

# 解適合格子法によるプラズマ粒子 (AMR-PIC)シミュレーション

藤本 桂三

国立天文台 理論研究部

# 内容

- 開発の動機、戦略    ➡    理解したい物理は何か？
- AMR-PICコードの開発
- 最近の研究成果
- まとめ

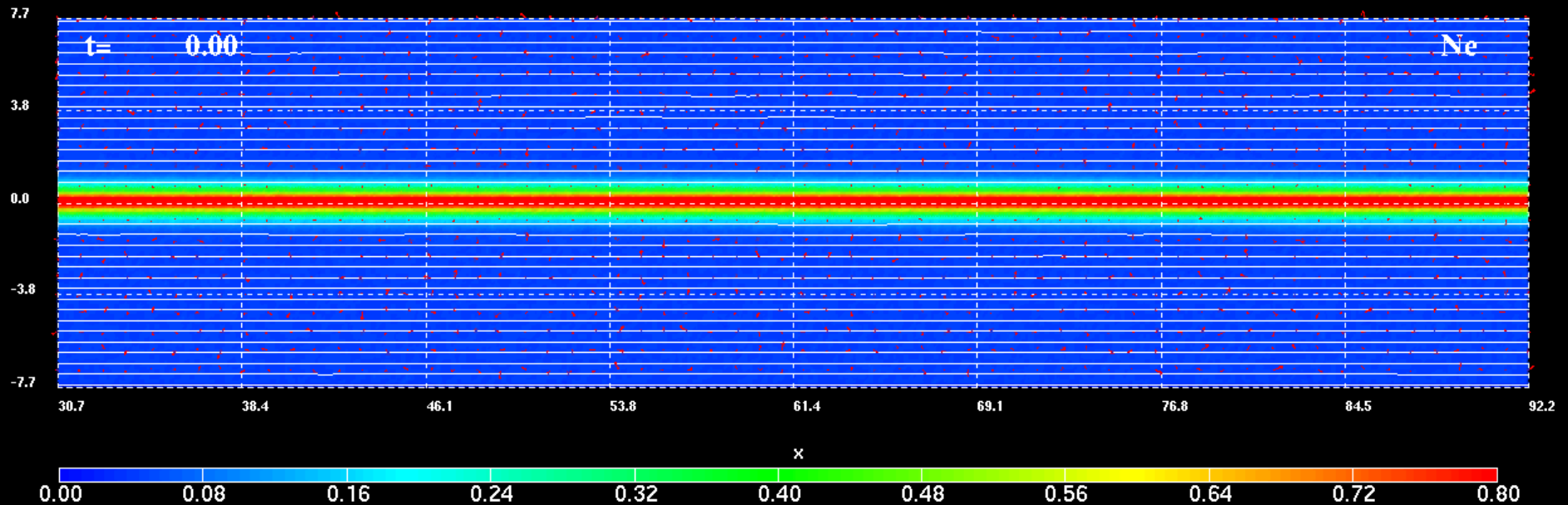
# 開発の動機、戦略

SS2012

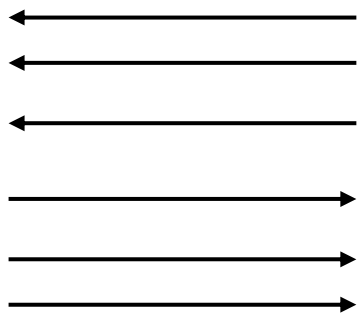
# Magnetic Reconnection

Line: Field line, Contour:  $n_e$   
Arrow:  $V_e$

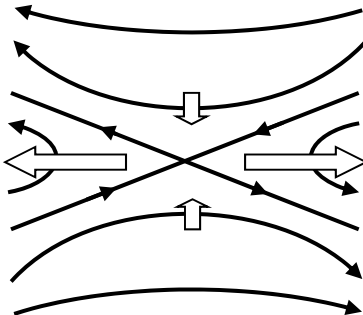
/home/rkeizo/AMR\_code/2d\_main\_cyc\_4/output/x1029mg50n4ip/part/03569.tif



Anti-parallel B-field



Particle acceleration

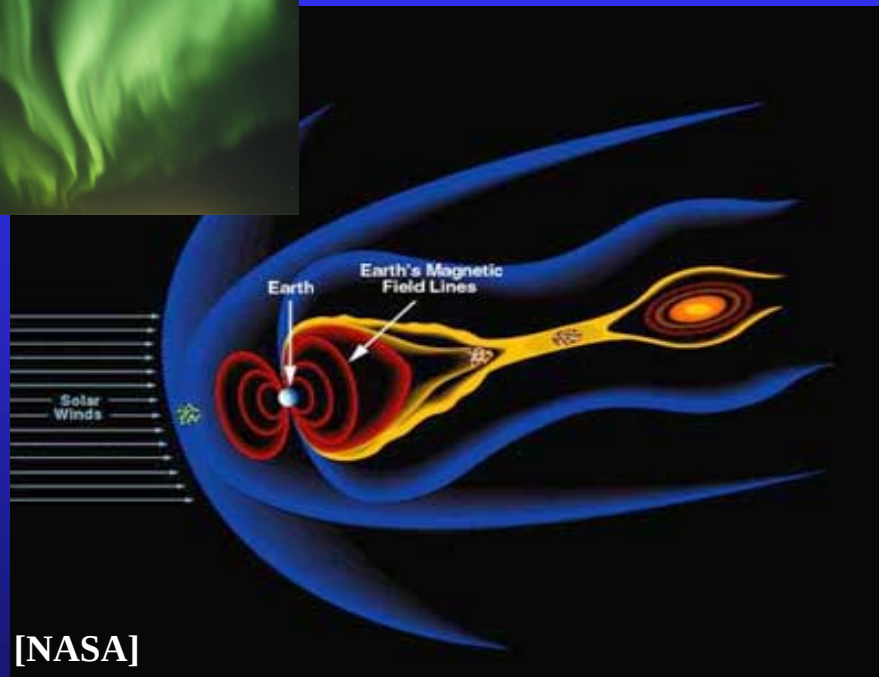


- Change in field line topology
- Plasma acceleration and heating

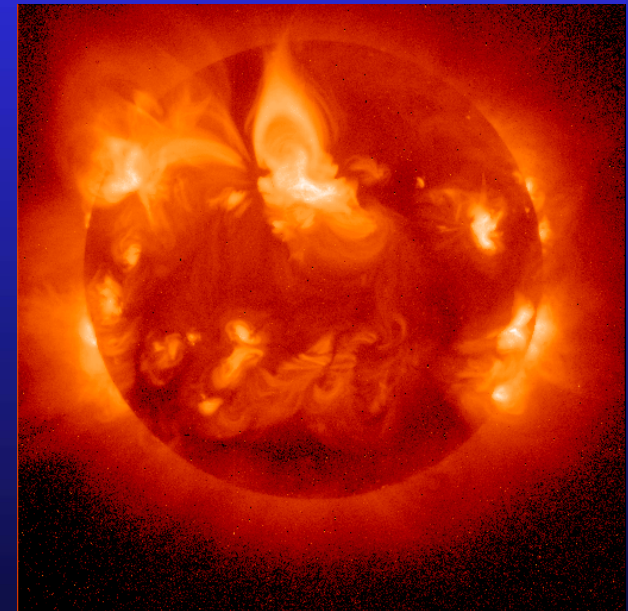
# Magnetic Reconnection in Space



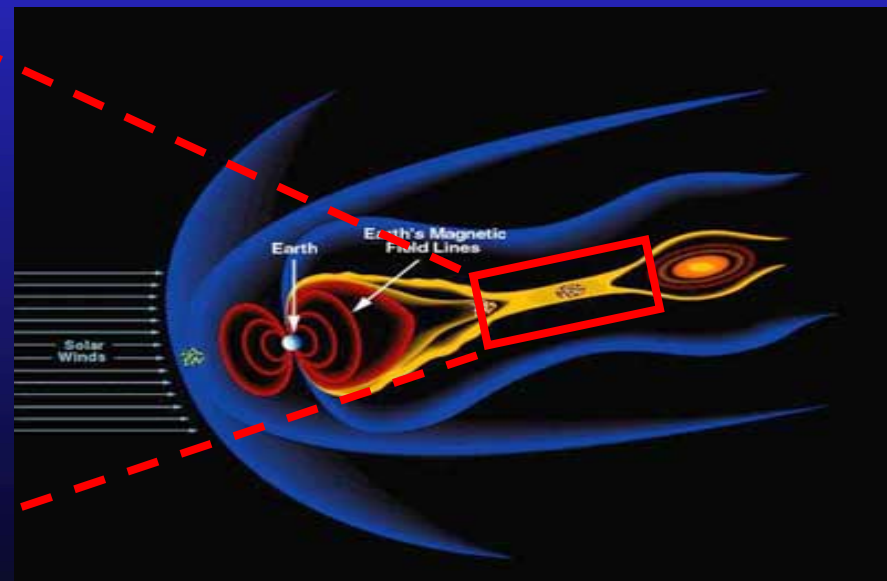
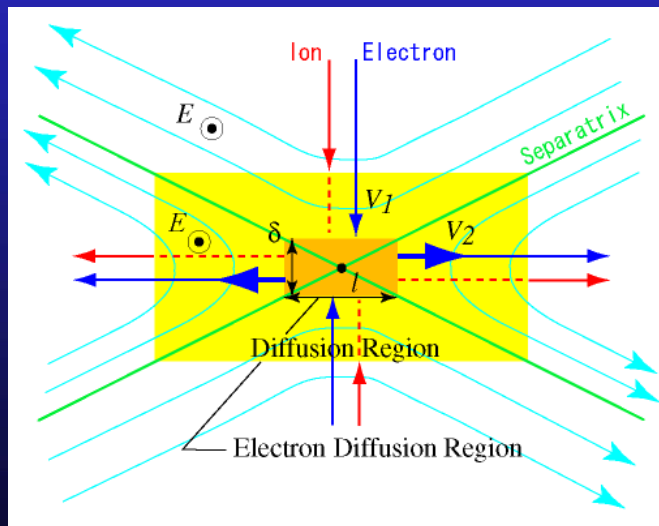
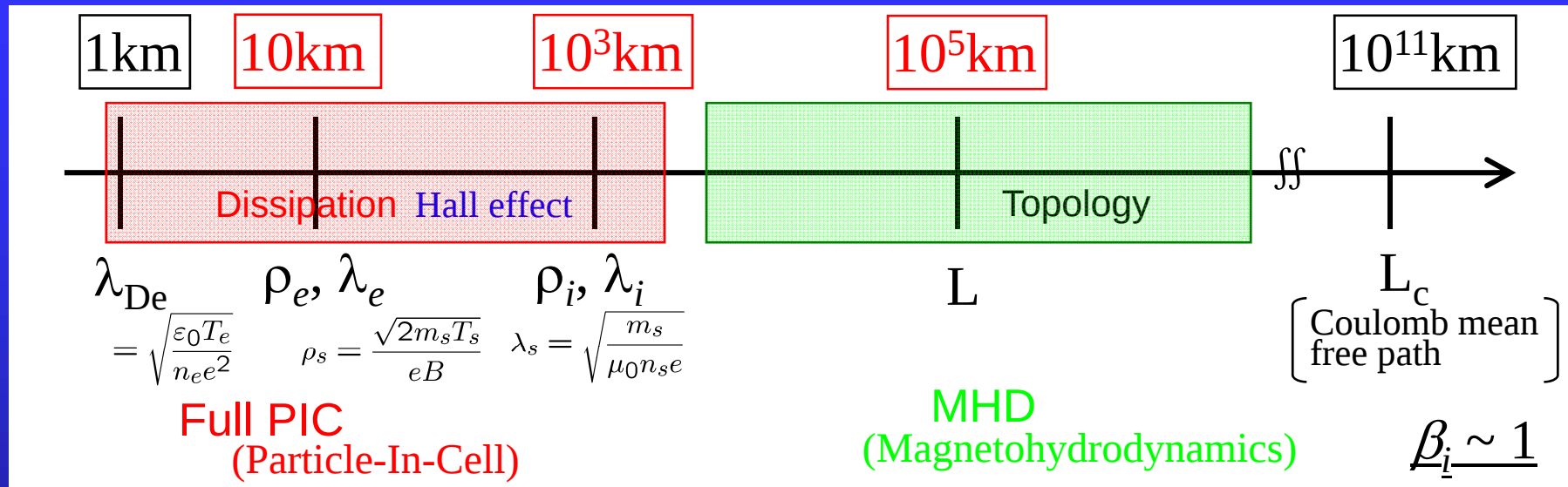
地球磁気圏(オーロラ)サブストーム



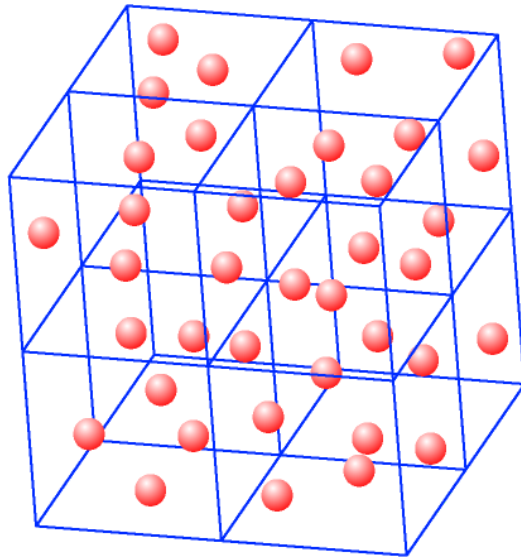
太陽フレア



# Multi-Scale Nature of Reconnection



# 陽的PICスキームの制約



Grid spacing

$$\Delta x \lesssim 3\lambda_{De} \quad \lambda_{De} = \sqrt{\frac{\epsilon_0 T_e}{n_e e^2}}$$

Num of particles per grid

$$N_p \gtrsim 10^2$$

## Memory requirement per grid

Field ( $n_s, J_s, E, B$ )  
14 × 4 Byte

<<

Particle ( $x_s, v_s$ )  
(12 × 4 Byte) × 10<sup>2</sup>

使用可能メモリ

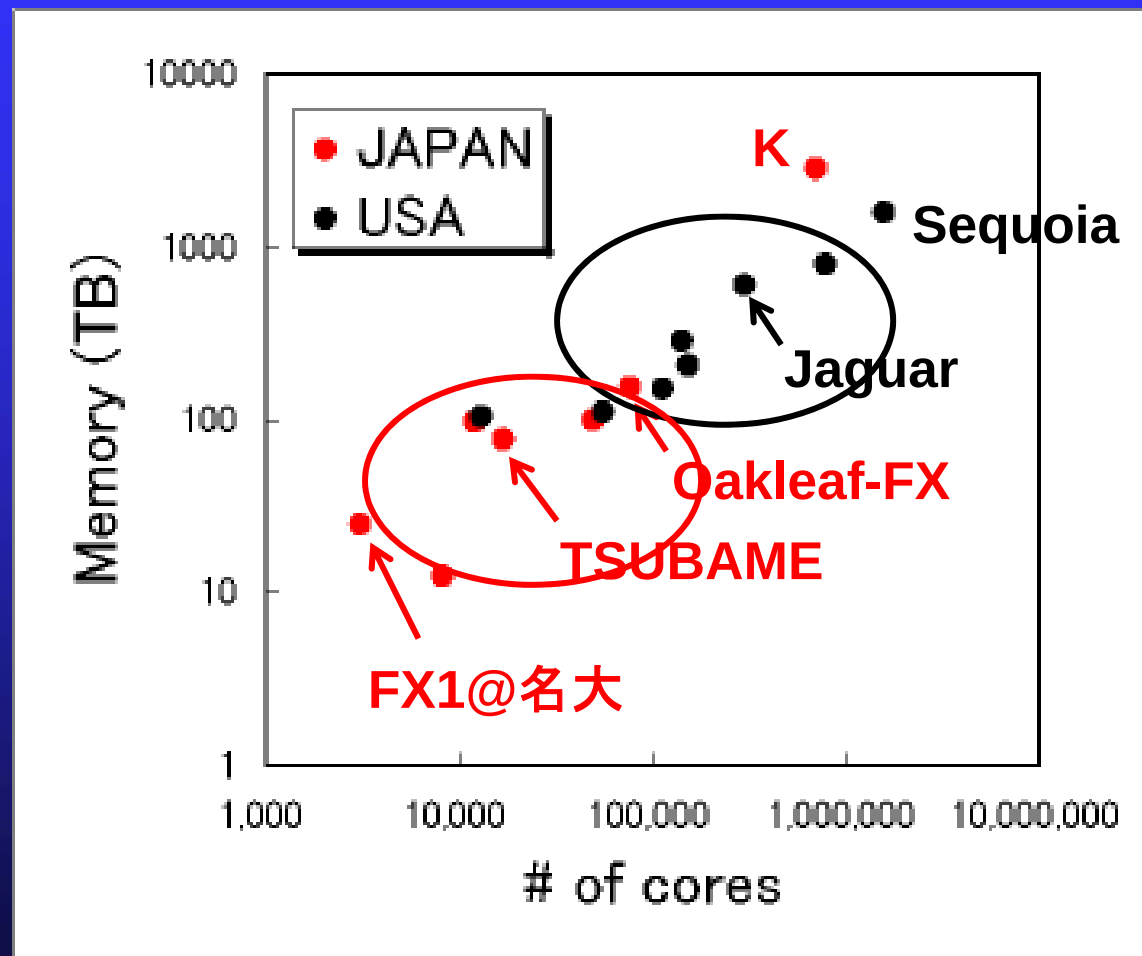


粒子数決定



システムサイズ

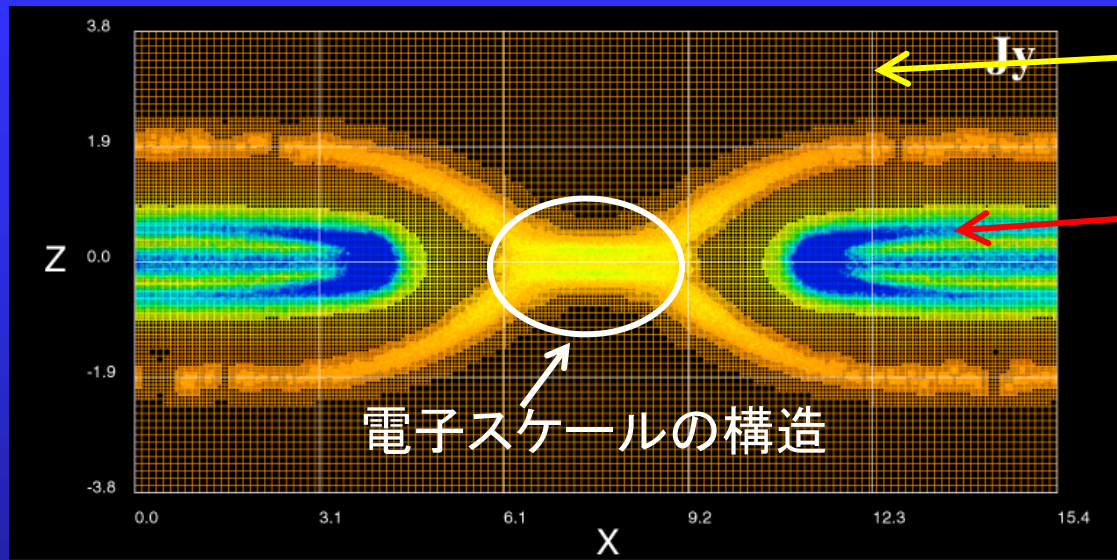
# 国内外のスパコン事情



10倍

# AMR-PICコード [Fujimoto, JCP, 2011] (Adaptive Mesh Refinement – Particle-in-Cell)

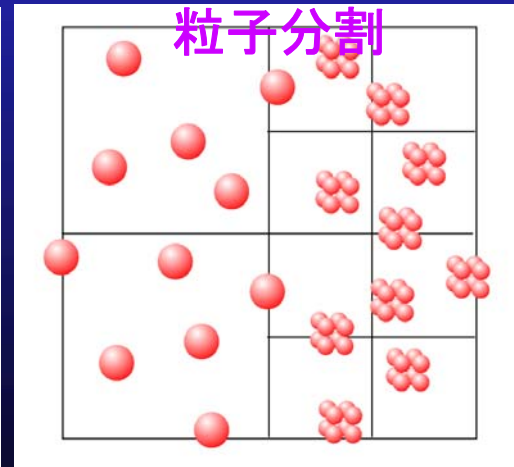
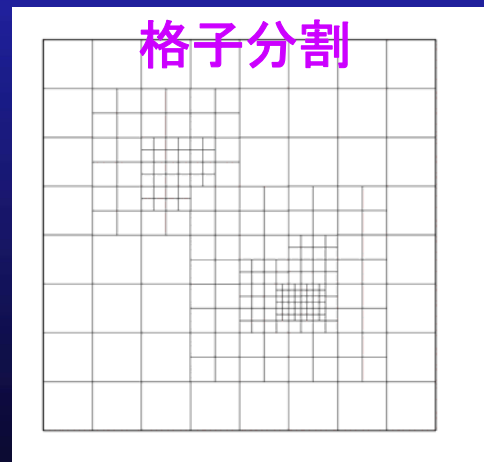
陽解法の制約  $\Delta x \lesssim 3\lambda_{De}$



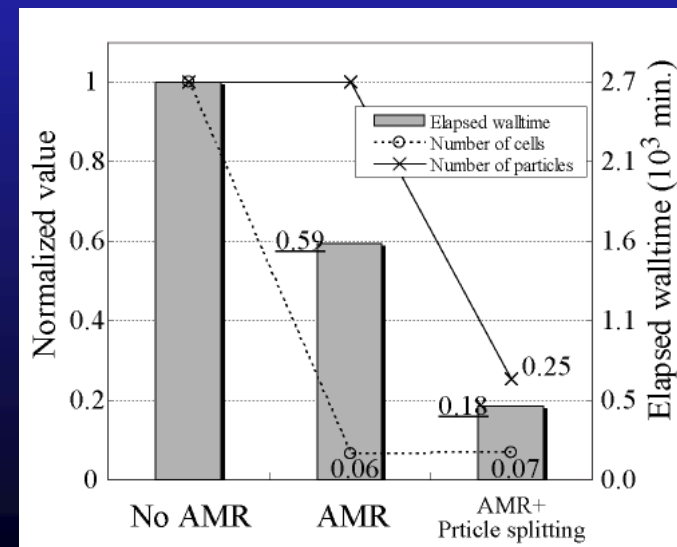
$$\lambda_{De,lobe} \sim 6 \times 10^3 \text{ m}$$

$$\lambda_{De,ps} \sim 3 \times 10^2 \text{ m}$$

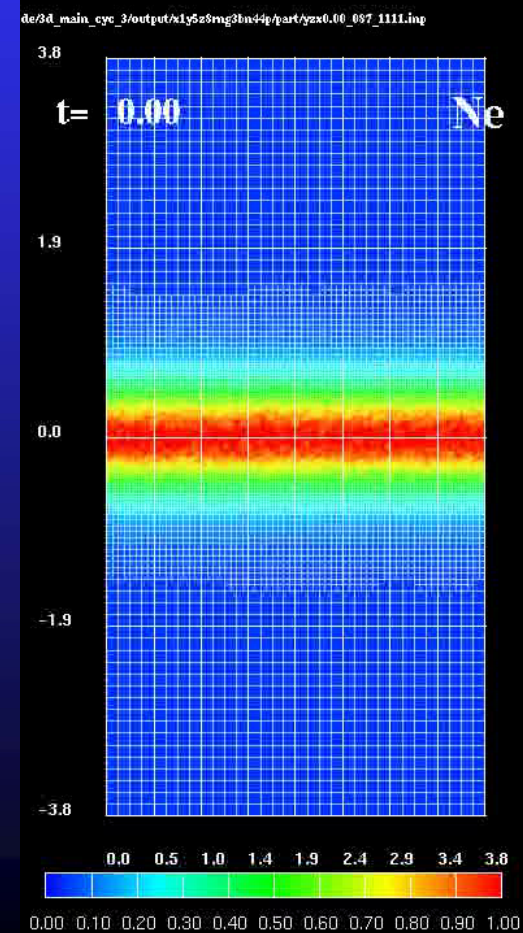
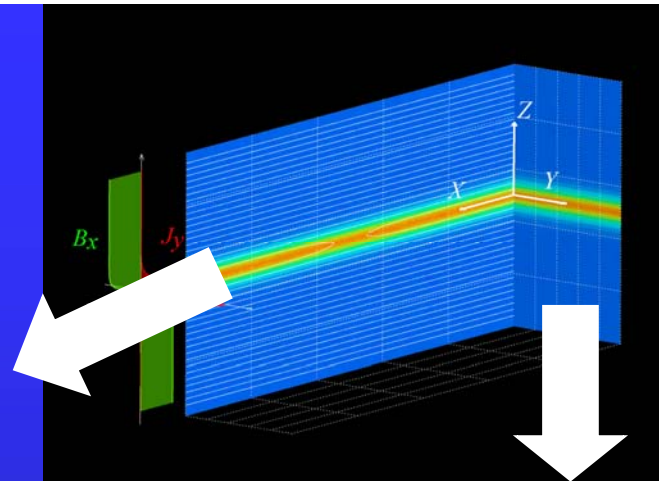
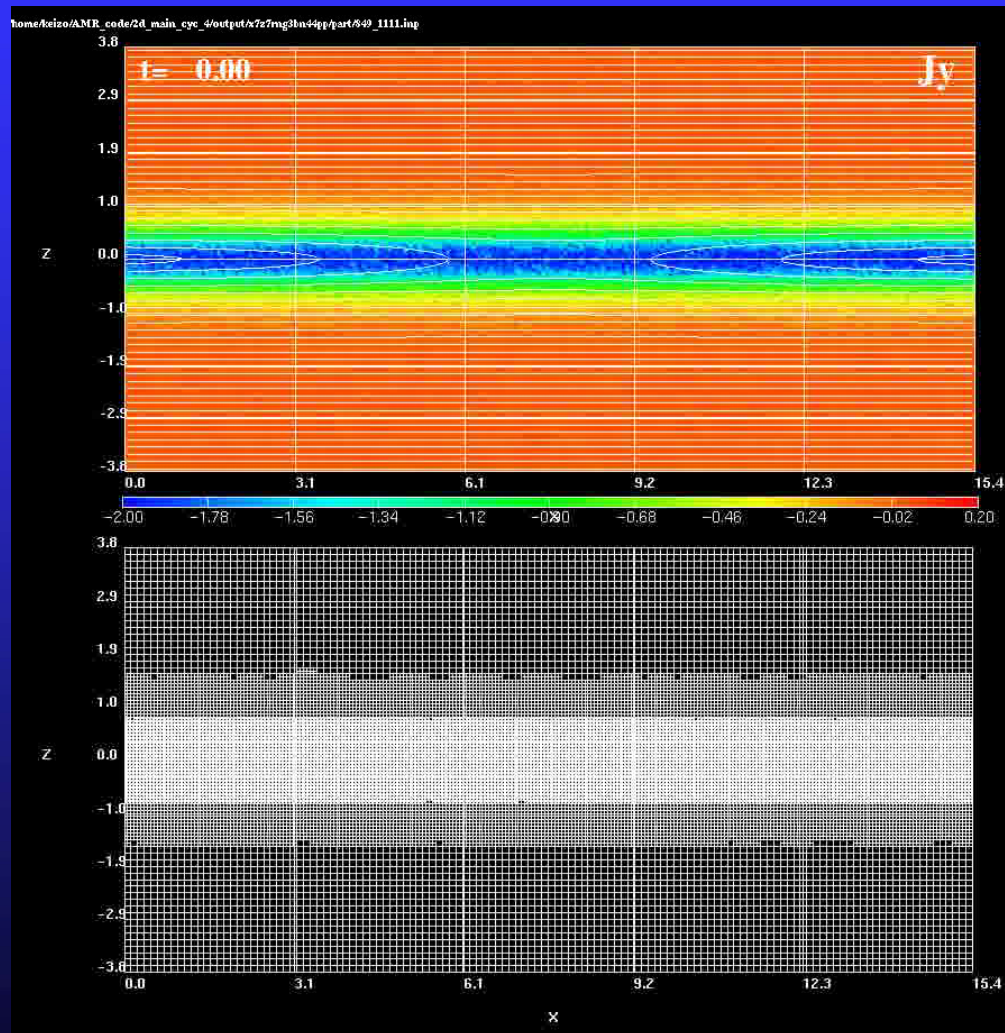
不必要な格子を取り  
除く！



SS2012



# AMR-PIC Simulations



# AMR-PICコードの開発

SS2012

# 陽的電磁粒子(EMPIC)コード

[Birdsall and Langdon, IOP, 1991]

$$\rho_{l,m,n} = \sum_s \sum_j q_{sj} S(\vec{x}_{sj} - \vec{X}_{l,m,n})$$

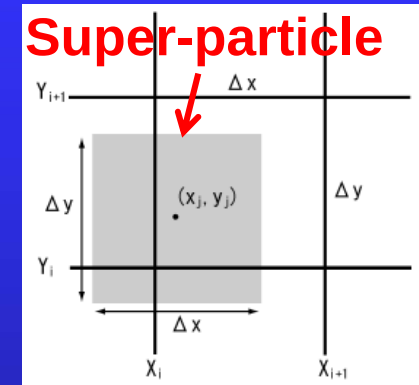
$$A(\vec{x}_{sj}) = \sum_l \sum_m \sum_n A_{l,m,n} S(\vec{x}_{sj} - \vec{X}_{l,m,n})$$

S: Shape function

## ● 超粒子(Buneman-Boris法)

$$\frac{\vec{v}_{sj}^{n+1/2} - \vec{v}_{sj}^{n-1/2}}{\Delta t} = \frac{q_{sj}}{m_{sj}} \left[ \vec{E}^n(\vec{x}_{sj}^n) + \frac{\vec{v}_{sj}^{n-1/2} + \vec{v}_{sj}^{n+1/2}}{2} \times \vec{B}^n(\vec{x}_{sj}^n) \right]$$

$$\frac{\vec{x}_{sj}^{n+1} - \vec{x}_{sj}^n}{\Delta t} = \vec{v}_{sj}^{n+1/2}$$



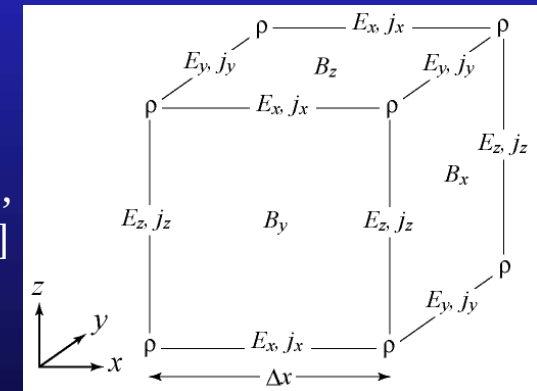
## ● 電磁場(Yee-Bunemanスキーム)

$$\frac{\vec{B}^{n+1/2} - \vec{B}^{n-1/2}}{\Delta t} = -\nabla \times \vec{E}^n$$

$$\frac{\vec{E}^{n+1} - \vec{E}^n}{\Delta t} = c^2 \nabla \times \vec{B}^{n+1/2} - \frac{1}{\epsilon_0} \vec{j}^{n+1/2}$$

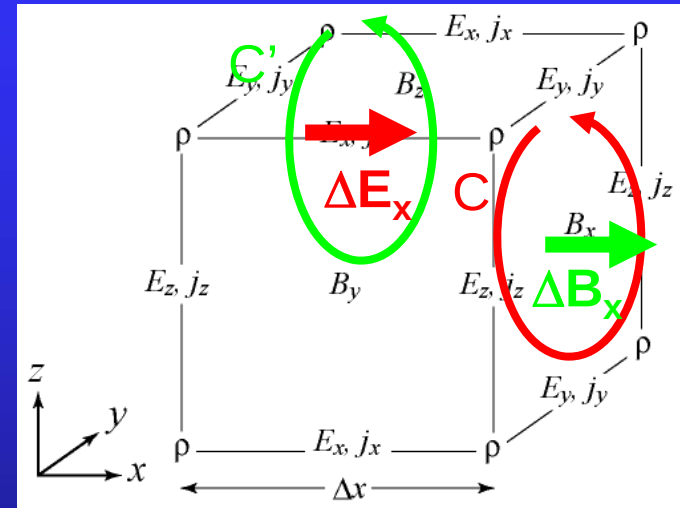
電荷保存法

[Villasenor & Buneman, 1992]



# Staggering Grid Scheme [Yee, 1966; Buneman, 1968]

$$\frac{B_x^{n+1/2} - B_x^{n-1/2}}{\Delta t} \Delta y \Delta z = - \oint_C \vec{E}^n \cdot d\vec{s}$$
$$\frac{E_x^{n+1} - E_x^n}{\Delta t} \Delta y \Delta z = c^2 \oint_{C'} \vec{B}^{n+1/2} \cdot d\vec{s}$$



- 電磁フラックスが厳格に保存
- 電磁波に対する数値散逸なし

# von Neumannの安定解析

$$E_j^n = g^n \exp[ik(j\Delta x)]$$
$$g = |g| \exp[-i\omega\Delta t]$$

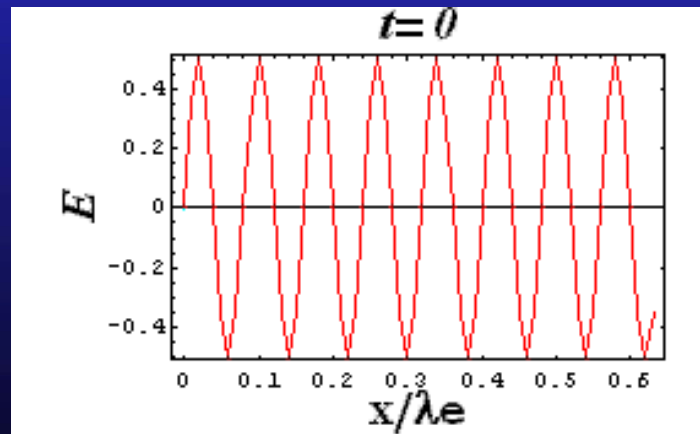


$$\frac{\vec{B}^{n+1/2} - \vec{B}^{n-1/2}}{\Delta t} = -\nabla \times \vec{E}^n$$
$$\frac{\vec{E}^{n+1} - \vec{E}^n}{\Delta t} = c^2 \nabla \times \vec{B}^{n+1/2}$$

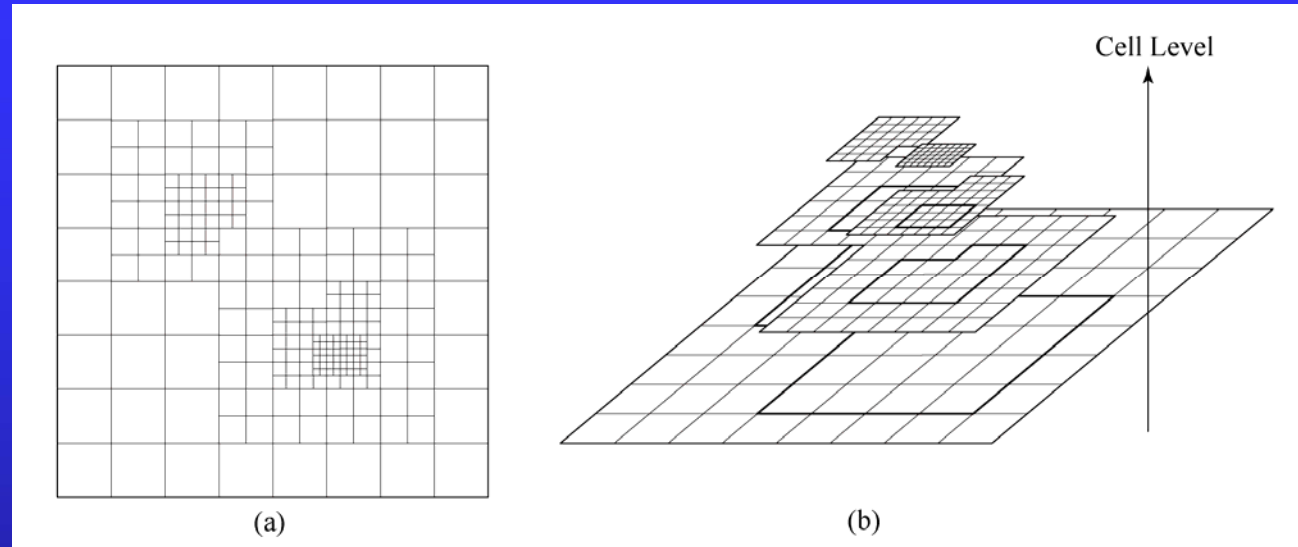
$$g = 1 - \frac{(\kappa c \Delta t)^2}{2} \pm i(\kappa c \Delta t) \sqrt{1 - \frac{(\kappa c \Delta t)^2}{4}} \quad (\Delta x / \Delta t > c)$$

$$|g| = 1$$

$$\kappa = \frac{\sin\left(\frac{k\Delta x}{2}\right)}{\left(\frac{k\Delta x}{2}\right)} k$$

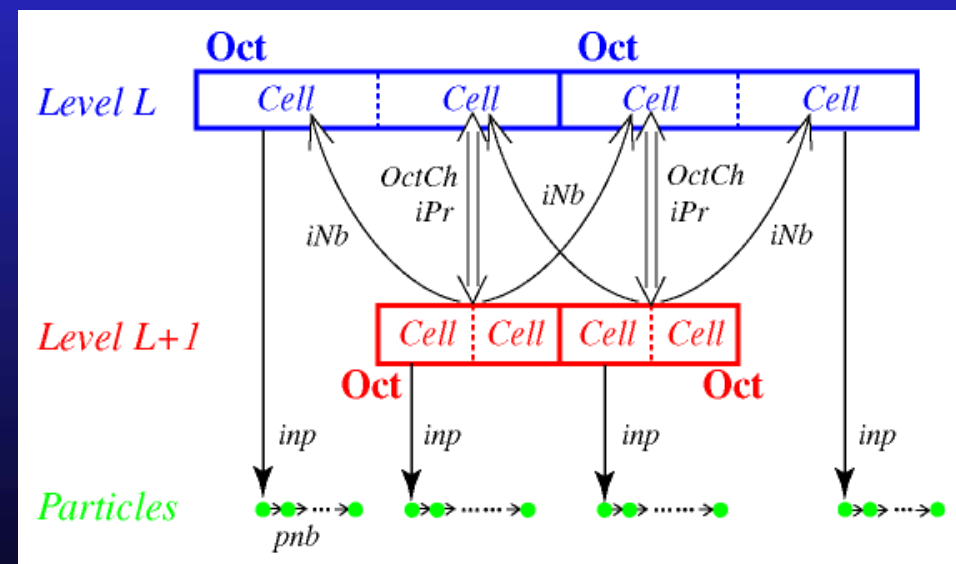


# AMRの適用: データ構造



Similar to a fully threaded tree (FTT) structure (Khokhlov, 1998).

[Fujimoto & Machida, JCP, 2006]



# AMRの適用: 階層間同期

細分化格子の解がより高精度

**Level L**

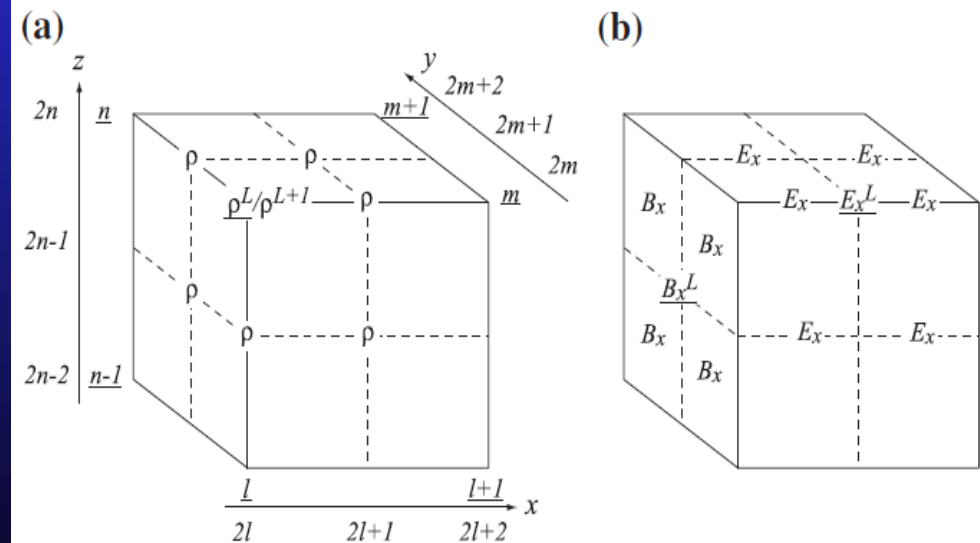
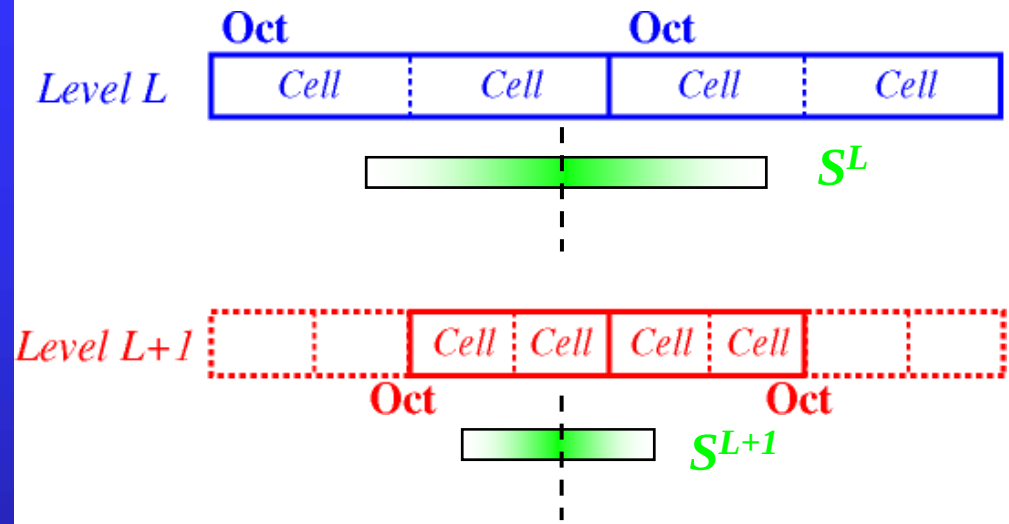
$$\begin{aligned} \frac{\partial \vec{B}^L}{\partial t} &= -\nabla^L \times \vec{E}^L \\ \frac{\partial \vec{E}^L}{\partial t} &= \nabla^L \times \vec{B}^L - \vec{j}^L \\ \frac{\partial \rho^L}{\partial t} + \nabla^L \cdot \vec{j}^L &= 0 \end{aligned}$$

$$\rho^L(\vec{X}^L) = \sum_s \sum_j q_{sj} S^L(\vec{x}_{sj} - \vec{X}^L)$$

$$\rho^L(\vec{X}^L) = \sum_i \rho^{L+1}(\vec{X}_i^{L+1}) S^L(\vec{X}_i^{L+1} - \vec{X}^L)$$

**Level L+1**

$$\begin{aligned} \frac{\partial \vec{B}^{L+1}}{\partial t} &= -\nabla^{L+1} \times \vec{E}^{L+1} \\ \frac{\partial \vec{E}^{L+1}}{\partial t} &= \nabla^{L+1} \times \vec{B}^{L+1} - \vec{j}^{L+1} \\ \frac{\partial \rho^{L+1}}{\partial t} + \nabla^{L+1} \cdot \vec{j}^{L+1} &= 0 \end{aligned}$$



# AMRの適用: 階層間同期

## Fine-to-Coarse Operation

$$\rho^L(\vec{X}^L) = \sum_i \rho_i^{L+1} S^L(\vec{X}_i^{L+1} - \vec{X}^L)$$



$$\frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{B} - \vec{j} \text{ を保存させる}$$

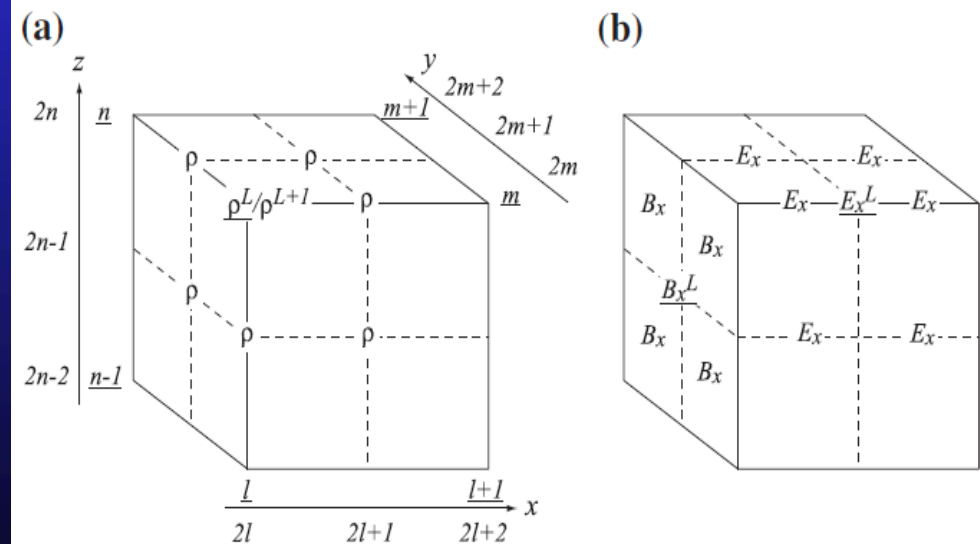
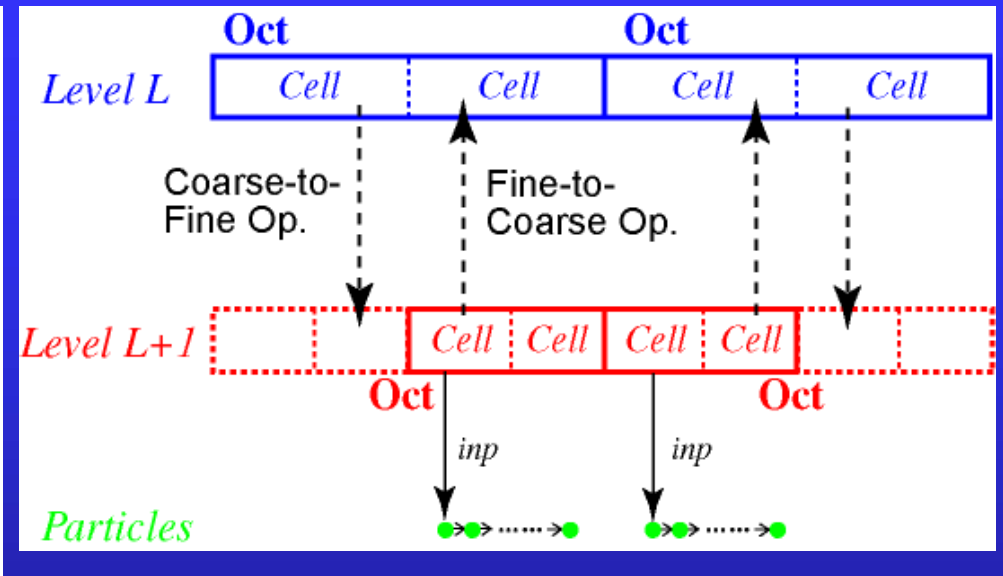
$$\vec{B}^L(\vec{X}^L) = \sum_i \vec{B}_i^{L+1} S^L(\vec{X}_i^{L+1} - \vec{X}^L)$$

## Coarse-to-Fine Operation

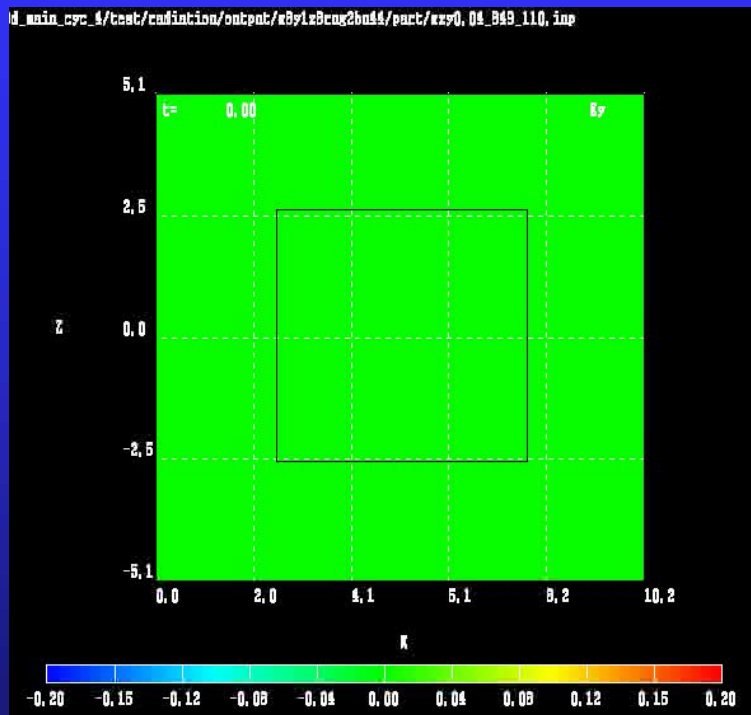
$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}$$

ファラデーの式を満たすように電磁フラックスを保存させる。

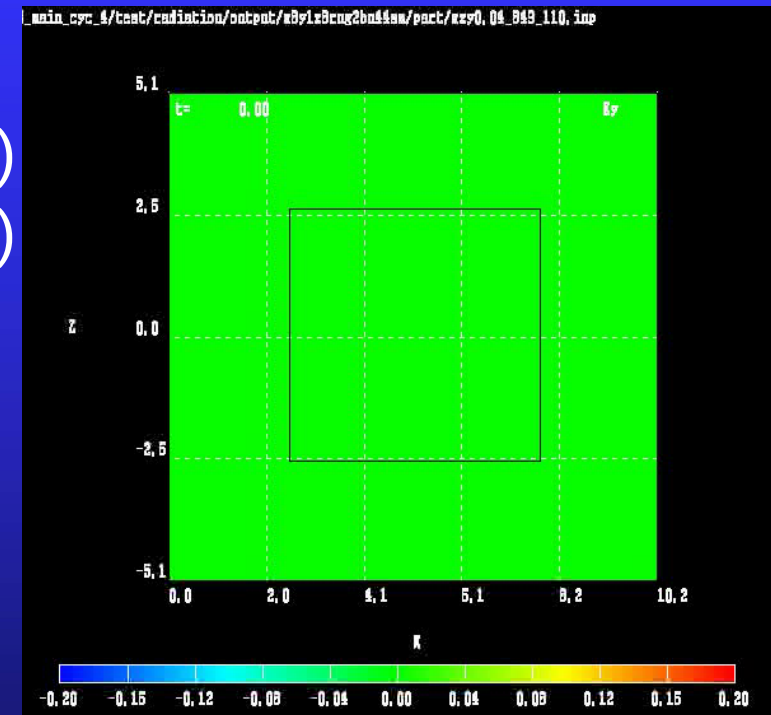
e.g.,  $\nabla^{L+1} \times \vec{E}^{L+1} = \nabla^L \times \vec{E}^L$   
on cell surfaces



# AMRの適用: 電磁場のスムージング

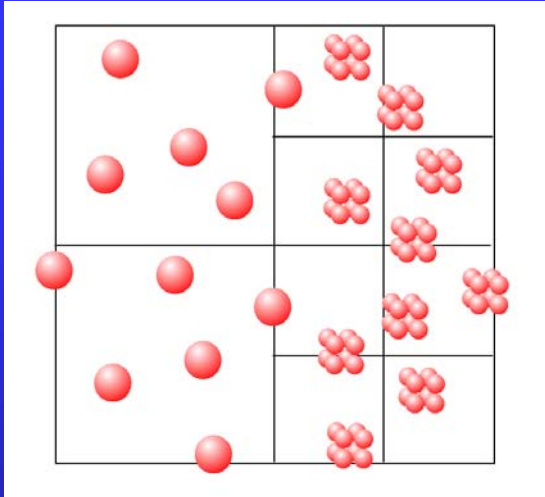


$$E_{SM} = f_{SM}(E)$$
$$B_{SM} = f_{SM}(B)$$



$$A_{SM,j} = f_{SM}(A_j) = \frac{\alpha A_{j-1} + A_j + \alpha A_{j+1}}{1 + 2\alpha} \quad (\alpha = 0.002)$$

# 超粒子の分割・統合 [Lapenta, JCP, 2002]

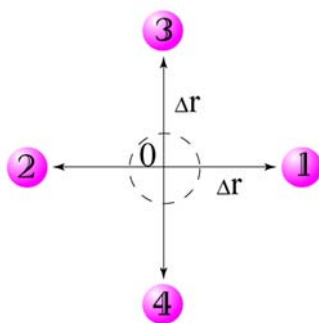


$$\frac{d\vec{v}_s}{dt} = \frac{q_s}{m_s} (\vec{E} + \vec{v}_s \times \vec{B})$$

分割前後で保存させる

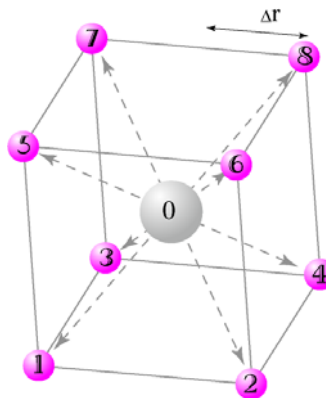
格子点上のモーメント ( $\rho_c, J$ )、全電荷・質量 ( $\Sigma\rho_c, \Sigma m$ )、粒子の全エネルギー ( $\Sigma mv^2/2$ )、粒子の分布関数 ( $f(v)$ )

2次元の場合



$$\vec{v}_j = \vec{v}_0 \quad (j = 1, 2, \dots, 8)$$

3次元の場合

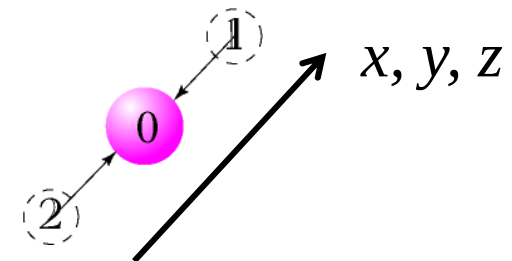


$$q_1 = q_2; m_1 = m_2$$

$$\mathbf{x}_0 = (\mathbf{x}_1 + \mathbf{x}_2)/2$$

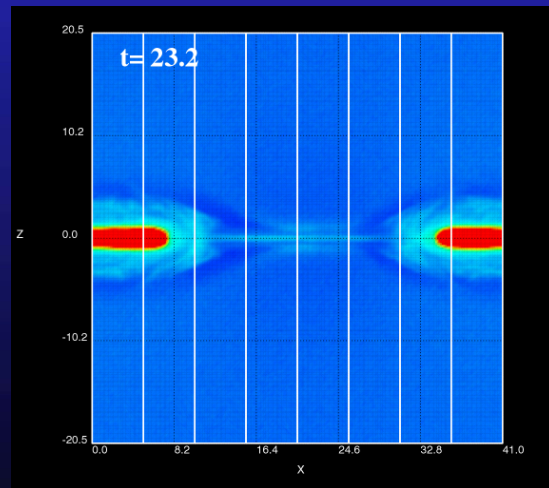
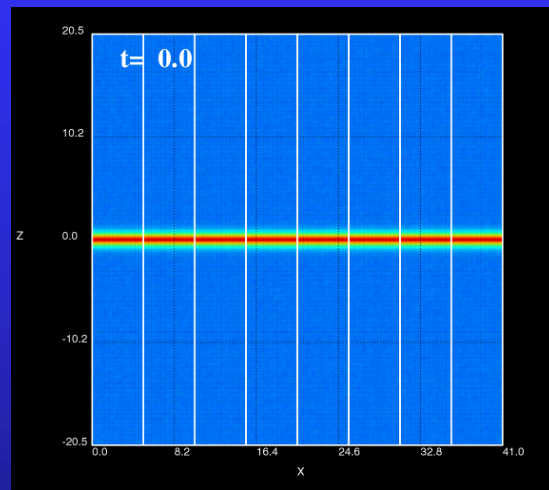
$$\mathbf{V}_0 = (\mathbf{V}_1 + \mathbf{V}_2)/2$$

$$q_0 = q_1 + q_2; m_0 = m_1 + m_2$$

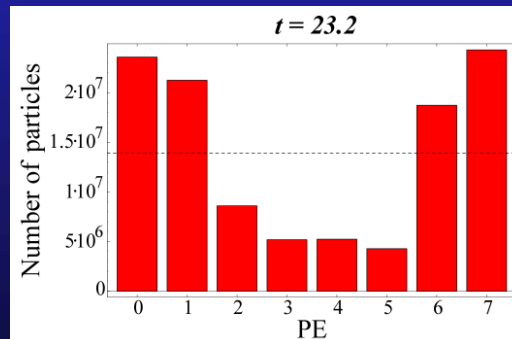
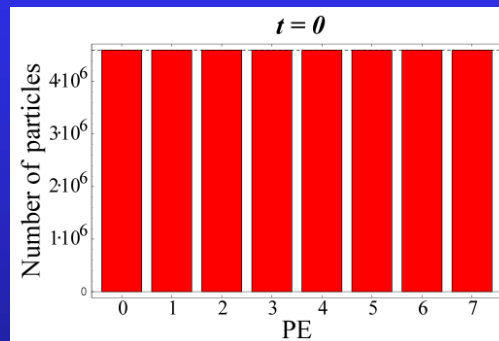


# 分散並列化: 問題点

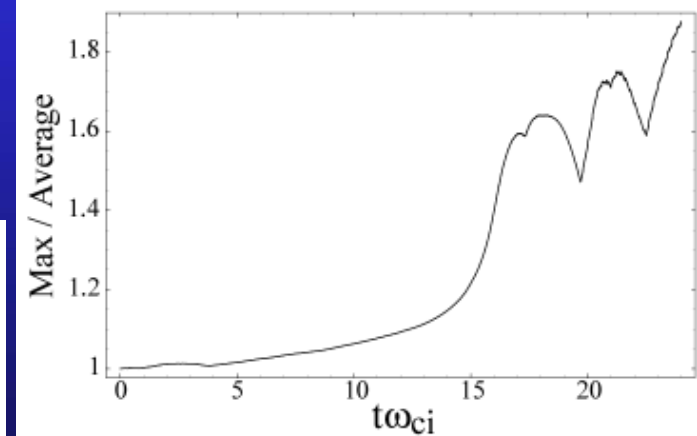
## 8ノード並列の例



PICシミュレーションでは全計算時間の99%以上を粒子に関する計算に要する。

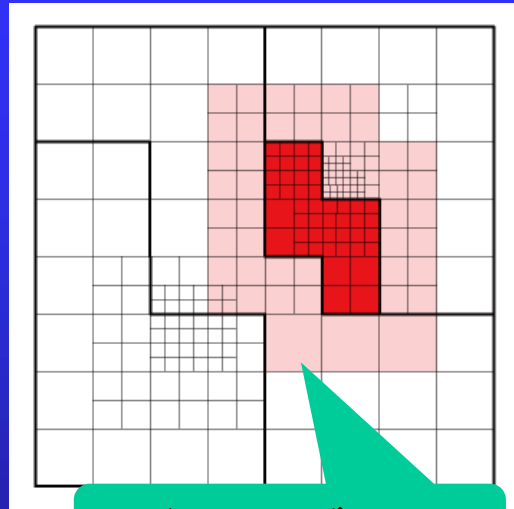


ノード当たり粒子数の不均衡性

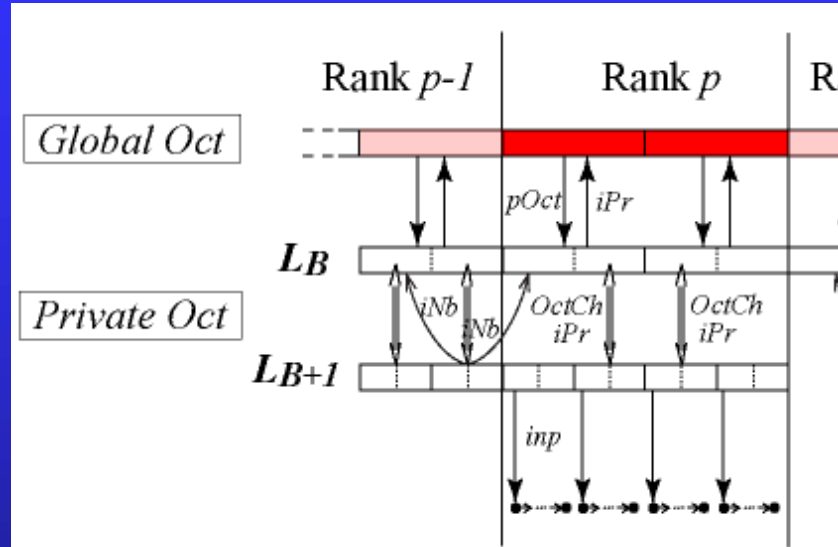


# 適合ブロック法

## バッファ領域とプライベートオクトの生成



バッファ領域



### グローバルオクト

すべてのノードでシェア

$grank$ : ノード番号

$gOctNb$ : 隣接するオクト



### プライベートオクト

各ノードで個別に設定

$rank$ : ノード番号

$iNb$ : 隣接するオクトの親セル

$iPr$ : 親セル

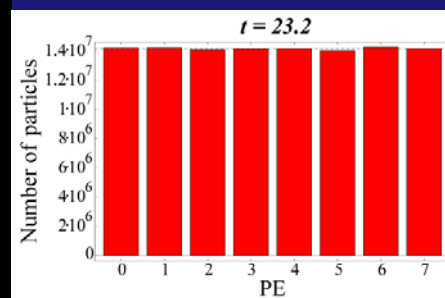
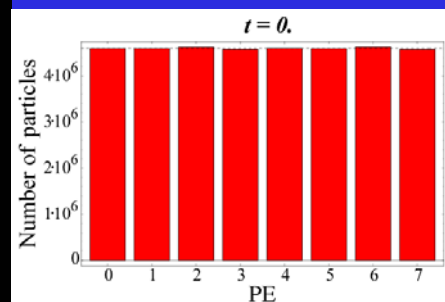
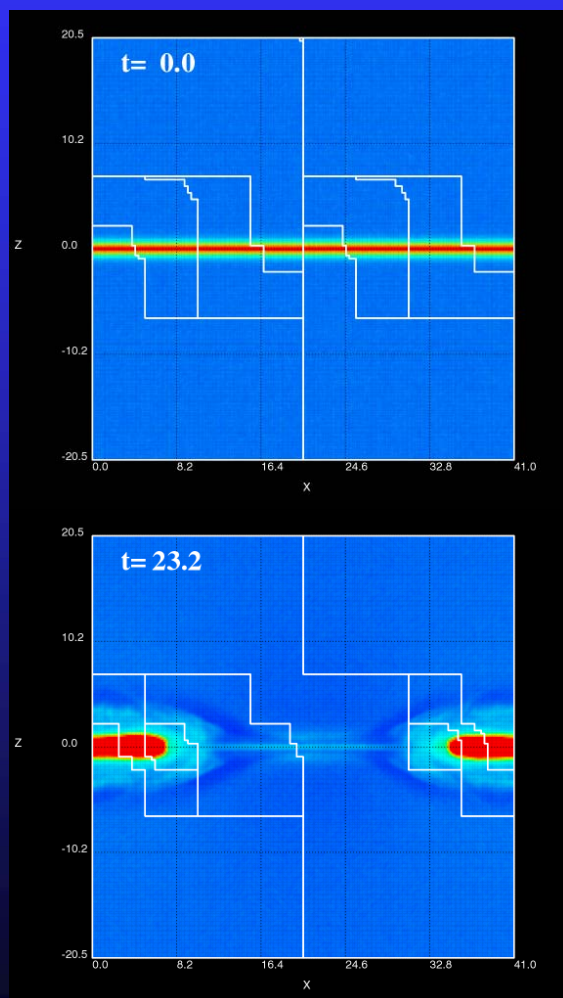
$OctCh$ : 子オクト

...

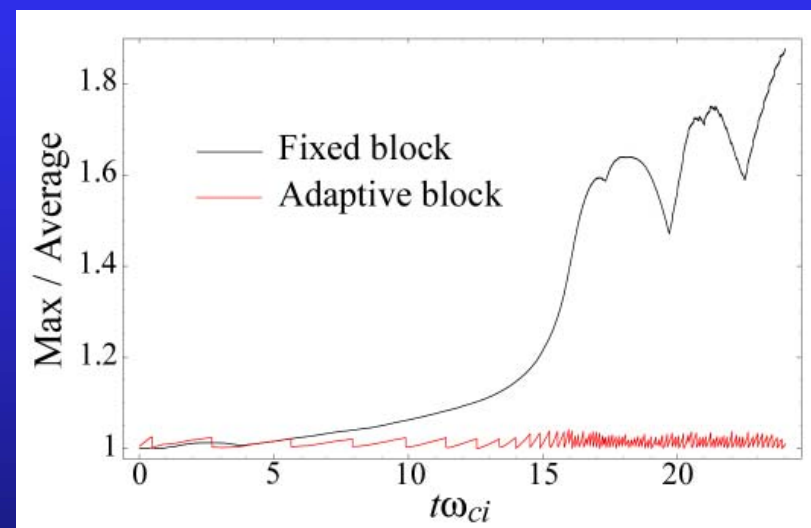
物理データ

# 適合ブロックを用いたシミュレーション

## 8ノード並列の例



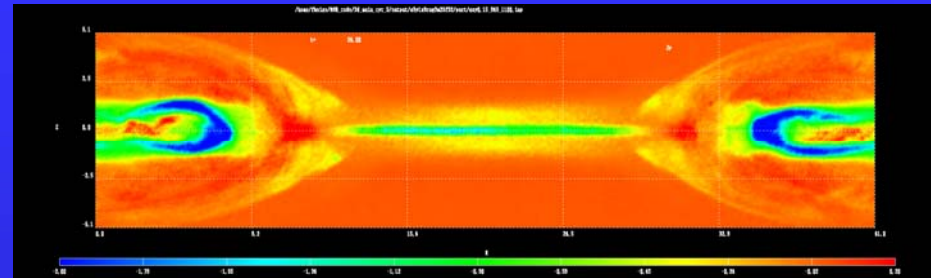
## ノード当たり粒子数の不均衡性



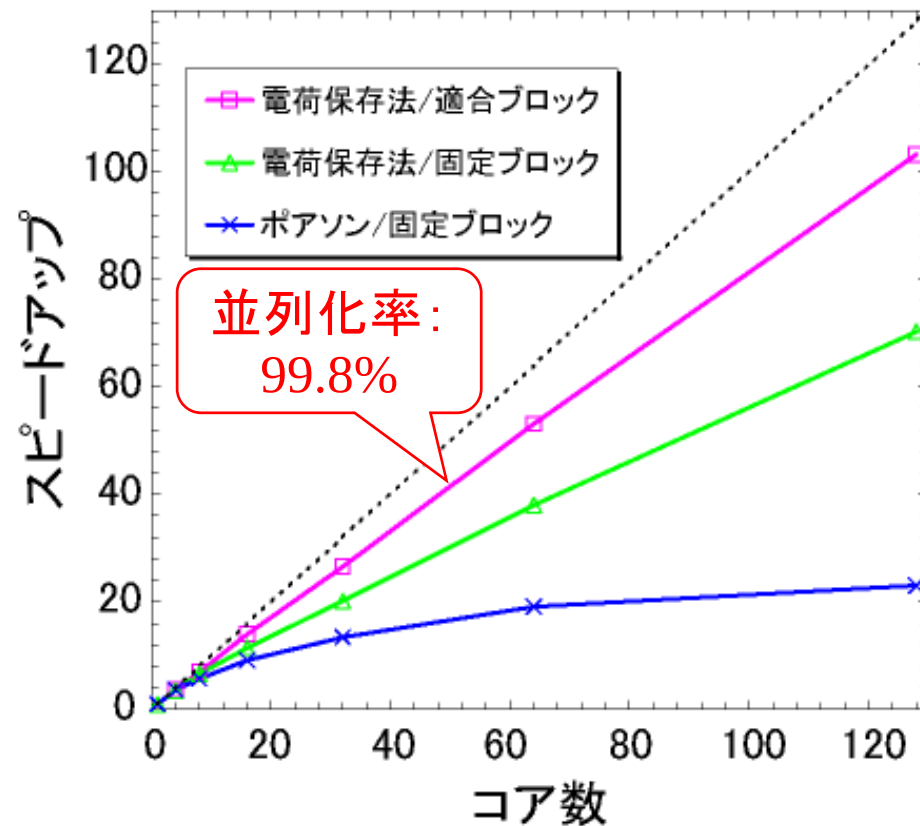
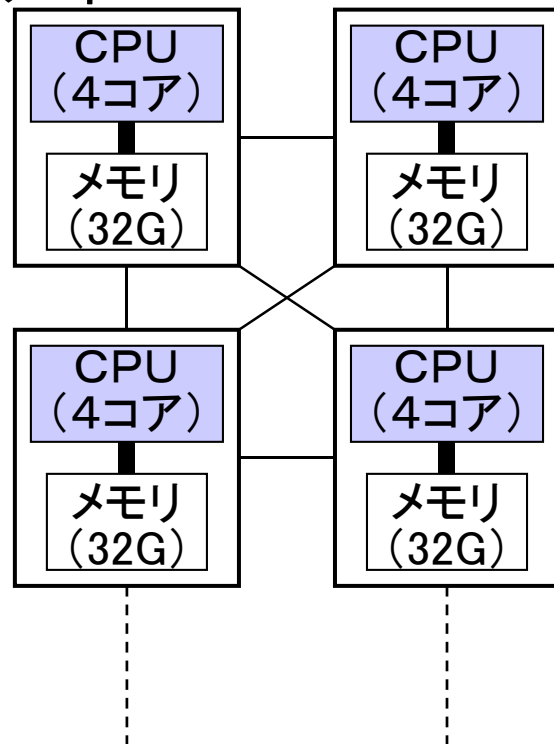
# 超並列AMR-PICコードの性能

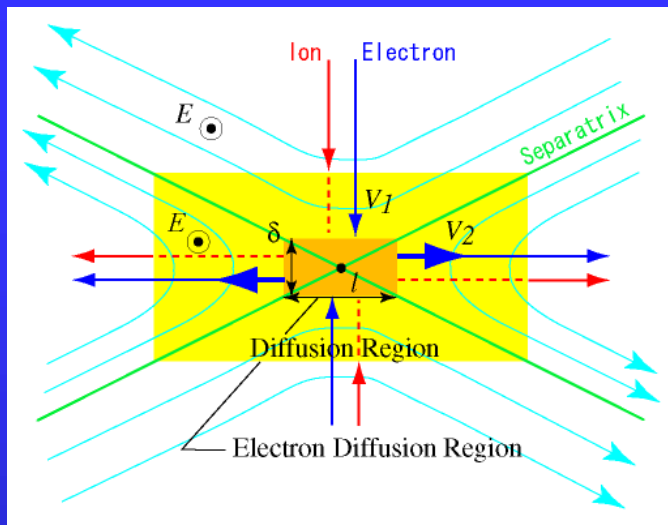
Fujitsu FX1

(名大情報基盤センター)



ノード



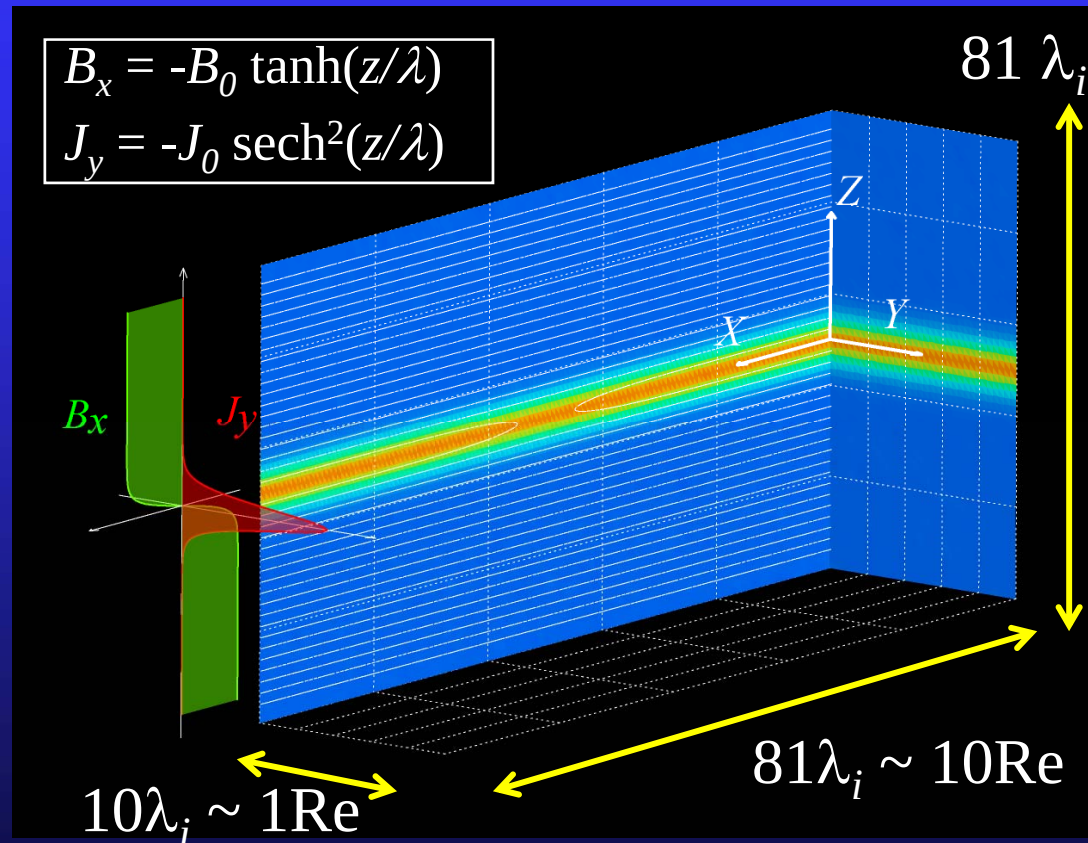


最近の研究成果

(磁気X線における磁気拡散機構)

# Simulation Setup

Fujitsu FX1, 1024 cores



$$m_i/m_e = 100$$

Max resolution:  
 $4096 \times 512 \times 4096 \sim 10^{10}$

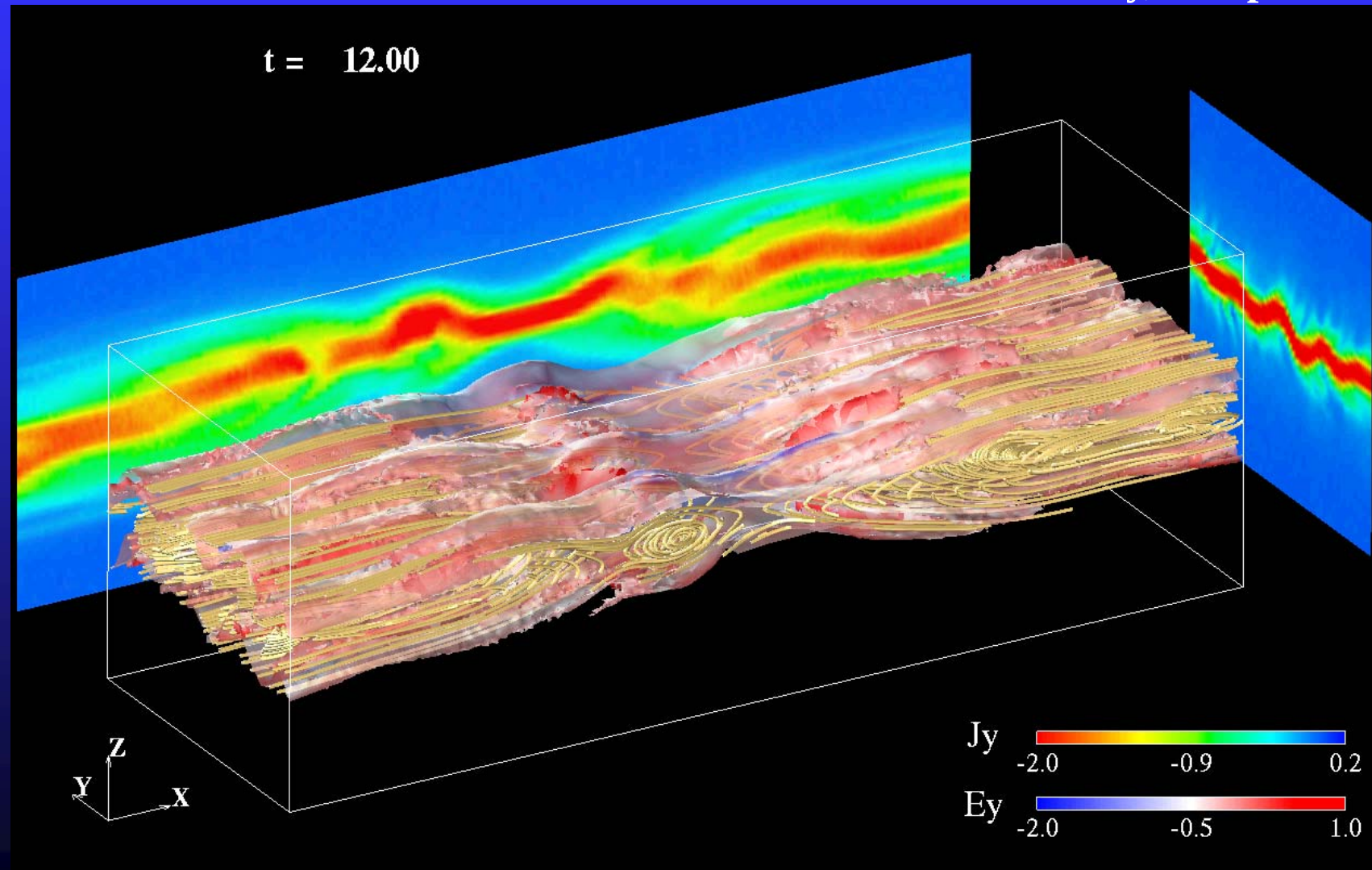
Max number of particles  
Ion + Electron  $\sim 10^{11}$

Max memory used  $\sim 6\text{TB}$

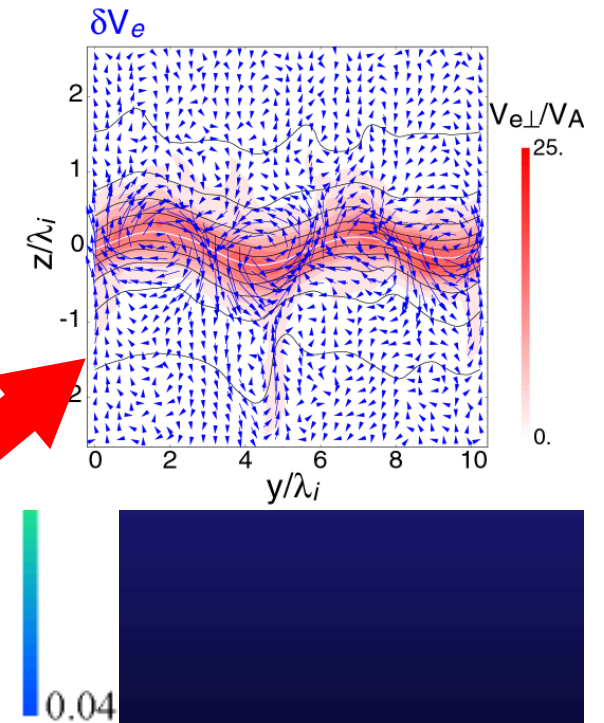
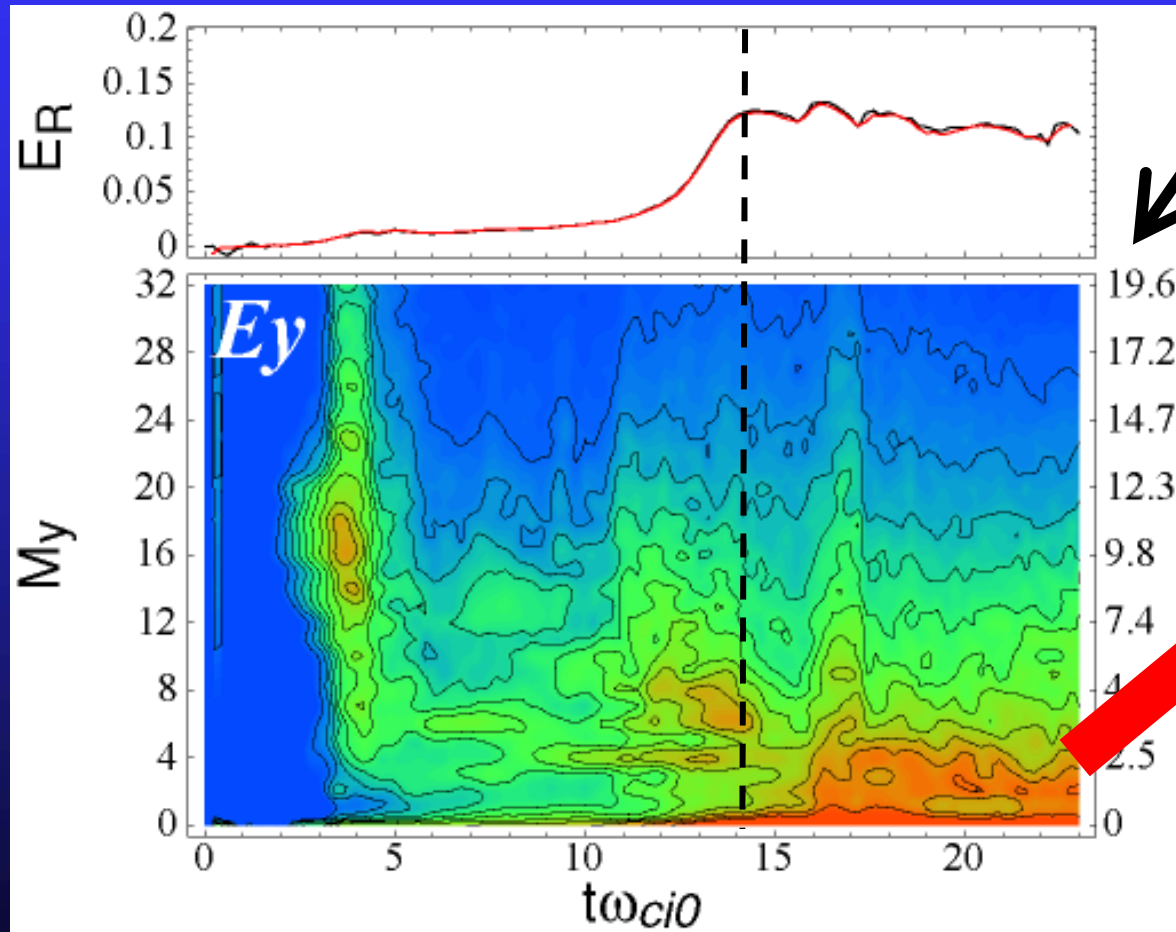
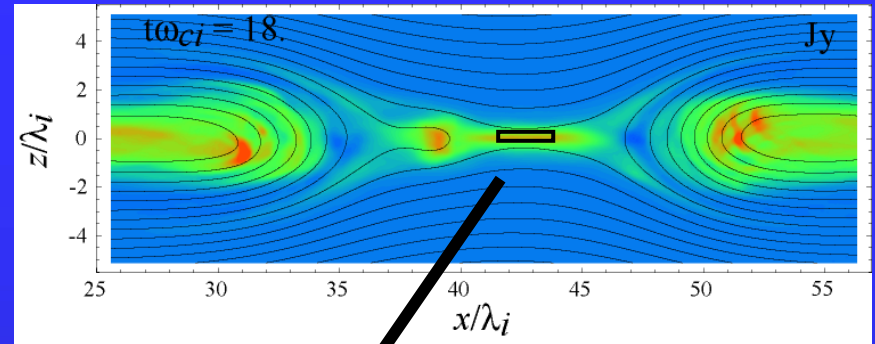
# Time Evolution of the Current Sheet

Surface:  $|J|$ , Line: Field line

Color on the surface:  $E_y$ , Cut plane:  $J_y$



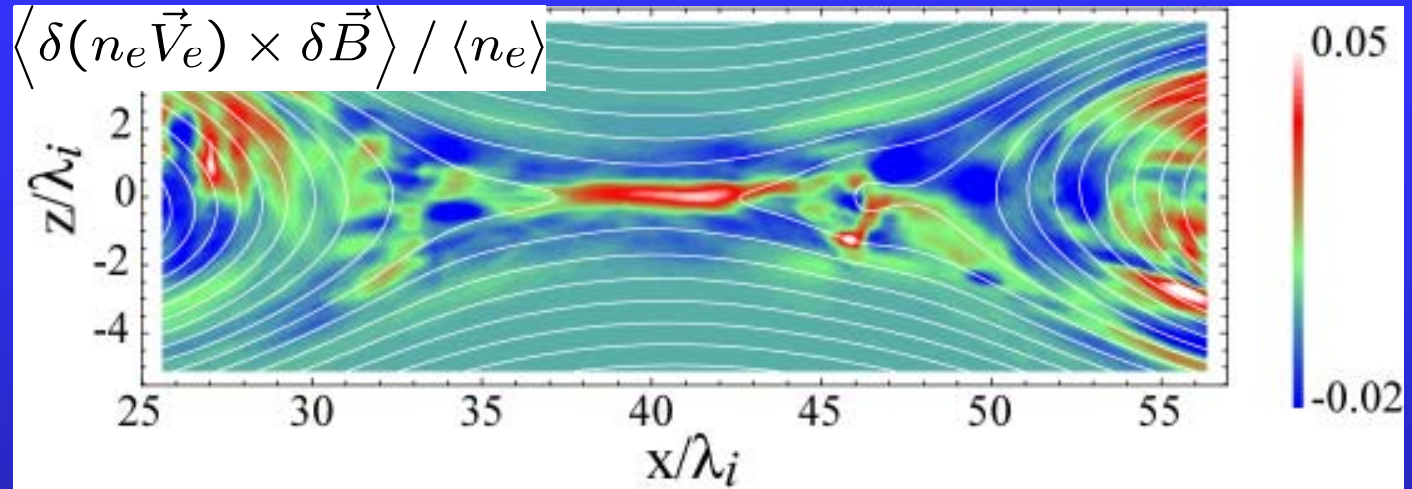
# Wave Activity



# Anomalous Momentum Transport

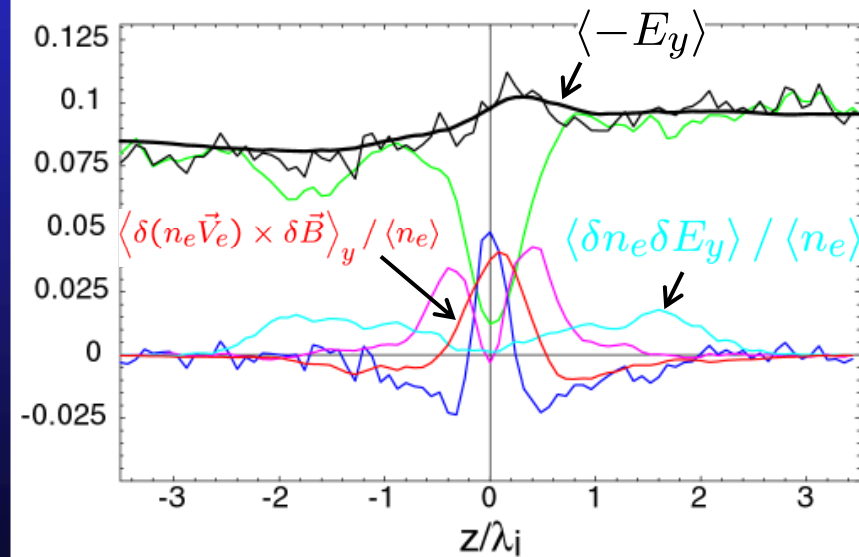
$$A = \langle A \rangle + \delta A$$

$$\left( \langle \cdot \rangle = \frac{1}{L_y} \int_0^{L_y} \cdot dy \right)$$

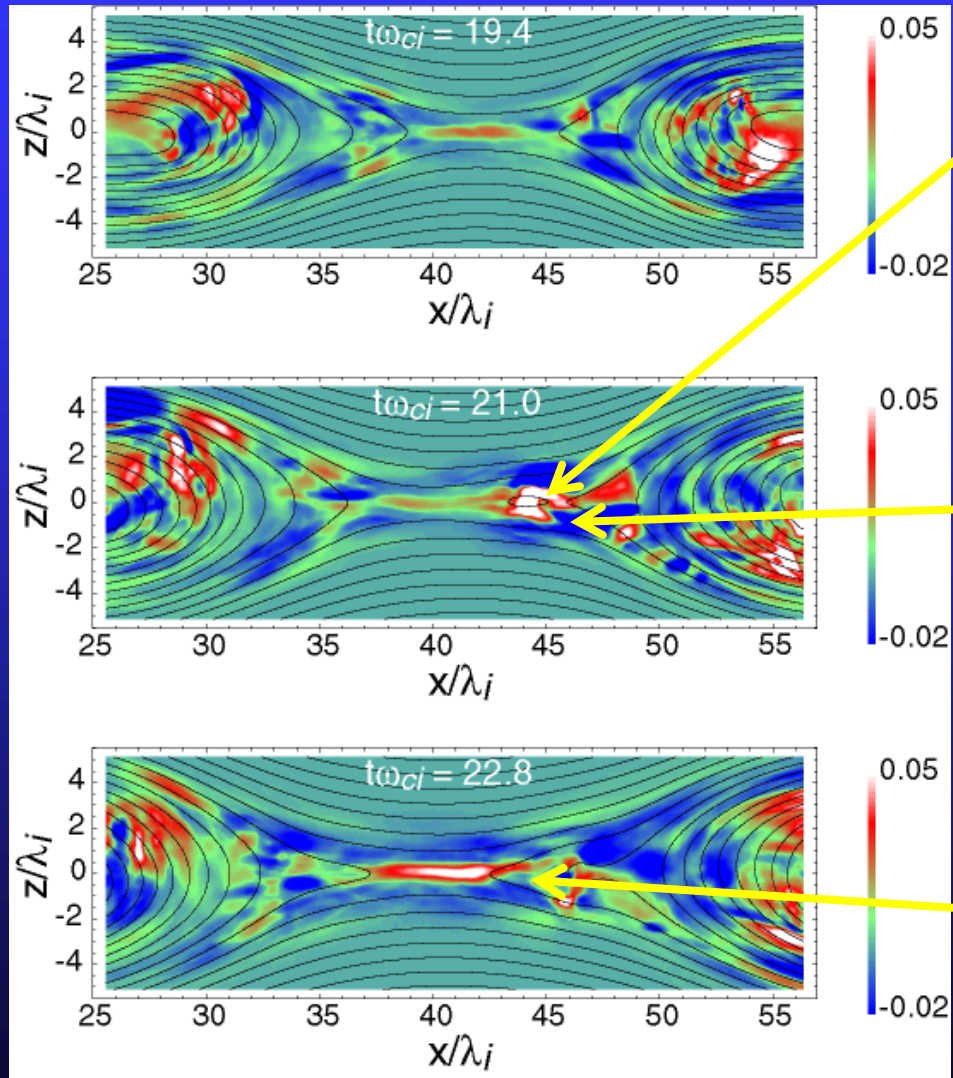


$$\begin{aligned} \langle -E_y \rangle = & \frac{1}{\langle n_e \rangle} \left( \langle n_e \vec{V}_e \rangle \times \langle \vec{B} \rangle \right)_y \\ & + \frac{1}{e \langle n_e \rangle} \langle \nabla \cdot \vec{P}_e \rangle_y \\ & + \frac{m_e}{e \langle n_e \rangle} \left\langle \frac{\partial V_{ey}}{\partial t} + \vec{V}_e \cdot \nabla V_{ey} \right\rangle \\ & + \frac{1}{\langle n_e \rangle} \langle \delta n_e \delta E_y \rangle \\ & + \frac{1}{\langle n_e \rangle} \langle \delta(n_e \vec{V}_e) \times \delta \vec{B} \rangle_y \end{aligned}$$

Anomalous effects



# Plasmoid-Induced Turbulence



Plasmoid formation

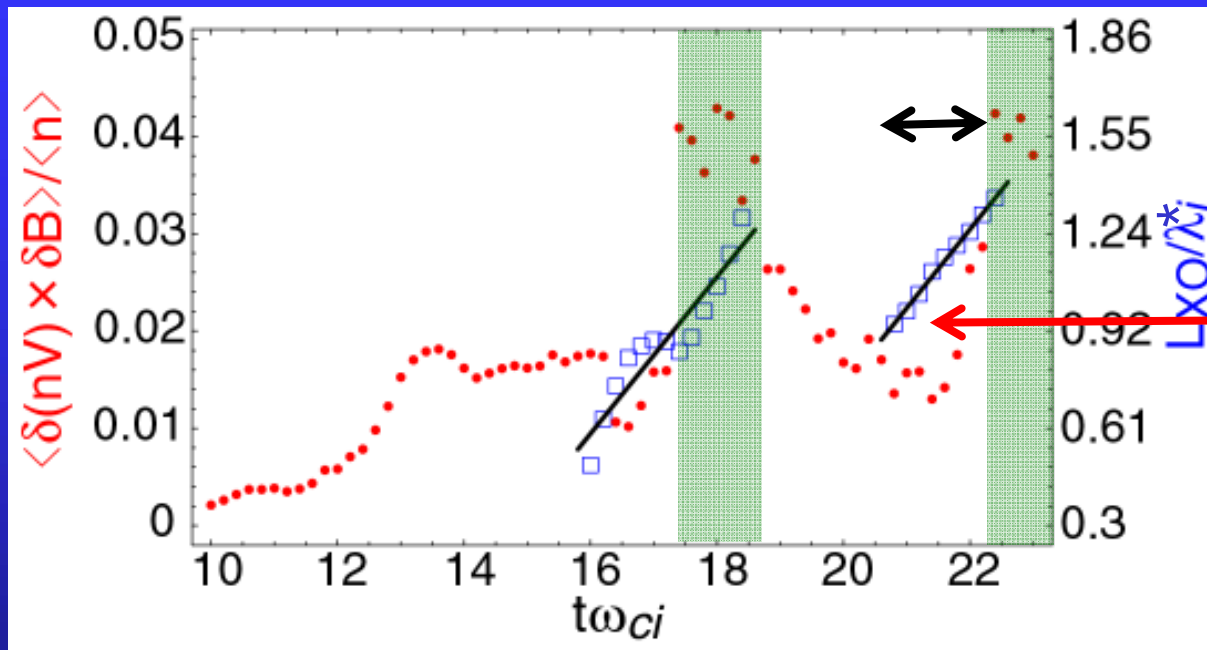
Wave  
amplification

Local turbulence  
enhancement

Propagation along  
the field line

Intensified turbulence  
at the x-line

# Plasmoid-Induced Turbulence



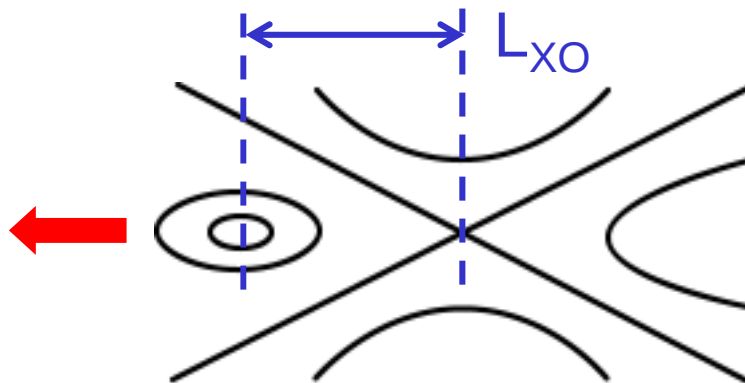
$$\Delta t = 1.6 \omega_{ci}^{-1}$$

$$L_{XO} = 0.95 \lambda_i^*$$



Information  
propagates at

$$V_p \sim V_A^* \\ (B^* = 0.5B_0)$$



## まとめ

### AMR-PICコードの必要性

1. 空間非一様性が強く、かつ、運動論効果が重要な現象
2. 国内汎用スパコンを用いて世界最大規模の計算を実現

### AMR-PICコード開発のポイント

1. 細分化レベル間の同期(場と粒子のconsistency保持)
2. 電磁場のスムージング(短波長モードの除去)
3. データ構造(ツリー構造が便利)
4. 粒子分割・統合(格子当たりの粒子数調節)

## 今後の課題

- 開放境界条件
- 逐次計算効率の向上  
(粒子データのソーティング)
- 数千コア並列以上の場合における最適化
- ソースコードの公開
- マニュアルの作成
- ユーザの拡大

# 適合ブロック法(その1)

(ブロック: 並列計算において各ノードが分担する計算領域)

全計算領域の基底オクト(グローバルオクト)を最適な順番に並べ替える。

出発点から数えて( $x_{oct}, y_{oct}, z_{oct}$ )番目にあるオクトの順番

$$D_m = \sum_{l=0}^{n_x+n_y+n_z} N_l d_l$$

$$x_{oct} = \sum_{i=0}^{n_x} b_{xi} d_{xi}; \quad y_{oct} = \sum_{j=0}^{n_y} b_{yj} d_{yj}; \quad z_{oct} = \sum_{k=0}^{n_z} b_{zk} d_{zk}$$

$$b_{xi} = 2^i; \quad b_{yj} = 2^j; \quad b_{z,n_z} = N_z, \quad b_{z,k-1} = \left\lceil \frac{b_{zk}}{3} \right\rceil, \quad b_{z,0} = 1$$

$$d_0 = d_{z0}$$

$$d_{l+1} = (i_{l+1} - i_l) d_{x,i_{l+1}} + (j_{l+1} - j_l) d_{y,j_{l+1}} + (k_{l+1} - k_l) d_{z,k_{l+1}}$$

$$l = i_l + j_l + k_l$$

$$N_l = b_{xi_l} b_{yj_l} b_{zk_l} \quad (b_{zk_l} \gtrsim b_{xi_l} \gtrsim b_{yj_l})$$

- 初期に1度だけ実施
- この順番に沿って粒子数を数え、各ノード(ブロック)の粒子数が均等になるように基底オクトを分割する。

➡ ブロックの表面積を小さくし、通信量を減らす。

