

希薄系における短距離分子動力学法シミュレーションの実装手法

林 亮子 樋渡 保秋[†] 堀口 進

北陸先端科学技術大学院大学 情報科学研究科

[†] 金沢大学 理学部 計算科学科

本稿では、短距離相互作用の働く希薄系で、シミュレーションの進行とともに原子が結合してクラスターを作るシミュレーションを扱う。このシミュレーションで使用する相互作用は短距離相互作用であるため、空間を小領域(セル)に分割する、領域分割に基づいたドメイン分割法を使用する。今回扱う原子の密度は希薄かつ不均一であるために、最適なセルの大きさは空間の部分ごとに変化する。これまではドメイン分割法を実装するために、セルごとのループを使用していたが、原子の密度変化にともなって実行時間が増加する欠点があった。そこでドメイン分割法の実装法として、セルごとのループを陽に使わず、原子ごとのループを使用する方法を用いて、これまでの実装法と性能を比較した結果を報告する。

Implementation methods of Short-range Molecular Dynamics Simulation for Dilute Gas Phase

Ryoko Hayashi Yasuaki Hiwatari[†] Susumu Horiguchi

School of Information Science, Japan Advanced Institute of Science and Technology

[†] Department of Computational Science, Faculty of Science, Kanazawa University

This report addresses a classical short-range molecular dynamics simulation for making clear the formation process of carbon clusters. This simulation uses short-range interactions, therefore we use domain decomposition method based on the division of simulation space into small pieces of space, cell. In this simulation, density of atoms is dilute and uneven, therefore optimum cell size becomes different between gas phase and clusters. Our program used cell-base implementation and repeat loops as the number of cells, but it causes an increase of execution time as the growth of clusters. In order to solve this problem, we try another implementation: hybrid implementation, based on loops that repeat as the number of atoms. Here we compare these implementations and discuss their performance.

1 はじめに

分子動力学法(MD)シミュレーションは、物質を構成する分子や原子一つ一つの運動を模擬することによって物質の状態の時間変化を調べるシミュレーション手法である[1]。より複雑な物質や詳細な現象を扱うために、並列処理を含むMDの高速化について多くの研究が行われている。MDの優れた並列化手法の一つに、空間を「セル」に分割して計算量を削減するリンクセル法を並列処理に拡張したドメイン分割法がある[2]。しかし、低密度で不均一な物理状態のシミュレーションでは、最適な空間の分割法が一

意に定まらず、ドメイン分割法の高速度実行は難しい。本稿では、ドメイン分割法の原子ごとのループによる実装法を検討し、セルごとのループを用いた実装法と比較した結果を報告する。

ドメイン分割法で低密度の希薄系を扱う場合、大きいサイズのセルを使うことで高速化できることが知られている[3]。これまでに筆者らはカーボンクラスター生成過程を調べるため、ターソフ・ポテンシャル[4][5]による短距離相互作用を用いた希薄系のMDシミュレーションプログラムを並列実装し、高速化を検討した[6]。その結果、密度が一樣な場合には最適であったセルサ

イズが、局所密度の変動によって最適でなくなり、オーバーヘッドを生じることを示した。この問題を解決するため、セルを陽に使わず、原子の数だけ処理を繰り返すハイブリッド実装法を提案した [7]。本稿では、ハイブリッド実装とこれまで利用してきたセルベース実装との性能比較を行なう。

本稿の構成は以下の通りである。第2章は、本稿で扱う MD シミュレーションおよびドメイン分割法の概要を示す。特に、本稿で比較した2つの実装法であるセルベース実装とハイブリッド実装の概要を述べる。第3章はセルベース実装とハイブリッド実装による実行時間を示し、その有効性を議論する。第4章はまとめである。

2 分子動力学法

2.1 シミュレーションの概要

本稿で扱う MD シミュレーションは古典 MD と呼ばれ、各分子に働く力を分子間相互作用から計算してニュートンの運動方程式に代入し、数値積分を行うことにより時間刻み (タイムステップ) ごとの各分子の位置と速度を計算する。分子間相互作用は多くの場合、分子間距離に依存するポテンシャル関数で定義する。本稿では、このポテンシャル関数にターソフ・ポテンシャル [4] を使用し、パラメータの数値は丸山ら [5] に従う。ターソフ・ポテンシャルの概要は文献 [4] および文献 [5] に譲る。ターソフ・ポテンシャルは3体力を考慮するので、本来の計算複雑度は N 個の原子を扱うと $O(N^3)$ である。しかし、このポテンシャルは相互作用の影響範囲が有限な短距離相互作用であるため、相互作用の局所性を利用して計算複雑度を $O(N)$ にすることができる。

本稿のシミュレーションは炭素気体からのクラスター生成、特に籠状分子であるフラレンの生成過程を想定している。実験的には、フラレンはヘリウム、アルゴンなどの希ガス中でそれらの分子と衝突し、運動エネルギーを失いながら成長する。MD で希ガス分子まで扱うのは実行時間の面から困難であるため、丸山らは

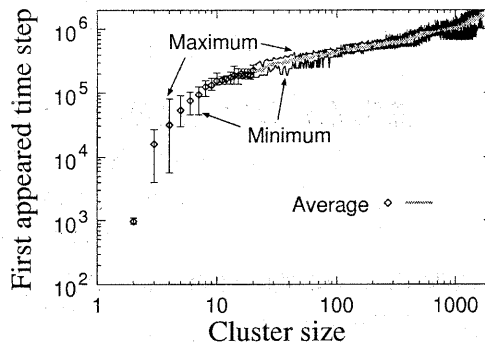


図1: クラスターの大きさとそのクラスターが最初に出現するタイムステップとの関係 (8000 原子, $\rho = 0.1\text{nm}^{-3}$, 3000K)

クラスターごとの温度制御によってクラスターと希ガス分子の衝突を簡略化して扱う手法を提案し、代表的なフラレン分子である C_{60} を生成した [5]。クラスターごとの温度制御は、クラスターの回転、振動、並進の運動エネルギーから得られる温度に係数を乗算して一定にする。本稿では丸山らの方法を使用し、クラスターごとに計算した3つの温度をそれぞれ 3000K に保存した。このシミュレーションはエネルギー一定の式を使用するために、温度は保存しない。温度を一定にするため、能瀬の方法による温度制御を 100 タイムステップごとに行ない、温度を 3000K にした。1 タイムステップは 0.407×10^{-15} 秒に相当する。

本稿のシミュレーションで想定すべき密度を議論する。ここで開発したプログラムの目標は、密度 0.1nm^{-3} でのシミュレーションである。図1はこのシミュレーションで生成するクラスターの大きさと、そのクラスターが初めて生成した時間の関係を示す。図1に示すように、本稿が目的とするシミュレーションでは、時間発展するとより大きなクラスターが生成することがわかる。このようなシミュレーションでは、局所的に原子の密度が不均一になる。一般に、相互作用影響範囲中に何個の原子があるかは、密度 ρ と相互作用影響範囲 r_c を用いて次式で表せる。

$$\frac{4}{3}\pi\rho r_c^3 \simeq 0.036\rho, (r_c = 0.2\text{nm}). \quad (1)$$

表 1: 密度と相互作用影響範囲内の原子の数の関係

ρ (nm^{-3})	相互作用影響範囲内の原子の数	備考
0.1	0.0036	平均密度
100.0	3.6	6 員環と同程度

想定すべき密度での、式 1 の数値を表 1 に示した。表 1 によると、空間全体の平均密度である $0.1nm^{-3}$ の状態では、ほとんどの空間に原子が存在しないことがわかる。一方、クラスターが生成すると、今回使用しているポテンシャルでは原子間距離 $0.18nm$ 程度で 6 員環に近い構造を作る。このことから、クラスターの内部は $100nm^{-3}$ 程度の密度になると予想されるので、 $0.1nm^{-3}$ から $100nm^{-3}$ までの密度を想定したシミュレーションプログラムを開発する必要がある。

2.2 ドメイン分割法の概要

ドメイン分割法の概要を示す。図 2 は、ドメイン分割法を 2 次元空間の MD シミュレーションに適用した例である。図 2 では正方形のセルを使用して、セルの大きさを相互作用影響範囲 r_c よりも大きく定める。そして 1 次元トラス結合の 4PE を使用し、1 行のセルを 1 つのドメインとして PE に割り当てる。セルの大きさが r_c よりも大きいため、分子間相互作用を計算すべき分子の組み合わせは、各セル自身と 8 近傍セル中に含まれる。そのため、図 2 では各 PE は 2 近傍の PE とだけデータを交換すればよい。3 次元空間のシミュレーションの場合はセルは立方体とし、セルの大きさを力の切断距離よりも大きく定める。このとき、各セル自身と 26 近傍セル中の分子を参照すれば、相互作用の計算に必要な分子の組み合わせが得られる。このように、ドメインを使用することで、計算量、通信量および通信相手数を削減することができる。ドメイン分割法では N 個の分子を P 個の PE を用いて計算するとき、計算複雑度を $O(N/P)$ にすることができる。

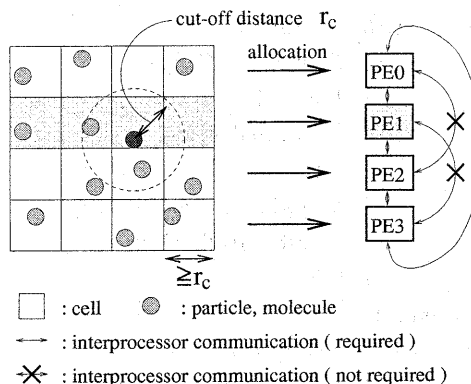


図 2: ドメイン分割法の概要

2.3 セルベース実装とハイブリッド実装

高性能計算の分野では良く知られているように、計算手法だけでなく、実装法によっても実行時間は大きく変化する。本稿では 2 つの実装法によるプログラムで実行時間を比較した。1 つはセルベース実装であり、もう 1 つはハイブリッド実装である。2 つの実装法の主な違いはループの構成である。分子動力学法プログラムのうち、実行時間の多くを占める中核部分は、各原子が相互作用する相手の原子を確定する処理である。2 つの実装法を用いたこの部分のループの構成を図 3 に示す。なお、図中の「 $makelist(i, j)$ 」は、 i 番目の原子と j 番目の原子の距離を計算し、相互作用影響範囲内にある場合にはそれぞれの原子の相互作用計算相手リストに登録するサブルーチンである。

図 3(a) はセルベース実装でのループ構成である。1 行目のループの「 $Cell-in-localPE$ 」は PE が担当するセルの数であり、セルベース実装では、セルの数だけ繰り返すループの中にセル中の原子の数だけ繰り返すループがある。このとき希薄系では、多数回まわるセルベースのループの中に少数の原子の数だけまわるループがあるため、アンローリングなどの高速化手法を利用しにくい。このような場合は、セルサイズを大きく設定することで、実行時間を短縮できることが経験的に知られている。しかし、セルサイズを大きく設定すると、密度の大きい場合は、相互作用影響範囲の外にある多数の原子につい

```

do i=1, Cell-in-localPE
  do j=1, Particle-in-Cell(i)
    do k=1, j-1
      call makelist(j, k)
    end do
    do k=j+1, Particle-in-Cell(i)
      call makelist(j, k)
    end do
  end do
do j=1, 26neighbor(i)
  do k=1, Particle-in-Cell(i)
    do l=1, Particle-in-Cell(j)
      call makelist(k, l)
    end do
  end do
end do
end do

```

(a) セルベース実装

```

do i=1, Particle-in-LocalPE
  do j=1, Particle-in-Cell(CellNo(i))
    call makelist(i, j)
  end do
  do j=1, 26neighbor(CellNo(i))
    do k=1, Particle-in-Cell(j)
      call makelist(i, k)
    end do
  end do
end do
end do

```

(b) ハイブリッド実装

図 3: ドメイン分割法の 2 つの実装法の違い

でも *makelist(i, j)* を実行する必要があることによるオーバーヘッドが生じる。

一方ハイブリッド実装では、原子の数だけ繰り返すループを一番外側に用いる。原子の数だけ繰り返すループでセルベース実装と同等の処理を行なうために、ハイブリッド実装では、原子と原子が所属するセル番号とを対応する対応表を導入する。つまり、図 3(b)2 行目の「*CellNo(i)*」は、*i* 番目の原子が所属するセル番号である。この実装法では、セルの数が陽に繰り返し数に現れないので、希薄系ではセル数だけ繰り返すループによるオーバーヘッドが小さくなることが期待される。

3 性能評価

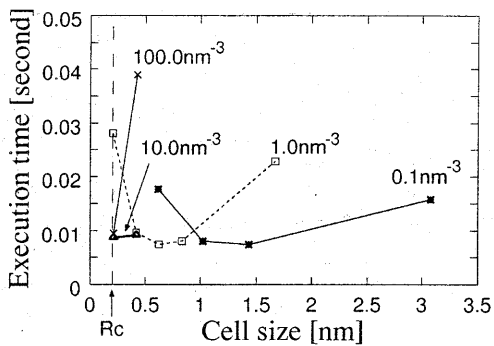
3.1 計算機環境

性能評価のために、1 ノードあたり 1PE を持つ 128 ノードの分散メモリ型並列計算機である T3E-1200 を使用した。T3E は CPU に 600MHz の DECchip 21164a を使用している。結合網は

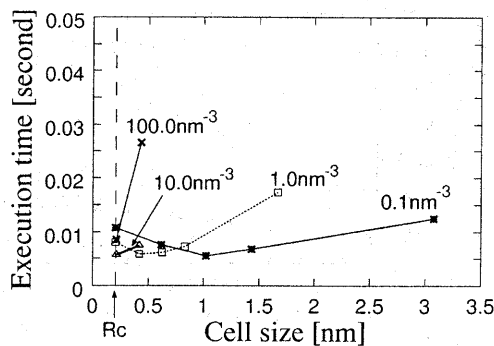
3 次元トラス型であり、通信性能はノードあたり 3.9GB/second である。MD シミュレーションの実装には、プログラミング言語 FORTRAN90 にメッセージ通信ライブラリ MPI (Message-Passing Interface) を併用し、SPMD 型処理を行った。本稿で用いる実行時間は、MPI の *MPLWTIME* 関数を使用して計測した。

3.2 2 つの実装法による実行時間

まず、密度およびセルの大きさと実行時間の関係を調べる。均一な状態での関係を調べるため、シミュレーション空間全体の平均密度を変化させてプログラム実行し、初期状態付近での実行時間を計測した。さらに、セルの大きさと実行時間の関係も調べた。それらの結果を図 4 に示す。図 4(a) によると、セルベース実装では、同じ密度であってもセルの大きさによって実行時間が大きく変化することがわかる。そのため、密度が局所的に変化する状態では、一様な大きさのセルではオーバーヘッドが生じることが予想される。一方図 4(b) によると、ハイブリッド



(a) セルベース実装



(b) ハイブリッド実装

図 4: 密度と実行時間の関係 (T3E の 1PE 使用, 1000 原子, 3000K)

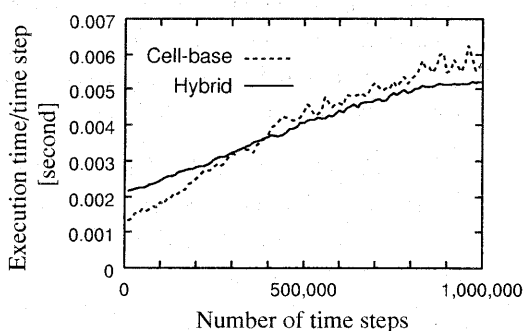


図 5: クラスタ生成シミュレーションの実行時間変化 (T3E の 1PE 使用, 216 原子, $\rho = 0.1\text{nm}^{-3}$, 3000K)

実装では、セルの大きさによって実行時間は変化するが、セルベース実装よりも変化が小さいことがわかる。セルベース実装とハイブリッド実装のどちらも、各密度で最適な大きさのセルを使用すると、実行時間は同程度であることがわかる。セルの大きさを相互作用影響範囲の r_c にした場合は、セルベース実装では実行時間が密度によって大きく変化するが、ハイブリッド実装では密度による違いが小さいことが予想される。

次に、クラスタを生成するシミュレーションを実行したときの、実行時間変化を図 5 に示す。ここで、セルベース実装ではセルサイズを図

4(a) 中での最適値に近い $7r_c$ とし、ハイブリッド実装ではセルサイズを r_c にした。図 5 によると、初期状態付近では、セルベース実装のほうが高速であるが、時間発展とともにハイブリッド実装の実行時間増加が小さいために、300,000 タイムステップ付近で実行時間の逆転が起こる。そしてシミュレーションの最終段階ではハイブリッド実装のほうが高速であった。

これらの実行時間の内訳を図 6 に示す。図中の量は以下の通りである。

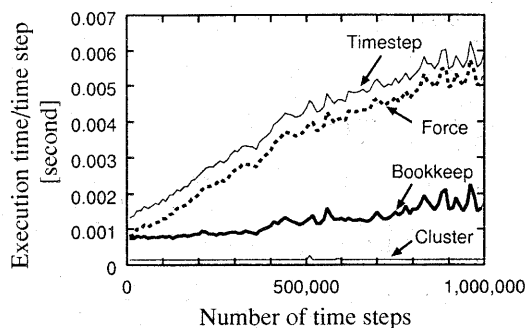
TimeStep 1 タイムステップの実行時間

Force 相互作用の計算時間 (相互作用計算相手を決定する処理を含む)

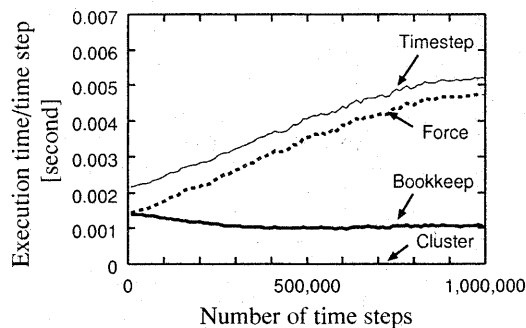
Bookkeep 相互作用計算相手を決定する処理時間

Cluster 原子の結合関係を調べてクラスタかどうかの判定をし、温度制御をする処理時間

図 6 によると、セルベース実装とハイブリッド実装の主な違いは Bookkeep の処理時間であり、Bookkeep の処理時間差が 2 つの実行時間の差をつくっている。図 6(a) を見ると、セルベース実装では最初は Bookkeep の時間が小さく、時間発展とともに増加する。一方図 6(b) を見ると、ハイブリッド実装では Bookkeep の時間が時間発展とともに減少する。そのため、当初はセルベース実装のほうが高速であるが、シミュレー



(a) セルベース実装



(b) ハイブリッド実装

図 6: クラスタ生成シミュレーションの実行時間内訳
(T3E の 1PE 使用, 216 原子, $\rho = 0.1 \text{ nm}^{-3}$, 3000K)

シミュレーションの途中でハイブリッド実装の実行時間の方が小さくなるのがわかる。

4 まとめ

本稿では、希薄かつ不均一な系におけるドメイン分割法を用いた MD シミュレーションの 2 つの実装法、セルベース実装とハイブリッド実装を比較した結果を報告した。本稿の物理状態では初期状態は均一であるが、時間発展とともに原子が結合してクラスタをつくるため、部分的な密度変化が生じ、最適なセルの大きさが変化することが実行時間を増大させることがわかった。部分的な密度変化は、並列計算機上では計算負荷の平均化問題を同時に起こす。今後はハイブリッド実装を組み合わせた動的負荷分散法の実装を検討する予定である。

謝辞

本研究の一部は、科学研究費補助金(若手研究 B 課題番号 13780215)を用いて行なわれた。関係各位に感謝する。

参考文献

[1] D. W. Heermann. シミュレーション物理学. シュプリンガー・フェアラーク東京, 1990.
[2] D. M. Beazley, P. S. Lomdahl, N. Grønbech-Jensen, R. Giles, and P. Tamayo.

Parallel Algorithms for Short-range Molecular Dynamics, volume 3 of *Annual Reviews in Computational Physics*, pages 119–175. World Scientific, 1995.

[3] 松原 正純, 板倉 憲一, 朴 泰佑. 超並列計算機 cp-pacs における大規模分子動力学法シミュレーション. 情報処理学会論文誌, 40(5):2172–2182, 1999.
[4] J. Tersoff. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Phys. Rev. B*, **39**(8):5566–5568, 1989.
[5] 山口 康隆 丸山 茂夫. フラーレン生成過程の分子動力学 (第 1 報, ケー ジ構造の形成と制御温度). 日本機化学会論文集 (B 編), 63(611):2398–2404, 1997.
[6] 堀口 進 林 亮子, 樋渡 保秋. カーボンクラスタ生成並列分子動力学法シミュレーションの高速化手法. 情報処理学会研究報告, 2000(2000-HPC-84-6):31–36, 2000.
[7] R. Hayashi, Y. Hiwatari and S. Horiguchi. Parallelization of a Molecular Dynamics Simulation for Carbon Cluster Formation Process. In *Proceedings of HPC Asia 2001*, 2001.