Title: LECTURE: Tensor networks and quantum algorithms

Speakers: Martin Ganahl

Collection: Quantum and AI Career Trajectories Mini-Course: Computational Methods and their Applications

Date: May 11, 2023 - 9:30 AM

URL: https://pirsa.org/23050100

Abstract: TBC

09:29 Donnerstag 11. Mai 56 %

Real/imaginary time evolution (TEBD) (Vidal)

Goal: compute the time evolution of a given initial MPS under a given Hamiltonian H: $|\psi(t)\rangle =$

Restriction: H is a nearest neighbor Hamiltonian $H =$

$\hat{h}_i$ ... acts on sites $i, i+1$ $h_i =$ 

Break t into n steps $\delta = \frac{t}{n}$

$$|\psi(t)\rangle = e^{-iHt}|\psi(t=0)\rangle = e^{-i\frac{t}{n}H} e^{-i\frac{t}{n}H} e^{-i\frac{t}{n}H} \cdots e^{-i\frac{t}{n}H}|\psi(t=0)\rangle$$

Step 1: approximate $e^{-i\delta H}$

Real/imaginary time evolution (TEBD) (Vidal)

Goal: compute the time evolution of a given initial MPS under a given Hamiltonian H: $|\psi(t)\rangle =$

Restriction: H is a nearest neighbor Hamiltonian $H =$

$\hat{h}_i$ ... acts on sites $i, i+1$ $h_i =$ 

Break t into n steps $\delta = \frac{t}{n}$

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

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme

Suzuki-Trotter:

Real / imaginary time evol...

Goal: compute the time evolution of a given initial MPS under a given Hamiltonian H: $|\psi(t)\rangle =$ $|\psi(t=0)\rangle$

many-body state in 1d

Restriction: H is a nearest neighbor Hamiltonian $H =$

$\hat{h}_i$ ... acts on sites $i, i+1$ $h_i =$ 

Break t into n steps $\delta = \frac{t}{n}$

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

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme

Suzuki-Trotter:

(1) given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B}$

09:35 Donnerstag 11. Mai

61 %

time evolution of a giv

Hamiltonian H: $|\psi(t)\rangle = e^{-iHt} |\psi(t=0)\rangle$

goal

many-body state in 1d

rest neighbor Hamiltonian $H =$

on sites $i, i+1$ $h_i =$ 

$\frac{t}{n}$

$|\uparrow\rangle |\downarrow\rangle |\uparrow\rangle |\downarrow\rangle |\downarrow\rangle |\uparrow\rangle$

$|\psi(t=0)\rangle =$ $|\psi(t=0)\rangle$

$e^{-i\delta H}$ using Trotter ~~Suzuki scheme~~

45 / 92

Goal: compute the time evolution of a given initial MPS under a given Hamiltonian H: $|\psi(t)\rangle = \underbrace{e^{-iHt}}_{\text{many-body state in 1d}} \underbrace{|\psi(t=0)\rangle}_{\text{many-body state in 1d}}$

Restriction: H is a nearest neighbor Hamiltonian $H = \sum_{i=1}^{N} h_i$

$\hat{h}_i$ ... acts on sites $i, i+1$ $\quad h_i =$ 



Break t into n steps $\delta = \frac{t}{n}$

$$|\psi(t)\rangle = e^{-iHt}|\psi(t=0)\rangle = \qquad |\psi(t=0)\rangle$$

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme

Suzuki-Trotter:

① given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B}$

h_i ... acts on sites $i, i+1$

$h_i = S_i^x S_{i+1}^x + \frac{B}{2}(S_i^z + S_{i+1}^z)$ $|\uparrow\rangle|\downarrow\rangle|\uparrow\rangle|\downarrow\rangle|\downarrow\rangle$

Break t into n steps $\delta = \frac{t}{n}$

$$|\psi(t)\rangle = e^{-iHt}|\psi(t=0)\rangle = e^{-i\delta t H} e^{-i\delta t H} \cdots e^{-i\delta t H}|\psi(t=0)\rangle$$

$\delta t = \frac{t}{N_1}$

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme

Suzuki-Trotter:

① given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B}$

② if $[A,B]\neq 0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B} + \mathcal{O}(\delta^2[A,B])$

Baker-Campbell-Hausdorff:

$e^X e^Y = e^Z$ with $Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] + \cdots$

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme $e^{-i\delta H}|\psi\rangle$

Suzuki-Trotter: $H = \sum_{i=\text{even}} h_i + \sum_{i=\text{odd}} h_i$, $[h_{i=\text{even}}, h_{i=\text{odd}}] = \{$

(1) given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A}\, e^{i\delta B}$

(2) if $[A,B]\neq 0$, then $e^{i\delta(A+B)} = e^{i\delta A}\, e^{i\delta B} + \mathcal{O}(\delta^2[A,B])$

Baker-Campbell-Hausdorff:

$e^X e^Y = e^Z$ with $Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] + \cdots$

(3) H is nearest neighbor: $H = \underbrace{\sum_{e\text{-even}} h_e}_{H_e} + \underbrace{\sum_{o=\text{odd}} h_o}_{H_o} \rightarrow \begin{cases} [h_e, h_{e'}] = 0 & \forall\, e, e' \text{ even} \\ [h_o, h_{o'}] = 0 & \forall\, o, o' \ldots \text{odd} \end{cases}$

$-i\delta H \quad -i\delta H_e \; -i\delta H_o \; -i\delta H_e \; \ldots \; i\delta h_e \; -i\delta H_o \; __ \; -i\delta h_o$

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme $e^{\quad}|\psi\rangle$

Suzuki-Trotter: $H = \sum_{a=\text{even}} h_i + \sum_{i=\text{odd}} h_i$, $[h_{i=\text{even}}, h_{j=\text{odd}}]$

① given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A}\, e^{i\delta B}$

② if $[A,B]\neq 0$, then $e^{i\delta(A+B)} = e^{i\delta A}\, e^{i\delta B} + \mathcal{O}(\delta^2[A,B])$

Baker-Campbell-Hausdorff:

$e^X e^Y = e^Z$ with $Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] + \cdots$

③ H is nearest neighbor: $H = \underbrace{\sum_{e=\text{even}} h_e}_{H_e} + \underbrace{\sum_{o=\text{odd}} h_o}_{H_o} \rightarrow \begin{cases} [h_e, h_{e'}] = 0 & \forall\, e, e' \text{ even} \\ [h_o, h_{o'}] = 0 & \forall\, o, o' \ldots \text{odd} \end{cases}$

$-i\delta H \quad -i\delta H_e \; -i\delta H_o \; -i\delta H_e \; \cancel{-i\delta h_e \; -i\delta H_o} \; -i\delta h_o$

Step 1: approximate $e^{-i\delta H}$ using Trotter-Suzuki scheme $|\psi\rangle$

Suzuki-Trotter: $H = \underbrace{\sum_{i=\text{even}} h_i}_{A} + \underbrace{\sum_{i=\text{odd}} h_i}_{B} + [h_{i=\text{even}}, h_{j=\text{odd}}]$

① given A, B with $[A,B]=0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B}$

② if $[A,B]\neq 0$, then $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B} + \mathcal{O}(\delta^2[A,B])$

Baker-Campbell-Hausdorff:

$$e^X e^Y = e^Z \text{ with } Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] + \cdots$$

③ H is nearest neighbor: $H = \underbrace{\sum_{e=\text{even}} h_e}_{H_e} + \underbrace{\sum_{o=\text{odd}} h_o}_{H_o} \rightarrow \begin{cases} [h_e, h_{e'}] = 0 \quad \forall\, e, e' \text{ even} \\ [h_o, h_{o'}] = 0 \quad \forall\, o, o' \ldots \text{odd} \end{cases}$

$$e^{-i\delta H} \approx e^{-i\delta H_e} e^{-i\delta H_o}, \quad e^{-i\delta H_e} = \prod_e e^{-i\delta h_e}, \quad e^{-i\delta H_o} = \prod_o e^{-i\delta h_o}$$

① given A, B with $[A,B]=0$, then $e^{A+B} = e^{A}\, e^{B}$

② if $[A,B]\neq 0$, then $e^{\delta t(A+B)} = e^{\delta t A}\, e^{\delta t B} + \mathcal{O}(\delta t^2 [A,B])$ — second order in δt

Baker-Campbell-Hausdorff:

$$e^{X}\, e^{Y} = e^{Z} \quad \text{with} \quad Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] + \cdots$$

③ H is nearest neighbor: $H = \underbrace{\sum h_e}_{H_e} + \underbrace{\sum h_o}_{H_o}$ (e = even, o = odd) $\rightarrow$

$$\begin{cases} [h_e, h_{e'}] = 0 \quad \forall\, e, e' \text{ even} \\ [h_o, h_{o'}] = 0 \quad \forall\, o, o' \ldots \text{odd} \end{cases}$$

$$e^{-i\delta H} \approx e^{-i\delta H_e}\, e^{-i\delta H_o}, \quad e^{-i\delta H_e} = \prod_{e} e^{-i\delta h_e}, \quad e^{-i\delta H_o} = \prod_{o} e^{-i\delta h_o}$$

45 / 92

Step 2: evolve a given MPS over a small time δ using above

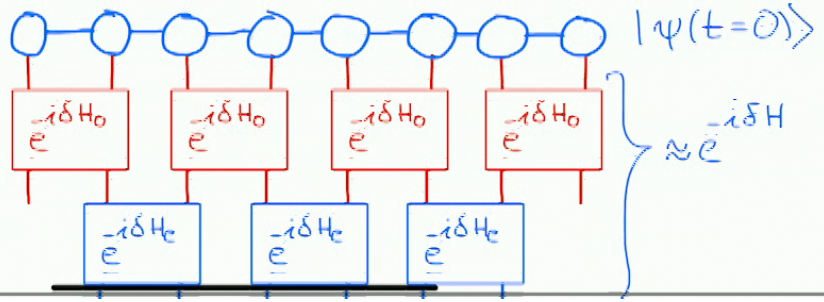
$e^X e^Y = e^Z$ with $Z = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] + \frac{1}{12}[Y,[Y,X]] + \ldots$

③ H is nearest neighbor: $H = \underbrace{\sum_{e\text{-even}} h_e}_{H_e} + \underbrace{\sum_{o=\text{odd}} h_o}_{H_o} \rightarrow \begin{cases} [h_e, h_{e'}] = 0 \quad \forall\, e, e' \text{ even} \\ [h_o, h_{o'}] = 0 \quad \forall\, o, o' \ldots \text{odd} \end{cases}$

$$e^{-i\delta H} \approx e^{-i\delta H_e} e^{-i\delta H_o}, \quad e^{-i\delta H_e} = \prod_e e^{-i\delta h_e}, \quad e^{-i\delta H_o} = \prod_o e^{-i\delta h_o}$$

Step 2: evolve a given MPS over a small time δ using above formula

$|\psi(t=\delta)\rangle =$

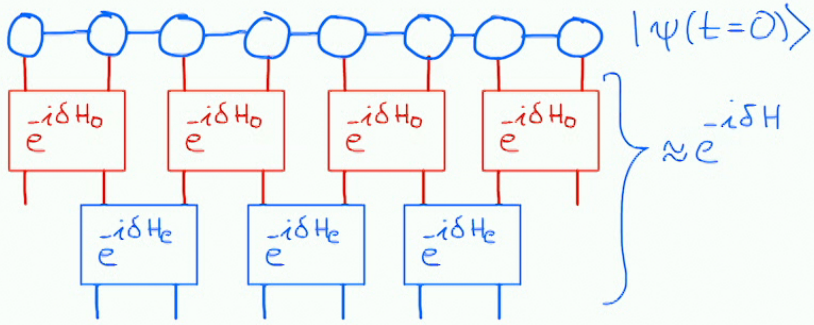


③ H is nearest neighbor: $H = \underbrace{\sum h_e}_{H_e} + \underbrace{\sum h_o}_{H_o}$ → $[h_o, h_{o'}] = 0 \quad \forall\, o, o' \ldots \text{odd}$

$$e^{-i\delta H} \approx e^{-i\delta H_e} e^{-i\delta H_o}, \quad e^{-i\delta H_e} = \prod_e e^{-i\delta h_e}, \quad e^{-i\delta H_o} = \prod_o e^{-i\delta h_o}$$

Step 2: evolve a given MPS over a small time δ using above formula

$|\psi(t=\delta)\rangle =$

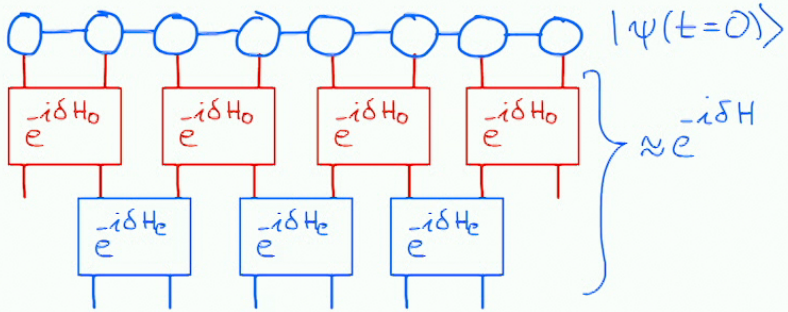


the MPS form is lost, need to recover it!

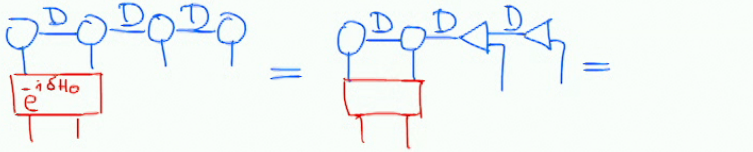
BCH-formula

Step 2: evolve a given MPS over a small time δ using above formula

$|\psi(t=\delta)\rangle =$



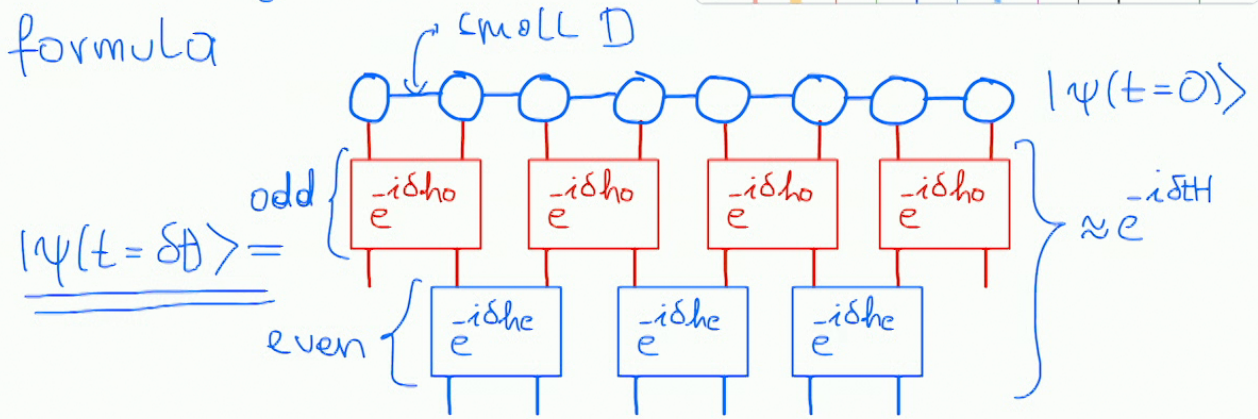
the MPS form is lost, need to recover it!
consider application of one to given MPS:





step 2 : evolve a given MPS over a

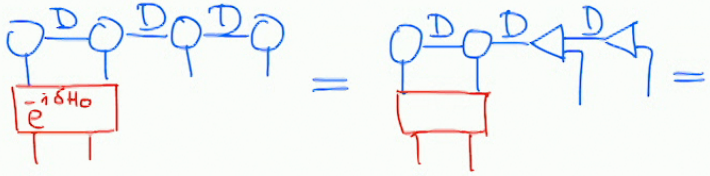
formula

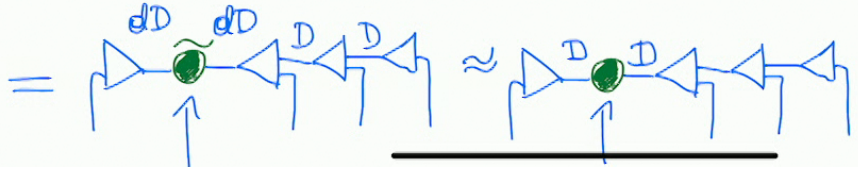


$|\psi(t=\delta t)\rangle =$

the MPS form is lost, need to recover it!

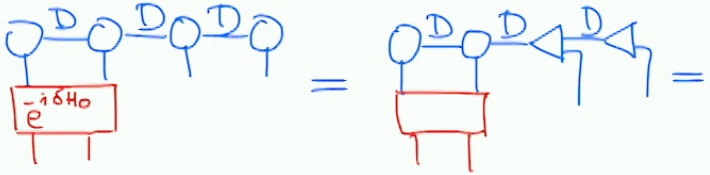
consider application of one to given MPS:

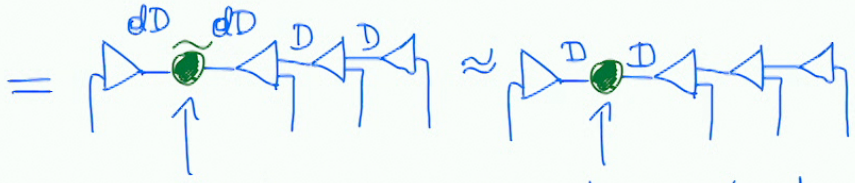




the MPS form is lost, need to recover it!

consider application of one to given MPS:





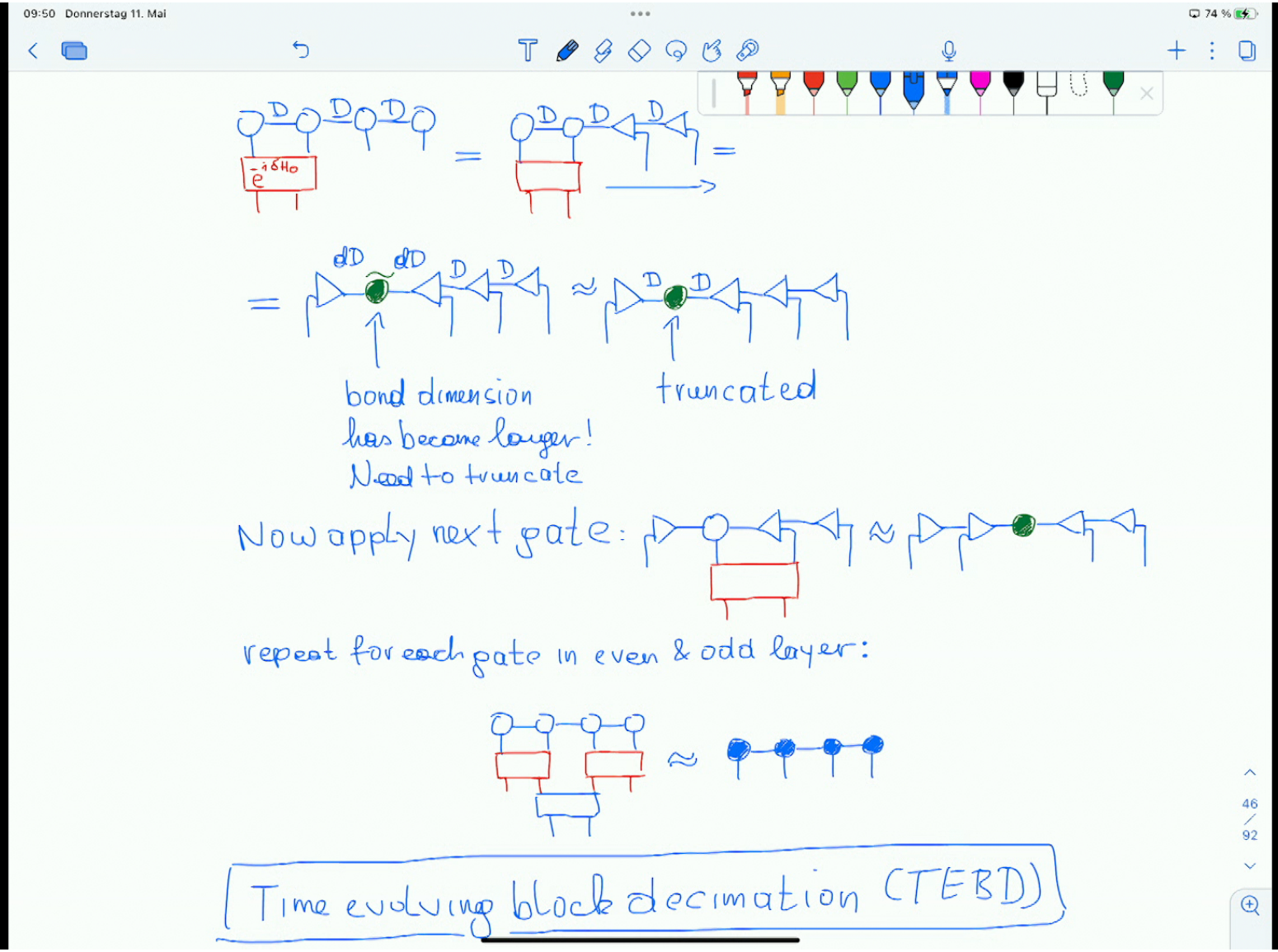
bond dimension has become larger! Need to truncate

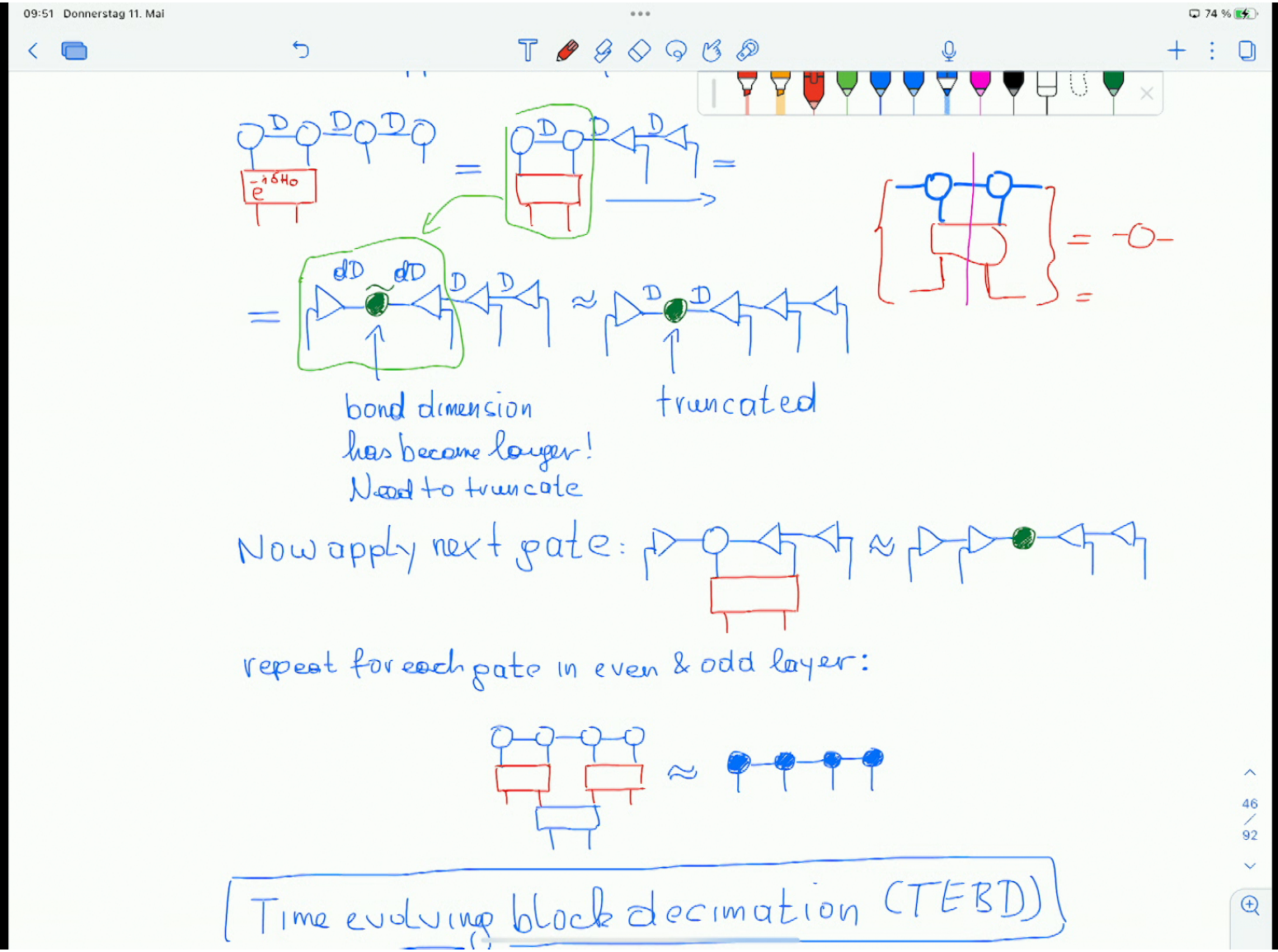
truncated

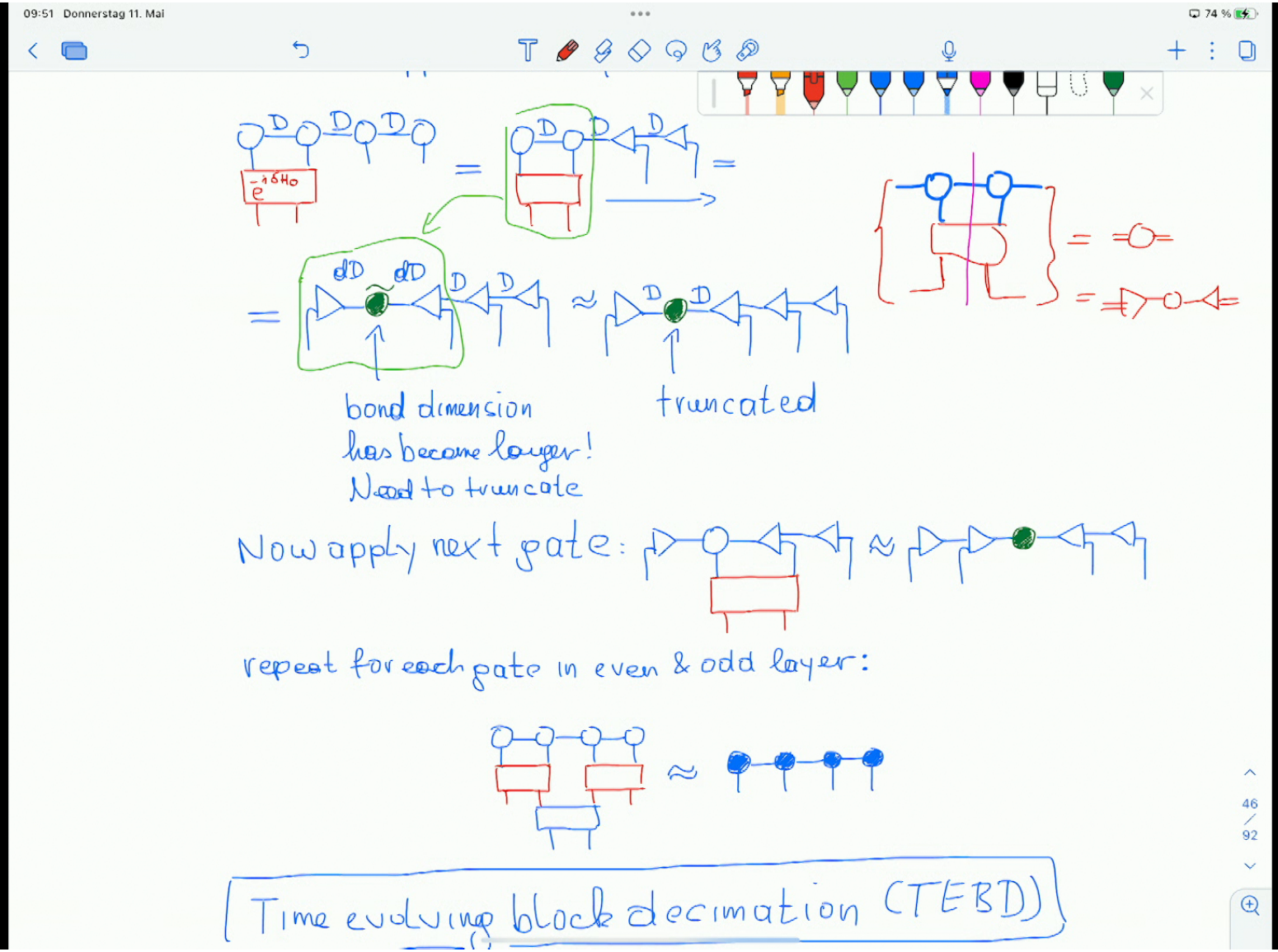
Now apply next gate:



repeat for each gate in even & odd layer:


09:50 Donnerstag 11. Mai
74 %
D
D
D
$e^{-i\delta H_0}$
dD
dD
bond dimension has become longer! Need to truncate
truncated
Now apply next gate:
repeat for each gate in even & odd layer:
Time evolving block decimation (TEBD)
46
92


09:51 Donnerstag 11. Mai
74 %
bond dimension has become larger! Need to truncate
truncated
Now apply next gate:
repeat for each gate in even & odd layer:
Time evolving block decimation (TEBD)
46
92


09:51 Donnerstag 11. Mai
74 %
bond dimension has become larger! Need to truncate
truncated
Now apply next gate:
repeat for each gate in even & odd layer:
Time evolving block decimation (TEBD)
46
92

TEBD has two sources of errors:

a) Trotter error due to $e^{i\delta(A+B)} = e^{i\delta A} e^{i\delta B} + \mathcal{O}(\delta^2)$

In most cases, this is negligible; can be reduced by higher order Trotter expansion!

b) Truncation error: after each gate application the state has to be truncated, otherwise computationally unfeasible.

Truncation is the main error source, and causes simulated evolution to deviate from real one exponentially fast

Consequence: real time evolution with MPS is classically unfeasible.

Example: global quantum quench of Ising model

Given $H = J \sum_i S^x_i S^x_{i+1} + B \sum_i S^z_i$

a) Trotter error due to e
In most cases, this is negligible; can be reduced by higher order Trotter expansion!

b) Truncation error: after each gate application the state has to be truncated, otherwise computationally unfeasible.
Truncation is the main error source, and causes simulated evolution to deviate from real one exponentially fast
Consequence: real time evolution with MPS is classically unfeasible.

Example: global quantum quench of Ising model

Given $H = J \sum_i S_i^x S_{i+1}^x + B \sum_i S_i^z$

Consequence: real time evolution unfeasible.

Example: global quantum quench of Ising model

Given $H = J \sum_i S_i^x S_{i+1}^x + B \sum_i S_i^z$ $\xrightarrow{\text{DMRG}}$ $|\psi(J,B)\rangle$

J, B

Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H

Consequence: ... unfeasible.

Example: global quantum quench of Ising model

Given $H = J \sum_i S_i^x S_{i+1}^x + B \sum_i S_i^z$ $\xrightarrow{\text{DMRG}}$ $|\psi(J,B)\rangle$ ← GS of H_{JB}

$e^{-iH_{JB}t} |\psi(J,B)\rangle = e^{-iEt} |\psi(J,B)\rangle$

"quench" $J, B \to J' \neq J$, $H_{J,B} \to |\psi(t)\rangle = e^{-iH_{J'B}t} |\psi(J,B)\rangle$

Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- [illegible] Suzuki-Trotter decomposition

Given $H = J \sum_i S^x_i S^x_{i+1} + B \sum_i S^z_i$ $\xrightarrow{\text{DMRG}}$ $|\psi(J,B)\rangle$

$e^{-iH_{J,B}t} |\psi(J,B)\rangle = e^{-iEt} |\psi(J,B)\rangle$

"quench" $J, B \longrightarrow J' \neq J$, $H_{J',B}$ $\Rightarrow$ $|\psi(t)\rangle = e^{-iH_{J',B}t} |\psi(J,B)\rangle$



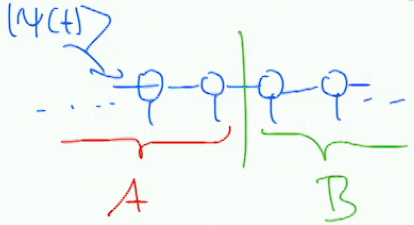
Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- [illegible] DMRG for [illegible]

Given $H = J\sum_i S^x_i S^x_{i+1} + B\sum_i S^z_i$ → DMRG → $|\psi(J,B)\rangle$

$e^{-iH_{J,B}t}|\psi(J,B)\rangle = e^{-iEt}|\psi(J,B)\rangle$

"quench" $J \to J' \neq J$, $H_{J,B} \to H_{J',B}$ → $|\psi(t)\rangle = e^{-iH_{J',B}t}|\psi(J,B)\rangle$



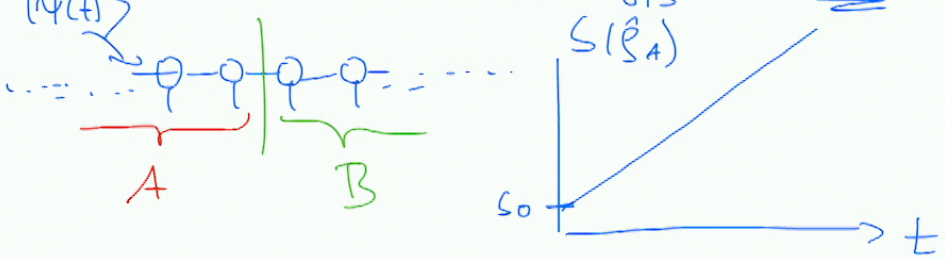
Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- [illegible] DMRG for [illegible] time evolution

Given $H = J \sum_i S_i^x S_{i+1}^x + B \sum_i S_i^z$ $\xrightarrow{\text{DMRG}}$ $|\psi(J,B)\rangle$

$e^{-iH_{JB}t}|\psi(J,B)\rangle = e^{-iEt}|\psi(J,B)\rangle$

"quench" $J, B \longrightarrow J' \neq J, H'_{J,B} \longrightarrow |\psi(t)\rangle = e^{-iH'_{JB}t}|\psi(J,B)\rangle$



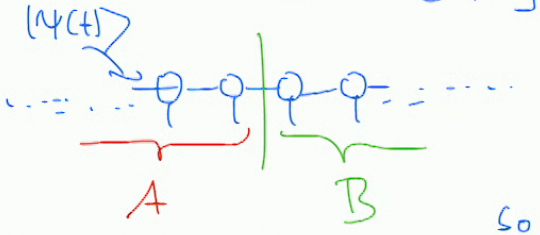
Some comments:

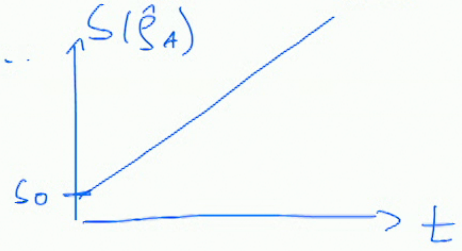
- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- ~~[illegible] DMRG for [illegible] time evolution~~

Given $H = J \sum_i S_i^x S_{i+1}^x + B \sum_i S_i^z$ $\xrightarrow{\text{DMRG}}$ $|\psi(J,B)\rangle$

$e^{-iH_{JB}t}|\psi(J,B)\rangle = e^{-iEt}|\psi(J,B)\rangle$

"quench" $J \to J' \neq J$, $H_{J,B} \to H'_{J,B}$ $\Rightarrow$ $|\psi(t)\rangle = e^{-iH'_{JB}t}|\psi(J,B)\rangle$

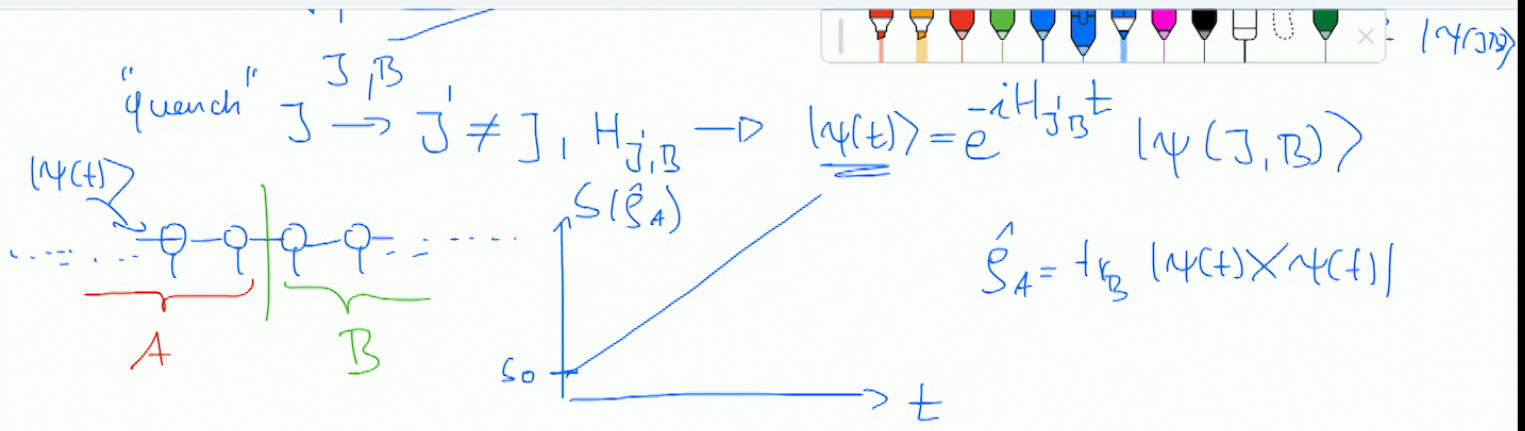




$\hat{\rho}_A = \mathrm{tr}_B |\psi(t)\rangle\langle\psi(t)|$

Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- ~~H [illegible] DMRG for [illegible] time evolution~~



$$\hat{\rho}_A = \mathrm{tr}_B \, |\psi(t)\rangle\langle\psi(t)|$$

Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- High-order MPOs for accurate time evolution

Density matrix renormalization group

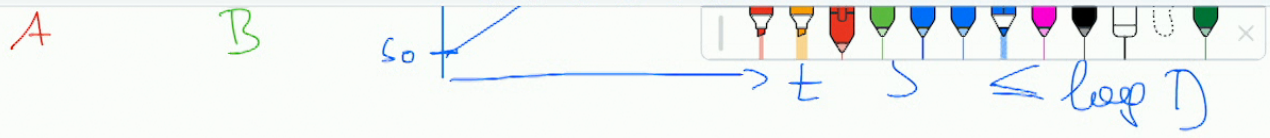
- Invented in 92 by Steve White; back then no connection to MPS
- Östlund & Rommer realized in 95 connection to MPS
- Modern formulation: Nishino, Vidal, Verstraete, Cirac, ...

Modern formulation of DMRG:

Goal: find an approximation to the ground state of a 1d Hamiltonian

Method: optimize a variational ansatz wavefunction (MPS)

DMRG is a variational method in the space of MPS:



$S(t) \sim \alpha t$ → D needs to grow exponentially

Some comments:

- TEBD can be used with imaginary time $\delta = i\tau$ to approximate GS of H
- in practice use higher order Suzuki-Trotter decomposition
- more modern approaches to time evolution: TDVP
- High-order MPOs for accurate time evolution
- Groundstates from imaginary time evolution: $t \to it = \tau$, $e^{-\hat{\tau}H}|\psi(t=0)\rangle$

Matrix product operators

MPOs are tensor networks representing many body operators $\hat{O}$

Density matrix renormalization group

- Invented in 92 by Steve White; back then no connection to MPS
- Östlund & Rommer realized in 95 connection to MPS
- Modern formulation: Nishino, Vidal, Verstraete, Cirac, ...

Modern formulation of DMRG:

Goal: find an approximation to the ground state of a 1d Hamiltonian

Method: optimize a variational ansatz wavefunction (MPS)

DMRG is a variational method in the space of MPS:

$$\min_{|\psi\rangle} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \quad \text{with } |\psi\rangle \text{ an MPS}$$

MPS

– Modern formulation: Nishino, Vidal, Verstraete, Cirac, ...

Modern formulation of DMRG:

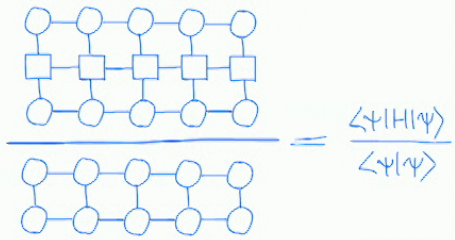
Goal: find an approximation to the ground state of a 1d Hamiltonian

Method: optimize a variational ansatz wavefunction (MPS)

DMRG is a variational method in the space of MPS:

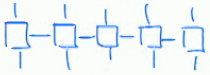
$$\min_{|\psi\rangle} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \quad \text{with } |\psi\rangle \text{ an MPS}$$

Diagrammatically:

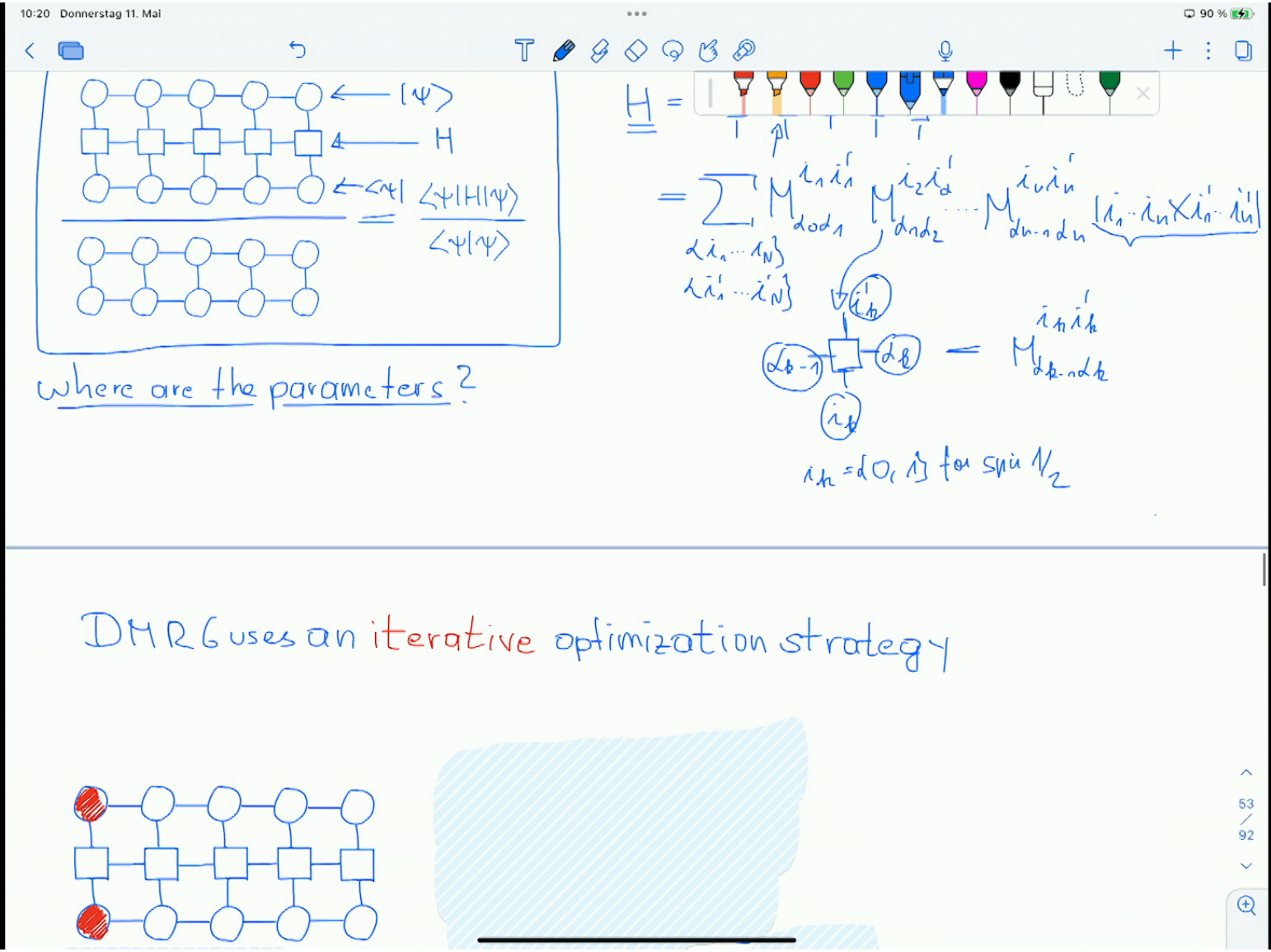


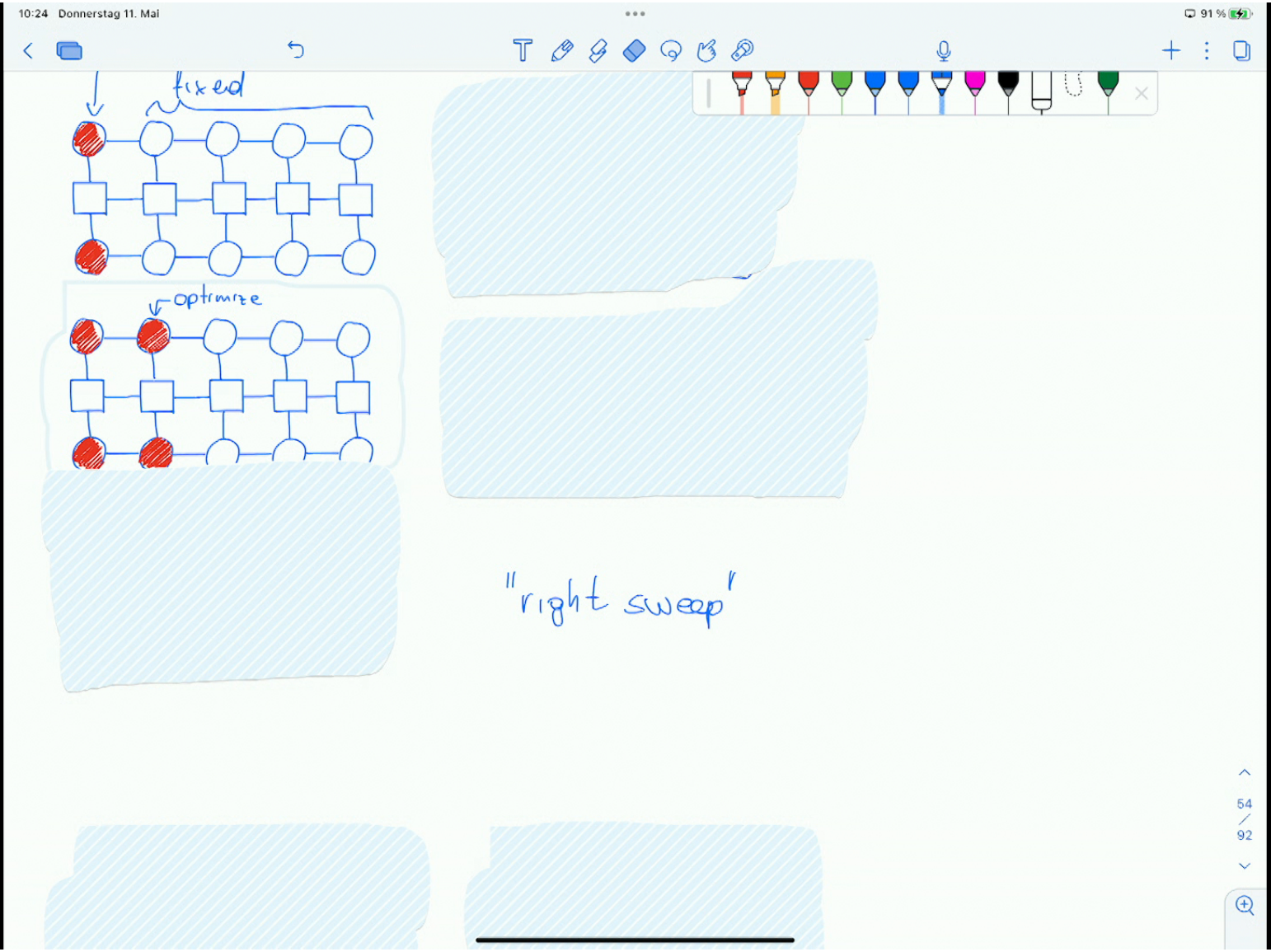
$= \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$

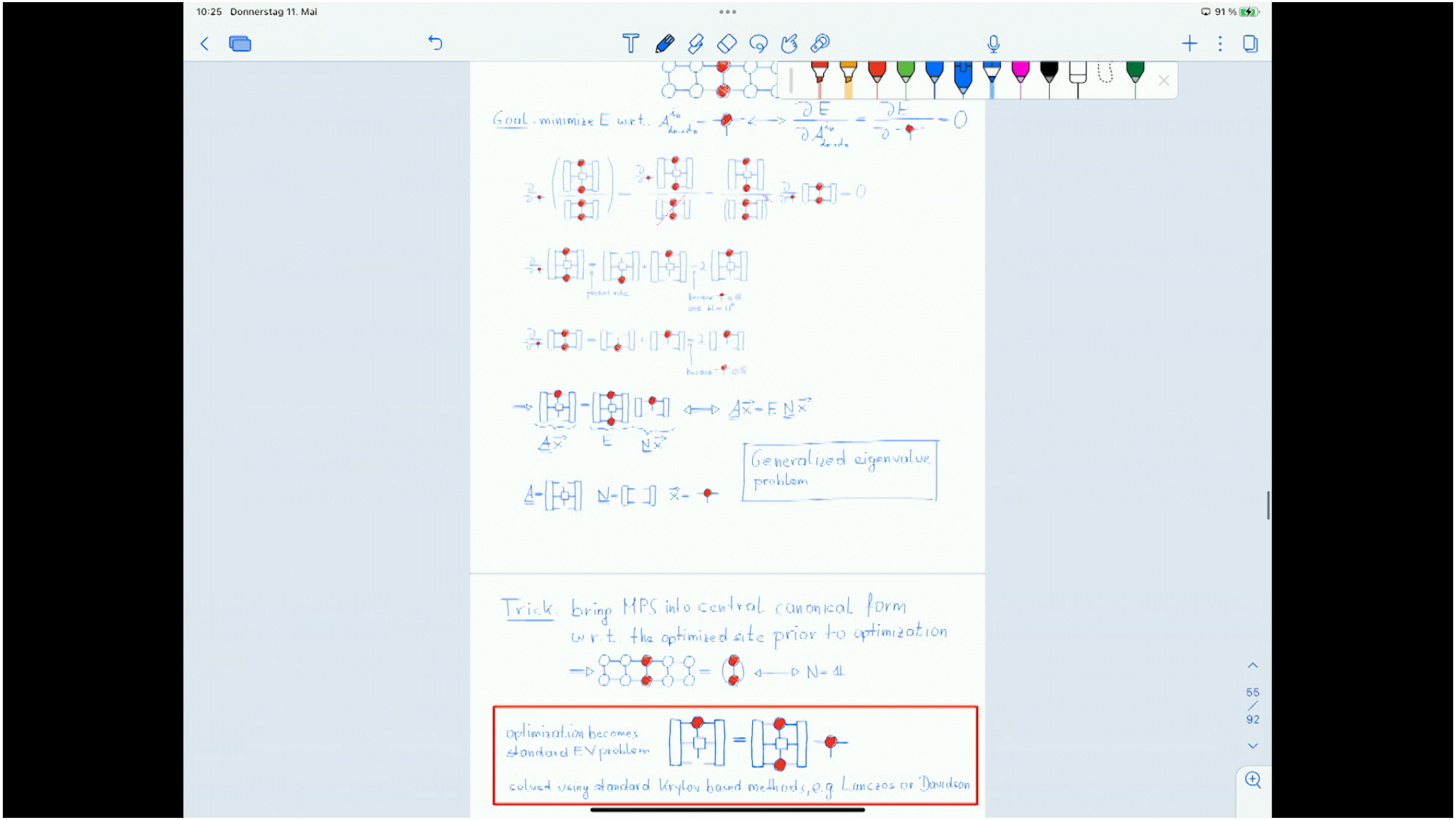
matrix product operator (MPO)

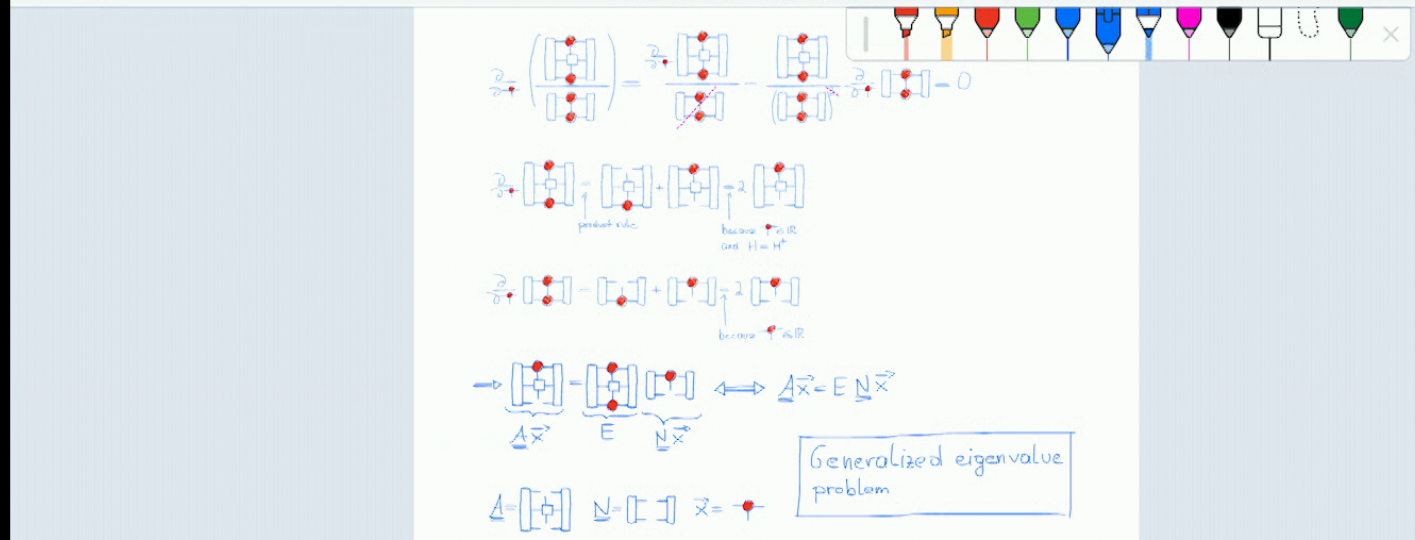
$\hat{H} =$ 



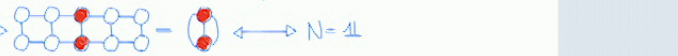

10:20 Donnerstag 11. Mai
90 %
|ψ⟩
H
⟨ψ|
⟨ψ|H|ψ⟩
⟨ψ|ψ⟩
Where are the parameters?
H =
= Σ M M M |i1 ... iN⟩⟨i'1 ... i'N|
{α i1 ... iN}
{α i'1 ... i'N}
i'k
αk-1
αk
ik
= M
ik = |0⟩, |1⟩ for spin 1/2
DMRG uses an iterative optimization strategy
53
92


10:24 Donnerstag 11. Mai
91 %
fixed
optimize
"right sweep"
54
92

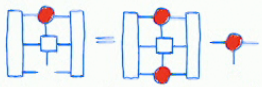

10:25 Donnerstag 11. Mai
91 %
Goal: minimize E wrt.
Trick: bring MPS into central canonical form
w.r.t. the optimized site prior to optimization
optimization becomes
standard EV problem
solved using standard Krylov based methods, e.g. Lanczos or Davidson
55
92



Trick: bring MPS into central canonical form w.r.t. the optimized site prior to optimization



optimization becomes standard EV problem



solved using standard Krylov based methods, e.g. Lanczos or Davidson

"Summary" of DMRG

- sweep from left to right and right to left

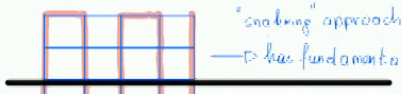
- contract the network to obtain A above for site "n"
- Use standard package to compute the lowest eigenstate of A. This will be your new tensor for site "n"

Note: we skipped over a few details here which are important for a practical implementation, but by and large this is how DMRG works

DMRG is one of the most successfull & powerful algorithms for quantum many-body systems! Original paper cited ~ 8000 times!

Applications of DMRG

- ground states of 1d systems
- ignited the field of tensor networks
- can be extended to low lying excited states
- DMRG forms basis for many more advanced methods, e.g.
  - time evolution
  - 2d-network contractions
  - supervised learning and generative modelling
- Is also applied directly to "2d" systems, e.g. infinite cylinders

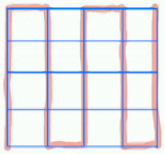


"scaling" approach
→ has fundamental limitations

DMRG is one of the most sucessfull & powerful algorithms for quantum manybody systems! Original paper cited ~ 8000 times!

Applications of DMRG

- ground states of 1d systems
- ignited the field of tensor networks
- can be extended to low lying excited states
- DMRG forms basis for many more advanced methods, e.g.
  - time evolution
  - 2d-network contractions
  - supervised learning and generative modelling
- Is also applied directly to "2d" systems, e.g. infinite cylinders



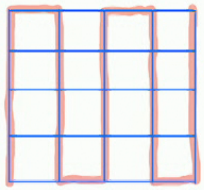
"snaking" approach
→ has fundamental limitations

- quantum chemistry

DMRG is one of the most sucessfull & powerful algorithms for quantum many body systems! Original paper cited ~ 8000 times!

Applications of DMRG

- ground states of 1d systems
- ignited the field of tensor networks
- can be extended to low lying excited states
- DMRG forms basis for many more advanced methods, e.g.
  - time evolution
  - 2d-network contractions
  - supervised learning and generative modelling
- Is also applied directly to "2d" systems, e.g. infinite cylinders



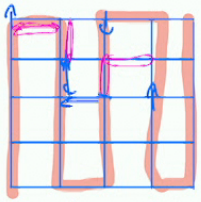
"snaking" approach
⟶ has fundamental limitations

- quantum chemistry

manybody systems! Original paper cit

Applications of DMRG

- ground states of 1d systems
- ignited the field of tensor networks
- can be extended to low lying excited states
- DMRG forms basis for many more advanced methods, e.g.
  - time evolution
  - 2d-network contractions
  - supervised learning and generative modelling
- Is also applied directly to "2d" systems, e.g. infinite cylinders



"snaking" approach
→ has fundamental limitations
→ H is long-range

- quantum chemistry

TN beyond MPS (PEPS & MERA)

Projected entangled pair states (PEPS)

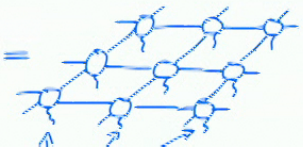
Motivation: can we extend TN description to ground states of 2d quantum lattices?

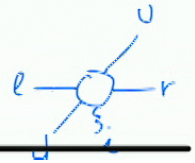
MPS: could do "counting" approach, but there is a fundamental problem:



- Area law:
- MPS can represent $S_A \leq \log D \Longleftrightarrow D \sim e^{S_A}$
- So area law in 2d requires $D \sim e^{L}$, i.e. exponential dependence!

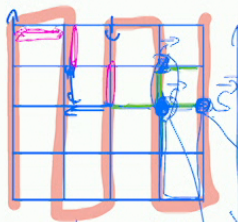
PEPS: efficient representation of many body wavefunctions in 2d

$|\psi\rangle =$  infinite or finite



$= A^{i}_{urdl}$

• Is also applied directly to "2d" sys



"snaking" approach
▷ has fundamental limitations
▷ H is long-range

• quantum chemistry

$H = \sum_{\langle ij \rangle} \vec{S}_i \vec{S}_j$

comput. scaling
$\sim N D^3$

Lecture IV

Finite temperature, open systems

TN beyond MPS (PEPS & MERA)

Projected entangled pair states (PEPS)

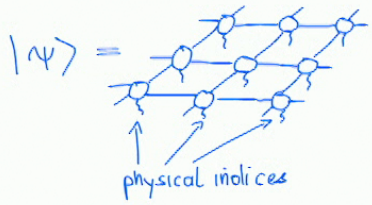
Motivation: can we extend TN description to ground states of 2d quantum lattices?

MPS: could do "snaking" approach, but there is a fundamental problem:

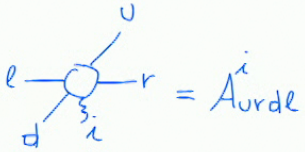


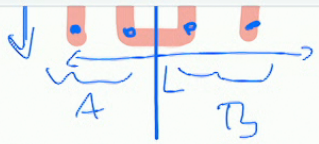
- Area law:
- MPS can represent $S_A \leq \log D \Longleftrightarrow D \sim e^{S_A}$
- So area law in 2d requires $D \sim e^{L}$, i.e. exponential dependence!

PEPS: efficient representation of many body wavefunctions in 2d



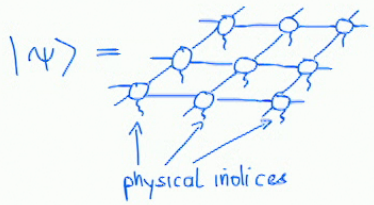
infinite or finite



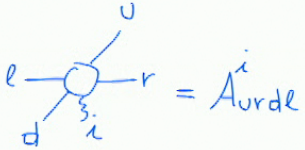


- So area law in 2d v
  i.e. exponential dependence!

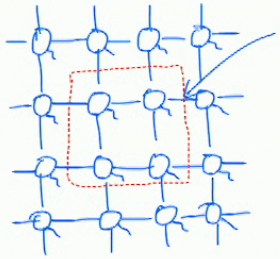
PEPS: efficient representation of many body wavefunctions in 2d



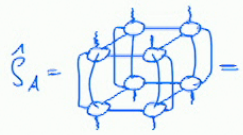
infinite or finite

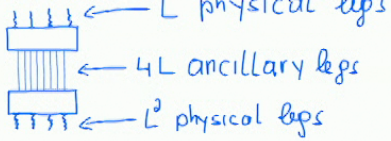


PEPS obey an area law (argument similar to MPS)!



region A of size L × L with reduced density matrix $\hat{\rho}_A$:

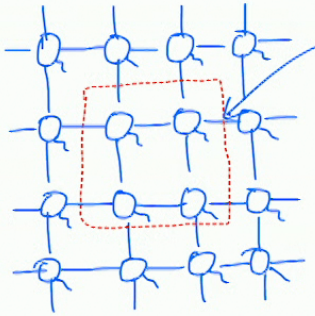




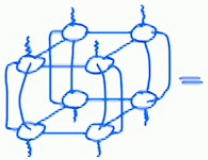
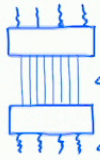
→ rank of $\rho_{A'} \leq D^{4L}$

→ $\rho_{A'}$ has at most D^{4L} non-zero EV

PEPS obey an area law (argument similar to MPS)!



region A of size $L \times L$ with reduced density matrix $\hat{\rho}_A$:

$\hat{\rho}_A =$  $=$ 

- L^2 physical legs
- $4L$ ancillary legs
- L^2 physical legs

$\Rightarrow$ rank of $\rho_A \leq D^{4L}$

$\Rightarrow$ ρ_A has at most D^{4L} non-zero EV

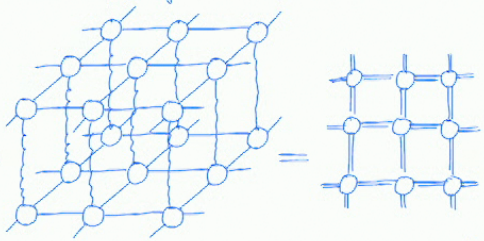
$$\Rightarrow S(\rho_A) \leq \sum_{i=1}^{D^{4L}} \frac{1}{D^{4L}} \log\left[D^{4L}\right] = 4L \log(D)$$

Numerics:

- Form variational class of w.f, similar to MPS: $\min_{\{A^{[r,c]}_{\alpha\beta\gamma\delta}\}} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$
- approximate variational optimization possible
- more expensive than MPS

Main problem: PEPS cannot be contracted exactly!

Example: computing norm $\langle\psi|\psi\rangle$

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

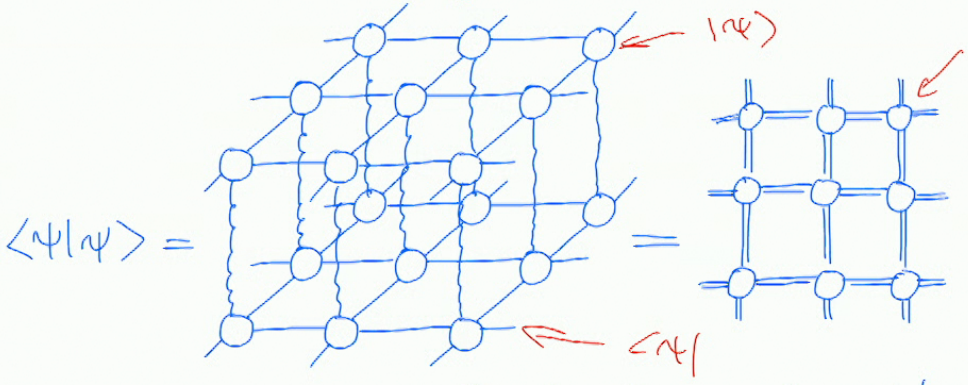
Many different approaches have been explored! One example is the Corner transfer matrix method:

10:48 Donnerstag 11. Mai

100 %

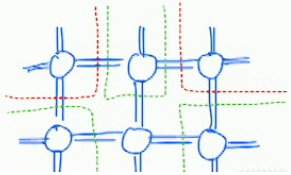
Main problem: PEPS cannot be contracted exactly!

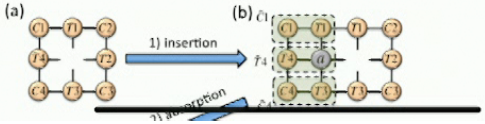
Example: computing norm $\langle\psi|\psi\rangle$



Many different approaches have been explored! One example is the Corner transfer matrix method:

Idea: use renormalization procedure to contract the network approximately

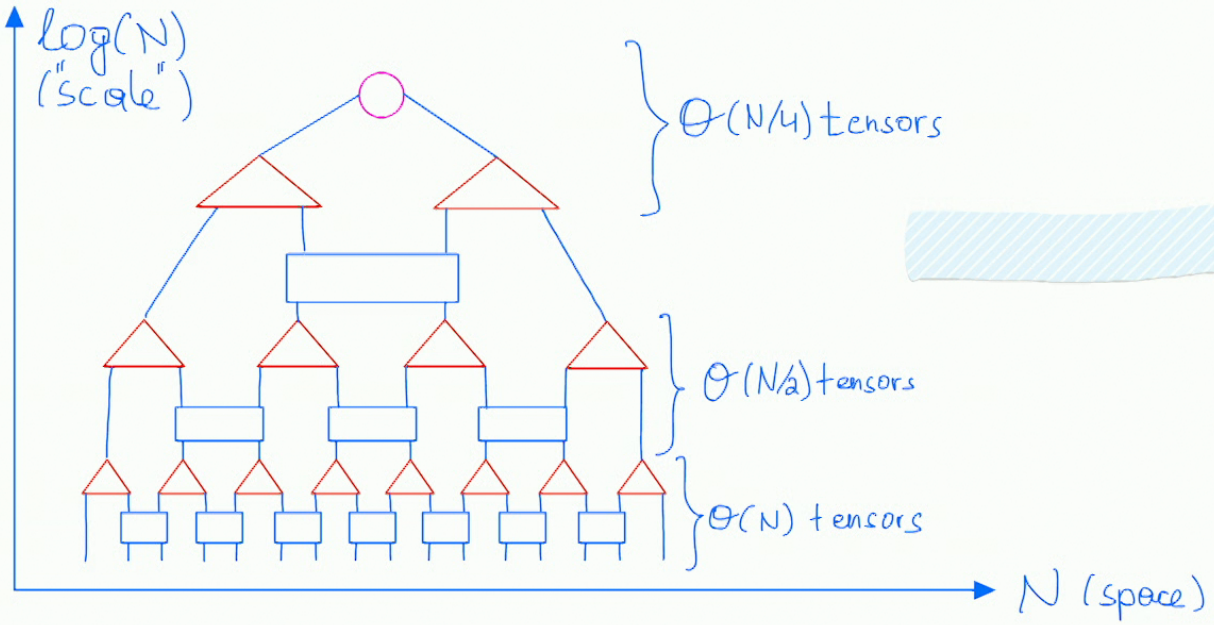




65 / 92

Multiscale entanglement renormalization ansatz

MERA is a TN ansatz to represent many body wave functions



How many parameters (i.e. tensors)? $N + \frac{N}{2} + \frac{N}{4} + \ldots = N\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) = 2N$

How many parameters (i.e. tensors)? $N + \frac{N}{2} + \frac{N}{4} + \ldots = N\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) = 2N$

 and are unitaries:

⟹ MERA network can be contracted efficiently without approximations!

MERA can be used as a variational TN ansatz and can be optimized efficiently!

Computing the norm of a MERA state

$$S(\hat{\rho}_L) = -\operatorname{tr} \hat{\rho}_L \log \hat{\rho}_L = -\sum_i p_i \log p_i \leq \sum_{i=1}^{D^{\log L}} \frac{1}{D^{\log L}} \log D^{\log L} = \log L$$

