Fourier-Bessel Particle-In-Cell code (FBPIC)

Remi Lehe (LBNL)

Developed in collaboration with
Manuel Kirchen & Soeren Jalas (CFEL, Hamburg)

Outline

- Overview of the algorithm: what does the code do?
- Useful features for laser-plasma acceleration
- From a user's perspective

Relation between Warp and FBPIC

Both codes have an **electromagnetic PIC solver**,
that can be used in order to **simulate laser-plasma acceleration**.

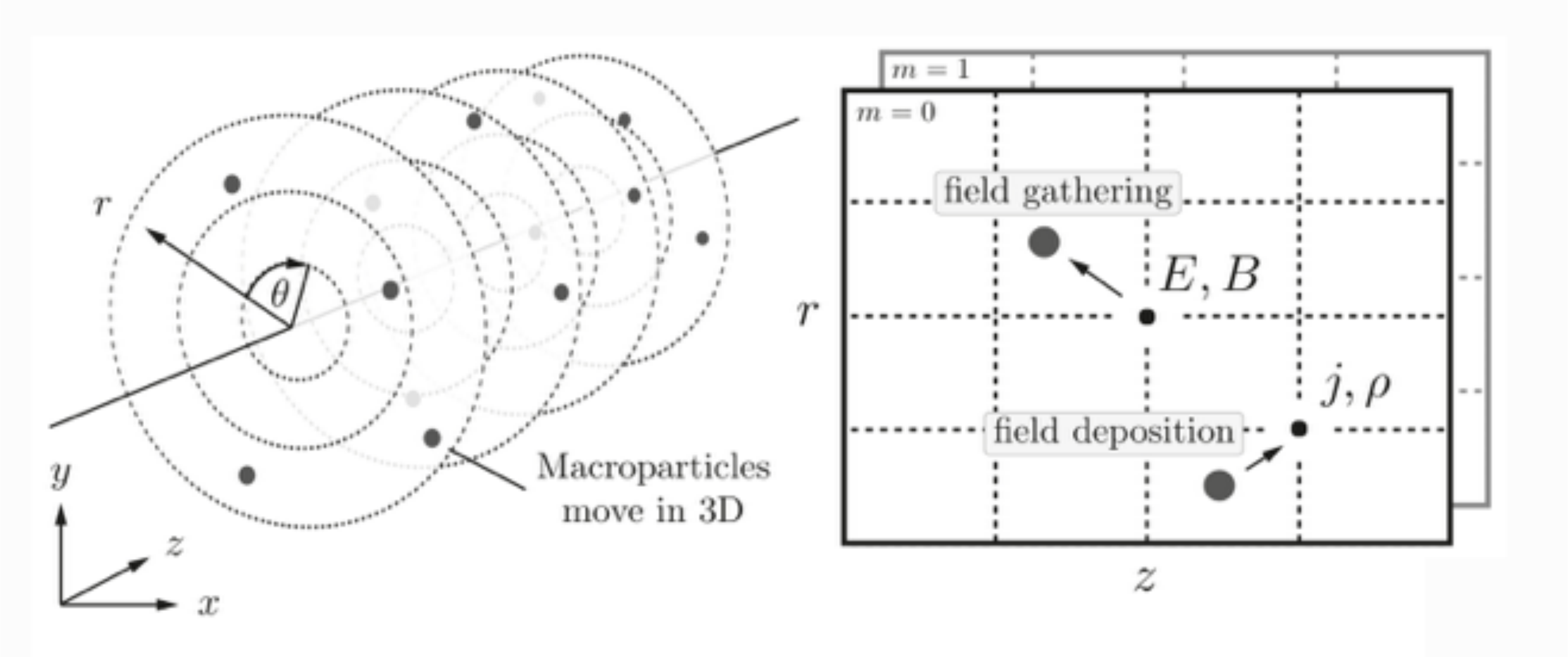
Warp

- **Available solvers/geometries for EM PIC:**
  - Cartesian, finite-difference
  - Cylindrical (quasi-3D), finite-difference
  - Cartesian, spectral
- **Also contains electrostatic solvers, quasi static solvers, e-gun mode, etc.**
- **Very generic framework, can be used for a variety of problems**

FBPIC

- **Available solver/geometry for EM PIC:**
  - Cylindrical (quasi-3D), spectral
- **Contains specific optimizations for this type of solver:**
  - specific MPI exchange patterns
  - fully ported to GPU
- **Specialized code, for plasma-based acceleration in nearly-cylindrical geometry**

Cylindrical (quasi-3D) geometry: faster simulations



- **The fields are decomposed into azimuthal modes**

$$F(r, z, \theta) = Re\left[\sum_{m=0}^{N_m - 1} \hat{F}_m(r, z) e^{im\theta}\right]$$

m=0: purely cylindrical mode
m=1: dipole mode
m=2: quadrupole mode

- **Each azimuthal mode is represented by a 2D r-z grid**

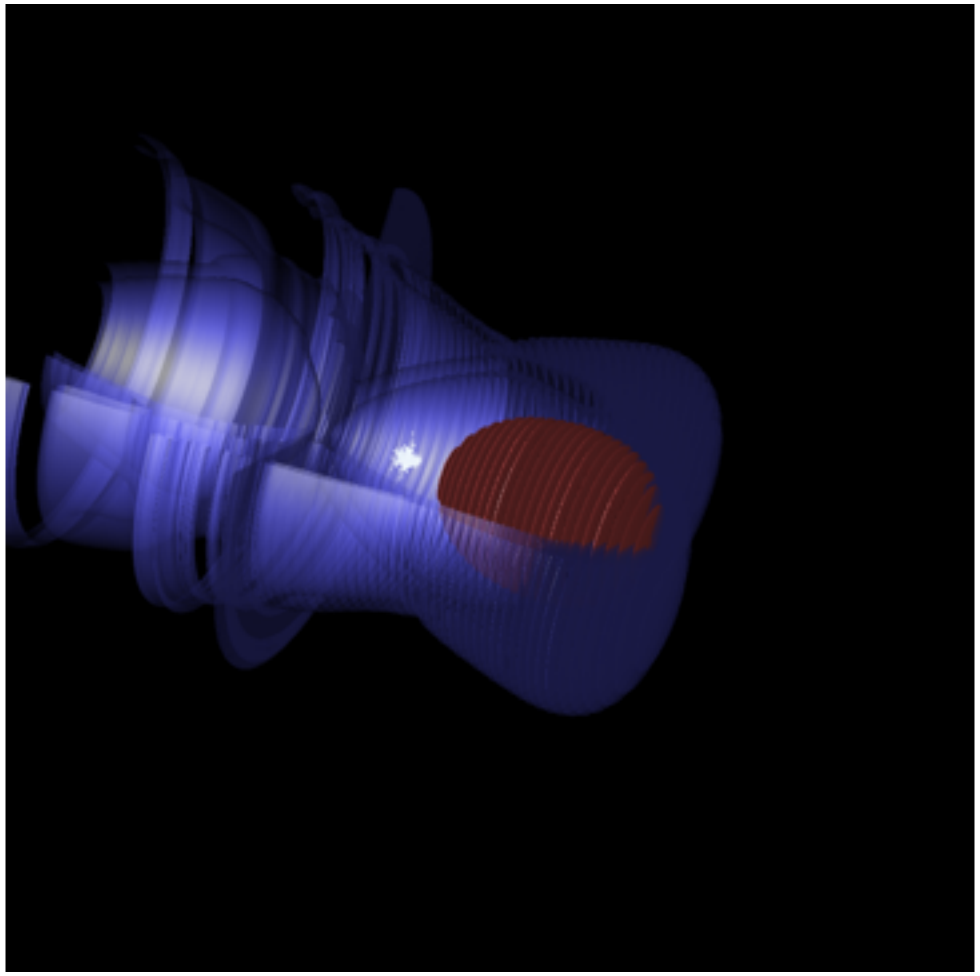
Cylindrical (quasi-3D) geometry: faster simulations

Key advantage

For many physical problems, **only a few modes** are needed.

Using only a few modes (2D grids) requires **vastly less computational time and memory** than a full 3D Cartesian grid.

- **Beam-driven plasma acceleration with <u>cylindrical</u> driver beam**
  m=0
- **Beam hosing <u>in the weakly-perturbed regime</u>**
  m=0 (unperturbed beam)
  and m=1 (first-order perturbation)
- **Laser-driven plasma acceleration with <u>cylindrical</u> laser**
  m=0 (for the wakefield)
  and m=1 (for the laser)



Spectral EM PIC: more robust

Finite-difference EM PIC

- **Derivatives (in the Maxwell equations) are evaluated with finite differences between grid points.**
- **Advantage:** Fast, easy to parallelize

Spectral EM PIC

- **Derivatives (in the Maxwell equations) are evaluated in spectral space** (involves Fourier and Hankel transforms)
- **Advantage:** Usually more accurate/robust

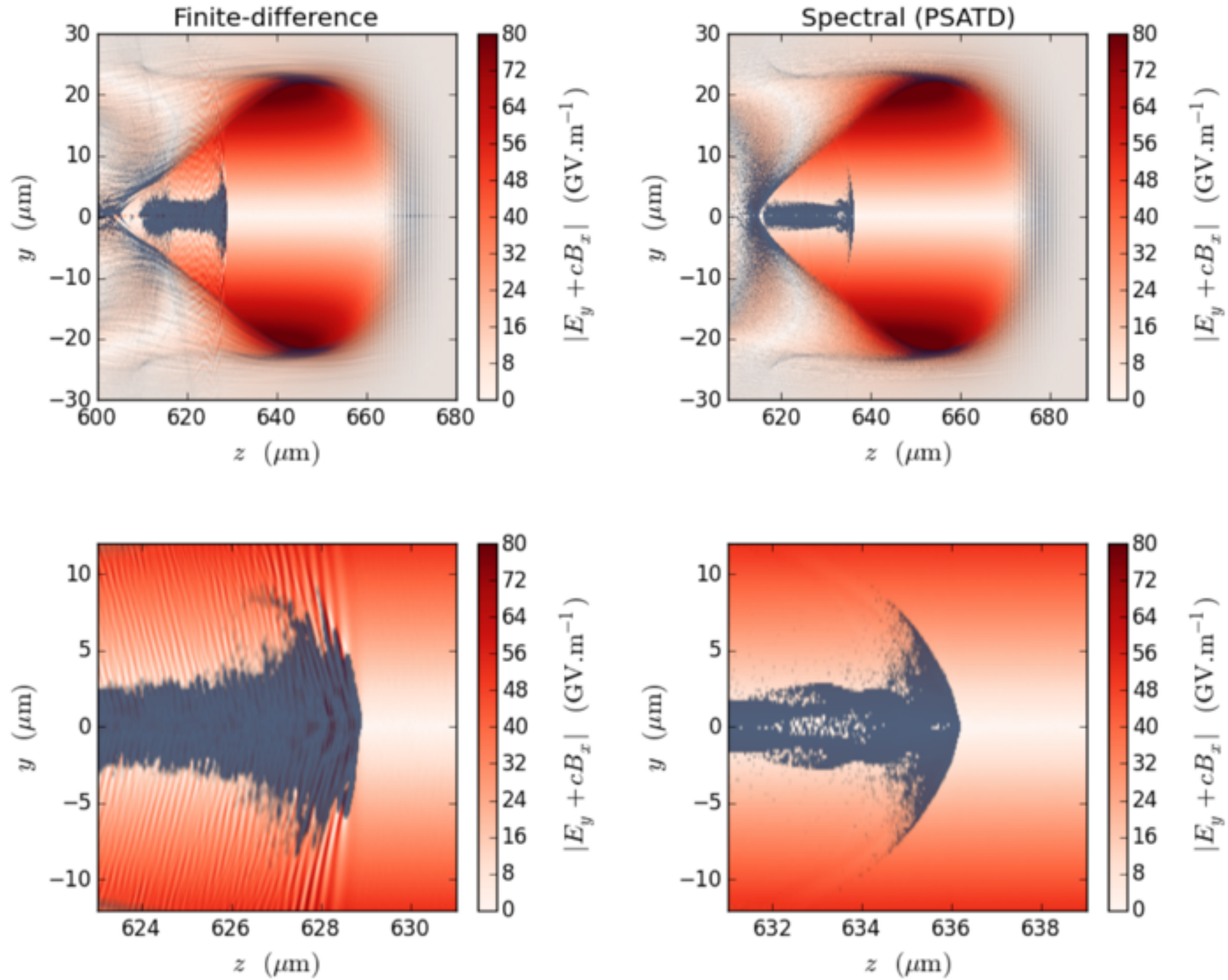
A few more specific advantages of spectral analytical EM PIC:

- **No Courant condition**
- **EM dispersion relation is much closer to the physical one**
- **Reduced noise on the axis (for cylindrical geometry)**

More details on Wednesday's lunch talk

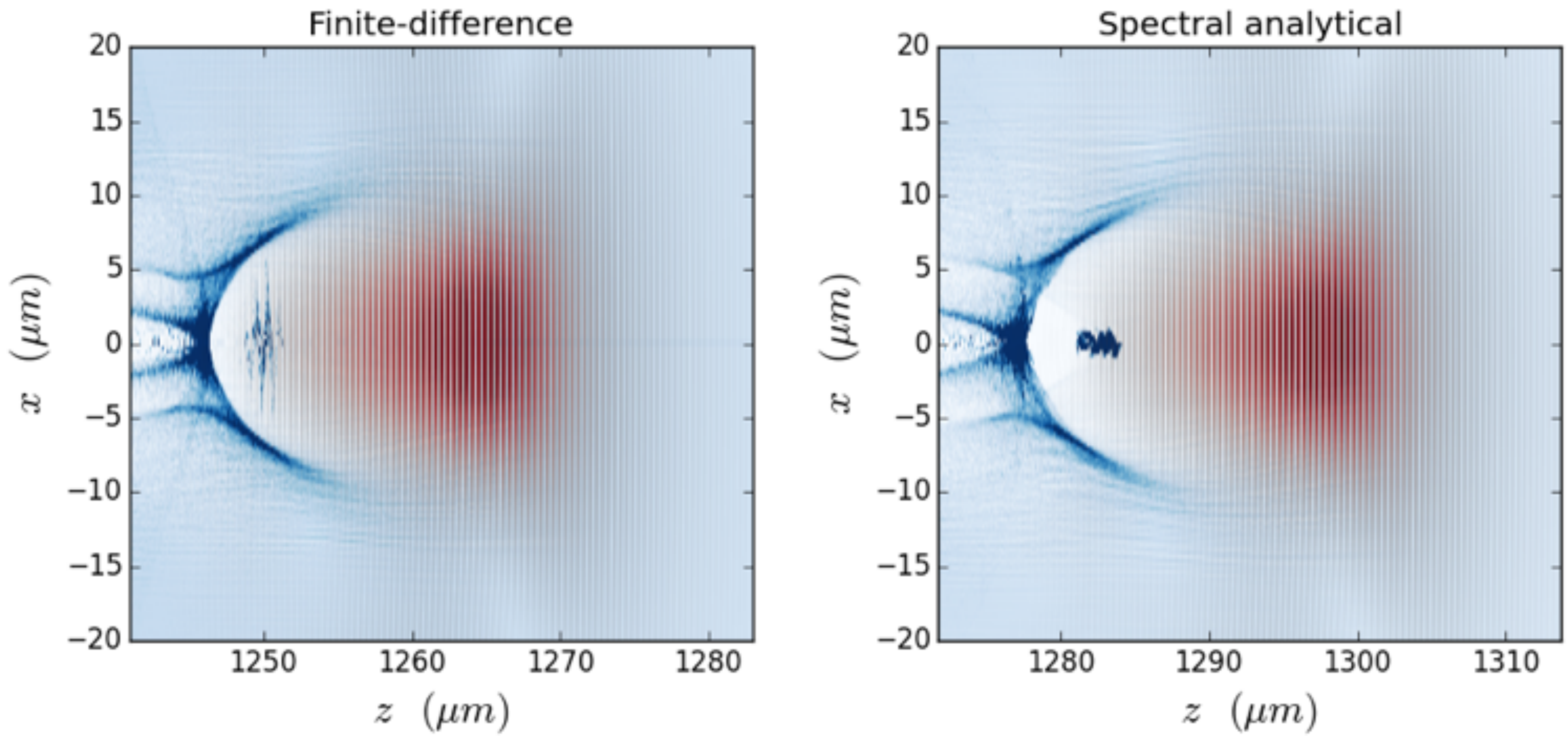
Spectral EM PIC: mitigated Cherenkov radiation

- Spurious Cherenkov radiation appears in **finite-difference**
- This radiation can interact with the bunch and increase its divergence.
- Does not happen for spectral codes because of the improved dispersion relation



Spectral EM PIC: E and B cancelation

- In finite-difference codes, E and B are usually staggered in time and space.
- This makes it difficult to accurately capture the **physical cancelation** between E and v x B
- This does not happen in **centered** spectral codes.



Outline

- Overview of the algorithm: what does the code do?
- Useful features for laser-plasma acceleration
- From a user's perspective

GPU acceleration

Because of the reduced memory requirements,
as opposed to 3D Cartesian,
simulations can often fit on a **single GPU**

The whole PIC loop was ported to GPU
(to avoid CPU-GPU transfer)

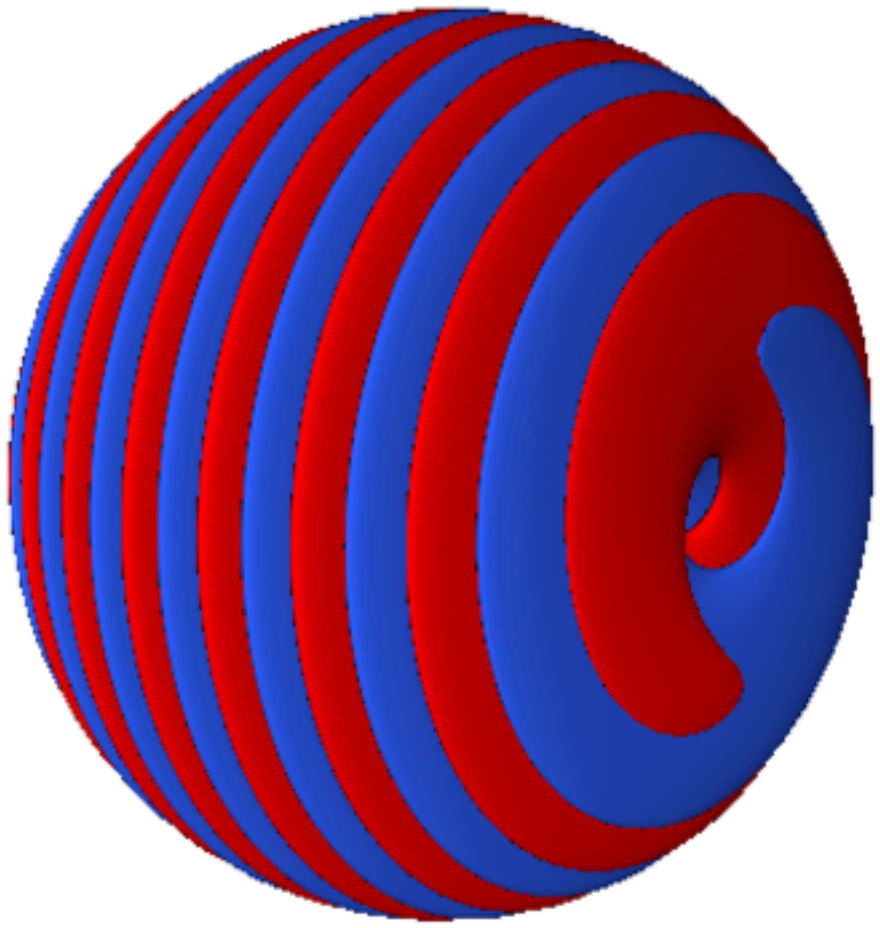
Acceleration of **~3x-10x** compared
to the multi-threaded CPU version
(for large simulations)

Relevant physics included

- **Beam initialization with space-charge**
  Automatic initialization of the E and B field from a relativistic beam

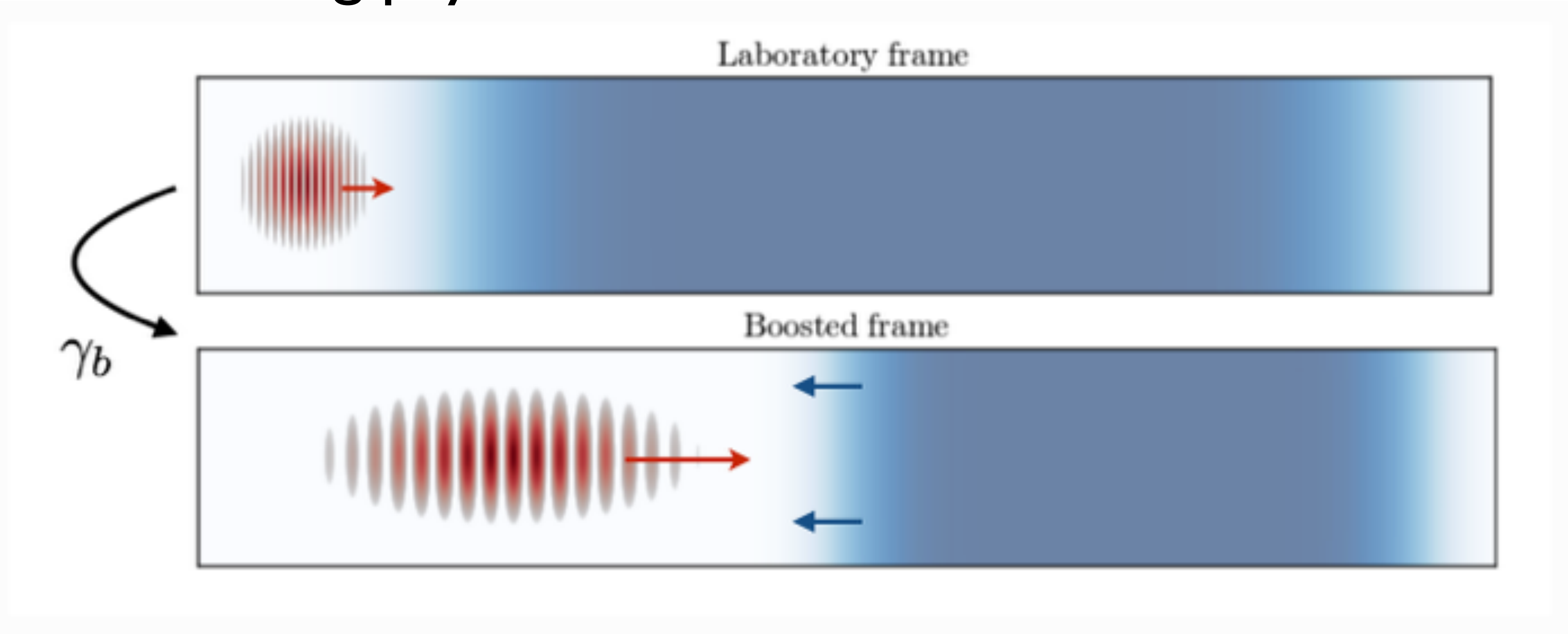
- **Laser initialization**
  Multiple laser profiles available (e.g. including chirp, Laguerre modes, etc.)

- **Field ionization**
  - ADK model
  - includes efficient GPU implementation



Boosted-frame

Simulations can be orders of magnitude faster by simulating physics in the boosted frame.



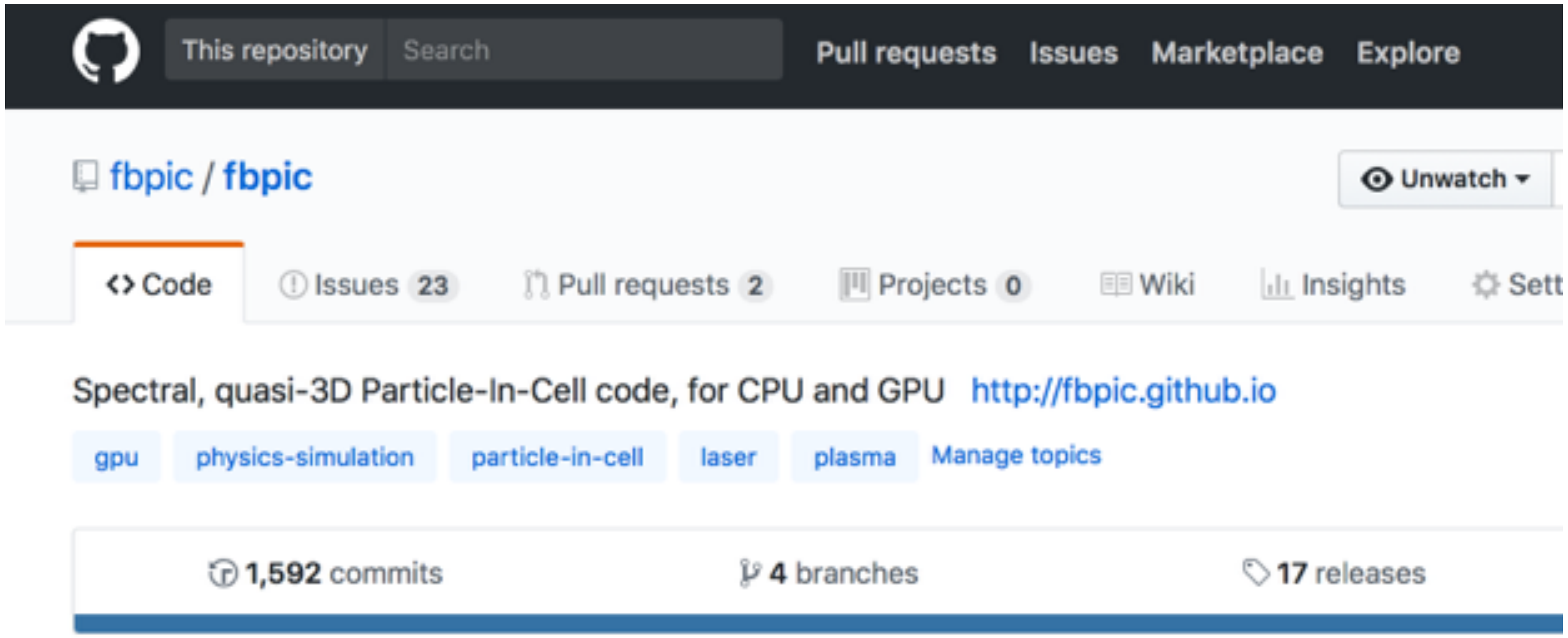
- **fbpic includes convenient functions to**
  - Convert input parameters to the boosted frame
  - Convert output to the lab frame on the fly (including efficient GPU implementation)
  - Avoid the Numerical Cherenkov Instability

Outline

- Overview of the algorithm: what does the code do?
- Useful features for laser-plasma acceleration
- From a user's perspective

Getting the code

- **Open-source, available on Github: https://github.com/fbpic/fbpic**



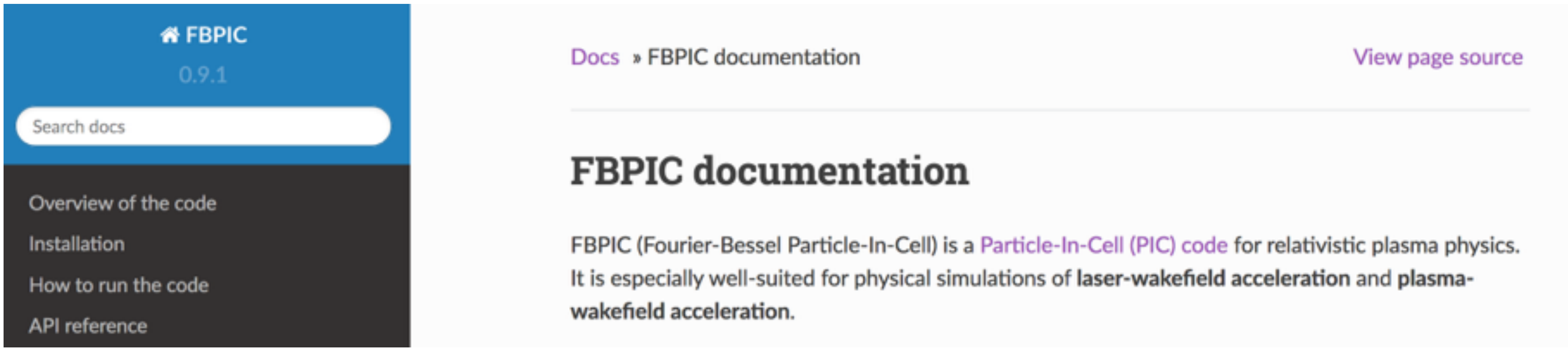
- **Easy installation through the Anaconda distribution of Python**

```
conda install numba scipy h5py mkl
pip install fbpic
```

For GPU support: `conda install pyculib`

Running the code

- **Online documentation: https://fbpic.github.io**



- **Running fbpic:**
  - Simple python file as a run file: python fbpic_script.py
  - No compilation step (compilation is automatically done on-the-fly)
  - Example files available in the online documentation

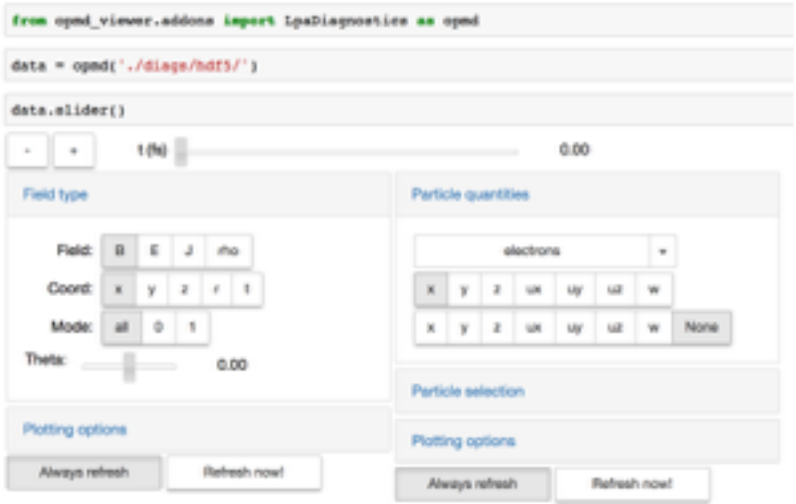
More details in the tutorial sessions on Monday, Tuesday, or Wednesday

Analyzing/visualizing the output of the code

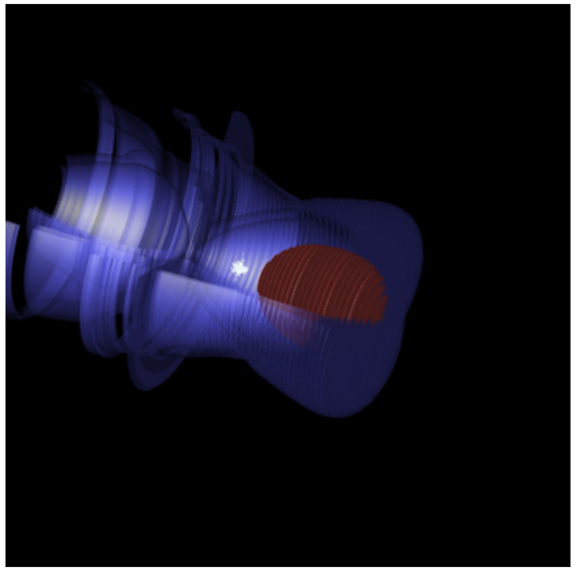
- **Code output uses the openPMD format**
  Standardized layout for HDF5 files,
  adopted by several PIC codes including Warp

- **Can leverage the open-source visualization tools that were developed for these codes**

openPMD-viewer:
github.com/openPMD/openPMD-viewer



Visit plugin (beta):
github.com/openPMD/openPMD-visit-plugin



More details in the tutorial sessions and on Wednesday morning's talk for openPMD

Thanks for your attention!