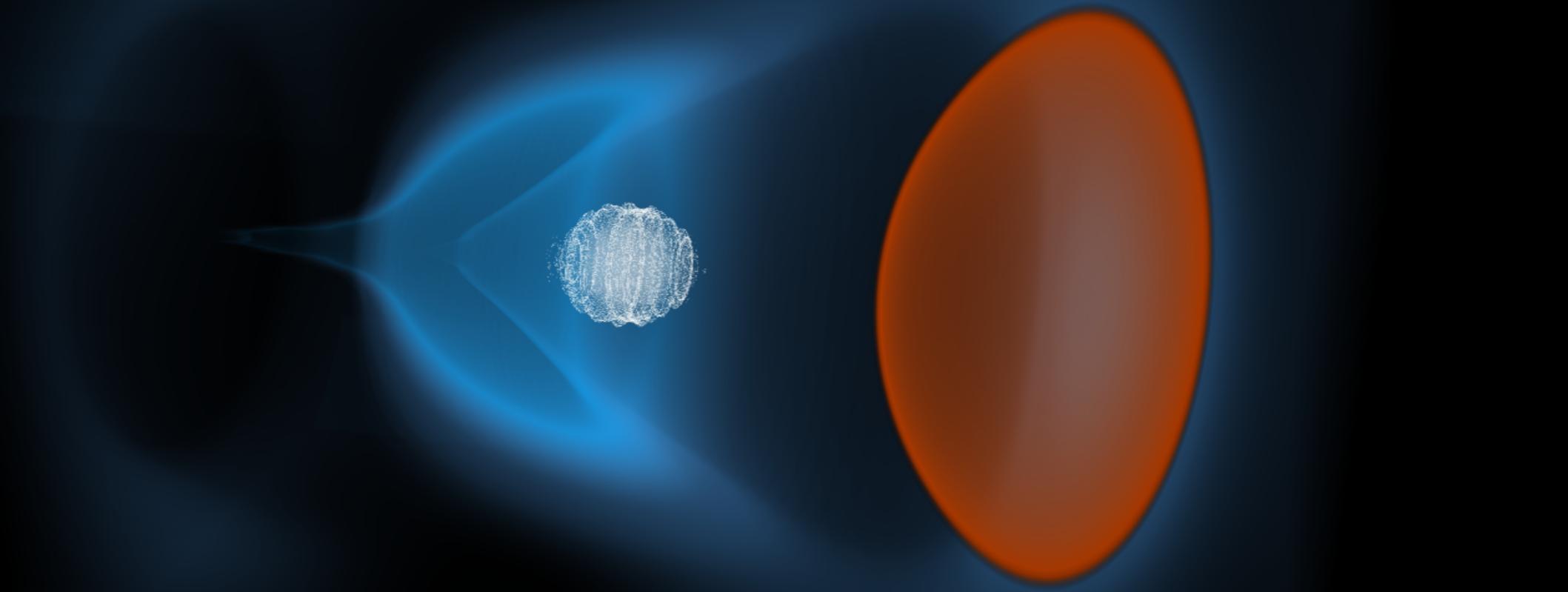


# Numerical simulation of laser wakefield acceleration with a PIC code



Francesco Massimo

12-15 dec 2021

# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical

# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical

# Particle accelerators applications ( just a few )

## Science:

- Particle physics
- Nuclear Physics
- Matter Physics
- 3D Study of proteins

## Industry - Medicine - Security - Energy - Heritage:

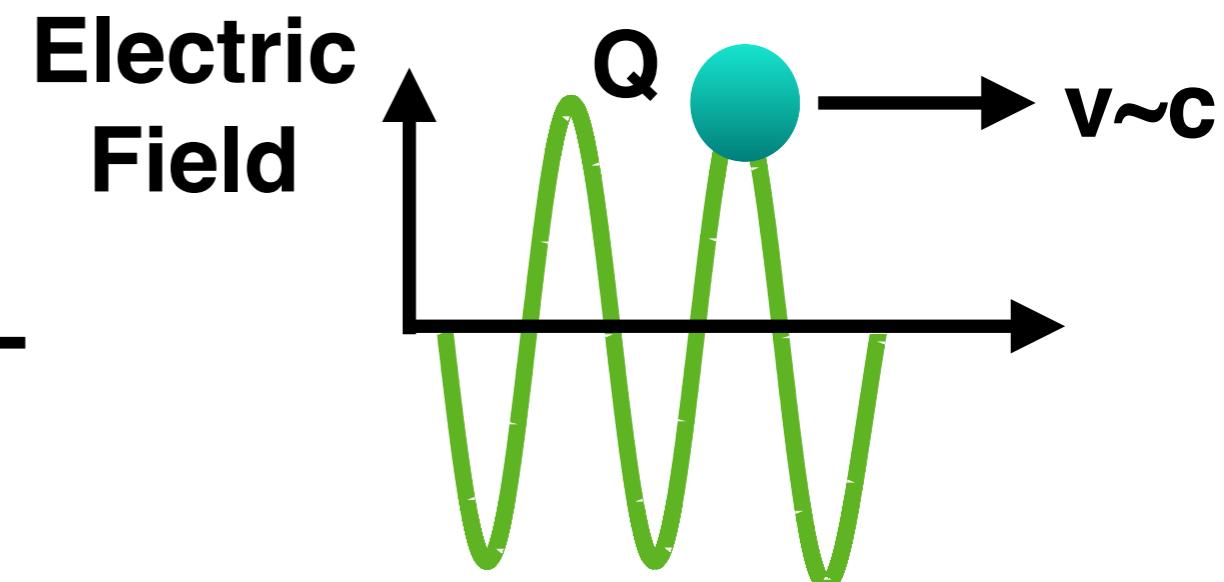
- Cancer treatment, medical imaging, biology studies
- Electronic industry, study of materials
- Authentication of artwork, security controls
- Elimination of polluting substances

# Particle accelerators size and cost

Relativistic charge **Q**

In an accelerating cavity of length **L**

With peak accelerating field **E**



$$\text{Maximum energy gain} = Q E L$$

Given a target energy,

if **E** is limited by technology, **L** increases

→ need more cavities

→ the accelerator size (and cost) increases

# Particle accelerators size and cost

**Maximum peak accelerating electric field  $E$  in metallic cavities:**

$$E_{\max} \sim 100 \text{ MV/m}$$

**Possible solution: plasma accelerators**

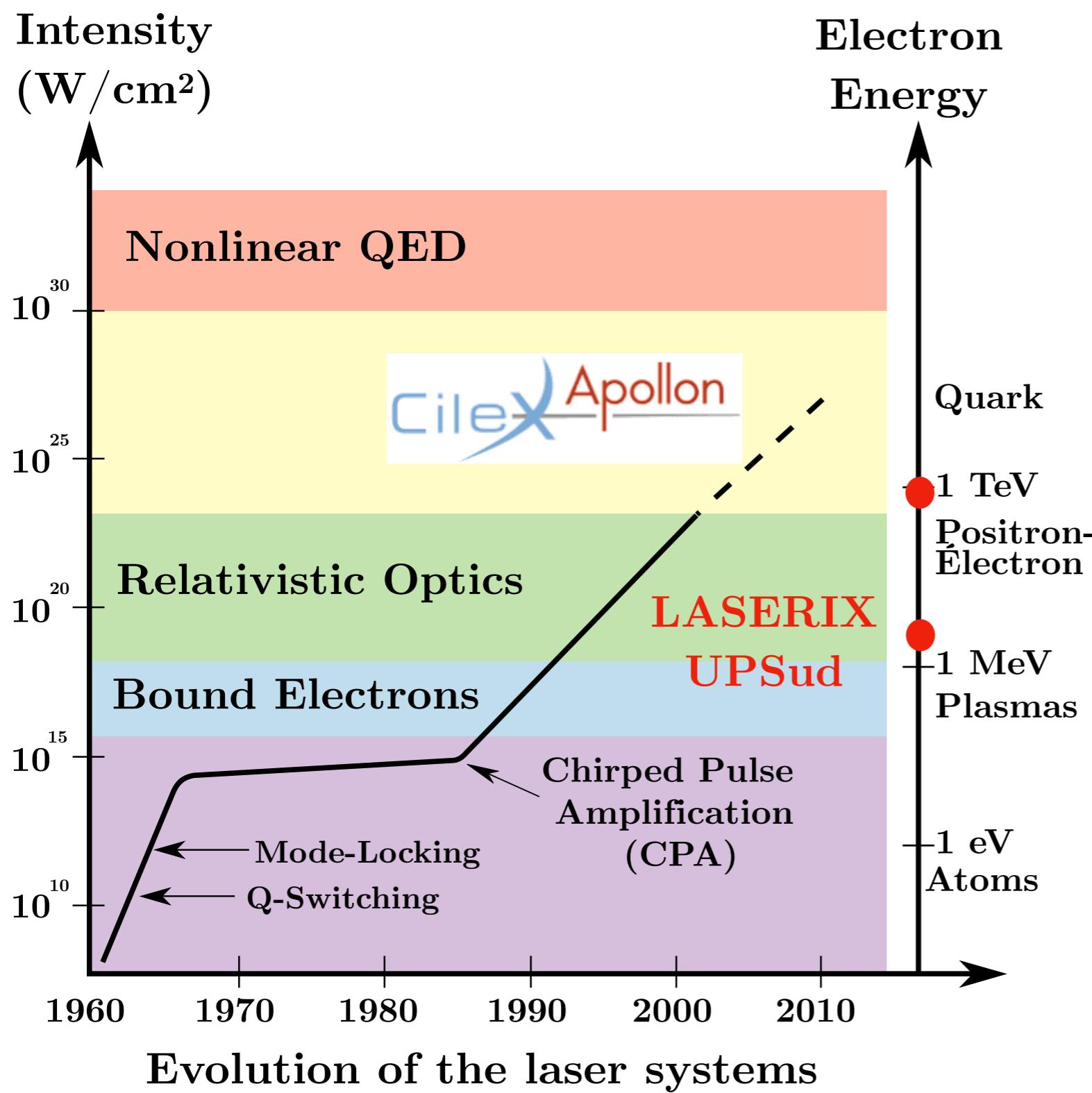
$$E_{\max} > 100 \text{ GV/m}$$

→ **Potential to have accelerators 1000 times smaller**

# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical

# High Intensity Laser-Matter Interaction



## Techniques:

- Chirped Pulse Amplification
- Femtosecond Lasers
- Nonlinear Optics
- ...

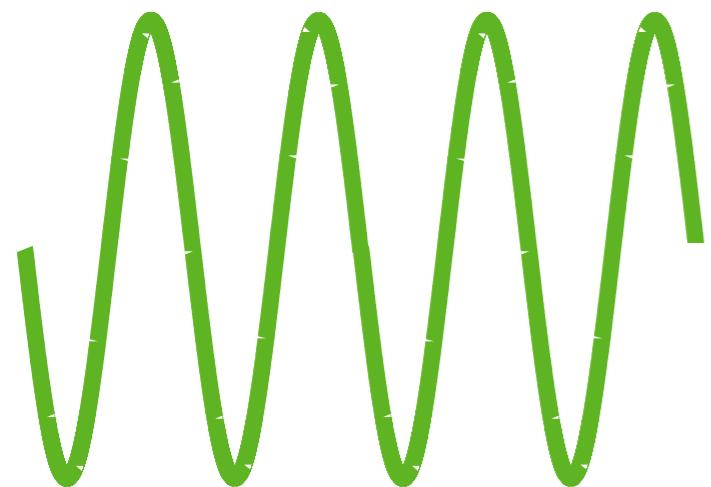
## Physics Domains:

- Relativistic Fluids
- Physics of Relativistic Plasmas
- Relativistic Optics
- Laboratory Astrophysics

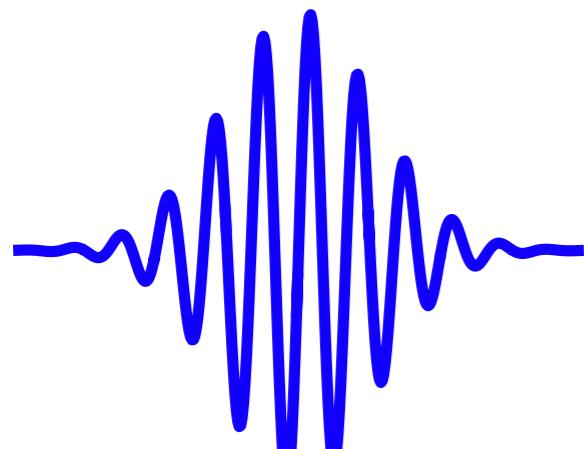
## Applications:

- Electron and ion sources
- Acceleration of particles
- Radiation sources (UV, X,  $\gamma$ )
- Novel, high resolution diagnostics
- Pump-probe measurements

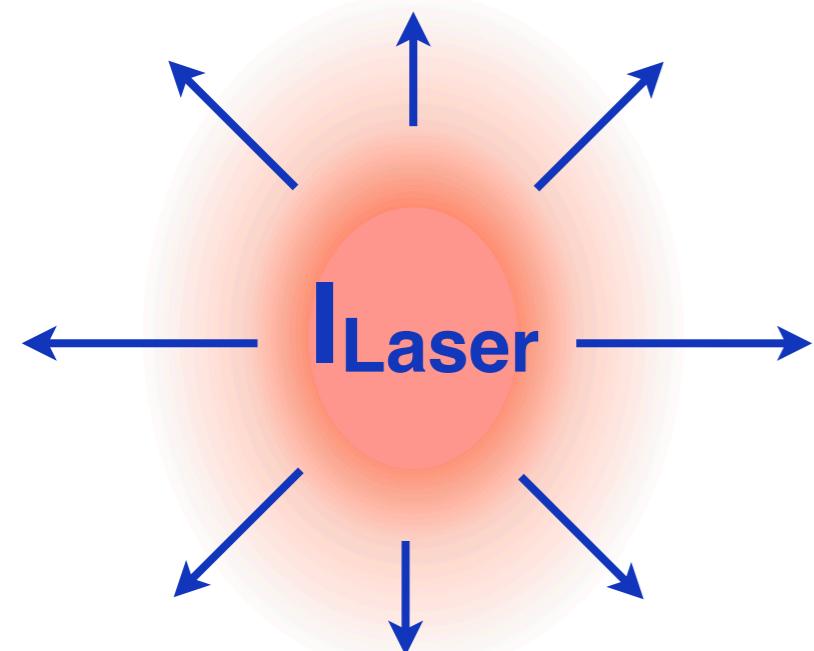
# Ponderomotive Force



Electron in infinite plane wave:  
Oscillating Force



Electron in finite laser pulse:  
Oscillating Force + Ponderomotive Force

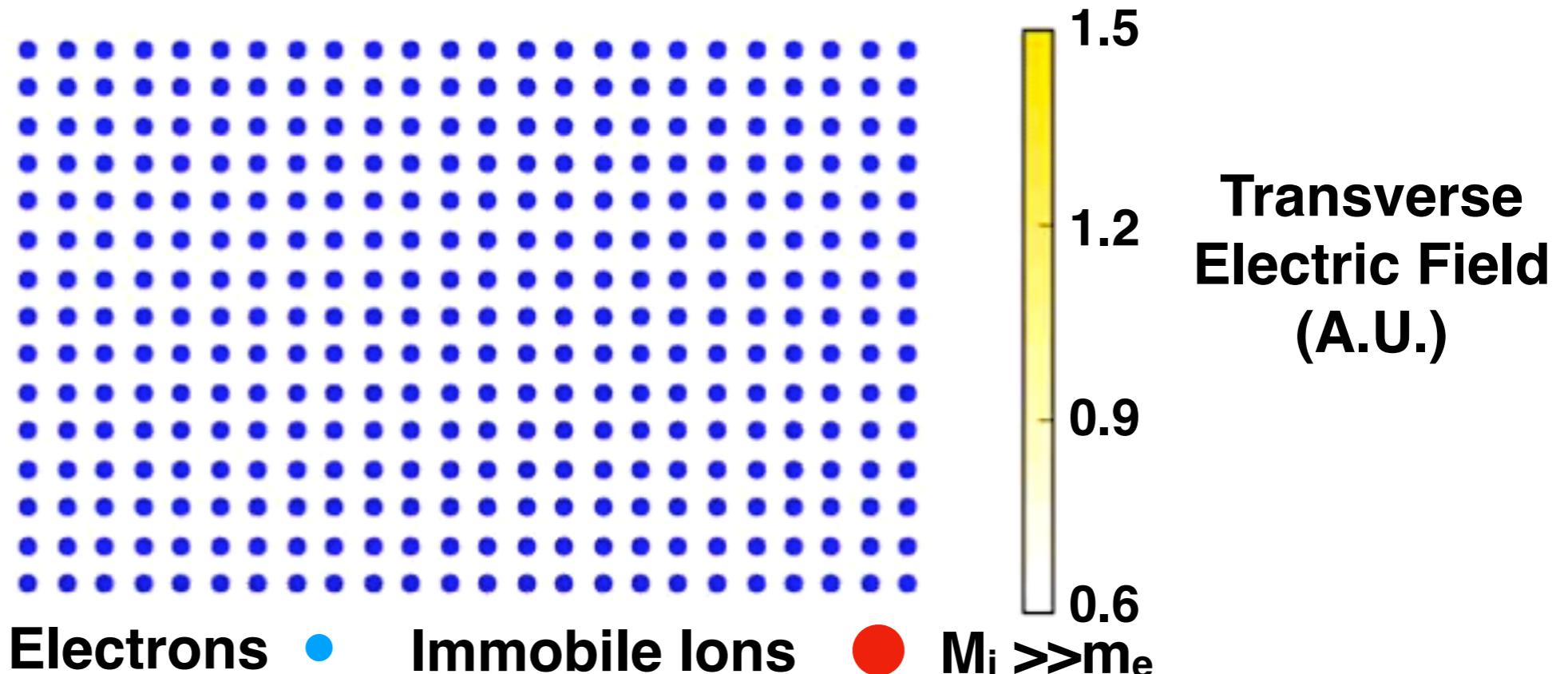


$$\mathbf{F}_{\text{pond}} \propto -\nabla I_{\text{Laser}} \text{ (a.k.a. radiation pressure)}$$

# Laser Wakefield Acceleration (LWFA): plasma wave excitation

Laser Beam  
Duration: 28 fs

Ponderomotive Force:  
 $F = -\nabla I_{\text{Laser}}$



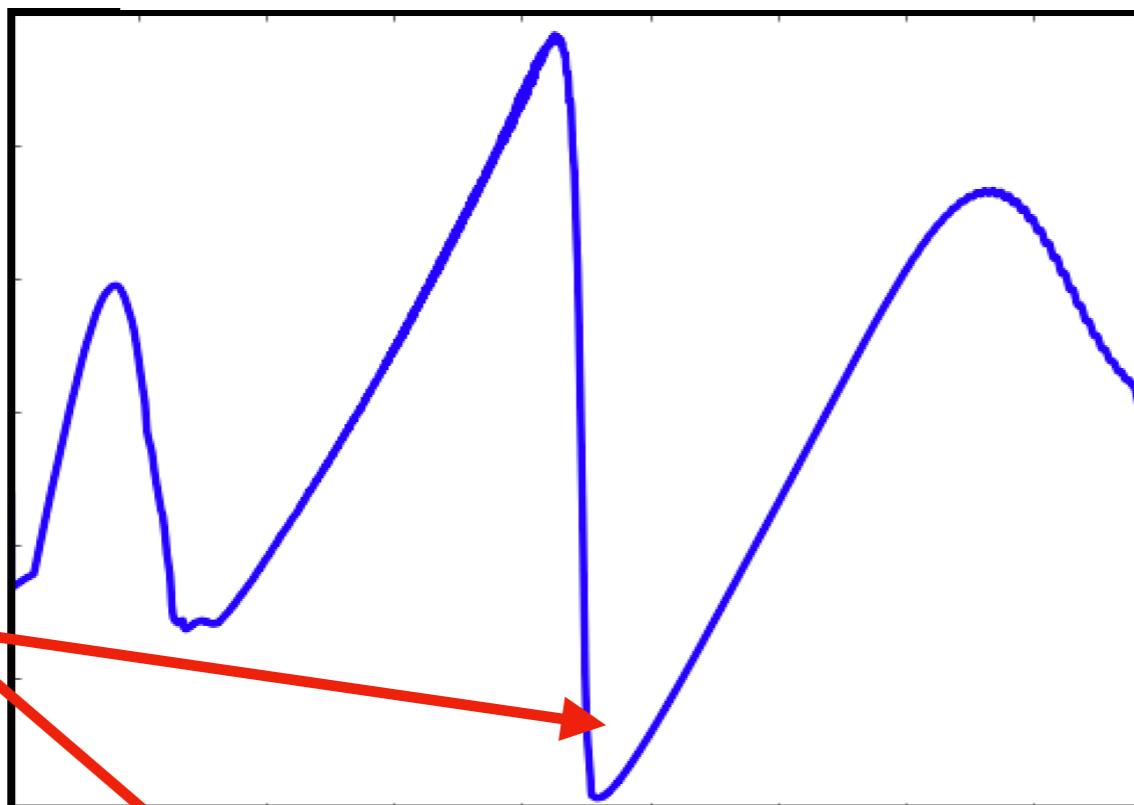
Plasma density:  
 $3 \cdot 10^{18} \text{ cm}^{-3}$

Plasma wavelength:  
 $\sim 20 \mu\text{m}$

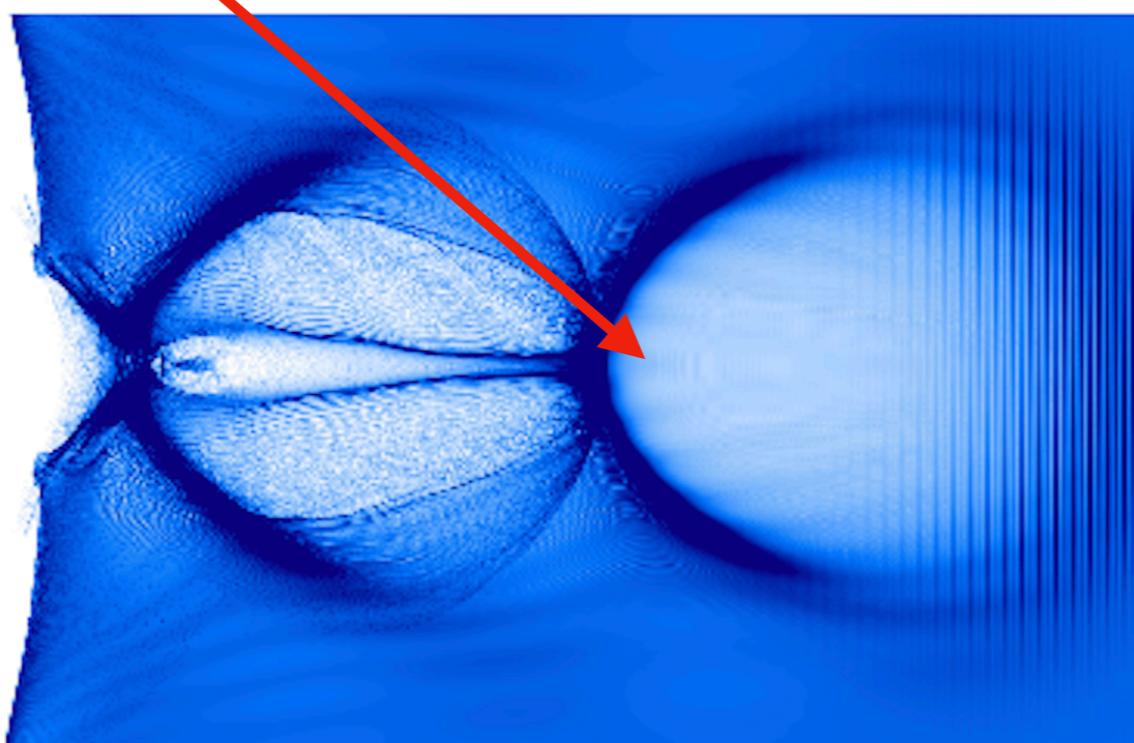
# Laser Wakefield Acceleration (LWFA): accelerating electric field

$E > 100 \text{ GV/m}$

Relativistic electrons injected here are accelerated towards the right

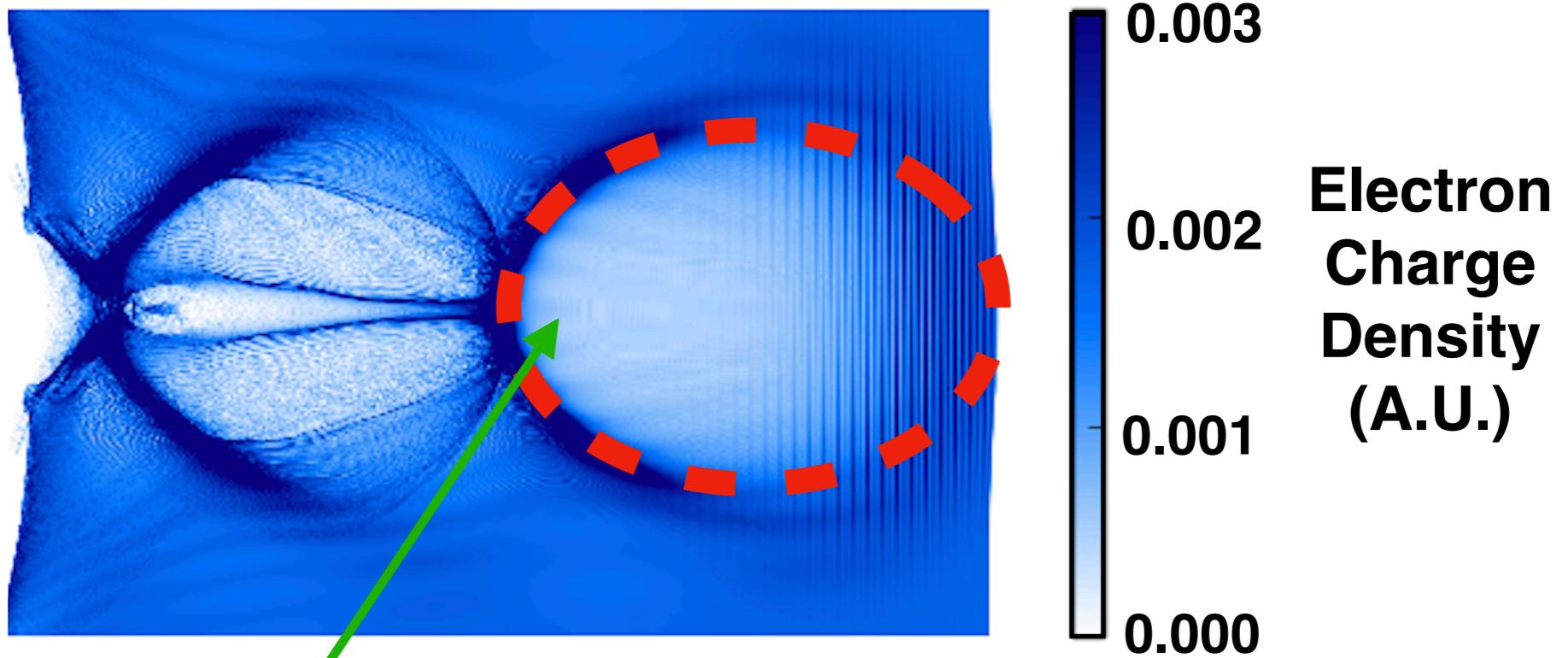


Longitudinal Electric Field (A.U.) on propagation axis

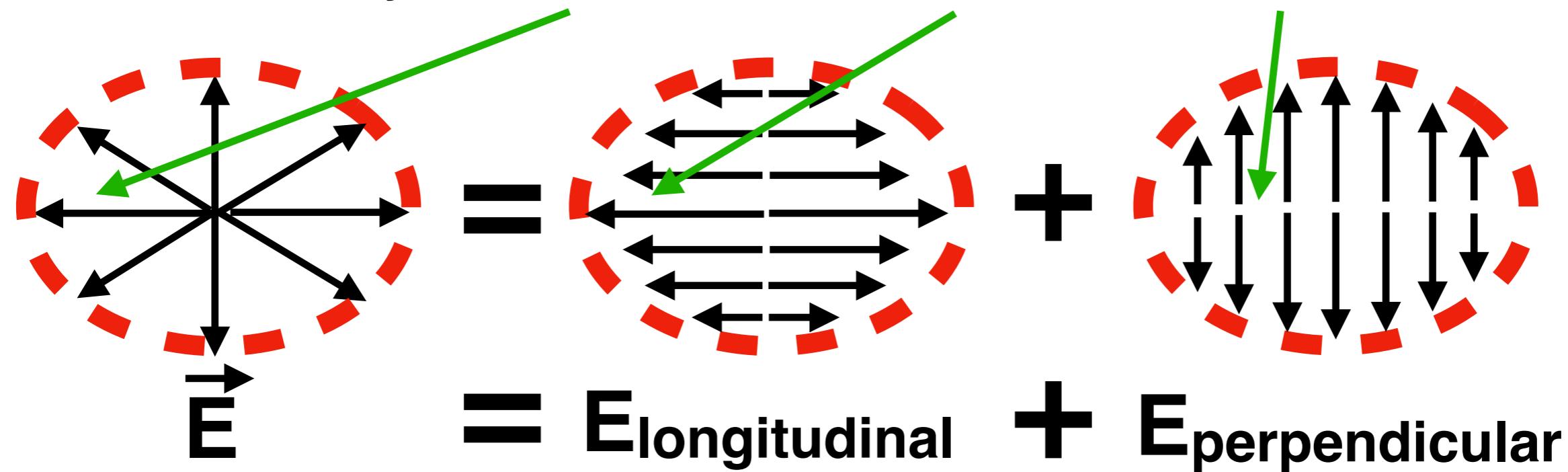


Electron Charge Density (A.U.)

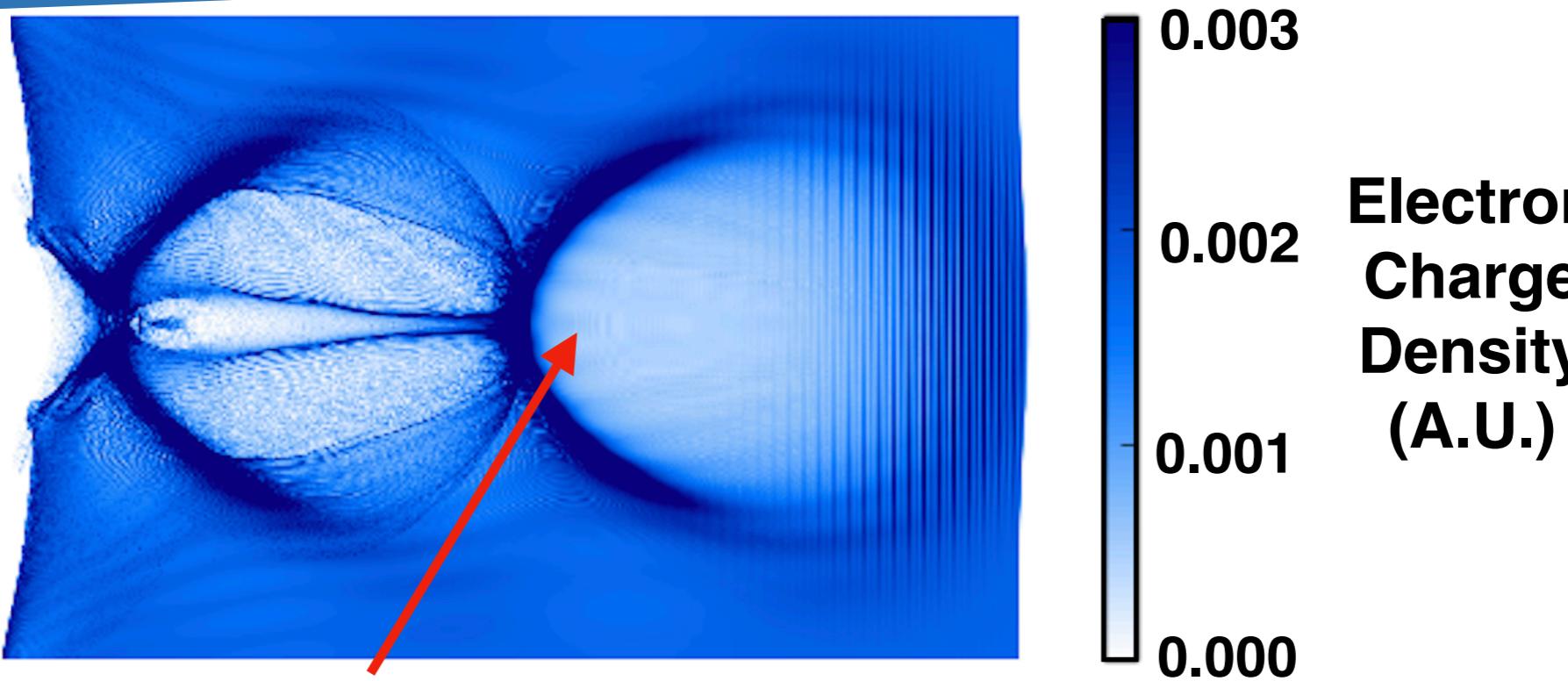
# Laser Wakefield Acceleration (LWFA): fields inside the bubble



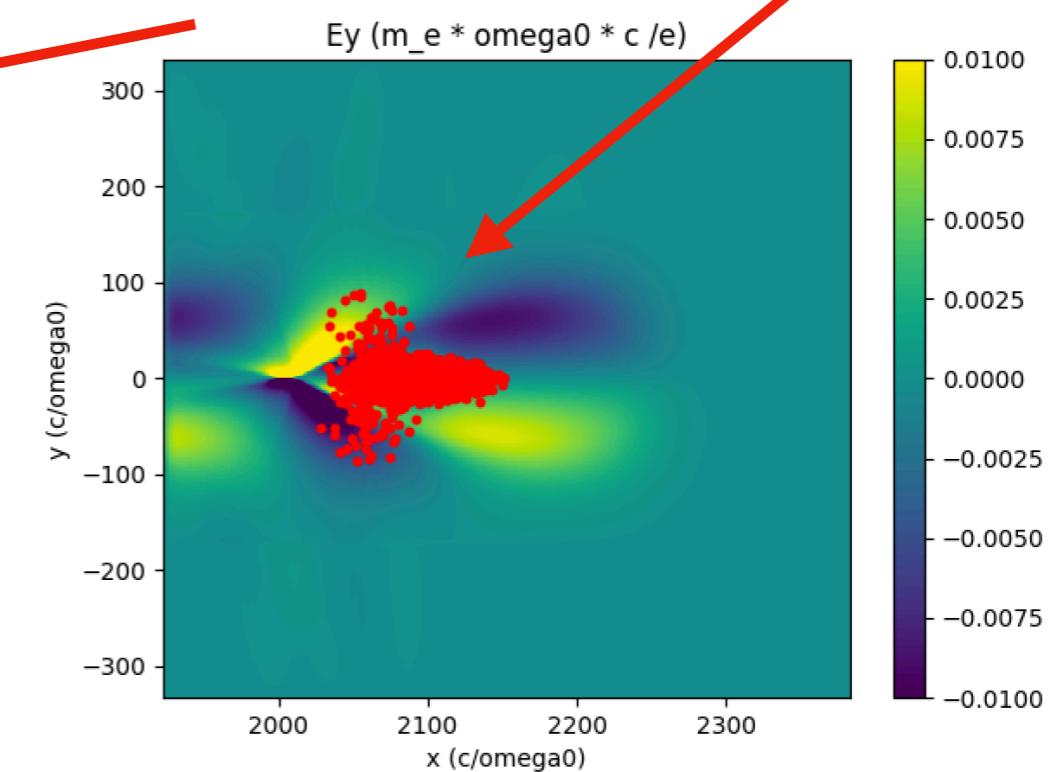
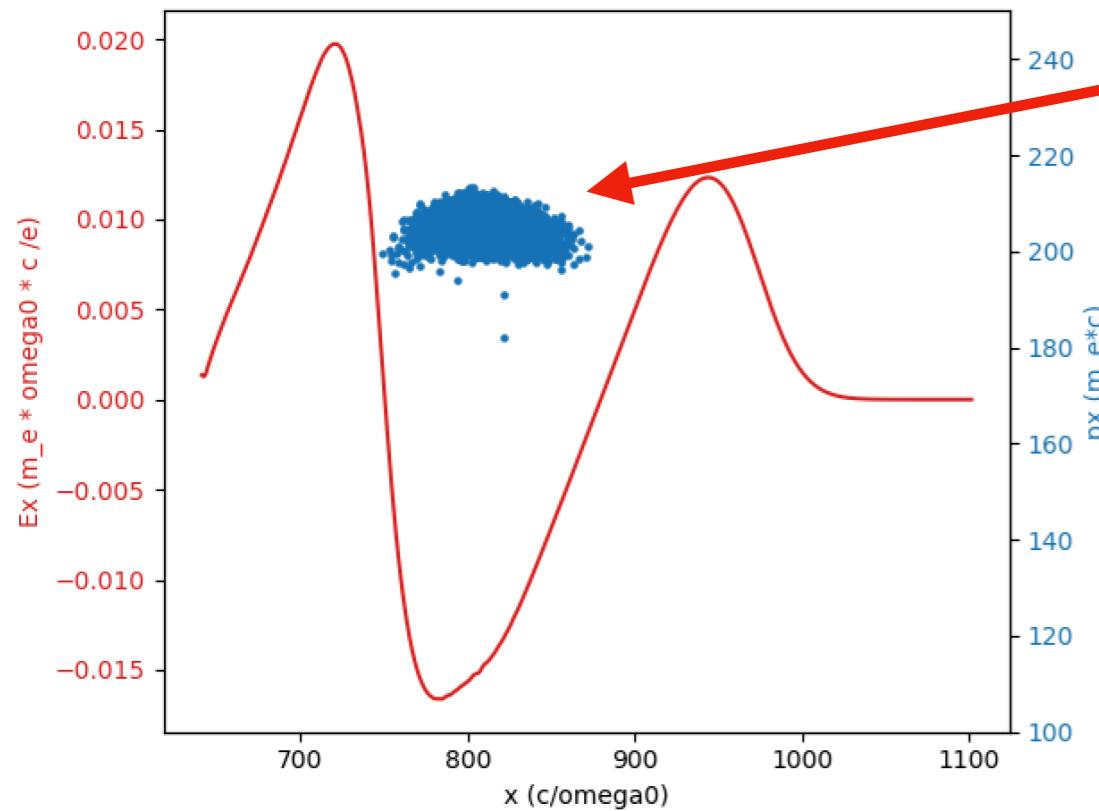
Electrons injected here are both accelerated and focused



# Laser Wakefield Acceleration (LWFA): fields inside the bubble

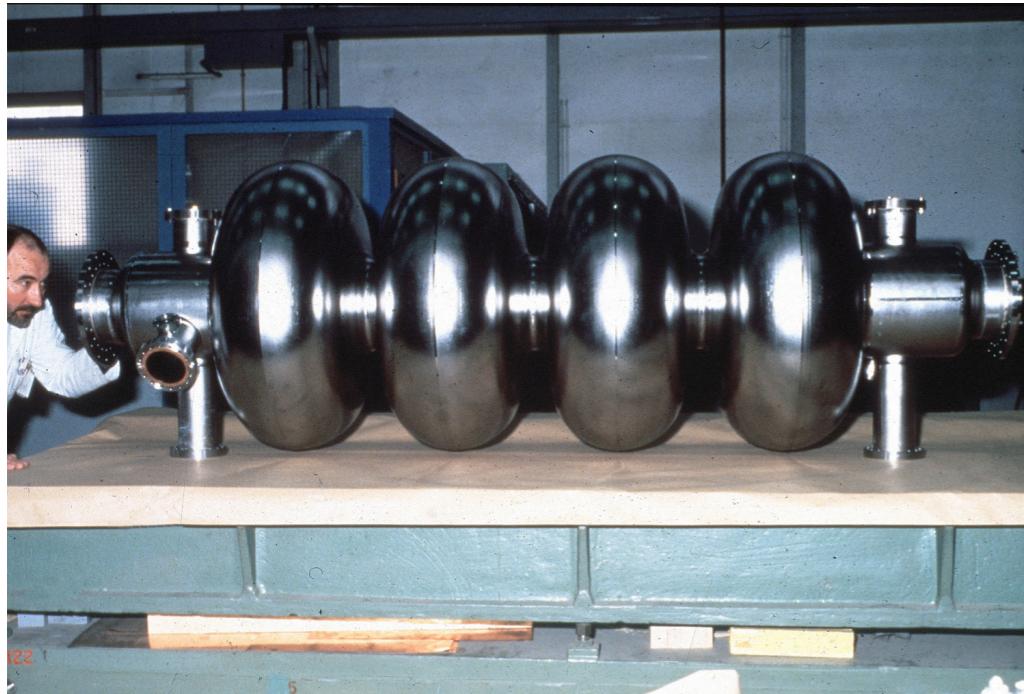


Relativistic electrons injected **here** are both **accelerated** and **focused**



# Compactness of laser - plasma accelerators

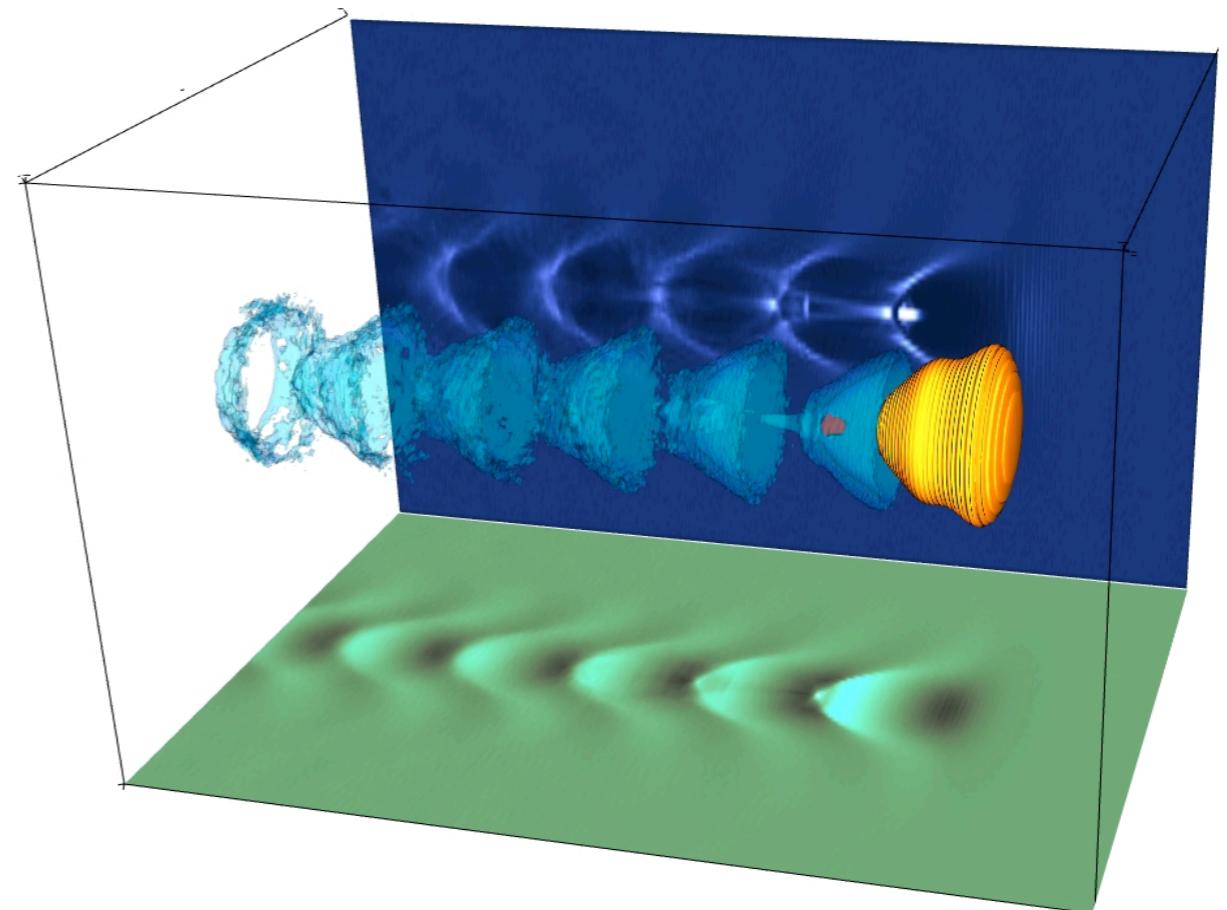
## RF Cavity



$1 \text{ m} \Rightarrow 50 \text{ MeV Gain}$

Electric field  $< 100 \text{ MV/m}$

## Plasma Cavity



$1 \text{ mm} \Rightarrow 100 \text{ MeV}$

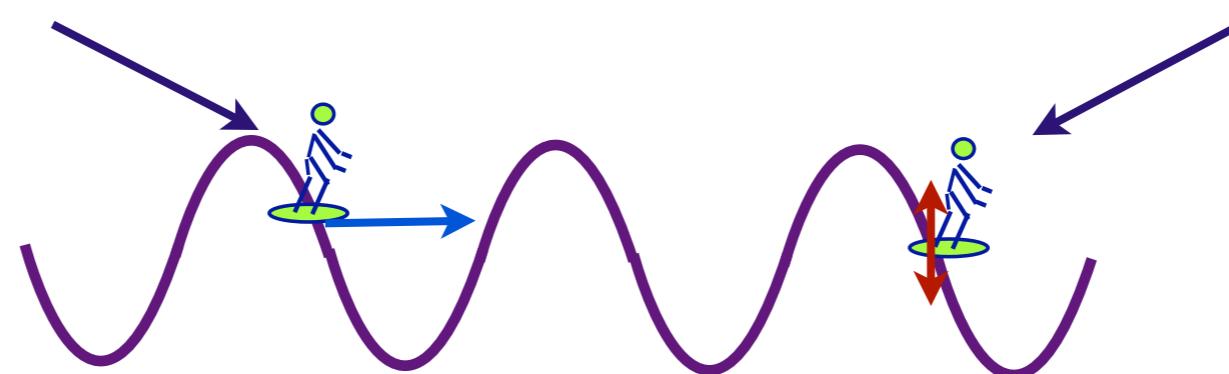
Electric field  $> 100 \text{ GV/m}$

V. Malka et al., Science 298, 1596 (2002)

# Laser Wakefield Acceleration (LWFA) challenges: injection

## Surfer with

- sufficient initial velocity,
- injected in the proper phase

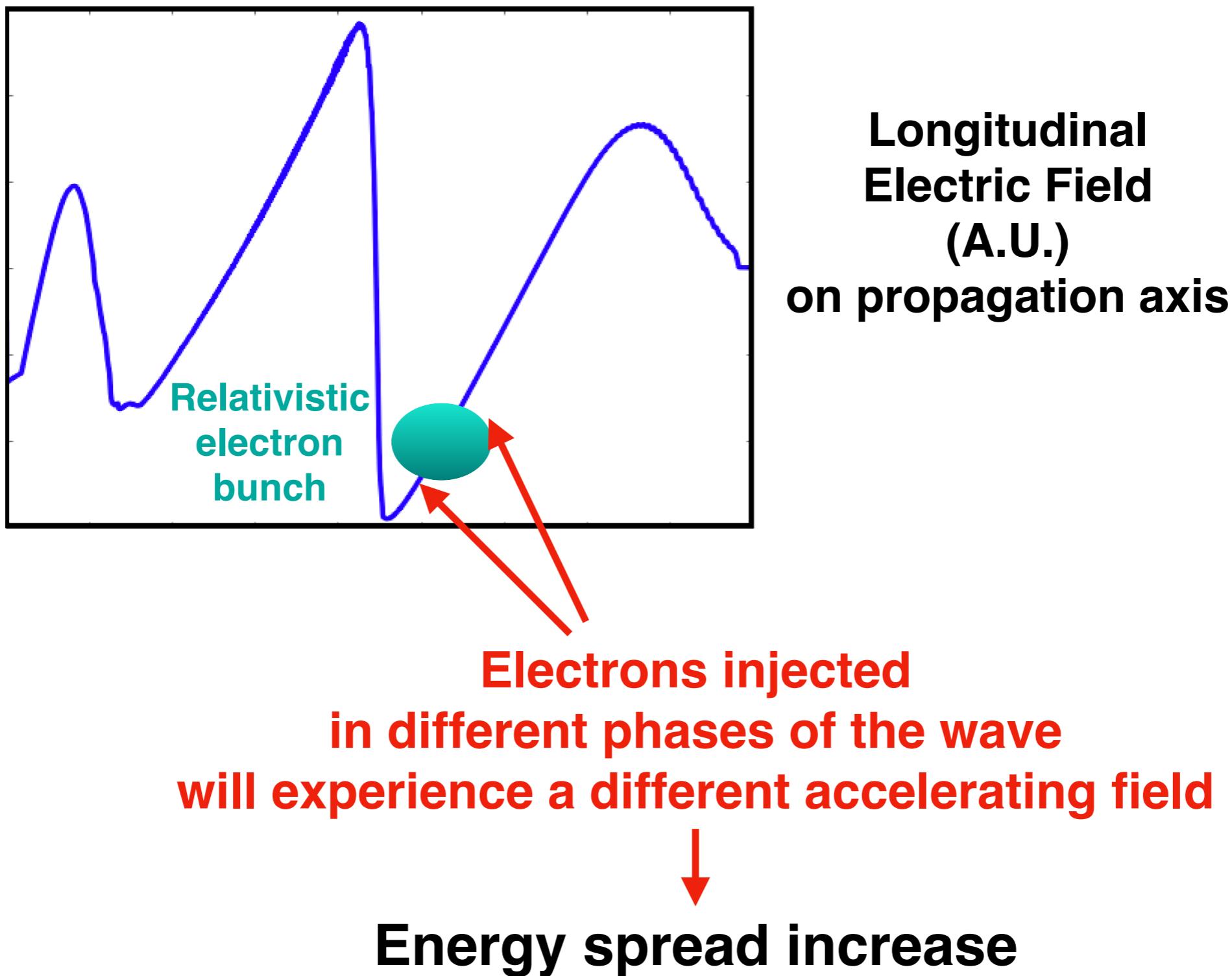


## Surfer with Initial zero velocity

**Surfer: electron**  
**wave: electric field of the plasma wave in the wake of the laser**

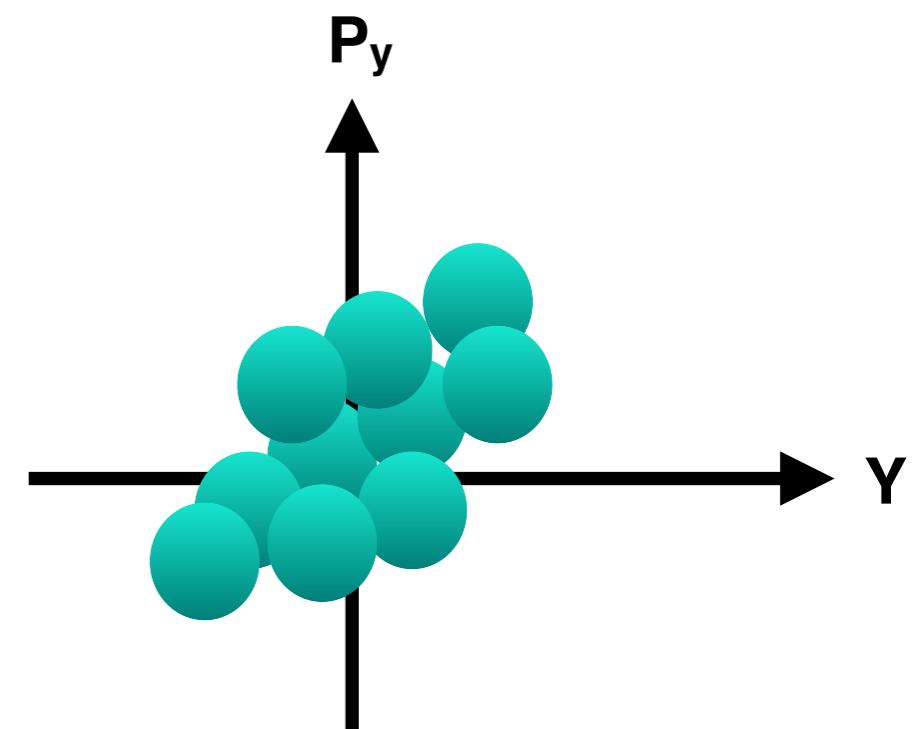
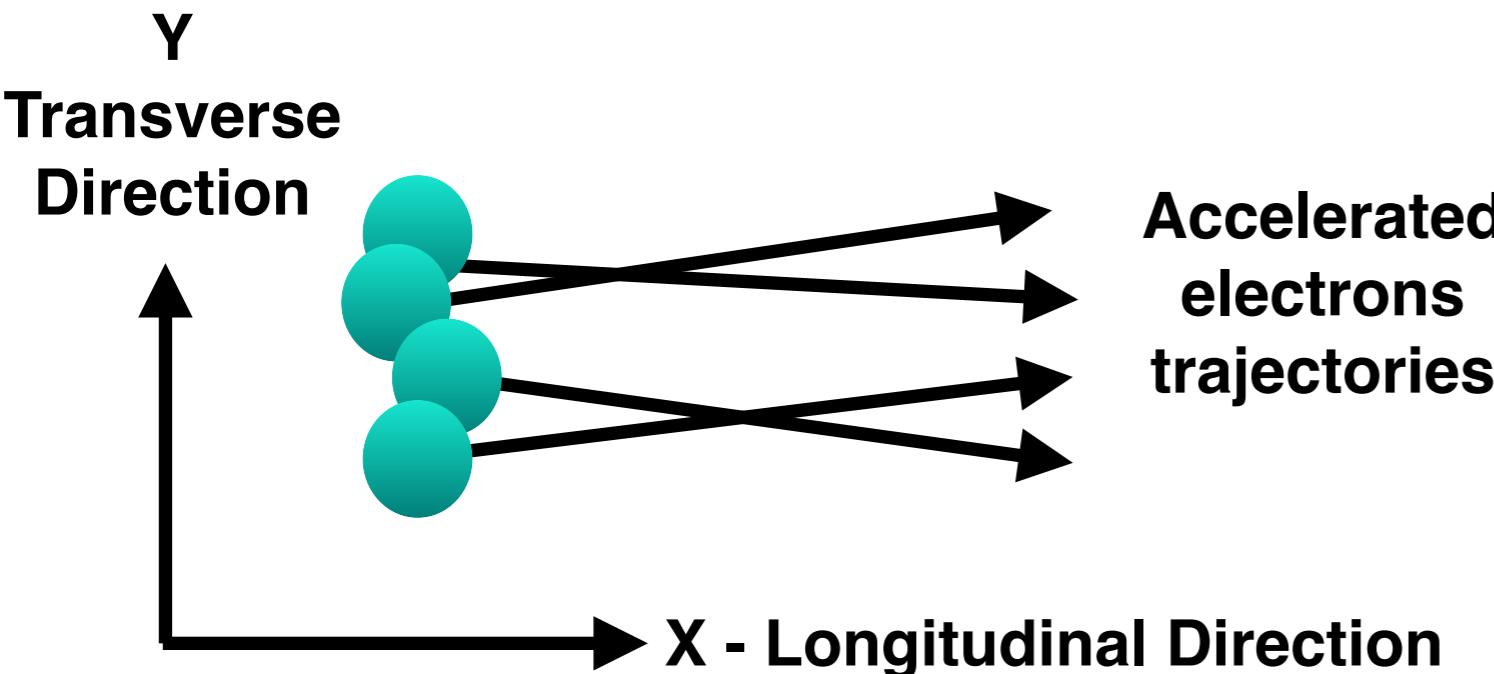
Plasma wavelength < 100 μm, Duration of the electron beam < 10 fs ~3 μm

# Laser Wakefield Acceleration (LWFA) challenges: energy spread



# Laser Wakefield Acceleration (LWFA) challenges: emittance

Bunch distribution  
in the transverse phase space



Most applications of accelerated beams require:

- Small transverse size (i.e. small  $\sigma_y$ )
- Small divergence (i.e. small  $\sigma_{p_y}/\text{Energy}$ )



Minimize  
Transverse Emittance

Transverse  
Normalized  
Emittance

$$\epsilon_{ny} = \frac{1}{m_e c^2} \sqrt{\sigma_y^2 \sigma_{p_y}^2 - \sigma_{y p_y}^2}$$

# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical

# Why numerical simulation is necessary for LWFA?

- **No general analytical solutions are available**
- **Often (ok, this happens always) you cannot know everything simultaneously from measurements**
- **Before the experiment:**
  - Study new physical phenomena
  - Conceive new kind of experiments
  - Design experiments
- **After the experiment:**
  - Analyze the data
  - Understand the physics

# Model for Laser-Plasma Interaction

## Complete Maxwell-Vlasov system

Plasma distribution function

$f(x, y, z, p_x, p_y, p_z, t) \longrightarrow 6 \text{ dimensions + time!}$

Coupled to the electromagnetic fields  $\longrightarrow$  Non-linearity

$$\frac{\partial f}{\partial t} + \beta c \cdot \nabla_x f - e(E + \beta c \times B) \cdot \nabla_p f = 0 \leftarrow \text{Collisions neglected}$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad \text{Current density } J \text{ of the plasma}$$

$$\nabla \times B = -\mu_0 ec \left( \int \beta f d^3 p \right) + \frac{1}{c^2} \frac{\partial E}{\partial t}$$

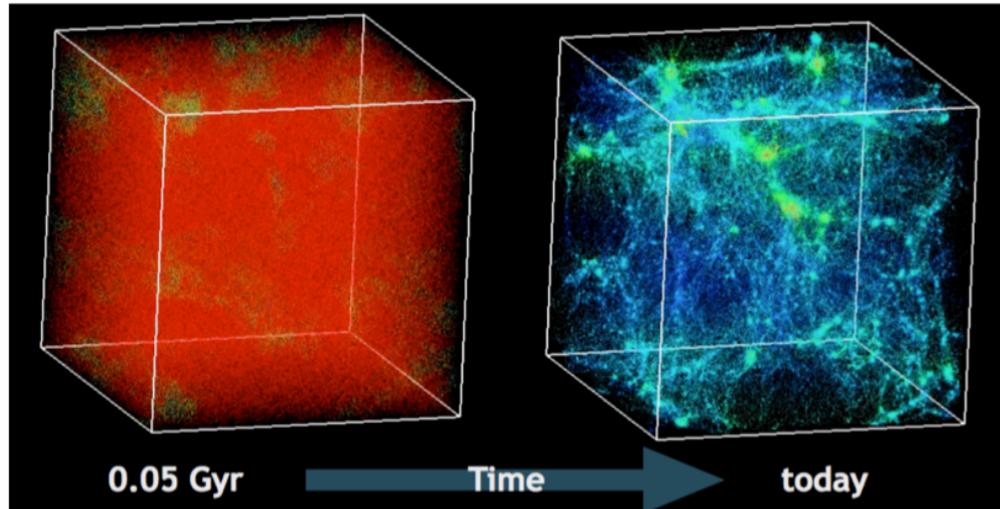
Explorable Physics:

- Relativistic optics ( laser self-focusing, ... )
- Nonlinear phenomena
- Wave-breaking
- Injection and acceleration of particles

## Approximate solution: Particle in Cell method

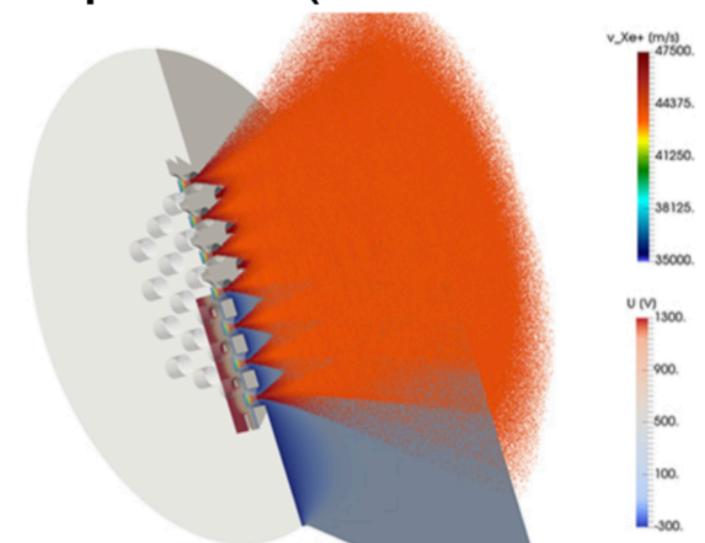
# Particle in Cell (PIC) method : essential plasma investigation technique

## Cosmology



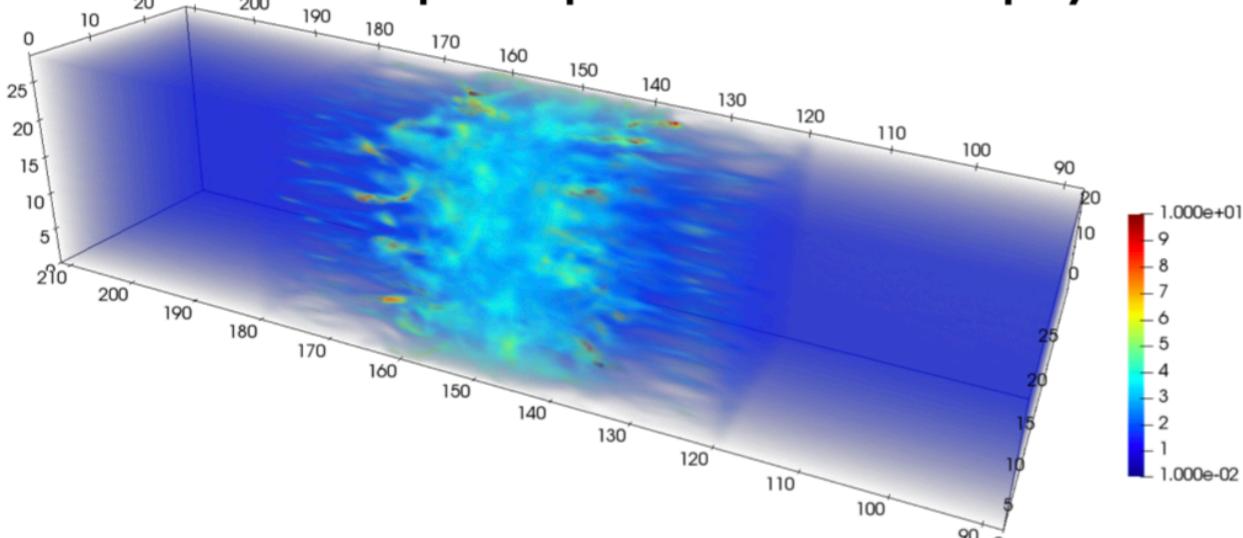
source: K. Heitmann, Argonne National Lab

## Space propulsion (Plasma thruster)



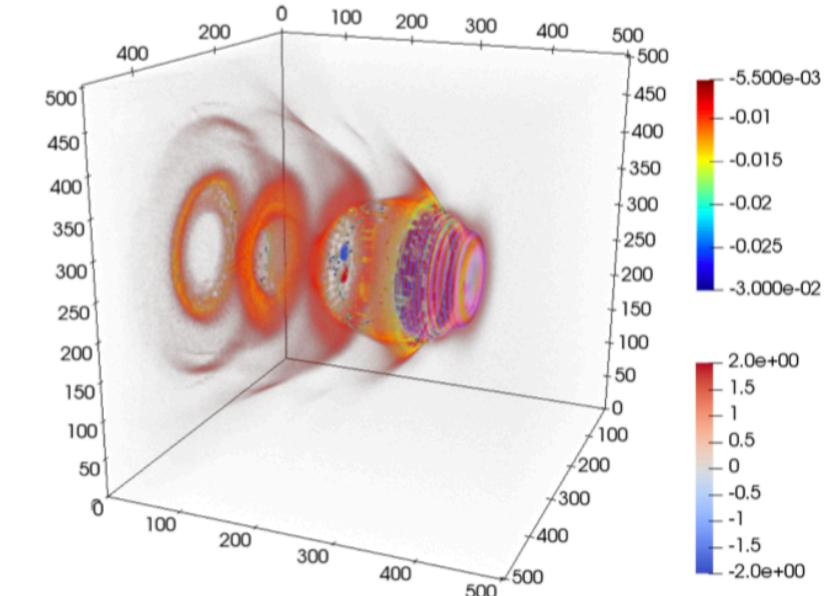
source: Gauss Center for Supercomputing

## Space plasmas & astrophysics



source: SMILEI dev-team

## Laser plasma interaction



source: SMILEI dev-team

- Wide range of physics applications
- Conceptually simple
- Efficiently implemented on small or massively parallel supercomputers

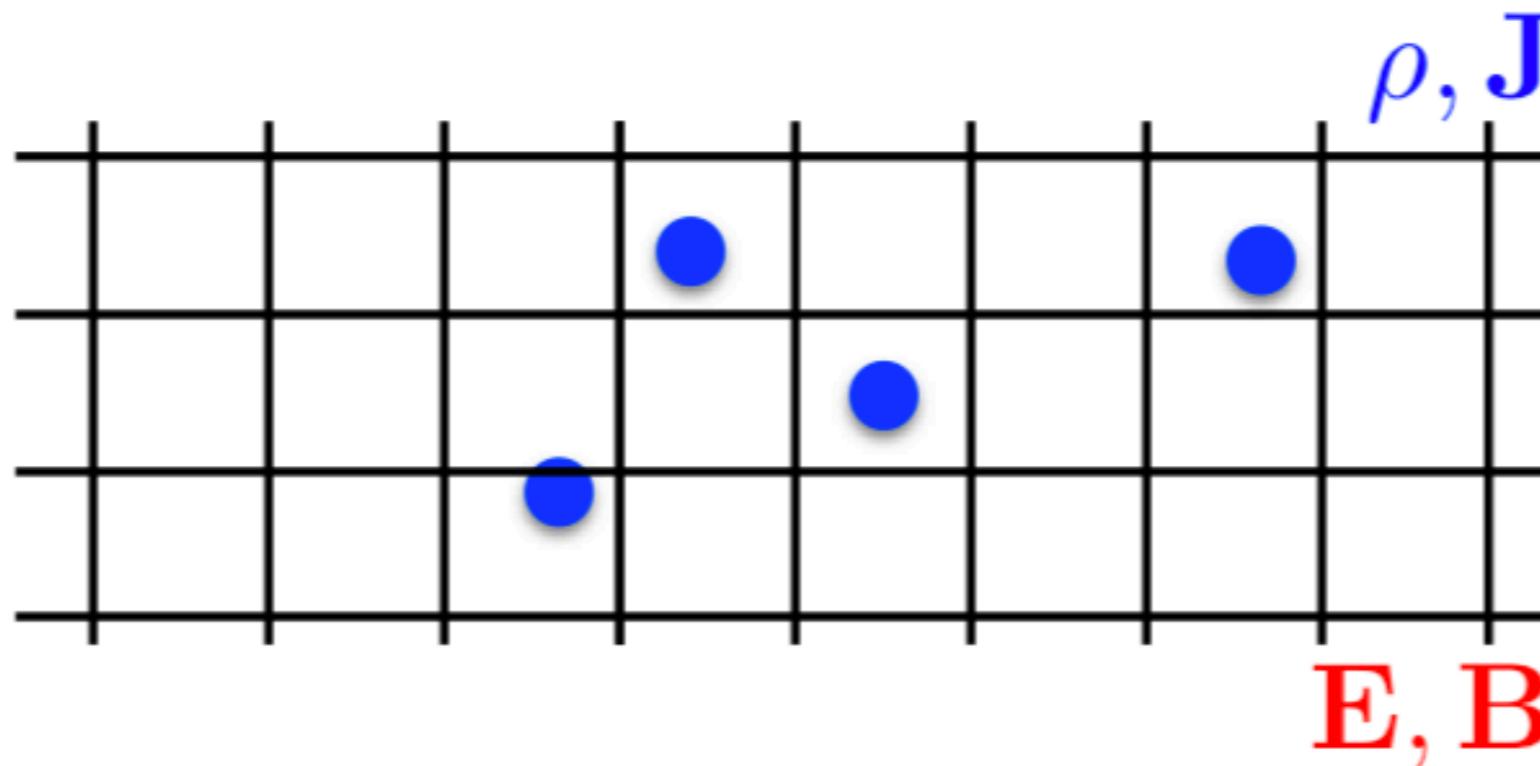
# Particle in Cell concept

Sample Plasma with Macro-Particles  
(1 Macroparticle = position, momentum, charge, ...)

+

Discretize space with computational grid

Define  $E$ ,  $B$ ,  $\rho$ ,  $J$  on the grid cells

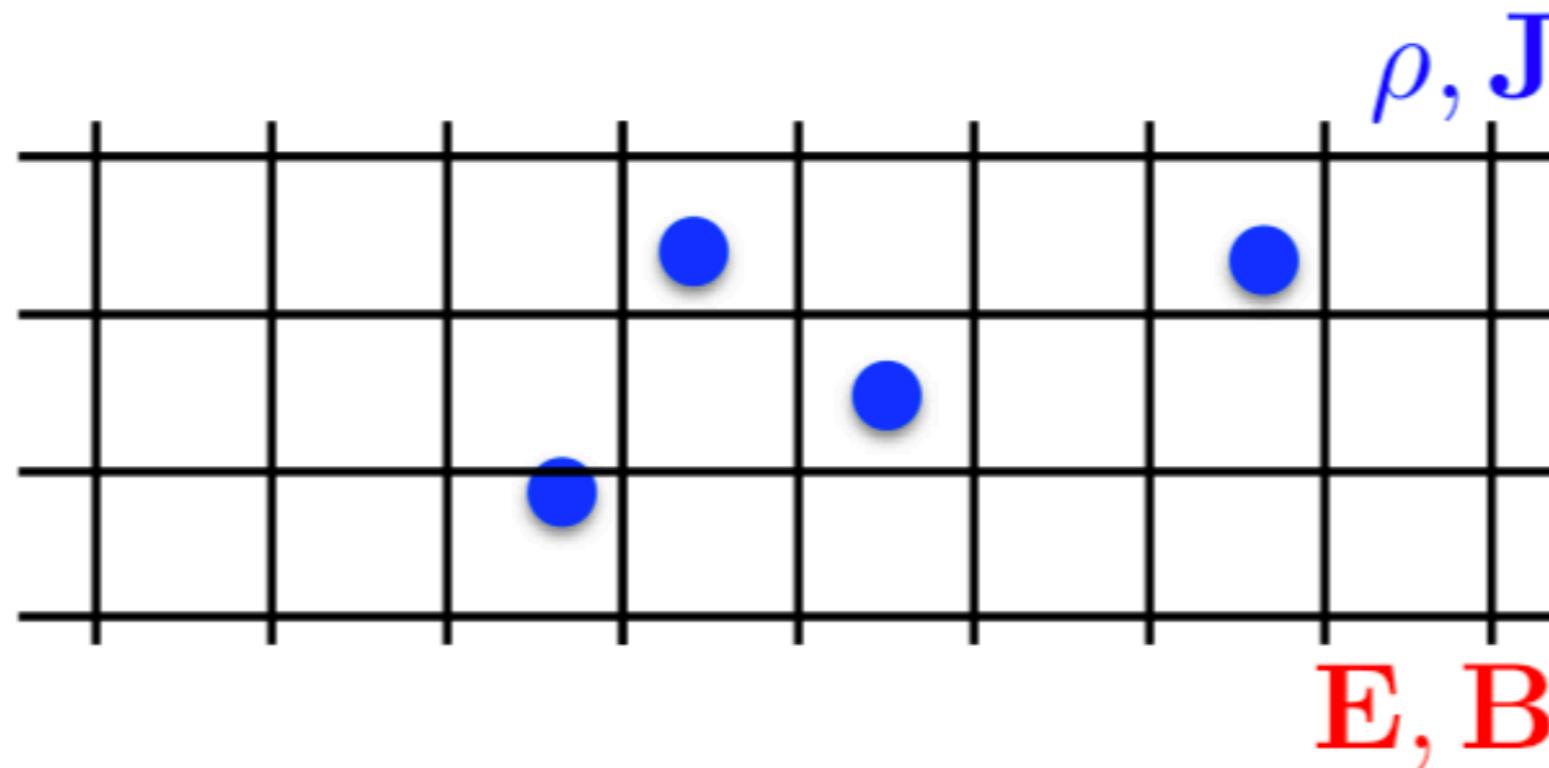


# Particle in Cell concept

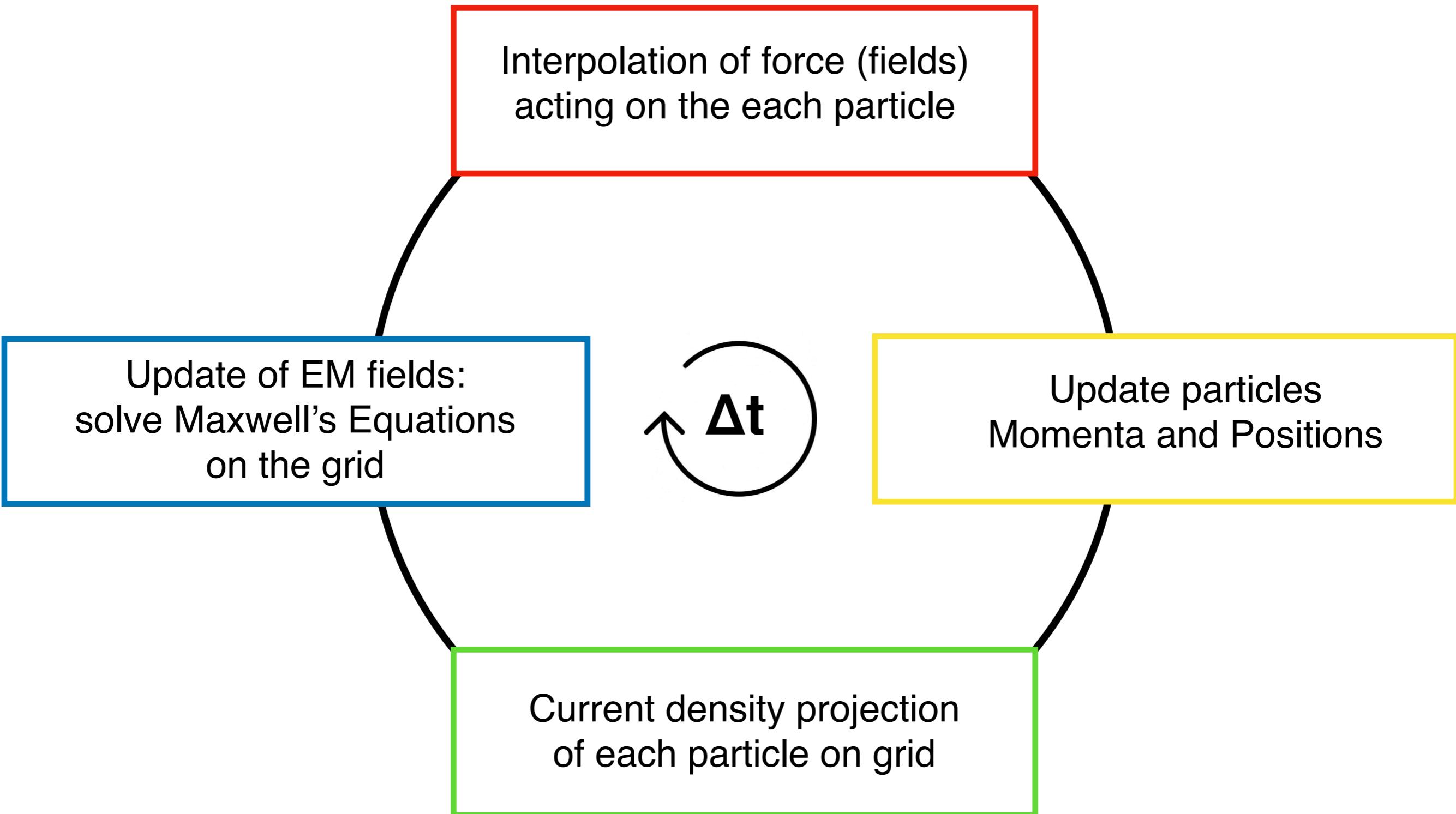
**Particles:** evolve following their equations of motion

+

**Electromagnetic Fields:** evolve following Maxwell's Equations



# Particle in Cell modelling is self-consistent



# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical



A collaborative, open source  
multi-purpose Particle in Cell code  
<https://smileipic.github.io/Smilei/index.html>

## User-friendly

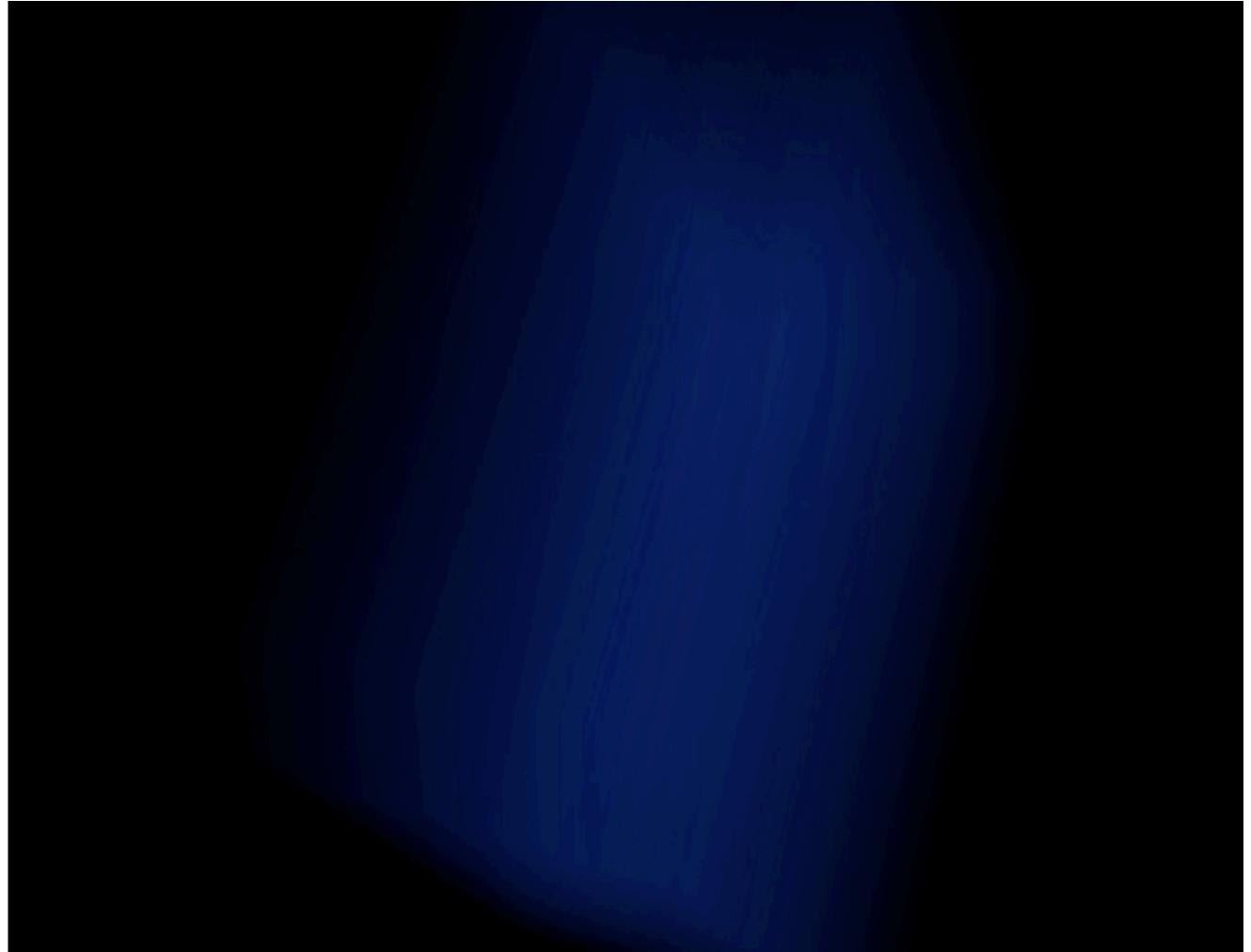
- online: documentation, tutorials
- Python input/output
- quick visualisation library
- teaching platform
- bi-annual training workshop

## High Performance

- MPI + OpenMP parallelization
- dynamic load balancing
- adaptive vectorization

## Multi-physics

- 1D, 2D, 3D, quasi-3D geometries
- ionization, collisions, strong-field QED
- laser envelope model
- relativistic beam field initialization



## High quality

- developers: experts of physics and HPC
- continuously benchmarked
- GitHub bug reporting
- OpenPMD standard

# Additional material

## Extensive Documentation

<https://smileipic.github.io/Smilei/index.html>

( Installation, Use, Postprocessing, ...)



<https://smileipic.github.io/tutorials/>

Registration for the 3rd Training Workshop (9-11 march 2022):

<https://indico.math.cnrs.fr/event/691/>

Presentations from the 2nd Training Workshop <https://indico.math.cnrs.fr/event/4002/>

- Grech PIC codes
- Vinci Collaborative Infrastructure
- Perez User Interface and post-processing
- Lobet Smilei Parallelism
- Massimo Laser Envelope Model
- Zemzemi Fourier Mode Decomposition
- Siminos Generation of intense subcycle pulses using laser-driven wakes

# Outline

- Why plasma accelerators?
- Basics of plasma acceleration
- Numerical simulation of plasma acceleration: PIC codes
- The PIC code Smilei
- Introduction to the case study and the practical

# Normalised units for the Vlasov Maxwell system

Velocity

$$c$$

Charge

$$e$$

Mass

$$m_e$$

Momentum

$$m_e c$$

Energy

$$m_e c^2$$

Time

$$1/\omega_r$$

Length

$$c/\omega_r$$

Number Density

$$\epsilon_0 m_e \omega_r^2 / e^2$$

Electric Field

$$m_e \omega_r c / e$$

Magnetic Field

$$m_e \omega_r / e$$

Fixed

Need to  
choose  $\omega_r$

For the practical (input and output), we choose  $\omega_r = 2\pi c/\lambda_0$   
corresponding to  $\lambda_0 = 0.8 \mu\text{m}$  (Ti:Sa laser system)

# Example of input file

```
Main(  
    geometry      = "1Dcartesian",  
    timestep      = 0.009,  
    cell_length   = [0.01],  
    ...  
)
```

```
x_center_plasma = 200.
```

```
def my_density_profile(x):  
    return exp(-(x-x_center_plasma)**2)
```

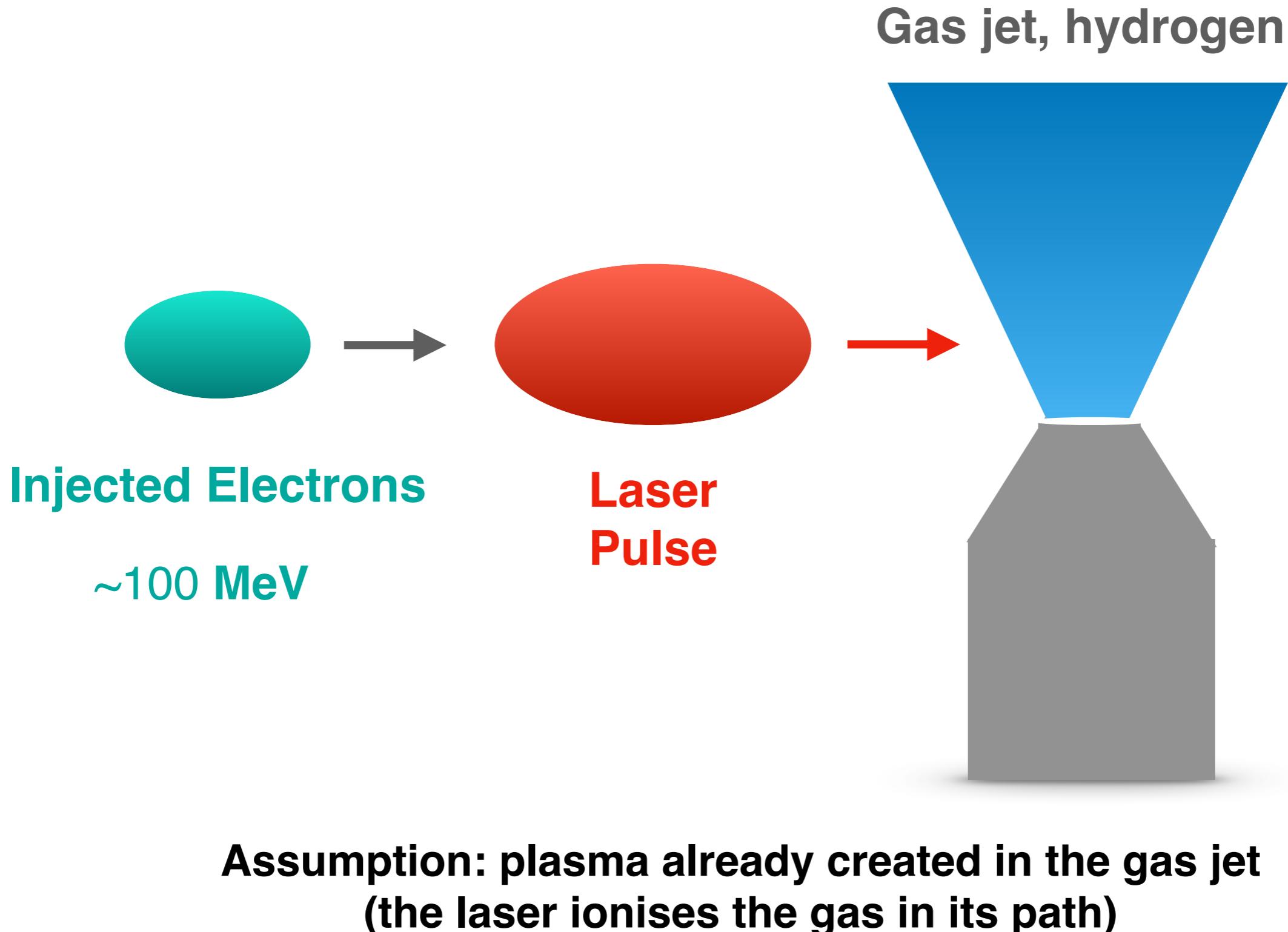
```
Species(  
    name          = "electron",  
    charge        = -1.,  
    mass          = 1.,  
    particles_per_cell = 100,  
    number_density = my_density_profile  
    ...  
)
```

- Normalised units
- Quantities can be computed at runtime

Laser/Plasma profiles = Functions  
(also user-defined)

If Python can read it, SMILEI can read it

# Case study: LWFA with external injection



# Practical lesson: how does it work?

- Read **in detail** the handouts in **Handouts.pdf**
- Set-up the machine following the instructions in sections 1-3 of **Instructions\_Cluster.pdf**
- Solve the exercises in **Handouts.pdf** progressively
- Run simulations when necessary, following the instructions in **Handouts.pdf** (blocks to uncomment etc.)  
(for a guide to run and analyse simulations, sections 3,4,5 of **Instructions\_Cluster.pdf**)
- Fill the answers report

# Suggestions

- Again read **in detail** the handouts
- **Understand** the physical set-up
- **Any doubts? Ask the instructor**
- Solve the exercises **progressively**
- Create one folder for each simulation asked by the exercises  
**(to avoid losing data)**
- Copy, paste and adapt the post processing commands in the handouts
- When an image is asked for the report, save it or make a screenshot
- **Better do few exercises but understand them at 100%**

# Your resources

- Code [Smilei](#) + Postprocessing library [happi](#)
- Input file for the simulation:
  - [InputNamelist.py](#)
- 5 Postprocessing scripts:
  - [Laser\\_waist\\_theory\\_vs\\_Smilei.py](#)
  - [Ex\\_linear\\_theory\\_vs\\_Smilei.py](#)
  - [Compute\\_bunch\\_parameters.py](#)
  - [Follow\\_electron\\_bunch\\_evolution.py](#)
  - [Export\\_data.py](#) (needed if you want to export output to .txt files)
  - [Save\\_VTK.py](#) (not mandatory for the TP)
- Instructions for the configuration of the machine:
  - [Instructions\\_Cluster.pdf](#)
- Handouts with instructions for the practical ([Handouts.pdf](#)) with:
  - Physical explanations
  - Postprocessing instructions
  - Exercises

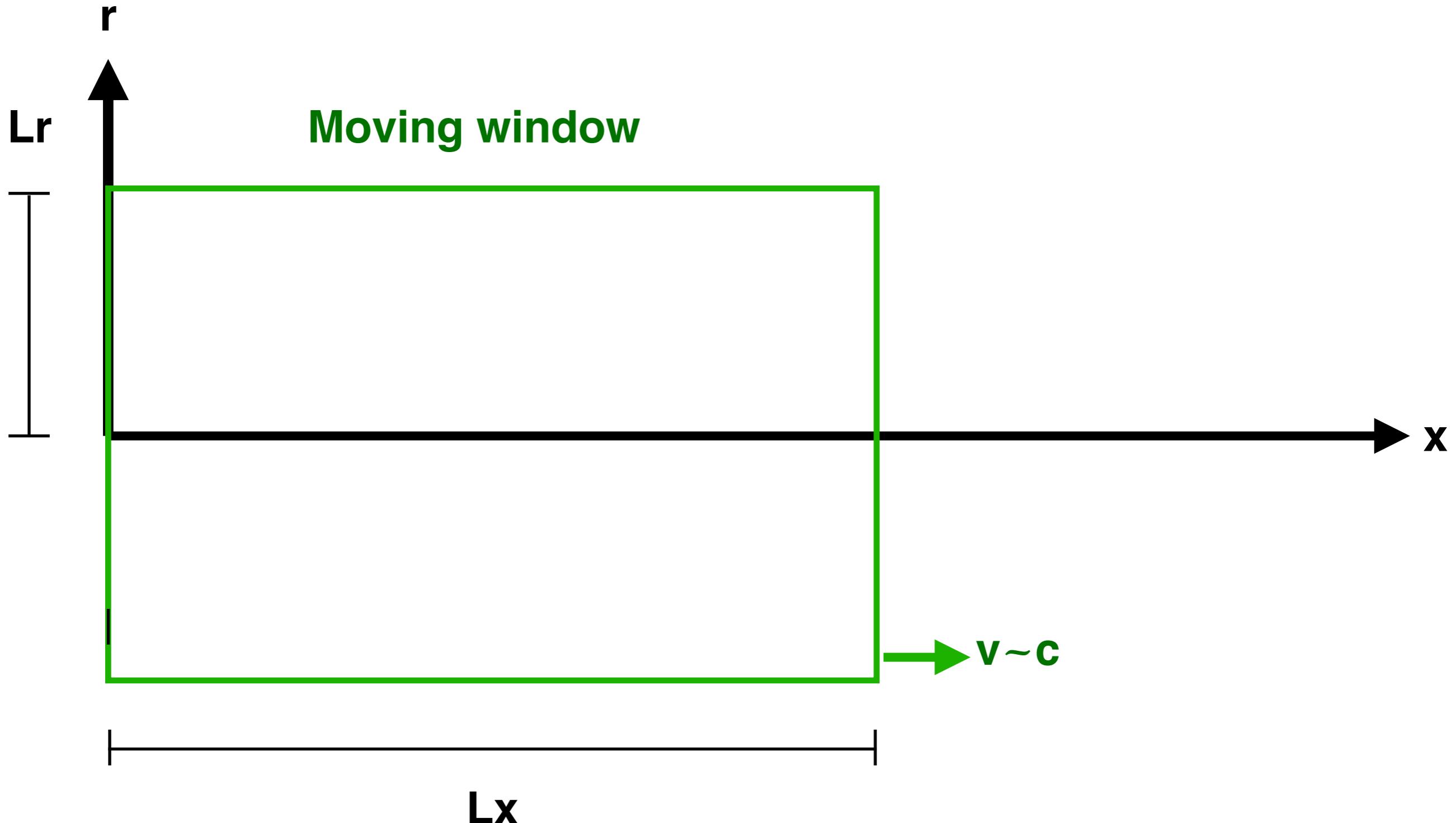
# Structure of the input file for the practical

- **Main Block**
- **Moving Window Block**
- **Laser Envelope Block**
- **Species Block (for the plasma)**
- **Species Block (for the relativistic electron bunch)**
- **Diagnostic Block (for Postprocessing)**

**Blue: Active**

**Gray: Commented = Inactive**

# Simulation setup: simulation window



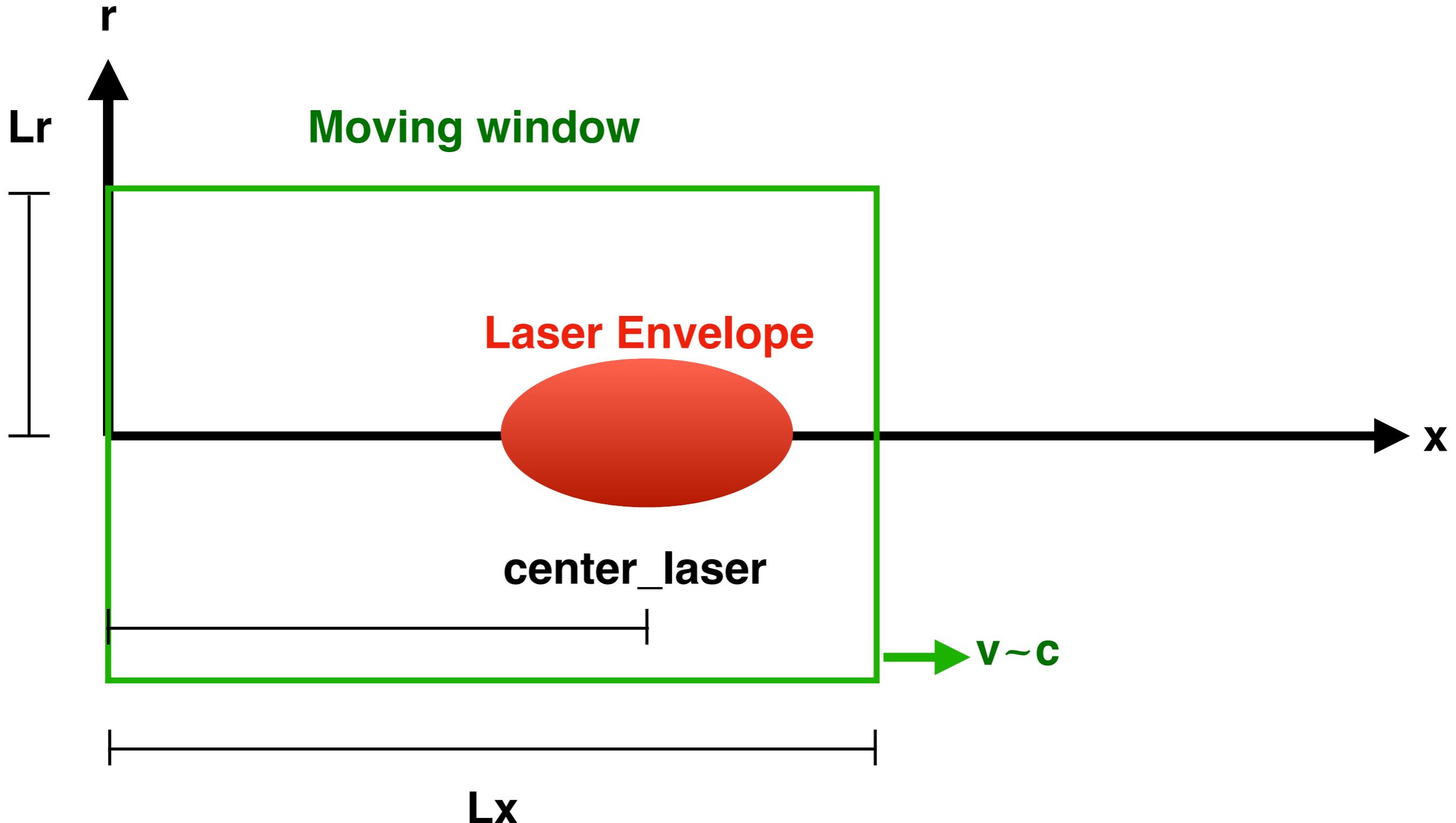
# Structure of the input file for the practical

- **Main Block**
- **Moving Window Block**
- **Laser Envelope Block**
- Species Block (for the plasma)
- Species Block (for the relativistic electron bunch)
- **Diagnostic Block (for Postprocessing)**

**Blue: Active**

**Gray: Commented = Inactive**

# Simulation setup: laser in vacuum



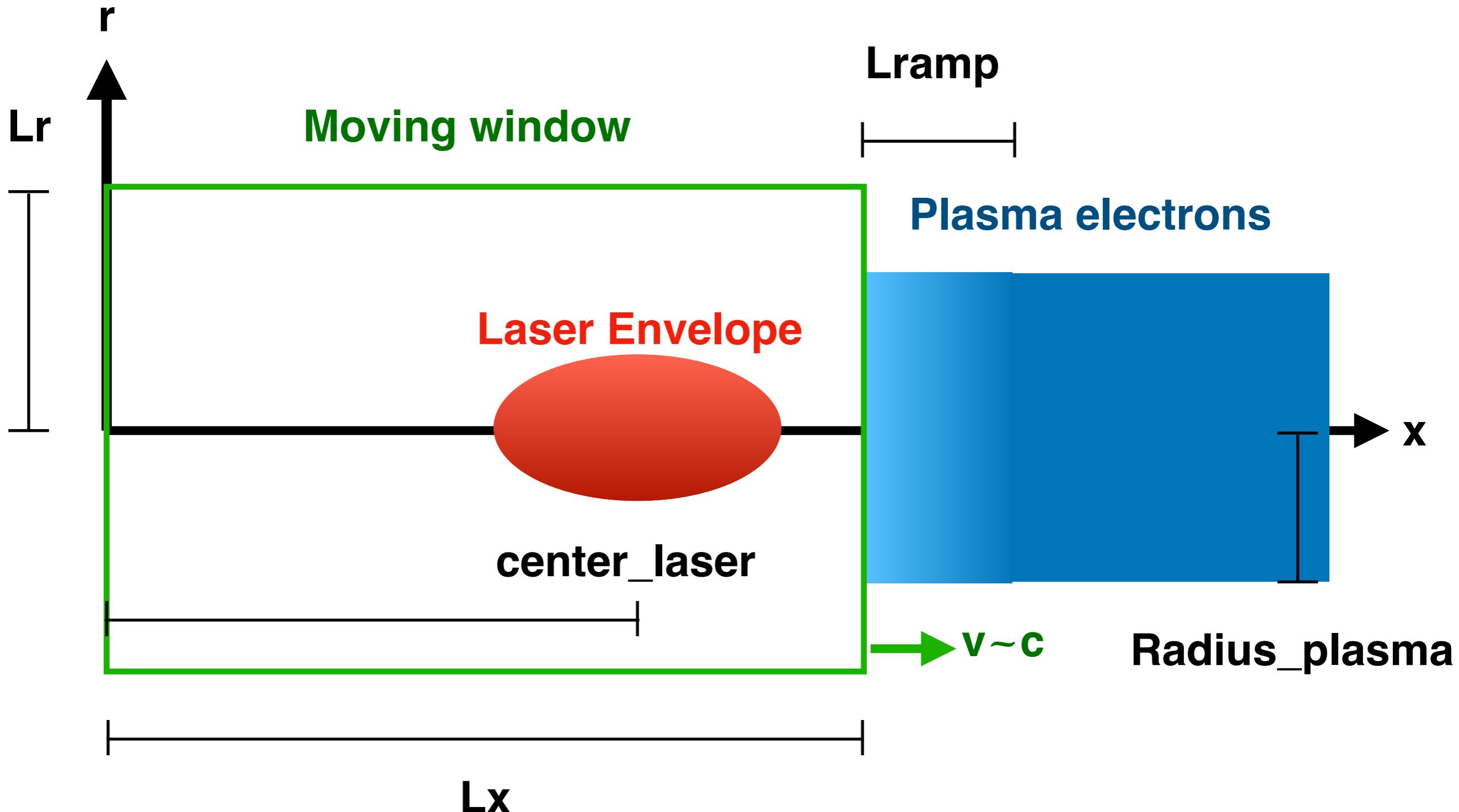
# Structure of the input file for the practical

- **Main Block**
- **Moving Window Block**
- **Laser Envelope Block**
- **Species Block (for the plasma)**
- Species Block (for the relativistic electron bunch)
- **Diagnostic Block (for Postprocessing)**

**Blue: Active**

**Gray: Commented = Inactive**

# Simulation setup: laser in plasma



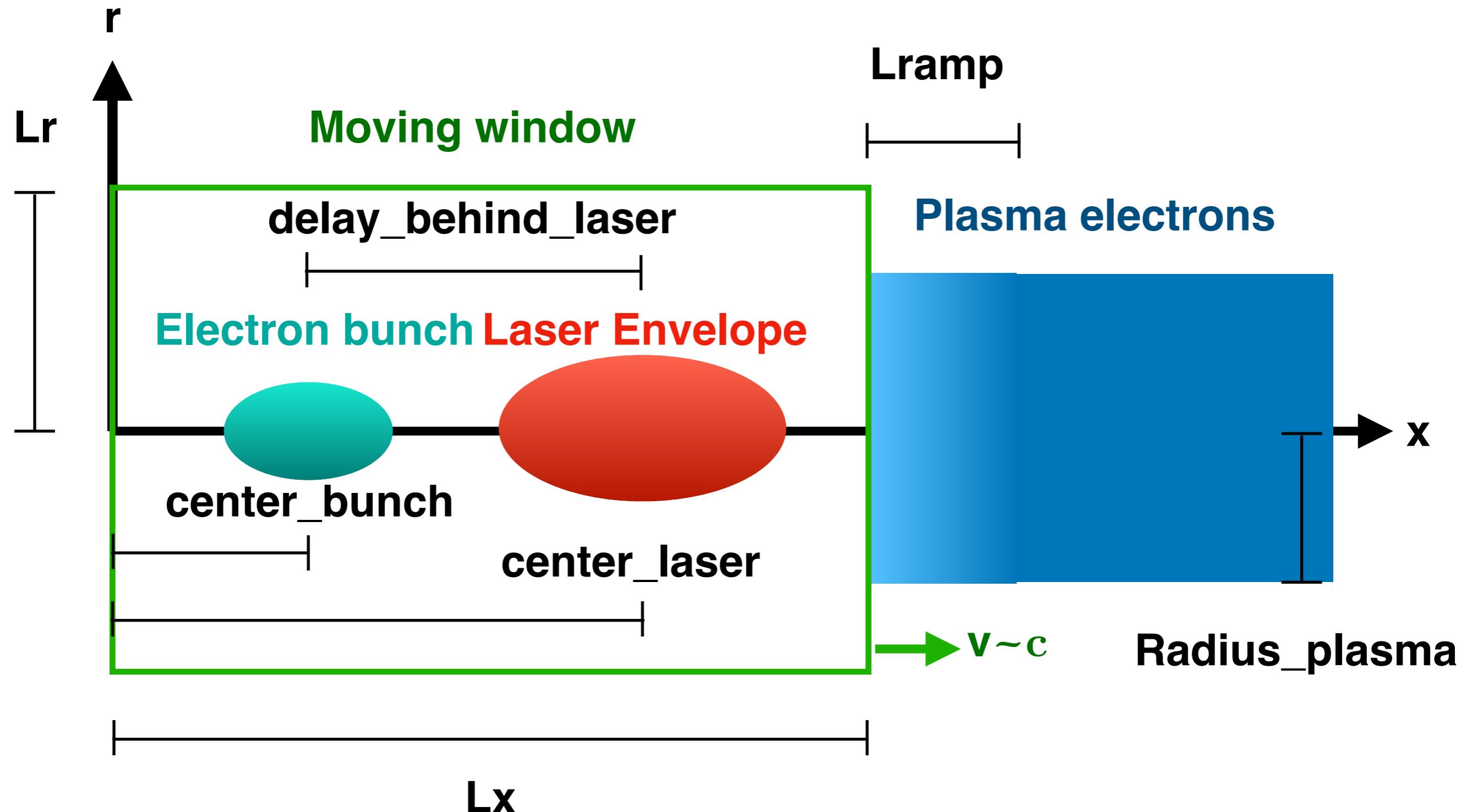
# Structure of the input file for the practical

- **Main Block**
- **Moving Window Block**
- **Laser Envelope Block**
- **Species Block (for the plasma)**
- **Species Block (for the relativistic electron bunch)**
- **Diagnostic Block (for Postprocessing)**

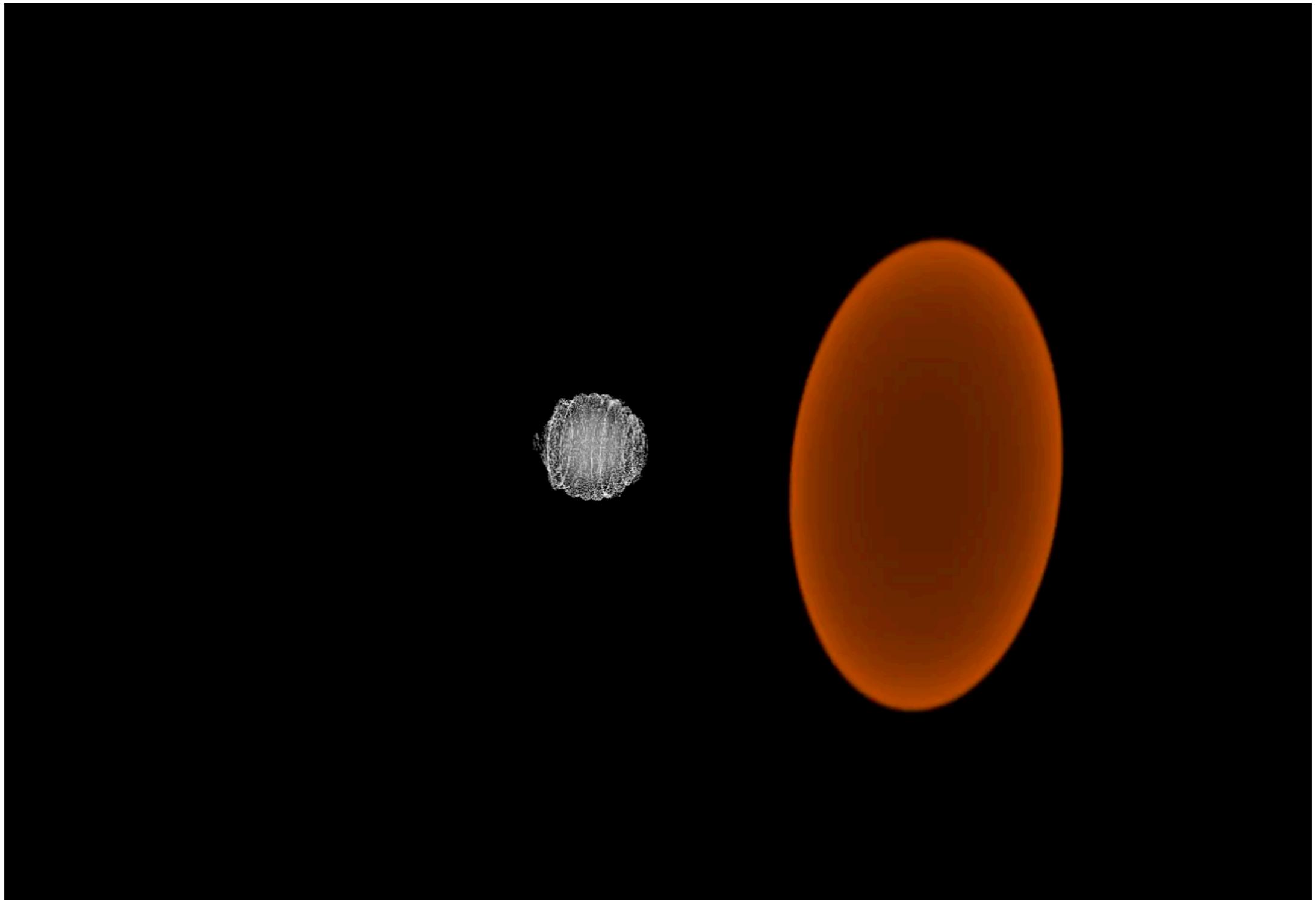
**Blue: Active**

**Gray: Commented = Inactive**

# Simulation setup: LWFA with external injection



# 3D rendering of LWFA with external injection



# Postprocessing for the practical

Included in the code you have a Python postprocessing library

```
$ make happy  
$ ipython  
In [1]: import happy
```

**Diagnostics available for this practical:**

- 1D Probe on the x axis (electromagnetic fields, density)
- 2D Probe on the xy plane (electromagnetic fields, density)
- DiagTrackParticles for the bunch electrons (phase space data)

**Tips:**

- Open the Ipython interface
- Copy and paste the happy commands from the handouts
- Adapt them for your purposes
- Use the post processing scripts available (see handouts)

# **Questions?**

# Additional Slides

# Exercise 10: Nonlinear plasma waves, density

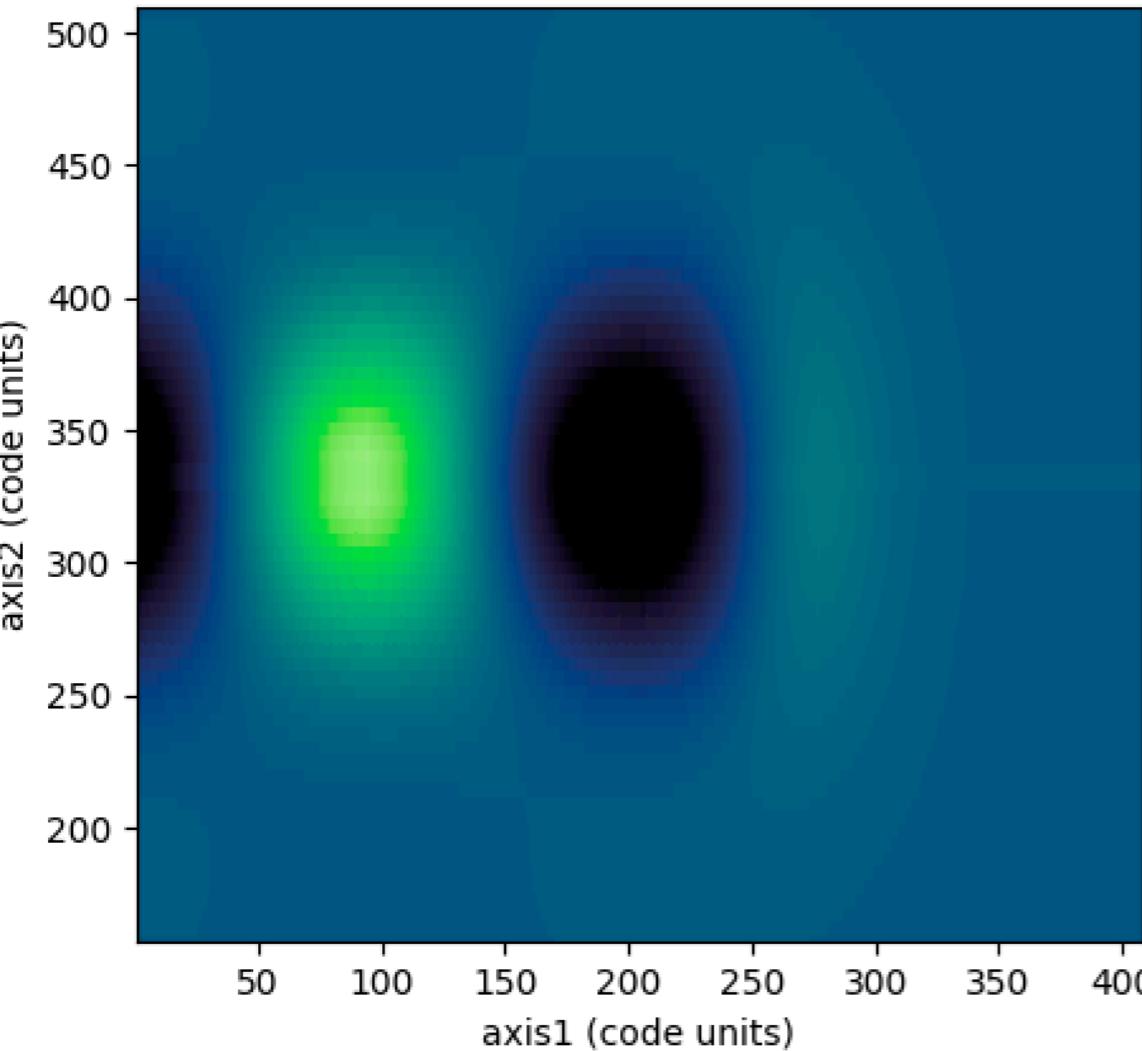
$a_0 \ll 1$  :  
Linear regime

(Sinusoidal plasma waves)

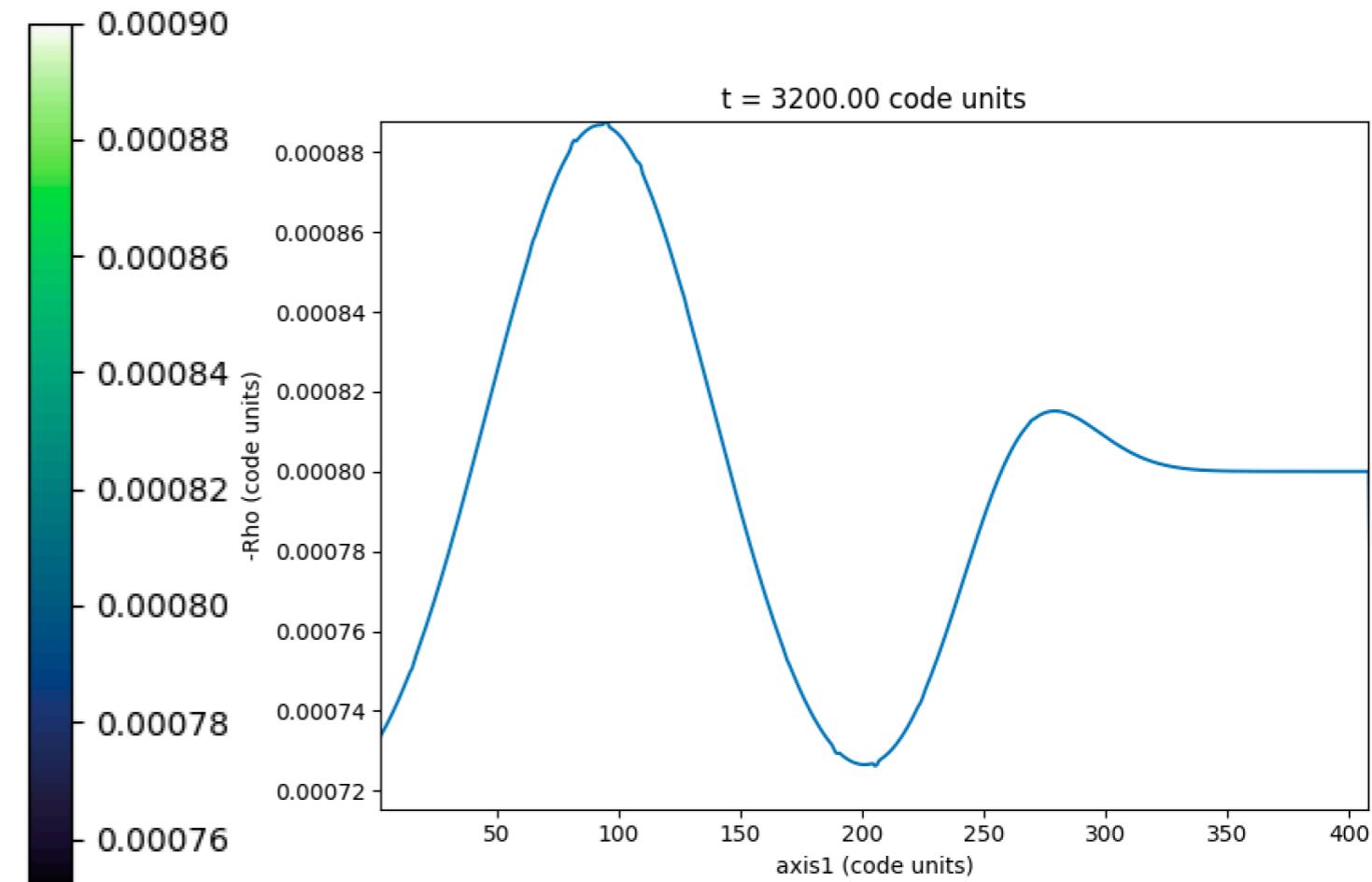
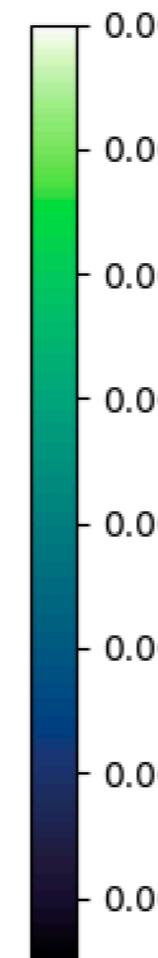
Laser Pulse propagation along the x  
axis



-Rho (code units) t = 3200.00 code units



2D charge density



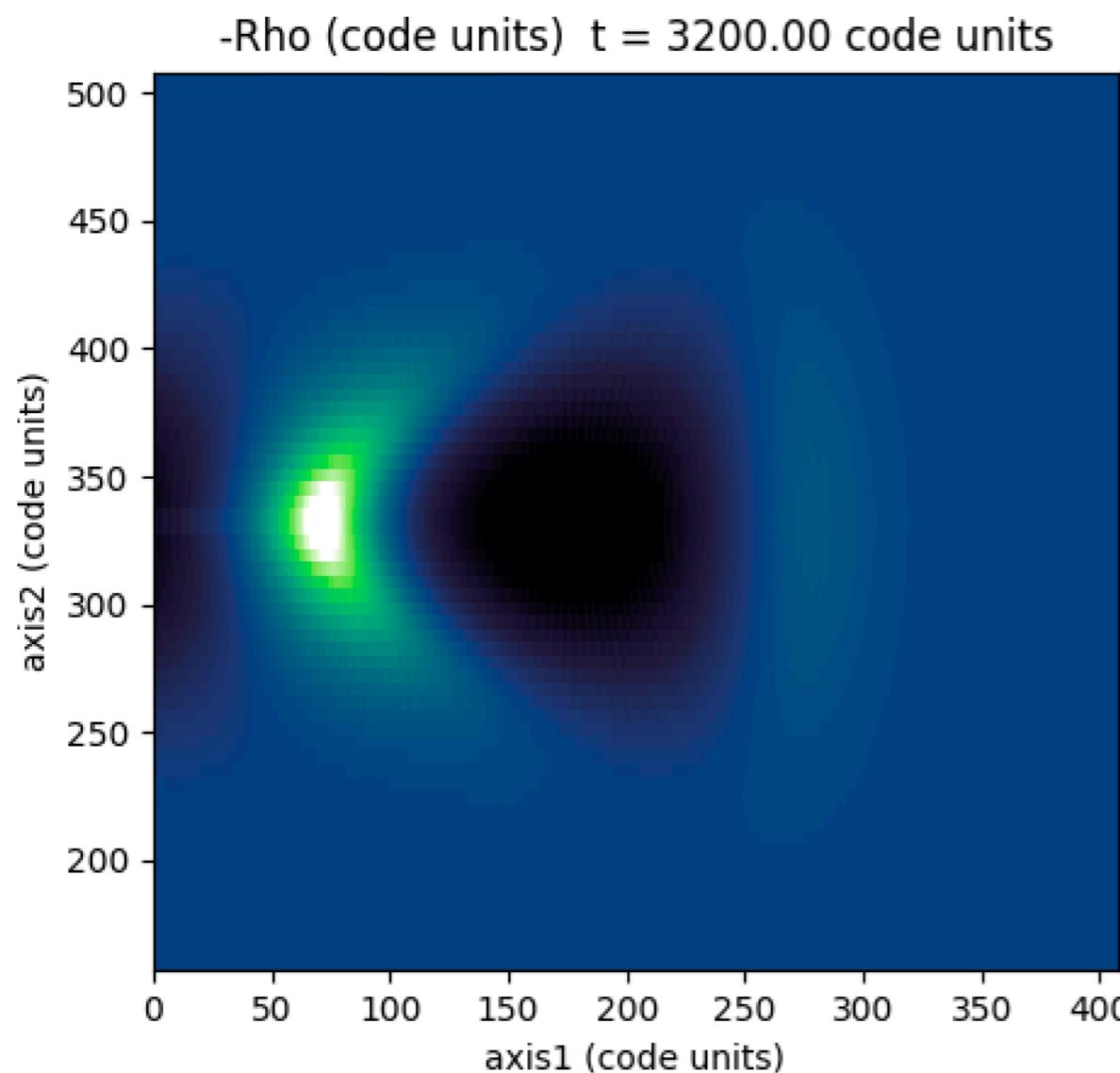
1D charge density  
on propagation axis

# Exercise 10: Plasma waves regimes, density

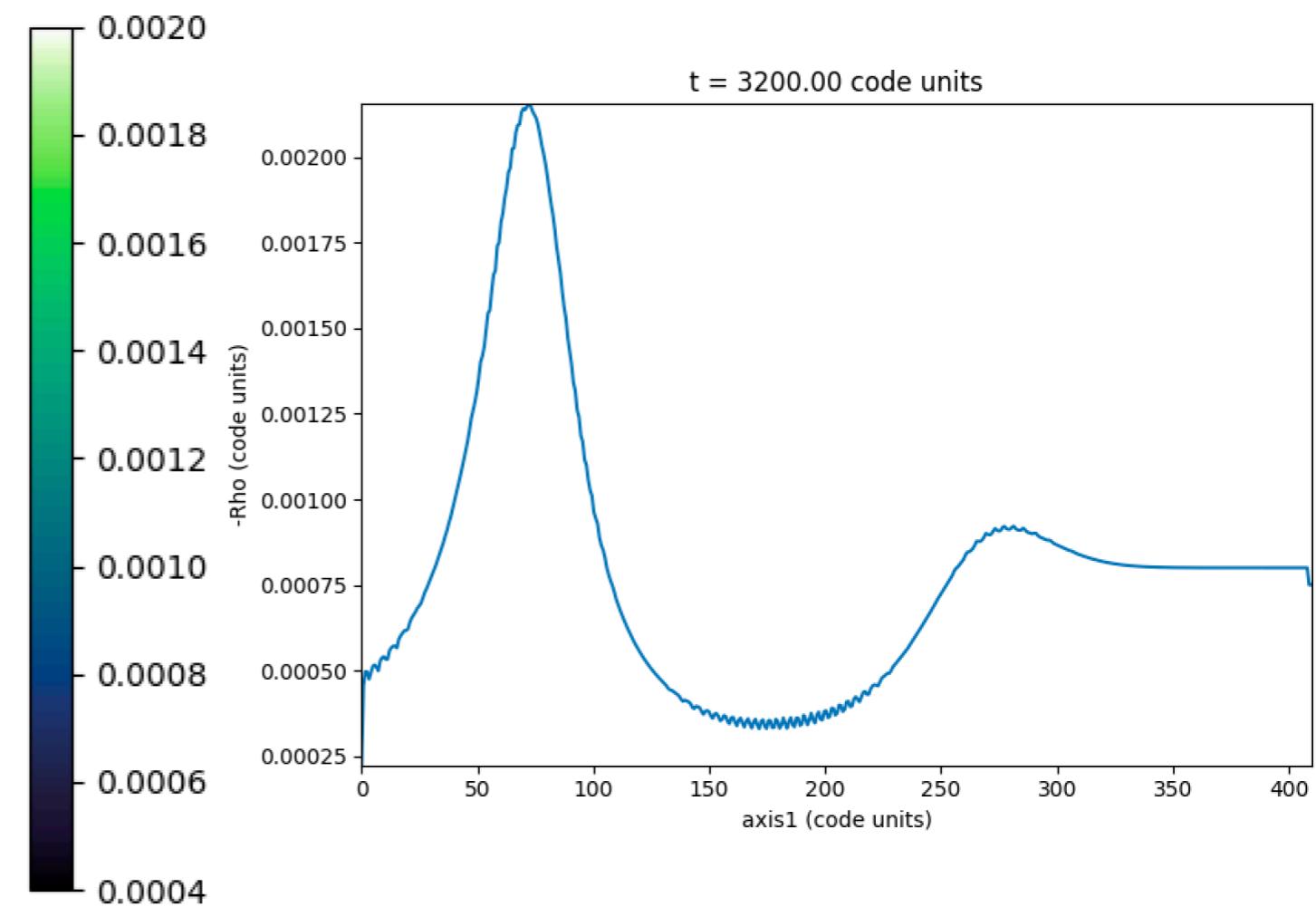
$a_0 \lesssim 1$  :  
Weakly nonlinear regime



Laser Pulse propagation along the x axis



2D charge density



1D charge density  
on propagation axis

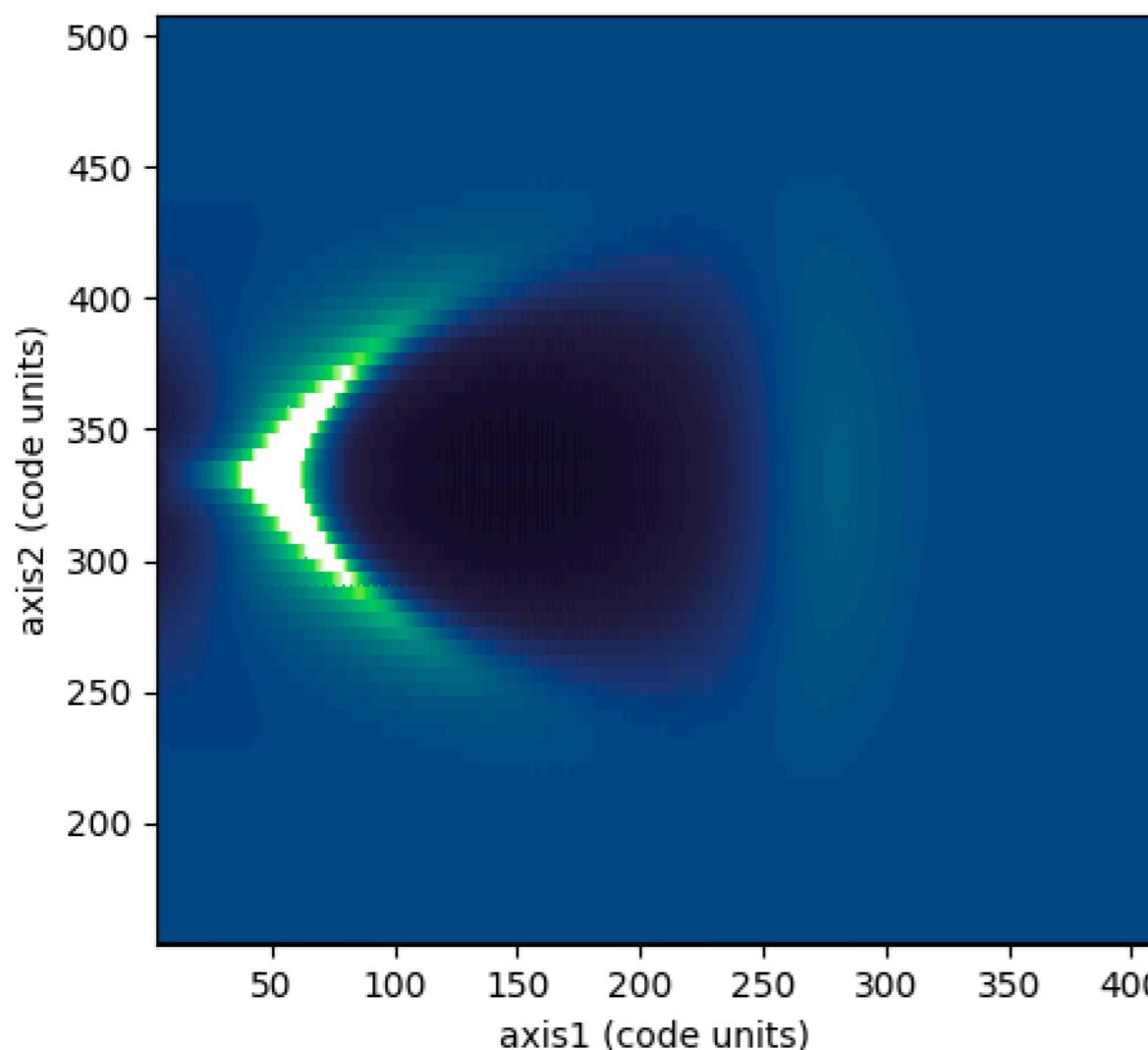
# Exercise 10: Plasma waves regimes, density

$a_0 > 1 :$

**Nonlinear regime  
(Bubble-like waves)**

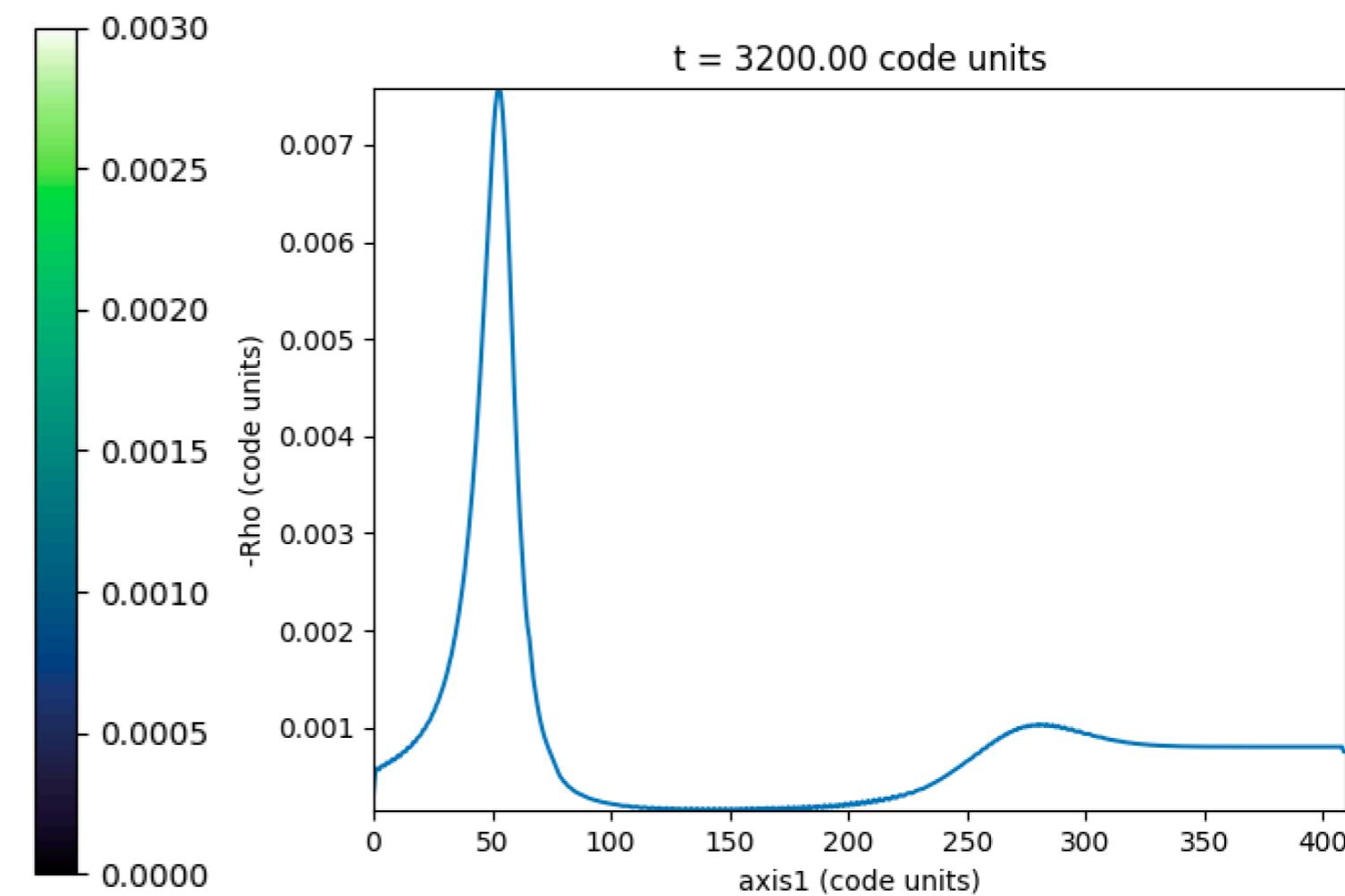
**Laser Pulse propagation along the x  
axis**

-Rho (code units)  $t = 3200.00$  code units



**2D charge density**

$t = 3200.00$  code units



**1D charge density  
on propagation axis**

# Exercise 10: Plasma waves regimes, Ex on axis

```
import happy
S1 = happy.Open("path/to/sim1"); Ex1 = S1.Probe.Probe0("Ex",timesteps=1000)
S2 = happy.Open("path/to/sim2"); Ex2 = S2.Probe.Probe0("Ex",timesteps=1000)
S3 = happy.Open("path/to/sim3"); Ex3 = S3.Probe.Probe0("Ex",timesteps=1000)
happy.multiPlot(Ex1,Ex2,Ex3,figure=3)
```

