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':

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

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

$$e^{-\beta\hat{H}} = \sum_{\alpha} e^{-\beta\hat{H}}|\alpha\rangle\langle\alpha| \xrightarrow{\beta\to\infty} e^{-\beta E_g}|g\rangle\langle g| \tag{2}$$

(α: complete set of energy eigenstates; $|g\rangle\langle g|$: projector onto ground state)

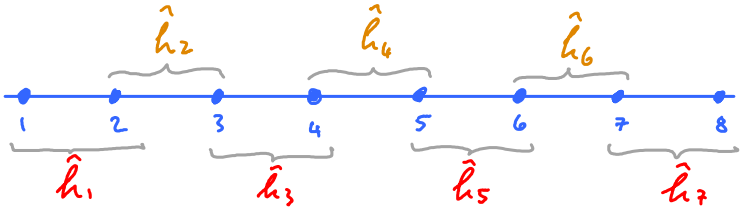
1. Trotter decomposition of time evolution operator

[Schollwöck2011, Sec. 7.1.1]

General: write Hamiltonian as

$$\hat{H} = \sum_{\ell} \hat{h}_\ell = \hat{H}_o + \hat{H}_e \tag{3}$$

($\hat{h}_\ell$: connects sites ℓ and $\ell+1$; $\hat{H}_o$: odd; $\hat{H}_e$: even)

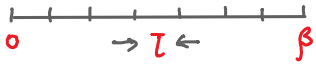


(4)

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

$$[\hat{h}_\ell, \hat{h}_{\ell'}] = 0 \quad \text{if } \ell,\ell' \text{ are both odd or both even} \tag{5}$$

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



(6)

Trotter decomposition:

$$e^{-\beta\hat{H}} = \left[e^{-\tau\hat{H}}\right]^N = \left[e^{-\tau(\hat{H}_o+\hat{H}_e)}\right]^N \tag{7}$$

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

or

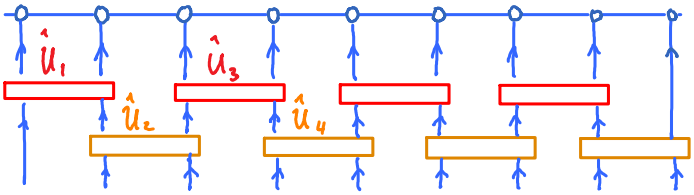
$$\simeq \left[e^{-\frac{\tau}{2}\hat{H}_o}\, e^{-\tau\hat{H}_e}\, e^{-\frac{\tau}{2}\hat{H}_o} + \mathcal{O}(\tau^3)\right]^N \quad \text{'second order Trotter approx.'} \tag{9}$$

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

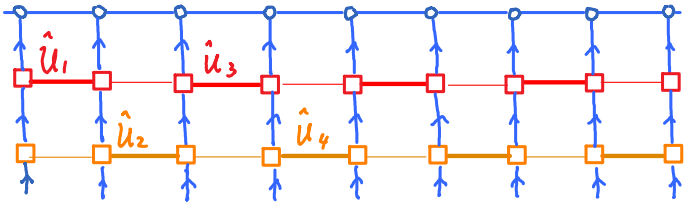
$$e^{-\tau\hat{H}_o} = e^{-\tau\hat{h}_1}\, e^{-\tau\hat{h}_3} \cdots e^{-\tau\hat{h}_{\ell-1}} := \hat{U}_1\,\hat{U}_3 \cdots \hat{U}_{\ell-1} \tag{10a}$$

$$e^{-\tau\hat{H}_e} = e^{-\tau\hat{h}_2}\, e^{-\tau\hat{h}_4} \cdots e^{-\tau\hat{h}_\ell} := \hat{U}_2\,\hat{U}_4 \cdots \hat{U}_\ell \tag{10b}$$

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)

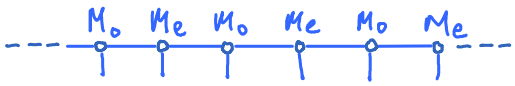
in MPO notation: $=$ 

(12)

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

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

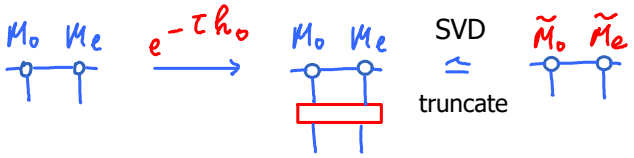
All of this can be done for finite chain of length $\mathcal{L}$. But a simplification occurs for $\mathcal{L} \to \infty$

Then we can exploit translational invariance: 

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

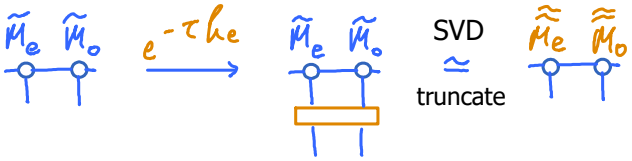
Step 1: time-evolve 'odd bond':

(first site odd, second site even)



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

(first site even, second site odd)



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

iTEBD 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.

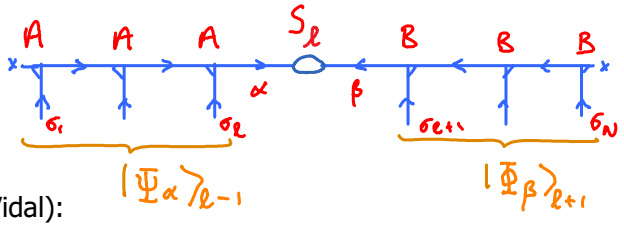
Main advantage of iTEBD: costs not proportional to system size, hence comparatively cheap.

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

to be explained below

Usual bond-canonical form of MPS:

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



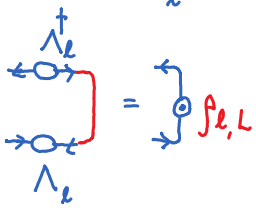
Choose S diagonal, and call it Λ (following Vidal):

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

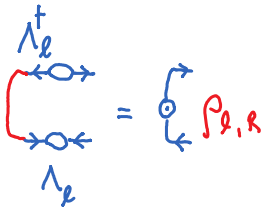
(2)

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

$$\rho_\ell = \mathrm{Tr}_R |\psi\rangle\langle\psi| = \sum_\alpha |\Psi_\alpha\rangle_\ell \underbrace{[\Lambda_\ell]^{\alpha\alpha}[\Lambda_\ell^\dagger]_{\alpha\alpha}}_{[\rho_{\ell,L}]^\alpha{}_\alpha} {}_\ell\langle\Psi^\alpha| \,, \qquad (3)$$

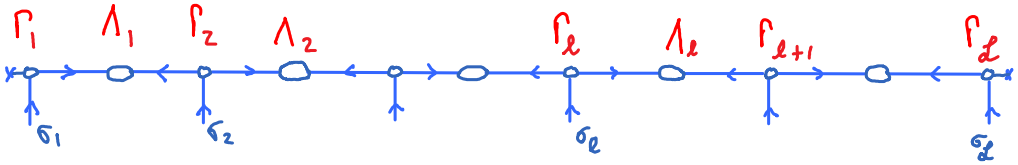


$$\rho_R = \mathrm{Tr}_L |\psi\rangle\langle\psi| = \sum_\alpha |\Phi_\alpha\rangle_{\ell+1} \underbrace{[\Lambda_\ell^\dagger]_{\alpha\alpha}[\Lambda_\ell]^{\alpha\alpha}}_{[\rho_{\ell,R}]^\alpha{}_\alpha} {}_{\ell+1}\langle\Phi^\alpha| \,, \qquad (4)$$



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 $\ell+1$:

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

with orthonormal sets on L:

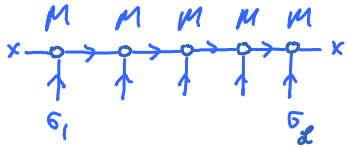
$${}_\ell\langle\Psi^{\alpha'}|\Psi_\alpha\rangle_\ell = \mathbb{1}^{\alpha'}{}_\alpha \qquad (7)$$

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]

$$|\psi\rangle = |\vec{\sigma}\rangle_\ell \, (M^{\sigma_1} \cdots M^{\sigma_\ell}) \qquad (9)$$



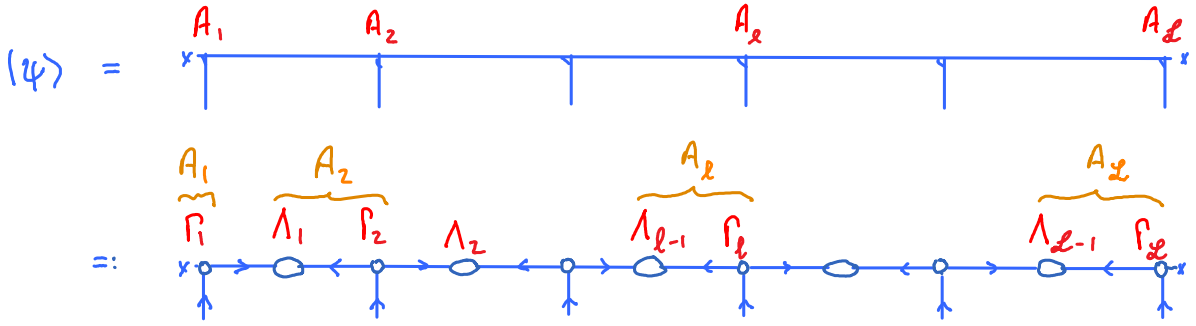
Successively use SVD on pairs of adjacent tensors: $MM' = \underbrace{US}_{A}\underbrace{V^\dagger M'}_{\tilde{M}} := A\tilde{M}$, (10)

to bring MPS into left-canonical form,

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

$$A_\ell^{\sigma_\ell} =: \Lambda_{\ell-1}\Gamma_\ell^{\sigma_\ell}, \quad \Lambda_0 = 1 \qquad (11)$$

physical index σ_ℓ of A_ℓ is associated with Γ_ℓ



(12) (13)

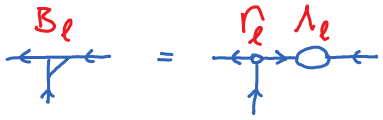
Note: in numerical practice, this involves dividing by singular values, $\Gamma_\ell^{\sigma_\ell} := \Lambda_{\ell-1}^{-1} A_\ell^{\sigma_\ell}$ (14)

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 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. (16)

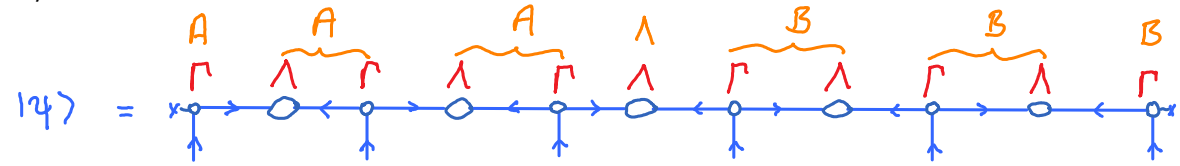
Similarly, if we start from the right, SVDs yield right-normalized B-tensors, and we can define

$$B_\ell^{\sigma_\ell} =: \Gamma_\ell^{\sigma_\ell}\Lambda_\ell, \quad \Lambda_L = 1, \qquad (17)$$



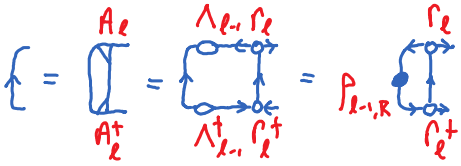
i.e. $\Gamma_\ell^{\sigma_\ell} := B_\ell^{\sigma_\ell}\Lambda_\ell^{-1}$ (18)

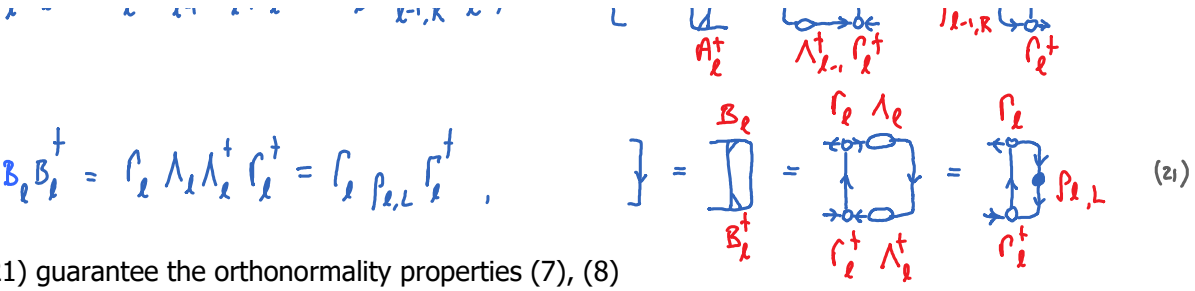
So, relation between standard bond-canonical form and 'canonical $\Gamma\Lambda$ form' is:



(19)

$$1 = A_\ell^\dagger A_\ell = \Gamma_\ell^\dagger \Lambda_{\ell-1}^\dagger \Lambda_{\ell-1}\Gamma_\ell = \Gamma_\ell^\dagger \rho_{\ell-1,R}\Gamma_\ell, \qquad (20)$$





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

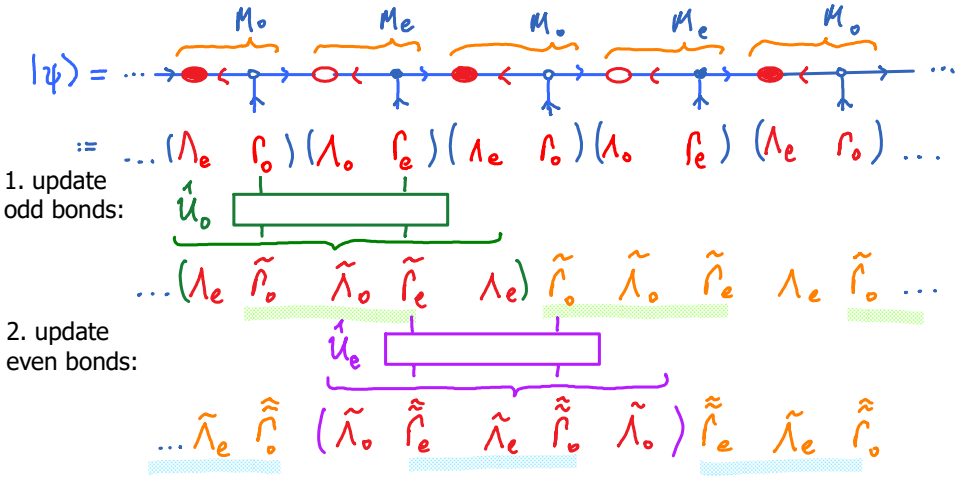
(20), (21) guarantee the orthonormality properties (7), (8)

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

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:



$M_o = \Lambda_e \Gamma_o$

$M_e = \Lambda_o \Gamma_e$ (1)

1. update odd bonds: $\hat{U}_o$

yields new $\tilde{\Gamma}_o \tilde{\Lambda}_o \tilde{\Gamma}_o$

insert these throughout chain

but leave Λ_e unchanged (2a)

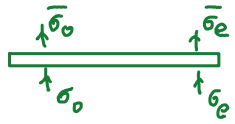
2. update even bonds: $\hat{U}_e$

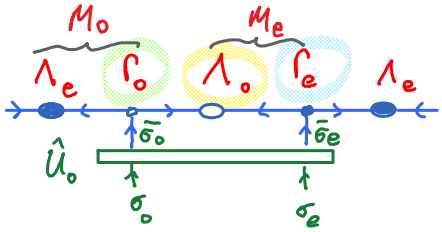
yields new $\tilde{\tilde{\Gamma}}_e \tilde{\Lambda}_e \tilde{\tilde{\Gamma}}_o$

insert these throughout chain

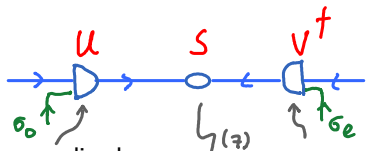
but leave $\tilde{\Lambda}_o$ unchanged (2b)

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

$\hat{U}_o := e^{-\tau \hat{h}_o}$  (3)



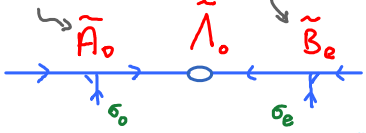
contract, reshape, = SVD



(4a)

left-normalized (7) right-normalized

truncate, normalize [see (7)] ≃ reshape

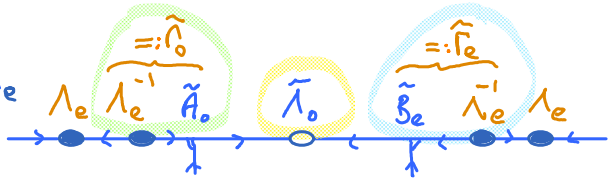


(4b)

define $\tilde{\Gamma}_o := \Lambda_e^{-1} \tilde{A}_o$

$\tilde{\Gamma}_e := \tilde{B}_e \Lambda_e^{-1}$

reinstate Λ_e =

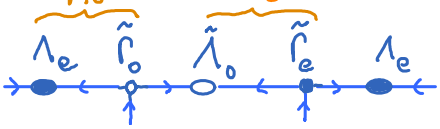


(5)

define $\tilde{M}_e := \tilde{\Lambda}_o \hat{\Gamma}_e = \tilde{\Lambda}_o \tilde{B}_e \Lambda_e^{-1}$

left-normalized $\tilde{A}_o$ $=: \tilde{M}_e$ not right-normalized

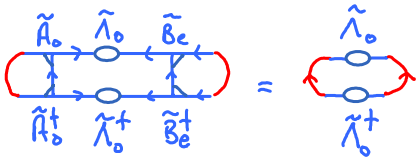
:=



(6)

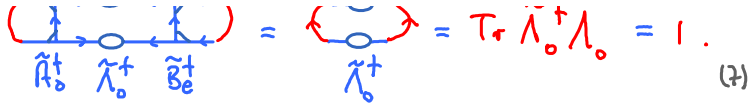
$\hat{U}_o$ is projector (not unitary operation), hence reduces norm. Thus, $\tilde{\Lambda}_o$ is normalized to unity by hand:

$\tilde{\Lambda}_o := \frac{S_{trunc}}{[\mathrm{Tr}(S_{trunc}^\dagger S_{trunc})]^{1/2}}$, 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 $|\tilde{\psi}\rangle \overset{(6)}{=} \sum_{\vec{\sigma}} \tilde{A}_o \tilde{M}_e \tilde{A}_o \tilde{M}_e \dots$ (8)

$\Lambda_o := \frac{S_{trunc}}{[\mathrm{Tr}(S_{trunc}^\dagger S_{trunc})]^{1/2}}$, then  (7)

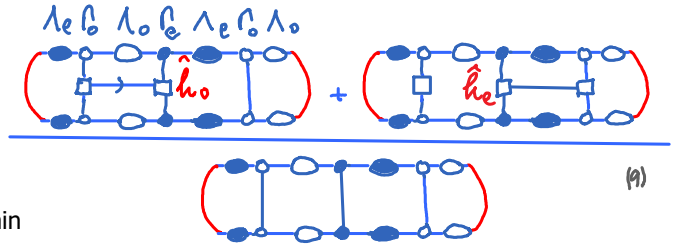
(6) completes update of odd bond. The updated MPS now has the form $|\tilde{\psi}\rangle \overset{(6)}{=} \sum_{\vec{\sigma}} \tilde{A}_o \tilde{\Lambda}_e \tilde{A}_o \tilde{\Lambda}_e \ldots$ (8)

Updated bond energy :

$\bar{h}_{bond} = \frac{1}{2}(\bar{h}_o + \bar{h}_e) \simeq$

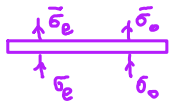
consider only two sites

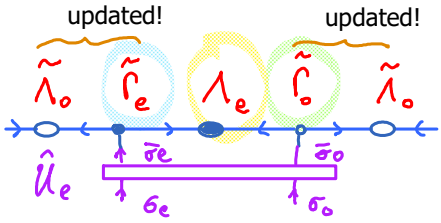
ignore tensors describing rest of chain

 (9)

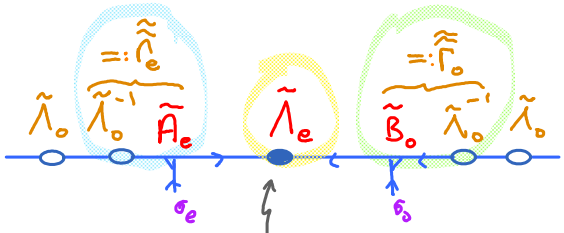
Updating odd bond lowers $\bar{h}_o$, slightly raises $\bar{h}_e$ ('odd bond much happier, even bond slightly unhappier').

Step 2: Time-evolve even bond $\tilde{\Gamma}_e \Lambda_e \tilde{\Gamma}_o$ and its environment $\tilde{\Lambda}_o \ldots \tilde{\Lambda}_o$

$\hat{U}_e := e^{-\tau \hat{h}_e} =$  (10)



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

 (11)

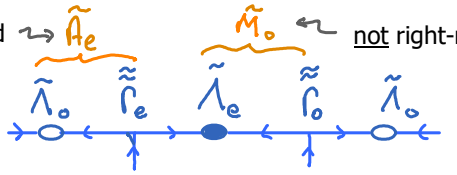
with normalization $\mathrm{Tr}\, \tilde{\Lambda}_e^\dagger \tilde{\Lambda}_e = 1$ (12)

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 $\tilde{A}_e$; $\tilde{M}_o$ not right-normalized

$=$  (13)

(12) completes update of even bond. Updated MPS now has the form $|\tilde{\tilde{\psi}}\rangle = \sum_{\vec{\sigma}} |\vec{\sigma}\rangle \tilde{M}_o \tilde{A}_e \tilde{M}_o \tilde{A}_e$ (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:

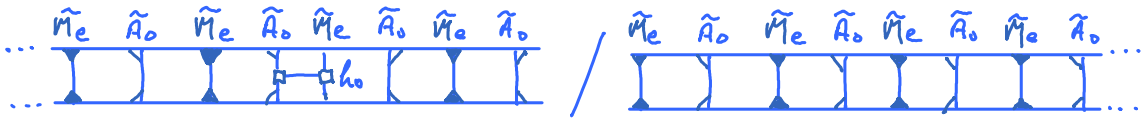
1. In principle, computation of Λ_o^{-1} , $\tilde{\Lambda}_e^{-1}$ can become unstable, because singular values can be very small. Thus: truncate by discarding smallest singular values $\lesssim 10^{-8}$, only then invert.

2. Note that $\tilde{A}_o$ $\tilde{A}_e$ 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'.

$\tilde{M}_o \overset{(13,11)}{=} \tilde{\Lambda}_e \tilde{B}_o \tilde{\Lambda}_o^{-1}$

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

 (15)



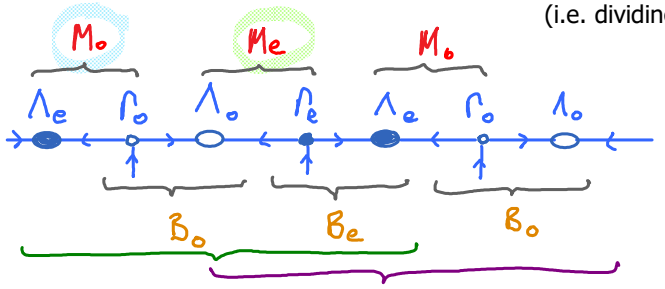
(15)

does not 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)

[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 $|\psi\rangle =$ 

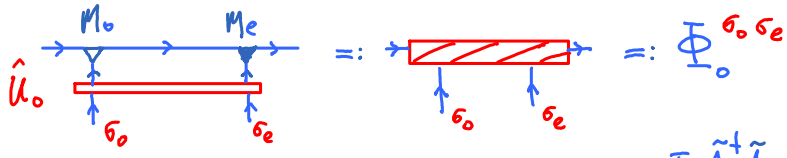
(1)

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

Step 1: Time-evolve odd bond $M_o \Lambda_o B_e$ 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)

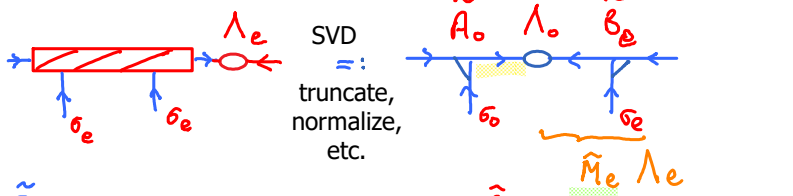
Step 2: Time-evolve even bond $\tilde{M}_e \Lambda_e B_o$ to define $\tilde{A}_e \tilde{\Lambda}_e \tilde{B}_o$ via SVD, and $\tilde{M}_o$ via contraction.
(instead of reinstatement of $\tilde{\Lambda}_o$)

Then rename $\tilde{M}_{o,e} \to M_{o,e}$, $\tilde{\Lambda}_{o,e} \to \Lambda_{o,e}$, $\tilde{B}_{o,e} = B_{o,e}$, and iterate.

Step 1 (odd-even): Compute 

(3)

known from initialization, or previous iteration

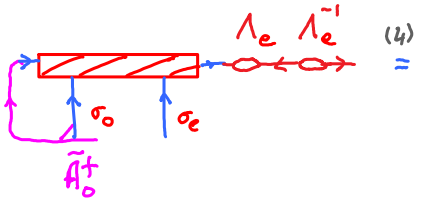
Do SVD on $\Phi_o^{\sigma_o \sigma_e} \Lambda_e$ = 

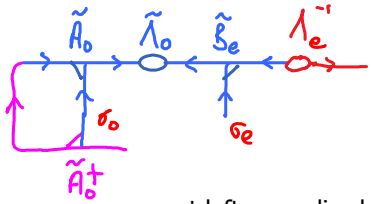
(4)

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 $\tilde{A}_o^\dagger$:

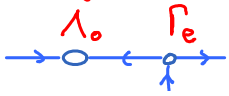
$$\tilde{M}_e^{\sigma_e} := \tilde{A}_{o\sigma_o}^\dagger \Phi_o^{\sigma_o \sigma_e}$$ [symbol M denotes: not left-normalized, see (7)] (5)

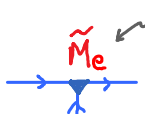
(Note: no inversion of singular matrix required!) Justification for this definition:

$\tilde{A}_{o\sigma_o}^\dagger \Phi_o^{\sigma_o \sigma_e}$ (3) = 

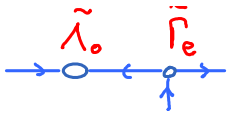
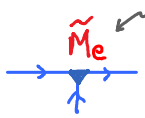
(4) = 

(6)

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

(5) =: 

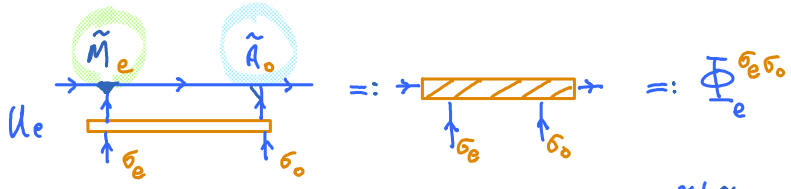
(7)

$\tilde{A}_{o\sigma_o}^{T}\,\tilde{A}_{o}^{\sigma_o} = \mathbb{1}$ =  (5) =:  (7)

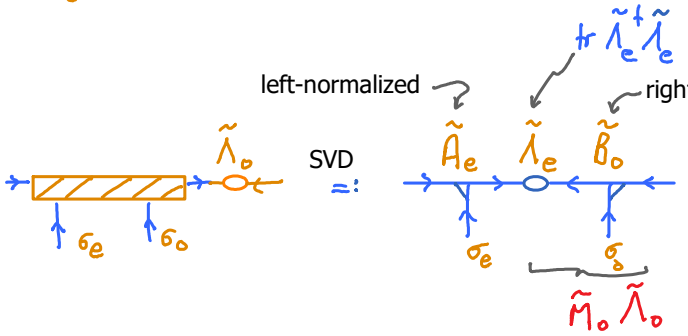
where we associated $\tilde{\Gamma}_e := \tilde{B}_e \Lambda_e^{-1}$ and $\tilde{M}_e := \tilde{\Lambda}_o \tilde{\Gamma}_e$ by analogy to (2) [but did not need Λ_e^{-1} explicitly!] (8)

This concludes step 2. 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  (9)

known from step 1: $\Phi_e^{\sigma_e\sigma_o}$ $\tilde{\Lambda}_o$

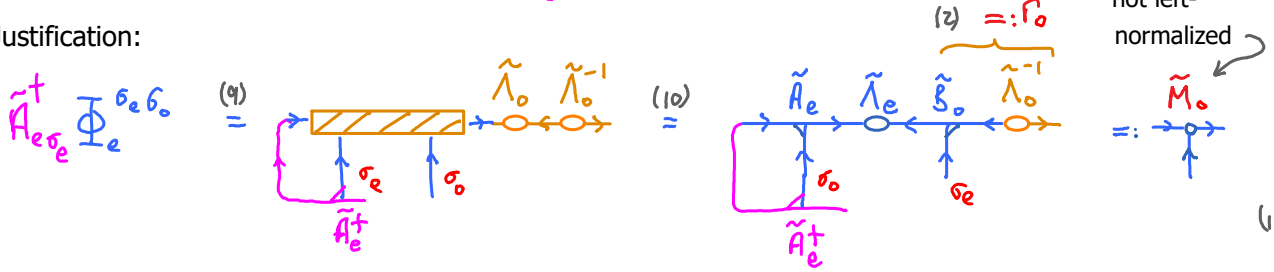
Do SVD on $\Phi_e^{\sigma_e\sigma_o}\,\tilde{\Lambda}_o$ = 

(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 $\tilde{M}_o$, so we need a definition of the latter with $\tilde{\Lambda}_e$ on incoming leg:

This is achieved by:

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

Justification: 

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 2. 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.

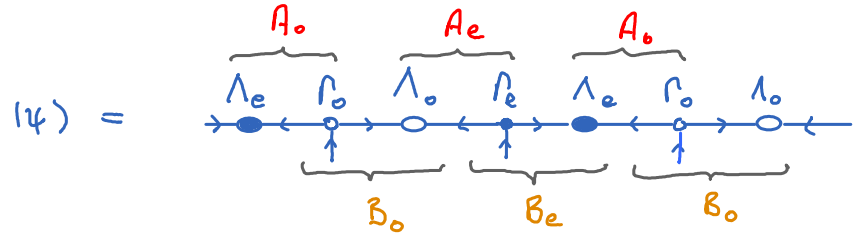
Main disadvantage: loss of orthogonality due to projection, without explicit reorthogonalization.

5. Orthonormalization (optional)

[Orus2008] [Schollwöck2011, Sec. 10.5.1]

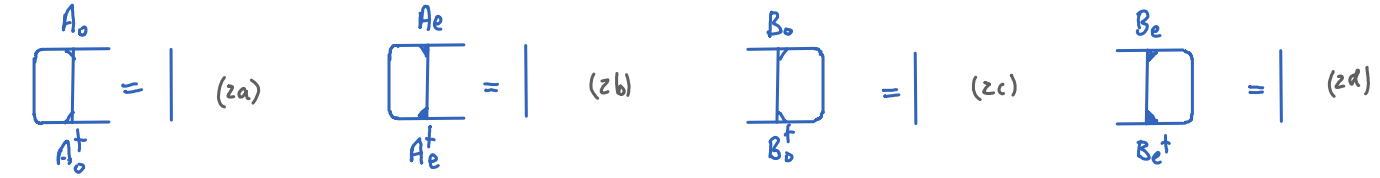
Needed for computing correlators via transfer matrix.

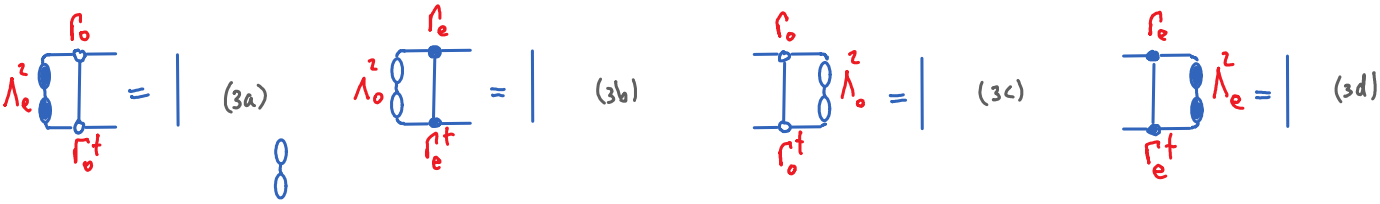
Definition: an infinite, translationally invariant MPS with two-site unit cell, expressed in the form [Orus2008]



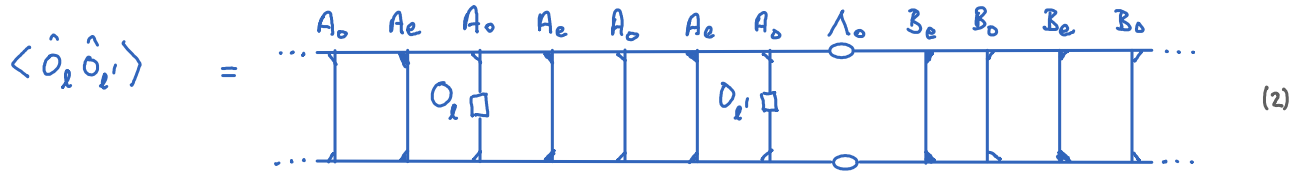
(1)

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



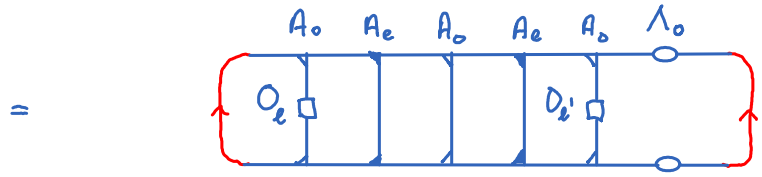


Correlators can then be computed using transfer matrix methods:



(2)

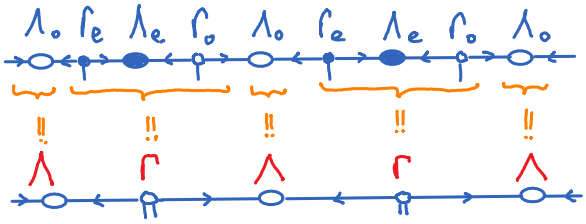
close zippers



(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

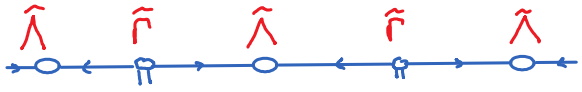


(4)

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

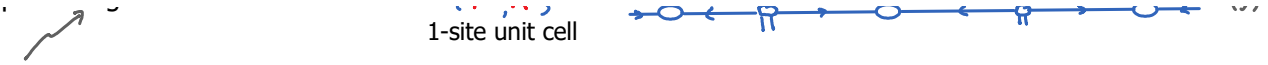
$\Gamma := \Gamma_e \Lambda_e \Gamma_o$, $\Lambda := \Lambda_o$

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



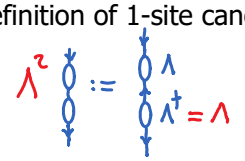
(5)

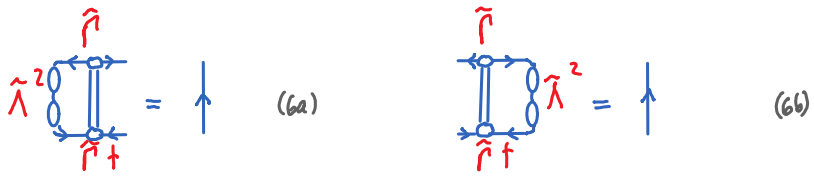
1-site unit cell



(how? will be explained further below)

Definition of 1-site canonical:





(6a) (6b)

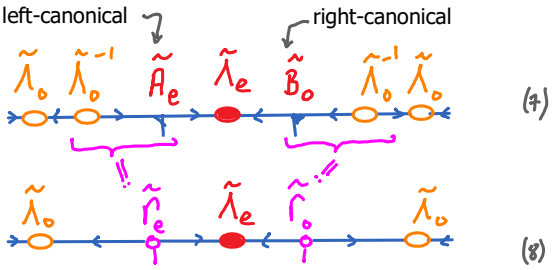
Step 3: 'fine-grain' via SVD,

reinstate $\tilde{\Lambda}_0 := \tilde{\Lambda}$

SVD

=:

reinstate



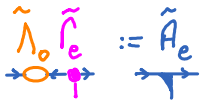
left-canonical right-canonical

(7) (8)

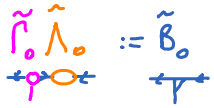
define $\{\tilde{\Gamma}_e, \tilde{\Lambda}_e, \tilde{\Gamma}_o, \tilde{\Lambda}_o\}$

2-site unit cell

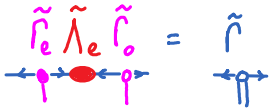
with $\tilde{\Lambda}_0 := \tilde{\Lambda}$ (9a),



(9b),



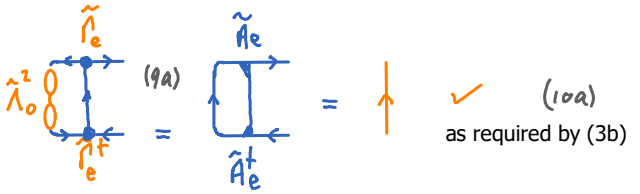
(9c),



(9d)

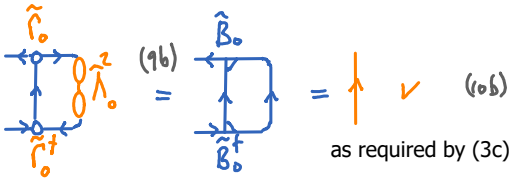
Claim: $\{\tilde{\Gamma}_e, \tilde{\Lambda}_e, \tilde{\Gamma}_o, \tilde{\Lambda}_o\}$ is in the desired 2-site canonical form.

Proof: Since $\tilde{A}_e$ and $\tilde{B}_o$ were obtained via SVD, they are left- and right-normalized, respectively. Hence:



(9a) (10a) ✓

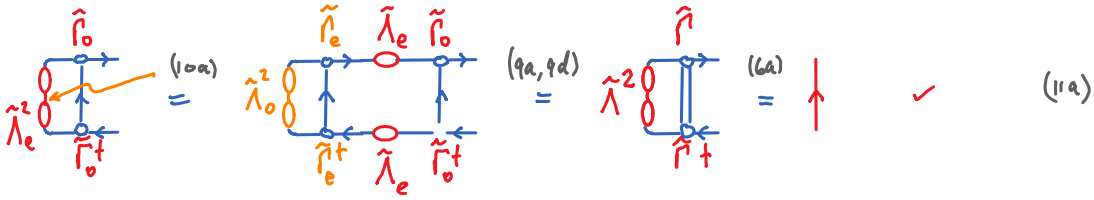
as required by (3b)



(9b) (10b) ✓

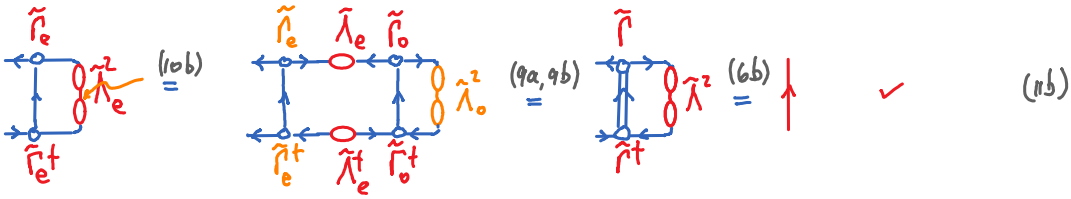
as required by (3c)

Moreover:



(10a) (9a,9d) (6a) ✓ (11a)

and:



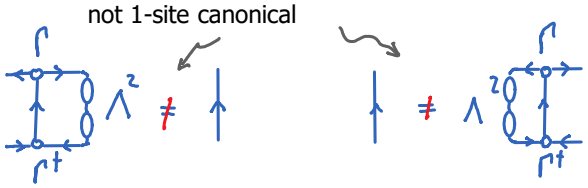
(10b) (9a,9b) (6b) ✓ (11b)

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

not 1-site canonical

Starting point:

(henceforth we draw single line for double physical index)



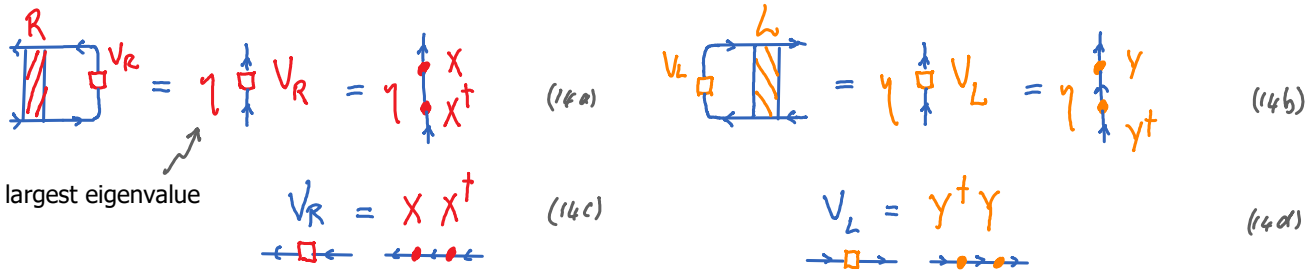
(12)

Thus, corresponding transfer matrices are not normalized:

(13)

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

Find dominant right- or left-eigenvectors of $\mathcal{R}$ and $\mathcal{L}$, and take their 'square root':



Since $\mathcal{R}$ and $\mathcal{L}$ 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:

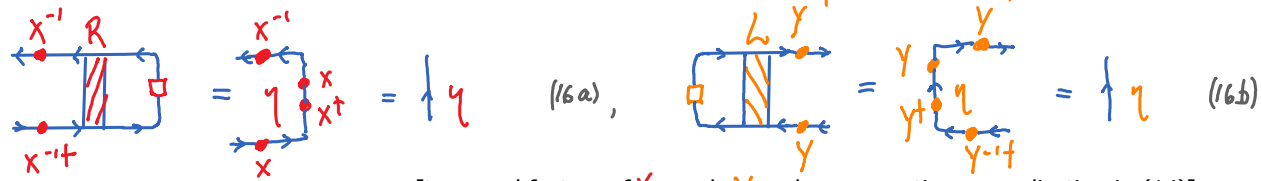
E.g.:

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

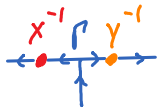
$$V_L = W' D' W'^\dagger = (W'\sqrt{D'})(\sqrt{D'}W'^\dagger) = Y^\dagger Y \qquad (15b)$$

Now divide one leg of each Γ by a 'square root':

Then



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

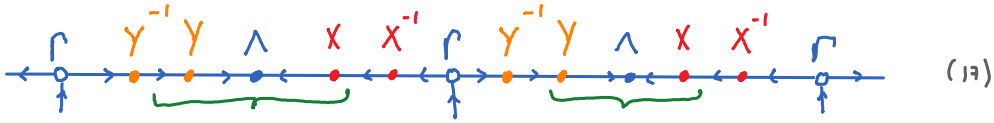
So,  might yield a properly normalized transfer matrix. Express MPS through such an object.

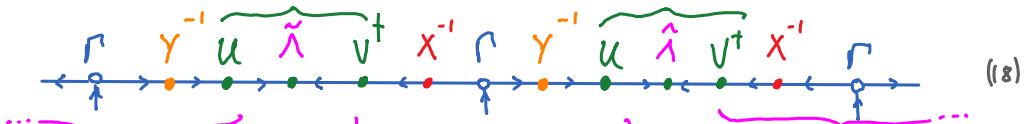
Insert identities:

Define new $\tilde{\Lambda}$ via SVD:

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

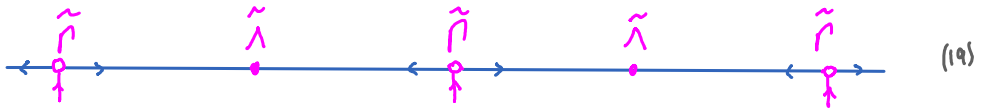
$U^\dagger U = \mathbb{1}$, $V^\dagger V = \mathbb{1}$





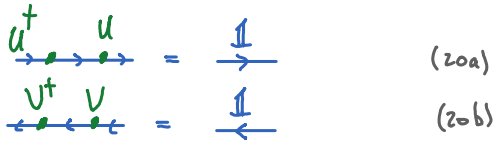
Gather remaining factors into $\tilde{\Gamma}$

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

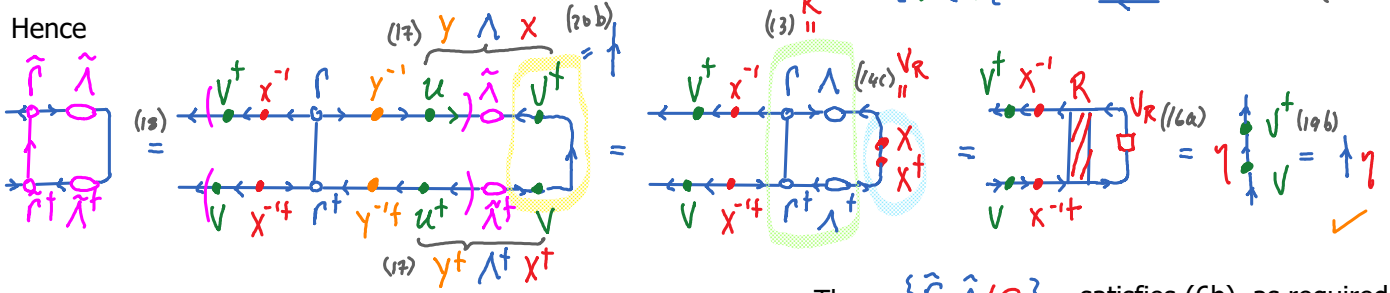


Claim: $\{\tilde{\Gamma}, \tilde{\Lambda}/\sqrt{\eta}\}$ is in the desired 2-site canonical form.

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



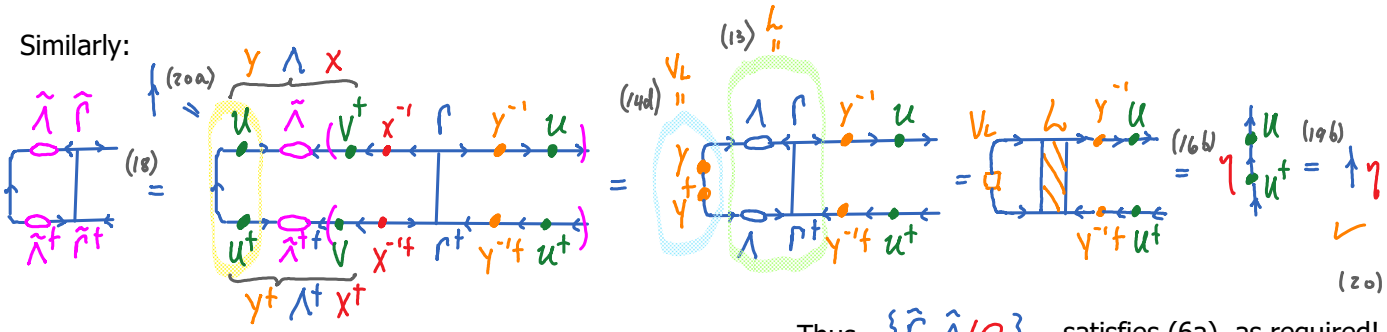
Hence



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

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

Similarly:



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