もう一つの高速多重極展開法
— Pseudo-particle multipole method

牧野

要約

新しい高速多重極展開法の実現方法を定式化した。特徴は、

- 従来の方法よりもずっと実装が簡単であり
- GRAPE を有効に使うことが出来る

ことである。

話す内容

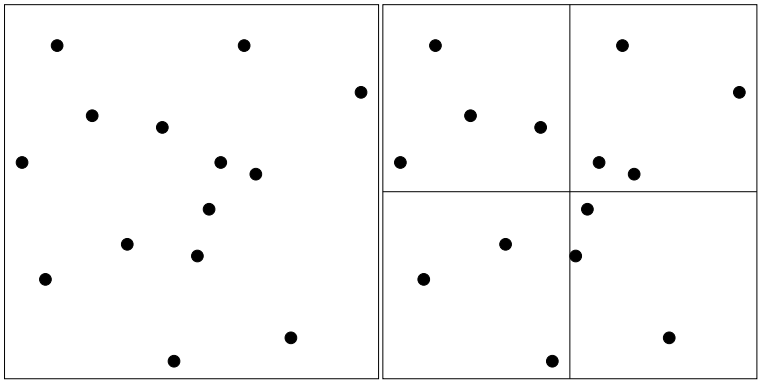
1. 高速多重極展開法とは？
2. アンダーソンの方法
3. Pseudoparticle method
4. 今後するべきこと

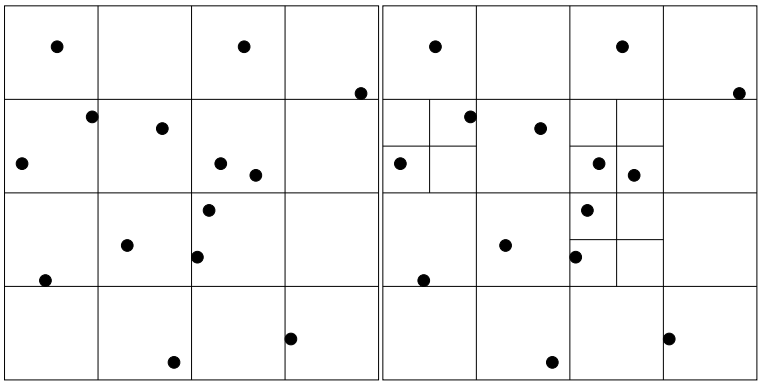
高速多重極展開法とは?

- 理論上は $O(N)$ で N 個の粒子間の重力相互作用を計算できる方法。
- ツリーコードに非常に良く似ている

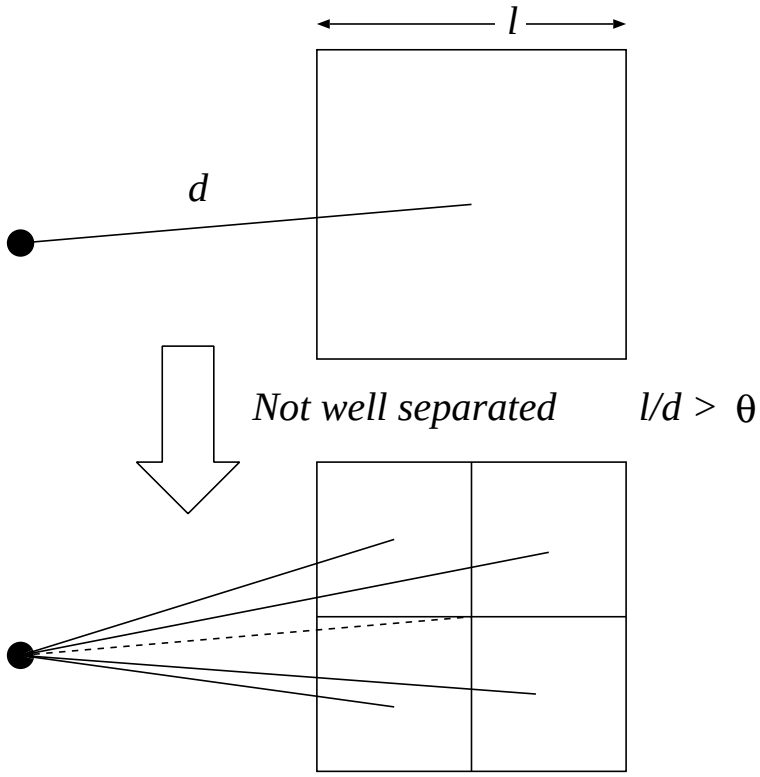
というわけで、まず、ツリーコードの復習：
原理：

- 遠くの粒子からの力はまとめる。
- まとめるのに、階層的なツリー構造を使う。





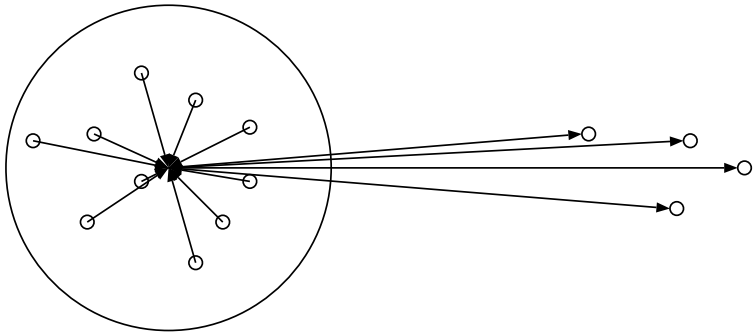
ツリーコードでの力の計算



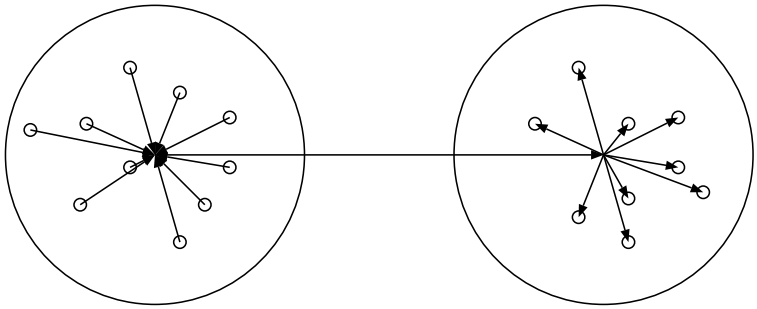
- 十分に離れている：重心（あるいは多重極展開）からの力
- そうでない：子ノードからの力の合計

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。（もちろん、ループの形に書き直せる）

高速多重極展開法



Tree



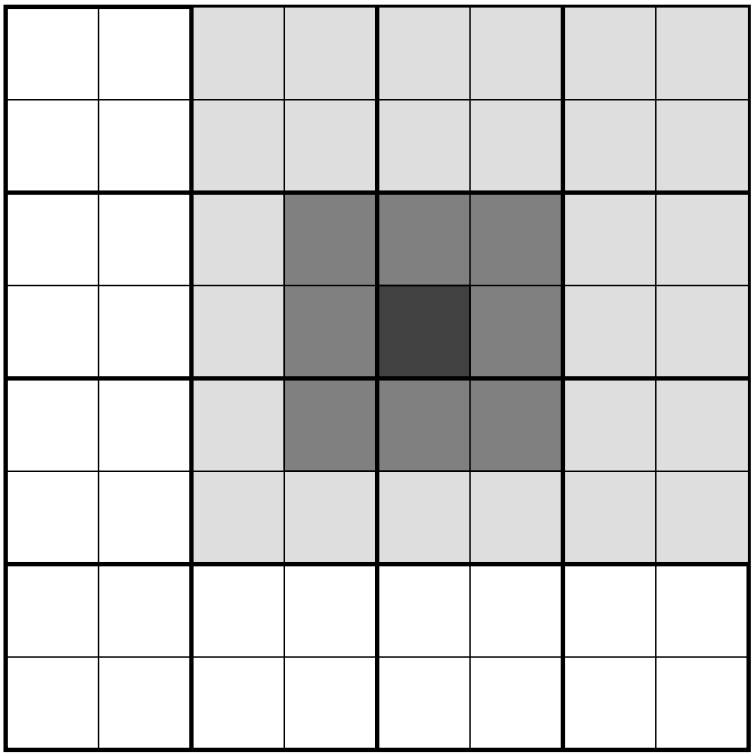
FMM

- 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

力を受ける方の計算法：

1. 規準点でのポテンシャルの（球面）調和関数展開を計算する。
2. 各粒子の位置での力を展開係数から評価する。

なぜ $O(N)$ になるか？



大雑把な話（トポロジカルに計算を分ける場合）

- 自分自身のセル：直接計算する
- 隣接セル：直接計算
- 親セルの隣接セルの子のうち、自分の隣接でない分： 多重極展開で計算する
- 親の親の隣接セルの子のうち、親の隣接でない分： 親の中心での多重極展開で計算する

実際の計算法：

- 各セルが、
  - 自分と同じレベルで
  - 親の隣接セルの子であって
  - 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）
- 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。
- 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）は、あらかじめ下から計算しておく。

多重極展開の算数： 2 次元の場合

粒子 1 個の式：

$$\phi_{z0}(z) = m\log(z - z_0) = m\log(z) - m\sum_{k=1}^{\infty}\frac{(z_0/z)^k}{k},$$

z_0：ポテンシャルを及ぼす粒子の位置
z：ポテンシャルを計算する位置

N 個の粒子

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

M: 全質量
α_k の定義：

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

局所展開

$$\phi(z) - \phi(z_0) = \sum_{k=1}^{\infty}\frac{\beta_k}{k}\left(\frac{z - z_0}{a}\right)^k,$$

α_k から β_k への変換式：ややこしいので省略
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：どれもけっこう面倒

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：どれも非常に面倒

多重極展開の算数： 3 次元の変換式

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

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

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

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

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

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

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

(From Elliot and Board, 1994)

難しいというほどのことはない？

高速計算は困難 — 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

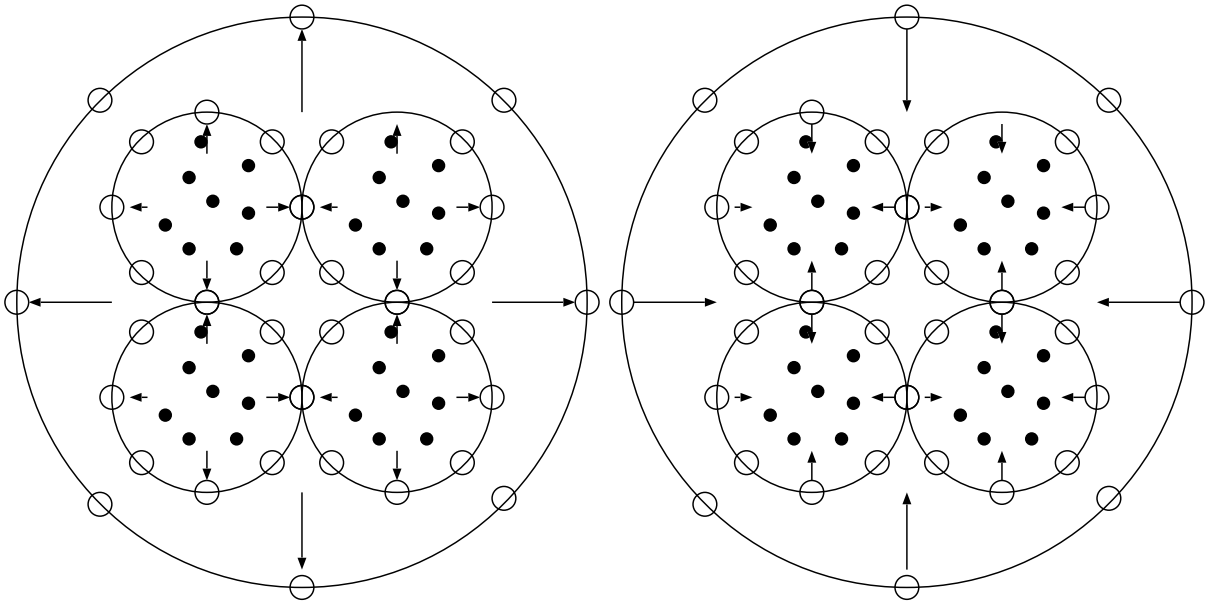
↓

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

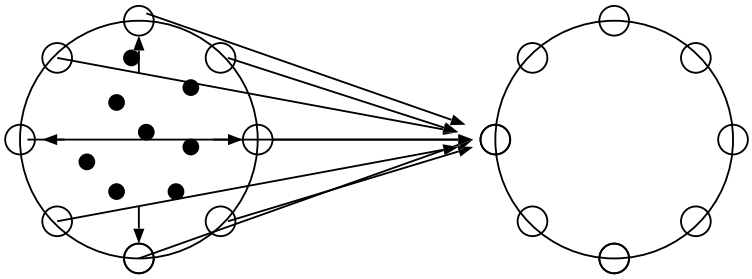
球面での点のとり方：別の研究があって、 13 次の公式 (6 次の球面調和関数まで表現出来る）まで知られている

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

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 を使うという観点からすると別にいいことはない

Pseudoparticle method

動機

- GRAPE を有効に使いたい
- アンダーソンの方法における収束性の問題を解決したい

計算量がもっとも多いところ：多重極展開から局所展開への変換。

↓

多重極展開が「粒子の集まり」で表現されていれば、局所展開への変換は各粒子の作るポテンシャルを評価するだけですむ。（局所展開にはアンダーソンの方法を使う）

↓

GRAPE を使って計算できる *!?*

多重極：粒子（電荷）の配置の極限

→「うまく」粒子を置くのは当然出来るはず。

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： $(p+1)^2$

  $\rightarrow$ 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、とりあえず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。

粒子のおき方の算数： 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。

粒子のおき方の算数： 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

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

（規格化が間違っているような気がする、、、）
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、

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

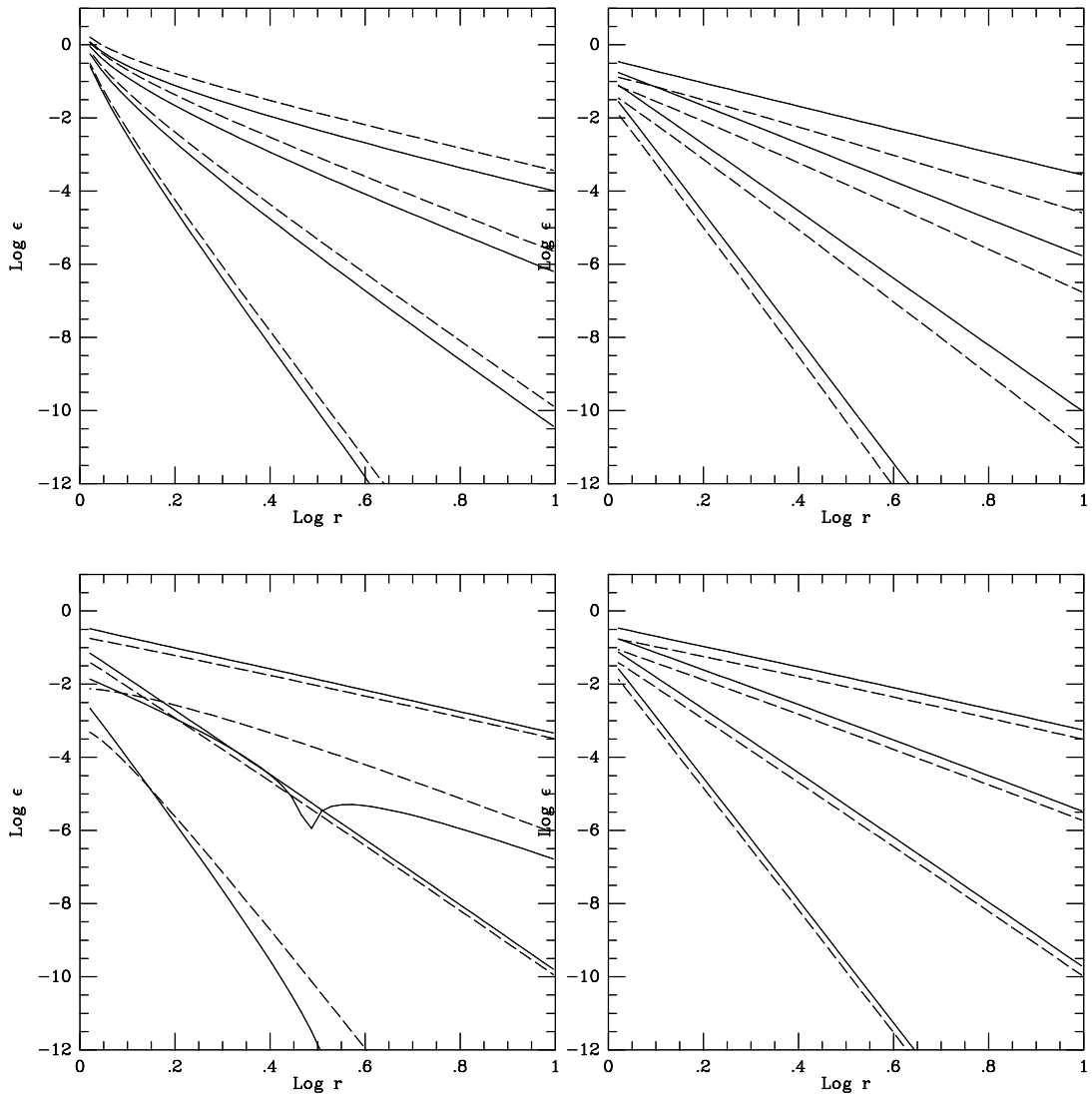
ここで γ は粒子の位置と疑似粒子の位置の間の角度。また、球面調和関数の加法定理

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

を使った。

数値例

2 次元で、粒子 1 個の作るポテンシャルの誤差を評価してみた。



粒子の位置： $(1,0)$, 疑似粒子をおく円の半径： 0.75

ポテンシャルを評価する位置の方向： 0 (左上), $2\pi/3$ (右上), $\pi/2$ (左下), π (右下),

展開次数： $2, 4, 8, 16$

実線：疑似粒子, 破線：調和関数展開

うまくいっているような気がする

誤差の入り方について

今回の方法では、疑似粒子を作るところまでは球面調和関数展開と数学的に等価

→ 普通の FMM にはない誤差が入ってくるのは、疑似粒子の作るポテンシャルを評価するところ：

もともとの球面調和関数展開にはなかった高次項が入ってくる

この高次項は、もともとの粒子分布が持つ (打ち切られた) 高次項と同程度

→ 大きく誤差が増えることはない。

アンダーソンの方法：球の半径をうまくとらないと精度が出ない：

最初に球面上でポテンシャルを評価するところで、高次項が入ってくるため。

ツリーコードへの応用

今回の方法は、多重極展開の表現法が違うだけ

↓

高次精度のツリーコードを作るのにもそのまま使える。

実用的には（GRAPE を使うなら特に）こちらのほうが有望かもしれない。
実装も、重心のかわりに複数粒子で表すだけなのでそんなに大変ではない。

まとめ

- 高速多重極展開法にもツリーコードにも使える、多重極展開を粒子分布で表現する方法を 2 次元と 3 次元の両方で定式化した。
- 2 次元の場合には、簡単な数値例で定式化が正しいことと精度が従来の方法にくらべてそれほど悪くないことを確認した
- この方法では、計算量のもっとも大きいところを GRAPE に任せることができる。高精度計算になるほど GRAPE の仕事が増えるが、これは計算時間にさほど影響しないと予想できる。

すなわち、高精度の FMM やツリーコードを GRAPE を使って実現することが可能になった。