

テンソルくりこみ群を用いた相関関数の評価

著者	中本 智洋
著者別表示	Nakamoto Norihiro
雑誌名	博士論文要旨Abstract
学位授与番号	13301甲第4474号
学位名	博士（理学）
学位授与年月日	2016-09-26
URL	http://hdl.handle.net/2297/46572



学位論文要旨

テンソルくりこみ群を用いた相関関数の評価
(Computation of correlation functions
by tensor renormalization group method)

金沢大学大学院自然科学研究科数物科学専攻
中本 智洋

概要

We demonstrate how to compute the correlation function by using the tensor renormalization group for 2D Ising model. From the correlation function around the critical temperature, we extract the correlation length and the spontaneous magnetization and then determine some critical exponents. Attractive feature of our method to extract the critical exponents is that numerical derivative, that causes a loss of significant digits, and additional simulation parameters, that demand more computational cost and complicated analysis, are *not* required. On the other hand, our approach requires an additional treatment of impurity tensors, but the additional cost is still tolerable.

1 導入

テンソルくりこみ群とは 2007 年に Levin 氏と Nave 氏が三角格子上的 Ising 模型で提案した数値くりこみ群の手法である。テンソルくりこみ群の特徴は粗視化の中で特異値分解を用い、特異値の自由度の切り捨てを制御する bond dimension (D_{cut}) を設定し、くりこみを実行する手法である。このテンソルくりこみ群の長所はくりこみをくり返すごとに指数関数的に格子体積を増大させることができることである。実際に臨界点近傍外ではテンソルくりこみ群は非常に高い精度の結果を再現できることが知られている。しかし、臨界温度近傍では高い精度を維持するためにより大きな D_{cut} が要求され、計算コストや使用メモリーの増大が問題となる。また、数値微分で帯磁率を得るためには外部磁場のような追加パラメーターをハミルトニアンの中に導入する必要があり、系の解析を複雑にさせる傾向がある。これらの問題点を改善するために本論文では数値微分や追加パラメーターを必要としないで相関関数を得ることができるテンソルくりこみ群の手法の検証し、得た相関関数を用いて臨界温度と臨界指数を抽出した。テンソルくりこみ群を用いて相関関数を計算する方法は 2008 年に Gu 氏, Levin 氏, Wen 氏によって提案されたものだが、最も基礎的な 2 次元 Ising 模型が実装されていなかった。そのため、今回の相関関数から臨界温度と臨界指数を抽出することは非自明な仕事である。従って、本研究では詳細にこの模型での相関関数の計算を検証し、ベンチマークテストとして相関関数から臨界温度と臨界指数をどれぐらいの精度で再現できるかを調査する。すなわち、この方法がテンソルくりこみ群の手法の有効性を高めることができるかを実証することがこの研究の目的である。

2 テンソルくりこみ群

テンソルくりこみ群とは文字通り複数のテンソルをくりこみ、一つの新しいテンソルにする手法である。くりこみを開始する初期テンソルは分配関数に基づき書き換えられたテンソルであり、異なる模型ごとにテンソルのサイズや中身が異なる。このようにテンソルで分配関数を表現することをテンソルネットワーク表示という。このテンソルを基にテンソルの分解と自由度の縮約を行い、複数のテンソルを一つの新しいテンソルにまとめる。このテンソルの分解は特異値分解を用い、一つのテンソルを厳密にあるいは近似的に二つのテンソルに分解する。この時に生成される二つのテンソルが持つ新しい自由度を粗視化された新しいテンソルの自由度とみなし、すべての古い自由度を縮約すると粗視化された新しいテンソルが作り出される。このようにテンソルの分解と自由度の縮約を行う一連の手順をテンソルくりこみという。そして、テンソルネットワーク表示に基づきテンソルの自由度の足しあげを行うことで分配関数を求めることができる。さらくりこみの回数を増やすことによって、大きな体積の分配関数を求めることができる。

2.1 テンソルネットワーク表示

2次元 Ising model の分配関数 Z とハミルトニアン H を以下のように定義する。

$$Z = \sum_{\{s\}} e^{-\beta H} \quad (1)$$

$$H = - \sum_{\langle a,b \rangle} s_a s_b \quad (2)$$

ここで、 β は温度の逆数、 $s_a \in \{-1, +1\}$ は座標 a のスピン変数、最近接格子の組み合わせ (a, b) をすべて取っている。このスピン自由度で書かれた分配関数をテンソルネットワーク表示に書き換えることを考える。このテンソルネットワーク表示は高温展開を用い、スピンの自由度とは別の新しい自由度を生成し、その代わりにすべてのスピンの自由度を総和し、新しい自由度だけを残すことによって得られる。具体的には次のように導出される。

$$Z = \sum_{\{s\}} \prod_{\langle a,b \rangle} e^{\beta s_a s_b} \quad (3)$$

$$= \sum_{\{s\}} \prod_{\langle a,b \rangle} (\cosh(\beta) + \sinh(\beta) s_a s_b) \quad (4)$$

$$(5)$$

上式において、高温展開を用いて、スピン変数 s_a を \exp の肩から取り出している。次にスピン和を実行することができて、テンソルネットワーク表示が得られる。

$$Z = \sum_{\{s\}} \prod_{\langle a,b \rangle} e^{\beta s_a s_b} \quad (6)$$

$$= \sum_{\{s\}} \prod_{\langle a,b \rangle} \cosh(\beta) (1 + \tanh(\beta) s_a s_b) \quad (7)$$

$$= \sum_{\{s\}} \prod_{\langle a,b \rangle} \cosh(\beta) \sum_{x_{ab}=0,1} (\tanh(\beta) s_a s_b)^{x_{ab}} \quad (8)$$

$$= \cosh(\beta)^{2V} \sum_{\{s\}} \prod_{\langle a,b \rangle} \sum_{x_{ab}=0,1} (\tanh(\beta) s_a s_b)^{x_{ab}} \quad (9)$$

$$= \cosh(\beta)^{2V} \sum_{\{s\}} \sum_{\{x\}} \prod_p \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} s_p^{x_{pi}+x_{pj}+x_{pk}+x_{pl}} \quad (10)$$

$$= \cosh(\beta)^{2V} \sum_{\{x\}} \prod_p \sum_{s_p=\pm 1} \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} s_p^{x_{pi}+x_{pj}+x_{pk}+x_{pl}} \quad (11)$$

$$= \cosh(\beta)^{2V} \sum_{\{x\}} \prod_p \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} (1 + (-1)^{x_{pi}+x_{pj}+x_{pk}+x_{pl}}) \quad (12)$$

$$= 2^V \cosh(\beta)^{2V} \sum_{\{x\}} \prod_p \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} \delta_{\text{mod}(x_{pi}+x_{pj}+x_{pk}+x_{pl}, 2), 0} \quad (13)$$

$$= 2^V \cosh(\beta)^{2V} \sum_{\{x\}} \prod_p A_{x_{pi}x_{pj}x_{pk}x_{pl}}^p \quad (14)$$

$$A_{ijkl}^p = \tanh(\beta)^{\frac{1}{2}(i+j+k+l)} \delta_{\text{mod}(i+j+k+l, 2), 0} \quad (15)$$

ここでは、 $i = x_{pi}, j = x_{pj}, k = x_{pk}, l = x_{pl}$ と再定義し、 A はテンソルである。

2.2 テンソルの分解

図 1-a と図 2-1to2 で図示されている分解は特異値分解とよばれる行列分解を行っている。下記の式はテンソル A_{ijkl} を添字の組み合わせが異なる 2 種類の特異値分解している式になる。まず、テンソル A_{ijkl} の添字を 2 つにまとめた行列 M_{ab} である。そして、その行列 M_{ab} を特異値分解を実行すると次のようになる。

$$M_{ab} = U_{ac}\Sigma_c V_{cb}^\dagger \quad (16)$$

ただし、 U, V はユニタリ行列であり、 Σ は対角成分以外は 0 を持つ行列である。そのために対角成分 Σ_c だけを用いている。最後に図 1-a のように表せるようにテンソル S を用いて、整理する。

$$A_{ijkl} = M_{(i\otimes j)(k\otimes l)} = U_{(i\otimes j)n}\Sigma_n V_{n(k\otimes l)}^\dagger = S_{ijn}^1 S_{kln}^3 \quad (17)$$

$$S_{ijn}^1 = U_{(i\otimes j)n}\sqrt{\Sigma_n}, \quad S_{kln}^3 = \sqrt{\Sigma_n}V_{n(k\otimes l)}^\dagger \quad (18)$$

$$A_{ijkl} = \tilde{M}_{(j\otimes k)(l\otimes i)} = \tilde{U}_{(j\otimes k)n}\Sigma_n \tilde{V}_{n(l\otimes i)}^\dagger = S_{jkn}^2 S_{lin}^4 \quad (19)$$

$$S_{jkn}^2 = \tilde{U}_{(j\otimes k)n}\sqrt{\Sigma_n}, \quad S_{lin}^4 = \sqrt{\Sigma_n}\tilde{V}_{n(l\otimes i)}^\dagger \quad (20)$$

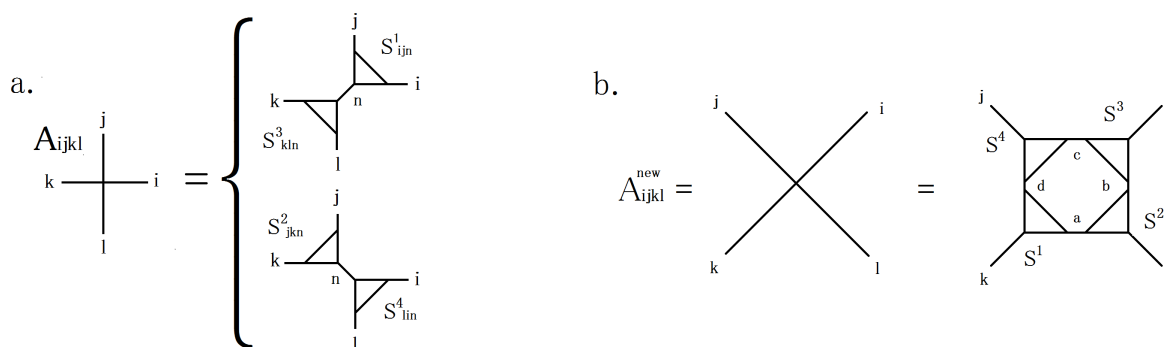


図 1 テンソルくりこみ群の分解と縮約

2.3 自由度の縮約

図 1-b、図 2-4to5 で図示されている総和は 4 つのテンソル S を 1 つのテンソル A にまとめている。下記の式は図 1-b で表しているように 4 つのテンソル S が持つ古い bond を表している添字 a, b, c, d をすべて総和し、残りの新しい bond を表している添字 i, j, k, l で表されるテンソル A_{ijkl}^{new} が生成されている式になる。そして、この手順を自由度の縮約という。

$$A_{ijkl}^{new} = \sum_{a,b,c,d} S_{adk}^1 S_{bal}^2 S_{cbi}^3 S_{dcl}^4 \quad (21)$$

2.4 テンソルくりこみ群

図 2-1 は各格子点にオリジナルテンソル A を持つ格子を表している。図 2-2 は全格子点に特異値分解 (図 1-a を参照) を行い、自由度の次元を下げる新しい bond を生成し、格子点の数を 2 倍にしている。

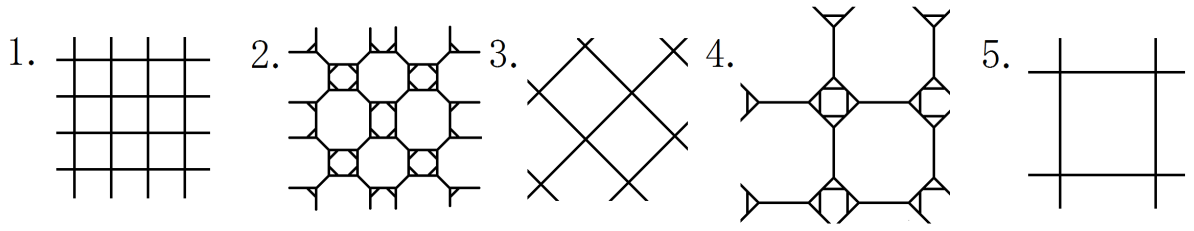


図2 テンソルくりこみ群のくりこみ過程

図 2-3 は古い bond をすべて総和し、4 点を 1 点としている (図 1-b を参照)。ここで 1 回のテンソルくりこみ群のくりこみを実行した時の結果になるが、このままでは境界条件が元の格子点と異なるため、もう一度テンソルくりこみ群のくりこみを行う。図 2-4 は図 2-2 と同様に全格子点に特異値分解を行い、新しい bond を生成している。図 2-5 は図 2-3 と同様に古い bond をすべて総和し、4 点を 1 点にくりこみ、新しいテンソル A^{new} を生成している。この 2 回のテンソルくりこみ群のくりこみにより、格子点の境界条件はくりこみ前のオリジナルテンソルを持つ格子点と同じになる。そして、この新しいテンソル A^{new} を基に再びくりこみを行うことでさらに大きな体積の系を計算することができる。

3 テンソルくりこみ群を用いた相関関数の計算手法

相関関数を求めるためにテンソルネットワークの中に一つだけ異なるテンソルを組み込む。このテンソルのことを不純物テンソルという。また、本論文では分配関数から求めたテンソルのことを純粋テンソルと呼ぶことにする。そして、不純物テンソルを含むテンソルネットワークをテンソルくりこみ群の手法でくりこみを実行することで一点関数を求めるために必要なテンソルネットワークが得られる。さらに、一点関数を求めるためにくりこみを何度も行われたテンソルネットワークを用いて、二点関数を求めるために必要なテンソルネットワークを得ることができる。最終的に、これらのテンソルネットワークを足しあげることによって一点関数と二点関数を求めることができる。くりこみの回数により、大きな体積でのそれぞれの関数を求められることができ、二点関数の引数である二点の距離を決めることができる。

3.1 不純物テンソル

2 節で説明したイジング模型のテンソルネットワーク表示の分配関数を導いたように、今度は分配関数の代わりに格子点上の q 点だけにスピン s_q を置いた次のように定義した Z_1 をテンソルネットワーク表示に書き換える。 Z_1 は実は一点関数の期待値 $\langle s_q \rangle = \frac{Z_1}{Z}$ の分子であり、テンソルネットワーク表示にすると簡単に次のように表せる。

$$\langle s_q \rangle = \frac{\sum_{\{s\}} s_q e^{\beta \sum_{(a,b)} s_a s_b}}{Z} \quad (22)$$

$$Z_1 \equiv \sum_{\{s\}} s_q e^{\beta \sum_{(a,b)} s_a s_b} \rightarrow \sum_{\{i,j,k,l\}} \check{A}_{ijkl}^q \prod_{p \neq q} A_{ijkl}^p \quad (23)$$

この Z_1 の中にある A は 2 章で説明したイジング模型のテンソルネットワーク表示のテンソル A と同じである。しかし、テンソル \check{A} はテンソル A とは異なり、この \check{A} を不純物テンソルと呼ぶ。

では、詳細に Z_1 をテンソルネットワーク表示に書き換えると次のように導くことができる。

$$Z_1 = \sum_{\{s\}} s_q \prod_{\langle a,b \rangle} e^{\beta s_a s_b} \quad (24)$$

$$= \sum_{\{s\}} s_q \prod_{\langle a,b \rangle} \cosh(\beta) (1 + \tanh(\beta) s_a s_b) \quad (25)$$

$$= \sum_{\{s\}} s_q \prod_{\langle a,b \rangle} \cosh(\beta) \sum_{x_{ab}=0,1} (\tanh(\beta) s_a s_b)^{x_{ab}} \quad (26)$$

$$= \cosh(\beta)^{2V} \sum_{\{x\}} \sum_{s_q=\pm 1} \tanh(\beta)^{\frac{1}{2}(x_{qi}+x_{qj}+x_{qk}+x_{ql})} s_q s_q^{x_{qi}+x_{qj}+x_{qk}+x_{ql}} \\ \prod_{p \neq q} \sum_{s_p=\pm 1} \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} s_p^{x_{pi}+x_{pj}+x_{pk}+x_{pl}} \quad (27)$$

$$= \cosh(\beta)^{2V} \sum_{\{x\}} \tanh(\beta)^{\frac{1}{2}(x_{qi}+x_{qj}+x_{qk}+x_{ql})} (1 + (-1)^{x_{qi}+x_{qj}+x_{qk}+x_{ql}+1}) \quad (28)$$

$$\prod_{p \neq q} \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} (1 + (-1)^{x_{pi}+x_{pj}+x_{pk}+x_{pl}}) \quad (29)$$

$$= 2^V \cosh(\beta)^{2V} \sum_{\{x\}} \tanh(\beta)^{\frac{1}{2}(x_{qi}+x_{qj}+x_{qk}+x_{ql})} \delta_{\text{mod}(x_{qi}+x_{qj}+x_{qk}+x_{ql}+1,2),0} \quad (30)$$

$$\prod_{p \neq q} \tanh(\beta)^{\frac{1}{2}(x_{pi}+x_{pj}+x_{pk}+x_{pl})} \delta_{\text{mod}(x_{pi}+x_{pj}+x_{pk}+x_{pl},2),0} \quad (31)$$

$$= 2^V \cosh(\beta)^{2V} \sum_{\{x\}} \check{A}_{x_{qi}x_{qj}x_{qk}x_{ql}}^q \prod_{p \neq q} A_{x_{pi}x_{pj}x_{pk}x_{pl}}^p \quad (32)$$

$$A_{ijkl}^p = \tanh(\beta)^{\frac{1}{2}(i+j+k+l)} \delta_{\text{mod}(i+j+k+l,2),0} \quad (33)$$

$$\check{A}_{ijkl}^p = \tanh(\beta)^{\frac{1}{2}(i+j+k+l)} \delta_{\text{mod}(i+j+k+l+1,2),0} \quad (34)$$

3.2 一点関数を求めるためのテンソルネットワークのくりこみ手順

不純物テンソルを含むテンソルネットワーク表示を用いて、一点関数を求める。図 3 は一点関数を求める場合のくりこみの過程を表している。まず、図 3-1-1 のように不純物テンソル \check{A} が 1 点だけ存在するテンソルネットワークを考える。(その他のテンソルはすべてテンソル A である。) 次に図 3-1-1to3 のようにテンソルの分解と縮約を行い、図 3-1-3 のように不純物テンソル \check{A} の影響を受けた 2 点の不純物テンソルが存在する新しいテンソルネットワークが生成される。そして、図 3-1-1to3 と同様に図 3-1-4to5 のようにテンソルの分解と縮約を行い、元の境界条件を満たすテンソルネットワークを生成する。この図 3-1-1to5 のくりこみを 1 回のくりこみとする。図 3-1-5 では不純物テンソル \check{A} の影響を受けた 3 点の不純物テンソル $\check{A}_1^1, \check{A}_2^1, \check{A}_3^1$ が存在する新しいテンソルネットワークが生成されている。最終的に、この手順を再び行くと図 3-3-1 になり、不純物テンソル \check{A} の影響を受けた 4 点の不純物テンソル $\check{A}_1^2, \check{A}_2^2, \check{A}_3^2, \check{A}_4^2$ が存在する新しいテンソルネットワークが生成されている。あとは図 3-3-1to5 をくり返すことで十分な大きな体積での一点関数を求めることができる。なお、図 3 を見ると、このくりこみをくり返しても不純物テンソル \check{A} の影響を受けた不純物テンソルは必ず 4 点に収束されることがわかる。

3.3 二点関数を求めるためのテンソルネットワークのくりこみ手順

一点関数を求める際に生成されるテンソルネットワークを用いて、二点関数を求める。

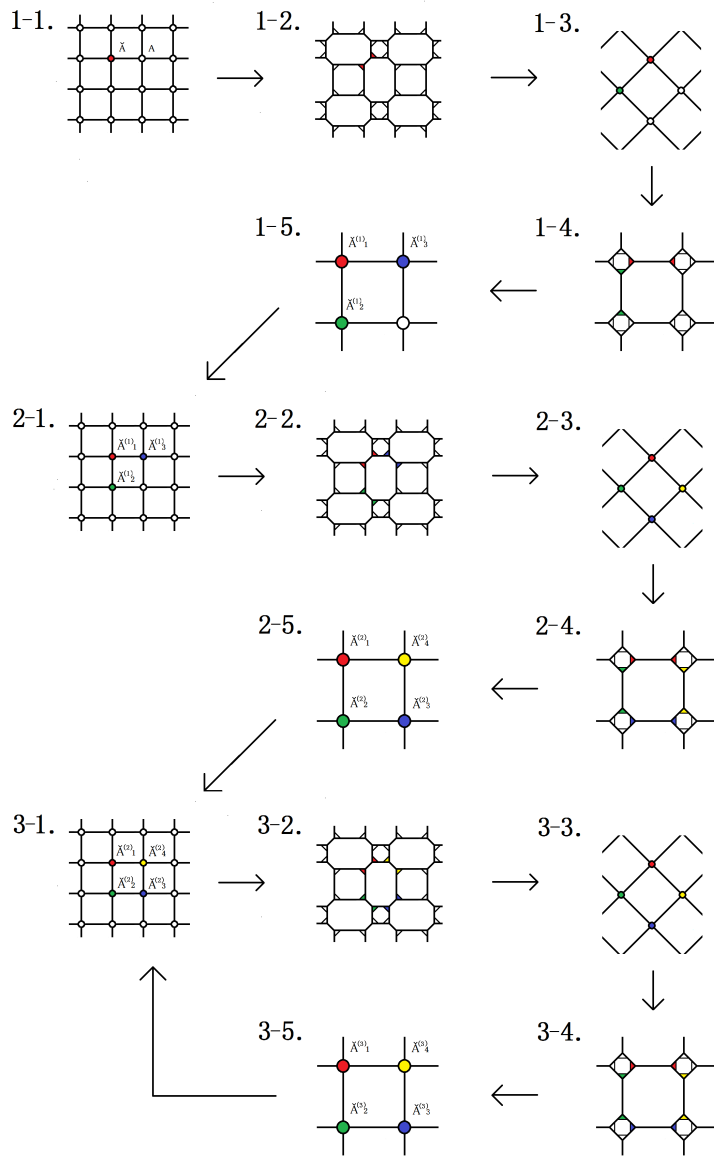


図3 一点関数を求めるテンソルくりこみ過程

図 4-1 は二点関数を求める場合の一回目のくりこみの過程を表している。まず、図 4-1-1 のように一点関数を求めるために得た 4 点の不純物テンソル $\check{A}_1^N, \check{A}_2^N, \check{A}_3^N, \check{A}_4^N$ が 2 セット存在するネットワークを組み合わせたテンソルネットワークを考える。次に図 4-1-1to3 のようにテンソルの分解と縮約を行うと、図 4-1-3 のように不純物テンソル \check{A} の影響を受けた 7 点の不純物テンソルが存在する新しいテンソルネットワークを生成される。そして、図 4-1-1to3 と同様に図 4-1-3to5 のようにテンソルの分解と縮約を行い、元の境界条件を満たすテンソルネットワークを生成する。図 4-1-5 では不純物テンソル \check{A} の影響を受けた 6 点の不純物テンソル $A_1^1, A_2^1, A_3^1, A_4^1, A_5^1, A_6^1$ が存在する新しいテンソルネットワークが生成されている。

図 4-2-1 は二点関数を求める場合の二回目のくりこみの過程を表している。まず、図 4-2-2 で求めた二点関数を求めるために得たテンソル $A_1, A_2, A_3, A_4, A_5, A_6$ が存在するネットワークを組み合わせた

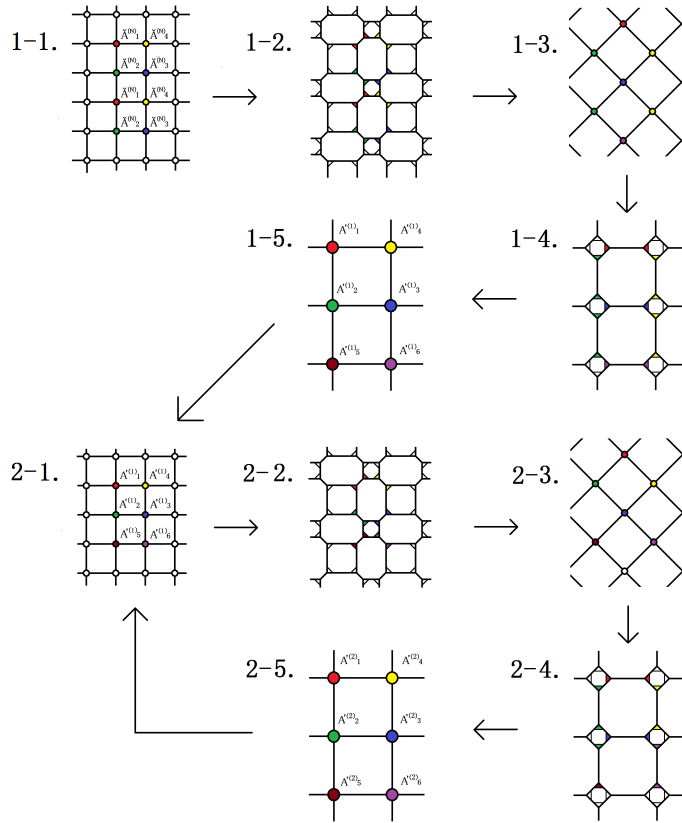


図 4 二点関数を求めるテンソルくりこみ過程

テンソルネットワークを考える。次に図 4-2-1to3 のようにテンソルの分解と縮約を行うと、図 4-2-3 のように 6 点の不純物テンソルの影響を受けた 6 点の不純物テンソルが存在する新しいテンソルネットワークを生成される。そして、図 4-2-1to3 と同様に図 4-2-3to5 のようにテンソルの分解と縮約を行い、元の境界条件を満たすテンソルネットワークを生成する。図 4-2-5 では不純物テンソル \tilde{A} の影響を受けた 6 点の不純物テンソル $A'_1, A'_2, A'_3, A'_4, A'_5, A'_6$ が存在する新しいテンソルネットワークが生成されている。最後に、図 4-2-1 に戻り、くりこみをくり返すことで十分な大きな体積での二点関数を求めることができ、図 4 のようにくりこみを行うごとに不純物テンソルは 8 点から 7 点へ、7 点から 6 点へと収束していき、最終的に何度くり返しても 6 点のテンソルに収束する。

4 数値計算結果

2 次元 Ising 模型の相関関数を求めるくりこみの手法を用い、 $D_{\text{cut}} = \{24, 32\}$ で数値計算した結果を示し、考察する。まず、フィット関数を用い、相関関数から相関長と臨界指数 η 、自発磁化を取り出す。この時、 $D_{\text{cut}} = \{24, 32\}$ から得た相関関数の差を誤差として、フィットの信頼性を高めるためとフィット領域の決定のために導入した。つぎに取り出した相関長と自発磁化から臨界温度と臨界指数 ν, β をフィット関数を用いることで取り出した。この時、フィット領域を $D_{\text{cut}} = \{24, 32\}$ から定義した誤差を用い、決定した。

4.1 相関関数から相関長 ξ と臨界指数 η , 自発磁化 m を取り出す方法

今回は外部磁場 h を $h = 0$ とし、体積 $V = 2^{21} \times 2^{21}$ の 2 次元 Ising 模型の数値計算を行う。まず、臨界温度付近のスピン変数の相関関数 $G_s(r)$ を次のように表すことができる。

$$G_s(r) \equiv \langle s(r)s(0) \rangle_{\text{connected}} \quad (35)$$

$$= \langle s(r)s(0) \rangle - \langle s(r) \rangle \langle s(0) \rangle \quad (36)$$

$$= \frac{A}{rE} e^{-r/\xi}, \quad r \rightarrow \infty \quad (37)$$

ここで、 A, E はフィットパラメーターである。一方、本論文では次の二点関数を用いて、解析を行う。

$$G(r) = \langle s(r)s(0) \rangle = G_s(r) + \langle s(r) \rangle \langle s(0) \rangle \quad (38)$$

$$= C^2 + \frac{A}{rE} e^{-r/\xi}, \quad r \rightarrow \infty \quad (39)$$

各温度でフィットを行い、 C, E, A, ξ の値を決定することができる。臨界温度直上では E は臨界指数 η になる。なお、臨界指数 η は 2 次元 Ising 模型では $\eta = 0.25$ であり、図 5 を見ると臨界温度直上では E は 0.25 に近づいていることが確認できる。このフィットで得た相関長 ξ とフィットパラメーター E の結果を図 5 に示す。

4.2 相関長 ξ から臨界温度 T_c と臨界指数 ν を取り出す方法

臨界温度 T_c 付近での相関長 ξ のフィット関数を次のように表すことができる。

$$\xi(T) \equiv B|T - T_c|^{-\nu} \quad (40)$$

前節で求めた相関長からこのフィット関数で臨界指数 ν と臨界温度 T_c を求める。そのフィットで得た臨界温度 T_c と臨界指数 ν を表 (1) に示す。

4.3 自発磁化 m から臨界温度 T_c と臨界指数 β を取り出す方法

このフィットパラメーター C を自発磁化 m とみなし、臨界温度 T_c と臨界指数 β を以下を求める。臨界温度 T_c 付近での自発磁化 m のフィット関数を次のように表すことができる。

$$C = m(T) = D|T - T_c|^\beta \quad (T < T_c) \quad (41)$$

臨界指数 β は 2 次元 Ising 模型では $\beta = 0.25$ である。1 節で求めたフィットパラメーター C からフィット関数で臨界温度 T_c と臨界指数 β を求める。そのフィットで得た臨界温度 T_c と臨界指数 β を表 (1) に示す。高温相 ($T > T_c$) における相関長と低温相 ($T < T_c$) における自発磁化をそれぞれの関数形を用いて、フィットで臨界温度 T_c と臨界指数 ν, β を求め、表 (1) で示した。その結果、高温相における相関長と低温相における自発磁化から得た臨界温度 T_c と臨界指数 ν, β は厳密値に誤差の範囲内で一致し、信頼の高い値をそれぞれ得たと考えている。また、低温相 ($T < T_c$) における相関長から得た臨界温度 T_c は高温相の相関長と自発磁化からの結果に比べると厳密値からのズレが大きかった。その原因だと考えられる一つにフィット関数が低温相では適したフィット関数ではなく、正しい臨界温度と臨界指数を取り出すことができなかつたという可能性がある。二つめに適したフィット領域を採用していない可能性がある。三つめに図 5 の臨界指数 η の振る舞いが明らかに不安定な振る舞いをしていることから低温相が高温相に比べて、スケーリングの破れが強い可能性がある。

	T_c	β	ν
厳密値	2.269185...	0.125	1
$\xi(T > T_c)$	2.26918(10)	–	0.983(30)
$m(T < T_c)$	2.26914(19)	0.1191(63)	–

表 1 2次元 Ising 模型の厳密値、テンソルくりこみ群の相関関数からの臨界指数

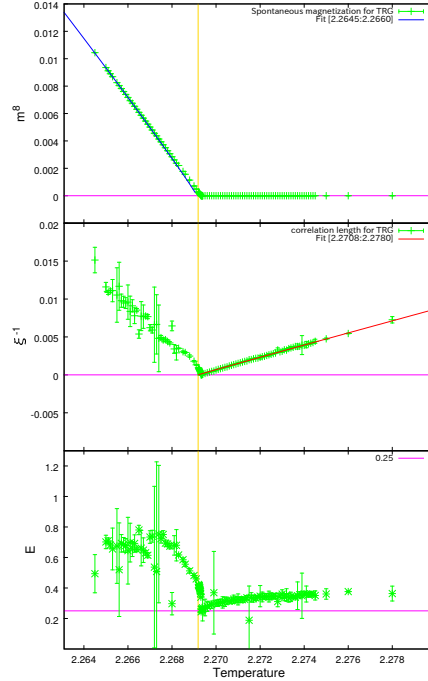


図 5 相関長と自発磁化と臨界指数 η の温度変化

5 まとめ

今まで説明した手法は不純物テンソルと呼ばれるテンソルを含むテンソルネットワークを構成し、テンソルくりこみ群を実行することで相関関数を得ることができる。この手法の長所はくりこみをくり返すごとに格子体積を指数関数的に増加させることができる。そのため、格子計算の分野でよく知られているモンテカルロ法と呼ばれる確率計算よりもはるかに大きな体積での数値計算を行うことができる。そして、得られた相関関数が正しいのであれば、相関関数から臨界指数や臨界温度を抽出できると考え、検証を行った。その結果は表 (1) に示した。この結果を見ると十分に 2次元 Ising 模型の臨界指数と臨界温度を再現することができたと考えられる。つまり、この手法は大きな D_{cut} や追加パラメータを要求せずに異なる模型や高次元の模型に対しても臨界指数や臨界温度を抽出する強力な方法になるということが推測される。

本論文は初めて 2次元 Ising 模型の相関関数から力学的な情報である臨界指数や臨界温度を抽出できることを実証し、大きい D_{cut} や追加パラメータを要求せずに正しい臨界指数や臨界温度を再現することができた。

学位論文審査報告書（甲）

1. 学位論文題目（外国語の場合は和訳を付けること。）

テンソルくりこみ群を用いた相関関数の評価

2. 論文提出者 (1) 所 属 自然科学研究科数物科学 専攻

(2) 氏 名 中本 智洋 (なかもと のりひろ)

3. 審査結果の要旨 (600~650 字)

テンソルくりこみ群はモンテカルロ法に比べて、符号問題がないことや、系の体積を非常に大きくできるなど優れた点がある。本論文は、テンソルくりこみ群の枠組みの中で、特に 2 次元スピン系において相関関数を計算する手法を確立し、その検証を行ったものである。

計算手法のベンチマーク指標として、厳密解が知られている 2 次元イジングモデルの臨界温度や臨界指数がよく使われている。テンソルくりこみ群では、臨界温度を決める方法として真空の縮退度を使った手法が用いられていたが、臨界点近傍で計算を行うことが必須の条件であった。しかし、テンソルくりこみ群は臨界点に近づくほど、その精度が失われるため、この方法の限界が指摘されていた。そこで、本論文では、臨界点から離れた精度が落ちない領域で計算した相関関数から臨界点の情報を引き出すという戦略を提唱し、その正当性と有用性を検証した。

本論文で提唱された手法は、他のモデルにも容易に適応できることから、今後の発展も期待される。この論文の計算はオリジナルであり独創性に問題はない。本博士論文の内容は、共同研究ではあるが、申請者が多くの面で主導しており、学位論文として十分であると認められるものである。

4. 審査結果 (1) 判 定 (いずれかに○印) 合 格 ・ 不合格

(2) 授与学位 博 士 (理 学)