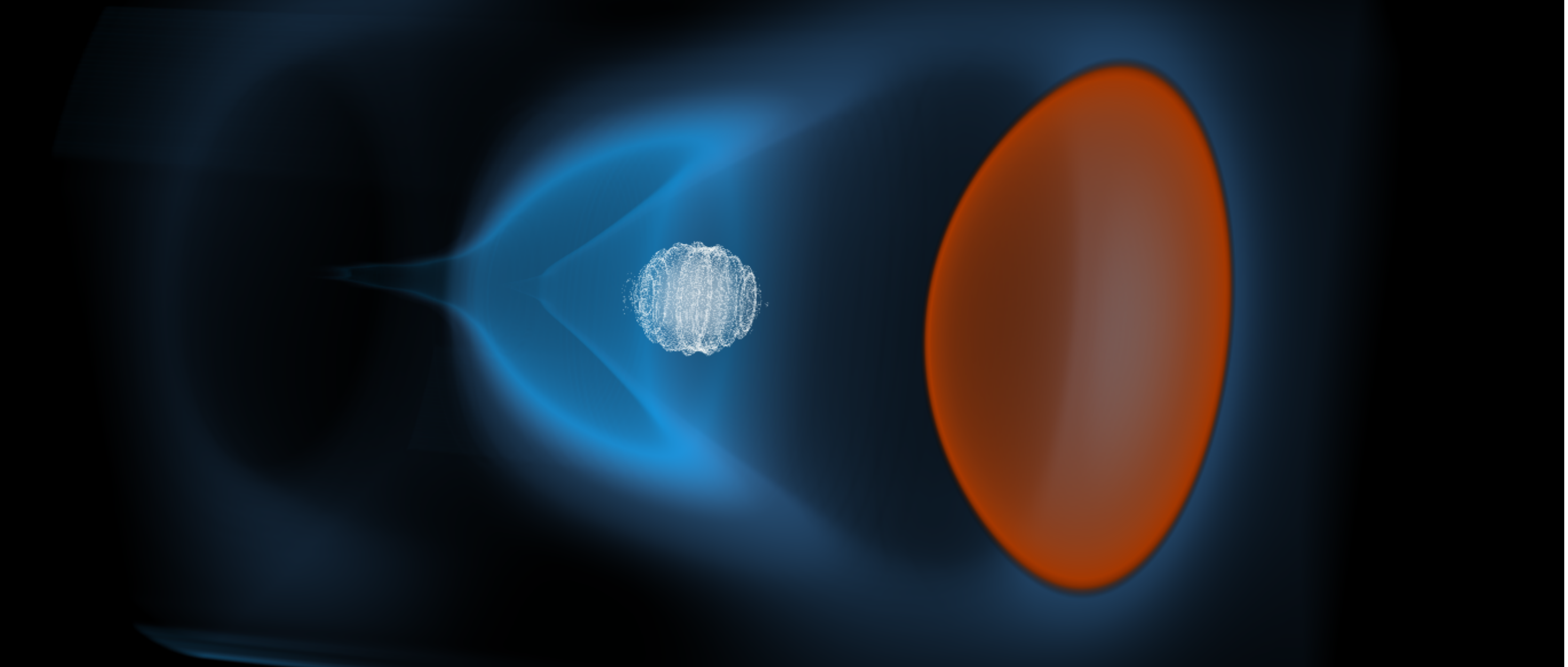
Numerical simulation of laser wakefield acceleration with a PIC code



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)

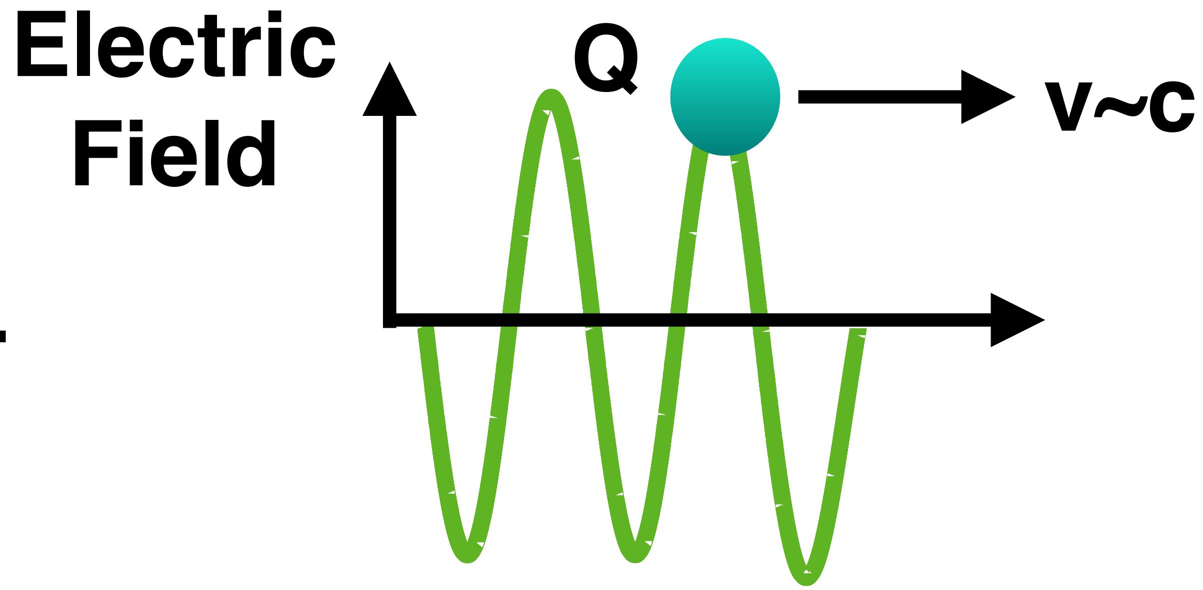
Science:

- Particle physics
- Nuclear Physics
- Matter Physics
- Biology

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**

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

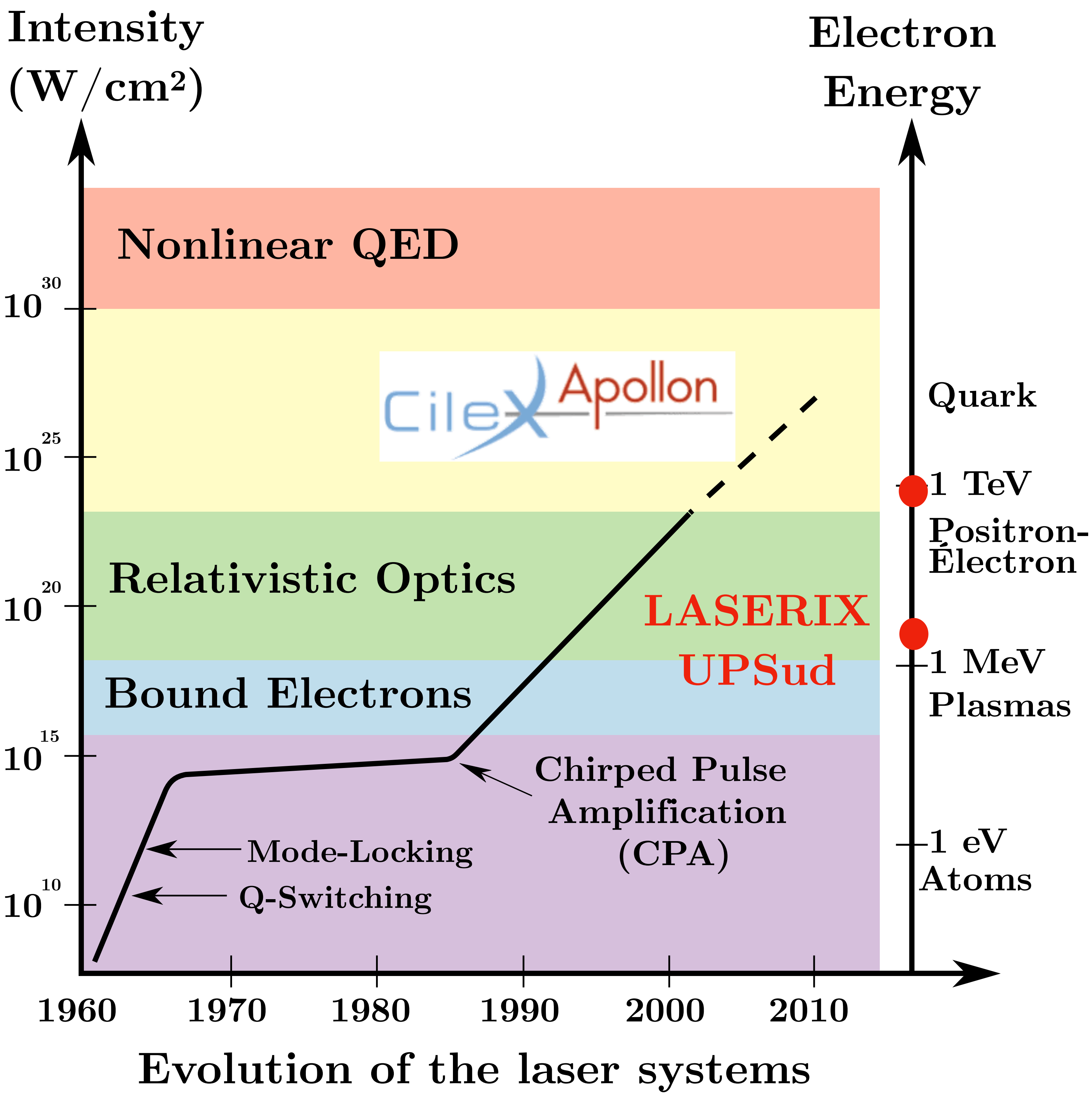
$$E_{max} > 100\ \text{GV/m}$$

→ **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**
- …

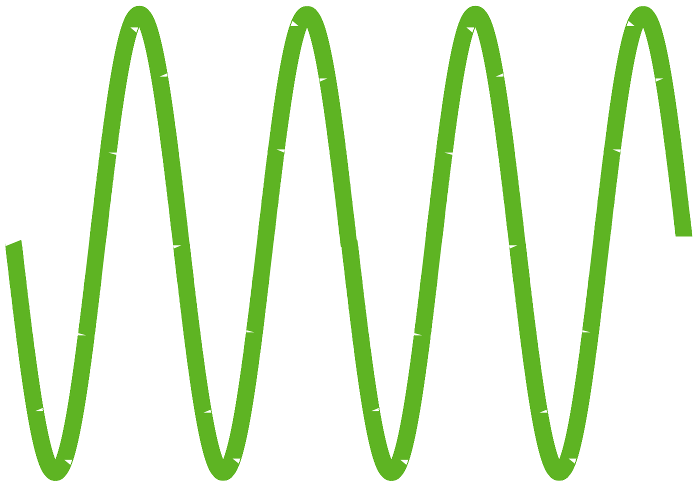
Physics Domains:

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

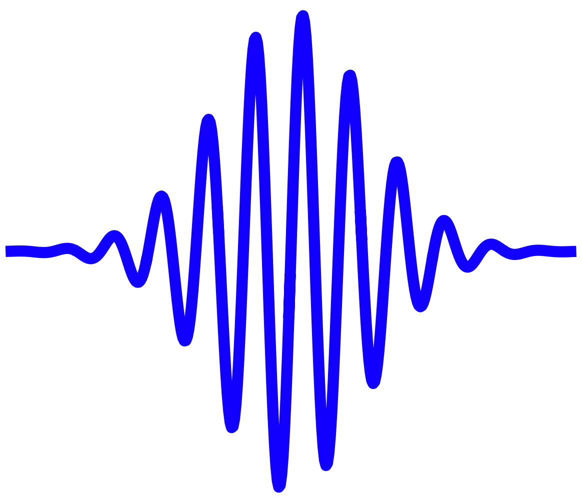
Applications:

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

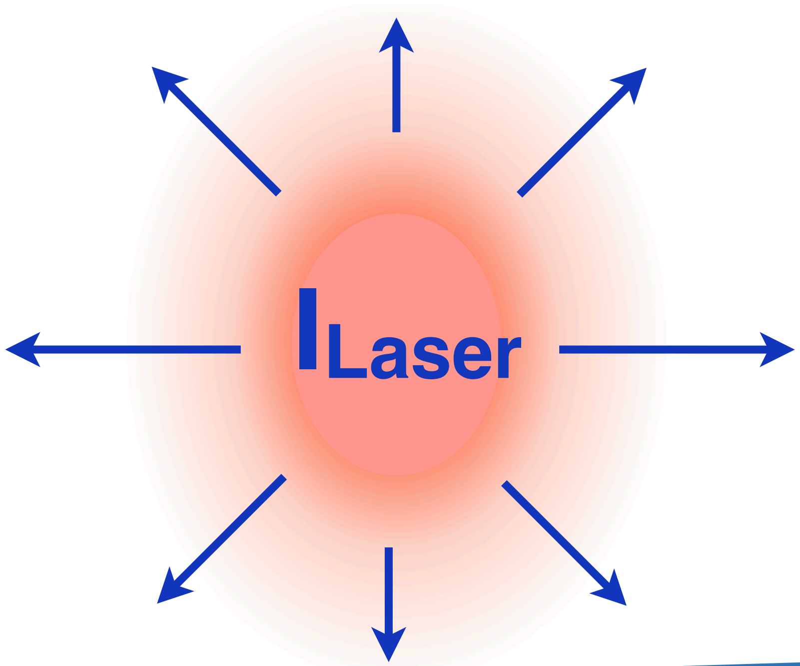
Ponderomotive Force



Electron in infinite plane wave:
Oscillating Force



Electron in finite laser pulse:
Oscillating Force + Ponderomotive Force



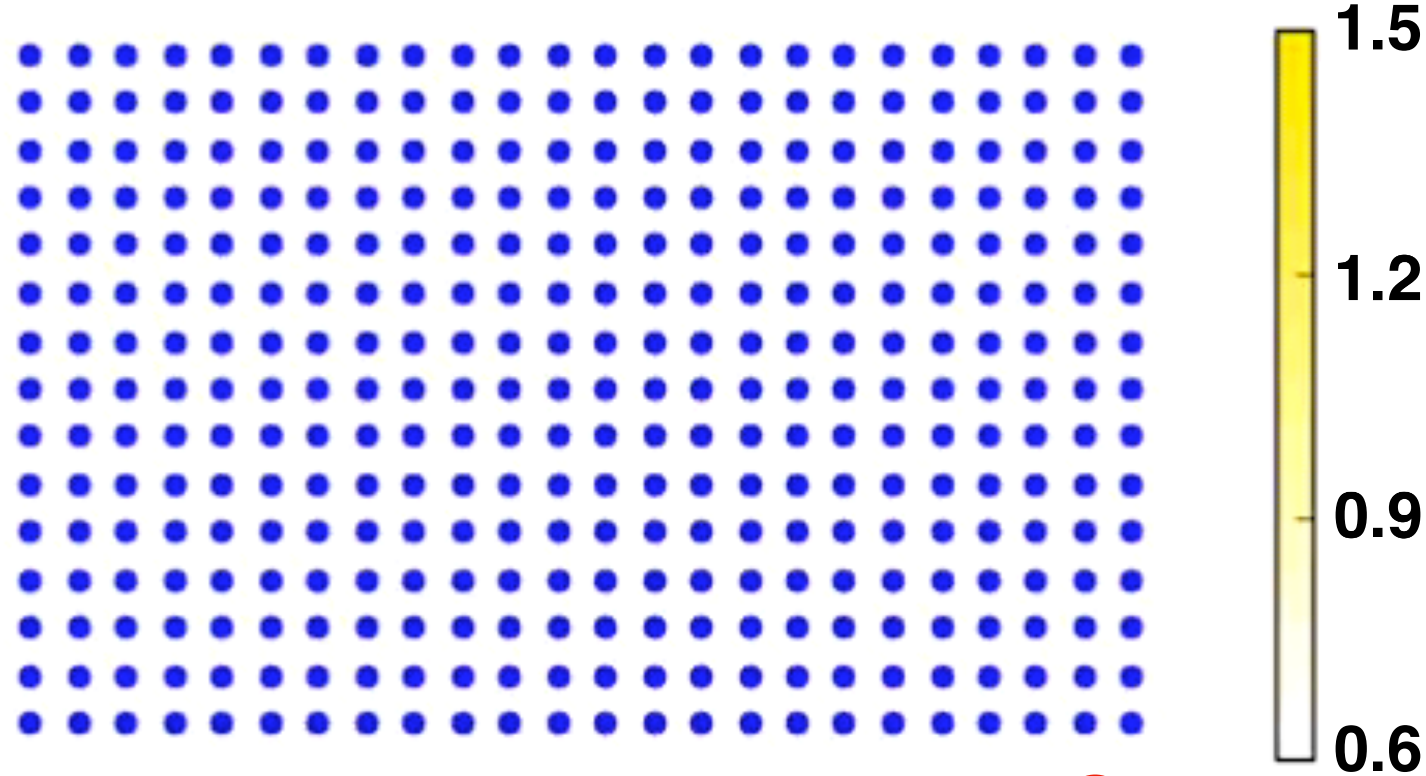
$F_{pond} \propto -\nabla I_{Laser}$ **(a.k.a. radiation pressure)**

Laser Wakefield Acceleration (LWFA): plasma wave excitation

Laser Beam Duration: 28 fs

Ponderomotive Force:

$F = -\nabla I_{Laser}$

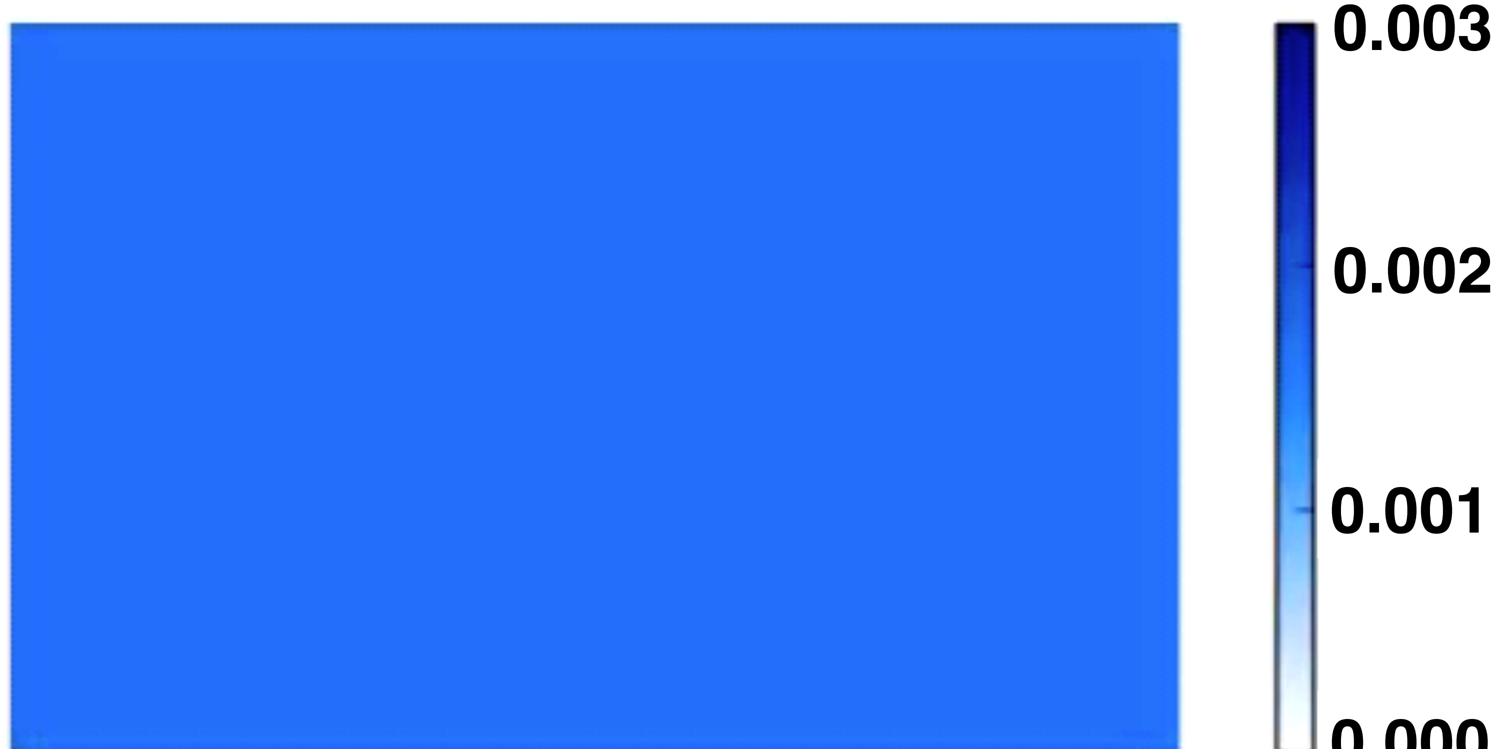


Transverse Electric Field (A.U.)

Electrons • **Immobile Ions** ● $M_i \gg m_e$

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

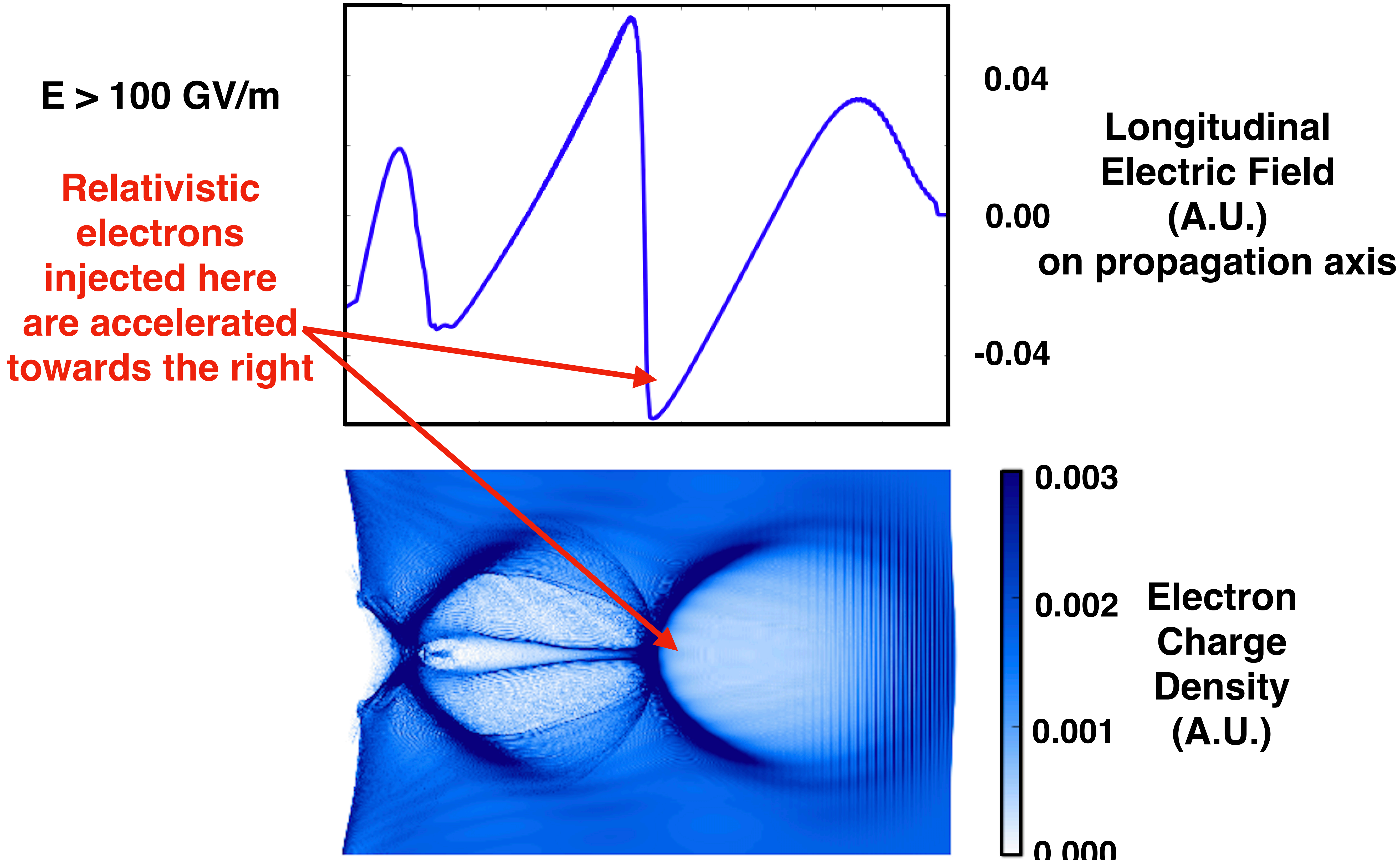
Plasma wavelength: ~20 μm



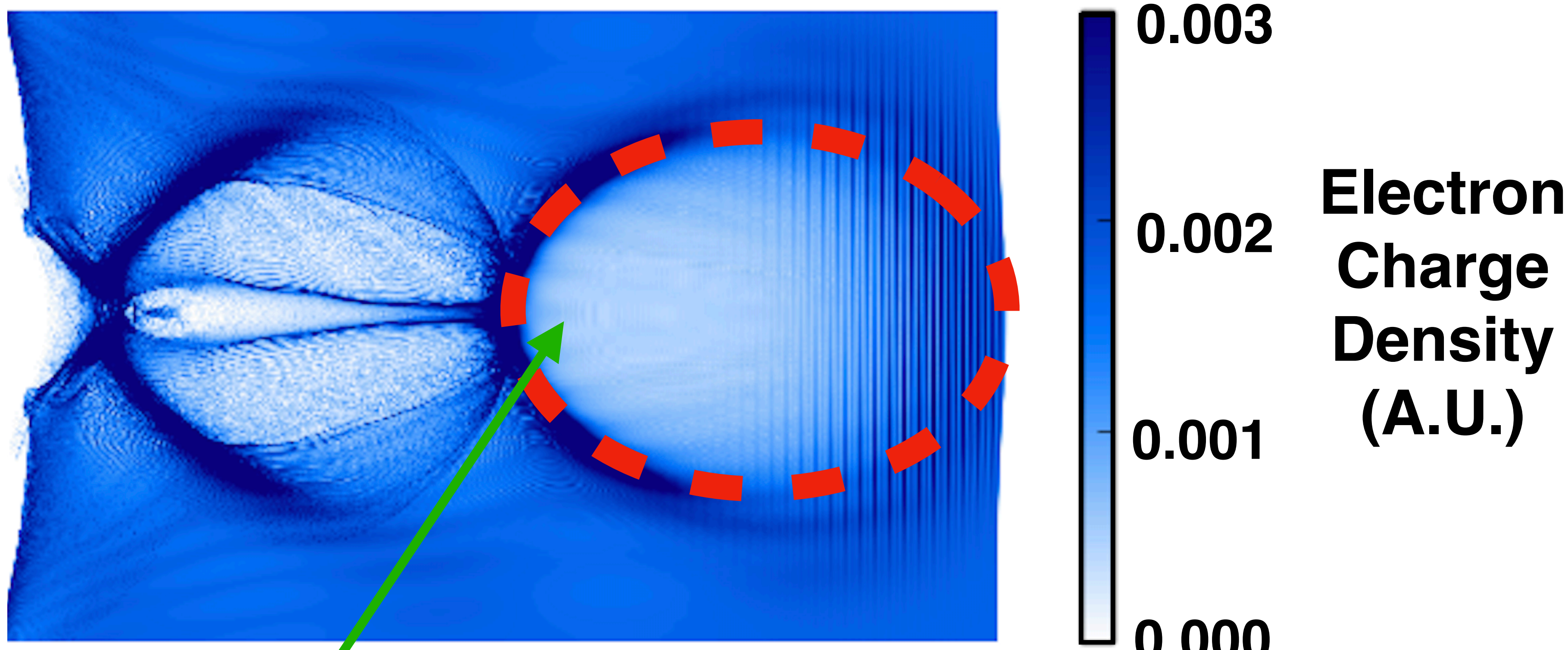
Electron Charge Density (A.U.)

Laser Wakefield Acceleration (LWFA): accelerating electric field

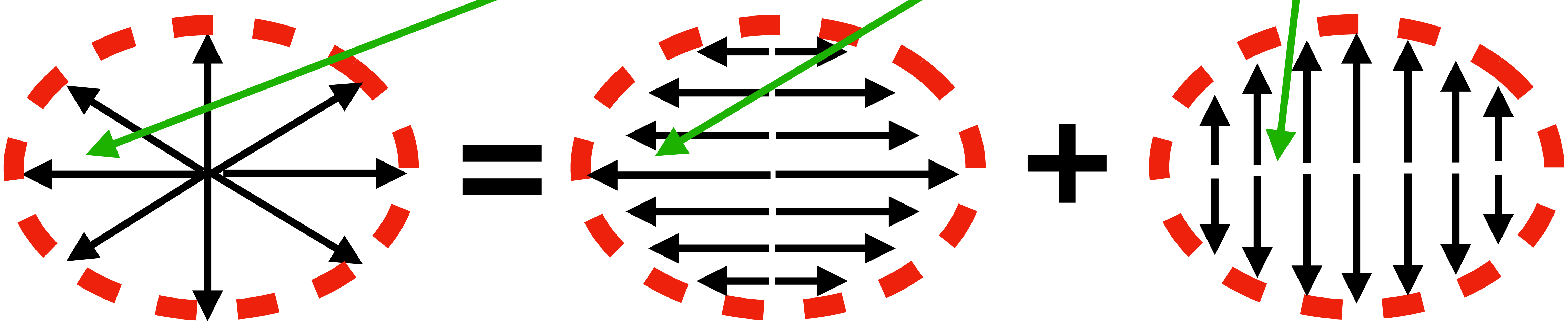
E > 100 GV/m



Laser Wakefield Acceleration (LWFA): fields inside the bubble

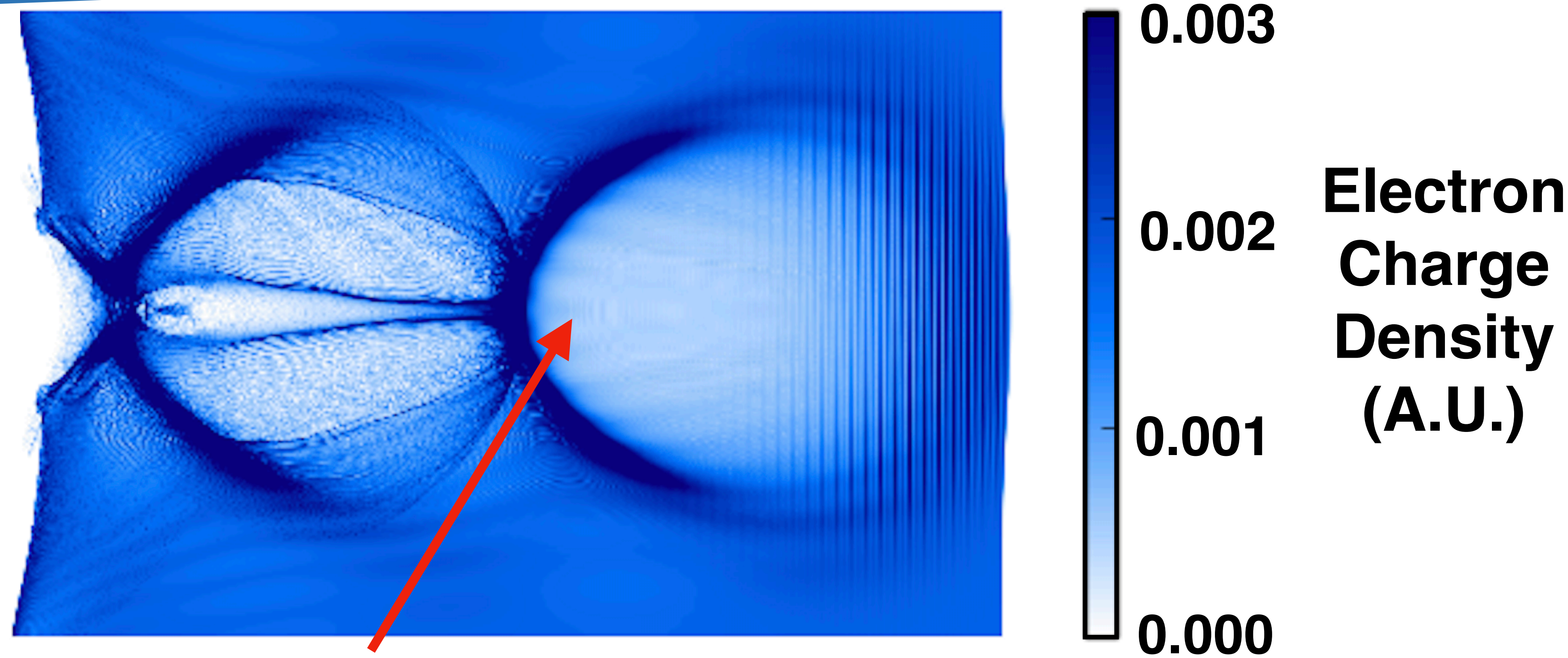


Electrons injected here are both accelerated and focused

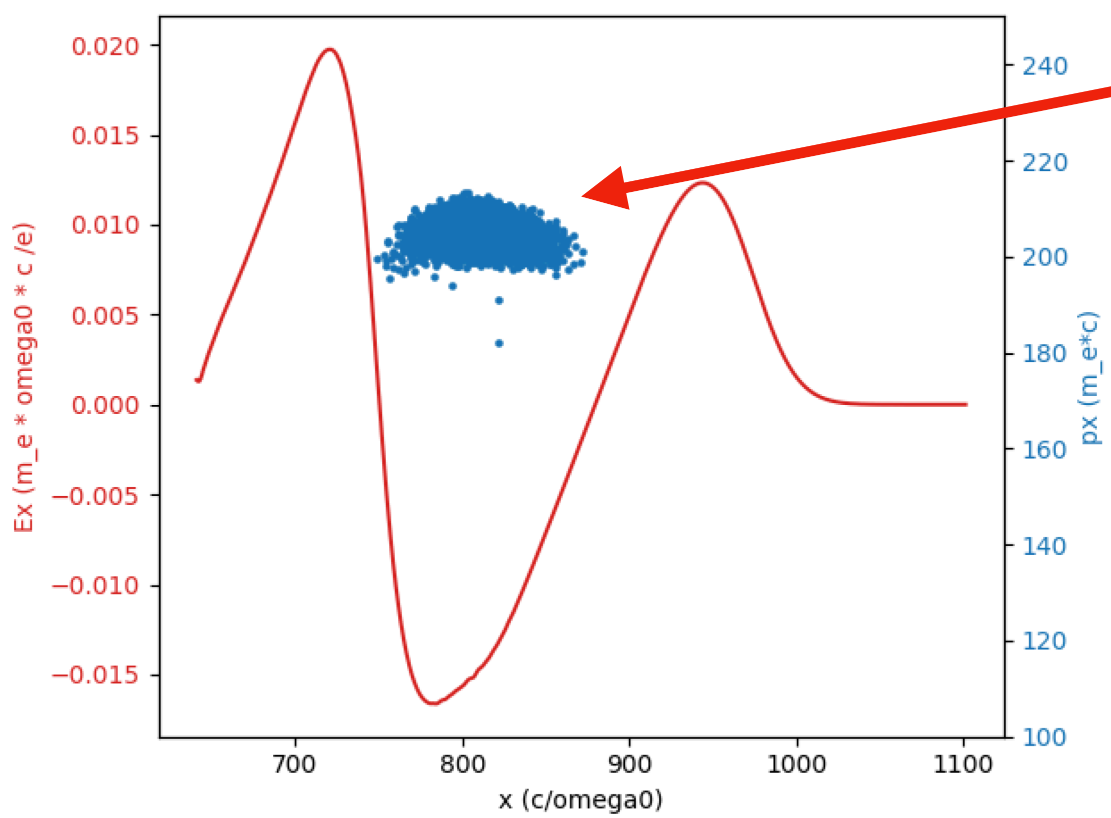


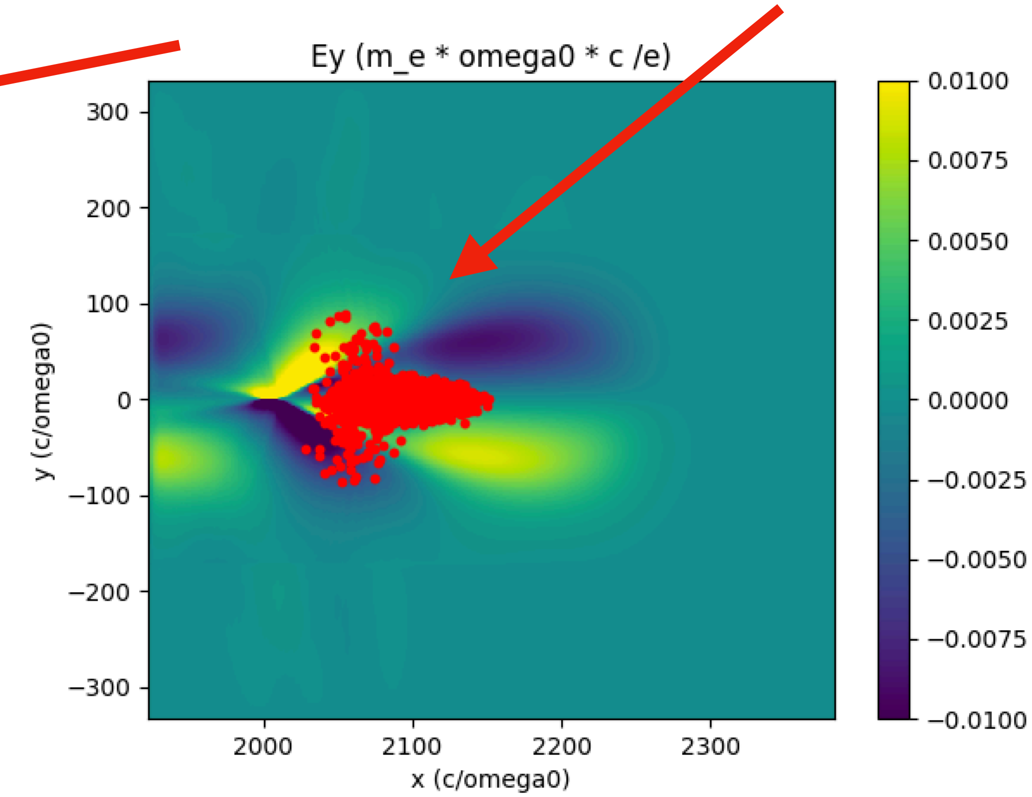
$$\vec{E} = E_{\text{longitudinal}} + E_{\text{perpendicular}}$$

Laser Wakefield Acceleration (LWFA): fields inside the bubble



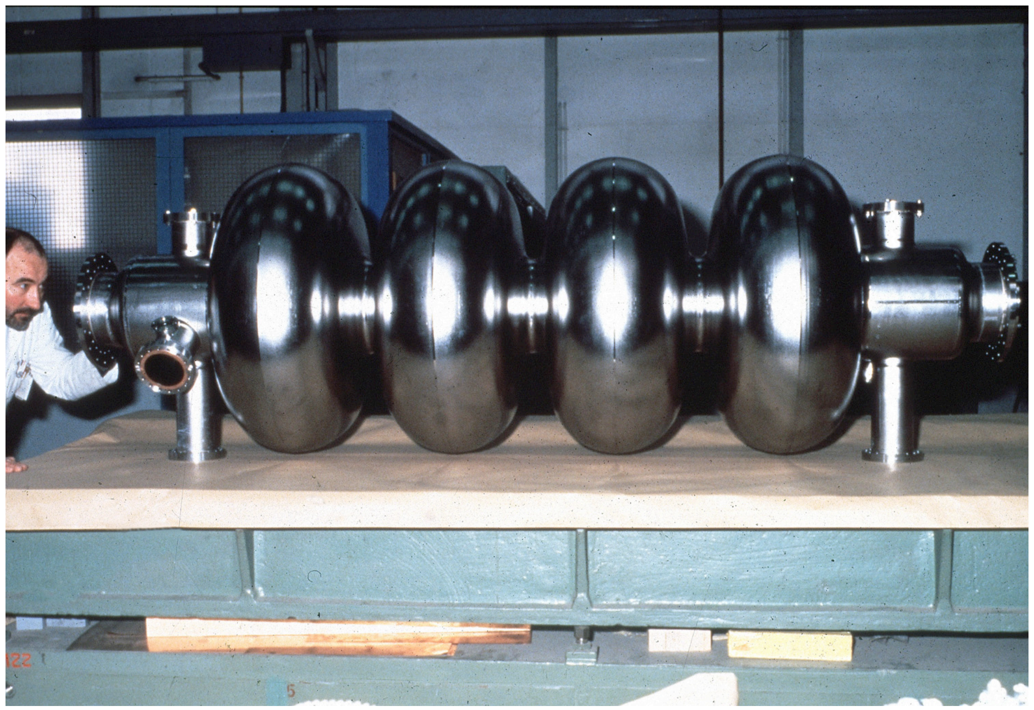
Relativistic electrons injected here are both accelerated and focused





Compactness of laser - plasma accelerators

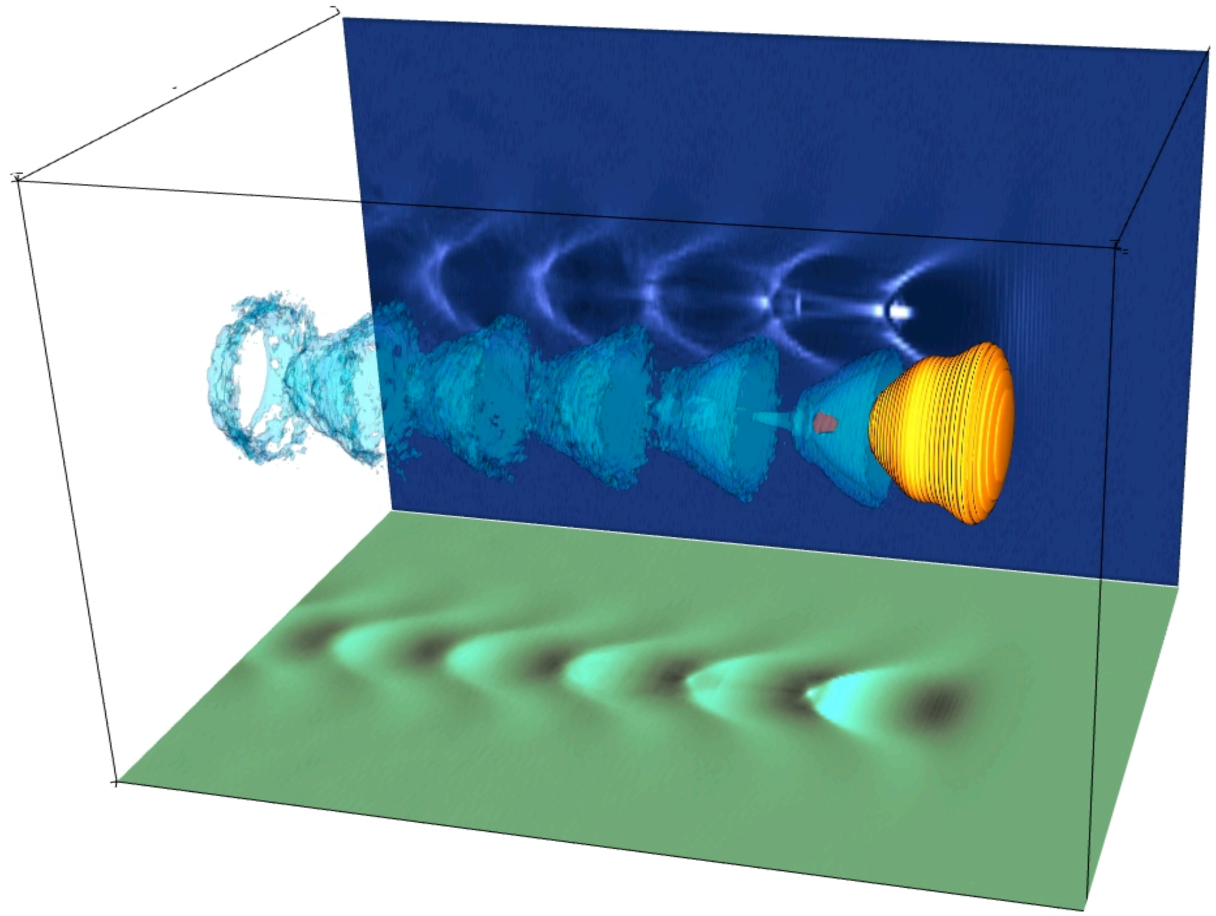
RF Cavity



1 m => 50 MeV Gain

Electric field < 100 MV/m

Plasma Cavity



1mm => 100 MeV

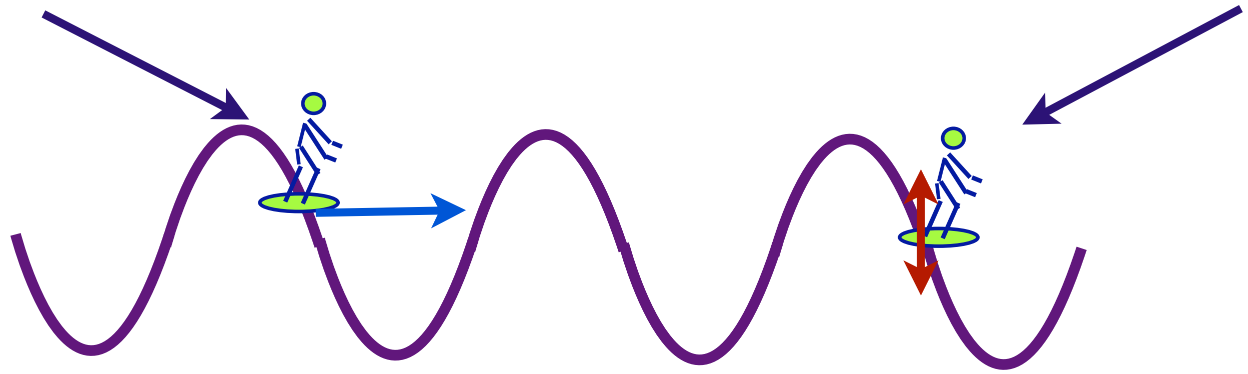
Electric field > 100 GV/m

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

Laser Wakefield Acceleration (LWFA) challenges: injection

Surfer with
- sufficient initial velocity,
- injected in the proper phase

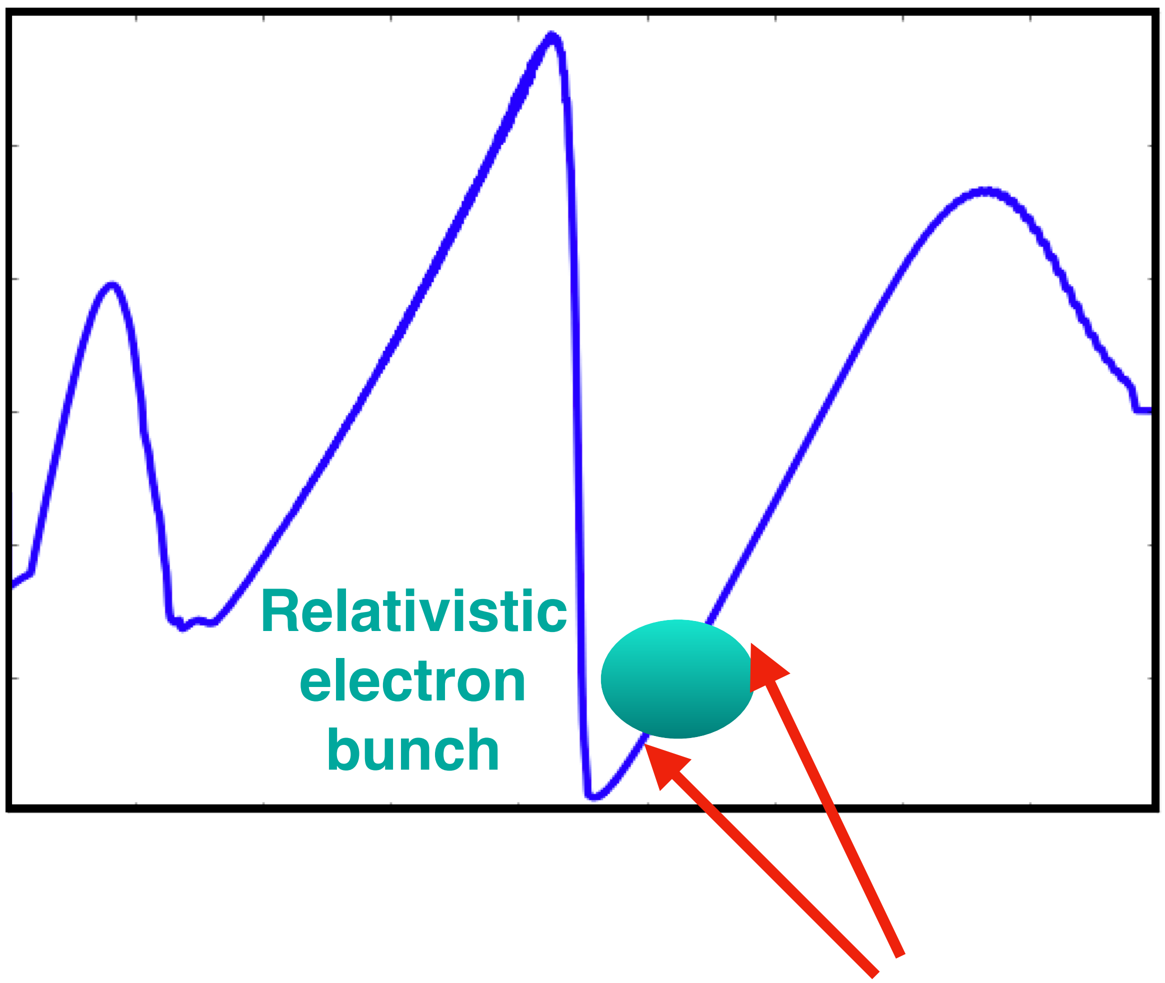
Surfer with
Initial zero velocity



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



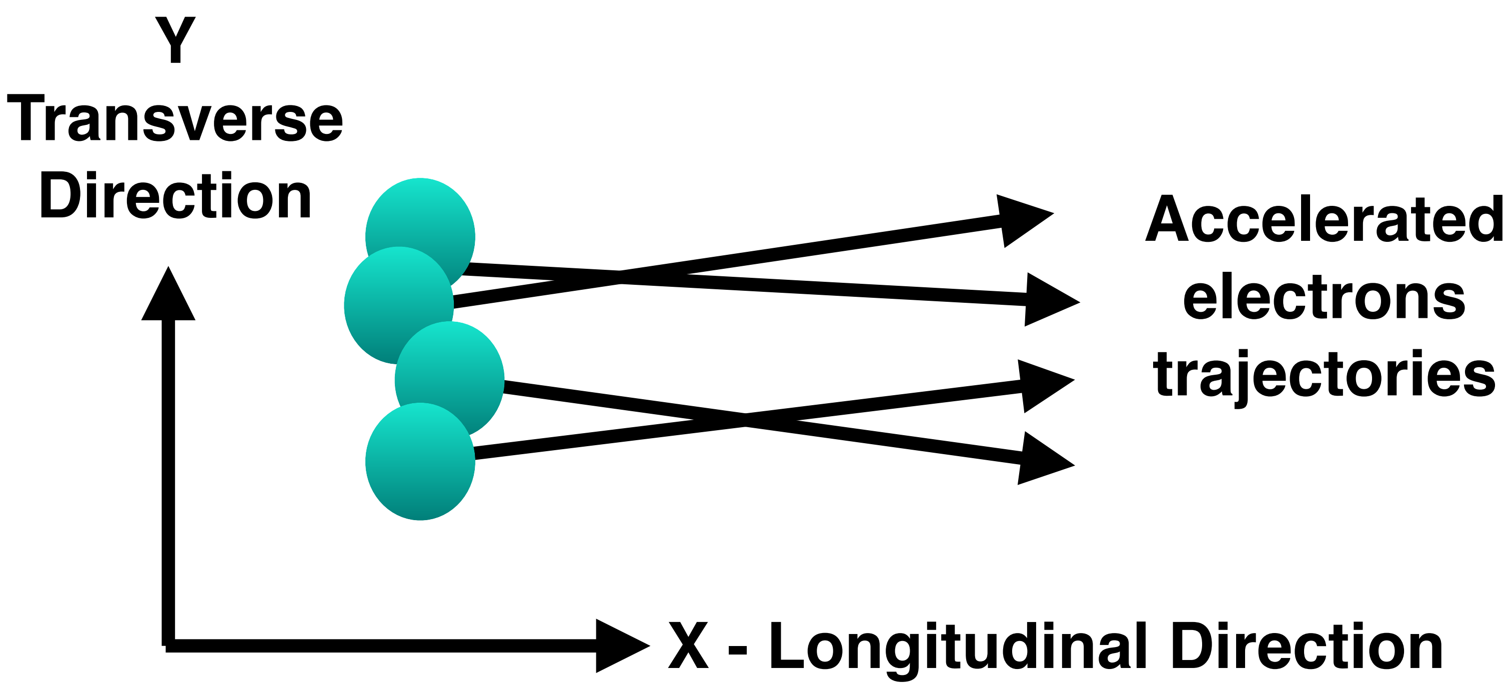
Longitudinal Electric Field (A.U.) on propagation axis

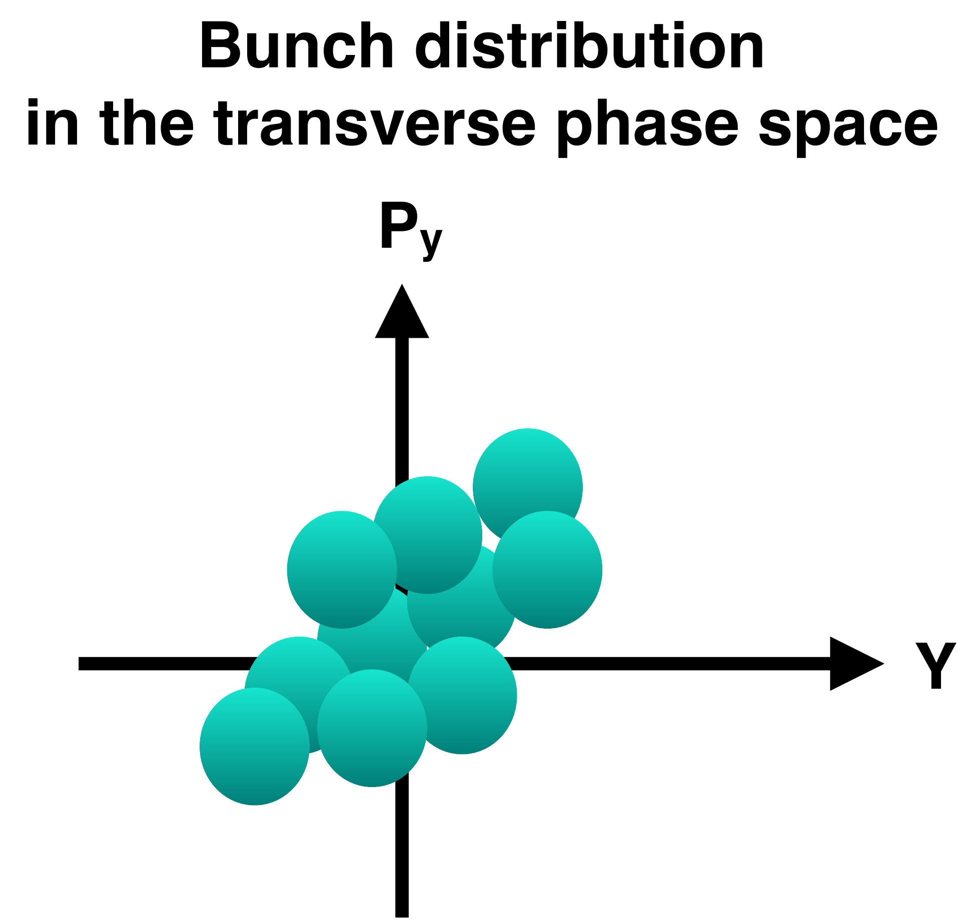
Electrons injected in different phases of the wave will experience a different accelerating field

↓

Energy spread increase

Laser Wakefield Acceleration (LWFA) challenges: emittance





Most applications of accelerated beams require:
- Small transverse size (i.e. small σ_y)
- Small divergence (i.e. small σ_{p_y}/Energy)

Minimize Transverse Emittance

Transverse Normalized Emittance

$$\varepsilon_{ny} = \frac{1}{m_e c^2} \sqrt{\sigma^2_y \, \sigma^2_{p_y} - \sigma^2_{yp_y}}$$

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

$f(x, y, z, p_x, p_y, p_z, t)$ → **6 dimensions + time!**

Coupled to the electromagnetic fields → **Non-linearity**

$$\frac{\partial f}{\partial t} + \boldsymbol{\beta}\, c \cdot \nabla_{\boldsymbol{x}} f - e\,(\boldsymbol{E} + \boldsymbol{\beta}\, c \times \boldsymbol{B}) \cdot \nabla_{\boldsymbol{p}} f = 0$$

← **Collisions neglected**

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

Current density J of the plasma

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

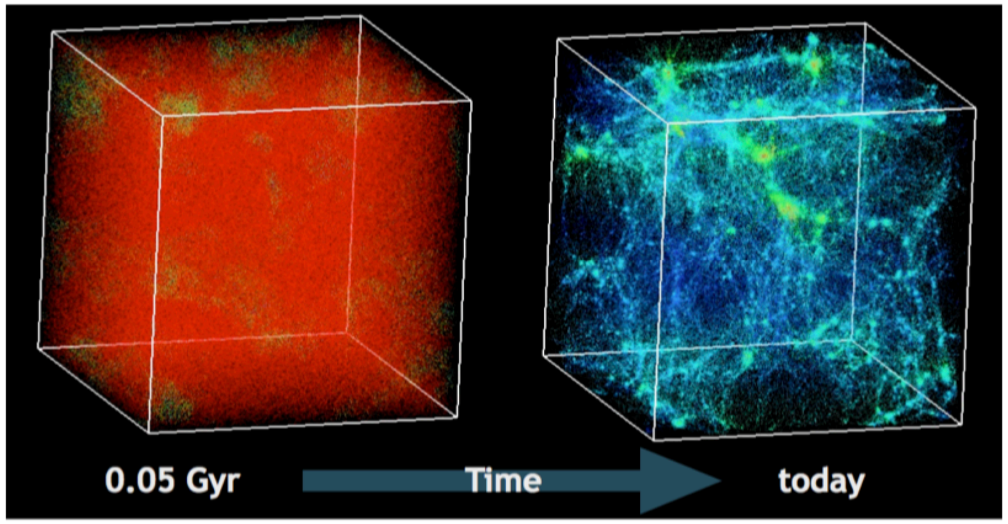
Explorable Physics:

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

Approximate solution: Particle in Cell method

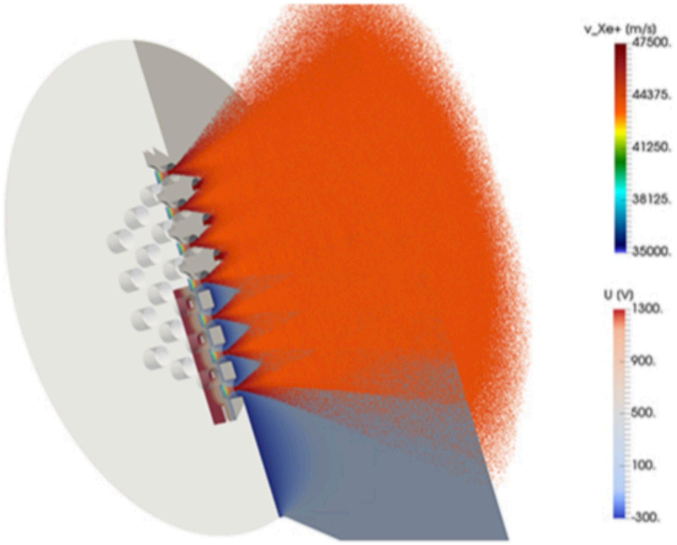
Particle in Cell (PIC) method : essential plasma investigation technique

Cosmology



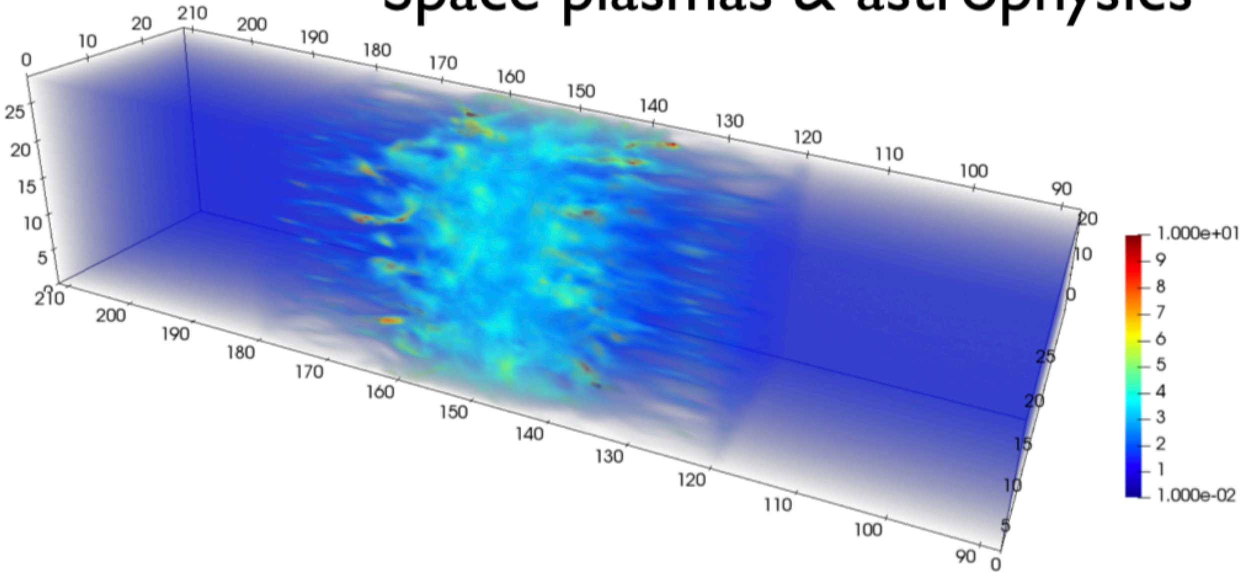
source: K. Heitmann, Argonne National Lab

Space propulsion (Plasma thruster)



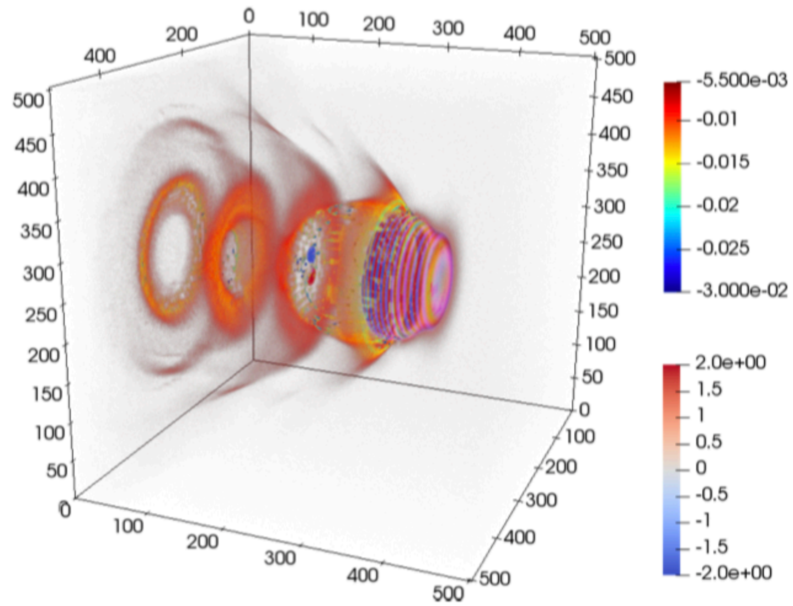
source: Gauss Center for Supercomputing

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

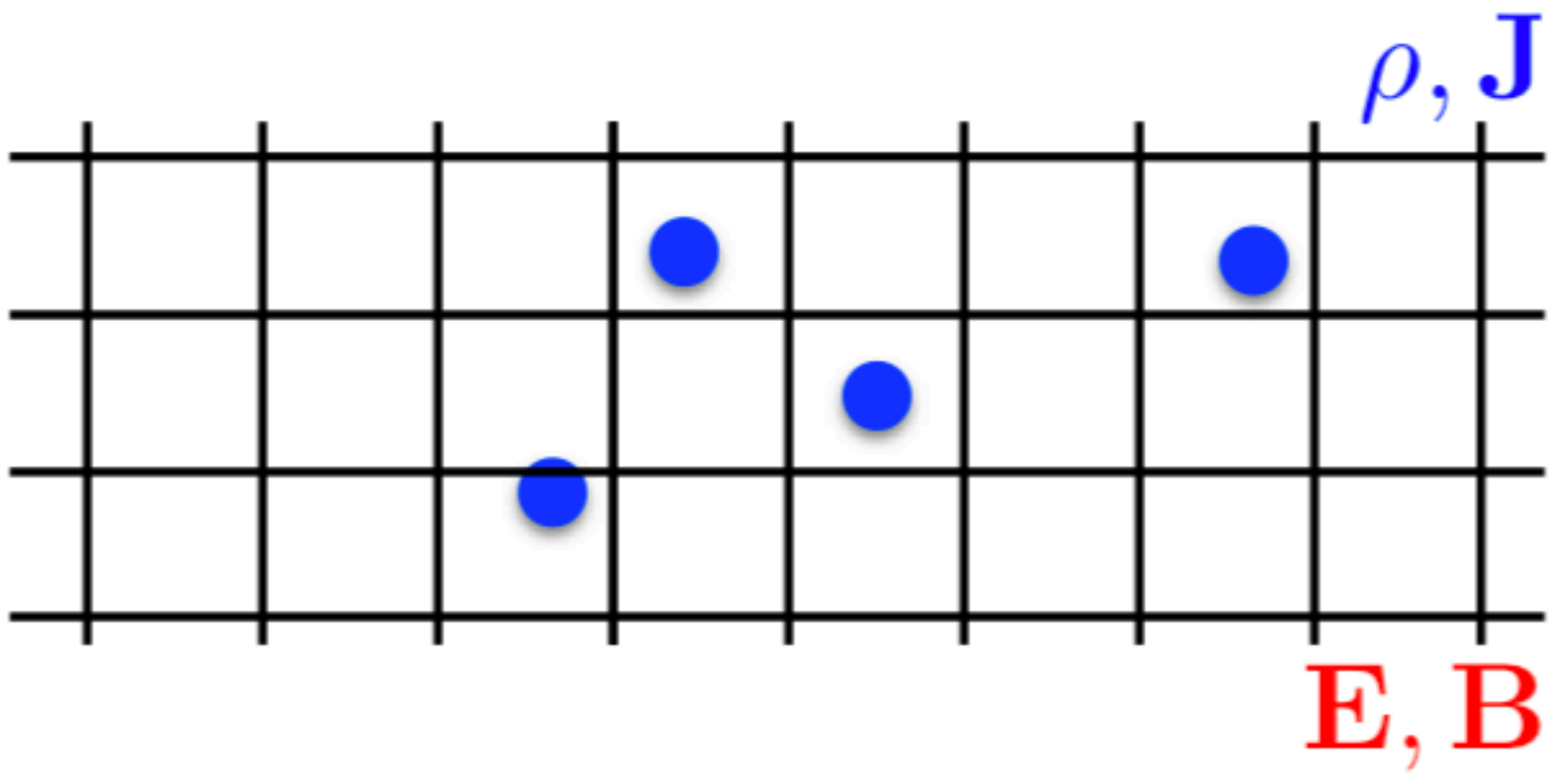
Particle in Cell concept

Sample Plasma with Macro-Particles
(1 Macro-particle = position, momentum, charge, …)

+

Discretize space with computational grid

Define E, B, ρ, J on the grid cells

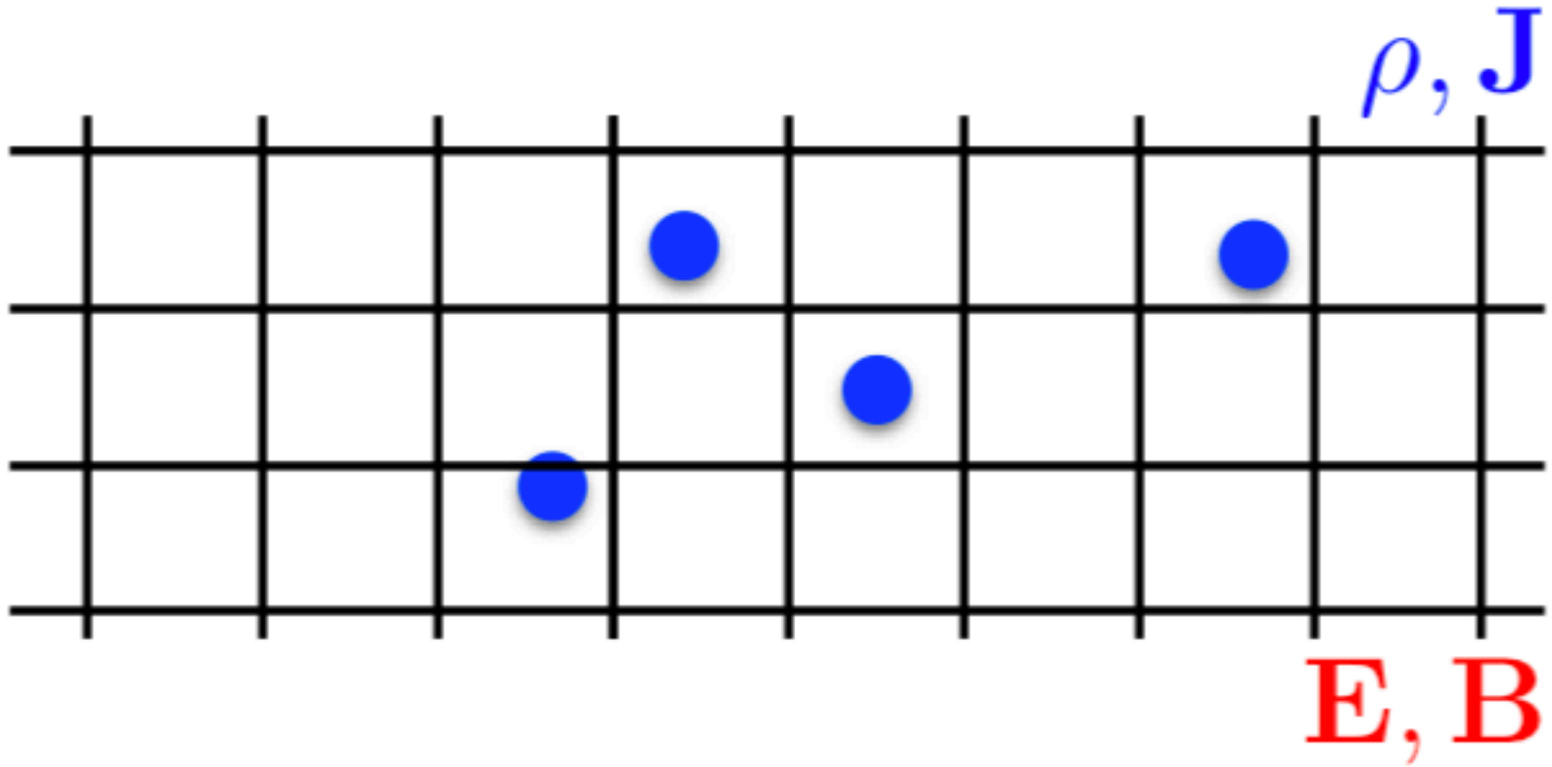


Particle in Cell concept

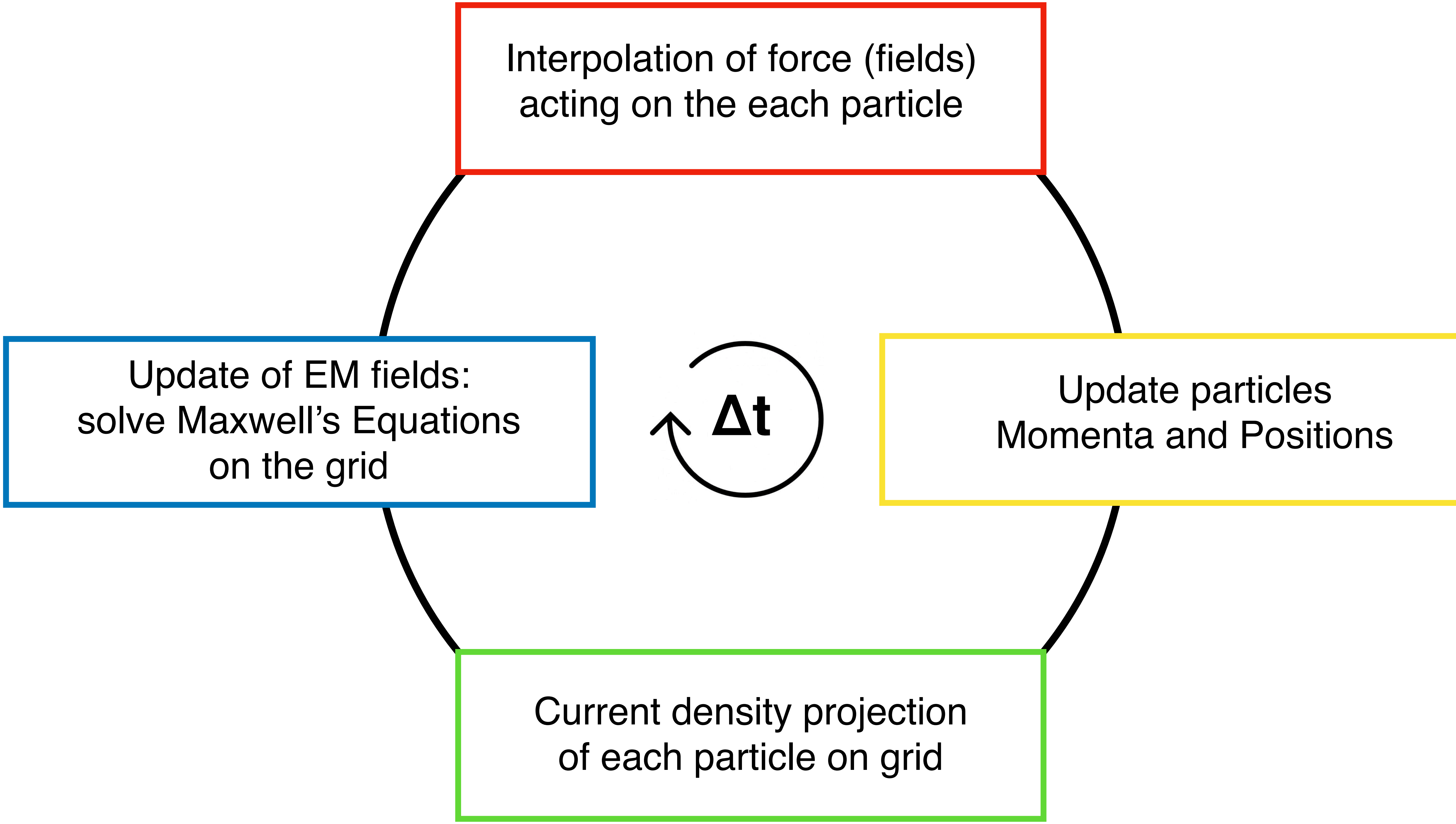
Macro-Particles: evolve following their equations of motion

+

Electromagnetic Fields: evolve following Maxwell's Equations



Particle in Cell modelling is self-consistent



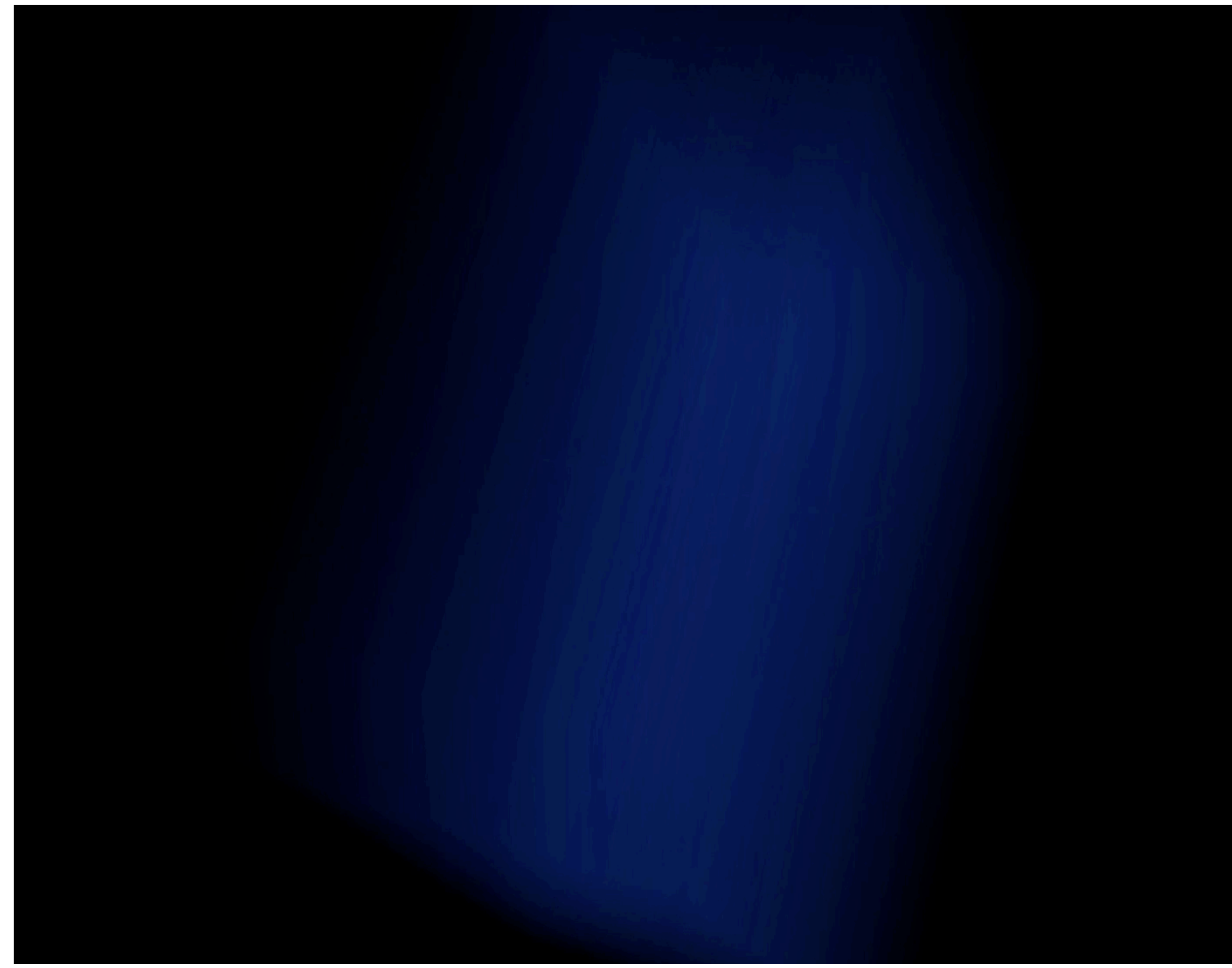
Outline

- 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

Smilei)

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

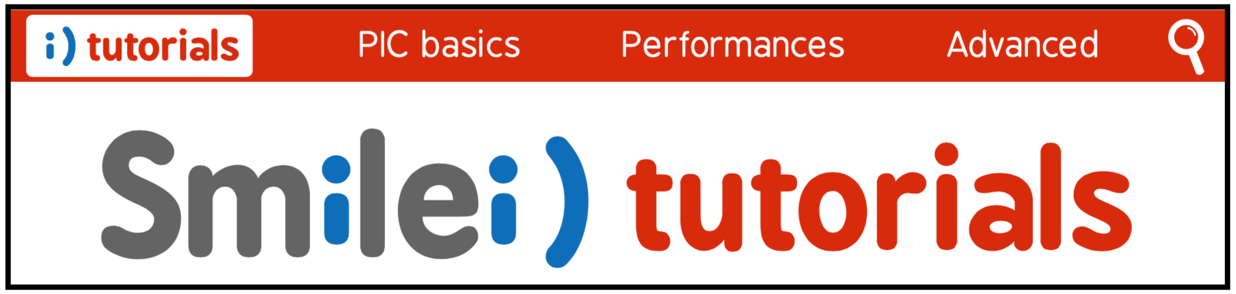
- developers: experts of physics and HPC
- continuously benchmarked
- GitHub bug reporting
- OpenPMD standard

MAISON DE LA SIMULATION

Additional material

Extensive Documentation
(Installation, Use, Postprocessing, …)

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



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

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

https://indico.math.cnrs.fr/event/6911/

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	Fixed
Charge	e	Fixed
Mass	m_e	Fixed
Momentum	$m_e\, c$	Fixed
Energy	$m_e\, c^2$	Fixed
Time	$1/\omega_r$	Need to choose ω_r
Length	c/ω_r	Need to choose ω_r
Number Density	$\varepsilon_0 m_e\, \omega^2_r/e^2$	Need to choose ω_r
Electric Field	$m_e\, \omega_r\, c\, /\, e$	Need to choose ω_r
Magnetic Field	$m_e\, \omega_r\, /\, e$	Need to choose ω_r

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

Example of input file

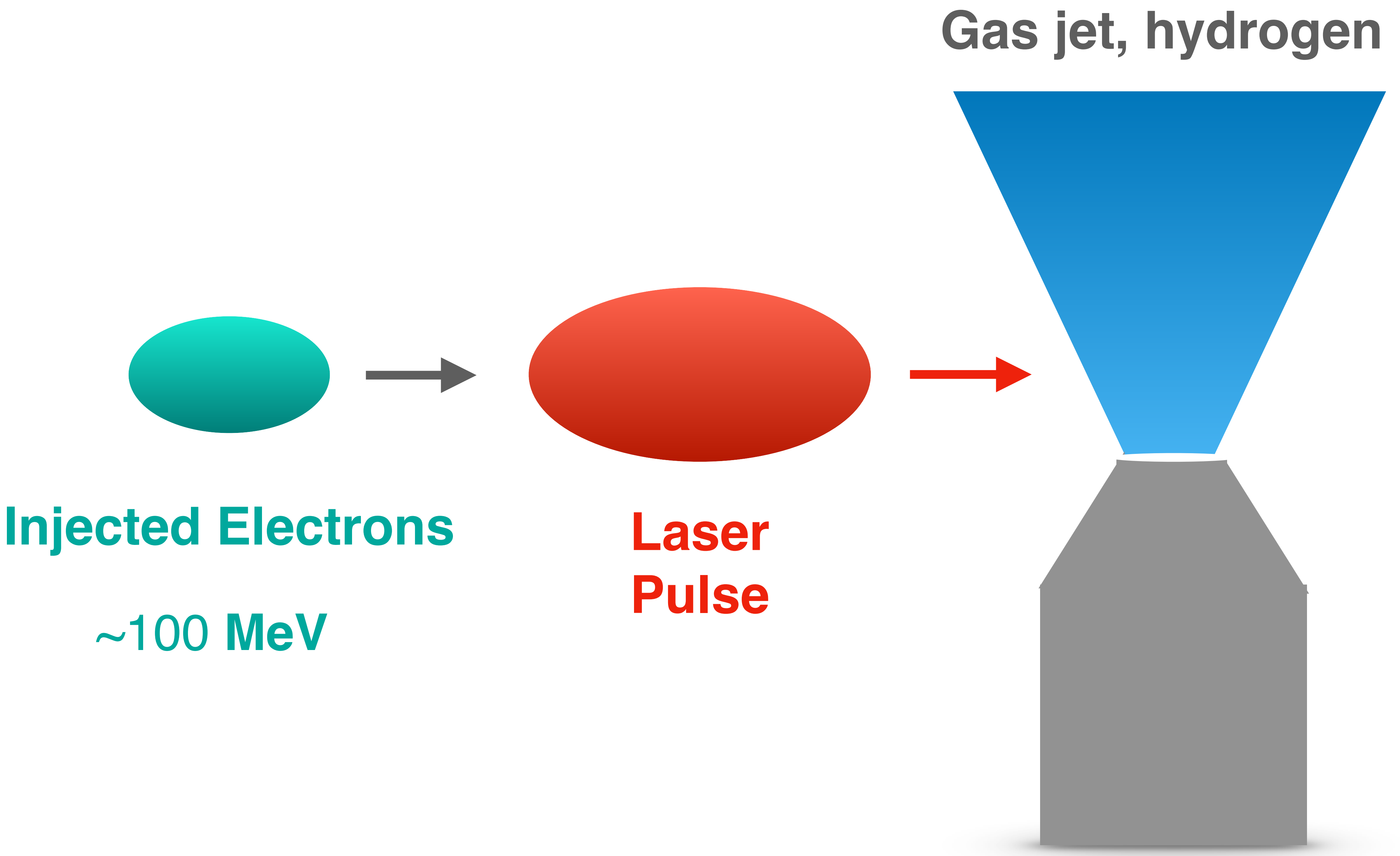
```
Main(
    geometry           = "1Dcartesian",
    timestep           = 0.009,
    cell_length        = [0.01],
    ...
)

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



Assumption: plasma already created in the gas jet
(the laser ionises the gas in its path)

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

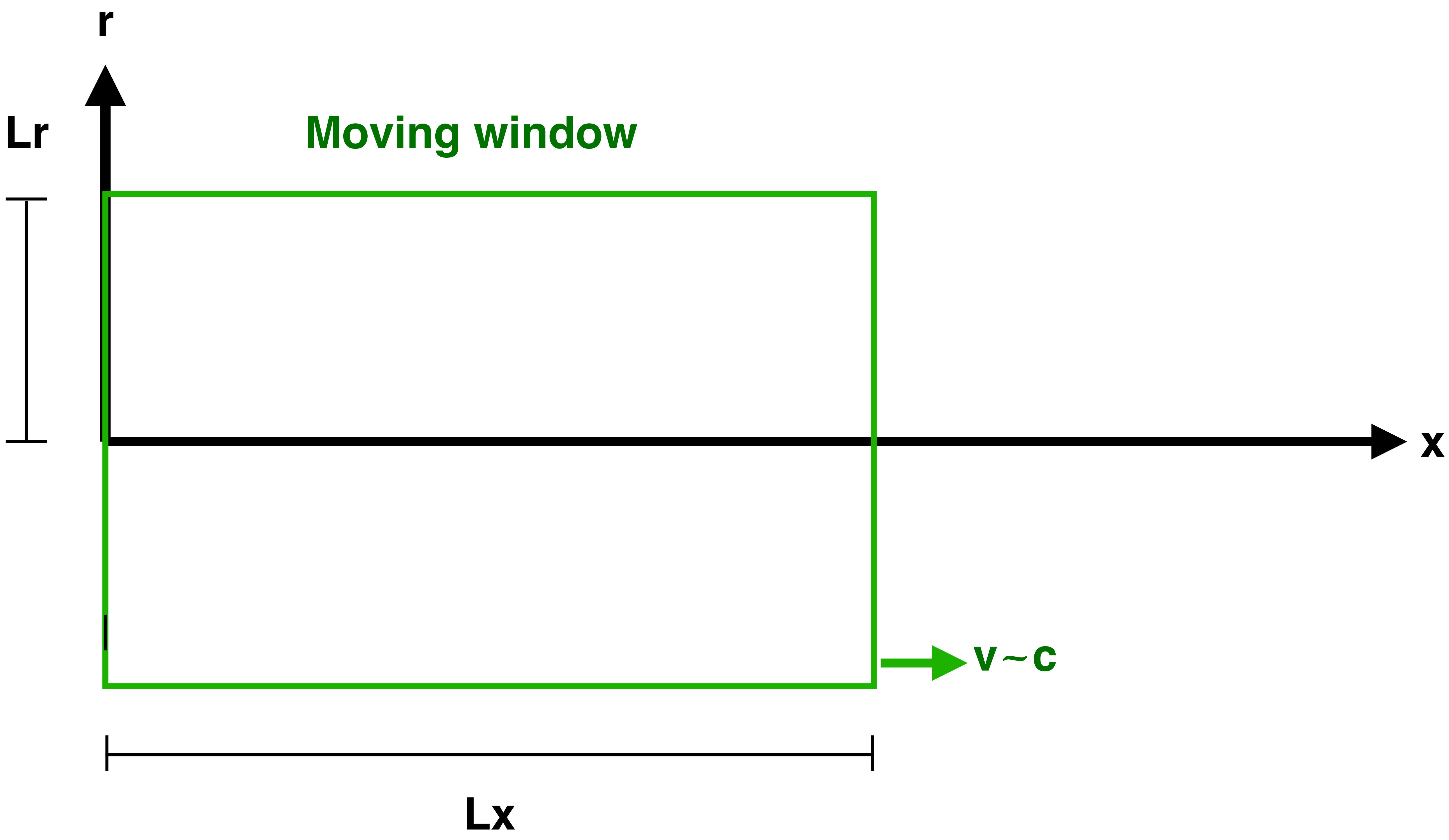
Structure of the input file for the practical

- **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



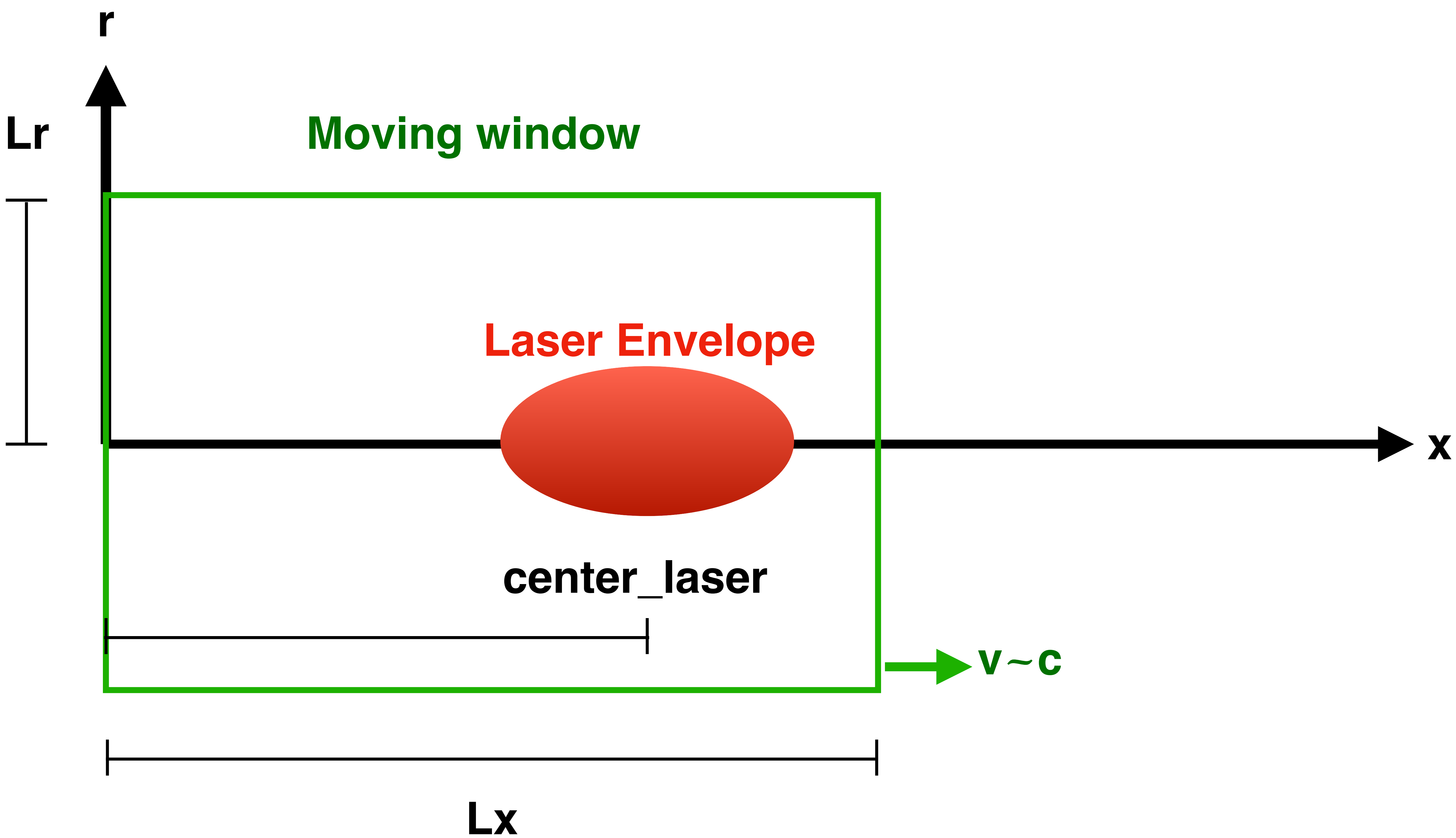
Structure of the input file for the practical

- 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



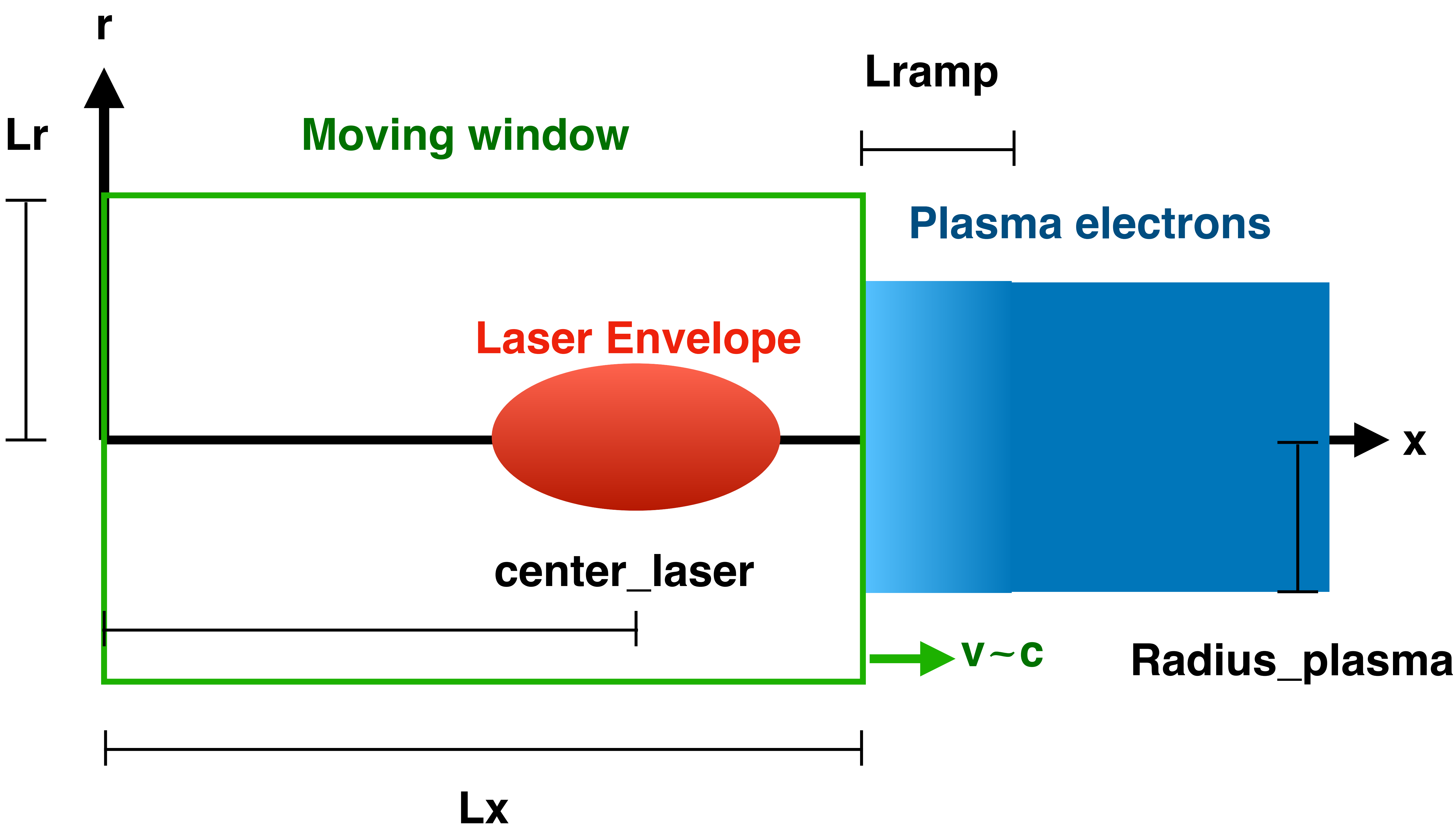
Structure of the input file for the practical

- 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



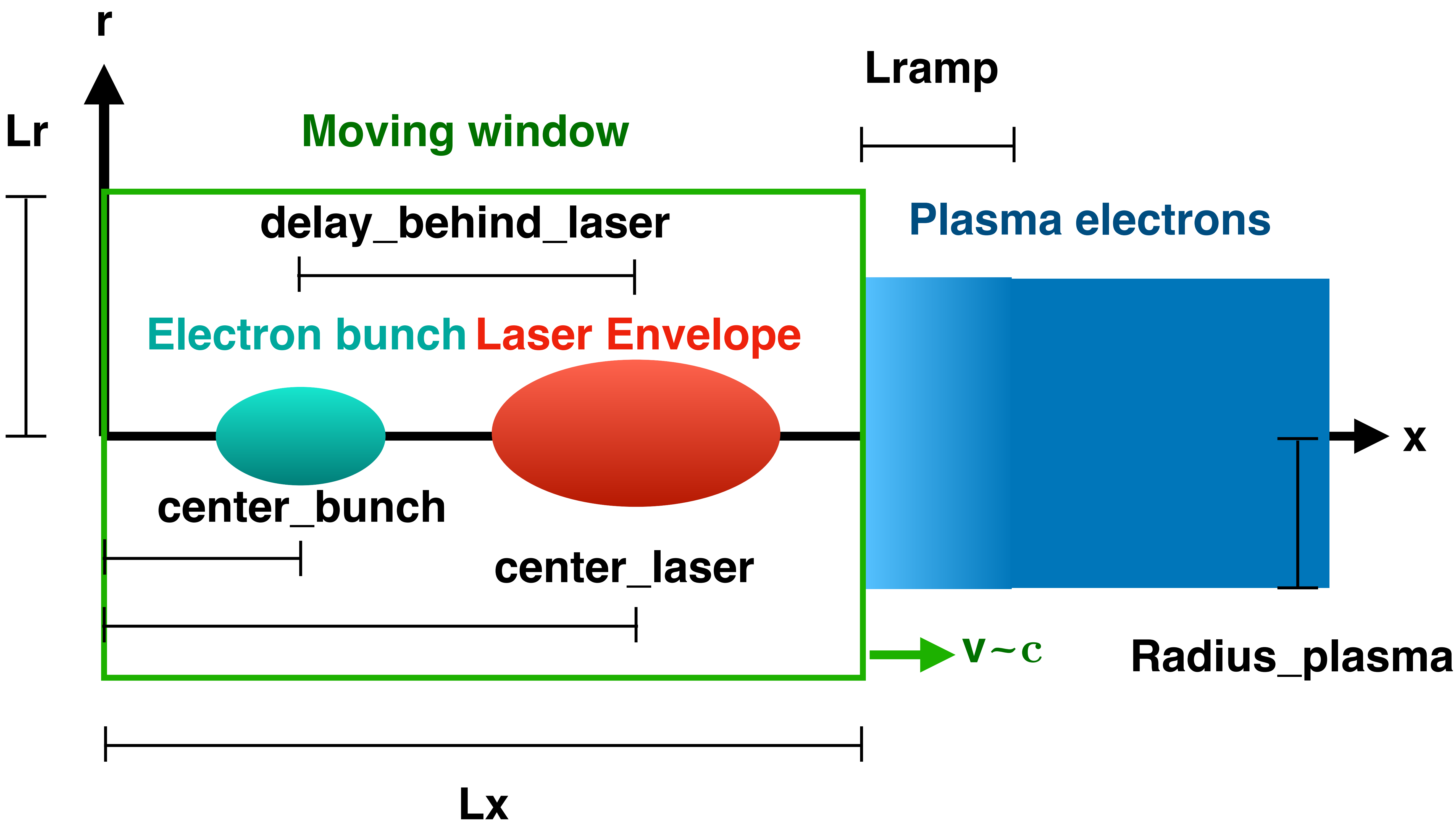
Structure of the input file for the practical

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

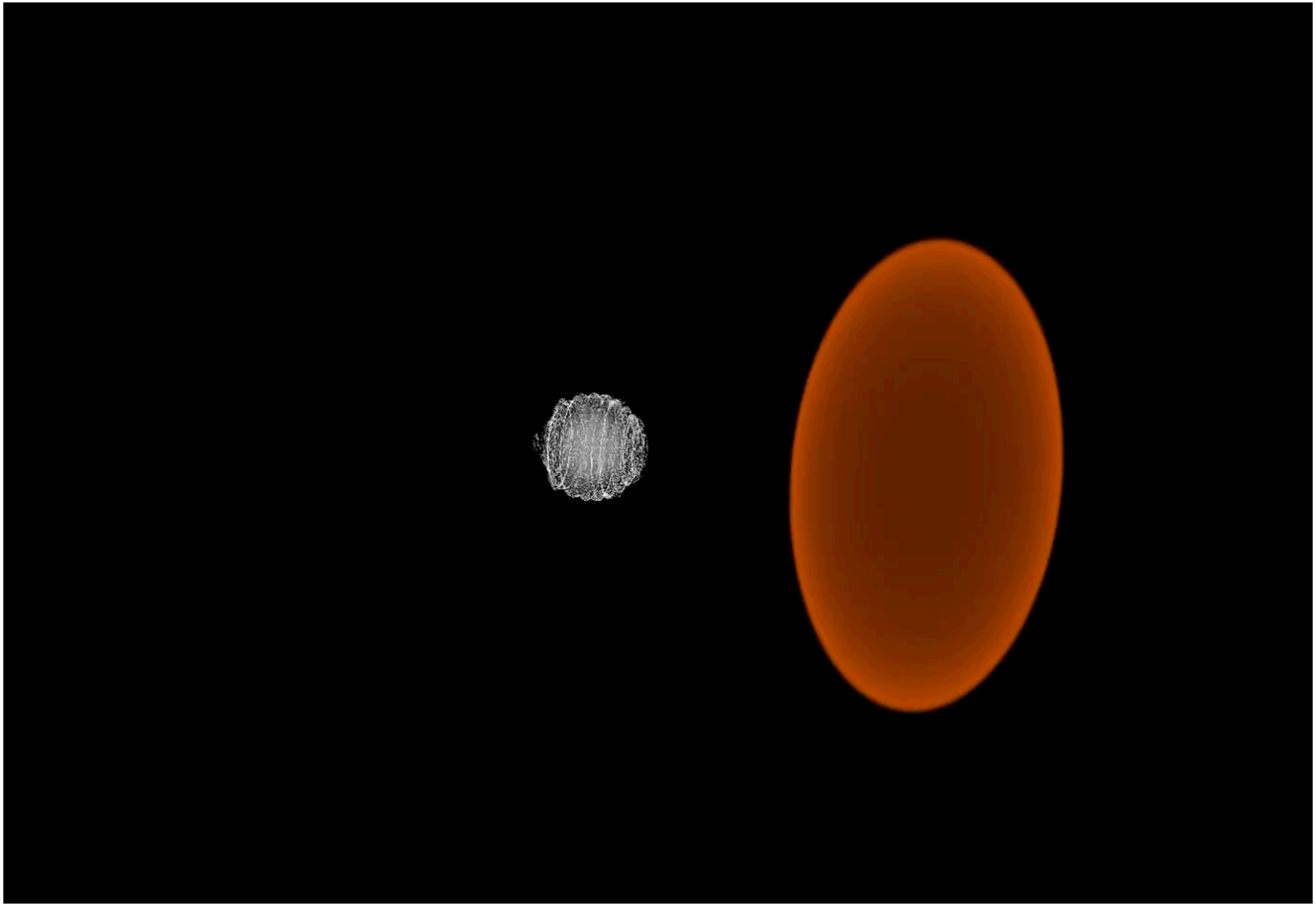
Blue: Active

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 happi
$ ipython
In [1]: import happi
```

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 happi commands from the handouts
- Adapt them for your purposes
- Use the post processing scripts available (see handouts)

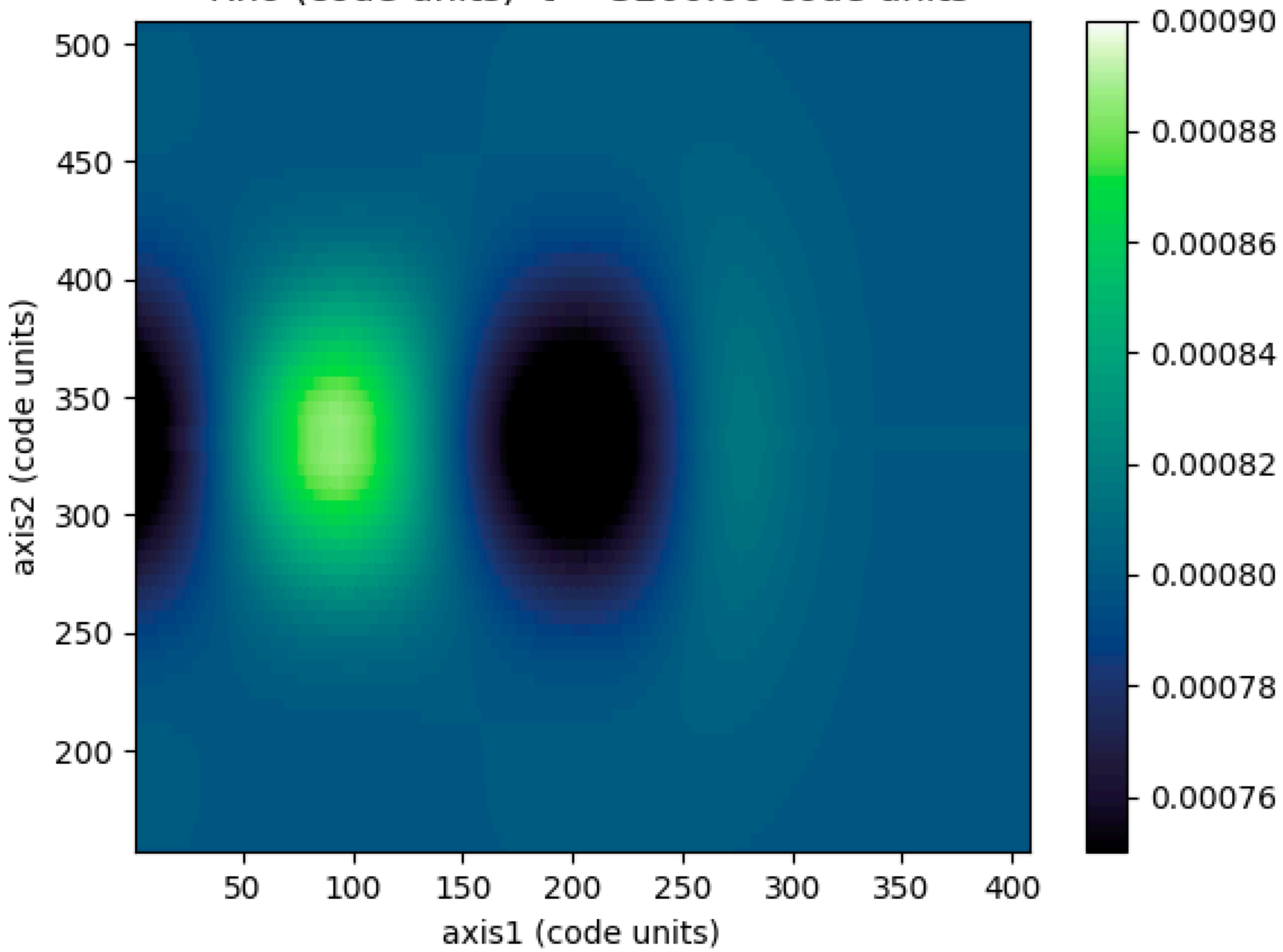
Questions?

Additional Slides

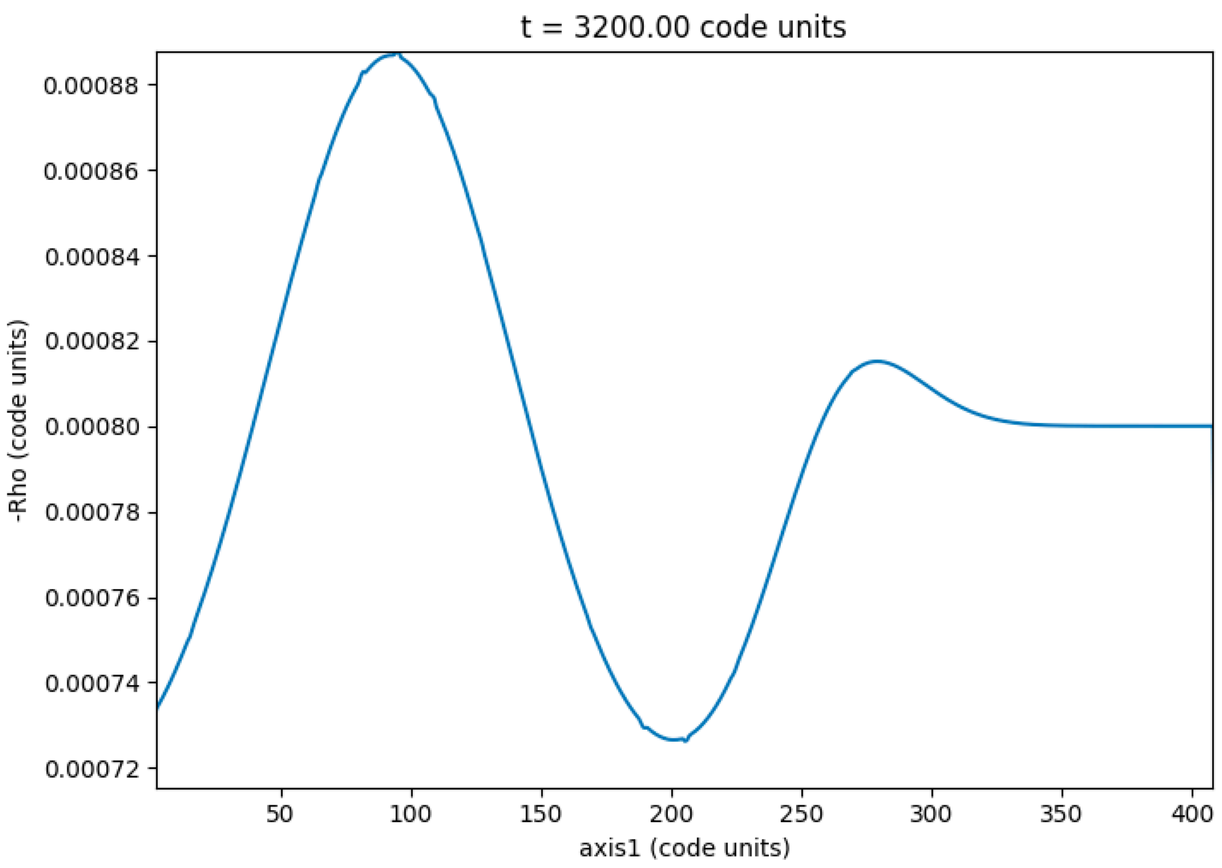
Exercise 10: Nonlinear plasma waves, density

$a_0 \ll 1$:
Linear regime
(Sinusoidal plasma waves)

Laser Pulse propagation along the x axis



2D charge density



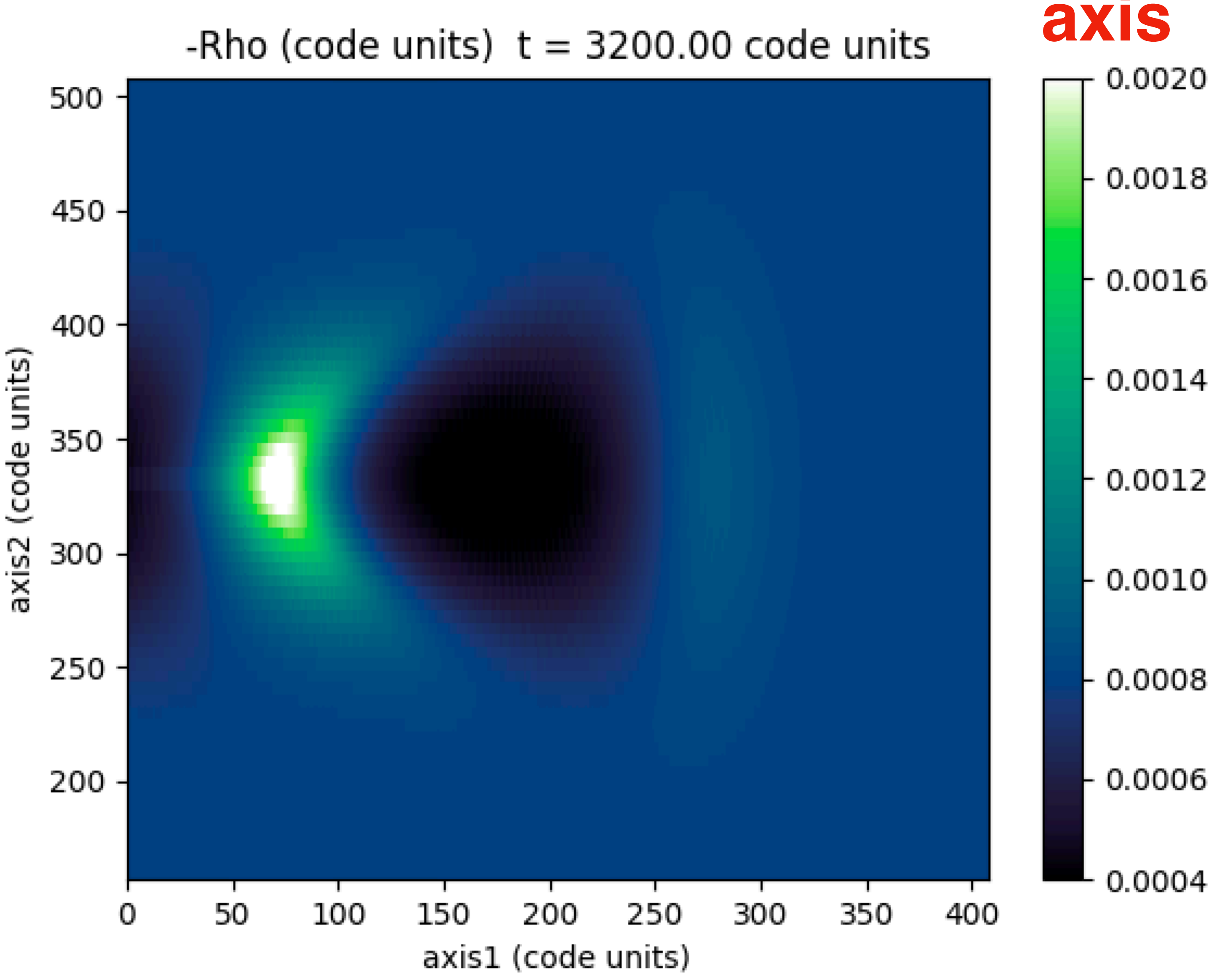
1D charge density on propagation axis

Exercise 10: Plasma waves regimes, density

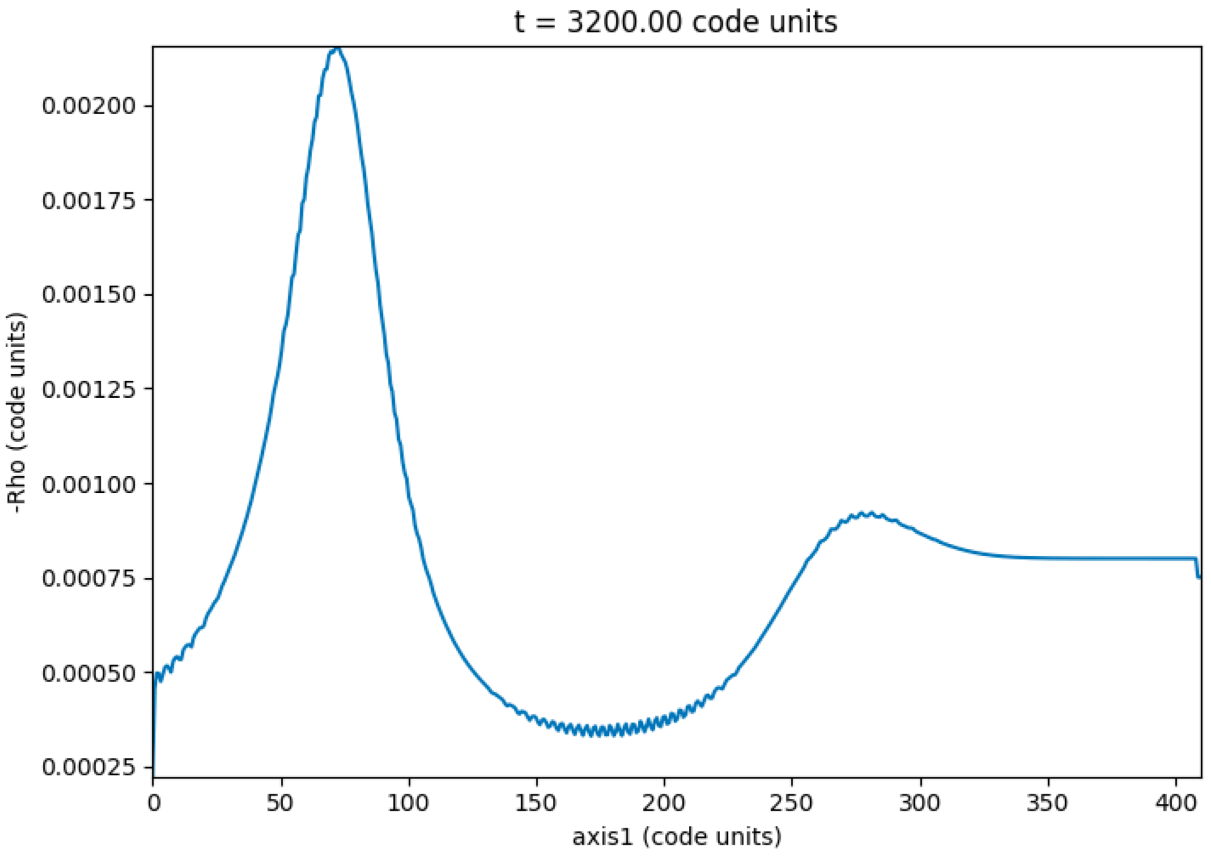
$a_0 \lesssim 1$:

Weakly nonlinear regime

Laser Pulse propagation along the x axis



2D charge density

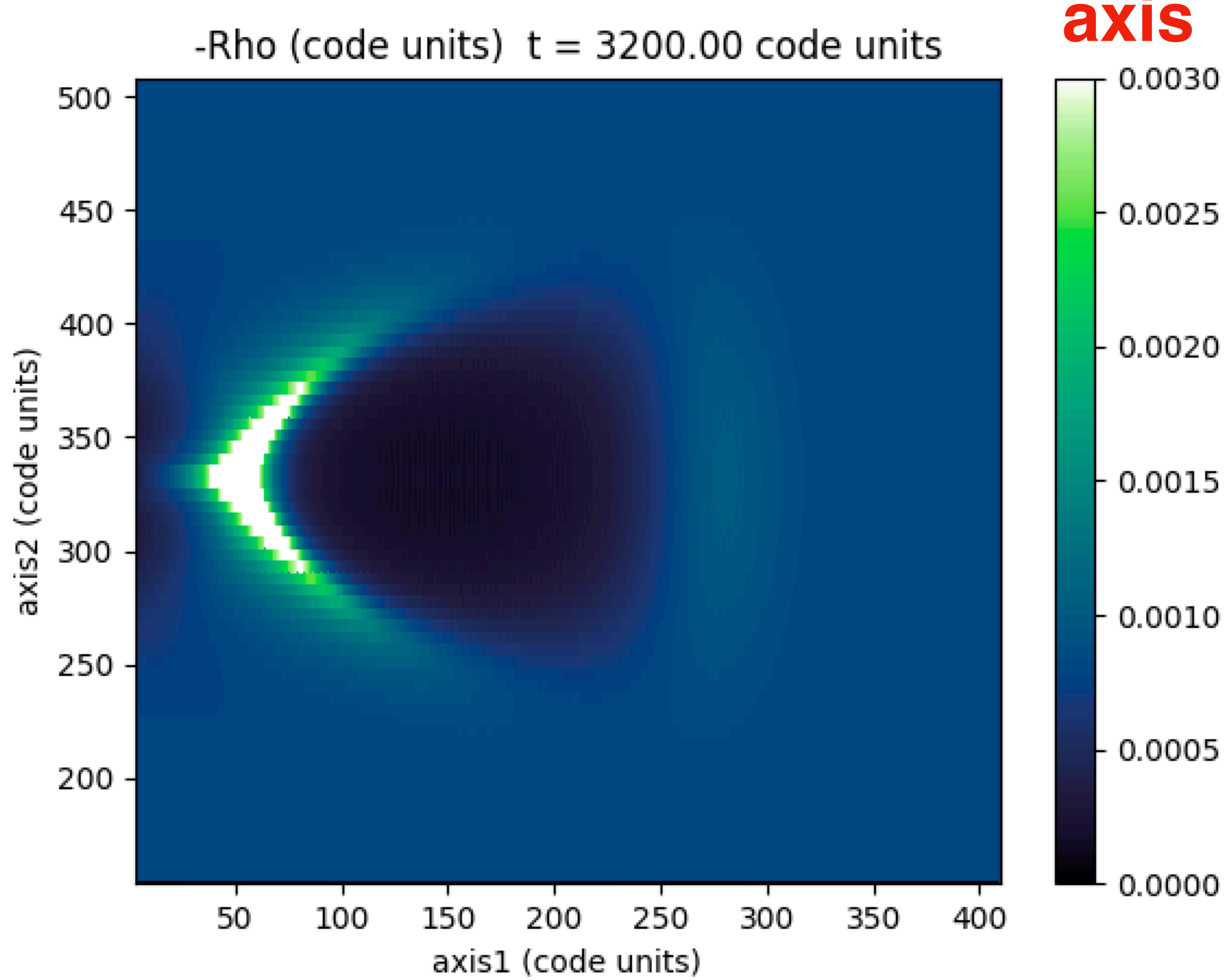


1D charge density on propagation axis

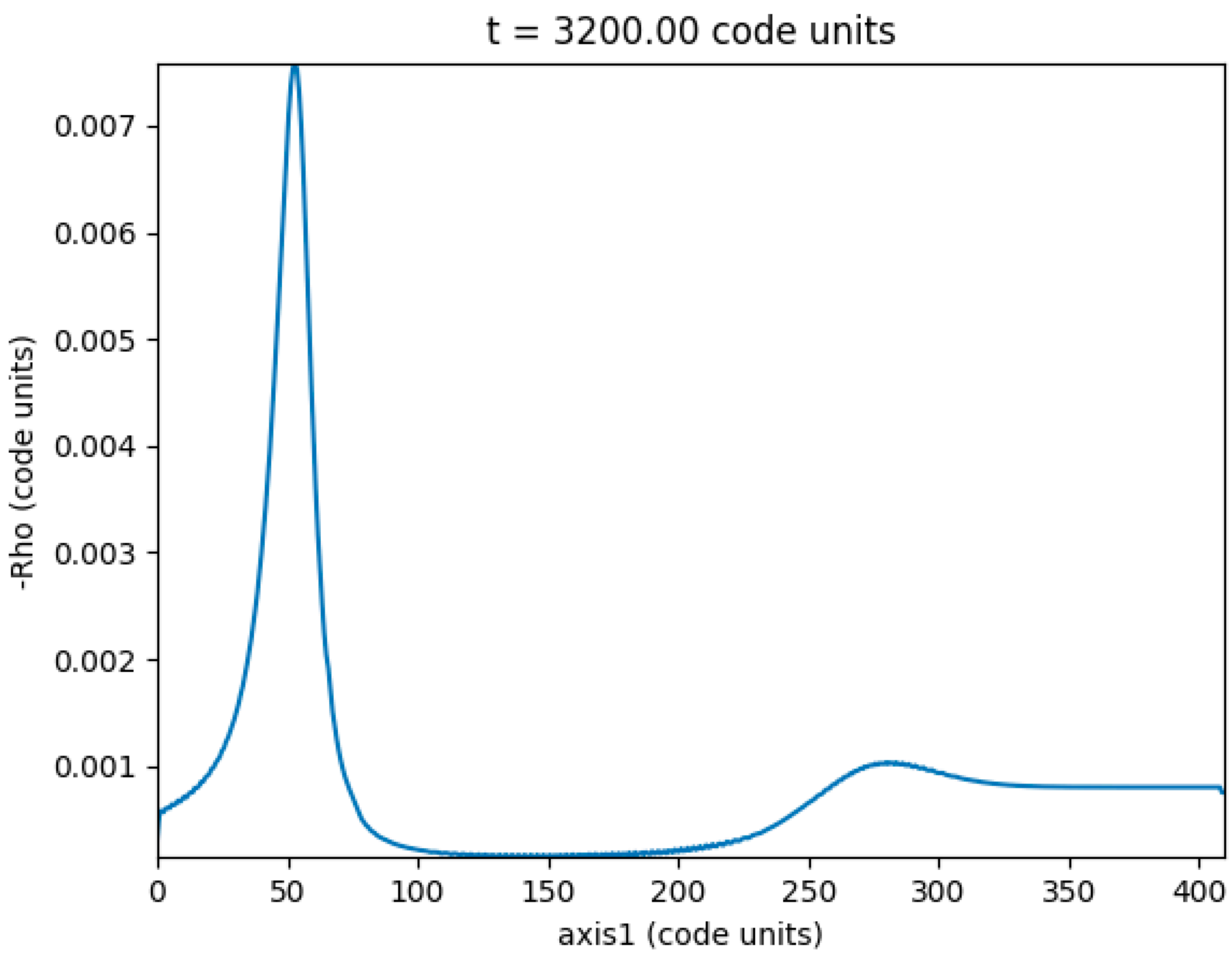
Exercise 10: Plasma waves regimes, density

$a_0 > 1$:
Nonlinear regime
(Bubble-like waves)

Laser Pulse propagation along the x axis



2D charge density



1D charge density on propagation axis

Exercise 10: Plasma waves regimes, Ex on axis

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