www.ino.it

INO-CNR
ISTITUTO
NAZIONALE DI
OTTICA

Introduction to *Kinetic* Simulation Methods for Collisionless Plasmas

Andrea MACCHI *

Seminario su
"Plasmi caldi:
ricerca di base
e applicazioni"
Politecnico di Milano,
20/01/2011

INO Research Unit "Adriano Gozzini"
CNR Research Area, Pisa, Italy

*also at Dipartimento di Fisica "Enrico Fermi",
Largo Bruno Pontecorvo 3, 56127 Pisa, Italy
www.andreamacchi.eu

OUTLOOK

- Plasma simulations for beginners: a simple electrostatic model

- Kinetic, collisionless, electromagnetic plasmas:
  tackling the Vlasov-Maxwell system numerically

- "Eulerian" vs "Lagrangian" (PIC-Particle-In-Cell) approach

- Basics of the PIC method

- Snapshots from multi-dimensional PIC simulations

FOR ABSOLUTE BEGINNERS: SHEET MODEL

Dawson, “One-dimensional plasma model”, *Phys.Fluids* **5** (1962) 445

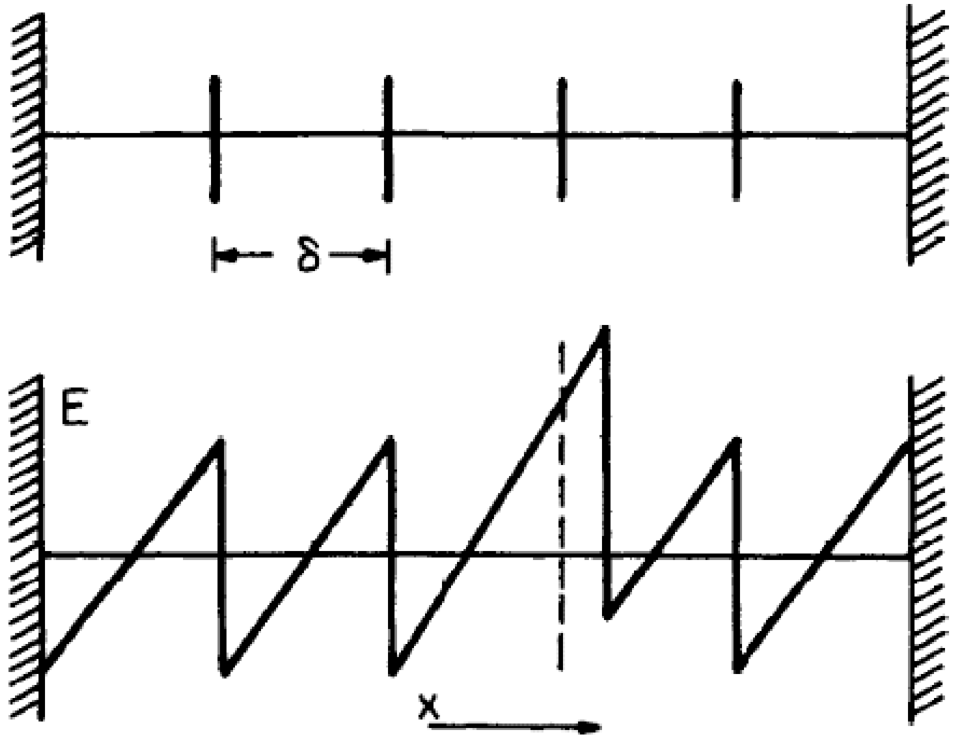


Figure 13-2a Original *Dawson* (1962) model, with thin electron sheets spaced $\delta = 1/n$ apart (in equilibrium) in a uniform positive ion background. The lower part shows $E(x)$ with one sheet displaced.

BASICS OF THE SHEET MODEL - I

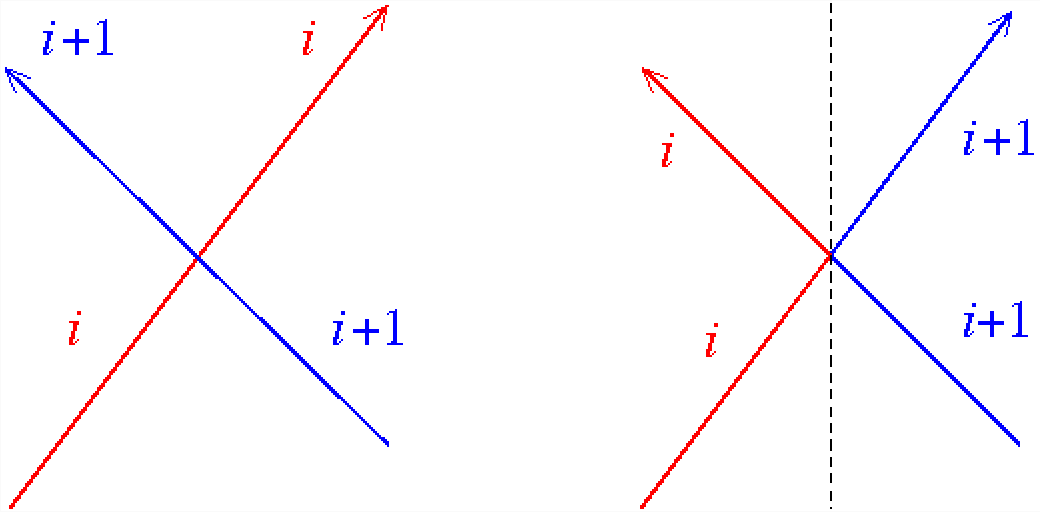
Motion of N electrons in 1D (so, "charge sheets"), fixed ions under the action of electrostatic field E_x (+ external force driver F_{ext})

$$
\begin{aligned}
X_i &= X_i(t)\,, \qquad i = 1,\ldots,N\,, \qquad X_i(0) = X_i^0 \\
\frac{d^2 X_i}{dt^2} &= -\frac{e}{m_e} E_x(X_i) + F_{\text{ext}} \\
E_x(X_i) &= \int_0^{X_i} 4\pi e(n_i - n_e)dx = 4\pi e n_i X_i - 4\pi \sum_{j<i} \sigma_j \\
&= 4\pi e n_i (X_i - X_i^0) \qquad (n_i \text{ uniform, } \textbf{no crossings})
\end{aligned}
$$

If $F_{\text{ext}}=0$ and no crossing occurs between the sheets, the latter oscillate around $x=X_i^0$ at the plasma frequency $\omega_p = \sqrt{4\pi e^2 n_i / m_e}$ (for a homogeneous plasma)

BASICS OF THE SHEET MODEL - II

Crossing of neighboring sheets can be modeled as an “exchange of velocity” equivalent to a remapping of the coordinate index: the field on a sheet due to other electron sheet is constant: $E_{\mathrm{el}}[X_i^0(t)]=E_{\mathrm{el}}(X_i^0)$



Using this trick + numerical integration of the equations of motion (with Runge-Kutta, Leapfrog, Verlet, ..., algorithms as preferred) yields an elementary plasma simulation code (can be generalized to inhomogenous plasma and/or external driver)

PLAYING WITH THE SHEET MODEL - I

The sheet model can be used for a first numerical insight into kinetic dynamics (plasma oscillations, Debye shielding, Landau damping, wake excitation and collisionless stopping)
[see Birdsall & Langdon, “*Plasma Physics via Computer Simulation*” (IOP, 1991) Chap.13, p.277-292]

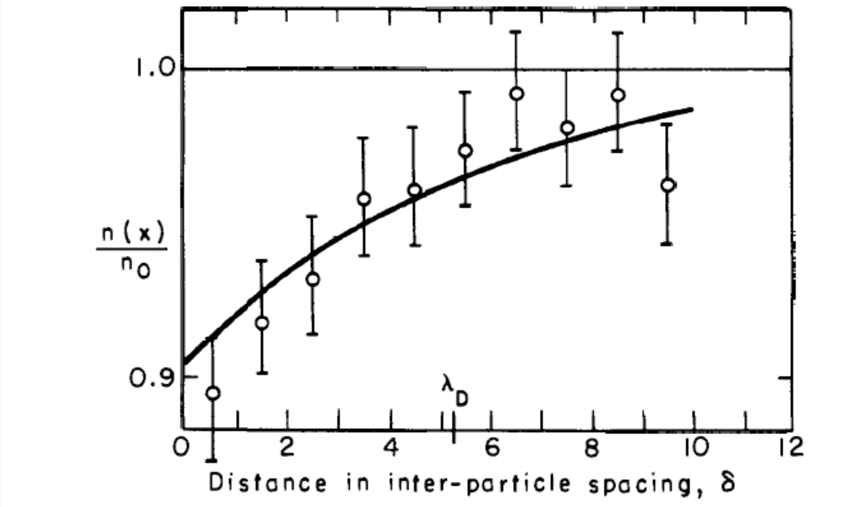
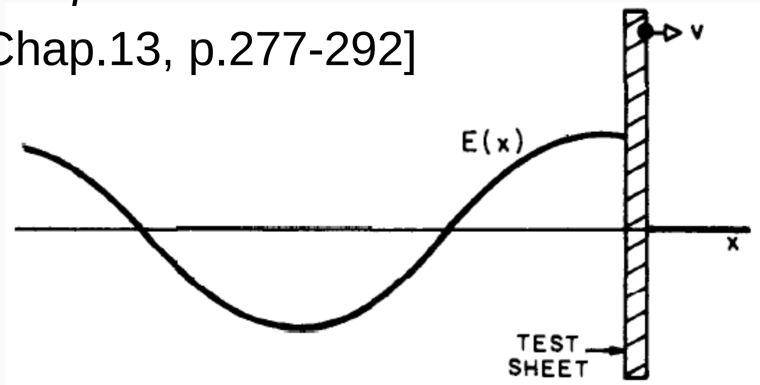
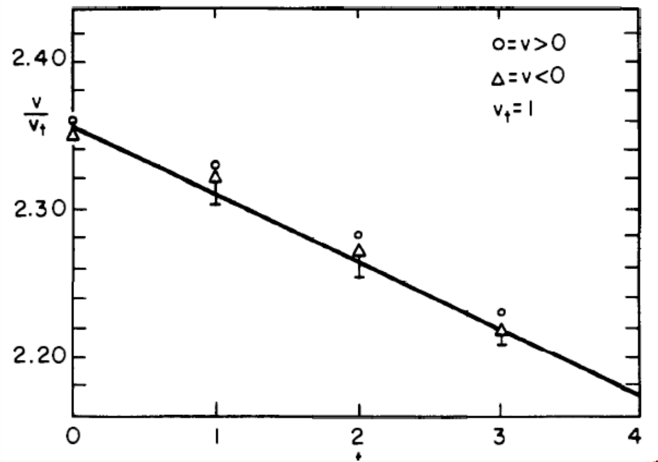


Figure 13-2c Average density of electrons around a test electron sheet at $x = 0$. The curve is the Debye shielding prediction. $n\lambda_D = 5.16$. *(From Dawson, 1962.)*





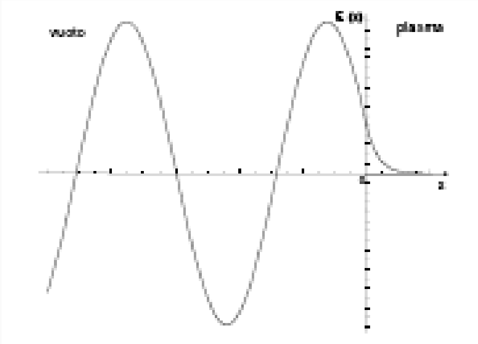
PLAYING WITH THE SHEET MODEL - II

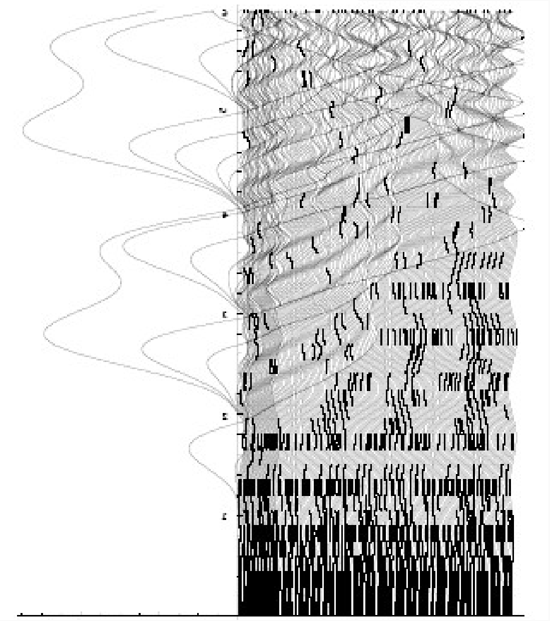
The model can be easily extended to simulate other phenomena such as stochastic heating in a plasma sheath (important for plasma discharge devices)

[Lieberman & Lichtenberg, *Principles of Plasma Discharges and Materials Processing* , 2nd Ed. (Wiley, 2005)]

or at a steep laser-plasma interface (important for collisionless laser absorption)

[Mulser & Bauer, *High Power Laser-Matter Interaction* (Springer, 2010)]





VLASOV-MAXWELL EQUATIONS

Basis for the kinetic description of a collisionless, classical plasma with fully self-consistent ElectroMagnetic fields

$$\frac{df_a}{dt} = \frac{\partial f_a}{\partial t} + \dot{\mathbf{x}}_a \frac{\partial f_a}{\partial \mathbf{x}} + \dot{\mathbf{p}}_a \frac{\partial f_a}{\partial \mathbf{p}} = 0$$

$$f_a = f_a(\mathbf{x}, \mathbf{p}, t), \qquad a = (e, i)$$

$$\dot{\mathbf{p}}_a = q_a(\mathbf{E} + \mathbf{v} \times \mathbf{B}), \qquad \dot{\mathbf{x}}_a = \frac{\mathbf{p}_a}{m_a \gamma_a} = \frac{\mathbf{p}_a}{\sqrt{\mathbf{p}_a^2 + m_a^2}},$$

$$\rho(\mathbf{x}, t) = \sum_{a=e,i} q_a \int d^3p f_a, \qquad \mathbf{J}(\mathbf{x}, t) = \sum_{a=e,i} q_a \int d^3p \mathbf{v} f_a,$$

$$\nabla \cdot \mathbf{E} = \rho, \qquad \nabla \cdot \mathbf{B} = 0, \qquad \nabla \times \mathbf{E} = -\partial_t \mathbf{B}, \qquad \nabla \times \mathbf{B} = \mathbf{J} + \partial_t \mathbf{E}$$

NUMERICAL IMPLEMENTATION

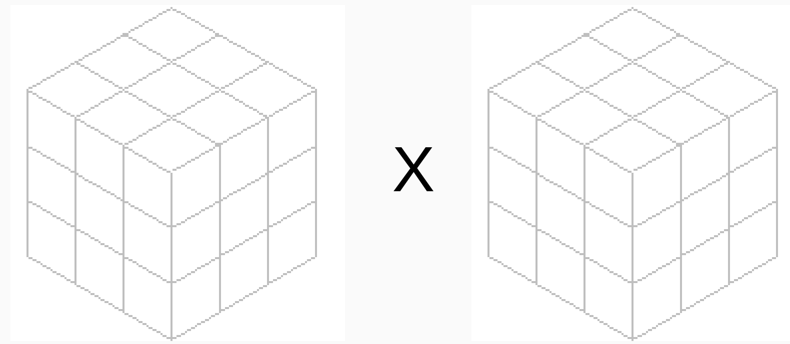
Assume an initial value problem (a study of plasma dynamics)

- Discretize phase space:

$x = i\Delta x$, $i = 0, 1, 2,$ ... ,

$p_x = j\Delta p$, $j = 0, 1, 2,$ ... ,

6D Numerical Grid
(not necessarily Cartesian!)



$x\ y\ z$ $\qquad$ $p_x p_y p_z$

- Discretize time: $t = n\Delta t$, $n = 0, 1, 2,$ ... ,

- Find an algorithm (i.e. finite differences) to advance f with the desired accuracy (test conservation laws: mass, energy, ...)

- Write the code, debug, optimize, and run
  (and if possible check if the results converge ...)

HOW LARGE MUST MY GRID BE?

In physical space the number of points on each axis is $N_q = L/\Delta x$

L = size of the system to be simulated

$\Delta x < d$ = smallest scale to be resolved: depending on the problem

$d = \lambda_D$, c/ω_p, λ (wavelength of a laser pulse), ...

Rule of thumb $L \sim 10d$, $\Delta x \sim d/10$ --> $N_q \sim 10^2$

In 3D we get $N_q^3 \sim 10^6$ gridpoints for the spatial sub-grid

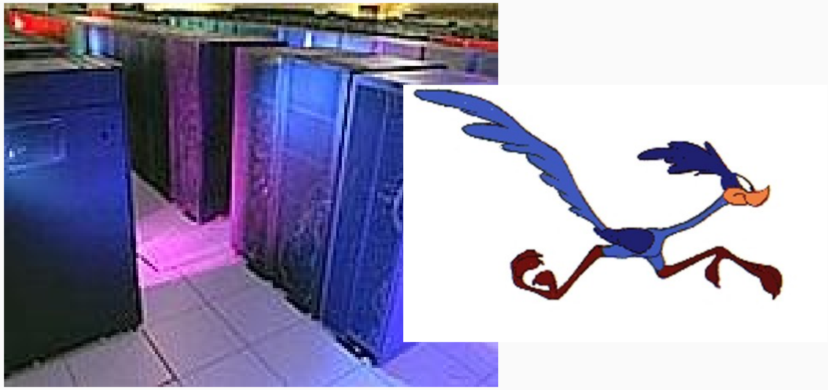
If the grid is similar for momentum space $N_p \sim 10^2$

total $N \sim N_q^3 N_p^3 \sim 10^{12}$ gridpoints

--> 8 TBytes allocated to represent f as a double precision number

SIZE MATTERS: HOW TO DEAL WITH IT?

"Plasma physics is just waiting for bigger computers" (Anonymous)



- Use ROADRUNNER if you can (needs efficient parallel programming)

- Use a different, memory-saving approach to the Vlasov equation: Particle-In-Cell (PIC) method – see next slides

- Restrict yourself to a "model problem": lower dimensionality (1D, 2D), "feasible" parameters, ...

- Remember: **NO simulation can be really "realistic"** i.e. take actual space and time scales with appropriate resolution

PARTICLE-IN-CELL (PIC) METHOD

Assume a discrete “particle” representation of f :

$$f(q,p,t) = f_0 \sum_{n=0}^{N_p-1} g[q - q_n(t)]\delta[p - p_n(t)]$$

By substituting into the Vlasov Eq. we obtain the Equations of Motion for the “characteristics”:

$$\dot{p}_n = \bar{F}_n \qquad \dot{q}_n = \frac{p_n}{m}$$

$$\bar{F}_n = \bar{F}_n(q_n, p_n, t) = \int g(q - q_n) F(q, p_n, t) dq$$

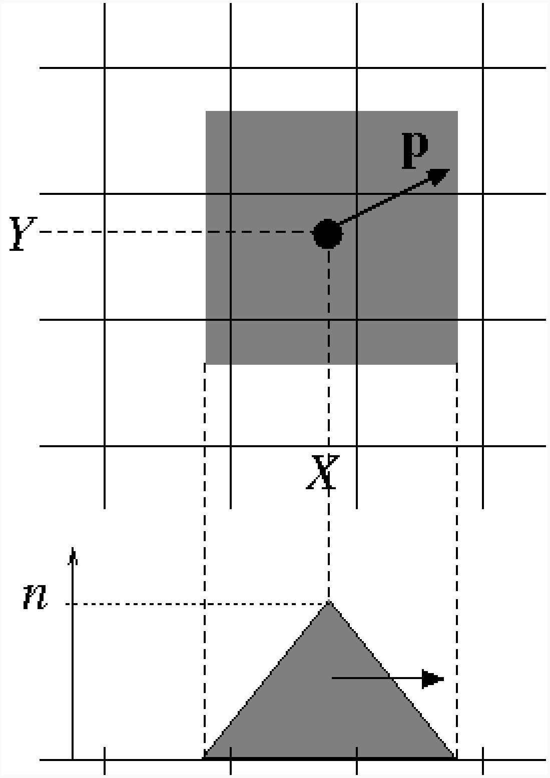
The phase space is represented as an ensemble of particles

(delta-like in p and extended in q via the function $g(q)$):

PIC is a “Lagrangian” approach vs. “Eulerian” (also called “Vlasov”)

THE PARTICLE AND THE CELL

- The plasma is represented by a large (but limited) set of computational particles (simply named “electrons” and “ions”)
- The EM fields are allocated on a discrete grid, i.e. “in the cell”
- Each particle (usually extended in space) contributes to the charge and current densities in its parent cell (and its neighbors)
- The Lorentz force is evaluated as an average over the fields in the overlapping cells

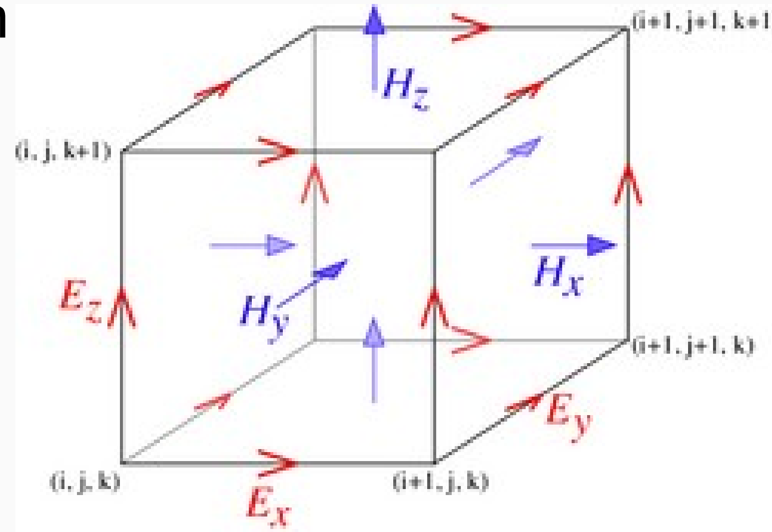


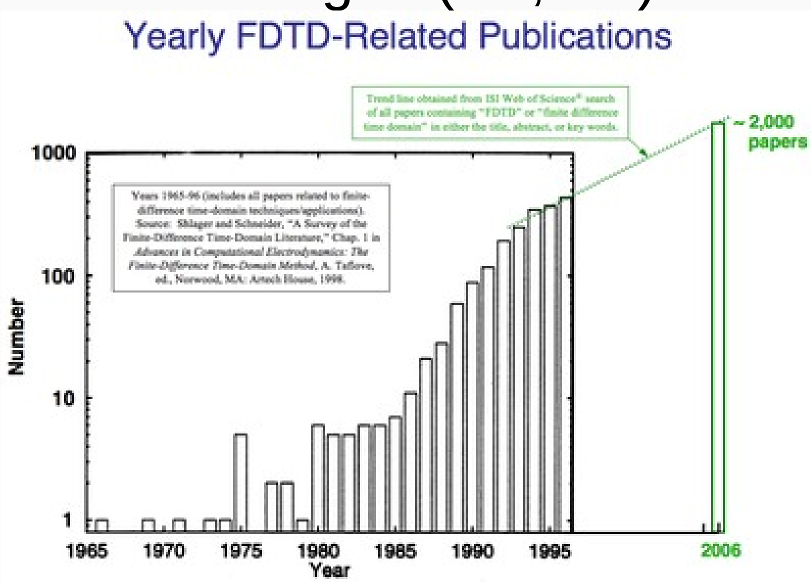
On a supercomputer up to some $\sim 10^9$ particles may be allocated (still typically orders of magnitude smaller than real numbers)

INO-CNR
Istituto Nazionale di Ottica

FDTD AND YEE LATTICE

- FDTD : Finite-Difference Time-Domain
  typical method to solve Maxwell
  equations in PIC codes
- Yee lattice: typical distribution of
  EM fields used in the cell of a
  Cartesian grid (2D, 3D)





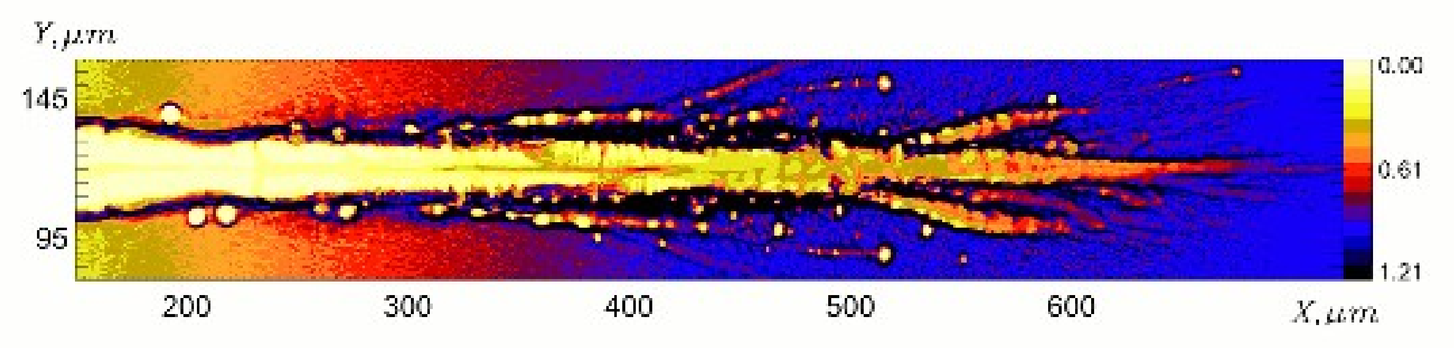
Pictures taken from Wikipedia:
http://en.wikipedia.org/wiki/Finite-difference_time-domain_method

INO-CNR Istituto Nazionale di Ottica

PIC VS VLASOV: PROS AND CONS

EASY / HARD / FEASIBLE / UNKNOWN	PIC	VLASOV
DEVELOPMENT	easy, quite general, well documented	non trivial, specific
NOISE	significant	negligible
WORKLOAD	saving	very large
DENSITY RESOLUTION	problems with statistics & large gradients	excellent
MOMENTUM SPACE	unbounded	bounded
PARALLELIZATION	well suitable but non trivial load balancing	straightforward for local algorithms
FLEXIBILITY	girokinetic approx., collisions, ionization, ...	?

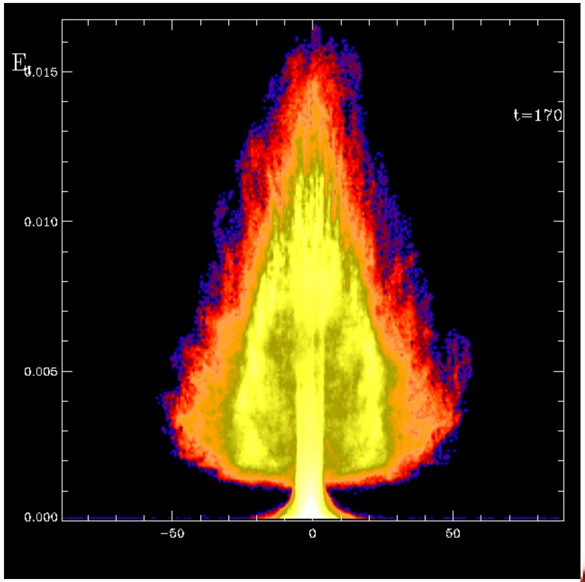
SOME LASER-PLASMA **PIC**-TURES - 2D



Top: self-channeling, breakup and soliton formation by an intense laser pulse

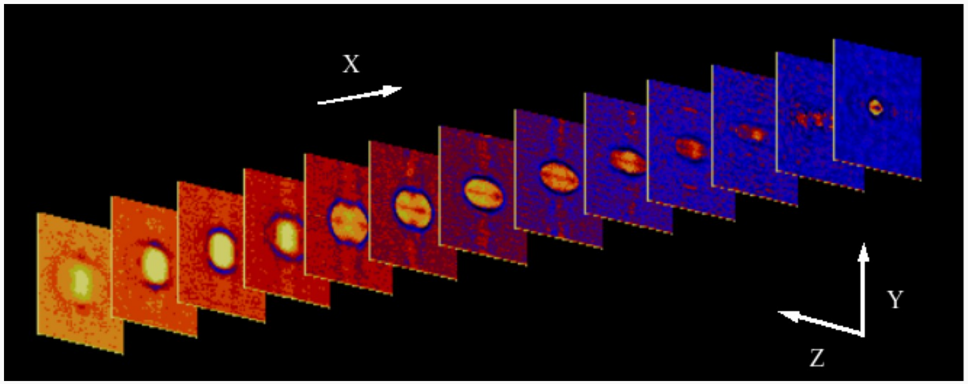
Right: momentum vs angle distribution of ions for radiation pressure acceleration of a dense plasma slab

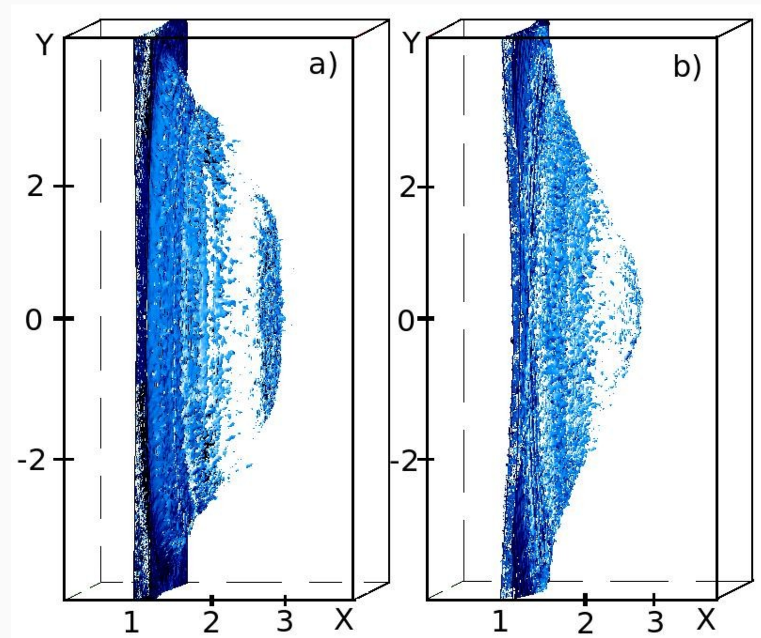
Simulations performed at CINECA, Italy



T.V.Liseykina and A.Macchi, IEEE Trans. Pl. Sc. **36**, 1136 (2008), Special Issue on "Images in Plasma Science"

SOME LASER-PLASMA **PIC**-TURES - 3D





Top: anisotropic self-channeling in 3D

Right: radiation pressure acceleration of a thin foil in 3D

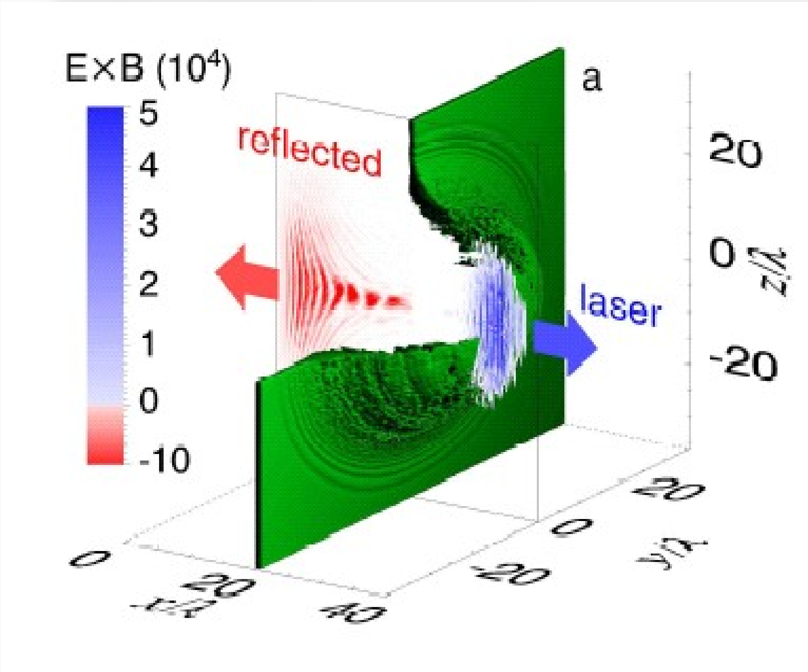
Simulations performed at CINECA, Italy

Presently running: supercomputing project ISCRA-TOFUSEX

Italian Super-Computing Resource Allocation -

"TOwards FUll-scale Simulation of EXperiments"

MORE BEAUTIFUL 3D **PIC**-TURES ...

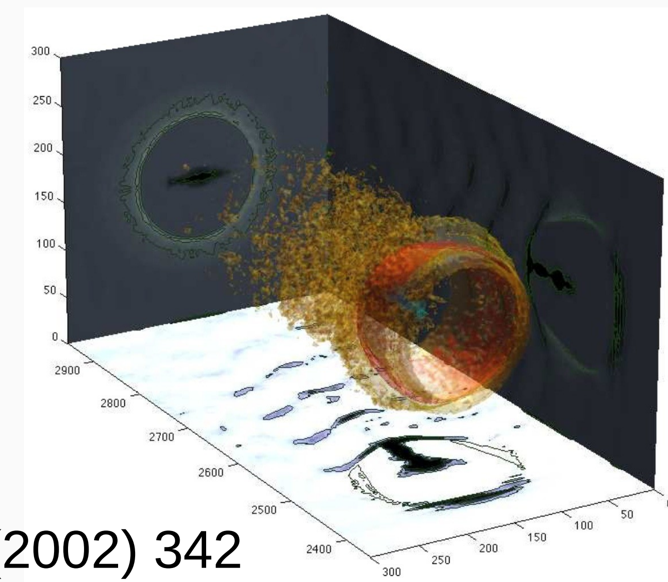


Left: radiation pressure acceleration

in the extreme intensity regime

T.Esirkepov et al,

Phys.Rev.Lett. **92** (2004) 175003



Right: plasma "bubble" formation

for laser-plasma electron acceleration

Simulation by OSIRIS code

L.Fonseca et, *Lect. Notes Comp. Sci.* **2331**(2002) 342

Smart data visualization is important (and a key to success...)

INO-CNR
Istituto Nazionale di Ottica

FINAL CONSIDERATIONS

- If interested in plasma simulation, begin putting you hands on a simple numerical model (e.g. the plasma sheet)
- If you need kinetic simulations, using PIC or Vlasov depends on the nature of the problem you have (see pros and cons)
- If you like playing with computers, this is a field where skills in parallel programming, algorithm development and optimization, advanced visualization techniques, ... , are highly appreciated

The trend is to increase code capabilities to include additional physics and/or to run on more and more powerful computers for more and more “realistic” simulations (but it's a long way...)