

**Massively parallel computing of
electromagnetic particle-in-cell model
with adaptive mesh and adaptive block
Part II**

Keizo Fujimoto

Computational Astrophysics Laboratory, RIKEN

Introduction

[XZ 平面]

- テアリングモード

[YZ 平面]

- LHDI
- イオン-イオンキンクモード

$$m_i/m_e \geq 100$$

$m_i/m_e = 256$ の場合

通常のPICコード

粒子数 $\sim 7 \times 10^{11}$

メモリ ~ 35 TByte

AMR-PICコード

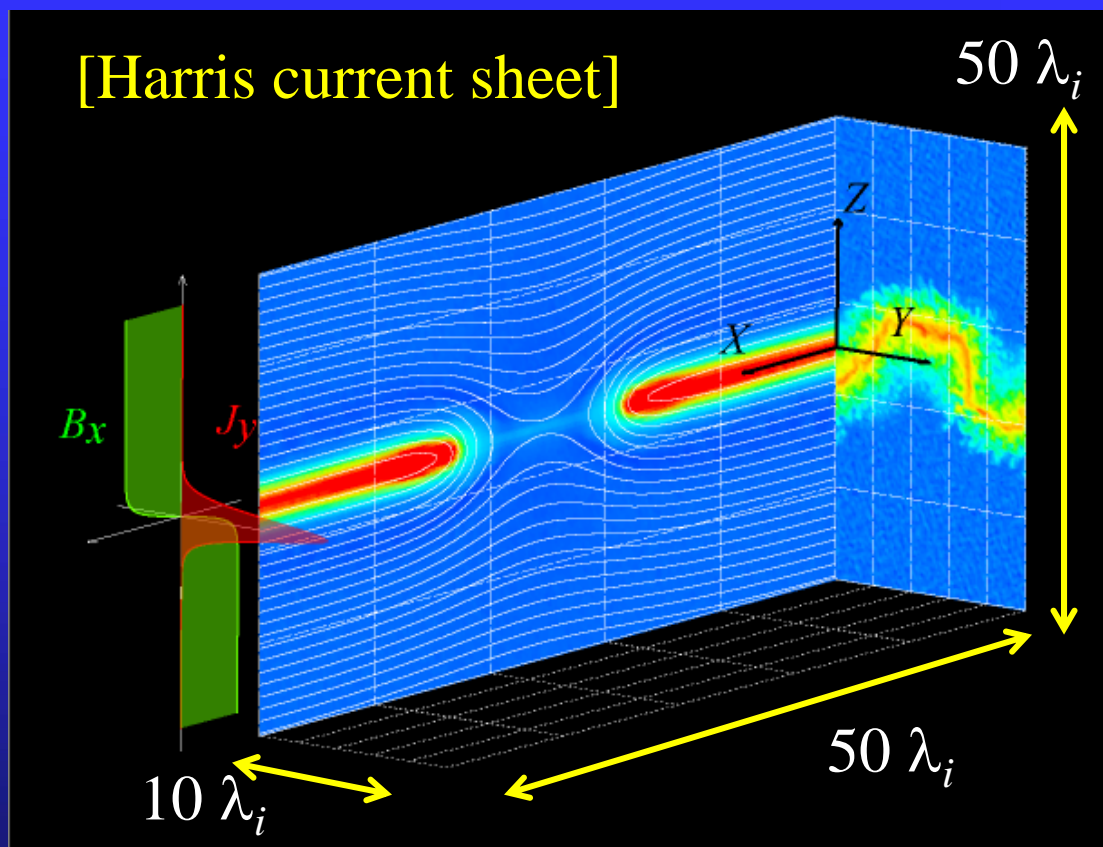
$\sim 7 \times 10^{10}$

~ 3.5 TByte

超並列AMR-PICコード

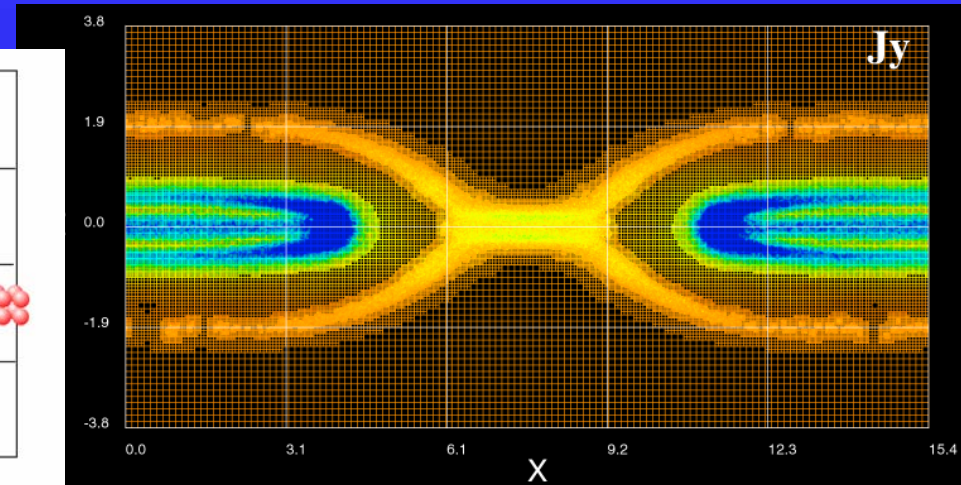
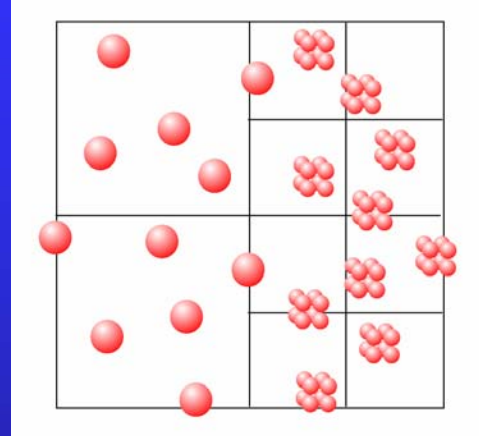
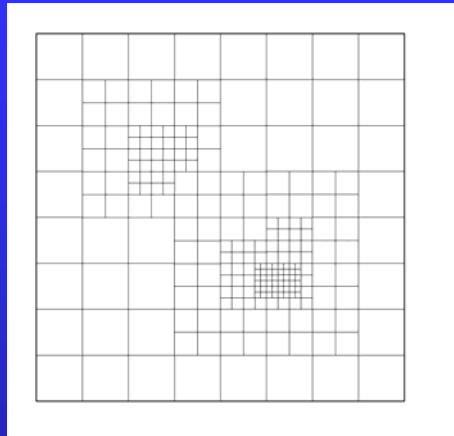
16 GByte/CPU $\times 256$

= 4.1 TByte 使用可能



AMR-PICコード (Adaptive Mesh Refinement)

[Fujimoto & Machida, 2006; Fujimoto & Sydora, 2008]



2003

開発スタート

2006

2D AMR-PIC
コード完成

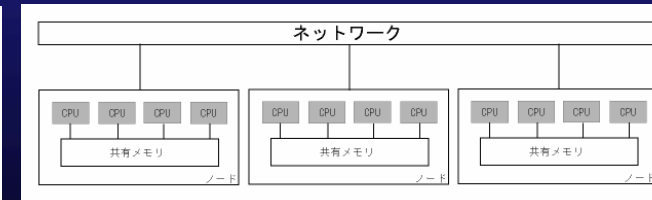
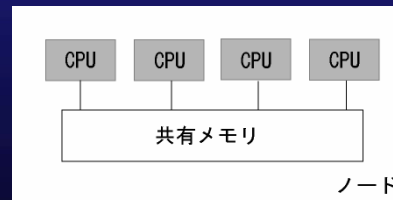
2008

3D AMR-PIC
コード完成

2010

超並列化完了
(電荷保存法、適合ブロック)

ポアソン方程式
ノード内並列



差分方程式

電磁場のスムージング

$$\vec{E}_{SM} = f_{SM}(\vec{E}), \quad \vec{B}_{SM} = f_{SM}(\vec{B})$$

Staggering grid method [Buneman, 1968]

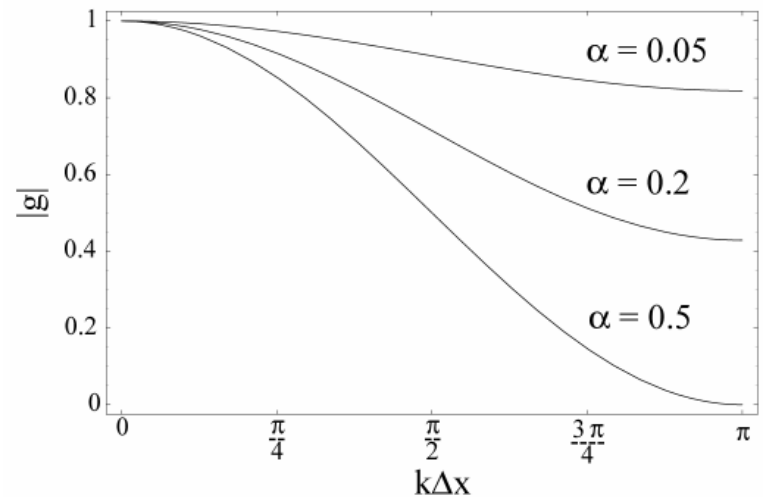
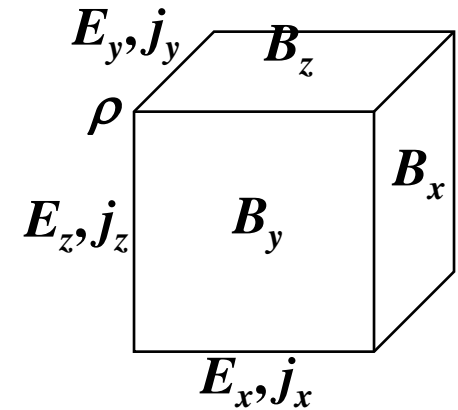
$$\vec{B}^{n+1/2} = \vec{B}_{SM}^{n-1/2} - \Delta t \nabla \times \vec{E}_{SM}^n$$

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

電荷保存法

[Villasenor & Buneman, 1992]

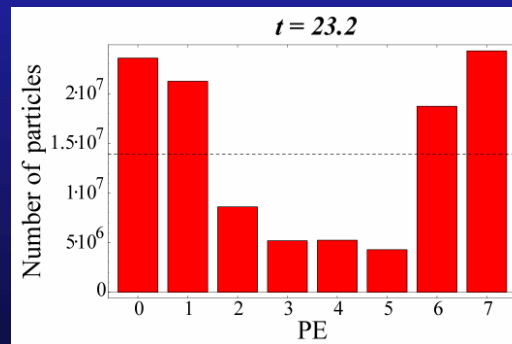
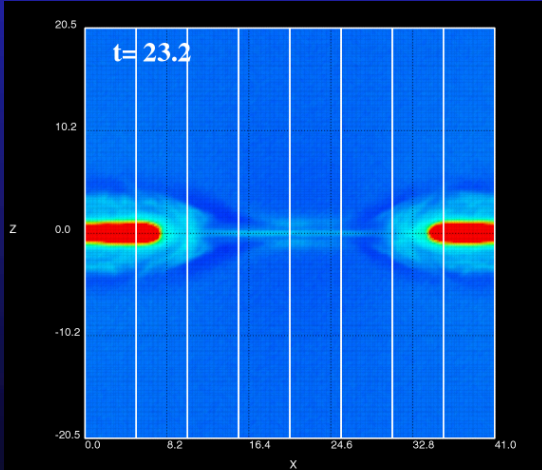
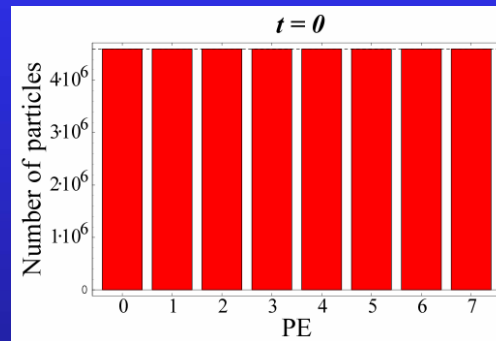
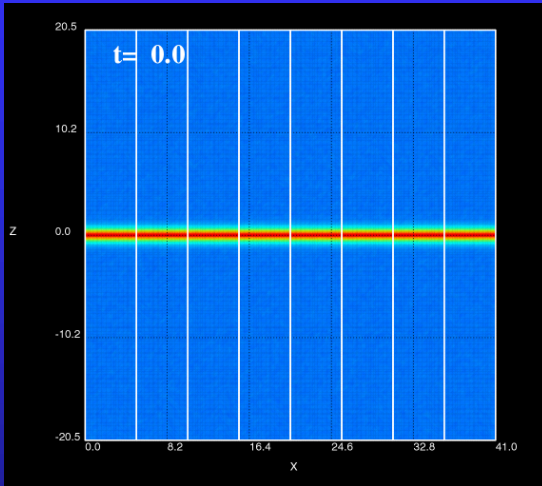
$$f_{SM}(A_j) = \frac{\alpha A_{j-1} + A_j + \alpha A_{j+1}}{1 + 2\alpha}$$
$$(0 \leq \alpha \leq 0.5)$$



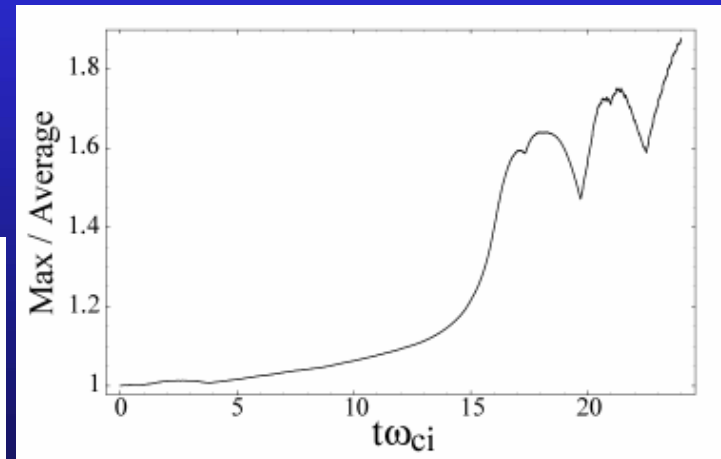
負荷バランス

8ノード並列の例

電磁粒子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(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 $(x_{\text{oct}}, y_{\text{oct}}, z_{\text{oct}})$ 番目にあるオクトの順番 $D_m = \sum_{l=0}^{n_x+n_y+n_z} N_l d_l$

$$x_{\text{oct}} = \sum_{i=0}^{n_x} b_{xi} d_{xi}; \quad y_{\text{oct}} = \sum_{j=0}^{n_y} b_{yj} d_{yj}; \quad z_{\text{oct}} = \sum_{k=0}^{n_z} b_{zk} d_{zk} \quad \left/ \quad \begin{array}{l} l = i_l + j_l + k_l \\ N_l = b_{xi_l} b_{yj_l} b_{zk_l} \quad (b_{zk_l} \gtrsim b_{xi_l} \gtrsim b_{yj_l}) \end{array} \right.$$

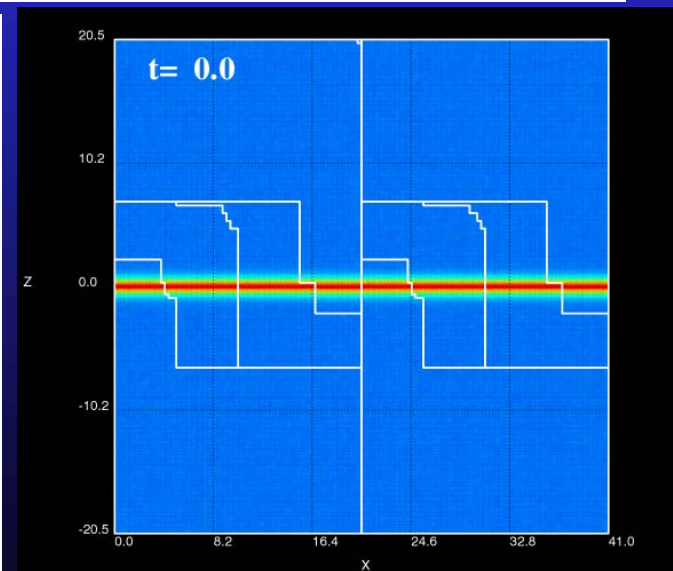
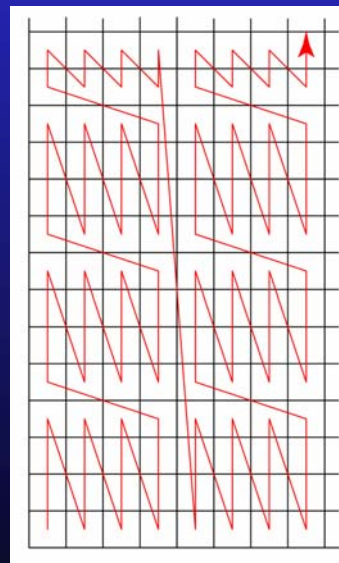
$$b_{xi} = 2^i; \quad b_{yj} = 2^j; \quad b_{z,n_z} = N_z, \quad b_{z,k-1} = \left\lceil \frac{b_{zk}}{3} \right\rceil, \quad b_{z,0} = 1$$

$$d_0 = d_{z0}$$

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

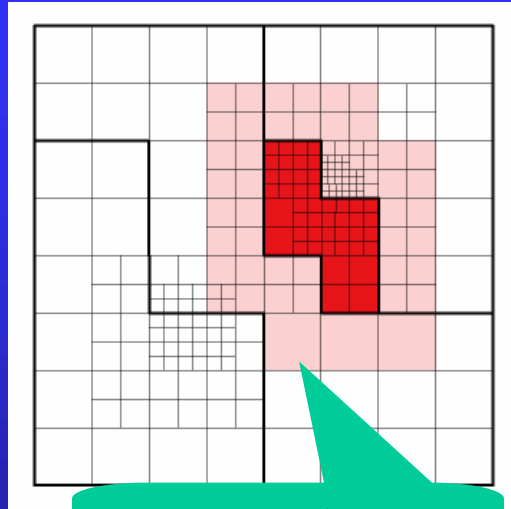
- 初期に1度だけ実施
- この順番に沿って粒子数を数え、各ノードの分担する粒子数が均等になるように基底オクトを分割する。

➡ ブロックの表面積を小さくし、通信量を減らす。

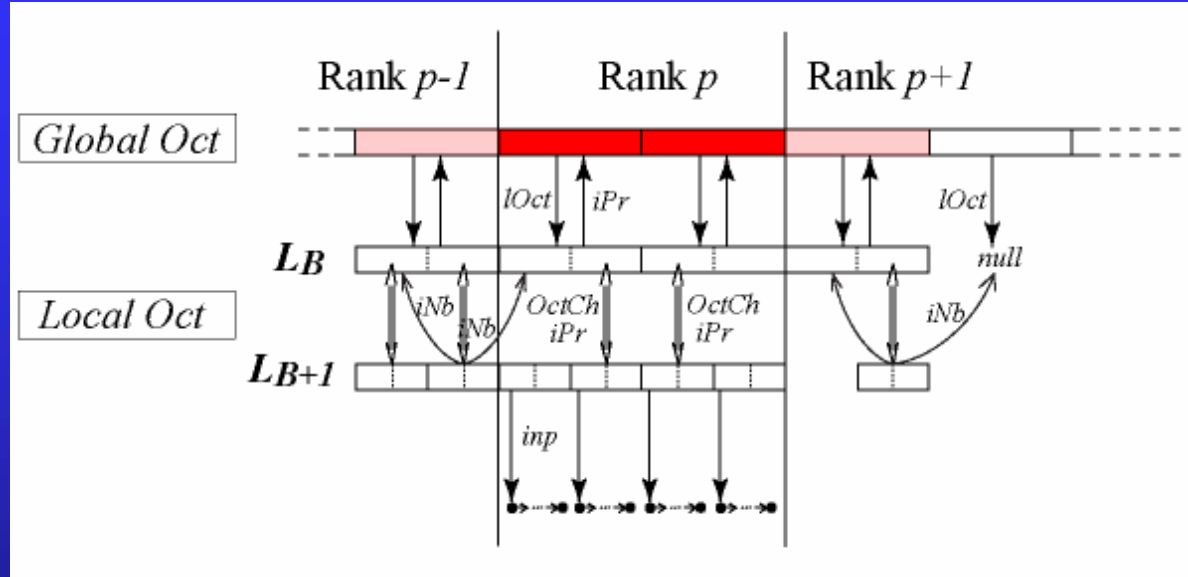


適合ブロック法(その2)

バッファブロックとローカルオクトの生成



バッファブロック



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



ローカルオクト

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

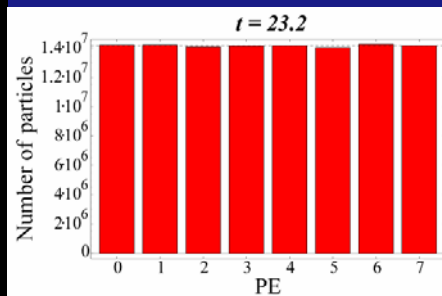
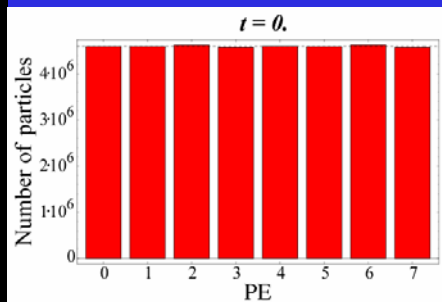
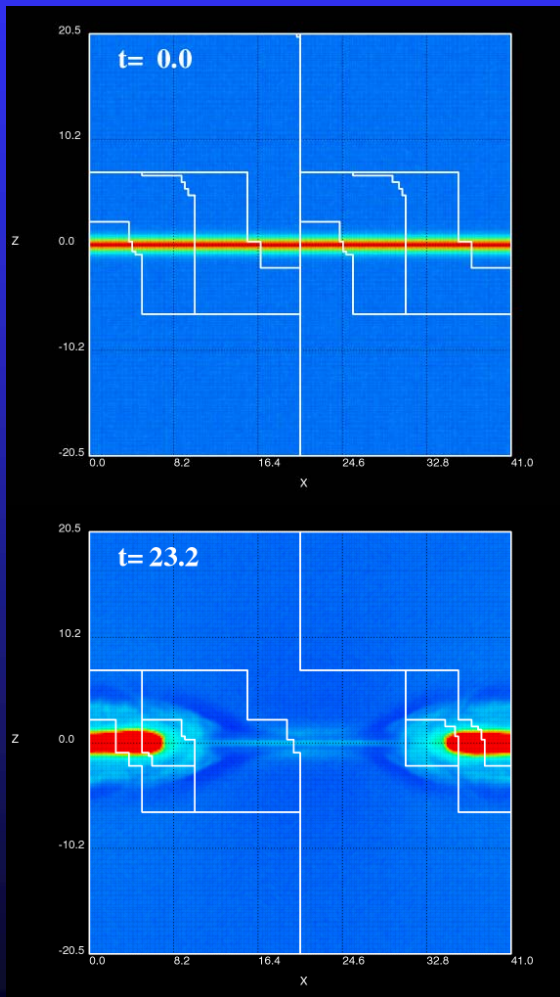
iPr: 親セル

OctCh: 子オクト

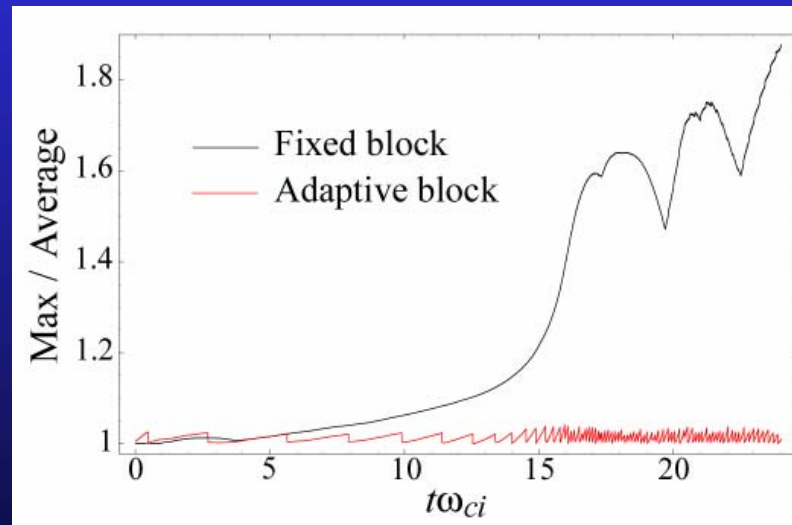
...

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



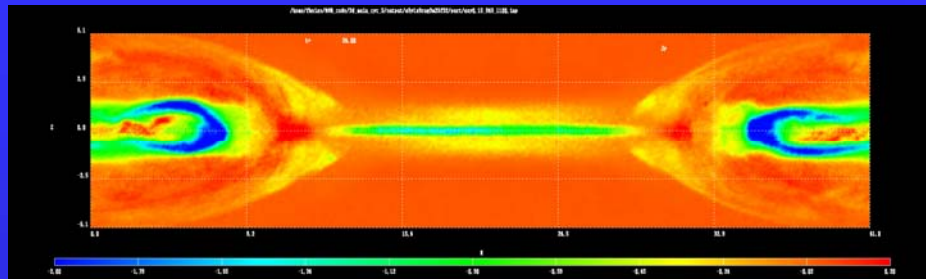
ノード当たり粒子数の不均衡性



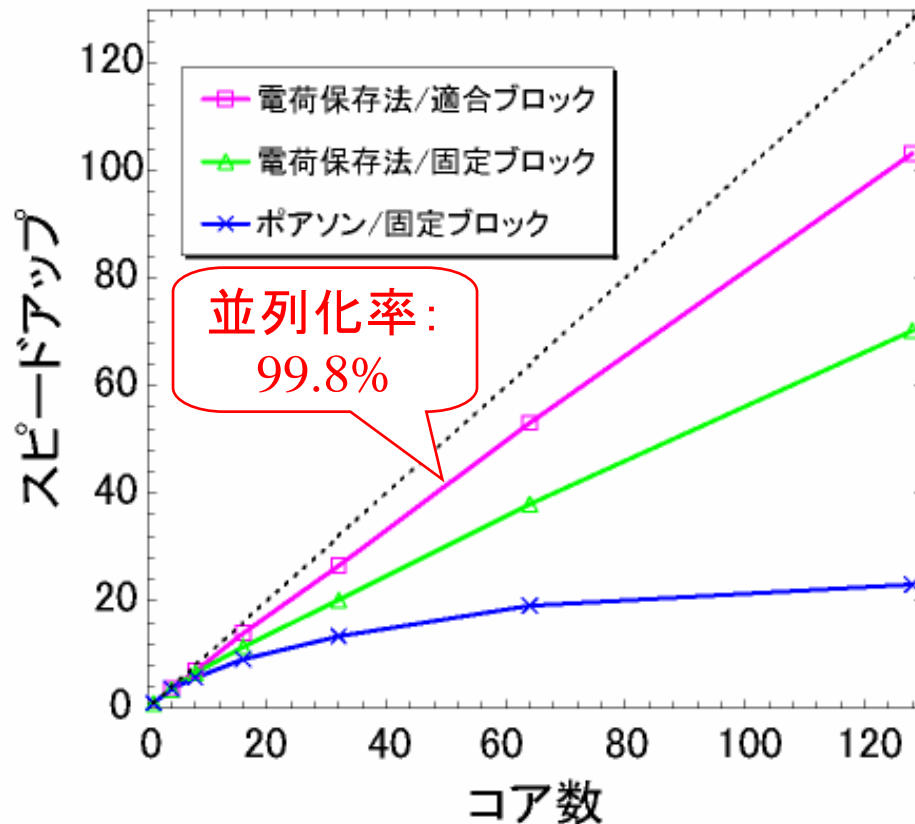
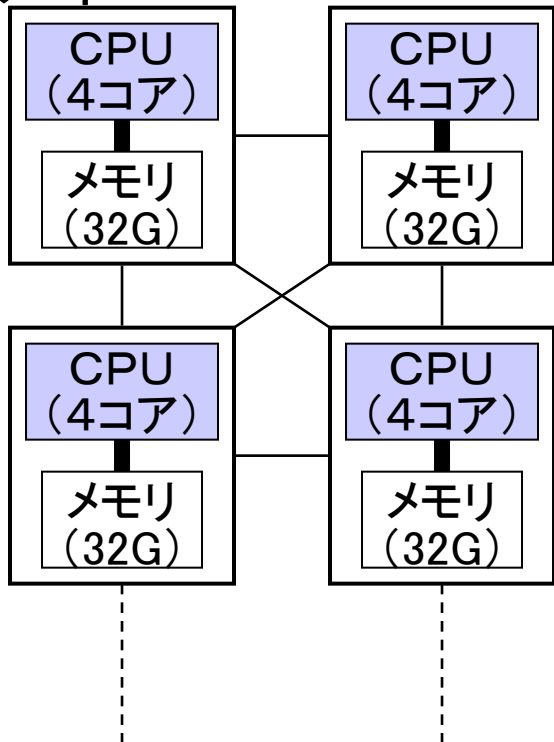
超並列AMR-PICコードの性能

Fujitsu FX1

(名大情報基盤センター)



ノード



まとめ

本研究では、磁気リコネクションの大規模な3次元粒子シミュレーションを実施することを目指して、OpenMPとMPIを組み合わせた超並列AMR-PICコードの開発に成功し、並列化率99.8%を達成した。

主な開発要素

1. 局所的な演算のみで構成され、かつ、AMRが適用可能な差分方程式系の構築。
2. 適合ブロック ⇒ ノード間負荷バランスを維持
 - 基底オクトをMorton序列的な順序に並べ替え、各ノードの粒子数が均等になるようにする。
 - ノード間通信は、バッファブロック内のデータに対してのみおこなう。
 - ほぼすべてのデータがローカルに与えられる(メモリの節約)。

今後の方針

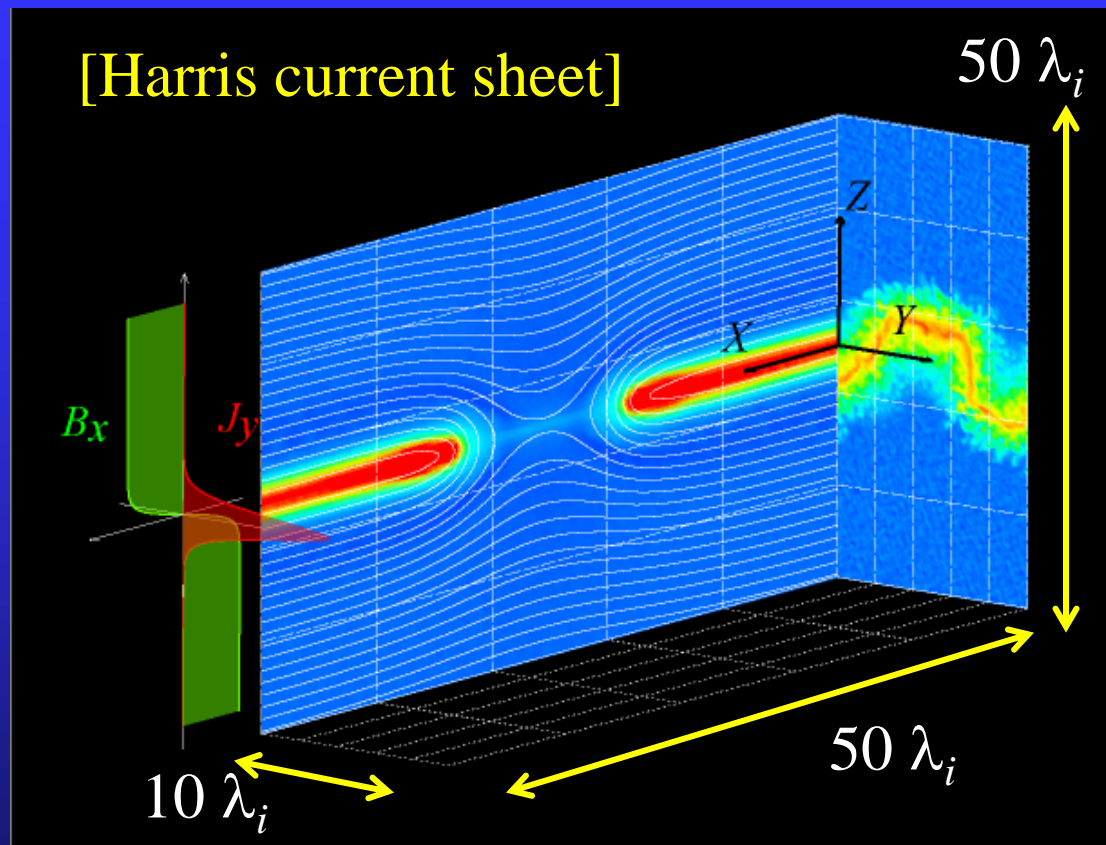
[XZ 平面]

- テアリングモード

[YZ 平面]

- LHDI
- イオン-イオンキンクモード

$$m_i/m_e \geq 100$$



- 磁気拡散機構の解明 ⇒ 異常電気抵抗の発生機構
- 粒子加速・加熱機構の解明