

PIConGPU

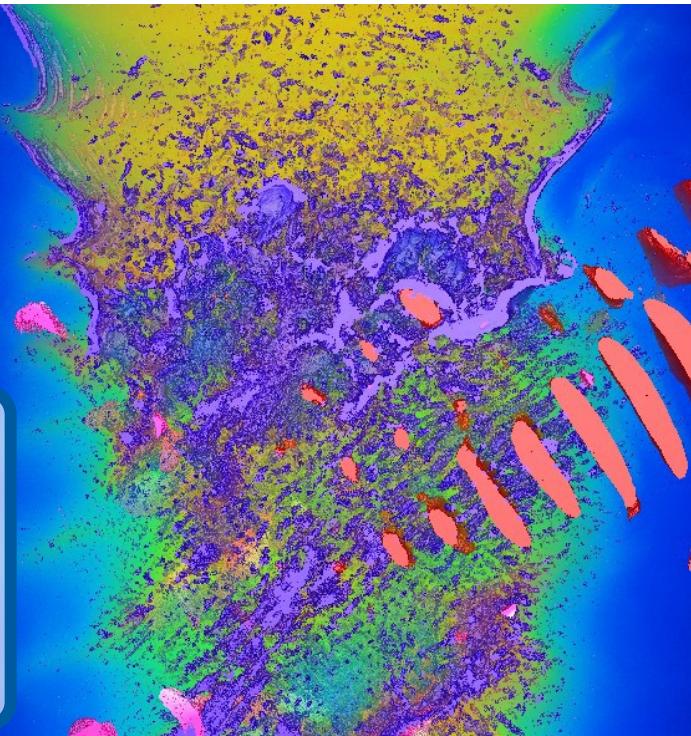
Introduction, Concepts and use of Libraries

HZDR · FWKT · Dr. Richard Pausch · r.pausch@hzdr.de · www.hzdr.de

PICon GPU

R. Pausch¹, S. Bastrakov¹, M. Bussmann^{1,2},
F.-O. Carstens^{1,3}, A. Debus¹, J. Lenz^{1,2},
B. E. Marre^{1,3}, T. Narwal^{1,2}, P. Ordyna^{1,3},
F. Pöschel^{1,2}, K. Steiniger^{1,2}, J. Tiebel^{1,3},
R. Widera¹

1 HZDR, 2 CASUS, 3 TU Dresden



collaborations and projects:



HELMHOLTZ AI

M T DMA

HiBEF

ATHENA

CASUS
CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

CAAR
OLCF
OAK RIDGE LEADERSHIP COMPUTING FACILITY



EuroHPC
Joint Undertaking

PLASMA
PEPSC

PIConGPU

Introduction, Concepts and use of Libraries

HZDR · FWKT · Dr. Richard Pausch · r.pausch@hzdr.de · www.hzdr.de

DRESDEN
concept



Laser Plasma Physics at HZDR

What are we using PIConGPU for?

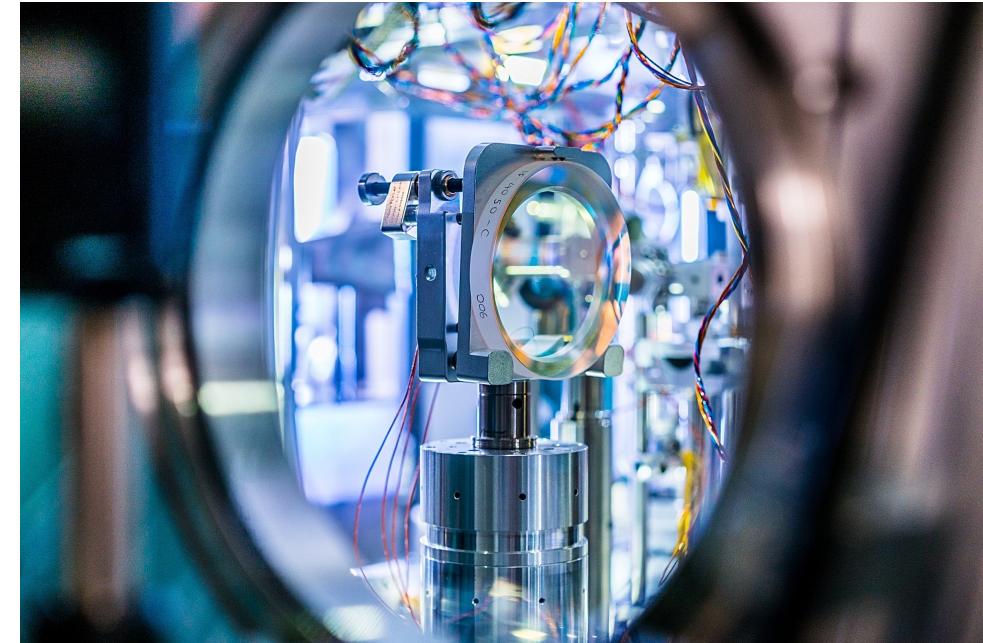
Laser Plasma Acceleration

Compact particle accelerators with high power lasers

short high-power laser pulses

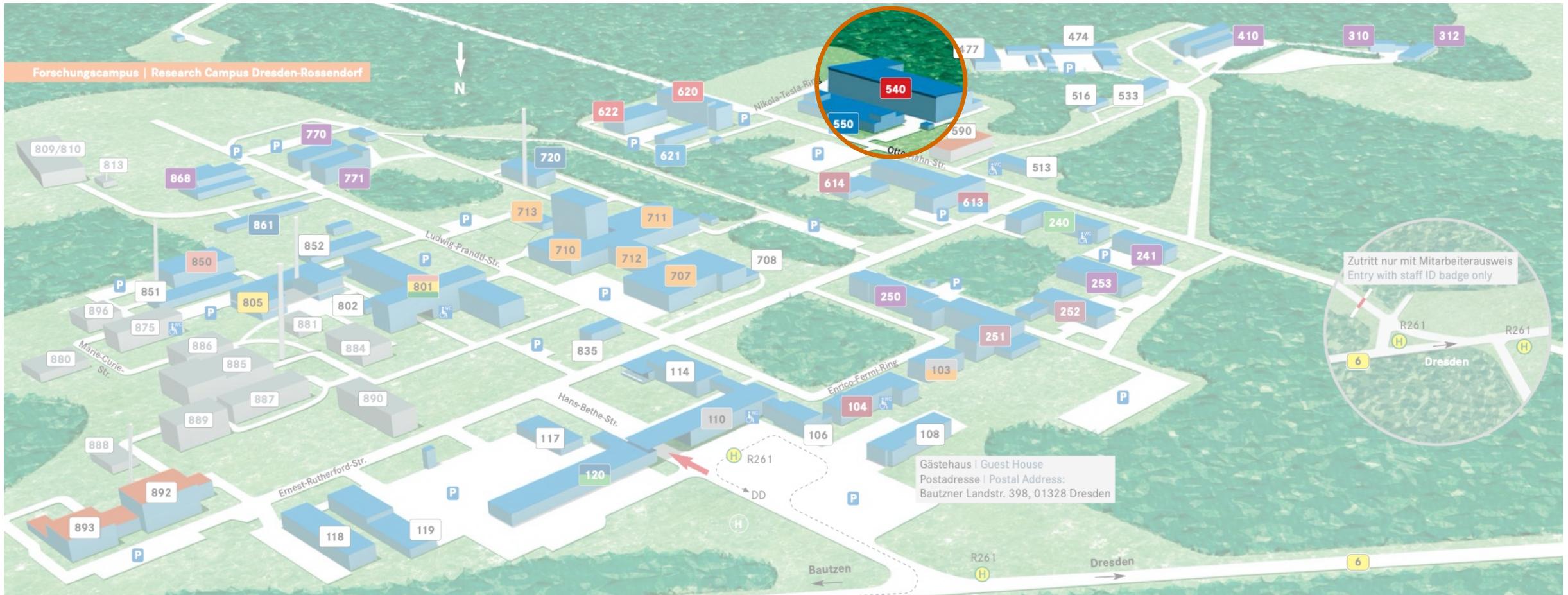


focused on a plasma to
accelerate particles



Laser Plasma Acceleration

Compact particle accelerators with high power lasers

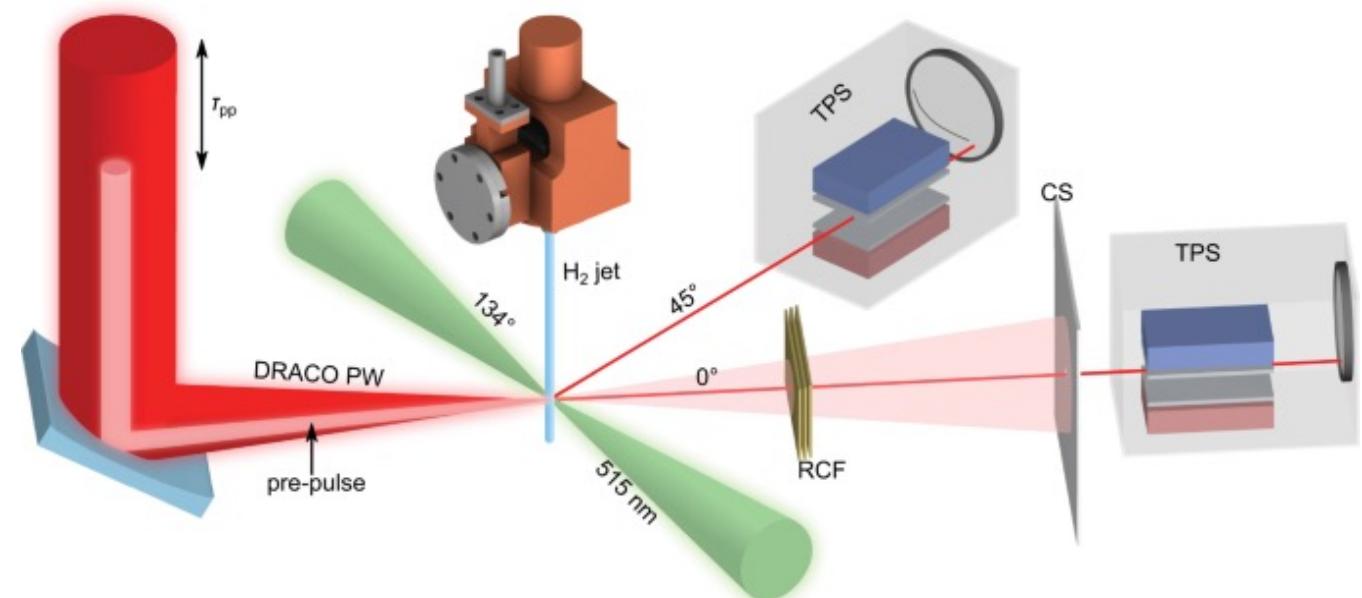


Laser Proton acceleration using a cryogenic hydrogen jet

Laser Proton acceleration using a cryogenic hydrogen jet

a near-critical, debris-free target to efficiently accelerate protons

- cryogenic hydrogen jet provides:
 - self-replacing
 - debris-free
 - near-critical target
- up to 80 MeV protons could be achieved
- much easier to diagnose the plasma state



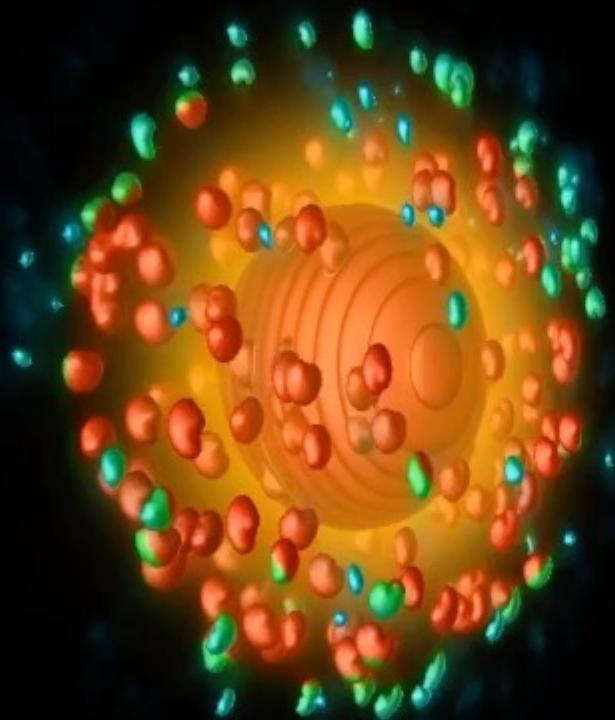
Rehwald, M., Assenbaum, S., Bernert, C. et al.

**Ultra-short pulse laser acceleration of protons
to 80 MeV from cryogenic hydrogen jets tailored
to near-critical density**

Nat Commun 14, 4009 (2023)

DOI: 10.1038/s41467-023-39739-0

Electron acceleration with laser wakefield acceleration (LWFA)



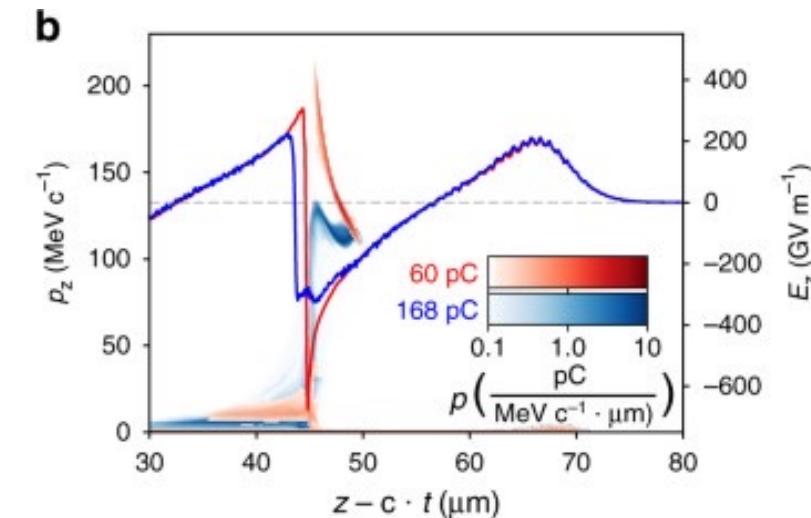
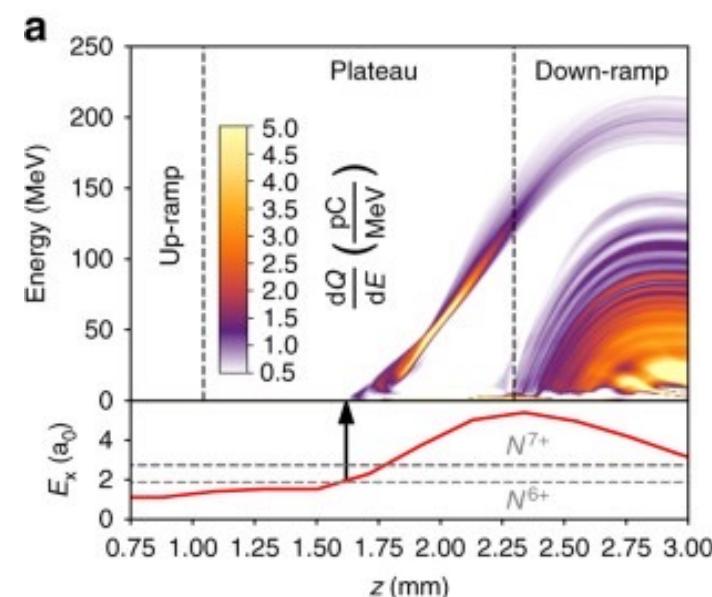
summit

ISAAČ PIConGPU

Electron acceleration with laser wakefield acceleration (LWFA)

electrons ride a plasma wave and are accelerated following the laser pulse

- short pulse laser creates wake
- electrons surf wake and accelerate
- various forms of injection
 - self-injection
 - ionization injection
 - downramp injection
 - ...
- variety of further applications developed



Couperus, J.P., Pausch, R., Köhler, A. et al

Demonstration of a beam loaded nanocoulomb-class laser wakefield accelerator

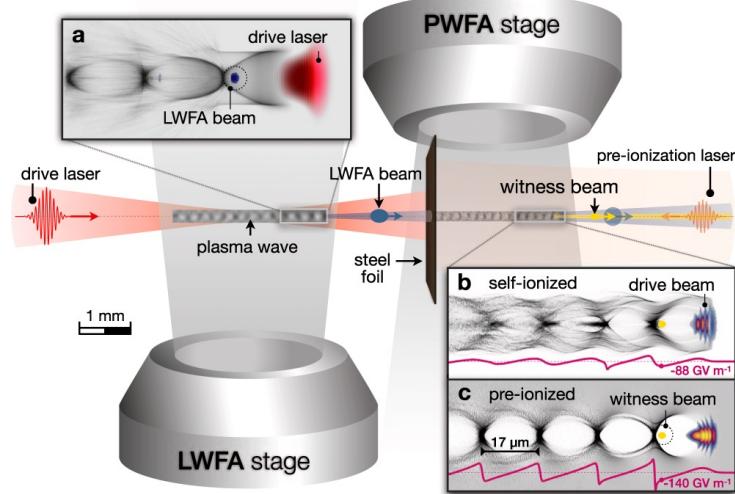
Nat Commun 8, 487 (2017)

DOI: 10.1038/s41467-017-00592-7

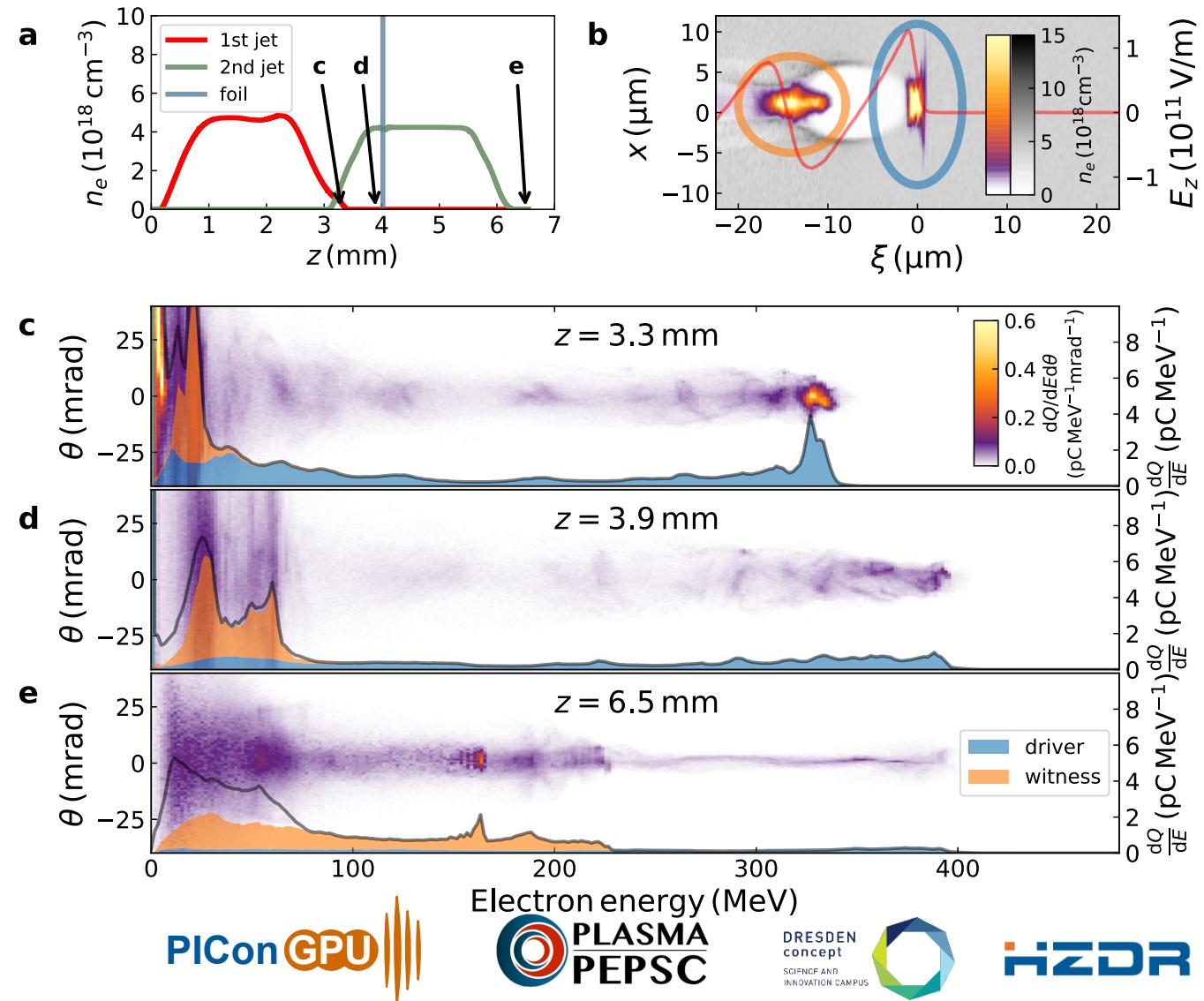
Using these unprecedented beams for first table-top PWFA ...

First start-to-end simulation of LWFA|PWFA hybrid accelerator

- first compact plasma wakefield accelerator (PWFA)
- first start-to-end hybrid simulations
- simulation enabled understanding the origin of the witness beam

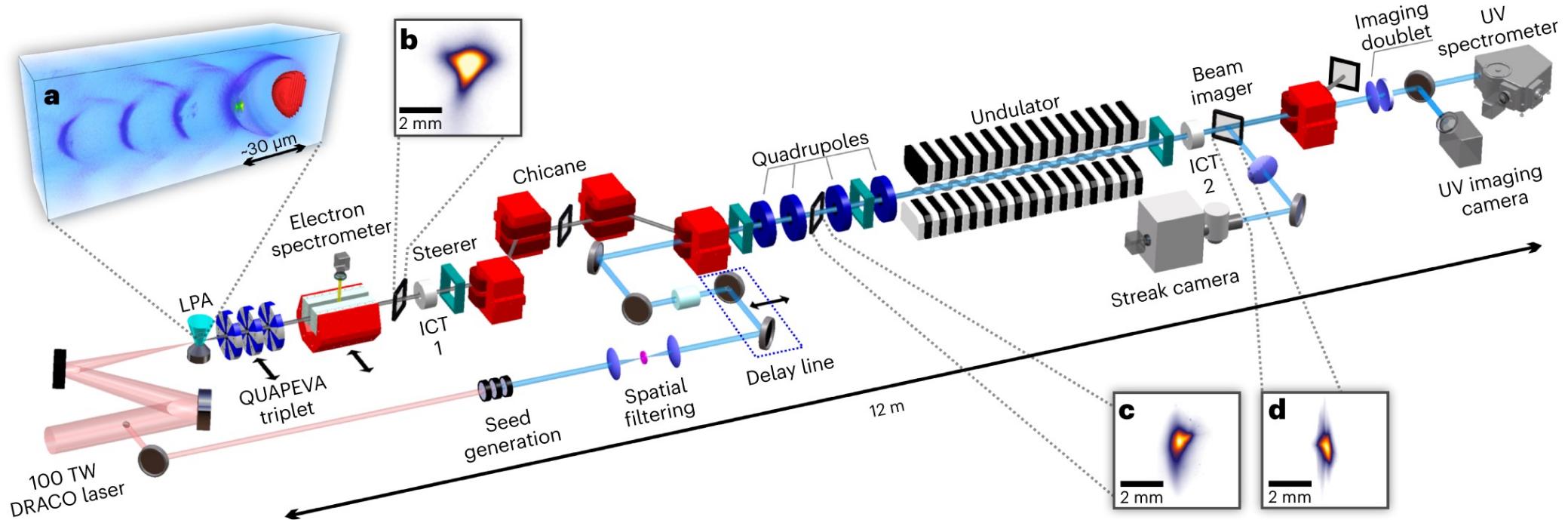


Kurz, T., et al. *Nature Commun* 12, 2895 (2021)



... and for the first seeded FEL driven by LWFA electrons

Going beyond standard FEL simulations by using particle-in-cell simulations as input

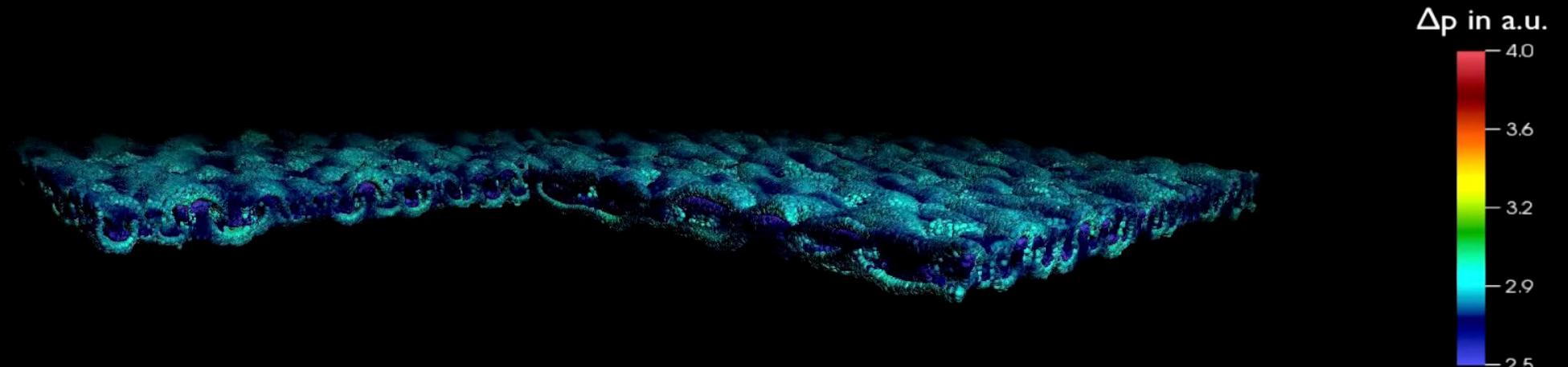


Labat, M., et al. *Nature Photon.* 17, 150–156 (2023)

Radiation from interstellar jets (KHI)

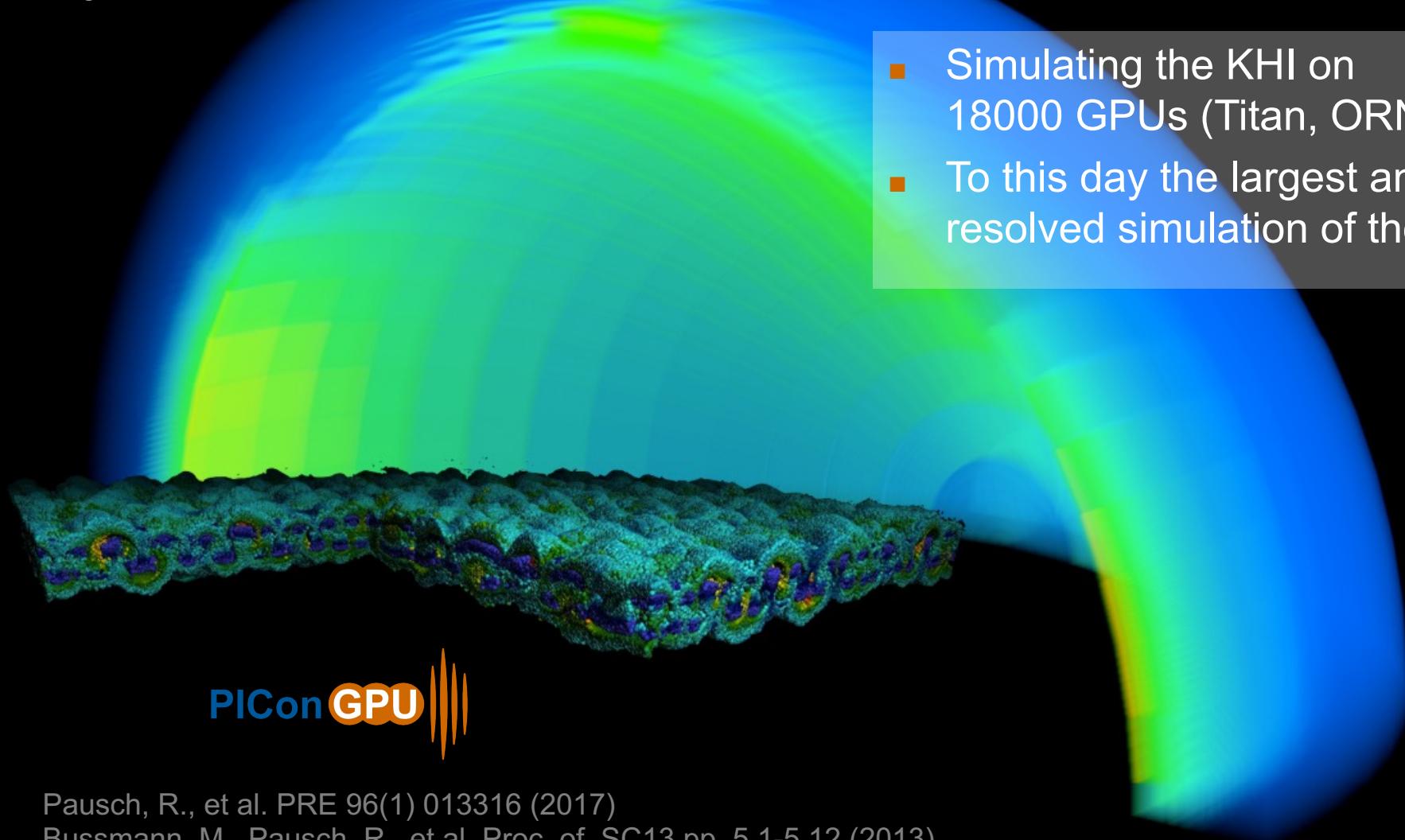
Time: $27 \omega_{pe}^{-1}$

Correlating changes in polarization to the onset of plasma instabilities



Radiation from interstellar jets (KHI)

Correlating changes in polarization to the onset of plasma instabilities



PIConGPU

Pausch, R., et al. PRE 96(1) 013316 (2017)

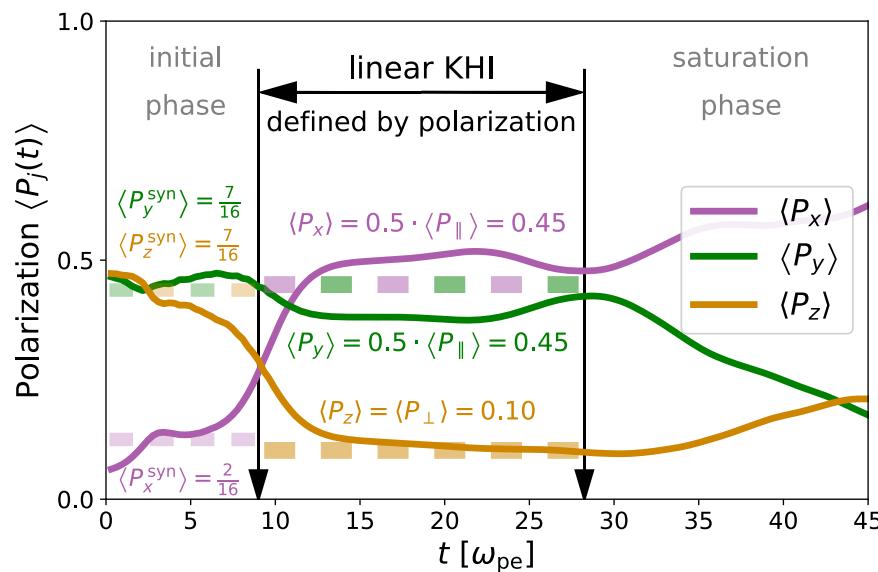
Bussmann, M., Pausch, R., et al. Proc. of. SC13 pp. 5.1-5.12 (2013)

Huebl, A., Pausch, R. et al. IEEE Trans. on Plas. Sci., 42(10), 2638–2639 (2014)

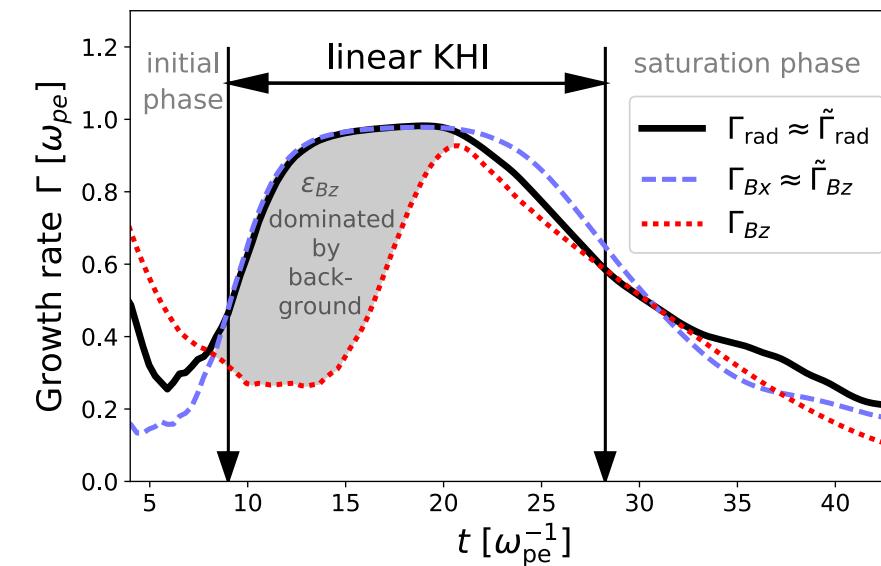
Radiation from interstellar jets (KHI)

Correlating changes in polarization to the onset of plasma instabilities

time evolution of radiation shows characteristic polarization



The growth of the radiation intensity allows determining the growth of the instability.



Pausch, R., et al. PRE 96(1) 013316 (2017)

Bussmann, M., Pausch, R., et al. Proc. of SC13 pp. 5.1-5.12 (2013)

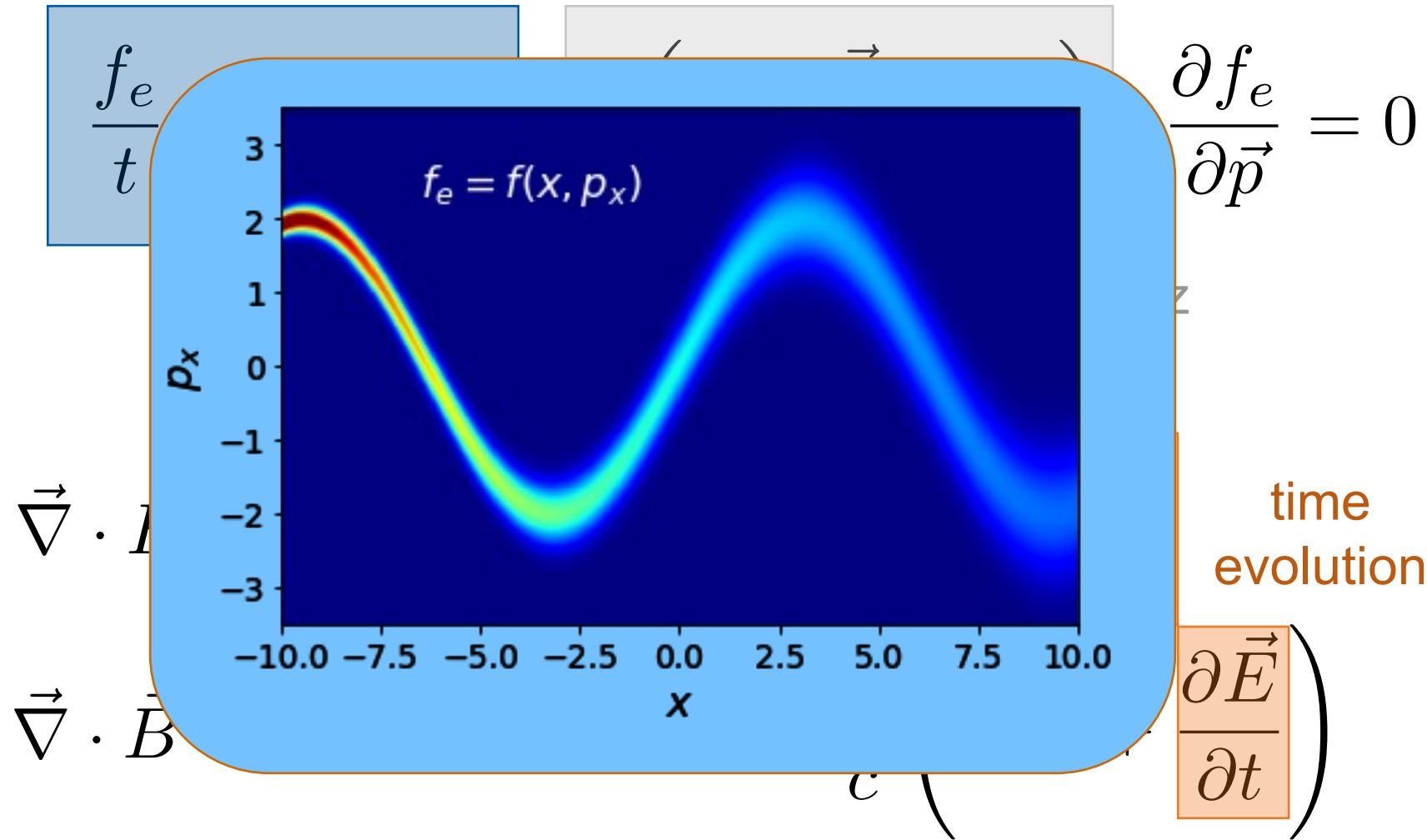
Huebl, A., Pausch, R. et al. IEEE Trans. on Plas. Sci., 42(10), 2638–2639 (2014)

Particle-in-cell codes

An introduction on how PIConGPU works

Why do we need particle-in-cell simulations?

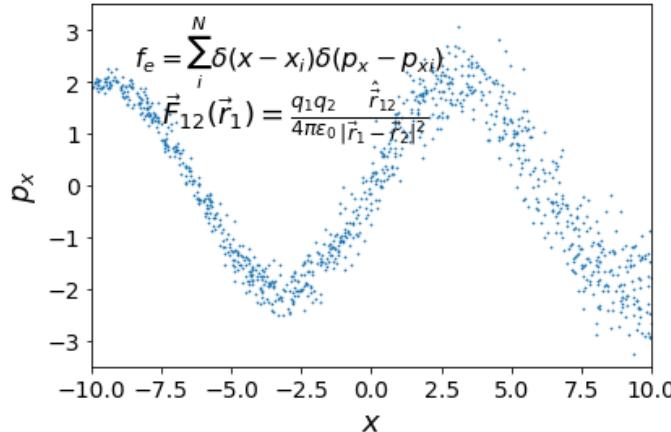
The Vlasov–Maxwell system of equations



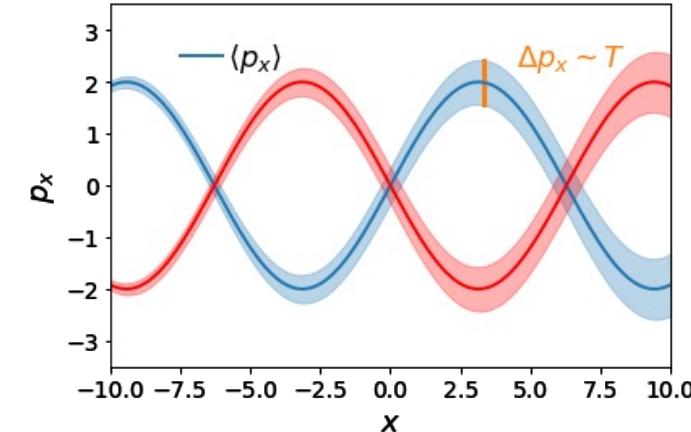
How does a particle-in-cell code work?

How could one solve the Vlasov-Maxwell equations?

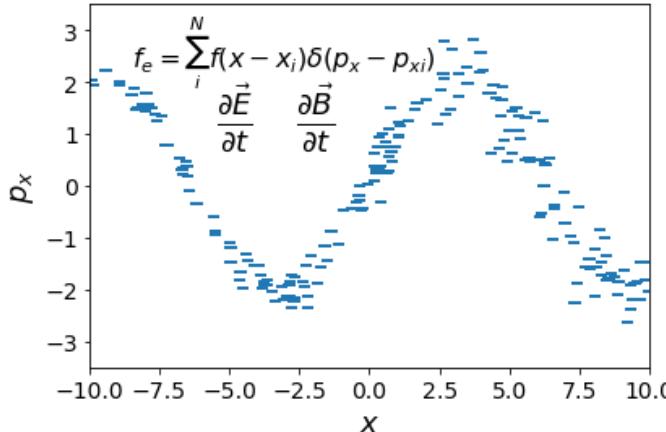
molecular dynamics (MD)



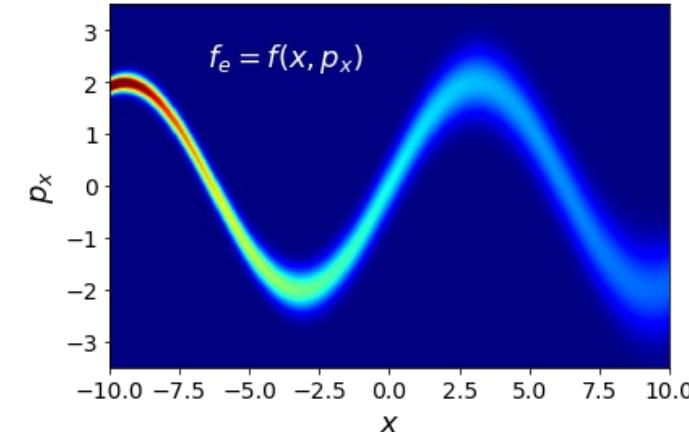
magneto-hydrodynamics (MHD)



particle-in-cell (PIC)



Vlasov-solver



How does a particle-in-cell code work?

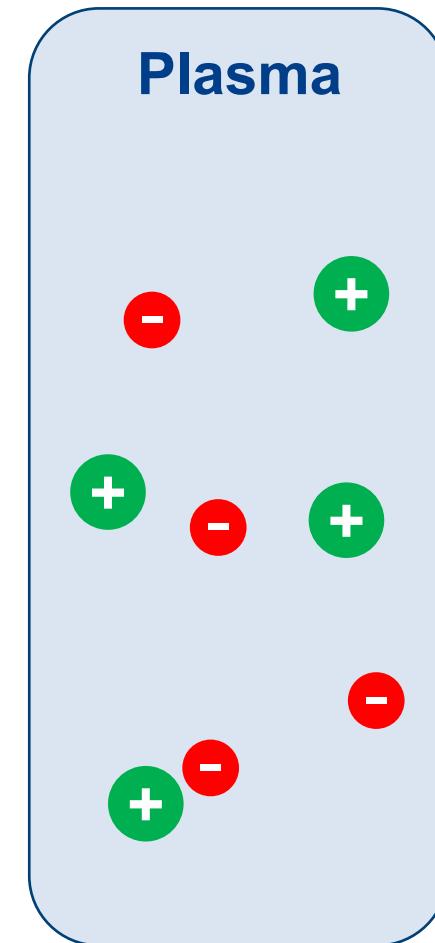
the particle-in-cell method (time evolution)

Freely moving charges generate electromagnetic fields.

Maxwell equations

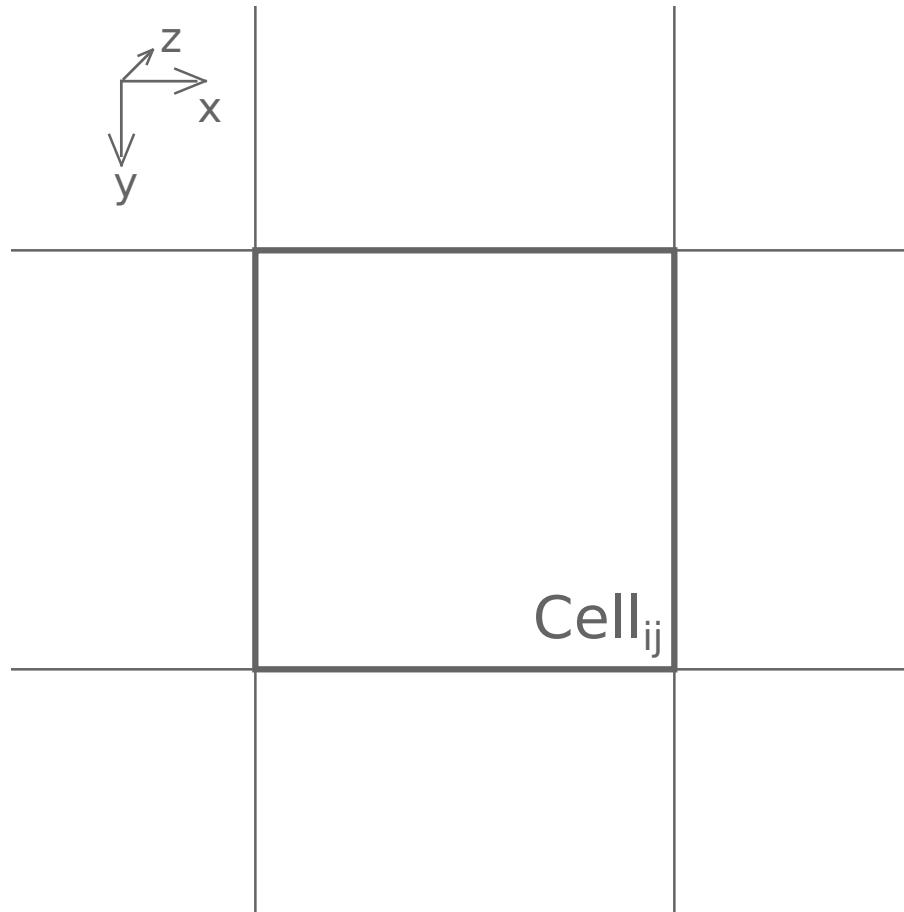
These fields cause forces to act on the charged particles.

Lorentz force



How does a particle-in-cell code work?

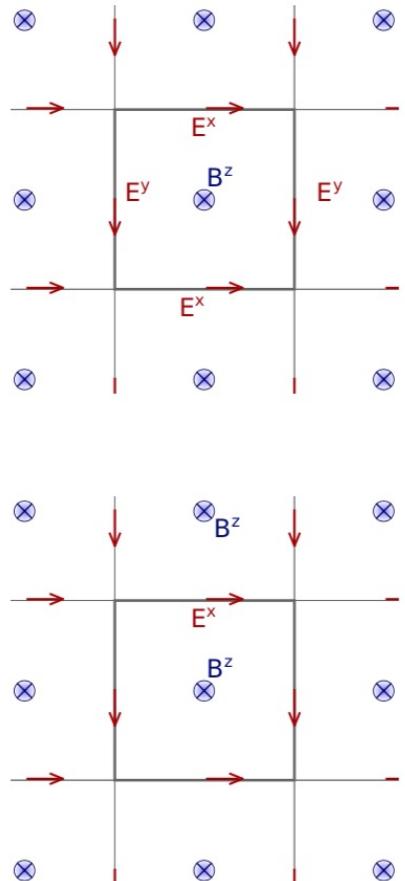
Discretizing the Maxwell equations on a grid



- Finite Difference Time Domain (FDTD) method
- common algorithm:
Yee-solver
- defines Yee-grid
- staggered grid
for higher
numeric accuracy

How does a particle-in-cell code work?

Discretizing the Maxwell equations on a grid



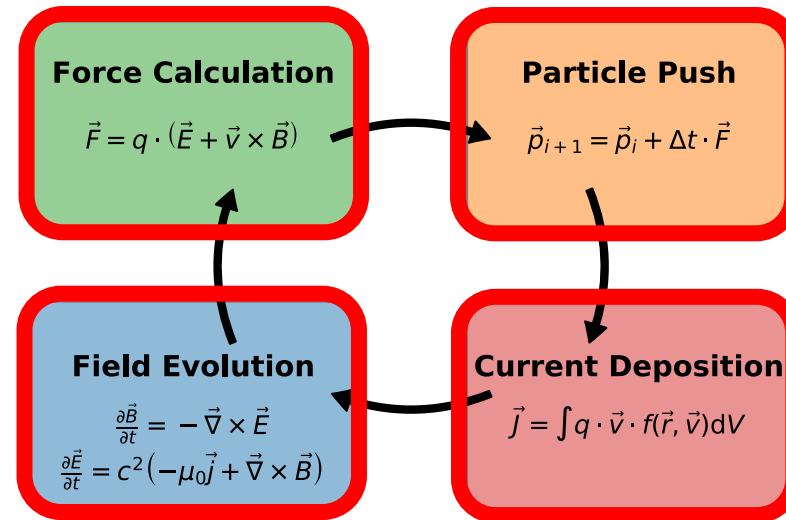
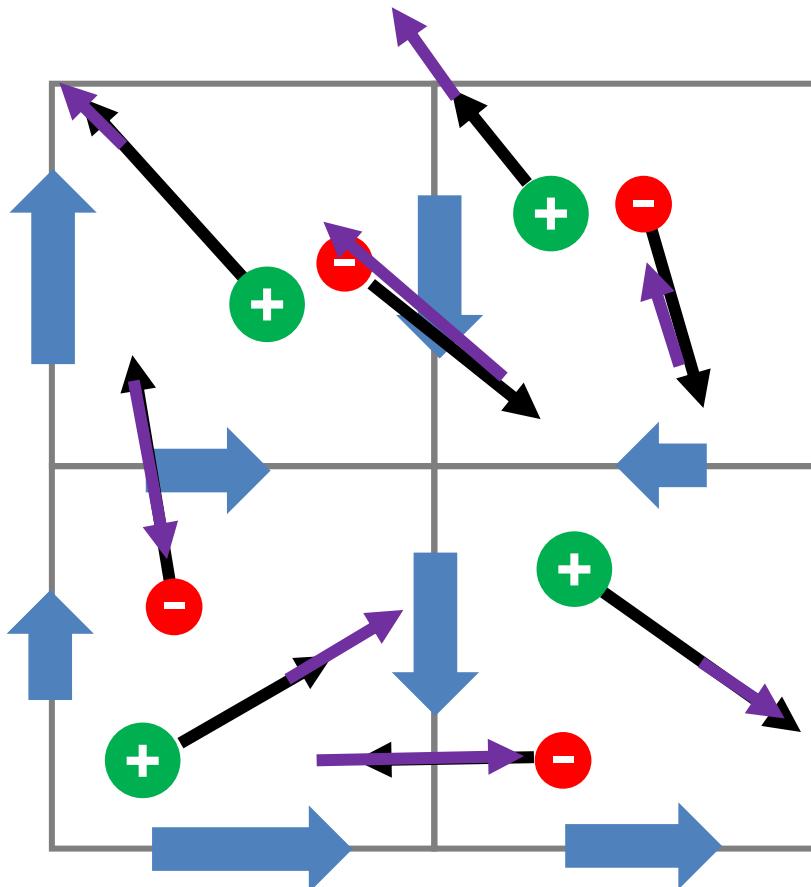
The diagram shows three Yee stencils on a 2D grid. The first stencil (top) shows the time derivative of the magnetic field $\frac{\partial}{\partial t} \mathbf{B}$ at a central node, with components E^x and E^y to the right and B^z above. The second stencil (middle) shows the time derivative of the vertical component of the magnetic field $\frac{\partial}{\partial t} B_{ij}^z$ at a central node, with components E^y and E^x to the left and right respectively. The third stencil (bottom) shows the time derivative of the horizontal component of the electric field $\frac{\partial}{\partial t} E_{ij}^x$ at a central node, with components B^z and E^x above and below.

$$\frac{\partial}{\partial t} \mathbf{B} = - \nabla \times \mathbf{E} \Rightarrow - \nabla^+ \times \mathbf{E}$$
$$\frac{\partial}{\partial t} B_{ij}^z = - (E_{i+1j}^y - E_{ij}^y) / \Delta x + (E_{ij+1}^x - E_{ij}^x) / \Delta y$$
$$\frac{\partial}{\partial t} \mathbf{E} = + \nabla \times \mathbf{B} \Rightarrow + \nabla^- \times \mathbf{B}$$
$$\frac{\partial}{\partial t} E_{ij}^x = + (B_{ij}^z - B_{ij-1}^z) / \Delta y$$

- Yee stencils
- better schemes exists
- Lehe, AOFDT, CKC, spectral solver, ...
- interaction with current highly important

How does a particle-in-cell code work?

the PIC cycle



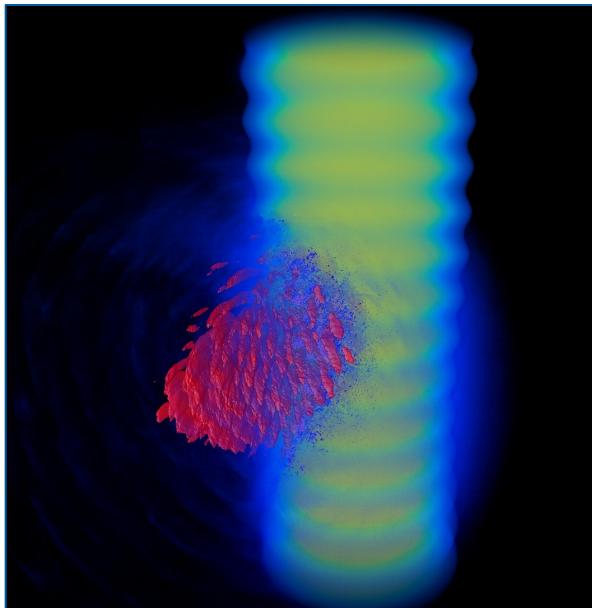
- one complete cycle corresponds to one time step.
- repeated cycles allow simulating longer time durations.
- local operations only: well suited for parallelization.

Real experiments need really large simulations

The need for large scale simulations equals the need for fast simulations

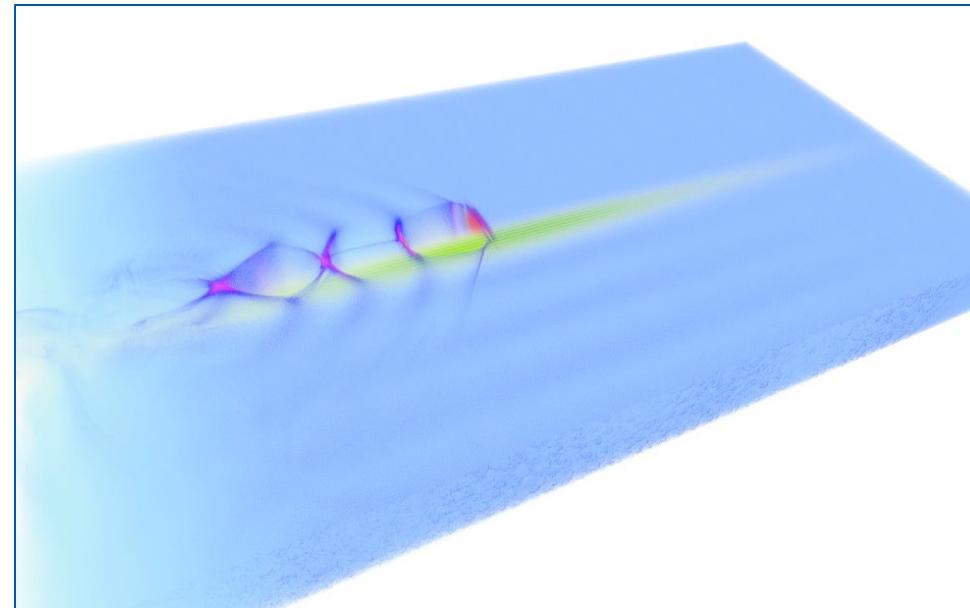
high resolution

cryogenic hydrogen jet



3D
(no symmetry)

electron acceleration with probe pulse

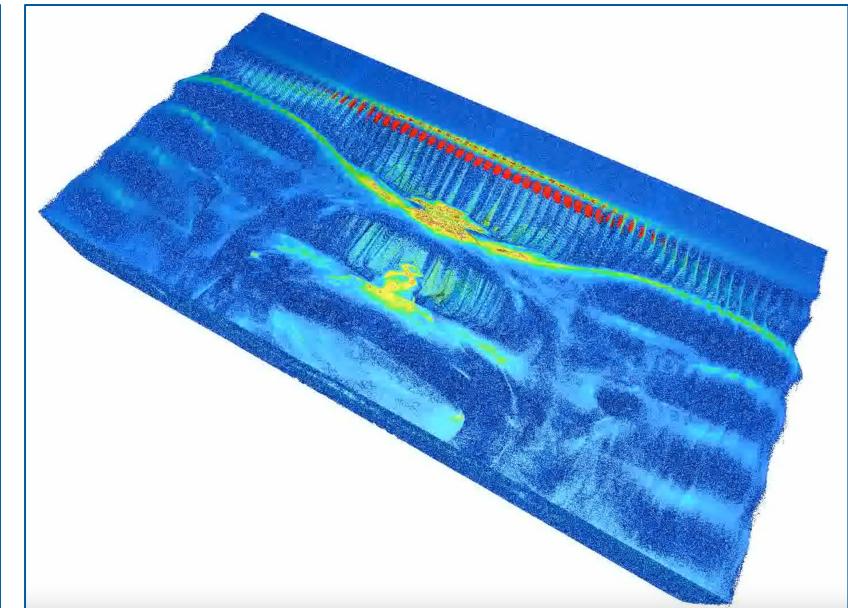


long duration

TB of data

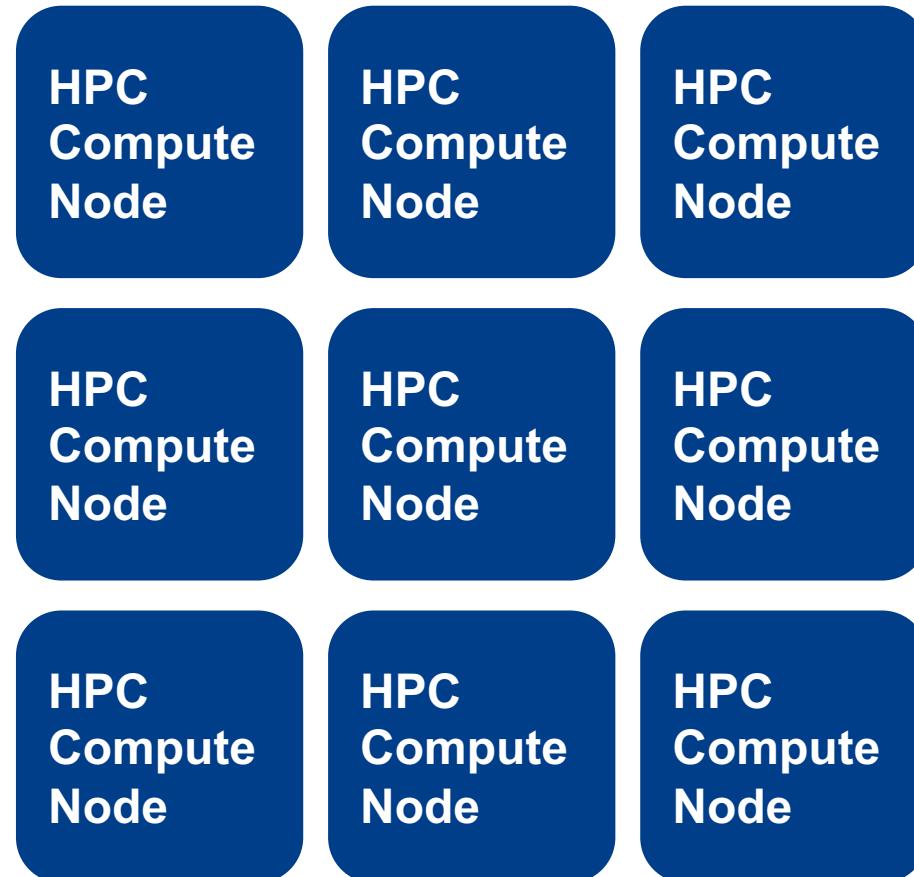
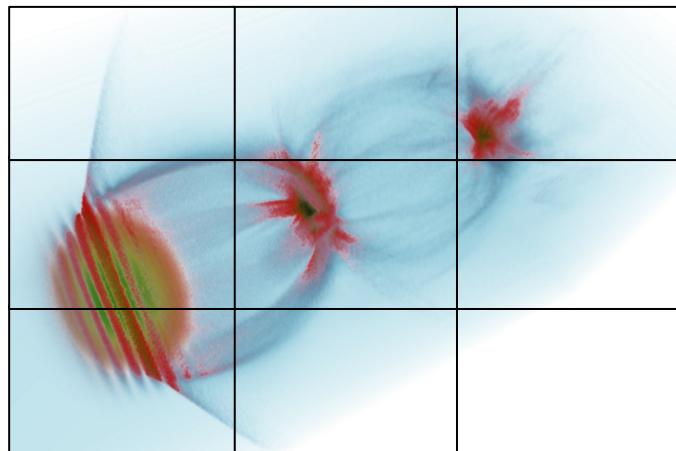
complex models

TWEAC



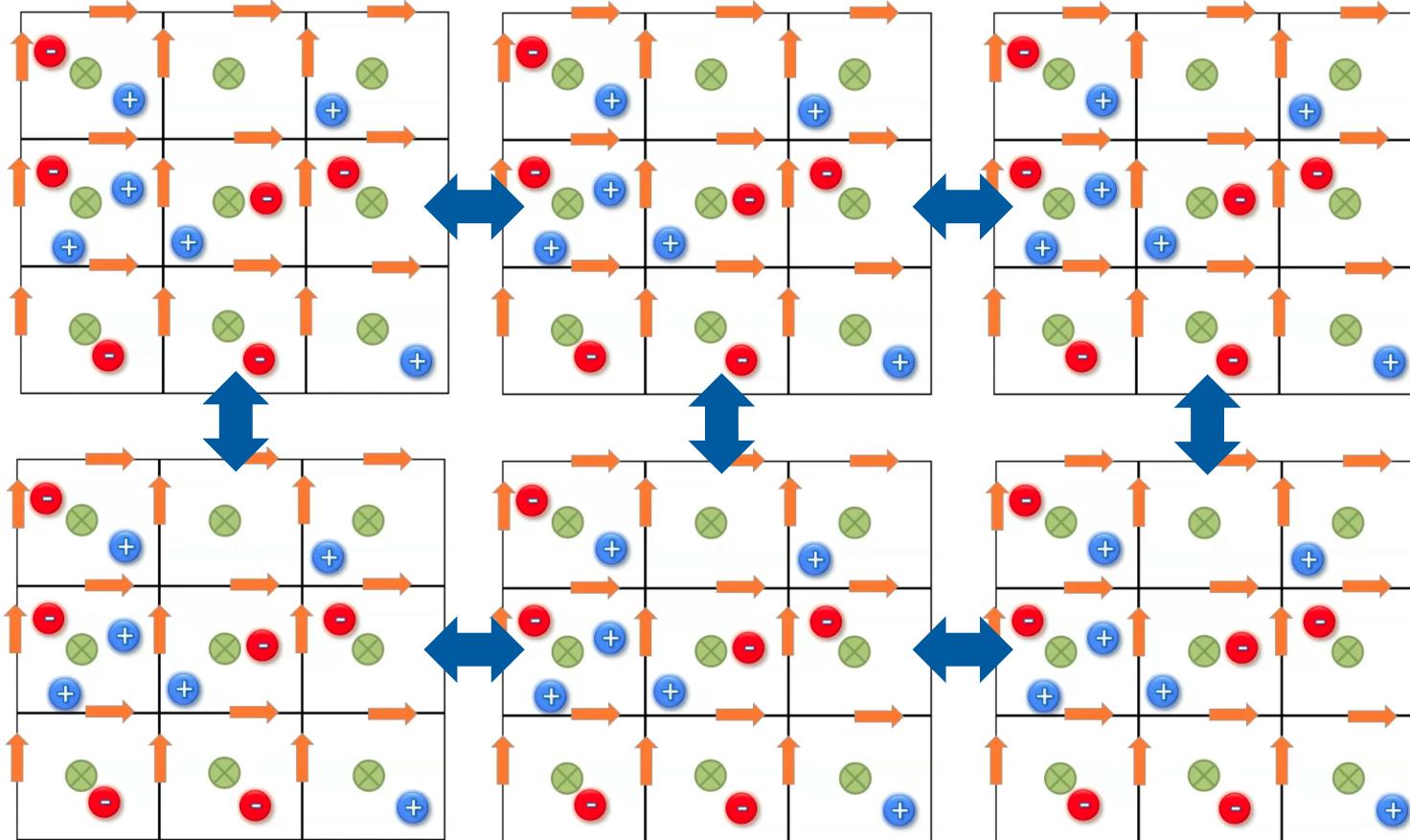
How does PIConGPU reaches its fast performance?

Domain decomposition – distributing spatial subregions into different computers



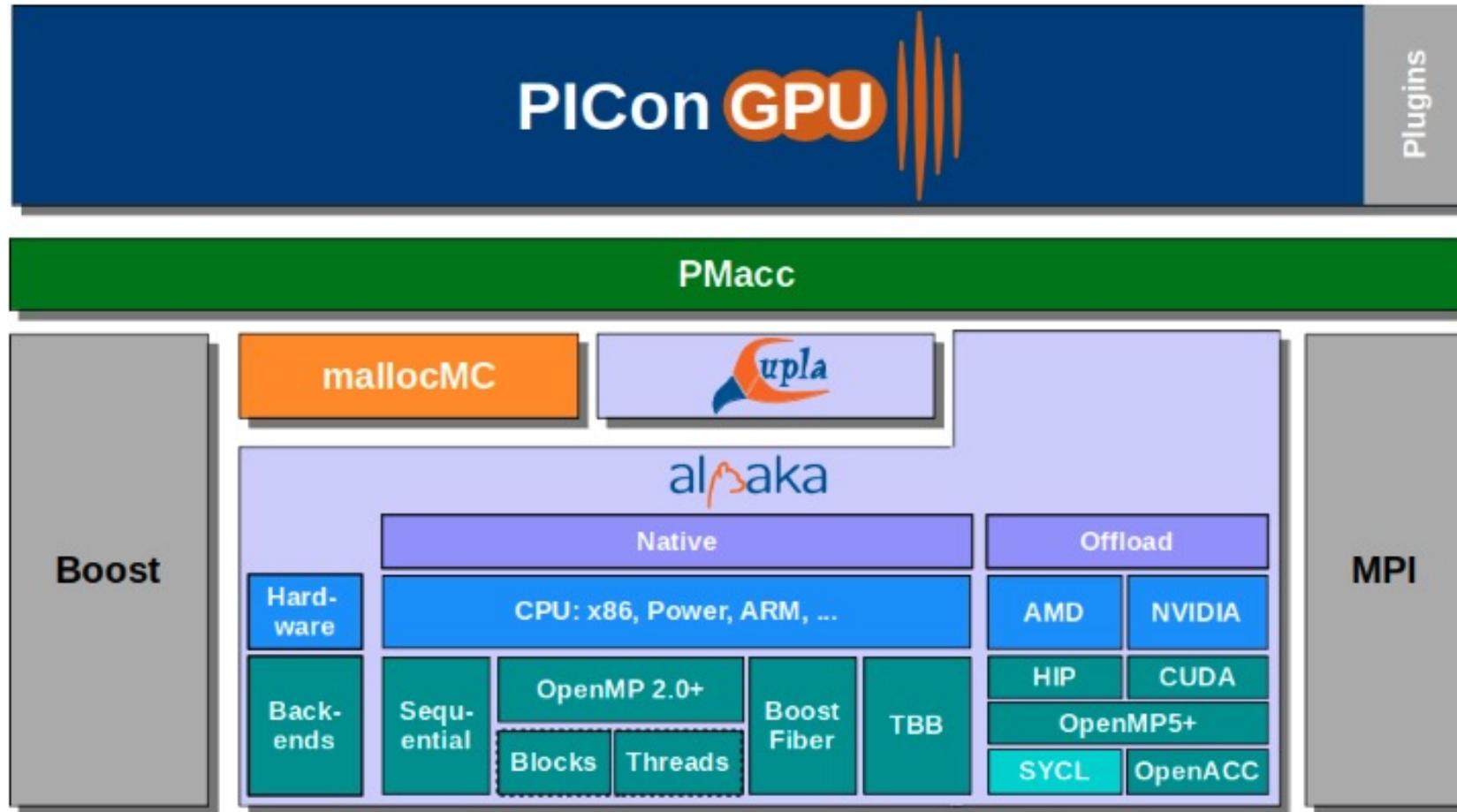
How does PIConGPU reaches its fast performance?

Domain decomposition – distributing spatial subregions into different computers



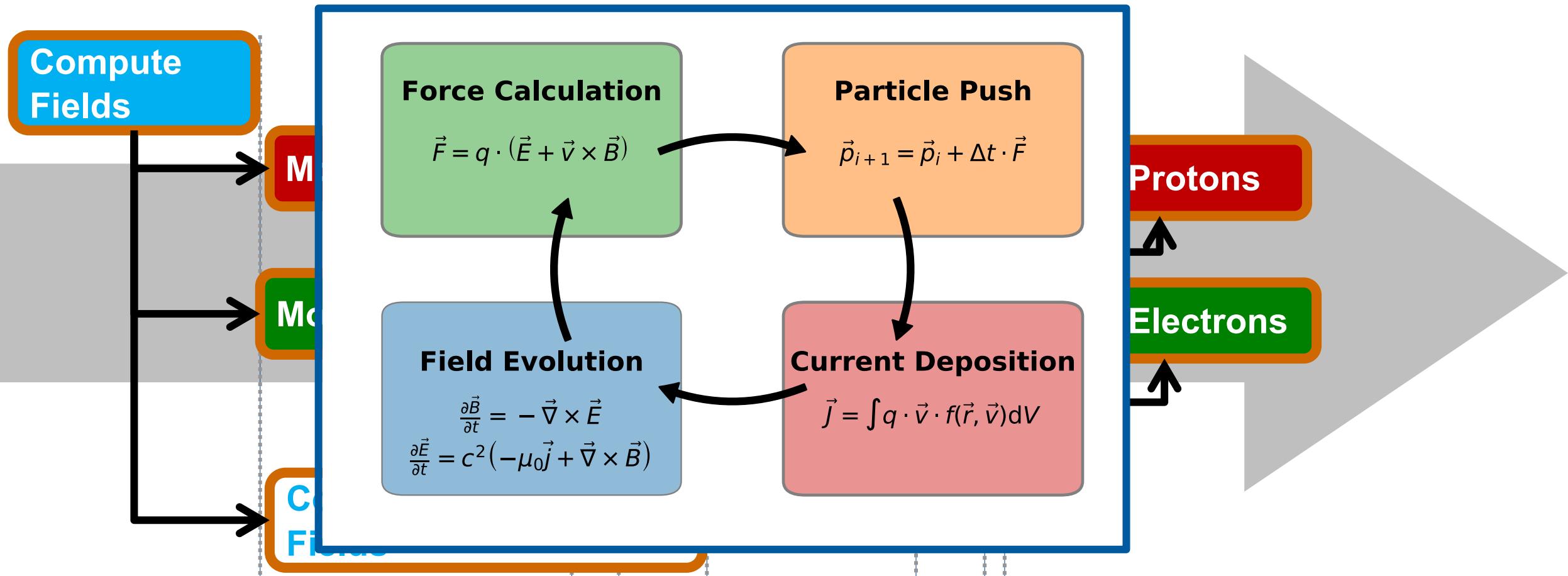
How does PIConGPU reaches its fast performance?

Abstractions of data structures and algorithms in libPMacc



How does PIConGPU reaches its fast performance?

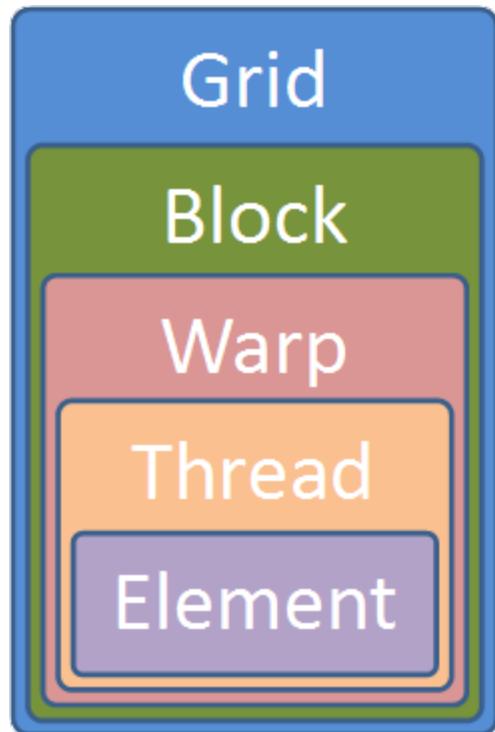
Handling data exchange – asynchronous communication and execution for the rescue



Quick recap: alpaka

How does the alpaka hardware abstraction work?

Data parallelism in modern hardware:



- **Grid** whole parallel task
- **Block** fully independent part of the grid
- **Warp** group of synchronous threads
- **Threads** executed concurrently
- **Elements** sub-thread, sequential lock-step



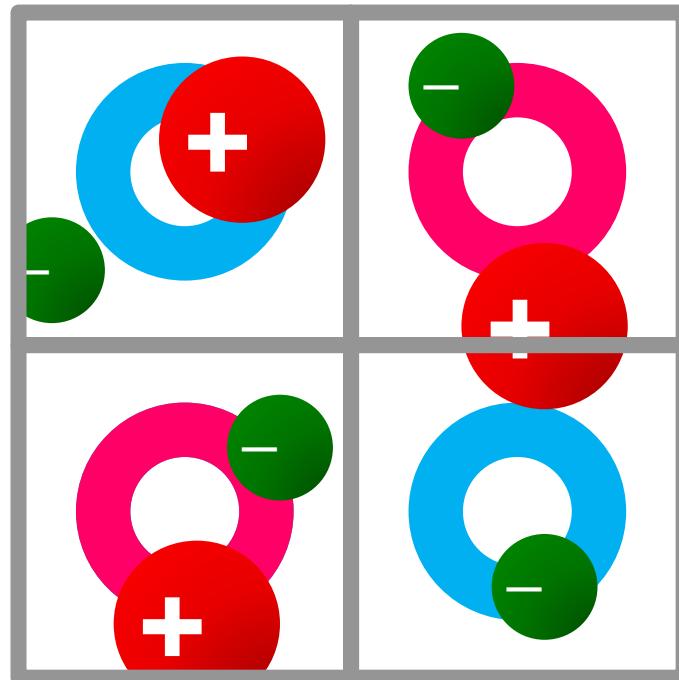
Zenker E. et al. (2016) IEEE Xpress DOI:10.1109/IPDPSW.2016.50

Zenker E. et al. ISC (2016) ISC High Performance DOI:10.1007/978-3-319-46079-6_21

Matthes A. et al., (2017) ISC High Performance (pp 496-514) DOI: 10.1007/978-3-319-67630-2_2

How does PIConGPU reaches its fast performance?

Domain decomposition – fields and particles

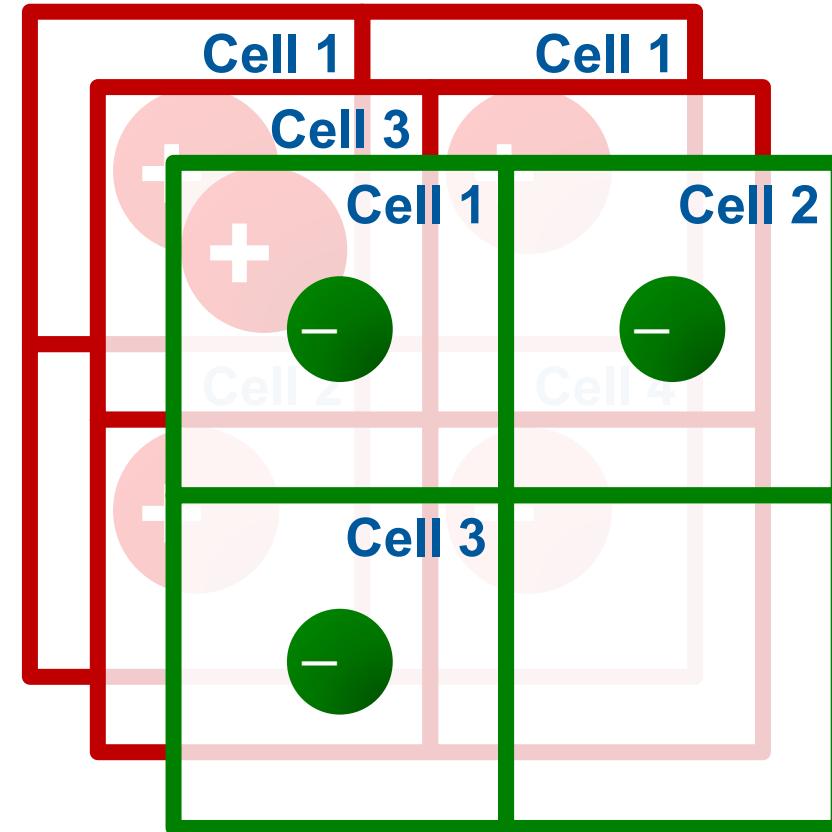
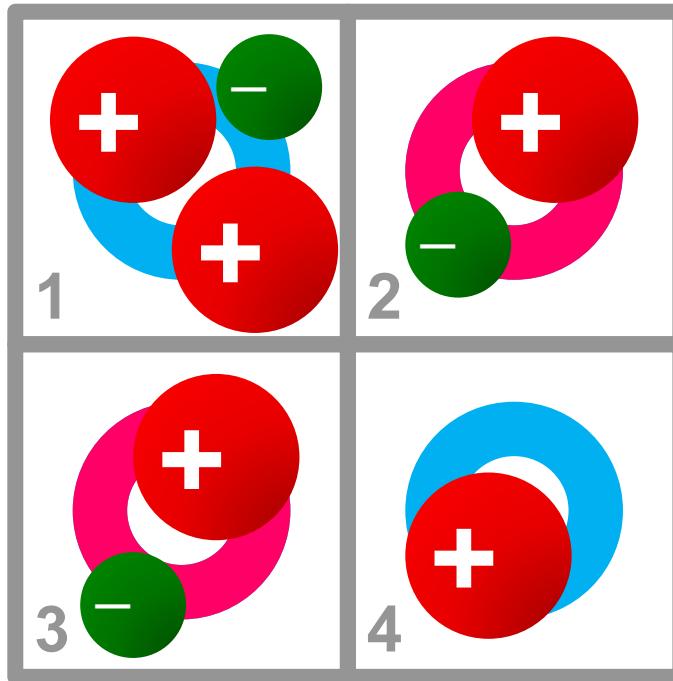


Field Domain

Particle Domain

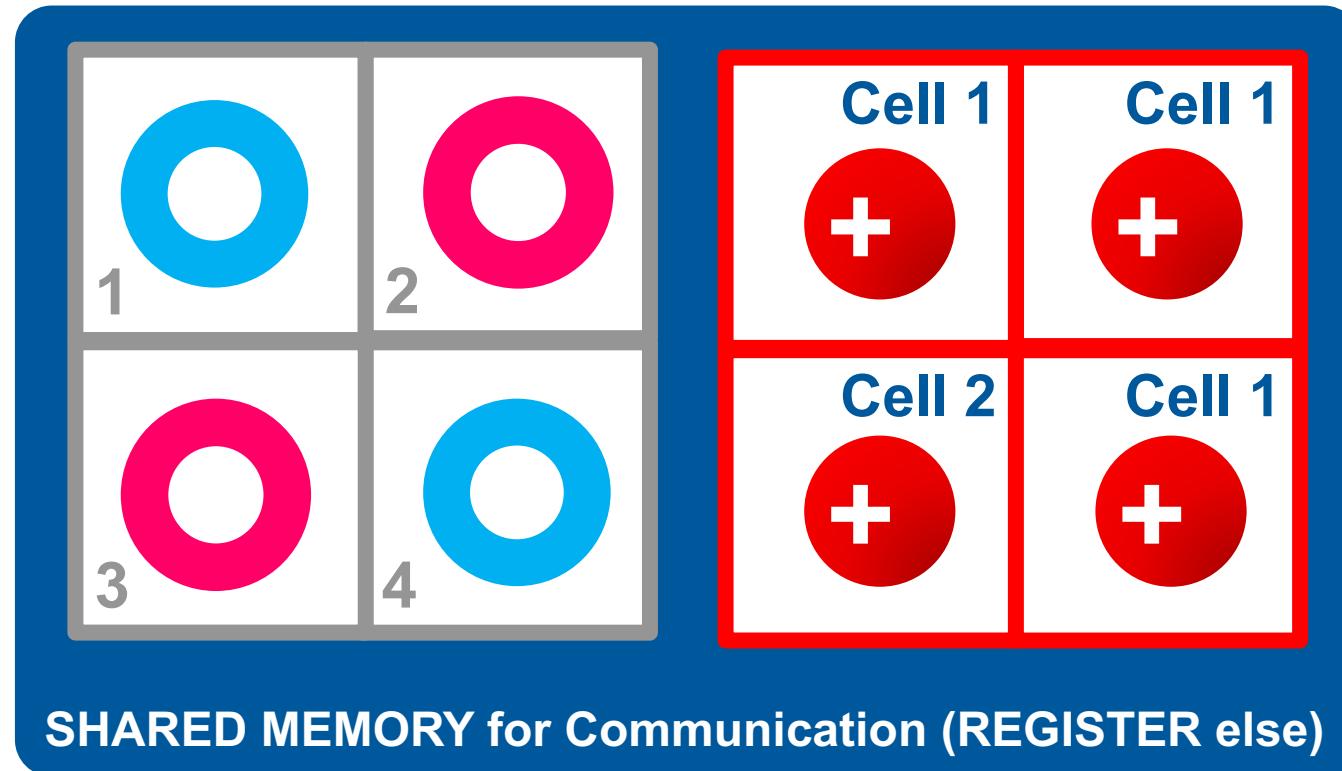
How does PIConGPU reaches its fast performance?

Domain decomposition – fields and particles – adding particles



How does PIConGPU reaches its fast performance?

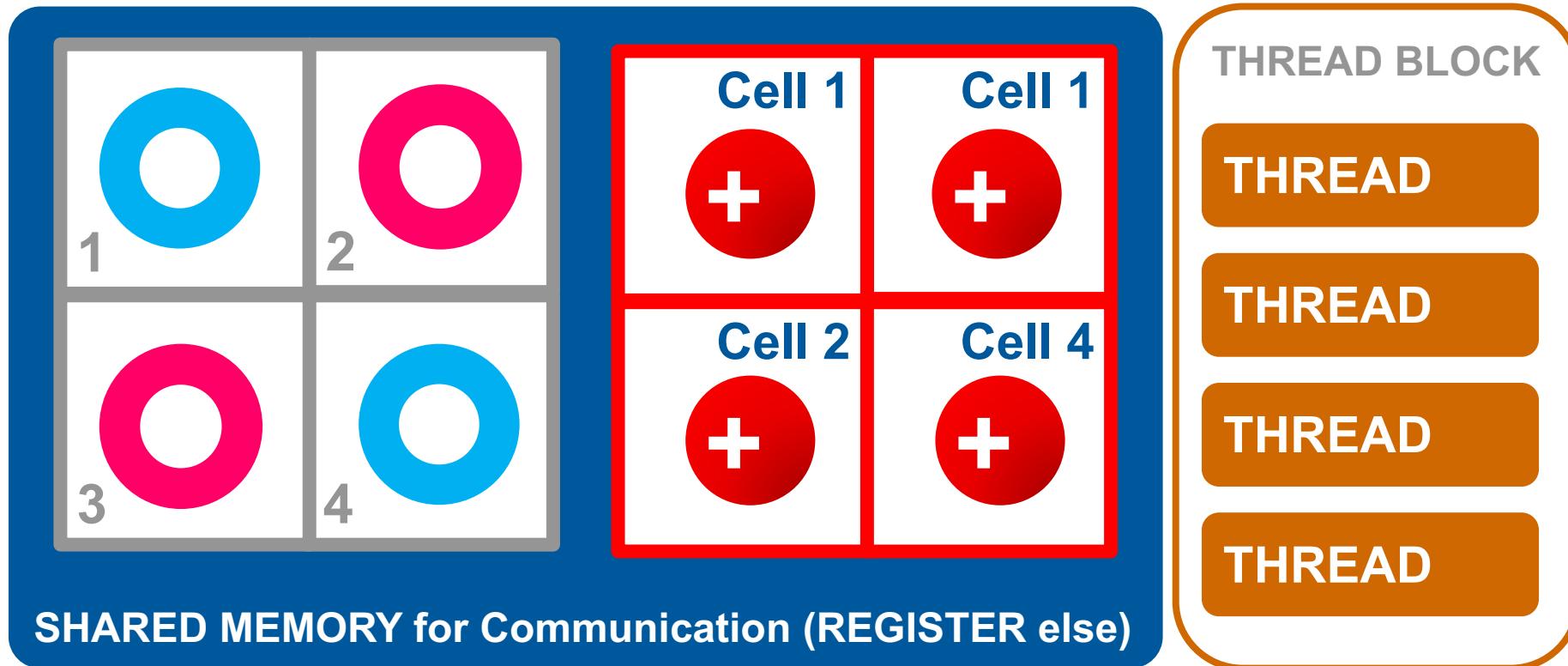
Domain decomposition – using register/shared memory for efficient data handling



How does PIConGPU reaches its fast performance?

Domain decomposition – using register/shared memory for efficient data handling

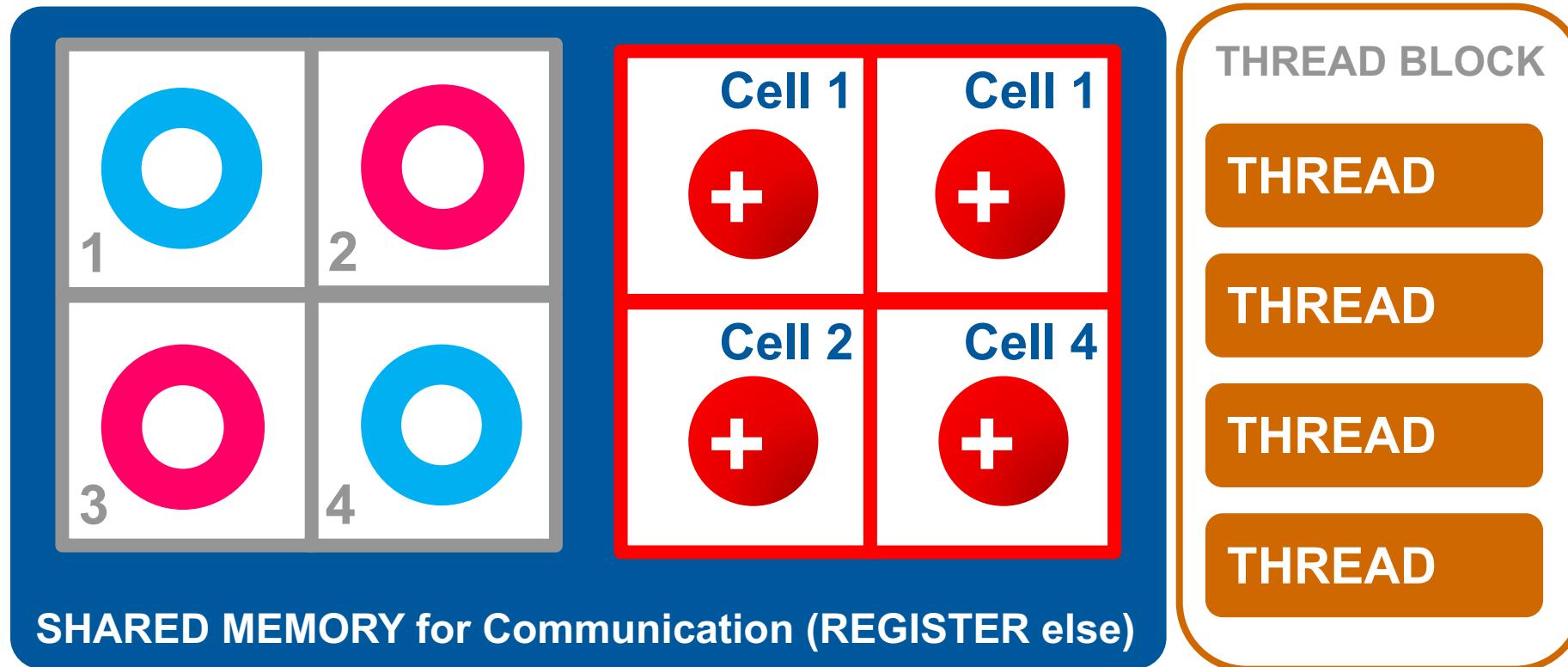
Cell-wise Threading



How does PIConGPU reaches its fast performance?

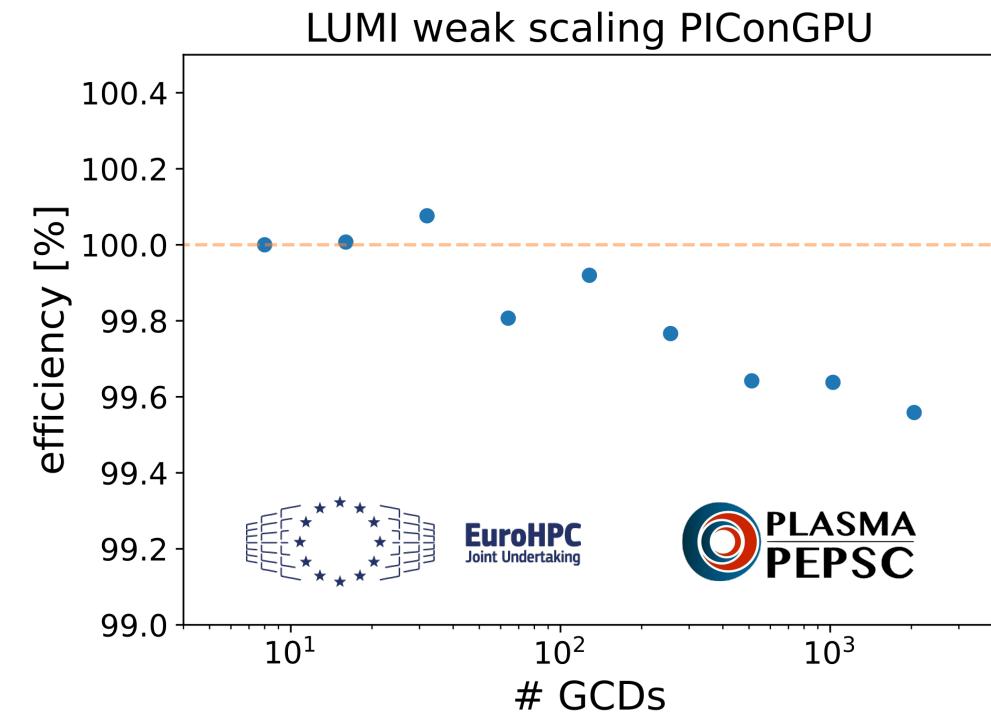
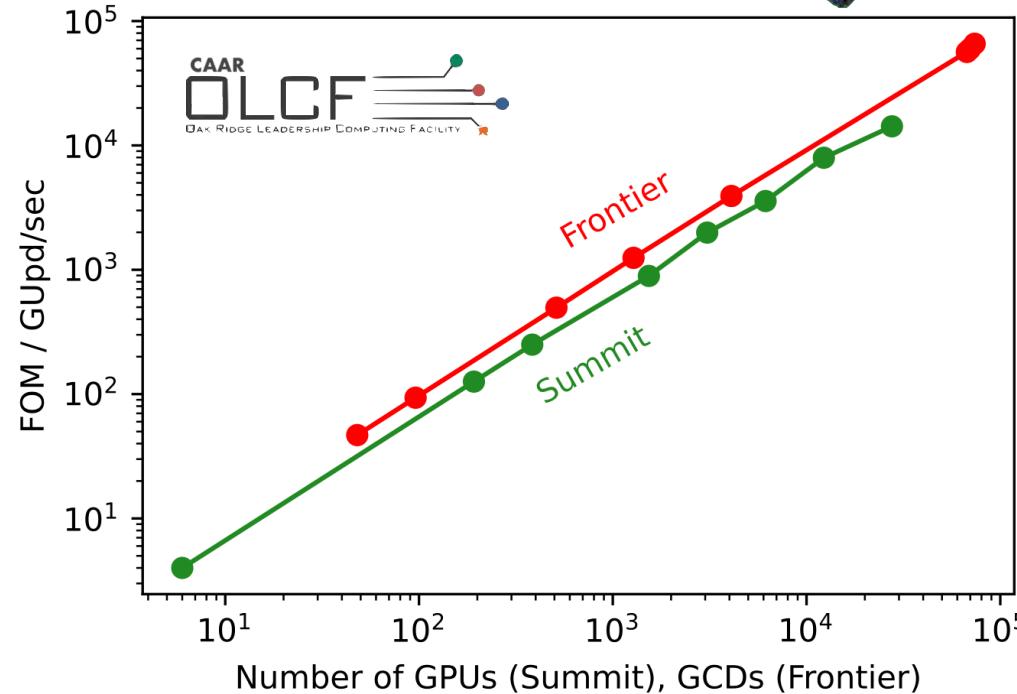
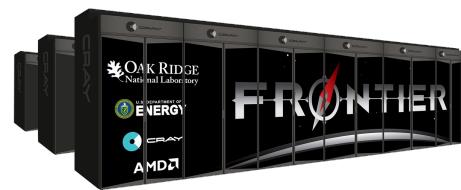
Domain decomposition – using register/shared memory for efficient data handling

Particle-wise Threading



How fast is PIConGPU's performance?

Scaling runs on the largest HPC systems in the world

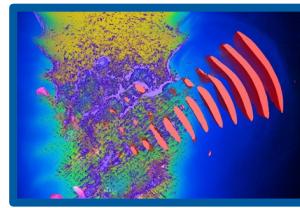
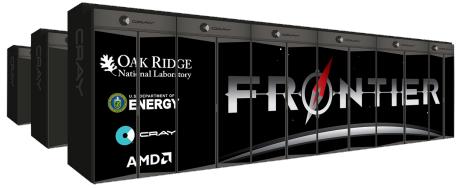


Fast, Faster, File-system full

Why in-situ data analysis and models are the future

What to do with this high performance?

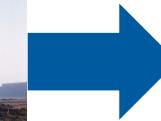
Going beyond “just” large simulation: complex in-situ data analysis



128 nodes = 1024 GPUs
with 64 GB RAM each
at >10 Hz iteration rate

0.6 PetaByte / s
0.1TB/s GPU to GPU 16GB/s internode

file System: 679 PB
full in 100s – or 1000 iterations
but simulation runs 15x longer

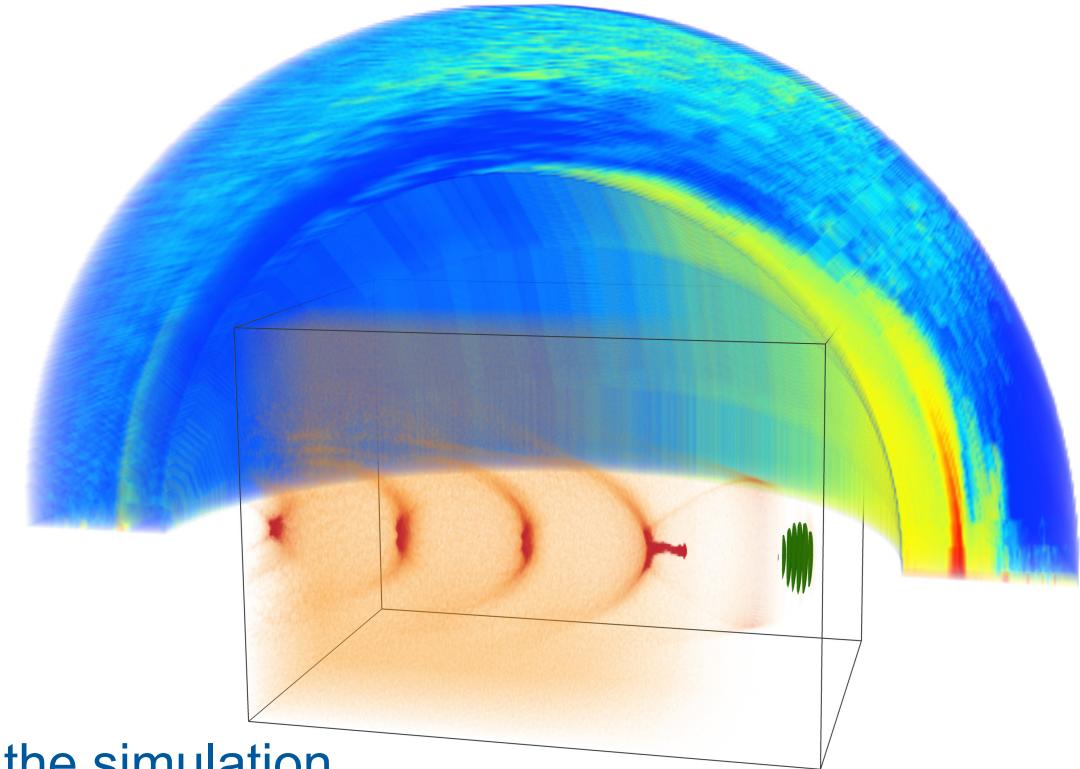


What to do with this high performance?

Going beyond “just” large simulation: complex in-situ data analysis

In-situ radiation plugin

- spectra range from **IR to x-ray**
Pausch, R., et al. NIMA 740, 250-256 (2014)
- resolves **coherent and incoherent radiation simultaneously**
Pausch, R., et al. NIMA 909, 419-422 (2018)
- includes **polarization properties**
Pausch, R., et al. PRE 96(1) 013316 (2017)
- resolves **temporal evolution of spectra**
Pausch, R., et al. Proc. of IPAC MOPRI069 (2014)
- computes radiation of all **billions of particles** in the simulation
Bussmann, M., Pausch, R., et al. Proc. of SC13 pp. 5.1-5.12 (2013)



$$\frac{d^2 W}{d\Omega d\omega} = \frac{1}{16\pi^3 \epsilon_0 c} \left| \sum_{k=1}^{N_p} \int_{-\infty}^{+\infty} q_k \cdot \frac{\vec{n} \times [(\vec{n} - \vec{\beta}_k) \times \dot{\vec{\beta}}_k]}{(1 - \vec{\beta}_k \cdot \vec{n})^2} \cdot e^{i\omega(t - \vec{n} \cdot \vec{r}_k(t)/c)} dt \right|^2$$

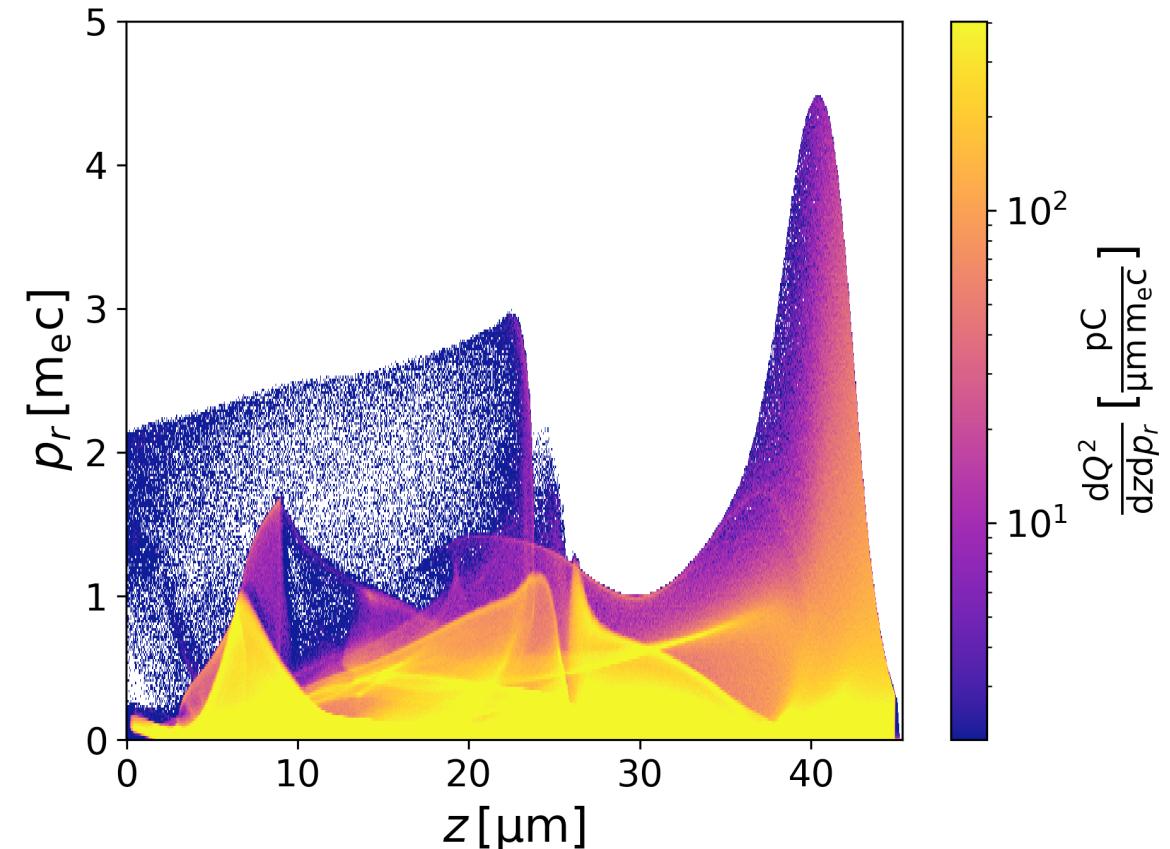
What to do with this high performance?

Going beyond “just” large simulation: complex in-situ data analysis

binning plugin

- Ultra flexible **bin quantities**
- automatically **distributed** and **parallelized**
- just a **few lines C++ code**

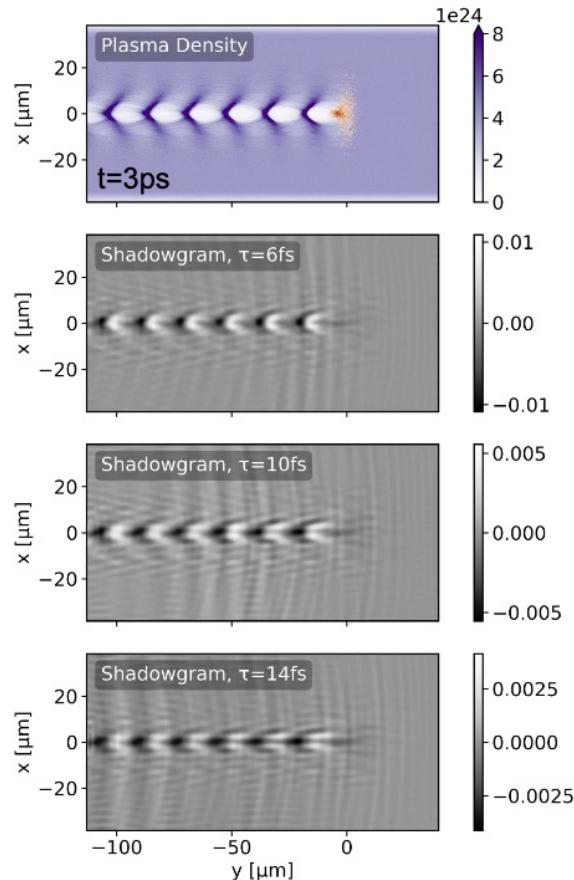
```
/// Axis 2
// Define Functor
auto getMomentumRadial
  = [] ALPAKA_FN_ACC(auto const& domainInfo,
                      auto const& worker,
                      auto const& particle) -> float_X
{
    return picongpu::math::sqrt(
        particle[momentum_][0] * particle[momentum_][0]
        + particle[momentum_][2] * particle[momentum_][2])
    / particle[weighting_];
};
```



What to do with this high performance?

Going beyond “just” large simulation: complex in-situ data analysis

Impact of Probe Duration



2D Shadowgram

shadowgram plugin

- models **full** probe laser plasma interaction
- automatically performs **Fourier optics** and **predicts experimental image**
- ideally suited to **compare to experiments**

3D PWFA Simulation

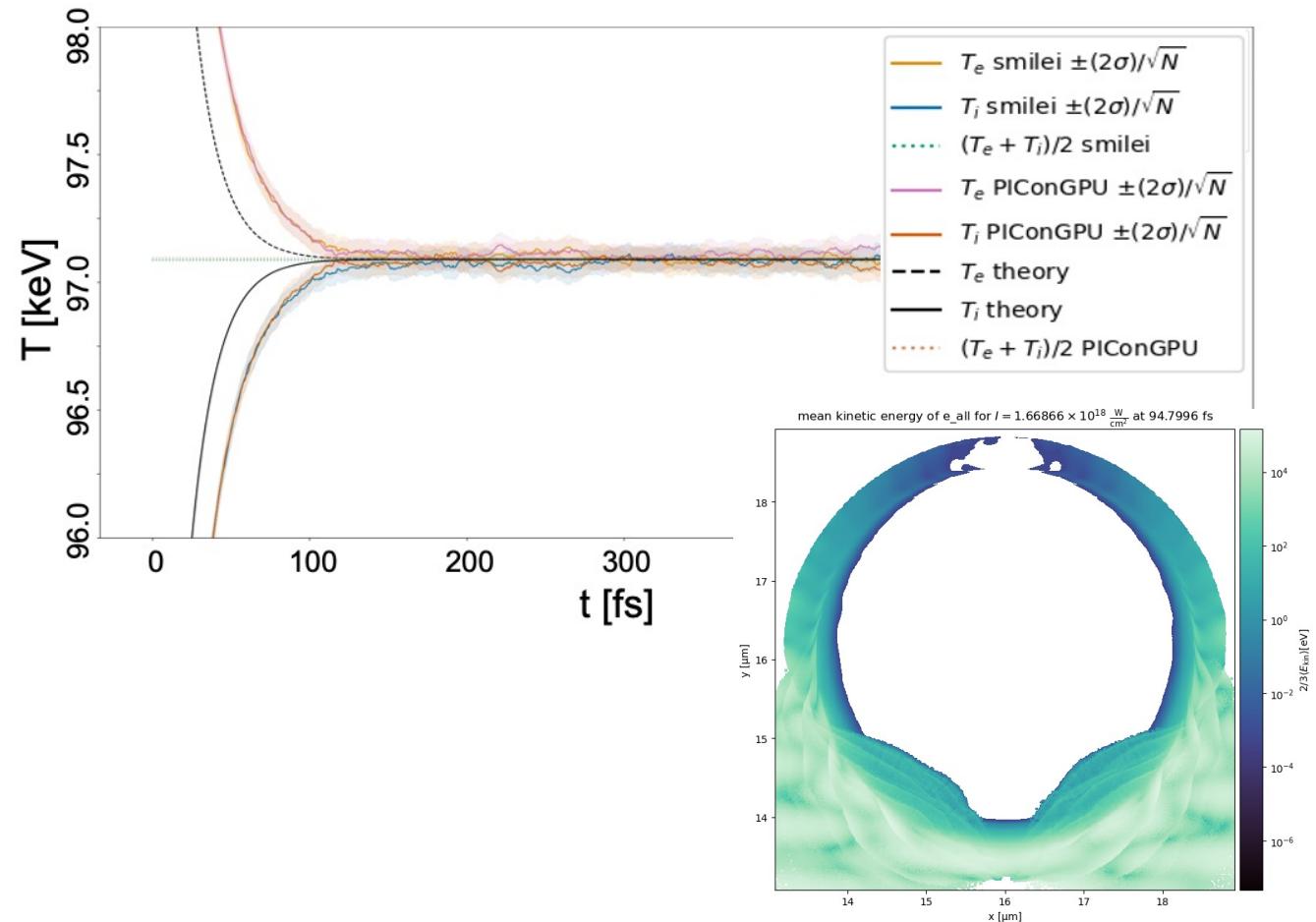
$\sim 10\mu\text{m}$

What is new in PIConGPU?

Particle-particle collisions to better model energy transfer in overdense targets

binary collisions

- needed to model **energy transfer** in overdense targets correctly
- collision frequency too much higher than PIC time step:
one collision per step with **average angle**
- handling various particle weightings via **rejection method**
(Higginson 2020)
- **code comparison** against Smilei



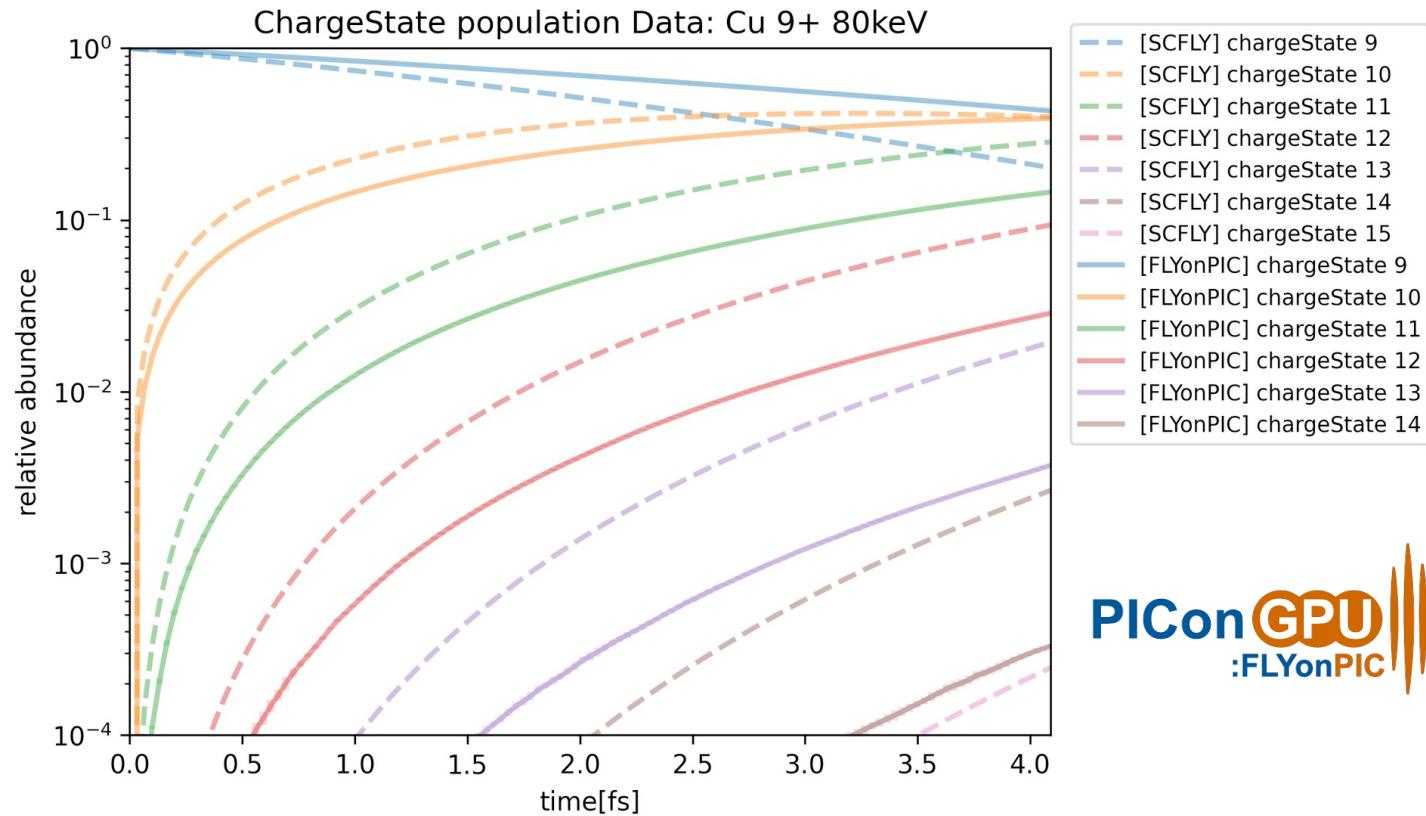
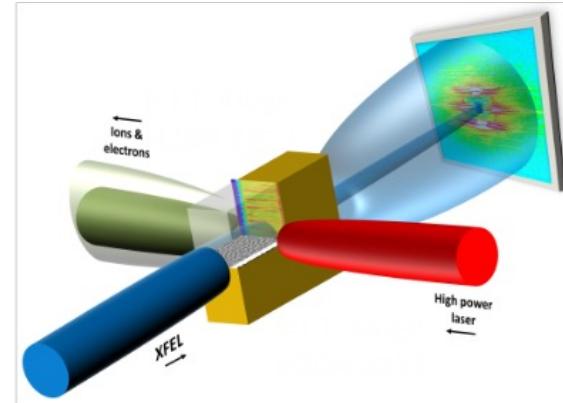
What is new in PIConGPU?

Going beyond “just” large simulation: complex in-situ data analysis

atomic physics

- going beyond BSI and ADK
- **non-thermal** atomic transitions
- average atom-state time-depended rate solver
- **code comparison** against SCFly
- Potentially experimentally validate at XFEL

HIBEF

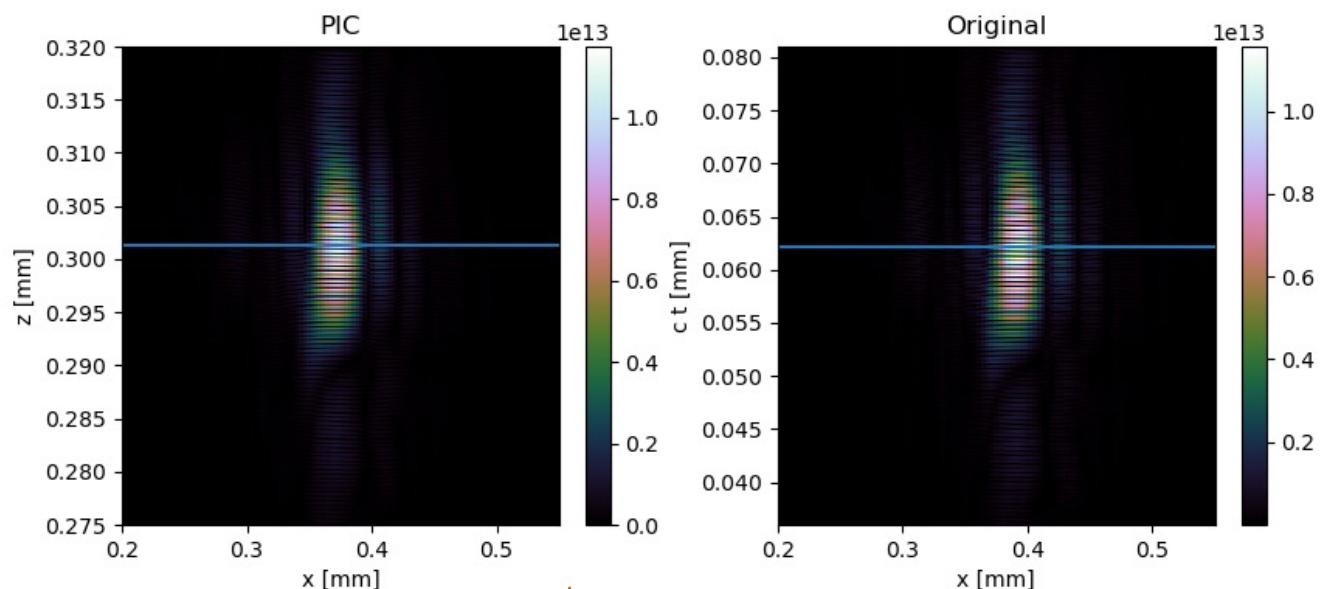
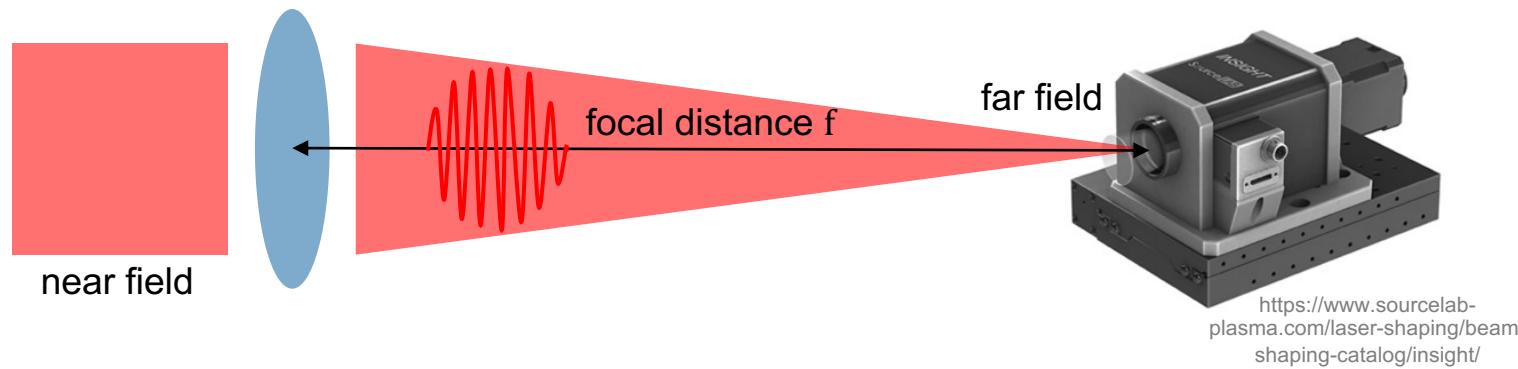


What is new in PIConGPU?

Going beyond “just” large simulation: complex in-situ data analysis

laser via openPMD

- real laser pulses are far from ideal Gaussians
- experimental laser characterization (like **INSIDE**) can measure special and temporal laser profile
- this can be included in PIConGPU via **openPMD laser input**
- also suited for other input
- suited to study influence of non-ideal (real) lasers

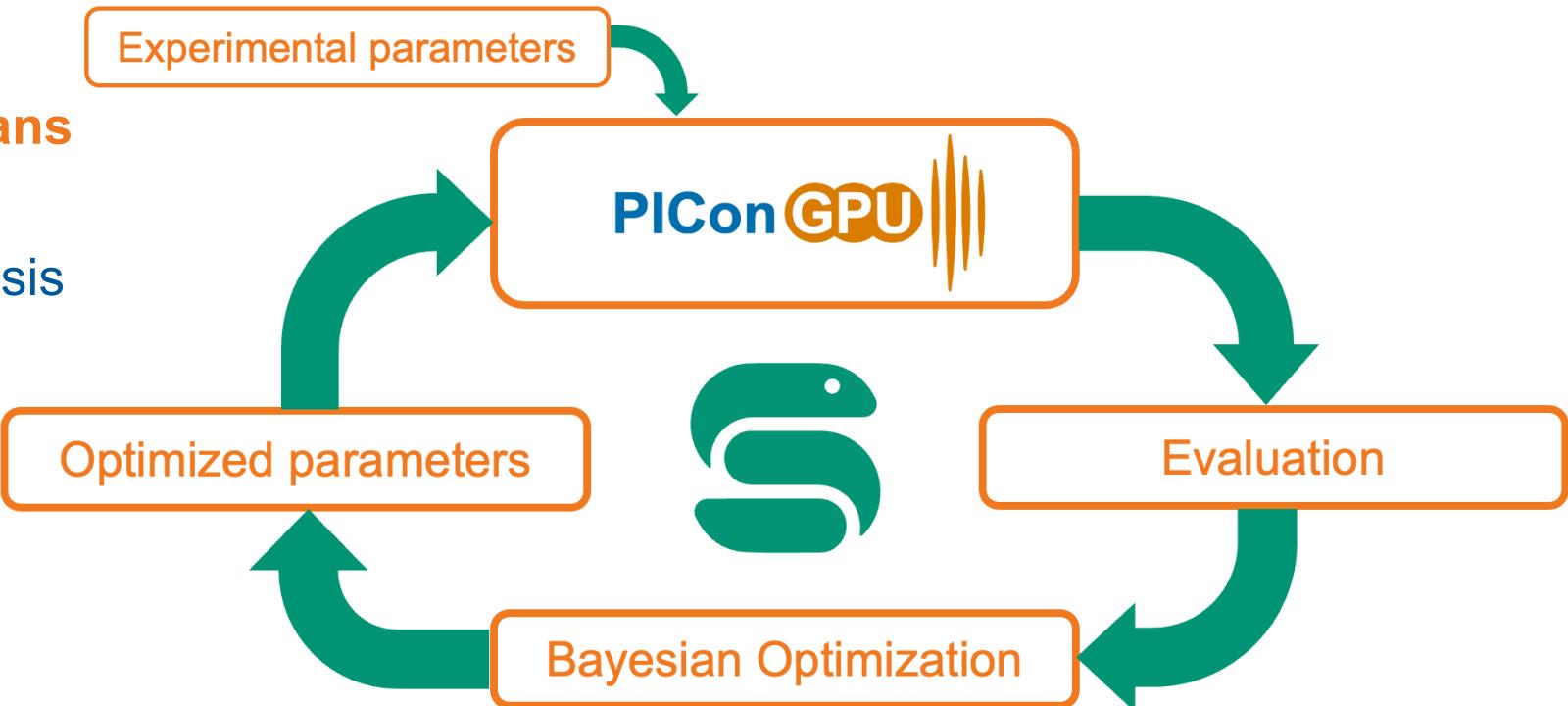
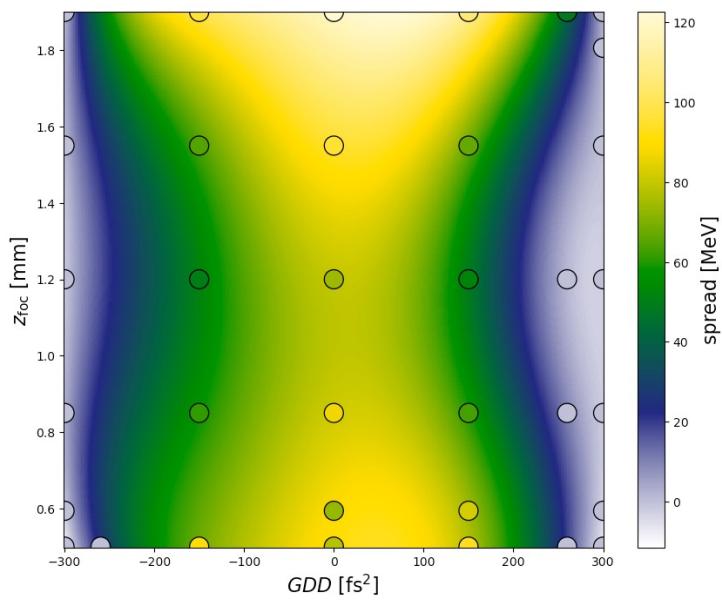


What is new in PIConGPU?

Going beyond “just” large simulation: complex in-situ data analysis

automated workflows

- support for **Snakemake**
- automated parameter scans
- including all stages:
compilation, running, analysis



PIConGPU: developed as an open source code

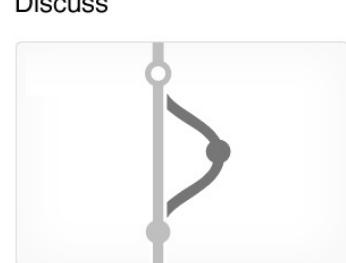
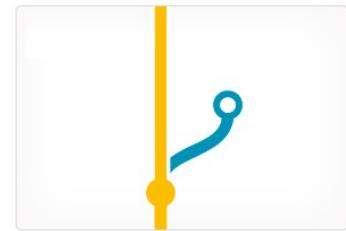
our development is open and relies on open platforms

- fully developed on GitHub
- automated CI tests (on GitHub and GitLab)
- open documentation via readthedocs
- open data via zenodo and rodare
- everybody can ask questions or contribute

 Some checks were not successful
2 errored, 2 failing, and 5 successful checks

[Hide all checks](#)

 ci/gitlab/gitlab.com/picongpu-unitest-compile-reduced-matrix — Pipeline canceled on GitLab	Details
 ci/gitlab/gitlab.com/pmacc-compile-reduced-matrix — Pipeline canceled on GitLab	Details
 ci/gitlab/gitlab.com — Pipeline failed on GitLab	Required Details
 ci/gitlab/gitlab.com/pypicongpu-full-matrix — Pipeline failed on GitLab	Details
 ci/gitlab/gitlab.com/picongpu-compile-reduced-matrix_benchmarks — Pipeline passed on GitLab	Details
 ci/gitlab/gitlab.com/picongpu-compile-reduced-matrix_examples — Pipeline passed on GitLab	Details



GitLab

Thank you for your attention.

Are there any questions?