計算科学における情報圧縮

Information Compression in Computational Science

2021.12.23

#11 Matrix product states
+ Application of MPS to eigenvalue problems

理学系研究科　大久保　毅

Graduate School of Science, **Tsuyoshi Okubo**

- This class is from 13:15 to 14:45 (90 min.).

Outline

- Matrix product states
    - Matrix product states (MPS)
    - Canonical form
    - infinite MPS (quick introduction only)
- Application to Eigenvalue problem
  (Ground state of quantum many-body systems)
    - Variational algorithm

Matrix product states（行列積状態）

(Tensor train decomposition)

Data compression of tensors (vectors)

Eg. 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.

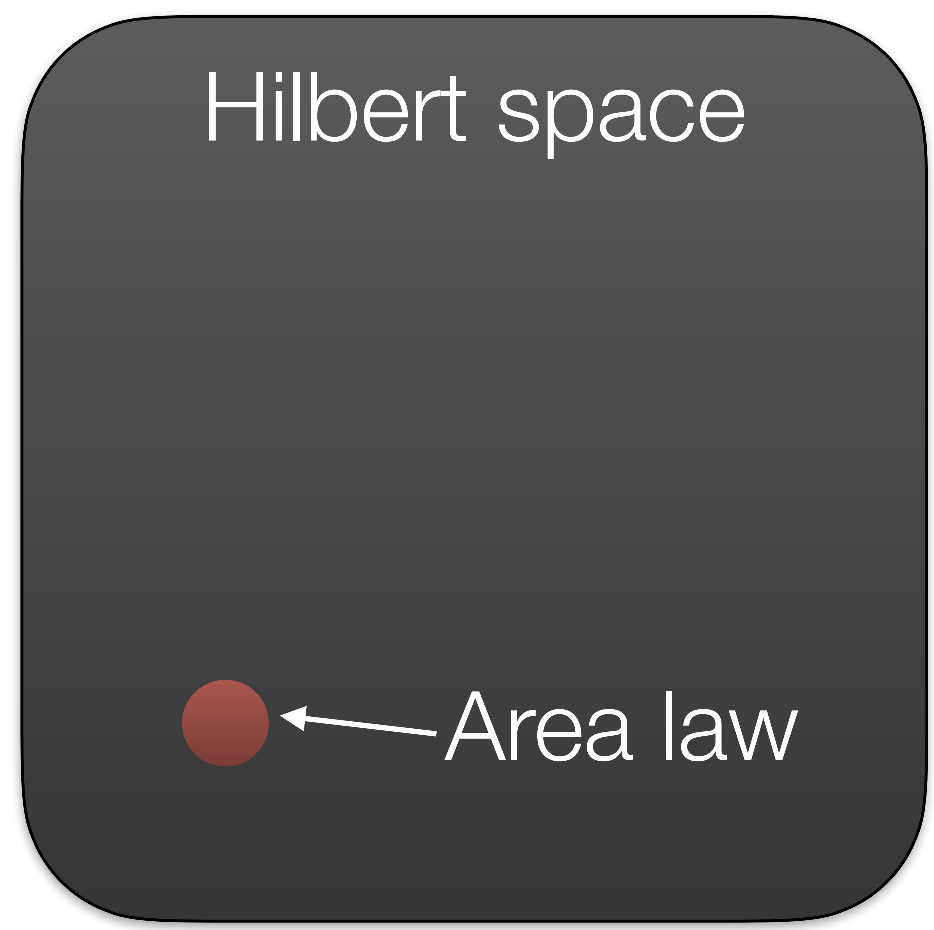
Ground states satisfy the area law.

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

Data compression by tensor decomposition:

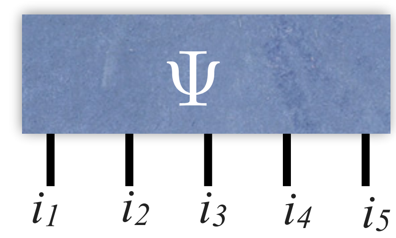
Tensor network decomposition

***Same idea holds for any tensors.**



Tensor network decomposition (tensor network states)

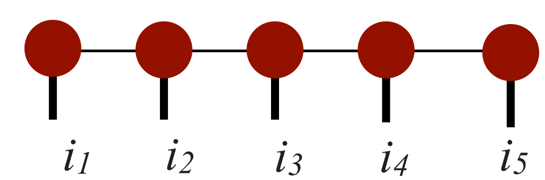
Vector (or N-leg tensor): $\Psi_{i_1 i_2 \dots i_N}$ =



# of Elements$=$a^N

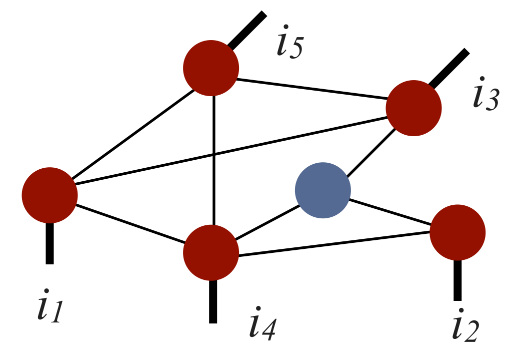
``Tensor network" decomposition

- 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

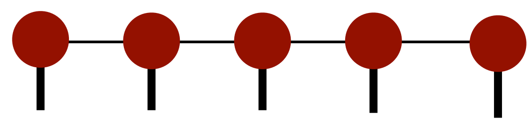
By choosing a ``good" network, we can express target vector efficiently.

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

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

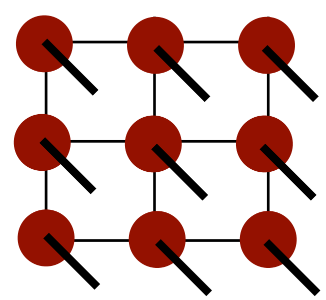
Examples of TNS

MPS:



Good for *1*d short range correlation
(e.g. *1*d gapped systems)

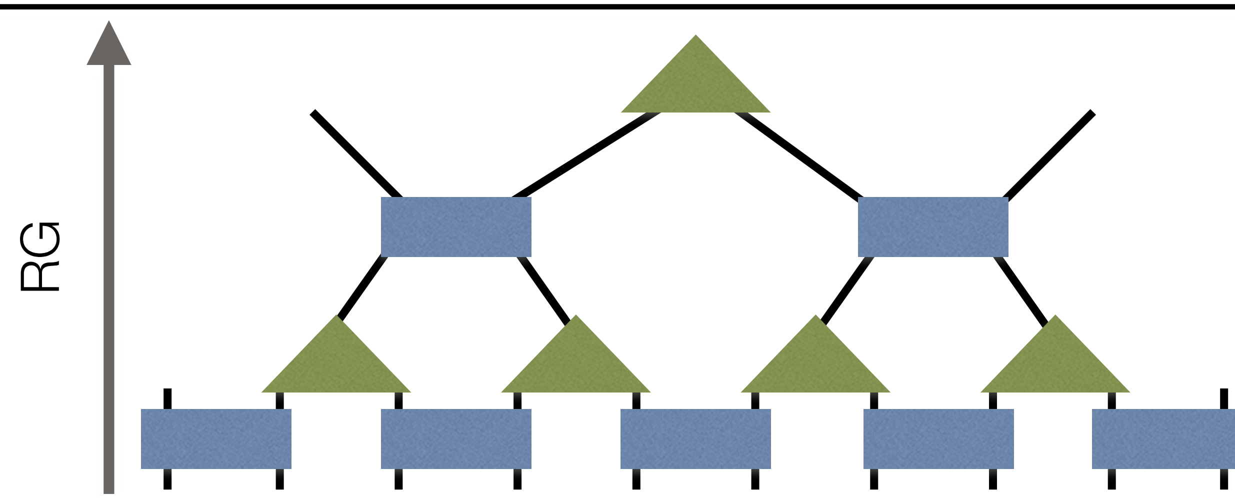
PEPS, TPS:



For higher dimensional correlation

Extension of MPS

MERA:



Scale invariant systems

Matrix product state (MPS)

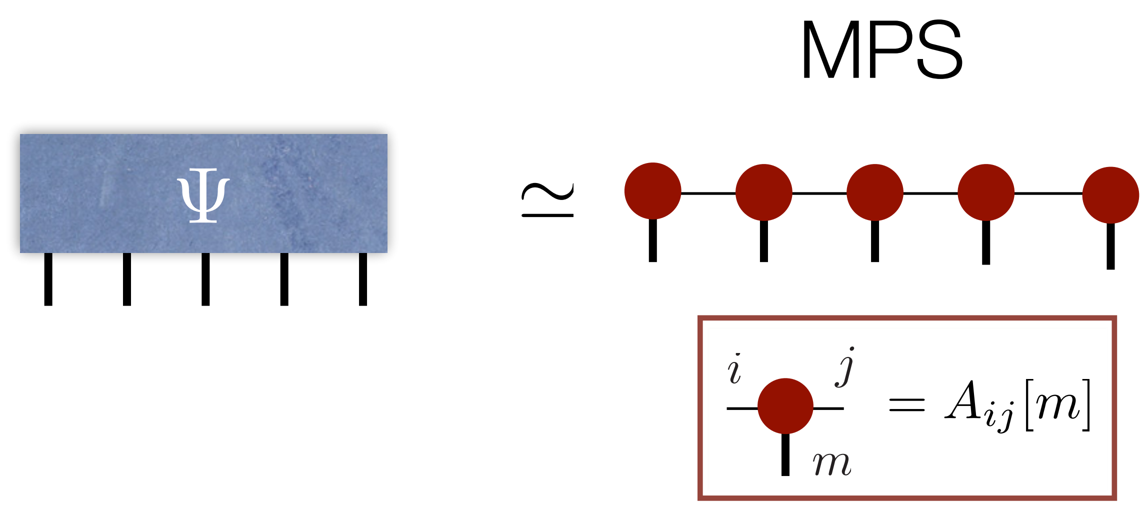
Good reviews:

(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



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)

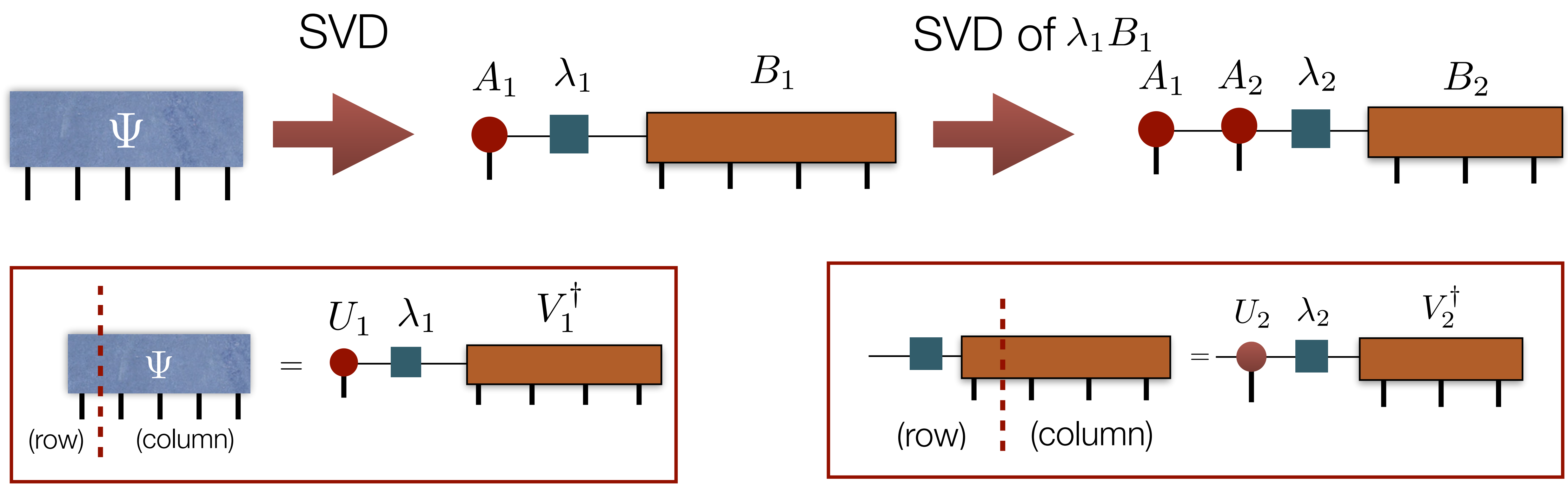
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

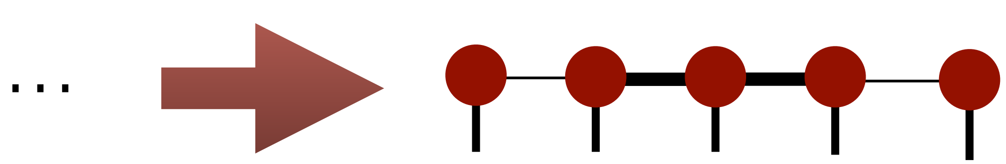
$$\Psi_{i_1 i_2 \ldots i_N} = \phi_1[i_1]\phi_2[i_2]\cdots\phi_N[i_N]$$

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

Matrix product state without approximation

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



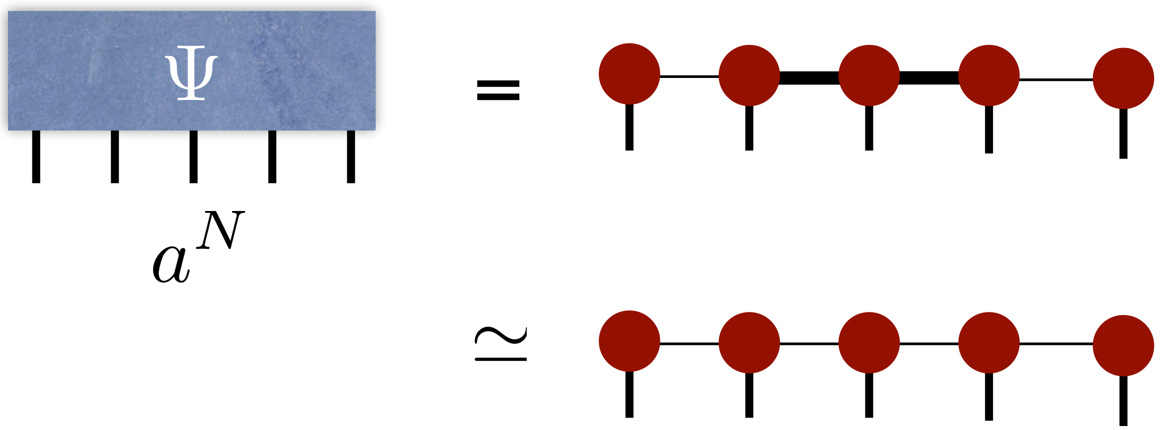


In this construction, the sizes of matrices depend on the position.

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

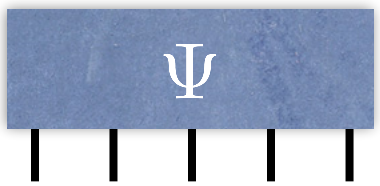
$\propto N a \chi^2$

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

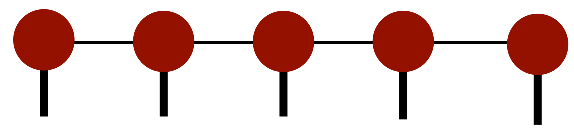
MPS is good for gapped 1d systems.

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

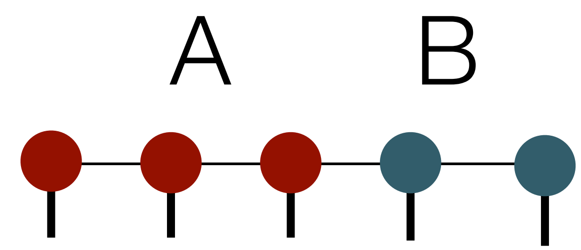


$\simeq$

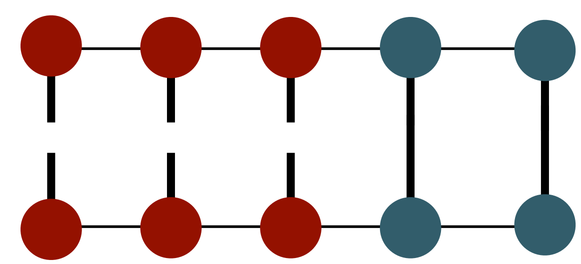


$\equiv |\tilde{\Psi}\rangle$:MPS with χ

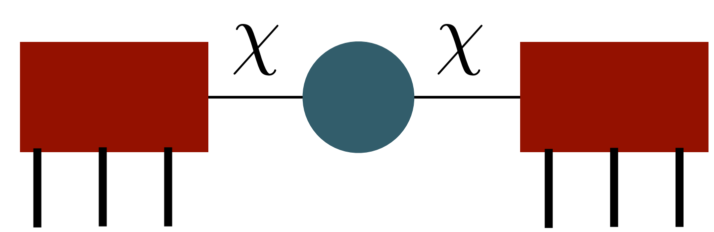
Reduced density matrix of region A:



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



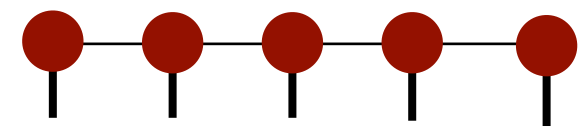
★ Structure of ρ_A:



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

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 $\qquad\qquad$ a^N

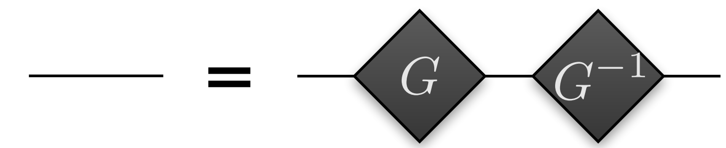
EE of the original vector	Required bond dimension in MPS representation
$S_A = O(1)$	$\chi = O(1)$
$S_A = O(\log N)$	$\chi = O(N^\alpha)$
$S_A = O(N^\alpha)$	$\chi = O(c^{N^\alpha})$

$(\alpha \leq 1)$

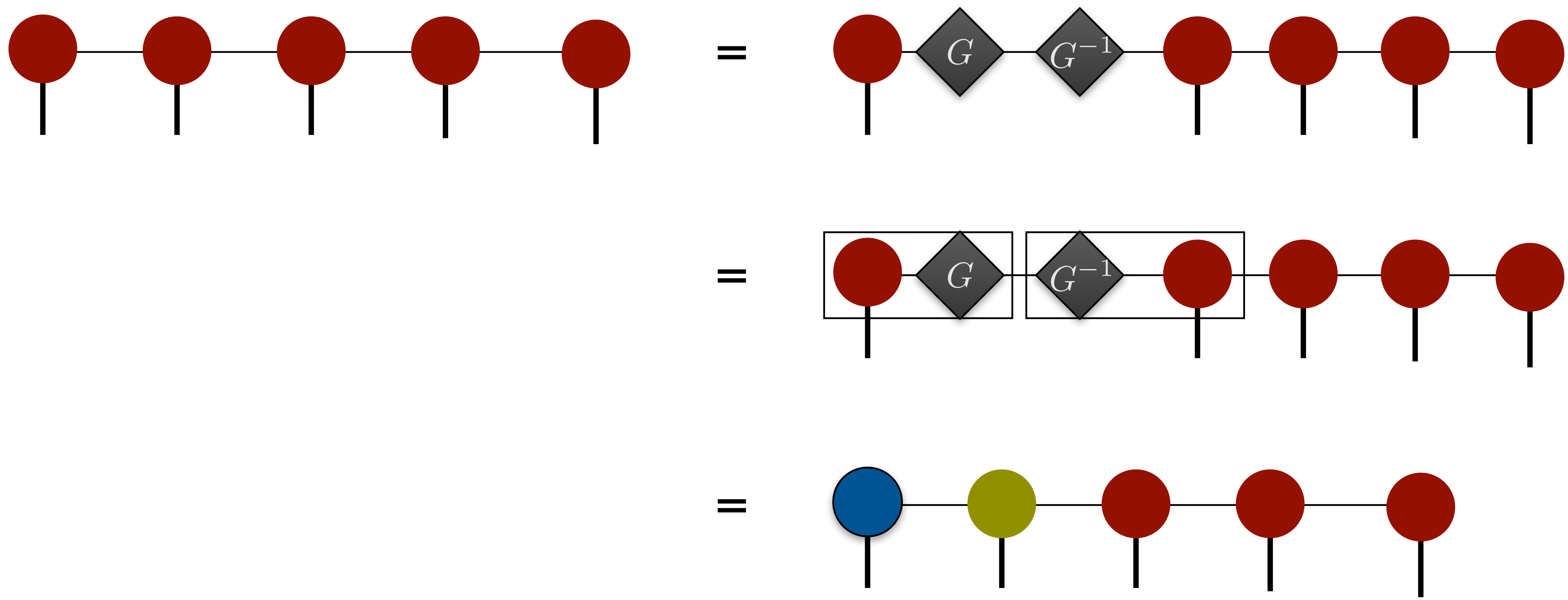
Matrix product states: Canonical form

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

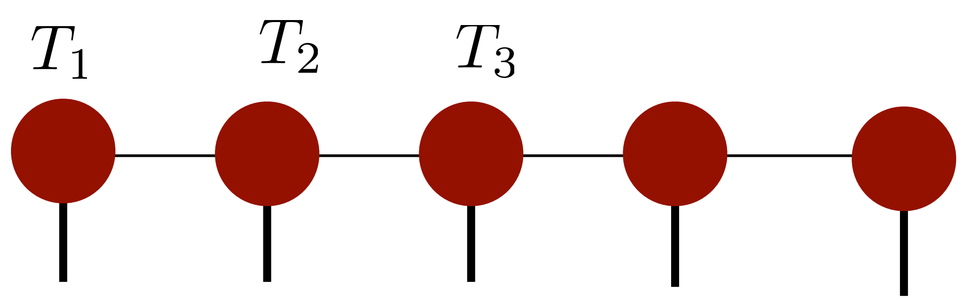


Canonical forms: Left and Right canonical forms

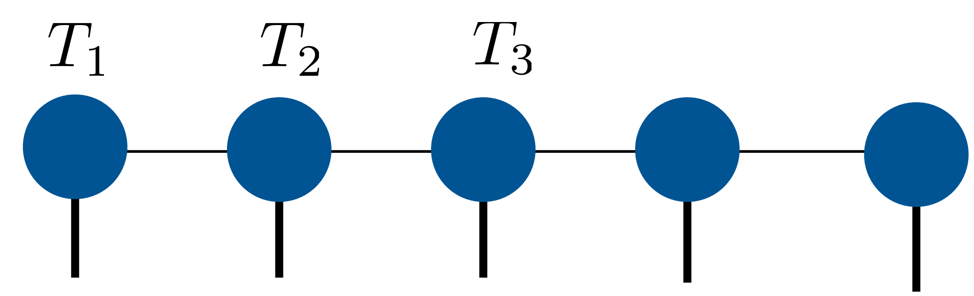
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

"canonical" forms of MPS

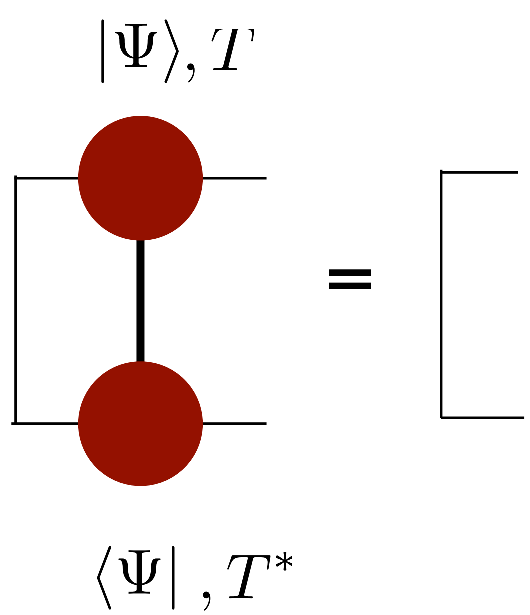
Left canonical form:

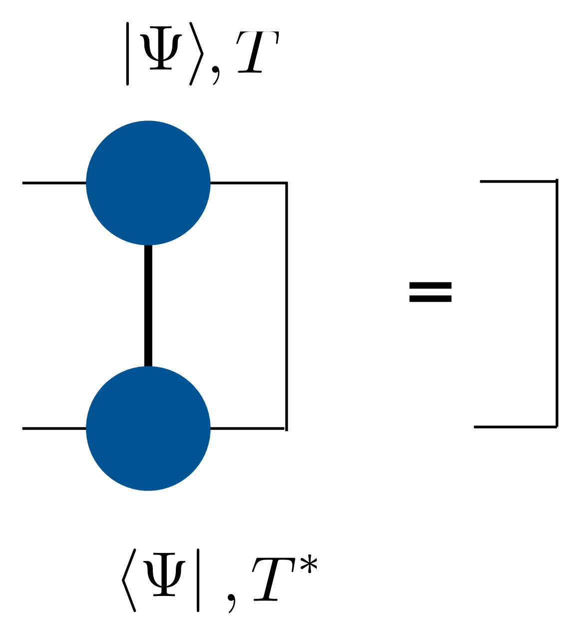


Right canonical form:



Satisfies (at least) left or canonical condition:



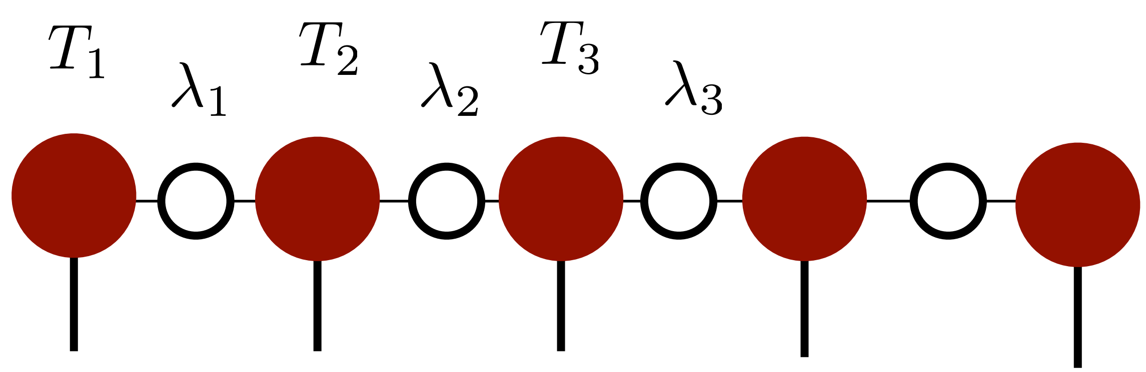


Gauge fix: Canonical form of MPS

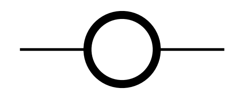
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Another canonical form of MPS: (Vidal canonical form)

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

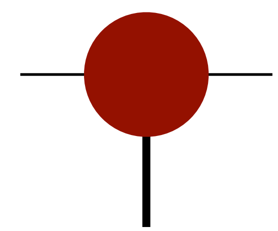


λ



:Diagonal matrix corresponding to Schmidt coefficient

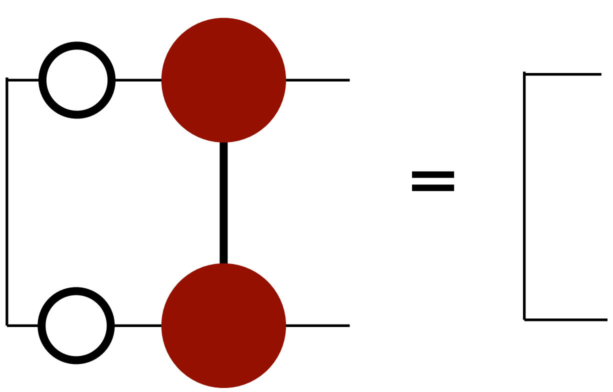
T



:Virtual indices corresponding to Schmidt basis

Left canonical condition:

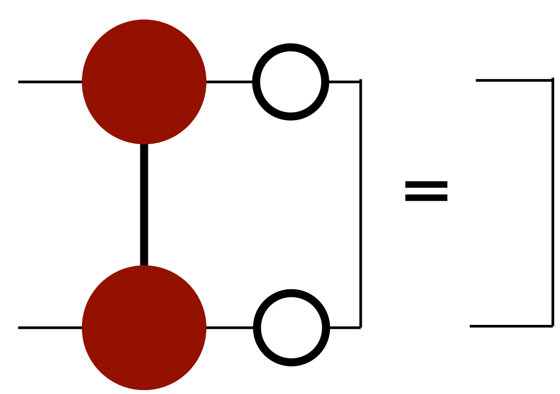
$|\Psi\rangle, T$



$\langle\Psi|\,, T^*$

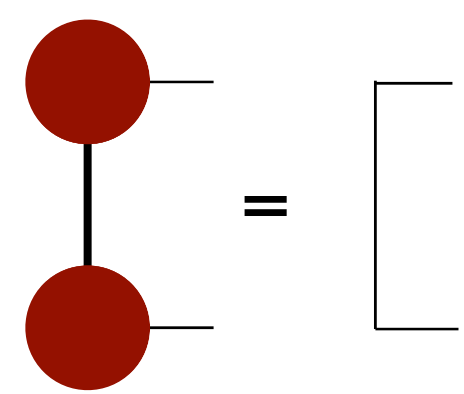
Right canonical condition:

$|\Psi\rangle, T$



$\langle\Psi|\,, T^*$

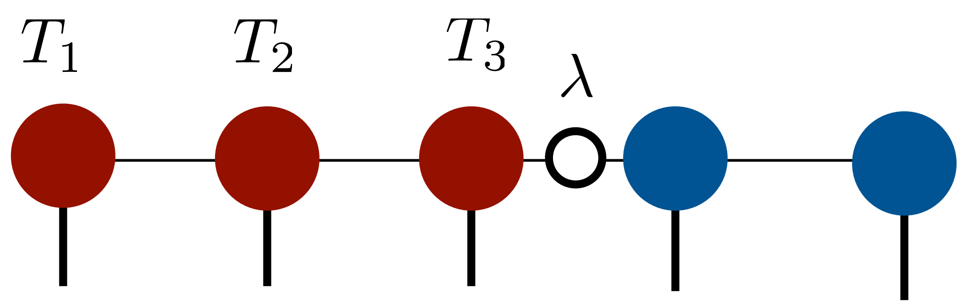
(Boundary)



Canonical forms: Mixed canonical forms

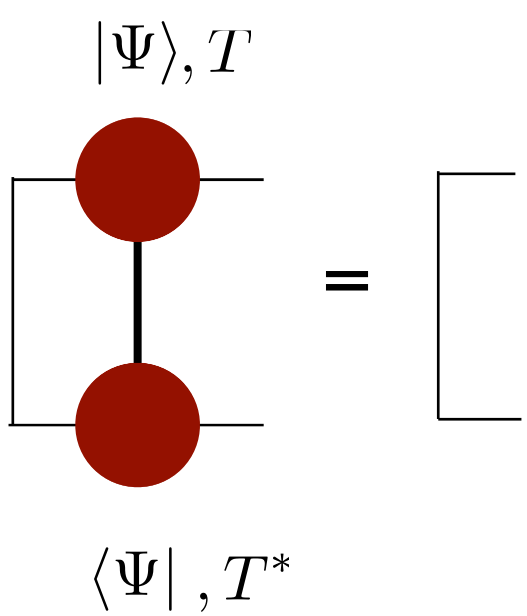
Ref. U. Schollwöck, Annals. of Physics **326**, 96 (2011)

Mixed canonical form:

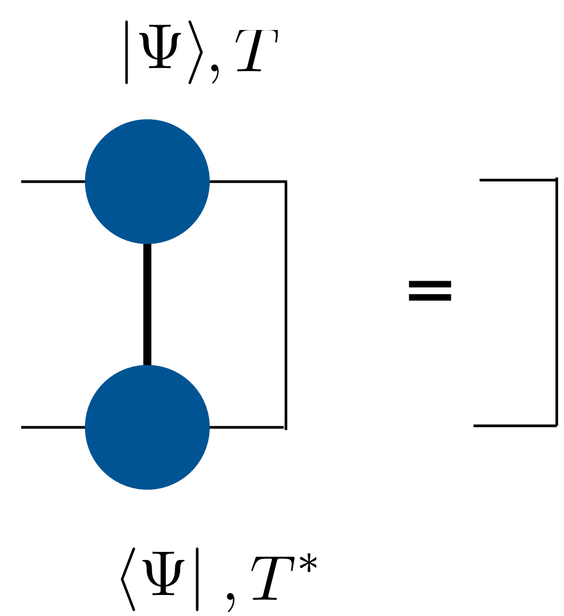


λ is identical with the Schmidt coefficient.

Left canonical condition:



Right canonical condition:

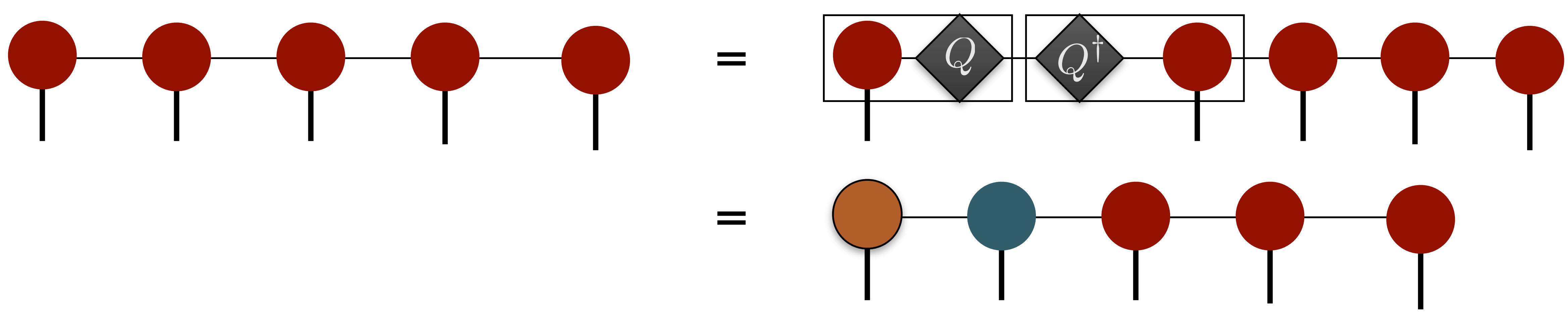


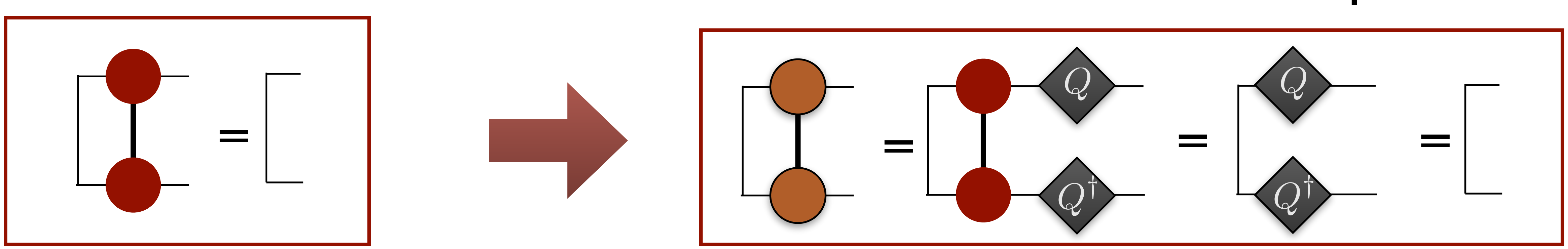
Canonical forms: Note

- 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

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

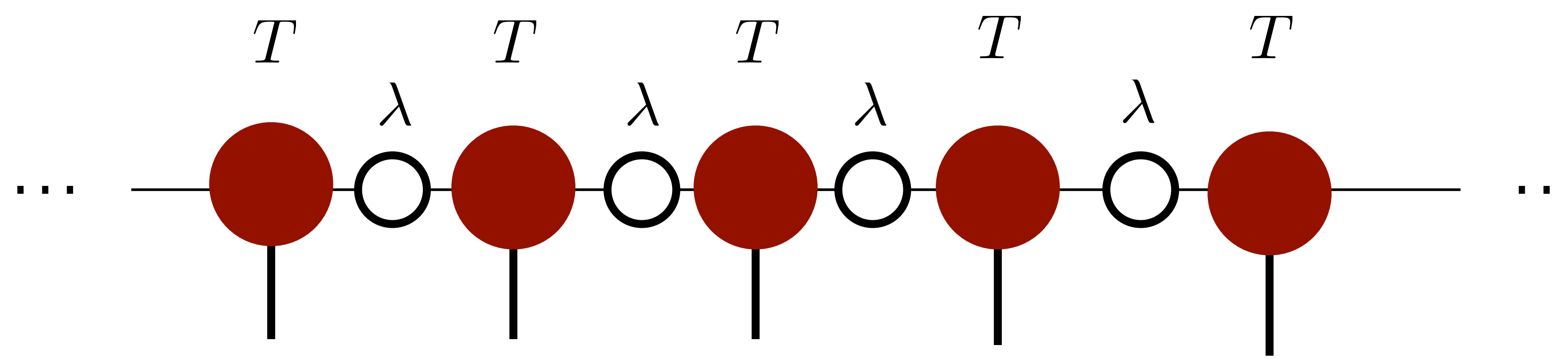




Matrix product states: infinite MPS

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!

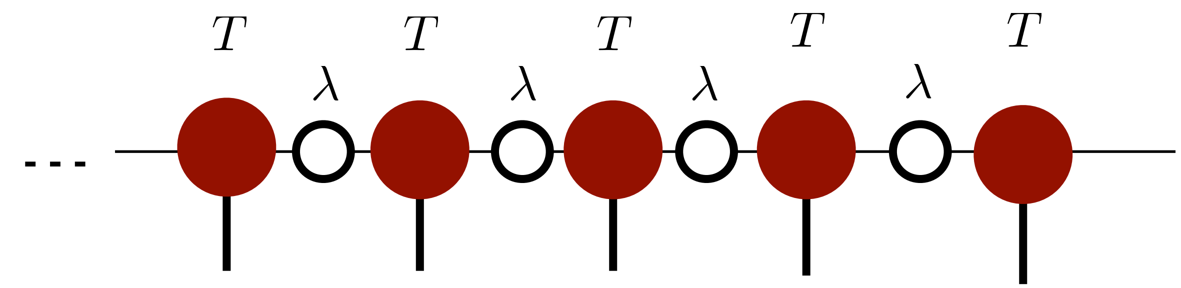
* Infinite MPS can be accurate when the EE satisfies the 1d area low (S~O(1)).

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)

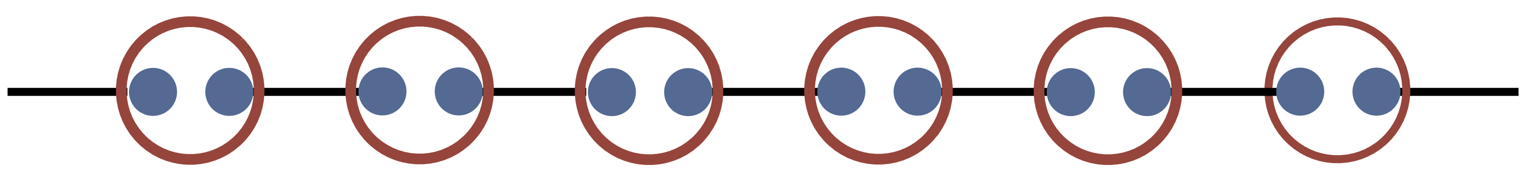
S=1 Affleck-Kennedy-Lieb-Tasaki (AKLT) Hamiltonian:



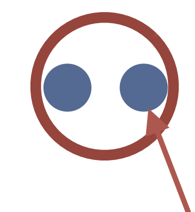
$$\mathcal{H} = J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \frac{J}{3} \sum_{\langle i,j \rangle} \left(\vec{S}_i \cdot \vec{S}_j \right)^2$$

$$(J > 0)$$

The ground state of AKLT model:



S=1 spin:



S=1/2 spin

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

Spin singlet

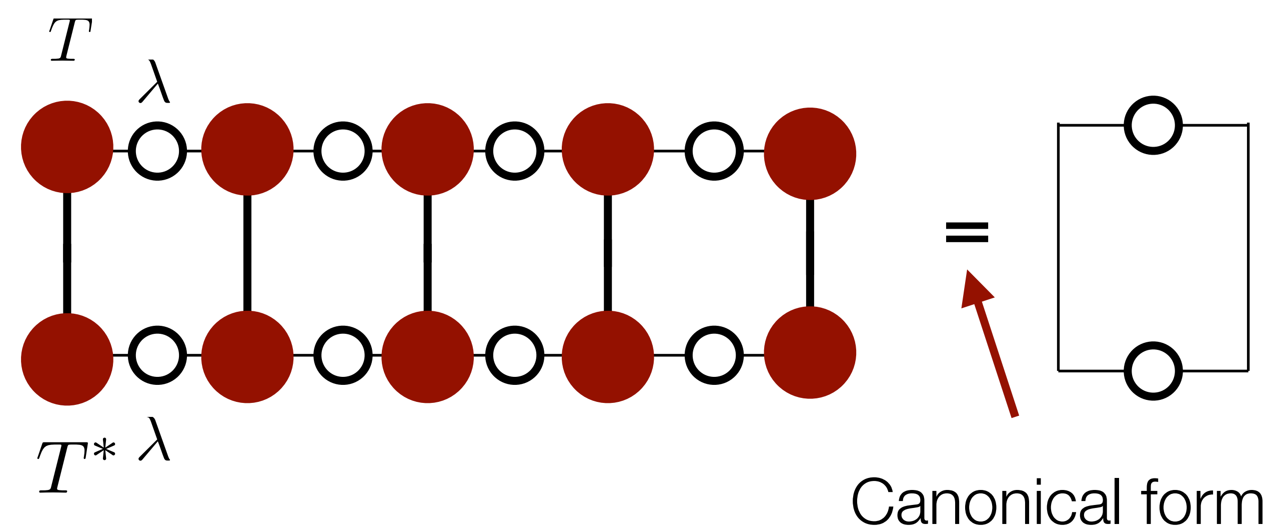
$$T[S_z = 1] = \sqrt{\frac{4}{3}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

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

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

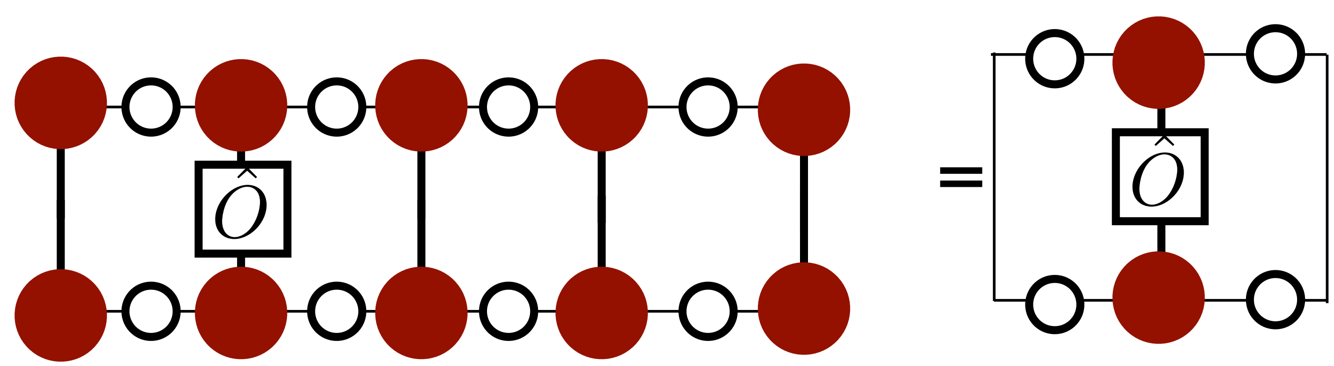
Calculation of expectation value (will be skipped)

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



$= \sum_i \lambda_i^2 = 1$

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



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.

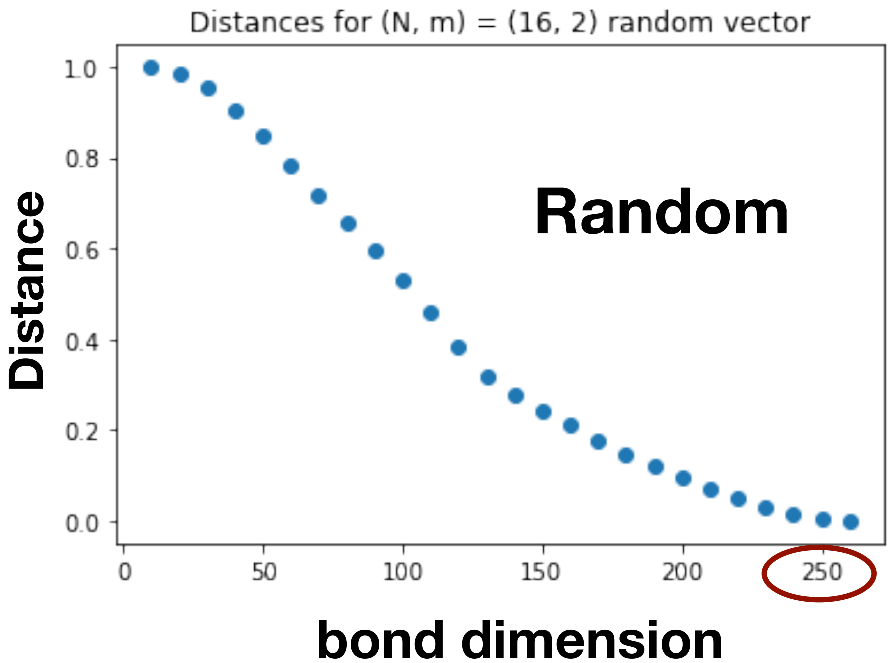
*In the case of Ex2-3 you also need picture file.

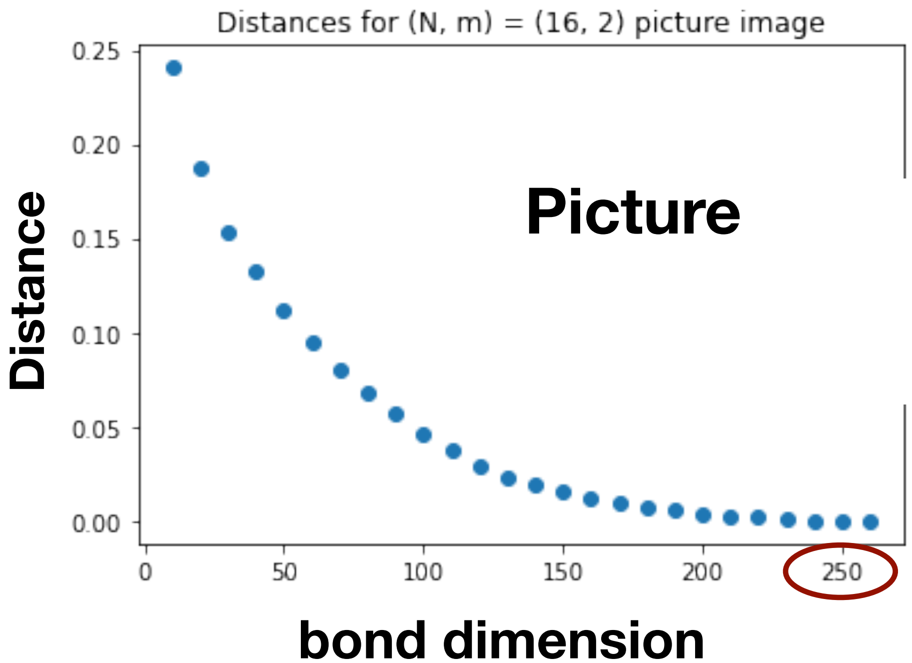
This exercise is similar to the Report1-2.

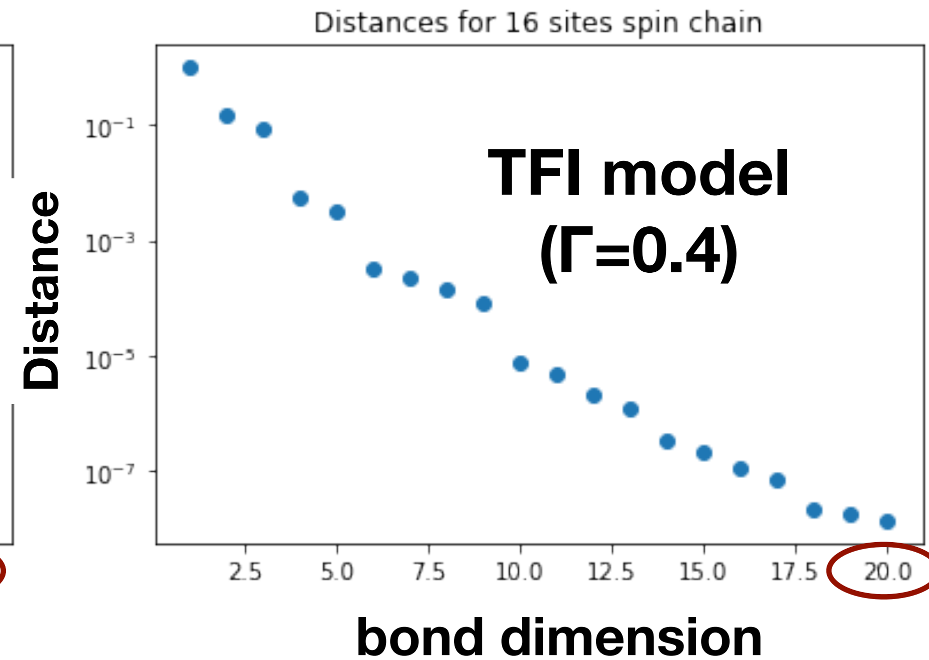
Exercise 2: Make MPS and approximate it

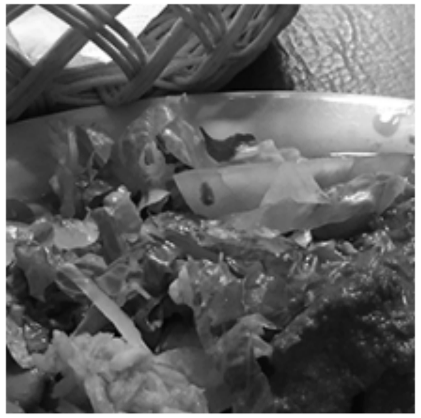
2^{16} dimensional vectors (=16-leg tensors)

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

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

$+$

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

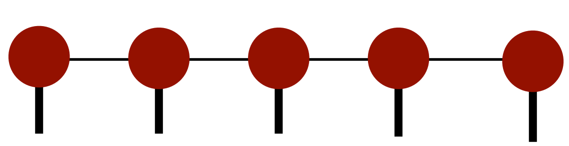
$$\mathcal{H}\vec{v}_0 = E_0\vec{v}_0$$

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

Variational calculation using MPS:

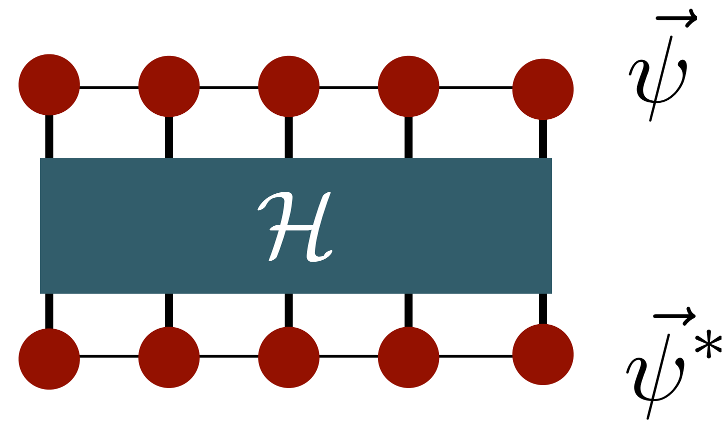
Cost function: $F = \dfrac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}}$

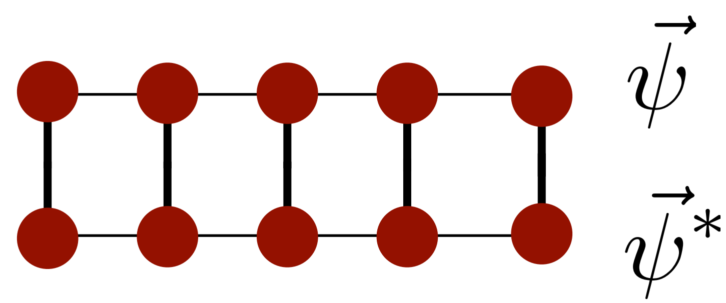
Find the MPS which minimizes F
by optimizing matrices in MPS.

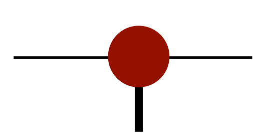
$\vec{\psi} =$ 

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

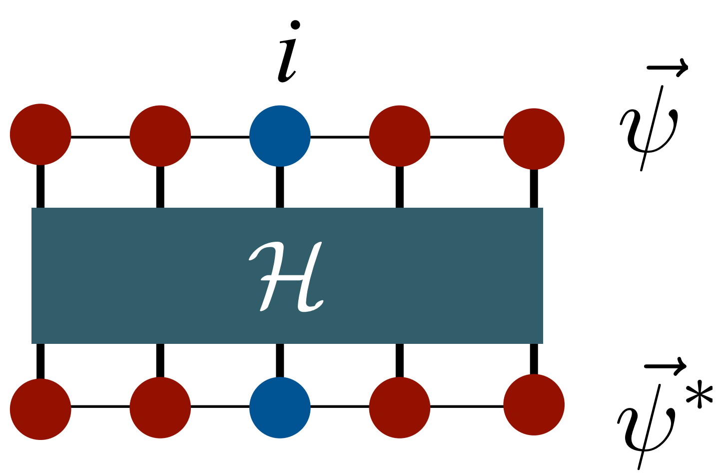
$$F = \frac{\vec{\psi}^\dagger(\mathcal{H}\vec{\psi})}{\vec{\psi}^\dagger\vec{\psi}} = \frac{A_i^\dagger(\tilde{\mathcal{H}}_i A_i)}{A_i^\dagger(\tilde{\mathcal{N}}_i A_i)}$$

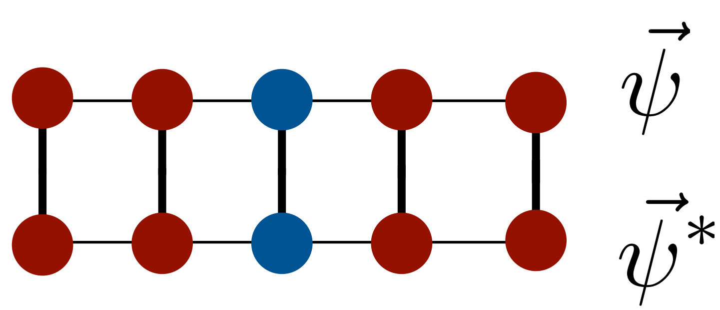
Solve generalized eigenvalue problem

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

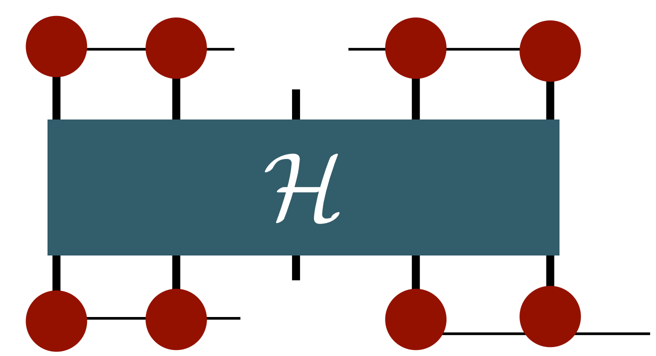
(Find the lowest eigenstate)

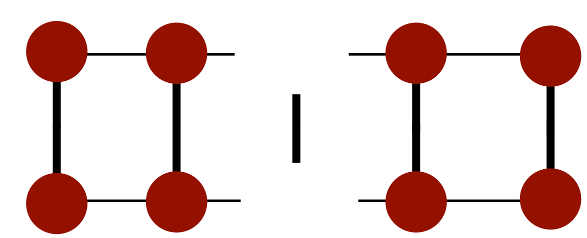
Notice: matrix size = $a\chi^2 \times a\chi^2$

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

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

Remove A_i

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

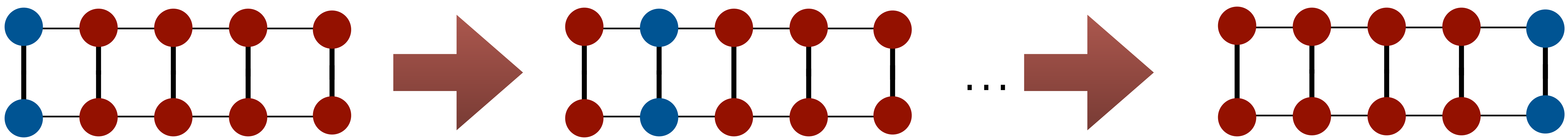
$\tilde{\mathcal{N}}_i \equiv$ 

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

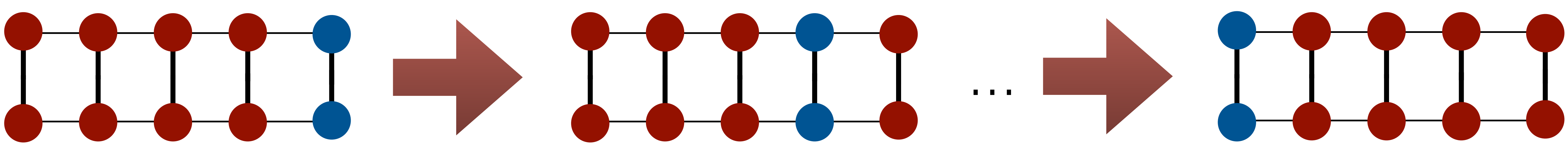
Iterative optimization

(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



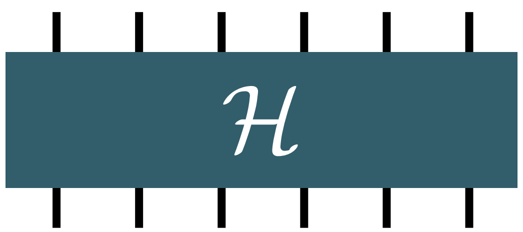
Repeat sweeping until convergence.

Compact representation of an operator

Notice!

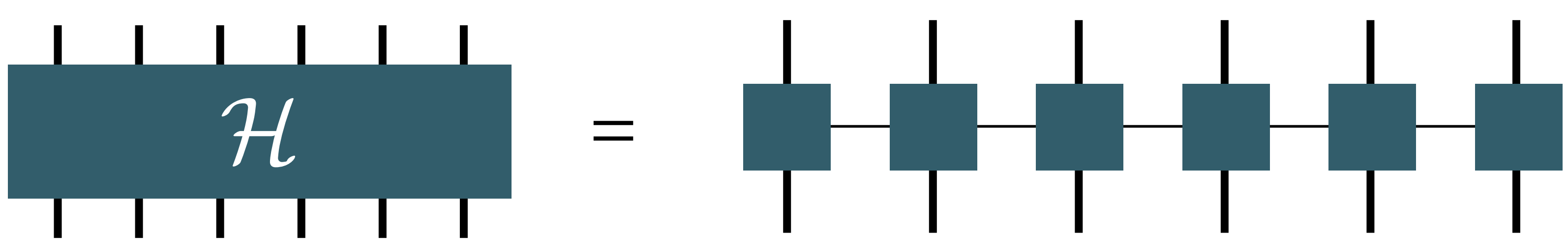
We can conduct this algorithm when we can represent the matrix efficiently.

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.



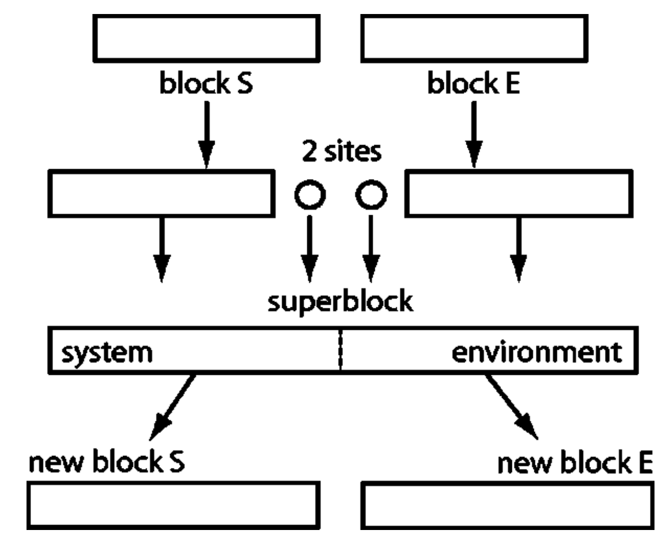
E.g. The Hamiltonian of the Heisenberg model is represented by MPO with bond dimension $\chi = 5$.

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)

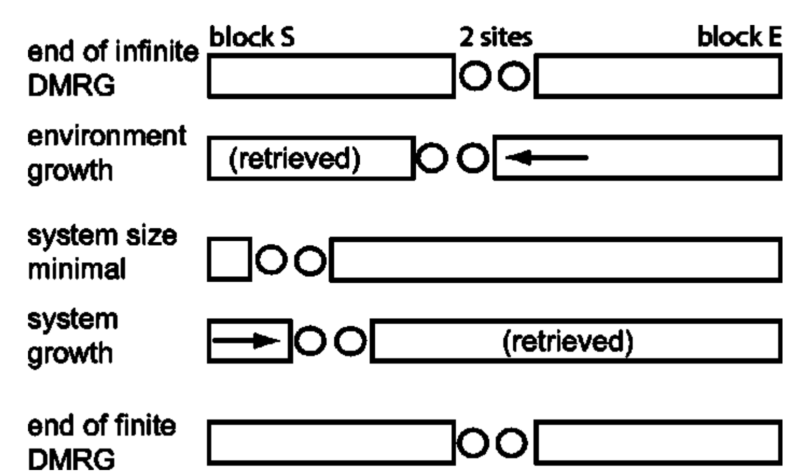
(S. R. White, Phys. Rev. Lett. **69**, 2863 (1992))
(U. Schollwöck, Rev. Mod. Phys. **77,** 259 (2005))
(U. Schollwöck, Annals. of Physics **326**, 96 (2011))

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.

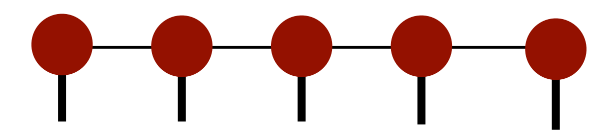
Relation to Density Matrix Renormalization Group

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

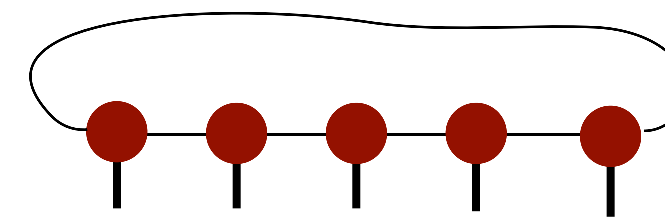
(F. Verstraete, D. Porras, and J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004))

Accuracy becomes worse if we consider systems with periodic boundary condition.

open:

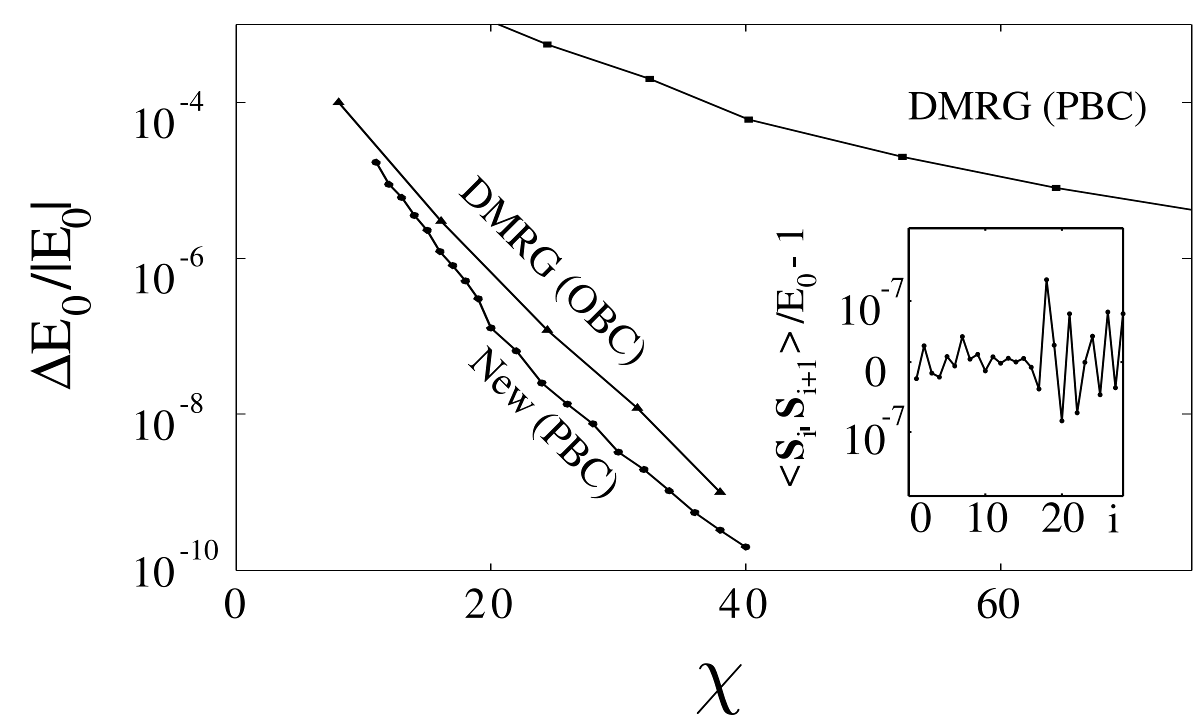


periodic:



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

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



Next (Jan. 6th)

1st: Huge data in modern physics (Today)

2nd: Information compression in modern physics

(+review of linear algebra)

3rd: Review of linear algebra (+ singular value decomposition)

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: Matrix product states + Application of MPS to eigenvalue problems

12th: Application of MPS to time evolution and data science

13th: Other tensor network representations

+ (Appendix: Information compression by tensor network renormalization)