

四重極項計算のSIMD化による ツリー法的高速化

千葉大学大学院 融合理工学府

児玉哲史

千葉大学 統合情報センター

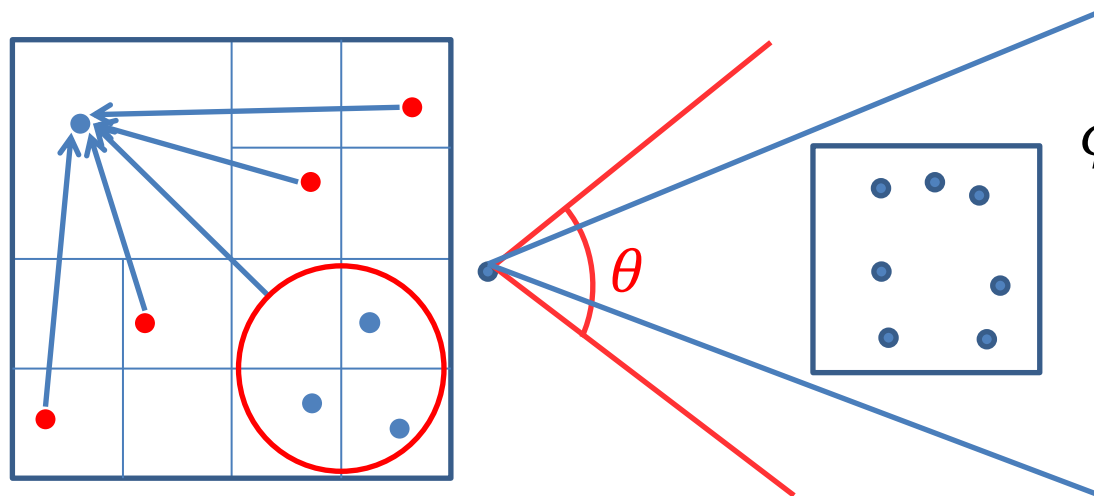
石山智明

重力多体シミュレーション

- 各粒子の運動方程式を数値積分
 - 粒子間で相互に重力が働く
- 宇宙の大規模構造や天体の理解に有用
 - 銀河・銀河団、星団、ダークマター構造形成など
- 直接計算法: 計算量 $O(N^2)$
- ツリー法: $O(N \log N)$

ツリー法(Barnes & Hut 1986)

- 近傍粒子からの作用は直接計算
- 遠方粒子群からの作用は多重極展開で近似
 - 近傍・遠方の判定基準は見込角 θ で決定
 - 粒子群の見込角 $< \theta$: 遠方と判断
 - 単極子では $\theta = 0.3 \sim 0.75$ が用いられることが多い
 - 単極子の計算はPhantom-GRAPEで高速化可能



$$\phi(r) = -\frac{GM}{r} - \frac{1}{2} \frac{G}{r^5} \mathbf{r} \cdot \mathbf{Q} \cdot \mathbf{r}$$

$\phi(r)$:ポテンシャル

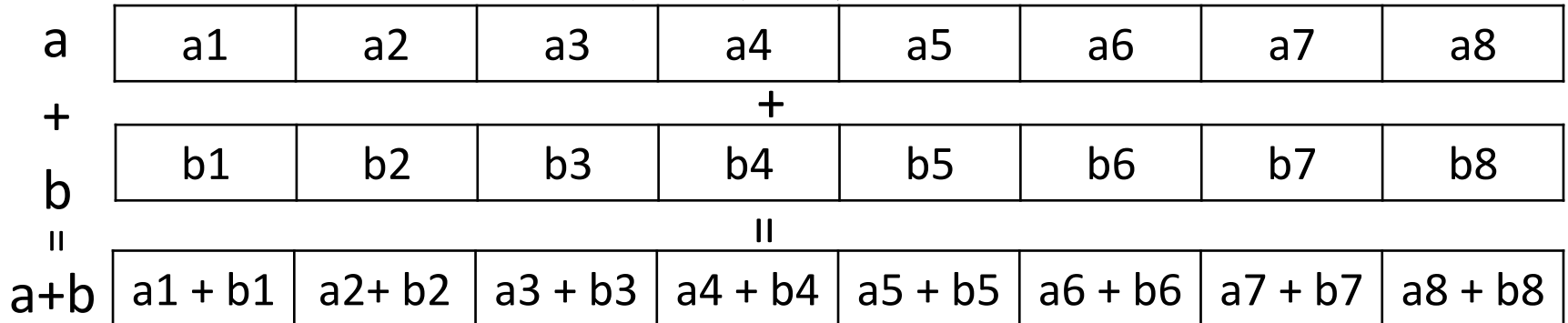
M :総質量

\mathbf{Q} :四重極テンソル

Phantom-GRAPEとSIMD

- Phantom-GRAPE(Tanikawa and Yoshikawa. 2012, 2013): **SIMD**を利用して、単極子項の計算を大幅に高速化
- **SIMD**:1命令で複数データを並列演算
- 現在のIntelのCPU:AVX, AVX2

– レジスタ長256bit=単精度浮動小数点データ8個を並列に演算可能

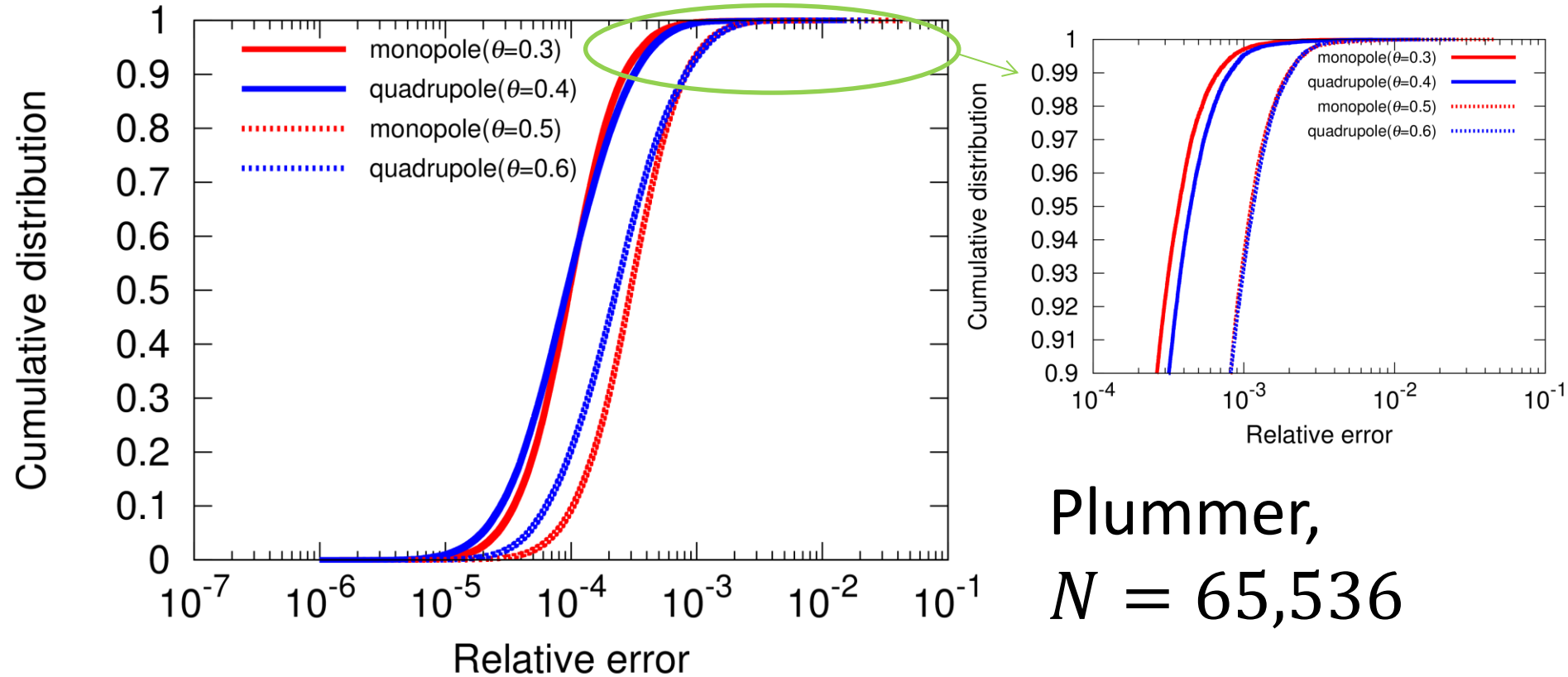


SIMD演算の概念図

本研究の目的

- Phantom-GRAPEでは四重極項の計算は不可
 - 擬似粒子法と組み合わせれば可能
- 我々は四重極までの計算をSIMDを利用して高速化するコードを実装
 - Phantom-GRAPEを拡張
- 単極子のみで行うシミュレーションを、四重極まで用いて高速化可能か検証
 - 同じ計算精度ならば単極子の場合よりも θ を大きくできる
- 今後レジスタ長増加に応じて我々の実装が有利になる可能性あり

直接計算法との相対誤差の累積分布



Plummer,
 $N = 65,536$

単極子 $\theta = 0.3$ の精度は四重極 $\theta = 0.4$ で得られる
単極子 $\theta = 0.5$ の精度は四重極 $\theta = 0.6$ で得られる

単極子のみの場合に相当する精度を得られる θ の値

一様球の場合		Plummerの場合		Diskの場合	
単極子	四重極	単極子	四重極	単極子	四重極
0.3	0.65	0.3	0.4	0.3	0.45
0.5	0.75	0.5	0.6	0.5	0.65

単極子項のみの場合との計算時間の比較(粒子数 $N = 4,194,304$)

一様球			Plummer			Disk		
プログラム	θ	合計[s]	プログラム	θ	合計[s]	プログラム	θ	合計[s]
単極子	0.3	23.99	単極子	0.3	41.23	単極子	0.3	23.24
四重極	0.65	10.88	四重極	0.4	44.65	四重極	0.45	24.86
単極子	0.5	10.26	単極子	0.5	15.13	単極子	0.5	11.75
四重極	0.75	8.35	四重極	0.6	22.26	四重極	0.65	12.67

- 一様球では約1.23 - 2.20倍高速化
- Plummerで約1.08 - 1.47倍、Diskで約1.07倍低速化
- 密度コントラストの小さい系で有効

実装したコードの一部

```
void c_GravityKernel(pIpdata ipdata, pFodata fodata, cJcdata
int j;
float five[8] = {5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0};
float half[8] = {0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5};

PREFETCH(jcdata[0]);■

VZEROALL;

// load i-particle
VLOADPS(*ipdata->x, XMM00);
VLOADPS(*ipdata->y, XMM01);
VLOADPS(*ipdata->z, XMM02);
VLOADPS(jcdata->xm[0][0], YMM14);
VLOADPS(jcdata->q[0][0], YMM08);
VLOADPS(jcdata->q[1][0], YMM15);
jcdata++;
VPERM2F128(YMM00, YMM00, YMM00, 0x00);
VPERM2F128(YMM01, YMM01, YMM01, 0x00);
VPERM2F128(YMM02, YMM02, YMM02, 0x00);

// load jcell's coordinate
VSHUFPS(YMM14, YMM14, YMM03, 0x00); //00
VSHUFPS(YMM14, YMM14, YMM04, 0x55); //55
VSHUFPS(YMM14, YMM14, YMM05, 0xaa); // aa

for(j = 0; j < nj; j += 2){
    // r_ij,x -> YMM03
    VSUBPS(YMM00, YMM03, YMM03);
    // r_ij,y -> YMM04
    VSUBPS(YMM01, YMM04, YMM04);
    VLOADPS(*ipdata->eps2, XMM01);
    VPERM2F128(YMM01, YMM01, YMM01, 0x00);
```

- コンパイラによる自動SIMD化では不十分
- SIMD命令を利用するコードを手動で記述

擬似粒子法との計算時間の比較 (粒子数 $N = 4,194,304$)

一様球			Plummer			Disk		
プログラム	θ	合計[s]	プログラム	θ	合計[s]	プログラム	θ	合計[s]
擬似粒子法	0.65	11.91	擬似粒子法	0.4	48.61	擬似粒子法	0.45	27.09
四重極	0.65	10.88	四重極	0.4	44.65	四重極	0.45	24.86
擬似粒子法	0.75	9.22	擬似粒子法	0.6	24.30	擬似粒子法	0.65	13.92
四重極	0.75	8.35	四重極	0.6	22.26	四重極	0.65	12.67

- 擬似粒子法＝四重極展開を仮想粒子で表現
 - Phantom-GRAPEをそのまま利用可能
- 我々の実装は擬似粒子法に比べ約1.1倍高速
 - 四重極テンソルの対角化が不要なため

レジスタ長の増加で有利になる理由

- 主な処理: ツリー構築、ツリー走査、重力計算
- 8分木を取り扱うため、ツリー構築とツリー走査で並列数8を超えるSIMD長の活用は困難
- 重力計算はSIMD長の増加が高速化に直結
 - 走査時の相互作用のリスト化が必要
 - 相互作用の数は非常に多い

単極子 ツリー構築 ツリー走査 重力計算

四重極
+ θ 増加 ツリー構築 ツリー走査 重力計算



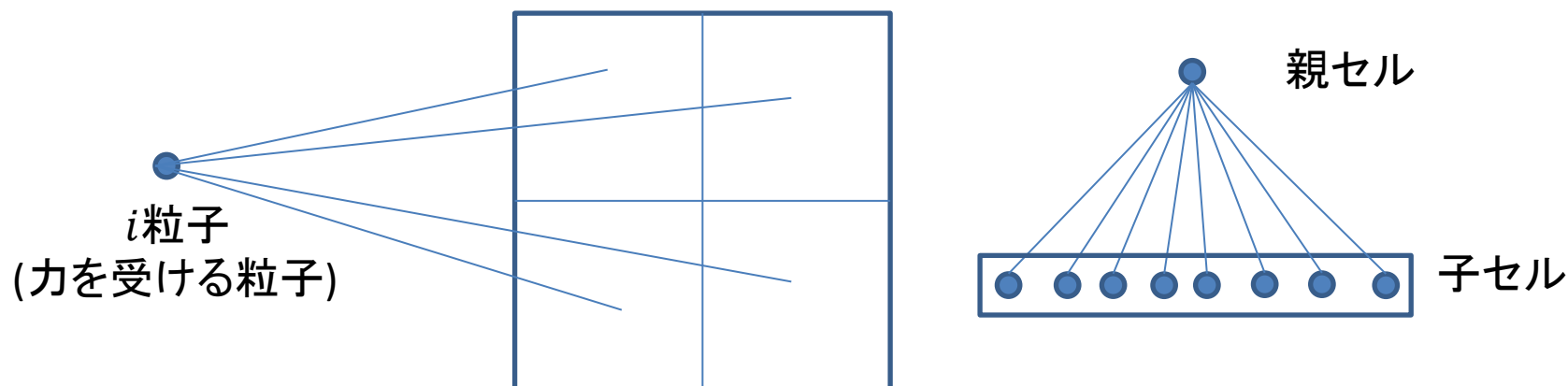
単極子 ツリー構築 ツリー走査 重力計算

四重極
+ θ 増加 ツリー構築 ツリー走査 重力計算

SIMD長の増加によって全体の計算時間が逆転する可能性あり

ツリー走査のSIMD化

- 8個の子セルの見込角の計算をSIMD化
- さらに深く走査するかは計算するまでわからないため、8並列を超えたSIMD化は困難



次世代SIMD命令セット:AVX-512で 実装時の計算時間見積もり

- 単極子項および四重極項の計算部分をAVX-512で実装した場合の見積もり(他の処理は変更無)
 - 他の処理(ツリー構築・ツリー走査)は8を超える並列数のSIMD化が困難
- 単極子項の計算は2倍高速化と予想
 - AVXに対してレジスタ長が2倍(512bit)であるため
- 四重極項の計算は約2.46倍高速化と予想
 - AVXに対してレジスタ長が2倍であるため
 - レジスタ本数がAVXより多く(32本)、 i 粒子データ等のロードを毎回実行する必要がなくなるため(約1.23倍)

レジスタ本数増加によるロード命令削減

四重極項までの計算を行うループの一部

```
for(j = 0; j < nj; j += 2){  
    // load i-particle  
    VLOADPS(*ipdata->x, XMM00);  
    VLOADPS(*ipdata->y, XMM01);  
    VLOADPS(*ipdata->z, XMM02);  
    VPERM2F128(YMM00, YMM00, YMM00, 0x00);  
    VPERM2F128(YMM01, YMM01, YMM01, 0x00);  
    VPERM2F128(YMM02, YMM02, YMM02, 0x00);  
  
    // load jcell's coordinate  
    VLOADPS(*jdata->x, YMM03);  
    VLOADPS(*jdata->y, YMM04);  
    VLOADPS(*jdata->z, YMM05);  
  
    // r_ij,x -> YMM03  
    VSUBPS(YMM00, YMM03, YMM03);  
  
    // r_ij,y -> YMM04  
    VSUBPS(YMM01, YMM04, YMM04);  
  
    // r_ij,z -> YMM05  
    VSUBPS(YMM02, YMM05, YMM05);  
  
    // r_ij^2 -> YMM01  
    VLOADPS(*ipdata->eps2, XMM01);  
    VPERM2F128(YMM01, YMM01, YMM01, 0x00);
```

- AVX-512ではこの部分が
必要なくなる
 - レジスタ本数が増加し、
常にデータを置けるため
- 左で示された範囲以外
にも、ループ内で2個の
ロード命令が削減可能
 - 定数0.5が並んだ配列を
レジスタにロードする命
令
 - 定数5が並んだ配列をレ
ジスタにロードする命令

AVX-512利用時の計算時間見積もり

一様球

Plummer

Disk

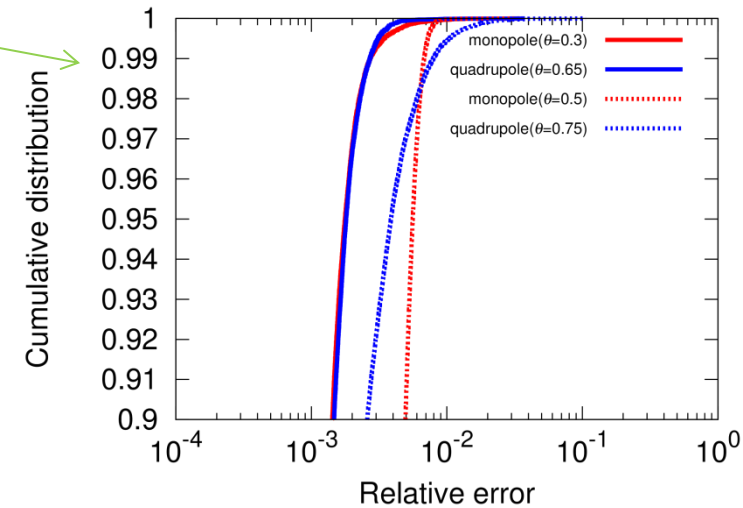
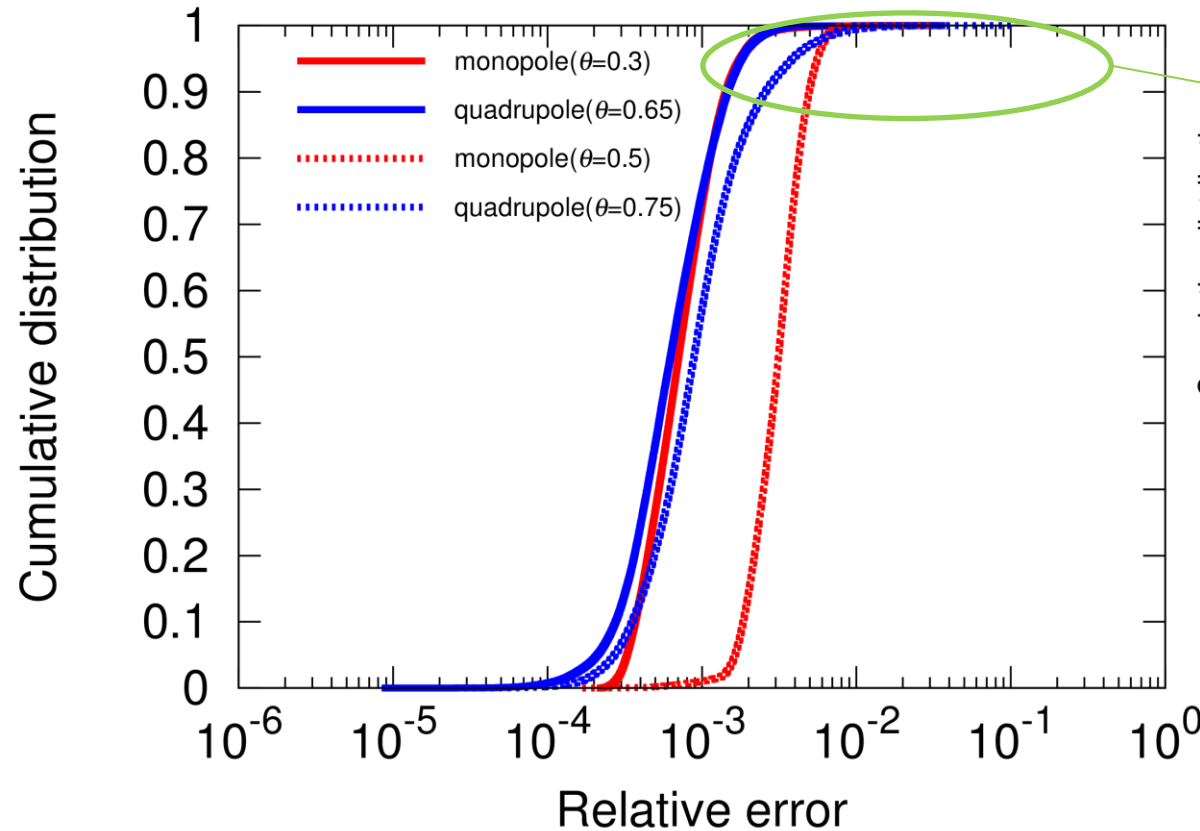
プログラム	θ	合計[s]	プログラム	θ	合計[s]	プログラム	θ	合計[s]
単極子	0.3	13.99	単極子	0.3	25.49	単極子	0.3	15.69
四重極	0.65	6.13	四重極	0.4	23.66	四重極	0.45	13.73
単極子	0.5	6.29	単極子	0.5	9.90	単極子	0.5	7.20
四重極	0.75	4.86	四重極	0.6	12.52	四重極	0.65	7.51

- AVX-512で実装時、密度コントラストの大きい系の高精度シミュレーションも高速化可能
 - Plummerで約1.08倍, Diskで約1.14倍

まとめ

- SIMD命令を利用して高速に四重極項を計算するコードを実装
- 我々の実装は擬似粒子法よりも約1.1倍高速
- 密度コントラストの小さい系では、本研究による実装を用いて四重極まで計算することで最大約2.2倍の高速化が可能
- AVX-512では密度コントラストが高い系でも最大約1.14倍の高速化が可能と見積もられる

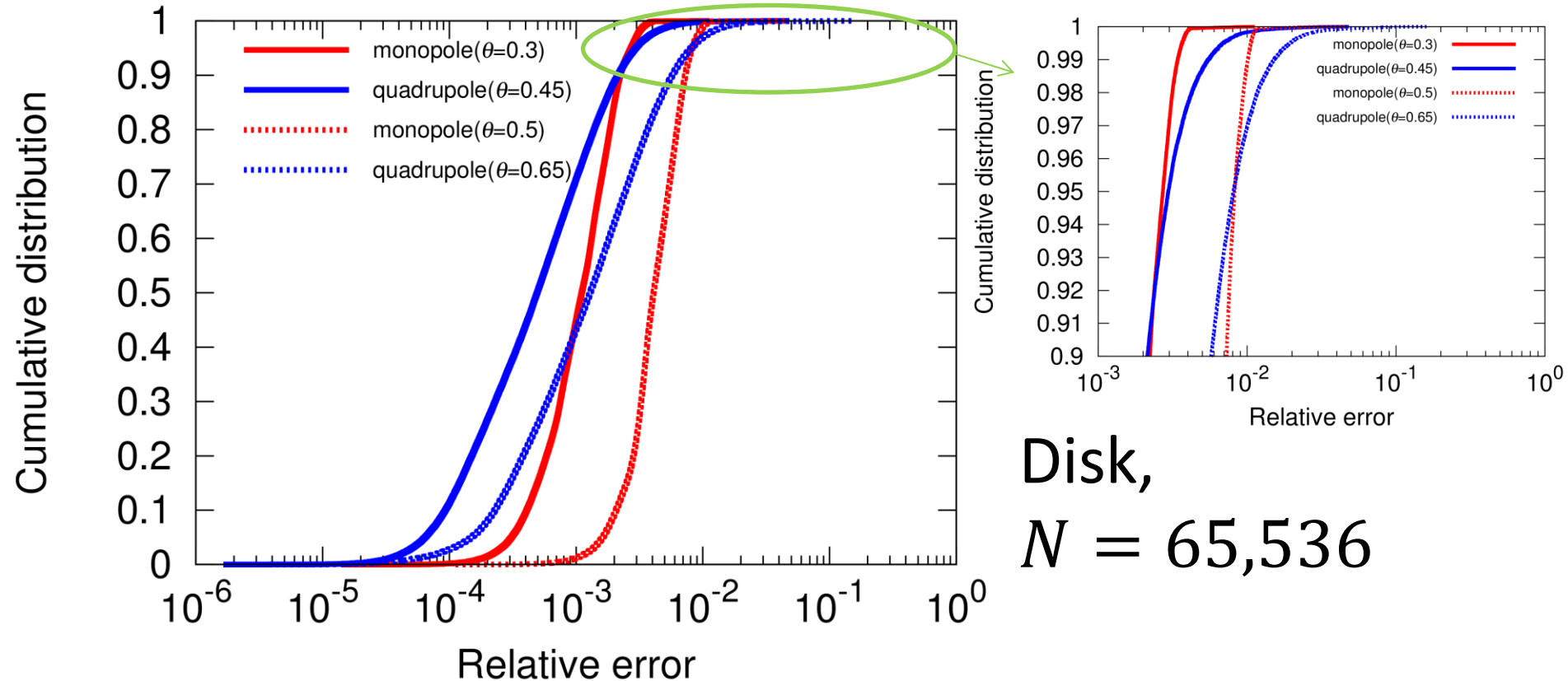
直接計算法との相対誤差の累積分布



一様球,
 $N = 65,536$

単極子 $\theta = 0.3$ の精度は四重極 $\theta = 0.65$ で得られる
単極子 $\theta = 0.5$ の精度は四重極 $\theta = 0.75$ で得られる

直接計算法との相対誤差の累積分布



単極子 $\theta = 0.3$ の精度は四重極 $\theta = 0.45$ で得られる
単極子 $\theta = 0.5$ の精度は四重極 $\theta = 0.65$ で得られる