

# 大規模並列化量子モンテカルロ法の開発と

## 冷却ボーズ原子系のシミュレーション

正木 晶子

東京大学物性研究所

本稿では筆者らが開発した大規模数値計算のための量子モンテカルロ法の新しいアルゴリズムである“並列化マルチワームアルゴリズム”の概要と、FX10を用いて行ったアルゴリズムの性能評価の結果について報告する。

### 1. はじめに

量子モンテカルロ法には様々なアルゴリズムが存在するが、本研究で用いるファインマンの経路積分表示に基づいた世界線モンテカルロ法<sup>1</sup>は、有限温度下において、与えられた初期条件の影響を受けることなく、統計誤差の範囲内で厳密な計算が可能な優れた手法の一つである。また、汎用性も高く、負符号問題<sup>2</sup>がない範囲であれば様々な物理系に適用可能である。この手法は原子核物理の分野におけるハドロン模型や、物性物理における強相関模型など、幅広い分野で用いられているため、読者も一度は耳にしたことがあるかもしれない。ここ20年ほどで量子モンテカルロ法は大きく発展をとげ、様々な分野の量子系の理論的研究に大きな功績を残してきた。中でも世界線モンテカルロ法のための**ワームアルゴリズム**<sup>3</sup>は最も汎用性が高く効率的なものとして知られている。

しかし、現在残されている重要な問題には、既存の量子モンテカルロ法のアルゴリズムを用いてもなお到達できない様な、大規模な系での計算を必要とする問題が多く存在する。例えば対称性の破れを伴うような相転移の臨界現象やそれに類似する弱い一時転移である。系を構成する粒子やスピンの互いに影響を及ぼし合っている範囲（相関長）が転移点に近づくにつれ増大するが、これを覆う大きさの系で計算しなければ、相転移の性質を正確に知ることはできない。また、液体ヘリウム4では超固体の存在の有無が古くから実験的にも理論的にも研究対象とされてきたが、量子モンテカルロ法でも多くの先行研究が存在するにもかかわらず、未だ説得力ある決定的な答えは示されていない。量子モンテカルロ法での第一原理的な実験の予測が重要視されているが、実験室系を再現できるような規模での精度の良い計算は難度の高い課題となっている。

このような大規模系の問題解決の糸口として考えられるのが、アルゴリズムの並列化である。TOP500を見てわかるように、近年のハイパフォーマンス・コンピューティングの傾向は、多量のコアを搭載したスカラ計算機を用いた大規模並列計算である。もはやアルゴリズムの並列化は、数値計算をする者にとっては無視できない課題となっている。

<sup>1</sup> M. Suzuki, Prog. Theor. Phys. 56, 1454 (1976); M. Suzuki, S. Miyashita, and A. Kuroda,

<sup>2</sup> 後述するような世界線の更新確率に負の符号が生じる問題。

<sup>3</sup> N. Prokof'ev, B. Svistunov, and I. Tupitsyn, Sov. Phys. JETP **87**, 310 (1998)など。

残念なことに、ワームアルゴリズムは後述する理由から、並列化アルゴリズムは非自明であり実現されてこなかった。そこで我々は大規模並列計算が可能なワームアルゴリズムの開発に取り組んだ。完成したアルゴリズムを用いて、FX10 上でフラット MPI による 3200 コア並列のベンチマーク計算を行ったものを本稿にて紹介する。

## 2. 量子モンテカルロ法と並列化マルチワームアルゴリズム

本章では、格子模型のための世界線量子モンテカルロ法について説明する。我々が開発した並列化マルチワームアルゴリズムの概要と並列化の方法について紹介するために、まずは最初の節で、量子モンテカルロ法の基本的な考え方やアルゴリズムについて簡単に解説する。

### 2-1. 世界線量子モンテカルロ法

典型的な世界線モンテカルロ法では、ファインマンの経路積分に基づいて描かれた  $d+1$  次元空間の世界線配置をサンプリングする。具体的には、逆温度  $\beta$  を  $N_\tau$  分割して  $\Delta\tau = \beta/N_\tau$  とし、鈴木-トロッター分解を用いて、分配関数を、

$$\begin{aligned} Z &= \text{Tr} e^{-\beta H} = \sum_i \langle i | e^{-\beta H} | i \rangle \\ &= \lim_{N_\tau \rightarrow \infty} \prod_{k=1}^{N_\tau} \sum_{\{i_k\}} \langle i_k | e^{-\Delta\tau H} | i_{k+1} \rangle \end{aligned}$$

と経路積分表示する。ただし  $H$  は系のハミルトニアンを、 $|i_k\rangle$  は完全直交系をそれぞれ表している。一行目から二行目の過程で、 $O(\Delta\tau^2)$  以上の項を無視し、また、 $N_\tau$  個の完全系を挿入した。しかし、このままでは各局所状態  $\langle i_k | e^{-\Delta\tau H} | i_{k+1} \rangle$  において全ての相互作用を考慮する必要がある。

ここではハミルトニアンを、 $H = \sum_{b=1}^{N_b} H_b$  と局所的なハミルトニアンの和で書ける場合限定して考える。例えば  $b$  は相互作用のあるボンドやサイトを表しているが、本稿では  $b$  はボンドを表すこととする。すると、

$$\begin{aligned} Z &= \sum_{\{i_k\}} \prod_{k=1}^{N_\tau} \prod_{b=1}^{N_b} \langle i_{k,b} | e^{-\Delta\tau H_b} | i_{k+1,b} \rangle \\ &= \sum_{\{i_k\}} W(\{i_k\}) \end{aligned} \quad (1)$$

となり、各局所状態を 2 体問題として扱うことが可能となる。式 (1) の  $W(\dots)$  の部分をグラフィカルに表現すると、図 1 の様になる。黒の太線が状態  $|i_k\rangle$  の虚時間発展を追った世界線である。式 (1) の和を実行するには、すべての世界線配置を再現すればよいが、そのまま計算するには膨大な状態を発生させなければならなくなる。そこで実際には、重み付きサンプリング (重要な寄与を持つ状態のみを発生させてサンプリングを行う) の精神に則り、エルゴード性と詳細釣り合いを満たすよう定義された確率過程のもとモンテカルロ更新を行うことで配置

をサンプリングする。シミュレーションの過程で、各配位空間での物理量をサンプルすることで、統計誤差の範囲内で厳密に物理量期待値を計算する。ただし、図 1 では、 $N_{\tau}=4$  の場合を図示したが、実際の計算では  $N_{\tau} \rightarrow \infty$  の極限をとっている。

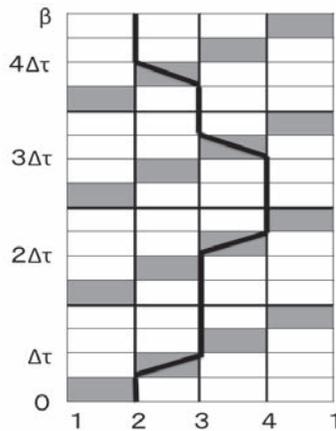


図 1：鈴木-トロッター分解による世界線の配位空間の図。

分かりやすいように実空間次元が 1 次元の場合を示した。縦軸は虚時間で 0 と  $\beta$  で周期境界条件になっている。横軸は実空間軸であり、こちらも周期境界条件が課されている。世界線はグレーのボックスでのみ斜めに進める。

## 2-2. ワームアルゴリズムと並列化の困難

ワームアルゴリズムは、世界線モンテカルロ法のための状態更新アルゴリズムである。ワームアルゴリズムでは  $\Delta \tau \rightarrow 0$  の連続虚時間極限で配位空間を表現できる。配位空間の更新は、**ワーム**と呼ばれる非対角演算子の対を挿入した、「拡張配位空間」中で行われる。具体的にどのような演算子なのかは扱う基底によって異なるが、世界線の不連続点を生成するように定義される。一対のうちの片方のワームが釣り合い条件を満たす確率過程のもと配位空間中を動きながら世界線配置を更新していき、自身の片割れと再び出会ったときに切れていた世界線がつながり、新しい世界線配置が完成する仕組みとなっている (図 2)。この演算子の対はグリーン関数に相当するものであり、よって、ワーム間には相関があり、この伸びた相関の範囲内での大域的な更新を行うのがワームアルゴリズムの特徴である。

ワームアルゴリズムは、汎用性も高く最も優れたアルゴリズムの一つであるが、アルゴリズムの並列化は困難な仕様となっており、これまで成し遂げられてこなかった。一つ注意しておきたいことは、ここで言う並列化とは、統計精度を上げるための単純な乱数並列計算 (自明な並列化) のことではなく、1つの状態空間に対して施す**非自明な並列計算**のことである。我々は系をドメイン分割し、各コアに割り当てることを考える。

これまでワームアルゴリズムの並列化が成し遂げられなかったのは、状態更新を行うワームが空間中の一点にのみ挿入されているためであると考えられる。このため、仮に配位空間をドメイン分割しても、ワームが挿入されているドメイン以外は更新されず並列化の意味をなさなくなる。

それでは、ドメインごとにワームを配置できれば解決するのか、と言えば全くそうではない。

従来のアルゴリズムと同様な仕様では、ワームがすべて消えるまで待つことになり、並列化効率が実行途中から著しく低下し、やはり並列化の意味をなさなくなる。しかし、それ以前に、アルゴリズムの根本にあるイベント駆動性が悪さをする。すなわち、ワームがドメイン境界に達するたびに通信を行えば、かなり不規則かつ頻繁に通信が発生してしまい、非常に非効率的である。そもそも、別の領域で発生する予定の更新の終状態を予測することは、ワームの動きをトレースしない限り困難であるため、移動するタイミングや移動のための更新確率を決めるのも困難である。

並列化の際には、これらの問題を解決するようにアルゴリズムを設計することはもちろんのこと、重み付きサンプリングが正しく実行されるように注意する必要もある。

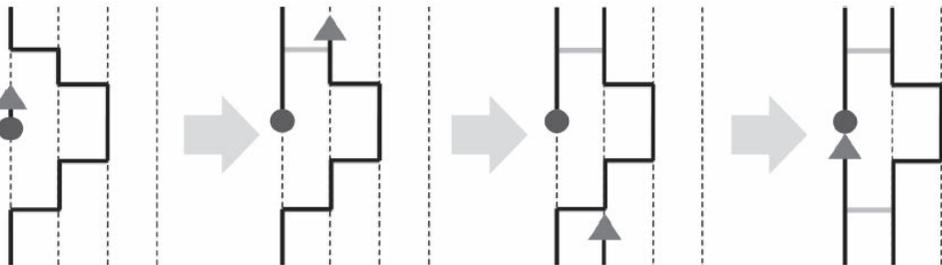


図 2：一対のワームが世界線の配置を更新する過程の模式図。

太線が世界線。三角と丸がワーム対であり、シミュレーション上では三角が動き回り（ワームのあたまと呼ばれる）、丸が止まっている（ワームのしっぽと呼ばれる）とする。最左図でワームが挿入され、中 2 つの図であたまが移動しながら状態を更新する。最右図でしっぽと出会い対消滅する。

### 2-3. 並列化マルチワームアルゴリズムの概要

我々が開発した並列化マルチワームアルゴリズムの特徴は、複数のワームを配位空間中に挿入でき、ドメイン分割による並列計算が可能であるところである。図 3-1 に模式図を示す。複数ワームの導入のために次のような、求めたい系のハミルトニアン  $H$  にソース場  $Q$  を付加した拡張ハミルトニアン  $H_\eta$  を定義する。

$$H_\eta = H + \eta Q$$

$\eta$  は任意のパラメータで、これを操作することでワームの数を調節できる。従来のアルゴリズムとの大きな違いは、ワームを対消滅させず、 $\eta \rightarrow 0$  への外挿によって求めたいハミルトニアンでの物理量を求める点である。

更新過程については、ワームの挿入方法が新しく、隣接する世界線同士の相互作用項（**パーテックス** = 図 3-1 の説明文参照）で区切られたセグメントごとに「**ワームの重み**」を定義し（図 3-2 の説明文参照）、セグメントごとに独立に挿入されるワーム数とその位置が決定される。この「ワームの重み」の概念は、ドメイン並列の際に威力を発揮する。各ドメインはフラット MPI で各コアに割り当てられ、ワームのドメイン間移動は、ワームの重みを MPI 通信によりやり取りすることで間接的に実行できる。これは 2-2 節で挙げた問題点を克服できている。これについては次の節で詳しく説明する。もしワームのドメイン間移動が行われないと、ドメイン境界上の状態が更新されなくなる（エルゴード性が破れる）。また、正しく移

動できてないと、ワーム間の相関が一つのドメイン内にとどまってしまう、本来の系の大き  
さで計算ができなくなってしまうので注意が必要である。

また、ここでは詳しく述べないが、複数のワームを挿入した場合、ワームが従来のアルゴ  
リズムと同じ動きをするのでは詳細釣り合いが満足されなくなる。我々のアルゴリズムでは、こ  
の点も克服するようなワームの動きを考えた。

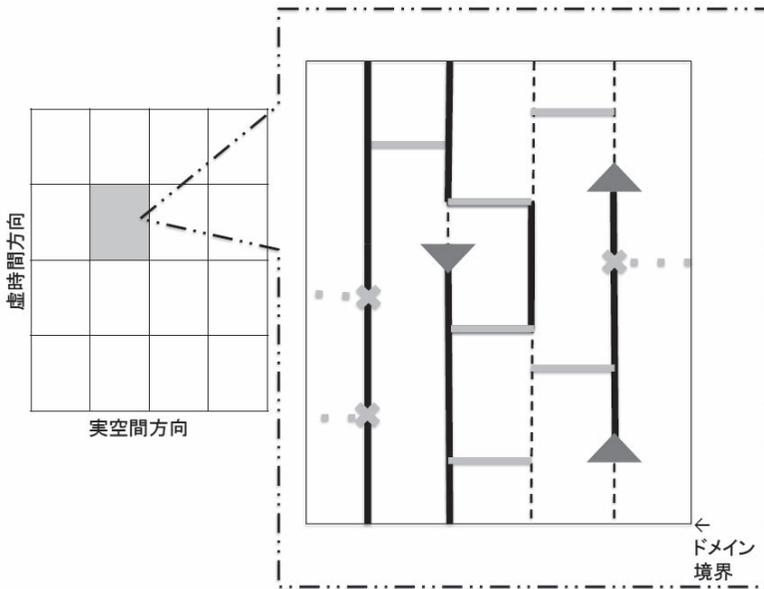


図 3-1 : 並列化マルチワームアルゴリズムの配位空間とドメイン分割の模式図。

左の図は配位空間をドメイン分割したもの。右の図（吹き出し内）は一つのドメインに着目して詳しく  
示したもので、横線はバーテックス、三角はワームを表しており、×は隣のドメインに所属する格子と  
の相互作用のバーテックスを表している。

#### 2-4. 並列化マルチワームアルゴリズムでの MPI 計算

ワームのドメイン間移動は、ワームが境界に達した時に行うのではなく、一定プロセスごと  
にワームの重みをドメイン間通信することで実現されるため、通信コストはそれほど大きくな  
らない。図 3-2 はワームの重みのドメイン間通信を模式的に示した図である。

ドメイン境界の状態とワームを更新するには、最初に図 3-2 の左図の矢印の方向へ  
MPI\_Sendrecv を実行し、Receive した情報からドメイン境界を更新し、先ほどの矢印と逆方向  
へ MPI\_Sendrecv を再度実行することで更新された状態を隣のドメインと共有する。

ドメイン表面に隣接するセグメントの情報とドメインをまたぐ相互作用の情報のみ通信すれ  
ばよいため、比較的少ない通信量で済む。我々のコードは C++ で設計されており、セグメント  
の情報とは、そのセグメントに所属するバーテックスのクラスの型とワームの重みである  
double 型の情報のことである。バーテックスクラスは、クラスの型のまま送信するのではなく、  
必要なメンバ変数だけをデータ型ごとに配列に格納し直し MPI\_Sendrecv を行う方が、全体の計  
算時間が速くなった。

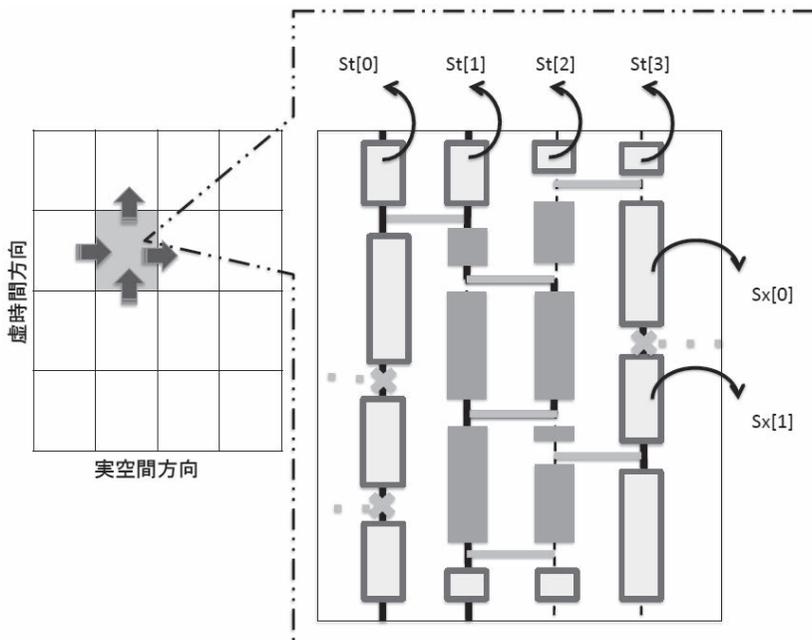


図 3-2 : 図 3-1 の配位空間の並列計算における通信の模式図。

左の図で、出て行く矢印の先に send し、入ってくる矢印のもとから receive するように MPI\_Sendrecv を実行する。矢印の右の図（吹き出し）は一つのドメインに着目して詳しく示したもので、四角は横線で区切られたセグメント上に割り当てられたワームの重みを表している。白抜きの四角は通信が必要な箇所、塗りつぶしの四角はそうでないものである。矢印の配列に情報を格納して 1 回目の MPI\_Sendrecv する。矢印がついてない白抜きの四角は、2 回目の MPI\_Sendrecv で通信を行う。

### 3. 冷却ポーズ原子系のためのベンチマーク計算

#### 3-1. ハードコア・ポーズ粒子模型でのポーズ・アインシュタイン凝縮秩序変数

ベンチマーク計算では、東大情報基盤センター-FX10 を用いて、次のような 2 重占有を禁止した 2 次元正方格子上的ハードコア・ポーズ粒子模型の計算を行った。

$$H = \sum_{(i,j)} \left[ t(b_i^\dagger b_j + b_i b_j^\dagger) + V b_i^\dagger b_i b_j^\dagger b_j + \frac{\mu}{2} \sum_i (b_i^\dagger b_i + b_i b_i^\dagger) \right]$$

ここで、 $b_i^\dagger(b_i)$  は格子点  $i$  でのハードコア・ポーズ粒子の生成（消滅）演算子である。 $(i,j)$  は隣接サイトの対（ボンド）を意味する。 $t$  はホッピング、 $V$  は隣接粒子間相互作用、 $\mu$  は化学ポテンシャルである。このモデルでもワームのソース場は

$$Q = \sum_i (b_i^\dagger + b_i)$$

である。

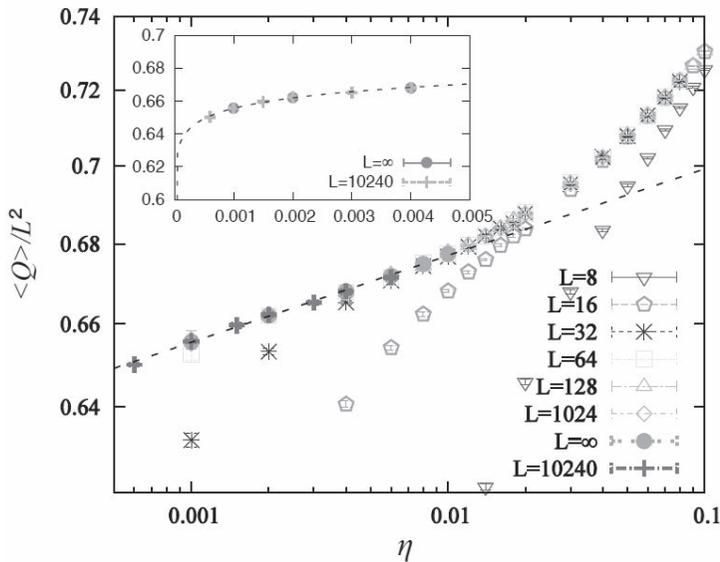


図 4 : ボーズ・アインシュタイン凝縮秩序変数の  $\eta$  依存性。

パラメータは  $\beta=16$ ,  $V=3.0$ ,  $\mu=1.2$  の超流動相を選んだ。  $L=8 \sim 10240$  のサイズでの値をそれぞれ log-log プロットした。ただ挿入は線形プロットである。  $L=\infty$  の外挿値は  $L=8 \sim 1024$  の値から導いた。点線は  $L=\infty$  をべきでフィッティングしたものである。参考文献にある筆者らの論文参照。

このモデルでは低温で、超流動相<sup>4</sup>が安定化されるが、この相内で有限温度下でのボーズ・アインシュタイン凝縮秩序変数  $\langle Q \rangle / L^2$  の  $\eta$  依存性を調べたのが図 4 である。なおすべての計算結果では、ホッピングを  $t=1.0$  にとっている。

各点は  $\eta$  についての外挿から求めたものである。系の一方向の格子点数を  $L$  とした。我々は 3200 のドメイン分割及び MPI 並列により、従来のアルゴリズムでは到達不可能なサイズである  $10240 \times 10240$  格子点の計算を達成した。結果を無限系 ( $L$  による外挿値) と比較したところ、誤差の範囲内で一致することが確認できた。これは熱力学極限に対応するような系のサイズを直接計算できたと言える。  $\langle Q \rangle$  は、従来のアルゴリズムでは直接計算できないものであり、2次元系で絶対零度では  $\eta$  に関して定数となる。また、有限温度相転移がおこる臨界領域では  $\langle Q \rangle$  の  $\eta$  に関するべきの指数から相転移現象の臨界指数を知ることができる。図 4 はこのどちらでもなく、  $\eta$  はべき的振る舞いをするが、  $\eta \sim 0.01$  と非常に小さい指数をとる結果となった。

### 3-2. 並列化による緩和効率の影響

アルゴリズムの緩和効率は非常に重要である。いくら大きな計算ができて、緩和に途方もない時間がかかるようでは意味がない。前節で述べたように、このアルゴリズムでは、並列化効率を上げる目的から、ドメイン間通信の回数を制限している。通信がない間、ワームは境界にはじかれて動き続けるため、ドメイン境界が増えるにしたがってその影響は増すことになる。もちろん通信時には境界は更新されるためエルゴード性と詳細釣り合いは満たしているが、更

<sup>4</sup> 有限温度の 2 次元系では真の意味での超流動長距離秩序は存在しないため、正確に言うと、準長距離秩序を持つコスタリッツ-サウレス相である。そのため、図 4 で示した秩序変数が  $\eta \rightarrow 0$  でべき減衰している。

新効率に影響を及ぼすことが懸念される。この様子を Strong scaling で確認したのが図5である。まずサンプル数を固定し、標準誤差の並列数  $N$  に対する変化を調べた (図5-1)。結果、標準誤差は分割数に応じて大きくなっているが、それは非常に緩やか ( $\sim N^{0.09}$ ) であるため、緩和に大きな影響はないことが分かった。 $N$  ごとに適当な虚時間方向の分割数  $N_\beta$  と実空間方向  $N_r$  の分割数を割り当て、 $N=N_\beta \times N_r$  となるようにしている。このアルゴリズムでは虚時間方向と実空間方向とはドメイン境界の更新方法が、本質的には同じであるが、詳細は異なるため、結果の図では若干のがたつきが生じているが、誤差の範囲内である。

さらに、計算のウォール・クロック時間を固定したときの標準誤差の  $N$  依存性も調べ、結果を従来のワームアルゴリズムの一種である向き付きループアルゴリズム (DLA)<sup>5</sup> と比較した (図5-2)。我々のアルゴリズムは DLA の考え方をベースに取り入れているため比較対象として選んだ。これにより、 $N$  が8以上で DLA の精度を上回ることを確認できた。また、標準誤差の大きさの変化は実質 100 パーセントの効率で小さくなっていくのが確認できた。これは Strong scaling が非常に効率よく働いているためであり、並列数を稼ぐことにより、従来のアルゴリズムを超える精度の向上を期待できる。

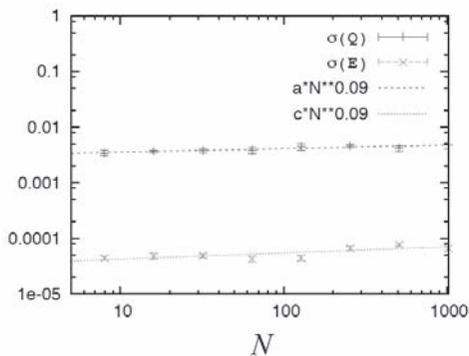


図5-1：サンプル数を固定したときの標準誤差  $\sigma$  の分割数 (並列数) 依存性。

$L=128$ ,  $\beta=128$ ,  $V=3.0$ ,  $\mu=1.2$ ,  $\eta=0.002$  とした。 $\langle Q \rangle / L^2$  とエネルギーの標準誤差で、統計平均を求める際のサンプル数 10000 に固定した。参考文献にある筆者らの論文参照。

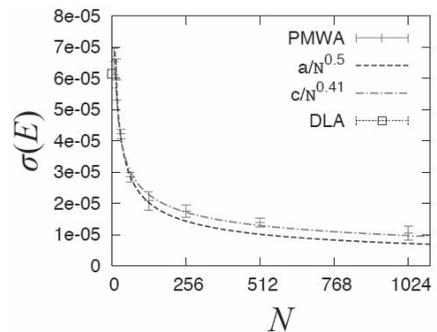


図5-2：実時間を固定したときの標準誤差  $\sigma$  の分割数 (並列数) 依存性。

$L=256$ ,  $\beta=16$ ,  $V=3.0$ ,  $\mu=1.2$ ,  $\eta=0.004$  とした。観測実時間を 10000 秒に固定したときのエネルギーの標準誤差で、向き付きループアルゴリズム (DLA) の結果と比較している。

## 4. 性能評価

### 4-1. Weak scaling による並列化効率

Weak scaling による並列化効率についても検証した。加速効率を図6に示す。Weak scaling の加速効率を以下のように定義した。

$$A(N_{\text{node}}) = \frac{T_1}{T_{N_{\text{node}}}} \times N_{\text{node}}$$

ここで  $N_{\text{node}}$  はノード数である。 $T_{N_{\text{node}}}$  は計算に  $N_{\text{node}}$  用いた時のウォール・クロック時間である。1 ノード用いたときのウォール・クロック時間を基準に取っている。 $\mu$  は超流動相と固体相を

<sup>5</sup> O. F. Syljuasen and A. W. Sandvik, Phys. Rev. E **66**, 046701 (2002).

とる値をそれぞれ用いた。超流動相と固体相の結果はほぼ一致しており、特に違いは見られなかった。2つの相内ではワームの数が根本的に異なるが、本アルゴリズムでは、ワームの数を積分した重みを用いて通信を行うため、並列化効率には影響がなかったものと推測できる。肝心のコア数依存性であるが、1024 コア (16 ノード) までほぼ100パーセントの効率を実現し、その後徐々に衰退している様子が見られたが、約3パーセント程度の減速にとどまった。今回調べた214ノード並列の範囲では並列化効率の飽和の振る舞いは見られなかった。

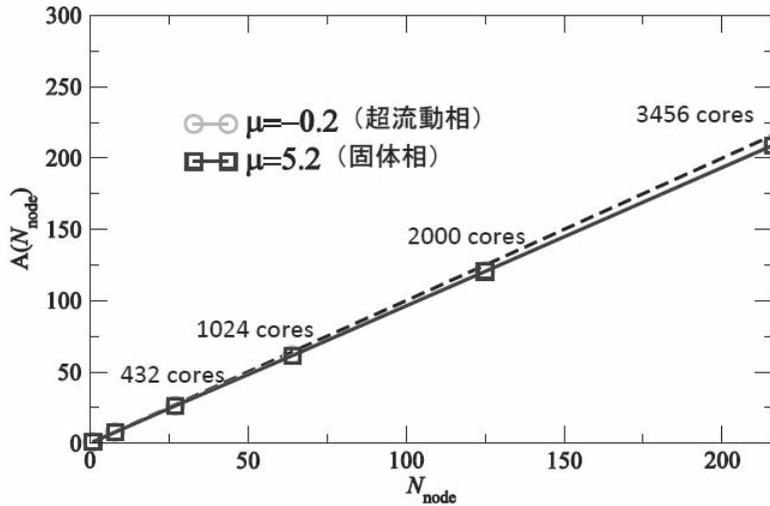


図6 : Weak scaling での加速効率。

丸は超流動相で  $\mu = -0.2$ 、四角は固体相で  $\mu = 5.2$ 。ドメインのサイズは実空間方向  $L_0=16$ 、虚時間方向  $\beta_0=8$  とし、 $V=3.0$ 、 $\eta=0.006$  とした。黒の点線は100パーセントの効率のときの加速効率である。

#### 4-2. プロファイリング結果

最後に、fapp によるプロファイリングを行ったので、結果を図7に示す。パラメータは  $L=256$ 、 $\beta=16$ 、 $\mu=1.2$ 、 $V=3.0$ 、 $\eta=0.008$  とし、Strong scaling による計算を行った。Performance monitor で命令処理回数 (MIPS) に対するピーク性能を確認したところ、約12~18パーセントあり、並列数を上げるにつれてピーク性能も高くなると分かり、比較的良好な結果であることが分かった。

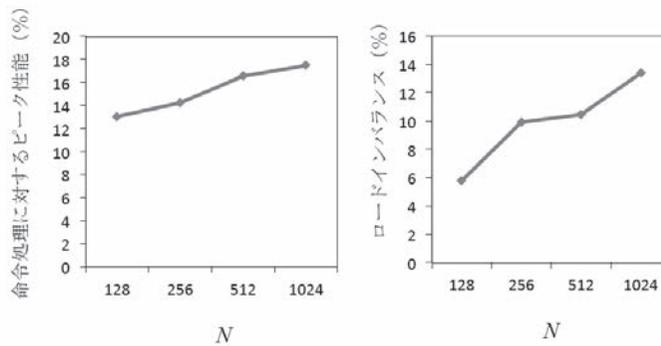


図7 : 命令処理(MIPS)に対するピーク性能とロードインバランスの分割数(並列数)依存性。

一方、ロードバランスは並列数を増やすに従いインバランスが大きくなっていることが分かった。これはアルゴリズムの性質上、Strong scaling で同じ  $\eta$  を用いると、系全体のワームの数は同程度でも、ドメインあたりのワーム数は減っていき、ドメインごとのワームの数にばらつきが目立つようになってくる。そうすると待ちのノードも増えることになり、インバランスが増すと推測される。これを解決するにはプログラムのチューニングよりも、 $\eta$  のチューニング（ワーム数の調整）が必要になると言える。ワームの数が一桁にならず、相が変化しない様な  $\eta$  を選ぶことが好ましいだろう。

## 5. まとめと今後の展望

複数ワームの導入によって、並列化可能なワームアルゴリズムを開発し、熱力学極限に対応する規模のハードコア・ボーズ粒子系のシミュレーションに成功した。今回のベンチマーク計算では単純な模型を用いたが、このアルゴリズムは原理的には負符号問題が出ない範囲において一般的なスピン模型やソフトコア・ボーズ粒子模型に適応することが可能である。我々のアルゴリズムは Weak scaling, Strong scaling とともに非常に高い並列化効率を持つが、物理量測定のプロセス中で統計平均をとる際などで、特定のランクのみで処理を行う箇所が存在しているため、そこでは Open MP とのハイブリッド並列を取り入れるなどすれば更なる効率向上が期待できる。

## 謝辞

若手・女性利用支援推薦制度を通じてお世話になりました、東京大学情報基盤センターの関係者の皆様に深く感謝致します。また、本研究の共同研究者ならびに計算物質科学イニシアティブ第一部会・川島・藤堂グループの皆様には、日頃から有益な議論とご教授を頂きましたこと、感謝申し上げます。本研究は、文部科学省の HPCI 戦略プログラム、および、計算物質科学イニシアティブの助成を受けており、計算の一部は、計算物質科学イニシアティブが確保した FX10 の利用枠を使用しています。また、計算物質科学イニシアティブのアプリ高速化支援サービスを通じてお世話になりました富士通の石附さんに感謝申し上げます。

## 参考文献

- N. Kawashima and K. Harada, J. Phys. Soc. Jpn. 73, 1379(2004).  
A. Masaki-Kato, T. Suzuki, K. Harada, S. Todo, N. Kawashima, Phys. Rev. Lett., **112**, 140603 (2014).