高速多重極展開法とツリー法
— $O(N)$ は $O(N \log N)$ より速いか？

牧野淳一郎
東京大学大学院総合文化研究科

要約

多数の質点間の重力相互作用や荷電粒子の間の静電相互作用を高速に計算する手法として 1980 年代中頃に提案された FMM (Fast Multipole Method, 高速多重極展開法）と、それと非常に関連が深く、やはり 1980 年代に提案されて広く使われるようになったツリー法について、その原理と研究の動向を概説する。

講演概要

1. 解くべき問題
2. ツリー法
3. 高速多重極展開法
4. 計算の効率化 (並列化等) について
5. どちらが速いのか？
6. アンダーソンの方法と Pseudoparticle method
7. 他の問題への適用等について

解くべき問題

例：（重力）N 体問題

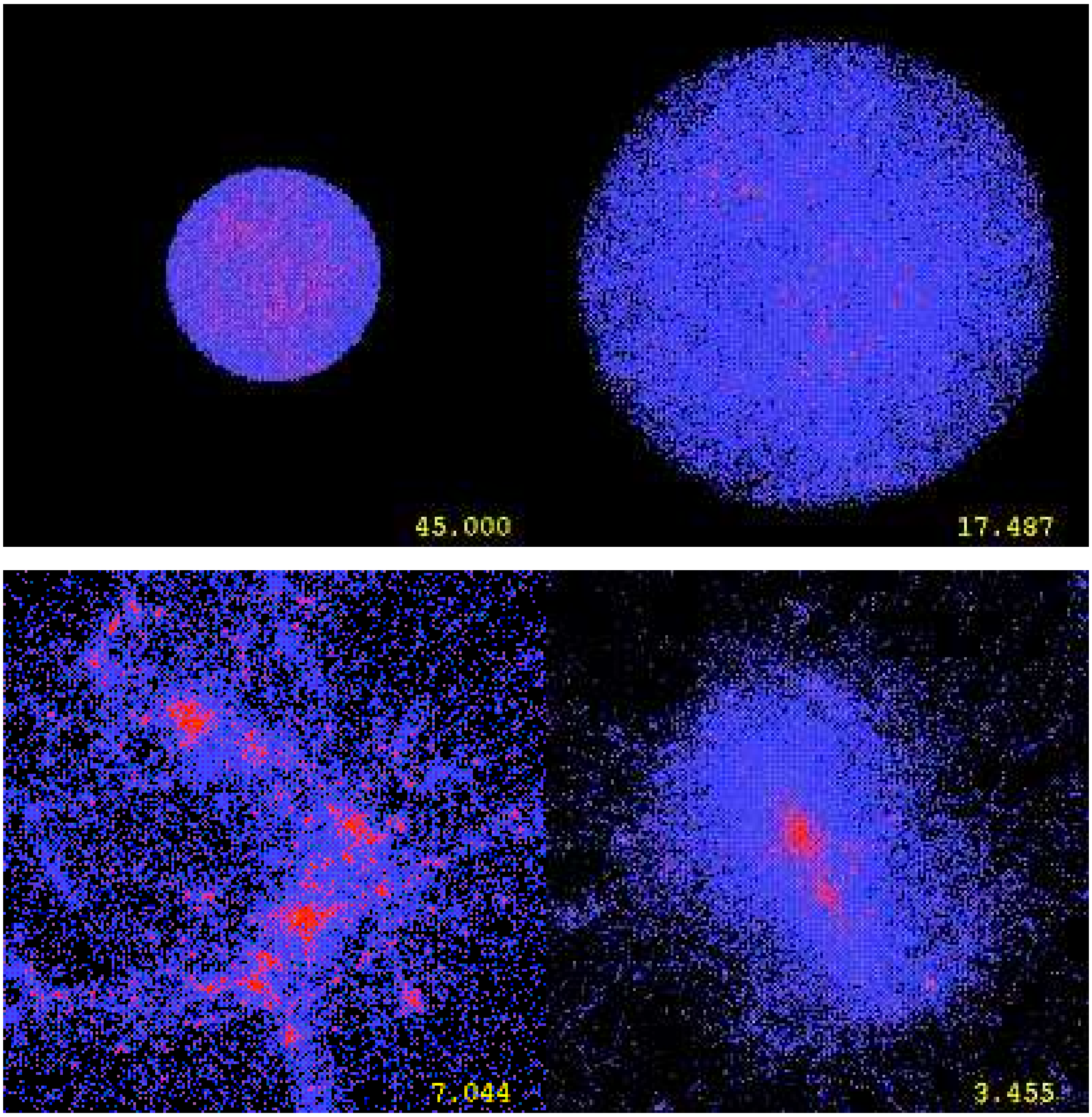
$$\frac{d^2\boldsymbol{x}_i}{dt^2} = G\sum_{j=1,j\neq i}^{N} m_j \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{|\boldsymbol{x}_j - \boldsymbol{x}_i|^3},$$

時間積分はなにか適当な方法（leap frog とか）でやる
local timestep とかいったこともとりあえずは考えない

いかにして粒子間相互作用を高速に計算するか？

ナイーブな方法：計算量が粒子数の 2 乗に比例

計算例



銀河形成過程のシミュレーションの例。画面の数字は赤方偏移 z であり、過去に遡るほど大きくなる ($z = 0$ が現在、 $z = \infty$ が宇宙の始まり)。色は密度をあらわす（赤：高密度、青：低密度）。
（8 万粒子、 GRAPE-4 ＋ 独立時間刻みによる直接計算)

古典的な高速化法

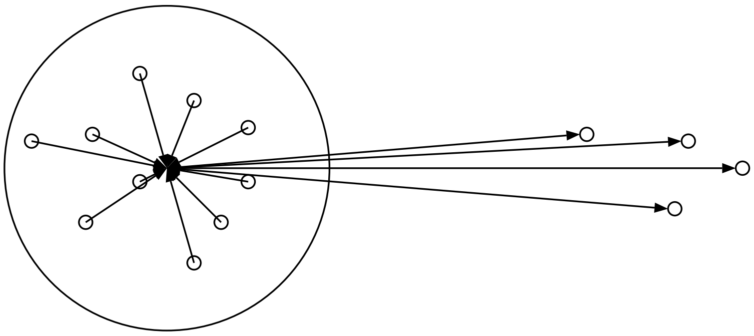
- **Particle-mesh** 法： 空間格子（通常等間隔）を切って、 **FFT** を使ってポアソン方程式を解く
  - 利点：高速である
  - 欠点：空間分解能、精度に制約がある（多重格子のような方法もあるが、、、）
- 球関数展開法： （扱う対象が、球形に近い時）密度分布を球関数展開する。
  - 利点：高速である。球対称に近ければ高精度
  - 欠点：球対称から遠くなると急速に計算量が増える

ツリー法、 FMM の基本的発想

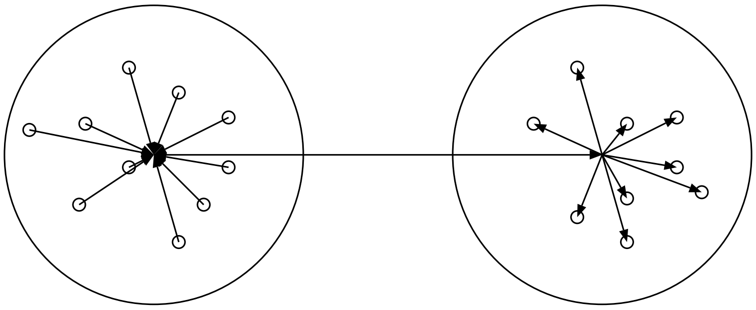
遠くの粒子からの力は弱い

↓

まとめて計算できないか？



Tree

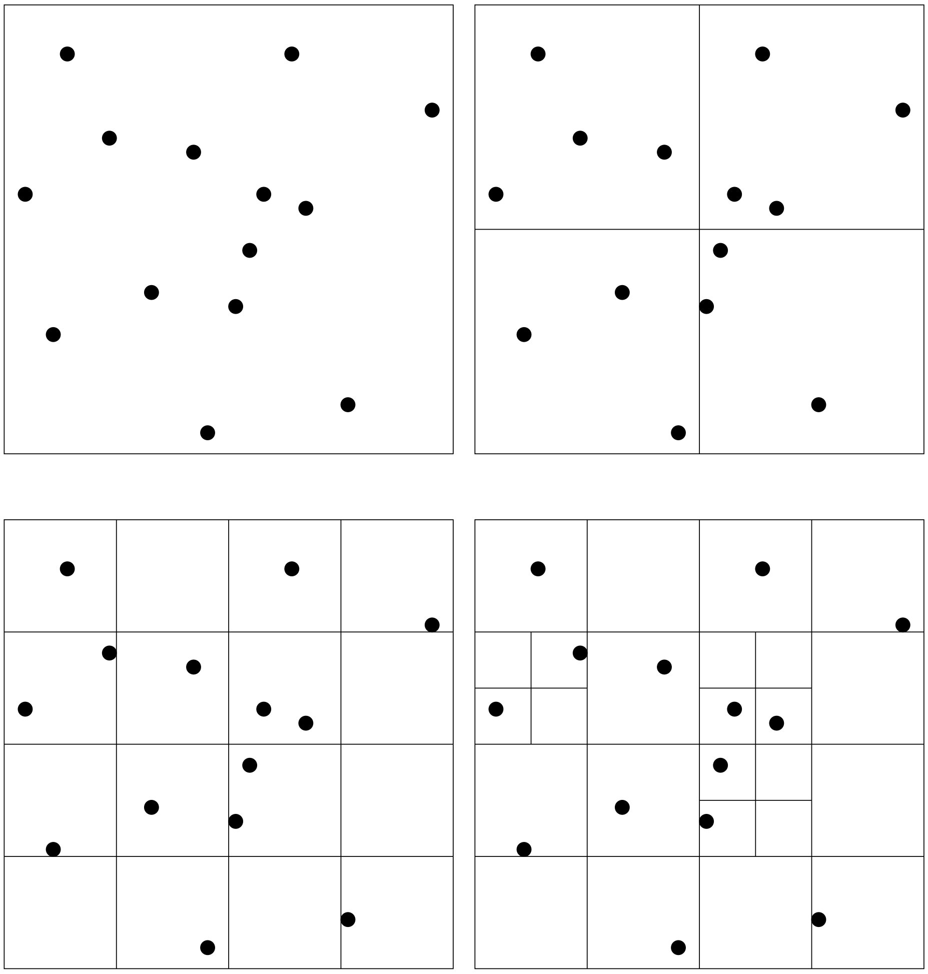


FMM

- ツリー：力を及ぼすほうだけをまとめて評価
- FMM：力を受けるほうもまとめて評価

どうやってまとめるか？ —— ツリー法の場合

- 階層的なツリー構造を使う。



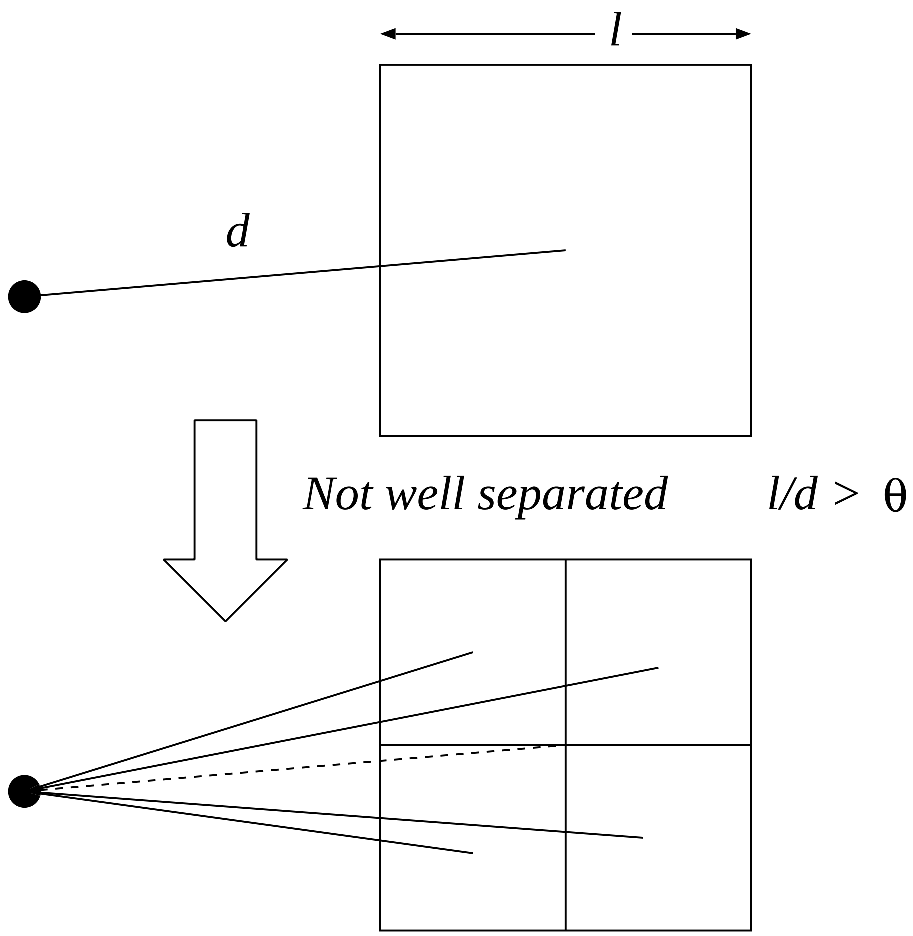
- まず、全体が入るセルを作る
- それを再帰的に 8 (2 次元なら 4) 分割する
- 中の粒子が 1 個 (あるいは 0 個) になったら止める

(多重極展開の次数によってどこで止めるべきかは変わってくる)

ツリー法での力の計算

各セルの多重極展開をあらかじめ再帰的に構成しておく。つまり、

- 最下層のセル: そのなかの粒子分布が作るポテンシャルを多重極展開
- それ以外：子セルの多重極展開の展開中心をシフトして加算



- 十分に離れている：重心（あるいは多重極展開）からの力
- そうでない：子ノードからの力の合計

系全体からの力 ＝ ルートノードからの力
あとは再帰的に計算される。

ツリー法の計算量と計算精度

簡単なオーダー見積り:

$$誤差 \propto \theta^{(p+1)}$$
$$計算量 \propto \theta^{-3} p^2 N \log N$$

p: 多重極展開の次数

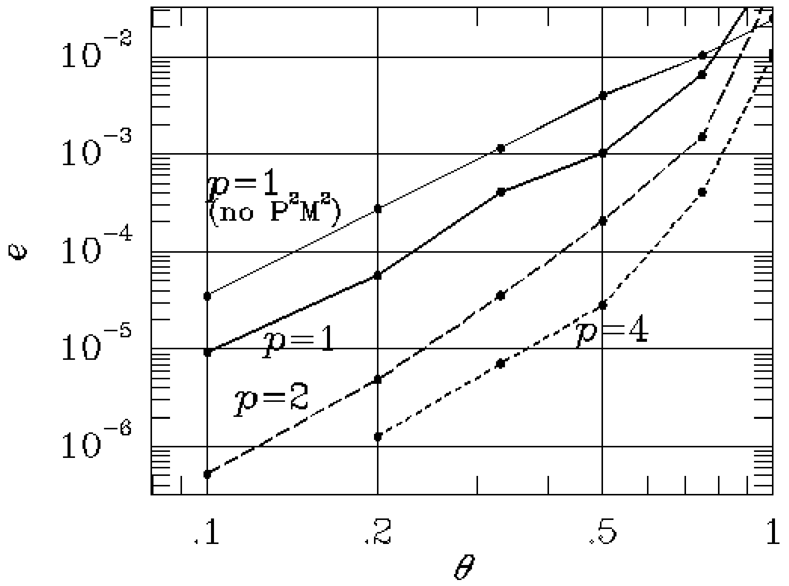
$\log N$: ツリーの階層数
θ^3 ある階層で、相互作用を計算するセルの数

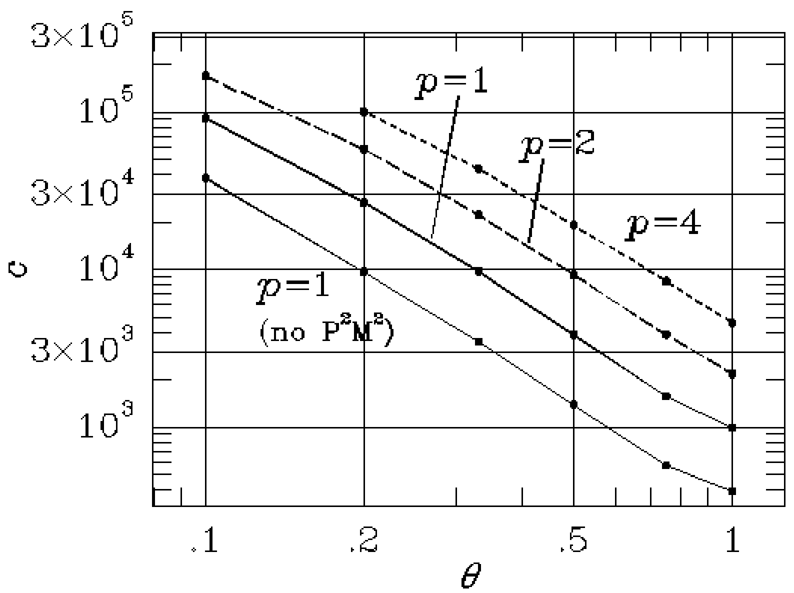
実際の振舞い：複雑。

- 精度は上の見積りよりよい（θ のべきが大きい）
- 計算量はそれほど増えない

ことが多い。（粒子の分布による）

計算精度の例

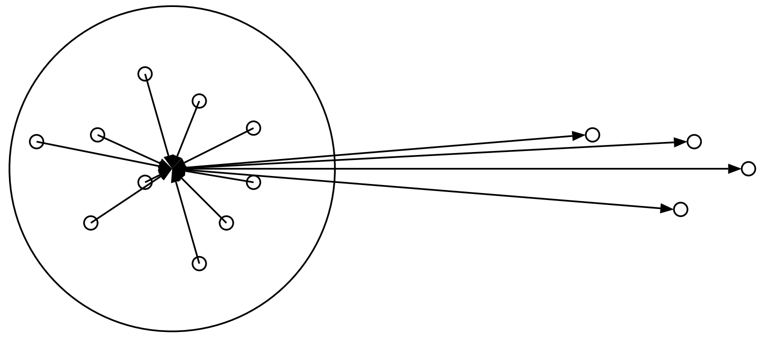




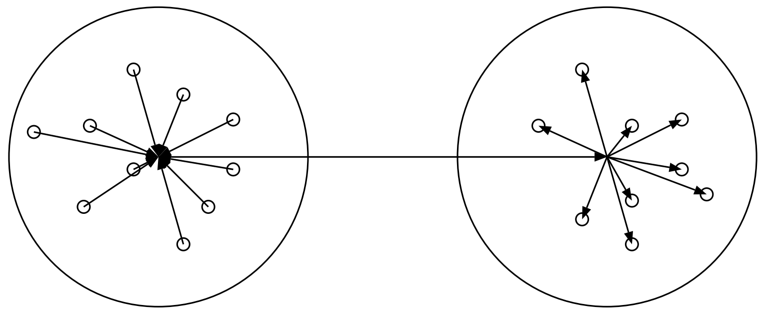
これは後述の疑似粒子多重極法によるもの。通常のツリー法では 4 重極まででですますことが多い。

p=4 の振舞いはちょっと変。

高速多重極展開法



Tree



FMM

- ツリー：力を及ぼすほうだけをまとめて評価
- **FMM**：力を受けるほうもまとめて評価

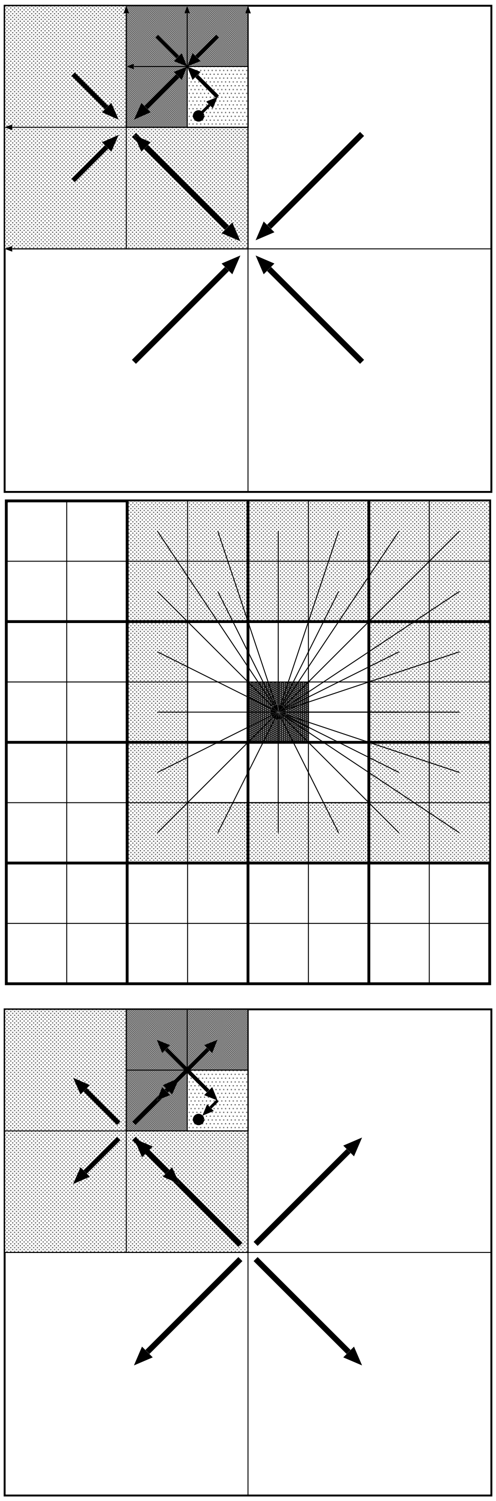
ツリーに比べたメリット：

$\boxed{\text{計算量が } O(N) \text{ になる！}}$

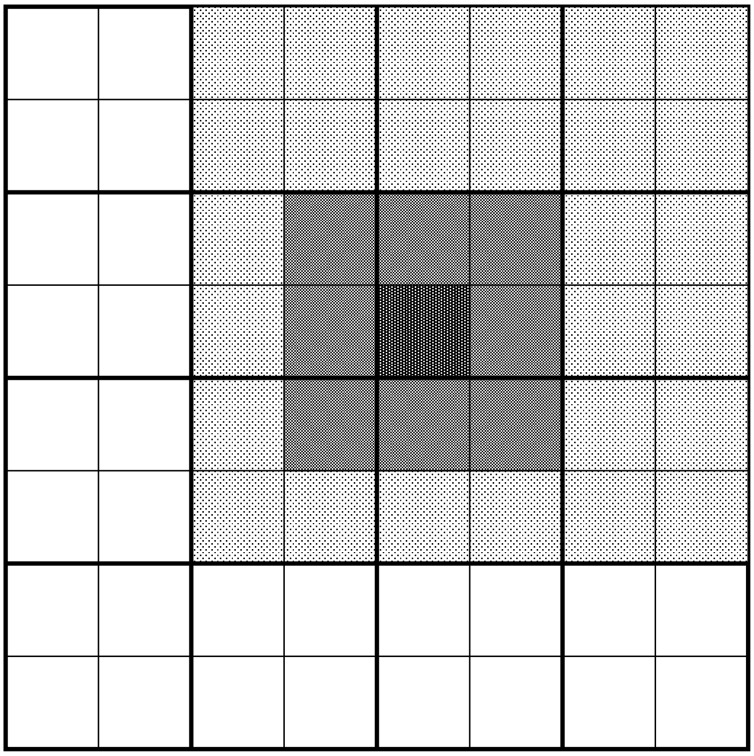
FMM の計算手順

- 各セルが外に作るポテンシャルの球面調和関数展開（多重極展開）を、自分の下のセルの多重極展開の展開中心をシフトして求める。（upward path）
- 各セルが、
  - 自分と同じレベルで
  - 親の隣接セルの子で
  - 自分の隣接セルでない

  セルによるポテンシャルの和を自分の中心での調和関数展開にしておく（局所展開）
- 一番上のレベルから順に、自分の受けているポテンシャルを子セルに伝える（局所展開の中心をシフトする，downward path））
- 最終的には各粒子が自分の入っているセルの中心での局所展開を評価する。



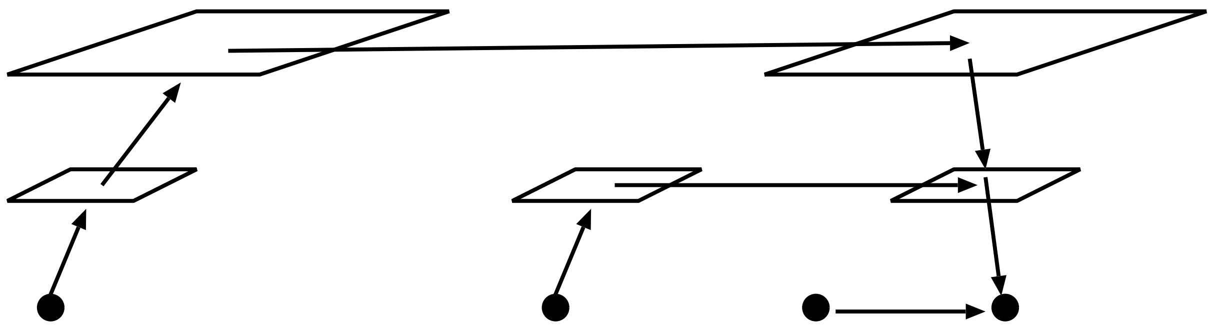
M2L ステージの計算範囲



- 同じレベルのセルとしか相互作用しないので、計算する相互作用の数はセル数に比例
- セル数は粒子数に比例

$\longrightarrow$ 計算量は $O(N)$
他のステージも $O(N)$、係数は小さい。

粒子間相互作用の計算されかた



- 近くの粒子は直接計算
- 遠くにいくほど、上のセルを通って相互作用が計算される。

ツリーとの違い

ツリーでは大きなセルからの力を粒子の位置で直接評価

$\downarrow$

$O(N)$ と $O(N \log N)$ の違い

FMM における計算精度と計算量

- 展開次数を変える
- **M2L** ステージを、 **1** つ離れたところとではなく、それ以上離れたところとで行なう (ツリーで θ を小さくすることに対応)

簡単なオーダー見積り:

$$\text{誤差} \propto \theta^{(p+1)}$$

$$\text{計算量} \propto \theta^{-3} p^4 N$$

計算量は減らせる：ツリーの最下層セルにはいっている粒子の数を n とすれば、計算量のオーダーは

$$Np^4/n + Nn$$

なので、

$$n \propto p^2$$

としておけば

$$\text{計算量} \propto \theta^{-3} p^2 N$$

となって、 $\log N$ がないぶん漸近的にはツリー法よりよい。

歴史（主に天体物理関連）

1981 Appel によるツリー法論文（プリンストン大学卒業論文）

1985 Appel によるツリー法論文出版

1984-85 Jernigan & Porter 研究会等でツリー法発表（論文は 1989）

1986 Barnes & Hut 八分木を使ったツリー法、 Nature 論文

1986-87 Rokhlin & Greengard FMM の 2, 3 次元での定式化

1987 Barnes, Hernquist, Makino 等によるツリー法のベクトル化、 Connection Machine への実装

1987 Zhao によるカーテシアン FMM の Connection Machine への実装 (「最初の」3 次元 FMM)

1987- カルテク並列計算グループによるツリー法の並列化

(ツリー法成熟期)

1992 アンダーソン法の提案

1994 M2L への FFT の利用の提案

1996 M2L への座標回転の利用の提案

1997 M2L への平面波展開の利用の提案

FMM はまだ発展途上（だから使えないというわけではないことに注意）

計算の効率化・高速化

2 方向のアプローチ

- 実装上の効率化
  - ベクトル化、並列化
- アルゴリズムの改良による計算量の減少
  - **FFT,** 回転等

ベクトル化、並列化

ベクトル化、並列化の技法はツリー法も FMM も同じ。

- 共有メモリの機械では容易 (アルゴリズムの並列度は実際上 N)
- メッセージパッシングの機械：効率の良い空間分割が必要。 Salmon & Warren (元カルテク) が様々な方法をテストした。
  - Orthogonal Recursive Bisection
  - Hashed Oct-tree (Morton ordering)
  - Hashed Oct-tree (Peano ordering)

  プログラムは面倒だが、作ってしまえば効率はよい。問題によっては ASCI RED のようなかなり使いにくい機械でもそこそこの性能が出る。
- 分散共有メモリの機械:

  Hashed Oct-tree の順序付けで粒子をならべかえる

  $\downarrow$

  プログラムとしては共有メモリ用のままで効率良く実行できる

  (Stanford DASH での結果)

- 粒子分布が一様に近ければ、空間分割も単に規則格子で切るだけでよくて簡単。一様から遠くなると面倒。

計算量の低減

ツリー法は通常 4 重極まで $\rightarrow$ アルゴリズムをいじる余地少ない。
天文ではもっと高精度が必要な場合には依然直接計算が主流。

FMM：多数提案がある。
ほとんどのものが、なんらかの方法で **M2L** の計算量のオーダーを下げようというもの（ここと近くの直接計算の計算量が主要）

・**FFT** を利用するもの
球面調和関数のシフトは、基本的には多項式の原点移動であり、係数のコンボリューションで表現できる
$\rightarrow$ **FFT** を使って、$O(p^2 \log p)$ で実行可能
現実には、$p = 8$ で **M2L** の速度が **2** 倍程度、アルゴリズム全体では **40%** 程度の高速化

・回転を利用するもの
シフトするまえに、座標軸（**z** 軸）を移動方向に向けておく
$\rightarrow$ **M2L** の計算量が、p^4 から p^3 に減少（角度方向が分離）
現実の効果はまだ未知数

その他にもいくつか提案はある。（理論家のオモチャ？）

ツリーと FMM の優劣：実測

直接に比較した例は非常に少ない。理由は良くわからない。

下の表は Blackston and Suel の結果。計算精度はかなり低い（ポテンシャルの相対精度が 10^{-3} 程度）の場合。計算機は 16 プロセッサの Cray T3E

粒子数	16K	64K	256K	1024K
FMM	1.9	7.5	28.5	103.5
Anderson	1.3	5.0	17.9	67.8
BH tree	0.57	2.7	12.4	54.0

定量的に意味がある比較とはいい難い

プログラムがどの程度チューニングされているかが良くわからない。

明らかなこと：10^6 体位になっても、$O(N)$ だから $O(N \log N)$ よりも速いというわけでもない。

FMM がそれほど良くないのは何故か？

- 誤差の入り方が違う
  - ツリー： 多重極展開を評価する時の誤差：多数のセルからの寄与がある程度打ち消し合う
  - FMM ：局所展開の誤差： 1 つしか項がないので、まるまる残る

- 計算量の前の係数が違う
  - FMM では 1 つのセルが最低 189 個のセルと相互作用
  - ツリーでは $30\theta^{-3}$ 程度
  - ツリーでは $O(p^2)$, FMM では $O(p^4)$

応用によっては FMM はもっと良い
例：

- 2 次元の場合：計算量 $O(Np)$, セル 27 個
- 境界要素法への応用のように、問題は 3 次元でも要素が分布している領域は 2 次元の場合：空でないセル数は減る

アンダーソンの方法と Pseudoparticle method

FMM、高精度ツリーの「実用上の欠点」

- 数学が結構ややこしい：実装、デバッグが大変、高速化も面倒 (球面調和関数の計算：漸化式なので速くするのが難しい)

もう少し簡単に書けないか？
→ FMM without Multipoles

- アンダーソンの方法 (Anderson 1992)
- 疑似粒子法 (Makino 1998)

その前に： FMM の数学の概要

多重極展開の算数： 3 次元の場合

展開係数

$$\alpha_l^m = \sum_{i=1}^{N} m_i r_i^l Y_l^{-m}(\theta_i, \phi_i)$$

球面調和関数

$$Y_l^m = (-1)^m \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) e^{im\phi}$$

多重極展開

$$\Phi(r,\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{\alpha_l^m}{r^{l+1}} Y_l^m(\theta,\phi).$$

局所展開

$$\Phi(r,\theta,\phi) = \Phi(0) + \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \beta_l^m r^l Y_l^m(\theta,\phi). \tag{1}$$

計算コードの実現には、

- α_k から β_k への変換
- α_k の展開中心のシフト
- β_k の展開中心のシフト

が必要

多重極展開の算数： 3 次元の変換式

$$\mathbf{M}_{n'}^{m'} = \sum_{n,m} \mathbf{T}_{n,m,n',m'}^{M2M}(\vec{\rho})\mathbf{M}_n^m$$

$$\mathbf{T}_{n,m,n',m'}^{M2M}(\vec{\rho}) = \begin{cases} \frac{(-1)^{n'-n} A_n^m A_{n'-n}^{m'-m}}{A_{n'}^{m'}} Y_{n'-n}^{*m'-m}(\alpha,\beta)\rho^{n'-n}, & \text{if } n' \ge n \text{ and} \\ & \text{if } |m'-m| \le n'-n \\ 0, & \text{otherwise} \end{cases}$$

$$A_n^m = \frac{1}{\sqrt{(n+m)!(n-m)!}}$$

$$\mathbf{L}_{n'}^{m'} = \sum_{n,m} \mathbf{T}_{n,m,n',m'}^{M2L}(\vec{\rho})\mathbf{M}_n^m$$

$$\mathbf{T}_{n,m,n',m'}^{M2L}(\vec{\rho}) = \frac{(-1)^{n'+m} A_n^m A_{n'}^{m'}}{A_{n'+n}^{m'-m}} \frac{Y_{n'+n}^{*m'-m}(\alpha,\beta)}{\rho^{n'+n+1}}$$

$$\mathbf{L}_{n'}^{m'} = \sum_{n,m} \mathbf{T}_{n,m,n',m'}^{L2L}(\vec{\rho})\mathbf{L}_n^m$$

$$\mathbf{T}_{n,m,n',m'}^{L2L}(\vec{\rho}) = \begin{cases} \frac{A_{n'}^{m'} A_{n-n'}^{m-m'}}{A_n^m} Y_{n-n'}^{m-m'}(\alpha,\beta)\rho^{n-n'}, & \text{if } n \ge n' \text{ and} \\ & \text{if } |m-m'| \le n-n' \\ 0, & \text{otherwise} \end{cases}$$

(From Elliot and Board, 1994)

高速計算は面倒 — Y_n^m が漸化式で計算されるので並列性が低い

アンダーソンの方法

“FMM without multipole expansion”

ポアソンの公式：球面で境界条件を与え、中（または外）でのラプラス方程式の解を与える

2 次元の場合:（外側）

$$\phi(r,\theta) = GM\log(r) + \frac{1}{2\pi}\int_0^{2\pi}\phi(a,s)\left[\frac{1-(a/r)^2}{1-2(a/r)\cos(\theta-s)+(a/r)^2}\right]ds,$$

3 次元（球面調和関数展開での形）

$$\Psi(\boldsymbol{x}) = \frac{1}{4\pi}\int_S\left[\sum_{n=0}^{\infty}(2n+1)\left(\frac{a}{r}\right)^{n+1}\mathrm{P}_n(\boldsymbol{s}\cdot\boldsymbol{x}/|x|)\right]\Psi(a\boldsymbol{s})ds,$$

内側

$$\Psi(\boldsymbol{x}) = \frac{1}{4\pi}\int_S\left[\sum_{n=0}^{\infty}(2n+1)\left(\frac{r}{a}\right)^{n}\mathrm{P}_n(\boldsymbol{s}\cdot\boldsymbol{x}/|x|)\right]\Psi(a\boldsymbol{s})ds,$$

上の積分を円周（2 次元の場合）または球面（3 次元の場合）上の適当な標本点をつかった数値積分に置き換える

↓

調和関数展開の操作がすべてポテンシャルの値の操作に置き換えられる

前のページの長い式の代わりに、上の式 2 本で全部済む。

標本点のとりかた

2 次元の場合：単に等間隔でいい。
これで調和関数展開 (結局フーリエ展開) の直交性が実現される

3 次元：球面調和関数の直交性を保つような最適配置をする必要がある

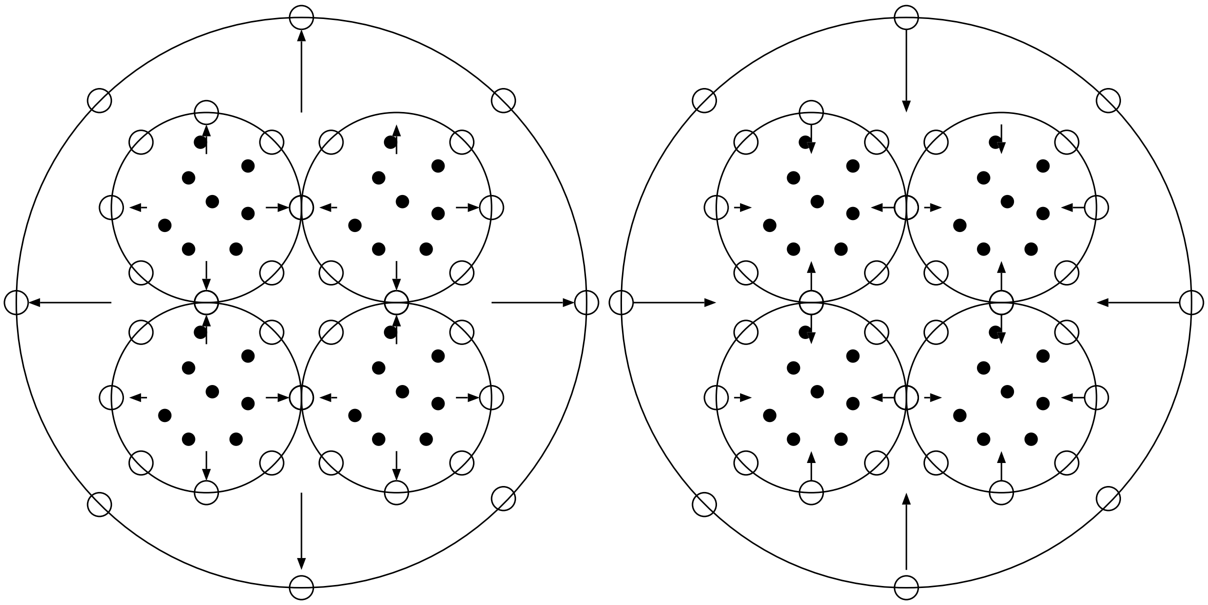
そのような配置： **“spherical** t**-design”**

球面上の点が **spherical** t**-design** をなす
$=$ t 次までの任意の多項式の球面上での平均値が、それらの点でのその多項式の値の平均に等しい。
この時、 $t/2$ 次までの球面調和関数の直交性が保てることになる。

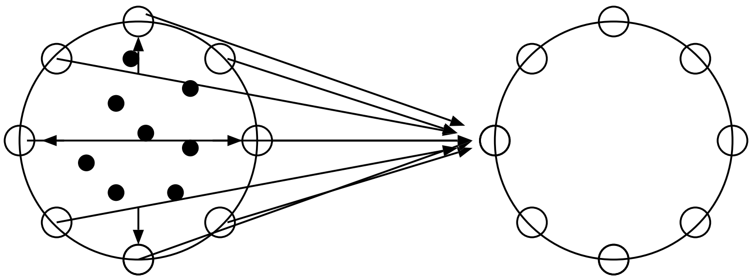
例：正 **4** 面体の頂点： **1-design**, 正 **20** 面体の頂点： **4-design**。
20-design (**216**点) といったものも作られている。

アンダーソンの方法の実際

多重極展開（左）と局所展開（右）のシフト



多重極展開から局所展開への変換



アンダーソンの方法の利点と欠点

利点

- プログラムが非常に簡単になる
- 次数等が同じなら普通の FMM より速い

但し、演算数を数えると普通の FMM より増えているはずなので、なぜ速くなっているのか（適切な評価かどうか）はちょっと疑問。

欠点

- 誤差、収束性が怪しい (パラメータをうまくとらないと誤差が大きい)

Pseudoparticle method

基本的な考え：

球面調和関数展開：多数の粒子が作るポテンシャル場の近似

$\downarrow$

逆に球面調和関数展開を必要最小限の数の粒子分布で表現できないか?

こういうことを考える理由

- 意味がわかりやすい
- 重力多体問題専用計算機 (GRAPE) が有効に使える

問題：どうやって配置を決めるか？

粒子配置の考え方

- 自由度の数から考える

  粒子 1 つの自由度： 4 個

  p 次までの球面調和関数の全自由度： $(p+1)^2$

  $\longrightarrow$ 例えば 3 個の粒子で $p=2$ を表現出来る???

  – 非線形の連立方程式を解かないといけない

  – 有意味な解があるかどうか良くわからない

- 粒子の空間配置は固定して、質量だけ変える

  – (欠点) 必要とする粒子の数が増える

  – (利点) 線形計算で配置を決めることが出来る

  – (利点) いつでも配置を求められる

以下では、粒子を置く座標はあらかじめ決めておくやりかたについて考える。
粒子のおき方：球面上 (2 次元なら円周上) とする。アンダーソンの方法での標本点のとりかたと同じ。

粒子配置の算数： 2 次元の場合

多重極展開：

$$\phi(z) = M\log(z) - \sum_{k=1}^{\infty}\frac{\alpha_k}{k}(a/z)^k,$$

$$\alpha_k = \sum_{i=1}^{N} m_i(z_i/a)^k.$$

半径 a の円周上の質量分布 $\rho(\theta)$ の多重極展開は

$$\alpha_k = (r/a)^k\int_0^{2\pi} e^{ik\theta}\rho(\theta)d\theta,$$

となるので、逆変換

$$\rho(\theta) = \frac{1}{2\pi}\left(\frac{a}{r}\right)^k\sum_{k=0}^{\infty}\alpha_k e^{-ik\theta}.$$

が粒子のおき方を与える。有限個の粒子なら、
$\theta_j = 0, 2\pi/(2p+1), 4\pi/(2p+1)...$ として、 m_j が

$$m_j = \frac{1}{2p+1}\left(\frac{a}{r}\right)^k\sum_{k=0}^{p}\alpha_k e^{-ik\theta_j}.$$

もとの粒子分布からいきなり求めると、

$$\begin{aligned} m_j &= \frac{1}{2p+1}\sum_{k=0}^{p}\sum_{i=1}^{n} m_i(z_i/z_j)^k \\ &= \frac{1}{2p+1}\sum_{i=1}^{n} m_i\frac{1-(z_i/z_j)^{p+1}}{1-z_i/z_j} \end{aligned}$$

ある点の回りの疑似粒子分布のシフトにも、この式がそのまま使える。

粒子配置の算数： **3** 次元の場合

多重極展開：

$$\alpha_l^m = \sum_{i=1}^{N} m_i r_i^l Y_l^{-m}(\theta_i, \phi_i)$$

半径 a の球面上の質量分布の多重極展開：

$$\alpha_l^m = a^l \int_S \rho(\theta, \phi) Y_l^{-m}(\theta, \phi) dS,$$

逆変換

$$\rho = \sum_{l=0}^{\infty} a^{-l} \sum_{m=-l}^{l} \alpha_l^m Y^*{}_l^{-m}$$

K 個の点での質量なら

$$m_j = \frac{1}{4\pi K} \sum_{l=0}^{p} \sum_{m=-l}^{l} \alpha_l^m Y^*{}_l^{-m}$$

2 次元の場合と同様に粒子分布から直接求めると、

$$m_j = \frac{2l+1}{K} \sum_{i=1}^{N} m_i \sum_{l=0}^{p} (r_i/r)^l P_l(\cos\gamma)$$

ここで γ は粒子の位置と疑似粒子の位置の間の角度。また、球面調和関数の加法定理

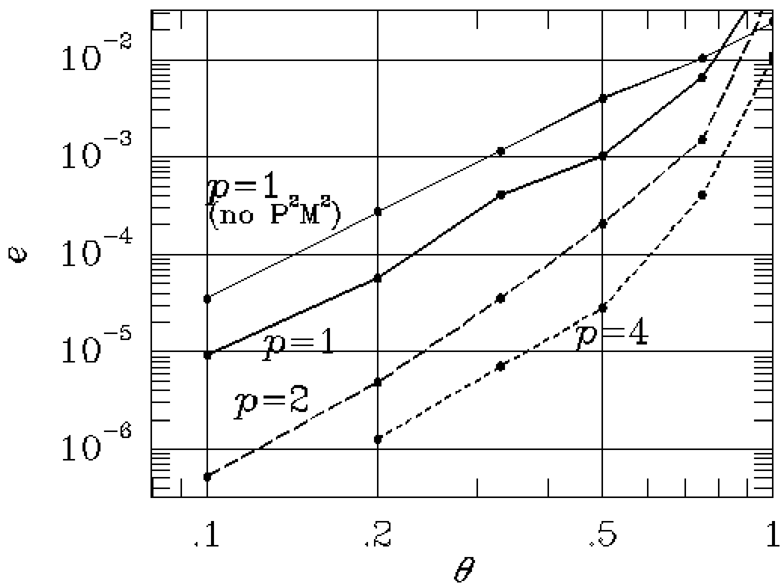
$$P_l(\cos\gamma) = \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_l^m(\theta, \phi) Y_l^{-m}(\theta', \phi')$$

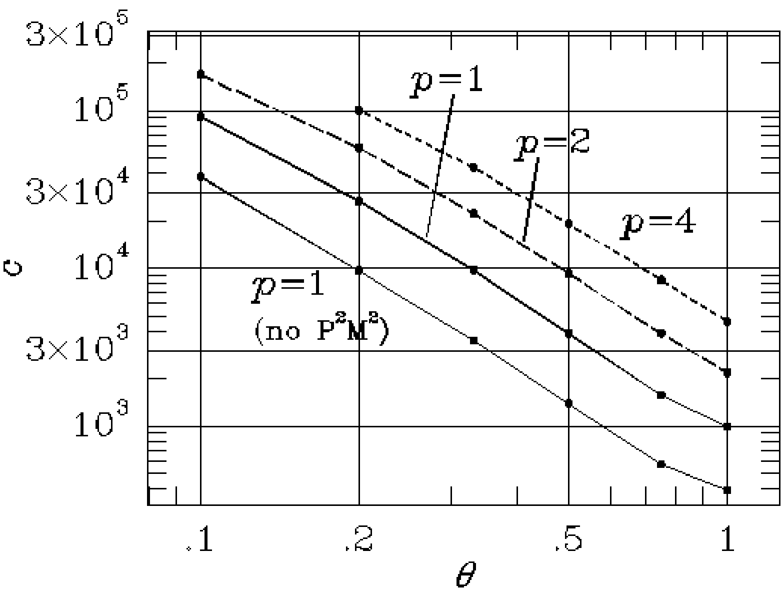
を使った。

ツリーコードへの応用

Barnes-Hut ツリーコードのプログラムに疑似粒子法を組み込んだ。

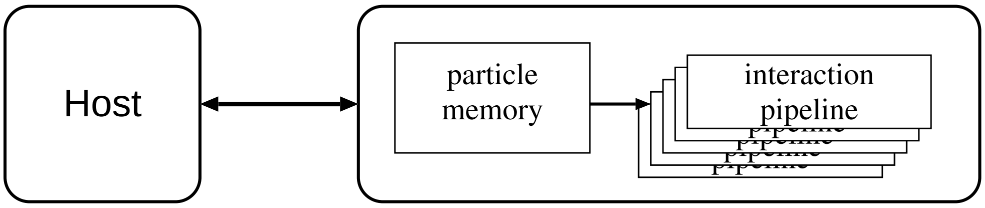
計算精度、計算量の例





専用計算機との組合わせ

GRAPE: 粒子間の重力相互作用だけを高速に計算する専用計算機



ホストがメモリに粒子データを書く

↓

ホストがパイプラインプロセッサのレジスタに粒子データを書く

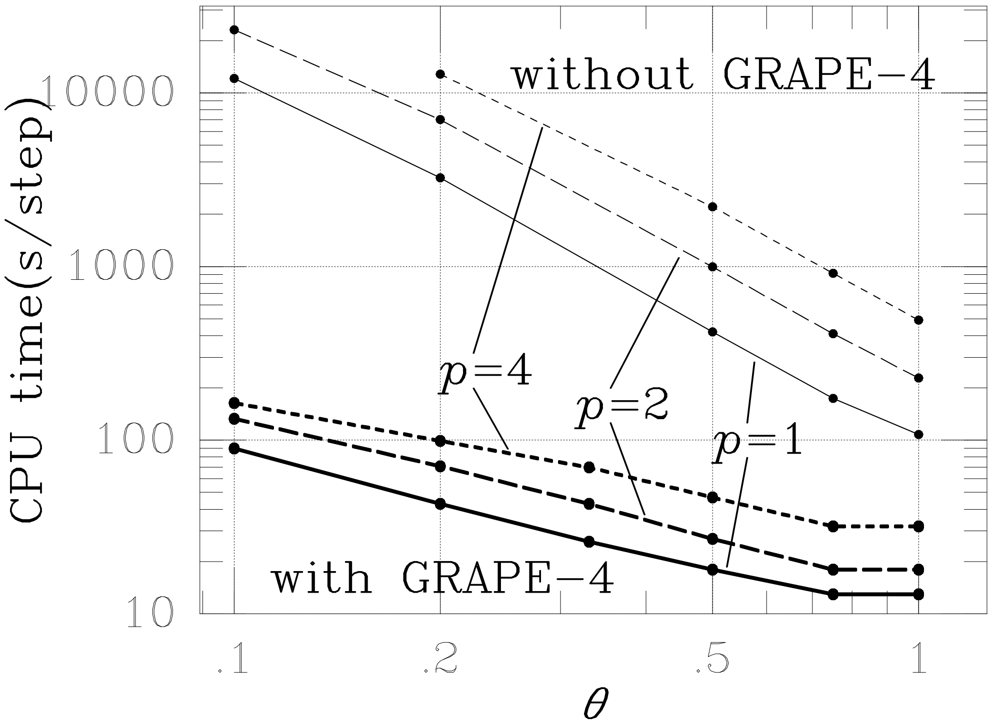
↓

パイプラインプロセッサがメモリ内の粒子からプロセッサが持つ粒子への重力を計算

通常のツリー、 **FMM:** 粒子間相互作用（モノポール）の計算にしか使えない

疑似粒子法：セルからの重力の計算にも使える

疑似粒子法の計算速度



GRAPE なし： DEC Alpha EV5 300 MHz （名目ピーク性能 600 M flops）

GRAPE ： GRAPE-4 （名目ピーク性能 15 Gflops）

粒子数： 262144

GRAPE を使うことで、 10-100 倍の高速化が可能。

多体問題以外への応用

FMM, ツリーの基本的な考え：多数の要素間の遠隔相互作用を、
「遠くのものはまとめる」という近似を使うことで計算量を減らす

$\downarrow$

遠くの要素ををまとめられるものであれば何にでも使える

- 重力・クーロン力以外のほとんどすべての 2 粒子間相互作用
- 境界要素法
- ラジオシティ法
- その他、、、

まとめ

- ツリー法と FMM の原理と、最近の研究の進展について概説した。
- ツリー法は、特に高精度を要求しない分野では広く使われており、並列化等の実用的な研究が進んでいる。
- FMM は、理論的なスケーリングはツリー法よりもよいが、実際の比較では大きな差はでていない。
- FMM については、計算量のスケーリングを減らす方法がいろいろ提案されている（決定版はない）。
- アンダーソンの方法や疑似粒子法では、計算量は減らないが実装を著しく簡単化できる。
- FMM, ツリー法の基本的な考え方は、様々な問題に応用できる。