The Pseudoparticle Multipole Method

Junichiro Makino
Department of Astronomy, School of Science
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan
and Atsushi Kawai
College of Arts and Science
The University of Tokyo
3-8-1 Komaba, Meguro-ku, Tokyo 153, Japan

概 要

Barnes-Hut (BH) treecode and Fast multipole method (FMM) are algorithms which can calculate the gravitational interaction between N particles in $O(N \log N)$ or $O(N)$ time. Both methods were proposed in 1980s and the treecode has become commonly used in cosomlogical simulations and simulations of galactic dynamics, where required accracy was low. For simulations which require high accuracy, however, the treecode has not been used because the implementation of high-order treecode is complex and calculation cost is high.

In this paper, we describe a new approach to implement high-order treecode and FMM, which we call Pseudoparticle Multipole Method ($\mathrm{P}^2\mathrm{M}^2$). In this method, we represent the multipole expansion back again by particles. As a result, the translation formulae for expansion coefficients are greatly simplified. Moreover, with this new method we can use GRAPE to accelerate the evaluation of the multipole expansion. Compated to the program on a workstation, the $\mathrm{P}^2\mathrm{M}^2$ treecode with GRAPE can achieve up to 100 times speedup.

1　はじめに

ツリー法 [1, 2] は N 個の質点間の重力相互作用を $O(N \log N)$ の計算コストで計算する方法であり、宇宙論的 N 体 (あるいはSPH) 計算や、銀河の形成、相互作用のシミュレーションでは広く使われている。計算法の原理については例えば [8] やそのなかであげた文献を参考にしていただくとして、基本的な発想は、遠くの粒子を適当にまとめて、それらからの重力を多重極展開におきかえて評価するというものである。遠くにいくほどまとめ方を大きくしていけるので、粒子一つへの重力を計算する計算量は $O(\log N)$ となり、全粒子への重力を計算するのに $N \log N$ のオーダーの計算量で済む。直接計算では計算量は N^2 に比例するので、粒子数が十分大きいところでツリー法 はまともにやるのに比べてはほとんど N に比例して速いということになる。

もちろん、$N \log N$ の前につく係数は決して小さくはないが、それほど高い計算精度を要求しなれば、 N が数百程度で直接計算よりも速くなる。[1]

さて、ツリー法は提案から 10 年以上過ぎ、銀河形成や銀河・銀河相互作用、あるいは宇宙論などの N 体計算や SPH を使った自己重力流体計算に広く使われるようになっている。これらの計算では、無衝突系を扱うという性格上、計算精度をあげるためには粒子数を増やして particle

---

[1] 昨年までの 天体力学研究会」では、 N というのは大きくても 10 程度であると想像され、このような N が大きいところの話というのはそぐわなかった感もあるが、 天体・N 体力学研究会」ではこのような話題も場違いではないものと信じる。

noise (あるいは2体緩和) を押えることが、粒子間相互作用の精度を上げることよりも重要であり、そのためにツリー法のほうが直接計算よりもよい結果が得られるためである。

これに対して、散開星団、球状星団の熱的進化や、あるいは惑星形成のような、2体緩和によって系が進化する過程を扱う場合には、ツリー法はほとんど使われていうない。これには以下のような事情があった。まず、長時間シミュレーションが必要になるために、扱える粒子数が減る。また、計算精度も高くする必要がある。そうすると、ツリー法の計算量と直接計算の計算量の差が小さくなるか、場合によっては逆転してしまう。このために、また、GRAPE[11, 9] によって直接計算がかなり高速化されたということもあり、これらの系には基本的には直接計算が用いられてきた。

必ずしも一般的な事情ではないが、GRAPE を作ってきた我々の観点からはまた以下のような問題があった。GRAPE を使ってツリー法を加速することは可能ではある [7] が、GRAPE で計算できるのは粒子（質点）の重力場だけである。従って、遠くの粒子をまとめる時も、それらを重心で置き換えることしかできず、近似の次数でいえばダイポールよりも上げることは出来なかった。このために、計算精度をあげようとすると非常に急速に計算時間が増えてしまい、ツリー法を使うメリットがなくなってしまっていた。

我々は、高精度のツリー法や FMM（Fast Multipole Method, 高速多重極展開法）のような高速計算法で精度をあげるときに必要になる高次の多重極展開を、そのままの形ではなく粒子に戻って表現する方法（疑似粒子多重極法、Pseudoparticle Multipole Method）を開発した。これにより、多重極展開の評価も GRAPE を使って行なえるようになり、高精度の計算が飛躍的に加速された。さらに、実は粒子の形で表現することで、従来よりも実装が容易になり、任意次数のツリー法が簡単に実現出来るようになった。

以下、まずツリー法についてごく簡単に説明したあと、疑似粒子多重極法の基本的な思想について述べ、それから数学的な定式化を与える。次に、計算精度と計算量、計算時間についての数値実験の結果をまとめる。

2　ツリー法の原理

計算したいものは要するに以下のような常微分方程式系の右辺である。

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

ここで N は粒子の数、$\boldsymbol{x}_i, m_i$ はそれぞれ粒子 i の位置、質量であり、G は重力定数である。

右辺を評価するもっとも単純な方法は、右辺をそのまま力任せに計算することである。粒子数が数百程度であれば、この方法（単に直接計算という）よりも速くするのは難しい。しかし、直接計算では $O(N^2)$ で計算量が増えていくので、粒子数がある程度大きくなると直接計算は現実的ではなくなる。計算量を減らすには、重力を及ぼす遠くの粒子を適当にまとめることが考えられる。まとめておいて多重極展開を作っておけば、精度は必要に応じてあげられるわけである。これがツリー法の基本的な考えである。

粒子がある空間を分割することで多重極展開や球面調和関数展開を使えるようにする。もっとも簡単に、空間を規則的な格子で分割した場合を考えてみると、粒子は、自分と、自分に隣接するセルに入っている粒子からの力は直接計算し、それ以外のセルからの力は多重極展開で計算することになる。粒子数を N、格子のセルの数を n とすれば、計算量は $N^2/n + Nn$ の程度となり、最適な n をとれば $O(N^{3/2})$ となる。

このやり方で無駄なところは、自分から見て非常に遠くにある粒子も近くにあるものも全部同じ大きさのセルにわけていることである。計算精度のことを考えるなら、遠くにいくほどセルが

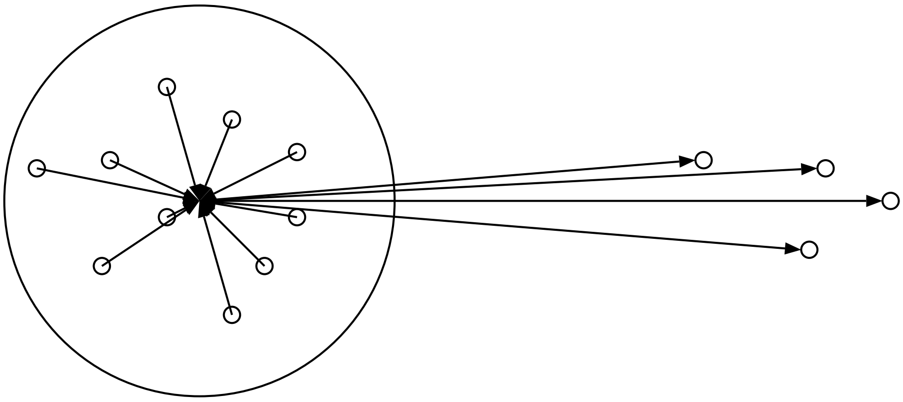

図 1: ツリー法の基本的アイディア。

大きくなってよい。これを実現するために、正方形（3次元なら立方体）を再帰的に分割していくことでツリー構造を作る。ある粒子への力を計算するには、トップレベルからツリーを降りていって、大きさに比べて十分遠い、つまり誤差が小さくなったとみなせるところで止めればよい。十分遠いかどうかの判定には、通常は見込み角と呼ばれる量

$$\theta = l/r \tag{2}$$

を使う。ここで r はセルの中心から力を評価する粒子までの距離、l はセルの一辺の長さである。

ここで、計算量と計算精度の間の関係が原理的にはどのようなものになるかを簡単に述べる。細かく考えるといろいろややこしいので、とりあえず一つのセルからの力の誤差について考えよう。展開次数を p とすれば、あるセルからの力の相対誤差は

$$e_{cell} \sim C_{p+1}(r/l)^{-(p+1)} = C_{p+1}\theta^{-(p+1)} \tag{3}$$

の程度になるはずである。ここで C_{p+1} は l^{p+1} で規格化した $p+1$ 次の多重極展開の係数の値から決まる定数である。規格化したので C_{p+1} には上限があり、上の式からツリー法の誤差限界を与えることができる。

さて、計算量の方はどうなるかといえば、極めて大雑把には

$$C \propto \theta^{-3}p^2 \tag{4}$$

で与えられるはずである。ここで、θ^{-3} に比例するのは、θ に比例してセルが小さくなるのと、3次元なのでセルの大きさの 3 乗に反比例してセルの数が増えるからである。また、多重極展開の係数の数が p^2 なので、1 つのセルからの力の計算量は p^2 に比例する。

ここで C_p が p によらないとか p が大きいとか仮定すれば、計算精度を決めた時に計算量がもっとも小さくなる θ の値は、計算精度に無関係に $e^{-2/3} \sim 0.5$ となる。つまり、精度をあげるためには次数 p をあげていくのが有効であり、見込み角を変えることで精度を調整しようとするのは計算量から見ると損になるのである。

しかし、現在広く使われているツリー法の実装は、$p=2$ の固定のものがほとんどである。また、GRAPE 上でのツリー法の実装は、GRAPE が計算するのが質点からの重力であるために原理的に $p=1$ しか扱えない。このために、精度をあげるのは困難であったわけである。

3 $\mathrm{P}^2\mathrm{M}^2$法

高次の項を扱うのに GRAPE を使う方法はないかと考えると、「多重極展開を粒子で表現する」ということができればよい。もともと多重極展開は実際の粒子分布の作る場を近似しているわけだから、適当に粒子（以下、こちらを本当の粒子と区別するために疑似粒子と呼ぶ）をおくことで多重極展開を表現しなおすということは原理的には可能なはずである。この時に、元の粒子よりも少ない数での疑似粒子で、ある次数までは元の粒子と同じ多重極展開を持たせることが出来れば、あとはそれらの疑似粒子からの重力を普通の粒子と同じように計算することでツリー法が実現できることになる。

重力の場合、いくつかの粒子の重心は $p = 1$ まで展開が合うという意味でそのような粒子のおき方の例になっている。

次に 4 重極項を考えてみる。重心回りの 4 重極項モーメントテンソルはトレースレスなので、対角化すれば独立な要素は 2 個である。言い換えれば、疑似粒子は第一軸と第二軸の張る平面上におけばよく、3 個の等質量の粒子を 2 等辺 3 角形におくことでで 4 重極項は表現出来る。

もっと高次の場合でも、このように粒子配置を構成するのも原理的には可能なはずだが、しかし、次数をあげていくと解くべき方程式がどんどんややこしくなるし、だんだんどうやって式を立てればいいかも良くわからなくなる（わかる人は是非やってみて欲しい）。従って、以下ではもうちょっと安直なアプローチを考える。

結局やりたいことはなんであったかというと、多重極展開の係数が実粒子の分布と等しくなるような疑似粒子の分布を見つけだすということである。ここで、疑似粒子をおく位置を球面上に限ってしまうと、疑似粒子の方の分布の多重極展開は、もちろん疑似粒子の質量分布の球面調和関数展開そのものであるから、別にこういう制限をつけても疑似粒子のおき方を決めることは依然として可能である。もちろん、位置の自由度のうちの 1 つを殺しているので、その分たくさん粒子がいることになるが、とりあえず大した問題ではないと信じよう。この時には、結局球面調和関数展開の逆変換で球面上の質量分布が求まり、それを適当に離散的な質点で近似することで疑似粒子を決めるということになる。

さて、依然として問題であるのは、どうやって球面上に粒子をおくかである。これは、上で求まった球面上の質量分布の、なんらかの意味での最良近似であるようなものを有限個の粒子で表現するという問題になる。ここで注意して欲しいのは、粒子一つの多重極展開は位置に対しては非線形な関数であるが、質量に対しては線形な関数であることである。従って、位置を固定して質量だけを変えるようにして、p^2 個の粒子を適当において質量だけを変えてやることで、必ず p^2 個の展開係数の値を合わせることが出来るはずである。

が、落ち着いて考えてみるとこれはまだいろいろ面倒そうである。計算の順番が、まず展開係数を求めてから、その展開係数を満たすような質量分布を決めるということになる。もちろん後半の計算では解くべき線形方程式の係数はいつも同じなので、LU 分解しておくか逆行列を求めておいて、それから計算すれば一応はいいことになる。ここでは、点の数が増えることはあまり気にしないで定式化やプログラムが簡単になる方向で考える。

形式的には、するべきことは実粒子分布を多重極展開し、その係数を逆球面調和関数展開して仮想粒子の質量に戻すということである。仮想粒子の数が無限に多い連続極限では、形式的には逆変換の式は極めて簡単で

$$\rho = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \alpha_l^m {Y^*}_l^{-m} \tag{5}$$

と書ける。ここで α_l^m は球面調和関数展開の係数であり、関数 $Y_l^m(\theta, \phi)$ は球面調和関数である。簡単になるのは球面上で球面調和関数が完全正規直交系を作っているからである。有限個の仮想

粒子ですます時も、もしもその上で球面調和関数を有限次数で切ったものが直交系をなしていれば、上の式と本質的に同じ簡単な変換式がつかえることになる。

p 次までの球面調和関数が有限個の仮想粒子の質量が作る空間の上で直交系をなすためには、その空間のなかで内積演算が定義されてそれが直交性を満たしていればよい。そのためには、結局 $2p$ 次の多項式の積分が厳密に求められればよい。そのように仮想粒子を置く方法はいろいろありえて、例えば θ 方向は p 次ルジャンドル多項式の零点に置き、 ϕ 方向は $2p+1$ 個を等間隔に置くというものでも実現できる。が、この場合、仮想粒子の重みづけが θ の値によって変わる。

もうすこしましな置き方はないかと思うわけだが、これは実は非常に昔から計算幾何学のほうで研究があり、 spherical designs という名前で知られている。これはまさに我々が必要としている、

- ある次数までの多項式の球面上の積分を厳密に与える。
- 数値積分の重みがすべての点で等しい

という性質を満たすものである。つまり、 t 次の spherical design というのは数学的には以下のように定義される。

単位球面 S 上の座標点 $r_i (1 \le i \le K)$ の集合で、t 次以下の任意の多項式 $g(\boldsymbol{r})$ に対して

$$\int_S g(\boldsymbol{r}) ds = \frac{4\pi}{K} \sum_{i=1}^{K} g(\boldsymbol{r}) \tag{6}$$

が成り立つようなもの。

これには古典的な研究 [10] と、最近になって詳細に評価しなおしたもの [6] があって、若干結果が違うが 20 次程度のものまで求められている。また、座標値そのものは WWW 上[2]に公開されているので、ファイルに落すのは非常に簡単である。次数 p が小さい時には、正多面体の頂点といったものになっているが、それ以上では基本的には切頭立方体の組合せといった方法で構成されている。点の数は結局ルジャンドル多項式を使う場合とあまり変わらないが、数値積分の重みがすべての点で等しいためにプログラムが簡単で数値的にも性質がよい。

さて、 spherical design を使うことに決めると、結局実粒子から仮想粒子への変換は以下のように計算できることになる。

まず、球面調和関数展開の係数 α_l^m は、

$$\alpha_l^m = \sum_{i=1}^{N} m_i r_i^l Y_l^{-m}(\theta_i, \phi_i) \tag{7}$$

で与えられる。ここで m_i は粒子 i の質量であり、 (r_i, θ_i, ϕ_i) はその位置の極座標表示である。関数 $Y_l^m(\theta, \phi)$ は球面調和関数であり、

$$Y_l^m = (-1)^m \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) e^{im\phi} \tag{8}$$

で与えられる。ここで P_l^m はルジャンドル陪関数である。これと同じ展開係数をもつ半径 a の球面上での質量分布$\rho(\theta, \phi)$ は

$$\alpha_l^m = \int_S \rho(\theta, \phi) Y_l^{-m}(\theta, \phi) dS, \tag{9}$$

を満たす必要がある。球面調和関数の直交性から、この質量分布ρは

$$\rho = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \alpha_l^m Y^{*}{}_l^{-m} \tag{10}$$

---

[2] `http://www.research.att.com/~njas/sphdesigns/index.html`

で与えられることになる。ここで積分を先ほどの K 個の仮想粒子上での数値積分で置き換えれば、仮想粒子 j の質量が

$$m_j = \frac{1}{4\pi K}\sum_{l=0}^{p}\sum_{m=-l}^{l} \alpha_l^m {Y^*}_l^{-m} \tag{11}$$

であたえられることになる。ここで、展開が p 次で打ち切られていることに注意してほしい。

さて、これで必要な変換は与えられたが、いちいち係数を求めてからそれをもう一回変換するのは美しくない。実粒子の質量から直接仮想粒子の質量を求めることを考えてみると、これは i, l, m の 3 重の総和になって一見計算が大変なような気がする。しかし、実は三角関数と同様に球面調和関数にも加法定理

$$P_l(\cos\gamma) = \frac{4\pi}{2l+1}\sum_{m=-l}^{l} Y_l^m(\theta,\phi)Y_l^{-m}(\theta',\phi') \tag{12}$$

というものがあり [ここで γ は方向 (θ,ϕ) と (θ',ϕ') の間の角度]、これを使えば m についての和は消えて、結局

$$m_j = \frac{2l+1}{K}\sum_{i=1}^{N} m_i \sum_{l=0}^{p}(r_i/r)^l P_l(\cos\gamma) \tag{13}$$

ということになる。

ここで注意してほしいことは、式 (13) は実粒子から仮想粒子への変換だけでなく、ツリーの上のレベルに上がっていく仮想粒子から仮想粒子への変換にも同様に使えることである。つまり、式 (13) で必要な変換はすべて表現されていることである。もうひとつ重要なことは、l の総和の式は結局 r_i/r と $\cos\gamma$ の l 次多項式なので、まともに計算すれば $O(l^2)$、漸化式を使えば $O(l)$, 区分近似多項式を準備しておけば $O(1)$ で計算できるということである。

なお、ツリー法の場合にはこれで話は終りであり、あとはセルから各粒子への力を計算するときに疑似粒子からの力の和として計算するというだけである。従って、従来のツリー法にわずかな手直しをするだけで任意次数のものが作れる。

FMM[4, 5] についても同様なことが考えられるが、ここでは省略する。その理由は、現在までに発表されたほとんどの論文で、計算精度が同程度ならばツリー法のほうが FMM よりも有意に速いという結果が示されているためである [3]。

ツリーは計算量が $O(N\log N)$ で FMM は $O(N)$ なのだから、FMM のほうが速そうなものであるが、実際にはそうはならない。ここではこの理由に深入りはしないが、簡単にいうとツリーでは多重極を評価するところだけで誤差が入るのに対し、FMM では多重極変換を局所変換に変換して、さらにそれ移動してというふうに段階が増え、それぞれの段階で誤差が入ってくるためである。

4 テスト計算

上の定式化に基づいた $\mathrm{P}^2\mathrm{M}^2$ を FORTRAN で書かれたツリー法に組み込んで、計算時間、計算精度の評価を行なってみた。まず、汎用スカラー計算機での結果を示し、次に GRAPE-4 での結果を示す。

4.1 汎用計算機での結果

図 2 に見込み角 θ と計算誤差の関係を示す。ここでは計算誤差は加速度の相対誤差の 2 乗平均として定義される。これは時間積分の精度には直接つながらないという意味であまりよい量では

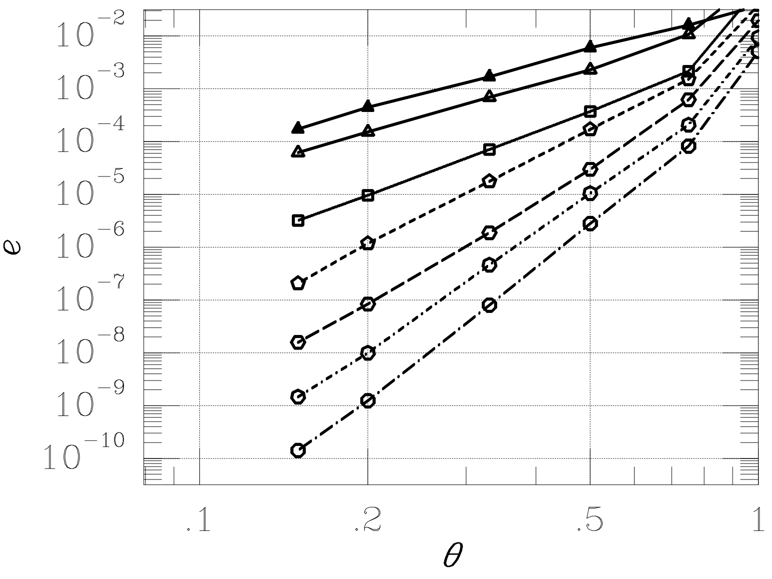

図 2: P^2M^2法での見込み角と計算精度の関係。上から従来のツリー法、$p = 1, \cdots 6$。

ないが、他の多くの論文でこれがもちいられているのでここではそれに従う。粒子分布は球のなかの一様分布で、粒子数は 262,144 である。

図 2 からわかるように、次数が高いほど計算精度は上がる。計算精度の見込み角への依存性は大雑把にいって $e \sim \theta^{p+1.5}$ の程度になる。計算量と見込み角の関係を示したものが図 3 で、いうまでもなく θが小さいほど、また次数が高いほど計算量は増える。なお、ここで計算量として書いているのは計算時間ではなく、ある粒子が力を受けている粒子（実粒子と疑似粒子をあわせて）の個数の平均値である。

さて、先に述べたように、必要な計算精度を決めれば最適な（計算量を最小にする）次数と見込み角の組合せが存在しているはずである。これは、計算量と計算精度の関係を見ればわかるので、それを図にしたものが図 4 である。要求する計算精度を上げていくと、最適次数は上がっていくことがわかる。逆に、見込み角はほぼ一定のままであることに注意して欲しい。

なお、$p = 1$ の場合はダイポールまで表現されていることになり、これは従来のツリー法では重心だけを使っている場合に相当する。計算量は疑似粒子を 4 個必要とするために 4 倍となっていて、この場合や、その次の $p = 2$ の場合（疑似粒子 12 個が必要）については、汎用計算機で計算するかぎり従来のツリー法に比べていいことはない。

4.2 GRAPE-4 での実装

GRAPE を使う場合の計算コードも、従来のものを少し手直しするだけで構成できる。GRAPE を使う場合の基本的な考えは、ホストでツリーをたどっていくのをあるグループに対して行ない、それで出来た相互作用リストとグループの中の粒子のリストを GRAPE に送って力を計算してもらおうというものであった [7]。このために、一粒子あたりの計算量はグループの中の粒子の数の数倍程度増えている。GRAPE が相対的に高速なので、この部分はあまり問題にならないわけである。疑似粒子を使った場合でも、この GRAPE を使うために計算量が増えている分は変わらないので、結局次数を上げてもそれほど計算量が増えないということが期待できる。

計算精度、計算時間等の測定結果を図 5 に示す。粒子分布は前と同じであり、汎用計算機は Alpha 21164 300 MHz (DEC Alphaserver 8400), GRAPE は GRAPE-4 ボード 1 枚（32 chip, 10 Gflops）

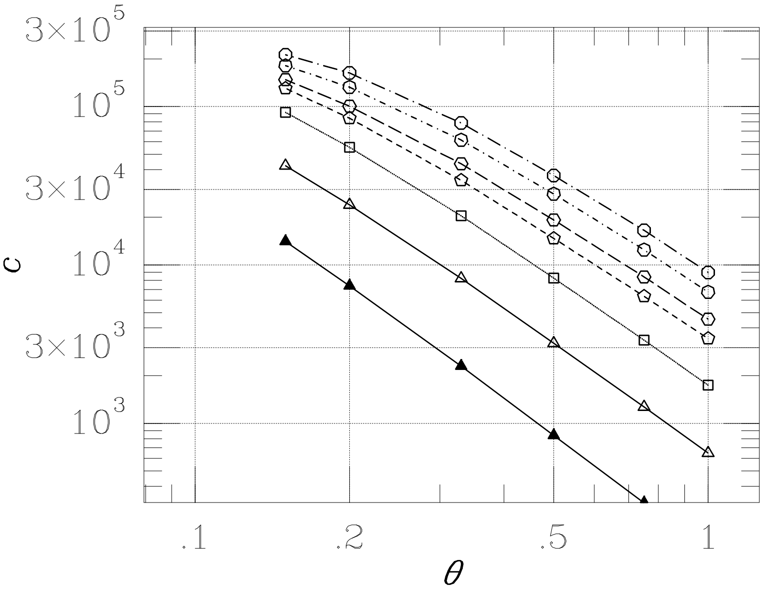

図 3: P^2M^2法での見込み角と計算量の関係。線、シンボルの意味は図 2 と同じ。

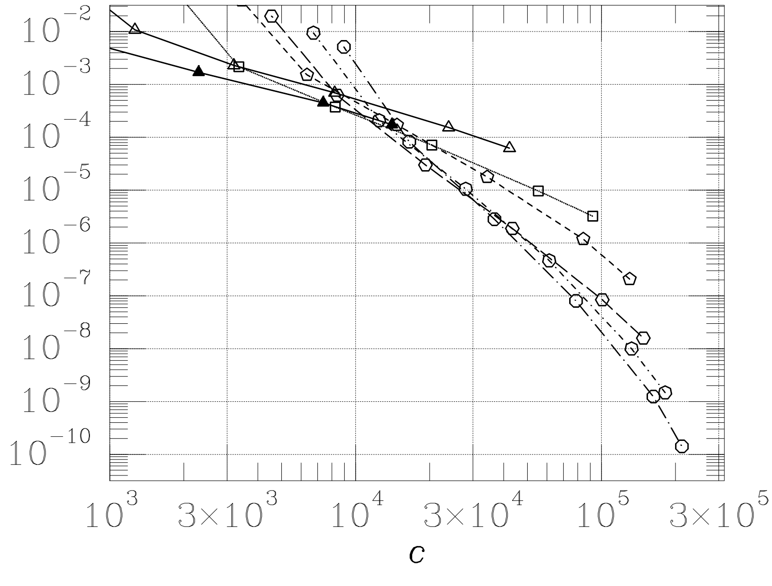

図 4: P^2M^2法での計算量と計算精度の関係。線、シンボルの意味は図 2 と同じ。

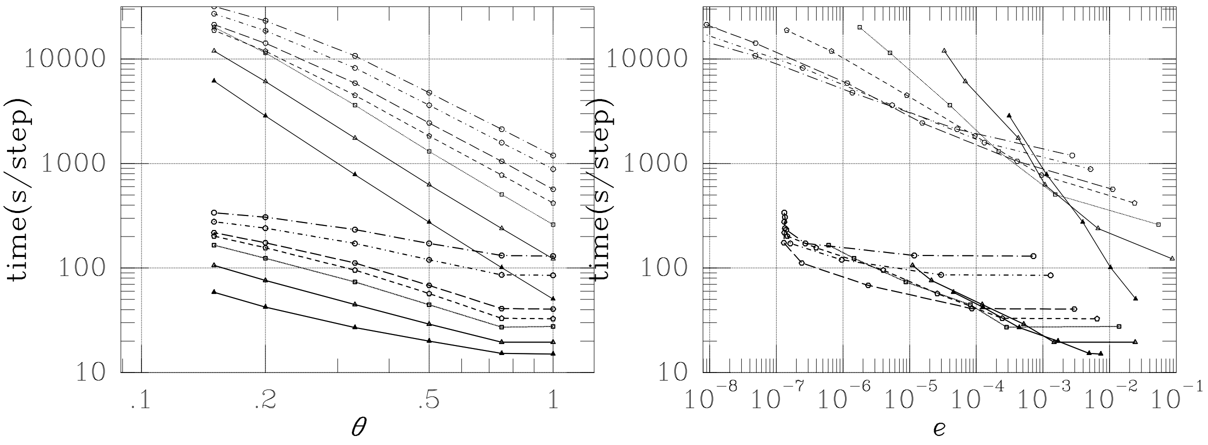

図 5: 汎用計算機と GRAPE-4 での（左）見込み角と計算時間、（右）計算時間と計算精度の関係。上は汎用計算機、下は GRAPE-4. シンボルの意味は図 2 と同じ。

である。上で期待したとおり、次数を上げても、また見込み角を小さくしても汎用計算機の場合のようには計算量が増えない。結果として、精度を上げるほど GRAPE を使うメリットは大きくなり、10^{-6} 程度の精度を要求するなら計算速度の差は 100 倍近くになる。GRAPE を使う場合には、もともと計算量が増えているということもあって疑似粒子法の計算量が多いという欠点が目立たなくなっていることがわかる。

5　まとめと議論

ここでは、高精度のツリー法を実現するために必要なポテンシャルの多重極展開を、疑似粒子をつかって表現する疑似粒子多重極法（$\mathrm{P}^2\mathrm{M}^2$ 法）について、その原理、定式化、数値実験の結果を概説した。この方法は、普通の計算機ではプログラムが簡単になるという利点はあるが計算量が増えるためかならずしも実用的ではないかもしれない。しかし、粒子間の重力を高速に計算する専用計算機 GRAPE を使って多重極展開が計算できるため、劇的な高速化が可能であるというのが大きな特長である。

参考文献

[1] A. W. Appel: "An efficient program for many-body simulation," *SIAM Journal on Scientific and Statistical Computing* 6, 85–103 (1985).

[2] J. Barnes and P. Hut: "A hiearchical $O(N \log N)$ force calculation algorithm," *Nature* 324, 446–449 (1986).

[3] D. Blackston and T. Suel: " Highly portable and efficient implementations of parallel adaptive n-body methods," *Proceedings of SC97.* ACM 1997, (CD–ROM).

[4] L. Greengard and V. A. Rokhlin: "A fast algorithm for particle simulations," *Journal of Computational Physics* 73, 325–348 (1987).

[5] L. Greengard and V. A. Rokhlin: "Rapid evaluation of potential fields in three dimensions," *Vortex Methods* (edited by Anderson C. and Greengard, C.), number 1360 in Lecture Notes in Mathematics, Springer-Verlag, Berlin, 121–141 (1988).

[6] R. H. Hardin and N. J. Sloane: “McLaren's improved snub cube and other new spherical designs in three dimensions,” *Discrete and Computational Geometry,* 15, 429 (1996).

[7] J. Makino: “Treecode with a special-purpose processor,” *Publ. Astron. Soc. J.* 43, 621–638 (1991).

[8] 牧野淳一郎: “高速多重極展開法とツリー法—多体シミュレーションのための高速算法,” 応用数理 8, 277-286 (1998).

[9] J. Makino and M. Taiji, *Special Purpose Computers for Scientific Simulations – The GRAPE systems*, John Wiley and Sons, Chichester (1998).

[10] A. D. McLaren: “Optimal numerical integration on a sphere,” *Math. Comput.,* 17, 361–383 (1963).

[11] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, and M. Umemura: “A special-purpose computer for gravitational many-body problems,” *Nature* 345, 33-45 (1990).