Application of GPU-accelerated Particle-in-cell Simulations in Magnetic Reconnection Associated with Energy Conversion between Field and Particles

Qiyang Xiong * & Shiyong Huang

School of Electronic Information, Wuhan University, Wuhan, China *Contact: qyxiong@whu.edu.cn

Supported by *Bharatkumar Sharma* & *Lvlin Kuang* From NVIDIA

1. Abstract

Magnetic reconnection is a fundamental physical process of rapidly converting magnetic energy into particles in space physics. The electron diffusion region (EDR), which can be split into the inner EDR and outer EDR, is the crucial region during magnetic reconnection. Here we present the studies associated with energy conversions around EDR using fully kinetic particle-in-cell (PIC) simulations of advanced GPU-accelerated computing and Magnetospheric Multiscale (MMS) mission observations. It is found that part of the electrons in the outer EDR are forced backward to the inner EDR by the magnetic tension force to be accelerated again, which we name it by magnetic Marangoni effect. And we also report a novel crater structure of magnetic field behind the reconnection front (RF) caused by the continuous impact of the high-speed outflow electron jets.

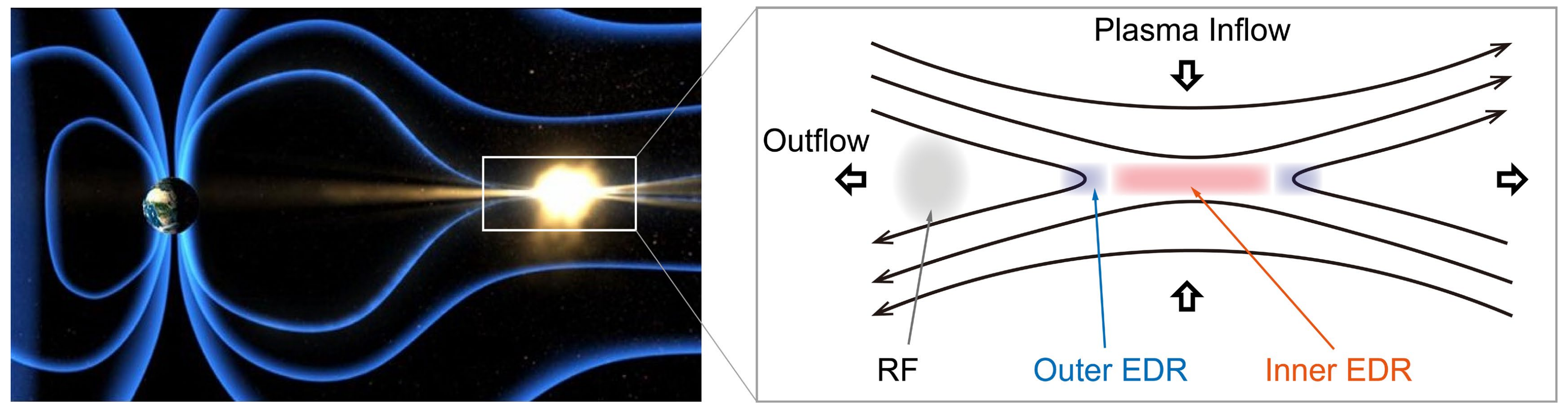


Figure 1. Sketch of magnetic reconnection in terrestrial magnetotail and its physical model.

2. GPU-Accelerated PIC Scheme

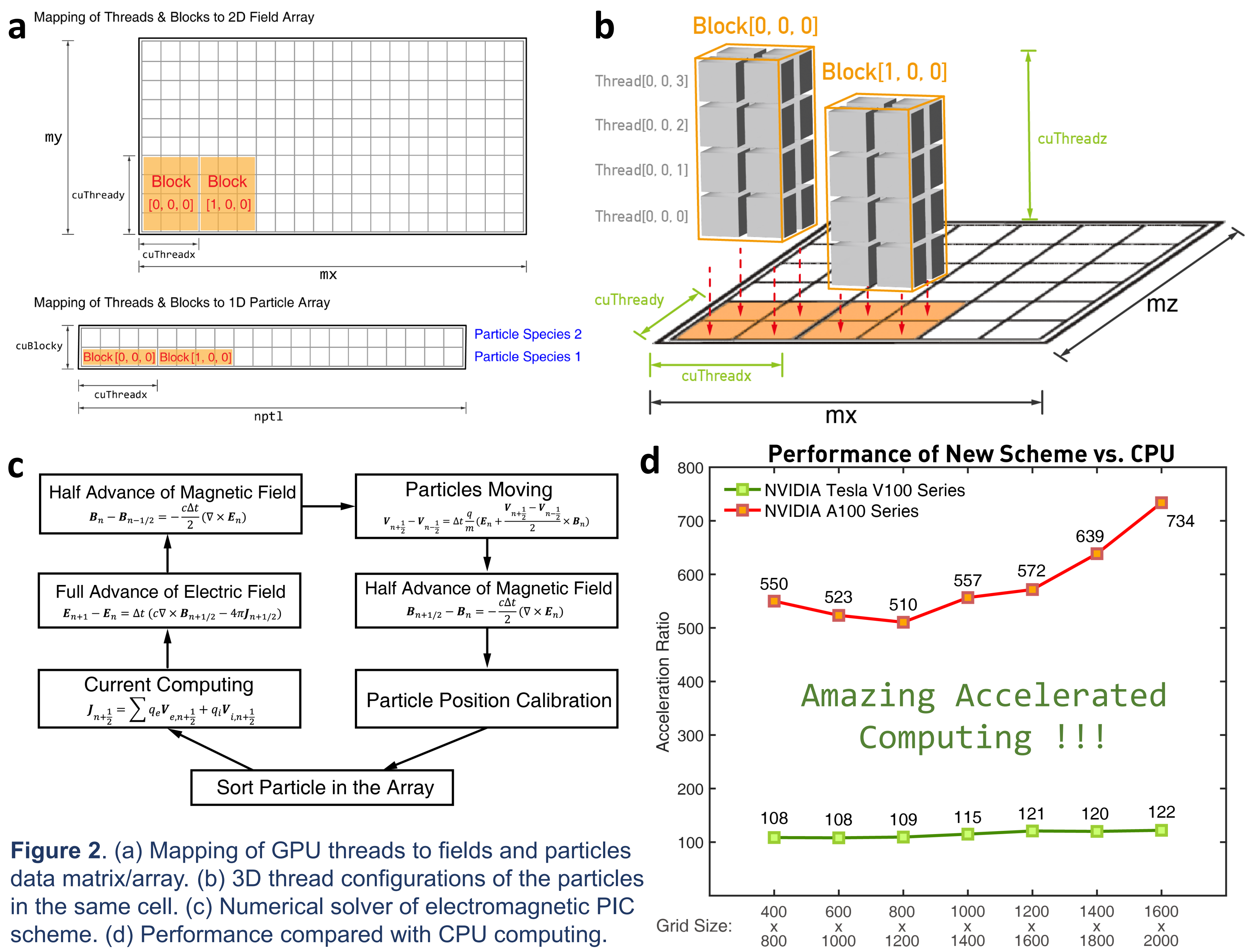

a
Mapping of Threads & Blocks to 2D Field Array
my
cuThready
Block [0, 0, 0]
Block [1, 0, 0]
cuThreadx
mx
Mapping of Threads & Blocks to 1D Particle Array
cuBlocky
Block[0, 0, 0] Block[1, 0, 0]
Particle Species 2
Particle Species 1
cuThreadx
nptl
b
Block[0, 0, 0]
Block[1, 0, 0]
Thread[0, 0, 3]
Thread[0, 0, 2]
Thread[0, 0, 1]
Thread[0, 0, 0]
cuThreadz
cuThready
cuThreadx
mz
mx
c
Half Advance of Magnetic Field
$B_n - B_{n-1/2} = \frac{c\Delta t}{2}(\nabla\times E_n)$
Particles Moving
Full Advance of Electric Field
$E_{n+1} - E_n = \Delta t\,(c\nabla\times B_{n+1/2} - 4\pi J_{n+1/2})$
Half Advance of Magnetic Field
$B_{n+1/2} - B_n = -\frac{c\Delta t}{2}(\nabla\times E_n)$
Current Computing
$J_{n+\frac{1}{2}} = \sum q_e V_{e,n+\frac{1}{2}} + q_i V_{i,n+\frac{1}{2}}$
Particle Position Calibration
Sort Particle in the Array
d
Performance of New Scheme vs. CPU
NVIDIA Tesla V100 Series
NVIDIA A100 Series
Acceleration Ratio
Amazing Accelerated Computing !!!
550 523 510 557 572 639 734
108 108 109 115 121 120 122
Grid Size:
400 x 800
600 x 1000
800 x 1200
1000 x 1400
1200 x 1600
1400 x 1800
1600 x 2000

Figure 2. (a) Mapping of GPU threads to fields and particles data matrix/array. (b) 3D thread configurations of the particles in the same cell. (c) Numerical solver of electromagnetic PIC scheme. (d) Performance compared with CPU computing.

3. Electron Backflow in the Outer EDR

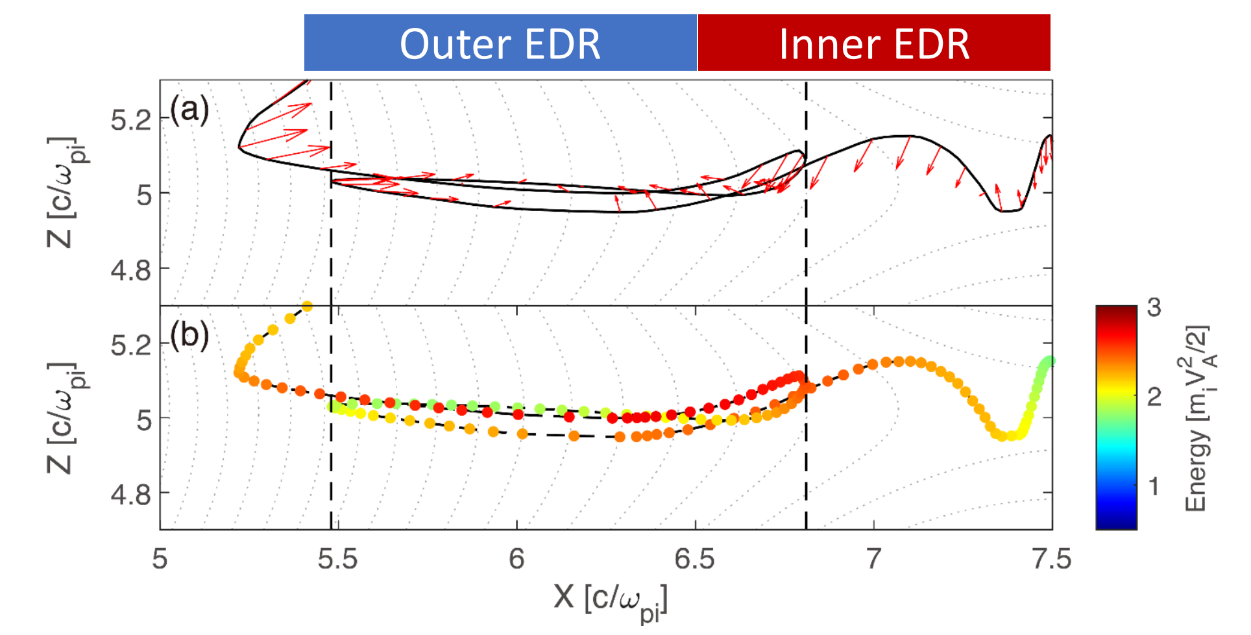


Figure 3. An instance of electron with backflow motion in the outer EDR. The black curves are the electron trajectories. Red arrows in (a) are the force condition at every trajectory points. This force is mainly contributed by magnetic tension force. Colored points in (b) represent the energy of electron at the corresponding points. The gray dashed curves are the magnetic field lines. This electron backflow motion in the outer EDR is named by ***magnetic Marangoni effect***.

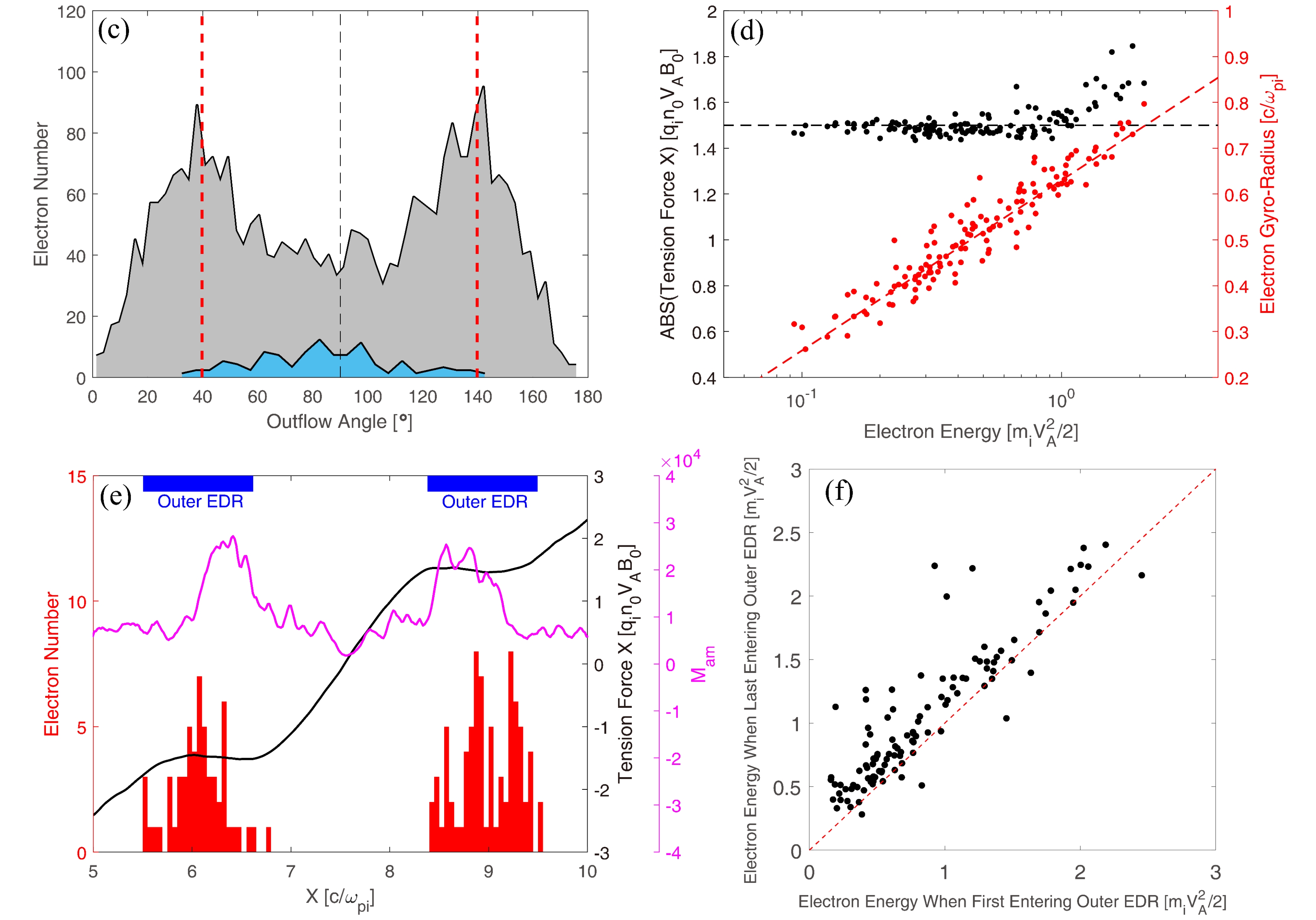


Figure 3. Statistical results of 110 electrons with similar movements. (c) Electron velocity angles vs. X direction when they enter the outer EDR. Blue distribution is the result of those 110 electrons, and the gray distribution is the overall electrons at this position. (d) Magnetic tension force condition (black dots) on those electrons and their gyro radius (red points) at the position where their moving directions are reversed. (e) Number of the electrons that begin to turn back (red histogram), magnetic tension force along X direction (black curve), and magnetic Marangoni number (M_{am}) (magenta curve) along X direction. ($M_{am} = U_0 d/D_m$, U_0 is characteristic velocity, d is layer thickness, and D_m is the magnetic diffusivity $D_m = (\mu_0\sigma_0)^{-1}$). (f) Electron final energy vs. initial energy when leaving outer EDR.

4. Crater Structure Behind RF

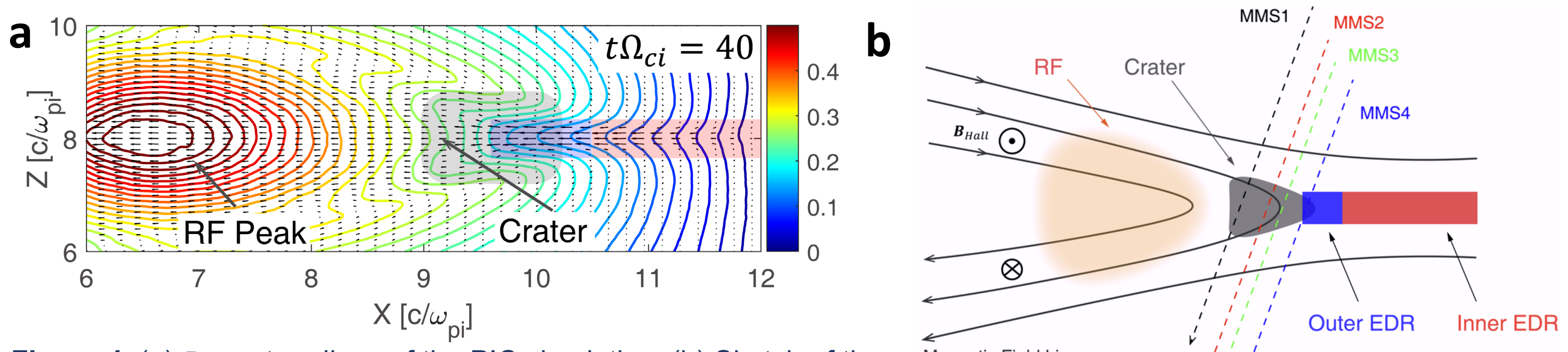


Figure 4. (a) B_z contour lines of the PIC simulation. (b) Sketch of the localized reconnection site. The dashed lines mark the MMS crossing pattern shown in the next figure.

4. Crater Structure Behind RF

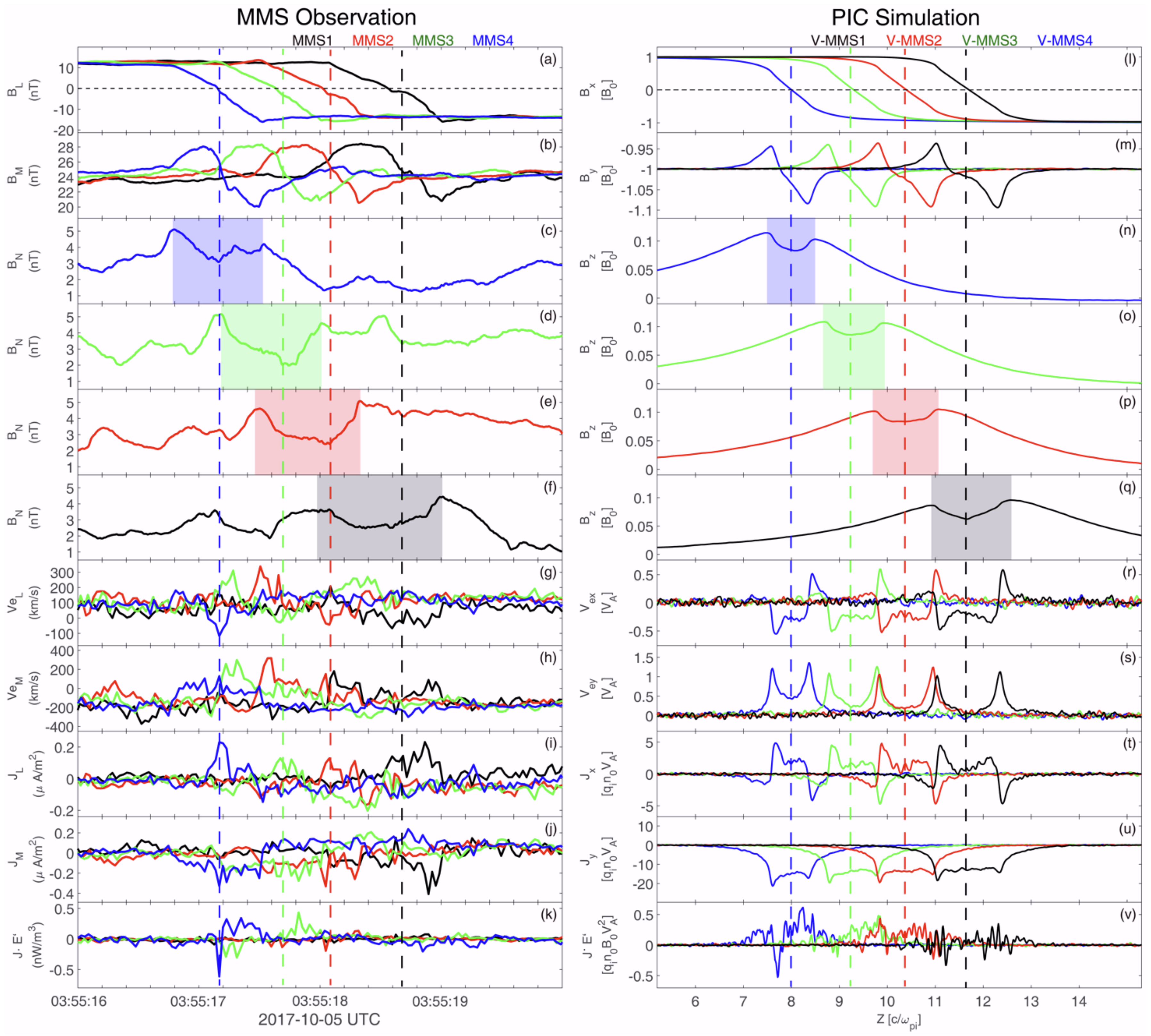


Figure 5. Comparison of MMS observation and PIC simulation results. The left part is 1D cuts along the trajectory as Figure 4 shows. The locations of the crater structure are highlighted by the colored squares.

5. Conclusions

- GPU-accelerated PIC scheme improves fast accessibility to the simulation results and provides valuable assistance in studying the physical process of magnetic reconnection.
- Electron backflow motion in the outer EDR, which is called magnetic Marangoni effect, can give part of electrons more chances to return inner EDR and be accelerated again. Thus, those electrons can attain higher energy level.
- The constant impact of the high-energy electron jets from the inner EDR on the pile-up region helps to form the crater structure behind RF. This structure could be an energy cache region where energy is transmitted between electrons and RF.

REFERENCES:

Xiong, Q., Huang, S., Yuan, Z., et al. (2023) A Scheme of Full Kinetic Particle-in-cell Algorithms for GPU Acceleration Using CUDA Fortran Programming. *The Astrophysical Journal Supplement Series*, 264, 3. DOI: 10.3847/1538-4365/ac9fd6

Xiong, Q., Huang, S., Yuan, Z., et al. (2024) GPIC: A Set of High-Efficiency CUDA Fortran Code Using GPU for Particle-in-cell Simulation in Space Physics. *Computer Physics Communications*, 295, 108994. DOI: 10.1016/j.cpc.2023.108994

Xiong, Q., Huang, S., Yuan, Z., et al. (2023) Electron Backflow Motions in the Outer Electron Diffusion Region During Magnetic Reconnection. *Geophysical Research Letters*, 50, e2023GL105300. DOI: 10.1029/2023GL105300

Huang, S., **Xiong, Q.**, Yuan, Z., et al. (2024) Crater Structure Behind Reconnection Front. *Geophysical Research Letters*, 51, e2023GL106581. DOI: 10.1029/2023GL106581

The Astrophysical Journal Supplement Series, 264:3 (7pp), 2023 January https://doi.org/10.3847/1538-4365/ac9fd6

A Scheme of Full Kinetic Particle-in-cell Algorithms for GPU Acceleration Using CUDA Fortran Programming

Q. Y. Xiong[1], S. Y. Huang[1,2], Z. G. Yuan[1], K. Jiang[1], Y. Y. Wei[1], S. B. Xu[1], J. Zhang[1], Z. Wang[1], R. T. Lin[1], and L. Yu[1]

[1] School of Electronic Information, Wuhan University, Wuhan, 430072, People's Republic of China; shiyonghuang@whu.edu.cn
[2] Hubei Luojia Laboratory, Wuhan, 430079, People's Republic of China

Received 2022 July 9; revised 2022 October 3; accepted 2022 November 1; published 2022 December 13

Abstract

The emerging computable devices, graphical processing units (GPUs), are gradually applied in the simulations of space physics. In this paper, we introduce an approach that implements full kinetic particle-in-cell simulations on GPU architecture devices using the CUDA Fortran language programming for the first time. Using the latest high-performance computing NVIDIA GPUs, this program, which follows the second-order leap-frog iteration method, can speed up the computing process by a factor of 150–285 on a single device compared with the time cost of running with a single core of an Intel Xeon Gold processor. Our scheme improves fast accessibility to the simulation results and provides valuable assistance in studying the physical process.

Unified Astronomy Thesaurus concepts: Solar magnetic reconnection (1504); GPU computing (1969)

1. Introduction

High-performance computing (HPC) architectures have undergone a remarkable evolution phase in the last decade. The central processing units (CPUs) designed for HPC are embedded with more cores and threads, and their clock rate is also raised, resulting in more powerful computability. Assisted by the application of the InfiniBand (IB) network and the Message Passing Interface (MPI), the traditional particle-in-cell (PIC) simulation program (e.g., P3D of Zeiler 2002; UPIC of Decyk 2007; iPic of Markidis et al. 2010; gcPIC of Lu et al. 2019) can run with a large domain size on multinodes and obtain the results in a short time. These simulation programs have been frequently used in previous studies and helped reveal the abundant physical mechanisms of plasma waves and magnetic reconnection and turbulence in terrestrial space, interplanetary space, and solar activities (e.g., Drake et al. 2006; Fu et al. 2006; Goldman et al. 2011; Winjum et al. 2013). Nevertheless, during the data communication or gathering process, the frequent data transmission operations between the nodes due to the distributed random access memory (RAM) may decrease the iteration efficiency.

Another emerging computing device, graphical processing units (GPUs), have been applied to numerical simulations recently. After introducing the Compute Unified Device Architecture (CUDA) by NVIDIA, GPUs are more actively participating in the computing industry through more than just displaying video frames on a monitor. Meanwhile, the video RAM (VRAM) on each computable GPU increases from hundreds of megabytes to dozens of gigabytes as the hardware of new generation products is upgraded. Therefore, it is nowadays technically possible to perform large-scale numerical simulations at once on a single GPU device. Besides, each GPU has billions of threads to execute instructions independently, unlike CPU architecture, thus it can master the corresponding amount of particle movements in PIC simulations and reach the execution of the maximum parallel instructions with a lower time cost. More importantly, all threads can access the VRAM directly on a GPU rather than the CPU's pattern, where data is accessed on RAM through a PCIE bus. Therefore, the bandwidth utilization of a GPU is significantly more advantageous than that of a CPU.

Various PIC simulation codes have been released in recent years designed for the GPU architecture devices (e.g., Decyk & Singh 2011, 2014; Burau et al. 2010; Abreu et al. 2011). These programs mainly write the host instructions in Fortran code and the CUDA kernels in CUDA C code, respectively. Therefore, two compilers are needed (e.g., `gfortran/ifort` for the Fortran codes; `nvcc` for the CUDA C codes) to obtain the Fortran program and CUDA kernels, and the CUDA kernels are called in the host codes. On the other hand, the first CUDA Fortran compiler (`pgi`) was developed more than 10 years ago, which aided convenient coding only using the Fortran language. Now, this compiler is embedded in NVIDIA HPC SDK[3] and evolves into a new compile command (`nvfortran`) that has more powerful functions than the `pgi` compiler. Under this circumstance, many characteristics and customs of Fortran programming can be reserved, and it becomes easier for the CPU codes to be implemented on a GPU device.

In this paper, we introduce, for first time, how to run the 2.5D-PIC simulation program on a GPU device using the CUDA Fortran programming approach. The technological process of a full kinetic PIC is given based on the electromagnetic field mode iteration. We compare the iteration results of both CPU and GPU computing under the same magnetic reconnection configuration, and this confirms the correctness of the GPU calculation results. Besides, the program is tested on different flagship computing–level GPUs issued recently to evaluate the performance. We also compare the computational efficiency between the CPU and GPU and propose possible improvements for the GPU program for a future update.

[3] https://docs.nvidia.com/hpc-sdk/

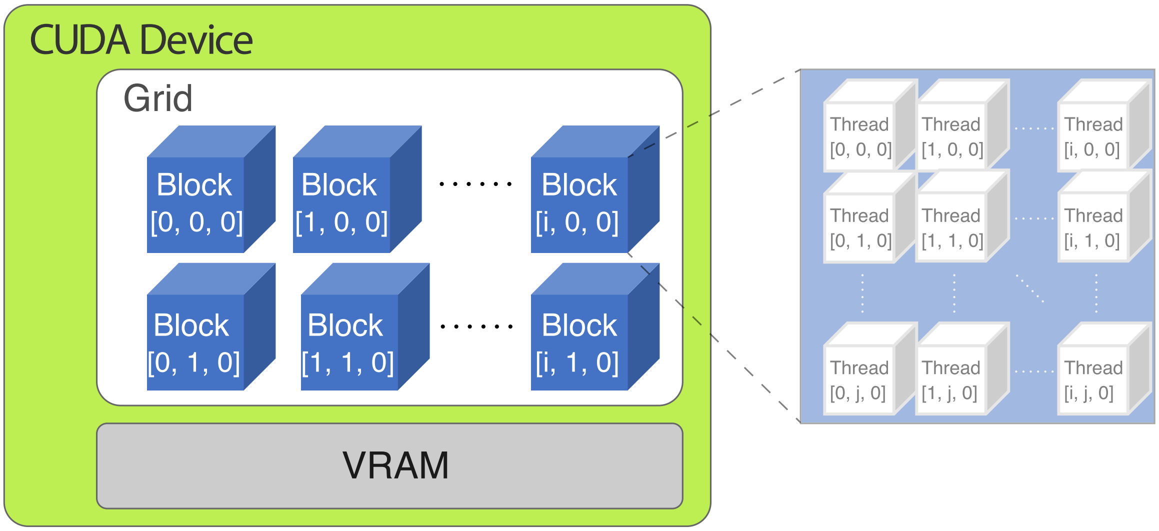


Figure 1. The brief architecture of a GPU device. The blue cubes are the blocks of the GPU in the grid. The white cubes are the threads in each block. The gray area is the GPU on-device memory.

2. Simplified Sketch of a Single-GPU Architecture

Figure 1 displays the basic architectural configuration of a GPU device. It mainly consists of two parts: the calculation units of the grid matrix and the memory on the device. The first layer of the computing unit consists of quite a lot of blocks in three dimensions (blue blocks in Figure 1), and the second layer embedded in the block is called the thread (white blocks in Figure 1). Each thread can be allocated with instruction commands and executes them independently. Different types of GPU usually have different specification parameters, and these specifications usually depend on the architecture design of the GPU device. For example, the GPU model NVIDIA Tesla V100-SXM2-16 GB has a maximum of 1024 threads in each block, and the maximum thread number of each dimension is [1024, 1024, 64]. Meanwhile, there are 2,147,483,647 blocks in the grid, and the maximum block number of each dimension is [2,147,483,647, 65,535, 65,535]. The upper limit of the block or thread number determines how much of the computing resource can be launched in the program.

3. Mapping of the Thread and Block Index to the Data Array Index

When calling the CUDA kernels in the host codes, the thread number in each block and the block number in the grid must be specified. The predefined variable `threadIdx` (the index of a thread within its block), `blockDim` (the number of threads in a block), and `blockIdx` (the index of the block within the grid) provide the offset to refer to each thread in different blocks. And the correspondence of each thread to each cell of the simulation domain should be built so that the threads can access the unique address of the data unit correctly.

The field data (e.g., magnetic field, electric field, etc.) is stored in a two-dimensional (2D) array. We pick the X component of magnetic field (B_x), for example, which is declared as `bx(mx,my)` in code. The parameter `mx` is the array length in the X direction, and `my` is the one in the Y direction. Under the GPU architecture, each thread launched by the program must correspond to a specific simulation grid point `bx(i,j)`, as mentioned above. Figure 2(a) shows an instance of the 2D global array index reference from the thread and block perspective. The orange square stands for a single block in a GPU device. In the case of Figure 2(a), it is settled that each block contains 3 × 4 × 1 threads, i.e., three threads in the X dimension and four threads in the Y dimension. Now that the global array size is `mx × my`, the program needs the block number of `ceiling(mx/3) × ceiling(my/4)` to cover all the array maps and avoid out-of-bounds memory access. Therefore, the global array index `(i,j)` for 2D field data can be written as:

$$i = (\text{blockIdx}\%x - 1)*\text{blockDim}\%x + \text{threadIdx}\%x$$
$$j = (\text{blockIdx}\%y - 1)*\text{blockDim}\%x + \text{threadIdx}\%y.$$

The particle array data (e.g., electron and ion positions, velocity, etc.) use another method for indexing. We also choose the X position of the particle, for example, which is declared as `x(nptl,2)` in code. The parameter `nptl` is the number of a species particle, and the second direction array size is set as 2 to present the two-particle species (electron and ion). In the code, the global particle array index is referred to as `x(i,j)`. Figure 2(b) illustrates the block and thread configurations for the particle array. For each block, we only launch a 1D thread (like 4 × 1 × 1, as Figure 2(b) shows) of each block for the program. And the Y-dimension block number is fixed at 2 on account of the two-particle species; therefore the block index along the Y direction can distinguish the particle species and determine the parameters like charge and mass. As a result, the number of blocks that should be launched to execute kernels in the program can be calculated as `ceiling(nptl/4) × 2`. Consequently, the global array index for the particle data is given as follows:

$$i = (\text{blockIdx}\%x - 1)*\text{blockDim}\%x + \text{threadIdx}\%x$$
$$j = \text{blockIdx}\%y.$$

4. Scheme of a Full Kinetic PIC Simulation

The physical algorithm of electromagnetic field evolution basically follows the Maxwell and relative Newton–Lorentz equations. And the numerical iteration method has been proposed or used in previous studies (e.g., Birdsall & Langdon 1991; Matsumoto & Omura 1993). The scheme of

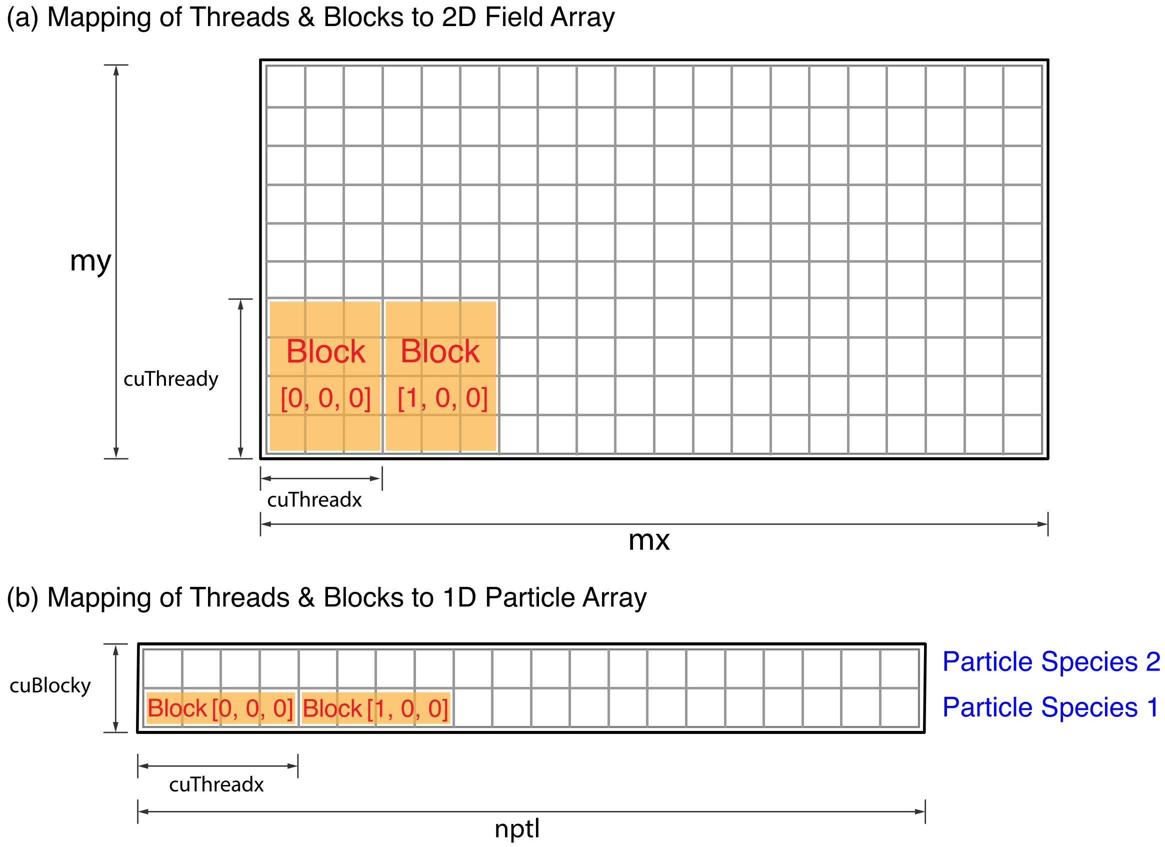


Figure 2. Mapping of the global threads index to the array index in 2D field data (a), and 1D particle data (b). The orange squares are the blocks in the GPU device. (a) mx and my are the field-data length along the X and Y directions, respectively. *cuThreadx* and *cuThready* are the thread number in the X and Y dimensions launched in each block. (b) nptl is the total particle number of each species. *cuBlocky* is the block number in the Y dimension and it is fixed at 2.

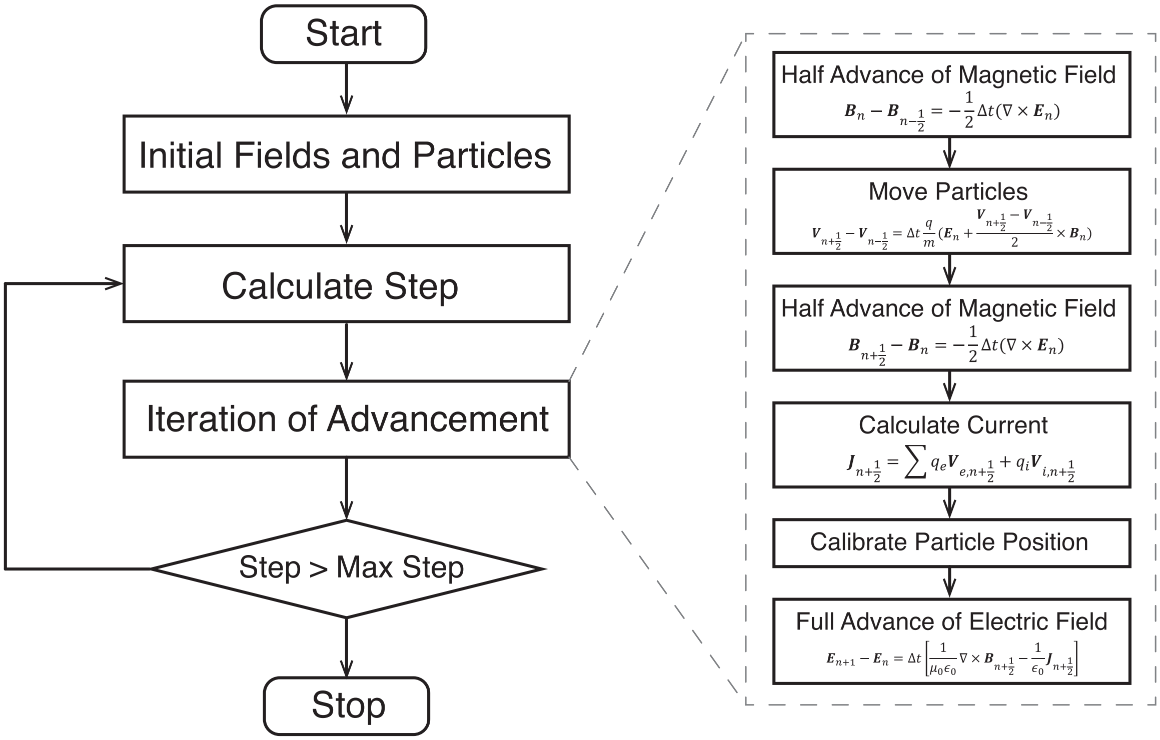

Start
Initial Fields and Particles
Calculate Step
Iteration of Advancement
Step > Max Step
Stop
Half Advance of Magnetic Field
$\boldsymbol{B}_n - \boldsymbol{B}_{n-\frac{1}{2}} = -\frac{1}{2}\Delta t(\nabla \times \boldsymbol{E}_n)$
Move Particles
$\boldsymbol{V}_{n+\frac{1}{2}} - \boldsymbol{V}_{n-\frac{1}{2}} = \Delta t \frac{q}{m}(\boldsymbol{E}_n + \frac{\boldsymbol{V}_{n+\frac{1}{2}} - \boldsymbol{V}_{n-\frac{1}{2}}}{2} \times \boldsymbol{B}_n)$
Half Advance of Magnetic Field
$\boldsymbol{B}_{n+\frac{1}{2}} - \boldsymbol{B}_n = -\frac{1}{2}\Delta t(\nabla \times \boldsymbol{E}_n)$
Calculate Current
$\boldsymbol{J}_{n+\frac{1}{2}} = \sum q_e \boldsymbol{V}_{e,n+\frac{1}{2}} + q_i \boldsymbol{V}_{i,n+\frac{1}{2}}$
Calibrate Particle Position
Full Advance of Electric Field
$\boldsymbol{E}_{n+1} - \boldsymbol{E}_n = \Delta t\left[\frac{1}{\mu_0\epsilon_0}\nabla \times \boldsymbol{B}_{n+\frac{1}{2}} - \frac{1}{\epsilon_0}\boldsymbol{J}_{n+\frac{1}{2}}\right]$

Figure 3. Flow chart of the full kinetic-PIC simulation using a GPU device. The dashed rectangle in the right part includes subroutines of the iteration. $\boldsymbol{B}$ is the magnetic field, $\boldsymbol{E}$ is the electric field, $\boldsymbol{V}$ is the particle velocity where $\boldsymbol{V}_e$ is the electron velocity and $\boldsymbol{V}_i$ is the ion velocity, $\boldsymbol{J}$ is the current, and q_e and q_i are the electron and ion charge, respectively. Δt is the time interval of each iteration step.

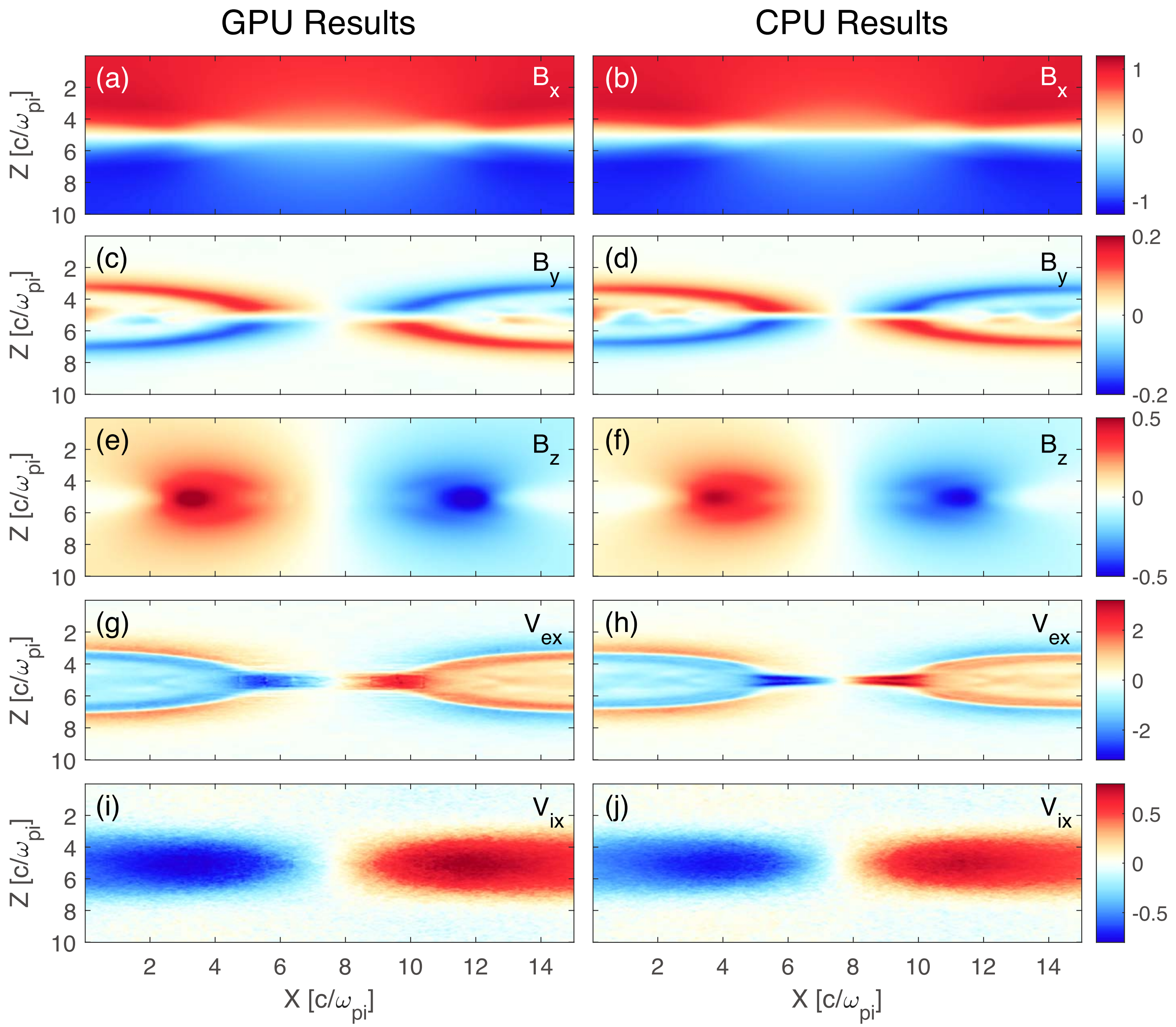


Figure 4. Comparison of the results calculated by the GPU and CPU, respectively. The simulations run at $t\Omega_{ci} = 36$. (a), (b) X component of magnetic field. (c), (d) Y component of magnetic field. (e), (f) Z component of magnetic field. (g), (h) X component of electron bulk velocity. (i), (j) X component of ion bulk velocity.

the global program and detailed iteration sequence is given in Figure 3. As for the left part, the Diagnose part for outputting the data is not included in the flow chart but has been written in the source code. In the Diagnose part, the raw data of the 2D fields are smoothed, and the particle data are calculated and gathered into the momentum form (which is also the 2D array) as the outputs when it encounters the given diagnose moments. The detailed method of the data diagnosis is not the primary technical implementation in this paper. Therefore, we briefly discuss the Diagnose part, and the time cost of this operation is excluded during the performance benchmark.

The subroutines in the iteration part all execute the computing instructions on the GPU device. The subscript of each variable (e.g., n, $n - 1/2$, $n + 1/2$, $n + 1$) represents the current step state. The overall routine is applied with the second-order leap-frog method for the time advancing, where the magnetic field updates twice every half step ($\Delta t/2$) and the electric field updates once every step (Δt). The advancement of the magnetic field uses Faraday's law. The moving of particles uses the Newton–Lorentz equation, and the relativity has been considered in the program but not shown there. The conducting current $\boldsymbol{J} = q_i n_i \boldsymbol{V}_i + q_e n_e \boldsymbol{V}_e$ is calculated through the particles' momentum data and moving distance $\boldsymbol{J} = \sum[q(\boldsymbol{x}_1 - \boldsymbol{x}_0)/\Delta t]$, where $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are the particles' old and new positions, respectively. Those particles that run out of the boundary should be considered. And the periodic boundary condition is applied in this program, calibrating the particles' positions for the next iteration step. Finally, the advancement of the electric field uses Ampere's law.

5. Algorithm Implementation Using CUDA Fortran

On account of achieving maximum thread parallelism computing on the GPU device, the CUDA Fortran code has certain different features compared with the traditional Fortran code (e.g., Fatica & Ruetsch 2014). The essential part lies in

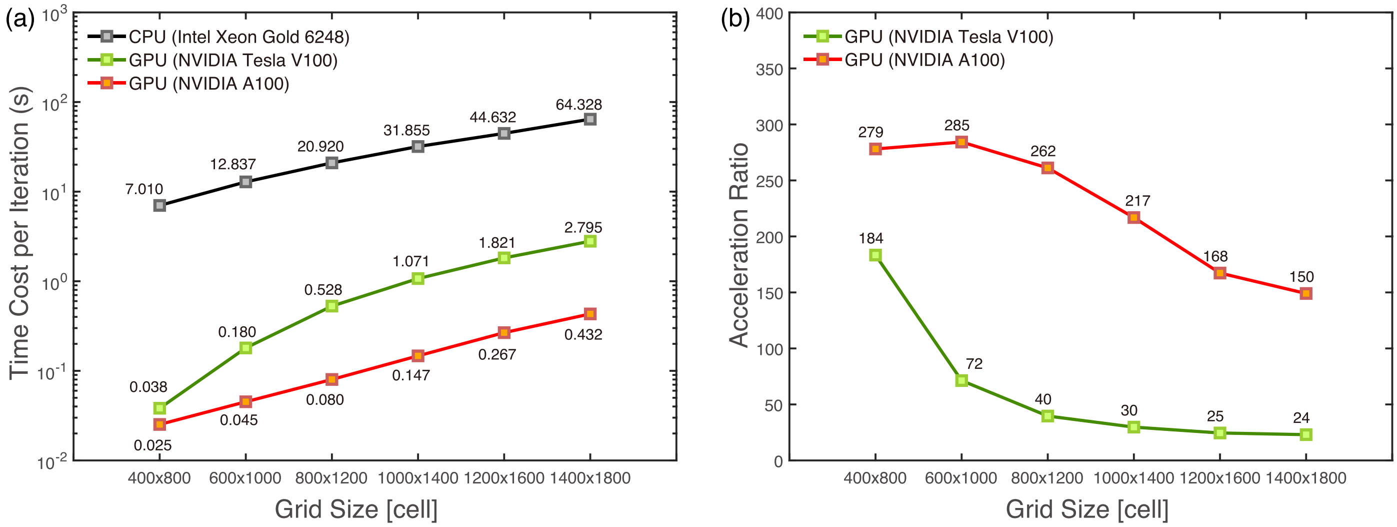


Figure 5. Benchmark results on a CPU and two types of GPU. (a) The time cost of the CPU (gray line), GPU NVIDIA Tesla V100 (green line), and NVIDIA A100 (orange line). (b) The acceleration ratio of these two types of GPU vs. the CPU.

the CUDA kernel subroutine, which has the prefix `attributes(global)` or `attributes(device)`, and is usually declared in the module part:

```
module kernel_iteration
   implicit none
   contains
      attributes(global) subroutine AdvanceMAG(...)
      attributes(device) subroutine GetCurrent(...)
end module
```

where `AdvanceMAG` and `GetCurrent` are the subroutines that update the magnetic field and calculate the particle current, respectively. More specifically, the subroutines with the global attribute can be called in both the host code (main program) and the device code (CUDA kernels), but those subroutines with the device attribute can only be called in the device code. After being compiled, these special subroutines are executed on the GPU device rather than the common subroutines running on the host. Meanwhile, it is necessary to provide the executed configurations when calling these subroutines. During these steps, the number of threads and blocks activated on the device by the host should be determined, for example:

```
call AdvanceMAG≪thds, blks≫(bx, by, bz, ex, ey, ez)
```

where `thds, blks` are the thread and block numbers launched on the device in this subroutine, and `bx, by, bz, ex, ey, ez` are the subroutine input variables (three components of the magnetic field and electric field). The detailed specifications of these subroutines guarantee the correctness and efficiency during the computing.

The former text has mentioned two kinds of array index forms for the field and particle parallelism (Figure 2). The standard numerical method of advancing the field refers to the finite difference (FD), which means the differential equation like $\partial B/\partial x$ is replaced by $(B(x+1)-B(x))/\Delta x$ in the meshed grid. In the traditional CPU calculation code, the loop for indexing each grid point `(i,j)` is assisted with the "do; end do" formation. In the case of the FD, the half advance of the magnetic field (Figure 3) under 2.5D configuration in Fortran code is organized as:

```
do j = 2, my-1
   do i = 2, mx-1
      bx(i,j=bx(i,j)-0.5*(ez(i,j+1)-ez(i,j))
      by(i,j)=by(i,j) + 0.5*(ez(i+1,j)-ez(i,j))
      bz(i,j) = bz(i,j) + 0.5*(ex(i,j+1)-ex(i,j)-ey(i+1,j) + ey(i,j))
   end do
end do
```

The array index of the 2D field is from the second to the last but one (2 to *my*-1 or *mx*-1), for which it is the boundary consideration. These commands mean that the CPU will traverse the whole 2D array successively. However, the CUDA Fortran code fulfilling the maximum parallelism is written as follows:

```
if(j >=2 .and. j<=my-1 .and. i >=2 .and. i<=mx-1)
    bx(i,j) = bx(i,j)-0.5*(ez(i,j+1)-ez(i,j))
    by(i,j) = by(i,j) + 0.5*(ez(i+1,j)-ez(i,j))
    bz(i,j) = bz(i,j) + 0.5*(ex(i,j+1)-ex(i,j)-ey(i+1,j) + ey(i,j))
end if
```

As mentioned before, each thread launched by the program can be assigned to a global array index `(i,j)`. If the block and thread number are fixed, each thread can master the calculation process of a unique grid point. Therefore, unlike the sequenced computing of CPU, the GPU device can simultaneously launch multithreads to finish the FD calculation of the corresponding grid point separately.

For the aspect of the particles, the *Y*-dimension length of the global particle array is fixed at 2, where the first `(j = 1)` and the second `(j = 2)` column store the ion and electron data, respectively, of each variable. That is why we launch two blocks in all *Y* dimensions in the GPU device: the first and the second blocks are responsible for the calculations of ions and electrons, respectively. Since the block index can distinguish the particle species at the Y dimension `(blockIdx%y)`, the ion- and electron-moving processes can be executed simultaneously within the same subroutine. The physical quantities concerning the different particle species can be configured as follows (taking ion and electron charge for example):

$$q = (j - 1) \ * \ qe - (j - 2)^{*}qi,$$

where `qe` and `qi` are the electron and ion charge, and `q` is the common charge used in the calculation formula. For the thread within the block where `blockIdx%y` equals 1, it obtains the common charge `q = qi` and calculates the variable of the ion species. If the block index at the Y dimension equals 2, it is the process that finishes the calculation concerning the electron. Based on this kind of design, the particle's movement can call the same subroutine but only needs to pay attention to the particle's mass and charge selection.

Another important issue about the particles is the current calculation. Usually, in sequent indexing, the conducting current contributed by the particle moving is the sum value of each particle. That means the accumulation command like `jx(i,j) = jx(i,j) + q*dx` is used in the traditional Fortran code. In the GPU environment, however, this command will give the wrong result during the update of the accumulated variable each time. First, all the particles participate in the calculation simultaneously on the GPU device. Then, it is confirmed that quite a few particles are located in the same cell in the grids. Therefore, the current contribution from these particles is added to this same cell. Indexing and accumulating the value to the same cell at once may cause a conflict because it is unpredictable which thread may access the value in this cell first. To prevent this situation, the atomic operation could be introduced in CUDA Fortran code as:

$$\begin{aligned}
&\mathrm{jxold} = \mathrm{atomicadd}(\mathrm{jx}(i, j),\ q^{*}dx/\Delta t)\\
&\mathrm{jyold} = \mathrm{atomicadd}(\mathrm{jy}(i, j),\ q^{*}dy/\Delta t)\\
&\mathrm{jzold} = \mathrm{atomicadd}(\mathrm{jz}(i, j),\ q^{*}\mathrm{Vz})
\end{aligned}$$

in a way where the left side of the command is the old data that records the value of the last writing operation, and this old value is nearly useless. In this algorithm, the main principle of the atomic operation is that it blocks other threads from accessing the data being operated and makes it invisible until the present operation on this data is finished. This process avoids crashes when the threads read and write data at the same physical address.

6. Benchmark Results on Modern HPC GPUs

Figure 4 displays the iteration results of magnetic reconnection obtained by CPU and GPU computing, respectively. The reconnection model is configured under double Harris current sheets with localized perturbation (e.g., Zhou et al. 2012; Huang et al. 2014, 2015; Xiong et al. 2022a, 2022b, 2022c). Some basic physical parameters are listed below: the ion inertial length (d_i) is 40 grids, and the mass ratio between ions and electrons (m_i/m_e) is fixed at 25; the temperature ratio between ions and electrons (T_i/T_e) is 5, and the frequency ratio of electrons (ω_{pe}/ω_{ce}) is 3; the simulation domain size is 800 × 1200 grids, and there are 100 pairs of ions and electrons in each cell, which means that 0.96×10^8 particles participate in the reconnection. The physical quantities presented in Figure 4 are all normalized. The magnetic field is normalized by the background magnetic field (B_0), and the particle velocity is normalized by the Alfvén speed (v_A).

The reconnection signatures can be detected as follows: current sheet contraction (Figures 4(a), (b)), quadrupole sign of the Hall magnetic field (Figures 4(c), (d)), reconnection front (Figures 4(e), (f)). Also, the high-speed electron and ion outflow jets are formed during the reconnection (Figures 4(g)–(j)). The separatrix of the reconnection can be clearly figured out through the Hall magnetic field (Figures 4(c)–(d)) and V_{ex} (Figures 4(g)–(h)). As the two sides of Figure 4 show, CPU and GPU computing results are consistent, indicating the correctness of the CUDA Fortran code. Noticeably, some localized distribution of these physical quantities introduces a slight difference between the results from the CPU and GPU. This phenomenon lies in the different modes of the random number generation at the host and the device. On account of this issue, the initial thermal velocity of the particles, which is controlled by the random number, cannot all be guaranteed to be the same in both CPU and GPU computing. Nevertheless, this neglected difference merely affects the correctness of the reconnection evolution itself.

Meanwhile, the time cost of each iteration step is also measured in both CPU and GPU computing. For the CPU part, the benchmark is carried out on the Intel Xeon Gold 6248 chip model. The equivalent time cost of a single core is estimated with the result of 40-core parallelism using the MPI. As for the GPU part, we select two modern HPC GPUs: the NVIDIA Tesla V100 (model: V100-SXM2-16GB) and NVIDIA A100 (model: A100-SXM4-40GB), which were initially released in the years 2017 and 2020, respectively. All the benchmark results are the average value of 10,000 steps' iteration time for reducing the accidental error. All the benchmark tests use the same GPU execution configuration: 16 × 16 threads in each block with a 2D fields treatment, and 128 × 1 threads in each block with a 1D particles treatment. Additionally, six different simulation domain sizes are chosen to investigate the changes of the time cost. The benchmark results are shown in Figure 5. It is apparent that the time cost of each iteration step is remarkably decreased on the GPU, especially on the newer generation GPU (NVIDIA A100) (Figure 5(a)). As the simulation scale increases, all the time costs raise accordingly. This is not surprising in the GPU case because the GPU device has a limited stream processor (SP) number to cover all the threads.

Based on the time cost, the computing acceleration ratio between the GPU and CPU can be obtained (Figure 5(b)). In a small-scale simulation (e.g., a grid size of 400 × 800), the computing efficiency on a GPU device can be hundreds of times greater than on a CPU. However, this acceleration ratio gradually decreases from a small-scale to a large-scale simulation domain. This phenomenon is also related to the SP number, in that the GPU having a higher SP number (NVIDIA A100) shows a slower decreasing trend of the acceleration ratio than one with a smaller SP number (NVIDIA Tesla V100) does. Attaining high computing acceleration of a full kinetic-PIC simulation on a GPU device is eminently feasible using CUDA Fortran Programming.

7. Conclusions and Discussions

A GPU device can efficiently accelerate the computing of a full kinetic PIC simulation. Using CUDA Fortran programming, it is convenient to transplant this process from the traditional Fortran code. Assisted with the next-generation HPC GPU, the time cost of the iteration on a single GPU matches the time taken by over 280 CPUs on clusters; we note that the multinode clusters also have additional communication costs. The further generation of HPC GPUs with more SP and memory, such as the NVIDIA H100, newly released in the year 2022, is preferable to apply to numerical simulations.

The current code version has the preliminary functionality to perform the PIC simulation on the GPU device under the magnetic reconnection configuration. Other capabilities of this code are expected to be fully explored and implemented. Besides the magnetic reconnection, another physical process that can be fulfilling to investigate, turbulence, is undergoing transplantation from the CPU code. As for the algorithm of the iteration part, the fourth-order leap-frog method is under consideration. Due to the high efficiency of GPU computing, more complicated algorithms can be acceptable owing to the relatively low time cost and the reduced iteration error present.

Meanwhile, our codes should also be optimized and upgraded in the future to reveal the advantages of the GPU device completely. The asynchronous data transmission can be applied to optimize the bandwidth utilization of the VRAM. In addition, the on-chip memory allocation is vital to speed up the data access, such as with shared memory instead of global memory. Besides, the situation using multi-GPUs should be developed as well to raise the simulation scale further. Under this condition, the communication between the GPUs should be considered carefully from both hardware and code perspectives. The product NVLink connects the GPUs' VRAM directly; therefore, the data on the current device can be accessed by other devices directly but not through the host transmission, which reduces the communication cost.

This work was supported by the National Natural Science Foundation of China (42074196, 41874191, 41925018) and the National Youth Talent Support Program. S.Y.H. acknowledges the project supported by Special Fund of Hubei Luojia Laboratory. The simulation in this work was carried out on the supercomputing system in the Supercomputing Center of Wuhan University.

Data and Software Availability: The comparison of the CPU and GPU simulation data is available at the Open Science Framework through doi:10.17605/OSF.IO/U8SN4. Both the CPU Fortran and GPU CUDA Fortran source codes can be accessed through the OSF system (doi:10.17605/OSF.IO/U8SN4).

ORCID iDs

Q. Y. Xiong https://orcid.org/0000-0003-1840-3281
S. Y. Huang https://orcid.org/0000-0002-3595-2525
K. Jiang https://orcid.org/0000-0001-7889-0507
Y. Y. Wei https://orcid.org/0000-0003-1199-5229
J. Zhang https://orcid.org/0000-0001-5111-2609
R. T. Lin https://orcid.org/0000-0003-4012-9418

References

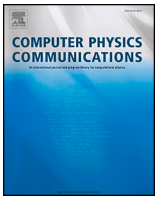
Abreu, P., Fonseca, R., Pereira, J., & Silva, L. 2011, ITPS, 39, 675
Birdsall, C., & Langdon, A. B. 1991, Plasma Physics via Computer Simulation (Bristol: Adam Hilger)
Burau, H., Widera, R., Honig, W., et al. 2010, ITPS, 38, 2831
Decyk, V. 2007, CoPhC, 177, 95
Decyk, V., & Singh, T. 2011, CoPhC, 182, 641
Decyk, V., & Singh, T. 2014, CoPhC, 185, 708
Drake, J., Swisdak, M., Che, H., & Shay, M. 2006, Natur, 443, 553
Fatica, M., & Ruetsch, G. 2014, CUDA Fortran for Scientists and Engineers: Best Practices for Efficient CUDA Fortran Programming (Amsterdam: Elsevier Science & Technology)
Fu, X., Lu, Q., & Wang, S. 2006, PhPl, 13, 012309
Goldman, M., Lapenta, G., Newman, D., Markidis, S., & Che, H. 2011, PhRvL, 107, 135001
Huang, S., Zhou, M., Yuan, Z., et al. 2014, JGRA, 119, 7402
Huang, S., Zhou, M., Yuan, Z., et al. 2015, JGRA, 120, 6188
Lu, Q., Ke, Y., Wang, X., et al. 2019, JGRA, 124, 4157
Markidis, S., Lapenta, G., & Rizwan-uddin 2010, Math. Comput. Simul., 80, 1509
Matsumoto, H., & Omura, Y. 1993, Computer Space Plasma Physics: Simulation Techniques and Software (Tokyo: Terra Scientific Pub. Co)
Winjum, B., Berger, R., Chapman, T., Banks, J., & Brunner, S. 2013, PhRvL, 111, 105002
Xiong, Q., Huang, S., Yuan, Z. G., et al. 2022c, JGRA, 127, e2022JA030760
Xiong, Q., Huang, S., Zhou, M., et al. 2022a, JGRA, 127, e2022JA030264
Xiong, Q., Huang, S., Zhou, M., et al. 2022b, GeoRL, 49, e2022GL098445
Zeiler, A. 2002, JGRA, 107, 1230
Zhou, M., Deng, X., & Huang, S. 2012, PhPl, 19, 042902

Computer Physics Communications 295 (2024) 108994

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

GPIC: A set of high-efficiency CUDA Fortran code using gpu for particle-in-cell simulation in space physics

Qiyang Xiong [a], Shiyong Huang [a,*], Zhigang Yuan [a], Bharatkumar Sharma [b], Lvlin Kuang [c], Kui Jiang [a], Lin Yu [a]

[a] School of Electronic Information, Hubei Luojia Laboratory, Wuhan University, Wuhan, China
[b] NVIDIA Graphics Pvt Ltd, Bangalore, India
[c] NVIDIA, Beijing, China

ARTICLE INFO

Edited by: David W.Walker

Keywords:
Particle-in-cell simulation
High-performance computing
GPU computing
CUDA Fortran programming

ABSTRACT

Previous implementation of the full kinetic particle-in-cell (PIC) simulation on the Graphical Processing Unit (GPU) device has shown advantages compared to traditional Central Processing Unit (CPU) computing. However, conventional PIC simulations using GPU computing have faced limitations in terms of low performance when the simulation box size or the number of particles per cell increases. In this study, we boost computing efficiency by designing novel schemes of kernels with the combination of numerical and technical aspects. The time expenses of the global data transferring and duplicate data fetching processes have been massively reduced by the utilization of the on-chip memory. The reduction treatment on the particles, the graded current computing scheme, and the 2.5D thread launch strategy are designed to accelerate the simulation iterations. The new scheme can reach up to about 5.5 times the acceleration ratio than the old one and attain the highest to 734 times faster than the program using CPUs. Our new scheme can realize large-scale PIC simulations with both decent performances on the legacy GPUs and the increasing trend of the acceleration ratio on the lasted GPUs.

Introduction

Numerical simulation is an acknowledged approach to exploring the evolution and mechanism of space physical processes, for instance, magnetic reconnection and turbulence. One of the extensively employed simulation methods is the full kinetic Particle-in-Cell (PIC), which can resolve the kinetic behaviors down to the electron scale [1–6]. Parallelized computing pattern is indispensably applied in large-scale PIC simulations to increase efficiency through multi-cores or multi-threads of the computing device. The Graphical Processing Unit (GPU), which has the ability to execute massive instructions concurrently, is a kind of emerging parallel computing device during the last decades [7–12]. A single GPU can launch billions of threads to cover the computing tasks concurrently, different from the traditional Central Processing Unit (CPU) pattern, therefore, it can realize maximum parallelism during PIC simulations, especially in the particle moving and current computing processes. The High-Performance Computing (HPC) GPUs designed for the data center guarantee the realization of billions of calculation operations within milliseconds. Particularly, the development platform Compute Unified Device Architecture (CUDA) released by NVIDIA assists scientists in solving complicated computing problems in their research fields. Recently, the highly integrated Software Development Kit by NVIDIA (NVIDIA HPC SDK) provides convenient access to comprehensive programming models, the compilers of different languages and reliant libraries. These foundations and advantages of GPUs precipitate widespread applications and remarkable performance in Machine Learning (ML)/Deep Learning (DL) as well as metadata computing.

Our previous implementation has demonstrated the incipient scheme of full kinetic 2.5D (two-dimensional space and three-component fields and velocities) PIC simulation transplanted from Message Passing Interface (MPI)-based mode to the GPU-based mode using CUDA Fortran programming [13]. Two types of thread-mapping strategies for field matrices and particle arrays have been given to cover each thread's SIMD (Single Instruction Multiple Data) assignments in the GPU device. This former study has also provided methods of field solving and current computing adapted to the GPU's specific architecture. The benchmark results show the outstanding accelerated computing by the data center

[*] Corresponding author.
E-mail address: shiyonghuang@whu.edu.cn (S. Huang).

https://doi.org/10.1016/j.cpc.2023.108994
Received 21 September 2023; Accepted 23 October 2023
Available online 30 October 2023

GPUs compared with the time cost of the CPU (Central Processing Unit). However, previous schemes and code designs have not completely excavated potential capabilities that benefited from the framework of the GPU. The constant utilization of the global memory instead of the on-chip memory will cost more time on the data fetching and transferring during the calculation process. Meanwhile, it is arranged in the previous scheme that one thread is assigned to one particle to attain the update of the particles' status and the contribution to the current. Simply employing this sequential SIMD ideology on computing unsorted particles will spend duplicate time accessing the field data for those particles in the identical cell. These factors cause the lower acceleration rate as the simulation domain size increases (see Fig. 5b in Ref. [13]).

In this work, we optimize the schemes of the kernels executed on the GPU device to overcome the high expense of data fetching and improve the performance in large-scale simulations. Three main aspects are focused on improving the performance of the simulation code: field solving, particle moving, and current computing. The high-bandwidth on-chip memory (e.g., shared memory and local register) is applied as much as possible to avoid repetitive global data fetching operations. The particles are classified according to their grid positions before being advanced by the local fields; whereafter, the reduced SIMD methodology is applied to update the particles' positions within each cell. Meanwhile, the three-stage current computing pattern is developed to reduce the substantial Read-Write Conflicts (RWC) on the global memory by atomic operations. Correspondingly, the new thread-launch strategy of the GPU device cooperates with the updated algorithms above to attain more thread occupancy. The benchmark results manifest that the new version scheme can reach a much higher computing speed than the old one in Ref. [13]. It also significantly improves the performance when the data volume of the simulation increases. The present code is ready to implement device-level and host-level parallelism in future development.

Development environment and background information

Two kinds of NVIDIA data center GPUs are employed in this study to testify to the scheme efficiency: NVIDIA Tesla V100 and NVIDIA A100, which were initially released in 2017 and 2020, respectively. Some specifications of these devices are listed in Table 1. After considering the requirements of multiple simulation scales varying from small to large in the benchmarks, we choose two different specifications of each type of GPU: NVIDIA Tesla V100–16G (model: V100-SMX2–16G), NVIDIA Tesla V100–32G (model: V100-SMX2–32G), NVIDIA A100–40G (A100-SMX4–40G), and NVIDIA A100–80G (A100-SXM4–80G). The difference between the same type of GPU primarily lies in the Video Random Access Memory (VRAM) size, and there also exists slight performance differences.

In the former scheme, each step of the PIC iteration on the GPU device consists of six procedures, i.e., half advance of the magnetic field, particle moving, half advance of the magnetic field, current computing, particle position calibration, and full advance of electric field (seen in the right part of Fig. 3 in Ref. [13]). The time cost proportions of these parts in each step are shown in Fig. 1. Apparently, on all devices, the more time-consuming processes belong to the particle moving (blue pie) and current computing (yellow pie), similar to the results in other studies [14,15]. These two computing operations occupy almost 99% of the one iteration time. These results are not surprising on account that the particle treatment requires more extensive computation than the fields and depends on the number of particles per cell (*ppc*). Meanwhile, the time cost percentage of the current computing on the A100 device (Fig. 1c-1d) is less than that on V100 (Fig. 1a-1b), which is the benefit from the higher computability and memory bandwidth of the newer generation device for A100. Apart from the updates of the device products, it is still urged to upgrade the schemes and algorithms of these two parts to adapt to the GPU architecture, thus further ameliorating the computing efficiency.

Table 1
Specifications of Modern Data Center GPUs.

Specification	NVIDIA Tesla V100	NVIDIA A100
GPU Codename	GV100	GA100
GPU Architecture	NVIDIA Volta	NVIDIA Ampere
GPU Boost Clock	1530 MHz	1410 MHz
CUDA Cores	5120	6912
Memory Size	16 GB/ 32 GB	40 GB/ 80 GB
Memory Data Rate	877.5 MHz	1215 MHz
Memory Bandwidth	900 GB/sec	1555 GB/sec

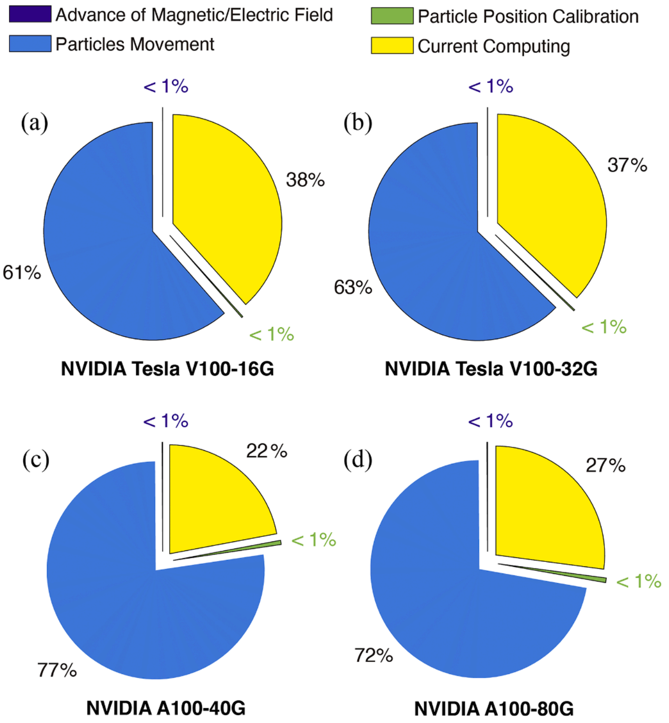


Fig. 1. Time expense percentage of different procedures in each iteration. These benchmark results are obtained from the simulations using domain size 600×1200 and 100 particles per cell. The time expense of each procedure is the average result of the 10,000 iterations. The percentage is calculated through the method: $t_{pi}/\sum_{i} t_{pi}$, where $i \in [1,\ 4]$ and t_{pi} refers to the time expense of the procedure *i*.

A more detailed architectural configuration of a GPU device than the previous sketch is illustrated in Fig. 2. The connections among the threads, blocks, Streaming Multiprocessor (SM) and various types of memory are vividly presented [16–18]. Each block contains certain threads (maximum 1024 threads for both V100 and A100) and a certain size of shared memory (configurable up to 96 KB/164 KB for V100/A100, respectively). The threads within the same block can access the shared memory of the corresponding block simultaneously. They also have their private local memory and registers to store the data. Besides, all threads in different blocks have access to the global, constant, and texture memory in the VRAM, where the constant and texture memories are read-only during the kernel execution. The SMs in a GPU (80 and 108 SMs in V100 and A100, respectively) cover the computing tasks of multiple threads. In the previous scheme [13], all threads are disposed to fetch the data from the global memory directly. It will cause duplicate data access operations by the threads in the present block, where these threads call for the same data. The central idea of the optimization in the following part is to reduce the frequent data transferring and increase the rate of data reutilization by reorganizing the algorithms of field solving, particle moving, and current computing.

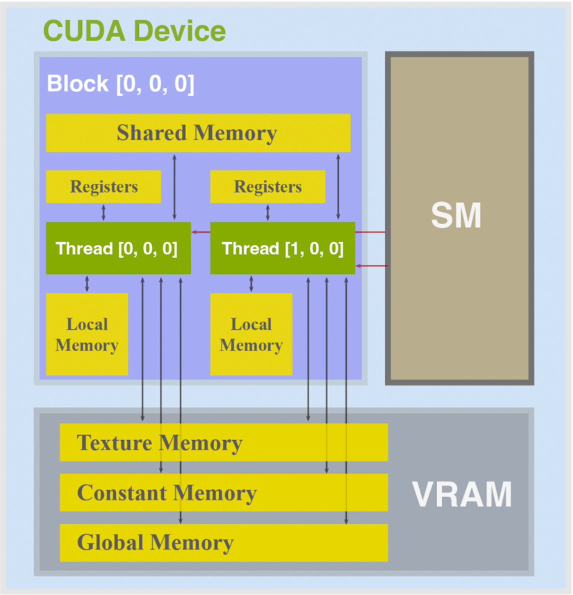


Fig. 2. The detailed architecture of a GPU device. The black arrows represent the data transferring between the threads and various types of memory. The red arrows represent the execution of the instructions on threads by the Streaming Multiprocessor (SM).

Optimization of the field solving

The field solving part contains two processes: advancing magnetic field (following Faraday's law: $\partial \boldsymbol{B}/\partial t = -c\,\nabla \times \boldsymbol{E}$, where $\boldsymbol{B}$ is the magnetic field, $\boldsymbol{E}$ is the electric field, and c is the light speed) and electric field (following Ampere's law: $\partial \boldsymbol{E}/\partial t = c\,\nabla \times \boldsymbol{B} - 4\pi \boldsymbol{J}$, where $\boldsymbol{J}$ is current). We pick the magnetic field to describe the scheme in detail. Under 2.5D spatial configuration (in the x-z plane), the three components of Faraday's law can be written in the partial derivative form:

$$\partial B_x/\partial t = c\ \partial E_y/\partial z$$

$$\partial B_y/\partial t = -c\ (\partial E_x / \partial z - \partial E_z / \partial x) \tag{1}$$

$$\partial B_z/\partial t = -c\ \partial E_y/\partial x$$

In the numerical simulation, these equations are approximated by their discrete forms (the first-order forward space difference is applied to this scheme, and we take the first half advance of the magnetic field, for example):

$$B_x^n(i,j) = B_x^{n-1/2}(i,j) + c(\Delta t / 2\Delta x)\left(E_y^n(i,j+1) - E_y^n(i,j)\right)$$

$$\begin{aligned} B_y^n(i,j) &= B_y^{n-1/2}(i,j) \\ &\quad - c(\Delta t / 2\Delta x)\left(E_x^n(i,j+1) - E_x^n(i,j) - E_z^n(i+1,j) + E_z^n(i,j)\right) \end{aligned} \tag{2}$$

$$B_z^n(i,j) = B_z^{n-1/2}(i,j) - c(\Delta t / 2\Delta x)\left(E_y^n(i+1,j) - E_y^n(i,j)\right)$$

where n is the present iteration step, i and j are the grid index in X and Z direction, Δt is the discrete time interval of each advance, and Δx is the length of each cell. Eq. (2) represents a typical numerical solution to the time-dependent Maxwell equations: Finite-Difference Time-Domain (FDTD) method [19]. It should be noticed that the value on the point (i, j) at the left-hand-side of Eq. (2) requires at least the values on two points (e.g., (i, j) and ($i + 1$, j), or (i, j) and (i, $j + 1$)) at right-hand-side to finish computing operation. That means the values on most of the grid points are fetched twice from the global memory address in the previous scheme, and this process consumes much time.

The utilization of the shared memory can effectively reduce much of the transferring between the threads and global memory in the FDTD method [20,21]. The shared memories exist in each block on the GPU device, and they participate in the computing without disturbing each other. Each block can bring a tile of data to the shared memory, and then each thread in this block can access all elements of the shared memory tile as needed [22]. Fig. 3 displays the configuration example of the shared memory tile size applied in the first-order FDTD method. Initially, the prototype size of the tile is determined by the thread number in the block (cuThreadx × cuThready) (green square in Fig. 3). However, the calculation at the point of the end row or column (i = cuThreadx or j = cuThready) in each block requires the data at the next row or col (i = cuThreadx + 1 or j = cuThready + 1). Therefore, the tile size of the shared memory should be extended by one more row and column (deep yellow rectangular in Fig. 3), and the final shared memory contains (cuThreadx + 1) × (cuThready + 1) data. The field data are preloaded from the global memory to the shared memory before proceeding to FDTD computing. This process reduces the time-expensed global data fetching operations on each single grid point data to once. And the data on the shared memory can be reused at least twice by the threads in the corresponding block.

During the programming implementation, the shared memory data is declared first in the device code (subroutines with the prefix attributes (global)):

```
real, shared, dimension (blockDim%x+1, blockDim%y+1) :: ex_s, ey_s, ez_s
```

Then, the thread mapping to the field matrix index should be built to access the data correctly. Except for the global array index which has been stated in the previous study:

```
i = (blockIdx%x - 1) × blockDim%x + threadIdx%x
j = (blockIdx%y - 1) × blockDim%y + threadIdx%y
```

the thread index within the block which is called the in-block array index should also be given for the convenience of accessing shared data:

```
i_s = threadIdx%x
j_s = threadIdx%y
```

After that, the global memory data is preloaded to the shared memory data before the computing operation:

```
if(i <= mx .and. j <= mz)then
```

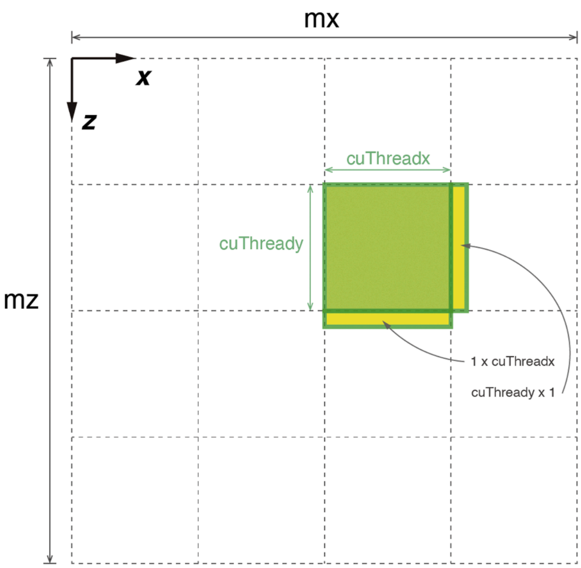


Fig. 3. Sketch of the shared memory size configuration. The green square represents the block size, and the deep yellow rectangular represents the additional stride data. These two parts constitute the tile size of the shared memory in each block. The parameters mx and mz are the simulation domain size in X and Z directions, and cuThreadx and cuThready are the threads number launched in X and Y dimensions in each block.

```
  ex_s(i_s,j_s) = ex(i,j); ey_s(i_s,j_s) = ey(i,j)
  ez_s(i_s,j_s) = ez(i,j)
  if(i_s == blockDim%x) then
  ez_s(i_s+1,j_s) = ez(i+1,j); ey_s(i_s+1,j_s) = ey
(i+1,j)
  end if
  if(j_s == blockDim%y) then
  ex_s(i_s,j_s+1) = ex(i,j+1); ey_s(i_s,j_s+1) = ey
(i,j+1)
  end if
  end if
```

During the data transferring process above, it is difficult to be aware of whether this operation is completed. Therefore, the barrier for the threads should be set to make all threads in a block wait until they are visible to the shared memory simultaneously:

```
  call syncthreads()
```

Ultimately, the magnetic field is advanced by the electric field data in the shared memory instead of the global memory:

```
  if(i >= 2 .and. i <= mx-1 .and. j >= 2 .and. j <= my-1)
then
  bx(i,j) = bx(i,j) + 0.5*c*(ey_s(i_s,j_s+1) - ey_s
(i_s,j_s))
  by(i,j) = by(i,j) - 0.5*c*(ex_s(i_s,j_s+1) - ex_s
(i_s,j_s) - ez_s(i_s+1,j_s) + ez_s(i_s,j_s))
  bz(i,j) = bz(i,j) - 0.5*c*(ey_s(i_s+1,j_s) - ey_s
(i_s,j_s))
  end if
```

The comparison of the benchmark results between the previous scheme and the present one is shown in Fig. 4. A series of simulation domain sizes are tested to compare the efficiency horizontally. From the acceleration ratios (blue lines in Fig. 4) of different GPU devices, it is informed that the V100 and A100 GPUs have an average value of 8% and 5% performance improvement, respectively. The low data reusability rate of the shared memory causes the limited benefit of the upgrade. In our scheme, the first-order forward difference only employs the shared memory data twice (the present and the adjacent points) on most grid points. If the higher order of the FDTD algorithm is applied, it could be expected with more significant improvement from the new scheme [23–25]. On the other hand, the preponderance of the new scheme is narrowed by the device hardware upgrading after comparing the results of A100 (Fig. 4c-4d) and V100 (Fig. 4a-4b). This phenomenon is eventuated by the promoted global memory rate and bandwidth of the data center GPU (Table 1). However, the new scheme can achieve preferable performance on the common GeForce game GPUs for their limited VRAM bandwidth. Nevertheless, it is still worthy of being reserved for future high-order FDTD and 3D model development.

Optimization of the particle moving

The particle moving predominantly complies with the Newton-

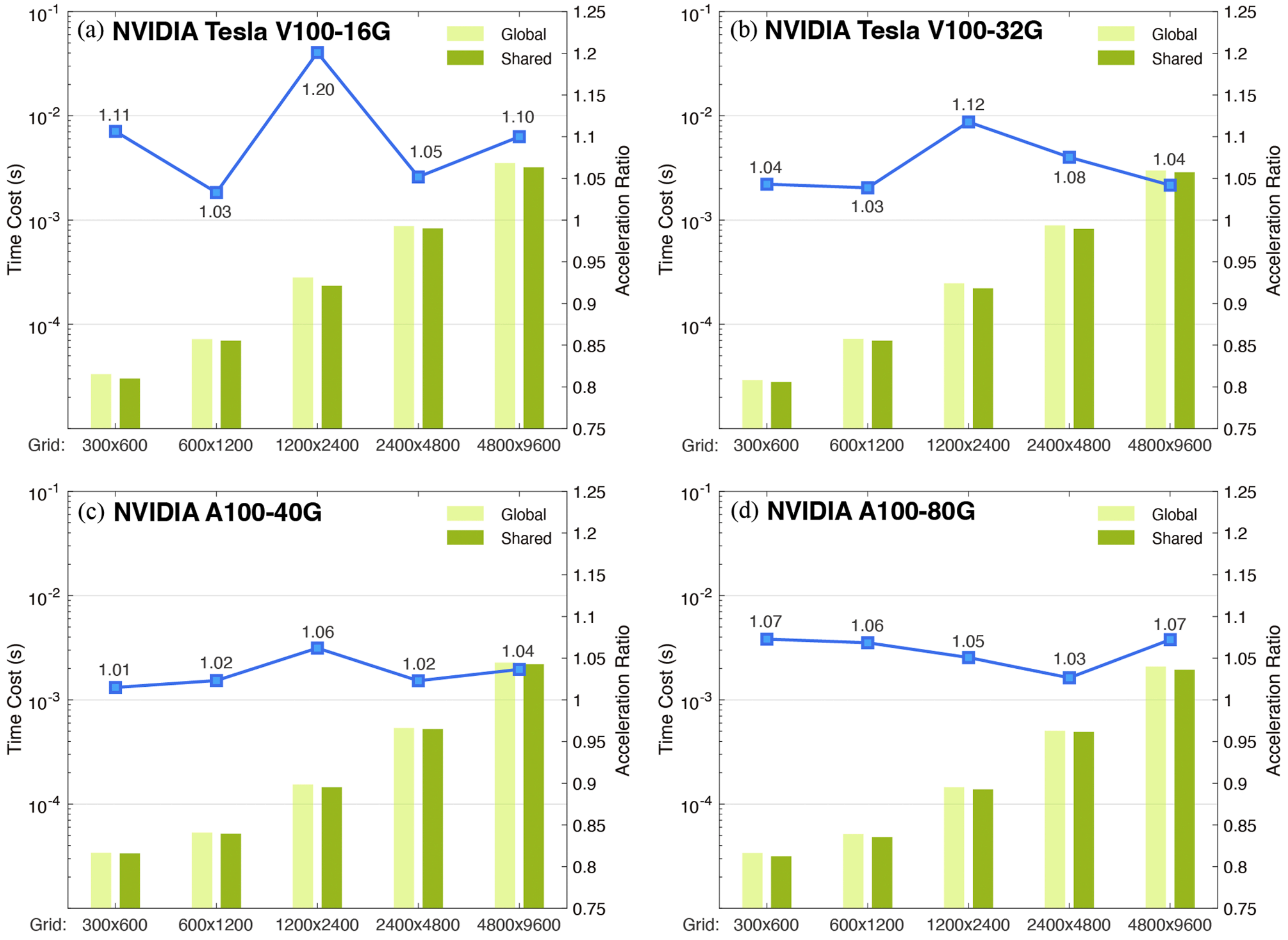


Fig. 4. Benchmark results of different data center GPUs with a series of simulation domain sizes in the FDTD part. The histograms are the time costs belonging to the left axis. The shallow green bars are the results of the previous scheme using the global memory, and the deep green bars are the results of the present scheme where the shared memory is applied. The blue lines are the acceleration ratio between the old and new scheme, and they are obtained by calculating the value of t_{global}/t_{shared}. The simulation domain sizes are given at the X label of the subfigures, and *ppc* is settled at 2 to preserve the memory for data storage of the large-scale run cases. All the data of the time costs are the average results of 10,000 iterations.

Lorentz equation. Specifically speaking for a single particle:

$$d\boldsymbol{x}_s/dt = \boldsymbol{v}_s \qquad d\boldsymbol{v}_s/dt = q_s(\boldsymbol{E}_0 + \boldsymbol{v}_s \times \boldsymbol{B}_0)/m_s \tag{3}$$

where $\boldsymbol{x}_s$ is the particle's position, $\boldsymbol{v}_s$ is the velocity, q_s is the charge, and m_s is the mass. The subscript s stands for the particle species (ion or electron). The parameters $\boldsymbol{E}_0$ and $\boldsymbol{B}_0$ are the local electric and magnetic fields at the particle's position. In the discrete format, the motion change of a single particle can be replaced by:

$$\frac{\boldsymbol{v}_{n+1/2} - \boldsymbol{v}_{n-1/2}}{\Delta t} = \frac{q}{m}\left[\boldsymbol{E}_{n0} + \frac{\boldsymbol{v}_{n-1/2} + \boldsymbol{v}_{n-1/2}}{2} \times \boldsymbol{B}_{n0}\right] \tag{4}$$

where all the vector parameters are added with the subscript n to represent the present iteration step. This treatment is desired to attain a centered-difference form of Eq. (3); therefore, the magnetic term is averaged by $\boldsymbol{v}_{n-1/2}$ and $\boldsymbol{v}_{n+1/2}$ [26]. The local fields data $\boldsymbol{E}_{n0}$ and $\boldsymbol{B}_{n0}$ are acquired through the interpolation method (linear interpolation is used in this scheme) using the grid point field data. The numerical method of solving Eq. (4) has been provided in previous studies [27–28]. Finally, the particle velocity update process is decomposed into four procedures: twice half-acceleration by the electric field and twice half-rotation by the magnetic field. And the particle position follows the linear increase from the newly-obtained velocity: $\boldsymbol{x}_{n+1/2} = \boldsymbol{x}_{n-1/2} + \Delta t\ \boldsymbol{v}_{n+1/2}$.

As a whole, the treatment of the particles is regarded as a SIMD pattern. That means all the particles obey the rules of Eq. (3) and (4) to update their status, and the computing implementation of each particle is independent. Those particles located in the same cell will access the same magnetic/electric field data on the grid points to obtain the interpolated local fields. Unlike other models using the CIC scheme (Cloud-in-Cell, Ref. [29]) or applying collisions [30], the only difference between those particles should be considered in the collisionless kinetic PIC simulation is the distance to the present cell's boundary. If the particles are disposed sequentially following the array storage order just as Fig. 5a, it will consume much unnecessary time on the duplicate data accessing the same fields. Meanwhile, the increase of the parameter *ppc* will further exacerbate this situation. One possible resolvent to this issue is categorizing the particles and applying reduction treatment (Fig. 5b), which is commonly adopted in PIC with Monte-Carlo collision [31–33]. The number of the particles and their array indices in each cell can be counted through the particles' present positions. Then the 2D thread-mapping method similar to that in the field solving part is adopted to handle the particle moving in each cell, instead of the 1D thread launch strategy proposed in the former study. This kind of shift from the sequential SIMD (particle-loop) to the reduction SIMD (cell--loop) potentially prevents the excessive threads charged by each SM.

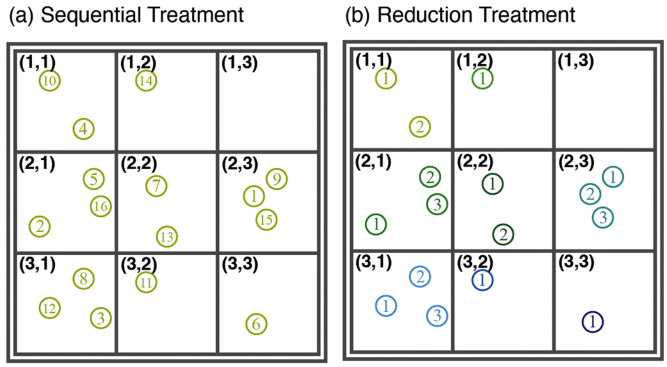


Fig. 5. Sketch of the comparison between the sequential scheme (a) and reduction scheme (b) in the particle moving process. The circles with a number represent the particles. The number in (a) is the index sequence of the particles in the global data array, and the number in (b) is the particle indices within each cell. The different colors of particles in (b) distinguish their cell belongings.

Assisted with the new scheme of the particle moving, it can conserve much of the time expense on the field data fetching. However, given the fact that the parameter *ppc* can vary to an enormous value, it would be inadequate for a single thread to cover the computing task of all the particles in a cell if it is simply applied with the 2D thread-mapping method. Moreover, the cell-loop scheme is not coalescing automatically, and the memory bandwidth is less effective during the data fetching [34]. Therefore, we raise a new concept of kernel launch model for the new scheme: 2.5D thread-block strategy, which is illustrated by the diagram in Fig. 6. Compared with the previous 2D thread-mapping method (Fig. 2a in Ref. [13]), the new strategy describes the pattern that the three-dimensional threads are employed in each block and two-dimensional blocks are used in the GPU device. More specifically, it is organized that the threads [0, 0, 0], [0, 0, 1], [0, 0, 2], and [0, 0, 3] (gray cubes in Fig. 6) in the block [0, 0, 0] (orange cubic outlines) collectively deal with the computing assignments of the cell [0, 0]. This kind of arrangement constitutes two levels of parallelism: the thread-level instructions concurrently carry out the tasks of particle moving within each cell, and block-level executions simultaneously cover all the cells in the simulation domain. This new strategy compensates for the deficiency of the launched threads number when the reduction SIMD method is adopted. One should be noted that this arrangement of the threads for particles can lead to asynchronous execution of threads in the same wrap, since the particle number across cells varies during PIC evolution. The threads in a warp executing different instructions will decline the performance of the device to a certain degree. This situation will be relieved when *ppc* increases, as the base particle number is much larger than the fluctuation number across different cells.

The implementation of the new scheme has a similar flow path to the field solving part. In the beginning, the fields data (bx, by, bz, ex, ey, ez) are preloaded to the shared memory (bx_s, by_s, bz_s, ex_s, ey_s, ez_s) in advance for the advantage of its low time expense even if there are repetitive and massive data accessing requests [35]. Accompanied by the data loading process, the field is also gathered from the mesh onto the macroparticles:

```
ex_s(i_s,j_s) = (ex(i-1,j) + ex(i,j))/2.0
ez_s(i_s,j_s) = (ez(i,j-1) + ez(i,j))/2.0
ey_s(i_s,j_s) = ey(i,j)
bx_s(i_s,j_s) = (bx(i,j-1) + bx(i,j))/2.0
bz_s(i_s,j_s) = (bz(i-1,j) + bz(i,j))/2.0
by_s(i_s,j_s) = (by(i-1,j-1) + by(i,j-1) + by(i-1,j)
+ by(i,j))/4.0
```

Then, apart from the X and Y dimension in-block thread indices mentioned in the previous section, the index in the Z dimension should also be defined:

```
k_s = threadIdx%z
```

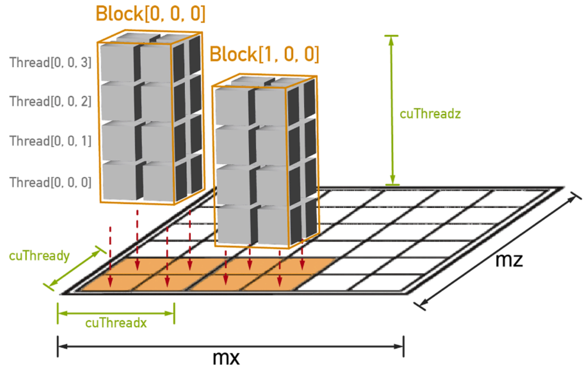


Fig. 6. Diagram of the 2.5D thread-block launch strategy. The gray cubes represent the threads. The orange outlines around the cubes and the orange squares on the 2D plane stand for the blocks. The parameter cuThreadz is the threads number launched in the Z dimension in each block.

After that, the particle moving process in each cell can proceed:

```
if(i >= 2 .and. i <= mx-1 .and. j >= 2 .and. j <= mz-1) then
  ! Number of particles in the cell (i,j) is stored in Ns [2]
  do sp = 1, 2
  qm = (sp - 1)*qme – (sp - 2)*qmi
  do n = k_s, Ns(sp), blockDim%z
  ! Interpolate the fields and calculate B0 and E0 using shared data b_s and e_s
  ! Move Particle n following the equation in discrete form
  end do
  end do
  end if
```

In the code above, the symbol sp represents the particle species (1 - ion and 2 - electron). The charge-mass ratio qm is determined by computing approach rather than the logical selection using the build-in module if; end if, because of the lower efficiency of the logic operation than the numerical calculation on the GPU device. The second do; end do module reflects the thread-level parallelism design achieved by the multi-threads cooperation on each cell. For instance, if one cell is responsible for 4 threads (blockDim%z equals 4, just as Fig. 6) and there are 64 particles in this cell (Ns equals 64), the thread [0, 0, 0] (k_s equals 1) covers computing tasks of the 1st, 5th, 9th …, 61st particles, and thread [0, 0, 1] (k_s equals 2) manages the 2nd, 6th, 10th, …, 62nd particles, and so on. All threads execute the computing instructions simultaneously to realize maximum parallelism and GPU device utilization.

The performance results of the new scheme are provided in Fig. 7. For the former generation data center GPUs (V100 series), all the benchmarks with different numbers of *ppc* show the outstanding improvement benefit from the reduced scheme (Fig. 7a and 7b). It can reach up to about 15 times of acceleration rate than the old scheme, which confirms the value of the development of the new algorithm. On the other hand, however, the profit from the updated scheme on the A100 GPUs is not so impressive. It only has about 1.4∼3.53 times acceleration ratio compared to the time expense of these two schemes (Fig. 7c and 7d). This outcome is not surprising as the hardware upgrade will narrow the ascendancy of the superior algorithms, which has been mentioned in the last section. Additionally, the new scheme has a significant acceleration ratio increase for all the GPU models when the parameter *ppc* rises. But this increasing trend is slowed down as *ppc* reaches 100 or even more. Nonetheless, the acceleration ratio almost has no sign of decline during the continuous increase of *ppc*. The 2.5D thread-block launch strategy can take credit for this excellent and satisfying feature.

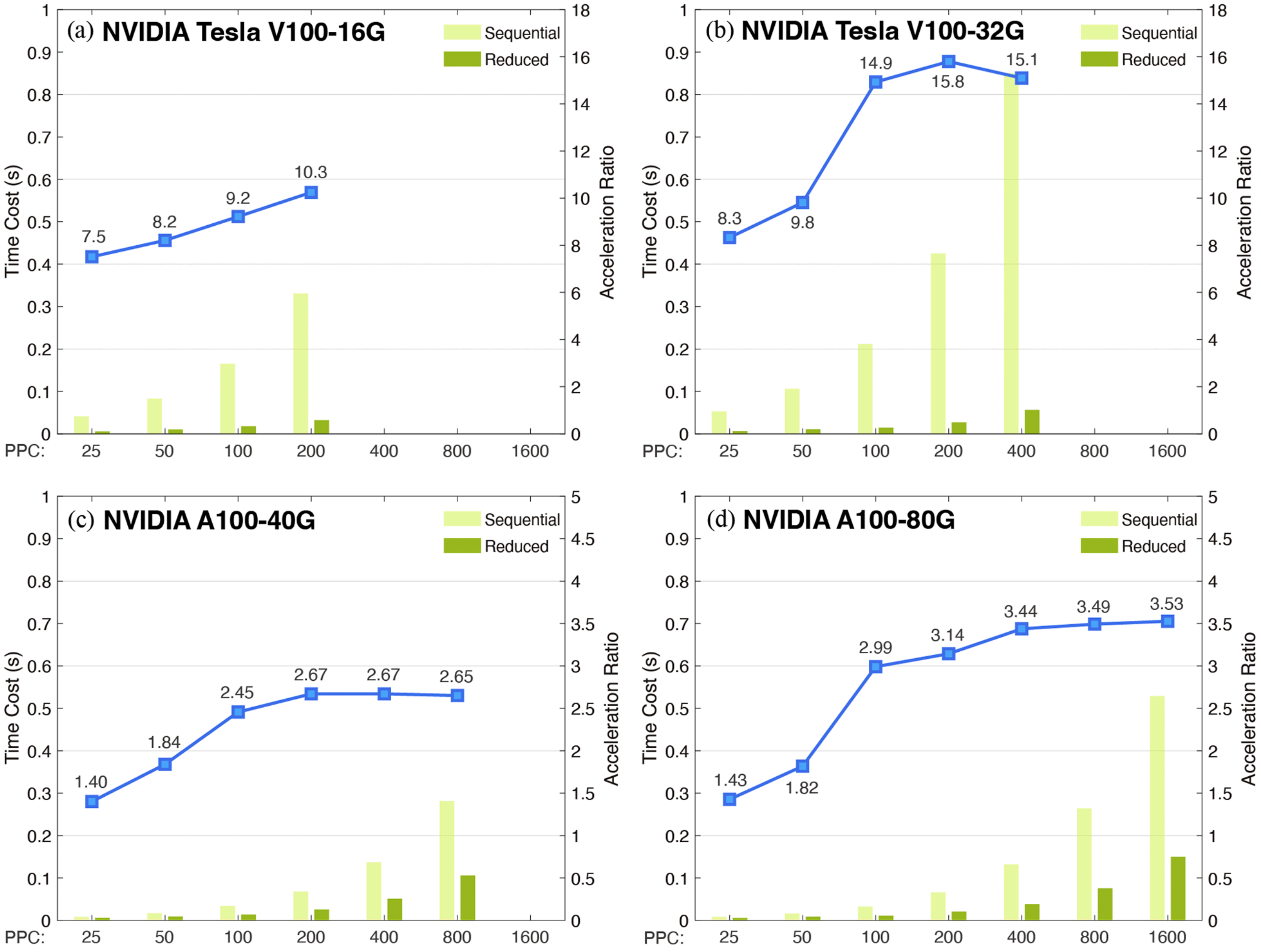


Fig. 7. Benchmark results of different data center GPUs with a series of *ppc* in the particle moving part. The shallow green bars are the results of the previous scheme where the sequential treatment is used, and the deep green bars are the results of the present scheme where the reduction method is applied. The acceleration ratio between the old and new scheme is obtained by calculating the value of $t_{sequential}/t_{reduction}$. The simulation domain size in this part is fixed at 600×1200, and variations of *ppc* are given at the *X* label of the subfigures. All the data of the time costs are the average results of 10,000 iterations.

Optimization of the current computing

During the evolution of the space environment and plasma condition, particle moving contributes to the formation of the conduction current:

$$\boldsymbol{J} = q_i n_i \boldsymbol{V}_i + q_e n_e \boldsymbol{V}_e \tag{5}$$

where n_i and n_e are the number density of ions and electrons, and $\boldsymbol{V}_i$ and $\boldsymbol{V}_e$ are the bulk flow velocity of these two species. The number density and the bulk flow velocity are the zero-order and first-order moment of the particle distribution function, respectively. Considering the individual particles in the discrete situation, the current is composed of their position difference within the time interval Δt [36,37]:

$$\boldsymbol{J} = \sum_{np} q_i \boldsymbol{v}_i + q_e \boldsymbol{v}_e = \frac{1}{\Delta t} \sum_{np} q_i \left(\boldsymbol{x}_{i,1} - \boldsymbol{x}_{i,0}\right) + q_e \left(\boldsymbol{x}_{e,1} - \boldsymbol{x}_{e,0}\right) \tag{6}$$

where *np* is the particle's total number, and the subscripts 0 and 1 stand for the previous and present particle statuses. In the meshed PIC simulation domain, the particles can cross a cell's boundary and enter other cells during the moving process. Thus, the conduction current produced by those particles should be split into at least two parts by the cell's boundary line to be weighted on the different cells' grid points [38]. In that case, Eq. (6) can further be expended as:

$$\boldsymbol{J} = \frac{1}{\Delta t} \sum_{np} q_i \left(\sum_{k} \boldsymbol{x}_{i,1,k} - \boldsymbol{x}_{i,0,k} \right) + q_e \left(\sum_{k} \boldsymbol{x}_{e,1,k} - \boldsymbol{x}_{e,0,k} \right) \tag{7}$$

In Eq. (7), the additive count number k represents that the trajectory of the particle np is split into k parts. After the current elements contributed by each particle are obtained, they are allocated to the grid point data multiplied by the corresponding weight:

$$\boldsymbol{J}(i,j) = \sum_{np} \boldsymbol{J}_{s,\ k} W(i,j) \tag{8}$$

where W is the weight factor of each current element to the four grid points of each cell. Thereby, the conduction current produced by the particle moving is deposited discretely to be stored in the data array.

Although the computing tasks of the current element from each particle are regarded as a SIMD process on the GPU device, the accumulation of the current elements on the same grid point cannot be dealt with simultaneously. The RWC problem occurs when multiple additive operations are executed on the identical data address, which has been proposed in the previous study. The prospective approach to this issue is the atomic operation which barriers the requests from other threads until the present calculation is finished. In conditions with large *ppc*, however, the frequent and massive atomic operations on the global memory will significantly decrease the computing efficiency [39,40]. The improvement method of tiles-based or partitions-based can be chosen to achieve the dynamic load of each block [41,42]. Here, we introduce a three-stage current computing scheme adapted to the GPU architecture, which is illustrated in Fig. 8. For the first level, all threads in each block are actively involved in the calculation of the current element and hold the results on the registers (functioning as a calculator). Then on the second level, the shared memory in each block gathers all the current elements calculated within the present block (functioning as an accumulator). And finally, the collected data in the shared memory are written back to the global memory on the third level (functioning as storage). Through this approach, the computing instructions complete most of the atomic operations on the shared memory. Benefiting from the higher bandwidth of the on-chip memory (registers and shared memory), the new scheme reduces much of the time expense from thread barriers.

The new scheme of the current computing reserves the features of reduction treatment on the particles and 2.5D thread-launch strategy, which are similar to the particle moving part. The shared memory data of the current is declared initially (jx_s, jy_s, jz_s). Next, the current computing process in each cell is followed as:

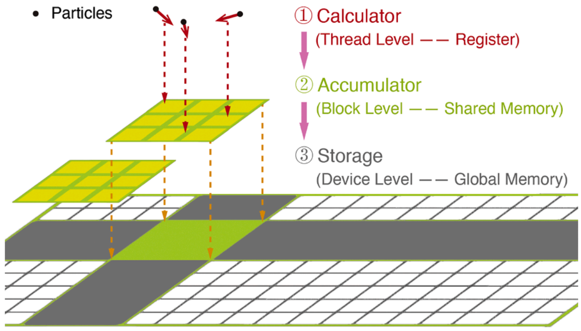

• Particles
① Calculator
(Thread Level —— Register)
② Accumulator
(Block Level —— Shared Memory)
③ Storage
(Device Level —— Global Memory)

Fig. 8. Sketch of the three-stage current computing scheme. The black dots stand for the particles. The solid red arrows represent the conduction current contributed by each particle. The green grids with deep yellow fill represent the shared memory array. The black grid at the bottom represents the global memory array.

```
  if(i >= 2 .and. i <= mx-1 .and. j >= 2 .and. j <= mz-1)
then
  do sp = 1, 2
  q = (sp - 1)*qe - (sp - 2)*qi
  do n = k_s, Ns(sp), blockDim%z
  call   COMPUTINGCURRENT(jx_s,jy_s,jz_s,x(n),y(n),u
(n),v(n),w(n),q)
  end do
  end do
  end if
```

In the code above, the symbol q is the present particle's charge, the symbols x and y are the particle's 2D positions, and the symbols u, v, and w are the 3D velocities. The subroutine (attributes(device) subroutine COMPUTINGCURRENT(...)) executed on the GPU device is responsible for computing each particle's current contribution and accumulating to the shared array. At the time when this process is complete, the temporarily stored data in jx_s, jy_s, and jz_s are collected to the global array jx, jy, and jz:

```
  if(i >= 2 .and. i <= mx-1 .and. j >= 2 .and. j <= mz-1)
then
  jxold = atomicadd(jx(i,j),jx_s(i_s,j_s))
  jyold = atomicadd(jy(i,j),jy_s(i_s,j_s))
  jzold = atomicadd(jz(i,j),jz_s(i_s,j_s))
  end if
```

Through the new scheme proposed above, the high time-expense atomic operation on the global memory is prevented from being applied by each particle. It also speeds up the data transferring rate and increases the efficiency, theoretically. Another essential point should be noted that the thread number in the Z dimension (threads dealing with the same cell) is expected not to be large in the current computing part. Otherwise, the cost of more atomic operations on the shared memory will restrain the advantage of thread-level parallelism in a single cell.

In Fig. 9, the improvement results of the new scheme are provided. The significant acceleration ratios are attained in all simulation benchmarks using the V100 GPUs. It can boost the program by about 3 ~ 7 times faster compared with the time expense of the old scheme (Fig. 9a and 9b). The situations on the A100 GPUs are not optimistic. It has a limited acceleration ratio of up to 2.5 times (Fig. 9d), and it even could have a negative performance when *ppc* is at a low value (Fig. 9c). The acceleration ratio differences reflect the architecture disparities of these two generation GPUs. The hardware upgrade prominently improves the data transferring speed of the global memory, which can result in low promotion compared with the old scheme on the newer generation GPUs. Meanwhile, as the parameter *ppc* increases, the acceleration ratios of V100 GPUs maintain a declining trend. In contrast, on A100 GPUs it

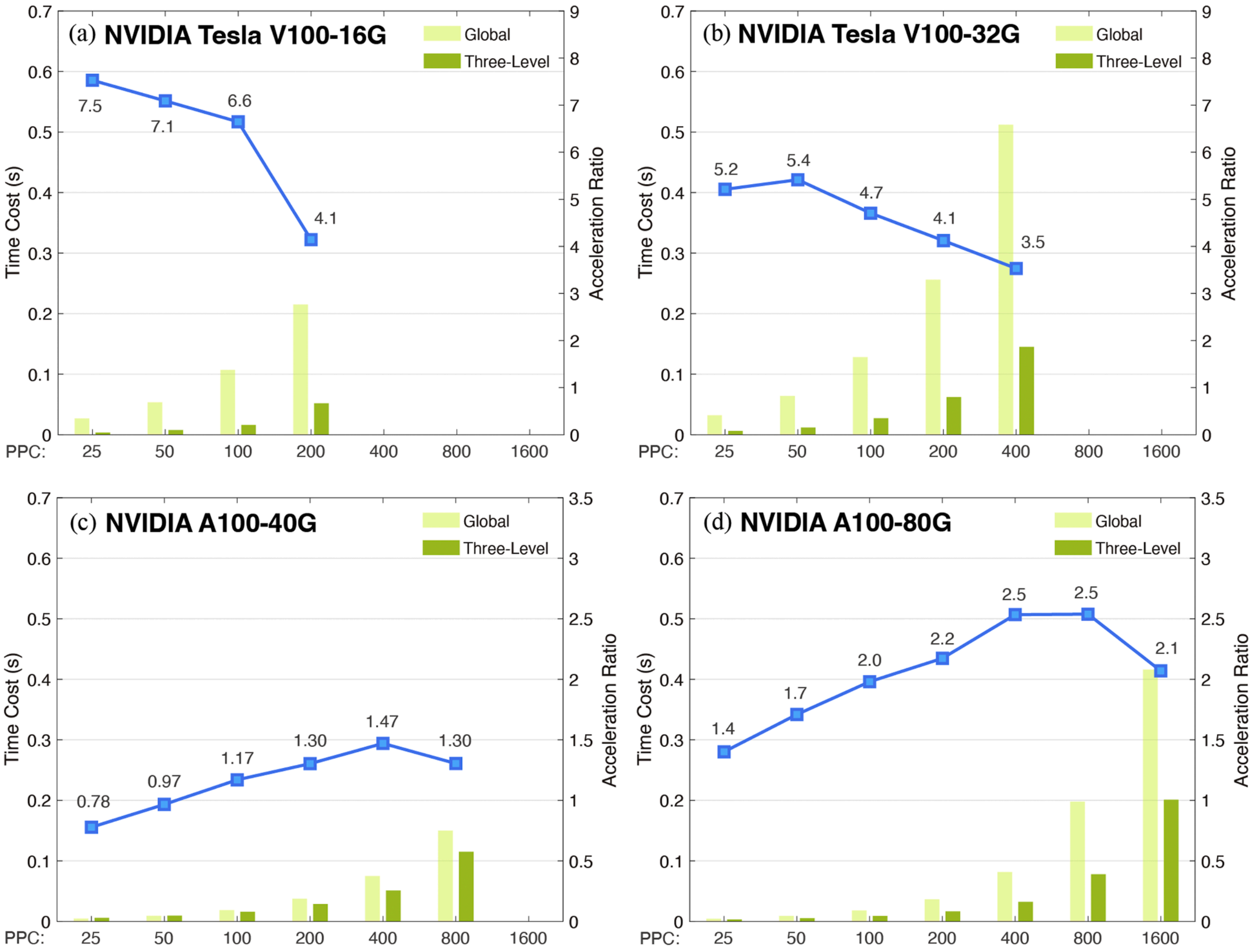


Fig. 9. Benchmark results of different data center GPUs with a series of *ppc* in the current computing part. The shallow green bars are the results of the previous scheme where the atomic operations are directly applied to the global memory. And the deep green bars are the results of the present scheme where the three-stage method is applied. The acceleration ratio between the old and new scheme is obtained by calculating the value of $t_{global}/t_{three-level}$. The simulation domain size in this part is fixed at 600×1200, and variations of *ppc* are given at the *X* label of the subfigures. All the data of the time costs are the average results of 10,000 iterations.

gradually increases at first and then reaches the bottleneck when *ppc* equals 400. This phenomenon is benefited from the advanced architecture which alleviates the RWC when using multiple atomic operations, so that the increase of *ppc* does not significantly lower the acceleration ratio as the former generation GPUs.

Summary and overall performance

In the present stage of development, the updated algorithms and schemes for the 2.5D full kinetic PIC simulation have been implemented theoretically and achieved remarkable performance in practice. Particularly, the particle moving and the current computing parts during the iteration attain a significant improvement over the previous code version. To complete the whole procedure of the consecutive iterations, the sequence of subroutines in the flow chat should be rearranged accordingly. Fig. 10a provides the new scheme based on the reliance of each computing assignment:

1) Firstly, the iteration begins with the half-advance of the magnetic field following Faraday's law. The shared-memory-based scheme for this FDTD is applied;
2) Then, the particles' positions and velocities are updated using the local electromagnetic fields following the Newton-Lorentz equation. The reduction treatment on the particles and the 2.5D thread launch strategy is adopted in this step;
3) Again, the half-advance of the magnetic field is applied to constitute a full-step advance;
4) After that, the particles that enter the ghost cells at the simulation domain boundary during Step 2 should be calibrated with their positions. The periodic boundary treatment on the particles is applied in this scheme;
5) Next, the particles' physical addresses in the data array should be sorted using the Radix sort method due to the operation in Step 4. Meanwhile, the number of particles in each cell is obtained. This procedure prepares for the reduction method of Step 6 in the present step and Step 2 in the next step;
6) Afterward, the conduction current is computed using the contribution from each particle's moving. The reduction treatment, the three-stage current accumulation scheme, and the 2.5D thread-launch strategy are applied in this step;
7) Finally, the electric field is advanced following Ampere's law.

The time expense percentage of each procedure is measured and shown in Fig. 10b. On both types of data center GPUs, the proportions of the particle moving (blue pies) and current computing (deep yellow pies) decrease and are comparable to the time expense of the particle position calibration (cyan pies). This feature is different from the previous scheme, where the former two procedures have the dominant status in each step (Fig. 1).

The ultimate benchmark results of the whole iteration subroutines

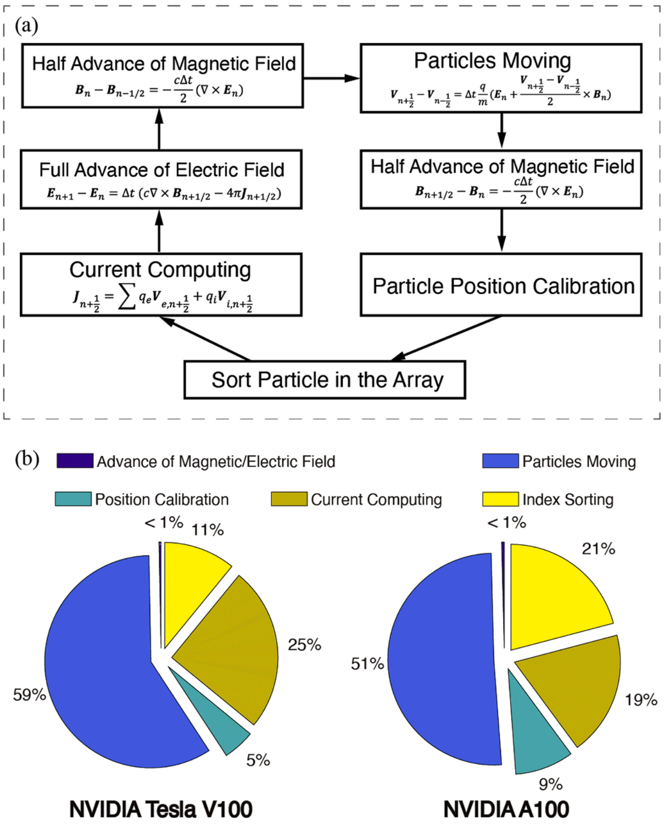

(a)
Half Advance of Magnetic Field
$B_n - B_{n-1/2} = -\frac{c\Delta t}{2}(\nabla \times E_n)$
Particles Moving
$V_{n+\frac{1}{2}} - V_{n-\frac{1}{2}} = \Delta t \frac{q}{m}(E_n + \frac{V_{n+\frac{1}{2}} - V_{n-\frac{1}{2}}}{2} \times B_n)$
Full Advance of Electric Field
$E_{n+1} - E_n = \Delta t\,(c\nabla \times B_{n+1/2} - 4\pi J_{n+1/2})$
Half Advance of Magnetic Field
$B_{n+1/2} - B_n = -\frac{c\Delta t}{2}(\nabla \times E_n)$
Current Computing
$J_{n+\frac{1}{2}} = \sum q_e V_{e,n+\frac{1}{2}} + q_i V_{i,n+\frac{1}{2}}$
Particle Position Calibration
Sort Particle in the Array
(b)
Advance of Magnetic/Electric Field
Particles Moving
Position Calibration
Current Computing
Index Sorting
< 1%
11%
25%
59%
5%
NVIDIA Tesla V100
< 1%
21%
51%
19%
9%
NVIDIA A100

Fig. 10. (a) Flow chat of iteration sequence in the new scheme. (b) Time expense percentage of different procedures in each iteration. These benchmark results are obtained from the simulations using domain size 600×1200 and 100 particles per cell. The time expense of each procedure is the average result of the 10,000 iterations. The percentage is calculated through the method: $t_{pi} / \sum_i t_{pi}$, where $i \in [1,\ 5]$ and t_{pi} refers to the time expense of the procedure i.

are presented in Fig. 11. Comparing the new scheme and the old scheme (Fig. 11a), the V100 series GPUs have highlighted improvements that can achieve close to 5.5 times faster than the previous design (green line in Fig. 11a, grid size 1600×2000). However, when the simulation domain size is small (grid size 400×800), the negative effect would appear (acceleration ratio < 1). From the perspective of the A100 series GPUs, the overall performance can reach as much outstanding as that of the V100 GPUs. And the highest improvement can reach approximately 5 times (orange line in Fig. 11a). Moreover, the acceleration ratio of V100 and A100 GPUs increases steadily as the simulation grid gradually raises, which is an excellent characteristic different from the schemes used in other studies. This valuable signature also brings the advantage to the total acceleration ratio compared with the CPU (orange line in Fig. 11b). The overall performance is enhanced (maximum 734 boosted), and it basically maintains the increasing trend during the increase of the grid size. Meanwhile, the results on V100 GPUs (green line in Fig. 11b) show that the new scheme effectively restrains the sharp drop of the acceleration ratio on V100 GPUs when the domain size increases (see Fig. 5b in Ref. [13]). As a result, it can easily attain an acceleration ratio over 100 in large-scale simulations instead of several dozen.

Discussions

The improvements benefited from the new scheme on the two types of data center GPUs have distinguishable features. On the one hand, it still can attain decent performance in large-scale simulations on GPUs with older architecture. This advantage is predominantly contributed by the proper utilization of the on-chip memory to increase the data transferring speed and reduce redundant data fetching operations. And it is suitable for those GPUs having limited memory rate and bandwidth, such as the legacy data center GPUs (NVIDIA Tesla K80, P100, and V100) and GeForce game GPUs (GTX and RTX series). In other words, the new scheme can guarantee fast access to the simulation results without high-cost GPU hardware upgrades. It also can be realistic to use GPU-embedded Personal Computer (PC) to perform PIC simulations. On the other hand, if it is possible to have access to the HPC center with advanced data center GPUs (NVIDIA A100 and H100), the acceleration ratio can maintain to be increasing and high value when the simulation scale increases. And this feature mainly benefits from the reduced algorithm on the particle treatments. Built with a larger memory size than the former generations, a single advanced GPU can also carry out large-scale simulations within a short time, substituting for the traditional computing model with thousands of CPUs.

Through the profiling tool of NVIDIA Nsight Compute for CUDA kernels, it suggests that the present schemes of particle moving and

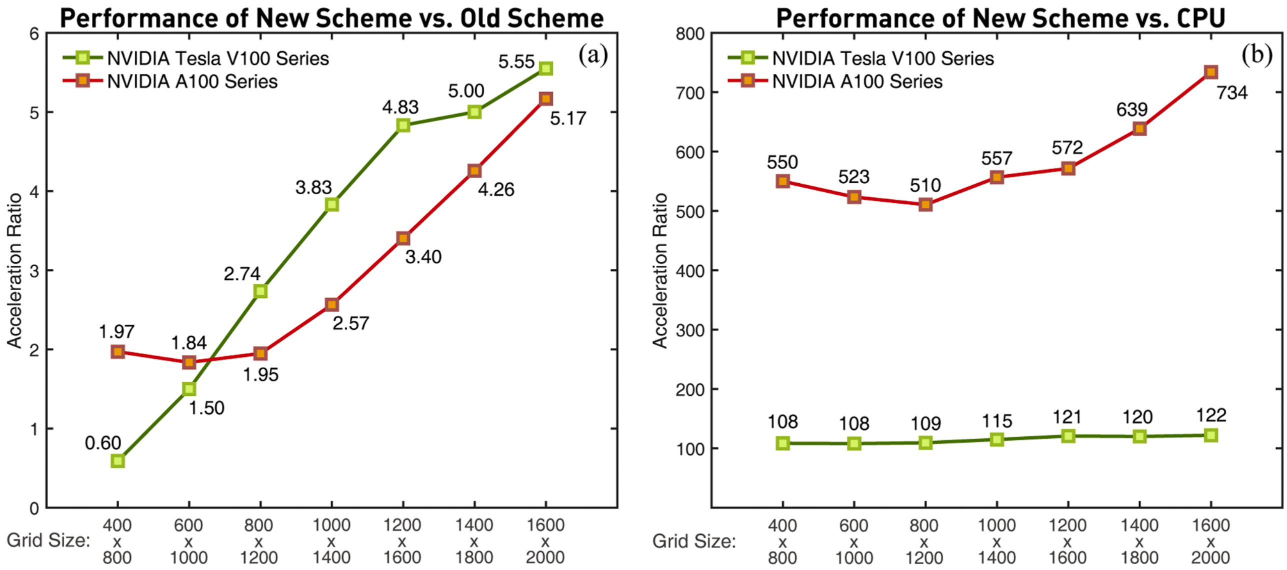


Fig. 11. Comparisons of the overall performances between the new scheme and the old scheme (a) and between the new scheme with GPUs and the CPU version (b). The green lines represent the benchmark results of NVIDIA Tesla V100 series GPUs, and the orange lines stand for the results of NVIDIA A100 series GPUs. The simulation domain sizes are given at the X label of the subfigures, and ppc is fixed at 100. All the benchmark results are the average values of the 10,000 iterations.

current computing have limitations in both memory and computational aspects. From the perspective of memory, it faces the uncoalesced global memory access due to the sorting process. When dealing with the data exchange for the particle array, the threads in a wrap will access different memory addresses across pages. This phenomenon will result in lower L1/TEX and L2 cache hit rates, and the device will spend more time searching addresses and then fetching data. The global memory bandwidth is not fully occupied and expresses latency. For the computational part, it will declare much of the temporary parameters on the registers of each thread during the kernel execution process. And the more complicated numerical operations like dividing and square root also consume more registers. As a result, the live register number of each thread during the computing is large and causes lower wrap occupancy and SM performance covering the corresponding threads. It could be expected to be addressed through the algorithm optimization in the future update. Additionally, the update of GPU architecture could also be expected to mitigate these issues.

Besides the procedures of each iteration step mentioned in Fig. 10a, there are other numerical issues that should also be considered. Since the current $\boldsymbol{J}$ is gathered from the particle's distribution using charge-conserving treatment, the discrepancies between $\nabla\cdot\boldsymbol{E}$ and the charge will gradually accumulate. One possible solution is to apply a filter to the current. The common filter for PIC simulations is $\boldsymbol{J}^f(i) = \alpha\boldsymbol{J}(i) + (1 - \alpha)(\boldsymbol{J}(i-1) + \boldsymbol{J}(i+1))/2$, where $\boldsymbol{J}^f$ is the filtered result and α is the weight factor. In our scheme, the 2D binomial filter is adopted ($\alpha = 0.5$) after the current accumulation from particles. Adding a "pseudo-current" to the Ampere law is another optional method [43,44]. Through this algorithm, the violations of Gauss law and the unphysical behaviors by numerical errors can be greatly prevented. However, more recent studies found that there are no major differences in the simulations with and without pseudo-current, and the numerical errors do not have significant building up as time [45]. For a standard PIC simulation program applied with Yee solver, it is not necessary to add a divergence cleaning process to each iteration step as Gauss law is satisfied in the beginning [46]. Though, it is still essential to consider the numerical errors especially in the simulations requiring enormous iteration steps.

Present kernels of iteration subroutines have already achieved satisfying performance. There are still other technologies or algorithms to boost the program further. The Tensor Core module in the SMs of the GPU device is an alternative computing source [47,48]. Tensor Cores enable mixed-precision computing, dynamically adapting calculations to accelerate throughout while preserving accuracy. The latest generation of Tensor Cores is equipped with AI inference ability to further accelerate computing. This technology can be applied in the process of data smoothing processes originally, for instance, the current smoothing in the iteration of the main program, and the 2D raw data filtering using convolution [49] for output in the diagnosis part. A possible diagram of the current smoothing using Tensor Cores is illustrated in Fig. 12. The convolution of the raw current data preloaded on the shared memory (green grid with deep yellow filling) and the smooth coefficients (green grid with gray filling) can be executed on the Tensor Core module (gray blocks). However, the appliance of the Tensor Cores has certain limitations [50]. If the coefficient size is small, the computing performance on the Tensor Core may be no better than the common FP32 or FP64 (single float point or double float point) calculators of the SMs. In addition, the earlier generation of the Tensor Core (e.g., 2nd on V100 series GPUs) only supports the highest to the FP16 (half-float point) data type multiplication natively. Still, the idea of using Tensor Cores can be reserved in future implementation for a more suitable algorithm.

For the future update of the GPIC program, it is expected to implement the multi-GPUs capability. The crucial point during the scheme design is the communications between different GPUs and the data transfer between the cluster host and GPU devices. These issues could potentially be disposed guided by the NVIDIA Collective Communications Library (NCCL) or the NVIDIA OpenSHMEM (NVSHMEM) application programming interface (API). NCCL is a library providing inter-GPU communication (point-to-point send or receive) primitives that are topology-aware. Besides, NCCL can provide fast data collection over multi-GPUs both within and across hosts, with a variety of interconnection technologies including PCIE (Peripheral Component Interconnect Express), NVLink (NVIDIA Link Bridges), and IB (InfiniBand). And NVSHMEM can provide the host-side interface to allocate symmetric memory distributed across the cluster. This symmetric memory is directly accessible to peer GPU on the host connected via NVLink. These two advanced communication libraries enable the adaptive functions of the clusters with different interconnection approaches. They also can realize faster data fetching process than the traditional model of GPU-aware MPI configuration.

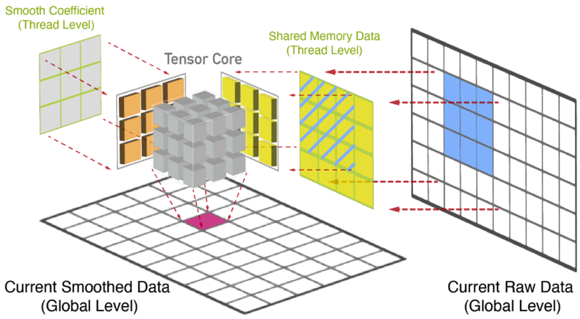


Fig. 12. Sketch of the appliance of the Tensor Core in the data smoothing process. The black grids stand for the matrix data on the global memory. The green grid with deep yellow filling represents the shared memory data loaded from the global memory. The green grid with gray filling represents the coefficients of the smooth process. The combination of the gray cubes, orange blocks, and deep yellow blocks represents the module of Tensor Core.

Data availability

GPIC source code at https://osf.io/u8sn4 or https://doi.org/10.17605/OSF.IO/U8SN4

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (42074196, 41925018) and the Special Fund of Hubei Luojia Laboratory. We would like to thank the mentors for their contribution to this work during 2023 Wuhan University Open Hackathon. The simulation in this work using the NVIDIA Tesla V100–16 G and NVIDIA A100–40 G was carried out on the supercomputing system in the Supercomputing Center of Wuhan University. And the benchmarks of the models NVIDIA Tesla V100–32 G and NVIDIA A100–80 G were performed on the AI Galaxy GPUs cloud server platform of Shanghai Gencong Information Technology Co., Ltd.

References

[1] M.I. Sitnov, M. Swisdak, A.V. Divin, Dipolarization fronts as a signature of transient reconnection in the magnetotail, J. Geophys. Res. 114 (2009) A04202, https://doi.org/10.1029/2008JA013980.

[2] W. Daughton, V. Roytershteyn, H. Karimabadi, et al., Role of electron physics in the development of turbulent magnetic reconnection in collisionless plasma, Nat. Phys. 7 (2011) 539–542, https://doi.org/10.1038/nphys1965.

[3] M. Zhou, X.H. Deng, S.Y. Huang, Electric field structure inside the secondary island in the reconnection diffusion region, Phys. Plasma 19 (2012), 042902, https://doi.org/10.1063/1.3700194.

[4] S.Y. Huang, M. Zhou, Z.G. Yuan, et al., Kinetic simulations of secondary reconnection in the reconnection jet, J. Geophys. Res. Space Phys. 120 (2015) 6188–6198, https://doi.org/10.1002/2014JA020969.

[5] Q. Lu, Y. Ke, X. Wang, et al., 2019, Two-dimensional gcpic simulation of rising-tone chorus waves in a dipole magnetic field, J. Geophys. Res. Space Phys. 124 (2019) 4157–4167, https://doi.org/10.1029/2019JA026586.

[6] Q.Y. Xiong, S.Y. Huang, Z.G. Yuan, et al., Distribution of negative J·E' in the inflow edge of the inner electron diffusion region during tail magnetic reconnection: simulations Vs. observations, Geophys. Res. Lett. 49 (2022), e2022GL098445, https://doi.org/10.1029/2022GL098445.

[7] V.K. Decyk, & T.V. Singh, Adaptable Particle-in-Cell algorithms for graphical processing units, Comput. Phys. Commun. 182 (2011) 641–648, https://doi.org/10.1016/j.cpc.2010.11.009.

[8] V.K. Decyk, & T.V. Singh, Particle-in-Cell algorithms for emerging computer architectures, Comput. Phys. Commun. 185 (2014) 708–719, https://doi.org/10.1016/j.cpc.2013.10.013.

[9] H. Burau, R. Widera, W. Hönig, et al., PIConGPU: a fully relativistic particle-in-cell code for a GPU cluster, IEEE Trans. Plasma Sci. 38 (2010) 2831–2839, https://doi.org/10.1109/TPS.2010.2064310.

[10] P. Abreu, R.A. Fonseca, J.M. Pereira, et al., PIC codes in new processors: a full relativistic PIC code in CUDA-enabled hardware with direct visualization, IEEE Trans. Plasma Sci. 39 (2011) 2, https://doi.org/10.1109/TPS.2010.2090905.

[11] S.W.D. Chien, J. Nylund, G. Bengtsson, et al., sputniPIC: an Implicit Particle-in-Cell Code for Multi-GPU Systems, in: 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2020, https://doi.org/10.1109/SBAC-PAD49847.2020.00030.

[12] R. Bird, N. Tan, S.V. Luedtke, et al., VPIC 2.0: next Generation Particle-in-Cell Simulations, IEEE Trans. Parallel Distrib. Syst. 33 (2022) 952–963, https://doi.org/10.1109/TPDS.2021.3084795.

[13] Q.Y. Xiong, S.Y. Huang, Z.G. Yuan, et al., A Scheme of Full Kinetic Particle-in-cell Algorithms for GPU Acceleration Using CUDA Fortran Programming, Astrophys. J. Suppl. S. 264 (2023) 3, https://doi.org/10.3847/1538-4365/ac9fd6.

[14] Q.M. Lu, & D.S. Cai, Implementation of parallel plasma particle-in-cell codes on PC cluster, Comput. Phys. Commun. 135 (2001) 93–104, https://doi.org/10.1016/S0010-4655(00)00227-7.

[15] H. Shah, S. Kamaria, R. Markandeya, M. Shah, B. Chaudhury, A novel implementation of 2D3V Particle-in-Cell (PIC) algorithm for kepler GPU architecture, in: 2017 IEEE 24th International Conference on High Performance Computing, 2017, pp. 378–387, https://doi.org/10.1109/HiPC.2017.00050.

[16] S. Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (2013). ISBN: 9780124159334.

[17] J. Cheng, M. Grossman, T. McKercher, Professional CUDA C Programming, John Wiley & Sons, Inc, 2014. ISBN: 9781118739327.

[18] T. Soyata, GPU Parallel Program Development Using CUDA (2018). ISBN: 9781498750752.

[19] K. Yee, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media, IEEE Trans. Antennas Propag. 14 (1966) 3, https://doi.org/10.1109/TAP.1966.1138693.

[20] M.F. Hadi, & S.A. Esmaeili, CUDA Fortran acceleration for the finite-difference time-domain method, Comput. Phys. Commun. 184 (2013) 1395–1400, https://doi.org/10.1016/j.cpc.2013.01.006.

[21] X. Wang, S. Liu, X. Li, S. Zhong, GPU-Accelerated Finite-Difference Time-Domain Method for Dielectric Media Based on CUDA, Int. J. RF Microwave Comput. Aided Eng. 26 (2016) 512–518, https://doi.org/10.1002/mmce.20997.

[22] G. Ruetsch, M. Fatica, CUDA Fortran for Scientists and Engineers: Best Practices for Efficient CUDA Fortran Programming (2013). ISBN: 9780124169708.

[23] J. Porter-Sobieraj, S. Cygert, D. Kikoła, J. Sikorski, M. Słodkowski, Optimizing the computation of a parallel 3D finite difference algorithm for graphics processing units, Concurr. Comput.: Pract. Exper. 27 (2015) 1591–1602, https://doi.org/10.1002/cpe.3351.

[24] E.E. Franco, H.M. Barrera, S. Laín, 2D lid-driven cavity flow simulation using GPU-CUDA with a high-order finite difference scheme, J. Brazil. Soc. Mech. Sci. Eng. 37 (2015) 1329–1338, https://doi.org/10.1007/s40430-014-0260-x.

[25] S.R. Miri Rostami, &.M. Ghaffari-Miab, Finite difference generated transient potentials of open-layered media by parallel computing using OpenMP, MPI, OpenACC, and CUDA, IEEE Trans. Antennas Propag. 67 (2019) 10, https://doi.org/10.1109/TAP.2019.2920253.

[26] O. Buneman, The advance from 2D electrostatic to 3D electromagnetic particle simulation, Comput. Phys. Commun. 12 (1976) 21–31, https://doi.org/10.1016/0010-4655(76)90007-2.

[27] J.P. Boris, Relativistic plasma simulation-optimization of a hybrid code, in: Proceedings of the Fourth Conference on Numerical Simulation Plasma, Naval Research Laboratory, Washington, D.C, 1970, pp. 3–67.

[28] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation (2004). ISBN: 9780750310253.

[29] S. Fatemi, A.R. Poppe, G.T. Delory, W.M. Farrell, AMITIS: a 3D GPU-based hybrid-PIC model for space and plasma physics, J. Phys. Conf. Ser. 837 (2017), 012017, https://doi.org/10.1088/1742-6596/837/1/012017.

[30] W. Gou, S. Zhang, Y. Zheng, Implementation of the moving particle semi-implicit method for free-surface flows on GPU clusters, Comput. Phys. Commun. 244 (2019) 13–24, https://doi.org/10.1016/j.cpc.2019.07.010.

[31] V. Vahedi, &.M. Surendra, A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges, Comput. Phys. Commun. 87 (1995) 179–198, https://doi.org/10.1016/0010-4655(94)00171-W.

[32] M.S. Rosin, L.F. Ricketson, A.M. Dimits, et al., Multilevel Monte Carlo simulation of Coulomb collisions, J. Comput. Phys. 274 (2014) 140–157, https://doi.org/10.1016/j.jcp.2014.05.030.

[33] S. Mattei, K. Nishida, M. Onai, et al., A fully-implicit Particle-In-Cell Monte Carlo Collision code for the simulation of inductively coupled plasma, J. Comput. Phys. 350 (2017) 891–906, https://doi.org/10.1016/j.jcp.2017.09.015.

[34] M.Y. Hur, J.S. Kim, I.C. Song, J.P. Verboncoeur, H.J. Lee, Model description of a two-dimensional electrostatic particle-in-cell simulation parallelized with a graphics processing unit for plasma discharges, Plasma Res. Express 1 (2019), 015016, https://doi.org/10.1088/2516-1067/ab0918.

[35] Z. Juhasz, J. Ďurian, A. Derzsi, et al., Efficient GPU implementation of the Particle-in-Cell/Monte-Carlo collisions method for 1D simulation of low-pressure capacitively coupled plasma, Comput. Phys. Commun. 263 (2021), 107913, https://doi.org/10.1016/j.cpc.2021.107913.

[36] J. Villasenor, &.O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69 (1992) 306–316, https://doi.org/10.1016/0010-4655(92)90169-Y.

[37] I.V. Sokolov, Alternating-order interpolation in a charge-conserving scheme for particle-in-cell simulations, Comput. Phys. Commun. 184 (2013) 320–328, https://doi.org/10.1016/j.cpc.2012.09.015.

[38] T. Umeda, Y. Omura, T. Tominaga, H. Matsumoto, A new charge conservation method in electromagnetic particle-in-cell simulations, Comput. Phys. Commun. 156 (2003) 73–85, https://doi.org/10.1016/S0010-4655(03)00437-5.

[39] H.V. Dang, &.B. Schmidt, CUDA-enabled Sparse Matrix-Vector Multiplication on GPUs using atomic operations, Parallel Comput. 39 (2013) 737–750, https://doi.org/10.1016/j.parco.2013.09.005.

[40] J. Mašek, &.M. Vořechovský, Parallel implementation of hyper-dimensional dynamical particle system on CUDA, Adv. Eng Softw. 125 (2018) 178–187, https://doi.org/10.1016/j.advengsoft.2018.03.009.

[41] X. Kong, M.C. Huang, C. Ren, & V.K. Decyk, Particle-in-cell simulations with charge-conserving current deposition on graphic processing units, J. Comput. Phys., 230 (230) 1676–1685. doi:10.1016/j.jcp.2010.11.032.

[42] K.G. Miller, R.P. Lee, A. Tableman, et al., Dynamic load balancing with enhanced shared-memory parallelism for particle-in-cell code, Comput. Phys. Commun. 259 (2021), 107633, https://doi.org/10.1016/j.cpc.2020.107633.

[43] B. Marder, A method for incorporating Gauss' law into electromagnetic PIC codes, J. Comput. Phys. 68 (1987) 48–55, https://doi.org/10.1016/0021-9991(87)90043-X.

[44] P.J. Mardahl, & J.P. Verboncoeur, Charge conservation in electromagnetic PIC codes; spectral comparison of Boris/DADI and Langdon-Marder methods, Comput. Phys. Commun. 106 (1997) 219–229, https://doi.org/10.1016/S0010-4655(97)00094-5.

[45] S. Markidis, &.G. Lapenta, The energy conserving particle-in-cell method, J. Comput. Phys. 230 (2011) 7037–7052, https://doi.org/10.1016/j.jcp.2011.05.033.

[46] J.L. Vay, C.G.R. Geddes, E. Cormier-Michel, D.P. Grote, Numerical methods for instability mitigation in the modeling of laser wakefield accelerators in a Lorentz-boosted frame, J. Comput. Phys. 230 (2011) 5908–5929, https://doi.org/10.1016/j.jcp.2011.04.003.

[47] S. Markidis, S.W.D. Chien, E. Laure, I.B. Peng, J.S. Vetter, NVIDIA Tensor core programmability, performance & precision, in: 2018 IEEE International Parallel and Distributed Processing Symposium Workshops, 2018, pp. 522–531, https://doi.org/10.1109/IPDPSW.2018.00091.

[48] J. Choquette, W. Gandhi, O. Giroux, N. Stam, R. Krashinsky, NVIDIA A100 Tensor Core GPU: performance and Innovation, IEEE Micro 41 (2021) 2, https://doi.org/10.1109/MM.2021.3061394.

[49] H.H. Chang, & Y.-N. Chang, CUDA-based acceleration and BPN-assisted automation of bilateral filtering for brain MR image restoration, Med. Phys. 44 (2017), https://doi.org/10.1002/mp.12157.

[50] H. Ootomo, &.R. Yokota, Recovering single precision accuracy from Tensor Cores while surpassing the FP32 theoretical peak performance, Int. J. High Perform. Comput. Appl. 36 (2022) 475–491, https://doi.org/10.1177/10943420221090256.

Geophysical Research Letters®

RESEARCH LETTER
10.1029/2023GL105300

Key Points:

- The magnetic tension force causes the deceleration of the electrons in the outer electron diffusion region (EDR) during magnetic reconnection
- Partial electrons are decelerated and even move back to the inner EDR, and they are accelerated again and attain higher energy
- The electron backflow motion in the outer EDR indicates a magnetic Marangoni effect in space plasma

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

S. Y. Huang,
shiyonghuang@whu.edu.cn

Citation:

Xiong, Q. Y., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Lin, R. T., & Yu, L. (2023). Electron backflow motions in the outer electron diffusion region during magnetic reconnection. *Geophysical Research Letters*, *50*, e2023GL105300. https://doi.org/10.1029/2023GL105300

Received 5 JUL 2023
Accepted 29 SEP 2023

Electron Backflow Motions in the Outer Electron Diffusion Region During Magnetic Reconnection

Q. Y. Xiong[1], S. Y. Huang[1], Z. G. Yuan[1], K. Jiang[1], S. B. Xu[1], R. T. Lin[1], and L. Yu[1]

[1]School of Electronic Information, Hubei Luojia Laboratory, Wuhan University, Wuhan, China

Abstract Magnetic reconnection is a fundamental physical process of rapidly converting magnetic energy into particles. The electron diffusion region (EDR) is the crucial region during magnetic reconnection. The outer EDR, which also plays a crucial role in magnetic reconnection, is responsible for energy conversion. In the outer EDR, the electrons are decelerated and return the energy to the magnetic field on the pileup region behind the reconnection front. In the present study, we used the fully kinetic particle-in-cell simulation and revealed that part of decelerated electrons in the outer EDR could even move back to the inner EDR. This phenomenon is caused by the dominant contribution from the magnetic tension force, and it suggests a magnetic Marangoni effect in space plasma, similar to the Marangoni effect in fluids. Our results potentially propose a brand-new physical process and a novel mechanism in the EDR during magnetic reconnection.

Plain Language Summary Plasma's energy can be changed through various approaches in the universe, and magnetic reconnection is one of those approaches to convert energy from the magnetic field to the plasma. In the reconnection site, the inner electron diffusion region (EDR) is an essential area where the energy is released, and the electron's energy is enhanced significantly. Meanwhile, in the outer EDR, the electrons are decelerated by the electric field, thus their energy decreases. However, part of those electrons can move backward to the inner EDR, and how this phenomenon comes up has no further investigation. In this study, we use numerical simulations to reveal the possible mechanism of this kind of electron's motion. It is found that the electron deceleration is caused by the magnetic tensor force. The electrons with specific conditions have the possibility to move backward. Those backflow electrons have a second chance to be accelerated again in the inner EDR. Such electron motion in plasma physics is not a kind of gyro movement but might indicate a so-called magnetic Marangoni effect similar to the Marangoni effect in fluid physics. Our findings propose a novel mechanism associated with electron acceleration in the EDR during magnetic reconnection.

1. Introduction

Magnetic reconnection, involved with active energy conversions between electromagnetic field and particles, is abundant in the solar wind (e.g., Phan et al., 2020), terrestrial magnetopause (e.g., Burch et al., 2018), magnetosheath (e.g., Phan et al., 2018), magnetotail (e.g., Huang et al., 2012, 2018; Lu et al., 2020), and interplanetary space (e.g., Singh et al., 2015). The reconfiguration of the magnetic reconnection topology is initialed with the electron diffusion region (EDR), where the magnetic field lines reorganize, and the electron frozen-in condition is broken (nonzero $\mathbf{E}' = \mathbf{E} + \mathbf{V}_e \times \mathbf{B}$). The EDR is divided into inner and outer EDR. The magnetic field converts its energy to the particles in the inner EDR and heats/accelerates the electrons (e.g., Burch et al., 2016b; Huang et al., 2021; Jiang et al., 2019; Torbert et al., 2018). In contrast, the outer EDR has the opposite process with negative energy conversion measurements ($\mathbf{J}\cdot\mathbf{E}' < 0$) compared with the inner EDR (e.g., Hwang et al., 2017; Karimabadi et al., 2007; Shay et al., 2007; Xiong et al., 2022c; Zenitani et al., 2012). These two regions regulate the reconnection regimes and manage the energy release budget.

The outer EDR connects the reconnection site and the pileup region downstream of the outflow exhaust. The high-speed electron jets from the inner EDR outrun the magnetic field in the outer EDR (e.g., Karimabadi et al., 2007; Shay et al., 2007), returning the energy to the magnetic field. This energy propels the magnetic flux accumulation behind the reconnection front (RF). The electric field at the electron rest frame, induced by the changing magnetic field and working as the temporary energy carrier, decelerates the electrons passing the outer EDR (e.g., Xiong et al., 2022c). From another perspective, the electrons can be regarded as magnetic generators during partial remagnetization (e.g., Payne et al., 2021). These outcomes denote that the energy budget in the outer EDR is highly correlative to the electron dynamics.

In the present study, we focus on the energy conversion in the outer EDR, where the negative **J**·**E**′ can be well balanced by the work done by the Lorentz force in both in-situ observation and PIC simulation. The dominant component is the magnetic tension force induced by the out-of-plane current. Through the single trajectory analysis, it is found that the electrons can be influenced by the magnetic tension force and even move back to the inner EDR and then be accelerated again. This backflow motion of the electrons is not a gyro motion but might be related to the so-called "magnetic Marangoni effect." It is also suggested that the electrons have a second chance to carry the energy from the reconnection site to the pileup region.

2. Instruments and Simulation Setup

The observational data from Magnetospheric Multiscale (MMS) mission (Burch et al., 2016a) with high resolution are used in this study. The magnetic field, the electric field, and the particle moments are from the Fluxgate Magnetometer (FGM) (Russell et al., 2016), the Electric Double Probe (EDP) (Ergun et al., 2016; Lindqvist et al., 2016), and the Fast Plasma Investigation (FPI) (Pollock et al., 2016), respectively.

The PIC simulation is performed using Harris current sheet equilibrium under zero guide field configuration (e.g., Huang et al., 2014, 2015; Xiong et al., 2022a, 2022b, 2022c; Zhou et al., 2012). The simulation domain size is 1,200 × 1,800 cells, and 200 pairs of ions and electrons are deposited in each cell. The light speed is $c/v_A = 30$, where v_A is the Alfvén speed. The mass ratio (m_i/m_e) is 100. The ion inertial length (d_i) is 60 cells. The temperature ratio between ion and electron (T_i/T_e) is 5. The plasma frequency and gyrofrequency ratio of electron (ω_{pe}/ω_{ce}) is 3. The normalized units and the methodology of electron trajectory analysis are detailed illustrated in Xiong et al. (2022a).

3. Results

We start from the outer EDR situations with both MMS observations and PIC simulation. The left part of Figure 1 is the event that MMS4 encountered the outer EDR of magnetic reconnection on 19 September 2015. The local coordinate vectors LMN and the reconnection signatures of this event have been addressed in the previous study (Hwang et al., 2017). Notably, current sheet crossing (Figure 1a) and high-speed outflow electron jets are detected (Figures 1c and 1d), as well as the nonzero electron frozen-in condition (Figure 1e). These features indicate that the spacecraft has traveled through the EDR. Here we mainly focus on energy balance characteristics in the outer EDR (shallow yellow region). Energy conversion (black line in Figure 1f) is primarily negative (Figure 1f), implying that the energy is converted from the particles to the magnetic field. Using the Poynting theorem:

$$\left(\partial B^2/2\mu_0\right)/\partial t + \nabla\cdot\mathbf{S} = -\mathbf{J}\cdot\mathbf{E}' - \mathbf{V}_{\rm e}\cdot(\mathbf{J}\times\mathbf{B}) \tag{1}$$

The negative **J**·**E**′ in the outer EDR, which is predominantly caused by the electron deceleration (e.g., Xiong et al., 2022c), is counter-balanced by the work done by the Lorentz force (blue line in Figure 1f). In other words, the Lorentz force in this region contributes to the electron deceleration.

There is a virtual satellite trajectory roughly sketched in the simulation results at the time $t\Omega_{\rm ci} = 36$ for comparison. From the simulation perspective, the work done by the Lorentz force on the electrons in the outer EDR (magenta dashed square in Figure 1h) can well balance the term **J**·**E**′ (Figure 1g) at the identical position, consistent with the observations (shallow yellow region in Figure 1f). The Lorentz force can be decomposed into the magnetic pressure force part and the tension force part:

$$\mathbf{J}\times\mathbf{B} = (\nabla\times\mathbf{B}/\mu_0)\times\mathbf{B} = -\nabla B^2/2\mu_0 + \mathbf{B}\cdot\nabla\mathbf{B}/\mu_0 \tag{2}$$

The work done by these two parts is also shown in Figures 1i and 1j. Apparently, the main contributor to the Lorentz force comes from the magnetic tension force. Therefore, the deceleration motion of the electrons in the outer EDR is dominantly controlled by the magnetic tension force.

To figure out how the magnetic tension force functions on the electrons, the term $\mathbf{V}_{\rm e}\bullet(\mathbf{B}\bullet\nabla\mathbf{B}/\mu_0)$ is further decomposed into its three-direction components. The X component (Figure 2a) has a high positive value, corresponding with the total volume of the magnetic tension force work (Figure 1i). The Y component (Figure 2b) is negative and counter-balanced mainly by the X component, while the Z component's contribution is negligible. Electron energy gain from the magnetic tension force in the Y direction cannot cover the loss in the X direction, thus

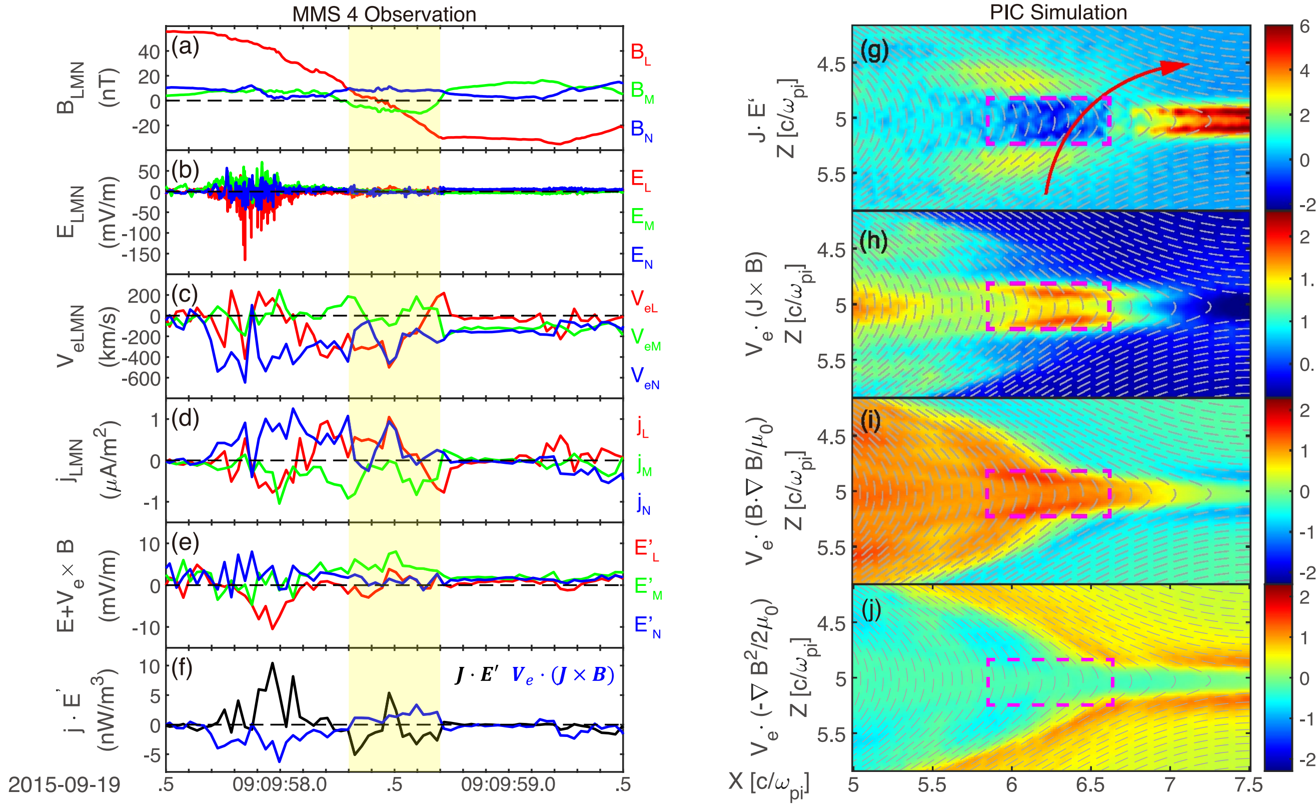


Figure 1. (a) Magnetic field; (b) Electric field; (c) Electron bulk velocity; (d) Current using plasma momentum; (e) Electric field in electron rest frame; and (f) Energy conversion (black line) and Lorentz work (blue line). In the right part, (g) Energy conversion; (h) Lorentz work; (i) Magnetic tension force work; and (j) Magnetic pressure work. The shallow yellow region of the left part delimits the outer electron diffusion region (EDR) boundary. The red arrow in panel (g) is the virtual satellite trajectory, and the magenta dashed squares in panels (h–j) mark the outer EDR boundary.

resulting in the net positive work of magnetic tension force. The directions of electron bulk velocity and magnetic tension force are the same in the X direction (Figures 2d and 2g) while opposite in the Y direction (Figures 2e and 2h), consequently leading to the deceleration and acceleration in X and Y directions, respectively. Noticeably, the magnetic tension force strength in the X direction is relatively large (Figure 2g), which takes the most responsibility for the electron deceleration motion.

Next, we attempt to discuss how the magnetic tension force forms in the outer EDR during the reconnection. The X and Y components of magnetic tension force can be split as follows:

$$(\mathbf{B}\cdot\nabla\mathbf{B}/\mu_0)_x = (B_x(\partial B_x)/\partial x + B_z(\partial B_x)/\partial z)/\mu_0 \tag{3}$$

$$(\mathbf{B}\cdot\nabla\mathbf{B}/\mu_0)_y = (B_x(\partial B_y)/\partial x + B_z(\partial B_y)/\partial z)/\mu_0 \tag{4}$$

And the four terms at the right-hand side of Equations 3 and 4 are shown in Figure S1 in Supporting Information S1. The term $B_z(\partial B_x)/\partial z$ is expected to take the dominant effect on the magnetic tension force X component (Figure S1b in Supporting Information S1). With the continuing reconnection process, it is suggested that the pileup effect behind the RF (B_z, Figure S1f in Supporting Information S1) and the contraction of the current sheet ($\partial B_x/\partial z$, Figure S1g in Supporting Information S1) contribute to forming the magnetic tension force in the X direction. Besides, the magnetic field has the relation of $\nabla\times\mathbf{B} = \mu_0\mathbf{J}$, and its Y component can be expanded as $\partial B_x/\partial z - \partial B_z/\partial_x = \mu_0 J_y$. As a result, the high out-of-plane current (J_y, Figure S1j in Supporting Information S1) impels to form the term $\partial B_x/\partial z$. And the term $B_z(\partial B_x)/\partial z$ evolves into the condition as Figure S1b in Supporting Information S1 shows. Meanwhile, the primary carriers of the current in the EDR are electrons (e.g., Xiong et al., 2022b). Therefore, as the reconnection continues, the energy-enhanced electron jets are constantly poured out toward the outer EDR, forming the intense current. The gradually accumulated current contributes to the magnetic tension force formation. Conversely, this generated tension force then decelerates the electrons passing the outer EDR.

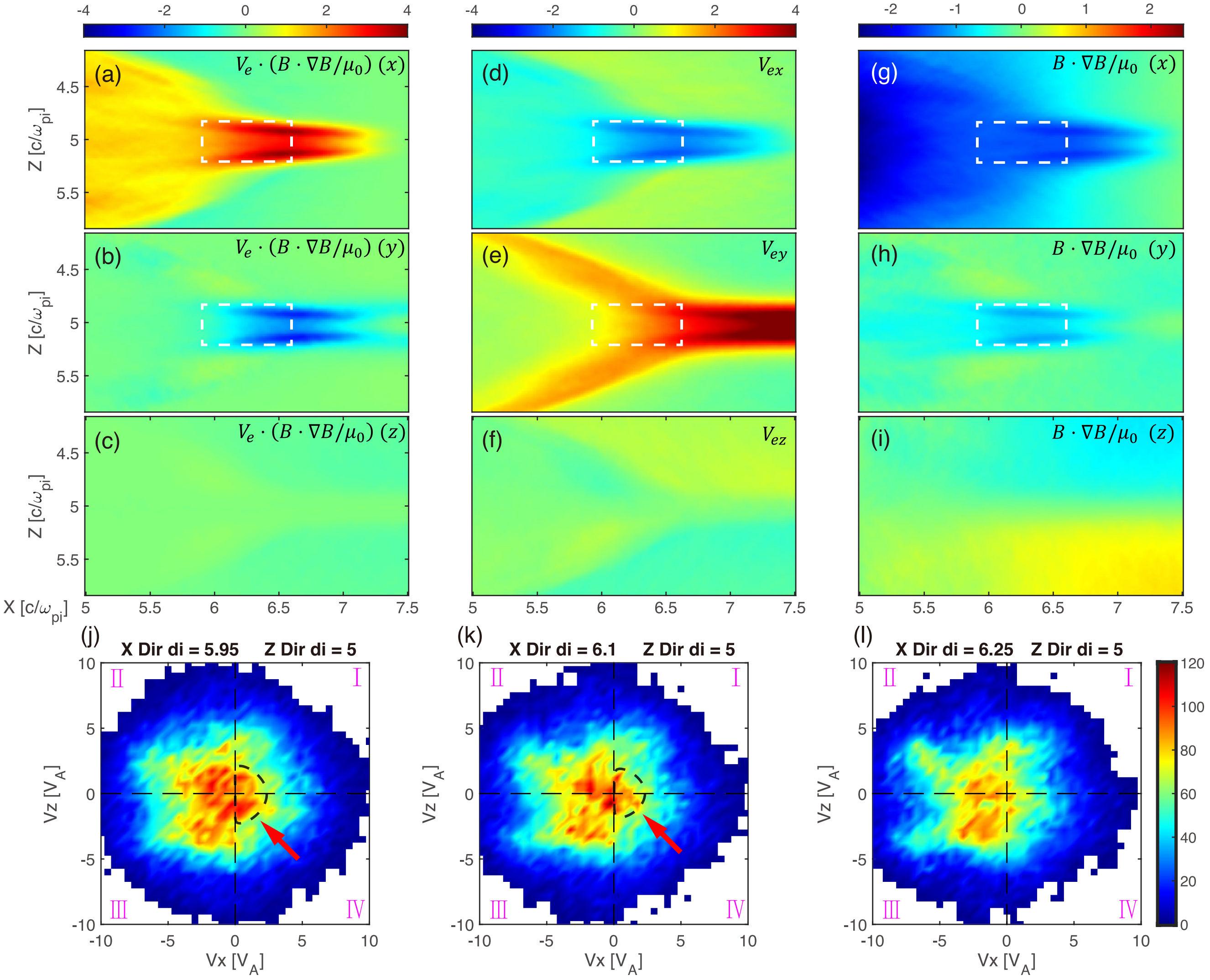


Figure 2. (a–c) Three components of magnetic tension force work, (d–f) electron bulk velocity, (g–i) magnetic tension force, (j–l) Electron velocity distribution at three positions of the outer electron diffusion region (EDR) ($Z = 5d_i$ and $X = 5.95d_i$, $6.1d_i$, and $6.25d_i$). The white dashed squares in panels (a, b), (d, e), and (g, h) are the boundary of the outer EDR.

The variance of the electron distributions passing the outer EDR also shows distinguishable properties. In Figures 2l–2j, the electron velocity distributions (EVD) in the outer EDR are presented to show the motion change of the electrons along the outflow direction. At the position $Z = 5\ d_i$ and $X = 6.25\ d_i$ which is closer to the inner EDR (Figure 2l), the EVD almost gathers in the II and III quadrants of the distribution panel, which means the electron jets toward the outflow direction. However, as the position is gradually farther from the inner EDR (from Figures 2j to 2l), the EVD gradually shows the trend of a significant population in the I and IV quadrants (black dashed semicircles in Figures 2j and 2k). This suggests that electrons are decelerated and even part of them have the possibility to move backward, propelled by the magnetic tension force in the outer EDR.

In Figure 3, one typical electron with such backward motion in the outer EDR is picked up, which has a similar orbit in previous research (e.g., Shuster et al., 2015), and its physical conditions along the traveling path are also shown. This electron is accelerated in the inner EDR and then ejected toward the outflow direction (X− direction). Afterward, the electron goes through the outer EDR and is decelerated even moving backward at the position around $X = 5.5\ d_i$, where it is forced mainly at X+ direction (red arrows in Figure 3a). During this deceleration process which lasts approximately 0.11 Ω_{ci}^{-1} (11 Ω_{ce}^{-1}), the electron's energy constantly decreases (color level in Figure 3b). Next, the electron travels back to the inner EDR and carries the energy again through the acceleration process at this region. Its moving direction comes to the second reverse at around $X = 6.8\ d_i$,

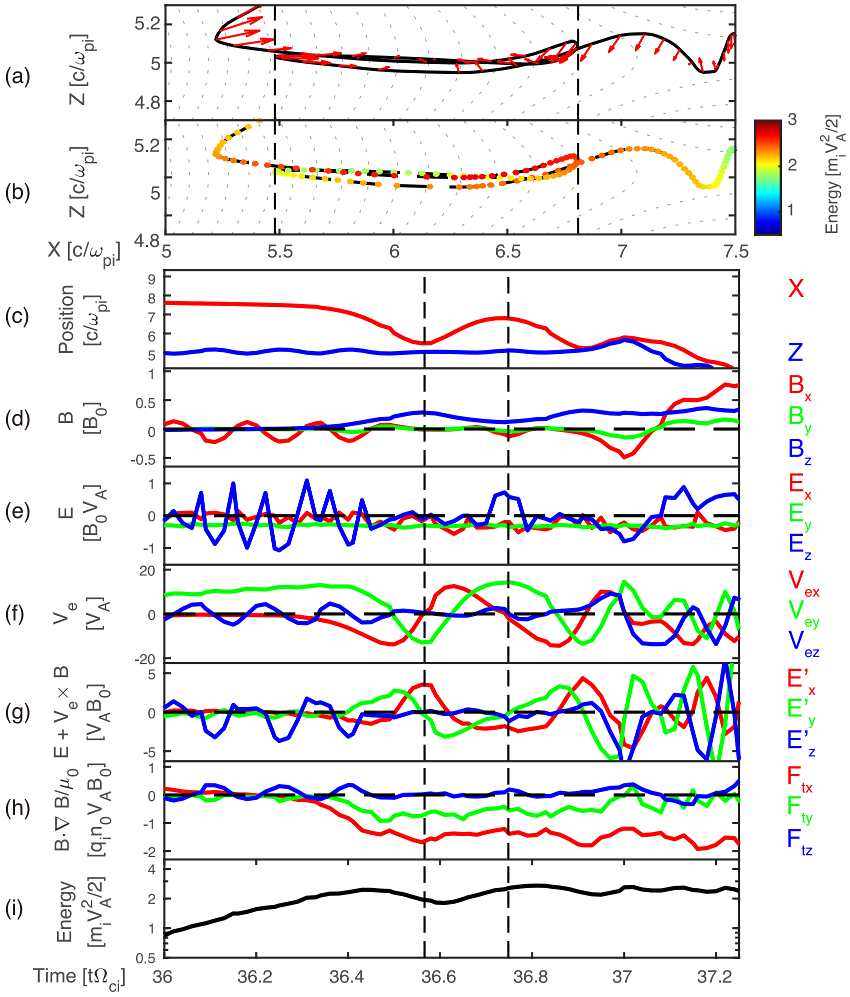


Figure 3. The red arrows in panel (a) represent the force direction and the relative magnitude. Color dots in panel (b) represent the electron energy. (c) Electron position in the 2D plane; (d, e) Magnetic field and electric field along the trajectory; (f) Electron velocity; (g) Electron frozen-in condition; (h) Magnetic tension force along the trajectory; and (i) Electron energy. Two vertical dashed lines mark the two moving direction reverse points.

and begins to move into the outer EDR again. Likewise, the electron certainly loses energy during this traveling phase. Eventually, the electron passes the EDR and reaches the exhaust region of the reconnection.

Meanwhile, time variations of the physical quantities of this electron and electromagnetic fields are also recorded and shown in Figures 3c–3i. During the early time when the electron is inside the inner EDR ($t\Omega_{ci} < 36.5$), the typical energization process of the electron can be identified (Figure 3i). Before the first reversal, the electron travels through the outer EDR and experiences deceleration mainly along the *X* direction (around $t\Omega_{ci} = 36.5$, Figure 3f). At the first reverse time (around $t\Omega_{ci} = 36.58$), the electron reaches the magnetic field pileup region (Figure 3d). It gains much velocity in the *Y* direction (Figure 3f), thus being forced mainly toward the *X*+ direction (Figure 3g). Around the second reverse time (around $t\Omega_{ci} = 36.67$), the electron returns to the inner EDR and is ready for the second acceleration process. During the whole traveling time in the EDR, the electron's energy has a temporary low value (around $t\Omega_{ci} = 36.6$, Figure 3i). This electron's reverse motion in the outer EDR allows it to experience the process twice, during which the electron carries the energy from the reconnection site to the pileup region.

We have screened out 110 electrons that have a similar backflow motion in the outer EDR from the total 4,453 recorded electron data in Xiong et al. (2022a). The characteristics of the backflow electrons are captured and shown in the statistical form in Figure 4. When these electrons begin to enter the outer EDR, their velocity angles versus the local magnetic field are different from the conditions of whole-tracked electrons (Figure 4a). Furthermore, the velocity distribution differences between the backflow electrons and the others within the two vertical dashed lines are

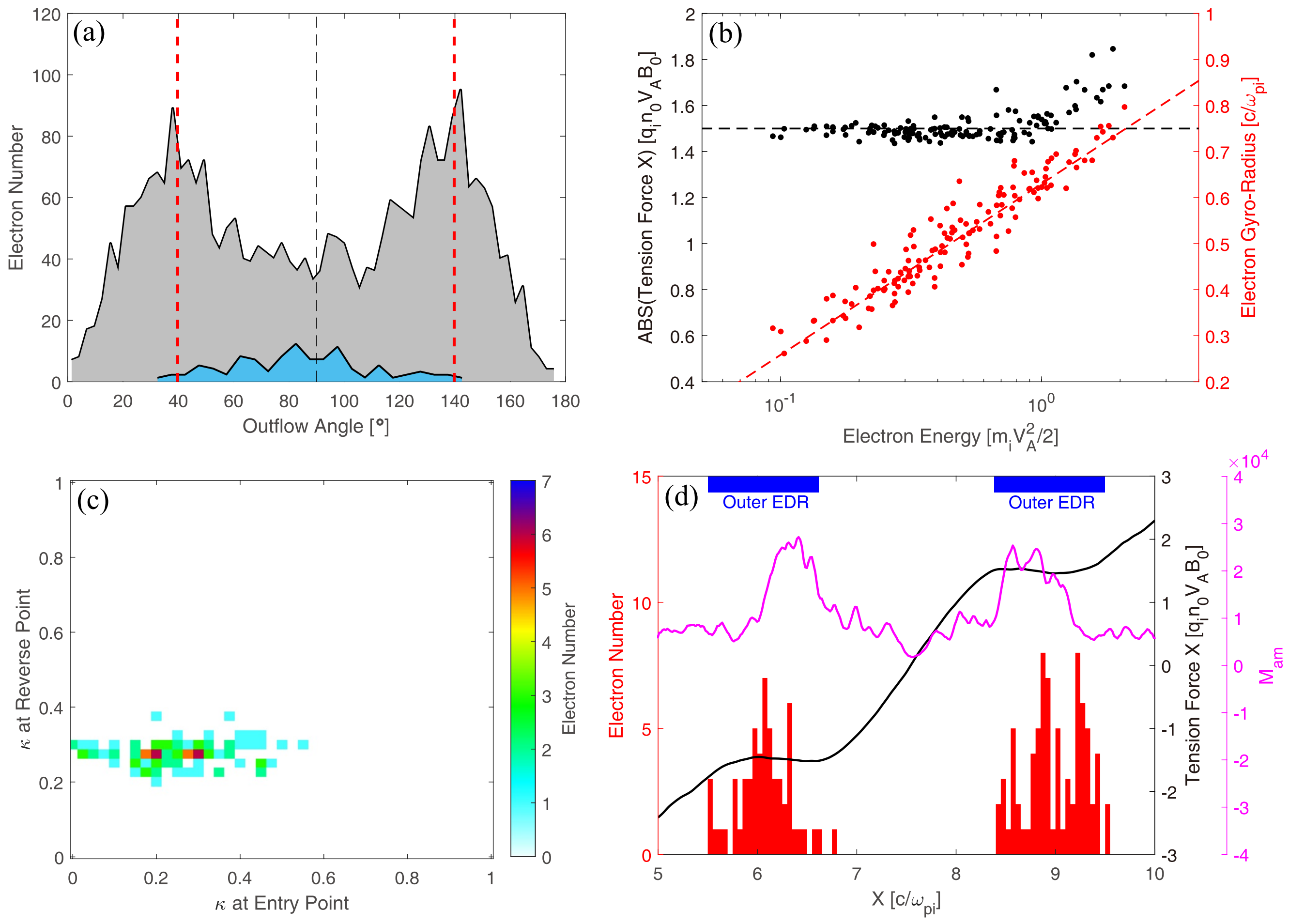


Figure 4. (a) Backflow motion electrons (blue area) and all tracked electrons (gray area) outflow angles at the entry point of the outer electron diffusion region. (b) Black dots are the *X* component of magnetic tension forces when the electrons begin to move backward, and the black dashed line is the fitting result; red dots are the electron gyro-radius at the reverse point, and the red dashed line is the fitting result. (c) Joint distribution of electron curvature parameter at both the entry point and the reverse point. (d) The red bars represent the electron number at the corresponding position; the back line is the 1D slice of the *X* component of the magnetic tension force along the *X* direction. The magenta line is the 1D slice of the magnetic Marangoni number.

presented in Figure S2 in Supporting Information S1, where it is suggested that the backflow electrons initially tend to have a higher outflow and lower *Z* direction velocity (Figures S2b and S2e in Supporting Information S1). At the reverse point, the electron's energy varies in a widespread energy level (Figure 4b). The absolute value of magnetic tension force along the *X* direction, almost gathers at around 1.5 normalized unit, though the force increases slightly when the electron energy is more significant than 1 normalized unit. This feature is reasonable because the higher electron's energy, the more significant force is needed to turn the electron backward. Based on the variations of electron energy during the backflow motion, we also compare the energy differences of all backflow electrons between the first entering and the last entering the outer EDR, and the results are presented in Figure S3 in Supporting Information S1. Most electrons can obtain higher energy than the first time they enter the outer EDR (Figure S3a in Supporting Information S1). Besides, the magnetic Marangoni effect has more significant energy enhancement on the electrons with lower energy when they first enter the outer EDR (Figure S3b in Supporting Information S1). This feature indicates that the electron backflow motion can provide more chances of acceleration for part of the electrons having low energy during the first acceleration in the inner EDR to reach a higher energy level.

Meanwhile, these electrons' gyro-radii are also calculated. The linear correlation between the electron energy and the gyro-radius can be obtained. During the electrons' outflow process, their motions are governed by the curvature parameter (e.g., Büchner & Zelenyi, 1989):

$$\kappa \equiv \sqrt{R_c/r_L} = |B_z/B_0|\sqrt{L/r_L} \tag{5}$$

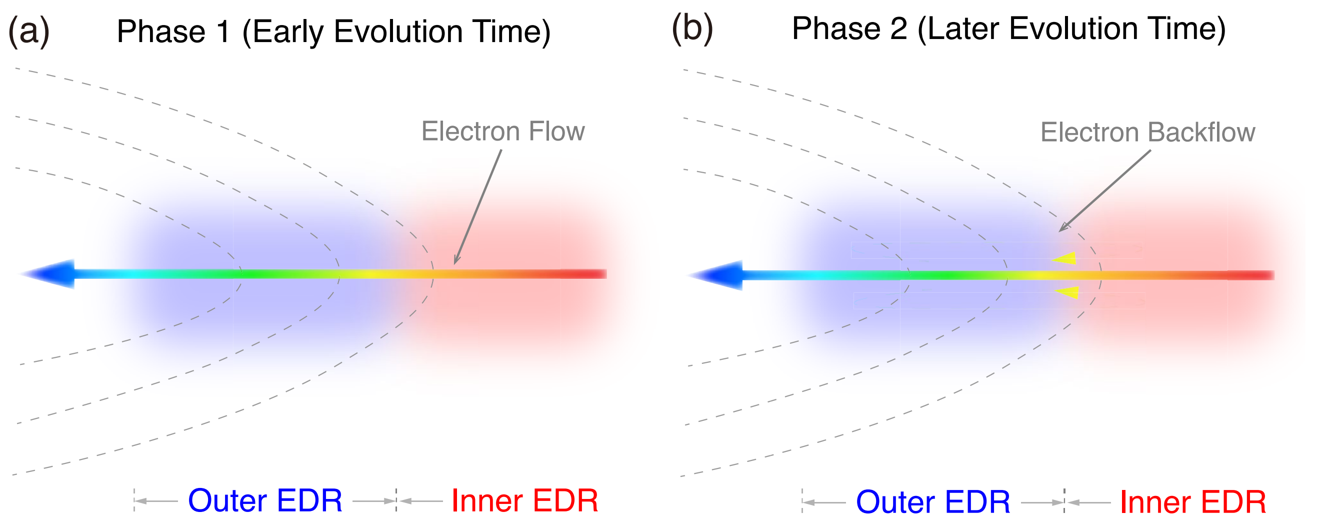


Figure 5. The red area is the inner electron diffusion region (EDR) and the blue one is the outer EDR. The colored arrows represent the electrons' flow direction, and the color of these arrows shows the energy level of the electrons. The color varies from red to blue representing the energy from high level to low level. The solid arrows stand for the major electrons passing through the outer EDR, and the dashed arrow represents the electrons with backflow motion.

where R_c is the curvature radius of the magnetic field, r_L is the electron gyro-radius, and L is the current sheet length scale. This parameter of the electron is calculated at both the entry point to the outer EDR and the reverse point, and their joint distribution is displayed in Figure 4c. Apparently, at both positions the curvature parameter is much less than 1, implying that the continuous process of outflow and backflow of these electrons refer to a non-gyro and non-regular motion (e.g., Zenitani & Nagai, 2016). Additionally, the reverse positions along the X direction of these electrons are also recorded and presented in the red histogram in Figure 4d. Most electrons turn back to the inner EDR at the position around 5.5–6.5 d_i and 8.5–9.5 d_i. The X component of the magnetic tension force sliced along the X direction is plotted at the right black axis of Figure 4d. Obviously, within the backflow spatial range mentioned above, the tension force is about ±1.5 normalized unit consistent with the result in Figure 4b.

In summary, there are two processes that manage the energy conversion between the electrons and magnetic field in the EDR, which are shown in Figure 5. At the early time of the reconnection, the electrons are accelerated in the inner EDR and decelerated in the outer EDR, then travel to the downstream exhaust (Figure 5a), which is the pattern described in the previous research (e.g., Hwang et al., 2017; Karimabadi et al., 2007; Shay et al., 2007; Xiong et al., 2022c; Zenitani et al., 2012). The electrons convey the energy from the X point to the magnetic field at the pileup region behind the RF. At the later time of the reconnection, the magnetic tension force induced by the strong out-of-plane current evolves stronger and drives part of the outflow electrons backward. These electrons return to the inner EDR and experience the acceleration and deceleration processes again, hence carrying the energy from the reconnection site to the pileup region twice (Figure 5b).

4. Conclusion and Discussion

The outer EDR has the function of decelerating the electrons and reassigning the electrons' energy to the magnetic field. The Lorentz force, especially the magnetic tension force component, is the dominant driver that hinders the electrons from moving and even turns them back to the inner EDR. During this process, the magnetic tension force is induced by the current, in which the primary carriers are electrons. These backflow electrons are accelerated and decelerated successively, bringing the energy from the reconnection site to the pileup region again. That kind of backflow motion of electrons is not related to the gyro effect by the magnetic field, and the backward points are almost located in the outer EDR. This twice deceleration motion of the electron brings another possibility to the energy conversion pattern.

From another insight, the process of those electrons ejected from the inner EDR and moving back to the inner EDR, is similar to the Marangoni effect in fluids (e.g., Scriven & Sternling, 1960). Marangoni effect in fluids refers to the mass transport with the existence of the surface-tension gradients and convection instability (e.g., Smith, 1966). In the fluid case, the temperature gradient is formed by the upstream heating on a free surface, that the cellular convection (or the Marangoni convection) is induced by the surface tension force (e.g., Pearson, 1958). The convection loop carries the responsibility of transferring the heat and mass during the absorption process of drop films forced by the surface tension (e.g., Isvoranu & Staicovici, 2004; Rongy et al., 2012). One of the critical criteria during the convection is defined by the Marangoni number (e.g., Boeck, 2005; Boeck & Thess, 1998): $M_a = \gamma q d^2/\lambda \rho \nu D$, where γ is the surface tension, q is the heat flux at the free surface, d is the layer thickness, λ is the fluid conductivity, ρ is the fluid density, ν is the viscosity, and D is the diffusivity. After quantifying the flow velocity scale, the equation above can be rewritten as (e.g., Shiratori et al., 2020): $M_a = U_0 d/D$, where U_0 represents the characteristic velocity. The large Marangoni number indicates a high relation between heat transformation and energy dissipation in fluid turbulence (e.g., Boeck & Thess, 1998). Based on the condition in the magnetohydrodynamic case, the magnetic Marangoni number can be extendedly defined as: $M_{am} = U_0 d/D_m$, where D_m is the magnetic diffusivity. The parameter D_m is defined as $D_m = (\mu_0 \sigma_0)^{-1}$, where μ_0 is the magnetic conductivity and σ_0 is the plasma conductivity. It refers to the outward expansion degree of the magnetic field ($\partial \mathbf{B}/\partial t = D_m \nabla^2 \mathbf{B}$) (e.g., Baumjohann & Treumann, 1997). In the EDR of the magnetic reconnection, specifically, U_0 is determined by electron bulk velocity, d refers to the current sheet thickness, and D_m indicate local expansion of the magnetic field spatially and temporally. The parameter $|D_m|$ is obtained by $D_m{}^2 = \Sigma_i D_{m,i}{}^2$, where $D_{m,i}$ ($i = \{x, y, z\}$) is derived from $\partial B_i/\partial t = D_{m,i} \nabla^2 B_i$, for considering three components of the magnetic field. Consequently, the magnetic Marangoni number can be obtained through the equation $M_{am} = U_0 d/|D_m|$ and shown in Figure 4d (magenta line). Obviously, M_{am} in the outer EDR is much larger than that in the inner EDR. Therefore, the electrons with particular outflow conditions in the outer EDR have the possibility to flow back to the inner EDR, indicating the magnetic Marangoni effect.

The occurrence of the magnetic Marangoni effect in the outer EDR can be inferred from the magnetic Marangoni number. From plasma perspective, it depends on the present bulk flow velocity toward the outflow direction. If the outflow velocity from the inner EDR is low, it is insufficient to form the magnetic tension force required to propel the magnetic Marangoni effect, indicating a low M_{am}. In addition, the thin current sheet is regarded as the preliminary stage of the reconnection which has limited electron outflow. It also implies the low possibility of the magnetic Marangoni effect. From the standpoint of the local field, the reconnection regimes have determined the regulation of magnetic annihilation in the inner EDR and the accumulation in the outer EDR, respectively. Therefore, the gradient of the magnetic field in the outer EDR is guaranteed to exist during reconnection, whether asymmetric or with guide field. This gradient contributes to tension force formation and reflects the magnetic diffusivity in the composition of the magnetic Marangoni number.

Data Availability Statement

The simulation data used in this study are available at the Open Science Framework at https://osf.io/mrz2x. The FGM, EDP, and FPI data of MMS are available at the MMS Science Data Center at https://lasp.colorado.edu/mms/sdc/public/data.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (42074196 and 41925018) and the National Youth Talent Support Program. S.Y.H. acknowledges the project supported by Special Fund of Hubei Luojia laboratory. The PIC simulation in this work was carried out on the supercomputing system in the Supercomputing Center of Wuhan University.

References

Baumjohann, W., & Treumann, R. A. (1997). *Basic space plasma physics*. Imperial College Press.

Boeck, T. (2005). Bénard-Marangoni convection at large Marangoni numbers: Results of numerical simulations. *Advances in Space Research*, *36*(1), 4–10. https://doi.org/10.1016/j.asr.2005.02.083

Boeck, T., & Thess, A. (1998). Turbulent Bénard-Marangoni convection: Results of two-dimensional simulations. *Physical Review Letters*, *80*(6), 1216–1219. https://doi.org/10.1103/PhysRevLett.80.1216

Büchner, J., & Zelenyi, L. M. (1989). Regular and chaotic charged particle motion in magnetotaillike field reversals: 1. Basic theory of trapped motion. *Journal of Geophysical Research*, *94*(A9), 11821–11842. https://doi.org/10.1029/JA094iA09p11821

Burch, J., Ergun, R., Cassak, P., Webster, J., Torbert, R., Giles, B., et al. (2018). Localized oscillatory energy conversion in magnetopause reconnection. *Geophysical Research Letters*, *45*(3), 1237–1245. https://doi.org/10.1002/2017GL076809

Burch, J., Moore, T., Torbert, R., & Giles, B. (2016a). Magnetospheric multiscale overview and science objectives. *Space Science Reviews*, *199*(1–4), 5–21. https://doi.org/10.1007/s11214-015-0164-9

Burch, J., Torbert, R., Phan, T., Chen, L., Moore, T., Ergun, R., et al. (2016b). Electron-scale measurements of magnetic reconnection in space. *Science*, *352*(6290), aaf2939. https://doi.org/10.1126/science.aaf2939

Ergun, R. E., Holmes, J. C., Goodrich, K. A., Wilder, F. D., Stawarz, J. E., Eriksson, S., et al. (2016). Magnetospheric multiscale observations of large amplitude, parallel, electrostatic waves associated with magnetic reconnection at the magnetopause. *Geophysical Research Letters*, *43*(11), 5626–5634. https://doi.org/10.1002/2016GL068992

Huang, S. Y., Jiang, K., Yuan, Z., Sahraoui, F., He, L., Zhou, M., et al. (2018). Observations of the electron jet generated by secondary reconnection in the terrestrial magnetotail. *The Astrophysical Journal*, *862*(2), 144. https://doi.org/10.3847/1538-4357/aacd4c

Huang, S. Y., Vaivads, A., Khotyaintsev, Y., Zhou, M., Fu, H., Retino, A., et al. (2012). Electron acceleration in the reconnection diffusion region: Cluster observations. *Geophysical Research Letters*, *39*(11), L11103. https://doi.org/10.1029/2012GL051946

Huang, S. Y., Xiong, Q., Song, L., Nan, J., Yuan, Z., Jiang, K., et al. (2021). Electron-only reconnection in an ion-scale current sheet at the magnetopause. *The Astrophysical Journal*, *922*(1), 54. https://doi.org/10.3847/1538-4357/ac2668

Huang, S. Y., Zhou, M., Yuan, Z., Deng, X., Sahraoui, F., Pang, Y., & Fu, S. (2014). Kinetic simulations of electric field structure within magnetic island during magnetic reconnection and their applications to the satellite observations. *Journal of Geophysical Research: Space Physics*, *119*(9), 7402–7412. https://doi.org/10.1002/2014JA020054

Huang, S. Y., Zhou, M., Yuan, Z. G., Fu, H. S., He, J. S., Sahraoui, F., et al. (2015). Kinetic simulations of secondary reconnection in the reconnection jet. *Journal of Geophysical Research: Space Physics*, *120*(8), 6188–6198. https://doi.org/10.1002/2014JA020969

Hwang, K., Sibeck, D., Choi, E., Chen, L., Ergun, R., Khotyaintsev, Y., et al. (2017). Magnetospheric multiscale mission observations of the outer electron diffusion region. *Geophysical Research Letters*, *44*(5), 2049–2059. https://doi.org/10.1002/2017GL072830

Isvoranu, D., & Staicovici, M. D. (2004). Marangoni convection basic mechanism explanation using two-point theory (TPT) of mass and heat transfer and the ammonia/water medium. *International Journal of Heat and Mass Transfer*, *47*(17–18), 3769–3782. https://doi.org/10.1016/j.ijheatmasstransfer.2004.04.009

Jiang, K., Huang, S. Y., Yuan, Z. G., Sahraoui, F., Deng, X. H., Yu, X. D., et al. (2019). The role of upper hybrid waves in the magnetotail reconnection electron diffusion region. *The Astrophysical Journal Letters*, *881*(2), L28. https://doi.org/10.3847/2041-8213/ab36b9

Karimabadi, H., Daughton, W., & Scudder, J. (2007). Multi-scale structure of the electron diffusion region. *Geophysical Research Letters*, *34*(13), L13104. https://doi.org/10.1029/2007GL030306

Lindqvist, P., Olsson, G., Torbert, R., King, B., Granoff, M., Rau, D., et al. (2016). The spin-plane double probe electric field instrument for MMS. *Space Science Reviews*, *199*(1–4), 137–165. https://doi.org/10.1007/s11214-014-0116-9

Lu, S., Wang, R., Lu, Q., Angelopoulos, V., Nakamura, R., Artemyev, A., et al. (2020). Magnetotail reconnection onset caused by electron kinetics with a strong external driver. *Nature Communications*, *11*(1), 5049. https://doi.org/10.1038/s41467-020-18787-w

Payne, D., Farrugia, C., Torbert, R., Germaschewski, K., Rogers, A., & Argall, M. (2021). Origin and structure of electromagnetic generator regions at the edge of the electron diffusion region. *Physics of Plasmas*, *28*(11), 112901. https://doi.org/10.1063/5.0068317

Pearson, J. R. A. (1958). On convection cells induced by surface tension. *Journal of Fluid Mechanics*, *4*(5), 489–500. https://doi.org/10.1017/S0022112058000616

Phan, T., Bale, S., Eastwood, J., Lavraud, B., Drake, J., Oieroset, M., et al. (2020). Parker solar Probe in situ observations of magnetic reconnection exhausts during encounter 1. *The Astrophysical Journal - Supplement Series*, *246*(2), 34. https://doi.org/10.3847/1538-4365/ab55ee

Phan, T., Eastwood, J., Shay, M., Drake, J., Sonnerup, B., Fujimoto, M., et al. (2018). Electron magnetic reconnection without ion coupling in Earth's turbulent magnetosheath. *Nature*, *557*(7704), 202–206. https://doi.org/10.1038/s41586-018-0091-5

Pollock, C., Moore, T., Jacques, A., Burch, J., Gliese, U., Saito, Y., et al. (2016). Fast plasma investigation for magnetospheric multiscale. *Space Science Reviews*, *199*(1–4), 331–406. https://doi.org/10.1007/s11214-016-0245-4

Rongy, L., Assemat, P., & De Wit, A. (2012). Marangoni-driven convection around exothermic autocatalytic chemical fronts in free-surface solution layers. *Chaos*, *22*(3), 037106. https://doi.org/10.1063/1.4747711

Russell, C., Anderson, B., Baumjohann, W., Bromund, K., Dearborn, D., Fischer, D., et al. (2016). The magnetospheric multiscale magnetometers. *Space Science Reviews*, *199*(1–4), 189–256. https://doi.org/10.1007/s11214-014-0057-3

Scriven, L., & Sternling, C. (1960). The Marangoni effects. *Nature*, *187*(4733), 186–188. https://doi.org/10.1038/187186a0

Shay, M., Drake, J., & Swisdak, M. (2007). Two-scale structure of the electron dissipation region during collisionless magnetic reconnection. *Physical Review Letters*, *99*(15), 155002. https://doi.org/10.1103/physrevlett.99.155002

Shiratori, S., Kato, D., Sugasawa, K., Nagano, H., & Shimano, K. (2020). Spatio-temporal thickness variation and transient Marangoni number in striations during spin coating. *International Journal of Heat and Mass Transfer*, *154*, 119678. https://doi.org/10.1016/j.ijheatmasstransfer.2020.119678

Shuster, J. R., Chen, L.-J., Hesse, M., Argall, M. R., Daughton, W., Torbert, R. B., & Bessho, N. (2015). Spatiotemporal evolution of electron characteristics in the electron diffusion region of magnetic reconnection: Implications for acceleration and heating. *Geophysical Research Letters*, *42*(8), 2586–2593. https://doi.org/10.1002/2015GL063601

Singh, C. B., Dal Pino, E. D. G., & Kadowaki, L. H. S. (2015). On the role of fast magnetic reconnection in accreting black hole sources. *The Astrophysical Journal Letters*, *799*(2), L20. https://doi.org/10.1088/2041-8205/799/2/L20

Smith, K. A. (1966). On convective instability induced by surface-tension gradients. *Journal of Fluid Mechanics*, *24*, 2–414. https://doi.org/10.1017/S0022112066000727

Torbert, R., Burch, J., Phan, T., Hesse, M., Argall, M., Shuster, J., et al. (2018). Electron-scale dynamics of the diffusion region during symmetric magnetic reconnection in space. *Science*, *362*(6421), 1391–1395. https://doi.org/10.1126/science.aat2998

Xiong, Q. Y., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Wei, Y. Y., et al. (2022a). Statistic properties of electron energy enhancement during the inner electron diffusion region crossing. *Journal of Geophysical Research: Space Physics*, *127*(10), e2022JA030760. https://doi.org/10.1029/2022JA030760

Xiong, Q. Y., Huang, S. Y., Zhou, M., Yuan, Z. G., Deng, X. H., Jiang, K., et al. (2022b). Distribution of negative J·E′ in the inflow edge of the inner electron diffusion region during tail magnetic reconnection: Simulations vs. observations. *Geophysical Research Letters*, *49*(11), e2022GL098445. https://doi.org/10.1029/2022GL098445

Xiong, Q. Y., Huang, S. Y., Zhou, M., Yuan, Z. G., Deng, X. H., Jiang, K., et al. (2022c). Formation of negative J·E′ in the outer electron diffusion region during magnetic reconnection. *Journal of Geophysical Research: Space Physics*, *127*(2), e2022JA030264. https://doi.org/10.1029/2022JA030264

Zenitani, S., & Nagai, T. (2016). Particle dynamics in the electron current layer in collisionless magnetic reconnection. *Physics of Plasmas*, *23*(10), 102102. https://doi.org/10.1063/1.4963008

Zenitani, S., Shinohara, I., & Nagai, T. (2012). Evidence for the dissipation region in magnetotail reconnection. *Geophysical Research Letters*, *39*(11), L11102. https://doi.org/10.1029/2012GL051938

Zhou, M., Deng, X., & Huang, S. (2012). Electric field structure inside the secondary island in the reconnection diffusion region. *Physics of Plasmas*, *19*(4), 042902. https://doi.org/10.1063/1.3700194

Geophysical Research Letters

RESEARCH LETTER
10.1029/2023GL106581

Key Points:

- A novel crater structure is first verified behind the Reconnection front (RF) by both Magnetospheric Multiscale observations and particle-in-cell simulations
- The formation of the crater structure appears to be associated with the high-speed electron jets from inner electron diffusion region
- A possible scenario that electron outflow constantly strikes the RF and then causes the formation of the crater structure

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

S. Y. Huang,
shiyonghuang@whu.edu.cn

Citation:

Huang, S. Y., Xiong, Q. Y., Yuan, Z. G., Jiang, K., Yu, L., Xu, S. B., & Lin, R. T. (2024). Crater structure behind reconnection front. *Geophysical Research Letters*, *51*, e2023GL106581. https://doi.org/10.1029/2023GL106581

Received 26 SEP 2023
Accepted 25 JAN 2024

Crater Structure Behind Reconnection Front

S. Y. Huang[1], Q. Y. Xiong[1], Z. G. Yuan[1], K. Jiang[1], L. Yu[1], S. B. Xu[1], and R. T. Lin[1]

[1]School of Electronic Information, Hubei Luojia Laboratory, Wuhan University, Wuhan, China

Abstract Magnetic reconnection is the physical process that converts the energy from the fields to the plasmas in space, astrophysical and laboratory plasmas. The Reconnection front (RF) is the structure generated in the reconnection outflow region and participates in the energy release budget. Here, we first report a novel crater structure of magnetic field behind the RF, which is well supported by both the in-situ observations from the Magnetospheric Multiscale mission and kinetic particle-in-cell simulations. The theoretical explanations from the simulations suggests that the formation of the crater structure is possibly due to that high-speed outflow electron jet from inner electron diffusion region constantly strikes the RF. From another perspective, the crater structure is the continuous impact of the electron jet. Our results can establish a new understanding of the RF and energy conversion during magnetic reconnection.

Plain Language Summary Magnetic reconnection is a natural process in space environments, astrophysical settings, and laboratories, where energy from magnetic fields is transformed into the energy of various particles. One crucial structure in this process is called the reconnection front (RF), which plays a big role in how energy is released. In our study, we have discovered something interesting: a unique crater-like structure behind the RF. We found evidence for this in observations from the Magnetospheric Multiscale mission and computer simulations that study the behavior of particles in magnetic reconnection. Our simulations suggest that this crater shape happens because electrons have the high-speed outflow and form current jets. It is like the electrons poured out from the inner electron diffusion region, hitting a speed bump. Another way to think about it is that this crater is formed by the continuous impact of fast-outflowing electron jets. Understanding this crater structure helps us better grasp how the RF works and how energy changes during magnetic reconnection. Our research finds and tries to explain a new piece of the puzzle in understanding the mysteries of space and plasmas in the magnetic reconnection process.

1. Introduction

Magnetic Reconnection, one of the crucial energy release processes, is abundant in space, astrophysical and laboratory plasmas. In the diffusion region, which is regarded as the core area of the reconnection, the magnetic field topology is changed and its energy is converted into the plasmas (e.g., Burch, Torbert, et al., 2016; Huang, Vaivads, et al., 2012, Huang, Xiong, et al., 2021; Torbert et al., 2018; Zenitani et al., 2011). The different behaviors of ions and electrons due to the different masses lead to the distinction of two-scale diffusion regions: ion diffusion region and electron diffusion region (EDR) (e.g., Pritchett, 2001). Particularly, the EDR is also characterized by two parts: the inner EDR, where the electrons receive the energy from the breaking magnetic field lines, and the outer EDR, where the electrons' energy is transferred to the magnetic field (e.g., Hwang et al., 2017; Karimabadi et al., 2007; Shay et al., 2007; Xiong et al., 2022a).

Reconnection front (RF), which is also called dipolarization front in the Earth's magnetotail, is the structure produced by the reconnection and propagating toward the outflow direction (e.g., Fu et al., 2012, 2013a, 2013b; Huang, Zhou, et al., 2012, 2015b, 2019; Jiang et al., 2020; Lin et al., 2023; Nakamura et al., 2009). In traditional cognition, the RF is formed through the continuous pile-up of the newly reconnected magnetic field lines, predominantly expressed by the quantity B_z (or B_N in LMN coordinates). Therefore, B_z gradually reaches the peak level from the center X point to the outflow exhaust (e.g., Sitnov & Swisdak, 2011). The RF carries the responsibility of transferring the energy from the reconnection point downstream during the propagation of RF (e.g., Fu et al., 2017; Shu et al., 2021). Furthermore, the electrons can be accelerated at and behind RF and form various kinds of distributions (e.g., Barbhuiya et al., 2022; Fu et al., 2011, 2012, 2020a, 2020b, 2022; Huang, Lu, et al., 2021; Wei et al., 2022, 2023; Xu et al., 2018, 2022; Yu et al., 2023; Zhao et al., 2019).

However, recent observations and simulations show that RF can have more complicated structures than ever expected. The rippled electron-scale structures can be generated at RF due to the lower hybrid drift instability (e.g., Bai et al., 2022; Pan et al., 2018), the interchange instability accompanied by multiple flow channels (e.g., Fu et al., 2019; Yu et al., 2022), and the electron-scale plateau of the magnetic field caused by electron vortex is observed at RF (e.g., Jiang et al., 2020). The crater-shaped flux ropes on the RF can be produced by Kelvin-Helmholtz waves (e.g., Farrugia et al., 2011; Hwang et al., 2020). Moreover, we can recognize a dip region behind the peak of RF so that the magnetic field lines are dented (e.g., Egedal et al., 2019; Song et al., 2019). However, the cause of this dented structure behind the RF lacks detailed investigation, since this dented region potentially connects the outer EDR and RF spatially and might be involved in the energy conversion budget during the process of energy propagation from the X-line to the downstream exhaust. In the present study, we provide direct observational evidence of this dented area (called crater structure in the rest of the text) behind the RF, which is captured by the Magnetospheric Multiscale (MMS) mission. We also perform the 2.5-D full kinetic particle-in-cell (PIC) simulations to verify this structure compared with the observation results and reveal how it is formed during the reconnection evolution.

2. Instruments and Numerical Methods

We take full advantage of MMS data for its high resolution and precision (Burch, Moore, et al., 2016). The magnetic field data with the sampling of 128 Hz is from the Fluxgate Magnetometer (FGM) (Russell et al., 2016). The electric field data with the sampling of 8,196 Hz is from the Electric Double Probe (EDP) (Ergun et al., 2016; Lindqvist et al., 2016), and the particle's moments data (150 ms for ions, 30 ms for electrons) are from the Fast Plasma Investigation (FPI) (Pollock et al., 2016).

We also carry out 2.5-D PIC simulations where the code has been used in previous studies (Huang et al., 2014, 2015a; Xiong et al., 2022a, 2022b, 2022c, 2023; Zhou et al., 2012, 2014). The mass ratio of ion and electron is $m_i/m_e = 100$ for the limited computational resource, the initial temperature ratio is $T_i/T_e = 5$, and the ratio between the electron frequency and the gyro-frequency ratio is $\omega_{\mathrm{pe}}/\omega_{\mathrm{ce}} = 3$. The simulation is performed in the domain with the grids of $1{,}600 \times 2{,}400$ ($32d_i \times 48d_i$, d_i is the ion inertial length), and therefore d_e equals 5 which can well resolve the electron dynamics. Periodic boundary conditions are applied along both X and Z directions. The spatial grid is normalized by d_i and the time is normalized by $\Omega_{\mathrm{ci}}{}^{-1}$. The magnetic fields are normalized by background magnetic field (B_0), the current is normalized by $q_i n_0 V_A$, the energy conversion is normalized by $q_i n_0 B_0 V_A{}^2$ (q_i is the ion charge, n_0 is the background density, V_A is Alfven speed), and the ion and electron velocity are normalized by V_A.

3. Results

The MMS observation detected the crater structure behind the RF in a reconnection event occurring at the dusk flank magnetopause on 5 October 2017 (Øieroset et al., 2021). Figures 1a–1k display the physical quantities at LMN coordinates transformed from the GSE coordinate system by the minimum variance analysis method (Sonnerup & Cahill, 1967). The typical reconnection signatures can be easily recognized: bipolar Hall magnetic field with respect to large guide field (Figure 1b), high-speed outflow (Figures 1g and 1h), strong out-of-plane current density (Figure 1j) predominately carried by the electrons (Figure 1h), and non-zero energy conversion (Figure 1k) during the current sheet crossing (B_L reverse, Figure 1a). Other detailed information, such as the identification of diffusion regions has been presented and discussed by Øieroset et al. (2021). We mainly focus on the crater structure behind the RF which is highlighted by the colored blocks in Figures 1c–1f. All four spacecraft cross the crater structure successively, capturing the local dips marked by four vertical dashed lines. The dip positions exactly correspond to the current sheet center (Figures 1a–1f) for all crossings of four spacecraft. Both ions and electrons are approximately demagnetized within this structure (not shown here). Noticeably, the peak centers of V_{eL} and V_{eM} of MMS1 are not well aligned with the dashed lines due to the deflection and distortion effect under the guide field (e.g., Goldman et al., 2011; Le et al., 2013; Tharp et al., 2013).

The approximate sketch of MMS traveling is presented in Figure 1w to help better understand. The sequence of crossing moments and positions of four spacecrafts are ordered based on the spatial evolution of the reconnection (Øieroset et al., 2021). The MMS4 is the closest to the diffusion region and even crosses the outer EDR (blue line in Figure 1k), thus observing negative $\mathbf{J} \cdot \mathbf{E}'$ (e.g., Hwang et al., 2017; Xiong et al., 2022a). As the crossing point along the L direction is further away from the EDR, the energy conversion between the fields and the particles

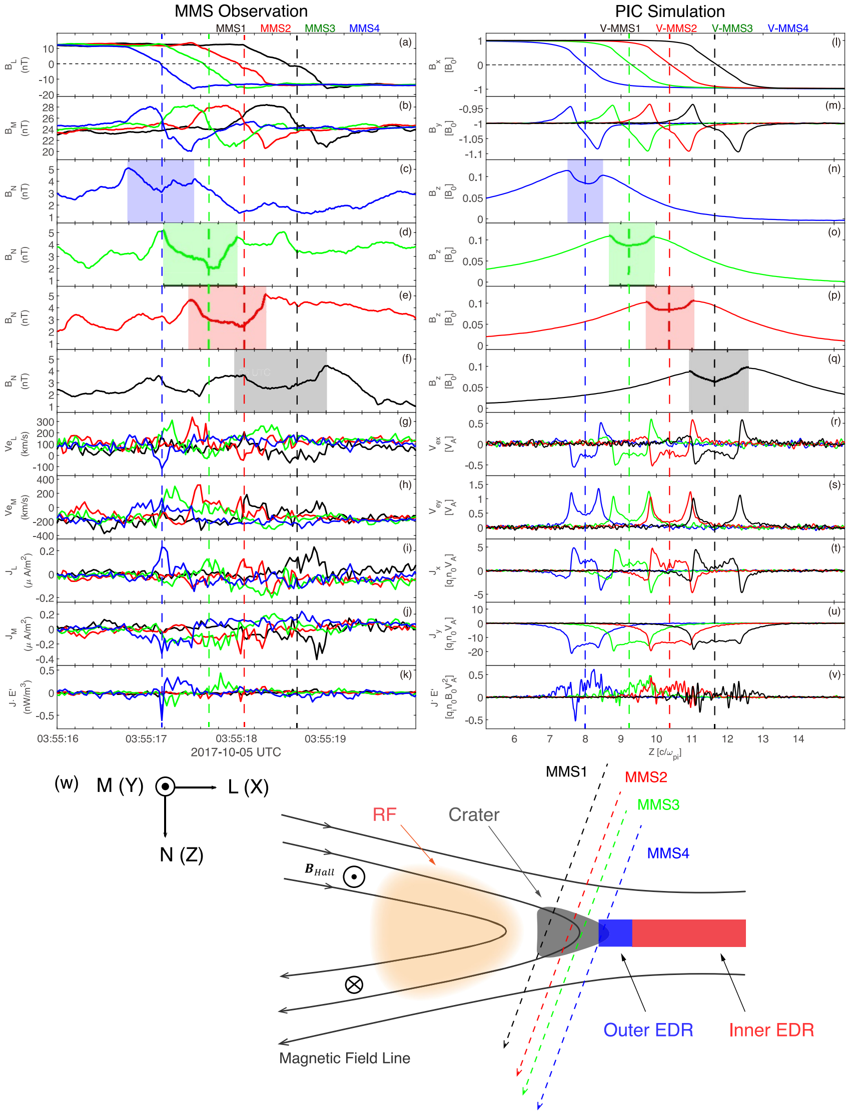


Figure 1. Comparison of magnetospheric multiscale (MMS) observation and particle-in-cell simulation. Four colors are used to distinguish the four MMS satellites or virtual satellites. (a) and (l) L and X component of the magnetic field. (b) and (m) M and Y component of the magnetic field. (c–f) and (n–q) N and Z component of the magnetic field. (g) and (r) Electron outflow velocity V_{ex}. (h) and (s) Electron out-of-plane velocity V_{ey}. (i) and (t) L and X component of the current density. (j) and (u) Out-of-plane current density. (k) and (v) Energy conversion term $\mathbf{J}\cdot\mathbf{E}'$. The vertical dashed lines mark the approximate position of the current sheet center. (w) reconnection front (RF) crossing pattern of the satellites. The dashed lines are the MMS trajectories. The solid black lines are the magnetic field lines. The red and blue areas are the inner and outer electron diffusion region, respectively. The gray area is the crater structure. The shallow orange area is the RF.

gets lower. The gray area is the crater structure, and its width expands with further outflow, verified by the width of four colored blocks in Figures 1c–1f. The crater structure has an intersection with the outer EDR as MMS4 detects both the crater structure (Figure 1c) and the outer EDR (Figure 1k). The RF is located ahead of the crater structure and forms the regime in the crater structure that connects the outer EDR with the RF.

PIC simulations are performed to make a comparison with the observations. We are intent on setting the guide field as $B_g = 1B_0$ in this run case to avoid obtaining a highly disturbed reconnection structure, though it is about $B_g = 2B_0$ in the observation. The ***XYZ*** coordinates are used in the simulations, and they correspond to the directions of the ***LMN*** coordinates of the observations, respectively. Four virtual satellites' trajectories are chosen just following the crossing pattern in Figure 1w to get the 1D slices from the 2D simulation results at the time $t\Omega_{\mathrm{ci}} = 28$. As is shown in Figures 1l–1v, the simulation results demonstrate the same conclusions with the left part

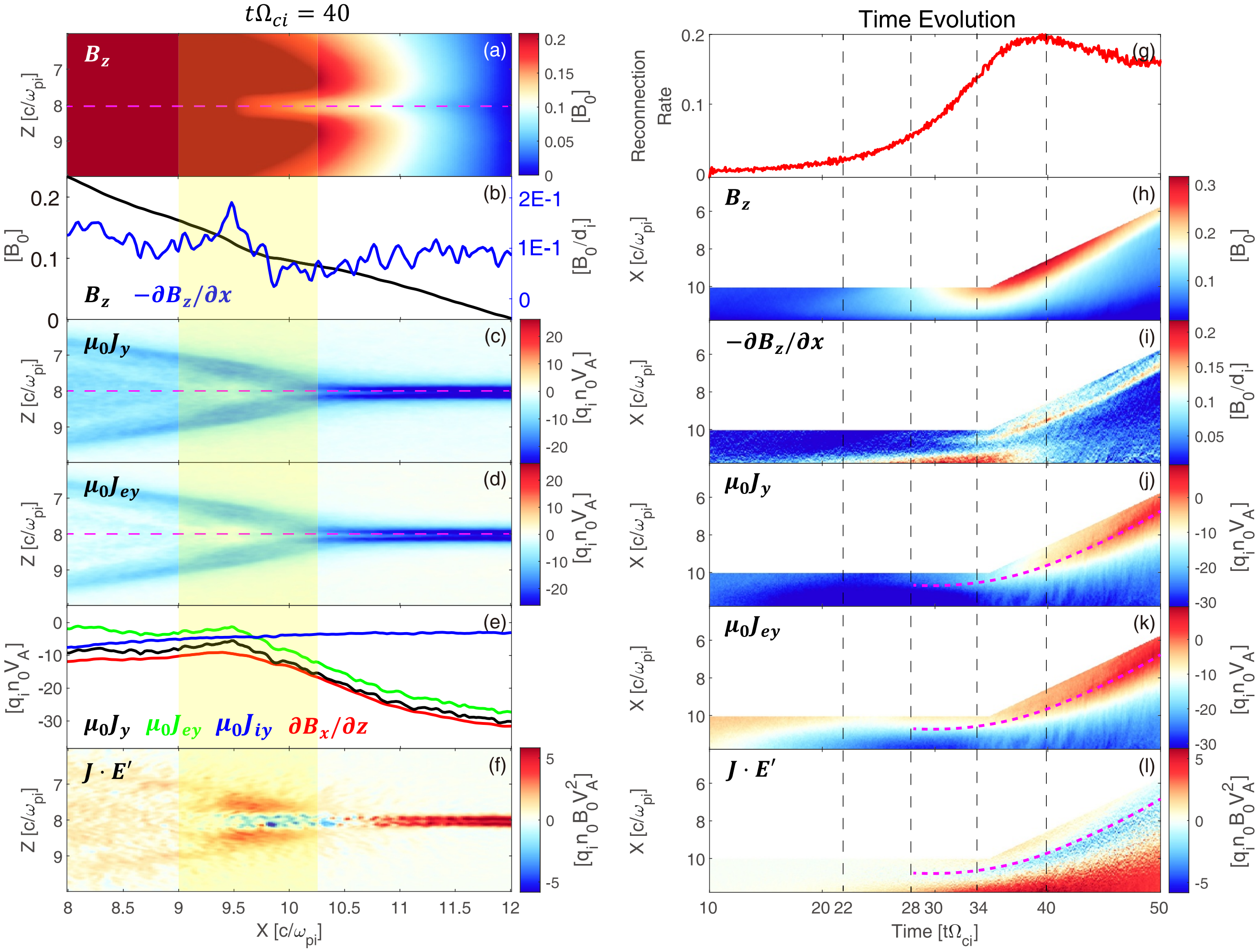


Figure 2. Simulation results of crater structure behind reconnection front. (a) 2D distribution of B_z. (b) 1D cut of B_z (black line) and the term $-\partial B_z/\partial x$ (blue line). (c) 2D distribution of $\mu_0 J_y$. (d) 2D distribution of $\mu_0 J_{ey}$. (e) 1D cut of the term $\mu_0 J_y$ (black line), $\mu_0 J_{ey}$ (green line), $\mu_0 J_{iy}$ (blue line), and $\partial B_x/\partial z$ (red line). (f) 2D distribution of energy conversion $\mathbf{J}\cdot\mathbf{E}'$. The magenta dashed lines in panels (a), (c), and (d) are the 1D cut position. The shallow yellow region highlights the crater structure area along X direction. (g) Reconnection rate normalized by $B_0 V_A$. (h–l) Time evolution of the 1D cuts of the term B_z, $-\partial B_z/\partial x$, $\mu_0 J_y$, $\mu_0 J_{ey}$, and $\mathbf{J}\cdot\mathbf{E}'$. The magenta dashed lines in panels (j–l) mark the largest gradient of crater structure long X direction. The four vertical black dashed lines shows the moments picked in Figure 3.

of Figure 1. The dips of B_z component are clearly captured, and the change in their widths also shows the spatial evolution of the RF from the EDR to the outflow downstream (Figures 1n–1q).

Next, we attempt to reveal how this crater structure forms behind the RF. The crater structure is guaranteed to exist under different levels of guide field from the results of observation ($B_g = 2B_0$) and simulation ($B_g = 1B_0$). On the one hand, since the existence of the guide field makes the reconnection regime more complex and difficult to investigate, and the structure of the reconnection is more regulated under zero guide field and symmetric case (e.g., Goldman et al., 2011; Le et al., 2013; Song et al., 2019). On the other hand, in the observations we could roughly capture symmetric B_M on two sides of the current sheet (Figure 1b), especially from MMS4 and MMS3. Besides, the peaks of electron outflow can be detected possibly due to the asymmetry of density and temperature at two sides of the current sheet center (e.g., Montag et al., 2020), and currents are exactly located in the current sheet center (Figures 1g and 1i, especially from MMS4 to MMS2). These signatures indicate that the reconnection in this event could approximately be regarded as a standard symmetric reconnection as the condition with zero guide field when considering the region close to EDR. Therefore, it makes sense to explain physically through the run case with zero guide field ($B_g = 0$). The left part of Figures 2a–2f show the parameters associated with the formation of crater structure at the time $t\Omega_{\mathrm{ci}} = 40$, when the reconnection rate reaches the highest (Figure 2g). The shallow yellow region marks the crater region in the X direction. There is an evident dented trend of B_z around the current sheet center ($Z = 8d_i$) (Figure 2a), which makes it a dip detection when crossing as the pattern in Figure 1w. The B_z changes from flattening to steep from $X = 10.25d_i$ to $X = 9d_i$ in the yellow shadow region (black line in Figure 2b), resulting in the sharp increase of the term $-\partial B_z/\partial x$ (blue line in Figure 2b). Meanwhile, the electrons are the dominant carrier of the out-of-plane current in EDR (Figures 2c–2e), and the ions begin to take effect at the region around $X < 9.5d_i$. Considering the fact that the outflow jets are dominated by the electron within the diffusion region and the crater structure is well resolved around $X < 9.5d_i$ which is basically within the EDR (Figure 2a), we proceed to only consider the contribution from electrons. Combined with the Y component of the approximated Ampere's law, these terms are connected as the equation $-\partial B_z/\partial x \sim \mu_0 J_{ey} - \partial B_x/\partial z$. Noticeably, the position of the crater structure connects with the outer EDR (Figure 2f), where $\mathbf{J}\cdot\mathbf{E}'$ is negative and the electrons experience deceleration (e.g., Xiong et al., 2022a). As a result, the current density gradually decreases at the crater structure from right to left and cannot match the gradient of B_x along Z direction, thus forming a peak of $-\partial B_z/\partial x$ and the crater behind the RF.

Following the time sequence, we also identify the evolution of the crater structure during the reconnection. The right part of Figure 2 illustrates the time evolution of the 1D slices of physical quantities in the left part. The slice region is $Z = 8d_i$ and $X = 4 \sim 12d_i$. The reconnection rate is shown in Figure 2g for a better comparison of different phases. The area farther away from the RF peak is set as blank in Figures 2h–2l to avoid interference from other fine structures behind the RF. At the time $t\Omega_{\mathrm{ci}} = 28$, the crater structure begins to form (Figure 2h) and the increase of the term $-\partial B_z/\partial x$ also shows the appearance (Figure 2i). When the reconnection evolves to a faster phase ($t\Omega_{\mathrm{ci}} = 34$), it is more obvious to detect the peak of the term $-\partial B_z/\partial x$ (Figure 2i) and the negative $\mathbf{J}\cdot\mathbf{E}'$ in the outer EDR (Figure 2l). At the stage of the peak reconnection rate ($t\Omega_{\mathrm{ci}} = 40$), the sharp increase of the term $-\partial B_z/\partial x$ also reaches the most significant level. The consistent variation paces of these variables with time demonstrate the formation process and spatial evolution of the crater structure (Figure 2i, magenta lines in Figures 2j–2l). Besides, the RF and crater structure propagate downstream no further than $X = 6d_i$ during the evolution. The particle outflows at two sides of the reconnection downstream have yet to converge. And the width of the crater structure along the Z direction is no more than $4d_i$ indicating that it maintains a distance of $6d_i$ distance to the upper Z boundary. These suggest that the reconnection is still a localized process and the formation process of this structure could not be affected by the boundary condition until the time of $t\Omega_{\mathrm{ci}} = 40$ we investigate.

Now that the formation of the crater structure is associated with the out-of-plane electron jets (Figures 2c–2e), we consider the connection between electron outflow, crater structure, and RF. Figure 3 shows the evolution of B_z (colored counter lines) and the electron flow (black arrows) at one side of the reconnection outflow. The vertical dashed lines in the right part of Figure 2 mark the four selected moments. Furthermore, the 1D sliced data at the position $Z = 8d_i$ are also presented in Figures 3e–3h. The inner EDR width expands toward outflow during the early phase, and the outer EDR gradually forms (Figures 3a and 3b). Meanwhile, the magnitude of B_z increases at two sides of the current sheet center initially; however, it maintains a low level at the current sheet center (black and red lines in Figure 3e). Moreover, B_z contour lines gradually dented toward the outflow direction (Figure 3b). These suggest that the peak of RF has not formed yet at the early stage of the reconnection, but the crater structure has shown its prototype shape (especially at $t\Omega_{\mathrm{ci}} = 28$). As for the later time, the RF's pileup process is accumulated and rises to a high level (red color contour lines in Figures 3c and 3d, peaks of green and blue lines in Figure 3e) and moves further away from the EDR. The crater structure becomes more apparent behind the peak of RF, which is indicated by the local negative peaks of B_z gradient (marked by shallow green and blue bars in Figures 3e–3h). Both inner and outer EDR get wider along the X direction, and B_z turns out to be thicker (along the Z direction) and more dented from the reconnection point to the outflow downstream (Figure 3d).

Interestingly, the outer EDR can roughly fit into the partial crater region behind the RF (especially in Figures 3c and 3d). Combined with the previous study (Xiong et al., 2022a), the relationship between the crater structure and the outer EDR can be inferred from the electron motion, which is summarized in Figures 3i and 3j. The high-speed outflow electron jets are poured out from the inner EDR and then would be decelerated by the outer EDR. On account of these processes, the electron energy is converted into the magnetic field, continuously propelling the RF formation. On the other hand, the speed of electron outflow gradually increases during the reconnection evolution through the inner EDR acceleration (Figures 3g and 3h), where the reconnection electric field plays the dominant role in this process (e.g., Xiong et al., 2022c). At the later phase (Figure 3j), these high-speed electron jets constantly strike the RF in the outflow region, where they flush the RF anterior section to be flatted and extended. In that case, the flux pileup region is gradually compressed, manifesting the crater region proposed in this study. The RF peak accumulates and propagates downstream pushed by the outflow electron jets.

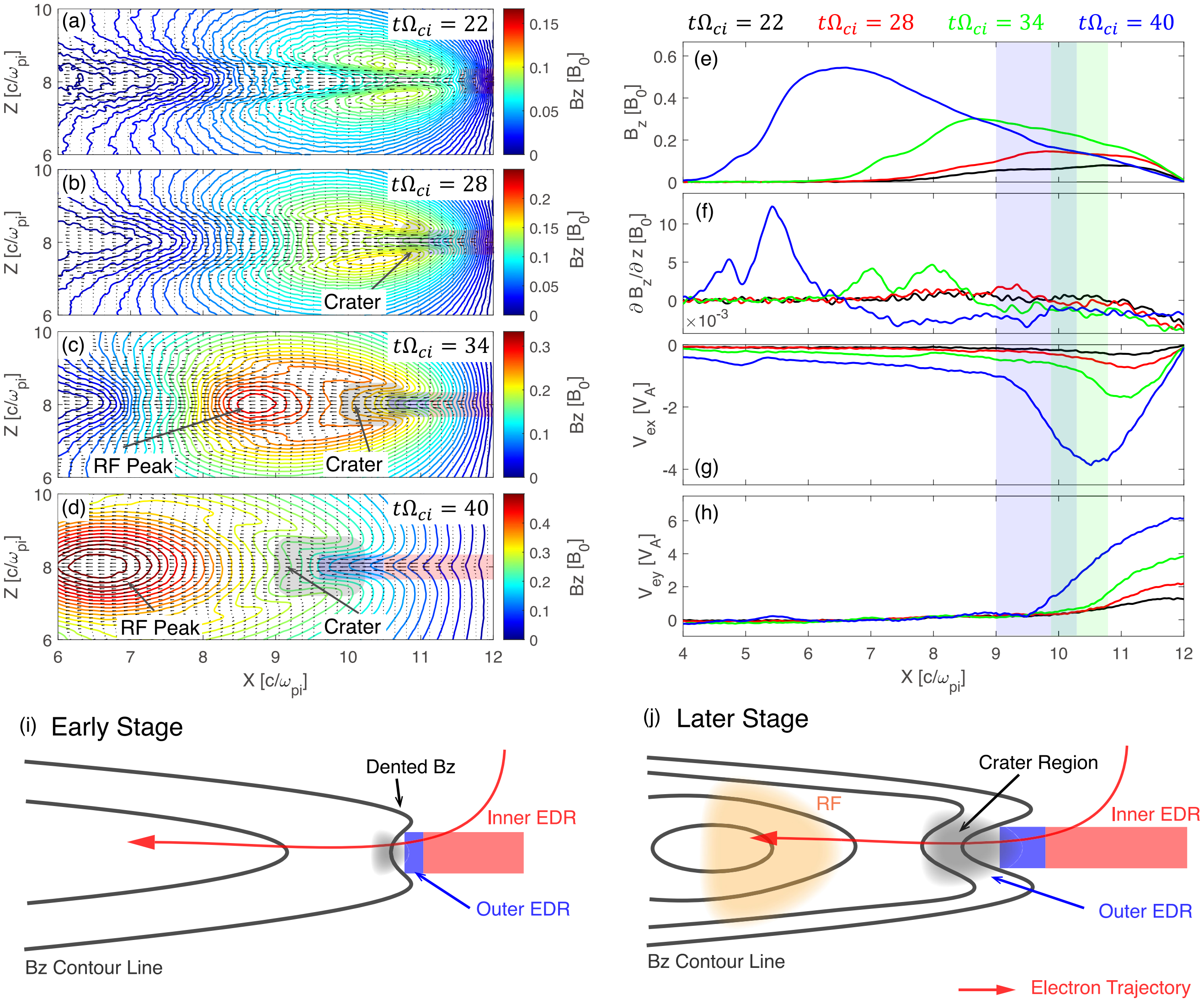


Figure 3. (a–d) Time evolution of B_z and the electron diffusion region (EDR) in 2D distribution at four moments: $t\Omega_{ci} = 22, 28, 34$, and 40. The colored contour lines are B_z value. The shallow red area is the inner EDR, and the shallow blue one is the outer EDR. The black arrows are the electron flows. (e–h) 1D slices of parameters B_z, $\partial B_z/\partial x$, V_{ex}, and V_{ey} at the position $Z = 8d_i$ Four different color lines represent the results of four moments. The shallow green and blue areas in panels (e) and (f) are the approximate positions crater structure. (i–j) Sketches of the crater structure evolution. The black solid lines are the B_z contour lines, and the red lines are the electrons trajectories. The red, blue, and gray areas are the inner EDR, outer EDR, and crater structure, respectively.

4. Conclusions and Discussions

With the support of the MMS observations and PIC simulations, we verified a novel crater structure behind the RF for the first time. Four MMS spacecrafts cross this structure successively and capture its spatial characteristics. The PIC simulations suggest that the crater structure generates behind the RF and probably connects to the outer EDR. Theoretically, the gradient of B_x along Z direction ($\partial B_x/\partial z$) cannot match the out-of-plane electron current (J_y) in the outflow region, thus forming the crater structure. The electrons have a direct contribution to this structure during the reconnection evolution. After accelerating in the inner EDR, the high-speed electron jets could be restrained in the outer EDR for its deceleration effect (e.g., Hwang et al., 2017; Xiong et al., 2022a; Zenitani et al., 2011). Moreover, in the outer EDR, the electrons convert the energy to the magnetic field, therefore building up the RF propagating toward outflow. Meanwhile, the high-speed electron outflow impacts the RF continuously, and the Z component of the magnetic field is dented. These two processes impel the formation of the crater structure behind the RF. It is also inferred that this structure could be a crucial area connecting the EDR and pileup region in both spatial and energy perspectives. The crater structure participates in the energy propagation from the X point to the downstream during the reconnection.

The observational results manifest the existence of the guide field which can restrain the reconnection rate and outflow jets. Considering the effect from the guide field in the simulation could be necessary perspective. The crater structure behind RF also can be found in the run case of $B_g = 1.0B_0$ at the time $t\Omega_{ci} = 40$ which is presented as the Figure S1 in Supporting Information S1. It is indicated by the dip area of B_z (Figure S1a in Supporting Information S1) and the sharp rise of B_z gradient (Figure S1b in Supporting Information S1). The guide field makes the jets (predominantly contributed by the electrons) deflected to one side of the current sheet (Figure S1c–S1d in Supporting Information S1). The energy conversion is more disturbed and it is difficult to recognize the outer EDR in the later evolution phase (Figure S1g in Supporting Information S1). In that case, it is barely to confirm the existence of the electron deceleration motion in the outer EDR. Perhaps the outer EDR in the guide field simulation would be restored by adding the temperature and density asymmetry just as the observation event. Still, the impact from the electron's constant striking to the pileup region can be followed through the arrows marked in Figure S1a and S1d in Supporting Information S1. This oblique striking could similarly make the magnetic field dented which is more complicated than the zero guide field condition. Without doubt, the crater structure is determined to form by the high-speed electron outflow jets even affected by the guide field.

The formation and the energy conversion at the RF are affected by various factors and the different phases. The curvature force can accelerate the outward movement of RF, and the maximum of the B_z is increased by the Poynting flux (e.g., Song et al., 2020). Also, the energy released by the reconnection at the X-line gradually propagates to the RF as the evolution goes on (e.g., Shu et al., 2021). Noticeably, the ions carry the majority of the current beyond the EDR but their current amplitude is relatively low (Figure 2e). This feature indicates a possibility that the ions are responsible for the downstream propagation of the crater structure. Combined with the electron's motions in the EDR, it could be inferred that there are two processes, following the spatial sequence, responsible for the evolution of the crater structure: (a) initially near the EDR at the electron scale, the electrons strike the pileup region to form the crater shape of the structure; (b) then at the outflow region beyond the EDR, the ions with low outflow speed drive the crater structure to the downstream at ion scale. Our results provide novel ideas about how the RF forms based on the new signature, that is, the crater structure. It potentially suggests that the energy lost by the electrons in the outer EDR could be temporarily cached in the crater structure, for which the crater region is spatially connected to the outer EDR. Then, the stored energy is delivered to the RF through another approach, which should be further excavated. Under this circumstance, the interaction between the particles from EDR and RF is more complicated than expected. A new scheme of the RF formation and energy budget is proposed based on the present simulation results, which broadens the understanding of the RF and magnetic reconnection with a new perspective.

Data Availability Statement

The simulation data used in this study are available at the open science framework (Huang et al., 2023). The FGM (Russell et al., 2016), EDP (Ergun et al., 2016; Lindqvist et al., 2016), and FPI (Pollock et al., 2016) data of the MMS are available publicly at the MMS Science Data Center https://lasp.colorado.edu/mms/sdc/public/data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (42074196, 41925018). S.Y.H. acknowledges the project supported by the Special Fund of Hubei Luojia laboratory. The PIC simulation in this work was carried out on the supercomputing system in the Supercomputing Center of Wuhan University.

References

Bai, K., Yu, Y., Huang, H., Tian, X., & Cao, J. (2022). Electron surfing acceleration at rippled reconnection fronts. *The Astrophysical Journal*, *931*(1), 70. https://doi.org/10.3847/1538-4357/ac67f1

Barbhuiya, M. H., Cassak, P. A., Shay, M. A., Roytershteyn, V., Swisdak, M., Caspi, A., et al. (2022). Scaling of electron heating by magnetization during reconnection and applications to dipolarization fronts and super-hot solar flares. *Journal of Geophysical Research*, *127*(8), e2022JA030610. https://doi.org/10.1029/2022JA030610

Burch, J., Moore, T., Torbert, R., & Giles, B. (2016). Magnetospheric multiscale overview and science objectives. *Space Science Reviews*, *199*(1–4), 5–21. https://doi.org/10.1007/s11214-015-0164-9

Burch, J., Torbert, R., Phan, T., Chen, L., Moore, T., Ergun, R., et al. (2016). Electron-scale measurements of magnetic reconnection in space. *Science*, *352*(6290). https://doi.org/10.1126/science.aaf2939

Egedal, J., Ng, J., Le, A., Daughton, W., Wetherton, B., Dorelli, J., et al. (2019). Pressure tensor elements breaking the frozen-in law during reconnection in Earth's magnetotail. *Physical Review Letters*, *123*(22), 225101. https://doi.org/10.1103/PhysRevLett.123.225101

Ergun, R., Tucker, S., Westfall, J., Goodrich, K., Malaspina, D., Summers, D., et al. (2016). The axial double probe and fields signal processing for the MMS mission. *Space Science Reviews*, *199*(1–4), 167–188. https://doi.org/10.1007/s11214-014-0115-x

Farrugia, C. J., Chen, L.-J., Torbert, R. B., Southwood, D. J., Cowley, S. W. H., Vrublevskis, A., et al. (2011). "Crater" flux transfer events: Highroad to the X line? *Journal of Geophysical Research*, *116*(A2), A02204. https://doi.org/10.1029/2010JA015495

Fu, H. S., Cao, J. B., Khotyaintsev, Y. V., Sitnov, M. I., Runov, A., Fu, S. Y., et al. (2013a). Dipolarization fronts as a consequence of transient reconnection: In situ evidence. *Geophysical Research Letters*, *40*(23), 6023–6027. https://doi.org/10.1002/2013GL058620

Fu, H. S., Grigorenko, E. E., Gabrielse, C., Liu, C., Lu, S., Hwang, K. J., et al. (2020a). Magnetotail depolarization fronts and particle acceleration: A review. *Science China Earth Sciences*, *63*(2), 235–256. https://doi.org/10.1007/s11430-019-9551-y

Fu, H. S., Khotyaintsev, Y. V., André, M., & Vaivads, A. (2011). Fermi and betatron acceleration of suprathermal electrons behind dipolarization fronts. *Geophysical Research Letters*, *38*(16), L16104. https://doi.org/10.1029/2011GL048528

Fu, H. S., Khotyaintsev, Y. V., Vaivads, A., André, M., Sergeev, V. A., Huang, S. Y., et al. (2012). Pitch angle distribution of suprathermal electrons behind dipolarization fronts: A statistical overview. *Journal of Geophysical Research*, *117*(A12), A12221. https://doi.org/10.1029/2012JA018141

Fu, H. S., Khotyaintsev, Y. V., Vaivads, A., Retinò, A., & André, M. (2013b). Energetic electron acceleration by unsteady magnetic reconnection. *Nature Physics*, *9*(7), 426–430. https://doi.org/10.1038/nphys2664

Fu, H. S., Vaivads, A., Khotyaintsev, Y. V., André, M., Cao, J. B., Olshevsky, V., et al. (2017). Intermittent energy dissipation by turbulent reconnection. *Geophysical Research Letters*, *44*(1), 37–43. https://doi.org/10.1002/2016GL071787

Fu, H. S., Xu, Y., Vaivads, A., & Khotyaintsev, Y. V. (2019). Super-efficient electron acceleration by an isolated magnetic reconnection. *The Astrophysical Journal Letters*, *870*(2), L22. https://doi.org/10.3847/2041-8213/aafa75

Fu, H. S., Zhao, M. J., Yu, Y., & Wang, Z. (2020b). A new theory for energetic electron generation behind dipolarization front. *Geophysical Research Letters*, *47*(6), e2019GL086790. https://doi.org/10.1029/2019GL086790

Fu, W. D., Fu, H. S., Cao, J. B., Yu, Y., Chen, Z. Z., & Xu, Y. (2022). Formation of rolling-pin distribution of suprathermal electrons behind dipolarization fronts. *Journal of Geophysical Research-Space Physics*, *127*(1), e2021JA029642. https://doi.org/10.1029/2021JA029642

Goldman, M., Lapenta, G., Newman, D., Markidis, S., & Che, H. (2011). Jet deflection by very weak guide fields during magnetic reconnection. *Physical Review Letters*, *107*(13), 135001. https://doi.org/10.1103/PhysRevLett.107.135001

Huang, K., Lu, Q., Lu, S., Wang, R., & Wang, S. (2021). Formation of pancake, rolling pin, and cigar distributions of energetic electrons at the dipolarization fronts (DFs) driven by magnetic reconnection: A two-dimensional particle-in-cell simulation. *Journal of Geophysical Research-Space Physics*, *126*(10), e2021JA029939. https://doi.org/10.1029/2021JA029939

Huang, S., Fu, H., Yuan, Z., Zhou, M., Fu, S., Deng, X., et al. (2015). Electromagnetic energy conversion at dipolarization fronts: Multispacecraft results. *Journal of Geophysical Research: Space Physics*, *120*(6), 6188–6198. https://doi.org/10.1002/2015JA021083

Huang, S., Jiang, K., Fu, H., Yuan, Z., Deng, X., Li, H., et al. (2019). Periodical dipolarization processes in Earth's magnetotail. *Geophysical Research Letters*, *46*(23), 13640–13648. https://doi.org/10.1029/2019GL086136

Huang, S., Vaivads, A., Khotyaintsev, Y., Zhou, M., Fu, H., Retinò, A., et al. (2012). Electron acceleration in the reconnection diffusion region: Cluster observations. *Geophysical Research Letters*, *39*(11), L11103. https://doi.org/10.1029/2012GL051946

Huang, S., Xiong, Q., Song, L., Nan, J., Yuan, Z., Jiang, K., et al. (2021). Electron-only reconnection in an ion-scale current sheet at the magnetopause. *The Astrophysical Journal*, *922*(1), 54. https://doi.org/10.3847/1538-4357/ac2668

Huang, S., Xiong, Q., Yuan, Z., Jiang, K., Yu, L., Xu, S., et al. (2023). Crater structure behind reconnection front [Dataset]. Open Science Framework. https://doi.org/10.17605/osf.io/esvzu

Huang, S., Zhou, M., Deng, X., Yuan, Z., Pang, Y., Wei, Q., et al. (2012). Kinetic structure and wave properties associated with sharp dipolarization front observed by Cluster. *Annales Geophysicae*, *30*(1), 97–107. https://doi.org/10.5194/angeo-30-97-2012

Huang, S., Zhou, M., Yuan, Z., Fu, H., He, J., Sahraoui, F., et al. (2015). Kinetic simulations of secondary reconnection in the reconnection jet. *Journal of Geophysical Research: Space Physics*, *120*(8), 6188–6198. https://doi.org/10.1002/2014JA020969

Huang, S. Y., Zhou, M., Yuan, Z. G., Deng, X. H., Sahraoui, F., Pang, Y., & Fu, S. (2014). Kinetic simulations of electric field structure within magnetic island during magnetic reconnection and their applications to the satellite observations. *Journal of Geophysical Research: Space Physics*, *119*(9), 7402–7412. https://doi.org/10.1002/2014JA020054

Hwang, K., Sibeck, D., Choi, E., Chen, L., Ergun, R., Khotyaintsev, Y., et al. (2017). Magnetospheric Multiscale mission observations of the outer electron diffusion region. *Geophysical Research Letters*, *44*(5), 2049–2059. https://doi.org/10.1002/2017GL072830

Hwang, K.-J., Dokgo, K., Choi, E., Burch, J. L., Sibeck, D. G., Giles, B. L., et al. (2020). Magnetic reconnection inside a flux rope induced by Kelvin-Helmholtz vortices. *Journal of Geophysical Research-Space Physics*, *125*(4), e2019JA027665. https://doi.org/10.1029/2019JA027665

Jiang, K., Huang, S., Yuan, Z., Deng, X., Xu, S., Wei, Y., et al. (2020). Observations of electron vortex at the dipolarization front. *Geophysical Research Letters*, *47*(13), e2020GL088448. https://doi.org/10.1029/2020GL088448

Karimabadi, H., Daughton, W., & Scudder, J. (2007). Multi-scale structure of the electron diffusion region. *Geophysical Research Letters*, *34*(13), L13104. https://doi.org/10.1029/2007GL030306

Le, A., Egedal, J., Ohia, O., Daughton, W., Karimabadi, H., & Lukin, V. (2013). Regimes of the electron diffusion region in magnetic reconnection. *Physical Review Letters*, *110*(13), 135004. https://doi.org/10.1103/PhysRevLett.110.135004

Lin, R. T., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Wei, Y. Y., et al. (2023). MAVEN observations of tailward reconnection front in the martin magnetotail. *Journal of Geophysical Research: Space Physics*, *128*(2), e2022JA031030. https://doi.org/10.1029/2022JA031030

Lindqvist, P., Olsson, G., Torbert, R., King, B., Granoff, M., Rau, D., et al. (2016). The spin-plane double probe electric field instrument for MMS. *Space Science Reviews*, *199*(1–4), 137–165. https://doi.org/10.1007/s11214-014-0116-9

Montag, P., Egedal, J., & Daughton, W. (2020). Influence of inflow density and temperature asymmetry on the formation of electron jets during magnetic reconnection. *Geophysical Research Letters*, *47*(20), e2020GL087612. https://doi.org/10.1029/2020GL087612

Nakamura, R., Retino, A., Baumjohann, W., Volwerk, M., Erkaev, N., Klecker, B., et al. (2009). Evolution of dipolarization in the near-Earth current sheet induced by Earthward rapid flux transport. *Annales Geophysicae*, *27*(4), 1743–1754. https://doi.org/10.5194/angeo-27-1743-2009

Øieroset, M., Phan, T., Ergun, R., Ahmadi, N., Genestreti, K., Drake, J., et al. (2021). Spatial evolution of magnetic reconnection diffusion region structures with distance from the X-line. *Physics of Plasmas*, *28*(12), 122901. https://doi.org/10.1063/5.0072182

Pan, D.-X., Khotyaintsev, Y. V., Graham, D. B., Vaivads, A., Zhou, X.-Z., André, M., et al. (2018). Rippled electron-scale structure of a dipolarization front. *Geophysical Research Letters*, *45*(22), 12116–12124. https://doi.org/10.1029/2018GL080826

Pollock, C., Moore, T., Jacques, A., Burch, J., Gliese, U., Saito, Y., et al. (2016). Fast plasma investigation for magnetospheric multiscale. *Space Science Reviews*, *199*(1–4), 331–406. https://doi.org/10.1007/s11214-016-0245-4

Pritchett, P. (2001). Geospace Environment Modeling magnetic reconnection challenge: Simulations with a full particle electromagnetic code. *Journal of Geophysical Research*, *106*(A3), 3783–3798. https://doi.org/10.1029/1999JA001006

Russell, C., Anderson, B., Baumjohann, W., Bromund, K., Dearborn, D., Fischer, D., et al. (2016). The magnetospheric multiscale magnetometers. *Space Science Reviews*, *199*(1–4), 189–256. https://doi.org/10.1007/s11214-014-0057-3

Shay, M., Drake, J., & Swisdak, M. (2007). Two-scale structure of the electron dissipation region during collisionless magnetic reconnection. *Physical Review Letters*, *99*(15), 155002. https://doi.org/10.1103/PhysRevLett.99.155002

Shu, Y., Lu, S., Lu, Q., Ding, W., & Wang, S. (2021). Energy budgets from collisionless magnetic reconnection site to reconnection front. *Journal of Geophysical Research: Space Physics*, *126*(10), e2021JA029712. https://doi.org/10.1029/2021JA029712

Sitnov, M., & Swisdak, M. (2011). Onset of collisionless magnetic reconnection in two-dimensional current sheets and formation of dipolarization fronts. *Journal of Geophysical Research*, *116*(A12), A12216. https://doi.org/10.1029/2011JA016920

Song, L., Zhou, M., Yi, Y., Deng, X., & Zhong, Z. (2019). Reconnection front associated with asymmetric magnetic reconnection: Particle-in-cell simulations. *The Astrophysical Journal*, *881*(1), L22. https://doi.org/10.3847/2041-8213/ab3655

Song, L., Zhou, M., Yi, Y., Deng, X., Zhong, Z., & Man, H. (2020). Force and energy balance of the dipolarization front. *Journal of Geophysical Research: Space Physics*, *125*(9), e2020JA028278. https://doi.org/10.1029/2020ja028278

Sonnerup, B. U. Ö., & Cahill, L. J., Jr. (1967). Magnetopause structure and attitude from Explorer 12 observations. *Journal of Geophysical Research*, *72*(1), 171–183. https://doi.org/10.1029/JZ072i001p00171

Tharp, T., Yamada, M., Ji, H., Lawrence, E., Dorfman, S., Myers, C., et al. (2013). Study of the effects of guide field on Hall reconnection. *Physics of Plasmas*, *20*(5), 055705. https://doi.org/10.1063/1.4805244

Torbert, R., Burch, J., Phan, T., Hesse, M., Argall, M., Shuster, J., et al. (2018). Electron-scale dynamics of the diffusion region during symmetric magnetic reconnection in space. *Science*, *362*(6421), 1391–1395. https://doi.org/10.1126/science.aat2998

Wei, Y. Y., Huang, S. Y., Jiang, K., Yuan, Z. G., Xu, S. B., Zhang, J., et al. (2023). Direct evidence of electron acceleration at the dipolarization front. *The Astrophysical Journal*, *950*(2), 112. https://doi.org/10.3847/1538-4357/acd1dd

Wei, Y. Y., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Deng, X. H., et al. (2022). Observations of pitch angle changes of electrons and high-frequency wave activities in the magnetotail plasma bubble. *Journal of Geophysical Research: Space Physics*, *127*(1), e2021JA029761. https://doi.org/10.1029/2021JA029761

Xiong, Q., Huang, S., Zhou, M., Yuan, Z., Deng, X., Jiang, K., et al. (2022a). Formation of negative J·E′ in the outer electron diffusion region during magnetic reconnection. *Journal of Geophysical Research: Space Physics*, *127*(2), e2022JA030264. https://doi.org/10.1029/2022JA030264

Xiong, Q., Huang, S., Zhou, M., Yuan, Z., Deng, X., Jiang, K., et al. (2022b). Distribution of negative J·E′ in the inflow edge of the inner electron diffusion region during tail magnetic reconnection: Simulations vs. observations. *Geophysical Research Letters*, *49*(11), e2022GL098445. https://doi.org/10.1029/2022GL098445

Xiong, Q. Y., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Lin, R. T., & Yu, L. (2023). Electron backflow motions in the outer electron diffusion region during magnetic reconnection. *Geophysical Research Letters*, *50*(21), e2023GL105300. https://doi.org/10.1029/2023GL105300

Xiong, Q. Y., Huang, S. Y., Yuan, Z. G., Jiang, K., Xu, S. B., Wei, Y. Y., et al. (2022c). Statistic properties of electron energy enhancement during the inner electron diffusion region crossing. *Journal of Geophysical Research: Space Physics*, *127*(10), e2022JA030760. https://doi.org/10.1029/2022JA030760

Xu, S. B., Huang, S. Y., Yuan, Z. G., Jiang, K., Wei, Y. Y., Zhang, J., et al. (2022). Successive dipolarization fronts with a stepwise electron acceleration during a substorm in Saturn's magnetotail. *Geophysical Research Letters*, *49*(5), e2021GL097227. https://doi.org/10.1029/2021GL097227

Xu, Y., Fu, H. S., Liu, C. M., & Wang, T. Y. (2018). Electron acceleration by dipolarization fronts and magnetic reconnection: A quantitative comparison. *The Astrophysical Journal*, *853*(1), 11. https://doi.org/10.3847/1538-4357/aa9f2f

Yu, Y., Fu, H. S., Wang, Z., Fu, W. D., & Cao, J. B. (2023). Formation of electron butterfly distribution by a contracting dipolarization front. *Geophysical Research Letters*, *50*(17), e2023GL104938. https://doi.org/10.1029/2023GL104938

Yu, Y., Wang, Z., Fu, H. S., & Cao, J. B. (2022). Direct evidence of interchange instabilities at dipolarization fronts. *Journal of Geophysical Research: Space Physics*, *127*(10), e2022JA030805. https://doi.org/10.1029/2022JA030805

Zenitani, S., Hesse, M., Klimas, A., & Kuznetsova, M. (2011). New measure of the dissipation region in collisionless magnetic reconnection. *Physical Review Letters*, *106*(19), 195003. https://doi.org/10.1103/PhysRevLett.106.195003

Zhao, M. J., Fu, H. S., Liu, C. M., Chen, Z. Z., Xu, Y., Giles, B. L., & Burch, J. L. (2019). Energy range of electron rolling pin distribution behind dipolarization front. *Geophysical Research Letters*, *46*(5), 2390–2398. https://doi.org/10.1029/2019GL082100

Zhou, M., Deng, X., & Huang, S. (2012). Electric field structure inside the secondary island in the reconnection diffusion region. *Physics of Plasmas*, *19*(4), 042902. https://doi.org/10.1063/1.3700194

Zhou, M., Deng, X., Tang, R., Pang, Y., Xu, X., Yuan, Z., & Huang, S. (2014). Evidence of deflected super-Alfvénic electron jet in a reconnection region with weak guide field. *Journal of Geophysical Research: Space Physics*, *119*(3), 1541–1548. https://doi.org/10.1002/2013JA019556