Kinetic Plasma Simulations

Anatoly Spitkovsky (Princeton)

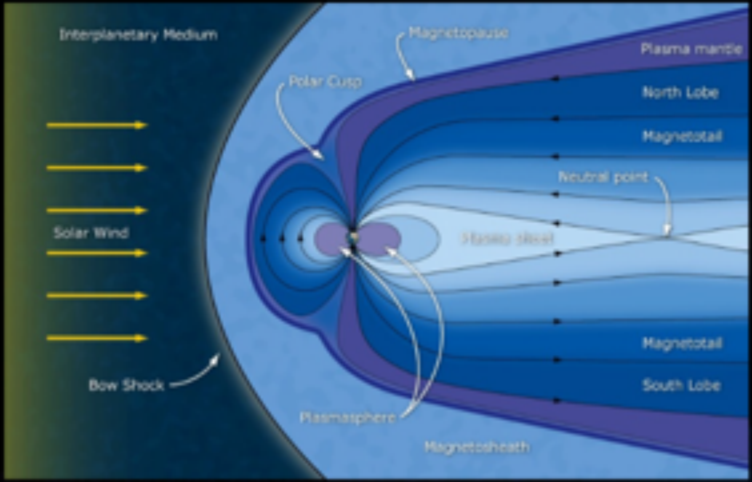
Kinetic Plasma Simulations

Anatoly Spitkovsky (Princeton)

Acknowledgement: many slides liberally borrowed from many colleagues

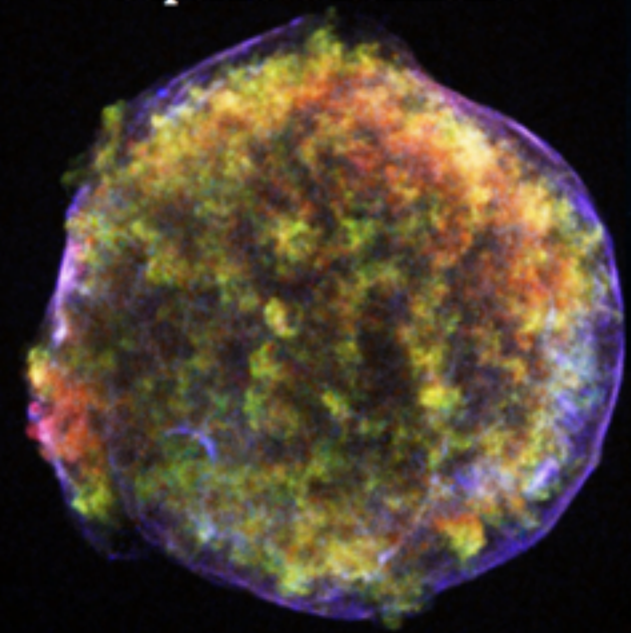
Astrophysical context

Planetary magnetospheres

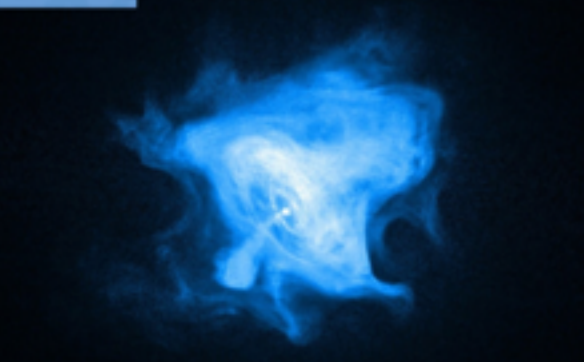


Solar corona & wind, heliosphere

Supernova Remnants



Pulsar Wind Nebulae

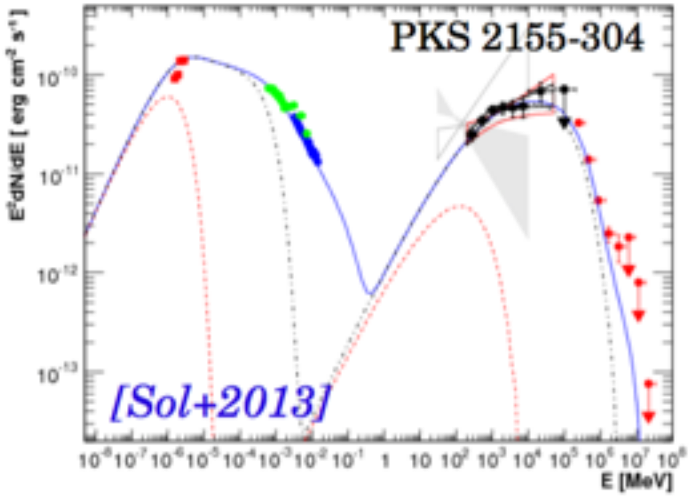


Gamma-ray bursts

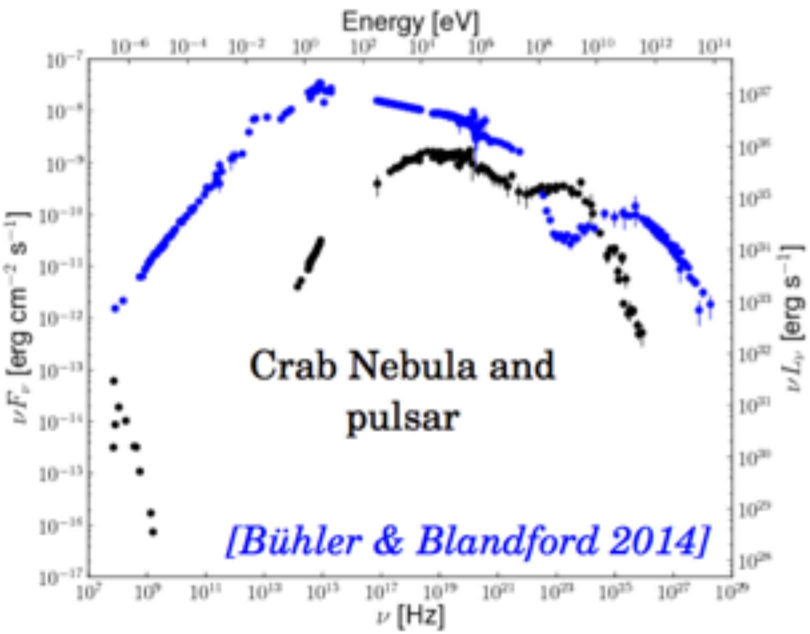
Jets

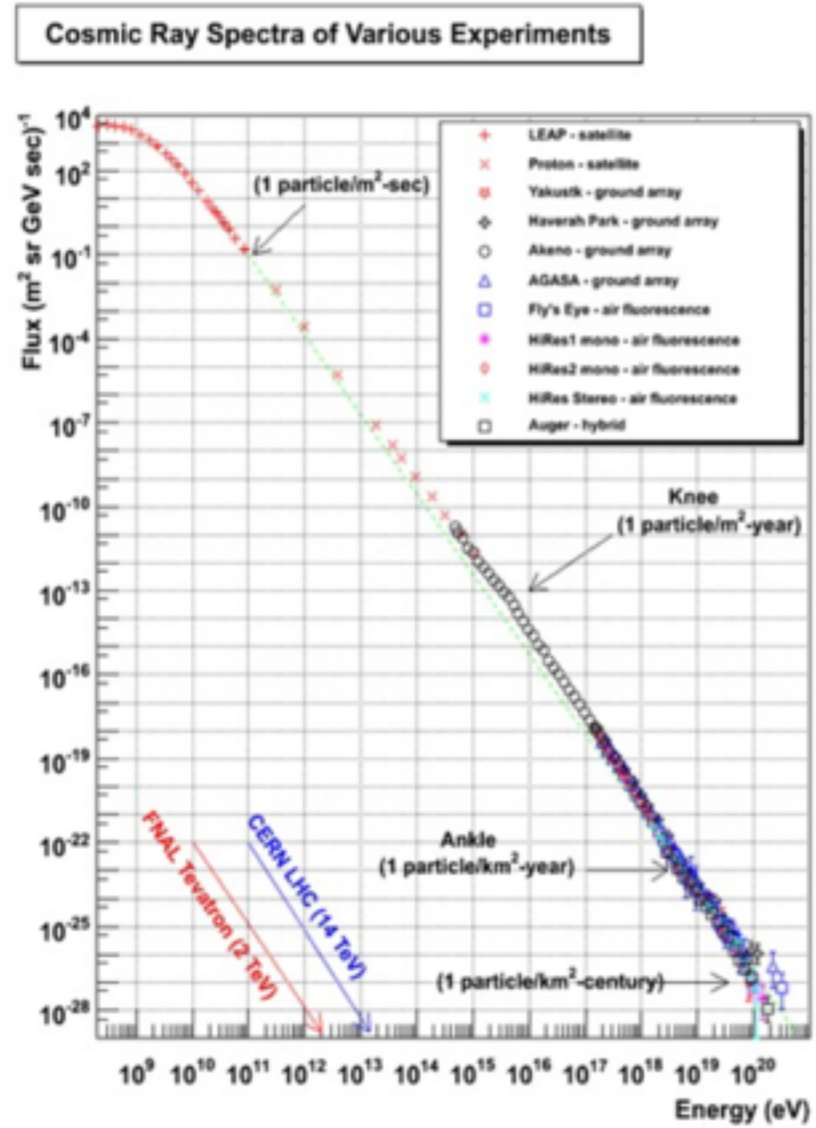
Broad non-thermal distributions

Blazars



Pulsars & Pulsar Wind Nebulae



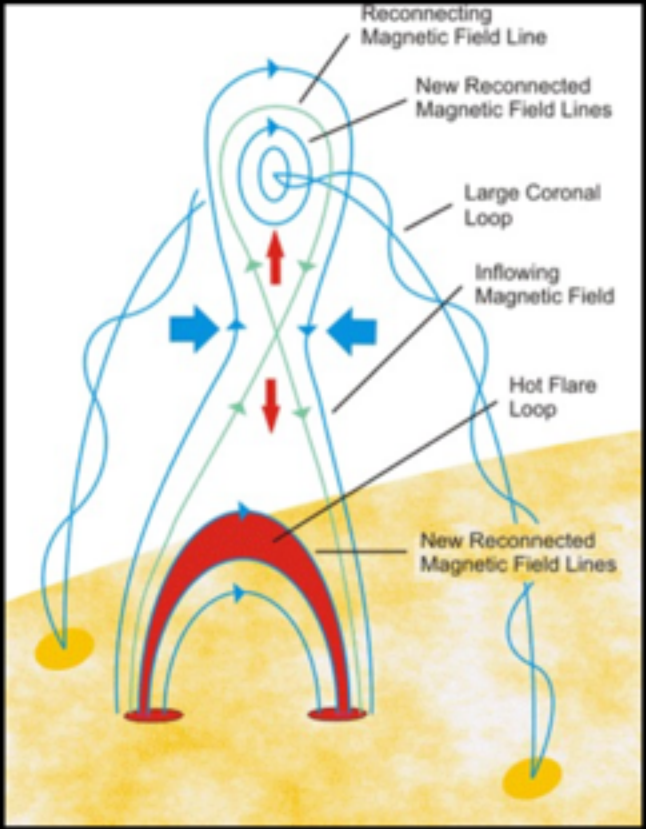


[http://www.physics.utah.edu/~whanlon/spectrum.html]

Particle acceleration processes

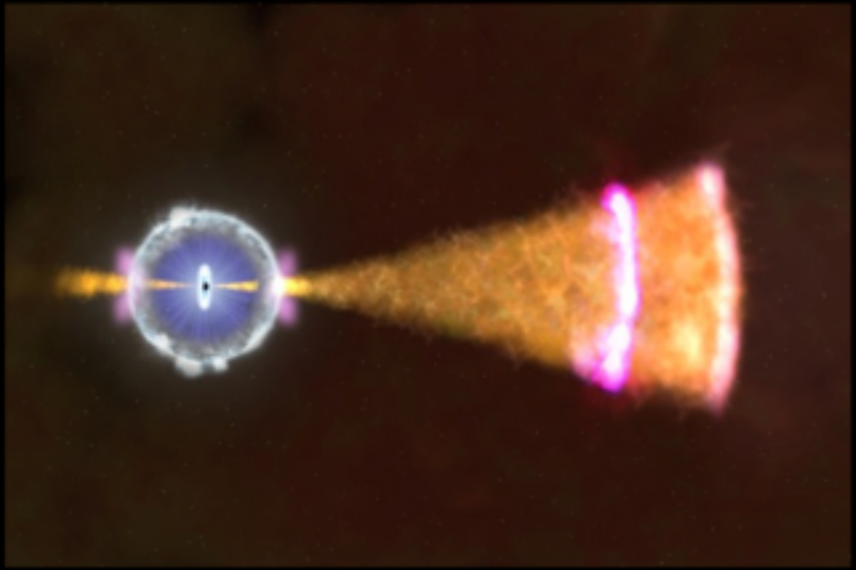
Magnetic reconnection

Magnetic energy => Particles



Shocks

Flow kinetic energy => Particles



Contents

Plasma physics on computers

Plasma: ionized gas (typically $T>10^4$K), 4th state of matter

Examples: stars, sun, ISM, solar wind, Earth magnetosphere, fluorescent lights, lightning, thermonuclear fusion

Plasma physics: studies plasma behavior through experiment, theory and … *simulation*!

Simulation needed to study collective and kinetic effects, especially in the nonlinear development.

Applications: reconnection, anomalous resistivity, instabilities, transport, heating, etc.

Plasma physics on computers

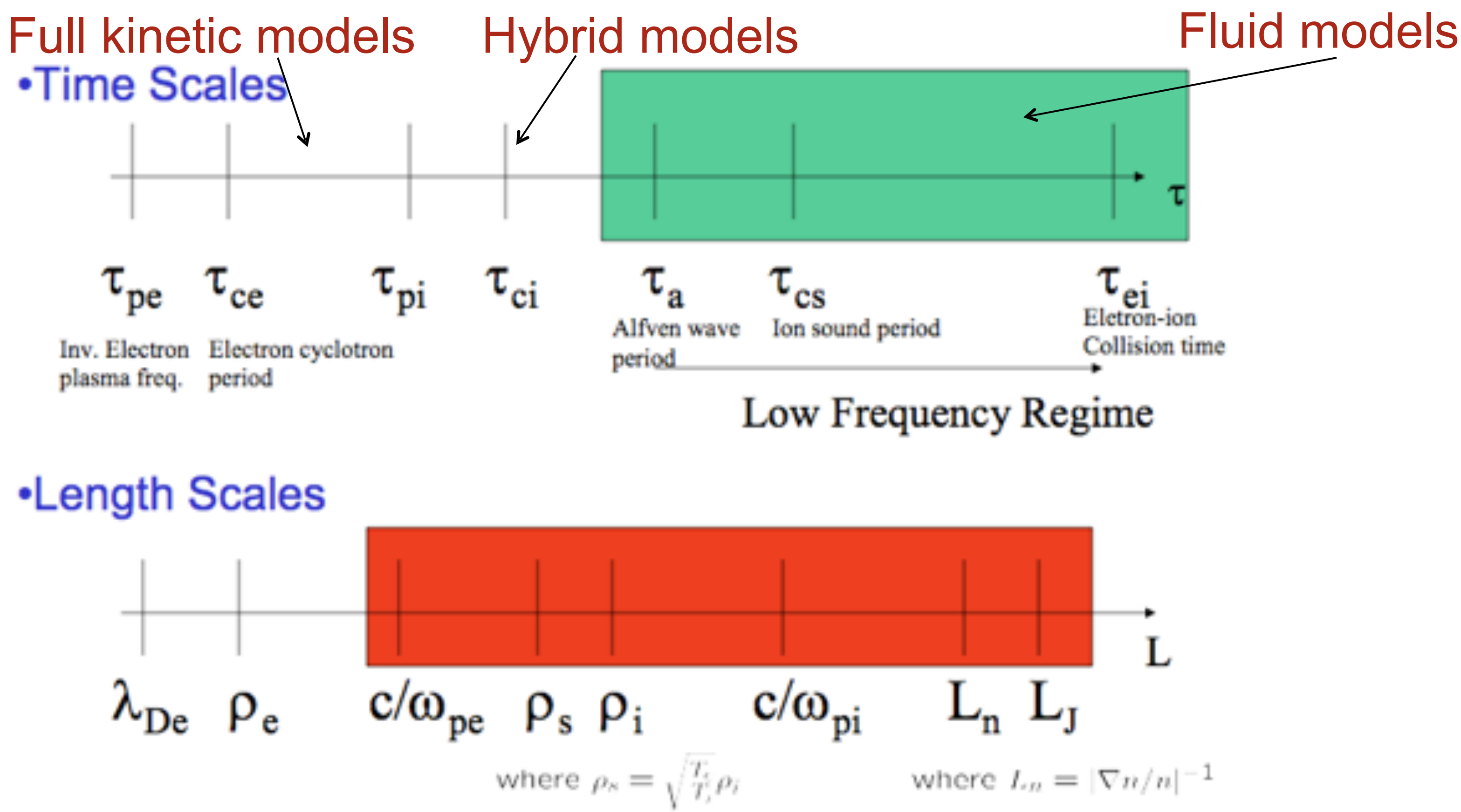
Characteristic time and length scales

$$\omega_p = \left(\frac{4\pi ne^2}{m}\right)^{1/2}$$
Plasma frequency

$$\lambda_D = \frac{V_{thermal}}{\omega_p} \propto \left(\frac{T}{n}\right)^{1/2}$$
Debye length

$$\lambda_{skin} = c/\omega_p$$
skin depth

$$\omega_c = \frac{eB}{mc}$$
Larmor



Plasma physics on computers

When are collisions important?

We are interested in

$$L \gg \lambda_D, \; t \gg \omega_P^{-1}, \; \omega_c^{-1}$$

Number of particles in Debye cube

$$N_D \equiv n\lambda_D^3$$

Plasma is collisionless if

$$L \gg \lambda_D, \; N_D \gg 1$$

Plasma Type	n cm^{-3}	T eV	ω_{pe} sec^{-1}	λ_D cm	$n\lambda_D{}^3$	ν_{ei} sec^{-1}
Interstellar gas	1	1	6×10^4	7×10^2	4×10^8	7×10^{-5}
Gaseous nebula	10^3	1	2×10^6	20	8×10^6	6×10^{-2}
Solar Corona	10^9	10^2	2×10^9	2×10^{-1}	8×10^6	60
Diffuse hot plasma	10^{12}	10^2	6×10^{10}	7×10^{-3}	4×10^5	40
Solar atmosphere, gas discharge	10^{14}	1	6×10^{11}	7×10^{-5}	40	2×10^9
Warm plasma	10^{14}	10	6×10^{11}	2×10^{-4}	8×10^2	10^7
Hot plasma	10^{14}	10^2	6×10^{11}	7×10^{-4}	4×10^4	4×10^6
Thermonuclear plasma	10^{15}	10^4	2×10^{12}	2×10^{-3}	8×10^6	5×10^4
Theta pinch	10^{16}	10^2	6×10^{12}	7×10^{-5}	4×10^3	3×10^8
Dense hot plasma	10^{18}	10^2	6×10^{13}	7×10^{-6}	4×10^2	2×10^{10}
Laser Plasma	10^{20}	10^2	6×10^{14}	7×10^{-7}	40	2×10^{12}

Collisionless system has a very large number of particles in Debye sphere

Plasma physics on computers

Collisionless plasma can be described by Vlasov-Maxwell system of equations for distribution function f(x,v,t):

$$\frac{\partial f}{\partial t} + \vec{v}\cdot\frac{\partial f}{\partial \vec{x}} + \frac{q}{m}\left(\vec{E} + \frac{\vec{v}\times\vec{B}}{c}\right)\cdot\frac{\partial f}{\partial \vec{v}} = 0$$

$$\nabla\cdot\vec{E} = 4\pi\int q f d^3\vec{v},$$

$$\nabla\times\vec{B} = \frac{1}{c}\frac{\partial \vec{E}}{\partial t} + \frac{4\pi}{c}\int q f \vec{v} d^3\vec{v},$$

$$\nabla\cdot\vec{B} = 0, \quad \nabla\times\vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}.$$

$$\frac{d\vec{v}_j}{dt} = \frac{q_j}{m_j}\left(\vec{E} + \frac{\vec{v}_j\times\vec{B}}{c}\right)$$

$$\frac{d\vec{x}}{dt} = \vec{v}$$

$$\nabla\times\vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}$$

$$\nabla\cdot\vec{B} = 0, \quad \nabla\cdot\vec{E} = 4\pi\rho$$

$$\nabla\times\vec{B} = \frac{4\pi\vec{j}}{c} + \frac{1}{c}\frac{\partial \vec{E}}{\partial t}$$

$$\rho(\vec{x}) = \sum_j q_j\delta(\vec{x}-\vec{x}_j)$$

$$\vec{j}(\vec{x}) = \sum_j q_j\vec{v}_j\delta(\vec{x}-\vec{x}_j)$$

Direct solution is 6D -- very expensive
Can solve along characteristics -- particles
Delta functions cause collisions -- smooth them

particle method!

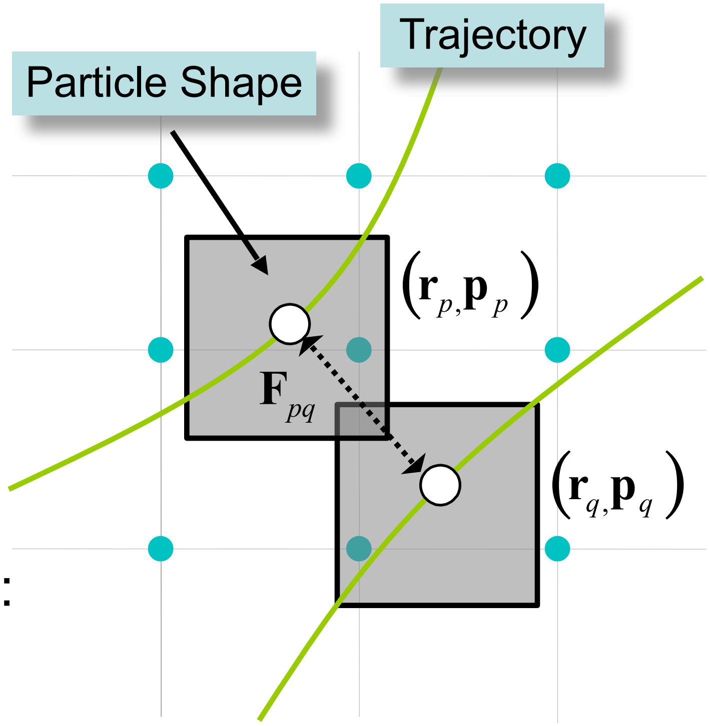
PIC Approach to Vlasov Equation (VE)

- 6D-VE not practical on a grid
- (Re-)introduce N_p computational particles for discretizing $f_1(\mathbf{r},\mathbf{p},t)$
- Macroscopic force $\langle \mathbf{F} \rangle$ becomes again granular (stochastic noise)

$$\delta \mathbf{F}^s \rightarrow \sqrt{1/N_p}$$

- Particle equations of motion (EQM):

$$\frac{d\mathbf{r}_p}{dt} = \frac{\mathbf{p}_p}{m}, \qquad \frac{d\mathbf{p}_p}{dt} = \mathbf{F}_p$$

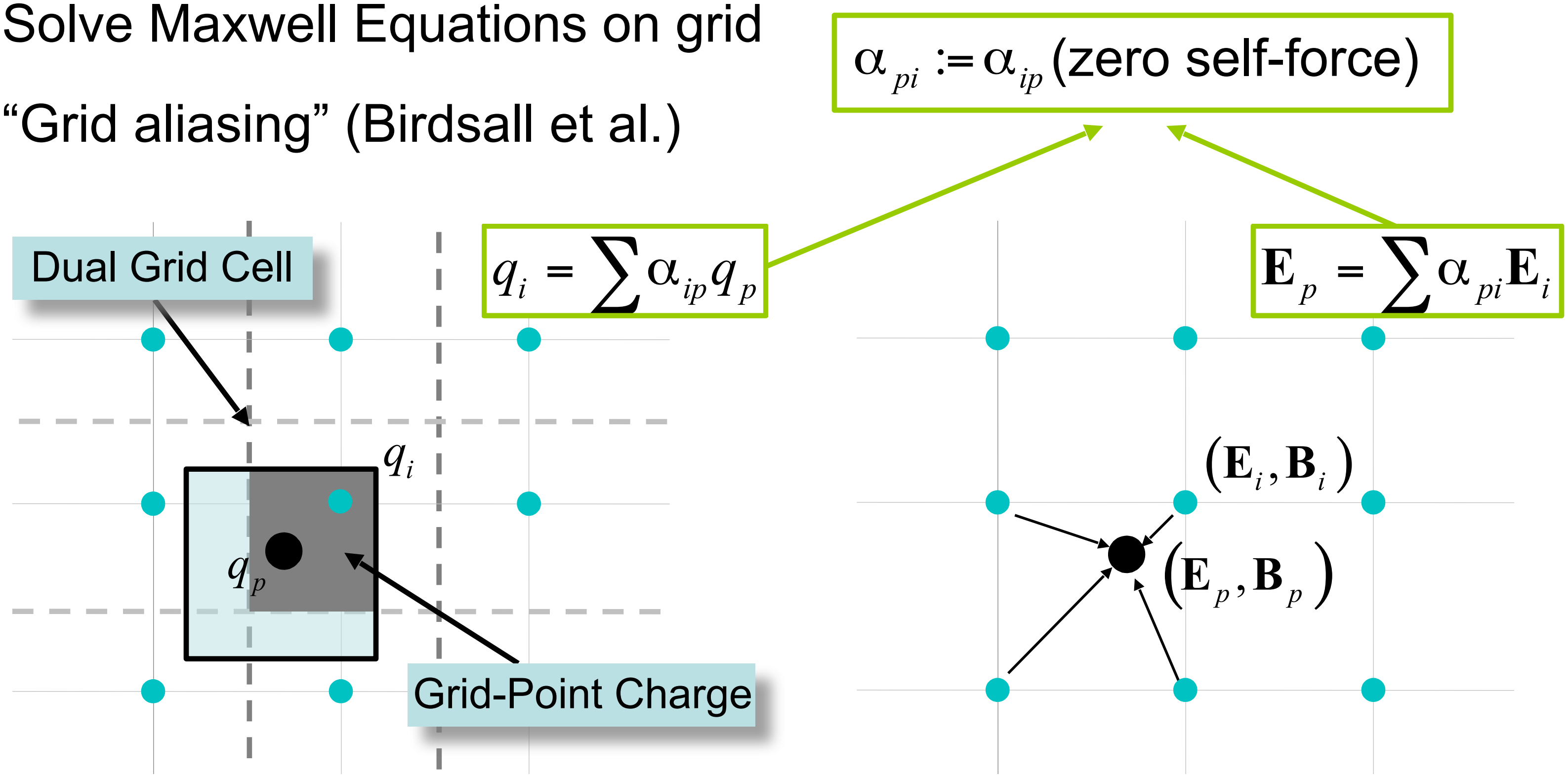


- VE characteristics: $f_1 = \text{const.}$
Particle strength (charge) const.

- Reduce operation count by computing forces on a grid

PIC Approach to Vlasov Equation

- Lorentz-Force: $\mathbf{F}_p = q\mathbf{E}_p + \frac{q}{m}\left(\mathbf{p}_p \times \mathbf{B}_p\right)$
- Solve Maxwell Equations on grid
- “Grid aliasing” (Birdsall et al.)



Charge Assignment

Force Interpolation

Momentum conservation

$\alpha_{pi} := \alpha_{ip}$ (zero self-force)

$$\frac{dP}{dt} = \sum_i F_i$$

$$\frac{dP}{dt} = \sum_i q_i \Delta x \sum_j E_j S(X_j - x_i)$$

$$\frac{dP}{dt} = \Delta x \sum_j E_j \sum_i q_i S(X_j - x_i)$$

$$\frac{dP}{dt} = \Delta x \sum_j \rho_j E_j$$

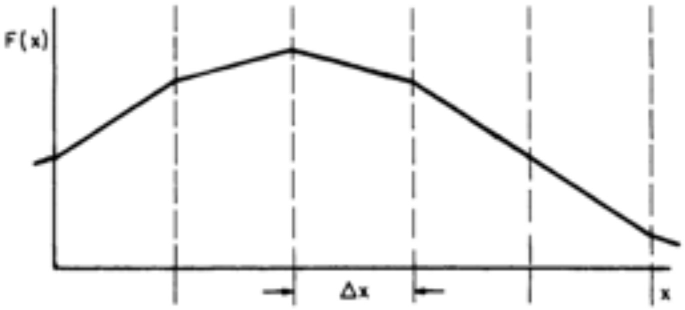


Figure 8-5b Force field $F(x)$ as a function of x is a set of straight-line segments, from grid point to grid point, in linear weighting.

For periodic system:

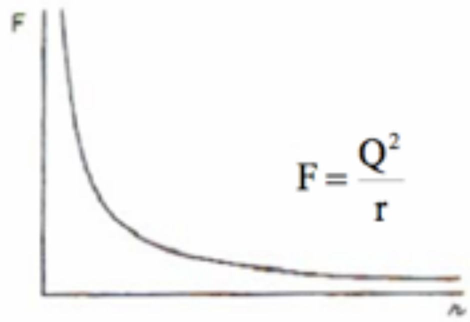
$$\Delta x \sum_j \rho_j E_j = 0$$

so momentum is conserved

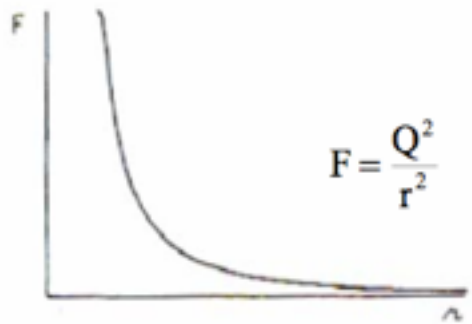
Interpolation to and from the grid have to be done in the same way

Finite-size particles

Coulomb force between point charges



short range force is responsible for collision effects



long range force is responsible for collective effects

FIG. 1. Coulomb force law between particles in two and three dimensions.

Since one simulation particle represents many point particles (Q=Nq), the short range force is over-estimated. So, finite-size particles are used.

The force law between finite-size particles

The finite-size particle considerably reduces the coulomb collision.

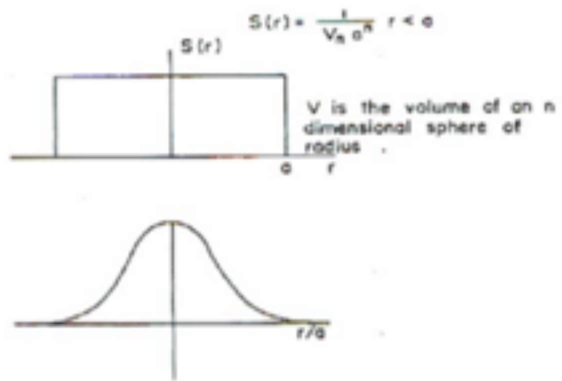


FIG. 4. Square and Gaussian charge shapes—two shapes often used for finite-sized particles.

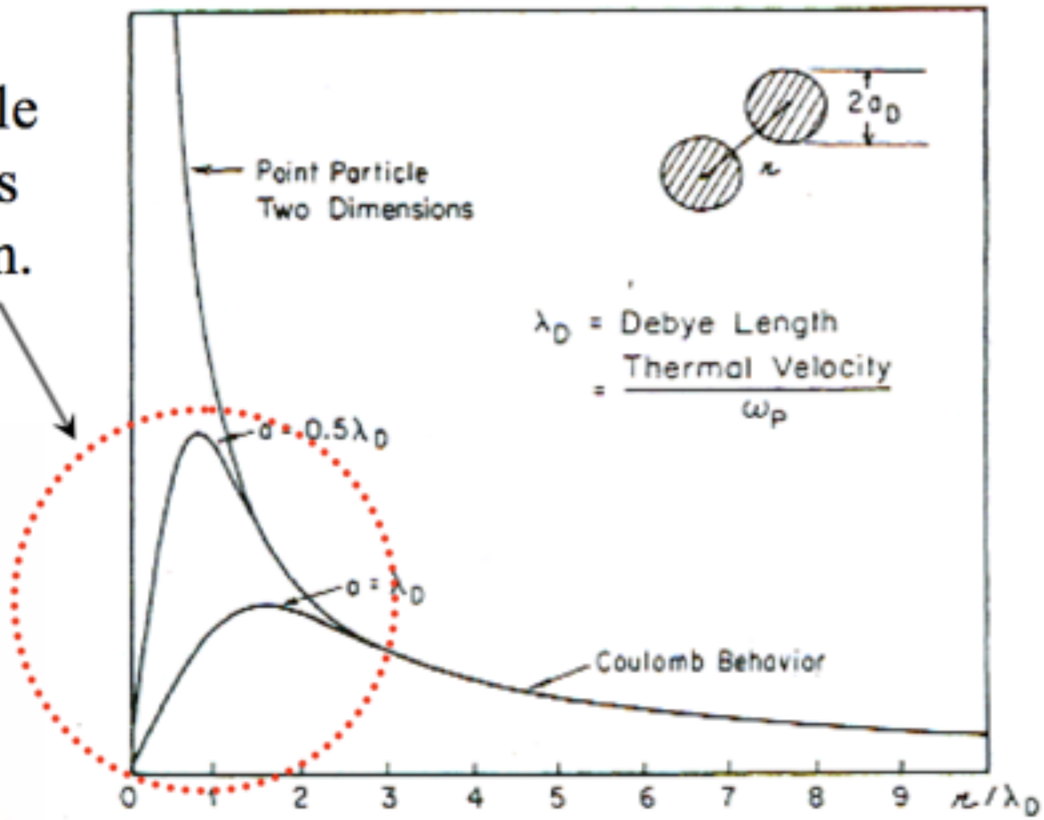
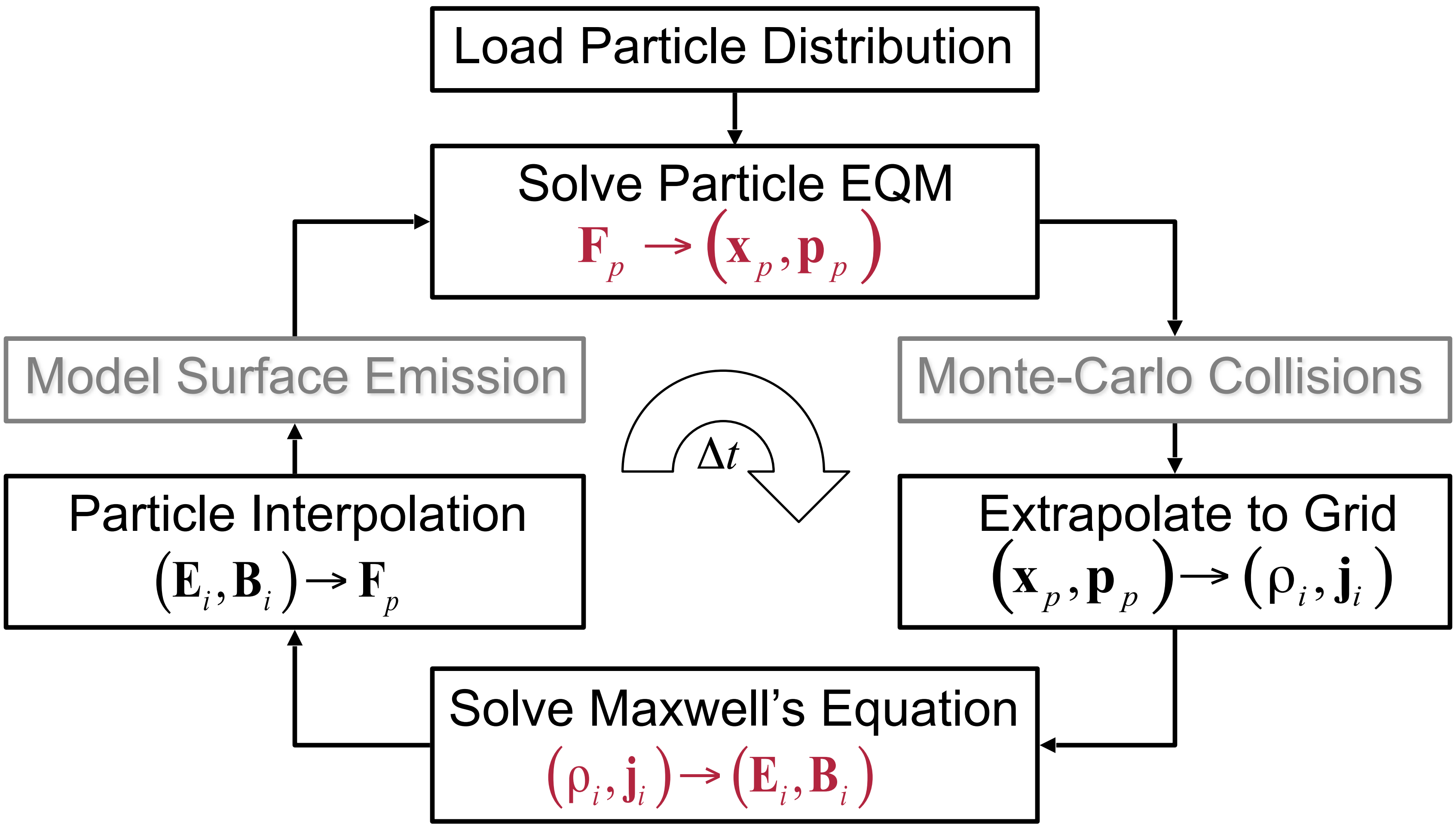


FIG. 2. Force law between finite-size particles in two dimensions for various sized particles. A Gaussian-shaped charge-density profile was used.

How PIC works

Simulation Flow-Chart



A bit of history:

In late 1950s John Dawson began 1D electrostatic "charge-sheet" experiments at Priceton, later @ UCLA.

1965 Hockney, Buneman -- introduced grids and direct Poisson solve

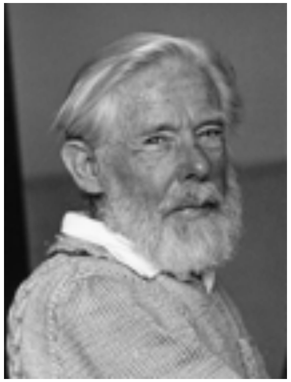
1970-s theory of electrostatic PIC developed (Langdon)

First electromagnetic codes

1980s-90s 3D EM PIC takes off

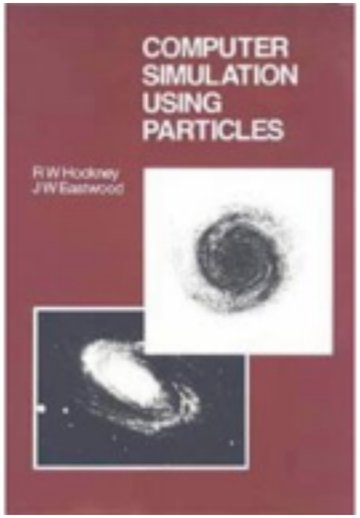
"PIC bibles" come out in 1988 and 1990

Always in step with Moore's law



Key names:

J. Dawson, O. Buneman, B. Langdon, C. Birdsall.



Contents

Electrostatic codes

Timescales of the system >> light crossing time; magnetic fields static.

$$\nabla^2\phi = -\rho(x)$$

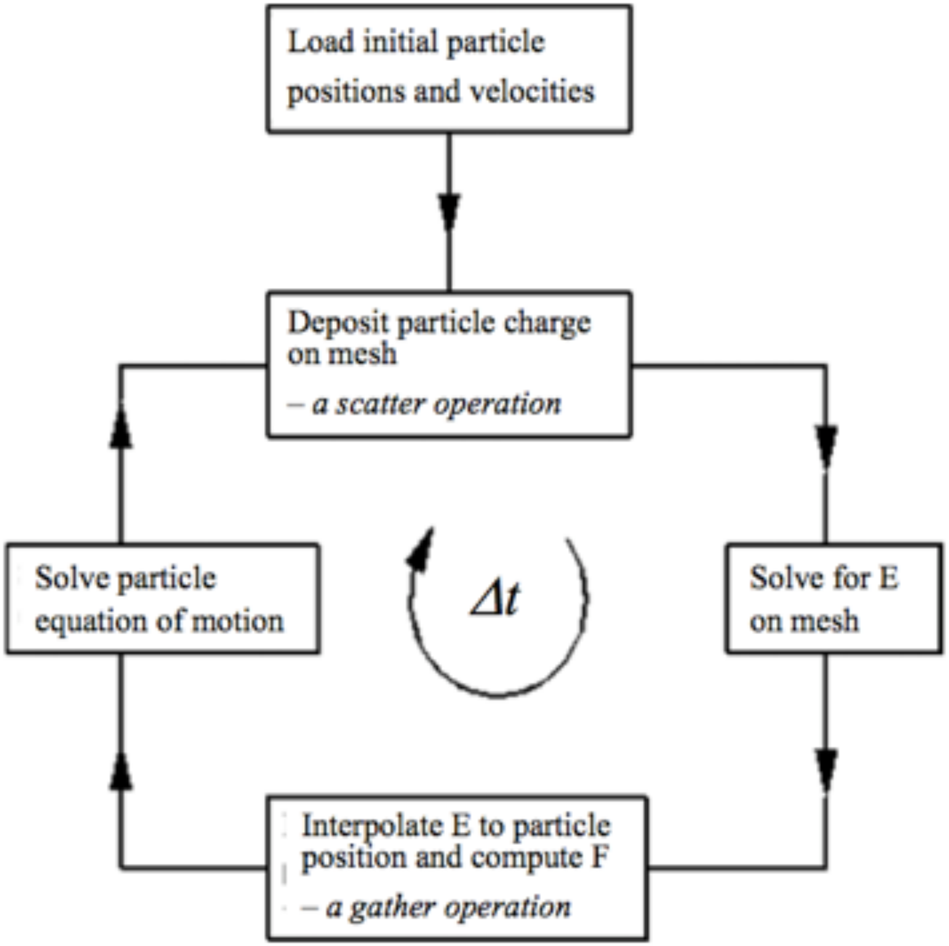
$$E(x) = -\nabla\phi$$

$$F_i = q_i E(x_i)$$

The number of floating point operations for the complete scheme scales as:

$$\alpha N_p + \beta N_g \ln N_g + \gamma N_g$$

if we use FFT to solve the Poisson's equation. Where N_g is the mesh number.



Four Major Criteria to Choose an Algorithm for Integration of Equations of Motion

•**Convergence** – Which means that the numerical solution converges to the exact solution of the differential equation in the limits as Δt and Δx tend to zero.

•**Accuracy** – Which means the truncation error associated with approximating derivatives with differences.

•**Stability** – If total errors (including truncation error and round-off error) grows in time, then the scheme is unstable.

•**Efficiency** – This is a critical consideration since whatever scheme we choose will be used for each particle at each time step.

Other Criteria to Choose an Algorithm for Integration of Equations of Motion

•**Dissipation** – The dissipation of some physical quantities caused by the truncation error associated with approximating derivatives with differences.

•**Conservation** – The deviation of the conservation law caused by the truncation error associated with approximating derivatives with differences.

Integration of Equations of Motion

The conventional wisdom is that the simple second order leapfrog achieves the best balance between accuracy, stability and efficiency.

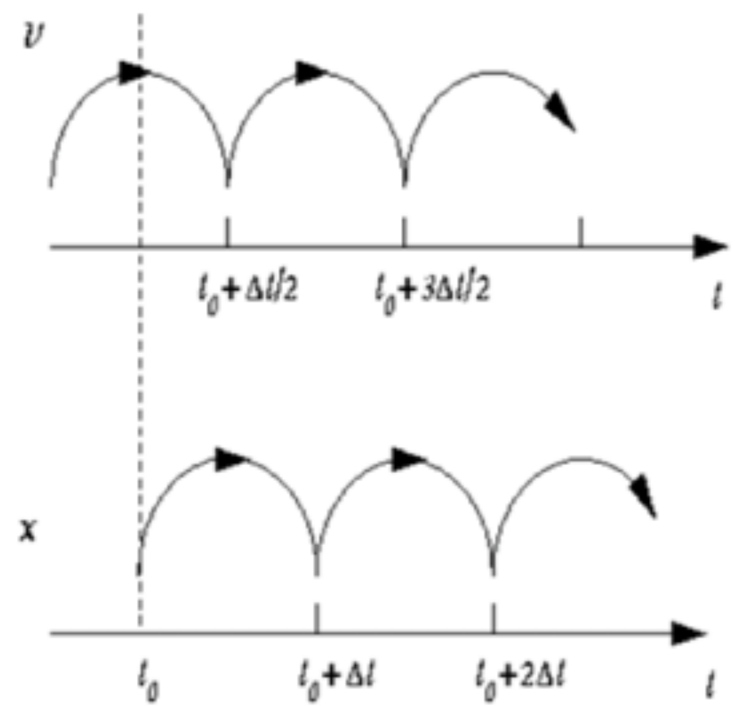
$$\frac{dx_i}{dt} = v_i$$

$$\frac{dv_i}{dt} = \frac{F_i}{m_i}$$

$$\downarrow$$

$$m_i \frac{v_i^{n+1/2} - v_i^{n-1/2}}{\Delta t} = F_i^n$$

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = v_i^{n+1/2}$$



Integration of Equations of Motion

For an electrostatic case, if $\omega_p \Delta t \leqq 2$, the leap frog scheme is stable.

$$\frac{d^2 x}{dt^2} = -\omega_0^2 x$$

↓

$$\sin\left(\frac{\omega \Delta t}{2}\right) = \pm \frac{\omega_0 \Delta t}{2}$$

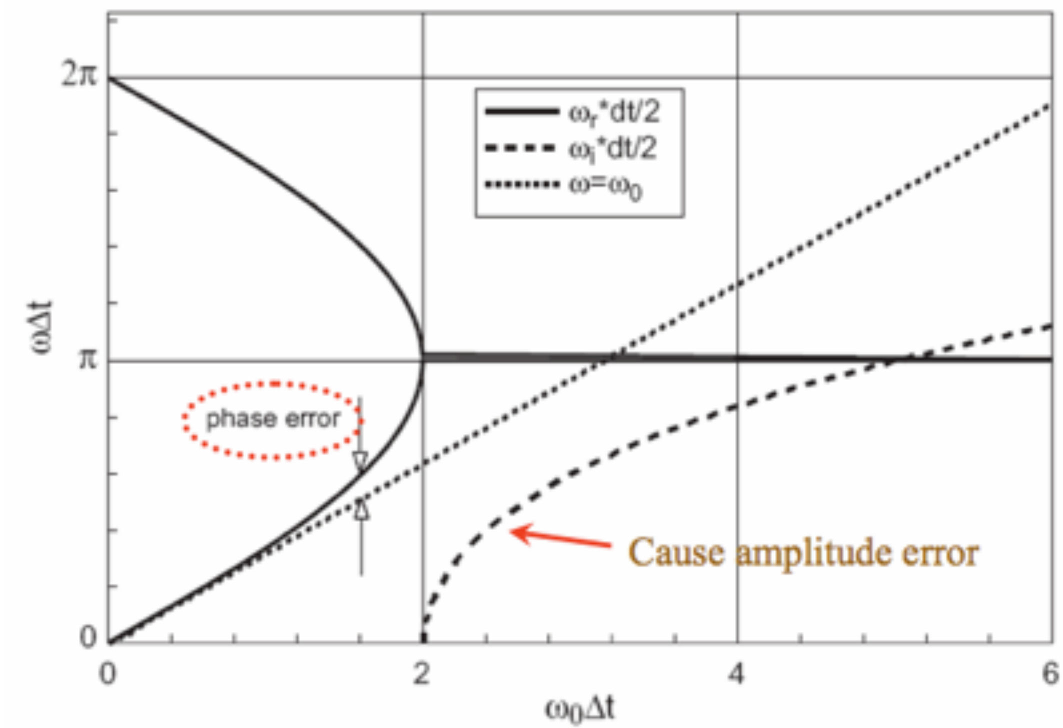
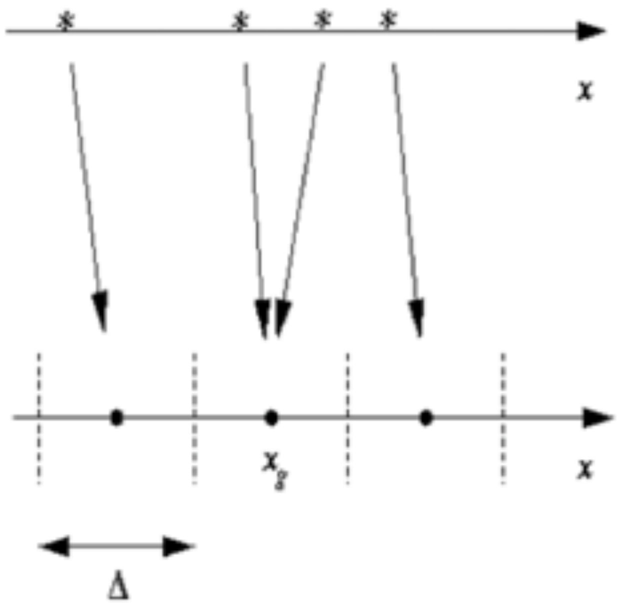


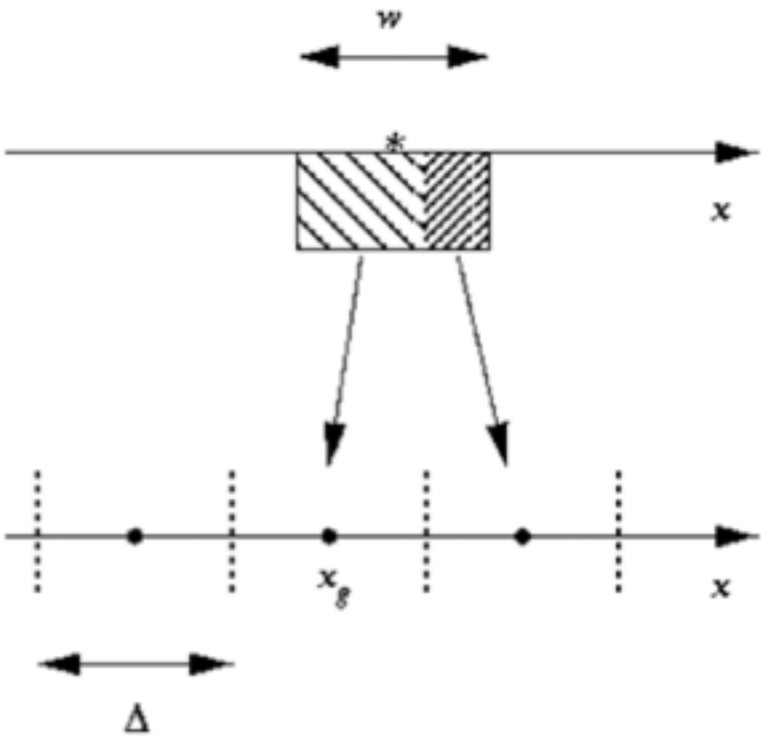
Figure 3. Angular frequency ω_r and numerical growth rate ω_i for the leapfrog scheme. Phase error is the difference between the numerical and exact frequency ω_0.

Charge Assignment and Force Interpolation

Once we introduce the grid we can no longer view the particles as point particles, this leads naturally to the idea of a finite sized particle.



Charge Assignment by (Nearest Grid Point) NGP in 1D.



Charge Assignment by (Cloud-in-Cell) CIC in 1D.

Charge Assignment and Force Evaluation by Cloud-in-Cell in 1D

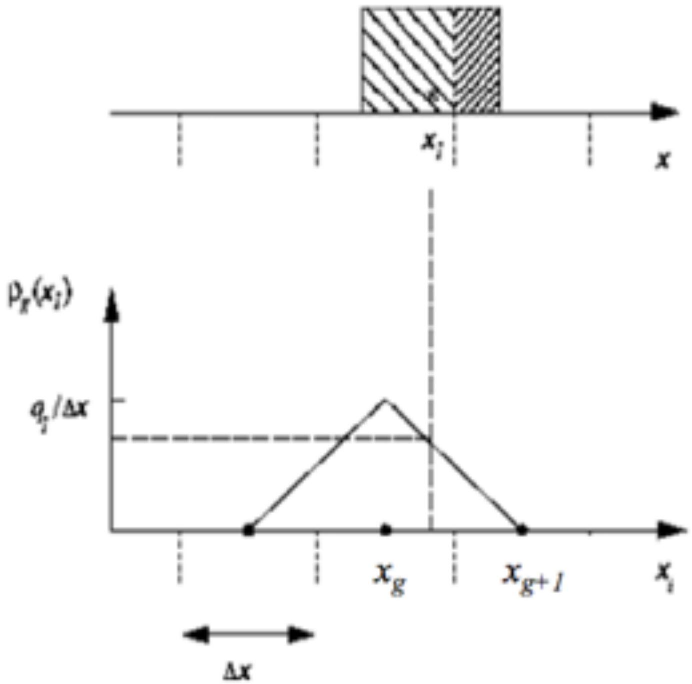
To ensure momentum conservation, the same interpolation scheme is used to compute the force on a particle as was used to perform the assignment of the particles charge to the mesh.

$$\rho_g = q_i \frac{x_{g+1} - x_i}{\Delta x}$$

$$\rho_{g+1} = q_i \frac{x_i - x_g}{\Delta x}$$

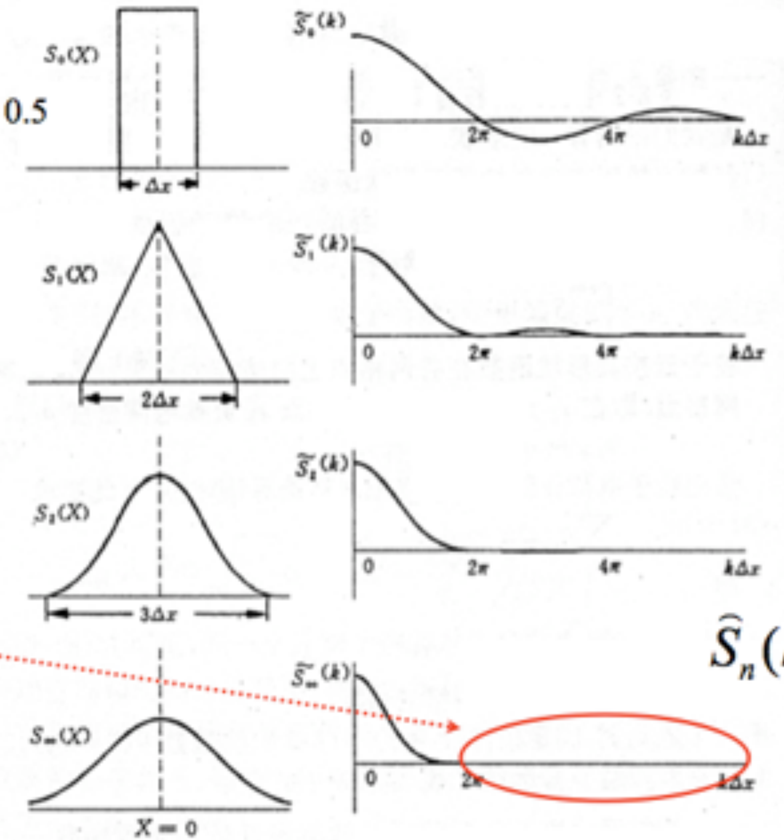
$$F_i = q_i \left(\frac{x_{g+1} - x_i}{\Delta x} E_g + \frac{x_i - x_g}{\Delta x} E_{g+1} \right)$$

$$where\ x_g \le x_i \le x_{g+1}$$



Filtering Action of Shape Functions

$$S_0(x) = \begin{cases} \dfrac{1}{\Delta x}, & -0.5 \le \dfrac{x}{\Delta x} \le 0.5 \\ 0, & \text{otherwise} \end{cases}$$



$$\hat{S}_0(k) = \sin\left(\tfrac{k\Delta x}{2}\right) \Big/ \tfrac{k\Delta x}{2}$$

$$\hat{S}_n(k) = [\hat{S}_0(k)]^{n+1}$$

$\hat{S}_n(k)$ decays with $k^{-(n+1)}$

High-frequency components are filtered by a smooth shape function.

Shape Functions of different orders

Corresponding Functions in Fourier Space

Integration of Field Equations

Here we solve the 1D form of Poisson's equation, then computes the electric field. L is the length of the system of interest.

$$\frac{\partial^2 \phi}{\partial x^2} = -\rho(x)$$

$$E_x = -\frac{\partial \phi}{\partial x}$$

Fourier Transform →

$$\hat{\phi}(k_l) = \frac{\hat{\rho}(k_l)}{k_l^2} \qquad k_l = \frac{2\pi\, l}{L}$$

$$\hat{E}(k_l) = -ik_l\hat{\phi}(k_l)$$

↓ Finite Difference

↑ $\Delta x \to 0$

$$K_l = k_l^2\left(\frac{\sin(\frac{1}{2}k_l\Delta x)}{\frac{1}{2}k_l\Delta x}\right)^2$$

$$\kappa_l = k_l\frac{\sin(k_l\Delta x)}{k_l\Delta x}$$

$$\frac{\phi_{g+1} - 2\phi_g + \phi_{g-1}}{\Delta x^2} = -\rho_g$$

$$E_g = \frac{\phi_{g+1} - \phi_{g-1}}{2\Delta x}$$

Fourier Transform →

$$\hat{\phi}(k_l) = \frac{\hat{\rho}(k_l)}{K_l}$$

$$\hat{E}(k_l) = -i\kappa_l\hat{\phi}(k_l)$$

In this case, differencing acts to dampen high k_l modes.

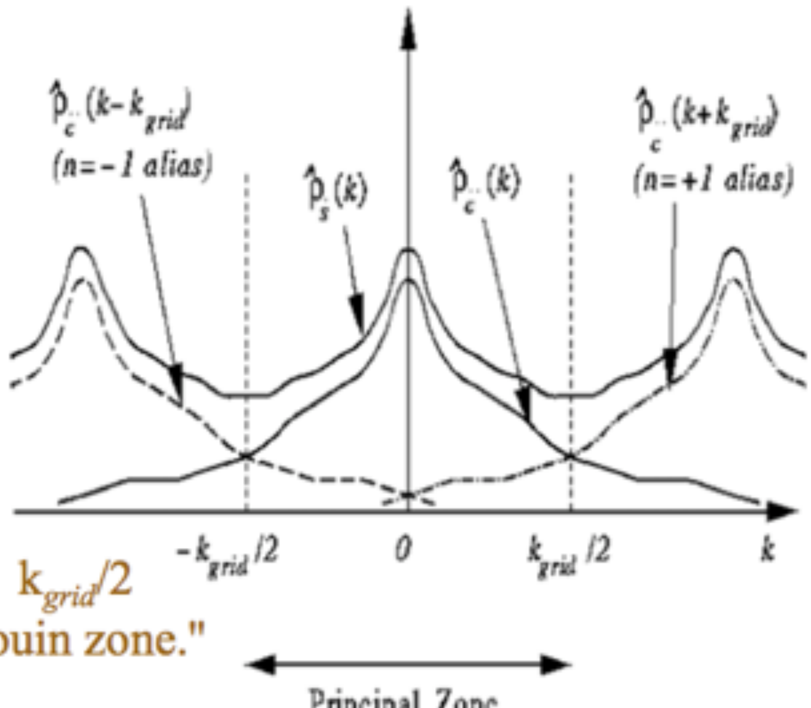
Aliasing

The spurious fluctuations which appear as a result of the loss of displacement invariance, manifest themselves in *k*-space as non-physical mode couplings, known as 'aliasing'.

By introducing a mesh we reduced our representation of $\rho(x)$ from a continuous representation $\rho_c(x)$ to a sampled representation $\rho_s(x_g)$.

$$\hat{\rho}_c(k) = \int_{-\infty}^{\infty} dx \rho_c(x) e^{-ikx}$$

$$\hat{\rho}_s(k) = \sum_{n=-\infty}^{\infty} \hat{\rho}_c(k + nk_{grid})$$



The wavenumber range $-k_{grid}/2 \leq k \leq k_{grid}/2$, called the ``principal zone" or ``first Brillouin zone."

The extra contributions (from $|n|>0$) to inside the principal zone are called aliases.

Aliasing

- The spurious fluctuations of high frequency causes the noise and error in the main lobe, which might make the numerical system to be unstable.
- The high-k components of S(k) is determined by the smoothness of $S(x)$; The high-k component of $n_c(k)$ is determined by the smoothness of $n_c(x)$, i.e. the number of particles.
- The major noise exists in the particle-in-cell method mainly comes from the aliasing effect. Two methods for reducing the aliasing effect:

  1. Increase the particle number.
  2. Increase the order of the shape function.

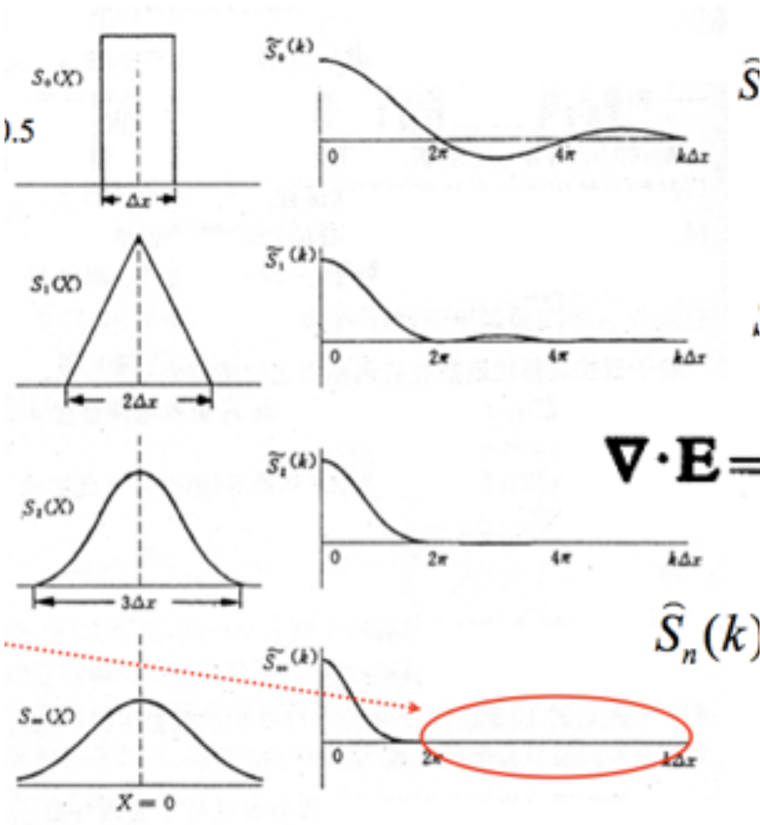
Noise Reduction in PIC

- The granularity of a particle representation inevitably introduces short-scale fluctuations into the force field, and the mean amplitude of these fluctuations is proportional to $\sqrt{n}$, where n is the particle number density.
- The ratio of the mean amplitudes of the fluctuations to the slowly varying component varies as $\frac{1}{\sqrt{n}}$, the effect of these fluctuations is greatly enhanced because our numerical model typically uses far fewer particles than are present in reality.
- We do not need to reduce the fluctuation amplitudes to their correct values, but merely to levels at which they no longer dominate the forces on the particles, or influence the particles significantly during the course of our simulation.

If Debye length is unresolved on the grid (<1cell), aliasing will heat up the plasma until Debye length is resolved -- num. heating

Electrostatic codes

Effects of particle shape factor on plasma dispersion



$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\frac{\partial f}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m}\cdot\frac{\partial f}{\partial \mathbf{v}} = 0$$

$$\mathbf{F}(\mathbf{r}_j) = q\int S(\mathbf{r}-\mathbf{r}_j)\mathbf{E}(\mathbf{r})\,d^n\mathbf{r}$$

$$\nabla\cdot\mathbf{E} = 4\pi q\int f(\mathbf{r}',\mathbf{v}')\,s(\mathbf{r}-\mathbf{r}')\,d^n\mathbf{r}'\,d^n v'$$

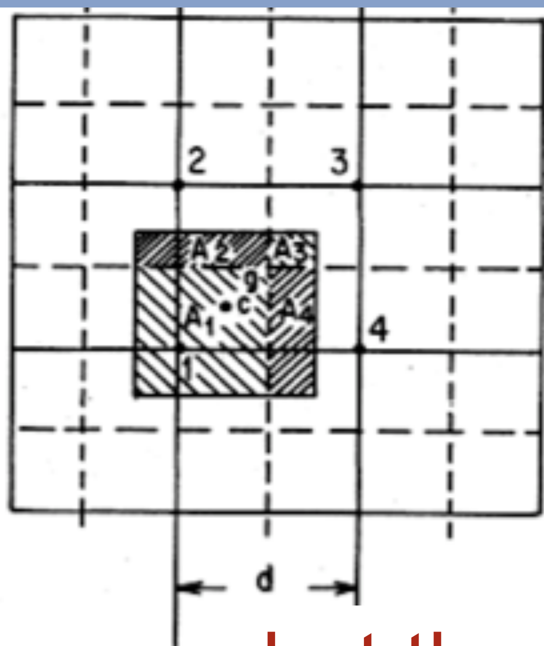
Plasma frequency is modified by smoothing

$$\omega^2(k) = \omega_p^2\,|S(k)|^2$$

$$\varepsilon(\mathbf{k},\omega) = 1 + \frac{\omega_p^2\,|S(\mathbf{k})|^2}{k^2}\int_{-\infty}^{\infty}\frac{\mathbf{k}\cdot\partial f_0/\partial v}{(\omega - \mathbf{k}\cdot\mathbf{v} + i\nu)}\,d^n v$$

Such Fourier space modifications also reduce collisions

Electrostatic codes



Extensions to 2D:

Usually, area weighting scheme is used for charge deposition and force interpolation

But -- can use other shape factors as well! Particles don’t have to be squares!!!

but the alternative is 6D Vlasov…

PIC issues:

- Particle discretization error
- Smoothing error (finite size particles)
- Statistical noise (granular force)
- Grid aliasing (grid assignment)
- Deterioration of quadrature in time integration
- Short-range forces (collisions) neglected

Summary for Restrictions of simulation parameters

Value of time step

1. Courant condition (rectangular coordinate)

$$dt < 1 \Big/ \sqrt{\frac{1}{dx_1^2} + \frac{1}{dx_2^2}} \qquad c = 1 \quad \omega_{pe} = 1$$

2. $\omega_{\max} dt < 0.25$

   ↑ The maximum frequency of system

3. $v_{\max} dt < \min(dx_1, dx_2)$

   particle move one time step < 1 cell (grid size)

Summary for Restrictions of simulation parameters

Resolution

1.Cell (grid) size

$$dx < \frac{\lambda}{m} \qquad m \approx 6 \sim 12 \qquad \lambda = \frac{2\pi}{k_{max}}$$

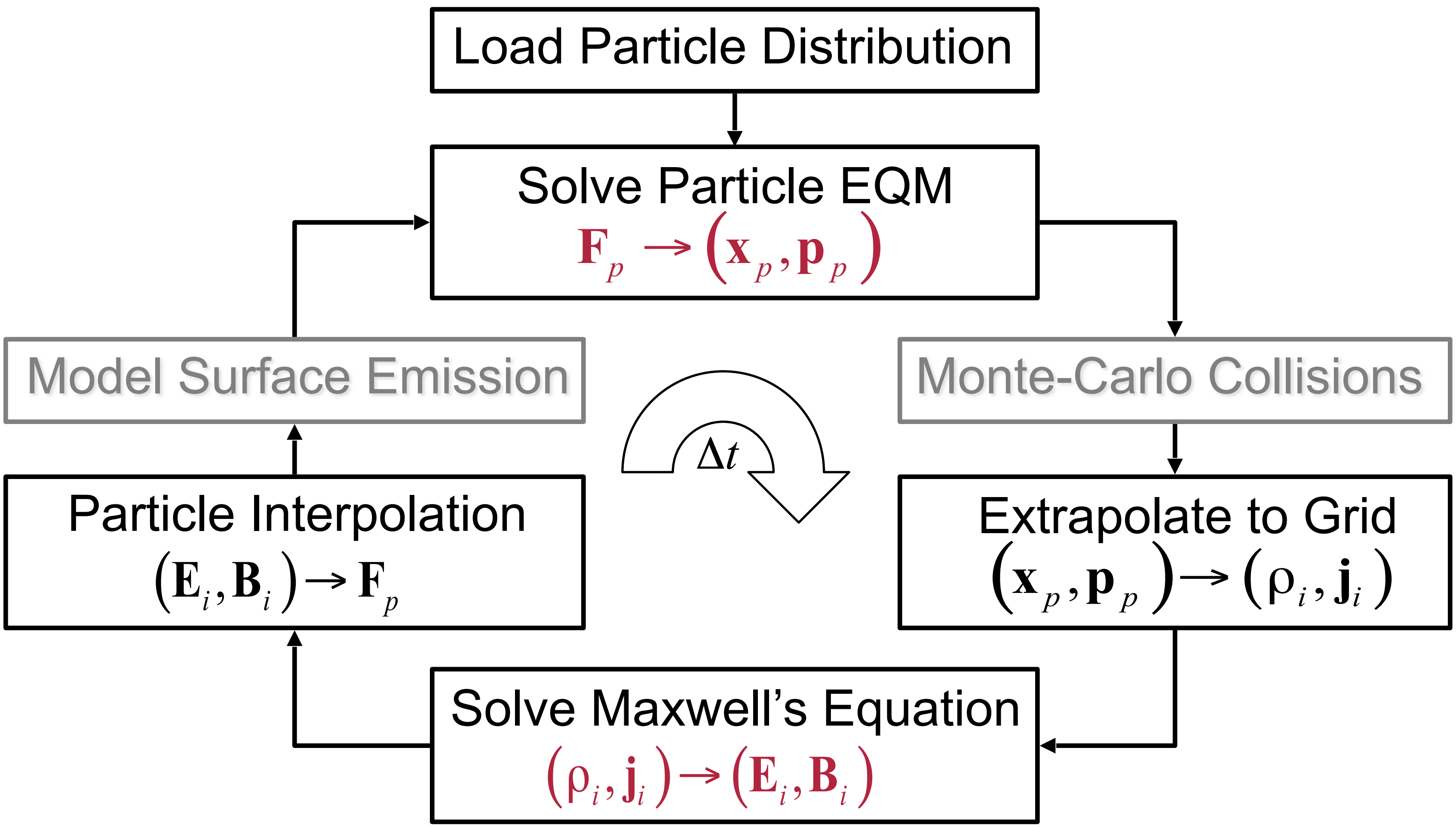
2.Particles per cell per species : 4-9

Contents

Electromagnetic codes

$$\partial E/\partial t = c(\nabla \times B) - 4\pi J\,, \qquad \nabla \cdot E = 4\pi\varrho\,, \qquad \nabla \cdot B = 0$$

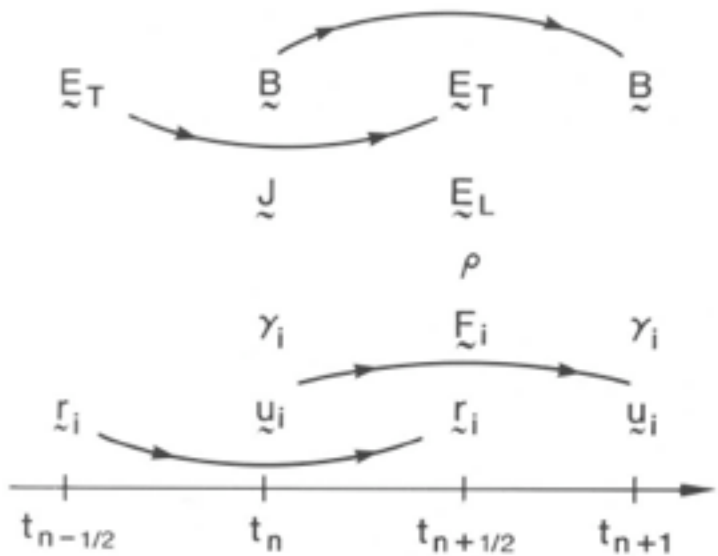
$$\partial B/\partial t = -c(\nabla \times E)\,, \qquad \frac{d}{dt}\gamma m\mathbf{v} = q(\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B})$$

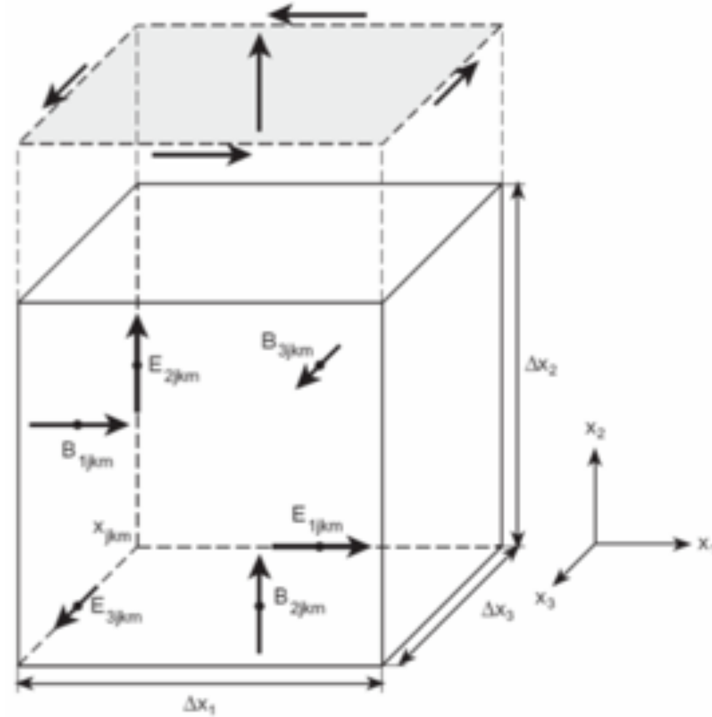


Electromagnetic codes

$$E^{n+1/2} = E^{n-1/2} + \Delta t[c(\nabla \times B^n) - 4\pi J^n]$$

$$B^{n+1} = B^n - c\Delta t \nabla \times E^{n+1/2} \, .$$





Fields defined on the Yee mesh. Currents, not shown, are co-located with the corresponding electric field components. Exploded view shows an integration surface.

Fields are decentered both in time and in space

Finite-difference Time-Domain Maxwell solver on Yee (1966) mesh: robust and very simple. Second order in space and time.

Decentering conserves div B to machine precision

Integration of Field Equations

The new set of field variables encapsulate the mesh metrics.

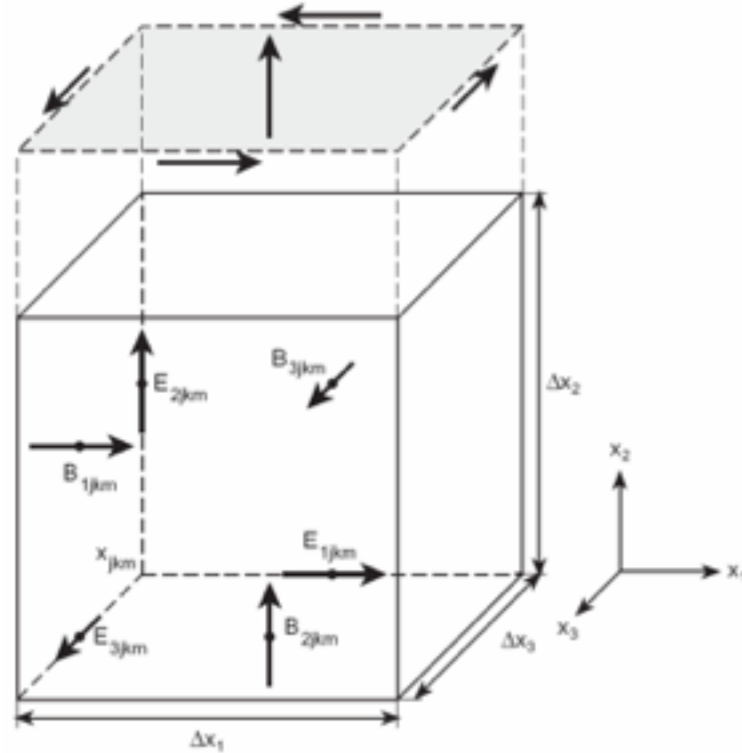
$$\tilde{E} = \int E \bullet dl \qquad \tilde{D} = \int D \bullet dS$$

$$\tilde{H} = \int H \bullet dl \qquad \tilde{B} = \int B \bullet dS$$

$$\tilde{I} = \int J \bullet dS$$

$$E^{n+1/2} = E^{n-1/2} + \Delta t[c(\nabla \times B^n) - 4\pi J^n]$$

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



Fields defined on the Yee mesh. Currents, not shown, are co-located with the corresponding electric field components. Exploded view shows an integration surface.

Electromagnetic codes

- We solve the discrete Maxwell equations:

$$E_j^n = \exp\left(i\omega n\Delta t + ikj\Delta x\right) \ ,$$
$$B_j^n = \exp\left(i\omega n\Delta t + ikj\Delta x\right) \ .$$

- The solution becomes:

$$\sin\left(\frac{\omega\Delta t}{2}\right) = \frac{\Delta t}{\Delta x}\sin\left(\frac{k\Delta x}{2}\right) \ .$$

- For $k \approx \pi/\Delta x$ and $\Delta t > \Delta x$ the solution adopts complex roots.
- For complex roots the solution grows to infinity.
- Then the scheme becomes unstable.

Electromagnetic codes

- The vacuum dispersion relation for the continuum system is:

$$\omega^2 = k^2 .$$

+ 1

- The vacuum dispersion relation for the discrete system becomes:

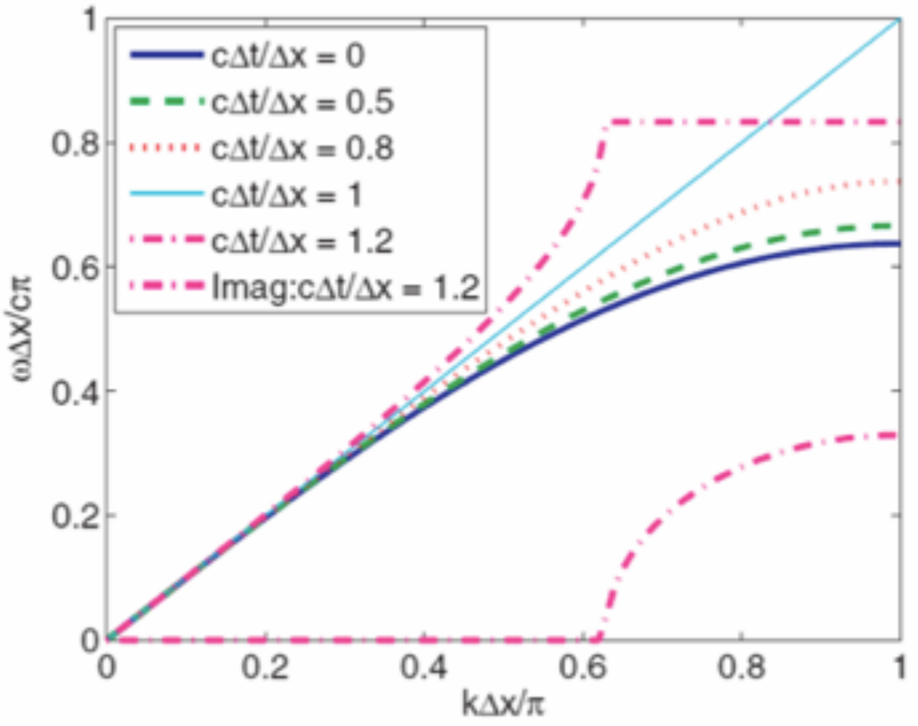
$$\sin^2\left(\frac{\omega \Delta t}{2}\right) = \left(\frac{\Delta t}{\Delta x}\right)^2 \sin^2\left(\frac{k \Delta x}{2}\right) .$$

- This means:

$$\frac{\omega \Delta t}{2} = \arcsin\sqrt{\left(\frac{\Delta t}{\Delta x}\right)^2 \sin^2\left(\frac{k \Delta x}{2}\right)} .$$

- For $k \approx \pi/\Delta x$ and $\Delta t > \Delta x$ no real valued solution is possible.

Electromagnetic codes



Vacuum dispersion curve for leapfrog difference scheme for wave equation.

$$\cos(\omega\Delta t) = \left(c\frac{\Delta t}{\Delta x}\right)^2 [\cos(\tilde{k}\Delta x) - 1] + 1$$

Courant–Levy Stability Criterion

$$\Delta t \leqslant \frac{1}{c}\left(\sum_i \frac{1}{(\Delta x_i)^2}\right)^{-1/2}$$

Numerical dispersion is anisotropic (best along grid diagonal)

Phase error for short wavelengths

Causes numerical Cherenkov radiation (when relativistic particles move faster than numerical speed of light)

Integration of Equations of Motion

Newton–Lorentz equations of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma m v = F = q\,(E + v \times B)$$

$$\gamma = \sqrt{\frac{1}{1-(v/c)^2}} = \sqrt{1+\left(\frac{u}{c}\right)^2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x = v$$

$$u = \gamma v$$

Leapfrog Scheme ↓ (Implicit Scheme)

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

$$\frac{x^{t+\Delta t} - x^t}{\Delta t} = \frac{u^{t+\Delta t/2}}{\gamma^{t+\Delta t/2}}$$

$$\gamma^t = (\gamma^{t-\Delta t/2} + \gamma^{t+\Delta t/2})/2$$

Integration of Equations of Motion

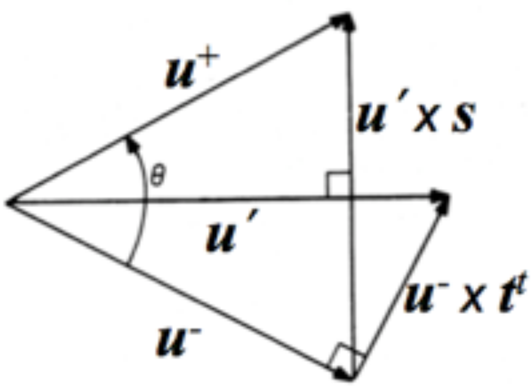
Boris Scheme* (Explicit Scheme)

$$u_{t-\Delta t/2} = u^- - \frac{qE}{m}\frac{\Delta t}{2}$$

$$u_{t+\Delta t/2} = u^+ + \frac{qE}{m}\frac{\Delta t}{2}$$

$$\downarrow$$

$$\frac{u^+ - u^-}{\Delta t} = \frac{q}{2m}(u^+ + u^-)\times B$$



$$\longrightarrow$$

$$u^- = u^{t-\Delta t/2} + \frac{q\Delta t E^t}{2m},$$

$$u' = u^- + u^- \times t',$$

$$u^+ = u^- + u' \times \frac{2t'}{1 + t'\cdot t'} \quad (\leftarrow S)$$

$$u^{t+\Delta t/2} = u^+ + \frac{q\Delta t E^t}{2m}$$

with

$$\mathbf{t}' = \frac{q\Delta t}{2\gamma^t m}\mathbf{B}^t$$

$$\theta = 2\arctan(t') = 2\arctan(qB\Delta t/2\gamma m)$$

* Boris J P 1970 Relativistic plasma simulation—optimization of a hybrid code *Proc. 4th Conf. on Numerical Simulation of Plasmas (Washington, DC)* pp 3–67.

Can overstep magnetic rotation without stability issues.

Electromagnetic codes

Charge and current deposition

$$\partial \mathbf{E}/\partial t = c(\nabla \times \mathbf{B}) - 4\pi \mathbf{J} ,$$

$$\partial \mathbf{B}/\partial t = -c(\nabla \times \mathbf{E}) ,$$

What to do about the Poisson equation?

Should we solve an elliptic equation in addition to hyperbolic Ampere's and Faraday's laws?

Turns out we can avoid solving Poisson equation if charge is conserved.

Take divergence of Ampere's law:

$$\frac{\partial \nabla \cdot E}{\partial t} = \cancel{c\nabla \cdot (\nabla \times B)} - 4\pi \nabla \cdot J$$

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot J$$

If charge is conserved, Poisson equation is just an initial condition. Like divB=0, if Poisson is true at t=0, it will remain satisfied.

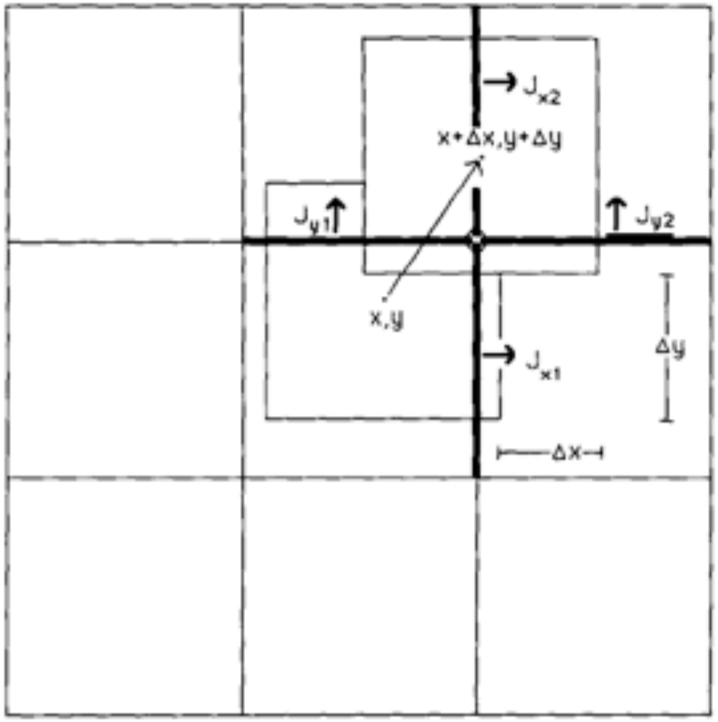
Charge and current deposition

Charge-conservative current deposition method
If just use volume-weighting, charge is not conserved.

Villasenor & Buneman (92):

Count what is the "volume current" through appropriate faces.

Also, need to know if the particle crosses four or 7 boundaries (2d).



Weighting Charge to the Grids

A single particle is weighted to surrounding nodes

$$Q_{j,k} = q(1-w_1)(1-w_2)$$

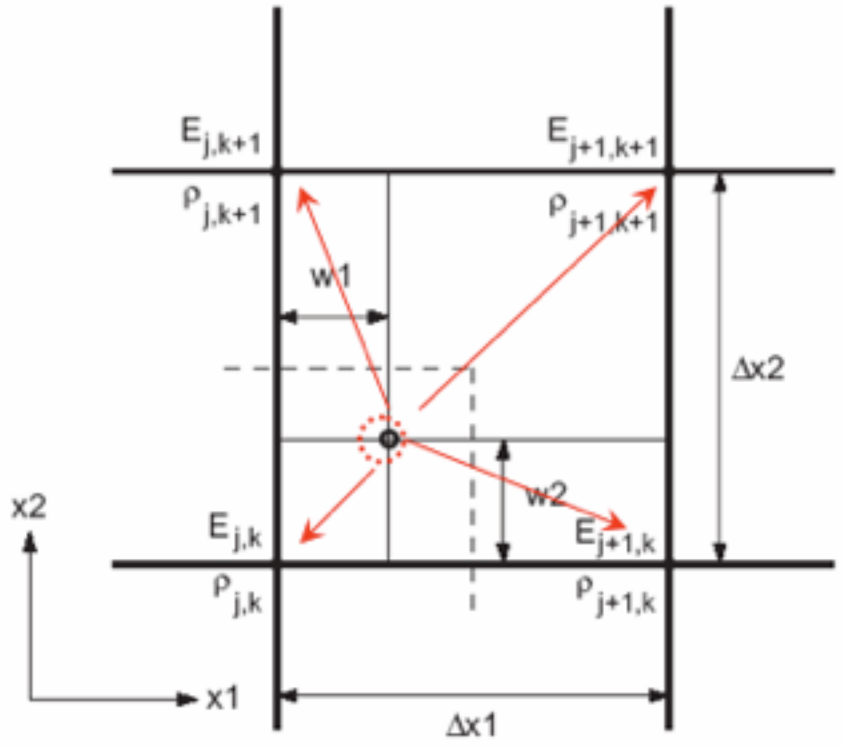
$$Q_{j+1,k} = qw_1(1-w_2)$$

$$Q_{j,k+1} = q(1-w_1)w_2$$

$$Q_{j+1,k+1} = qw_1w_2$$

The charge density is calculated using

$$\rho_{j,k} = Q_{j,k} / (\Delta x_1 \times \Delta x_2)$$



Weighting Current to the Grids

$$I_{1,j+\frac{1}{2},k} = \sum_i \frac{q_i}{\Delta t} \Delta w_1 (1 - \overline{w}_2)$$

$$I_{1,j+\frac{1}{2},k+1} = \sum_i \frac{q_i}{\Delta t} \Delta w_1 \overline{w}_2$$

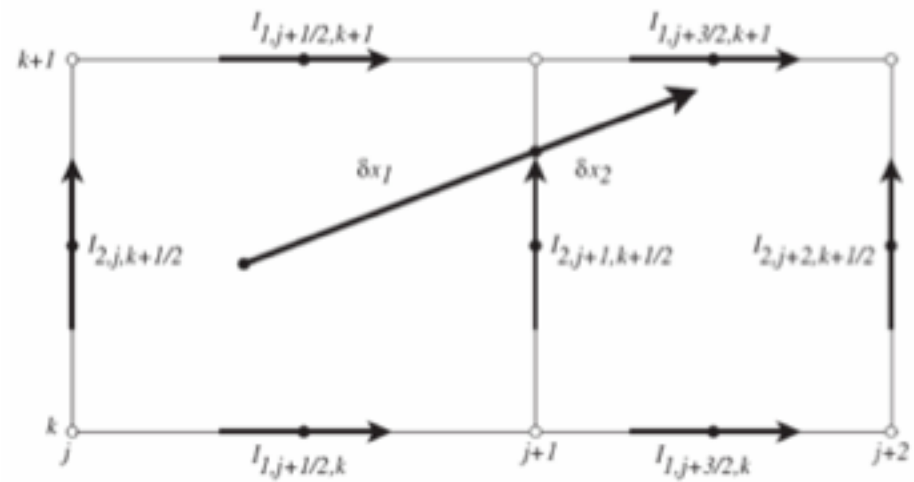
$$I_{2,j,k+\frac{1}{2}} = \sum_i \frac{q_i}{\Delta t} \Delta w_2 (1 - \overline{w}_1)$$

$$I_{2,j+1,k+\frac{1}{2}} = \sum_i \frac{q_i}{\Delta t} \Delta w_2 \overline{w}_1$$

$$w = x_i - X_{jk}$$

$\boldsymbol{x}_i$: refers to the position of the i_{th} particle
$\boldsymbol{X}_{jk}$: the position of the nearest lower mesh node

Charge Conserving Current Weighting

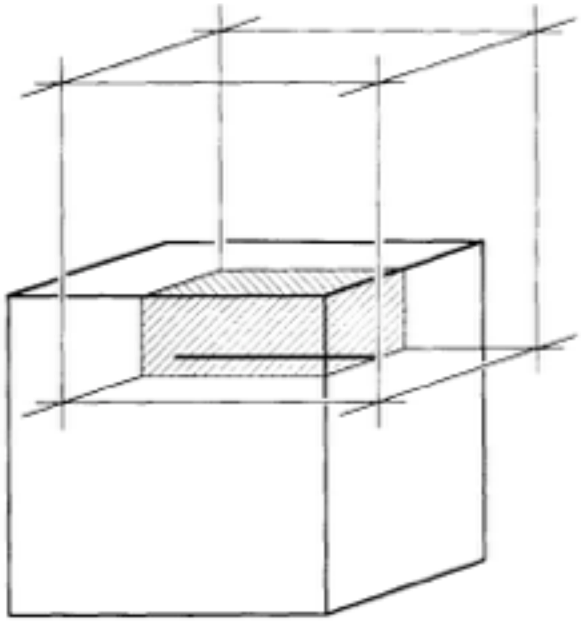


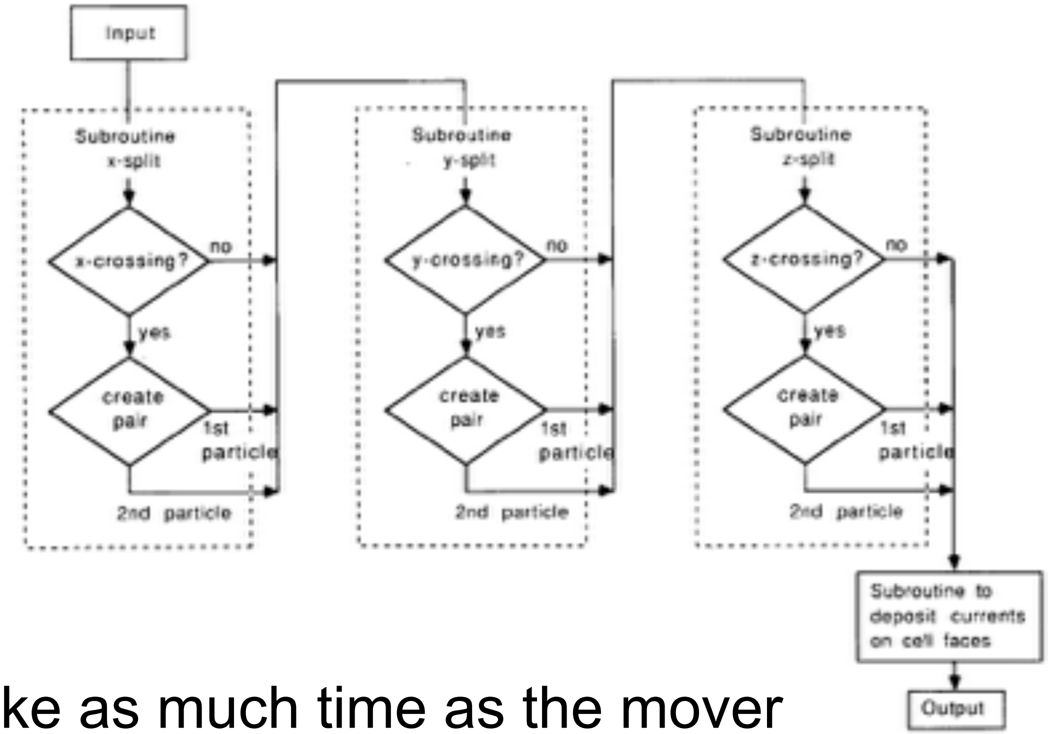
$$\Delta w = w^{t+\Delta t} - w^t$$

$$\bar{w} = \frac{w^{t+\Delta t} + w^t}{2}$$

Electromagnetic codes

Charge and current deposition





Current deposition can take as much time as the mover (sometimes more). More optimized deposits exist (Umeda 2003).

Higher order schemes possible (Esirkepov 2001, Umeda 2004) Charge conservation makes the whole Maxwell solver local and hyperbolic (like nature intended!). Static fields can be established dynamically.

Electromagnetic codes

Special sauce

Particle shape should be smoothed to reduce noise. We use current filtering after deposition to reduce high frequency aliases.

Higher order FDTD schemes (4th spatial order) work better at reducing unphysical Cherenkov instability.

Boundary conditions

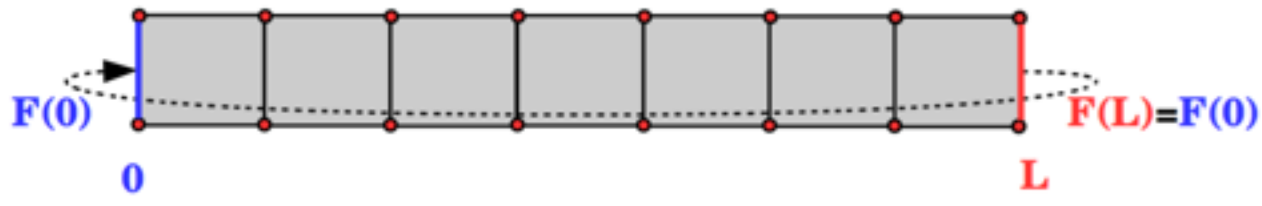
Periodic is simple -- just copy ghost zones and loop particles. Should not forget particle charge on the other side of the grid!

Conducting BCs: set E field parallel to boundary to 0. Boundary has to lie along the grid.

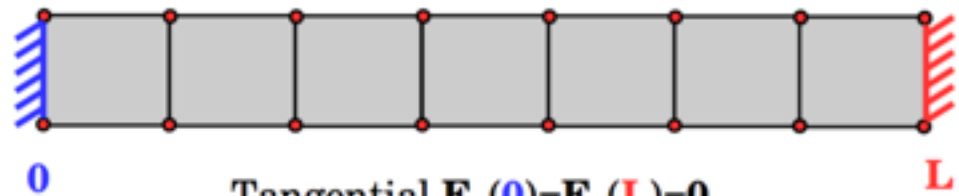
Outgoing BCs: match an outgoing wave to E, B fields at boundary (Lindman 1975).

Field boundary conditions: a few examples

Periodic



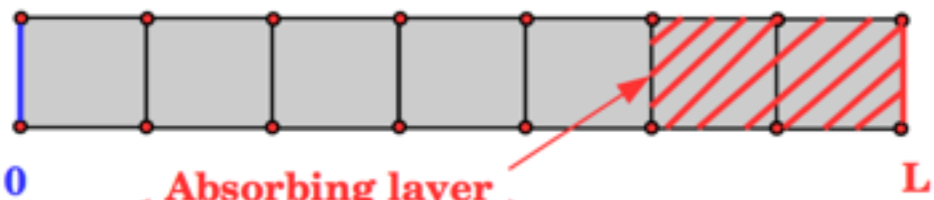
Perfectly conducting walls



Tangential $\mathbf{E}_T(0)=\mathbf{E}_T(L)=0$

Perpendicular $\mathbf{B}_\perp(0)=\mathbf{B}_\perp(L)=0$

Absorbing layer (open boundary)



$$\frac{\partial \boldsymbol{E}}{\partial t}+\sigma \boldsymbol{E}=c\nabla\times\boldsymbol{B}-4\pi\boldsymbol{J} \qquad \frac{\partial \boldsymbol{B}}{\partial t}+\sigma^{*}\boldsymbol{B}=-c\nabla\times\boldsymbol{E}$$

Multi-D generalization : **Perfectly Matched Layer** (see *Bérenger 1994-1996*)

Electromagnetic codes

Boundary conditions

Perfectly matched layer (Berenger 1994) -- works like absorbing material with different conductivity for E and B fields)

Moving window: simulation can fly at c to follow a fast beam. Outgoing plasma requires no conditions.

Injection: particles can be injected from boundary, or created in pairs throughout the domain. We implemented moving injectors and expanding domains for shock problems.

Parallelization

We use domain decomposition with ghost zones that are communicated via MPI. In 3D we decompose in slabs in y-z plane, so all x-s are on each processor (useful for shocks).

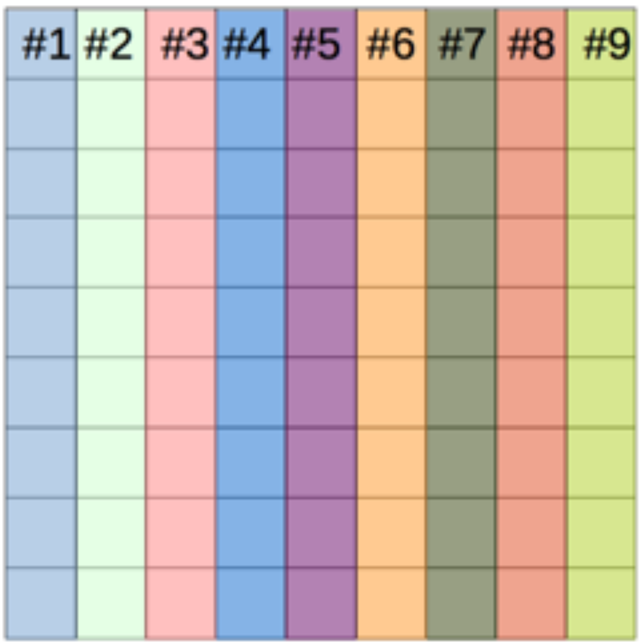
Parallelization: Domain decomposition

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

Applicable to an arbitrary number of CPUs

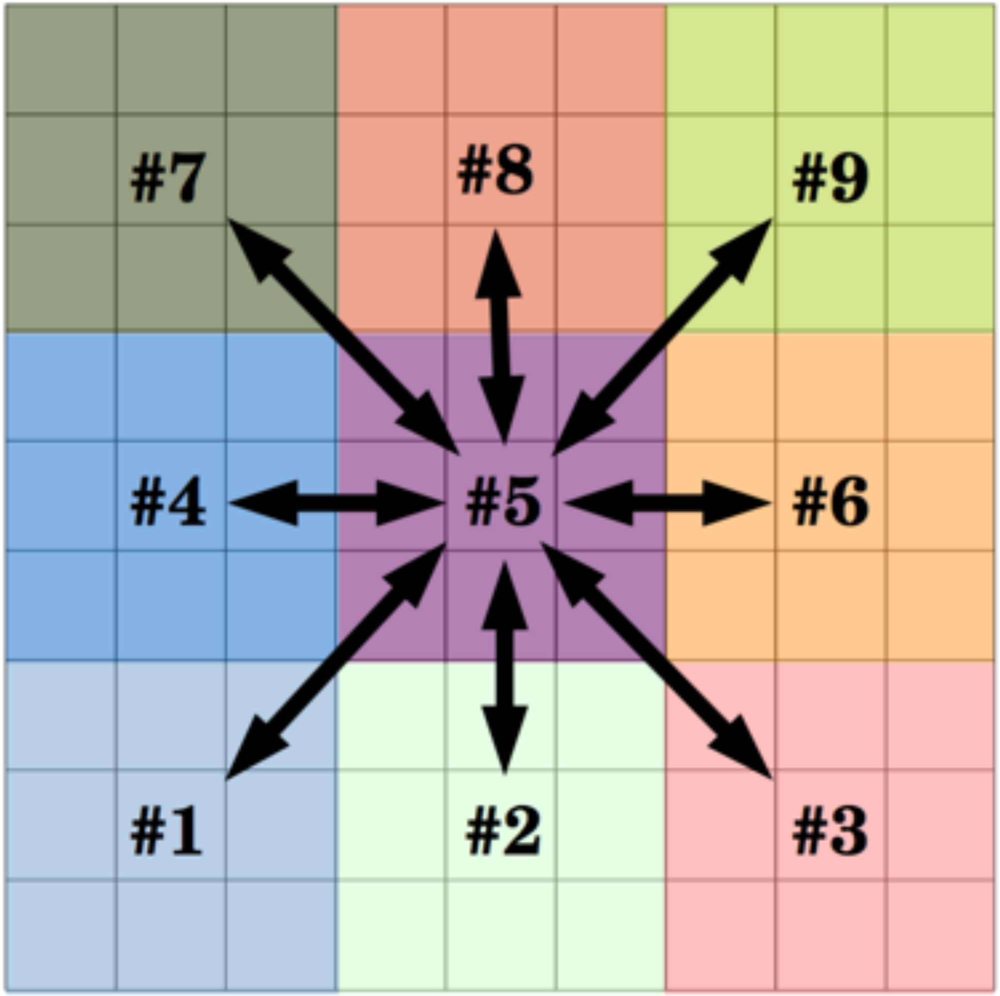
Choice decomposition depends on the problem

Parallelization: Domain decomposition

MPI Communications

1D: Up to **2 / CPU** **2D:** Up to **8 / CPU** **3D:** Up to **26 / CPU**

Example: 2D decomposition
Communications of CPU #5

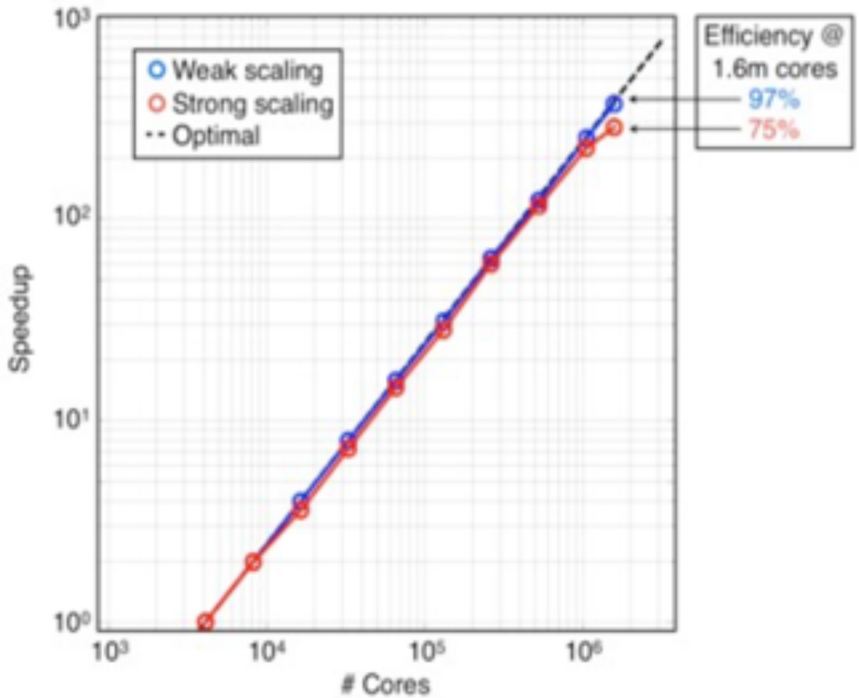


PIC codes scale well to large number of CPUs

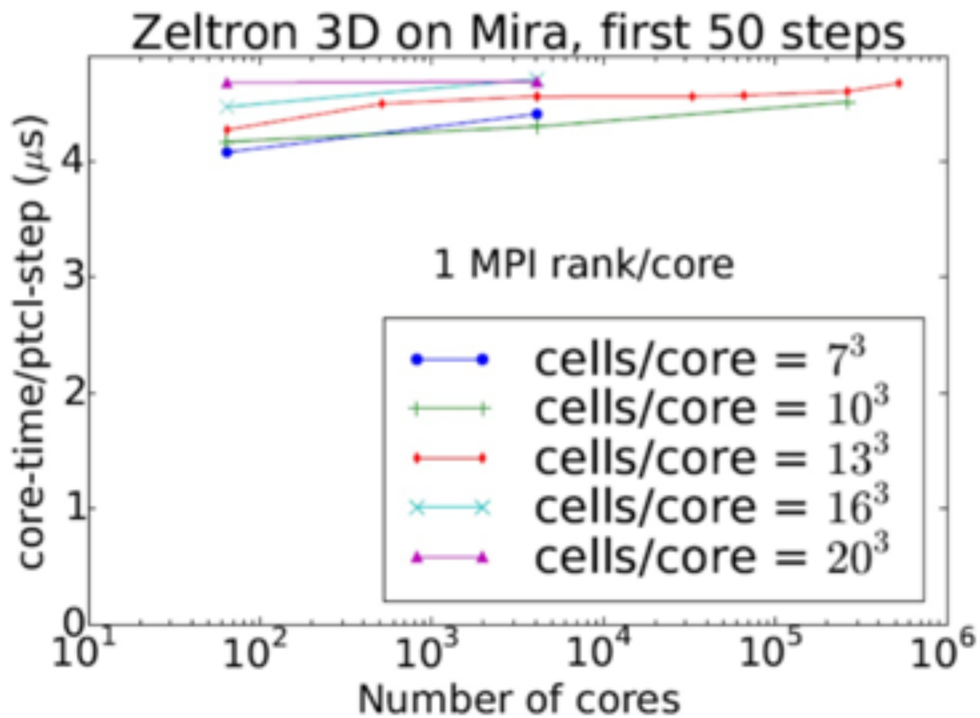
The era of **High-Performance Computing!** Today ~> **10^6 CPUs**

See http://www.top500.org/

OSIRIS Code



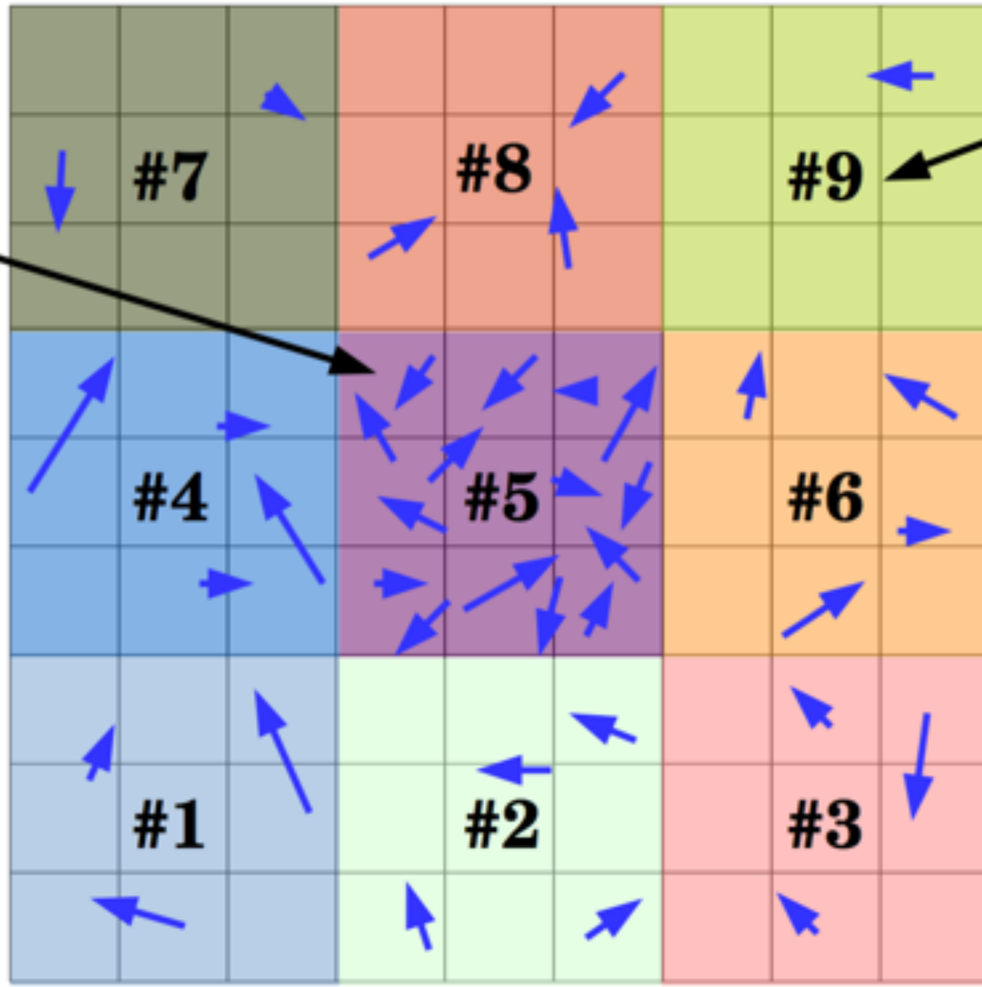
Zeltron Code



Load balancing issues

Computing time (without communications): ~ 90% **particles**, ~10% **fields**

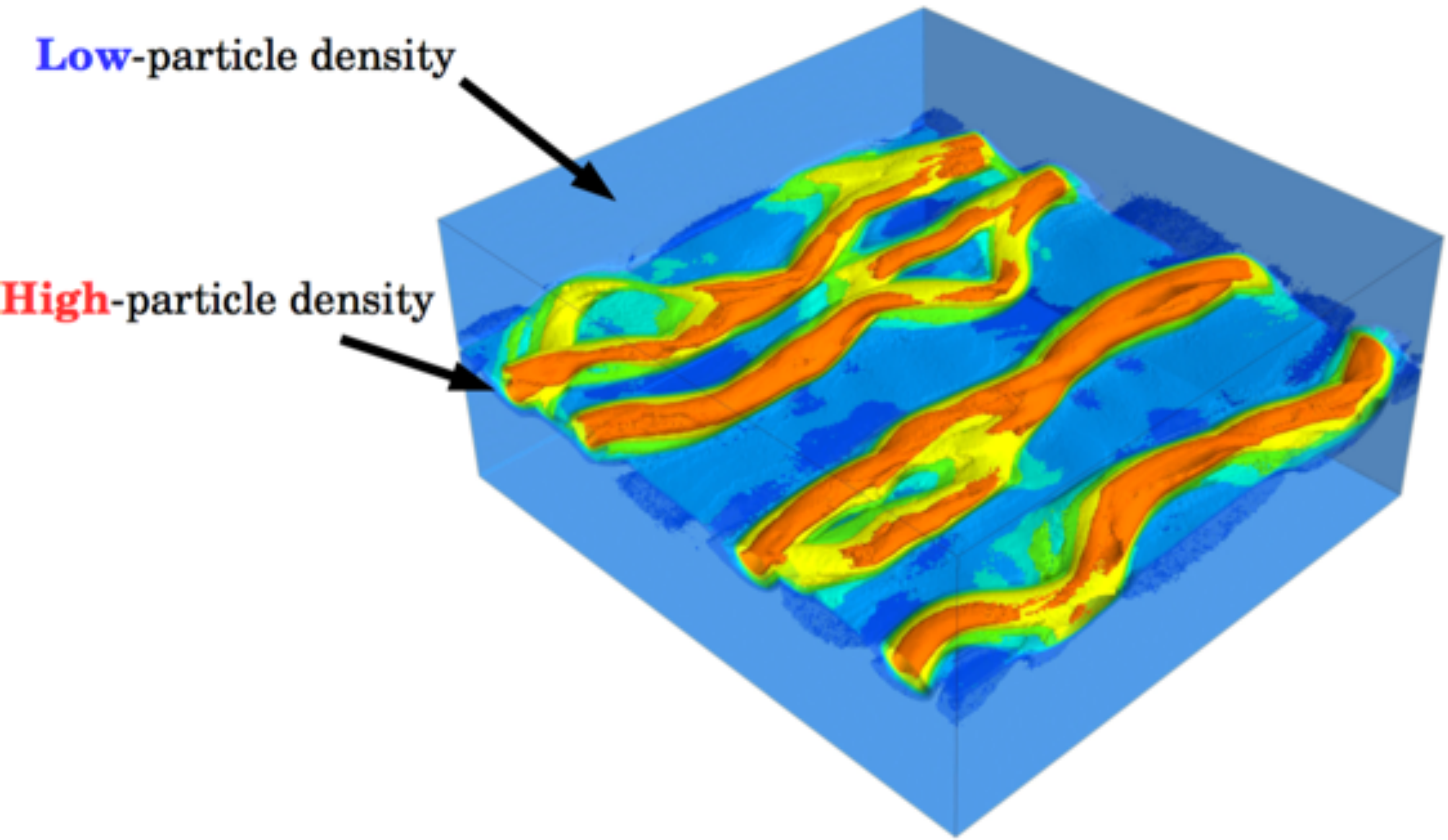
Many particles **Processor #5** is **slowing down** all the others



Few particles **Processor #9** is **waiting** for all the others

A specific example: a reconnecting layer

Density contrast ~>**10!**



Some solutions:

- Appropriate domain decomposition
- Dynamical changes of the decomposition
- Varying particle weights
- Hybrid code: MPI-OpenMP

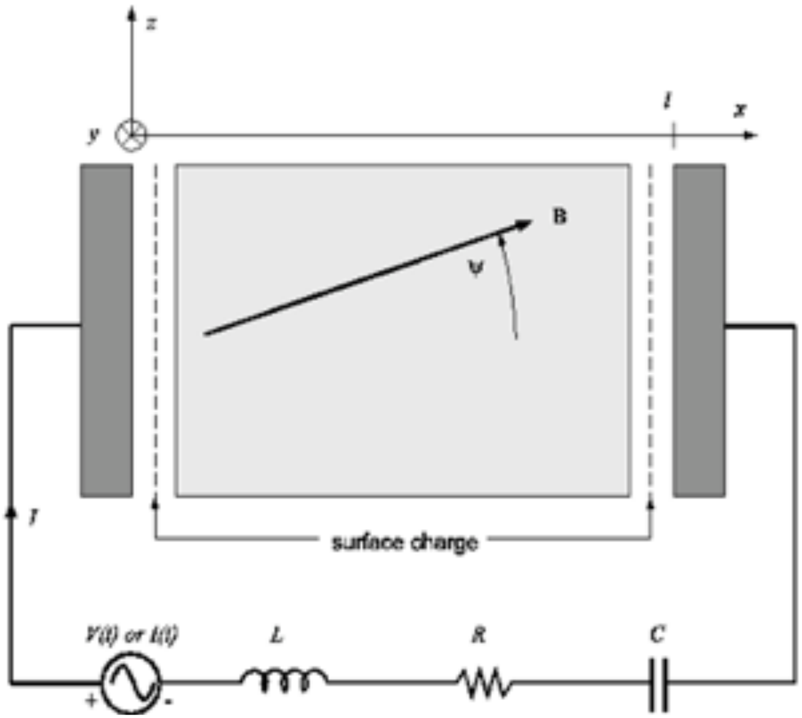
...

Public codes

http://ptsg.egr.msu.edu/

Download the software now.

- XES1
- XPDP1
- XPDC1
- XPDS1
- XPDP2
- XOOPIC
- XIBC
- XGRAFIX



Our most recent, popular and well kept up codes are on bounded plasma, plasma device codes XPDP1, XPDC1, XPDS1, and XPDP2. The P, C, and S mean planar, cylindrical, or spherical bounding electrodes; the 1 means 1d 3v and the 2 means 2d 3v. These are electrostatic, may have an applied magnetic field, use many particles (like hundreds to millions), particle-in-cell (PIC), and allow for collisions between the charged particles (electrons and ions, + or -) and the background neutrals (PCC-MCC). The electrodes are connected by an external series R, L, C, source circuit, solved by Kirchhoff's laws simultaneously with the internal plasma solution (Poisson's equation), The source may be V(t) or I(t), may include a ramp-up (in time). XPDP2 is planar in x, periodic in y or fully bounded in (x,y), driven by one or two sources.(For detailed information, click here)

Public and not so public codes

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.

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

Contents

Notes on PIC

No “subscale” physics – resolve the smallest scales! Converse is expense

Usually deal with non-clumped flows, hence AMR is not needed. Some exceptions -- reconnection simulations.

FDTD conserves divergence of B to machine precision.

PIC issues:

- Particle discretization error
- Smoothing error (finite size particles)
- Statistical noise (granular force)
- Grid aliasing (grid assignment)
- Deterioration of quadrature in time integration
- Short-range forces (collisions) neglected

- Analysis of large-scale simulations is nontrivial

but the alternative is 6D Vlasov integration...

Outlook

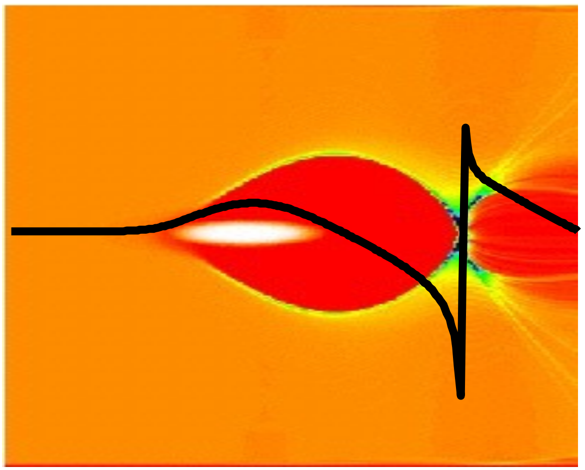
PIC is a versatile robust tool for self-consistent solution of plasma physics.

- Electrostatic method is well understood, and analytical theory of numerical plasma exists.
- Electromagnetic model is more diverse, and many alternative formulations exist. Multidimensional theory of the simulation is not as well developed.
- Implicit methods are now common for large timestep solutions.
- Long term stability is an issue for largest runs.
- In astrophysics PIC has the potential to answer the most fundamental theoretical questions: particle acceleration, viability of two-temperature plasmas, dissipation of turbulence.

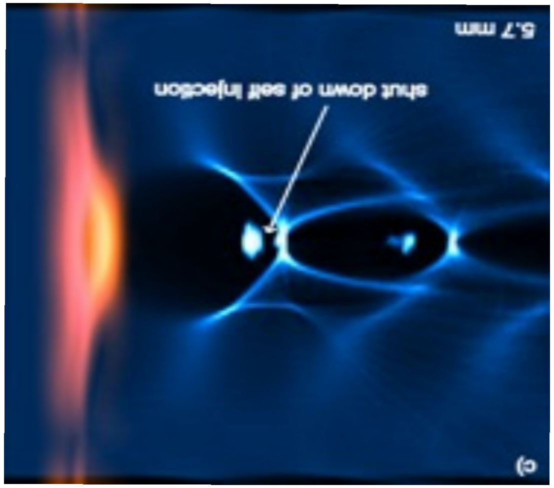
Applications

Laser-plasma interaction and plasma based accelerators

Laser driver:



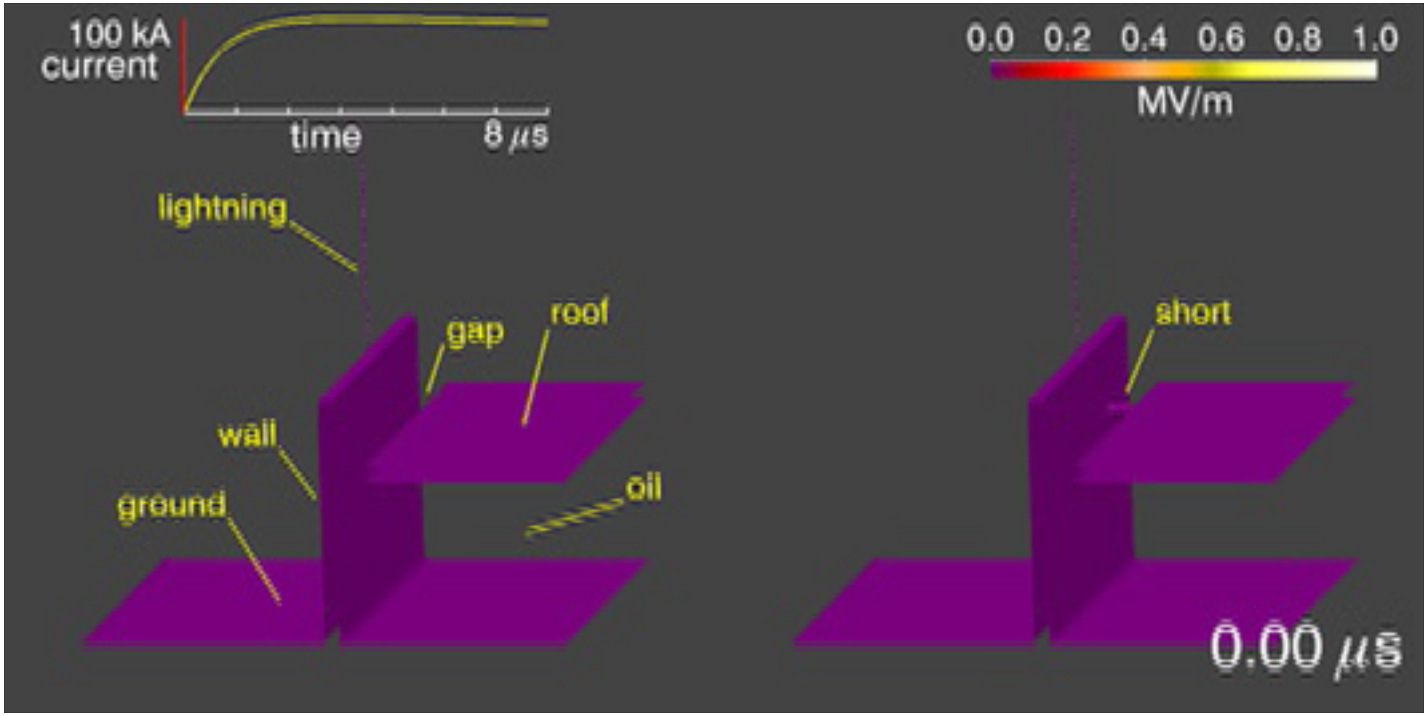
Beam driver:



Engineering:

Gas discharges, plasma processing, film deposition. PIC with Monte-Carlo collisions and external circuit driving.

Lightning-oil tank interaction!



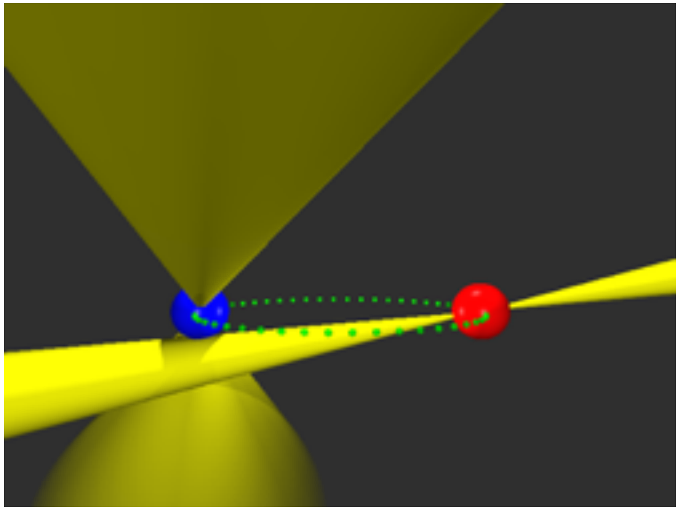
Applications

Astrophysics:

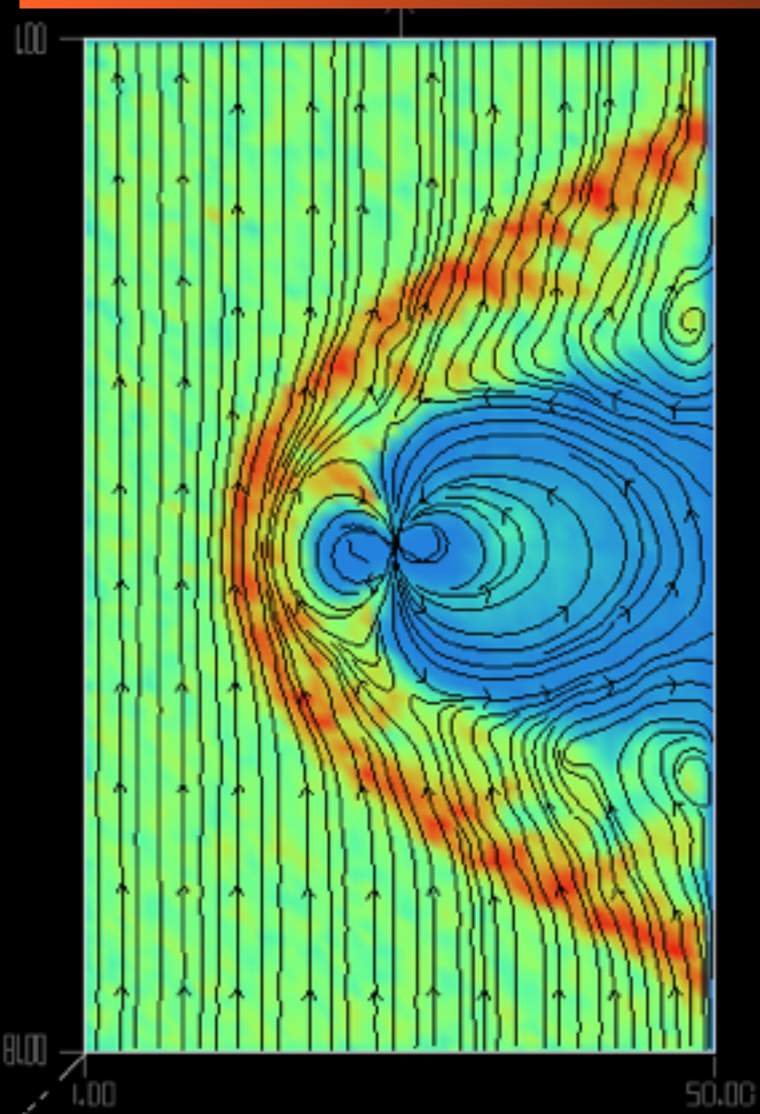
Collisionless shocks (solar wind, interstellar medium, relativistic jets), wind-magnetosphere interactions, pulsar magnetospheres.

Rapid reconnection, particle acceleration.

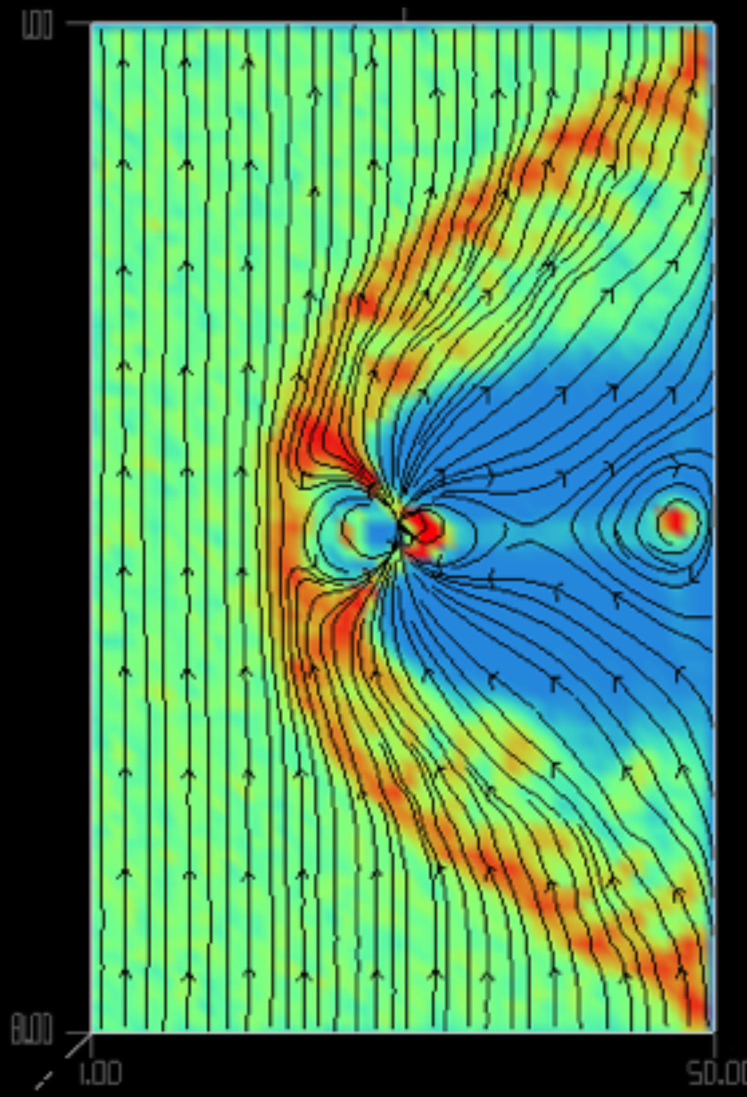
Case study: Wind-magnetosphere interaction in double pulsar binary J0737. Attempt to simulate macroscopic system with PIC. Possible if the size of the system is > 50 skindepths.



Shock and magnetosheath of pulsar B



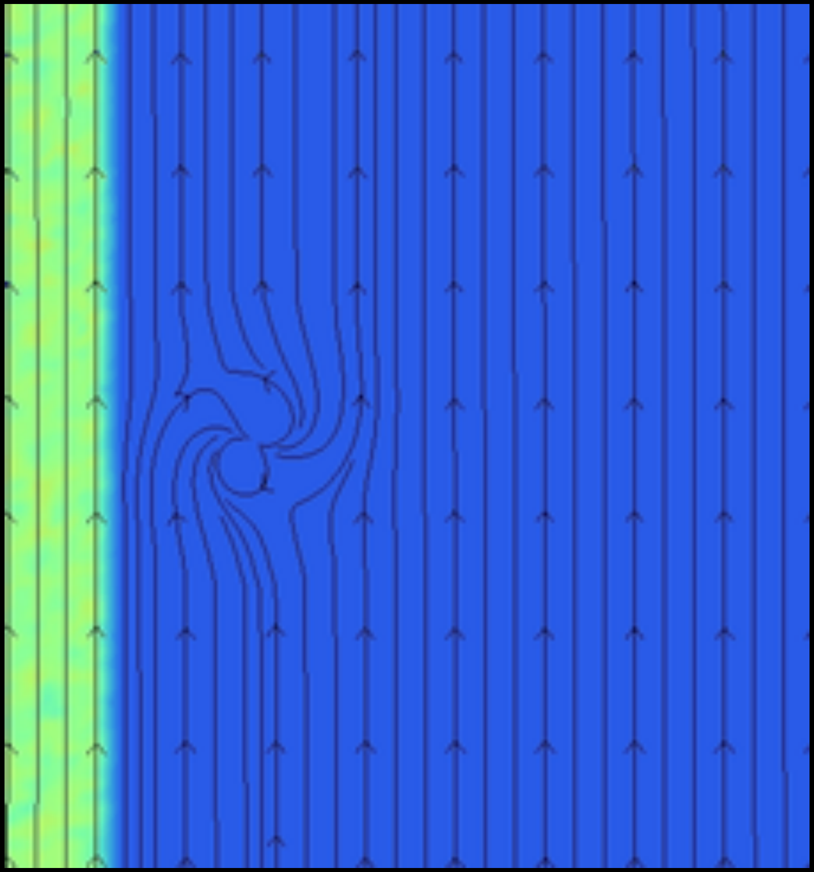
No “dayside” reconnection



With “dayside” reconnection

Similar to the interaction between Earth magnetosphere and solar wind.

Shock and magnetosheath of pulsar B: effects of rotation



Shock modulated at 2Ω

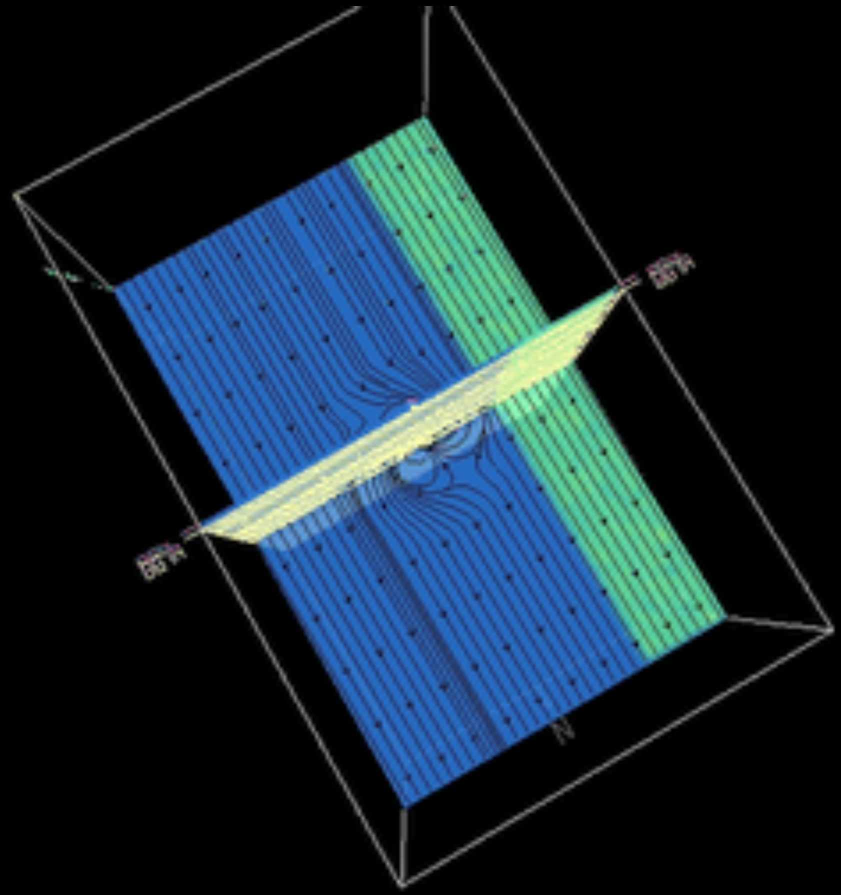
Reconnection once per period

Cusp filling on downwind side

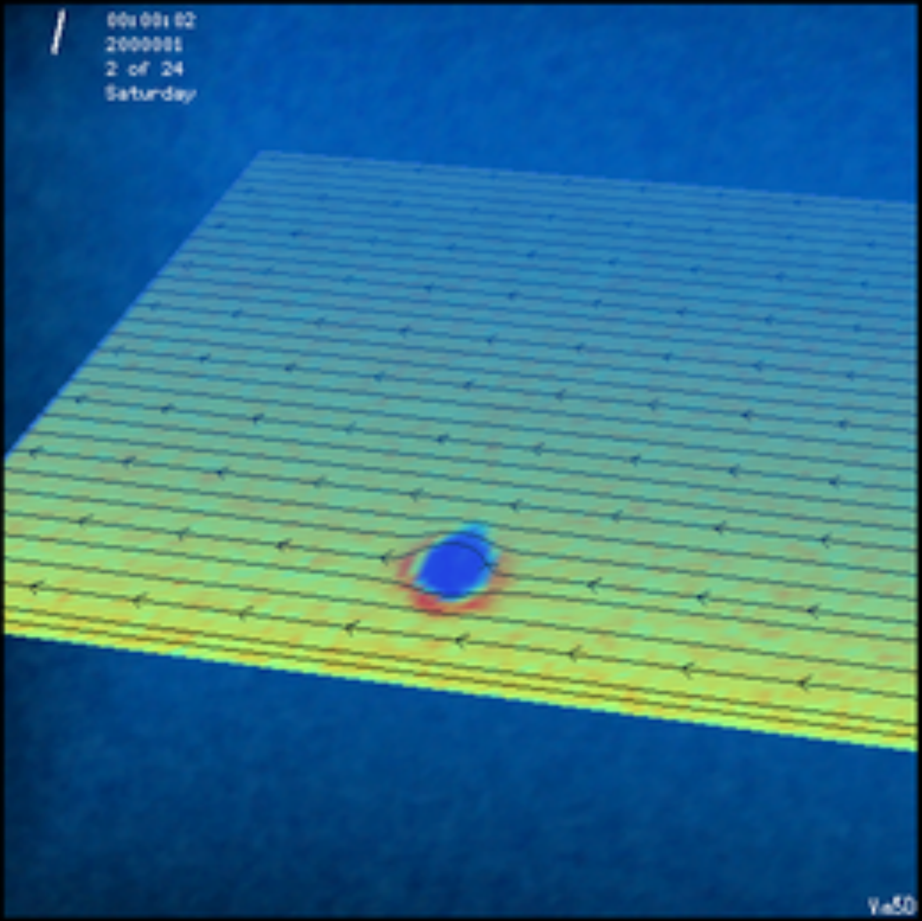
Density asymmetries

R_m~50000 km

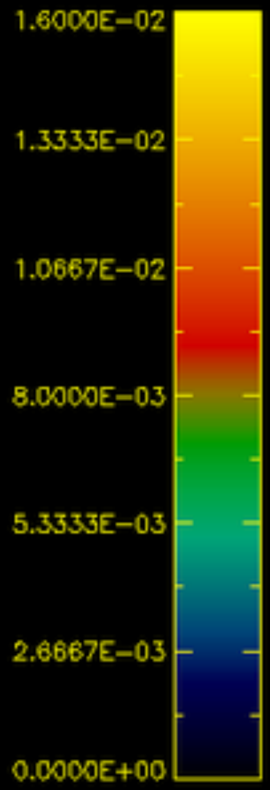
3D magnetosphere



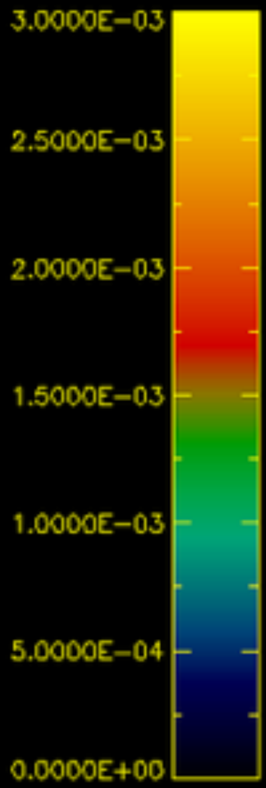
3D magnetosphere



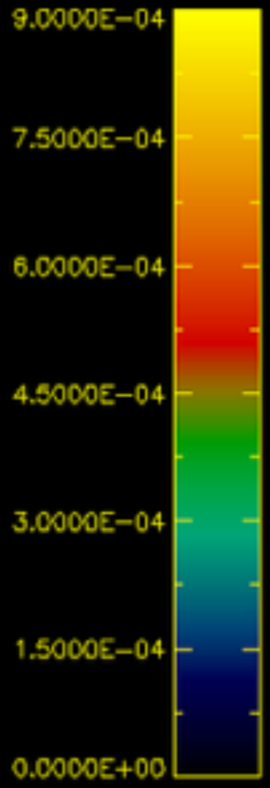
Counterstreaming instabilities

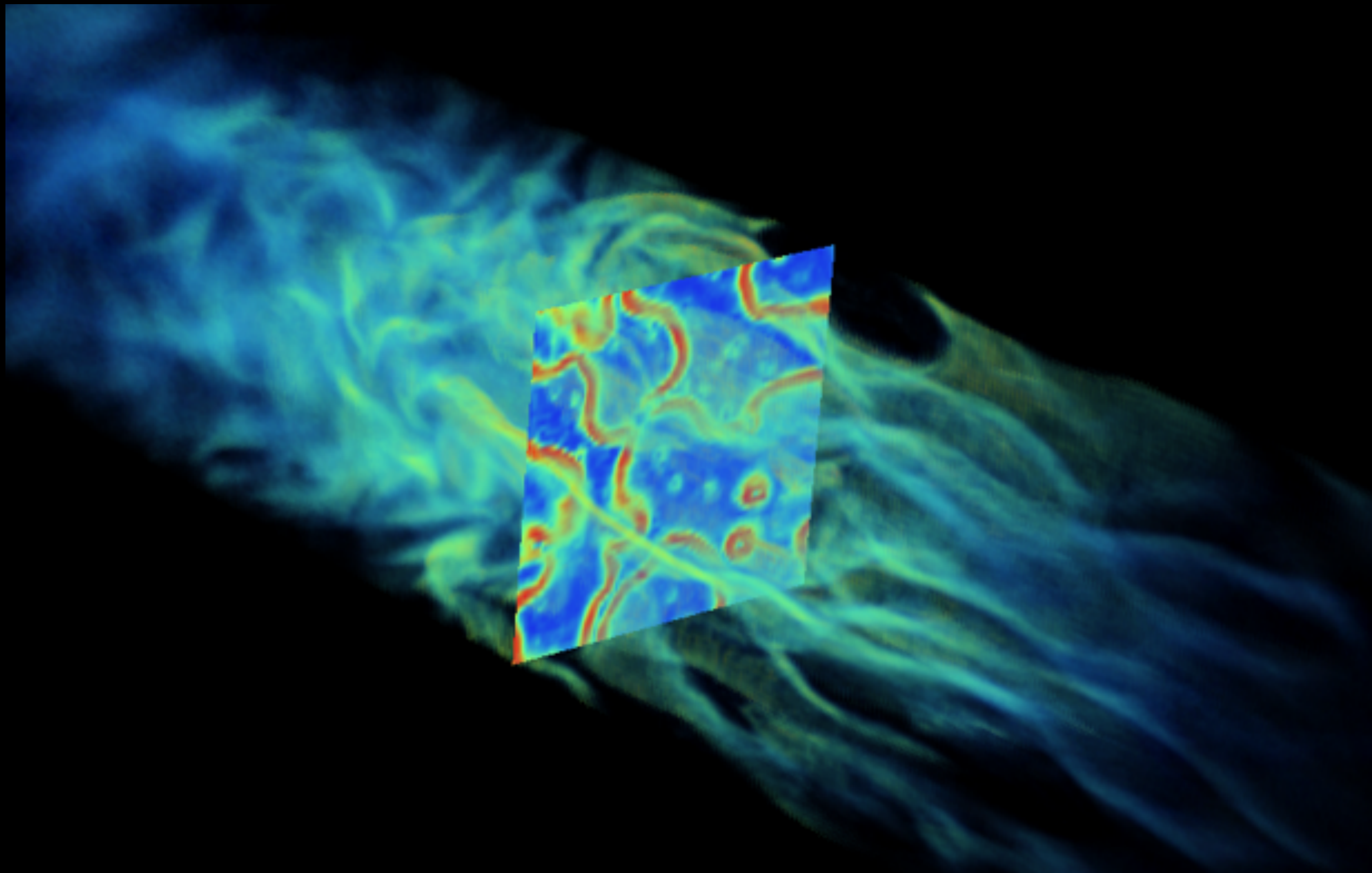


Counterstreaming instabilities



Counterstreaming instabilities

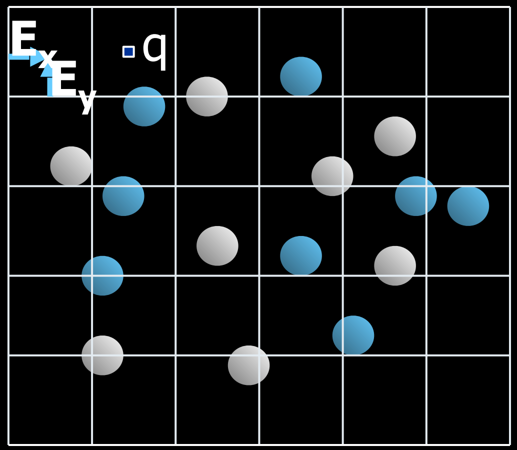




Simulations of Relativistic Shocks

Anatoly Spitkovsky (Princeton)

Numerical simulation of collisionless shocks



Particle-in-cell method:

- *Collect currents at the cell edges*
- *Solve fields on the mesh (Maxwell's eqs)*
- *Interpolate fields to particles positions*
- *Move particles under Lorentz force*

Code "TRISTAN-MP":

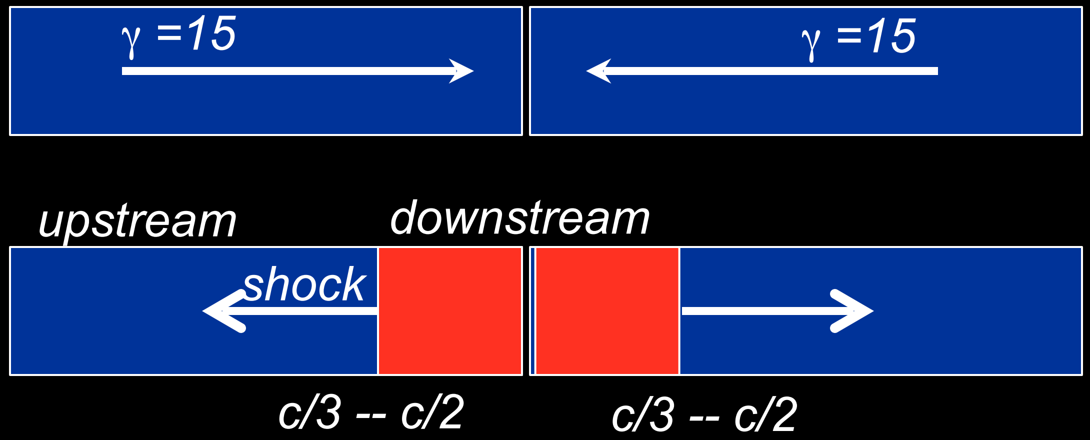
- *3D (and 2D) cartesian electromagnetic particle-in-cell code*
- *Radiation BCs; moving window*
- *Charge-conservative current deposition (no Poisson eq)*
- *Filtering of current data*
- *Fully parallelized (512proc+) domain decomposition*
- *Routinely work with upto 10 billion+ particles*

Large simulations needed for interesting steady states!!!
In 3D grids are up to 10000x1024x1024 cells
In 2D grids are up to 150000x4000 cells

Simulation setup:

Relativistic $e^{\pm}$ or e^{-}- ion wind (γ =15) with B field ($\sigma = \omega_c^2 / \omega_p^2 = B^2/(4\pi n \gamma m c^2)$ = 0-10)
Reflecting wall (particles and fields)
Upstream c/ω_p=10 cells, c/ω_c>5 cells;

Problem setup



"Shock" is a jump in density & velocity

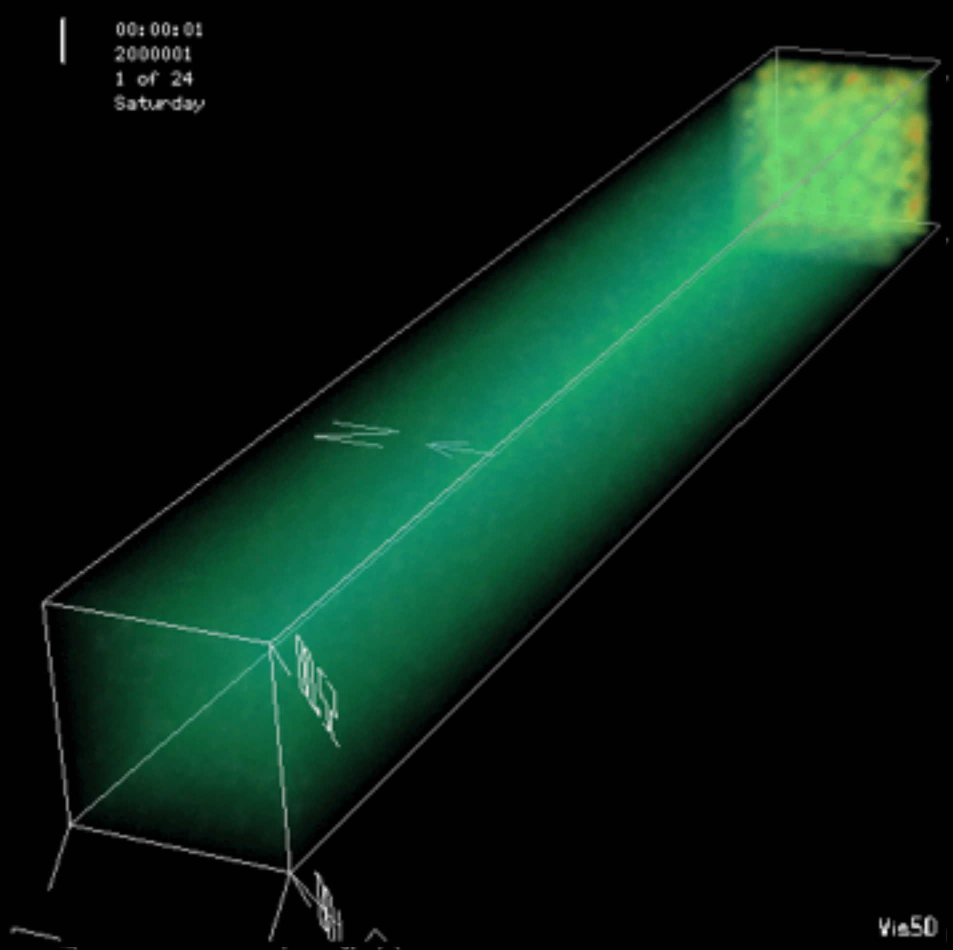
Use reflecting wall to initialize a shock

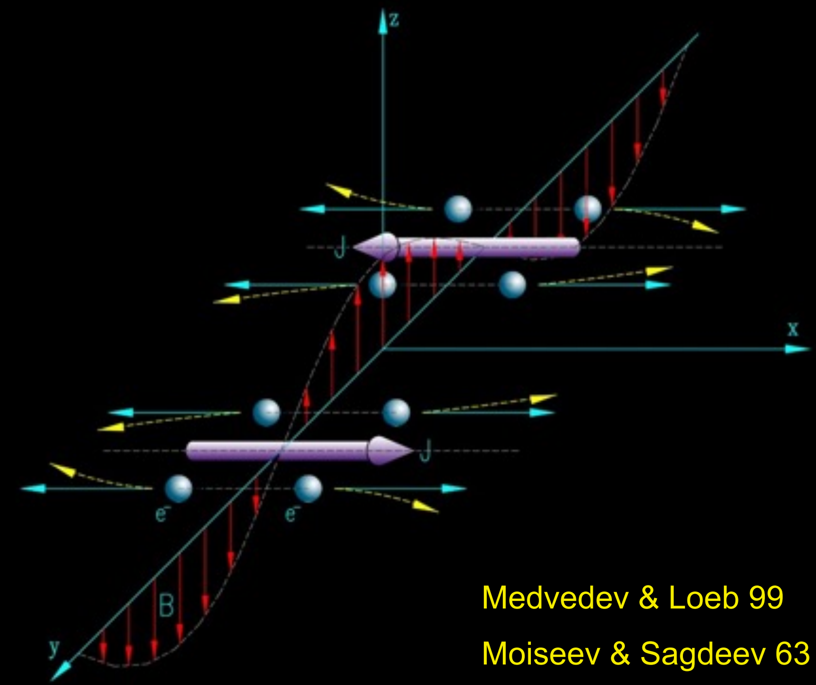
Simulation is in the downstream frame. If we understand how shocks work in this simple frame, we can boost the result to any frame to construct astrophysically interesting models. Disadvantage -- upstream flow has to move over the grid -- potential long term instabilities.

Unmagnetized pair shock

Why does a shock exist?

Particles are slowed down either by instability (two-stream-like) or by magnetic reflection. Electrostatic reflection is important for nonrelativistic shocks and when ions are present.





Medvedev & Loeb 99

Moiseev & Sagdeev 63

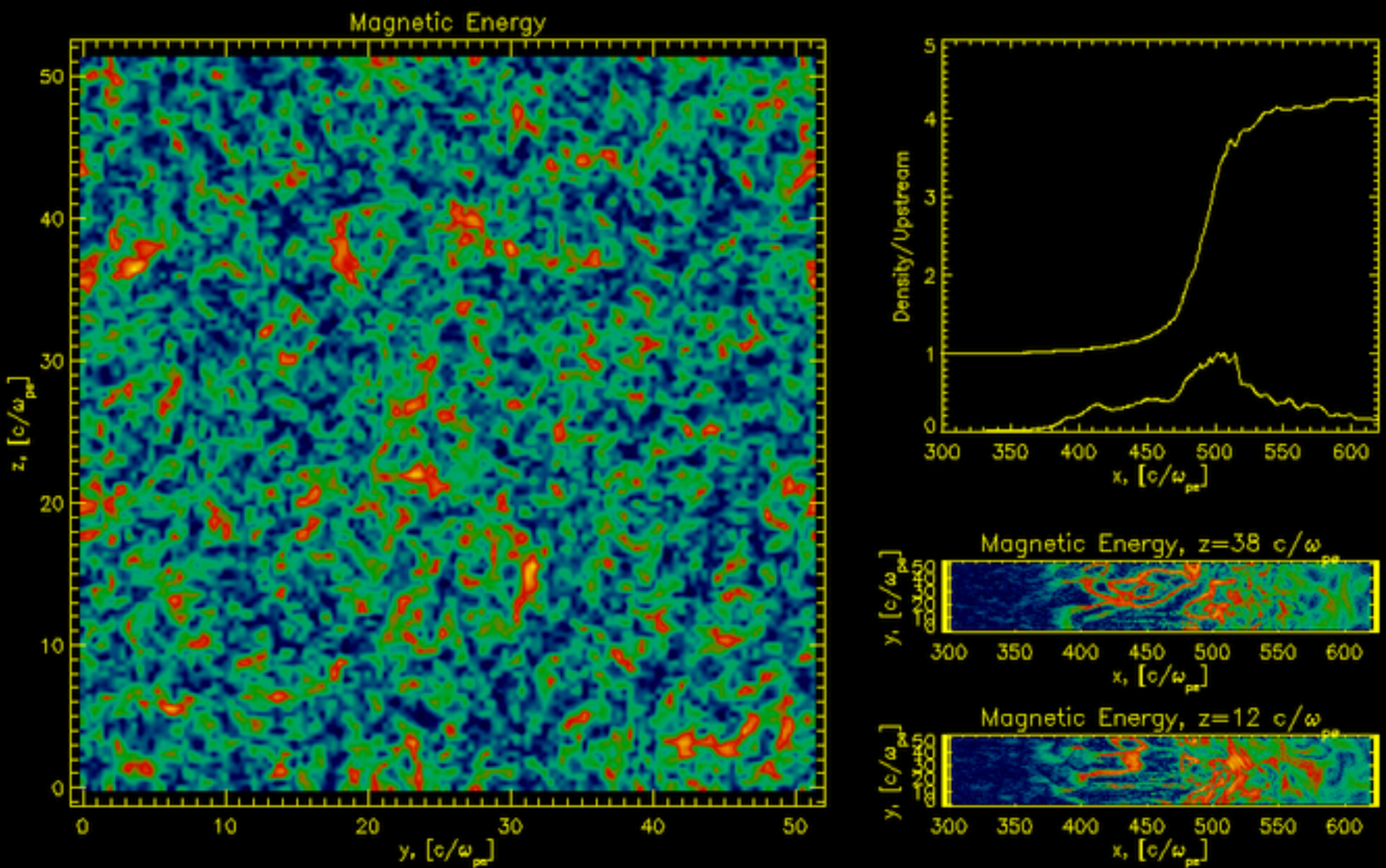
Weibel instability *(Weibel 1959)*

Spatial growth scale c/ω_p; timescale $10/\omega_p$

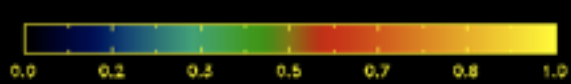
$$L \approx c/\omega_{pe} = 10 \text{ km} \sqrt{\gamma / n_0[\text{cm}^{-3}]}$$

$$T \approx 1/\omega_p = 30 \ \mu\text{s} \sqrt{\gamma / n_0[\text{cm}^{-3}]}$$

3D shock structure: long term

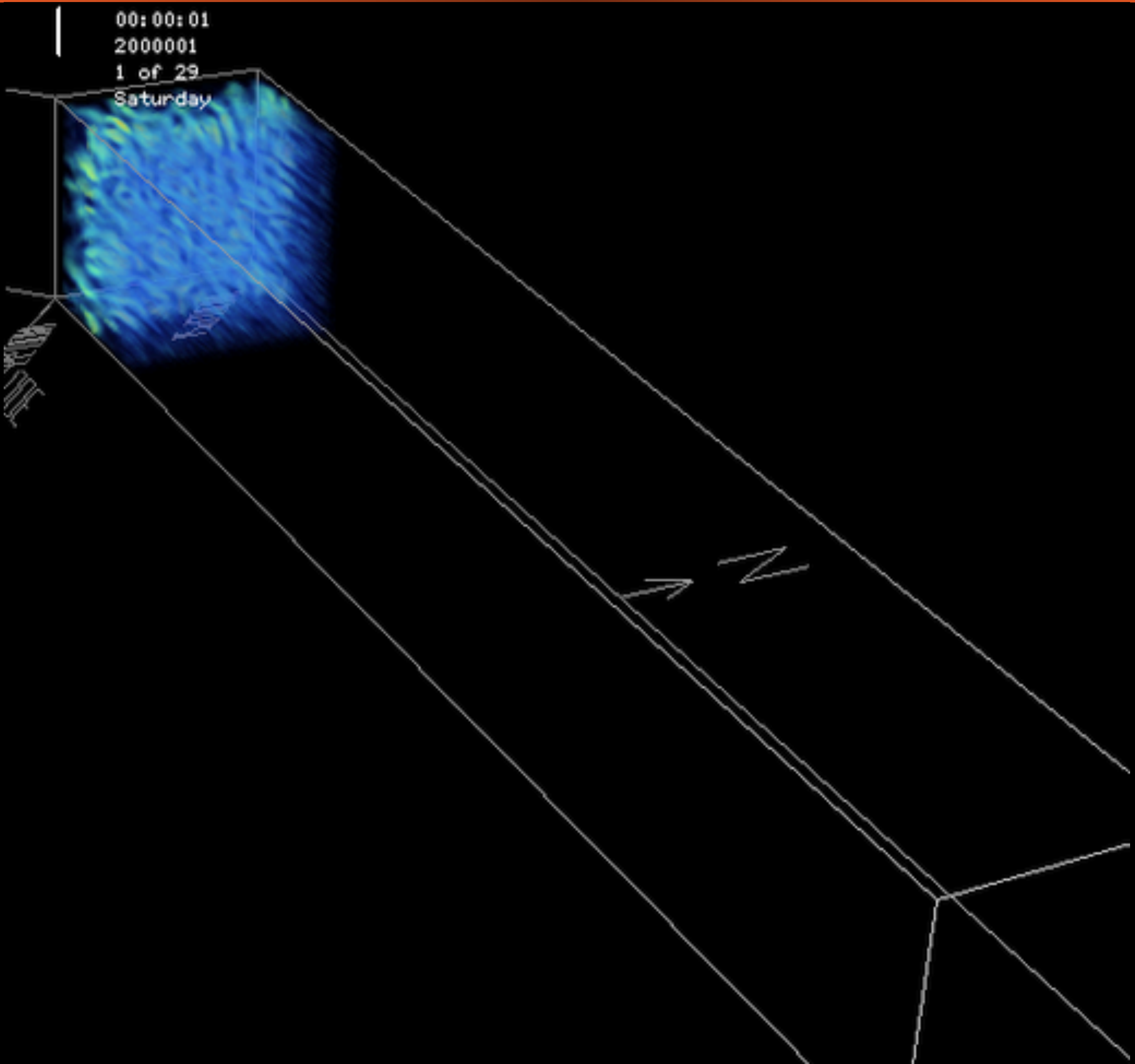


50x50x1500 skindepths. Current merging (like currents attract).



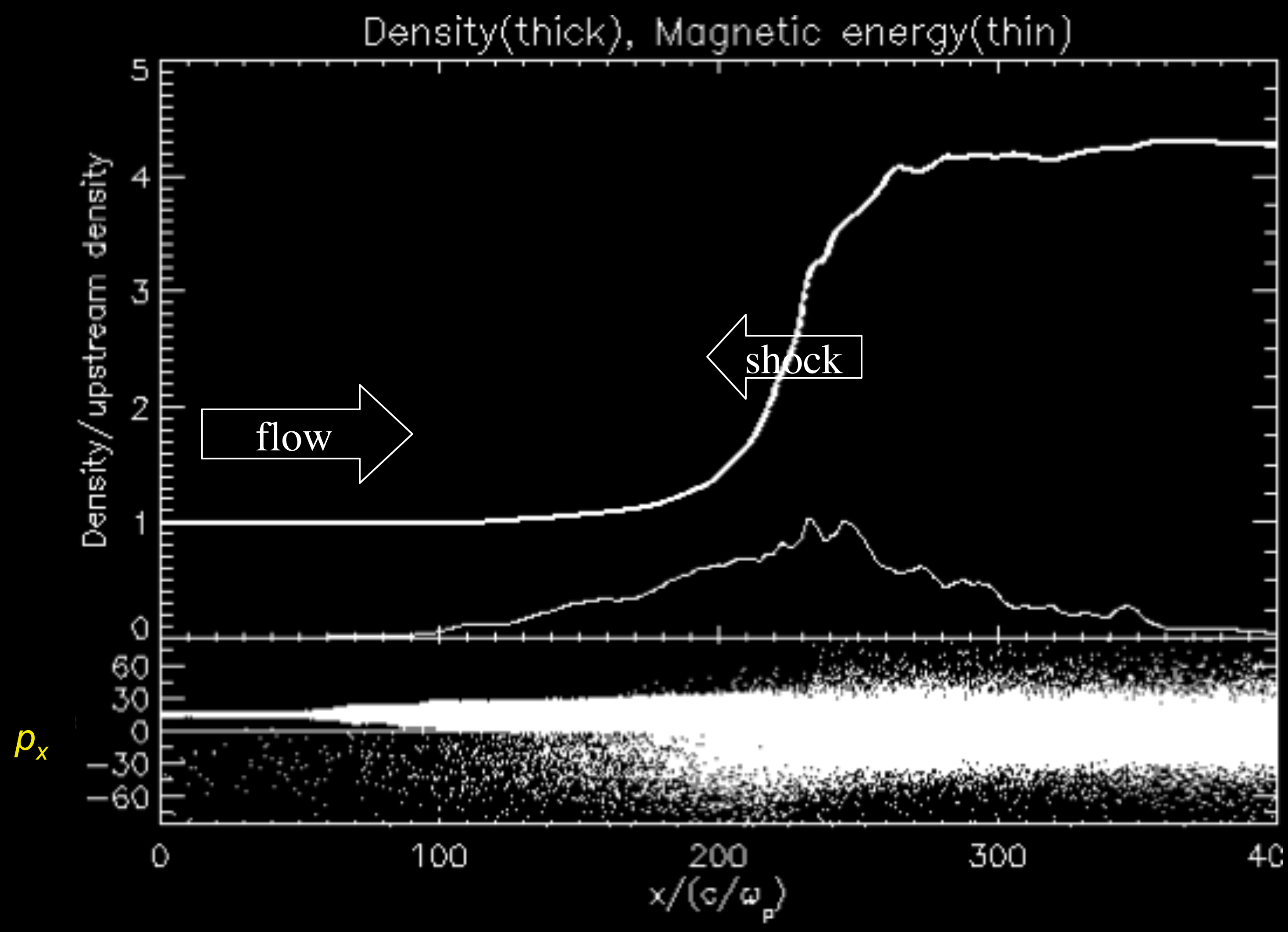
Secondary Weibel instability stops the bulk of the plasma. Pinching leads to randomization.

3D unmagnetized pair shock: magnetic energy



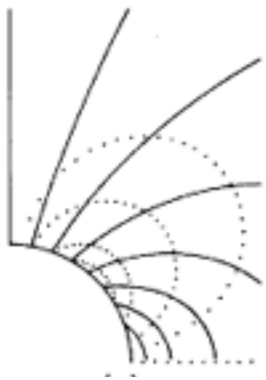
Unmagnetized pair shock

Steady state counterstreaming leads to self-replicating shock structure



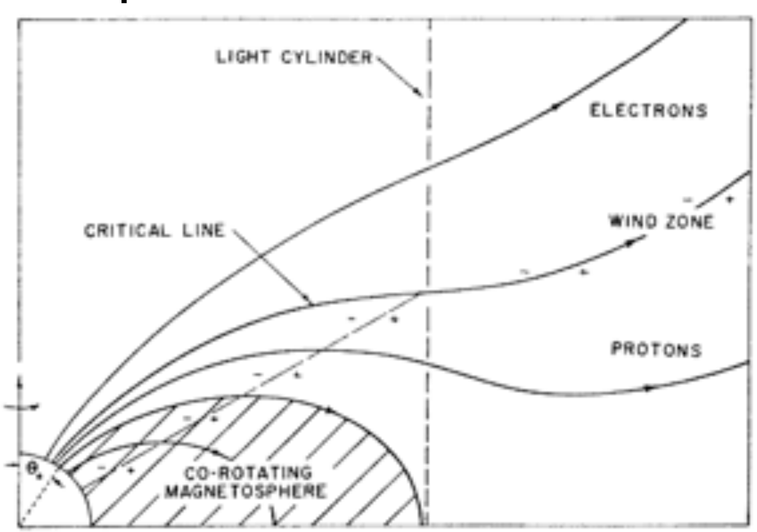
Astrophysics:

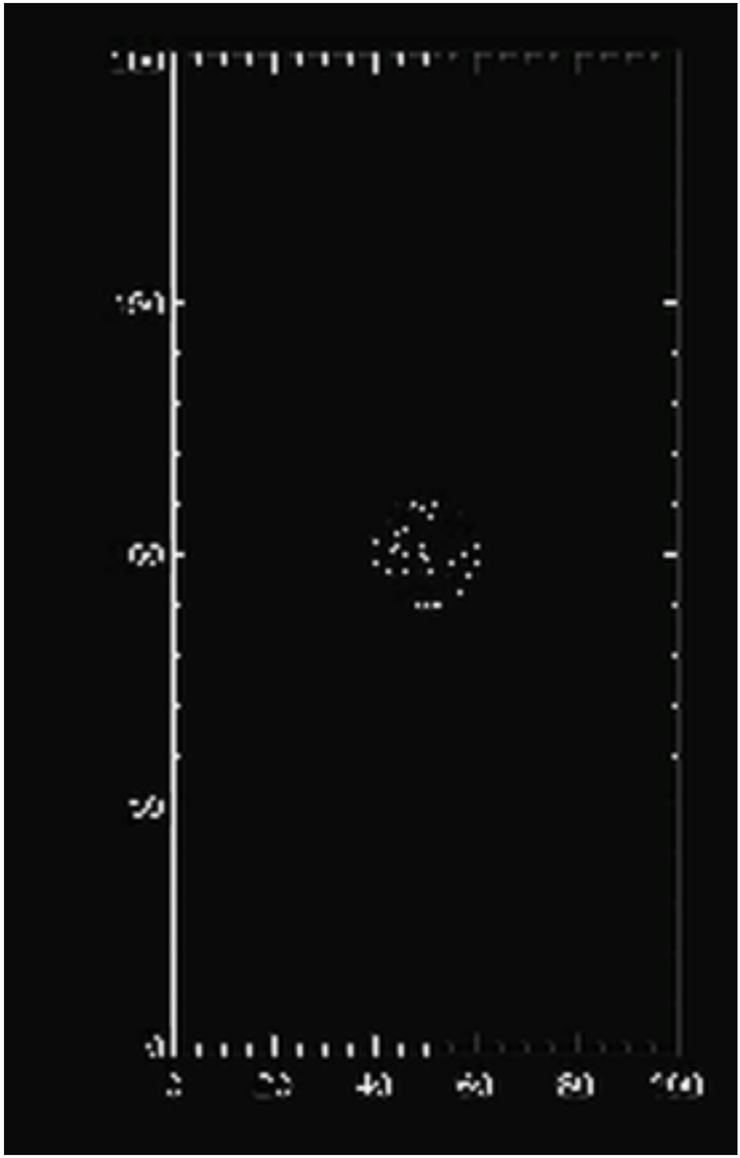
Nonneutral plasma physics in pulsar magnetospheres



Electric field on the surface
extracts charges. Does magnetosphere form?

Expect to see this:

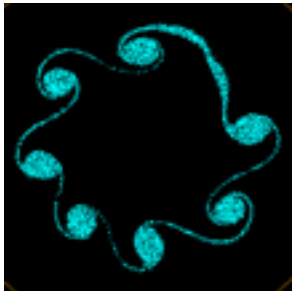


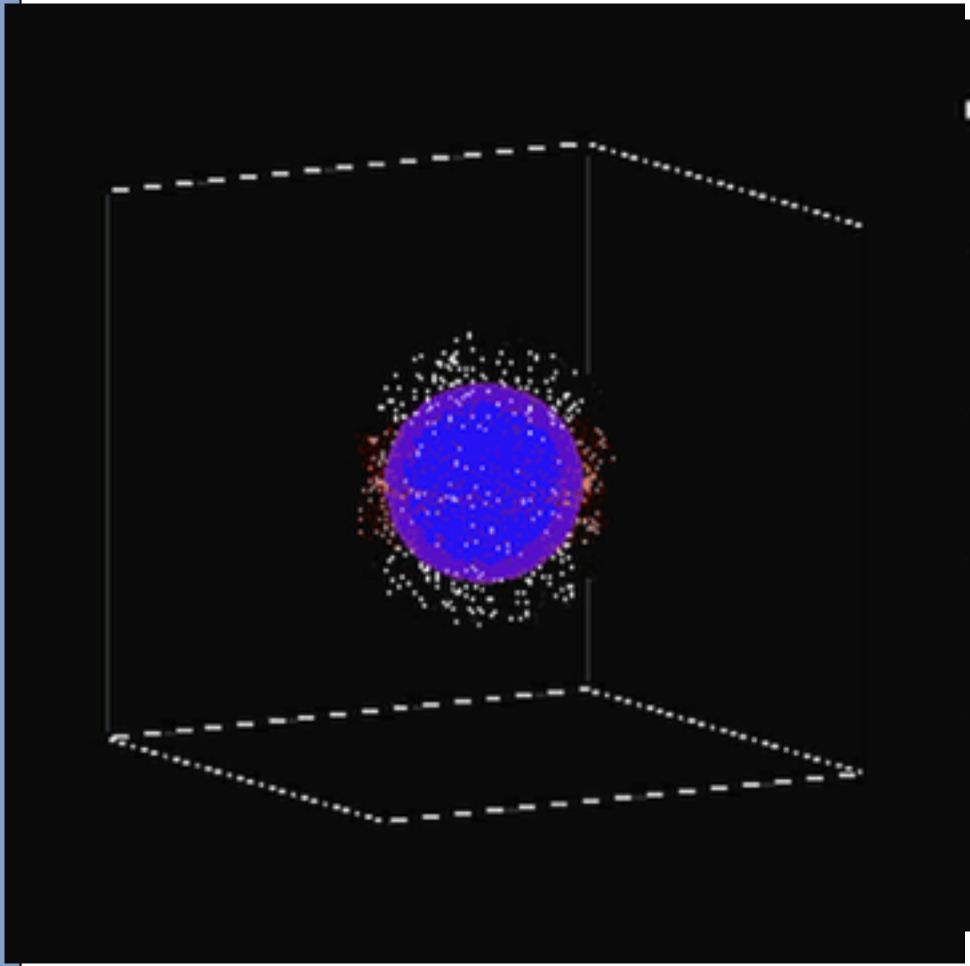


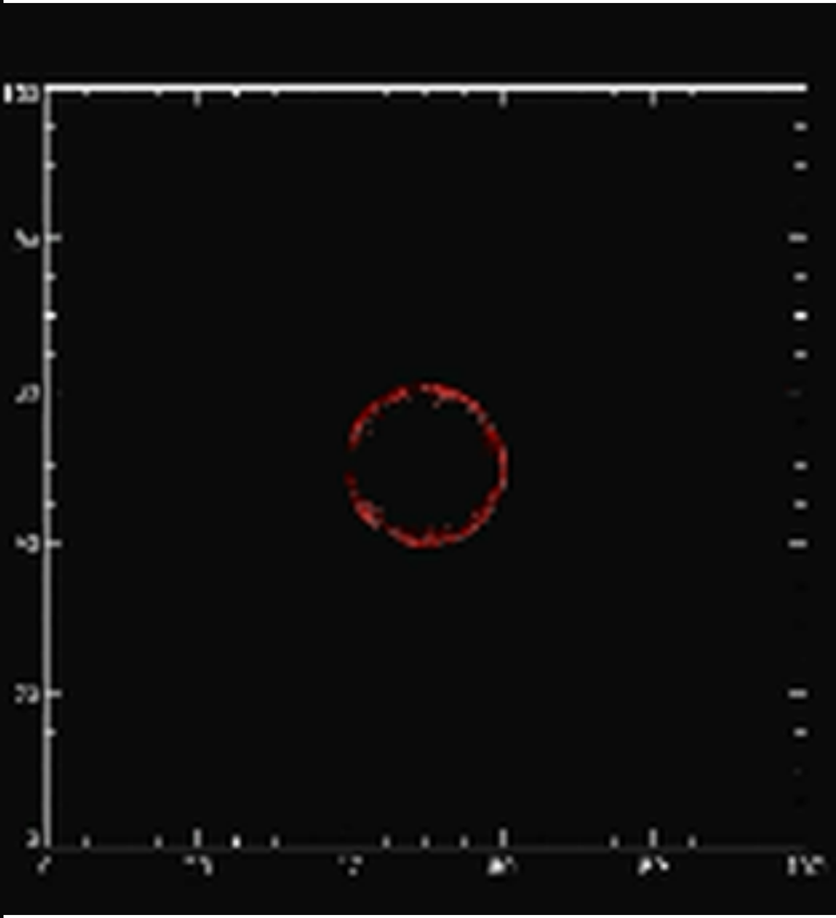
Applications

Astrophysics:

Nonneutral plasma physics in pulsar magnetospheres. Diocotron instability



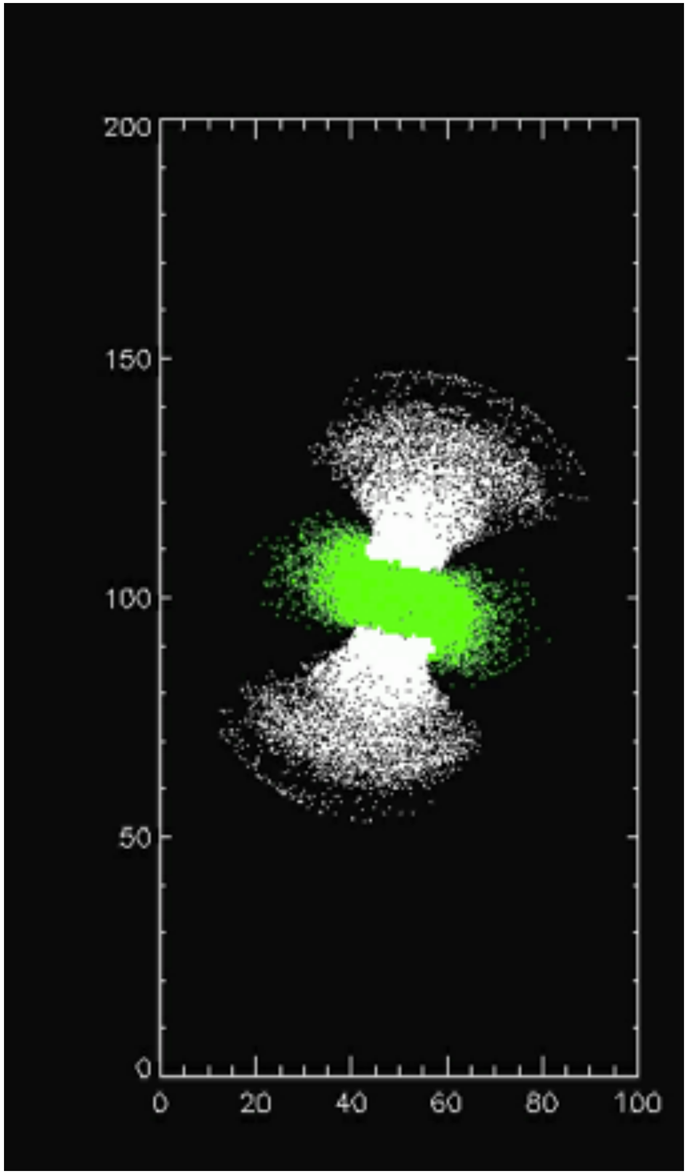




Applications

Astrophysics:

Nonneutral plasma physics in pulsar magnetospheres. Diocotron instability



Outlook

PIC is a versatile robust tool for self-consistent solution of plasma physics.

- Electrostatic method is well understood, and analytical theory of numerical plasma exists.
- Electromagnetic model is more diverse, and many alternative formulations exist. Multidimensional theory of the simulation is not as well developed.
- Implicit methods are now common for large timestep solutions.
- Long term stability is an issue for largest runs.
- In astrophysics PIC has the potential to answer the most fundamental theoretical questions: particle acceleration, viability of two-temperature plasmas, dissipation of turbulence.