重力多体計算のアルゴリズム

牧野淳一郎
(東京大学理学系研究科)

講演概要

- 基礎方程式
- なにが大変か ？
- 計算例
- 計算法 — 時間領域
- 計算法 — 空間領域
- まとめ

基礎方程式

$$\frac{d^2 x_i}{dt^2} = -\sum_{j \neq i} G m_j \frac{x_j - x_i}{|x_j - x_i|}$$

太陽系およびそれよりも大きなシステムを記述する基本方程式
（流体力学、輻放、磁場、、、といったものが重要でないとき）

- 要するに、ある星　(惑星、小惑星、あるいはダークマター粒子、、、) は、系の他のすべての粒子からの重力を受けて運動するということ。
- ほとんどの場合に相対論的効果は無視できる。
- 解析的には解けない — 数値積分が必要。

なにが大変か ？

…問題による。

大雑把には、粒子数 N と積分時間 T (典型的な軌道周期で測った) で決まる。

極端な例 (1) — 宇宙の大規模構造

$N \sim 10^{10}, T \sim 1$

極端な例 (2) — 太陽系の超長期安定性 (太陽系は 50 億年存在できたのか ？)

$N \sim 10, T \sim 10^{10}$

中間的な例 :球状星団、惑星形成

$N \sim 10^{6}, T \sim 10^{6}$

何が大変か ？(続き)

N が大きい時 :相互作用計算が大変 (時間積分は短い = それほど特別なことは必要ない)

N が小さい時 :時間積分が大変 (粒子数が小さい = 相互作用計算はあまり時間がかからない)

N, T が中くらいの時 :両方が大変

「大計算」の例

- 宇宙論
  ASCI RED 4096 processor, Barnes and Hut treecode, $N = 3.2 \times 10^8$, 300 timesteps, 10 hours
- 球状星団の進化
  GRAPE-4 1 cluster (peak 270 Gflops), direct summation with individual timestep, $N = 32768$, (effectively) 3×10^6 timsteps, 2 months
- 太陽系の超長期安定性
  various RISC/Intel box, MVS (Mixed variable Symplectic), $N = 10$, (effectively) 10^9 timsteps, 数カ月、、、

計算法 ── 時間領域

基本的には、大規模な常微分方程式の初期値問題。

しかし、普通のライブラリにあるような方法 (高次ルンゲクッタ / 線形多段階法) は、そのままではほとんど使いものにならない。

重力は特徴的なスケールを持たない引力

↓

現れる距離、時間スケールに制限がない。

2つのキーアルゴリズム

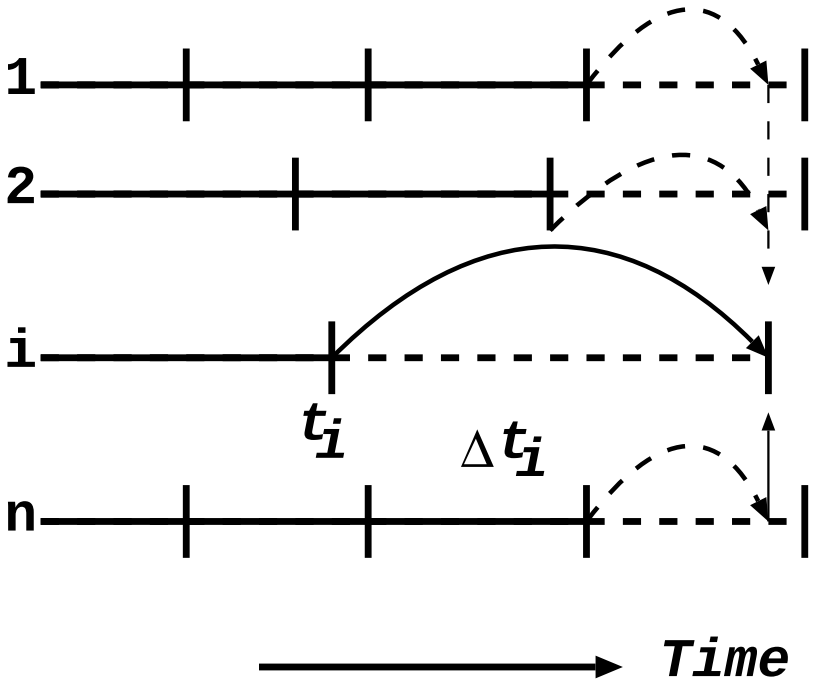
- 独立時間刻み
  - 近接散乱 $\longrightarrow$ 短いタイムステップ
  - 高密度中心核 $\longrightarrow$ 短いタイムステップ

  通常の適応時間刻みに比べた加速: $O(N^{1/3} \sim N)$.
- 近接散乱や連星系のための特別な扱い：
  - 独立時間刻みだけでは近接散乱の全てをカバーできない（桁落ちが問題になる）
  - 準安定な連星系（および、準安定な階層的なサブシステム）:摂動論的扱いが必要。

  計算精度と計算時間の両方に大きな効果（これらなしでは計算不可能）

独立時間刻み

(Aarseth 1963)



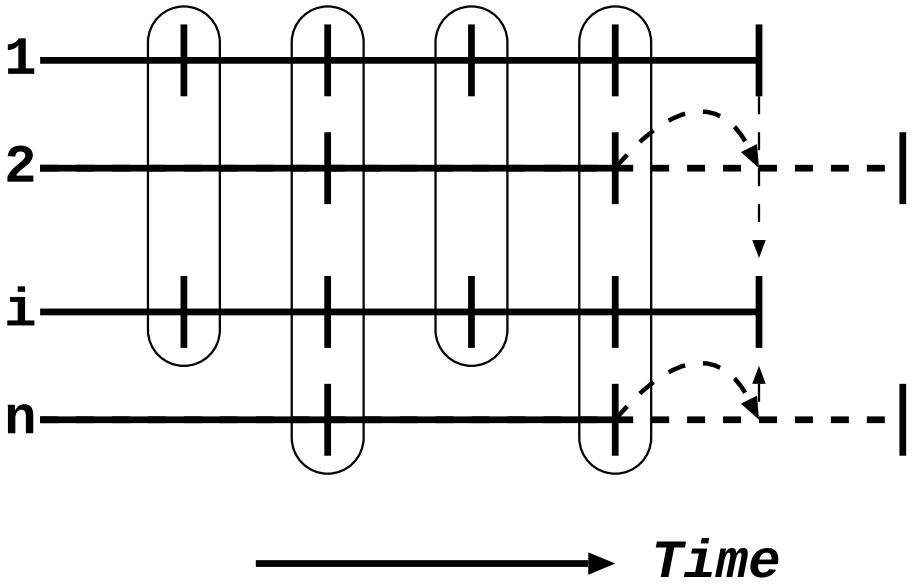
- 各粒子にそれぞれ時刻と時間刻みを与える
- 「イベント駆動」時間積分 — $t_i + \Delta t_i$ がもっとも小さい粒子が積分される

問題点

- 並列化が困難 (一度に一粒子)

ブロック時間刻み法

(McMillan 1986) ベクター / パラレル計算機のための改良
最初は Cyber205 のために導入された



- 時間刻みを 2^{-k} に制限する。
- $t_i + \Delta t_i$ が同じ (Δt_i は違ってもよい) 粒子は同時に積分される。

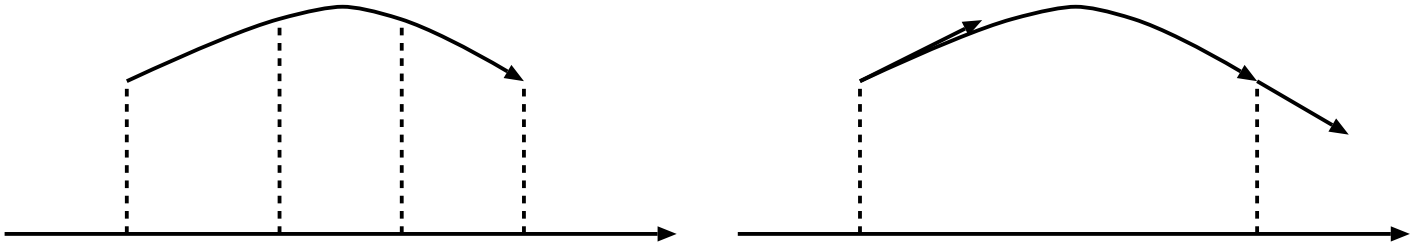
$O(N_c^{2/3})$ 個の粒子 (N_c は高密度コアのなかの粒子数) を並列計算

— そんなに大きな数ではない。

時間積分公式

粒子が同期していない→ 可変刻み多段階法以外の選択肢は事実上ない。
次数 :4 次くらいが適当 (Makino 1990)。

- Aarseth scheme (Aarseth 1963): 可変刻みのアダムス法、 PEC モード、4 次、2 階の方程式用。
- Hermite scheme (Makino 1990): ラグランジュ補間 (ニュートン補間) の代わりに、加速度の一階時間導関数も使ってエルミート補間を構成。



連星等の特別な扱い

問題 ：

- 桁落ち $|x_i| >> |x_i - x_j|$
- (準) 安定な連星等 :直接積分では計算時間がかかり過ぎる。

解決法 — 主に Aarseth とその共同研究者らによる

桁落ち → 相対座標 (と正則化) を使う

連星等 → 「孤立した」系では積分をとばす。弱い摂動を受けている系では摂動論的扱い、、、

(もちろん、ほとんど安定だけれども孤立しているとはみなせないところにもっとも計算時間がかかる、、、)

今後の方向 （？）

- シンプレクティック積分スキーム、時間対称型スキームとの組み合わせ
  - 潜在的な利点 :エネルギー等の時間 1 次の誤差がなくなる
  - 困難 :普通に考えると 「未来」の情報が必要 ？
- 時間方向の並列化 (常微分方程式の初期値問題一般の研究方向)

計算法 — 空間領域

運動方程式の右辺をどうやって評価するか ?という問題。

以下、独立時間刻みのことはとりあえず棚上げにして話をすすめる。
(後で 2 つの組合せについては議論する)

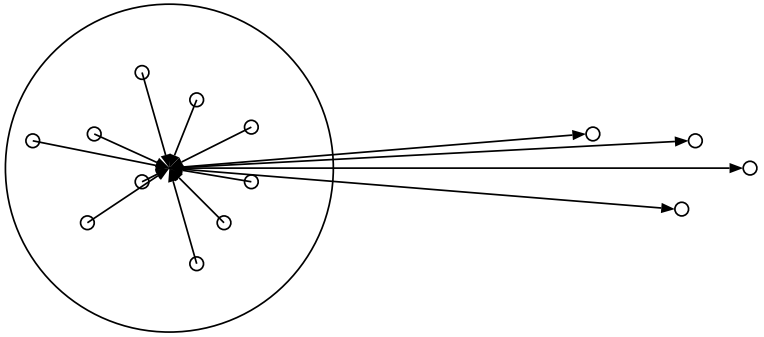
広く使われている方法 : Barnes-Hut treecode
有名な方法 : 高速多重極法

ツリー法、FMMの基本的発想

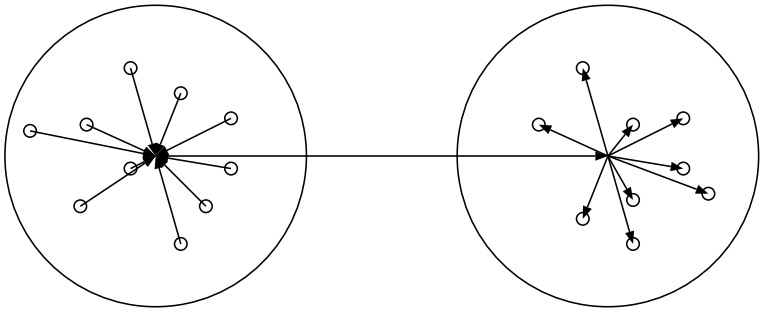
遠くの粒子からの力は弱い

↓

まとめて計算できないか？



Tree

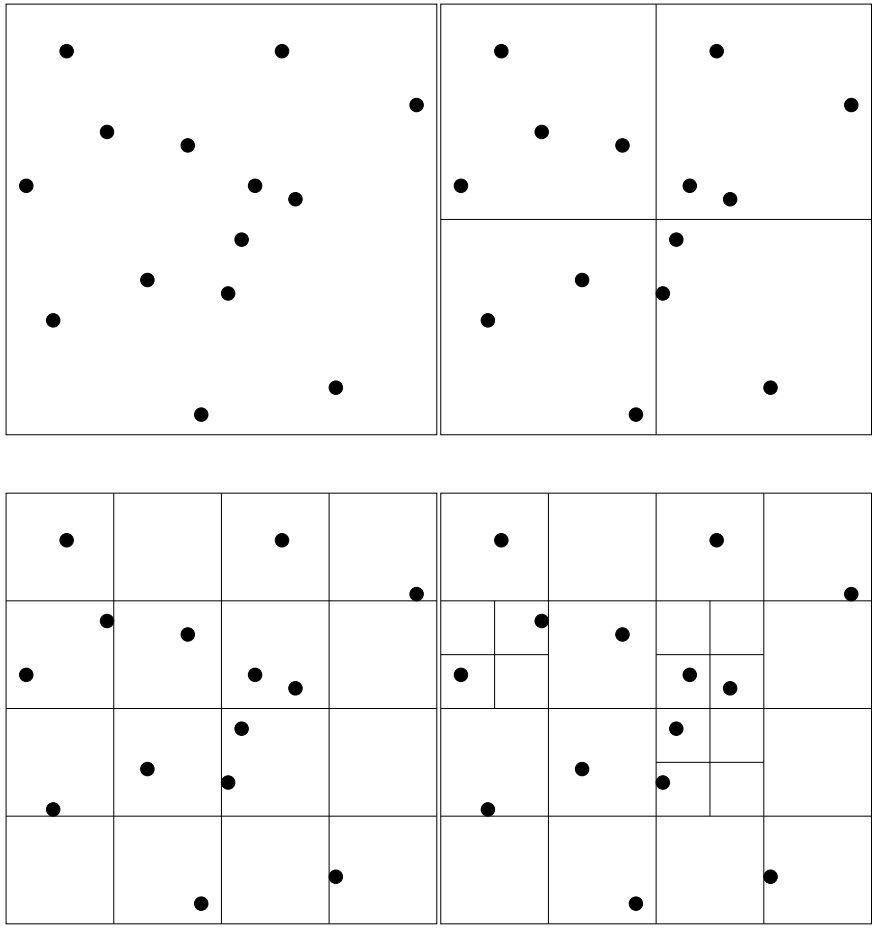


FMM

- 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

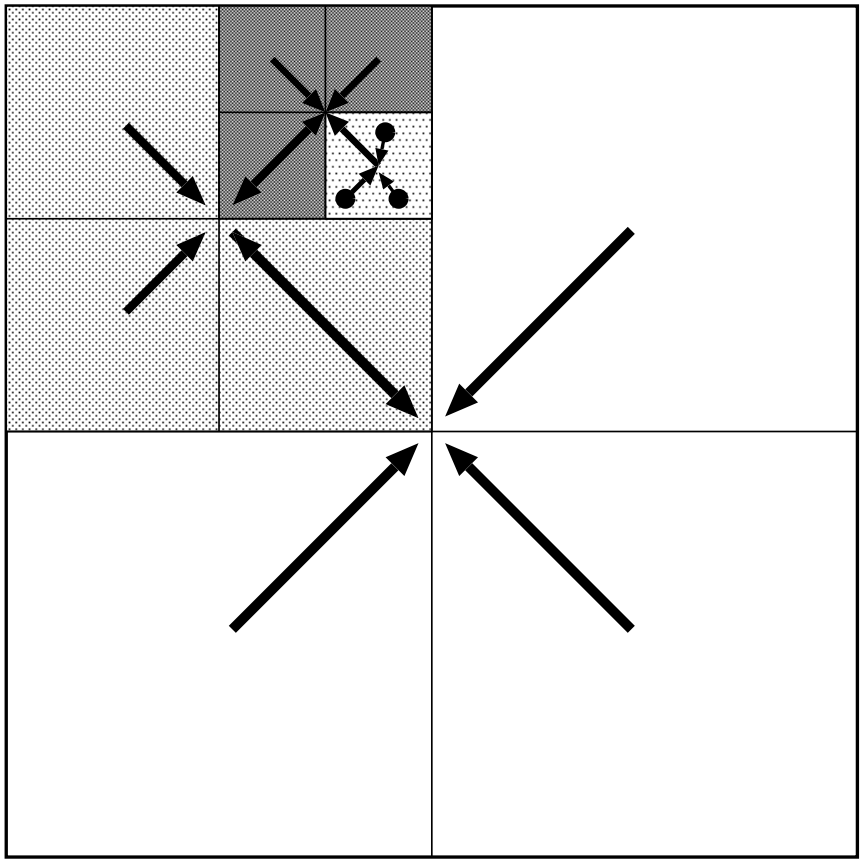
階層的なツリー構造を使う。



- 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
- 中の粒子がある数以下になったら止める (上の例では 1 個)

多重極展開の構成

まず、ツリーの各セルのなかの粒子がつくるポテンシャルの多重極展開を計算する。

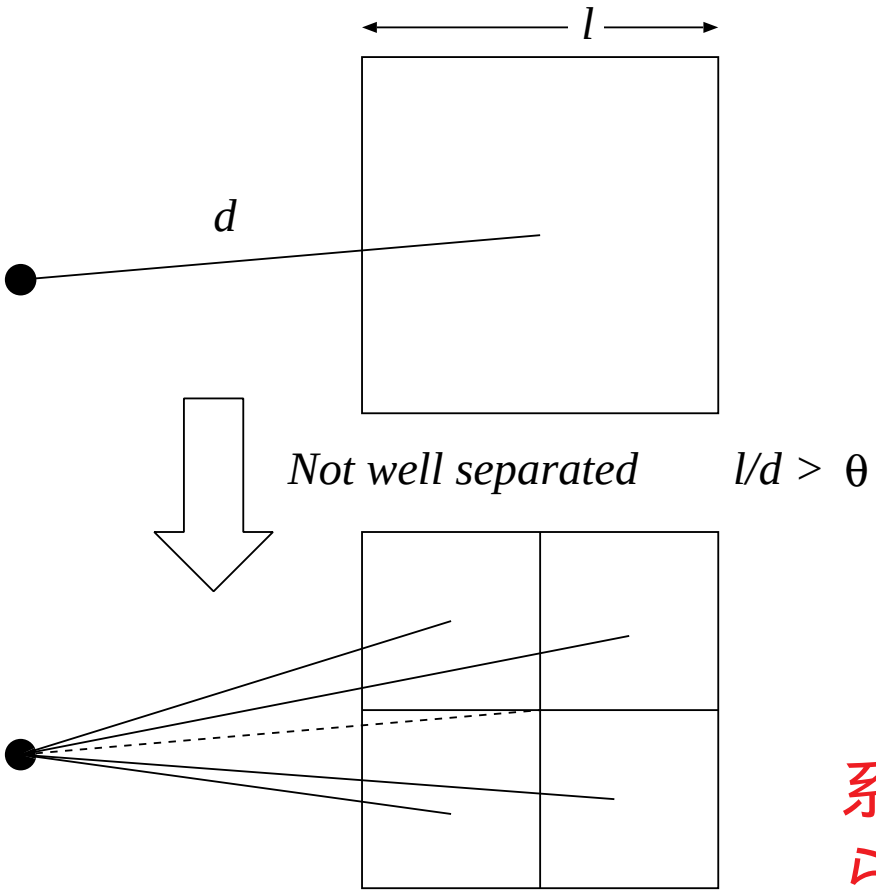


- 最下層のセル:そのなかの粒子が作るポテンシャルを多重極展開
- それ以外　:子セルの多重極展開の展開中心をシフトして加算

下から順に計算していけばよい。
計算量は $O(N)$。展開をシフトする式はかなり複雑。

ツリー法での力の計算

再帰的な形に表現すると格好がいい。



- 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 ＝ ルートからの力

ツリー法の計算量と計算精度

簡単なオーダー見積り:

$$\text{誤差} \propto \theta^{(p+1)}$$

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

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)

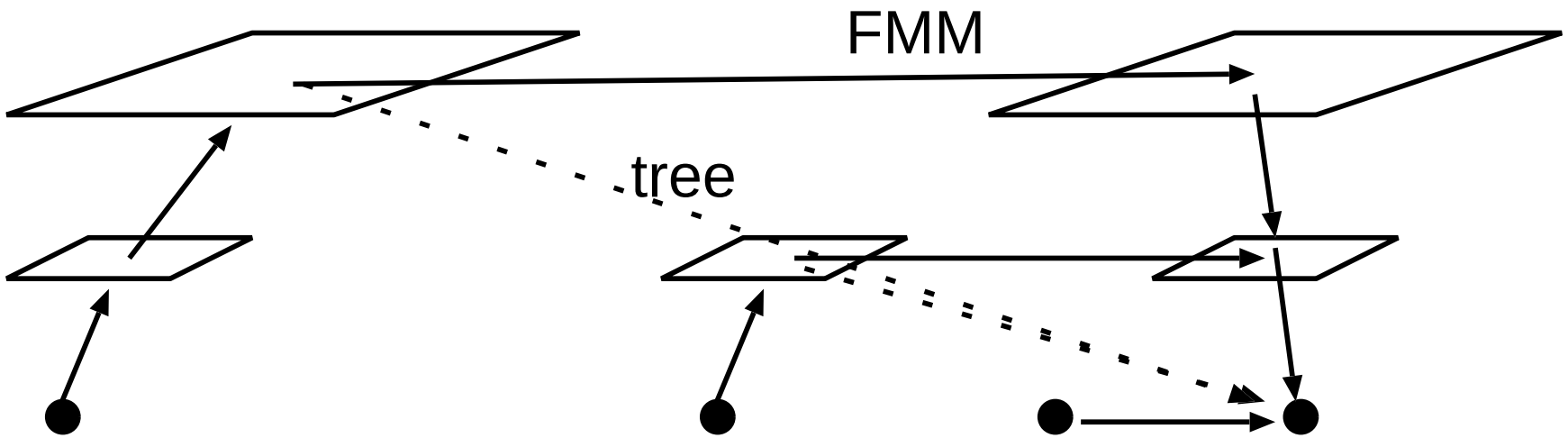
高速多重極展開法(FMM)

- 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　:

計算量が $O(N)$ になる　！

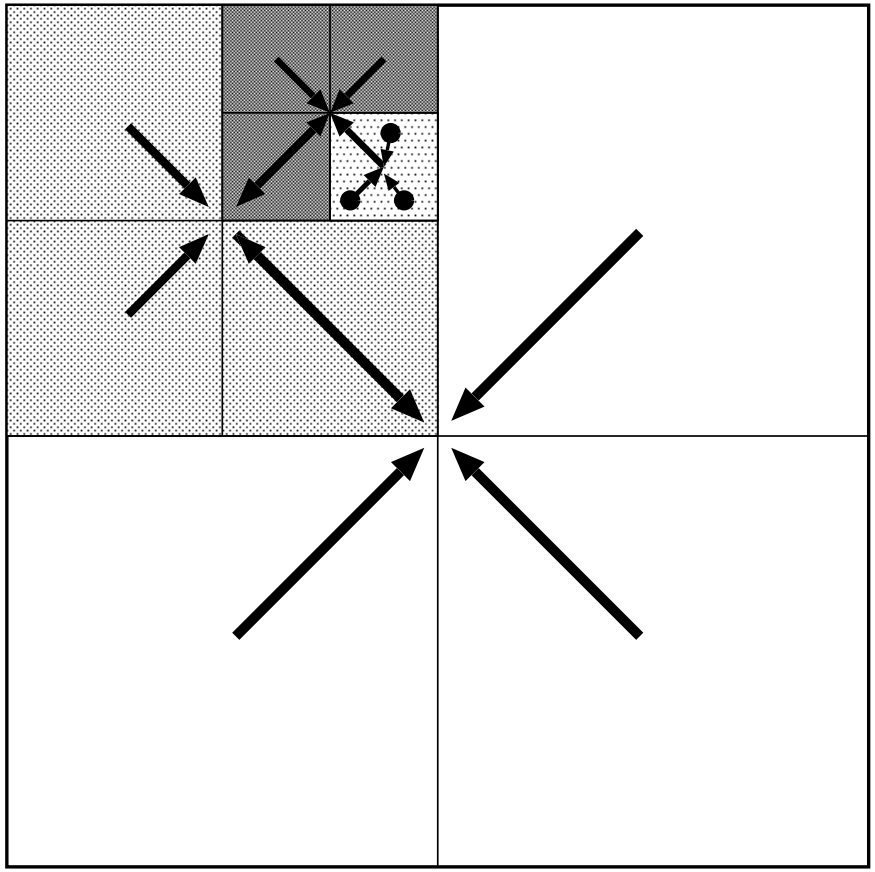
粒子間相互作用の計算されかた



- 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。

FMMの計算手順 - (1) M2M

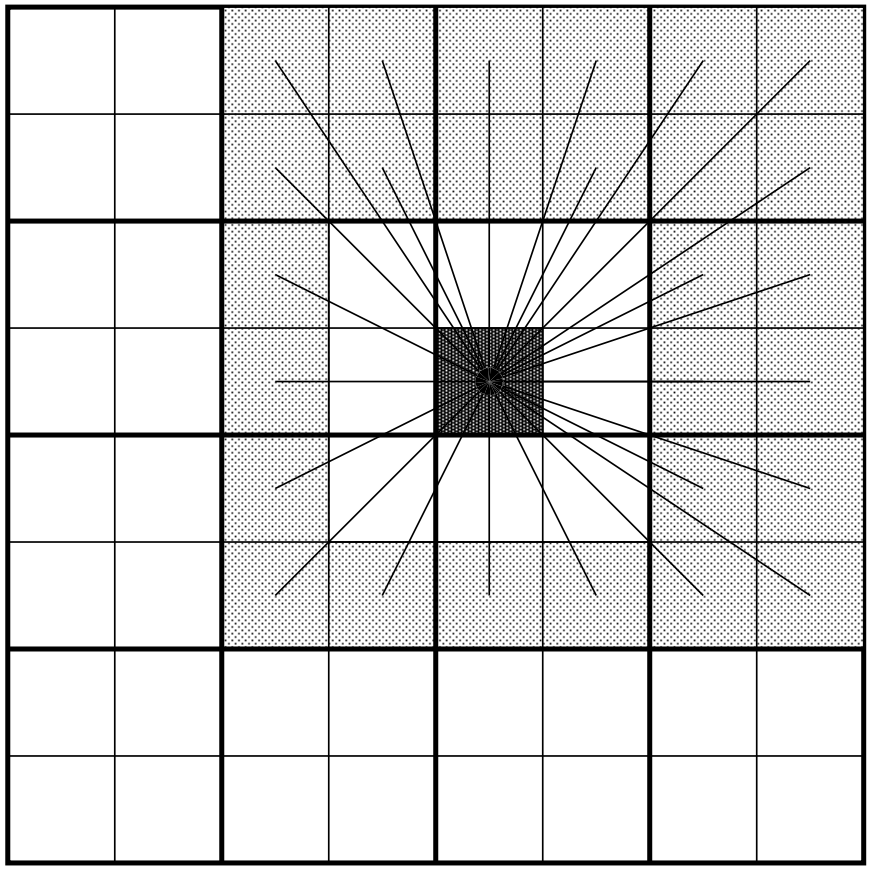
各セルが外に作るポテンシャルの球面調和関数展開（多重極展開）を、自分の下のセルの多重極展開の展開中心をシフトして求める。(M2M)
(ここはツリーと同じ)



FMMの計算手順 - (2) M2L

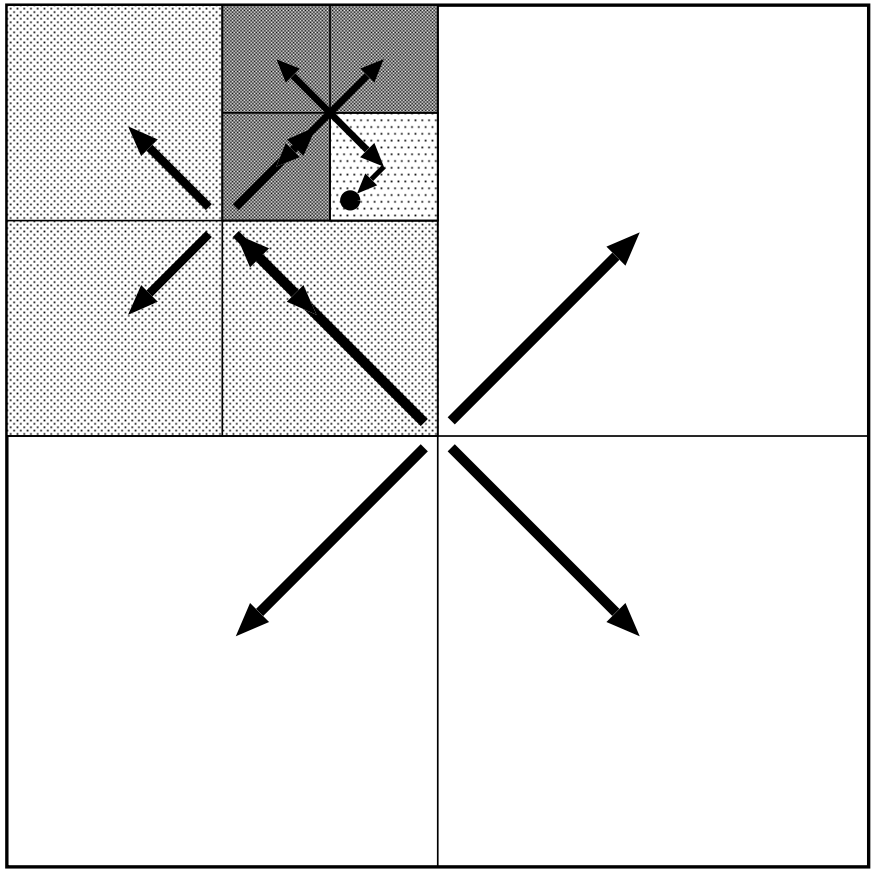
- 自分と同じレベルで
- 親の隣接セルの子で
- 自分の隣接セルでない

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。(M2L)



FMMの計算手順 - (3) L2L

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, L2L)
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。



FMMにおける計算精度と計算量

簡単なオーダー見積り:

$$\text{誤差} \propto \theta^{(p+1)}, \qquad \text{計算量} \propto \theta^{-3} p^4 N$$

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

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

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)

アンダーソンの方法

多重極展開を球面上でのポテンシャルの値で表す。
球面を境界条件とするラプラス方程式→ポアソンの公式

ポアソンの公式 (外部解と内部解)

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

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

積分を球面上の適当な標本点をつかった数値積分に置き換える
→調和関数展開がポテンシャルの値に置き換わる。
→もともとのFMM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。
調和関数展開 (フーリエ展開) の直交性がなり立つ。

3 次元 :球面調和関数の直交性を保つ配置をする必要がある

そのような配置 : “spherical t-design”

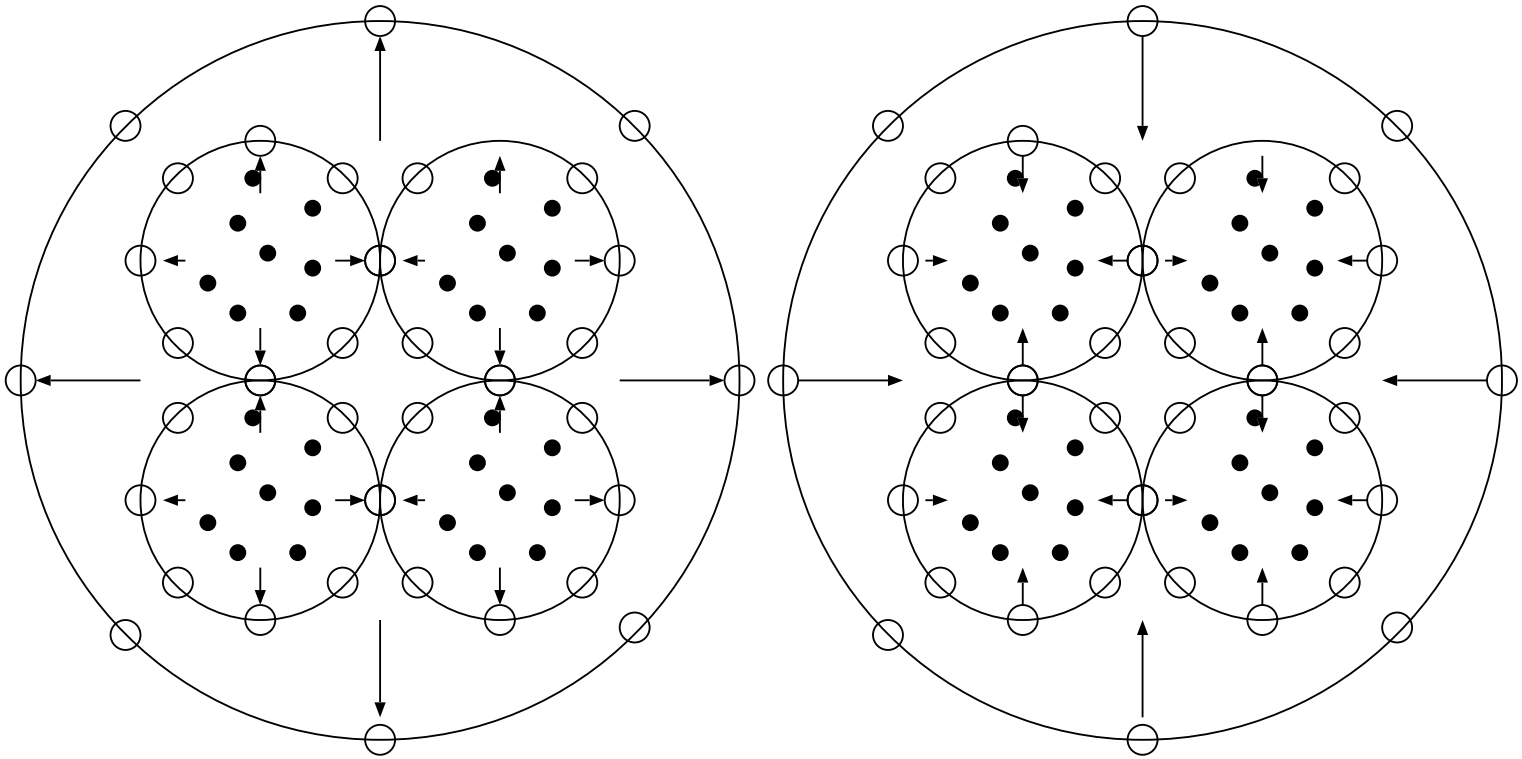
球面上の点が spherical t-design をなす
$=$ t 次までの任意の多項式の球面上での積分が、それらの点上での多項式の値の和 (の $4\pi/N$ 倍) に等しい。

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 (M2M, L2L)

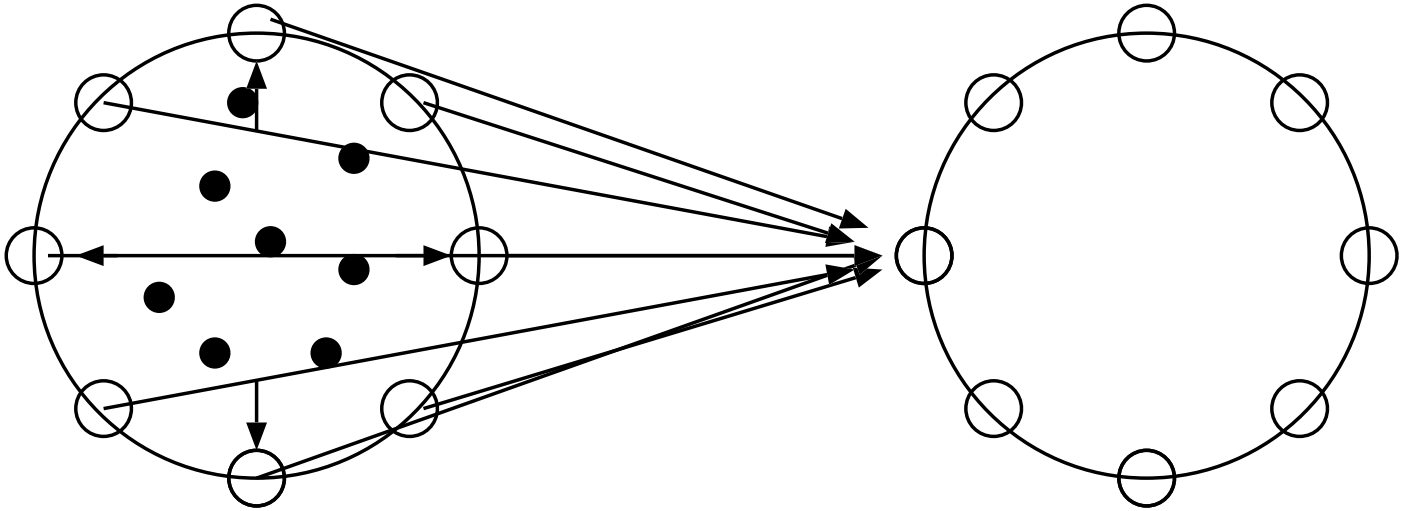
多重極展開 (左) と局所展開 (右) のシフト



(本当は球の半径はもっと大きい、、、立方体セルを完全に含まないといけないので)

アンダーソンの方法の実際 (M2L)

多重極展開から局所展開への変換



M2M, M2L, L2L のどの場合でも、単に各点のポテンシャルの値をポアソンの公式に従って数値積分するだけ。

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、実測では普通の FMM より速いようである

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)
- GRAPE はそのままでは使えない、、、

疑似粒子法

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

↓

逆に球面調和関数展開を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)いつでも線形計算で配置を決めることが出来る

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。

疑似粒子法の数学 (M2M)

疑似粒子のおき方 :アンダーソンの方法と同じく球面上とする。最終的な式 :

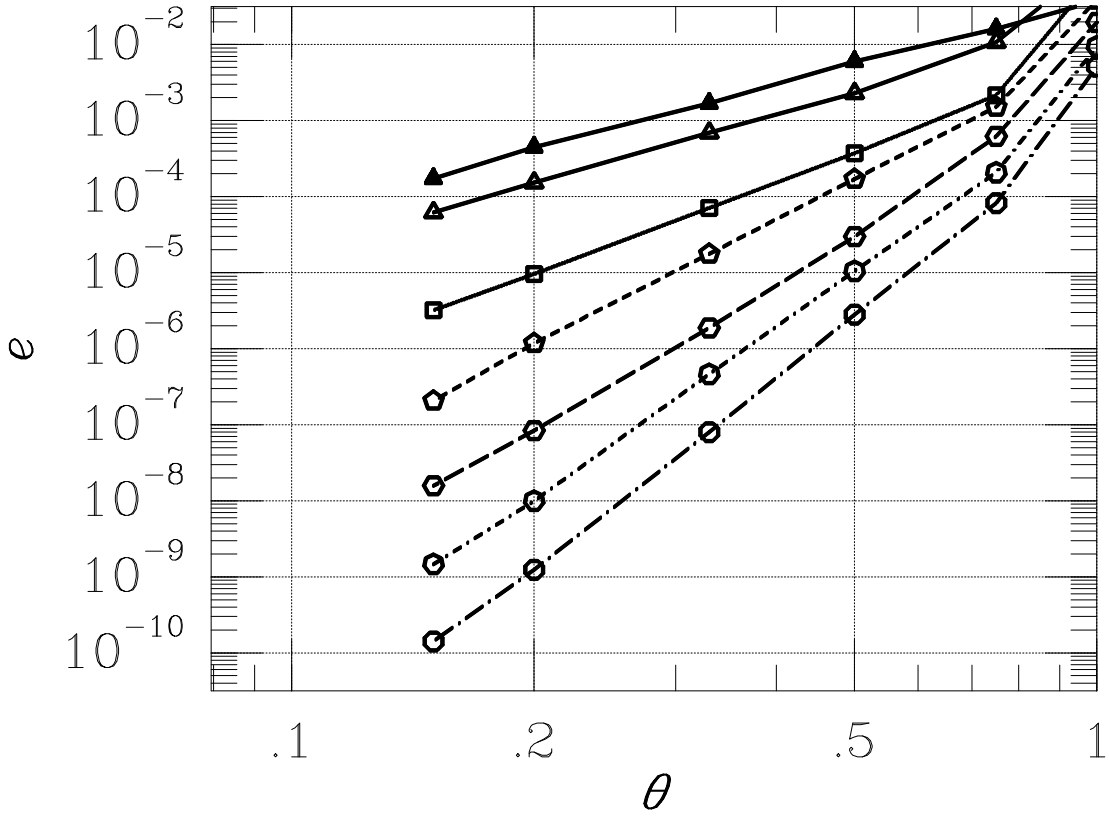
$$m_j = \frac{2l+1}{K} \sum_{i=1}^{N} m_i \sum_{l=0}^{p} (r_i/r)^l P_l(\cos\gamma)$$

ここで γ は粒子の位置と疑似粒子の位置の間の角度。

- 上の式は、実粒子から仮想粒子への変換と仮想粒子から上位セルの仮想粒子への変換の両方に使える。
- 収束性を考えると、球の半径をセルの大きさより少し小さい程度にとっておくのが良い。

ツリー法との組合せ

計算精度の例



粒子分布は球内一様、
26 万体。

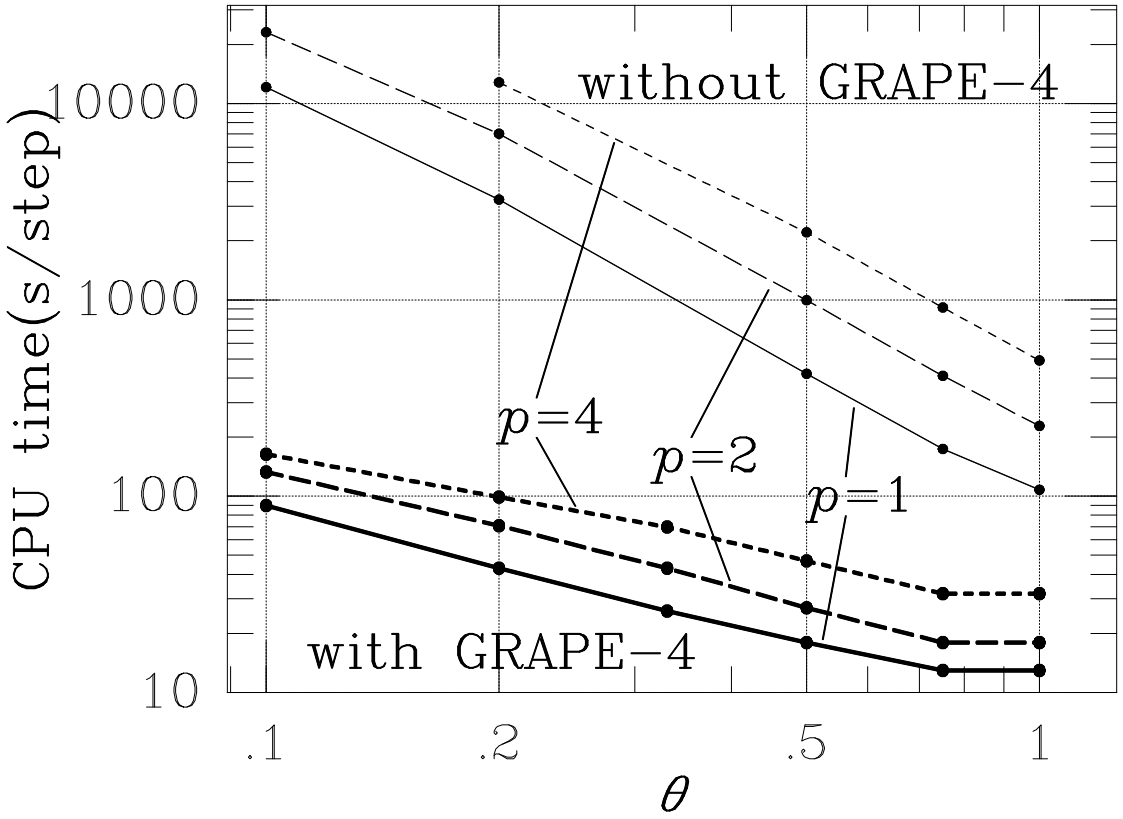
次数は上の 2 つが一次
(定式化が違う) その
後順に 2–6 次。

ツリー ・FMM と GRAPE

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

疑似粒子法は GRAPE で効率的に加速出来る (ツリーでも FMM でも)

実際の計算速度　(ツリーの場合)



GRAPE なし ：
DEC Alpha EV5
300 MHz
GRAPE ：
GRAPE-4
粒子数 ：262144

GRAPE を使うことで、 10-100 倍の高速化が可能。

ツリー / FMM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** が様々な方法をテストした。プログラムは面倒だが、作ってしまえば効率はよい。
- 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
  → 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
- 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。

アルゴリズムの　改良」

ツリー法は通常 4 重極まで → 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。
(独立時間刻みとの組み合わせが困難なため)
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)

FMM の　高速化」

・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

→ 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

→ 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？）

独立時間刻みとツリー、FMM

- 組み合わせることは可能 (McMillan and Aarseth 1993)
- 高精度なもののMPP への実装は存在しない
- GRAPE への実装もない

実装が困難な根本的な理由 :計算量が減った分並列度が下がっているため

今後の大きな課題といえる。

まとめ

- 重力多体計算の主要アルゴリズムである、独立時間刻み、ツリー、FMM について、その概要を述べた。
- それぞれ、単独では並列化もできており、高い性能が得られている
- いくつかの問題では、高精度のツリーまたはFMMを独立時間刻みと組み合わせる必要があるが、そのような方式の並列化は困難である。