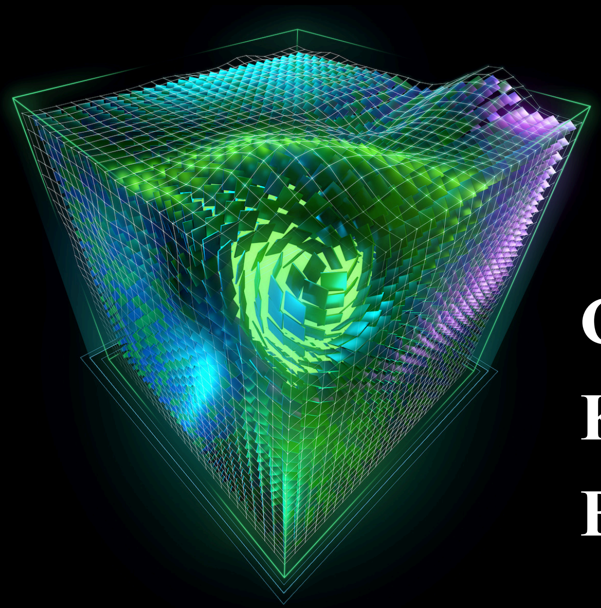
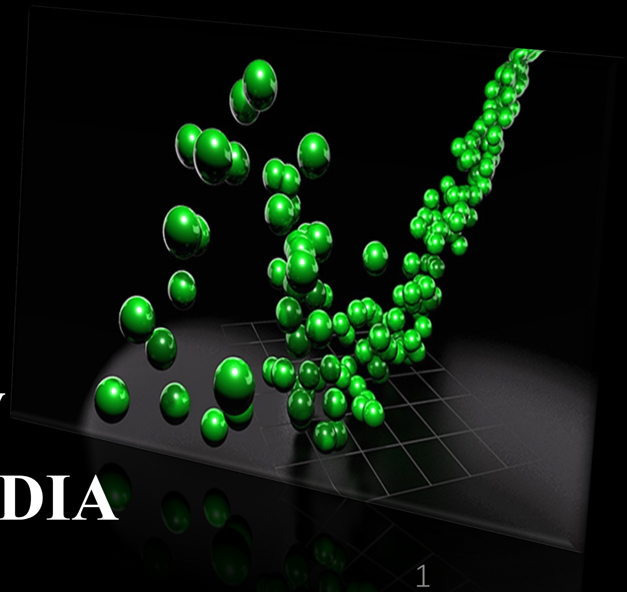
GPIC: An Advanced Particle-In-Cell Code Using GPU Acceleration and its Application in Magnetic Reconnection

Shiyong Huang

Wuhan University, China

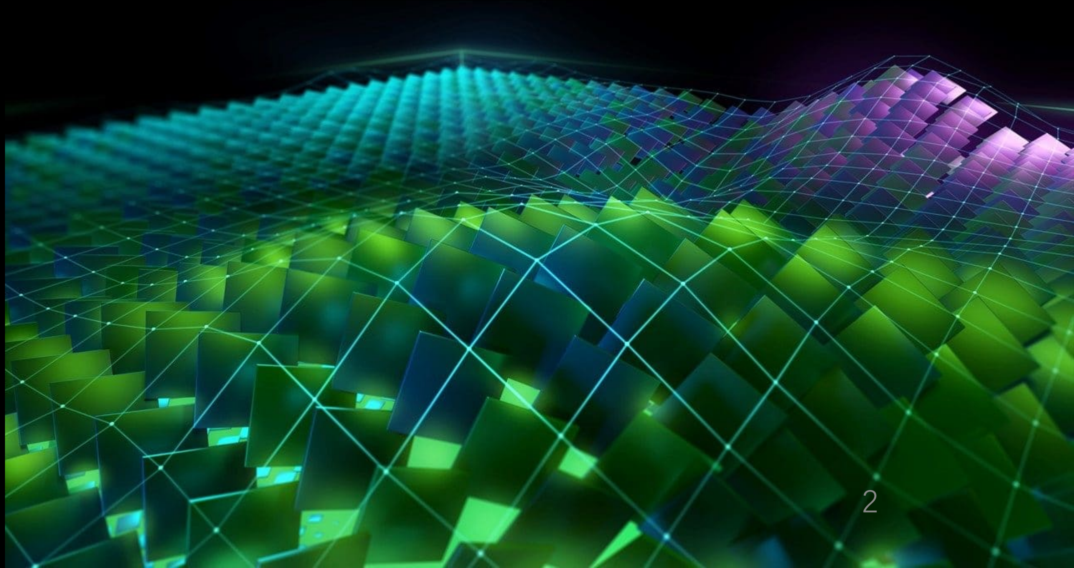


Collaborators: Qiyang Xiong, Zhigang Yuan,
Kui Jiang, Jian Zhang, from Wuhan University
Bharatkumar Sharma, Lvlin Kuang, from NVIDIA

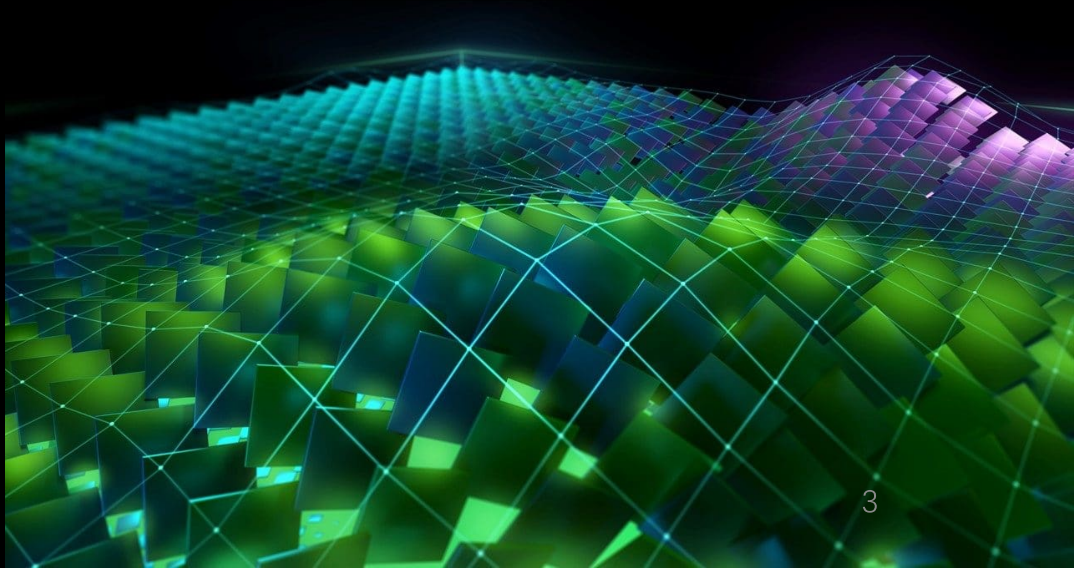


Outline

- ✓Introduction of PIC
- ✓Development of GPIC
- ✓Performance of GPIC
- ✓Application in MR
- ✓Conclusions



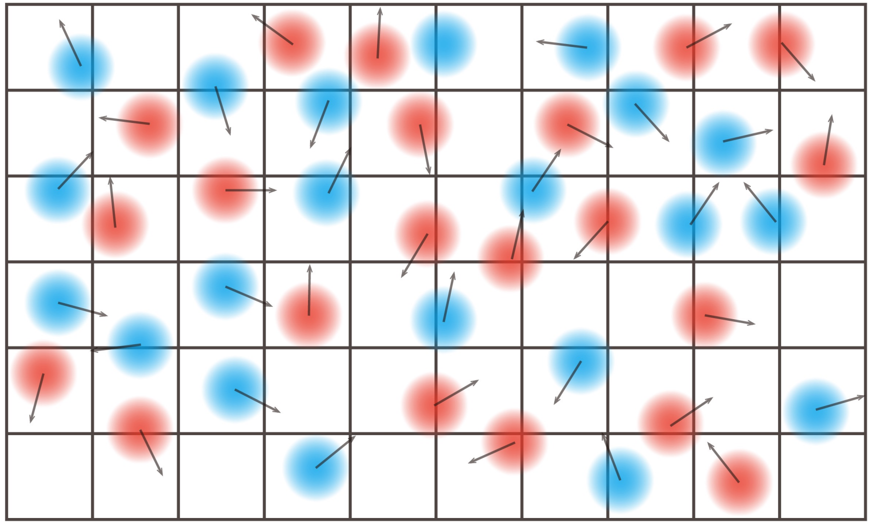
Introduction of Particle-in-Cell Simulation



Introduction of Particle-in-Cell Method

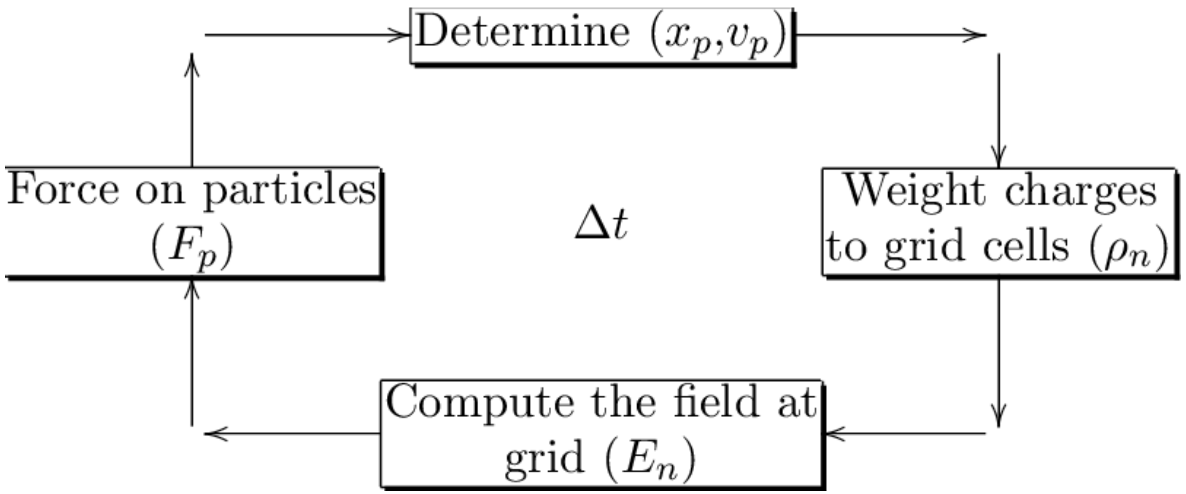
General Concept of Particle-in-Cell

- A limited spatial area is meshed using certain grid resolution for field;
- Using finite number of macro-particles to represent the certain density plasma in real space;
- The system evolves self-consistent with time following physical laws.



Common Steps For Solver:

1. Particles are forced by the local fields;
2. Currents/Charges are contributed by the particles;
3. Solve the field according to the relation.



Introduction of Particle-in-Cell Method

Explicit Numerical Solver of Collisionless Electromagnetic Scheme

Field:

Mesh Grid: Yee staggered grid.

Solver: Faraday's law and Ampere's law in discrete form.

$\partial \boldsymbol{B}/\partial t = -c(\nabla\times\boldsymbol{E})$ $\quad$ $\partial \boldsymbol{E}/\partial t = (\nabla\times\boldsymbol{B}) - 4\pi\boldsymbol{J}$

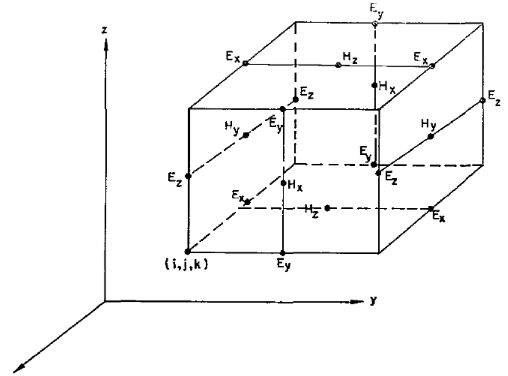
Particle:

Solver: Newton-Lorentz law. $\partial \boldsymbol{v}/\partial t = q/m(\boldsymbol{E} + v\times\boldsymbol{B})$

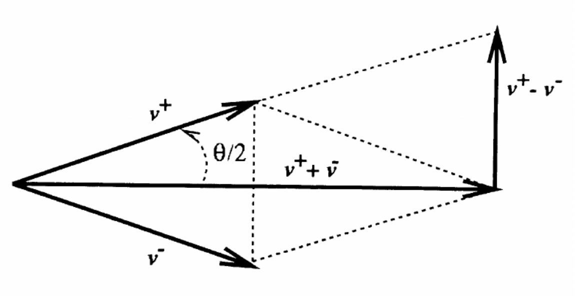
Implement: Buneman-Boris Rotation.

Overtime:

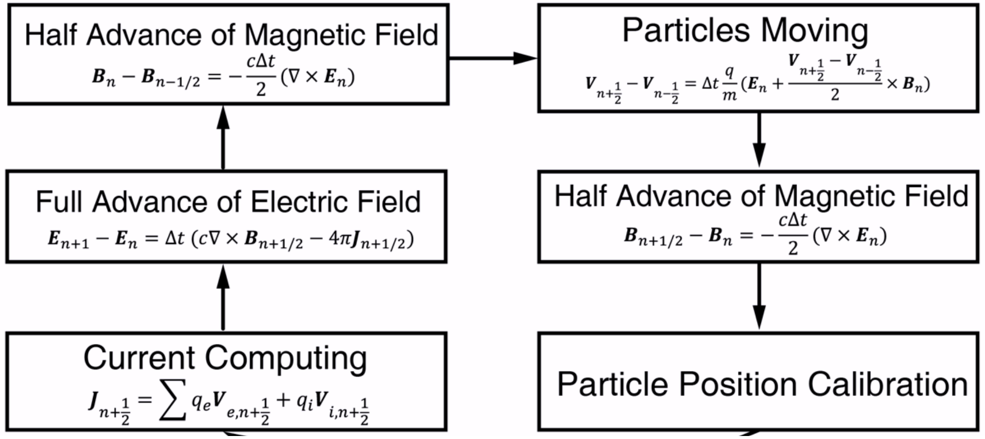
Solver: Leap-frog Method (Second-order in Time).



[Yee, 1966]



[Boris, 1970; Buneman, 1976]



Half Advance of Magnetic Field
$\boldsymbol{B}_n - \boldsymbol{B}_{n-1/2} = -\frac{c\Delta t}{2}(\nabla\times\boldsymbol{E}_n)$

Particles Moving
$\boldsymbol{V}_{n+\frac{1}{2}} - \boldsymbol{V}_{n-\frac{1}{2}} = \Delta t\frac{q}{m}(\boldsymbol{E}_n + \frac{\boldsymbol{V}_{n+\frac{1}{2}} - \boldsymbol{V}_{n-\frac{1}{2}}}{2}\times\boldsymbol{B}_n)$

Half Advance of Magnetic Field
$\boldsymbol{B}_{n+1/2} - \boldsymbol{B}_n = -\frac{c\Delta t}{2}(\nabla\times\boldsymbol{E}_n)$

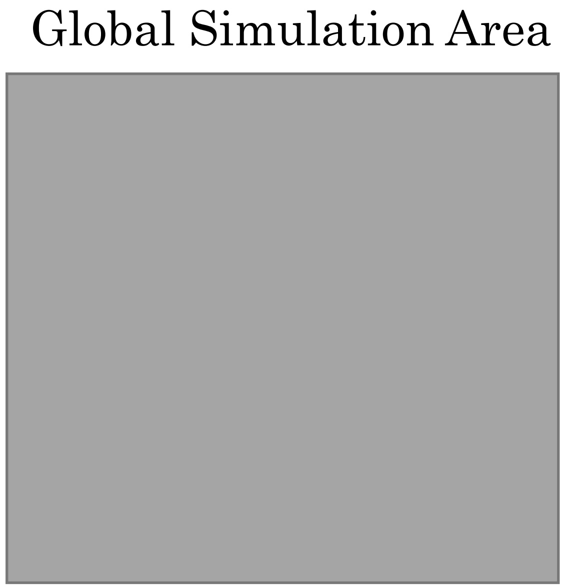
Particle Position Calibration

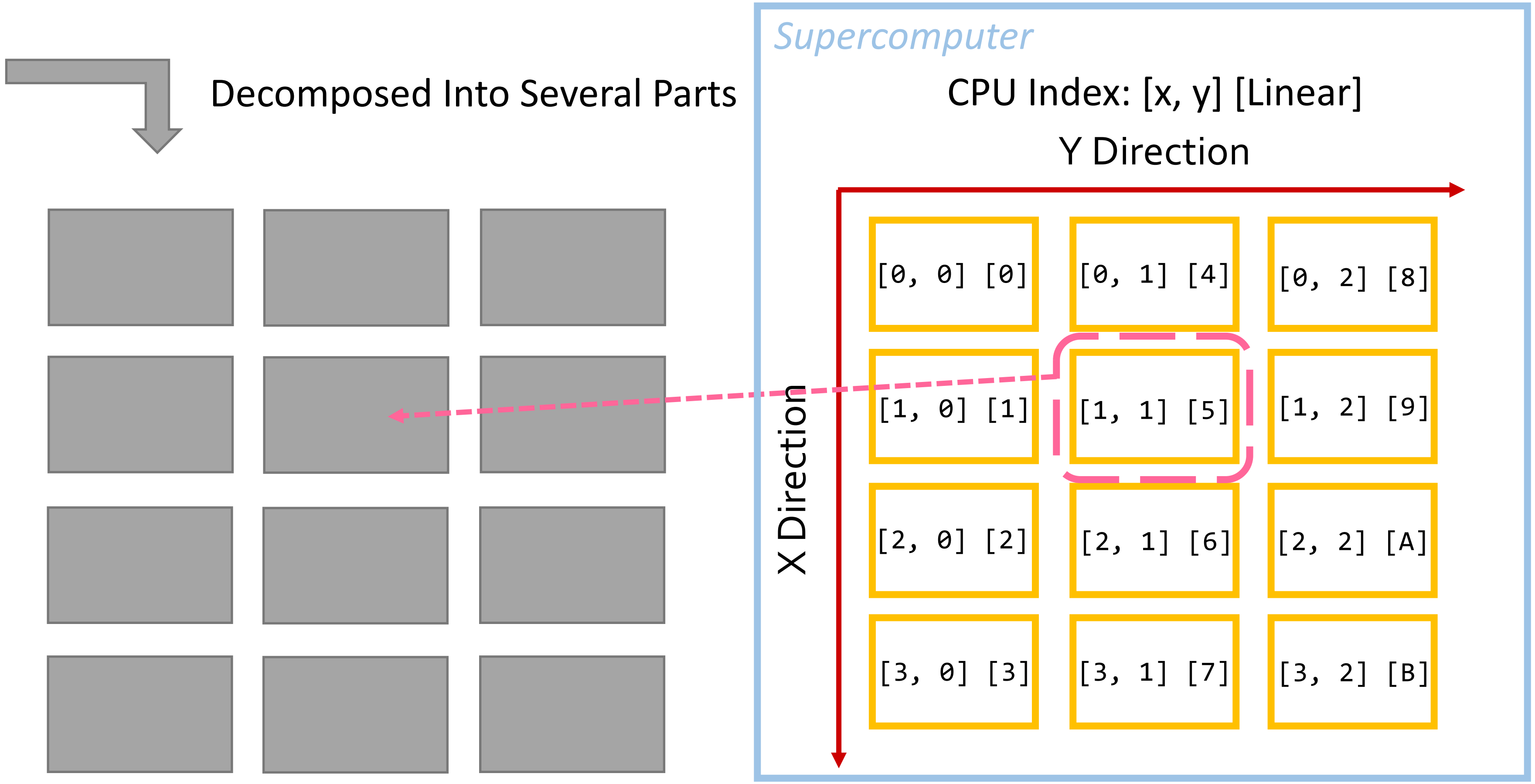
Current Computing
$\boldsymbol{J}_{n+\frac{1}{2}} = \sum q_e\boldsymbol{V}_{e,n+\frac{1}{2}} + q_i\boldsymbol{V}_{i,n+\frac{1}{2}}$

Full Advance of Electric Field
$\boldsymbol{E}_{n+1} - \boldsymbol{E}_n = \Delta t\,(c\nabla\times\boldsymbol{B}_{n+1/2} - 4\pi\boldsymbol{J}_{n+1/2})$

Introduction of Particle-in-Cell Method

High-Performance Computing of PIC Simulation – MPI (Message Passing Interface)





Supercomputer

CPU Index: [x, y] [Linear]

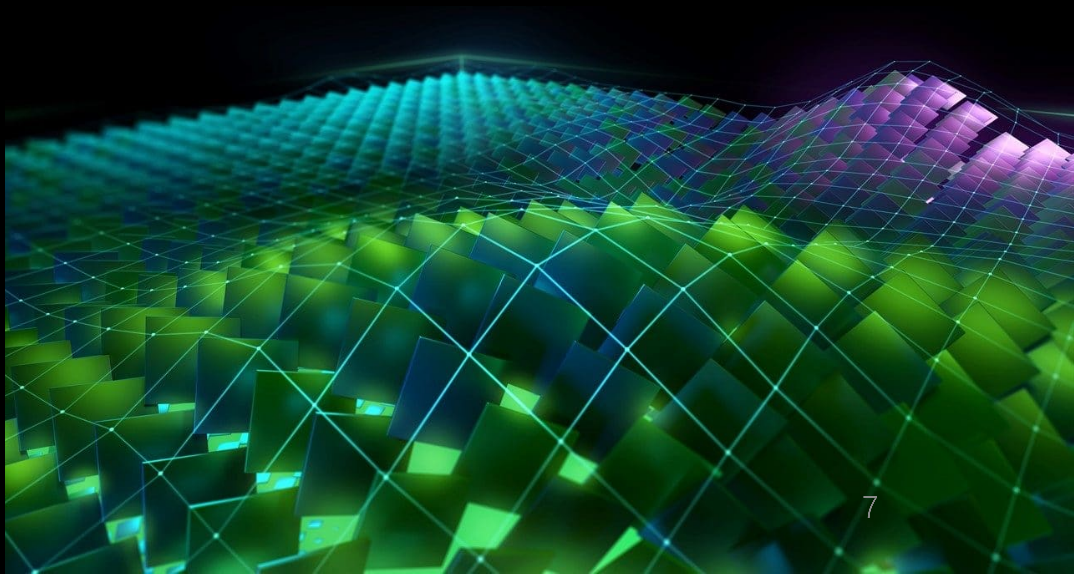
Y Direction

X Direction

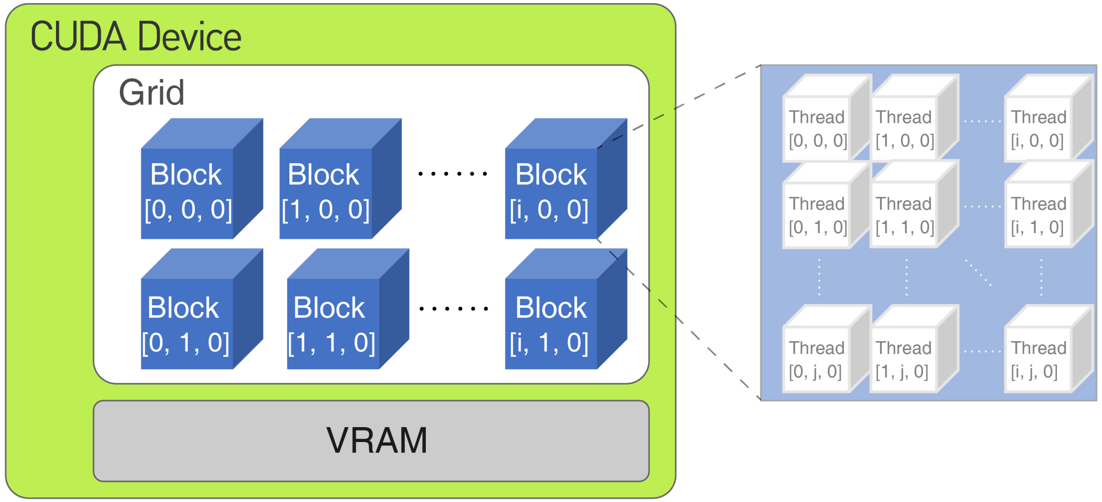
[0, 0] [0]	[0, 1] [4]	[0, 2] [8]
[1, 0] [1]	[1, 1] [5]	[1, 2] [9]
[2, 0] [2]	[2, 1] [6]	[2, 2] [A]
[3, 0] [3]	[3, 1] [7]	[3, 2] [B]

- Field-Decomposition Method: Each CPU handles the computing of corresponding subarea.

Development of PIC Simulation Using GPU Computing

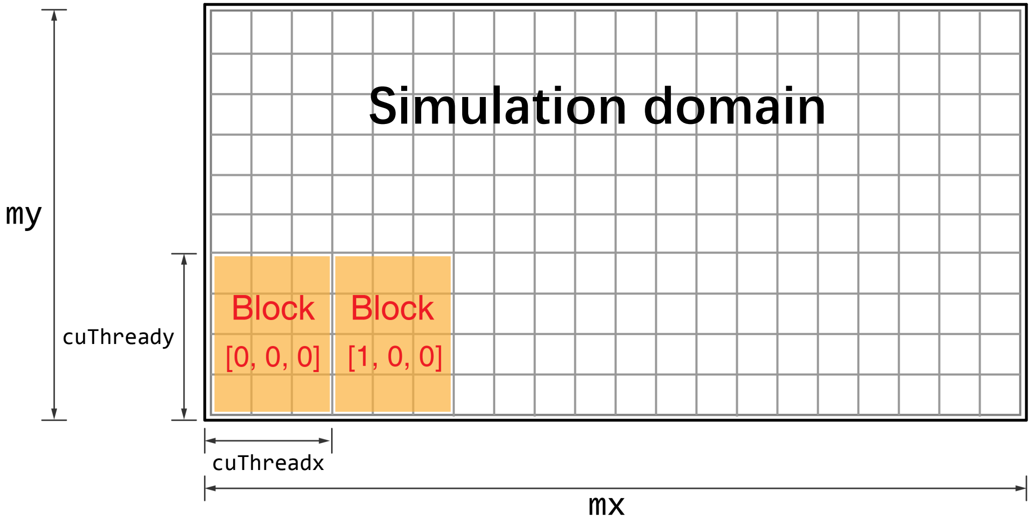


General Computing of GPU Device – Thread & Block



- A single GPU device contains numerous “Block”, and each Block contains numerous “Thread”;
- Each Thread can execute computing instructions independently.

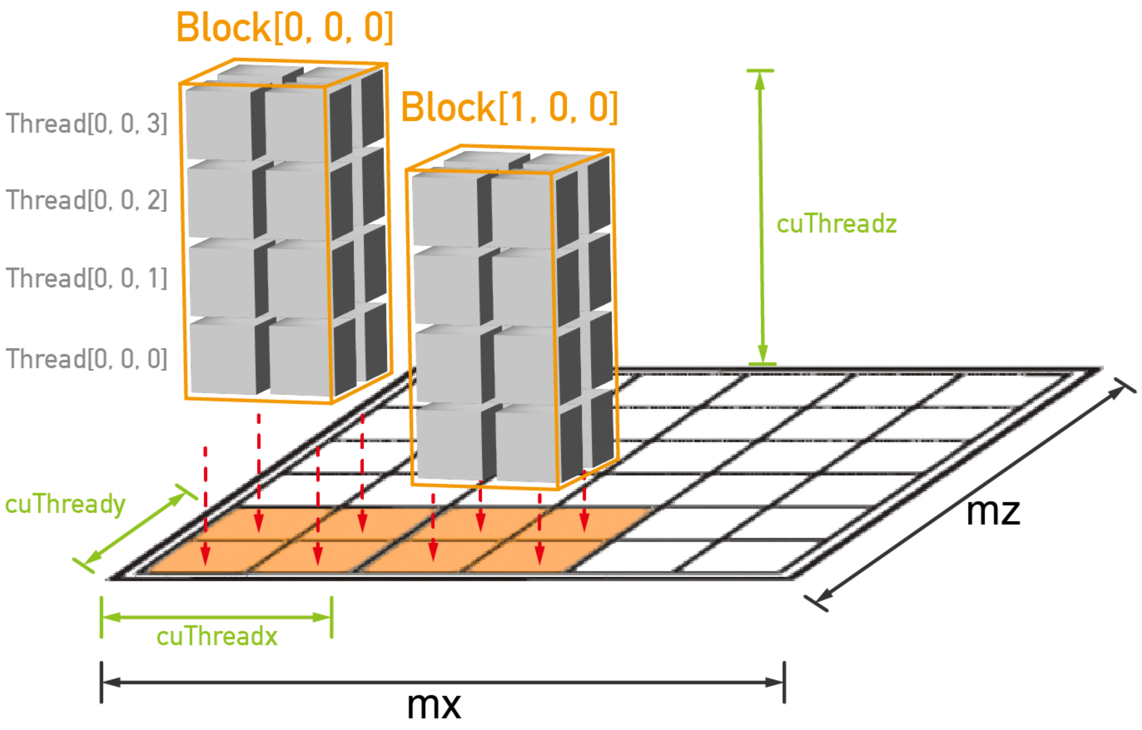
(a) Mapping of Threads & Blocks to 2D Field Array



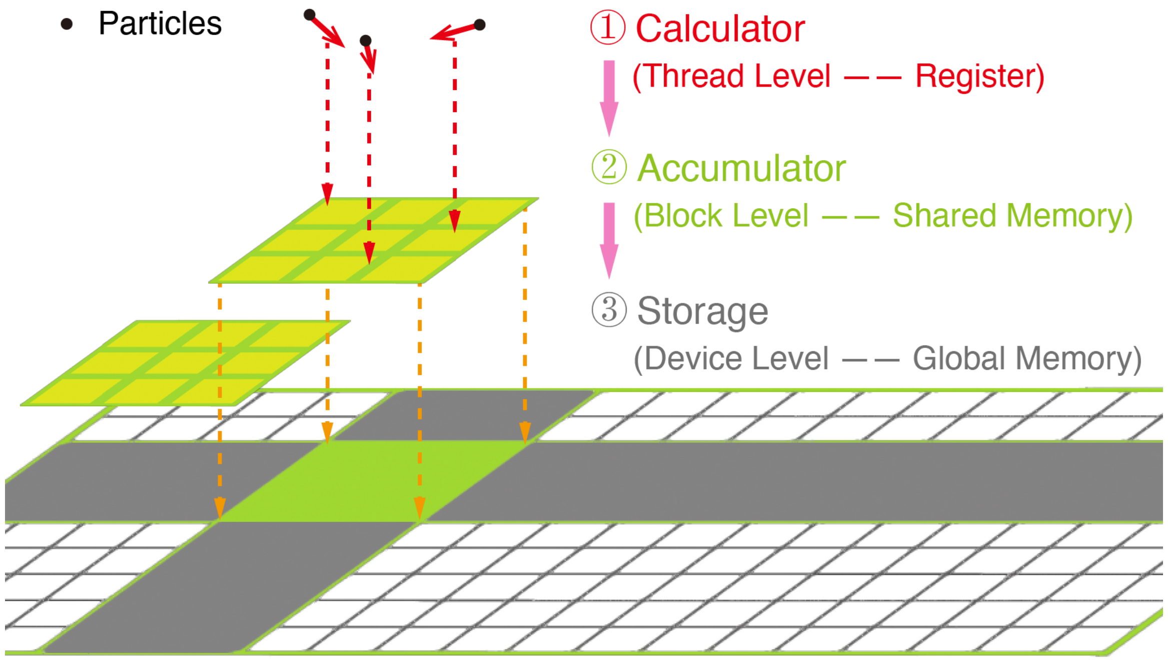
(b) Mapping of Threads & Blocks to 1D Particle Array

Scheme Design of PIC on GPU

Multiple Thread Dealing With Single Grid



Three-Level Data Exchange Strategy



① Calculator
(Thread Level —— Register)

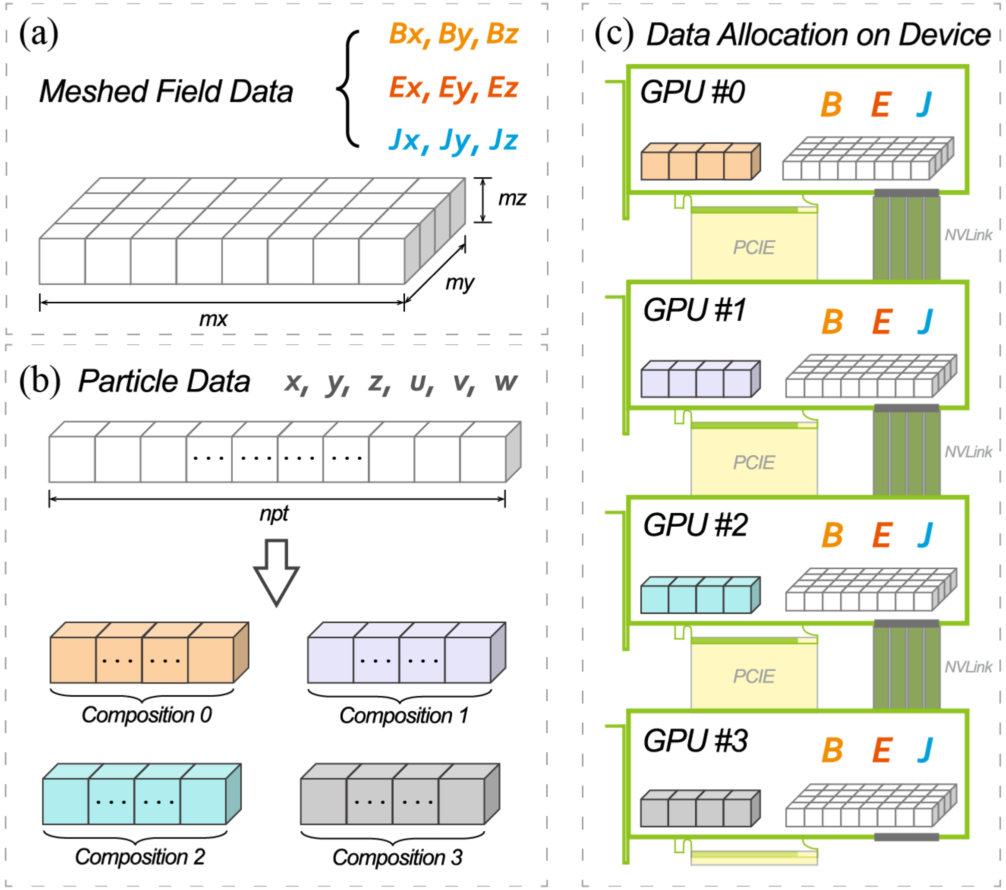
② Accumulator
(Block Level —— Shared Memory)

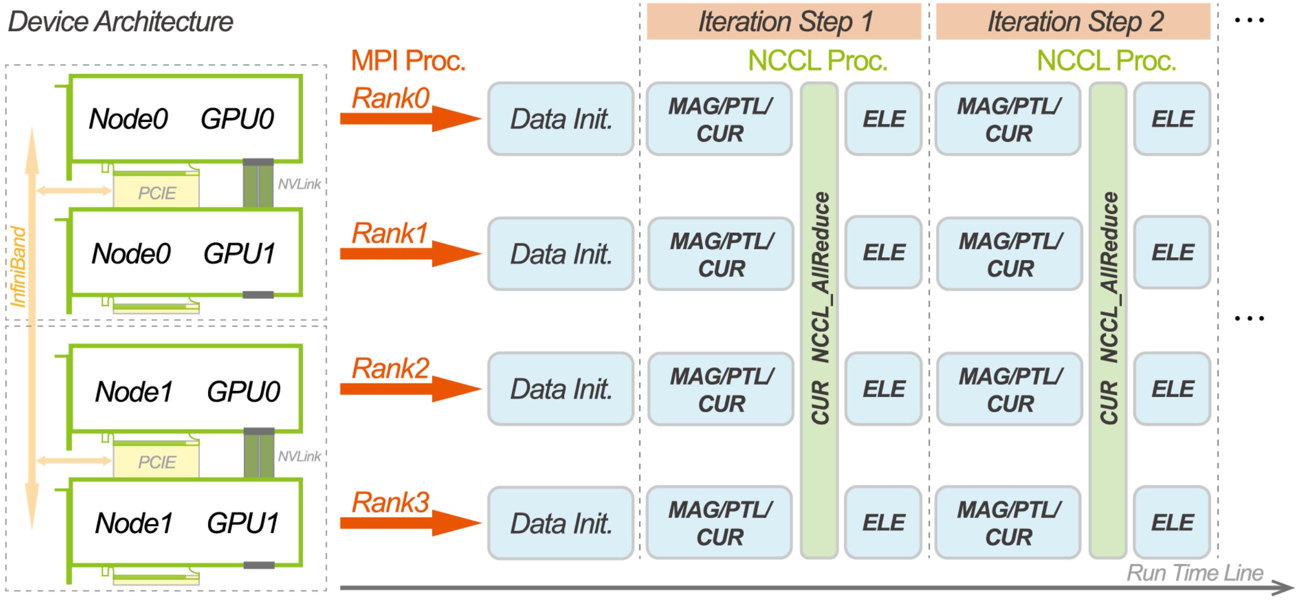
③ Storage
(Device Level —— Global Memory)

Development of PIC Simulation Using GPU Computing

Scheme Design of PIC on GPU

Multi-GPU Computing Pattern





- Field-Duplication Method:

Each GPU holds the identical field data and different compositions of particle data.

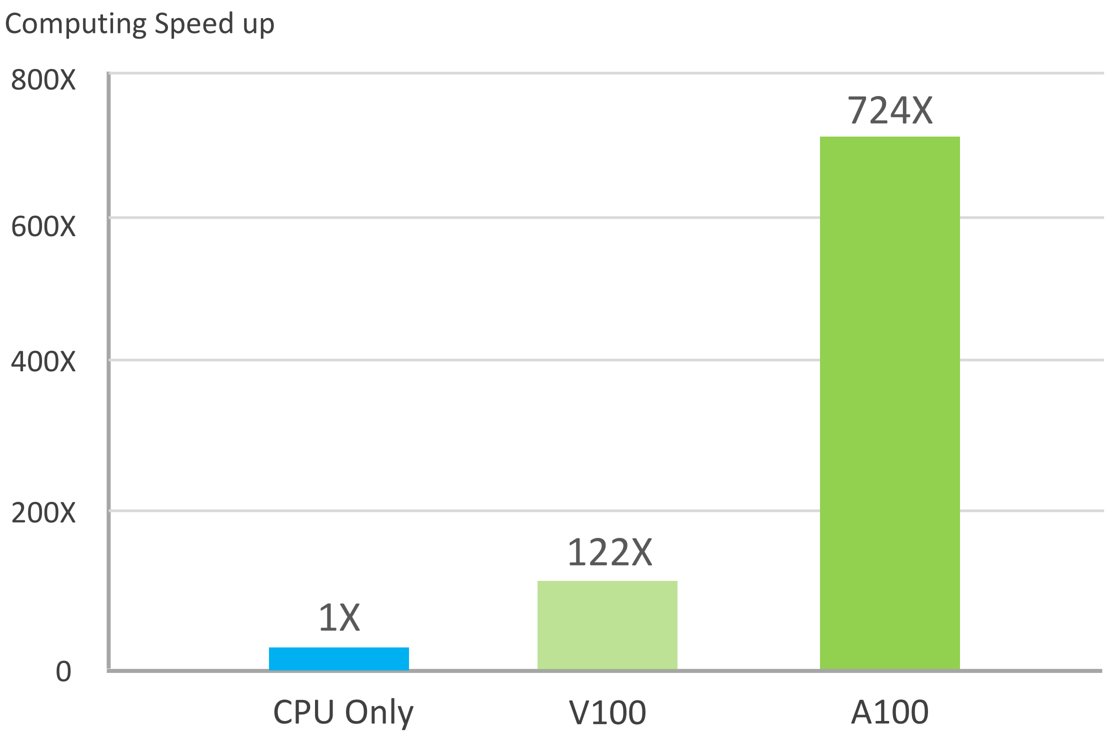
Summary of **GPIC** (GPU-PIC) Program

Computing Platform:	NVIDIA HPC SDK
Language:	CUDA Fortran `(.f90, .f08)`
Compiler:	*nvfortran/mpif90*
Communication Library:	*HPC-X, NCCL(NVIDIA Collective Communication Library)*
Math Library:	Thrust, cuRand, cuTensor
Supportive:	All NVIDIA Series GPUs (Capability > 2.5, CUDA Version > 6.0)

Development of PIC Simulation Using GPU Computing

Performance of GPIC

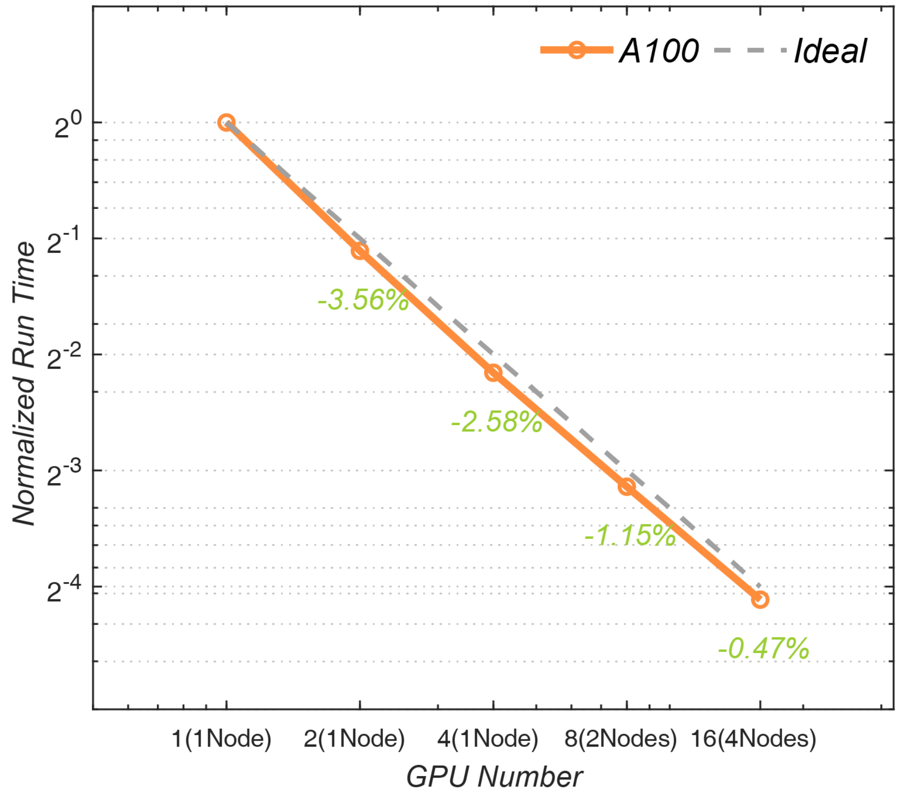
Peak Performance of Single GPU Device



Time Per 10,000 Iterations –Relative Performance

CPU Only: Intel Xeon Gold 6248 @ 2.50 GHz | V100: NVIDIA TESLA V100-SXM2-16GB | A100: NVIDIA A100-SXM4-40GB

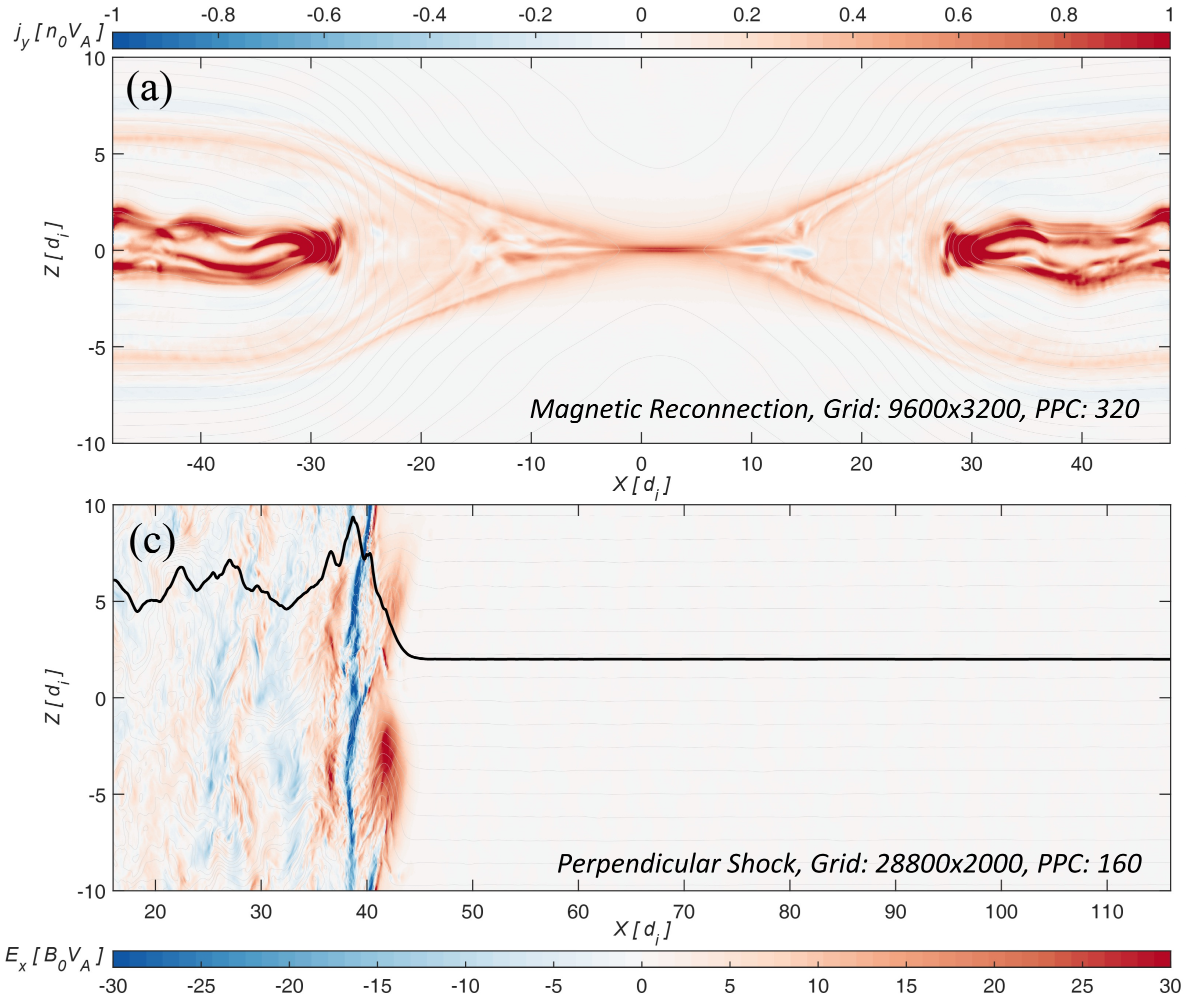
Acceleration Rate on Multiple GPU Devices



Internal Link: NVLink 600GB/s; External Link: NVIDIA Connect-X 6, Infiniband, EDR, 100GB/s

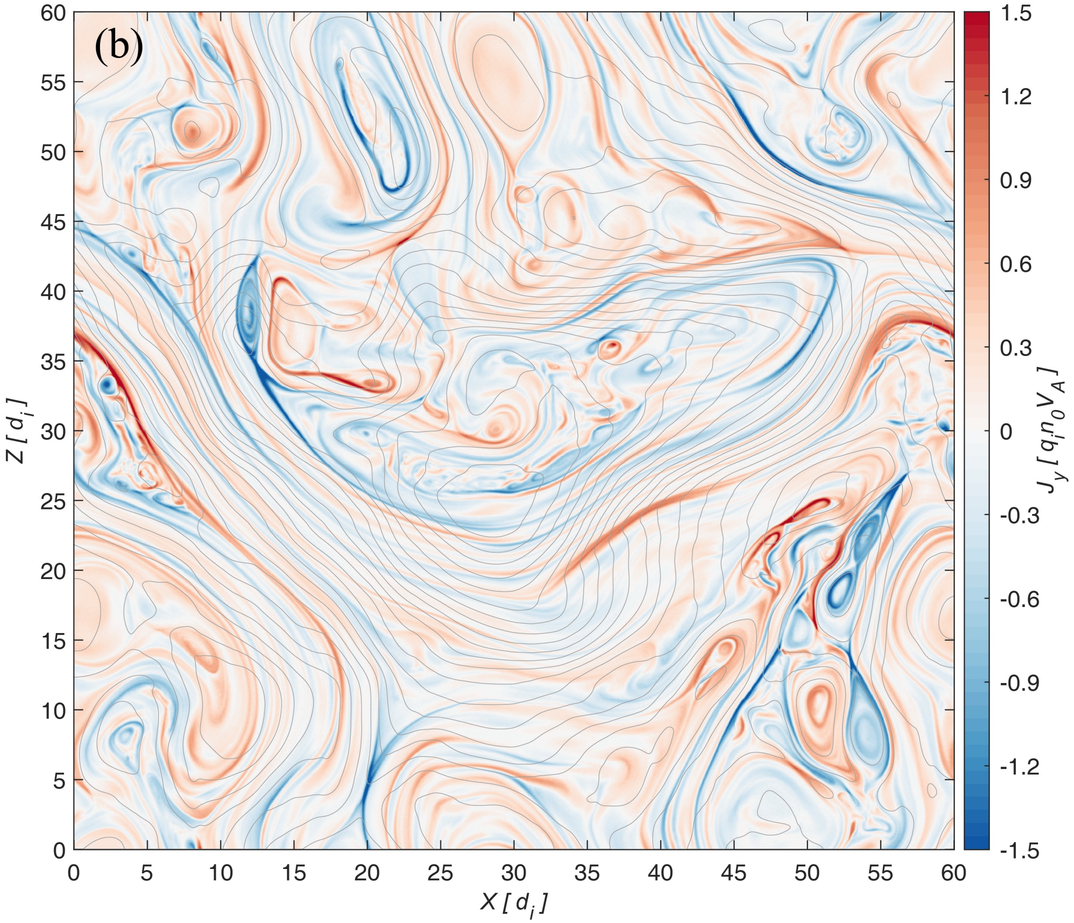
Development of PIC Simulation Using GPU Computing

Examples of GPIC Simulations



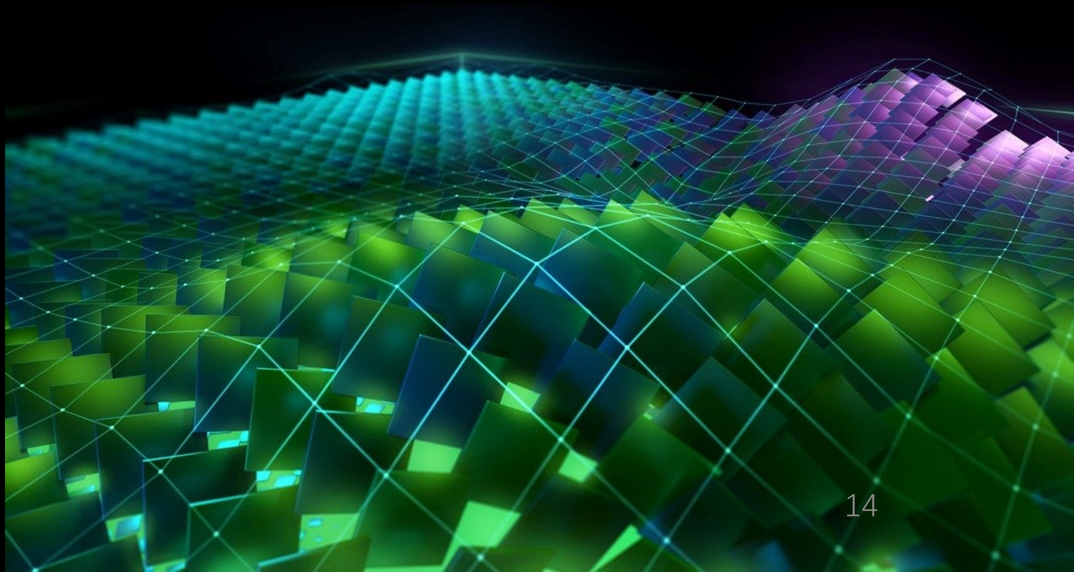
Magnetic Reconnection, Grid: 9600x3200, PPC: 320

Perpendicular Shock, Grid: 28800x2000, PPC: 160



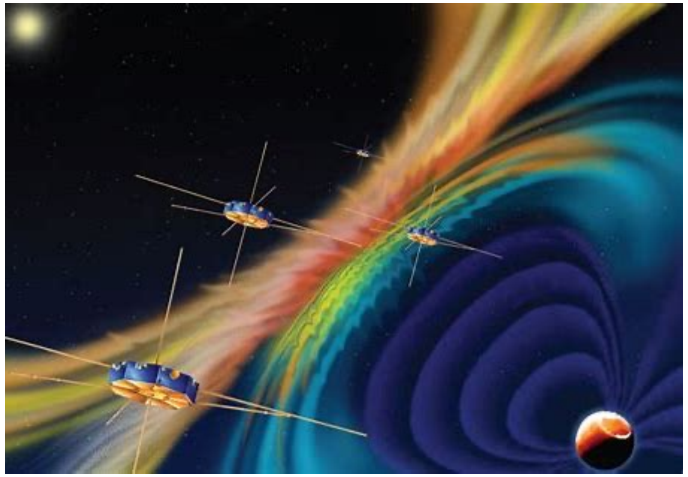
Plasma Turbulence, Grid: 2400x2400, PPC: 3200

Application in Magnetic Reconnection



Application in Magnetic Reconnection

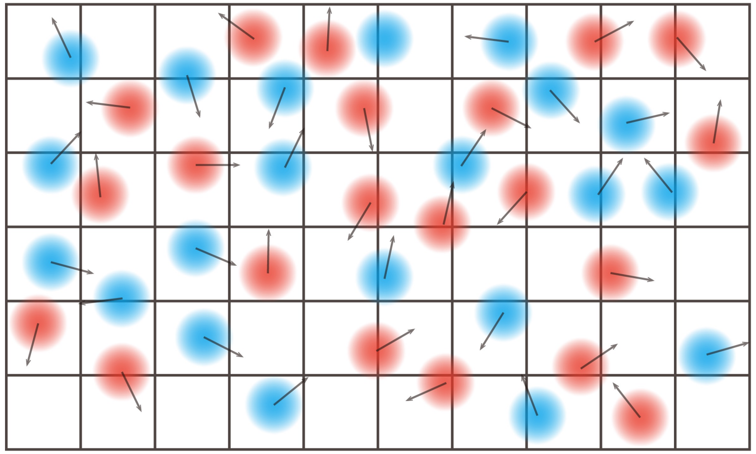
MMS Spacecrafts Observation
[Burch et al., 2016]



Data Resolutions

- FGM: 128 Hz;
- EDP: 8196 Hz;
- FPI: 150 ms for electron; 30 ms for ions.

GPIC Simulation Program
[Xiong, Huang, et al., 2023, 2024]



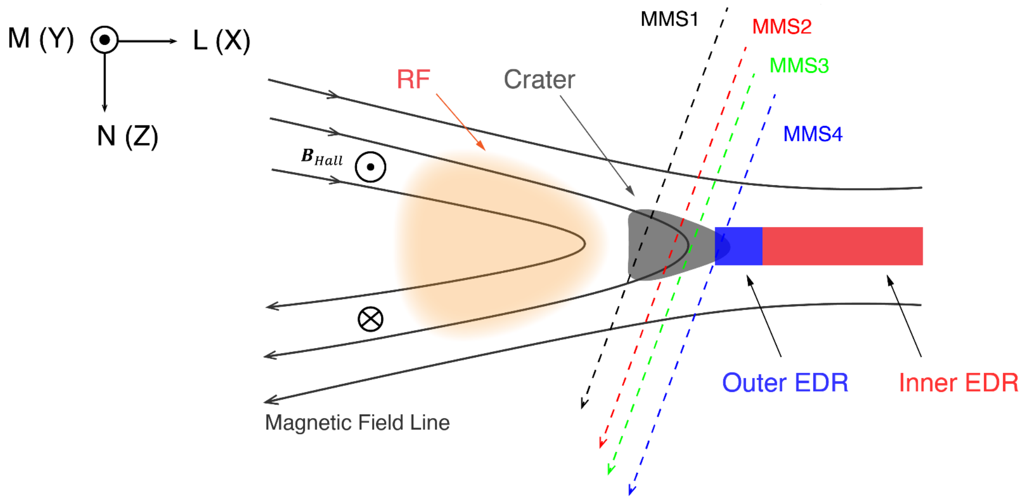
Basic Parameters:

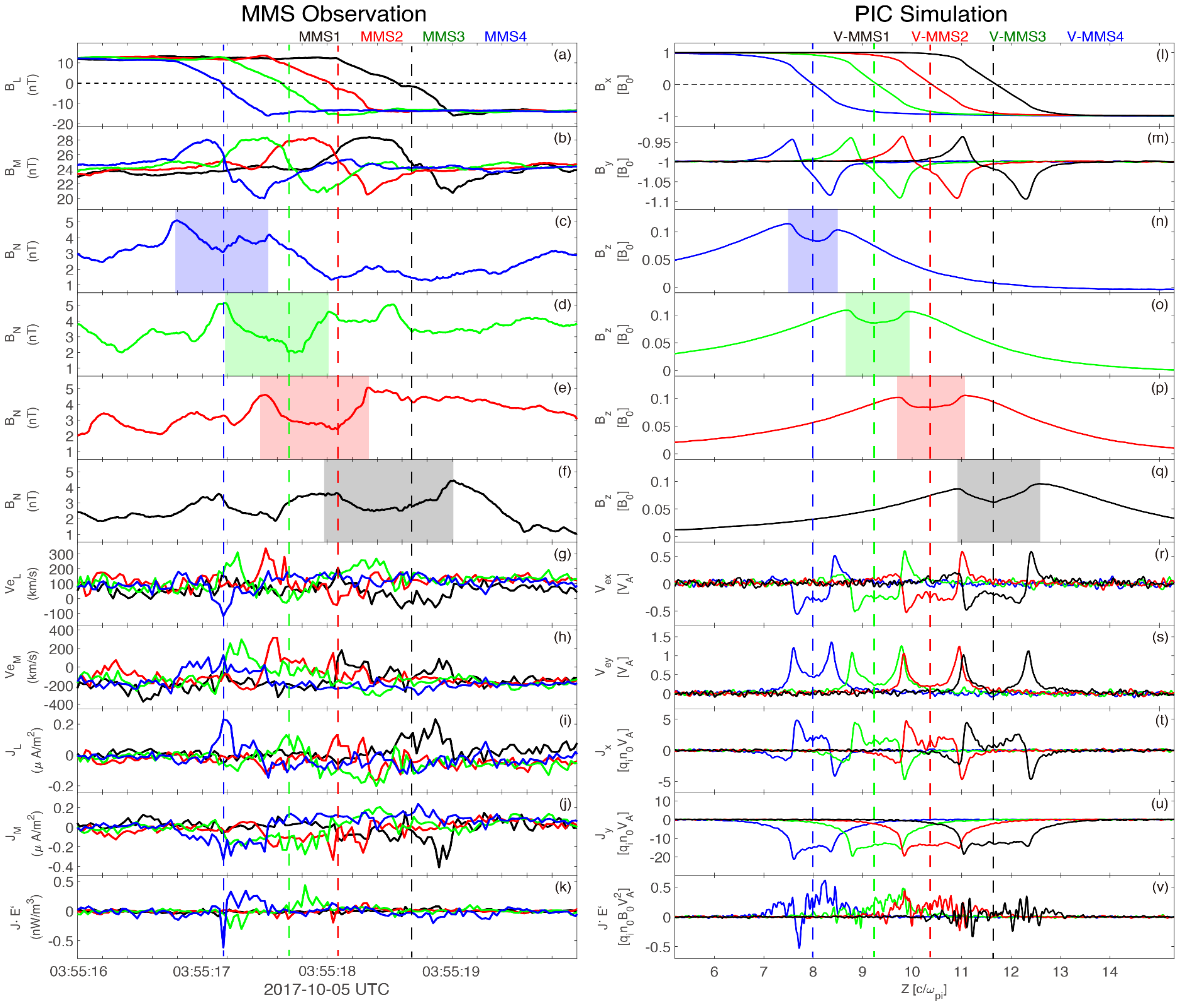
- Harris current sheet (2.5D); $B_x = B_0 \tanh(z/\lambda)$
- $m_i/m_e = 100$, $T_i/T_e = 5$, $\omega_{pe}/\omega_{ce} = 3$.
- Macro Particle Per Cell: 100

Application in Magnetic Reconnection (Ⅰ) – Crater Structure behind RF

Crater Structure Location

- The crater structure locates at the position in-between outer EDR (electron diffusion region) and RF;
- All four MMS spacecrafts cross the crater structure successively mainly along N direction.





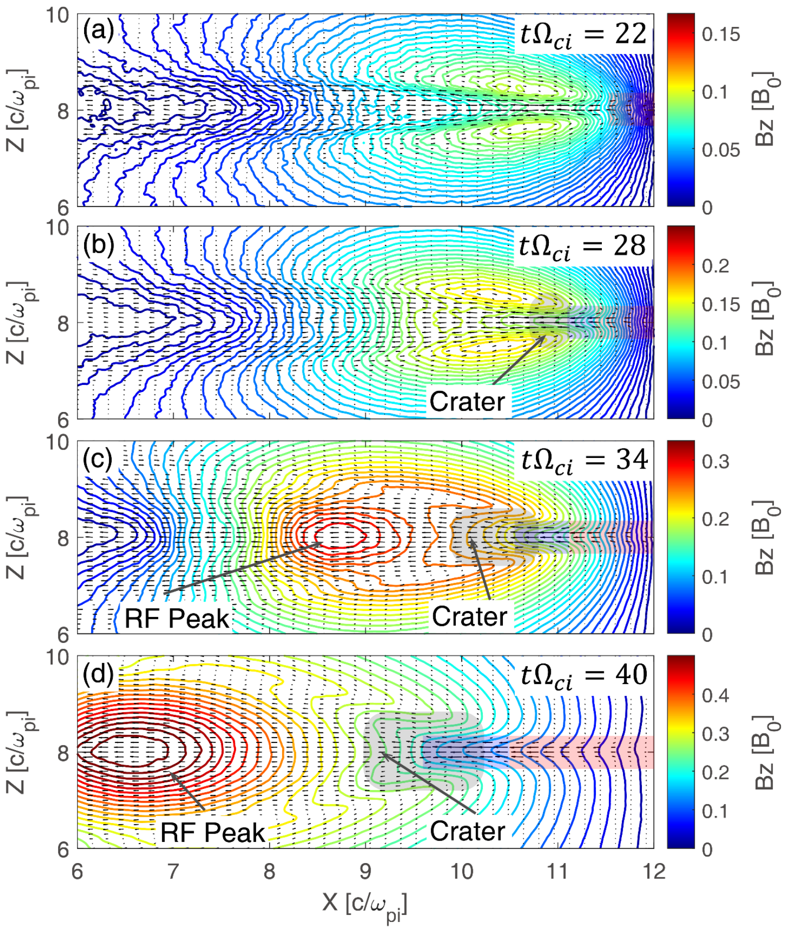
(Other Simulation Parameters:

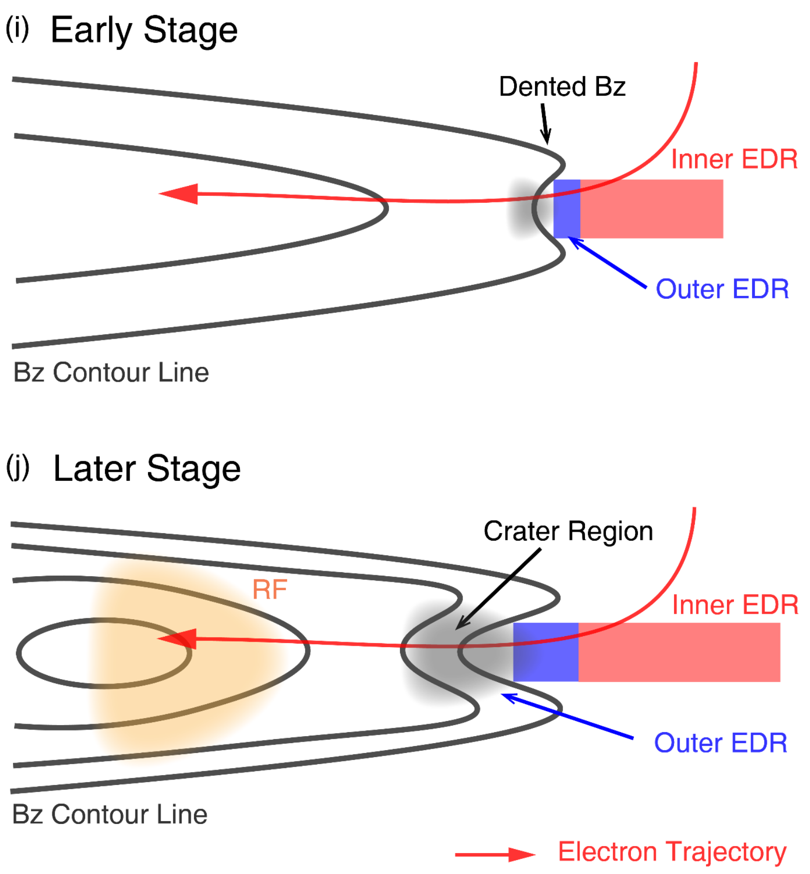
Grid: 1600 x 2400 ($32d_i x 48d_i$); Guide Field: $B_g = 0$ and 1)

Simulation results are highly consistent with observations!

Application in Magnetic Reconnection (I) – Crater Structure behind RF

Evolving Process of Crater Structure in Two-Dimensional Presentation:



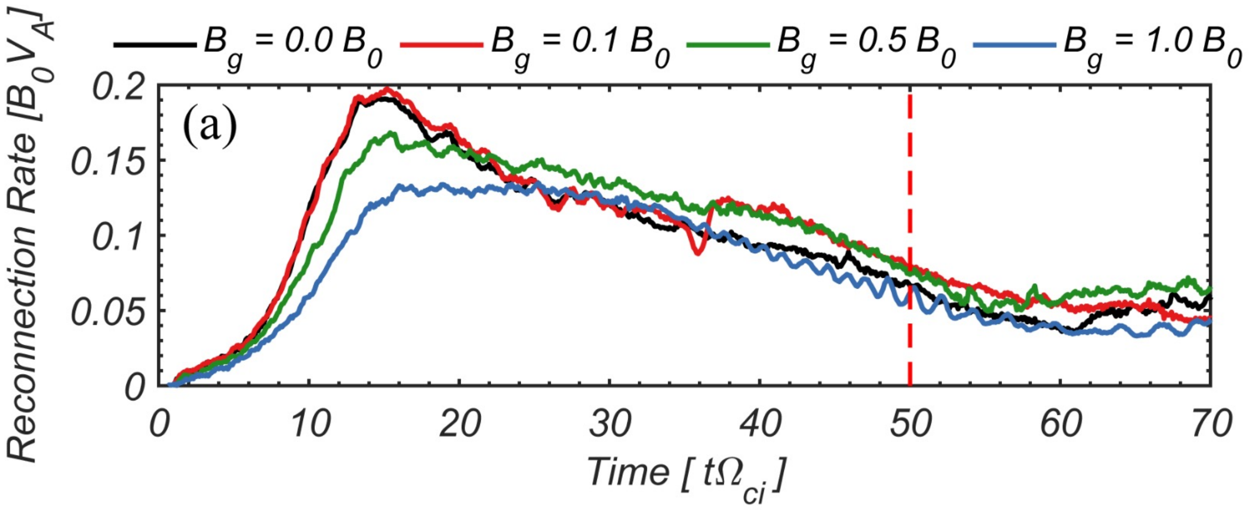


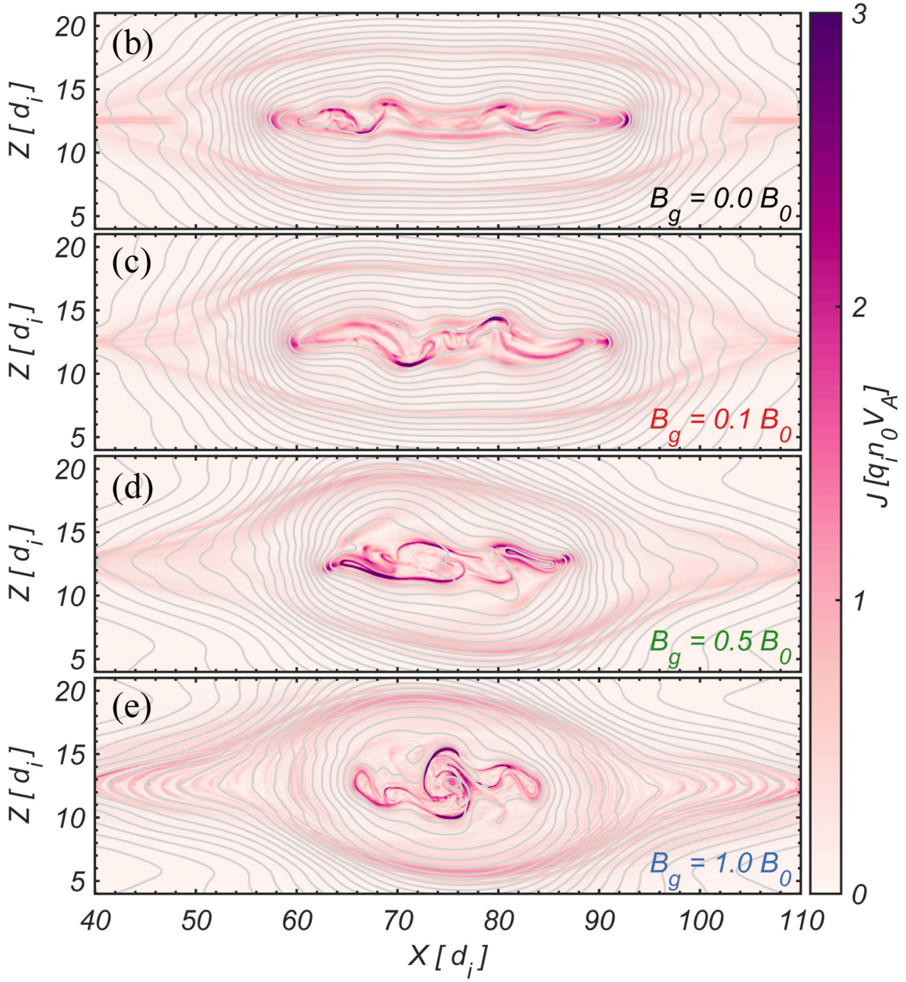
- At the early stage, the electron outflow velocity is relative low. The B_z only has a little dented trend, and RF has not formed yet;
- At the later stage, the high-speed electron outflow, like hot lava from active volcano eruption, constantly strikes the pileup region and makes B_z collapsed. Then, the crater structure is left behind RF.

Application in Magnetic Reconnection (II) –Turbulent Reconnection Outflow

Appearance of Turbulent Outflow

Status of Turbulent Outflow Under Different Guide Field Level





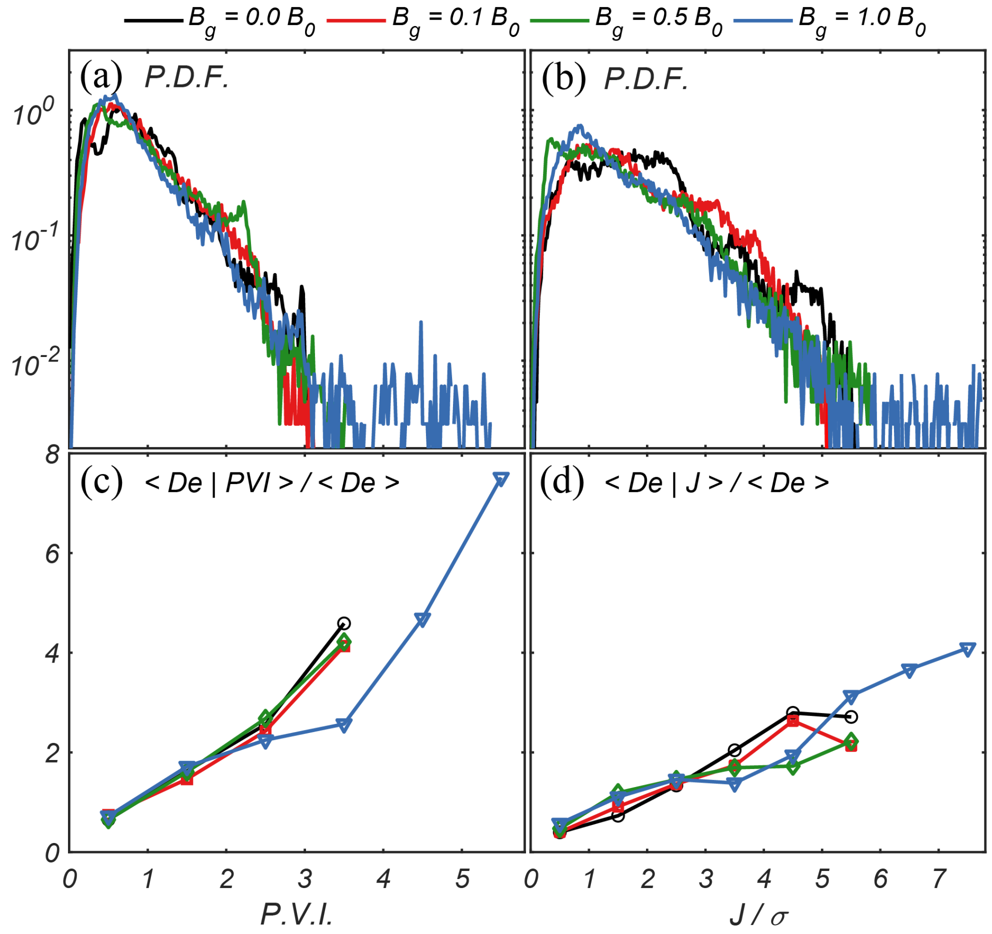
- ☐ Four runs are performed using different guide field level;
- ☐ Under larger guide field, reconnection outflow can be more chaotic, and more intense currents are generated.

(Other Simulation Parameters:

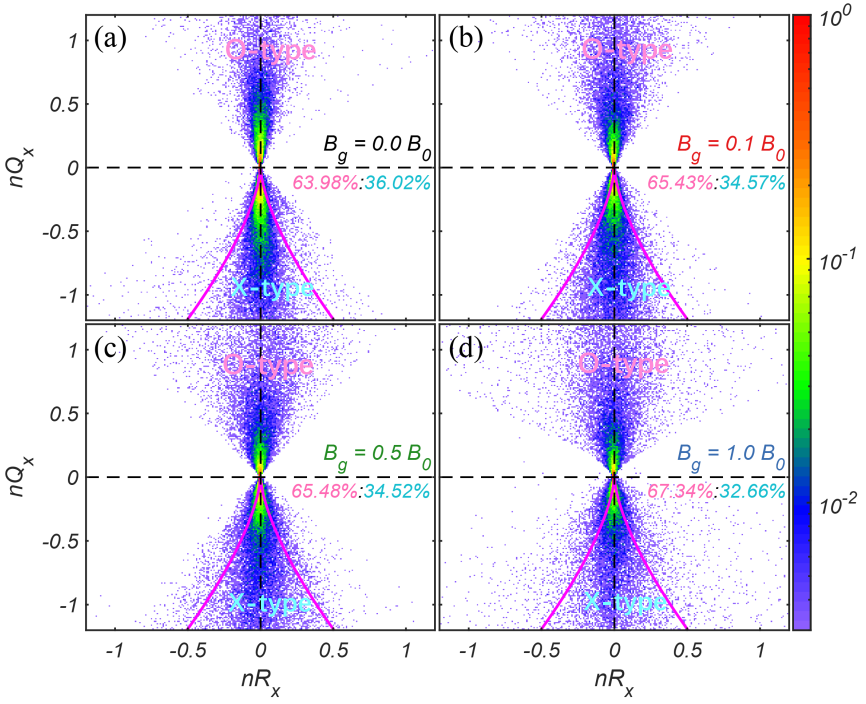
Grid: 6000 x 2000 (150d_ix 50d_i); Guide Field: $B_g = [0, 0.1, 0.5, 1.0]$)

Application in Magnetic Reconnection (II) –Turbulent Reconnection Outflow

Energy Conversion and Magnetic Topology in Turbulent Outflow



- ☐ The turbulent outflow with larger guide field can attain higher PVI and Current, associated with larger energy conversion;

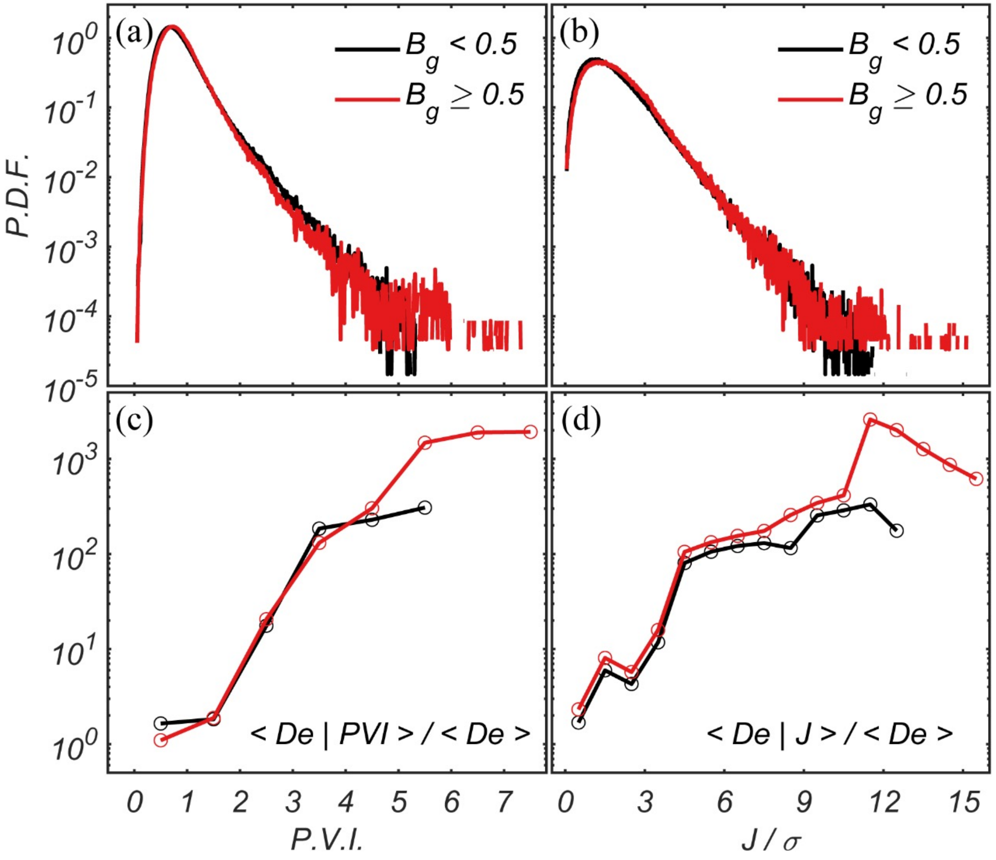


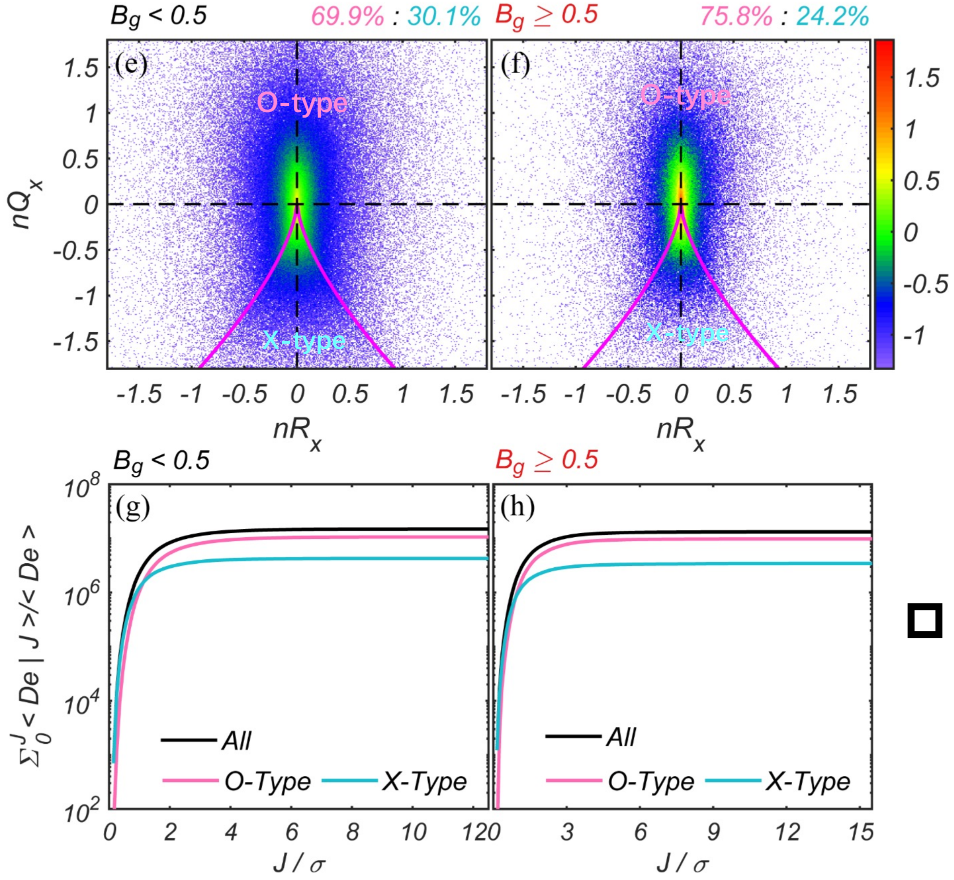
- ☐ Using the geometrical invariants, it is found that the larger guide field can promote the generation of O-type topology.

Application in Magnetic Reconnection (II) –Turbulent Reconnection Outflow

Energy Conversion in Turbulent Outflow

Evidence From MMS Observations (122 Events are Captured.)





☐ Well-consistent with the simulation results.

Summary

- **GPU computing can be applied in fully kinetic PIC simulation, and it can amazingly speed up computing process.**
- **A novel crater structure is found behind reconnection front via GPIC simulations and in-situ observations, which is caused by the high-speed electron outflow.**
- **Both simulations and observations show that Larger guide field can promote the generation of O-type topology structures and energy conversion in turbulent outflow.**

References:

[1] S. Y. Huang, Q. Y. Xiong, Z. G. Yuan, et al. (2024), Crater Structure Behind Reconnection Front. ***Geophys. Res. Lett.***, 51, e2023GL106581.

[2] S. Y. Huang, J. Zhang, Q. Y. Xiong, Z. G. Yuan, et al. (2023), Kinetic-scale Topological Structures Associated with Energy Dissipation in the Turbulent Reconnection Outflow, ***The Astrophysical Journal***, 958, 189, https://doi.org/10.3847/1538-4357/acf847

[3] Q. Y. Xiong, S. Y. Huang, J. Zhang, et al. (2024) Guide Field Dependence of Energy Conversion and Magnetic Topologies in Reconnection Turbulent Outflow. ***Geophys. Res. Lett.***, 51, e2024GL109356

[4] Q. Y. Xiong, S. Y. Huang, Z. G. Yuan, et al. (2024) GPIC: A Set of High-Efficiency CUDA Fortran Code Using GPU for Particle-in-cell simulation in space physics. ***Computer Phys. Comm.***, 295, 108994.

[5] Q. Y. Xiong, S. Y. Huang, Z. G. Yuan, et al. (2023) A Scheme of Full Kinetic Particle-in-cell Algorithms for GPU Acceleration Using CUDA Fortran Programming. ***Astrophys. J. Supp. S.***, 264, 3.

Thank You !