テンソルネットワーク法による情報圧縮

東京大学大学院理学系研究科　大久保　毅

大久保の自己紹介

- 1999 -
  - 九州大学理学部物理学科
- 2002 - 2008
  - 物性理論研究室（指導教員：小田垣先生）
  - ランダムパッキング、社会物理学、境界摂動問題
- 2008 - 2012
  - 大阪大学　宇宙地球科学専攻　特任研究員（Supervisor：川村先生）
  - フラストレート磁性体（古典スピン模型の秩序、ダイナミクス等）
- 2012 - 2017
  - 東京大学　物性研究所　特任研究員（Supervisor：川島先生）
  - 脱とじ込め転移、量子スピン模型、テンソルネットワーク等
- 2017 -
  - 東京大学大学院　物理学専攻　特任講師
- 2019.10 -
  - JST さきがけ研究者（量子情報処理領域）
  - テンソルネットワークを活用して、量子コンピュータで量子多体問題を解く

研究の興味：多体系の協力現象一般。相転移、新奇秩序、非平衡ダイナミクス…

研究手法：主に、計算機シミュレーション

この講義の進め方と成績評価

- 講義はスライドを用いて行います
  - 講義スライドのpdfは大久保のwebページからダウンロードできます
    - google検索：「大久保毅 東京大学」で検索
      - https://exa.phys.s.u-tokyo.ac.jp/ja/members/okubo/lecture
  - スライドは日本語の部分と英語の部分が混在しています
- 成績評価
  - 出席点とレポート（最後の回に課題を提示）で評価します

Tentative lecture schedule

1日目	1.現代物理学における巨大なデータと情報圧縮
	2.格子スピン模型の統計力学
	3.線形代数の復習
2日目	4.特異値分解と低ランク近似
	5.テンソルネットワーク繰り込みによる情報圧縮
	6.情報のエンタングルメントと行列積表現
3日目	7.行列積表現の固有値問題への応用
	8.テンソルネットワーク表現への発展
Optional	9.フラストレート磁性体への応用

現代物理学における巨大なデータと情報圧縮

Huge data in physics

（イントロなのでふんわり分かればOK!）

Huge data in physics

Many-body problems in physics

- Celestial movement **（天体運動）**
- Gases, Liquids
- Molecules, Polymers (eg. Proteins), ...
- Electrons in molecules and solids
- Elemental particles (Quantum Chromo Dynamics)
  （量子色力学）

In these problems, "systems" contain huge degrees of freedoms:

$6N$-dimensional phase space for classical mechanics

$O(e^N)$-dimensional Hilbert space for quantum system

Complex particle system

Eg. Poliovirus capsid in electrolyte solution

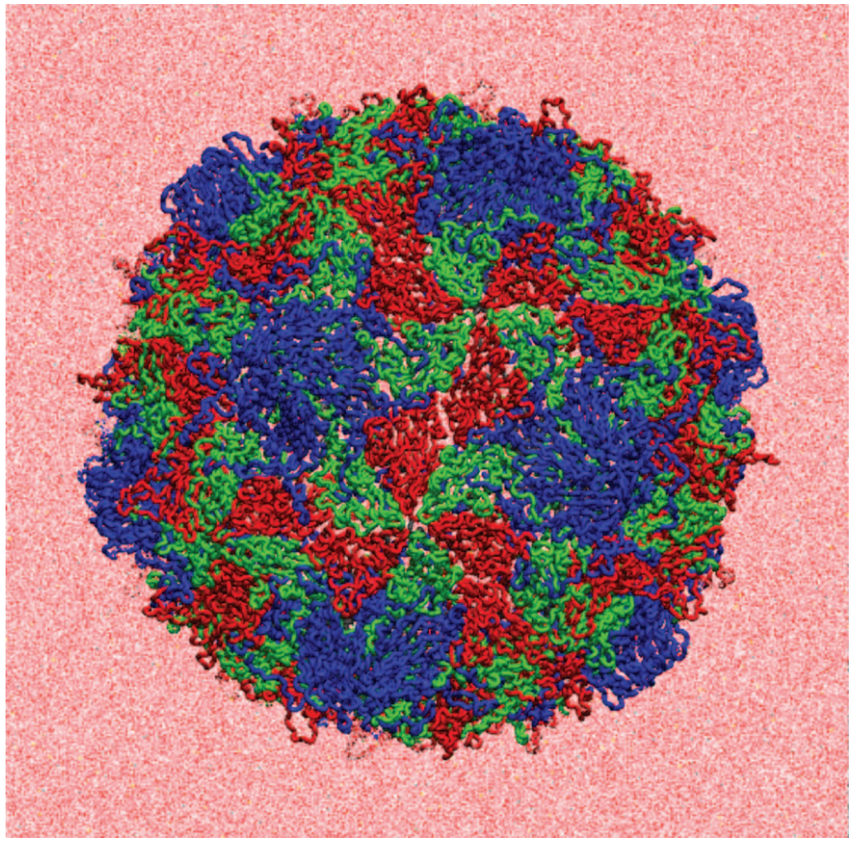
（ポリオウイルス　カプシド）　　（電解質溶液）

Y. Ando et al, J. Chem. Phys. **141**, 165101(2014).

Long-range coulomb interaction

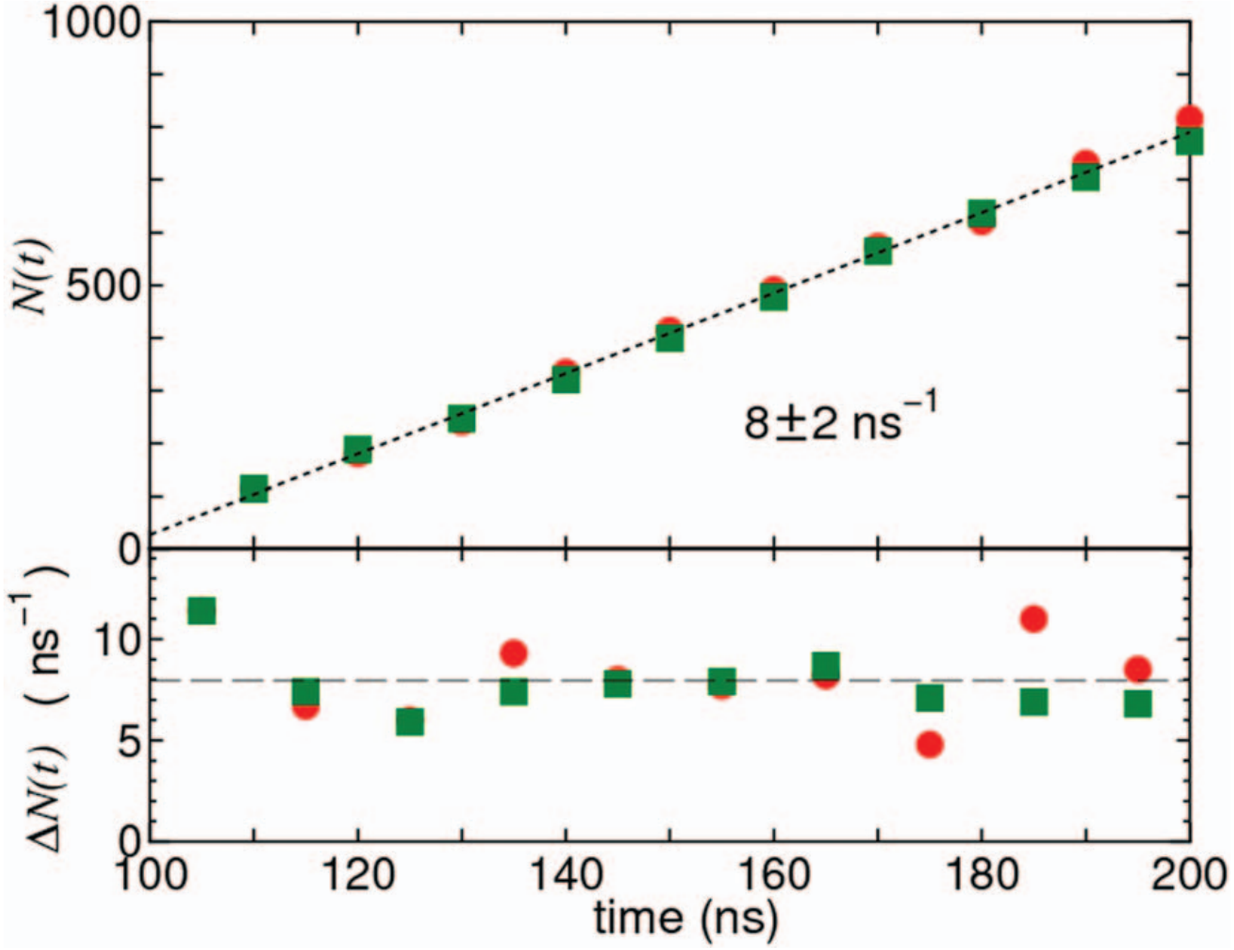
（クーロン相互作用）

Poliovirus capsid



6.5 million atoms

Dynamics of water molecules



Localized electrons as quantum spin systems

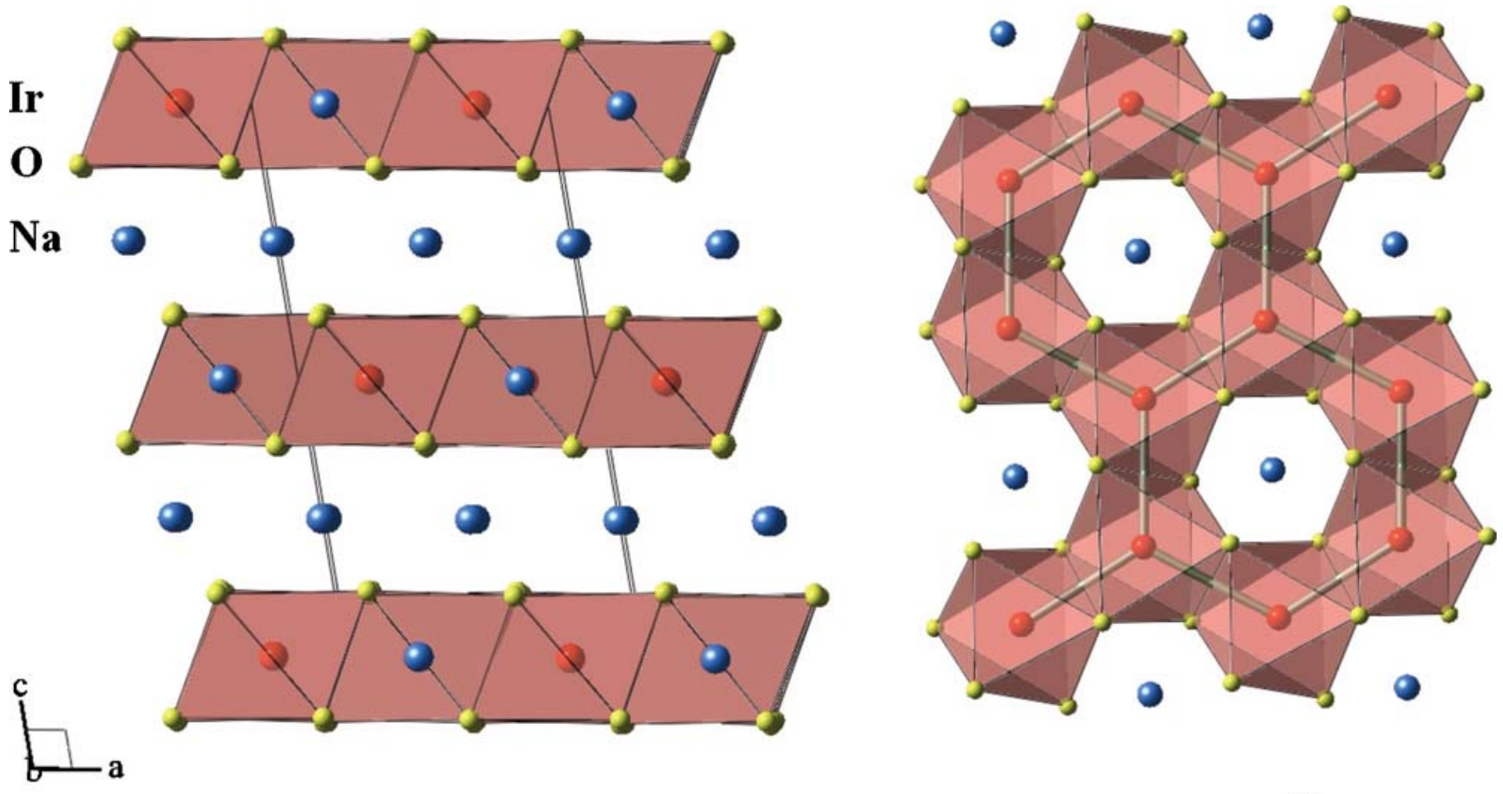
Eg. Antiferromagnetic Mott insulator Na_2IrO_3

（反強磁性）　　（モット絶縁体）

Y. Singh and P. Gegenwart, Physical Review B **82**, 064412 (2010)

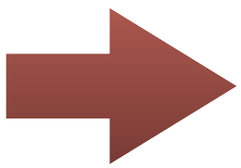
$$\mathcal{H} = \sum_{i,j} J_{ij} S_i S_j$$

S_i : spin operator



Why we need information compression?

1. We can not understand huge information directly.



We try to characterize "systems" thorough a few parameters.

Examples:

Thermodynamics:

Systems are characterized by thermodynamic quantities,

Internal energy, Entropy, Pressure, Volume, Particle number,...

Critical phenomena:

Critical systems are characterized by a few critical exponents.

(臨界指数)

Why we need information compression?

2. We can not treat entire data in the present computers.

Available memories in the present computers:

		Double precision real number = 8 Bite
Personal computers:	~10 GB	$\sim 10^9$
Super computers:	~100 GB / node	$\sim 10^{10}$
K@RIKEN, **Oakforest-PACS**@UTokyo and Tsukuba Univ, **Sekirei**@ISSP, UTokyo ...	~1 PB (whole system)	$\sim 10^{14}$

Notice: In quantum system, the size of Hilbert space is $O(e^N)$

Why we need information compression?

2. We can not treat entire data in the present computers.

→ Try to reduce the "effective" dimension of (Hilbert) space.

By taking proper basis set,
we can represent a quantum state efficiently.

- Krylov subspace
- Matrix product state
- Tensor network states
- ...

Examples of information compression 1

Krylov subspace

linear subspace generated by a square matrix (M) and a vector (v) as

$$\mathcal{K}_n(M, \vec{v}) = \mathrm{span}\left\{\vec{v}, M\vec{v}, M^2\vec{v}, \ldots, M^{n-1}\vec{v}\right\}$$

For quantum many body problems:

$M = \mathcal{H}$:Hamiltonian

$\vec{v} = |\phi\rangle$:wavevector

Solve the eigenvalue problem within
a restricted space (Krylov subspace)

Lanczos method, Arnoldi method

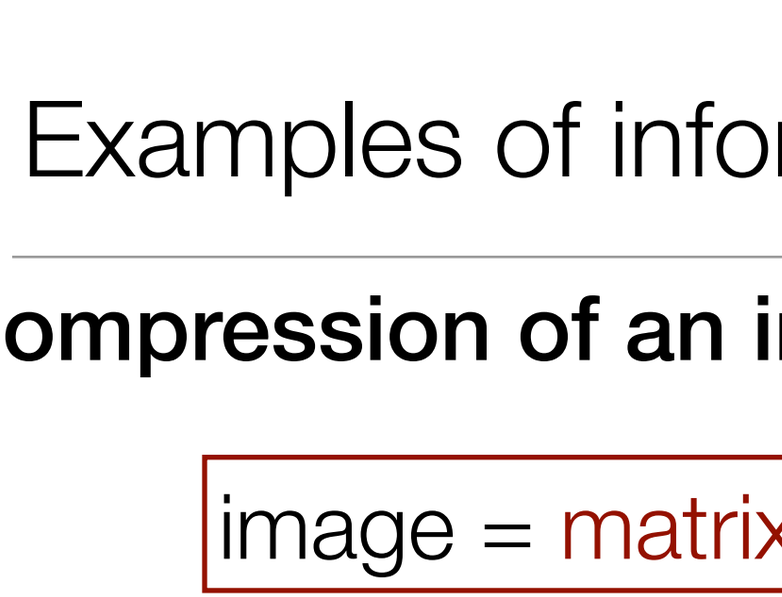
* In these method, we do not necessarily need explicit matrix.
It is enough to know the result of matrix vector multiplication.

Examples of information compression 2

Compression of an image

image = matrix

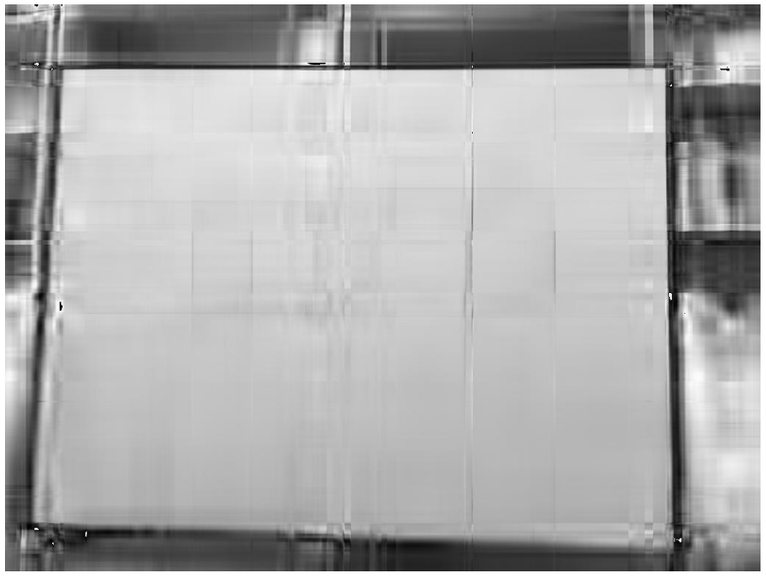
Original



$\chi = 768$

of "singular values"

Compressed



$\chi = 10$

$\chi = 100$

Examples of information compression 2

Compression of a color image

image = tensor

Original

$$\chi = 768$$

About 10% compressed

SVD

HOSVD

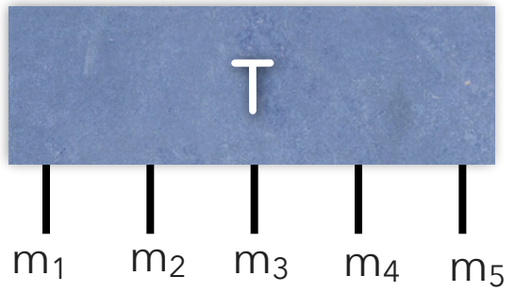
Examples of information compression 3

Wave function: $|\Psi\rangle = \sum_{\substack{\{m_i=\uparrow\downarrow\} \\ \text{or} \\ \{m_i=0,1\}}} T_{m_1,m_2,\cdots,m_N}|m_1,m_2,\cdots,m_N\rangle$

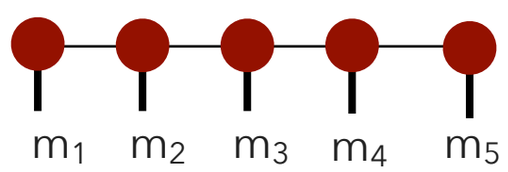
（波動関数）

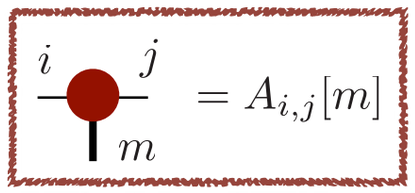
$T_{m_1,m_2,\cdots,m_N}$ N-rank tensor (or Vector)

# of Elements＝2^N



Approximation as
a product of "matrices"





Matrix Product States (**行列積状態**)

（Tensor train decomposition）

$$T_{m_1,m_2,\cdots,m_N} \simeq A_1[m_1]A_2[m_2]\cdots A_N[m_N]$$

$A[m]$ ：Matrix for state m

Singular value decomposition **(特異値分解)**

Singular value decomposition (SVD):

For a K × L matrix M,

$$M = U \Lambda V^\dagger$$

$$M_{i,j} = \sum_m U_{i,m} \lambda_m V^\dagger_{m,j}$$

$U, V^\dagger$: (half) unitary

Λ : Diagonal

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}$$

Singular values: $\lambda_m \geq 0$

Singular vectors:

$$\sum_i U_{i,m} U^\dagger_{m,j} = \delta_i, j$$

$$\sum_i V_{i,m} V^\dagger_{m,j} = \delta_i, j$$

By taking only several larger singular values,
we can approximate M as a lower rank matrix.

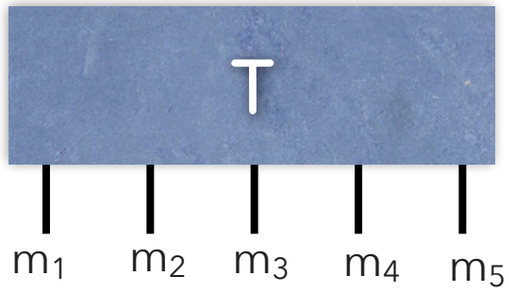
Examples of information compression 3

Wave function: $|\Psi\rangle = \sum_{\substack{\{m_i=\uparrow\downarrow\} \\ \text{or} \\ \{m_i=0,1\}}} T_{m_1,m_2,\cdots,m_N}|m_1,m_2,\cdots,m_N\rangle$

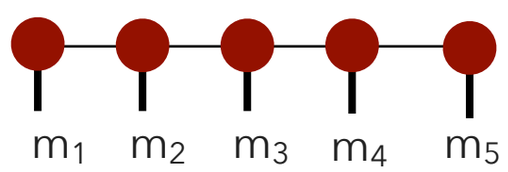
（波動関数）

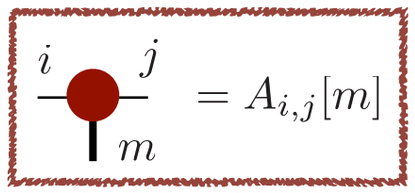
$T_{m_1,m_2,\cdots,m_N}$ N-rank tensor (or Vector)

# of Elements＝2^N



Approximation as
a product of "matrices"





Matrix Product States (**行列積状態**)

（Tensor train decomposition）

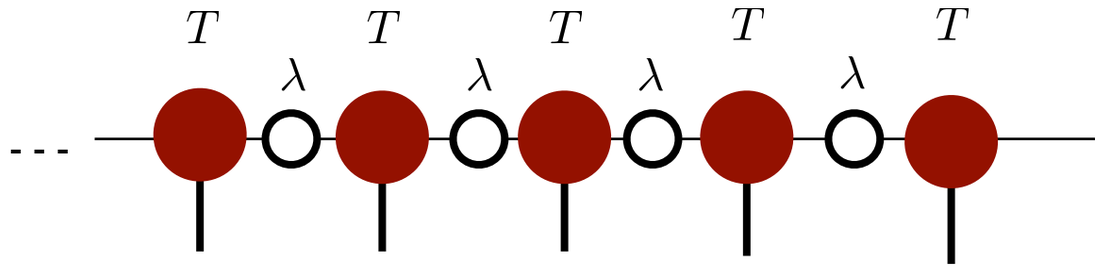
$$T_{m_1,m_2,\cdots,m_N} \simeq A_1[m_1]A_2[m_2]\cdots A_N[m_N]$$

$A[m]$ ：Matrix for state m

Example of MPS: AKLT state

S=1 Affleck-Kennedy-Lieb-Tasaki (AKLT) Hamiltonian:

$$\mathcal{H} = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \frac{J}{3} \sum_{\langle i,j \rangle} \left(\vec{S}_i \cdot \vec{S}_j \right)^2 \qquad (J > 0)$$



The ground state of AKLT model:

S=1 spin:

S=1/2 spin

Spin singlet

χ=2 iMPS: (U. Schollwock, Annals. of Physics **326**, 96 (2011))

$$T[S_z = 1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$T[S_z = 0] = \sqrt{\frac{2}{3}} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad , \lambda = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$T[S_z = -1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}$$

Application of MPS to data science

Tensor train (TT) and tensor ring (TR) decompositions for real data.

COIL-100 dataset = 32 x 32 x 3 x 7200 tensor → Compression by tensor ring decomposition.

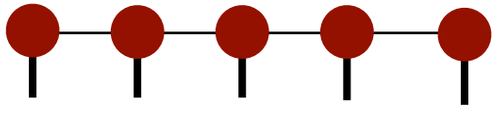
(Q. Zhao, et al arXiv:1606.05535)

error **Rank**

	ϵ	r_{max}	$\bar{r}$	Acc. (%) ($\rho = 50\%$)	Acc. (%) ($\rho = 10\%$)
CP-ALS	0.20	70	70	97.46	80.03
	0.30	17	17	97.56	83.38
	0.39	5	5	90.40	77.70
	0.47	2	2	45.05	39.10
TT-SVD	0.19	67	47.3	99.05	89.11
	0.28	23	16.3	98.99	88.45
	0.37	8	6.3	96.29	86.02
	0.46	3	2.7	47.78	44.00
TR-SVD	0.19	23	12.0	99.14	89.29
	0.28	10	6.0	99.19	89.89
	0.36	5	3.5	98.51	88.10
	0.43	3	2.3	83.43	73.20

Examples of tensor decompositions

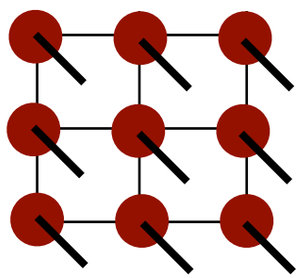
MPS:



Good for 1d gapped systems

(1d correlation in data)

PEPS, TPS:

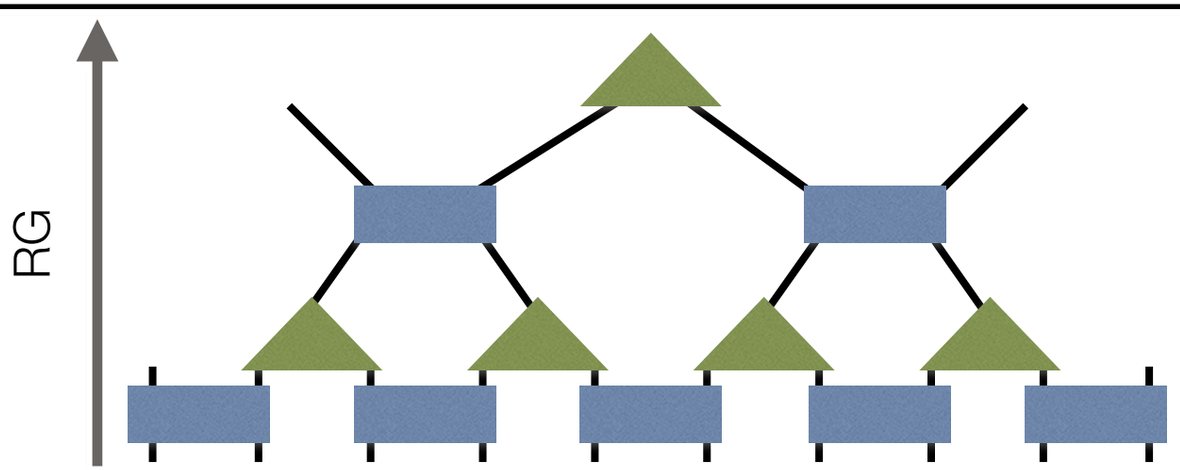


For higher dimensional systems

Extension of MPS

(higher correlation in data)

MERA:

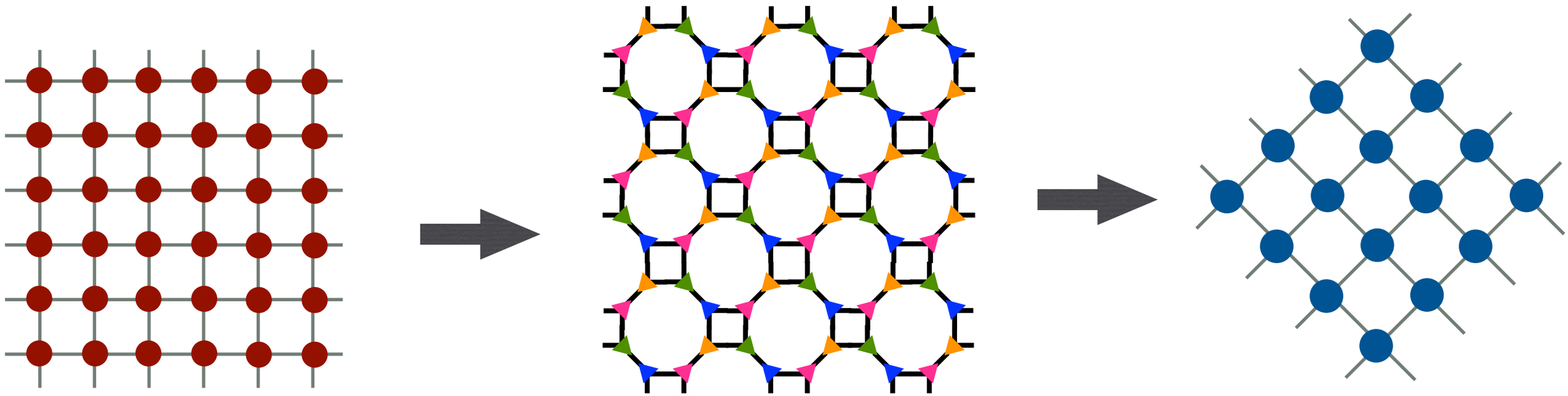


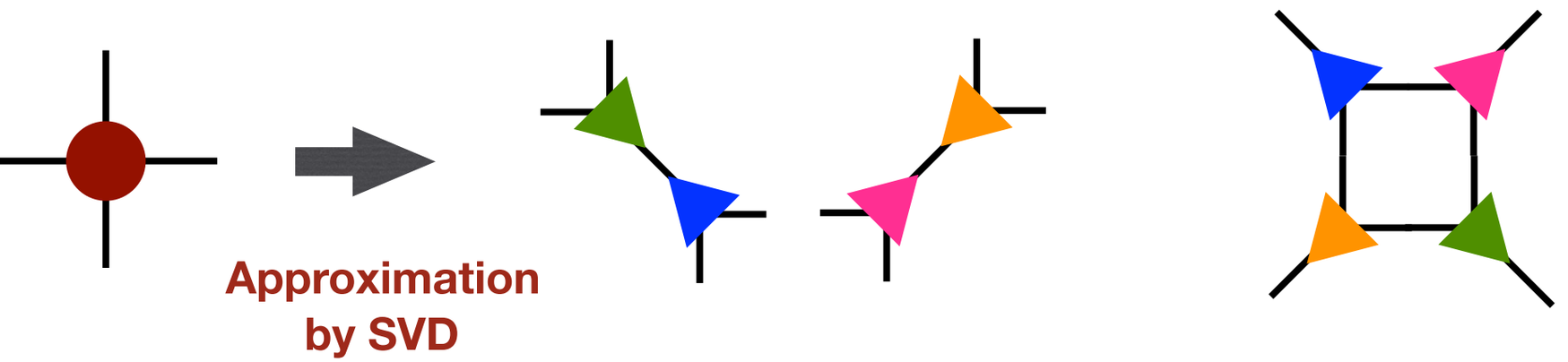
Scale invariant systems

（スケール不変）

Real space renormalization（実空間繰り込み）

Corse graining of a tensor network representing a **scaler.**





Example of calculation

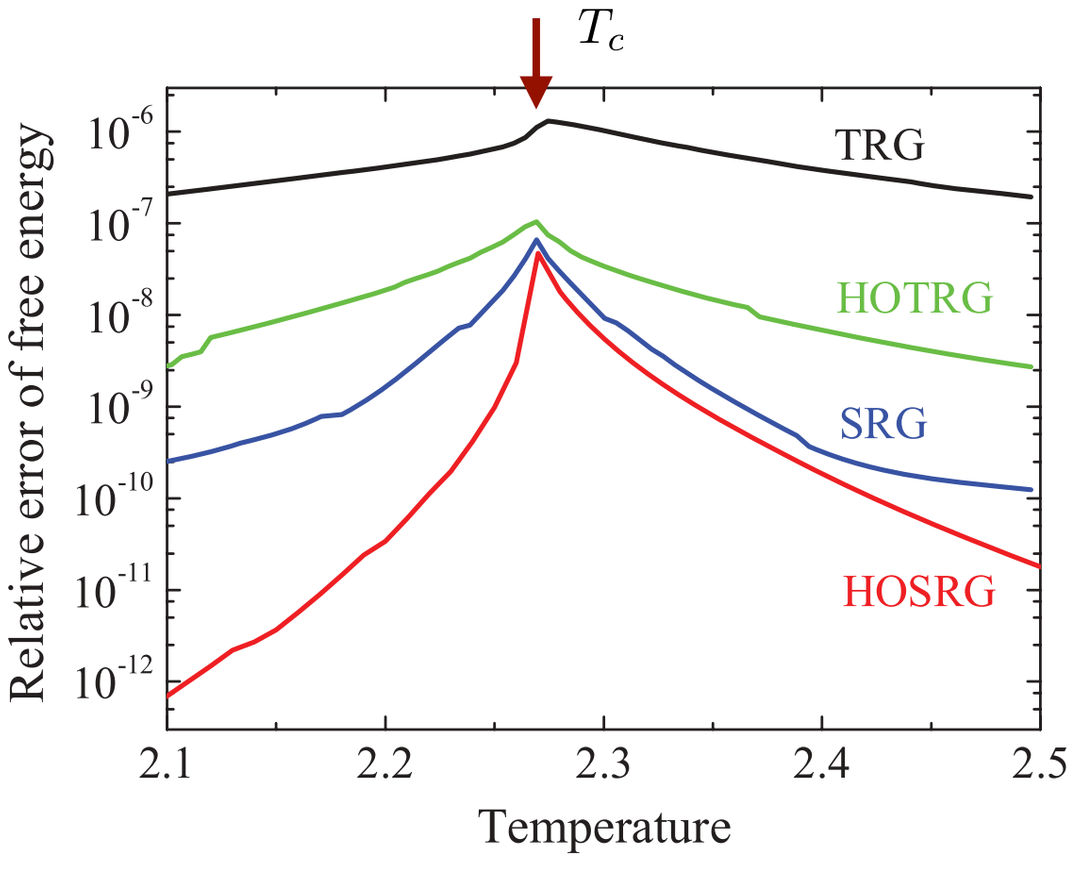
Ising model in infinite size

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

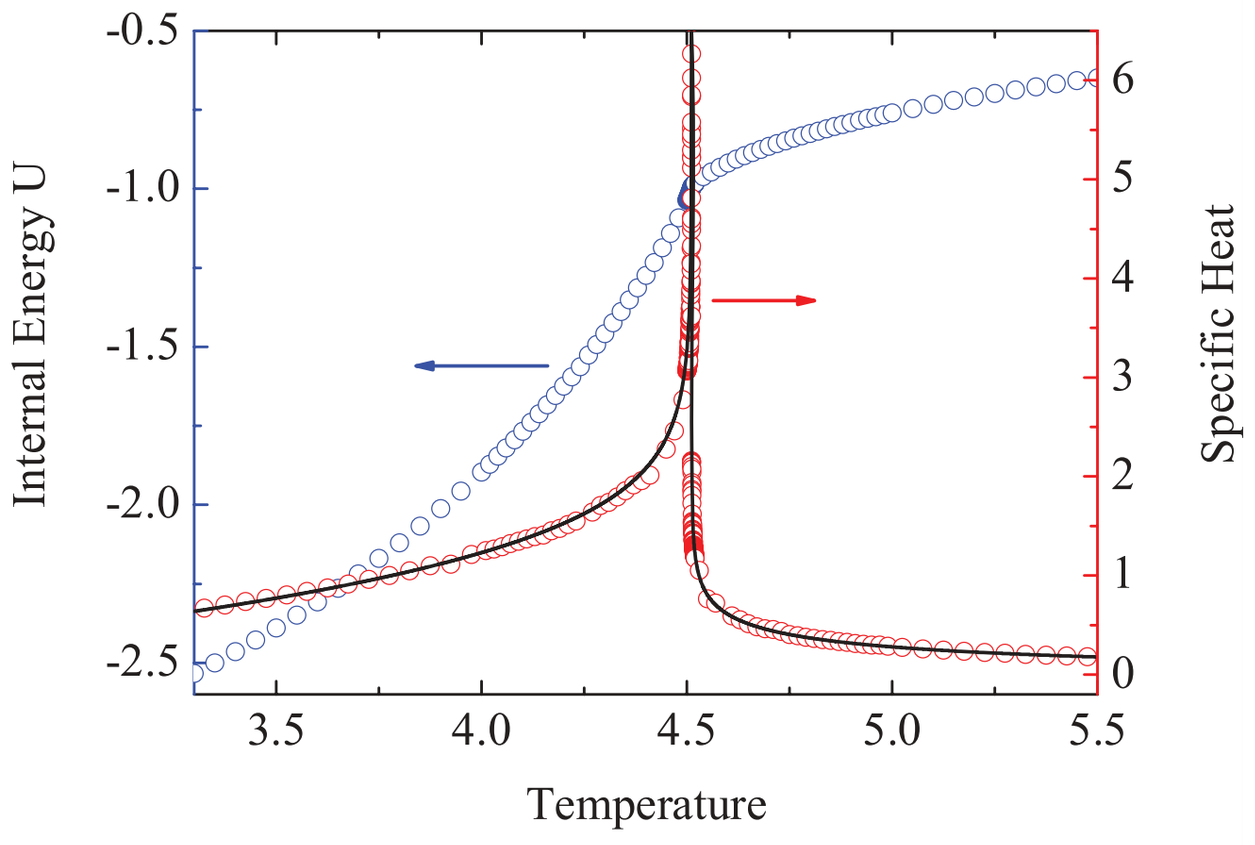
Partition function = tensor network

Z. Y. Xie *et al*, Phys. Rev. B **86**, 045139 (2012)

Error of free energy for 2D Ising model



Energy and specific heat of 3D Ising model



$$T_c/J = \frac{2}{\ln(1+\sqrt{2})} \simeq 2.269$$

Tentative lecture schedule

1日目

1.現代物理学における巨大なデータと情報圧縮

2.格子スピン模型の統計力学

3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似

5. テンソルネットワーク繰り込みによる情報圧縮

6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用

8. テンソルネットワーク表現への発展

Optional

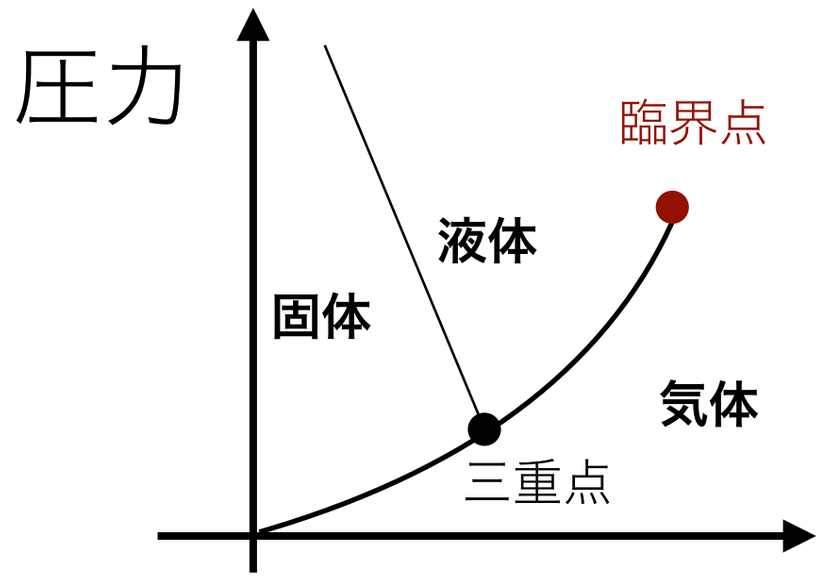
9. フラストレート磁性体への応用

Examples of many body problems:

格子スピン模型と統計力学

相転移

- 温度や圧力等の“パラメタ”を変えると自由エネルギーに異常（特異点）が現れる場合がある。→相転移
  - 相転移で区別された状態＝相
  - 水だと、常圧で温度を下げると気体→液体→固体の３つの相が出現



物性物理の研究対象の一つ

- どんな相があるか？
  - 長距離秩序、トポロジカル秩序、…
- それらを分ける相転移の性質は？

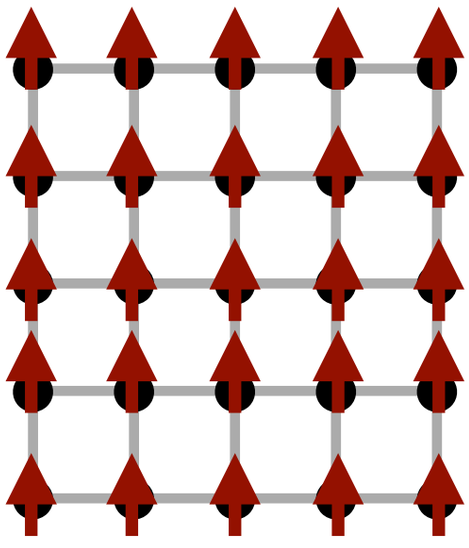
磁性体（スピン模型）の相

典型的には２つの相が存在

磁気秩序相

無秩序相

強磁性



相転移

反強磁性

実際の物質や複雑なスピン模型では、

多種多様な磁気秩序が生じる

１次転移と２次転移

- 相転移には大きく分けて不連続転移と連続転移が存在
  - 不連続転移：

    相転移で自由エネルギーの１階微分が不連続に変化＝１次転移
    - 例：液体⟷固体の相転移....
  - 連続転移：

    自由エネルギーの１階微分は連続に変化
    - 多くの場合、２階微分が不連続に変化する＝２次転移
    - 例：気体⟷液体の臨界点、イジング模型の相転移...

臨界現象

２次転移では臨界現象が生じる

相転移点（臨界点）では、特徴的な長さスケールが発散

→ スケール不変性

種々の物理量が非自明なべき関数の振る舞い

相関長：　$\xi \sim |T - T_c|^{-\nu}$

比熱：　$C \sim |T - T_c|^{-\alpha}$

感受率：　$\chi \sim |T - T_c|^{-\gamma}$

べき指数＝臨界指数

ユニバーサリティ

臨界指数は相転移で“破れる”対称性と空間次元で決まり系の詳細には依存しない

→ 臨界現象は対称性に注目したシンプルな模型で調べられる

統計力学とカノニカル分布

カノニカル分布

$$P(\Gamma) \propto e^{-\beta \mathcal{H}(\Gamma)}$$

Γ ：状態（例えば、粒子の位置・運動量）

$P(\Gamma)$: Γが実現する確率

$\beta = \dfrac{1}{k_B T}$: 逆温度

$\mathcal{H}$ ：ハミルトニアン

分配関数＝カノニカル分布の規格化因子

$$Z = \sum_{\Gamma} e^{-\beta \mathcal{H}(\Gamma)}$$

熱力学自由エネルギーとの関係

$$F = -k_B T \ln Z$$

分配関数の対数＝自由エネルギー

カノニカル分布での物理量の期待値

物理量Oの期待値：$\langle O \rangle \equiv \frac{1}{Z} \sum_{\Gamma} O(\Gamma) e^{-\beta \mathcal{H}(\Gamma)}$

物理量の期待値⟷マクロな系で観測される物理量

→ すべての状態の和が計算出来れば、
熱力学量が分かる

現実：$\sum_{\Gamma}$ はとてつもなく大きいので、手では計算できない
（計算機を使っても厳密に計算するのは難しい）

計算科学の手法＝分配関数・期待値を数値的に計算する

格子スピン模型：格子

格子スピン模型：

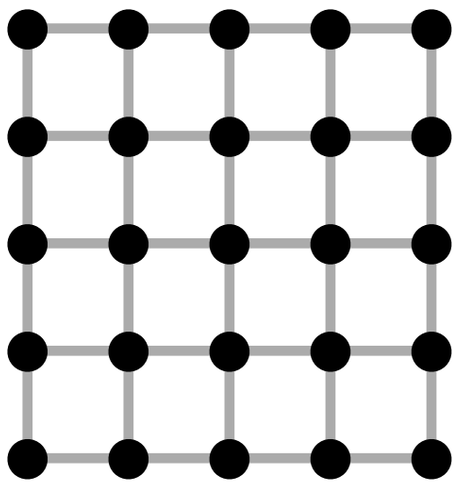
格子上にスピン自由度が定義され相互作用する模型

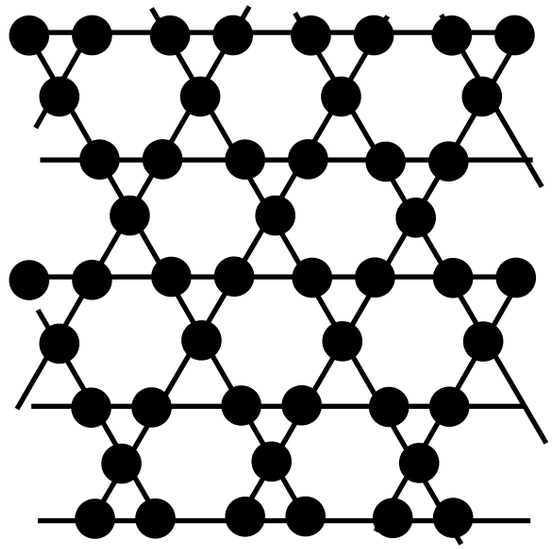
格子

1 次元　　鎖

正方格子　　カゴメ格子

2 次元





3 次元　　立方格子，FCC格子，…

格子スピン模型：スピン自由度

スピン自由度 S_i

＊量子スピン

S_i は角運動量演算子　　$S_i = (S_i^x, S_i^y, S_i^z)$

スピン量子数Sで特徴付け　$S = \frac{1}{2}, 1, \frac{3}{2}, 2, \cdots$

Sが小さいほど量子効果が強く、新規量子相が実現する可能性

＊古典スピン

$S \to \infty$ の極限に対応する

イジングスピン：　$S_i = \pm 1 = \uparrow, \downarrow$　　上か下かしか向かない

ハイゼンベルグスピン：　$S_i = (S_i^x, S_i^y, S_i^z)$

3成分の単位ベクトル：　$(S_i^x)^2 + (S_i^y)^2 + (S_i^z)^2 = 1$

普通の磁気秩序は古典スピンで十分理解できる

格子スピン模型：相互作用

典型的なハミルトニアン：スピン自由度の２体相互作用

$$\mathcal{H} = -\sum_{i,j} J_{ij} S_i S_j$$

$J_{ij} > 0$：同じ向きを向くと得　（強磁性相互作用）

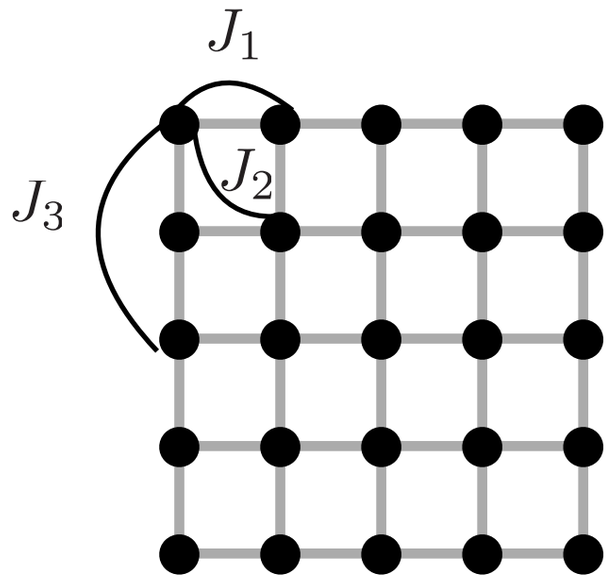
$J_{ij} < 0$：反対向きを向くとエネルギーが得（反強磁性相互作用）

現実の物質では J_{ij} は $|\boldsymbol{r}_i - \boldsymbol{r}_j| \to$ 大で十分に小さくなる

➡ 最近接格子点間のみの相互作用を考えるのが良い近似

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

$\sum_{\langle i,j \rangle}$：最近接格子点ペアの和



スピン模型と相転移

例：正方格子強磁性イジング模型

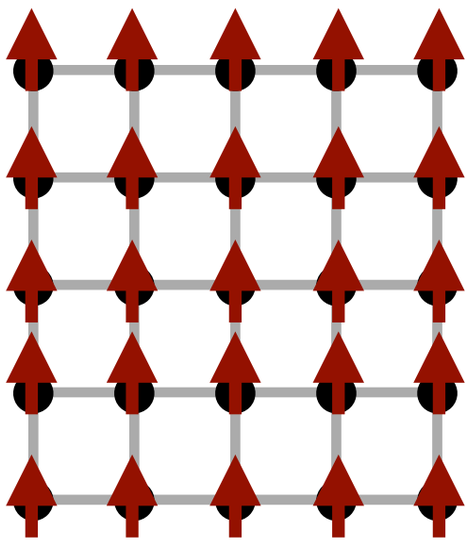
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

自由エネルギー：

$$F = E - ST$$

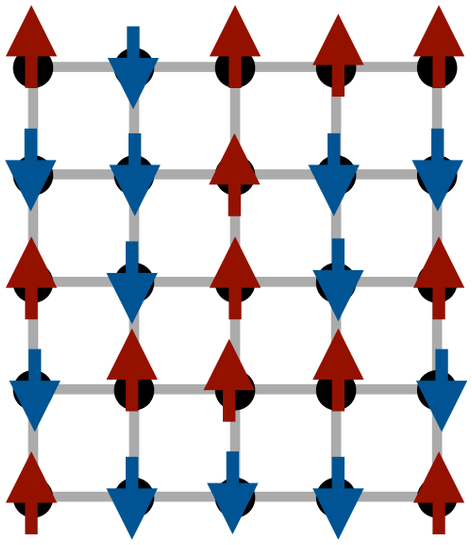
低温：エネルギーが小さい方がFが小さい
高温：エントロピーが大きい方がFが小さい

低エネルギー



相転移！

高エントロピー



（磁化の和がほぼゼロになる状態はたくさん）

イジング模型の分配関数

統計力学の処方箋：

分配関数を計算したい

$$Z = \sum_{\{S_i = \pm 1\}} e^{\beta J \sum_{\langle i,j \rangle} S_i S_j} \qquad \mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

しかし、和 $\sum_{\{S_i = \pm 1\}}$ は、スピンがN個の時、 2^N

で指数的に大きい！（N=100でも、10^{30}の項がある！）

→ 定義通りに計算することは困難

→ 計算科学によるアプローチ

（古典）格子スピン模型の数値解法

- 大きく分けて２種類のアプローチ
  - 乱数を使って物理量の期待値を統計誤差つきで求める
    - 乱択アルゴリズム、モンテカルロ法
    - 統計誤差の範囲内で得られる期待値は厳密
  - 分配関数を近似的に計算する
    - 得られた分配関数には近似に基づく系統的な誤差
    - 誤差は計算の規模を増大することで減らせる
    - 転送行列法、テンソルネットワーク法

Quantum spin systems
（テンソルネットワーク法の主要な対象）

Quantum systems

Quantum system: governed by Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \mathcal{H}|\Psi\rangle$$

$\mathcal{H}$:Hamiltonian

$|\Psi\rangle$:Wave function (state vector)
（波動関数 or 状態ベクトル）

Inner product:

$$(|a\rangle, |b\rangle) = \langle b|a\rangle$$

Nature: Elementary particles, e.g. electrons, obey quantum mechanics.
素粒子

➡ Static problems: Time-independent Schrödinger equation

$$\mathcal{H}|\Psi\rangle = \underline{E}|\Psi\rangle$$
Energy

= Eigenvalue problem

Quantum systems

Example of quantum system: Array of quantum bits

1 bit ● A quantum bit is represented by two basis vectors.

$$|0\rangle, |1\rangle \quad \text{or} \quad (|\uparrow\rangle, |\downarrow\rangle)$$

2 bits ●—● The Hilbert space is spanned by four basis vectors.

ヒルベルト空間

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

Simple notation: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$

$$\Rightarrow |\Psi\rangle = \sum_{\alpha,\beta=0,1} C_{\alpha,\beta}|\alpha\beta\rangle$$

$C_{\alpha,\beta}$:complex number

The Hamiltonian for 2 bits system can be represented in these bases.

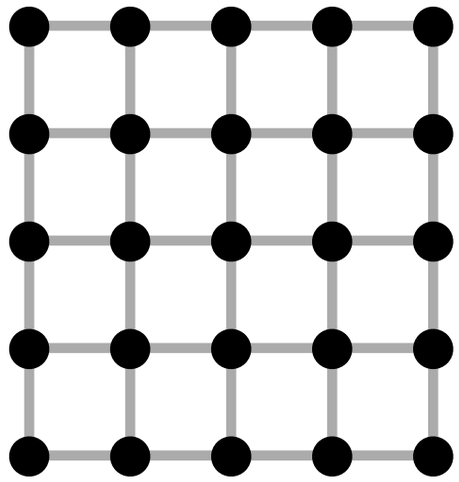
$$\Rightarrow \mathcal{H} \to \begin{pmatrix} H_{0,0;0,0} & H_{0,0;0,1} & H_{0,0;1,0} & H_{0,0;1,1} \\ H_{0,1;0,0} & H_{0,1;0,1} & H_{0,1;1,0} & H_{0,1;1,1} \\ H_{1,0;0,0} & H_{1,0;0,1} & H_{1,0;1,0} & H_{1,0;1,1} \\ H_{1,1;0,0} & H_{1,1;0,1} & H_{1,1;1,0} & H_{1,1;1,1} \end{pmatrix}$$

Matrix element: $H_{\alpha,\beta;\alpha',\beta'} \equiv \langle\alpha\beta|\mathcal{H}|\alpha'\beta'\rangle$

Quantum systems

Example of quantum system: Array of quantum bits

N bits: Dimension of the Hilbert space $= 2^N$



➡ Hamiltonian is $2^N \times 2^N$ matrix

Need to solve eigenvalue problem of huge matrix!

In physics,

- We often interested in the "ground state" (smallest eigenvalue)

基底状態

➡ We can concentrate to a special state.

- Typical system only has "short range" interactions

➡ Hamiltonian matrix becomes sparse.

(Quantum) spin system

Spin systems:

Spin degree of freedoms defined on a lattice and interact each other

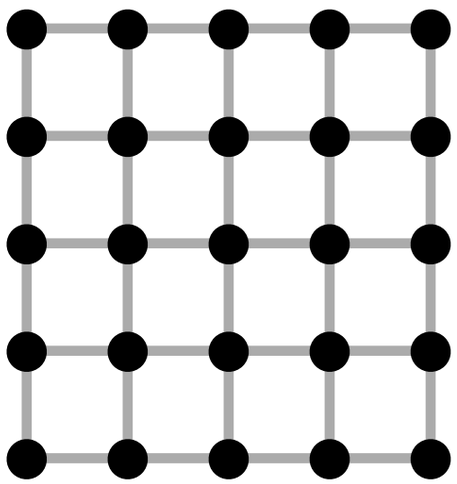
Lattice

1D

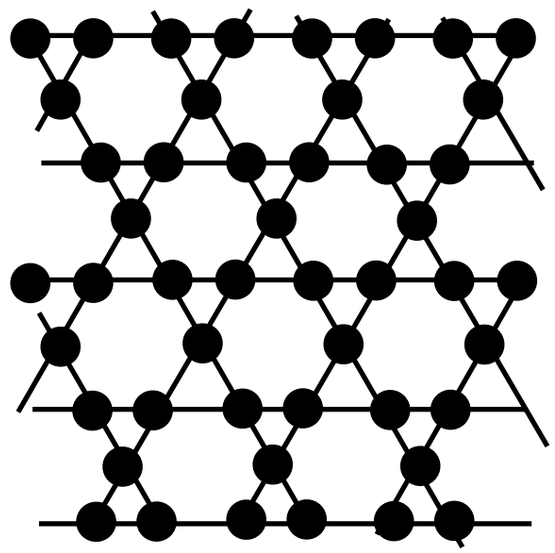
Chain

2D

Square lattice



Kagame lattice



3D simple cubic, FCC lattice, BCC lattice, …

Quantum spin

Spin operator: (S_x, S_y, S_z)

Commutation relation

（交換関係）

$$[S_x, S_y] = i\hbar S_z, [S_y, S_z] = i\hbar S_x, [S_z, S_x] = i\hbar S_y$$

$$[A, B] \equiv AB - BA$$

Spin quantum number operator:

（スピン量子数）

$$S^2 = S_x^2 + S_y^2 + S_z^2$$

Simultaneous eigenstate of S^z and S^2: $|S_z, S\rangle$

$$S^2|S_z, S\rangle = \hbar^2 S(S+1)|S_z, S\rangle$$
$$S_z|S_z, S\rangle = \hbar S_z|S_z, S\rangle$$

Quantized spin number

$$S = \frac{1}{2}, 1, \frac{3}{2}, 2, \ldots$$
$$S_z = -S, -S+1, \ldots, S-1, S$$

(Hereafter, we set $\hbar = 1$)

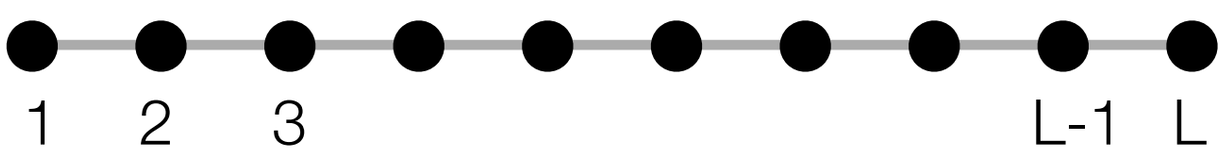
Quantum spin: S=1/2

Matrix representation of the spin operators: $S = \frac{1}{2}$

$$S_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, S_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, S_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

We can consider S=1/2 spin as a quantum bit : $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Spins on a chain:



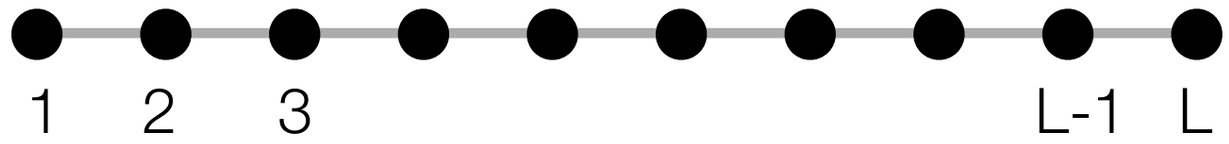
"Transverse field Ising model"　（横磁場イジング模型）

$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z} S_{i+1,z} - \Gamma \sum_{i=1}^{L} S_{i,x}$$

L=2

$$\mathcal{H} = \begin{pmatrix} -1/4 & -\Gamma/2 & -\Gamma/2 & 0 \\ -\Gamma/2 & 1/4 & 0 & -\Gamma/2 \\ -\Gamma/2 & 0 & 1/4 & -\Gamma/2 \\ 0 & -\Gamma/2 & -\Gamma/2 & -1/4 \end{pmatrix}$$

Quantum spin: S=1/2

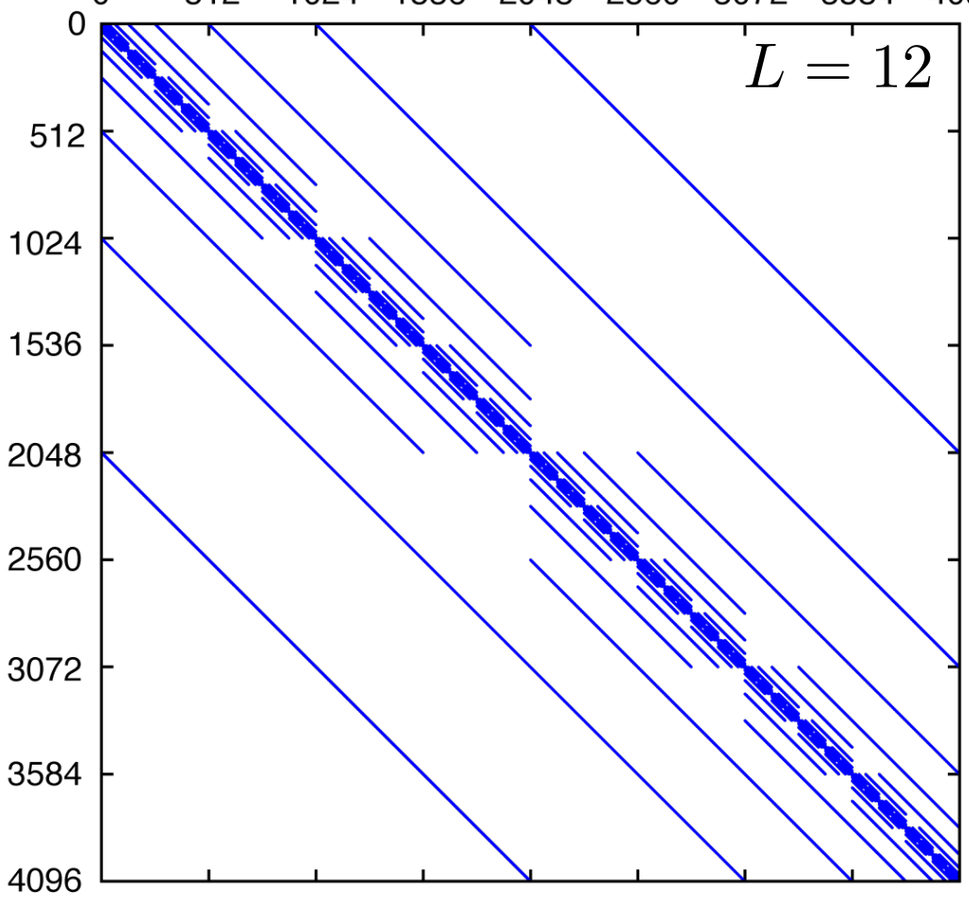


"Transverse field Ising model"

$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z} S_{i+1,z} - \Gamma \sum_{i=1}^{L} S_{i,x}$$

Non-zero elements in the Hamiltonian

(Figure from Yamaji-sensei)



Total matrix elements=2^{2L}

# of non-zero elements $\sim O(Le^L)$

Sparse!

Classical problems

Two types of classical many-body problems

1. Approximation of quantum problems

Nature: Elementary particles obey quantum mechanics.

$$i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \mathcal{H}|\Psi\rangle$$

➡ Classical mechanics is an approximation

2. Pure classical problems

Classical problems not necessary based on quantum mechanics

- Percolation, covering, packing, …
- Stochastic process, “dynamical” system, ..
- Critical phenomena
- …

Classical problems as an approximation: magnetism

Electron Spin: "Quantum" degree of freedom

For accurate treatment, the spin quantum number S is important

S=1/2, 1, 3/2, …

However, we can approximate the system by taking the limit of $S \to \infty$.
"classical" spin model

- Classical Heisenberg model
- Anisotropy: Ising model, XY model
- ….

Classical spin degree of freedom

Classical spin:

1. $S \to \infty$ limit of quantum spin
2. simple degree of freedom reflecting symmetry

1. Heisenberg spin $S_i = (S_i^x, S_i^y, S_i^z)$

Three component unit vector: $(S_i^x)^2 + (S_i^y)^2 + (S_i^z)^2 = 1$

A lot of magnetism can be understand through classical Heisenberg spin

2. Ising spin $S_i = \pm 1 = \uparrow, \downarrow$
   - Strong easy axis anisotropy
   - Representing underlying Z_2 symmetry
3. XY spin $S_i = (S_i^x, S_i^y)$ Two component unit vector: $(S_i^x)^2 + (S_i^y)^2 = 1$
   - Strong easy plane anisotropy
   - Representing underlying U(1) symmetry

Classical Ising spin vs. quantum spin

Ising spin

$$S_i = \pm 1 = \uparrow, \downarrow$$

"Ising model"

$$\mathcal{H} = -\sum_{i=1}^{L-1} S_i S_{i+1} - h \sum_{i=1}^{L} S_i$$

$$\mathcal{H} = \begin{pmatrix} -1-2h & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1+2h \end{pmatrix}$$

S=1/2 quantum spin

$$S_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, S_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, S_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

"Transverse field Ising model"

$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z} S_{i+1,z} - \Gamma \sum_{i=1}^{L} S_{i,x}$$

$$\mathcal{H} = \begin{pmatrix} -1/4 & -\Gamma/2 & -\Gamma/2 & 0 \\ -\Gamma/2 & 1/4 & 0 & -\Gamma/2 \\ -\Gamma/2 & 0 & 1/4 & -\Gamma/2 \\ 0 & -\Gamma/2 & -\Gamma/2 & -1/4 \end{pmatrix}$$

In the case of classical system, the Hamiltonian is "diagonal".

- We do not need explicit diagonalization.
- "State" can be represented by a product of local DOF.

$$\sim O(L)$$

(Degrees Of Freedom)

- **Although, # of states is** $\sim O(2^L)$

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

線形代数の復習

（必要なければ飛ばします）

Outline

- Vector space- Abstract vectors-
  - General vector space (with inner product)
  - Basis and relation to coordinate vector space
  - Vector subspace and spanned vector subspace
- Matrix and linear map
  - Relation between matrices and linear maps
  - Important properties and operations for matrices
  - Relation to simultaneous linear equations
- Eigenvalue problem and diagonalization

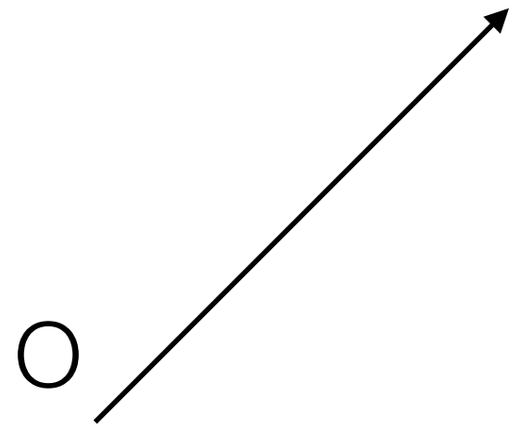
Vector space -Abstract vectors-

Geometric vector

Geometric vector: Arrow on the plane (or the space) ,

which has "Direction" and "Length"

$$\vec{v} \equiv \overrightarrow{OP}$$



We can express a vector by its component:

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} x_p - x_o \\ y_p - y_o \\ z_p - z_o \end{pmatrix}$$

Properties of vector

Properties of addition:

$$\vec{a}+\vec{b}=\vec{b}+\vec{a}$$ Commutative property（交換法則）

$$\left(\vec{a}+\vec{b}\right)+\vec{c}=\vec{a}+\left(\vec{b}+\vec{c}\right)$$ Associative property（結合法則）

$$\vec{a}+\vec{0}=\vec{a}$$ zero vector

$$\vec{a}+(-\vec{a})=\vec{0}$$ inverse vector

Multiplication of scaler $c \in \mathbb{R}$（実数）：

$$c(\vec{a}+\vec{b})=c\vec{b}+c\vec{a}$$
$$(c+d)\vec{a}=c\vec{a}+d\vec{a}$$ Distributive property（分配法則）
$$(cd)\vec{a}=c\,(d\vec{a})$$

Inner product of vector

Inner product:

$$(\vec{a}, \vec{b}) \equiv \vec{a} \cdot \vec{b}$$

$$= a_x b_x + a_y b_y + a_z b_z$$

Example:

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}, \vec{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$

Properties:

$$(\vec{a}, \vec{a}) \geq 0$$

$$(\vec{a}, \vec{b}) = (\vec{b}, \vec{a})$$

$$(\vec{a} + \vec{b}, \vec{c}) = (\vec{a}, \vec{c}) + (\vec{b}, \vec{c})$$

$$(c\vec{a}, \vec{b}) = c(\vec{a}, \vec{b}) \qquad c \in \mathbb{R}$$

Norm (length):

$$\|\vec{a}\| \equiv \sqrt{(\vec{a}, \vec{a})}$$

Vector space (linear space)

Vector space $\mathbb{V}$: generalization of geometric vector

Set of elements (vectors) satisfying following **axioms**（公理）

Properties of addition:

$$\vec{a}+\vec{b}=\vec{b}+\vec{a}$$ Commutative property（交換法則）

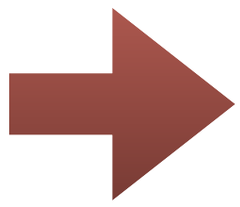
$$\left(\vec{a}+\vec{b}\right)+\vec{c}=\vec{a}+\left(\vec{b}+\vec{c}\right)$$ Associative property（結合法則）

$$\vec{a}+\vec{0}=\vec{a}$$ Existence of unique zero vector

$$\vec{a}+(-\vec{a})=\vec{0}$$ Existence of unique inverse vector

Multiplication of scaler c :

$$c(\vec{a}+\vec{b})=c\vec{b}+c\vec{a}$$
$$(c+d)\vec{a}=c\vec{a}+d\vec{a}$$
$$(cd)\vec{a}=c\,(d\vec{a})$$



$c \in \mathbb{R}$: Real vector space

$c \in \mathbb{C}$: Complex vector space

Inner product space (metric vector space)
(計量空間)

Inner product space:

Vector space + definition of inner product

Inner product: $(\vec{a}, \vec{b})$

Axiom:

$$(\vec{a}, \vec{a}) \geq 0$$

$$(\vec{a}, \vec{b}) = (\vec{b}, \vec{a})^*$$

$$(\vec{a} + \vec{b}, \vec{c}) = (\vec{a}, \vec{c}) + (\vec{b}, \vec{c})$$

$$(c\vec{a}, \vec{b}) = c(\vec{a}, \vec{b})$$

*If a norm defined from the inner product is "complete"（完備）,

that space is called **Hilbert space**.

Examples of vector spaces

(1) Coordinate space（数ベクトル空間） $\mathbb{R}^n, \mathbb{C}^n$

Vector: $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \quad v_i \in \mathbb{R} \text{ or } \mathbb{C}$

Inner product: $(\vec{a}, \vec{b}) \equiv \vec{a} \cdot \vec{b}^*$

(2) Wave vectors in quantum physics

Vector: $|\Psi\rangle$

Inner product: $(|a\rangle, |b\rangle) = \langle b|a\rangle$

Linearly independent or dependent

（線形独立）　（線形従属）

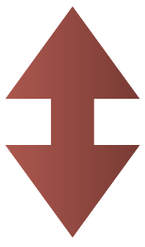
Linear combination:

$$\vec{x} = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots c_k\vec{v}_k$$

$$\vec{v}_i \in \mathbb{V} \qquad c_i \in \mathbb{R} \text{ or } \mathbb{C}$$

A set $\{\vec{v}_1, \vec{v}_2, \cdots \vec{v}_k\}$ is **linearly independent** when

$\vec{x} = \vec{0}$ is satisfied if and only if $c_1 = c_2 = \cdots = c_k = 0$



A set $\{\vec{v}_1, \vec{v}_2, \cdots \vec{v}_k\}$ is **linearly dependent** when

it is not linearly independent.

Basis of vector space
（基底）

A set $\{\vec{e}_1, \vec{e}_2, \cdots \vec{e}_n\}$ $(\vec{e}_i \in \mathbb{V})$ is a basis（基底）of $\mathbb{V}$ when

$\{\vec{e}_1, \vec{e}_2, \cdots \vec{e}_n\}$ is linearly independent.

and

Any vectors in $\mathbb{V}$ are represented by its linear combination.

$\vec{e}_i$: basis vector

# of basis vectors (n) is called **dimension**（次元）of $\mathbb{V}$.

$$n = \dim \mathbb{V}$$

Relation (map) to coordinate vector space

By using a basis $\{\vec{e}_1, \vec{e}_2, \cdots \vec{e}_n\}$, $\vec{v} \in \mathbb{V}$ is uniquely represented as

(* From linear independency)

$$\vec{v} = v_1\vec{e}_1 + v_2\vec{e}_2 + \cdots v_n\vec{e}_n$$

We can represent $\vec{v}$ as a coordinate vector

$$\vec{v} \rightarrow \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \end{pmatrix} \in \mathbb{C}^n (\text{ or } \mathbb{R}^n)$$

By selecting a basis, we obtain a "concrete" coordinate vector for an "abstract" vector

Orthonormal basis （正規直交基底）

When a vector space has an inner product,

$\vec{a}, \vec{b}$ is **orthogonal** （直交） if $(\vec{a}, \vec{b}) = 0$.

Orthonormal basis

A basis $\{\vec{e}_1, \vec{e}_2, \cdots \vec{e}_n\}$ is an orthonormal basis when

$$\|\vec{e}_i\| = 1 \qquad (i = 1, 2, \ldots, n)$$
$$(\vec{e}_i, \vec{e}_j) = 0 \qquad (i \neq j; i, j = 1, 2, \ldots, n)$$

*A basis can be transformed into an orthonormal basis.

cf. Gram-Schmidt orthonormalization

Example: wave vector

2 qbits: We can choose following four vectors as the (orthonormal) basis.

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

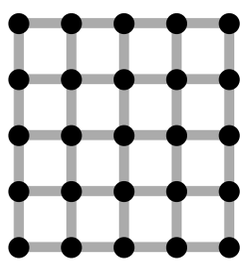
Simple notation: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$

$$|\Psi\rangle = \sum_{\alpha,\beta=0,1} C_{\alpha,\beta}|\alpha\beta\rangle$$

$C_{\alpha,\beta} = \langle\alpha\beta|\Psi\rangle$:complex number

$$C \in \mathbb{C}^4$$

Many qbits:



basis: $|m_1, m_2, \cdots, m_N\rangle = |00\cdots0\rangle, |00\cdots1\rangle, |01\cdots0\rangle, \ldots$

$$|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1,m_2,\cdots,m_N}|m_1, m_2, \cdots, m_N\rangle$$

$$T_{m_1,m_2,\cdots,m_N} = \langle m_1, m_2, \cdots, m_N|\Psi\rangle \quad T \in \mathbb{C}^{2^N}$$

Vector subspace (linear subspace)

Vector subspace（ベクトル部分空間）**:**

A subset $\mathbb{W}$ of a vector space $\mathbb{V}$ is a vector subspace of $\mathbb{V}$ when $\mathbb{W}$ satisfies the same axioms of vector space with $\mathbb{V}$.

The following conditions are necessary and sufficient.

$$\vec{a}, \vec{b} \in \mathbb{W} \Rightarrow \vec{a} + \vec{b} \in \mathbb{W}$$

$$\vec{a} \in \mathbb{W}, c \in \mathbb{C} \Rightarrow c\vec{a} \in \mathbb{W}$$

(In the case of complex vector space)

Spanned vector subspace

Spanned subspace:

For a subset $\mathbb{S}$ of a vector space $\mathbb{V}$, a set of linear combinations

$$\{c_1\vec{s}_1 + c_2\vec{s}_2 \cdots + c_k\vec{s}_k | c_i \in \mathbb{C}, \vec{s}_i \in \mathbb{S}\}$$

becomes a vector subspace of $\mathbb{V}$.

We often use

$$\mathrm{Span}\{\vec{s}_1, \vec{s}_2, \cdots, \vec{s}_k\}$$

to represents a vector subspace spanned by a set of vectors

$$\{\vec{s}_1, \vec{s}_2, \cdots, \vec{s}_k\}$$

Matrix and linear map

Matrix（行列）

Matrix: "Table" of (complex) numbers in a rectangular form

M × N matrix

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,N} \\ A_{21} & A_{22} & \cdots & A_{2,N} \\ \vdots & \vdots & & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{M,N} \end{pmatrix}$$

$A_{ij} \in \mathbb{C}(\text{ or } \mathbb{R})$

Product of matrices: $C = AB$

$$C_{ij} = \sum_{k=1}^{K} A_{ik} B_{kj}$$

$A : M \times K$

$B : K \times N$

$C : M \times N$

In general: $XY \neq YX$

*We also know addition, multiplication of scalar.

Identity matrix（単位行列）

Identity matrix:

N × N matrix
(Square matrix)

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Product:

$$IA = A \qquad A : N \times M$$

$$BI = B \qquad B : K \times N$$

* Element of the identity matrix: $I_{ij} = \delta_{ij}$ (Kronecker delta)

$$\delta_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$$

Transpose, complex conjugate and adjoint

Transpose:
(転置)

$$A^t \qquad (A^t)_{ij} = A_{ji}$$

Complex conjugate:
(複素共役)

$$A^* \qquad (A^*)_{ij} = A^*_{ij}$$

Adjoint:
(随伴)

or

Hermitian conjugate:
(エルミート共役)

$$A^\dagger = (A^t)^* = (A^*)^t$$

$$(A^\dagger)_{ij} = A^*_{ji}$$

("Dagger" is convention in physics)

Multiplication to coordinate vector

$$A : M \times N \qquad \vec{v} \in \mathbb{C}^N \qquad \vec{v}' \in \mathbb{C}^M$$

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,N} \\ A_{21} & A_{22} & \cdots & A_{2,N} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{M,N} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} v'_1 \\ v'_2 \\ \vdots \\ v'_M \end{pmatrix}$$

M × N matrix transforms a N-dimensional coordinate vector to a M-dimensional coordinate vector.

M × N matrix ⟷ **Linear map:** $\mathbb{C}^N \to \mathbb{C}^M$ （線形写像）

1 to 1

General linear map

Map: $f : \mathbb{V} \to \mathbb{V}'$

$$f(\vec{v}) = \vec{v}' \qquad (\vec{v} \in \mathbb{V}, \vec{v}' \in \mathbb{V}')$$

Linear map:

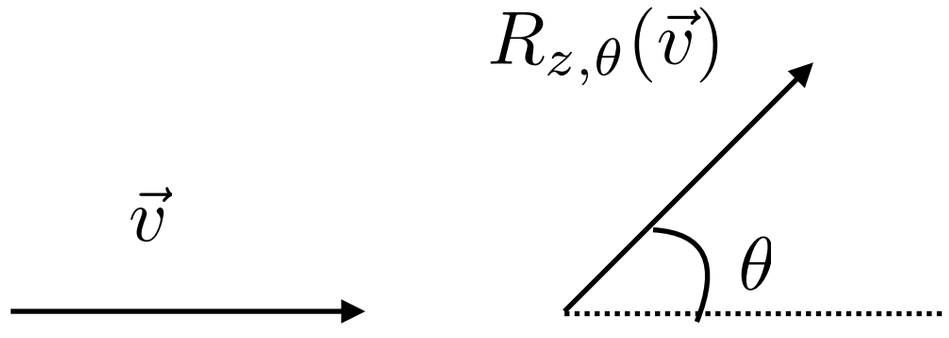
$$f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$$

$$f(c\vec{x}) = cf(\vec{x}) \qquad (\vec{x}, \vec{y} \in \mathbb{V}, c \in \mathbb{C})$$

Examples:

Rotation (e.g. θ rotation around z-axis)

$$R_{z,\theta} : \mathbb{C}^3 \to \mathbb{C}^3$$



Hamiltonian operator

$$\mathcal{H} : \mathbb{V} \to \mathbb{V}$$

$$|\Psi\rangle \qquad \mathcal{H}|\Psi\rangle$$

Matrix representation of linear map

By using a basis, we can represent a linear map in a matrix.

$$f : \mathbb{V} \to \mathbb{V}'$$

Vector space $\mathbb{V} : \dim \mathbb{V} = N$ → $\mathbb{V}' : \dim \mathbb{V}' = M$

Basis $\{\vec{e}_1, \vec{e}_2, \cdots, \vec{e}_N\}$ → $\{\vec{e'}_1, \vec{e'}_2, \cdots, \vec{e'}_M\}$

Transformation of basis vectors:

$$f(\vec{e}_j) = f_{1j}\vec{e'}_1 + f_{2j}\vec{e'}_2 + \cdots + f_{Mj}\vec{e'}_M$$

→ $f : \mathbb{V} \to \mathbb{V}'$ ⟷ $\begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1,N} \\ f_{21} & f_{22} & \cdots & f_{2,N} \\ \vdots & \vdots & & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{M,N} \end{pmatrix}$

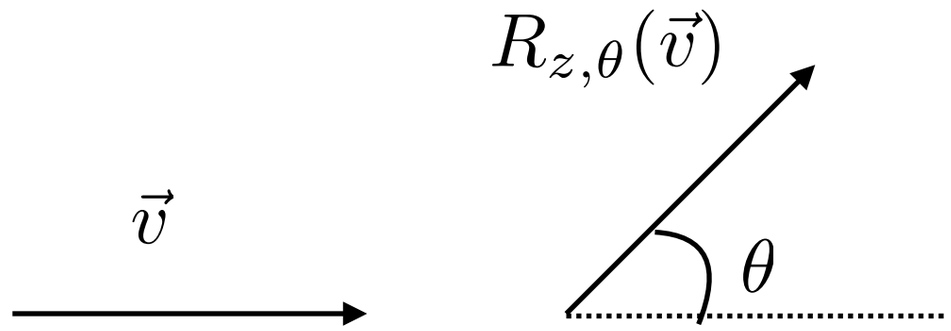
1 to 1
(if we fix basis)

Examples of matrix

Rotation (e.g. θ rotation around z-axis)

$$R_{z,\theta} : \mathbb{C}^3 \to \mathbb{C}^3$$

$$R_{z,\theta} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Hamiltonian operator

$$\mathcal{H} : \mathbb{V} \to \mathbb{V} \qquad \mathcal{H} \to \begin{pmatrix} H_{0,0;0,0} & H_{0,0;0,1} & H_{0,0;1,0} & H_{0,0;1,1} \\ H_{0,1;0,0} & H_{0,1;0,1} & H_{0,1;1,0} & H_{0,1;1,1} \\ H_{1,0;0,0} & H_{1,0;0,1} & H_{1,0;1,0} & H_{1,0;1,1} \\ H_{1,1;0,0} & H_{1,1;0,1} & H_{1,1;1,0} & H_{1,1;1,1} \end{pmatrix}$$

Matrix element: (行列要素) $H_{\alpha,\beta;\alpha',\beta'} \equiv \langle \alpha\beta | \mathcal{H} | \alpha'\beta' \rangle$

* In this notation, basis should be orthonormal.

Image of a map

$f : \mathbb{V} \to \mathbb{V}'$

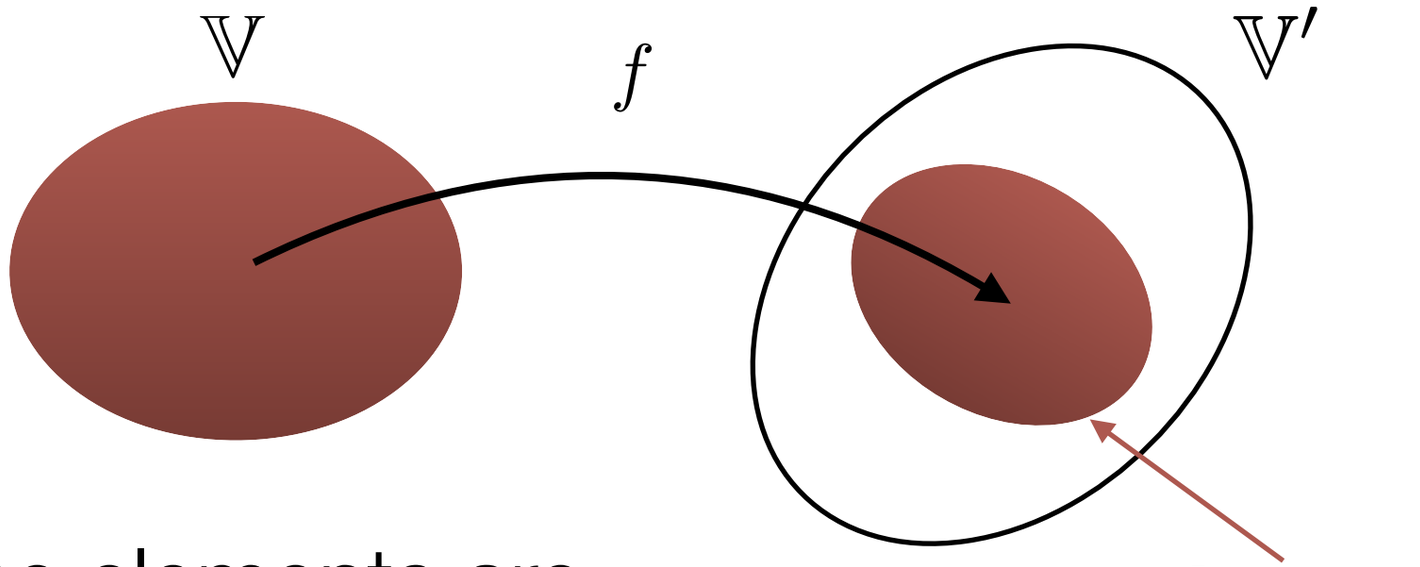


Image of f:
(像)

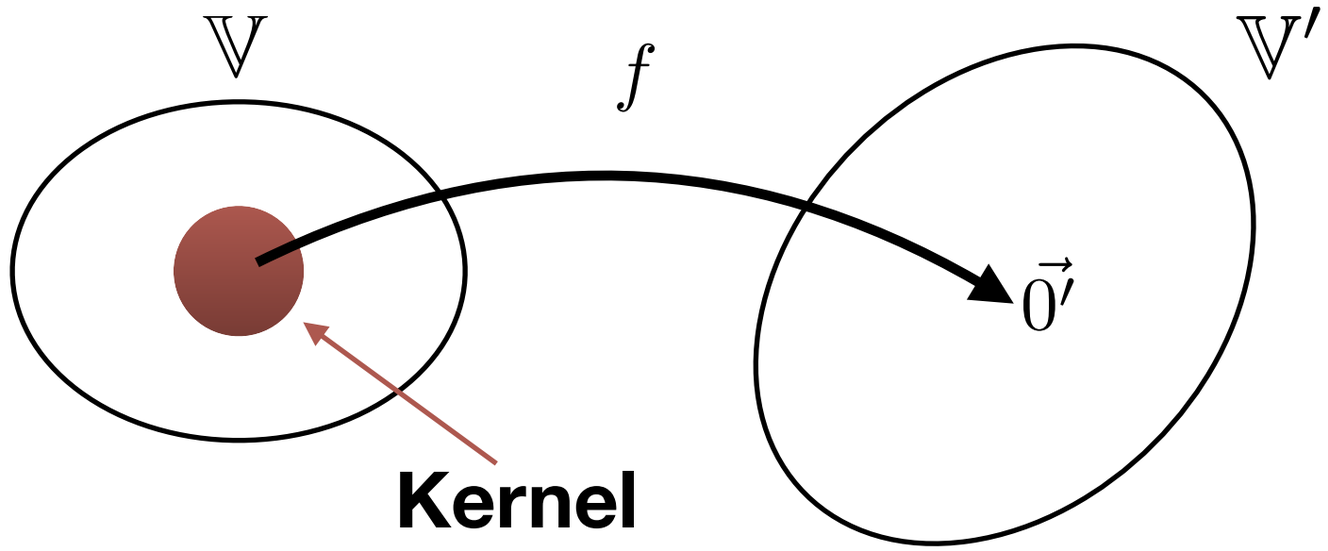
Vector subspace whose elements are mapped from $\mathbb{V}$ by f.

$$\mathrm{img}(f) = \{\vec{v}' | \vec{v} \in \mathbb{V}, \vec{v}' = f(\vec{v})\}$$

Kernel of a map

$f : \mathbb{V} \to \mathbb{V}'$

Kernel of f:
(核)



Vector subspace whose elements are mapped into zero vector by f .

$$\ker(f) = \{\vec{v} | \vec{v} \in \mathbb{V}, f(\vec{v}) = \vec{0}'\}$$

Theorem:

$$\dim(V) = \dim(\ker(f)) + \dim(\mathrm{img}(f))$$

Rank of matrix

Rank （ランク or 階数）of a matrix *A*:

$$\mathrm{rank}(A) \equiv \dim(\mathrm{img}(A))$$

Rank is identical with

Maximum # of linearly independent column vectors （列ベクトル）in *A*

Maximum # of linearly independent row vectors （行ベクトル）in *A*

$$\mathrm{rank}(A) \leq \min(M, N)$$

for a *N*×*M* matrix *A*.

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,N} \\ A_{21} & A_{22} & \cdots & A_{2,N} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{M,N} \end{pmatrix}$$

Regular matrix and its inverse matrix

A square matrix *A* is a **regular matrix** （正則）if a matrix *X* satisfying

$$AX = XA = I$$

exists. The matrix *X* is called inverse matrix（逆行列） of *A* and it is written as $X = A^{-1}$.

Properties:

A^{-1} is unique.

$$(A^{-1})^{-1} = A$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

A is a regular matrix $\Leftrightarrow$ $\mathrm{rank}(A) = N$

Can we consider an "inverse matrix" of a non-regular matrix (including a rectangular matrix) ?

Eigenvalue problems and diagonalization

Eigenvalue and Eigenvector

For a square matrix A

$$A\vec{v} = \lambda\vec{v}$$

$\vec{v} \neq \vec{0}$:eigenvector（固有ベクトル）

$\lambda \in \mathbb{C}$:eigenvalue（固有値）

Properties:

If $\vec{v}$ is an eigenvector, $c\vec{v}$ is also an eigenvector.

Eigenspace（固有空間）:

The set of eigenvectors corresponds an eigenvalue λ.

Eigenvectors corresponding to different eigenvalues are linearly independent.

Right and left eigenvectors

In general, left eigenvectors can be different from the right eigenvectors.

$$A\vec{v} = \lambda\vec{v}$$
$$(\vec{u}^*)^t A = \lambda(\vec{u}^*)^t$$

$\vec{v}$:Right eigenvector

$(\vec{u}^*)^t$:Left eigenvector

Properties:

Set of eigenvalues are identical between the right and the left eigenvectors.

A left eigenvector and a right eigenvector are orthogonal when they correspond to different eigenvalues.

$$\vec{u}_i^* \cdot \vec{v}_j = 0 \quad (\lambda_i \neq \lambda_j)$$

Diagonalization

Diagonalizaiton（対角化）：

$$A : N \times N \qquad P^{-1}AP = \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_N \end{pmatrix}$$

A can be diagonalized. ⟺ **necessary and sufficient** ⟺ *A* has *N* linearly independent eigenvectors.

$$\alpha_i = \lambda_i$$

$$P = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

$$(P^{-1})^t = (\vec{u}_1^*, \vec{u}_2^*, \cdots, \vec{u}_N^*)$$

Normalization: $\vec{u}_i^* \cdot \vec{v}_i = 1$

Meaning of diagonalization

General transform using a regular matrix: $P^{-1}AP$

It is a transform of the basis:

$$\{\vec{e}_1, \vec{e}_2, \cdots, \vec{e}_N\} \rightarrow \{P\vec{e}_1, P\vec{e}_2, \cdots, P\vec{e}_N\}$$

Diagonalization:

By using eigenvectors as a basis,
we can obtain a simple linear map
represented by a diagonal matrix.

$$A \rightarrow P^{-1}AP$$

* The determinant of *A* is invariant under this transformation:

$$\det(P^{-1}AP) = \det(P^{-1})\det(A)\det(P^{-1}) = \det(A)\det(P^{-1}P) = \det(A)$$

$$\Rightarrow \quad \det(A) = \prod_i^N \lambda_i$$

(This relation is true even if *A* cannot be diagonalized)

Unitary matrix

Unitary matrix（ユニタリ行列）：$U^\dagger = U^{-1}$

Real Orthogonal matrix（実直交行列）：$P^t = P^{-1}, (P_{ij} \in \mathbb{R})$

When we consider a unitary matrix as a set of vectors:

$$U = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

it is a orthonormal basis: $\vec{v}_i^* \cdot \vec{v}_j = \delta_{i,j}$

The linear map represented by a unitary matrix (**unitary transformation**) does not change

- the norm of a vector

$$\|U\vec{v}\| = \|\vec{v}\|$$

- "distance" between two vectors

$$\|U\vec{v}_1 - U\vec{v}_2\| = \|\vec{v}_1 - \vec{v}_2\|$$

Normal matrix

Normal matrix（正規行列）：$A^\dagger A = A A^\dagger$

We can always diagonalize it by a unitary matrix

$$U^\dagger = U^{-1}$$

as

$$U^\dagger A U = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix} \qquad \lambda_i \in \mathbb{C}$$

Its eigenvalues could be complex.

(even if *A* is a real matrix)

Hermitian matrix and its eigenvalue

Hermitian matrix（エルミート行列）：$A^\dagger = A$

Real symmetric matrix（実対称行列）：　$A^t = A, \quad (A_{ij} \in \mathbb{R})$

It is a special normal matrix. $A^\dagger A = A A^\dagger = A A$

Its eigenvalues are real.

We can always diagonalize it by a unitary matrix

$$U^\dagger A U = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix} \qquad \lambda_i \in \mathbb{R}$$

Hermitian (or real symmetric) matrices often appear in physics.

Generalization of diagonalization

- Eigenvalue problems and diagonalizations are defined for a square matrix.
- Even if A is a square matrix, it may not be diagonalized.
    - Is it possible to transform all square matrixes into diagonal forms by generalizing the diagonalization?
    - Is it possible to generalize it to a rectangular matrices?

Yes. **The singular value decomposition** (特異値分解) is an generalization of the diagonalization.

(We can also consider a decomposition of a tensor.)

Tentative lecture schedule

1日目

1.現代物理学における巨大なデータと情報圧縮

2.格子スピン模型の統計力学

3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似

5. テンソルネットワーク繰り込みによる情報圧縮

6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用

8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

特異値分解と低ランク近似

Outline

- Singular value decomposition (SVD)
- Low rank approximation
  - Low rank approximation by SVD
  - Low "rank" approximation for tensor
- Application of the low rank approximations to images

Singular value decomposition

Diagonalization

Diagonalizaiton（対角化）：

$$A : N \times N$$

(Square matrix)

$$P^{-1}AP = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix}$$

$$A\vec{v} = \lambda\vec{v}$$
$$(\vec{u}^*)^t A = \lambda(\vec{u}^*)^t$$

$$P = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$
$$(P^{-1})^t = (\vec{u}_1^*, \vec{u}_2^*, \cdots, \vec{u}_N^*)$$

- Eigenvalue problems and diagonalizations are defined for a square matrix.
- Even if A is a square matrix, it may not be diagonalized.
  - Normal or Hermitian matrices are always diagonalized by a unitary matrix

Spectral decomposition

(For a normal matrix A,)

Spectral decomposition
(スペクトル分解)

$$A = U \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix} U^\dagger$$

$$= \sum_{i=1}^{N} \lambda_i \vec{u}_i \vec{u}_i^\dagger$$

$$\left(= \sum_{i=1}^{N} \lambda_i |u_i\rangle\langle u_i| \right)$$

Note:

$$\vec{u}_i \vec{u}_i^\dagger = \begin{pmatrix} u_1 u_1^* & u_1 u_2^* & \cdots & u_1 u_N^* \\ u_2 u_1^* & u_2 u_2^* & \cdots & u_2 u_N^* \\ \vdots & \vdots & \cdots & \vdots \\ u_N u_1^* & u_N u_2^* & \cdots & u_N u_N^* \end{pmatrix}$$

Matrix decomposition into a sum of projectors onto its eigen subspaces.

Projector:

$$P^2 = P$$

Singular value decomposition (SVD)

Singular value decomposition
(特異値分解)

$$A : M \times N$$

$$A_{ij} \in \boldsymbol{C}$$

$$A = U \Sigma V^{\dagger}$$

$U : M \times M$ **Unitary**

$V : N \times N$ **Unitary**

$$\Sigma = \begin{pmatrix} \Sigma_{r \times r} & 0_{r \times (N-r)} \\ 0_{(M-r) \times r} & 0_{(M-r) \times N-r} \end{pmatrix}$$

$$\Sigma_{r \times r} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix}$$

Diagonal matrix with
non-negative real elements

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

Singular values

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

$A : M \times N \Rightarrow A^\dagger A : N \times N$

$*A^\dagger A$ is a Hermitian matrix.

$$(A^\dagger A)^\dagger = A^\dagger A \Rightarrow$$

It can be diagonalized by
a unitary matrix V .

$$V^\dagger (A^\dagger A) V = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$$

$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

$\vec{v}_i$: eigenvector

* $A^\dagger A$ is a positive semi-definite matrix.
(半正定値、準正定値)

$$\vec{x}^* \cdot (A^\dagger A \vec{x}) = \|A\vec{x}\|^2 \geq 0 \Leftrightarrow$$

Its eigenvalues are
non-negative

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$$

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

$$V^\dagger(A^\dagger A)V = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$$
$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

$\Leftrightarrow$

$$(A\vec{v}_i)^* \cdot (A\vec{v}_j) = \lambda_i \delta_{ij}$$
$$(\|A\vec{v}_i\|^2 = \lambda_i)$$

Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_N$

(There are r positive eigenvalues.)

Make new orthonormal basis $U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$ in $\mathbb{C}^M$

For $(i = 1, 2, \ldots, r)$ $\quad \sigma_i = \sqrt{\lambda_i}\,, \vec{u}_i = \dfrac{1}{\sigma_i} A\vec{v}$

For $(i = r+1, \ldots, M)$ Any orthonormal basis orthogonal to $\vec{u}_i \quad (i = 1, 2, \ldots, r)$

$$\vec{u}_i^* \cdot (A\vec{v}_j) = \sigma_i \delta_{ij} \quad (i = 1, \ldots, M; j = 1, \ldots, N)$$

(For simplicity, we set $\sigma_i = 0$ for $i > r$.)

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

   We can perform same "proof" by using $AA^\dagger$.

   $U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$ is the unitary matrix which diagonalize $AA^\dagger$ as

$$U^\dagger (AA^\dagger) U = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_r, \underbrace{0, \ldots, 0}_{M-r}\}$$

In summary,

- A matrix A can be decomposed as SVD: $A = U\Sigma V^\dagger$
- Singular values are related to the eigenvalues of $A^\dagger A$ and $AA^\dagger$ as

$$\sigma_i = \sqrt{\lambda_i} \ .$$

- V and U are eigenvectors of $A^\dagger A$ and $AA^\dagger$,respectively.

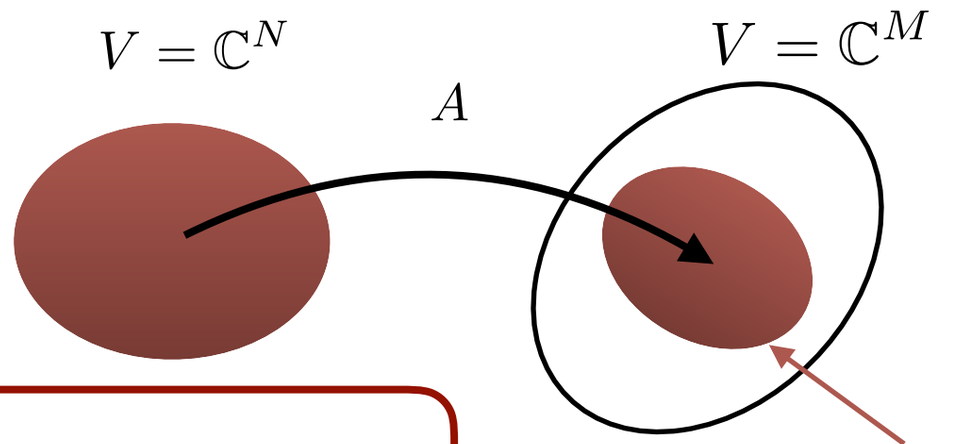
Properties of SVD 2

$$A = U\Sigma V^\dagger$$

2. # of positive singular values is identical with the rank.

$$A : M \times N \Rightarrow A : \mathbb{C}^N \to \mathbb{C}^M$$

$$\mathrm{rank}(A) \equiv \dim(\mathrm{img}(A))$$



Remember

The orthonormal basis $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N\}$ satisfies

$$(A\vec{v}_i)^* \cdot (A\vec{v}_j) = \lambda_i \delta_{ij}$$

Here, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_N$ and $\sigma_i = \sqrt{\lambda_i}$

$$\forall \vec{x} \in \mathbb{C}^N, \vec{x} = \sum_{i=1}^{N} C_i \vec{v}_i \Rightarrow A\vec{x} = \sum_{i=1}^{N} C_i (A\vec{v}_i) = \sum_{i=1}^{r} C_i (A\vec{v}_i)$$

$$\Rightarrow \mathrm{img}(A) = \mathrm{Span}\{A\vec{v}_1, A\vec{v}_2, \ldots, A\vec{v}_r\}$$

$$\Rightarrow \dim(\mathrm{img}(A)) = r$$ **=# of positive singular values**

Properties of SVD 3

$$A = U\Sigma V^\dagger$$

3. Singular vectors

$A : M \times N \qquad U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)\,, V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$

For $i = 1, 2, \ldots, r$

$$A\vec{v}_i = \sigma_i \vec{u}_i \ , \ A^\dagger \vec{u}_i = \sigma_i \vec{v}_i$$

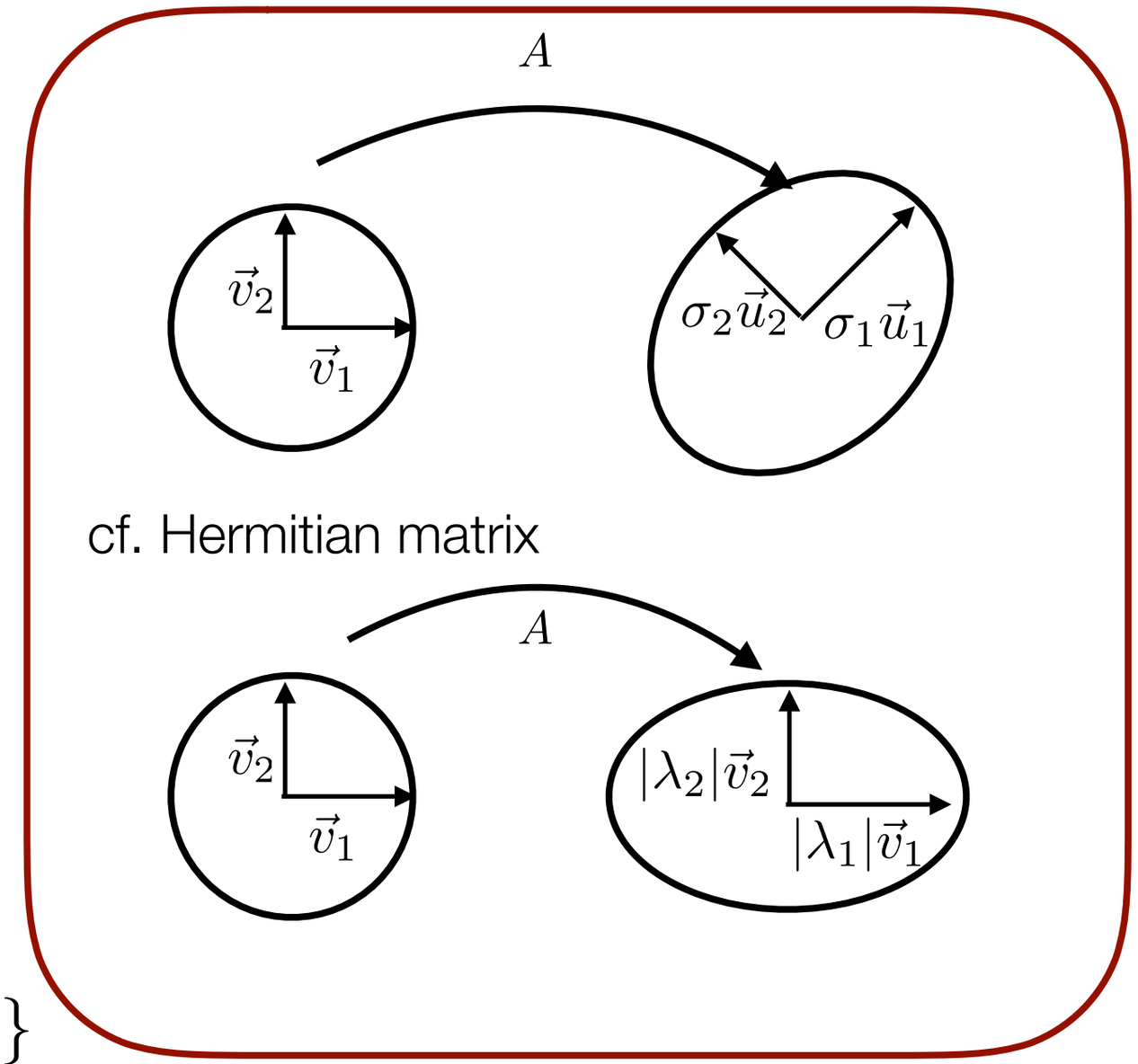
$\vec{v}_i$: right singular vector

$\vec{u}_i$: left singular vector

Relation to image and kernel:

$$\mathrm{img}(A) = \mathrm{Span}\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r\}$$
$$\ker(A) = \mathrm{Span}\{\vec{v}_{r+1}, \vec{v}_{r+2}, \ldots, \vec{v}_N\}$$

$$\mathrm{img}(A^\dagger) = \mathrm{Span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r\}$$
$$\ker(A^\dagger) = \mathrm{Span}\{\vec{u}_{r+1}, \vec{u}_{r+2}, \ldots, \vec{u}_M\}$$



Properties of SVD 4 (optional)

$$A = U\Sigma V^\dagger$$

4. Min-max theorem (Courant-Fischer theorem)

$A : N \times N$, Hermitian matrix

Suppose its eigenvalues are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$.

$$\lambda_k = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq k-1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \vec{x}^* \cdot A\vec{x}$$

$$\boldsymbol{S}^\perp = \{\vec{x} : \vec{x}^* \cdot \vec{y} = 0, \vec{y} \in \boldsymbol{S}\}$$

Orthonormal complement（直交補空間）

We can prove this by considering vector subspace spanned by eigenvectors. (see references)

Intuitive examples:

Maximum appears for the eigenvector.

$$A\vec{u}_i = \lambda_i \vec{u}_i$$

$$\lambda_1 = \max_{\vec{x}\in\boldsymbol{C}^N;\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) \quad \Rightarrow \quad \vec{x} = \vec{u}_1$$

$$\lambda_2 = \max_{\vec{x}\in\boldsymbol{C}^N;\vec{x}\perp\vec{u}_1,\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq 1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) \quad \Rightarrow \quad \vec{x} = \vec{u}_2$$

Properties of SVD 4 (optional)

$$A = U\Sigma V^\dagger$$

4. Min-max theorem (Courant-Fischer theorem)

$A : M \times N$

Suppose its singular values are $\sigma_1 \geq \sigma_2 \geq \cdots$.

$$\sigma_k = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq k-1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \|A\vec{x}\|$$

By setting k=1,

$$\sigma_1 = \max_{\vec{x}\in\boldsymbol{C}^N,\|\vec{x}\|=1} \|A\vec{x}\|$$

which means

$$\|A\vec{x}\| \leq \sigma_1 \|\vec{x}\|$$

for $\vec{x} \in \boldsymbol{C}^N$

We can easily prove this by using

$A^\dagger A$: Hermitian

$$A^\dagger A \vec{v}_i = \lambda_i$$

$$\sigma_i = \sqrt{\lambda_i}$$

Properties of SVD 5 (optional) $A = U\Sigma V^\dagger$

5. Singular values for multiplication and addition

$\sigma_i(A)$: singular value of matrix A

(for $i > \mathrm{rank}(A)$, we set $\sigma_i = 0$)

*Following properties can be proven by using min-max theorem.

Multiplication: $A : M \times L, B : L \times N$

$$\sigma_k(AB) \leq \sigma_1(A)\sigma_k(B) \qquad (k = 1, 2, \dots)$$

$$(\sigma_k(AB) \leq \sigma_k(A)\sigma_1(B))$$

$$\Rightarrow \mathrm{rank}(AB) \leq \min(\mathrm{rank}(A), \mathrm{rank}(B))$$

Addition: $A, B : M \times N$

$$\sigma_{k+j-1}(A+B) \leq \sigma_k(A) + \sigma_j(B) \qquad (k, j = 1, 2, \dots)$$

$$(\sigma_{k+j-1}(A+B) \leq \sigma_j(A) + \sigma_k(B))$$

$$\Rightarrow \text{If } \mathrm{rank}(B) \leq r \text{ ,}$$

$$\sigma_{k+r}(A+B) \leq \sigma_k(A)$$

Libraries for SVD

There are **LAPACK** routines for SVD.

DGESDD, ZGESDD

DGESVD, ZGESVD

(For dense matrices)

*Linear Algebra PACKage

At *netlib.org* (reference implementations)

+

A lot of vender implementations

- Intel MKL
- Apple Accelerate Framework
- Fujitsu SSLII
- ...

numpy and **scipy** modules in python have routines for SVD.

numpy.linalg.svd

scipy.linalg.svd

(For dense matrices)

scipy.sparse.linalg.svds

(For sparse matrices or
calculation of partial singular values)

Computation cost

For a $M \times N$ matrix ($M \leqq N$):

Full SVD: $\mathrm{O}(NM^2)$

Partial SVD: $\mathrm{O}(NMk)$

k : # of singular values
to be calculated

Low rank approximation

Amount of data in SVD representation

$A : M \times N$

$$A = U\Sigma V^\dagger = U \begin{pmatrix} \Sigma_{r\times r} & 0_{r\times(N-r)} \\ 0_{(M-r)\times r} & 0_{(M-r)\times N-r} \end{pmatrix} V^\dagger$$

neglect zero singular values →

$$= \bar{U}\Sigma_{r\times r}\bar{V}^\dagger$$

$$\bar{U} : M \times r, \bar{V}^\dagger : r \times N$$

If rank(A) is much smaller than M and N,

$$r \ll M, N$$

we can reduce the data to represent A.

(At this stage, no data loss)

$$U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$$
$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

↓

$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

Low rank approximation

Low rank approximation（低ランク近似）

Find an approximate matrix

$$A \simeq \tilde{A}$$

with lower rank:

$$\mathrm{rank}(\tilde{A}) > \mathrm{rank}(A)$$

Through the low rank approximation,
we can reduce amount of the data.

An example of information compressions.

Notice! In order to quantify accuracy of the approximation,
we need a measure of distance between matrices.

Low rank approximation by SVD

Consider a matrix obtained by neglecting smaller singular values

$$A = \bar{U}\Sigma_{r\times r}\bar{V}^\dagger \quad \Rightarrow \quad \tilde{A} = \tilde{U}\Sigma_{k\times k}\tilde{V}^\dagger \qquad (k < r)$$

$$\Sigma_{r\times r} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$
$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

$$\mathrm{rank}(A) = r$$

$$\Sigma_{k\times k} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$$
$$\tilde{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)$$
$$\tilde{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k)$$

Keep the largest k singular values
(and corresponding singular vectors).

$$\mathrm{rank}(\tilde{A}) = k < r$$

This approximation is one of the low rank approximation.

* For this approximation, we need $\mathrm{O}(MNk)$ calculations for SVD of a $M\times N$ matrix.

Norm of matrices $\|A\|$

There are two popular norms:

(1) **Frobenius norm** （フロベニウス　ノルム）

$$\|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2} = \sqrt{\mathrm{Tr}(A^\dagger A)}$$

*Trace（対角和）

$$\mathrm{Tr}(X) = \sum_i X_{ii}$$

(2) **Operator norm**（作用素ノルム）

$$\|A\|_O = \inf\{c \geq 0; \|A\vec{x}\| \leq c\|\vec{x}\|\}$$
$$= \sigma_1(A)$$

*inf =infimum（下限）

*We define the norm for a vector as

$$\|\vec{x}\| = \sqrt{\sum_i |x_i|^2}$$

By using these norms, we define the distance between matrices:

$$\|A - \tilde{A}\|$$

Accuracy of low rank approximation by SVD

Theorem

For $A : M \times N$

$$\min\{\|A - B\|_F : \mathrm{rank}(B) = k\} = \sqrt{\sum_{i=k+1}^{\min(N,M)} \sigma_i^2}$$

$$\min\{\|A - B\|_O : \mathrm{rank}(B) = k\} = \sigma_{k+1}$$

Because the k-rank approximation by SVD gives

$$\|A - \tilde{A}\|_F = \sqrt{\sum_{i=k+1}^{\min(N,M)} \sigma_i^2}, \qquad \|A - \tilde{A}\|_O = \sigma_{k+1}$$

it is an "optimal" approximation with rank k.

Short proof of the theorem: Frobenius norm (optional)

*This proof is based on
"システム制御のための数学（１）"太田快人 著

From the inequality of singular values for matrix addition (property 5),

for j=1,... ,$\min(M,N)$ $\quad (\mathrm{rank}(B) = k)$

$$\sigma_{j+k}(A) = \sigma_{j+k}((A-B)+B) \leq \sigma_j(A-B)$$

Property 5

By taking a square and summing up them

$$\sum_{i=k+1}^{\min(M,N)} \sigma_i^2(A) \leq \sum_{j=1}^{\min(M,N)} \sigma_j^2(A-B) = \|A-B\|_F^2$$

*Note $\sigma_j(A) = 0 \quad (j > \mathrm{rank}(A))$

Short proof of the theorem: operator norm (optional)

*This proof is based on
"システム制御のための数学（１）"太田快人 著

From the min-max theorem of singular values (property 4),

$$(\mathrm{rank}(B) = k)$$

$$\sigma_{k+1}(A) \leq \max_{\vec{x}\in\ker(B),\|\vec{x}\|=1} \|A\vec{x}\| = \max_{\vec{x}\in\ker(B),\|\vec{x}\|=1} \|(A-B)\vec{x}\|$$

Property4 with
$\boldsymbol{S} = \mathrm{img}(B^\dagger)$
$\boldsymbol{S}^\perp = \ker(B)$

$B\vec{x} = 0 \quad (\vec{x} \in \ker(B))$

$$\leq \max_{\|\vec{x}\|=1} \|(A-B)\vec{x}\| = \|A-B\|_O$$

Expand the vector space

Definition of the operator norm

Relation to **principal component analysis** （主成分分析）

Data set $\{X_{ij}\}$: X: $N \times M$ matrix

i = index for data, j = data type (coordinates, momentum, ...)

* Suppose the mean of data is 0: $\sum_i X_{ij} = 0$

Covariance matrix（共分散行列）: $C = X^T X$

Principal component analysis (PCA):

Data compression through the spectrum decomposition of C.

$C = V \Lambda V^T$ $\quad$ Λ: diagonal matrix, $\Lambda_{ii} = \lambda_i \geq 0$

$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$

$\vec{v}_i$ corresponding to large λ_i contains important information.

By construction, λ and V are related to SVD of X!

$X = U \Sigma V^T$ $\quad , \sigma_i = \sqrt{\lambda_i}$

PCA can be regarded as the low rank approximation X.

Generalization to tensors

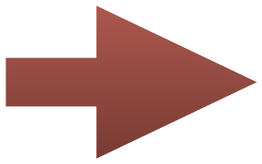
Scalar, Vector, Matrix, Tensor,...

Scalar: c Number

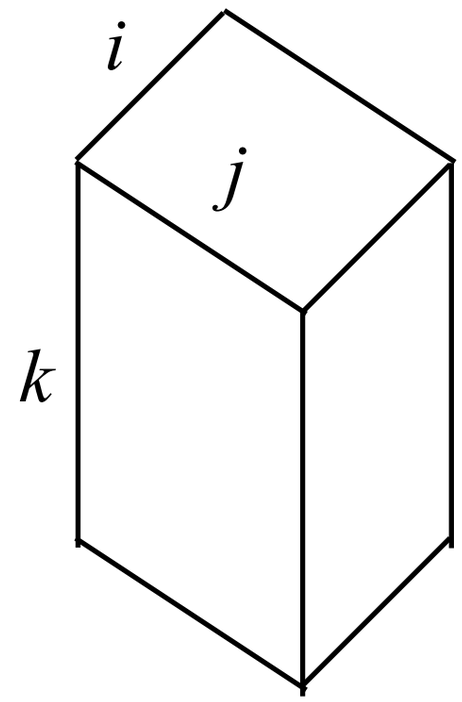
Vector: v_i One dimensional array of numbers

Matrix: M_{ij} Two dimensional array of numbers

Tensor: $T_{ijk\ldots}$ Higher dimensional array of numbers



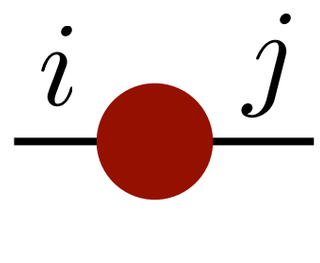
Scalar: 0-dim. tensor
Vector: 1-dim. tensor
Matrix: 2-dim. tensor



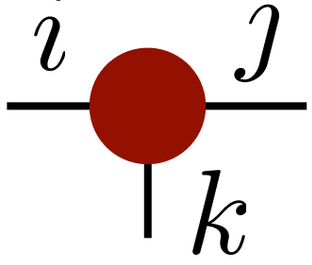
Graphical representations for tensor network

- Vector $\vec{v} : v_i$

- Matrix $M : M_{i,j}$

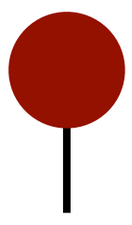


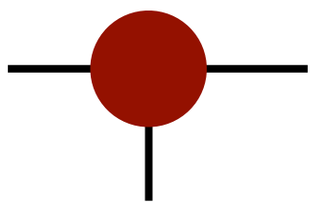
- Tensor $T : T_{i,j,k}$



*** n-rank tensor = n-leg object**

When indices are not presented in a graph, it represent a tensor itself.

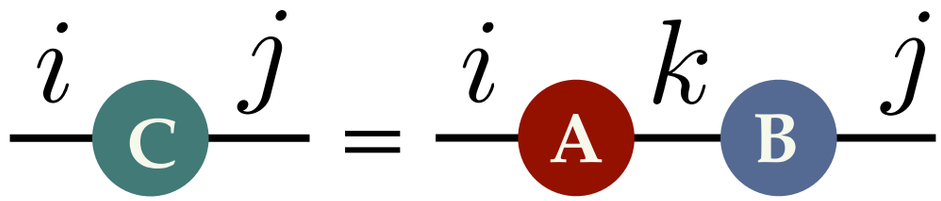
$\vec{v} =$ 

$T =$ 

Graphical representations for tensor network

Matrix product

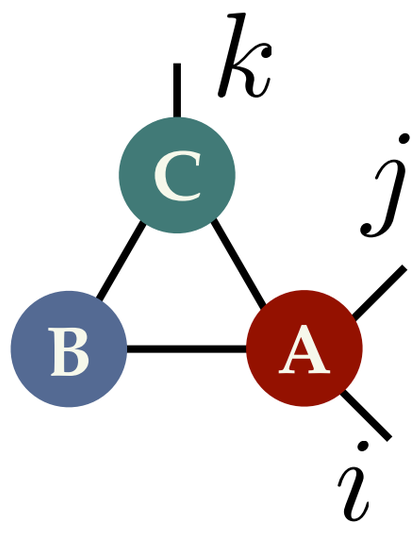
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$



$$C = AB$$

Generalization to tensors

$$\sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$



Contraction of a network = Calculation of a lot of multiplications

（縮約）

Low rank approximation: generalization to tensor

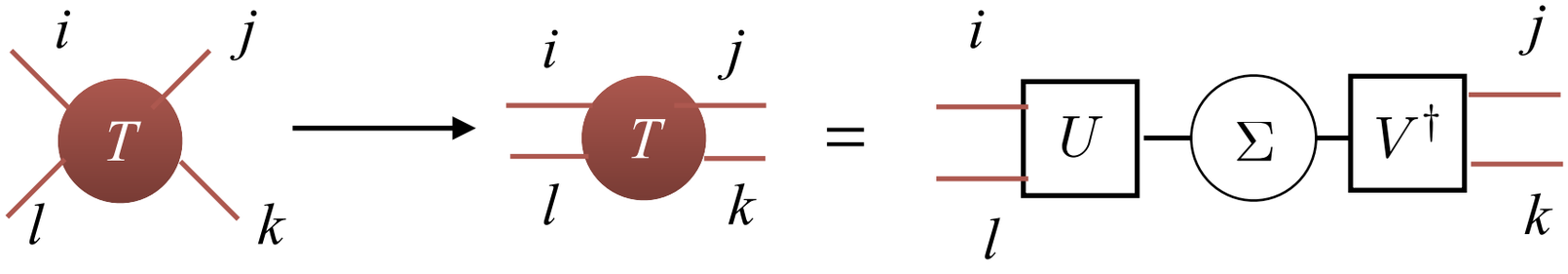
Tensor: $T_{ijk\ldots}$

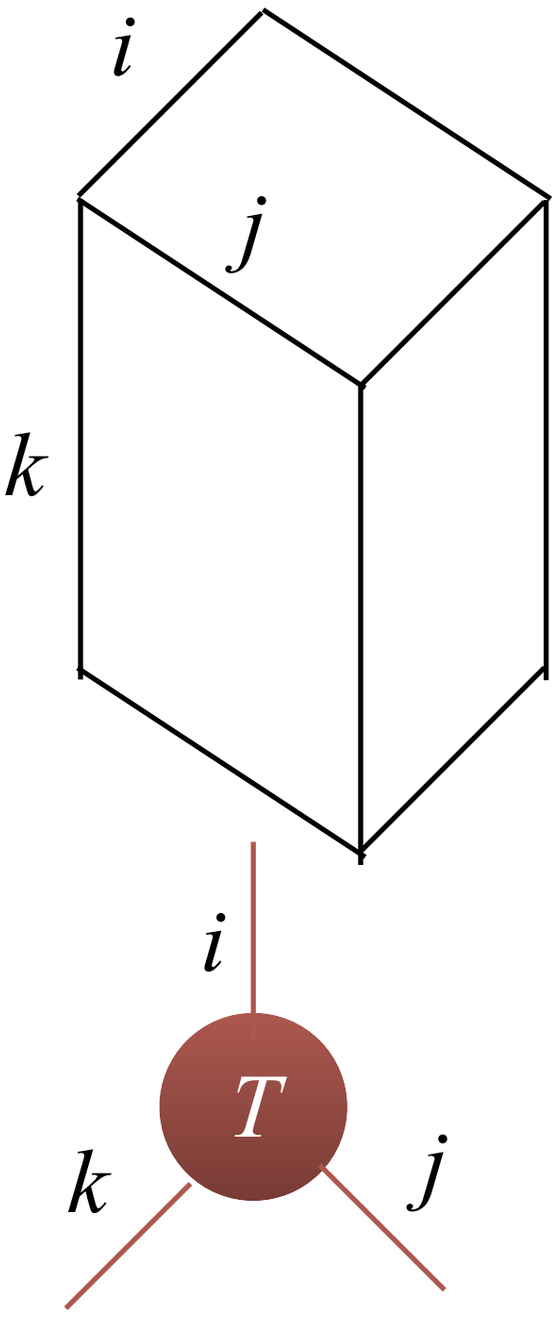
Naive application of SVD:

Make a matrix by dividing indices into two parts.

$$T_{ijkl} \to T_{(il),(jk)}$$

Then apply SVD (and low rank approximation).



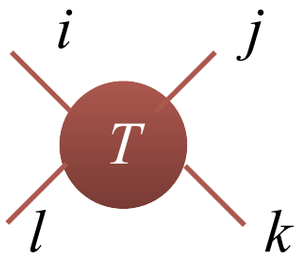


Note: The result depends on the initial mapping to a matrix.

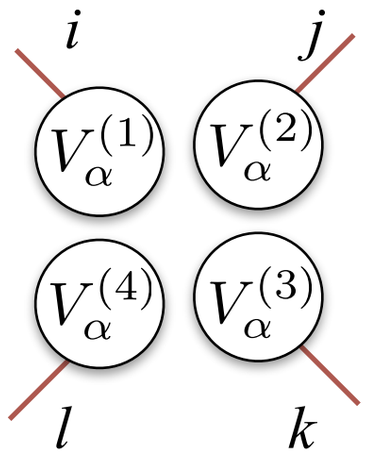
CP decomposition

Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

CP (Canonical Polyadic) decomposition: Hitchcock (1927)



$$= \sum_{\alpha=1}^{r}$$



$$T_{ijkl} = \sum_{\alpha=1}^{r} (V_\alpha^{(1)})_i (V_\alpha^{(2)})_j (V_\alpha^{(3)})_k (V_\alpha^{(4)})_l$$

min (r) = tensor rank

*Determining tensor rank is NP-hard problem.

Low "rank" approximation

$$T_{ijkl} \approx \sum_{\alpha=1}^{r'} (\tilde{V}_\alpha^{(1)})_i (\tilde{V}_\alpha^{(2)})_j (\tilde{V}_\alpha^{(3)})_k (\tilde{V}_\alpha^{(4)})_l$$

$r' < r$

rank-r' approximation

Tucker decomposition: generalization of SVD

Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

Tucker decomposition:

(Tucker (1963))

$U^{(i)}$: Factor matrix (usually unitary)

S :Core tensor

$$T_{ijkl} = \sum_{i'=1}^{I}\sum_{j'=1}^{J}\sum_{k'=1}^{K}\sum_{l'=1}^{L} S_{i'j'k'l'} U^{(1)}_{ii'} U^{(2)}_{jj'} U^{(3)}_{kk'} U^{(4)}_{ll'}$$

*If S is "diagonal", Tucker decomposition becomes CP decomposition.

Low "rank" approximation

$$T_{ijkl} = \sum_{i'=1}^{I'}\sum_{j'=1}^{J'}\sum_{k'=1}^{K'}\sum_{l'=1}^{L'} \tilde{S}_{i'j'k'l'} \tilde{U}^{(1)}_{ii'} \tilde{U}^{(2)}_{jj'} \tilde{U}^{(3)}_{kk'} \tilde{U}^{(4)}_{ll'}$$

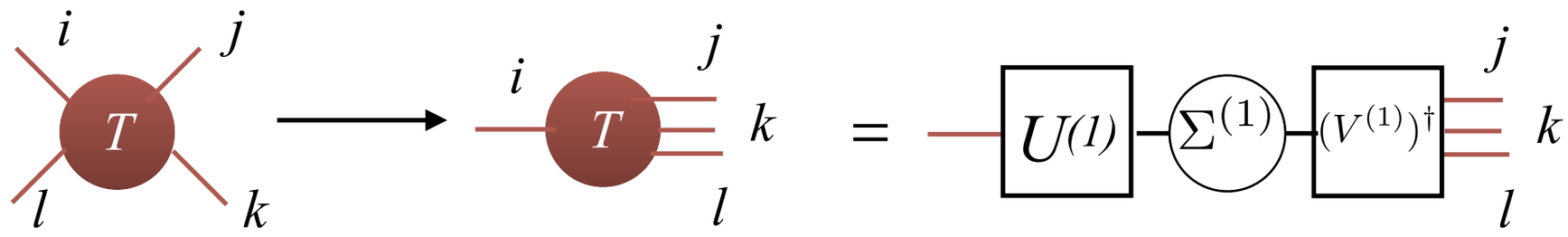
$I' < I, \quad J' < J, \quad K' < K, \quad L' < L$

rank-(I',J', K', L') approximation

Higher order SVD (HOSVD)

L. De Lathauwer et al, SIAM J. Matrix Anal. & Appl., **21**, 1253 (2000)

Define a factor matrix from matrix SVD:



Core tensor is calculated as

$$S_{i'j'k'l'} \equiv \sum_{ijkl} T_{ijkl} (U^{(1)})^\dagger_{i'i} (U^{(2)})^\dagger_{j'j} (U^{(3)})^\dagger_{k'k} (U^{(4)})^\dagger_{l'l}$$

Properties of the core tensor

$$S^*_{:,i_n=\alpha,:,:} \cdot S_{:,i_n=\beta,:,:} = \begin{cases} 0 & (\alpha \neq \beta) \\ (\sigma^{(n)}_\alpha)^2 & (\alpha = \beta) \end{cases}$$

Dot product

$$A \cdot B \equiv \sum_{i,j,k,l} A_{ijkl} B_{ijkl}$$

Generalization of the diagonal matrix Σ in matrix SVD.

* Low-rank approximation based on HOSVD is not optimal.

Application of low rank approximation

Sample codes will be uploaded on the website

SVD_sample.zip

(python3 (or python2.7) + numpy + PIL)

(python and jupyter notebook codes)

Image compression: grayscale image

Image: 1024 × 768 pixels

768 x1024 matrix A

$$\mathrm{rank}(A) = 768$$

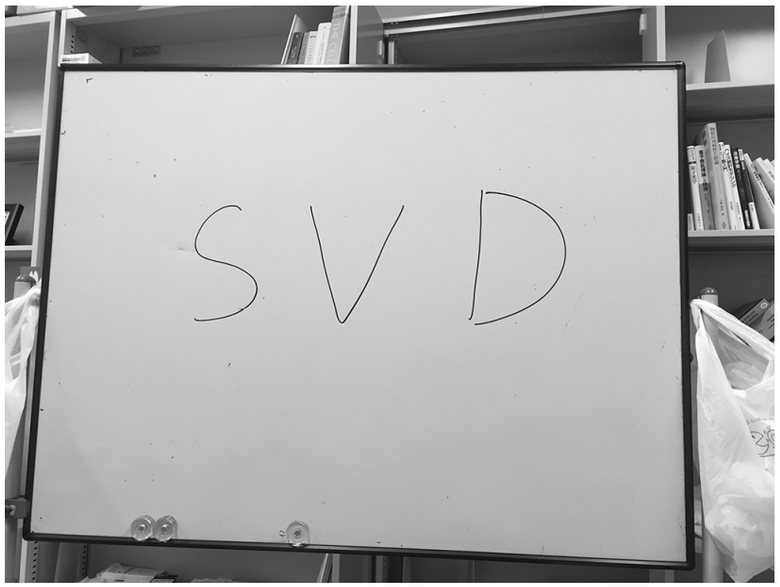
Amount of data=786,432

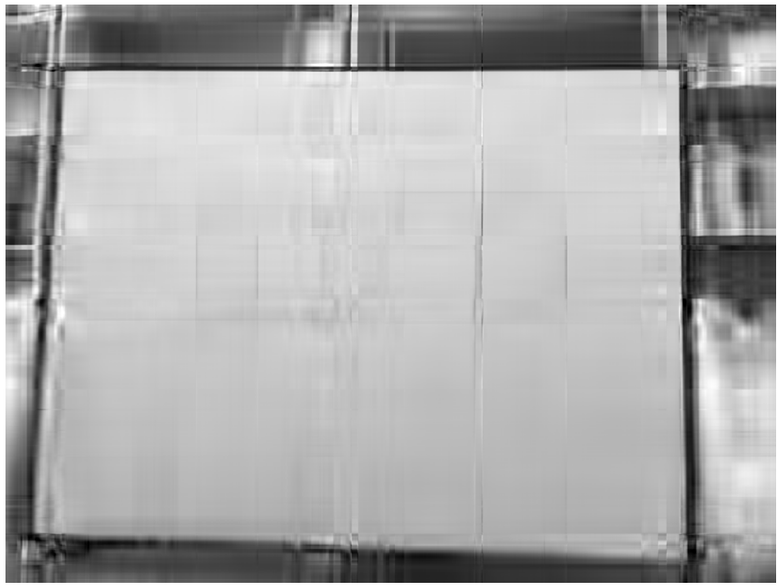
Perform SVD of A: $A = U\Sigma V^{\dagger}$

$\mathrm{rank}(\chi)$ approximation

Amount of data=(768 +1024 + 1)×χ

Image compression: grayscale image





Rank: $\chi = 768$ $\chi = 100$ $\chi = 10$

Data: **786,432** **179,300** **179,30**

(Original)

Image compression: color image

Image: 1024 × 768 pixels

768 x1024x3 tensor T

Amount of data=2,359,296

* Sub matrices for RGB colors

$$R_{ij} = T_{ij1}, \quad G_{ij} = T_{ij2}, \quad B_{ij} = T_{ij3}$$

Two image compressions:

Perform SVD for R, G, B → $\mathrm{rank}(\chi)$ approximation for RGB matrices

Amount of data=3×(768 +1024 + 1)×χ

Perform HOSVD for T → $\mathrm{rank}-(\chi', \chi', 3)$ approximation

Amount of data= (768 +1024 + 3χ)×χ

Image compression: color image1

Original

Data: **2,359,296**

~10%
Compression

SVD

$\chi = 50$

268,950

HOSVD

$\chi' = 100$

209,200

Image compression: color image2

Original

Data: **2,359,296**

~10%
Compression

SVD

$\chi = 50$

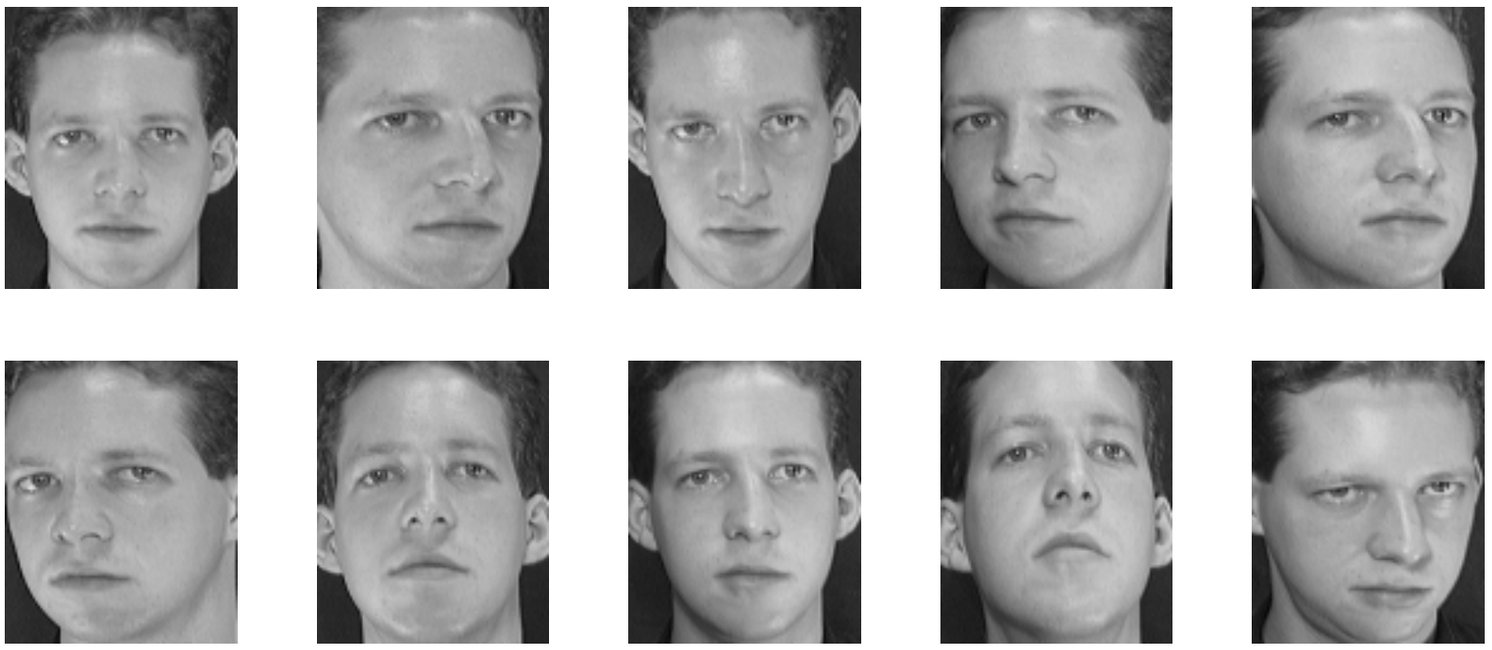
268,950

HOSVD

$\chi' = 100$

209,200

Image compression: multi images



92×122 pixels 10 images

92 x122 x 10 tensor T

Amount of data=112,240

by HOSVD

$\mathrm{rank}-(\chi, \chi, \chi')$ approximation

Amount of data=
$(92 + 122) \times \chi + 10 \times \chi' + \chi^2 \chi'$

Images were taken from ORL Database of Faces, AT&T Laboratories Cambridge

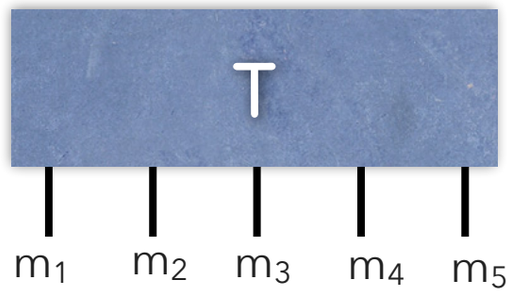
Image compression: multi images

		Data
Original		112,240
$\chi = 30$ $\chi' = 10$		15,520
$\chi = 30$ $\chi' = 9$		14,610
$\chi = 30$ $\chi' = 5$		10,970

SVD of wave function?

Wave function: $|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1,m_2,\cdots,m_N} |m_1, m_2, \cdots, m_N\rangle$

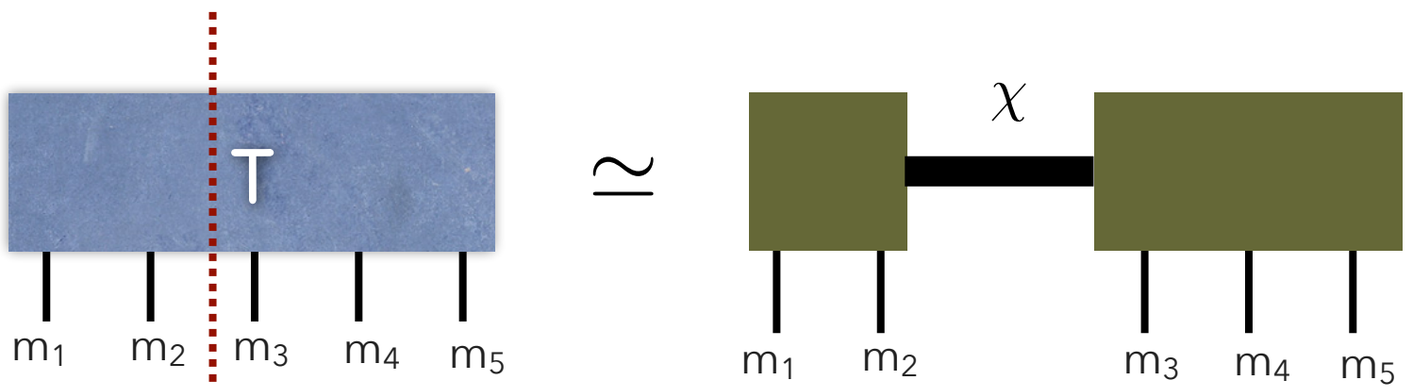
$T_{m_1,m_2,\cdots,m_N}$:N-leg tensor (or Vector)



We can consider it as a matrix by making two groups:

$T_{\{m_1,m_2,\cdots,m_M\},\{m_{M+1},\cdots,m_N\}}$

➡ We can perform the low rank approximation of T.



What does it mean?

References:

- 齋藤正彦、「線形代数入門」東京大学出版会
- 太田快人、「システム制御のための数学（１）ー線形代数編ー」、コロナ社
- T. G. Kolda et al, SIAM Review **51**, 455 (2006).

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

テンソルネットワーク繰り込みによる情報圧縮
(古典スピン模型の実空間繰り込み群)

Outline

- Tensor network representation of a scalar
  - Partition functions in statistical mechanics
- Tensor network renormalization
  - Tensor Renormalization Group (TRG) in two dimension
  - Generalization to higher dimensions
- Tensor network renormalization around critical point
  - Fixed point of TRG: Corner double line tensors
- Report problems

Tensor network representation of a scalar

イジング模型の分配関数

統計力学の処方箋：

分配関数を計算したい

$$Z = \sum_{\{S_i = \pm 1\}} e^{\beta J \sum_{\langle i,j \rangle} S_i S_j} \qquad \mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

しかし、和 $\sum_{\{S_i = \pm 1\}}$ は、スピンがN個の時、 2^N

で指数的に大きい！（N=100でも、10^{30}の項がある！）

→ 定義通りに計算することは困難

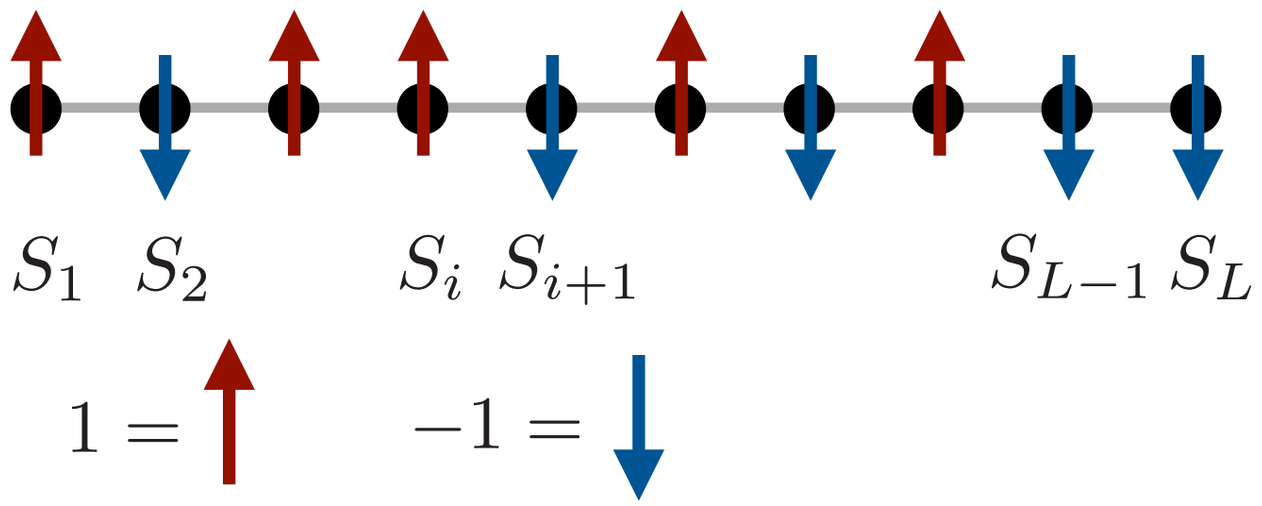
→ 計算科学によるアプローチ

転送行列

例：１次元イジング模型

$$\mathcal{H} = -J \sum_{i=1}^{L-1} S_i S_{i+1}$$

$$S_i = 1, -1$$



分配関数

$$\begin{aligned} Z &= \sum_{\{S_i=\pm 1\}} e^{\beta J \sum_i S_i S_{i+1}} \\ &= \sum_{\{S_i=\pm 1\}} \prod_{i=1}^{L-1} e^{\beta J S_i S_{i+1}} \\ &= \sum_{S_1=\pm 1, S_L=\pm 1} \left(T^{L-1}\right)_{S_1, S_L} \end{aligned}$$

転送行列

$$T = \begin{pmatrix} e^{\beta J} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J} \end{pmatrix} \begin{matrix} +1 \\ -1 \end{matrix}$$

(+1, -1)

$$T_{S_i, S_{i+1}} = e^{\beta J S_i S_{i+1}}$$

分配関数は転送行列の積でかける

転送行列の対角化

転送行列

$$T = \begin{pmatrix} e^{\beta J} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J} \end{pmatrix}$$

転送行列は実対称行列

➡ 固有値は実で、直行行列で対角化可能

$$T = P^t \begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix} P$$

$$\lambda_+ = 2\cosh\beta J$$

$$\lambda_- = 2\sinh\beta J$$

$$|\lambda_+| > |\lambda_-|$$

$$P^t P = P P^t = I$$

分配関数

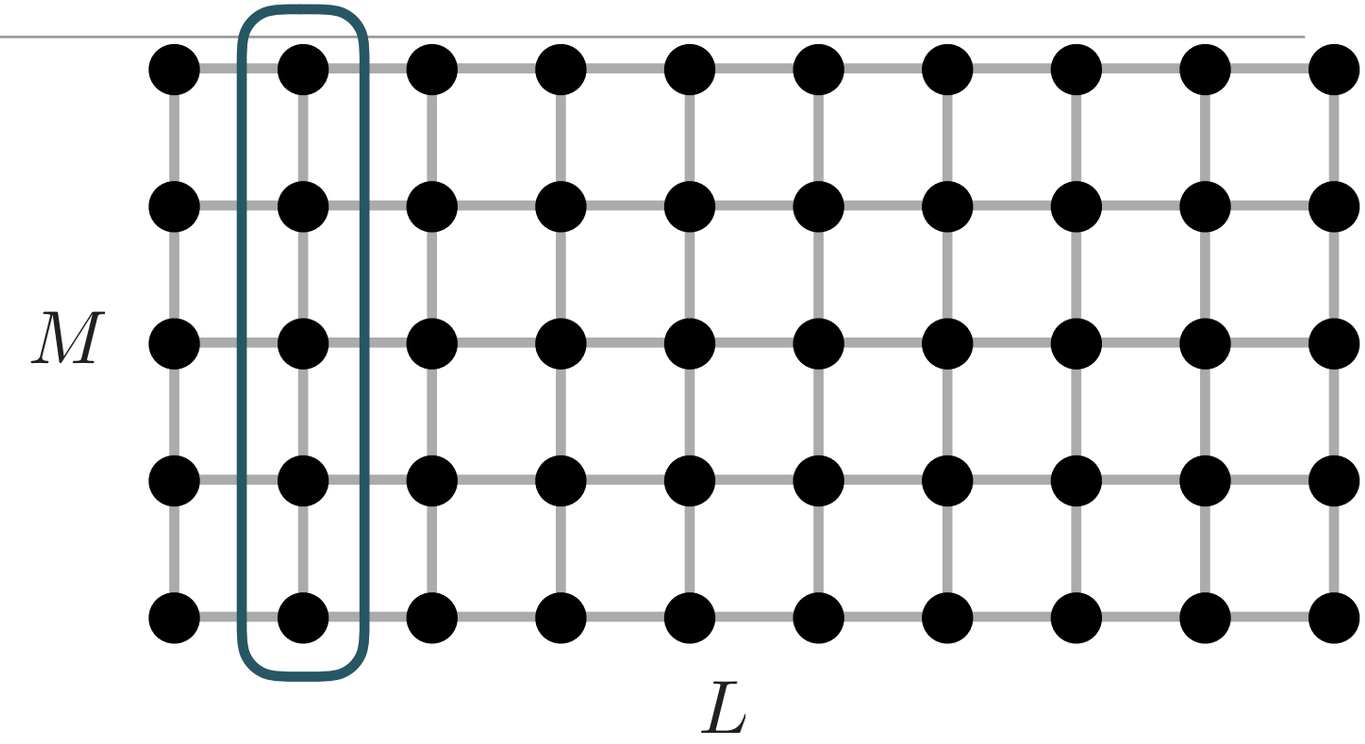
$$Z = \sum_{S_1=\pm 1, S_L=\pm 1} \left[P^t \begin{pmatrix} \lambda_+^{L-1} & 0 \\ 0 & \lambda_-^{L-1} \end{pmatrix} P \right]_{S_1, S_L}$$

分配関数の計算→転送行列の対角化

２次元系の転送行列

L×Mの２次元系

M個のスピンを1セットで
考えると１次元系と同等



転送行列の大きさ

1次元系：2×2

LxMの２次元系：$2^M \times 2^M$ (or $2^L \times 2^L$)

２次元以上では転送行列が系サイズに関して指数的に大！

厳密な計算はすぐに破綻する

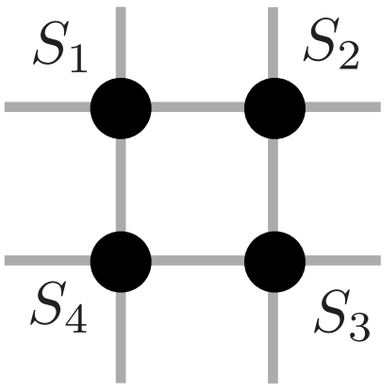
２次元イジング模型だったら、M=40程度が限界
(疎行列の対角化問題)

転送行列の積を近似的に計算

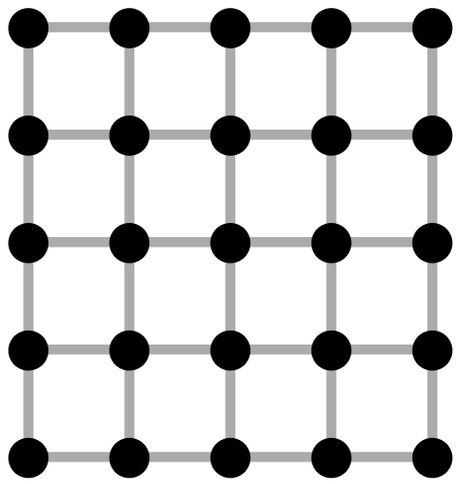
転送行列表現の拡張

２次元正方格子模型の分配関数

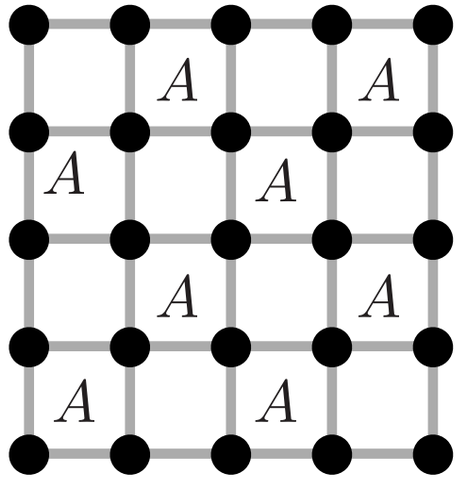
一つの正方形に注目



正方格子



Aの配置



各辺のボルツマン重みの積：４階の“テンソル”

$$A_{S_1,S_2,S_3,S_4} = e^{\beta J(S_1S_2+S_2S_3+S_3S_4+S_4S_1)}$$

イジング模型

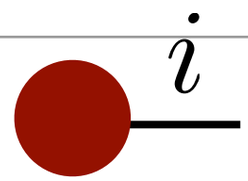
Aは2×2×2×2のテンソル

分配関数＝テンソルの"掛け算"

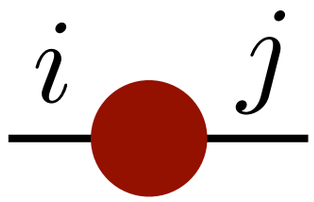
$$Z = \sum_{\{S_i=\pm 1\}} A_{S_1,S_2,S_3,S_4} A_{S_2,S_5,S_6,S_7} \cdots A_{S_i,S_j,S_k,S_l} \cdots$$

ダイアグラムを用いたテンソル表記

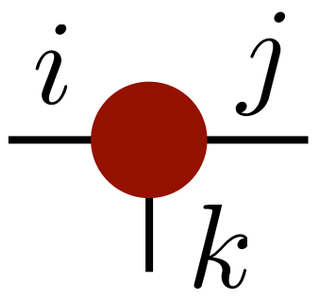
- ベクトル　　$\vec{v}$: v_i



- 行列　　M : $M_{i,j}$



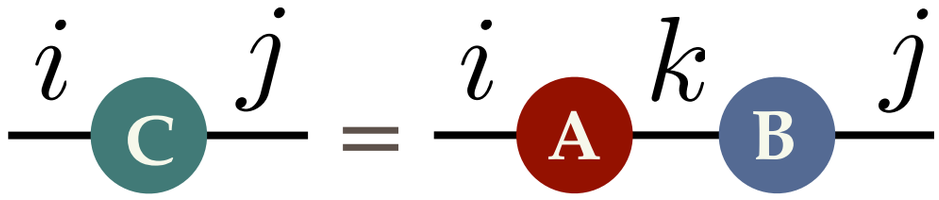
- テンソル　　T : $T_{i,j,k}$



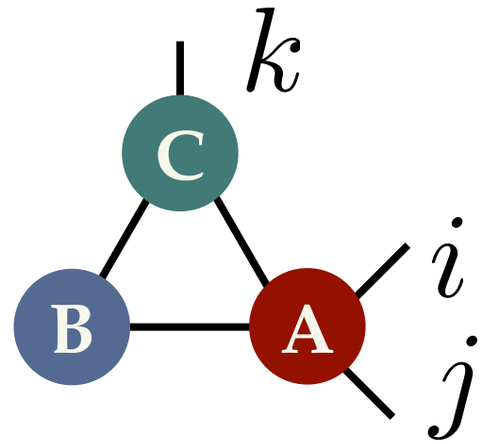
＊n階のテンソル＝n本の足

テンソルの積（縮約）の表現

$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

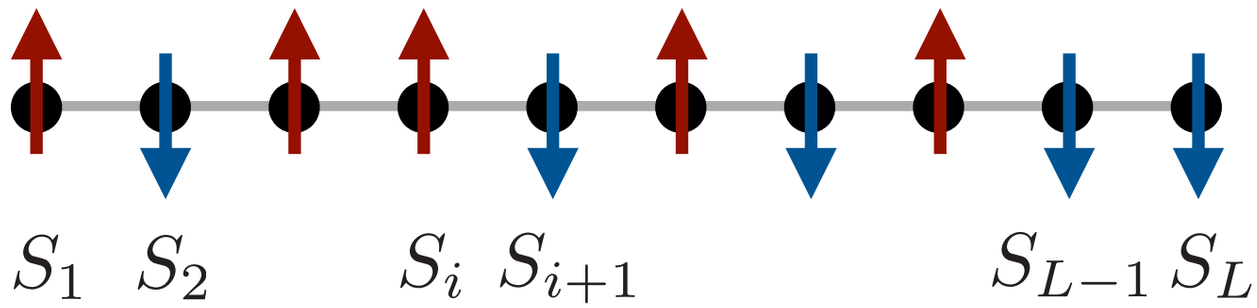


$$\sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$

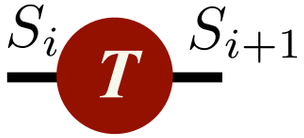


分配関数のダイアグラム

例：１次元イジング模型

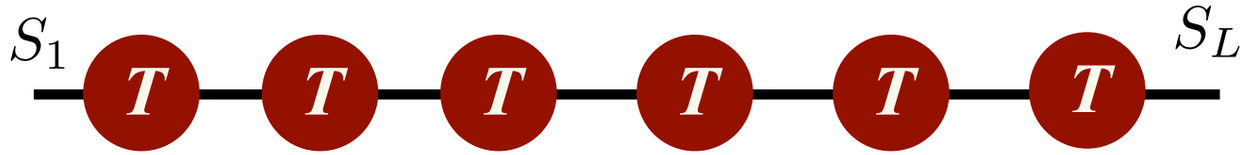


$$T_{S_i,S_{i+1}} = e^{\beta J S_i S_{i+1}}$$



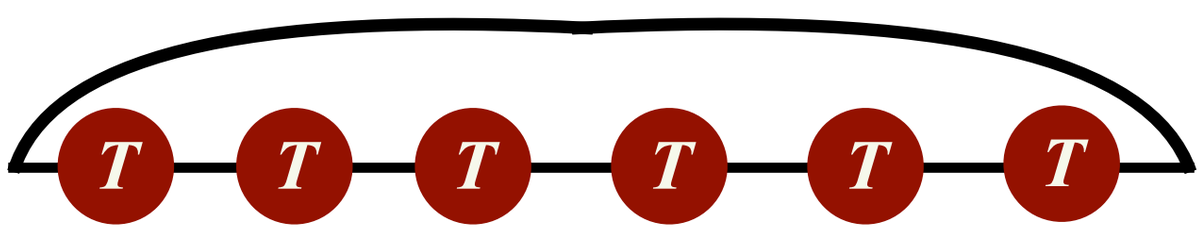
$$Z = \sum_{S_1=\pm1,S_L=\pm1} \left(T^{L-1}\right)_{S_1,S_L}$$

$$= \sum_{S_1=\pm1,S_L=\pm1}$$



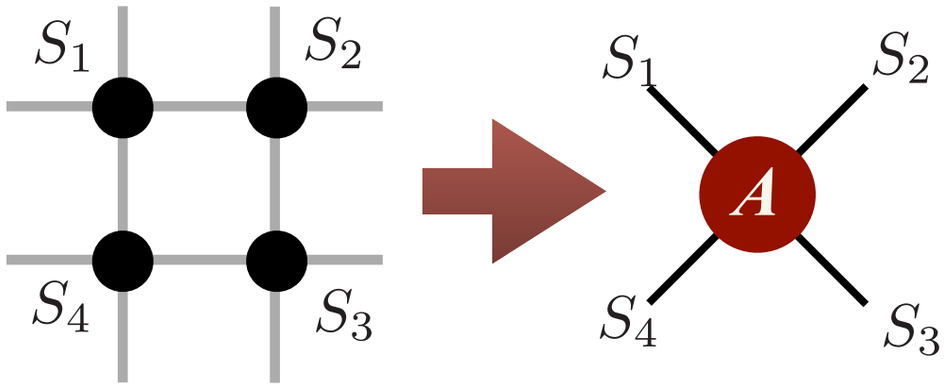
*周期境界条件の場合

$$Z = \mathrm{Tr}\ T^L =$$

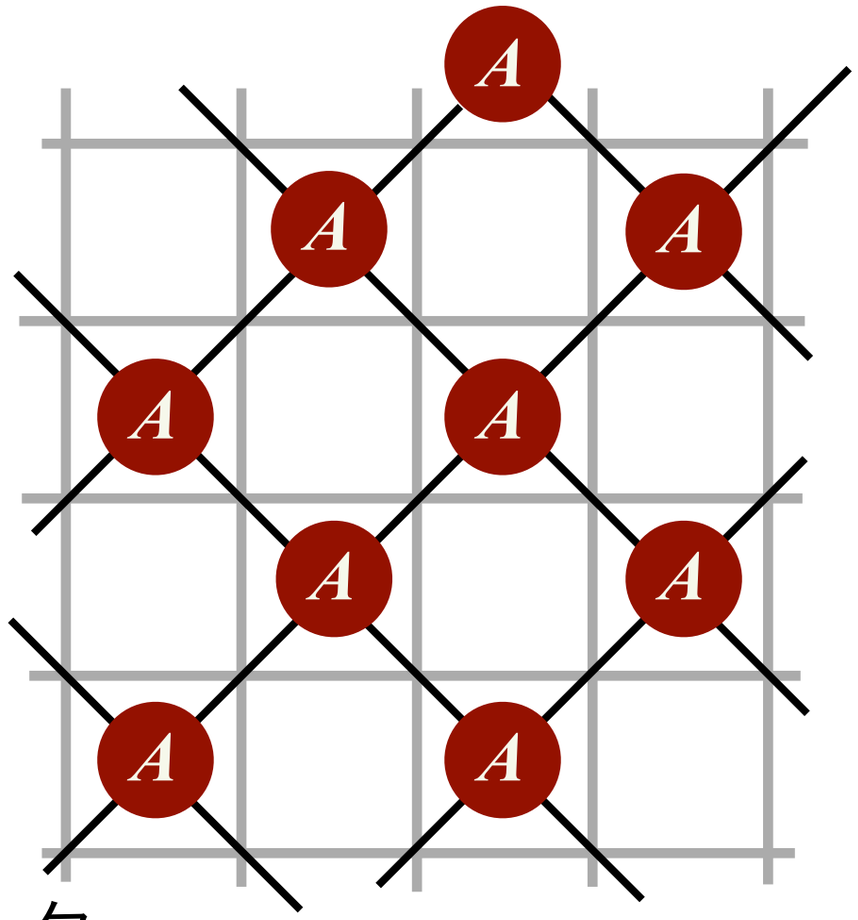


分配関数のテンソルネットワーク表現

$$A_{S_1,S_2,S_3,S_4} = e^{\beta J(S_1S_2+S_2S_3+S_3S_4+S_4S_1)}$$



$$Z =$$



分配関数＝テンソルAの積のネットワーク

テンソルネットワーク

正方格子イジング模型→45度傾いた正方格子ネットワーク

(Real space) Renormalization group
(補足資料)

Example: Ising model

Ising model

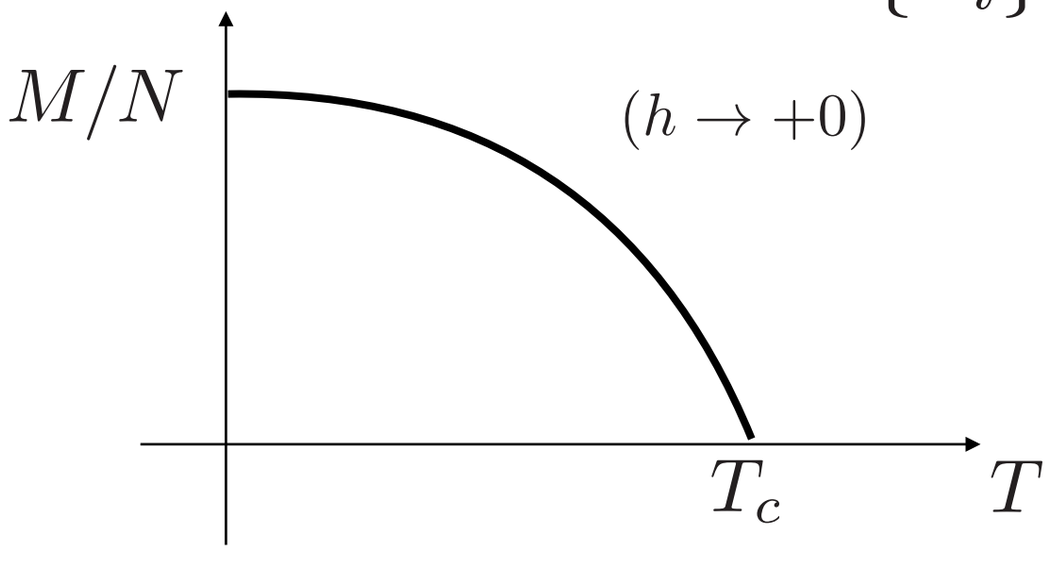
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j - h \sum_i S_i \qquad (S_i = \pm 1 = \uparrow, \downarrow)$$

Canonical ensemble: $P(\Gamma; T) = \frac{1}{Z} \exp\left(-\frac{1}{k_B T} \mathcal{H}(\Gamma) \right)$

Magnetization at T:

$$M(T) = \left\langle \sum_i S_i \right\rangle_T = \sum_\Gamma \sum_i S_i P(\Gamma; T)$$

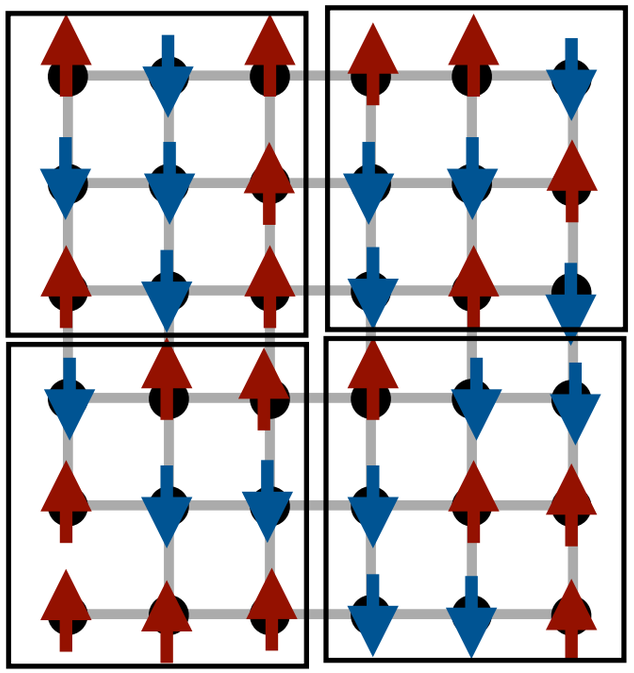
$\Gamma = \{S_i\}$



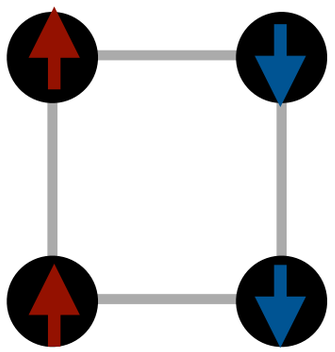
Coarse graining （粗視化）

Block spin transformation
（ブロックスピン変換）

:1　:-1



6×6 system



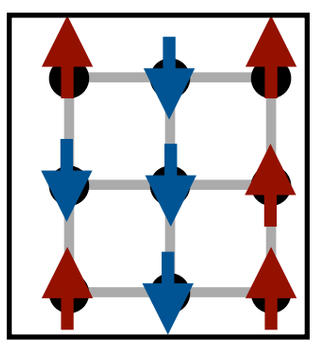
2×2 system

"Length scale" changes

coarse grained spin

$$\sum_{i\in\text{block}} S_i > 0 \quad : \quad \uparrow$$

$$\sum_{i\in\text{block}} S_i < 0 \quad : \quad \downarrow$$



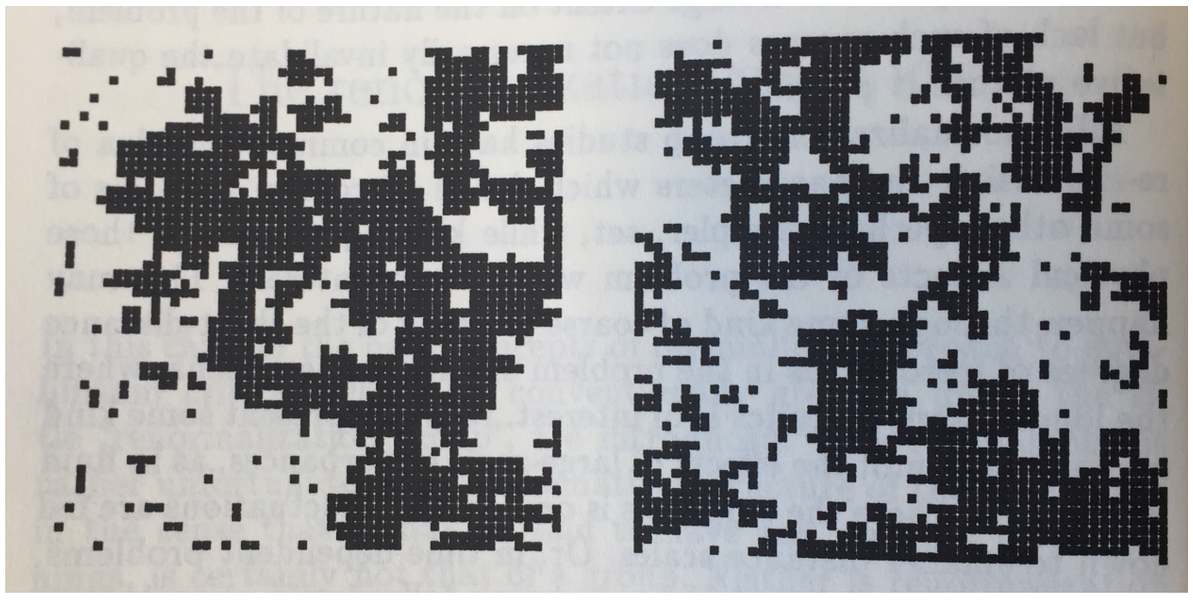
=　↑

Example of block spin transformation

Figure taken from a book "Scaling and Renormalization in Statistical Physics", John Cardy

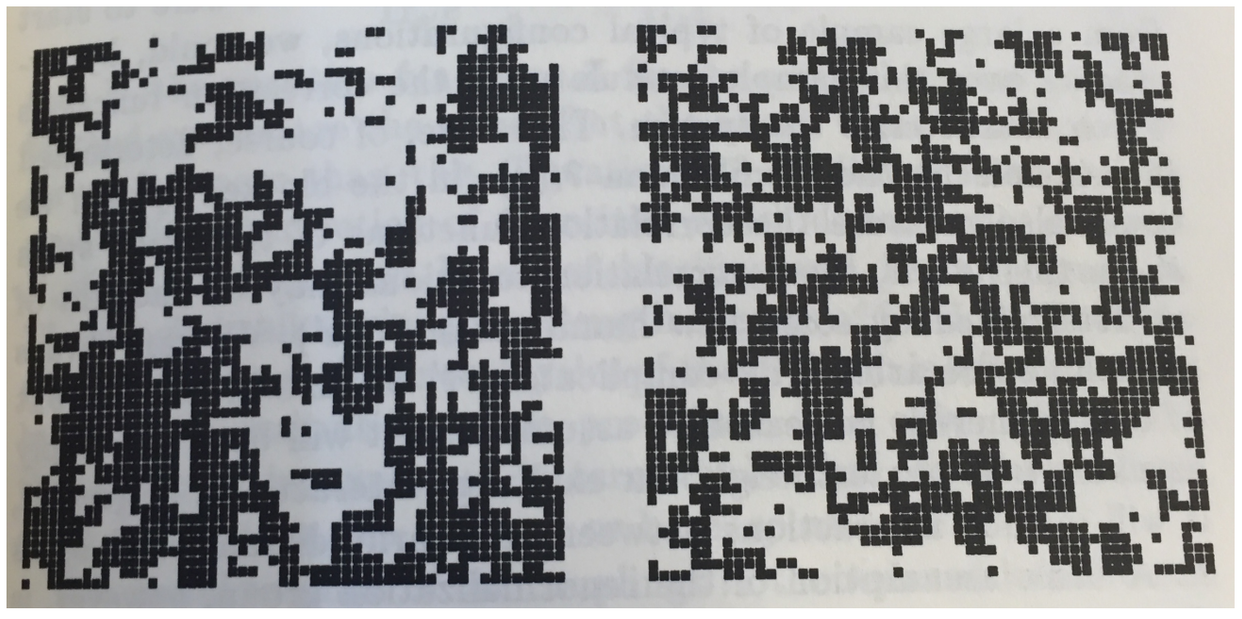
$T=T_c$ (critical point)

(befor) (after)



$T > T_c$

(befor) (after)



- At the critical point, the block spin transformation does not change"image" qualitatively.

  → "Scale invariance"

- At T > Tc, the block spin transformation changes typical "cluster size"

Partition function after coarse graining

Partition function after a block spin transformation:

(for simplicity, we set $J/k_BT = K$)

$$Z = \sum_{\{S_i = \pm 1\}} e^{K \sum_{\langle i,j \rangle} S_i S_j} = \sum_{\{S'_i = \pm 1\}} e^{-\mathcal{H}'(\{S'_i\})}$$

2^{L^d}

(d-dimensional system with length L)

$2^{(L/b)^d}$

(d-dimensional system with length L/b)

By block spin transformation, the partition function is represented by smaller # of spins with a modified Hamiltonian

→ Information compression by "tracing out" short range fluctuations

Coarse grained Hamiltonian

Partition function after a block spin transformation:

$$e^{-\mathcal{H}'(\{S'_i\})} = \sum_{\{S_i\}\in\{S'_i\}} e^{K\sum_{\langle i,j\rangle} S_i S_j}$$

Sum over spin configurations corresponds to {S'}

Suppose H' has the same form with the original Hamiltonian, which characterized only one parameter K:

$$\mathcal{H}' = K' \sum_{\langle i,j\rangle} S'_i S'_j$$

By repeating the procedure, we can draw a flow of "K"

$$K \to K' \to K'' \to \cdots \to K^{\infty}$$

"renormalization group"
(繰り込み群)

$$K' = \mathcal{R}_b(K)$$

R_b : transformation with scale *b*

Renormalization flow

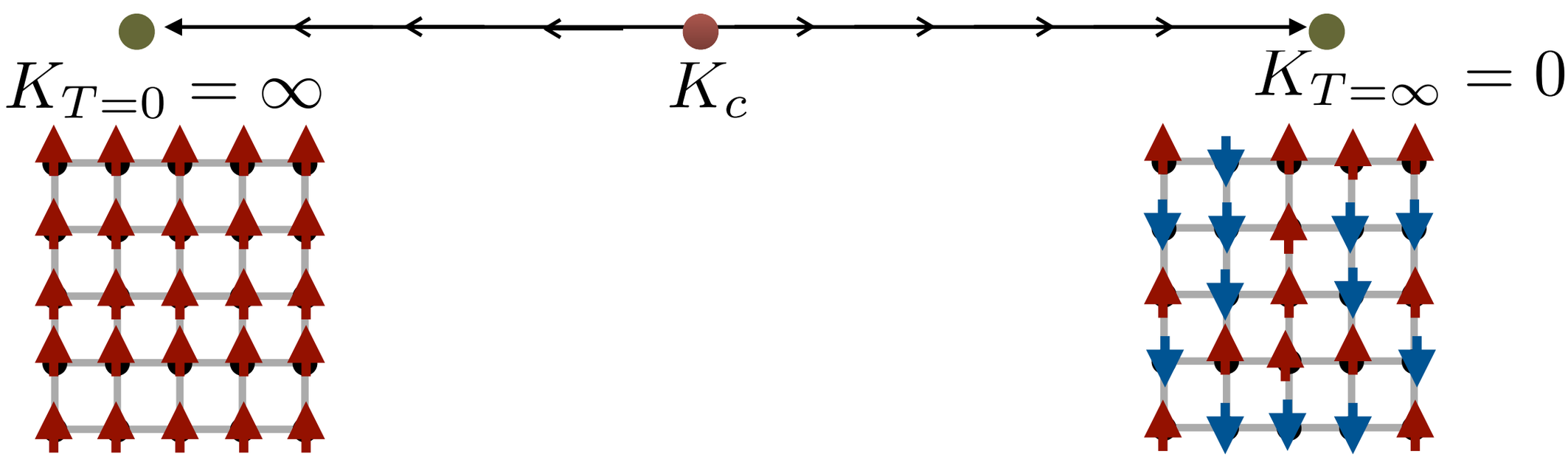
Renormalization group: $K' = \mathcal{R}_b(K)$

Fixed point（固定点）: $K^* = \mathcal{R}_b(K^*)$

Unchanged under renormalization

Typically, we have three fixed points for a phase transition:

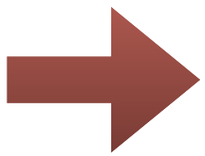
Corresponding T=0, T=∞, and T=Tc



General case

$$e^{-\mathcal{H}'(\{S_i'\})} = \sum_{\{S_i\}\in\{S_i'\}} e^{K\sum_{\langle i,j\rangle} S_i S_j}$$

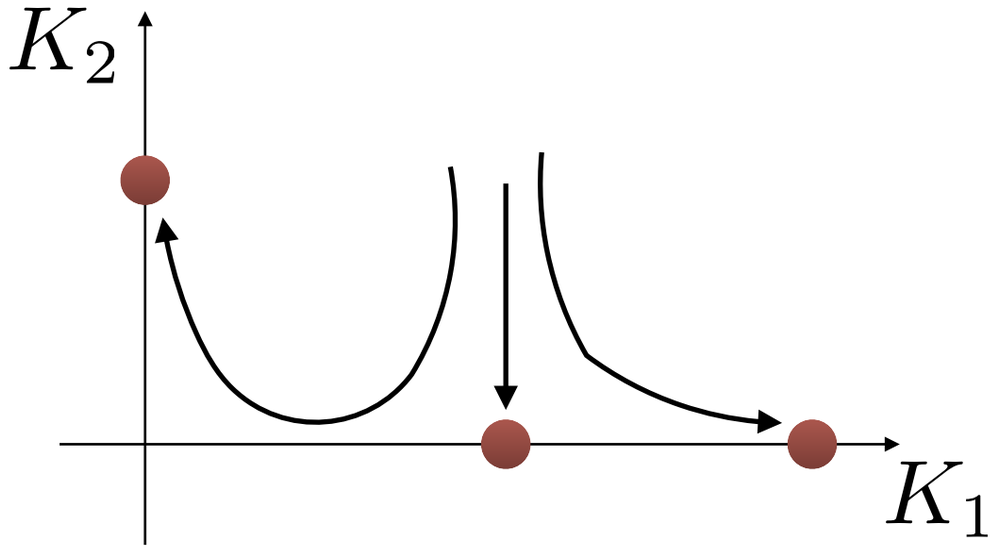
In general, H' contains many body interaction such as $S_iS_jS_kS_l$.



We need more than one parameter: $\{K_1, K_2, \dots\}$

Renormalization group: $\vec{K}' = \mathcal{R}_b(\vec{K})$

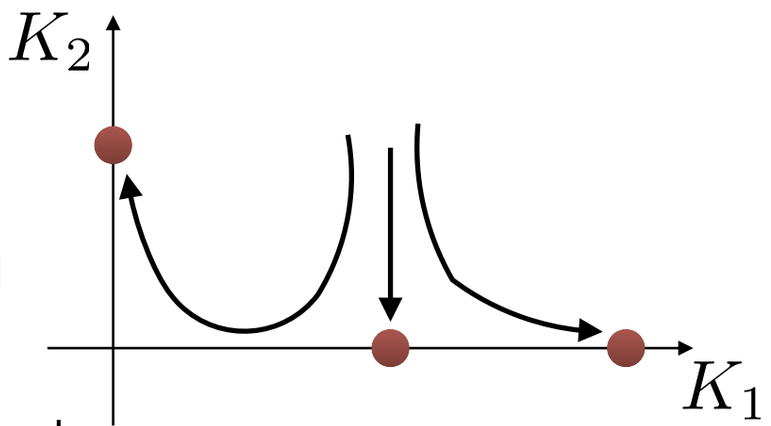
RG characterize a flow in parameter space



Critical exponents and eigenvalues

Linearization around Kc: $\vec{K}' = \mathcal{R}_b(\vec{K})$

$$\vec{K}' - \vec{K}_c \simeq \mathcal{M}_b(\vec{K} - \vec{K}_c)$$



$\mathcal{M}_b$:Matrix applied in parameter space

y_i :Eigenvalue of $\mathcal{M}_b$

$\delta\vec{K}_i$:Eigenvector

"Relevant" $|y_i| > 1$ → $\delta\vec{K}_i$ increases along renormalization

"Irrelevant" $|y_i| < 1$ → $\delta\vec{K}_i$ decreases along renormalization

Critical exponents are related to relevant eigenvalues!

Tensor network renormalization

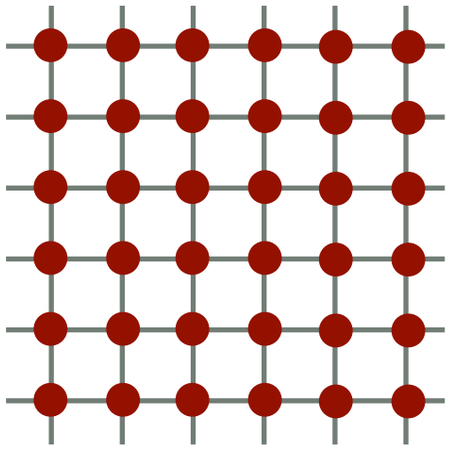
Tensor network renormalization
（テンソルネットワーク繰り込み）

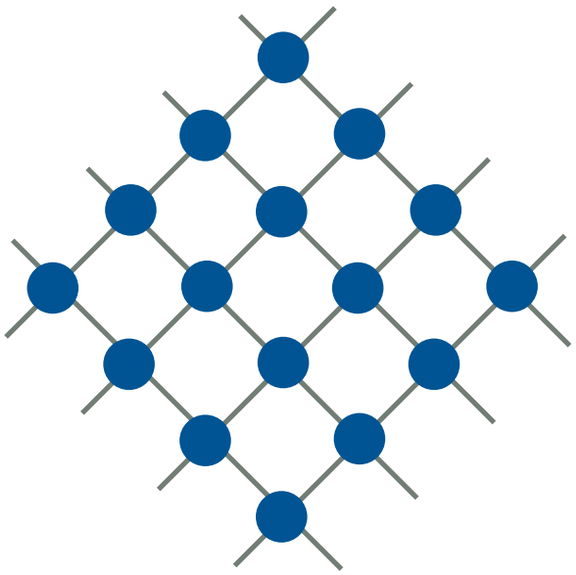
- Approximate calculation of a tensor network contraction by using "coarse graining" （粗視化）of the network
    - Coarse graining ⟷ Real space renormalization
    - （粗視化）⟷（実空間繰り込み）
- It can be applicable to (basically) any lattices, and the idea (algorithm) is independent on "models" represented by tensor networks.
    - Potential application to wide range of the science.

Outline of tensor network renormalization

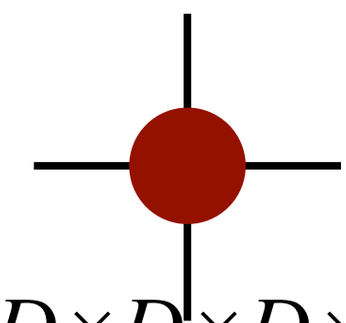
Scaler represented
by $L \times L$ tensors

$(L \times L)/2$ tensors



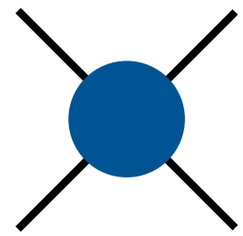


Corse graining (Renormalization)
into $\sqrt{2}$ times longer scale.



A: $D \times D \times D \times D$

Approximation

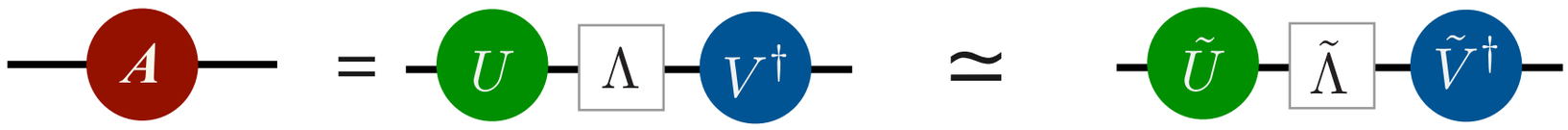


$\tilde{A}$: $D \times D \times D \times D$

Reduce the number of tensors
keeping their size constant

Key technique: low rank approximation by SVD

Best low-rank approximation of a matrix = SVD



$A : M \times N$ (M ≦ N)

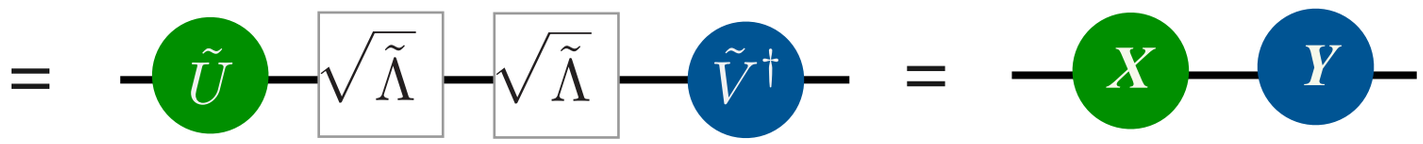
$\Lambda : M \times M$ (Diagonal matrix)

$U, V : (\mathrm{M},\mathrm{N}) \times M$

$\tilde{\Lambda} : R \times R$ (Keeping the R largest singular values)

$\tilde{U}, \tilde{V} : (\mathrm{M},\mathrm{N}) \times R$

In addition,



$\sqrt{\tilde{\Lambda}}$:Diagonal matrix those elements are $\sqrt{\lambda}$

$X = \tilde{U}\sqrt{\tilde{\Lambda}} \quad : \mathrm{M} \times R$

$Y = \sqrt{\tilde{\Lambda}}\tilde{V}^{\dagger} \quad : \mathrm{R} \times M$

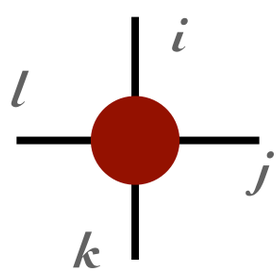
By SVD, we can decompose a matrix into a product of "small" matrices.

Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

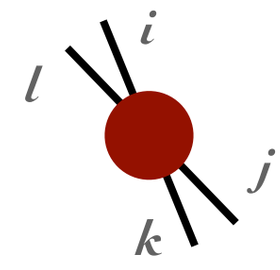
1. Decomposition

Regard a tensor as a matrix



$A_{i,j,k,l}$

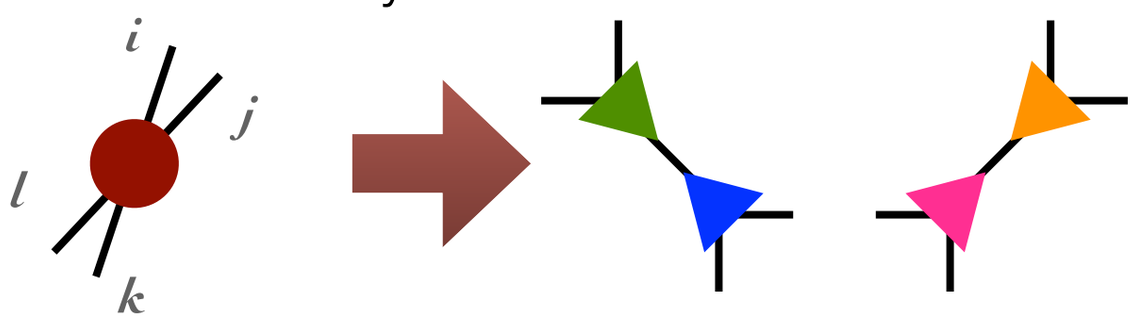
A: $D \times D \times D \times D$



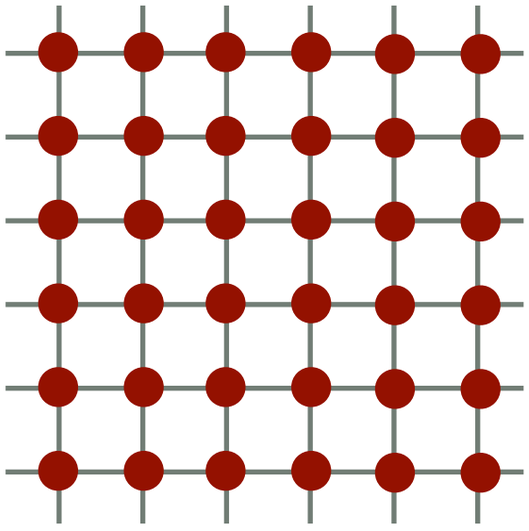
$A_{(i,l),(j,k)}$

$A : D^2 \times D^2$

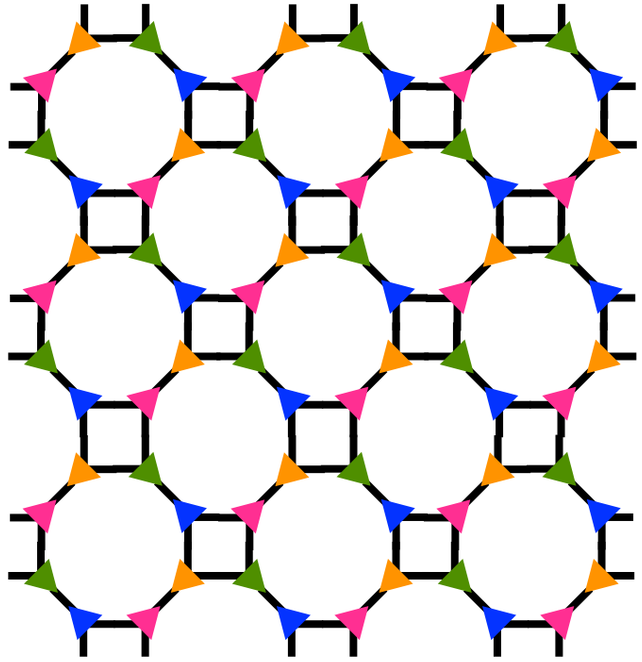
D-rank approximation
by SVD



$A_{(i,j),(k,l)}$



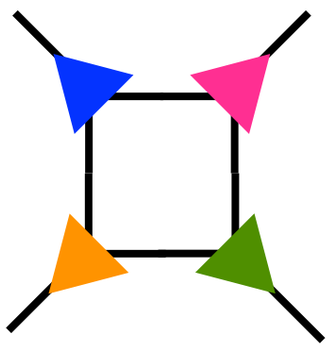
Approximation



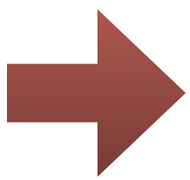
Recipe of Tensor Renormalization Group (TRG)

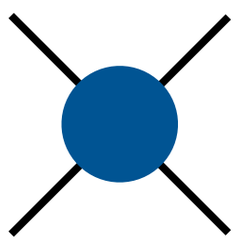
M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)

2. Coarse graining



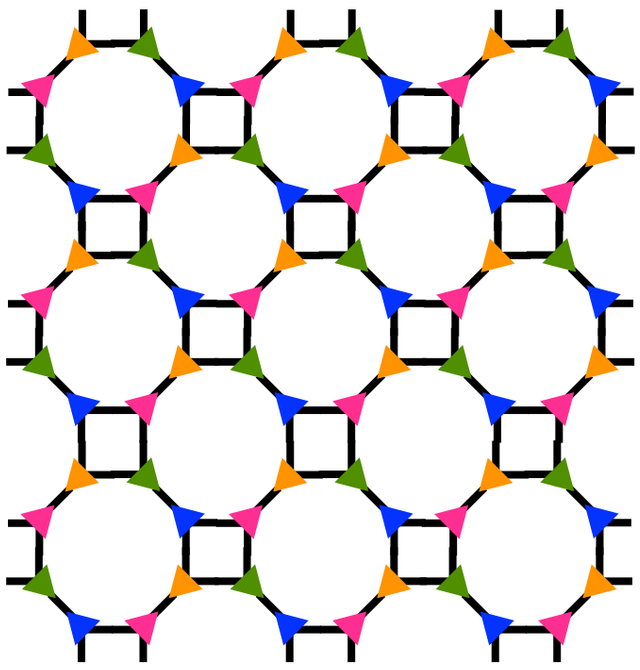
Contraction of inner indices

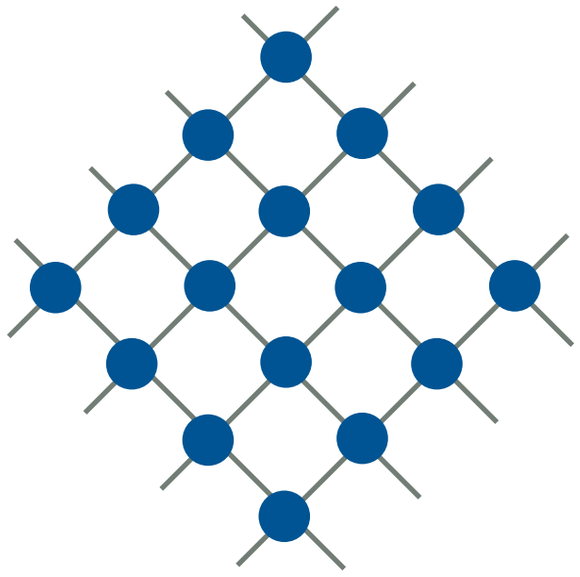




$$\tilde{A} : D \times D \times D \times D$$

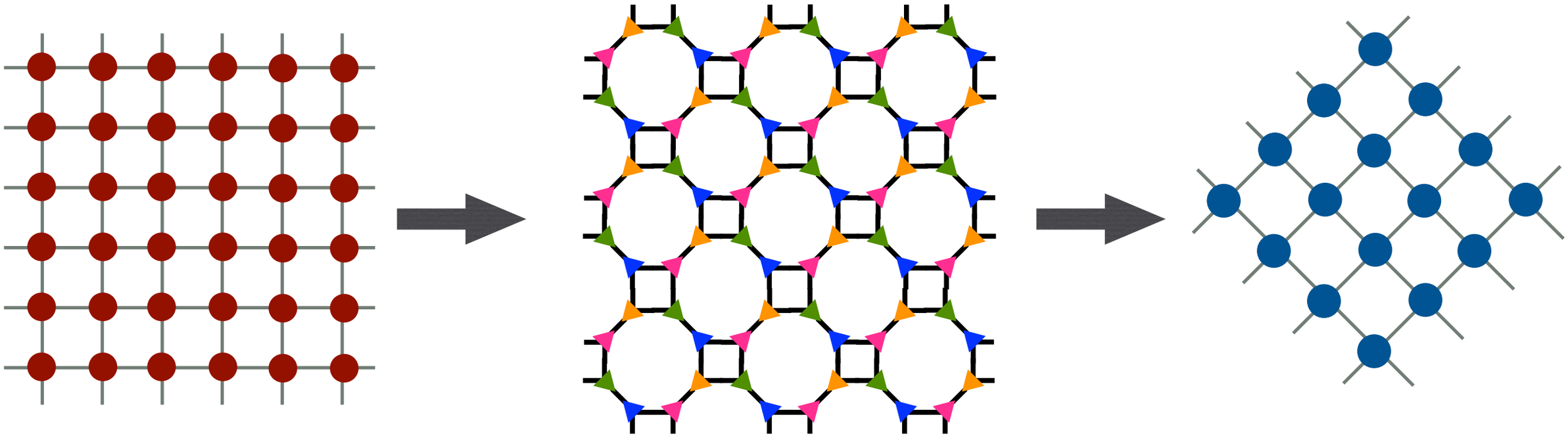
In total, two original tensors are coarse grained into a new tensor.





Recipe of Tensor Renormalization Group (TRG)

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007)
Z.-C. Gu, M. Levin and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008)



Calculation cost: SVD= $O(D^6)$

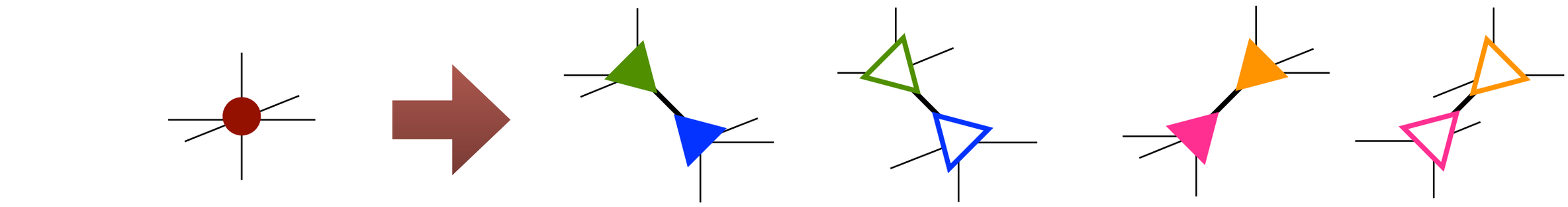
Contraction= $O(D^6)$ (per tensor)

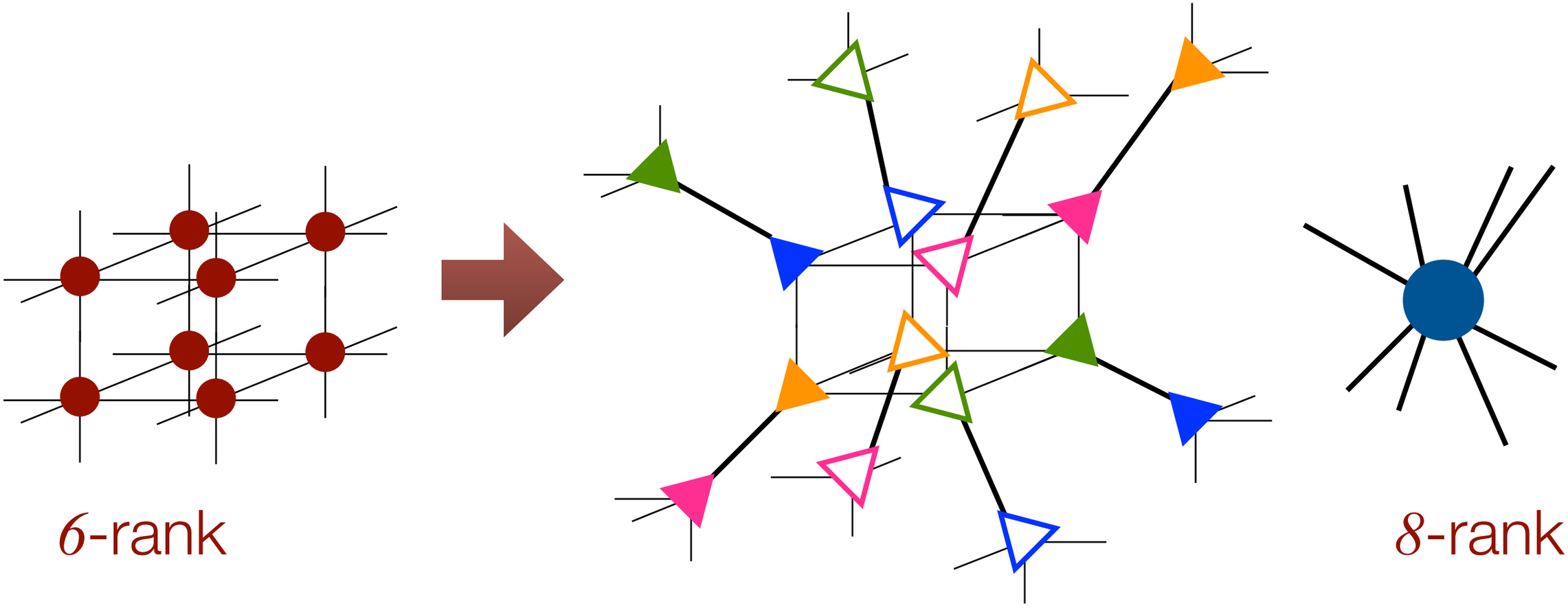
*By one TRG step, # of tensors is reduced by 1/2.

We can calculate the contraction in polynomial cost!

Tensor renormalization group for higher dimensions

Simple generalization of TRG to cubic lattice (three dimension)

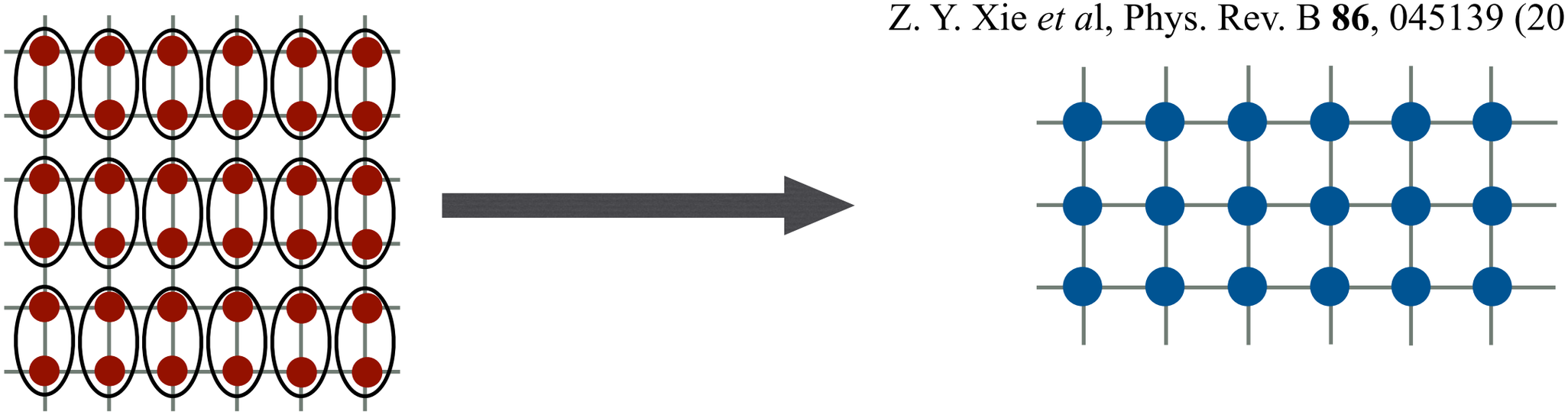




Tensor renormalization group by using HOSVD

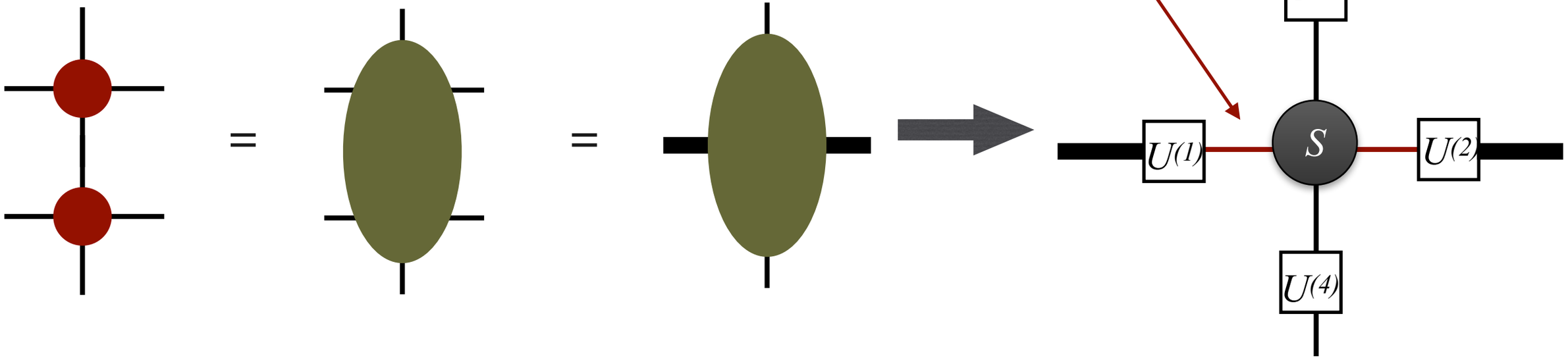
Anisotropic coarse graining by using HOSVD instead of SVD

Z. Y. Xie *et al*, Phys. Rev. B **86**, 045139 (2012)



Basic idea of HOTRG algorithm:

(For details, see the original paper.)



(Reference of HOSVD)

Tucker decomposition: generalization of SVD

Tucker decomposition:

(Tucker (1963))

Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

$U^{(i)}$: Factor matrix (usually unitary)

S :Core tensor

$$T_{ijkl} = \sum_{i'=1}^{I}\sum_{j'=1}^{J}\sum_{k'=1}^{K}\sum_{l'=1}^{L} S_{i'j'k'l'} U^{(1)}_{ii'} U^{(2)}_{jj'} U^{(3)}_{kk'} U^{(4)}_{ll'}$$

*If S is "diagonal", Tucker decomposition becomes CP decomposition.

Low "rank" approximation

$$T_{ijkl} = \sum_{i'=1}^{I'}\sum_{j'=1}^{J'}\sum_{k'=1}^{K'}\sum_{l'=1}^{L'} \tilde{S}_{i'j'k'l'} \tilde{U}^{(1)}_{ii'} \tilde{U}^{(2)}_{jj'} \tilde{U}^{(3)}_{kk'} \tilde{U}^{(4)}_{ll'}$$

$I' < I, \quad J' < J, \quad K' < K, \quad L' < L$

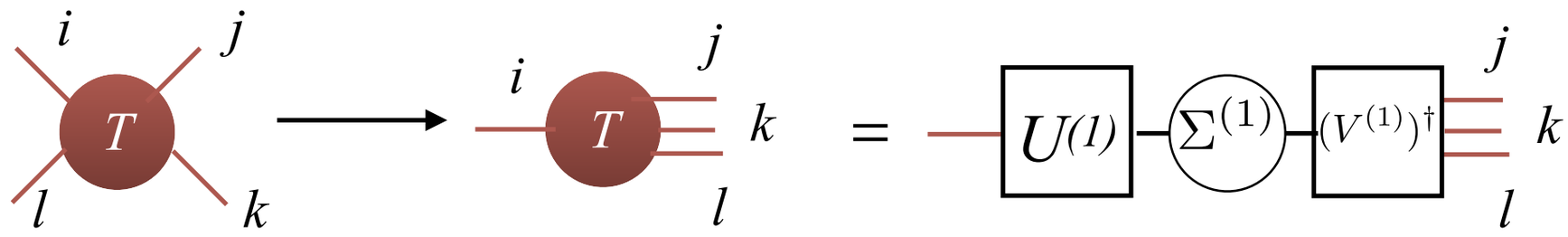
rank-(I',J', K', L') approximation

(Reference of HOSVD)

Higher order SVD (HOSVD)

L. De Lathauwer et al, SIAM J. Matrix Anal. & Appl., **21**, 1253 (2000)

Define a factor matrix from matrix SVD:



Core tensor is calculated as

$$S_{i'j'k'l'} \equiv \sum_{ijkl} T_{ijkl} (U^{(1)})^\dagger_{i'i} (U^{(2)})^\dagger_{j'j} (U^{(3)})^\dagger_{k'k} (U^{(4)})^\dagger_{l'l}$$

Properties of the core tensor

$$S^*_{:,i_n=\alpha,:,:} \cdot S_{:,i_n=\beta,:,:} = \begin{cases} 0 & (\alpha \neq \beta) \\ (\sigma^{(n)}_\alpha)^2 & (\alpha = \beta) \end{cases}$$

Dot product

$$A \cdot B \equiv \sum_{i,j,k,l} A_{ijkl} B_{ijkl}$$

Generalization of the diagonal matrix Σ in matrix SVD.

* Low-rank approximation based on HOSVD is not optimal.

Power of the HOTRG

Advantage:

- HOTRG does not change the network structure.
  - We can easily generalize it to higher dimensions.
- Low-rank approximation is based on the cluster of two tensors.
  - At the approximation, we take into account more information.
  - More efficient than TRG where SVD is done for a single tensor.

Disadvantage:

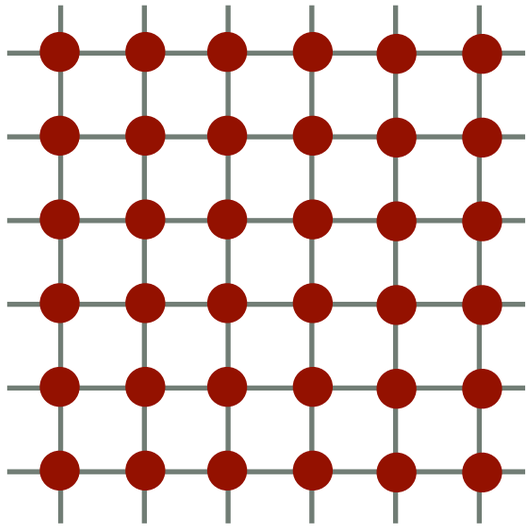
- HOTRG needs higher cost than TRG.
  - $O(D^7)$ in HOTRG $\longleftrightarrow$ $O(D^6)$ in TRG

Application to a classical partition function

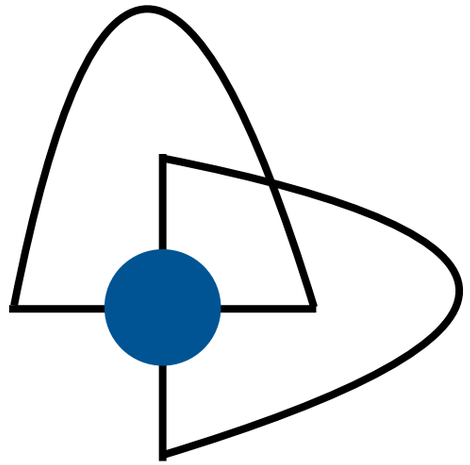
Partition function

(Periodic boundary condition)

$Z =$



Repeat TRG step until only a few tensors remain.



We can easily calculate physical quantities from Z.

Free energy: $F = -k_B T \ln Z$

Energy: $E = -\dfrac{\partial \ln Z}{\partial \beta}$ (Use difference approximation)

Specific heat: $C = \dfrac{1}{k_B T^2} \dfrac{\partial^2 \ln Z}{\partial \beta^2}$ (Use difference approximation)

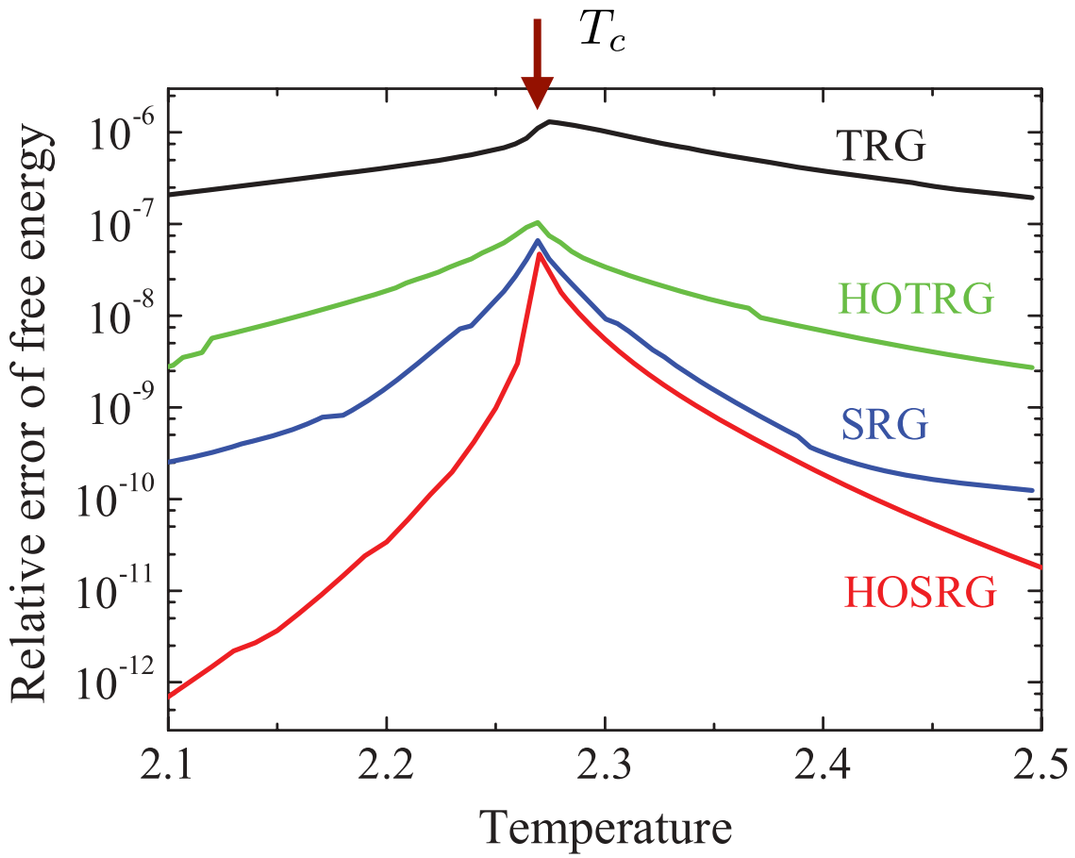
Example of calculation

Ising model in infinite size

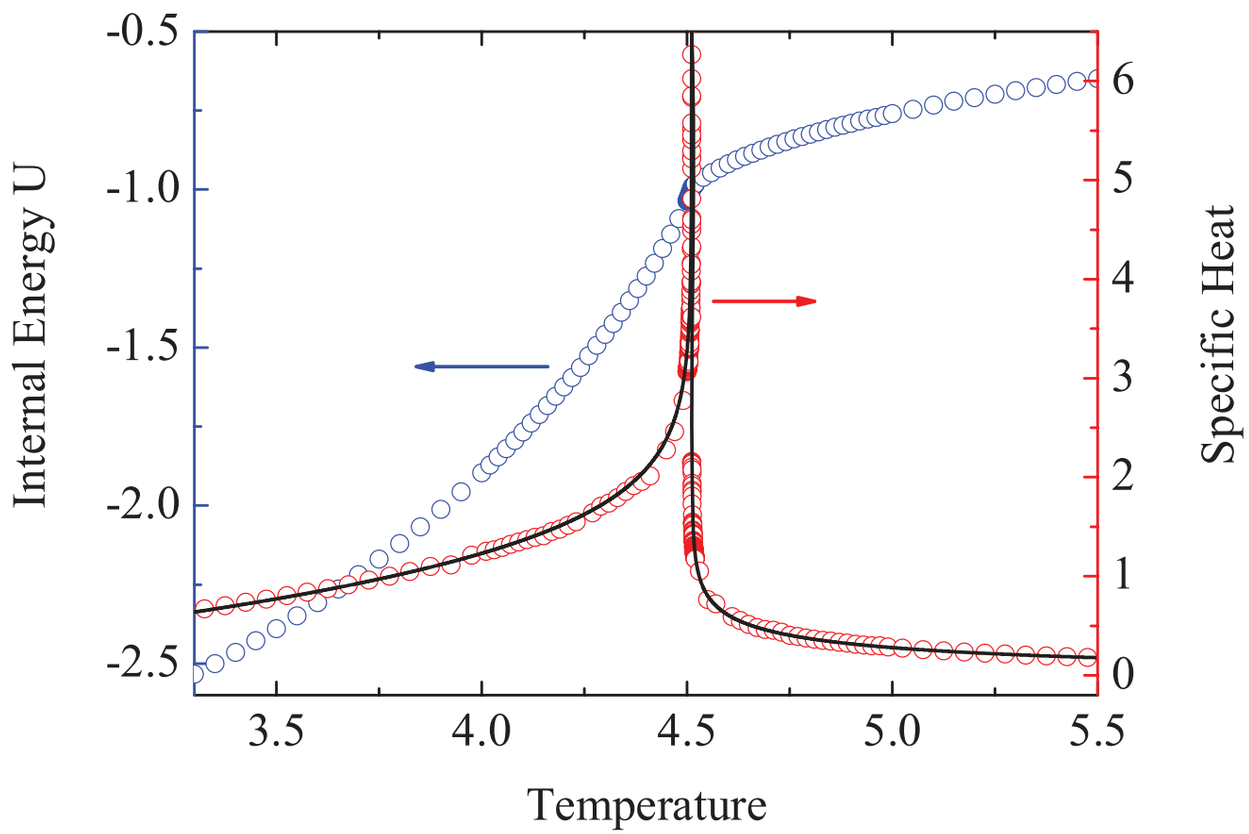
$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

Z. Y. Xie *et al*, Phys. Rev. B **86**, 045139 (2012)

Error of free energy for 2D Ising model



Energy and specific heat of 3D Ising model



$$T_c/J = \frac{2}{\ln(1+\sqrt{2})} \simeq 2.269$$

Tensor network renormalization at the critical point

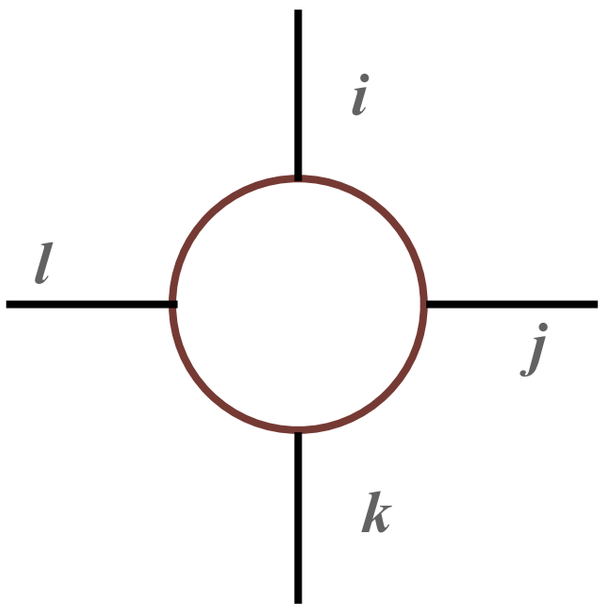
— When the accuracy of TRG becomes worse?

(少し高度な内容)

Correlation (entanglement) within a tensor

General tensor

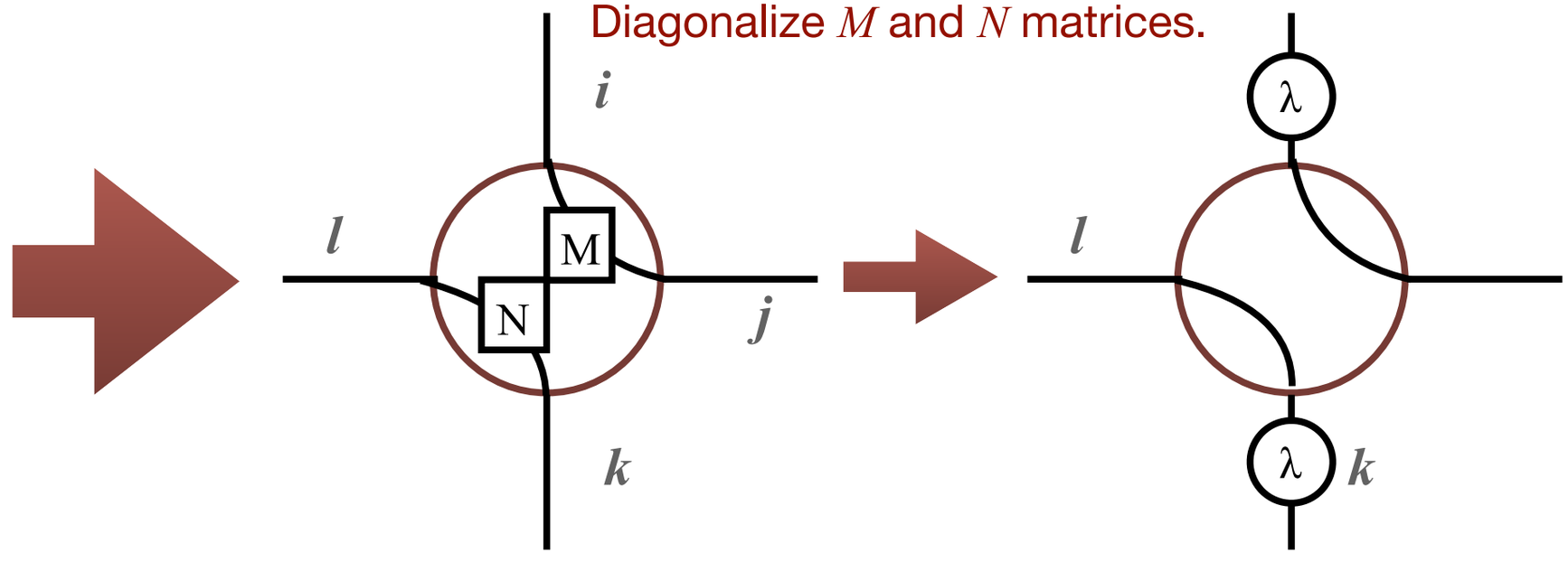
A_{ijkl}



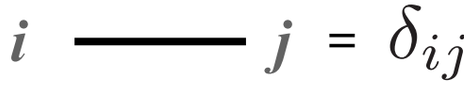
Eg. *Correlation* in (i,j) and (k,j)

$$A_{ijkl} = M_{ij} N_{kl} \quad \Rightarrow \quad A_{ijkl} = \lambda_i^{(M)} \lambda_k^{(N)} \delta_{ij} \delta_{kl}$$

Diagonalize M and N matrices.



New rule for representation of the correlation:



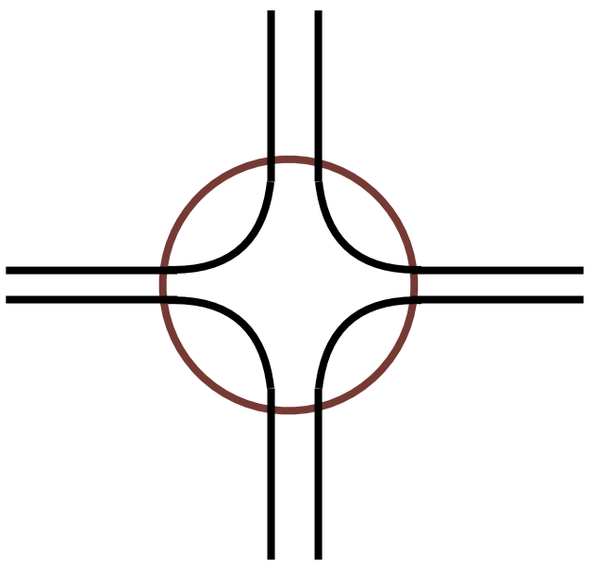
$= \delta_{ij}$

(+ we neglect eigenvalues in the graph.)

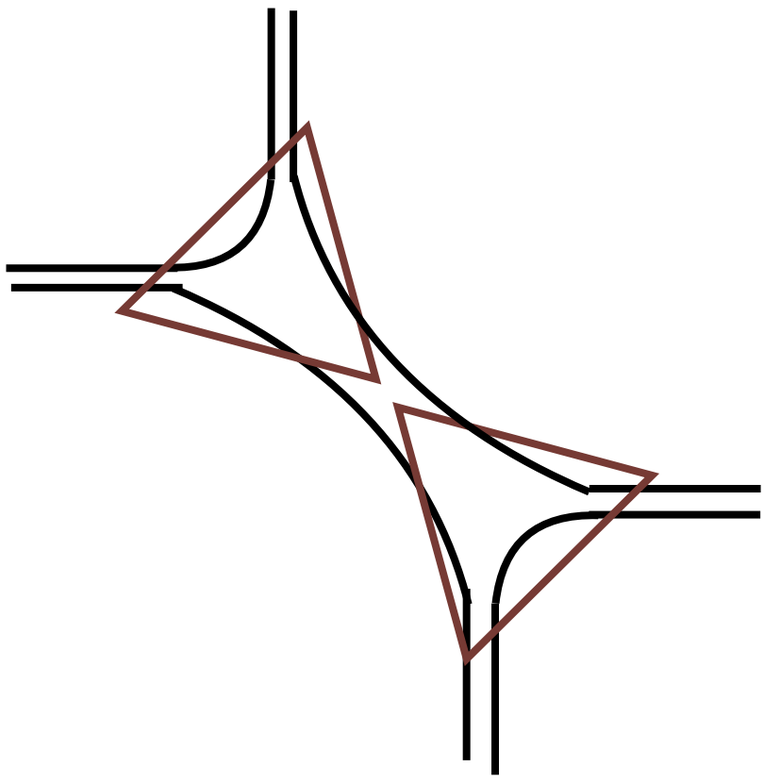
Fixed point of TRG: Corner Double Line tensor
(固定点)

Corner Double Line (CDL) tensor:

There are correlations among the nearest legs.



SVD



Original bond dimension = D

Single line: bond dimension $\sqrt{D}$

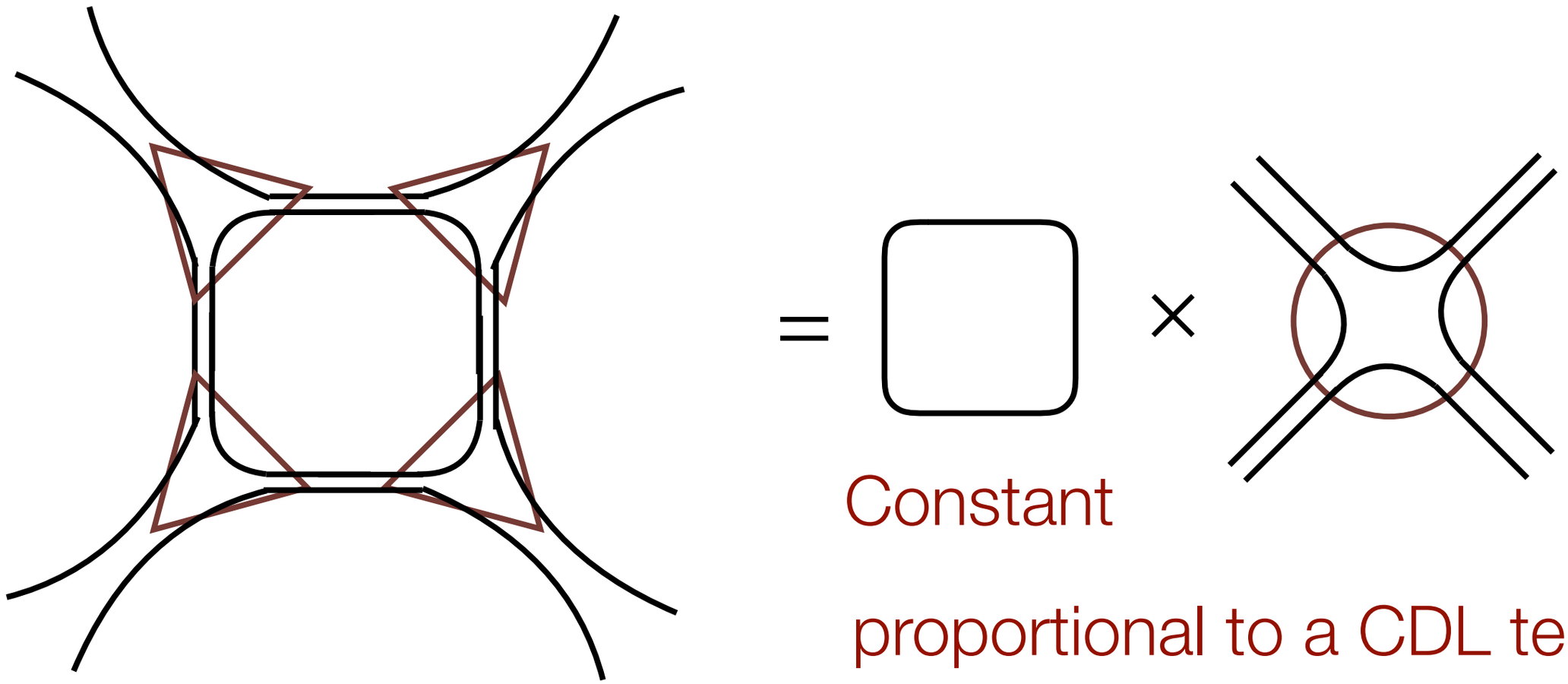
Degree of freedoms connecting two tensors.

Two lines = D

No truncation error at SVD
(Original rank = D)

Fixed point of TRG: Corner Double Line tensor

Contraction of four tensors in TRG:



Constant

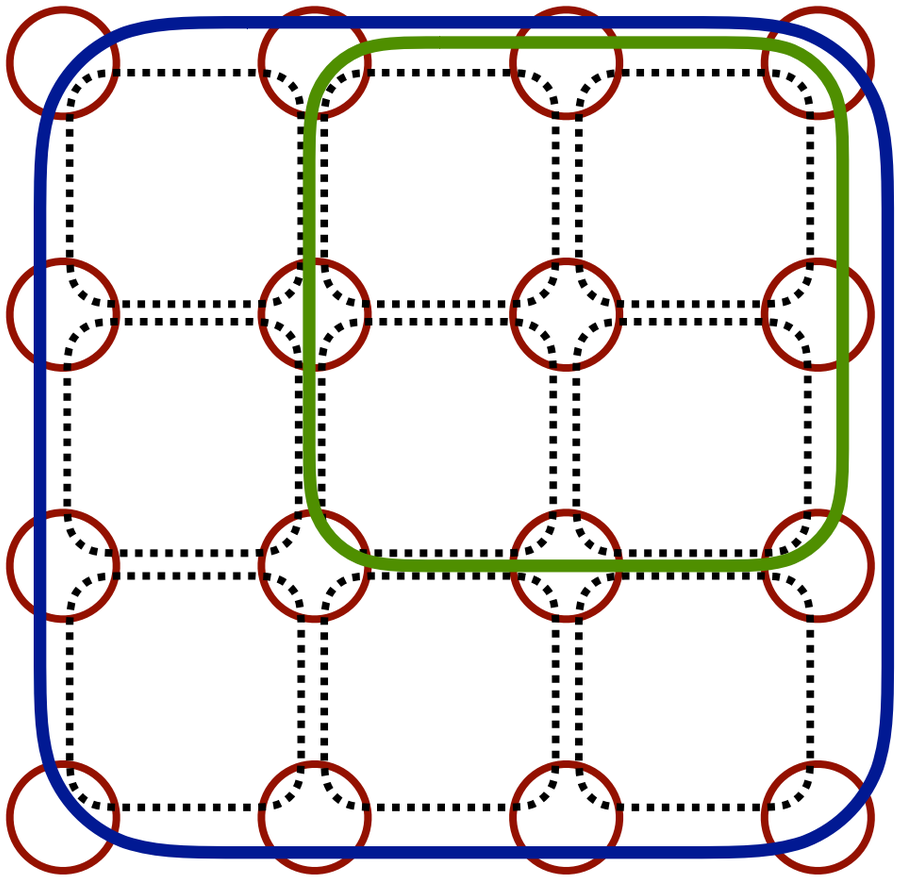
proportional to a CDL tensor!

CDL tensor is a fixed point of TRG (and also HOTRG).

CDL tensor remains as CDL tensor along TRG.

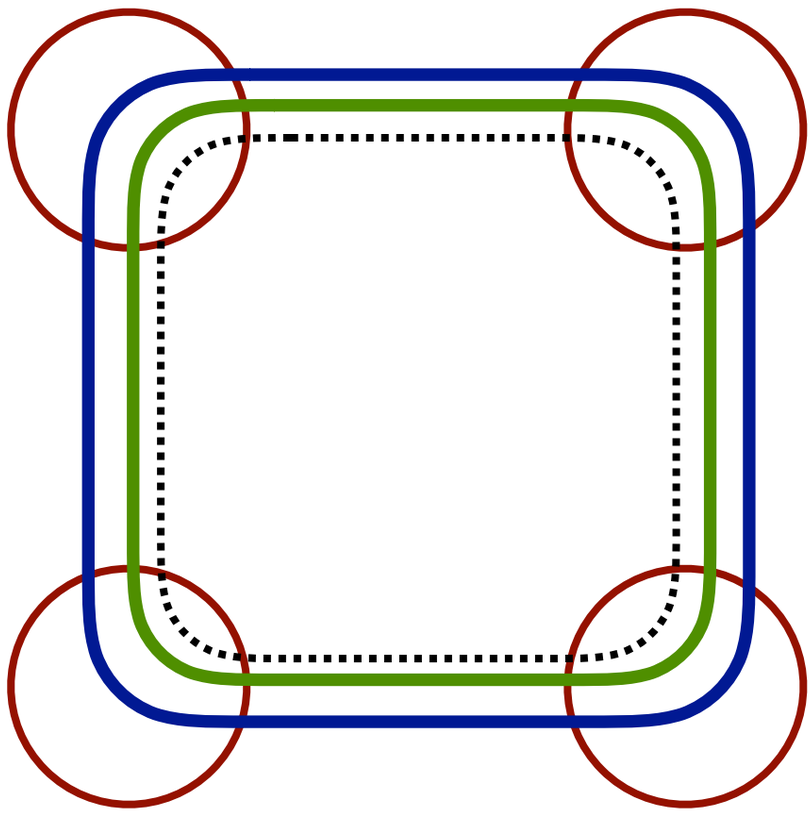
Problems in TRG: accumulation of correlations

Correlation in several scales



TRG

Correlations remains after TRG.



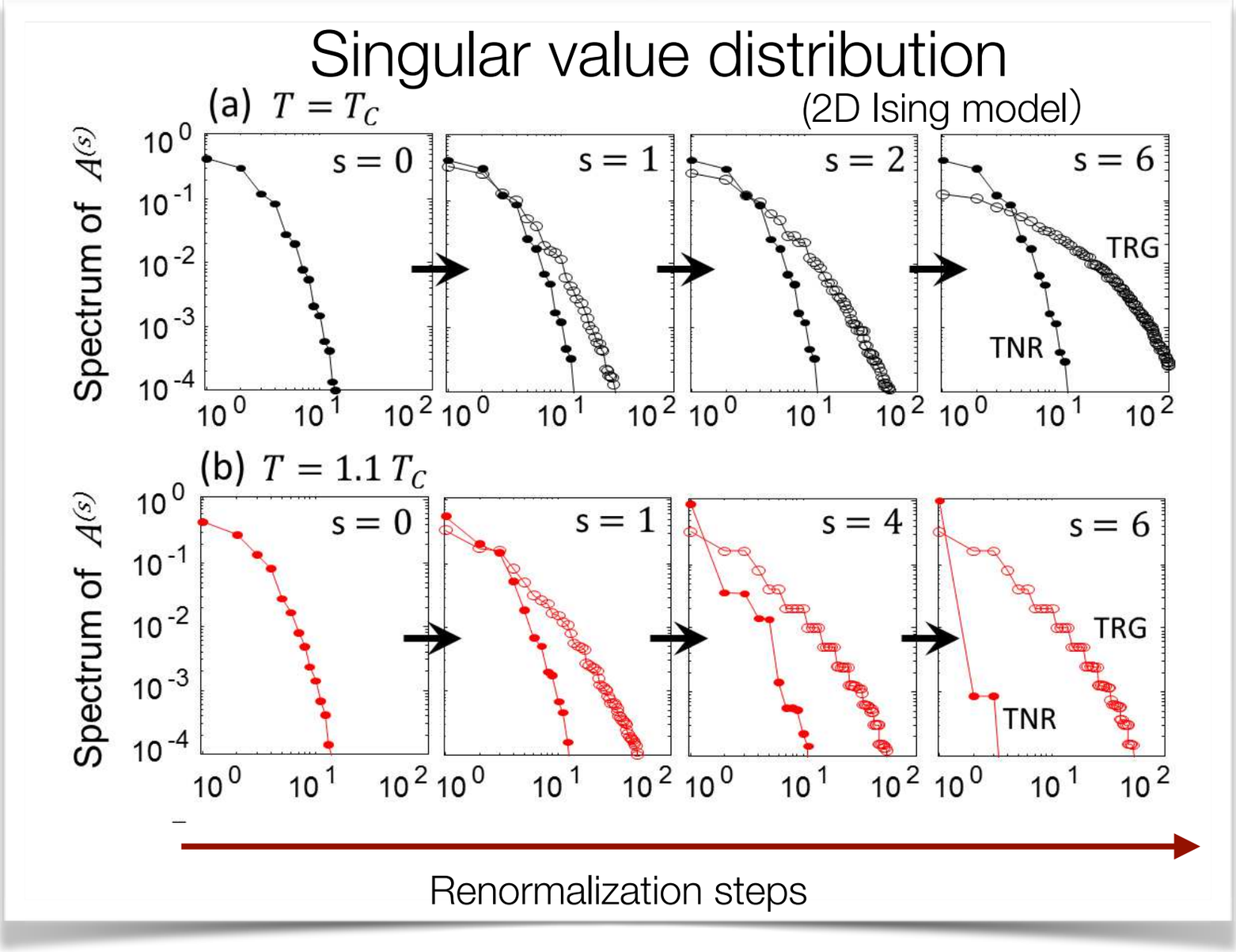
Ideal renormalization:

Correlation in shorter scales should be removed.
Only the correlation in the present scale exists.

TRG：

Correlations in all scales remain.

Problem in TRG: increase of truncation error



G. Evenbly and G. Vidal
Phys. Rev. Lett. **115**, 180405 (2015)

In TRG, the width of the singular value distribution increases along renormalization.

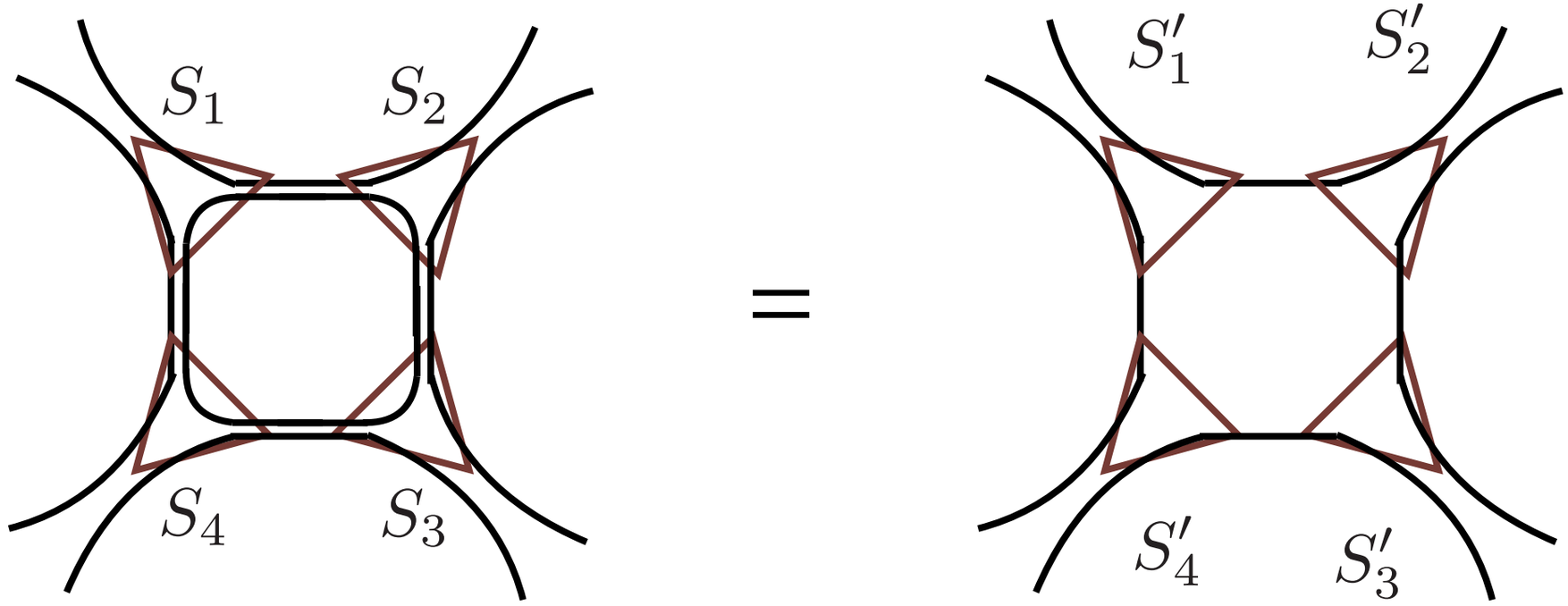
→ Increase of truncation error (decrease of accuracy)

Improvement of TRG：Entanglement Filtering

Try to remove CDL structure at renormalization steps.

Z.-C. Gu and X.-G Wen, Phys. Rev. B 80, 155131 (2009)

Idea:



S: $D \times D \times D$

S': $D \times D' \times D'$

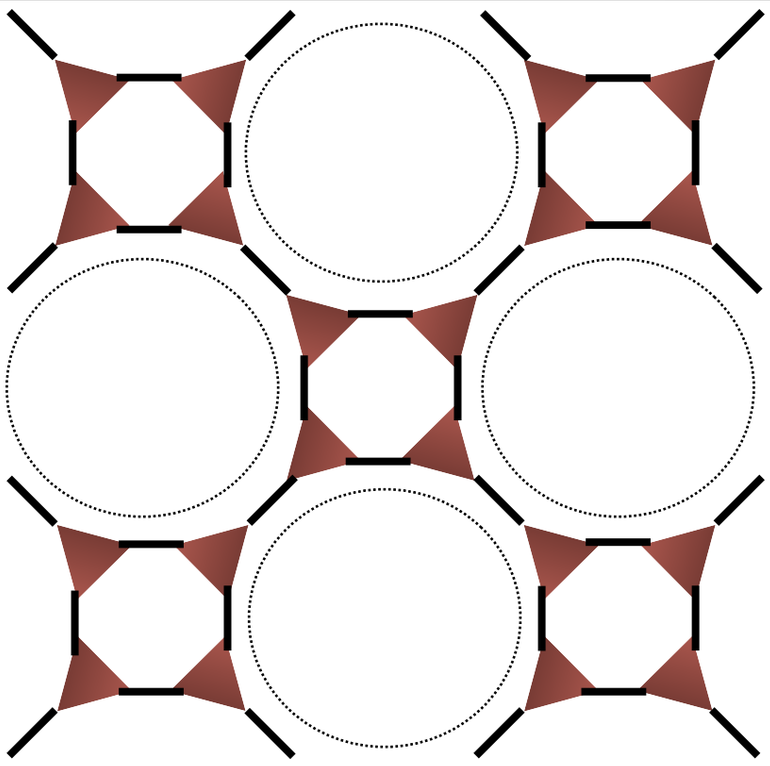
$D' \sim \sqrt{D}$

S' = $\square^{1/4}$ S

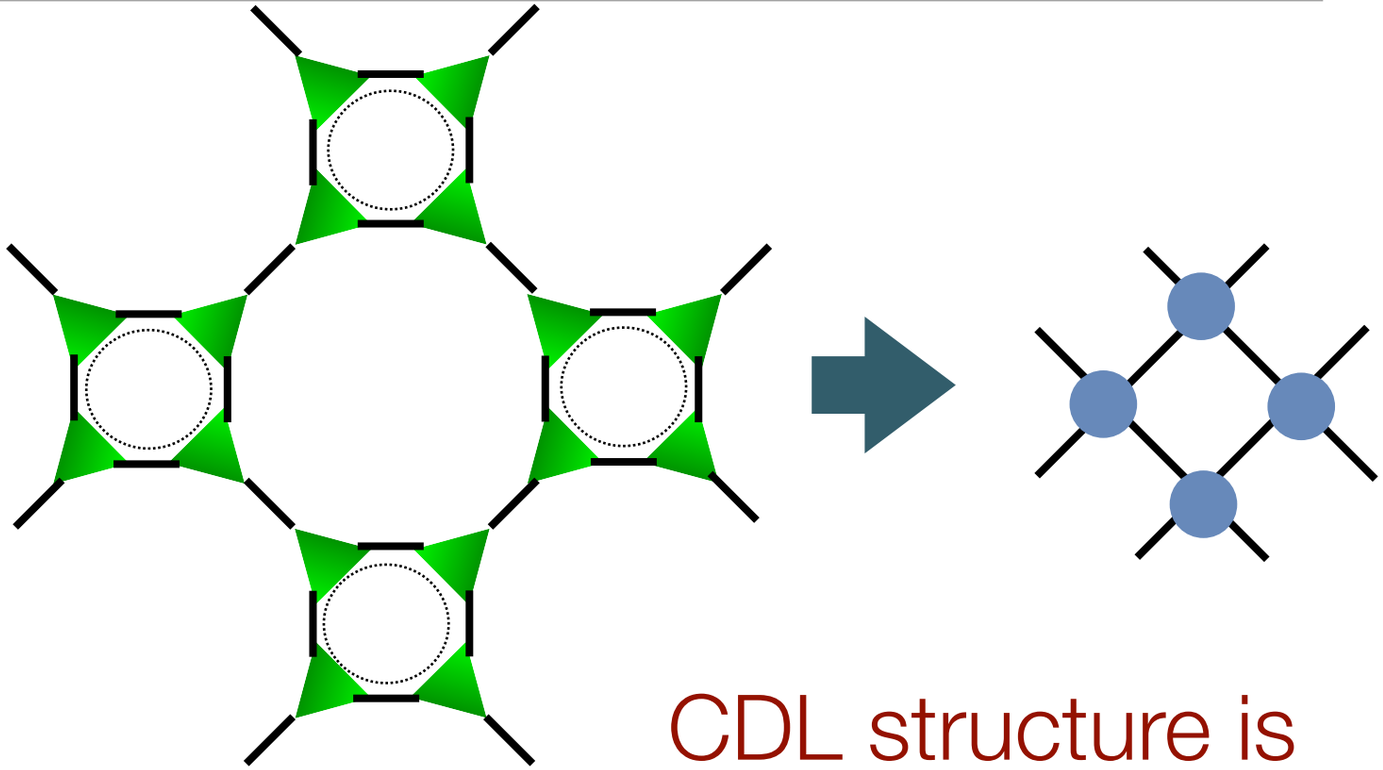
Insert this "approximation" into the TRG algorithm.

Tensor Entanglement Filtering Renormalization

Z.-C. Gu and X.-G Wen, Phys. Rev. B 80, 155131 (2009)



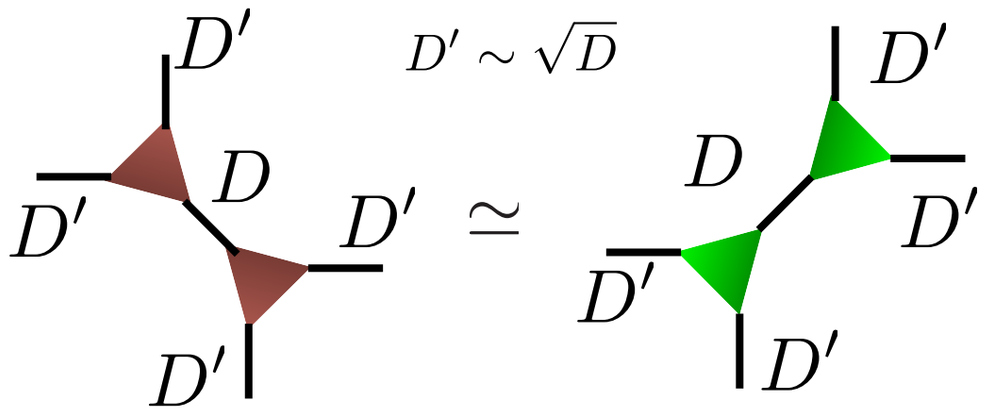
S' (after approximation)



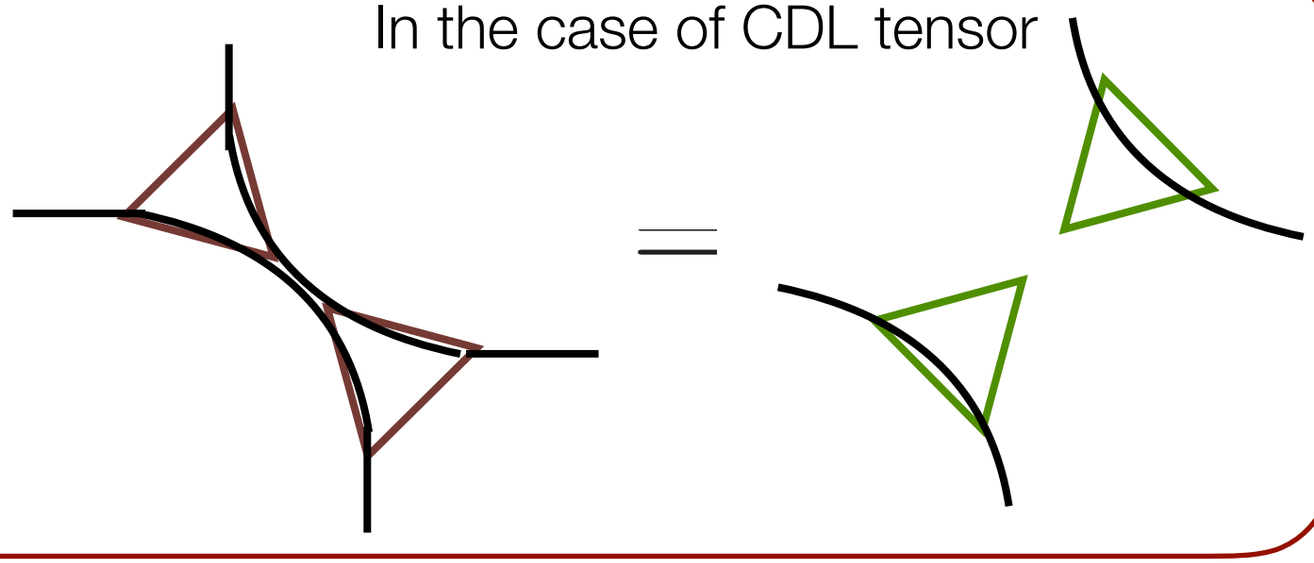
CDL structure is perfectly removed!

Change of SVD: $D' < D$

$D' \sim \sqrt{D}$



In the case of CDL tensor



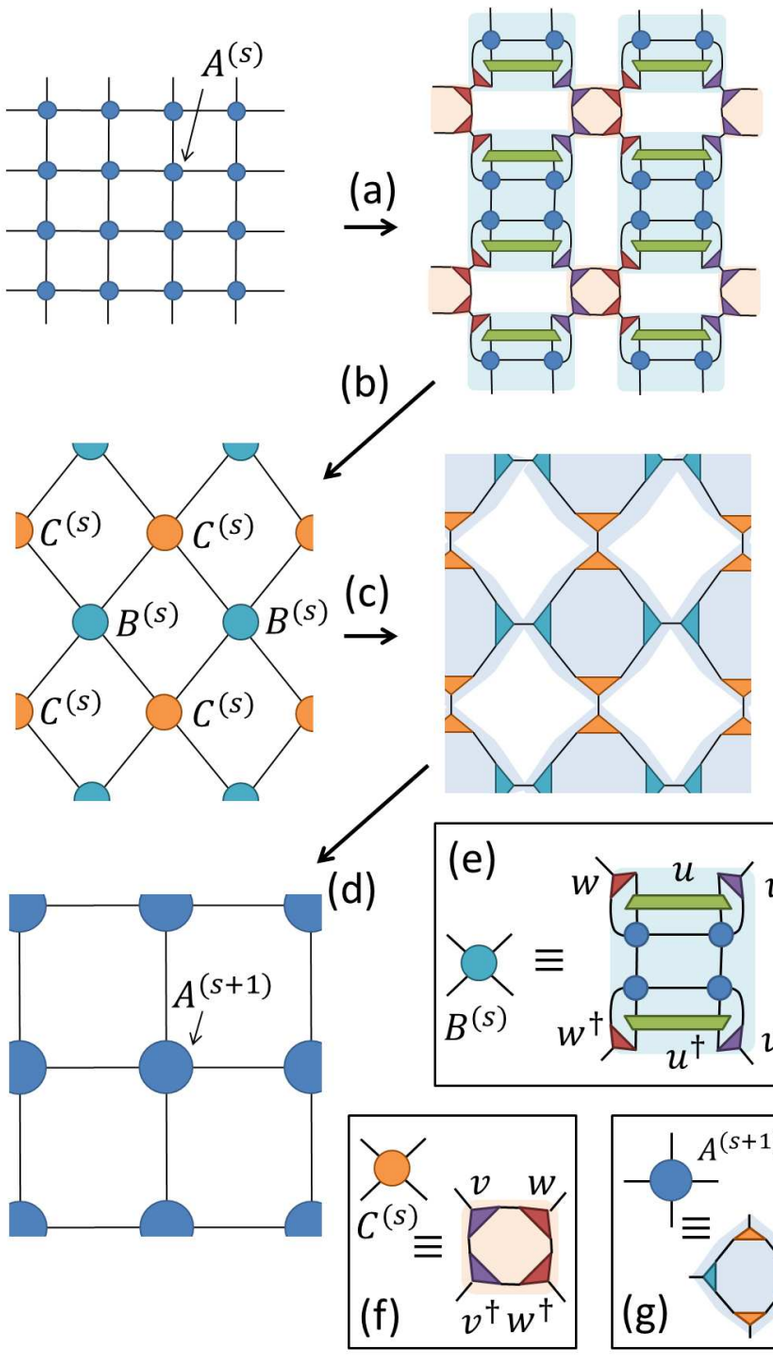
Remaining problem in TEFR

- TEFR works well far from the critical point.
  - Because it can remove CDL structure.
- In the vicinity of the critical point, the accuracy is still poor.
  - Because the actual entanglement is not necessarily perfect CDL structure.
- In order to improve further, we need to consider the entanglement structure beyond CDL tensor.

Recent progress: Tensor Network Renormalization

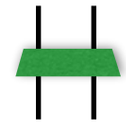
G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405(2015).
arXiv:1412.0732.

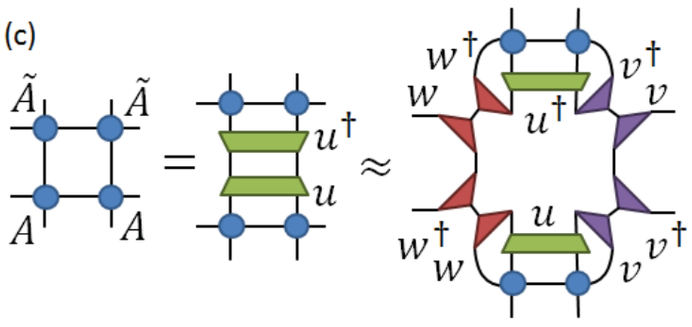
Tensor Network Renormalization



Point of TNR

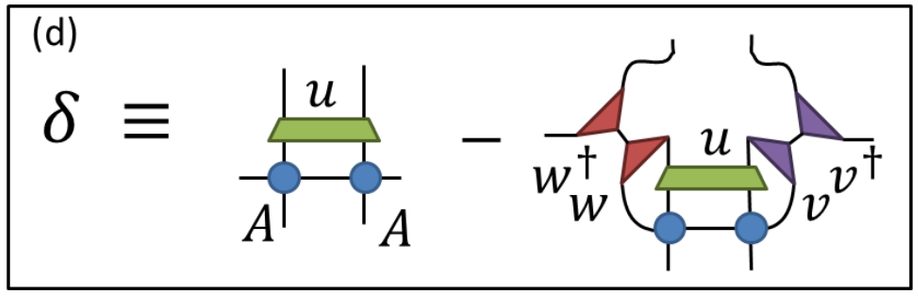
Use of a disentangler(Unitary tensor)





It can remove short range entanglement efficiently.
(Not only CDL)

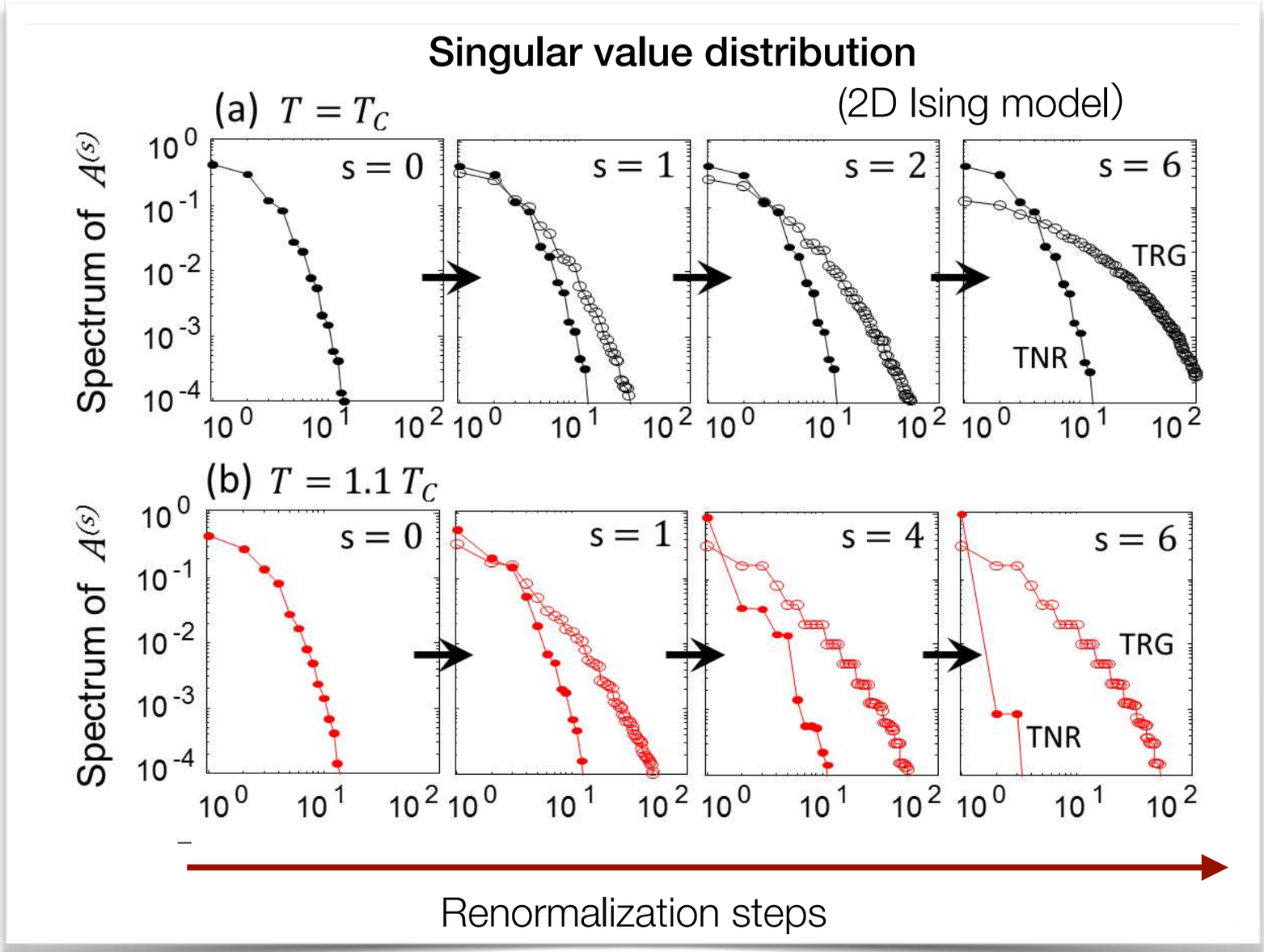
Approximation by using two-tensor cluster:

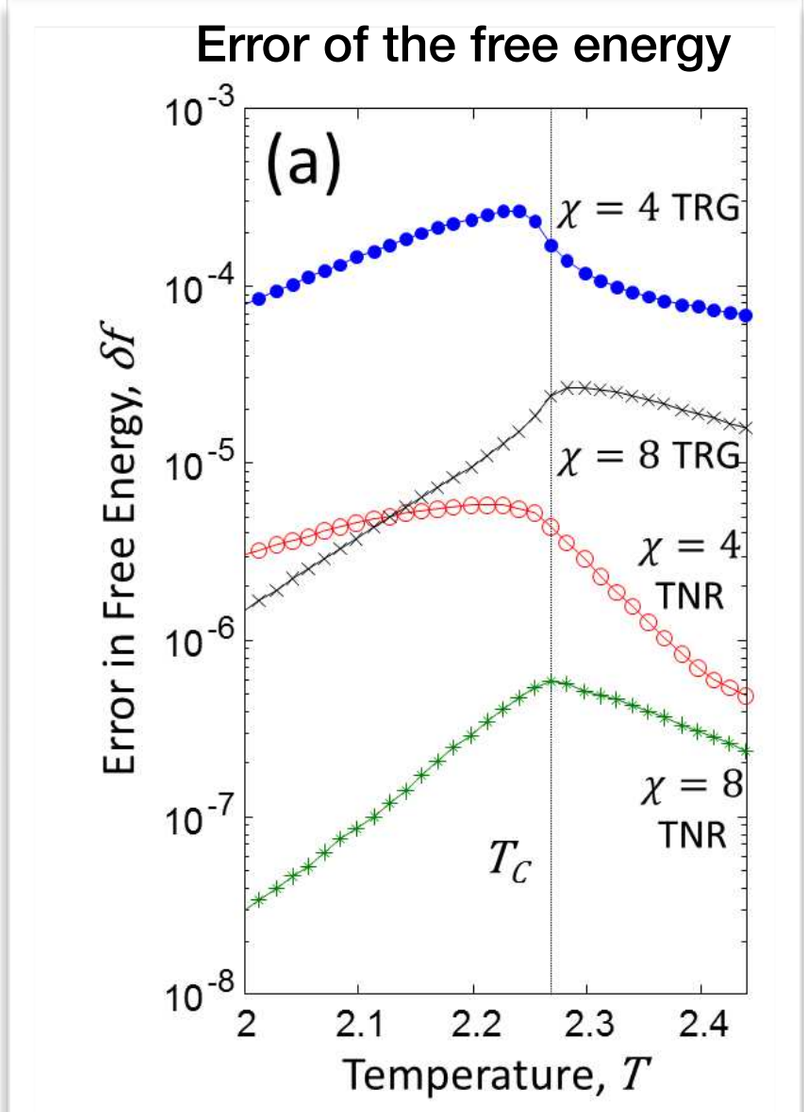


Better accuracy than the simple SVD of single tensor

Power of TNR

G. Evenbly and G. Vidal,Phys. Rev. Lett. **115**, 180405 (2015)
,arXiv: 1412.0732v2 (free energy).





In TNR:
- The singular value distribution is narrower than that of TRG.
- It is almost unchanged at Tc.
  - Indicating scale invariance of the critical system.

Interesting topics in tensor network renormalization

- Try to find efficient algorithm to remove "short range" entanglement
  - TNR, Loop-TNR, GILT, Gauge fixing

TNR: G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015)
Loop-TNR: S. Yang, Z.-C. Gu and , X.-G. Wen, Phys. Rev. Lett. **118**, 110504 (2017)
GILT: M. Hauru, C. Delcamp. S. Mizera Phys. Rev. B **97**, 045111 (2018)
Gauge fixing: G. Evenbly, Phys. Rev. B **98**, 085155 (2018)

- Application to lattice QCD
  - TRG with Grassmann algebra

Z.-C. Gu, F. Verstraete, and X.-G. Wen, arXiv:1004.2563
S. Takeda, and Y. Yoshimura PTEP **2015**, 043B1 (2015).

- Property at the criticality
  - Relation between TNR and MERA
  - Relation to Conformal invariance

G. Evenbly and G. Vidal,Phys. Rev. Lett. **115**, 200401 (2015)
G. Evenbly, Phys. Rev. B **95**, 045117 (2017)

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

情報のエンタングルメントと行列積表現

Outline

- Outline of tensor network decomposition
- Entanglement
  - Schmidt decomposition
  - Entanglement entropy and its area law
- Matrix product states
  - Matrix product states (MPS)
  - Canonical form
  - infinite MPS

Outline of tensor network decomposition

Classification of Information Compression by Memory Costs

Linear algebra for huge data: $\vec{v} \in \mathbb{C}^M$

(1) A matrix can be stored

Required memory~$O(M^2)$

(2) Although a matrix cannot be stored, vectors can be stored

Required memory~$O(M)$

(3) A vector cannot be stored

Required memory $\ll O(M)$

We try to approximate a vector in a compact form.

$M \sim a^N$ → Memory ~ $O(N^x)$

Exponential **Polynomial**

N :problem size (e.g. system size)

When we efficiently compress a vector?

$$\vec{v} = \sum_{i=1}^{M} C_i \vec{e}_i \qquad \vec{v} \in \mathbb{C}^M$$

If we can find a basis where the coefficients have a structure (correlation).

(1) Almost all C_i are zero (or very small).

➡ We store only a few finite elements $\{(i, C_i)\}$

E.g.

Fourier transformation $\vec{v} = \sum_{k=1}^{M} D_k \vec{f}_k$

If we can neglect larger wave numbers, we can efficiently approximate the vector with smaller number of coefficients.

Classical state $|\Psi\rangle = |01011\ldots00\rangle$

In this case, we know that only a specific C_i is non-zero.

We need only an integer corresponding to the non-zero element.

When we efficiently compress a vector?

$$\vec{v} = \sum_{i=1}^{M} C_i \vec{e}_i \qquad \vec{v} \in \mathbb{C}^M$$

(2) All of C_i are not necessarily independent.

→ We store **"structure"** and **"independent elements"**.

$$\{(i, C_i)\}$$

E.g. Product state ("generalized" classical state)

A vector is decomposed into product of small vectors.

$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

e.g.

$$|\phi_1\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\phi_1\rangle = |01\rangle - |10\rangle$$

structure: **"product state"**

independent elements: **small vectors**

Tensor network decomposition of a vector

Target:

Exponentially large Hilbert space

$$\vec{v} \in \mathbb{C}^M \quad \text{with} \quad M \sim a^N$$

+

Total Hilbert space is decomposed as a product of "local" Hilbert space.

$$\mathbb{C}^M = \mathbb{C}^a \otimes \mathbb{C}^a \otimes \cdots \mathbb{C}^a$$

Tensor network decomposition

$$v_i = v_{i_1,i_2,\ldots,i_N} = \sum_{\{x\}} T^{(1)}[i_1]_{x_1,x_2,\ldots} T^{(2)}[i_2]_{x_1,x_3,\ldots} \cdots T^{(N)}[i_N]_{x_3,x_{100},\ldots}$$

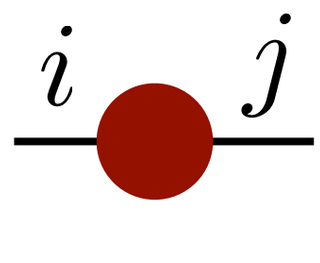
$i_n = 0, 1, \ldots, a-1$: index of local Hilbert space

$T[i]_{x_1,x_2,\ldots}$: local tensor for "state" i

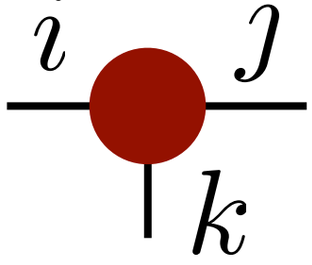
Graphical representations for tensor network

- Vector $\vec{v} : v_i$

- Matrix $M : M_{i,j}$

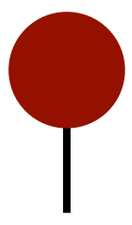


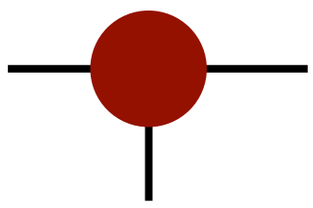
- Tensor $T : T_{i,j,k}$



*** n-rank tensor = n-leg object**

When indices are not presented in a graph, it represent a tensor itself.

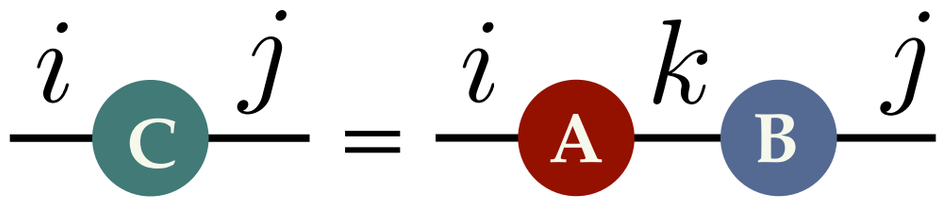
$\vec{v} =$ 

$T =$ 

Graphical representations for tensor network

Matrix product

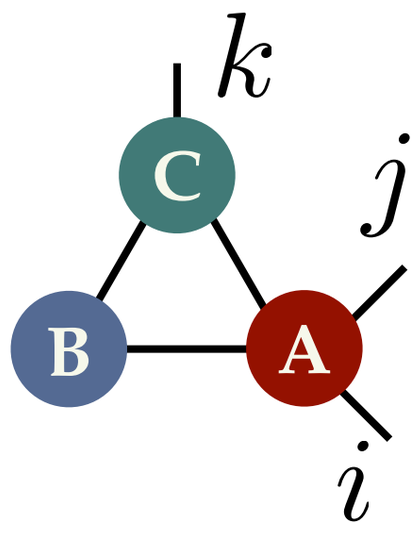
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$



$$C = AB$$

Generalization to tensors

$$\sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$

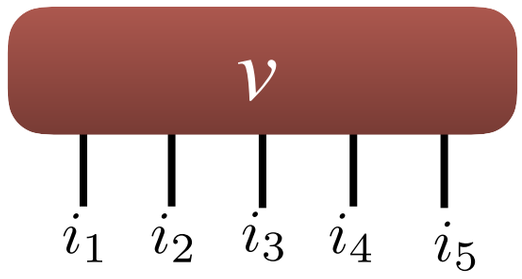


Contraction of a network = Calculation of a lot of multiplications

（縮約）

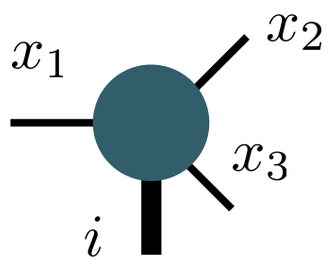
Graph for a tensor network decomposition

- Vector v_{i_1,i_2,i_3,i_4,i_5}



*Vector looks like a tensor

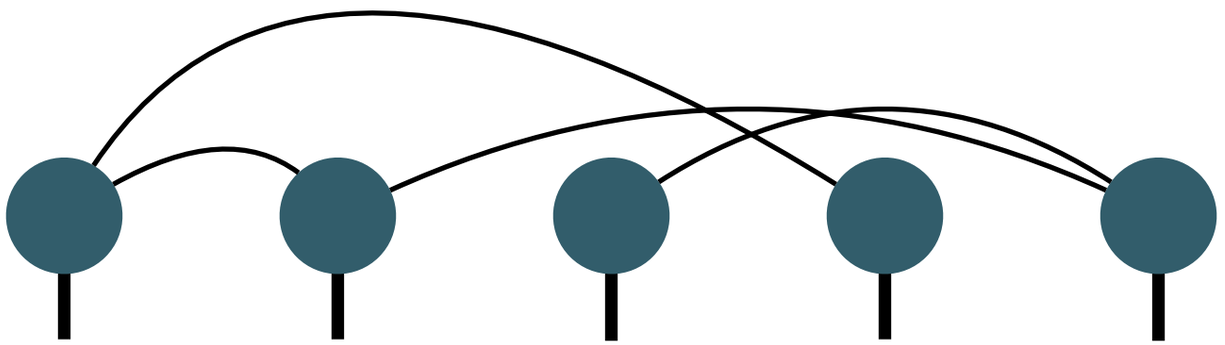
- Tensor $T[i]_{x_1,x_2,x_3}$



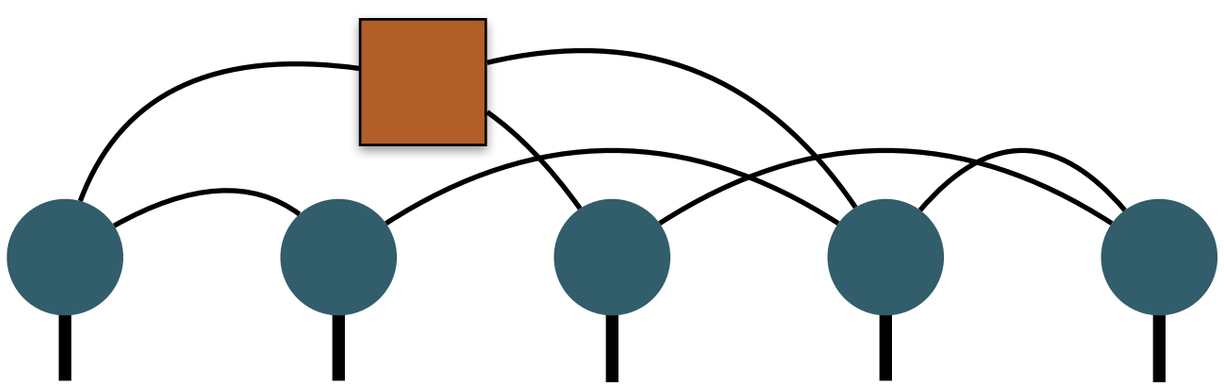
*We treat i as an index of the tensor.

Tensor network decomposition

$\vec{v} =$



$\vec{w} =$



*We can consider tensors independent on i.

Entanglement（エンタングルメント）

N-qubit system (S=1/2 quantum spin system)

Example vector: Wave function of N-qubit systems

takes two states $|0\rangle, |1\rangle$ $(|\uparrow\rangle, |\downarrow\rangle)$

$$|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_N\rangle$$

$$= \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$$

Coefficients = vector: $\vec{\Psi} \in \mathbb{C}^{2^N}$

* Inner product: $\langle \Phi | \Psi \rangle = \vec{\Phi}^* \cdot \vec{\Psi}$

Schmidt decomposition

General vector: $\vec{x} \in \boldsymbol{V}_1 \otimes \boldsymbol{V}_2$ $\quad \dim \boldsymbol{V}_1 = n_1, \dim \boldsymbol{V}_2 = n_2$ $(n_1 \geq n_2)$

Schmidt decomposition

$$\vec{x} = \sum_{i=1}^{n_2} \lambda_i \vec{u}_i \otimes \vec{v}_i$$

Orthonormal vectors $\quad \{\vec{u}_1, \vec{u}_2, \ldots \vec{u}_{n_1}\} \in \boldsymbol{V}_1$, $\{\vec{v}_1, \vec{v}_2, \ldots \vec{v}_{n_2}\} \in \boldsymbol{V}_2$

Schmidt coefficient $\quad \lambda_i \geq 0$

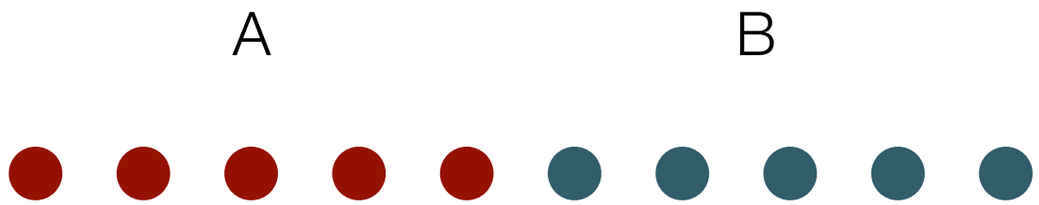
Schmidt decomposition is unique.

Schmidt decomposition for wave function

Wave function: $|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$

Schmidt decomposition

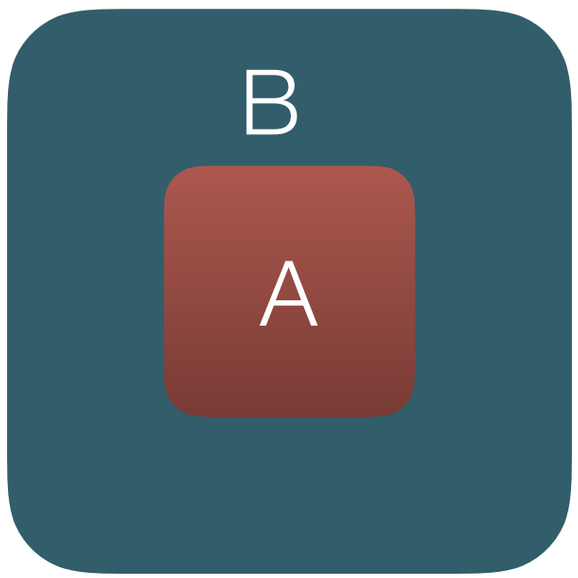
Divide system into two parts, A and B:



General wave function can be represented by a superposition of orthonormal basis set.

$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle = \sum_i \lambda_i |\alpha_i\rangle \otimes |\beta_i\rangle$$

$$M_{i,j} \equiv \Psi_{\underbrace{(i_1,\ldots)}_{\text{A}},\underbrace{(\ldots,i_N)}_{\text{B}}} \qquad |A_i\rangle = |i_1, i_2, \ldots\rangle \qquad |B_j\rangle = |\ldots, i_{N-1}, i_N\rangle$$



Orthonormal basis: $\langle \alpha_i | \alpha_j \rangle = \langle \beta_i | \beta_j \rangle = \delta_{i,j}$

Schmidt coefficient: $\lambda_i \geq 0$

Partial trace and reduced density matrix

For $\vec{x} \in \boldsymbol{V}_1 \otimes \boldsymbol{V}_2$ $\quad \dim \boldsymbol{V}_1 = n_1, \dim \boldsymbol{V}_2 = n_2$ $\quad |\vec{x}| = 1$

Density matrix: $\rho \equiv \vec{x}\vec{x}^\dagger \quad (\rho_{ij} = x_i x_j^*)$

(密度行列) $(\rho = |x\rangle\langle x|)$ $\quad$ *Note: $\mathrm{rank}\ \rho = 1$

Orthonormal basis: $\{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_{n_1}\} \in \boldsymbol{V}_1 \quad \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_{n_2}\} \in \boldsymbol{V}_2$

→ Basis for $\vec{x}$: $\vec{g}_{i_1,i_2} = \vec{e}_{i_1} \otimes \vec{f}_{i_2}$

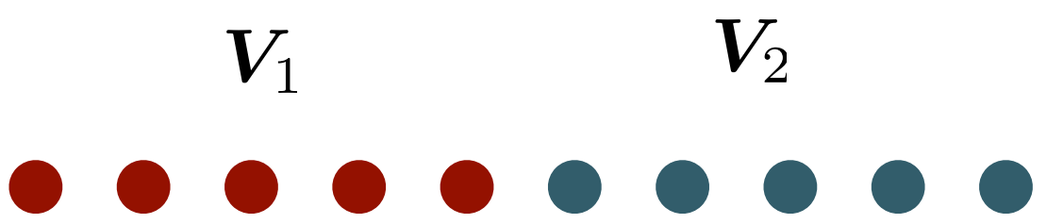
Index: $i = (i_1, i_2)$

Reduced Density matrix:

(縮約密度行列)

$\rho_{\boldsymbol{V}_1} \equiv \mathrm{Tr}_{\boldsymbol{V}_2}\ \rho$: a positive-semidefinite square matrix in $\boldsymbol{V}_1$

$$(\rho_{\boldsymbol{V}_1})_{i_1,j_1} = \sum_{i_2} \rho_{(i_1,i_2),(j_1,i_2)}$$



Relation between SVD and Schmidt decomposition

Singular value decomposition (SVD):

For a K × L matrix M,

$$M_{i,j} = \sum_m U_{i,m} \lambda_m V^\dagger_{m,j}$$

Singular values: $\lambda_m \geq 0$

Singular vectors:

$$\sum_i U_{i,m} U^\dagger_{m,j} = \delta_i, j$$

$$\sum_i V_{i,m} V^\dagger_{m,j} = \delta_i, j$$

Relation to the Schmidt decomposition:

$$|\Psi\rangle = \sum_{i,j} M_{i,j} |A_i\rangle \otimes |B_j\rangle = \sum_m \lambda_m |\alpha_m\rangle \otimes |\beta_m\rangle$$

$$|\alpha_m\rangle = \sum_i U_{i,m} |A_i\rangle$$

$$|\beta_m\rangle = \sum_j V^\dagger_{m,j} |B_j\rangle$$

$$\Rightarrow \quad \langle \alpha_i | \alpha_j \rangle = \langle \beta_i | \beta_j \rangle = \delta_{i,j}$$

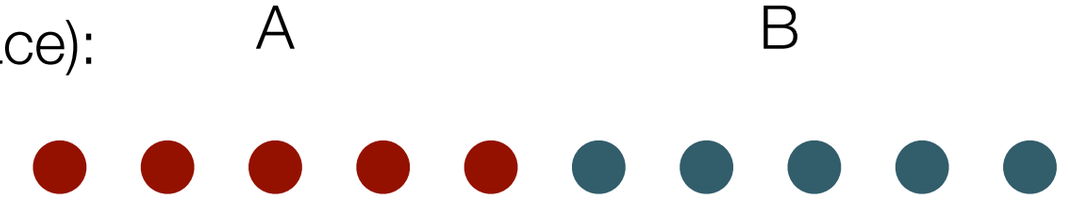
By using SVD, we can perform Schmidt decomposition.
(and can calculate entanglement entropy.)

Entanglement entropy

Entanglement entropy:

Reduced density matrix of a sub system (sub space):

$$\rho_A = \mathrm{Tr}_B |\Psi\rangle\langle\Psi|$$



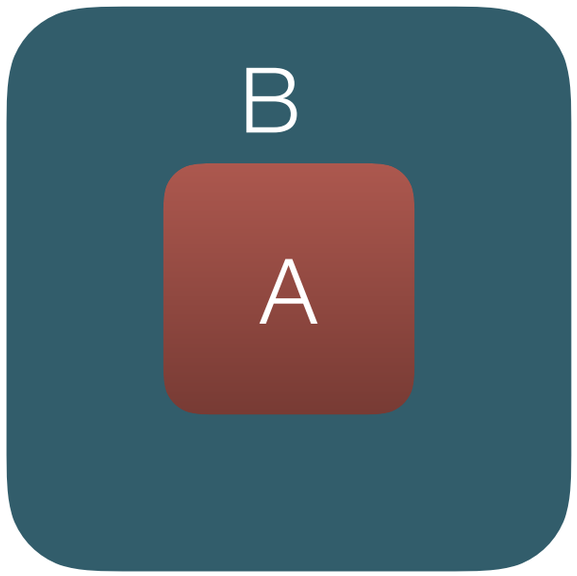
Entanglement entropy = von Neumann entropy of ρ_A

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right)$$

Schmidt decomposition $|\Psi\rangle = \sum_i \lambda_i |\alpha_i\rangle \otimes |\beta_i\rangle$

$$\rho_A = \sum_i \lambda_i^2 |\alpha_i\rangle\langle\alpha_i| \qquad \text{(*Exercise)}$$

$$S = -\sum_i \lambda_i^2 \log \lambda_i^2$$

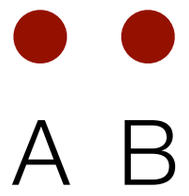


Entanglement entropy is calculated through
the spectrum of Schmidt coefficients

Intuition for EE: two s=1/2 spins

1. $|\Psi\rangle = |\uparrow\rangle \otimes |\downarrow\rangle$

   A product state → $\lambda = 1,\ S = 0$



Product state : S=0

2. $|\Psi\rangle = \frac{1}{2}(|\uparrow\rangle - |\downarrow\rangle) \otimes (|\uparrow\rangle - |\downarrow\rangle)$

   Another product state → $\lambda = 1\ ,\ S = 0$

3. $|\Psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle \otimes |\downarrow\rangle - |\downarrow\rangle \otimes |\uparrow\rangle)$

   Spin singlet → $\lambda_1 = \lambda_2 = \frac{1}{\sqrt{2}}\ ,\ \ S = \log 2$ **Maximally entangled State**

4. $|\Psi\rangle = \left(x|\uparrow\rangle \otimes |\downarrow\rangle + \sqrt{1-x^2}|\downarrow\rangle \otimes |\uparrow\rangle\right)$

   Complicated state → $\lambda_1 = |x|, \lambda_2 = \sqrt{1-x^2}$

$$S = x^2 \log x^2 + \sqrt{1-x^2}\log(1-x^2)$$

Large entanglement entropy ~ Large correlation between two parts

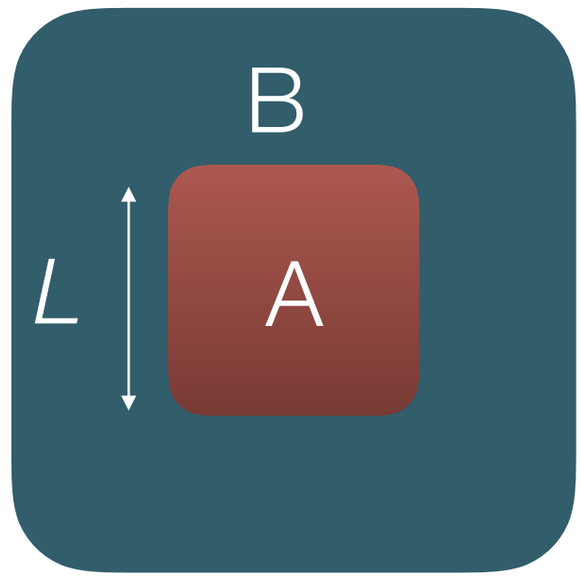
Area law of the entanglement entropy in physics

General wave functions:

EE is proportional to its **volume** (**# of qubits**).

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right) \propto L^d$$

(c.f. random vector)



Ground state wave functions:

For a lot of ground states, EE is proportional to its area.

J. Eisert, M. Cramer, and M. B. Plenio, Rev. Mod. Phys, 277, **82** (2010)

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right) \propto L^{d-1}$$

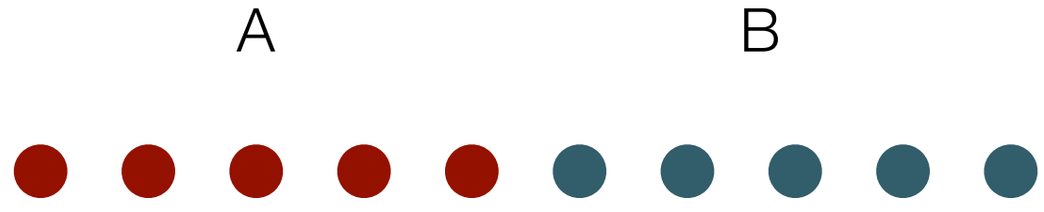
In the case of one-dimensional system:

Gapped ground state for local Hamiltonian

M.B. Hastings, J. Stat. Mech.: Theory Exp. P08024 (2007)

$$S = O(1)$$

Ground state are in a small part of the huge Hilbert space



Exercise: examples of Schmidt decomposition

1-1: Random wave function (Sample code: Ex1-1.py or Ex1-1.ipynb)

- Make a random vector
- SVD it and see singular value spectrum and EE

1-2: Ground state of S=1 Heisenberg chain (Sample code: Ex1-2.py or Ex1-2.ipynb)

$$\mathcal{H} = \sum_i \vec{S}_i \cdot \vec{S}_{i+1}$$

- Calculate GS by diagonalizing Hamiltonian
- SVD it and see singular value spectrum and EE

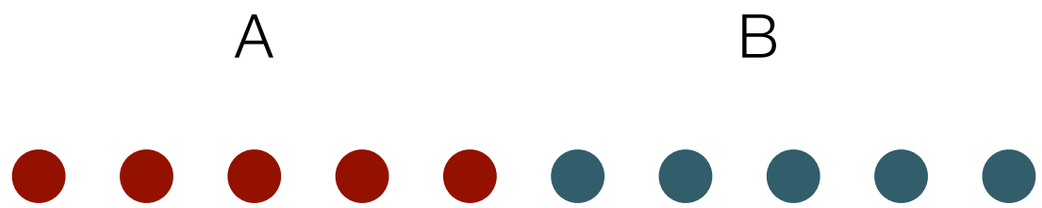
*Note: the ground state of this model is gapped

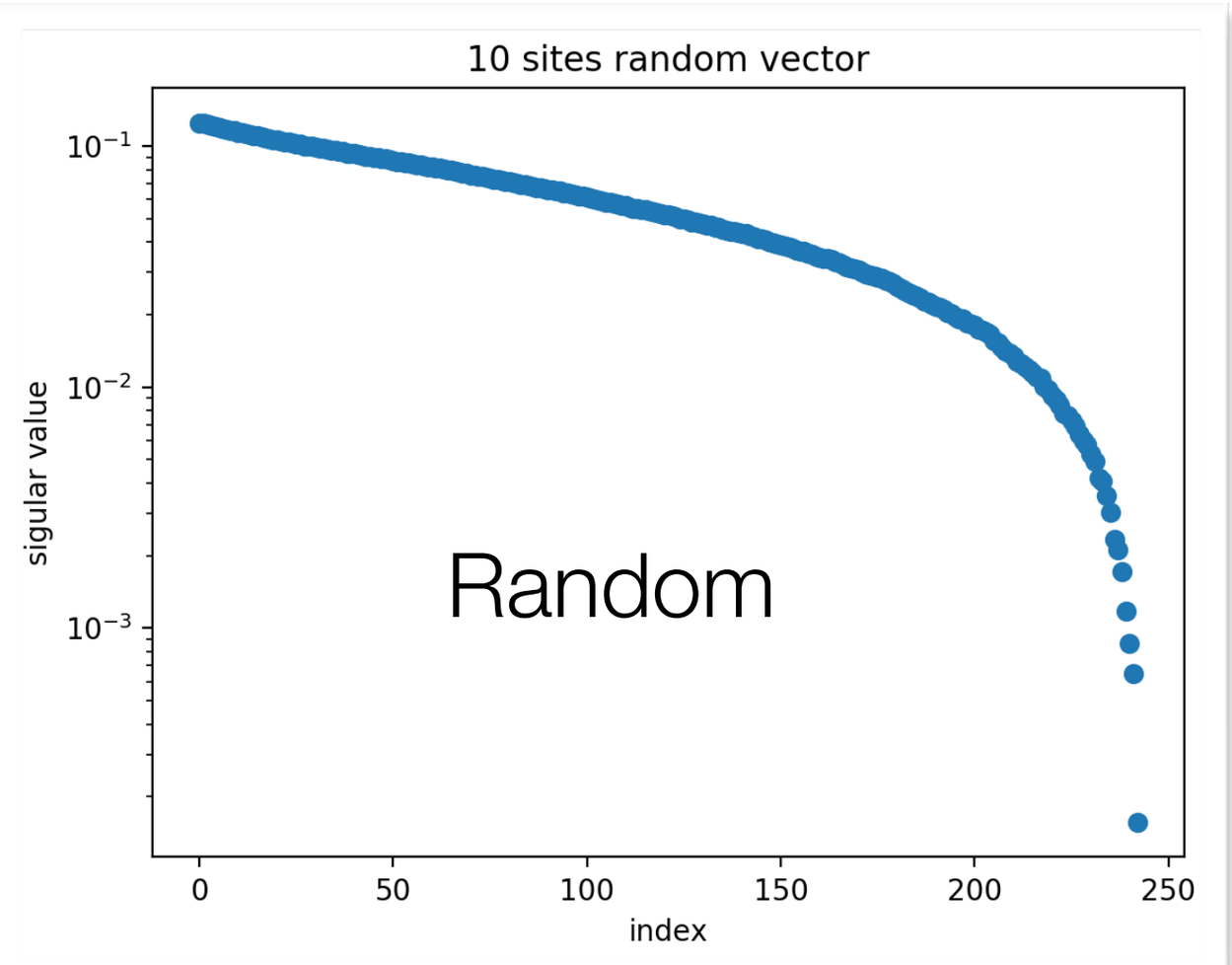
show help: *python Ex1-2.py -h*

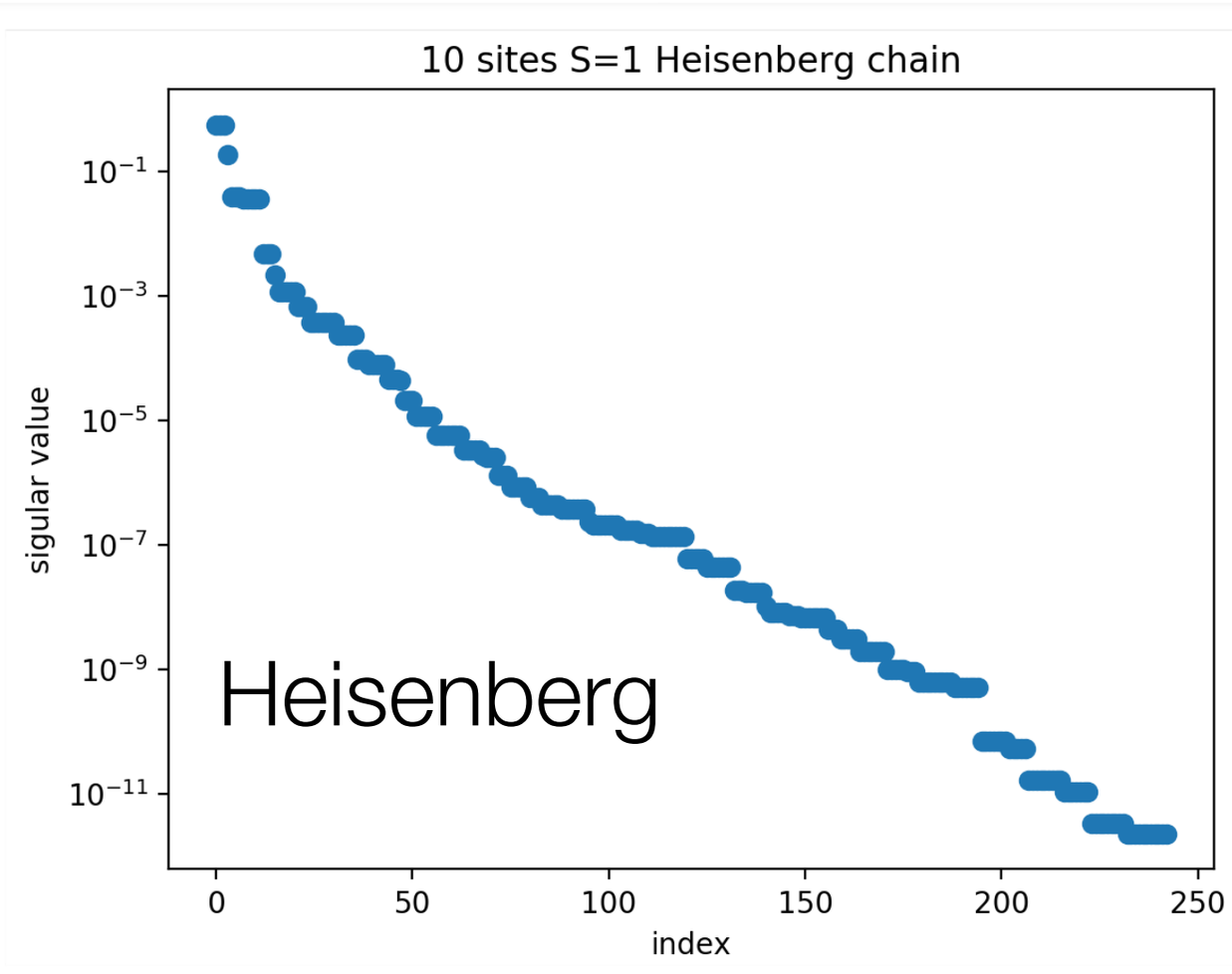
*** Try to simulate different system size "N"**
*** You can simulate other S by changing "m"**

Result: N=10 spectrum

$$\vec{\Psi} \in \mathbb{C}^{3^{10}}$$







Ground state wave function has lower entanglement!

Matrix product states（行列積状態）

Data compression of wave functions (vectors)

General wave function:

$$|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$$

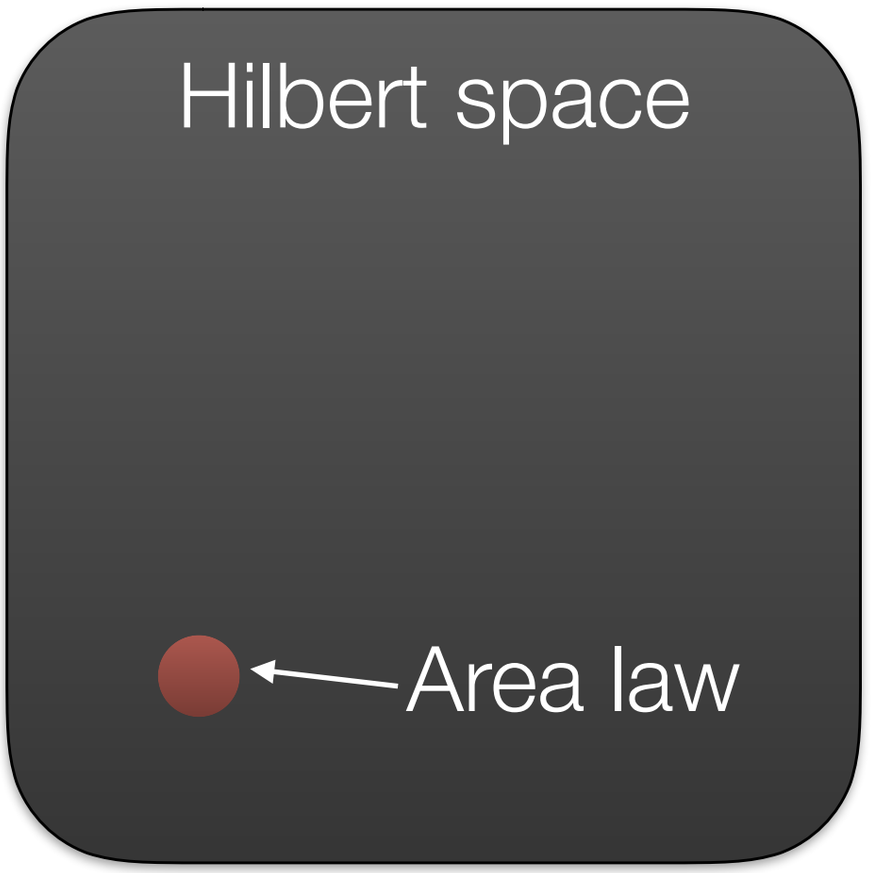
Coefficient vector can represent any points in the Hilbert space.

Ground states satisfy the area law.

In order to represent the ground state accurately,
we might not need all of a^N elements.

Data compression by tensor decomposition:

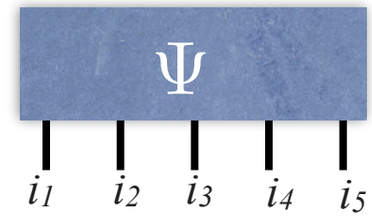
Tensor network states



Tensor network state

G.S. wave function: $|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$

Vector (or N-rank tensor): $\Psi_{i_1 i_2 \ldots i_N}$ =

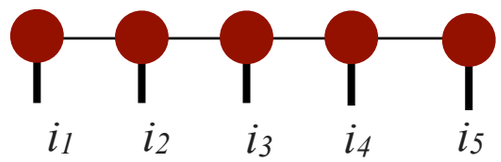


# of Elements = a^N

``Tensor network" decomposition

* Matrix Product State (MPS)

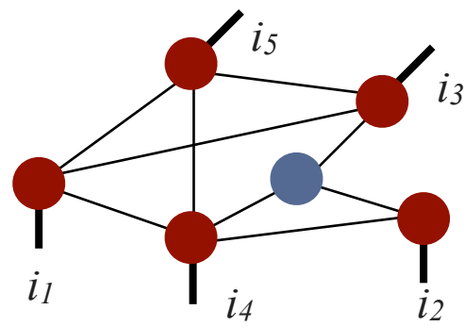
$A_1[i_1] A_2[i_2] \cdots A_N[i_N]$ =



$A[m]$: Matrix for state m

* General network

$\mathrm{Tr} X_1[i_1] X_2[i_2] X_3[i_3] X_4[i_4] X_5[i_5] Y$



X,Y : Tensors

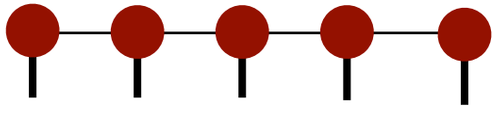
Tr : Tensor network contraction

By choosing a ``good" network, we can express G.S. wave function efficiently.

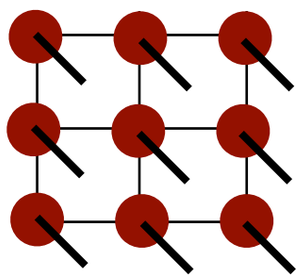
ex. MPS: # of elements $= 2ND^2$ D: dimension of the matrix A

Exponential → Linear *If D does not depend on N…

Examples of TNS

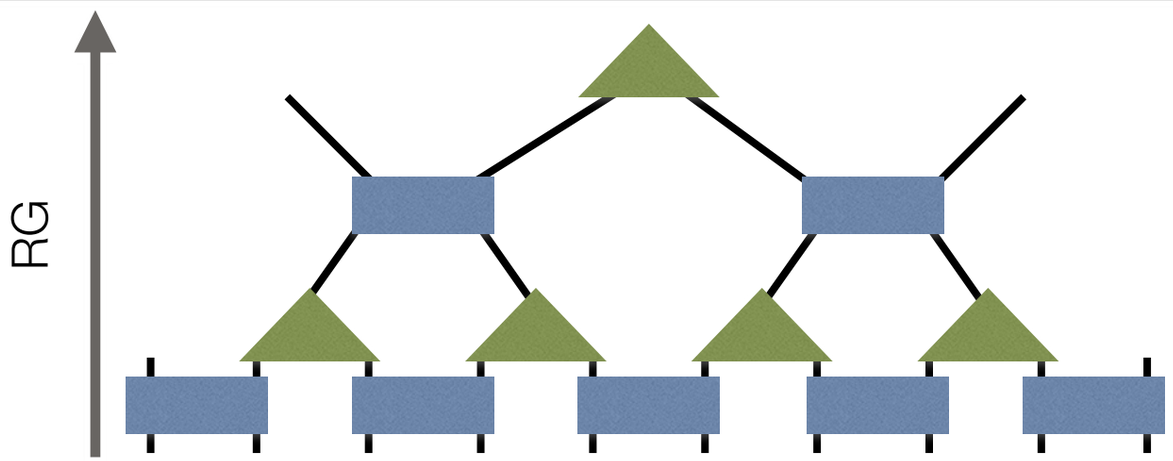
MPS: 

Good for 1-d gapped systems

PEPS, TPS: 

For higher dimensional systems

Extension of MPS

MERA: 

Scale invariant systems

Matrix product state (MPS)

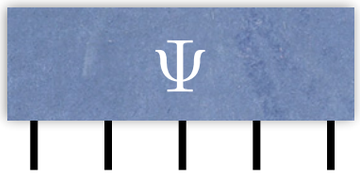
Good reviews:

(U. Schollwöck, Annals. of Physics **326**, 96 (2011))
(R. Orús, Annals. of Physics **349**, 117 (2014))

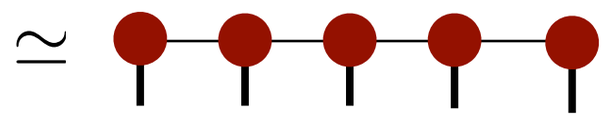
$$|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$$

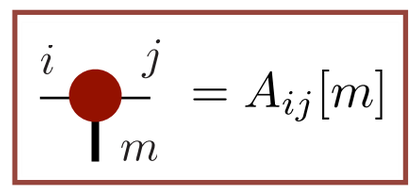
$$\Psi_{i_1 i_2 \ldots i_N} \simeq A_1[i_1] A_2[i_2] \cdots A_N[i_N]$$

$A[i]$: Matrix for state i



MPS





Note:

- MPS is called as "tensor train decomposition" in applied mathematics

(I. V. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011))

- A product state is represented by MPS with 1×1 "Matrix" (scalar)

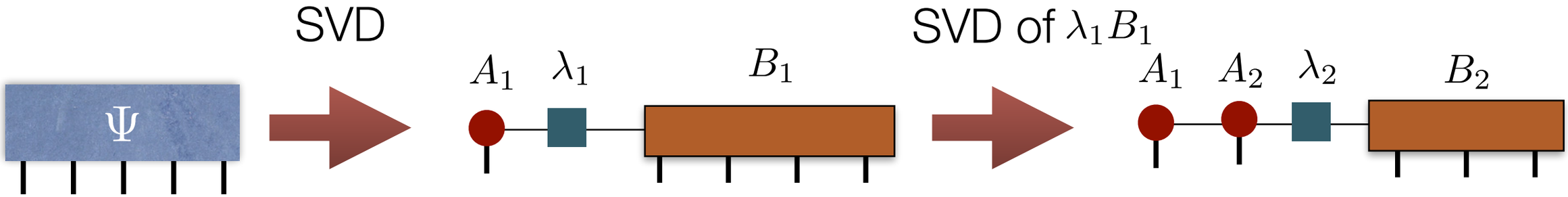
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

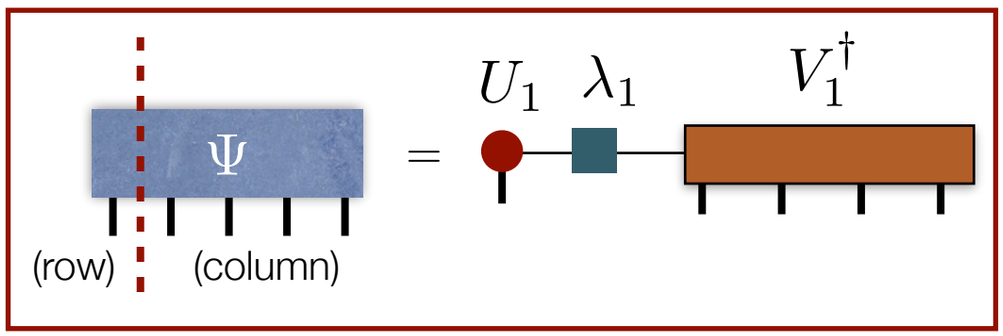
$$\Psi_{i_1 i_2 \ldots i_N} = \phi_1[i_1] \phi_2[i_2] \cdots \phi_N[i_N]$$

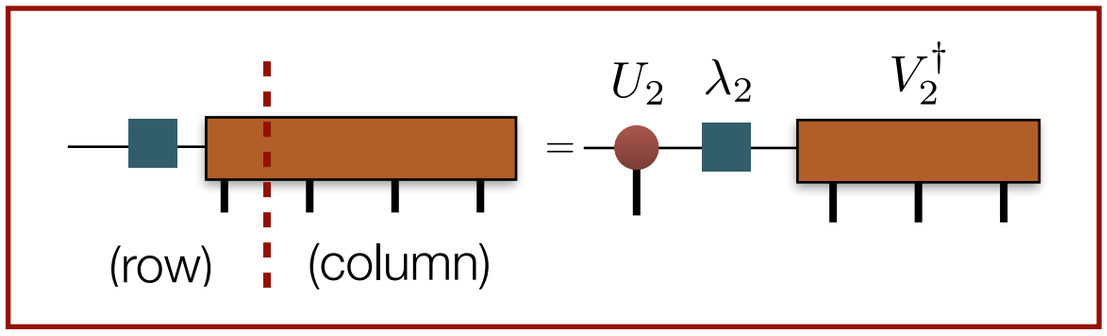
$$\phi_n[i] \equiv \langle i | \phi_i \rangle$$

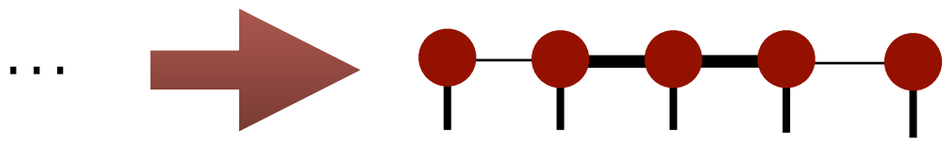
Matrix product state without approximation

General wave function (or vector) can be represented by MPS exactly through successive Schmidt decompositions







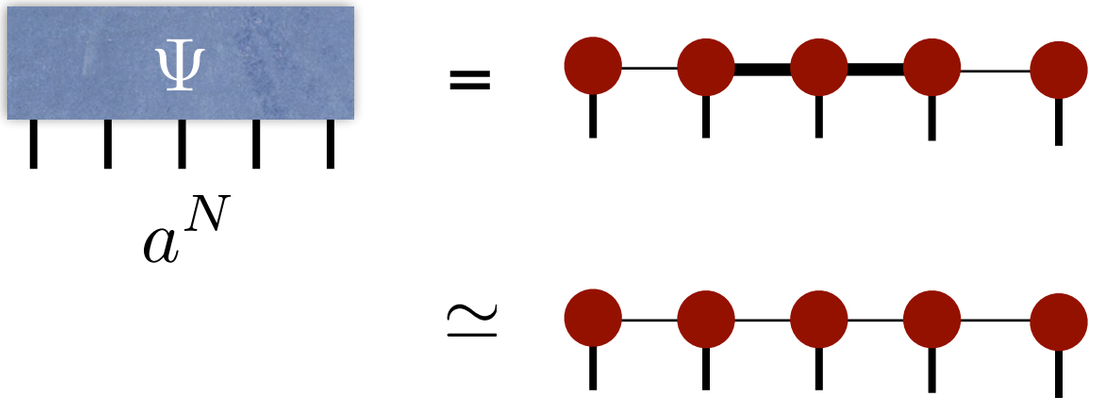


In this construction, the sizes of matrices depend on the position.

Maximum bond dimension = $a^{N/2}$

At this stage, **no data compression.**

Matrix product state: Low rank approximation



Maximum bond dimension = $a^{N/2}$

a^N

Constant bond dimension = χ

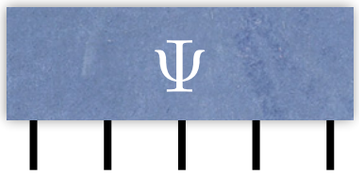
$\propto Na\chi^2$

If the entanglement entropy of the system is **O(1)** (independent of *N*),
matrix size "χ" can be small for accurate approximation.

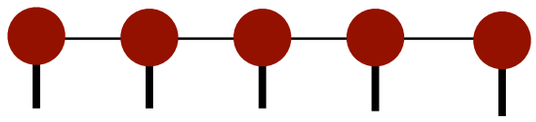
→ MPS is good for gapped 1d systems.

On the other hand, if the EE increases as increase *N*,
"χ" must be increased to keep the same accuracy.

Upper bound of Entanglement entropy

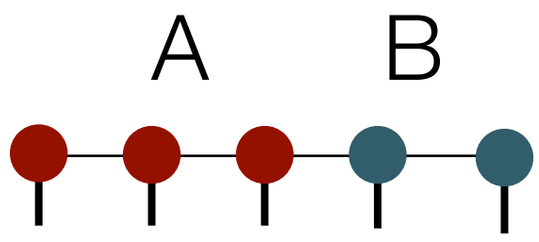


$\simeq$

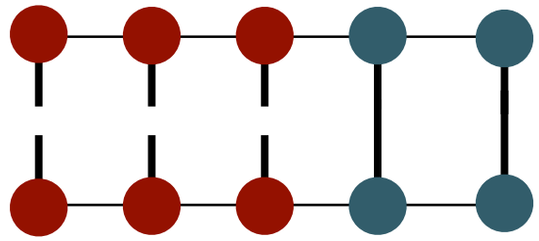


$\equiv |\tilde{\Psi}\rangle$:MPS with χ

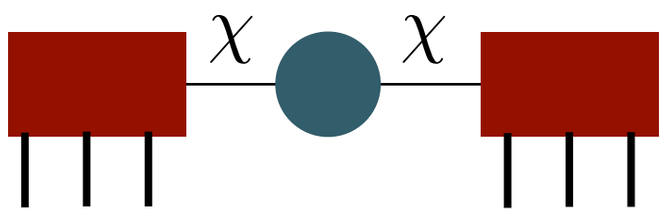
Reduced density matrix of region A:



$$\rho_A = \mathrm{Tr}_B |\tilde{\Psi}\rangle\langle\tilde{\Psi}| =$$



★ Structure of ρ_A:

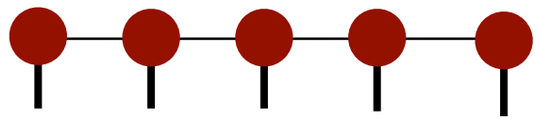


$$\mathrm{rank}\ \rho_A \leq \chi$$

$$S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi$$

Required bond dimension in MPS representation

$$S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi$$



The upper bound is independent of the "length".

length of MPS ⇔ size of the problem

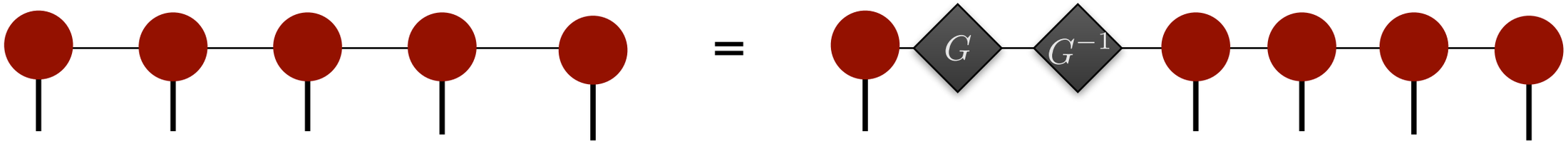
N $\qquad\qquad$ a^N

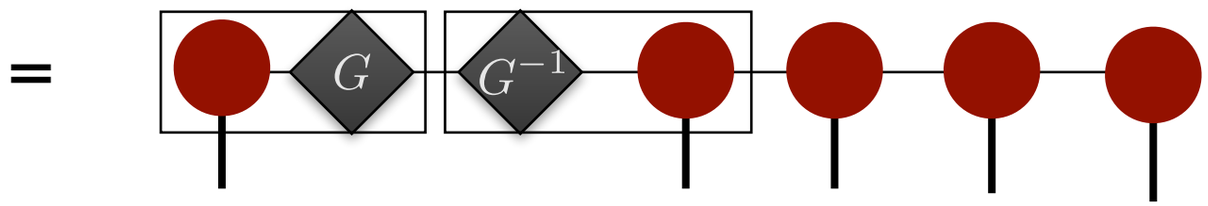
EE of the original vector	Required bond dimension in MPS representation
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

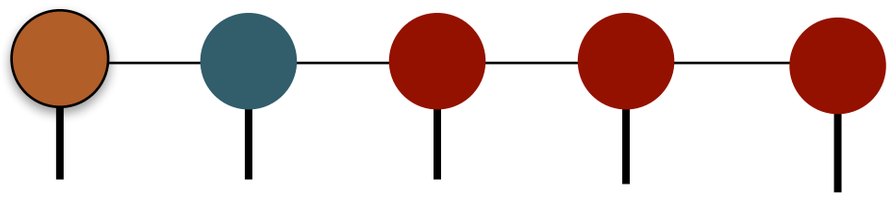
Gauge redundancy of MPS

MPS is not unique: gauge degree of freedom $I = GG^{-1}$

We can insert a pair of matrices GG^{-1} to MPS





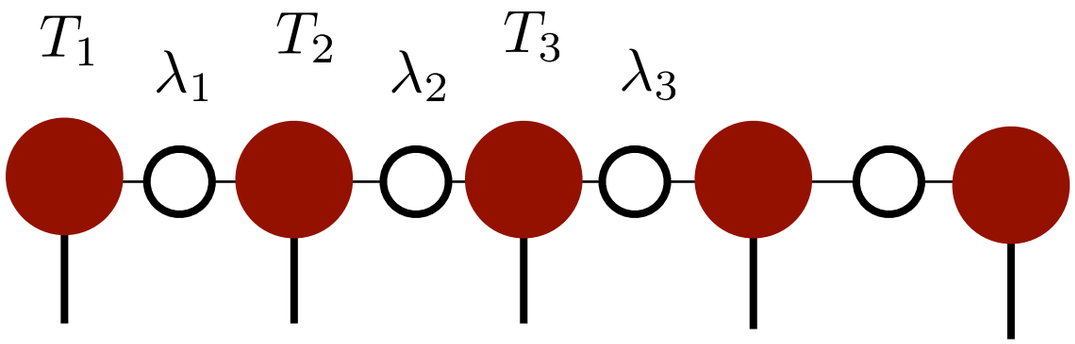


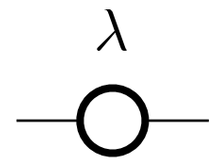
Gauge fix: Canonical form of MPS

Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

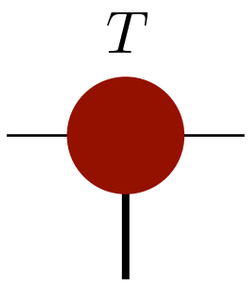
Canonical form of MPS: (Vidal canonical form)

(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003)



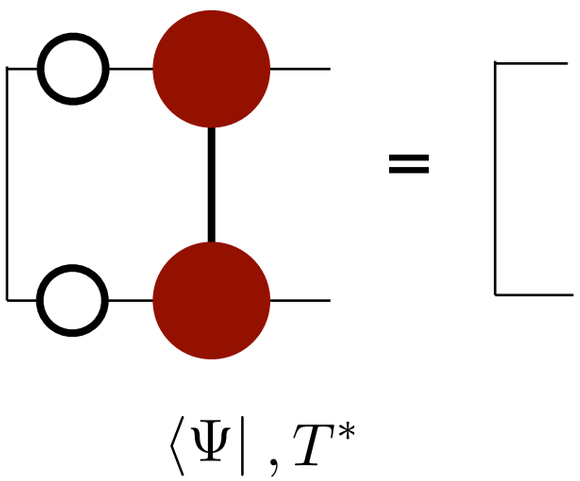


:Diagonal matrix corresponding to Schmidt coefficient

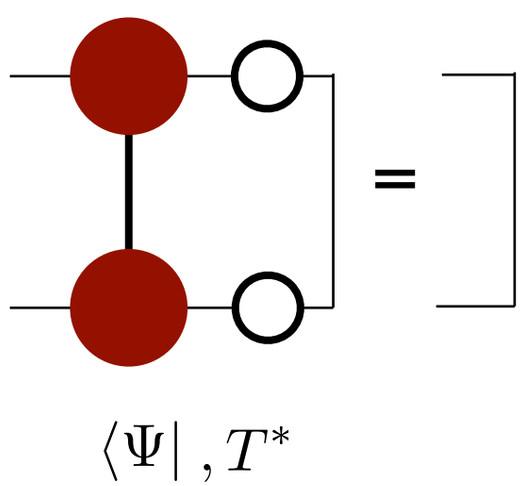


:Virtual indices corresponding to Schmidt basis

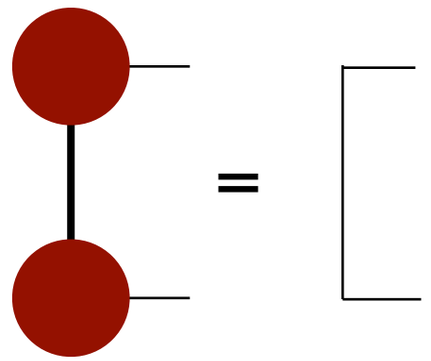
Left canonical condition:



Right canonical condition:



(Boundary)

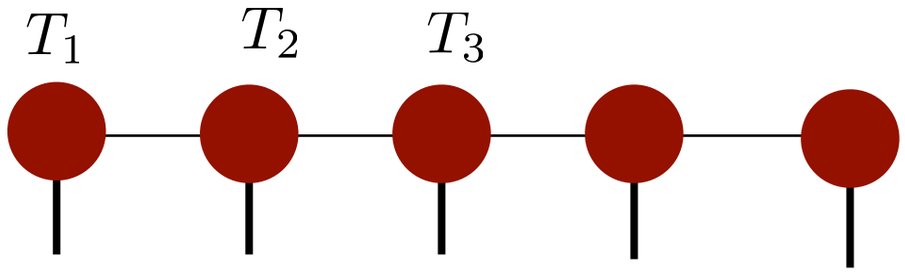


Canonical forms: Left and Right canonical forms

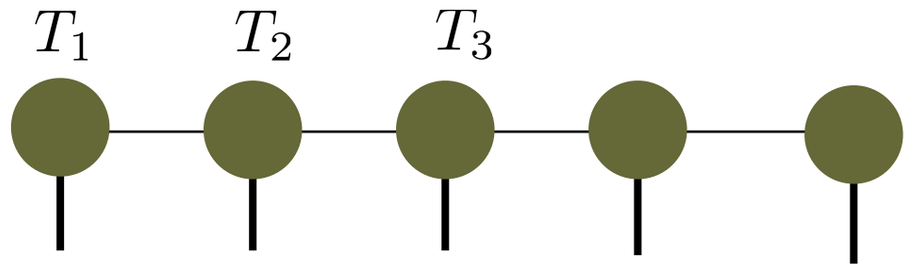
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Other "canonical" forms of MPS

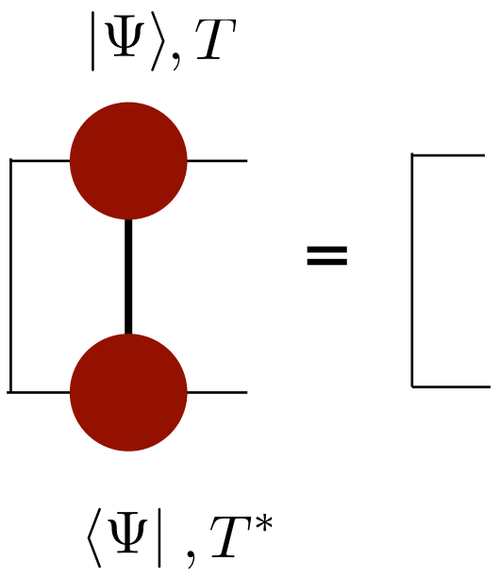
Left canonical form:

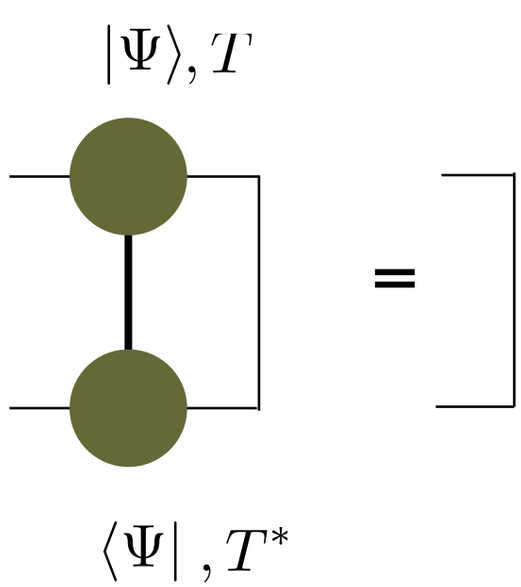


Right canonical form:



Satisfies (at least) left or canonical condition:

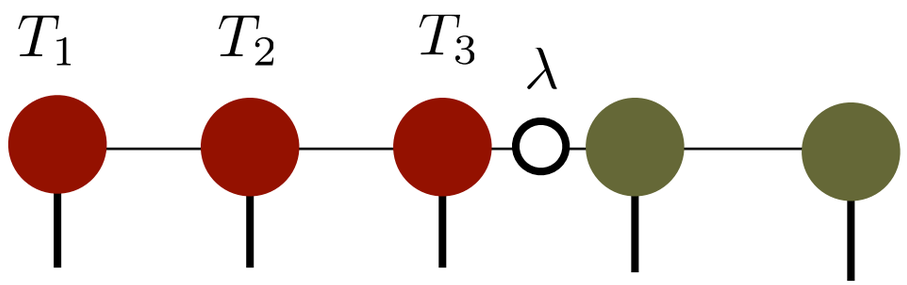




Canonical forms: Mixed canonical forms

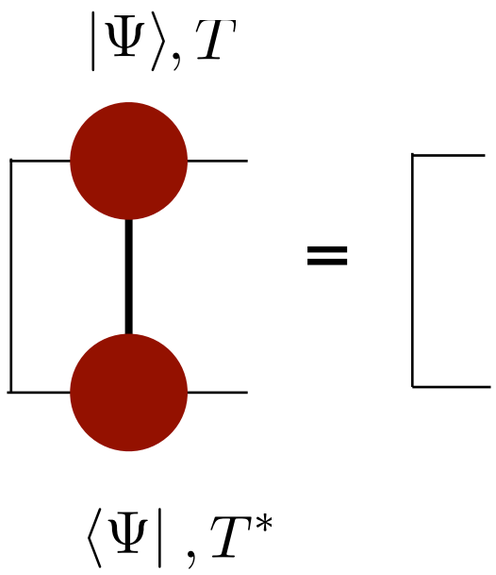
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Mixed canonical form:

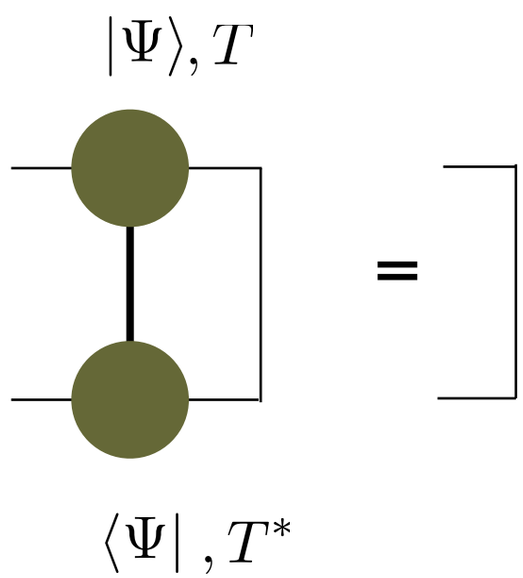


λ is identical with the Schmidt coefficient.

Left canonical condition:



Right canonical condition:

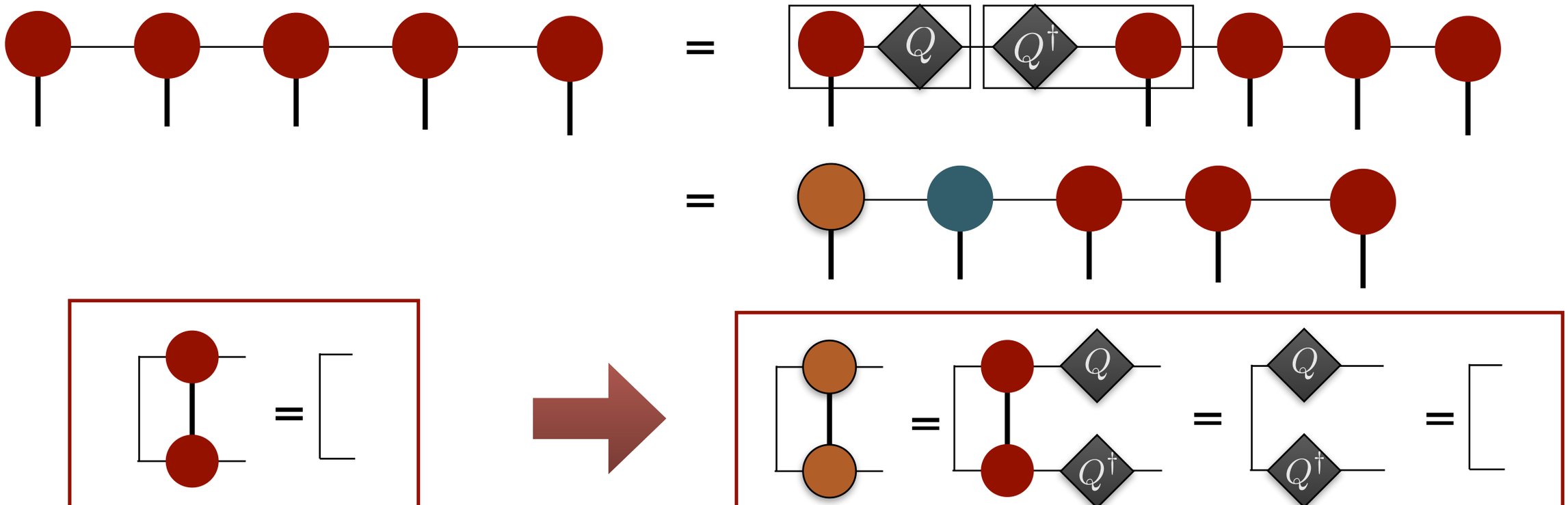


Canonical forms: Note

- Vidal canonical form is unique, up to trivial unitary transformation to virtual indices which keep the same diagonal matrix structure (Schmidt coefficients).

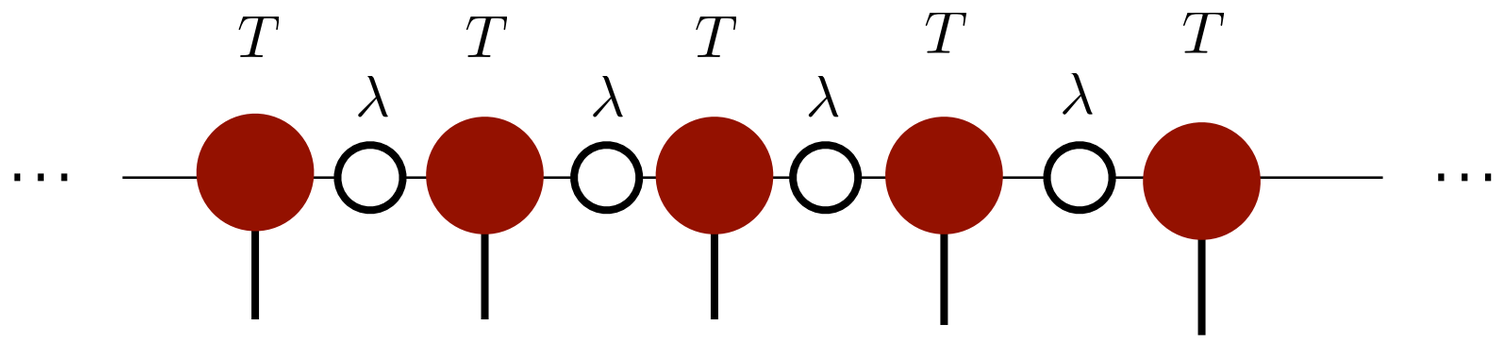
- Left, right and mixed canonical form is "not unique". Under general unitary transformation to virtual indices, it remains to satisfy the canonical condition

$$QQ^\dagger = Q^\dagger Q = I$$



MPS for infinite chains

By using MPS, we can write the wave function of
a translationally invariant **infinite chain**



Infinite MPS (iMPS) is made by repeating T and λ infinitely.

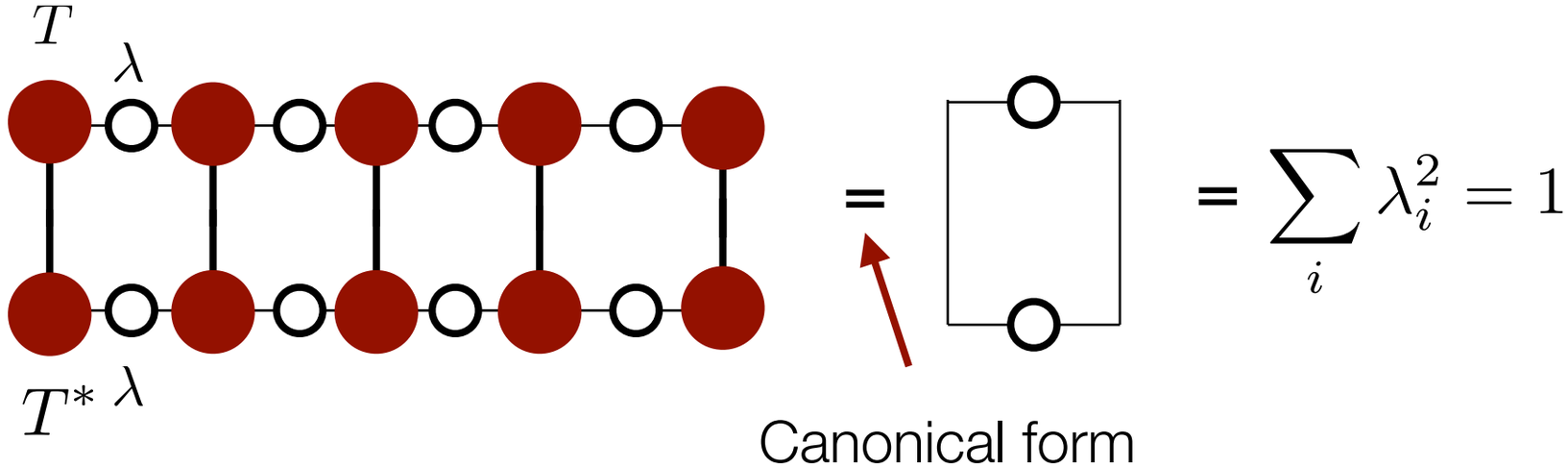
Translationally invariant system → T and λ are independent of positions!

* Infinite MPS can be accurate when the EE satisfies the 1d area low (S~O(1)).

If the EE increases as increase the system size,
we may need infinitely large χ for infinite system.

Calculation of expectation value

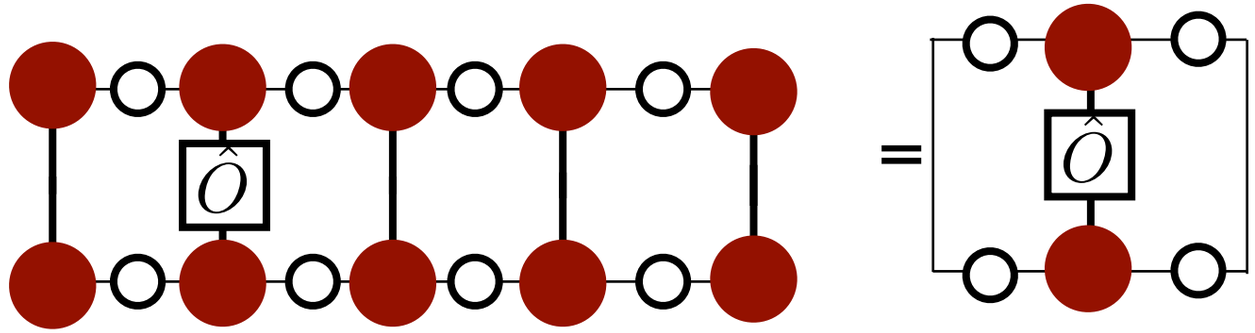
$\langle \Psi | \Psi \rangle =$



Canonical form

$= \sum_i \lambda_i^2 = 1$

$\langle \Psi | \hat{O} | \Psi \rangle =$

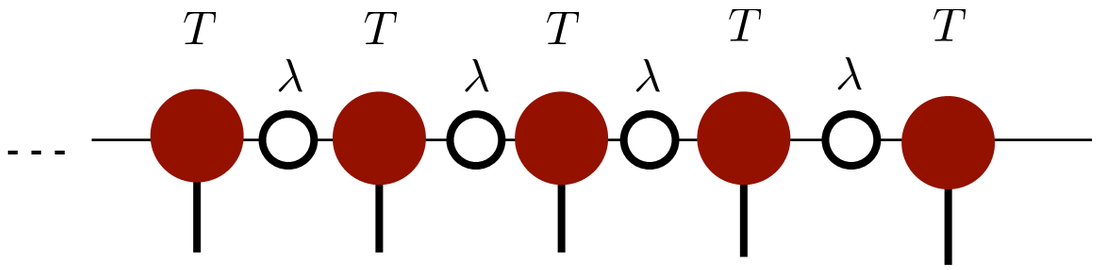


For iMPS, if it is in the (Vidal) canonical form,
the final graph is identical to the above finite system.

Example of iMPS: AKLT state

S=1 Affleck-Kennedy-Lieb-Tasaki (AKLT) Hamiltonian:

$$\mathcal{H} = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \frac{J}{3} \sum_{\langle i,j \rangle} \left(\vec{S}_i \cdot \vec{S}_j \right)^2 \qquad (J > 0)$$



The ground state of AKLT model:

S=1 spin:

S=1/2 spin

Spin singlet

χ=2 iMPS: (U. Schollwock, Annals. of Physics **326**, 96 (2011))

$$T[S_z = 1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$T[S_z = 0] = \sqrt{\frac{2}{3}} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad , \lambda = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$T[S_z = -1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}$$

Exercise 2: Make MPS and approximate it

2-1: Make exact MPS from GS wave function obtained by ED

(We can easily check that the MPS obtained by successive SVD
satisfy the canonical condition.)

Sample code: Ex2-1.py or Ex2-1.ipynb

show help: *python Ex2-1.py -h*

2-2: Approximate the MPS by truncating singular values

- Calculate approximate GS energy and compare it with ED
- Change *chi_max* and see energies

Sample code: Ex2-2.py or Ex2-1.ipynb

show help: *python Ex2-2.py -h*

Requirement for running sample scripts

File: Exercise_No10.zip

Python environment: python2.7 or python3

Modules: numpy, scipy and matplotlib

Usage:

For jupyter notebook, type

jupyter notebook

and select Ex?-?.ipynb .

For python (command line), type

python Ex?-?.py -h

, then you can know how to change the parameters.

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

行列積表現の固有値問題への応用

Outline

- Application to Eigenvalue problem
  (Ground state of quantum many-body systems)
  - Variational algorithm
- Application to time evolution of quantum system
  - TEBD algorithm
- Application to Machine learning

Application to eigenvalue problem

Calculation of minimum (or maximum) eigenvalue

Target vector space:

Exponentially large Hilbert space

$$\vec{v} \in \mathbb{C}^M \quad \text{with} \quad M \sim a^N$$

+

Total Hilbert space is decomposed as a product of "local" Hilbert space.

$$\mathbb{C}^M = \mathbb{C}^a \otimes \mathbb{C}^a \otimes \cdots \mathbb{C}^a$$

Target matrix:

$\mathcal{H}$:Hermitian, square, and sparse

(Typically, only $O(M)$ ($=O(a^N)$) elements are finite.)

Notice:

We consider the situation where
we cannot store $O(M)$ variables in the memory.

Problem:

Find the smallest eigenvalue and its eigenvector

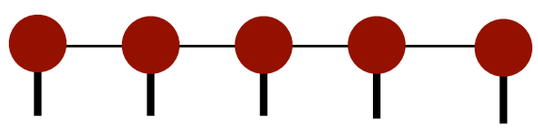
$$\mathcal{H}\vec{v}_0 = E_0\vec{v}_0$$

$$\Rightarrow \min_{\vec{\psi}\in\mathbb{C}^M} \frac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}} \left(= \min_{|\psi\rangle} \frac{\langle\psi|\mathcal{H}|\psi\rangle}{\langle\psi|\psi\rangle}\right)$$

Variational calculation using MPS:

Cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

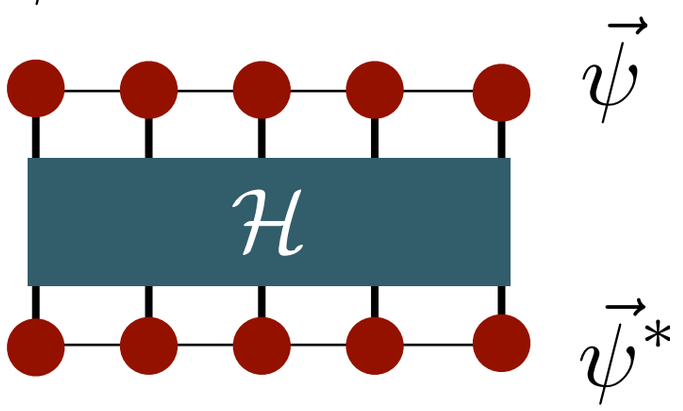
Find the MPS which minimizes F by optimizing matrices in MPS.

$\vec{\psi} =$ 

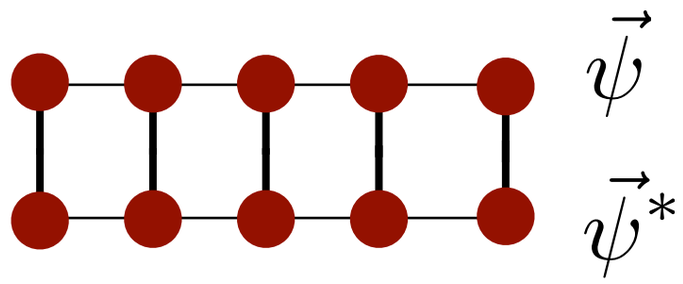
Problem in graphical representation

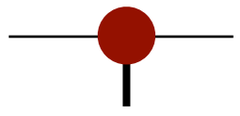
Cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

$\vec{\psi}^\dagger(\mathcal{H}\vec{\psi}) =$



$\vec{\psi}^\dagger\vec{\psi} =$



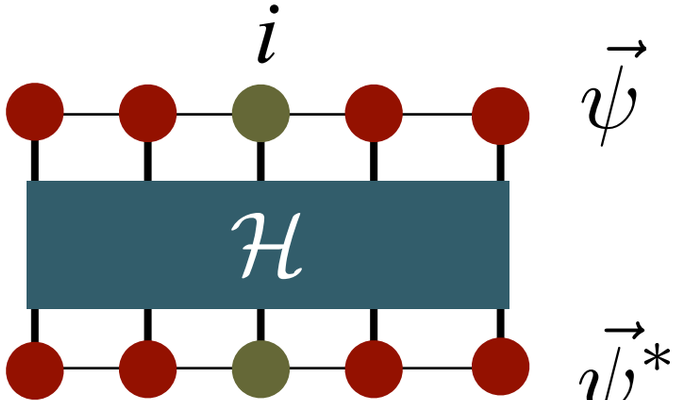
Find $A_i[\sigma_i] =$  which minimizes F.

Iterative optimization

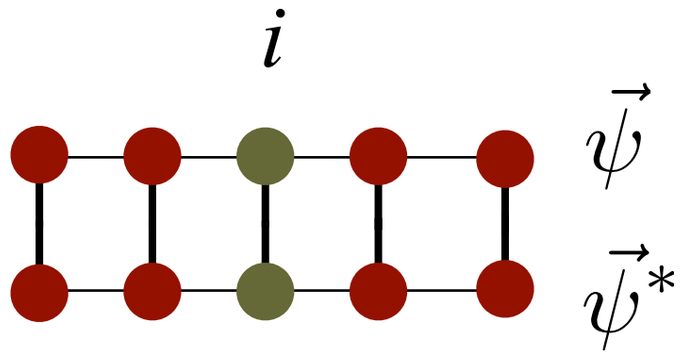
(F. Verstraete, D. Porras, and J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004))

Local optimization problem when we focus on a "site" i :

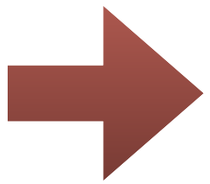
$$\vec{\psi}^\dagger(\mathcal{H}\vec{\psi}) =$$



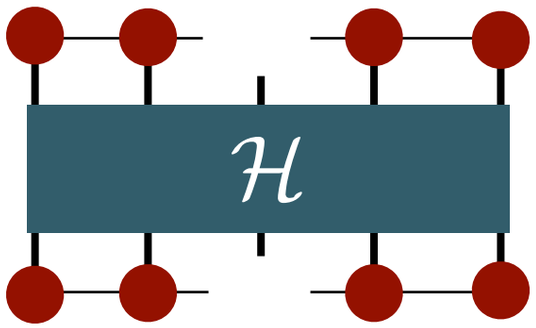
$$\vec{\psi}^\dagger\vec{\psi} =$$



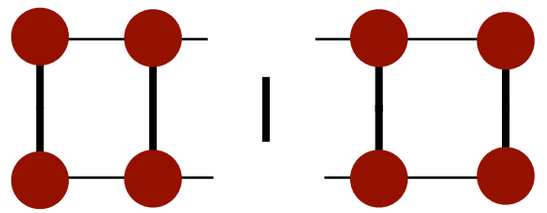
Remove A_i



$$\tilde{\mathcal{H}}_i \equiv$$

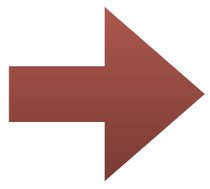


$$\tilde{\mathcal{N}}_i \equiv$$



Minimize

$$F = \frac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}} = \frac{A_i^\dagger(\tilde{\mathcal{H}}_i A_i)}{A_i^\dagger(\tilde{N}_i A_i)}$$



Solve generalized eigenvalue problem

$$\tilde{\mathcal{H}}_i A_i = \epsilon\tilde{\mathcal{N}}_i A_i$$

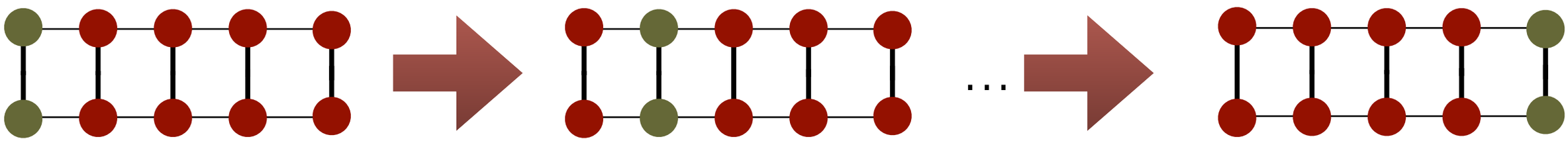
(Find the lowest eigenstate)

Notice: matrix size = $a\chi^2 \times a\chi^2$

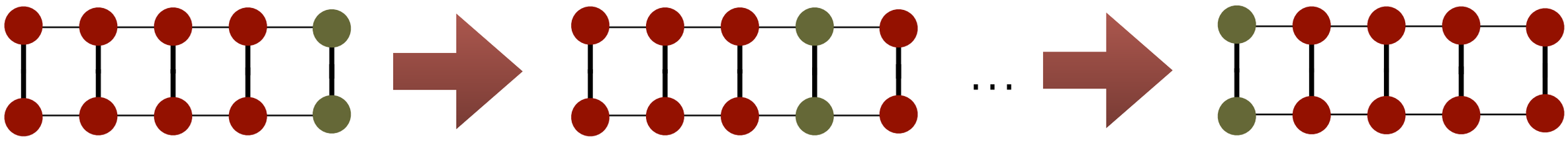
Iterative optimization

(F. Verstraete, D. Porras, and J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004))

Update A_is by "sweeping" sites i =1 to N



Backward "sweeping" sites i =N to 1



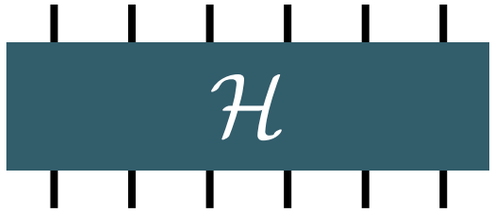
Repeat sweeping until convergence.

Compact representation of an operator

Notice!

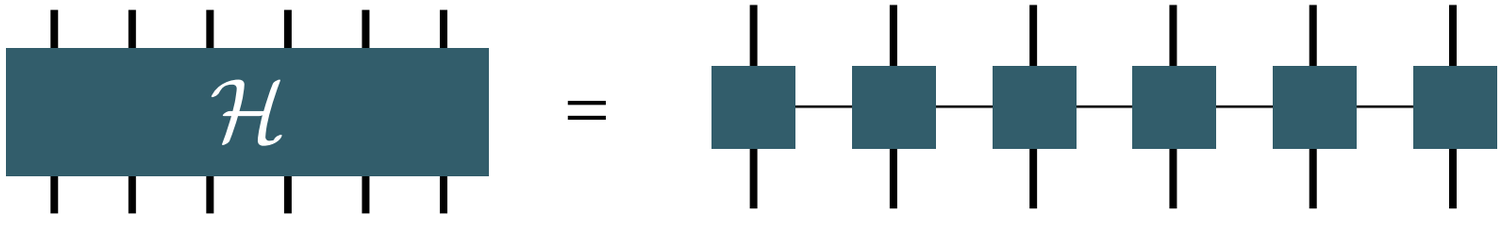
We can conduct this algorithm when we can represent the matrix efficiently.

We consider the situation where we cannot store the matrix in the memory.



$a^N \times a^N$

In practical applications, we usually represent the matrix in so called Matrix Product Operator (MPO) form.



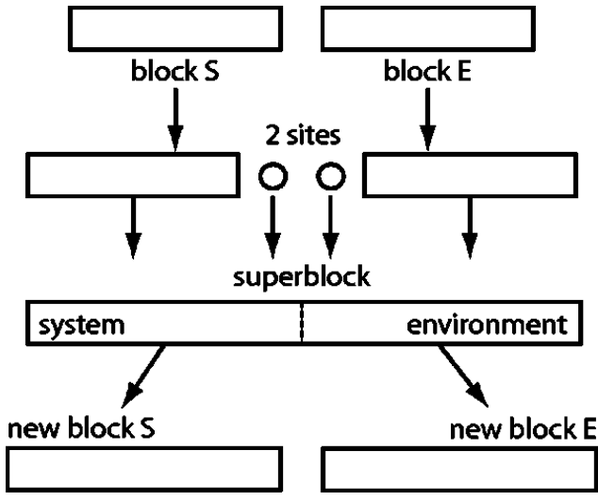
E.g. The Hamiltonian of the Heisenberg model is represented by MPO with bond dimension $\chi = 5$.

Relation to Density Matrix Renormalization Group

The variational MPS method is essentially same with Density Matrix Renormalization Group (DMRG) algorithm.
（密度行列繰り込み群）

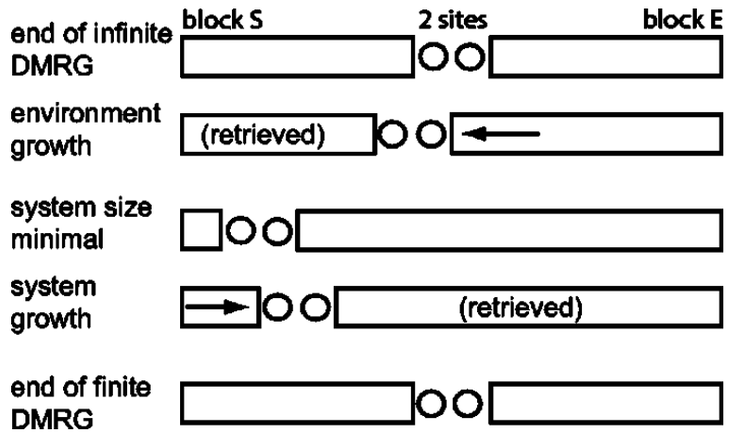
DMRG selects compact basis based on entanglement between "System" and "Environment" blocks.

(S. R. White, Phys. Rev. Lett. **69**, 2863 (1992))
(U. Schollwöck, Rev. Mod. Phys. **77,** 259 (2005))
(U. Schollwöck, Annals. of Physics **326**, 96 (2011))



DMRG is a powerful tool in physics and chemistry

- One-dimensional spin systems
- One-dimensional electron systems
- Small molecules
- Small two-dimensional systems



The original DMRG did not use MPS explicitly.
But, MPS gives us a theoretical background for why DMRG works well.

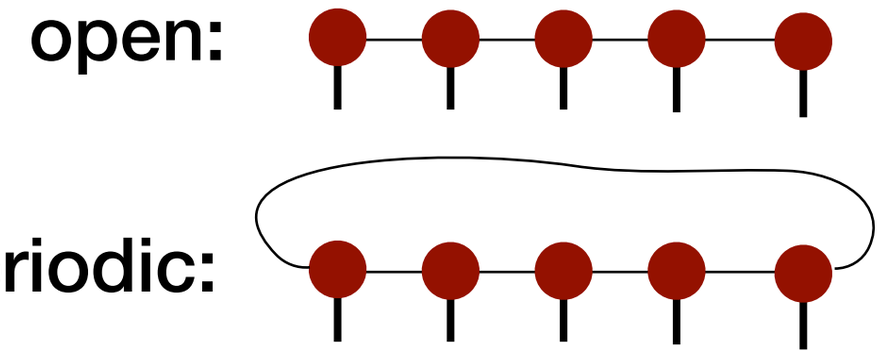
Relation to Density Matrix Renormalization Group

Conventional DMRG algorithm corresponds to variational calculation using open boundary MPS.

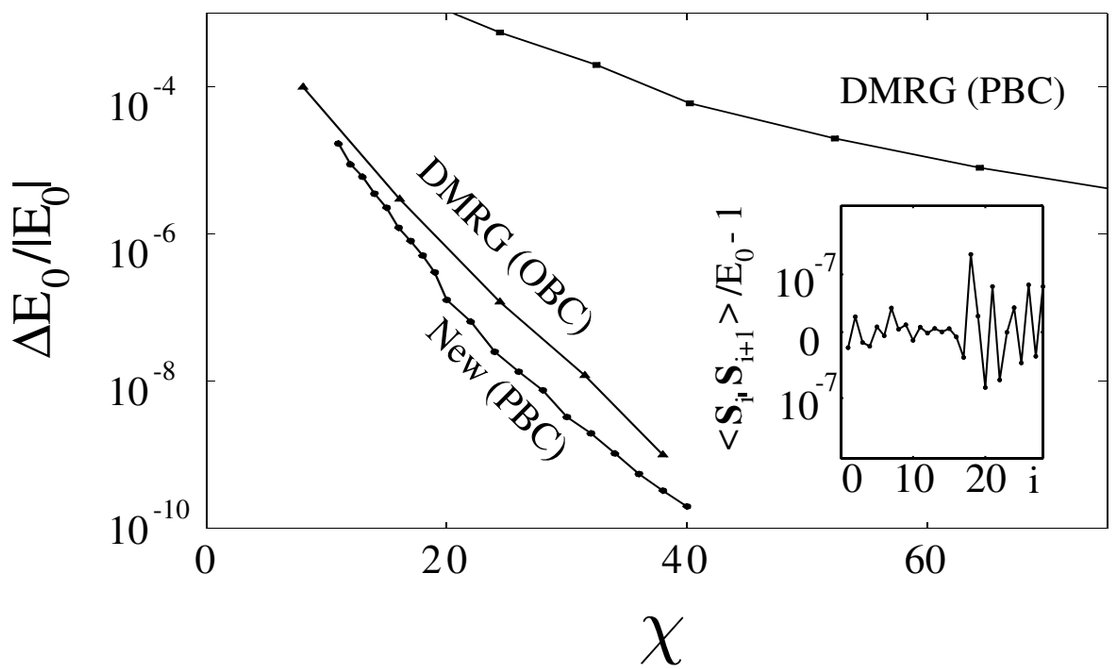
(F. Verstraete, D. Porras, and J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004))

Accuracy becomes worse if we consider systems with periodic boundary condition.

If we use periodic MPS instead of open MPS, we can represent the ground state more efficiently.



S=1/2 Heisenberg chain, (N=40)



Application to time evolutions of quantum system

Time evolution of a quantum system

Schrödinger equation: $i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}|\psi(t)\rangle$

Formal solution:

$$|\psi(t)\rangle = e^{-it\mathcal{H}/\hbar}|\psi(0)\rangle$$

Time evolution operator
(時間発展演算子)

Time evolution using MPS:

1. Multiply the time evolution operator to a MPS.
2. Find an approximate MPS representation for it.

When the time step (t) is small,
we can perform the above step efficiently.

Time evolution of a quantum system using MPS

Target: (Basically) one-dimensional quantum system with short range interaction

Typical example: Chain of qbits or quantum spins

Transverse field Ising model

$$\mathcal{H} = -\sum_{i=1}^{N-1} S_i^z S_{i+1}^z - h \sum_{i=1}^{N} S_i^x$$

Heisenberg model

$$\mathcal{H} = \sum_{i=1}^{N-1} (S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + S_i^z S_{i+1}^z) - h \sum_{i=1}^{N} S_i^z$$

Typical situation: Quantum quench

Initial state: Ground state of a Hamiltonian which well approximated by MPS

t > 0: Hamiltonian suddenly changes from the initial one.

For a "short" time interval, evolving state is approximated by MPS efficiently.

Suzuki-Trotter decomposition

Suzuki-Trotter decomposition: (M. Suzuki, Commun. Math. Phys. **51**, 183 (1976))

Systematic approximation of exponential operator

$$e^{\tau(\mathcal{A}+\mathcal{B})} = e^{\tau\mathcal{A}}e^{\tau\mathcal{B}} + O(\tau^2) \qquad \text{(1st order)}$$

$$(\mathcal{A}\mathcal{B} \neq \mathcal{B}\mathcal{A}) \qquad = e^{\tau/2\mathcal{A}}e^{\tau\mathcal{B}}e^{\tau/2\mathcal{A}} + O(\tau^3) \quad \text{(2nd order)}$$

$$= e^{\tau/2\mathcal{B}}e^{\tau\mathcal{A}}e^{\tau/2\mathcal{B}} + O(\tau^3) \quad \text{(2nd order)}$$

If our Hamiltonian is represented as a sum of "local" operators,

$$\mathcal{H} = \sum_i H_i$$

E.g. transverse field Ising model

$$H_i = -S_i^z S_{i+1}^z - \frac{h}{2}(S_i^x + S_{i+1}^x)$$

Time evolution operator can be approximated as

$$e^{-it\mathcal{H}/\hbar} = \left(e^{-i\delta\mathcal{H}}\right)^M = \left(\prod_j e^{-i\delta H_j}\right)^M + O(\delta) \qquad \text{(1st order)}$$

$$\delta \equiv t/(M\hbar)$$

Graphical representation of Suzuki-Trotter decomposition

Suppose the Hamiltonian can be decomposed into the sum of two-body local terms

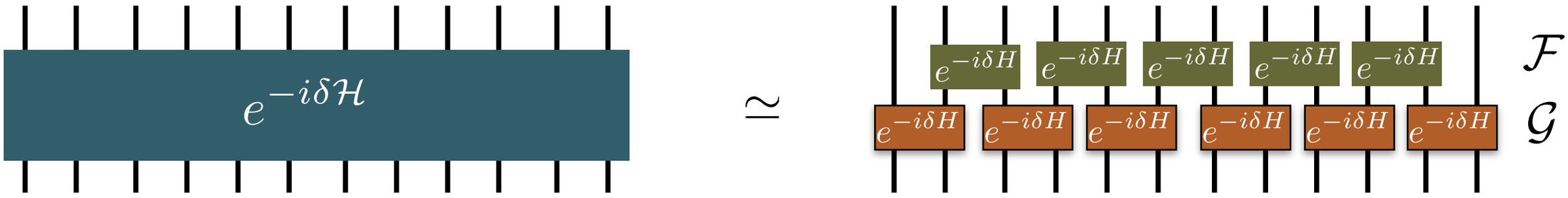
$$\mathcal{H} = \sum_i H_i = \sum_{i \in \text{even}} H_i + \sum_{i \in \text{odd}} H_i$$

$$= \mathcal{F} + \mathcal{G} \qquad\qquad [\mathcal{F}, \mathcal{G}] \neq 0$$

Suzuki-Trotter decomposition of time evolution operator

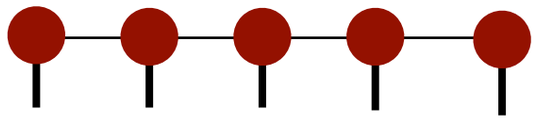
$$e^{-i\delta\mathcal{H}} = e^{-i\delta\mathcal{F}} e^{-i\delta\mathcal{G}} + O(\delta^2)$$

(1st order)

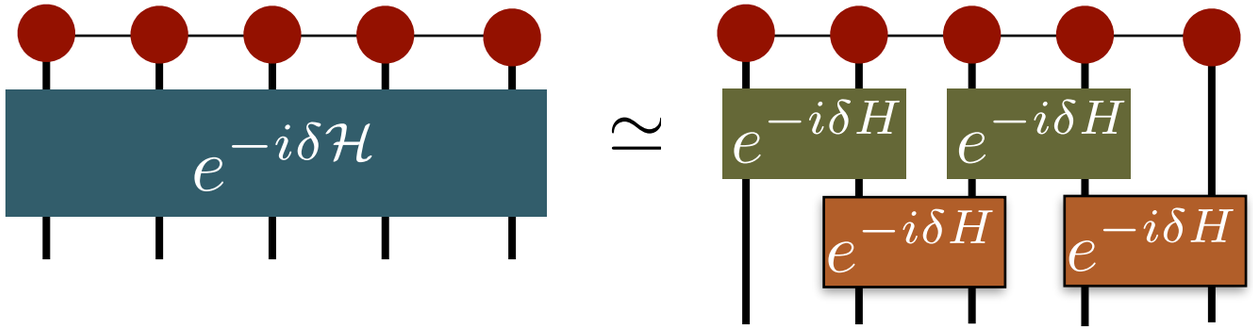


Multiplication of time evolution operator

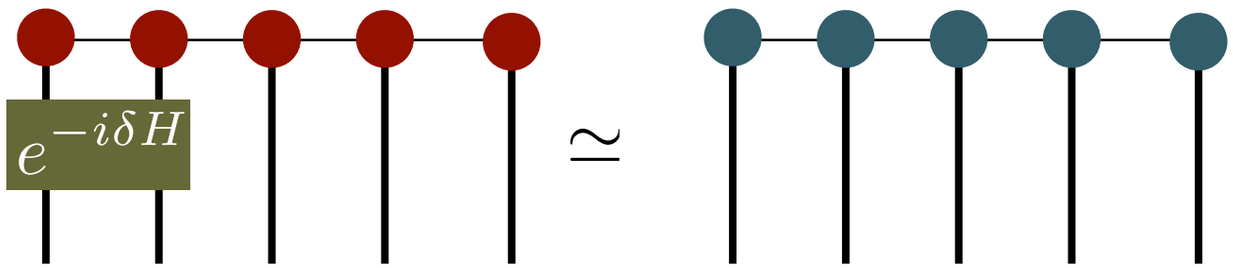
If we have MPS representation of $|\psi\rangle$

$|\psi\rangle =$ 

Multiplying the time evolution operator is represented as

$e^{-i\delta H}|\psi\rangle =$ 

If we can perform the transformation



(Generally, all matrices change for better approximation)

We continue the time evolution repeatedly.

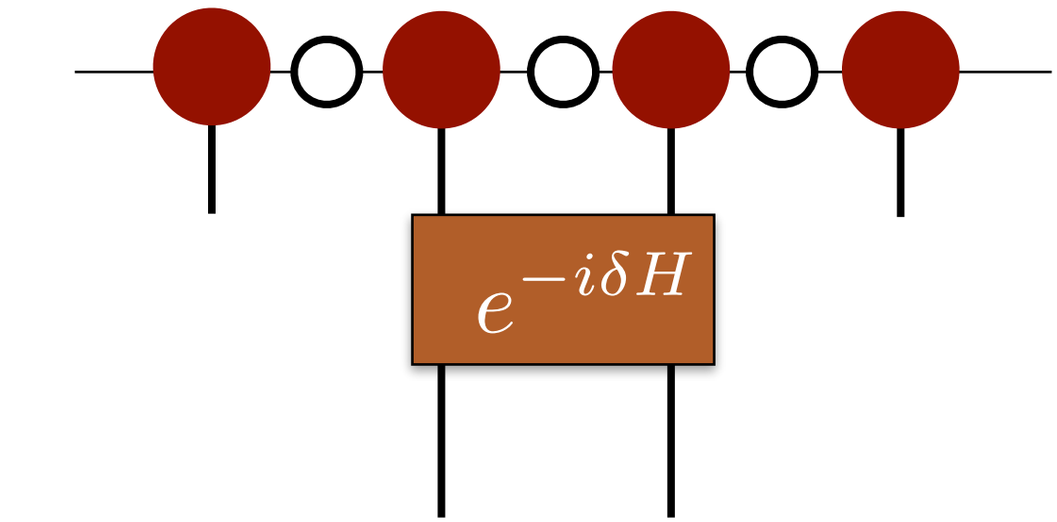
Notice: we want to keep the bond dimension χ constant along time evolution.

TEBD algorithm:

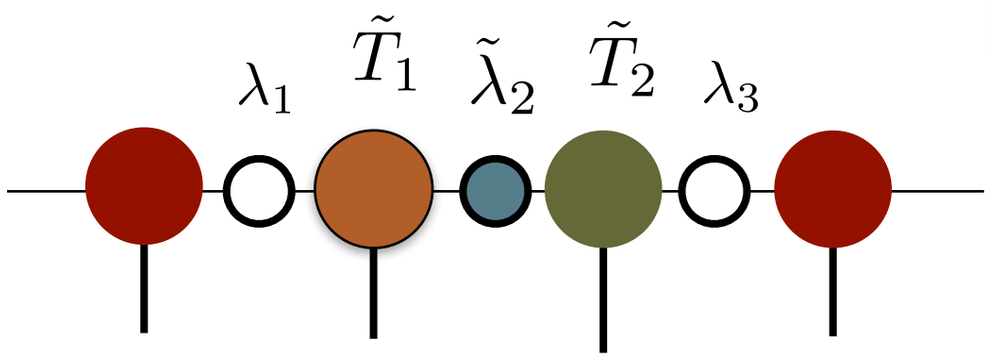
(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Time Evolving Block Decimation (TEBD)

We can perform the accurate transformation locally by using canonical MPS.



Approximation

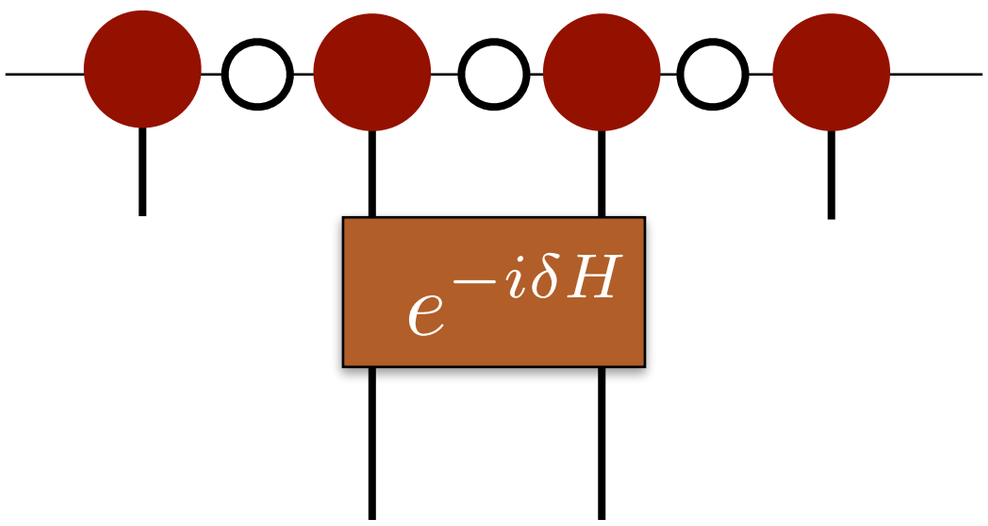


Only the two matrices which are directly applied TE operator changes in MPS.

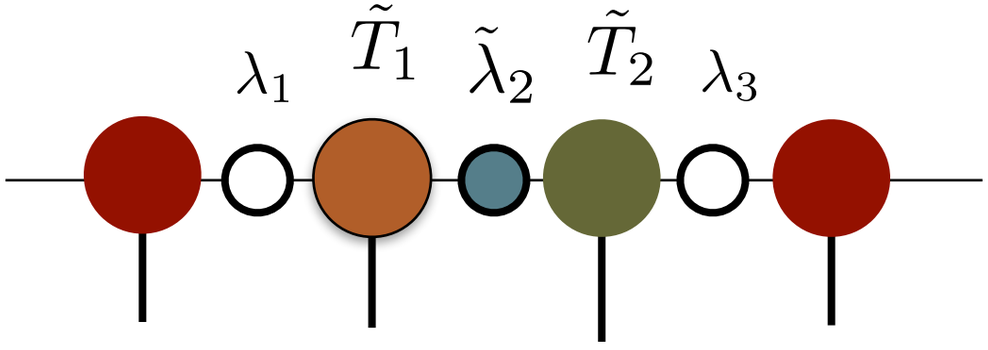
TEBD algorithm:

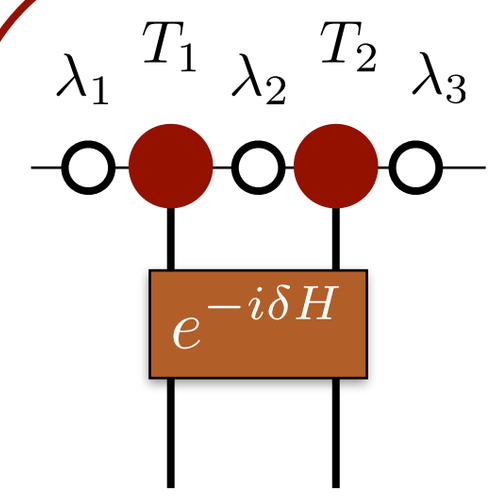
(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Apply TE operator

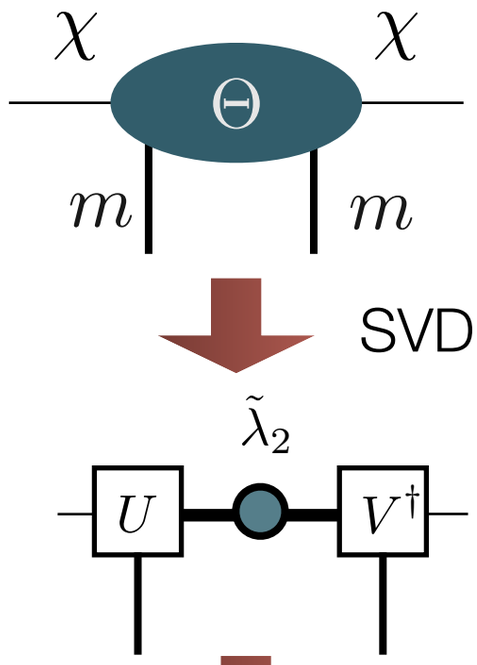


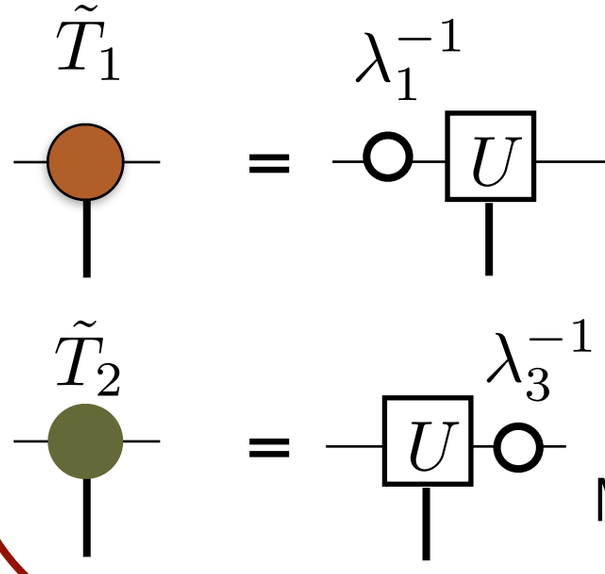
Approximation





Combine and make matrix

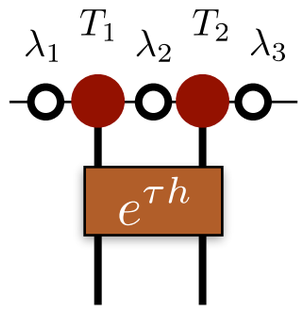




Make tensor

Why TEBD is accurate?

1. For accurate calculation, the canonical form is important.



If λ is equal to the Schmidt coefficient, it contains
all information of the remaining part of the system.

→ Truncation based on local SVD can be globally optimal,
even if we look at a part of the MPS.

2. If the operator is unitary, MPS keeps canonical form within truncation error

→ If we chose the initial MPS as the canonical form,
TEBD algorithm almost keep it.
(So, TEBD is almost "globally optimal")

Extension to infinite sytem iTEBD:

Finite system: TEBD

(G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007))

(R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008))

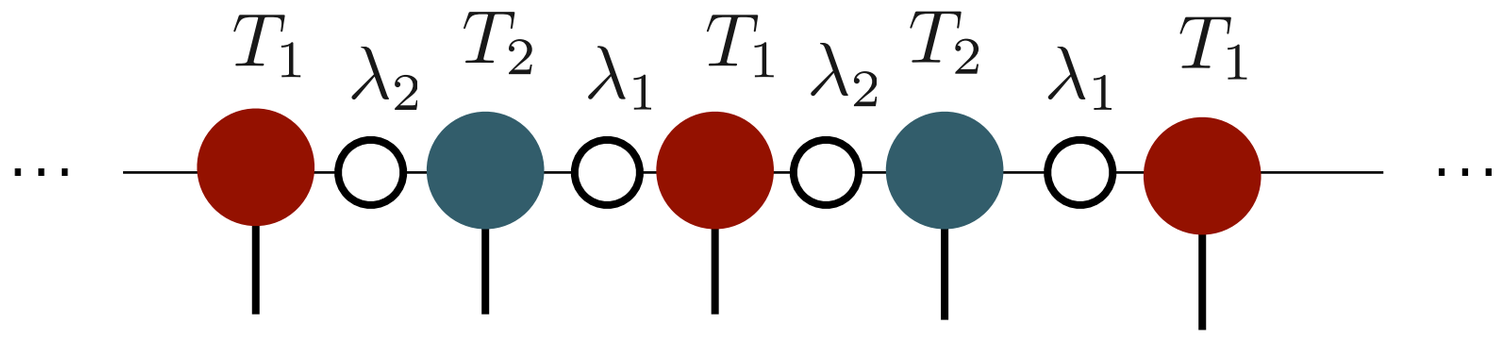
Sequentially apply ITE operators → $O(N)$ SVD for each step

Infinite system: iTEBD

Due to the translational invariance, all SVD are equivalent. → $O(1)$ SVD for each step

*Note

Because of SVD in iTEBD algorithm, we need at least two independent tensors even in translationally invariant system



(i)TEBD can be used for eigenvalue problem

Method to optimize MPS for GS of a specific Hamiltonian

1. Variational optimization

   Change matrix elements to reduce the energy: $\min_{T,\lambda} \dfrac{\langle\Psi|\mathcal{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle}$

2. Imaginary time evolution

Simulate imaginary time evolution: $|\Psi_{\mathrm{GS}}\rangle \propto \lim_{\beta\to\infty} e^{-\beta\mathcal{H}}|\Psi_0\rangle$
（虚時間発展）

For a initial state $\langle\Psi_{\mathrm{GS}}|\Psi_0\rangle \neq 0$

→ By replacing the time evolution operator to the imaginary time evolution operator,

$$e^{-i\mathcal{H}t} \to e^{-\tau\mathcal{H}} \qquad (t \to -i\tau)$$

We can use (TEBD) algorithm for eigenvalue problem.

Difference between TE and ITE

$e^{-i\mathcal{H}t}$:Time evolution operator is unitary

$e^{-\mathcal{H}\tau}$:Imaginary time evolution operator is not unitary

In general, by multiplying imaginary time evolution operator to MPS, the canonical form is destroyed and TEBD becomes less accurate.

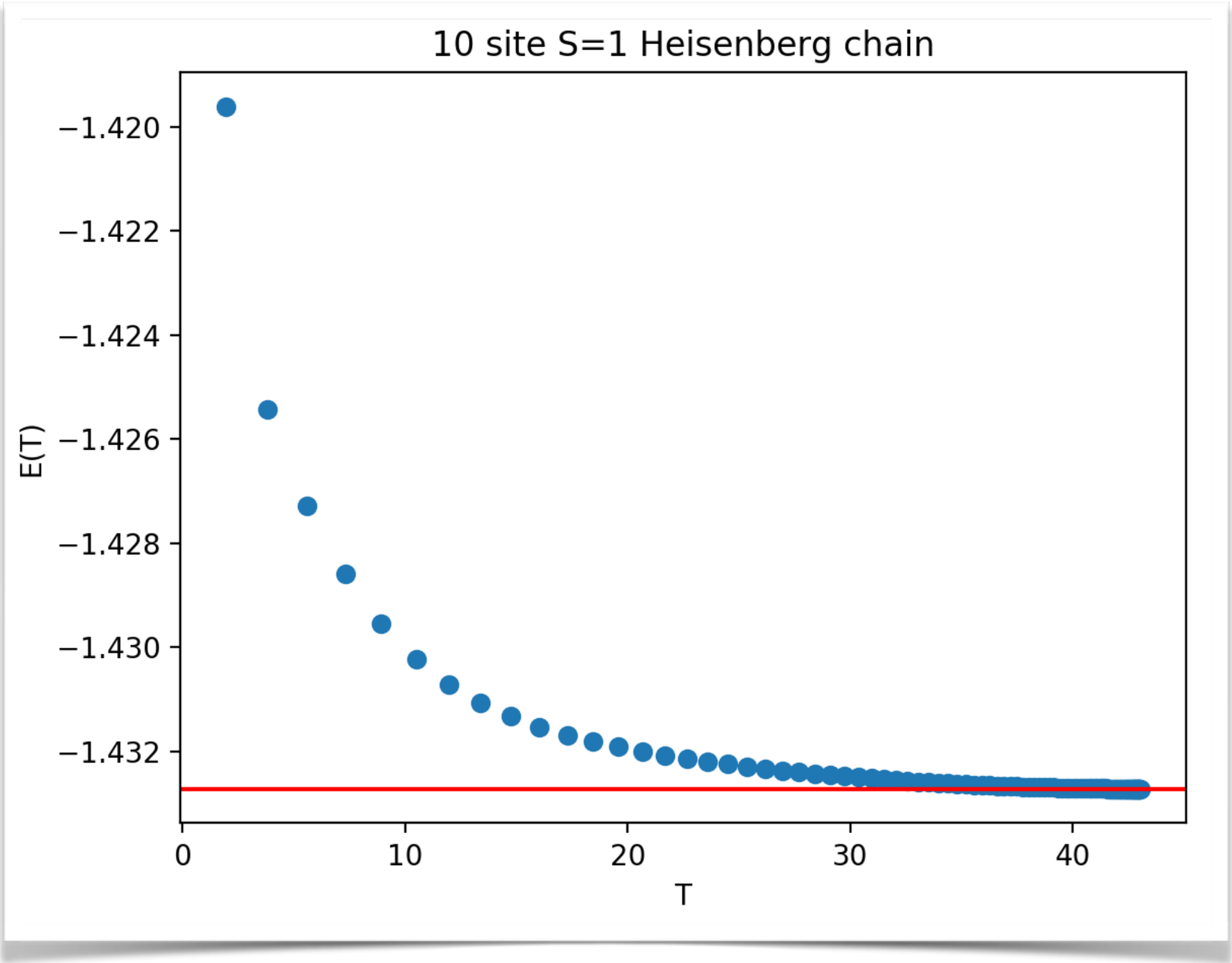
However, when τ is small the operator is almost unitary.

(Because it is almost identity matrix)

If we chose the initial MPS as the canonical form,
TEBD algorithm almost keep it.
(So, TEBD is almost "globally optimal" even in the case of the imaginary time evolution.)

*Instead, we can transform the MPS into the canonical form after multiplying ITE operator in every steps.

3-1: Energy dynamics in TEBD



Exercise 3: (TEBD and) iTEBD simulation (ITE)

3-1: TEBD simulation

Simulate small finite size system and compare energy with ED

Sample code: Ex3-1.py or Ex3-1.ipynb

show help: *python Ex3-1.py -h*

3-2: iTEBD simulation

Simulate infinite system and calculate energy

Sample code: Ex3-2.py or Ex3-2.ipynb

show help: *python Ex3-2.py -h*

*** Try simulation with different "chi_max", "T_step"**

Requirement for running sample scripts

Python environment: python2.7 or python3

Modules: numpy, scipy and matplotlib

Usage:

For jupyter notebook, type

jupyter notebook

and select Ex?-?.ipynb .

For python (command line), type

python Ex?-?.py -h

, then you can know how to change the parameters.

Application to machine learning

E. Miles Stoudenmire and D. J. Schwab, NIPS 2016

Machine learning for classification problem

Problem: we want to classify an input vector by several labels

E.g Classification of handwriting images

Standard procedure:

First, input vector $\boldsymbol{x}$ is mapped onto higher dimensional space

$$\vec{\psi}(\boldsymbol{x})$$ (non-linear feature map)

Then it is transformed to labels through a linear map

$$f^l = W^l \vec{\psi}(\boldsymbol{x})$$

In the case of supervised machine learning, we optimize W based on the correct labels of a lot of input vectors.

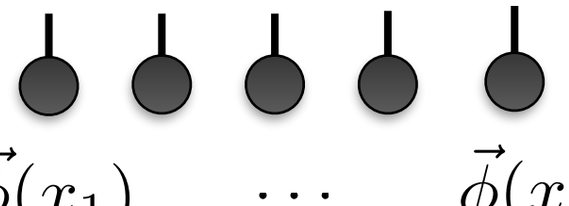
MPS representation of the classification problem

E. Miles Stoudenmire and D. J. Schwab, NIPS 2016

If we select a "product state" as a feature map

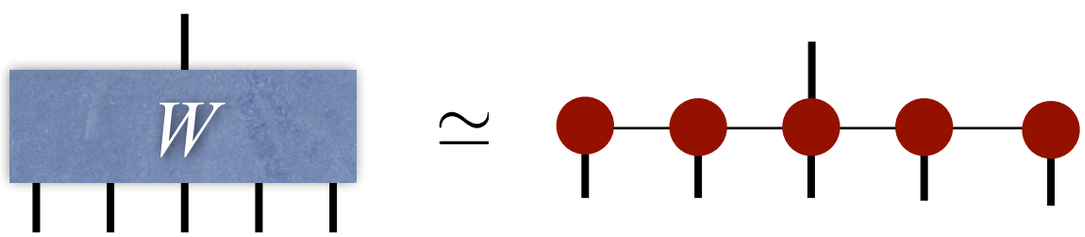
$$\psi_{i_1,i_2,\ldots,i_N}(\boldsymbol{x}) = \phi_{i_1}(x_1) \otimes \phi_{i_2}(x_2) \otimes \cdots \otimes \phi_{i_N}(x_N)$$

$$\vec{\psi} =$$

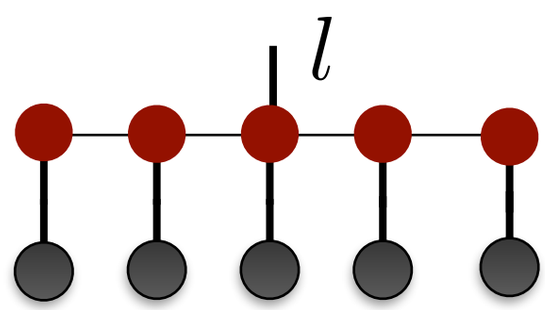


The dimension of vector space is a^N

Then we can apply MPS approximation for W



$$f^l = W^l \vec{\psi}(\boldsymbol{x}) =$$



MPS representation of the classification problem

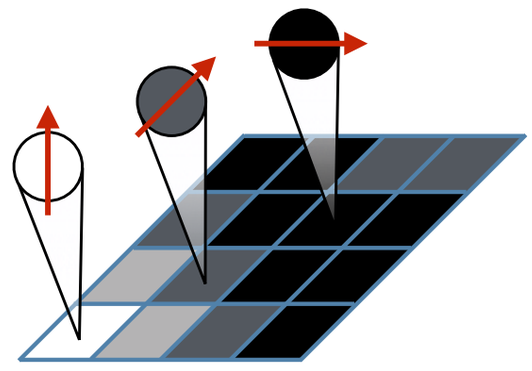
E. Miles Stoudenmire and D. J. Schwab, NIPS 2016

Feature map

$$\psi_{i_1,i_2,\ldots,i_N}(\boldsymbol{x}) = \phi_{i_1}(x_1) \otimes \phi_{i_2}(x_2) \otimes \cdots \otimes \phi_{i_N}(x_N)$$

Their feature map:

$$\phi^{s_j}(x_j) = \left[\cos\left(\frac{\pi}{2}x_j\right),\ \sin\left(\frac{\pi}{2}x_j\right)\right]$$



For the case of grayscale image

Application to MNIST database of handwritten digits

(handwritten numbers from 0 to 9)

χ	Test set error	
10	~5%	500/10000
20	~2%	200/10000
120	~0.97%	97/10000

It is comparable with the state of the art! <1%

We can enjoy demo: http://itensor.org/miles/digit/index.html

References for application to machine learning

Low-Rank Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Problems: Perspectives and Challenges PART 1

A. Cichocki, N. Lee, I.V. Oseledets, A.-H. Phan, Q. Zhao, D. Mandic

Foundations and Trends in Machine Learning, vol. 9, no. 4-5, pp. 249-429, 2016 (arXiv.1609.00893)

Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations. Part 2 Applications and Future Perspectives

A. Cichocki, A-H. Phan, Q. Zhao, N. Lee, I.V. Oseledets, M. Sugiyama, D. Mandic

Foundations and Trends in Machine Learning: Vol. 9: No. 6, pp 431-673, 2017 (arXiv.1708.09165)

Topics:

- Supervised Learning with Tensors
- Tensor Train Networks for Selected Huge-Scale Optimization Problems
- Tensor Networks for Deep Learning
- ...

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

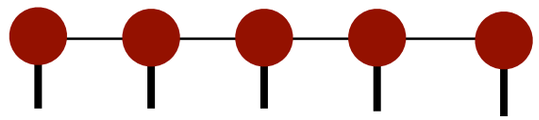
9. フラストレート磁性体への応用

テンソルネットワーク表現への発展

Breakdown of MPS representation

Required bond dimension in MPS representation

$$S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi$$



The upper bound is independent of the "length".

length of MPS ⇔ size of the problem

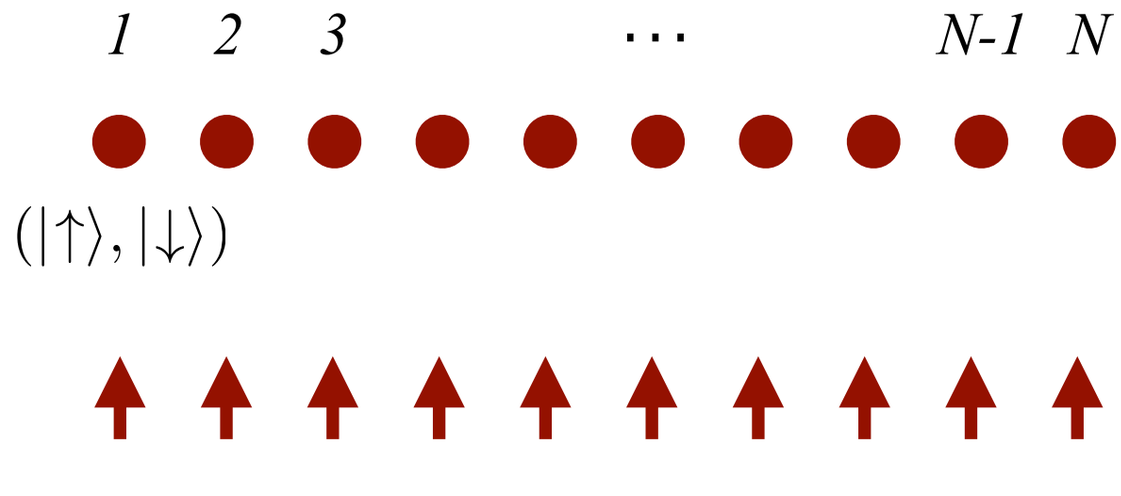
N $\qquad\qquad$ a^N

EE of the original vector	Required bond dimension in MPS representation
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

Phase transition

Transverse field Ising chain:

$$\mathcal{H} = -\sum_{i=1}^{N-1} S_i^z S_{i+1}^z - h\sum_{i=1}^{N} S_i^x$$



Ground state $|\Psi\rangle$

$h = 0$:Ferromagnetic state

$h \to \infty$:Disordered state
(Field induced ferro)

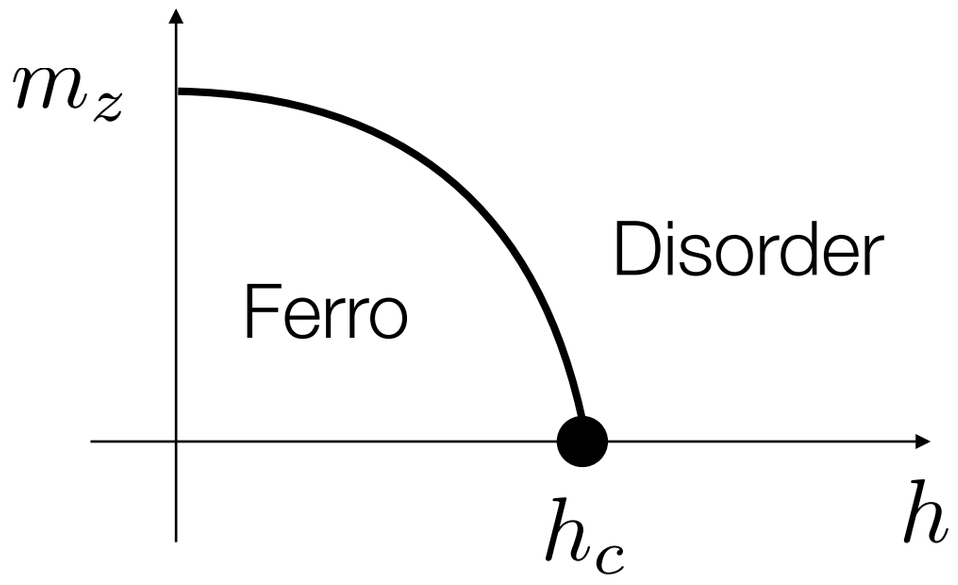
$$\rightarrow = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$$

In between these two limits, there is a phase transition.

At the phase transition, order parameter becomes zero.
（秩序変数）

(Spontaneous) Magnetization
（自発磁化）

$$m_z = \frac{1}{N}\sum_i \langle\Psi|S_i^z|\Psi\rangle$$



Critical point and correlation length

$h = h_c$:Critical point （臨界点）

Behavior of a correlation function:

$0 \leq h < h_c$:Ferromagnetic state

$$\langle \Psi | S_i^z S_{i+r}^z | \Psi \rangle \sim C e^{-\frac{r}{\xi}} + m_z^2$$

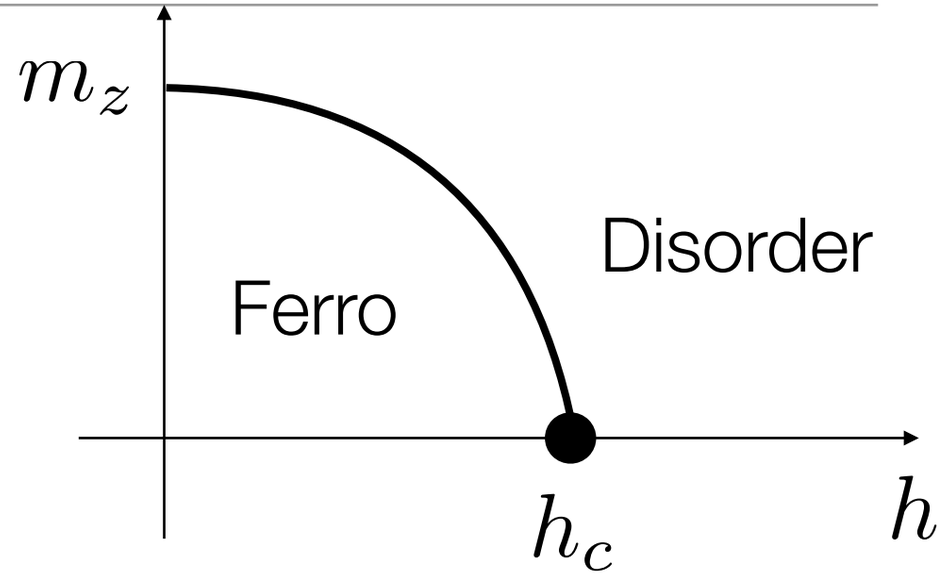
$h_c < h$:Disordered state

$$\langle \Psi | S_i^z S_{i+r}^z | \Psi \rangle \sim e^{-\frac{r}{\xi}}$$

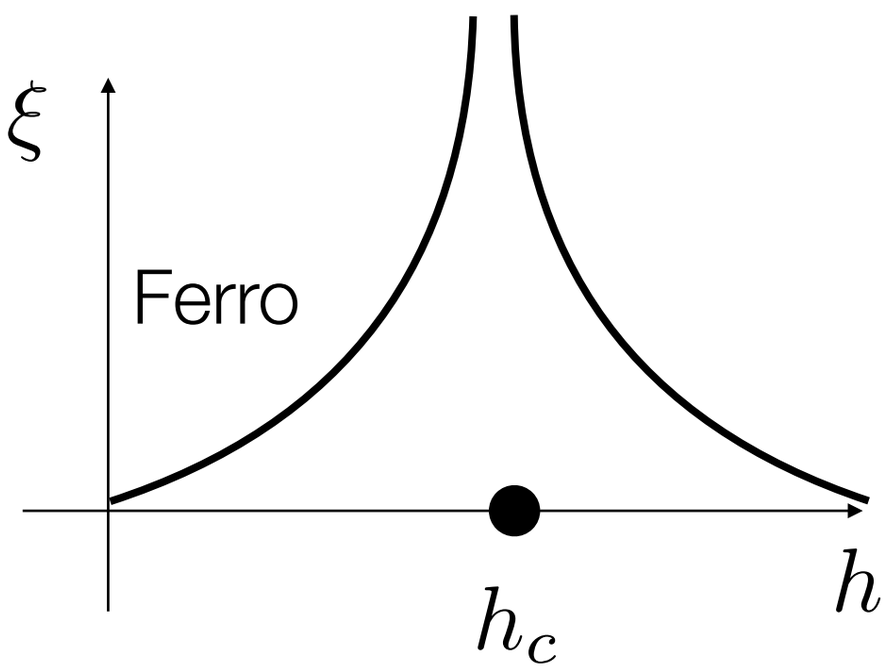
$h = h_c$:Critical point

$$\langle \Psi | S_i^z S_{i+r}^z | \Psi \rangle \sim r^{-2p}$$

Correlation length diverges at critical point!



$\xi = \xi(h)$:Correlation length (相関長)



Scale invariance at the critical point

$h = h_c$:Critical point （臨界点）

$$C(r) \equiv \langle \Psi | S_i^z S_{i+r}^z | \Psi \rangle \sim r^{-2p}$$

Power low decay!

After a scale transformation $r' = br$

$$\Rightarrow \quad C(r') = C(br) = b^{-2p} C(r)$$

Change in the correlation function is only a constant factor.

$\Rightarrow$ If we scale spins as $\tilde{S}_i^z = b^p S_i^z$

the correlation function becomes

$$\tilde{C}(r') \equiv \langle \Psi | \tilde{S}_i^z \tilde{S}_{i+r'}^z | \Psi \rangle = C(r)$$

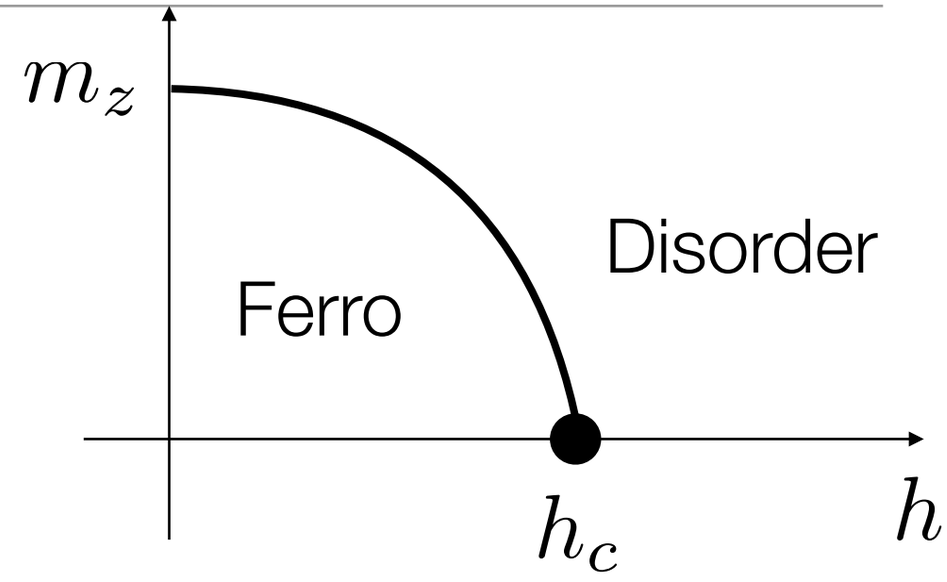
This property is called as "scale invariance". （スケール不変性）

Physics (properties) in different scale is essentially same.

DMRG (variational MPS) calculation of TFI model

Ö. Legeze, and G. Fáth, Phys. Rev. B **53**, 14349 (1996)

Errors of the ground and the 1st excited states energies varying system size N.



For a fixed dimension m,

Ferro and disordered states:

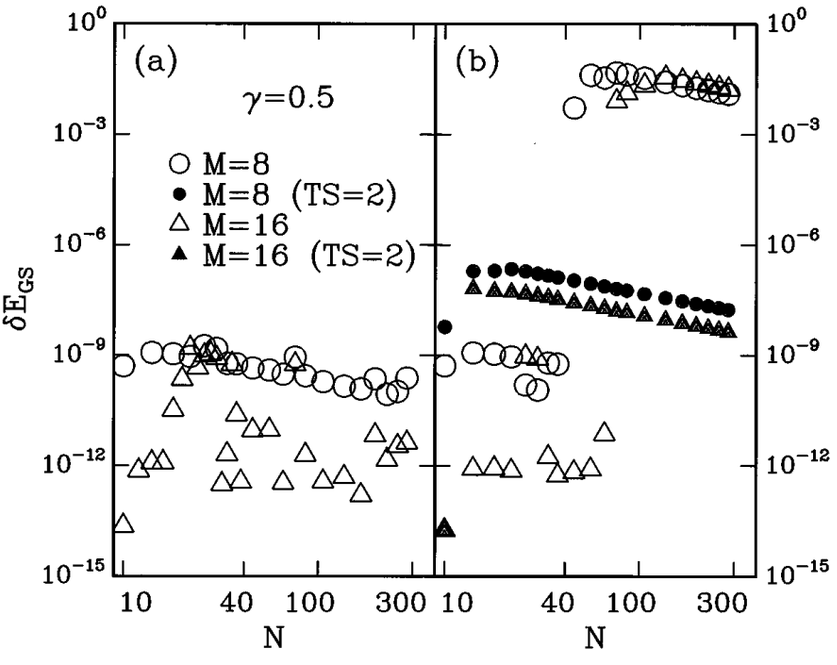
The errors are almost independent of N.

Critical point:

The errors gradually increases as increase N.

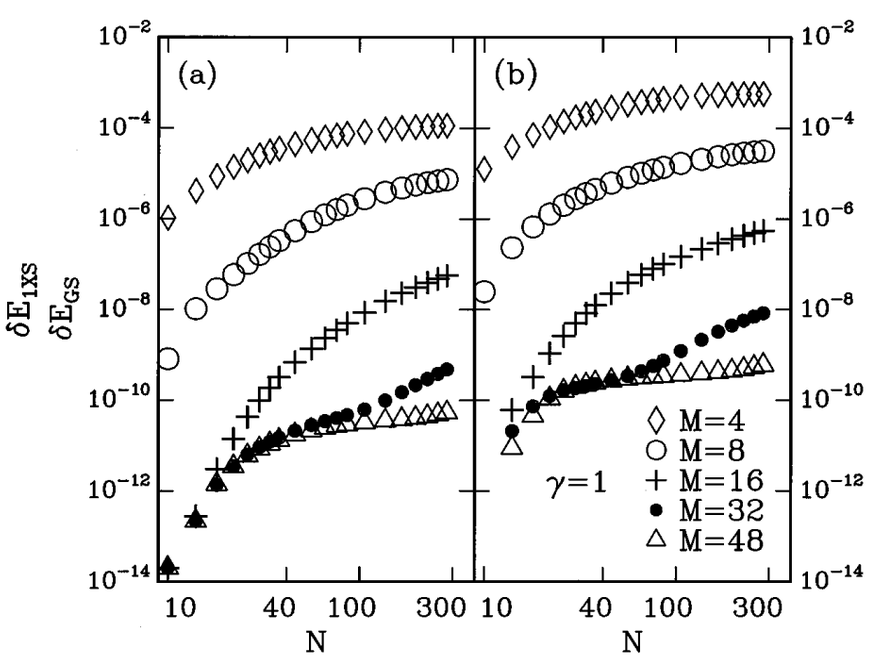
$0 \leq h < h_c$

$h = 0.25$



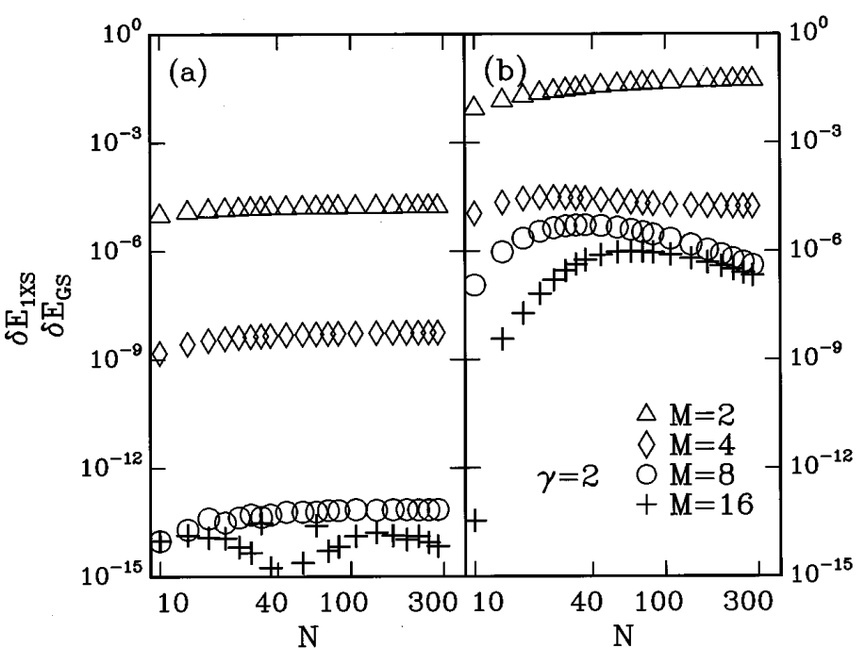
$h = h_c$

$h = 0.5$



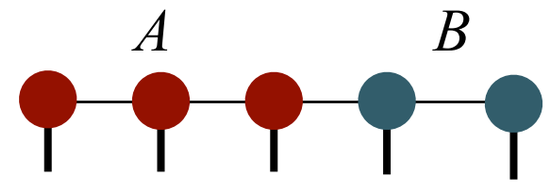
$h_c < h$

$h = 1.0$



Entanglement entropy of TFI model

Entanglement entropy: $S_A = -\mathrm{Tr}\ \rho_A \log \rho_A$



State	EE of the original vector	Required bond dimension
Ferro or Disordered	$S_A = O(1)$	$\chi = O(1)$
Critical	$S_A = O(\log N)$	$\chi = O(N^\alpha)$

We need polynomially large bond dimension for critical system!

→ More efficient tensor network for critical systems?

Key point: Scale invariance of the system

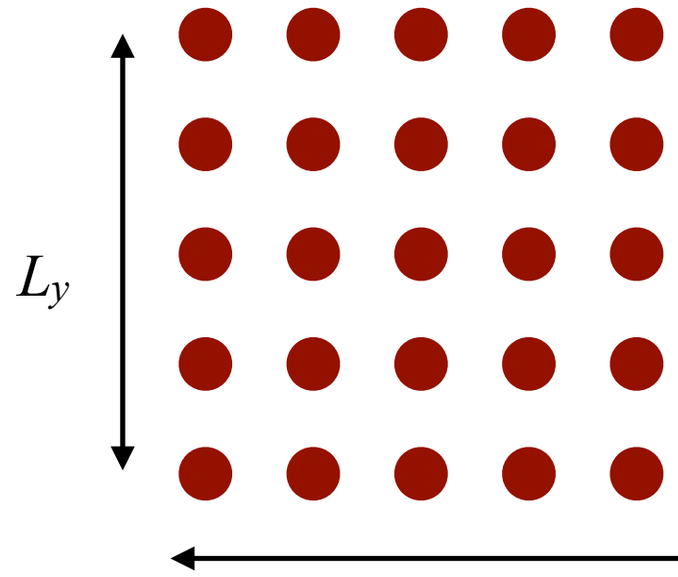
Higer dimensional system

Transverse field Ising model on square lattice:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} S_i^z S_j^z - h \sum_{i=1}^{N} S_i^x$$

$\sum_{\langle i,j \rangle}$:Summation over the nearest neighbor pair

Two-dimensional array



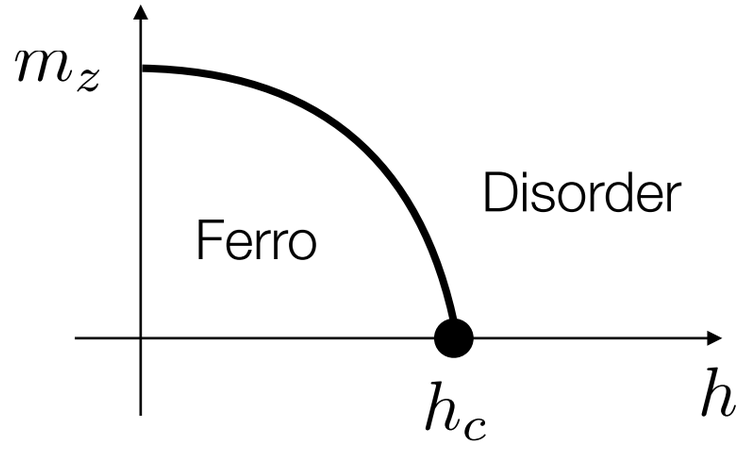
$$N = L_x \times L_y$$

Area law

Even in ferro and disordered phases,
the entanglement entropy depends on size N.

$$S_A \sim \sqrt{N} = L$$

Phase diagram



MPS for two-dimensional system

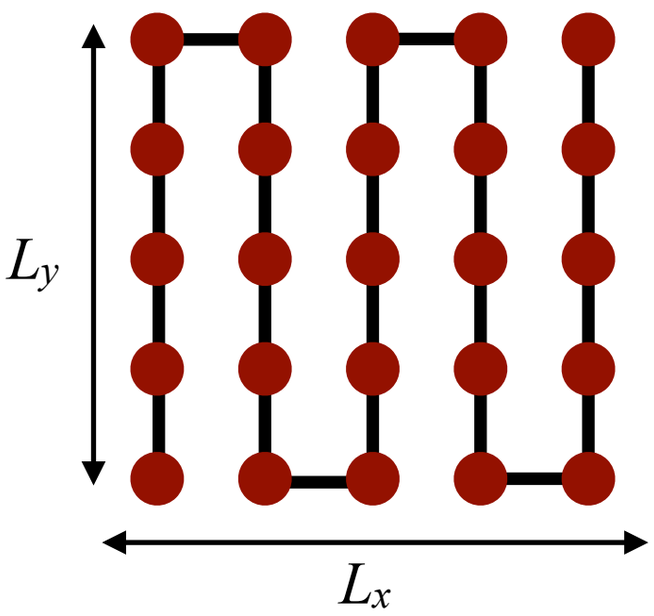
When we apply MPS representation for a square lattice system:

Setting **(1)** $S_A \leq L_x \log \chi$:Satisfying area law?

Setting **(2)** $S_{A'} \leq \log \chi$:Break down of the area law!

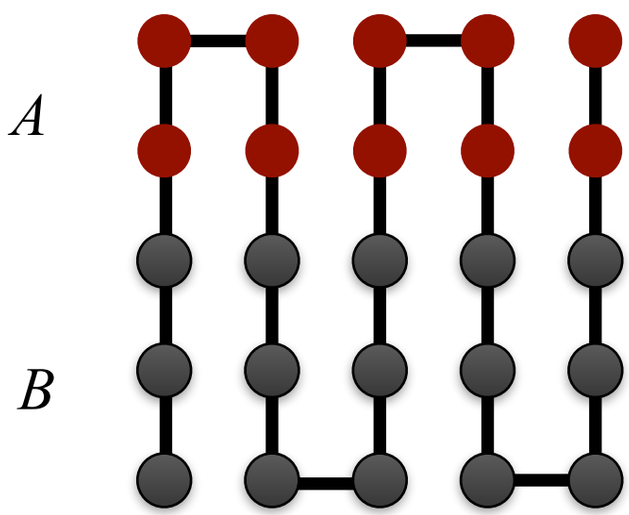
MPS cannot cover the area law of the entanglement entropy in higher (d =2,3, ...) dimensions.

Possible MPS (Snake form)

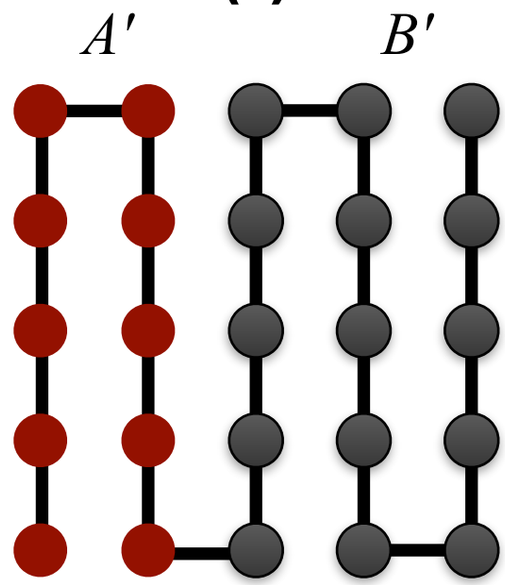


Two settings of system and environment

(1)



(2)



MPS for two-dimensional system: comment

MPS can treat "**rectangular**" or "**quasi one dimensional**" lattice.

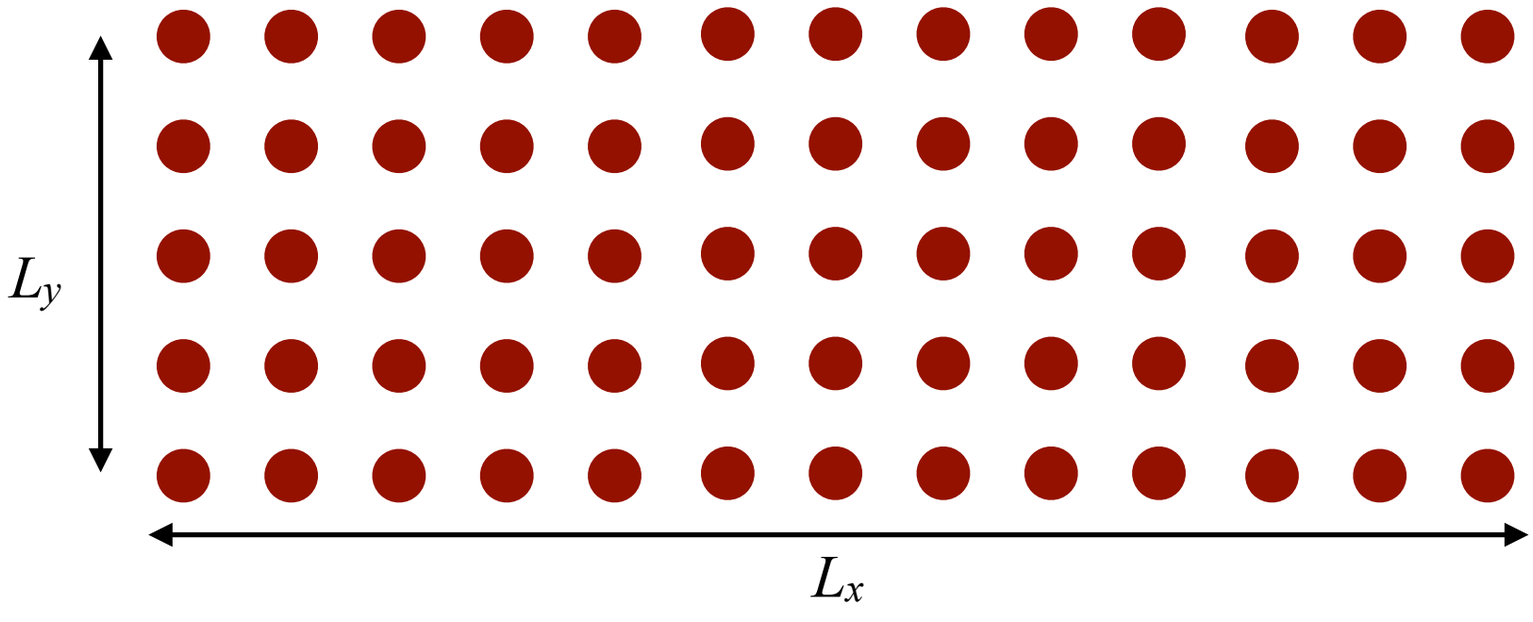
In setting (1), MPS can satisfy the area low partially.

➡ We can increase L_x easily with keeping L_y constant.

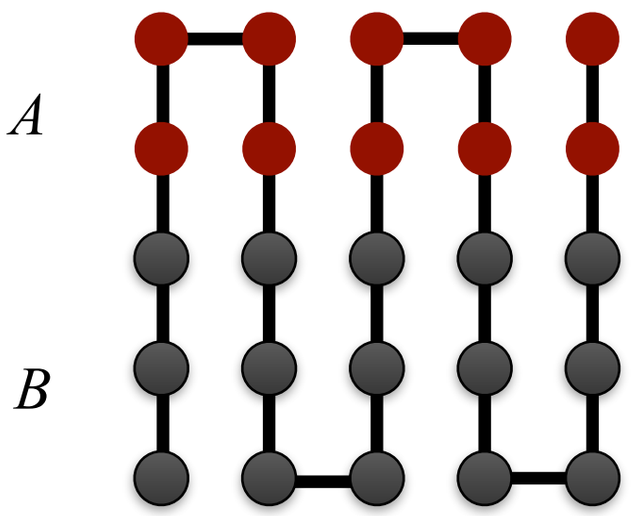
$$\chi = O(e^{L_y})$$

$$L_y \lesssim 10, L_x \gg L_y$$

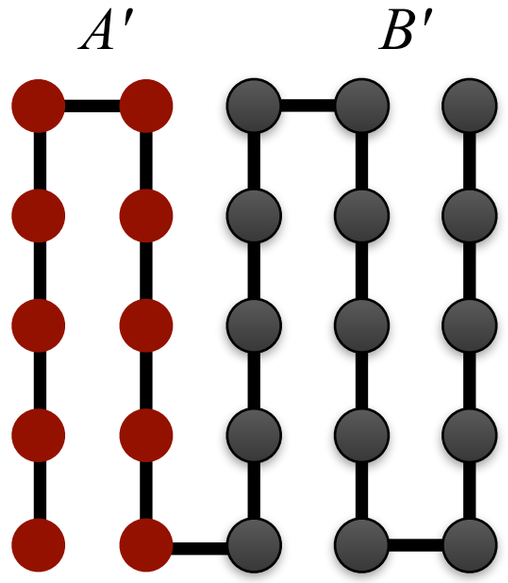
Quasi one dimensional system ("strip" or "cylinder")



(1) $S_A \leq L_x \log \chi$



(2) $S_{A'} \leq \log \chi$



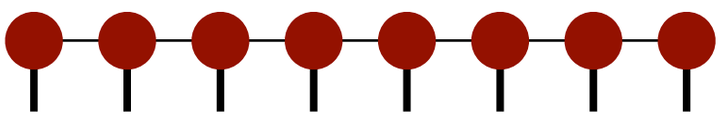
Tensor network for critical systems:
Multi-scale Entanglement Renormalization Ansatz

Hierarchical structure: tree tensor network

Critical system  Scale invariance

A simple scale invariant tensor network: tree tensor network

MPS



"scale"

(renormalization)

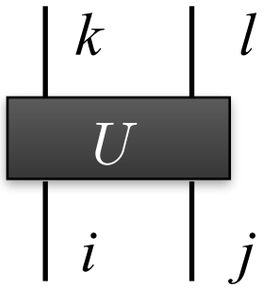
Tree tensor network state

Notice:

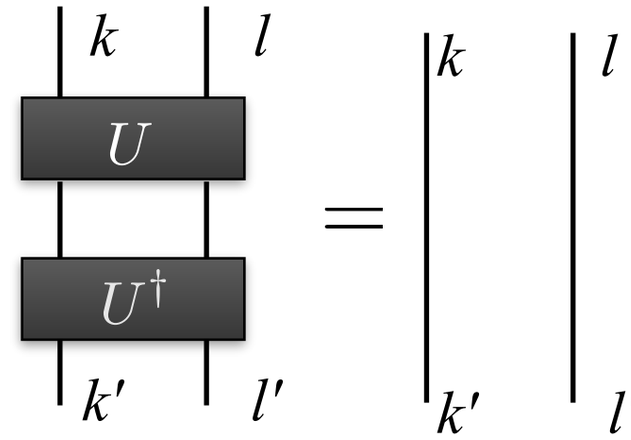
Unitary tensors

Unitary tensor

$$U_{ij}^{kl} =$$

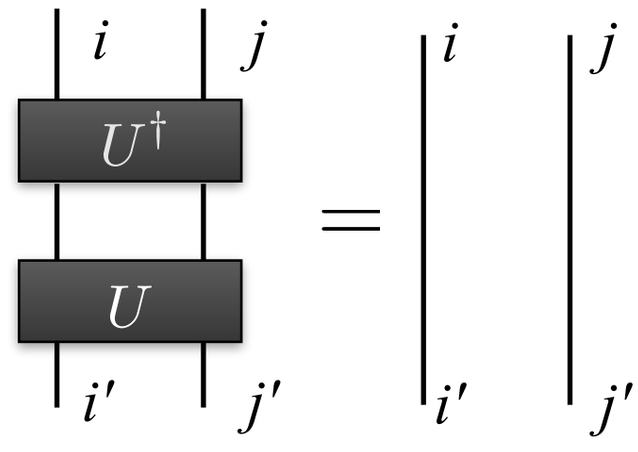


$$\sum_{i,j} U_{ij}^{kl} (U^\dagger)_{k'l'}^{ij} = \delta_{kk'}\delta_{ll'}$$



$$(U^\dagger)_{kl}^{ij} = (U_{ij}^{kl})^*$$

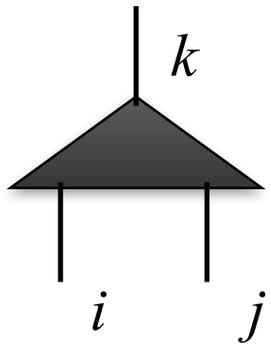
$$\sum_{k,l} (U^\dagger)_{kl}^{ij} U_{i'j'}^{kl} = \delta_{ii'}\delta_{jj'}$$



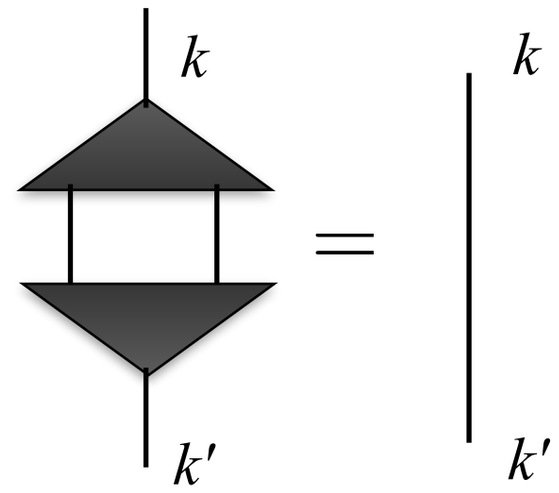
Isometric tensors

Isometric tensor (half unitary tensor) = Isometry

$$W_{ij}^{k} =$$



$$\sum_{i,j} W_{ij}^{k} (W^{\dagger})_{k'}^{ij} = \delta_{kk'}$$



Unitarity condition only for "bottom" legs.

Isometry works as a "projector" from the bottom space to the top space.

$$\dim(\text{bottom}) \geq \dim(\text{top})$$

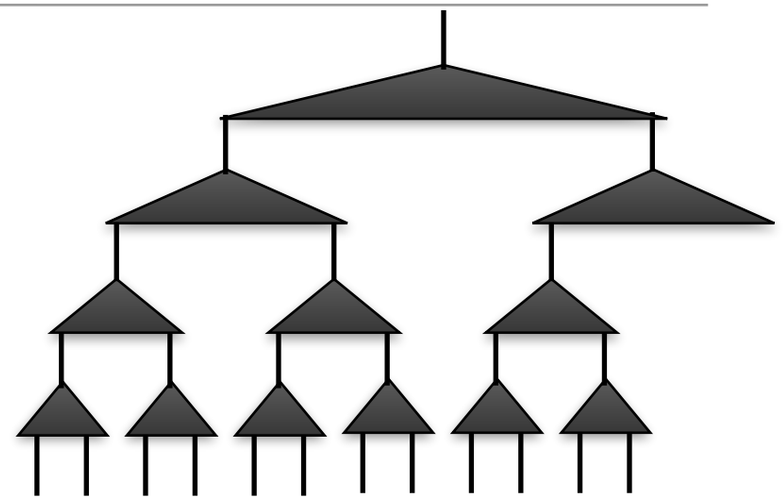
It is also related to the "renormalization" of degree of freedoms.

We pick up "important" degree of freedoms by isometries.

Isometric tree tensor network and its scale invariance

Consider an (infinite) tree tensor network consists of identical isometries as a wave function.

$|\Psi\rangle =$



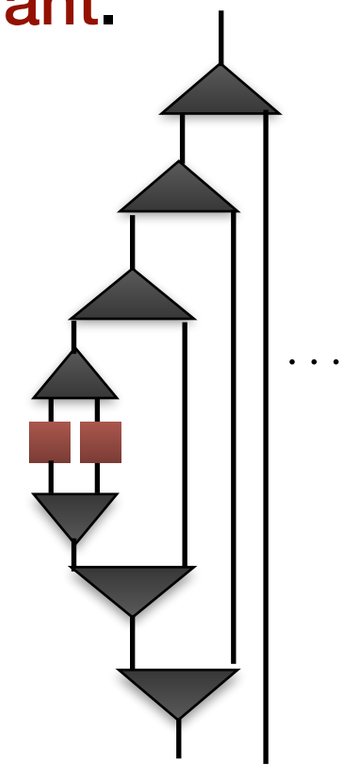
Properties:

1. It is normalized as $\langle\Psi|\Psi\rangle = 1$

(Trivial from the definition of the isometry)

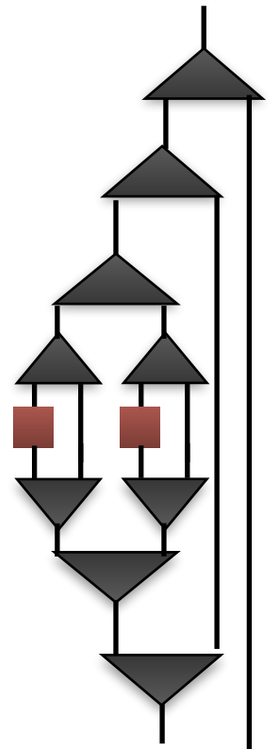
2. It can be scale invariant.

$C(1) \equiv \langle\Psi|S_1^z S_2^z|\Psi\rangle =$



$\cdots$

$C(2) \equiv \langle\Psi|S_1^z S_3^z|\Psi\rangle =$



$\cdots$

$S_i^z =$

spin

$(S_i^z)' =$

"renormalized" spin

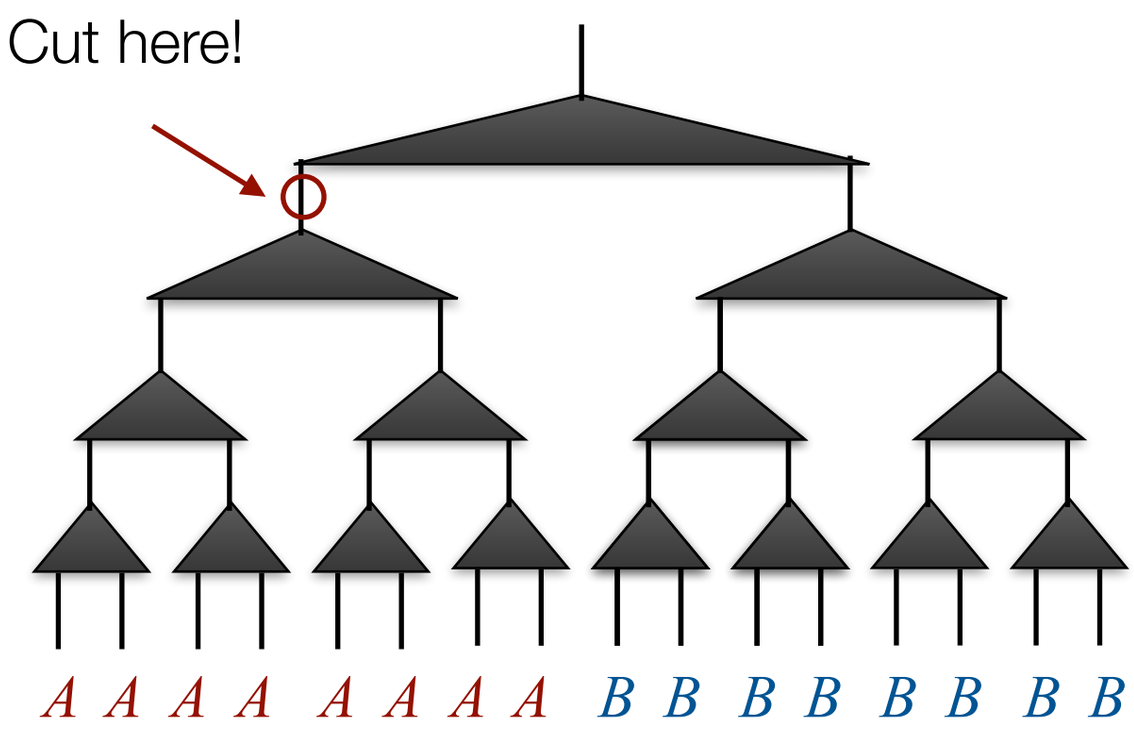
If $(S_i^z)' = 2^{-p} S_i^z$, then $C(2r) = 2^{-2p} C(r)$ **Scale invariant!**

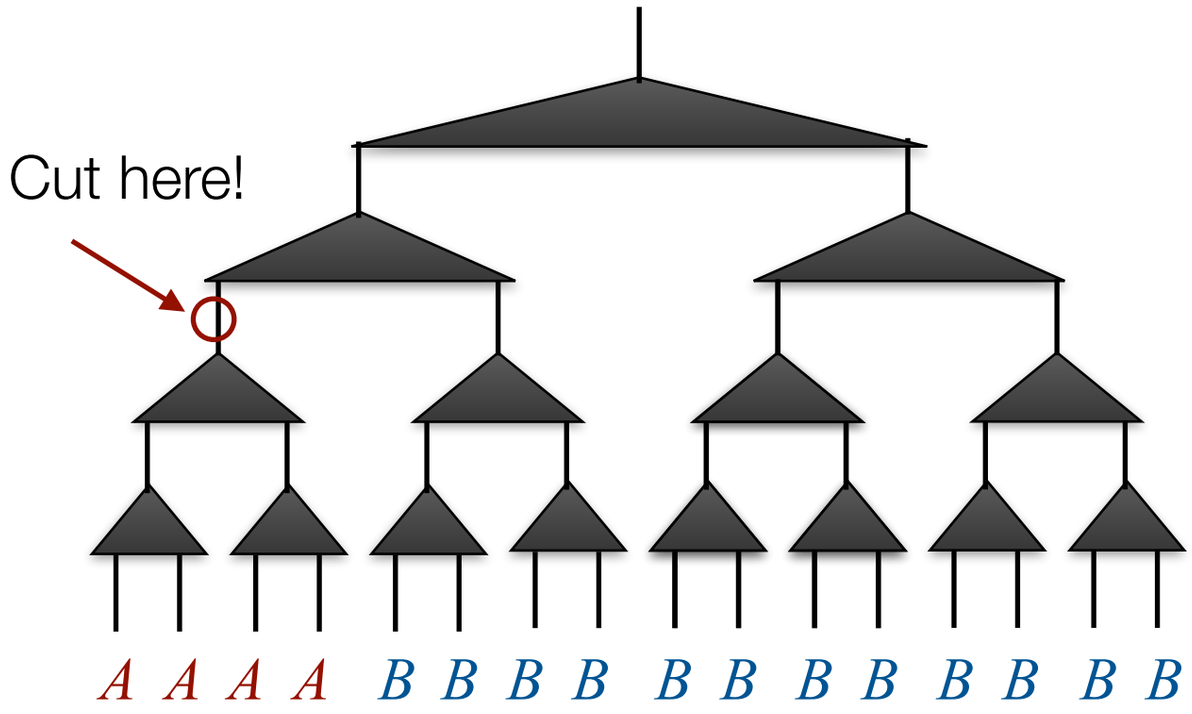
Entanglement entropy of TTN

Entanglement entropy of tree tensor networks (TTN):

Due to the tree structure, two regions are connected by only "one bond".
(or a few)

$$S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi$$



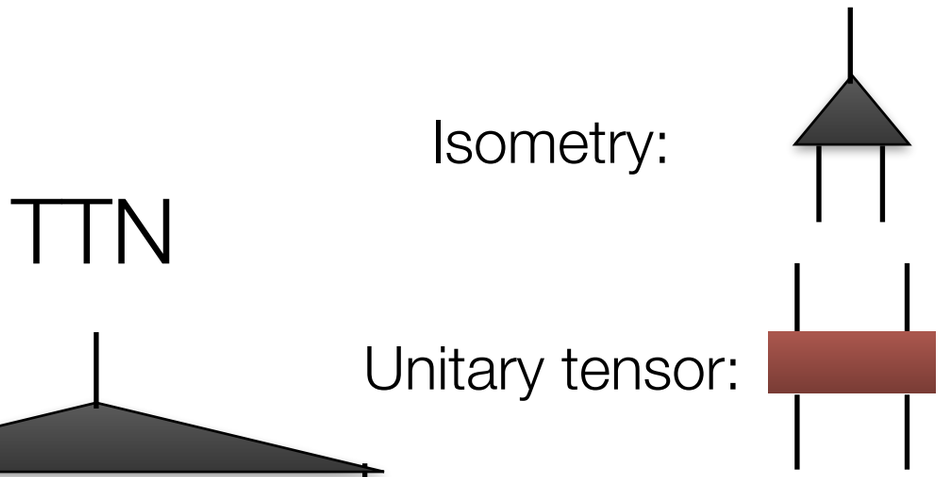


MERA

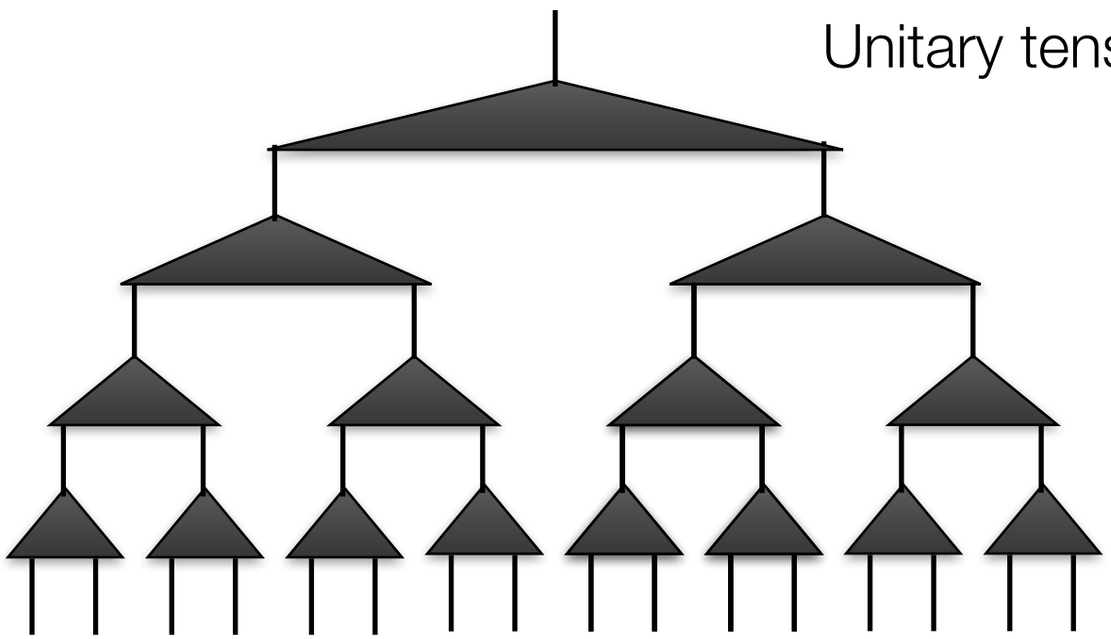
(G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007))
(G. Vidal, Phys. Rev. Lett. **101**, 110501 (2008))

Multi-scale Entanglement Renormalization Ansatz (**MERA**)

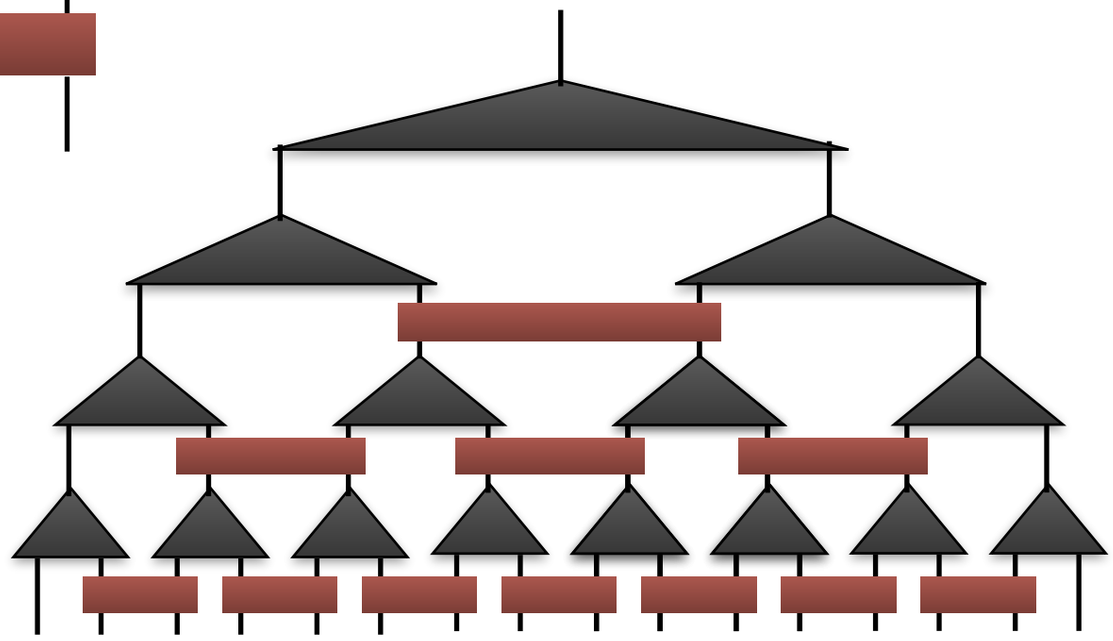
Before applying isometry, insert a unitary tensor.



TTN



MERA



- [x] Normalization
- [x] Scale invariance (if we set the identical tensors)

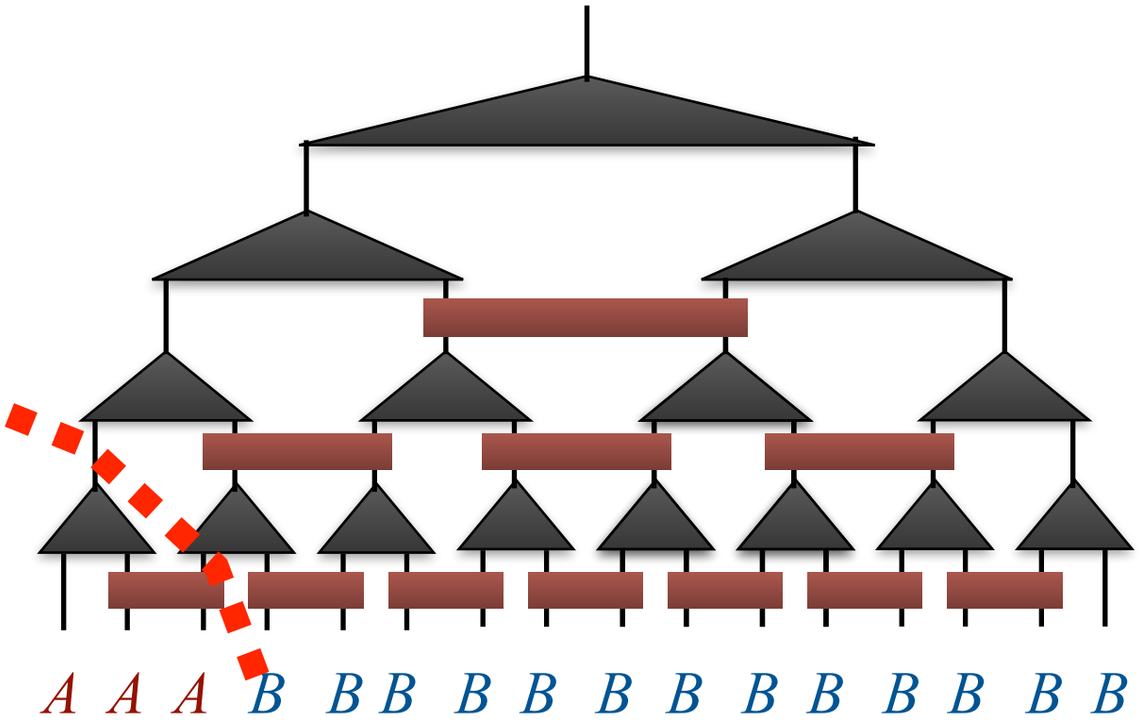
Entanglement entropy of MERA

Due to the unitary matrices, # of bonds connecting two regions logarithmically increase.

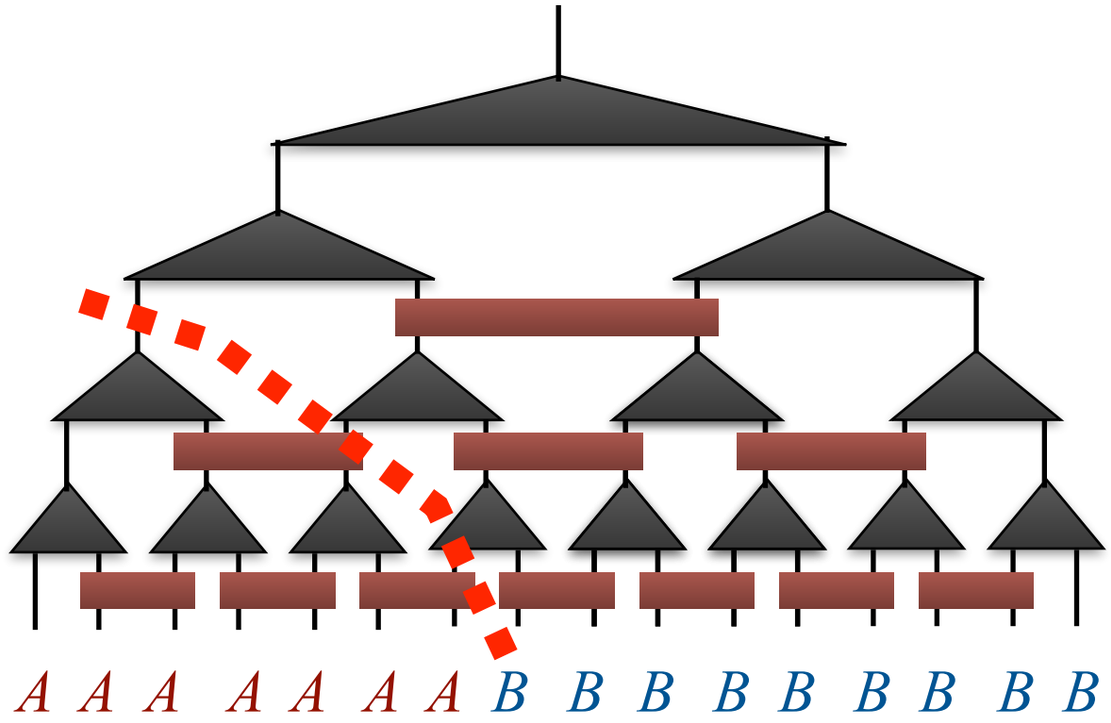
$$\text{rank } \rho_A \leq \chi^{N_c(N)} \sim \chi^{\log N}$$
$$S_A = -\text{Tr } \rho_A \log \rho_A \leq (\log \chi) \log N$$

$N_c(N)$

# of minimum cut for a N-site region



Minimum # of cuts = 2



Minimum # of cuts = 3

Application of MERA

Transverse field Ising chain:

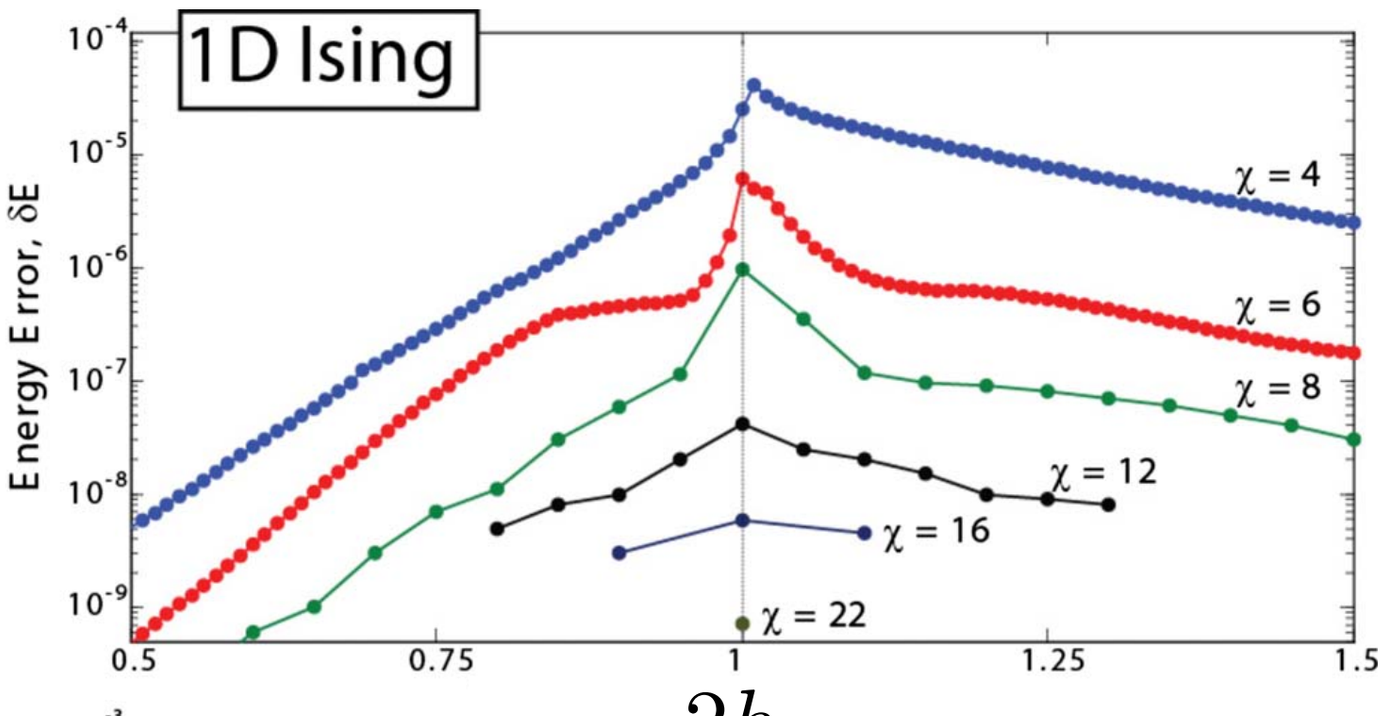
$$\mathcal{H} = -\sum_{i=1}^{N-1} S_i^z S_{i+1}^z - h \sum_{i=1}^{N} S_i^x$$

MERA can represent very large (Infinite) critical system!

Energy errors:

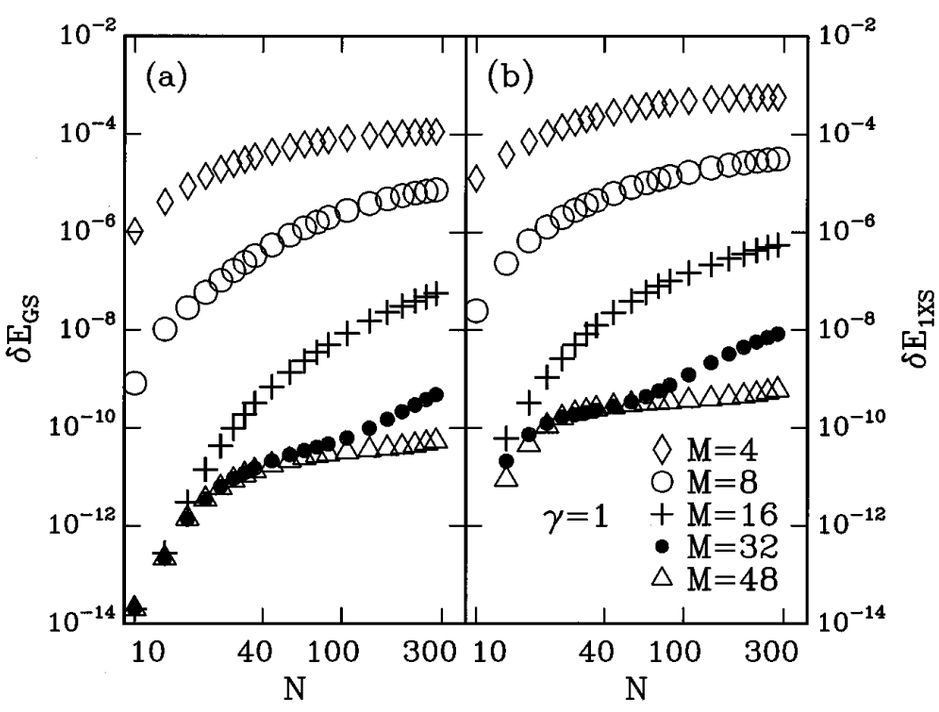
MERA (Infinite chain)

(G. Evenbly and G. Vidal, Phys. Rev. B. **79**, 144108 (2009))



DMRG (finite chain)

Ö. Legeze, and G. Fáth (1996)



Interesting topics related to MERA

- By using scale invariance of MERA, we can calculate properties of critical system accurately.

  (G. Evenbly and G. Vidal, Phys. Rev. B. **79**, 144108 (2009))
  (R.N.C. Pfeifer, (G. Evenbly and G. Vidal, Phys. Rev. A. **79**, 040301(R) (2009))

  - Critical exponents and Operator product expansion coefficients in the Conformal Filed Theory (CFT)

- We can consider MERA in higher dimensions

  - It is scale invariant but satisfies the area low

    (G. Evenbly and G. Vidal, Phys. Rev. Lett. **102**, 180406 (2009))

  - For the system with logarithmic correction in the EE, such as metal, "branching MERA" has been proposed.

    (G. Evenbly and G. Vidal, Phys. Rev. Lett. **112**, 220502 (2014))
    (G. Evenbly and G. Vidal, Phys. Rev. B. **89**, 235113 (2014))

- Relation between MERA and other fields

  - Wavelet transform (G. Evenbly and S. R. White, Phys. Rev. Lett. **112**, 140403 (2016))

  - AdS/CFT (quantum gravity, black hole)

    (M. Nozaki, S. Ryu, and T. Takayanagi, J. High Energy Phys. **10**, 193 (2012))

Tensor network for higher dimensional systems: Tensor Product State (Projected Entangled Pair State)

Entanglement entropy in higher dimensions

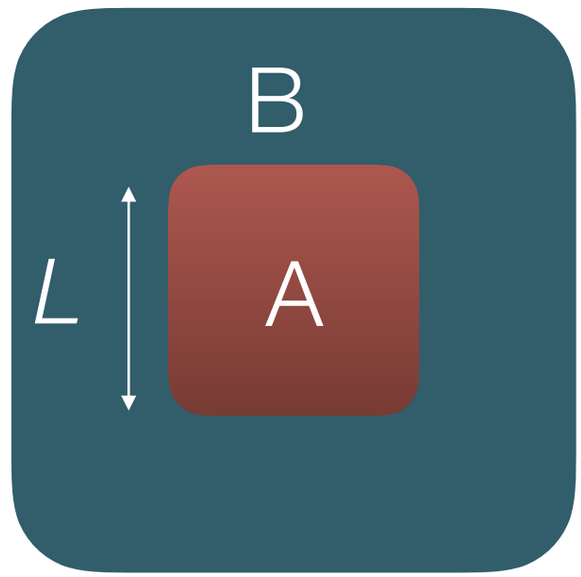
Ground state wave functions:

For a lot of ground states, EE is proportional to its area.

J. Eisert, M. Cramer, and M. B. Plenio, Rev. Mod. Phys, 277, **82** (2010)

Area low:

$$S = -\mathrm{Tr}\left(\rho_A \log \rho_A\right) \propto L^{d-1}$$



In d=1, MPS satisfies the area low.

Q. What is a simple generalization of MPS to d > 1?

A. It is Tensor Product State (TPS)!

Tensor Product State (TPS)

TPS (Tensor Product State) (AKLT, T. Nishino, K. Okunishi, …)

PEPS (Projected Entangled-Pair State)

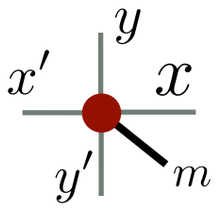
(F. Verstraete and J. Cirac, arXiv:cond-mat/0407066)

d-dimensional tensor network representation
for the wave function of a d-dimensional quantum system

$$|\Psi\rangle = \sum_{\{m_i=1,2,\cdots,m\}} \mathrm{Tr}\ A_1[m_1]A_2[m_2]\cdots A_N[m_N]|m_1 m_2 \cdots m_N\rangle$$

Tr: tensor network "contraction"

$A_{x_i x_i' y_i y_i'}[m_i]$: Rank 4+1 tensor

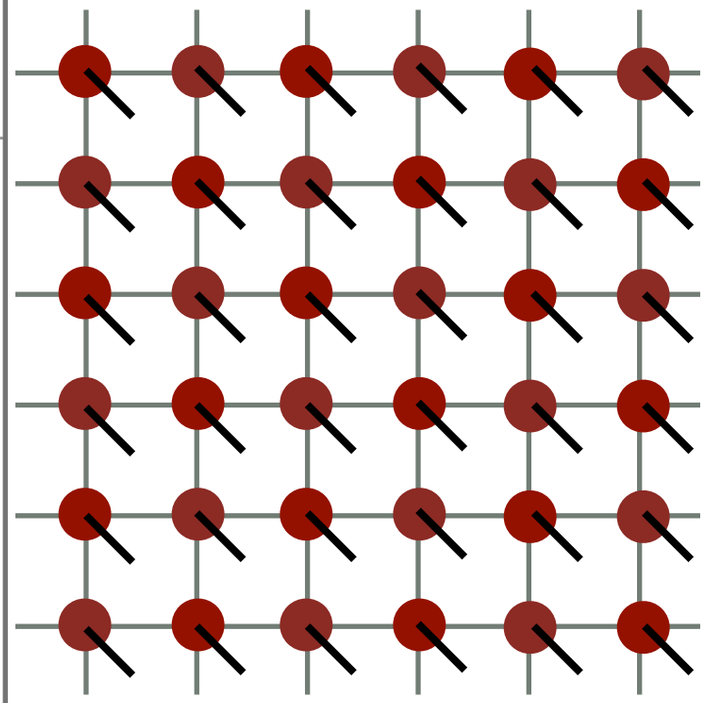


$x,y,x',y' = 1,2, \ldots D$ $\quad D =$ "bond dimension"

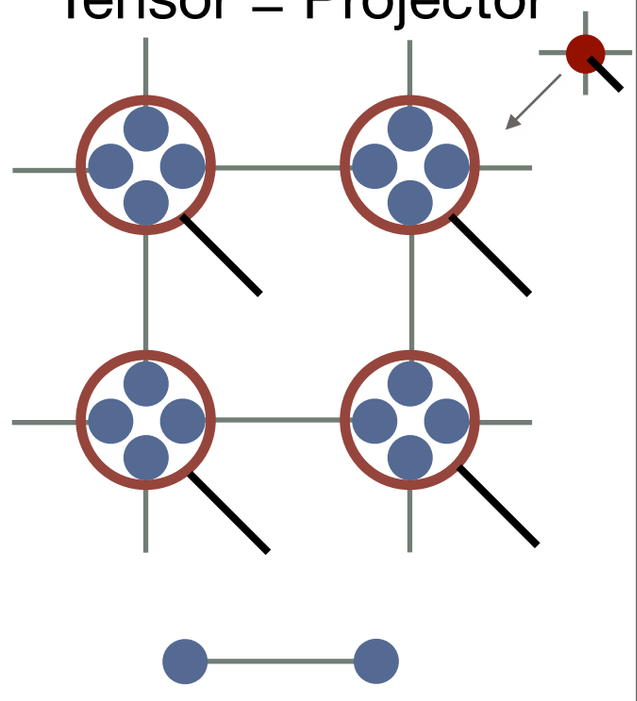
$m_i = 1,2, \ldots m$ $\quad m =$ dimension of the local Hilbert space

*D can be larger than m. "Virtual state "

TPS on square lattice

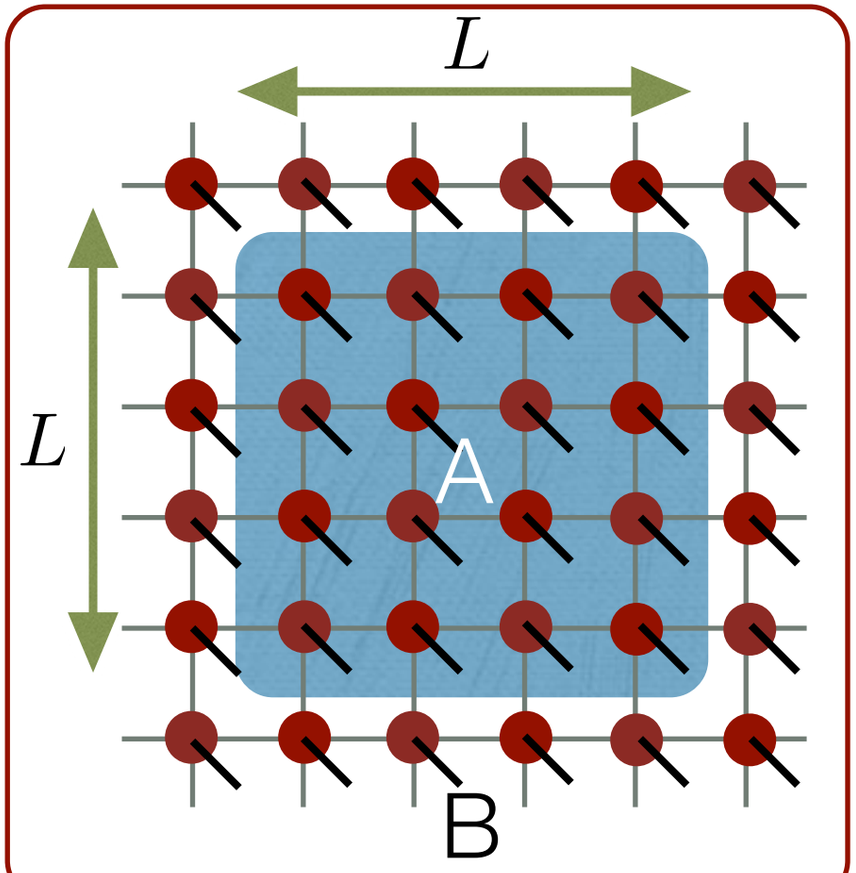


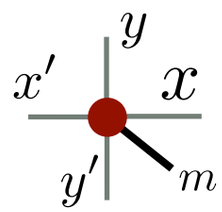
Tensor = Projector



Maximally entangled state
between D-state spins

Entanglement entropy of TPS (PEPS)





Bond dimension = D

# of bonds connecting regions A and B

$N_c(L) = 4L$ (square lattice)

$N_c(L) = 2dL^{d-1}$ (d-dimensional hyper cubic lattice)

$$\text{rank } \rho_A \leq D^{N_c(L)} \sim D^{2dL^{d-1}}$$

$$S_A = -\text{Tr } \rho_A \log \rho_A \leq 2dL^{d-1} \log D$$

TPS can satisfies the area law even for d >1.

We can efficiently approximate vectors in higher dimensional space by TPS.

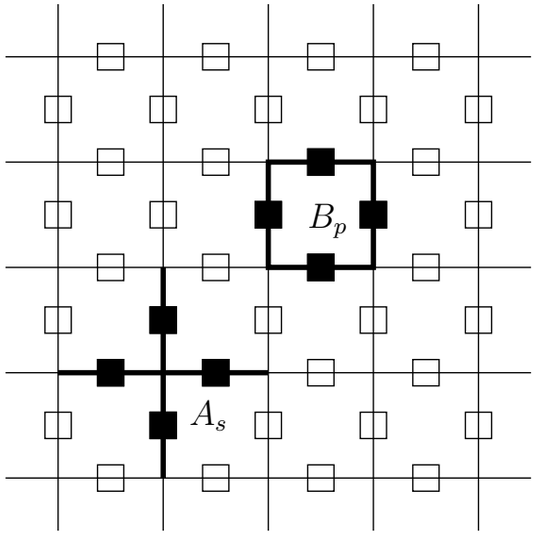
* Similar to the MPS in 1d, TPS can approximate infinite system!

Example: Ground state represented by TPS

Toric code model (A. Kitaev, Ann. Phys. **303**, 2 (2003).

$$\mathcal{H} = -\sum_s A_s - \sum_p B_p$$

$$A_s = \prod_{j \in \mathrm{star}(s)} \sigma_j^x \qquad B_p = \prod_{j \in \partial p} \sigma_j^z.$$

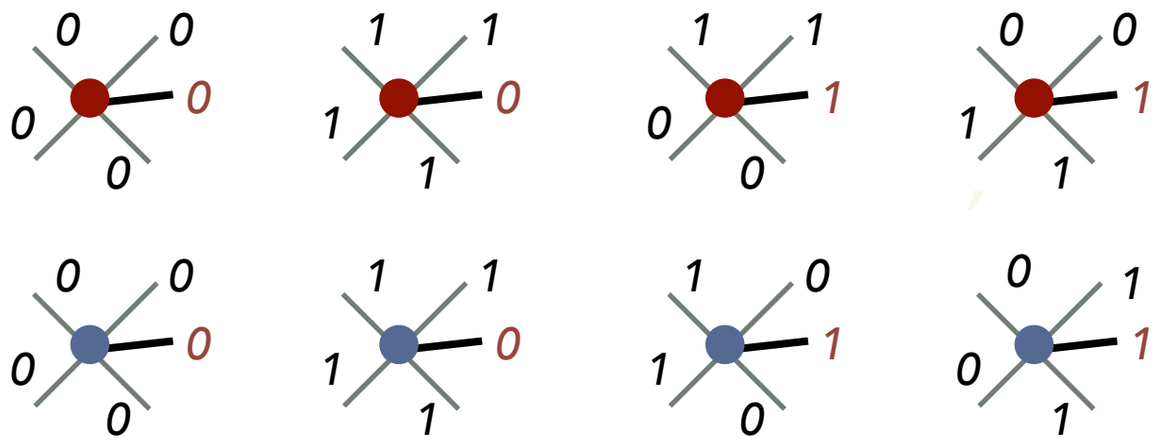


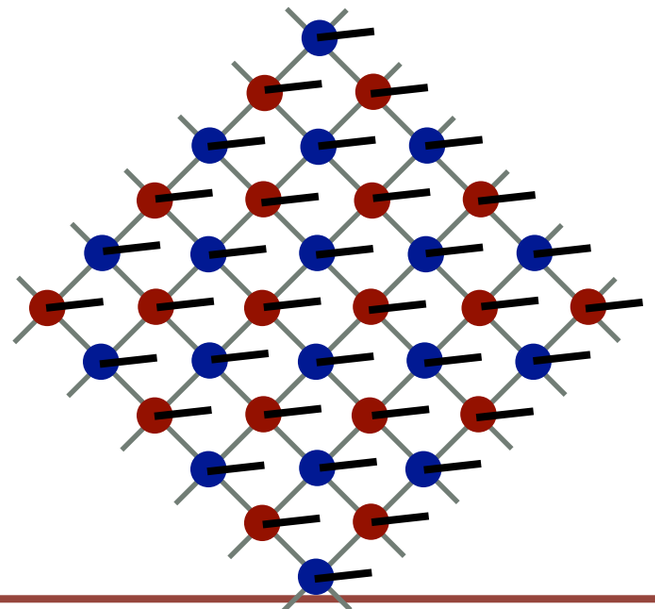
Its ground state is so called Z_2 spin liquid state.

"Spin liquid" is a novel phase different from conventional magnetic orders.

It can be represented by D=2 TPS. (F. Verstraete, et al, Phys. Rev. Lett. **96**, 220601 (2006).

0,1: eigen state of σ_x (Non-zero elements of tensor)





Difference between MPS and TPS

Cost of tensor network contraction:

d-dimensional cubic lattice $N = L^d$

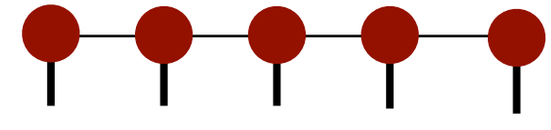
MPS: $O(N)$

TPS: $O(e^{L^{d-1}})$

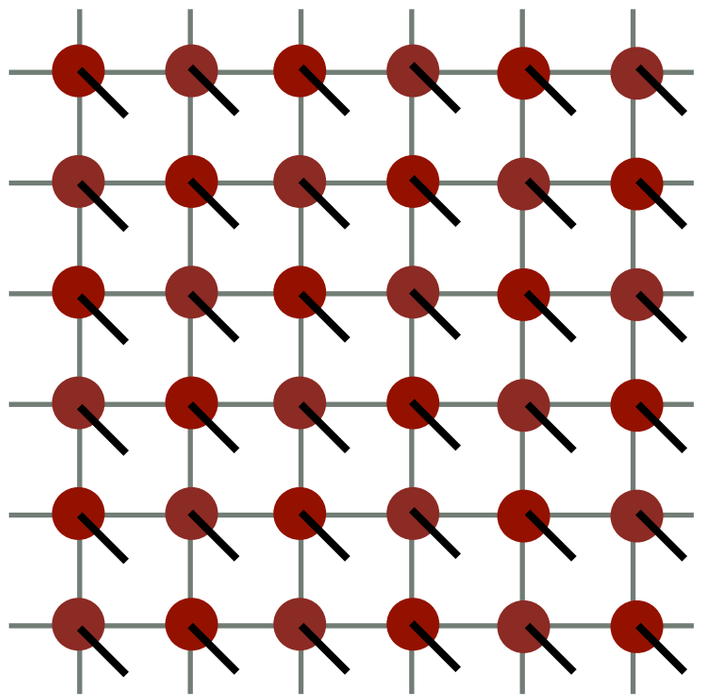
It is impossible to perform exact contraction even if we know local tensors in the case of TPS.

In the case of TPS, usually we approximately calculate the contraction.

MPS



TPS (PEPS)



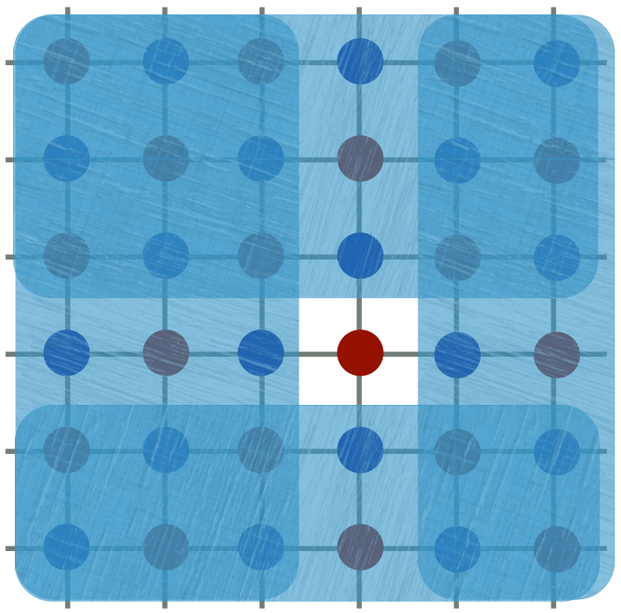
Example of approximate contraction: CTM method

For (infinite) open boundary system

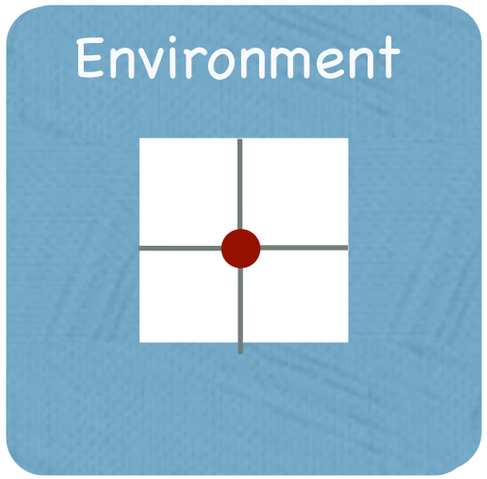
(T. Nishino and K. Okunishi, JPSJ **65**, 891 (1996))
(R. Orus *et al*, Phys. Rev. B **80**, 094403 (2009))

Infinite PEPS

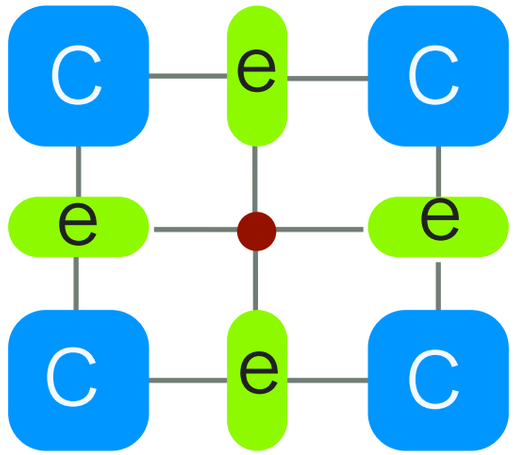
(with a translational invariance)



Environment

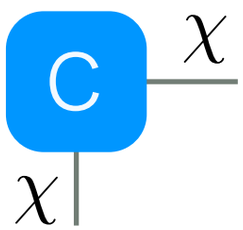


Corner transfer matrix Representation

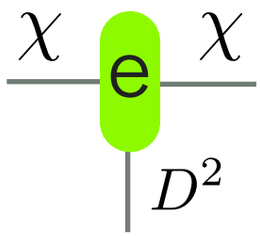


Double tensor

Corner transfer matrix



Edge tensor

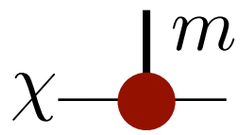


Mapping to a "classical" system

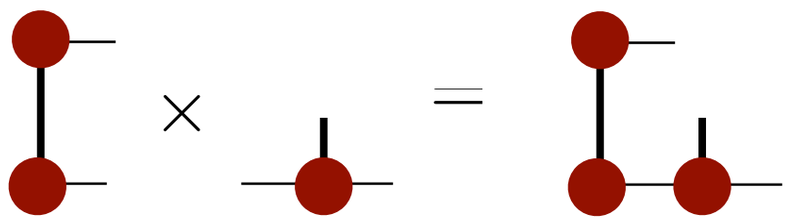
χ =bond dimension $\chi \sim D^2$

Cost of (approximate) contraction

MPS:

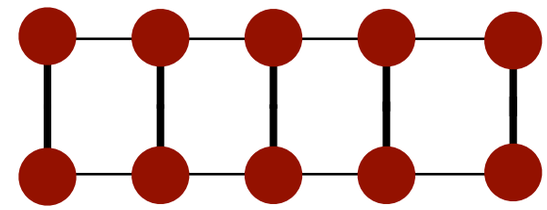


$(m \ll \chi)$

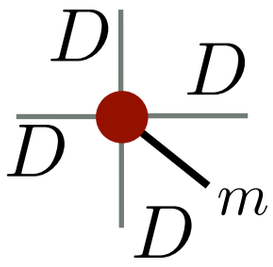


$O(\chi^3)$

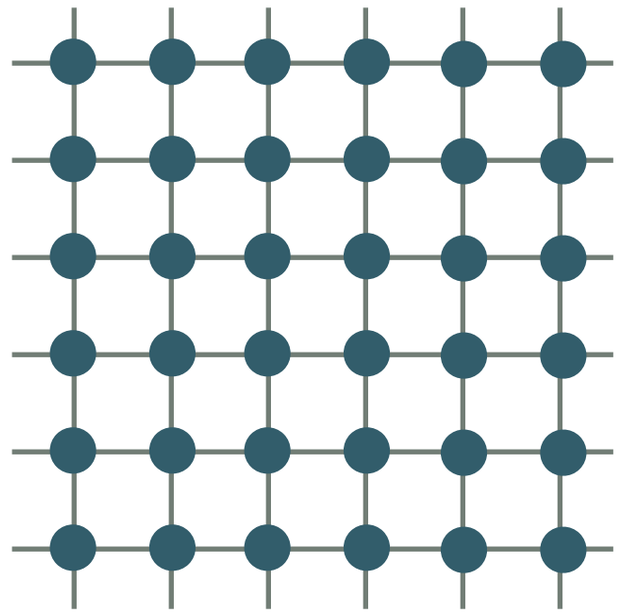
$\langle \Psi | \Psi \rangle =$



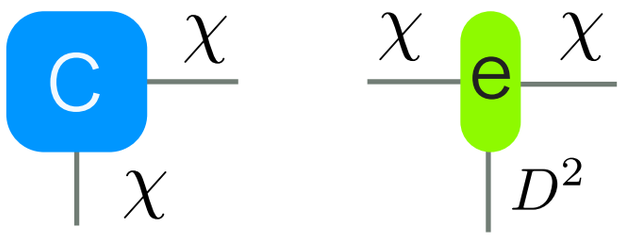
TPS:



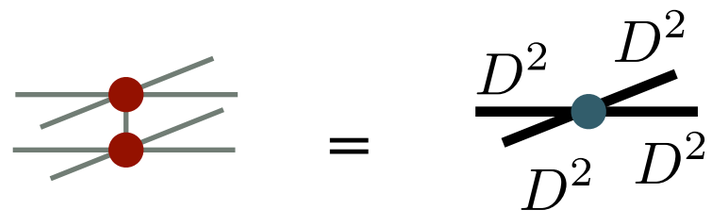
$\langle \Psi | \Psi \rangle =$



When we use CTM environment in 2D,



$$O(\chi^2 D^6), O(\chi^3 D^4) \sim O(D^{10}) \quad (\chi \sim D^2)$$



We can treat very small bond dimensions in TPS!

Application of TPS to eigenvalue problem

For calculation of minimum eigenvalues and its eigenvector,
we can use similar techniques to those in MPS

Variational method:

(P. Corboz, Phys. Rev. B **94**, 035133 (2016))
(L. Vanderstraeten, Phys. Rev. B **94**, 155123 (2016))

minimize cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

Imaginary time evolution:

Simulate imaginary time evolution:
（虚時間発展）

$$|\Psi_{\mathrm{GS}}\rangle \propto \lim_{\beta\to\infty} e^{-\beta\mathcal{H}}|\Psi_0\rangle$$

For a initial state $\langle\Psi_{\mathrm{GS}}|\Psi_0\rangle \neq 0$

(H. G. Jiang *et al*, Phys. Rev. Lett. **101**, 090603 (2008))
(J. Jordan *et al*, Phys. Rev. Lett. **101**, 250602 (2008))

Example of application: Honeycomb lattice Kitaev Model

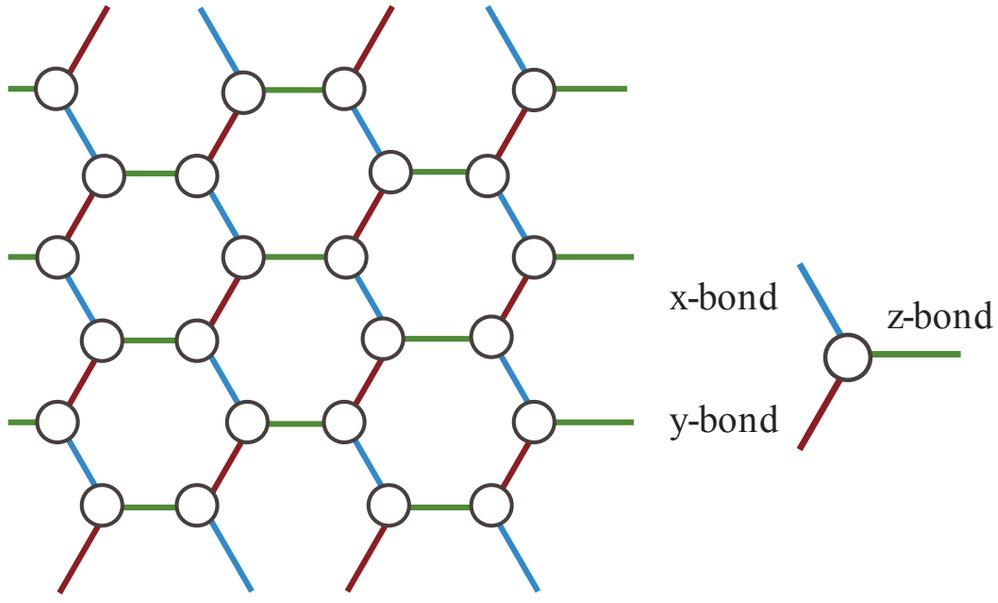
A. Kitaev, Annals of Physics 321, 2 (2006)

Kitaev model

$$\mathcal{H} = -\sum_{\gamma,\langle i,j\rangle_\gamma} J_\gamma S_i^\gamma S_j^\gamma$$

γ:bond direction

Depending on the bond direction, only specific spin components interact.



Exactly solvable by introducing Majorana fermion

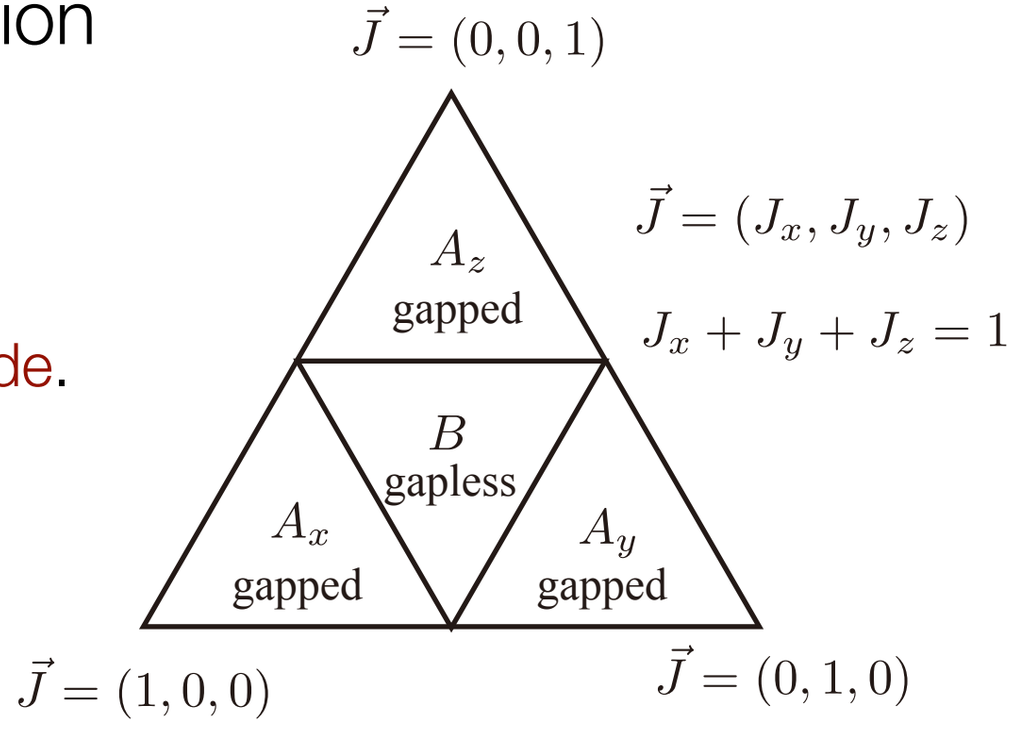
Isotropic region (B) : gapless spin liquid

Anisotropic region (A) : gapped spin liquid

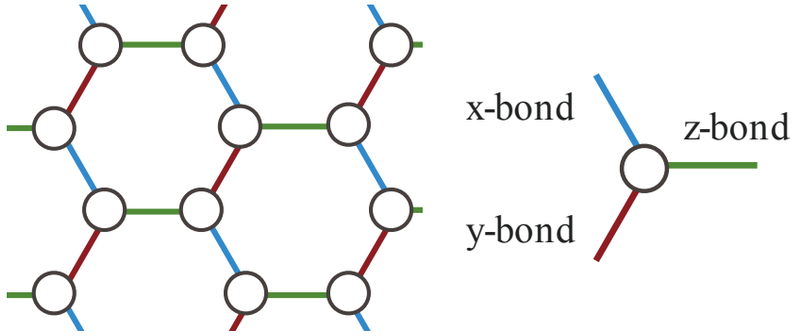
Cf. The anisotropic limit corresponds to the Toric code.

*Recently, researchers have realized that this type of models might appear in real materials. Hot topic!

Phase diagram



Application : Kitaev spin liquid



Honeycomb lattice Kitaev model

$$\mathcal{H} = -\sum_{\gamma,\langle i,j\rangle_\gamma} J_\gamma S_i^\gamma S_j^\gamma \qquad (\gamma = x, y, z)$$

At $J_x = J_y = J_z$, the ground state is

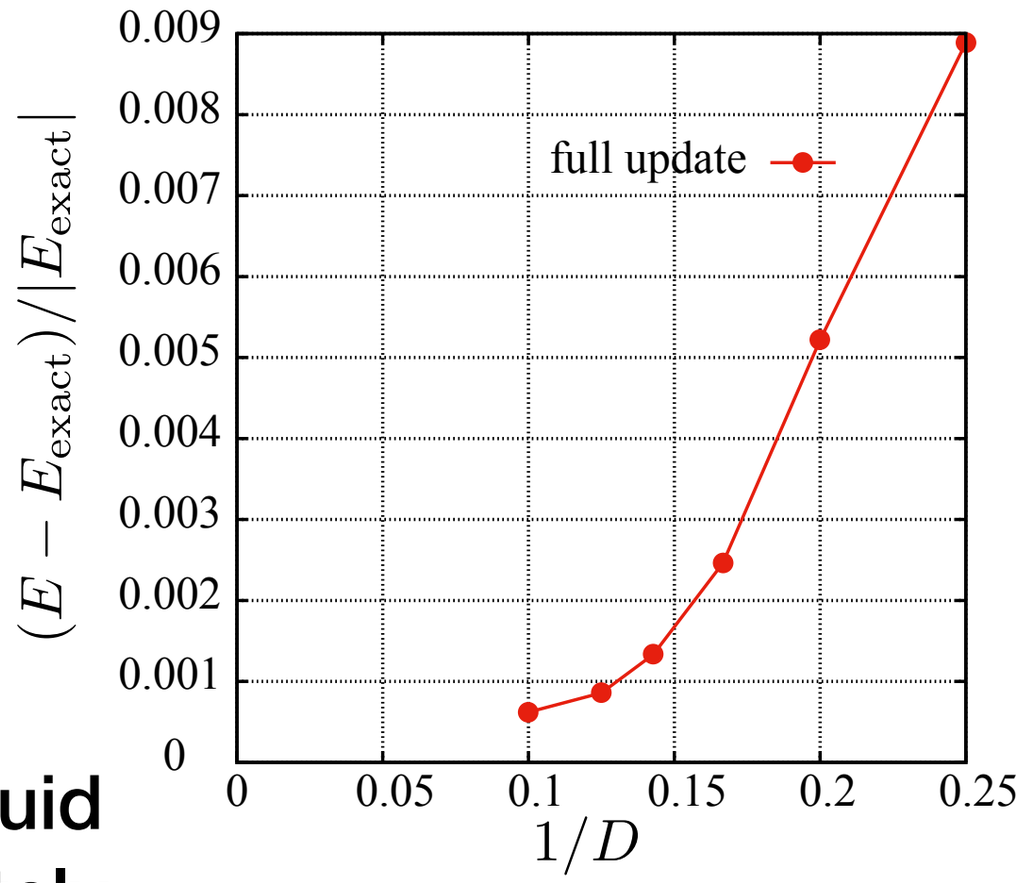
a gapless spin liquid.

In the present (super)computers,
we can access around D=10 (maybe 16)
by using massively parallel code.

The error of the ground state
energy is **less than 10^{-3}**
for infinite system!

Energy error obtained by iTPS

(T. okubo et al, unpublished)



iTPS can represent Kitaev spin liquid
in the thermodynamic limit accurately.

Interesting topics related to TPS

- Application to itinerant electron system, which may break the area law

  (P. Corboz et al, Phys. Rev. B. **81**, 165104 (2010))

  (P. Corboz, Phys. Rev. B. **93**, 045116 (2016))

- Characterization of topologies in wave function

  - Symmetric tensor network and modular matrix

    (J.-W. Mei et al, Phys. Rev. B. **95**, 235107 (2017))

- Application to three dimensions

  - So far, there is no practical calculations for non-trivial models.

  - Mainly, due to the scaling: $O(D^{18})$?

Tentative lecture schedule

1日目

1. 現代物理学における巨大なデータと情報圧縮
2. 格子スピン模型の統計力学
3. 線形代数の復習

2日目

4. 特異値分解と低ランク近似
5. テンソルネットワーク繰り込みによる情報圧縮
6. 情報のエンタングルメントと行列積表現

3日目

7. 行列積表現の固有値問題への応用
8. テンソルネットワーク表現への発展

Optional

9. フラストレート磁性体への応用

フラストレート磁性体への応用

（optional: 時間があれば。）

フラストレートレーション

- 複数の最適化条件を同時に満たせない状態
- あちらを立てればこちらが立たず

磁性体におけるフラストレーション

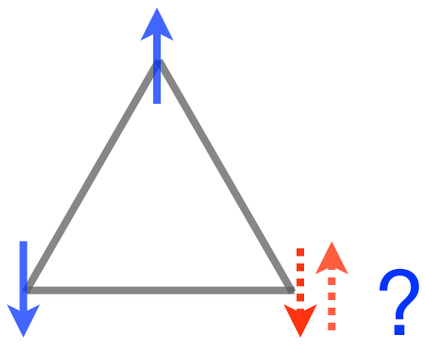
最適化条件：エネルギーを最小にしたい

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \qquad J < 0$$

反強磁性

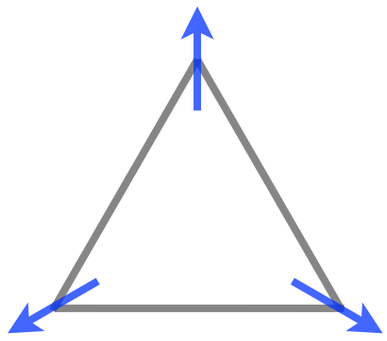
局所的なエネルギー最小：各ペアでスピンを反平行

イジングスピン



全ての辺で同時に
スピンを反平行に出来ない

ベクトルスピン



傾いた磁気秩序
（120°構造）

フラストレートスピン系

・古典スピン模型：
  - 大規模な基底状態の縮重
  - 弱い摂動による新規秩序
  - 傾いた磁気秩序

・量子スピン模型：
  - スピン液体
  - 隠れた秩序、トポロジー

フラストレート量子スピン系の数値計算

☆フラストレート量子スピン系における数値計算の課題

熱力学極限に迫れる手法が少ない

- 量子モンテカルロ法：　負符号問題のため、精度が出ない
- 厳密対角化法　　：　厳密だが、少数クラスターしか取り扱えない
- 変分モンテカルロ法：　大きな系を扱えるが、バイアスが存在

↓

テンソルネットワーク法

変分波動関数として、（基底状態の）波動関数を“効率的”に表現した

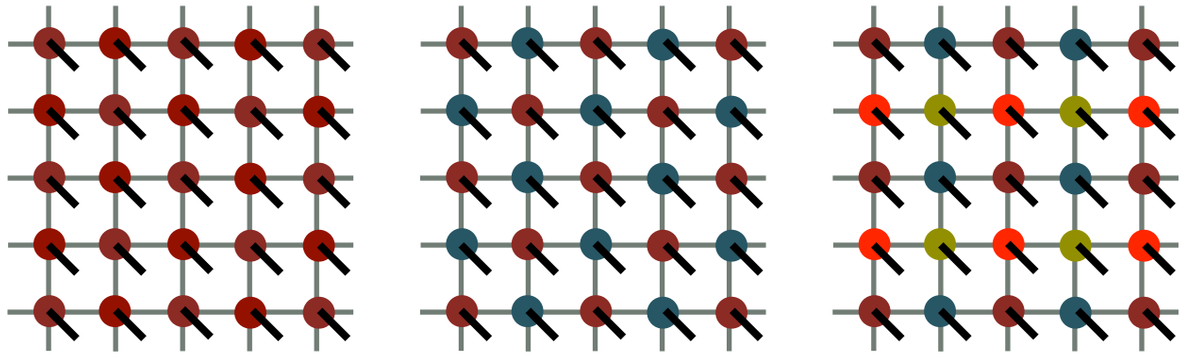
テンソルネットワーク状態を用いる

→ ・フラストレーションがあっても問題なし！
・無限系の波動関数を直接取り扱える！

変分波動関数としてのTPS

- Area lawを満たせるので、ボンド次元Dを十分に大きい有限の値にとれば、多くの基底状態を表現できる
  - 並進対称性を仮定すれば、無限系でさえ有限のDで計算可能

並進対称性の例：



- テンソルの最適化、期待値の計算の手法がかなり発達してきた
  - スピン液体相等の非自明な基底状態でもOK
    e.g. ハニカム格子上のKitaev スピン液体

J. O. Iregui, P. Corboz, and M. Troyer, et al., PRB 90, 195102 (2014).

Application of TPS to eigenvalue problem

For calculation of minimum eigenvalues and its eigenvector,
we can use similar techniques to those in MPS

Variational method:

(P. Corboz, Phys. Rev. B **94**, 035133 (2016))
(L. Vanderstraeten, Phys. Rev. B **94**, 155123 (2016))

minimize cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

Imaginary time evolution:

Simulate imaginary time evolution:
（虚時間発展）

$$|\Psi_{\mathrm{GS}}\rangle \propto \lim_{\beta\to\infty} e^{-\beta\mathcal{H}}|\Psi_0\rangle$$

For a initial state $\langle\Psi_{\mathrm{GS}}|\Psi_0\rangle \neq 0$

(H. G. Jiang *et al*, Phys. Rev. Lett. **101**, 090603 (2008))
(J. Jordan *et al*, Phys. Rev. Lett. **101**, 250602 (2008))

Application1: Honeycomb lattice Kitaev Model

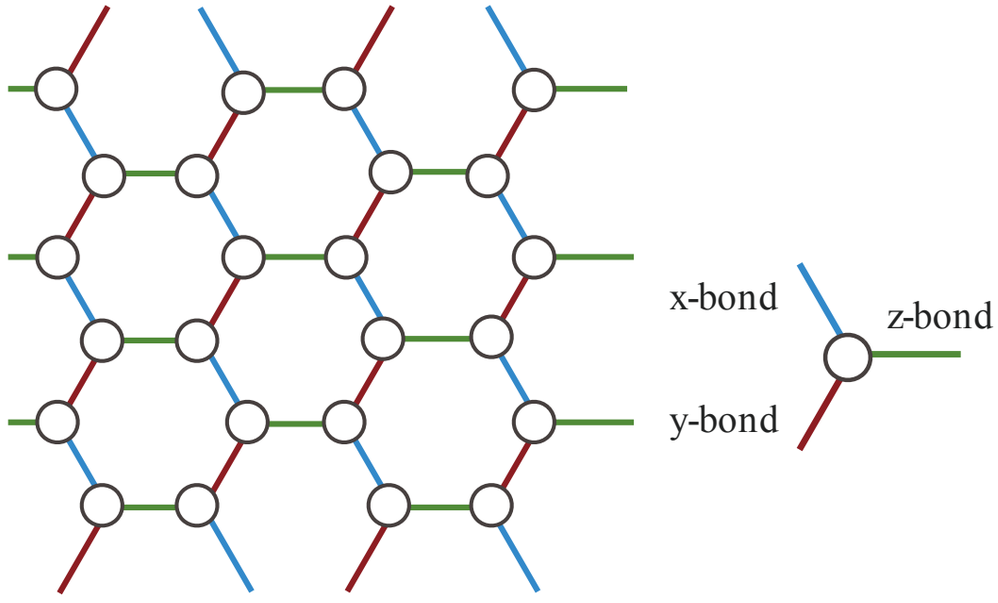
A. Kitaev, Annals of Physics 321, 2 (2006)

Kitaev model

$$\mathcal{H} = -\sum_{\gamma,\langle i,j\rangle_\gamma} J_\gamma S_i^\gamma S_j^\gamma$$

γ:bond direction

Depending on the bond direction, only specific spin components interact.



Exactly solvable by introducing Majorana fermion

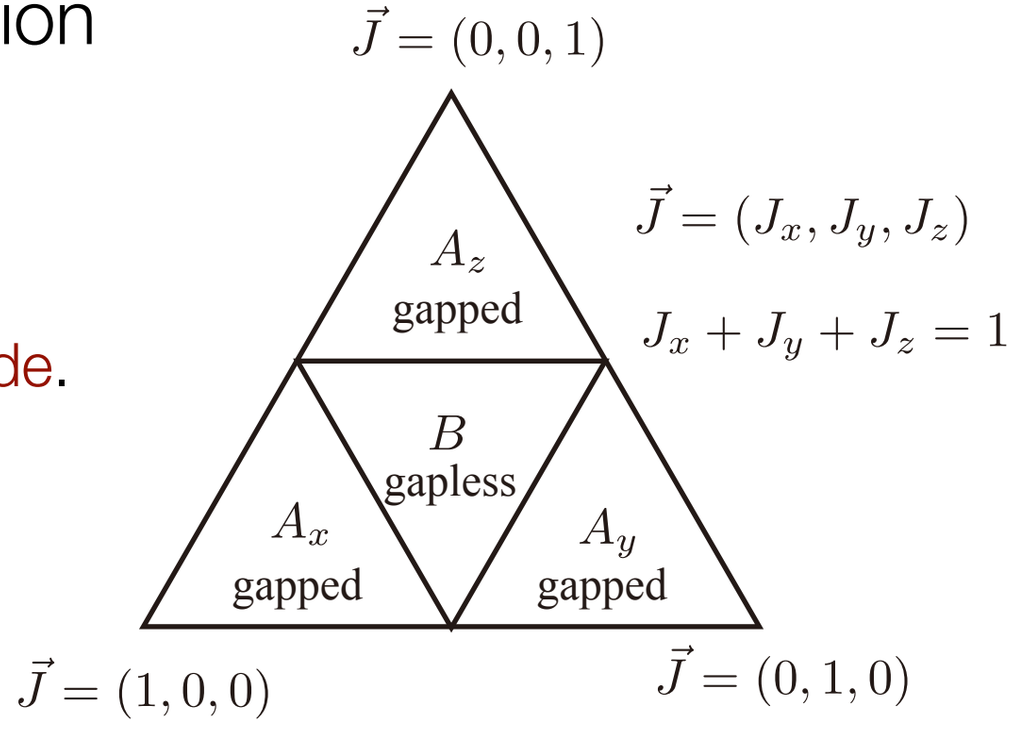
Isotropic region (B) : gapless spin liquid

Anisotropic region (A) : gapped spin liquid

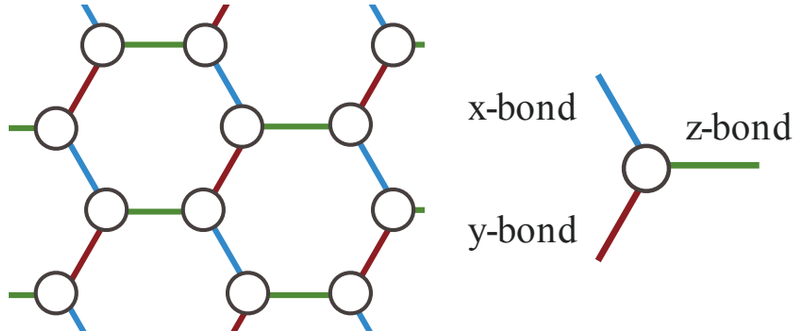
Cf. The anisotropic limit corresponds to the Toric code.

*Recently, researchers have realized that this type of models might appear in real materials. Hot topic!

Phase diagram



Application1 : Kitaev spin liquid



Honeycomb lattice Kitaev model

$$\mathcal{H} = -\sum_{\gamma,\langle i,j\rangle_\gamma} J_\gamma S_i^\gamma S_j^\gamma \qquad (\gamma = x, y, z)$$

At $J_x = J_y = J_z$, the ground state is

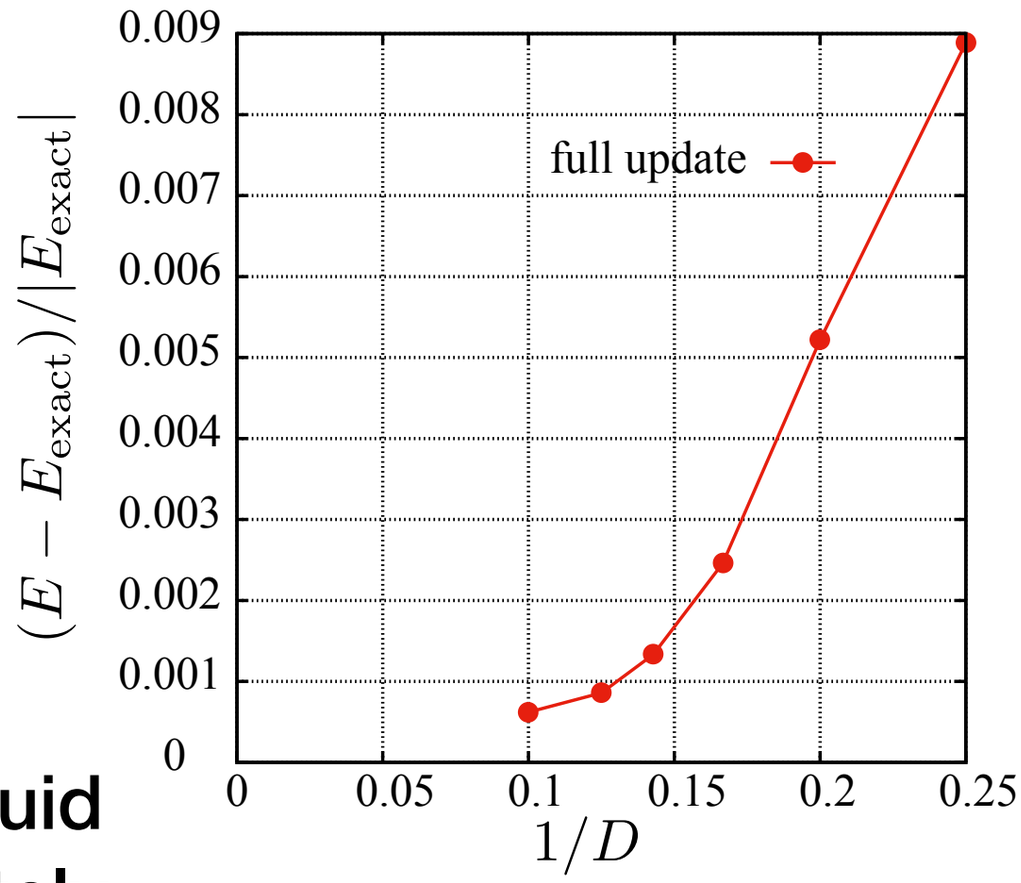
a gapless spin liquid.

In the present (super)computers,
we can access around D=10 (maybe 16)
by using massively parallel code.

The error of the ground state
energy is **less than 10^{-3}**
for infinite system!

Energy error obtained by iTPS

(T. okubo et al, unpublished)



iTPS can represent Kitaev spin liquid
in the thermodynamic limit accurately.

A_2IrO_3

Iridium Oxides

Strong spin-orbit coupling → Effective `spin" moment: $J_{\text{eff}} = \frac{1}{2}$

Na_2IrO_3, Li_2IrO_3

G.Jackeli, et al., PRL 102, 017205 (2009)
J. Chaloupka, et al., PRL 105, 027204 (2010)

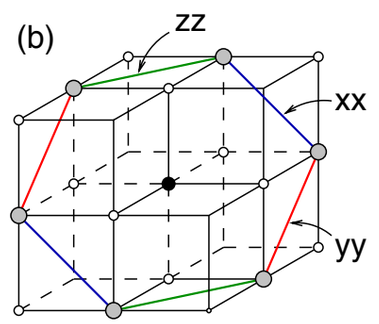
Ir ions form an honeycomb lattice.

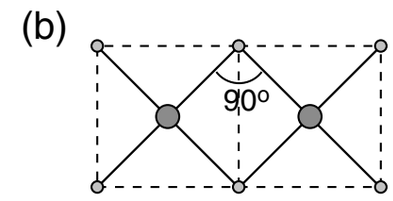
Ir - Ir direct exchange: Heisenberg interaction

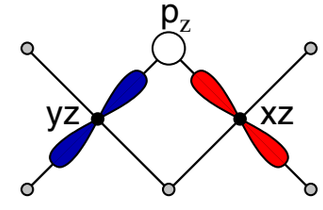
Ir - O - Ir exchange: Anisotropic Kitaev interaction

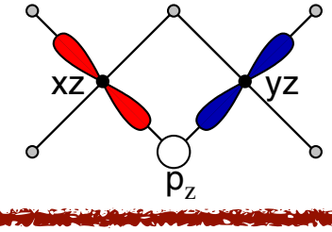
Depending on the bond direction, only specific spin component interact.

$$H_K^{(\gamma)} = -JS_i^\gamma S_j^\gamma$$









ab initio Hamiltonian of Na_2IrO_3

(Y. Yamaji et al. Phys. Rev. Lett. **113**, 107201(2014))

ab initio Hamiltonian

$$\hat{H} = \sum_{\Gamma=X,Y,Z} \sum_{\langle \ell,m \rangle \in \Gamma} \vec{\hat{S}}_\ell^T \mathcal{J}_\Gamma \vec{\hat{S}}_m,$$

$$\mathcal{J}_Z = \begin{bmatrix} J & I_1 & I_2 \\ I_1 & J & I_2 \\ I_2 & I_2 & K \end{bmatrix}, \mathcal{J}_X = \begin{bmatrix} K' & I_2'' & I_2' \\ I_2'' & J'' & I_1' \\ I_2' & I_1' & J' \end{bmatrix}, \mathcal{J}_Y = \begin{bmatrix} J'' & I_2'' & I_1' \\ I_2'' & K' & I_2' \\ I_1' & I_2' & J' \end{bmatrix},$$

Kitaev coupling K and Heisenberg like coupling J

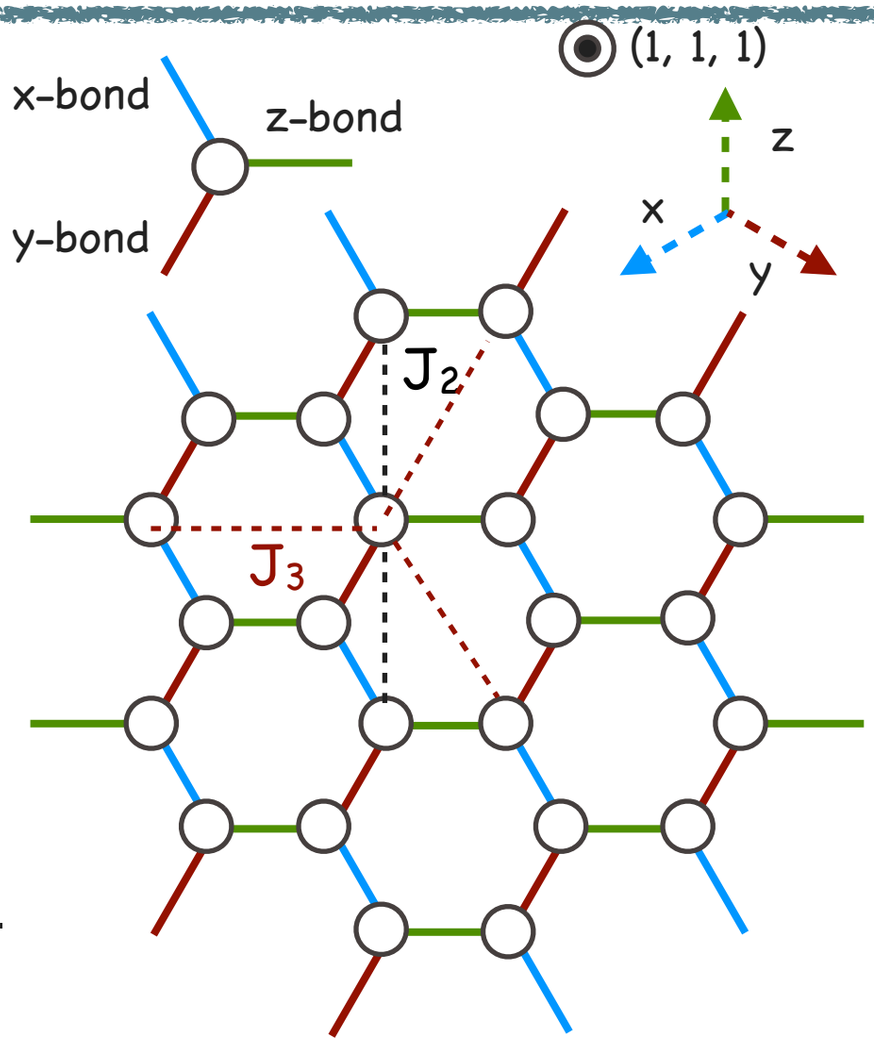
+

Off-diagonal couplings I_1 and I_2

It also contains J_2 and J_3 interaction term.

J_2 : only “z-bond” which is perpendicular to NN z-bond.

J_3 : all of the three third neighbors

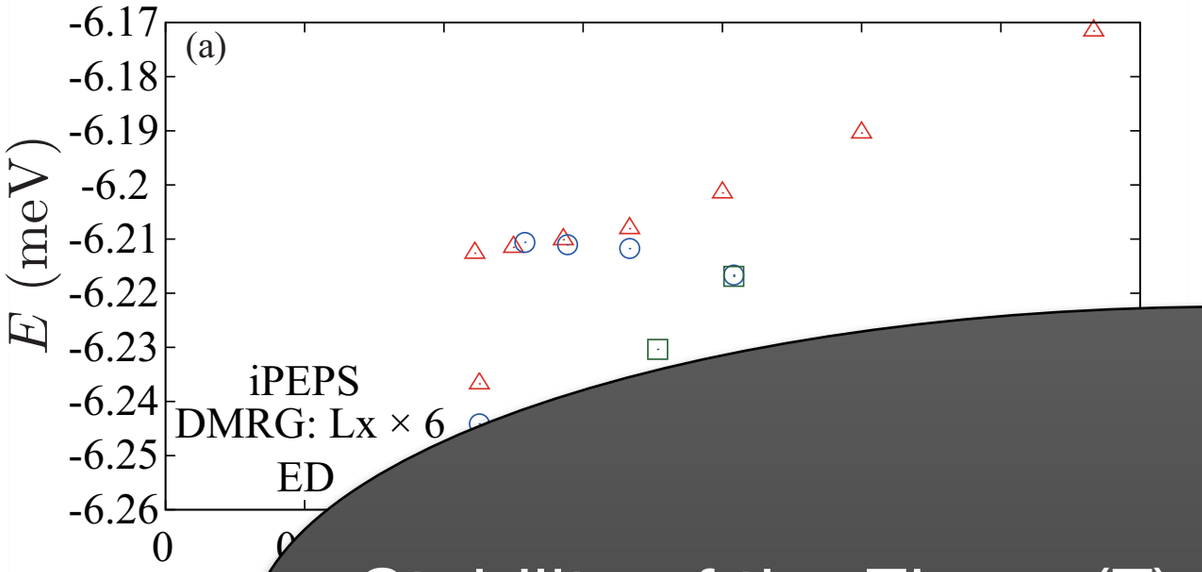


Due to the trigonal distortion, the *ab initio* Hamiltonian contains strong off-diagonal couplings, together with J_2 and J_3 interaction

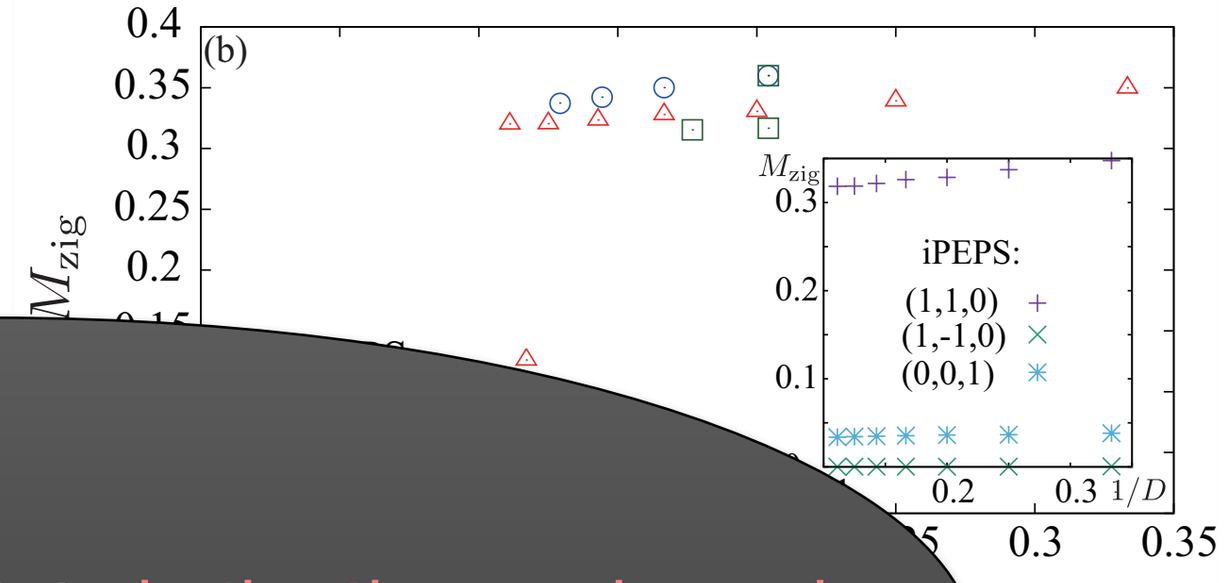
Results: comparison with other methods

T. Okubo *et al,* PRB **96**, 054434 (2017).

Energy (per sites)



Order parameter



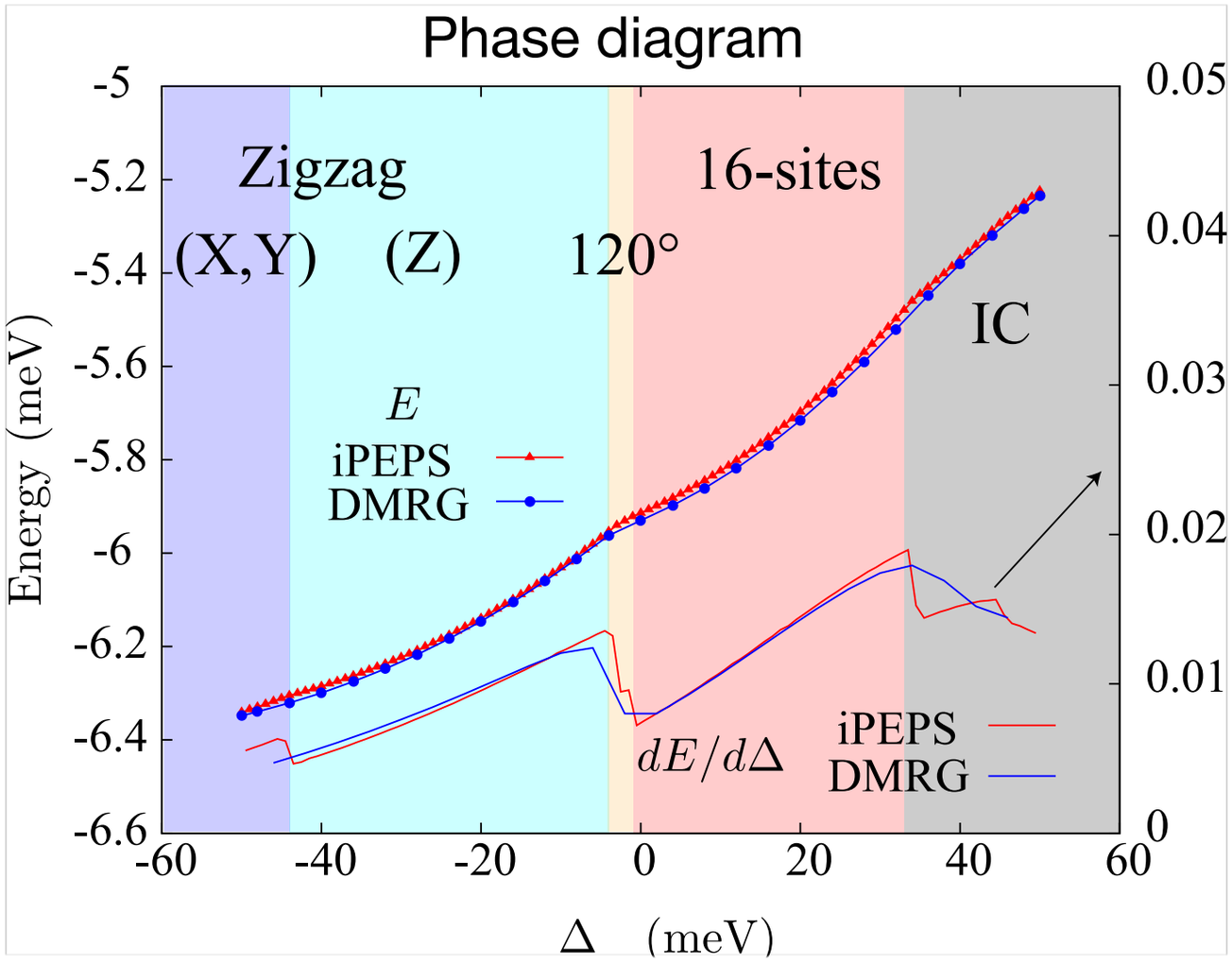
Stability of the Zigzag(Z) state in the thermodynamic limit is confirmed by iPEPS calculation.

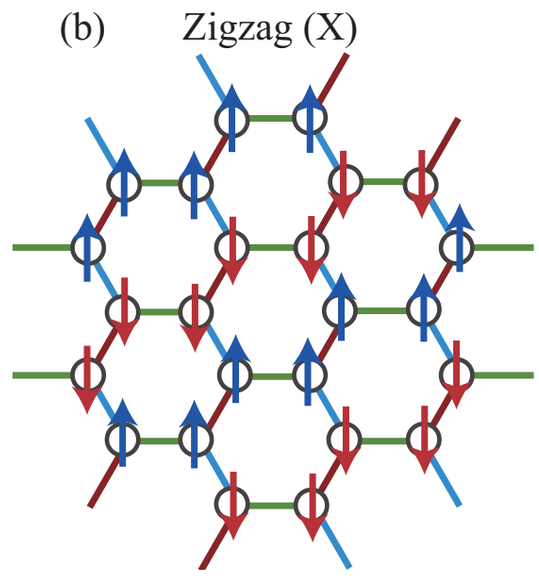
Energies of iP… consistent.

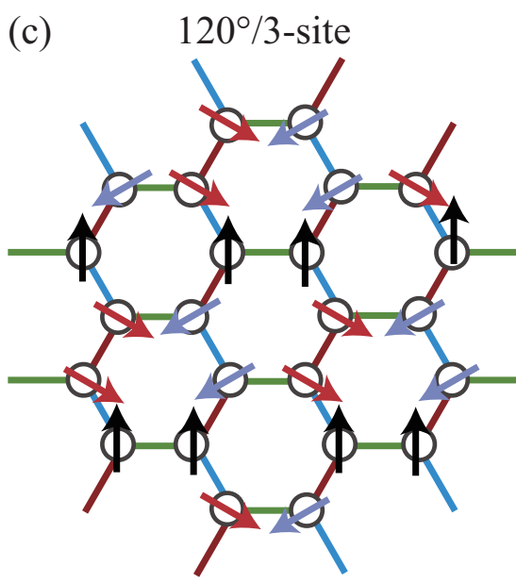
- For 4 × 6 latti… of them are <M> ~ 0.3 give almo… energy.
- Finite D of iPEPS and finite Lx of DMRG are overlapped.
- Spins are almost along (1,1,0) direction, which is consistent with the experimental observations.

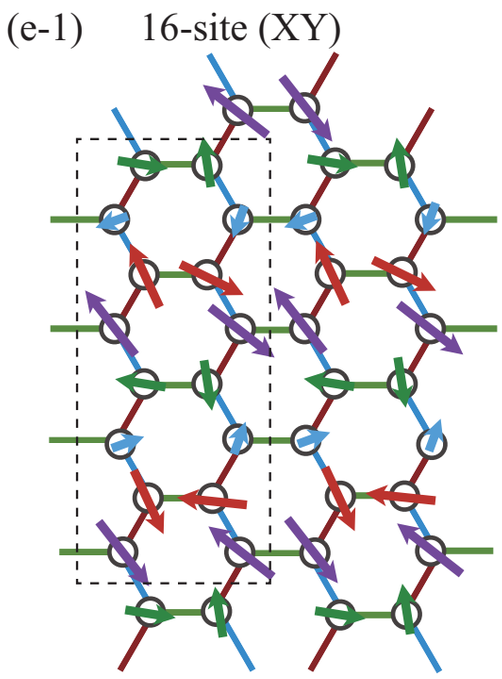
Phase diagram varying the trigonal distortion

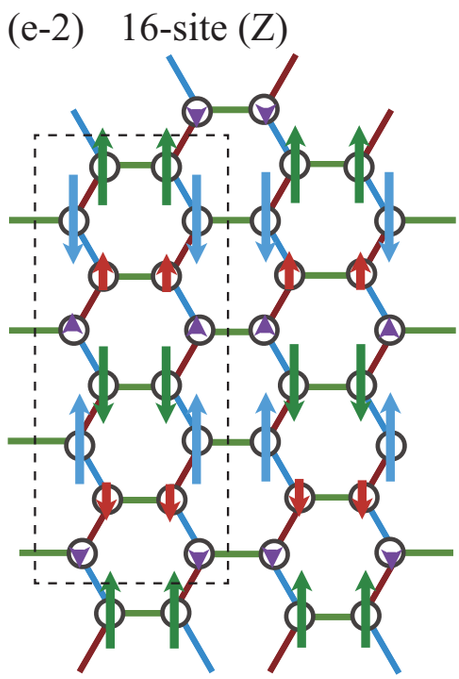
T. Okubo *et al,* PRB **96**, 054434 (2017).











DMRG: 6x8 cluster

iPEPS: 4x4, 2x6, 6x8, 8x12, 6x10 unit cells

- Energies obtained by iPEPS and DMRG are consistent
- New phases are stabilized compared with the previous ED reports

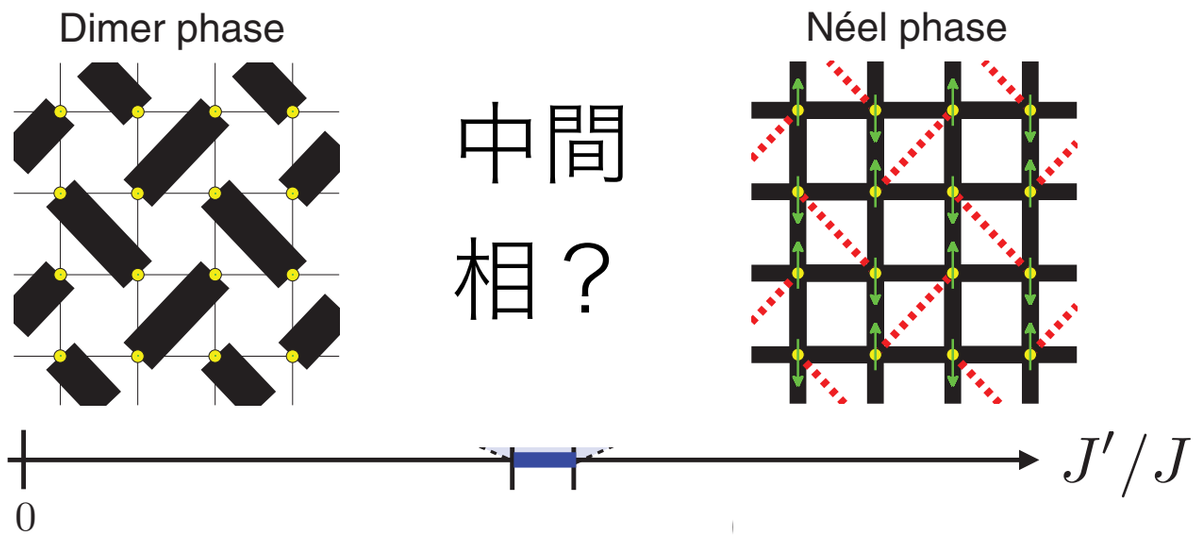
応用例2：Shastry-Sutherland 格子模型

P. Corboz and F. Mila, PRL **87**, 115144 (2013)

S=1/2 Shastry-Sutherland 格子ハイゼンベルグ模型

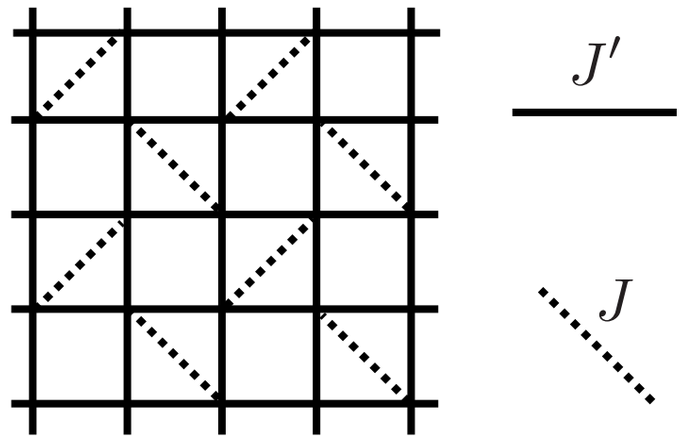
$$\mathcal{H} = J' \sum_{\langle i,j \rangle} S_i S_j + J \sum_{\langle\langle i,j \rangle\rangle} S_i S_j$$

基底状態相図

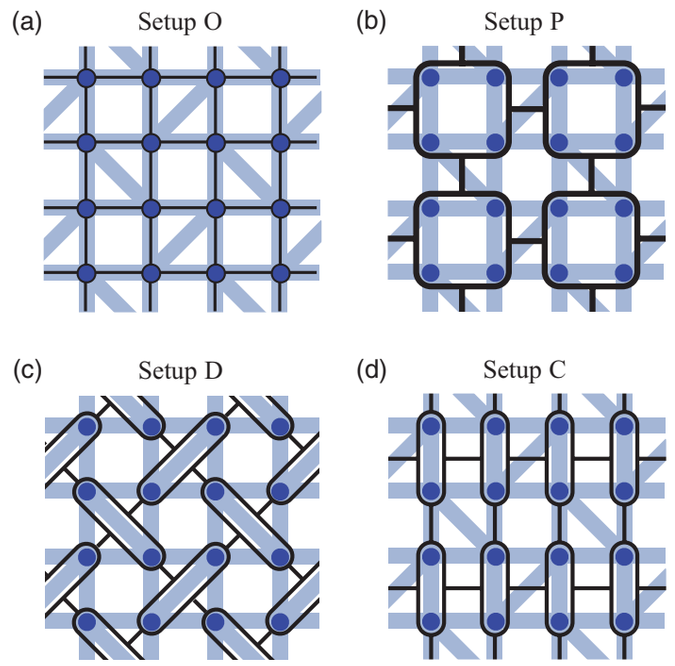


無限系のTPSを用いて解明！

Shastri-Sutherland 格子



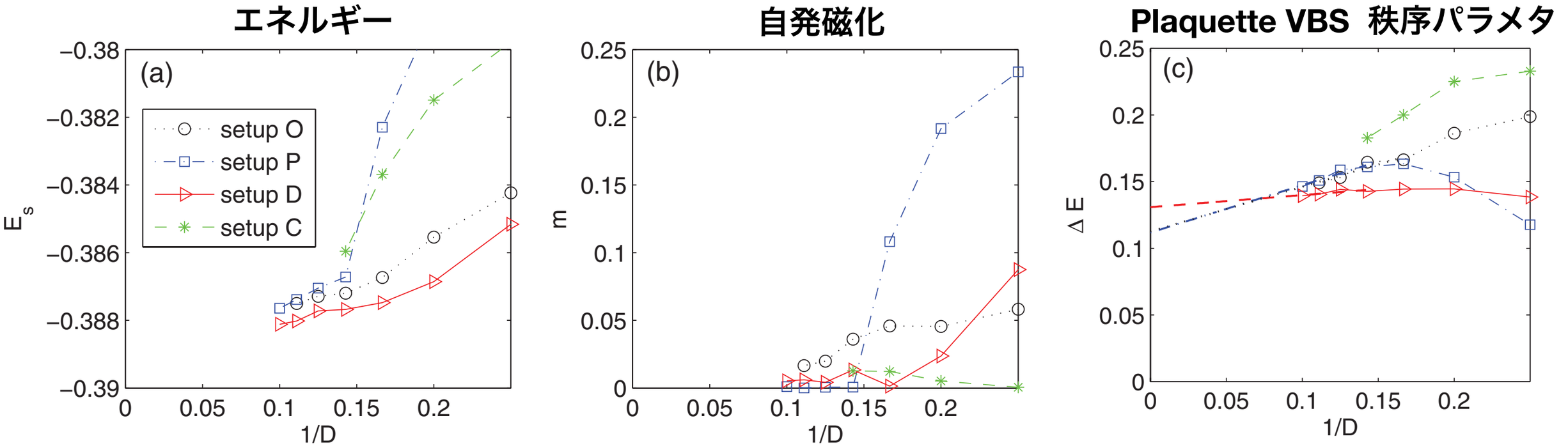
波動関数のiTPS表現

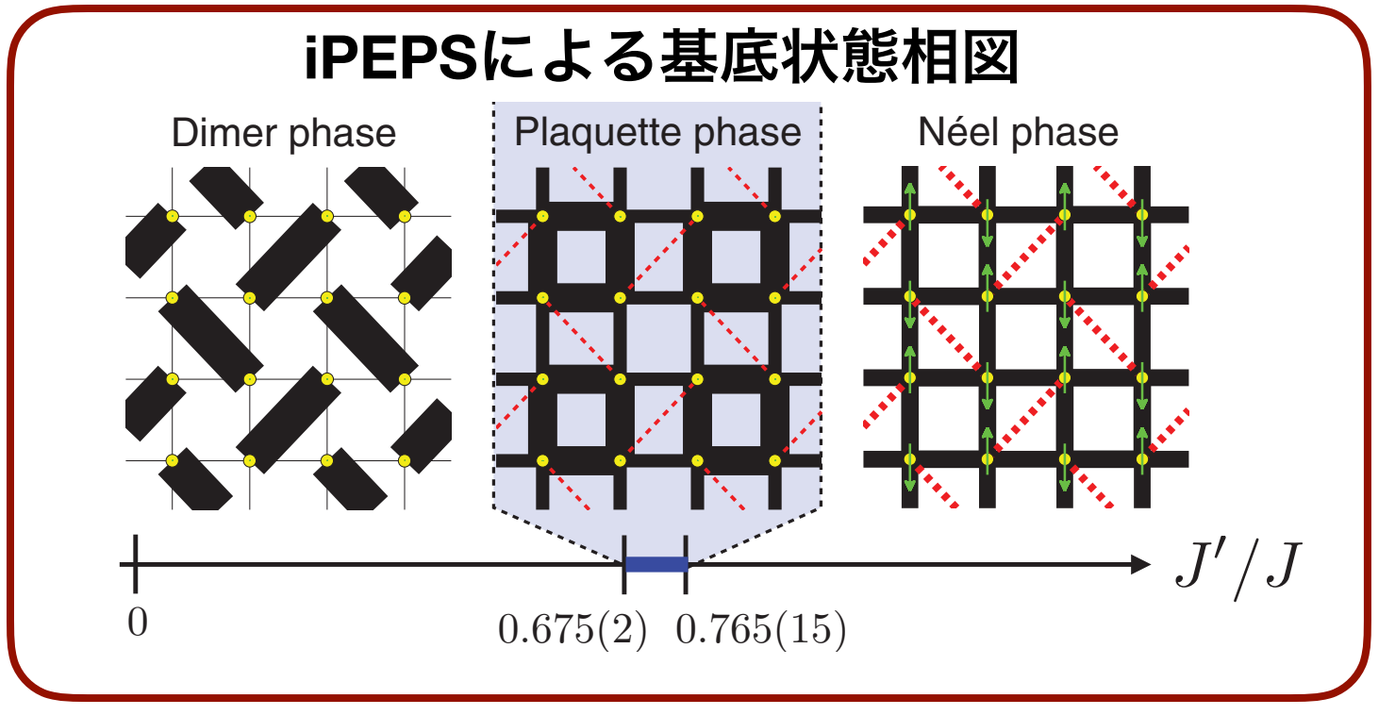


応用例2：Shastry-Sutherland 格子模型

P. Corboz and F. Mila, PRB **87**, 115144 (2013)

中間相での物理量





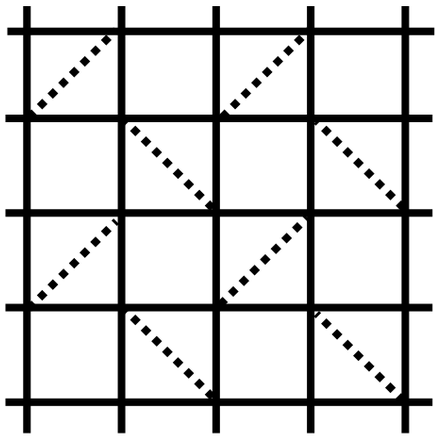
中間相は

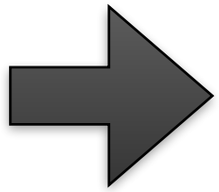
Plaquette型のVBS状態

応用例2：Shastry-Sutherland 格子模型の磁化過程

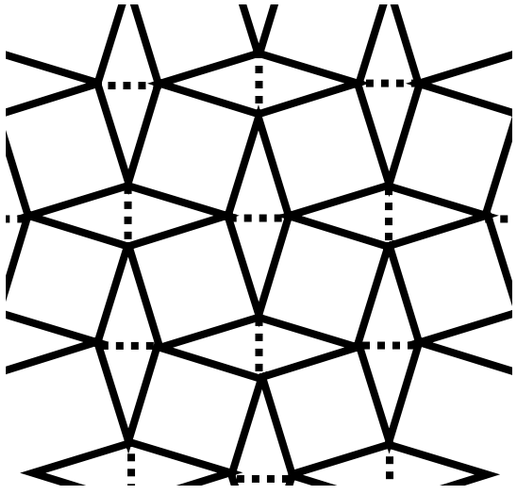
実際の物質に実現例が存在！

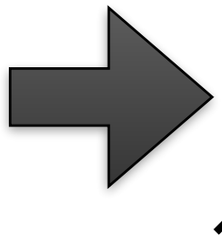
Shastri-Sutherland 格子





直交dimer





$SrCu(BO_3)_2$

H. Kageyama *et al,* PRL **82**, 3168 (1999)

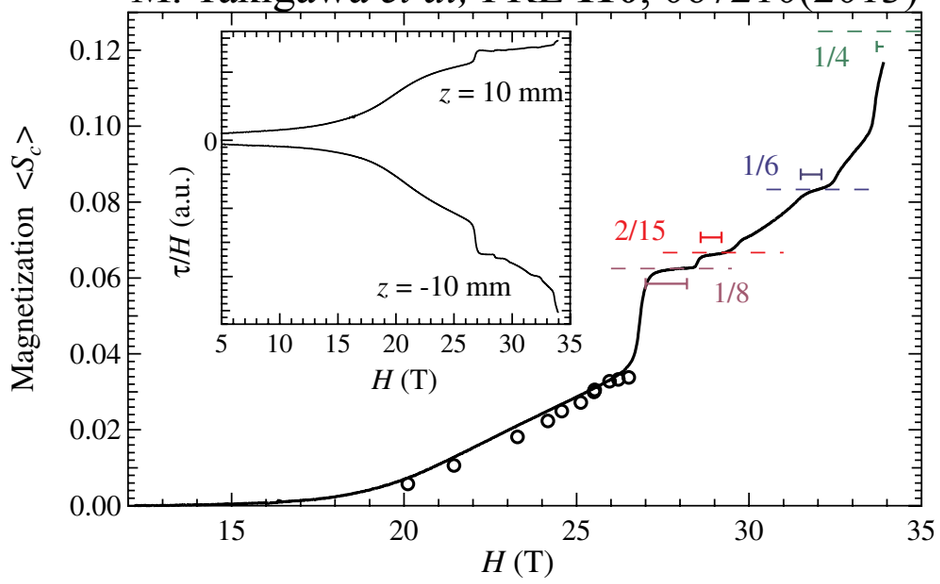
$SrCu(BO_3)_2$

- 基底状態はDimer 状態
- 磁化過程に複数の磁化プラトー

M/M_{Sat} = 1/8, 2/15, 1/6, 1/4, 1/3, 1/2

磁化過程

M. Takigawa *et al,* PRL **110**, 067210(2013)



応用例2：Shastry-Sutherland 格子模型の磁化過程

無限系のTPSによる計算

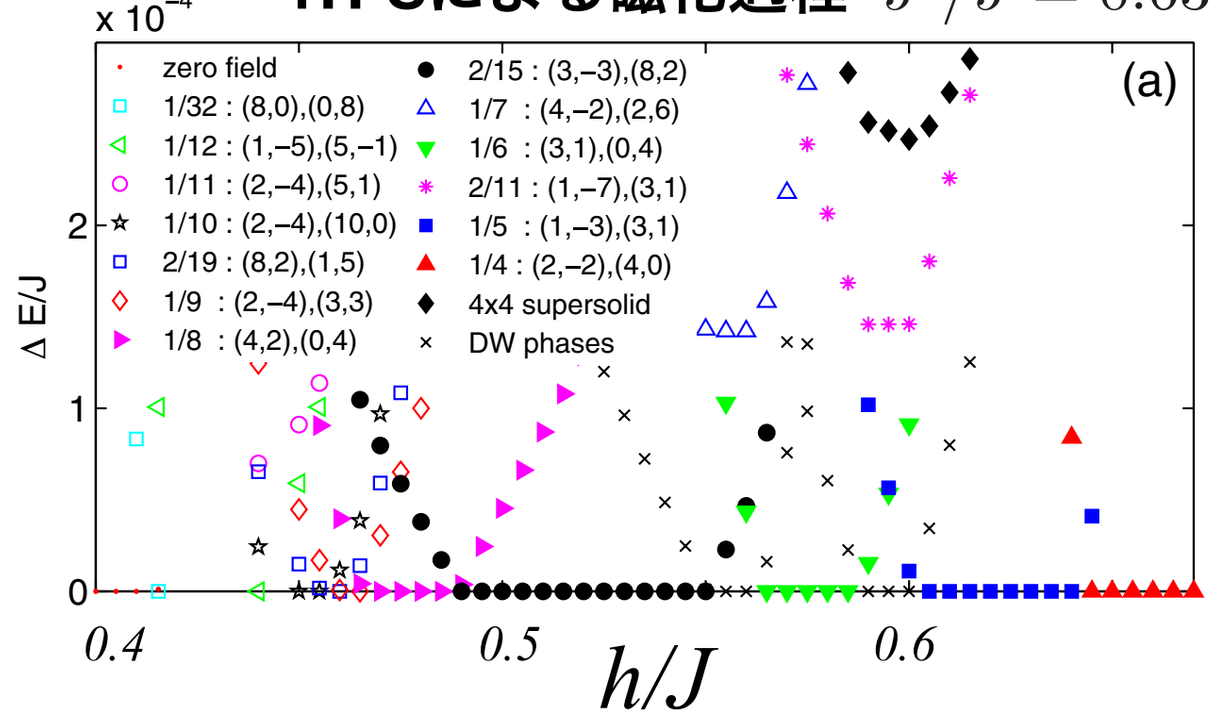
P. Corboz and F. Mila, PRL **112**, 147203 (2014)

cf. MERAの計算：J. Lou *et al,* arXiv:1212.1999

ポイント

波動関数に仮定する周期性を変えることで、

様々な並進対称性の破れを表現できる



複数の磁化プラトーが実現

M/M_{Sat} = 1/8, 2/15, 1/6, 1/5, 1/4,…

実験とよく一致！

＊DM相互作用により

実験に無い1/5プラトーは消失

実験： 1/8, 2/15, 1/6, 1/4, 1/3, 1/2

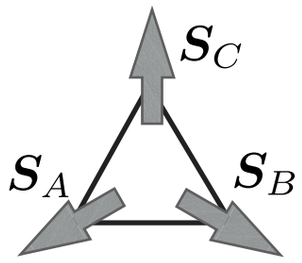
h/J

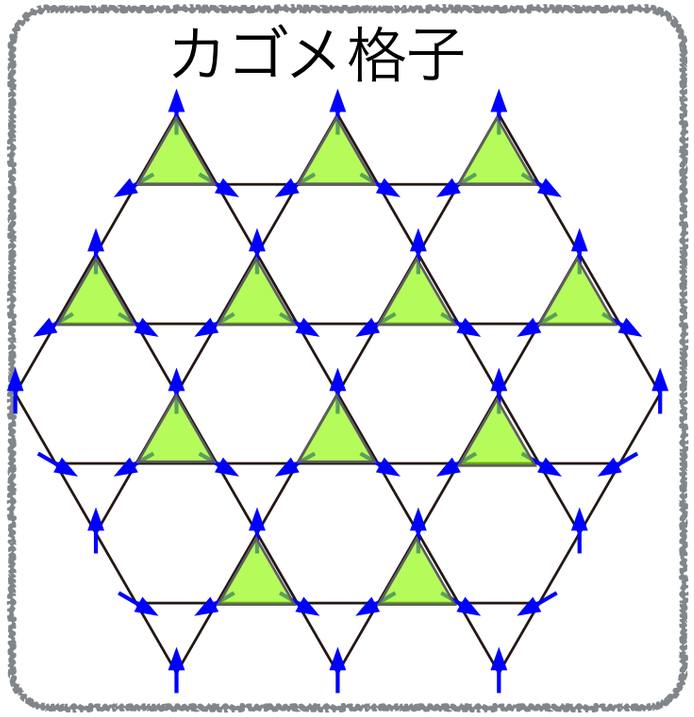
応用例3：カゴメ格子ハイゼンベルグ模型

T. Okubo and N. Kawashima, *in preparation*

ハミルトニアン

$$\mathcal{H} = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j - h \sum_i S_{i,z}$$





- ゼロ磁場での基底状態

古典模型：　各三角形で“120 度構造”を満たす任意の状態

→ 立体構造を含んで大規模に縮退

量子ゆらぎ：

S=1/2量子スピン系

スピン液体の候補

- Z_2 spin liquid
- U(1) Dirac spin liquid
- …

＊最近のiTPSによる計算：

H. J. Liao, *et al*, arXiv:1610.04727

S. Jiang, *et al*, arXiv:1610.02024

→ U(1) Dirac spin liquidを示唆

籠目

Taken from http://koharu2009.blogspot.jp/

S=1/2模型の磁化過程

- “グランドカノニカル”DM

  5つの磁化プラ

  M/M$_{Sat}$

  ⭐1/3 プラト

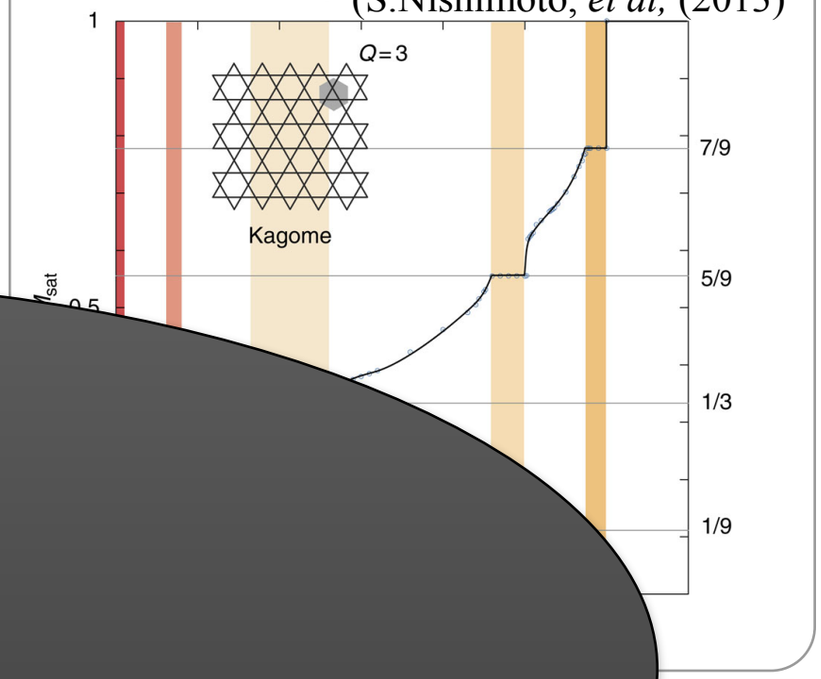
- 厳密対角化（$N \leq 42$） (H

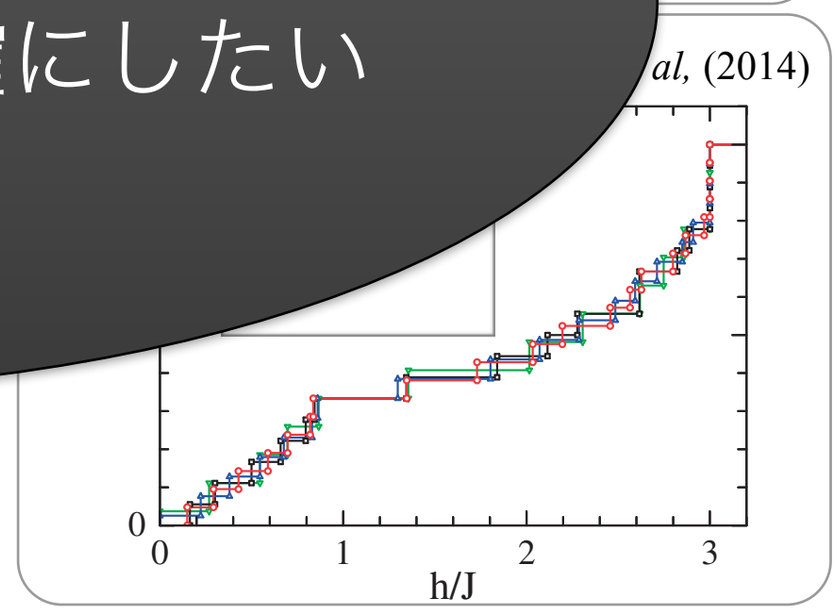
  ⭐M/M$_{Sat}$ =1/3は“

目的

1/3プラトー状態を明確にしたい

DMRG (S.Nishimoto, *et al,* (2013)





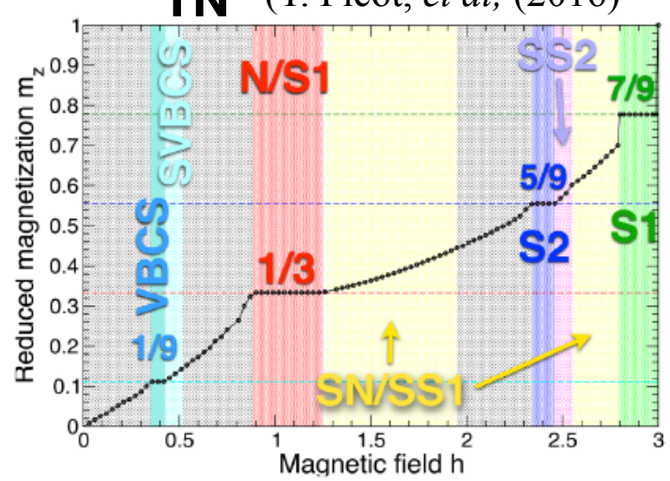
- テンソルネットワーク (T. Picot, *et al,* (2016), T. Okubo *et al,* (2015年秋物理学会))

  DMRGと同じ位置に４つの磁化プラトー

  cf. M/M$_{Sat}$ = 0にはギャップがないように見える

  ⭐M/M$_{Sat}$ =1/3は古典的な、up-up-down状態

TN (T. Picot, *et al,* (2016)

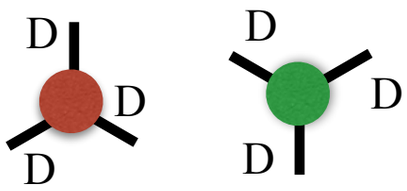


手法：PEPS テンソルネットワーク法

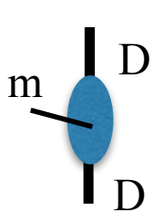
波動関数のExtended PEPS (PESS) 表現

D. Poilblanc et al, PRB 87, 140407(R) (2012)
Z. Y. Xie et al, PRX 4, 011025(2014))

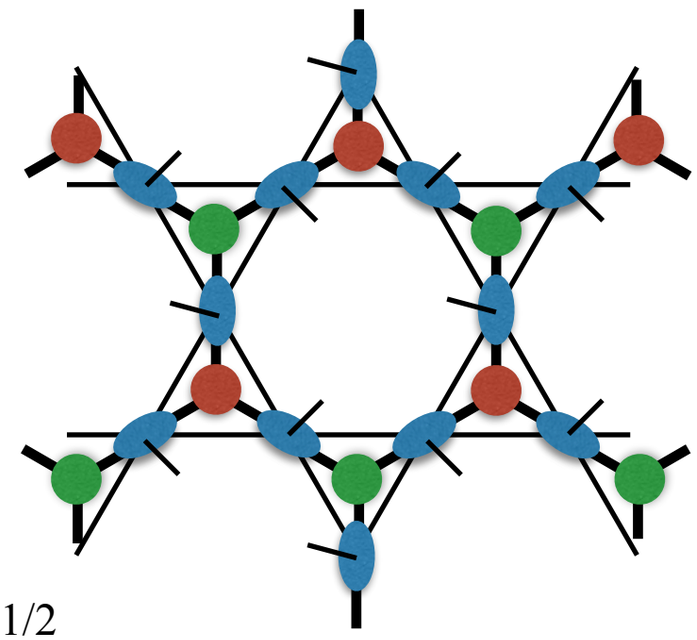
スピンに依存しないテンソル



スピンに依存するテンソル

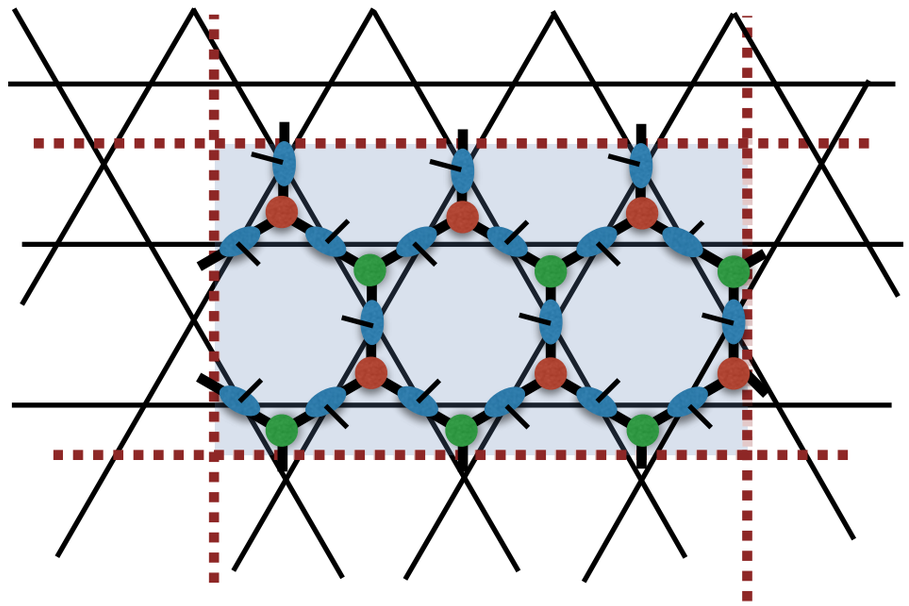


D:ボンド次元　　m: Sz = ±1/2



18-sites ユニットセルによる

(√3 × √3 構造に整合)



二段階の計算：

1．テンソルの最適化: 虚時間発展　$\lim_{\tau\to\infty} e^{-\tau\mathcal{H}}|\Psi_0\rangle = |\Psi_{\mathrm{G.S}}\rangle$　（＋ Suzuki-Trotter 分解）

打ち切り法: Simple update

2．物理量の計算:

近似的な縮約：Corner Transfer Matrix 法

結果：磁化過程

４つの明確な磁化プラトー

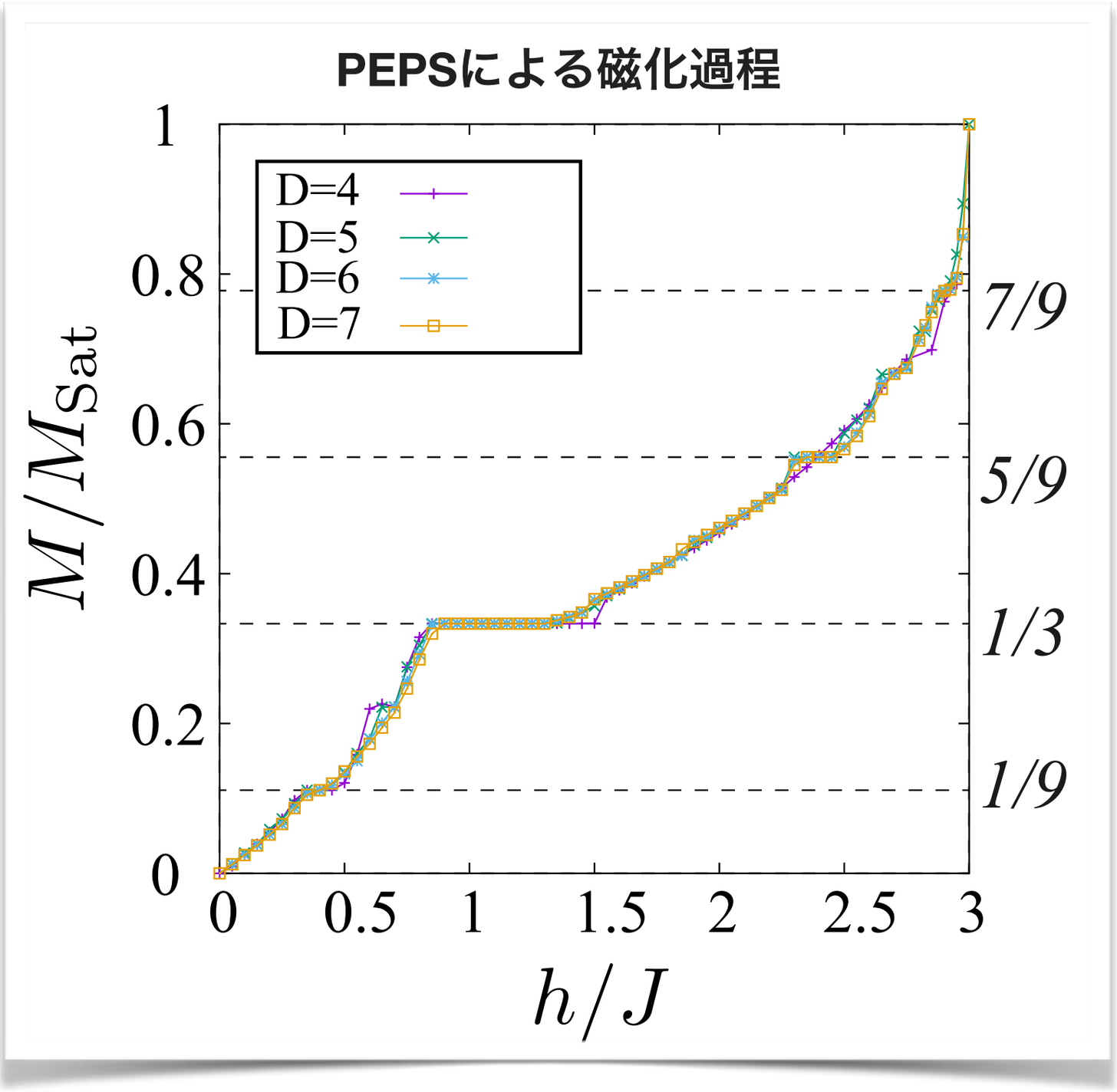
・D=7までで、ほぼ収束

M/M_{Sat}=1/9, 1/3, 5/9, 7/9

➡ DMRGとコンシステント

・ゼロ磁場近傍にギャップは見えない

➡ DMRGとは異なる



結果：1/3プラトー状

1/3プラトーは
DMRGと同様のResonated状態

テンソル最適化

従来：

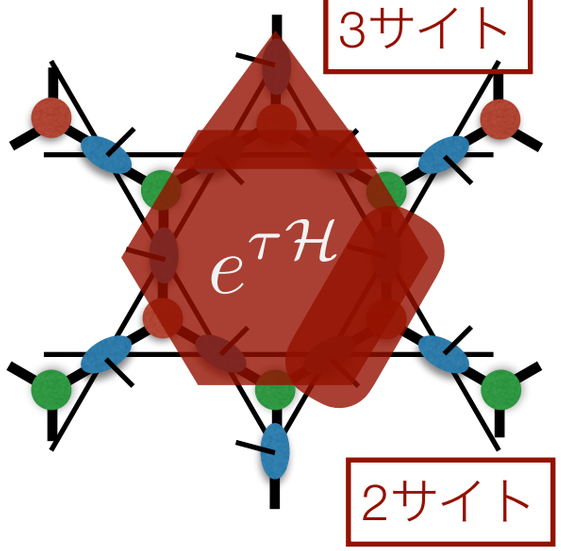
- 2サイト（T. Okubo et al)
- 3サイト（T. Picot et al)

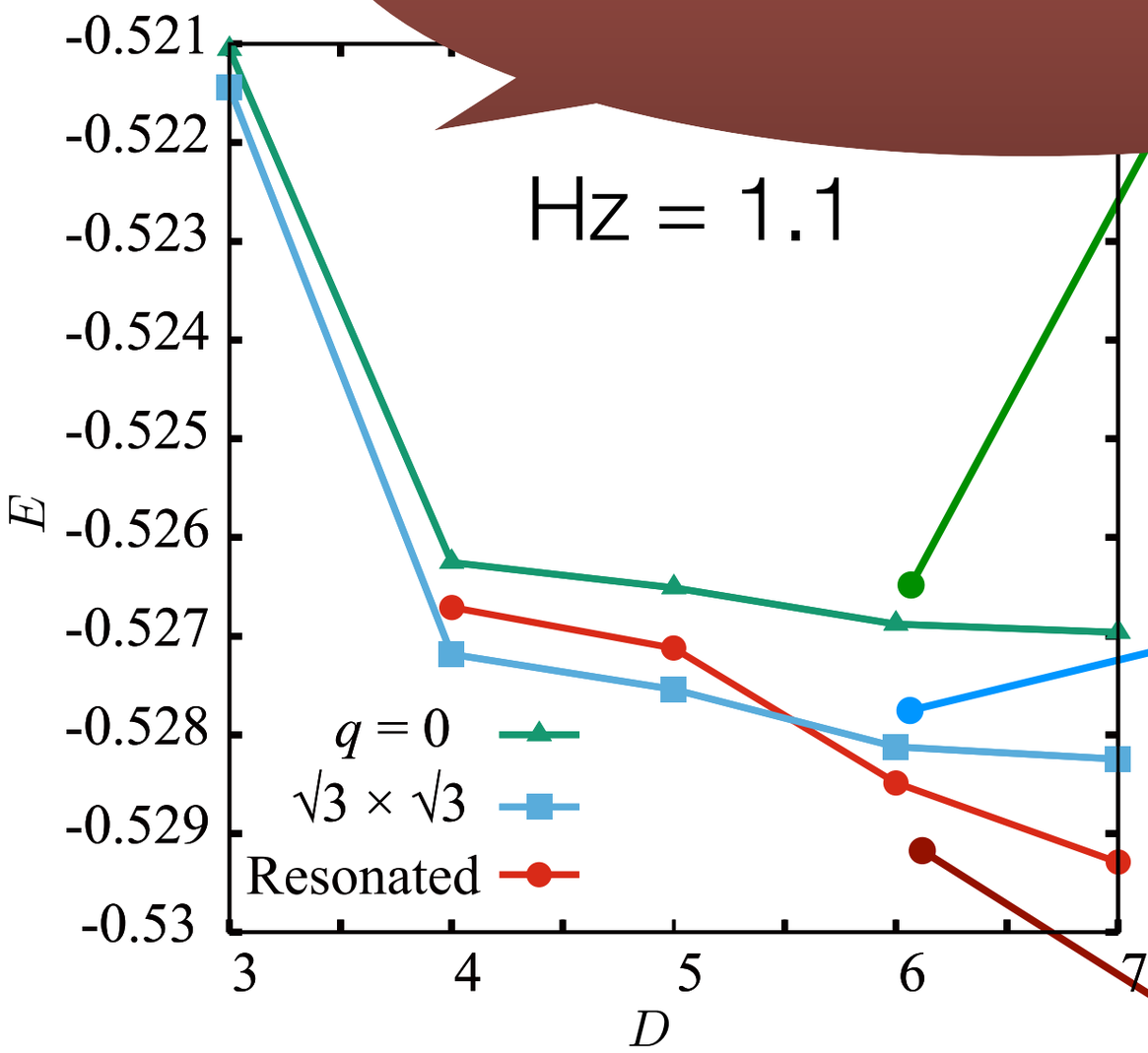
の虚時間発展演算子

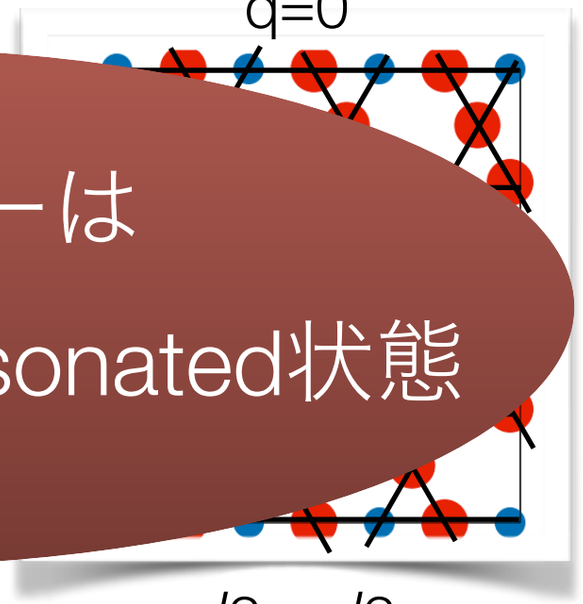
六角形ループの相関を
十分に取り込めない？

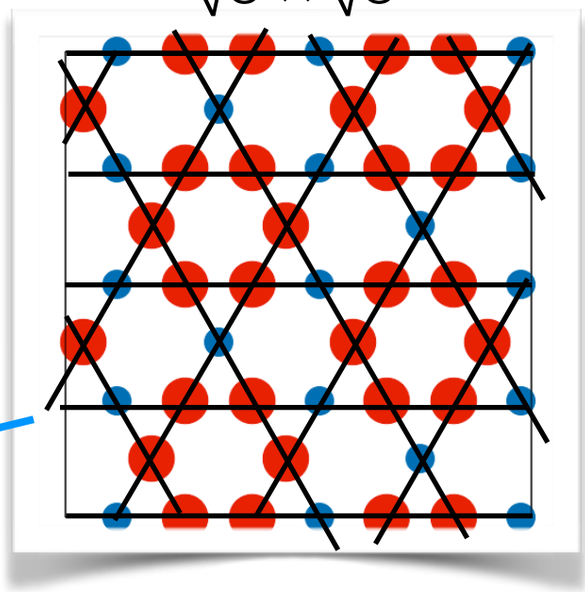
今回：

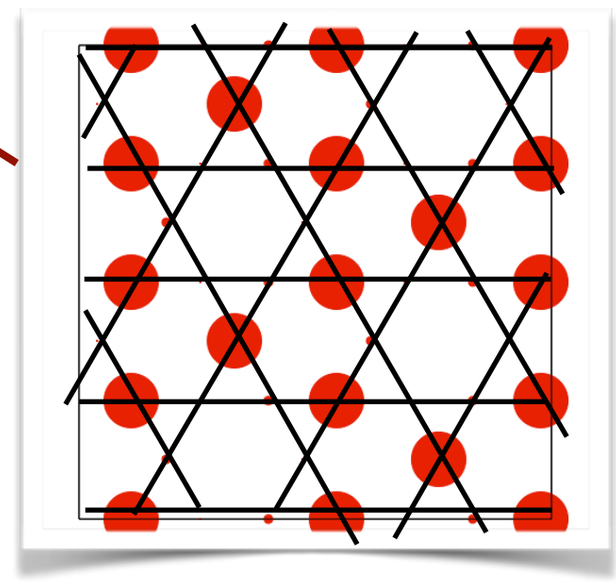
6サイトの虚時間発展











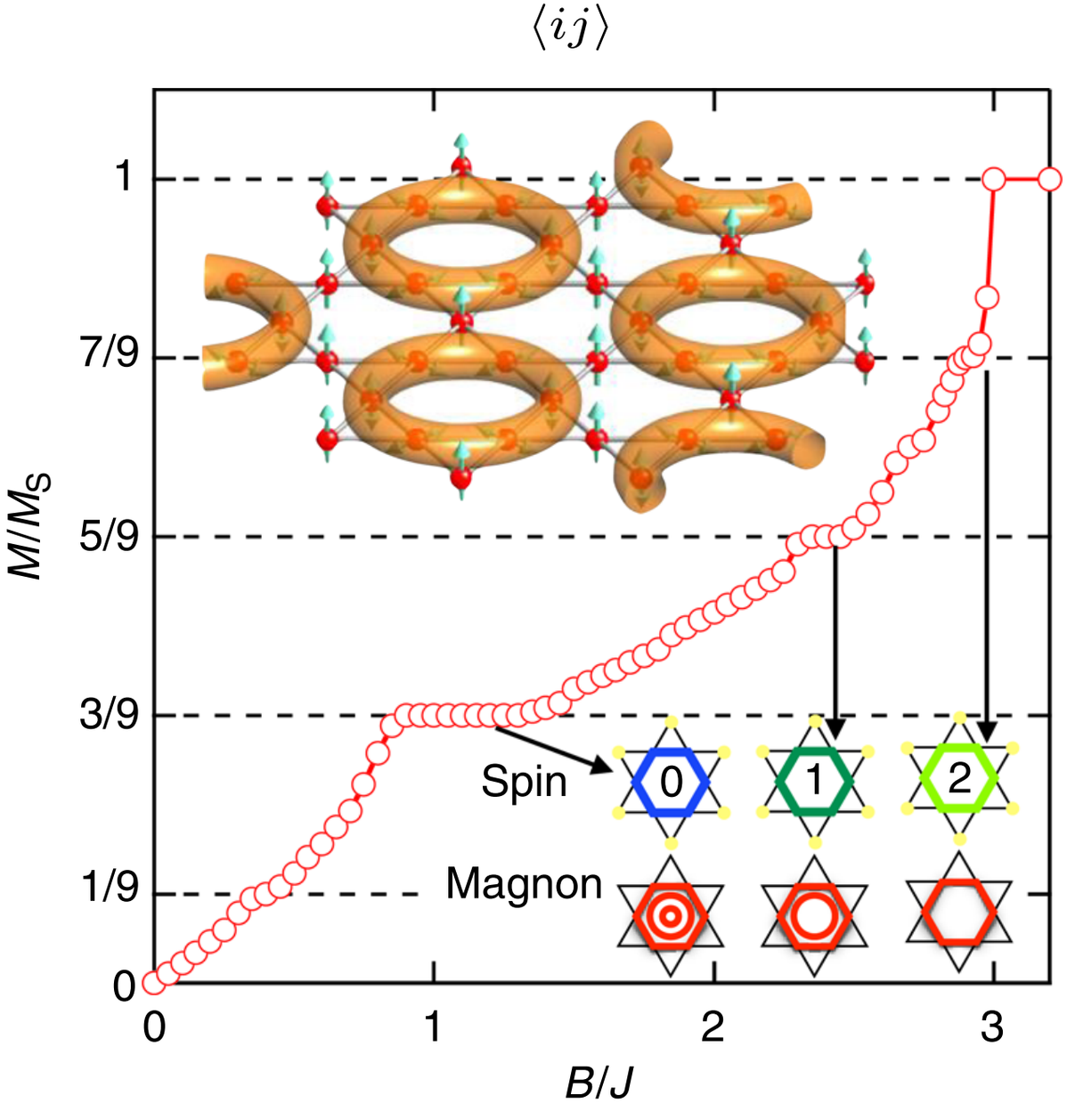
- 6サイトの最適化を行うことで、DMRGと同様の Resonated 状態が準安定状態として出現
- Resoneted 状態のエネルギーはDの増大で、up-up-down状態よりも低くなる

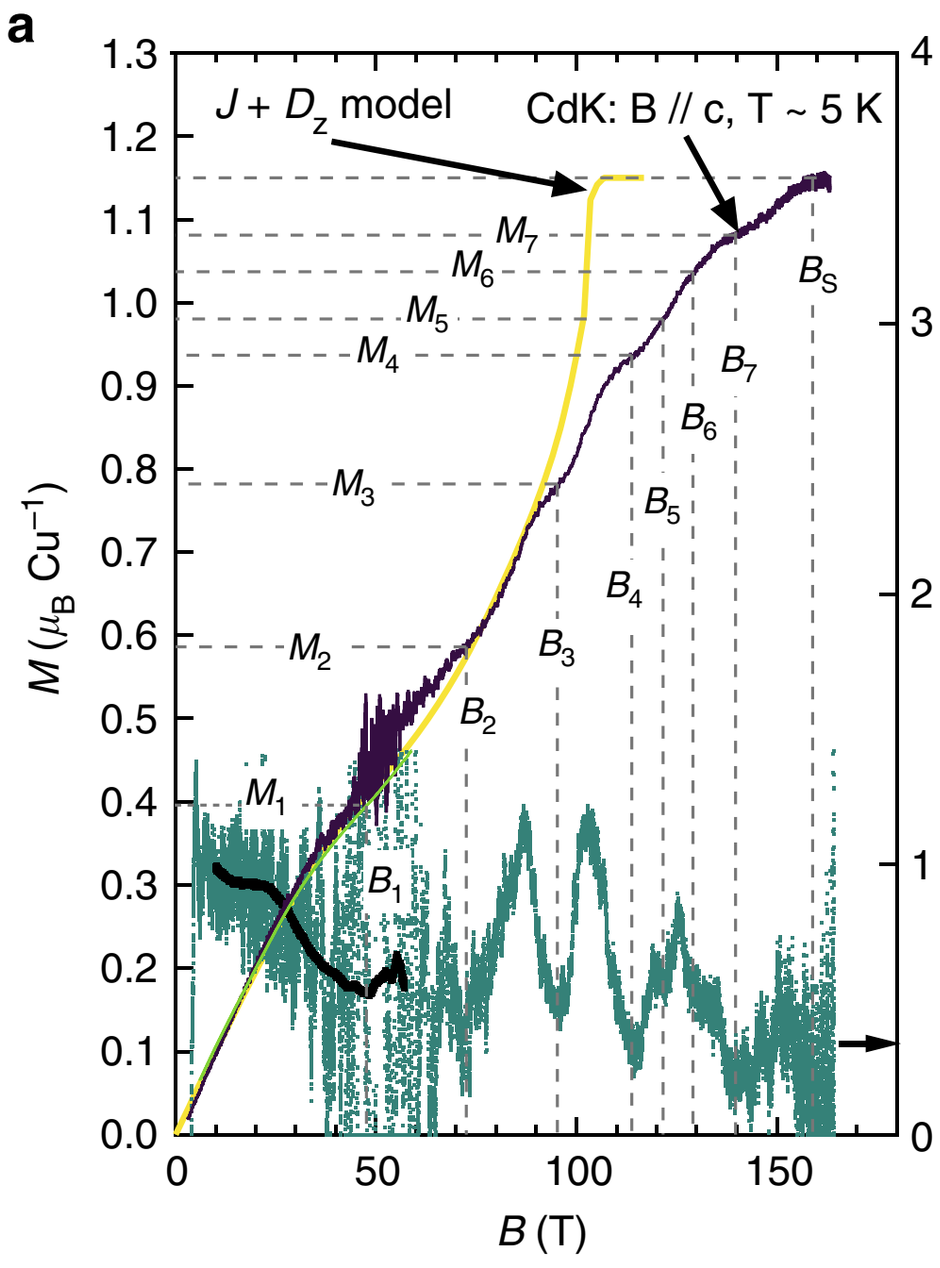
Comparison with experiments

R. Okuma, D. Nakamura, T.Okubo et al, Nat. Commun. **10**, 1229 (2019).

Real compounds often contain Dzyaloshinskii-Moriya Interaction:

$$\mathcal{H}_{ij} = \sum_{\langle ij \rangle} \vec{D_{ij}} \cdot (\vec{S}_i \times \vec{S}_j)$$





最後に

- テンソルネットワーク表現を用いると、効率的に情報を圧縮できる場合がある
  - 分配関数のテンソルネットワーク表示は、テンソルネットワーク繰り込みを用いた実空間繰り込み群により精度よく計算できる。
    - 基本となる情報圧縮手段は、特異値分解（SVD）による低ランク近似
  - 適切なテンソルネットワークにより（基底状態の）波動関数を非常に効率的に表現できる
    - 適切なネットワークを探すには、エンタングルメントエントロピーの面積則が大事
      - １次元量子系では、行列積状態（MPS）が非常に成功している
      - ２次元以上の量子系では、MPSは大きな系を効率的に近似できないが、テンソル積状態（TPS）ではうまくいく
- テンソルネットワーク表現のアイデアは、物理に限らず、様々な分野へ応用可能
  - MPS、MERAなどの（量子）機械学習への適用
  - 量子回路デザインへの応用

今回紹介しなかったけど重要な話題

- 基底状態に限らず、有限温度の計算手法の開発も進んできている
  - 密度行列のテンソルネットワーク表現

    P. Czarnik et al., Phys. Rev. B **99**, 035115 (2019).

    A. Kshetrimayum et al, Phys. Rev. Lett. **122** 070502 (2019).

- Fermion系にも適用可能で、Hubbard模型の計算では最先端に匹敵する精度を出しつつある

  (P. Corboz et al, Phys. Rev. B. **81**, 165104 (2010))

  (P. Corboz, Phys. Rev. B. **93**, 045116 (2016))
  - （注）金属状態ではエンタングルメントエントロピーの面積則が破れる

- 対称性を持つテンソルを用いた、波動関数トポロジーの特徴付け

  (S. Jiang and Y. Ran, Phys. Rev. B. **92**, 104414 (2015))

  (J.-W. Mei et al, Phys. Rev. B. **95**, 235107 (2017))

Report problem

以下の二点を含めて、A4で1枚くらい（多くても可）
のレポートを提出してください

提出先：大久保へメールで提出（11/21〆切）

t-okubo@phys.s.u-tokyo.ac.jp

（pdf、wordなど。手書きの場合は写真でも可）

- 講義の感想と講義を通して理解できたことの概要（まとめ）

- 自身の研究分野（もしくは興味のある分野）でテンソルネットワーク（または情報圧縮）が役に立ちそうかどうかの検討

モンテカルロ法

乱拓アルゴリズム

乱拓アルゴリズム：　（擬似）乱数を実行中に参照し
その値によって振る舞いを変える

例：モンテカルロ積分

円の面積　$\int_{x^2+y^2\leq 1} dxdy$

アルゴリズム

$N_a \leftarrow 0, N \leftarrow 0$　　初期化

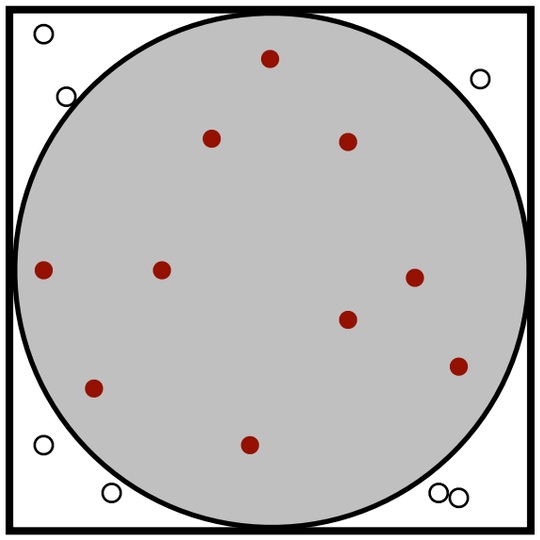
loop i

$x_i \in [-1,1]$
$y_i \in [-1,1]$　　一様乱数から
とってくる

$N \leftarrow N+1$

if $x_i^2+y_i^2 \leq 1$ then $N_a \leftarrow N_a+1$

end loop



$$\lim_{N\to\infty} \frac{N_a}{N} = \frac{\pi}{4}$$

統計誤差は $\frac{1}{\sqrt{N}}$ に比例

次元の呪い

このやり方（棄却法）は高次元積分に無力

n次元立方体（1辺の長さ2）に対する、d次元球の体積の割合

$$r = \frac{\pi^{d/2}/\Gamma(\frac{d}{2}+1)}{2^d} \sim \left(\frac{e\pi}{2d}\right)^{d/2}$$

Γ関数の漸近形

$$\Gamma(x) \sim \left(\frac{x}{e}\right)^x$$

dが大きくなると指数的に球の体積の割合が減る

→ 球に“ヒット”する確率が減り、誤差が指数的に増大してしまう

（誤差を一定に保つには、サンプリング数Nを指数的に大きくする必要がある）

Importance Sampling

- 積分に寄与する部分を重点的にサンプリングする

スピン模型の例：

ボルツマン重み　$e^{\beta J \sum_{\langle i,j \rangle} S_i S_j}$

が大きいところを重点的にサンプリングする

$$\sum_{\{S_i = \pm 1\}} \simeq$$

ランダムに発生した

部分空間での和

規格化定数（分配関数）が未知なので、

「マルコフ連鎖モンテカルロ法」で

重点的サンプリングを実現する。

マルコフ連鎖モンテカルロ法

ボルツマン分布を確率過程の定常分布として生成したい。

状態Γ を"ランダムに" Γ’ に変えていく

$$\Gamma = \{S_1, S_2, \ldots, S_N\} \Rightarrow \Gamma' = \{S'_1, S'_2, \ldots, S'_N\}$$

$W_{\Gamma\to\Gamma'}$ ：状態Γ が Γ’ に変わる確率

このとき、時刻tの確率分布 $\rho_t(\Gamma)$ は、マスター方程式

$$\rho_{t+1}(\Gamma) = \rho_t(\Gamma) + \sum_{\Gamma'} W_{\Gamma'\to\Gamma}\rho_t(\Gamma') - \sum_{\Gamma'} W_{\Gamma\to\Gamma'}\rho_t(\Gamma)$$

を満たす（確率の保存則）。

＊マルコフ＝右辺がひとつ前の時刻tの情報だけ

この確率過程が時間無限大で、ある確率分布 $P(\Gamma)$ に収束

$$\lim_{t\to\infty} \rho_t(\Gamma) = P(\Gamma)$$

$P(\Gamma)$ ：今の場合、ボルツマン分布

⇒ この確率過程で現れる点の集合はボルツマン分布になっている

マルコフ連鎖モンテカルロ法の収束条件

$P(\Gamma)$に収束するための十分条件

1. “**エルゴード性**”
   - 任意の二つの状態 Γ とΓ’ が有限の回数でつながる。
     - W を行列としてみると、この条件は

$$^{\exists}T > 0, ^{\forall}(\Gamma, \Gamma'), \quad [(W)^t]_{\Gamma,\Gamma'} > 0, (^{\forall}t \geq T)$$

2. “**釣り合い条件**”
   - 確率の流れが、$P(\Gamma)$に対して釣り合う

$$^{\forall}(\Gamma, \Gamma'), \quad \sum_{\Gamma'} W_{\Gamma\to\Gamma'} P(\Gamma) = \sum_{\Gamma'} W_{\Gamma'\to\Gamma} P(\Gamma')$$

特別な場合: **詳細釣り合い条件**

$$W_{\Gamma\to\Gamma'} P(\Gamma) = W_{\Gamma'\to\Gamma} P(\Gamma')$$

イジング模型への適用：メトロポリス法

Local update:

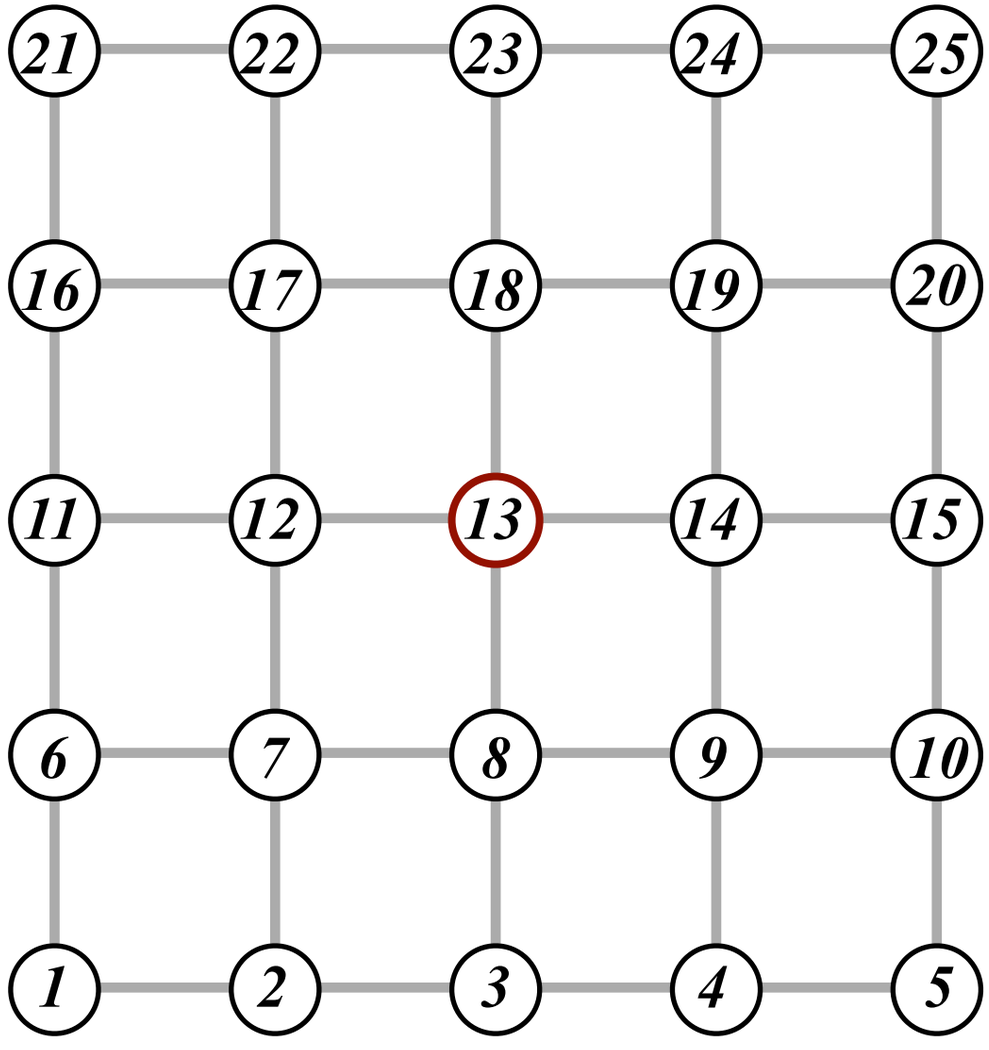
スピンの一部（典型的には１つのスピン）
の状態を変えるマルコフ連鎖モンテカルロ法

$$\Gamma = (S_1, S_2, \cdots, S_{13}, \cdots, S_{25})$$

$$\Gamma' = (S_1, S_2, \cdots, S'_{13}, \cdots, S_{25})$$

Γ から Γ’への変更で $S_1, S_2, .. S_{12}, S_{14}, S_{15}, \ldots, S_{25}$,
を固定して、 S_{13}だけを変更する。

このような local update では,
条件を満たす遷移確率 Wを
比較的簡単に見つけられる。

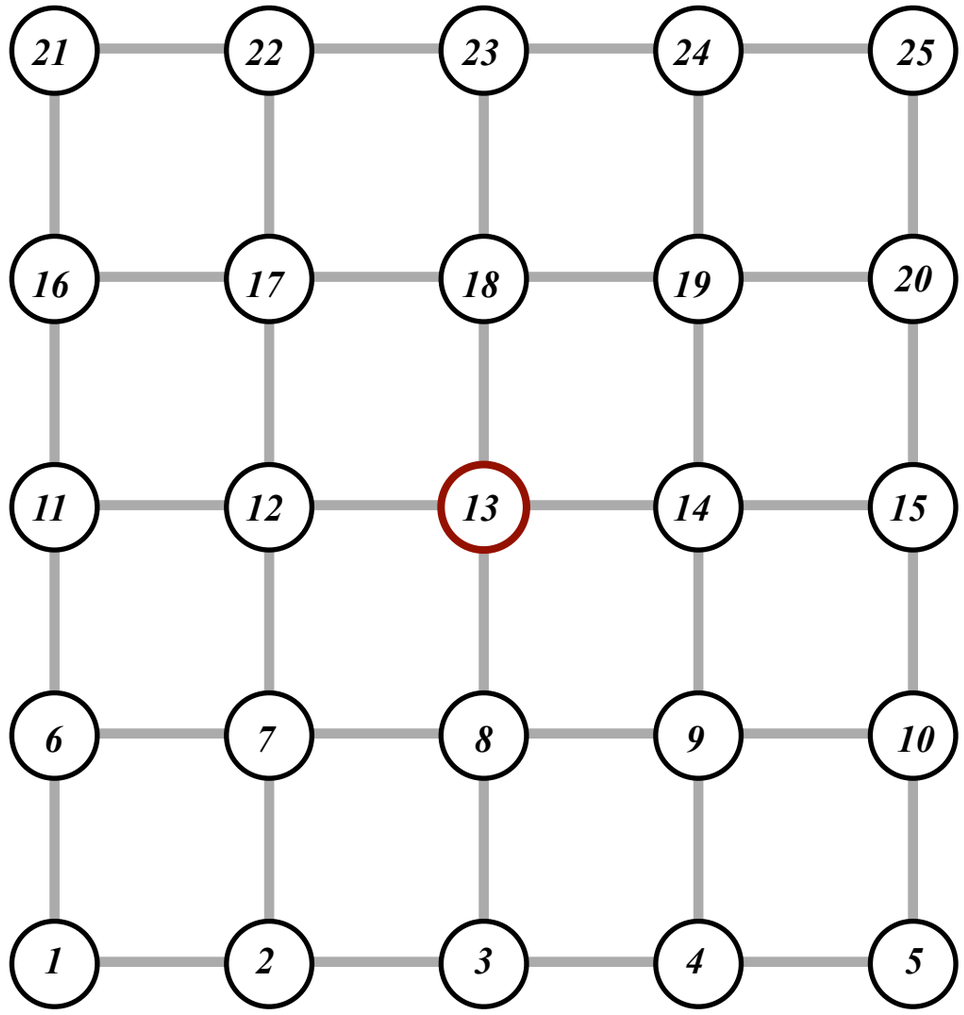


メトロポリス法：イジングスピン

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

- ひとつスピンを選んで（13番）スピンを反転する

$$S_{13} \to S'_{13} = -S_{13}$$



この時、詳細釣り合い条件

$$W_{\Gamma \to \Gamma'} P(\Gamma) = W_{\Gamma' \to \Gamma} P(\Gamma')$$

を満たすWは

$$\frac{W_{\Gamma \to \Gamma'}}{W_{\Gamma' \to \Gamma}} = \frac{P(\Gamma')}{P(\Gamma)} = e^{-\beta[\mathcal{H}(\Gamma') - \mathcal{H}(\Gamma)]} = e^{-\beta \Delta E}$$

を満たす。したがって、例えば

$$W_{\Gamma \to \Gamma'} = \min\left(1, e^{-\beta \Delta E}\right)$$

$$\begin{aligned} \Delta E &= -J(S_8 + S_{12} + S_{14} + S_{18})(S'_{13} - S_{13}) \\ &= 2J(S_8 + S_{12} + S_{14} + S_{18}) S_{13} \end{aligned}$$

とすれば良い。

メトロポリス法の流れ

Step 0: 初期状態を準備する $\Gamma_0 = (S_1, S_2, \cdots, S_N)$

loop t

i 番目のスピンを選ぶ

1. S_iから候補、Γ'を作る
   - イジング模型：$S_i' = -S_i$
2. エネルギー差を計算 $\Delta E = \mathcal{H}(\Gamma') - \mathcal{H}(\Gamma)$
3. 乱数を生成 $r \in [0, 1]$
4. r の値に応じて、次の状態 Γ_{t+1}を

$$\Gamma_{t+1} = \begin{cases} \Gamma' & r \leq e^{-\beta \Delta E} \\ \Gamma_t & \text{otherwise} \end{cases}$$

   で決める
5. 物理量 $O(\Gamma_t)$を計算する

典型的には
- random state ($T \to \infty$)
- ordered state ($T \to 0$)

エネルギーが減る場合ΔE < 0),
新しい状態は確率1で採択.

通常、物理量はNスピンの試行が終わる毎に計算することが多い。
"1 モンテカルロステップ"
= N スピンの反転の試行

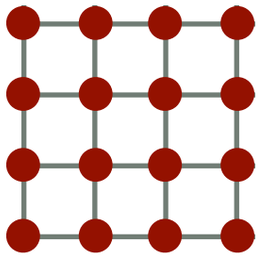
$$\langle O \rangle = \sum_{\Gamma} O(\Gamma) e^{-\beta \mathcal{H}(\Gamma)} \simeq \frac{1}{T} \sum_{t=1}^{T} O(\Gamma_t)$$

角転送行列繰り込み群（CTMRG）

- 奥西・西野ら (1995)による逐次的な“繰り込み”によるテンソルネットワークの計算方法
  - Corner Transfer Matrix Renormalization Group（CTMRG）
- 分配関数のテンソルネットワーク表現をL→L+2のように数サイトずつ大きくしていくことで、徐々に計算する
- 近年、２次元量子多体系の基底状態計算アルゴリズム（PEPS法、TPS法）の一部にも使われる

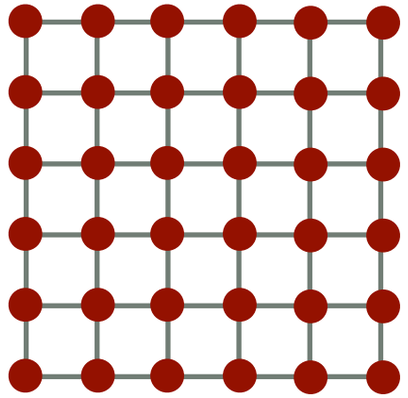
CTMRGでやりたいこと

L×Lの分配関数が
（近似的に）計算できた

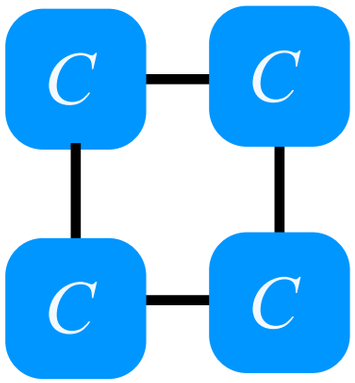


系を少し大きくする

(L+2)×(L+2)の分配関数を計算

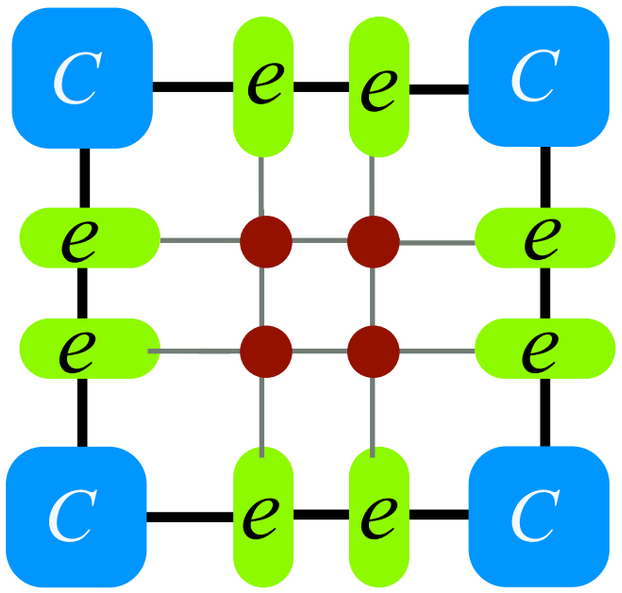


角転送行列表現

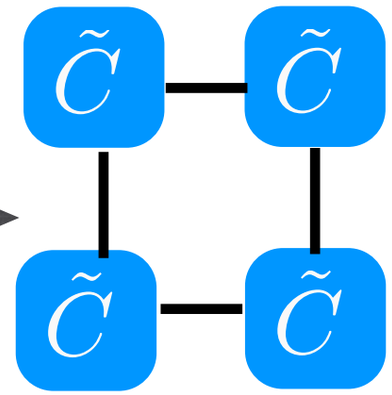


$C: D \times D$

系を少し大きくする



（近似）

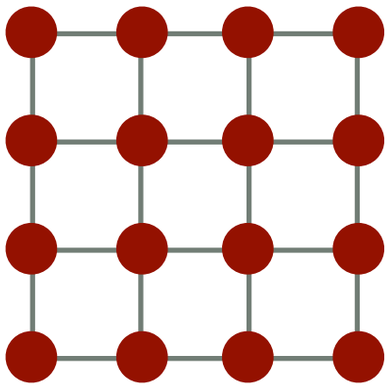


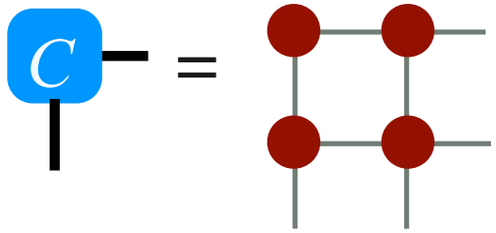
$\tilde{C}: D \times D$

Cの大きさを変えずに
系を大きくする

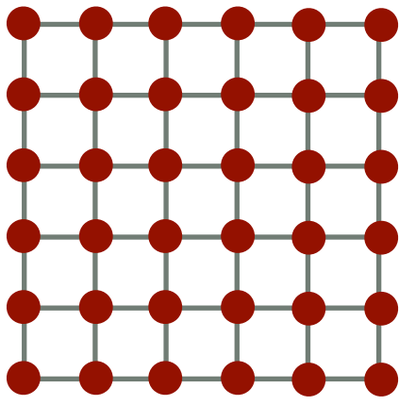
角転送行列の意味

4×4

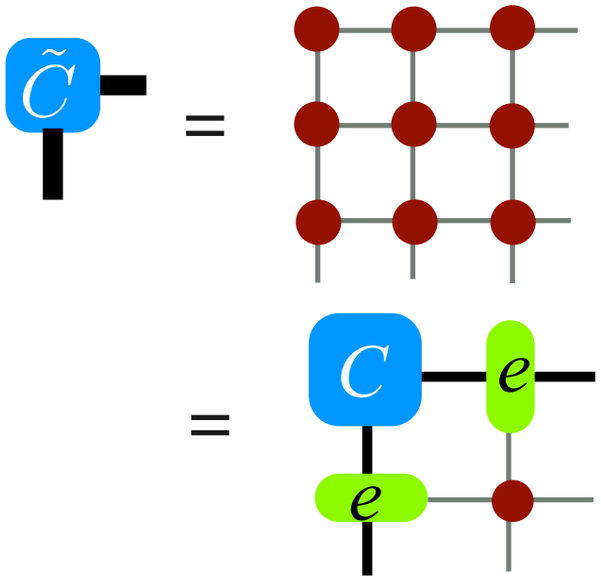




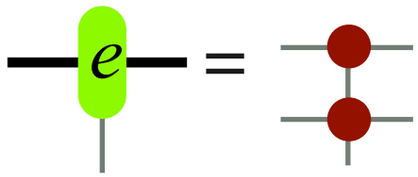
6×6

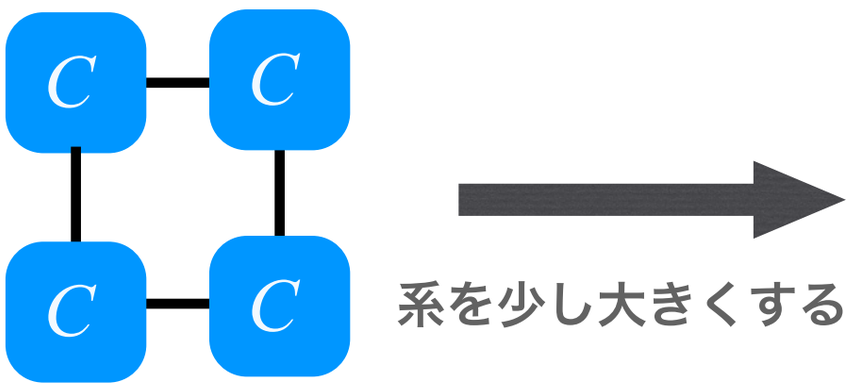


（近似なし）

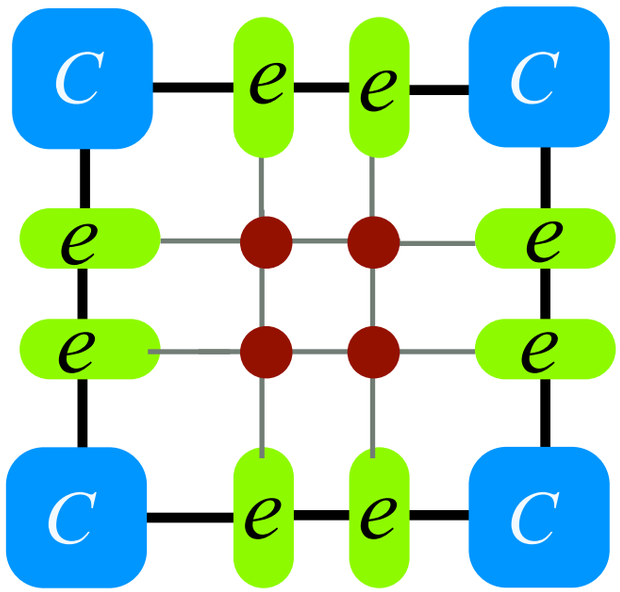


角転送行列表現

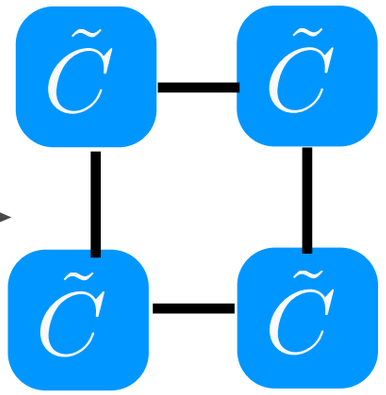




系を少し大きくする



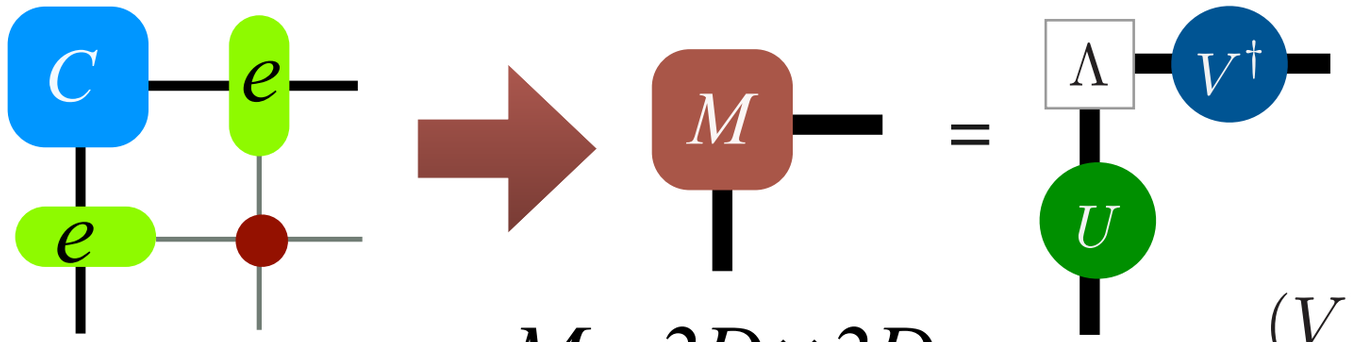
（近似）



CTMRGのレシピ

1. SVDによる分解

行列と思ってSVD

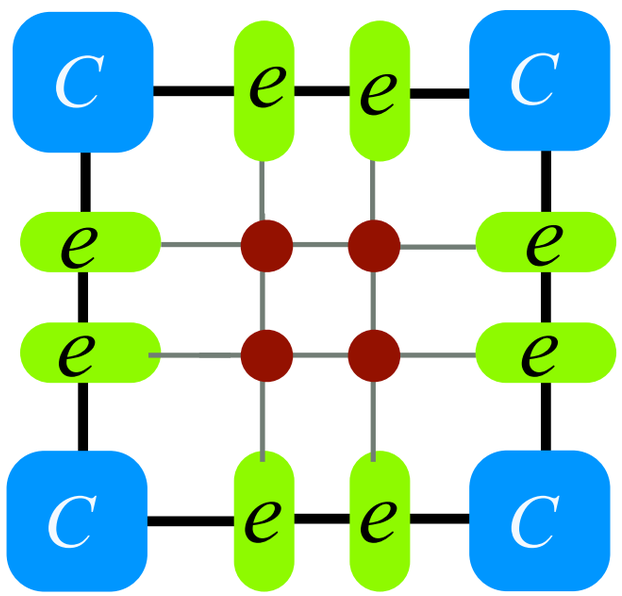


$M : 2D \times 2D$

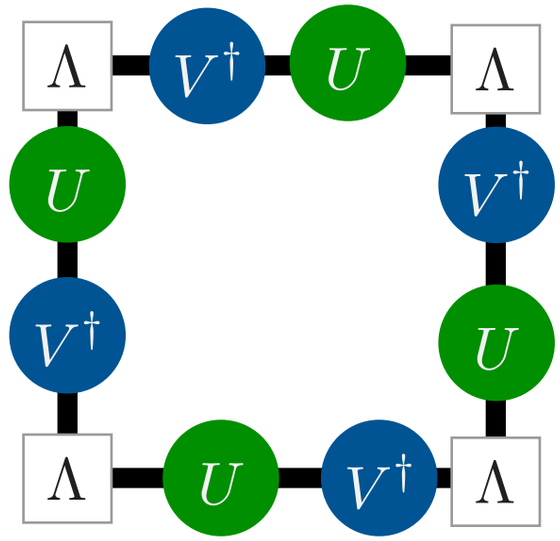
Mは実対称（エルミート）行列

$$(V^\dagger U)_{i,j} = (U^\dagger V)_{i,j} = (-1)^{\eta_i}\delta_{i,j}$$

$$\eta_i = 0, 1$$



=



＊対称性を仮定

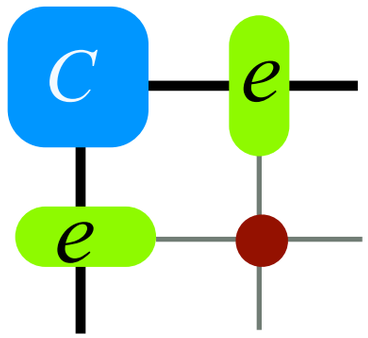
$$= \sum_i \lambda_i^4 (-1)^{4\eta_i}$$

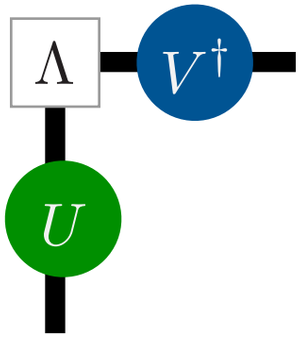
$$= \sum_i \lambda_i^4$$

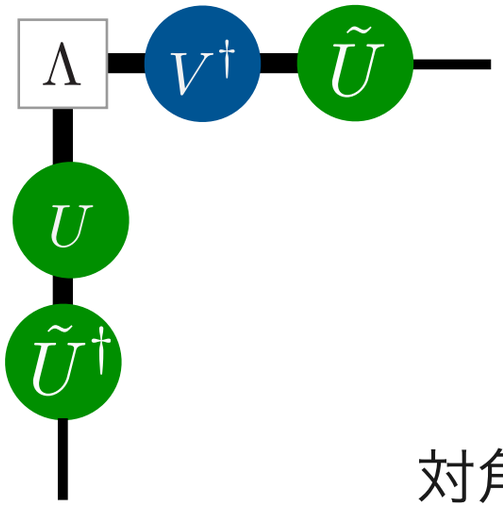
特異値が大きいものD個を残せば良い近似！

CTMRGのレシピ

2. SVDを使って近似

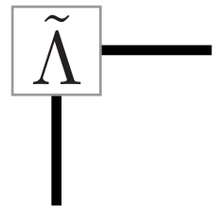






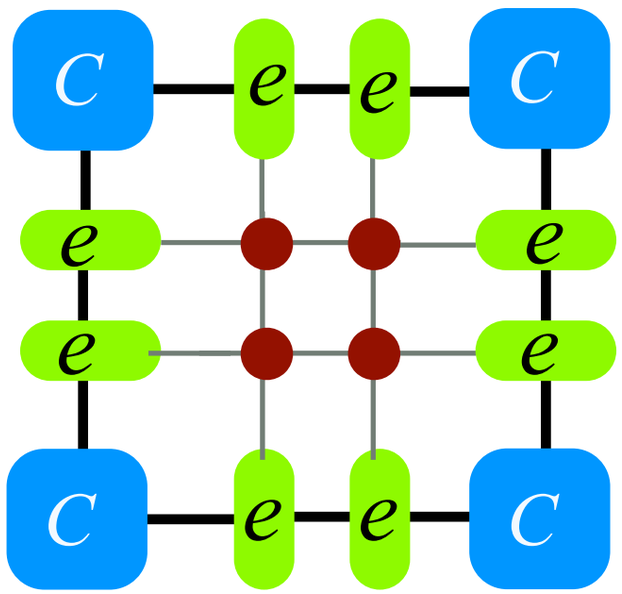
$\tilde{U} : 2D \times D$

$=$

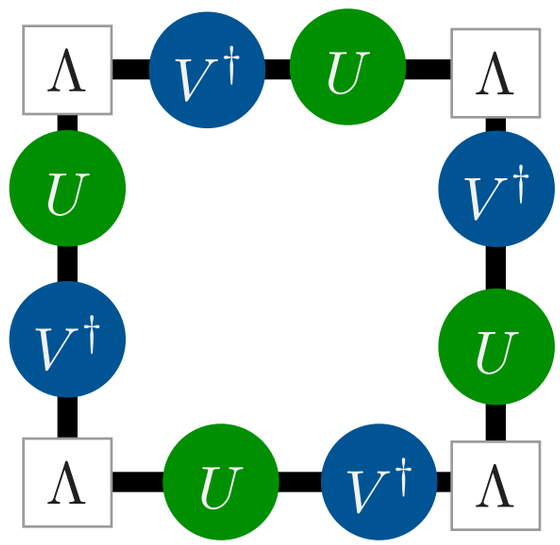


$\tilde{\Lambda} : D \times D$

対角成分が $(-1)^{\eta_i}\lambda_i$
の対角行列

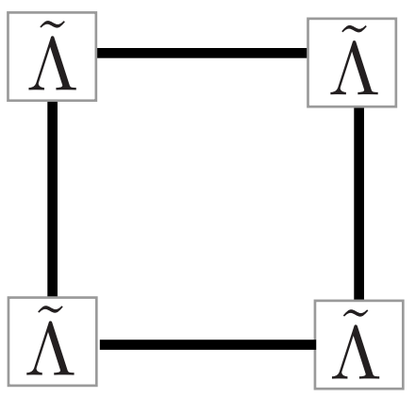


$=$



$$Z = \sum_i^{2D} \lambda_i^4$$

$\simeq$ 近似

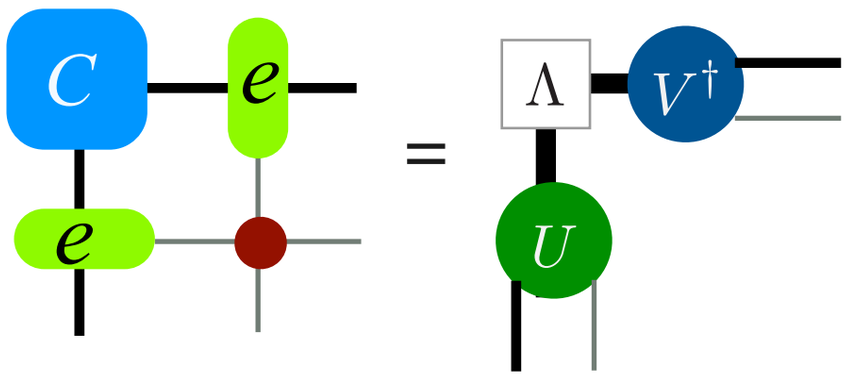


$$Z \simeq \sum_i^{D} \lambda_i^4$$

CTMRGのレシピ

くりこみ変換まとめ

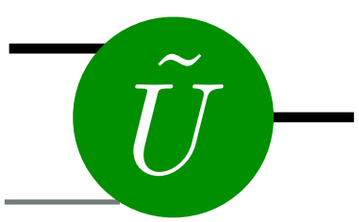
1. LxL の系の角転送行列をSVD



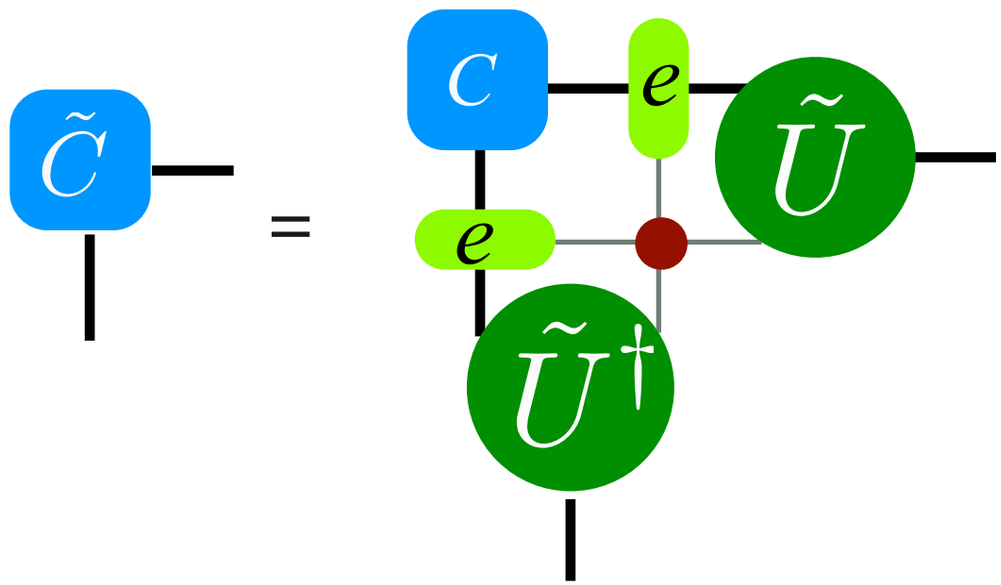
2. Projectorを作る

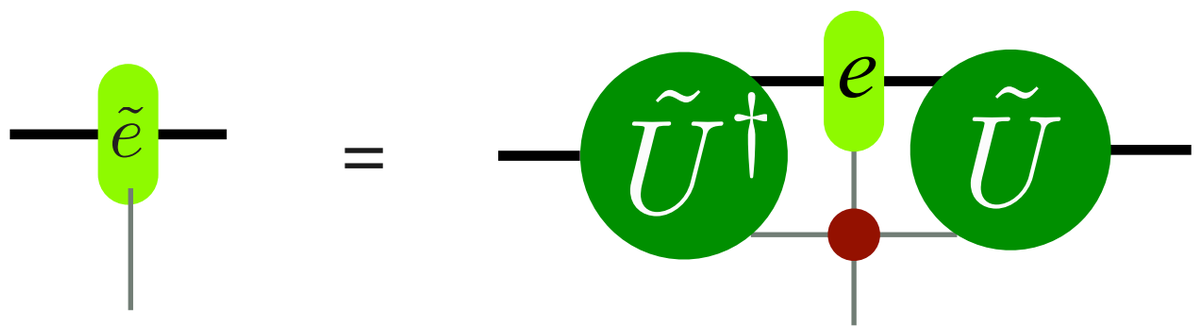
特異値が大きい方から

D個だけ残す



3. (L+2)×(L+2)の角転送行列を作成





大きさLの系の分配関数が逐次求まる