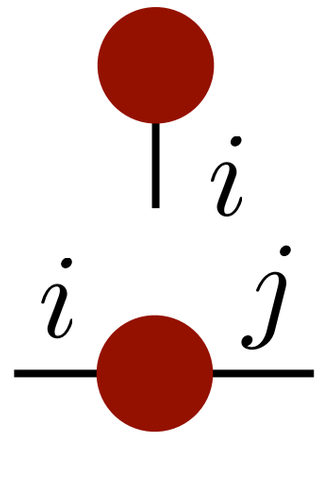
Quantum simulation by tensor network methods
（テンソルネットワークによる量子シミュレーション）

Univ. Tokyo, Tsuyoshi Okubo

Contents

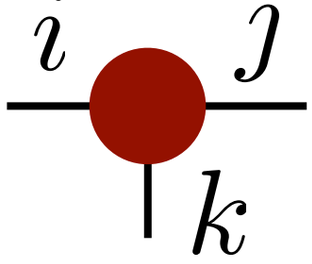
Graphical representations for tensor network

- Vector $\vec{v} : v_i$



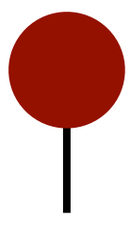
- Matrix $M : M_{i,j}$

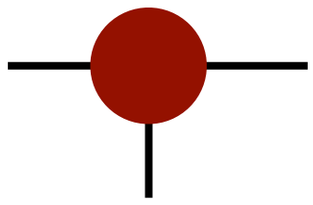
- Tensor $T : T_{i,j,k}$



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

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

$\vec{v} =$ 

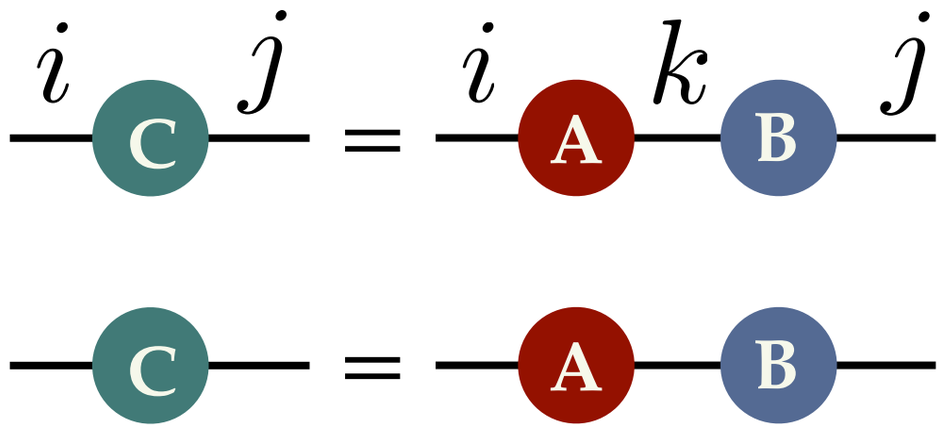
$T =$ 

Graphical representations for tensor network

Matrix product

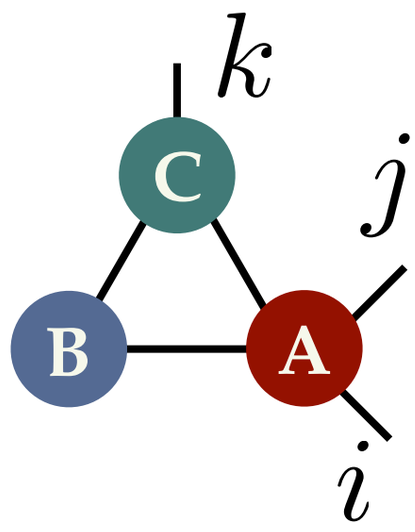
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

$$C = AB$$



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

Tensor network for quantum many-body states

Quantum many-body state $\mathcal{H}|\Psi\rangle = E|\Psi\rangle$

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

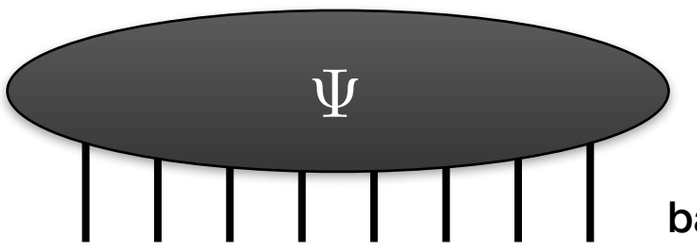
Basis

Spin, qubits $i = \uparrow, \downarrow = |0\rangle, |1\rangle$

$|010100\cdots 0\rangle = |0\rangle \otimes |1\rangle \otimes \cdots$

The coefficient can be seen as a tensor

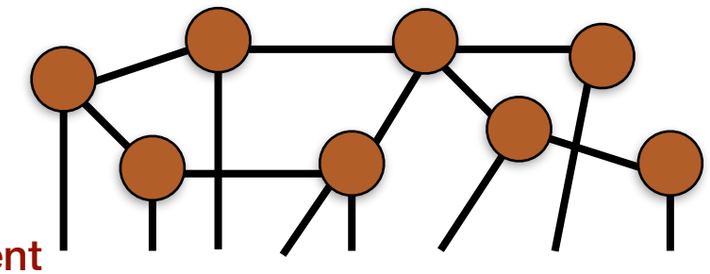
Quantum many-body state



$\sim e^N$ independent elements

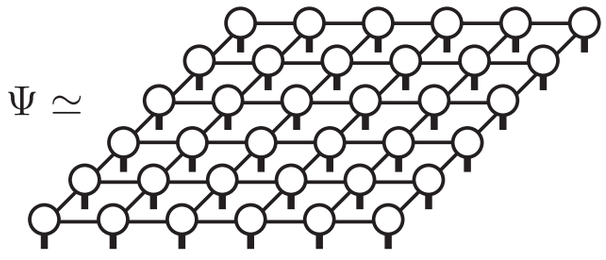
Approximation based on entanglement

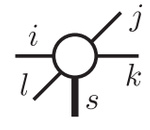
Tensor network decomposition



$\sim O(N)$ independent elements

PEPS, TPS (for 2d system)



$T_{ijkl}[s] =$ 

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

ex. TPS: # of elements ~ ND^4 D: dimension of the tensor T

Exponential→ Linear

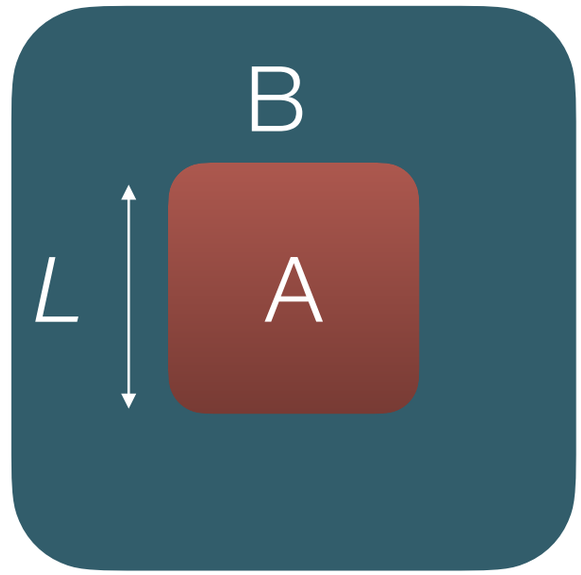
Area law of the entanglement entropy in physics

General wave functions (vector):

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)

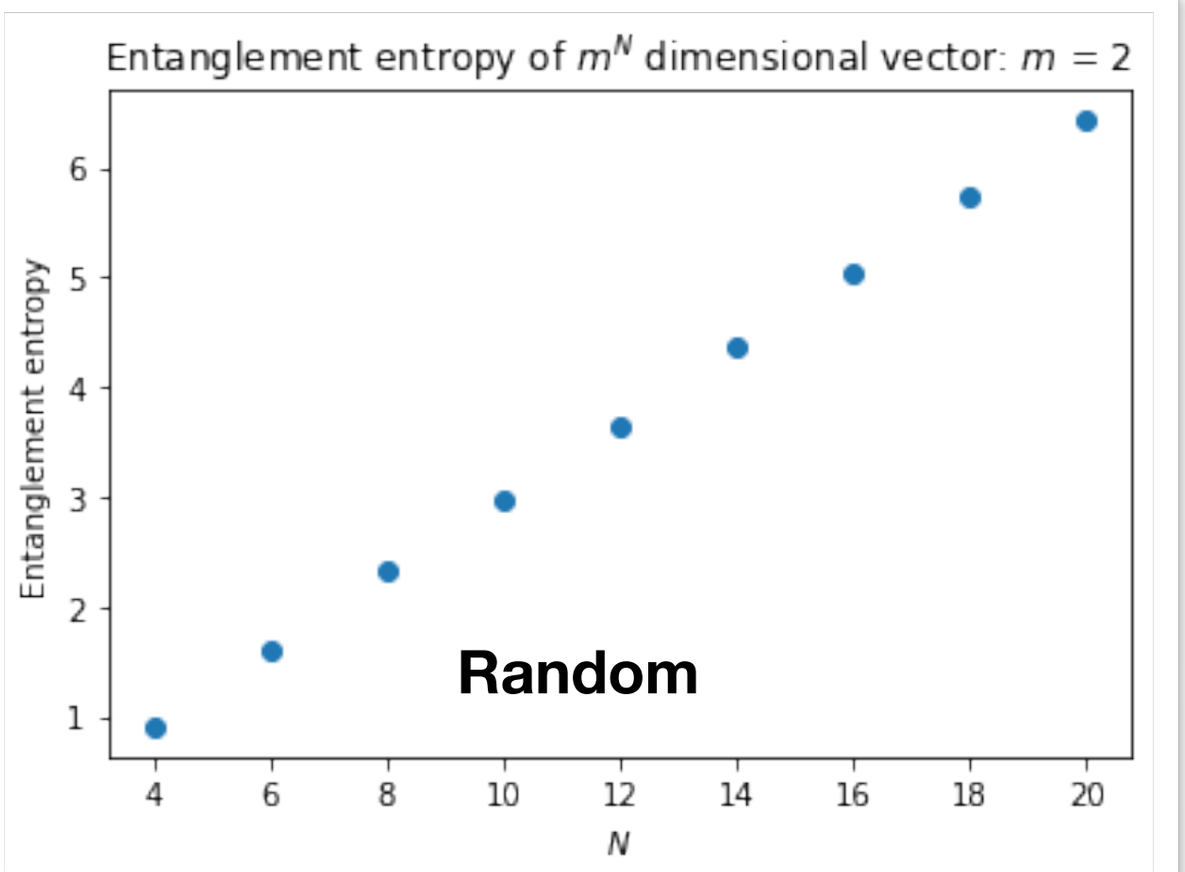
$$S = O(1)$$

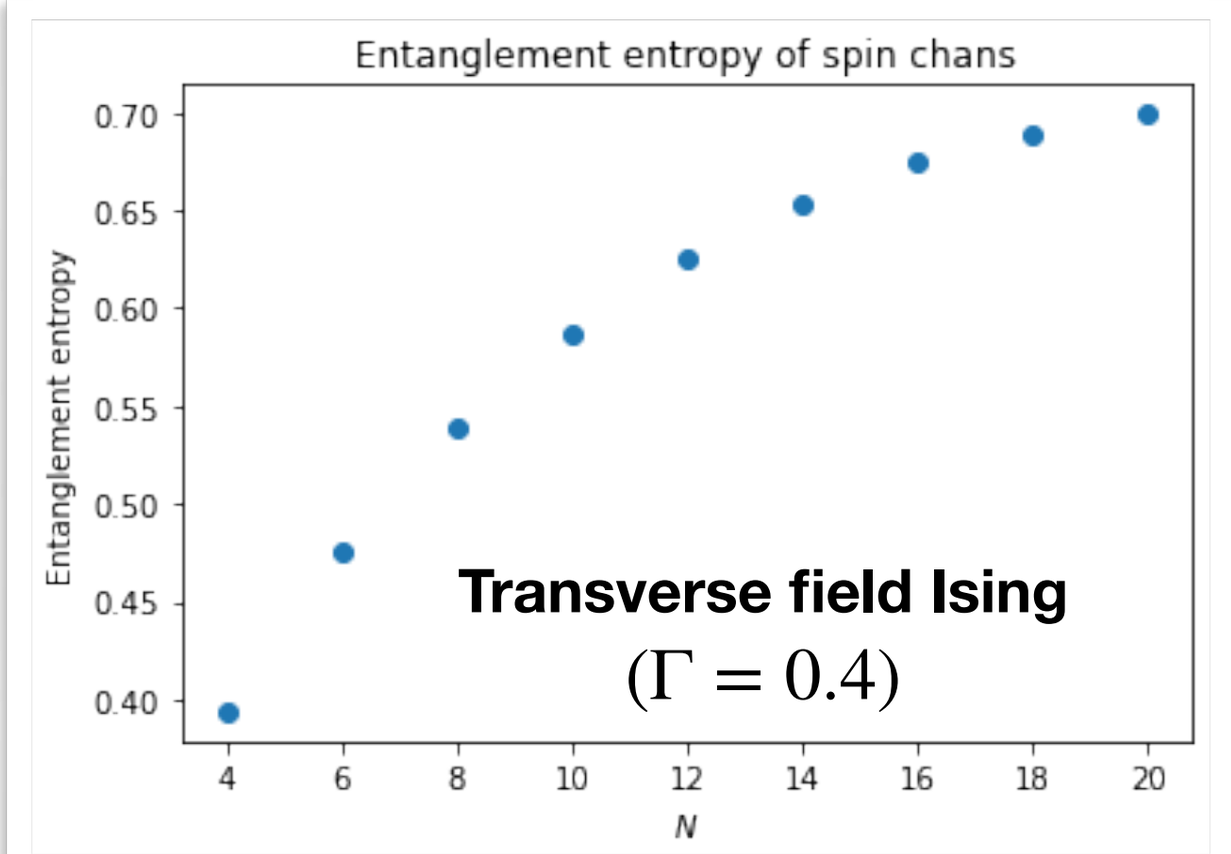
Ground state are in a small part of the huge Hilbert space

Scaling of the entanglement entropy

$$\vec{v} \in \mathbb{C}^{2^N}$$

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



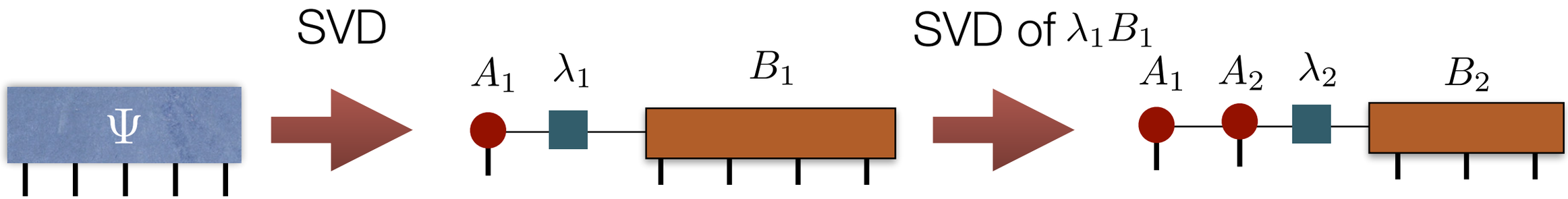


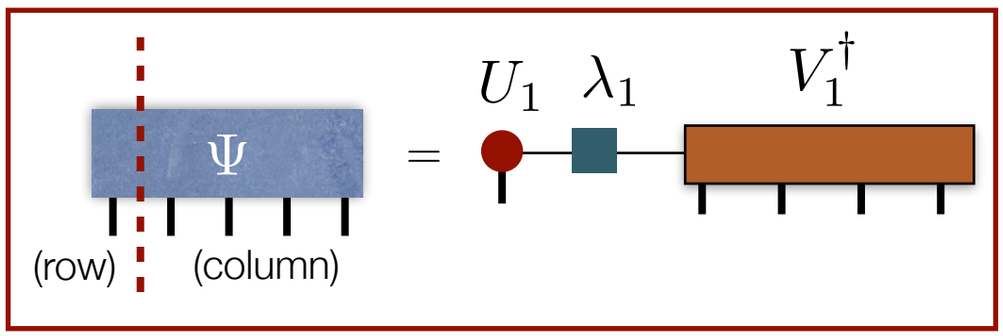
Random vector: Volume low
Ground state: Area low

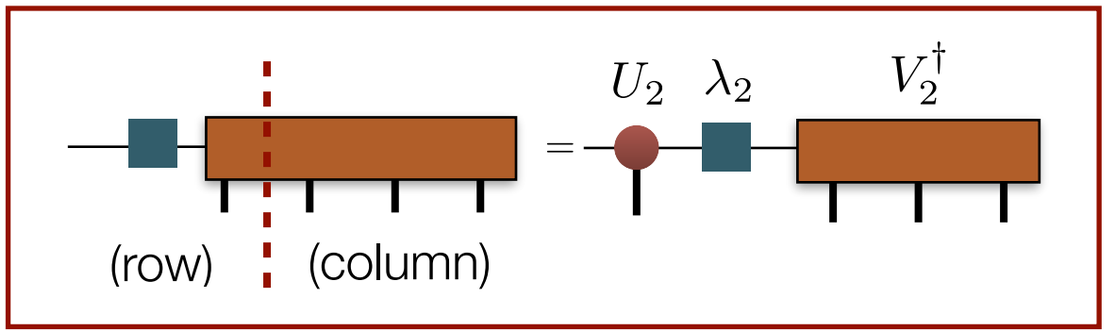
Contents

Matrix product state without approximation

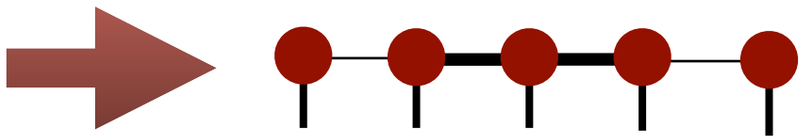
General vectors can be represented by MPS exactly
through successive Schmidt decompositions







…



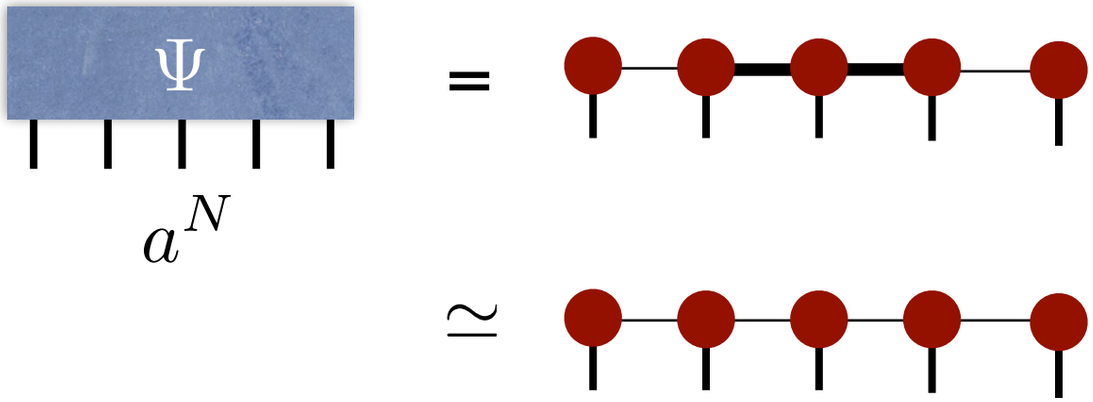
In this construction, the sizes of matrices
depend on the position.

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

(ボンド次元)

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

Matrix product state: Low rank approximation



a^N

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

a^N

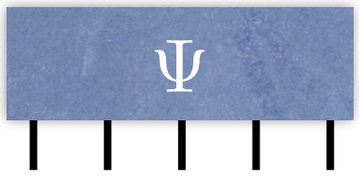
Maximum bond dimension = χ

$\propto Na\chi^2$

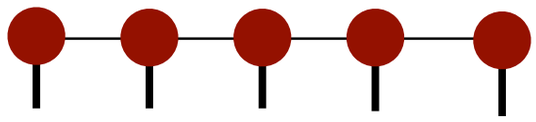
If an MPS can accurately approximate the original tensor with a small "χ", it is a huge data compression from exponential to a polynomial of *N*!

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

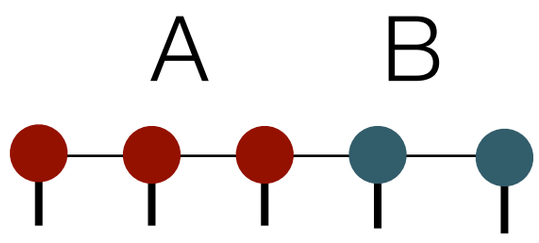


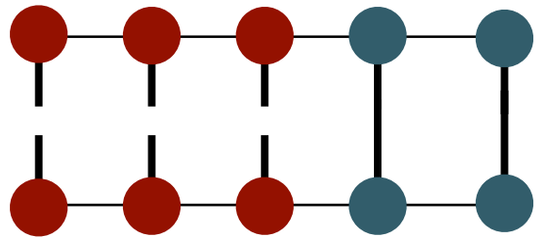
$\simeq$



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

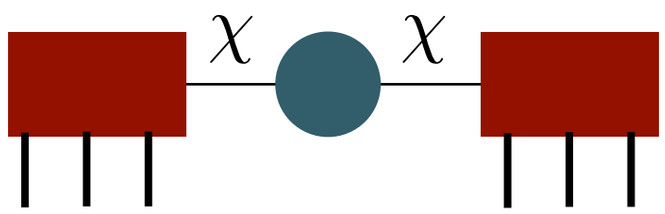
Reduced density matrix of region A:



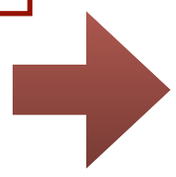


$$\rho_A = \mathrm{Tr}_B |\tilde{\Psi}\rangle\langle\tilde{\Psi}| =$$

★ Structure of ρ_A:



Here we used the properties of SVD 5: “multiplication” explained in #3.

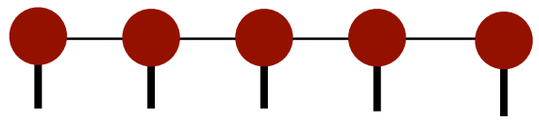


$$\mathrm{rank}\ \rho_A \leq \chi$$

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

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
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

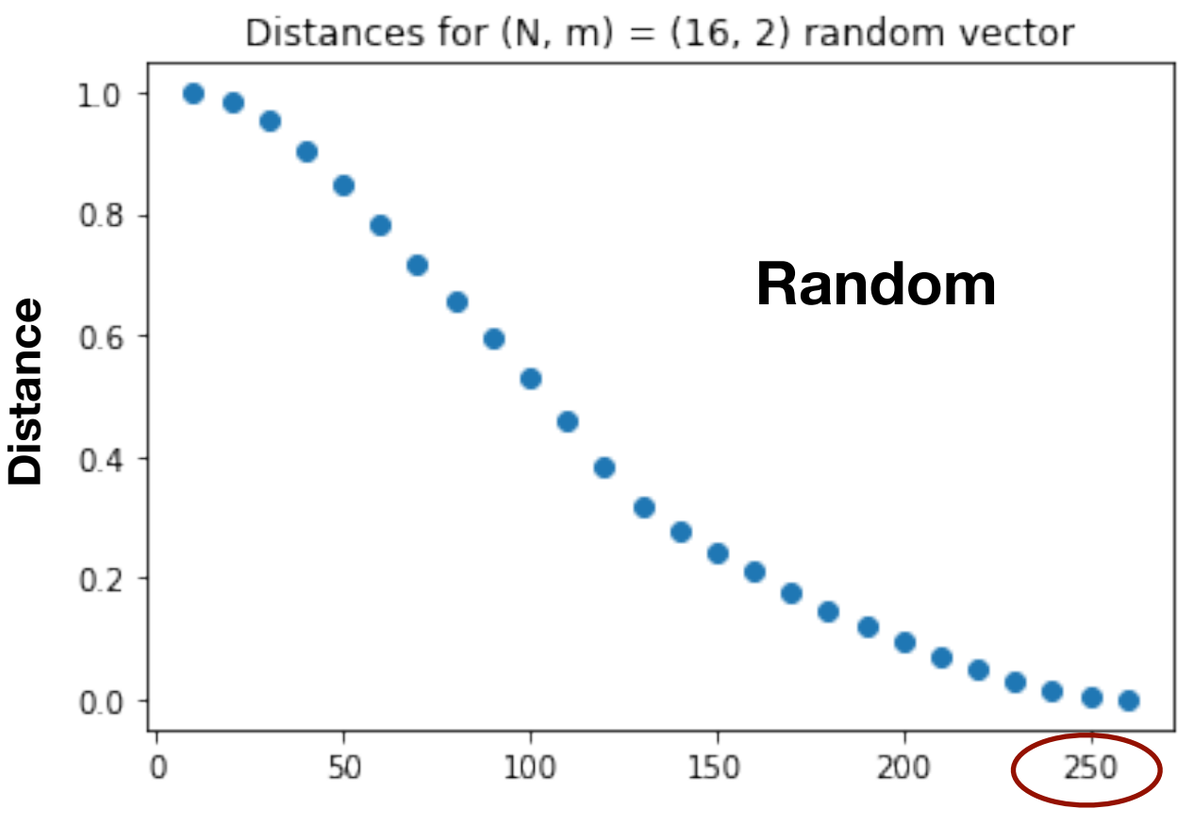
$(\alpha \leq 1)$

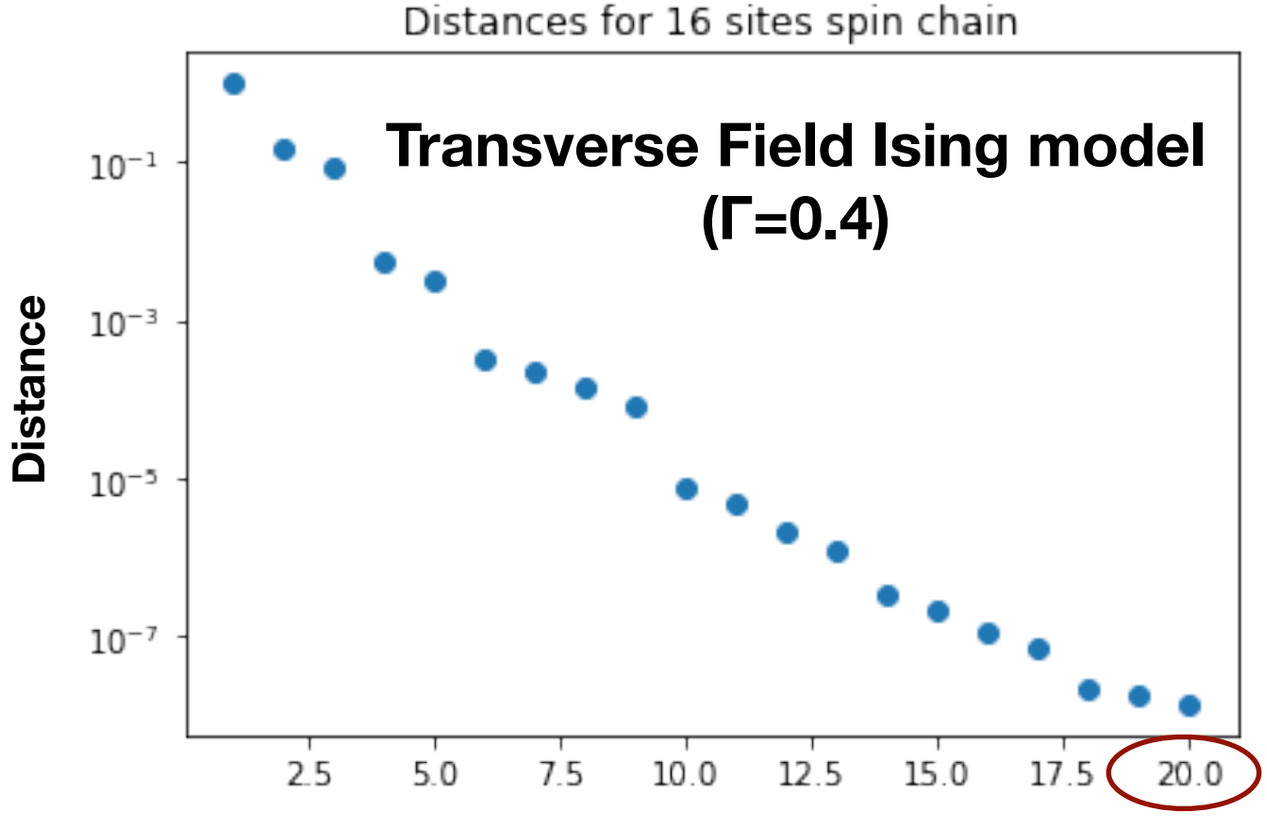
Accuracy of MPS in physics

16 qubits:

Distance from the exact state: $\| |\Psi\rangle - |\Psi_{\mathrm{MPS}}\rangle \|$

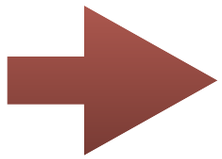
$$\mathcal{H} = -\sum_{i=1}^{L-1} S_{i,z} S_{i+1,z} - \Gamma \sum_{i=1}^{L} S_{i,x}$$





Time evolution of a quantum system

Schrödinger equation: $i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}|\psi(t)\rangle$



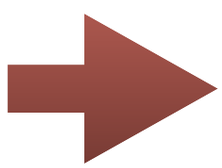
Formal solution:

$$|\psi(t)\rangle = e^{-it\mathcal{H}/\hbar}|\psi(0)\rangle$$

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

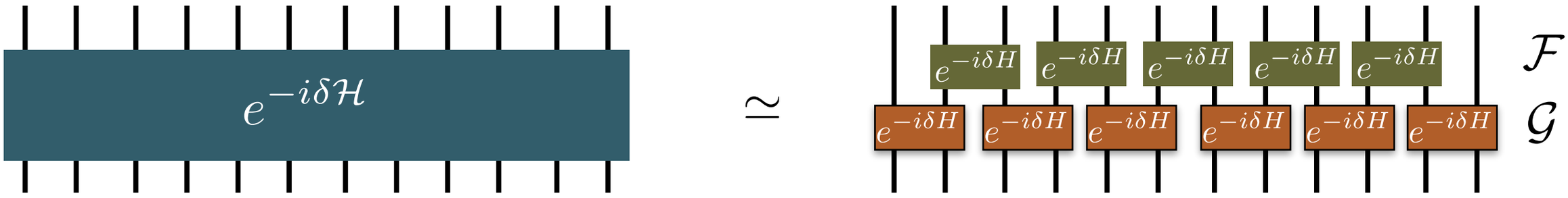
$$\mathcal{H} = \sum_i H_i = \sum_{i \in \text{even}} H_i + \sum_{i \in \text{odd}} H_i$$

$$= \mathcal{F} + \mathcal{G} \qquad\qquad [\mathcal{F}, \mathcal{G}] \neq 0$$

Suzuki-Trotter decomposition of time evolution operator

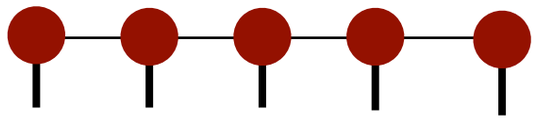
$$e^{-i\delta\mathcal{H}} = e^{-i\delta\mathcal{F}} e^{-i\delta\mathcal{G}} + O(\delta^2)$$

(1st order)

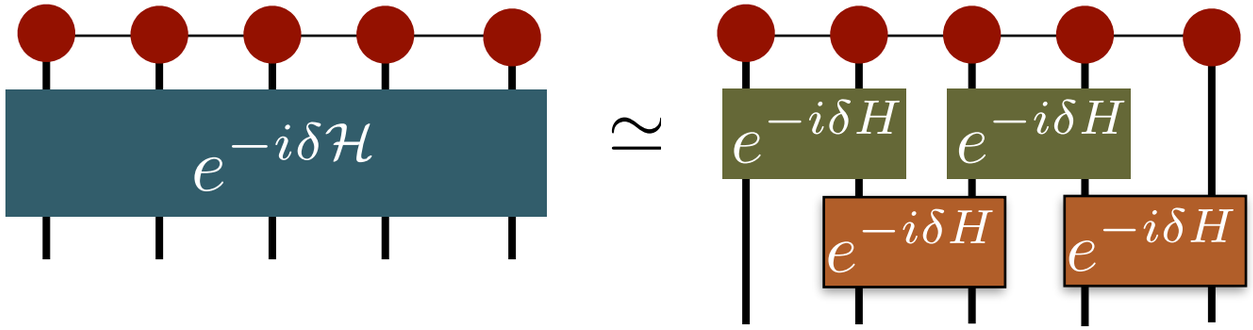


Multiplication of time evolution operator

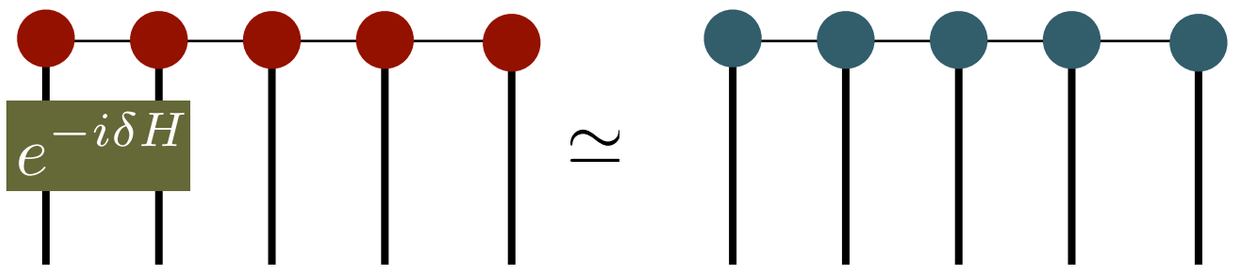
If we have MPS representation of $|\psi\rangle$

$|\psi\rangle =$ 

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)

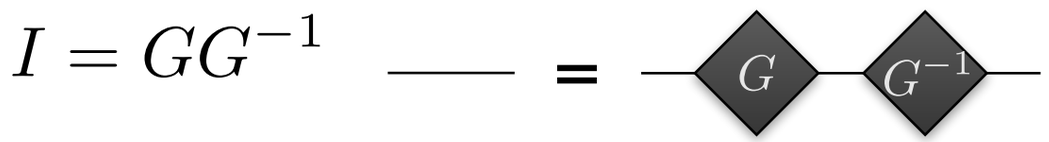
We continue the time evolution repeatedly.

Notice: we want to keep the bond dimension χ constant along time evolution.

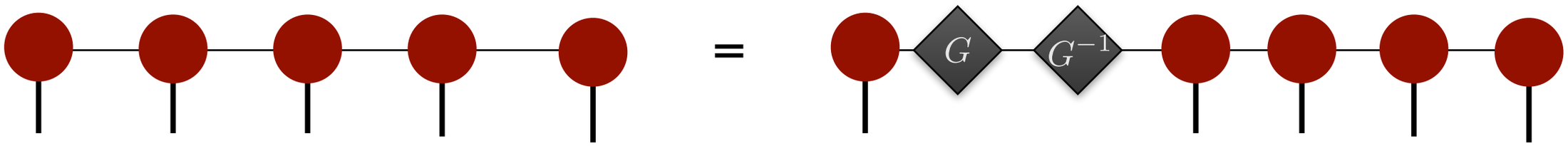
TEBD algorithm

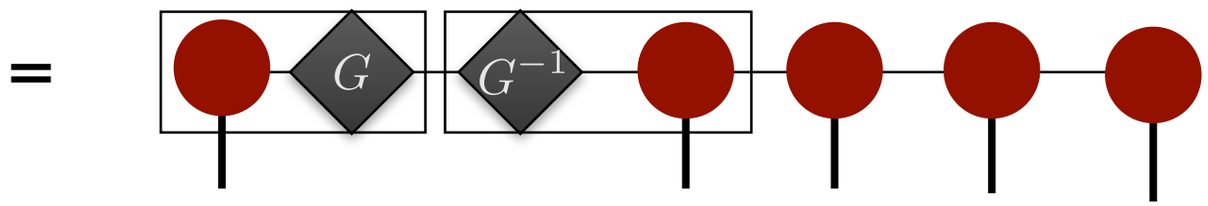
Gauge redundancy of MPS

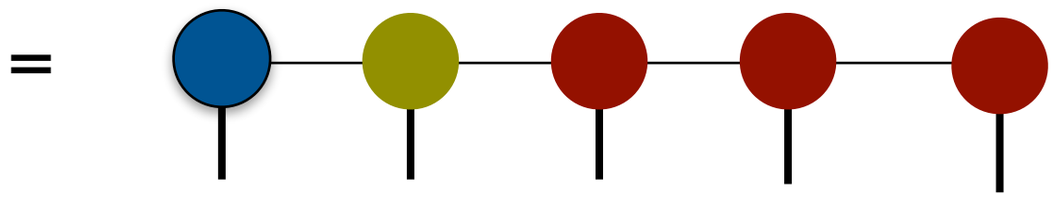
MPS is not unique: gauge degree of freedom $I = GG^{-1}$



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



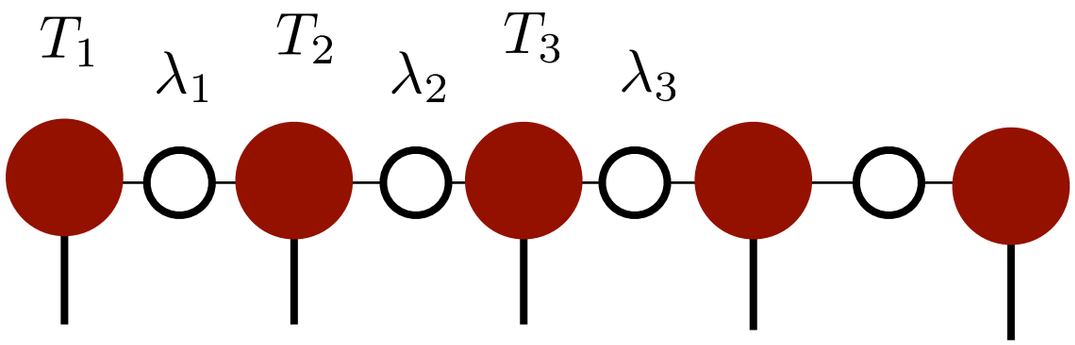




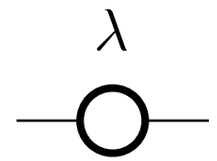
Canonical form of MPS

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

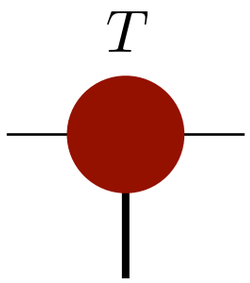
Vidal canonical form



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

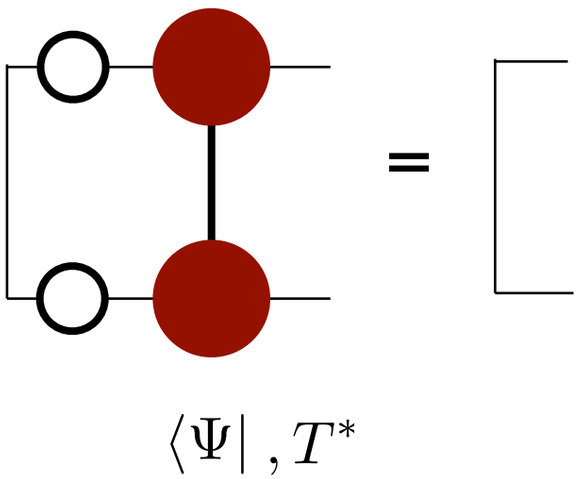


:Diagonal matrix corresponding to Schmidt coefficient

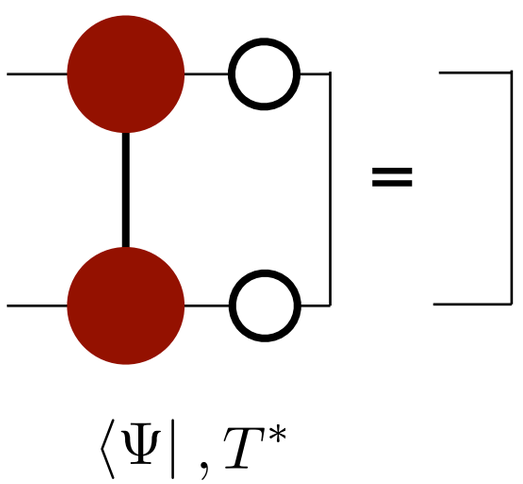


:Virtual indices corresponding to Schmidt basis

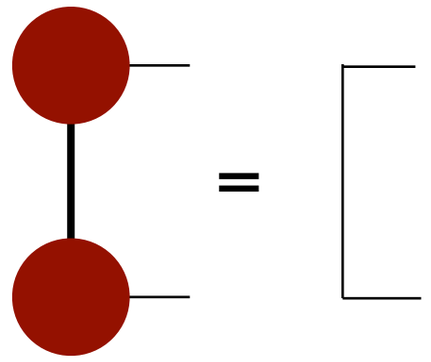
Left canonical condition:



Right canonical condition:

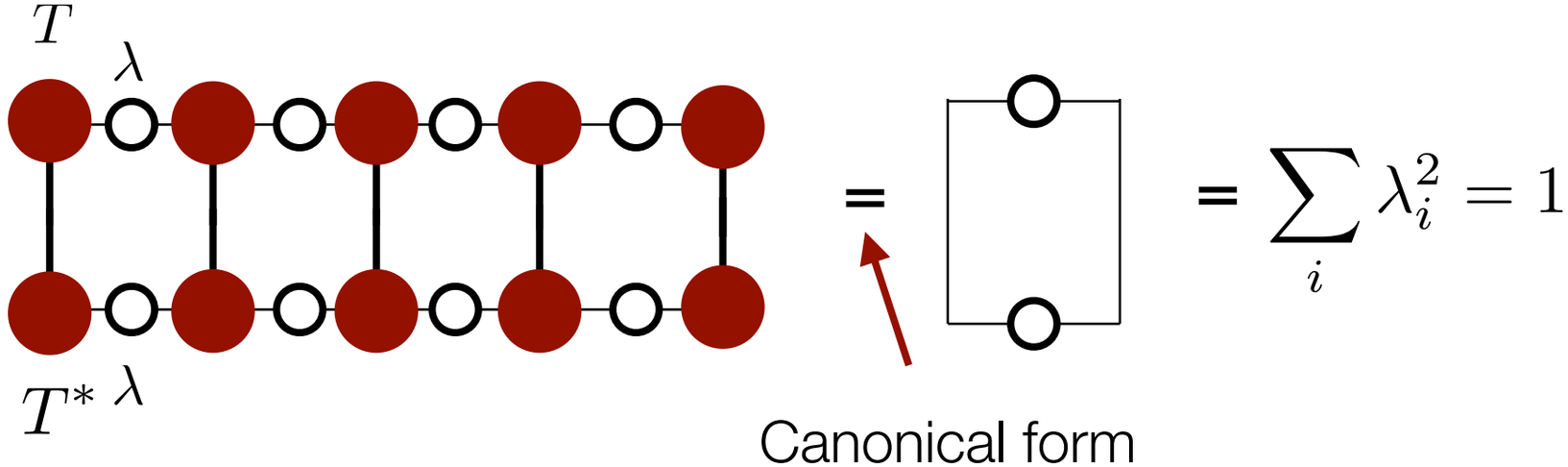


(Boundary)

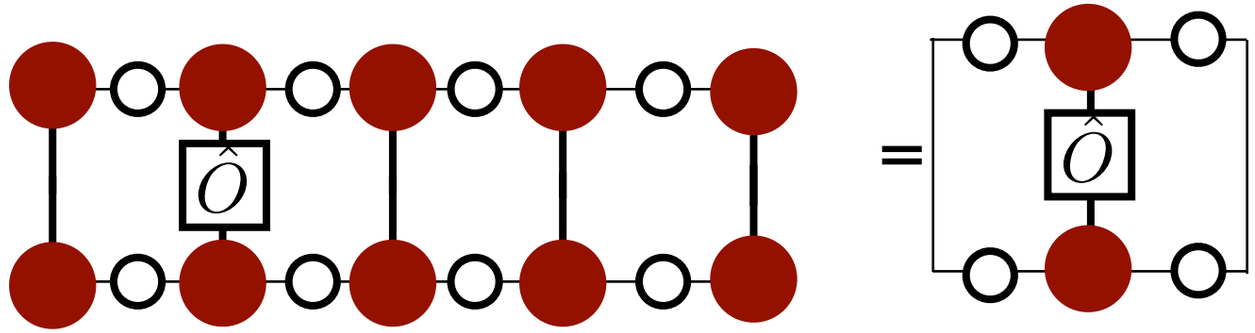


Expectation value with canonical forms

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



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



In the canonical form, the diagrams corresponding to expectation values become local objects.

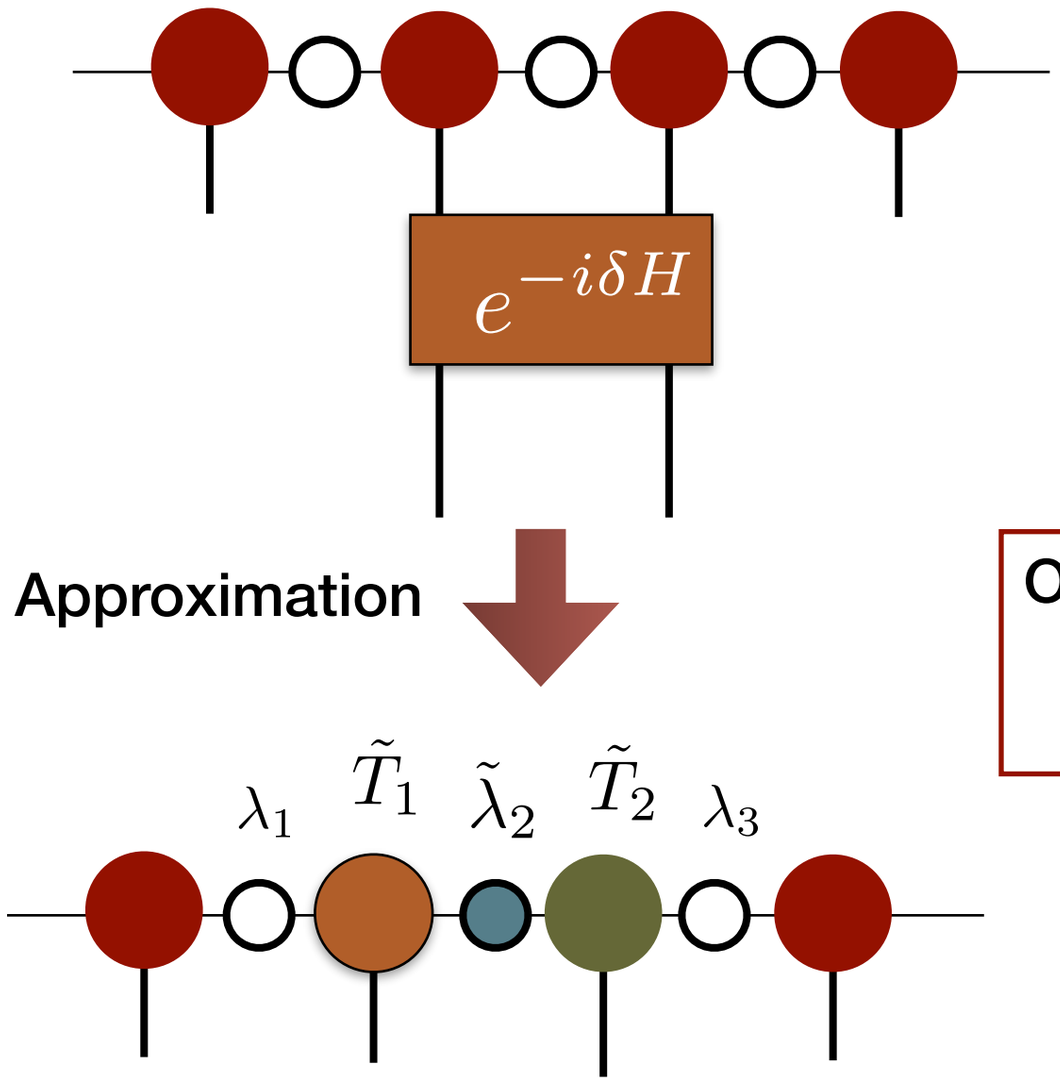
(When we consider a mixed canonical form,
we also obtain a similar simple diagram. (exercise!))

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.

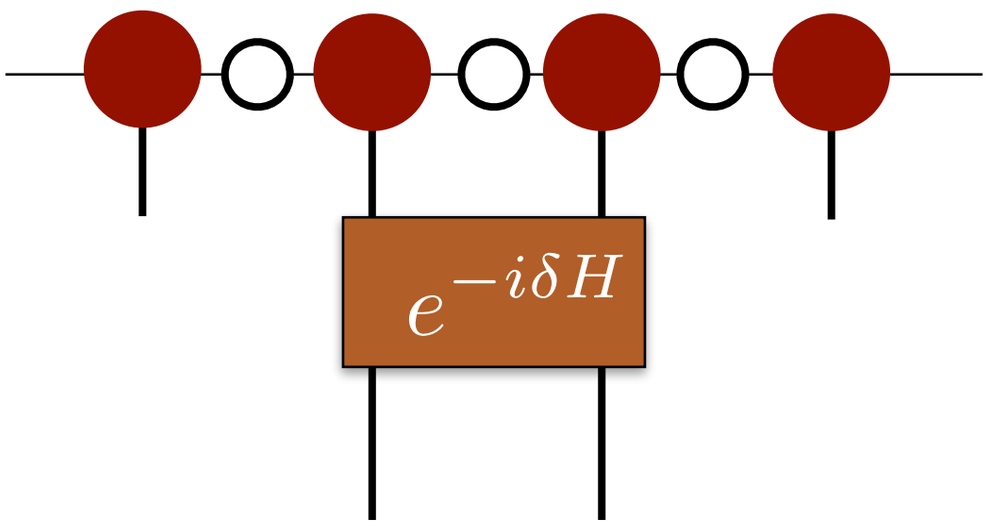


Only the two matrices which are directly applied TE operator changes in MPS.

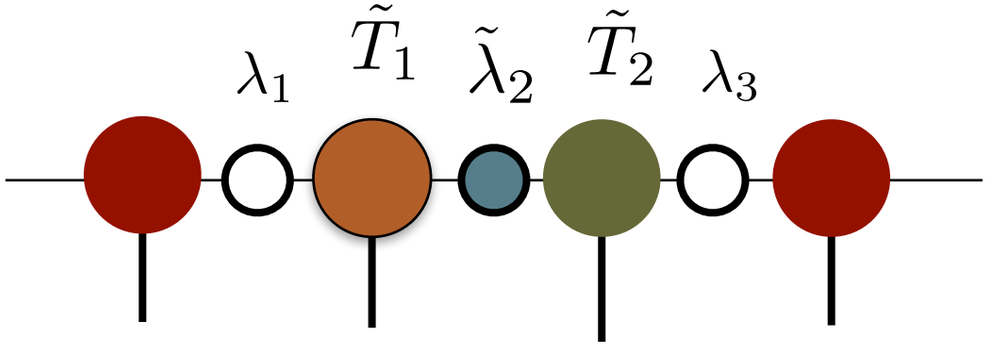
TEBD algorithm:

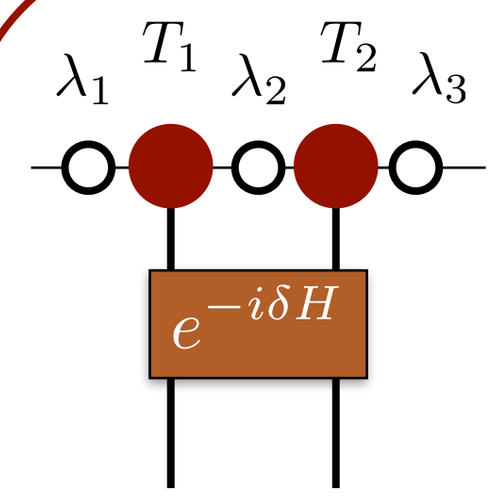
(G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003))

Apply TE operator

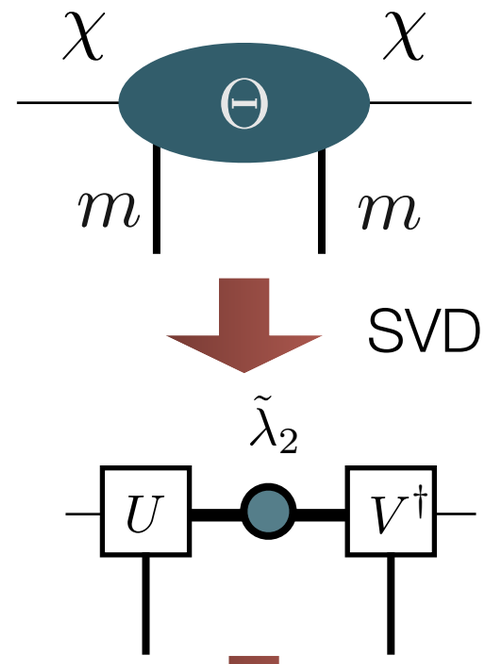


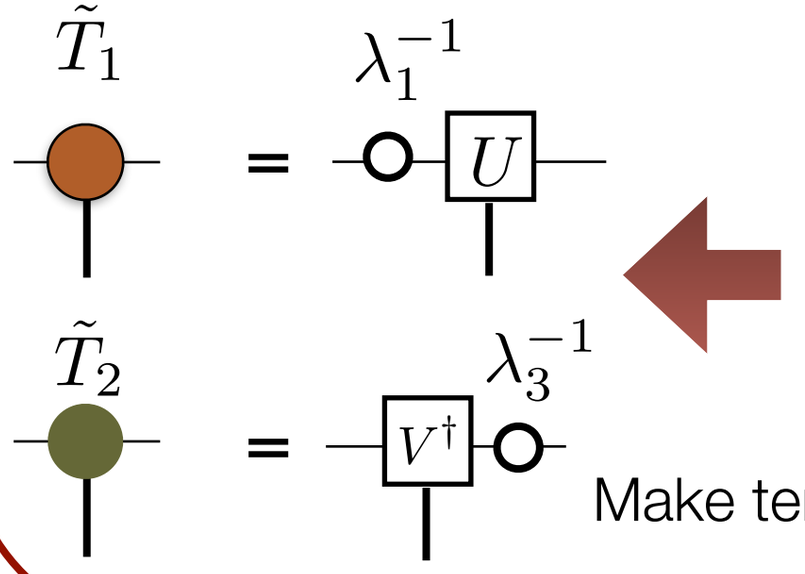
Approximation





Combine and make matrix

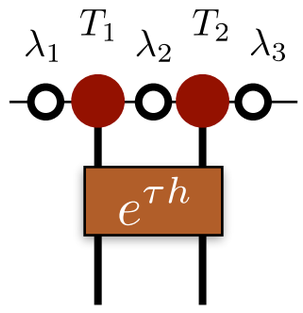




Make tensor

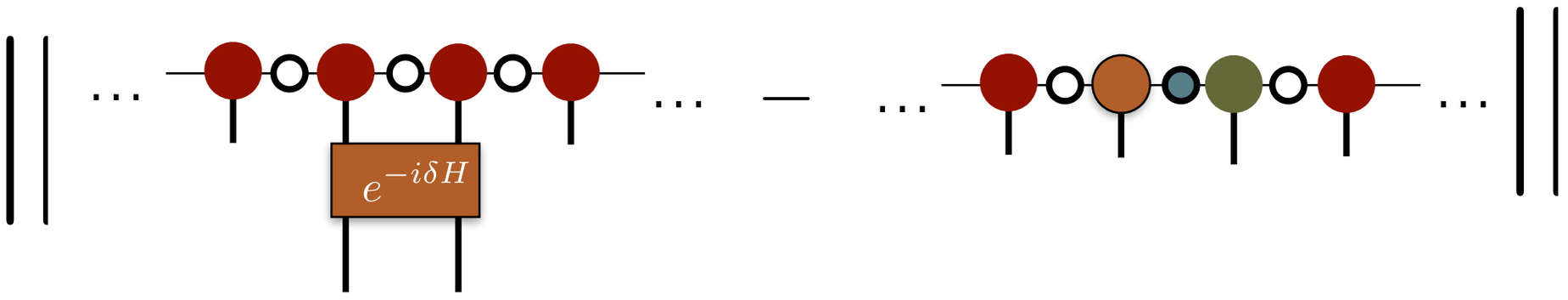
Why TEBD is accurate?

1. For accurate calculation, the canonical form is important.



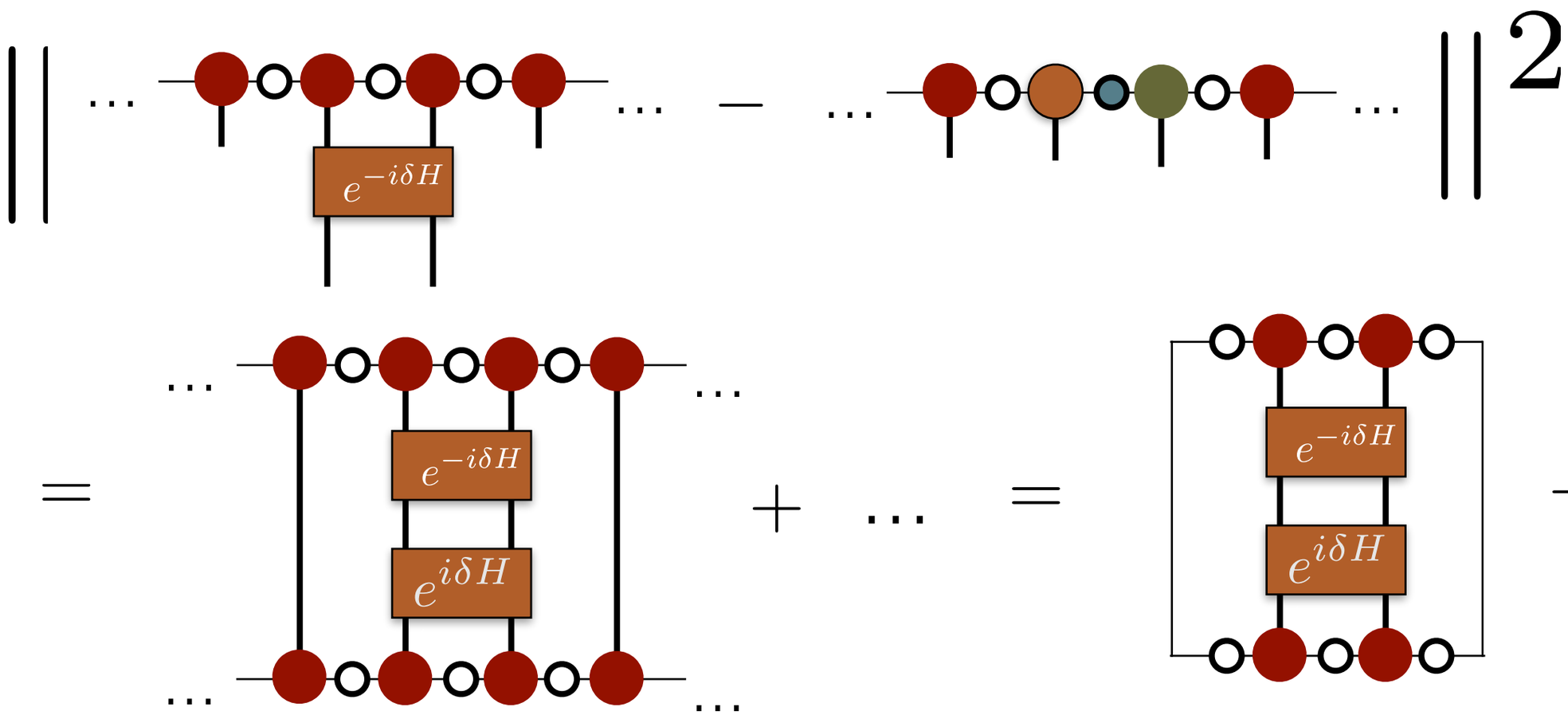
If λ is equal to the Schmidt coefficient, it contains all information of the remaining part of the system.

Due to the canonical form, we can prove that TEBD algorithm minimize the distance of two quantum states:



Truncation based on local SVD can be globally optimal, even if we look at a part of the MPS.

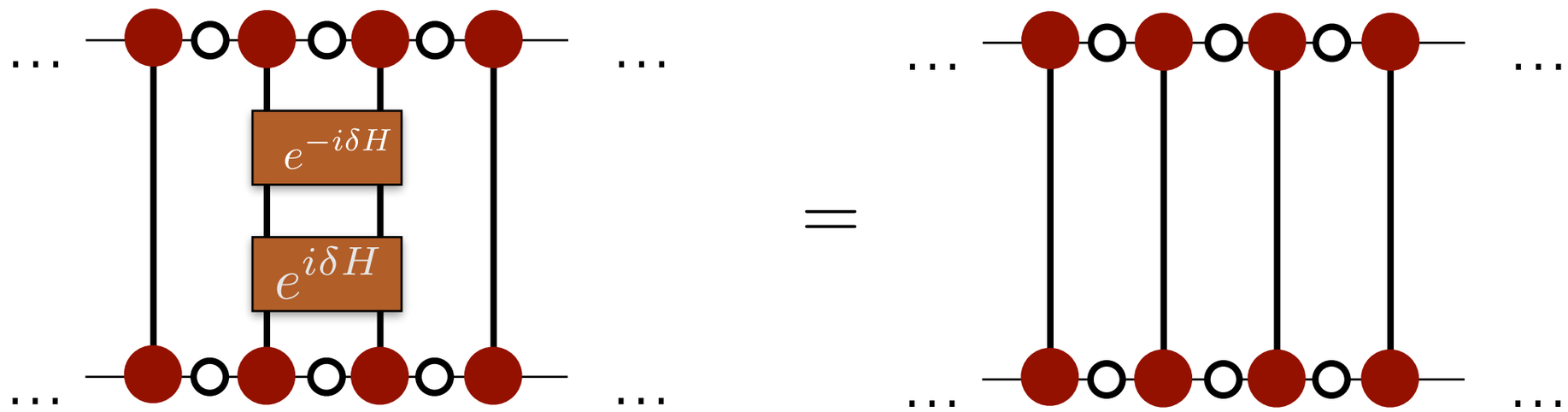
Calculation of the distance



*This is minimized by the matrix Θ

Why TEBD is accurate?

2. If the operator is unitary, MPS keeps canonical form within truncation error



*Unitary operator does not affect to the other Schmidt coefficients

If we chose the initial MPS as the canonical form,
TEBD algorithm almost keep it.
(So, TEBD is almost "globally optimal")

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.

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,

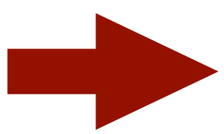
$$e^{-i\mathcal{H}t} \to e^{-\tau\mathcal{H}} \qquad (t \to -i\tau)$$

We can use (TEBD) algorithm for eigenvalue problem.

Difference between TE and ITE

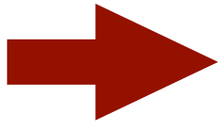
[Diffferences in practical applications]

1. In imaginary-time evolutions, the quantum state changes to the ground state, indicating less entanglement. In contrast, entanglement increases in real-time evolutions.



For longer real-time simulations, we need larger bond dimensions to keep the accuracy.

2. In imaginary-time evolutions, we are typically interested in the final state (the ground state), while the trajectories are also important in real-time evolutions.



The accuracies at each time slice are affected by the error in Suzuki-Trotter decomposition in addition to the bond dimensions.

Contents

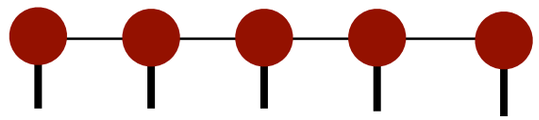
Contents

- Basics of tensor network methods: part 1
  - Area law
  - MPS and TEBD algorithm
  - Hands-on 1: Real-time evolution by TEBD
- Basics of tensor network methods: part 2
  - TPS (PEPS) and simple update
  - Hands-on 2: Real-time evolution by TEBD for MPS and PEPS in 2d
- Basics of tensor network methods: part 3
  - Infinite tensor networks
  - Hands-on 2: Real-time evolution by iMPS and iPEPS

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
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

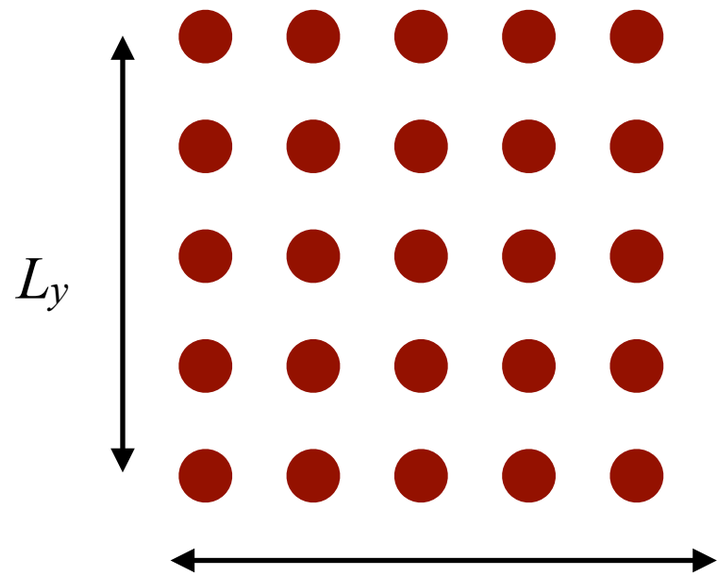
Higer dimensional system

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



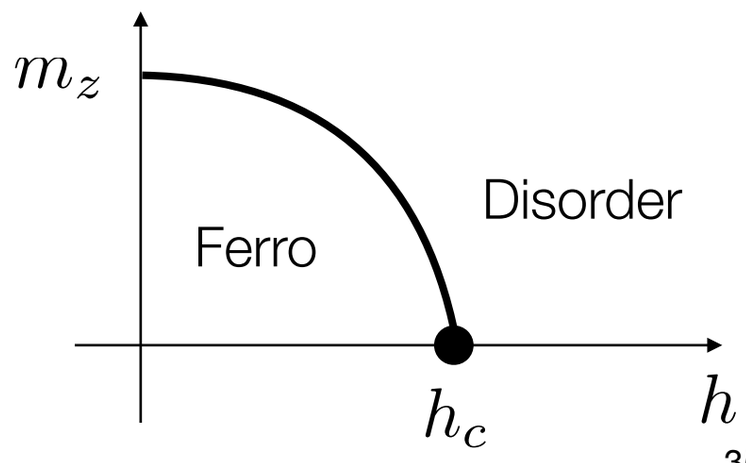
$$N = L_x \times L_y$$

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

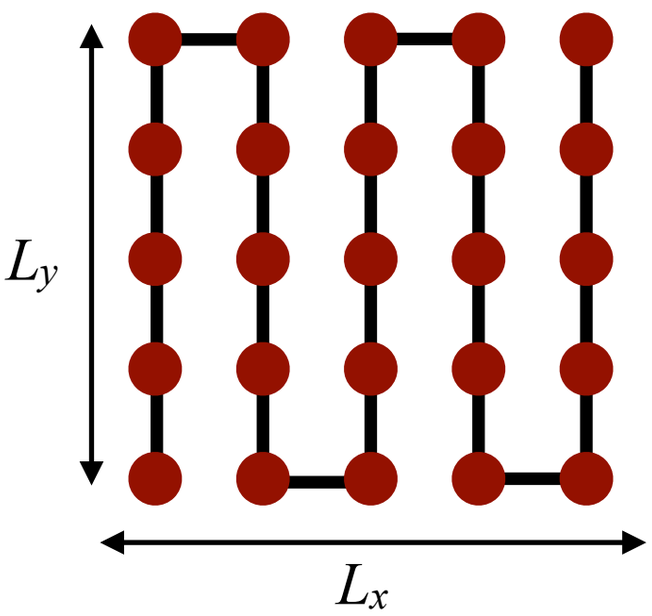
When we apply MPS representation for a square lattice system:

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!

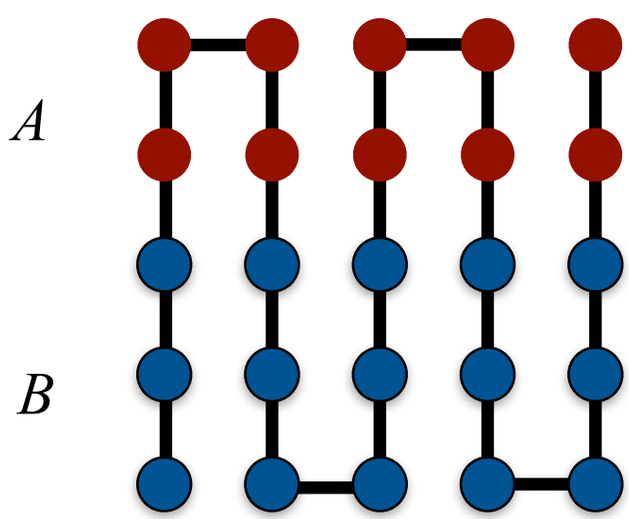
MPS cannot cover the area law of the entanglement entropy in higher (d =2,3, ...) dimensions.

Possible MPS (Snake form)

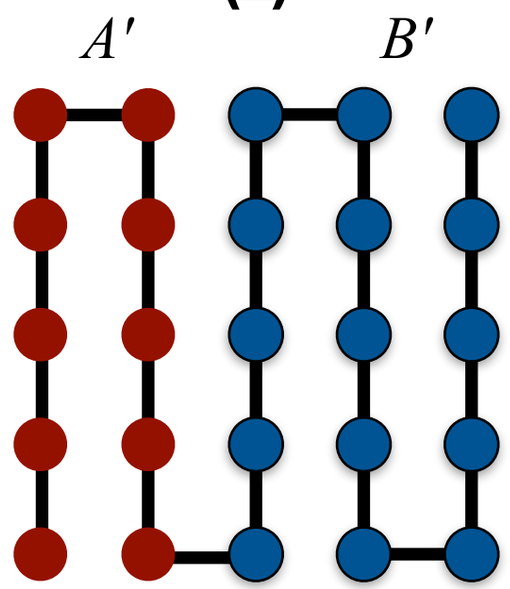


Two settings of system and environment

(1)



(2)



MPS for two-dimensional system: comment

MPS can treat "**rectangular**" or "**quasi one dimensional**" lattice.

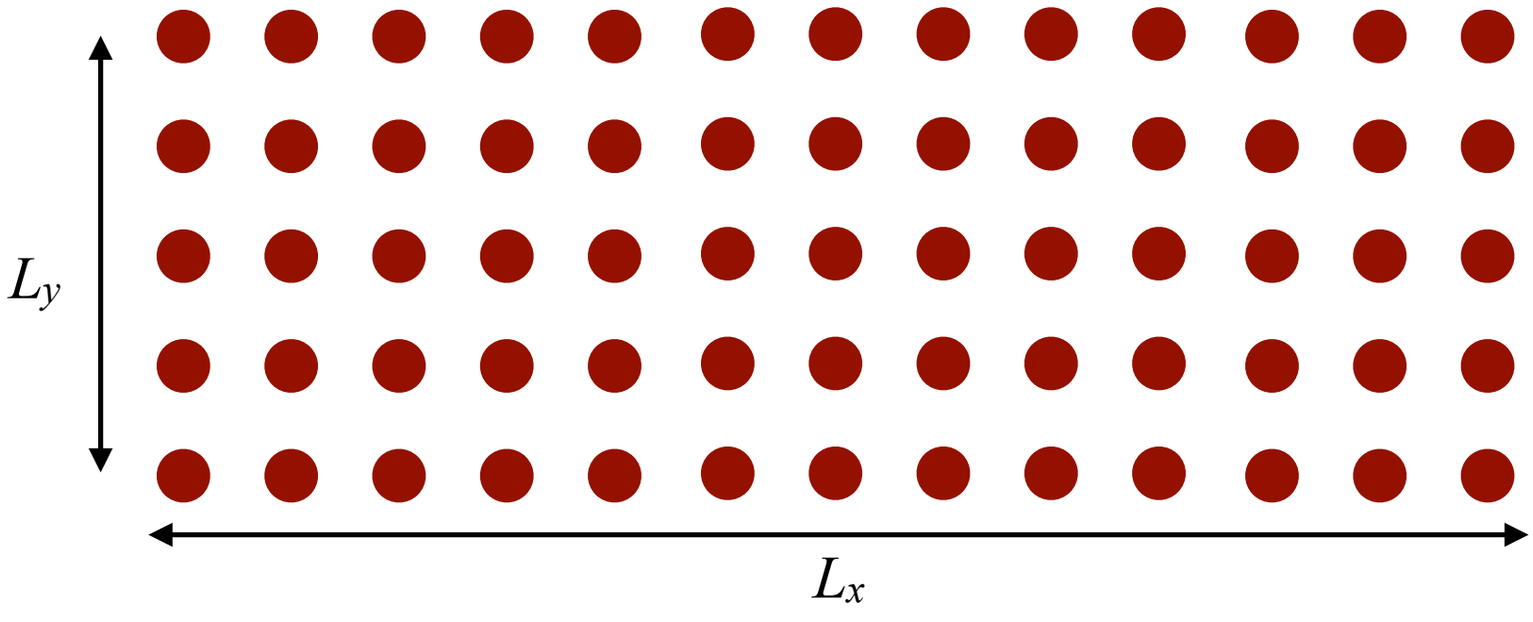
In setting (1), MPS can satisfy the area low partially.

We can increase L_x easily with keeping L_y constant.

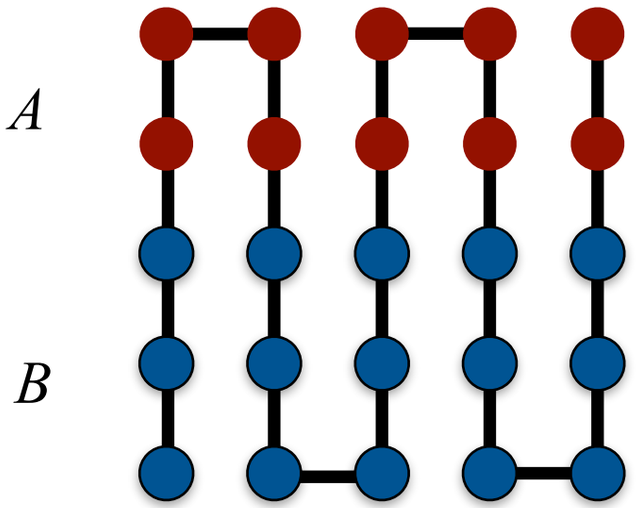
$$\chi = O(e^{L_y})$$

$$L_y \lesssim 10, L_x \gg L_y$$

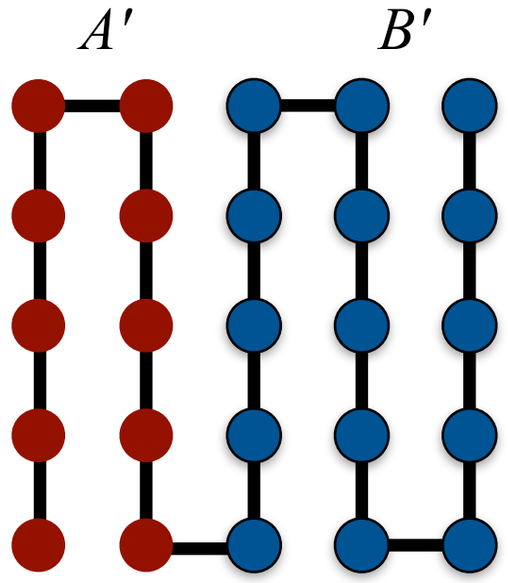
Quasi one dimensional system ("strip" or "cylinder")



(1) $S_A \leq L_x \log \chi$



(2) $S_{A'} \leq \log \chi$



Tensor network for higher dimensional systems: Tensor Product State (Projected Entangled Pair State)

Entanglement entropy in higher dimensions

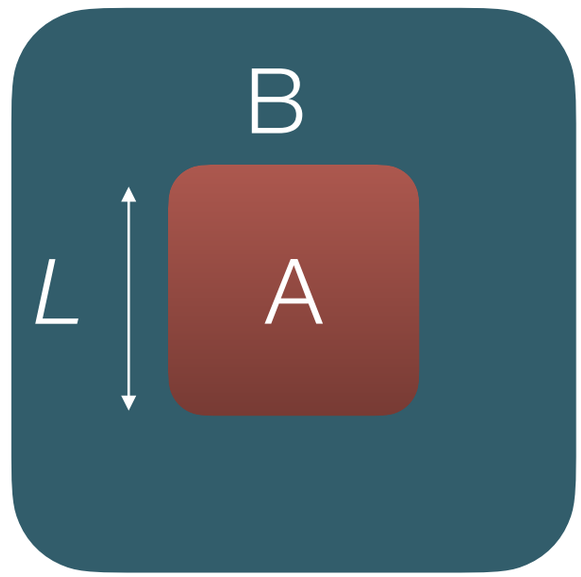
Ground state wave functions:

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?

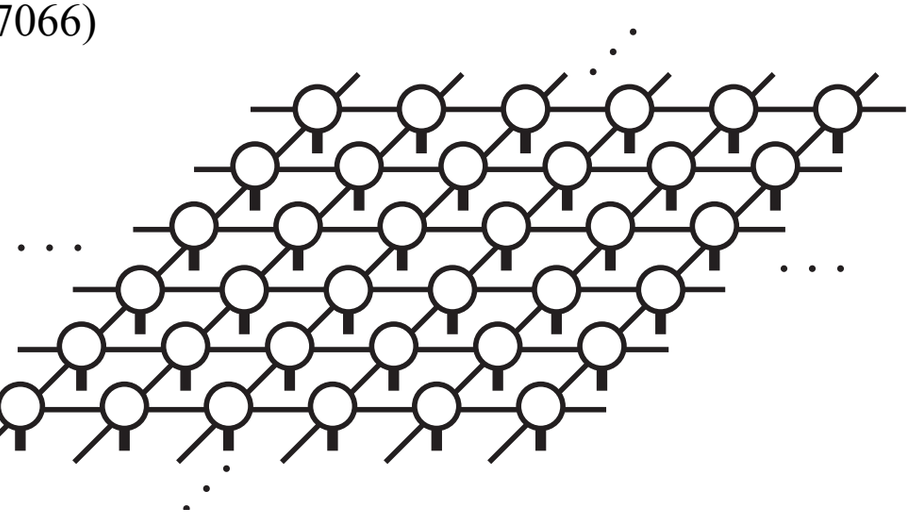
A. It is Tensor Product State (TPS)!

Tensor product states (TPS)

TPS (Tensor Product State) (AKLT, T. Nishino, K. Okunishi, …)

PEPS (Projected Entangled-Pair State)

(F. Verstraete and J. Cirac, arXiv:cond-mat/0407066)



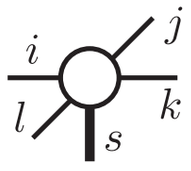
Ex. : Square lattice TPS

Network consist of 4+1-leg tensors

local degree：*s*

Virtual degrees：*i, j, k, l*

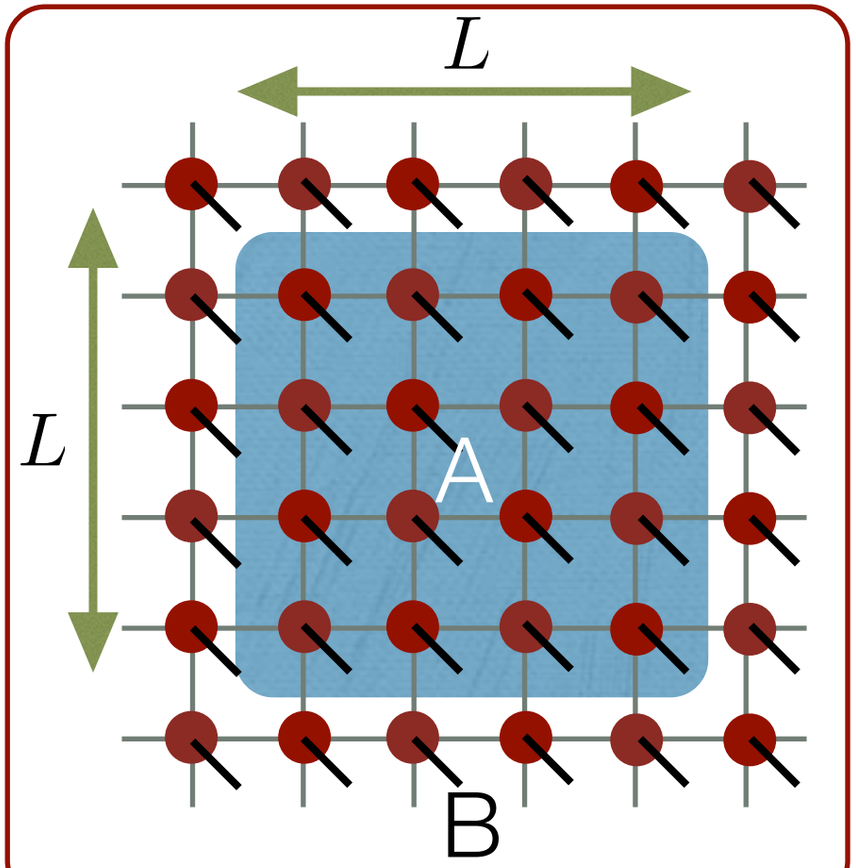
$$T_{ijkl}[s] =$$

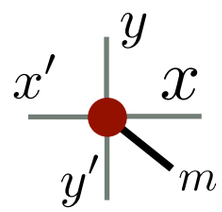


Dimension of each indices ＝ **bond dimension（*D*）**

It determines the accuracy of the approximated wavefunction. **(*D*→∞ it becomes exact.)**

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.

Difference between MPS and TPS

Cost of tensor network contraction:

d-dimensional cubic lattice $N = L^d$

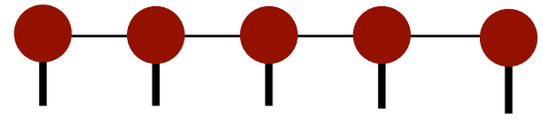
MPS: $O(N)$

TPS: $O(e^{L^{d-1}})$

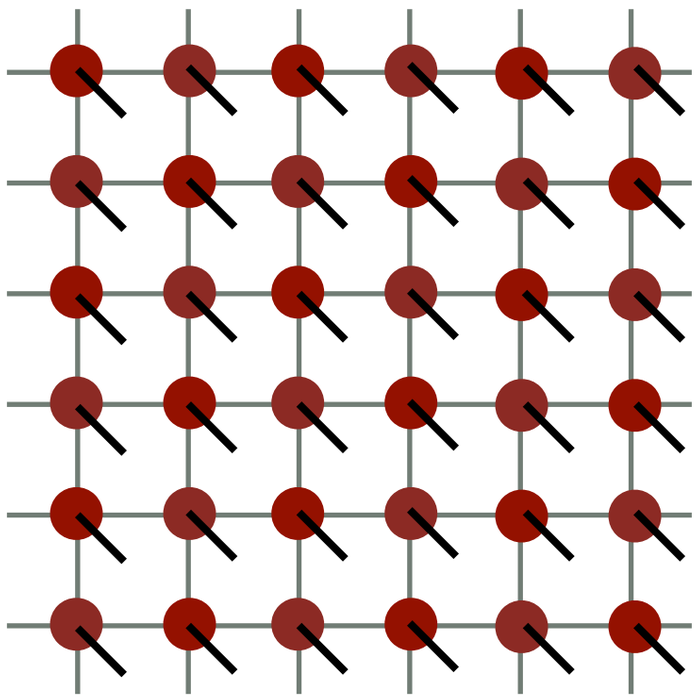
It is impossible to perform exact contraction even if we know local tensors in the case of TPS.

In the case of TPS, usually we approximately calculate the contraction.

MPS



TPS (PEPS)



Truncations in time evolutions

- Full update

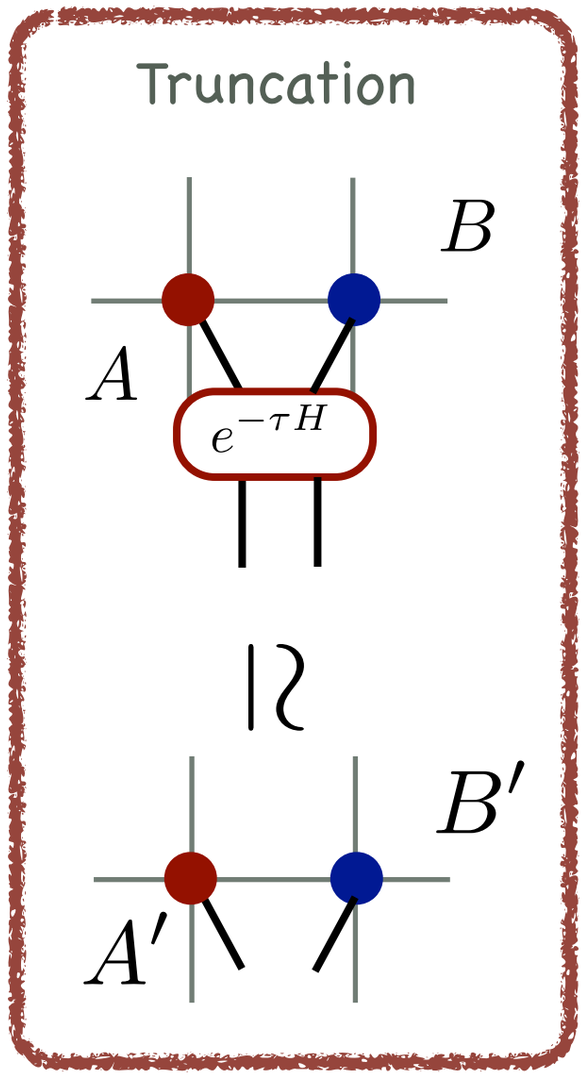
  Minimize the difference between two wave functions:

$$||\Psi\rangle - |\Psi'\rangle||^2 = \langle\Psi|\Psi\rangle + \langle\Psi'|\Psi'\rangle - 2\mathrm{Re}\ \langle\Psi|\Psi'\rangle$$

  $|\Psi\rangle$: wave function (after ITE)

  $|\Psi'\rangle$: wave function after truncation

  - Ideal approximation for finite TPS
  - We need tensor network contractions, $O(D^8) \sim O(D^{10})$



- Simple update (H. G. Jiang *et al*, Phys. Rev. Lett. **101**, 090603 (2008))

  Truncation by using local information

  - Low computation cost : $O(D^5)$
  - TPS tends to represent only short-range correlations

Simple update

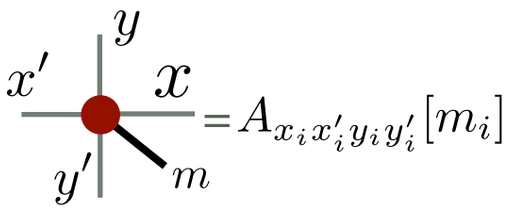
(H. G. Jiang *et al*, Phys. Rev. Lett. **101**, 090603 (2008))

Extended TPS:

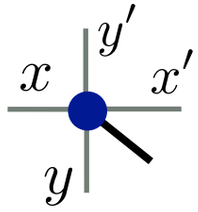
Insert (positive) diagonal matrix representing "weight" of bonds.

(cf. TEBD)

$$|\Psi\rangle = \mathrm{Tr} \prod_{i\in A, j\in B} \lambda_{x_i}\lambda_{x'_i}\lambda_{y_i}\lambda_{y'_i} A_{x_i x'_i y_i y'_i}[m_i] B_{x_j x'_j y_j y'_j}[m_j] |m_i m_j\rangle$$



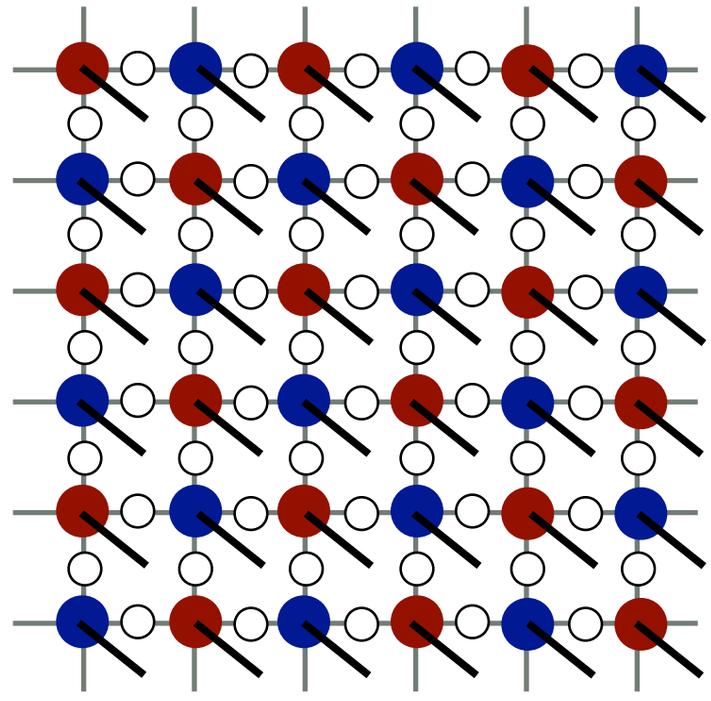
$= A_{x_i x'_i y_i y'_i}[m_i]$



$= B_{x_j x'_j y_j y'_j}[m_j]$ —○— $= \lambda$

x,y,x',y' = 1,2, ... D

Extended PEPS



Simple update with naive SVD

$$e^{-\tau H_x}|\Psi\rangle = \mathrm{Tr}\prod_{i\in A, j\in i+x}\sum_{m_i,m_j}\langle m_i' m_j'|e^{-\tau H_{ij}}|m_i m_j\rangle \lambda_{x_i}\lambda_{x_i'}\lambda_{y_i}\lambda_{y_i'}A_{x_i x_i' y_i y_i'}[m_i]B_{x_j x_j' y_j y_j'}[m_j]|m_i' m_j'\rangle$$

Truncation by SVD

1. Define a matrix “S”

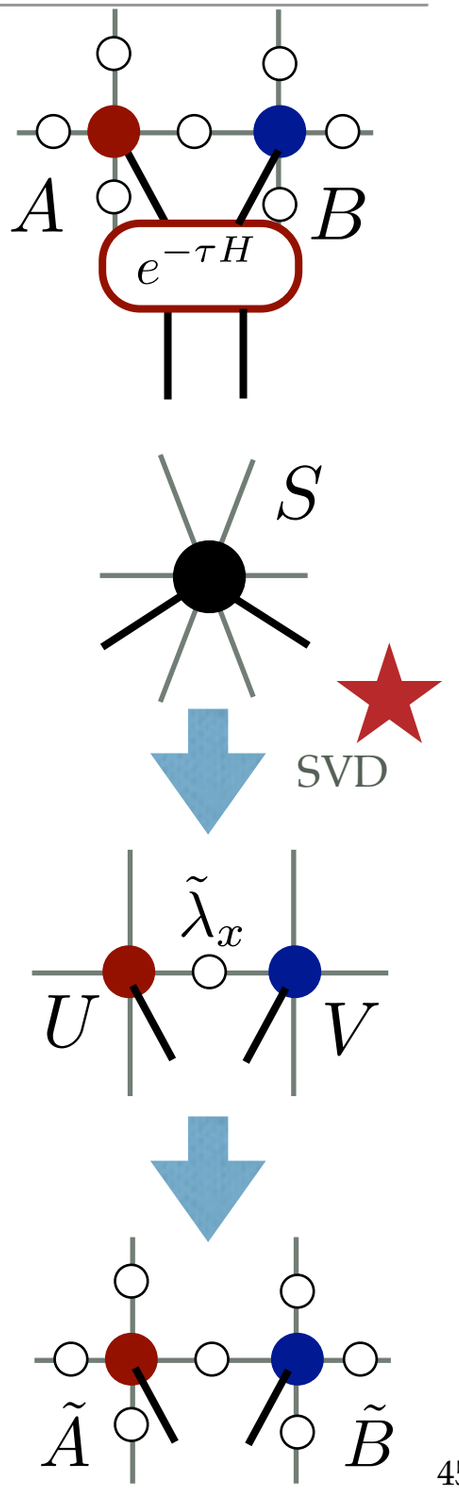
$$S_{y_i x_i' y_i' m_i, y_j x_j' y_j' m_j} = \sum_{m_i,m_j}\sum_{x}\langle m_i' m_j'|e^{-\tau H_{ij}}|m_i m_j\rangle \lambda_{y_i}\lambda_{x_i'}\lambda_{y_i'}A_{x y_i x_i' y_i'}[m_i]\lambda_x B_{x y_j x_j' y_j'}[m_j]\lambda_{y_j}\lambda_{x_j'}\lambda_{y_j'}$$

2. Do SVD ★

$$S_{y_i x_i' y_i' m_i, y_j x_j' y_j' m_j} = \sum_x U_{y_i x_i' y_i' m_i, x}\tilde{\lambda}_x V^T_{x, y_j x_j' y_j' m_j}$$

3. Truncate the matrix leaving upper D singular values

$$\tilde{A}_{x y_i x_i' y_i'}[m_i] = \lambda_{y_i}^{-1}\lambda_{x_i'}^{-1}\lambda_{y_i'}^{-1}U_{y_i x_i' y_i' m_i, x}$$
$$\tilde{B}_{x y_j x_j' y_j'}[m_j] = \lambda_{y_j}^{-1}\lambda_{x_j'}^{-1}\lambda_{y_j'}^{-1}V_{y_j x_j' y_j' m_j, x}$$



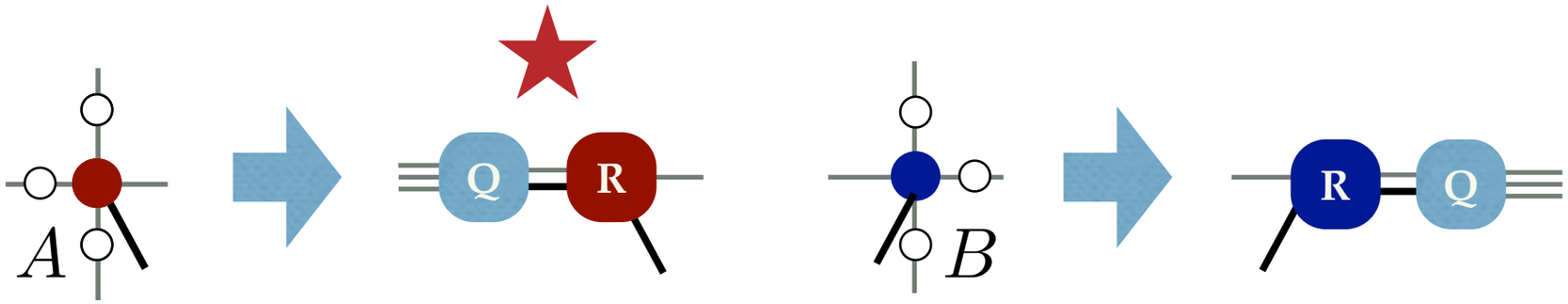
*** Meaning of λ**

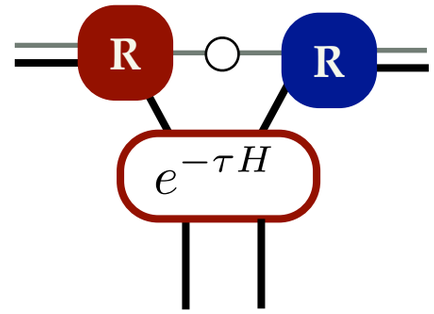
At SVD, λ provides information of local environment.

⟷ In the case of iMPS, λ give us global information, thanks to the canonical form.

Simple update with QR decompositions

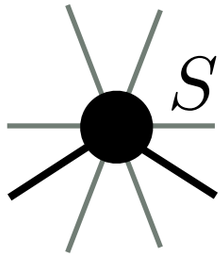
QR decomposition before SVD





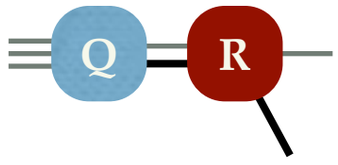
Calculation cost

Direct SVD:

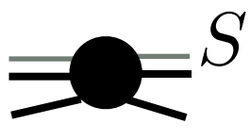


$O(D^9 m_d^3)$

QR decomposition:

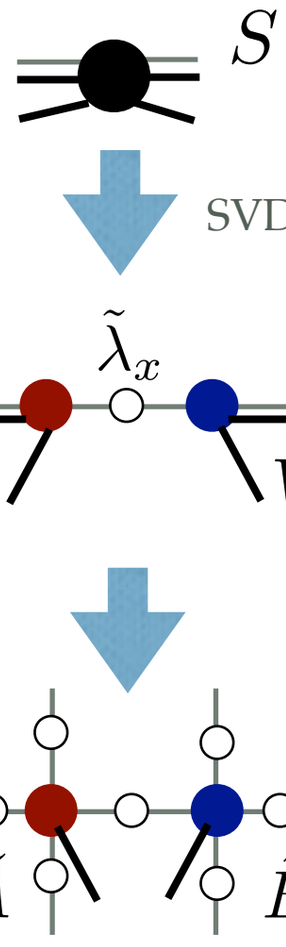


$O(D^5 m_d^2)$



$O(D^3 m_d^6)$

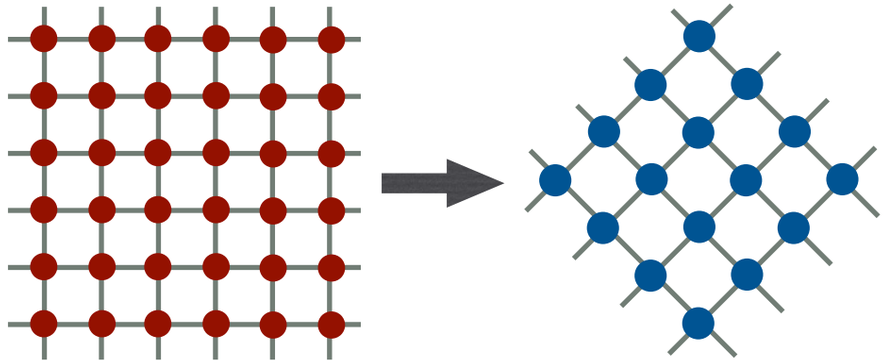
Usually D > m_d → QR method is cheaper.



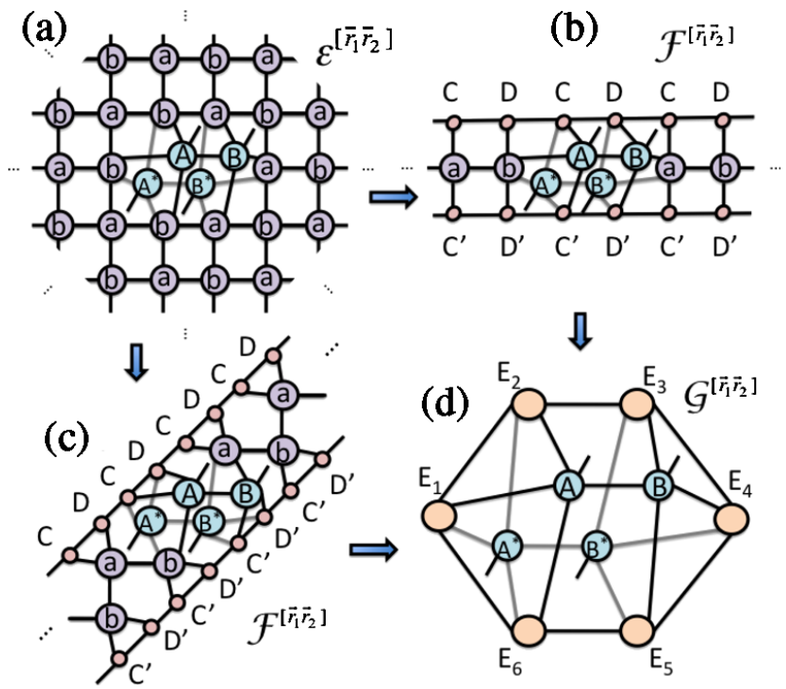
Contraction of TPS

Methods for approximate contraction of TPS:

- Tensor network renormalizations
  - TRG, HOTRG, SRG, TNR, loop-TNR, ...
- Boundary MPS
  - (Y. Hieida *et al* (1999) , J. Jordan *et al*, Phys. Rev. Lett. **101**, 250602 (2008))



- Corner transfer matrix
  - T. Nishino and K. Okunishi, JPSJ **65**, 891 (1996), R. Orús *et al*, Phys. Rev. B **80**, 094403 (2009).
- Single layer approaches
  - bMPS: H. J. Liao *et al,* PRL **118**, 137202 (2017), Z. Y. Xie *et al*,PRB **96**, 045128 (2017).
  - CTM: Chih-Yuan Lee *et al*, PRB **98**, 224414 (2018) .
- Mean-field environment
  - S. Jharomi and R. Orús, PRB **99**, 195105 (2019).

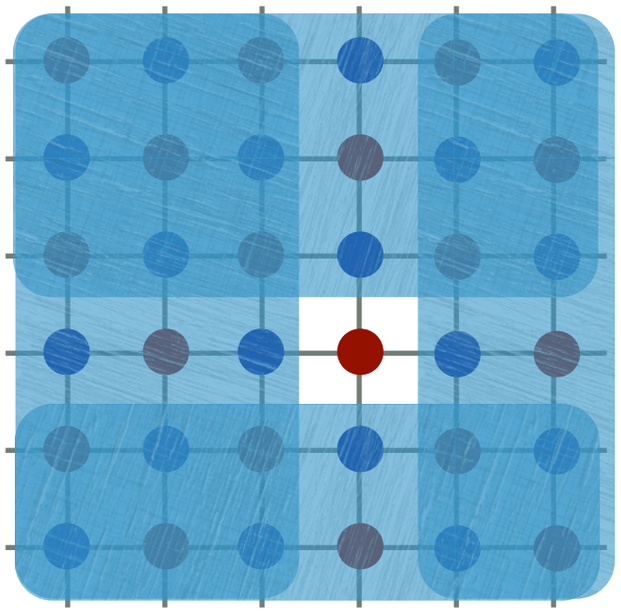


Example of approximate contraction: CTM method

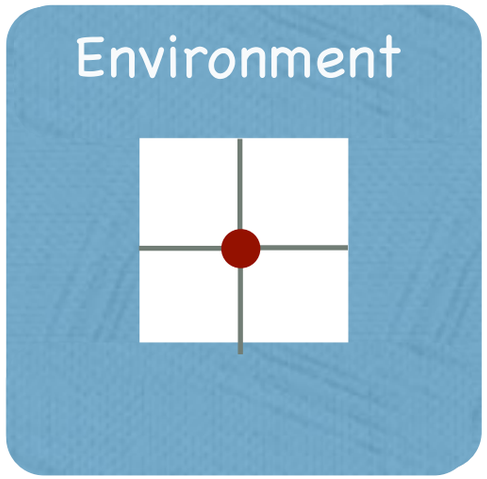
For open boundary system

(T. Nishino and K. Okunishi, JPSJ **65**, 891 (1996))
(R. Orus *et al*, Phys. Rev. B **80**, 094403 (2009))

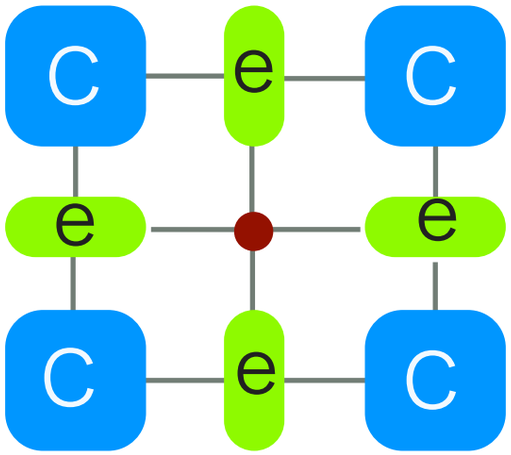
Open boundary PEPS



Environment

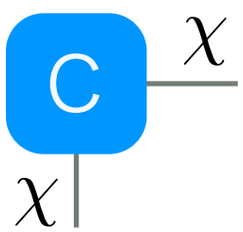


Corner transfer matrix Representation

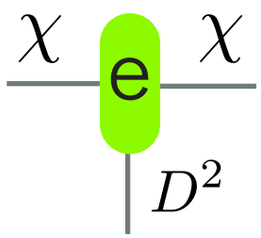


Double tensor

Corner transfer matrix



Edge tensor

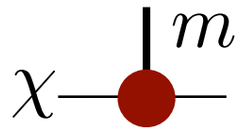


Mapping to a "classical" system

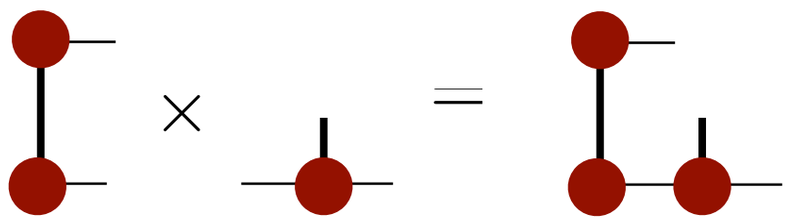
χ =bond dimension $\chi \sim D^2$

Cost of (approximate) contraction

MPS:

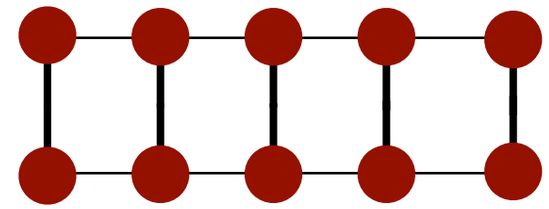


$(m \ll \chi)$

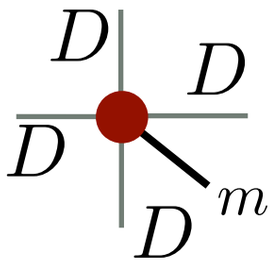


$O(\chi^3)$

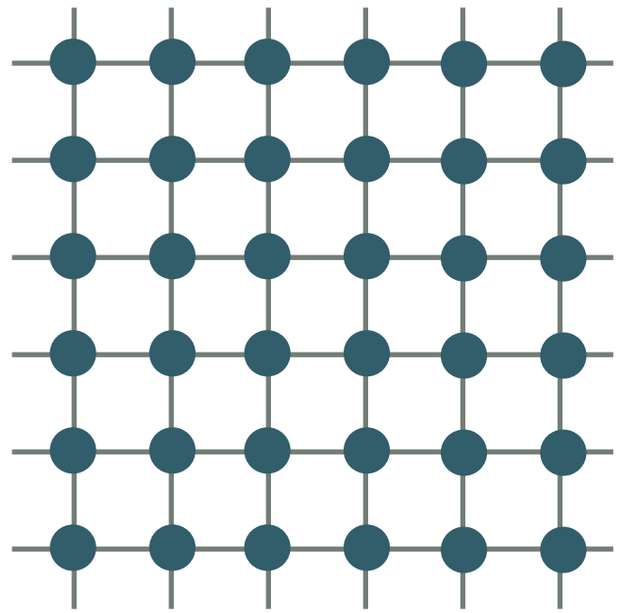
$\langle \Psi | \Psi \rangle =$



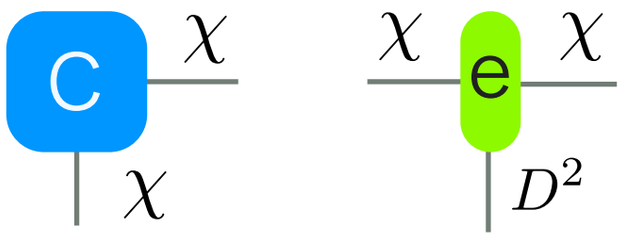
TPS:



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

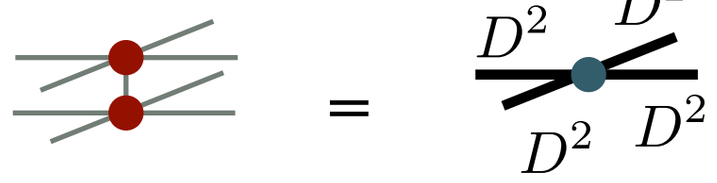


When we use CTM environment in 2D,



$$O(\chi^2 D^6), O(\chi^3 D^4) \sim O(D^{10}) \quad (\chi \sim D^2)$$

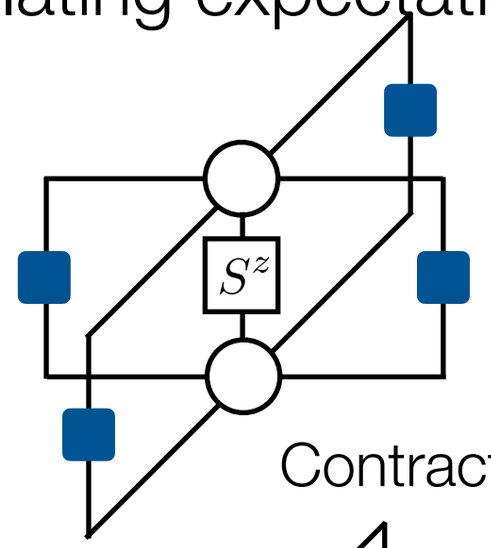
We can treat very small bond dimensions in TPS!



Mean-field environment

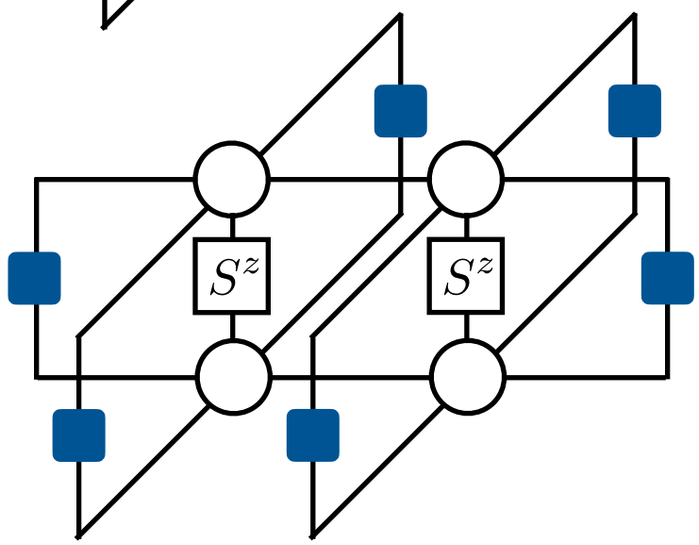
Consider a mean-field-like environment for each bond for calculating expectation values

$$\langle\Psi|S^z|\Psi\rangle =$$



Contraction cost: $O(D^4)$

$$\langle\Psi|S^z_i S^z_{i+1}|\Psi\rangle =$$

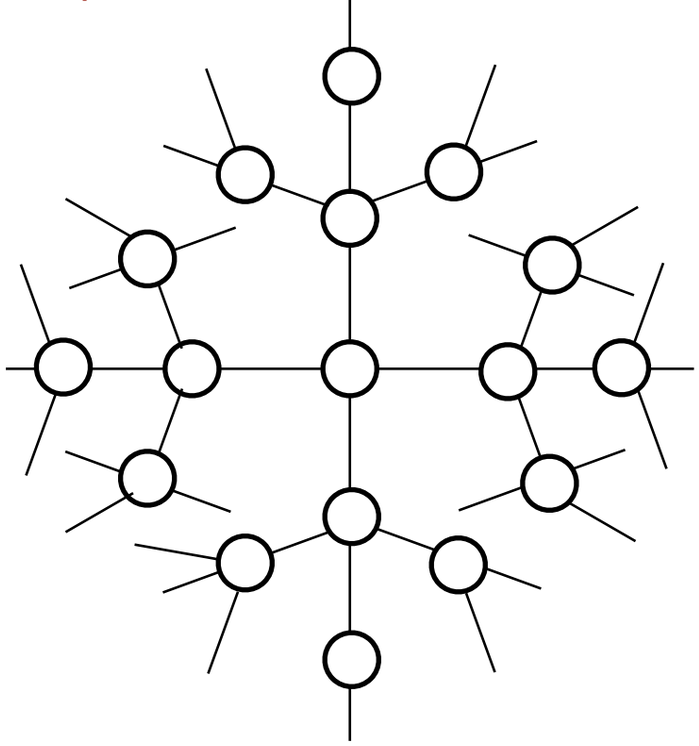


Contraction cost: $O(D^5)$

Although this is very cheap, it contains huge approximation!

: mean-field tensor

It becomes exact when we ignore loops in a tensor network.



* For a square lattice, it is an approximation, and energy is not variational!

Contents

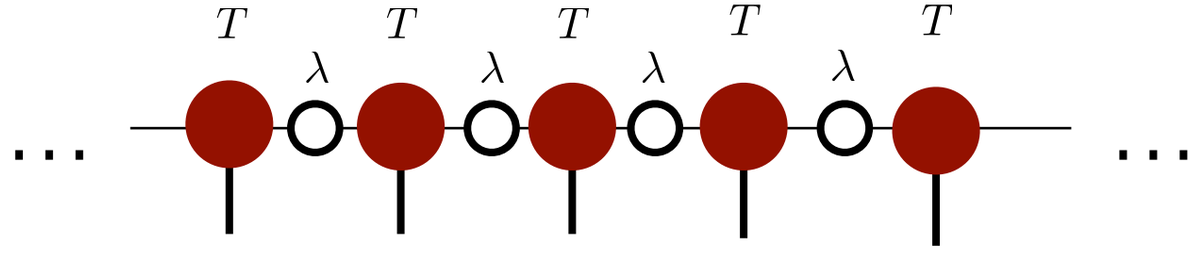
- Basics of tensor network methods: part 1
  - Area law
  - MPS and TEBD algorithm
  - Hands-on 1: Real-time evolution by TEBD
- Basics of tensor network methods: part 2
  - TPS (PEPS) and simple update
  - Hands-on 2: Real-time evolution by TEBD for MPS and PEPS in 2d
- Basics of tensor network methods: part 3
  - Infinite tensor networks
  - Hands-on 2: Real-time evolution by iMPS and iPEPS

Contents

- Basics of tensor network methods: part 1
  - Area law
  - MPS and TEBD algorithm
  - Hands-on 1: Real-time evolution by TEBD
- Basics of tensor network methods: part 2
  - TPS (PEPS) and simple update
  - Hands-on 2: Real-time evolution by TEBD for MPS and PEPS in 2d
- Basics of tensor network methods: part 3
  - Infinite tensor networks
  - Hands-on 2: Real-time evolution by iMPS and iPEPS

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!

*By absorbing λ, we can also construct an iMPS without the bond weights.

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

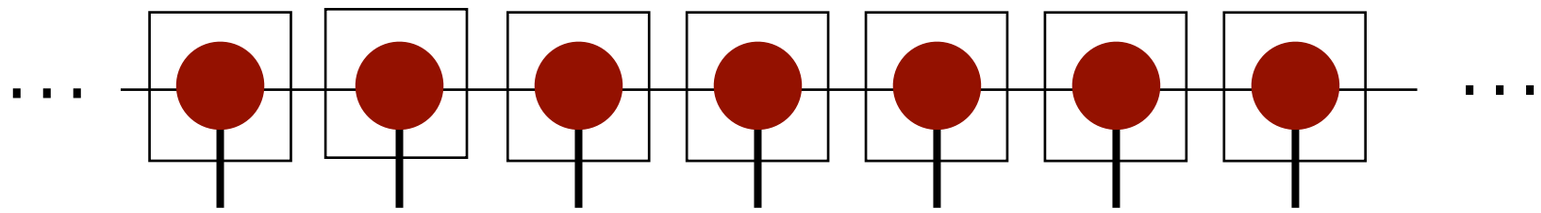
If the EE increases as increase the system size,
we may need infinitely large χ for infinite system.

(In practice, we can obtain a reasonable approximation with finite χ.)

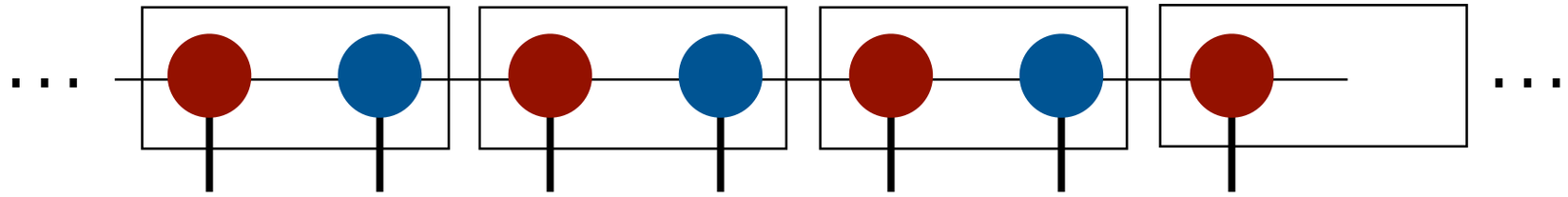
MPS for infinite chains: unit cell

We can also consider a different periodicity depending on the target state.

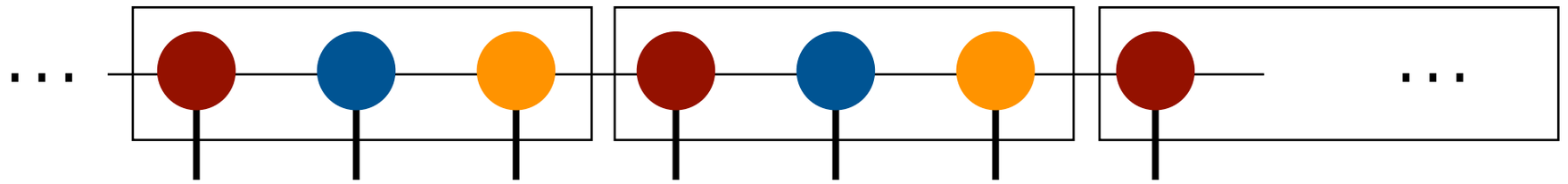
1-site unit cell

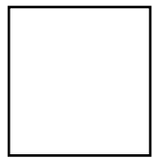


2-site unit cell



3-site unit cell





: unit cell

* Note that iMPS is different from a finite system with periodic boundary condition.

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))

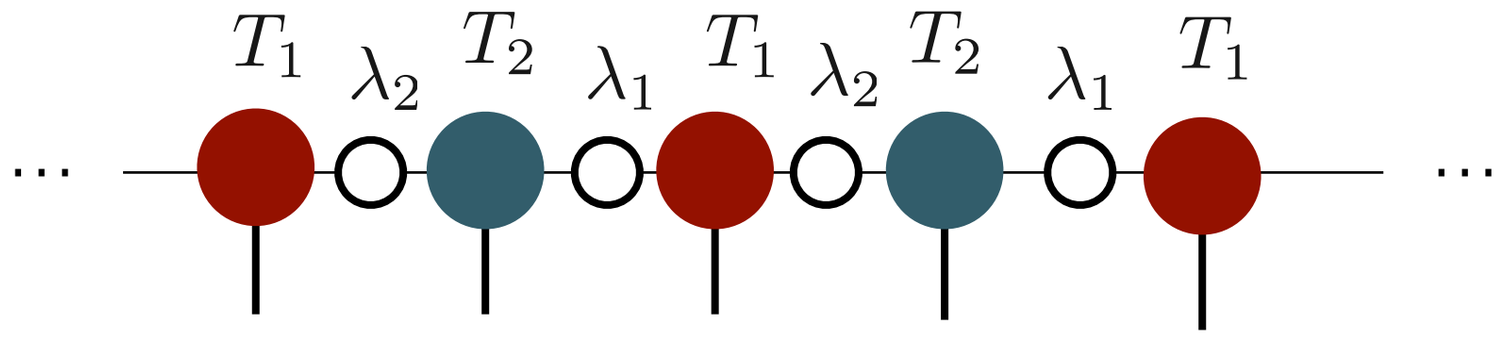
Sequentially apply ITE operators → $O(N)$ SVD for each step

Infinite system: iTEBD

Due to the translational invariance, all SVD are equivalent. → $O(1)$ SVD for each step

*Note

Because of SVD in (i)TEBD algorithm, we need at least two independent tensors even in translationally invariant system

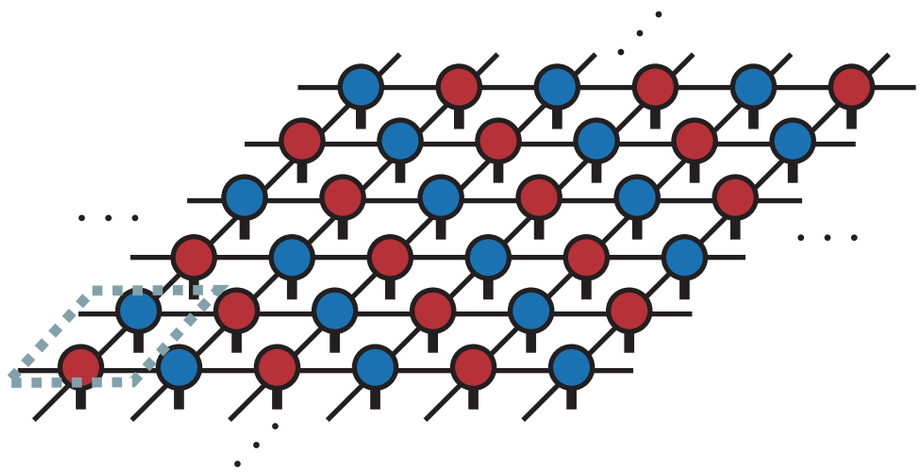


Tensor product states for infinite system: iTPS/iPEPS

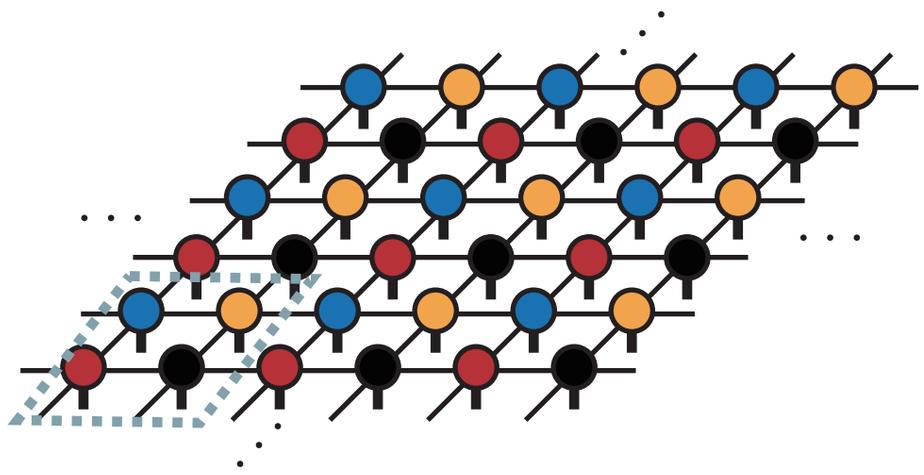
When the state has a translational invariance: $\underline{T}|\Psi\rangle = |\Psi\rangle$

translation — No phase factor

Quantum state for the infinite system can be represented
by repeating the identical tensors periodically.
→We can represent infinite system by a finite degrees of freedom.



2-site unit cell

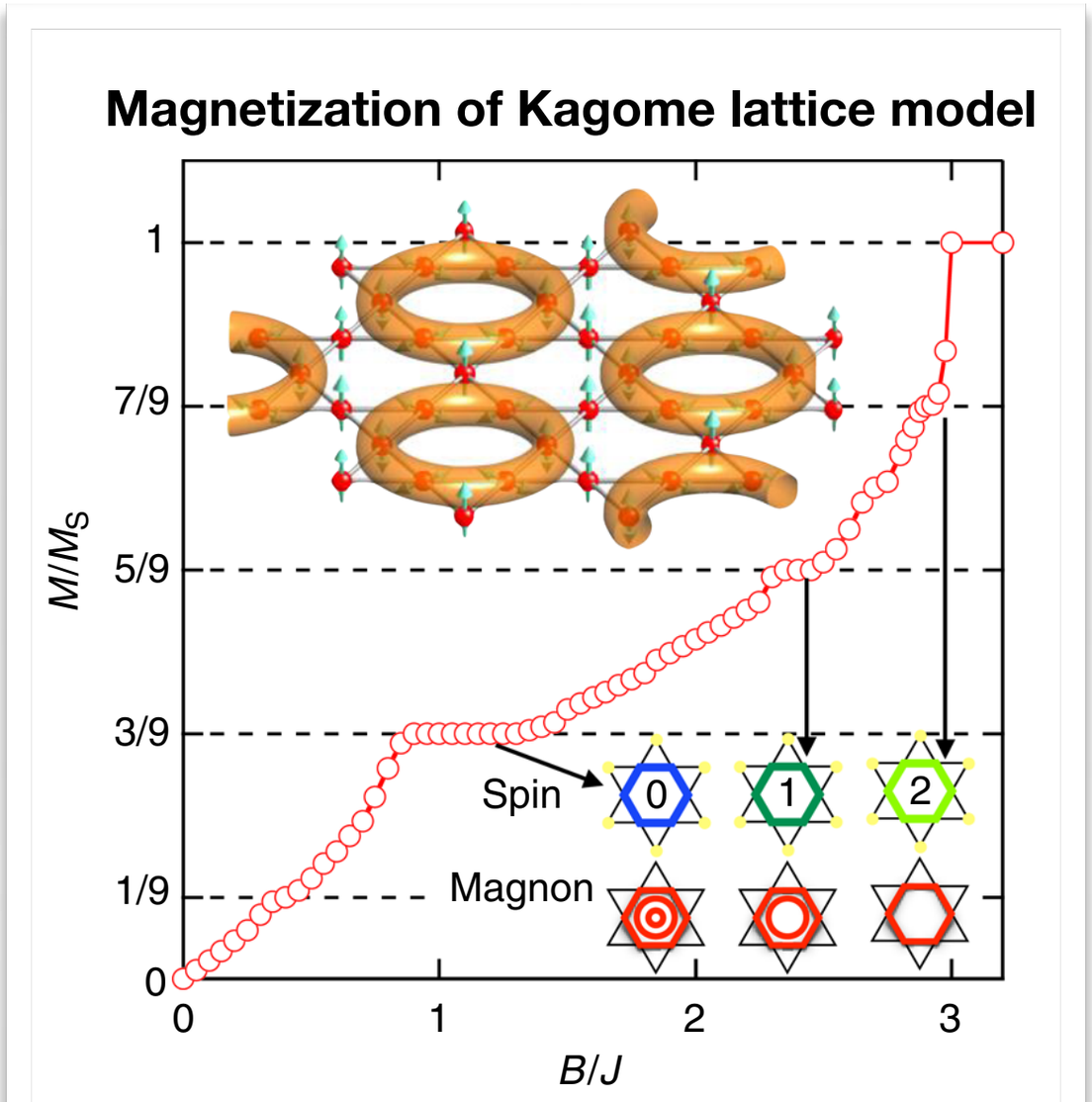


4-site unit cell

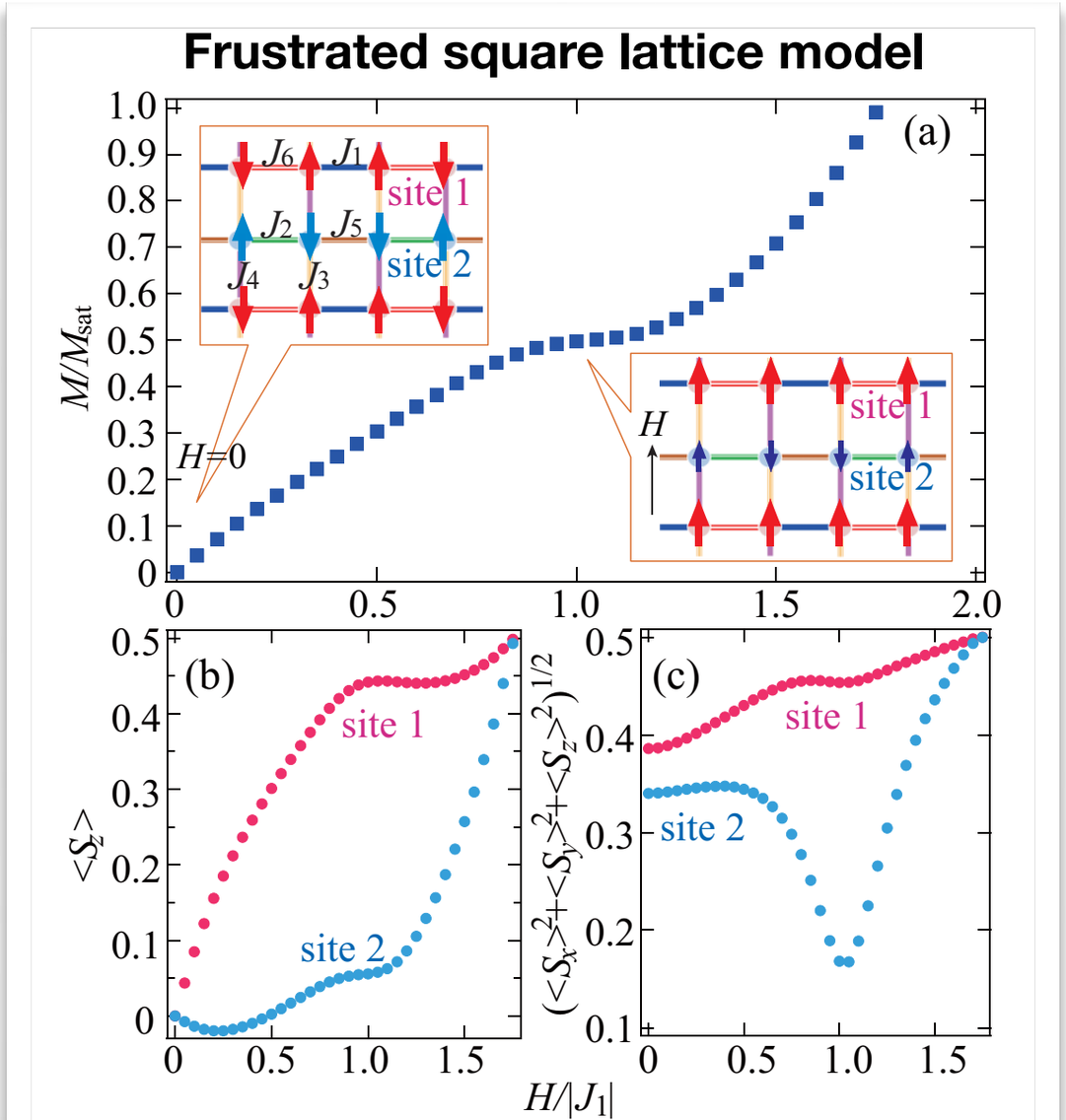
* A lot of interesting problems in quantum spin systems
satisfy such translational symmetry.

Application to quantum many-body systems

Examples: Frustrated spin systems (We can not apply QMC due to the sing problem.)



R. Okuma, D. Nakamura, T. Okubo et al,
Nat. Commun. **10**, 1229 (2019).



H. Yamaguchi, Y. Sasaki, T. Okubo,
Phys. Rev. B **98**, 094402 (2018).

Tensor Netwok Solver (TeNeS)

Y. Motoyama, T. Okubo, K. Yoshimi, et al., Comput. Phys. Commun. **279**, 108437 (2022).
Y. Motoyama, T. Okubo, K. Yoshimi, et al., Comput. Phys. Commun. **315**, 109692 (2025)

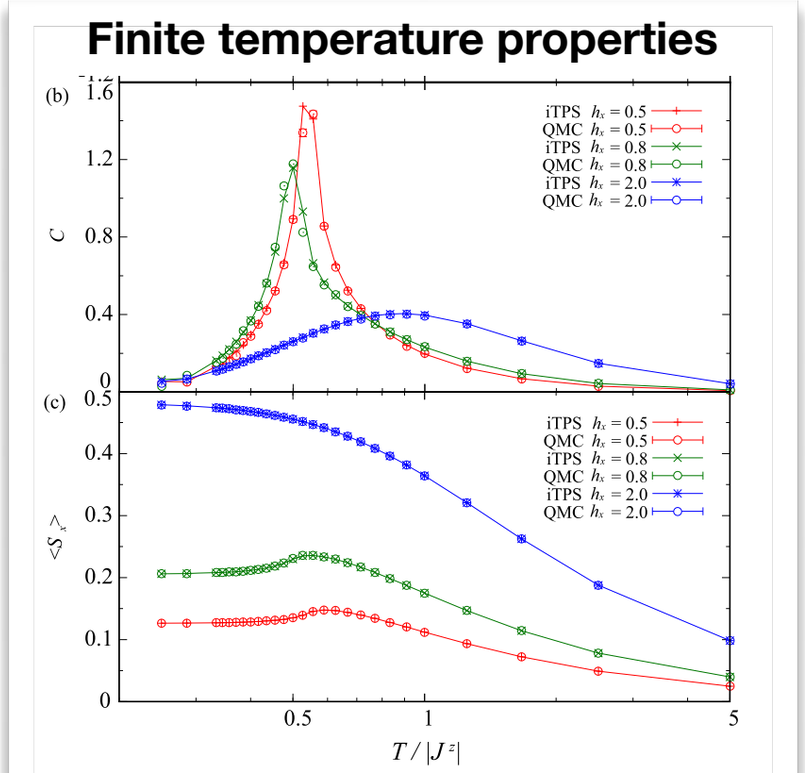
TeNeS

https://github.com/issp-center-dev/TeNeS

Ground state calculation of 2d quantum models by iTPS

(TeNeS also supports real-time and finite temperature simulations)

- ☑ Tensor optimization by the imaginary time evolution
- ☑ Massively parallelized by MPI/OpenMP
  - ☑ parallelization of tensor operations by mptensor (Morita)
- ☑ Easy calculations for quantum spins or bosons
- ☑ Easy access to standard 2d lattices
  - ☑ In principle, we can use any 2d lattices



Developpers:

- Tsuyoshi Okubo: Core algorithms
- Satoshi Morita: Related libraries and tools
- Yuichi Motoyama: Main programs
- Kazuyoshi Yoshimi: User tests, tutorials, and project management
- Tatsumi Aoyama: User tests and tutorials
- Takeo Kato: User tests and tutorials
- Naoki Kawashima: Original project leader