計算科学における情報圧縮

Information Compression in Computational Science

2020.12.10

#11:行列積表現の応用

Application of MPS

理学系研究科　物理学専攻　大久保　毅

Department of Physics, **Tsuyoshi Okubo**

Outline

- Matrix product states (cont.)
    - Matrix product states (MPS)
    - Canonical form
    - infinite MPS (quick introduction only)
- Application to Eigenvalue problem
  (Ground state of quantum many-body systems)
    - Variational algorithm
- Application to time evolution of quantum system
    - TEBD algorithm
- ~~Application to data science~~ (Next week)

Matrix product states（行列積状態）

(Tensor train decomposition)

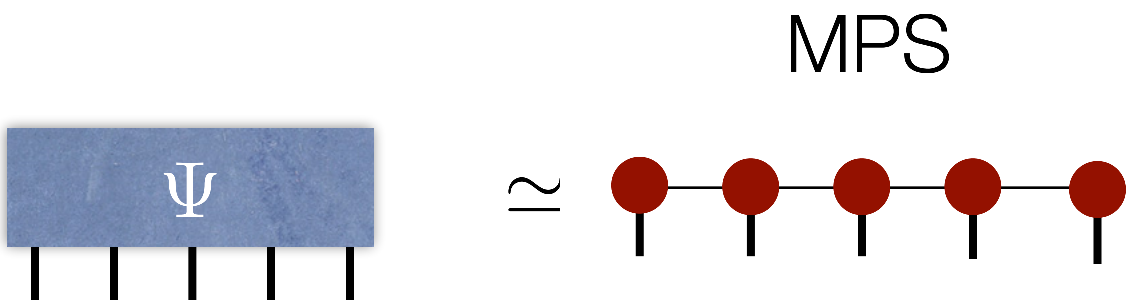
Matrix product state (MPS)

Good reviews:

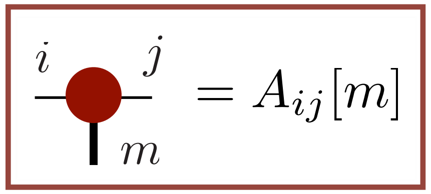
(U. Schollwöck, Annals. of Physics **326**, 96 (2011))
(R. Orús, Annals. of Physics **349**, 117 (2014))

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

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



$A[i]$: Matrix for state i



Note:

- MPS is called "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)

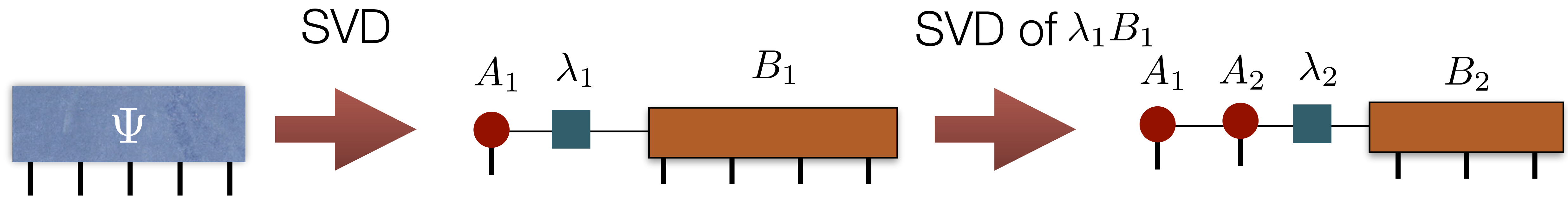
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

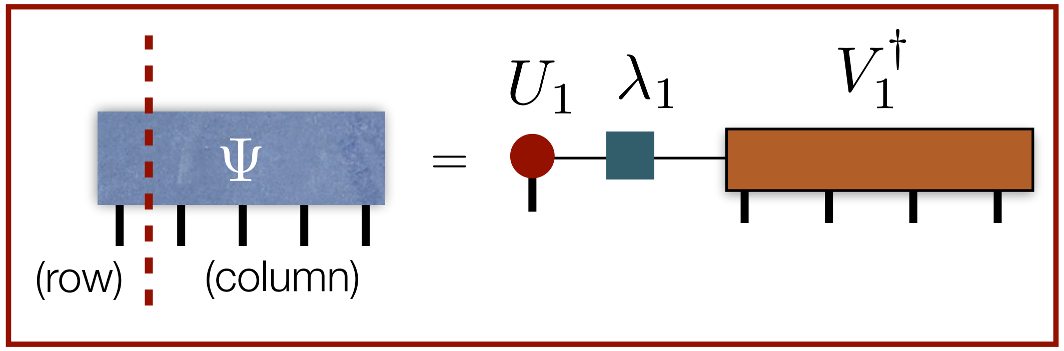
$$\Psi_{i_1 i_2 \ldots i_N} = \phi_1[i_1]\phi_2[i_2]\cdots\phi_N[i_N]$$

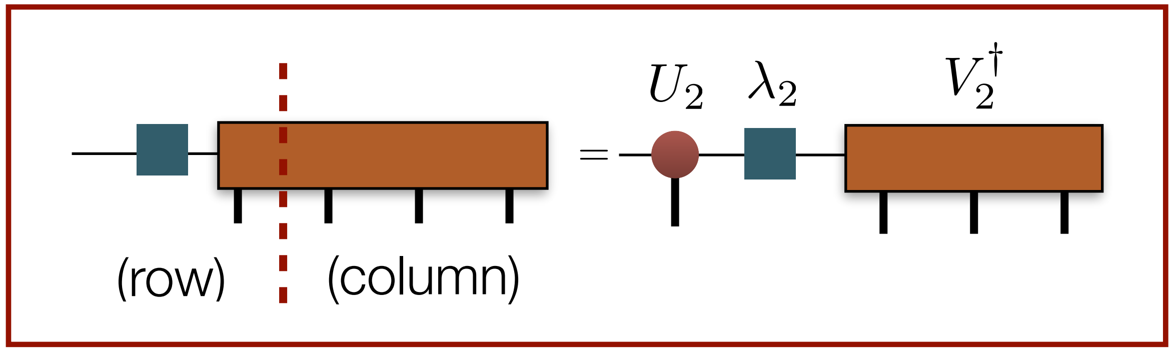
$$\phi_n[i] \equiv \langle i|\phi_i\rangle$$

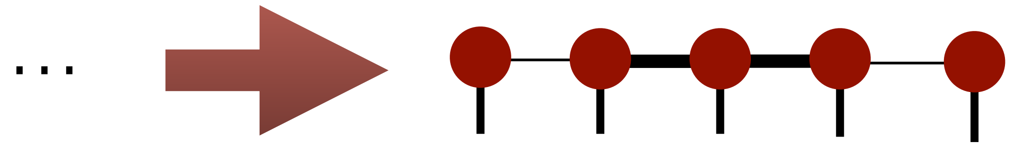
Matrix product state without approximation

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







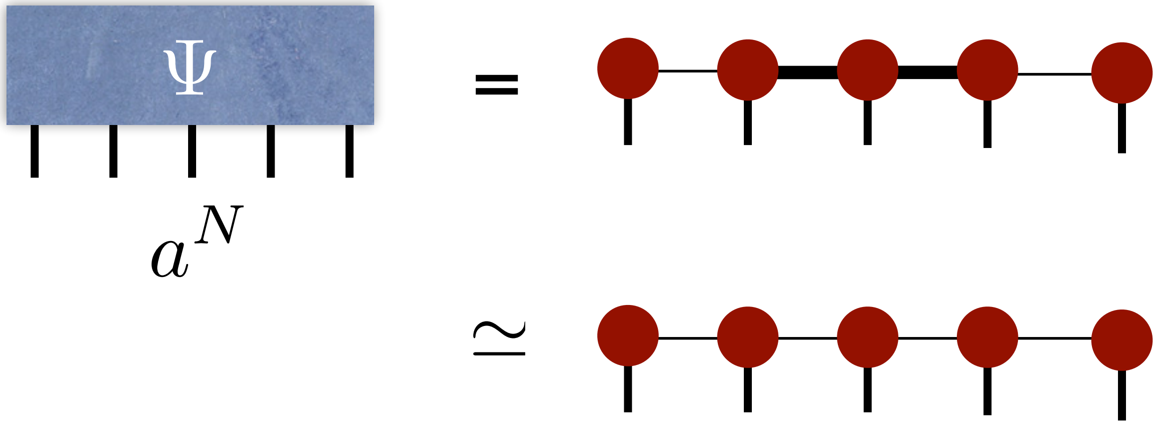


In this construction, the sizes of matrices depend on the position.

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

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

Matrix product state: Low rank approximation



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

a^N

Constant bond dimension = χ

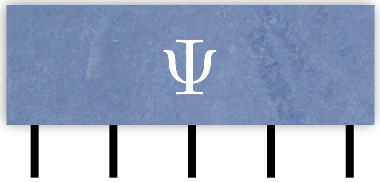
$\propto Na\chi^2$

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

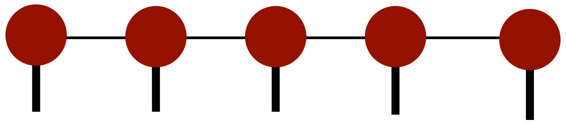
➡ MPS is good for gapped 1d systems.

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

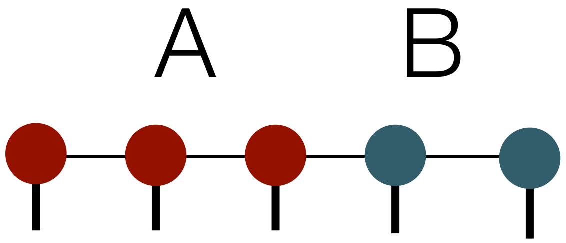


$\simeq$

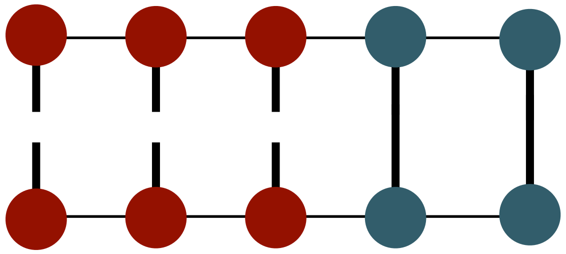


$\equiv |\tilde{\Psi}\rangle$:MPS with χ

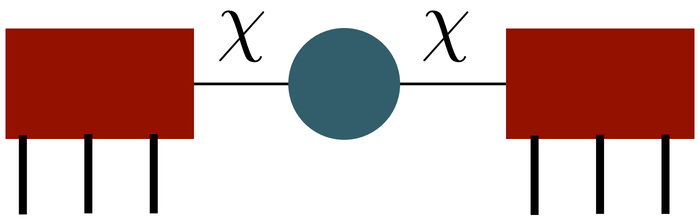
Reduced density matrix of region A:



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



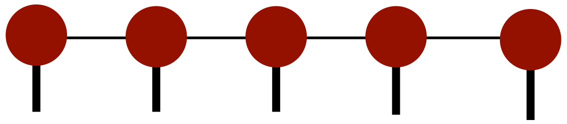
★ Structure of ρ_A:



$$\boxed{\begin{aligned} &\mathrm{rank}\ \rho_A \leq \chi \\ &S_A = -\mathrm{Tr}\ \rho_A \log \rho_A \leq \log \chi \end{aligned}}$$

Required bond dimension in MPS representation

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



The upper bound is independent of the "length".

length of MPS ⇔ size of the problem

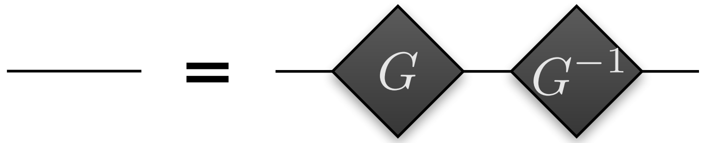
N $\qquad$ a^N

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

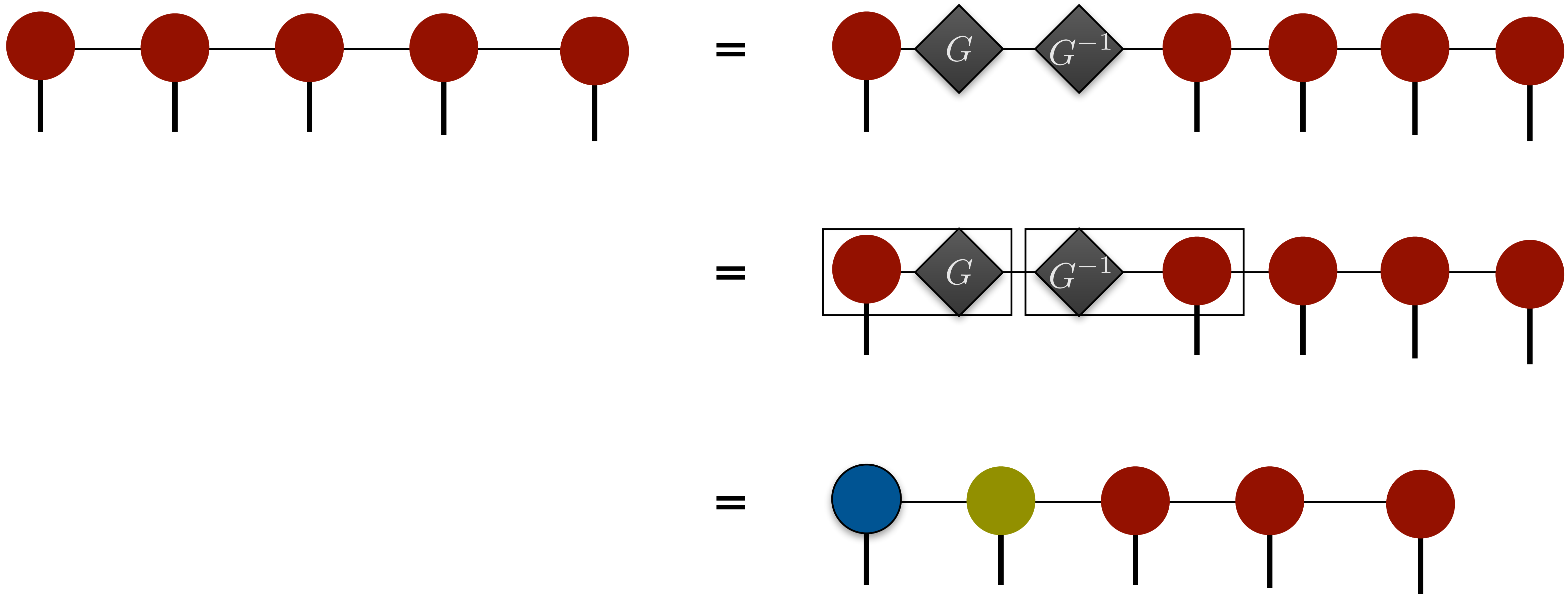
$(\alpha \leq 1)$

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

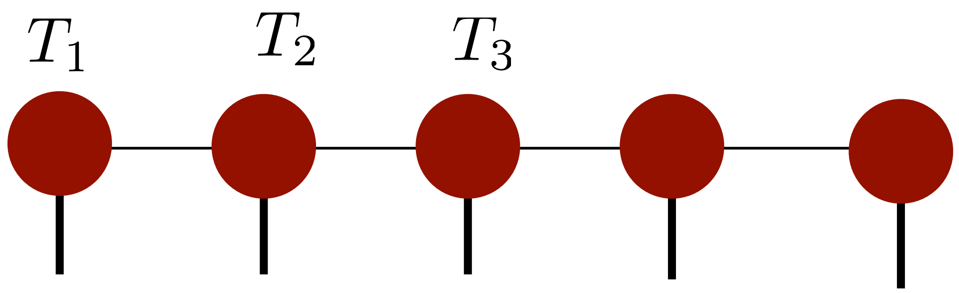


Canonical forms: Left and Right canonical forms

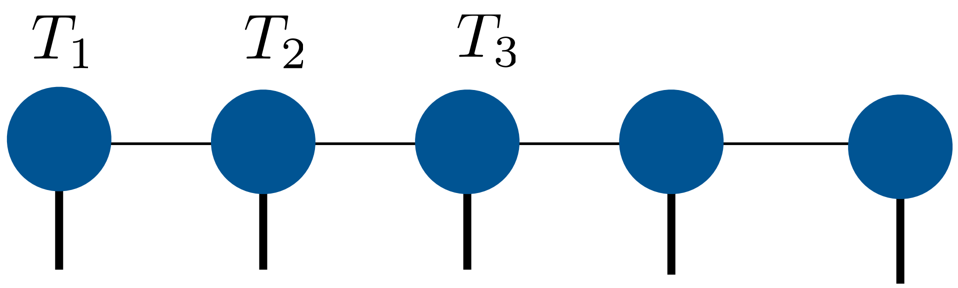
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

"canonical" forms of MPS

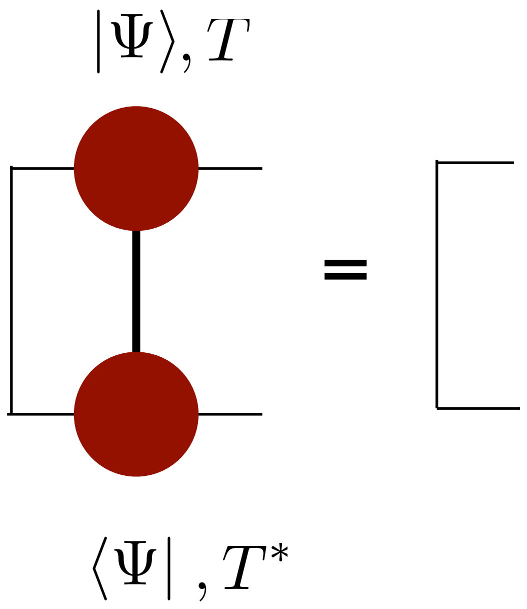
Left canonical form:

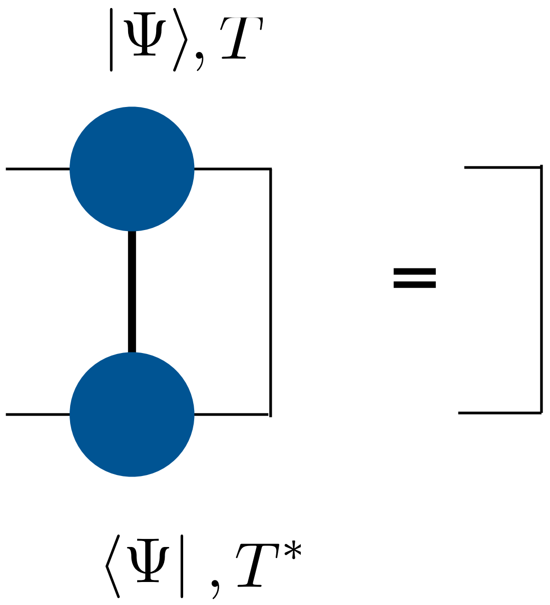


Right canonical form:



Satisfies (at least) left or canonical condition:



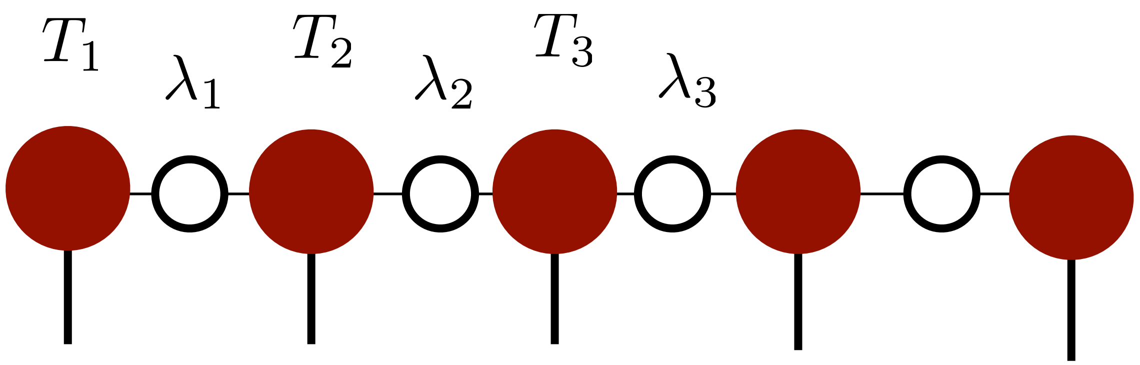


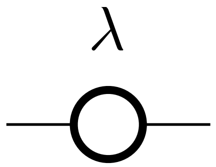
Gauge fix: Canonical form of MPS

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

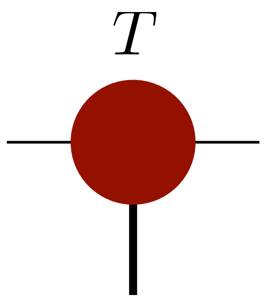
Another canonical form of MPS: (Vidal canonical form)

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



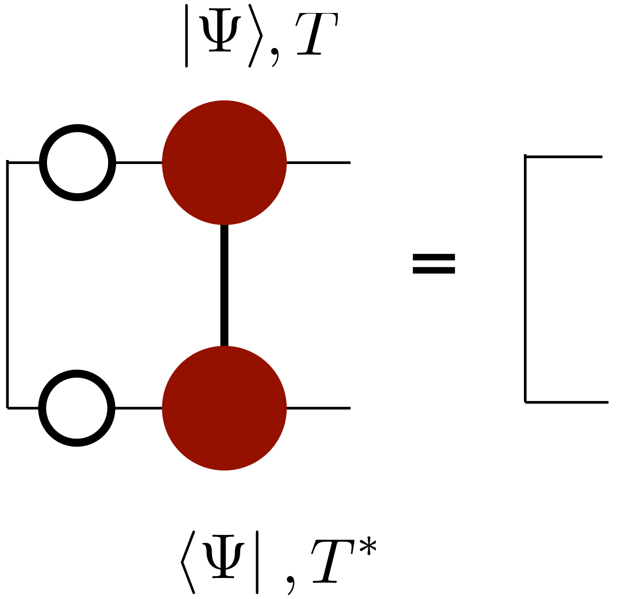


:Diagonal matrix corresponding to Schmidt coefficient

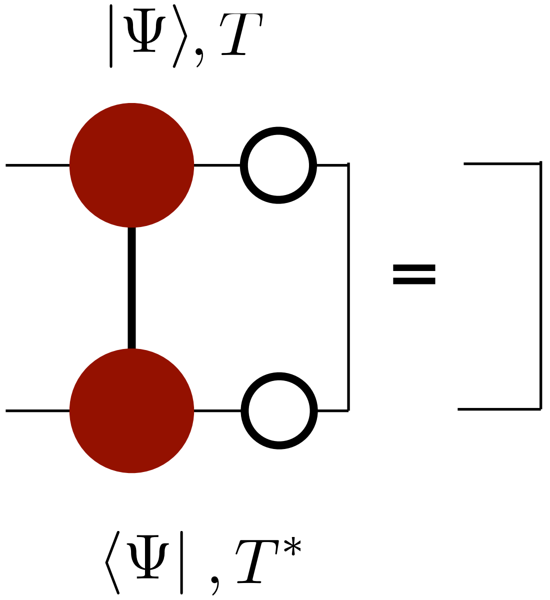


:Virtual indices corresponding to Schmidt basis

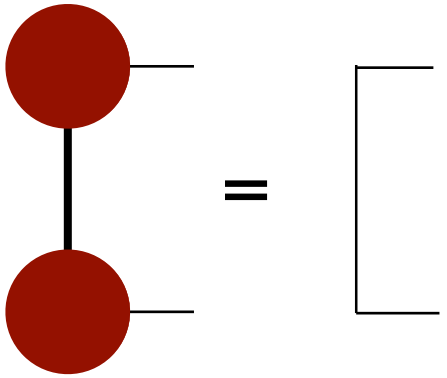
Left canonical condition:



Right canonical condition:



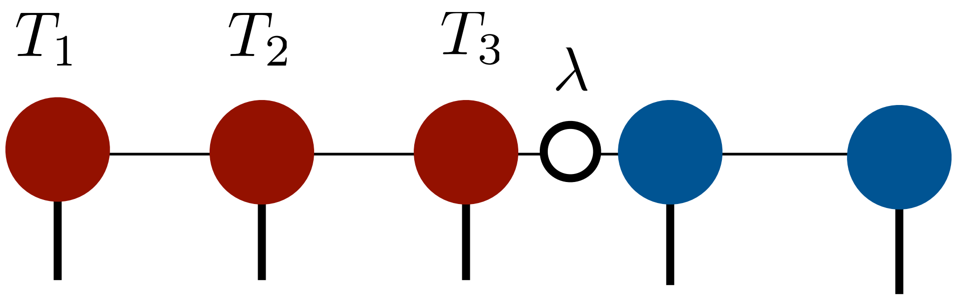
(Boundary)



Canonical forms: Mixed canonical forms

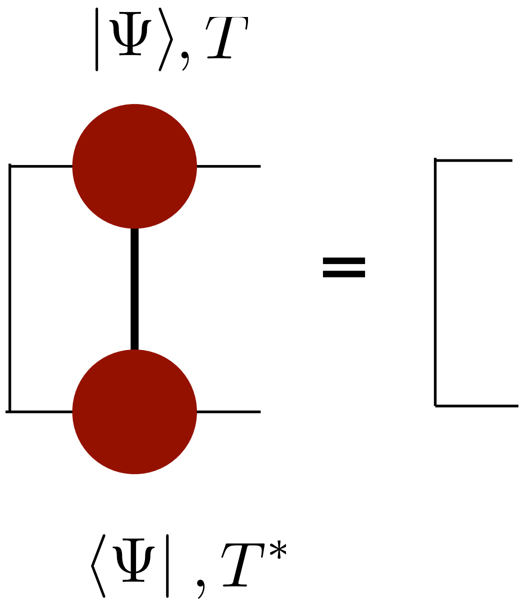
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Mixed canonical form:

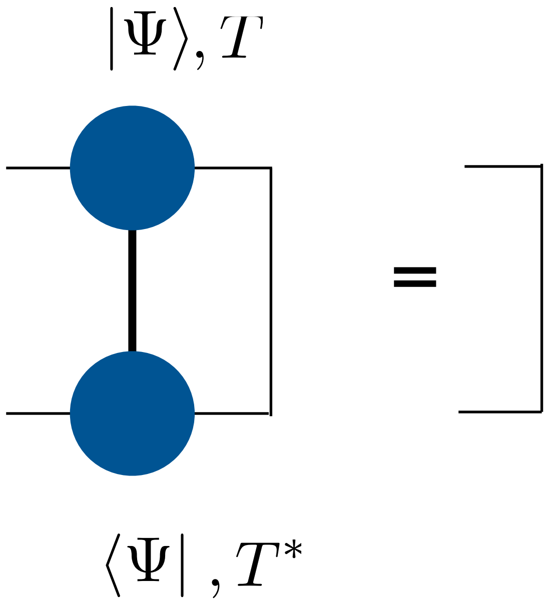


λ is identical with the Schmidt coefficient.

Left canonical condition:



Right canonical condition:

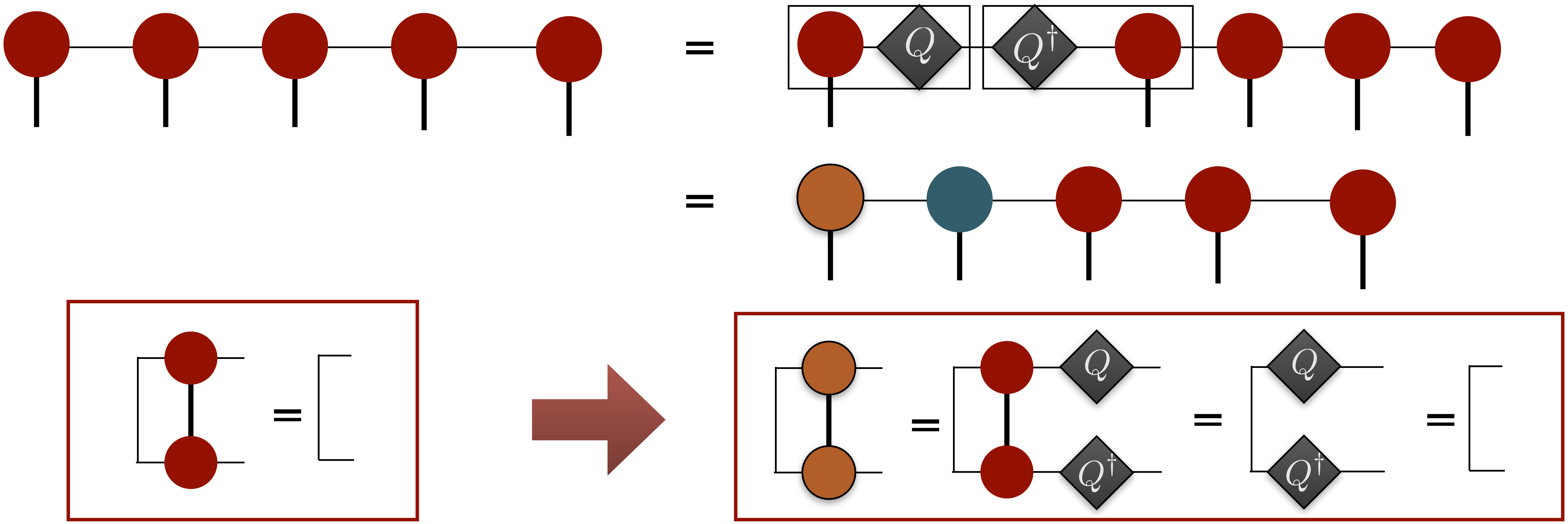


Canonical forms: Note

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

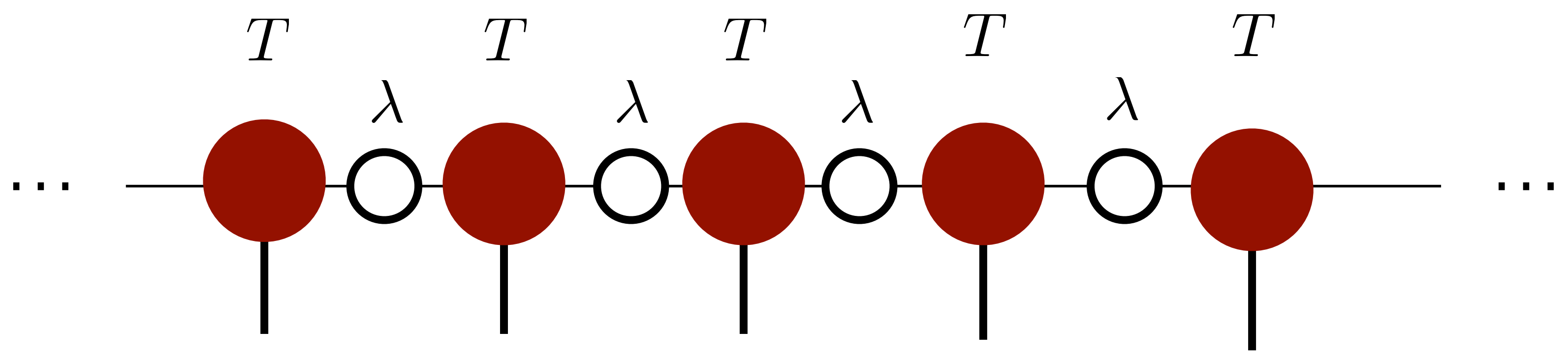
- Left, right and mixed canonical form is "not unique". Under general unitary transformation to virtual indices, it remains to satisfy the canonical condition

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



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.

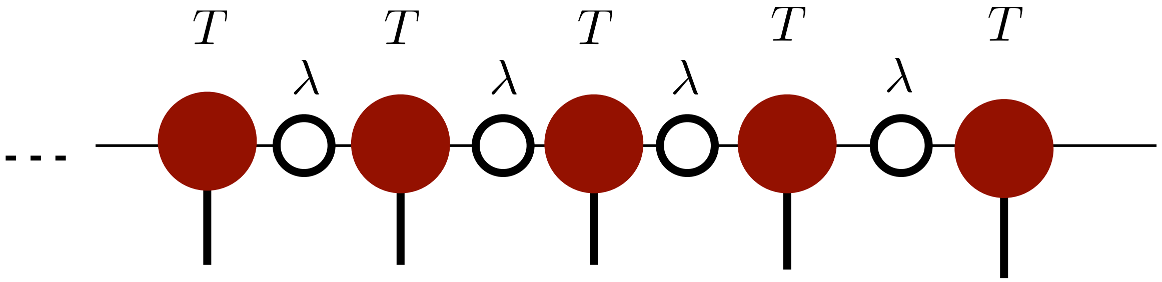
(In practice, we can obtain a reasonable approximation with finite χ.)

Example of iMPS: AKLT state (will be skipped)

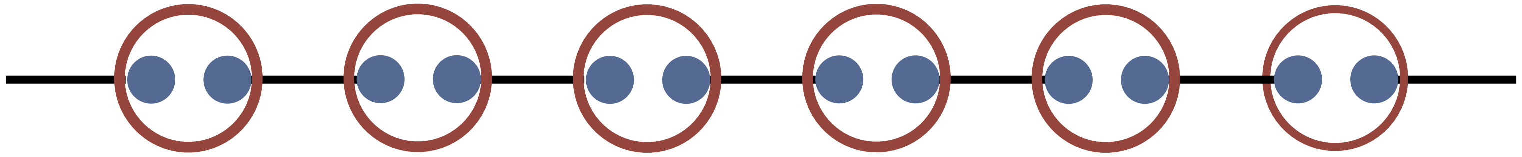
S=1 Affleck-Kennedy-Lieb-Tasaki (AKLT) Hamiltonian:

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

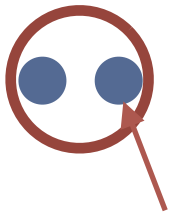
$$(J > 0)$$



The ground state of AKLT model:



S=1 spin:



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

Spin singlet

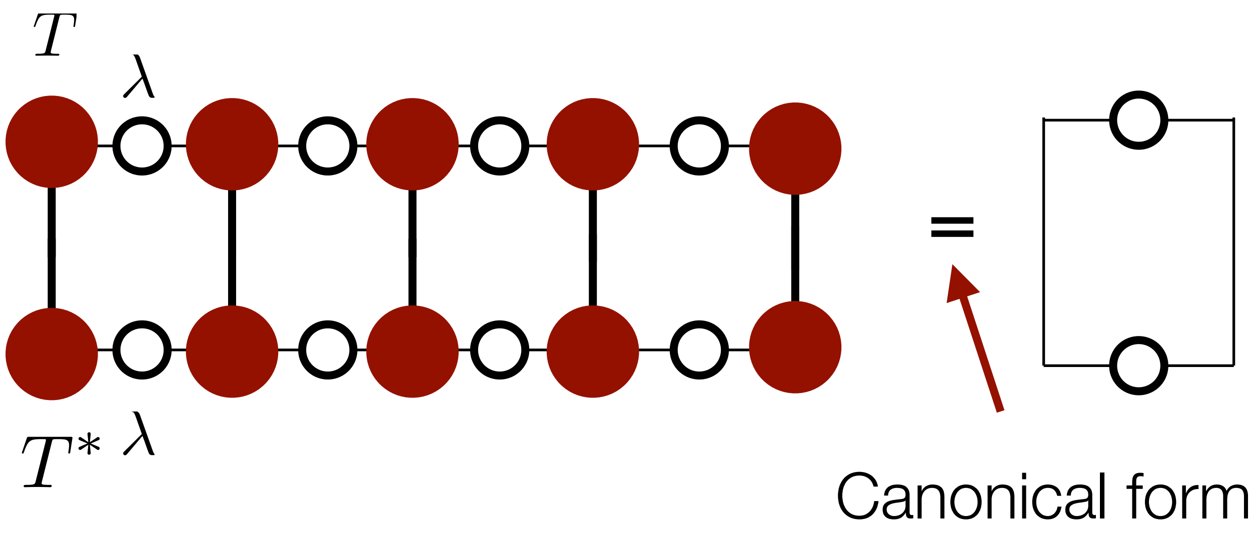
$$T[S_z = 1] = \sqrt{\frac{4}{3}}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

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

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

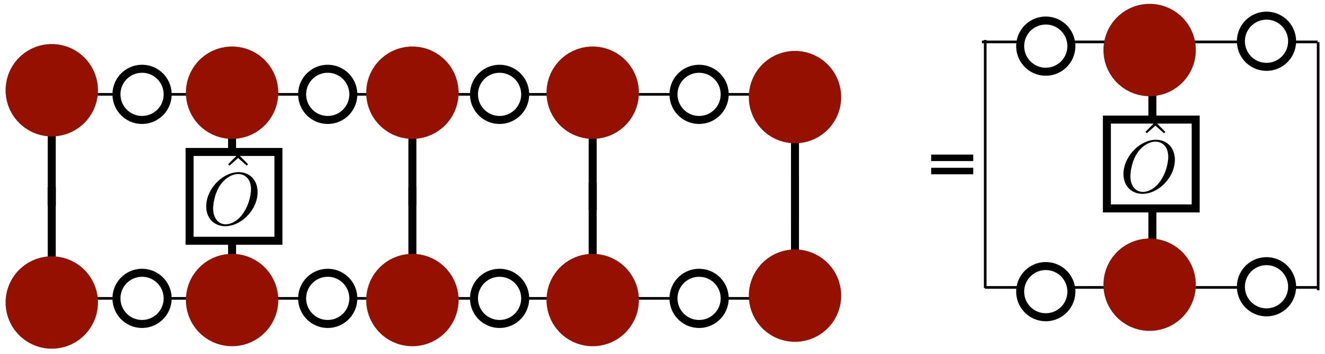
Calculation of expectation value (will be skipped)

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



$$= \sum_i \lambda_i^2 = 1$$

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



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

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

Exercise 2: Make MPS and approximate it

2: Make exact MPS and approximate it by truncating singular values

Try MPS approximation for a random vector, GS of spin model, or a picture image.
Let's see how the approximation efficiency depends on the bond dimensions and vectors.

Sample code: Ex2-1, Ex2-2, Ex2-3.ipynb, or .py

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

These codes correspond to random vector, spin model and picture image, respectively.

I recommend *.ipynb because it contains an appendix part.

*If you run them at Goole Colab, please upload MPS.py in addition to the *.ipynb.

*In the case of Ex2-2 you also need ED.py.

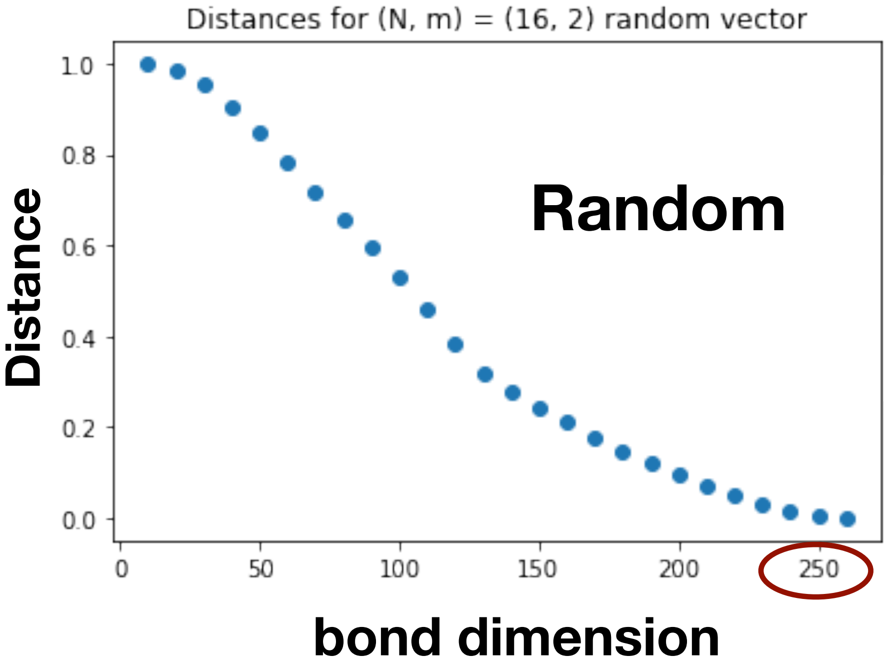
*In the case of Ex2-3 you also need picture file.

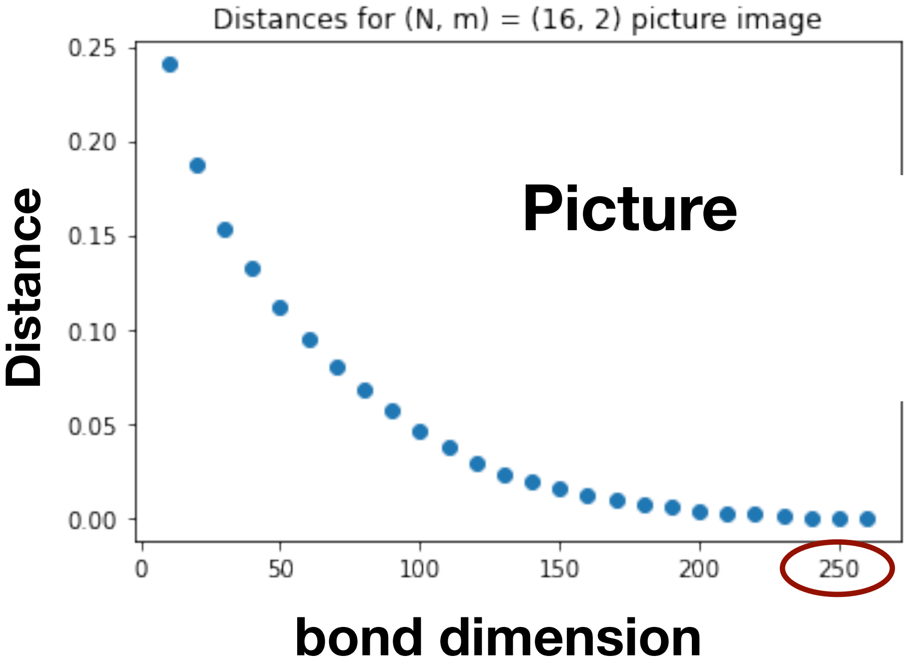
This exercise is almost same with the Report1-2.

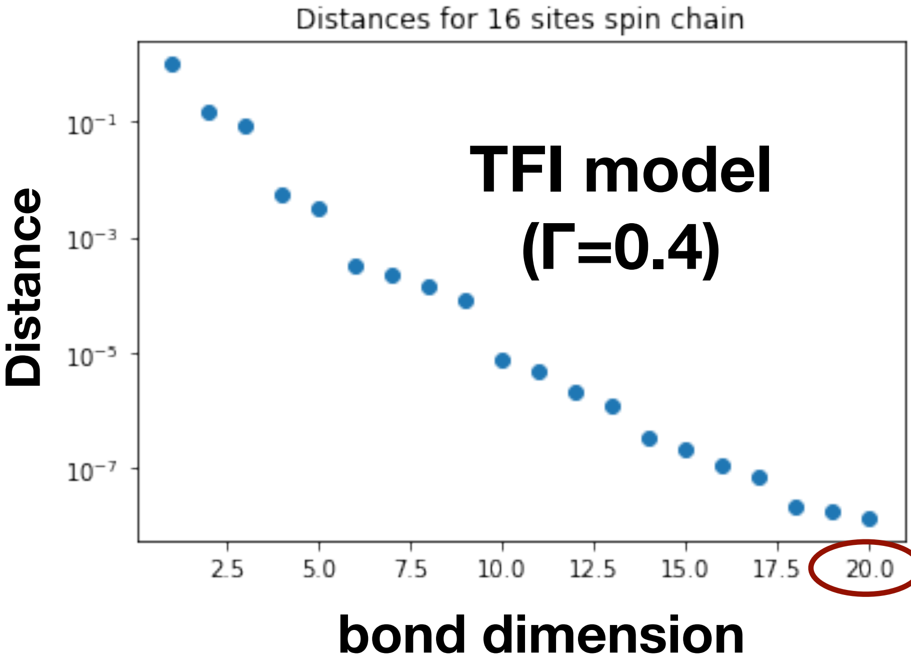
Exercise 2: Make MPS and approximate it

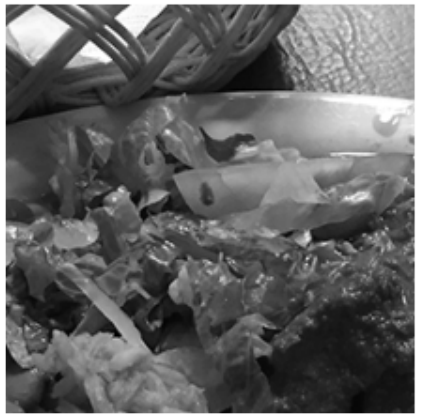
2^{16} dimensional vectors (=16-leg tensors)

Distance between the original and approximated vectors: $||\vec{v}_{ex} - \vec{v}_{ap}||$









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

Application to eigenvalue problem

Calculation of minimum (or maximum) eigenvalue

Target vector space:

Exponentially large Hilbert space

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

$+$

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

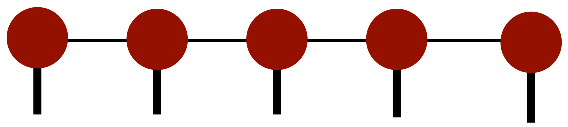
$$\mathcal{H}\vec{v}_0 = E_0\vec{v}_0$$

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

Variational calculation using MPS:

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

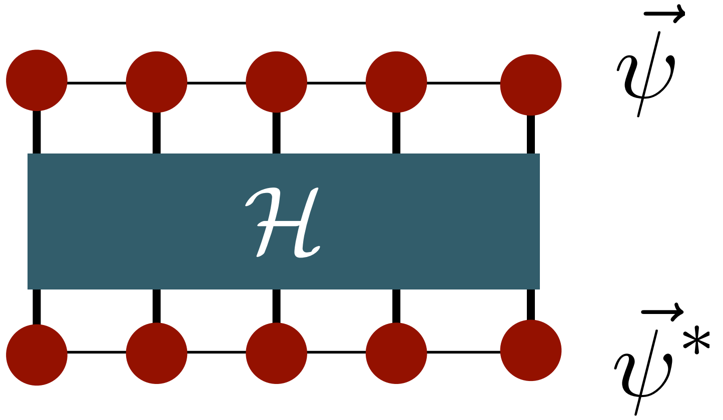
Find the MPS which minimizes F
by optimizing matrices in MPS.

$\vec{\psi} =$ 

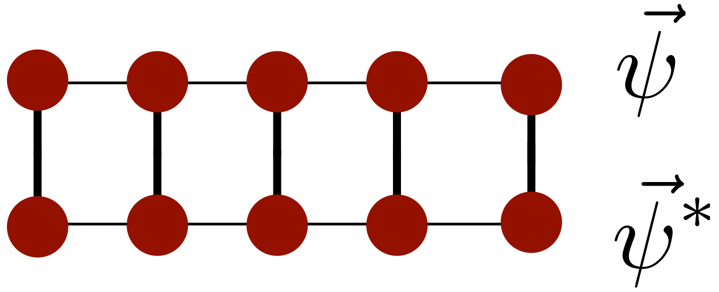
Problem in graphical representation

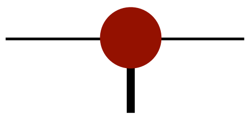
Cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

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



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



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 :

Minimize

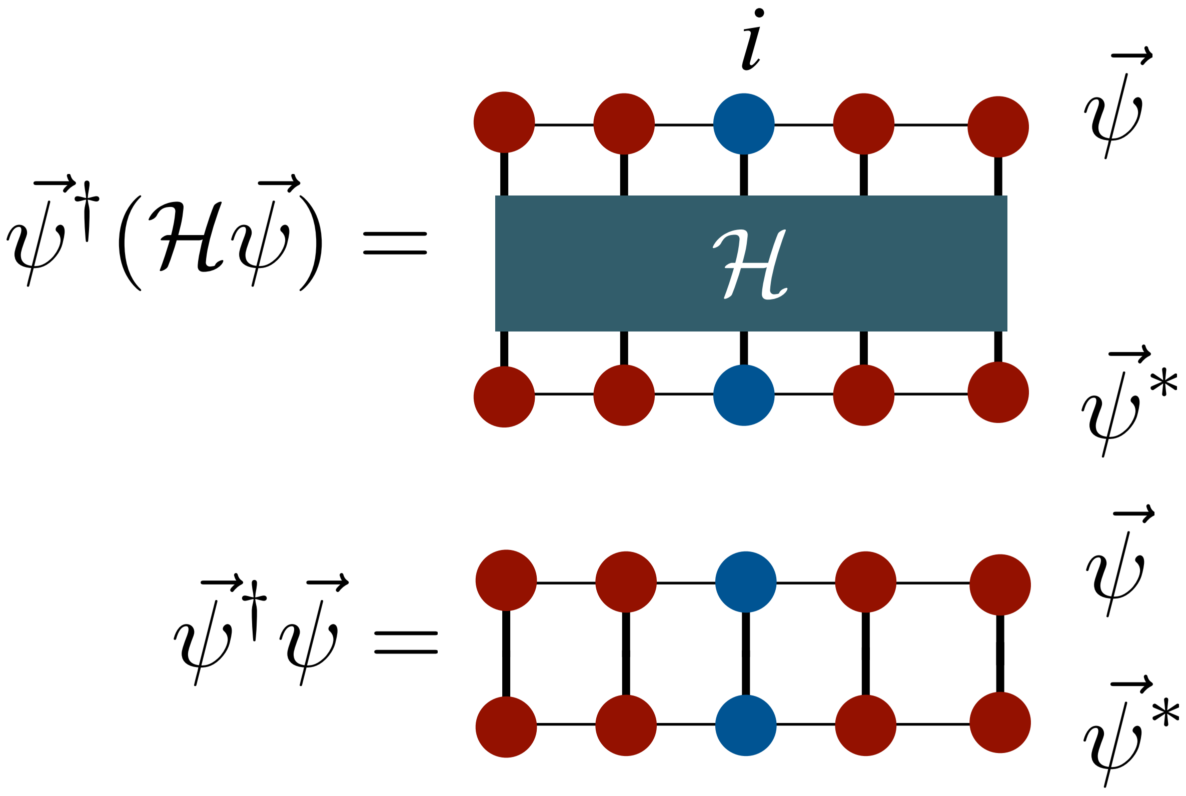
$$F = \frac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}} = \frac{A_i^\dagger(\tilde{\mathcal{H}}_i A_i)}{A_i^\dagger(\tilde{N}_i A_i)}$$

Solve generalized eigenvalue problem

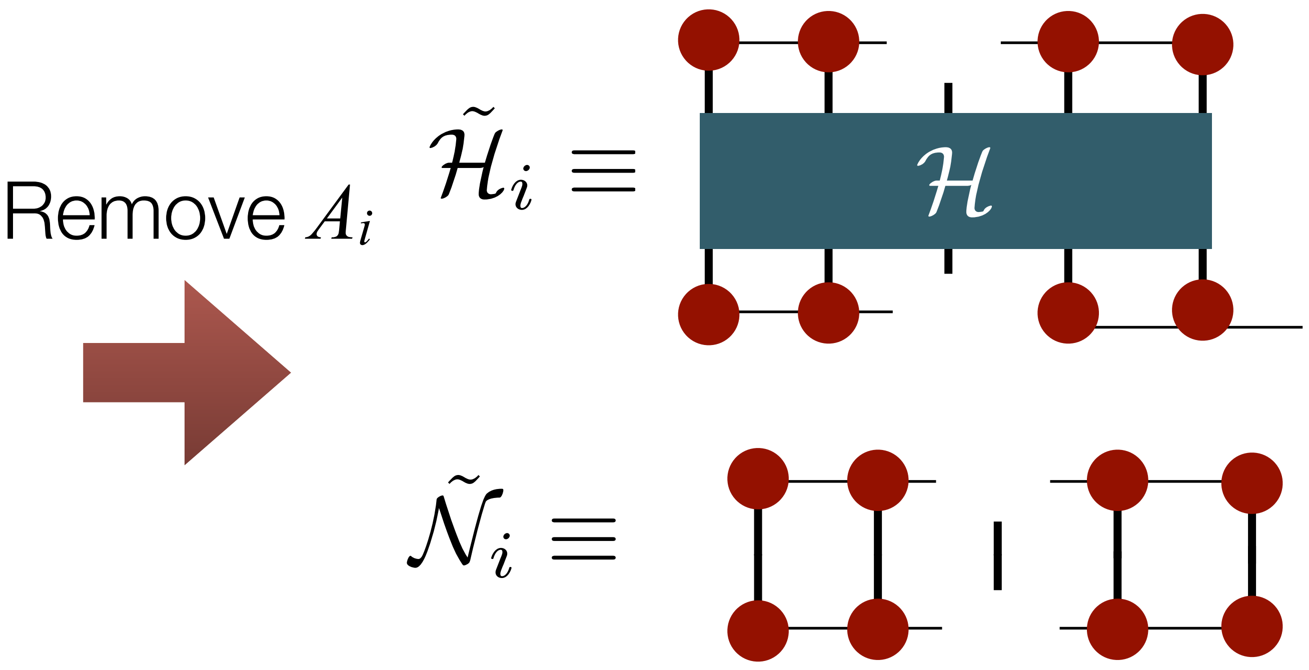
$$\tilde{\mathcal{H}}_i A_i = \epsilon \tilde{\mathcal{N}}_i A_i$$

(Find the lowest eigenstate)

Notice: matrix size = $a\chi^2 \times a\chi^2$



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



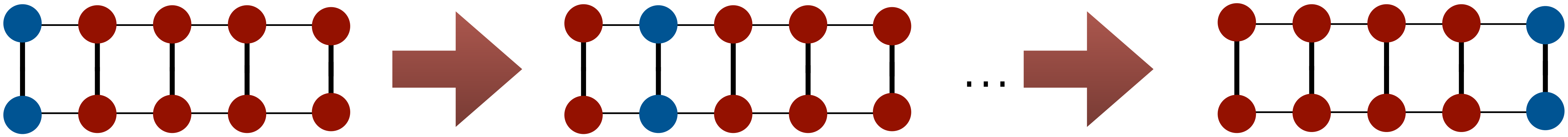
Remove A_i
$\tilde{\mathcal{H}}_i \equiv$
$\mathcal{H}$
$\tilde{\mathcal{N}}_i \equiv$

Notice: If we impose canonical form, $\tilde{\mathcal{N}}$ becomes a simple identity matrix.

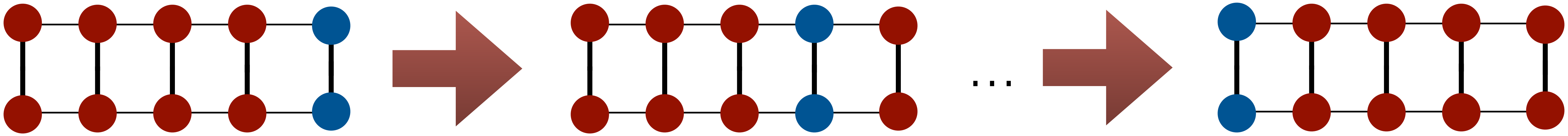
Iterative optimization

(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



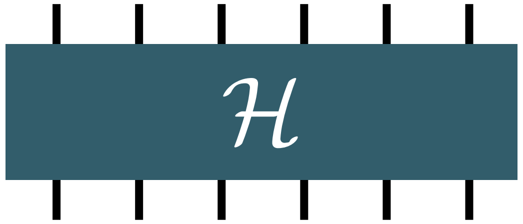
Repeat sweeping until convergence.

Compact representation of an operator

Notice!

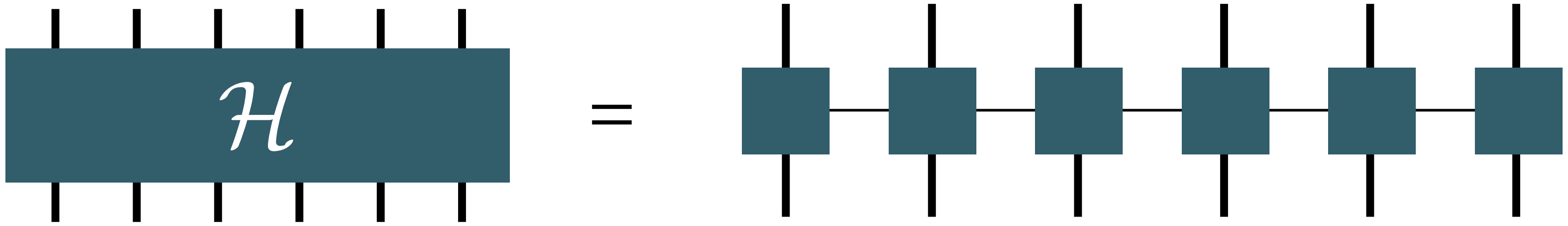
We can conduct this algorithm when we can represent the matrix efficiently.

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.



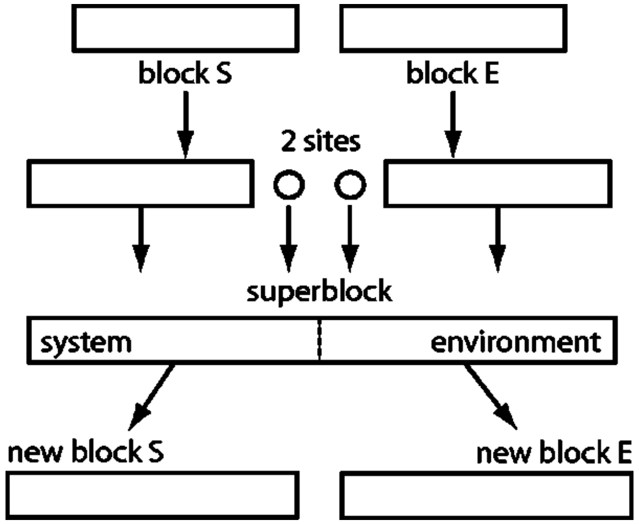
E.g. The Hamiltonian of the Heisenberg model is represented by MPO with bond dimension $\chi = 5$.

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）

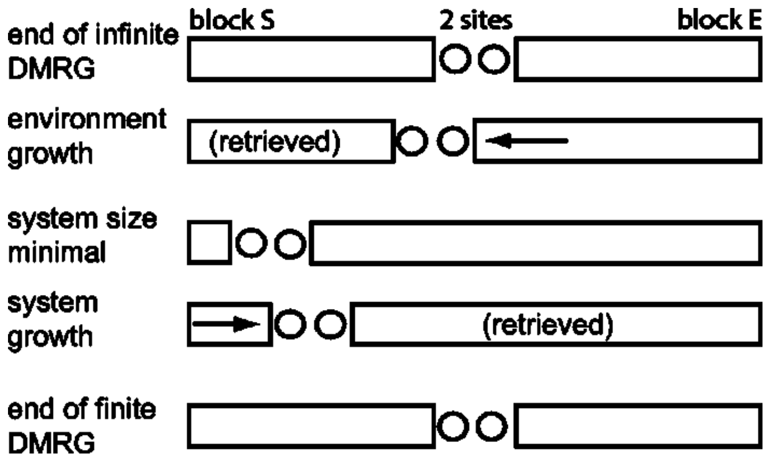
(S. R. White, Phys. Rev. Lett. **69**, 2863 (1992))
(U. Schollwöck, Rev. Mod. Phys. **77,** 259 (2005))
(U. Schollwöck, Annals. of Physics **326**, 96 (2011))

DMRG selects compact basis based on entanglement between "System" and "Environment" blocks.



DMRG is a powerful tool in physics and chemistry

- One-dimensional spin systems
- One-dimensional electron systems
- Small molecules
- Small two-dimensional systems



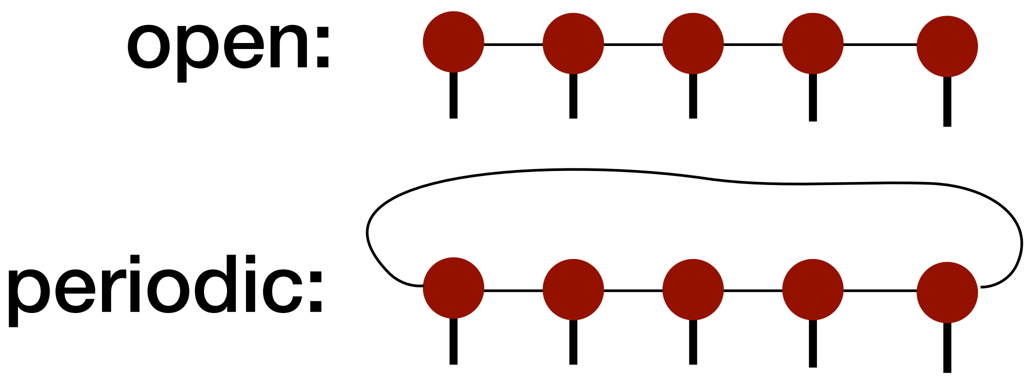
The original DMRG did not use MPS explicitly. But, MPS gives us a theoretical background for why DMRG works well.

Relation to Density Matrix Renormalization Group

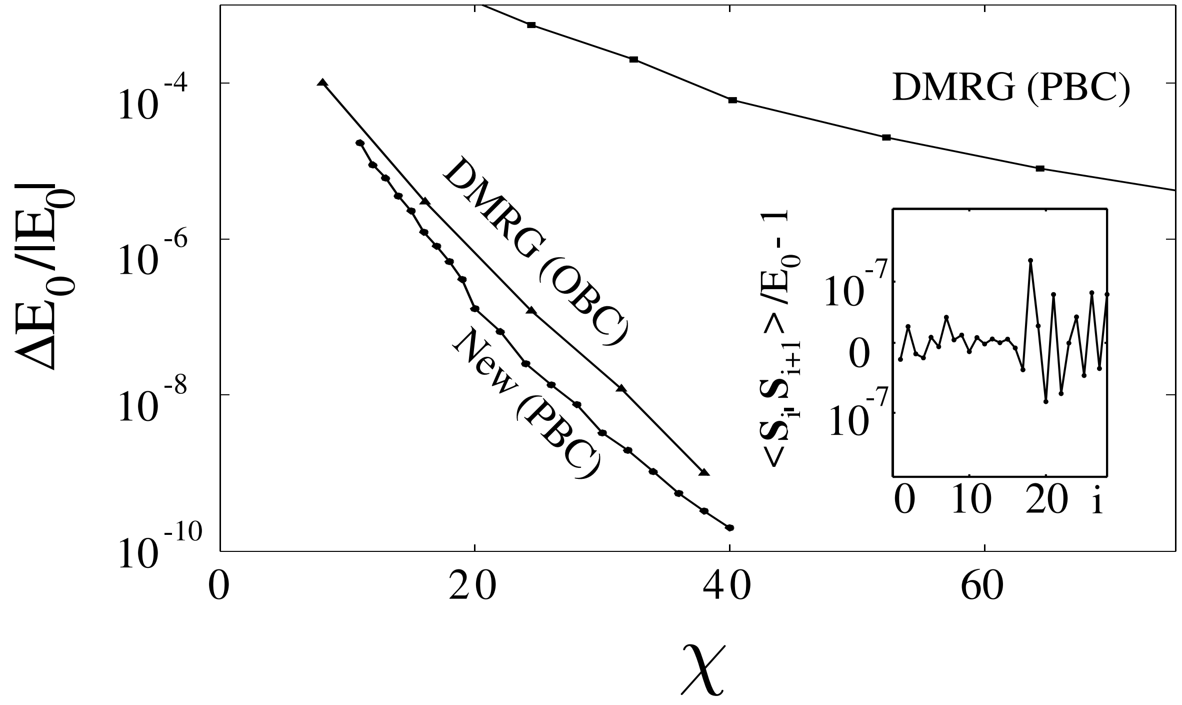
Conventional DMRG algorithm corresponds to variational calculation using open boundary MPS.

(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.



S=1/2 Heisenberg chain, (N=40)



If we use periodic MPS instead of open MPS, we can represent the ground state more efficiently.

Application to time evolutions of quantum system

*Similar algorithm can be used to simulate quantum circuit.

A. McCaskey et al, PLoS ONE **13**, e0206704 (2018)

Related reference:

C. Guo et al, Phys. Rev. Lett. **123**, 190501 (2019)

Time evolution of a quantum system

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

→ Formal solution:

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

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

$$e^{\tau(\mathcal{A}+\mathcal{B})} = e^{\tau\mathcal{A}}e^{\tau\mathcal{B}} + O(\tau^2) \qquad \text{(1st order)}$$

$(\mathcal{A}\mathcal{B} \neq \mathcal{B}\mathcal{A})$

$$= e^{\tau/2\mathcal{A}}e^{\tau\mathcal{B}}e^{\tau/2\mathcal{A}} + O(\tau^3) \qquad \text{(2nd order)}$$

$$= e^{\tau/2\mathcal{B}}e^{\tau\mathcal{A}}e^{\tau/2\mathcal{B}} + O(\tau^3) \qquad \text{(2nd order)}$$

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

$$e^{-it\mathcal{H}/\hbar} = \left(e^{-i\delta\mathcal{H}}\right)^M = \left(\prod_j e^{-i\delta H_j}\right)^M + O(\delta)$$

(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

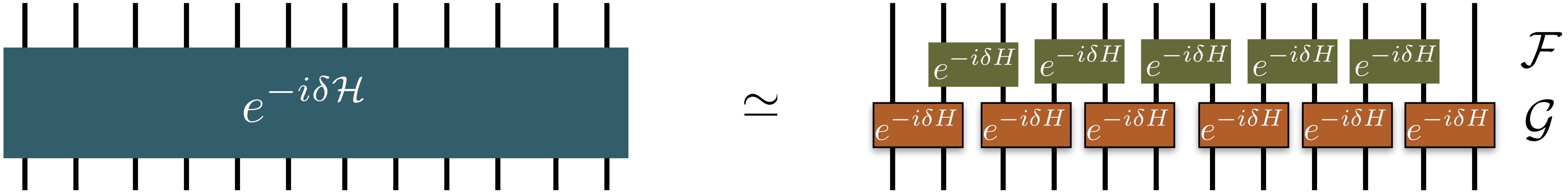
$$\mathcal{H} = \sum_i H_i = \sum_{i \in \mathrm{even}} H_i + \sum_{i \in \mathrm{odd}} H_i$$

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

Suzuki-Trotter decomposition of time evolution operator

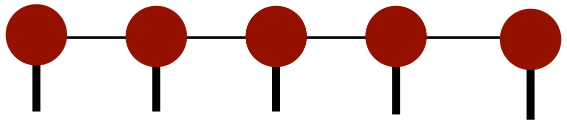
$$e^{-i\delta\mathcal{H}} = e^{-i\delta\mathcal{F}} e^{-i\delta\mathcal{G}} + O(\delta^2)$$

(1st order)

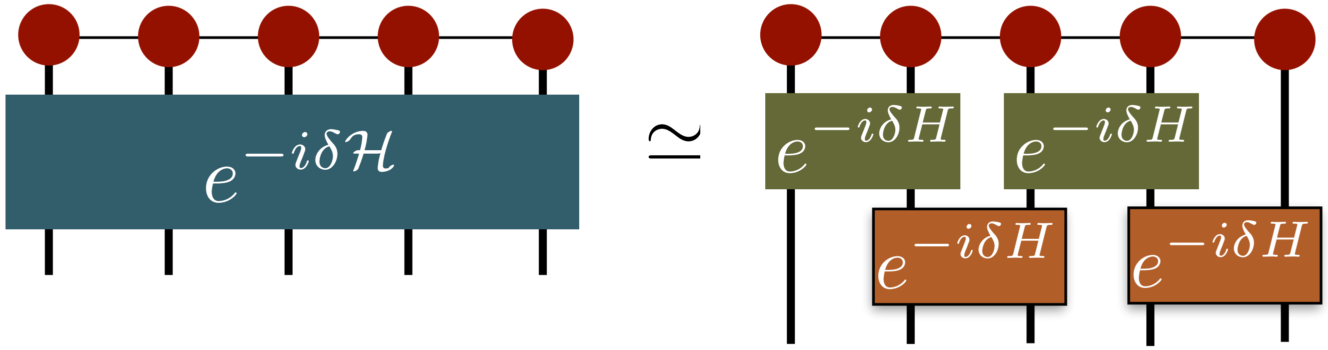


Multiplication of time evolution operator

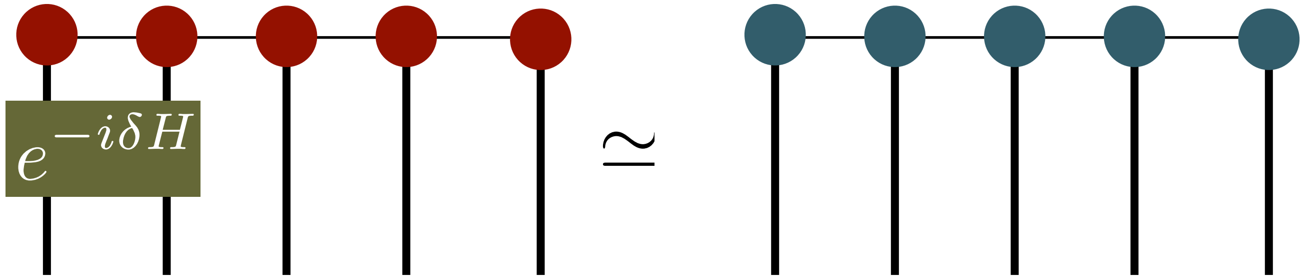
If we have MPS representation of $|\psi\rangle$

$|\psi\rangle =$ 

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.

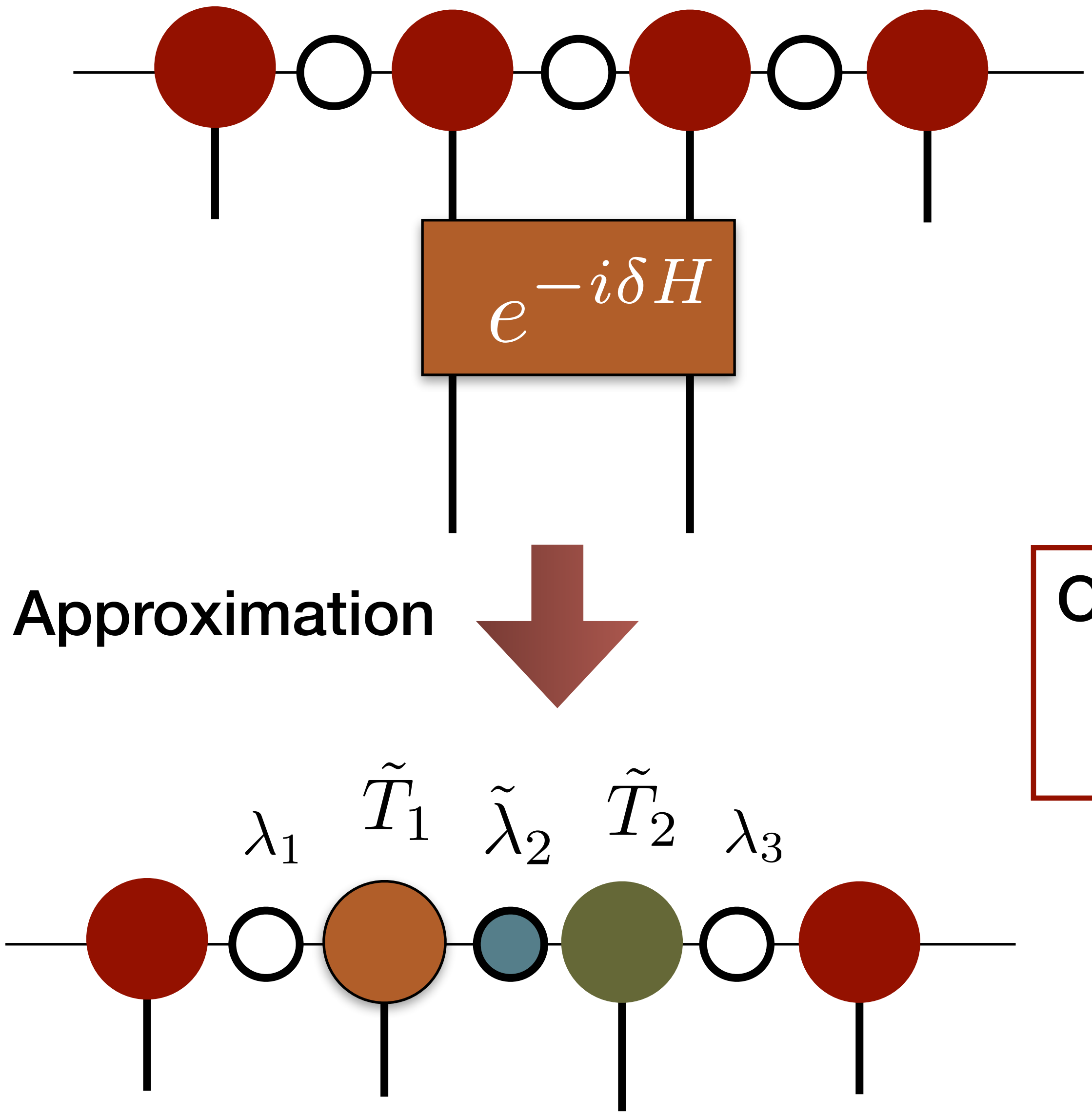
Notice: we want to keep the bond dimension χ constant along time evolution.

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.

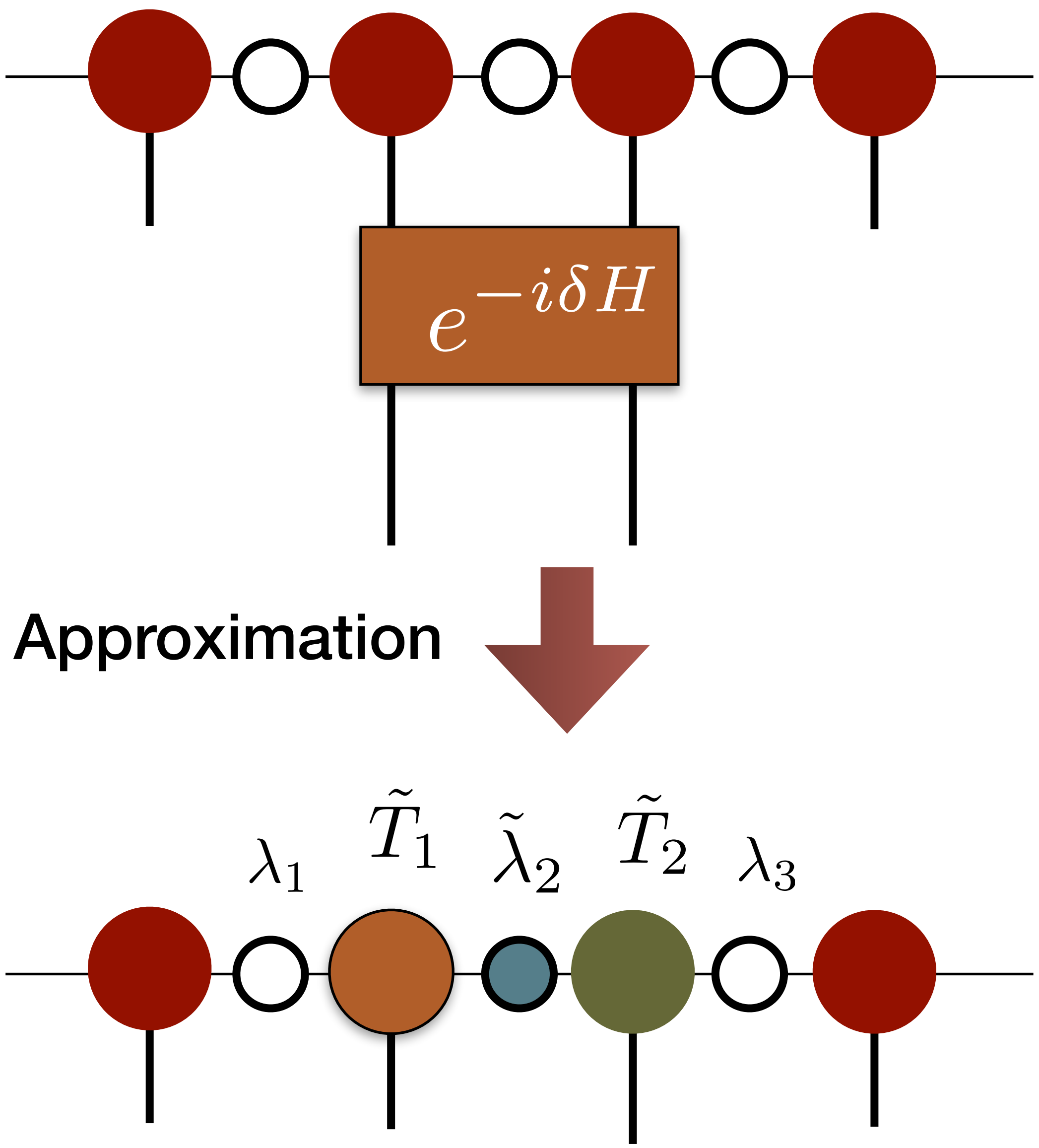


Only the two matrices which are directly applied TE operator changes in MPS.

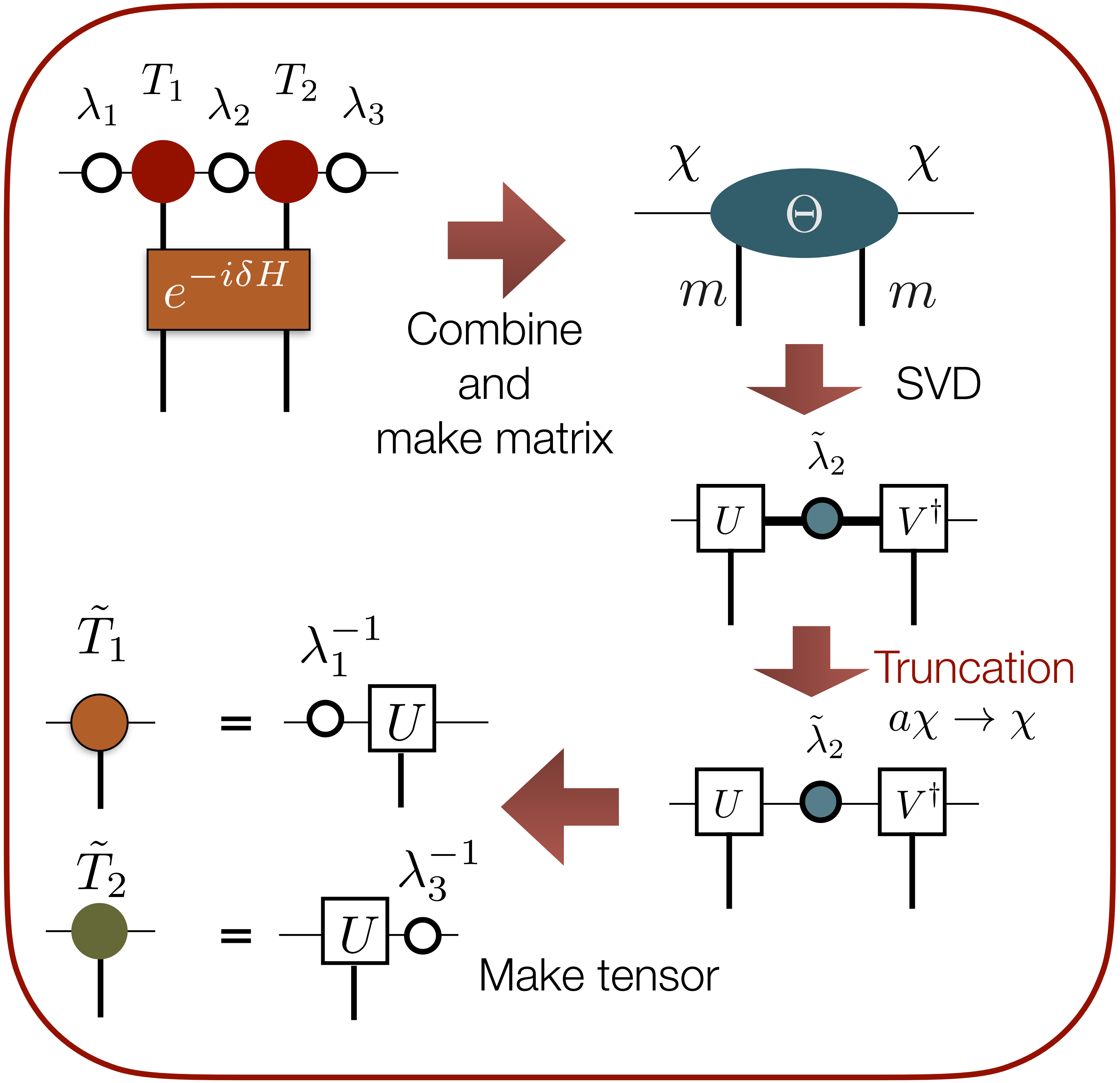
TEBD algorithm:

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

Apply TE operator



Approximation



Combine and make matrix

SVD

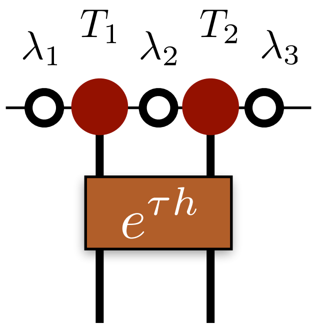
Truncation

$a\chi \to \chi$

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:

(G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007))

(R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008))

Finite system: TEBD

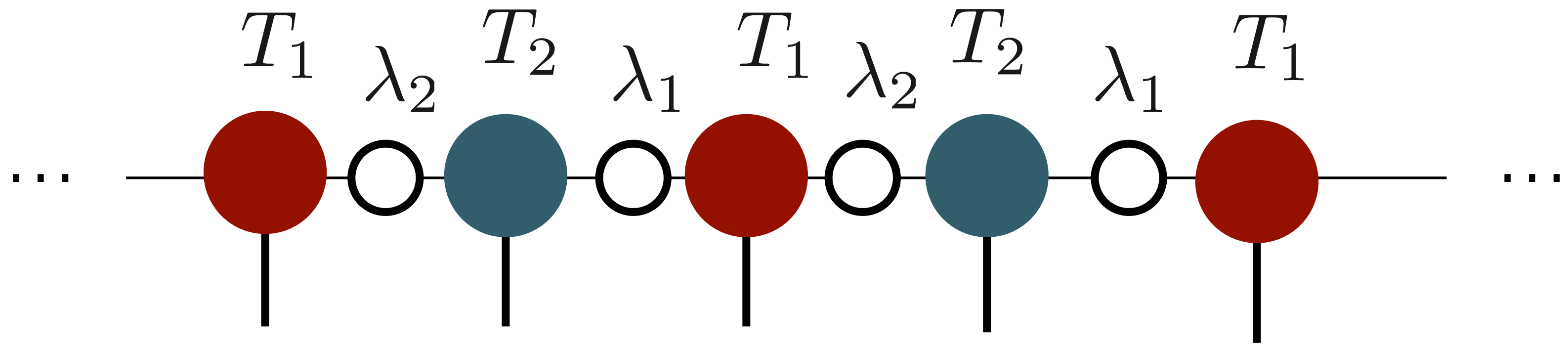
Sequentially apply ITE operators → $O(N)$ SVD for each step

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: $\displaystyle \min_{T,\lambda} \frac{\langle \Psi | \mathcal{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle}$

2. Imaginary time evolution

Simulate imaginary time evolution: $\displaystyle |\Psi_{\mathrm{GS}}\rangle \propto \lim_{\beta \to \infty} e^{-\beta \mathcal{H}} |\Psi_0\rangle$

（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.

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

3-2: iTEBD simulation

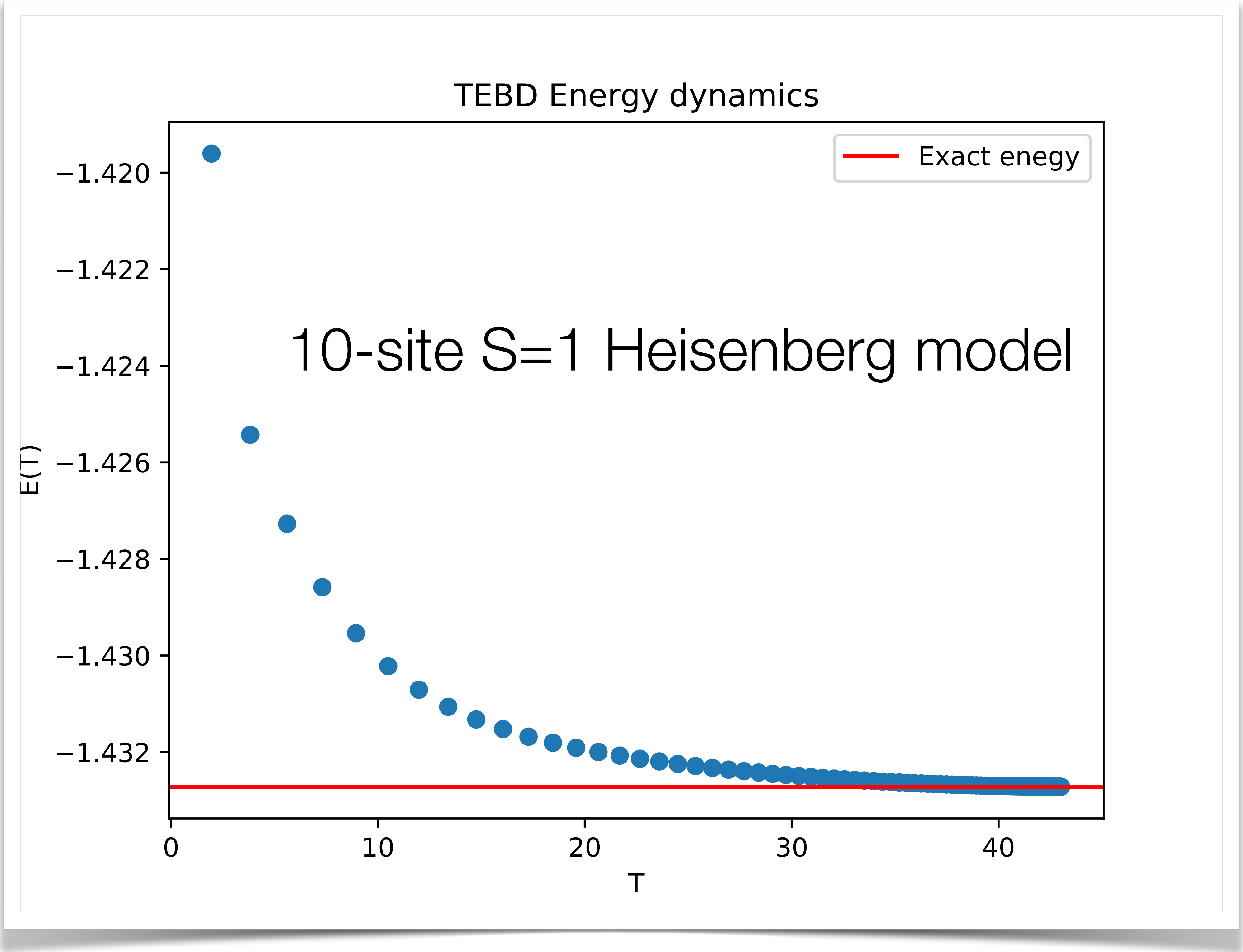
Simulate infinite system and calculate energy

Sample code: Ex3-2.py or Ex3-2.ipynb

*** Try simulation with different "chi_max", "T_step"**

*If you run them at Goole Colab, please upload ED.py and TEBD.py for Ex3-1.ipynb,
and please upload TEBD.py and iTEBD.py for Ex3-2.ipynb.

3-1: Energy dynamics in TEBD



10-site S=1 Heisenberg model

Next week

1st: Huge data in modern physics

2nd: Information compression in modern physics

3rd: Review of linear algebra

4th: Singular value decomposition and low rank approximation

5th: Basics of sparse modeling

6th: Basics of Krylov subspace methods

7th: Information compression in materials science

8th: Accelerating data analysis: Application of sparse modeling

9th: Data compression: Application of Krylov subspace method

10th: Entanglement of information and matrix product states

11th: Application of MPS

12th: Application of MPS to data science +

General tensor network representations

13th: Information compression by tensor network renormalization