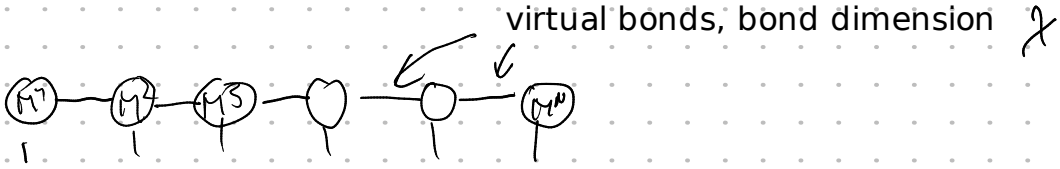
Introduction to DMRG

Johannes Hauschild, UC Berkeley
Winterschool Barcelona Sept 2021

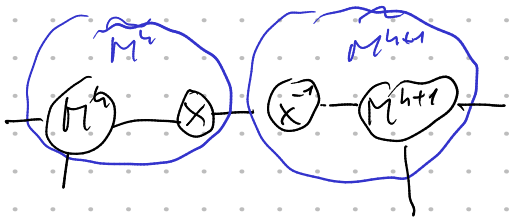
References:
White PRL 69, 2863 (1992)
JH, Pollmann arXiv:1805.0055 (follow here)
Schollwoeck arXiv:1008.3477

Recap

MPS ansatz



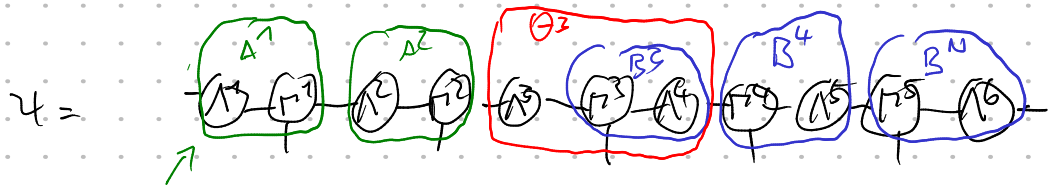
Gauge freedom



yields same MPS

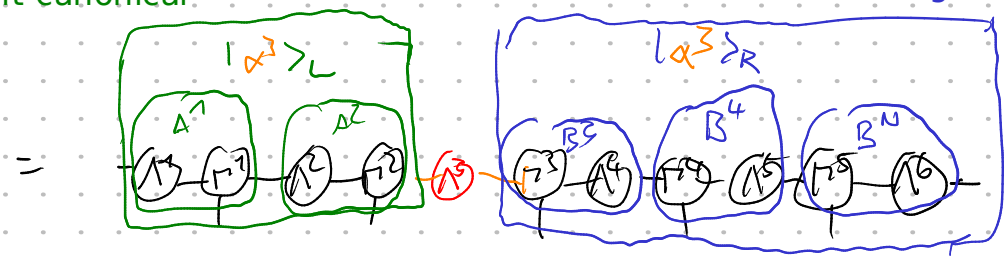
for $X \quad X^{-1} = \mathbb{1}$

can be used to bring the MPS in canonical from defined by the Schmidt decomposition



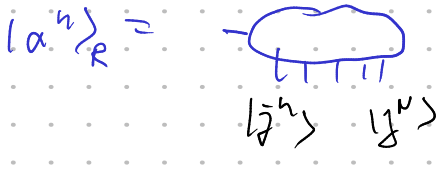
left canonical

right canonical

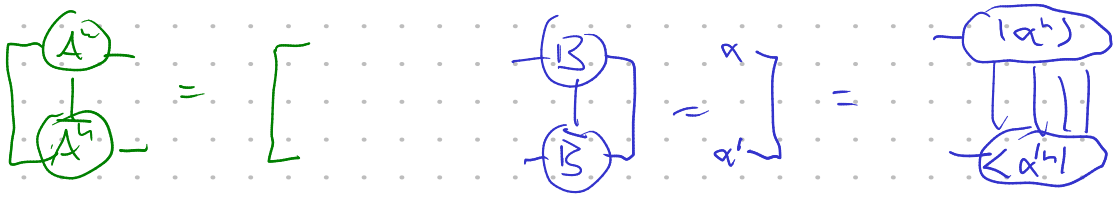


Schmidt states form an orthonormal basis on left/right part!

→ identify virtual bond indices with Schmidt states



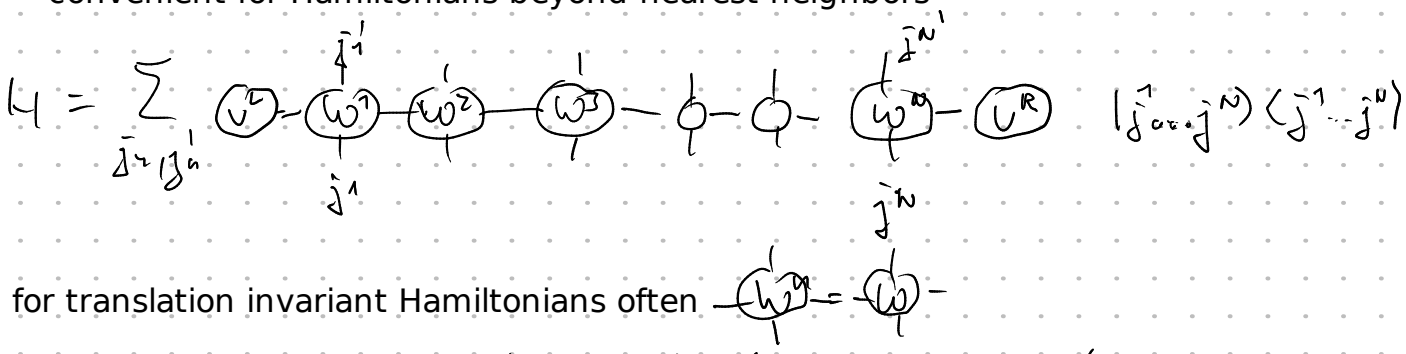
Orthonormality conditions



Matrix product operators

= generalization of MPS to operators

convenient for Hamiltonians beyond nearest neighbors



for translation invariant Hamiltonians often

↳ entries = local operators on site n

$$W^{(n)} = \begin{pmatrix} \mathbb{1}_n & Z_n & -gX_n \\ 0 & 0 & -JZ_n \\ 0 & 0 & \mathbb{1} \end{pmatrix} \qquad v_L = (1\ 0\ 0) \ , \quad v_R = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\rightarrow \quad v_L W W v_R = (\mathbb{1}_1 \ Z_1 \ -gX_1) \begin{pmatrix} -gX_2 \\ -JZ_2 \\ \mathbb{1}_2 \end{pmatrix}$$

$$= -g\mathbb{1}_1 X_2 - J Z_1 Z_2 - g X_1 \mathbb{1}_2$$

$$\rightarrow H = -J \sum_n Z_n Z_{n+1} - g \sum_n X_n \qquad = \text{transverse field ising model}$$

General block structure for the MPO

$$W = \begin{pmatrix} \mathbb{1} & C & D \\ & A & B \\ & & \mathbb{1} \end{pmatrix} \qquad v_L = (1, 0 \dots 0, 0) \ , \quad v_R = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

yields

$$H = \sum_n D_n + C_n B_{n+1} + C_n A_{n+1} B_{n+2} + C_n A_{n+1} A_{n+2} B_{n+3} + \dots$$

$$W = \begin{pmatrix} \mathbb{1} & x^h & y^h & -h z^h \\ & & & J x^h \\ & & & J y^h \\ & & & \mathbb{1} \end{pmatrix} \quad \rightsquigarrow H = J \sum_h \left(x^h x^{h+1} + y^h y^{h+1} \right) - h \sum_h z^h$$

$$W = \begin{pmatrix} \mathbb{1} & p^h & 0 & 0 \\ & & x^h & 0 \\ & & & p^h \\ & & & \mathbb{1} \end{pmatrix} \quad \rightsquigarrow \quad H = \sum_h p_h \, x_{h+1} \, p_{h+2}$$

One can formalize this with the finite state machine picture to autogenerate MPOs given the terms of H

Side note: W is not diagonalizable, it Jordan block strucute:

$$W \cong \begin{pmatrix} 1 & \varepsilon \\ & 1 \end{pmatrix}$$

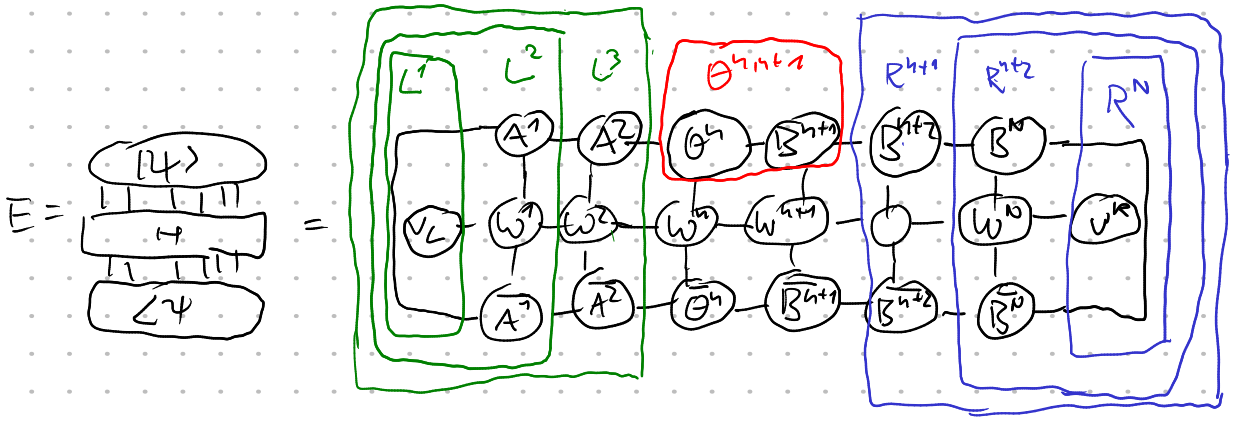
$$\begin{pmatrix} 1 & \varepsilon \\ & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \varepsilon \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & \varepsilon \\ & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

H is extensive, applying the transfer matrix adds energy density!

Density Matrix Renormalization Group (DMRG) algorithm

goal: variational minimization of the energy $E = \langle \psi | H | \psi \rangle$



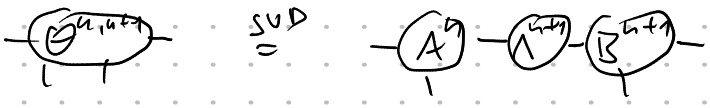
left/right environments
contain only left/right-canonical MPS tensors!

Finite DMRG

Start with an MPS in right canonical form

for n in {1, 2, ... N-1}: right sweep

find the optimal  keeping other tensors fix

split  and truncate

use A^n to calculate updated L^{n+1}

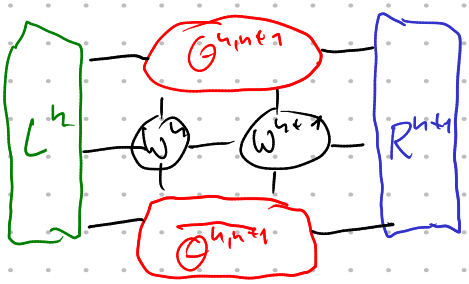
for n in {N-2, ..., 1}: left sweep

find optimal $\Theta^{n,n+1}$

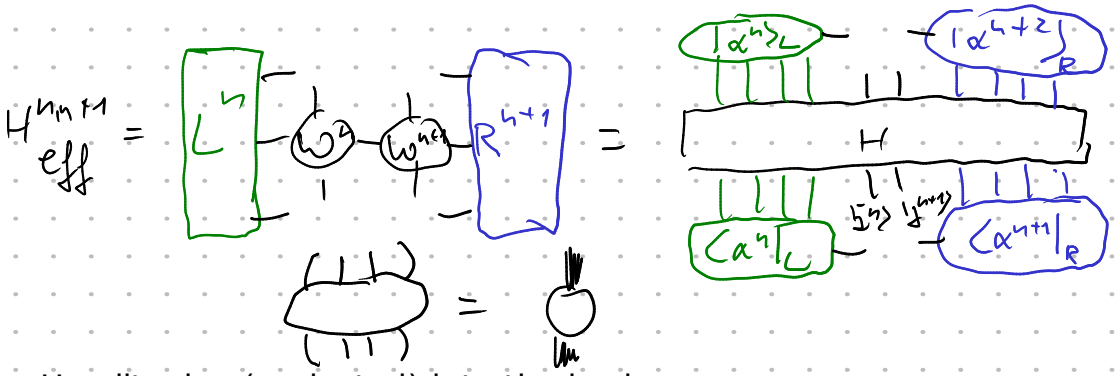
split  and truncate

use B^{n+1} to calculate the updated R^n

How to find optimal $\Theta^{n,n+1}$?

$E =$ 

define



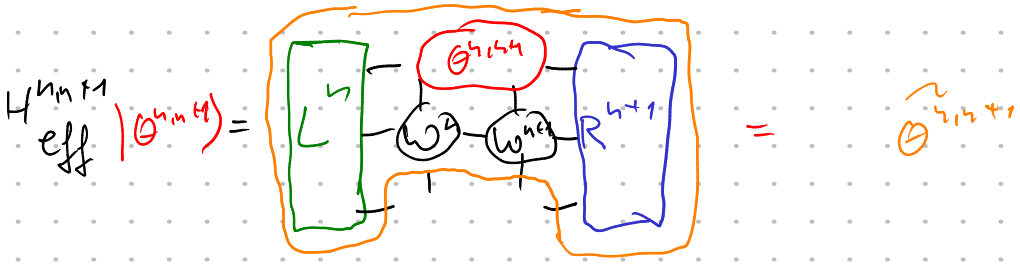
This is the Hamiltonian (projected) into the basis

$$|\alpha^n\rangle_L \otimes |j^n\rangle \otimes |j^{n+1}\rangle \otimes |\alpha^{n+2}\rangle_R$$

This is an orthonormal basis!

Hence, the optimal  is simply given by diagonalizing $H_{\text{eff}}^{n,n+1}$

viewed as a matrix acting on the top indices



The ground state of $H_{\text{eff}}^{n,n+1}$ can be found with Lanczos algorithm (or similar)

Note: contraction order matters!

$O(\chi^3 D d^2)$ vs $O(\chi^4 d^4)$

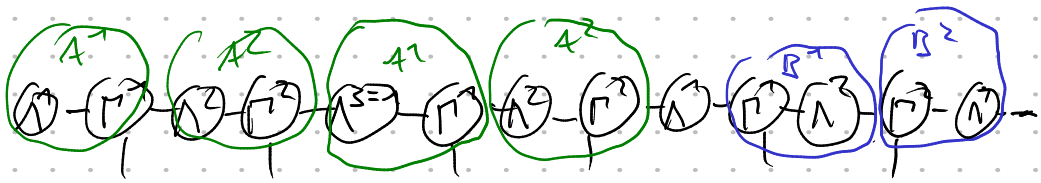
One can also do this optimizing a single site at once

Infinite DMRG

consider a system which is translation invariant with unit cells with N sites

say N=2

$|\psi\rangle = \cdots$



Start with a small system of just one unit cell

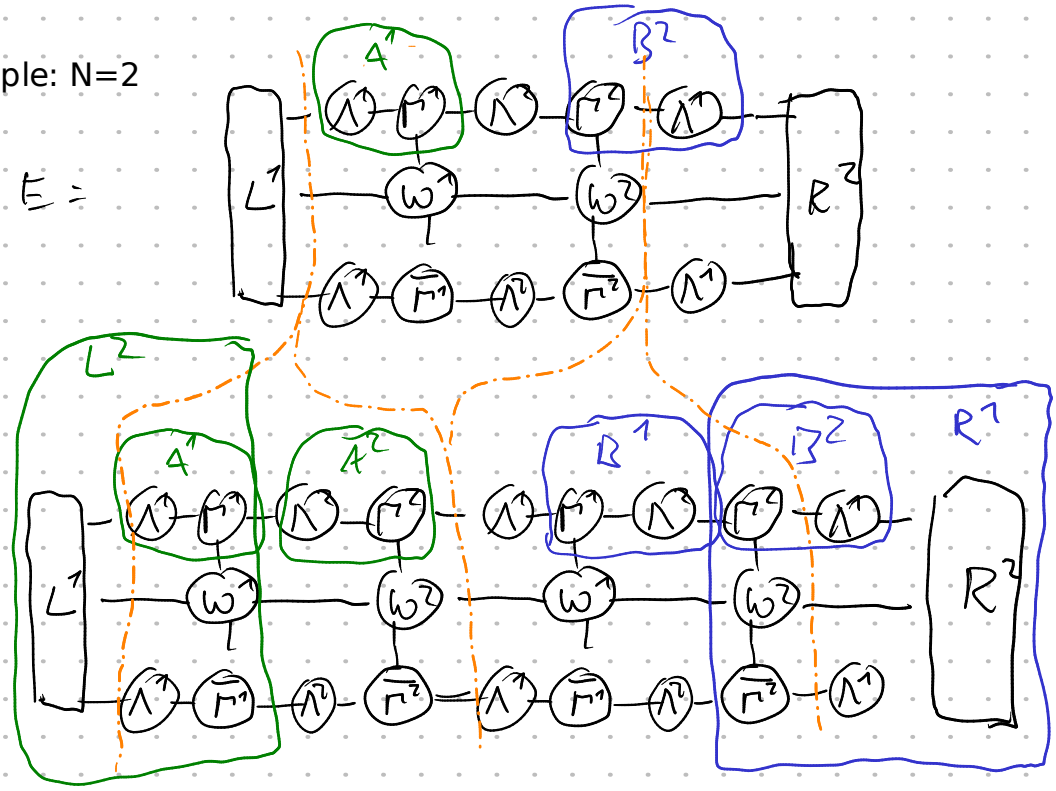
sweep left-right left in a single unit cell (as in finite DMRG)
but include the bond between unit cells (which is trivial for finite DMRG)

During each sweep, grow the left/right environets by one unit cell,
assuming translation invariance

Repeat until actual translation invariance is reached

Example: N=2

$E =$



Note: total energy increases by energy density!