テンソルくりこみ群

坂井涼

金沢大自然

2018/8/22

第 1 回 HPC-Phys 勉強会

京都大学基礎物理学研究所湯川記念館
パナソニック国際交流ホール

背景: 複素位相問題 (符号問題)

- 物理量 $\mathcal{O}$ の期待値の表式

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \det D\left[U\right] \mathcal{O}\left[U\right] e^{-S_{\mathrm{G}}[U]},$$

  - U，S_{G}: ゲージ場，ゲージ作用
  - $\det D\left[U\right]$: フェルミオン行列式
  - Z: 分配関数

  の中で，Boltzmann 因子は必ずしも実・正定符号ではない;

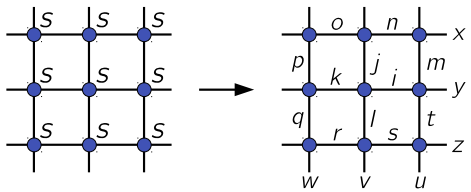
$$\frac{1}{Z} \det D\left[U\right] e^{-S_{\mathrm{G}}[U]} \in \mathbb{C}.$$

- 超対称性模型，カイラルゲージ理論，有限密度系等において確率的なシミュレーションは破綻する
- 決定論的な手法を模索する強い動機
  $\rightarrow$ テンソルネットワーク法

テンソルくりこみ群 (TRG) [M. Levin and C. P. Nave, Phys. Rev. Lett. 99 (2007)]

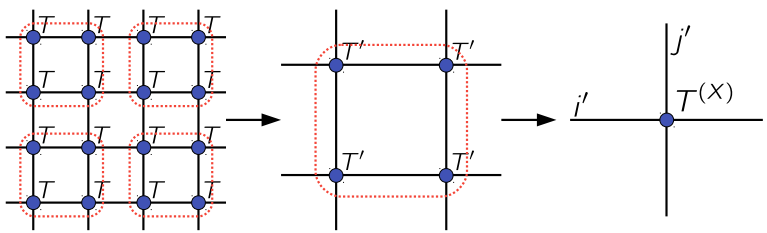
1 Z をテンソルネットワークとして表す

$$Z = \sum_{\{s\}} e^{-\beta H[s]} \to \sum_{\ldots,i,j,k,l,\ldots} \cdots T_{ijkl} T_{ymit} T_{xunm} \cdots$$



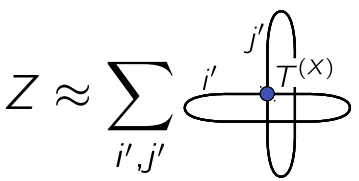
2 テンソルネットワークを粗視化する

$$TT \cdots T \underbrace{\to \cdots \overset{\text{粗視化}}{\to} \cdots \to}_{X \text{ 回}} T^{(X)}$$



3 「有効テンソル」$T^{(X)}$ を縮約して Z を得る

$$Z \approx \sum_{i',j'} T^{(X)}_{i'j'i'j'}$$



- ✓ 決定論的な情報圧縮
- ✓ 符号問題と完全に無縁
- ✓ 系統誤差は 1 パラメータ ($T^{(X)}$ のサイズ) で制御できる.

分配関数をテンソルネットワークとして表す

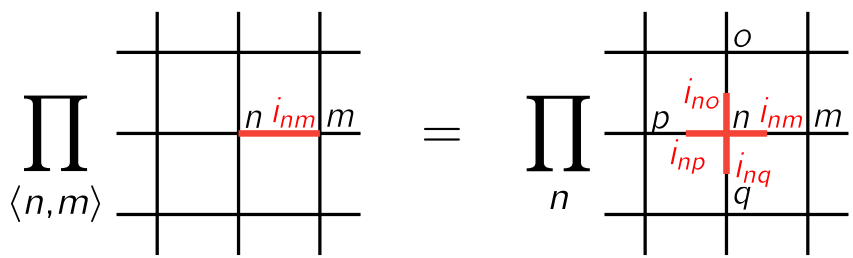
2 次元 Ising 模型の分配関数:

$$Z = \sum_{\{s\}} e^{-\beta H[s]} = \sum_{\{s\}} \prod_{\langle n,m \rangle} e^{\beta s_n s_m},$$

$$\text{ただし，} \quad H[s] = -\sum_{\langle n,m \rangle} s_n s_m$$

Z のテンソルネットワーク表現の作り方:

1. Boltzmann 因子を整数自由度で展開する
   - (例) 高温展開: $e^{\beta s_n s_m} = \cosh\beta \sum_{i_{nm}=0}^{1} (s_n s_m \tanh\beta)^{i_{nm}}$
2. $\prod_{\langle n,m \rangle}$ を $\prod_n$ に読み替え，古い自由度 s の和をとりきる



3. 残った整数自由度をテンソルの添字とみなす

Z のテンソルネットワーク表現

$$Z = \sum_{\{s\}} \prod_{\langle n,m \rangle} e^{\beta s_n s_m} = \sum_{\{i\}} \prod_{n} T_{i_{nm} i_{no} i_{np} i_{nq}},$$

ただし，$T_{i_{nm} i_{no} i_{np} i_{nq}} = \cosh\beta \left(\sqrt{\tanh\beta}\right)^{i_{nm}+i_{no}+i_{np}+i_{nq}} \cdot 2\delta_{(i_{nm}+i_{no}+i_{np}+i_{nq}) \bmod 2,0}$

- Z はスピン s の Tr → テンソル添字 i の Tr として計算できる!
- 他の多くの模型もテンソルネットワークとして表現できる
- しかし，これは自由度の読み替えに過ぎず，$\sum_{\{i\}}$ は簡単には実行できない ($\mathcal{O}\left(2^{2V}\right)$ の計算！)
  → テンソルネットワークを「粗視化」する

テンソルネットワークの粗視化

Eckart–Young の定理 [C. Eckart and G. Young, Psychometrika 1 (1936)] によると，
特異値分解 (SVD) は行列の低ランク近似として最良 [1] の方法.

$$T_{ijkl} = M_{(jk)(li)} = \sum_{m=1}^{D^2} U_{(jk)m} \sigma_m V^\dagger_{m(li)}$$

$$\approx \sum_{m=1}^{D_{\mathrm{cut}}} U_{(jk)m} \sigma_m V^\dagger_{m(li)} \qquad \leftarrow \text{一番良い!}$$

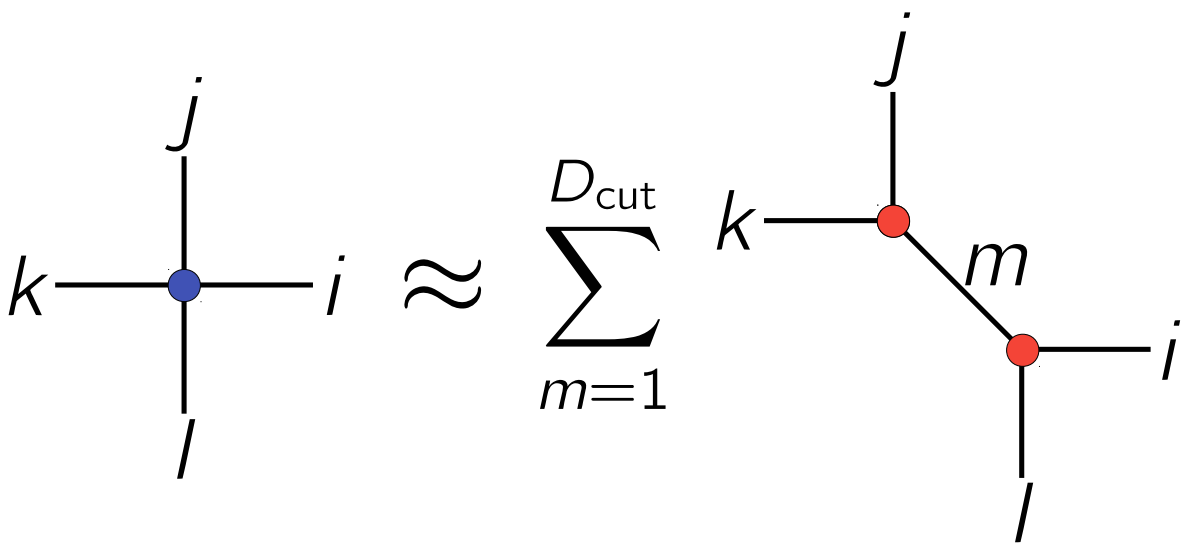
- $1 \le i, j, k, l \le D$
- $D_{\mathrm{cut}} \le D^2$
- $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_{D^2} \ge 0$

[1] 近似前/後の行列の差の Frobenius ノルムが最小になることを「最良」とする.

テンソルネットワークの粗視化

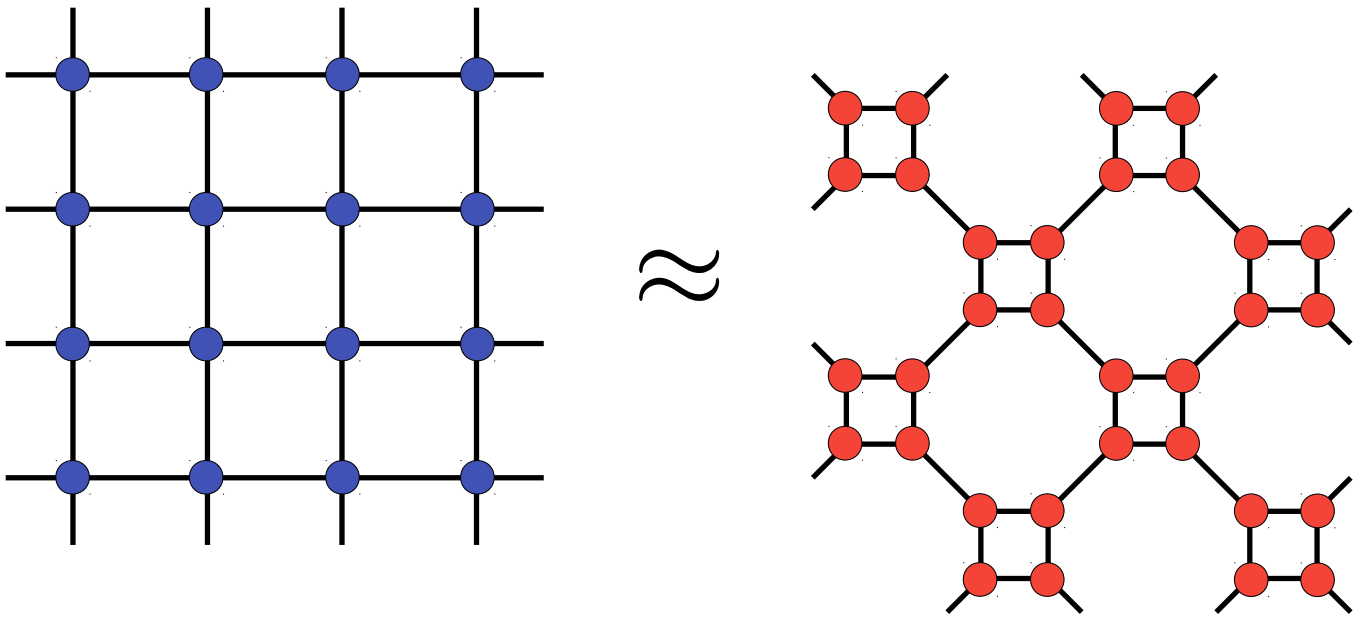
Eckart–Young の定理 [C. Eckart and G. Young, Psychometrika 1 (1936)] によると，
特異値分解 (SVD) は行列の低ランク近似として最良の方法.

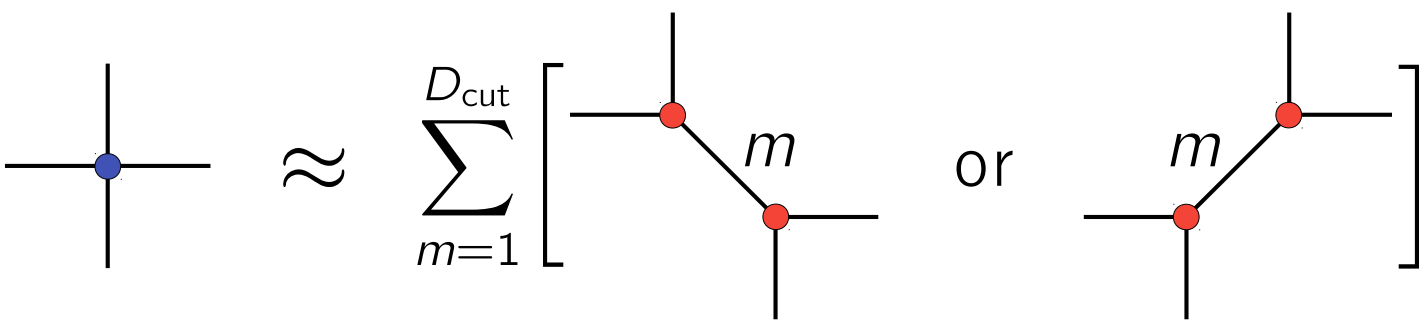
$$T_{ijkl} = M_{(jk)(li)} \approx \sum_{m=1}^{D_{\mathrm{cut}}} U_{(jk)m} \sigma_m V^{\dagger}_{m(li)}$$



テンソルネットワークの粗視化

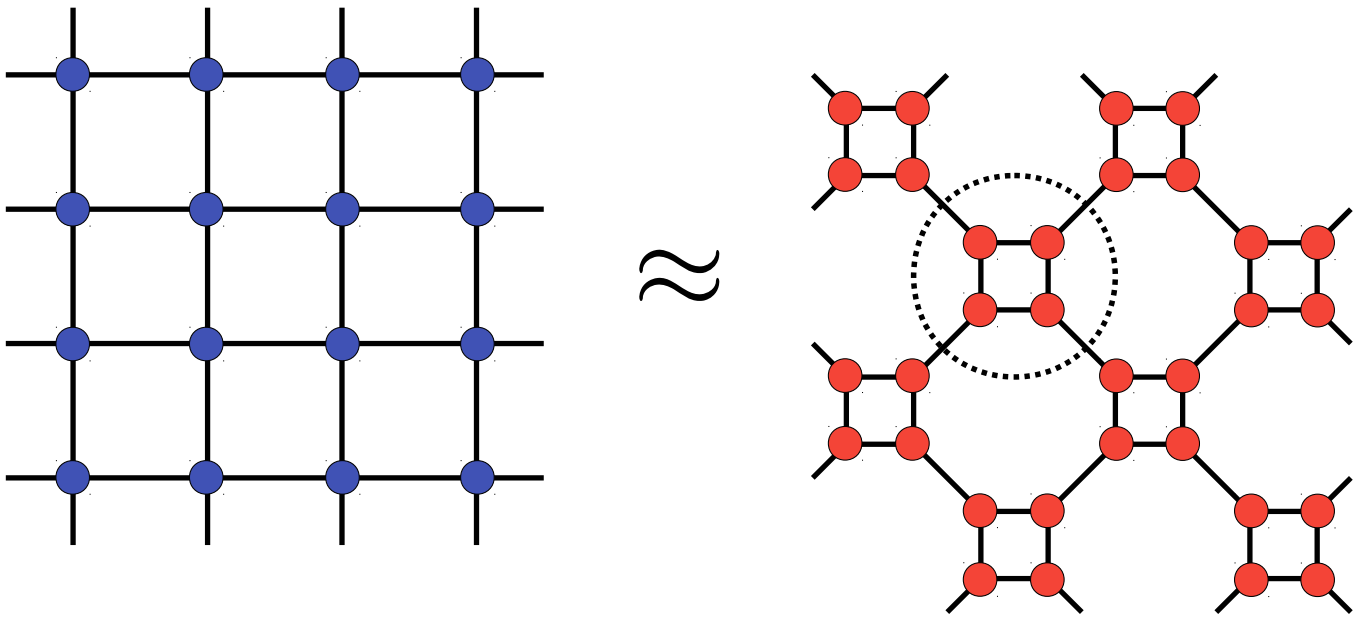
Eckart–Young の定理 [C. Eckart and G. Young, Psychometrika 1 (1936)] によると，
特異値分解 (SVD) は行列の低ランク近似として最良の方法.



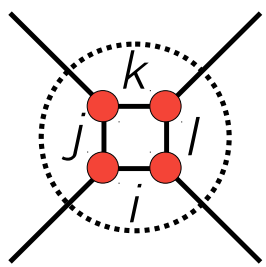


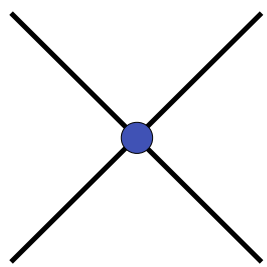
テンソルネットワークの粗視化

Eckart–Young の定理 [C. Eckart and G. Young, Psychometrika 1 (1936)] によると，
特異値分解 (SVD) は行列の低ランク近似として最良の方法.



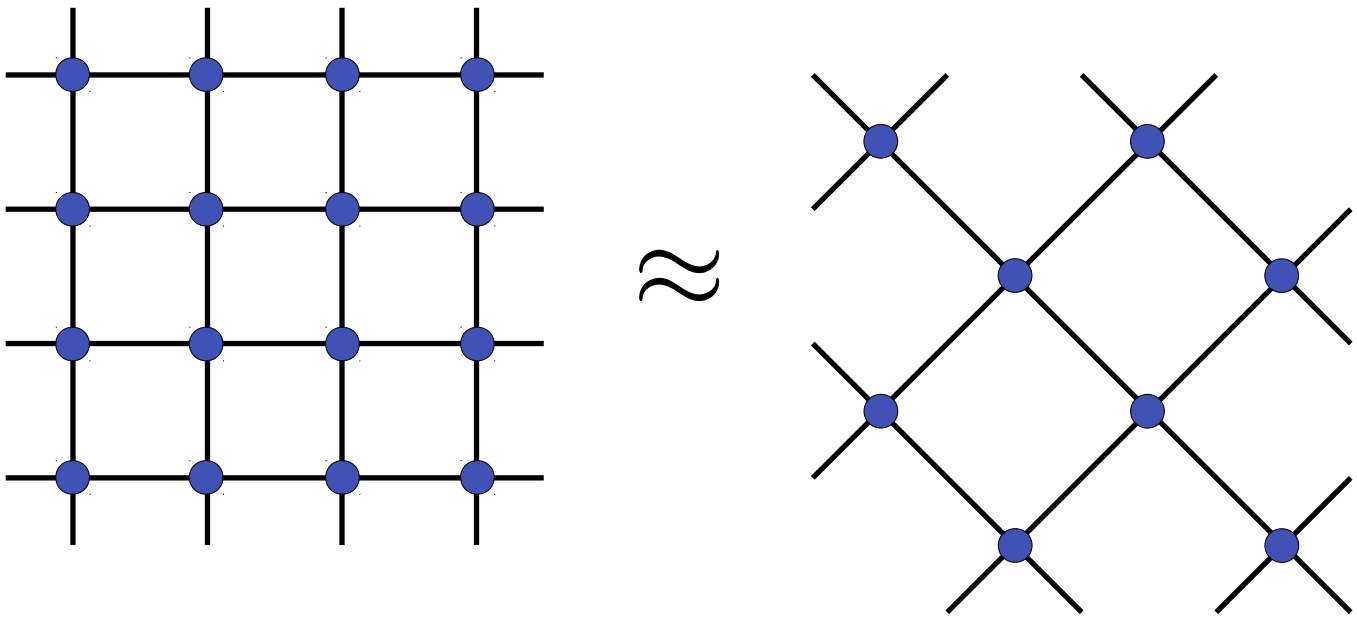
$$\sum_{i,j,k,l=1}^{D}$$



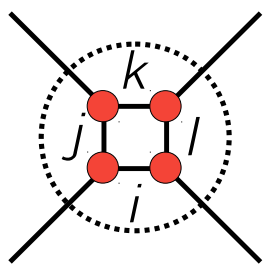


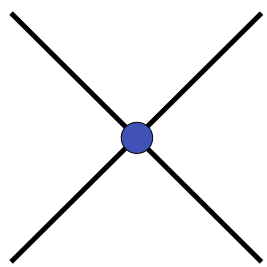
テンソルネットワークの粗視化

Eckart–Young の定理 [C. Eckart and G. Young, Psychometrika 1 (1936)] によると，
特異値分解 (SVD) は行列の低ランク近似として最良の方法.

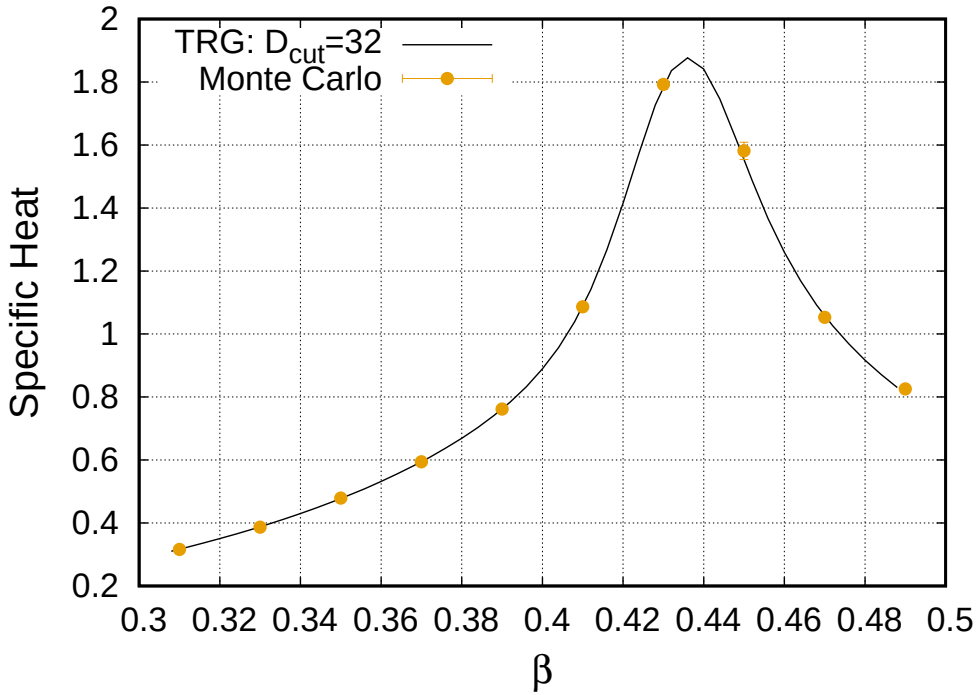


$$\sum_{i,j,k,l=1}^{D}$$





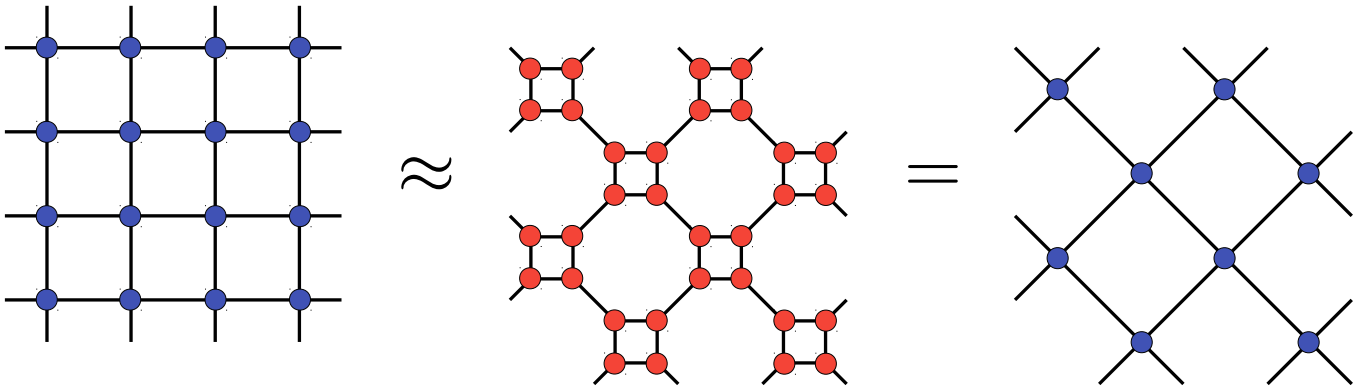
2 次元 Ising 模型の比熱，TRG vs. MCMC



$$C = \frac{\beta^2}{V} \frac{\partial^2 \ln Z}{\partial \beta^2}.$$

- $V = 32 \times 32$.
- β: 逆温度
- D_{cut}: テンソルのボンド次元 (サイズ)，実験的に決める

テンソルくりこみ群の「実装」



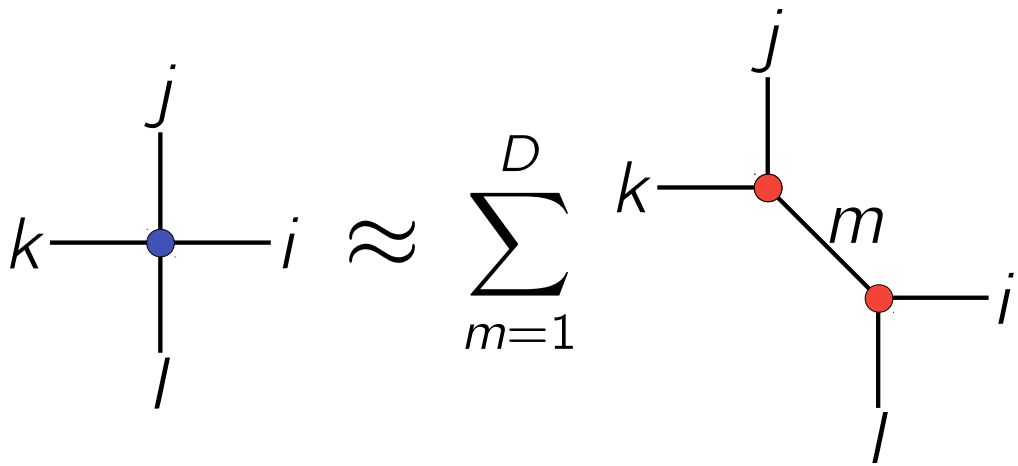
- テンソルくりこみ群のアルゴリズムはテンソルの分解とかけ算から成る
- 「テンソルの」と言いつつ，実際に行なっているのは「行列と見たテンソルの」分解とかけ算である

$\Rightarrow$ BLAS や LAPACK といったライブラリのルーチンを使って実装できる

$*$ 以降では，必要がない限り簡単のために $D_{\mathrm{cut}} = D$ とする

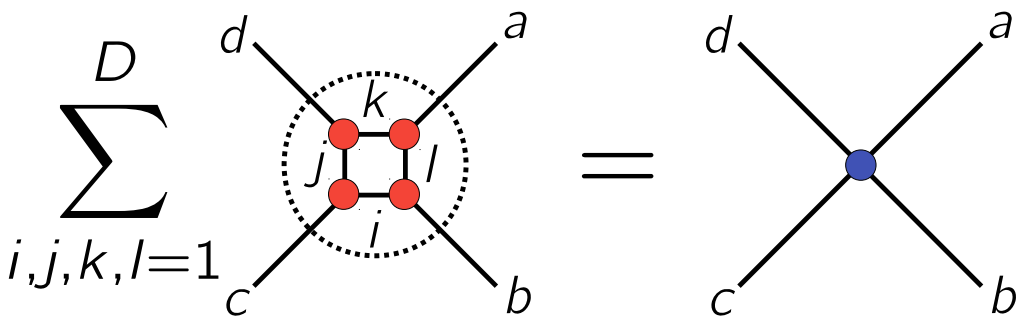
テンソルの特異値分解

$$T_{ijkl} = M_{(jk)(li)} \approx \sum_{m=1}^{D} U_{(jk)m} \sigma_m V^{\dagger}_{m(li)}$$



- これはテンソルの，というよりは行列 M の特異値分解である
- 行列の特異値分解には LAPACK の*gesvd が使える
- $D^2 \times D^2$ 行列の特異値分解の計算量は D^6 に比例する

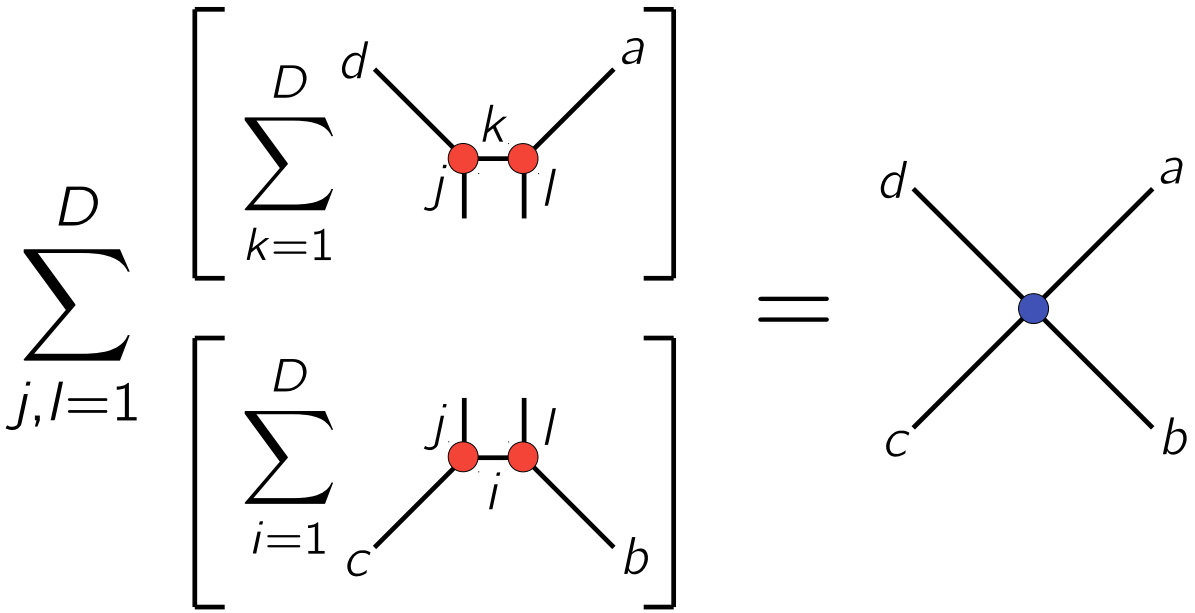
テンソルのかけ算



この計算は，

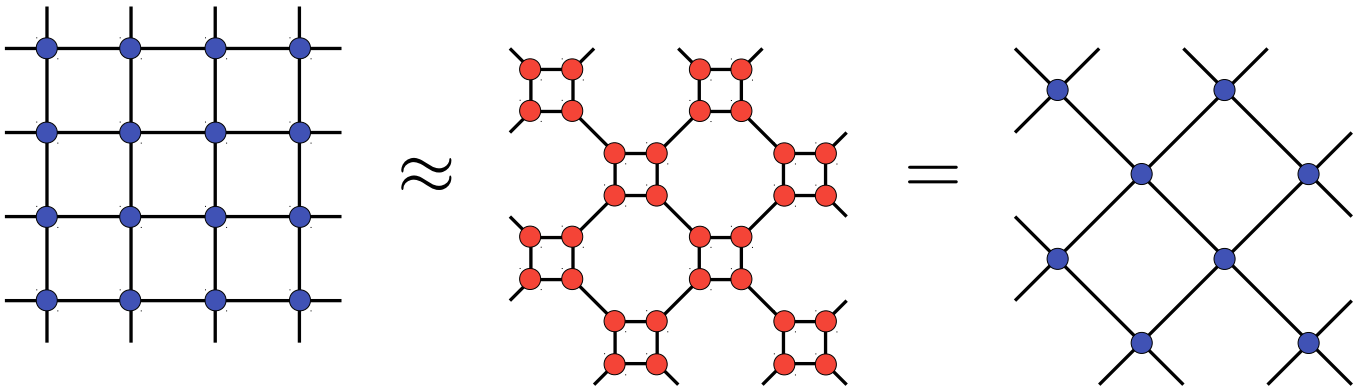
- 「テンソル 4 つのかけ算」として一度に実行すると
  `for(a=1; a<=D; a++){···for(i=1; i<=D; i++){···}}}}}}}}`
  で D^8 に比例する計算量がかかる
- かけ算の順番を工夫することで，行列-行列積の組み合わせと見なせる

テンソルのかけ算



- 括弧の中の行列-行列積の計算量はそれぞれ D^5 に比例する
- 括弧同士の行列-行列積の計算量は D^6 に比例する
- 各部の行列-行列積では BLAS の*gemm が使える
- D^8 から D^6 に，大きく計算量が減らせる！

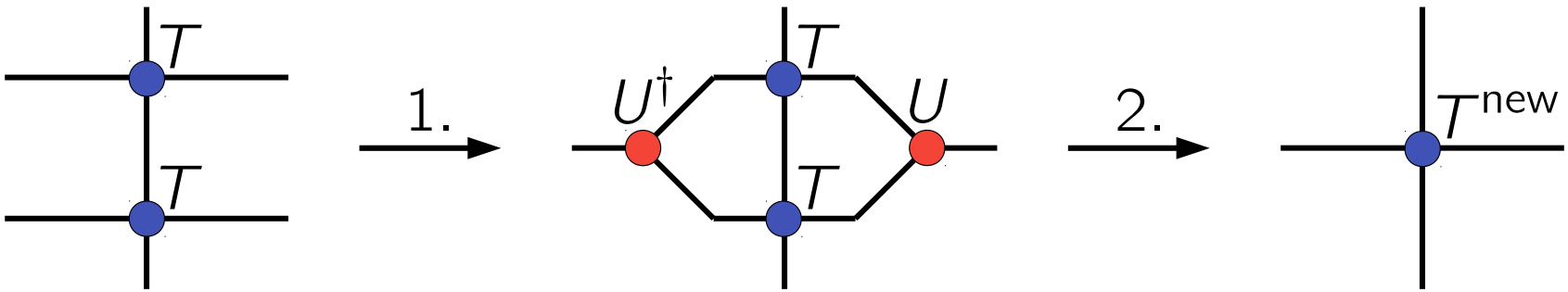
テンソルくりこみ群の計算コスト

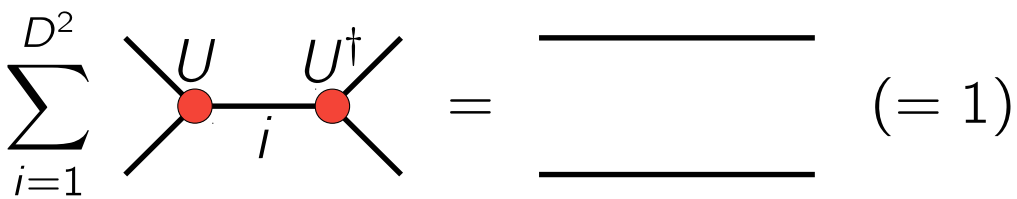


- 特異値分解
  - LAPACK の*gesvd が使えて，計算量は D^6 に比例する
- テンソルのかけ算
  - BLAS の*gemm が使えて，計算量は D^6 に比例する
- 体積依存性
  - 1 回の粗視化で体積が半分になるので，倍々ゲームの要領で簡単に大体積にアプローチできる: 10 回で $V = 1024 \times 1024$, 20 回で $V = 2^{20} \times 2^{20}$!
  - つまり全体の計算量は $D^6 \times \log_2 V$ に比例する
- 必要なメモリサイズ
  - 4 添字のテンソルを保持するので，$D^4 \times$ (1 成分のサイズ [B])

高次テンソルくりこみ群 (2 次元系での説明) [Z. Y. Xie et al., Phys. Rev. B86 (2012)]

1. ユニタリ行列 (アイソメトリー) をネットワークに挿入する
2. T，U，$U^\dagger$ のかけ算をとることで新しいテンソル T^{new} を作る

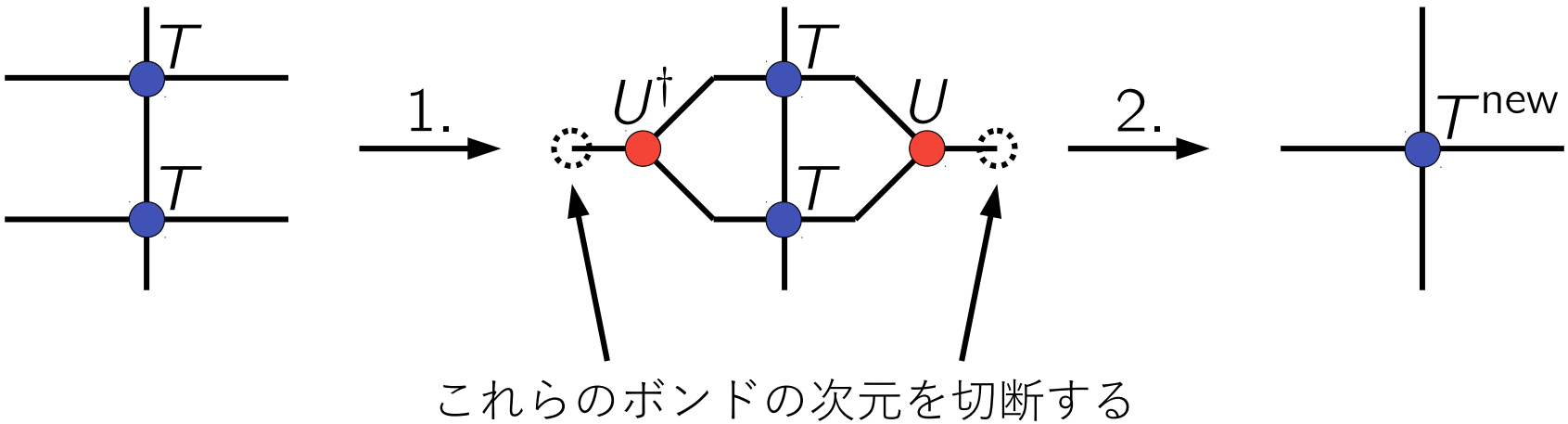




- T: テンソル
- $U^{(\dagger)}$: ユニタリ行列
- D: T のボンド次元

高次テンソルくりこみ群 (2 次元系での説明) [Z. Y. Xie et al., Phys. Rev. B86 (2012)]

1. ユニタリ行列 (アイソメトリー) をネットワークに挿入する
2. T，U，$U^\dagger$ のかけ算をとることで新しいテンソル T^{new} を作る

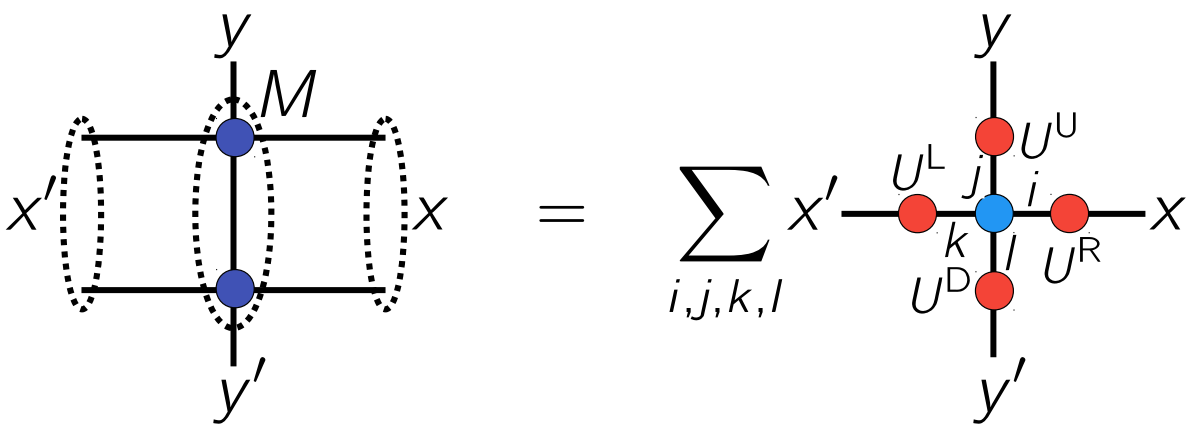


- $D_{\mathrm{cut}} \leq D^2$ とボンド次元を切断することで，この手続きは近似になる
- D_{cut} は T^{new} のサイズを意味するパラメータで，この近似の精度を決める

高次テンソルくりこみ群 (2 次元系での説明) [Z. Y. Xie et al., Phys. Rev. B86 (2012)]

- 近似に使うユニタリ行列は，高次特異値分解 (HOSVD) によって作られる

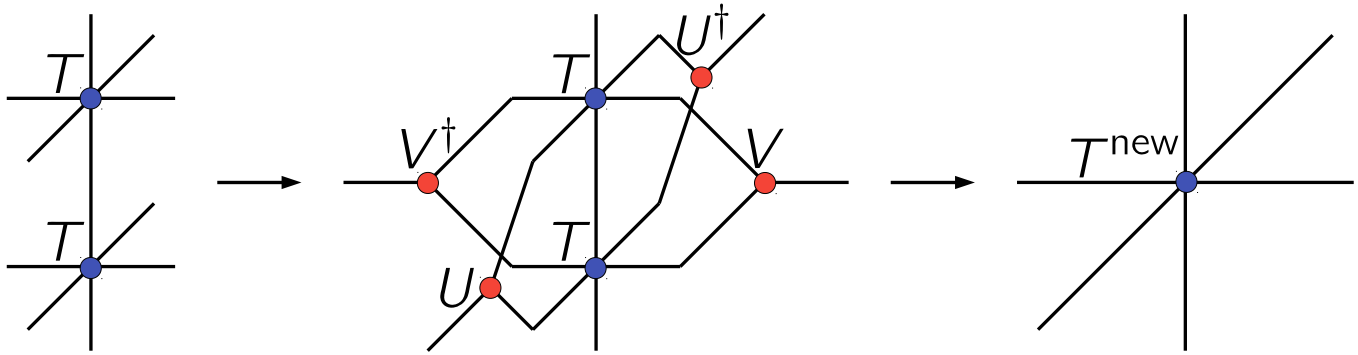
$$M_{xyx'y'} = \sum_{i,j,k,l} S_{ijkl} U^{\mathrm{R}}_{xi} U^{\mathrm{U}}_{yj} U^{\mathrm{L}}_{x'k} U^{\mathrm{D}}_{y'l}.$$



- M: 隣接する 2 つのテンソルの積
- $U^{R\,(L,U,D)}$: 右 (左，上，下) 特徴行列
- S: コアテンソルと呼ばれるテンソル

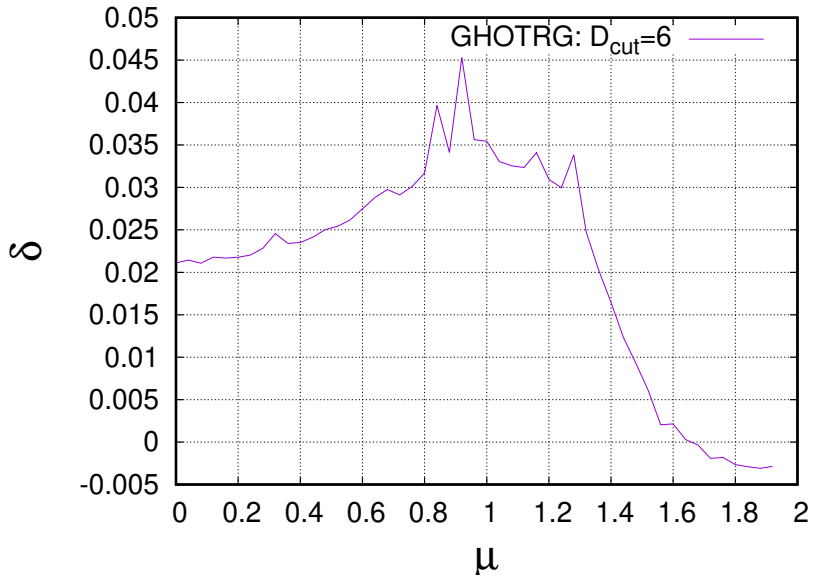
高次テンソルくりこみ群 [Z. Y. Xie et al., Phys. Rev. B86 (2012)]

- 3 次元高次テンソルくりこみ群:



- 任意の次元に自明に拡張できる
- 通常の TRG と同様に，固有値分解 (dsyevd，zheevd) と行列-行列積 (*gemm) を利用した実装に落とし込める
- d 次元系での計算コストは
  - △ 計算時間 $\propto D^{4d-1}$,
  - △ 必要なメモリサイズ $\propto D^{2d}$.
- フェルミオン (G 数) を含む系のための「G 高次テンソルくりこみ群」が我々によって考案されていて，原理的には 4 次元フェルミオン系も扱える

3 次元有限密度自由フェルミオン系の自由エネルギー [Prog. Theor. Exp. Phys. 2017 (2017)]



$$\delta = \frac{\ln Z_{\text{exact}} - \ln Z\left(D_{\text{cut}}\right)}{\left|\ln Z_{\text{exact}}\right|}.$$

- $V = 8 \times 8 \times 8$.
- $\mathcal{O}$ (day) の計算で，∼5%の精度が実現できた.
- (4 次元はまだ難しいが，) 3 次元では非自明な系にアプローチできるかも.
- Ising 模型や Potts 模型の臨界温度を精密に求める試みはなされている.

まとめ

- (高次) テンソルくりこみ群は，現状 3 次元以下の系では有力なアルゴリズムである
- 大変な部分の計算は全て BLAS や LAPACK に任せることができる
- 高次元系では，計算時間もさることながら D^{2d} というメモリサイズもかなり厳しい

d	D	$D^{2d} \times 8$ B
2	8	0.03 MB
2	16	0.5 MB
2	32	8 MB
4	8	128 MB
4	16	32768 MB = 32 GB
4	32	8838860 MB = 8 TB

$\rightarrow$ 分散メモリ並列化が望まれる