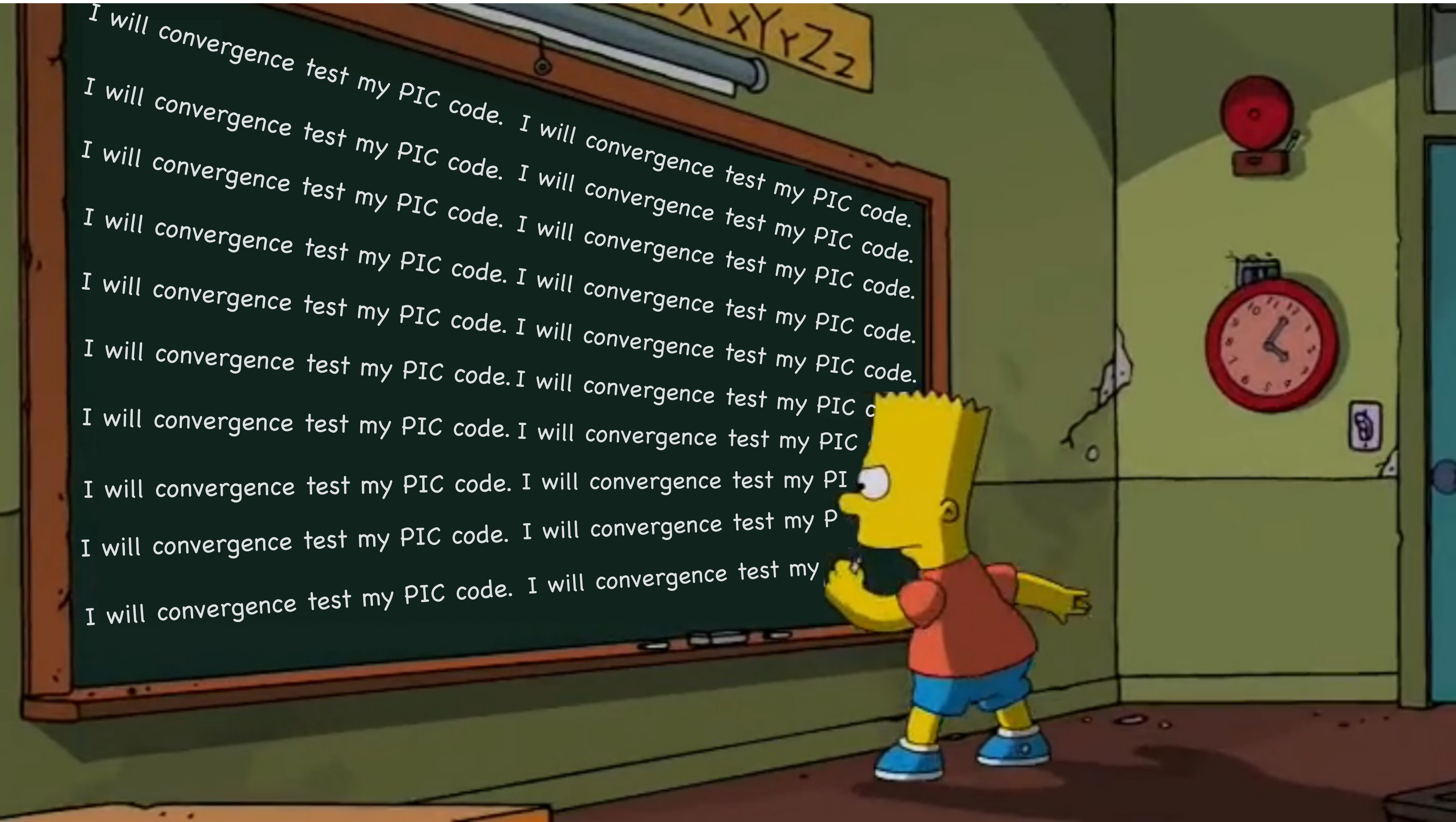
GTS (PPPL), GTC (UCI & PPPL),
GEM (Colorado), GT3D (JAEA, Japan),
ORB5 (CRPP, Switzerland)

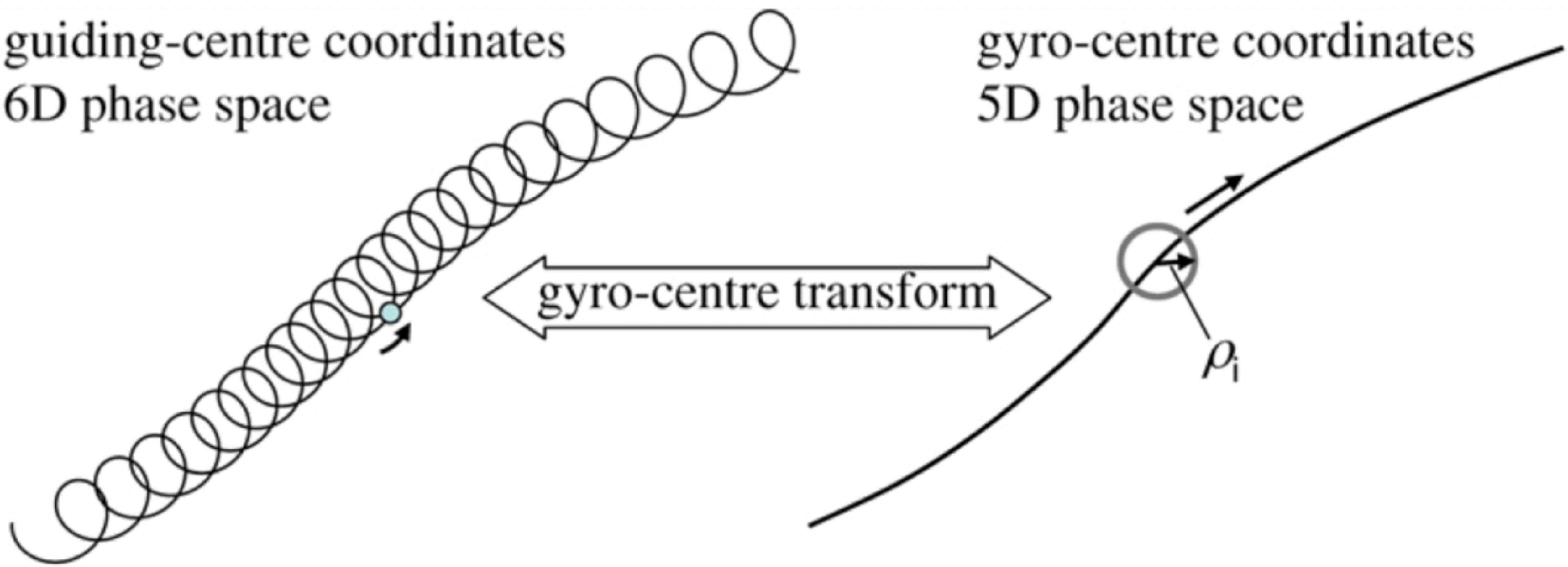
continuum:

GS2 (UMD), GENE (IPP, Germany),
GYRO (GA), GKV (NIFS, Japan),
GYSELA (CEA, France),
Astro-GK (UMD, Iowa), GKELL (PPPL)

thank you

with special thanks to:
Jim Stone, Anatoly Spitkovsky, Xuening Bai, Benoît Cerutti,
Greg Hammett, Damiano Caprioli, Alex Schekochihin, Geoffroy Lesur




guiding-centre coordinates
6D phase space
gyro-centre coordinates
5D phase space
gyro-centre transform
ρ_i