IV. Simulations with MPS

MPS form good approx. e.g. for ground states, and many quantities can be extracted efficiently:
Can we use them as a numerical tool to study the physics of 1D systems?

1. Ground states: The DMRG method and beyond

Use MPS to find ground states of 1D Hamiltonians.

Most important method: The "Density Matrix Renormalization Group" (DMRG) method — in its modern interpretation as a variational method over MPS.

a) Idea & basic algorithm

i) Given $H = \sum h_i$ OBC, local, 1D.

Use OBC MPS ansatz

$$|\psi\rangle = \sum A^{i_1,(1)} A^{i_2,(2)} \cdots A^{i_N,(N)} |i_1,\dots,i_N\rangle$$

for ground state, and optimize the $A^{i_k,(k)}$ such as to minimize the energy

$$\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} .$$

ii) To optimize the $A^{i_k,(k)}$, pick <u>one</u> k_0 (the "working site"),

$A^{i_{k_0},(k_0)} \equiv X^{i_{k_0}}$, keep all other $A^{i_k,(k)}$ fixed, and minimize

$$\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \equiv \frac{\langle\psi[X]|H|\psi[X]\rangle}{\langle\psi[X]|\psi[X]\rangle} \quad \text{as}$$

a function of X.

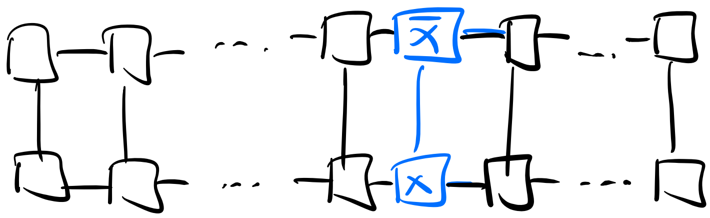
iii) What is the form of $\langle\psi[X]|H|\psi[X]\rangle$ and $\langle\psi[X]|\psi[X]\rangle$ as a function of X?

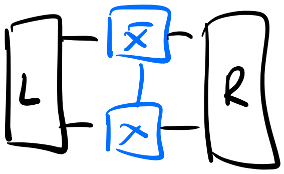
$$|\psi[X]\rangle = \sum A^{i_1(1)} \cdot \ldots \cdot X^{i_{k_0}} \cdot \ldots \cdot A^{i_N(N)} |i_1, \ldots, i_N\rangle$$

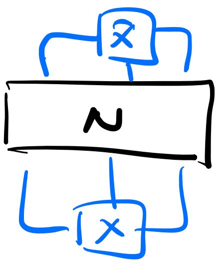
is <u>linear</u> in X $\Longrightarrow$

$\langle\psi[X]|H|\psi[X]\rangle$ and $\langle\psi[X]|\psi[X]\rangle$ are <u>quadratic</u> in X!

<u>More explicitly:</u>

$\langle\psi[X]|\psi[X]\rangle =$ 

$=$ 

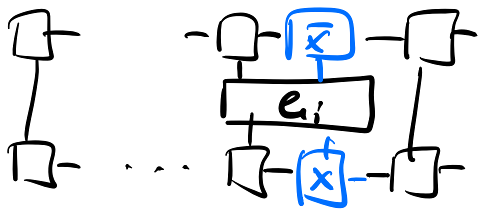
$=$ 

$$= \vec{x}^{\dagger} N \vec{x},$$

with $\vec{x}$ the "vectorized version" of X (i.e. a vector w/ components $(X^{i}_{\alpha\beta})_{i,\alpha,\beta}$).

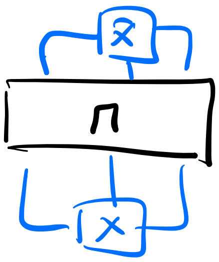
Similarly,

$\langle\psi[X]|H|\psi[X]\rangle = \dots +$



....

$=$

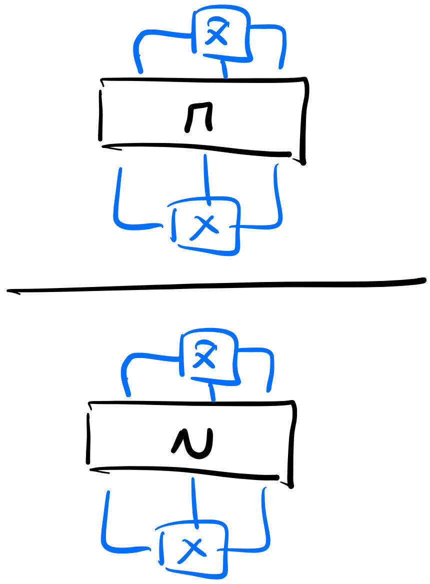


sum of ~N Ham. terms!

$$= \vec{x}^{\dagger} M \vec{x}.$$

iv) The minimization problem in (ii) is thus of the form

$$E_{min} = \min_{X} \frac{\vec{x}^{\dagger} M \vec{x}}{\vec{x}^{\dagger} N \vec{x}}$$

$= \min_{x}$ 

By redefining $\vec{y} = \sqrt{N}\,\vec{x}$ $(N \geq 0\,!)$,

this gives

$$E_{min} = \min_{y} \frac{\vec{y}^{\dagger}\left(\frac{1}{\sqrt{N}}\, H\, \frac{1}{\sqrt{N}}\right)\vec{y}}{\vec{y}^{\dagger}\cdot\vec{y}}$$

$\Rightarrow$ minimum E_{min} is given by <u>smallest</u> <u>eigenvalue</u> of $\frac{1}{\sqrt{N}}\, H\, \frac{1}{\sqrt{N}}$ & corresp. <u>eigenvector</u> !

$$E_{min}\,\vec{y}_{opt} = \frac{1}{\sqrt{N}}\, H\, \frac{1}{\sqrt{N}}\,\vec{y}_{opt}$$

$\sqrt{N}\,\vec{X} \qquad\qquad = \vec{X}$

$$\Longleftrightarrow \quad E_{min}\, N\, \vec{X}_{opt} = \Pi\, \vec{X}_{opt}$$

$\boxed{\text{"generalized eigenvalue problem"}}$

Can be solved efficiently!

v) We can now ~~move~~ back and forth ("sweep") through the system and for each working site k_0 replace $A^{i_{k_0},(k_0)}$ by the optimal $A^{i_{k_0},(k_0)}$ — i.e. the one which minimizes the energy.

This procedure will only decrease the energy. Remarkably, typically it will converge to a very good approx. of the ground state after a few sweeps! (Sometimes w/ some tricks...)

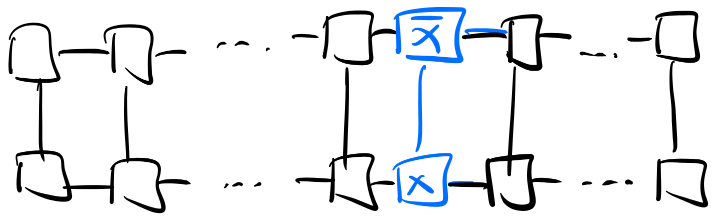
This is the <u>essence of the DMRG algorithm</u>.

<u>vi)</u> There are a number of <u>key optimizations</u>:

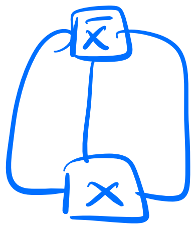
- Work in the <u>mixed gauge</u> around k_0:

Then,

$$\langle \psi[X] | \psi[X] \rangle =$$



$$=$$



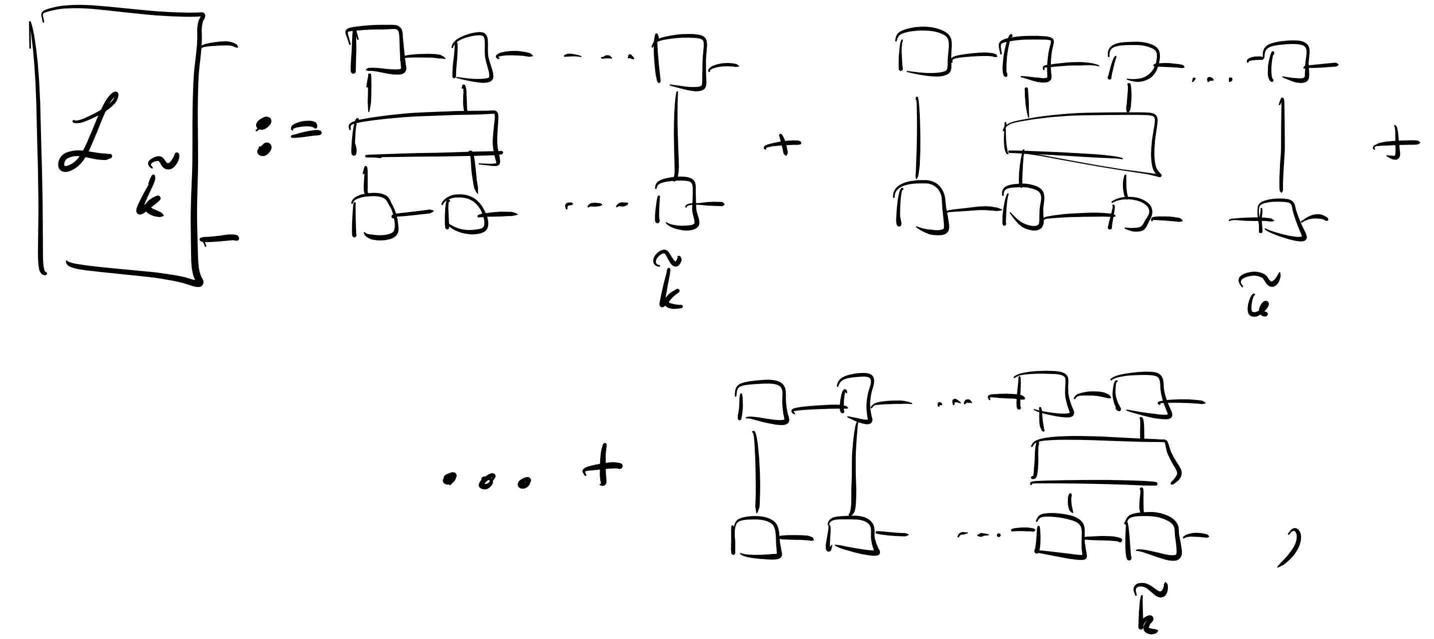
$$= \vec{X}^{\dagger} \vec{X} \, ,$$

i.e. the minimization in (iv) becomes a <u>normal</u> eigenvalue problem

$$E_{min} \vec{X} = M \vec{X} .$$

(This is <u>essential</u> for the stability of the method, as N typ. becomes ill-conditioned $\rightarrow$ this is a big drawback of the PBC algorithm!)

- When moving $k_0 \to k_0 \pm 1$ (one step in a forward/backward sweep) after optimizing the tensor $A^{i_{k_0},(k_0)}$, the canonical form can be updated with $O(D^3)$ operations (indep. of N!), since only $A^{i_{k_0},(k_0)}$ has been changed, and only $A^{(k_0)}$ and the adjacent tensor need to be updated.
- For the Hamiltonian, we can pre-compute for <u>all</u> cuts $\tilde{k}$ left of the working site k_0,



and similarly $R_{\tilde{k}}$ from the right.

Then, $\langle \psi[X] | H | \psi[X] \rangle$ can be computed in $O(1)$ operations (indep. of N),

and after updating $A^{i_{k_0}, (k_0)}$ and moving to the right (left), L_{k_0} (R_{k_0}) can be efficiently computed from L_{k_0-1} (R_{k_0+1}) with $O(1)$ operations.

<u>vii)</u> With these optimizations, the entire procedure takes $O(D^3)$ per step, and thus $O(ND^3)$ operations per sweep.

If # sweeps $\sim$ constant (often the case), the total effort scales as $O(ND^3)$ (and if $D \sim \mathrm{poly}(\frac{N}{\epsilon})$, as $O(\mathrm{poly}(\frac{N}{\epsilon}))$ itself).

This is the heart of the <u>DMRG algorithm</u>!

b) MPO encoding of Hamiltonian

Keeping track of the different Hamiltonian terms & updating them is technically challenging.

Better: Express Hamiltonian as a Matrix Product Operator.

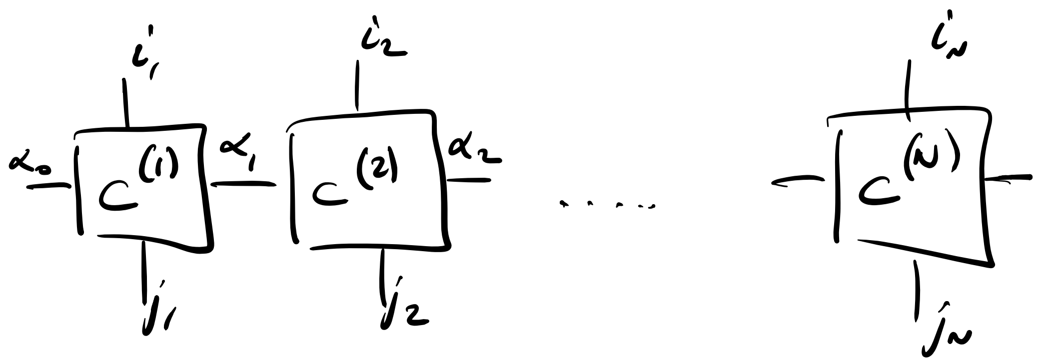
Matrix Product Operators

Definition: A Matrix Product Operator (MPO) is an operator $O : (\mathbb{C}^d)^{\otimes N} \to (\mathbb{C}^d)^{\otimes N}$ of the form

$$O = \sum_{\substack{i_1,\dots,i_N \\ j_1,\dots,j_N}} \operatorname{tr}\left[C^{i_1 j_1 (1)} C^{i_2 j_2 (2)} \cdots C^{i_N j_N (N)} \right] |i_1,\dots,i_N\rangle\langle j_1,\dots,j_N|$$

where the $C^{i_k j_k (k)}$ are $D_{k-1} \times D_k$ - matrices,

or

$O =$ 

.

(In essence, an MPS with 2 phys. indices per site, where one is the ket & one the bra of O.)

MPOs can be used to describe density operators (MPDOs), unitaries (MPUs), or Hamiltonians.

Hamiltonians as MPOs

Local Hamiltonians can be naturally expressed as MPOs.

General construction: Homework (# ...)

Here, we give some examples.

Example 1:

$$H_{Ising} = -\sum \sigma_z^i \sigma_z^{i+1} - h \sum \sigma_x^i \quad \text{(Ising model)}$$

thev. OBC MPO.

Construction: Use "agent" picture.

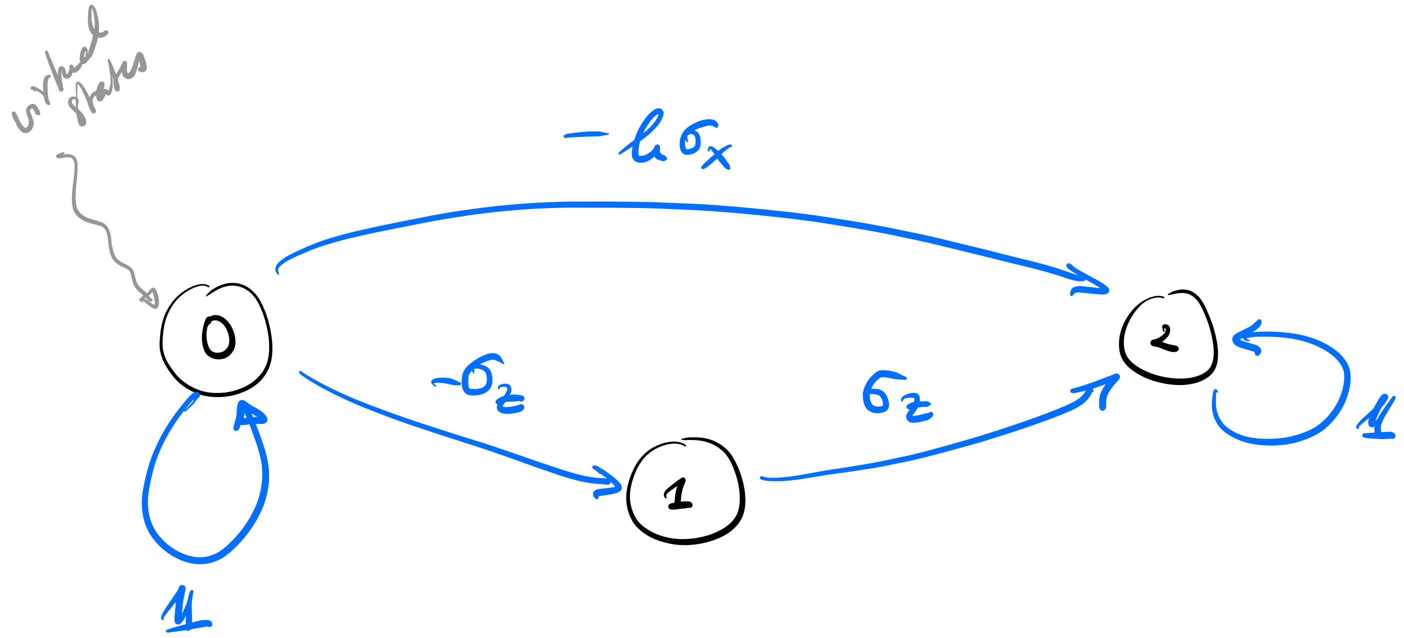
Start on left in 0, end up in 2.

On the way, implement <u>exactly once</u>

<u>either</u> $-h \cdot \sigma_x$,

<u>or</u> $(-\sigma_z)$, immediately followed by σ_z,

and <u>everywhere else</u> $\mathbb{1}$.



arrows: possible transition induced by MPO tensor & corresp. "physical state" (i.e., operation on that site)

Encode this into MPO tensors:

physical indices

$$\left(C^{ij}_{\underbrace{00}}\right)_{ij} = \mathbb{1} \quad ; \quad \left(C^{ij}_{22}\right)_{ij} = \mathbb{1}$$

virtual indices

$$\left(C^{ij}_{02}\right)_{i'j'} = -h\sigma_x \, ;$$

$$\left(C^{ij}_{01}\right)_{i'j'} = -\sigma_z \, ; \quad \left(C^{ij}_{12}\right)_{i'j'} = \sigma_z \, .$$

and zero otherwise.

Or shorthand notation – use virtual indices as <u>matrix indices</u> and put physical action as <u>matrix entry</u>:

$$C = \left(\begin{array}{c|c|c} \mathbb{1} & -\sigma_z & -h\sigma_x \\ \hline 0 & 0 & \sigma_z \\ \hline 0 & 0 & \mathbb{1} \end{array}\right)$$

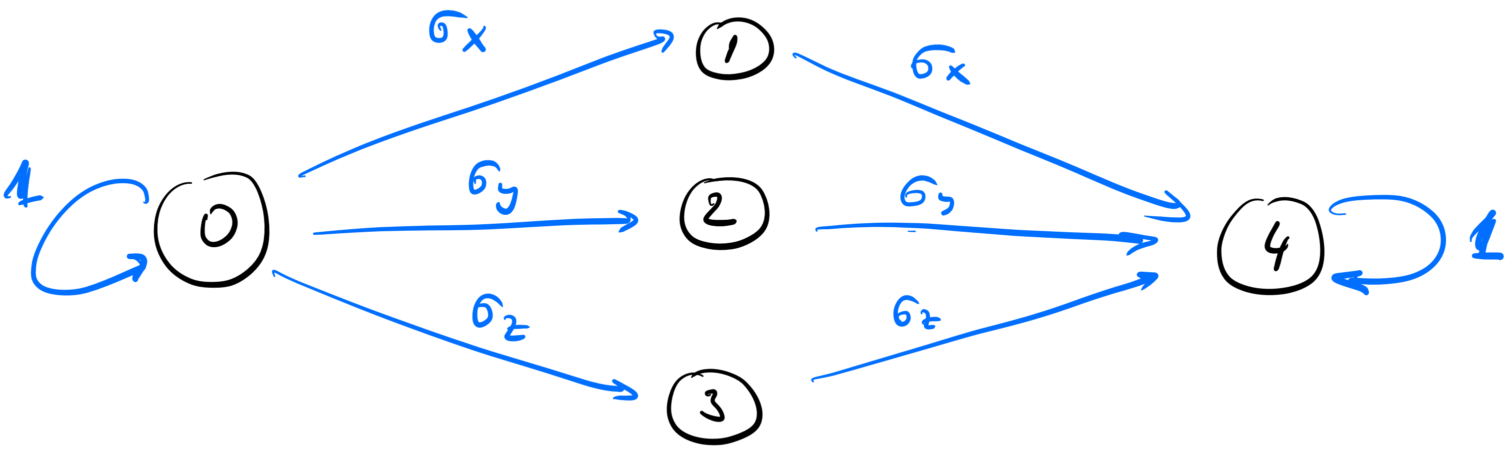
& choose $\langle 0|$ and $|2\rangle$ as boundary conditions:

$$H_{Ising} = \langle 0| \, \text{—}\boxed{C}\text{—}\boxed{C}\text{—} \cdots \text{—}\boxed{C}\text{—} \, |2\rangle .$$

Example 2:

$$H_{Heis} = \sum \vec{\sigma}^{\,i} \cdot \vec{\sigma}^{\,i+1}$$

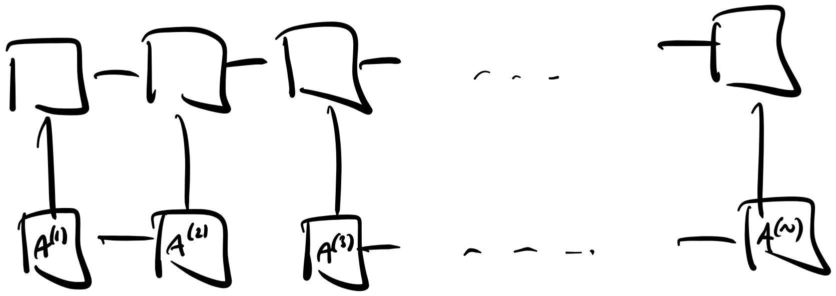
$$= \sum \left(\sigma_x^{i} \sigma_x^{i+1} + \sigma_y^{i} \sigma_y^{i+1} + \sigma_z^{i} \sigma_z^{i+1} \right)$$



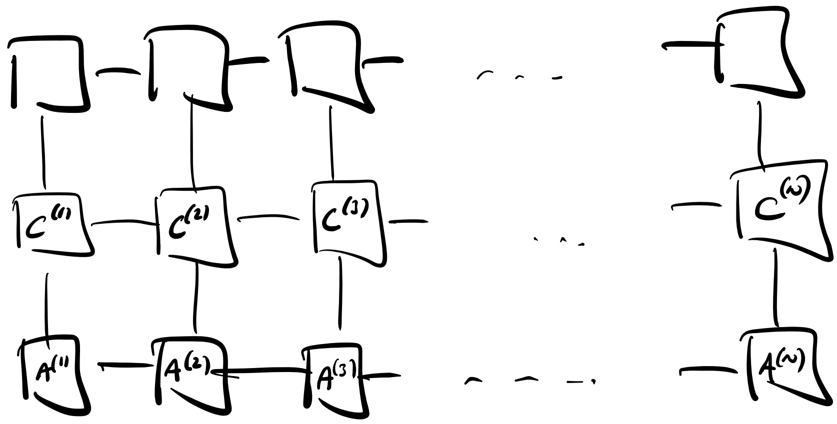
$$C = \begin{pmatrix} \mathbb{1} & \sigma_x & \sigma_y & \sigma_z & 0 \\ & & & & \sigma_x \\ & & & & \sigma_y \\ & & & & \sigma^z \\ & & & & \mathbb{1} \end{pmatrix} .$$

c) DMRG with MPOs

- Normalization:



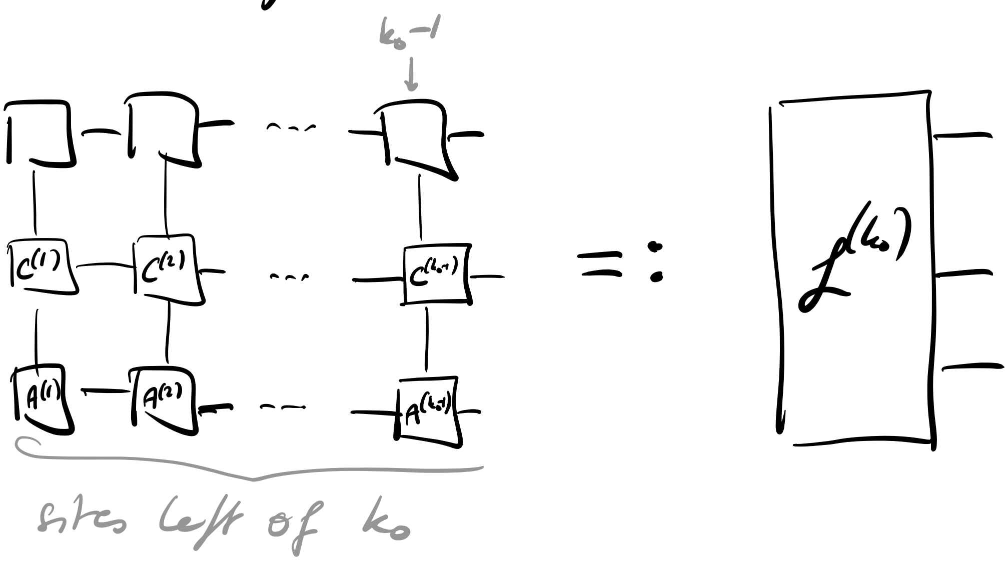
- Energy:



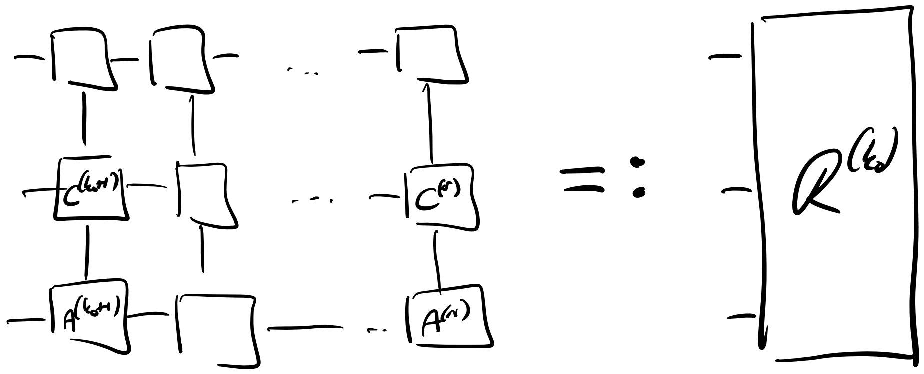
- contraction of the Hamiltonian:

Work in mixed gauge around a working site k_0.

For the working site k_0, define



and

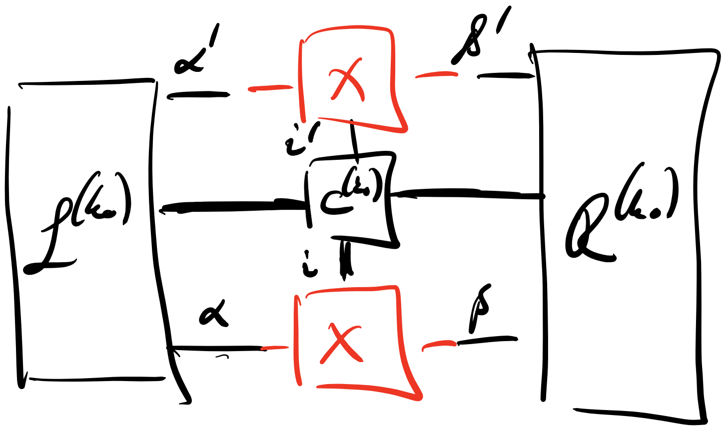


Then, $\mathcal{L}^{(k_0)}$ and $R^{(k_0)}$ can be updated in O(1) operations when moving k_0, as long as we store $\mathcal{L}^{(k)}$ and $R^{(k)}$ $\forall k$ left/right of k_0.

- <u>Optimization:</u>

  With CF around working site k_0, we need to minimize

$E(X) =$

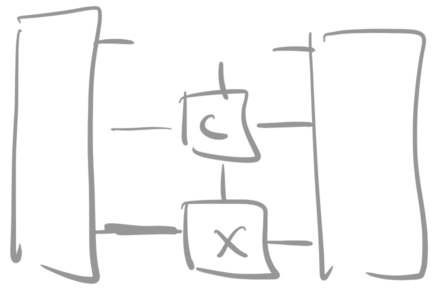


This can be solved by solving the eigenvalue problem (leading eigenvalue) of the linear map $(\alpha, \beta, i) \longmapsto (\alpha', \beta', i')$.

(Note: This is a $dD^2 \times dD^2$ matrix

$\Rightarrow$ diagonalization requires $d^3 D^6$ operations.

But we can use a Krylov method, where we just apply

$X \longmapsto$ 

$\Rightarrow$ scales as $O(D^3)$.)

$\Rightarrow$ solve eigenvalue problem & set $A^{(k_0)}$ to the eigenvector with smallest eigenvalue.

Note: The eigenvalue automatically returns the current total energy.

- Full algorithm:

As before: initialize with $k_0 = 1$, and then do

i) right-sweeps:
optimize at k_0
$k_0 \rightarrow k_0 + 1$
repeat until $k_0 = N$.

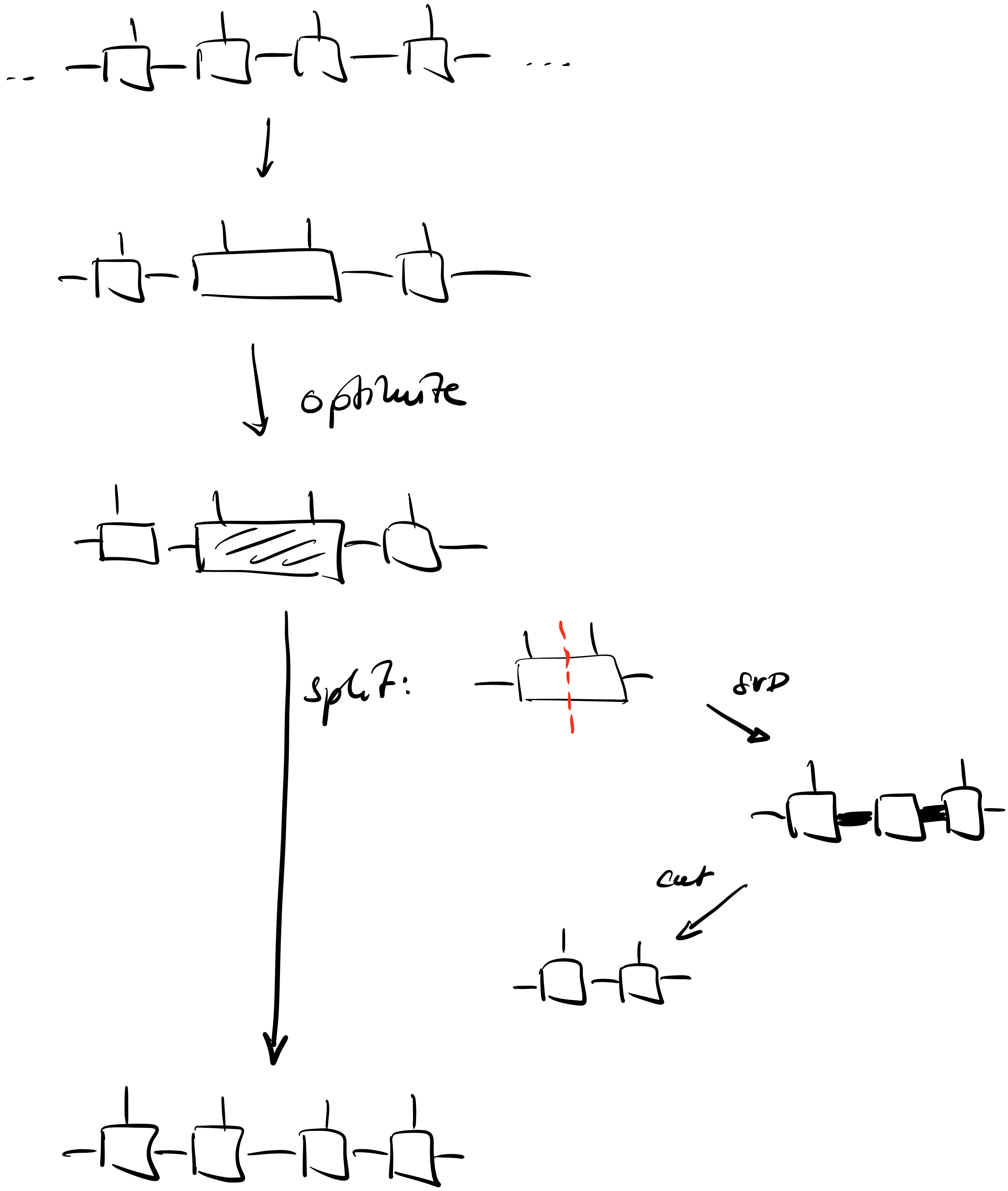
ii) left-sweeps:
optimize at k_0
$k_0 \rightarrow k_0 - 1$
repeat until $k_0 = 1$.

until convergence of energy is reached.

d) Two-site DMRG

A refinement of the method is 2-site DMRG:

There, one groups two sites and optimizes over the joint tensor at 2 sites, which is then split using an SVD:



This allows the optimization to explore a bigger space, and is less likely to get stuck in local minima.

2. Other MPS-based algorithms

Brief overview of other MPS-based algorithms

a) Time evolution (real/imaginary)

Can we use MPS to simulate real/imag. time evolution:

$$|\psi(0)\rangle \mapsto |\psi(t)\rangle = e^{-iHt}|\psi\rangle\text{, or}$$

$$|\psi(0)\rangle \mapsto |\psi(t)\rangle = e^{-H\tau}|\psi\rangle\text{,}$$

where $|\psi(0)\rangle$ is some MPS?

Trotter expansion

Consider e.g. NN Ham. $H = \sum h_i$. ← N.N. terms

$$H = \sum h_i = \underbrace{\sum_{i\,\text{even}} h_i}_{=: H_{even}} + \underbrace{\sum_{i\,\text{odd}} h_i}_{=: H_{odd}}$$

$$e^{-iHt} = e^{-i(H_{even} + H_{odd})t}$$

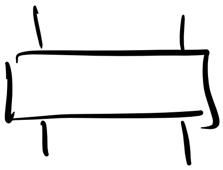
$$= \left(e^{-i(H_{even}+H_{odd})\,t/K}\right)^K$$

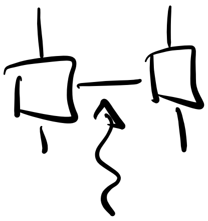
$$\approx \left(e^{-iH_{even}\,t/K}\, e^{-iH_{odd}\,t/K}\right)^K$$

Can get better accuracy w/ higher-order expansion!

$$= \underbrace{\left(\prod_{i\ even} e^{-ih_i\,t/K} \prod_{i\ odd} e^{-ih_i\,t/K}\right)^K}_{\circledast}$$

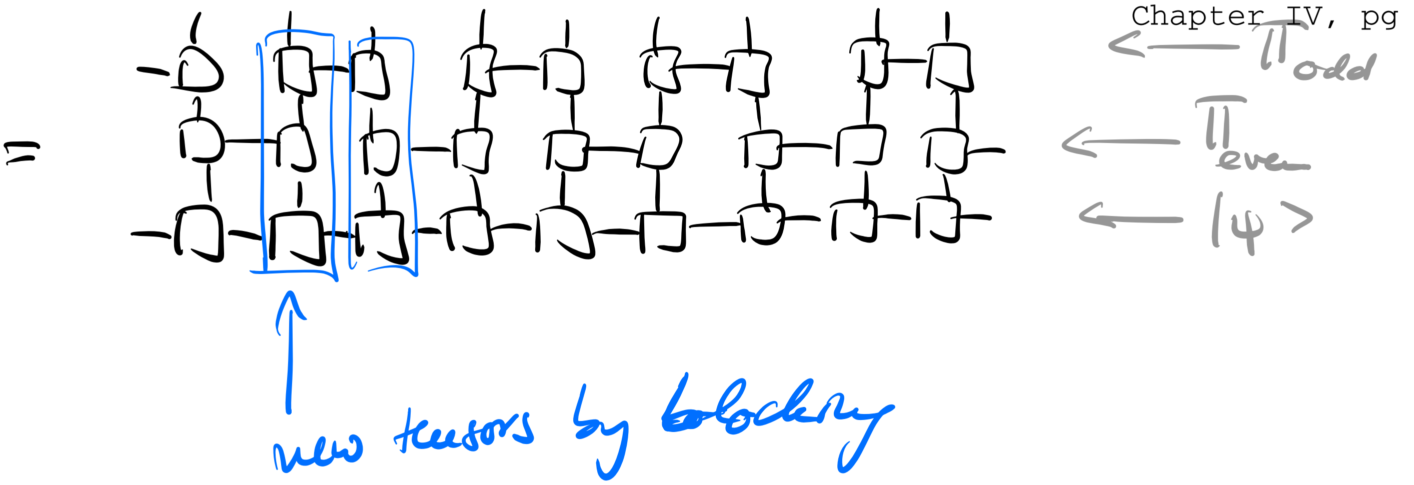
Tensor network formulation

$e^{-ih_i\,t/K}$ =  ← operator acting on 2 sites

=  ← TN decomposition (via SVD/QR)

dim. χ

$$\Rightarrow |\psi(t/K)\rangle = \left(\prod_{i\ even} e^{-ih_i\,t/K} \prod_{i\ odd} e^{-ih_i\,t/K}\right) |\psi(0)\rangle$$



$\Rightarrow$ $|\psi(t/k)\rangle$ is an MPS with an enlarged bond dimension $D\chi$.

$\Rightarrow$ Can now use some truncation scheme (e.g. as in II.1, or by maximizing the overlap with an MPS with bond dim. D – this can be done analogous to DMRG) to get the bond dim. back to D.

(In fact, for *real* time evolution e^{-iHt}, the truncation can be done *entirely local* if we use a suitable canonical form: the **TEBD algorithm** ("time evolving block decimation").

Can be used as an alternative method for finding ground states, using $|\psi\rangle \mapsto e^{-H\tau}|\psi\rangle$, or for simulating time evolution.

<u>Caveat</u>: The entanglement in time evol. can grow <u>linearly</u> in time (i.e., D grows <u>exponentially</u> in time), if initial state has finite energy density

$\Longrightarrow$ simulation becomes quickly inaccurate (can be seen from truncation error, or by evolving back again & checking for consistency).

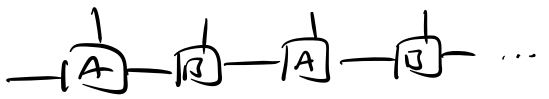
<u>Note</u>: Truncation errors don't matter for imag. time evol., in case we want to reach the ground state!

b) Infinite systems

Can we simulate <u>infinite</u> systems (with a t.inv. Hamiltonian $H = \sum h_i$)?

all equal.

<u>Ansatz:</u> t.inv. MPS (possibly with larger unit cell,

e.g. ).

– also termed iMPS ("infinite MPS") in this context.

Can again either use <u>variational methods</u> (as DMRG) or real/imag. <u>time evolution</u> methods.

<u>Variational methods:</u>

- <u>iDMRG</u>: Start with 2 sites, optimize, and add a site in the middle, which is optimized next, and so on. No sweeps – the "old" tensor are pushed to the outside. The tensors in the centers should converge to the iMPS tensor.

- VUMPS (variational uniform MPS): Formulate the energy optimization problem directly in the thdyn. limit, by using a canonical form for the iMPS around a working site.

  In essence, the problem is linearized by keeping all but the central tensor fixed — this gives a quadratic problem, which is then solved. The new tensor is updated everywhere in a neat way (respecting the canonical form).

- more generally, we can linearize the system, by "keeping all but one A fixed", and then either minimize that A (and then insert it everywhere), or just move along the gradient.

Time evolution (real/imaginary):

- TEBD (for real time evolution)

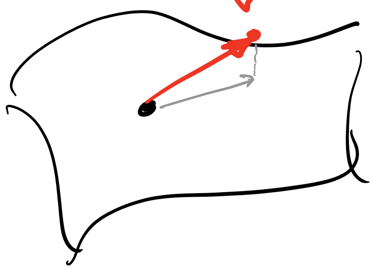
  can also be used in a therm. setting

  TDVP (time-dependent variational principle)

  Consider the space of iMPS w/ bond dim. D as a manifold.

  Consider evolution

$$|\psi(t+\delta t)\rangle = e^{-iH\delta t}\,|\psi(t)\rangle$$



evolution leads out of manifold

$\Downarrow$

find best projection back onto manifold.

"improved" gradient method, which takes geometry of space into account.