# Three-dimensional Electromagnetic Particle Code with Adaptive Mesh Refinement Technique

Keizo Fujimoto[1,2,3] and Richard D. Sydora[1]

1. Department of Physics, University of Alberta
2. NiCT
3. Research Fellow (PD) of the JSPS

# Introduction

Three-dimensional (3D) effects of local plasma kinetic processes are, in many cases, believed to have a significant impact on the global energy transport and conversion processes occurring in the solar-terrestrial system. Magnetic reconnection is one of the most important phenomena which support the fast energy release associated with the magnetospheric substorm and solar flares. It facilitates the fast conversion of energy stored in a compressed magnetic field into plasma kinetic and thermal energies. However, its 3D processes are poorly understood. The difficulties are mainly caused due to its strong nonlinearity and the limited computer resources.

In this study, we successfully constructed a new 3D electromagnetic particle code using the adaptive mesh refinement (AMR) technique, which enables efficient kinetic simulations on many plasma processes. We show the code description, several test simulations, and the performance of the code.

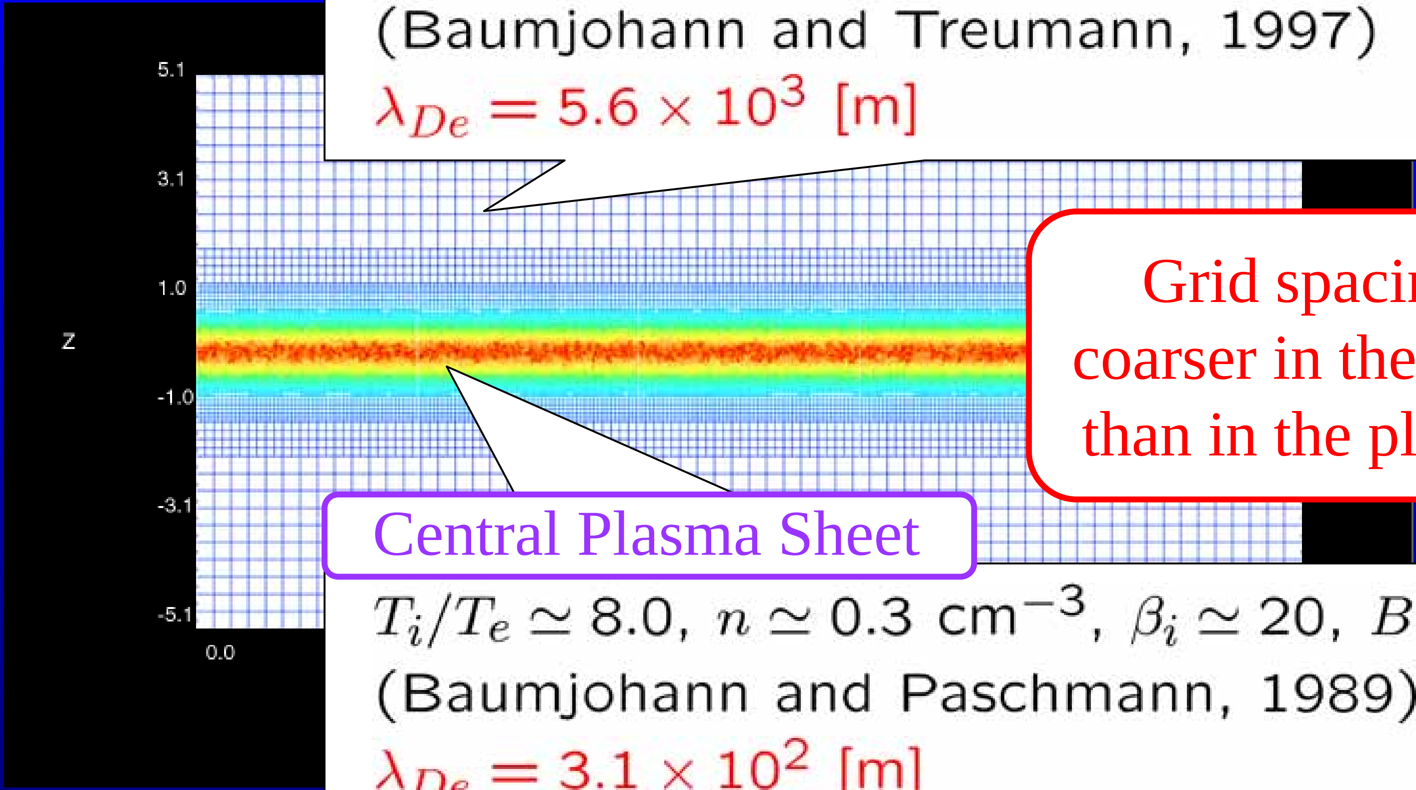
# *Introduction*

Restriction in full particle code: $\Delta x \lesssim 3\lambda_{De} \sim 1km$

[*Birdsall & Langdon*, 1995]

Magnetotail Lobe

$T_i/T_e \simeq 4.0$, $n \simeq 0.01$ cm$^{-3}$, $\beta_i \simeq 0.1$, $B = 30$ nT.
(Baumjohann and Treumann, 1997)
$\lambda_{De} = 5.6 \times 10^3$ [m]

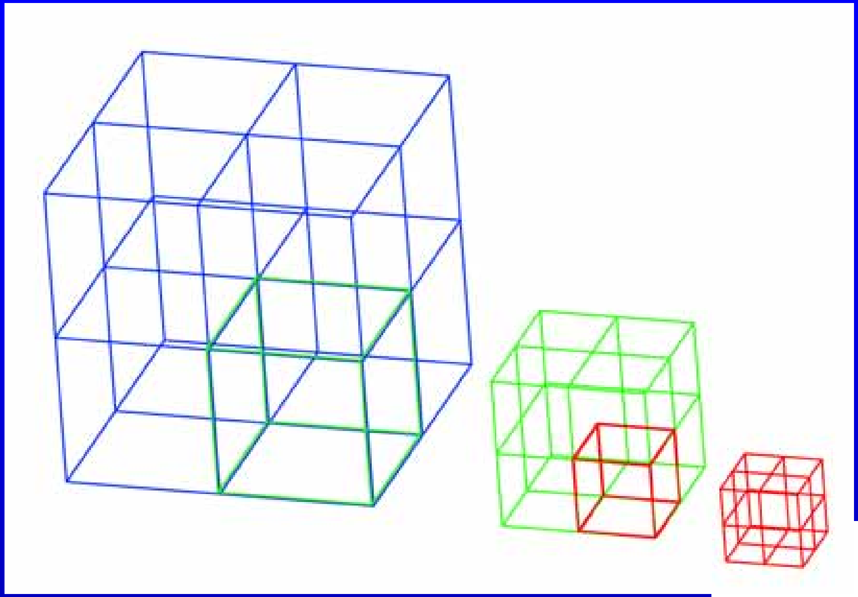


Grid spacing can be coarser in the lobe region than in the plasma sheet.

Central Plasma Sheet

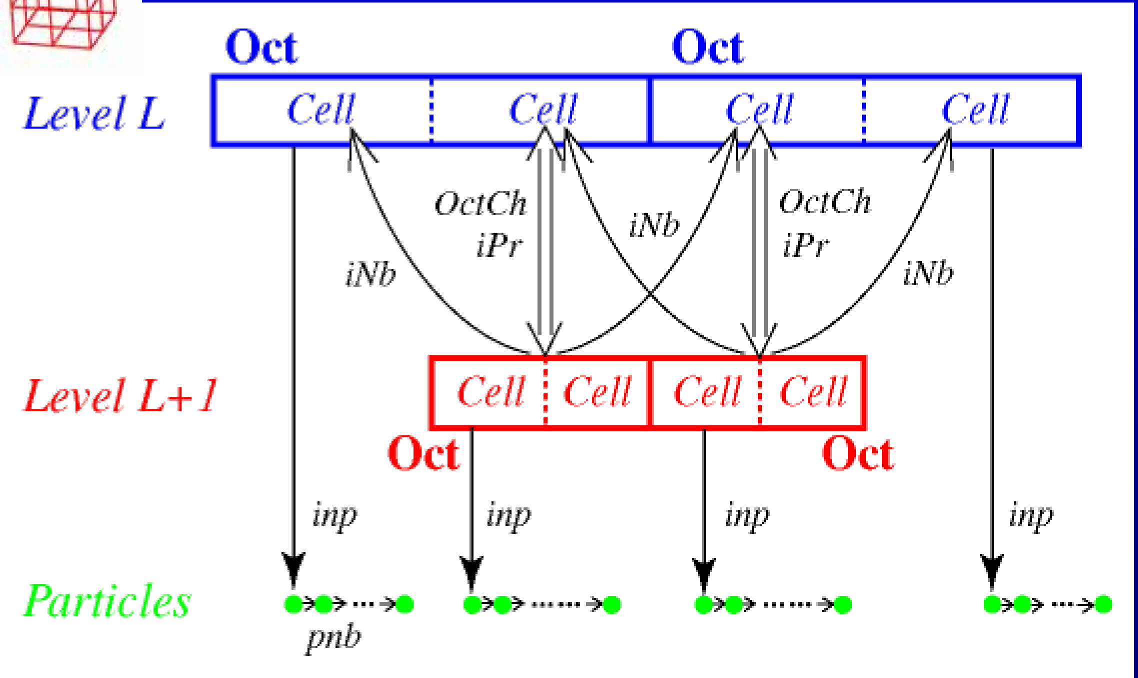
$T_i/T_e \simeq 8.0$, $n \simeq 0.3$ cm$^{-3}$, $\beta_i \simeq 20$, $B = 5$ nT.
(Baumjohann and Paschmann, 1989)
$\lambda_{De} = 3.1 \times 10^2$ [m]

# Data Structure



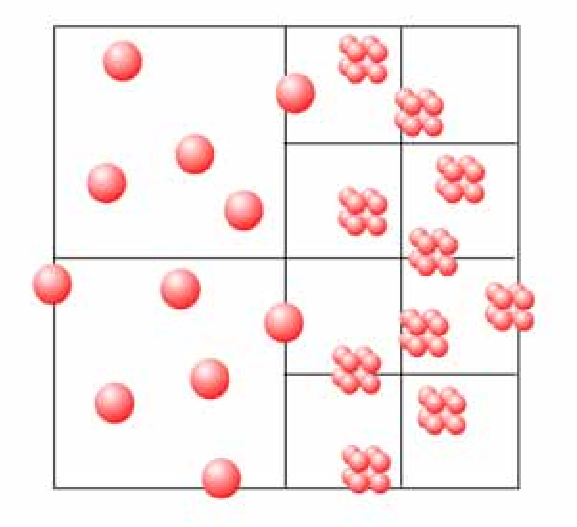
Cells are treated as independent units organized in refinement trees rather than elements of arrays, so that a very flexible cell hierarchy is achieved.

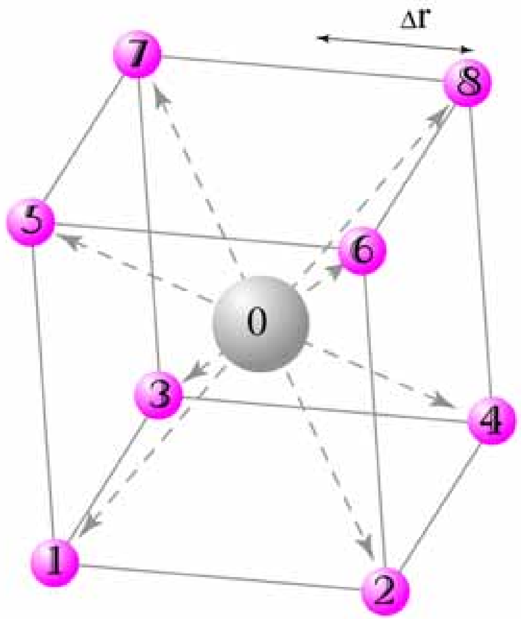
The hierarchical cell structure is supported by a set of pointers, which is basically same as the fully threaded tree (FTT) structure [*Khokhlov*, 1989].



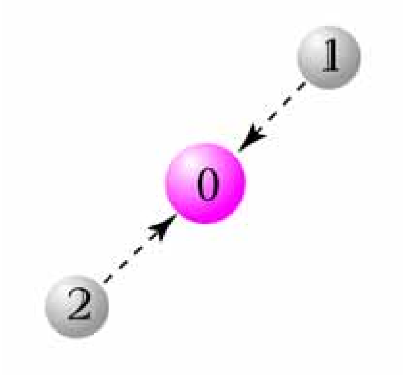
# Particle Splitting and Coalescence

The mesh splitting in the PIC code leads to the decrease in the number of superparticles per cell, which causes the increase in the numerical noise. Particle splitting and coalescence is one of the solutions for controlling the number of particles per cell.





$$x_{i,j} = x_{0,j} - (-1)^k \Delta r$$
$$k = [(i-1)/2^{j-1}]$$
$$v_{i,j} = v_{0,j}$$
$$q_i = q_0/8$$
$$m_i = m_0/8$$
$$\Delta r = \Delta_L / N_p^{1/3}$$
$$i = 1, 2, ..., 8$$
$$j = 1, 2, 3$$



$$q_1 = q_2; m_1 = m_2$$
$$x_0 = (x_1 + x_2)/2$$
$$v_0 = (v_1 + v_2)/2$$
$$q_0 = q_1 + q_2; m_0 = m_1 + m_2$$

# Basic Equations and Mesh Splitting Criteria

- Basic equations

$$\frac{d\boldsymbol{x}_s}{dt} = \boldsymbol{v}_s, \qquad \frac{d\boldsymbol{v}_s}{dt} = \frac{q_s}{m_s}[\boldsymbol{E}(\boldsymbol{x}_s) + \boldsymbol{v}_s \times \boldsymbol{B}(\boldsymbol{x}_s)] \quad (s = i,\, e)$$

$$\boldsymbol{E}_L^{n+1} = -\nabla\phi^{n+1}, \qquad \nabla^2\phi^{n+1} = -\rho^{n+1}/\varepsilon_0$$

$$\left(1 - \frac{c^2\Delta t^2}{4}\nabla^2\right)\boldsymbol{E}_T^{n+1} = \left(1 + \frac{c^2\Delta t^2}{4}\nabla^2\right)\boldsymbol{E}_T^n + c^2\Delta t\,\nabla\times\boldsymbol{B}^n - \Delta t\,\boldsymbol{j}_T^{n+1/2}/\varepsilon_0,$$

$$\boldsymbol{j}_T^{n+1/2} = \boldsymbol{j}^{n+1/2} - (\varepsilon_0/\Delta t)\,\nabla(\phi^{n+1} - \phi^n)$$

$$\boldsymbol{E}^{n+1} = \boldsymbol{E}_L^{n+1} + \boldsymbol{E}_T^{n+1}$$

$$\boldsymbol{B}^{n+1} = \boldsymbol{B}^n - \frac{\Delta t}{2}\nabla\times(\boldsymbol{E}_T^n + \boldsymbol{E}_T^{n+1})$$

- Mesh splitting criteria

$$\{\Delta x > 2\lambda_{De}\} \quad \text{or} \quad \left\{\left|\frac{1}{B}\nabla_\perp B\right|^{-1} < \lambda_{gi} \quad \text{and} \quad |\boldsymbol{j}_e| > j_{crit}\right\}$$

# *Initial Setting for the Test Simulations*

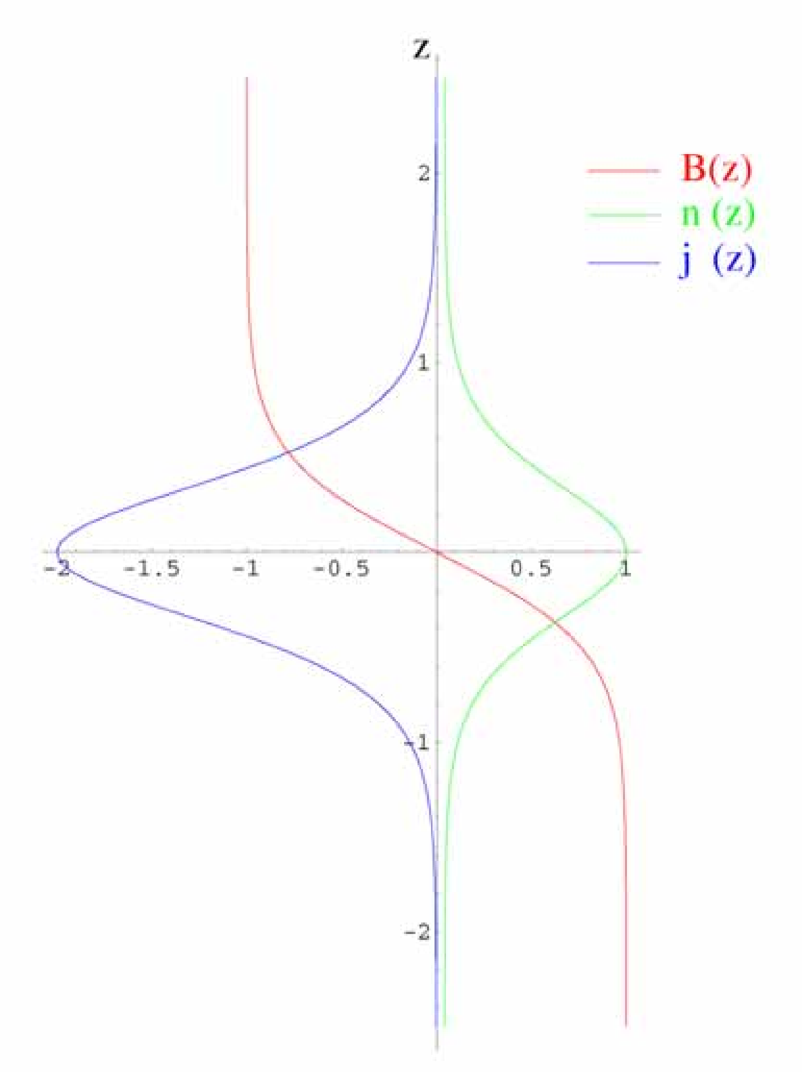
Harris-type current sheet

$m_i/m_e$ = 100; Ti/Te = 5; $c/v_{e,th}$ = 5;
$\lambda/\lambda_i$ = 0.5; $n_{bk}$ = 0.044;

$$B_x(z) = - B_0 \tanh[z/\lambda]$$

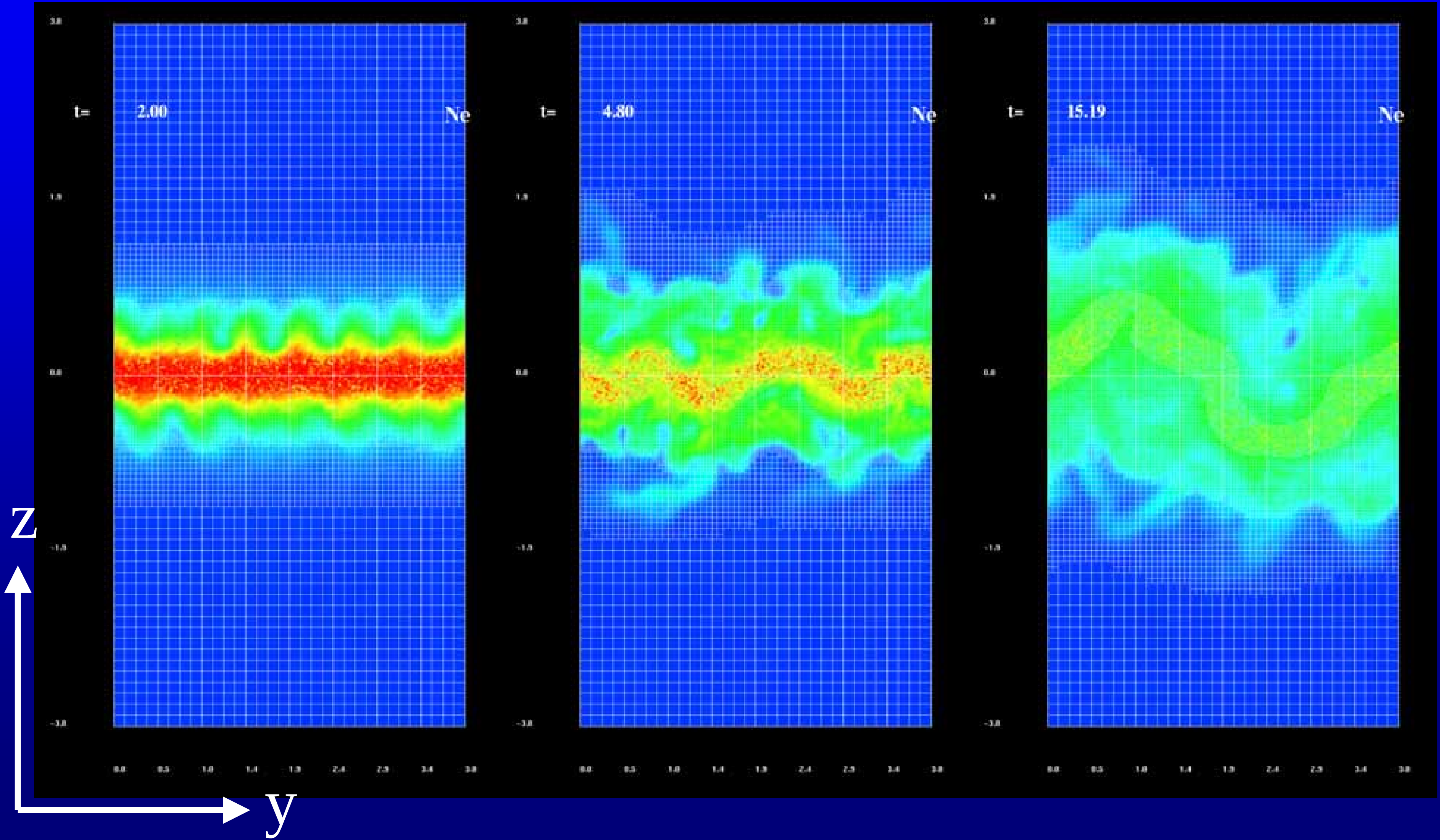
$$n (z) = n_0/\cosh^2[z/\lambda] + n_{bk} \tanh^2[z/\lambda]$$

$$j (z) = en_0(Vi - Ve)/\cosh^2[z/\lambda]$$



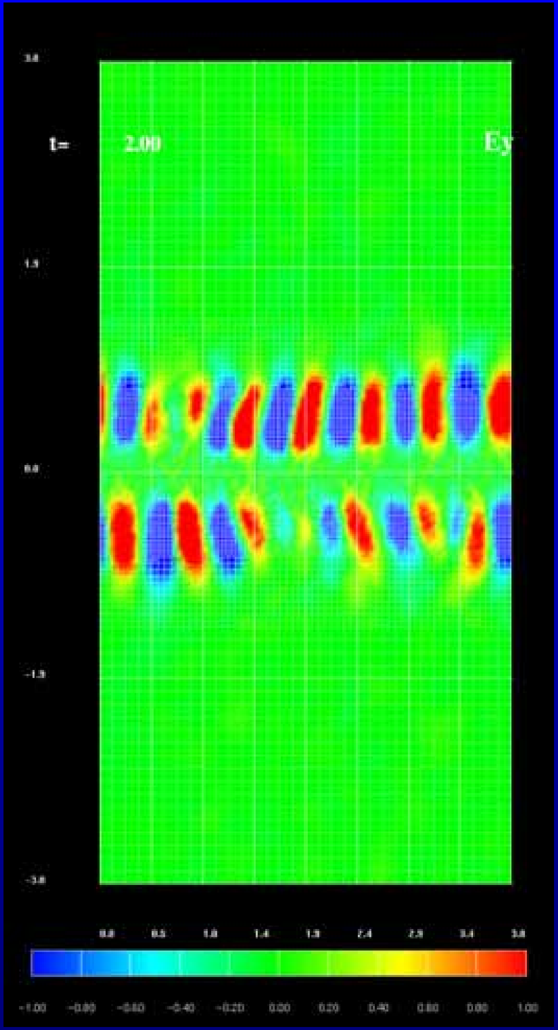
# *Test Simulations in the Non-tearing System*

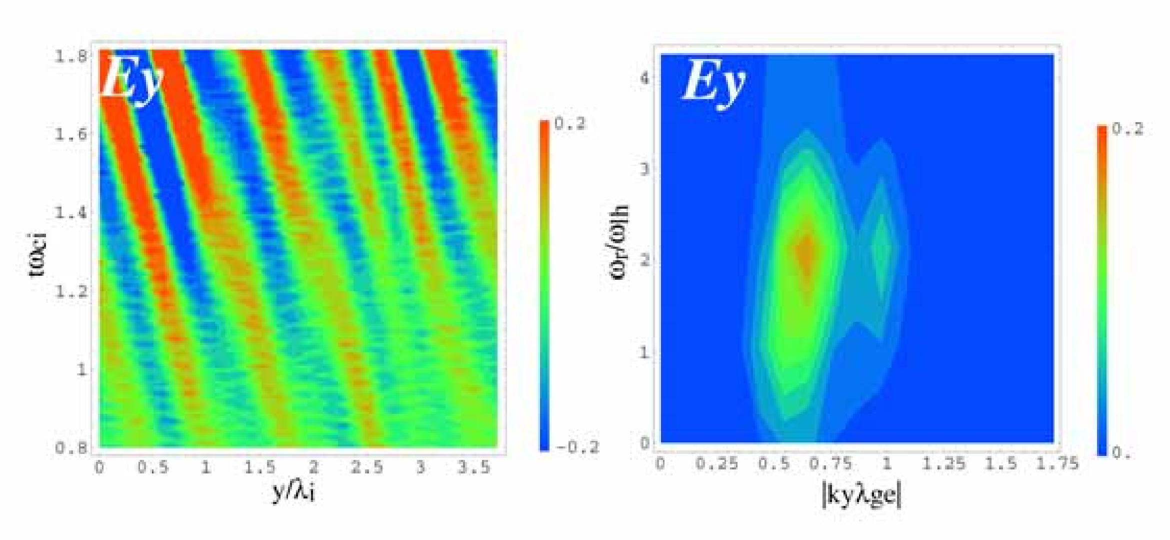
Lx ✕ Ly ✕ Lz = 0.24λi ✕ 3.84λi ✕ 30.72λi



# Test Simulations in the Non-tearing System

- Linear dispersion of the lower hybrid drift instability (LHDI)





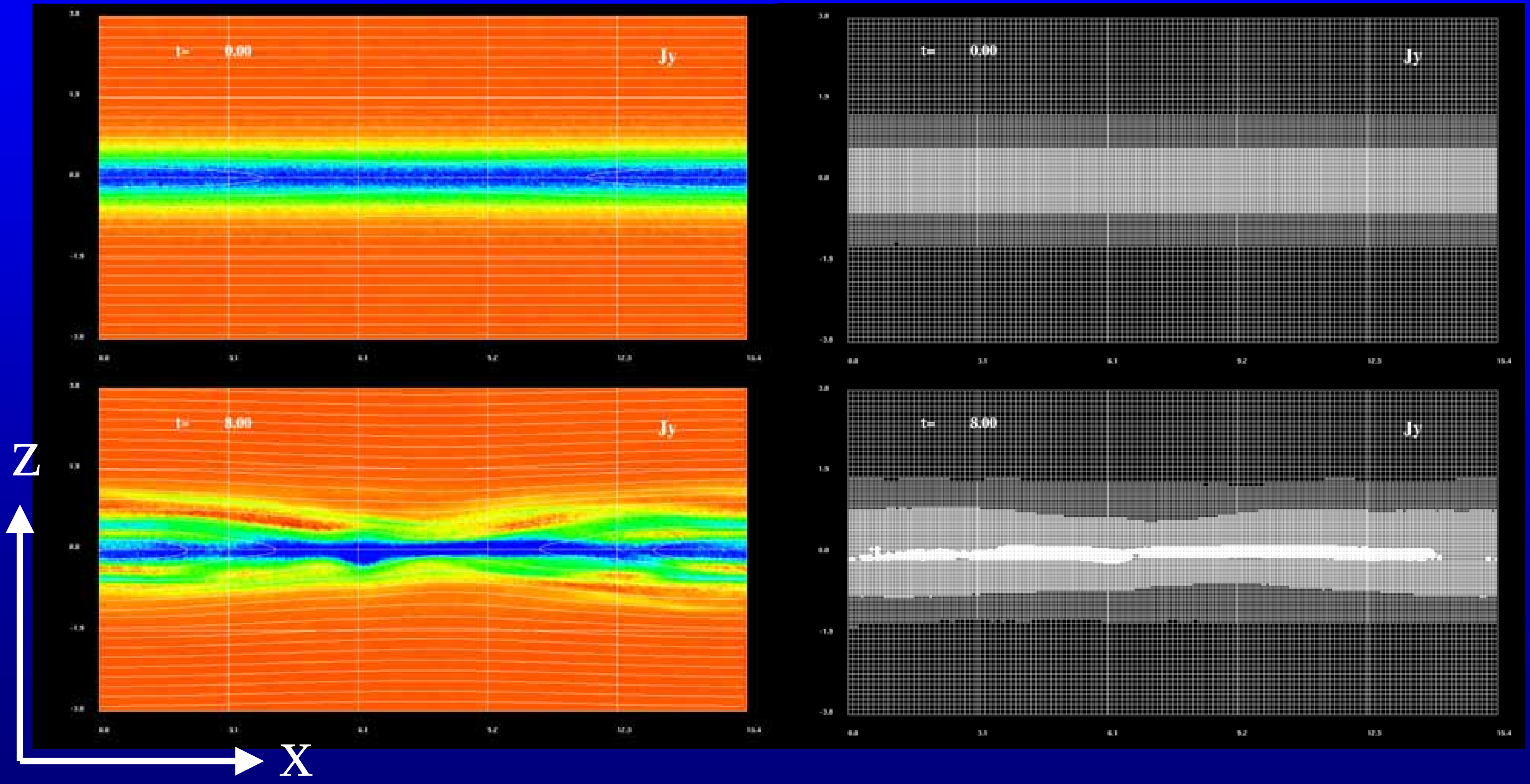
Analytical solution [*Davidson et al.*, 1977]

$$|k_M \lambda_{ge}| = \lambda_{ge} \frac{\sqrt{2}\omega_{pi}}{v_i} \left( \frac{1 + \beta_i/2}{1 + \omega_{pe}^2/\omega_{ce}^2} \right)^{1/2} \simeq 0.770$$

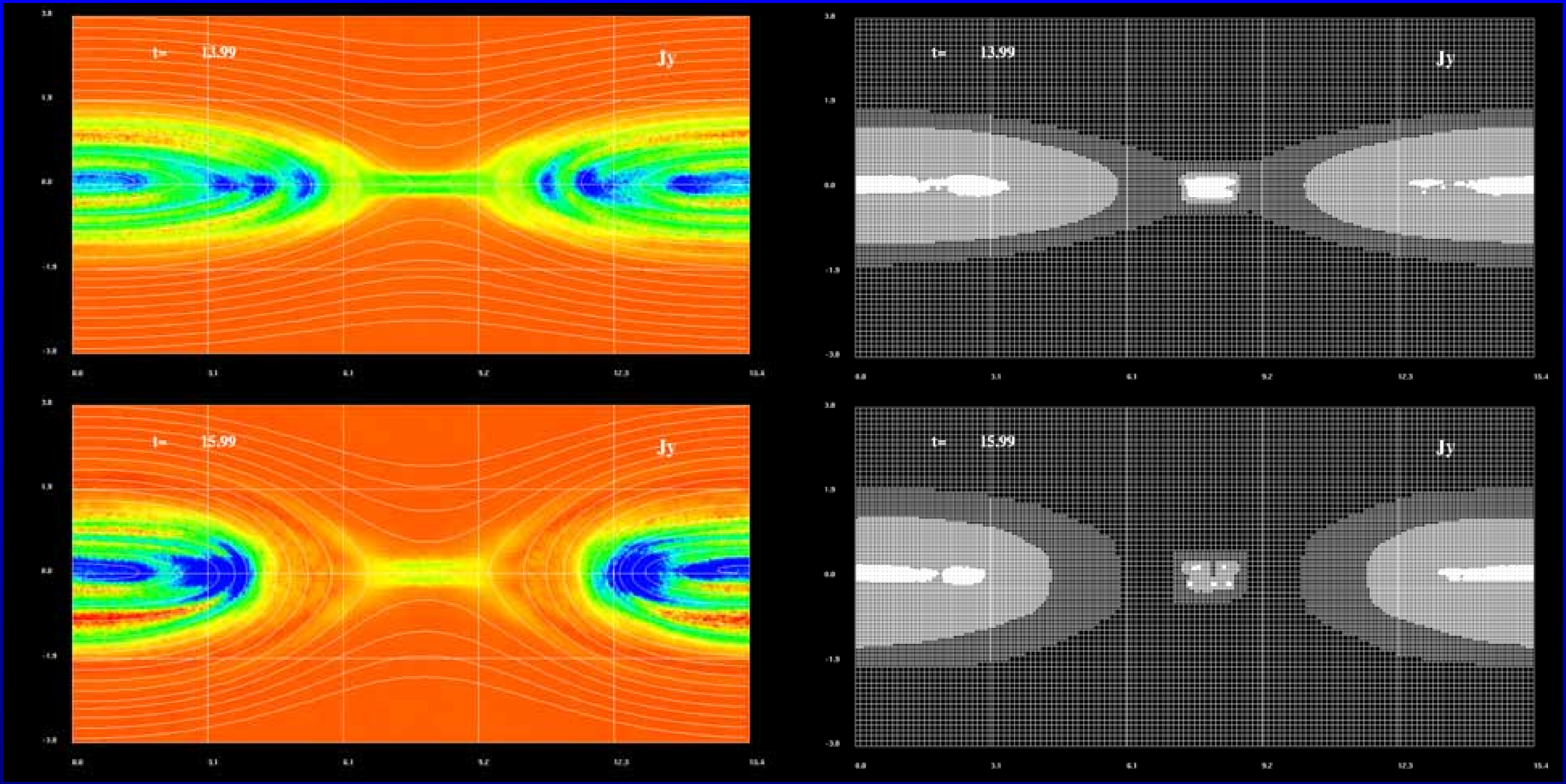
$$\omega_M/\omega_{lh} = \left( \frac{1 + \beta_i/2}{2} \right)^{1/2} \frac{|V_{di}|}{v_i} \simeq 1.68$$

# Test Simulations on Magnetic Reconnection

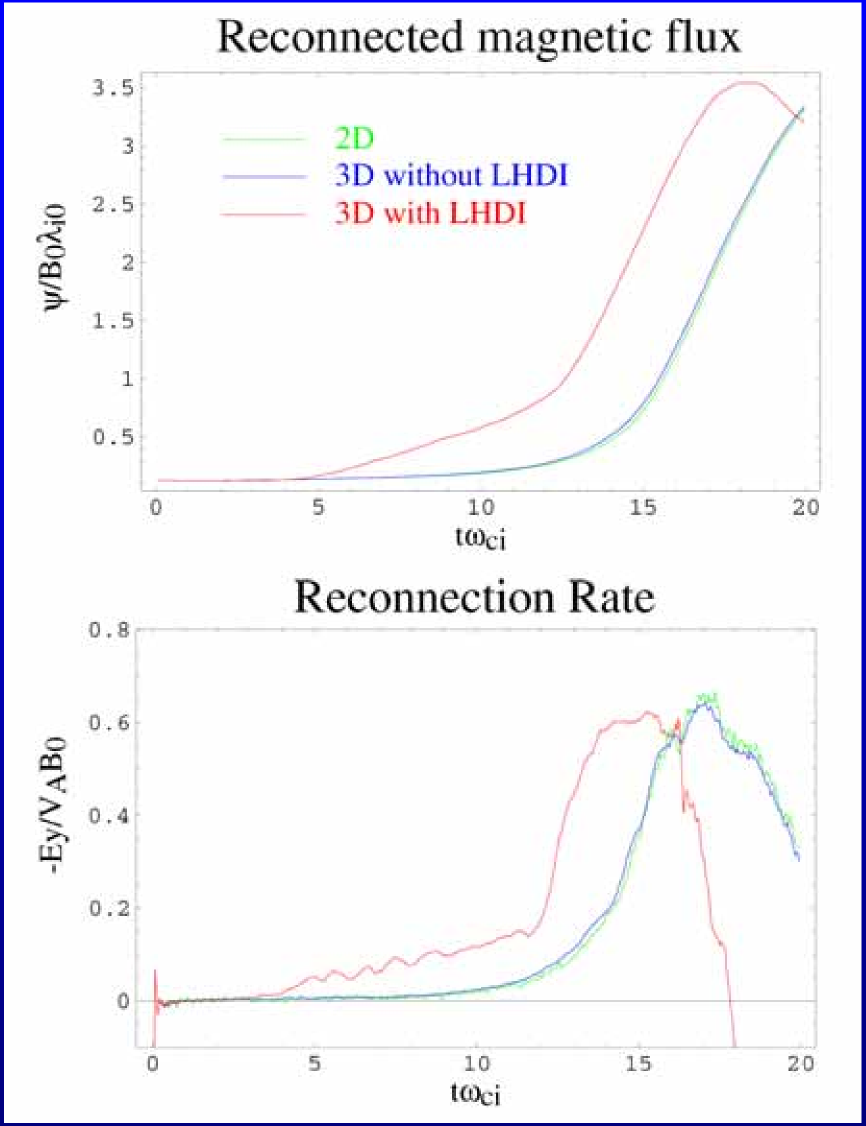
Lx×Ly×Lz = 15.36λi×0.96λi×15.36λi



# *Test Simulations on Magnetic Reconnection*



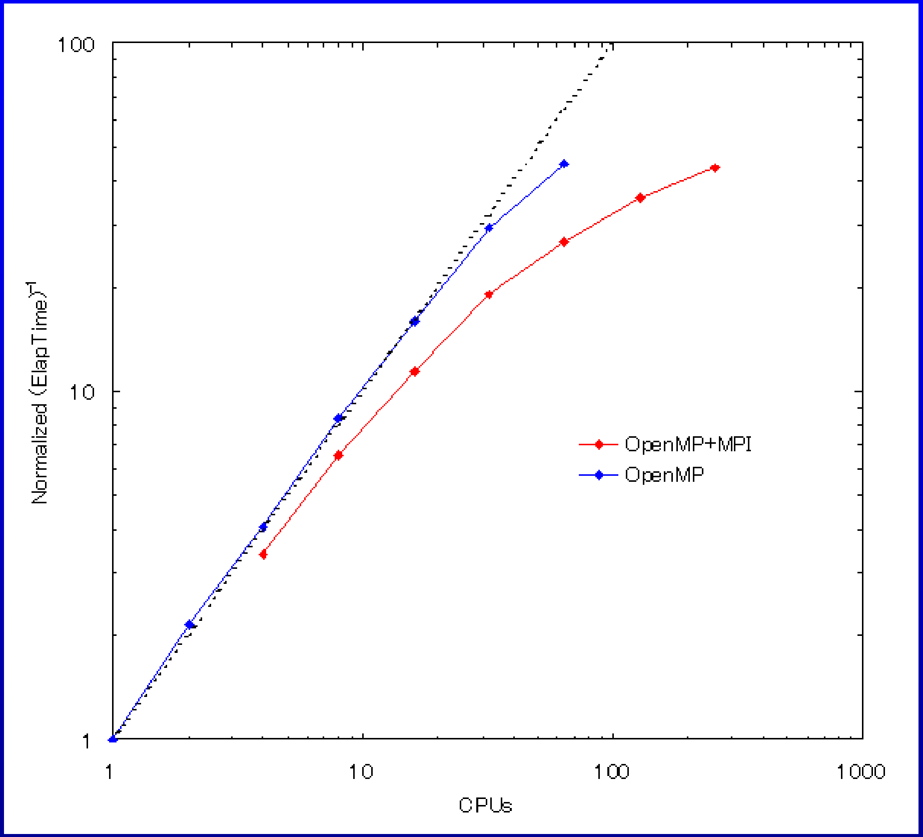
# Test Simulations on Magnetic Reconnection



One of the main differences arising between 3D and 2D reconnections is whether the system involves the LHDI or not. The LHDI is caused by the drifting plasma at the flanks of the current sheet and flattens the density profile. Due to the resultant pressure imbalance, the current sheet is pushed from the both sides by the flank magnetic pressure. As a result, the current density is enhanced at the center of the current sheet, which facilitates the onset of a fast reconnection.

# Performance on the Parallel Computing

Implementing 3D simulations is generally much harder than 2D simulations in realistic systems, since it requires much more cells and particles. We overcome this difficulty by parallelizing the code, using the OpenMP and the message passing interface (MPI). The performance is greatly enhanced by using a number of threads and nodes.



(The performance test was conducted by FUJITSU HPC2500 installed at Nagoya University.)

# *Summary and Conclusions*

We have constructed a new 3D electromagnetic particle code with the adaptive mesh refinement technique.

The code was checked against two kinds of instability: the LHDI, and the tearing instability. We found that the wave dispersion of the LHDI obtained from the simulations is consistent with the theoretical prediction. Furthermore, it is clear that the nonlinear evolution of the tearing instability in the 3D system is basically identical with the 2D results if the LHDI is not excited. It is also confirmed that the fine meshes are properly distributed according to the mesh splitting criteria.

We improved the computing performance and efficiency by carefully parallelizing the code, using the OpenMP and MPI.

Therefore, we can conclude that the code is successfully developed and provides a high efficiency.