III. 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

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

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, $A^{i_{k_0},(k_0)} \equiv X^{i_{k_0}}$, 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.

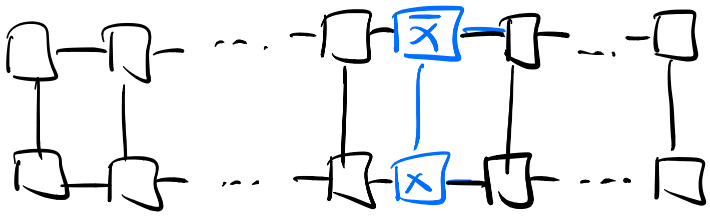
iii) What is the form of $\langle\psi[X]|H|\psi[X]\rangle$ and $\langle\psi[X]|\psi[X]\rangle$ as a function of X?

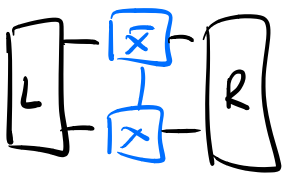
$$|\psi[X]\rangle = \sum A^{i_1,(1)} \cdot \ldots \cdot X^{i_{k_0}} \cdot \ldots \cdot A^{i_N,(N)} |i_1, \ldots, i_N\rangle$$

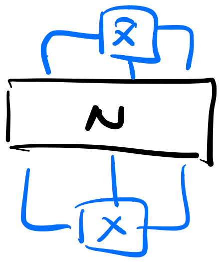
is <u>linear</u> in X $\Longrightarrow$

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

<u>More explicitly:</u>

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

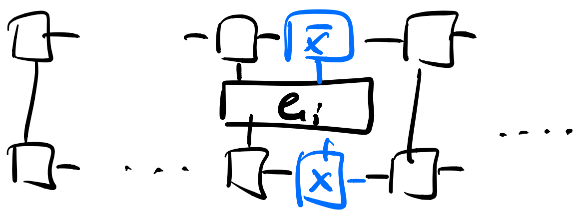
$=$ 

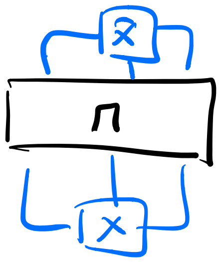
$=$ 

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

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

Similarly,

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

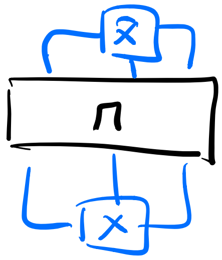
$=$  sum of $\sim N$ Ham. terms!

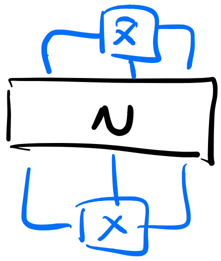
$$= \vec{x}^\dagger M \vec{x}.$$

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 <u>redefining</u> $\vec{y} = \sqrt{N}\,\vec{x}$ $(N \geq 0\,!)$,
this gives

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

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

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

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

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

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

Can be solved efficiently!

v) We can now ~~move~~ back and forth ("sweep") through the system and for each position 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.

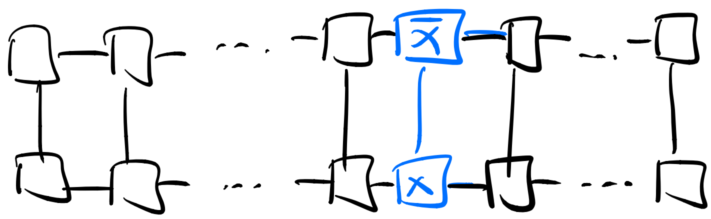
This procedure will only decrease the energy. Remarkably, typically it will converge to a very good approx. of the ground state after a few sweeps! (Sometimes w/ some tricks...)

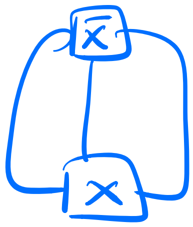
This is the essence of the DMRG algorithm.

vi) A number of optimizations are useful:

- Work in the mixed gauge around k_0:

Then,

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

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

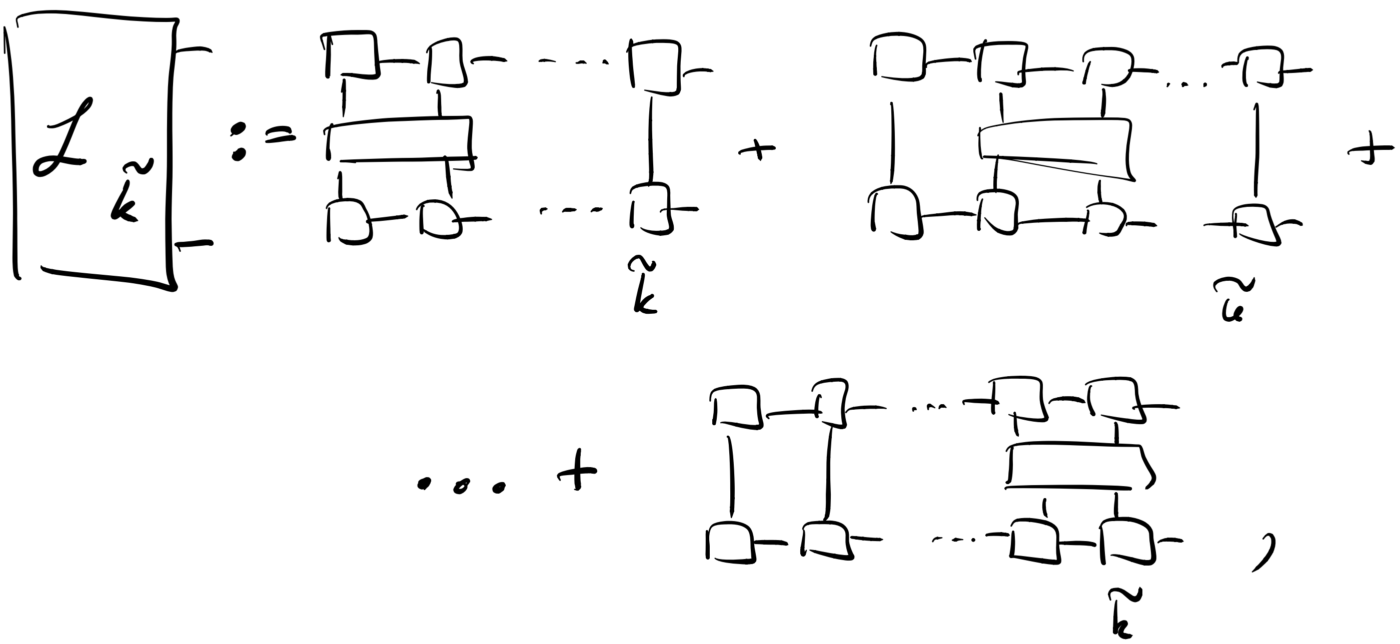
i.e. the minimization in (iv) becomes a normal eigenvalue problem

$$E_{min}\,\vec{x} = M\,\vec{x}.$$

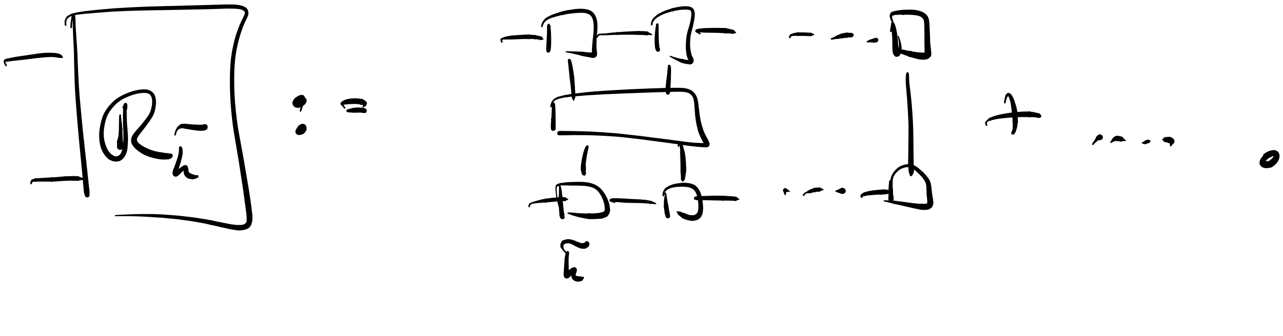
- When moving $k_0 \to k_0 \pm 1$ (one step in a forward/backward sweep) after optimizing the tensor $A^{i_{k_0},(k_0)}$, the canonical form can be updated with $O(D^3)$

operations (indep. of N!), since only $A^{i_{k_0},(k_0)}$ has been changed.

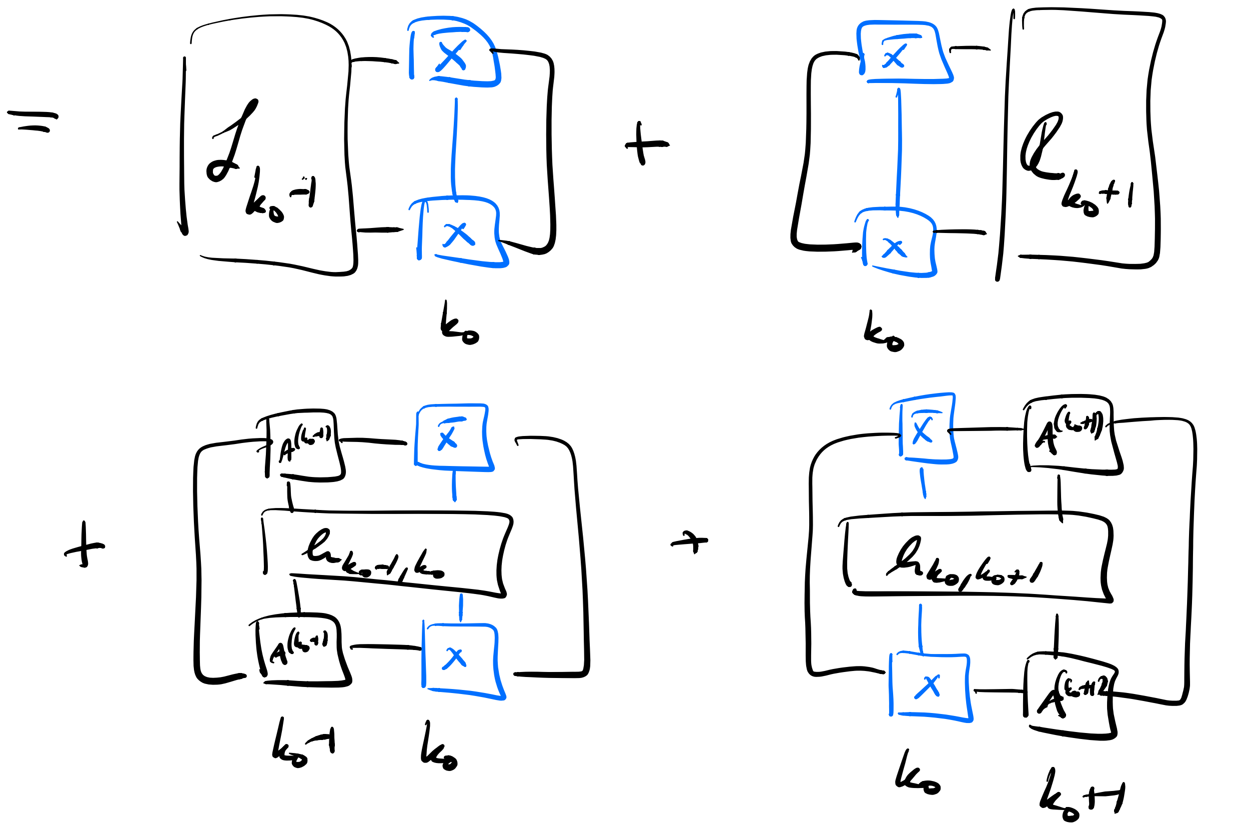
- For the Hamiltonian, we can pre-compute for all cuts k:



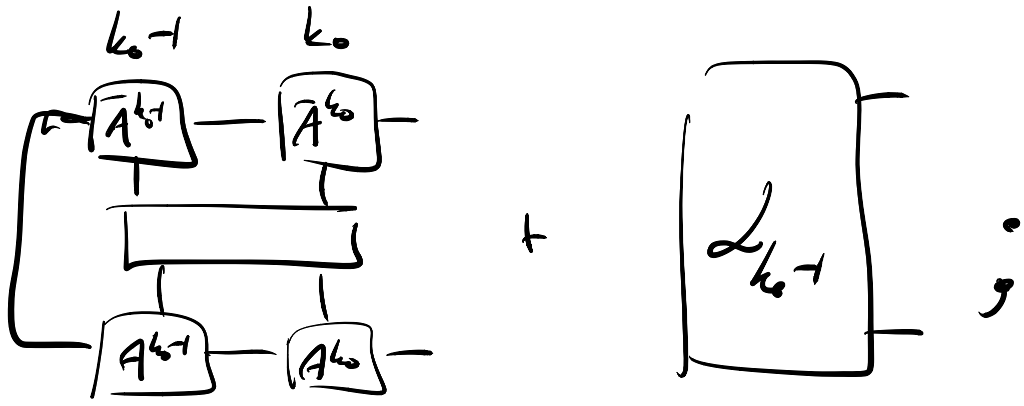
and similarly



Then, $\langle \psi[X] | H | \psi[X] \rangle = \vec{X} \mathcal{H} \vec{X} =$



i.e. Π can be computed efficiently, and after updating $A^{i_{k_0},(k_0)}$ and moving to the right (left), $\mathcal{L}_{k_0}$ $(\mathcal{R}_{k_0})$ can be efficiently computed from $\mathcal{L}_{k_0-1}$ $(\mathcal{R}_{k_0+1})$ as

$\mathcal{L}_{k_0} =$  ;

all other L_k & R_k remain unchanged,

$\Longrightarrow$ need to only update $O(1)$ terms,

and complexity <u>indep. of N</u>!

(similar to transfer op: D^3 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 $D \sim \mathrm{poly}\left(\frac{N}{\varepsilon}\right)$, as $O\left(\mathrm{poly}\left(\frac{N}{\varepsilon}\right)\right)$ 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.

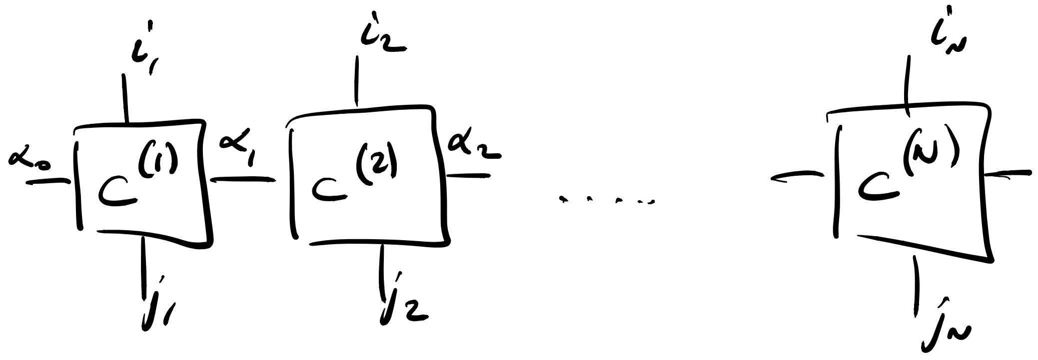
Matrix Product Operators

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

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

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:

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

trav. OBC MPO.

Construction: Use "agent" picture.

Start on left in 0, end up in 2.

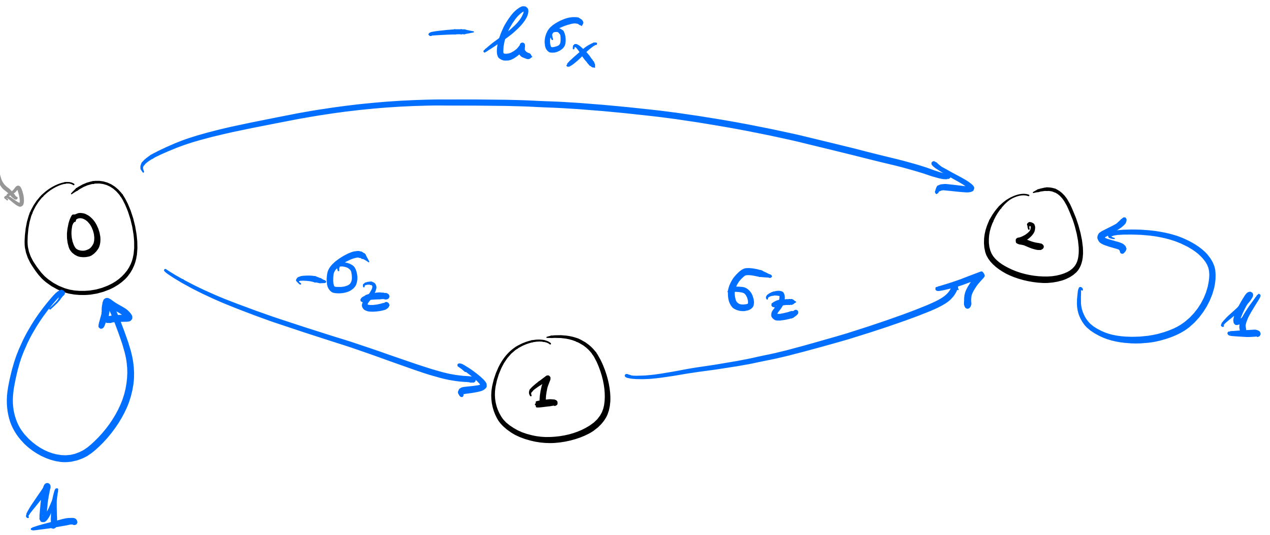
On the way, implement exactly once

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

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

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

virtual states



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

Encode this into MPO tensors:

physical indices

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

virtual indices

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

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

and zero otherwise.

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

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

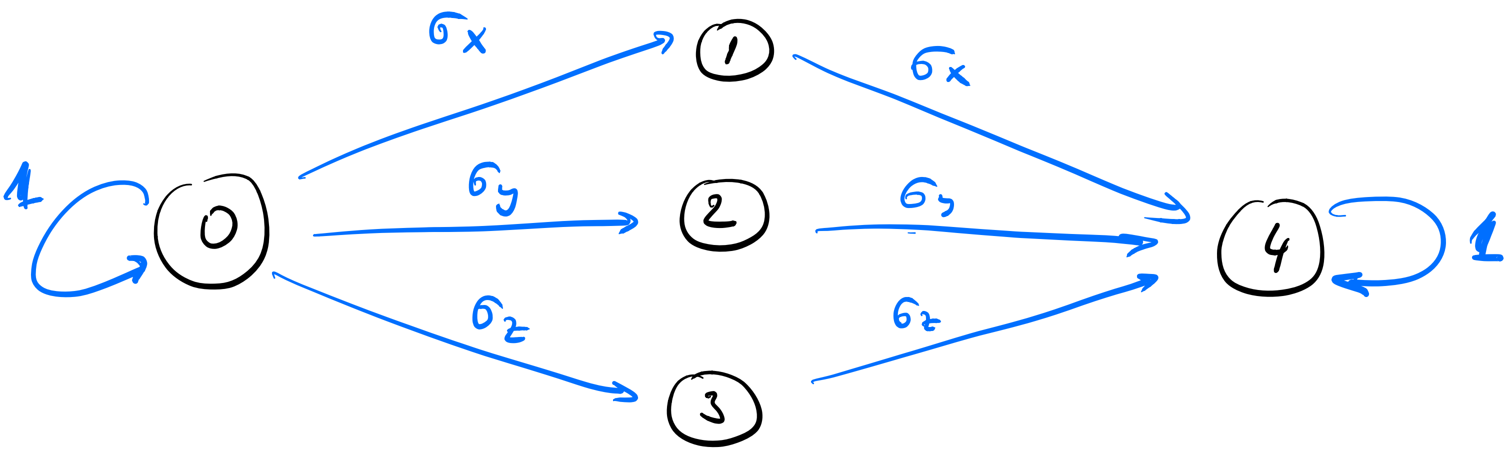
& choose $\langle 0|$ and $|2\rangle$ as boundary conditions:

$$H_{Ising} = \langle 0| \; C \; C \; \cdots \; C \; |2\rangle .$$

Example 2:

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

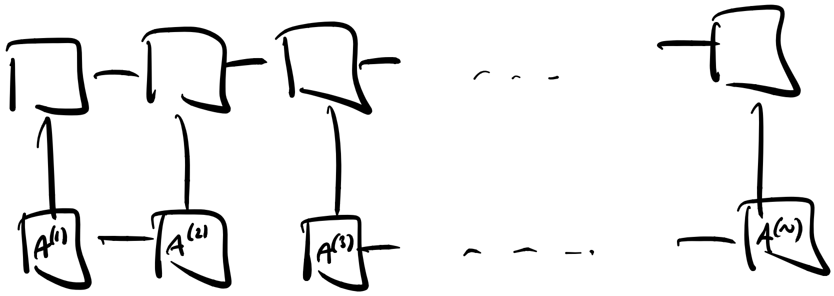
$$= \sum \left(\sigma_x^i \sigma_x^{i+1} + \sigma_y^i \sigma_y^{i+1} + \sigma_z^i \sigma_z^{i+1} \right)$$



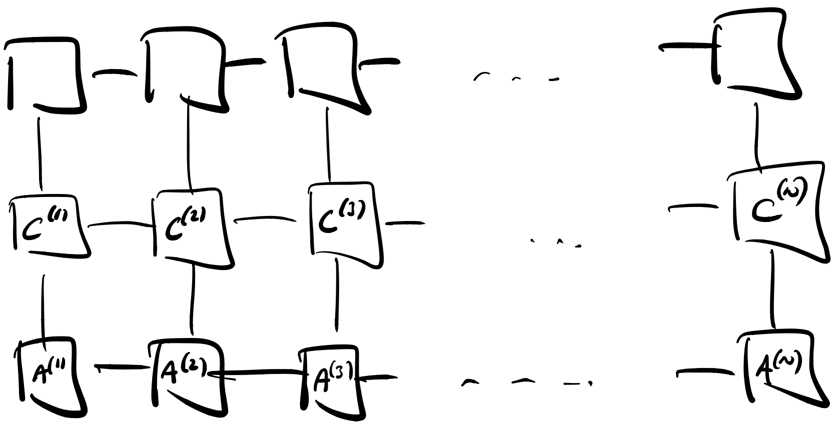
$$C = \left(\begin{array}{c|c|c|c|c} \mathbb{1} & \sigma_x & \sigma_y & \sigma_z & 0 \\ \hline & & & & \sigma_x \\ \hline & & & & \sigma_y \\ \hline & & & & \sigma^z \\ \hline & & & & \mathbb{1} \end{array} \right) .$$

c) DMRG with MPOs

- Normalization:



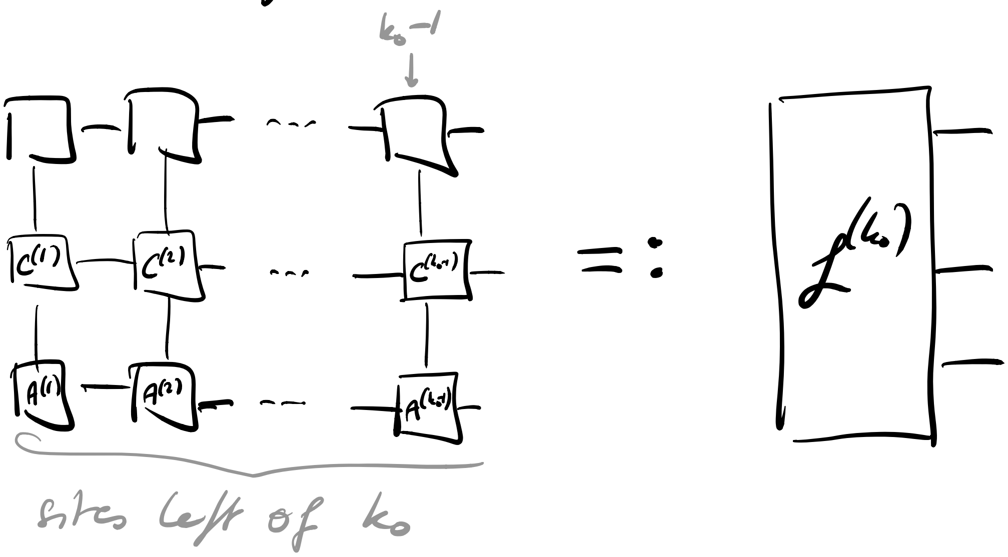
- Energy:



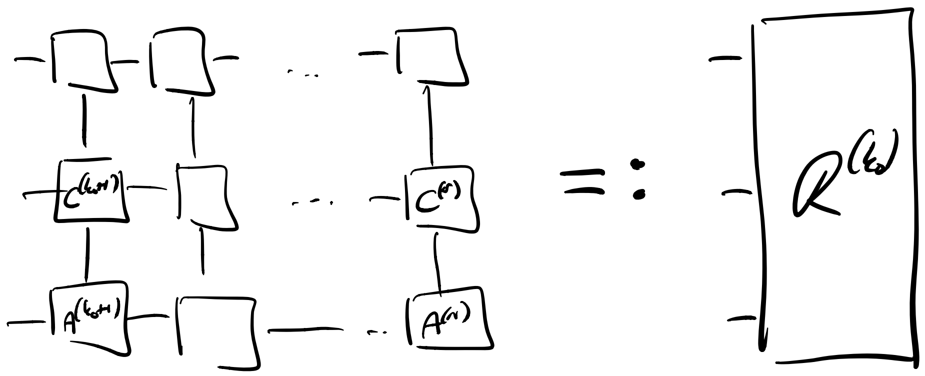
- contraction of the Hamiltonian:

Work in mixed gauge around a working site k_0.

For the working site k_0, define



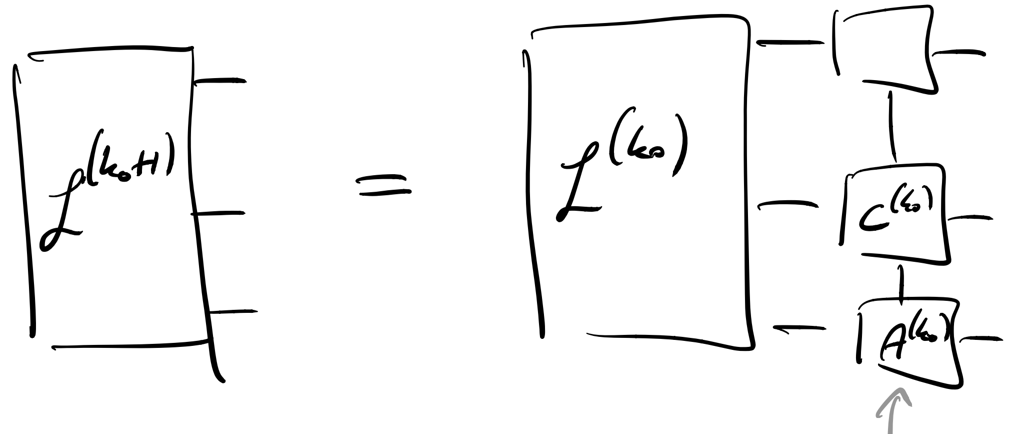
and



- <u>Move the working site</u>

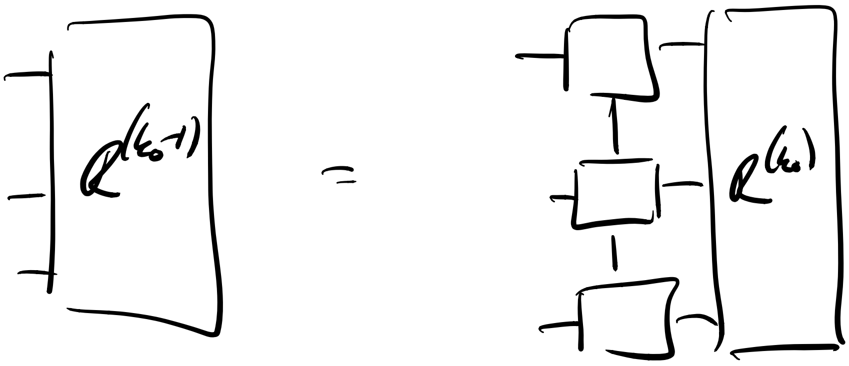
The working site can be moved right/left by <u>locally</u> updating the can. form.

If we store all $\mathcal{L}^{(k)}$ for $k \leq k_0$ and all $R^{(k)}$ for $k > k_0$, we only have to re-compute



<u>after</u> bringing it to CF!

for a right-move or



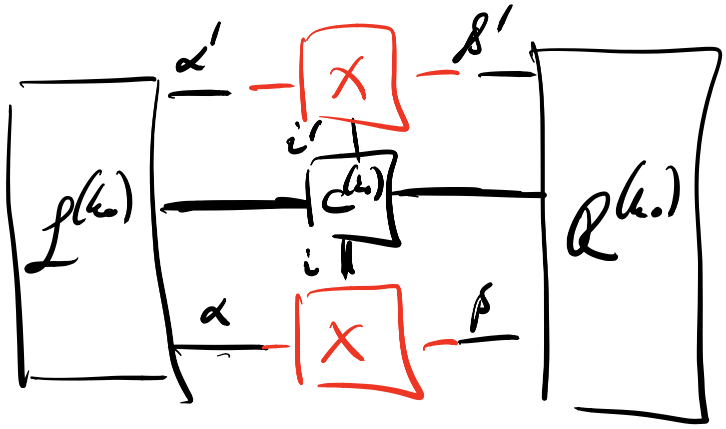
for a left-move.

- Initialization:

  The initialization at $k_0 = 1$ can be carried out by using the code for left-moving the gauge & $R^{(k)}$

- Optimization:

  With CF around working site k_0, we need to minimize

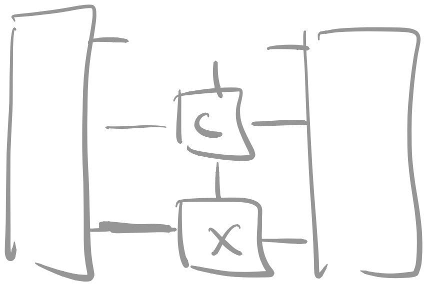
$E(X) =$ 

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 $\mathcal{O}(D^3)$.)

$\Rightarrow$ 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.

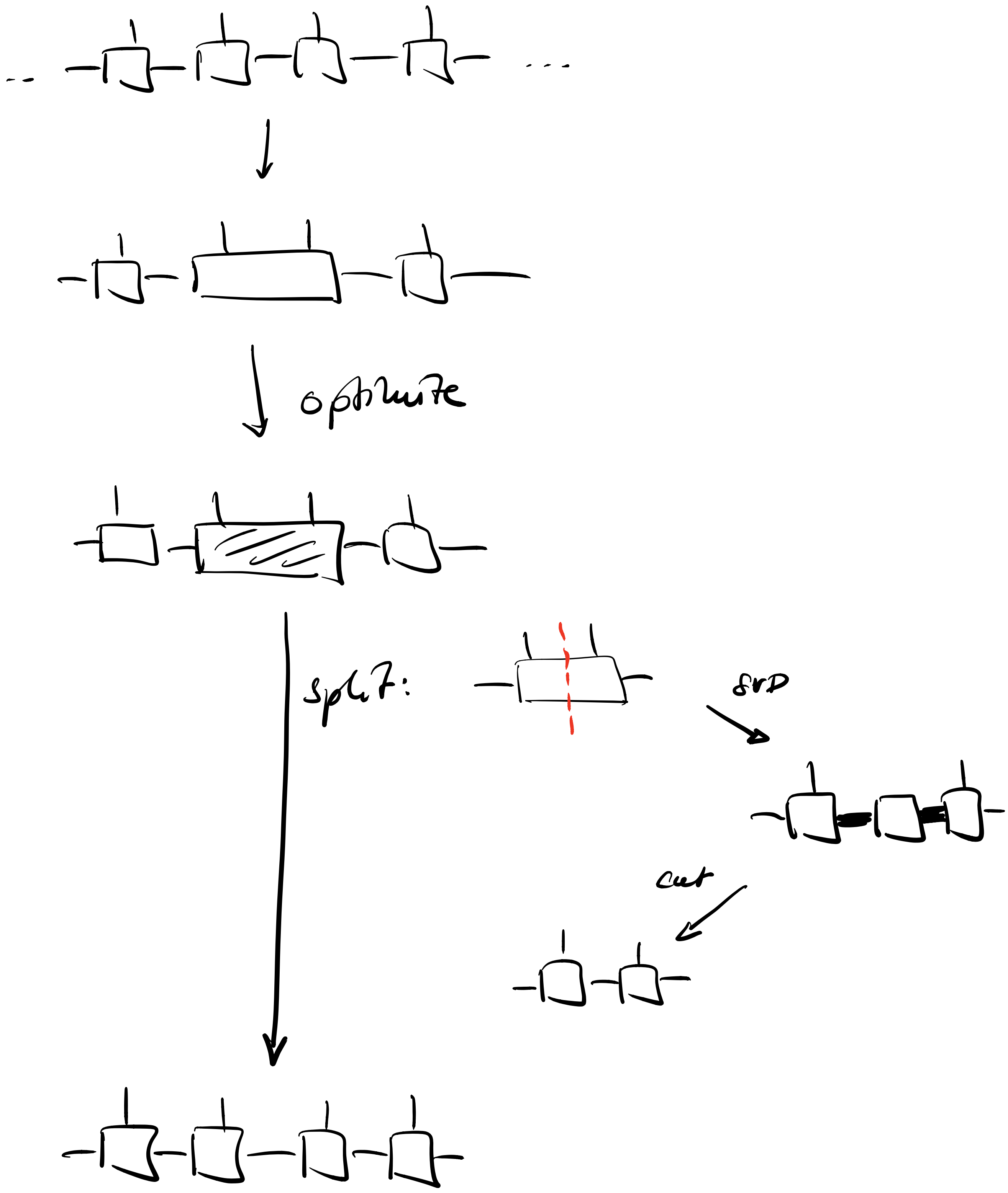
d) Two-site DMRG

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.