重力多体系の物理と数値計算 — 大自由度系における平衡と緩和

牧野淳一郎
東京大学理学系研究科天文学専攻

2000 年 8 月 21 日

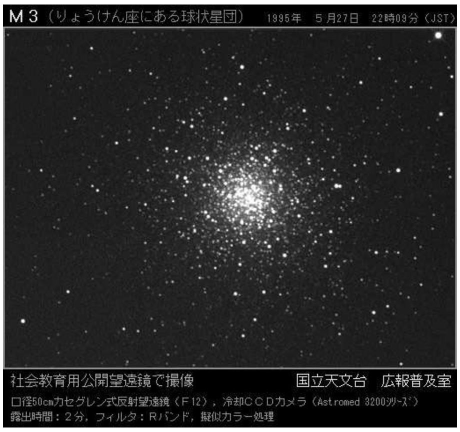
今日の予定

1. 初めに :重力多体系とはどんなものか
2. 理想化された系の統計力学
   (a) 「熱平衡状態がない」ということについて
   (b) それでも、熱平衡状態について
   (c) 支配方程式
   (d) 流体近似
   (e) 重力熱力学的不安定性
   (f) 有限振幅での進化
   (g) 重力熱力学的振動
3. 応用例
   (a) 球状星団 ·散開星団
   (b) 宇宙の大規模構造 ·銀河団
   (c) 惑星形成
4. N 体シミュレーションの技法
   (a) 時間積分法
   (b) 独立時間刻み
   (c) ツリー法 ·FMM
   (d) 計算機の高速化

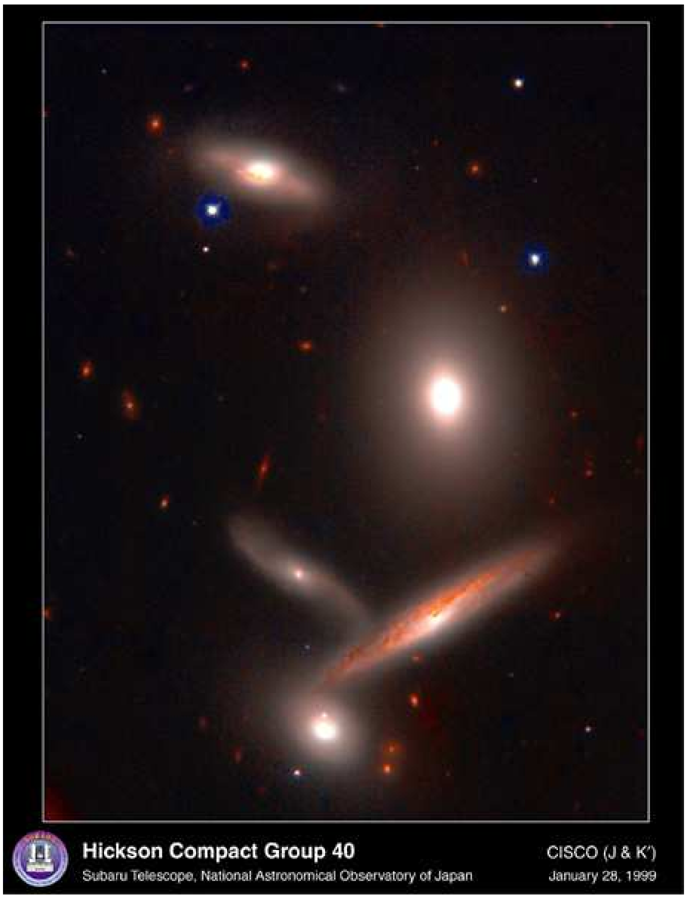
重力多体系とは ？

銀河

球状星団



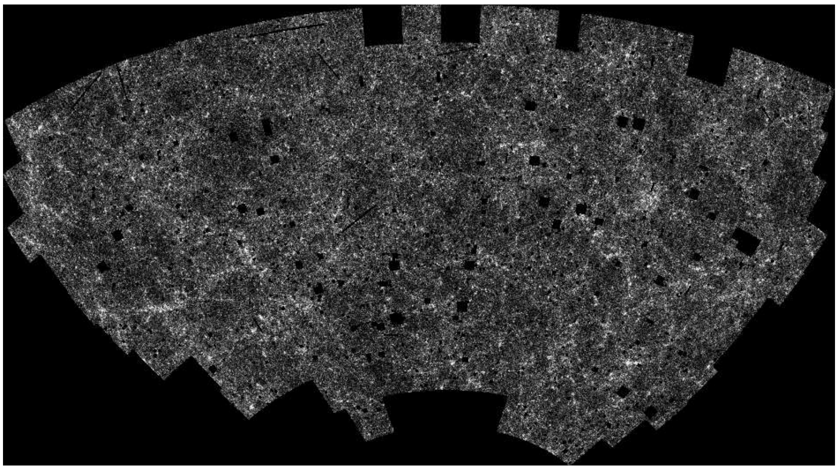
銀河群


Hickson Compact Group 40
Subaru Telescope, National Astronomical Observatory of Japan
CISCO (J & K')
January 28, 1999

銀河団

http://antwrp.gsfc.nasa.gov/apod/ap950917.html

大規模構造



http://www-astro.physics.ox.ac.uk/~wjs/apm_grey.gif

重力多体系とは（2）

- 多数の粒子がお互いの重力で引き合って、全体として作るポテンシャルの中で運動しているような系
- 星団・銀河・銀河群・銀河団・超銀河団、、、
- 構成要素：恒星（星間ガス、ダークマター、、、）
- 銀河群、銀河団等：銀河が構成要素ともいえる

サイズ、質量、構造：極めて多様

基本的な問題：

- なぜ、このような多様なスケールが「多数の粒子が重力で集まっている」というだけのことから生じるのか？
- 本当にそれだけのことから生じるのか？

理想化された系の統計力学

多数の星が集まってできている
↓
（熱平衡なら）古典統計が使える??

細かいことをいえば、、、

- 構成要素の星は一つ一つ個性があり、質量も違う

といったこともあるが、とりあえずそういうことは忘れる。

つまり、

重力多体系
＝　「同じ質量 m の星（粒子）が N 個集まったもの」

と考える。

これなら統計力学が使えそうな気がする。

熱平衡状態の非存在

重力多体系に統計力学 / 熱力学を適用する際の基本的な問題 ：

「重力多体系には熱平衡状態がない」ということ

「熱平衡状態がない」というのはそもそもどういうことか ？

示し方はいろいろ、、、以下は 1 例

古典統計である

したがって、速度分布関数はマックスウェル・ボルツマンでなければならない

これは不可能

自己重力系にある粒子のエネルギーは、定義からして無限遠での重力ポテンシャルの値を超えない

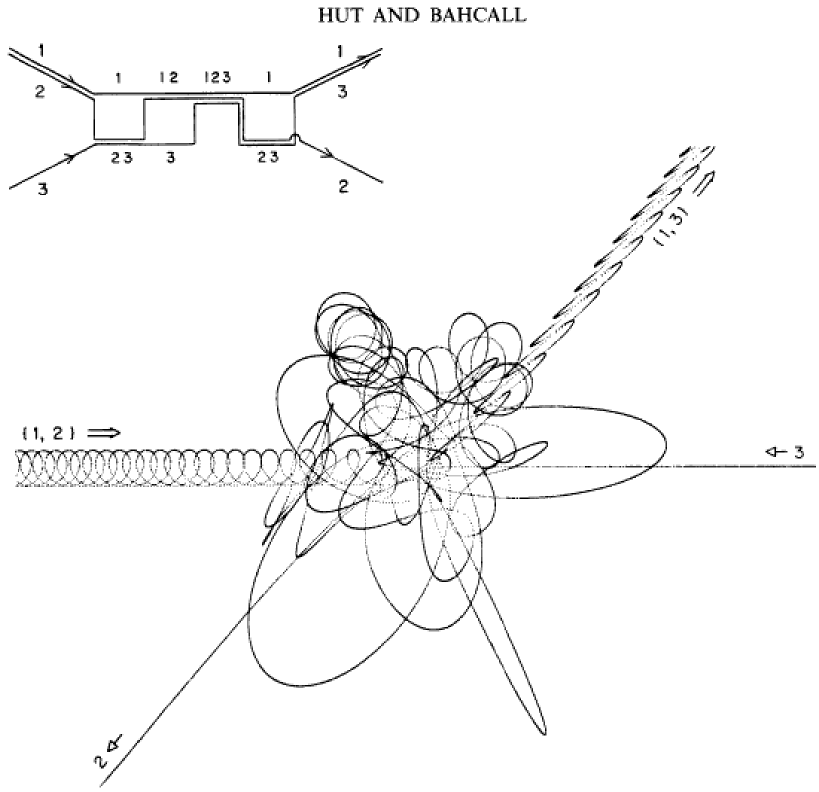
したがって、ある空間のある点での速度分布には必ず上限がある。

言い換えると、上限を超えた粒子は逃げていってしまう、、、

例 — 重力3体問題

2 体問題では周期解になるだけで何も起きないので、3 体問題。

重力 3 体問題の解の一例



重力3体問題(2)

全系のエネルギーが負で、全体としては自己重力で集まっている適当な初期条件　(前の例では2つは連星というちょっと特別な例)から運動させる

しばらくは3つがそれぞれお互いの回りを運動する

そのうちに、粒子のうちの2つが強く結合した状態になり3つめがその反作用ではね飛ばされるという現象が起きる。

3体問題としては、これが　最終状態」

粒子数が多い系でも、本質的には同じことが起きる。

つまり、粒子同士の散乱の結果、高エネルギーの粒子が作られるとそれは系から逃げていってしまうのである。

というわけで、統計力学からいえること：

重力多体系は、十分長い時間が立てば蒸発してしまう

しかし、これは間違いではないけど、でも、現実に存在している重力多体系の理解には役に立たない。

それでも、熱平衡状態について

まあ、それでも、やっぱり、熱平衡というものを考えたい

- いきなり非平衡とかいわれてもわけがわからない
- ある種の重力多体系では、熱平衡からそんなに遠くない状態が実現されている

というわけで、熱平衡状態がどういうものかを理解しておくことは極めて重要。

とはいえ、厳密には熱平衡はない。

問題：どうやって、「本質」を失わないで熱平衡がある系に持っていけるか？

- 空間の次元を落す。
  空間 1 次元にすると、熱平衡がある。（粒子間ポテンシャルが無限大で発散するので）
- 空間は 3 次元のままで、エネルギー無限大でも逃げないようにする（壁をつける）

どちらもいろいろな人がいろいろなことをしている。

熱平衡状態について(2)

1次元系の話はいろいろ面白いが、今日は省略。興味がある人は、

Tsuchiya, Konishi and Gouda, PR, E50, 1996, 2210

とか

Makino, J., astro-ph/9903238

を見ること。

今日1次元系の話をしないのは、結局1次元系では3次元系で起きるいろいろな重要な物理が失われるから。

壁をつけるのは、「熱平衡がある」という他はあまり大きく現実と変わらないので、以下こっちの話。で、さらに話を簡単にするために壁は球対称とする。

支配方程式

ここまで、数式なしで話をしてきたが、さすがにここからはそうもいかない。

重力多体系の運動方程式：

$$\frac{d^2\mathbf{x}_i}{dt^2} = -\sum_{j\neq i} Gm_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3}, \tag{1}$$

$\mathbf{x}_i$ と m_i は粒子 i の位置と質量、G は重力定数。右辺の和は全粒子についてとる。

が、この式をじっとみていてもほとんど何もわからない（シミュレーションするなら話は別）。

統計力学でいこうということなので、ボルツマン方程式を考える。

$f(\mathbf{x}, v, t)$：位相空間での（一体）分布関数

$$\frac{\partial f}{\partial t} + v \cdot \nabla f - \nabla\Phi \cdot \frac{\partial f}{\partial v} = 0, \tag{2}$$

ここで Φ は重力ポテンシャル。以下のポアソン方程式の解。

$$\nabla^2\phi = -4\pi G\rho. \tag{3}$$

G は重力定数であり、ρ は空間での質量密度

$$\rho = m\int dvf \tag{4}$$

支配方程式(2)

通常のボルツマン方程式との違い：衝突項がない

自己重力多体系の場合、粒子数無限大の極限では重力場が滑らかになって衝突項が消える。

もちろん、熱力学的な時間発展が起きるためには衝突項が必要。この話はちょっと後回し。

力学平衡

力学平衡：ある（定常）分布関数 $f(x,v)$ が上の無衝突ボルツマン方程式の解であるとき、f は力学平衡解ということになる。

熱力学的な進化：力学平衡解が（ほぼ）力学平衡をみたしたままゆっくりと違う状態に移っていく。

ジーンズの定理

> 分布関数が力学平衡であるための必要十分条件は、分布関数が運動の積分の関数として書けることである。

力学平衡を与える分布関数は無限にたくさんあり、断熱壁をつけなければその全ては熱平衡でない。

衝突項がなければ、力学平衡な分布関数はそれ以上進化しない。

衝突項があれば、エントロピー生成があるので普通は熱平衡に向かう。

衝突項 (2体緩和)

重力多体系の特性 :$N \rightarrow \infty$ の極限で衝突項がなくなること。

逆にいうと、有限粒子数では衝突項がある (2体緩和)

衝突項の起源 :有限の数の粒子が動いているため、重力場が厳密には定常でないこと。

粒子同士がお互いの近くを通った時にお互いの重力で軌道を変えることといっても同じ。

この効果を見積もっておく :

話を簡単にするため、速度 0 で一様 (無限に広がってるかどうかはちょっと目をつぶる) に数密度 n で質量 m の粒子が分布しているところに、質量 0 のテスト粒子を飛ばす場合を考える。

テスト粒子が一つの粒子から距離 (インパクトパラメータ) b を速度 v で通った時に曲がる角度 :

$$\begin{aligned} \tan\theta &= \frac{2b}{(b/b_0)^2 - 1} \\ b_0 &= \frac{Gm}{v^2} \end{aligned} \tag{5}$$

単位時間当たり、インパクトパラメータが $(b, b+db)$ の範囲にある散乱の回数は $2\pi nvbdb$ である。

散乱の方向はランダムであると仮定すれば、時間平均をとれば 1 次のモーメントは 0 になるが、 2 次は 0 にならない。つまり、

$$< \Delta\theta^2 >= 2\pi nv \int_0^{b_{max}} \delta\theta^2 bdb \tag{6}$$

積分範囲 ?

2体緩和 （続き）

積分範囲の上限 :どうとるのかについては議論があった。

Chandrasekhar : 「平均粒子間距離よりもインパクトパラメータが大きいような散乱は、多体の干渉によって効かなくなるのでそこで積分を打ち切ってよい」

Spitzer 他 :システムの全部の粒子が効く=システムサイズまで入る。

数値実験等では、「システムサイズまで」

というわけで、結局

$$< \Delta\theta^2 > \sim Gnv^{-3}m^2 \log(R/b_0) \tag{7}$$

R :システムの大きさ

緩和時間 :

$$t_\theta \sim \frac{v^3}{Gnm^2 \log\Lambda} \tag{8}$$

Λは上の R/r_0を単に書き換えただけ。

系の質量が一定の場合 ($nm = \mathrm{const.}$): $t_\theta \propto 1/m$

粒子数が多い程緩和が遅い

本当は、バックグラウンドが動いている時に有限の質量のテスト粒子を動かしてどうなるか見ないといけないが、緩和時間の密度、質量への依存性は上の例と同じ。

2体緩和 （続き2）

典型的な自己重力系の場合 :壁がない自己重力系に対してビリアル定理を使うと、

$$U + 2T = 0 \tag{9}$$

$U \sim -GM^2/2R, T = -M < v^2 > /2$

$$t_{rh} \sim 0.1 \frac{NR^{3/2}}{M^{1/2}G^{1/2}\log\Lambda} \tag{10}$$

ここで R は系の「典型的な」半径

現実の系を考える上で注意するべきこと：

- 質量が違う粒子間では、平均としては等分配になろうとする

  → 重い粒子は速度を失って、ポテンシャルの底に沈む（**dynamical friction**）
- 球対称でない系（円盤銀河等）では元の定義に戻って考える必要がある
- 同じ系でも場所によっては全然違う

いずれにしても、2体の重力散乱によるミクロスコピックなエネルギーの再分配を通して、系の熱力学的な進化が起きる。

熱平衡分布

以下、熱平衡分布を導く。

球対称→運動の積分はエネルギー E と全角運動量 J の $\mathbf{2}$ つだけ

熱平衡→速度分布がマックスウェル分布

結局、分布関数 f がエネルギーだけに依存

$$f(\mathcal{E}) = \frac{\rho_1}{(2\pi\sigma^2)^{3/2}} e^{\mathcal{E}/\sigma^2} = \frac{\rho_1}{(2\pi\sigma^2)^{3/2}} \exp\left(\frac{\Psi - v^2/2}{\sigma^2}\right) \tag{11}$$

$\mathcal{E}$ の定義 ：

$$\Psi = -\Phi + \Phi_0, \qquad \mathcal{E} = -E + \Phi_0 = \Psi - v^2/2 \tag{12}$$

Φ_0 は定数で、普通は $\mathcal{E} > 0$ で $f > 0$, $\mathcal{E} \leq$ で $f = 0$ となるようにとる。

速度分布はそういうわけで初めからわかっているが、空間分布を知りたい。

このためには、f を速度空間で積分して密度 ρ をポテンシャルの関数として表す。

誤差関数についての

$$\frac{2}{\sqrt{\pi}} \int_0^\infty e^{-x^2} dx = 1 \tag{13}$$

を使うと、

$$\rho = \rho_1 e^{\Psi/\sigma^2} \tag{14}$$

熱平衡分布 （続き）

ポアソン方程式に前の式を入れると

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Psi}{dr}\right) = -4\pi G\rho \tag{15}$$

従って、

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\log\rho}{dr}\right) = -4\pi G\sigma^2\rho \tag{16}$$

となって密度についての常微分方程式になる。

- 特異解 — **singular isothermal sphere**

$$\rho = \frac{\sigma^2}{2\pi G r^2} \tag{17}$$

- 特異でない解 :中心密度が有限。
  中心から外側に向かって解いていけば求まる。
  $r \to \infty$ の極限で $\rho \propto r^{-2}$
  どこか有限半径で打ち切らないと質量が発散

自己重力流体との関係

熱平衡なら（そのときだけ）、分布関数は同じ。

流体の方程式系：

静水圧平衡の式は

$$\frac{dP}{dr} = -\rho \frac{GM_r}{r^2} \tag{18}$$

これと、等温の状態方程式

$$P = \frac{k_B T}{m}\rho \tag{19}$$

から P を消して、さらに M_r を微分してみれば、

$$\frac{d}{dr}\left(r^2 \frac{d\log\rho}{dr}\right) = -\frac{4\pi Gm}{k_B T}\rho r^2 \tag{20}$$

要するに、上と同じ方程式が出てくる。

まあ、分布関数が同じなのでそうならないといけないわけではある。

熱平衡分布の熱力学的安定性

以下、熱平衡分布が安定かどうかを考える。

で、とりあえず流体の場合を考える。流体だと普通に温度や圧力、エントロピーとか考えられて、熱力学的な取り扱いができるから。

何に対して安定か :熱 (エネルギー) の再分配を与えるような摂動に対して。

普通の統計力学では、平衡状態は安定に決まっている。

どこかから熱をとって、別のところに与える → とられたところは温度が下がり、もらったところは温度が上がる → とられたところには回りから熱が流れ込む、もらったところは流し出す → 等温に戻る。

重力が働くと、そうとは限らない :

ここまでで求めた等温平衡状態 :中心では重力をささえるために密度が高い。外側にいくほど密度が下がる。

↓

熱を奪われると、圧力が下がって少し収縮する

↓

重力が強くなるので、もうちょっと収縮する

↓

回りより温度が上がる

ということが起こり得る。

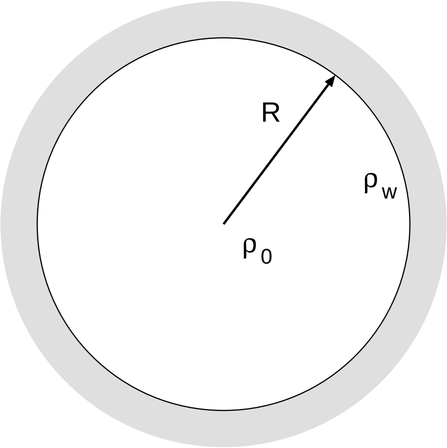
熱力学的安定性（続き）

熱をとられて温度が下がる

＝「見かけの」比熱が負である

以下、線形安定性解析の結果を見ていくことにする。

改めて、絵を書いておく。



半径 R の中に質量 M のガスがあり、中心密度 ρ_0 と壁のところでの密度 ρ_w の比が D であるとする。

$D=1$：重力なし（温度無限大）

$D=\infty$：特異解 $\rho \propto r^{-2}$

熱力学的安定性 （続き２）

熱力学的安定性の解析方法はいろいろ ：

- **Linear series analysis (Chandrasekhar, Lynden-Bell and Wood 1968) :** D を変えた時のエネルギーの変化を見る。極小値があれば、そこから先は熱力学的に不安定。
- 摂動を与えて、エントロピーの **2** 次の変分を見る **(Hachisu and Sugimoto 1978)**
- 摂動を与えて、その時間発展を見る （普通の線形解析, **Makino and Hut 1991** ）

以下、**3** 番目の方法に沿って話す。

熱伝導の式 ：

$$K\frac{\partial T}{\partial r} = -\frac{L}{4\pi r^2} \tag{21}$$

$L(r)$ ：半径 r のところでの熱流束

K ：熱伝導の係数

仮定 ：

$$K = \rho^{\alpha} \tag{22}$$

ここで、一応重力多体系に似せる ：$\alpha = 1$

エントロピーについての式 ：

$$\frac{\partial L}{\partial r} = -4\pi r^2 \rho T \frac{\partial s}{\partial t}|_M. \tag{23}$$

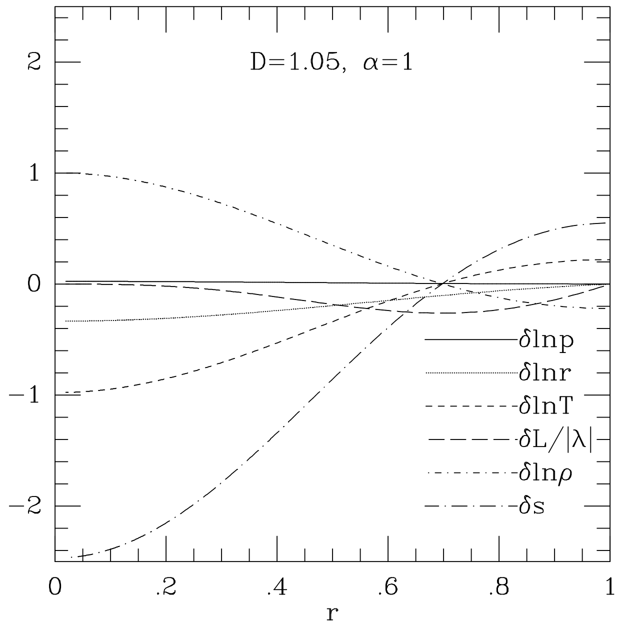
境界条件 ：

$$L = 0 \quad \text{for} \quad M = 0 \quad \text{and} \quad M = 1. \tag{24}$$

熱力学的安定性 （続き３）

以下、式の細かいところを見たい人は原論文に当たってもらうとして、結果だけ書く。

安定な場合



λ: 固有値

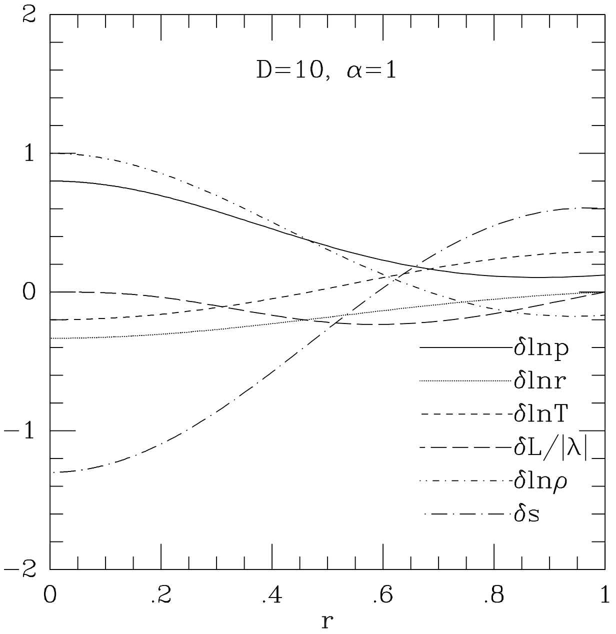
$D = 1.05$: 重力が無視できる場合。

- 圧力は変化しない
- エントロピーと温度が比例

要するに、普通の断熱容器のなかのガス。

熱力学的安定性 （続き４）

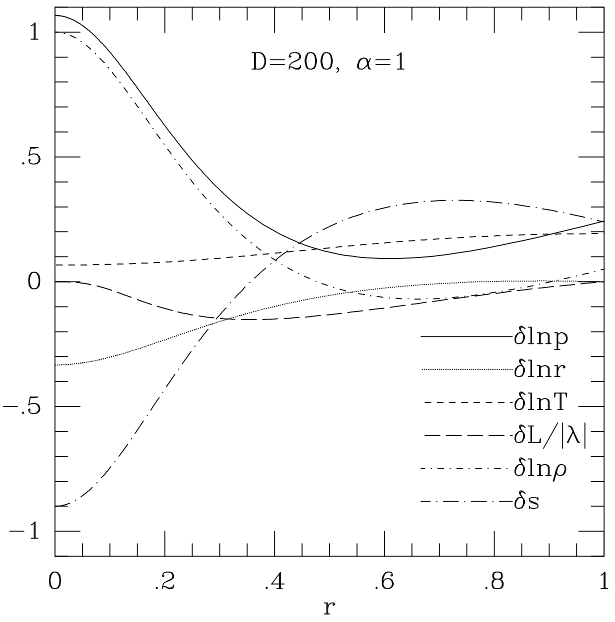
安定な場合 $(\mathbf{2})$, $D=10$



- 中心で圧力が上がる
- 温度は断熱変化の影響も受けるので、エントロピーとずれる

熱力学的安定性　(続き５)

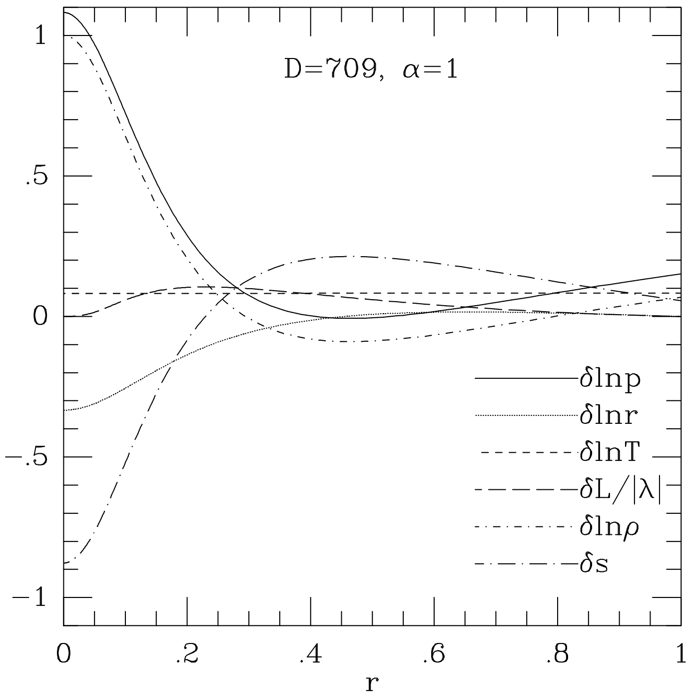
安定な場合 (3), $D = 100$



- 中心で温度も上がる
- 温度勾配はエントロピー変化を減らす向き　(この場合中心の方が低温)
- 熱力学的には安定

熱力学的安定性 (続き６)

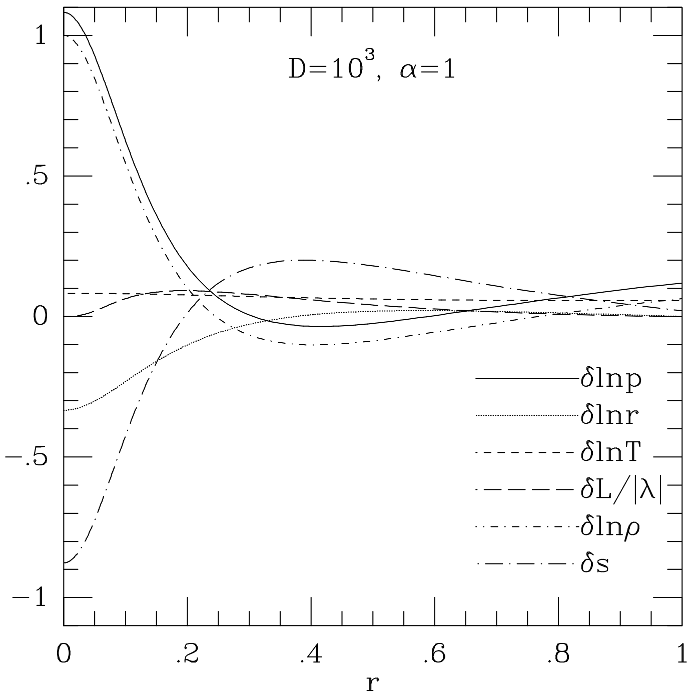
中立安定, $D = 709$



- 温度勾配ができない

熱力学的安定性 （続き７）

不安定, $D = 1000$



- 中心のほうが温度上昇が大きい

不安定になっている

有限振幅での進化

というわけで、線形解析の結果 :

断熱壁をつけて等温の平衡状態を作っても、重力が効いていると熱力学的に不安定

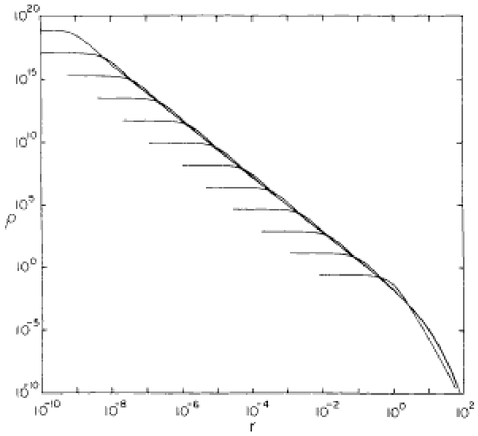
一応、 重力熱力学的不安定性」 **gravothermal instavility** という名前がついている。

摂動が有限振幅まで成長したあとの進化 :数値計算で調べる。

Hachisu *et al.* **(1978)** : 自己重力流体について数値計算した。

Lynden-Bell & Eggleton (1980): 流体の自己相似解を求めた

Cohn (1980): 流体近似を使わない軌道平均フォッカー ·プランク方程式の数値積分から、自己相似解が実現していることを示した。



ガスとN体の違い

このあたりの進化、すなわち重力熱力学的不安定や自己相似解については、流体近似、フォッカー・プランク近似を使って分布関数の進化を数値計算する方法、N体計算の間の一致は素晴らしくよい。

何故か、、、特に、流体近似がそれらしい結果を出す理由というのは、本当には良くわかっていない。

ミクロには、流体と多体系は全然違う。

流体 :平均自由行程が短い。局所的には熱平衡

多体系 :平均自由行程 (緩和時間 × 速度) が系の大きさより大きい。分布関数が局所的でない。(運動の積分の関数)

流体近似がうまくいかない場合 :

- 粒子が等質量ではない時
- 速度分布に非等方性が発達する場合

自己相似解の後の進化

自己相似解 :有限時間で中心密度が発散

「その後」どうなるか ?が問題

- 球状星団
- 矮小銀河の中心部
- 普通の銀河の中心部 （？）

今までに密度が発散しているはずのものもある。

- 途中でブラックホールに、、、(普通の球状星団では難しい)
- 連星によるエネルギー供給で反転

連星 ：3 体散乱で確率的にできる (3 つ粒子があれば、そのうち 2 つを連星にして重力エネルギーをとり出した方がエントロピー的には得なので)。

普通の星の中心での核融合と同様にエネルギー供給として働く

→ 収縮を止めて、安定な定常状態になる ?

自己相似解の後の進化(2)

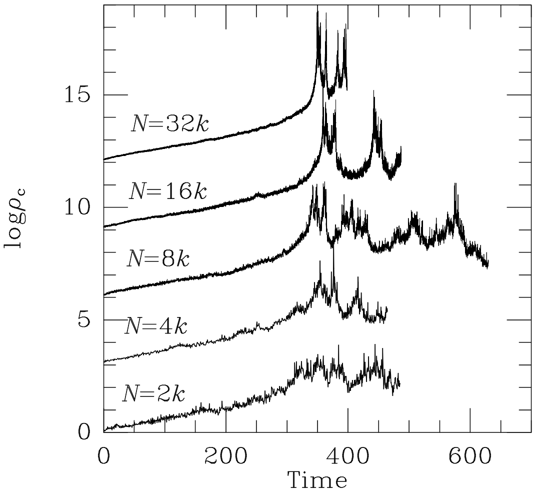
Henon (エノン ·ハイレスのエノンと同一人物) **(1975):** モンテカルロ法 (といっても、メトロポリス法みたいなのとは全然違う、、、詳しくは後述) での計算にエネルギー供給を人為的に入れて、ホモロガスに膨張する解を得た。

Heggie 1984, Goodman 1984: 同様な結果

Sugimoto and Bettwiser 1983: ホモロガスに膨張が実は不安定であることを、流体近似の数値計算から 発見」

1985 年には他のグループも確認、 **1986** 年にはフォッカープランクでも確認。

1996 年 :N 体計算でも確認



応用例

というのか、最近どんな研究がされているのかを脈絡なく紹介。

- 球状星団
  - 星の進化、親銀河との相互作用
  - 蒸発のメカニズム
- 大規模構造・銀河団
  - **“Universal” profile**
- 惑星形成
  - 暴走成長
  - 暴走成長の停止とその後

というような話をちょっとします。

球状星団

球状星団とはどんなものか：

- 銀河内にある（系外銀河にもたくさん見つかっている）
- 典型的な質量は $10^{5\sim6}$ 太陽質量。星の数ではもう数倍程度
- 典型的な大きさは **10** パーセクくらい
- 星の年齢：決定誤差の範囲内で宇宙年齢と一致（宇宙より古いとかそういう話もないではないが、、、）
- ほぼ球形で、回転もあまりない
- 銀河内の球状星団の分布も、ほぼ球形
- 年齢：緩和時間より大部長い

おおむねここまで述べてきたような熱力学的な進化をしてきているはず。

銀河とかと違って丸く、あまり特徴がない形をしている：緩和している？

但し：

- もちろん、壁があったりはしない。逆に、親銀河の作る **tidal field** で星が出ていきやすくなっている
- 星の質量はいろいろである
- 星は進化して、超新星になったり白色矮星になったりする

というあたりが違う。

球状星団の数値計算

理想化された単純な系でも、結局は数値計算しないとどうなるかはよくわからない

現実的なものがどうなるかの理解にはいっそう数値計算が重要

が、実はそういう研究は最近まであまりされていなかった。

理由：

- 理想化されたものがどうなるのかあまり良くわかってなかったため
- 計算機の能力不足(上のことの理由でもある)

比較的初期の現実的な計算の例：

Chernoff and Weinberg (1990): 1次元フォッカー・プランク方程式に星の質量分布と進化、銀河の tidal field の効果を近似的にとり入れ、初期条件の広い範囲でどのような進化が起きるかをサーベイした。

数値計算の方法について

特に多体計算の方法については、後でもう少し詳しくやるとして、もうちょっと原理的にどういう方法があるかについて。

- 流体近似 :形式的には、ボルツマン方程式のモーメント方程式を作ると流体の方程式が出てくる。
  が、平均自由行程は違うので、、、
- 軌道平均フォッカー・プランク近似 (**FP**) :分布関数の時間発展を差分法で数値積分。衝突項は **2** 次までのモーメントでよいというフォッカー・プランク近似を使う。衝突項自体は、粒子の軌道にそって局所的に密度、速度分布から決まるものを積分 (軌道平均) して求める。
  分布関数の次元を落さないと使えない。 $f(E)$, 最近になって $f(E,J)$. いずれも球対称のみ。
- モンテカルロ法 :上の、分布関数の時間発展を、モンテカルロ積分する。
  $f(E,J)$ (球対称) のみ
- N 体シミュレーション : 多体系の運動方程式を直接積分。
  原理的には何でもできる。計算時間が、、、

それぞれ、一長一短というか、万能とはいかない、、、

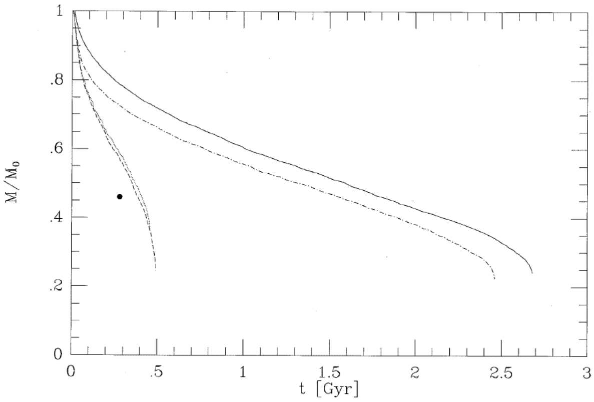
現実的な計算の例

Chernoff and Weinberg (1990): $f(E)$ を使う FP 計算。

Fukushige and Heggie (1995): N 体計算で「追試」。全然結果が違う、、、ただし、原因を明確にはしていない。

Takahashi and Portegies Zwart (1998, 2000): $f(E, J)$ を使う FP コードを N 体計算の結果を使ってキャリブレーション。多分、そこそこ正しい。

Chernoff and Weinberg とはやはり全然違った。



黒丸 ：**CW**

点線、破線 ：**Takahashi** の $f(E)$ の計算実線、鎖線 ：**Takahashi** の $f(E, J)$ の計算

違いの原因

基本的な違い :粒子が逃げる条件

$f(E)$ の計算:エネルギーでしか条件を付けられない。粒子がどこにあるのかは？

$f(E,J)$ の計算:実際に粒子がどこまで遠くにいくかがわかる。

但し、実際の銀河の効果はさらに複雑 (時間変化、 **tidal heating**)
これらはまだよく調べられていない。

大規模構造、銀河団

宇宙全体の構造：

大雑把には、宇宙の質量のほとんどは銀河が担っている。

銀河の宇宙内での分布は一様からは程遠い(宇宙の大規模構造)。

また、銀河団のような集団を作っている銀河というのも結構多い(ちょっとこれは定義の問題でもあるが、、、)

これらの構造はどのようにしてできたのか？

現在の標準的な理解：ビッグバンからの宇宙膨張の間に、もともとあったゆらぎが重力不安定(これは熱力学的なものではなくて力学的な不安定)によって成長してできた。

これは、基本的には球状星団とは違って熱力学的な進化として理解できるわけではない。

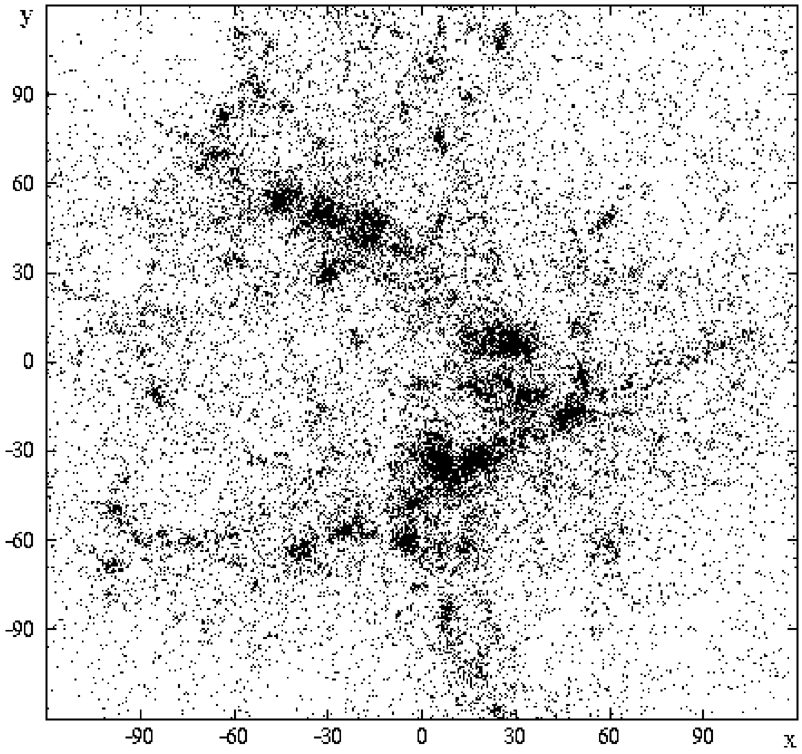
理由：

宇宙年齢にくらべて緩和時間が圧倒的に長いため。

が、そういうことを全然考えなくてもいいかというと、そうでもない

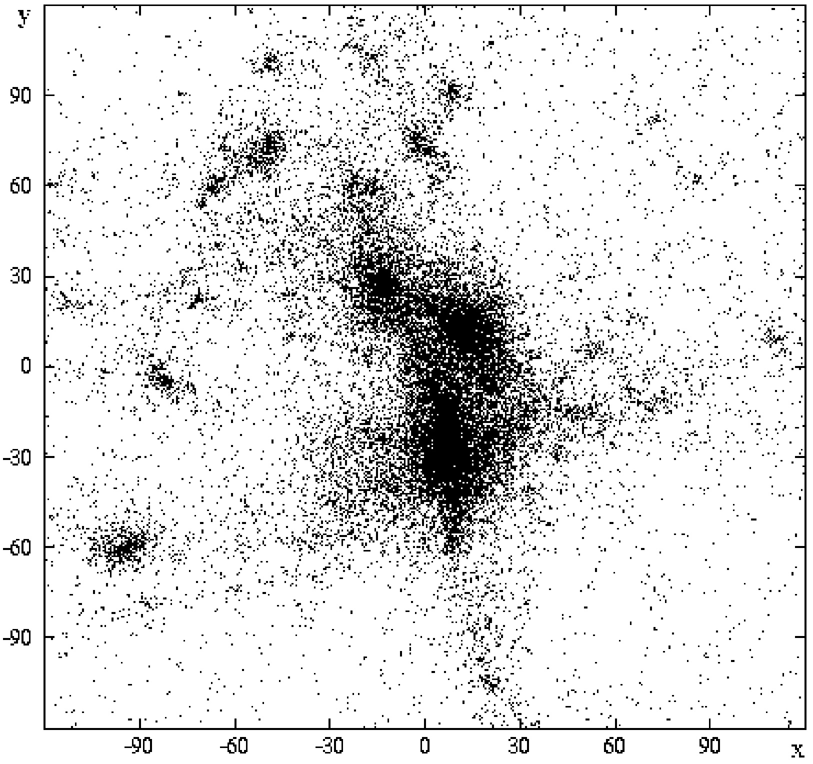
- 銀河団や大規模構造を考える時には、銀河が構成要素：実効的な粒子数は小さい
- 現実では効かなくても、数値計算では熱力学的な進化のために結果が違ってきているかもしれない

計算例　(絵）1



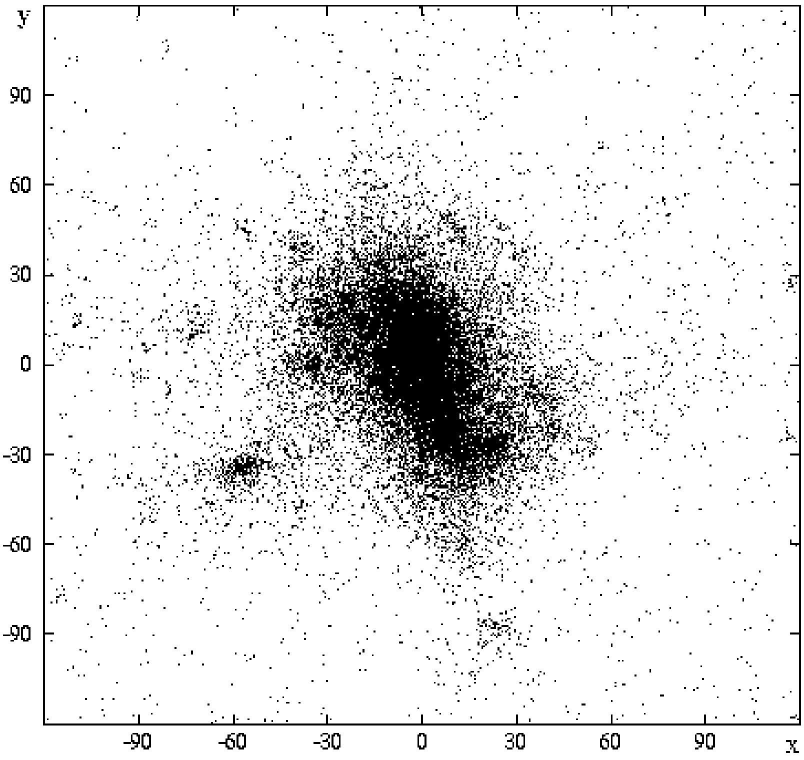
Fukushige and Makino 1996 から。銀河形成の計算例 $T = 0.5 (z = 8.7)$

計算例　(絵）2



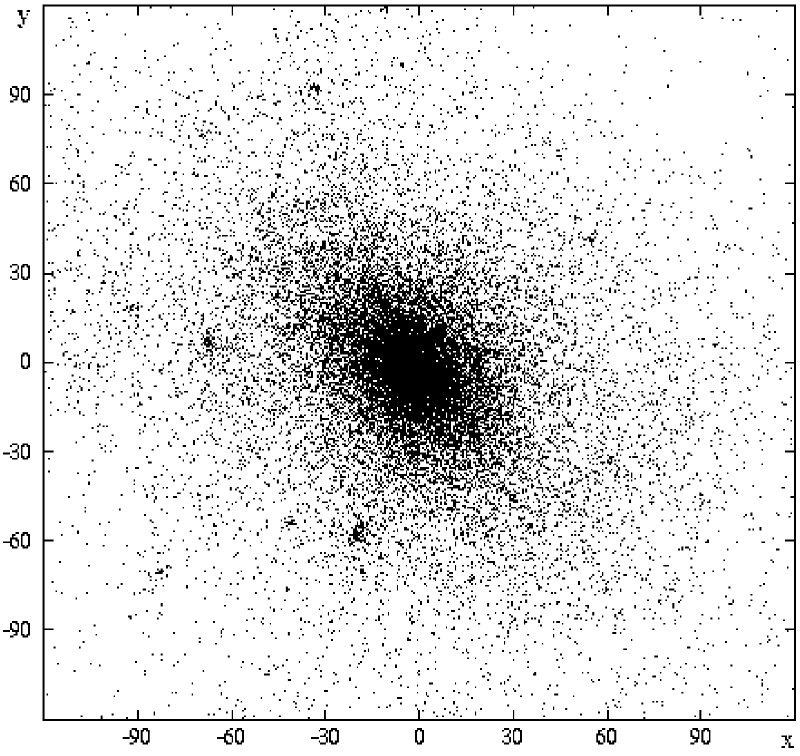
$T = 1$

計算例（絵）3



$T = 1.375$

計算例　(絵）4



$T = 3.125 (z = 1.8)$

数値計算の問題の例

Navarro, Frenk and White (1996)

様々な宇宙モデルで与えられる初期条件から、銀河団の形成の数値シミュレーションをして、結果としてできる銀河団の密度構造が、ほとんど初期条件によらないで

$$\rho = \rho_0 \frac{1}{(r/r_0)(1+r/r_0)^2} \tag{25}$$

で良く近似できるということを「発見」した。

使った粒子数：**2** 万くらい。普通の計算機で、ある程度の数シミュレーションを流そうとおもうとこれくらいが当時の限界。

Fukushige and Makino (1997): 専用計算機を使ってほぼ同じ計算を **77** 万粒子でやった。

→ 答が違う**!**

Moore et al. (1998): 並列計算機で **300** 万粒子

→ **Fukushige and Makino** と傾向は同じ。

$$\rho = \rho_0 \frac{1}{(r/r_0)^{1.5}(1+r/r_0)^{1.5}} \tag{26}$$

と、中心で密度が高い。

Navarro, Frenk and White が間違った理由：粒子数の不足のため、中心では熱力学的な緩和がおきていた。

惑星形成

惑星形成の標準モデル (京都モデル) — **1980** 年代の理解

原始太陽系星雲の中でダストが赤道面内に沈澱

↓

重力不安定でダスト円盤が分裂、微惑星に ($10^{18} g$ くらい)

↓

互いの重力で微惑星同士が合体を繰り返す

↓

そのうちに惑星になる

標準モデルの問題点 :惑星が出来るのに時間がかかり過ぎる。外側程時間がかかって、木星あたりで太陽系の年齢を超える。

が、木星はある

ので、なにかがおかしいはず。

何故時間がかかるか？

微惑星が合体・成長するにつれて、成長が遅くなる

成長する
↓
ガスの抵抗が相対的に弱くなる
微惑星間の相互作用の効き方が大きくなる（緩和時間が短くなる）
↓
ランダム速度が大きくなって、実効的な衝突断面積が小さくなる
↓
成長が遅くなる

1980 年代末に、**Wetherill** 他が「これはちょっと違うのではないか」といい出した。

上の議論：「微惑星はみなおなじように成長する」と暗黙に仮定されていた。

まあ、「大きくなると成長が遅くなる」んだから、大きさが揃ってくると思うのは一見自然。

Wetherill の議論

世の中には **dynamical friction** というものがある

$\downarrow$

重い粒子と軽い粒子が混在していれば、重い方のランダム速度が小さくなる

$\downarrow$

重い粒子のほうが成長が速くなって、暴走的に成長する

彼らは、速度分散の質量依存性をモデル化したモンテカルロ計算で上のようなことが起きることを示した。

これだと、木星でもなんでもあっというまにできてしまうことになる。

問題 :本当にそんなにうまくいくのか ?

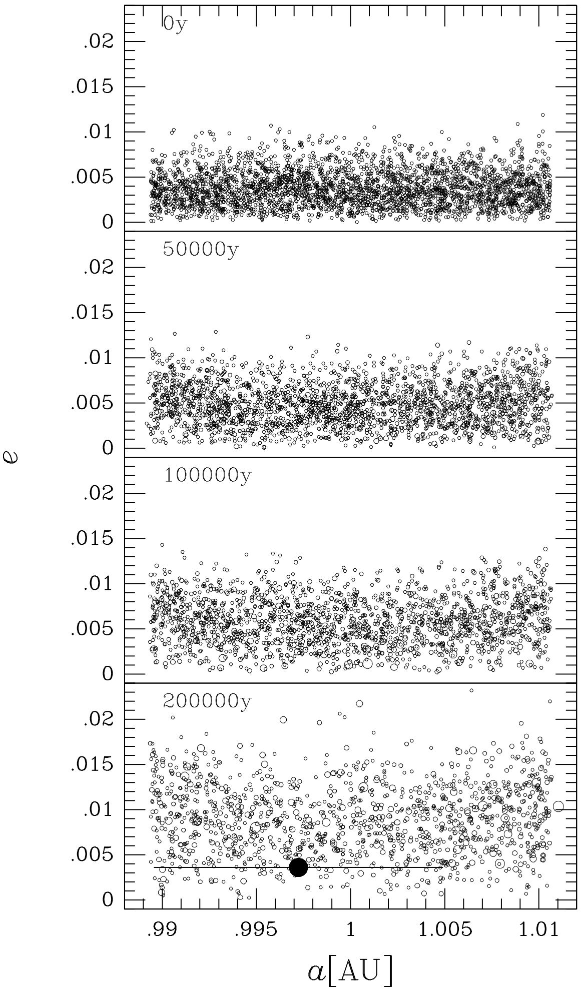
多体シミュレーションでの結果

1990年代・井田、小久保らによるN体計算での研究：

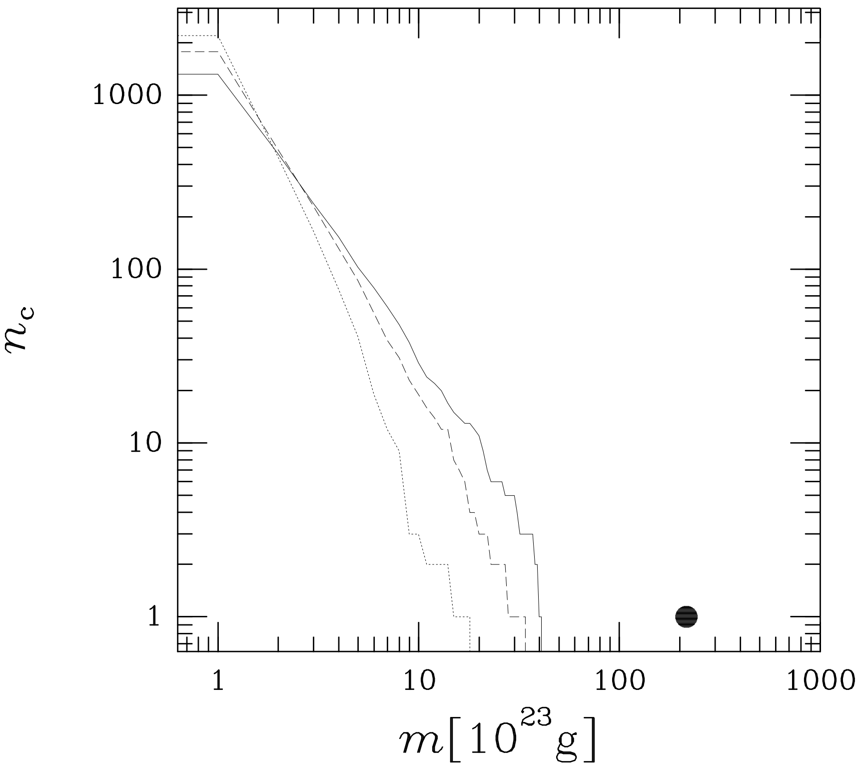
- **Dynamical friction** は確かに効く
- 但し、暴走成長したものがある程度重くなると、それが回りを加熱して結局成長速度が下がる
- 従って、ある程度重いものが少しづつ離れてできる、「寡占的成長」が実現する

- 木星コアくらいまでは成長できる
- 地球は、**1/10** くらいのところで止まる
  そこからどうやってできたかはまだ良くわかっていない

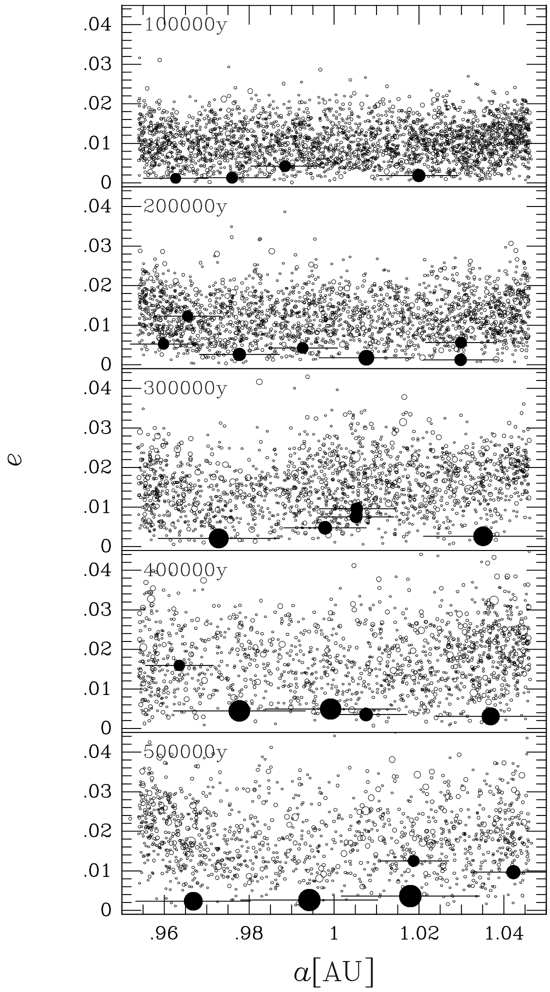
暴走成長の計算例


0y
50000y
100000y
200000y
e
.02
.015
.01
.005
0
.99
.995
1
1.005
1.01
a[AU]

微惑星の質量分布の進化



寡占的成長



100000y
200000y
300000y
400000y
500000y
e
0
.01
.02
.03
.04
.96
.98
1
1.02
1.04
a[AU]

数値計算の方法等

この種の問題 :基本的に

- より大粒子数で
- より正確な

計算をすることで、「新しいことがわかる」(こともある)。

↓

どうやって今までより「良い (大きく、正確な)」計算ができるようにするかが問題

- 計算法を改良する
- 計算機を速くする (速くなるのを待つ、自分で作る)

以下、それぞれちょっとだけ話をする (つもり)

計算法

原理的には、多体シミュレーション (業界では N 体シミュレーションということが多い、、、) はとっても単純 :

運動方程式

$$\frac{d^2\mathbf{x}_i}{dt^2} = -\sum_{j\neq i} Gm_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3}, \tag{27}$$

を数値積分するだけ。

右辺を計算するプログラム : **2** 重ループで **10** 行くらい

時間積分 : なにかルンゲクッタとか適当なものを使えばいい

というだけで話が済めばいいけれど、もちろん世の中はそんなに簡単ではない。

何が問題か ？

- 計算精度の問題 ： **2** 粒子の近接散乱、自己重力による構造形成 — 時間刻みをどんどん短くしないとちゃんと計算できなくなる
- 計算量の問題 ：右辺の計算量が $O(N^2)$ — N が少し大きくなるとすぐに計算時間が現実的ではなくなる

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

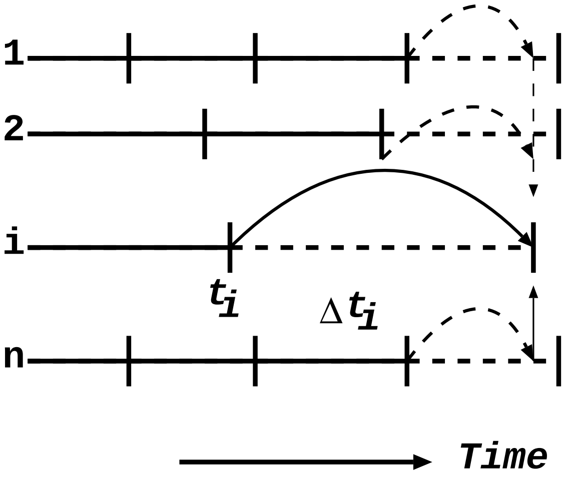
- 独立時間刻み
  - 近接散乱 → 短いタイムステップ
  - 高密度中心核 → 短いタイムステップ

  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)



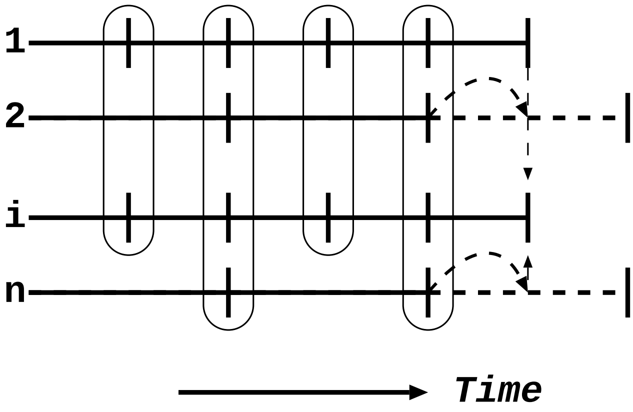
- 各粒子にそれぞれ時刻と時間刻みを与える
- 「イベント駆動」時間積分 — $t_i + \Delta t_i$ がもっとも小さい粒子が積分される

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。

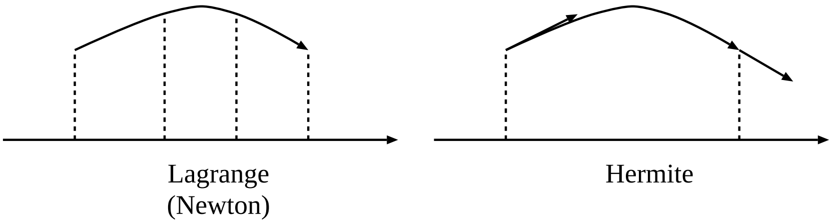
$O(N_c^{2/3})$ 個の粒子（N_c は高密度コアのなかの粒子数）を並列計算
— そんなに大きな数ではない。

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）

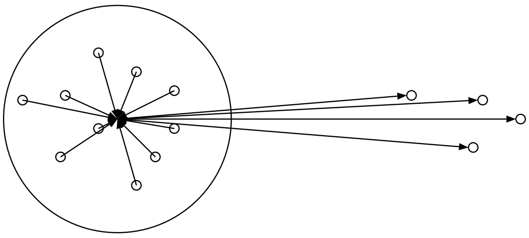
計算法 — 空間領域

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

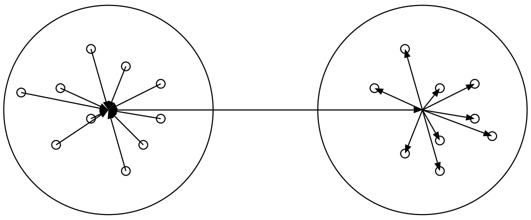


Tree

遠くの粒子からの
力は弱い

↓

まとめて計算でき
ないか ?

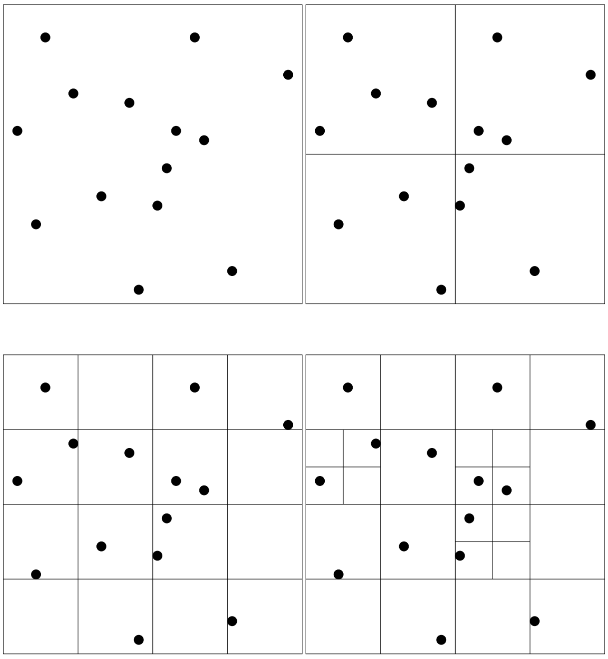


FMM

- 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

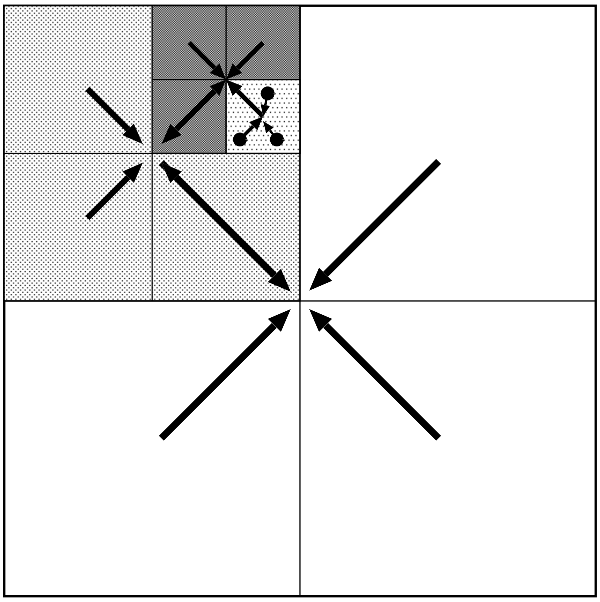
階層的なツリー構造を使う。



- 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。



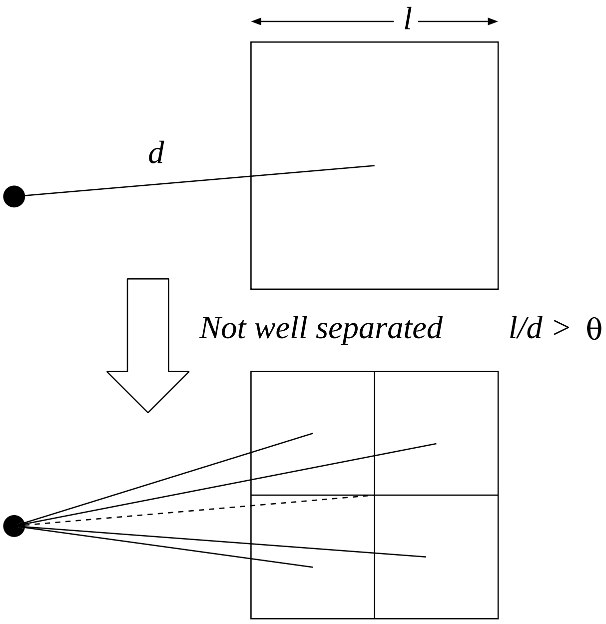
- 最下層のセル:そのなかの粒子が作るポテンシャルを多重極展開
- それ以外　:子セルの多重極展開の展開中心をシフトして加算

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)

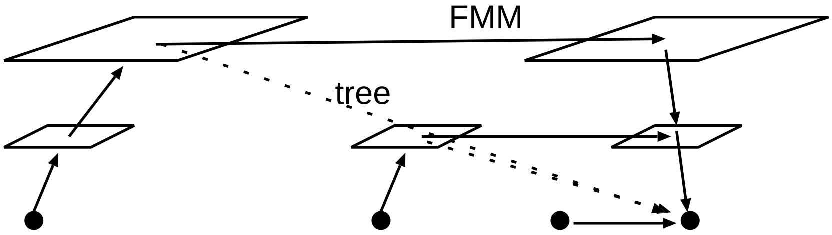
高速多重極展開法(**FMM**)

- 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

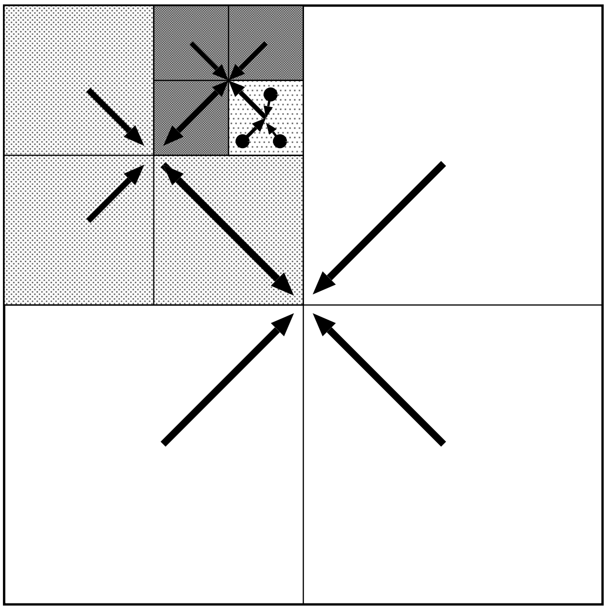


- 近くの粒子は直接計算
- 遠くにいくほど、上のセルを通って相互作用が計算される。

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)

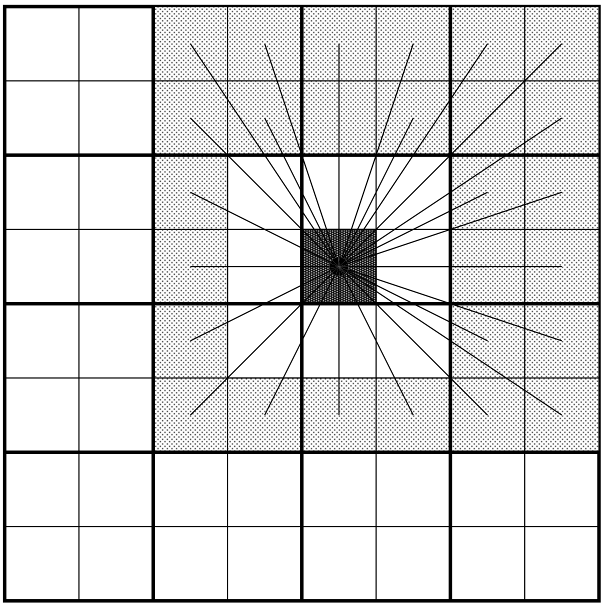
(ここはツリーと同じ)



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

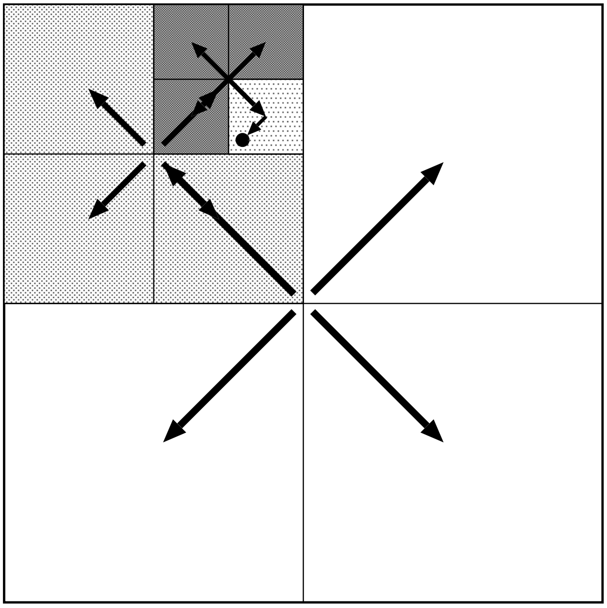
セルによるポテンシャルの和を自分の中心での調和関数展開にしておく。(**M2L**)



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:

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

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

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

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)

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

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

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”

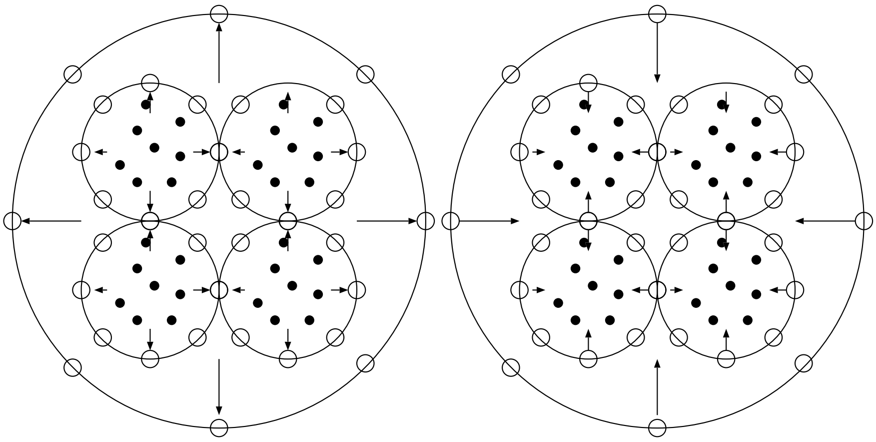
> 球面上の点が spherical t-design をなす
> $=t$ 次までの任意の多項式の球面上での積分が、それらの点上での多項式の値の和 (の $4\pi/N$ 倍) に等しい。

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)

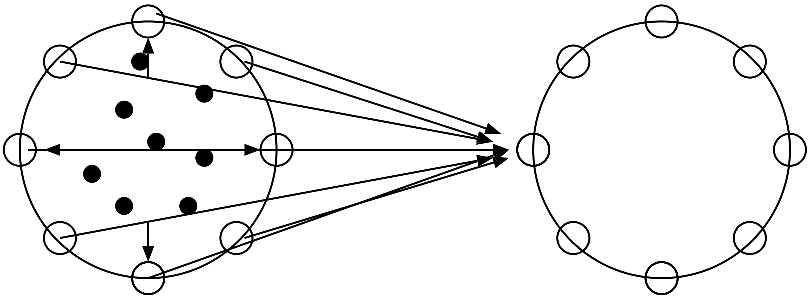
多重極展開（左）と局所展開（右）のシフト



（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

$\downarrow$

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

<u>問題</u>：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) 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 :

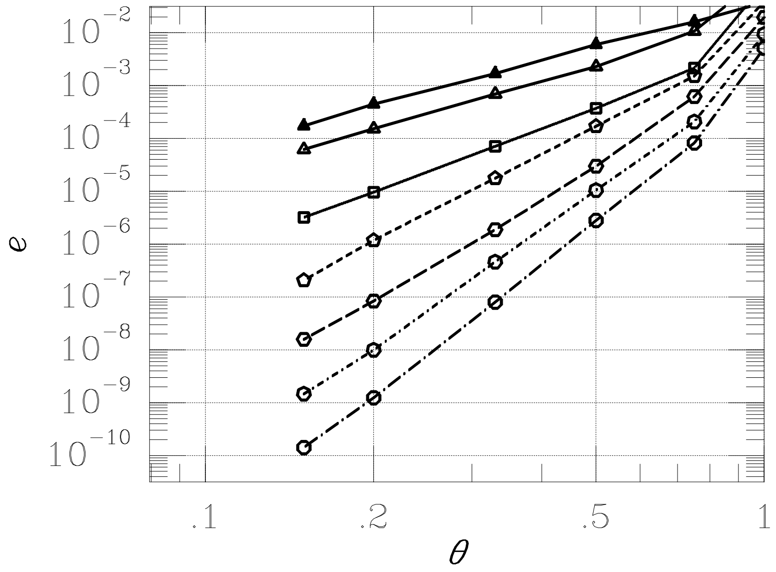
$$m_j = \frac{2l+1}{K}\sum_{i=1}^{N} m_i \sum_{l=0}^{p} (r_i/r)^l P_l(\cos\gamma)$$

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。

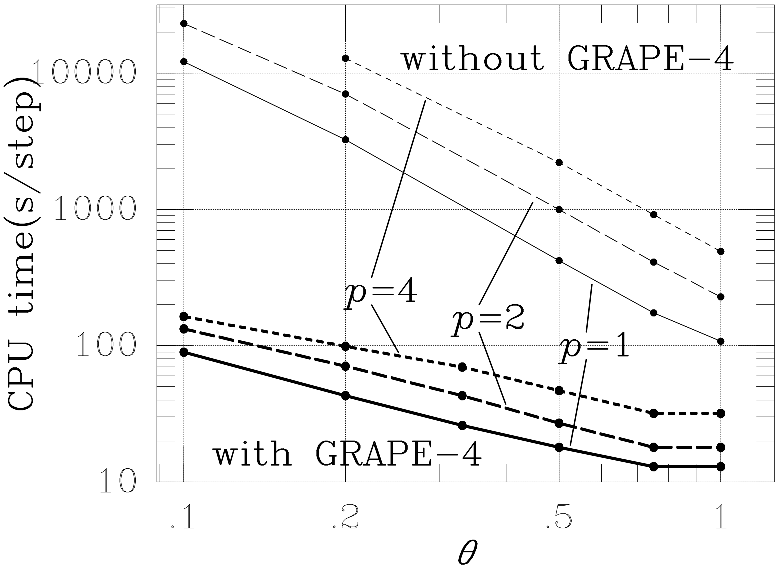
ツリー・FMM と GRAPE

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

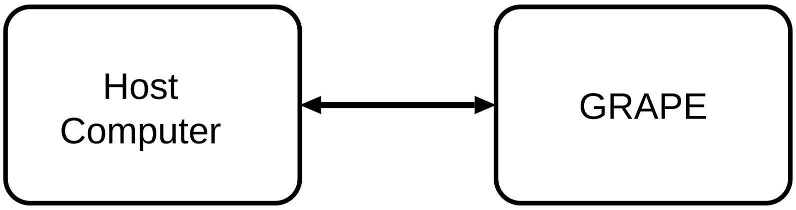
今後の大きな課題といえる。

計算機を速くする

- 寝て待っていれば速くなる (10 年で 100 倍) まあ、少しは働かないといけない
  - 並列化
  - PC クラスタを組み立てたり、、、
- 自分で作る (GRAPE)

GRAPE の基本的な考え

(GRAPE, GRAvity PipE)

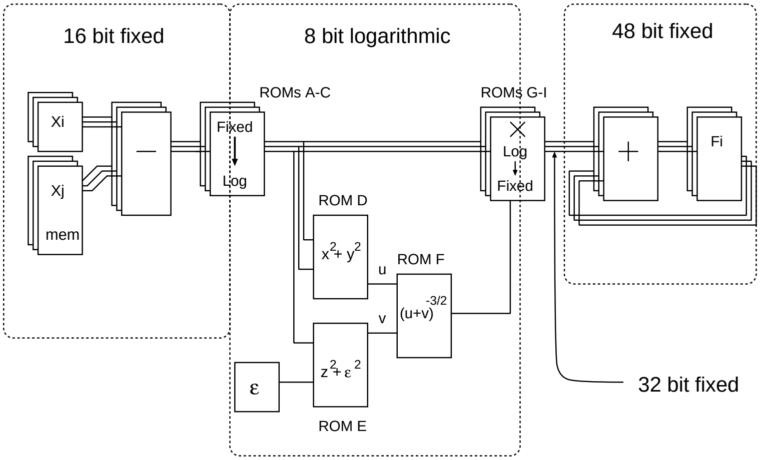


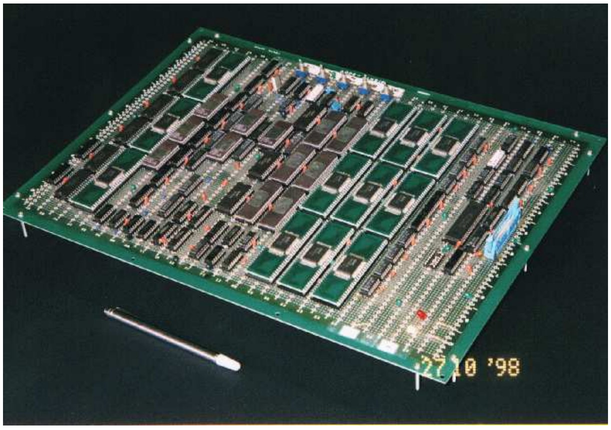
Time integration etc.　　Interaction calculation

- 専用ハードウェア
  - ハードウェア化した論理 → ソフトウェア不要
  - 相互作用用計算専用のパイプライン → 高い性能
- 汎用ホスト計算機
  - 高レベル言語が使える (**Fortran, C, C++...**)
  - 既存のプログラムが多少の手直しで使える
  - ツリーコード、独立時間刻みなども使える

GRAPE-1 パイプライン

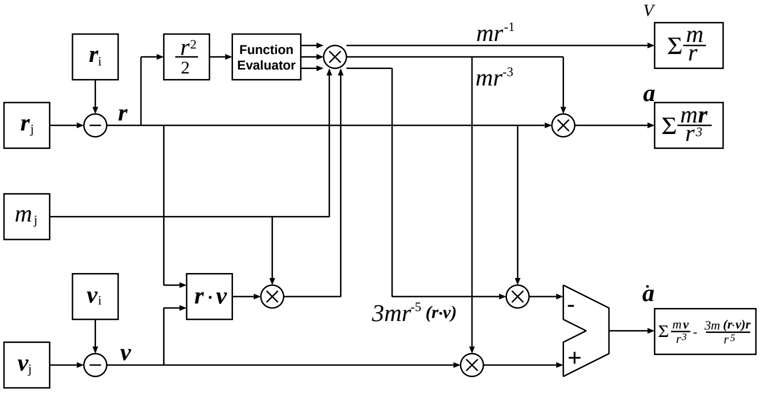
(1989, 240 Mflops)

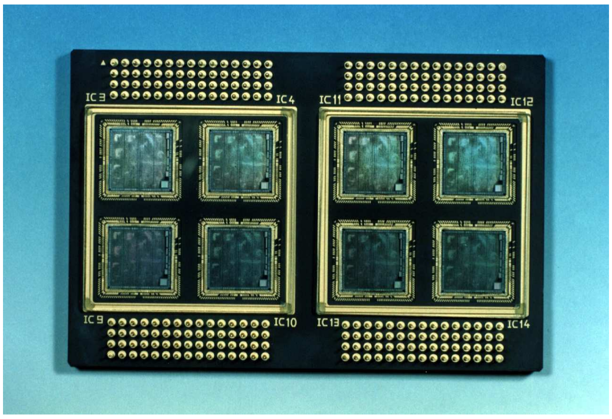




GRAPE-4 パイプライン

(1993, 600 Mflops/chip)

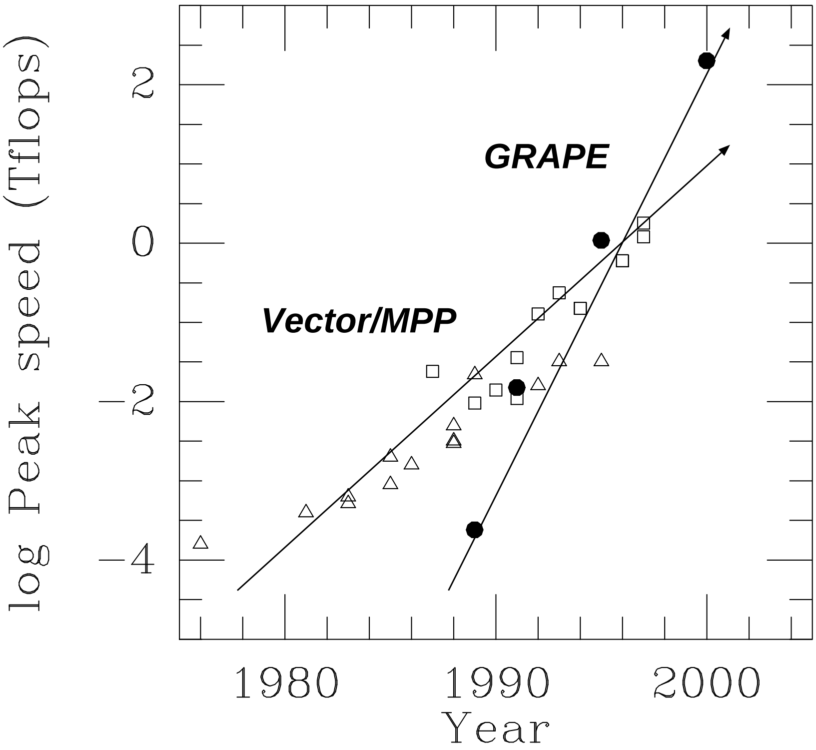




GRAPE-4

(1995, 1.08 Tflops)

汎用計算機との速度比較



GRAPE-6 計画

目標 ：

1. 重力計算用に GRAPE-4 から 100 倍以上性能を向上させた、サブペタフロップス計算機を実現し、粒子系向け専用計算機の可能性を実証する
2. FPGA を使った粒子系向け多用途計算機を構築する
3. 上の 2 つを組み合わせて、高速で応用範囲の広いシステムを構築する

一応、2001 年度で完成の予定
（「ほぼ」予定通り進行中）