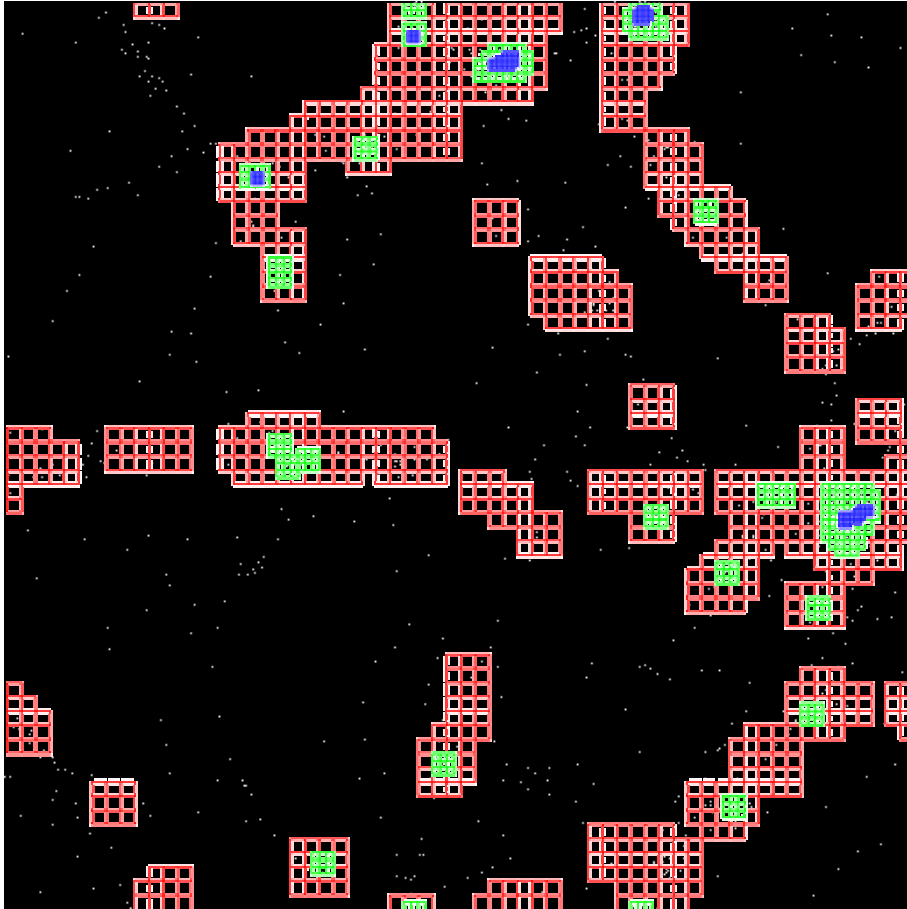


AMR N体コードの現状

矢作 日出樹

(国立天文台 理論研究部)

AMR N 体コード



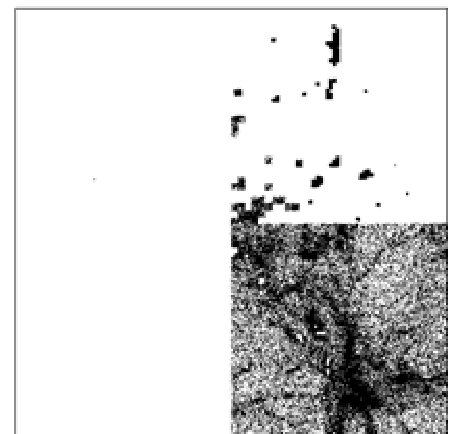
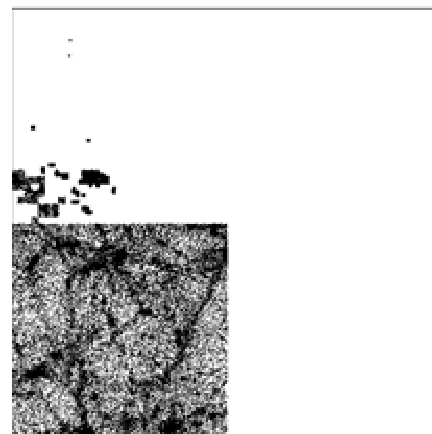
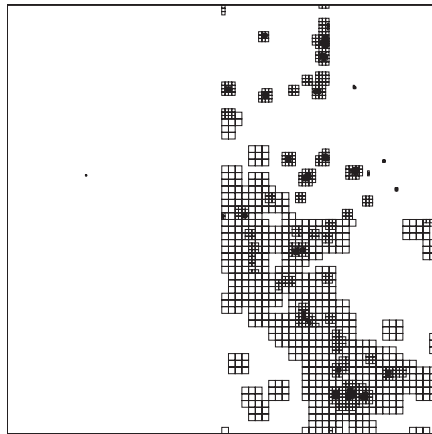
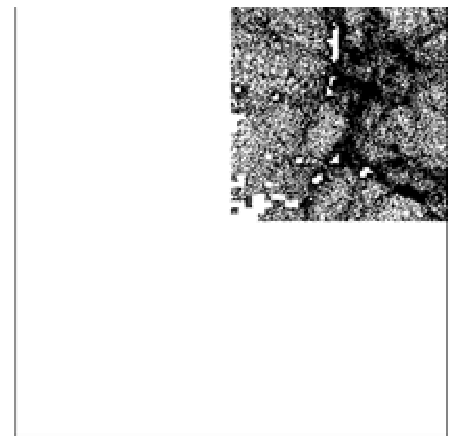
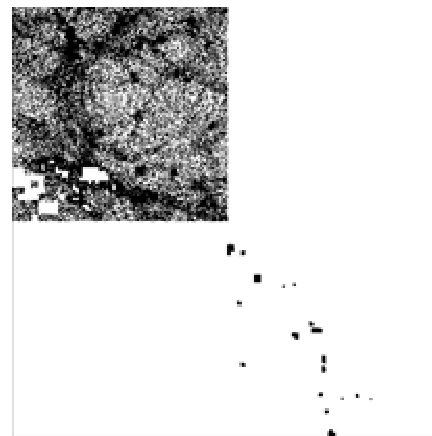
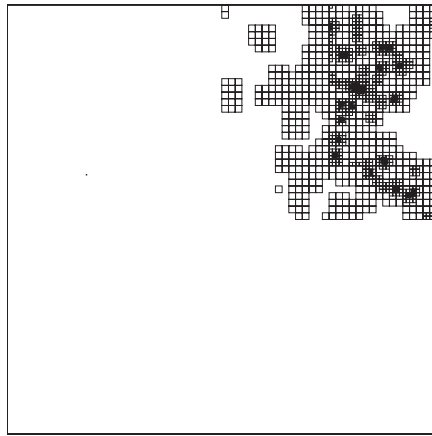
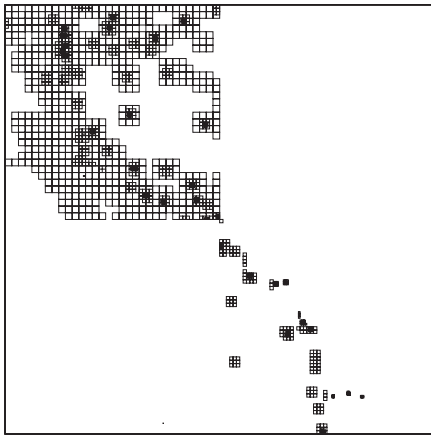
- Adaptive Mesh Refinement (AMR) N 体コード
 - Particle-Mesh法の発展版
 - 通常のPM法は格子間隔より小さい構造を捉えることができない
 - AMR法では、粒子が集中している領域に、階層格子を再帰的に配置することによって解決

(Kravtsov et al. 1997; Knebe et al. 2001; Yahagi & Yoshii 2001; Teyssier 2002)

並列化のためのデータ分割 I

- 階層格子のデータ分割
 - 各階層毎に階層格子をMorton順序(再帰的N型順序)で整列し、CPU数で分割する。
 - 親と子を異なるCPUが担当する場合があるので、階層間通信を用意する必要がある。
 - 孤立領域を複数のCPUが担当する可能性があるため、階層内通信も用意する必要がある。
- 粒子のデータ分割
 - 各粒子を担当するCPUは、その粒子を含む階層格子のうち最も小さい格子が担当するCPUと同一になるようにする。
 - 均等分割は保証されない。(後述)
 - 重力内挿や質量割当の際には階層格子のループに伴う通信を除き、通信は発生しない。

並列化のためのデータ分割 II



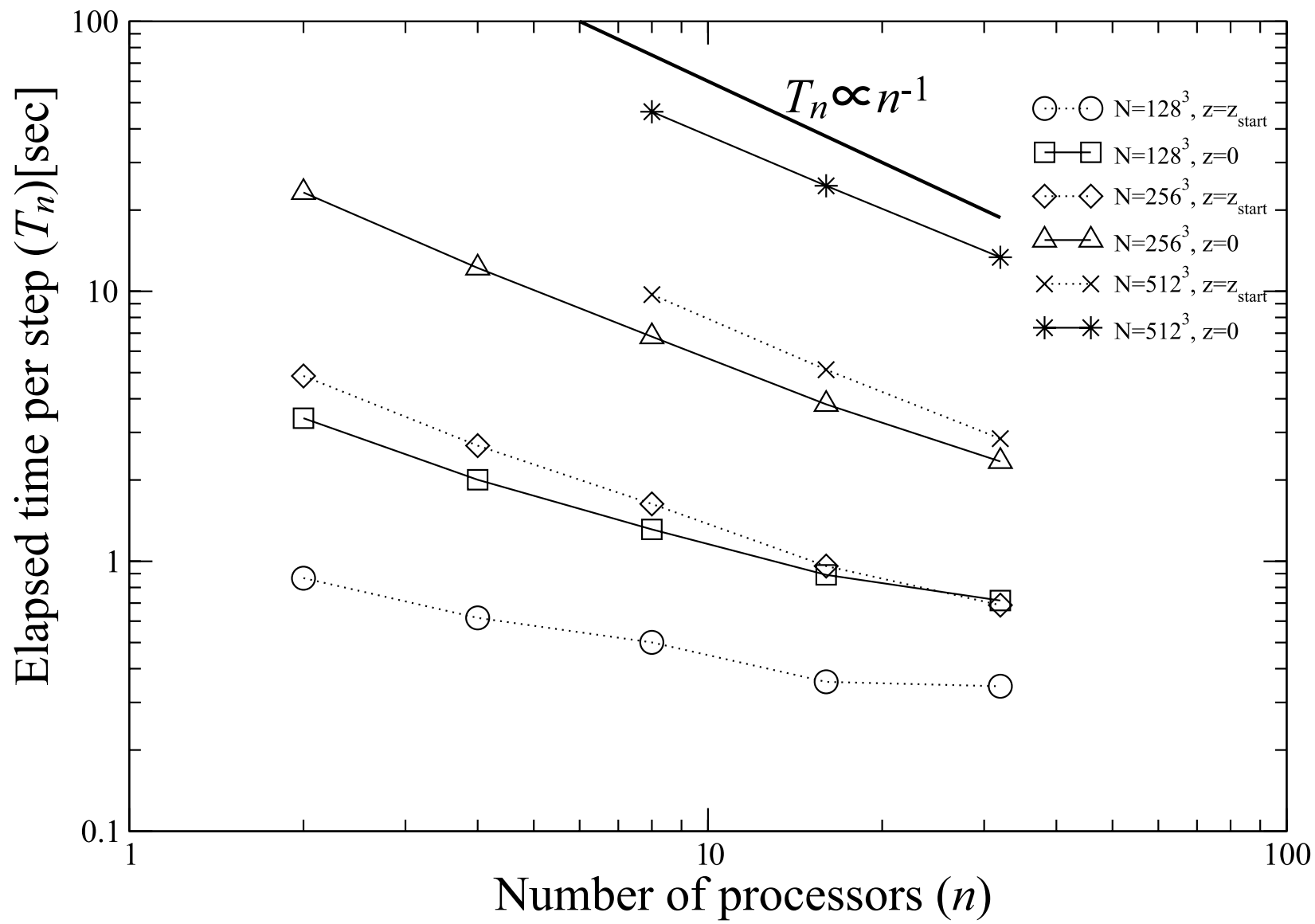
階層格子

粒子

計時解析 I

- Λ CDM simulations
 - $\Omega = 0.3$, $\lambda = 0.7$, $h = 0.7$, $n = 0.9$, $\Omega_b = 0.048$
- 使用機種
 - VPP5000@ADAC/NAOJ
- 粒子数
 - $N = 128^3$, 256^3 , 512^3
- CPU数
 - $N_{\text{proc}} = 2, 4, 8, 16, 32$

計時解析 II



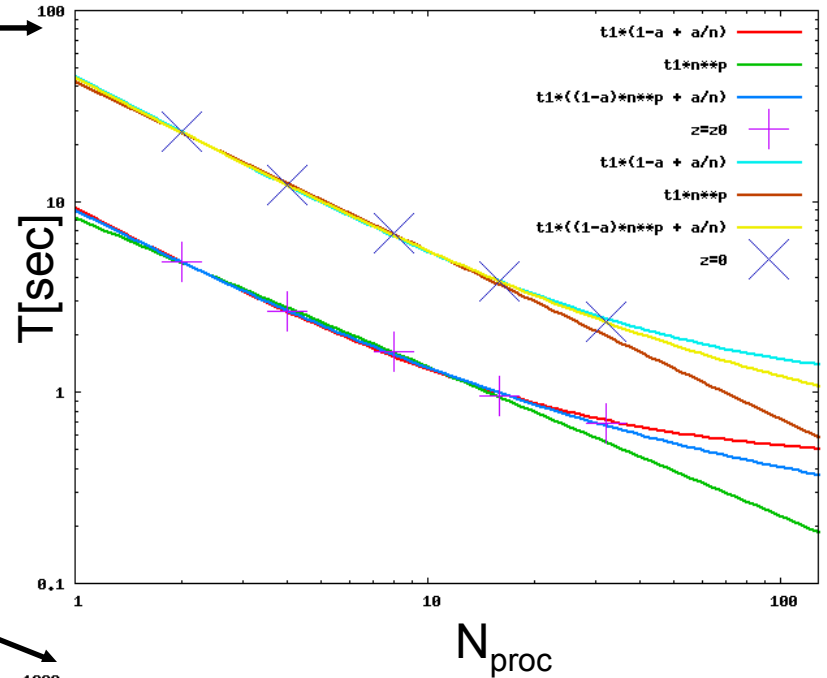
Λ CDMシミュレーションの経過時間を以下の三つの関数型でフィッティングした。

$$T_n = T_1(1 - \alpha + \alpha/n)$$

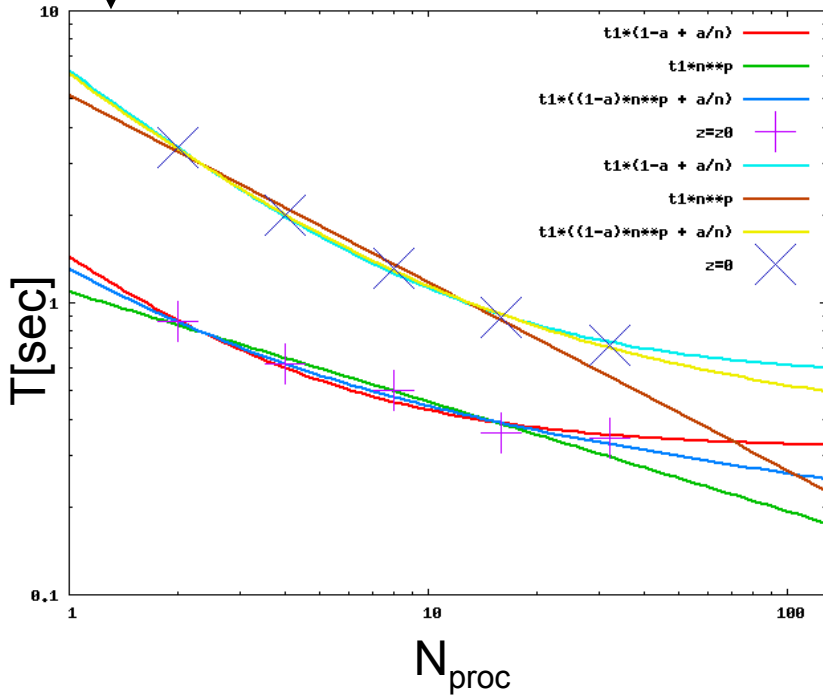
$$T_n = T_1 n^p$$

$$T_n = T_1((1 - \alpha)n^p + \alpha/n)$$

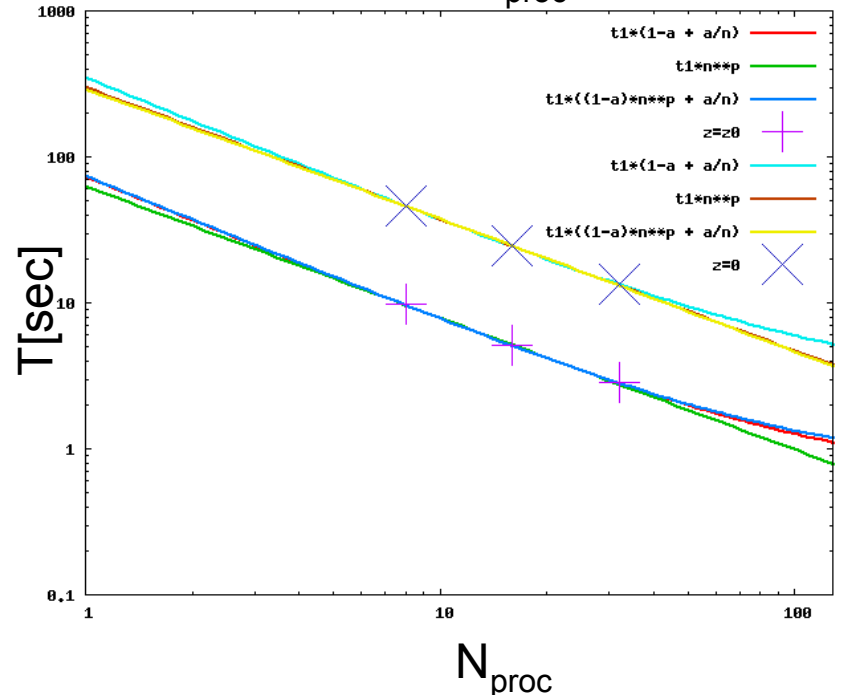
$N=256^3 \rightarrow$



$N=128^3 \downarrow$



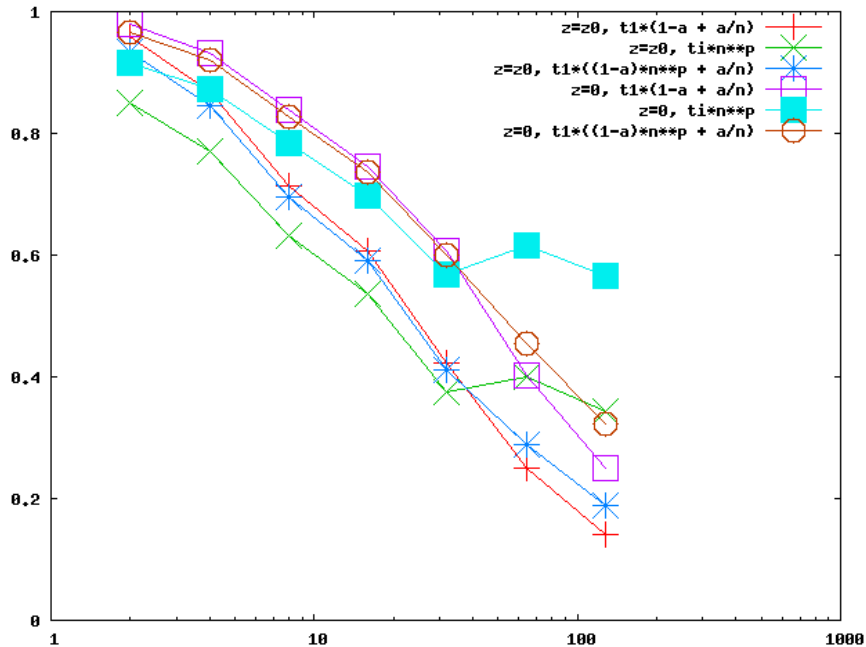
$N=512^3 \rightarrow$



$N=256^3 \rightarrow$

前項のフィッティングから T_1 を見積もり、
 並列化効率 $p_n = (T_1 / (nT_n))$ を見積もった。
 また、 T_{64} , T_{128} も前項のフィッティング
 から見積り、 p_{64} , p_{128} を算出した。その
 結果から、 $n \leq 128$ まで、どのような見積
 りをしても、50%以上の並列化効率を維持
 できるものと予測される。

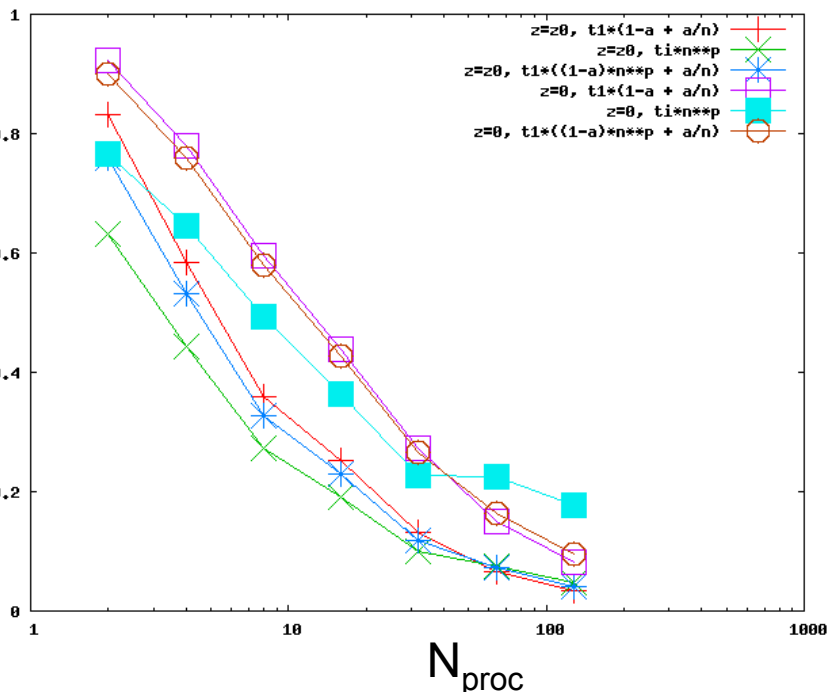
Parallel Efficiency



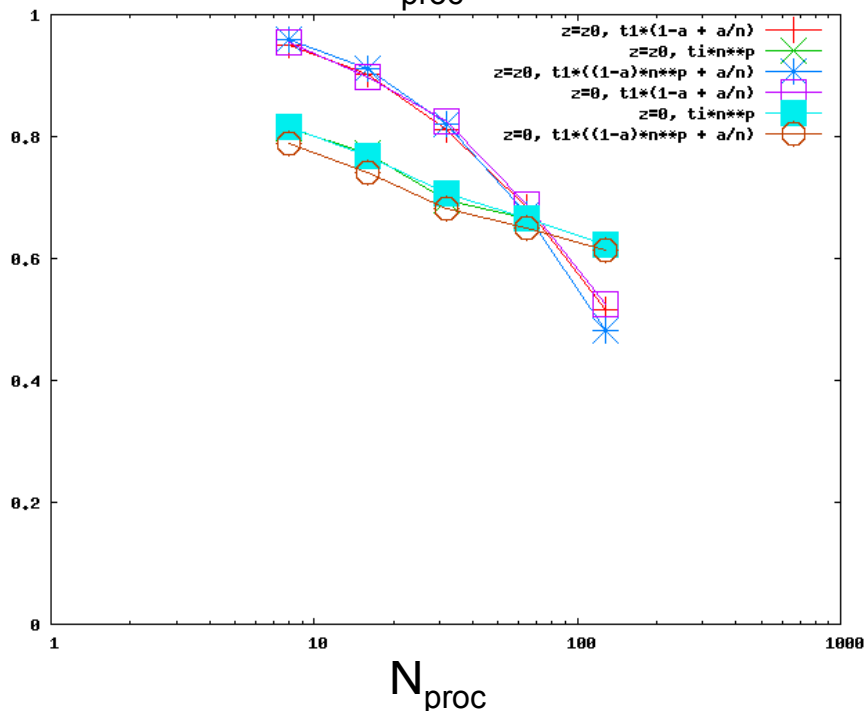
$N=128^3$

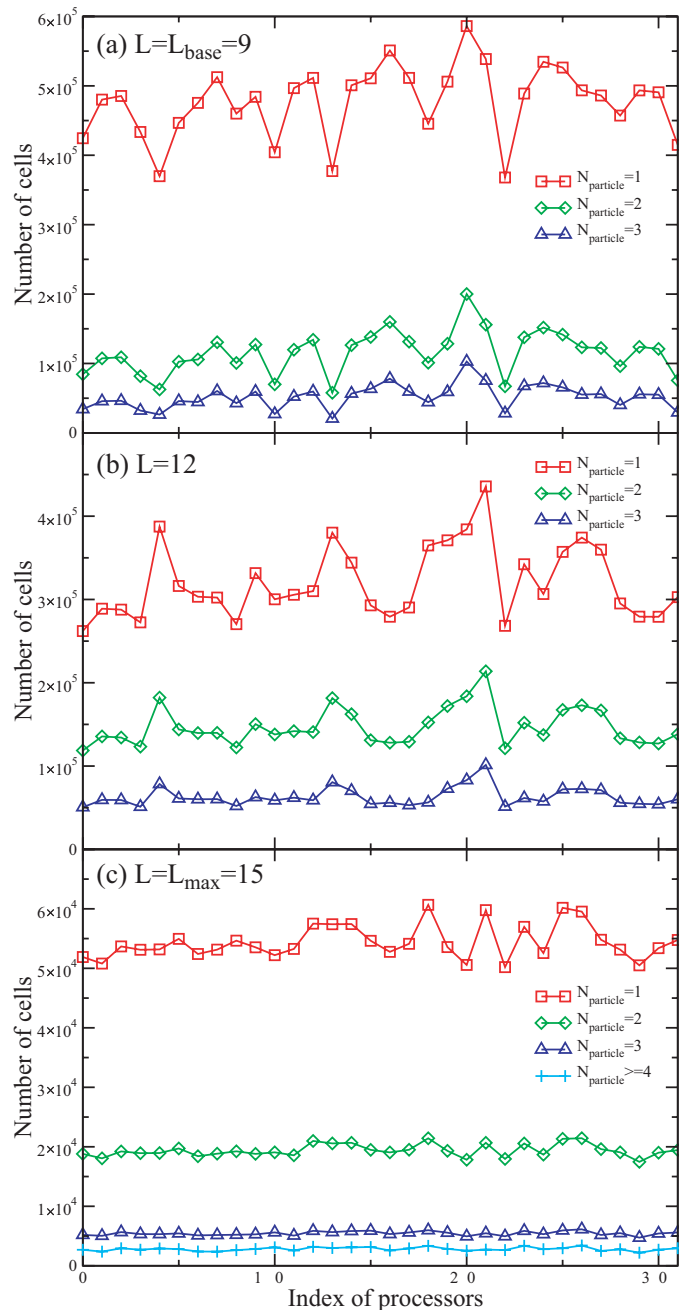
$N=512^3$

Parallel Efficiency



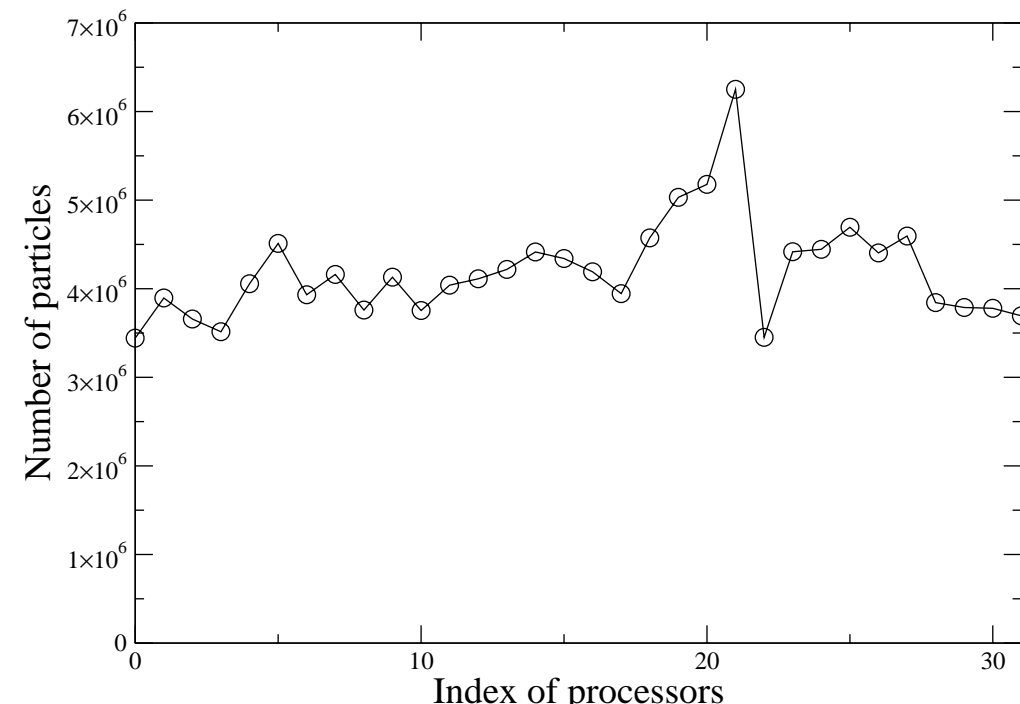
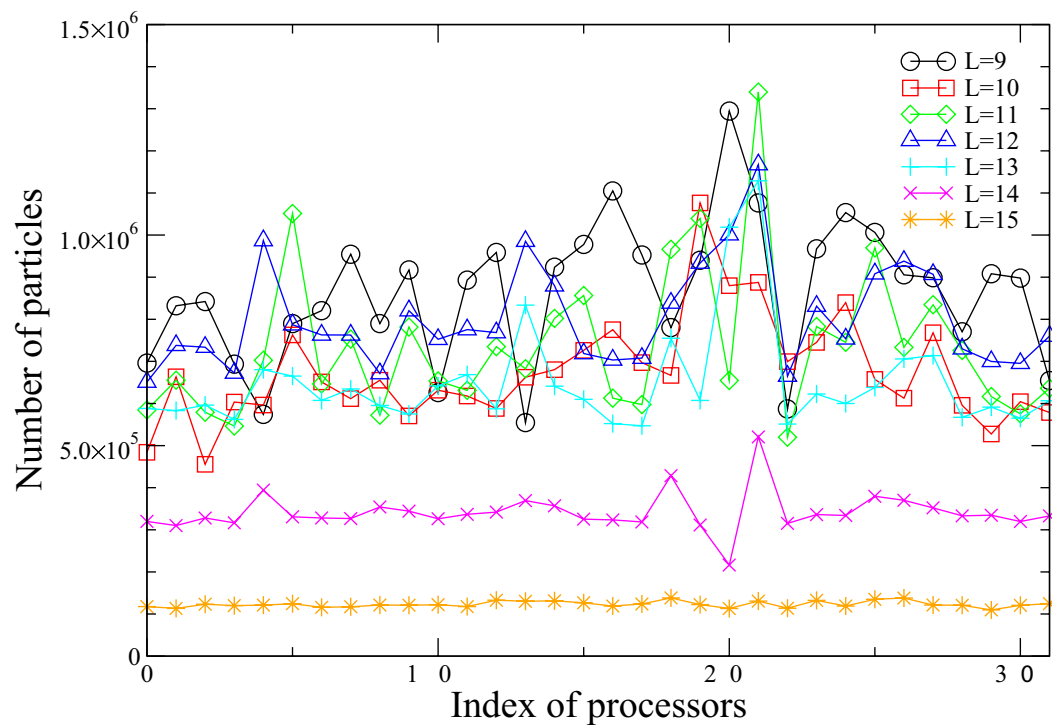
Parallel Efficiency





• 粒子分割

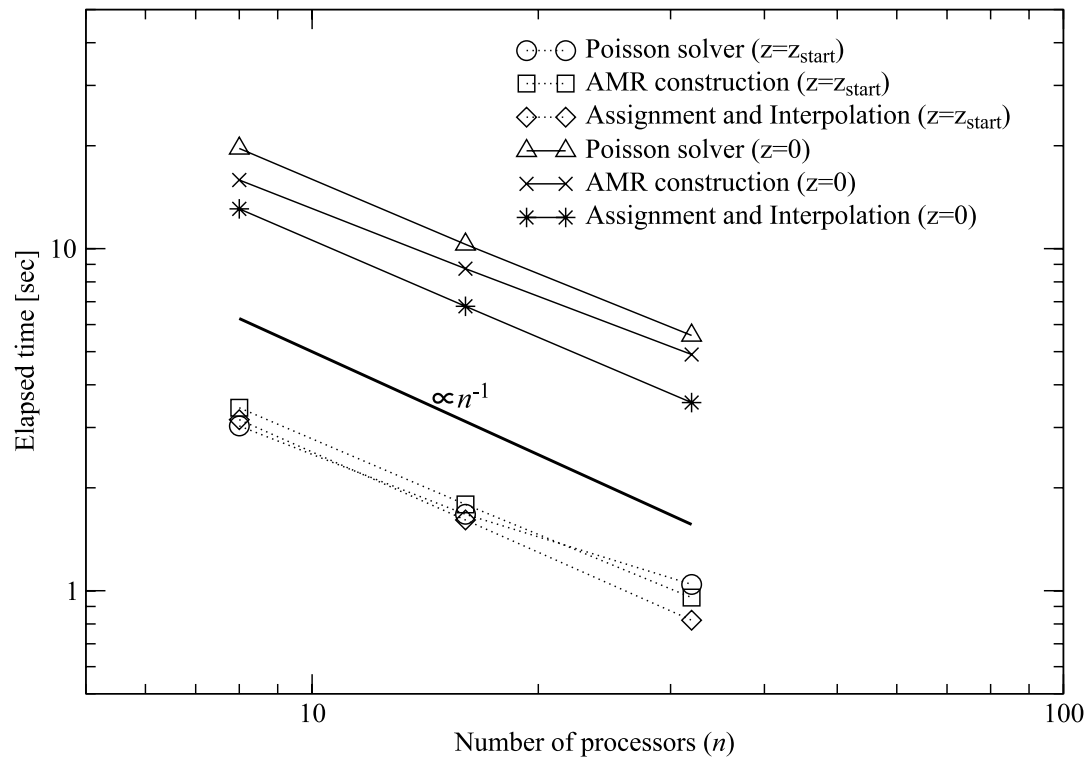
- 上段が一番粗い階層の格子数、下段が一番細かい階層の格子数、中段はその中間の階層の格子数を表している。
- 赤、緑、青、水色は、粒子を一つ含む格子数、二つ含む格子数、三つ含む格子数、四つ以上含む格子数をそれぞれ表している。
- 粒子分割は深い階層の方が浅い階層より均等に分割されている。



• 粒子分割

- 上段は各階層毎の各CPUの担当する粒子数、下段は各CPUの全粒子数を表している。
- 各格子に入る粒子数に制限のない一番細かい格子が均等に分割できないと思われるが、実際には、逆に最も良く均等分割されている。
- 均等分割から、最大で約50%の不均等が生じている。

各ルーチン毎の計時解析



- AMR construction 及び Assignment and Interpolation では粒子データを取り扱うが、これら二つのケースでも理想的な状況に近いスピードアップが実現できている。

誤差解析 I

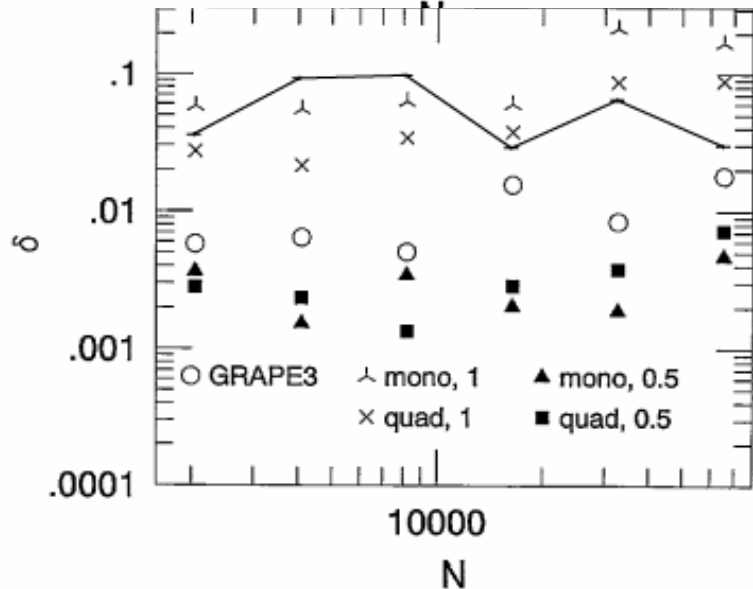
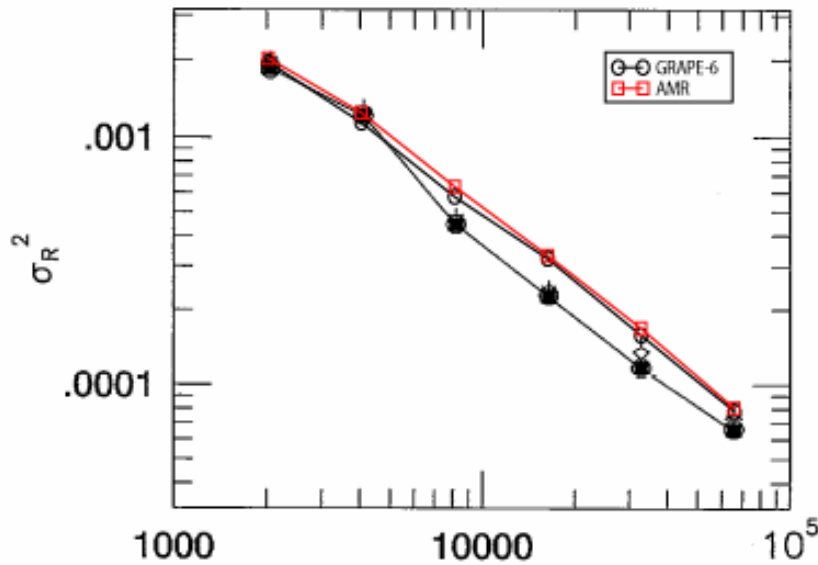
- 共同研究者: 牧野淳一郎
- Hernquist, Hut, & Makino (1993; HHM93)
 - 定常解では、全体の力学的エネルギーだけでなく、個々の粒子の力学的エネルギーが保存する
 - N体シミュレーションで Plummer 解を計算し、個々の粒子の力学的エネルギーの変化を調べた

$$\sigma_R = \left(\frac{1}{N} \sum_{i=1}^{i=N} \left(\frac{E_{i,t} - E_{i,0}}{E_{i,0}} \right)^2 \right)^{1/2}$$

誤差解析 II

- $G=1, M=1, E=-1/4$
 - $\varepsilon = 1/32, \delta t=2$
 - $t=0$ から $t=8$ までの4区間について平均を取る
- スケーリング
 - AMRN体コードでは、最も細かい格子間隔を1とする単位系を採用。
 - 以下のスケーリングをした。
 - $G=1, E=-1/4, R = 32 \frac{3\pi G}{64 |E|}, M = \left(\frac{64 |E| R}{3\pi G} \right)^{1/2}$.
 - $T \propto M^{5/2} |E|^{-3/2}$.

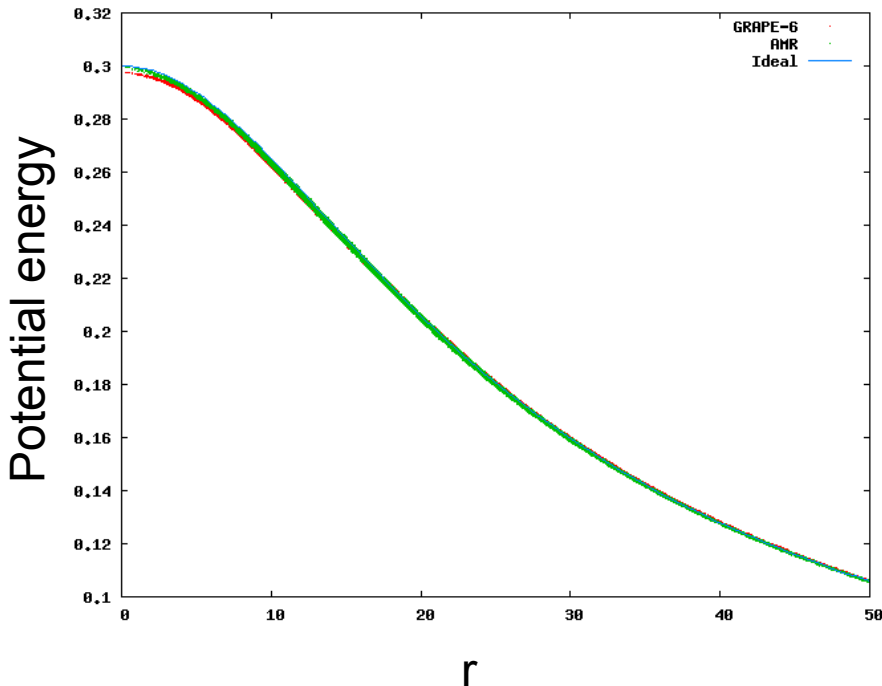
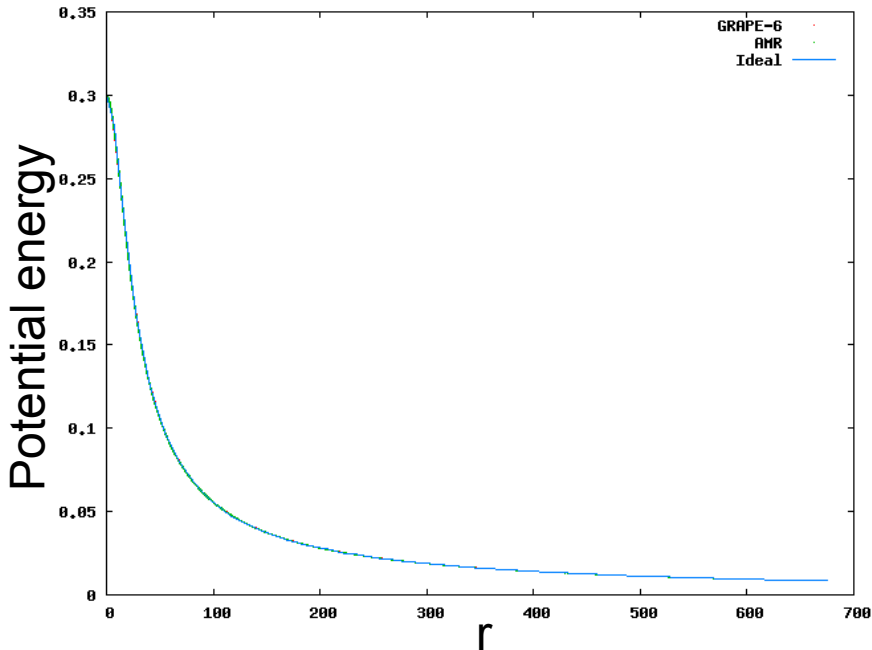
誤差解析 III



- HHM93の図の上に、今回の結果を重ね合わせた(上図)。およそ近い値の誤差になっている。

- $$\delta = \left(\frac{\sigma_R}{\sigma_{R, \text{GRAPE-6}}} \right)^2 - 1$$
をHHM93の図の上にプロットしたものである(下図)。AMRの結果はツリー法の $\theta = 1.0$ の結果に近い結果となった。

ポテンシャル



- 大局的にはGRAPE-6の結果と、AMRの結果に差は無い。N=65536.
- 中心付近を拡大すると、GRAPE-6よりAMRの結果の方が解析解に近い。
- $\sigma^2 = \frac{1}{N} \sum_{i=0}^N \left(\frac{\phi_i - \phi(r)}{\phi(r)} \right)^2$
 - GRAPE-6: 1.91367e-05
 - AMR: 1.51096e-05

まとめ

- 分散並列計算機向けにAMR_N体コードの並列化を行った。
- Λ CDMシミュレーションによる計時解析を行った。
 - 粒子数が 512^3 の場合、CPU数が32でも、約80%の並列化効率を出すことが出来た。
- Plummerモデルを用いた誤差解析を行った。
 - AMRコードの誤差はBHツリーコードの $\theta = 1.0$ の場合に近い結果を出した。