

多くの小塊を持つガス中での銀河形成

— GRAPE-5 システム用コードの開発 —

幸田 仁 (東京大学大学院理学系研究科附属天文学教育研究センター)

アブストラクト

本プロジェクトでは、分子ガスによる冷却効果も考慮し、銀河形成初期に存在した可能性のある分子ガス雲 ($\sim 10^6 M_\odot$) の、銀河形成に果たす役割の理解を目的としている。また銀河ガスの“温度”を如何にモデル化すべきか、を解明する。計算の分解能によって、計算しているガスの“温度”は以下の2つの異なる意味を持つ: (1) 低分解能ではガス雲の速度分散、(2) 高分解能では原子の速度分散。これまで数値計算は低分解能で、扱うガスは(1)の性質を持ち、銀河円盤の動力学計算ではしばしば等温円盤を仮定した。しかし銀河形成計算は低分解能にもかかわらず、本来(2)の温度で決まる放射冷却効果を、(1)の温度から計算している。本プロジェクトでは、N-body/SPH法を採用し、初期分子ガス雲まで分解して温度の問題を解決し、銀河ガスの収縮過程を正しく扱うことを目的とする。1銀河内部での粒子数 $N = 10^{6-7}$ 、質量分解能 $10^{3-4} M_\odot$ の計算を目標とし、本年度は、そのためのコードの開発を行った。特に、重力計算部分について、Barnes のツリーコードを元に、ツリー + GRAPE コードの開発を行った。

Tree+GRAPE コード

このコードでは Barnes のツリー法をベースに、まず(1)ツリー構造を作り、(2)各レベルのセル内にある粒子数を検索する。その後、(3)セル・セル間のオープン条件を適用しながら、再帰的にツリーを下り、重力

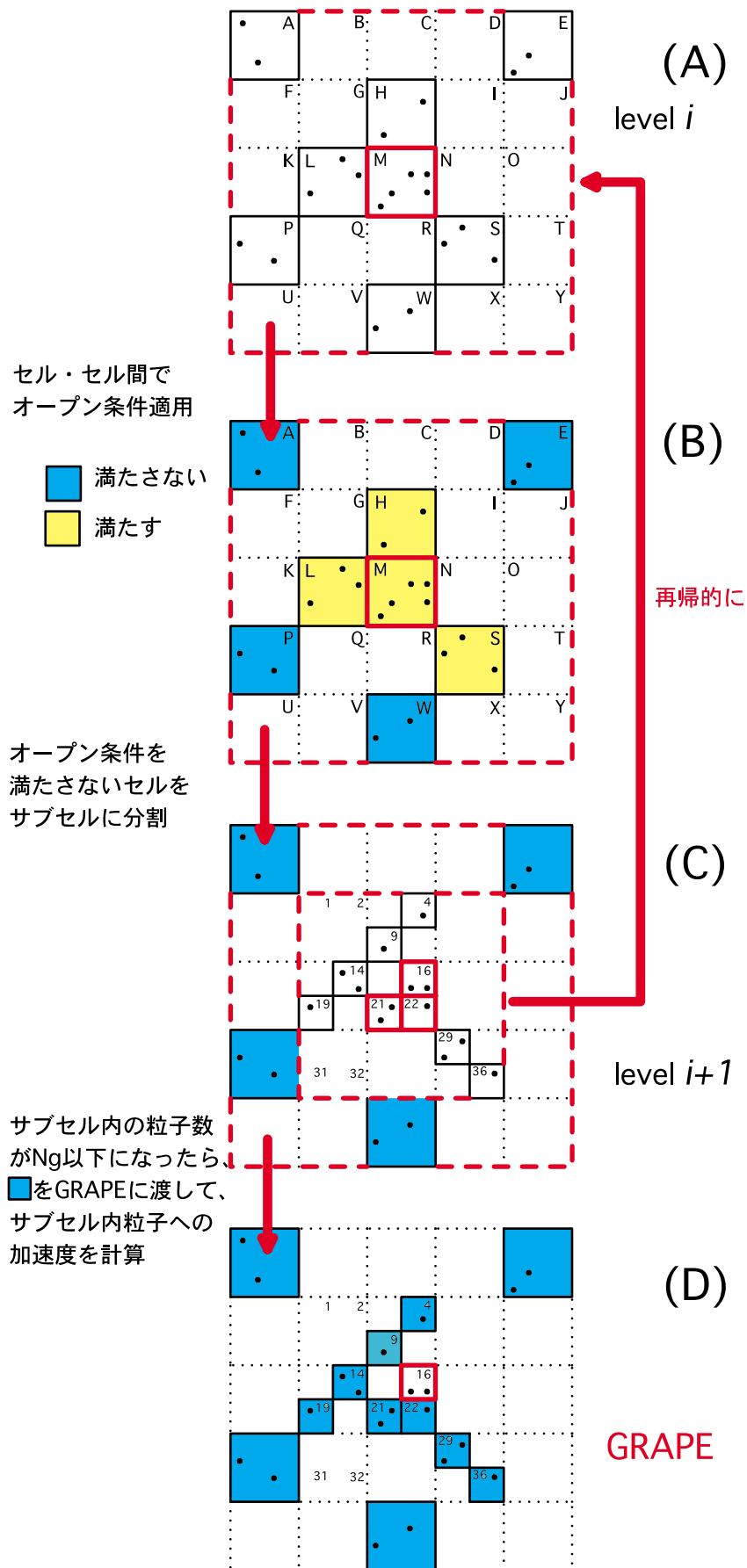
を計算する。

図は(3)について模式図である。まずあるレベル i に着目する [図 A]。A, B, ... は同じレベル i にあるセル。まずセル M に着目し、セル M のオープン条件を満たすセルを、同レベルの中から選ぶ (たとえば A, E, P, W、図 B)。

次に1つ下のレベル $i+1$ に移る [図 C]。セル M と、セル M のオープン条件を満たさなかったセルを、サブセルに分割する (1, 2, 3, ...)。内側赤破線の外側のセル (A, E, P, W) は、すでにセル M のサブセル 16, 21, 22 の相互作用リストに入っている。つまりサブセル 16, 21, 22 のオープン条件は、内側赤破線の内部だけに適用すればよい。図 C の内側赤破線の内部だけを考え、たとえばセル 16 に注目すると、状況は図 A と同じになっている。図 (A)-(C) の操作を、再帰的に繰り返しツリーを下る。

上記のようにすると、各レベル毎に1セル内の多数の粒子にたいして、一度に相互作用リストを作ることになり、粒子毎にツリーを下る場合に比べて操作の数が激減し、高速化が可能になる。

さらに、注目したセル内部の粒子数が N_g 個 (=2000 個程度が良い) 以下になった場合には、それまで作った相互作用リストを GRAPE に渡し、注目したセル内部の粒子に対する重力を GRAPE を用いて計算する [図 D]。この方法では、ツリー法での相互作用リストと GRAPE に渡す粒子のリストが自然に一致し、ツリー法と GRAPE を組み合わせる良い方法と言える。



結果

Plummer モデルを用いて CPU 時間の測定を行った。GRAPE-5 システムの 1 ノードを使った結果を表に示す。1 ステップあたり、ツリー構築 (mkt)、重力計算 (grv) に掛る CPU 時間、全 CPU 時間、全使用メモリ量を示してある。

テストした範囲 (粒子数 10^4 – 10^7) で、計算時間は粒子数に比例する。1 ノードでの計算は、メモリリミットではなく CPU リミットになっていて、銀河形成のように高時間分解能で長時間積分が必要な系の場合、現実的な時間で、粒子数 $\sim 10^6$ 程度まで計算可能である。重力に関しては、1 ノードで最低目標粒子数に到達出来る。

表には、ツリー構築に掛る CPU 時間の全 CPU 時間に対する割合が示してある。全体として約 3 割がツリー構築に使われている。また粒子数が増えるにつれ、ツリーが深くなるため、割合は上がっている。最終的な目標である粒子数 $\sim 10^7$ の計算のためには、並列化が必要になる。重力計算の時間は粒子数と比例関係にあるため、並列化した場合に重力計算部分はノード数に反比例すると考えられる。しかしツリー構築部分は、単純に並列化した場合にはボトルネックになると予想される。

現在はセルに 1 粒子しか入らない場合をツリーの末端としているが、GRAPE を使う場合、粒子に行き着くまでセルを開かず、セル内の全粒子について重力を計算してしまう方が速い場合も多い。そのため、ツリーを末端まで作らず、かなり上のレベルでツリー構築をやめることで、ツリー構築に掛る時間を減らすことが出来るのではないかと、考えている。

このコードを使って、今後、まず 1 ノードの計算から、本計算を行っていきたい。

粒子数	10^4	10^5	10^6	10^7
CPUmkt	0.067	0.26	3.7	58
CPUgrv	0.017	0.93	9.3	108
CPUtot	0.087	1.2	13	167
mkt/tot (%)	20	22	29	35
メモリ		18M	136M	1.3G

表 1: 各粒子数に対する、1 ステップあたりのメインルーチンの CPU 時間 (単位は秒)。CPUmkt: ツリー構築。CPUgrv: 重力計算 (図を参照)。CPUtot: 全 CPU 時間。