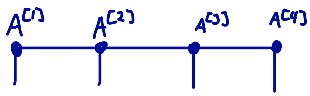
$\underline{\text{VI}}$ Tensor-Product States in 2D systems

Review of 1D MPS Simplified update Boundary MPS Tensor network RG

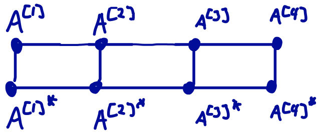
Review of 1D: Matrix-Product States (MPS)

$$|\psi\rangle = \sum_{\{\sigma_1,\dots,\sigma_N\}} \psi_{\sigma_1,\dots,\sigma_N} |\sigma_1,\dots,\sigma_N\rangle$$



$d^N \rightarrow N\chi^2 d$

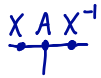
Simple to do calculations: $\langle\psi|\psi\rangle =$

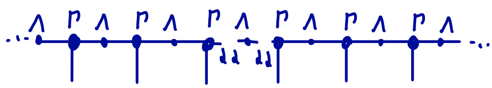


Uniform (infinite) MPS and canonical form

Use gauge degree of freedom:

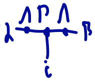
$\rightsquigarrow$





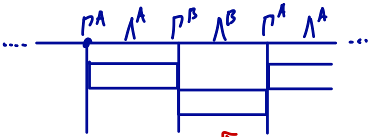
$$\equiv \sum_\alpha |\alpha\rangle_L \; \Lambda_\alpha \; |\alpha\rangle_R$$

$\rightsquigarrow |\psi\rangle = \sum_{\alpha\beta i}$

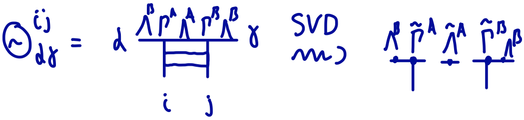


$|\alpha\rangle_L |i\rangle |\beta\rangle_R$

TEBD Algorithm

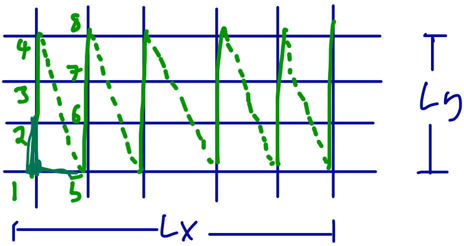


Trotter decomposition of $\mathcal{U}$



$\rightsquigarrow$ GS

How to go beyond 1D?



Mapping of 2D strip (cylinder) to 1D system with long range interactions!

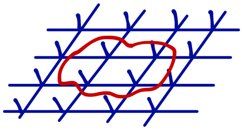
Area law $S(L_x, L_y) \sim L_y \rightsquigarrow \boxed{\chi \sim \exp(L_y)}$.

Generalization of MPS to capture the area law in higher dimensions

How to capture the 2D area?

$$|\psi\rangle = \sum \psi_{i_1 \dots i_N} |i_1 \dots i_N\rangle$$

$$\psi_{i_1 \dots i_N} =$$



Area law: $S \sim L \sim$ # „legs" cut

⇝ bond dimension independent of system size

Classical Ising $\quad H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$

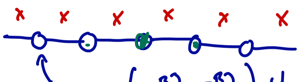
1D: Transfermatrix

$$H = -J \sum_{b \in \text{bonds}} \sigma_{b-\frac{1}{2}} \sigma_{b+\frac{1}{2}} \qquad \sigma_{b-\frac{1}{2}} \sigma_{b+\frac{1}{2}} = \begin{cases} -J & ++ \\ J & +- \\ J & -+ \\ -J & -- \end{cases}$$

$$e^{-\beta H} = e^{-\beta J \sigma_{\frac{1}{2}} \sigma_{\frac{3}{2}}} e^{-\beta J \sigma_{3/2} \sigma_{5/2}} \dots$$

$$Z = \sum_{\{\sigma\}} e^{-\beta H} = \langle \sigma_1 | e^{-\beta J \sigma_{\frac{1}{2}} \sigma_{\frac{3}{2}}} | \sigma_2 \rangle \langle \sigma_2 | e^{-\beta J \sigma_{3/2} \sigma_{5/2}} | \sigma_3 \rangle \langle \sigma_3 | \dots$$

$$Z = \mathrm{tr}\, T^N = (2\cosh \beta J)^N + (2\sinh \beta J)^N$$

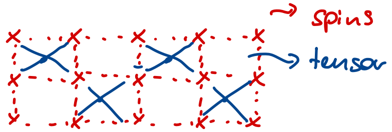


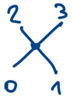
$$T = \begin{array}{c} \begin{pmatrix} e^{\beta J} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J} \end{pmatrix} \begin{array}{c} +1 \\ -1 \end{array} \\ \begin{array}{cc} +1 & -1 \end{array} \end{array}$$

$$\boxed{\lim_{N \to \infty} Z = (2\cosh \beta J)^N}$$

2D: Express the partition function as TPS

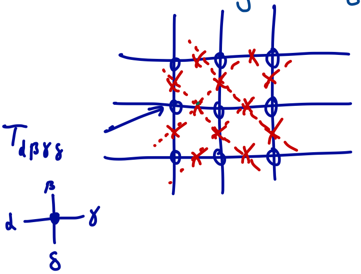
Consider lattice of spins:





$$z' = e^{\beta J(\sigma_0\sigma_1 + \sigma_2\sigma_3 + \sigma_0\sigma_2 + \sigma_1\sigma_3)} = T_{0123}$$

Rotate lattice by 45 degrees:



$$T_{++++} = T_{----} = e^{4\beta J}$$
$$T_{+---} = T_{-+--} = T_{-+++} \cdots = 1$$
$$T_{++--} = T_{-++-} = \cdots = 1$$
$$T_{+-+-} = T_{-+-+} = e^{-4\beta J}$$

⇝ Simple $\chi = 2$ tensors to express the partition function!

Challenge: Contract network → more and more free legs
⇝ scaling is exponential.

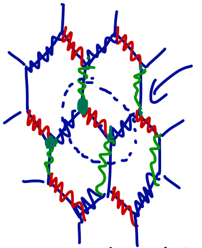
Solution: Start at the lower and upper boundary and represent boundary state as MPS with max bond dimension χ.

Contract network with TEBD until fixed point is reached (Details are given below for quantum systems).

Other options exist too!

2D TEBD

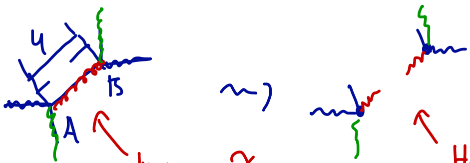
Trotter decomposition on a honeycomb lattice



$e^{-\delta H_{bond}}$ (imaginary time)

H_A, H_B, H_C

Use again translation invariance: Two tensors A, B



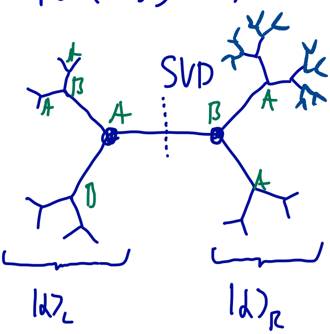
Increases χ

How to reduce χ?

What we want: Minimize $\| |\psi\rangle - |\psi_\chi\rangle \|$
(Full update is approximately doing that)

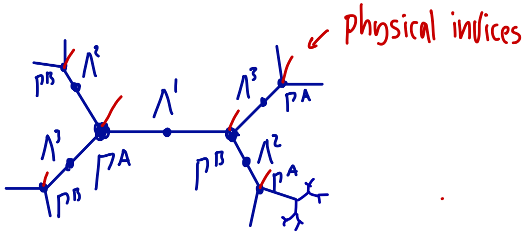
Simple update

Ignore the loops (Cayley tree)



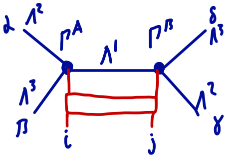
$$|\psi\rangle = \sum \Lambda_\alpha |\alpha\rangle_L |\alpha\rangle_R$$

Canonical form similar to 1D!



⇝ We have $\{ \Gamma^{i[A/B]}_{\alpha\beta\gamma} , \Lambda^{[1/2/3]}_{\alpha} \}$

Using this simplification, TEBD is straight forward!



$= \Theta^{ij}_{\alpha\beta\gamma\delta}$

SVD of $\Theta_{i\alpha\beta, j\gamma\delta}$ yields $U_{i\alpha\beta, \varepsilon} \tilde{\Lambda}_{\varepsilon} V_{\varepsilon, j\gamma\delta}$

Multiplying with inverses of Λ^2 and Λ^3 ⇝ $\tilde{\Gamma}^A$ and $\tilde{\Gamma}^B$!

Applying this iteratively to bonds 1,2,3, 1,2,3,... yields the GS TPS!

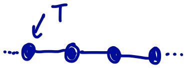
Crude approximation as the truncation is not optimal for the honeycomb lattice!

Contraction of TPS

Calculate $\langle\Psi|\Psi\rangle$!

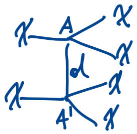
Recall: MPS 

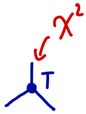
$= T_{\alpha\alpha',\beta\beta'}$

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

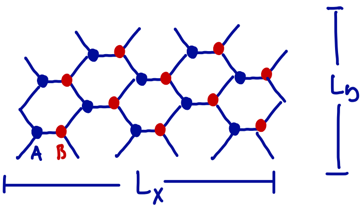
$\sim d\cdot\chi^3$

Now for TPS

$T_{\alpha\alpha',\beta\beta',\gamma\gamma'} =$ 

$\equiv$ 

(top view on r.h.s.)

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

$\sim \exp(L_x)$!

Exponentially hard to contract! $\rightsquigarrow$ Need further approximations.

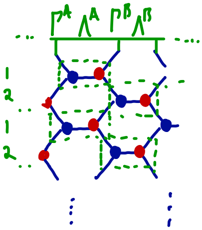
Boundary MPS

Contraction similar to a Trotterized time evolution:

$\rightsquigarrow$ Use the TEBD algorithm!

Example: $\langle\Psi|O|\Psi\rangle$ for infinite / translationally invariant system

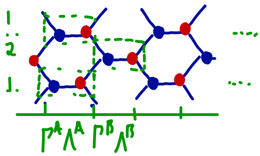
1) Use iTEBD to obtain fix point from the top:



$\rightsquigarrow |\Psi_{Top}\rangle$ (Complete on full cycle 1,2)

$\rightsquigarrow$ Parameter χ_{cut}

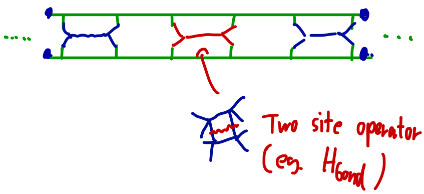
2) Use iTEBD to obtain fix point from the bottom:



$\rightsquigarrow |\Psi_{Bottom}\rangle$ (complete on half cycle 1)

$\rightsquigarrow$ Parameter χ_{cut}

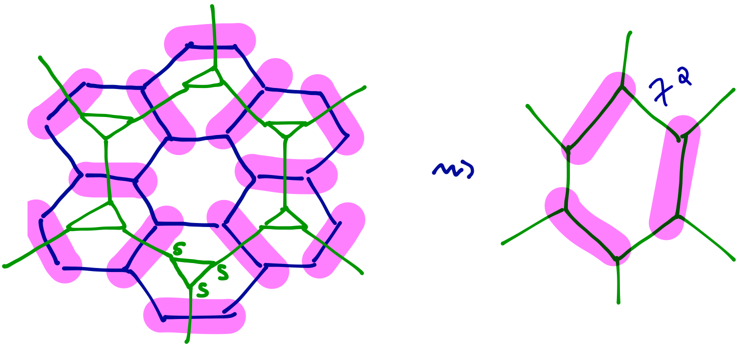
3) Calculate expectation values:

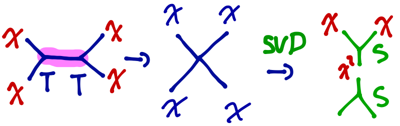
$\langle\Psi|O|\Psi\rangle =$ 

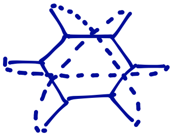
$\rightsquigarrow$ Works surprisingly well! (Compare also corner transfer matrix)

Tensor renormalization

Basic idea: Coarse grain the lattice and contract resulting tensors exactly.





$\langle\psi|\psi\rangle =$ 

Truncate the bond dimension at some χ_{cut}.
Repeat until fix point is reached!

$\rightsquigarrow$ Truncation is not optimal : χ_{cut} much larger than boundary MPS

See also: https://arxiv.org/abs/1412.0732