

TSUBAME 共同利用 平成 25 年度 学術利用 成果報告書

利用課題名 TSUBAME2 GPU によるスピン系のクラスターアルゴリズム・モンテカルロシミュレーション
英文: Cluster algorithm Monte Carlo simulations of spin systems using TSUBAME2 GPU

利用課題責任者 岡部 豊
First name Surname Yutaka Okabe

所属 首都大学東京 理工学研究科
Affiliation Graduate School of Science and Engineering, Tokyo Metropolitan University

邦文抄録(300 字程度)

格子スピンモデルは、物質内の相転移現象を解析するのに使われる。我々はスピン系のクラスターアルゴリズムの並列計算を提案してきた。Swendsen-Wang マルチクラスターアルゴリズムのサンプルプログラムを Comp. Phys. Commun. Library [http://cpc.cs.qub.ac.uk/summaries/AERM_v1_0.html] に登録した。古典スピン系として、イジングモデル、 q 状態ポッツモデル、XY モデルを扱った。格子としては、2 次元正方格子と 3 次元立方格子をとりあげた。プログラムの詳細を説明すると共に、3 次元 XY モデルの計算結果を示し、相転移を論じる。さらに、複数 GPU を用いた大規模計算の改良を行った。

英文抄録(100 words 程度)

The classical spin model is one model for understanding phase transition. We have developed the parallel computation method for the cluster spin flip algorithm. We present sample CUDA programs for the GPU computing of the Swendsen-Wang multi-cluster spin flip algorithm, as a program library of Comp. Phys. Commun. Library; [http://cpc.cs.qub.ac.uk/summaries/AERM_v1_0.html]. We deal with the classical spin models; the Ising model, the q -state Potts model, and the classical XY model. As for the lattice, both the 2D (square) lattice and the 3D (simple cubic) lattice are treated. We here explain the details of sample programs. We also show the calculated results for the 3D XY model, and discuss phase transitions. Moreover, we have improved the algorithm of multi-GPU calculation for large scale cluster-flip simulations.

Keywords: Monte Carlo Simulation, classical spin model, cluster algorithm

背景と目的

多体系の統計物理学の標準的な数値シミュレーションの手法として、モンテカルロ法が広く用いられているが、シングルスピンフリップのモンテカルロ法には転移温度付近で緩和時間が急激に大きくなる問題があり、これを解消する一つの手法としてクラスターアルゴリズムが提唱されている。しかし、今まで、クラスターアルゴリズムは並列計算に適していないとされていた。本プロジェクトでは、格子スピンモデルに対し、Swendsen-Wang (SW) クラスターアルゴリズム・モンテカルロシミュレーション[1]の GPU を用いた並列計算の方法を開発し[2]、さらに複数 GPU を用いた大規模計算を実現してきた[3, 4]。25 年度は、我々のグループが提唱してきた SW クラスターアルゴリズムの GPU 計算の方法を整理し、プログラムライブラリーとして登録した[5]。

概要

ここでは、
http://cpc.cs.qub.ac.uk/summaries/AERM_v1_0.html
の Comp. Phys. Commun. Library に登録してあるプログラムについて、説明する。ライブラリー登録としては、古典スピン系のモデルとして、イジングモデル、 q 状態ポッツモデル、 q 状態クロックモデル(q が無限大の極限で XY モデルとなる)の 3 つのモデルを取り上げた。また、格子として、2 次元正方格子、3 次元立方格子の 2 つを扱ったので、計 6 つのプログラムを登録した。
すべてに共通する SW クラスターアルゴリズムの基本的なスピンの状態の更新は、3 つのプロセスから成る。すなわち、
(1) 活性ボンドの決定: 隣接するスピンの状態

に応じて、温度に依存するボルツマン因子を反映して、ボンドをつなぐかつながないかを判定する。

- (2) クラスター・ラベリング： 次に、活性ボンドのつながり具合により、スピンをクラスターに分解する。
- (3) スピン・フリップ： ラベリングのプロセスで分解されたクラスター内のスピンを一斉に更新する。

のプロセスである。なお、クロックモデルのようなベクトルスピン系では、Wolff[6]により提唱された imbedded cluster とよばれる、スピン空間で、ある軸に直交する平面に射影したときに、プラスになるかマイナスになるかで、“イジング” 的に分類する方法で、クラスター生成を行う。

クラスター・アルゴリズムを並列化するには、(2)のクラスター・ラベリングの並列化は単純ではない。通常の CPU 計算では、Hoshen-Kopelman [7] のラベリング手法が用いられるが、これは並列化できない。最近、GPU 用のクラスター・ラベリングの並列化技法が提唱されてきた。Hawick 達[8]により提唱された方法、さらにそれを改良した Kalentev 達[9]により提唱された方法を、ここでは採用する。

CUDA プログラムの概要は次の通りである。kernel 関数として、まず、スピンの初期状態を設定する

```
device_function_spinset;
```

を呼び出し、次に活性ボンドを決定する

```
device_function_init;
```

を呼び出す。クラスター・ラベリングのプロセスは

```
device_function_scanning;
```

```
device_function_analysis;
```

```
device_function_labeling;
```

により実現するが、Kalentev 達のアルゴリズムでは、初めの 2 つの kernel function だけでよい。スピン・フリップのプロセスは

```
device_function_spin_select;
```

```
device_function_spin_flip;
```

で実現する。また、物理量は、

```
device_function_sum;
```

の kernel function で計算する。

結果および考察

論文[5]では、3 次元イジングモデルと XY モデルの場合の性能評価のデータを示し、2 次元の場合と同様に、GPU 計算により高速化されること、サイズが大きくなるとクラスター・アルゴリズムの並列計算の効果が顕著になること、Hawick 達のアルゴリズムと比べて Kalentev 達のアルゴリズムが高速であることを示した。

実際の物理量の計算例として、3 次元 XY モデルのモーメント比の温度依存性を図 1 に示す。磁化の 4 次モーメントと 2 次モーメントの 2 乗の比は、Binder 比[10]として、2 次相転移の解析に広く用いられている。サイズによらない交点が 2 次相転移温度を与え（補正はあるが）、図 1 から、転移温度が $T_c=2.202$ 程度であることがわかる。なお、温度は相互作用定数 J を単位としている。

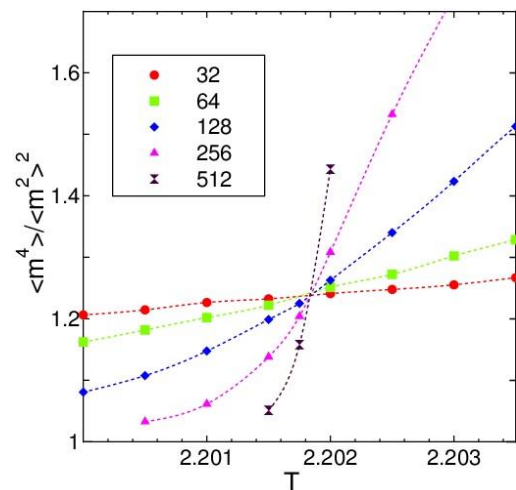


図 1: 3 次元 XY モデルの各サイズのモーメント比

さらに、有限サイズスケーリングを用いると、サイズについて相関距離に関する臨界指数 ν を用いて $L^{1/\nu}$ でスケールした変数を用いると、異なるサイズのモーメント比のデータが一つの曲線上に乗る。それを図 2 に示してある。ここで用いた、ベストフィットのデータは、臨界温度が $T_c=2.201836(6)$ 、そして、臨界指数は $\nu=0.672(4)$ である。この値は、従来報告されてきた数値と一致しているが、従来のものより、高精度である。

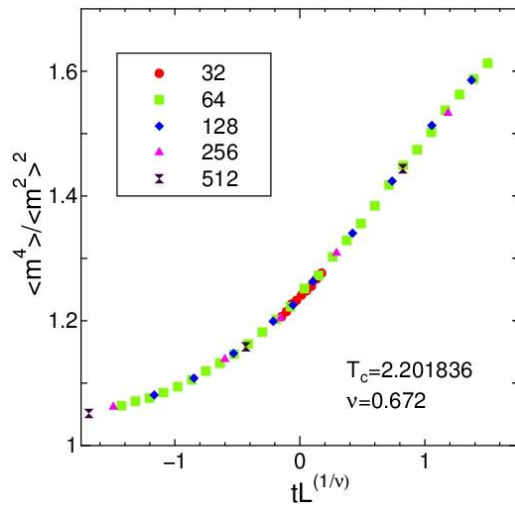


図 2: 3 次元 XY モデルのモーメント比の
有限サイズスケーリング

まとめ、今後の課題

本報告では、Comp. Phys. Commun. のライブラリー登録について記したが、本年度は複数 GPU を用いた大規模並列計算についても、改良を加えて、高速化した。

次年度、複数 GPU を用いた改良された大規模計算をさらにテストをすると共に、実際の問題へ適用をしていく。

なお、本研究は、首都大学東京客員研究員であり、原子力研究開発機構所属の小村幸浩君との共同研究である。

参考文献

- [1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987) 86-88.
- [2] Y. Komura and Y. Okabe. Comp. Phys. Comm. 183 (2012) 1155-1161.
- [3] Y. Komura and Y. Okabe. Comp. Phys. Comm. 184 (2013) 40-44.
- [4] Y. Komura and Y. Okabe. J. Phys. Soc. Jpn. 81, (2012) 113001.
- [5] Y. Komura and Y. Okabe. Comp. Phys. Comm. 185 (2014) 1038-1043.
- [6] U. Wolff, Phys. Rev. Lett. 62 (1989) 361-364.
- [7] J. Hoshen and R. Kopelman, Phys. Rev. B 14 (1976) 3438-3445.

[8] K.A. Hawick, A. Leist, and D. P. Playne, Parallel Computing 36 (2010) 655-678..

[9] O. Kalentev, A. Rai, S. Kemnitzb, and R. Schneider, J. Parallel Distrib. Comput. 71 (2011) 615-620

[10] K. Binder, Z. Phys. B: Condens. Matter 43 (1981) 119-140.