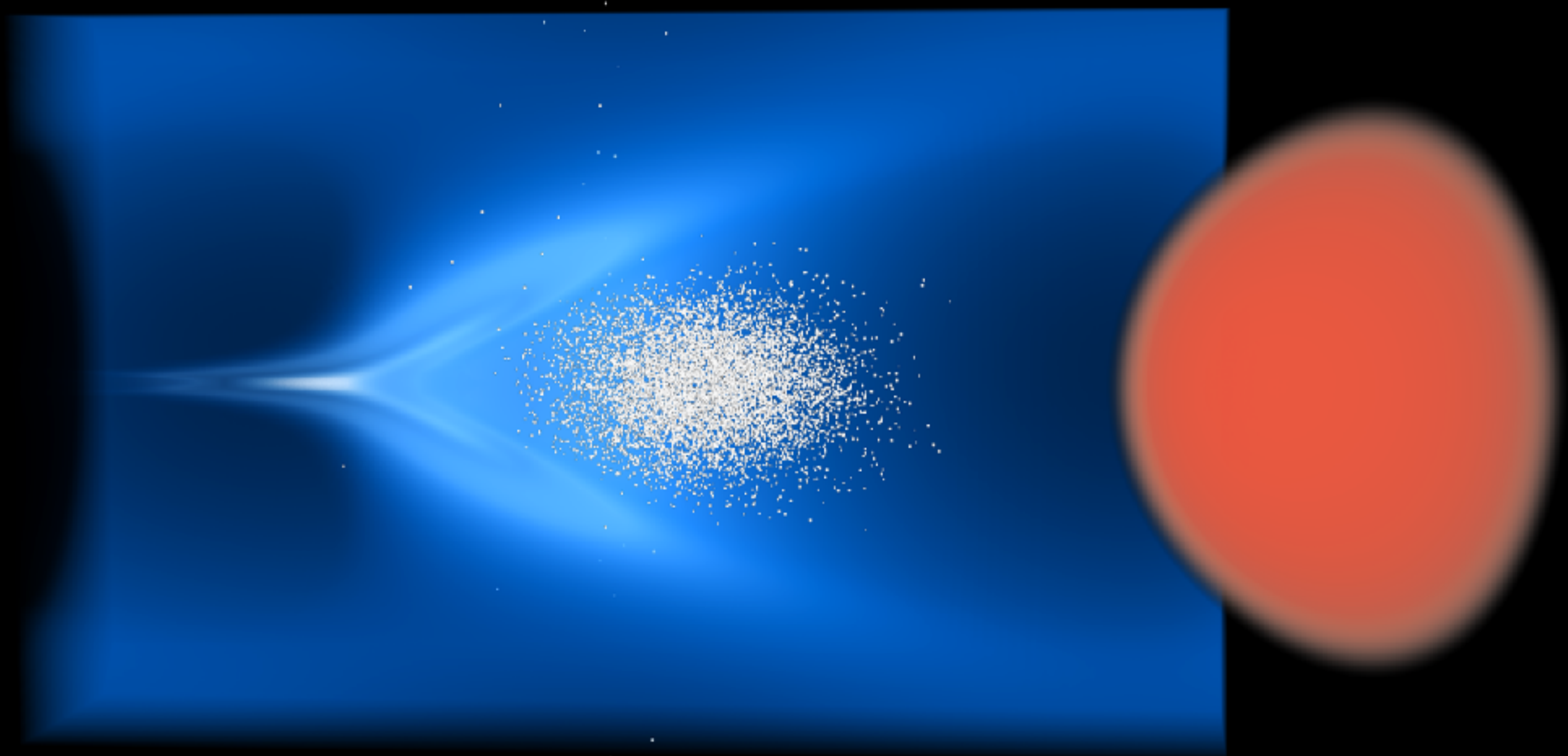
Simulation of Laser Wakefield Acceleration



Francesco Massimo, LPGP

28 nov, 1 dec, 5 dec 2025

Outline

- Basics of laser wakefield acceleration
- Numerical simulation of plasma acceleration: PIC codes
- Introduction to the case study and the practical

Outline

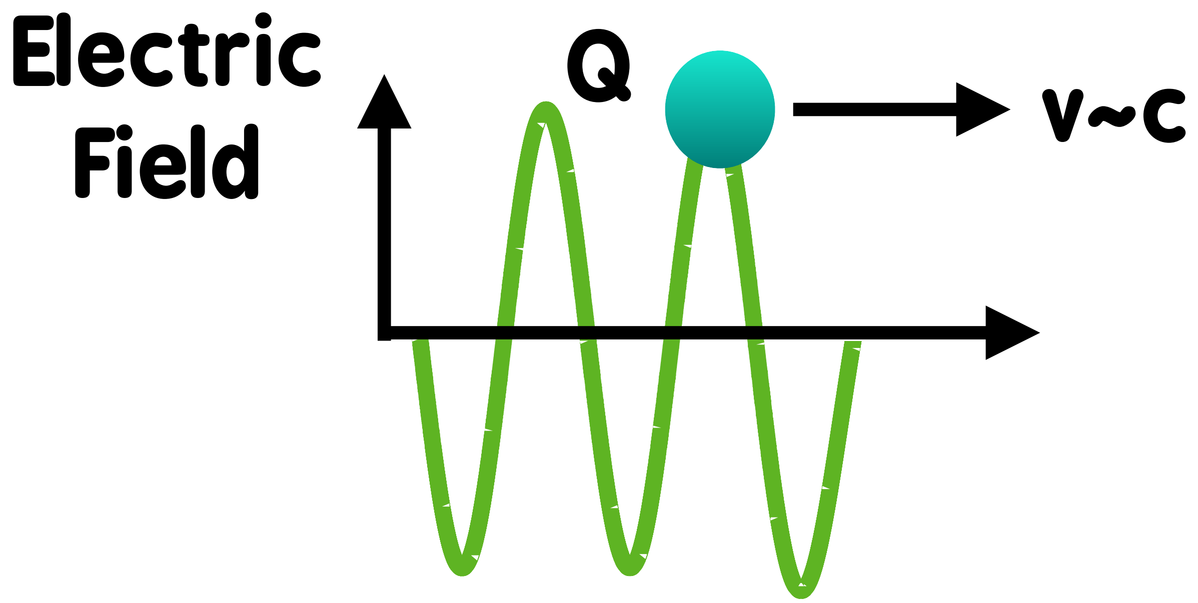
- Basics of laser wakefield acceleration
- Numerical simulation of plasma acceleration: PIC codes
- Introduction to the case study and the practical

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,

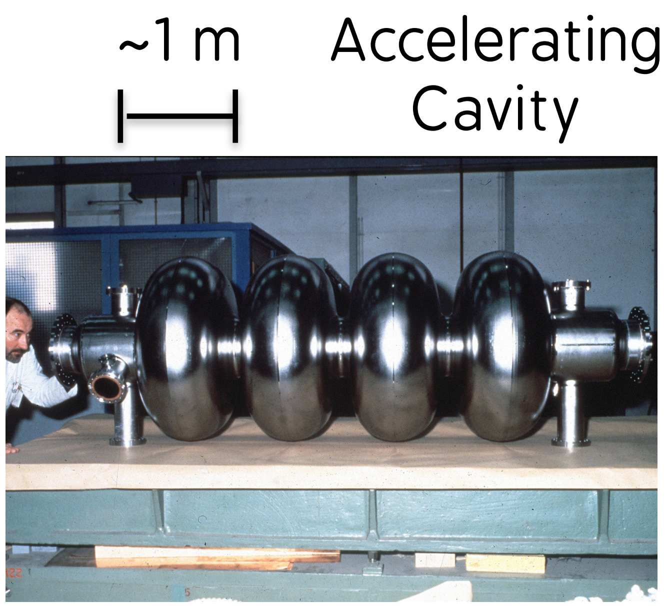
if **E** is limited by technology, **L** increases

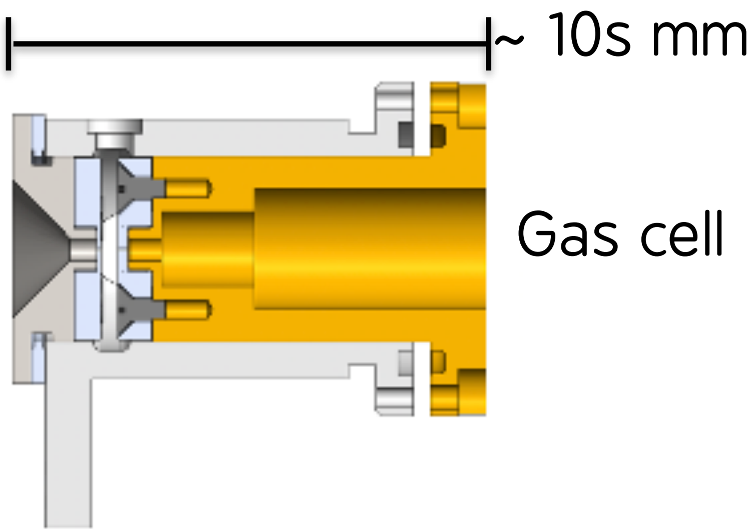
→ need more metallic cavities

→ the accelerator size and cost increase

Accelerating E-field of Laser Wakefield Acceleration

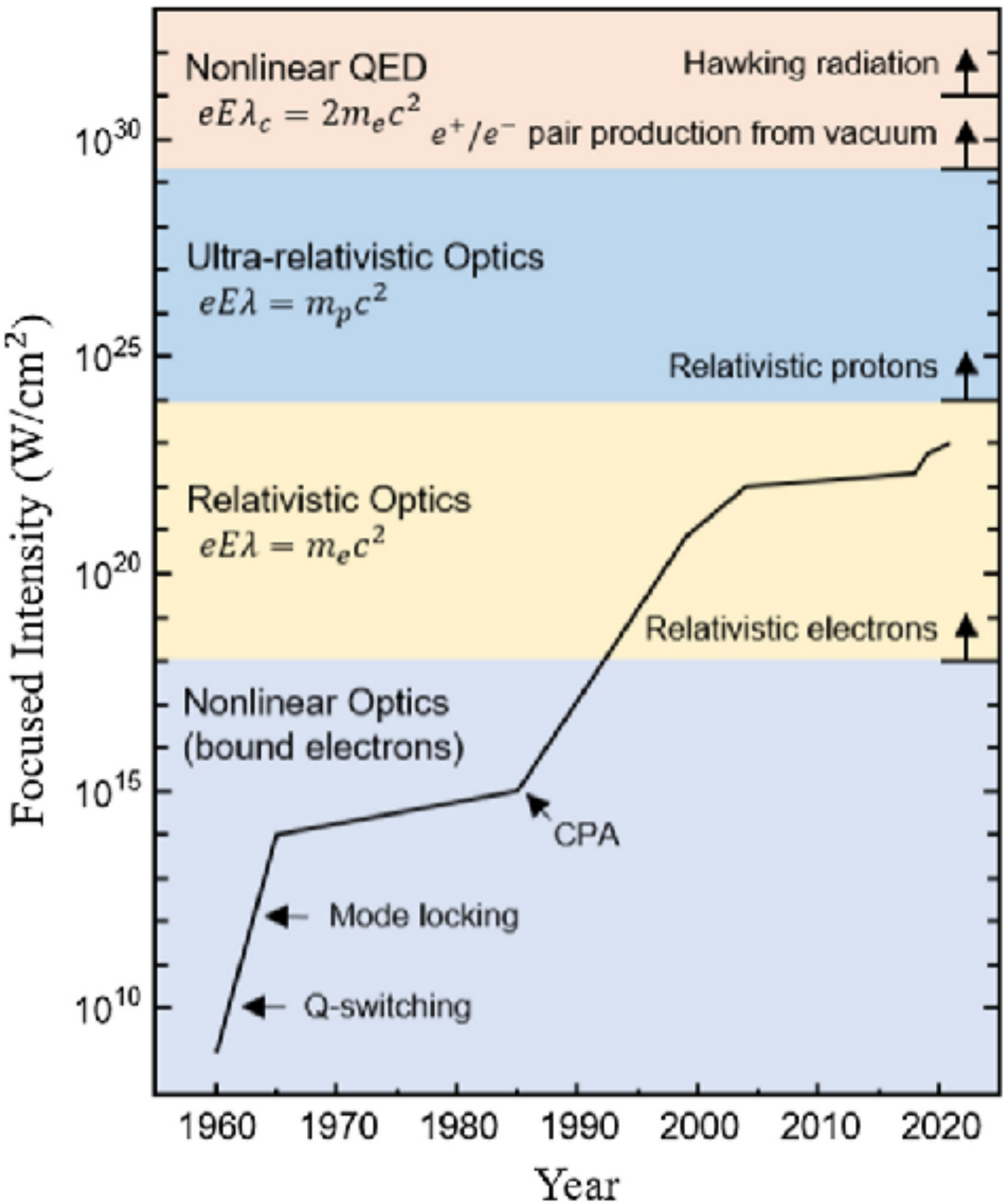
Accelerator technology	Peak Accelerating Field	Acceleration length to gain 100 MeV
Radiofrequency metallic cavities	Max: ~10^2 MV/m Typical: ~10 MV/m	Min: ~1 m Typical: ~10 m
Laser Wakefield Acceleration (LWFA)*	Max: ~10^5 MV/m Typical: ~10^4 MV/m	Min: ~0.001 m Typical: ~0.01 m





***Open challenge: improve performances of Laser Wakefield Acceleration. Numerical modeling is necessary!**

High Intensity Lasers and their interaction with matter



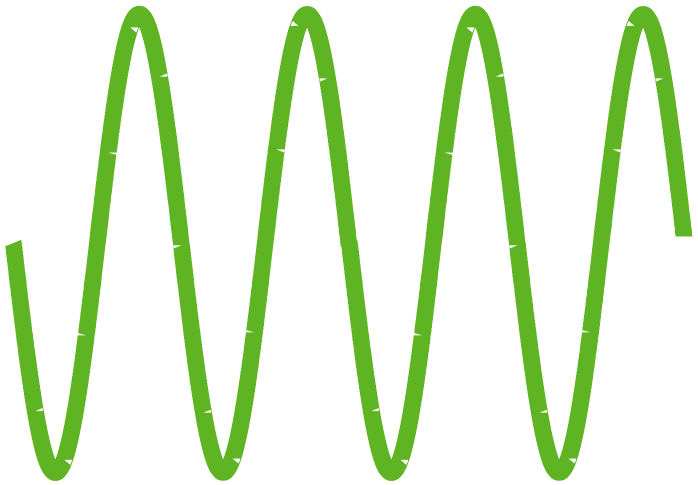
W. Bang et al, Journal of the Korean Physical Society 80,698–716 (2022)

Techniques:
- **Chirped Pulse Amplification**
- **Femtosecond Lasers**
- **Nonlinear Optics**
- …

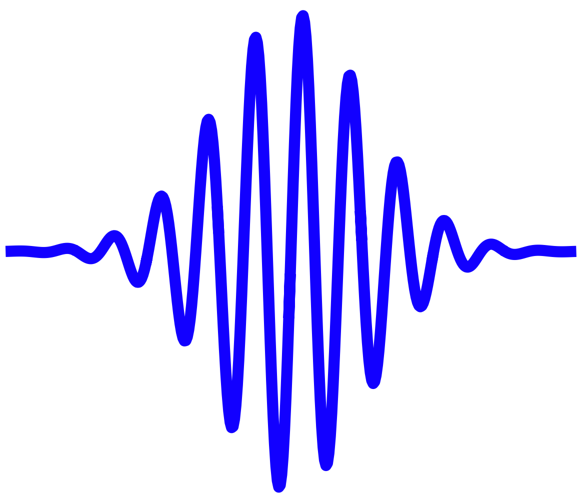
Physics Domains:
- Relativistic Fluids
- Physics of Relativistic Plasmas
- Relativistic Optics
- Laboratory Astrophysics

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

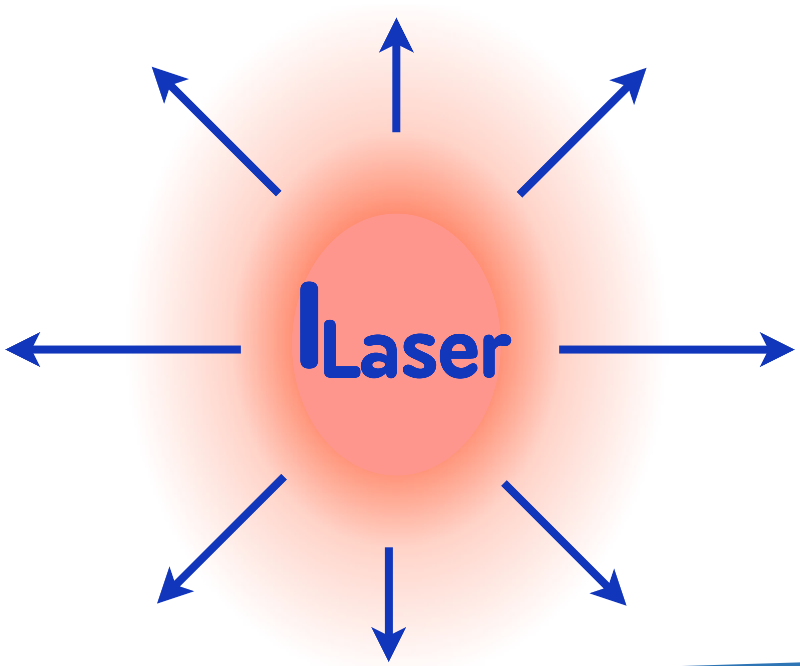
Ponderomotive Force:
the trigger for laser wakefield excitation in plasmas



Electron in infinite plane wave:
Oscillating Force



Electron in finite laser pulse:
Oscillating Force + Ponderomotive Force



$$F_{pond} \propto -\nabla I_{Laser}$$

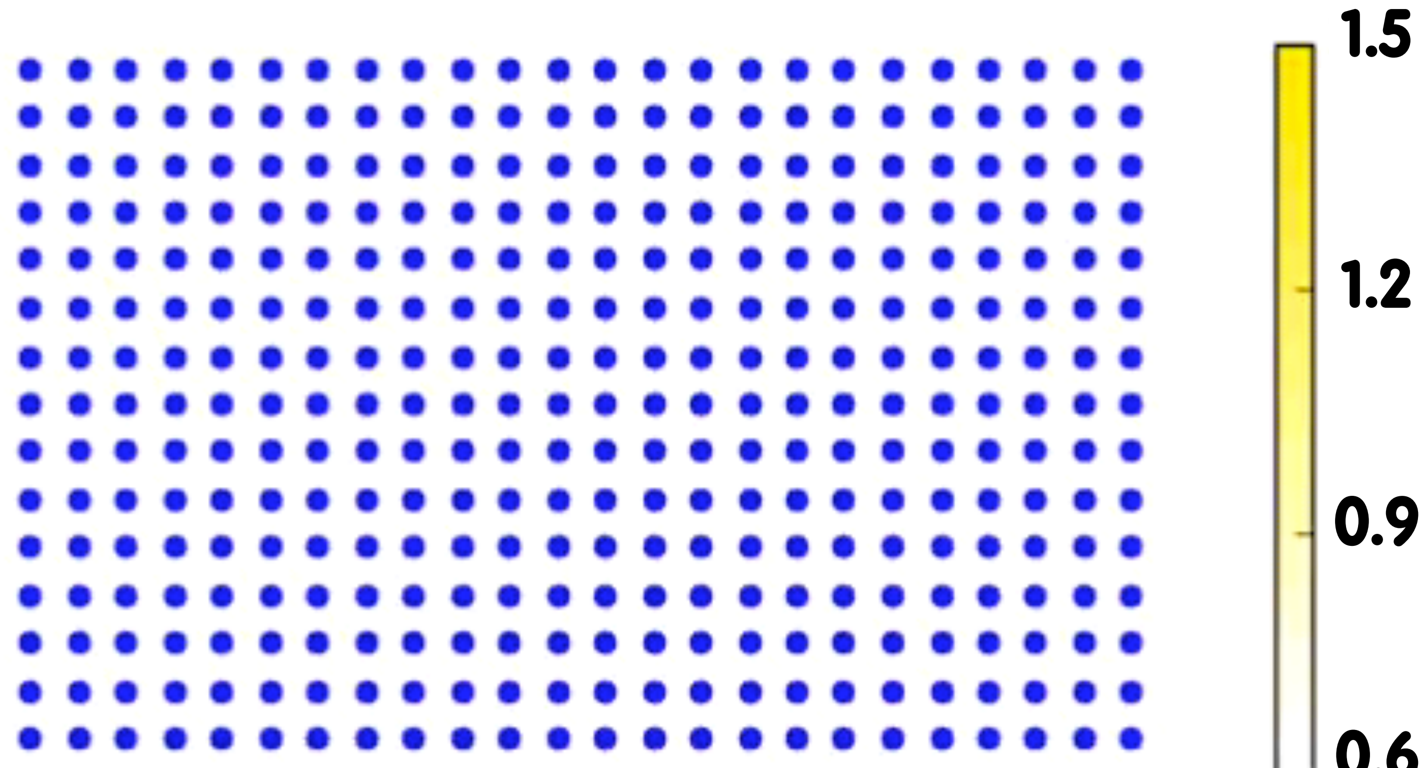
a.k.a. radiation pressure
(Relativistic formula is more complex)

Laser Wakefield Acceleration (LWFA): plasma wave excitation by the laser ponderomotive force

Laser Beam Duration: 28 fs

Ponderomotive Force:

$F = -\nabla I_{Laser}$



Transverse Electric Field (A.U.)

Electrons ● Immobile Ions ● $M_i >> m_e$

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

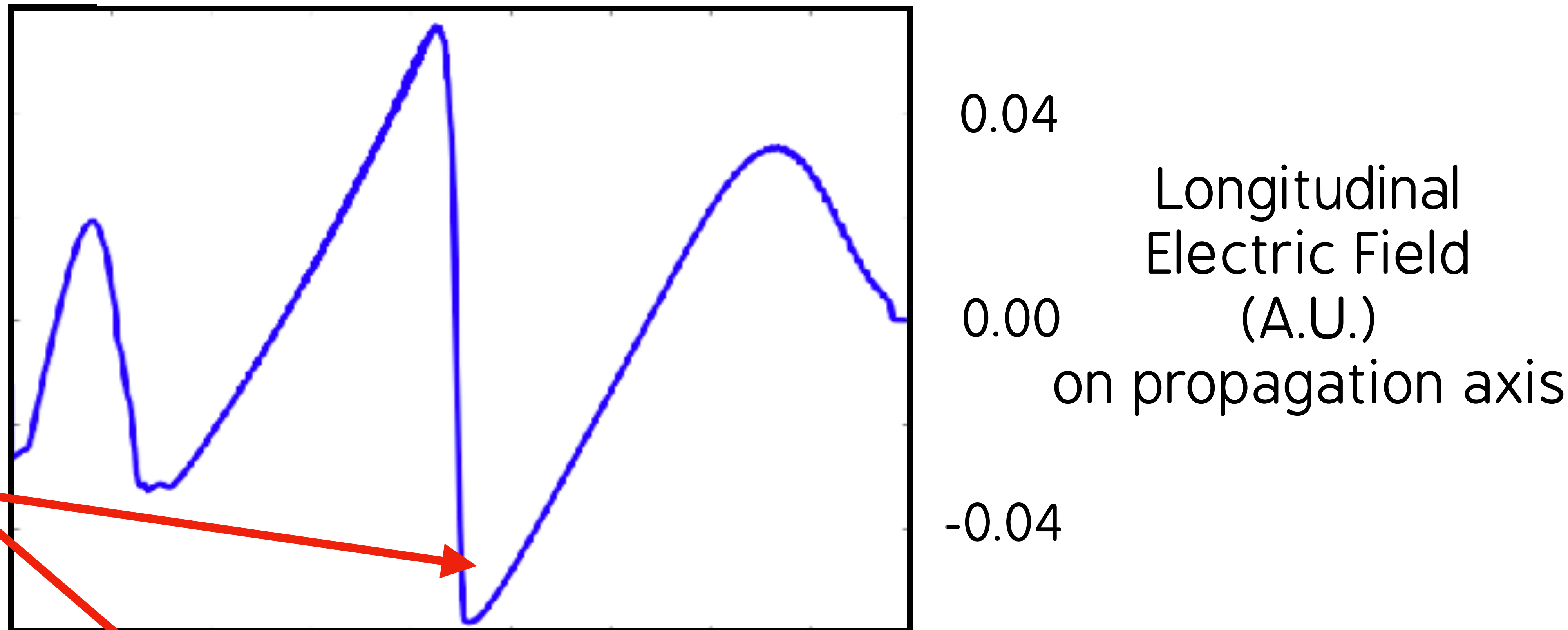
Plasma wavelength: ~20 μm

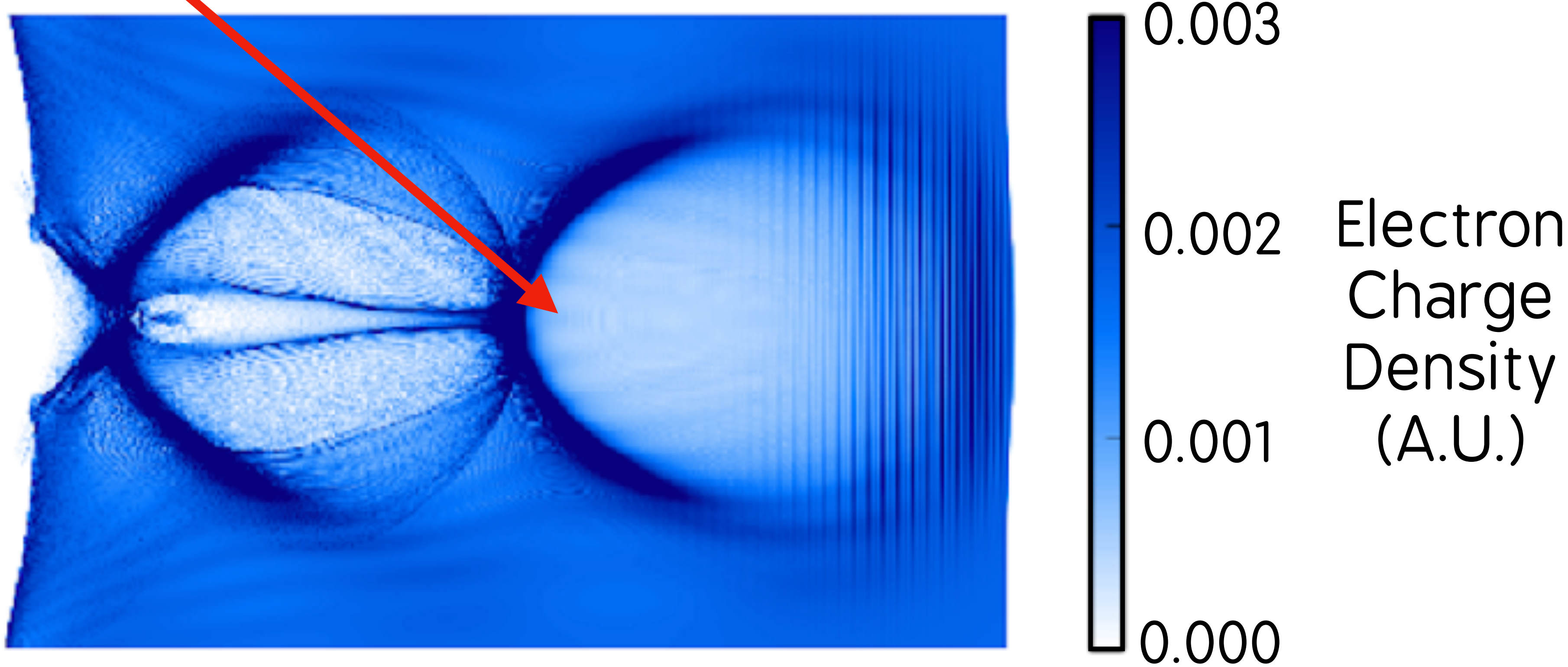
Electron Charge Density (A.U.)

Laser Wakefield Acceleration (LWFA): accelerating electric field

E > 100 GV/m

Relativistic electrons injected here are accelerated towards the right

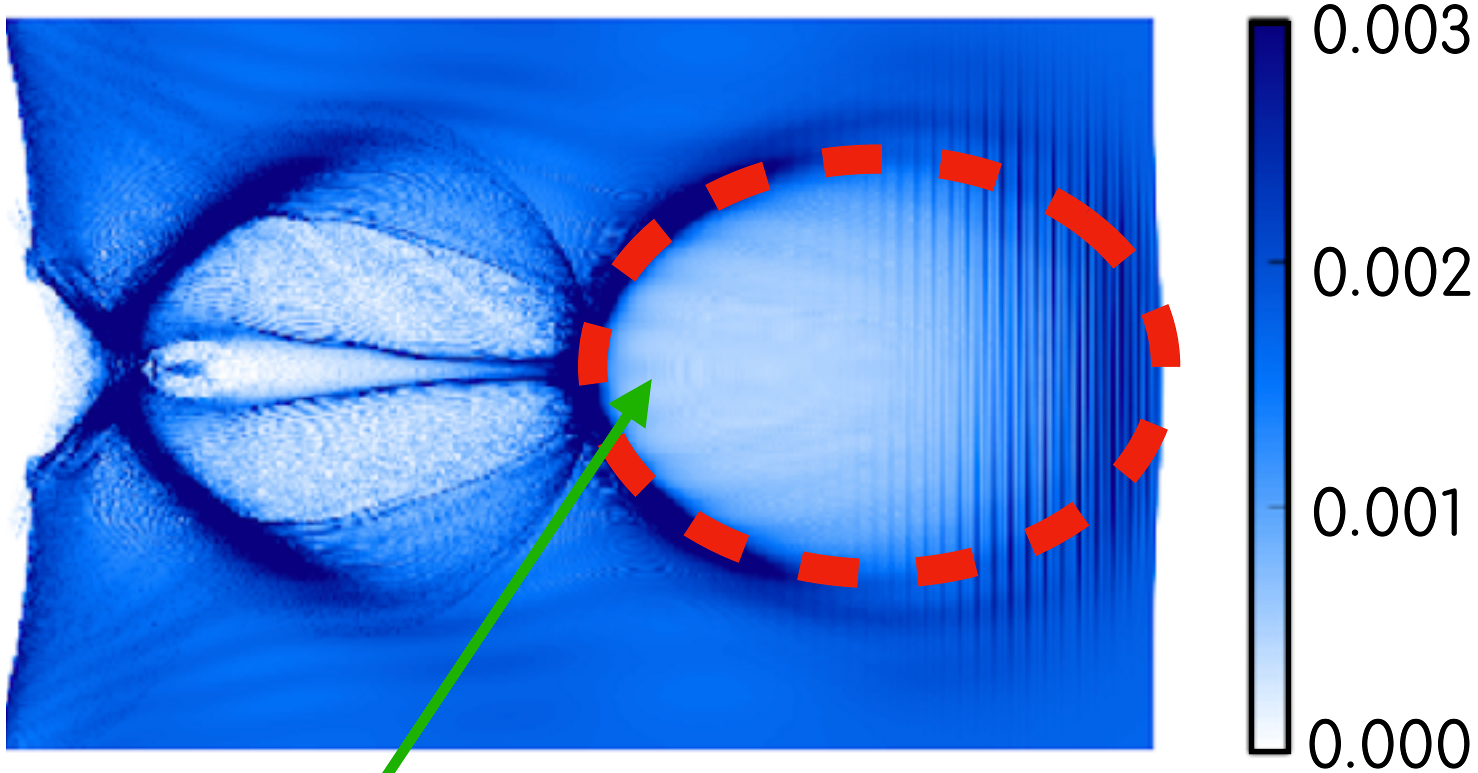




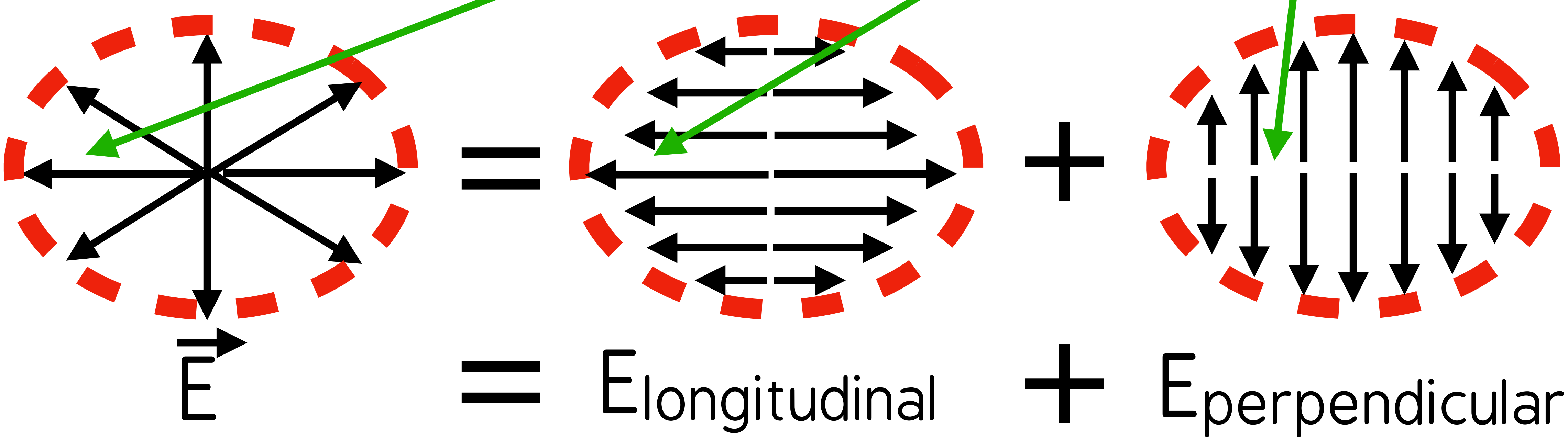
Nonlinear regime here

LWFA: Electric fields inside the "bubble"

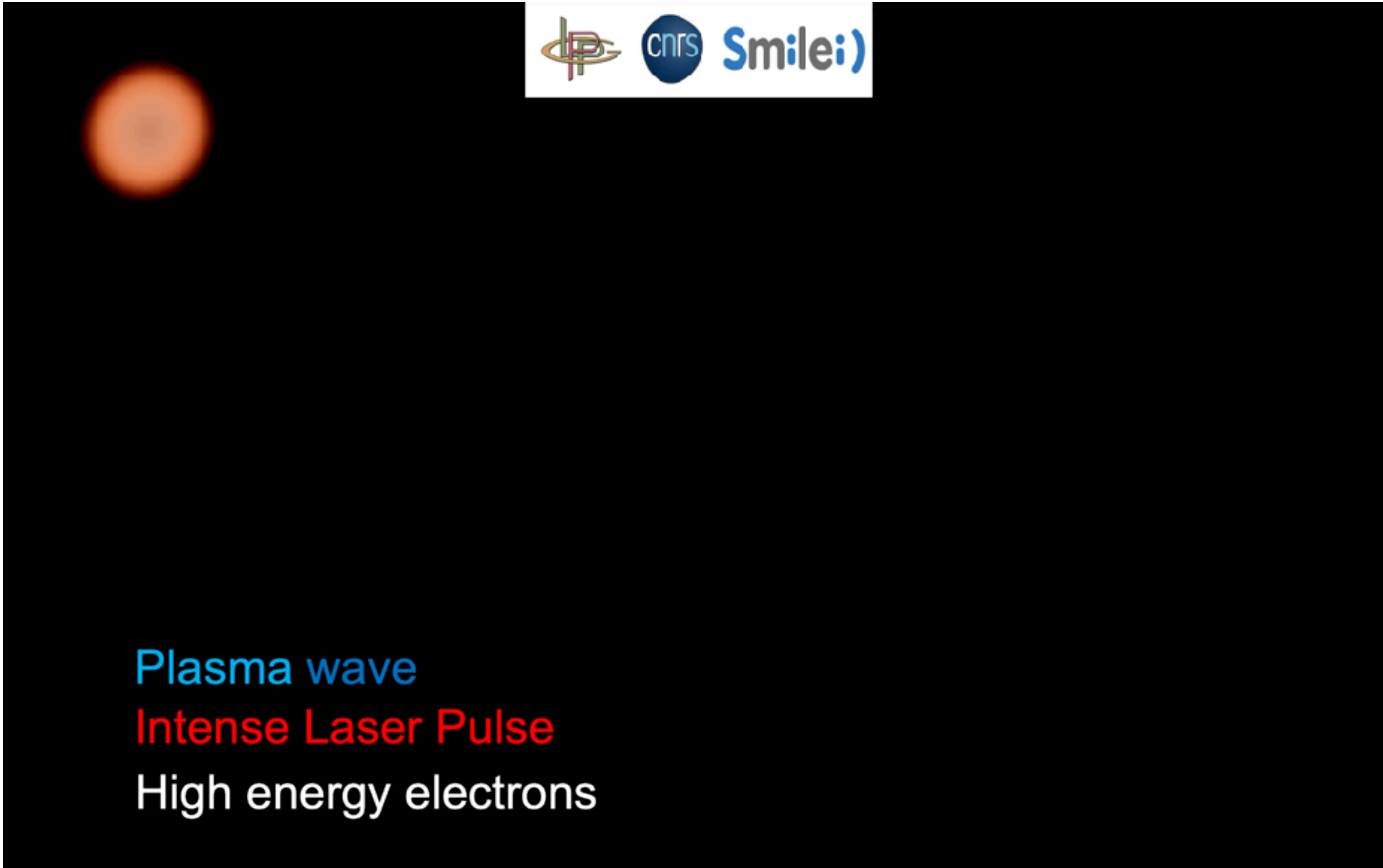
Plasma wavelength: ~20 μm



Relativistic electrons injected here are both accelerated and focused

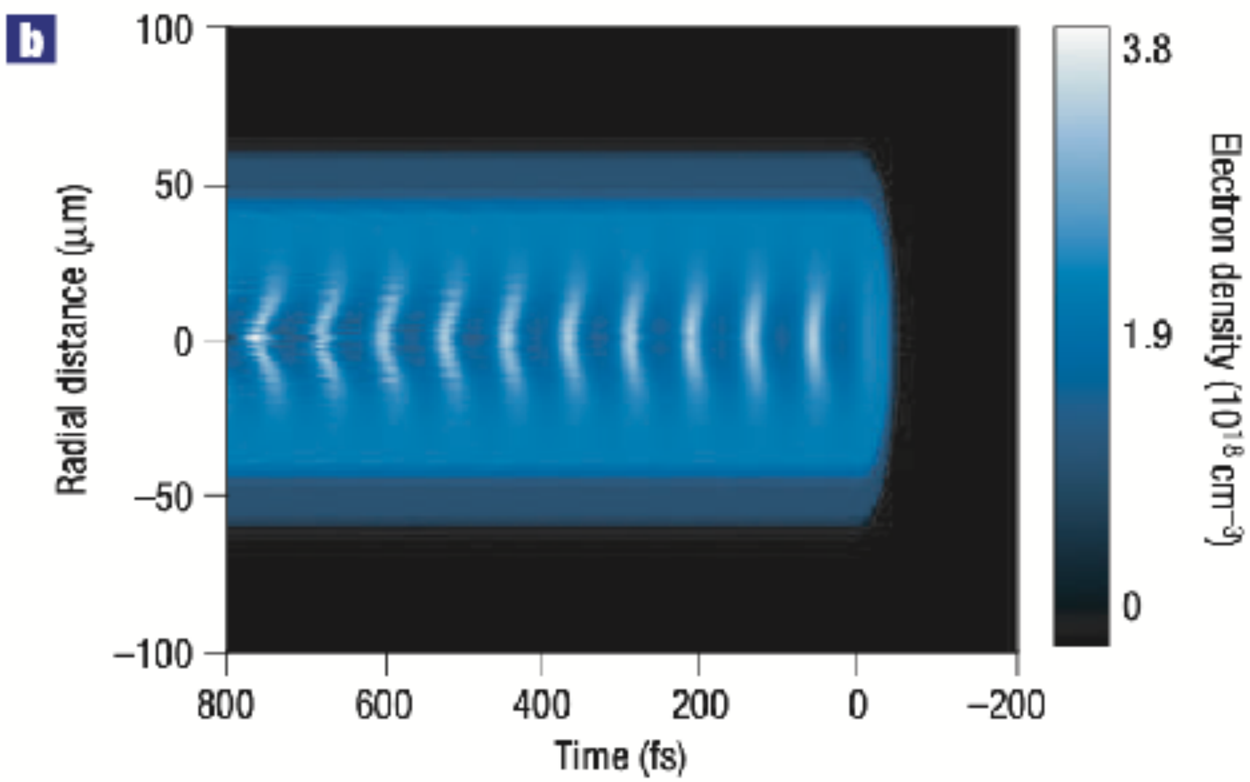


Laser Wakefield Acceleration with ionization injection

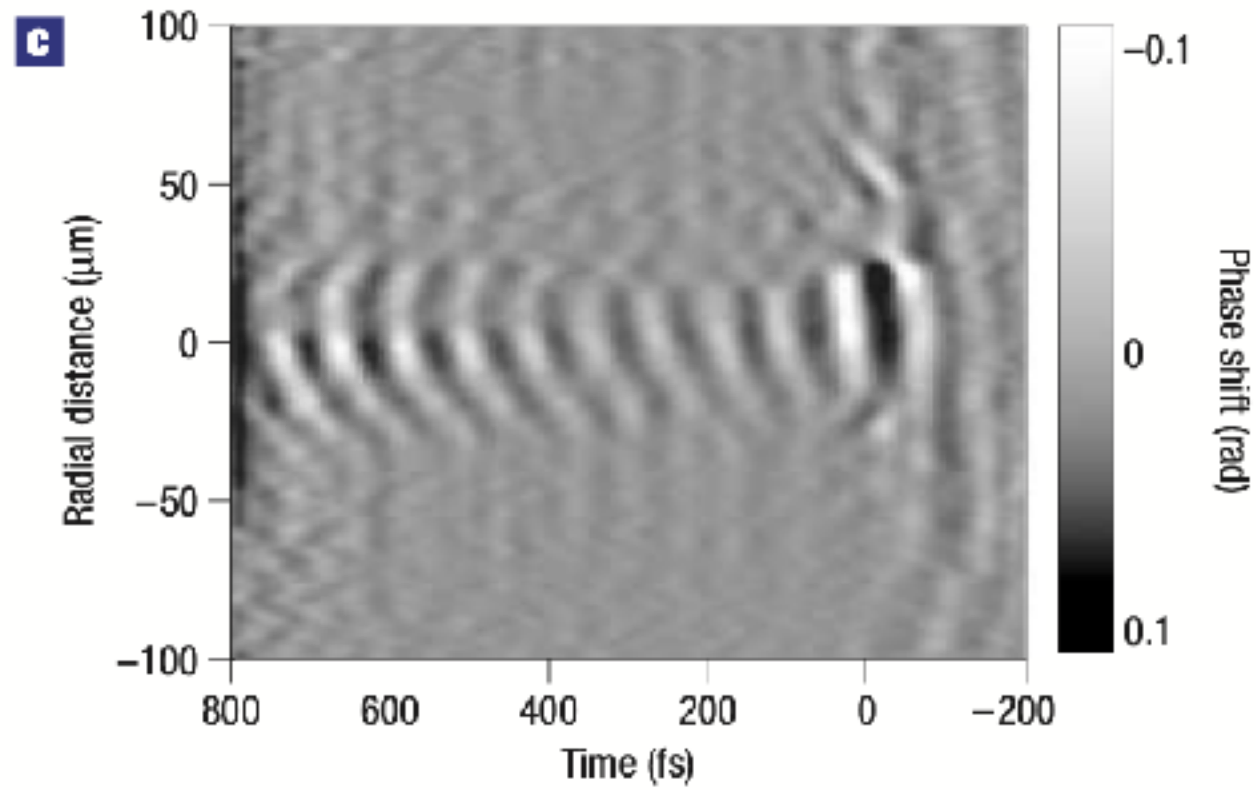


Laser Wakefield Acceleration (LWFA): visualising the plasma waves

Simulation



Experiment



N. Matlis et al., Snapshots of laser wakefields, Nature Physics (2006)

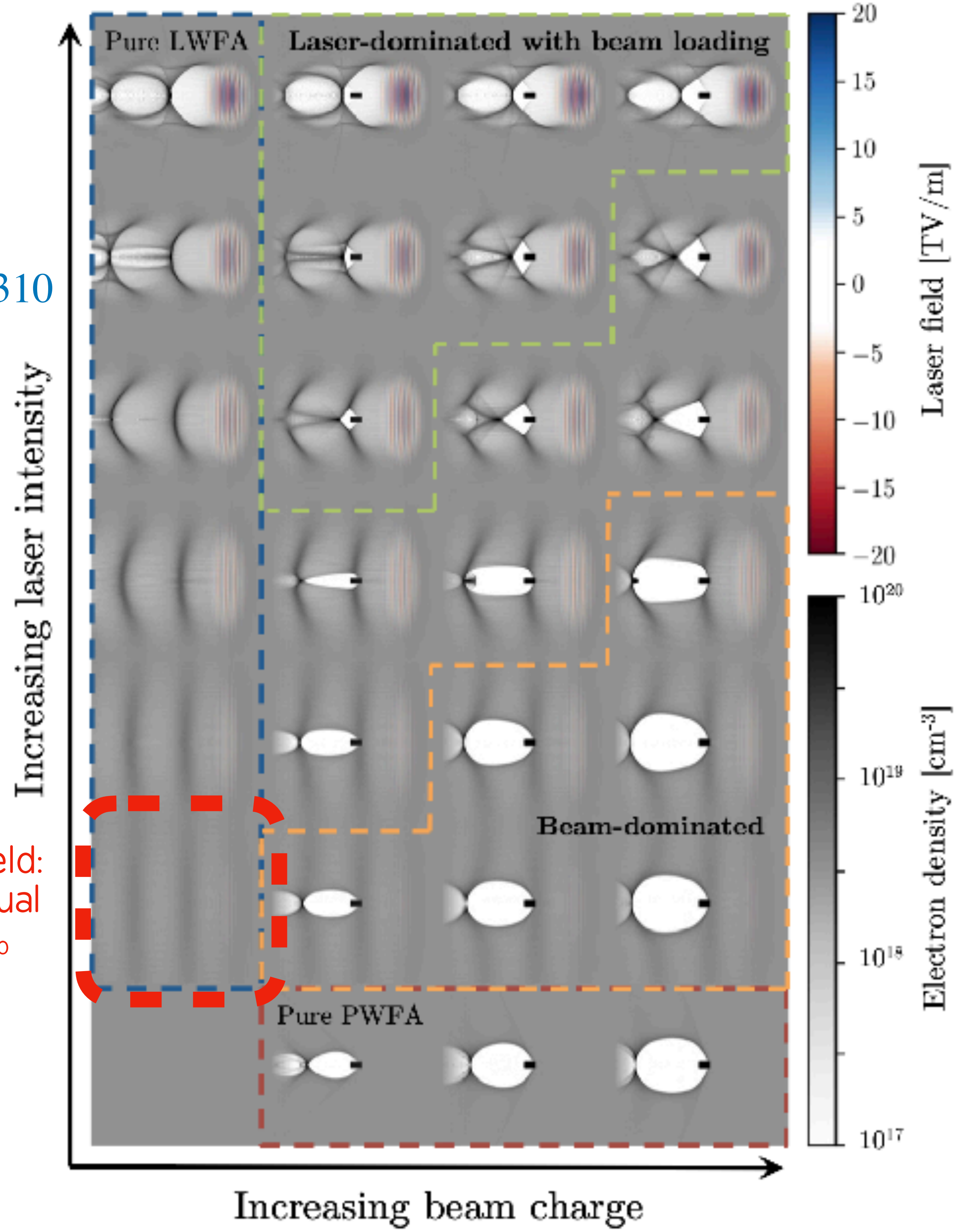
Plasma wakefields can be driven by electron beams too

J. Götzfried et al.,

Phys. Rev. X (2020)

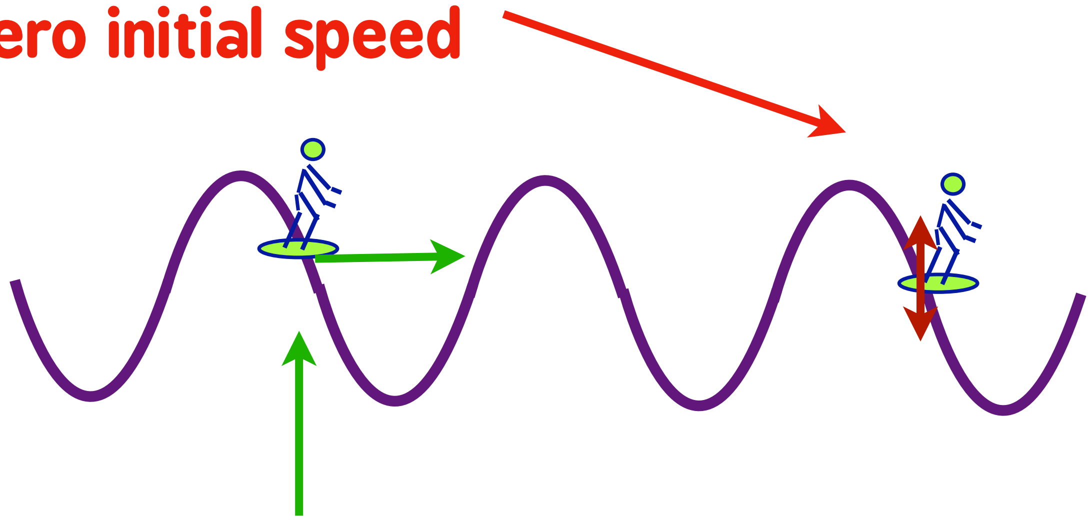
https://arxiv.org/pdf/2004.10310

Linear regime of laser wakefield:
Plasma waves with period equal
to the plasma wavelength λ_p

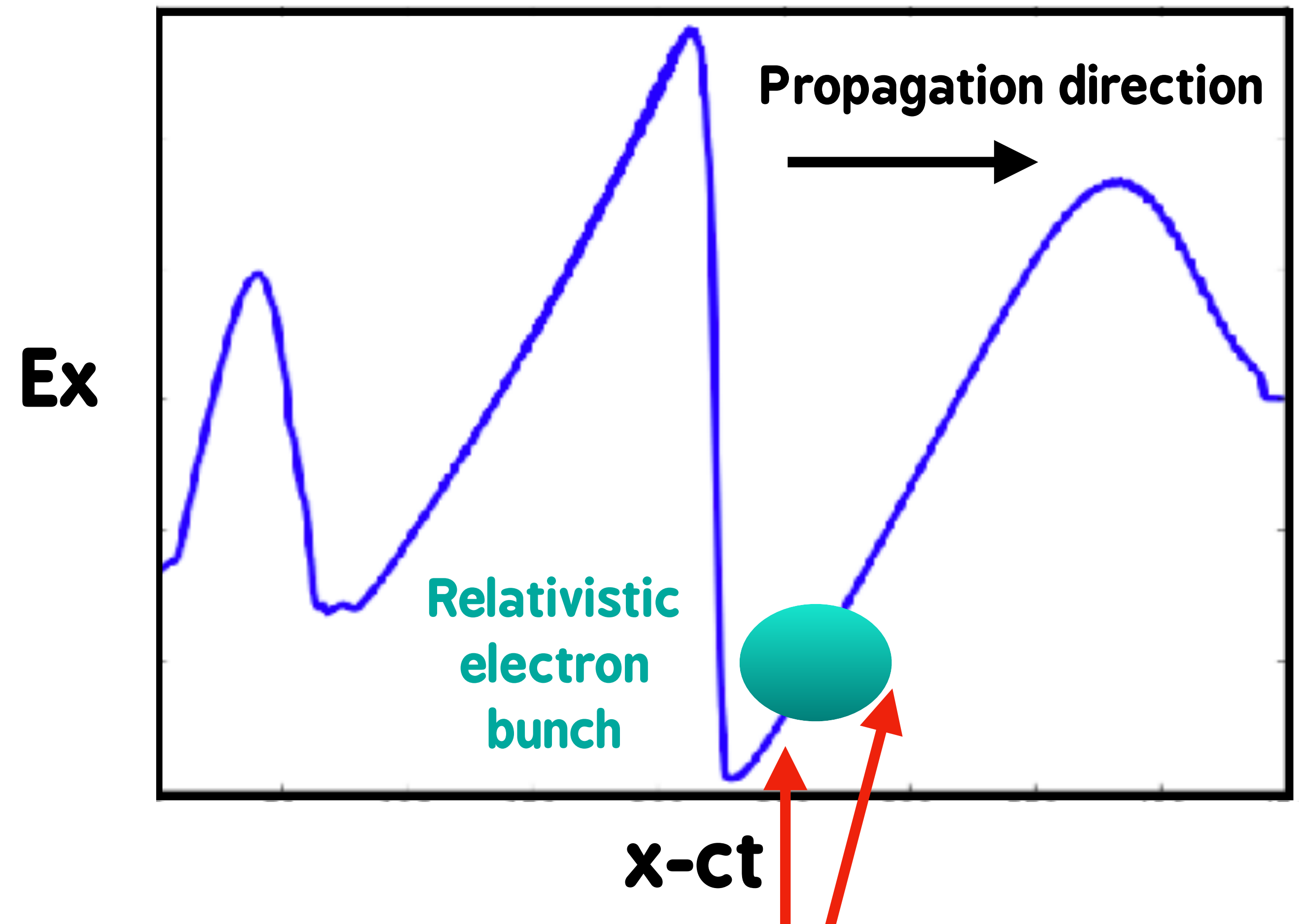


Laser Wakefield Acceleration (LWFA) challenges: injection and energy spread

Surfer with Zero initial speed



Surfer with sufficient initial speed, injected in the proper phase



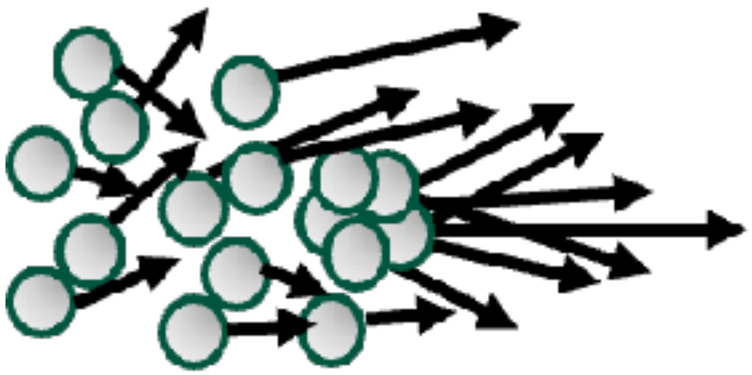
Electrons injected in different phases of the wave will experience a different accelerating field

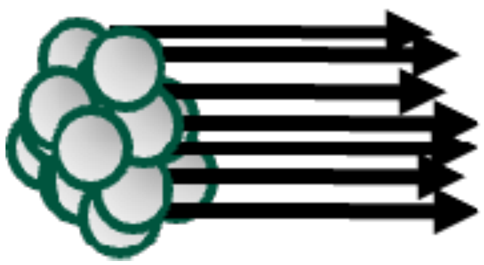
Energy spread increase

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

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

LWFA objective (not the only one): realize compact electron accelerators for applications





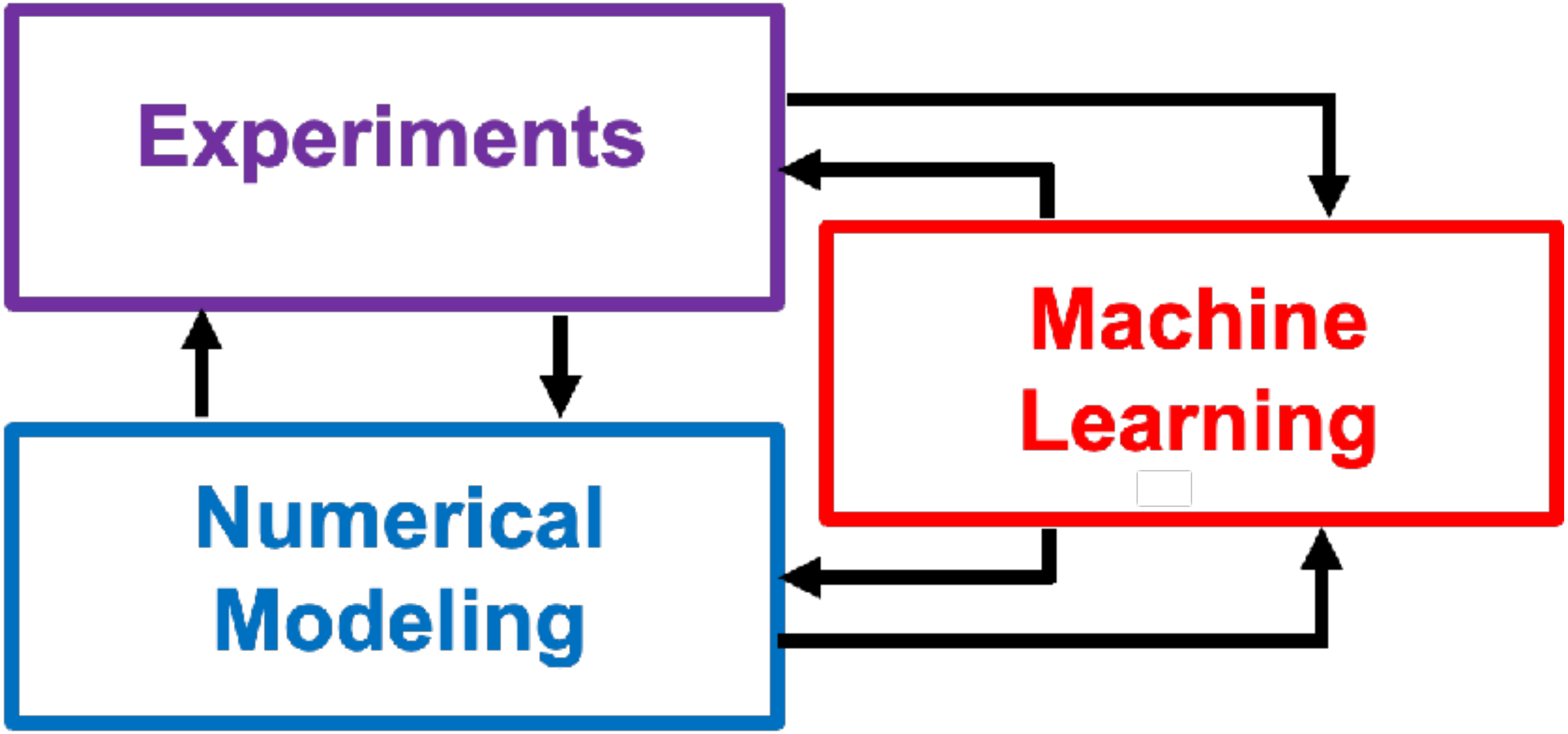
- Few electrons or many dispersed in space
- High divergence
- Different Energies (high energy spread)

- Many electrons in a small volume
- Low divergence
- Similar Energies (low energy spread)

Outline

- Basics of laser wakefield acceleration
- Numerical simulation of plasma acceleration: PIC codes
- Introduction to the case study and the practical

LWFA investigation techniques



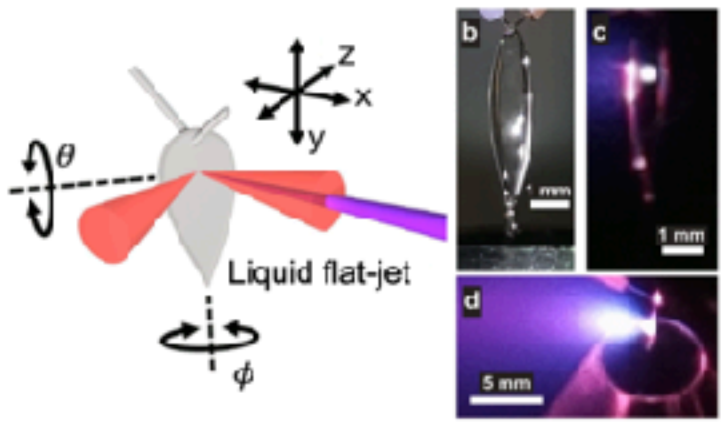
Why numerical simulation is necessary for LWFA?

- **No 3D analytical solutions are available for the general case**
- **You cannot measure everything simultaneously**
- **Simulations before the experiment:**
  - Study new physical phenomena
  - Conceive new kind of experiments
  - Design experiments (also using Machine Learning)
- **Simulations after the experiment:**
  - Analyze the data (also using Machine Learning)
  - Understand the physics

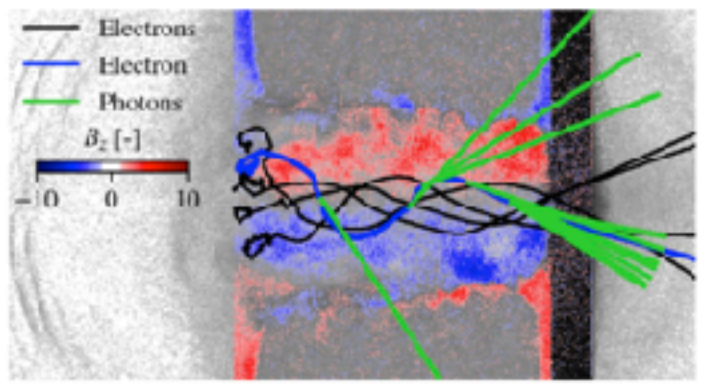
Particle in Cell (PIC) method :
essential kinetic plasma investigation technique

Applications From Laboratory plasmas to Space and Astrophysical plasmas

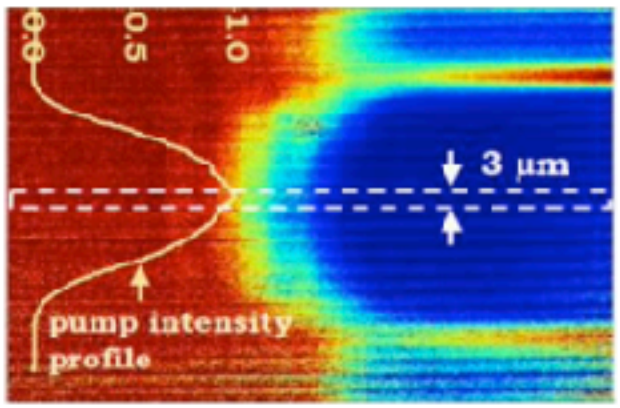
High-Harmonic Generation



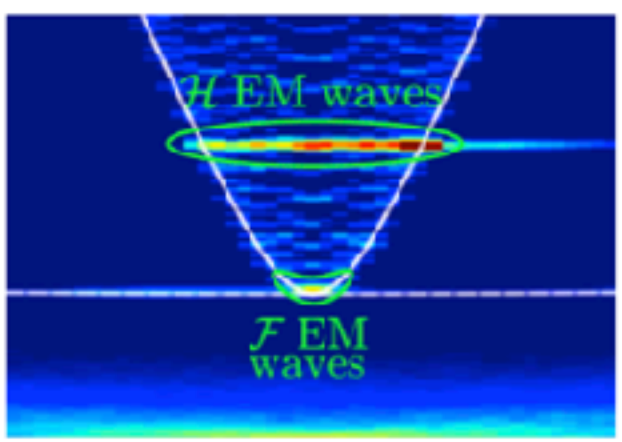
Radiation sources from DLT



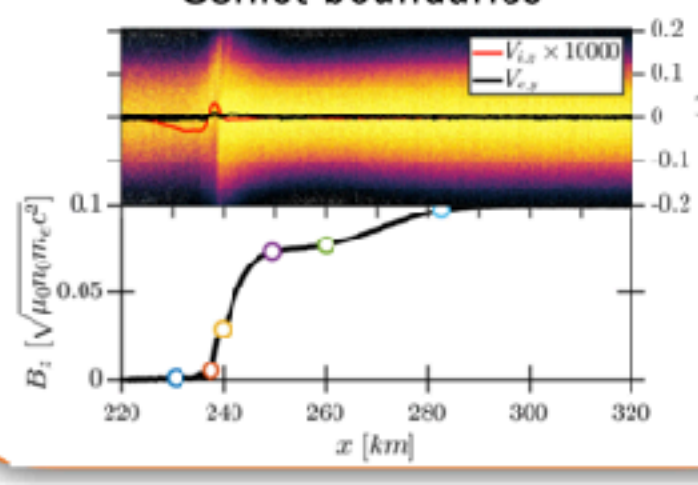
Solid to plasma transition



Solar radio-burst



Comet boundaries



Collisionless shocks & Dark Matter

Some studies performed with Smilei)

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

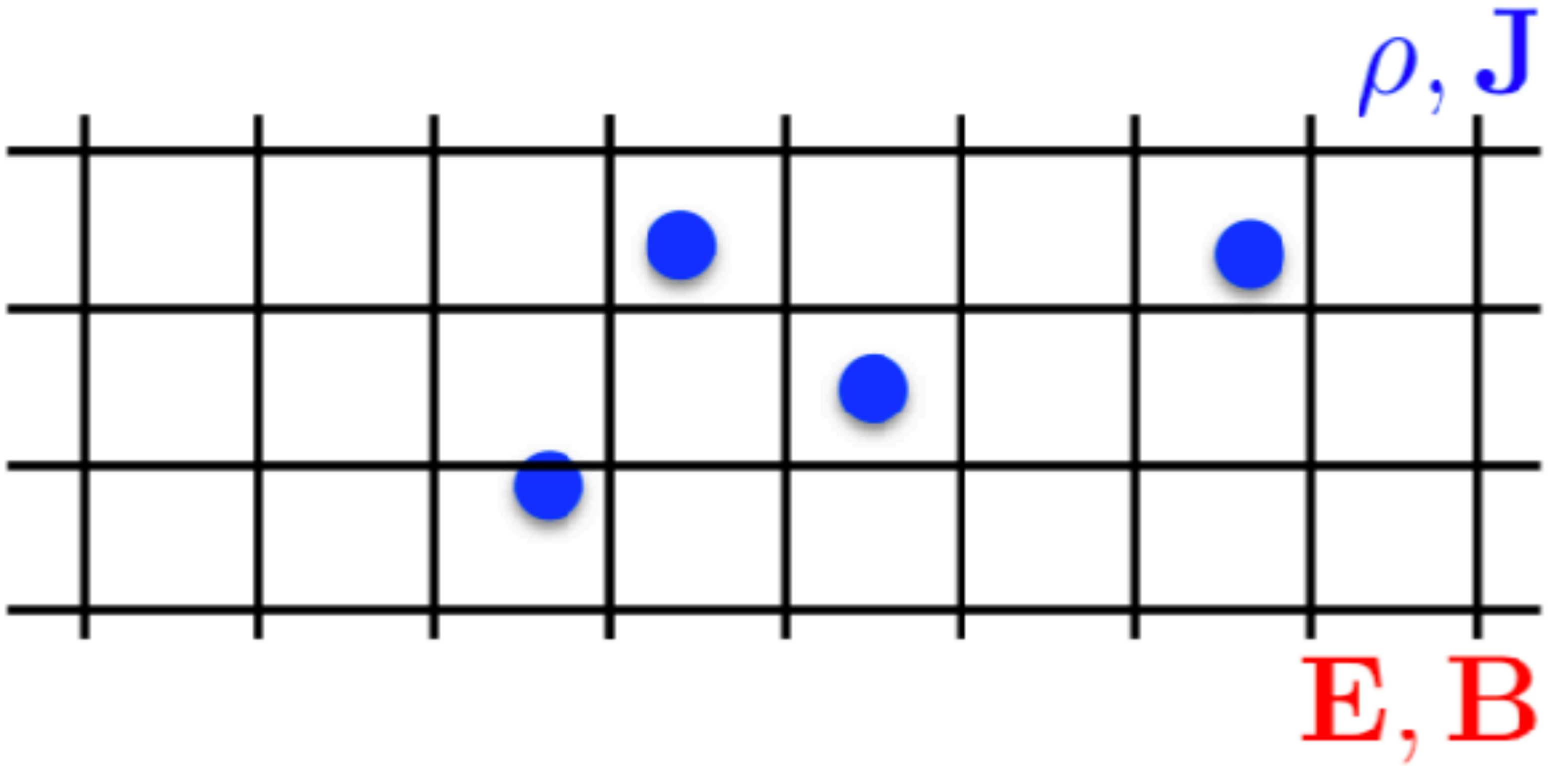
Particle in Cell concept

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

+

Discretize space with computational grid

Define E, B, ρ, J on the grid cells

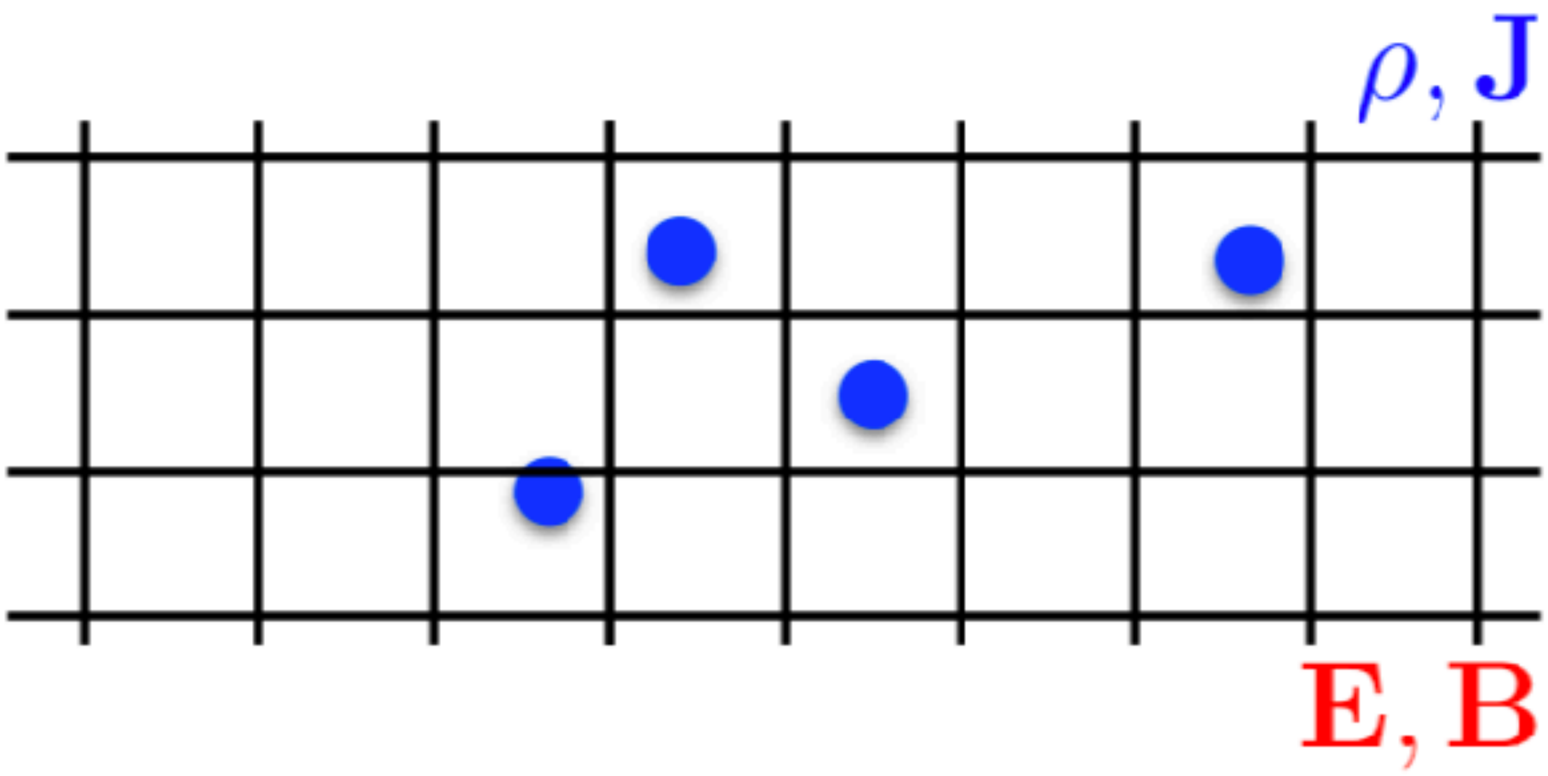


Particle in Cell concept

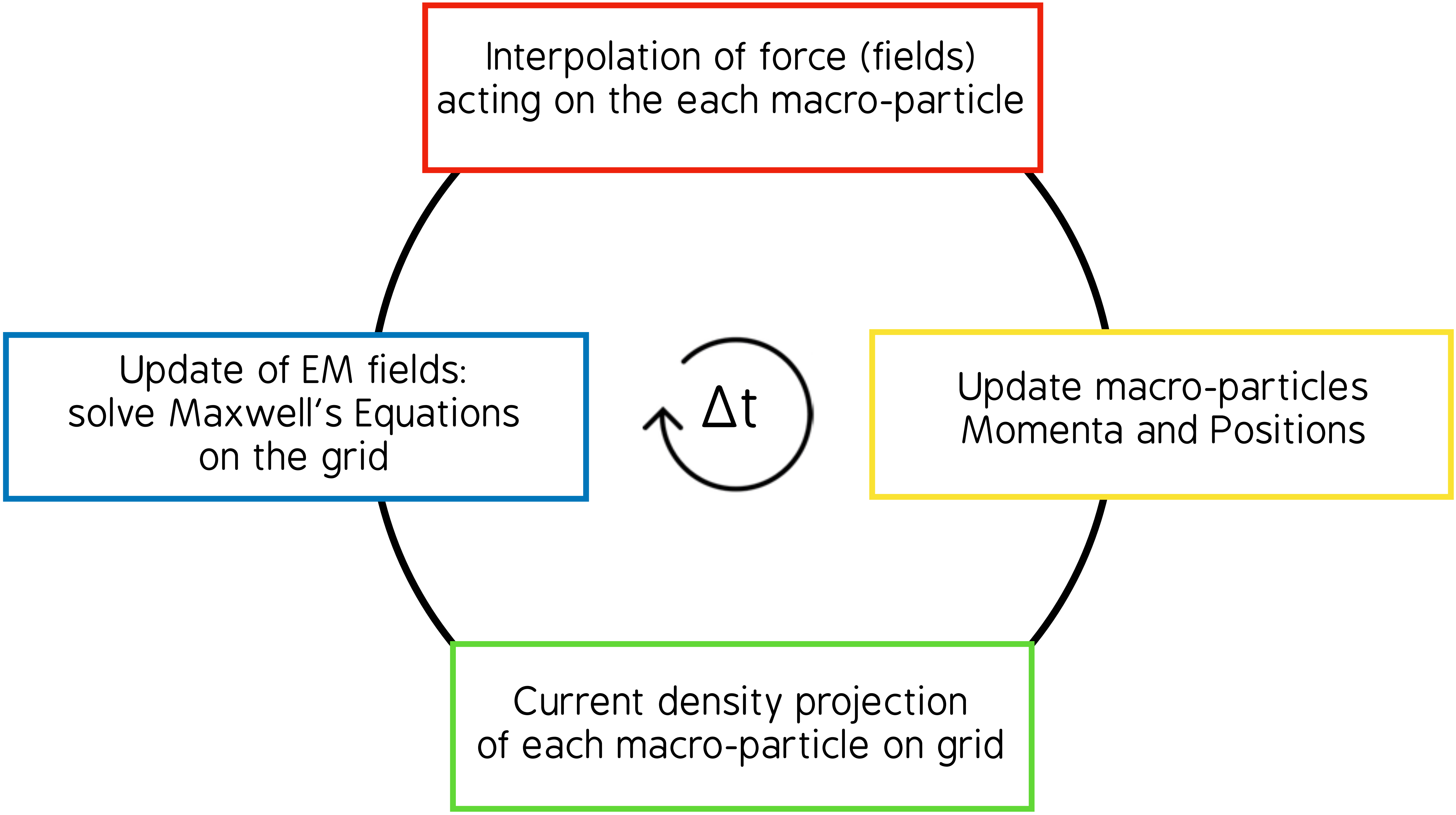
Macro-Particles:
evolve following the characteristics of Vlasov Equation
(In the PIC method they look like equations of motion)

+

Electromagnetic Fields: evolve following Maxwell's Equations



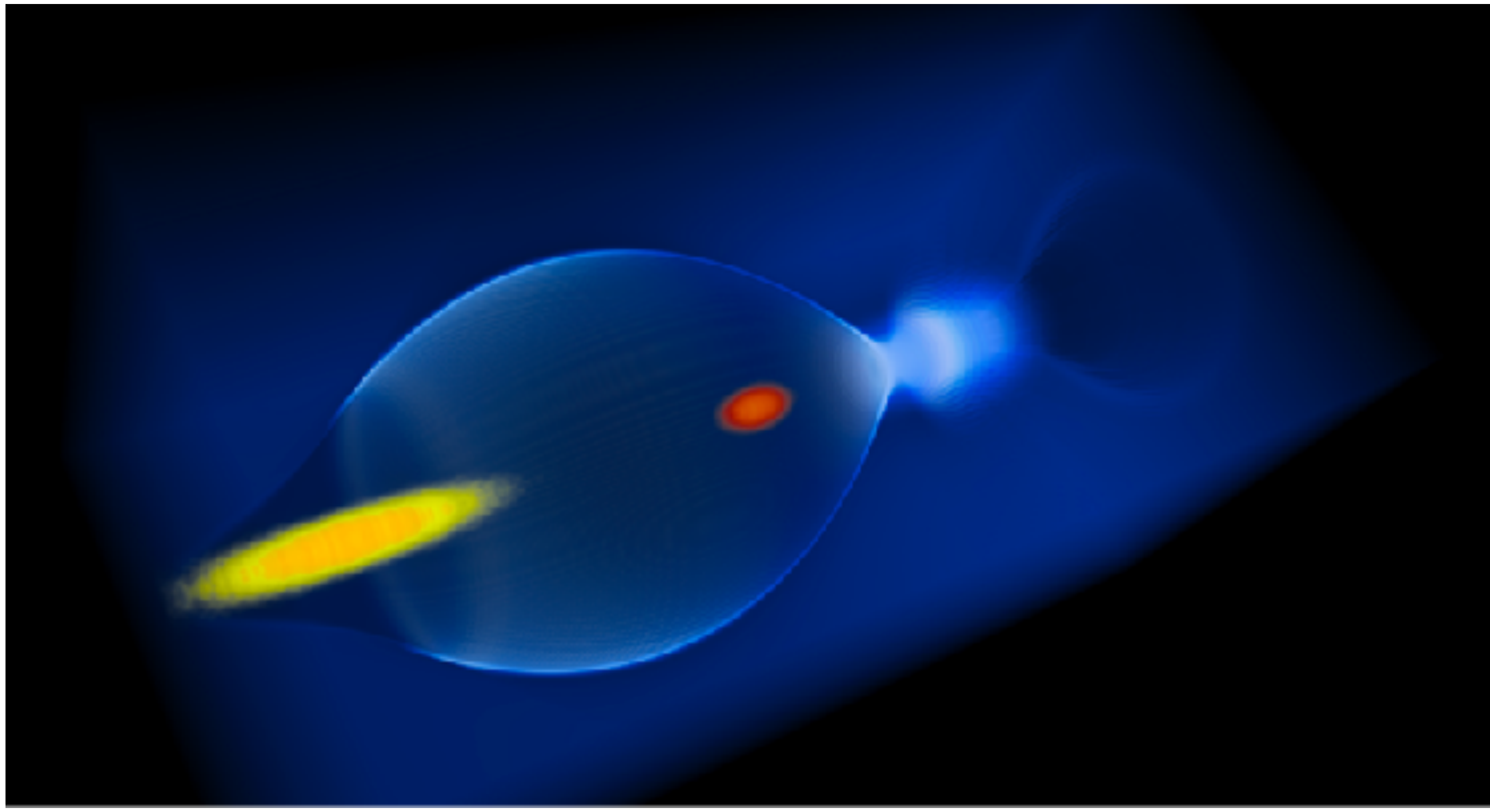
Particle in Cell modelling is self-consistent



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
- Element chat with the developers

High Performance

- MPI + OpenMP parallelization
- dynamic load balancing
- adaptive vectorization
- GPU computing on both NVIDIA and AMD architectures

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
- >200 publications using the code up to 2024

Additional material

Extensive Documentation
(Installation, Use, Postprocessing, ...)

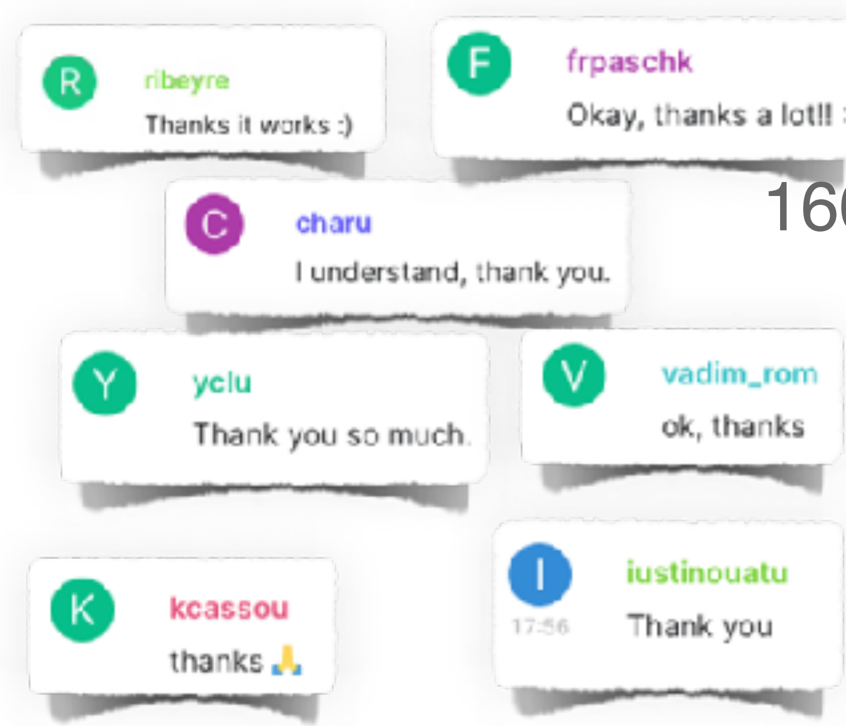
https://smileipic.github.io/Smilei/index.html

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

Presentations from the
4th User & Training Workshop at ELI-Beamlines

https://indico.math.cnrs.fr/event/9577/

Questions?
We answer on the Element chat



1600 messages in 2021

Outline

- Basics of laser wakefield acceleration
- Numerical simulation of plasma acceleration: PIC codes
- Introduction to the case study and the practical

Example of input file

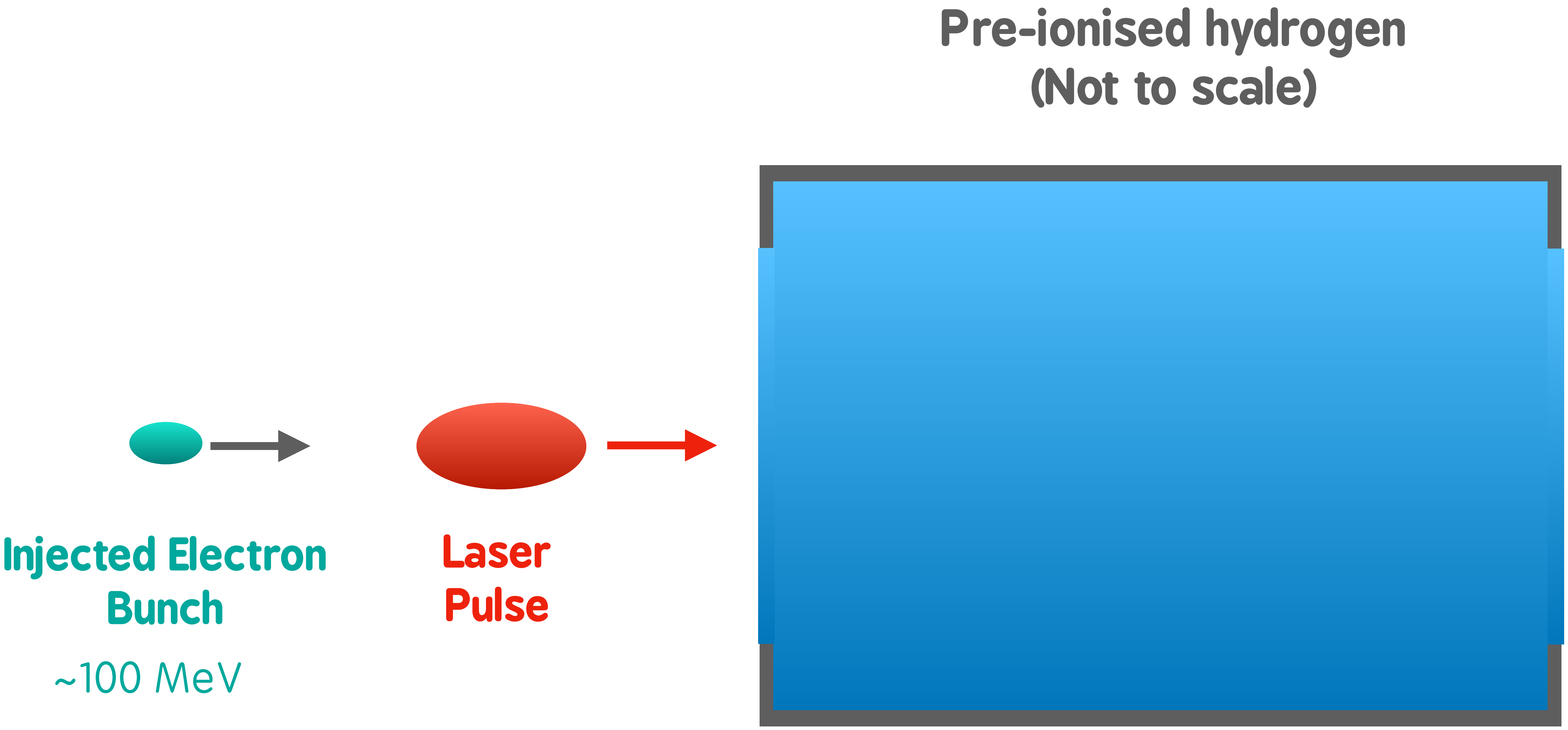
```
Main(
    geometry           = "1Dcartesian",
    timestep           = 0.009,
    cell_length        = [0.01],
    ...
)

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**

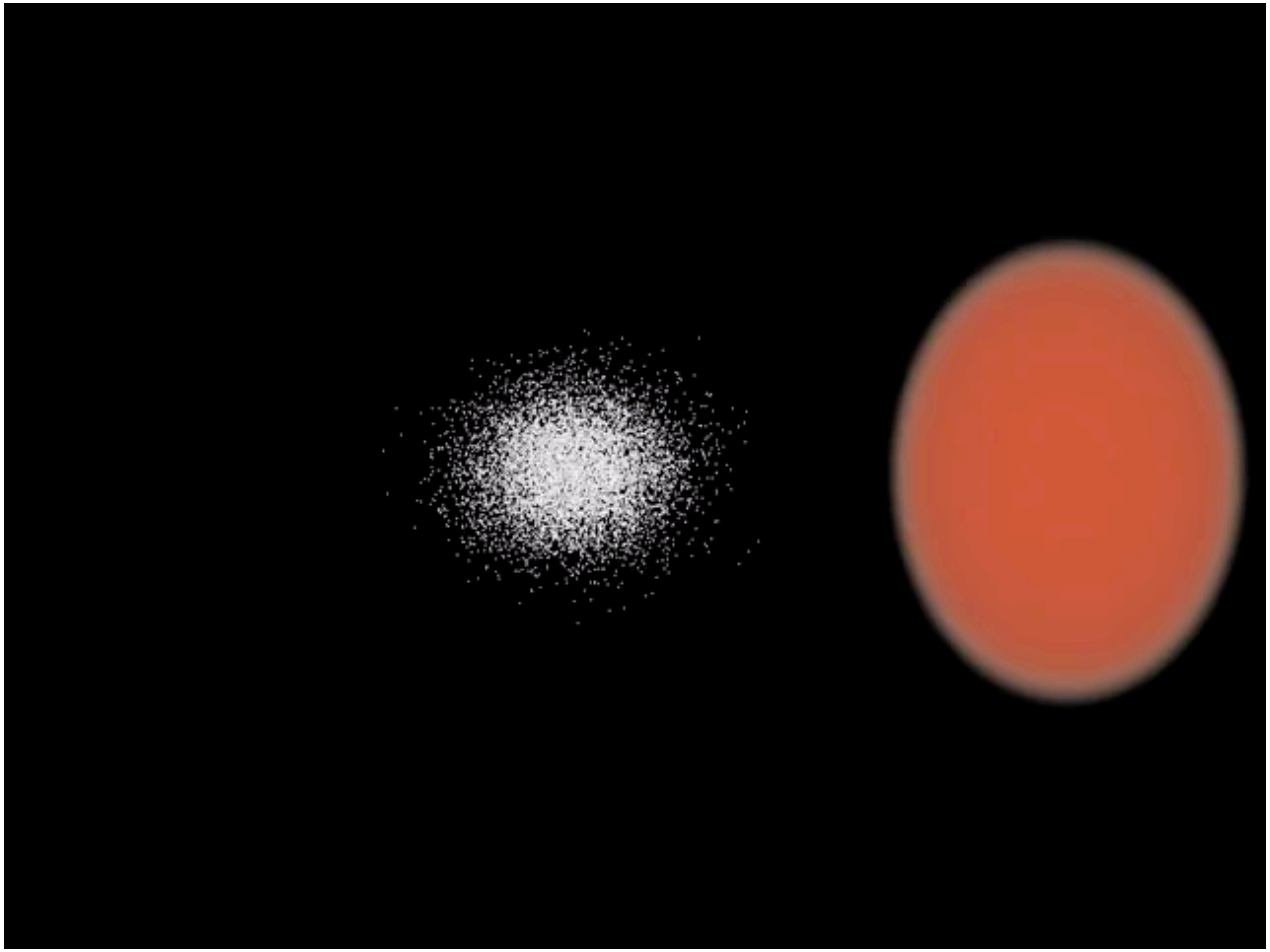
Laser / Plasma profiles = Functions (also user-defined)

If Python can read it, SMILEI can read it

Case study: LWFA with external injection



3D rendering of LWFA with external injection

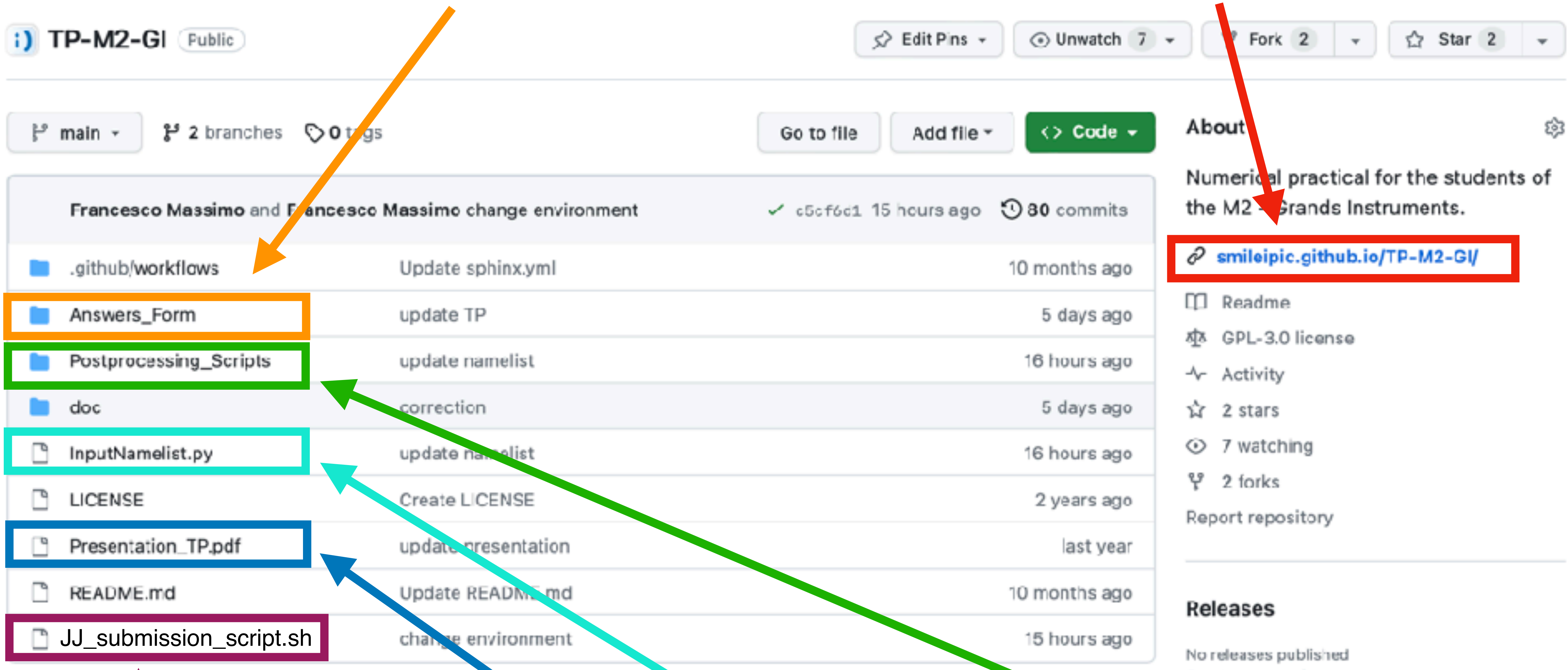


Practical material

https://github.com/SmileiPIC/TP-M2-GI/tree/main

Answer Form for the Exercises

Link to TP documentation and Exercises



These slides

Input Namelist file to run simulations

Postprocessing Scripts

Simulation submission file

Practical exercises

- Read **in detail** the **TP documentation**
  https://smileipic.github.io/TP-M2-GI/index.html
- Solve the exercises in the **TP documentation** progressively, e.g.

> **Exercise 1:**
>
> Assuming $\lambda_0 = 0.8\mu m$ (a Ti:Sa laser system), what is the value of the critical density n_c?
>
> What is the value of the reference electric field $E_0 = (2\pi m_e c^2)/(e\lambda_0)$?
>
> This choice of λ_0 will be used throughout all subsequent exercises.
>
> **Hint:** Some lines at the start of the `InputNamelist.py` file can help you in the calculations.

> **Exercise 5:**
>
> In the next exercise we will check that the Gaussian laser pulse diffracts following the theory for a Gaussian beam [Siegman]: $w(x) = w_0\sqrt{1+x^2/x_R^2}$, where w_0 is the laser waist size at the focal plane position, $w(x)$ the laser waist size at propagation distance x, x_R is the Rayleigh length $x_R = \pi w_0^2/\lambda_0$.
>
> What is the theoretical Rayleigh length x_R?

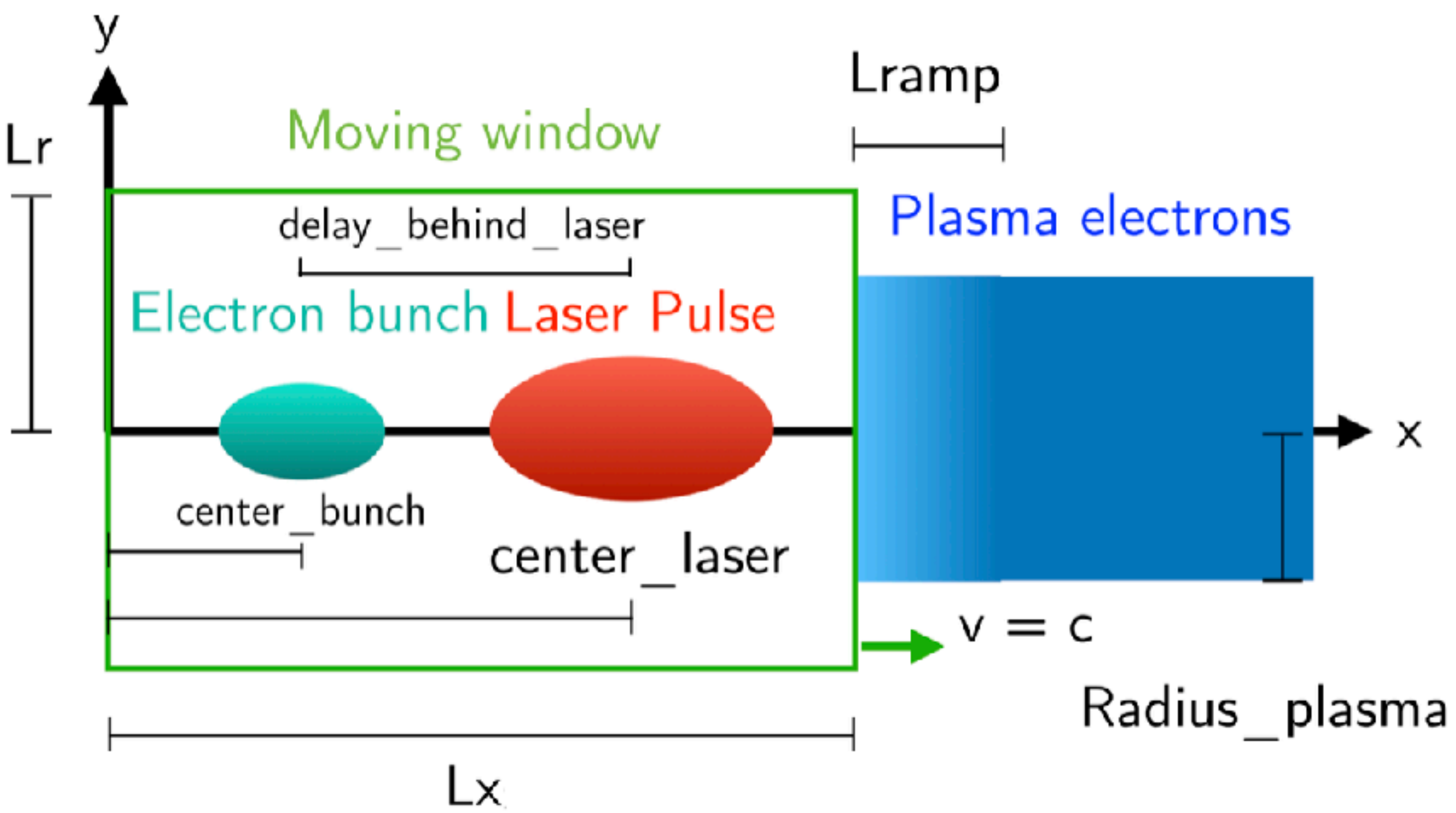
> **Exercise 10:**
>
> Launch a new simulation with `a0=1.8`. This simulation will be in the nonlinear regime ($a_0 > 1$), so the plasma wave will not be sinusoidal. You can visualize both the normalized absolute value of the envelope of the laser field and the electron number density by defining a transparency for the parts where the latter field is lower than a threshold `vmin`:
>
> ```
> Env_E  = S.Probe.Probe1("Env_E_abs",units=["um"],cmap="hot",vmin=0.8,transparent="unde
> Rho    = S.Probe.Probe1("-Rho/e",units=["fs","um","1/cm^3"],cmap="Blues_r",vmin=0.,vma
> happi.multiSlide(Rho,Env_E,xmin=0,figure=10, xlabel="x [um]",ylabel="y [um]")
> ```
>
> Using `timestep=2500` in the definition of `Env_E` and `Rho`, and then using `multiPlot` instead of `multiSlide`, you should have a plot of the data at half of the propagation length.
>
> Include this image in your answers.

- Run simulations when necessary,
  following the instructions in the **TP documentation**
- Fill the answers report **Answers_Form.docx**
  (You can download it from GitHub
  https://github.com/SmileiPIC/TP-M2-GI/tree/main)

Suggestions

- Read **in detail** the **TP documentation**
- **Understand** the physical set-up
- **Any doubts? Ask the instructor**
- Solve the exercises **progressively**
- **Be organized:** Create one folder for each simulation asked by the exercises (to **avoid losing data**)
- Feel free to adapt the commands in the **TP documentation**
- When an image is asked for the report, save it or make a screenshot
- **Better do few exercises but understand them at 100%**

Physical elements in the last simulations of the TP



Input Namelist structure

As in Python, to decomment one block, remove the # symbol

Decomment also the variable used in the block!

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

Example: commented Laser block (Laser OFF)

```
######################### Define the laser pulse

## laser parameters
#laser_fwhm_field    = 25.5*math.sqrt(2)*fs
#laser_waist         = 12*um
#x_center_laser      = Lx-1.7*c_normalized*laser_fwhm_field
#x_focus_laser       = (x_center_laser+0.1*c_normalized*laser_fwhm_field)
#a0                  = 2.3

## Define a Gaussian bunch with Gaussian temporal envelope
#LaserEnvelopeGaussianAM(
#    a0              = a0,
#    omega           = (2.*math.pi/lambda0*c)/omega0,
#    focus           = [x_focus_laser,0.],
#    waist           = laser_waist,
#    time_envelope   = tgaussian(center=x_center_laser, fwhm=laser_fwhm_field),
#    envelope_solver                = 'explicit_reduced_dispersion',
#    Envelope_boundary_conditions = [ ["reflective"],["PML"] ],
#    Env_pml_sigma_parameters     = [[0.9 ,2     ],[80.0,2]     ,[80.0,2     ]],
#    Env_pml_kappa_parameters     = [[1.00,1.00,2],[1.00,1.00,2],[1.00,1.00,2]],
#    Env_pml_alpha_parameters     = [[0.90,0.90,1],[0.65,0.65,1],[0.65,0.65,1]]
#)
```

Example: decommented Laser block (= Laser ON)

```
########################### Define the laser pulse

# laser parameters
laser_fwhm_field      = 25.5*math.sqrt(2)*fs
laser_waist           = 12*um
x_center_laser        = Lx-1.7*c_normalized*laser_fwhm_field
x_focus_laser         = (x_center_laser+0.1*c_normalized*laser_fwhm_field)
a0                    = 2.3

# Define a Gaussian bunch with Gaussian temporal envelope
LaserEnvelopeGaussianAM(
   a0                = a0,
   omega             = (2.*math.pi/lambda0*c)/omega0,
   focus             = [x_focus_laser,0.],
   waist             = laser_waist,
   time_envelope     = tgaussian(center=x_center_laser, fwhm=laser_fwhm_field), ,
   envelope_solver                = 'explicit_reduced_dispersion',
   Envelope_boundary_conditions = [ ["reflective"],["PML"] ],
   Env_pml_sigma_parameters     = [[0.9 ,2     ],[80.0,2]     ,[80.0,2     ]],|,
   Env_pml_kappa_parameters     = [[1.00,1.00,2],[1.00,1.00,2],[1.00,1.00,2]],|,
   Env_pml_alpha_parameters     = [[0.90,0.90,1],[0.65,0.65,1],[0.65,0.65,1]] |
)
```

Postprocessing for the practical

Included in the code: the Python postprocessing library **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 (command **ipython** in the terminal)
- Copy and paste the happi commands from the handouts
- Adapt them for your purposes
- Use the post processing scripts available (see handouts)

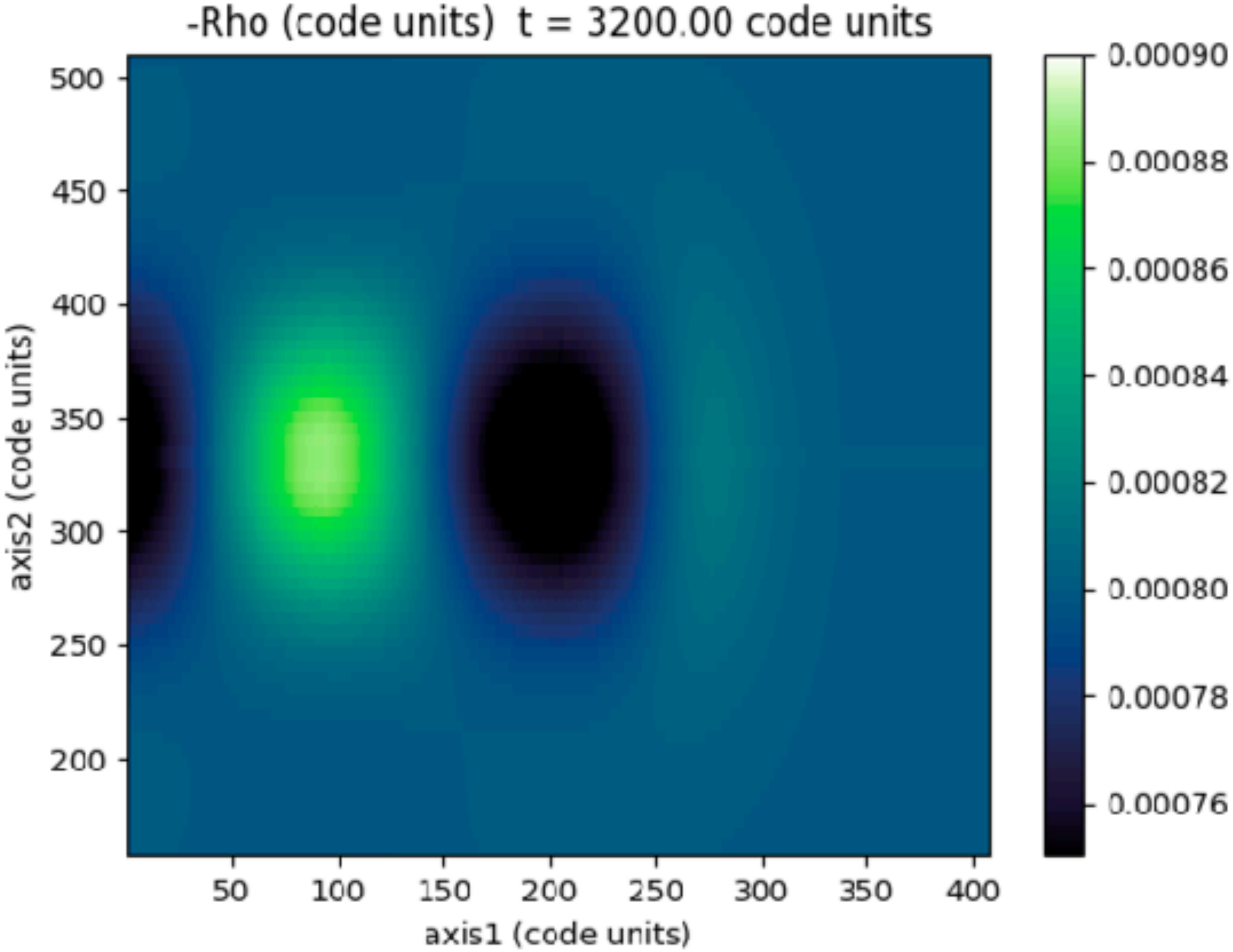
Questions?

Extra slides

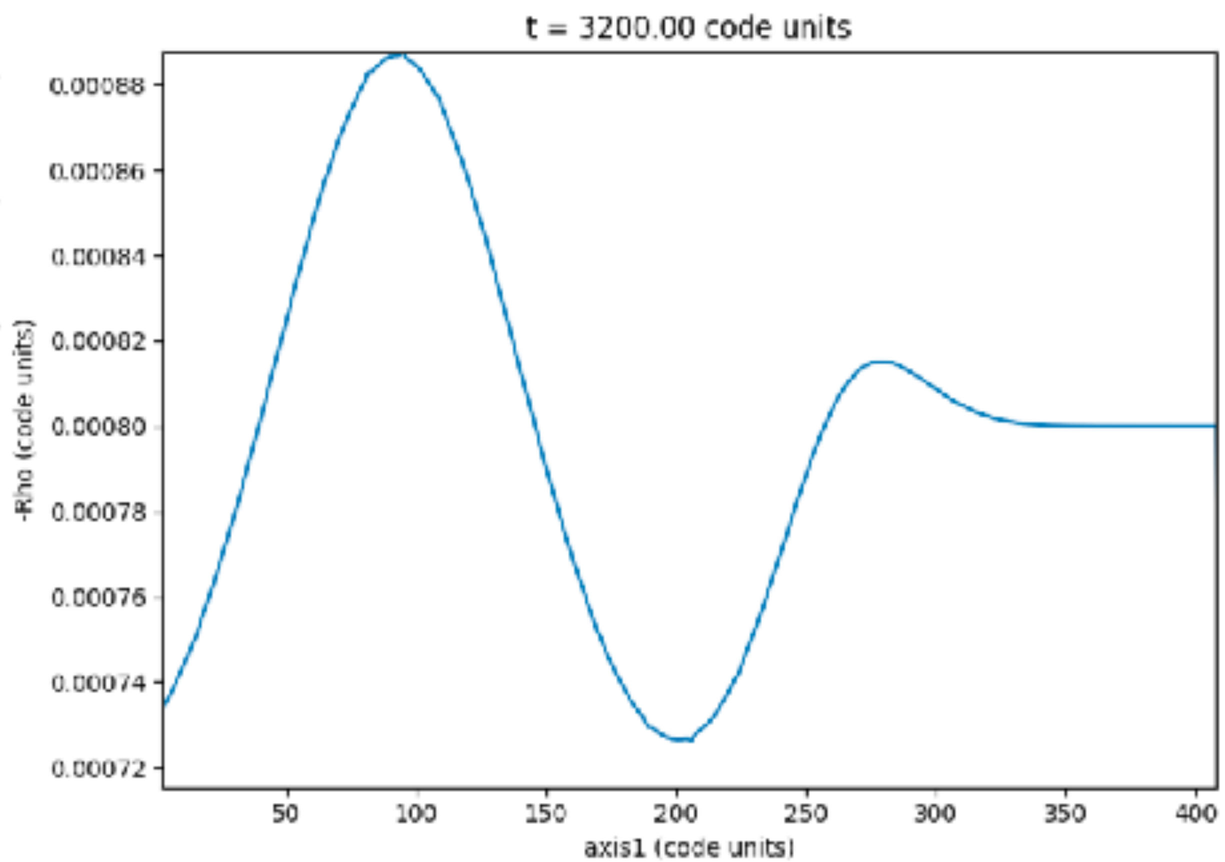
Laser wakefield excitation: linear regime

a_0 = normalized laser peak field = $0.86\ \lambda_0[\mu m]\ (I\ [10^{18}\ W/cm^2])^{1/2}$

$a_0 << 1$: Linear regime
Sinusoidal plasma waves at the plasma frequency



2D charge density

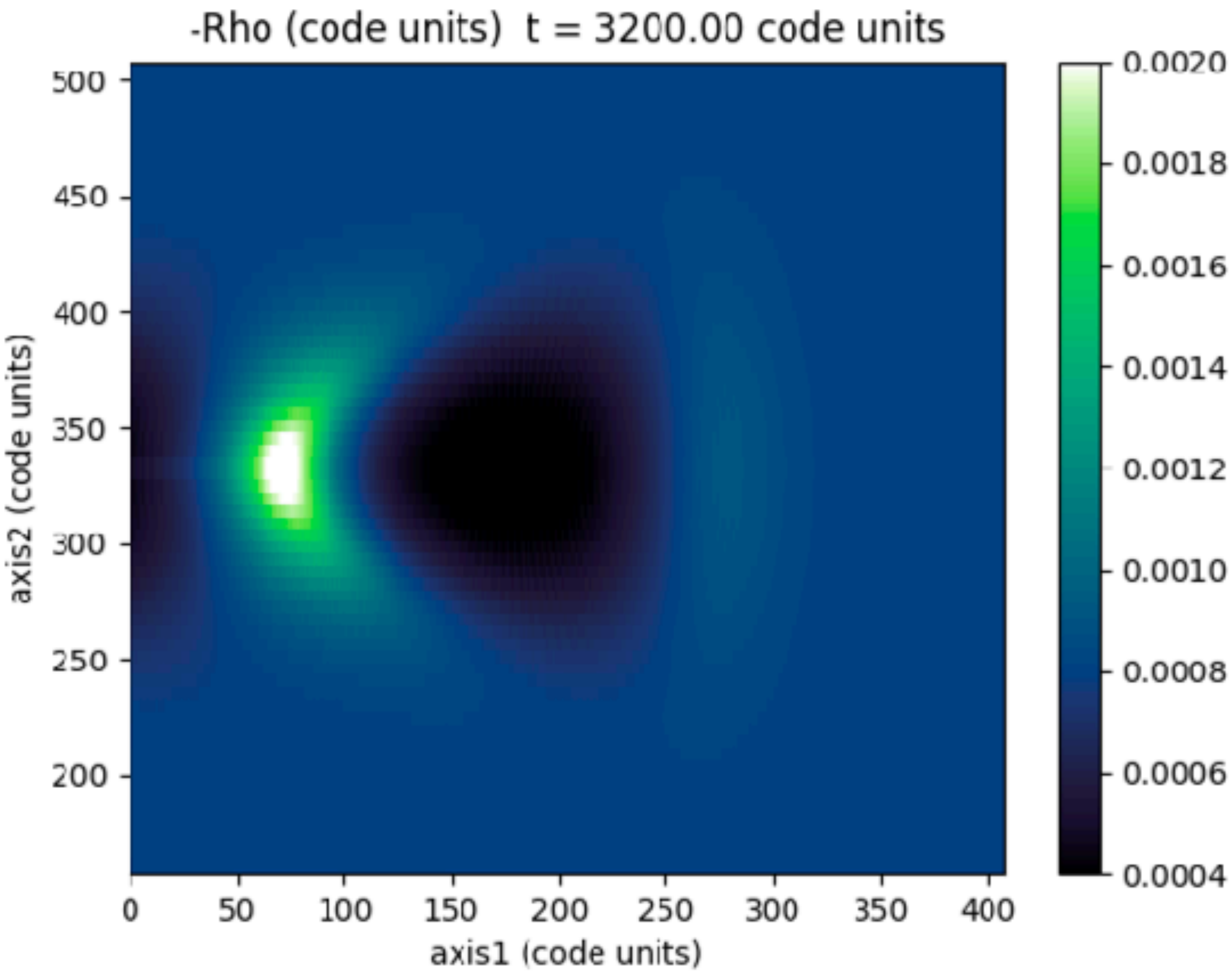


1D charge density
on propagation axis

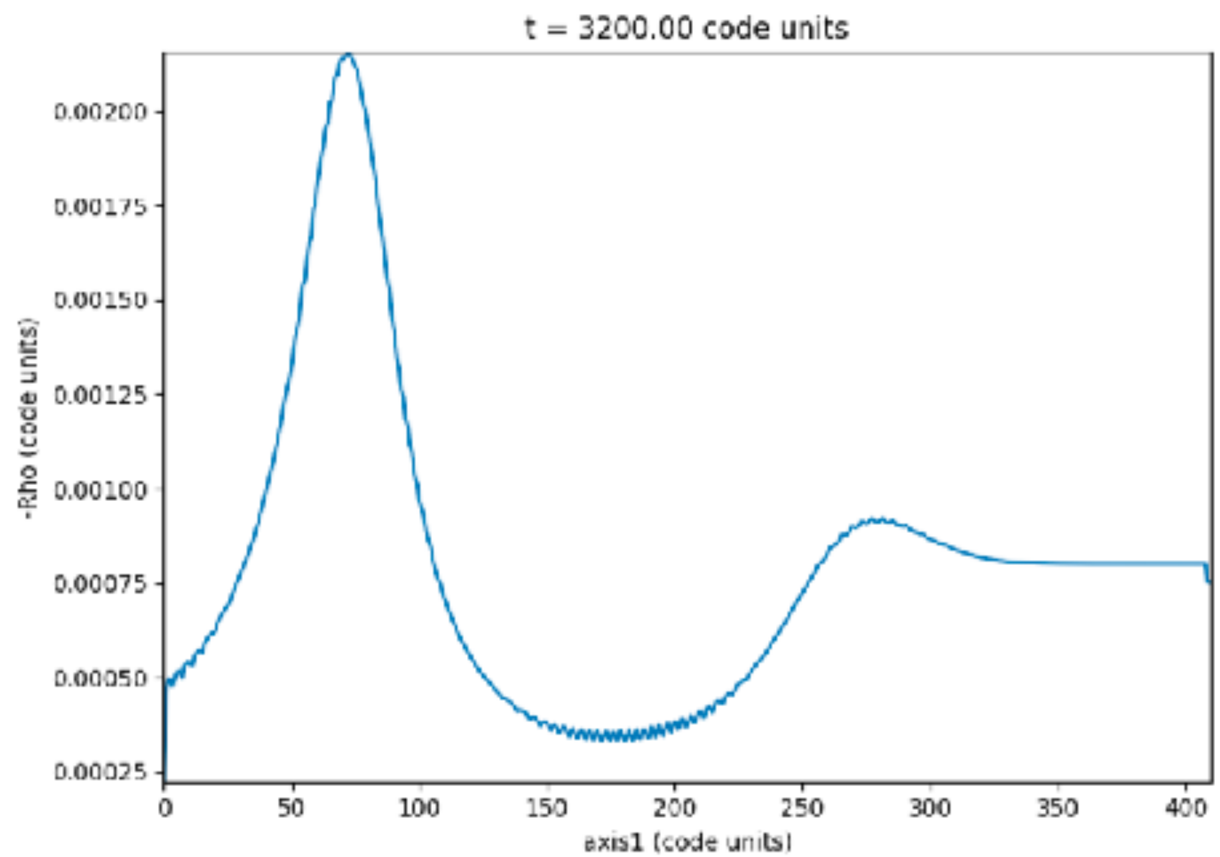
Laser wakefield excitation: weakly linear regime

a_0 = normalized laser peak field = $0.86\ \lambda_0[\mu m]\ (I\ [10^{18}\ W/cm^2])^{1/2}$

$a_0 \lesssim 1$: Weakly nonlinear regime



2D charge density

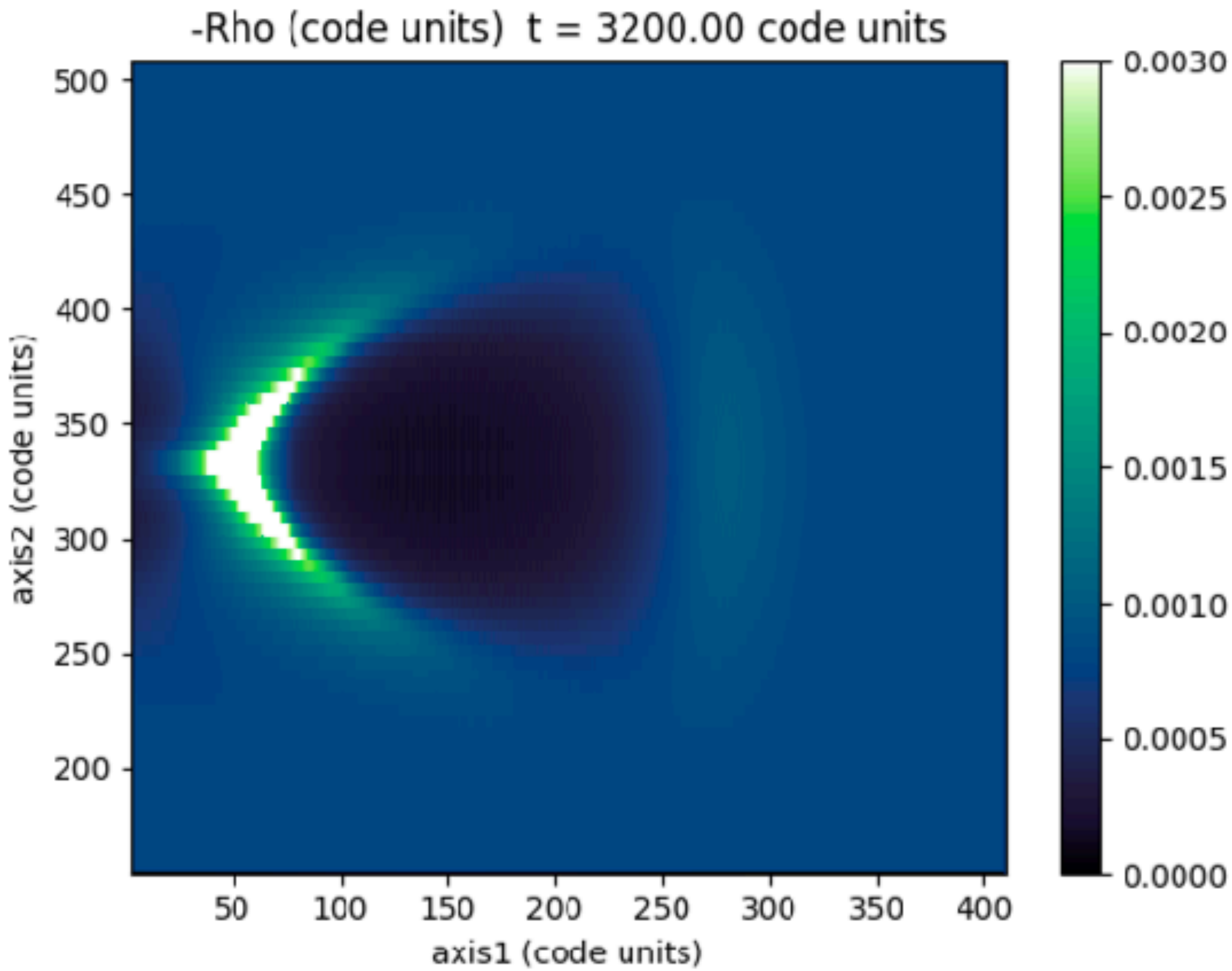


1D charge density
on propagation axis

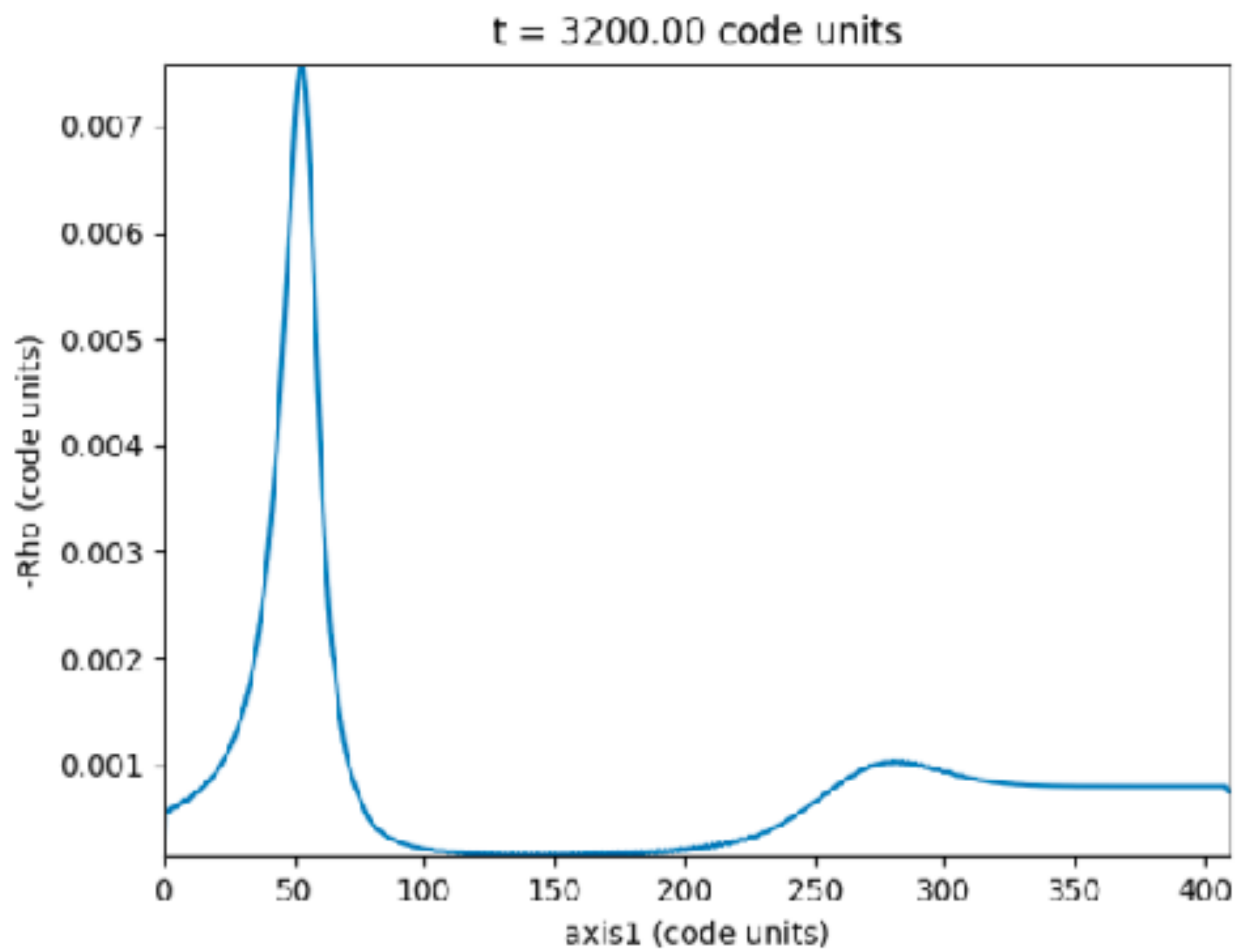
Laser wakefield excitation: nonlinear regime

a_0 = normalized laser peak field = $0.86\ \lambda_0[\mu m]\ (I\ [10^{18}\ W/cm^2])^{1/2}$

$a_0 > 1$: Nonlinear regime
"Bubble"-like waves



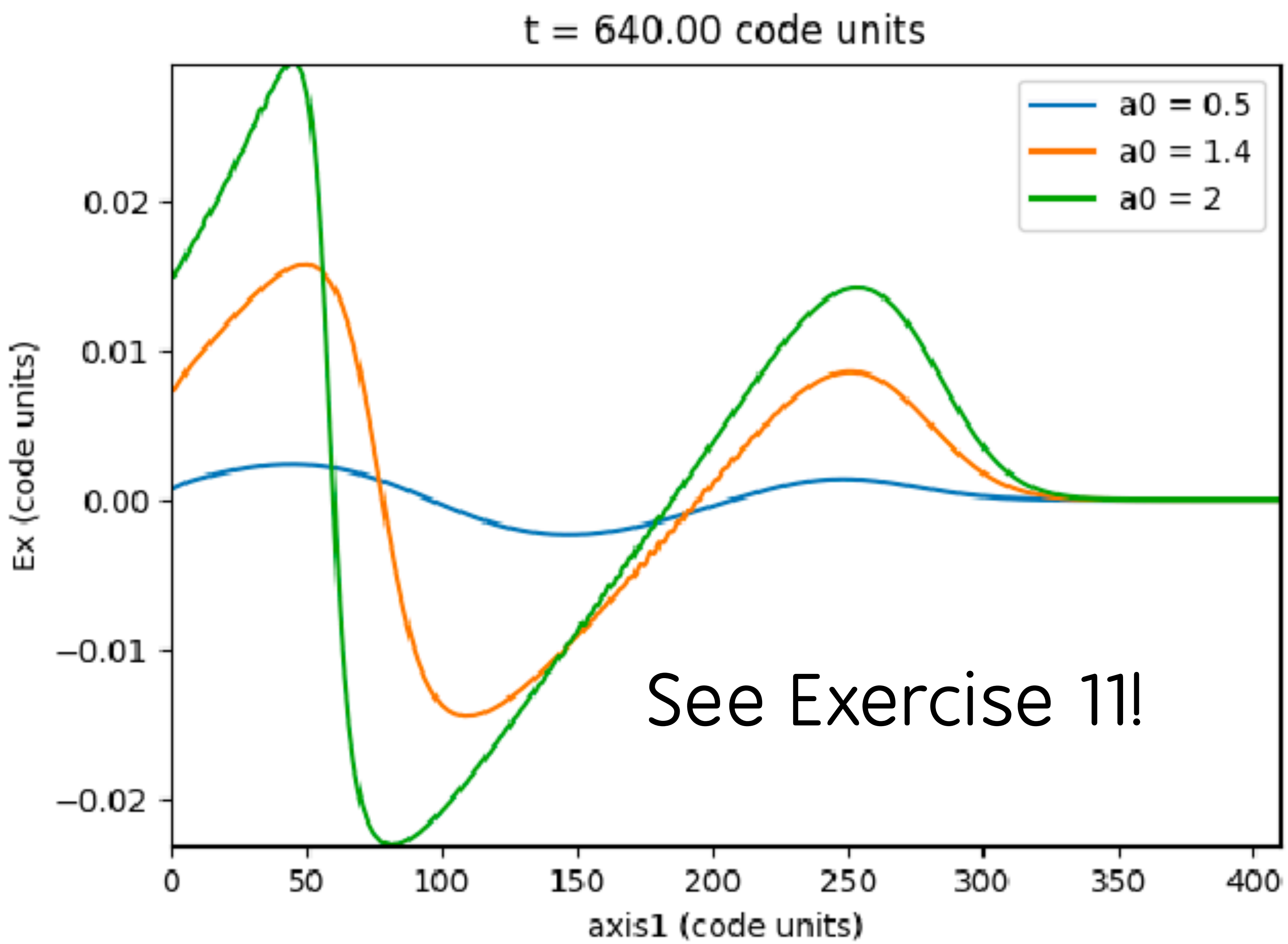
2D charge density



1D charge density
on propagation axis

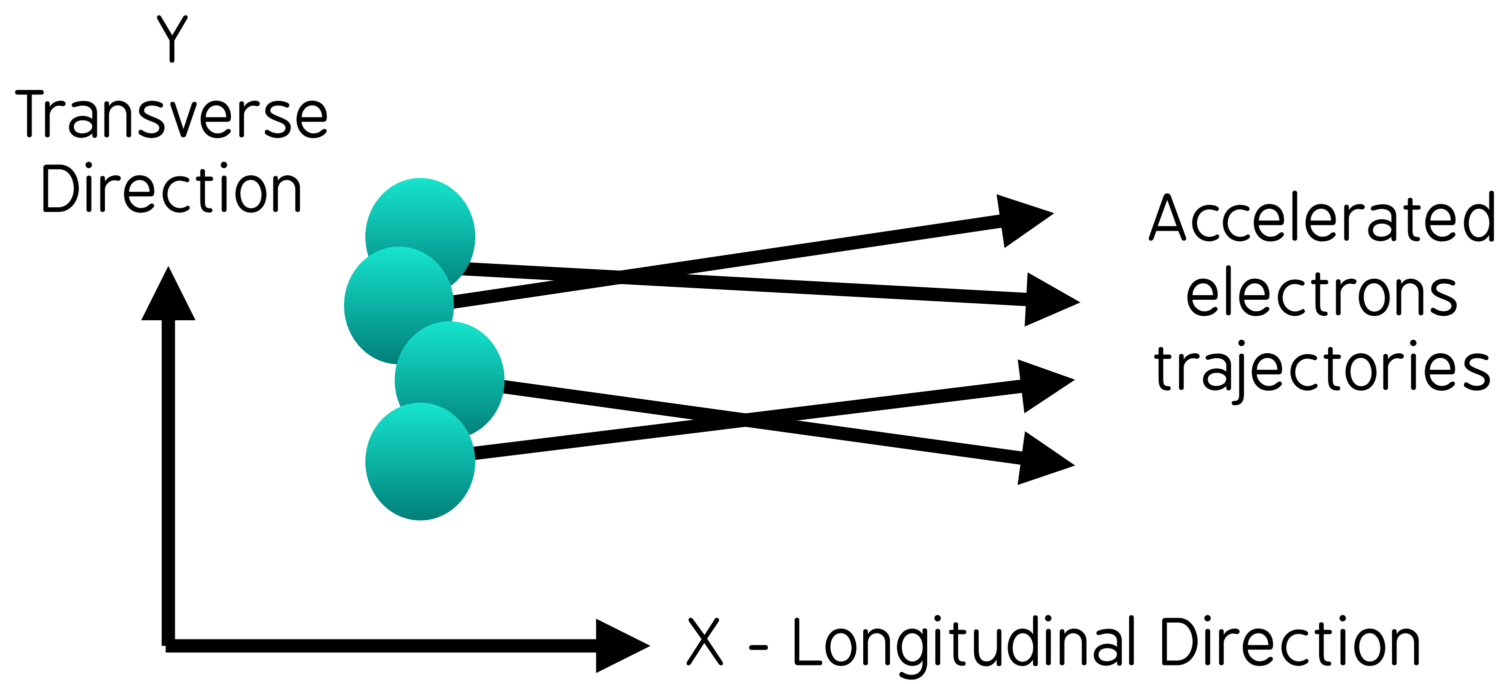
Laser wakefield excitation regimes: Ex on axis

```
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)
```

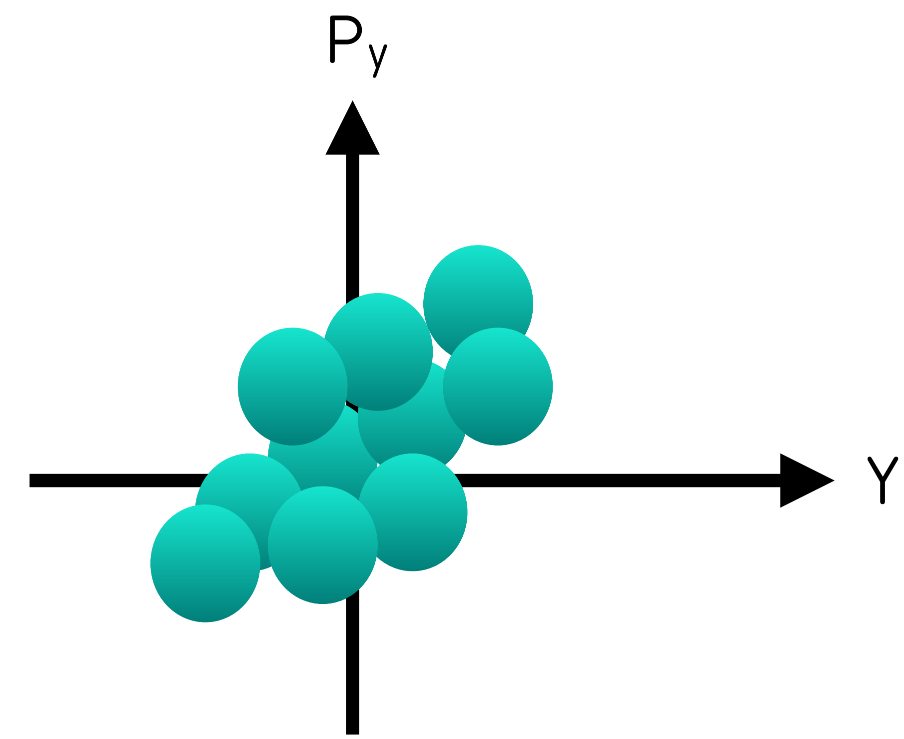


See Exercise 11!

LWFA challenges: lowering the emittance



Bunch distribution
in the transverse phase space



Most applications of accelerated beams require:

- Small transverse size (i.e. small σ_y)
- Small divergence [i.e. small $\sigma_{\arctan(p_y/p_x)}$]

→ **Minimize Transverse Emittance**

Transverse Normalized Emittance

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