O(3) シグマモデルにおけるテンソル繰り込み群

M2 井元 航希

2020/10/9

Introduction

テンソル繰り込み群 (TRG) とは

符号問題 (複素作用問題)

作用が複素数になると直接モンテカルロ法で計算することができない

$\Downarrow$

Levin & Nave 2007

テンソル繰り込み群 (Tensor Renormalization Group, TRG)

Z. Y. Xie, et al. 2012

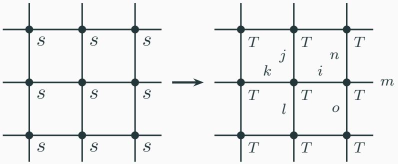
高次テンソル繰り込み群 (Higher Order TRG, HOTRG)

- 格子モデルの分配関数を粗視化を用いて計算する決定論的アルゴリズム
- 符号問題はクリア
- 本研究では O(3) シグマモデルを用いて 2 点相関関数の計算を試みる

TRG の流れ

1. 分配関数をテンソルネットワークとして書き換える

$$
\begin{aligned}
Z &= \int \left(\prod_i \frac{\mathrm{d}\Omega_i}{4\pi} \right) e^{-S} \\
&\approx \sum_{i,j,k,l,\ldots} \cdots T_{ijkl} T_{mnio} \cdots \\
&\equiv \mathrm{Tr} \left[\prod_p T(p) \right]
\end{aligned}
$$



2. テンソルネットワークを粗視化する

$$TT \cdots T \approx \cdots \approx T^*$$

3. 有効テンソル T^* を縮約することで Z が計算できる

$$Z \approx \sum_{i,j} T^*_{ijij}$$

2 次元 O(3) シグマモデルのテンソルネットワーク表現

特異値分解 (Singular Value Decomposition, SVD) は行列の低ランク近似として最良の方法であることが知られている

$$
M_{ab} = \sum_{c=1}^{\min(m,n)} U_{ac}\sigma_c V^\dagger_{cb}, \quad
\begin{cases}
M : m \times n \text{ 行列} \\
U : m \times m \text{ ユニタリ行列} \\
V : n \times n \text{ ユニタリ行列} \\
\sigma : MM^\dagger \text{の正の固有値の平方根 } (\sigma_i \geq 0)
\end{cases}
$$

Gauss-Legendre quadrature は定積分を比較的少ない演算で精度良く求めることができるアルゴリズムである．積分可能な任意の関数 $f(x)$ に対して以下のように近似する．

$$\int_{-1}^{1} \mathrm{d}x f(x) \approx \sum_{i=1}^{n} w_i f(x_i), \quad \begin{cases} x_i : P_n(x_i) = 0 & \cdots \text{ Gauss node} \\ w_i : 2/[(1-x_i^2)\{P_n'(x_i)\}^2] & \cdots \text{ weight} \end{cases}$$

$P_n(x)$ は n 次の Legendre 多項式で，一般に node 数 n を大きく取るほど精度が向上する．

2 次元 O(3) シグマモデルのテンソルネットワーク表現

なぜ O(3) 非線形シグマモデル？

$\rightarrow$ 漸近的自由性が見られ，4 次元非可換ゲージ理論と類似している

ハミルトニアン

$$H = -\sum_{\langle i,j \rangle} \boldsymbol{S}_i \cdot \boldsymbol{S}_j, \quad \boldsymbol{S} = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}$$

分配関数

$$Z = \int \left(\prod_p \frac{\mathrm{d}\Omega_p}{4\pi} \right) e^{-\beta H} = \left(\prod_p \frac{1}{4\pi} \int_0^{\pi} \mathrm{d}\theta_p \int_0^{2\pi} \mathrm{d}\varphi_p \sin\theta_p \right) \prod_{\langle i,j \rangle} e^{\beta \boldsymbol{S}(\theta_i,\varphi_i)\cdot \boldsymbol{S}(\theta_j,\varphi_j)}$$

2 次元 O(3) シグマモデルのテンソルネットワーク表現

まずは分配関数の積分を (θ, φ の積分区間を [-1,1] に直してから) Gauss-Legendre quadrature で離散化する

$$Z \approx \sum_{\{a,b\}}^{K} \prod_{p} \frac{\pi^2}{2} w_{a_p} w_{b_p} \cos(\pi x_p^{(a_p)}/2) \left(\prod_{\mu=1}^{2} f_{k_p(a_p,b_p),k_{p+\hat{\mu}}(a_{p+\hat{\mu}},b_{p+\hat{\mu}})} \right)$$

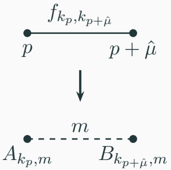
$f_{k_p(a_p,b_p),k_{p+\hat{\mu}}(a_{p+\hat{\mu}},b_{p+\hat{\mu}})}$ $\ldots$ $e^{\beta \boldsymbol{S}(\theta_i,\varphi_i)\cdot \boldsymbol{S}(\theta_j,\varphi_j)}$を離散化した部分

$f_{k_p,k_{p+\hat{\mu}}}$ の部分が $K^2 \times K^2$ の行列の形になっているので SVD

この時 m の和も K^2 になるが，ボンド次元 D_{cut} で打ち切る

$$\begin{aligned} f_{k_p,k_{p+\hat{\mu}}} &\approx \sum_{m=1}^{D_{\mathrm{cut}}} U_{k_p,m} \sigma_m V^{\dagger}_{m,k_{p+\hat{\mu}}} \\ &= \sum_{m=1}^{D_{\mathrm{cut}}} A_{k_p,m} B_{k_{p+\hat{\mu}},m} \end{aligned}$$

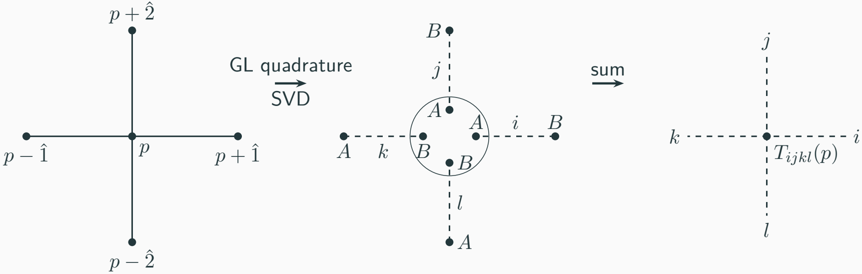
$$\begin{cases} A_{k_p,m} = U_{k_p,m}\sqrt{\sigma_m} \\ B_{k_{p+\hat{\mu}},m} = V^{\dagger}_{m,k_{p+\hat{\mu}}}\sqrt{\sigma_m} \end{cases}$$



2 次元 O(3) シグマモデルのテンソルネットワーク表現

最後に各格子点上で a, b について和を取ることでテンソルが完成

$$T_{ijkl}(p) = \sum_{a,b=1}^{K} w_a w_b \cos(\pi x^{(a)}/2) A_{(a,b)i} A_{(a,b)j} B_{(a,b)k} B_{(a,b)l}$$

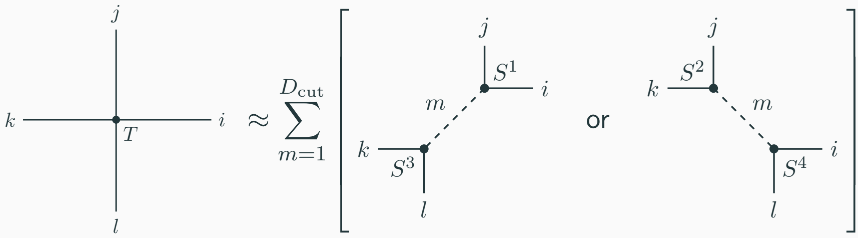


アルゴリズムの説明

TRG の縮約方法

次元 D_{cut} の 4 階テンソルを $D_{\mathrm{cut}}^2 \times D_{\mathrm{cut}}^2$ の行列とみなして SVD

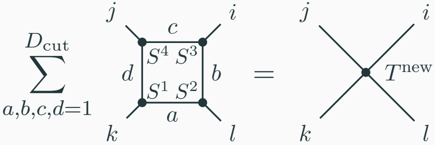
$$
T_{ijkl} = \begin{cases} M_{(ij)(kl)} \approx \displaystyle\sum_{m=1}^{D_{\mathrm{cut}}} U_{(ij)m} \sigma_m V^{\dagger}_{m(kl)} = \displaystyle\sum_{m=1}^{D_{\mathrm{cut}}} S^1_{(ij)m} S^3_{m(kl)} \\ M_{(jk)(li)} \approx \displaystyle\sum_{m=1}^{D_{\mathrm{cut}}} U_{(jk)m} \sigma_m V^{\dagger}_{m(li)} = \displaystyle\sum_{m=1}^{D_{\mathrm{cut}}} S^2_{(jk)m} S^4_{m(li)} \end{cases}
$$

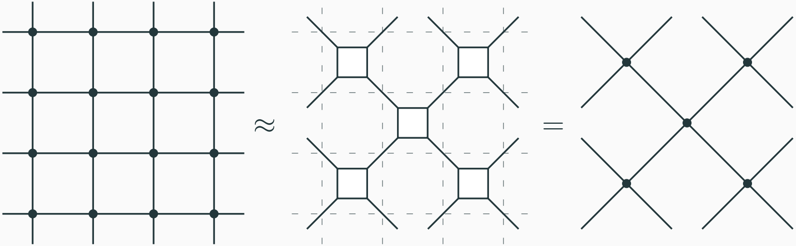


TRG の縮約方法

SVD で求めたテンソル S^1, S^2, S^3, S^4 を縮約して新しいテンソルを得る

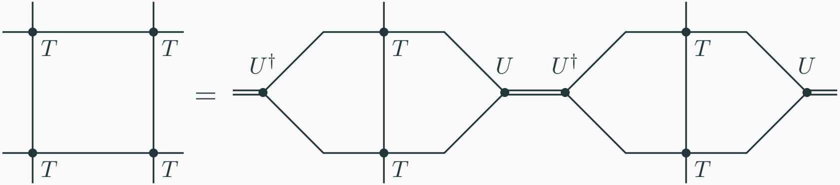
$$T^{\text{new}}_{ijkl} = \sum_{a,b,c,d=1}^{D_{\text{cut}}} S^1_{(ad)k} S^2_{(ba)l} S^3_{i(cb)} S^4_{j(dc)}$$



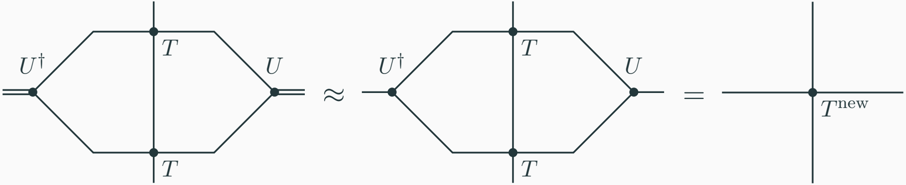


HOTRGの縮約方法

ユニタリ行列をテンソルネットワークに挿入する



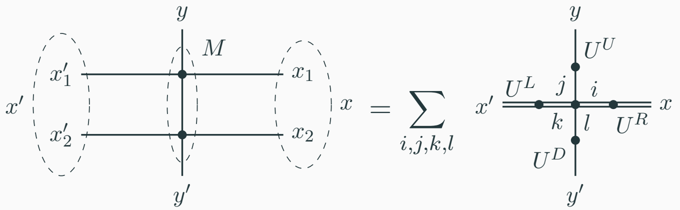
D_{cut} で U のボンド次元を切断し，2 つの T と $U, U^\dagger$ を縮約することで新しいテンソルを得る



HOTRG の縮約方法

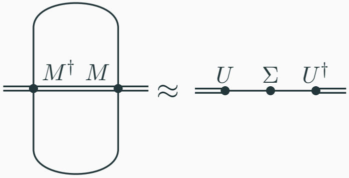
ユニタリ行列 U は高次特異値分解 (HOSVD) を利用して得る

$$
\begin{aligned}
&M_{xx'yy'}\\
&\equiv \sum_a T_{x_1 y x_1' a} T_{x_2 a x_2' y'}\\
&= \sum_{i,j,k,l} S_{ijkl} U^R_{xi} U^U_{yj} U^L_{x'k} U^D_{y'l}
\end{aligned}
$$



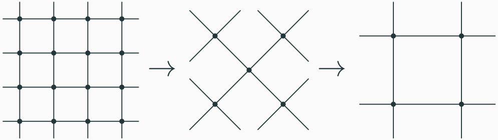
実際には U^R, U^L のみを計算して誤差が少ない方を採用する．計算方法は $MM^\dagger$ を作って SVD するだけである

$$
(MM^\dagger)_{ab} = \begin{cases} \displaystyle\sum_{x'yy'} M_{ax'yy'} M_{bx'yy'} \approx \sum_{m=1}^{D_{\mathrm{cut}}} U^R_{am} \sigma_m (U^R)^\dagger_{mb} \\ \displaystyle\sum_{xyy'} M_{xayy'} M_{xbyy'} \approx \sum_{m=1}^{D_{\mathrm{cut}}} U^L_{am} \sigma_m (U^L)^\dagger_{mb} \end{cases}
$$

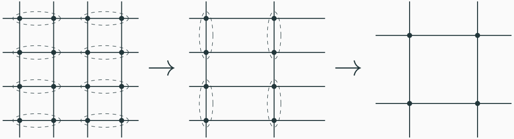


どちらも 1 回の繰り込みで体積が半分になるので N 回で $V = 2^N$ の体積が計算でき，計算を工夫すれば必要なメモリサイズを $O(D_{\mathrm{cut}}^4)$ に抑えることができる

- TRG
    - 計算時間 $O(D_{\mathrm{cut}}^6)$
    - 高次元に拡張するのが難しい

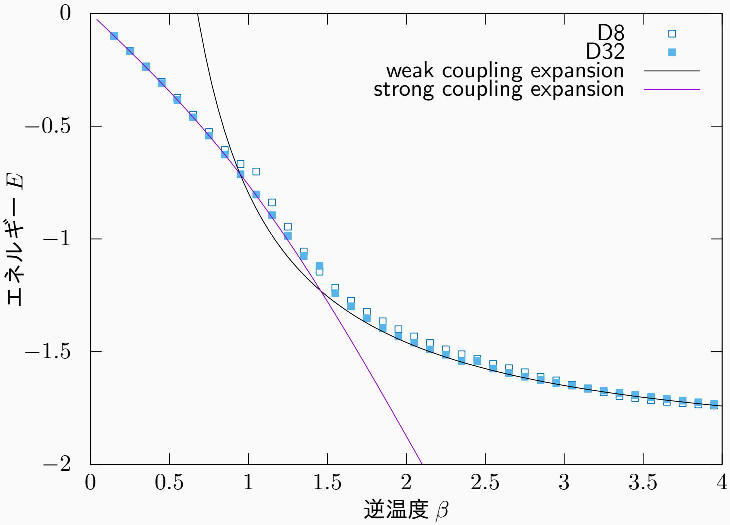


- HOTRG
    - 計算時間 $O(D_{\mathrm{cut}}^7)$
    - 比較的容易に高次元に拡張可能



計算結果

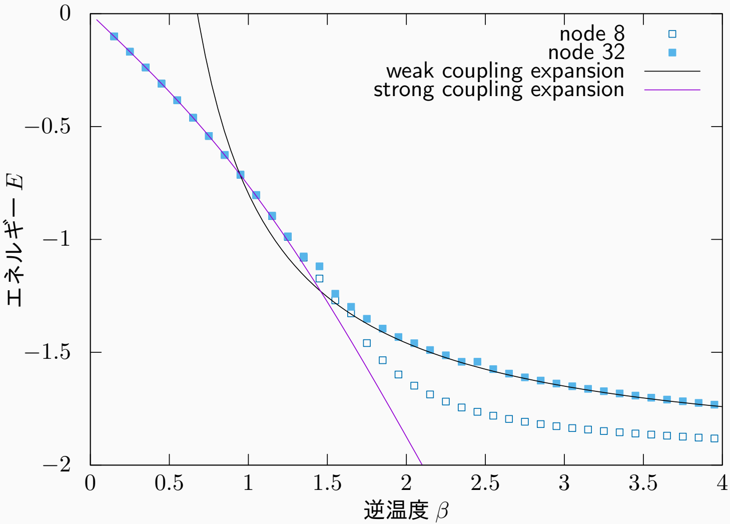
エネルギーの計算



vs D_{cut}

- TRG
- $V = 2^{10} \times 2^{10}$
- $K = 32$

エネルギーの計算



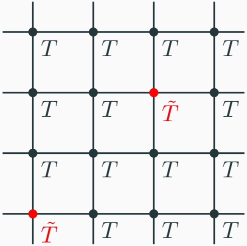
vs K

- TRG
- $V = 2^{10} \times 2^{10}$
- $D_{\mathrm{cut}} = 32$

2 点関数の計算

2 点関数はテンソルネットワークで書くと次のように表せる

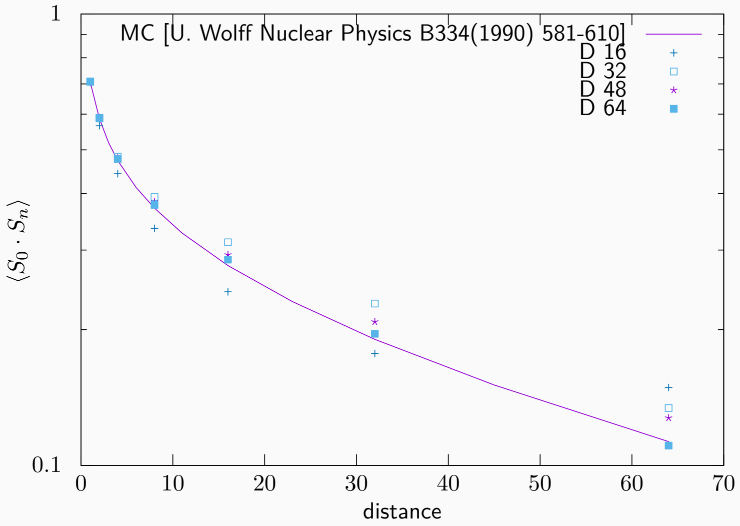
$$\langle \boldsymbol{S}_0 \cdot \boldsymbol{S}_n \rangle = \frac{1}{Z_0} \sum_{\mu=1}^{3} \mathrm{Tr} \left[\tilde{T}^{\mu}(0) \tilde{T}^{\mu}(n) \prod_{p \neq 0,n} T(p) \right]$$



$\tilde{T}$ は impure tensor と呼ばれるテンソルで，pure tensor T を作る際に spin S を追加で差し込んだ形をしている

$$\tilde{T}^{\mu}_{ijkl}(p) = \sum_{a,b=1}^{K} w_a w_b S^{\mu}(x^{(a)}, x^{(b)}) \cos(\pi x^{(a)}/2) A_{(a,b)i} A_{(a,b)j} B_{(a,b)k} B_{(a,b)l}$$

2 点関数の計算



vs D_{cut}

- HOTRG
- $V = 2^{20} \times 2^{20}$
- $K = 32$
- $\beta = 1.9$

まとめ & 今後の課題

- 2 次元 O(3) シグマモデルにおいて 2 点相関関数を HOTRG で計算することに成功した
- D_{cut} を上げることで精度が向上することが実際に数値として見ることができた
- zero-memontum projection した相関関数

$$C(t) = \sum_{x,y} \langle \boldsymbol{S}(x,t) \cdot \boldsymbol{S}(0,0) \rangle$$

  を計算することで mass を求めたい.
- 2 点関数の相関長からも mass が求められるのでそちらも試す.