Infinite Time-Evolving Block Decimation (iTEBD)

[Vidal2007], [Schollwöck2011, Sec. 10.4]

Goal: ground state search for infinite system while exploiting translational invariance.

We will use Vidal's $\Gamma\Lambda$ notation [see Section 2], but the strategy can be expressed in other notations, too.

Basic idea: 'imaginary time evolution':

$$\lim_{\beta\to\infty} e^{-\beta\hat H}|\psi\rangle \propto |g\rangle \qquad (1)$$

Reason: high-energy states die out quickly (if ground state is gapped):

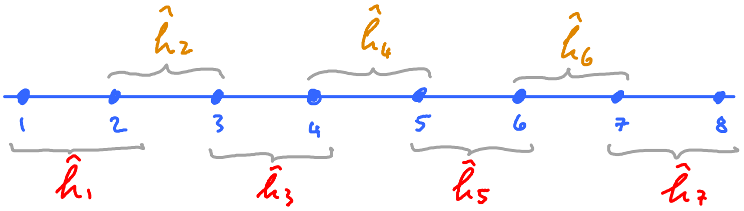
$$e^{-\beta\hat H} = \sum_{\underset{\text{complete set of energy eigenstates}}{\alpha}} e^{-\beta\hat H}|\alpha\rangle\langle\alpha| \xrightarrow{\beta\to\infty} \underbrace{e^{-\beta E_g}|g\rangle\langle g|}_{\text{projector onto ground state}} \qquad (2)$$

1. Trotter decomposition of time evolution operator

[Schollwöck2011, Sec. 7.1.1]

General: write Hamiltonian as

$$\hat H = \sum_\ell \underset{\text{connects sites } \ell \text{ and } \ell+1}{\hat h_\ell} = \underset{\text{odd}}{\hat H_o} + \underset{\text{even}}{\hat H_e} \qquad (3)$$

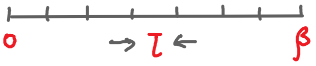


(4)

Then all odd terms mutually commute, and all even terms mutually commute:

$$[h_\ell, h_{\ell'}] = 0 \quad \text{if } \ell, \ell' \text{ are both odd or both even} \qquad (5)$$

Divide time interval into N slices: $\beta = \tau N$ (6)



$$e^{-\beta\hat H} \overset{\text{Trotter decomposition}}{=} \left[e^{-\tau\hat H}\right]^N = \left[e^{-\tau(\hat H_o + \hat H_e)}\right] \qquad (7)$$

$$\xrightarrow{N\to\infty,\ \tau\to 0} \simeq \left[e^{-\tau\hat H_o} e^{-\tau\hat H_e} + \mathcal{O}(\tau^2)\right]^N \quad \text{'first order Trotter approx.'} \qquad (8)$$

or

$$\simeq \left[e^{-\frac{\tau}{2}\hat H_o} e^{-\tau\hat H_e} e^{-\frac{\tau}{2}\hat H_o} + \mathcal{O}(\tau^3)\right]^N \quad \text{'second order Trotter approx.'} \qquad (9)$$

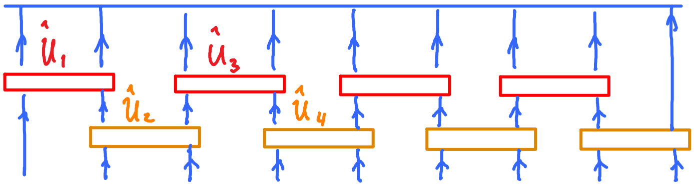
Exploiting (5), odd and even exponents can both be expanded separately without further approximation:

$$e^{-\tau\hat H_o} = e^{-\tau\hat h_1} e^{-\tau\hat h_3} \dots e^{-\tau\hat h_{\mathcal{L}-1}} := \hat U_1 \hat U_3 \dots \hat U_{\mathcal{L}-1} \qquad (10a)$$

$$e^{-\tau\hat H_e} = e^{-\tau\hat h_2} e^{-\tau\hat h_4} \dots e^{-\tau\hat h_{\mathcal{L}}} := \hat U_2 \hat U_4 \dots \hat U_{\mathcal{L}} \qquad (10b)$$

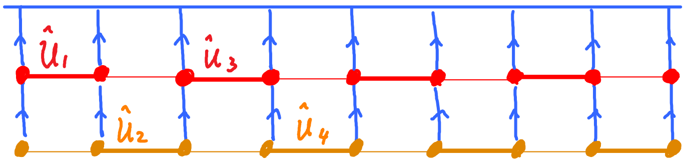
So, when applying $e^{-\beta \hat{H}}$ to $|\psi\rangle$, we can successively apply all odd terms, then truncate, then all even ones, then truncate, etc.

$e^{-\tau \hat{H}_e} \, e^{-\tau \hat{H}_o} \, |\psi\rangle =$



(11)

in MPO notation: $=$



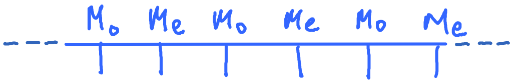
(12)

since $\hat{H}_o$ factorizes, even bonds have dimension $D_{w,e} = 1$

since $\hat{H}_e$ factorizes, odd bonds have dimension $D_{w,o} = 1$

All of this can be done for finite chain of length $\mathcal{L}$. But a simplification occurs for $\mathcal{L} \to \infty$

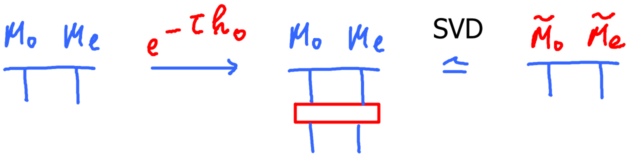
Then we can exploit translational invariance:



Adopt a two-site unit cell (no left- or right-normalization implied).

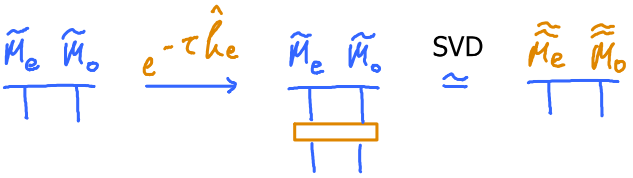
Step 1: time-evolve 'odd bond':

(first site odd, second site even)


M_o M_e
$\xrightarrow{e^{-\tau \hat{h}_o}}$
M_o M_e
SVD
$\simeq$
$\tilde{M}_o$ $\tilde{M}_e$

Step 2: time-evolve (updated!) even bond:

(first site even, second site odd)


$\tilde{M}_e$ $\tilde{M}_o$
$\xrightarrow{e^{-\tau \hat{h}_e}}$
$\tilde{M}_e$ $\tilde{M}_o$
SVD
$\simeq$
$\tilde{\tilde{M}}_e$ $\tilde{\tilde{M}}_o$

Iterate until convergence! (To discuss details, we will use $\Gamma\Lambda$ notation.)

iETBD is a 'power method': the projector to the ground state is constructed as an increasing number of powers of $e^{-\tau \hat{H}_e} \, e^{-\tau \hat{H}_o}$.

This is to be contrasted to DMRG ground state search, which is a variational method.

Main advantage of iTEBD: costs not proportional to system size, hence comparatively cheap.

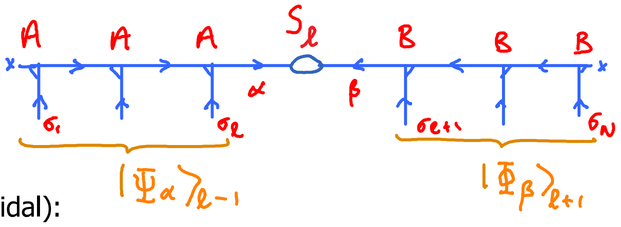
Main disadvantage: loss of orthogonality due to projection, without explicit reorthogonalization.

to be explained below

2. Gamma-Lambda Notation [Vidal2003], [Schollwöck2011, Sec. 4.6] iTEBD.2

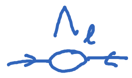
Usual bond-canonical form of MPS:

$$|\psi\rangle = |\Psi_\alpha\rangle_\ell \, |\Phi_\beta\rangle_{\ell+1} \, [S_\ell]^{\alpha\beta} \qquad (1)$$



Choose S diagonal, and call it Λ (following Vidal):

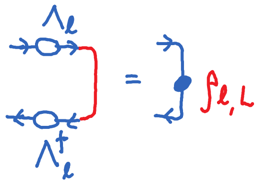
$$|\psi\rangle = \sum_\alpha |\Psi_\alpha\rangle_\ell \, |\Phi_\alpha\rangle_{\ell+1} \, [\Lambda_\ell]^{\alpha\alpha}$$ (Schmidt decomposition)



(2)

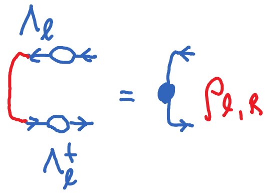
Then reduced density matrices of left and right parts are diagonal, with eigenvalues $(\Lambda_\ell^{\alpha\alpha})^2$:

$$\rho_L = \mathrm{Tr}_R |\psi\rangle\langle\psi| = \sum_\alpha |\Psi_\alpha\rangle_\ell \underbrace{[\Lambda_\ell]^{\alpha\alpha}[\Lambda_\ell^\dagger]_{\alpha\alpha}}_{[\rho_{\ell,L}]^{\alpha}{}_{\alpha}} {}_\ell\langle\Psi^\alpha| \, ,$$



(3)

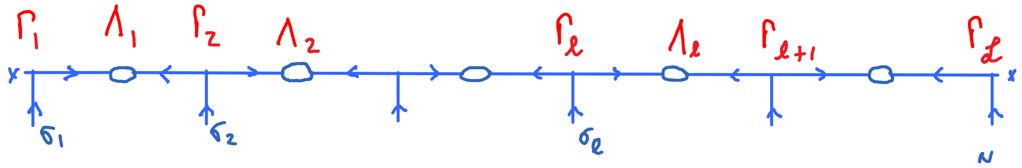
$$\rho_R = \mathrm{Tr}_L |\psi\rangle\langle\psi| = \sum_\alpha |\Phi_\alpha\rangle_{\ell+1} \underbrace{[\Lambda_\ell^\dagger]_{\alpha\alpha}[\Lambda_\ell]^{\alpha\alpha}}_{[\rho_{\ell,R}]^{\alpha}{}_{\alpha}} {}_{\ell+1}\langle\Phi^\alpha| \, ,$$



(4)

Vidal introduced MPS representation in which Schmidt decomposition can be read off for each bond:

$|\psi\rangle$ =



(5)

where Λ_ℓ = diagonal matrix, consisting of Schmidt coefficients for bond ℓ between sites ℓ and $\ell+1$:

$$|\psi\rangle = |\Psi_\alpha\rangle_\ell \, |\Phi_\beta\rangle_{\ell+1} \, \Lambda_\ell^{\alpha\alpha} \, , \qquad \rho_{\ell,L} := \Lambda_\ell \Lambda_\ell^\dagger = \Lambda_\ell^\dagger \Lambda_\ell =: \rho_{\ell,R} \qquad (6)$$

with orthonormal sets on L:

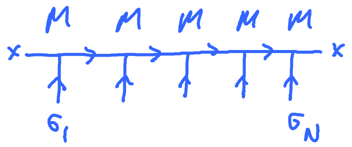
$${}_\ell\langle\Psi^{\alpha'}|\Psi_\alpha\rangle_\ell = \mathbb{1}^{\alpha'}{}_\alpha \qquad (7)$$

and on R:

$${}_\ell\langle\Phi^{\beta'}|\Phi_\beta\rangle_\ell = \mathbb{1}^{\beta'}{}_\beta \qquad (8)$$

Any MPS can always be brought into $\Gamma\Lambda$ form. Proceed in same manner as when left-normalizing, [cf. MPS-I.4]

$$|\psi\rangle = |\vec{\sigma}\rangle_N \left(M^{\sigma_1} \ldots M^{\sigma_N}\right)$$



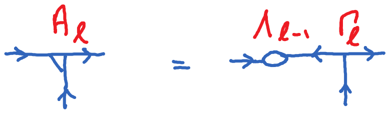
(9)

Successively use SVD on pairs of adjacent tensors: $MM' = \underbrace{U}_{A}\underbrace{SV^\dagger M'}_{\tilde{M}} := A\tilde{M}$, (10)

to bring MPS into left-canonical form,

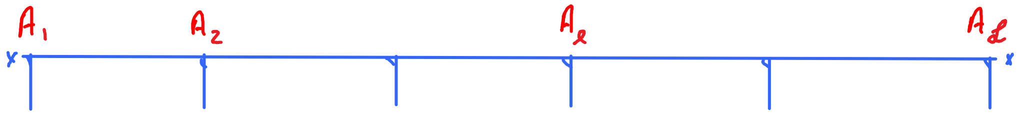
store singular values, $\Lambda_\ell := S_\ell$ and at end define

$A_\ell^{\sigma_\ell} =: \Lambda_{\ell-1}\Gamma_\ell^{\sigma_\ell}$, $\Lambda_0 = 1$

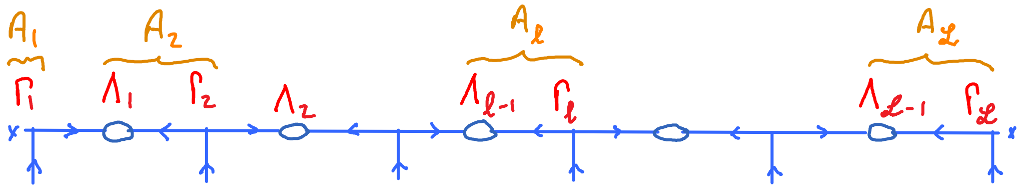


(11)

physical index σ_ℓ of A_ℓ is associated with Γ_ℓ

$|\psi\rangle =$


(12)

$=:$


(13)

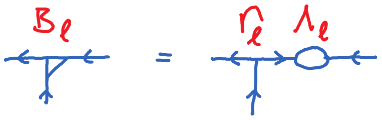
Note: in numerical practice, this involves dividing by singular values, $\Gamma_\ell^{\sigma_\ell} := \Lambda_{\ell-1}^{-1} A_\ell^{\sigma_\ell}$ (14)

So, first truncate states for which $S_{\ell-1}^{\alpha\alpha} = 0$, (15)

Even then, the procedure can be numerically unstable, since arbitrarily small singular values may arise. So, truncate states for which (say) $S_{\ell-1}^{\alpha\alpha} < 10^{-8}$. In practice, this should be done in (16) any case, because when computing norms and matrix elements, singular value s contributes weight s^2 and when $s^2 < 10^{-16}$, its contribution gets lost in numerical noise. Inverting the remaining singular values, $s > 10^{-8}$, is unproblematic in numerical practice.

Similarly, if we start from the right, SVDs yield right-normalized B-tensors, and we can define

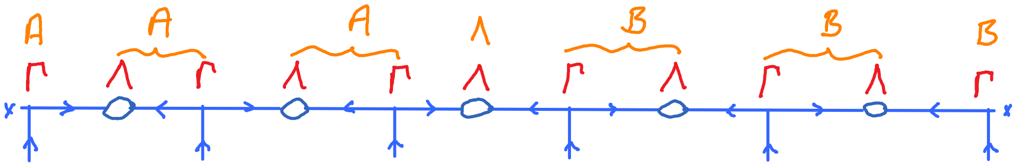
$B_\ell^{\sigma_\ell} =: \Gamma_\ell^{\sigma_\ell}\Lambda_\ell$, $\Lambda_{\mathscr{L}} = 1$,



(17)

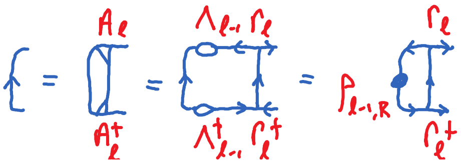
i.e. $\Gamma_\ell^{\sigma_\ell} := B_\ell^{\sigma_\ell}\Lambda_\ell^{-1}$ (18)

So, relation between standard bond-canonical form and 'canonical $\Gamma\Lambda$ form' is:

$|\psi\rangle =$


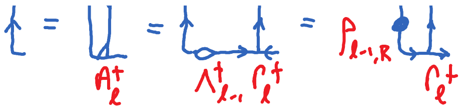
(19)

$\mathbb{1} = A_\ell^\dagger A_\ell = \Gamma_\ell^\dagger \Lambda_{\ell-1}^\dagger \Lambda_{\ell-1}\Gamma_\ell = \Gamma_\ell^\dagger \rho_{\ell-1,R}\Gamma_\ell$,

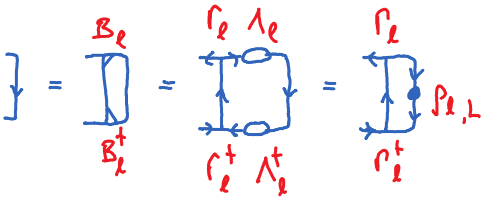


(20)

$$\mathbb{1} = A^\dagger_\ell A_\ell = \Gamma^\dagger_\ell \Lambda^\dagger_{\ell-1} \Lambda_{\ell-1} \Gamma_\ell = \Gamma^\dagger_\ell \, \rho_{\ell-1,R} \, \Gamma_\ell \,, \qquad (20)$$



$$\mathbb{1} = B_\ell B_\ell^\dagger = \Gamma_\ell \Lambda_\ell \Lambda_\ell^\dagger \Gamma_\ell^\dagger = \Gamma_\ell \, \rho_{\ell,L} \, \Gamma_\ell^\dagger \,, \qquad (21)$$



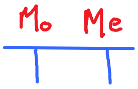
If ρ_ℓ has very small singular values, Γ_ℓ must have large elements! Can lead to unstable behavior…

3. iTEBD: Explicit formulation

[Vidal2007], [Schollwöck2011, Sec. 10.4]

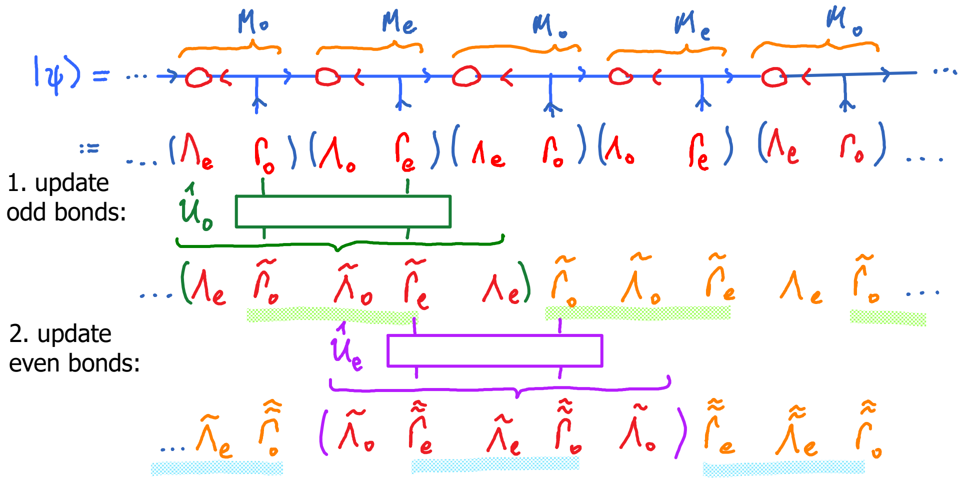
iTEBD.3

For infinite, translationally invariant system, use two-site unit cell, , repeated periodically.



(to avoid cluttering, σ indices on Γ, A are not displayed, but implicitly understood)

Each iTEBD iteration involves two steps, updating first odd bonds, then even bonds:


M_o M_e M_o M_e M_o
$|\psi\rangle = \cdots$
$:= \ldots(\Lambda_e\ \Gamma_o)(\Lambda_o\ \Gamma_e)(\Lambda_e\ \Gamma_o)(\Lambda_o\ \Gamma_e)(\Lambda_e\ \Gamma_o)\ldots$
1. update odd bonds:
$\hat U_o$
$\ldots(\Lambda_e\ \tilde\Gamma_o\ \tilde\Lambda_o\ \tilde\Gamma_e\ \Lambda_e)\ \tilde\Gamma_o\ \tilde\Lambda_o\ \tilde\Gamma_e\ \Lambda_e\ \tilde\Gamma_o\ \ldots$
2. update even bonds:
$\hat U_e$
$\ldots\tilde\Lambda_e\ \tilde{\tilde\Gamma}_o\ (\tilde\Lambda_o\ \tilde{\tilde\Gamma}_e\ \tilde\Lambda_e\ \tilde{\tilde\Gamma}_o\ \tilde\Lambda_o)\ \tilde{\tilde\Gamma}_e\ \tilde\Lambda_e\ \tilde{\tilde\Gamma}_o$

$$M_o = \Lambda_e \Gamma_o$$
$$M_e = \Lambda_o \Gamma_e \qquad (1)$$

yields new $\tilde\Gamma_o\ \tilde\Lambda_o\ \tilde\Gamma_e$

insert these throughout chain

but leave Λ_e unchanged (2a)

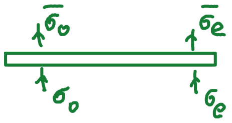
yields new $\tilde{\tilde\Gamma}_e\ \tilde\Lambda_e\ \tilde{\tilde\Gamma}_o$

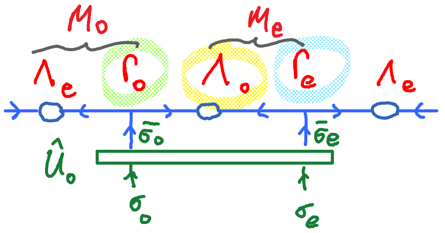
insert these throughout chain

but leave $\tilde\Lambda_o$ unchanged (2b)

Step 1: Time-evolve odd bond $\Gamma_o\Lambda_o\Gamma_e$ and its environment $\Lambda_e \ldots \Lambda_e$

$$\hat U_o := e^{-\tau \hat h_o} = \qquad (3)$$





contract, reshape,

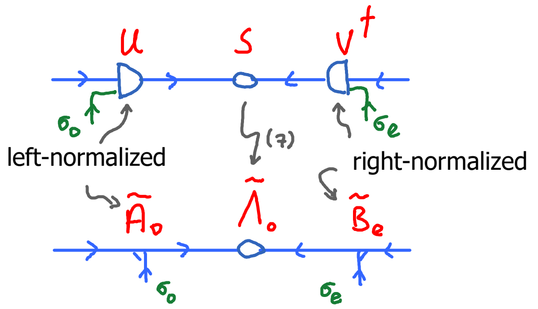
=

SVD

truncate, normalize

≃

reshape



(4a)

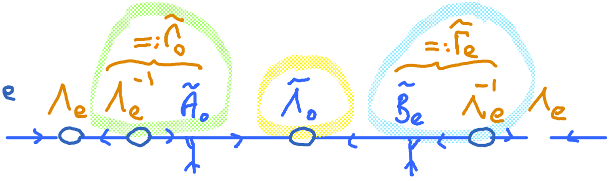
(4b)

define $\tilde\Gamma_o := \Lambda_e^{-1}\tilde A_o$

$\tilde\Gamma_e := \tilde B_e \Lambda_e^{-1}$

reinstate Λ_e

=



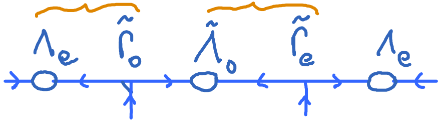
(5)

define

$$\tilde M_e := \tilde\Lambda_o \tilde\Gamma_e = \tilde\Lambda_o \tilde B_e \Lambda_e^{-1}$$

left-normalized → $\tilde A_o$ $=: \tilde M_e$ ← not right-normalized

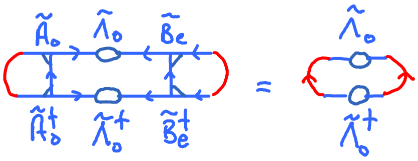
:=



(6)

$\hat U_o$ is projector (not unitary operation), hence reduces norm. Thus, $\tilde\Lambda_o$ is normalized to unity by hand:

$$\tilde\Lambda_o := \frac{S_{trunc}}{[\mathrm{Tr}(S^\dagger_{trunc} S_{trunc})]^{1/2}}$$, then



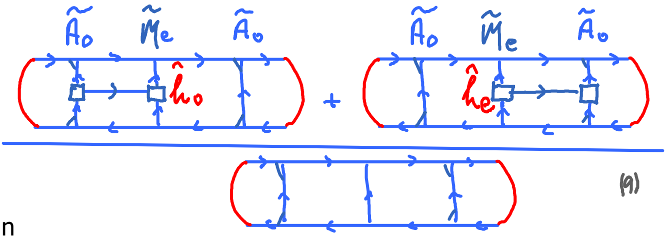
$= \mathrm{Tr}\,\tilde\Lambda_o^\dagger \tilde\Lambda_o = 1 .$ (7)

(6) completes update of odd bond. The updated MPS now has the form $|\tilde\psi\rangle \overset{(6)}{=} \Pi\ \tilde A\ \tilde M_o\ \tilde A_o \tilde M_o$

(6) completes update of odd bond. The updated MPS now has the form $|\tilde\psi\rangle \overset{(6)}{=} \prod_{\vec\sigma} \tilde A_o \tilde M_e \tilde A_o \tilde M_e \dots$ (8)

Updated bond energy :

$$\bar h_{bond} \underset{\uparrow}{=} \frac{1}{2}\left(\bar h_o + \bar h_e\right) \underset{\nearrow}{\simeq}$$



(9)

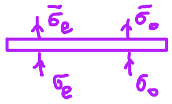
consider only two sites

ignore tensors describing rest of chain

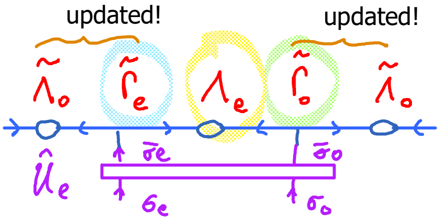
Updating odd bond lowers $\bar h_o$, slightly raises $\bar h_e$ ('odd bond much happier, even bond slightly unhappier').

Step 2: Time-evolve even bond $\tilde\Gamma_e \Lambda_e \tilde\Gamma_o$ and its environment $\hat\Lambda_o \dots \hat\Lambda_o$

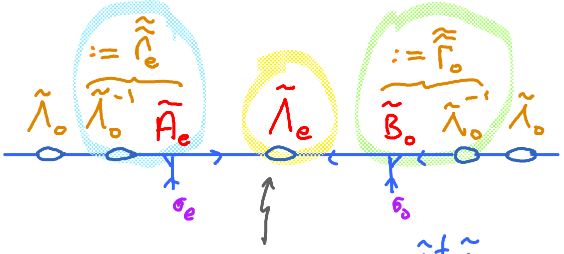
$\hat U_e := e^{-\tau \hat h_e} =$



(10)



contract, reshape, SVD, $\simeq$ reshape, truncate, normalize, reinstate



(11)

with normalization $\mathrm{Tr}\,\tilde\Lambda_e^\dagger \tilde\Lambda_e = 1$ (12)

define

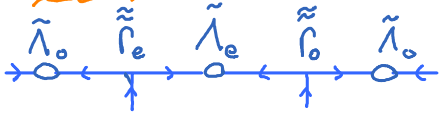
$$\tilde{\tilde\Gamma}_e := \tilde\Lambda_o^{-1} \tilde A_e$$

$$\tilde{\tilde\Gamma}_o := \tilde B_o \tilde\Lambda_o^{-1}$$

$$\tilde M_o := \tilde\Lambda_e \tilde{\tilde\Gamma}_o = \tilde\Lambda_e \tilde B_o \tilde\Lambda_o^{-1}$$

left-normalized $\rightsquigarrow$ $\tilde A_e$ $\tilde M_o$ $\leftarrow$ not right-normalized

$=$



(13)

(12) completes update of even bond. Updated MPS now has the form $|\tilde{\tilde\psi}\rangle = \prod_{\vec\sigma} |\vec\sigma\rangle \tilde M_o \tilde A_e \tilde M_o \tilde A_e$ (14)

Compute updated bond energy using (8), with $o \leftrightarrow e$.

Updating even bond lowers $\bar h_e$, slightly raises $\bar h_o$ ('even bond much happier, odd bond slightly unhappier').

Now iterate: rename $\tilde M_{o,e} \to M_{o,e}$, $\tilde\Lambda_{o,e} \to \Lambda_{o,e}$, $\tilde\Gamma_{o,e} \to \Gamma_{o,e}$

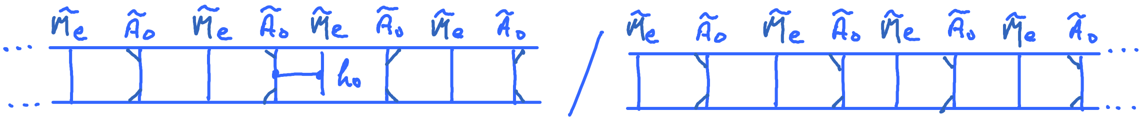
then apply $\hat U_o$, then $\hat U_e$, etc.) until convergence is reached (monitor ground state energy…)

Remarks:

1. In principle, computation of $\tilde\Lambda_o^{-1}$, $\tilde\Lambda_e^{-1}$ can become unstable, because singular values can be very small. Thus: truncate by discarding smallest singular values $\lesssim 10^{-8}$, only then invert.

2. Note that $\tilde A_o$ is left-normalized, but $\tilde M_e \overset{(6,5)}{=} \tilde\Lambda_o \tilde B_e \tilde\Lambda_e^{-1}$ is not! 'Loss of orthogonality'.
$\tilde A_e$ $\qquad \tilde M_o \overset{(13,11)}{=} \tilde\Lambda_e \tilde B_o \tilde\Lambda_o^{-1}$

This causes problems when computing expectation values. For example, odd bond energy, given by


$\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$
h_o
/
$\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$ $\tilde{M}_e$ $\tilde{A}_o$

(15)

does <u>not</u> reduce to (9), because zippers can not be closed from left and right. Hence (9) involves an approximation, namely ignoring the rest of the chain.

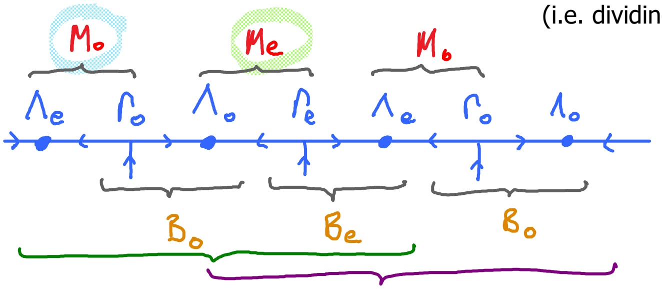
4. iTEBD: Hastings' method (optional)

iTEBD.4

[Hastings2009, Sec. II.A], [Schollwöck2011, Sec. 7.3.2]

Goal: avoid 'reinstatement' of Λ_e, Λ_o, since this requires inverting singular-value matrix. (i.e. dividing by small singular values)

Write $|\psi\rangle =$ (1)



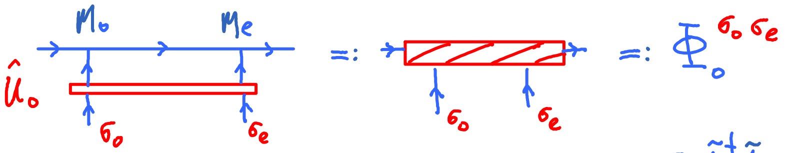
with $M_o = \Lambda_e \Gamma_o \qquad M_e = \Lambda_o \Gamma_e \qquad B_o = \Gamma_o \Lambda_o, \qquad B_e = \Gamma_e \Lambda_e$ (2)

Step 1: Time-evolve odd bond $\underbrace{M_o \Lambda_o B_e}$ to define $\tilde A_o \, \tilde\Lambda_o \, \tilde B_e$ via SVD, and $\tilde M_e$ via contraction. (instead of reinstatement of Λ_e)

Step 2: Time-evolve even bond $\underbrace{\tilde M_e \Lambda_e B_o}$ to define $\tilde A_e \, \tilde\Lambda_e \, \tilde B_o$ via SVD, and $\tilde M_o$ via contraction. (instead of reinstatement of $\tilde\Lambda_o$)

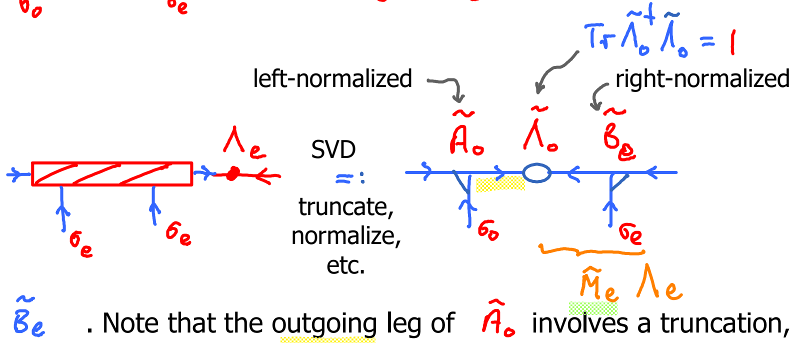
Then rename $\tilde M_{o,e} \to M_{o,e}\,, \quad \tilde\Lambda_{o,e} \to \Lambda_{o,e}\,, \quad \tilde B_{o,e} = B_{o,e}$, and iterate.

Step 1 (odd-even): Compute (3)



known from initialization, or previous iteration

Do SVD on $\Phi_o^{\sigma_o\sigma_e}\Lambda_e =$ (4)



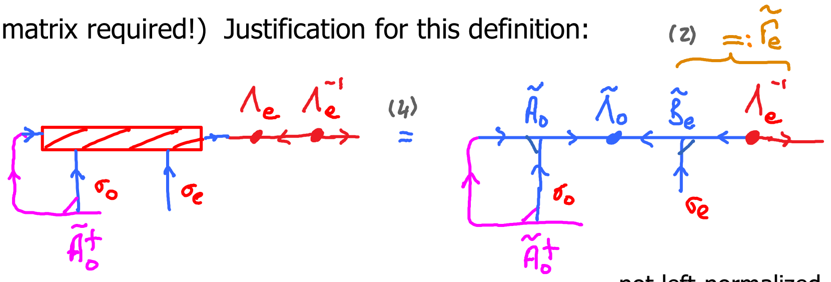
SVD yields updated tensors $\tilde A_o, \tilde\Lambda_o, \tilde B_e$. Note that the outgoing leg of $\tilde A_o$ involves a truncation, governed by $\tilde\Lambda_o$. Since this is also the incoming leg of what will be called $\tilde M_e$, we need a definition of the latter involving a truncation governed by $\tilde\Lambda_o$ on its incoming leg. This is achieved by left-contraction with $\tilde A_o^\dagger$:

$$\tilde M_e^{\sigma_e} := \tilde A^\dagger_{o\sigma_o} \Phi_o^{\sigma_o\sigma_e} \qquad (5)$$

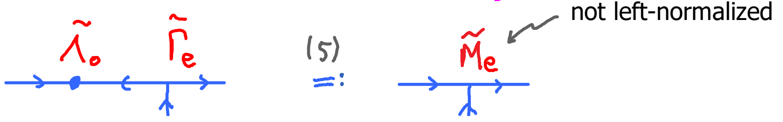
[symbol M denotes: not left-normalized, see (7)]

(Note: no inversion of singular matrix required!) Justification for this definition:

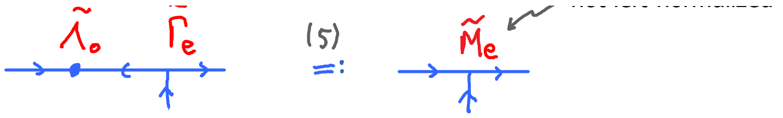
$\tilde A^\dagger_{o\sigma_o}\Phi_o^{\sigma_o\sigma_e} \overset{(3)}{=}$ (6)



$\tilde A^\dagger_{o\sigma_o}\tilde A_o^{\sigma_o} = \mathbb{1}$ $=$ (7)



$\tilde A^{T}_{o\sigma_o} \tilde A^{\sigma_o}_{o} = \mathbb{1}$ $=$



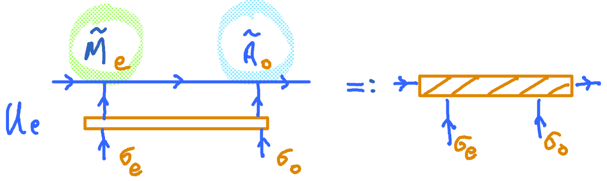
(7)

where we associated $\tilde\Gamma_e := \tilde B_e \Lambda_e^{-1}$ and $\tilde M_e := \tilde\Lambda_o \tilde\Gamma_e$ by analogy to (2) [but did not need Λ_e^{-1} explicitly!] (8)

This concludes step 1. We now have updated tensors

$M_o \to \tilde A_o$, $\Lambda_o \to \tilde\Lambda_o$, $B_e \to \tilde B_e$, $M_e \to \tilde M_e$, but not updated Λ_e , B_o

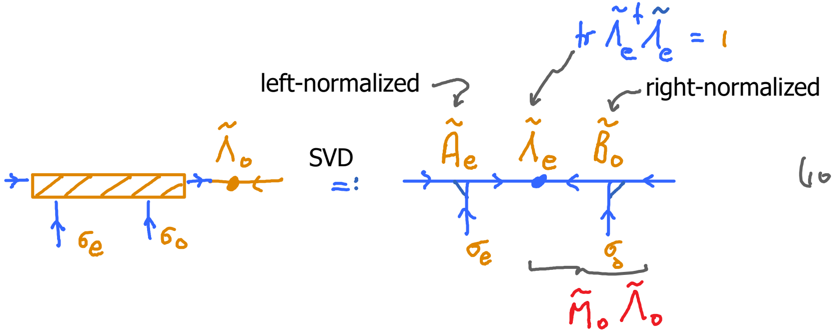
Step 2 (even-odd): Compute



$=: \Phi_e^{\sigma_e\sigma_o}$ (9)

known from step 1:

Do SVD on $\Phi_e^{\sigma_e\sigma_o}\tilde\Lambda_o$ =



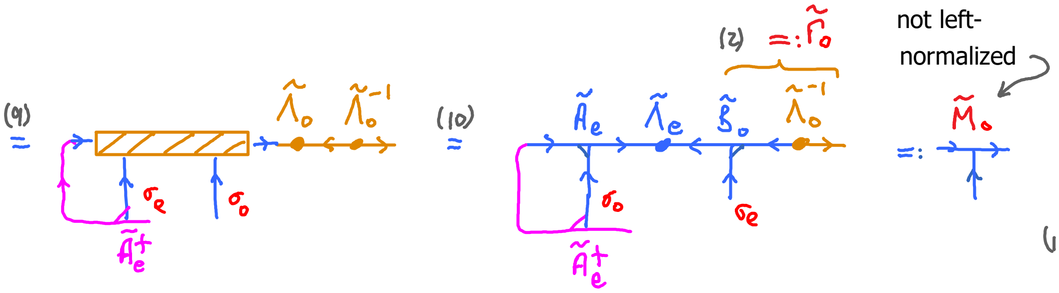
(10)

The SVD yields updated tensors $\tilde A_e, \tilde\Lambda_e, \tilde B_o$, and $\tilde A_e$ has a $\tilde\Lambda_e$ truncation on its outgoing leg, i.e. incoming leg of what will be called $\tilde M_o$, o we need a definition of the latter with $\tilde\Lambda_e$ on incoming leg: This is achieved by:

$$\tilde M_o^{\sigma_o} := \tilde A^{\dagger}_{e\sigma_e} \Phi_e^{\sigma_e\sigma_o}$$ [not left-normalized, see (12)] (11)

Justification:

$\tilde A^{\dagger}_{e\sigma_e} \Phi_e^{\sigma_e\sigma_o}$



(12)

where we associated $\tilde\Gamma_o := \tilde B_o \tilde\Lambda_o^{-1}$ and $\tilde M_o := \tilde\Lambda_e \tilde\Gamma_o$ by analogy to (2) [but did not need $\tilde\Lambda_o^{-1}$ explicitly!] (13)

This concludes step 1. We now have updated tensors

$\tilde M_e \to \tilde A_e$, $\Lambda_e \to \tilde\Lambda_e$, $B_o \to \tilde B_o$, $M_o \to \tilde M_o$ without changing $\tilde\Lambda_o , \tilde B_e$

Now iterate (apply $\hat U_o$, then $\hat U_e$, etc.) until convergence of bond energy is reached.

Compute bond energy using (iTEBD3.9) for step 1, or its $o \leftrightarrow e$ version for step 2.

Concluding remarks:

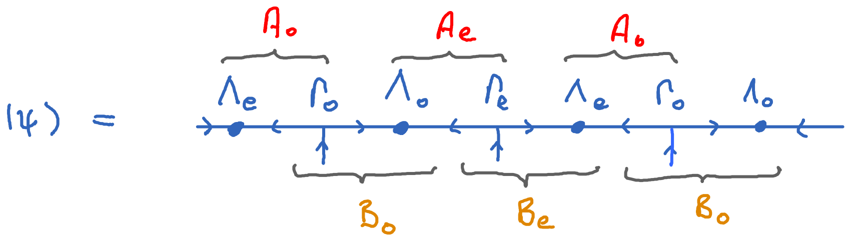
Main advantage of iTEBD: costs not proportional to system size, hence comparatively cheap.

Main disadvantage: loss of orthogonality due to projection, without explicit reorthogonalization.

5. Orthonormalization (optional) [Orus2008] [Schollwöck2011, Sec. 10.5.1] iTEBD.5

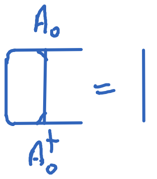
Needed for computing correlators via transfer matrix.

Definition: an infinite, translationally invariant MPS with two-site unit cell, expressed in the form [Orus2008]

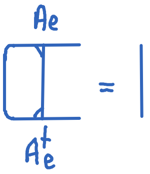


(1)

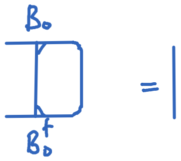
is called 'two-site canonical' if $A_{o,e}$ are left-normalized and $B_{o,e}$ are right-normalized:



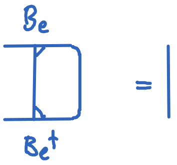
(2a)



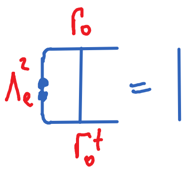
(2b)



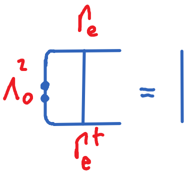
(2c)



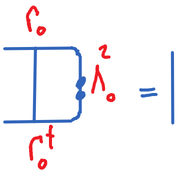
(2d)



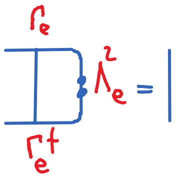
(3a)



(3b)



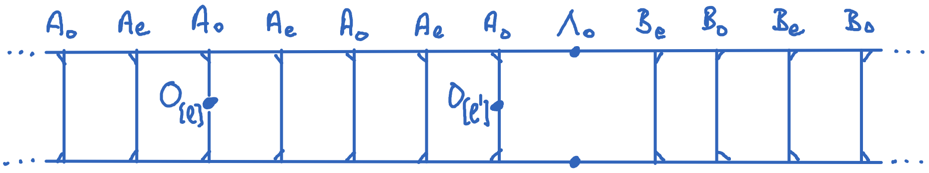
(3c)



(3d)

Correlators can then be computed using transfer matrix methods:

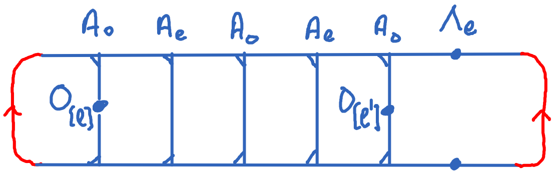
$\langle \hat{O}_{[\ell]} \hat{O}_{[\ell']} \rangle =$



(2)

close zippers

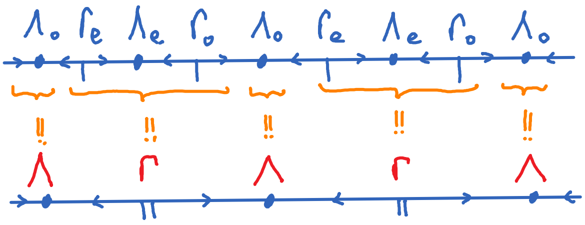
$=$



(3)

Problem: iTEBD (including Hastings' version) yields infinite MPS that are not in canonical form, due to loss of orthogonality. It is possible to restore orthogonality (albeit at the cost of inverting singular value matrices).

Strategy: given $\{\Gamma_e, \Lambda_e, \Gamma_o, \Lambda_o\}$:
2-site unit cell

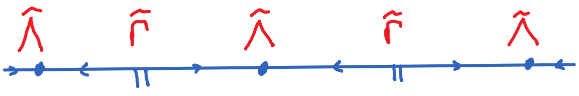


(4)

Step 1: 'coarse-grain' to get $\{\Gamma, \Lambda\}$:
1-site unit cell

$\Gamma := \Gamma_e \Lambda_e \Gamma_o \,, \quad \Lambda := \Lambda_o$

Step 2: bring into 1-site canonical form $\{\tilde{\Gamma}, \tilde{\Lambda}\}$:
1-site unit cell

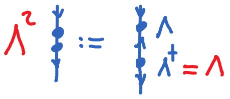


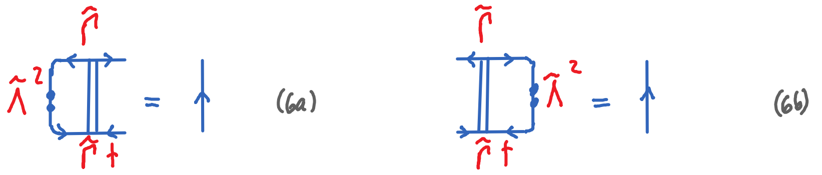
(5)

(how? will be explained further below)

1-site unit cell

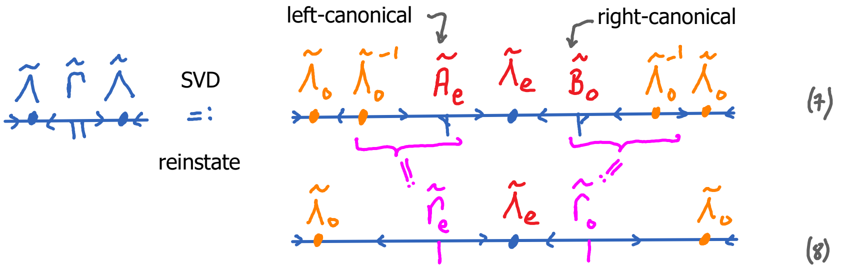
Definition of 1-site canonical:





Step 3: 'fine-grain' via SVD,

reinstate $\tilde\Lambda_o := \tilde\Lambda$

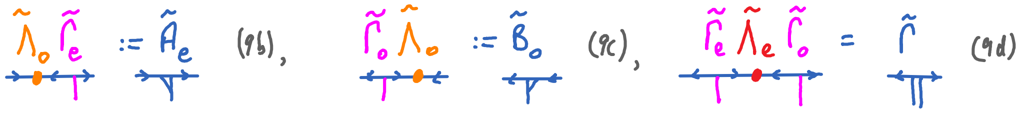


$\tilde\Lambda$ $\tilde\Gamma$ $\tilde\Lambda$
SVD
=:
reinstate
left-canonical
right-canonical
$\tilde\Lambda_o$ $\tilde\Lambda_o^{-1}$ $\tilde A_e$ $\tilde\Lambda_e$ $\tilde B_o$ $\tilde\Lambda_o^{-1}$ $\tilde\Lambda_o$
(7)
‖
‖
$\tilde\Lambda_o$ $\tilde\Gamma_e$ $\tilde\Lambda_e$ $\tilde\Gamma_o$ $\tilde\Lambda_o$
(8)

define $\{\tilde\Gamma_e, \tilde\Lambda_e, \tilde\Gamma_o, \tilde\Lambda_o\}$

2-site unit cell

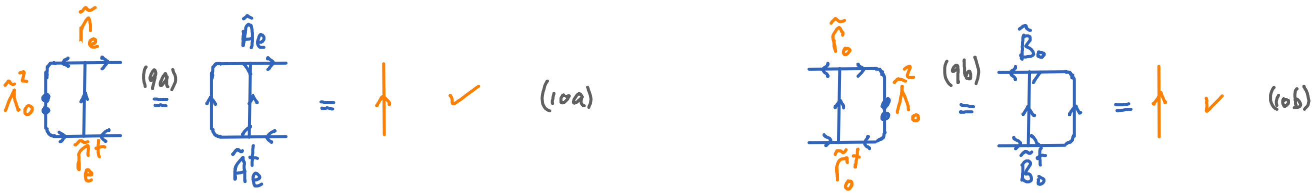
with $\tilde\Lambda_o := \tilde\Lambda$ (9a),



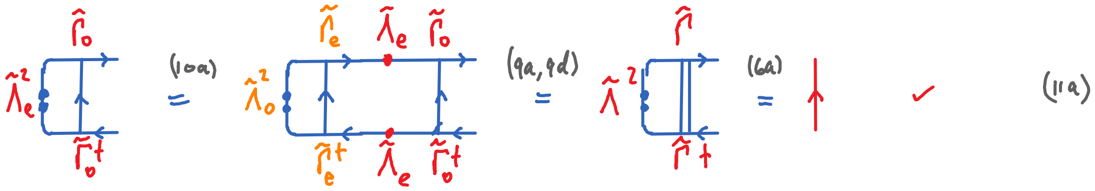
$\tilde\Lambda_o$ $\tilde\Gamma_e$
:=
$\tilde A_e$
(9b),
$\tilde\Gamma_o$ $\tilde\Lambda_o$
:=
$\tilde B_o$
(9c),
$\tilde\Gamma_e$ $\tilde\Lambda_e$ $\tilde\Gamma_o$
=
$\tilde\Gamma$
(9d)

Claim: $\{\tilde\Gamma_e, \tilde\Lambda_e, \tilde\Gamma_o, \tilde\Lambda_o\}$ is in the desired 2-site canonical form.

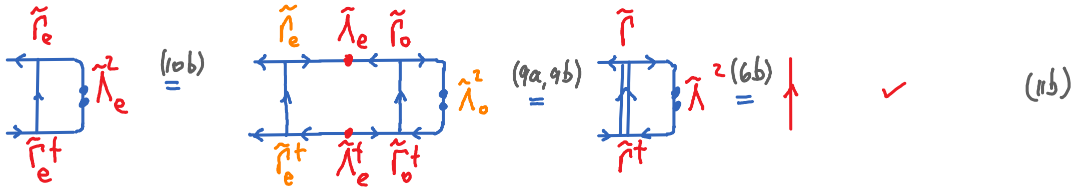
Proof: Since $\tilde A_e$ and $\tilde B_o$ were obtained via SVD, they are left- and right-normalized, respectively. Hence:



Moreover:



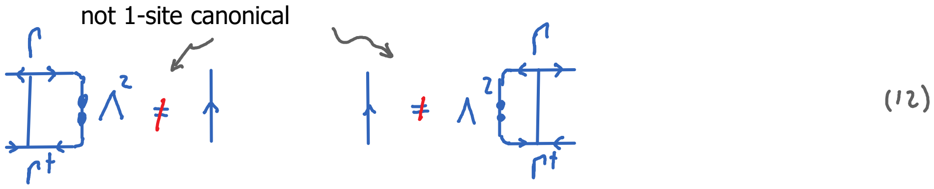
and:



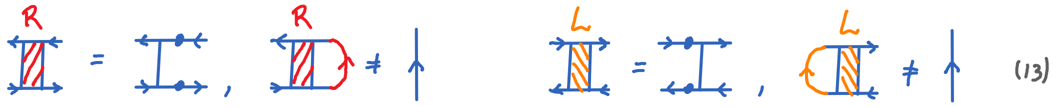
Back to step 2: How to bring arbitrary $\{\Gamma\ \Lambda\}$ into 1-site canonical form $\{\tilde\Gamma, \tilde\Lambda\}$:

Starting point:

(henceforth we draw single line for double physical index)

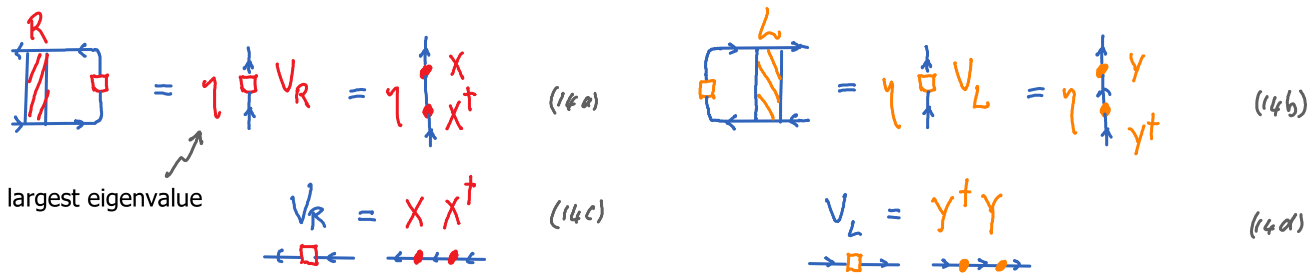


Thus, corresponding transfer matrices are not normalized:



Goal: normalize them! Strategy: 'divide' Γ by the 'square roots' of their dominant right- or left-eigenvectors.

Find dominant right- or left-eigenvectors of R and L, and take their 'square root':



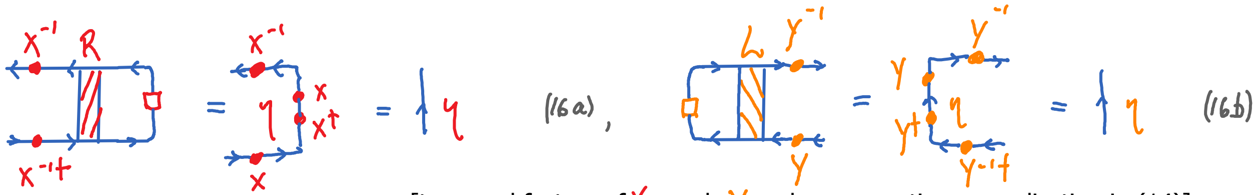
Since R and L are constructed as products' of sets of non-orthogonal vectors, their eigenvectors V_R and V_L are Hermitian and non-negative, hence their 'square roots' exist. They can be found via diagonalization:

E.g.:

$$V_R = W D W^\dagger = (W\sqrt{D})(\sqrt{D}\,W^\dagger) = X \; X^\dagger \qquad (15)$$

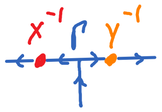
$$V_L = W' D' W'^\dagger = (W'\sqrt{D'})(\sqrt{D'}\,W'^\dagger) = Y^\dagger Y \qquad (15b)$$

Then



[to cancel factors of X and Y when computing normalization in (14)]

So,



might yield a properly normalized transfer matrix. Express MPS through such an object.

Insert identities:

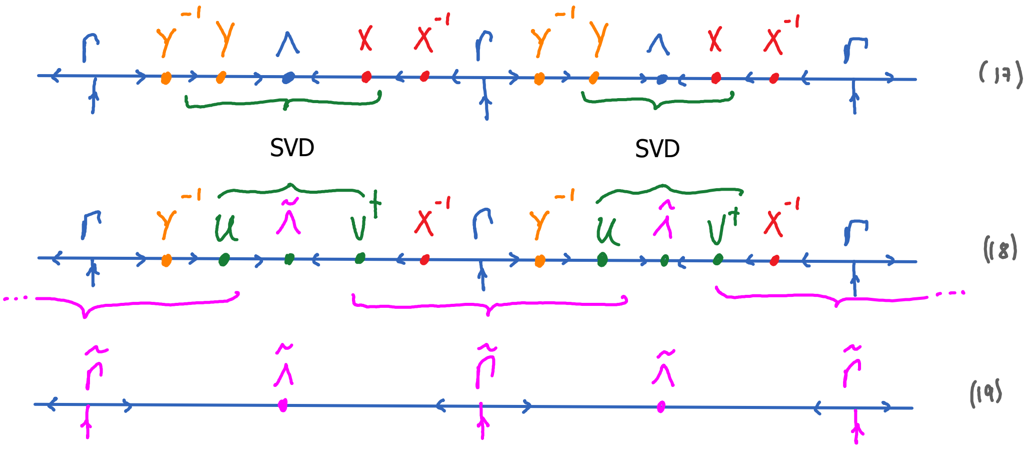
Define new $\tilde\Lambda$ via SVD:

$$Y\Lambda X = U\tilde\Lambda V^\dagger$$

$$U^\dagger U = 1, \; V^\dagger V = 1$$

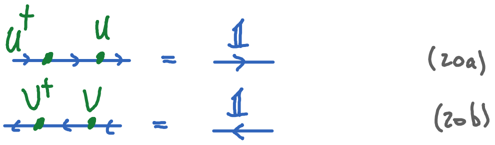
Gather remaining factors into $\tilde\Gamma$

$$\tilde\Gamma = V^\dagger X^{-1}\Gamma Y^{-1} U$$

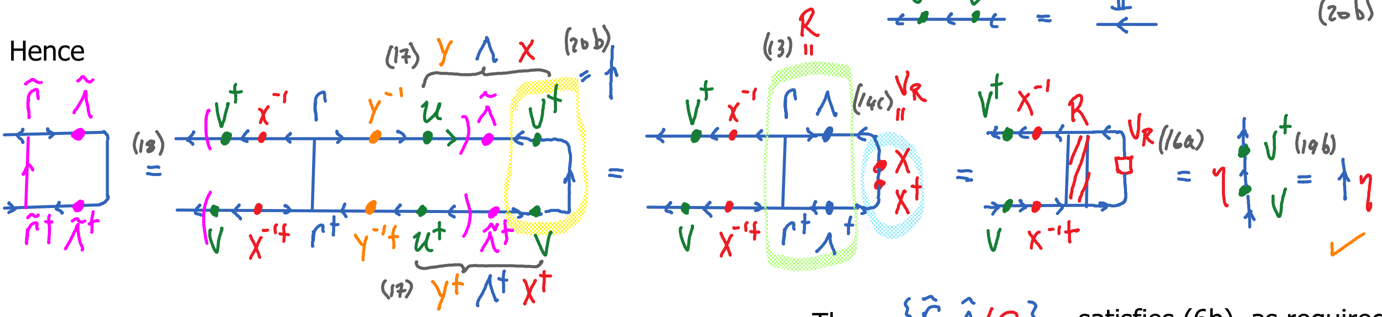

Γ Y^{-1} Y Λ X X^{-1} Γ Y^{-1} Y Λ X X^{-1} Γ
(17)
SVD
SVD
Γ Y^{-1} U $\tilde\Lambda$ $V^\dagger$ X^{-1} Γ Y^{-1} U $\tilde\Lambda$ $V^\dagger$ X^{-1} Γ
(18)
$\tilde\Gamma$ $\tilde\Lambda$ $\tilde\Gamma$ $\tilde\Lambda$ $\tilde\Gamma$
(19)

Claim: $\{\tilde\Gamma, \tilde\Lambda/\sqrt{\eta}\}$ is in the desired 2-site canonical form.

Proof: Since U and $V^\dagger$ were obtained via SVD, they satisfy

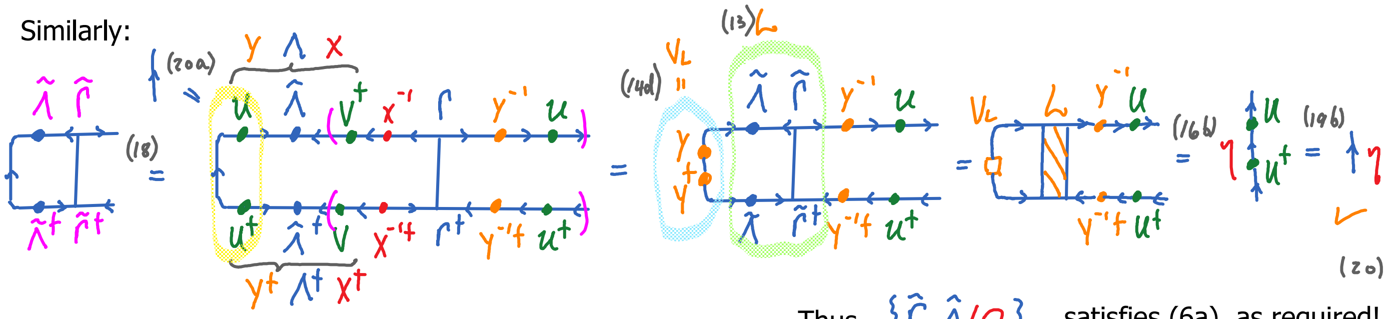


Hence


$\tilde\Gamma$ $\tilde\Lambda$
$\tilde\Gamma^\dagger$ $\tilde\Lambda^\dagger$
(19) =
(17) Y Λ X
(20b)
$V^\dagger$ X^{-1} Γ Y^{-1} U $\tilde\Lambda$ $V^\dagger$
V $X^{-1\dagger}$ $\Gamma^\dagger$ $Y^{-1\dagger}$ $U^\dagger$ $\tilde\Lambda^\dagger$ V
(17) $Y^\dagger$ $\Lambda^\dagger$ $X^\dagger$
=
(13) R
$V^\dagger$ X^{-1} Γ Λ (14c) V_R X $X^\dagger$
V $X^{-1\dagger}$ $\Gamma^\dagger$ $\Lambda^\dagger$
=
$V^\dagger$ X^{-1} R V_R (16a)
V $X^{-1\dagger}$
$= \eta$ $V^\dagger$ V (19b) $= \eta$
✓

Thus, $\{\tilde\Gamma, \tilde\Lambda/\sqrt{\eta}\}$ satisfies (6b), as required!

Similarly:


$\tilde{\Lambda}$ $\tilde{\Gamma}$
$\tilde{\Lambda}^\dagger$ $\tilde{\Gamma}^\dagger$
(18)
=
(20a)
=
Y Λ X
U $\hat{\Lambda}$ $V^\dagger$ X^{-1} Γ Y^{-1} U
$U^\dagger$ $\hat{\Lambda}^\dagger$ V $X^{-1\dagger}$ $\Gamma^\dagger$ $Y^{-1\dagger}$ $U^\dagger$
$Y^\dagger$ $\Lambda^\dagger$ $X^\dagger$
=
(14d)
V_L
=
Y
$Y^\dagger$
(13)
L
$\tilde{\Lambda}$ $\tilde{\Gamma}$ Y^{-1} U
$\tilde{\Lambda}$ $\tilde{\Gamma}^\dagger$ $Y^{-1\dagger}$ $U^\dagger$
=
V_L L Y^{-1} U
$Y^{-1\dagger}$ $U^\dagger$
(16b)
=
η
U
$U^\dagger$
(19b)
=
η
✓
(20)

Thus, $\{\hat{\Gamma}, \hat{\Lambda}/\sqrt{\eta}\}$ satisfies (6a), as required!