

修士論文

4次元系でのテンソル繰り込み群手法の改良
Improved tensor renormalization group method
for four-dimensional systems

東北大学大学院理学研究科
物理学専攻

杉本 悠斗

2024年

4次元系でのテンソル繰り込み群手法の改良

Improved tensor renormalization group method for four-dimensional systems

東北大学 理学研究科 物理学専攻

C3SM2031 杉本悠斗

背景

量子色力学 (QCD) は、強い相互作用を記述する基本理論であり、陽子や中性子の内部に存在するクォークやグルーオンのダイナミクスを記述する重要な理論である。しかし、QCD は低エネルギー領域で結合定数が大きくなり摂動論的手法が適用できないため、通常は格子 QCD と呼ばれる数値的な手法が理論解析に用いられる。格子 QCD では物理量を統計的な手法であるモンテカルロ法に基づいて計算するが、有限密度での計算においては符号問題という深刻な困難が生じることが知られている。符号問題は、モンテカルロ法で取り扱う確率が複素数となるため、確率解釈が不可能になる現象であり、量子多体系に頻繁に現れる問題である。

モンテカルロ法に変わる新たなアルゴリズムとして、テンソル繰り込み群法 (TRG: Tensor Renormalization Group) [1] が近年注目を集めている。TRG 法では、分配関数などの物理量をテンソルの縮約 (テンソルネットワーク) という形式で表現し、特異値分解による情報の圧縮を繰り返すことで近似的に物理量を評価する手法である。TRG 法は、モンテカルロ法と異なり確率的操作を必要としないため、符号問題が存在せず、また熱力学極限を容易に取る事ができるなどのメリットもあり、有限密度 QCD の第一原理計算への応用が期待されている。しかしながら、高次元でのテンソル繰り込み群アルゴリズム (HOTRG) [2] は計算コストが大きく、特に 4次元系においてはその計算コストが $O(D^{15})$ (D はボンド次元と呼ばれる計算時間と精度をコントロールするパラメーターで、通常 $O(10)$ 程である。) に及ぶため、QCD などの内部自由度の大きい系を取り扱うには現実的ではない。そのため、HOTRG 法を近似したアルゴリズムである ATRG 法 [3] が 4次元系の計算において用いられている [4]。ATRG 法は計算コストを $O(D^9)$ まで削減できたものの、HOTRG 法同様に計算コストの増大によりボンド次元を十分に大きく取れない問題がある。一方、HOTRG 法の近似として ATRG 法と異なるアプローチを取ったものが、近年提案された MDTRG 法 [5] である。同手法では、精度の低下を抑えつつ計算コストを削減する手法として、改良された Triad 表現 [6] と呼ばれるテンソルネットワークの表現を用いている。実際に 3次元の Ising モデルにおいて、HOTRG 法と同程度の系統誤差を達成しながらも、計算コストを大幅に削減することに成功しているが、4次元系において適用された例はない。上記のような背景を踏まえ、本研究では将来の QCD などの複雑な 4次元理論の計算の実現に向けて更に高速なアルゴリズムや計算手法の開発を行う事を主題とする。

本研究の概要

本研究では、4次元のテンソル繰り込み群の高速化手法として、ATRG 法に Triad 表現を導入した Triad-ATRG 法を提案した。Triad-ATRG 法は、ATRG 法の精度を保ちつつ計算コストを $O(D^9) \rightarrow O(D^7)$ に大幅に削減したアルゴリズムであり、より大きなボンド次元 D に対して計算を実行する事が可能である。また、本研究では、ATRG や Triad-ATRG をテンソルの縮約計算に適したハードウェアである GPU (画像処理用の演算装置) 上で実装するアルゴリズムの開発も同時に行った。

まず、Triad-ATRG 法の CPU 上での計算時間のスケーリングを調査し、理論通りの計算時間の削減が達成できることを確認した。同時に、GPU 上での計算時間のスケーリングも調査し、ATRG

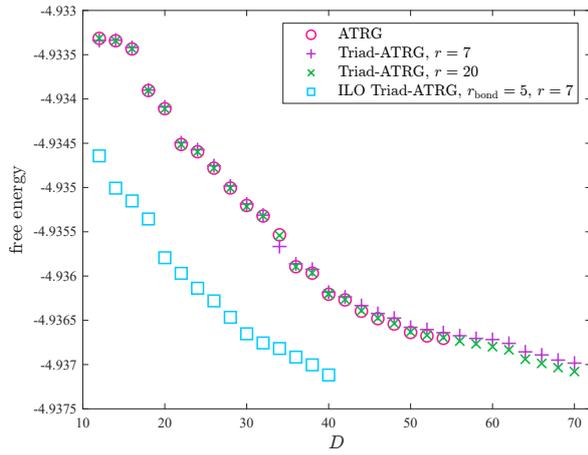


図 1: Triad-ATRГ 法, ATRГ 法, ILO Triad-ATRГ 法の自由エネルギーの比較.

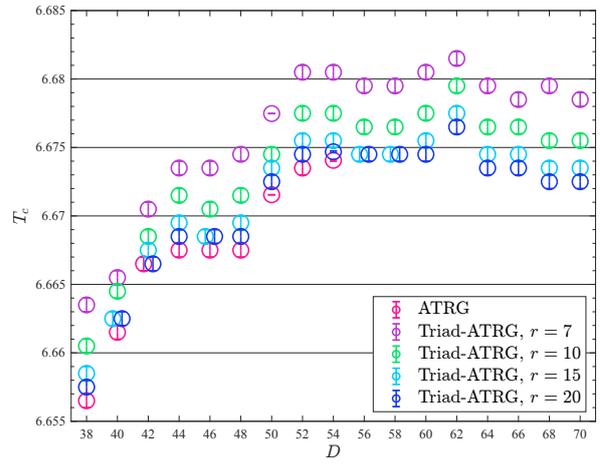


図 2: ATRГ 法, Triad-ATRГ 法での相転移点の比較. 誤差は温度の解像度である.

法と Triad-ATRГ 法ともに CPU と比較してよりスケーリングの改善が確認された. 特に, Triad-ATRГ 法は $O(D^6)$ 以下のスケーリングを示しており, ボンド次元を大きくするに従い大幅な計算時間の短縮が可能となった.

次に, Triad-ATRГ 法が ATRГ 法と同等の精度を与えていることを, 4次元の Ising モデルの自由エネルギーを計算することで確認した (図 1). 特に, $D = 54$ では, ATRГ 法と Triad-ATRГ 法 ($r = 20$) との差異は, わずか 0.00015% であった.

次に, 相転移点や内部エネルギーなどの物理量についても, Triad-ATRГ 法が ATRГ 法と同等の精度を持つことを実際の数値計算で確認した. 特に, 相転移点の計算では, 先行研究より大きい $D = 70$ までの計算が可能となり, 相転移点の収束性が D を大きくすることで向上することを示した (図 2). 以上の結果は, Triad-ATRГ 法が ATRГ 法と同等の精度を持ちながら, 大幅に計算コストを削減した画期的なアルゴリズムであり, 4次元系の計算において強力な手法になることを示唆している.

次に, Triad-ATRГ 法や ATRГ 法で一番近似が激しい部分であるボンド入れ替えと呼ばれる操作について, [5] で提案された内線オーバーサンプリングの手法を Triad-ATRГ 法に適用したアルゴリズム (ILO Triad-ATRГ) を提案し, さらなる精度の向上を図った. この結果, ILO Triad-ATRГ 法は ATRГ 法よりも, 同じボンド次元でより低い自由エネルギーを達成する事が示された (図 1). 論文では, 以上に加えて, Triad-ATRГ 法を用いた 4次元イジングモデルの相構造の解析も行なっている. 本研究で得られた結果は, Triad-ATRГ 及び ILO Triad-ATRГ が 4次元において効率的な手法であることを強く示しており, 将来の有限密度 QCD などの複雑な 4次元理論の計算の実現に向けた重要な基盤手法となることが期待される.

参考文献

- [1] M. Levin and C. P. Nave, *Phys. Rev. Lett.* **99**, 120601 (2007), [arXiv:cond-mat/0611687].
- [2] Z. Y. Xie et al., *Phys. Rev. B* **86**, 045139 (2012), [arXiv:1201.1144].
- [3] D. Adachi et al., *Phys. Rev. B* **102**, 054432 (2020), [arXiv:1906.02007].
- [4] S. Akiyama and Y. Kuramashi, *JHEP* **10**, 077, [arXiv:2304.07934].
- [5] K. Nakayama (2019), [arXiv:2307.14191].
- [6] D. Kadoh and K. Nakayama (2019), [arXiv:1912.02414].

発表論文

- [7] Y. Sugimoto and S. Sasaki, *PoS LATTICE2024*, 038 (2025), [arXiv:2412.14104]

目次

Notation	3
0.1 記法	3
0.2 テンソル演算のダイアグラム表記	3
0.3 テンソルネットワーク	4
0.4 縮約順序	4
第1章 序論	6
1.1 格子 QCD と符号問題	6
1.2 繰り込み群	6
1.3 テンソル繰り込み群研究のこれまで	7
1.4 本研究の目的	8
第2章 テンソル繰りこみ群の基礎	9
2.1 数学的基礎	9
2.1.1 特異値分解	9
2.1.2 低ランク近似	10
2.1.3 高次特異値分解	11
2.2 テンソルネットワーク表現	15
2.2.1 分配関数のテンソルネットワーク表現	15
2.2.2 相関関数のテンソルネットワーク表現	17
2.3 テンソル繰り込み群のアルゴリズム	18
2.3.1 Levin-Nave TRG	18
2.3.2 HOTRG	21
2.3.3 impurity テンソルの入ったネットワークの粗視化	24
2.3.4 ATRG	25
2.3.5 Triad-TRG 及び MDTRG	30
2.3.6 TRG の応用の現状と課題	33
第3章 本研究で使用する模型	35
3.1 臨界現象とスケールリング則	35
3.2 Ising モデル	35
第4章 研究 (アルゴリズム)	38
4.1 Triad-ATRГ	38
4.2 GPU による実装	43
4.3 ボンド入れ替えのオーバーサンプリング	44

第 5 章	数値計算結果	47
5.1	CPU での計算コストのスケーリング	47
5.2	GPU での計算時間のスケーリング	48
5.3	自由エネルギーの精度	49
5.4	相転移点の精度	50
5.5	内部エネルギーの計算	52
5.6	HOOI による Triad 表現	52
5.7	ボンド入れ替えのオーバーサンプリング	54
5.8	4次元イジングモデルの相転移の性質	55
5.8.1	内部エネルギーの体積依存	55
5.8.2	臨界指数	57
第 6 章	結論と考察及び展望	60
6.1	結論と考察	60
6.2	今後の展望	62
A	特異値分解に関する定理の証明	65
A.1	Theorem 1 の証明	65
A.2	Theorem 2 の証明	66
B	繰りこみ群とスケーリング則	67
C	TRG アルゴリズムの補足	71
D	乱択化特異値分解	72
E	2次元での TRG アルゴリズム	73
E.1	Bond-weighted TRG	73
E.2	Tensor Network Renormalization	76
E.2.1	CDL テンソル	76
E.2.2	TNR	77
E.2.3	Gilt	79

Notation

0.1 記法

本研究では、特にことわりのない限り以下の記法や定義を用いる。

- ・ \mathbf{u} や \mathbf{v} などの太文字の小文字は列ベクトルを表す。
- ・ $A, B, U, \Sigma, V^\dagger$ などの大文字は行列またはテンソルを表す。
- ・ a_{ij}, T_{ijkl} などの添字付きのアルファベットはテンソルや行列の成分を表す。
- ・ テンソルや行列はすべて複素数として扱う。
- ・ 第2章以降では、基本的にテンソルや行列の各添字は特にことわりのない限り1から D まで走る。この D をボンド次元と呼ぶ。たとえば $M = (m_{ij}) \in \mathbb{C}^{D \times D}$ 。
- ・ Einstein の総和記法を用いない。

0.2 テンソル演算のダイアグラム表記

一般に多数のテンソルの縮約や分解は複雑であり、その際にダイアグラム表記を使用することが有用である。本研究では以下のようなダイアグラム表記を用いる。

- ・ ベクトル: $v_i = \text{---} \textcircled{\mathbf{v}} \text{---} \overset{i}{\text{---}}$
- ・ 行列: $M_{ij} = \text{---} \textcircled{\mathbf{M}} \text{---} \overset{j}{\text{---}}$
- ・ テンソル: $T_{ijkl} = \text{---} \textcircled{\mathbf{T}} \text{---} \overset{j}{\text{---}} \overset{i}{\text{---}} \underset{k}{\text{---}} \underset{l}{\text{---}}$

色付きの円はテンソル自体を表し、線はテンソルの添字を表す。このように定義した行列に対し、行列積やテンソルの演算を視覚的に表現することができる。縮約を取る際は二つの足を結び、内線はすべて縮約を取るものとし、外線は新しくできるテンソルの足を表す。逆にテンソルを分解する場合は、新たな内線を引く。例えば、

- ・ 行列積および行列の分解:

$$K_{ij} = \sum_k^D M_{ik} N_{kj}$$

$$\Leftrightarrow \text{---} \textcircled{\mathbf{K}} \text{---} \overset{j}{\text{---}} = \text{---} \textcircled{\mathbf{M}} \text{---} \overset{k}{\text{---}} \text{---} \textcircled{\mathbf{N}} \text{---} \overset{j}{\text{---}}$$

- ・ テンソルの縮約及び分解:

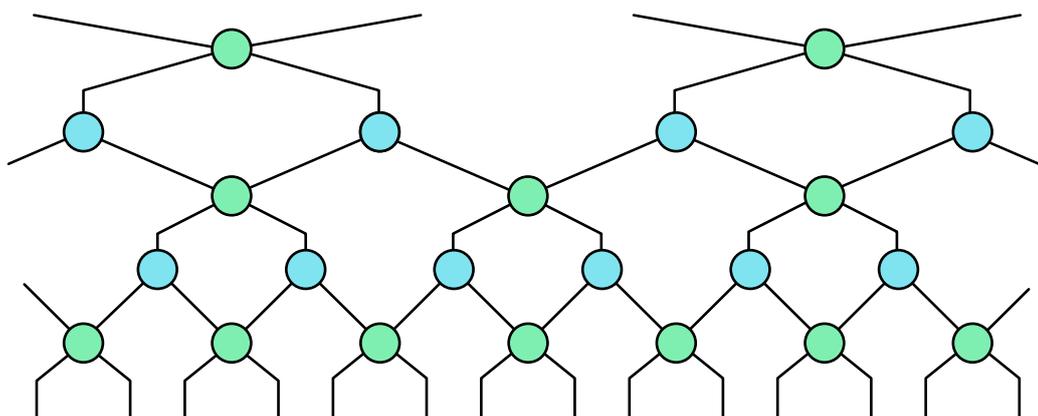
$$T_{ijkl} = \sum_{\alpha} L_{jka} R_{\alpha li}$$

$$\Leftrightarrow \begin{array}{c} j \\ | \\ \text{---} \text{---} T \text{---} \\ | \\ k \quad i \\ | \\ l \end{array} = \begin{array}{c} j \\ | \\ \text{---} \text{---} L \text{---} \\ | \\ k \quad \alpha \end{array} \begin{array}{c} \\ | \\ \text{---} \text{---} R \text{---} \\ | \\ l \quad i \end{array}$$

のように表す.

0.3 テンソルネットワーク

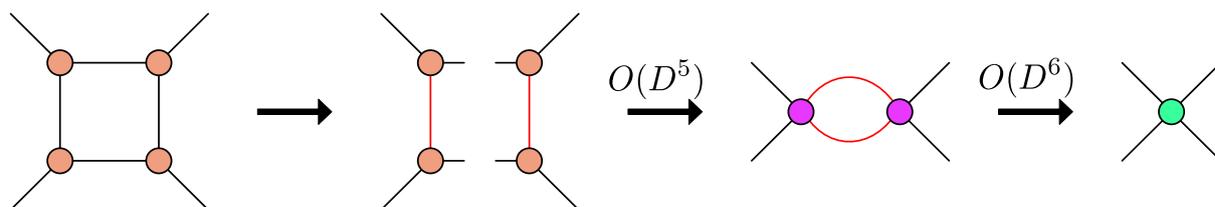
テンソル繰り込み群では, 多数のテンソルがつながった縮約を考えることが多い. そのようなテンソルが多数繋がったグラフをテンソルネットワークと呼ぶ. 例えば, 下のようなものがテンソルネットワークである.



本研究ではテンソルの分解, 縮約を頻繁に行うため, 積極的にダイアグラム表記を用いる.

0.4 縮約順序

テンソルネットワークの計算を行う上で, 縮約の順序は重要である. どの内線を順番に縮約するかによって, 計算の効率が大きく変わる. 例えば以下のようなテンソルネットワークの縮約を考えた場合, 素朴にすべての足を for 文で足し上げると $O(D^8)$ のコストがかかる. しかし, 縮約の順番を図のように適切に分け, 複数回に分け行うことで, 計算コストは $O(D^6)$ になる. 赤い線はそのステップで縮約を取る内線を表す.



このような縮約の順序の最適化は NP 困難であることが知られているが, 効率的に縮約順を決定するアルゴリズムが提案されている [1]. 本研究では, 最適な計算順序を括弧で示す. 例えば上図

の縮約の計算は,

$$T_{ijkl} = \sum_{\alpha, \beta, \gamma, \delta} (S_{i\alpha\beta} S_{j\beta\gamma}) (S_{k\gamma\delta} S_{l\delta\alpha})$$

のように表す. 計算順序が理論の次元や内線の大きさによって変わる場合, また複雑な場合は別途説明を加える.

第1章 序論

1.1 格子 QCD と符号問題

QCD は、強い相互作用を記述する基礎的な理論である。陽子や中性子といったハドロンはその内部にクォークやグルーオンといった粒子が QCD に基づく複雑な多体系を形成している。また、近年注目されている中性子星は、内部で高密度な状態が形成されていると考えられており、そのような状態では核子の波動関数がオーバーラップし、クォーク、グルーオンといった自由度が無視できないため、QCD からの根源的理解が必要である。しかしながら QCD は低エネルギー領域では結合定数が大きくなり、摂動的計算が困難となる。そのため、時空を格子点に区切り、第一原理計算を行う格子 QCD と呼ばれる手法が広く用いられている。格子 QCD では、膨大な多重積分で表される物理量を統計的な手法であるモンテカルロ法によって計算を行う。具体的には、ボルツマン因子を確率として扱い、その確率に従ってサンプルを多数作成し、その統計的平均を取ることによって物理量を計算する。この方法により、ハドロンの質量などを高精度で予言することに成功してきた。一方、有限密度では、ボルツマン因子が複素数となるため、上記のような確率解釈が困難となる。この問題は符号問題と呼ばれ、格子 QCD のみならず量子多体系で頻繁に現れる問題である。符号問題はモンテカルロ法が原理的に持つ困難であり、新たなアプローチが必要とされる。複素ランジュバン法 [2,3] や、Lefschetz thimble 法 [4] などは符号問題を回避する手法の一例であり、有限密度 QCD への応用が期待されている。また、Levin と Nave によって提唱されたテンソル繰り込み群 [5] も、符号問題を回避する手法として近年注目されている。

1.2 繰り込み群

テンソル繰り込み群の概念を理解する上で、重要なことは自由度の粗視化という考え方である。一般に、自然界が持つ自由度はあまりに膨大で、複雑であり、全ての現象を厳密に解析することは困難である。そのため、物理学では系の持つ詳細な情報は一旦破棄し、重要な特徴を残しモデルを立てたり、低エネルギーでのみ有効な理論を扱うことがある。このように系の重要な特徴を残し、物理現象を粗く見ることは、物理学において最も基本的なアプローチである。これは、低エネルギーやマクロなスケールでの物理現象は系の詳細なメカニズムに依存しないという、自然界の階層性の構造を利用したものである。

上記のような考え方は、Wilson による繰り込み群 [6] の考え方を基礎に、有効理論の定式化に見ることが出来る。繰り込み群は、系の詳細な情報を粗視化して行き、有効な自由度を取り出すような手法である。ブロックスピン変換 [7] などは離散的な繰り込み変換の例である。例えば、スピンの格子状に配置されている系を考えた場合、隣接する 4 個の組を平均して、新たなスピンとして取り扱うようなブロックスピン変換 (図 1.1) を考えてみる。

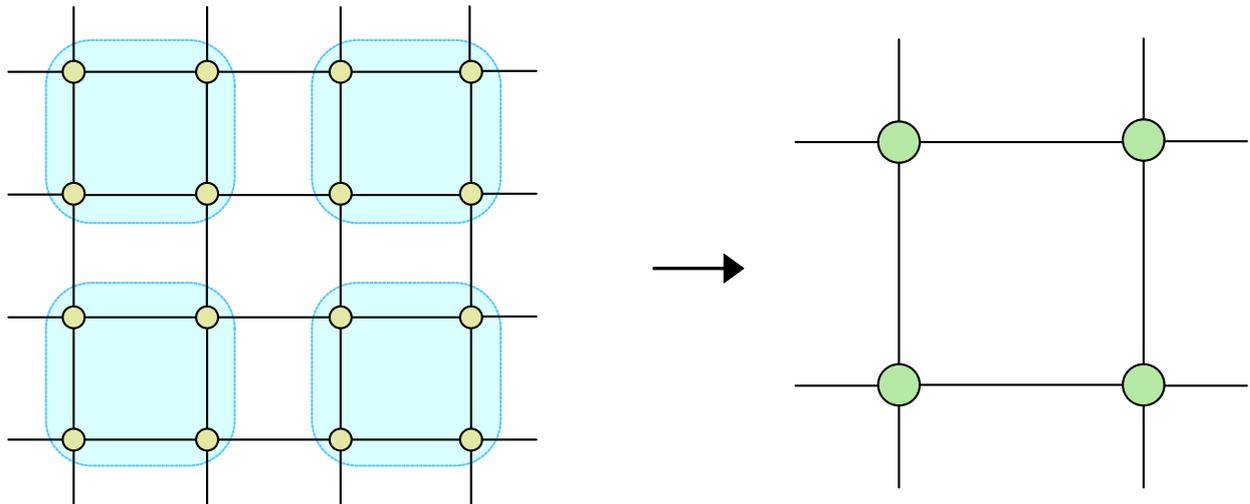


図 1.1: $N = 4$ のブロックスピン変換. 4つのスピンを1つのスピんに粗視化する.

この操作により, 系のスピンの数は $1/4$ になる. これは, 系を 2 倍に粗視化したということであり, 新しく定義された粗い格子よりもミクロな物理 (元のスピンの情報) は, 新たなスピんに “繰り込まれて” 見えなくなっている. このような変換を何度も行っていけば, その格子間隔は巨視的なスケールへと移っていく事がわかる. 最後に得られた “粗視化された” スピン系は, 最終的には高々数個のスピンのみからなる, 巨視的な格子間隔を持つ有効理論であり, 容易に計算が可能となる. このように有効な自由度を取り出して, 系を粗くみるという手法は, 物理学の根本的な “繰り込み” の思想に基づいた重要な概念である.

1.3 テンソル繰り込み群研究のこれまで

テンソル繰り込み群は, White の考案した密度行列繰り込み群 (DMRG) [8,9] に基づいている. DMRG は, 多体系の波動関数を行列積状態 (MPS) [10] と仮定して, 特異値分解と呼ばれる行列の圧縮法に基づく変分法で波動関数を最適化し, 1次元量子系の基底状態を効率よく計算するアルゴリズムである¹. DMRG は, ボンド次元と呼ばれる重要な寄与を持つ D 個の基底のみを有効な自由度として取り扱うことで, 波動関数を圧縮する. このように基底状態の重要な情報を残しつつ, ミクロな自由度を制限する (粗く見る) という点はまさに繰り込み群の思想に基づいている. DMRG は一次元系の基底状態を高精度で求める事ができる. これは, 特異値分解による情報選択が非常に効率的であるためである.

このような DMRG の圧縮法を用いてブロックスピン変換を近似的に実行するアルゴリズムが, Levin と Nave によるテンソル繰り込み群 (TRG: Tensor Renormalization Group) [5] である. TRG では, 2次元古典統計系や1次元量子系の密度行列をテンソルの縮約で表現する. このネットワークに対し, 特異値分解による局所的な近似を行なっていく事で, 分配関数などを近似する. TRG では, ブロックスピン変換のように, 複数のテンソルを一つにくりこむことで, 系を表現するテンソルの数を減らしていく. 同手法は, 2次元系では高精度で系の自由エネルギーを求める事ができる上, 以下のようなメリットがある.

- (1) 局所的なテンソルの近似のみを行うため, 系の体積を V としたとき計算コストが $\log V$ に比例する. これはモンテカルロ法 ($\propto V^\alpha$) よりも大幅に小さく, 熱力学極限 ($V \rightarrow \infty$) を取る事が容易である.

¹詳しくは [11] を参照.

- (2) 確率的操作を挟まないため、符号問題が発生しない。
- (3) フェルミオンを直接扱える。

このように、従来のモンテカルロ法にない利点を持つ手法であり、有限密度格子 QCD などへの応用が期待されている。しかしながら、TRG の高次元への拡張は、非自明であった。Xie らは、各時空軸を順番に繰り込んでいく事で、高次元でも TRG 計算が行える手法を提案した [12]。この手法は、HOTRG (Higher-Order Tensor Renormalization Group) と呼ばれる。しかしながら、HOTRG は高次元になるほど、計算コストが大幅に増大するという問題があり、ボソンの自由度を含むような内部自由度に富んだ系に対しての適用は難しく、特に QCD が定義される時空次元と同じ 4 次元では、自由度の少ない Ising 模型への適用にとどまっている [13]。そのため、HOTRG の計算コストを削減する様々な手法が発展してきた。その一つが ATRG (Anisotropic Tensor Renormalization Group) [14] である。ATRG は、HOTRG の手法に対して追加の近似を行う事で、計算コストを削減する手法であり、この手法により、符号問題を含むような 4 次元のボソン系やフェルミオン系、離散ゲージ系への応用が多数なされた [15–20]。しかしながら、計算コストは削減されたものの依然として高く、4 次元非可換ゲージ理論などの複雑な系への応用には至っていない。

1.4 本研究の目的

本研究は、4 次元におけるテンソル繰り込み群法の計算コストを大幅に削減することにより、将来の有限密度 QCD などへの応用の基礎となるようなアルゴリズムを提案する事を目的とする。

本研究では、4 次元の ATRG の計算コストの削減を、アルゴリズムとハードの二つの側面から行った。アルゴリズム面では、近年提唱された MDTRG (minimally-decomposed TRG) [21] の手法を取り入れる事で、ATRG の計算時間スケールを改善する新手法、Triad-ATRG を提唱した。ハード面では、近年その並列計算性能の高さから科学技術計算への応用が進められている GPU (画像処理装置) を用いた並列アルゴリズム及び計算コードの開発を行った。さらに本研究では、新手法を用いた 4 次元 Ising モデルの相転移解析を行い、Triad-ATRG の有効性を示した。

本論文は、以下のように構成されている。2 章では、テンソルネットワークの数学的基礎及び詳細なアルゴリズムについて説明する。3 章では、臨界現象と 4 次元 Ising モデルの相転移について説明する。4 章では、Triad-ATRG のアルゴリズムを概説し、5 章ではその数値計算結果について述べる。6 章では、本研究の結論と考察、及びこれからの展望について述べる。

第2章 テンソル繰りこみ群の基礎

テンソル繰りこみ群は、分配関数や物理量をテンソルの縮約の形式で表し、特異値分解 (SVD) を利用した近似的な実空間繰りこみ群手法を構成することにより、経路積分やボルツマン和などを近似的に評価する手法である。テンソルネットワーク形式の場の理論では、物理量の期待値や分配関数は多数のテンソルの縮約という形式で表される。そのテンソルに対して、分解及び低ランク近似を行うことで、縮約を近似的に実行し、系の重要な情報のみを取り出すというのがテンソル繰りこみ群の主旨である。すなわち、場の理論の数値計算は、いかに効率的にテンソルを近似するかという問題に置き換えることが可能である。行列及びテンソルの近似は、SVD を用いて数学的に計算される。本章では基礎となる SVD、さらにその一般化である高次特異値分解 (HOSVD)、これらを用いた行列やテンソルの低ランク近似とそのアルゴリズムについて説明をし、テンソル繰りこみ群の具体的なアルゴリズムについて簡潔にレビューを行う。

2.1 数学的基礎

2.1.1 特異値分解

一般に正方行列には、ジョルダン標準形と呼ばれる固有値分解の一般化された標準形が一意に定まることが知られている。しかし、一般の行列に対しても固有値分解のような、特異値分解 (SVD: Singular Value Decomposition) という操作を行うことが可能である。

Theorem 1. $m, n \in \mathbb{N}$ とする。任意の階数 r の複素行列 $A \in \mathbb{C}^{m \times n}$ に対し、正定値の特異値 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ が一意に定まり、

$$A = U \Sigma V^\dagger \quad (2.1)$$

なる分解が存在する。ここで $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ はそれぞれユニタリ行列であり、

$$\Sigma = \begin{pmatrix} \text{diag}(\sigma_1, \dots, \sigma_r) & O_{r, n-r} \\ O_{m-r, r} & O_{m-r, n-r} \end{pmatrix} \quad (2.2)$$

なる行列である。

ここで与えた SVD は \mathbf{u} と \mathbf{v} を全て求めることから、Full SVD と呼ばれている。対照的に、 U, V を階数 r で打ち切った非正方行列とし、 Σ を対角行列とする表現を Reduced SVD と呼ぶ [22]。本研究では Reduced SVD を用いる。両者の表現の違いは以下のようになる。

$$A \stackrel{\text{Full}}{=} (\mathbf{u}_1 \ \dots \ \mathbf{u}_n) \left(\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & O_{m-r, r} & & \\ & & & & O_{m-r, n-r} & \end{array} \right) \begin{pmatrix} \mathbf{v}_1^\dagger \\ \vdots \\ \mathbf{v}_m^\dagger \end{pmatrix} \quad (2.3)$$

$$\stackrel{\text{Reduced}}{=} (\mathbf{u}_1 \ \dots \ \mathbf{u}_r) \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\dagger \\ \vdots \\ \mathbf{v}_r^\dagger \end{pmatrix} \quad (2.4)$$

Reduced SVD では $U = (\mathbf{u}_1 \cdots \mathbf{u}_r) \in \mathbb{C}^{n \times r}$ であり, $U^\dagger U = I_r$, $UU^\dagger = P_u$ となることに注意する. ただし P_u は $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ で張られる r 次元空間への射影行列である. これは, V についても同様である. SVD をダイアグラムで表すと以下ようになる.

$$\text{---} \overset{A}{\circ} \text{---} = \text{---} \overset{U}{\circ} \overset{\Sigma}{\circ} \overset{V^\dagger}{\circ} \text{---} \quad (2.5)$$

2.1.2 低ランク近似

テンソルネットワークの計算を行う上で, 行列を近似することが重要である. 近似をするために, 行列のノルムの素朴な定義としてフロベニウスノルムを導入する.¹

Definition 1. $A \in \mathbb{R}^{m \times n}$ としたとき, A のフロベニウスノルムを

$$\|A\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} \quad (2.6)$$

とする.

このノルムは以下の性質を満たす.

Proposition 1. 式 (2.1), (2.2) を満たす $A = (a_{ij})$ において

$$\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n a_{ij}^* a_{ij} = \text{tr}[A^\dagger A] \quad (2.7)$$

が成り立ち, 式 (2.1) を用いると,

$$\text{tr}[A^\dagger A] = \text{tr}[V \Sigma^\dagger U^\dagger U \Sigma V^\dagger] = \text{tr}[\Sigma^\dagger \Sigma] = \sum_{i=1}^r \sigma_i^2 \quad (2.8)$$

が成り立つ. ただし, トレースの巡回性を用いた.

この性質を用いて, 以下の定理が証明できる [23].

Theorem 2. 式 (2.1) を満たす行列 A に対し $\|A - X\|_F = \min_{\text{rank } X' = k} \|A - X'\|_F$ を満たす階数 $k < r$ の複素行列 $X \in \mathbb{C}^{m \times n}$ は,

$$X = \sum_{m=1}^k \sigma_m \mathbf{u}_m \mathbf{v}_m^\dagger \quad (2.9)$$

$$= (\mathbf{u}_1 \cdots \mathbf{u}_n) \left(\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & O_{k,n-k} & \\ \hline & & & O_{m-k,k} & & \\ & & & & O_{m-k,n-k} & \end{array} \right) \begin{pmatrix} \mathbf{v}_1^\dagger \\ \vdots \\ \mathbf{v}_m^\dagger \end{pmatrix} \quad (2.10)$$

$$= (\mathbf{u}_1 \cdots \mathbf{u}_k) \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\dagger \\ \vdots \\ \mathbf{v}_k^\dagger \end{pmatrix} \quad (2.11)$$

で与えられる.²

¹別の定義として, L2 ノルムを導入してもよい.

²L2 ノルムを用いても最良の近似であることが証明できる.

この定理から、特異値分解は行列の最良な近似になっていることがわかる。この性質は画像処理や機械学習における主成分分析など幅広く応用されている。特異値分解の応用例を画像圧縮の例を用いて示す。図 2.1 は 3000×3000 ピクセルの画像を、グレースケールで行列として表し、特異値分解を用いて低ランク近似した結果である。オリジナルの画像は 3000 個の特異値を持つが、低ランク近似を行い 100 個、30 個にまで圧縮している。圧縮後の画像は少ない特異値で表現されるにも関わらず、オリジナルの画像の特徴をよく捉えていることがわかる。

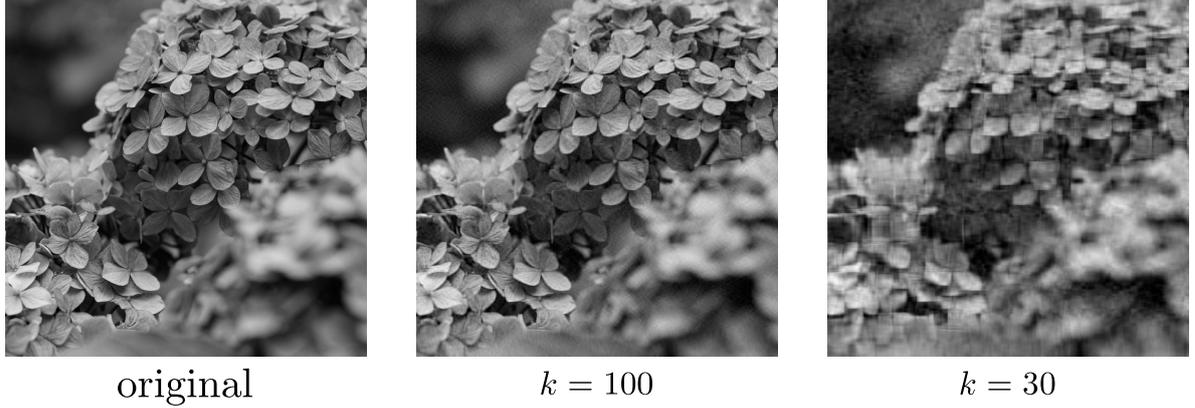


図 2.1: 特異値分解による画像圧縮の例。画像は筆者撮影。

2.1.3 高次特異値分解

高次特異値分解 (HOSVD: Higher Order Singular Value Decomposition) はテンソルに対する SVD の一般化として提唱された [24]。一般にテンソル A をコアテンソルと射影行列の積に分解する操作を Tucker 分解という。その一つとして HOSVD が存在する。HOSVD の性質を見るために、以下のような定義を行う。

以下、テンソル $A \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$ を成分が $a_{i_1 i_2 \dots i_N}$, ($1 \leq i_k \leq I_k$) であるテンソルとする。

Definition 2. Scaler Product

B を A と同じサイズのテンソルとする。このとき、内積は、

$$\langle A, B \rangle := \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} a_{i_1 i_2 \dots i_N}^* b_{i_1 i_2 \dots i_N} \quad (2.12)$$

で与えられる。このときフロベニウスノルムは、

$$\|A\|_F = \sqrt{\langle A, A \rangle} \quad (2.13)$$

で与えられる。

Definition 3. n -mode product

テンソル A と行列 $(u_{j_n, i_n}) = U \in \mathbb{C}^{J_n \times I_n}$ の n -mode product \times_n は、

$$(A \times_n U)_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{j_n i_n} \quad (2.14)$$

で与えられる。 n -mode product は以下の性質を満たす。

- (1) $(A \times_n F) \times_m G = (A \times_m G) \times_n F$, ($F \in \mathbb{C}^{J_n \times I_n}, G \in \mathbb{C}^{J_m \times I_m}$)
- (2) $(A \times_n F) \times_n G = A \times_n (GF)$, ($F \in \mathbb{C}^{J_n \times I_n}, G \in \mathbb{C}^{K_n \times J_n}$)

n -mode product はテンソルの各添え字方向に転置行列を書ける操作に対応する。
次に行列の n -mode 展開を導入する。

Definition 4. n -mode 展開行列

テンソル A を各モードについて行列表現したものを, $A_{(n)} \in \mathbb{C}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 \cdots I_{n-1})}$ を n -mode 展開行列と呼ぶ。この行列の成分は,

$$[A_{(n)}]_{jk} = A_{i_1 \dots i_n \dots i_N}, \quad \text{ただし } j = i_n, \quad k = 1 + \sum_{m \neq n} (i_m - 1) \prod_{\substack{l=1 \\ l \neq n}}^{m-1} I_l$$

で表される [25]。この時, n -mode rank を $\text{rank}_n A := \text{rank} A_{(n)}$ で定義する。

これらの定義を用いて, 次の高次特異値分解の定理が成り立つ。

Theorem 3. $A \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ に対し,

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} \quad (2.15)$$

なる分解が存在する。ただし,

(1) $U^{(n)} \in \mathbb{C}^{I_n \times I_n}$ のユニタリ行列

(2) $S \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ かつ,

- (a) $\langle S_{i_n=\alpha}, S_{i_n=\beta} \rangle = 0$, when $\alpha \neq \beta$
 (b) $\|S_{i_n=1}\|_F \geq \|S_{i_n=2}\|_F \geq \cdots \|S_{i_n=I_n}\|_F \geq 0$

が成り立つ。この分解を HOSVD と呼ぶ。

この定義は, SVD の持つ, 特異値が非負であるという性質を多次元に拡張したものになっている。また, HOSVD の別の表現として以下のようなものがある。

Theorem 4. n -mode 展開行列 $A_{(n)}$ が

$$A_{(n)} = U^{(n)} \Sigma^{(n)} V^{(n)\dagger}$$

と SVD されるとき

$$S = A \times_1 U^{(1)\dagger} \times_2 \cdots \times_N U^{(N)\dagger}$$

とすれば,

$$A = S \times_1 U^{(1)} \times_2 \cdots \times_N U^{(N)} \quad (2.16)$$

が成り立つ。 U の列の直交性と, 式 (2.4) の形から $\|S_{i_n=k}\|_F = \|A_{(n)}^{i_n=k}\|_F = \sigma_n^{(k)}$ となるので, 性質 (2a), (2b) が従う。ただし, $\sigma_k^{(n)}$ は $A_{(n)}$ の k 番目の特異値である。

例として, $T \in \mathbb{C}^{D \times D \times D \times D}$ のテンソルの HOSVD をダイアグラムで表すと以下ようになる。

$$T = U^{(2)} B U^{(1)} U^{(3)} U^{(4)} \quad (2.17)$$

SVD の時と異なり, HOSVD はテンソルの最良の近似ではないことが知られている. すなわち, $\text{rank}(R_1, R_2, \dots, R_N)$ のテンソル A に対し, $U^{(k)}$ の r_k 列目までを取った isometry $\hat{U}^{(k)} \in \mathbb{C}^{I_k \times r_k}$ を考えたとき, 近似されたテンソル \hat{A}

$$\hat{A} = S \times_1 \hat{U}^{(1)} \times_2 \cdots \times_N \hat{U}^{(N)}$$

は, $\text{rank}(r_1, r_2, \dots, r_N)$ のテンソルであり,

$$\|A - \hat{A}\|_F^2 \leq \sum_{i_1=r_1+1}^{R_1} (\sigma_{i_1}^{(1)})^2 + \sum_{i_2=r_2+1}^{R_2} (\sigma_{i_2}^{(2)})^2 + \cdots + \sum_{i_N=r_N+1}^{R_N} (\sigma_{i_N}^{(N)})^2 \quad (2.18)$$

を満たす. すなわち上限は与えるが最良の近似とは限らない. そこで iteration により局所的な近似解を求める方法が HOOI(Higher Order Orthogonal Iteration) である [26, 27].

Theorem 5. 与えられた $\text{rank}(R_1, \dots, R_N)$ のテンソル $A \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ に対し, A を近似する $\text{rank}(r_1, \dots, r_N)$ のテンソル \hat{A} を考える. \hat{A} のランクの条件より,

$$\hat{A} = B \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}, \quad B \in \mathbb{C}^{r_1 \times \cdots \times r_N}, U^{(1)} \in \mathbb{C}^{I_1 \times r_1}, \dots, U^{(N)} \in \mathbb{C}^{I_N \times r_N}$$

という分解が存在する. ただし U はすべての列が直交する isometry である. この時, 目的関数 $\|A - \hat{A}\|_F^2$ を最小化する B は, 与えられた U に対して最小二乗問題を解くことで,

$$B = A \times_1 U^{(1)\dagger} \times_2 \cdots \times_N U^{(N)\dagger} \quad (2.19)$$

であることが従う. この時

$$f(\hat{A}) = \|A - \hat{A}\|_F^2 \quad (2.20)$$

$$(2.21)$$

より, この最小化問題は

$$\min_{\substack{U^{(1)}, U^{(2)}, \dots, U^{(N)} \\ B}} f(\hat{A}) = \min_{U^{(1)}, U^{(2)}, \dots, U^{(N)}} \|A - A \times_1 (U^{(1)}U^{(1)\dagger}) \times_2 \cdots \times_N (U^{(N)}U^{(N)\dagger})\|_F^2 \quad (2.22)$$

と置き換えることができ, これは各 $U^{(n)}$ を固定し, 一方向の添え字のみに Theorem 2 を適用し, U を更新しながらノルムを最小化することで極小値が得られると期待される.

Algorithm 1 HOOI

input: $A \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$

output: $\text{rank}(r_1, \dots, r_N)$ の最良近似の $\hat{A} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$

initialize: $U^{(n)}, n = 1, \dots, N$ を HOSVD により求める.

repeat

for $n = 1, \dots, N$ **do**

$Y \leftarrow A \times_1 U^{(1)\dagger} \times_2 \cdots \times_{n-1} U^{(n-1)\dagger} \times_{n+1} U^{(n+1)\dagger} \cdots \times_N U^{(N)\dagger}$

$U^n \leftarrow Y_{(n)}$ をランク r の truncated SVD した際の $U^{(n)}$

end for

until Convergence of f

return $\hat{A} = A \times_1 (U^{(1)}U^{(1)\dagger}) \times_2 \cdots \times_N (U^{(N)}U^{(N)\dagger})$

注意すべき点として, HOOI はグローバルな最小値を与えるとは限らないということがある.

HOSVD と HOOI をカラー画像に適用したのが図 2.2 である。カラー画像は RGB の情報を持っているので、 $3 \times 3000 \times 3000$ のテンソルである。この画像に $\text{rank}(50, 50, 50)$ の HOSVD と HOOI を適用した。HOSVD も HOOI もテンソルの特徴をうまく捉えている。

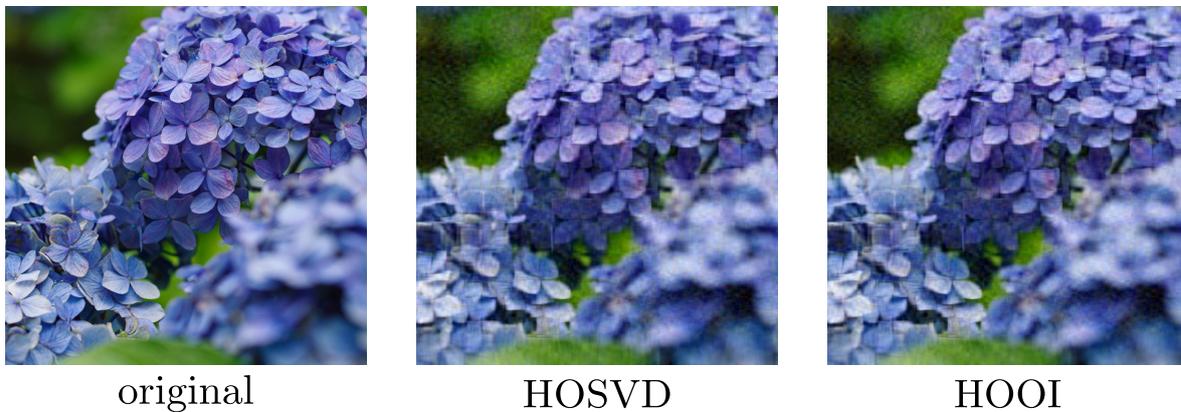


図 2.2: 高次特異値分解による画像圧縮の例。画像は筆者撮影。

2.2 テンソルネットワーク表現

2.2.1 分配関数のテンソルネットワーク表現

経路積分やボルツマン和で表された分配関数をテンソルネットワーク表示に変換する方法を考える。まず、次のような形の分配関数を考える。

$$Z = \int \mathcal{D}\phi e^{-S[\phi]}. \quad (2.23)$$

この $\int \mathcal{D}\phi$ は物理変数による和を取ると言う記号で、経路積分やボルツマン和を表す。例えばイジングモデルの場合は $\prod_n \sum_{\sigma_i}$ を表すし、スカラーの場理論では、 $\prod_n \int_{-\infty}^{\infty} d\phi_n$ を表す。以後、物理変数、量子場の理論、統計力学は区別せず、抽象的な場を ϕ と表す。変数はボソンのであるとし、フェルミオンやゲージ場は考えない。各時空点に ϕ_n という場があると考え、典型的な分配関数を

$$Z = \int \mathcal{D}\phi \prod_n \left[\exp \left(\sum_{\mu} W[\phi_n, \phi_{n+\hat{\mu}}] + K[\phi_n] \right) \right] \quad (2.24)$$

としてあらわす。ここで、 W は近接相互作用を表す関数であり、 K は外場の項である。これらの作用は並進対称性を有していると仮定する。ここで近接相互作用のボルツマン因子について次のような行列を定義する。

$$X_{\phi_n \phi_{n+\hat{\mu}}} = \exp(W[\phi_n, \phi_{n+\hat{\mu}}]) \quad (2.25)$$

ただし行列の添え字 ϕ は離散的になっているので、この時点で物理変数に対して何らかの正則化を入れる必要がある。例えば、Gauss-Hermite 求積法を用いる方法 [28] や、ゲージ場に対してはキャラクター展開などを用いることができる [29]。ただし、ゲージ場は格子上ではリンク上に存在するため、異なる構成の仕方となる。イジング模型の場合では、 X は、

$$X_{\sigma_n \sigma_{n+\hat{\mu}}} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix} \quad (2.26)$$

のような値をとる。

上記のようにして構成した X に対し、SVD を行う (図 2.3)。

$$X_{\phi_n \phi_{n+\hat{\mu}}} = \exp(W[\phi_n, \phi_{n+\hat{\mu}}]) = \sum_{i_{\mu}(n)} U_{\phi_n i_{\mu}(n)} S_{i_{\mu}(n) i_{\mu}(n)} V_{\phi_{n+\hat{\mu}} i_{\mu}(n)}^*. \quad (2.27)$$

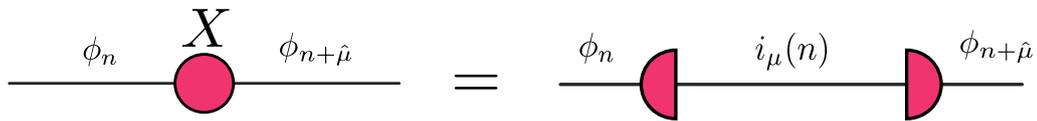


図 2.3: X の SVD.

ここで $i_{\mu}(n)$ は格子点をつなぐリンク上に定義した、ボンドと呼ばれる新たな自由度である。このようにして得た新しい自由度を残し、古い場の変数について和をとることで双対なテンソルネットワーク表現を得ることができる。基本となるテンソルを、

$$T_{i_1(n) \dots i_d(n) i_1(n-\hat{1}) \dots i_d(n-\hat{d})} = \sum_{\phi_n} e^{K[\phi_n]} \prod_{\mu} U_{\phi_n i_{\mu}(n)} V_{\phi_n i_{\mu}(n-\hat{\mu})}^* \sqrt{S_{i_{\mu}(n) i_{\mu}(n)}} \sqrt{S_{i_{\mu}(n-\hat{\mu}) i_{\mu}(n-\hat{\mu})}}, \quad (2.28)$$

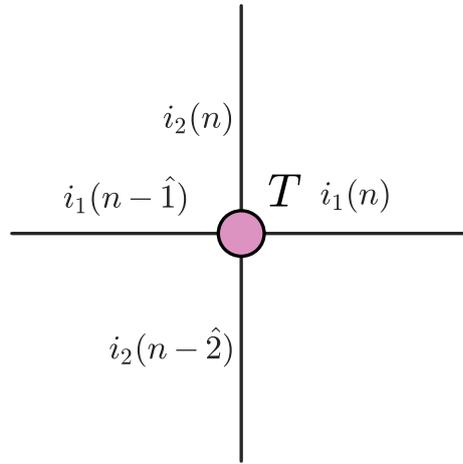


図 2.4: 二次元での基本テンソル $T_{i_1(n)i_2(n)i_1(n-\hat{1})i_2(n-\hat{2})}$ の図示.

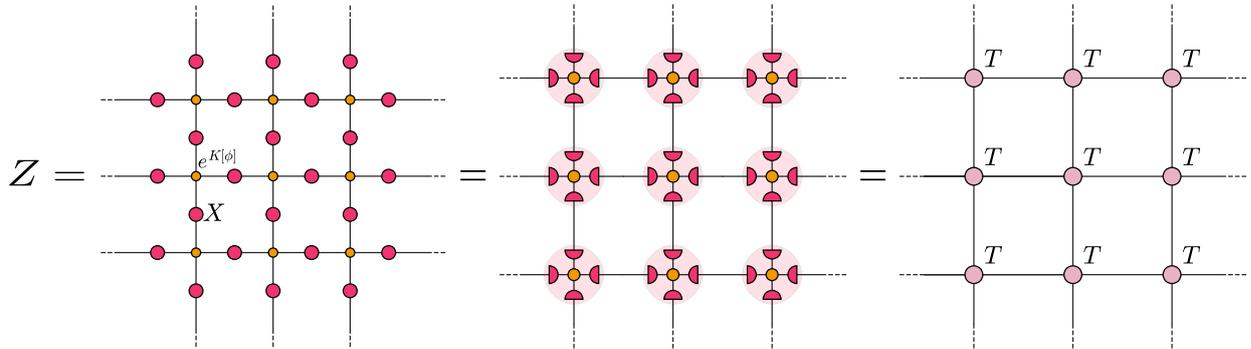


図 2.5: 分配関数のテンソルネットワーク表現. 点線はテンソルの繰り返しを表し, 境界同士のテンソルの添字は結ぶ.

のように定義する (図 2.5). この T は各時空点に配置され, 隣接する時空点とを結ぶ添字を持つ. 例えば, $i_1(n)$ は, 時空点 n から $n + \hat{1}$ をつなぐ添字, $i_d(n - \hat{d})$ は時空点 $n - \hat{d}$ から n をつなぐ添字である. ここで, 足の向きは時空の正方向に向かうように定義している. 二次元での基本テンソル T を図示すると, 図 2.4 のようになる. すると, 分配関数は,

$$Z = \text{tTr} \left[\sum_{\{i\}} \prod_n^V T_{i_1(n) \dots i_d(n) i_1(n-\hat{1}) \dots i_d(n-\hat{d})} \right] \quad (2.29)$$

というテンソルの縮約形式で表すことができる. ただし, $\sum_{\{i\}}$ はすべての添字について和を取ること, tTr は $i_1(n) = i_1(n + \hat{L})$ のように境界でテンソルの足をつなぐ操作を表す. このようなテンソルの縮約形式で表された図をテンソルネットワークと呼ぶ. この基本テンソル T は, 並進対称性及び回転対称性がある場合にはすべてのサイトで同一のものになることに注意する. 実際には $i_\mu(n)$ は非常に大きな値をとるので, 式 (2.27) においてボンド次元 D と呼ばれる有限のカットオフを入れ, 有効な自由度を制限する.

$$X_{\phi_n \phi_{n+\hat{\mu}}} = \exp(W[\phi_n, \phi_{n+\hat{\mu}}]) = \sum_{i_\mu(n)}^D U_{\phi_n i_\mu(n)} S_{i_\mu(n) i_\mu(n)} V_{\phi_{n+\hat{\mu}} i_\mu(n)}^* \quad (2.30)$$

次節以降で導入するテンソル線りこみ群と呼ばれる手法は、1 サイトの自由度^{体積}の成分を持つテンソルネットワークで表された分配関数を、足の大きさを有限のボンド次元 D で制限されたたった一つのテンソルで近似するというアルゴリズムである。ボンド次元 D の上限を設定しなければ、テンソルネットワーク表現は元の経路積分などで表された理論と全く等価である。

d 次元イジングモデルの場合、行列

$$W = \begin{pmatrix} \cosh \beta & \sinh \beta \\ \cosh \beta & -\sinh \beta \end{pmatrix} \quad (2.31)$$

を用いて、

$$T_{i_1(n) \dots i_d(n) i_1(n-1) \dots i_d(n-1) \dots i_1(n-d) \dots i_d(n-d)} = \sum_{\sigma_n} W_{\sigma_n i_1(n)} \cdots W_{\sigma_n i_d(n)} W_{\sigma_n i_1(n-1)} \cdots W_{\sigma_n i_d(n-d)} \quad (2.32)$$

と表せる。

2.2.2 相関関数のテンソルネットワーク表現

前節で導入した分配関数のテンソルネットワーク表現は容易に相関関数の計算に応用することができる。式 (2.24) と同様にして時空点 k に関する一点関数を考えると、

$$\langle \phi_k \rangle = \frac{1}{Z} \int D\phi \phi_k \prod_n \left[\exp \left(\sum_{\mu} W[\phi_n, \phi_{n+\hat{\mu}}] + K[\phi_n] \right) \right] \quad (2.33)$$

となる。これをテンソルネットワーク表現に変換すると、式 (2.28) の基本テンソルは、時空点 k のみ以下のような異なる impurity (不純物) テンソルとなる、

$$S_{i_1(k) \dots i_d(k) i_1(k-1) \dots i_d(k-1) \dots i_1(k-d) \dots i_d(k-d)} = \sum_{\phi_k} \phi_k e^{K[\phi_k]} \prod_{\mu} U_{\phi_k i_{\mu}(k)} V_{\phi_k i_{\mu}(k-\hat{\mu})}^* \sqrt{S_{i_{\mu}(k) i_{\mu}(k)}}} \sqrt{S_{i_{\mu}(k-\hat{\mu}) i_{\mu}(k-\hat{\mu})}}. \quad (2.34)$$

従って一点関数は、

$$\langle \phi_k \rangle = \frac{1}{Z} \text{tTr} \left[\sum_{\{i\}} S_{i_1(k) \dots i_d(k) i_1(k-1) \dots i_d(k-1) \dots i_1(k-d) \dots i_d(k-d)} \prod_{n \neq k} T_{i_1(n) \dots i_d(n) i_1(n-1) \dots i_d(n-d)} \right] \quad (2.35)$$

のように表される。図 2.6 に $\langle \phi_k \rangle$ のテンソルネットワークを示した。分子のネットワークは時空点 k を除いて分母の分配関数のテンソルネットワークと同じである。二点関数なども同様に、複数の impurity テンソルを含んだテンソルネットワークで表すことができる。たとえば、イジングモデルの 1 サイトあたりの内部エネルギー U は、impurity テンソルを

$$S_{i_1(k) \dots i_d(k) i_1(k-1) \dots i_d(k-1) \dots i_1(k-d) \dots i_d(k-d)} = \sum_{\sigma_n} \sigma_n W_{\sigma_n i_1(k)} \cdots W_{\sigma_n i_d(k)} W_{\sigma_n i_1(k-1)} \cdots W_{\sigma_n i_d(k-d)} \quad (2.36)$$

と定義して、

$$U = \frac{-d}{Z} \text{tTr} \left[\sum_{\{i\}} S_{i_1(k) \dots i_d(k) i_1(k-1) \dots i_d(k-1) \dots i_1(k-d) \dots i_d(k-d)} S_{i_1(k+1) \dots i_d(k+1) i_1(k) \dots i_d(k-d+1)} \prod_{n \neq k, k+1} T_{i_1(n) \dots i_d(n) i_1(n-1) \dots i_d(n-d)} \right] \quad (2.37)$$

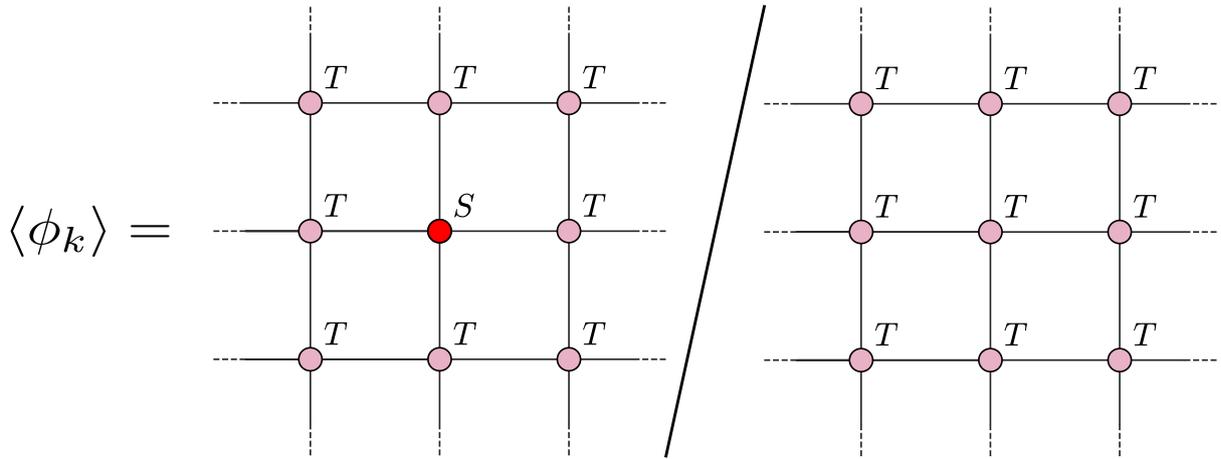


図 2.6: 一点関数のテンソルネットワーク表現. 点線はテンソルの繰り返しを表し, 境界同士のテンソルの添字は結ぶ.

と表せる. ここで $-d$ の因子は回転対称性によるものである.

ここまでの変形は, 元の理論をボンドと呼ばれる新しい自由度に置き換えることで等価なテンソルネットワーク表現に置き換えるものであった. しかしながら, 現実的には体積の数だけテンソルがつながっているようなネットワークを直接評価することは困難である. 例えば, 図 2.5 のようなテンソルネットワークを直接評価する場合, D^{2V} の計算量がかかる. これは分配関数を直接評価する場合とほぼ同じ計算量であり, 現実的ではない. しかし, この膨大なネットワークを, 重要な情報のみを残したまま, たった一つのテンソルで近似できるとしたら, 計算は容易になる. このような近似を行うための枠組みがテンソル繰り込み群である.

2.3 テンソル繰り込み群のアルゴリズム

分配関数のテンソルネットワーク表現は形式的な置き換えであり, その計算の困難さは変わらない. しかしそれぞれのテンソルに適切な近似を導入して, 分配関数の実効的なランクを下げ, 低次のテンソルで近似することができれば計算が可能になる. テンソル繰り込み群では SVD による高精度な分配関数の近似法を提案している. この節では, テンソル繰り込み群の基本的なアルゴリズム, 及びその高次元への拡張について説明する.

2.3.1 Levin-Nave TRG

式 (2.28) のテンソル縮約を直接実行するのは困難であるので, 近似的な縮約法を利用して, 分配関数を一つのテンソルで表すことを考える. Levin と Nave は基本テンソル T に対して局所的な特異値分解を実行することで, 実空間繰り込み群の操作を行ない, 高精度で 2 次元イジングモデルの自由エネルギーを得ることに成功した [5]. そのアルゴリズムを以下に示す.

まず, 基本テンソル T を次のように偶数サイトと奇数サイトでそれぞれ Truncated SVD する.

$$T_{i_1(n)i_2(n)i_1(n-1)i_2(n-2)} \simeq \begin{cases} \sum_{\alpha}^{\min(D, \text{size}(\alpha))} U_{i_1(n)i_2(n)\alpha}^{\text{even}} S_{\alpha\alpha}^{\text{even}} V_{i_1(n-1)i_2(n-2)\alpha}^{*\text{even}}, & (\text{偶数サイト}) \\ \sum_{\beta}^{\min(D, \text{size}(\beta))} U_{i_1(n-1)i_2(n)\beta}^{\text{odd}} S_{\beta\beta}^{\text{odd}} V_{i_1(n)i_2(n-2)\beta}^{*\text{odd}}, & (\text{奇数サイト}) \end{cases} \quad (2.38)$$

そして新たに $S^{[1]}, S^{[2]}, S^{[3]}, S^{[4]}$ を次のように定義する。

$$S_{i_1(n)i_2(n)\alpha}^{[1]} = U_{i_1(n)i_2(n)\alpha}^{\text{even}} \sqrt{S_{\alpha\alpha}^{\text{even}}} \quad (2.39)$$

$$S_{i_1(n-1)i_2(n-2)\alpha}^{[2]} = \sqrt{S_{\alpha\alpha}^{\text{even}}} V_{i_1(n-1)i_2(n-2)\alpha}^{\text{even}} \quad (2.40)$$

$$S_{i_1(n-1)i_2(n)\beta}^{[3]} = U_{i_1(n-1)i_2(n)\beta}^{\text{odd}} \sqrt{S_{\beta\beta}^{\text{odd}}} \quad (2.41)$$

$$S_{i_1(n)i_2(n-2)\beta}^{[4]} = \sqrt{S_{\beta\beta}^{\text{odd}}} V_{i_1(n)i_2(n-2)\beta}^{\text{odd}} \quad (2.42)$$

このとき、もともと添字 k_i は D^2 まで走るところ、ボンド次元 D に圧縮している点が重要である。これらのテンソル $S \in \mathbb{C}^{D \times D \times D}$ は、以下のフロベニウスノルム

$$\|T - S^{[1]}S^{[2]}\|_F \quad (2.43)$$

$$\|T - S^{[3]}S^{[4]}\|_F \quad (2.44)$$

を最小化していることに注意する。すなわちこのような分解は、局所的なテンソル T に着目した場合、最適な近似となっている。

この分解を 2×2 のサブネットワークについて図示したものが図 2.7 である。上記の分解は T の

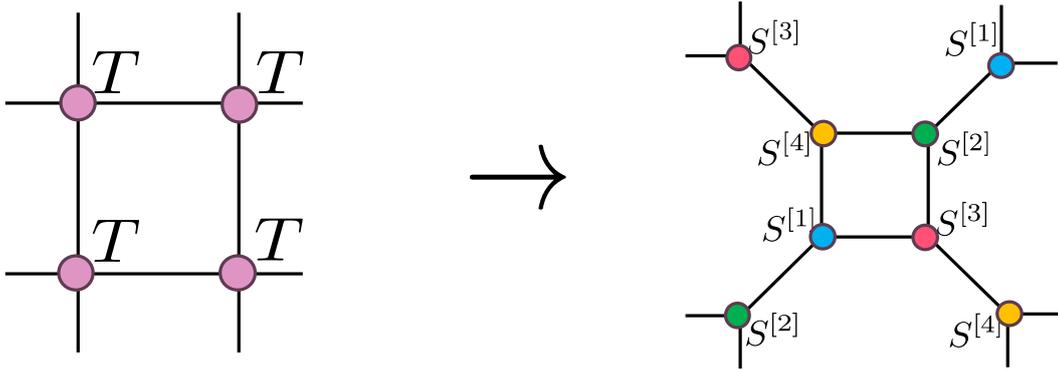


図 2.7: T の分解.

並進対称性を仮定しているため、局所的ではあるが、すべての時空点において同じ近似を行うことができることが大きな特徴である。

次に、繰り込まれたテンソルを得るために、

$$T_{k_1 k_2 k_3 k_4}^{(1)} = \sum_{i_1(n), i_2(n), i_1(n+2), i_2(n+1)} S_{i_1(n)i_2(n)k_1}^{[1]} S_{i_1(n), i_2(n+1)k_2}^{[3]} S_{i_1(n+2), i_2(n+1)k_3}^{[2]} S_{i_1(n+2), i_2(n)k_4}^{[4]} \quad (2.45)$$

のように 4 つのテンソルを一つのテンソルにまとめる縮約操作を行う (図 2.8)。すべてのサイトで同じ操作を行うと、ネットワークは $T^{(1)}$ で構成された格子状のテンソルネットワークとなる。 k の添字を新たな時空点の添字として再定義すると、分配関数は

$$Z = \text{tTr} \left[\sum_{\{k\}} \prod_n^{V/2} T_{k_1(n)k_2(n)k_1(n-1)k_2(n-2)}^{(1)} \right] \quad (2.46)$$

と表される。このとき、テンソルの足のサイズは D のまま一定に保たれながら、全体のテンソルの数は V 個から $V/2$ 個に削減されていることが大きな特徴である。図 2.9 に粗視化のフローを示した。

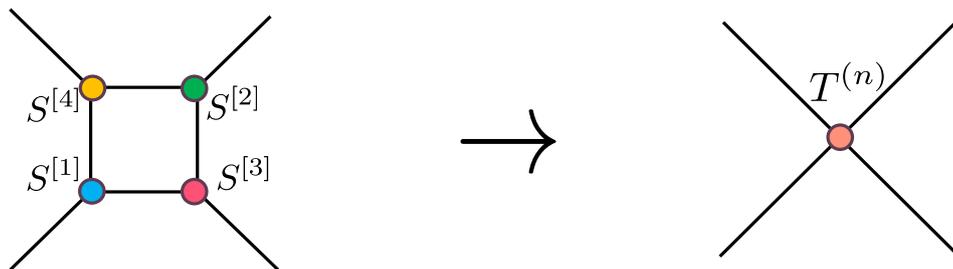


図 2.8: テンソルの縮約操作. 4つのテンソルから新たに繰り込まれたテンソルを作る. このとき足の大きさは変わらず D である.

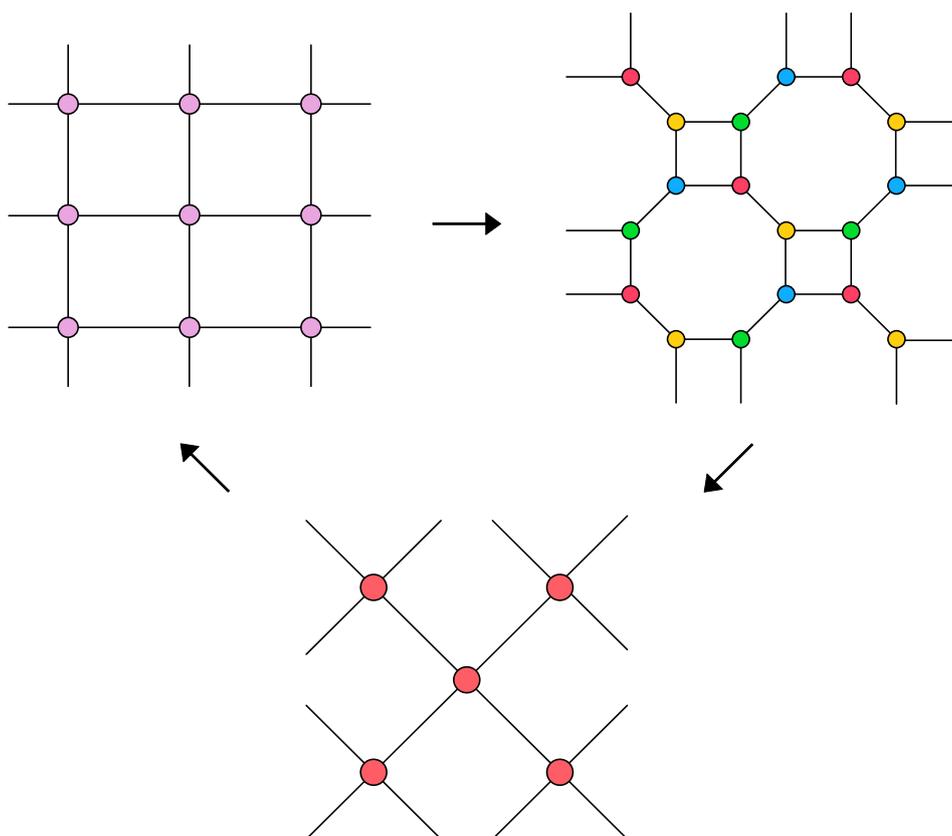


図 2.9: テンソル繰り込み群のフロー. 一度の粗視化ごとにテンソルの数は半分になる.

実際にすべてのサイトで同じ操作を行うわけではなく, 並進対称性により一度縮約を取れば, ”すべてのサイトで同じ縮約を行なったことになる”ことに注意する. この操作を何度も繰り返すと, 次のようにフローし,

$$T^{(1)} \rightarrow T^{(2)} \rightarrow \dots \rightarrow T^{(\log V)} \quad (2.47)$$

最終的に分配関数は一つのテンソル $T^{(\log V)}$ で表される.

$$Z = \sum_{i_1, i_2}^D T_{i_1 i_2 i_1 i_2}^{(\log V)} \quad (2.48)$$

ただし, $T^{(n)}$ は n 回目の反復後の繰り込まれたテンソルである. このようにして, V 個からなるテンソルネットワークを一つのテンソルで近似することができる. 全体にかかる計算コストとしては, 式 (2.38), 式 (2.45) の SVD と縮約操作がそれぞれ $O(D^6)$ を要する. また, 各ステップでテンソルの数は半分になるから, $\log V$ 回の反復を行えば, V 個のテンソルを一つに”繰り込んだことになる”³. すなわち全体の計算コストは $O(D^6) \log V$ となる. 実際には縮約を露わに取らずに RSVD や変分法を用いて, $O(D^5) \log V$ にまで計算コストを減らすことが可能である [30, 31]. テンソル繰り込み群法の最も大きな利点は, 計算コストが $\log V$ に比例するため, モンテカルロ法の V^β ($\beta > 0$) のコストに比べ, 熱力学極限を取ることが容易であるという特徴である. これにより有限体積効果の影響を受けず, 相転移現象の解析に非常に有利である. 次の利点として, すべての操作に確率論的操作が入っておらず, 特異値分解は直接複素数を扱えるので, 符号問題が存在しないという点が挙げられる.

注意すべき点として, TRG による近似はあくまでも局所的なテンソルに対し行うものであるので, 全系のフロベニウスノルム

$$\left\| \prod_n^V T - \left(\prod_{n:\text{even}} S^{[1]} S^{[2]} \right) \left(\prod_{n:\text{odd}} S^{[3]} S^{[4]} \right) \right\|_F^2 \quad (2.49)$$

を最小化するわけではないことに注意する. このような系全体を考慮した近似を行うには, テンソル T と周りの環境テンソルを最適化しなければならない. このような手法は第二テンソル繰り込み群 (SRG: Second Renormalization Group) と呼ばれる [32].

2.3.2 HOTRG

TRG は二次元古典系の分配関数などを求めるアルゴリズムであった. したがって二次元量子系や 3 次元系などへの拡張は非自明であった. Xie らは高次元に対し TRG 法を適用するためのアルゴリズムである, 高次テンソル繰り込み群 (HOTRG: Higher Order Tensor Renormalization Group) を提案した [12]. HOTRG の大きな特徴は, TRG とは対照的に, 一つの空間軸に着目し, 非等方的に HOSVD による繰り込みを行うところにある. そのアルゴリズムを以下に示す. まず, $\hat{1}$ 方向に着目し, 縦に T を 2 つ並べた以下のサブネットワークを考える (図 2.10).

$$\Gamma = \sum_{i_1(n)} T_{i_1(n+\hat{1}) \dots i_d(n+\hat{1}) i_1(n) \dots i_d(n+\hat{1}-\hat{d})} T_{i_1(n) \dots i_d(n) i_1(n-\hat{1}) \dots i_d(n-\hat{d})} \quad (2.50)$$

HOTRG では, isometry と呼ばれる打ち切られたユニタリ行列を用いて, $\hat{1}$ 方向以外の足を一本に繰り込む. isometry は, 次のような二つのフロベニウスノルムのうち, 最も下限を達成する $U \in \mathbb{C}^{D \times D \times D}$ または $V \in \mathbb{C}^{D \times D \times D}$ として定義される. ただし簡単のため, 2 次元の図を用いて説明する.

³無論, 得られた $T^{(\log V)}$ を使い新たに 2 回反復を行い $T^{(\log V+2)}$ を作成すれば, $4V$ の体積を繰り込んだことになる.

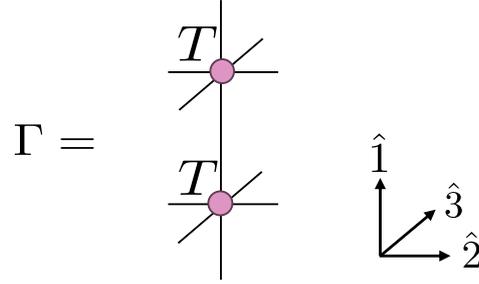


図 2.10: HOTRG で考えるサブネットワーク (3次元)

$$\epsilon_R = \left\| \left\| \begin{array}{c} T \\ T \end{array} \right. - \begin{array}{c} T \\ T \end{array} \begin{array}{c} U^\dagger \\ U \end{array} \right\| \right\| F$$

$$\epsilon_L = \left\| \left\| \begin{array}{c} T \\ T \end{array} \right. - \begin{array}{c} V \\ V^\dagger \end{array} \begin{array}{c} T \\ T \end{array} \right\| \right\| F$$

それぞれのフロベニウスノルムの下限は、 Γ に対し、それぞれのモードで SVD を行うことで、容易に達成できる。ただし、実際には頭に Γ を作ることはせず、 Γ のグラム行列を用いて計算する。例えば ϵ_R の場合、

$$\begin{aligned} & (\Gamma \Gamma^\dagger)_{i_2(n+\hat{1})i_2(n)\bar{i}_2(n+\hat{1})\bar{i}_2(n)} \\ &= \sum_{/i_2(n+\hat{1}), i_2(n), \bar{i}_2(n+\hat{1}), \bar{i}_2(n)} \left(T_{i_1(n+\hat{1})i_2(n+\hat{1})\dots i_d(n+\hat{1})i_1(n)\dots i_d(n+\hat{1}-\hat{d})} T_{i_1(n+\hat{1})\bar{i}_2(n+\hat{1})\dots i_d(n+\hat{1})i_1(n)\dots i_d(n+\hat{1}-\hat{d})}^* \right) \\ & \quad \times \left(T_{i_1(n)i_2(n)\dots i_d(n)i_1(n-\hat{1})\dots i_d(n-\hat{d})} T_{i_1(n)\bar{i}_2(n)\dots i_d(n)i_1(n-\hat{1})\dots i_d(n-\hat{d})}^* \right) \end{aligned} \quad (2.51)$$

$$\simeq \sum_{\delta}^D U_{i_2(n+\hat{1})i_2(n)\delta}^R \left(S_{\delta\delta}^R \right)^2 U_{\bar{i}_2(n+\hat{1})\bar{i}_2(n)\delta}^{*R} \quad (2.52)$$

のように SVD を行い、 $U = U^R$ を得る。同様の手順で $-\hat{2}$ 方向の足に対しても SVD を行い、 V を得る。これにより大きなメモリの Γ を作らずに SVD を行うことができる。このとき、打ち切り誤差は

$$\epsilon^R = \sqrt{\sum_{\delta=D+1}^{D^2} \left(S_{\delta\delta}^R \right)^2} \quad (2.53)$$

$$\epsilon^L = \sqrt{\sum_{\delta=D+1}^{D^2} \left(S_{\delta\delta}^L \right)^2} \quad (2.54)$$

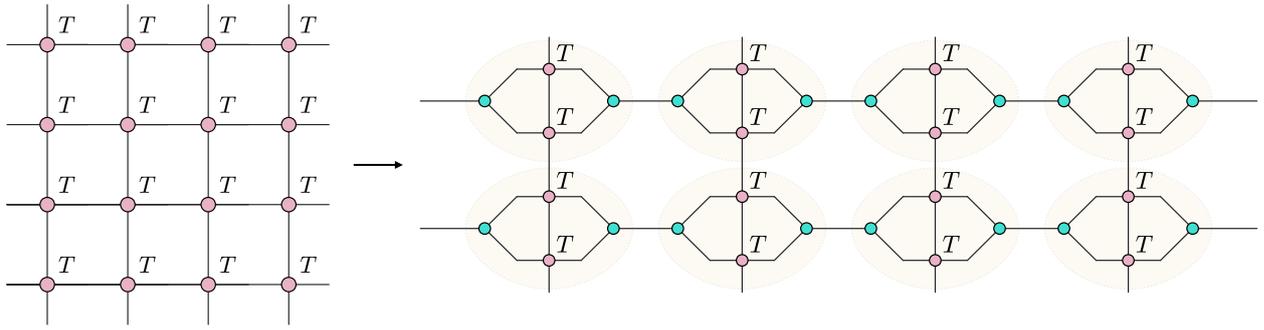


図 2.11: isometry を挿入した HOTRG のネットワーク. パッチは縮約を取る領域を表す.

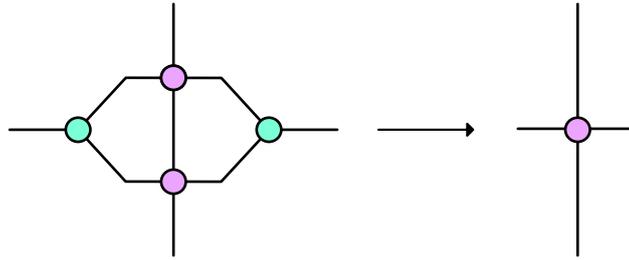


図 2.12: 2次元の HOTRG での縮約操作.

のように表される. これらの大小を比べ, $\epsilon^R > \epsilon^L$ なら isometry として V を選び, $\epsilon^R < \epsilon^L$ なら U を選ぶ. このようにして得た $\hat{2}$ 方向の isometry を $U^{(2)}$ と置く. 同様に $U^{(3)}, \dots, U^{(d)}$ を求める. このようにしてすべての時空点において

$$\Gamma \simeq \Gamma \times_{v_1} (U^{(2)}U^{(2)\dagger}) \times_{v_2} \dots \times_{v_d} (U^{(d)}U^{(d)\dagger}) \quad (2.55)$$

によって近似する. これは Γ の HOSVD になっていることに注意する. ただし v_m は \hat{m} 方向で $\epsilon^R < \epsilon^L$ なら $\{i_m(n + \hat{1}), i_m(n)\}$ のモード, それ以外なら $\{i_m(n + \hat{1} - \hat{m}), i_m(n - \hat{m})\}$ のモードを指す. このようにして, Γ の v_1, \dots, v_d のモードについて HOSVD による近似を行なったネットワークは図 2.11 のようになる. このようにすべてのサイトに isometry を挿入すると, 黄色のパッチで囲われた領域を縮約することができ,

$$\begin{aligned} T_{i_1(n+\hat{1})k_2 \dots k_d i_1(n)k'_2 \dots k'_d}^{\text{new}} &= \sum_{i_1(n+\hat{1}), i_1(n), k} T_{i_1(n+\hat{1})i_2(n+\hat{1}) \dots i_d(n+\hat{1})i_1(n) \dots i_d(n+\hat{1}-\hat{d})} T_{i_1(n) \dots i_d(n)i_1(n-\hat{1}) \dots i_d(n-\hat{d})} \\ &\times U_{i_2(n+\hat{1})i_2(n)k_2}^{*(2)} U_{i_2(n+\hat{1}-\hat{2})i_2(n-\hat{2})k'_2}^{(2)} \dots U_{i_d(n+\hat{1})i_d(n)k_d}^{*(d)} U_{i_d(n+\hat{1}-\hat{d})i_d(n-\hat{d})k'_d}^{(d)} \end{aligned} \quad (2.56)$$

と新たな繰り込まれたテンソル T^{new} を得る. ここで, 簡単のため v の方向は正方向になっているとした. この縮約を図 2.12 に示した. 図の通り, D^2 あったテンソルの足をプロジェクターを用いることにより, D に圧縮することができる. すなわち, 2つのテンソルを1つに繰り込んだことになる. この縮約操作で全系のテンソルの数は $\hat{1}$ 方向に半分になる. この操作を $\hat{2}, \dots, \hat{d}$ に対して順番に行えば, テンソルの数が $1/2^d$ になった超立方体を得られる. この繰り込み操作は, 高次元にも容易に拡張でき, 例えば, 3次元では図 2.13 のようになる. このとき, 計算コストのボトルネックとなるのは縮約ステップであり, $O(D^{4d-1})$ の計算量と基本テンソル T の保持コスト $O(D^{2d+1})$ がかかる.⁴ただし d は次元である. HOTRG では TRG と比べ大きな範囲のサブネットワーク Γ

⁴縮約の際に現れる中間テンソルは, T よりも多い足を持つが, 添字を分割して和を実行することでメモリ使用量は抑えられる [33].

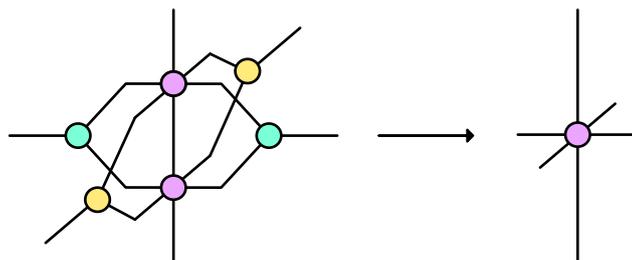


図 2.13: 3次元のHOTRGでの縮約操作.

を近似している。これは分配関数自体により近いネットワークの近似になっているため、TRGよりも同じボンド次元では高い精度を持つ。図 2.14 は温度に対する TRG と HOTRG の 2次元 Ising 模型の自由エネルギーの厳密解との相対誤差を比較したものである ([12] より引用)。HOTRG は TRG よりも厳密解との誤差が小さいことがわかる。

また、TRG, HOTRG 共に、臨界点付近 ($T \sim 2.27$) で精度が低下していることがわかる。この主な原因として、HOTRG や TRG では適切に近距離相関を除去できていない、という事が挙げられる。このような近距離相関を正しく考慮した繰り込みを行うアルゴリズムについては Appendix で解説する。

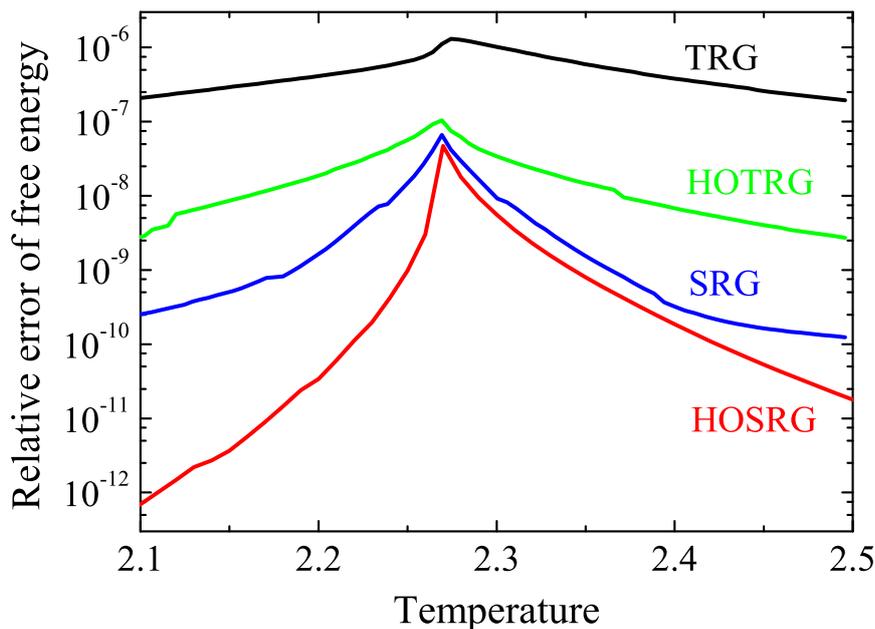


図 2.14: TRG と HOTRG の比較。 [12] より引用。HOTRG は TRG よりも同一のボンド次元における近似精度が高い。

2.3.3 impurity テンソルの入ったネットワークの粗視化

2.2.2 節で導入した impurity テンソルを含むネットワークの HOTRG による粗視化法を考える。このような粗視化を行う場合、impurity テンソルを基本テンソルと同時に繰り込むことで最終的に 1 つの impurity テンソルの状態に粗視化する。例えば 1 点関数のような 1 つの impurity テンソルが入ったネットワークを考える場合、図 2.15 のように HOTRG の縮約ステップで使用した

isometry を用いて, impurity テンソルと基本テンソルを一つずつ含んだようなサブネットワークを繰り込むことで, 各ステップごとに新たな impurity テンソルを得ることが可能である.

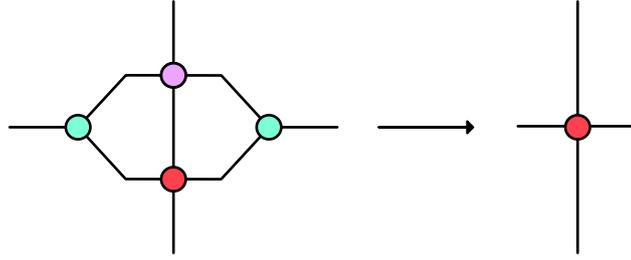


図 2.15: impurity テンソルを含むネットワークの粗視化. 赤丸は impurity テンソル.

この時, 図のような単純なアップデートを行う場合には, 基本テンソルで構成されたサブネットワークの縮約に加えて impurity テンソルを含んだサブネットワークの縮約も行うため, 計算量は 2 倍になる. さらに高次のモーメントを求める場合には, すべての縮約パターンを足し合わせるような操作が必要である [33]. また, impurity テンソルを用いず, 自動微分をにより物理量を求める方法も用いられている [34].

2.3.4 ATRG

HOTRG は高次元に TRG を拡張したアルゴリズムであったが, その計算コストの高さから 4 次元などでボンド次元を大きくする事が難しい. 高次元において計算コストが比較的小さく有用なアルゴリズムが ATRG (Anisotropic Tensor Renormalization Group) [14] である. この方法では, HOTRG に加え, 基本のテンソルの分解とボンド入れ替えという操作で成り立っている. HOTRG に比べ, 計算コストが $O(D^{2d+1})$, メモリコストが $O(D^{d+1})$ となり, 4 次元系での計算も可能である. オリジナルの方法ではメモリコスト及び計算コストが大きかったが, [35] などによりコストの削減が図られた. ここでは [36] の方法に従う. ATRG ではまず, 基本のテンソル $T_{i_1(n+1)i_2(n+1)\dots i_d(n+1)i_1(n)i_2(n+1-2)\dots i_d(n+1-d)}$ を HOTRG 同様に繰り込む方向を $\hat{1}$ 方向に固定し, 次のように分解する. ただし簡略化のために $i_k(n - \hat{k}) = j_k(n)$ とする.

$$T_{i_1(n+1)i_2(n+1)\dots i_d(n+1)j_1(n+1)j_2(n+1)\dots j_d(n+1)} \simeq \sum_{\alpha}^{\min(D, \text{size}(\alpha))} U_{i_1(n+1)i_2(n+1)\dots i_d(n+1)\alpha} S_{\alpha\alpha} V_{j_1(n+1)j_2(n+1)\dots j_d(n+1)\alpha}^* \quad (2.57)$$

となる. そしてサブネットワーク $\Gamma = TT$ を基本となるテンソル A, B, C, D を以下のように定義し書き換える (図 2.18 (a)).

$$A_{i_1(n+1)i_2(n+1)\dots i_d(n+1)\alpha} = U_{i_1(n+1)i_2(n+1)\dots i_d(n+1)\alpha} \quad (2.58)$$

$$B_{j_1(n+1)j_2(n+1)\dots j_d(n+1)\alpha} = S_{\alpha\alpha} V_{j_1(n+1)j_2(n+1)\dots j_d(n+1)\alpha}^* \quad (2.59)$$

$$C_{i_1(n)i_2(n)\dots i_d(n)\beta} = U_{i_1(n)i_2(n)\dots i_d(n)\beta} S_{\beta\beta} \quad (2.60)$$

$$D_{j_1(n)j_2(n)\dots j_d(n)\beta} = V_{j_1(n)j_2(n)\dots j_d(n)\beta}^* \quad (2.61)$$

とおく. A, D はユニタリになっていることに注意する. この基本ネットワークの形式では, 初期テンソルのサイズが小さい場合, メモリ $O(D^{2d})$ が必要にならず大幅なメモリ削減が可能となっている. $d = 4$ で $D = 50$ と取った場合, A は約 2.3GB のメモリを必要とする. この事実はこの

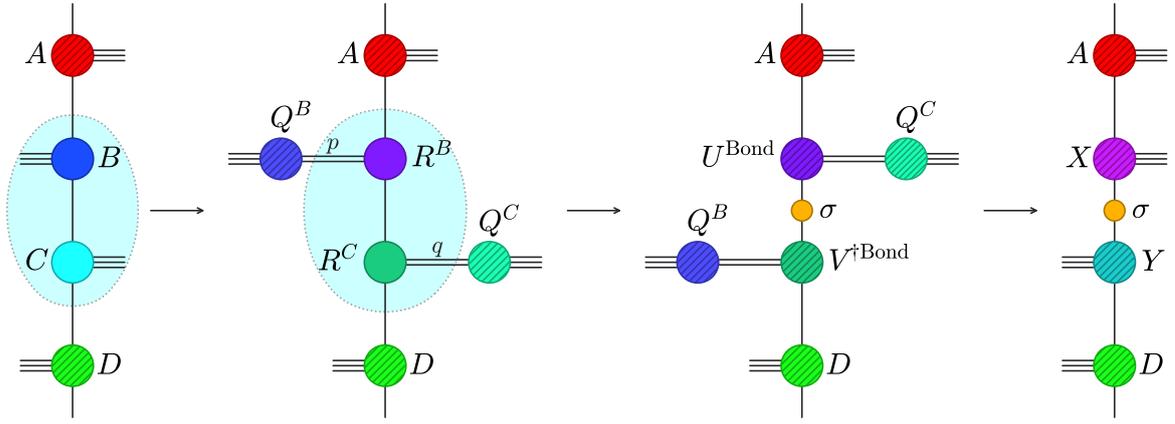


図 2.16: ボンド入れ替え. 斜線はアイソメトリーを表す.

に GPU 演算を行う際に重要である. 次にボンド入れ替えと呼ばれる操作を行う. そのために, B と C に対し次のように QR 分解を行う.

$$B_{j_1(n+\hat{1})j_2(n+\hat{1})\dots j_d(n+\hat{1})\alpha} = \sum_p^{\min(D^2, D^{d-1})} Q_{j_2(n+\hat{1})\dots j_d(n+\hat{1})p}^B R_{j_1(n+\hat{1})\alpha p}^B \quad (2.62)$$

$$C_{i_1(n)i_2(n)\dots i_d(n)\beta} = \sum_q^{\min(D^2, D^{d-1})} Q_{i_2(n)\dots i_d(n)q}^C R_{i_1(n)\beta q}^C \quad (2.63)$$

ここで R^B と R^C の縮約をとり,

$$M_{\alpha p \beta q} = \sum_{j_1(n+\hat{1})=i_1(n)}^D R_{j_1(n+\hat{1})\alpha p}^B R_{i_1(n)\beta q}^C \quad (2.64)$$

に対し SVD,

$$M_{\alpha p \beta q} \simeq \sum_{\gamma}^D U_{\alpha q \gamma}^{\text{Bond}} \sigma_{\gamma \gamma} V_{\beta p \gamma}^{*\text{Bond}} \quad (2.65)$$

を行い, $(\alpha p)(\beta q) \rightarrow (\alpha q)(\beta p)$ という足の入れ替えを行う. しかしこの M は $O(\min(D^6, D^{2d}))$ のメモリコストを必要とするので, $2 \leq d \leq 4$ では基本テンソルのメモリコスト $O(D^{d+1})$ を超えてしまうため許容できない. そこで RSVD を用いてメモリコストを減らしつつ, スワッピングを行う. まず,

$$Y_{\alpha q m} = \sum_{j_1(n+\hat{1})=i_1(n), \beta, p}^{D, D, D^2} R_{i_1(n)\beta q}^C \left(R_{j_1(n+\hat{1})\alpha p}^B \Omega_{\beta p m} \right) \quad (2.66)$$

と縮約する. ここで $\Omega \in \mathbb{R}^{D \times D^2 \times (D+l)}$ のランダムテンソルであり, l はオーバーサンプリングパラメーターである. ここでは [16] に従い, $l = O(D)$, $rD = D + l$ と取る. さらに Y を QR 分解し, $Q_{\alpha q m'}$ を得る. Range finder の操作を

$$Y'_{\beta p m'} = \sum_{j_1(n+\hat{1})=i_1(n), \alpha, q}^{D, D, D^2} R_{j_1(n+\hat{1})\alpha p}^{*B} \left(R_{i_1(n)\beta q}^{*C} Q_{\alpha q m'} \right) \quad (2.67)$$

とし、 Y' を QR 分解し、 $Q'_{\beta pm}$ を作成する。これを式 (2.66) の Ω に置き換える。この操作を q 回繰り返す、

$$\sum_{j_1(n+\hat{1})=i_1(n)}^D R_{j_1(n+\hat{1})\alpha p}^B R_{i_1(n)\beta q}^C \simeq \sum_{j_1(n+\hat{1})=i_1(n), \alpha', q', m'} Q_{\alpha q m'} \left\{ \left(Q_{\alpha' q' m'}^* R_{i_1(n)\beta q'}^C \right) R_{j_1(n+\hat{1})\alpha' p}^B \right\} \quad (2.68)$$

$$\simeq \sum_{m' \gamma}^{rD, D} Q_{\alpha q m'} \tilde{U}_{m' \gamma} \sigma_{\gamma \gamma} V_{\beta p \gamma}^{*\text{Bond}} \quad (2.69)$$

$$= \sum_{\gamma}^D U_{\alpha q \gamma}^{\text{Bond}} \sigma_{\gamma \gamma} V_{\beta p \gamma}^{*\text{Bond}} \quad (2.70)$$

と RSVD を行う。これらの操作は式 (2.65) において M を RSVD することと等価である。その後基本テンソル BC を

$$X_{\alpha i_2(n) \dots i_d(n) \gamma} = \sum_q^{\min(D^2, D^{d-1})} U_{\alpha q \gamma}^{\text{Bond}} Q_{i_2(n) \dots i_d(n) q}^C \quad (2.71)$$

$$Y_{\beta j_2(n+\hat{1}) \dots j_d(n+\hat{1}) \gamma} = \sum_p^{\min(D^2, D^{d-1})} V_{\beta p \gamma}^{*\text{Bond}} Q_{j_2(n+\hat{1}) \dots j_d(n+\hat{1}) p}^B \quad (2.72)$$

として、 $X\sigma Y$ と置き換える。これにより正の方向と負の方向の足が隣り合うようになる。ボン入れ替えの様子を図 2.16 に示した。ボン入れ替えの操作は、計算量は $O(qr \min(D^6, D^{d+3}))$ であり、メモリコストは $O(rD^4)$ である。このサブネットワーク $\Gamma = AX\sigma YD$ は、 σ 以外のテンソルが isometry になっており、Canonical form [37] になっていることに注意する。

次に $\hat{1}$ 方向に繰り込むために、これらを用いて、コスト関数

$$\|\Gamma\Gamma - \Gamma E^{(2)} F^{(2)} \Gamma\|_F \quad (2.73)$$

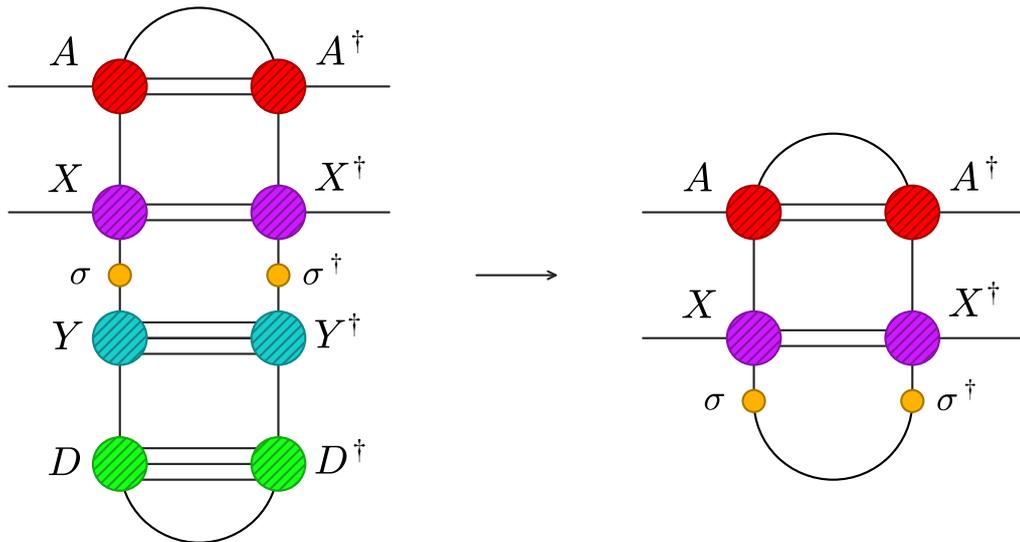


図 2.17: Canonical Form. Y, D は isometry なので、グラム行列は単位行列になる。これにより $AX\sigma YD$ のグラム行列は $AX\sigma$ のグラム行列に相当する。

を最小化する Squeezer, $E^{(2)}, F^{(2)} \in \mathbb{C}^{D \times D \times D}$ を導入する. 簡単のため $\hat{2}$ 方向の Squeezer の求め方を考える. まず, 次のように $AX\sigma$ と σYD の Reduced density matrix(グラム行列) を求める. Canonical form の導入により, これは Γ 自体のグラム行列に相当する. (図 2.17) これにより AX のアイソメトリーは Γ 自体の最適なアイソメトリーになる. HOTRG とは異なり, まず次のように打ち切っていない isometry, U^{AX} と特異値 S^{AX} を EVD(SVD) を用いて求める⁵.

$$\begin{aligned} & (AX\sigma A^* X^* \sigma^*)_{i_2(n+\hat{1})i_2(n)\bar{i}_2(n+\hat{1})\bar{i}_2(n)} \\ &= \sum_{/i_2(n+\hat{1}),i_2(n),\bar{i}_2(n+\hat{1}),\bar{i}_2(n)} \left(A_{i_1(n+\hat{1})i_2(n+\hat{1})\dots i_d(n+\hat{1})\alpha} A_{i_1(n+\hat{1})\bar{i}_2(n+\hat{1})\dots i_d(n+\hat{1})\bar{\alpha}}^* \right) \\ & \quad \times \left(X_{\alpha i_2(n)\dots i_d(n)\gamma} \sigma_{\gamma\gamma} \sigma_{\bar{\gamma}\bar{\gamma}} X_{\bar{\alpha} i_2(n)\dots i_d(n)\bar{\gamma}}^* \right) \end{aligned} \quad (2.74)$$

$$= \sum_{\delta}^{D^2} U_{i_2(n+\hat{1})i_2(n)\delta}^{AX\sigma} \left(S_{\delta\delta}^{AX\sigma} \right)^2 U_{i_2(n+\hat{1})\bar{i}_2(n)\delta}^{*AX\sigma} \quad (2.75)$$

ただし, / は except for を表す. σYD に対しても同様に U^{YD} を求める. 式 (2.73) のフロベニウスノルムの下限を達成する Squeezer は, $AX\sigma\sigma YD$ の間接的 SVD から,

$$\begin{aligned} & (AX\sigma\sigma YD)_{i_1(n+\hat{1})\dots i_d(n+\hat{1})\dots i_d(n)\gamma j_1(n+\hat{2})\dots j_d(n+\hat{2})\dots j_d(n+\hat{1}+\hat{2})\gamma'} \\ &= \sum (AX\sigma)_{i_1(n+\hat{1})i_2(n+\hat{1})\dots i_d(n+\hat{1})i_2(n)\dots i_d(n)\gamma} U_{i_2(n+\hat{1})i_2(n)\delta}^{*AX\sigma} S_{\delta\delta}^{-1AX\sigma} \left(S_{\delta\delta}^{AX\sigma} U_{i_2(n+\hat{1})\bar{i}_2(n)\delta}^{AX\sigma} \right. \\ & \quad \left. U_{i_2(n+\hat{1})\bar{i}_2(n)\delta'}^{\sigma YD} S_{\delta'\delta'}^{\sigma YD} \right) S_{\delta'\delta'}^{-1\sigma YD} U_{j_2(n+\hat{2})j_2(n+\hat{1}+\hat{2})\delta'}^{*\sigma YD} (\sigma YD)_{j_1(n+\hat{2})j_2(n+\hat{2})\dots j_d(n+\hat{2})j_2(n+\hat{1}+\hat{2})\dots j_d(n+\hat{1}+\hat{2})\gamma'} \end{aligned} \quad (2.76)$$

とし,

$$\sum_{\bar{i}_2(n+\hat{1}),\bar{i}_2(n)} S_{\delta\delta}^{AX\sigma} U_{i_2(n+\hat{1})\bar{i}_2(n)\delta}^{*AX\sigma} U_{i_2(n+\hat{1})\bar{i}_2(n)\delta'}^{\sigma YD} S_{\delta'\delta'}^{\sigma YD} \simeq \sum_{k_2}^D P_{\delta k_2} S_{k_2 k_2}^{PQ} Q_{\delta' k_2} \quad (2.77)$$

と SVD による近似を行い,

$$E_{i_2(n+\hat{1})i_2(n)k_2}^{(2)} = \sum_{\delta} U_{i_2(n+\hat{1})i_2(n)\delta}^{*AX\sigma} S_{\delta\delta}^{-1AX\sigma} P_{\delta k_2} \sqrt{S_{k_2 k_2}^{PQ}} \quad (2.78)$$

$$F_{j_2(n+\hat{2})j_2(n+\hat{1}+\hat{2})k_2}^{(2)} = \sum_{\delta'} \sqrt{S_{k_2 k_2}^{PQ}} Q_{\delta' k_2} S_{\delta'\delta'}^{-1\sigma YD} U_{j_2(n+\hat{2})j_2(n+\hat{1}+\hat{2})\delta'}^{*\sigma YD} \quad (2.79)$$

とすることで定義できる⁶. 同様の手順で $E^{(3)}, F^{(3)}, \dots, E^{(d)}, F^{(d)}$ を全て求める. この手順での計算コストは, 式 (2.91) の縮約が $O(D^{d+3})$, SVD が $O(D^6)$ である. SVD のコストは, RSVD により $O(D^5)$ に減らすことができるので, $d=2$ でも計算コストはボトルネックにならない. 高次元での RSVD は大きな系統誤差を生むので, $d \geq 3$ では通常の SVD を使用する. そして繰り込み群の手順として,

⁵ Γ^\dagger はグラム行列であるので EVD と SVD は等価であるが, SVD の方が数値的に安定である. 特異値はライブラリに依存しない結果を与えるカットオフを導入している. グラム行列を経由して特異値を求めた場合, 精度は単精度まで落ちることがある. しかし 4 次元ではそれ以上の精度で計算することがそもそも難しいため, 問題にはならない.

⁶この E,F がフロベニウスノルムの下限を達成していることは, $AX\sigma = U^{AX\sigma} S^{AX\sigma} V^{\dagger AX\sigma}$ と SVD され, 式 (2.76) において, $U^{\dagger AX\sigma} S^{-1AX\sigma}$ を掛けることで, $V^{\dagger AX\sigma}$ が残り, 式 (2.77) が間接的な Truncated SVD を行っているということにより理解できる. このような隣接するサブネットワークのフロベニウスノルムを最小化するような射影テンソルは, Oblique projector と呼ばれ, MPS において内線のボンド次元を圧縮するために [38] で, また TRG においてはスピングラス系を適切に取り扱うため [39], また境界条件のある系を適切に取り扱うため [40] において導入された. Oblique projector は初期テンソルが対称的でなくても精度を保つことが示されている [41]. 対称的なプロジェクターの作り方は, [42] において考案されている.

$$G_{i_1 k_2 \dots k_d \gamma} = \sum_{/\gamma, i_1, k} A_{i_1 i_2(n+1) \dots i_d(n+1) \alpha} X_{\alpha i_2(n) \dots i_d(n) \gamma} \sigma_{\gamma \gamma} E_{i_2(n+1) i_2(n) k_2}^{(2)} \dots E_{i_d(n+1) i_d(n) k_d}^{(d)} \quad (2.80)$$

$$H_{j_1 k'_2 \dots k'_d \gamma} = \sum_{/\gamma, j_1, k'} D_{j_1 j_2(n) \dots j_d(n) \beta} Y_{\beta j_2(n+1) \dots j_d(n+1) \gamma} F_{j_2(n) j_2(n+1) k'_2}^{(2)} \dots F_{j_d(n) j_d(n+1) k'_d}^{(d)} \quad (2.81)$$

という縮約をとる。この縮約コストは $O(D^{2d+1})$ であり、 $d \geq 3$ ではボトルネックとなる⁷。またメモリコストは中間テンソルに対して添え字を分割して縮約をとるループブロッキング [30] を用いることで $O(D^{d+1})$ に抑えることが可能である。

$$T_{i_1(n+1) i_2(n+1) \dots i_d(n+1) j_1(n+1) j_2(n+1) \dots j_d(n+1)}^{\text{new}} = \sum_{\gamma}^D G_{i_1(n+1) i_2(n+1) \dots i_d(n+1) \gamma} H_{j_1(n+1) j_2(n+1) \dots j_d(n+1) \gamma} \quad (2.82)$$

とすることで、 $\hat{1}$ 方向の繰りこみが完了する。実際にはあらわに T^{new} を作ることはせず、

$$G_{i_1 k_2 \dots k_d \gamma} = \sum_{\tau}^D Q_{i_1 k_2 \dots k_d, \tau}^G R_{\tau \gamma}^G \quad (2.83)$$

$$H_{j_1 k'_2 \dots k'_d \gamma} = \sum_{\tau'}^D Q_{j_1 k'_2 \dots k'_d, \tau'}^H R_{\tau' \gamma}^H \quad (2.84)$$

とし、

$$K_{\tau \tau'} = \sum_{\gamma}^D R_{\tau \gamma}^G R_{\tau' \gamma}^H = \sum_{\alpha}^D U_{\tau \alpha}^K S_{\alpha \alpha}^K V_{\tau' \alpha}^{*K} \quad (2.85)$$

のように分解し、新たに A, B, C, D を

$$A_{i_1(n+1) i_2(n+1) \dots i_d(n+1) \alpha} = \sum_{\tau}^D Q_{i_1(n+1) i_2(n+1) \dots i_d(n+1), \tau}^G U_{\tau \alpha}^K \quad (2.86)$$

$$B_{j_1(n+1) j_2(n+1) \dots j_d(n+1) \alpha} = \sum_{\tau'}^D Q_{j_1(n+1) j_2(n+1) \dots j_d(n+1), \tau'}^H S_{\alpha \alpha}^K V_{\tau' \alpha}^{*K} \quad (2.87)$$

$$C_{i_1(n) i_2(n) \dots i_d(n) \beta} = \sum_{\tau}^D Q_{i_1(n) i_2(n) \dots i_d(n), \tau}^G U_{\tau \beta}^K S_{\beta \beta}^K \quad (2.88)$$

$$D_{j_1(n) j_2(n) \dots j_d(n) \beta} = \sum_{\tau'}^D Q_{j_1(n) j_2(n) \dots j_d(n), \tau'}^H V_{\tau' \beta}^{*K} \quad (2.89)$$

と定義することで基本テンソルに戻ることができる。以上が ATRG の繰りこみ手順である。ATRГ は常に隣り合ったテンソル同士で分解、縮約が行われるため計算コストが小さい。しかしこれは局所的な分解を行っていることを意味しているわけではなく、すべての分解が Γ に対する SVD となっていることに注意する⁸。ATRГ 法はボンド入れ替えの操作が $D^d \rightarrow D$ 個の特異値の切り捨てを行っているため、HOTRG 法に比べ同じ D での近似精度は下がるが、同一計算時間では HOTRG に比べ低い自由エネルギーを達成する [14]。

⁷式 (2.78, 2.79) の最適な縮約順は次元により異なるのであらわに書かない。例えば 4 次元では、 $i_2(n+1), i_3(n), \alpha, i_2(n), i_3(n+1), i_4(n+1), i_4(n)$ の順に縮約を取り、 $i_4(n+1), \alpha$ を分割するとよい。

⁸一見ボンド入れ替えの操作は局所的な分解のように見えるが、 A, D が isometry なために、 Γ 自体の間接的 SVD になっている。

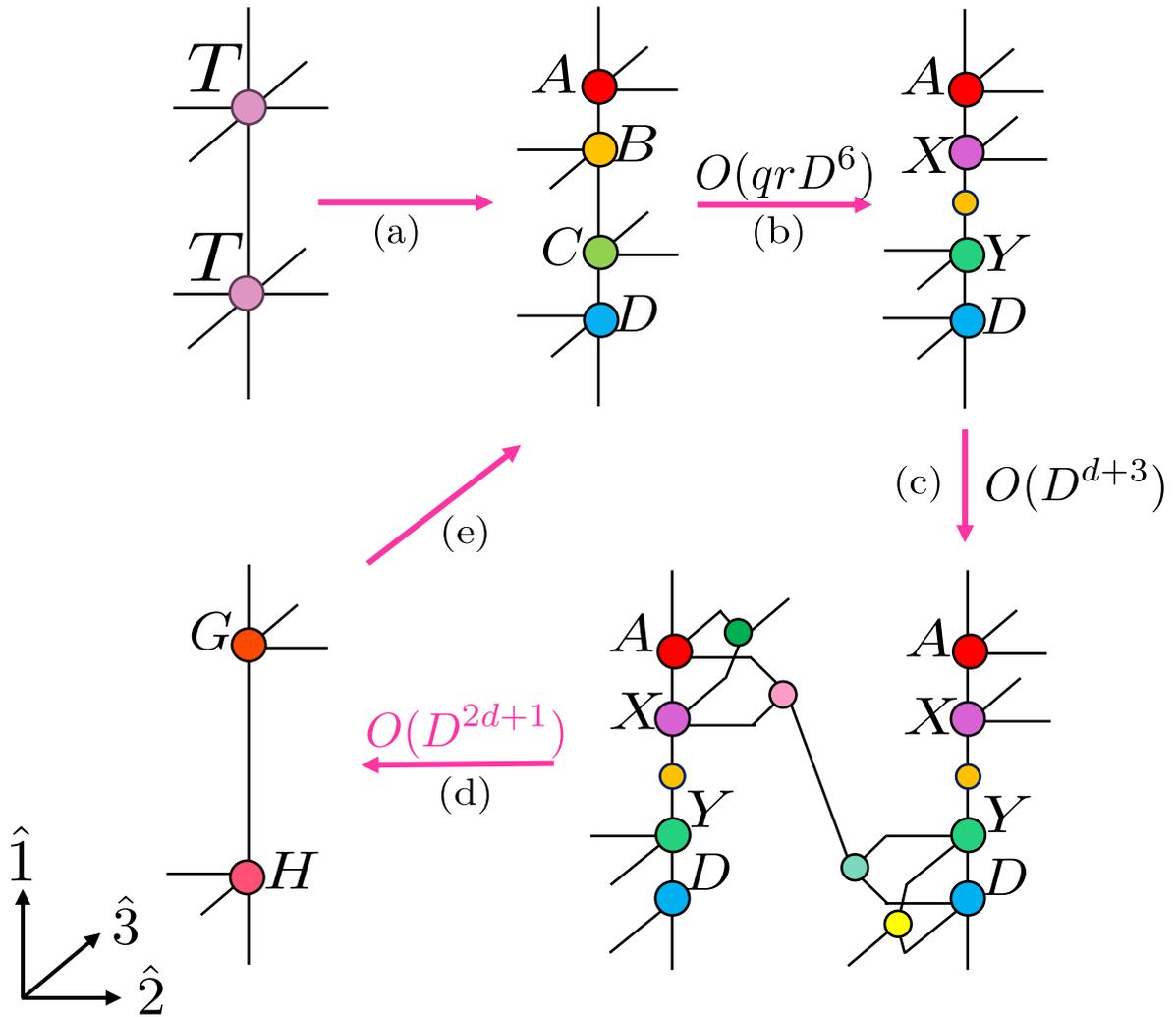


図 2.18: ATRG の一連の操作を表した図. (a) 基本テンソルの分解. (b) ボンド入れ替え. (c)sqeeaeer の導入. (d) 縮約. (e) 次のステップの基本テンソルを作成.

2.3.5 Triad-TRG 及び MDTRG

ATRG は HOTRG のサブネットワークに対して追加の近似を行い、計算コストを削減した手法であった。このような計算の軽量化は様々なものが存在する。その一つが Triad の手法である [43]。Triad TRG (TTRG) では、サブネットワークを構成するすべてのテンソルを SVD を用いて次のように 3 本足のテンソルに分解する (図 2.19 (a))。

$$T_{i_1(n)i_2(n)\dots i_d(n)j_1(n)j_2(n)\dots j_d(n)} = A_{\alpha_{d-2}i_d(n)\alpha_{d-1}}^{(d-1)} \cdots A_{i_2(n)i_3(n)\alpha_2}^{(2)} A_{\alpha_1 i_1(n)\alpha_{d-1}}^{(1)} \\ \times B_{\beta_{d-2}j_d(n)\beta_{d-1}}^{(d-1)} \cdots B_{j_2(n)j_3(n)\beta_2}^{(2)} B_{\alpha_1 j_1(n)\beta_{d-1}}^{(1)} \quad (2.90)$$

このようにして近似したサブネットワーク Γ を使用して、HOTRG 同様に isometry を求め、縮約を行う。このとき、縮約の過程で局所的なテンソルに対して RSVD を行いながら縮約することで、計算コストは $O(D^{d+3})$ にまで抑えられる。TTRG は 3 次元において、HOTRG や ATRG と比べ最も早く自由エネルギーが収束するという結果が得られている [43]。しかしながら、TTRG は局所的な分解を繰り返し、計算コストを減らすという性質上、HOTRG より系統誤差を多く含む。

このような問題に対し、Triad 表現のオーバーサンプルとサブネットワークの SVD という手法を用いて系統誤差を減らしたアルゴリズムが、MDTRG 及び Triad-MDTRG (Minimally Decomposed

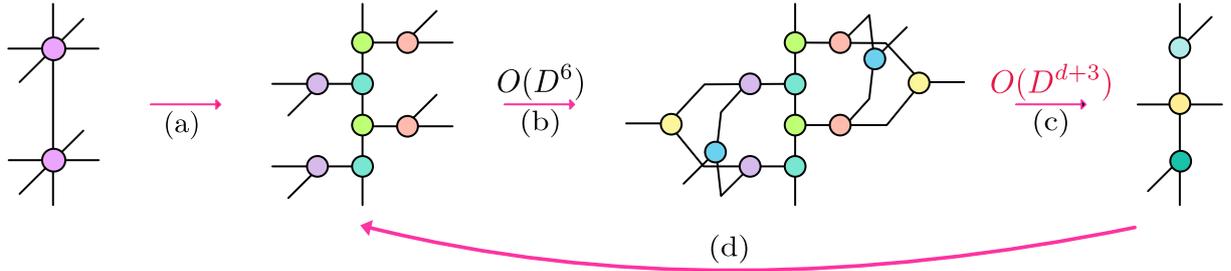


図 2.19: 3次元での Triad TRG のアルゴリズム. (a) 基本テンソルの分解. (b) isometry の導入. (c) 縮約. (d) 次のステップの基本テンソルを作成.

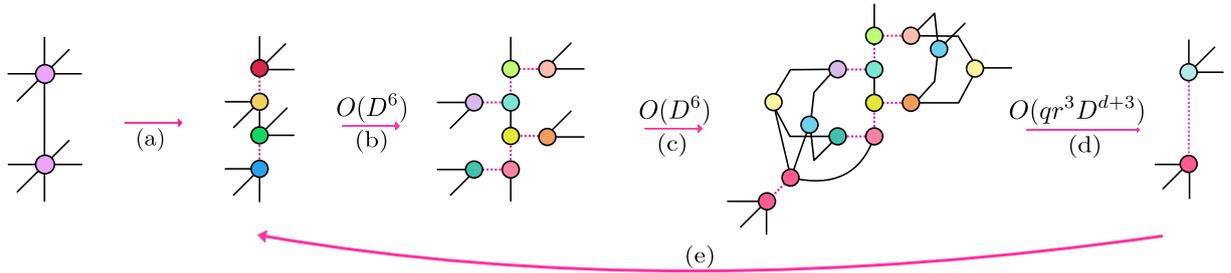
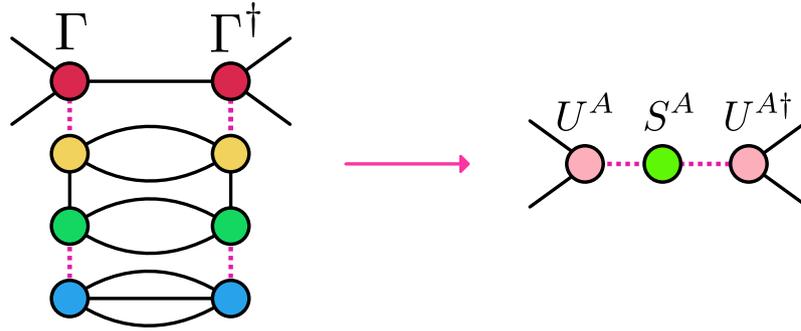


図 2.20: 3次元での Triad-MDTRG のアルゴリズム. 点線は rD にオーバーサンプルされている. (a) 基本テンソルの分解. (b) Triad 表現の作成. (c) isometry の導入. (d) RSVD を用いた縮約. (e) 次のステップの基本テンソルを作成.

TRG) である [21, 44]. Triad-MDTRG では, Triad 表現を作るステップ, isometry を求めるステップ, 縮約ステップで行う SVD を, すべてサブネットワーク Γ (これを Unit-Cell と呼ぶ) の分解として考えることで, 系統誤差を減らす. 以下に三次元での Triad-MDTRG アルゴリズムの概略を示す. MDTRG では, ATRG と同様に, 基本テンソルを $d+1$ 本足のテンソル A, B, C, D に分解する (図 2.20(a)). この時, A と B , C と D の間を結ぶポンドは rD にオーバーサンプルする. すなわち, 内線のみ r 倍の特異値を使用している. ここで r はオーバーサンプリングパラメータと呼ばれる. 次に, Γ のグラム行列から, Triad 表現を作成する (図 2.20(b)). 具体的には, Γ のグラム行列 (図 2.21) の SVD から作られる打ち切られた isometry, U^A, U^B, U^C, U^D を作成する.

$$\begin{aligned}
 & (ABCD A^* B^* C^* D^*)_{i_2(n+\hat{1}) i_3(n+\hat{1}) \bar{i}_2(n+\hat{1}) \bar{i}_3(n+\hat{1})} \\
 & \simeq \sum_{\delta}^{rD} U_{i_2(n+\hat{1}) i_2(n)\delta}^A \left(S_{\delta\delta}^A \right)^2 U_{\bar{i}_2(n+\hat{1}) \bar{i}_2(n)\delta}^{*A X \sigma} \quad (2.91)
 \end{aligned}$$

図 2.21: Γ のグラム行列から isometry を作る.

この時, isometry の内線は rD にオーバーサンプルする. この isometry のペアを A, B, C, D に演算する (図 2.22).

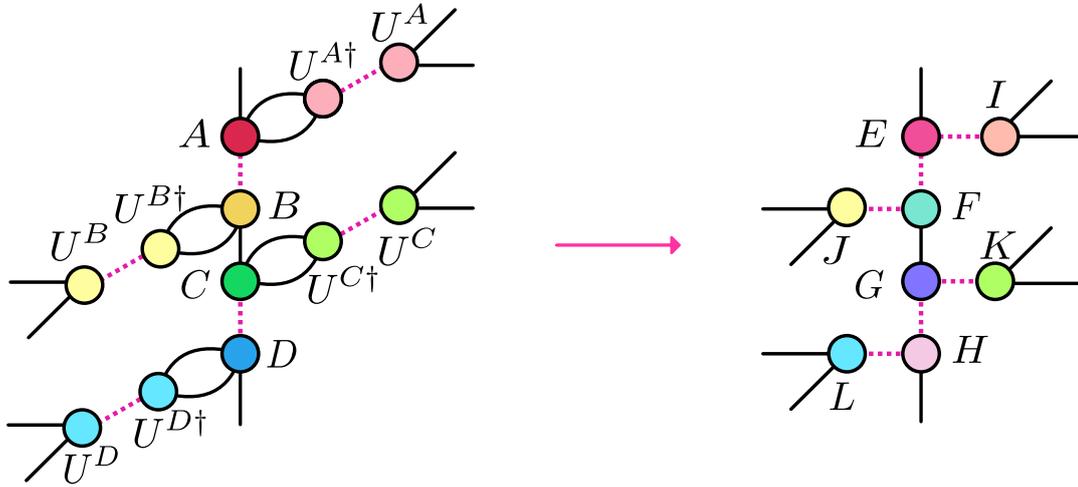


図 2.22: Triad 表現の作成.

この操作によって, Γ を 3 脚のテンソル E, F, G, H, I, J, K, L の積に分解できた. この分解は Γ の HOSVD になっているので, T の局所的な分解を行っていた TTRG よりも広い範囲の近似である. また, 内線のオーバーサンプルによっても, Triad 表現を作成する際の系統誤差を減らすことに成功している. このように作成した Triad 表現を用いて, Γ の SVD による isometry を求める (図 2.20(c)). 縮約ステップでは, 直接縮約を取るのではなく, RSVD の Range finder の方法を用いて

$$\left\| U^{(2)}U^{(3)}\Gamma U^{(2)\dagger}U^{(3)\dagger} - QQ^\dagger \left(U^{(2)}U^{(3)}\Gamma U^{(2)\dagger}U^{(3)\dagger} \right) \right\|_F \quad (2.92)$$

を最小にするような isometry $Q \in \mathbb{C}^{D \times D \times rD}$ を求める. この射影行列 QQ^\dagger を用いて Γ のランクを落としながら縮約を取ることで, 縮約のコストは $O(qr^3 D^{d+3})$ になる. ただし, q は QR 分解の反復回数である. Triad 表現を用いない通常の MDTRG は $O(qr^2 D^{2d+1})$ のコストとなる. また, メモリコストは $O(rD^{d+1})$ である.

MDTRG はオーバーサンプル, Γ の分解を行うことで自由エネルギーの系統誤差を HOTRG 同等にまで改善している (図 2.23, [21] より引用.). MDTRG は, Γ の RSVD を行い縮約のコストを抑えるという点で ATRG と共通しているが, MDTRG では内線をオーバーサンプリングしている点, ATRG では縮約ステップの前に RSVD を用いている点が異なる. ATRG 同様, RSVD の

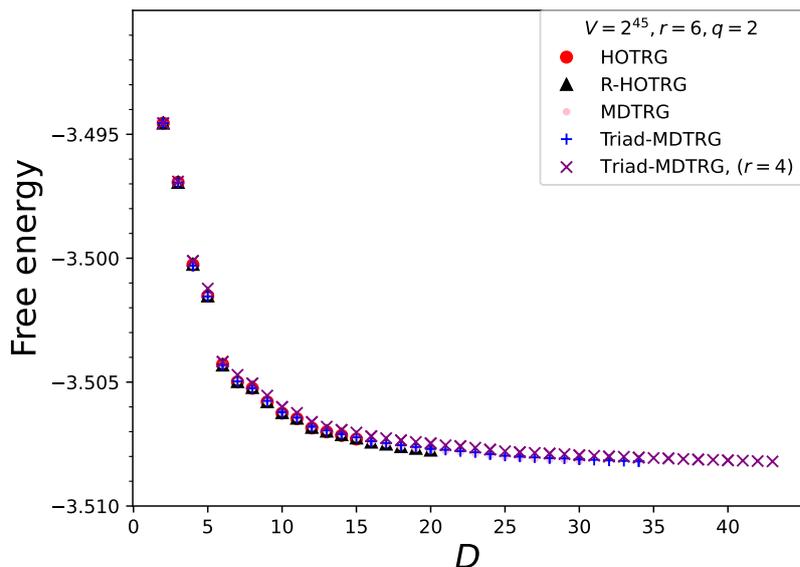


図 2.23: HOTRG, MDTRG 及び Triad-MDTRG の自由エネルギーの比較。図は [21] より引用。

パラメーター q を $O(D)$ に取ると、MDTRG の計算コストは $O(r^2 D^{2d+2})$ となることに注意する。同様に考えると、Triad-MDTRG は $O(r^3 D^{d+4})$ となる。このため、4次元では Triad 表現を用いた場合でも、 r の値によっては ATRG よりも高いコストが必要となることが予想される。

各手法の計算コストを表 2.1 にまとめる。

表 2.1: 各手法の計算コスト

Method	HOTRG	ATRG	MDTRG	Triad-MDTRG
計算コスト	$O(D^{4d-2})$	$O(D^{2d+1})$	$O(qr^2 D^{2d+1})$	$O(qr^3 D^{d+3})$
メモリコスト	$O(D^{2d})$	$O(D^{d+1})$	$O(rD^{d+1})$	$O(rD^{d+1})$

2.3.6 TRG の応用の現状と課題

HOTRG や ATRG は様々な高次元系に対しての応用例がある。例えば、HOTRG では 2,3次元でのスピン模型 [45–48] などへの応用が多数存在する。また、連続変数を持つボソンやゲージ系に対しては、変数の離散化や求積法による正則化、サンプリングなどを用いた計算を行う事で、符号問題を持つ系などにも拡張が可能である [49–56]。また、TRG アルゴリズムは、基本のテンソルを grassman 数を含む grassman テンソル [57] として扱う事でフェルミオン系に対しても計算コストを変えずに応用が可能であり、ゲージ場などを含めた応用例も多数存在する [28, 58–71]⁹。以上のように、TRG アルゴリズムは有限密度 QCD の第一原理計算に向け、様々なモデルへの適用やアルゴリズムが開発されており、4次元においても、ATRG を用いた様々な系への応用が行われている [15–20] が、連続ゲージ場などを含めた応用例は存在しない。QCD などの内部自由度の大きい 4次元系に対して TRG 法を応用するには、計算コストの削減や新たな並列アルゴリズムの開発が必要である。ATRG に変わる手法として、Triad-MDTRG などのアルゴリズムは存在するものの、4次元系に対しての応用例はなく、RSVD の反復パラメータやオーバーサンプリングパラメータを大きく取る場合には、計算コストが高くなる事が予想される。本研究の目的は、4次元

⁹フェルミオン系に関する詳しいレビューは [72] を参照。

系において一定の成功を収めている ATRG を基に, MDTRG の手法を用いて, 計算コストを削減すると同時に, 大規模な計算に向けた並列アルゴリズムの開発を行うことである.

第3章 本研究で使用する模型

3.1 臨界現象とスケーリング則

臨界現象は、二次相転移付近において物理量が特異な振る舞いを起こす現象である。このような現象は、例えば、流体の気液相転移や磁性体の相転移などで見られる。臨界現象の振る舞いは臨界指数とスケーリング則によって特徴付けられる。系の温度を T 、臨界温度を T_c 、磁場を h とする。臨界指数 $\alpha, \beta, \gamma, \delta, \nu, \eta$ は熱力学極限においてそれぞれ次のような物理量の振る舞いによって定義される。

臨界指数	スケーリング関係	条件
α	$c \sim T - T_c ^{-\alpha}$	$T - T_c \rightarrow 0^\pm, h = 0$
β	$m \sim (T_c - T)^\beta$	$T - T_c \rightarrow 0^-, h = 0$
γ	$\chi \sim T - T_c ^{-\gamma}$	$T - T_c \rightarrow 0^\pm, h = 0$
δ	$m \sim h ^{\frac{1}{\delta}}$	$T = T_c, h \rightarrow 0$
ν	$\xi \sim T - T_c ^{-\nu}$	$T - T_c \rightarrow 0^\pm, h = 0$
η	$G(r) \sim r^{-d+2-\eta}$	$T = T_c, h = 0$

表 3.1: 物理量とそのスケーリング関係

ただし、 $c, m, \chi, \xi, G(r)$ はそれぞれ比熱、磁化、磁化率、相関長、相関関数である。臨界指数は次のようなスケーリング則を満たすことが知られている。

$$\alpha + 2\beta + \gamma = 2, \quad (3.1)$$

$$\beta(1 + \delta) = 2 - \alpha, \quad (3.2)$$

$$\nu(2 - \eta) = \gamma, \quad (3.3)$$

$$\nu d = 2 - \alpha \quad (3.4)$$

このうち、式 (3.4) のような次元を含むようなスケーリング則を、ハイパースケーリングと呼ぶ。臨界指数は、ある次元 d_u より上では平均場のものに厳密に従う。このとき、ハイパースケーリング則は $d = d_u$ と代入することにより満たされる。このような d_u を上部臨界次元と呼ぶ。逆に、 d_u より低い次元では、熱揺らぎの効果が大きく、臨界指数は異なる値を取る。スケーリング則は繰り込み群の考え方をを用いて導出することができる。

3.2 Ising モデル

Ising モデルは、強磁性体の相転移を記述する最もシンプルな古典統計模型である。すべてのスピンは up か down の 2 つの状態を取り、最近接のスピン間の相互作用を記述する。 d 次元 Ising モデルのハミルトニアンは、格子点 n のスピン $\sigma_n = \pm 1$ と結合定数 J 、外部磁場 h を用いて、以下

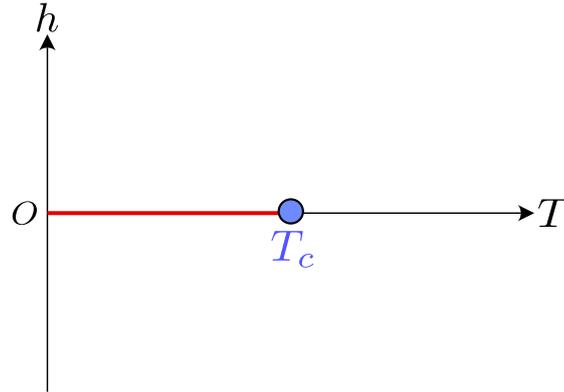


図 3.1: Ising モデルの相図. 赤い線は一次相転移線, 青丸は臨界点を示す.

のように記述される.

$$H = -J \sum_n \prod_{\mu}^d \sigma_n \sigma_{n+\hat{\mu}} - h \sum_n \sigma_n \quad (3.5)$$

このハミルトニアンはスピンのフリップに起因する Z_2 対称性を有している. Ising モデルの分配関数は, 逆温度 $\beta = 1/T$ (T は温度) を用いて,

$$Z = \sum_{\{\sigma\}} e^{-\beta H} \quad (3.6)$$

と書ける. 系のエネルギー E は温度 T によって定まり, 低温だとスピンの揃ったハミルトニアンの基底状態が支配的である. 逆に, 高温になるにつれ熱励起によってスピンはバラバラの値を取る. Ising モデルは, 最も基本的な相転移を起こすモデルである. 図 3.1 に典型的な Ising モデルの相図を示す. Ising モデルでは, 以下のように相が分類できる.

- ・ $h = 0, T < T_c$ では, 系は秩序相にあり, スピンは揃う傾向にある.
- ・ $h = 0, T > T_c$ では, 系は無秩序相にあり, スピンはランダムな配置を取る.
- ・ h を変化させた場合, $h = 0$ で磁化が不連続に変化する一次相転移が起こる.
- ・ 臨界点 $T = T_c$ では, 系は臨界現象を示す.

$d = 2, 3$ では, Ising モデルは臨界温度 T_c で二次相転移を起こす. すなわち, 磁化は連続的に変化し, 臨界点で 0 になる. 2次元では $h = 0$ の場合の厳密解が知られており, 臨界温度は $T_c = \frac{2}{\ln(1+\sqrt{2})}$ である [73]. 臨界指数は熱揺らぎの影響により平均場と異なる値を取る. Ising モデルの上部臨界次元は $d_u = 4$ であり, $d > 4$ では, 二次相転移を示し, 臨界指数も厳密に平均場の値を取る. 以下に各次元の Ising モデルの臨界指数を示す.

次元 d	α	β	γ	δ	ν	η
2^1	0	$\frac{1}{8}$	$\frac{7}{4}$	15	1	$\frac{1}{4}$
3^2	~ 0.110	~ 0.326	~ 1.237	~ 4.790	~ 0.630	~ 0.0363
4^3	0	$\frac{1}{2}$	1	3	0	$\frac{1}{2}$

表 3.2: Ising モデルの臨界指数

上部臨界次元 $d = 4$ では、厳密な結果 [75] によれば、このモデル二次相転移を示し、臨界指数は、平均場のものと一致する [75–77] が、以下のような対数補正が付く場合がある [78].

$$c(T) \sim |T - T_c|^{-\alpha} |\ln |T - T_c||^{\tilde{\alpha}} \quad (3.7)$$

$$m(T) \sim (T_c - T)^\beta |\ln (T_c - T)|^{\tilde{\beta}} \quad (3.8)$$

$$m(h) \sim |h|^{\frac{1}{\delta}} |\ln |h||^{\tilde{\delta}} \quad (3.9)$$

$$\chi(T) \sim |T - T_c|^{-\gamma} |\ln |T - T_c||^{\tilde{\gamma}} \quad (3.10)$$

$$\xi(T) \sim |T - T_c|^{-\nu} |\ln |T - T_c||^{\tilde{\nu}} \quad (3.11)$$

ϕ^4 理論ではこのような対数補正は厳密に正しいことが知られている [79–82]. しかしながら、 ϕ^4 理論の強結合極限である Ising モデルにおいては厳密な結果が得られていない.

数値計算では、モンテカルロ法による 4 次元 Ising モデルの計算は複数行われており、有限サイズスケリングによる対数補正の発散は見つかっていないものの、臨界指数は平均場のものと一致している [83–87].

一方、テンソル繰り込み群を用いた計算では、 $V = 1024^4$ の大体積において、4 次元イジングモデルは弱い一次相転移を示すという結果が得られている [13, 15]. これは厳密な結果と矛盾するものであり、さらに精度を上げた計算、および臨界指数等の計算が必要であると考えられる.

¹厳密な結果によるもの [7].

²共形ブーストラップ法によるもの [74].

³厳密な結果によるもの [75–77].

第4章 研究(アルゴリズム)

本章では、4次元の ATRG 法の高速度化手法をアルゴリズムの改善及び GPU 上での実装法を述べる。一部の内容は、[88]に基づいている。

4.1 Triad-ATRG

ATRG の計算コストは $O(D^{2d+1})$ であった。これは基本のテンソル式 (2.58-2.61) が D^{d+1} の大きさであったこと、およびボンド入れ替えにより最終の縮約ステップでの isometry の数が HOTRG に比べ半分になっていることに起因している。しかしながら現在世界で行われている計算の中で最大のボンド次元は D は $D = 55$ [18] であり、2次元などの場合の TRG アルゴリズムと比べると大幅に小さく、初期テンソルのサイズの観点でも、精度という観点でも十分ではない状況である。さらなる高次元への応用として、HOTRG の基本テンソルを、すべて足が3本のテンソルに分解し、RSVD などによる局所分解を繰り返しコストを $O(D^{d+3})$ まで削減したアルゴリズムである Triad TRG [43] がある。同手法は3次元での Ising 模型において、他手法と比べ最も早く自由エネルギーが収束するという結果が得られているが、局所分解による系統誤差の増大が4次元では大きくなってしまい、ATRG に比べ収束性が悪いという弱点がある。この弱点を克服したのが MDTRG [21] であり、Triad TRG で行った局所分解をすべてサブネットワーク Γ の分解として置き換え、内線を $D \rightarrow rD$ とオーバーサンプリングすることで、精度を HOTRG 同様に収束させることに成功している。同手法の計算コストは最終の縮約ステップで近似的な縮約である RSVD を利用しており、そのコストは、QR 分解の回数を q として $O(qr^3D^{d+3})$ である。同手法は高速であり精度も収束の範囲内で HOTRG と同等であるが、ボンド入れ替えを行わない HOTRG をベースとしたアルゴリズムであるため、ATRG と比べ最終の縮約ステップでの isometry の数が倍となっているために、縮約ステップで RSVD を用いなければ計算コストを $O(D^7)$ にまで削減する事ができない。このとき、ATRG のボンド入れ替えと同様に $q = O(D)$ と取った場合、4次元では r と D をどの程度大きく取れるかは不明瞭である。

そこで本研究では4次元の ATRG の手法において、Triad TRG および MDTRG の手法を取り入れた新たなアルゴリズムである、Triad-ATRG を提案する。その計算コストは $O(r^2D^7)$ であり、自由エネルギーや他物理量は数値的安定である。

本手法では、4次元の ATRG においてボンド入れ替えを行った後の基本テンソルである、

$$A_{i_1(n+\hat{1})i_2(n+\hat{1})\dots i_4(n+\hat{1})\alpha} = U_{i_1(n+\hat{1})i_2(n+\hat{1})i_3(n+\hat{1})i_4(n+\hat{1})\alpha} \quad (4.1)$$

$$X_{\alpha i_2(n)\dots i_4(n)\gamma} = \sum_q^{D^2} U_{\alpha q \gamma}^{\text{Bond}} Q_{i_2(n)i_3(n)i_4(n)q}^C \quad (4.2)$$

$$Y_{\beta j_2(n+\hat{1})\dots j_4(n+\hat{1})\gamma} = \sum_p^{D^2} V_{\beta p \gamma}^{*\text{Bond}} Q_{j_2(n+\hat{1})j_3(n+\hat{1})j_4(n+\hat{1})p}^B \quad (4.3)$$

$$D_{j_1(n)j_2(n)\dots j_4(n)\beta} = V_{j_1(n)j_2(n)j_3(n)j_4(n)\beta}^* \quad (4.4)$$

を Triad 表現に置き換えることからスタートする. 本研究での Triad 表現とは,

$$A_{i_1(n+\hat{1})i_2(n+\hat{1})\dots i_4(n+\hat{1})\alpha} = \sum_{k_A}^{rD} E_{i_1(n+\hat{1})k_A i_4(n+\hat{1})\alpha} I_{i_2(n+\hat{1})i_3(n+\hat{1})k_A} \quad (4.5)$$

のように 3 本足のテンソルと 4 本足のテンソルに近似することを言う. オリジナルの Triad TRG はすべてのテンソルを三つ足にしていたが, 4 次元では上記の形にしても計算コストが変わらない. すべてを 3 本足にするためには追加の近似が必要となり, 新たな系統誤差を生むため, 本研究ではこの形を採用する. また, どの方向の組を Triad にするかにも任意性があり, 上記では $\hat{2}, \hat{3}$ を Triad 方向に設定しているが, ほかの方向を選んでもよい. どの形が最適なジオメトリかは, モデルの対称性などによって大きく変わることが予想される. しかしながら本研究ではベンチマークとして等方的な 4 次元 Ising 模型を採用するため, このような形状の最適化は行わない.

MDTRG の手法にしたがい, Triad 表現を求める. ATRG 同様サブネットワーク Γ を考え, Γ に対する SVD として Triad 表現を求めることで, 局所分解による系統誤差を削減する. テンソルの分解は以下のように Reduced density matrix から求めた isometry を演算することにより実行される. $\Gamma = AX\sigma YD$ のネットワークは Canonical form になっているため, $AX\sigma, \sigma YD$ の Reduced density matrix を求めることで Γ 自体を最適化できることに注意して, A に対して,

$$(AX\sigma A^* X^* \sigma^*)_{i_2(n+\hat{1})i_3(n+\hat{1})\overline{i_2(n+\hat{1})}\overline{i_3(n+\hat{1})}} \simeq \sum_{k_A}^{rD} U_{i_2(n+\hat{1})i_3(n+\hat{1})k_A}^A \left(S_{k_A k_A}^A \right)^2 U_{\overline{i_2(n+\hat{1})}\overline{i_3(n+\hat{1})}k_A}^{*A} \quad (4.6)$$

のように isometry を求める. X, Y, D に対しても同様にこの操作を行う. この過程では, 特異値を $D^2 \rightarrow rD$ 個に近似している. 計算コストは直接 SVD する場合 $O(D^7)$ であり, RSVD を用いる場合は $O(qrD^6)$ である. そして, 新たな基本テンソルとして,

$$E_{i_1(n+\hat{1})k_A i_4(n+\hat{1})\alpha} = \sum_{i_2(n+\hat{1}), i_3(n+\hat{1})} A_{i_1(n+\hat{1})i_2(n+\hat{1})i_3(n+\hat{1})i_4(n+\hat{1})\alpha} U_{i_2(n+\hat{1})i_3(n+\hat{1})k_A}^{*A} \quad (4.7)$$

$$F_{\alpha k_X i_4(n)\gamma} = \sum_{i_2(n), i_3(n)} X_{\alpha i_2(n) i_3(n) i_4(n)\gamma} U_{i_2(n) i_3(n) k_X}^{*X} \sigma_{\gamma\gamma} \quad (4.8)$$

$$G_{\beta k_Y j_4(n+\hat{1})\gamma} = \sum_{j_2(n+\hat{1}), j_3(n+\hat{1})} Y_{\beta j_2(n+\hat{1}) j_3(n+\hat{1}) j_4(n+\hat{1})\gamma} U_{j_2(n+\hat{1}) j_3(n+\hat{1}) k_Y}^{*Y} \quad (4.9)$$

$$H_{j_1(n)k_D j_4(n)\beta} = \sum_{j_2(n), j_3(n)} D_{j_1(n) j_2(n) j_3(n) j_4(n)\beta} U_{j_2(n) j_3(n) k_D}^{*D} \quad (4.10)$$

$$I_{i_2(n+\hat{1})i_3(n+\hat{1})k_A} = U_{i_2(n+\hat{1})i_3(n+\hat{1})k_A}^A \quad (4.11)$$

$$J_{i_2(n)i_3(n)k_X} = U_{i_2(n)i_3(n)k_X}^X \quad (4.12)$$

$$K_{j_2(n+\hat{1})j_3(n+\hat{1})k_Y} = U_{j_2(n+\hat{1})j_3(n+\hat{1})k_Y}^Y \quad (4.13)$$

$$L_{j_2(n)j_3(n)k_D} = U_{j_2(n)j_3(n)k_D}^D \quad (4.14)$$

のように分解する (図 4.1). これは, Γ 自体を 4 回 SVD していることに相当する.¹

次に, Squeezer を導出する. この際, Γ は Triad 表現になっている為, もはや Canonical form ではない. 従って, E, F, G, H と G, H, K, L を独立に扱うことはできない. そこで, 次のように,

$$\begin{aligned} O_{\gamma\bar{\gamma}} &= \sum_{|\gamma, \bar{\gamma}} (EFIJ)(EFIJ)_{\gamma\bar{\gamma}}^\dagger \\ &= \sum_{|\gamma, \bar{\gamma}} \left\{ E_{i_1(n+\hat{1})k_A i_4(n+\hat{1})\alpha} E_{i_1(n+\hat{1})\bar{k}_A i_4(n+\hat{1})\bar{\alpha}}^* \left(I_{i_2(n+\hat{1})i_3(n+\hat{1})k_A} I_{i_2(n+\hat{1})i_3(n+\hat{1})\bar{k}_A}^* \right) \right\} \end{aligned} \quad (4.15)$$

¹正確には 4 つの方向について HOSVD による低ランク近似を行っている.

$$\times F_{\alpha k_X i_4(n)\gamma} F_{\bar{\alpha} \bar{k}_X i_4(n)\bar{\gamma}}^* \left(J_{i_2(n) i_3(n) k_X} J_{i_2(n) i_3(n) \bar{k}_X}^* \right) \quad (4.16)$$

のようにして $EFIJ$ のグラム行列 O を求める. 同様に $GHKL$ のグラム行列 P を求める. これらを Squeezer の導出に含めることにより, Γ 自体の SVD を行うことができる.

このようにして導出した O, N を用いて, $\hat{2}$ 方向の Squeezer を考える. 式 (2.91) に相当するグラム行列を,

$$\begin{aligned} & (EFIJNE^*F^*I^*J^*)_{i_2(n+\hat{1})i_2(n)\bar{i}_2(n+\hat{1})\bar{i}_2(n)} \\ &= \sum_{i_2(n+\hat{1}), i_2(n), \bar{i}_2(n+\hat{1}), \bar{i}_2(n)} \left(E_{i_1(n+\hat{1})k_A i_4(n+\hat{1})\alpha} E_{i_1(n+\hat{1})\bar{k}_A i_4(n+\hat{1})\bar{\alpha}}^* I_{i_2(n+\hat{1})i_3(n+\hat{1})k_A} I_{i_2(n+\hat{1})i_3(n+\hat{1})\bar{k}_A}^* \right) \\ & \quad \times \left(F_{\alpha k_X i_4(n)\gamma} P_{\gamma\bar{\gamma}} F_{\bar{\alpha} \bar{k}_X i_4(n)\bar{\gamma}}^* J_{i_2(n) i_3(n) k_X} J_{i_2(n) i_3(n) \bar{k}_X}^* \right) \end{aligned} \quad (4.17)$$

$$= \sum_{\delta}^{D^2} U_{i_2(n+\hat{1})i_2(n)\delta}^{EFIJ} \left(S_{\delta\delta}^{EFIJ} \right)^2 U_{\bar{i}_2(n+\hat{1})\bar{i}_2(n)\delta}^{*EFIJ} \quad (4.18)$$

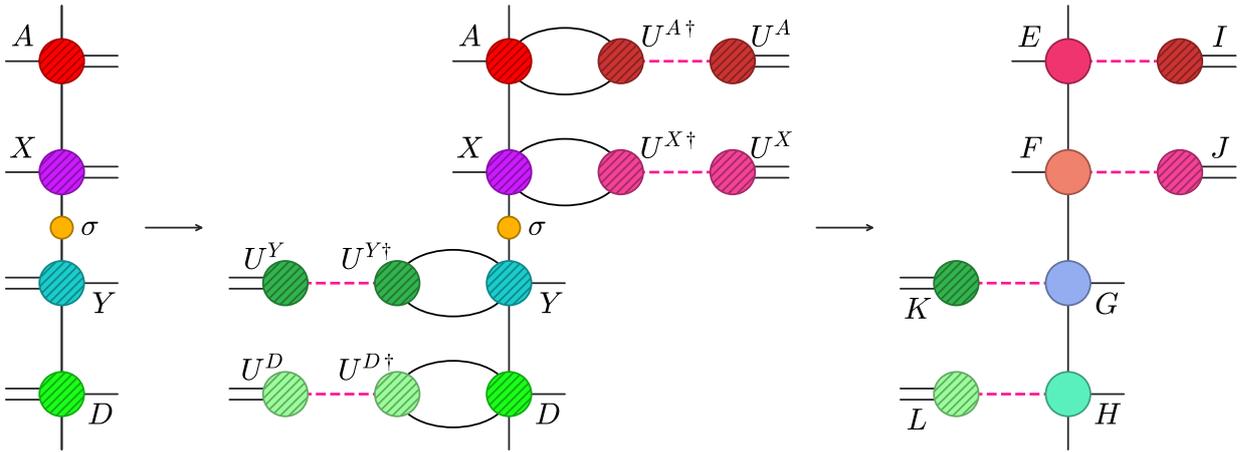


図 4.1: Triad 表現. isometry を演算することで足の本数を減らす. 点線の足の大きさは rD にオーバーサンプルされている.

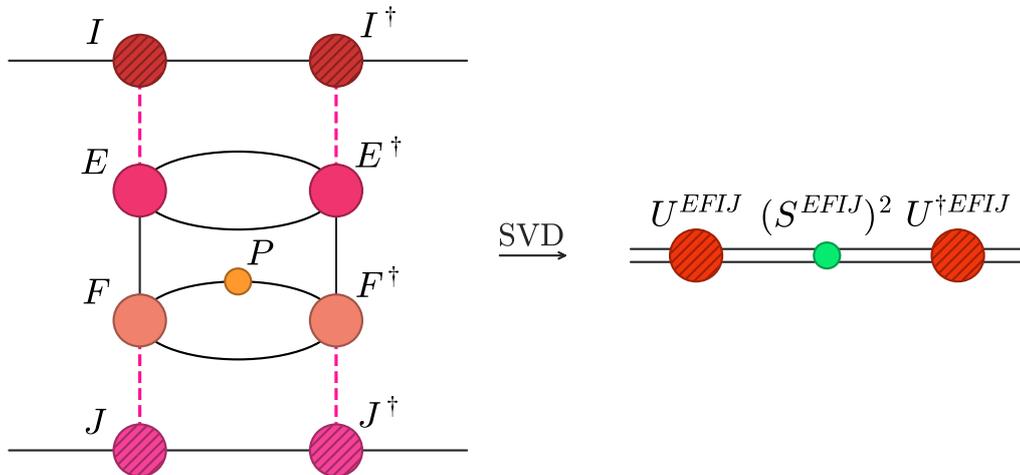


図 4.2: 式 (4.17) の Reduced density matrix の作り方. P を含めることで Γ 自体の分解になる.

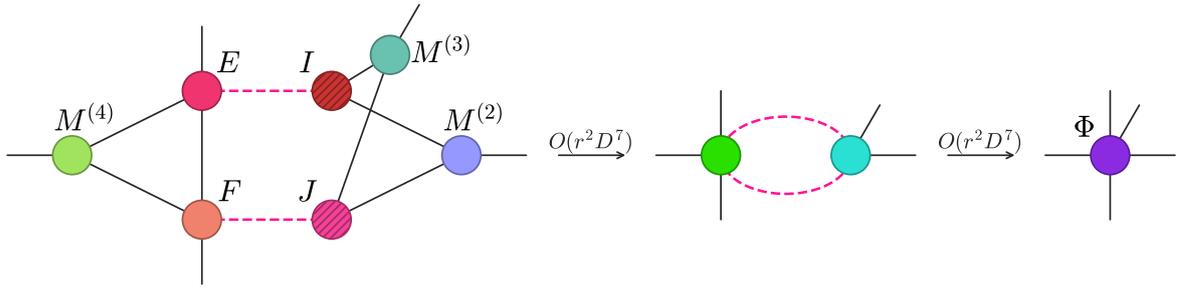


図 4.3: 式 (4.19,4.20) の縮約の様子.

のように求める。(図 4.2) $G H K L$ についても同様に求め、ATRG と同様の手法で、Squeezer である $M^{(2)}, N^{(2)}$ を求める。このコストは $O(\min(D^7, r^2 D^6))$ である。 $\hat{3}, \hat{4}$ 方向に関して、適切な縮約順で Squeezer を求める。式 (2.80, 2.81) に相当する縮約は、以下のように行う (図 4.3)。

$$\begin{aligned} \Phi_{i_1 k_2 k_3 k_4 \gamma} = & \sum_{/\gamma, \hat{1}, k} \left(E_{i_1(n+\hat{1})k_A i_4(n+\hat{1})\alpha} F_{\alpha k_X i_4(n)\gamma} M_{i_d(n+\hat{1})i_d(n)k_d}^{(4)} \right) \\ & \times \left(I_{i_2(n+\hat{1})i_3(n+\hat{1})k_A} J_{i_2(n)i_3(n)k_X} M_{i_2(n+\hat{1})i_2(n)k_2}^{(2)} M_{i_3(n+\hat{1})i_3(n)k_3}^{(3)} \right) \end{aligned} \quad (4.19)$$

$$\begin{aligned} \Psi_{j_1 k'_2 k'_3 k'_4 \gamma} = & \sum_{/\gamma, \hat{1}, k'} \left(G_{\beta k_Y j_4(n+\hat{1})\gamma} H_{j_1(n)k_D j_4(n)\beta} N_{j_4(n)j_4(n+\hat{1})k'_4}^{(4)} \right) \\ & \times \left(K_{j_2(n+\hat{1})j_3(n+\hat{1})k_Y} L_{j_2(n)j_3(n)k_D} N_{j_2(n)j_2(n+\hat{1})k'_2}^{(2)} N_{j_3(n)j_3(n+\hat{1})k'_3}^{(3)} \right) \end{aligned} \quad (4.20)$$

この縮約でかかる計算コストは、 $O(r^2 D^7)$ である。ここで Triad 表現では、 $\hat{2}, \hat{3}$ 方向と $\hat{1}$ 方向は独立に縮約をとることができるため、ATRG より計算コストが軽くなる。 $d = 4$ の Triad 表現の ATRG では、 Γ に対するとして、Triad 表現を作る isometry を求めるステップで $D^2 \rightarrow rD$ 個の特異値の切り捨てが 4 回、ボンド入れ替えで $D^3 \rightarrow D$ 個の特異値の切り捨てが 1 回ある。そして Squeezer ステップで $\Gamma\Gamma$ の $D^2 \rightarrow D$ 個の SVD が 3 回行われる。合計の計算コストは $O(r^2 D^7)$ であり、メモリコストは $O(D^5)$ である。以上の計算コストを表 4.1 に、Triad-ATRG の一連の操作を表したものを図 4.4 にまとめる。

表 4.1: 各手法の計算コスト

Step	ATRG	Triad ATRG
ボンド入れ替え	$O(qrD^6)$	$O(qrD^6)$
Triad 表現の構成	None	$O(D^7)$
Squeezer の導出	$O(D^7)$	$O(\min(D^7, r^2 D^6))$
最終の縮約	$O(D^9)$	$O(r^2 D^7)$

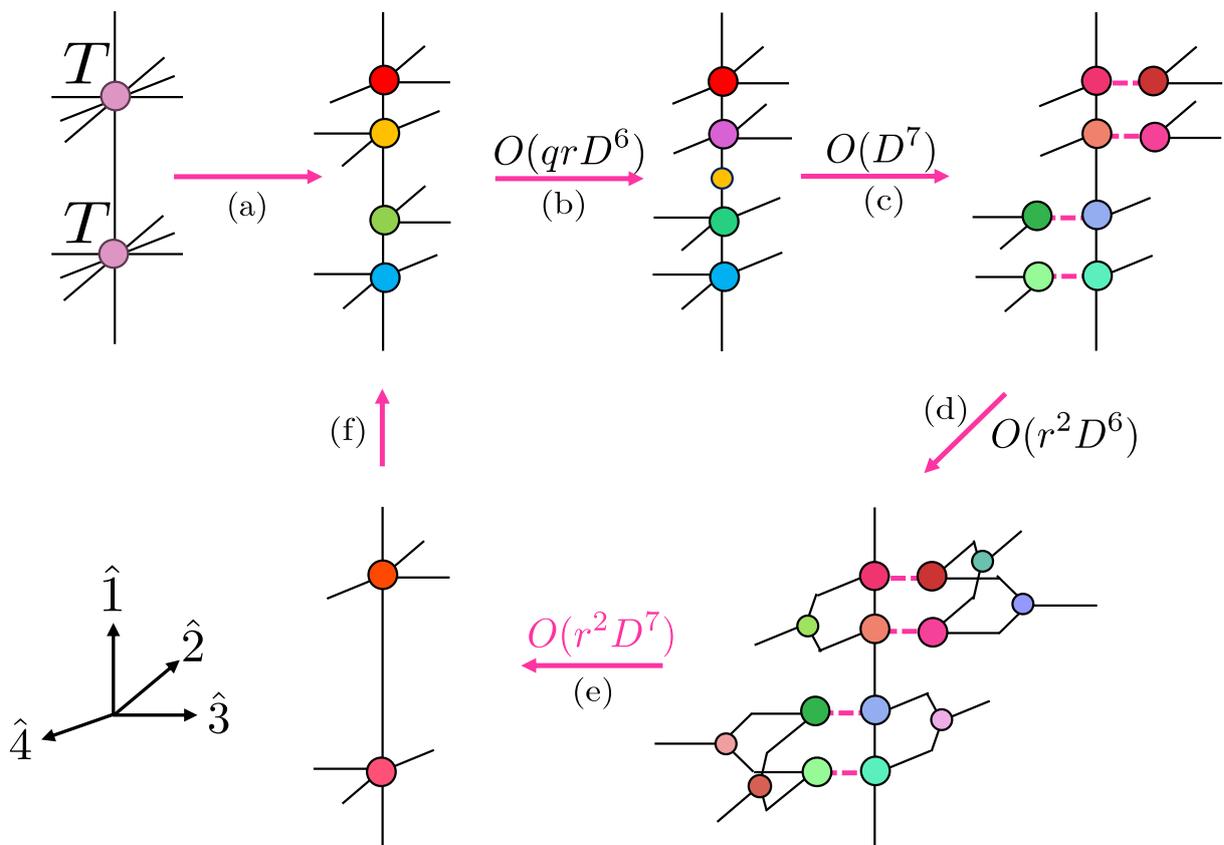


図 4.4: Triad-ATRG の一連の操作を表した図. (a) 基本テンソルの分解. (b) ボンド入れ替え. (c) Triad 表現の構成. (d) squizzer の導出. (e) 縮約. (f) 次のステップの基本テンソルを作成.

4.2 GPU による実装

テンソル繰りこみ群の計算は、テンソルの分解及び縮約（行列積）からなるアルゴリズムであり、並列化の恩恵を受けやすい。従って大規模な MPI 並列計算が行われることが多く、並列用のアルゴリズムも存在する [89]。例えば、式 (2.80) の縮約は、 A の i_1 のインデックスを D 個に分割し、 D 個のノードに縮約処理を分配することで、計算コストを $O(D^9)$ から $O(D^8)$ に削減することができる [16]。このような CPU を用いた並列計算は、メモリコストを分散しながら、計算資源を増やすことで大規模な系への応用も可能である。しかしながら現実には計算資源と性能の制限があるため、4次元においてさらに大きなボンド次元を達成することは難しい。このような事情により、近年注目されているのが GPU (画像処理装置) を用いたテンソル演算である。GPU による科学技術計算は、機械学習や AI などの開発において広く用いられている。GPU は CPU に比べ多数のコアを持ち、単純な処理を並列して行うことに適している。特に、行列積は並列化が容易であり、CPU よりも圧倒的な計算速度を有している。したがって、GPU はテンソル繰りこみ群において有用なハードウェアであると考えられる。実際に 2次元において、 $O(D^2)$ ほどの削減が達成がされている [90]。しかしながら、GPU が持つビデオメモリは CPU のものと比べて少ないため、大規模なデータを一度に保有することは難しい。したがって基本テンソルのサイズが大きい高次元では、メモリの不足により十分な高速化がされず、実用には至っていない。しかしながら近年では大きなメモリを有する科学計算用の GPU も存在し、それらを用いれば 4次元でも十分に実用化することが可能であると考えられる。以下に ATRG と Triad-ATRG の GPU 上での実装の簡潔なアルゴリズムを示す。

通常 GPU に搭載されるメモリは 10GB から 80GB であり、本研究で用いた京都大学のスーパーコンピュータ yukawa21 には、16GB のメモリを有する Tesla V 100 が 2 台搭載されている。また、大阪大学のスーパーコンピュータ SQUID には 40GB のメモリを有する NVIDIA A100 が 1 ノードにつき 8 台搭載されている。これに対し、ATRG で基本となるテンソル A, B, C, D のサイズは D^5 であり、倍精度では 1 つのテンソルにつき $\frac{8D^5}{1024^3}$ GB のメモリを必要とする。表 (4.2) より、例えば、 $D = 50$ では 1 つにつき 2.33GB 必要なので、4 つのテンソルを取めるには 9.31GB のメモリを必要とする。 $D = 70$ では 50GB のメモリが必要となる。したがって、単体の GPU では大きなボンド次元を扱うのは難しい。しかしながら、ATRG はその特性上、常に隣り合った 2 つのテンソル同士で分解、縮約を行うため、実際には同時に 4 つの基本テンソルを GPU 上に保持する必要はない。さらに、Triad 表現を作るステップや isometry を求めるステップでは 2 台の GPU にそれぞれ 2 つずつのテンソルを分配し、縮約及び SVD を行えば、メモリコストを削減しつつ半分の計算時間に抑えることが可能である。また、Triad-ATRG ではサブネットワーク Γ が Canonical form ではないため、 E, F, I, J 及び G, H, K, L 単体では計算が行えないが、各 GPU 上で先に式 (4.15) を計算し、プロセス通信を行うことで実装が可能である。最終の縮約ステップでは、 $E, F, I, J, \{M\}$ と $G, H, K, L, \{N\}$ を n 台の GPU に分配する。この際、縮約に関わらない E

表 4.2: 各ボンド次元での基本テンソルのメモリコスト

D	サイズ (GB)
45	1.37
50	2.33
55	3.75
60	5.79
65	8.64
70	12.5

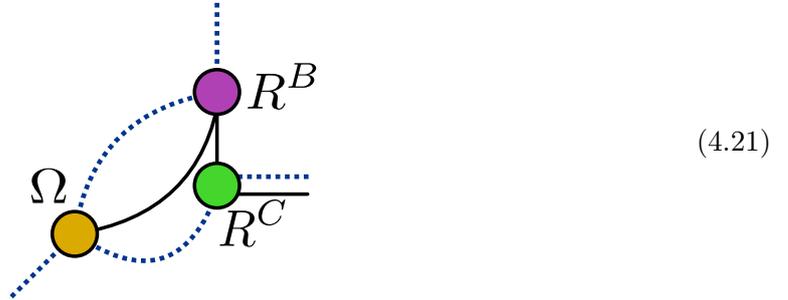
と H の 1 つ目の添え字は各 GPU に均等に分配することで, 1 台での計算よりおよそ n 倍の効率化を図ることができる. 表 4.3 に各ステップでのテンソルの分配の様子を示した.

表 4.3: n 台の GPU での各ステップでのテンソルの分配の様子. $E(1 : D/n, :, :, :)$ は, E の最初の添字を 1 から D/n だけを用いることを意味する.

GPU	ボンド入れ替え	Triad 表現の構成	Squeezer の導出	Φ の縮約	Ψ の縮約
GPU 1	R^B, R^C	A, X, σ	E, F, I, J, P	$E(1 : D/n, :, :, :), F, I, J, \{M\}$	$H(1 : D/n, :, :, :), G, K, L, \{N\}$
GPU 2	None	σ, Y, D	G, H, K, L, O	$E(D/n : 2D/n, :, :, :), F, I, J, \{M\}$	$H(D/n : 2D/n, :, :, :), G, K, L, \{N\}$
GPU 3	None	None	None	$E(2D/n : 3D/n, :, :, :), F, I, J, \{M\}$	$H(2D/n : 3D/n, :, :, :), G, K, L, \{N\}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
GPU n	None	None	None	$E((n-1)D/n : D, :, :, :), F, I, J, \{M\}$	$H((n-1)D/n : D, :, :, :), G, K, L, \{N\}$

4.3 ボンド入れ替えのオーバーサンプリング

MDTRG [21, 44] では, Triad 表現の内線に加えて, 基本テンソル A と B , C と D を繋ぐ内線もオーバーサンプルすることで, 系統誤差を削減していた. このオーバーサンプルは, 縮約ステップ (図 2.20(e)) において, RSVD の第一段階として演算する QQ^\dagger のランクを rD にすることで実現されていた. ATRG においてもボンド入れ替えのオーバーサンプルを考えることが可能である. ATRG の場合, ボンド入れ替えでオーバーサンプリングした内線は次のステップでの基本テンソル A と B , C と D を繋ぐ内線と同じ大きさになる. このオーバーサンプリングパラメーターを r_{bond} とおくと, ボンド入れ替えでは式 (2.66) において, β, α, p, q がそれぞれ $r_{\text{bond}}D, r_{\text{bond}}D, r_{\text{bond}}D^2, r_{\text{bond}}D^2$ の大きさを持つようになる. したがってボンド入れ替えでは,



のような縮約を行う. ここで青色の点線は $r_{\text{bond}}D$ の大きさを持つ. したがって, ボンド入れ替えのコストは, $O(qr_{\text{bond}}^4 D^6)$ となる. $q = O(D)$ と取った場合, 計算コストは $O(r_{\text{bond}}^4 D^7)$ となる. ボンド入れ替え後のサブネットワークは, 図 4.5 のようになる.

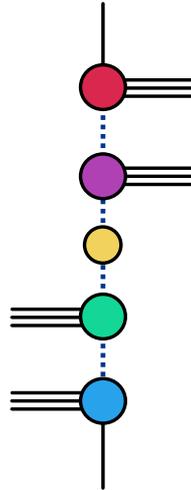
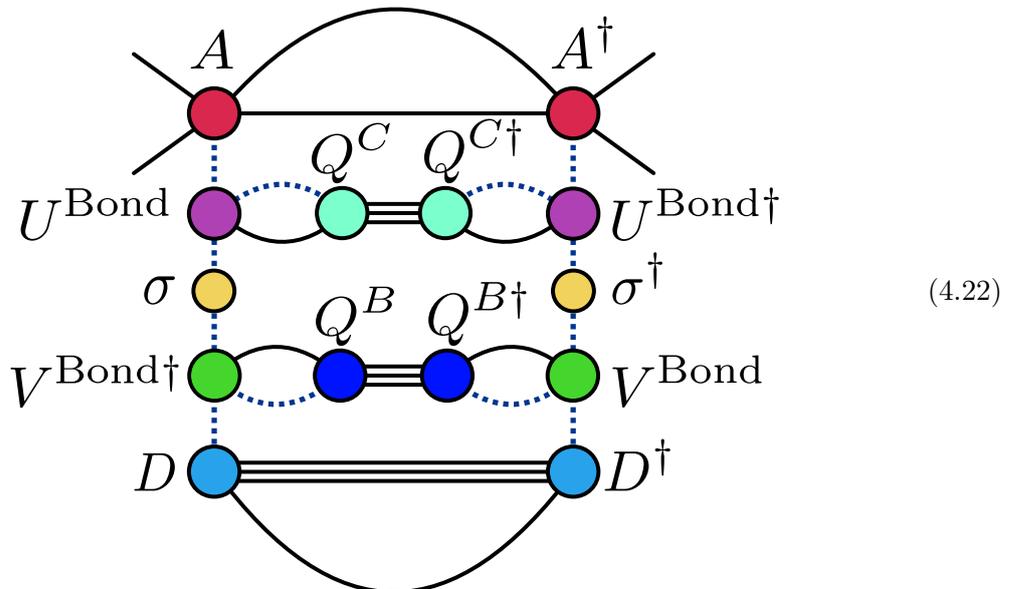


図 4.5: ボンド入れ替え後のサブネットワーク. 青色の点線は $r_{\text{bond}}D$ の大きさを持つ.

この時, $O(r_{\text{bond}}^2 D^5)$ のメモリコストを持つ基本テンソルが 2 つ生まれてしまう. 例えば $D = 40$, $r_{\text{bond}} = 5$ と取った場合, 20GB 近くのメモリが必要になる². 従ってボンド次元を大きくすることが困難になる. また, ボンド入れ替えをオーバーサンプリングした場合, 最終の縮約のコストは $O(r_{\text{bond}}^2 D^9)$ となり, 元の ATRG の数十倍の計算コストになってしまい困難である.

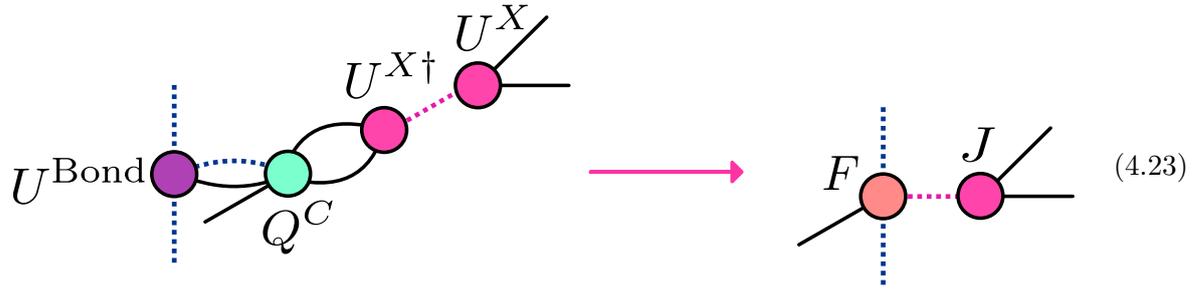
上記の問題は, Triad-ATRG の内線オーバーサンプリングを考えることで解決できる. このアルゴリズムを以下, ILO Triad-ATRG (Internal Line Oversampling Triad-ATRG) と呼ぶ. Triad-ATRG の場合, ボンド入れ替えの後, 式 (2.71, 2.72) の縮約を行わず, $Q^B, Q^C, U^{\text{bond}}, \sigma, V^{\text{bond}\dagger}$ を用いて Γ のグラム行列を作り Triad 表現に移行することが可能である. 例えば, 次のようなグラム行列を縮約し, U^A を求めることができる.



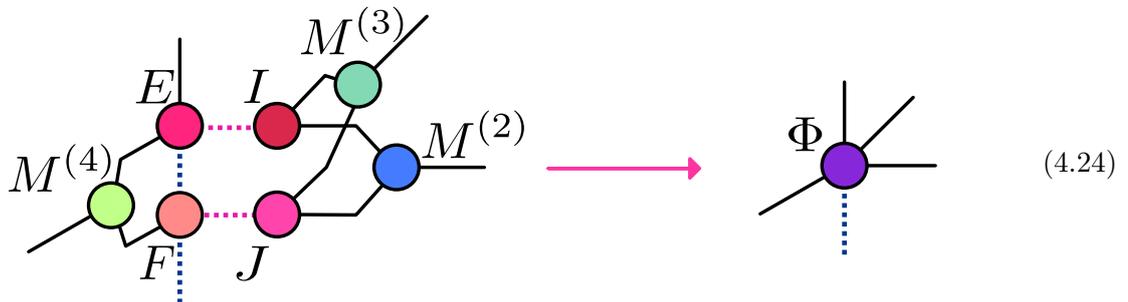
このようにして U^A, U^X, U^Y, U^D , を求めた後, 通常の Triad-ATRG 同様に Triad 表現に移行で

²原理的には X, Y を作らずに, squeezer の導出と縮約ステップで Q^B, Q^C の添字を分割すればメモリコストは増えない. ただしこの場合でも ATRG の r_{cond}^2 倍の特異値を使用しているコストがかかる問題は変わらない.

きる．例えば， F, J を求める際は以下のように縮約する．



このようにして得た Triad 表現に対し，squeezer を求め，縮約を行う．縮約ステップでは以下のようなネットワークを縮約する．



この計算コストは， $O(r^2 r_{\text{bond}}^2 D^7)$ である．従って oversampled Triad-ATRG の計算コストのボトルネックとなるのは，ボンド入れ替えの $O(r_{\text{bond}}^4 D^7)$ と縮約ステップの $O(r^2 r_{\text{bond}}^2 D^7)$ となる．このコストは Triad-ATRG よりも大きいですが，ATRG の $O(D^9)$ より小さいため，実現可能であると考えられる．

第5章 数値計算結果

本章では, Triad-ATRG を用いた具体的な数値計算結果を 4 次元の Ising モデルを用いて示す. なお, 本研究では CPU でのスケーリング計測以外の計算はすべて yukawa21 及び SQUID の GPU ノードにて行っている.

5.1 CPU での計算コストのスケーリング

ATRG の計算コストが $O(D^9)$ であるのに対し, Triad-ATRG の計算コストは $O(D^7)$ となる事が前章で示された. この計算コストが実際に成り立っていることを確かめるために, 1 コアの CPU を用いて計算時間を計測したものが以下である (図 5.1). 計算条件は $q = 20$, $N = 1024^4$ とおいた. また, Triad-ATRG では $r = 7$ としている. 実線がそれぞれ理論予想のスケーリングを表す. 図 5.1 より確かに ATRG と Triad ATRG の計算コストは D が大きい領域で理論通りにスケールしていることがわかる. また $D = 31$ において Triad-ATRG は ATRG に対して 10 倍ほどの速度向上が見られているのは特筆すべき点であり, 実際の計算で用いる $D = 50$ などではおよそ 20 倍程度の速度向上が見込まれることになる. これらの結果より, Triad-ATRG は ATRG よりも効率的に計算が行えることが確認された.

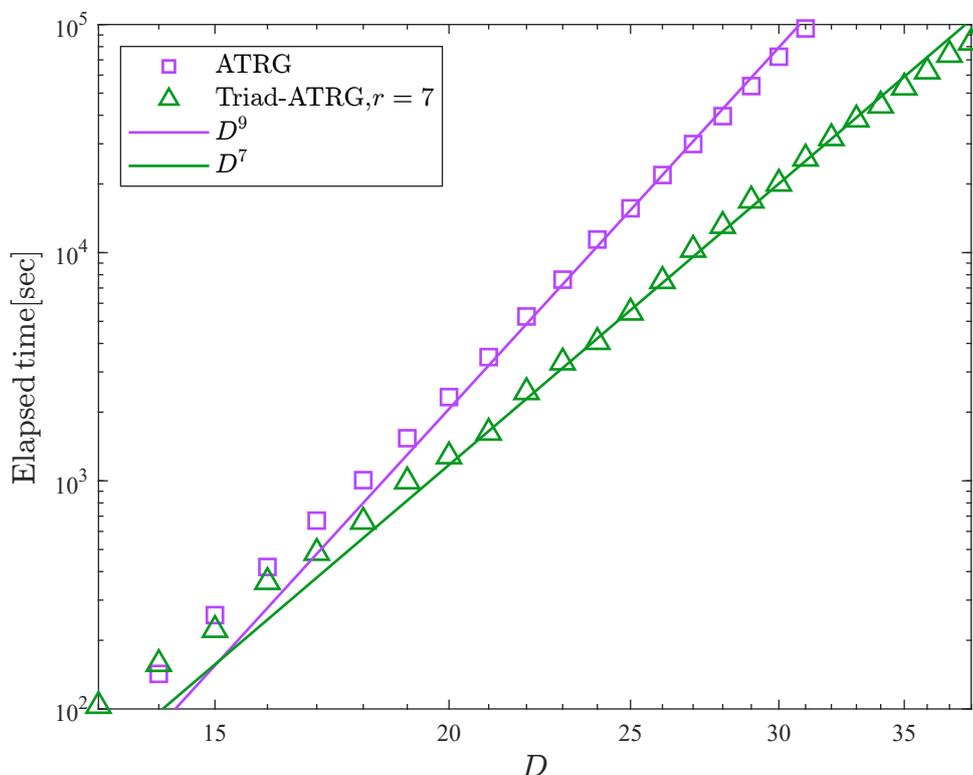


図 5.1: 実行時間の計測. D が大きい領域で ATRG は D^9 , Triad ATRG は D^7 のスケーリングに従っていることが確認できる.

5.2 GPU での計算時間のスケールリング

次に, yukawa21 において 2 台の GPU を用いて自由エネルギーを計算した計算時間のスケールリングを図 5.2 に示す. ATRG では理論値の $O(D^9)$ のスケールリングに比べ, $O(D^8)$ のスケールリングを示している. この結果は, [90] において示されていた計算時間のスケールリングの向上を, 高次元においても達成したことを意味する. これは [15] において $2D$ 個の CPU プロセスで計算されたスケールリングと同じであり, 2 台の GPU での計算のみでスケールリング向上が達成されているのは特筆すべき点である.

次に Triad-ATRG での結果に関して述べる. ATRG が $O(D)$ のスケールリング改善であったのに対し, Triad-ATRG では理論値の $O(D^7)$ に比べ $O(D^6)$ よりも低いスケールリングに従っている. これは, ボトルネックとなる縮約計算時のメモリ空き容量が関係していると考えられる. なぜなら Triad-ATRG においては縮約時に D^5 の基本テンソルを有する必要がないため, 大きなブロックサイズをとることが可能であり, GPU 上でのインデックスの入れ替えやアロケーションにかかる一定のコストを減らすことができるためである. 以上の結果より, 大きなボンド次元計算を GPU クラスタ上で実行する際には, より低いスケールリングを示す Triad-ATRG が有利であると考えられる.

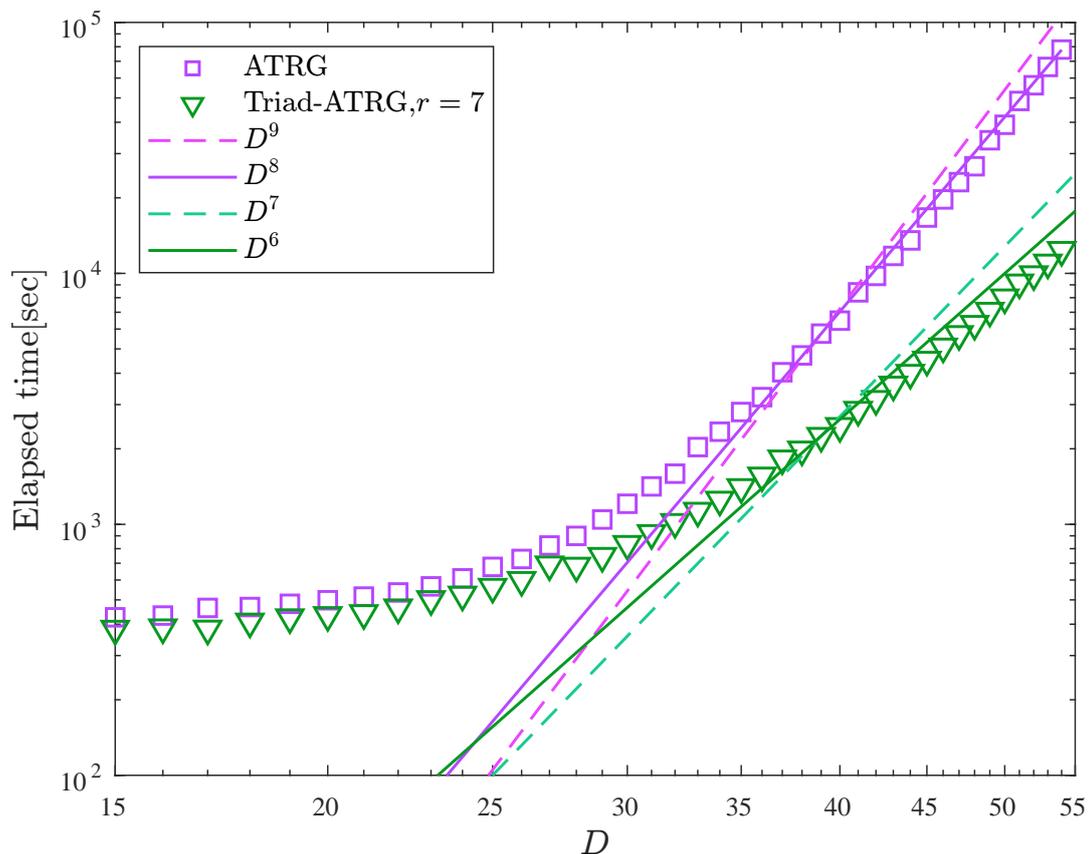


図 5.2: 2 台の GPU を用いた実行時間の計測. D が大きい領域で ATRG は D^8 , Triad-ATRG は D^6 以下のスケールリングに従っていることが確認できる.

5.3 自由エネルギーの精度

次に Triad-ATRG の精度について考察する．ATRG で最も特異値を切り捨てるのはbond入れ替えのステップであり，その切り捨て誤差よりも Triad 表現の切り捨て誤差が小さければ，定性的には ATRG と同じ精度を与えるであろうことが予想できる．実際に自由エネルギーの収束を比較したものが図 5.3 である．Triad-ATRG の結果は $r = 7$, $r = 20$ ともに ATRG での結果に収

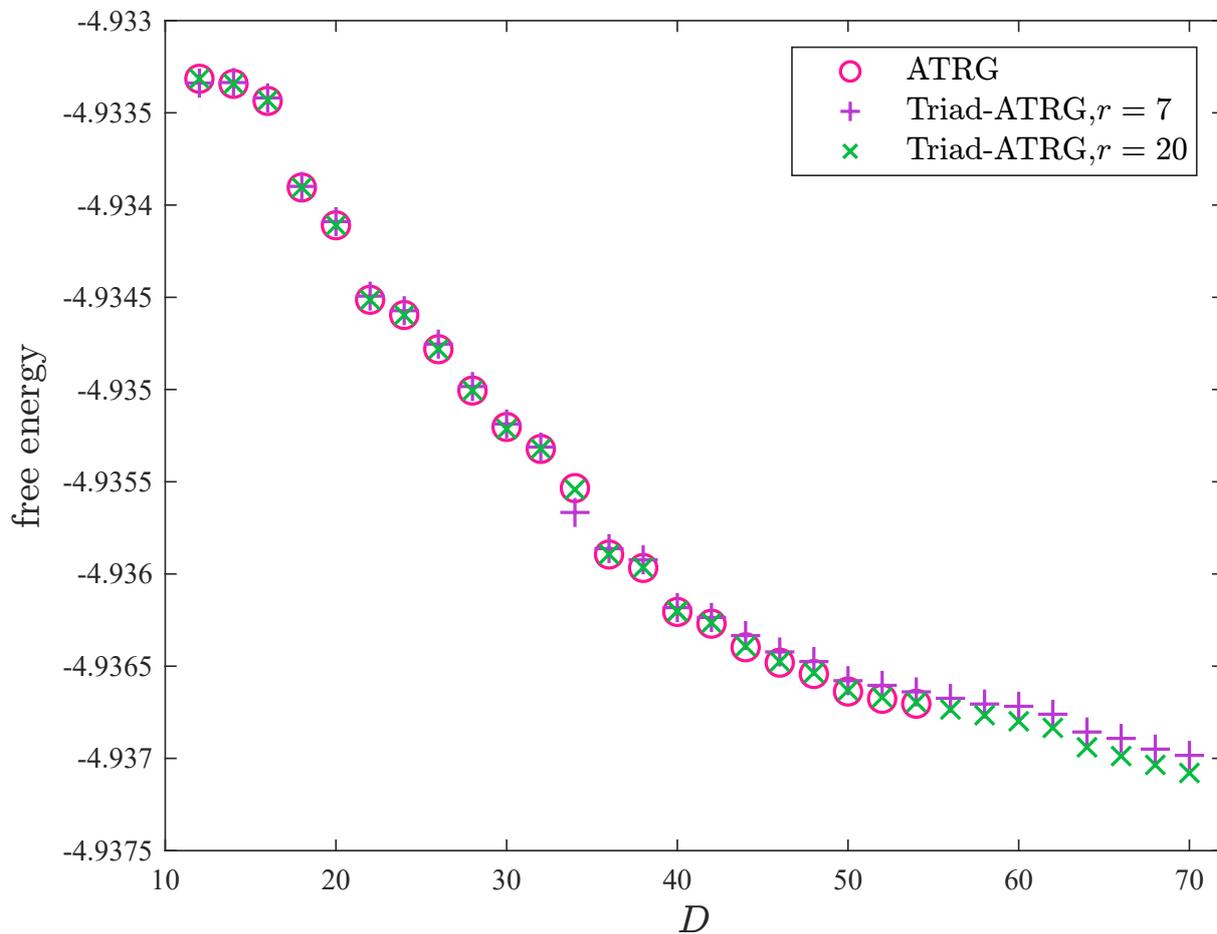


図 5.3: 自由エネルギーの計算．両手法は高い精度で一致している

D	ATRG	$r = 7$	$r = 20$
38	-4.9359676	-4.9359235	-4.9359646
40	-4.9362060	-4.9361825	-4.9362026
42	-4.9362695	-4.9362360	-4.9362648
44	-4.9363974	-4.9363340	-4.9363918
46	-4.9364809	-4.9364227	-4.9364746
48	-4.9365426	-4.9364745	-4.9365357
50	-4.9366373	-4.9365787	-4.9366312
52	-4.9366769	-4.9366039	-4.9366695
54	-4.9367035	-4.9366392	-4.9366959

表 5.1: $r = 7, 20$ での Triad-ATRG と ATRG の自由エネルギーの数値．

束条件精度の範囲内で一致していることがわかる。具体的な数値を表 5.1 に示した。 $D = 54$ では ATRG との差異は 0.0013% ($r = 7$), 0.00015% ($r = 20$) となっている。これに対し、ATRG の結果はボンド次元に対し収束しておらず、例えば ATRG の $D = 48$ と $D = 54$ の差異 0.0032% 程度であり、これは $D = 54$ での Triad-ATRG との差異よりも大きい。したがって、Triad-ATRG は ATRG と束条件精度内で同等な結果を与えていると解釈できる。従って [21] において指摘されていた、 Γ 自体の分解が大きな系統誤差を生まないという結果は ATRG でも有効であることがわかる。以上から、自由エネルギーの計算では Triad-ATRG は ATRG と同等の精度を持つことが確認された。

5.4 相転移点の精度

次に、4 次元 Ising モデルの相転移点の精度の検証を行う。相転移点は、 $D = 13$ の HOTRG [15] において $T_c = 6.650365(5)$ と求められており、ATRG においては $D = 38$ において、HOTRG から 0.12% の差異があった。また、近年のモンテカルロ法による計算 [87] では、 $T_c = 6.6803040(5)$ と求められており、テンソル繰り込み群によるものと大きな差異がある。

本研究では先行研究と同様の方法で、ATRG, $r = 7, 10, 15, 20$ の Triad-ATRG でその束条件精度を調査する。テンソル繰り込み群において Ising モデルなどの Z_n 対称性を持つ理論の相転移を検出するうえで有用な量が [91] で導入された

$$X^{(m)} = \frac{(\text{Tr} A^{(m)})^2}{\text{Tr}(A^{(m)})^2}, \text{ with } A_{kl}^{(m)} = \sum T_{i_1 i_2 i_3 k i_1 i_2 i_3 l}^{(m)} \quad (5.1)$$

である。 $T^{(m)}$ は m 回目の粗視化を行った基本テンソルである。この量は $A^{(m)}$ の一番大きな固有値の縮退度を数えており、基底状態の縮退度に相当する。 Z_2 対称性が破れた秩序相では $X = 2$, 非秩序相では $X = 1$ に収束する。 $X^{(m)}$ の ATRG と Triad-ATRG での相転移点付近での典型的なふるまいが図 5.4a, 5.4b である。図より、どちらの手法もある温度を境に $X^{(m)}$ の値が 2 と 1 に収束することがわかる。

以上の背景をもとに、複数の D, r での相転移点を比較したものを図 5.5 に示す。誤差棒は本研究の数値計算で用いた温度の解像度を表す。この結果より $D = 54$ において、ATRG と Triad-ATRG の差異は、 $r = 7$ で約 0.1% , $r = 10$ で約 0.05% であることがわかる。また、 $r = 20$ と取ると ATRG との差異は 0.01% 程度に収まる。以上より、Triad-ATRG は十分な精度で相転移点が求められていることがわかる。さらに、Triad-ATRG の計算では $D = 70$ までの計算が可能になり、 $D > 54$ での相転移点の D に対する束条件性は、 $D < 54$ までの束条件性に比べて向上していることが図よりわかる。ただし、Triad-ATRG の結果もボンド次元 D の大きさに対して束条件していないため、さらなる大きな D での比較をしなければ束条件値の精度については不明であることに注意が要する。特筆すべき点として、相転移点はどの D においてもモンテカルロ法の $T_c = 6.6803040(5)$ よりも低い値を示す傾向にある点が挙げられる。

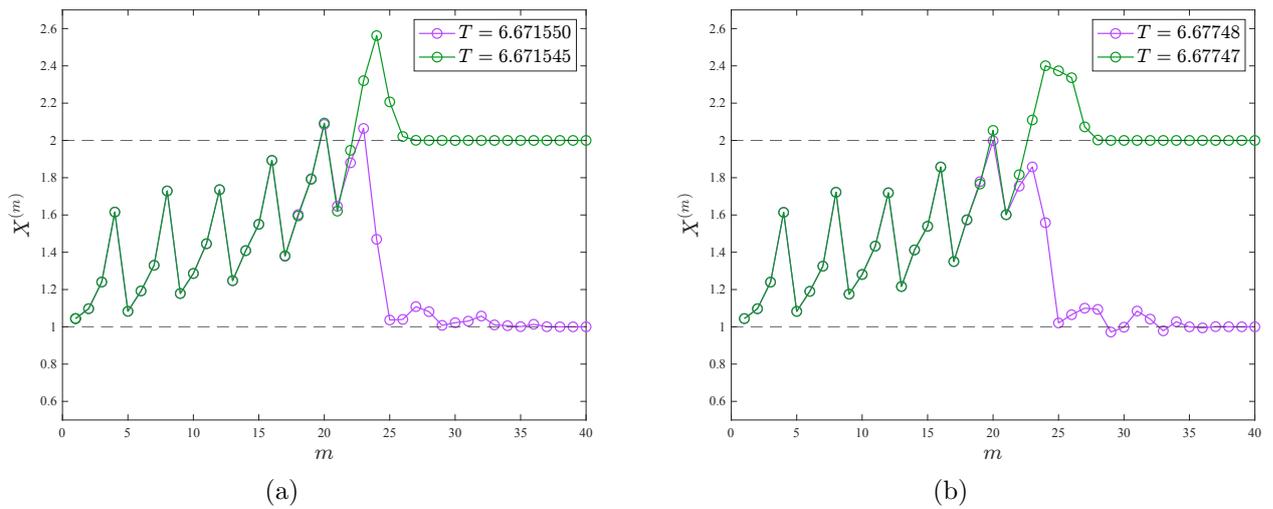


図 5.4: (a) ATRG での $X^{(m)}$ の振る舞いと (b) Triad-ATRG での $X^{(m)}$ の振る舞い

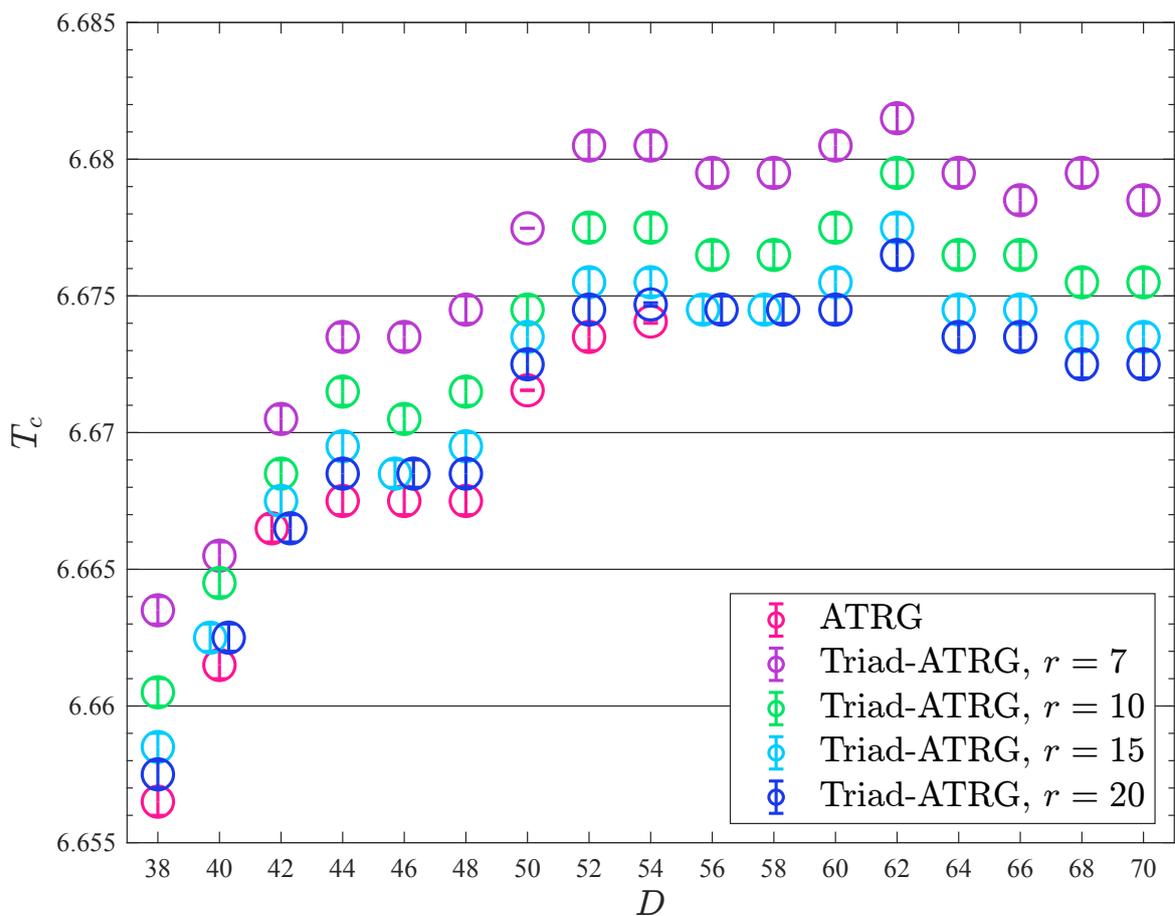


図 5.5: ATRG, Triad-ATRG での相転移点の比較. 誤差はそれぞれの計算での温度の解像度である.

5.5 内部エネルギーの計算

相転移点や自由エネルギーは、impurity テンソルを含まないネットワークのみで計算可能な物理量であった。本節では、impurity テンソルを含むような内部エネルギーの精度が ATRG と Triad-ATRG においてどのように変わるのかを調べる。内部エネルギーは、式 (2.37) で定義されたネットワークを縮約することによって計算される [13]。図 5.6 に $D = 54$ における ATRG と Triad-ATRG の結果を示す。ただし $r = 20$ と取った。両者の結果は、相転移点の立ち上がり場所が異なるため、低温側と高温側で振る舞いが異なるが、高温側 ($T = 6.67475$) では、0.002% の差異、低温側 ($T = 6.674$) では 0.2% でありどちらも十分小さい差異であると言える。以上から、impurity テンソルが含まれたネットワークでも、Triad-ATRG は ATRG と同等の精度を持つと結論付けられる。

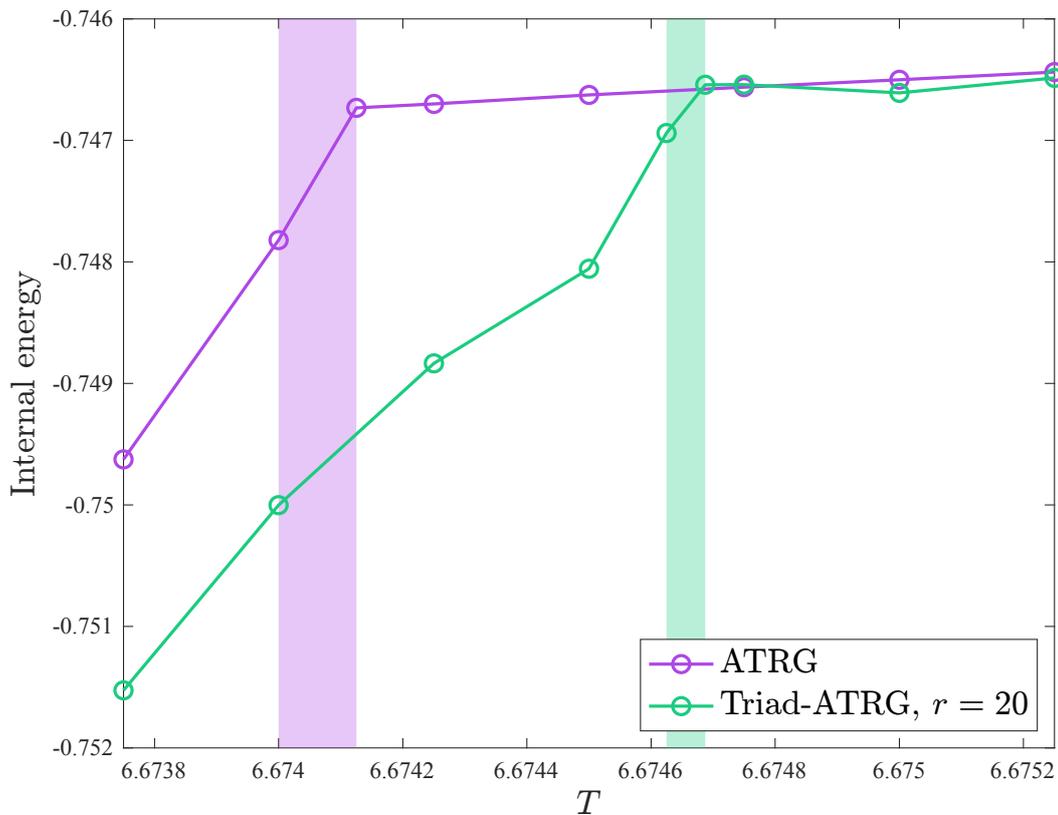


図 5.6: $D = 54$ における、ATRG と Triad-ATRG($r = 20$) の内部エネルギーの比較。横軸は温度である。紫のバンド、緑のバンドはそれぞれ ATRG と Triad-ATRG での相転移点を示している。

5.6 HOOI による Triad 表現

Triad-ATRG は、オリジナルの ATRG のサブネットワーク Γ に HOSVD [27] を用いた近似を行うことで、低ランクのテンソルに近似するアルゴリズムであった。本節では、HOSVD で求めた Triad 表現に対し、HOOI [26] を用いた近似を行い、その精度を調べる。初期の isometry である U^A, U^X, U^Y, U^D を HOSVD によって求めたものとし、アルゴリズム 1 の手順を行った。ただし反復回数は 200 回とした。図 5.7 に、 $D = 40$ における、各ステップ m における HOSVD による Triad 表現による Γ のノルムと、そのネットワークに HOOI を用いた Triad 表現のノルムの比

$$\frac{\|\Gamma_{\text{HOOI}}\|_F}{\|\Gamma_{\text{HOSVD}}\|_F} \quad (5.2)$$

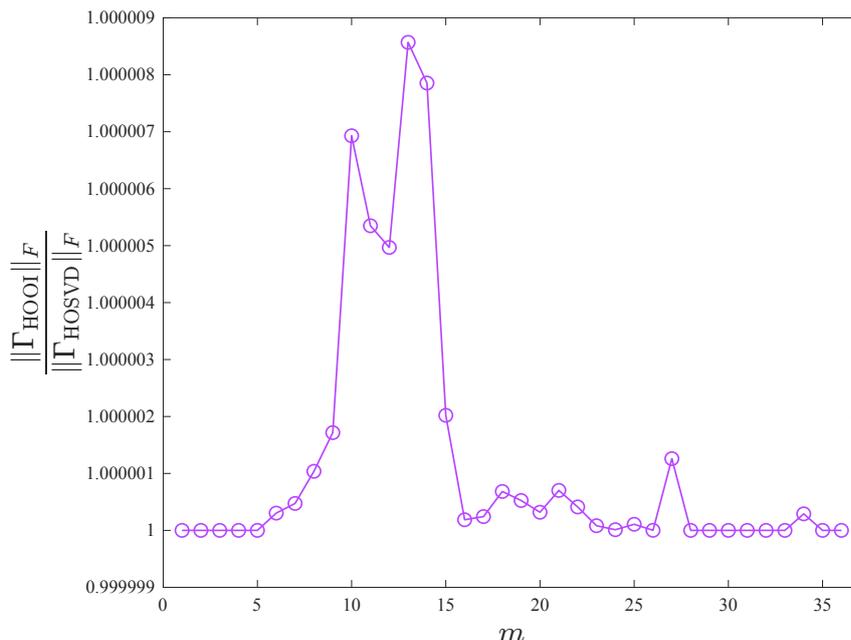


図 5.7: $D = 40$ における各繰り込みステップでの HOOI と HOSVD のノルムの比.

を $r = 7$ において粗視化ステップごとに示した. ただし, 次のステップで用いる繰り込まれたテンソルは, HOOI による近似のものを用いた. 図より, 各くりこみステップにおいて, HOOI は HOSVD よりも大きなノルムを達成する傾向にある事がわかる. この意味で, HOOI によるサブネットワークの近似は, 元のサブネットワークのノルムをよく近似している.

次に, $D = 40$ において HOOI を用いた各ステップごとの自由エネルギーと, HOSVD のみを用いた場合の自由エネルギーの収束値を図 5.8 に示す. 図より, HOOI は HOSVD のみを用いた場合よりも正確な Γ の近似を与えるにも関わらず, 自由エネルギーの計算では HOSVD のものよりも大きい場所へ収束するということがわかった. これは, 元の ATRG との誤差が HOSVD の場合よりも大きくなっていることを示している. これには次の二つの理由が考えられる.

- ・ HOOI は変分法により isometry を作成するので, 結果として元の ATRG が持つ特異値の対称性などを壊しており, 自由エネルギーの精度低下に繋がっている可能性がある.
- ・ HOSVD による近似がもとより自由エネルギーを過大評価しており, HOOI による近似がそれを修正している可能性がある.

しかしどちらであっても, HOOI による自由エネルギーの差異は Triad-ATRG の収束精度よりも十分小さく, また局所解を得るために反復回数も大きく取る必要があり, 計算コストの観点でも HOOI 表現を用いるのは現実的ではない. したがって Triad-ATRG における HOSVD による近似は, 既に十分な精度を持っており, 計算コストの観点からも妥当であると言える.

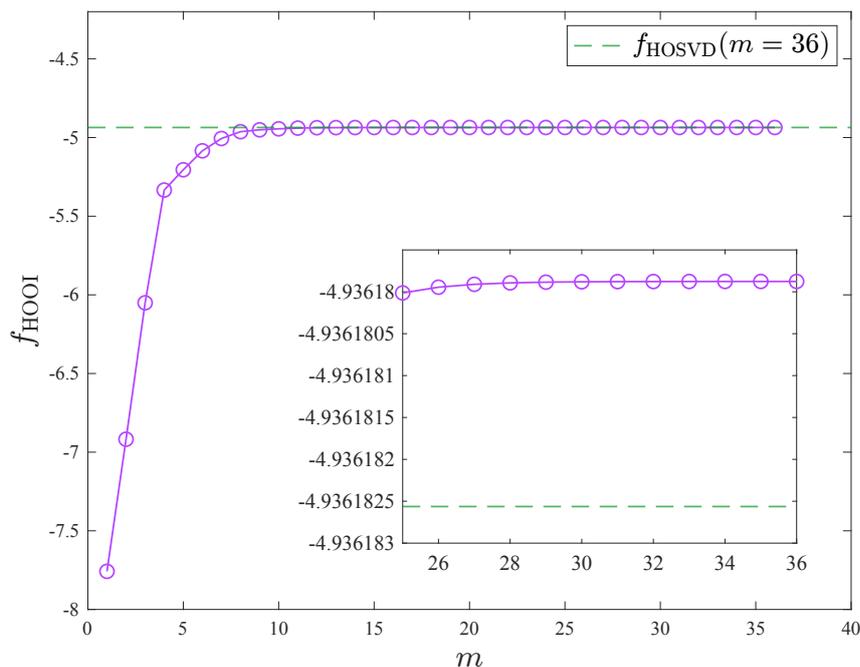


図 5.8: $D = 40$ における各繰り込みステップでの HOOI の自由エネルギー. 点線は HOSVD による自由エネルギーの収束値で, 右下はその拡大図である.

5.7 ボンド入れ替えのオーバーサンプリング

本節では, ボンド入れ替えのオーバーサンプリングの効果を検証する. 図 5.9 に, ATRG, Triad-ATRG, そして $r_{\text{bond}} = 5, r = 7$ とおいた ILO Triad-ATRG の自由エネルギーの結果を示す. ボンド入れ替えのオーバーサンプリングをした場合, すべてのボンド次元で ATRG よりも低い自由エネルギーを達成している. 特に $D = 40$ の結果であっても $D = 70$ とおいた Triad-ATRG よりも低い自由エネルギーであることは, 特筆すべき点である. このことから, ATRG において最も近似が激しいボンド入れ替えをオーバーサンプルすることは, 劇的な精度向上をもたらすことがわかる. しかしながら, 現在の手法の場合, 並列化できないボンド入れ替えの計算コストがボトルネックになり得る. 仮に $r = 1$ と取ってもボンド入れ替えの計算量は $O(r_{\text{bond}}^4 D^7)$ で固定されているので, 計算量は変わらない. したがって, 並列向けのボンド入れ替えの手法を開発することが今後必要となる.

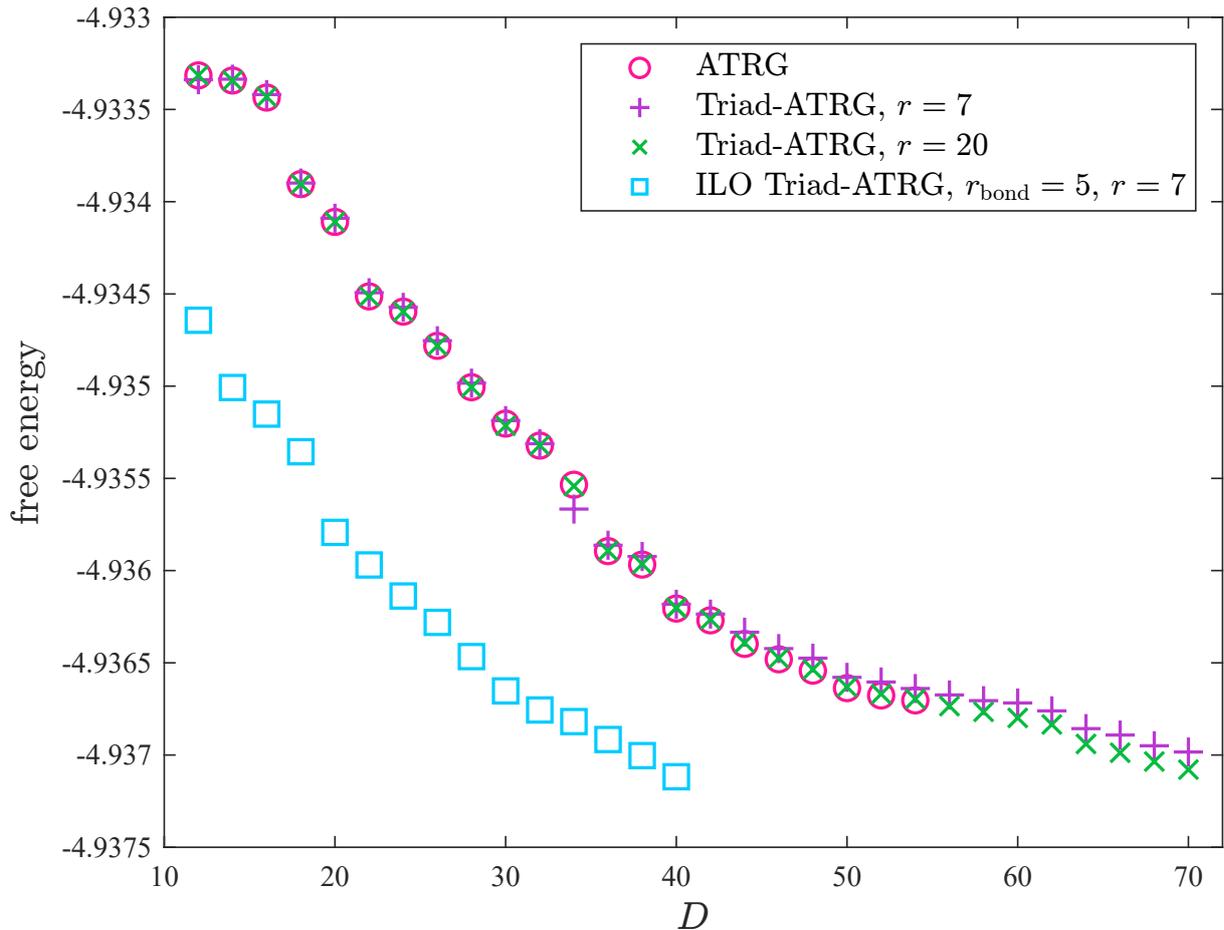


図 5.9: ILO Triad-ATRG と通常の Triad-ATRG($r_{\text{bond}} = 1$) の自由エネルギーの比較。

5.8 4次元イジングモデルの相転移の性質

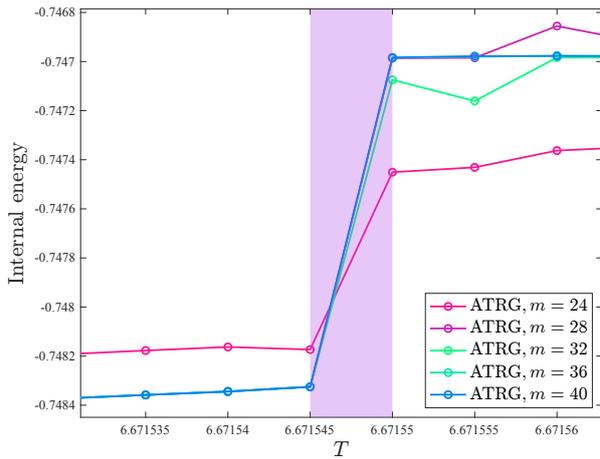
5.8.1 内部エネルギーの体積依存

本節では、4次元イジングモデルでの数値計算が厳密な結果と一致しているかどうかを調べた。まず、HOTRGによる先行研究 [13, 15] で指摘されていた、弱い一次相転移の様相が現れるかを調査する。厳密な結果 [75] によれば、このモデルは二次相転移を示す。先行研究では、内部エネルギーが相転移点でギャップを持つこと、及び内部エネルギーが相転移点で体積を増やすと交差する計算結果が得られていた。これは、弱い一次相転移の特徴である [92]。

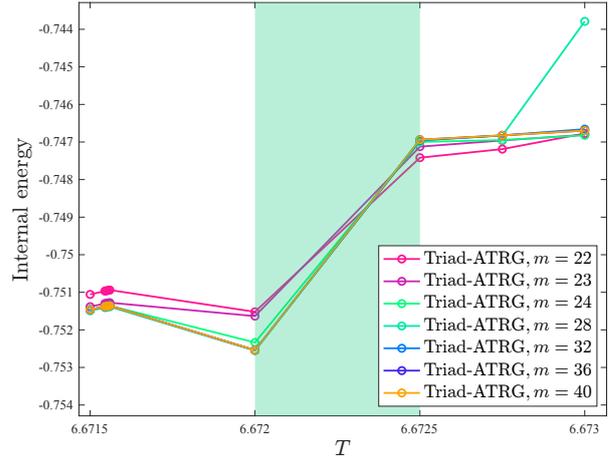
本研究では、まず $D = 50$ の ATRG と Triad-ATRG($D = 50, r = 20$) で内部エネルギーの体積依存性をしらべた。図 5.10a, 5.10b はそれぞれ、 $D = 50$ における ATRG, Triad-ATRG の内部エネルギーの体積依存性である。ともに体積を増やすにしたがって内部エネルギーの交差が起きていることがわかる。また、どちらも相転移点において有限な飛びが観測された。したがって、 $D = 50$ ではどちらの手法も先行研究と同じ弱い一次相転移を示していることがわかる。

図 5.11a, 5.11b は、 $D = 54$ における ATRG と Triad-ATRG($r = 20$) の内部エネルギーの体積依存性を示している。どちらのアルゴリズムでも、有限の内部エネルギーの飛びは起こっていないが、体積の増加による交差は同様に確認された。注意すべき点としてエネルギーの飛びは、温度の解像度が低いため捉えられていない可能性がある。

いずれにせよ、Triad-ATRG の内部エネルギーの体積依存性は、ATRG と同様の振る舞いを見せることが以上のことからわかる。これらを踏まえて、次に、Triad-ATRG を用いてボンド次元を

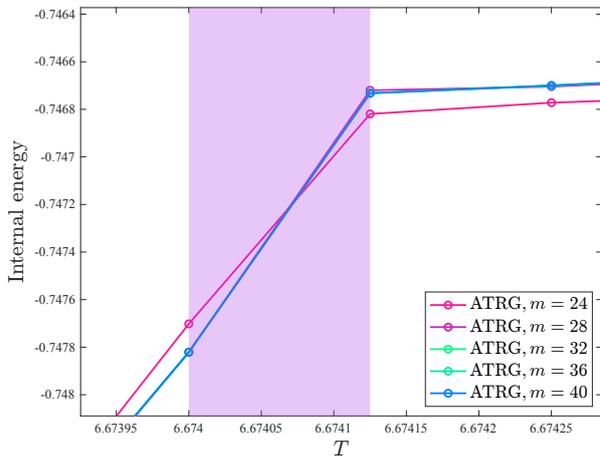


(a) $D = 50$ の ATRG における内部エネルギーの体積依存.

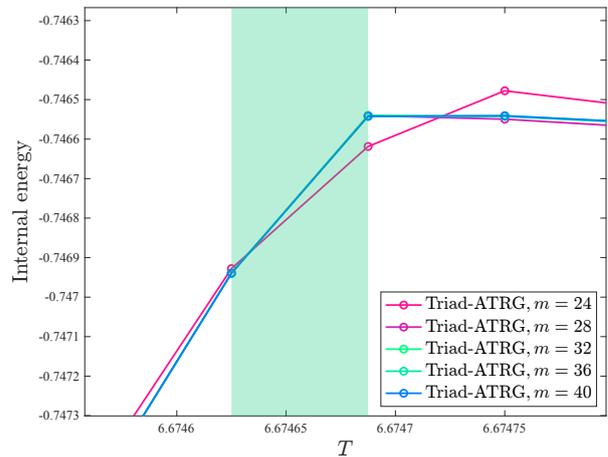


(b) Triad-ATRG ($D = 50, r = 20$) における内部エネルギーの体積依存.

図 5.10: 内部エネルギーの体積依存の比較



(a) ATRG, $D = 54$ における内部エネルギーの体積依存.



(b) Triad-ATRG ($D = 54, r = 20$) における内部エネルギーの体積依存.

図 5.11: 内部エネルギーの体積依存の比較

更に大きくした計算を行った. 図 5.12 に示すように, $D = 70, r = 20$ においては Triad-ATRG でも交差が起きていることが確認された. また, この場合でも $D = 50$ の場合のような有限のギャップはやはり観測されなかった.

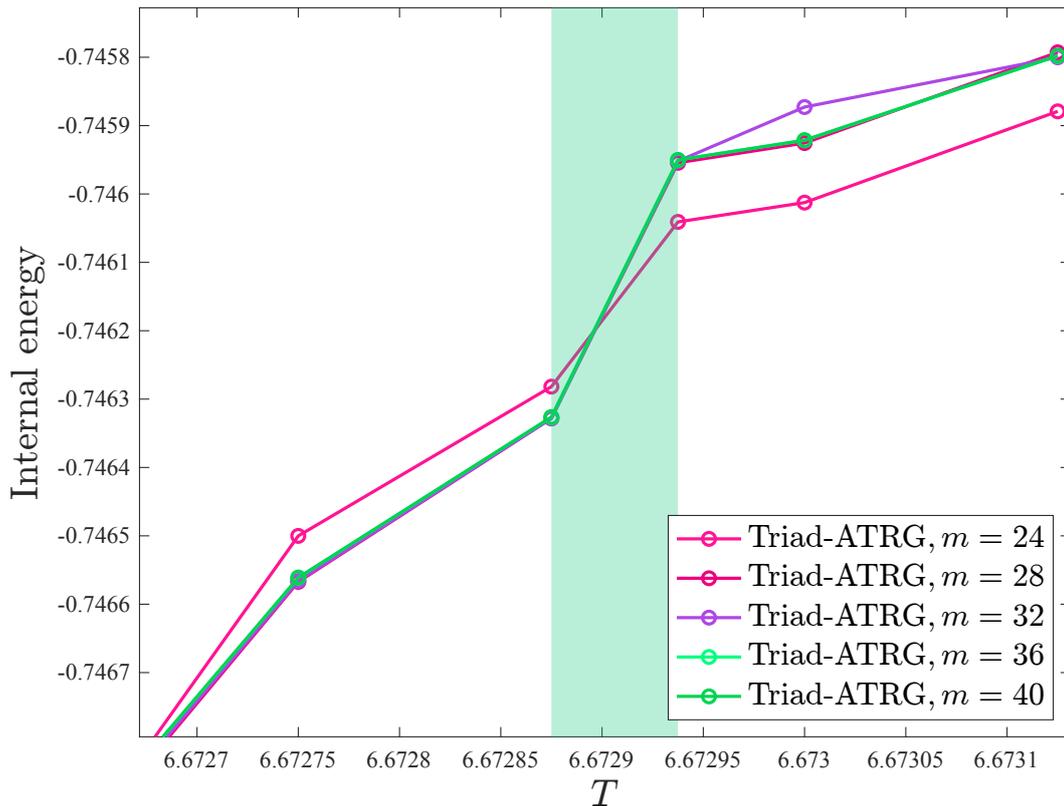


図 5.12: $D = 70$, $r = 20$ の Triad-ATRG における内部エネルギーの体積依存.

したがって $D = 70$, $r = 20$ でも有限のギャップは観測されていないが、体積の依存性は弱い一次相転移のものと一致していることがわかった。

5.8.2 臨界指数

厳密な結果によれば、4次元イジングモデルの臨界指数は、摂動論から予想される、式 (3.7-3.11) で表される対数の補正を除いて、平均場のものと一致する事が知られている [75–77]。この事実を用いて、現在の TRG 計算が正しい臨界指数を与えているかを調べる。

まず、対数補正がないと仮定した場合について考える。スケーリング理論によれば、磁化 m は逆温度 $\tau = (T - T_c)/T_c$ と磁場 h の関数として、

$$\frac{m}{h^{\frac{1}{\delta}}} = f\left(\frac{\tau}{h^{\frac{1}{\beta\delta}}}\right) \quad (5.3)$$

と書くことができる。ここで、 β, δ は臨界指数、 f はスケーリング関数である。ただし、 h は $f(0) = 1$ を満たすように規格化されている。式 (5.3) によれば、複数の h で計算した結果を正しく規格化すれば、1つのスケーリング関数上に乗るはずである。図 5.13 に、 $D = 50$ における Triad-ATRG ($r = 20$) において、平均場の臨界指数を仮定した場合と、平均場以外の臨界指数を仮定した場合のスケーリングの比較を示す。図より、平均場の臨界指数を仮定した場合は、スケーリング関数が h に依存しているのに対し、平均場からずれた臨界指数を仮定した場合は、全ての点が同じスケーリング関数上に乗っていることがわかる。すなわち、対数補正がないと仮定した場合には $D = 50$ では正しい臨界指数を計算できていないことがわかる。

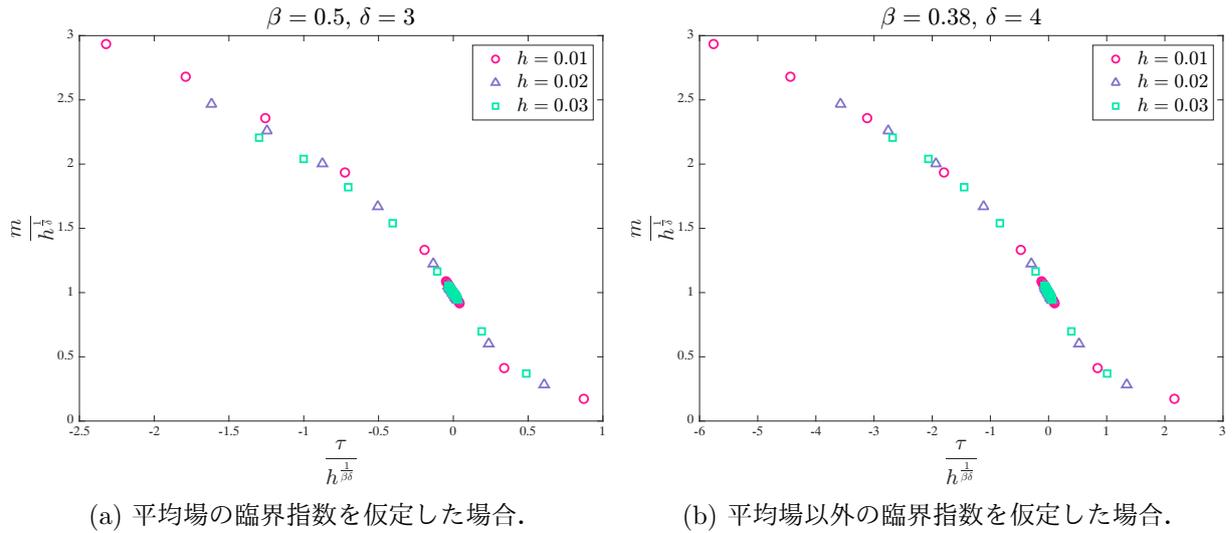


図 5.13: Triad-ATRG($D = 50, r = 20$) における scaling collapse の比較.

次に、式 (3.7-3.11) で定義される対数補正を考慮した場合について考える。対数補正を考慮したスケーリング関数は、 $\tau = |T - T_c|/T_c$ として

$$\frac{m}{h^{1-\frac{\gamma}{\Delta}} |\ln |h||^{\tilde{\gamma} + \frac{\gamma\tilde{\Delta}}{\Delta}}} = f\left(\frac{\tau^{\Delta} |\ln |\tau||^{\tilde{\Delta}}}{h}\right) \quad (5.4)$$

で表される [93]. ここで、 $\Delta = \frac{3}{2}, \gamma = 1$ である。また、Ising モデルが ϕ^4 理論と同じユニバーサルリティクラスに属すると仮定した場合、 $\tilde{\Delta} = 0, \tilde{\gamma} = \frac{1}{3}$ となる [93-95]. 図 5.14, 5.15 に、 $D = 50$ における Triad-ATRG($r = 20$) において、対数補正を仮定した場合と、対数補正を仮定しない場合の低温相でのスケーリングの比較を示す。図より、対数補正を仮定した場合と仮定しない場合では、対数補正を仮定した場合の方が同じスケーリング関数上に乗っていることが見て取れる。従って、 $D = 50$ においては、 ϕ^4 理論と同じ臨界指数を仮定した場合には、対数補正の存在を支持していることがわかる。

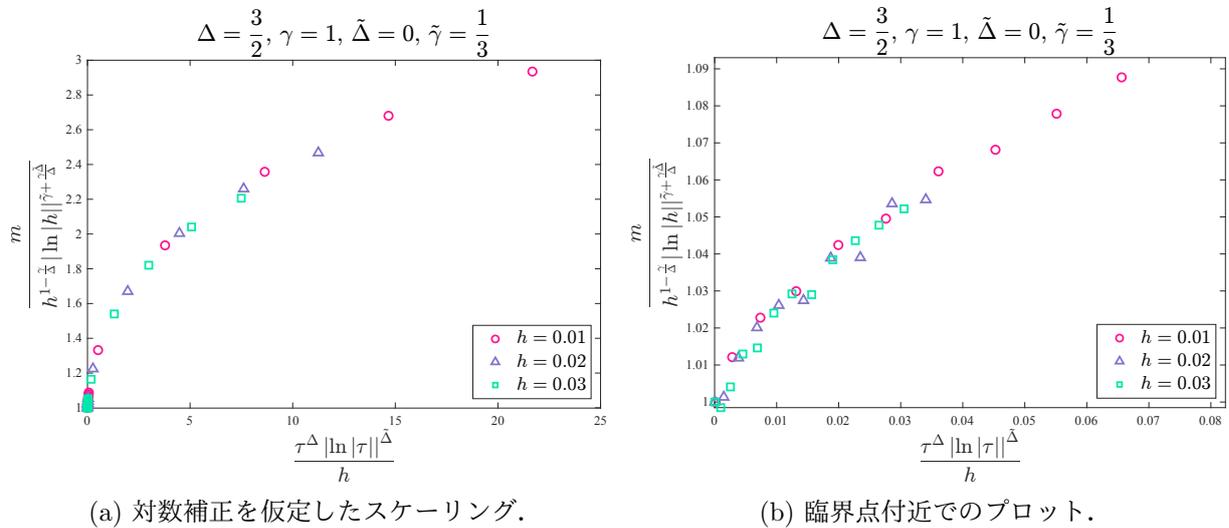


図 5.14: Triad-ATRG($D = 50$, $r = 20$) における対数補正を仮定した scaling collapse の比較.

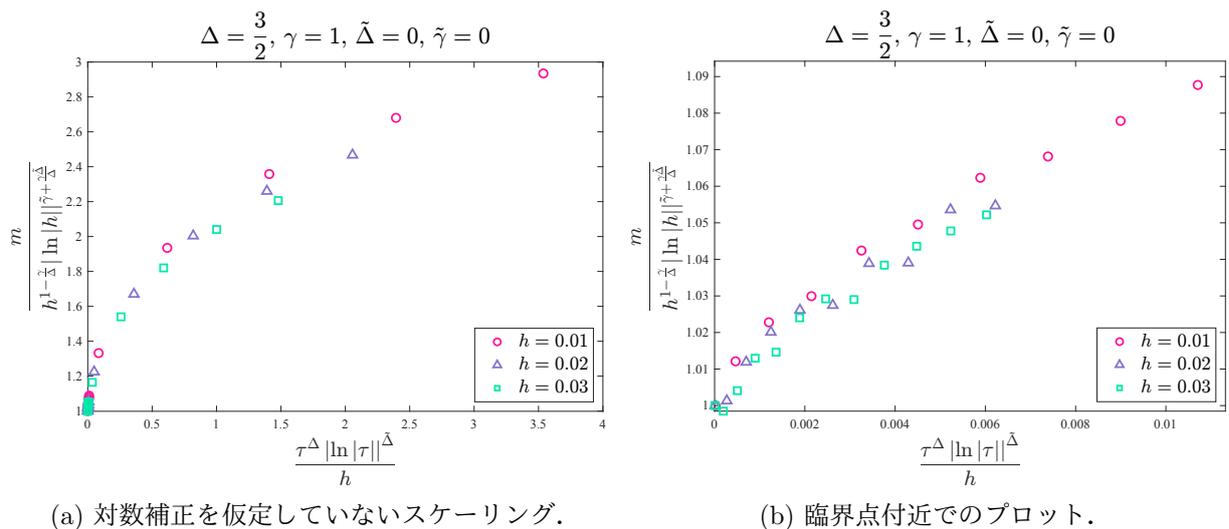


図 5.15: Triad-ATRG($D = 50$, $r = 20$) における対数補正を仮定していない scaling collapse の比較.

第6章 結論と考察及び展望

6.1 結論と考察

本研究では、有限密度格子 QCD の第一原理計算などへの応用が期待されているテンソル繰り込み群のアルゴリズムが、4次元において計算コストが増大するという問題に対して、4次元において広く用いられている ATRG の計算コストの大幅な削減方法として Triad-ATRG を提案した。

節 4.1 ではボンド入れ替え後のサブネットワーク Γ に対する HOSVD を考えることによって、Triad 表現を導入した。このとき、canonical form を用いて計算コストを削減し、並列計算向けのアルゴリズムを同時に提案した。Triad-ATRG は、ATRG の縮約ステップにおいて、内線を低いランクに近似することで、計算コストを $O(D^9) \rightarrow O(r^2 D^7)$ に削減した。これは、節 5.1 において、CPU での計算時間のスケーリングを測定することで実証することができた。

また、節 4.2 では、高次元におけるテンソル繰り込み群の GPU アルゴリズムを示し、節 5.2 において、ATRG、Triad-ATRG とともに大幅な計算コストを削減が可能であることを示した。特に、Triad-ATRG は $O(D^6)$ 以下のスケーリングを示しており、今後4次元系を GPU クラスタ上などで解析する上で、強力かつスタンダードな手法となる可能性を示唆している。

節 5.3 では、Triad-ATRG の自由エネルギーの精度を ATRG と比較し、Triad-ATRG 複数の r で ATRG と同等の精度を持つことを示した。また、節 5.4 では、 $D = 54, r = 20$ において相転移点が ATRG とほぼ同等の精度で求められることも確認された。また、高速化されたアルゴリズムによって、 $D = 70$ にまで自由エネルギーと相転移点を計算し、 $D > 54$ において相転移点のボンド次元に対する収束性の改善が確認された。

節 5.5 では、impurity テンソルを含む内部エネルギーの計算を行い、 $D = 54, r = 20$ において ATRG と同等の結果が得られることを示した。また、節 5.6 において、 Γ の HOSVD による Triad 表現が十分なサブネットワークの近似になっていることが確認された。従って、Triad-ATRG アルゴリズムは ATRG と同等の精度を持ち、かつ大幅な計算コストの削減を達成した。

節 4.3 では、従来の ATRG では行われていないボンド入れ替えのオーバーサンプルのアルゴリズムを、Triad-ATRG と組み合わせたアルゴリズムである ILO Triad-ATRG を提案し、節 5.7 において ATRG よりも大幅に低い自由エネルギーを達成できることを数値計算で示した。

上記に示した結果から、Triad-ATRG は4次元において計算のスケーリングを大幅に削減し、なおかつ自由エネルギー、物理量ともに ATRG に匹敵する十分な精度を持っていると言える。したがって従来の ATRG よりも効率的なアルゴリズムである。また、ボンド入れ替えのオーバーサンプリングが可能であるため、ATRG を超える高精度な計算を実現できることも明らかになった。これらの特性から、Triad-ATRG は将来の4次元系の計算における応用可能性が非常に高いと結論づけられる。

節 5.8 では、実際に上記で軽量化されたアルゴリズムである Triad-ATRG を用いて、4次元イジングモデルの相転移点の性質を調査した。先行研究 [75, 76] において $D = 13$ の HOTRG で示されていた弱い一次相転移の描像は、有限のギャップと体積依存性の交差として $D = 50$ の ATRG と Triad-ATRG ($r = 20$) において再現された。また、 $D = 54$ においては、ATRG と Triad-ATRG ($r = 20$) において有限のギャップは観測されなかったが、体積依存性は弱い一次相転移のものと同じであった。さらにボンド次元を上げ、 $D = 70$ において計算した結果、定性的振る舞い

は $D = 54$ と変わらなかった。

以上のことから, Triad-ATRG での内部エネルギーの体積依存性は二次相転移のものとは異なっている。また, 本研究で求められた臨界温度は, モンテカルロ法によるものとは大きく異なっている。これらのことは, 現在のテンソル繰り込み群計算では, 4次元 Ising モデルの性質を正確に捉えることができていない可能性, または対数補正などが内部エネルギーの体積依存性の計算に影響を与えている可能性が示唆される。これは, 過去の ATRG を用いた先行研究 [18] において, 強結合極限で Ising モデルに一致する ϕ^4 理論の計算において 3次元の二次相転移の場合は内部エネルギーが交差しないが, 4次元の二次相転移の場合では交差する例が存在することが知られていたが, 本研究によってそれが強く裏付けられた。

また, スケーリング関数を用いた臨界指数の計算では, 以下の二つの可能性が本研究の結果より示唆された。

- (1) 対数補正を仮定していない場合において, Triad-ATRG が間違った臨界指数を与えている可能性。
- (2) 対数補正を仮定した場合において, Triad-ATRG が正しいスケーリングと臨界指数を与えている可能性。

(1) のシナリオの場合, Triad-ATRG(及び ATRG) が相転移を正しく捉えられていない事を意味する。これは, $D = 50$ において相転移点がモンテカルロ法のものとは大きく異なること, 内部エネルギーの飛びがある事, 体積の増加による交差が存在する事, 臨界指数が厳密解から異なっている事から推測される。この場合, 精度の不足の原因として考えられるのは, ボンド次元の大きさの不足及び, 内部相関の取り扱いの不備である。ATRG のようなテンソル繰り込み群アルゴリズムは, 不要な内部相関をうまく除去できていないことが知られており, その影響の蓄積によって誤差が増大する。この影響が, 本研究や先行研究での計算結果に現れている可能性がある。この問題は特に, エンタングルメントエントロピーが増大する高次元において深刻であると考えられており [96, 97], 2次元系ほどの高精度を実現するには, よりボンド次元を大きくするか, Gilt [98] のような近距離相関を正しく繰り込むアルゴリズムを取り扱うことが重要であると考えられる。

(2) のシナリオの場合, Ising モデルには ϕ^4 理論と同じくスケーリングに対数補正が現れる可能性を強く示唆する。これは, Triad-ATRG の計算が $V = 1024^4$ の大体積で行われており, モンテカルロ法のような有限サイズスケーリングに頼らず, 熱力学極限での計算を行う事により, 対数補正を正確に捉えることができていない可能性がある。現在, 4次元の Ising モデルのモンテカルロ計算 [87] では, 最大 $V = 256^4$ の体積において, 有限サイズスケーリング仮説を用いて比熱を計算し, ϕ^4 理論と異なるユニバーサリティクラスであると結論付けられている。しかしながら, $V = 256^4$ は, Triad-ATRG の $V = 1024^4$ に比べて十分な体積であるとは言えず, また有限サイズスケーリングによるフィッティングでは対数補正を捉える事が難しく¹, スケーリング関数の仮定に依存する結果が得られているため, 有限サイズスケーリングによる解析が適切でない可能性がある。ただし, モンテカルロ法による計算では比熱の特異性のみを調査しているため, 本研究での磁化の計算との直接的な比較は難しいことは注意する必要がある。

以上では2つのシナリオを示したが, 現時点の臨界指数の計算では十分なデータ点数がなく, データのフィッティングを行うことが難しいため, どちらのシナリオであるかは明確にはわからない。また, 対数補正がある場合でも, ϕ^4 理論と Ising モデルが同じユニバーサリティクラスに属するという厳密な証明はないため, パラメーターの選択にも疑問が残る。しかしながら, 熱力学極限による数値計算において, 対数補正を支持する結果が得られたことは, Ising モデルのユニバーサリティクラスの問題に大きな進展を与える。

¹有限サイズスケーリングによる比熱の解析では, $(\ln L)^{\frac{1}{3}}$ のような項をフィッティングするが, $L = 1024$ と $L = 16$ でも 1.35 倍ほどの差しかない。

6.2 今後の展望

上記で述べたとおり, Triad-ATRG は 4 次元のテンソル繰り込み群計算を大幅に効率化するものであり, 更なる応用が期待される. 従って, 今後の展望として, フェルミオン系や, ゲージ系の応用が挙げられる. また, ボンド入れ替えのオーバーサンプリングは更なる精度向上を達成する事が示唆されたため, 更なるボンド次元の拡張を行い, 精度を調査する事が望ましい. そのために, ボンド入れ替えの並列化アルゴリズムを開発することは喫緊の課題である. また, Ising モデルの対数補正の有無を明確にするため, さらに大きなボンド次元において臨界指数の計算を行うことも重要である.

謝辞

本研究は、京都大学基礎物理学研究所のスーパーコンピューター「yukawa-21」および大阪大学D3センターのスーパーコンピューター「SQUID」を使用して行いました。これらの計算資源の提供に感謝いたします。

また、本論文の執筆にあたり、多くの方々にご支援とご助言をいただきました。ここに心より感謝申し上げます。

まず、指導教官である佐々木勝一准教授には、研究全般にわたり多大なご指導と温かいご助言を賜り、深く感謝いたします。

また、肥山教授をはじめとする原子核理論研究室の皆様には、研究室セミナーや議論の場で多くの有益なアドバイスをいただきました。この場を借りて厚く御礼申し上げます。

また、研究を進めるにあたり、金沢大学の武田様、相澤様、早崎様、理化学研究所の中山様、筑波大学の藏増様、秋山様から多くの貴重な議論と助言をいただきました。特に、秋山様にはアルゴリズム、数値計算の面や、イジング模型の議論などで数々のご指導を賜り、心より感謝いたします。

本研究は、東北大学宇宙創成物理学国際共同大学院プログラム (GP-PU) に参加して進めたものであり、GP-PU からの経済的支援に深く感謝いたします。

最後に、研究を支えてくれた家族には、温かい理解と支援に深く感謝します。

A 特異値分解に関する定理の証明

A.1 Theorem 1 の証明

この証明は主に [99] を参考にした。

Proof. まず, $A^\dagger A$ が半正定値行列であることを示す. $A^\dagger A$ はエルミート行列であり, ユニタリ対角化が可能である. その固有値と固有ベクトルをそれぞれ λ_i, \mathbf{v}_i ($i = 1, \dots, n$) とおくと,

$$\lambda_i = \langle \mathbf{v}_i, A^\dagger A \mathbf{v}_i \rangle = \langle A \mathbf{v}_i, A \mathbf{v}_i \rangle = |A \mathbf{v}_i|^2 \geq 0 \quad (\text{A.1})$$

が成り立つので, $A^\dagger A$ は半正定値行列である. したがって $\sigma_i = \sqrt{\lambda_i}$ をすべての固有値に対して定義することができる. ここで, 次元定理より,

$$\begin{cases} \text{rank} A + \dim(\text{Ker} A) = n \\ \text{rank} A^\dagger A + \dim(\text{Ker} A^\dagger A) = n \end{cases} \quad (\text{A.2})$$

であり, 二つの行列の核について考えると, あるベクトル \mathbf{x} に対して,

- $A \mathbf{x} = 0 \Rightarrow A^\dagger A \mathbf{x} = 0$ は自明.
- $A^\dagger A \mathbf{x} = 0 \Rightarrow A \mathbf{x} = 0$ は, $0 = \langle \mathbf{x}, A^\dagger A \mathbf{x} \rangle = |A \mathbf{x}|^2$ より従う.

より,

$$A \mathbf{x} = 0 \Leftrightarrow A^\dagger A \mathbf{x} = 0 \quad (\text{A.3})$$

なので, $\dim(\text{Ker} A) = \dim(\text{Ker} A^\dagger A)$ が成り立つので, 式 (A.2) より, $\text{rank} A = \text{rank} A^\dagger A$ を得る. 対角化可能な行列の非ゼロの固有値の数は, 階数に一致するので r は $A^\dagger A$ の非ゼロ固有値の数である. したがって,

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0 \quad (\text{A.4})$$

として σ_i を再定義し,

$$\mathbf{u}_i = \frac{1}{\sigma_i} A \mathbf{v}_i \quad (1 \leq i \leq r) \quad (\text{A.5})$$

と定義すると,

$$A A^\dagger \mathbf{u}_i = \sigma_i^2 \mathbf{u}_i \quad (\text{A.6})$$

より, \mathbf{u}_i は $A A^\dagger$ の固有ベクトルである. また, $\text{rank} A = \text{rank} A^\dagger = \text{rank} A A^\dagger$ より, $A A^\dagger$ の非ゼロの固有値も r 個である. また,

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \frac{1}{\sigma_i \sigma_j} \langle \mathbf{v}_i, \sigma_j^2 \mathbf{v}_i \rangle = \delta_{ij} \quad (\text{A.7})$$

より, $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ は \mathbb{C}^r 上の正規直行基底である. 特異値と \mathbf{u}_i の定義, 式 (A.3) より,

$$A\mathbf{v}_i = \begin{cases} \sigma_i \mathbf{u}_i & (1 \leq i \leq r) \\ \mathbf{0} & (r+1 \leq i \leq n) \end{cases} \quad (\text{A.8})$$

であるので, A は正規直行基底 $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ をそれぞれ $\{\mathbf{u}_1, \dots, \mathbf{u}_r, \mathbf{0}, \dots, \mathbf{0}\}$ に射影するから, A のスペクトル分解は,

$$A = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\dagger \quad (\text{A.9})$$

で与えられる. また, 正規直行基底 $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ を拡張して, $\{\mathbf{u}_1, \dots, \mathbf{u}_m\}$ を用いれば¹, $U = (\mathbf{u}_1 \cdots \mathbf{u}_m)$, $V = (\mathbf{v}_1 \cdots \mathbf{v}_n)$ と定義でき, 式 (A.9) の行列表現として式 (2.1) を得る. \square

A.2 Theorem 2 の証明

Proof. この証明では, von neumann's trace inequality [100]

$$|\text{tr}[AB]| \leq \sum_{i=1}^{\min(m,n)} \sigma_i^A \sigma_i^B \quad (\text{A.10})$$

を用いる. ただし σ_i^A, σ_j^B はそれぞれ A, B の特異値である. これを用いると, $\text{rank} B = k < r$ の行列 B において,

$$\begin{aligned} \|A - X\|_F^2 &= \|A\|_F^2 - 2\langle A, B \rangle + \|B\|_F^2 \\ &\geq \sum_{i=1}^r [(\sigma_i^A)^2 - 2\sigma_i^A \sigma_i^B + (\sigma_i^B)^2] \\ &= \sum_{i=1}^k (\sigma_i^A - \sigma_i^B)^2 + \sum_{i=k+1}^r (\sigma_i^A)^2 \end{aligned} \quad (\text{A.11})$$

が成り立つ. ただし $\sigma_{k+1}^B = \dots = \sigma_m^B = 0$ であることを用いた. 式 (A.11) を最小化するのは,

$$\sigma_i^A = \sigma_i^B, \quad (1 \leq i \leq k)$$

が成立するときである. 式 (2.9) の X はこの条件を満たす. \square

¹ $(\mathbf{u}_1, \dots, \mathbf{u}_r)^\dagger \mathbf{x} = \mathbf{0}$ の解を直行化すればよい.

B 繰りこみ群とスケーリング則

本章では、繰りこみ群とスケーリング則について、簡潔にまとめる。本章は主に [101] を参考にしている。繰りこみ変換とは、系のスケールを変換するような写像である。例として、以下のような結合定数で貼られる理論の集合 S を考える。

$S =$ (ある対称性を満たすような運動項が規格化された作用積分の集合で、カットオフが Λ のもの)

スケール t の繰りこみ変換 R_t は、 $S \in S$ としたときに、次のような性質を満たす写像である。

$$(1) R_t : S \rightarrow S_t \quad (S_t \in S)$$

$$(2) R_t(R_s(S)) = R_{t+s}(S)$$

$$(3) Z[S] = Z[S_t]$$

$$(4) \xi(S_t) = \frac{\xi(S)}{e^t}$$

ただし、 $Z[S]$ は理論 S の分配関数、 $\xi(S)$ は理論 S の相関長である。 R_t は、 t 倍だけ系のスケールを粗視化するような理論である。 R_t は (半) 群をなすので、この群を繰りこみ群と呼ぶ。

具体的な繰りこみ群の例として、2次元のスピンに対して、ブロックスピン変換を考えてみる。図 1.1 のように、隣接した N 個のスピンを平均を取り、新たな繰りこまれたスピンを定義する。例えば、

$$\phi_n^B = \frac{1}{NC_B} \sum_{i,j=0}^N \phi_{n+i\hat{1}+j\hat{2}} \quad (\text{B.1})$$

のようにして新たなスピンを定義する。 C_B は規格化因子である。この操作がスケール N の繰りこみ変換の定義を満たしていることは、容易に確認できる。例えば、分配関数は、

$$Z = \int D\phi e^{-S[\phi]} \quad (\text{B.2})$$

$$= \int D\phi D\phi^B \delta\left(\phi_n^B - \frac{1}{N^2 C_B} \sum_{i,j=0}^N \phi_{n+i\hat{1}+j\hat{2}}\right) e^{-S[\phi]} \quad (\text{B.3})$$

$$= \int D\phi^B e^{-S_t[\phi^B]} \quad (\text{B.4})$$

のように普遍に保たれている¹。繰りこみ変換で新たにできた理論 S_t は ϕ^B で記述される有効理論である。繰りこみ変換では、高エネルギー (近距離) の情報は失われてしまうが、低エネルギーでは同じ物理を記述する²。

上記では繰りこみ変換の例として、ブロックスピン変換を取り扱ったが、高運動量の成分を直接積分することによって、連続的な繰りこみ変換も容易に構成できる。したがって、 t を連続変数

¹このようにスピンの数が $1/N^2$ になった新たな粗い格子を作成した場合、物理的な距離は普遍であるので、格子間隔で数えた場合には元の系の相関長は新たな格子の相関長の N 倍になることも容易にわかる。したがってこの例では $t = N$ である。

²実際、相関関数などの期待値は両者の理論で一致する。

として, R_t を定義することができる. t をあるパラメーターとみなせば, 繰り込み変換に対する微分方程式を書き下すことが可能である. いま, 微小な繰り込み変換 $R_{\delta t}$ を考える. このとき,

$$R_{\delta t}(S_t) = S_{t+\delta t} \quad (\text{B.5})$$

であるので, $\delta t \rightarrow 0$ として,

$$G(S_t) = \lim_{\delta t \rightarrow 0} \frac{R_{\delta t}(S_t) - S_t}{\delta t} \quad (\text{B.6})$$

とすると,

$$\frac{dS_t}{dt} = G(S_t) \quad (\text{B.7})$$

という微分方程式が得られる. これを繰り込み群方程式と呼ぶ. この方程式は, あるパラメーター t に沿った繰り込み群の流れを与える. Wilson は, 相関長が無限の部分空間である臨界面 S_{cr} の存在を仮定した. すなわち,

$$S_{\text{cr}} = \{S \mid \xi(S) = \infty\} \quad (\text{B.8})$$

である. ただし簡単のために臨界面は一つであると仮定した. この臨界面上の理論は, 繰り込み変換によっても臨界面上に留まることが容易にわかる. すなわち

$$R_t(S) \in S_{\text{cr}} \quad \text{for any } S \in S_{\text{cr}} \quad (\text{B.9})$$

である. この臨界面上の理論に無限回の繰り込み変換を行うと, ある繰り込み普遍的な固定点 S^* に収束する.

$$\lim_{n \rightarrow \infty} R_t^n(S) = S^* \quad \text{for any } S \in S_{\text{cr}} \quad (\text{B.10})$$

すなわち, 臨界面上の理論は実質的に S^* によって記述される. この固定点は繰り込み群方程式 (B.7) の不動点である. 固定点の周りで方程式を線形化することを考えよう. いま, 固定点近くの理論を $S_t = S^* + \delta S_t$ とおき, 方程式を変形すると,

$$\frac{d}{dt} \delta S_t = L^*(\delta S_t) \quad (\text{B.11})$$

となる. ただし L^* は G のヤコビ行列である. L^* の固有値を λ_k , 固有汎函数を O_k とおくと,

$$S_t = S^* + \sum_k g_k(0) e^{\lambda_k t} O_k \quad (\text{B.12})$$

という解を得る. ただし, $S_t = \{g_1(t), g_2(t), \dots, g_n(t)\}$ という表現を用いた. 式 (B.12) から, 以下のように固有解の分類を行うことができる.

- relevant: $\lambda_k > 0$ のとき, 繰り込み変換によって固定点から遠ざかる解を記述する.
- irrelevant: $\lambda_k < 0$ のとき, 繰り込み変換によって固定点に収束する解を記述する.
- marginal: $\lambda_k = 0$ のとき, 繰り込み変換によって (固定点近くで) 変化しない解を記述する.

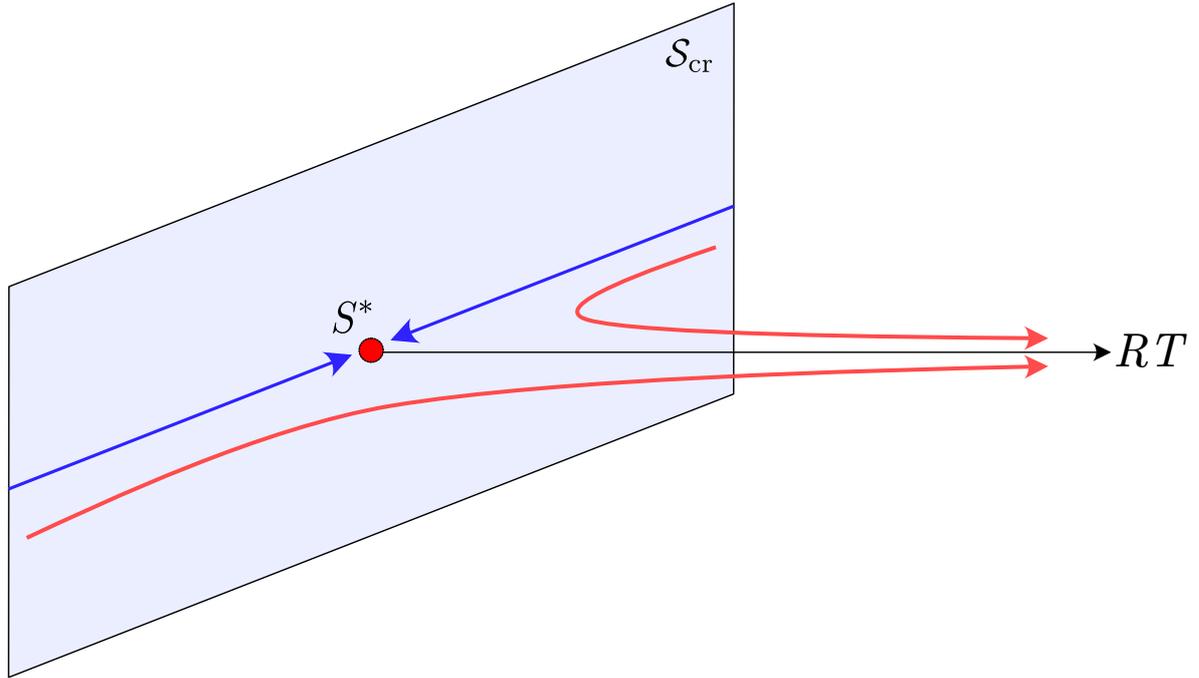


図 B.1: 繰りこみ群の流れの一例.

したがって、臨界面は relevant パラメーターがすべて 0 となる部分空間であることがわかる。臨界面付近の理論に繰りこみ群変換を繰り返すと、relevant な固有値が支配的となって、irrelevant なパラメーターがすべて 0 の繰り込まれた軌跡 (Renormalized Trajectory) に沿って臨界点から遠ざかる。この relevant パラメーターは、Ising モデルでは臨界温度 $\tau = \frac{T-T_c}{T_c}$ や h に対応する。繰りこみ変換は低エネルギーの有効理論を記述するから、irrelevant なパラメーターは、この意味で低エネルギーで支配的ではない。図 B.1 に繰りこみ群の流れの一例を示した。青い面は臨界面であり、青い矢印は臨界面上から出発した繰りこみ群の流れ、赤い矢印は臨界点近傍から出発した繰りこみ群の流れを示している。RT は固定点から湧き出す流れに乗っている。

いま、固定点に理論を近づけていった時のスケーリングについて考える。繰りこみ変換は相関長を $\xi[S_t] = e^{-t}\xi[S]$ すなわち、長さのスケールを e^{-t} 倍にする。系の単位体積あたりの自由エネルギー $\beta f = -\frac{1}{V} \ln Z[S]$ は、次元解析により、繰りこみ変換によって次のようにスケーリングすることがわかる。

$$(\beta f)[S_t] = e^{td}(\beta f)[S] \quad (\text{B.13})$$

いま、式 (B.12) の標識を用いると、

$$\beta f(g_1(0), \dots, g_n(0)) = e^{-td} \beta f(g_1(t), \dots, g_n(t)) \quad (\text{B.14})$$

というようにスケールする。irrelevant なパラメーターは t をいくらでも大きくとることで 0 に収束するから、有効なパラメーターは relevant なものだけである。

いま、Ising 模型のように relevant なパラメーターが 2 つの場合を考える。 $\lambda_1, \lambda_2 > 0$ とし、 $g_1(0) = \tau$, $g_2(0) = h$ とし、 $t = -\frac{1}{\lambda_1} \ln |\tau|$ とおく。³この時、相関長 ξ は、 $h = 0$ において、

$$\xi(\tau, 0) = |\tau|^{\frac{1}{\lambda_1}} (\beta f)(\text{sgn}(\tau), 0) \quad (\text{B.15})$$

³ $\tau \rightarrow 0$ に近づける際に、繰り込まれた結合定数を一定に (臨界的な領域から脱出できるように) 取るように t を調整する。

というスケーリング則を示すことがわかる。すなわち,

$$\nu = \frac{1}{\lambda_1} \quad (\text{B.16})$$

であることがわかる。非熱 $C = -\frac{\partial(\beta f)}{\partial \tau^2}$ は,

$$C(\tau, 0) = |\tau|^{-2+\frac{d}{\lambda_1}} (\beta f)(\text{sgn}(\tau), 0) \quad (\text{B.17})$$

とスケールするので,

$$\alpha = 2 - \frac{d}{\lambda_1} \quad (\text{B.18})$$

である。また、磁化 $m = \frac{\partial f}{\partial h}|_{h=0}$ は,

$$m(\tau, 0) = |\tau|^{\frac{d}{\lambda_1} - \frac{\lambda_2}{\lambda_1}} (\beta f)(\text{sgn}(\tau), 0) \quad (\text{B.19})$$

のようにスケールするから,

$$\beta = \frac{d - \lambda_2}{\lambda_1} \quad (\text{B.20})$$

が得られる。同様に,

$$\gamma = \frac{2\lambda_2 - d}{\lambda_1} \quad (\text{B.21})$$

を得る。また、 $t = -\frac{1}{\lambda_2} \ln |h|$ とおけば、同様に

$$\beta f(\tau, h) = |h|^{\frac{d}{\lambda_2}} \beta f(\tau |h|^{-\frac{\lambda_1}{\lambda_2}}, \text{sgn}(h)) \quad (\text{B.22})$$

のようにスケールするので,

$$\delta = \frac{\lambda_2}{d - \lambda_2} \quad (\text{B.23})$$

をえる。これらから、スケーリング則

$$\alpha + 2\beta + \gamma = 2 \quad (\text{B.24})$$

$$\gamma = \beta(\delta - 1) \quad (\text{B.25})$$

が導かれる。すなわち、臨界指数は、 λ_1, λ_2 によって決まる。

また、式 (5.3) は、式 (B.22) より、

$$m = |h|^{\frac{d}{\lambda_2} - 1} f(\tau |h|^{-\frac{\lambda_1}{\lambda_2}}, \text{sgn}(h)) \quad (\text{B.26})$$

$$= |h|^{\frac{1}{\delta}} \left(\frac{\tau}{|h|^{\frac{1}{\beta\delta}}}, \text{sgn}(h) \right) \quad (\text{B.27})$$

のように証明できる。

C TRG アルゴリズムの補足

TRG アルゴリズムは、分配関数などのテンソルネットワークを繰り込み、一つのテンソルに近似し、そのトレースを取るというものであった。しかしながら実際には、有限のノルムを持つテンソル T を繰り込んでいくと、数値計算がオーバーフローしてしまう。そこで、各ステップごとにテンソルを規格化する必要がある。規格化として、テンソルのフロベニウスノルムを用いることにすると、

$$\tilde{T}^{(m)} = \frac{T^{(m)}}{N_m} \quad (\text{C.1})$$

のように規格化する。ただし、 $N_m = \|T^{(m)}\|_F$ である。すると、 n 回の粗視化後の 1 サイトあたりの自由エネルギーは、

$$f^{(n)} = -T \frac{\ln Z^{(n)}}{2^n} = -T \frac{\ln \left((N_0)^{2^n} (N_1)^{2^{n-1}} \cdots (N_{n-1})^2 \times \text{tTr}[T^{(n)}] \right)}{2^n} \quad (\text{C.2})$$

$$= -T \left(\sum_{m=0}^{n-1} \frac{N_m}{2^m} + \frac{\text{tTr}[T^{(n)}]}{2^n} \right) \quad (\text{C.3})$$

のように表せる。

impurity テンソルが含まれている場合は以下のように計算する。 $N'_m = \|S\|_F$ とすると、

$$\langle \phi_k \rangle = \frac{1}{Z} \text{tTr} \left[\sum_{\{i\}} S_{i_1(k) \dots i_d(k) i_1(k-\hat{1}) \dots i_d(k-\hat{d})} \prod_{n \neq k} T_{i_1(n) \dots i_d(n) i_1(n-\hat{1}) \dots i_d(n-\hat{d})} \right] \quad (\text{C.4})$$

$$= \frac{N'_0 (N_0)^{2^n-1} N'_1 (N_1)^{2^{n-1}-1} \cdots N'_{n-1} N_{n-1} \times \text{tTr}[S^{(n)}]}{(N_0)^{2^n} (N_1)^{2^{n-1}} \cdots (N_{n-1})^2 \times \text{tTr}[T^{(n)}]} \quad (\text{C.5})$$

$$= \frac{N'_1 N'_2 \cdots N'_{n-1}}{N_1 N_2 \cdots N_{n-1}} \times \frac{\text{tTr}[S^{(n)}]}{\text{tTr}[T^{(n)}]} \quad (\text{C.6})$$

と計算できる。

D 乱択化特異値分解

通常特異値分解を用いた数値計算において必要とされるのは、すべての特異値ではなく、上位のいくつかの特異値だけである。一般に Full SVD は多くの計算コストを必要とする。例えば $A \in \mathbb{C}^{m \times n}$, $m > n$ の行列の全ての特異値を求めるには $O(mn^2)$ の計算量が必要である。しかし例えば上位 k 個の特異値のみが必要な場合、Arnoldi 法等を用いれば $O(mn\chi)$ の計算量になる。これは $\chi \ll m$ の時には大きな計算時間の短縮になる。さらにテンソル繰りこみ群では、大きなメモリを必要とする中間テンソルを経由せずに、メモリ、計算量を抑えて Truncated SVD を実行することが可能であるため、非常に有用である。その中でテンソル繰りこみ群計算において得に有用なのが、乱択化特異値分解 (Randomized Singular Value Decomposition:RSVD) [102] である。RSVD ではまず、大きな特異値の貼る部分空間を QR 分解の反復によって求め、 $\|A - QQ^\dagger A\|_F$ を最小化する Q を求め、行列のサイズを下げてから SVD を実行する。そのアルゴリズムを Algorithm 2 に示す。Arnoldi 法と異なりすべての操作が行列行列積で行われるため、BLAS などのパッケージの恩恵を受けやすい。また、一連の操作はターゲット行列 A がテンソルの縮約で表されていても行うことができることに注意する。トータルの計算コストは $O(qmn\chi)$ となっている。

Algorithm 2 RSVD

input: $A \in \mathbb{C}^{m \times n}$, 反復回数 q , オーバーサンプリングパラメーター p

output: A のランク r の Truncated SVD $A \simeq U\Sigma V^\dagger$

initialize : $l = \chi + p$ として、ランダム行列 $\Omega \in \mathbb{C}^{n \times l}$ を生成.

$Q \leftarrow \text{qr}(A\Omega)$

for $n = 1, \dots, q$ **do**

$\Omega \leftarrow \text{qr}(A^\dagger Q)$

$Q \leftarrow \text{qr}(A\Omega)$

end for

$U', \Sigma, V^\dagger = \text{svd}(Q^\dagger A)$

$U \leftarrow QU'$

return U, Σ, V^\dagger

E 2次元での TRG アルゴリズム

TRG 法は2次元では計算コストが比較的軽く、様々なアルゴリズムを選択することが可能である。その中で現在最も広く用いられているアルゴリズムを紹介する。

E.1 Bond-weighted TRG

Bond-Weighted TRG はオリジナルの Levin-Nave のテンソル繰り込み群の改良版として考案された [103]。そのアルゴリズムはとてもシンプルで、式 (2.39) において、あるハイパーパラメーター k による特異値の分配の比率を行う。

$$S_{i_1(n)i_2(n)\alpha}^{[1]} = U_{i_1(n)i_2(n)\alpha}^{\text{even}} (S_{\alpha\alpha}^{\text{even}})^{\frac{1-k}{2}} \quad (\text{E.1})$$

$$S_{i_1(n-1)i_2(n-2)\alpha'}^{[2]} = (S_{\alpha'\alpha'}^{\text{even}})^{\frac{1-k}{2}} V_{i_1(n-1)i_2(n-2)\alpha'}^{\text{even}} \quad (\text{E.2})$$

$$B_{\alpha\alpha'}^{[12]} = \delta_{\alpha\alpha'} (S_{\alpha\alpha}^{\text{even}})^k \quad (\text{E.3})$$

$$S_{i_1(n-1)i_2(n)\beta}^{[3]} = U_{i_1(n-1)i_2(n)\beta}^{\text{odd}} (S_{\beta\beta}^{\text{odd}})^{\frac{1-k}{2}} \quad (\text{E.4})$$

$$S_{i_1(n)i_2(n-2)\beta'}^{[4]} = (S_{\beta'\beta'}^{\text{odd}})^{\frac{1-k}{2}} V_{i_1(n)i_2(n-2)\beta'}^{\text{odd}} \quad (\text{E.5})$$

$$B_{\beta\beta'}^{[34]} = \delta_{\beta\beta'} (S_{\beta\beta}^{\text{odd}})^k \quad (\text{E.6})$$

ここで $B^{[12]}$, $B^{[34]}$ はボンド上に定義された重みである。この分解は $k=0$ とした場合、Levin-Nave の TRG と一致することに注意する。ダイアグラムでこの分解を表すと、図 E.1 のようになる。

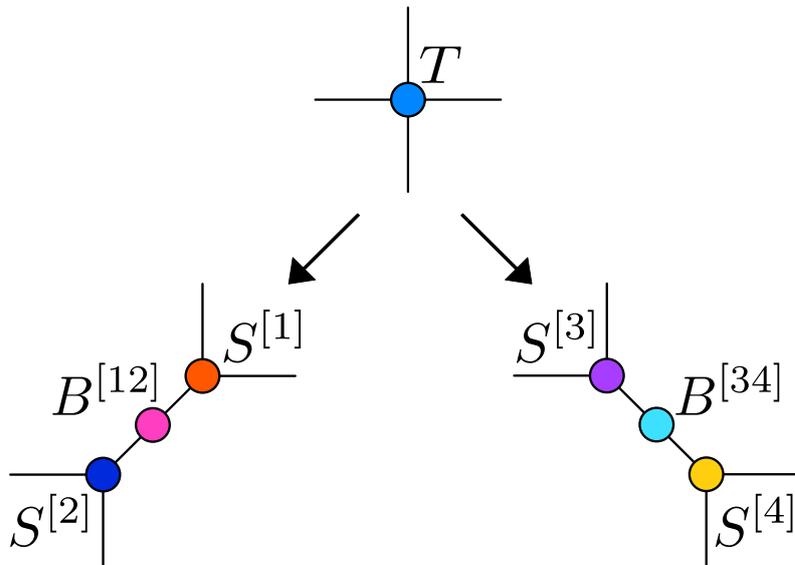


図 E.1: BTRG の分解.

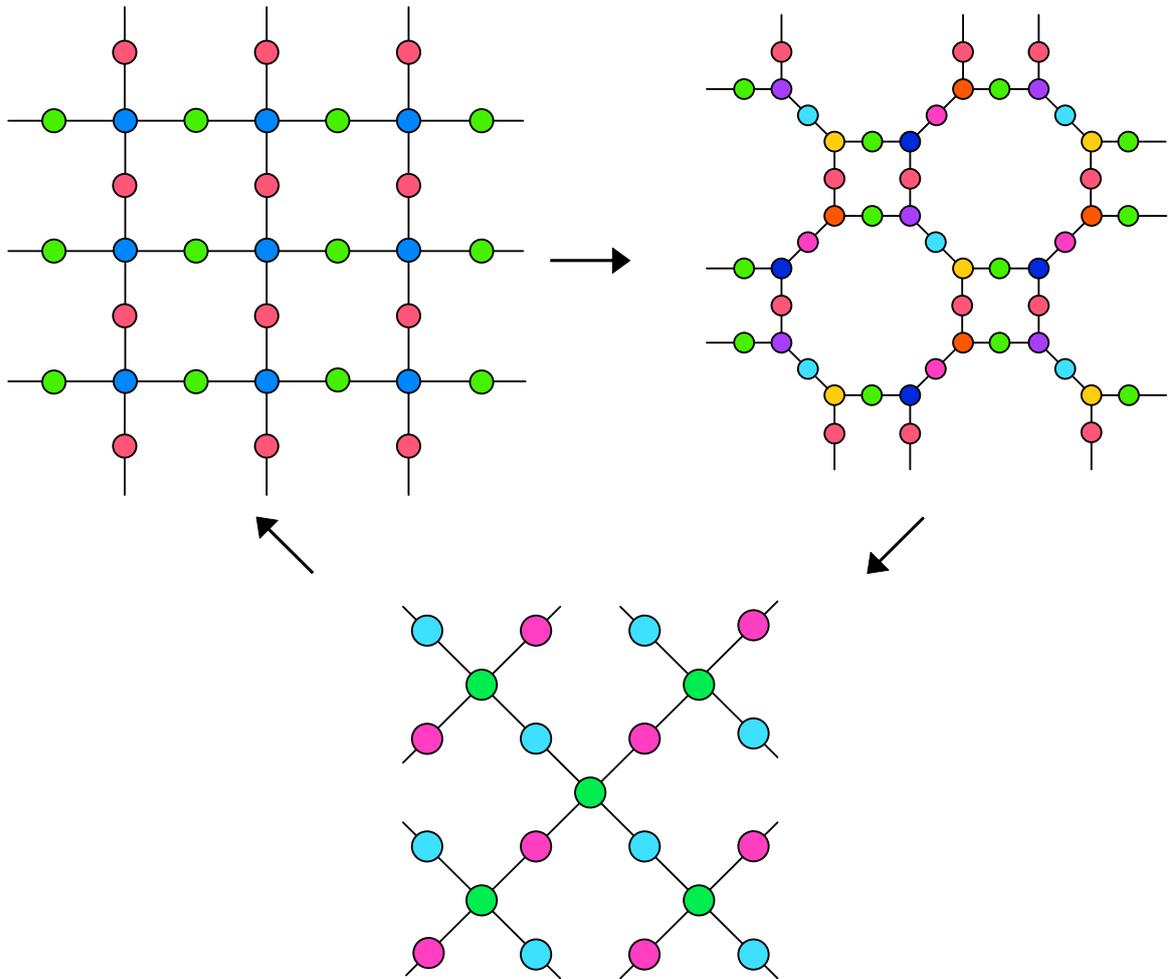


図 E.2: BTRG のアルゴリズムの流れ

分解後，Levin-Nave TRG 同様，一つ前のステップで作成したボンド重みを含めた縮約を行い新たなテンソルを作成する．ボンド重みは，SRG [32]などで用いられる，環境テンソルを平均場で近似する手法 [104] と類似したものである．BTRG のアルゴリズムの流れを図 E.2 に示す．ハイパーパラメーター k は，スケール不変なテンソルを得るための以下の定常条件を満たすように選択される．

$$\left(S^{\frac{1-k}{2}}\right)^4 \left(S^k\right)^4 = S \quad (\text{E.7})$$

これを満たす k は， $k = -\frac{1}{2}$ である．BTRG の計算コストは，オリジナルと変わらず $O(D^5)$ であり，Levin-Nave TRG や HOTRG よりも正確に自由エネルギーを計算することができる．図 E.3 に，TRG，BTRG，HOTRG の自由エネルギーの厳密解からの相対誤差をボンド次元 χ に対してプロットしたものである (図は [103] より引用)．BTRG は最も正確な自由エネルギーを与えていることがわかる．BTRG をフェルミオン系に適用した Grassman-BTRG [68] や，impurity テンソルを含んだ BTRG [105] も考案されている．

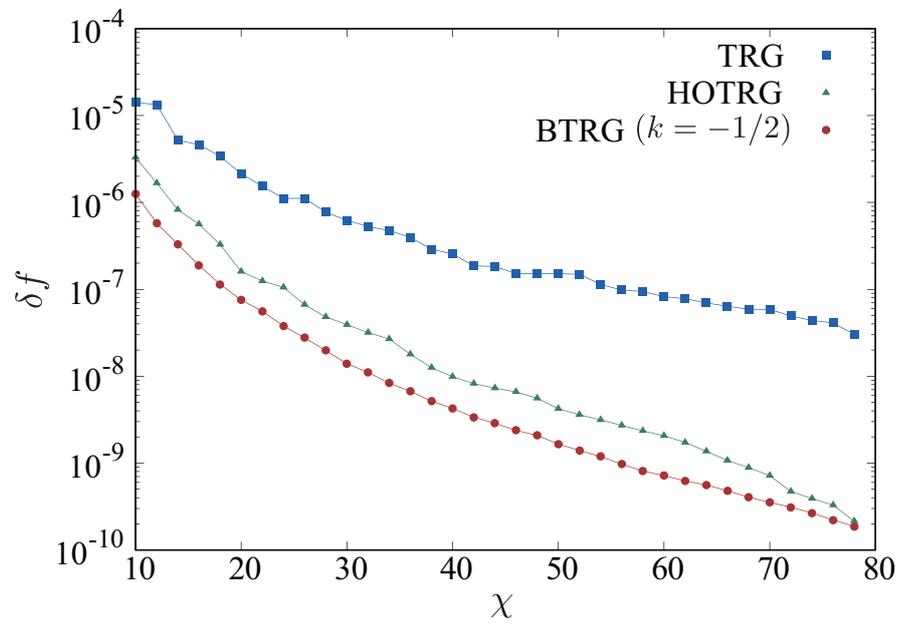


図 E.3: TRG, BTRG, HOTRG の自由エネルギーの相対誤差. 図は [103] より引用.

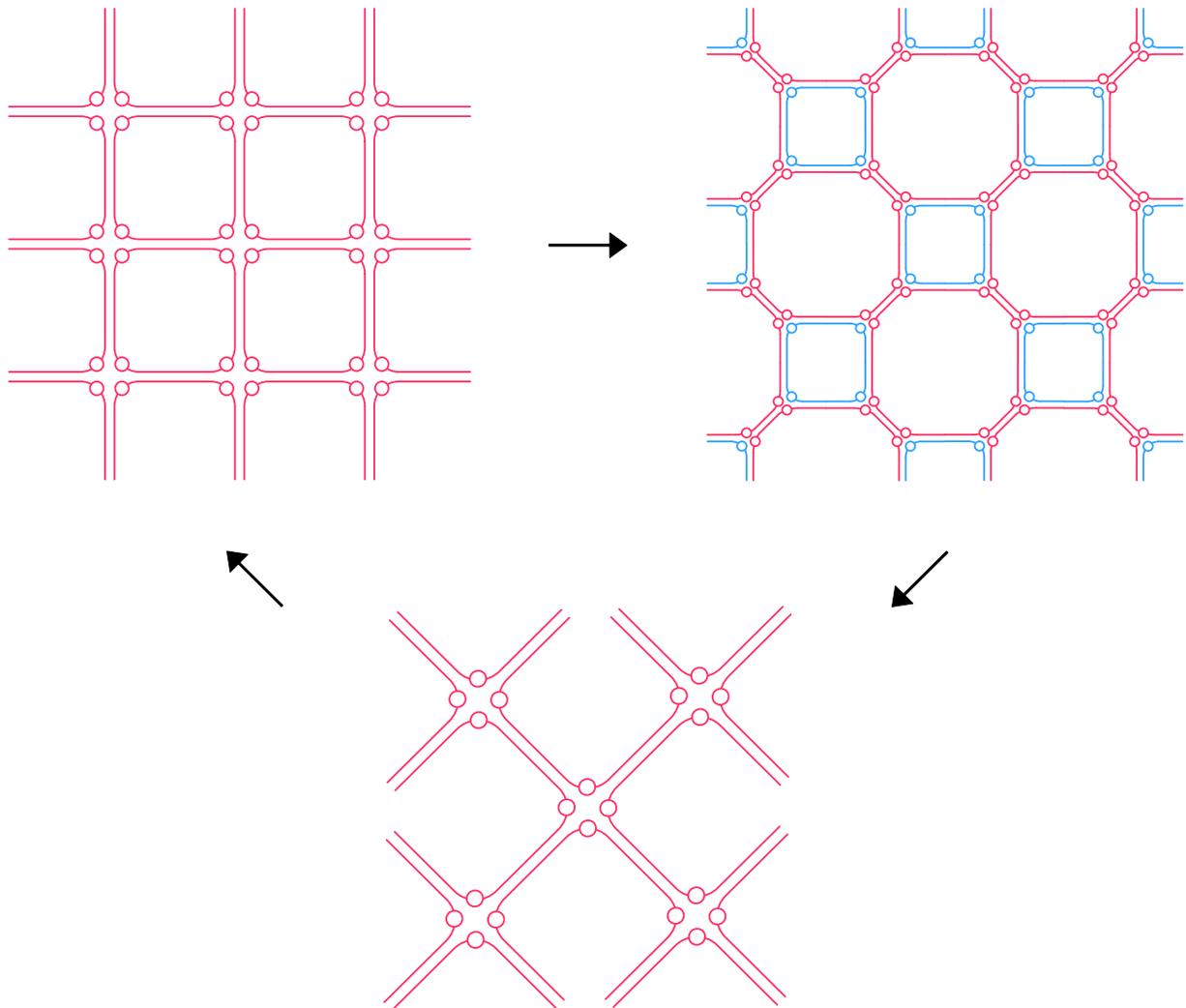


図 E.4: CDL テンソルの繰り込み. 青色のループは繰り込まれるが, 赤色のループは繰り込まれずに残る. 結果として CDL テンソルが固定テンソルとなる.

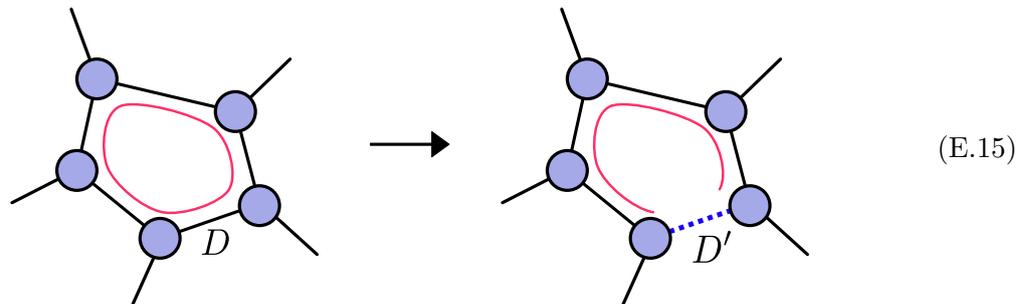
いが適切でないために, 各繰り込みステップにおいて前のステップの近距離の情報が残り続ける. 特に, エンタングルメントエントロピーが増大する臨界系においてこの傾向は顕著であり, 結果として臨界点付近での精度が著しく低下する. これは HOTRG でも同様である [106]. このような間違っただ固定点の問題に対し, Tensor Entanglement Filtering Renormalization Group (TEFR) [91] は非臨界系において CDL テンソルの次元を削減することで自明な固定点テンソルを得ることに成功した. しかしながら TEFR では, 臨界系においては非自明な固定点テンソル構造により, 精度の向上は得られなかった.

E.2.2 TNR

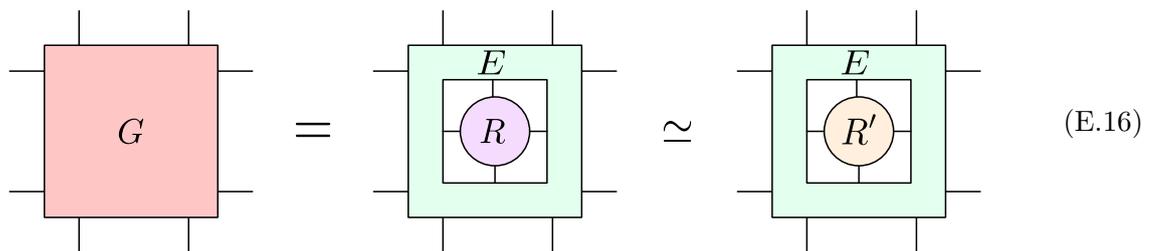
臨界系においても正確なアルゴリズムを構成したのが, Evenbly らの Tensor Network Renormalization (TNR) [107] である. TNR では, MERA [108] で導入された近距離相関を断ち切るためのユニタリ行列である (dis)entangler をネットワークに挿入し, projective truncation (一種の変分法, [109] に基づく) によって各射影テンソルを広いネットワークで最適化することで, 臨界点においてスケール普遍的な固定点テンソルを構成することに成功した. TNR では, 以下のような

E.2.3 Gilt

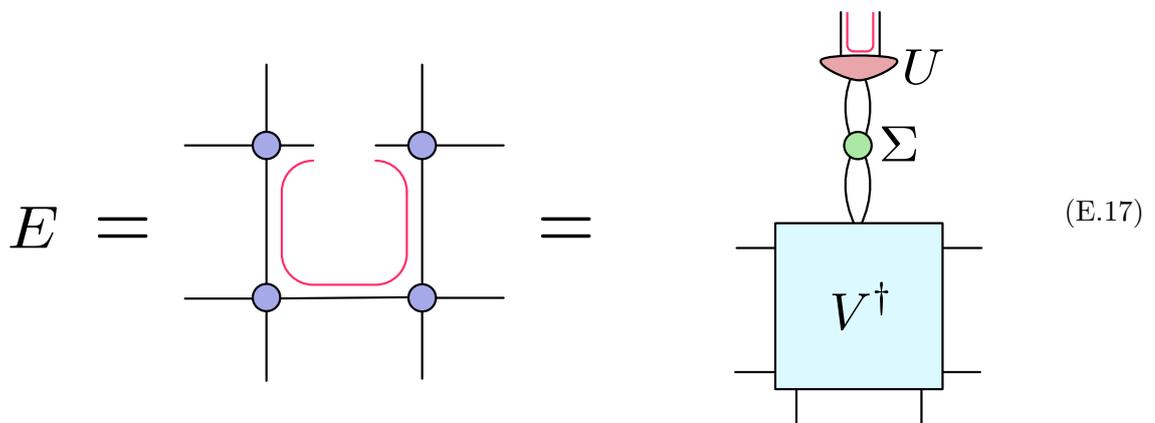
高次元の理論や、様々な形状のグラフでも変分法に頼らずに短距離相関を切るには、どうすれば良いだろうか？ Hauru らの Graph independent local truncation (Gilt) [98] は、その一般的な方法を与える。Gilt では、ある (一般にループを含むような) ネットワークのボンド (次元を D とする) に着目し、ネットワークのノルムをほとんど変えずに、短距離相関を切りながら、着目するボンドの次元を D から D' に削減することを目標としている³。グラフで表すと、以下のような置き換えである。



ではどのように元のネットワークに影響を与えずに、このような置き換えを行うことが出来るだろうか？ Hauru らは、あるネットワーク G から部分系 R を取り除いた環境テンソル E の SVD を考えた。もし取り除いた部分系を $R \rightarrow R'$ のようにした時、 $R - R'$ が E の零空間 (もしくは小さい特異値の空間) に含まれるならば、この置き換えは G に影響を与えないことが予想される。



いま、 2×2 のブラケット G に対し、着目する辺⁴を R とおき、それ以外を環境 E とする。 $E = U\Sigma V^\dagger$ と SVD を行おう。



³ただし、この削減は、短距離の余分な自由度を切り捨てることでボンド次元を削減するという意。例えば CDL ループの場合は、実質スカラー値がブラケットから現れるが、これは基本テンソルで見ればランク 1 の行列がくっついている形になる。したがって $\sqrt{\chi} \rightarrow 1$ というボンド次元の削減が可能である。ただし CDL は単純化した模型であるから、より一般には (特に臨界点などでは) ランク 1 とはならない。

⁴正確にはクロネッカーのデルタ

参考文献

- [1] R.N.C. Pfeifer, J. Haegeman and F. Verstraete, *Faster identification of optimal contraction sequences for tensor networks*, *Phys. Rev. E* **90** (2014) 033315 [[arXiv:1304.6112](#)] [[Google Scholar](#)]. [Cited on page 4.]
- [2] G. Parisi and Y.-s. Wu, *Perturbation theory without gauge fixing*, *Sci. Sin.* **24** (1981) 483 [[INSPIRE-HEP](#)]. [Cited on page 6.]
- [3] D. Sexty, *Simulating full QCD at nonzero density using the complex Langevin equation*, *Phys. Lett. B* **729** (2014) 108 [[arXiv:1307.7748](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 6.]
- [4] E. Witten, *Analytic continuation of chern-simons theory*, *AMS/IP Stud. Adv. Math.* **50** (2011) 347 [[arXiv:1001.2933](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 6.]
- [5] M. Levin and C.P. Nave, *Tensor renormalization group approach to 2D classical lattice models*, *Phys. Rev. Lett.* **99** (2007) 120601 [[arXiv:cond-mat/0611687](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 6, 7, 18, and 76.]
- [6] K.G. Wilson, *Renormalization group and critical phenomena. I. Renormalization group and the kadanoff scaling picture*, *Phys. Rev. B* **4** (1971) 3174 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 6.]
- [7] L.P. Kadanoff, *Scaling laws for ising models near T_c* , *Physics Physique Fizika* **2** (1966) 263 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 6 and 37.]
- [8] S.R. White, *Density matrix formulation for quantum renormalization groups*, *Phys. Rev. Lett.* **69** (1992) 2863 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 7.]
- [9] S.R. White, *Density-matrix algorithms for quantum renormalization groups*, *Phys. Rev. B* **48** (1993) 10345 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 7.]
- [10] S. Östlund and S. Rommer, *Thermodynamic limit of density matrix renormalization*, *Phys. Rev. Lett.* **75** (1995) 3537 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 7.]
- [11] U. Schollwöck, *The density-matrix renormalization group in the age of matrix product states*, *Annals of Physics* **326** (2011) 96 [[arXiv:1008.3477](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 7.]
- [12] Z.Y. Xie, J. Chen, M.P. Qin, J.W. Zhu, L.P. Yang and T. Xiang, *Coarse-graining renormalization by higher-order singular value decomposition*, *Phys. Rev. B* **86** (2012) 045139 [[arXiv:1201.1144](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 21, and 24.]

- [13] S. Akiyama, Y. Kuramashi, T. Yamashita and Y. Yoshimura, *Phase transition of four-dimensional Ising model with higher-order tensor renormalization group*, *Phys. Rev. D* **100** (2019) 054510 [[arXiv:1906.06060](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 37, 52, and 55.]
- [14] D. Adachi, T. Okubo and S. Todo, *Anisotropic Tensor Renormalization Group*, *Phys. Rev. B* **102** (2020) 054432 [[arXiv:1906.02007](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 25, and 29.]
- [15] S. Akiyama, Y. Kuramashi, T. Yamashita and Y. Yoshimura, *Phase transition of four-dimensional Ising model with tensor network scheme*, *PoS LATTICE2019* (2019) 138 [[arXiv:1911.12954](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 33, 37, 48, 50, and 55.]
- [16] S. Akiyama, D. Kadoh, Y. Kuramashi, T. Yamashita and Y. Yoshimura, *Tensor renormalization group approach to four-dimensional complex ϕ^4 theory at finite density*, *JHEP* **09** (2020) 177 [[arXiv:2005.04645](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 26, 33, and 43.]
- [17] S. Akiyama, Y. Kuramashi, T. Yamashita and Y. Yoshimura, *Restoration of chiral symmetry in cold and dense Nambu–Jona-Lasinio model with tensor renormalization group*, *JHEP* **01** (2021) 121 [[arXiv:2009.11583](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8 and 33.]
- [18] S. Akiyama, Y. Kuramashi and Y. Yoshimura, *Phase transition of four-dimensional lattice ϕ^4 theory with tensor renormalization group*, *Phys. Rev. D* **104** (2021) 034507 [[arXiv:2101.06953](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8, 33, 38, and 61.]
- [19] S. Akiyama and Y. Kuramashi, *Tensor renormalization group study of (3+1)-dimensional \mathbb{Z}_2 gauge-Higgs model at finite density*, *JHEP* **05** (2022) 102 [[arXiv:2202.10051](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8 and 33.]
- [20] S. Akiyama and Y. Kuramashi, *Critical endpoint of (3+1)-dimensional finite density \mathbb{Z}_3 gauge-Higgs model with tensor renormalization group*, *JHEP* **10** (2023) 077 [[arXiv:2304.07934](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 8 and 33.]
- [21] K. Nakayama, *Randomized higher-order tensor renormalization group*, July, 2019. [[arXiv:2307.14191](#)] [[INSPIRE-HEP](#)]. [Cited on pages 8, 31, 32, 33, 38, 44, and 50.]
- [22] L. Trefethen and D. Bau, *Numerical Linear Algebra*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics (1997). [Cited on page 9.]
- [23] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, *Psychometrika* **1** (1936) 211 [[Google Scholar](#)]. [Cited on page 10.]
- [24] L. De Lathauwer, B. De Moor and J. Vandewalle, *A Multilinear Singular Value Decomposition*, *SIAM J. Matrix Anal. & Appl.* **21** (2000) 1253 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 11.]

- [25] M.A.O. Vasilescu, *A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision and Machine Learning*, Ph.D. thesis, 2009. [Cited on page 12.]
- [26] L. De Lathauwer, B. De Moor and J. Vandewalle, *On the Best Rank-1 and Rank-(R_1, R_2, \dots, R_N) Approximation of Higher-Order Tensors*, *SIAM J. Matrix Anal. Appl.* **21** (2000) 1324 [Google Scholar]. [Cited on pages 13 and 52.]
- [27] T. Kolda, *Multilinear Operators for Higher-Order Decompositions.*, Ph.D. thesis, Apr., 2006. 10.2172/923081. [Cited on pages 13 and 52.]
- [28] R. Sakai, D. Kadoh, Y. Kuramashi, Y. Nakamura, S. Takeda and Y. Yoshimura, *Application of tensor network method to two dimensional lattice $\mathcal{N} = 1$ Wess-Zumino model*, *EPJ Web Conf.* **175** (2018) 11019 [arXiv:1711.01892] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 15 and 33.]
- [29] Y. Liu, Y. Meurice, M.P. Qin, J. Unmuth-Yockey, T. Xiang, Z.Y. Xie et al., *Exact blocking formulas for spin and gauge models*, *Phys. Rev. D* **88** (2013) 056005 [arXiv:1307.6543] [INSPIRE-HEP] [Google Scholar]. [Cited on page 15.]
- [30] S. Morita, R. Igarashi, H.-H. Zhao and N. Kawashima, *Tensor renormalization group with randomized singular value decomposition*, *Phys. Rev. E* **97** (2018) 033310 [arXiv:1712.01458] [Google Scholar]. [Cited on pages 21 and 29.]
- [31] Y. Nakamura, H. Oba and S. Takeda, *Tensor renormalization group algorithms with a projective truncation method*, *Phys. Rev. B* **99** (2019) 155101 [arXiv:1809.08030] [INSPIRE-HEP] [Google Scholar]. [Cited on page 21.]
- [32] Z.Y. Xie, H.C. Jiang, Q.N. Chen, Z.Y. Weng and T. Xiang, *Second Renormalization of Tensor-Network States*, *Phys. Rev. Lett.* **103** (2009) 160601 [arXiv:0809.0182] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 21 and 74.]
- [33] S. Morita and N. Kawashima, *Calculation of higher-order moments by higher-order tensor renormalization group*, *Comput. Phys. Commun.* **236** (2019) 65 [arXiv:1806.10275] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 23 and 25.]
- [34] H.-J. Liao, J.-G. Liu, L. Wang and T. Xiang, *Differentiable programming tensor networks*, *Phys. Rev. X* **9** (2019) 031041 [arXiv:1903.09650] [INSPIRE-HEP] [Google Scholar]. [Cited on page 25.]
- [35] H. Oba, *Cost reduction of the bond-swapping part in an anisotropic tensor renormalization group*, *PTEP* **2020** (2020) 013B02 [arXiv:1908.07295] [INSPIRE-HEP] [Google Scholar]. [Cited on page 25.]
- [36] S. Akiyama, *Tensor Renormalization Group Approach to Higher-Dimensional Lattice Field Theories*, Ph.D. thesis, University of Tsukuba, 2022. [https://tsukuba.repo.nii.ac.jp/records/2005504]. [Cited on page 25.]
- [37] Y.-Y. Shi, L.-M. Duan and G. Vidal, *Classical simulation of quantum many-body systems with a tree tensor network*, *Phys. Rev. A* **74** (2006) 022320 [arXiv:quant-ph/0511070] [INSPIRE-HEP] [Google Scholar]. [Cited on page 27.]

- [38] L. Wang and F. Verstraete, *Cluster update for tensor network states*, Oct., 2011. [arXiv:1110.4362](#). [Cited on page 28.]
- [39] C. Wang, S.-M. Qin and H.-J. Zhou, *Topologically invariant tensor renormalization group method for the Edwards-Anderson spin glasses model*, *Phys. Rev. B* **90** (2014) 174201 [[arXiv:1311.6577](#)] [[Google Scholar](#)]. [Cited on page 28.]
- [40] S. Iino, S. Morita and N. Kawashima, *Boundary tensor renormalization group*, *Phys. Rev. B* **100** (2019) 035449 [[arXiv:1905.02351](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 28.]
- [41] K. Nakayama and M. Schneider, *Initial tensor construction and dependence of the tensor renormalization group on initial tensors*, *Phys. Rev. D* **110** (2024) 094501 [[arXiv:2407.14226](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 28.]
- [42] J. Bloch, R. Lohmayer, M. Meister and M. Nunhofer, *Improved local truncation schemes for the higher-order tensor renormalization group method*, *Nucl. Phys. B* **987** (2023) 116107 [[arXiv:2210.02266](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 28.]
- [43] D. Kadoh and K. Nakayama, *Renormalization group on a triad network*, Dec., 2019. [arXiv:1912.02414](#) [[INSPIRE-HEP](#)]. [Cited on pages 30 and 38.]
- [44] K. Nakayama, *Application of the projective truncation and randomized singular value decomposition to a higher dimension*, *PoS LATTICE2023* (2024) 029 [[arXiv:2401.06389](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 31 and 44.]
- [45] J.F. Yu, Z.Y. Xie, Y. Meurice, Y. Liu, A. Denblyker, H. Zou et al., *Tensor renormalization group study of classical XY model on the square lattice*, *Phys. Rev. E* **89** (2014) 013308 [[arXiv:1309.4963](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [46] S. Wang, Z.Y. Xie, J. Chen, B. Normand and T. Xiang, *Phase Transitions of Ferromagnetic Potts Models on the Simple Cubic Lattice*, *Chinese Phys. Lett.* **31** (2014) 070503 [[arXiv:1405.1179](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [47] S. Hong and D.-H. Kim, *Logarithmic finite-size scaling correction to the leading Fisher zeros in the p -state clock model: A higher-order tensor renormalization group study*, *Phys. Rev. E* **101** (2020) 012124 [[arXiv:1906.09036](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [48] A. Samlodia, V. Longia, R.G. Jha and A. Joseph, *Phase diagram of generalized XY model using the tensor renormalization group*, *Phys. Rev. D* **110** (2024) 034504 [[arXiv:2404.17504](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [49] Y. Shimizu, *Analysis of the (1+1)-dimensional lattice ϕ^4 model using the tensor renormalization group*, *Chin. J. Phys.* **50** (2012) 749 [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [50] H. Kawauchi and S. Takeda, *Tensor renormalization group analysis of CP(N-1) model*, *Phys. Rev. D* **93** (2016) 114503 [[arXiv:1603.09455](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]

- [51] D. Kadoh, Y. Kuramashi, Y. Nakamura, R. Sakai, S. Takeda and Y. Yoshimura, *Tensor network analysis of critical coupling in two dimensional ϕ^4 theory*, *JHEP* **05** (2019) 184 [[arXiv:1811.12376](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [52] C. Delcamp and A. Tilloy, *Computing the renormalization group flow of two-dimensional ϕ^4 theory with tensor networks*, *Phys. Rev. Res.* **2** (2020) 033278 [[arXiv:2003.12993](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 33 and 80.]
- [53] M. Fukuma, D. Kadoh and N. Matsumoto, *Tensor network approach to two-dimensional Yang–Mills theories*, *PTEP* **2021** (2021) 123B03 [[arXiv:2107.14149](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [54] K. Nakayama, L. Funcke, K. Jansen, Y.-J. Kao and S. Kühn, *Phase structure of the $CP(1)$ model in the presence of a topological θ -term*, *Phys. Rev. D* **105** (2022) 054507 [[arXiv:2107.14220](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [55] T. Kuwahara and A. Tsuchiya, *Toward tensor renormalization group study of three-dimensional non-Abelian gauge theory*, *Progress of Theoretical and Experimental Physics* **2022** (2022) 093B02 [[arXiv:2205.08883](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [56] S. Akiyama, R.G. Jha and J. Unmuth-Yockey, *$su(2)$ principal chiral model with tensor renormalization group on a cubic lattice*, *Phys. Rev. D* **110** (2024) 034519 [[arXiv:2406.10081](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [57] Z.-C. Gu, F. Verstraete and X.-G. Wen, *Grassmann tensor network states and its renormalization for strongly correlated fermionic and bosonic states*, Apr., 2010. [[arXiv:1004.2563](#)] [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [58] Y. Shimizu and Y. Kuramashi, *Grassmann Tensor Renormalization Group Approach to One-Flavor Lattice Schwinger Model*, *Phys. Rev. D* **90** (2014) 014508 [[arXiv:1403.0642](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [59] Y. Shimizu and Y. Kuramashi, *Critical behavior of the lattice Schwinger model with a topological term at $\theta = \pi$ using the Grassmann tensor renormalization group*, *Phys. Rev. D* **90** (2014) 074503 [[arXiv:1408.0897](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [60] S. Takeda and Y. Yoshimura, *Grassmann tensor renormalization group for the one-flavor lattice Gross–Neveu model with finite chemical potential*, *PTEP* **2015** (2015) 043B01 [[arXiv:1412.7855](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [61] R. Sakai, S. Takeda and Y. Yoshimura, *Higher order tensor renormalization group for relativistic fermion systems*, *PTEP* **2017** (2017) 063B07 [[arXiv:1705.07764](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [62] Y. Yoshimura, Y. Kuramashi, Y. Nakamura, S. Takeda and R. Sakai, *Calculation of fermionic Green functions with Grassmann higher-order tensor renormalization group*, *Phys. Rev. D* **97** (2018) 054511 [[arXiv:1711.08121](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]

- [63] Y. Shimizu and Y. Kuramashi, *Berezinskii-Kosterlitz-Thouless transition in lattice Schwinger model with one flavor of Wilson fermion*, *Phys. Rev. D* **97** (2018) 034502 [[arXiv:1712.07808](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [64] S. Akiyama and D. Kadoh, *More about the Grassmann tensor renormalization group*, *JHEP* **10** (2021) 188 [[arXiv:2005.07570](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [65] M. Asaduzzaman, S. Catterall, Y. Meurice, R. Sakai and G.C. Toga, *Improved coarse-graining methods on two dimensional tensor networks including fermions*, *J. High Energ. Phys.* **2023** (2023) 24 [[arXiv:2210.03834](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [66] A. Yosprakob, J. Nishimura and K. Okunishi, *A new technique to incorporate multiple fermion flavors in tensor renormalization group method for lattice gauge theories*, Sept., 2023. [arXiv:2309.01422](#) [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [67] A. Yosprakob, *GrassmannTN: A Python package for Grassmann tensor network computations*, Oct., 2023. [arXiv:2309.07557](#) [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [68] S. Akiyama, *Bond-weighting method for the Grassmann tensor renormalization group*, *JHEP* **11** (2022) 030 [[arXiv:2208.03227](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 33 and 74.]
- [69] M. Asaduzzaman, S. Catterall, Y. Meurice, R. Sakai and G.C. Toga, *Tensor network representation of non-abelian gauge theory coupled to reduced staggered fermions*, *JHEP* **05** (2024) 195 [[arXiv:2312.16167](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [70] K.H. Pai, S. Akiyama and S. Todo, *Grassmann tensor renormalization group approach to (1 + 1)-dimensional two-color lattice QCD at finite density*, Oct., 2024. [arXiv:2410.09485](#) [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [71] H. Kanno, S. Akiyama, K. Murakami and S. Takeda, *Grassmann Tensor Renormalization Group for $n_f = 2$ massive Schwinger model with a θ term*, Dec., 2024. [arXiv:2412.08959](#) [[INSPIRE-HEP](#)]. [Cited on page 33.]
- [72] S. Akiyama, Y. Meurice and R. Sakai, *Tensor renormalization group for fermions*, *J. Phys. Condens. Matter* **36** (2024) 343002 [[arXiv:2401.08542](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 33.]
- [73] L. Onsager, *Crystal statistics. I. A two-dimensional model with an order-disorder transition*, *Phys. Rev.* **65** (1944) 117 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 36.]
- [74] C.-H. Chang, V. Dommès, R.S. Erramilli, A. Homrich, P. Kravchuk, A. Liu et al., *Bootstrapping the 3d Ising Stress Tensor*, Nov., 2024. [arXiv:2411.15300](#) [[INSPIRE-HEP](#)]. [Cited on page 37.]
- [75] M. Aizenman and R. Fernández, *On the critical behavior of the magnetization in high-dimensional Ising models*, *J Stat Phys* **44** (1986) 393 [[Google Scholar](#)]. [Cited on pages 37, 55, 57, and 60.]

- [76] A.D. Sokal, *A rigorous inequality for the specific heat of an Ising or ϕ^4 ferromagnet*, *Physics Letters A* **71** (1979) 451 [[Google Scholar](#)]. [Cited on pages 37, 57, and 60.]
- [77] H. Duminil-Copin and R. Panis, *New lower bounds for the (near) critical Ising and ϕ^4 models' two-point functions*, Apr., 2024. [arXiv:2404.05700](#). [Cited on pages 37 and 57.]
- [78] R. Kenna, *Universal scaling relations for logarithmic-correction exponents*, in *Order, Disorder and Criticality: Advanced Problems of Phase Transition Theory Volume 3*, pp. 1–46, World Scientific (2013), [DOI](#). [Cited on page 37.]
- [79] T. Hara and H. Tasaki, *A rigorous control of logarithmic corrections in four-dimensional ϕ^4 spin systems*, *Journal of Statistical Physics* **47** (1987) 99 [[Google Scholar](#)]. [Cited on page 37.]
- [80] T. Hara, *A rigorous control of logarithmic corrections in four-dimensional ϕ^4 spin systems*, *Journal of Statistical Physics* **47** (1987) 57 [[Google Scholar](#)]. [Cited on page 37.]
- [81] R. Bauerschmidt, D.C. Brydges and G. Slade, *Scaling limits and critical behaviour of the 4-dimensional n-component $|\varphi|^4$ spin model*, *Journal of Statistical Physics* **157** (2014) 692 [[arXiv:1403.7424](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [82] M. Aizenman and H. Duminil-Copin, *Marginal triviality of the scaling limits of critical 4D Ising and ϕ_4^4 models*, *Annals Math.* **194** (2021) 163 [[arXiv:1912.07973](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [83] E. Sanchez-Velasco, *A finite-size scaling study of the 4D Ising model*, *J. Phys. A: Math. Gen.* **20** (1987) 5033 [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [84] E. Bittner, W. Janke and H. Markum, *Ising spins coupled to a four-dimensional discrete Regge skeleton*, *Phys. Rev. D* **66** (2002) 024008 [[arXiv:hep-lat/0205023](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [85] P.H. Lundow and K. Markström, *Critical behavior of the Ising model on the four-dimensional cubic lattice*, *Phys. Rev. E* **80** (2009) 031104 [[arXiv:1202.3031](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [86] P.H. Lundow and K. Markstrom, *Non-vanishing boundary effects and quasi-first order phase transitions in high dimensional Ising models*, *Nucl. Phys. B* **845** (2011) 120 [[arXiv:1010.5958](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 37.]
- [87] P.H. Lundow and K. Markström, *Revising the universality class of the four-dimensional Ising model*, *Nucl. Phys. B* **993** (2023) 116256 [[arXiv:2209.05292](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 37, 50, and 61.]
- [88] Y. Sugimoto and S. Sasaki, *Applying the Triad network representation to four-dimensional ATRG method*, *PoS LATTICE2024* (2025) 038 [[arXiv:2412.14104](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 38.]
- [89] T. Yamashita and T. Sakurai, *A parallel computing method for the higher order tensor renormalization group*, *Comput. Phys. Commun.* **278** (2022) 108423 [[arXiv:2110.03607](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 43.]

- [90] R.G. Jha and A. Samlodia, *GPU-Acceleration of Tensor Renormalization with PyTorch using CUDA*, *Comput. Phys. Commun.* **294** (2024) 108941 [arXiv:2306.00358] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 43 and 48.]
- [91] Z.-C. Gu and X.-G. Wen, *Tensor-Entanglement-Filtering Renormalization Approach and Symmetry Protected Topological Order*, *Phys. Rev. B* **80** (2009) 155131 [arXiv:0903.1069] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 50, 76, and 77.]
- [92] M. Fukugita, H. Mino, M. Okawa and A. Ukawa, *Finite-size scaling of the three-state Potts model on a simple cubic lattice*, *Journal of Statistical Physics* **59** (1990) 1397 [Google Scholar]. [Cited on page 55.]
- [93] R. Kenna, D.A. Johnston and W. Janke, *Scaling relations for logarithmic corrections*, *Phys. Rev. Lett.* **96** (2006) 115701 [arXiv:cond-mat/0605162] [INSPIRE-HEP] [Google Scholar]. [Cited on page 58.]
- [94] R. Kenna, *Finite size scaling for $o(n)$ φ^4 -theory at the upper critical dimension*, *Nucl. Phys. B* **691** (2004) 292 [arXiv:hep-lat/0405023] [INSPIRE-HEP] [Google Scholar]. [Cited on page 58.]
- [95] R. Kenna, D.A. Johnston and W. Janke, *Self-consistent scaling theory for logarithmic-correction exponents*, *Phys. Rev. Lett.* **97** (2006) 155702 [arXiv:cond-mat/0608127] [INSPIRE-HEP] [Google Scholar]. [Cited on page 58.]
- [96] X. Lyu and N. Kawashima, *Essential difference between 2D and 3D from the perspective of real-space renormalization group*, Nov., 2023. arXiv:2311.05891 [INSPIRE-HEP]. [Cited on page 61.]
- [97] X. Lyu and N. Kawashima, *Three-dimensional real space renormalization group with well-controlled approximations*, Dec., 2024. arXiv:2412.13758 [INSPIRE-HEP]. [Cited on page 61.]
- [98] M. Hauru, C. Delcamp and S. Mizera, *Renormalization of tensor networks using graph independent local truncations*, *Phys. Rev. B* **97** (2018) 045111 [arXiv:1709.07460] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 61, 79, and 80.]
- [99] [<https://gregorygundersen.com/blog/2018/12/20/svd-proof/>]. [Cited on page 65.]
- [100] J. Von Neumann, *Some Matrix-Inequalities and Metrization of Matric Space* (1937). [Cited on page 66.]
- [101] [<http://aries.phys.cst.nihon-u.ac.jp/~ohya/stat-mech/main.pdf>]. [Cited on page 67.]
- [102] N. Halko, P.G. Martinsson and J.A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, *SIAM Review* **53** (2011) 217 [arXiv:0909.4061] [Google Scholar]. [Cited on page 72.]
- [103] D. Adachi, T. Okubo and S. Todo, *Bond-weighted Tensor Renormalization Group*, *Phys. Rev. B* **105** (2022) L060402 [arXiv:2011.01679] [INSPIRE-HEP] [Google Scholar]. [Cited on pages 73, 74, and 75.]

- [104] H.H. Zhao, Z.Y. Xie, Q.N. Chen, Z.C. Wei, J.W. Cai and T. Xiang, *Renormalization of tensor-network states*, *Phys. Rev. B* **81** (2010) 174411 [[arXiv:1002.1405](#)] [[Google Scholar](#)]. [Cited on page 74.]
- [105] S. Morita and N. Kawashima, *Multi-impurity method for the bond-weighted tensor renormalization group*, Nov., 2024. [arXiv:2411.13998](#) [[INSPIRE-HEP](#)]. [Cited on page 74.]
- [106] H. Ueda, K. Okunishi and T. Nishino, *Doubling of Entanglement Spectrum in Tensor Renormalization Group*, *Phys. Rev. B* **89** (2014) 075116 [[arXiv:1306.6829](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 77.]
- [107] G. Evenbly and G. Vidal, *Tensor network renormalization*, *Phys. Rev. Lett.* **115** (2015) 180405 [[arXiv:1412.0732](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 77.]
- [108] G. Vidal, *Class of quantum many-body states that can be efficiently simulated*, *Phys. Rev. Lett.* **101** (2008) 110501 [[arXiv:quant-ph/0610099](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 77.]
- [109] G. Evenbly and G. Vidal, *Algorithms for entanglement renormalization*, *Phys. Rev. B* **79** (2009) 144108 [[arXiv:0707.1454](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 77.]
- [110] G. Evenbly, *Algorithms for tensor network renormalization*, *Phys. Rev. B* **95** (2017) 045117 [[arXiv:1509.07484](#)] [[Google Scholar](#)]. [Cited on page 78.]
- [111] S. Yang, Z.-C. Gu and X.-G. Wen, *Loop optimization for tensor network renormalization*, *Phys. Rev. Lett.* **118** (2017) 110504 [[arXiv:1512.04938](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 78.]
- [112] M. Bal, M. Mariën, J. Haegeman and F. Verstraete, *Renormalization group flows of Hamiltonians using tensor networks*, *Phys. Rev. Lett.* **118** (2017) 250602 [[arXiv:1703.00365](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 78.]
- [113] K. Homma, T. Okubo and N. Kawashima, *Nuclear norm regularized loop optimization for tensor network*, *Physical Review Research* **6** (2024) 043102 [[arXiv:2306.17479](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on pages 78 and 80.]
- [114] K. Harada, *Entanglement branching operator*, *Phys. Rev. B* **97** (2018) 045124 [[arXiv:1710.01830](#)] [[Google Scholar](#)]. [Cited on page 78.]
- [115] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, *Commun. ACM* **55** (2012) 111 [[arXiv:0805.4471](#)] [[Google Scholar](#)]. [Cited on page 80.]
- [116] N. Ebel, T. Kennedy and S. Rychkov, *Rotations, negative eigenvalues, and newton method in tensor network renormalization group*, [[arXiv:2408.10312](#)] [[INSPIRE-HEP](#)] [[Google Scholar](#)]. [Cited on page 80.]
- [117] L. Ying, *Tensor network skeletonization*, *Multiscale Modeling & Simulation* **15** (2017) 1423 [[arXiv:1607.00050](#)] [[Google Scholar](#)]. [Cited on page 80.]
- [118] G. Evenbly, *Gauge fixing, canonical forms and optimal truncations in tensor networks with closed loops*, *Phys. Rev. B* **98** (2018) 085155 [[arXiv:1801.05390](#)] [[Google Scholar](#)]. [Cited on page 80.]