(Beam) field Initialization in OSIRIS 4.0

TÉCNICO LISBOA

ISCTE IUL
Instituto Universitário de Lisboa

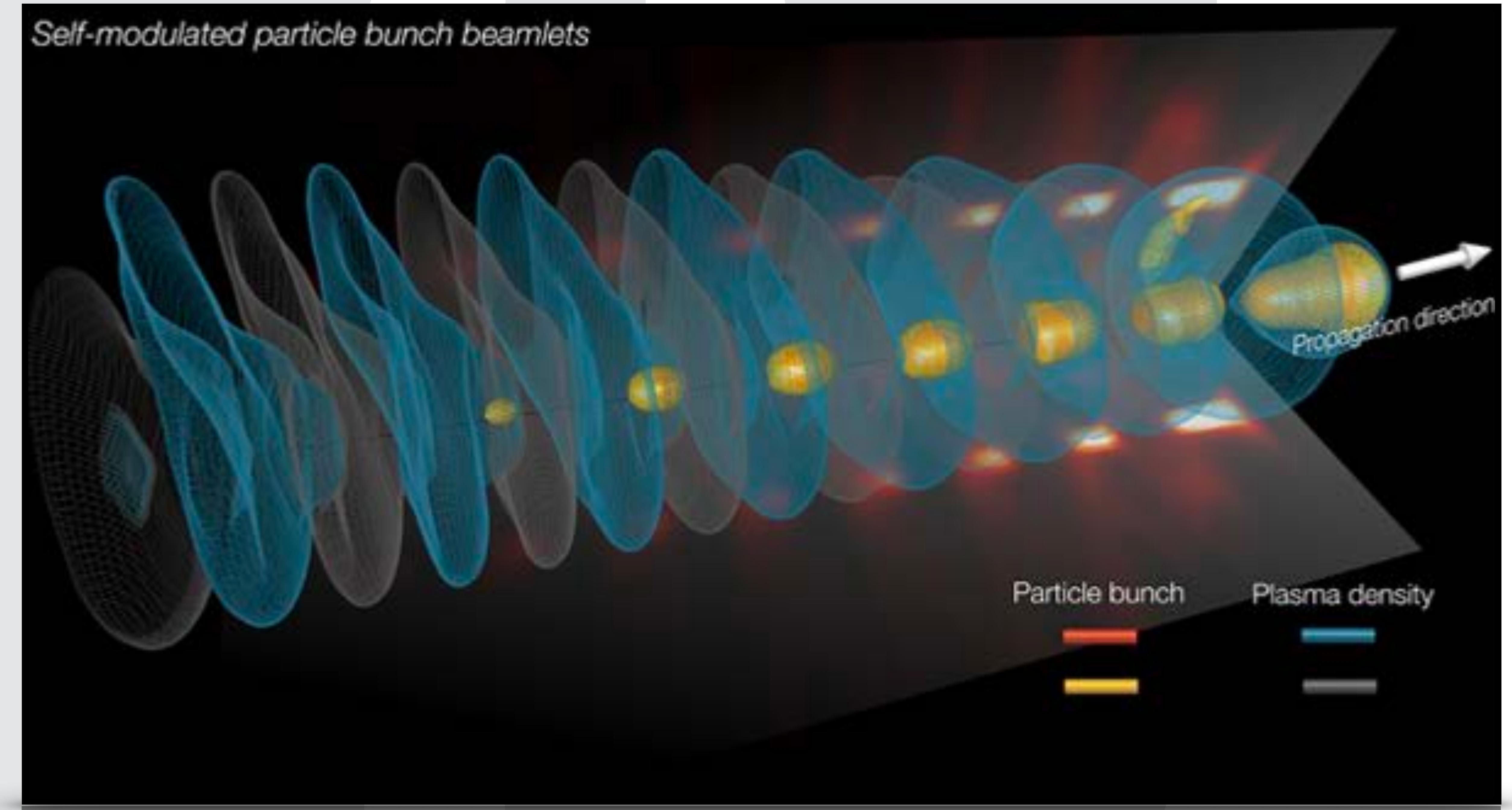
UCLA

A. Balzarini[1], R. A. Fonseca[1,2]

[1] GoLP/IPFN, Instituto Superior Técnico, Lisboa, Portugal
[2] DCTI, ISCTE-Instituto Universitário de Lisboa, Portugal

Setting the Initial fields for an OSIRIS simulation

- **The EM-PIC algorithm assumes self consistent initial electric and magnetic fields**
  - For a non-neutral plasma this leads to assuming a neutralizing background
  - By default OSIRIS sets **E** and **B** to zero initially
  - In OSIRIS 4.0 you can specify initial values for **E** and **B** using a mathematical expression
    - See the documentation for the *el_mag_fld* section
- **EM fields associated with a charged particle beam are complex**
  - Depending on the beam shape it may not be straightforward to compute an analytical expression for the initial field values
  - The existing algorithm works by slowly accelerating | free streaming the particle beam over a fixed number of time steps
    - Iterative calculation

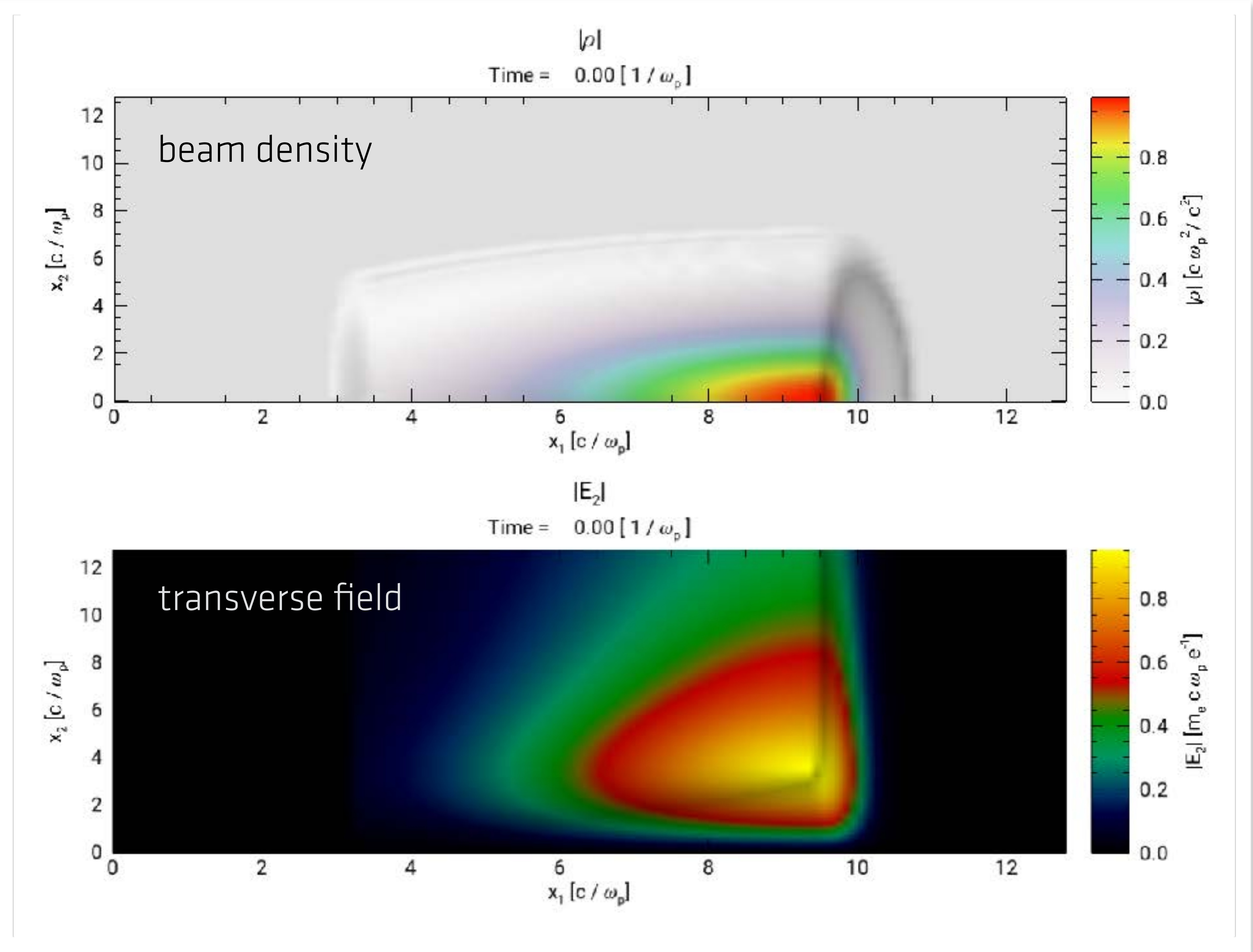


Calculating the Initial fields

- **Fields are calculated in the beam reference frame**
  1. Deposit beam charge in simulation frame
  2. Boost to beam frame
  3. Calculate electric field (no magnetic field in this frame)
  4. Boost back to simulation frame, calculating magnetic field

- **Electric field calculations**
  - Are performed using Coulomb's law
    - This assumes open boundaries
  - In the beam frame each charge cell becomes elongated along propagation direction
  - Assume each cell is an infinite slab (2D) or rod (3D) [*]
    - Only transverse fields!

- **Additional considerations**
  - The algorithm has a specific parallelization making it very efficient
  - If using PML boundaries the algorithm also adjusts the values inside the guard cells to avoid spurious reflections



[*] In 2D it is possible to choose (at compile time) a different version that considers the cells to be rectangles but the differences are negligible

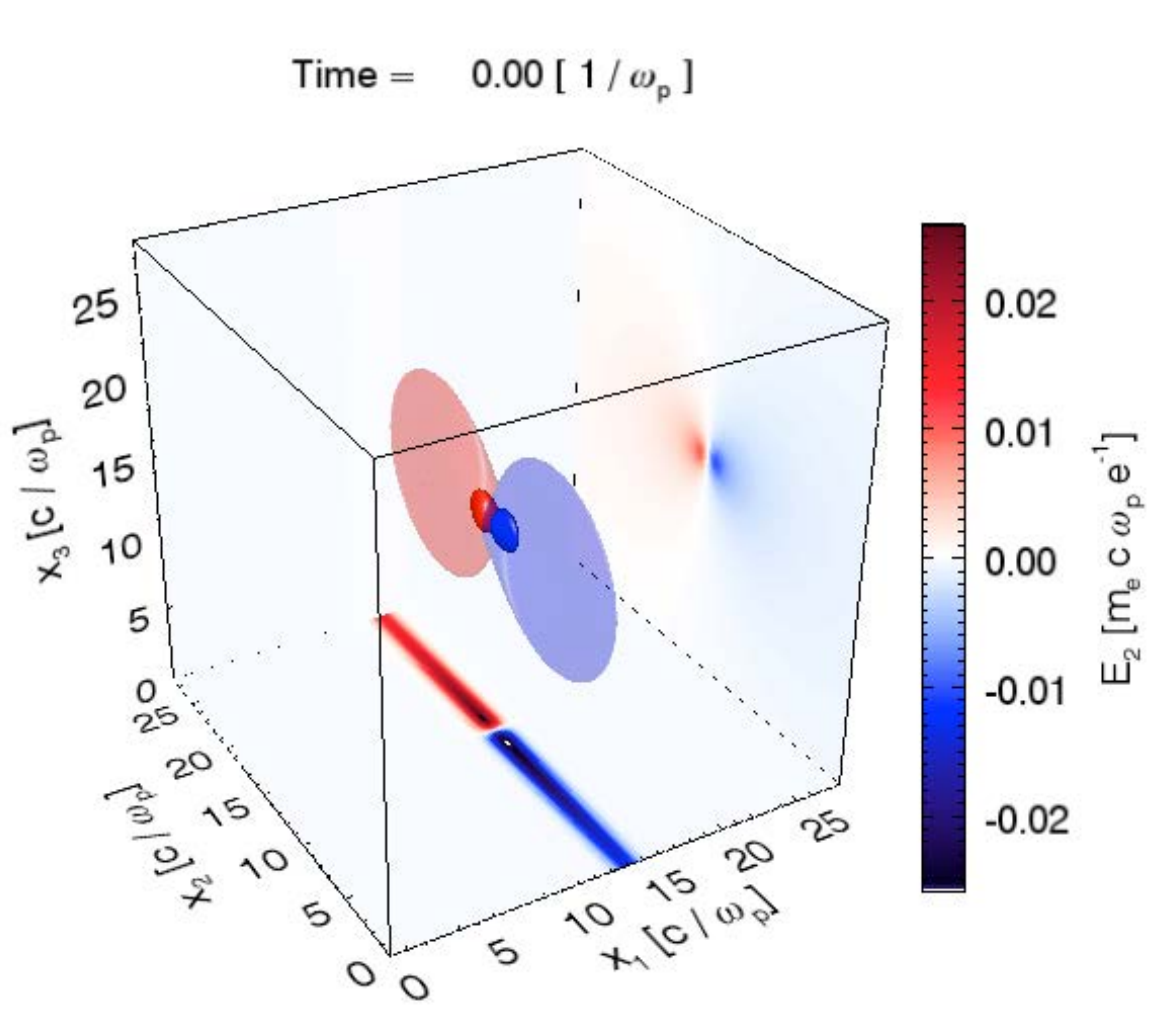
How to use it / limitations

- **The species section includes a new parameter**
  - The code also uses the parameters set in the udist section

```
species
{
  name = "beam",
  rqm = -1.0d0,
  num_par_x(1:2) = 2, 2,

  init_fields = .true.,
}
```

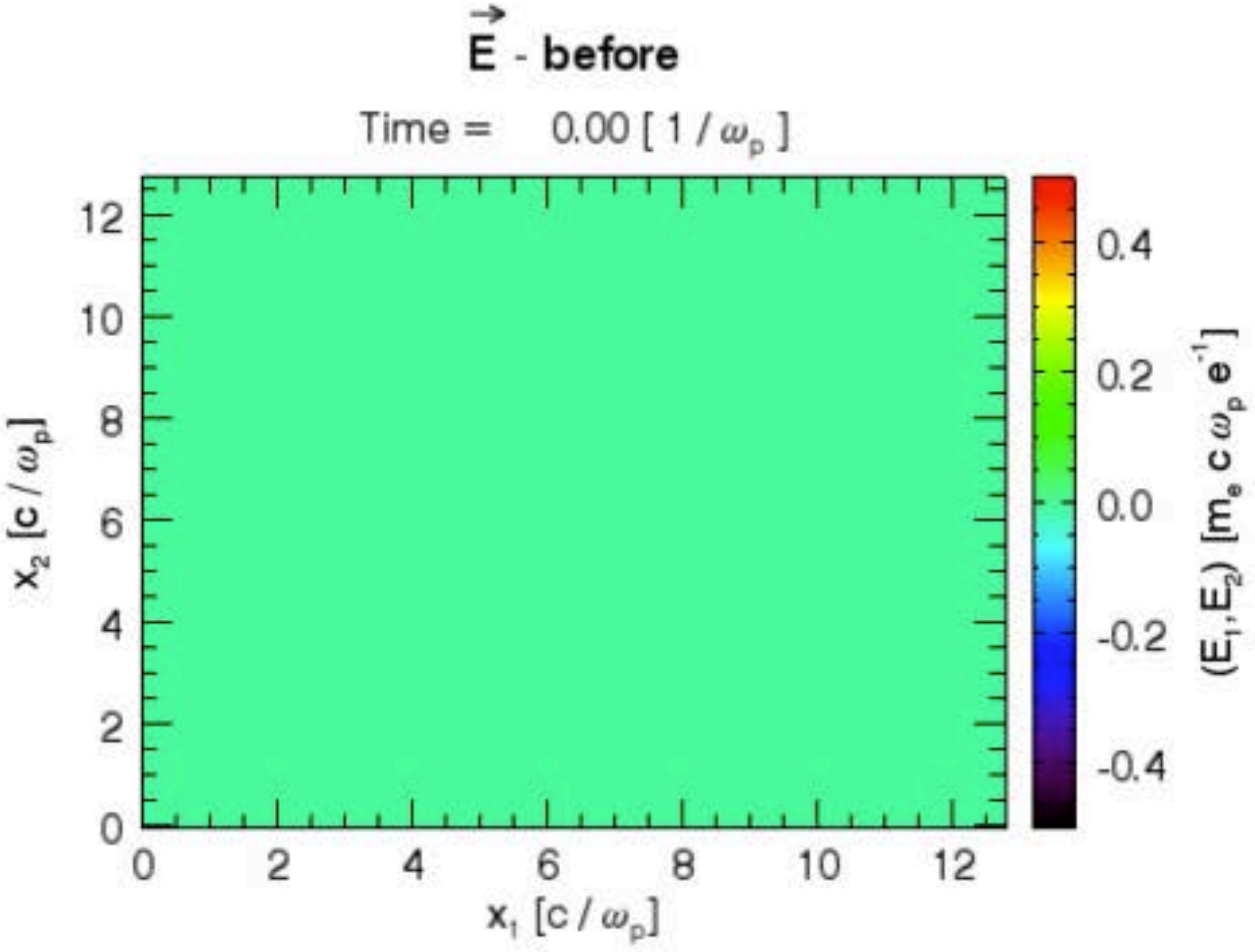
- **Limitations**
  - The calculated fields assume open boundaries, they are not consistent with conducting or periodic boundaries
  - The approximation used for calculating the fields does not hold for very slow beams
  - No longitudinal fields are calculated

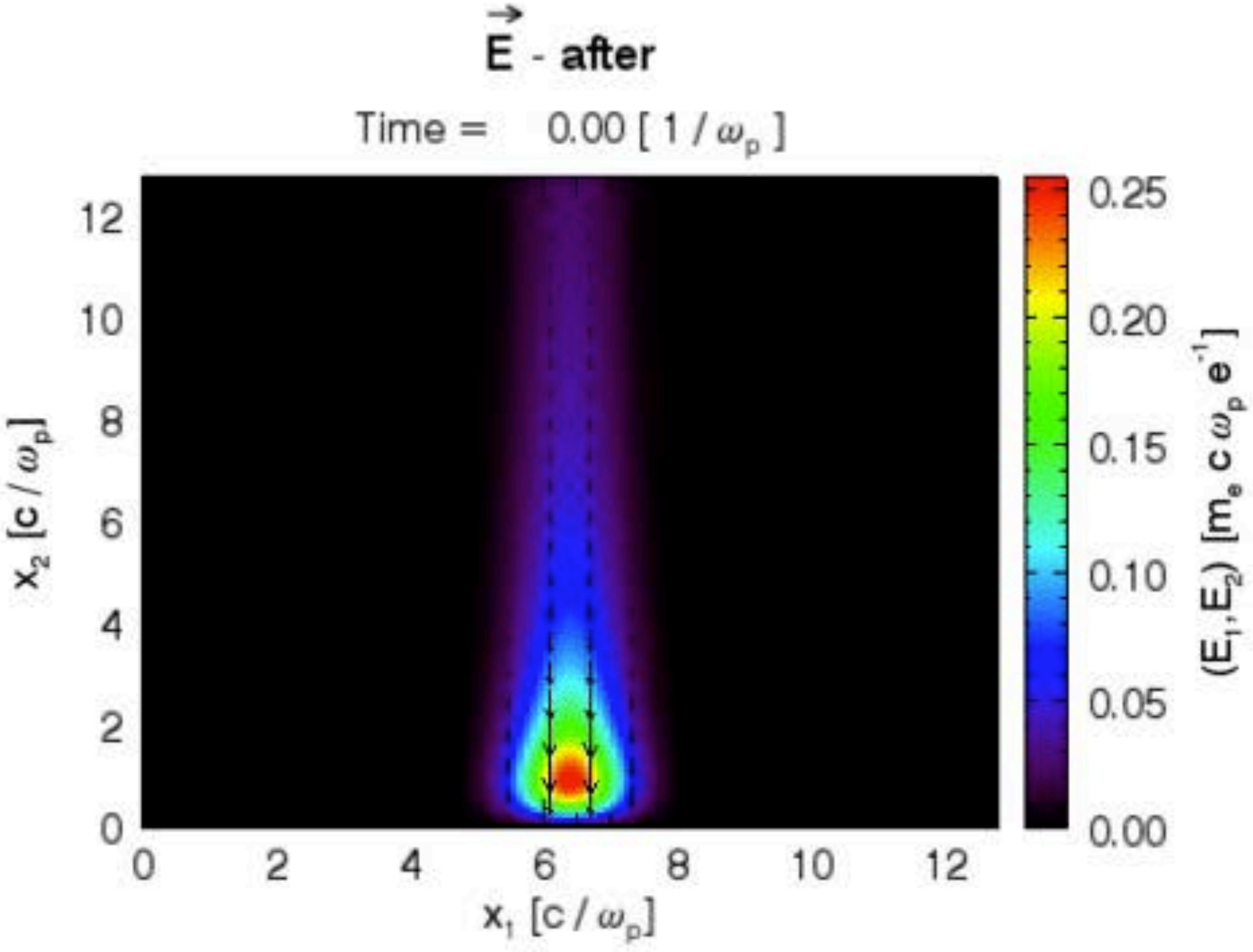


Old method vs. new method

2D cylindrical geometry

Gaussian particle beam





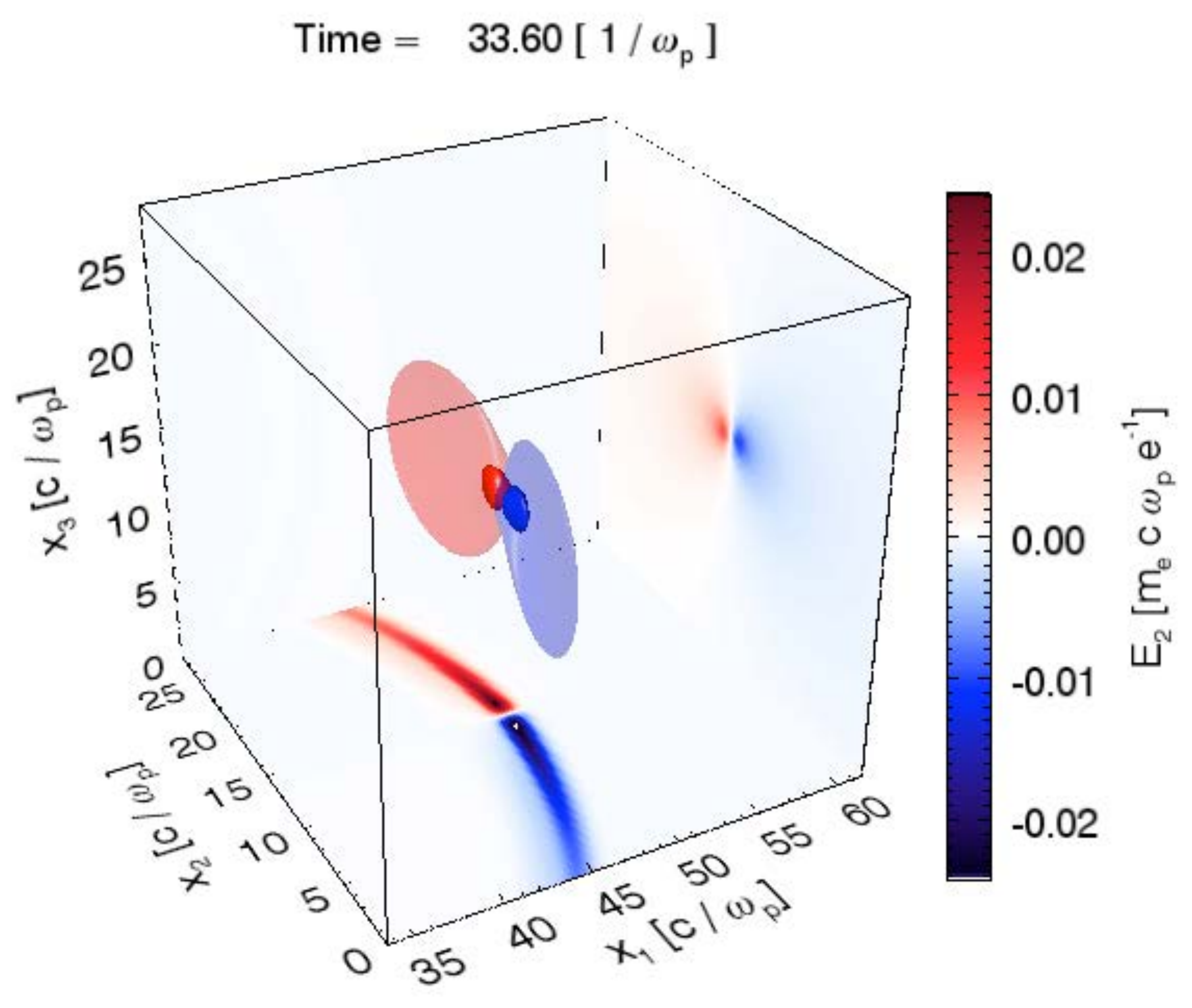
Beam initialization using old method

- n_accelerate = 1000

Beam initialization using new method

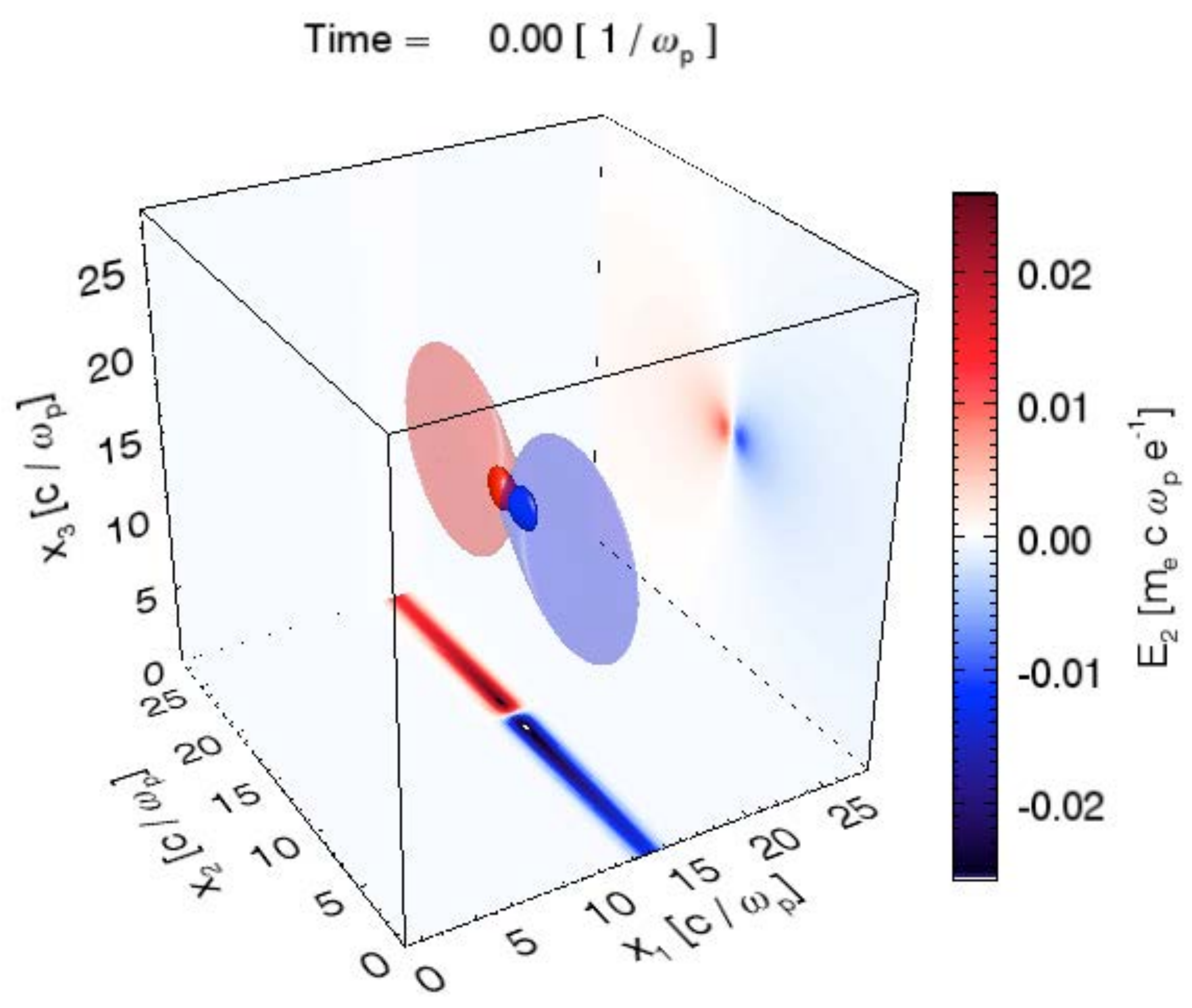
Performance considerations

Old algorithm - 4000 timesteps



- Simulation parameters
  - 256^3 cells
  - 2^3 particles per cell
  - 400 cores
- Time for initialization: **3 hours**

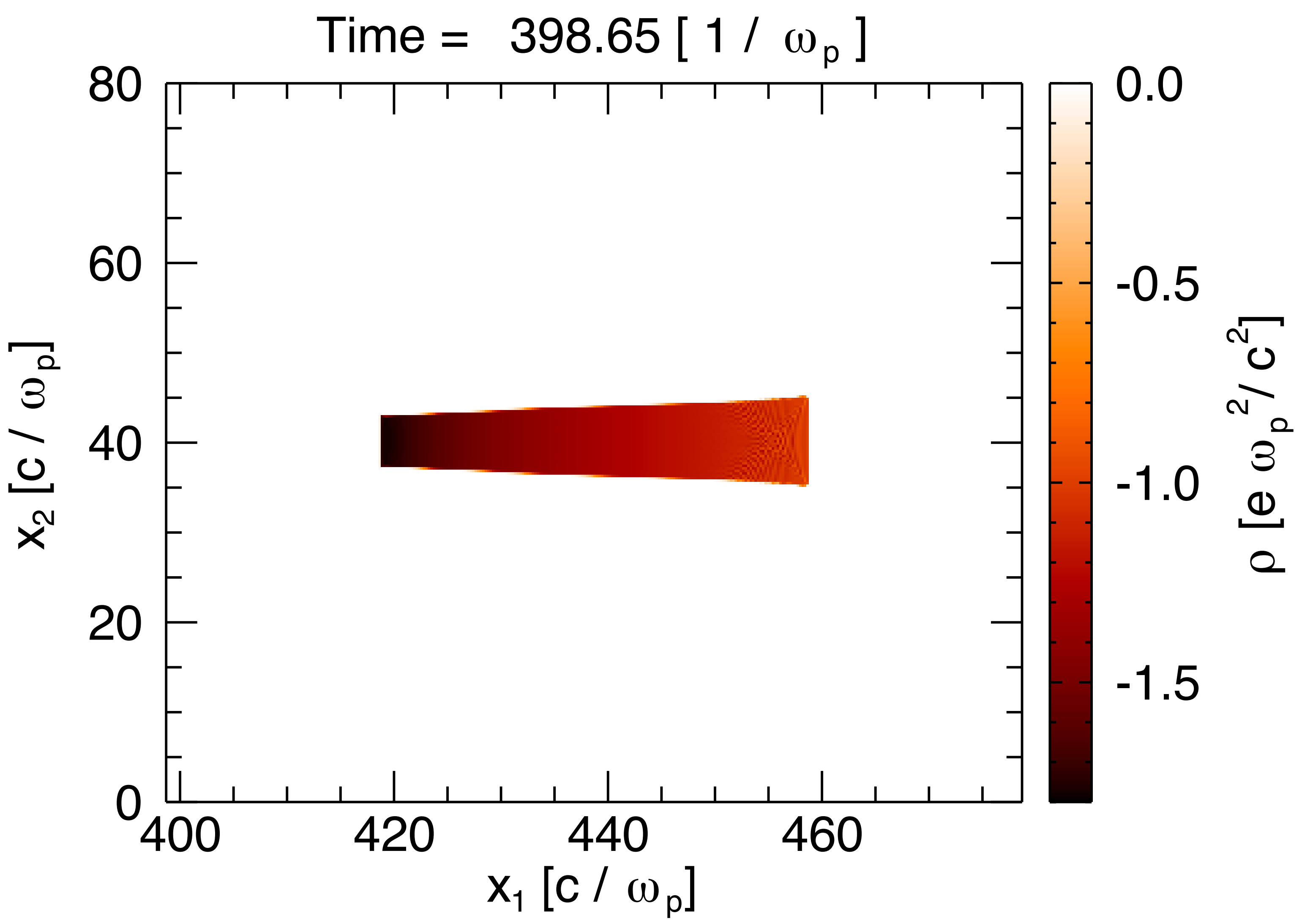
New algorithm



- Simulation parameters
  - 256^3 cells
  - 2^3 particles per cell
  - 400 cores
- Time for initialization: **10 s**

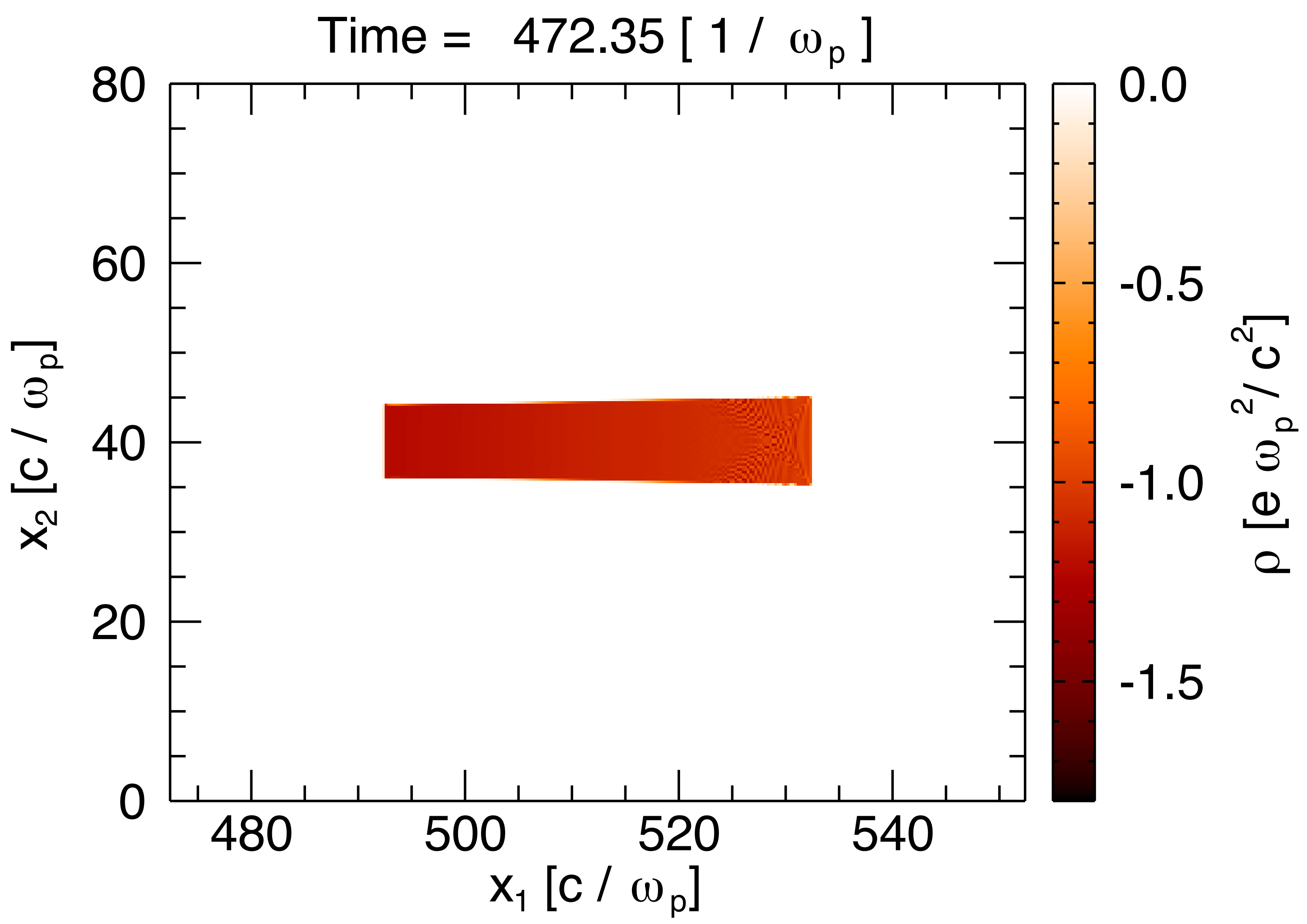
Avoids non-physical beam pinching

Flat top beam propagating in vacuum



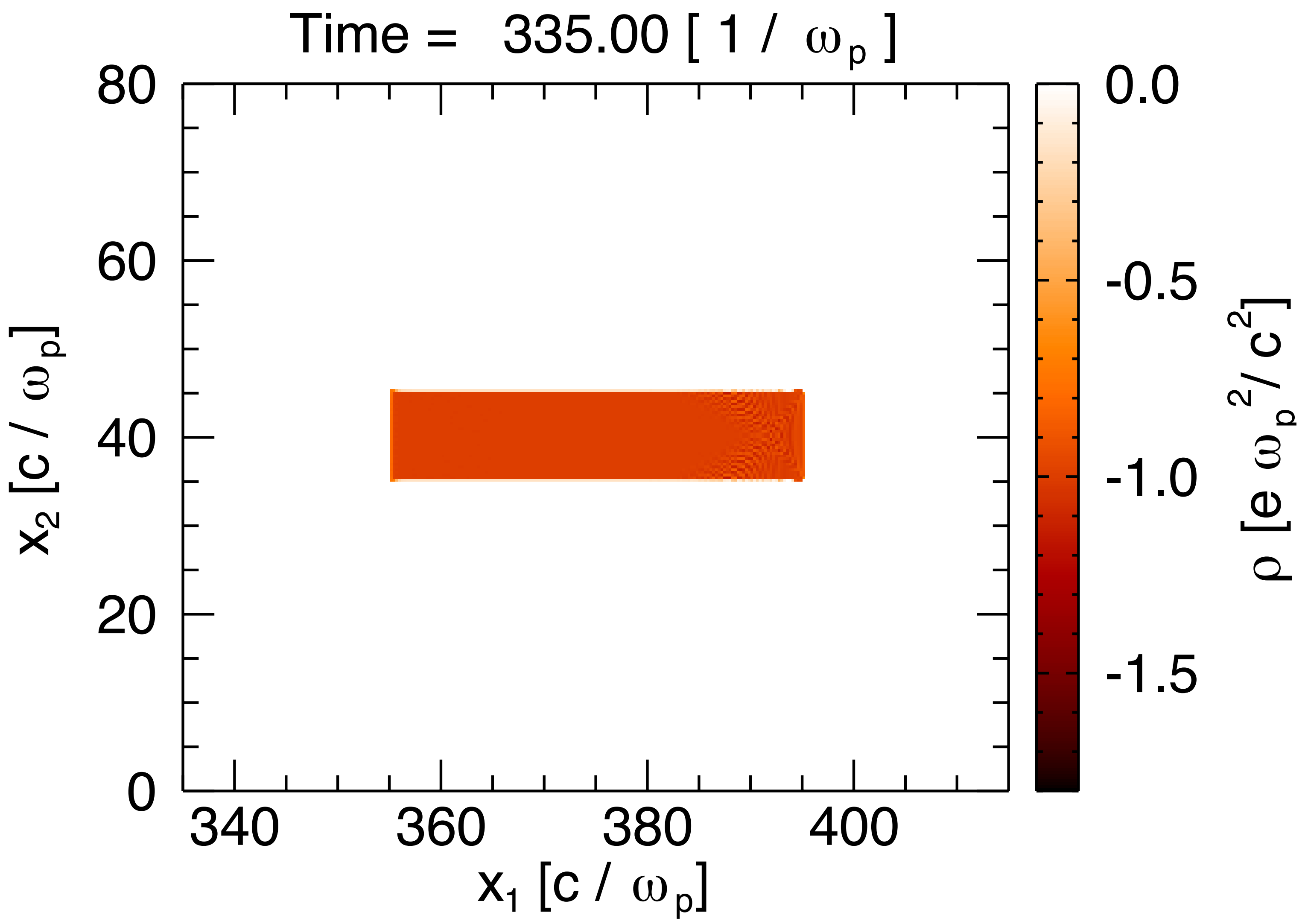
Old algorithm

1000 timesteps



Old algorithm

2000 timesteps



New algorithm

The different times corresponding to the same time after the initialization part.

Overview

- **New method for calculating initial beam fields**
  - Uses Couloumb's law in beam reference frame
  - Also works for non-neutral plasma distributions at rest (2D)
- **Much faster/accurate than existing algorithm**
  - Assumes $u_{\text{beam}} \gg 1$
  - Only transverse components are calculated
- **Available in 4.0 series**
  - Use the *init_fields* parameter in the species section
  - May give substantially different results from the previous method
    - If the previous method was not given time to converge
  - Also see the field initialization in the *el_mag_field* section