

# Simulating plasmas

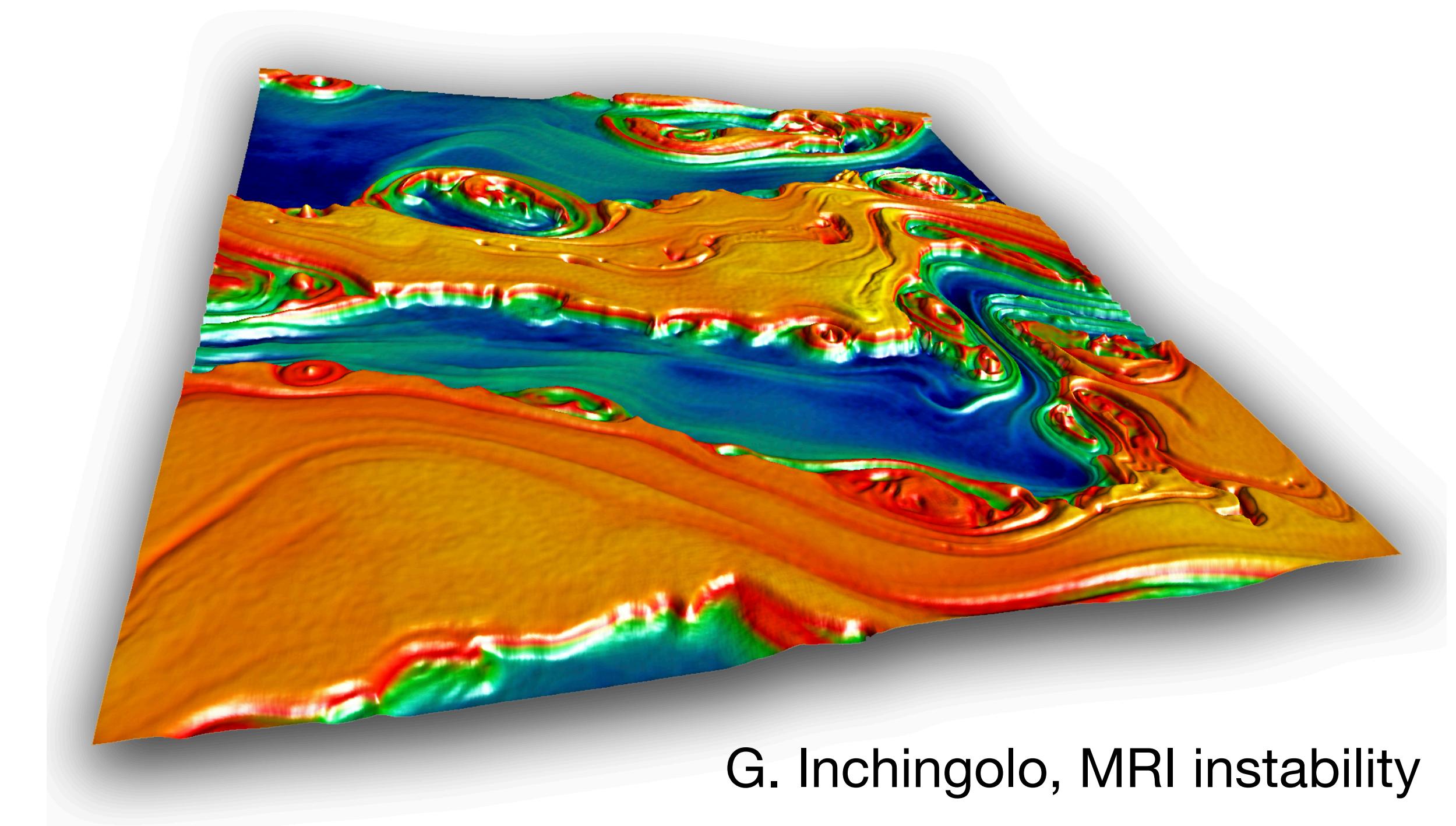
Marija Vranic

GoLP / Instituto de Plasmas e Fusão Nuclear  
Instituto Superior Técnico, Lisbon, Portugal

[epp.tecnico.ulisboa.pt](http://epp.tecnico.ulisboa.pt) || [golp.tecnico.ulisboa.pt](http://golp.tecnico.ulisboa.pt)



**ipfn**  
INSTITUTO DE PLASMAS  
E FUSÃO NUCLEAR

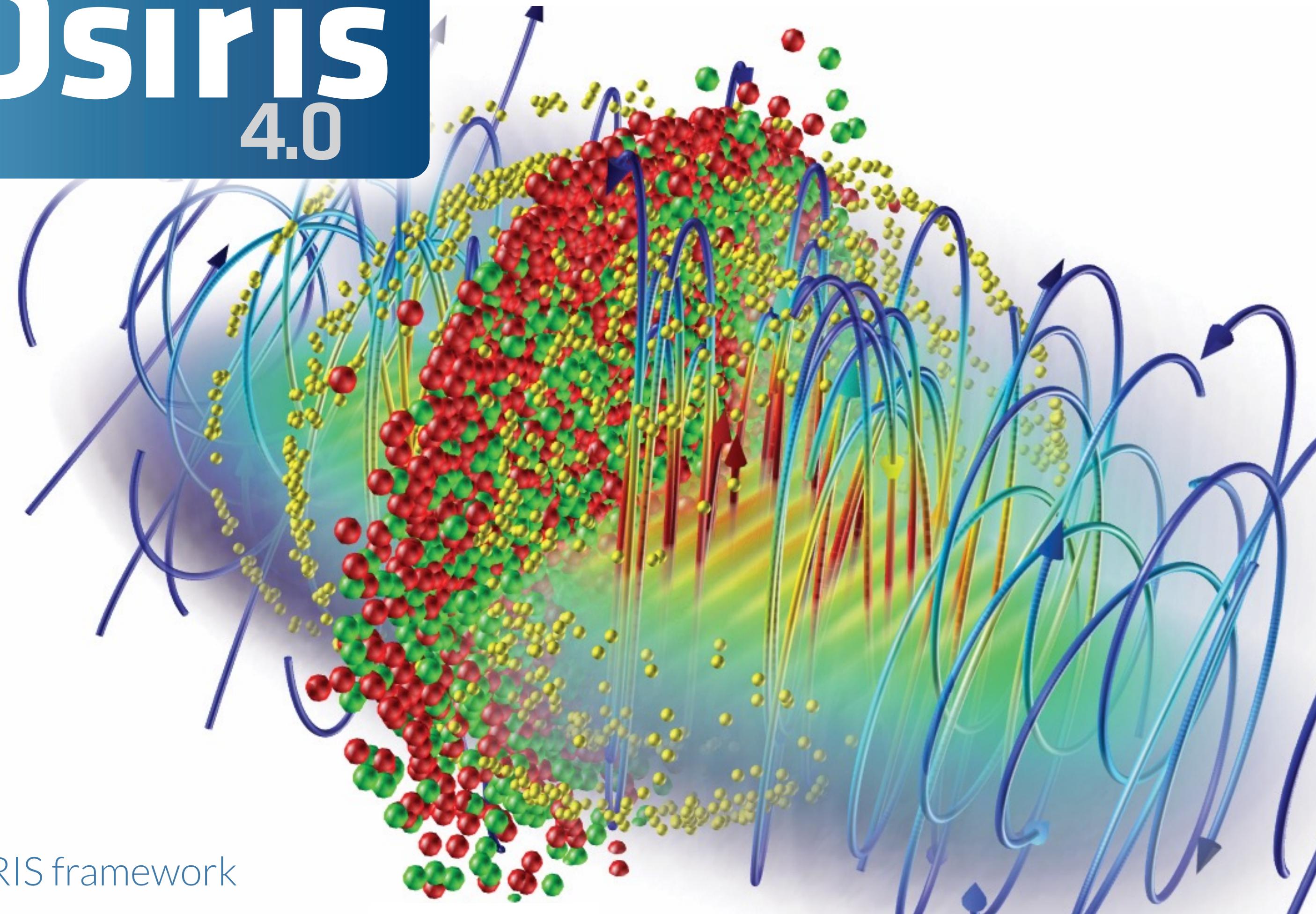


G. Inchingolo, MRI instability



## OSIRIS framework

- Massively Parallel, Fully Relativistic Particle-in-Cell Code
- Support for advanced CPU / GPU architectures
- Extended physics/simulation models
- AI/ML surrogate models and data-driven discovery



# Open-source version available

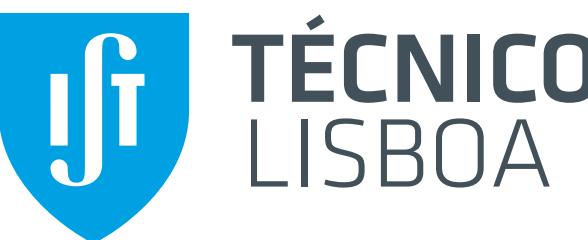
## Open-access model

- 40+ research groups worldwide are using OSIRIS
- 400+ publications in leading scientific journals
- Large developer and user community
- Detailed documentation and sample inputs files available
- Support for education and training

## Using OSIRIS 4.0

- The code can be used freely by research institutions after signing an MoU
- Open-source version at:

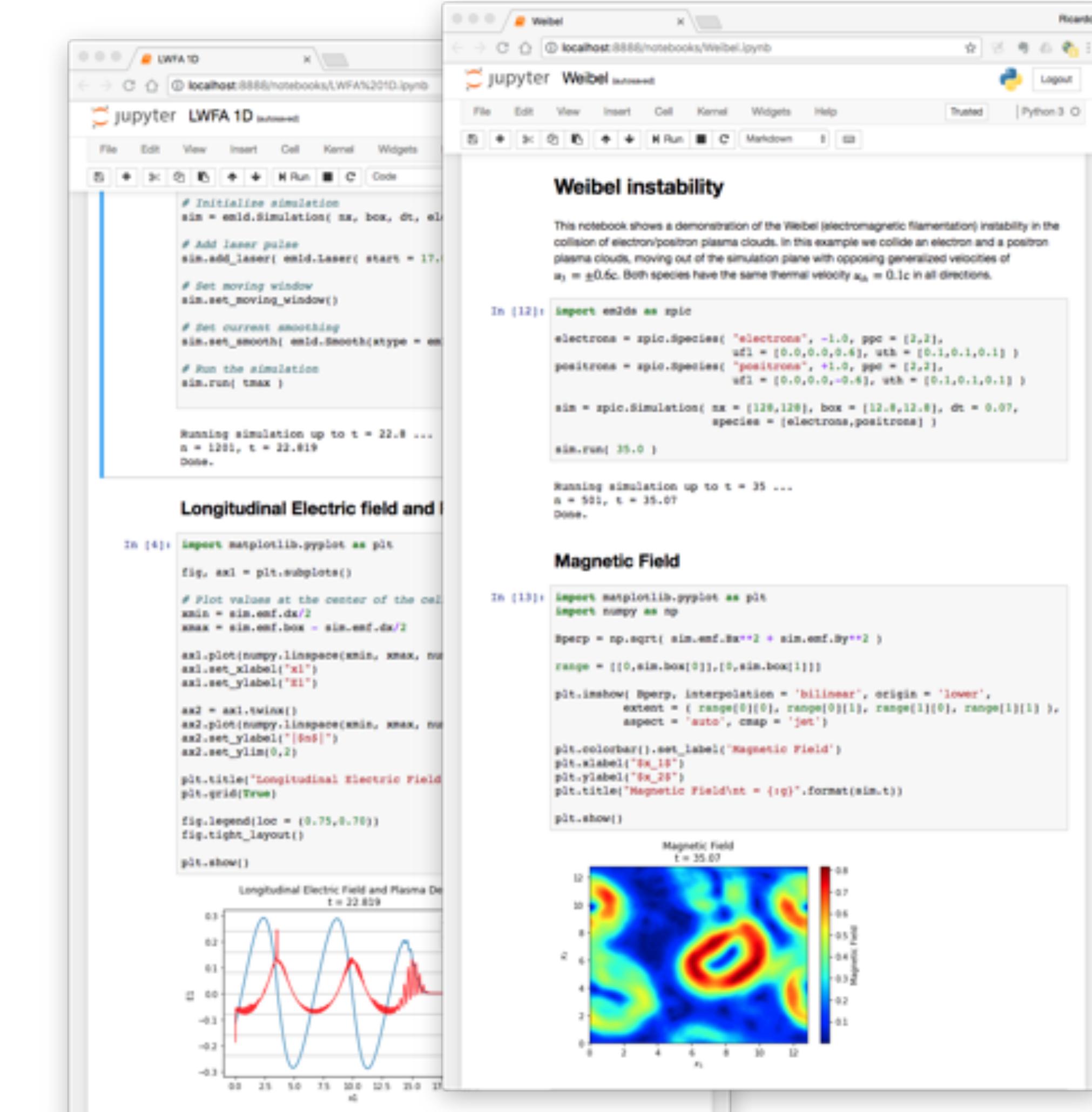
<https://osiris-code.github.io/>



Ricardo Fonseca: ricardo.fonseca@tecnico.ulisboa.pt

# The ZPIC educational code suite

- **ZPIC code suite**
  - Open-source PIC code suit for plasma physics education
  - Fully relativistic 1D and 2D EM-PIC algorithm
  - Electrostatic 1D/2D PIC algorithm
- **Requirements**
  - No external dependencies, requires only C99 compiler
  - Python interface (optional)
- **Jupyter Notebooks**
  - Includes set of Python notebooks with example problems
  - Detailed explanations of code use and physics
- **Also available through Docker**
  - If you just want to run the notebooks you can use a Docker image available on DockerHub: [zamb/zpic](#)



The screenshot shows a Jupyter Notebook interface with several code cells and their corresponding outputs.

**Weibel instability**

```
# Initialize simulation
sim = emid.Simulation( nx, box, dt, el )
# Add laser pulse
sim.add_laser( emid.Laser( start = 17. ) )
# Set moving window
sim.set_moving_window()
# Set current smoothing
sim.set_smooth( emid.Smooth( type = 'on' ) )
# Run the simulation
sim.run( tmax )

Running simulation up to t = 22.0 ...
n = 1281, t = 22.01
Done.
```

**Longitudinal Electric field and I**

```
In [4]: import matplotlib.pyplot as plt
fig, ax1 = plt.subplots()
# Plot values at the center of the cell
xmin = sim.emf.dx/2
xmax = sim.emf.box - sim.emf.dx/2
ax1.plot(numpy.linspace(xmin, xmax, num=100), sim.emf.Ez)
ax1.set_xlabel("x1")
ax1.set_ylabel("Ez")
ax2 = ax1.twinx()
ax2.plot(numpy.linspace(xmin, xmax, num=100), sim.emf.I)
ax2.set_xlabel("x1")
ax2.set_ylabel("I")
plt.title("Longitudinal Electric Field and Plasma Density")
plt.grid(True)
fig.legend(loc = (0.75,0.75))
fig.tight_layout()
plt.show()
```

**Magnetic Field**

```
In [13]: import matplotlib.pyplot as plt
import numpy as np
Bperp = np.sqrt( sim.emf.Bx**2 + sim.emf.By**2 )
range = [(0,sim.box[0]),(0,sim.box[1])]
plt.imshow( Bperp, interpolation = 'bilinear', origin = 'lower',
           extent = ( range[0][0], range[0][1], range[1][0], range[1][1] ),
           aspect = 'auto', cmap = 'jet' )
plt.colorbar().set_label('Magnetic Field')
plt.xlabel("x1_10")
plt.ylabel("x2_10")
plt.title("Magnetic Field\nat = %g" % format(sim.t))
plt.show()
```

Longitudinal Electric Field and Plasma Density  
t = 22.01

Magnetic Field  
t = 35.07

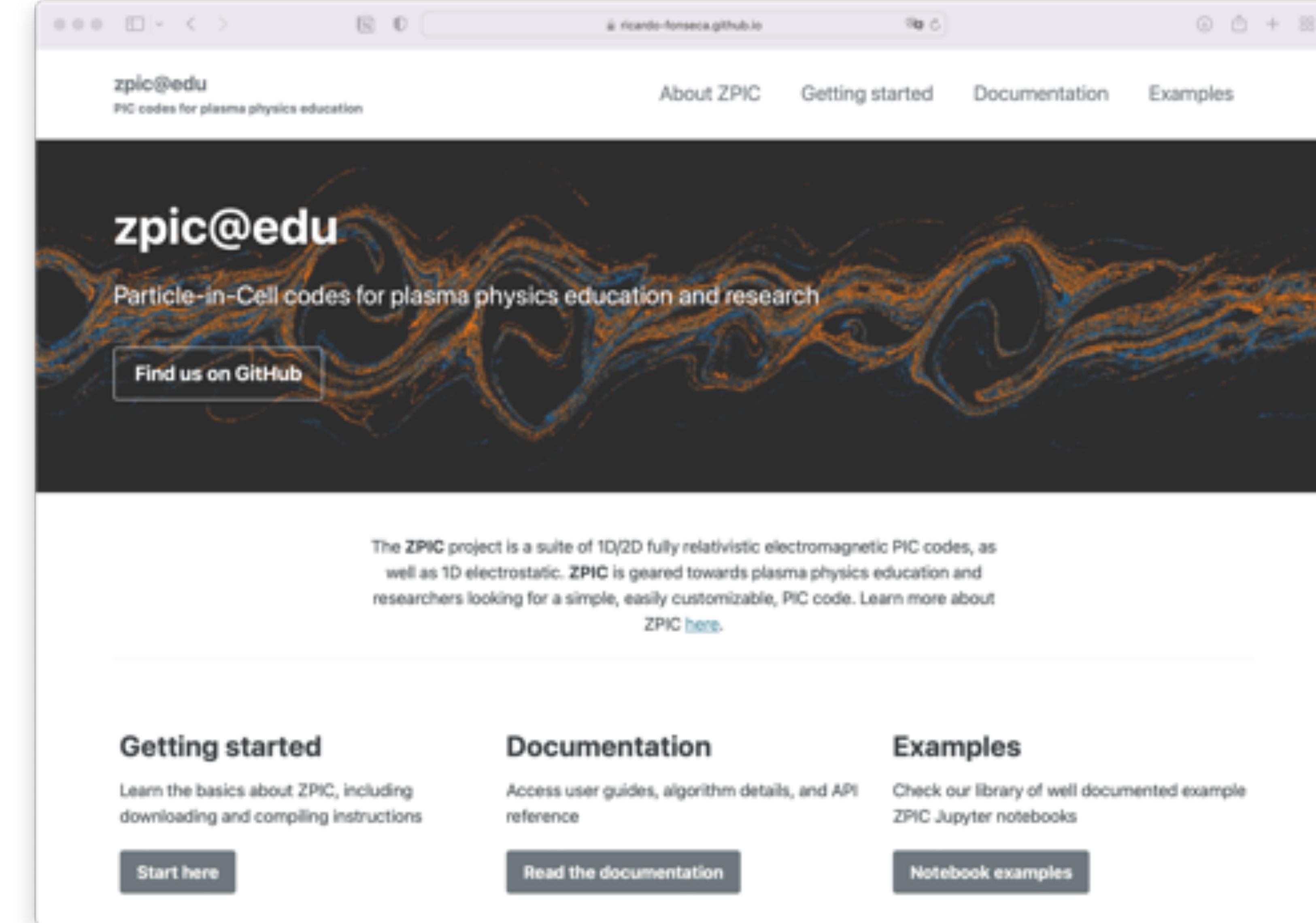
**zpic@edu**



Come find us on GitHub  
[github.com/ricardo-fonseca/zpic](https://github.com/ricardo-fonseca/zpic)

# ZPIC documentation

<https://zpic-plasma.github.io/>



The screenshot shows a web browser displaying the ZPIC documentation page. The header includes the URL 'https://zpic-plasma.github.io/' in the address bar, the page title 'zpic@edu' with the subtitle 'PIC codes for plasma physics education', and a navigation menu with links to 'About ZPIC', 'Getting started', 'Documentation', and 'Examples'. The main content features a large, dark background image of a plasma simulation with blue and orange filaments. Overlaid on this image are the text 'zpic@edu' and 'Particle-in-Cell codes for plasma physics education and research', along with a 'Find us on GitHub' button. Below this, a descriptive paragraph explains the project's purpose: 'The ZPIC project is a suite of 1D/2D fully relativistic electromagnetic PIC codes, as well as 1D electrostatic. ZPIC is geared towards plasma physics education and researchers looking for a simple, easily customizable, PIC code. Learn more about ZPIC [here](#)'. The page is divided into three main sections at the bottom: 'Getting started', 'Documentation', and 'Examples', each with a brief description and a call-to-action button.

**zpic@edu**  
PIC codes for plasma physics education

About ZPIC Getting started Documentation Examples

**zpic@edu**

Particle-in-Cell codes for plasma physics education and research

Find us on GitHub

The ZPIC project is a suite of 1D/2D fully relativistic electromagnetic PIC codes, as well as 1D electrostatic. ZPIC is geared towards plasma physics education and researchers looking for a simple, easily customizable, PIC code. Learn more about ZPIC [here](#).

**Getting started**

Learn the basics about ZPIC, including downloading and compiling instructions

[Start here](#)

**Documentation**

Access user guides, algorithm details, and API reference

[Read the documentation](#)

**Examples**

Check our library of well documented example ZPIC Jupyter notebooks

[Notebook examples](#)

# Examples



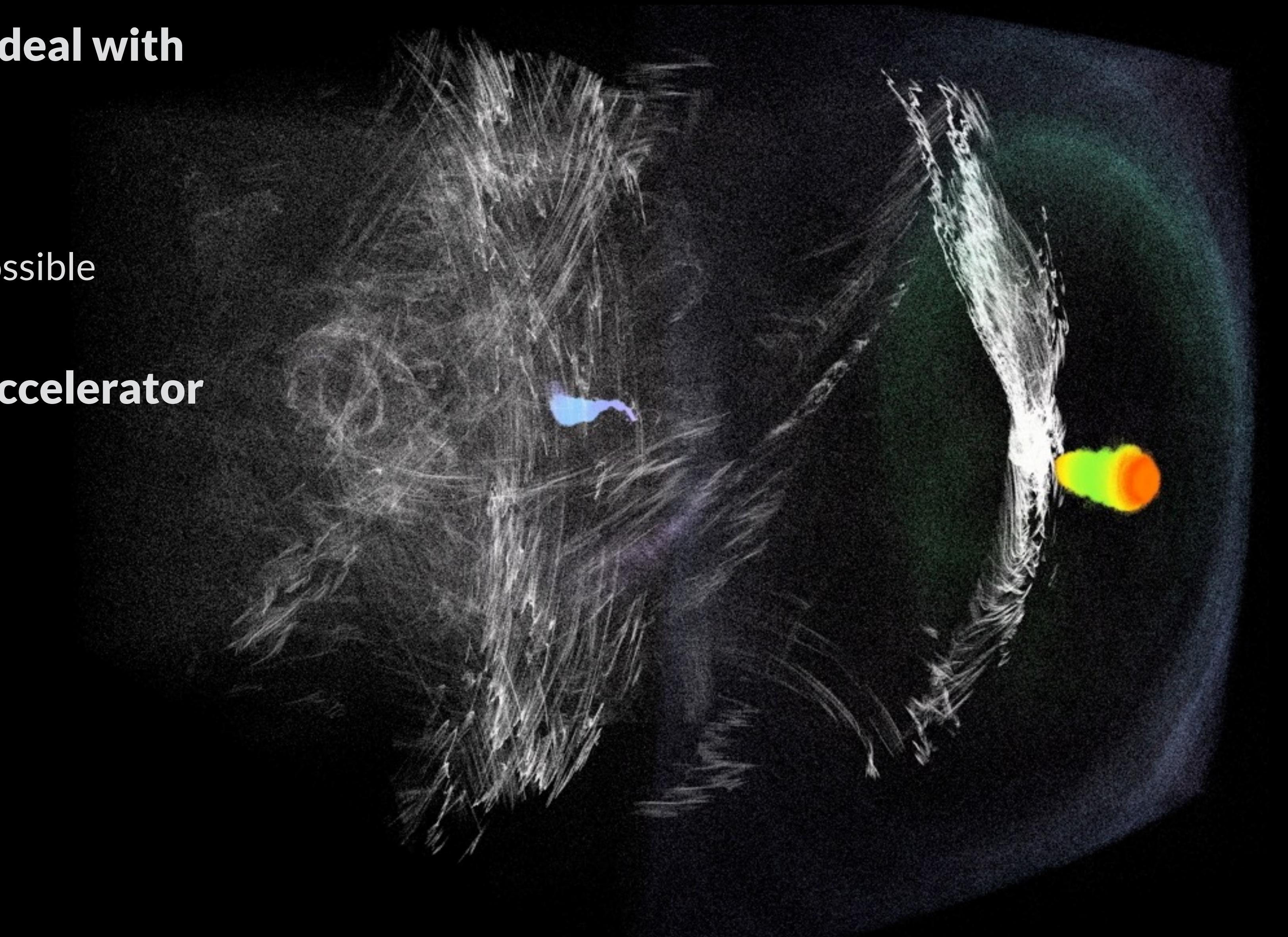
# Large Scale Numerical Experiments

- **Computational Physics needs to deal with increasingly complex systems**

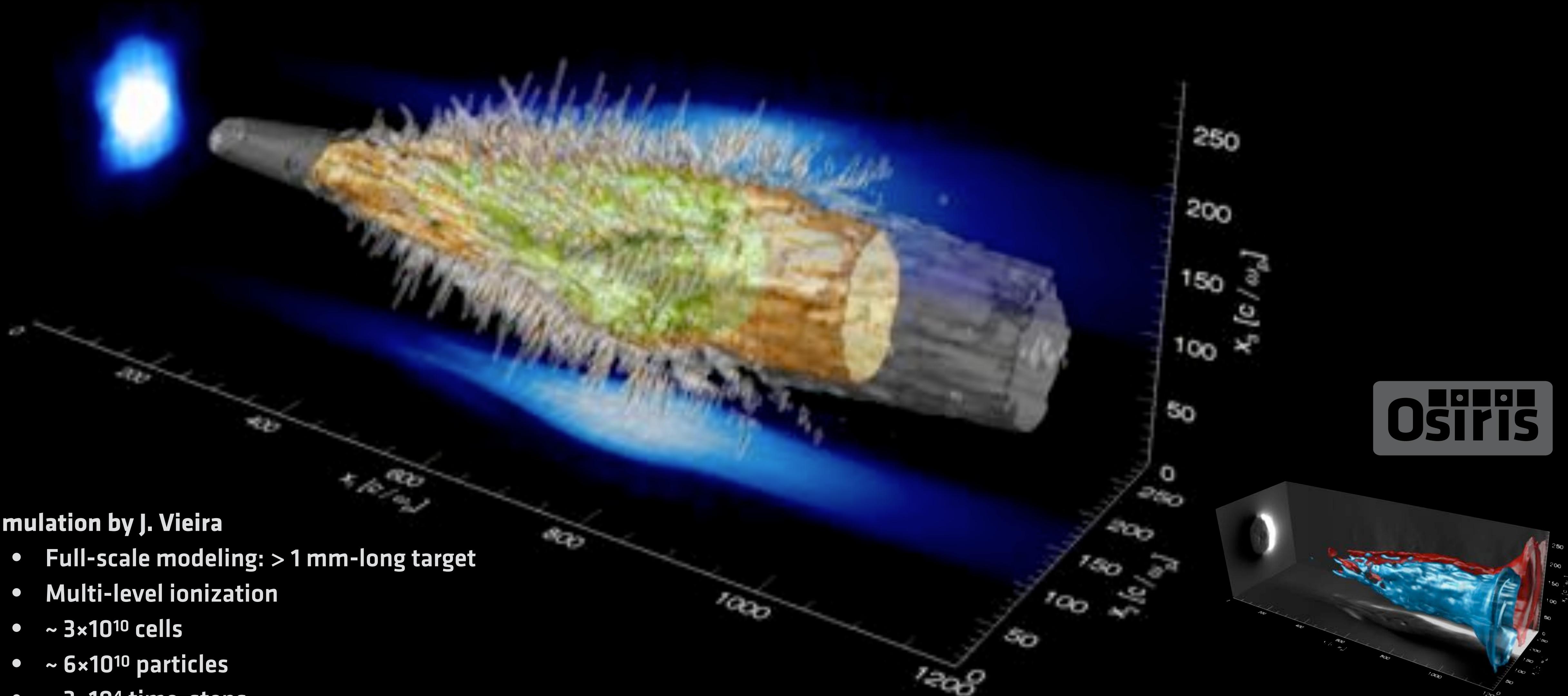
- Highly nonlinear processes
- Multi-scale phenomena
- Purely theoretical descriptions impossible

- **Example: 10 GeV Laser-plasma accelerator**

- Laser  $\lambda_0 \sim 1 \mu\text{m}$
- Propagation length  $L \sim 0.5 \text{ m}$
- $\sim 10^9$  grid cells
- $\sim 10^{10}$  particles
- Iterations  $\sim 10^6 - 10^7$
- Memory  $\sim 1 - 10 \text{ TB}$
- Operations  $\sim 10^{18} - 10^{19}$
- **Run-time on single core  $\sim 4 \text{ years!}$**



# Full scale 3D LWFA modeling

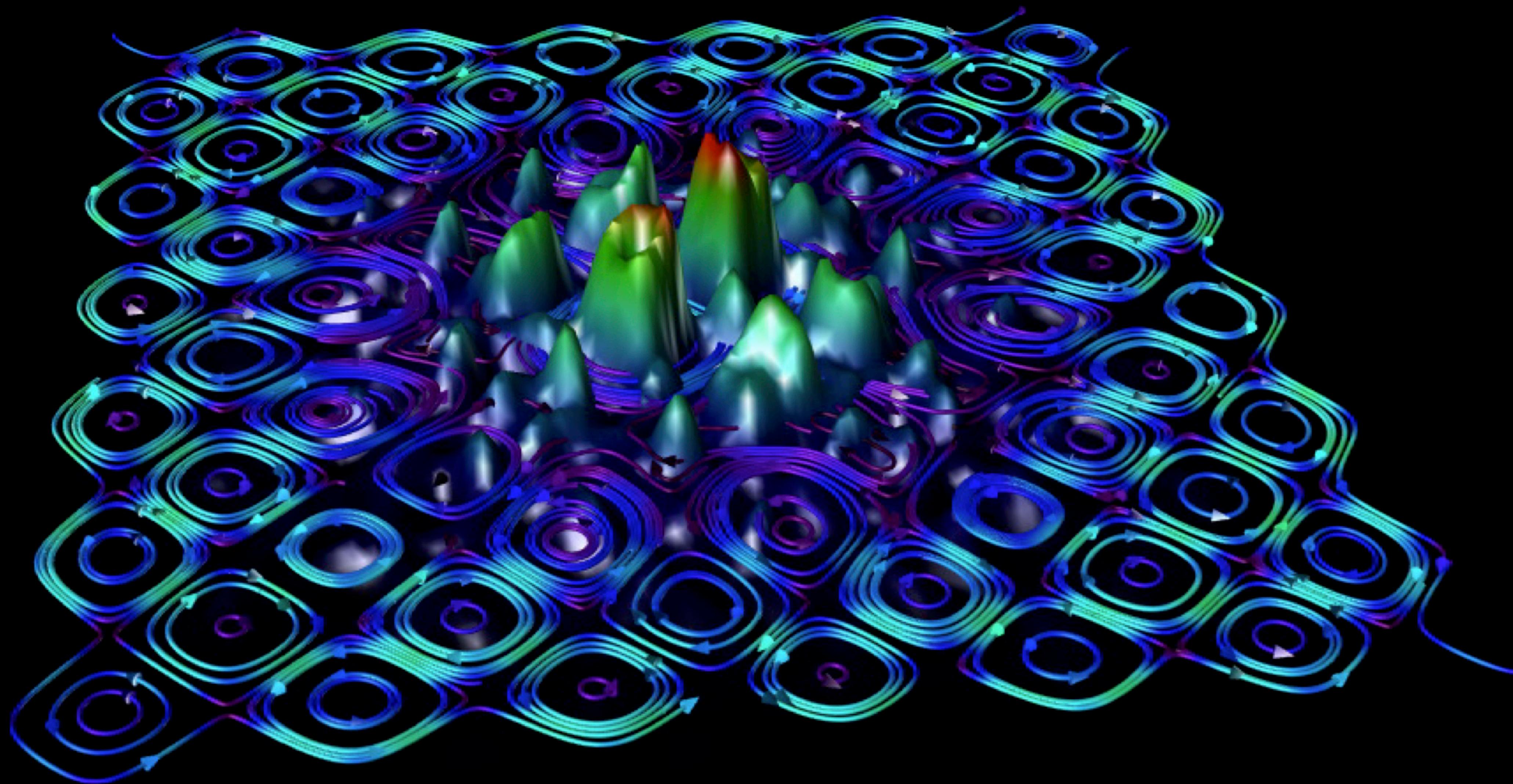


## Simulation by J. Vieira

- Full-scale modeling: > 1 mm-long target
- Multi-level ionization
- $\sim 3 \times 10^{10}$  cells
- $\sim 6 \times 10^{10}$  particles
- $\sim 3 \times 10^4$  time-steps
- $\sim 0.4$  million core h ( $\sim 16$  k€)

**Osiris**

# Plasma in extreme conditions - kinetic PIC + Monte Carlo



# The PIC algorithm



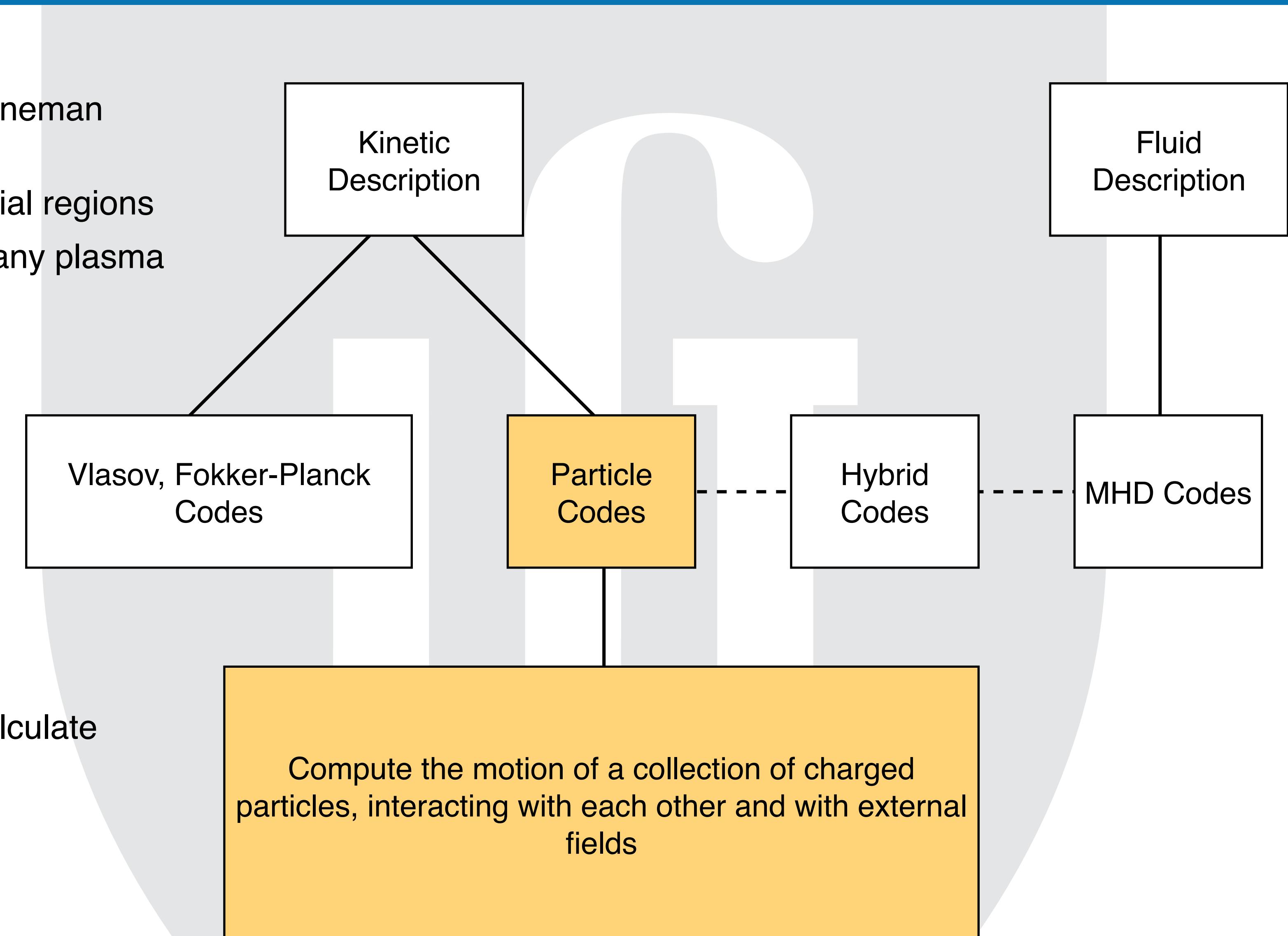
# Overview of Plasma Simulation Algorithms

- **Plasma Simulations Using Particles**

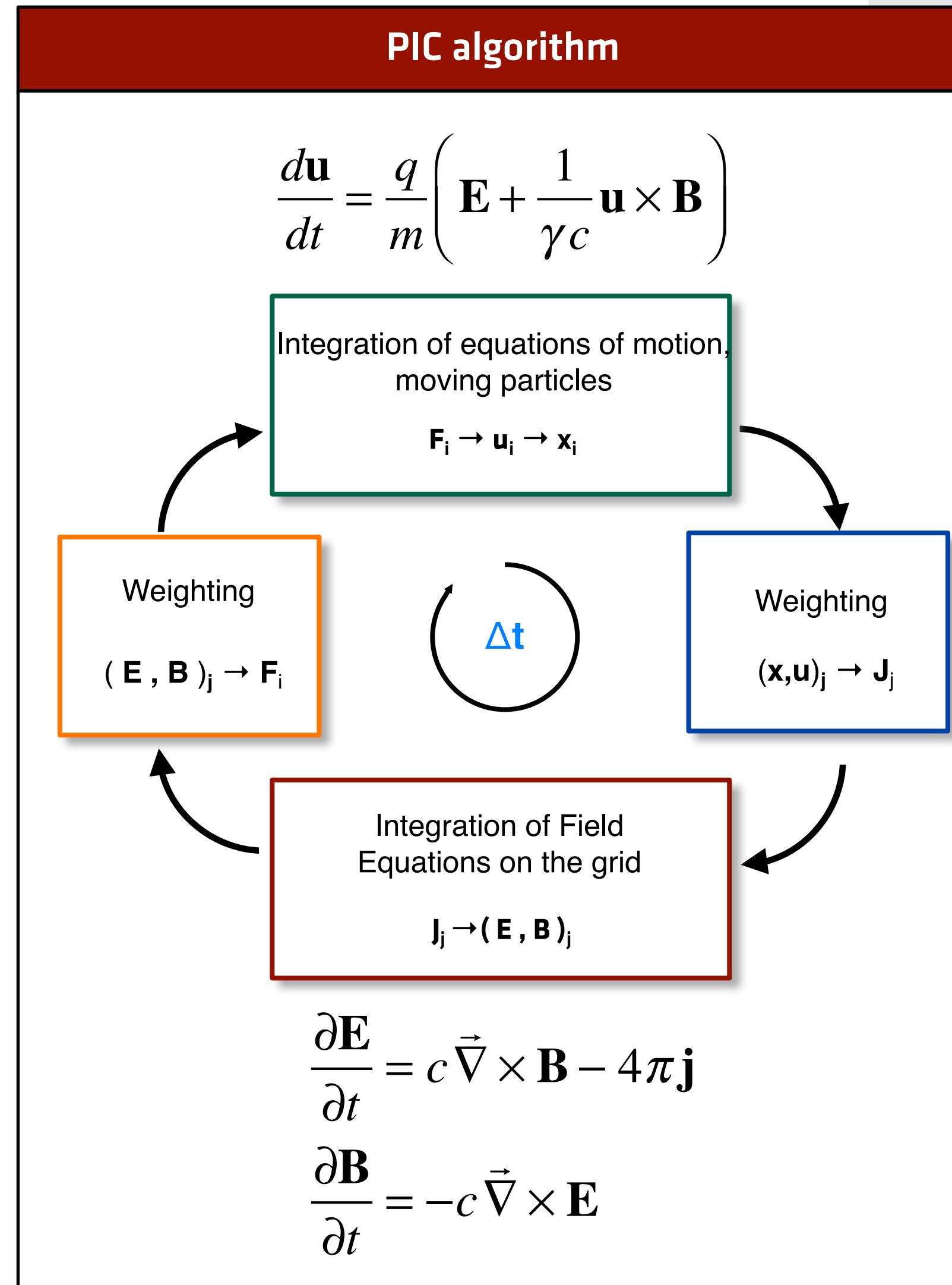
- Pioneered by John Dawson and Oscar Buneman circa 1960
- Use macro particles to simulate large spatial regions
  - 1 simulation particle corresponds to many plasma particles
- Particle-Particle simulations
  - Computations go with  $O(N_p^2)$
  - Computationally very demanding

- **Particle-In-Cell algorithms**

- Interact particles through fields
- Discretize fields on grids
- Interpolate fields at particle positions to calculate forces
- Deposit particle charge/current on a grid
- Particle-Mesh algorithm
  - Computations go with  $O(N_p)$
  - Still computationally heavy but much more tractable



# The particle-in-cell (PIC) Algorithm

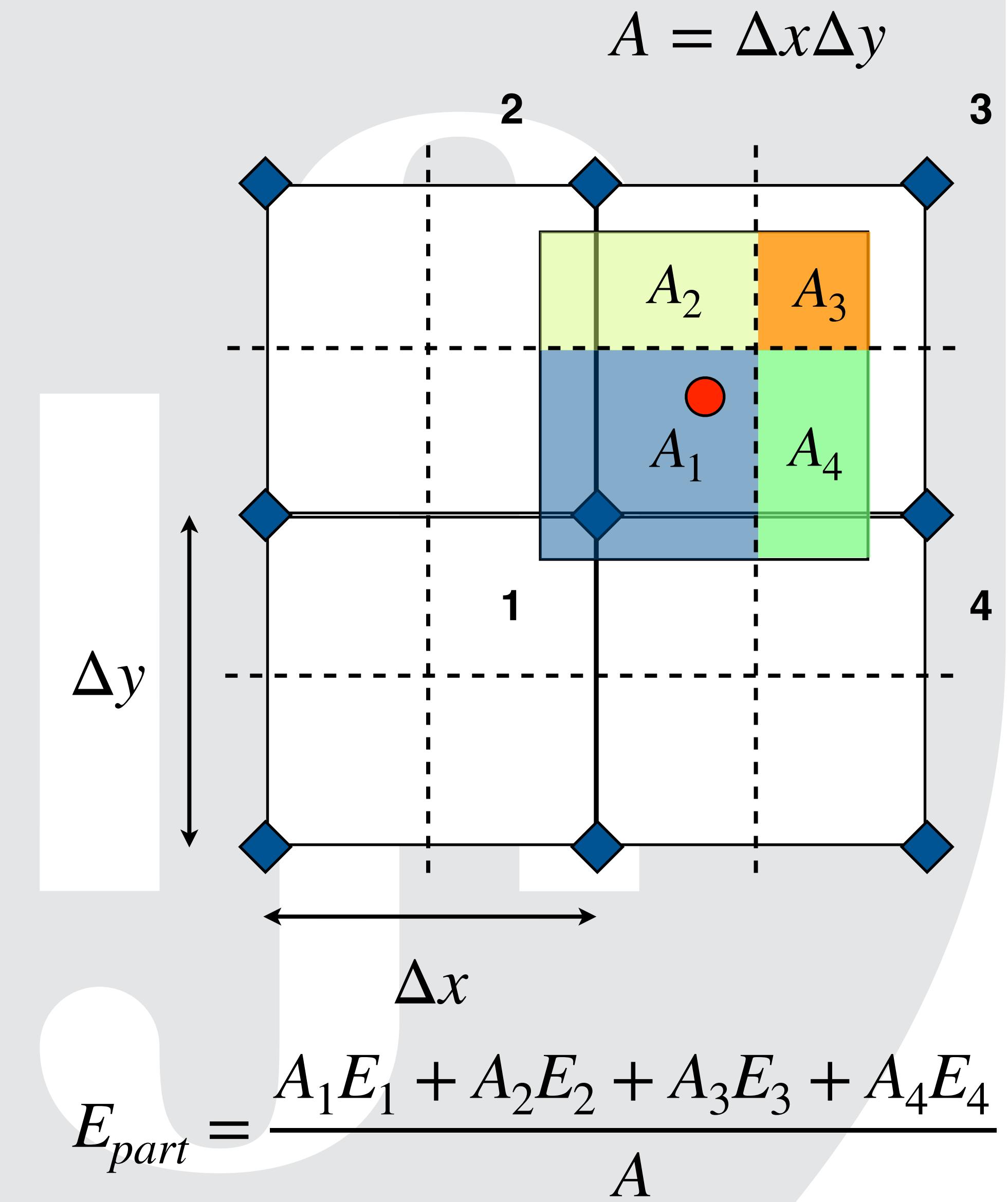


- **Fully Relativistic, Electromagnetic Particle-In-Cell algorithm**
  - Discretize Electric and Magnetic fields on a grid
  - Cell size must resolve shortest relevant lengths in the simulation
    - Typically the laser wavelength or the plasma skin depth
  - Represent plasma particles with simulation macro-particles
    - Free to move in entire  $nD-3V$  phase-space
    - Each macro-particle represents several plasma particles
  - Must have enough particles per cell to properly resolve velocity distributions
- **Fields and particles don't exist in the same simulation topology**
  - Field quantities are limited to grid points
  - Field interpolation connects fields → particles
  - Current deposition connects particles → fields
- **Four major steps**
  - Field interpolation
  - Particle advance
  - Current deposition
  - Field advance

zpic@edu

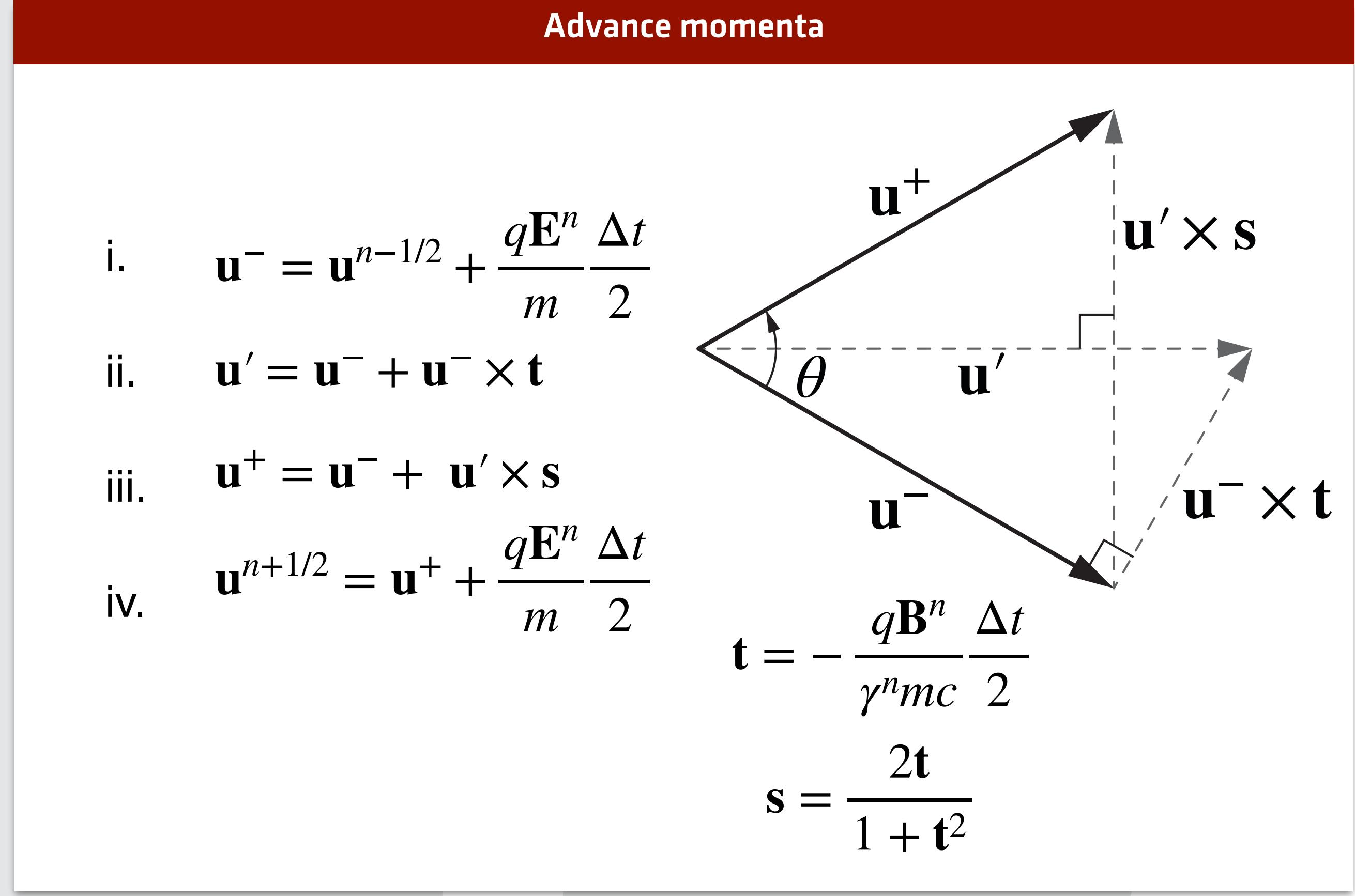
# Interpolating the fields

- **Particles are free to move to any position**
  - Field are discretized on a grid
  - Field values at particle positions are required to advance particle momenta
- **Interpolate fields at particle positions**
  - ZPIC uses linear interpolation
  - In 2D this can be viewed as area weighting
  - The interpolating scheme must be consistent with charge / current deposition
- **Momentum conserving algorithm**
  - Avoids self-forces
  - $dp/dt = 0$  for single particle



# Pushing the particles

- Advance generalized velocity and position of individual particles
  - ZPIC is a fully relativistic code so we work with  $\mathbf{u} = \gamma \beta$  instead  $\mathbf{v}$ .
  - We use a leap-frog scheme to integrate particle motion:
    - Positions ( $\mathbf{x}$ ) are defined at integral time-steps  $t^n$
    - Velocities ( $\mathbf{u}$ ) are defined at half time-steps  $t^{n+1/2}$
  - Second-order accuracy in time
- Velocities are integrated using a relativistic Boris pusher
  - Separate  $\mathbf{E}$  and  $\mathbf{B}$  contributions
    - Accelerate with  $\frac{1}{2}$  electric impulse
    - Full magnetic field rotation
    - Add remaining  $\frac{1}{2}$  electric impulse
  - Fully relativistic, second order time accurate
  - By construction, no work from  $\mathbf{B}$  field
- Position advance is straightforward
  - ZPIC stores cell index and position inside cell



**Advance positions**

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{\mathbf{u}^{n+1/2}}{\gamma^{n+1/2}} \Delta t$$

# Depositing the current

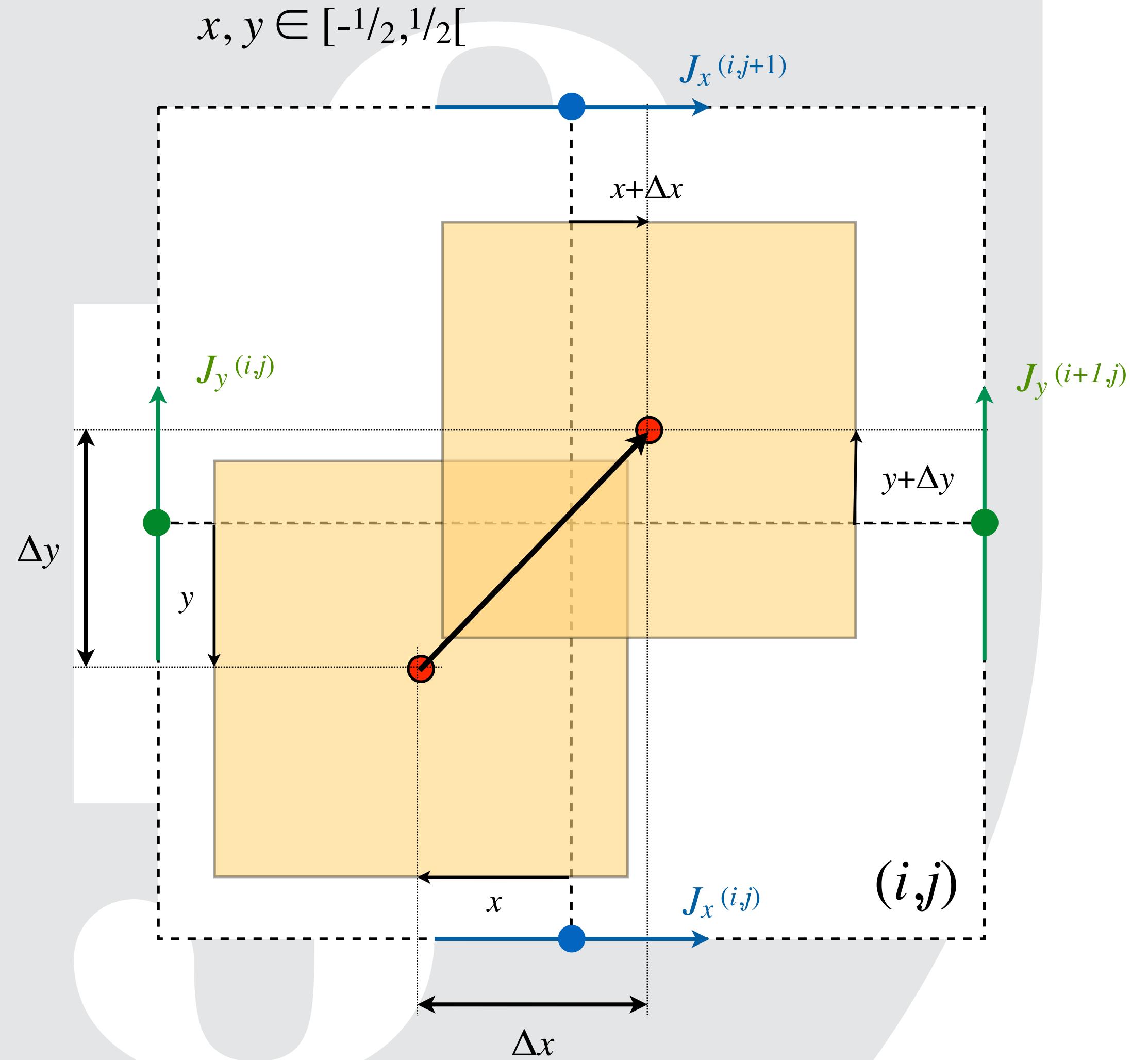
- **Connects particle motion to field equations**
  - Current deposition must satisfy continuity equation:

$$\frac{\partial \rho}{\partial t} = - \nabla' \cdot \mathbf{j}$$

- The operator  $\nabla'$  corresponds to the finite difference approximation
- Simply depositing  $\rho \mathbf{v}$  does not conserve charge
- Critical to guarantee the solutions to Maxwell's equations are self-consistent

- **Exact charge conserving current deposition scheme**

- Developed by Villaseñor and Buneman for linear interpolation
- Looks at particle motion, not velocity
- Limited to motion inside single cell
  - If particles cross cell boundary, motion is split into segments that don't cross boundaries



# Advancing the EM fields

- **EM Fields are advanced in time using Maxwell's equations using the deposited current as source terms**

- Rearrange Ampère's and Faraday's laws:

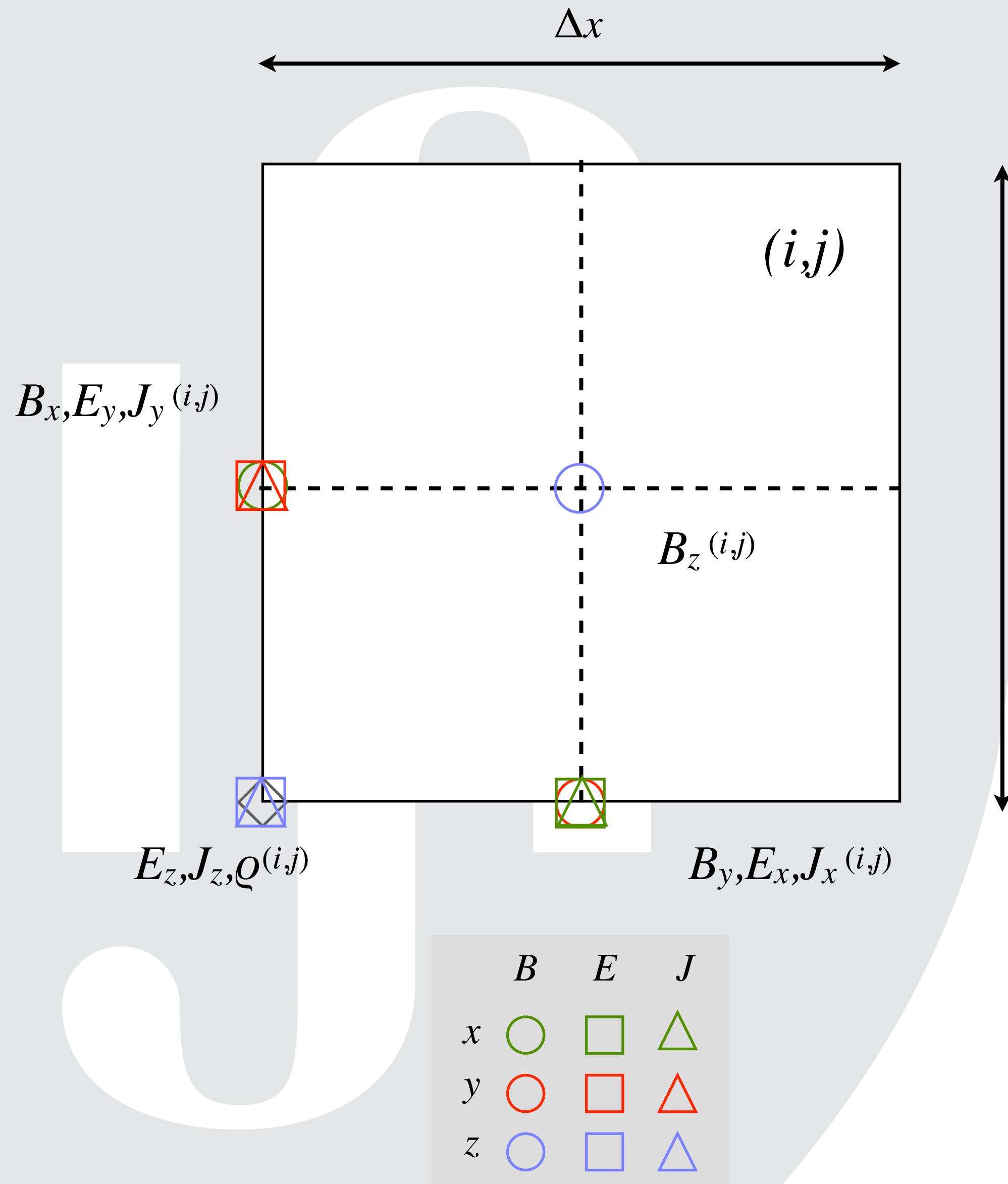
$$\frac{\partial \mathbf{E}}{\partial t} = \nabla' \times \mathbf{B} - \mathbf{j}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla' \times \mathbf{E}$$

- Discretize temporal and spatial derivatives using finite differences

- **Careful time and spacial centering of quantities leads to 2<sup>nd</sup> order accuracy**

- ZPIC uses the Finite Difference Time Domain (FDTD) algorithm
  - Fields are staggered in time for 2<sup>nd</sup> order accuracy
    - $\mathbf{E}$  is defined at times  $t^n$
    - $\mathbf{B}$  and  $\mathbf{j}$  are defined at times  $t^{n+1/2}$
    - $\mathbf{B}$  is later time centered for use in particle advance
  - And also in space:
    - Spatial derivatives are also 2<sup>nd</sup> order accurate



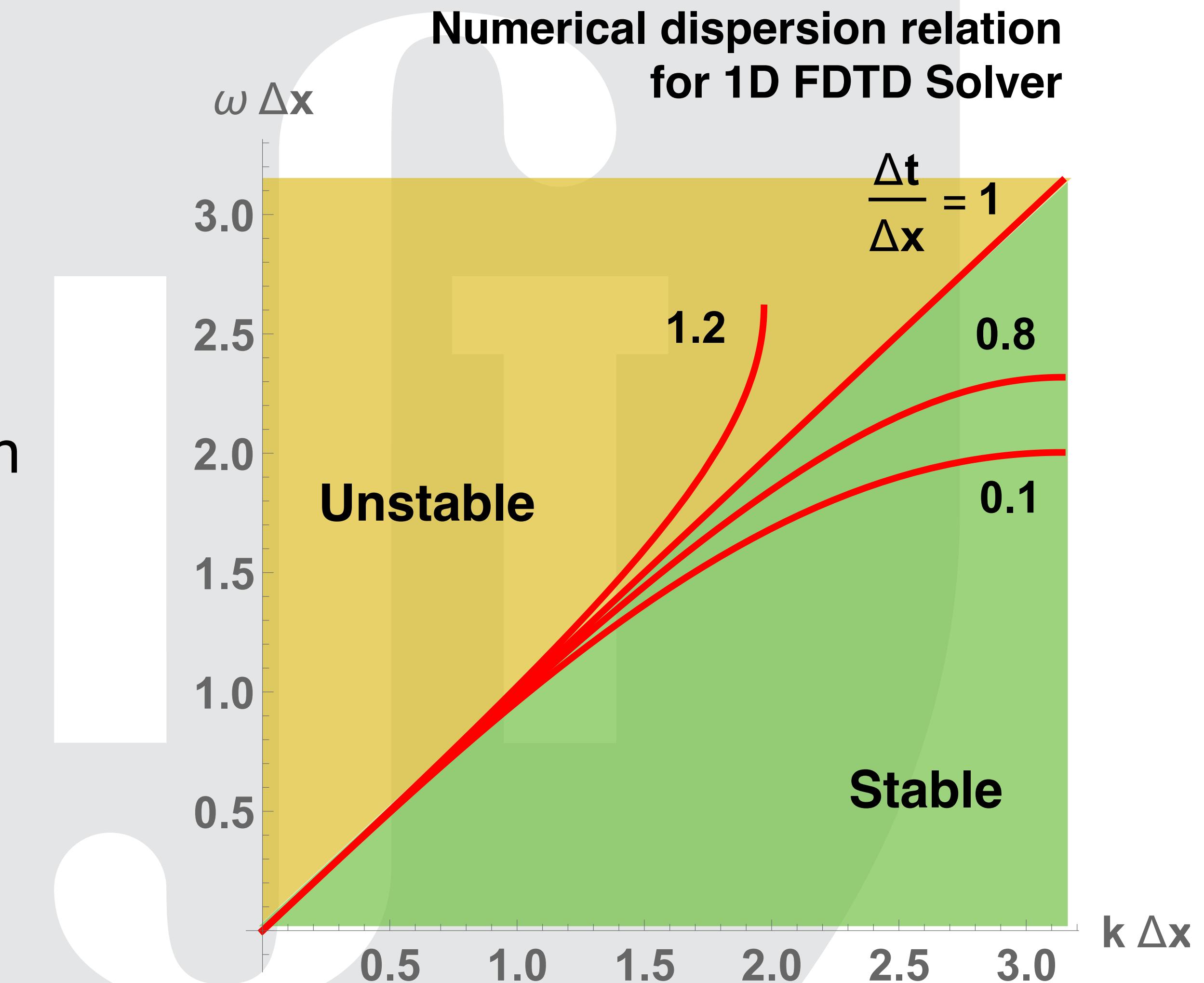
# Choice of time-step

- Choice of time-step is dominated by the FDTD solver (in sim. units):

$$\text{1D} \quad \Delta t \leq \Delta x$$

$$\text{2D} \quad \Delta t \leq (\Delta x^{-2} + \Delta y^{-2})^{-\frac{1}{2}}$$

- If time step is larger than Courant condition the field solver becomes unstable
- If time step is much smaller than courant condition for large  $k$ ,  $v_{ph}$  drops as low as  $2/\pi = 0.637 c$
- Relativistic particles may have  $v > v_0$ 
  - Numerical Cherenkov



# Units and Normalization in ZPIC

- **Careful choice of units and normalization is critical**
  - Avoids multiplication by several constants (e.g.  $m_e$ ,  $e$  and  $c$ ) improving performance and numerical accuracy.
  - By expressing the simulation quantities in terms of fundamental plasma quantities the results are general and not bound to some specific units we may choose
- **Units and normalization in ZPIC**
  - The frequencies are normalized to a normalization frequency,  $\omega_n$ . Time is normalized to  $\omega_n^{-1}$ .
  - Proper velocities are normalized to the speed of light,  $c$ . Space is normalized to  $c/\omega_n$ .
  - Charge and mass are normalized to the absolute electron charge,  $e$ , and the electron mass,  $m_e$ . The fields are then normalized appropriately.
  - The density is normalized to  $\omega_n^2$  (the normalization frequency squared). So if the density is 1 at a given location then the normalization frequency is the plasma frequency at that location.
  - If the laser frequency is 1, then the normalization frequency is the laser frequency and the density is normalized to the critical density (for that laser frequency).

zpic units

$$\mathbf{x}' = \frac{\omega_n}{c} \mathbf{x}$$

$$\mathbf{p}' = \frac{\mathbf{p}}{m_{sp}c} = \frac{\gamma \mathbf{v}}{c} = \frac{\mathbf{u}}{c}$$

$$\mathbf{E}' = e \frac{c/\omega_n}{m_e c^2} \mathbf{E}$$

$$\mathbf{B}' = e \frac{c/\omega_n}{m_e c^2} \mathbf{B}$$

$m_{sp}$  is the mass of the species

**zpic@edu**