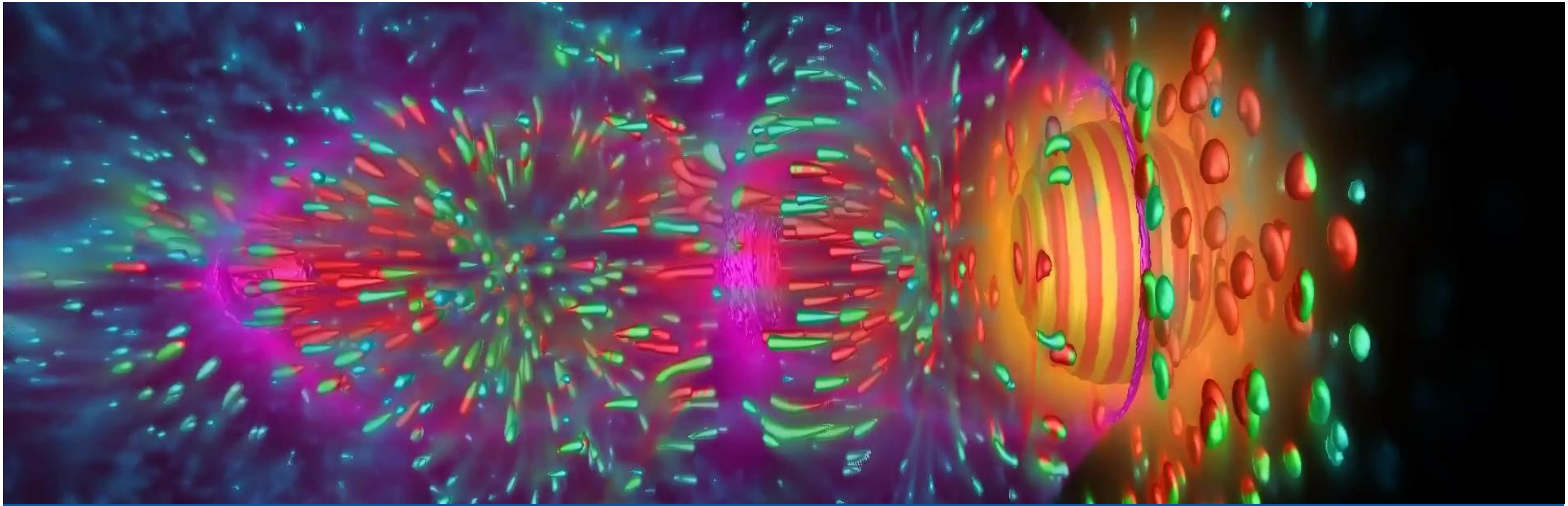


EZ: An Efficient, Charge Conserving Current Deposition Algorithm for Electromagnetic Particle-In-Cell Simulations

Laser Particle Acceleration Division · Computational Radiation Physics Group · Klaus Steiniger · k.steiniger@hzdr.de · www.hzdr.de/crp

EZ: An Efficient, Charge Conserving Current Deposition Algorithm for Electromagnetic Particle-In-Cell Simulations

K. Steiniger, R. Widera, S. Bastrakov, G. Juckeland, J. Kelling, R. Pausch, U. Schramm, A. Debus – *HZDR*

M. Bussmann – *CASUS*

S. Chandrasekaran, M. Leinhauser – *University of Delaware*

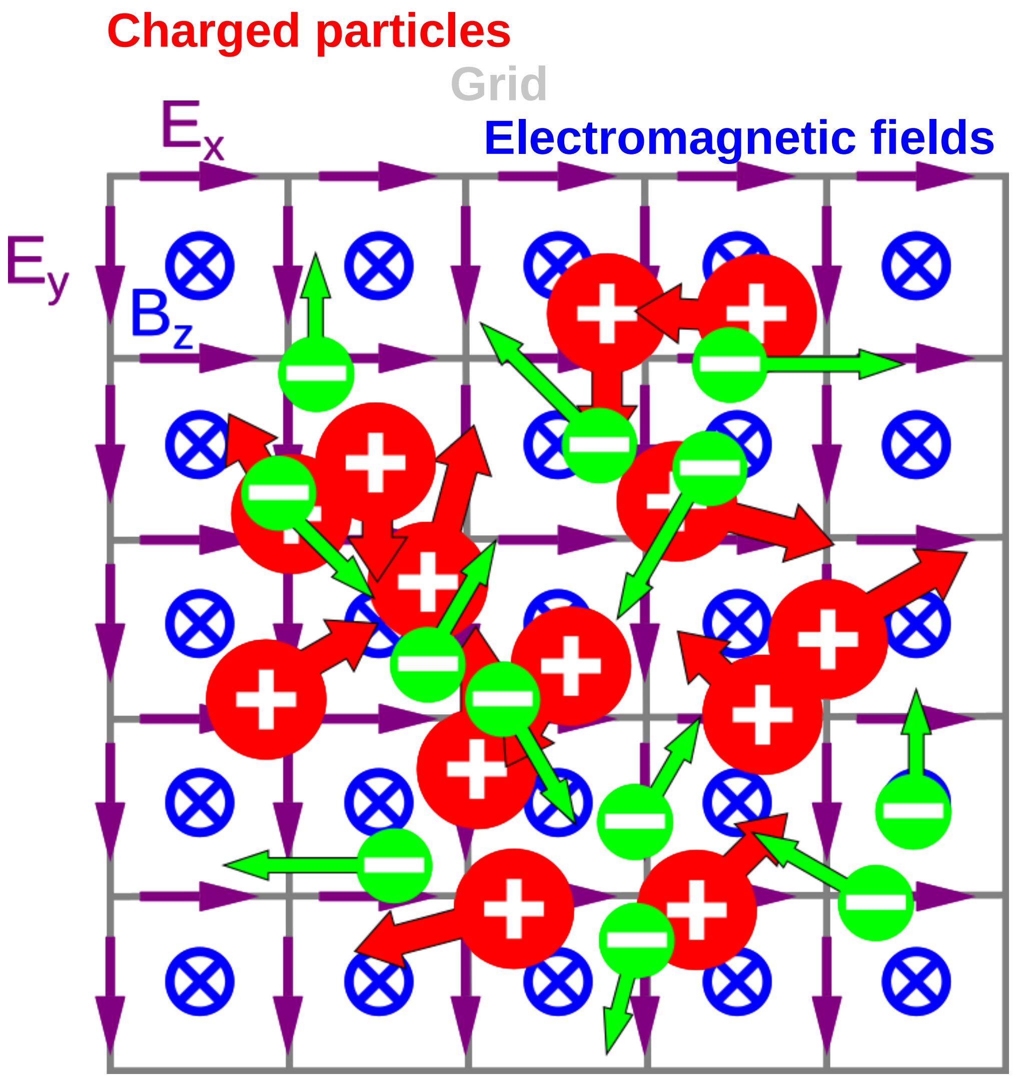
B. Hernandez, D. Rogers – *Oakridge National Laboratory*

A. Huebl – *Lawrence Berkeley National Laboratory*

J. Young – *Georgia Tech*

Manuscript submitted to *Computer Physics Communications*

Current Deposition connects particle motion with field generation



Fields exert force on particles

Particles move

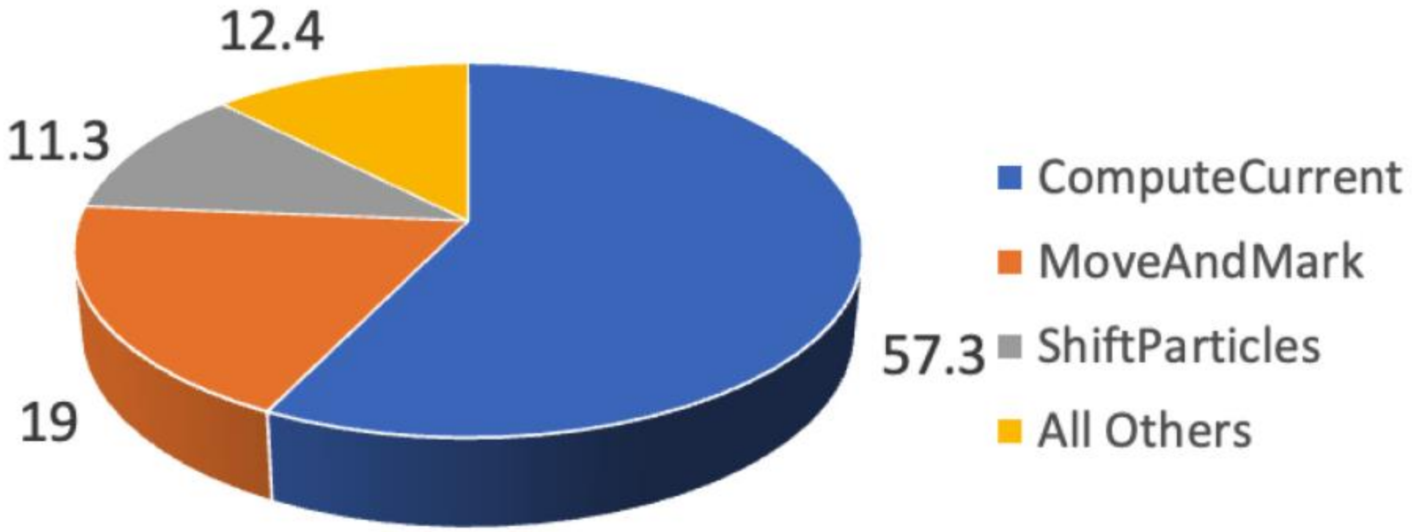
Motion is identified with an electric current

Current generates electric fields

Particle operations are performance critical...

...even percent level optimizations can save a lot of compute time

Particle operations take the main compute time within a time step



In a simulation campaign of a few ten simulations, percent-level optimizations equal entire simulations.

M. Leinhauser et. al., „Metrics and Design of an Instruction Roofline Model for AMD GPUs“, *ACM Transactions on Parallel Computing* 9.1 (2022), doi:10.1145/3505285

How does it work?

Current deposition solves continuity equation to conserve charge in the simulation

1.) Single macro-particle current density is obtained by solving continuity equation

2.) Total current density is the sum of all single macro particle current densities ***J***.

Continuity equation

$$\nabla \boldsymbol{J} = -\frac{\partial \rho}{\partial t}$$

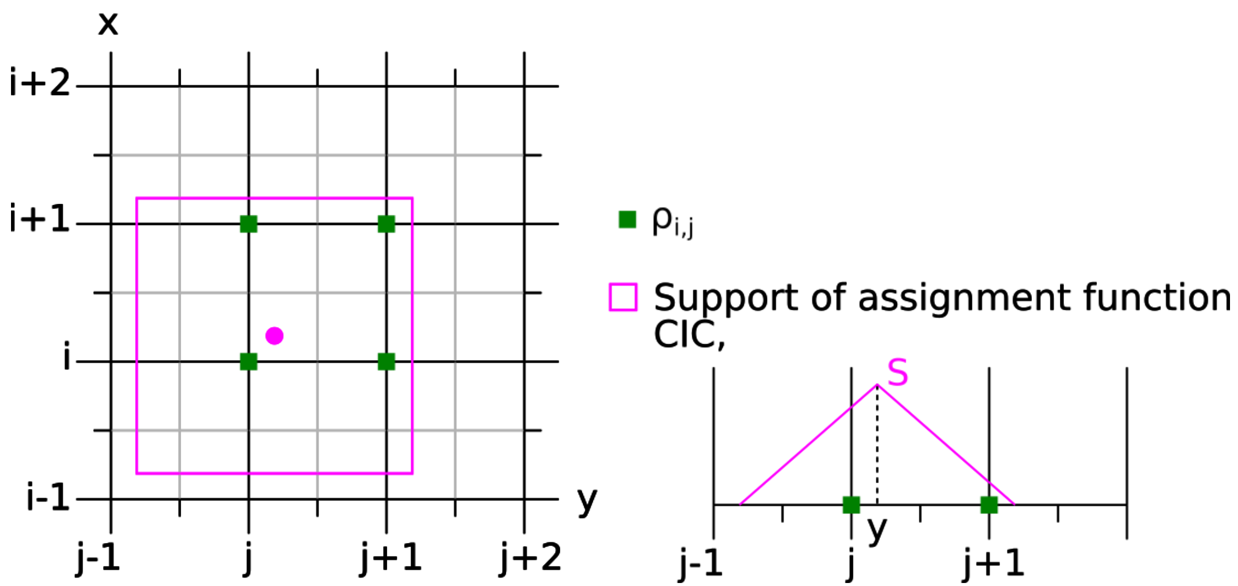
Calculation of charge density on the grid

$$\rho_{i,j}^{\text{single}-\text{mp}} = Q_p \, S(x_p - i\Delta x) S(y_p - j\Delta y)$$

<u>Where:</u>

- **Single macro-particle charge density *ρ***
depends on macro-particle assignment function *S.*

- ***Assignment-function S*** distributes the macro-particle‘s contribution to fields over several grid nodes.
This ensures smoothness of the physical observables.



Macro-particle move determines macro-particle current density

Continuity equation

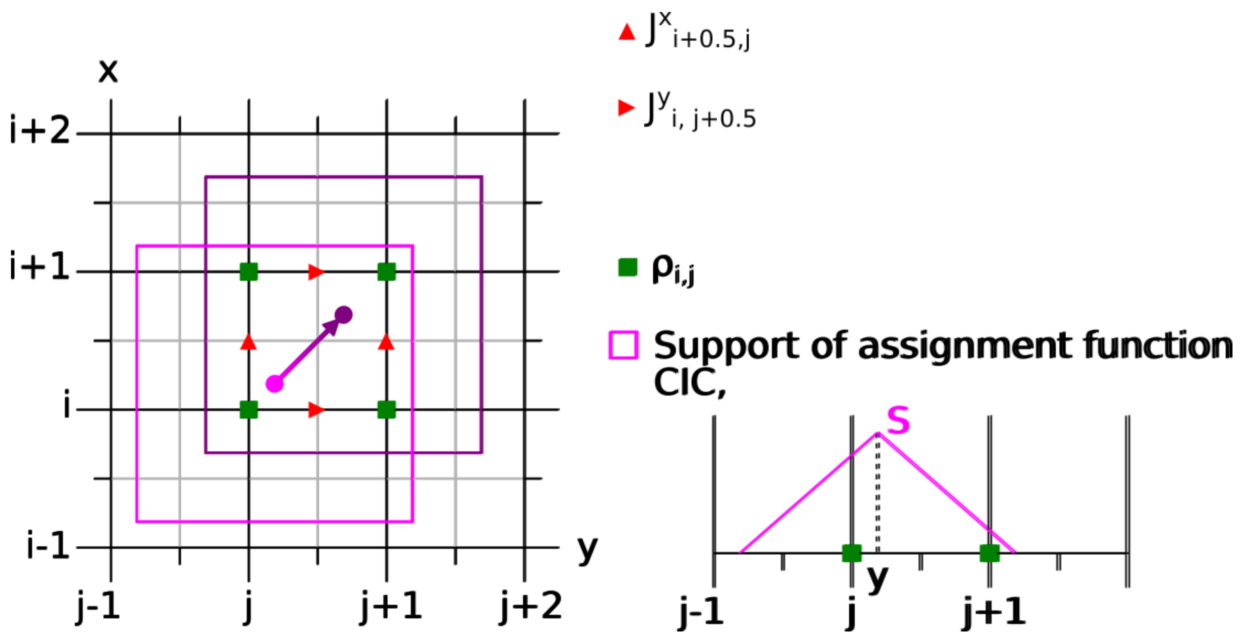
$$\nabla \boldsymbol{J} = -\frac{\partial \rho}{\partial t}$$

Calculation of charge density on the grid

$$\rho_{i,j}^{\text{single}-\text{mp}} = Q_p\, S(x_p - i\Delta x) S(y_p - j\Delta y)$$

Due to macro-particle movement, charge density on grid nodes changes

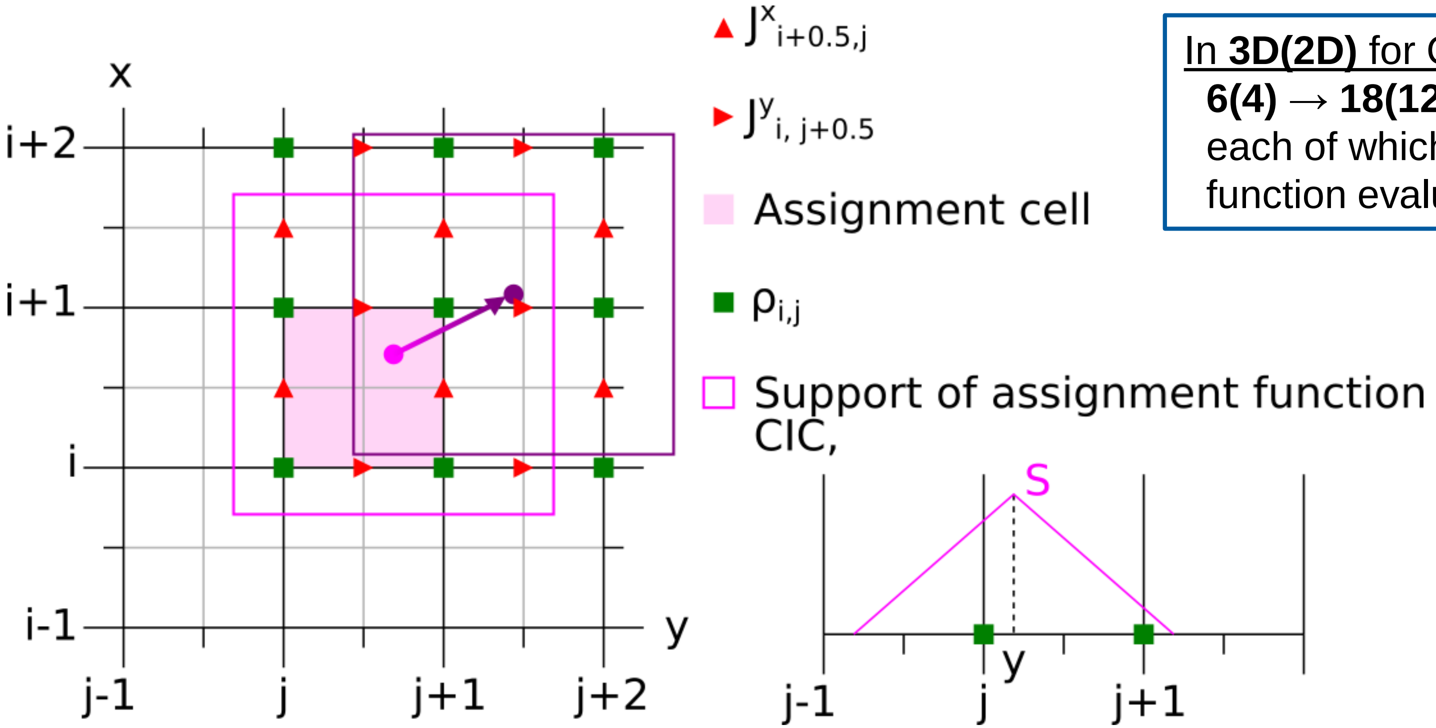
→ From the difference, the single macro-particle current density is calculated



Current Deposition becomes more costly once the macro-particle leaves its Assignment Cell

The **Assignment Cell** marks the volume within which the macro-particle can move without changing the grid nodes to which charge density is assigned.

Once the macro-particle leaves its **Assignment Cell,** its charge contributes current density to more grid nodes.

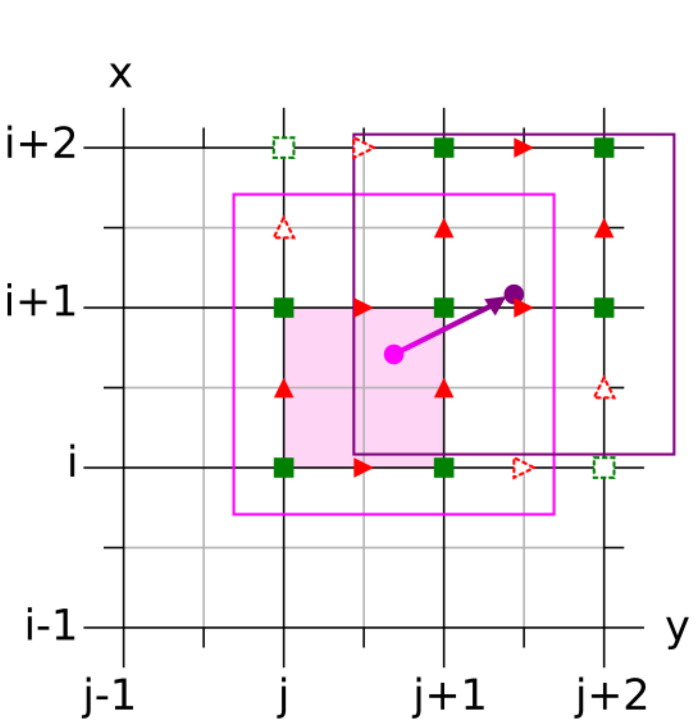


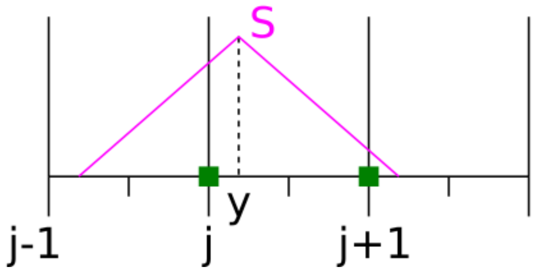
<u>In **3D(2D)** for CIC (1st order) with Esirkepov's method:</u>
6(4) → **18(12)** current density calculations each of which requires 6(4) assignment function evaluations

EZ becomes more performant by saving computations

EZ = Esirkepov meets ZigZag

Actually, a number of current density values vanish





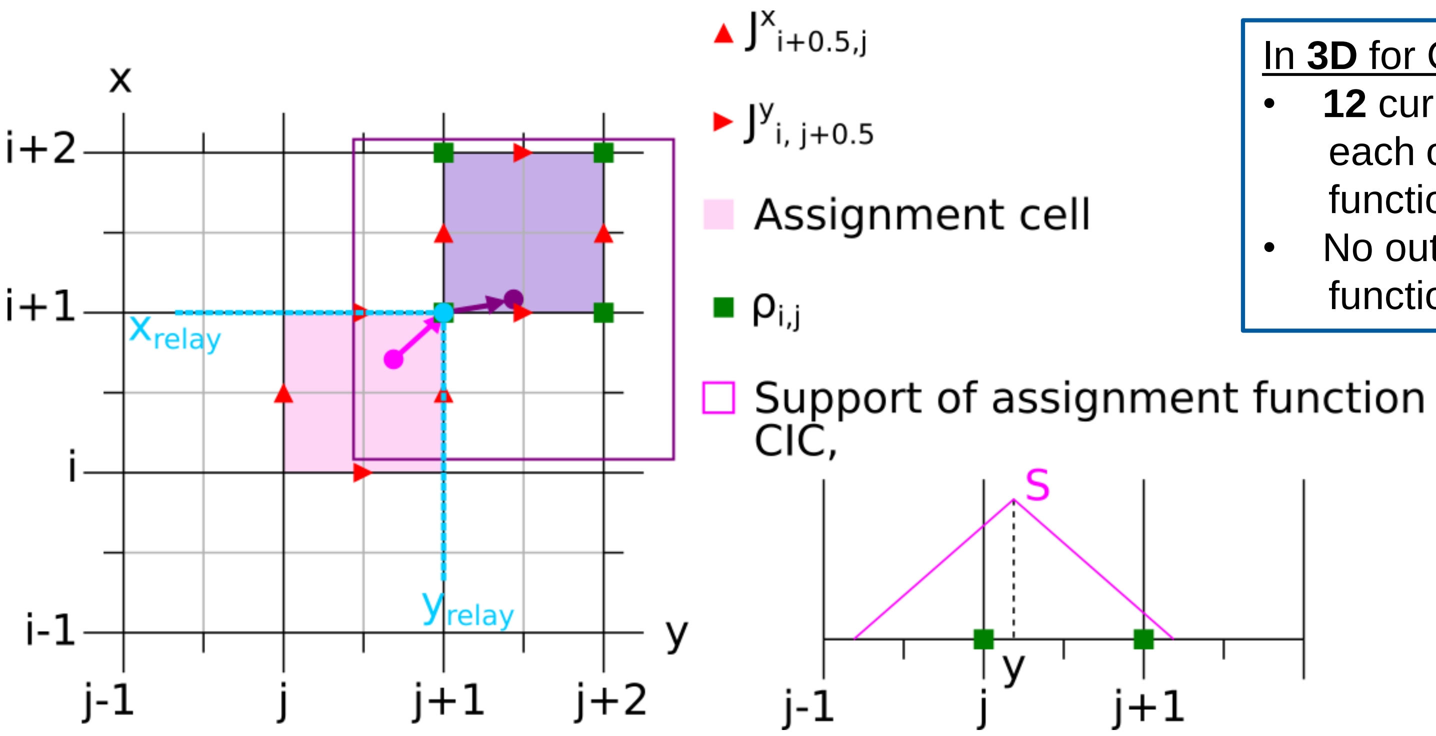
In **3D** for CIC (1st order):

Only 2x6 = **12** current density calculations are required.

33% saving in contrast to Esirkepov's 18.

EZ saves superfluous computations by design

1) Introduce **relay point** at the intersection of assignment cells at initial and final position
2) Split macro-particle trajectory and redirect movement over relay point
3) Calculate current density for each trajectory segment
   → **Reduced computational effort since each movement is confined to assignment cell**
4) Sum the two individual current densities to obtain the total current density



In **3D** for CIC (1st order) with EZ:

- **12** current density calculations each of which requires 6 assignment function evaluations
- No out-of-support evaluations of assignment functions reducing function footprint

Trajectory splitting is not new

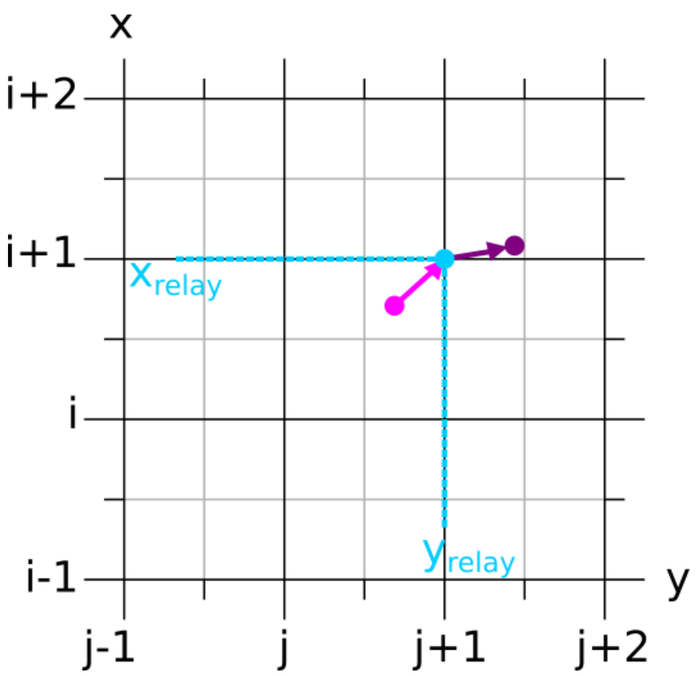
ZigZag employs trajectory splitting as well

Notable differences between EZ and ZigZag:

- EZ uses Esirkepov's method to calculate current density for each trajectory segment
  (which is different from ZigZag)

- EZ is generalized to arbitrary assignment-functions while ZigZag is not

- EZ's choice of the relay point is different from ZigZag's



EZ is more performant for assignment-functions up to 3rd order

Test case: Warm, relativistic plasma simulation with PIConGPU

assignment-functions]		average time per step [ms]		time per step
name	order l	Esirkepov's method	EZ	speedup w/ EZ
NVIDIA V100 16GiB				
CIC	1	88.15 ± 0.07	80.04 ± 0.06	1.100
TSC	2	164.24 ± 0.06	151.65 ± 0.04	1.083
PQS	3	356.16 ± 0.03	319.91 ± 0.06	1.113
AMD MI100				
CIC	1	144.66 ± 0.52	139.43 ± 0.79	1.034
TSC	2	226.93 ± 0.79	217.02 ± 0.73	1.046
PQS	3	363.83 ± 0.80	366.70 ± 0.57	0.992
AMD EPYC 7662 64-core				
CIC	1	892.1 ± 0.4	864.9 ± 0.4	1.031
TSC	2	1904.3 ± 0.5	1674.2 ± 2.1	1.137
PQS	3	4658.0 ± 0.9	5037.6 ± 1.1	0.925

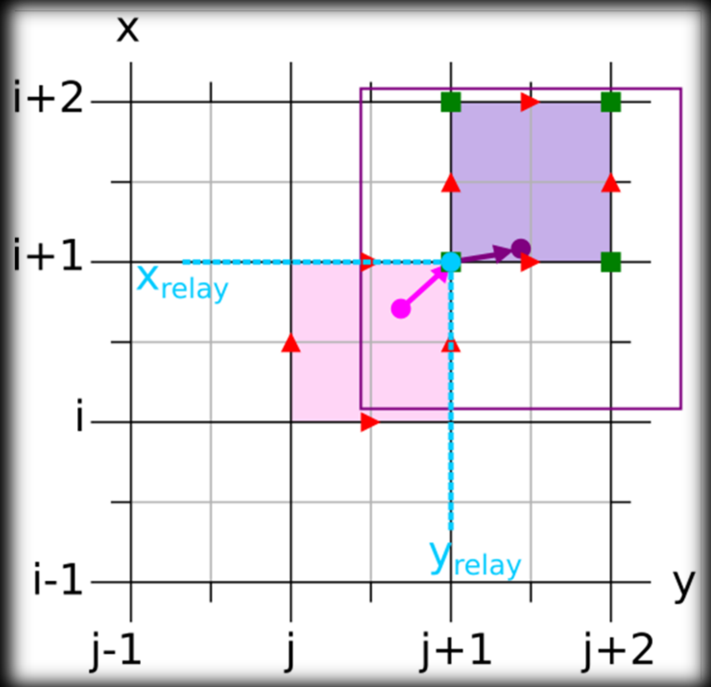
Interested? Check the paper for...

... **detailed instructions** for the calculation of the single macro-particle current density

... **optimizations** for the splitting point location in order to reduce the number of current calculations

... **validation and benchmarks** to Esirkepov‘s method in test cases

... **profiling results** which are the basis of the discussion of performance benchmark results



K. Steiniger, R. Widera, et. al.,
„EZ: An Efficient, Charge Conserving Current Deposition Algorithm for Electromagnetic Particle-In-Cell Simulations“, submitted to *Comp. Phys. Comm.*