

# 宇宙大規模構造形成シミュレーション

佐々木 さゆり (筑波大学大学院 数理物質科学研究科)

## Abstract

宇宙の質量分布は大規模構造と呼ばれる巨大な構造をもっていることが知られているが、この宇宙構造形成の時間発展を調べるには一様密度からの密度揺らぎを摂動として扱う摂動理論や、質量分布を粒子分布で表し重力多体系として扱う N 体シミュレーションがある。摂動理論は解析的な取り扱いが可能であるが、適用範囲が線形領域・準線形領域に限られる。一方、N 体シミュレーションは密度揺らぎが非線形な領域も扱うことができるため、非常に強力な手法である。しかしながら N 体シミュレーションにおける粒子間の重力計算は計算コストが非常に大きいため、計算コストを軽減するため様々な計算手法 (Particle-Mesh 法・Tree 法・TreePM 法など) が使われてきた。

今回は、これら宇宙大規模構造形成の N 体シミュレーション手法とそれらの長所・短所をまとめる。

## 1 Introduction

宇宙では、星や銀河などが一様に広がっているわけではなく、巨大な網目状の構造を成していることが知られており、宇宙大規模構造と呼ばれる。この構造は時間の経過と共に現在観測されているような構造に形成されたと考えられている。この構造がいつ、どのようにして形成されてきたのか明らかにすることが課題だ。

初期の宇宙で密度分布はほぼ一様と言われているが、わずかに密度のムラ “ 密度ゆらぎ ” があったこともこれまでの観測によりわかっている。この密度ゆらぎが重力により成長していき、現在のような構造ができたと考えられている。つまり密度ゆらぎの時間発展の様子を解明していくことが、宇宙の構造形成を知ることに直結している。

ここでは宇宙の構造形成を調べる方法の一つである N 体シミュレーションに焦点を当て、その手法を紹介していく。

## 2 摂動論と N 体シミュレーション

宇宙の大規模構造がどのように形成されたのか考えるには現在 2 つの方法が考えられている。それが、摂動論と N 体シミュレーションだ。

摂動論とは近似的に解析解を求める方法の一つ。密

度が一様な状態から密度ゆらぎを摂動として導入する。解析的に解けるため、時間発展を知りたい問題において有効だが、適用範囲が線形領域・準線形領域であり、非線形な状況ではうまく近似できないことが難点である。

N 体シミュレーションとは数値解を求める方法。コンピューター上に仮想的な宇宙を作り、その進化を数値計算によって追っていく。その際、宇宙の質量分布を N 個の粒子分布とみなし、重力多体系として扱う。数値計算のため、非線形な領域でも有効と言える。

## 3 N 体シミュレーションの手順

N 体シミュレーションは粒子 (分布にみたてた質量分布) の時間発展を追っていくものであるから、運動方程式が基本の方程式となる。

用いる運動方程式は物理座標での Lagrangian

$$L = \frac{1}{2} m |\mathbf{u}|^2 - m \Phi(\mathbf{r}, t) \quad (1)$$

を共動座標系で書き直した Lagrangian

$$\mathcal{L} = \frac{1}{2} m a^2 |\dot{\mathbf{x}}|^2 - m \phi(\mathbf{x}, t) \quad (2)$$

に Euler-Lagrange 方程式を適用させることで求めることができる。求められた運動方程式は以下。

$$\ddot{\mathbf{x}} + 2 \frac{\dot{a}}{a} \dot{\mathbf{x}} = -\frac{1}{a^2} \nabla \phi \quad (3)$$

共同座標： $x = \frac{r}{a} \Rightarrow r = ax$   
 $u = \dot{r} = \dot{a}x + a\dot{x} = \dot{a}x + v$   
 $x$ ：共動位置ベクトル  $v$ ：共動速度ベクトル

## 4 重力計算

式 (3) の右辺は重力相互作用による力を表している。“相互”であるから、 $N$  個の粒子があれば重力計算は  $N^2$  回行わなければならないことになる。 $N$  の値がそれほど大きくなければそのまま計算を行っても特に問題はないが、大規模構造形成シミュレーションでは  $N$  は少なくとも  $10^7$  以上になる。そのため、直接重力相互作用を計算すると膨大な計算コストがかかることになる。この計算コストを少しでも削減するため主に以下のような手法が考えられてきた。

### 4.1 PM(Particle Mesh) 法

PM 法とは領域を細かくメッシュに分け、そのメッシュ上で重力ポテンシャルを計算し、その微分から重力加速度を求める方法である。粒子間の相互作用を直接計算するのではなく、Poisson 方程式を解くことになる。計算回数のオーダーは  $N_g \log_2 N_g$  ( $N_g$  はメッシュ数)

手順は以下の通りである。

1. 計算領域をいくつかのメッシュに区切る  
十分な解像度を持つためにメッシュ数は粒子数、もしくはそれ以上にしなければならない
2. 粒子密度分布からメッシュ上の密度分布を得る  
各粒子に仮想的なプロファイルを持たせ計算を行う。プロファイルには以下のようなものが考えられている

#### (a) Nearest Grid Point (NGP) 法

- ある粒子の全質量をその粒子に最も近いメッシュに割り当てる
- 最も簡単な考え方であり計算も早いですが、粗い密度分布となる

#### (b) Cloud In Cell (CIC) 法

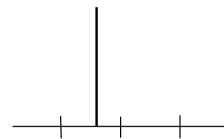


図 1: NGP 法のイメージ

$$S(x) = \frac{1}{\Delta} \delta\left(\frac{x}{\Delta}\right) \quad (4)$$

$\delta\left(\frac{x}{\Delta}\right)$  は  $x_p$  での  $\delta$  関数

- 質量を 1 メッシュ幅に様に分布させ、近いメッシュに割り当てる
- NGP 法に比べなめらかな密度分布だが、1 回微分は連続ではない

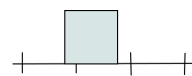


図 2: CIC 法のイメージ

$$S(x) = \frac{1}{\Delta} \Pi\left(\frac{x}{\Delta}\right) \quad (5)$$

$\Pi\left(\frac{x}{\Delta}\right)$  は  $(x_p - \frac{1}{2\Delta}) \sim (x_p + \frac{1}{2\Delta})$  の範囲で 1 となる関数

#### (c) Triangular Shaped Cloud (TSC) 法

- 粒子の位置から 2 メッシュ分二等辺三角形の底辺となるよう質量を分布
- 密度分布が連続的、1 回微分も連続

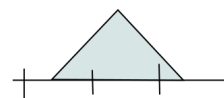


図 3: TSC 法のイメージ

$$S(x) = \frac{1}{\Delta} \Lambda\left(\frac{x}{\Delta}\right) \quad (6)$$

$\Lambda\left(\frac{x}{\Delta}\right)$  は  $(x_p - \frac{1}{\Delta}) \sim x_p$  で  $-x + \frac{1}{\Delta}$ 、 $x \sim (x_p + \frac{1}{\Delta})$  で  $x + \frac{1}{\Delta}$  をとる関数

示したようなプロファイルで 1 つの粒子がメッシュに与える質量の割合  $W$  は

$$W(\mathbf{x}_p, \mathbf{x}_i) = \int_{V_i} S(\mathbf{x}_p - \mathbf{x}) dV \quad (7)$$

となり、メッシュ上の密度場  $\rho_i$  は

$$\rho_i = \frac{1}{V_i} \sum_{p=1}^N m_p W(\mathbf{x}_p, \mathbf{x}_i) \quad (8)$$

$V_i$ ：メッシュ領域の体積、 $m_p$ ：粒子の質量で求めることができる。

3. FFT(Fastest Fourier Transform) を用いてメッシュ上で Poisson 方程式を解く

離散化された Poisson 方程式を

$$\text{離散 Fourier 変換 } \hat{f}_k = \sum_{j=1}^N f_j \exp(-ikj\Delta) \text{ で変換する}$$

ex) 離散化された一般的な Poisson 方程式

$$\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta^2} = 4\pi G\rho_j \quad (9)$$

↓ 離散 Fourier 変換を適用すると

$$\begin{aligned} \hat{\phi}_k [\exp(ik\Delta) + \exp(-ik\Delta) - 2] &= 4\pi G\Delta^2 \hat{\rho}_k \\ ([e^{ik\Delta} + e^{-ik\Delta} - 2] &= [2i \sin(\frac{k\Delta}{2})]^2 \text{より}) \\ \hat{\phi}_k &= -\frac{4\phi G\Delta^2}{\sin(k\Delta/2)} \hat{\rho}_k \end{aligned} \quad (10)$$

得たポテンシャルの Fourier 成分を逆 Fourier 変換でメッシュ上のポテンシャル  $\phi_j$  に直す

#### 4. 求めたメッシュ上のポテンシャルから

加速度を計算し、粒子の位置、速度を更新する  $a(x_i) = -\nabla\phi_i$  でメッシュ上の加速度、 $a(x_p) = \sum_i W(x_p, x_i)a(x_i)$  で粒子の加速度に直す。

この PM 法の長所は、FFT を用いることで計算が非常に早くなるという点である。また、Fourier 変換を用いているため、宇宙大規模構造のシミュレーションに必要な周期的境界条件が自然と満たされているのも利点である。しかし、メッシュサイズ以下の解像度がなく小さなスケールでの精度が必要な計算になると不向きと言えるだろう。

### 4.2 Tree 法

ある程度遠くの粒子からの相互作用をまとめて計算する方法。セル中の粒子が 0 か 1 になるまで全体を階層的に分割していく。計算回数のオーダーは  $N \log N$  ( $N$  は粒子数)

手順は以下の通りである。

1. 粒子が入った Box を 1 辺の長さが半分になる 8 個の Box に分ける。この作業を Box 内の粒子が 1 または 0 になるまで繰り返す

この階層的な様子を分割される Box を親、分割された後の小さい Box を子としてわかりやすい図にすると以下のようなになる



図 4: Tree 構造イメージ 1

(図は 2 次元で書いているので正方形を 4 つの正方形に区切っている)

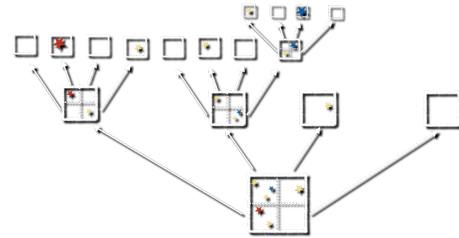


図 5: Tree 構造イメージ 2

このように最初の Box を幹として枝分かれしていく木のような様子から Tree 法と呼ばれている

2. Tree 法は遠くの粒子はまとめて重力計算を行う “遠くの粒子” の判断は以下のように行う

子 Box がまとまっている親 Box を節点と呼ぶとする。ある粒子が受ける力を考えると、その粒子から見てある程度小さな立体角以下に収まっている節点の中にある全ての粒子はその節点の重心にまとめてあるとみなし、ある粒子への力を考える。この立体角は

$$\frac{l}{d} < \theta \quad (11)$$

で定義する。 $l$  は節点に対応する立方体の 1 辺の長さ、 $d$  はある粒子から節点の重心までの距離、パラメータ  $\theta$  は見込み角と呼ばれる。この  $\theta$  を調整することで計算精度と計算量を制御する。この方法で重力計算を行い加速度を得て、粒子の位置や速度を更新していく。

この Tree 法の長所は、直接相互作用を計算すれば  $N^2$  かかる計算量を  $N \log N$  までと格段に少なくできることである。また、遠くの粒子たちはまとめて近似的に計算するが、近傍の粒子は直接相互作用を計算するため、小さいスケールの変化を精度よく見たい場合有効である。しかし粒子数が膨大になって

くと、いくら計算量が  $N^2$  に比べ少なくて来たとは言え、かなりの計算時間がかかってしまうことも考えられる。さらに周期的境界条件を実現するが簡単ではない。

### 4.3 組み合わせた手法

それぞれの手法での短所を補い、長所を有効に用いるためハイブリッド的な手法が考えられている。

#### PP-PM 法

遠方粒子から受ける力は PM 法を用いて評価、近傍粒子から受ける力は直接の和により計算する方法  
これにより、FFT による高速性、周期的境界条件を保ちつつ、近距離の力も精度よく評価できる  
精度が必要な大粒子数の計算にはあまり向かない。

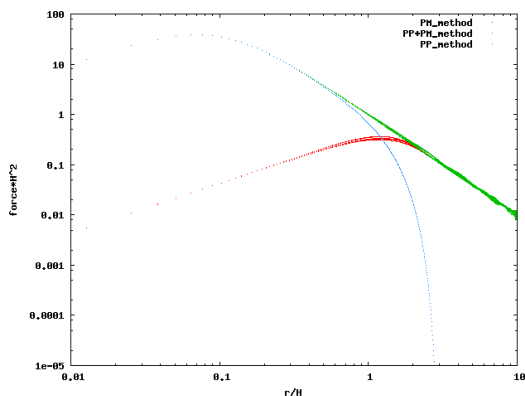


図 6: PP 法+PM 法での force Shape

#### Tree-PM 法

遠方粒子から受ける力は PM 法を用いて評価、近傍粒子から受ける力は Tree 法を用いて計算する方法  
膨大な粒子数にも向いておりかつ高精度を有する。  
現在の宇宙大規模構造形成のシミュレーションにはこの方法がよく用いられている。

## 5 軌道積分

今回は無衝突系でよく用いられる Leap frog 法について述べる。先に求めた式 (3) を以下の二つの式

に分けて積分を行う。

$$\begin{cases} v_{n+\frac{1}{2}} = \frac{1 - H\Delta t}{1 + H\Delta t} v_{n-\frac{1}{2}} - \frac{\Delta t}{a^2(1 + H\Delta t)} \nabla \phi & (12) \\ x_{n+1} = x_n + v_{n+\frac{1}{2}} \Delta t & (13) \end{cases}$$

式を見てわかるように式 (12) は速度に関する、式 (13) は位置に関する積分を書き下した形になっている。ただし、 $\dot{a}/a$  を  $H$  とした。  $H$  は Hubble パラメータと呼ばれる宇宙論パラメータの一つである。また、速度は  $v_{n+1/2}, v_{n-1/2}$ 、位置は  $x_{n+1}, x_n$  と時間のずれたものが入り混ざっている。このようにすることで、図のようなかえる跳びのように速度と位置を更新していくことができる。

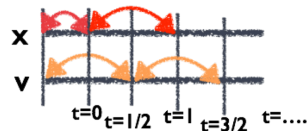


図 7: Leap frog のイメージ

この Leap frog 法の特徴は、時間についての 2 次精度、シンプレティック性 (エネルギーを良く保存する) といった点が挙げられる。

## 6 Summary

このように様々な数値計算の手法がこれまでに考えられている。ただ 1 度計算を行っただけではどの方法でも計算が正しくできているか判断はできない。いずれにおいても粒子数やパラメータを変えるなどして複数回計算する必要があると考える。

また、今回紹介した数値計算の方法は重力相互作用による粒子の時間発展のみを考えるものであった。しかし現実には宇宙に存在するバリオン粒子は流体でもある。このバリオン粒子の流体力学効果も含んだ数値計算を自身は行っていきたいと考えている。

## Reference

Yoshikawa, K., Fukushige, T., 2005, PASJ, 57, 849  
Hockney, R.W., Eastwood, J. W. 1988, Computer simulation using particles, 1988,  
Xu, G. 1995, ApJS, 98, 355