Lecture 9: Tensor-Network Renormalization Group (TNRG)

Naoki KAWASHIMA

ISSP, U. Tokyo

June 10, 2024

In this lecture, we see ...

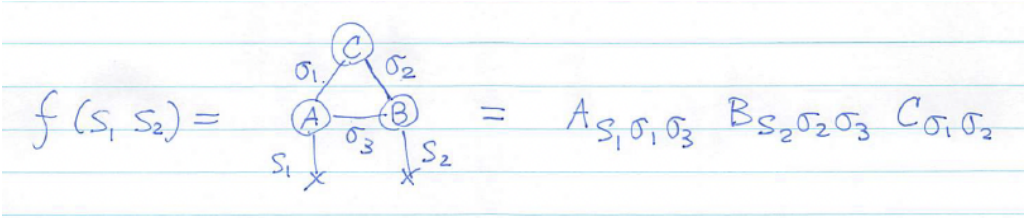
- The Migdal-Kadanoff RG was easy to compute, but, we do not know when we can expect this approximation to be good or how we can improve it systematically.
- Real-space renormalization group method based on tensor-network representation (TNRG) provides us with a method for computing the partition function. While TNRG is also an approximation, it comes with a method for systematic improvements, and may produce the exact critical exponents in the limit.

Tensor-network renormalization group (TNRG)

- Most of statistical-mechanical models on lattices are tensor networks.
- Quantum many-body states on lattices are also described by tensor networks.
- As we have seen, after renormalization transformation, we need infinitely many parameters to describe the resulting system.
- By working with the TN representation, and introducing "data compression" at all length scales, we can overcome the disadvantage of the previous approximates for the real-space RG.

What is a tensor network?

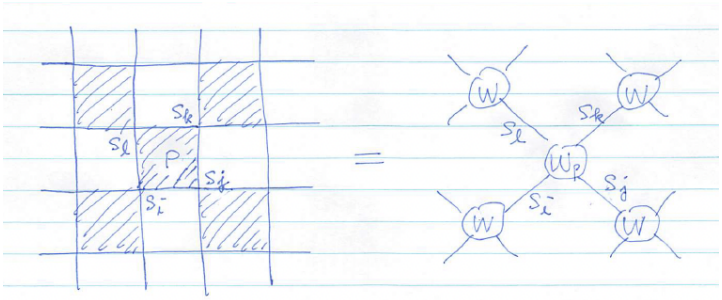
- Consider a graph consisting of circles and crosses connected by lines so that all lines are terminated by circles or crosses. A cross can terminate only one line.
- To each line, we assign an integer variable. To each circle we assign a degree-k tensor $T_{\sigma_1\sigma_2\cdots\sigma_k}$ where k is the number of lines connected to the circle, and σ_k is the variable assigned to each line. We call the variable *virtual* if the line is terminated by two circles, whereas we call it *physical* if an end of the line is terminated by a cross.
- Consider the product of all the tensors, and apply the Einstein convention. The result is some function of physical variables. This is the tensor network representation of the function.



Statistical-mechanical models are tensor networks

The partition function of lattice models can be regarded as a TN with no physical variables. In the case of the Ising model, for example,

$$Z = \sum_{S} \prod_{p=(i,j,k,l)} W_{S_i S_j S_k S_l} \quad \left(W_{S_i S_j S_k S_l} \equiv e^{K(S_i S_j + S_j S_k + S_k S_l + S_l S_i)} \right)$$



We can regard $W_{S_i S_j S_k S_l}$ as a $2 \times 2 \times 2 \times 2$ degree-4 tensor. Then, the above equation is a TN representation of the partition function.

Wave function can be represented as TN

- Consider a quantum many-body system defined on N sites.
- The local Hilbert space H_i $(i = 1, 2, \cdots, N)$ associated with the site i is spanned by the local state vectors, $|S_i\rangle_i$, with S_i taking one of d values.
- The whole Hilbert space is the product of them $H \equiv \bigotimes_i H_i$.
- Any global wave function $|\Psi\rangle$ can be expanded as

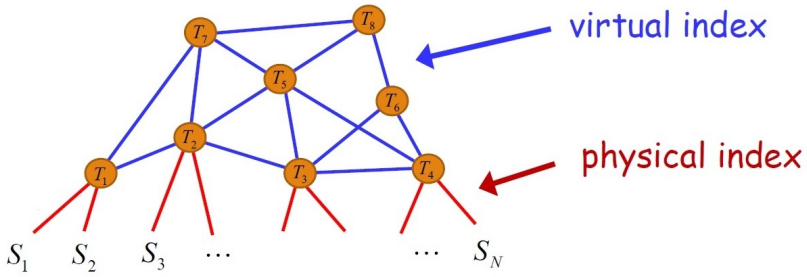
$$|\Psi\rangle \equiv \sum_{\{S_i\}} C_{S_1 S_2 \cdots S_N} |S_1, S_2, \cdots, S_N\rangle \equiv \sum_{\boldsymbol{S}} C_{\boldsymbol{S}} |\boldsymbol{S}\rangle$$

  where $\boldsymbol{S} \equiv (S_1, S_2, \cdots, S_N)$, and $|\boldsymbol{S}\rangle \equiv |S_1\rangle_1 \otimes |S_2\rangle_2 \otimes \cdots \otimes |S_N\rangle_N$.
- In the case of the $S = 1/2$ Heisenberg model, for example, $|\boldsymbol{S}\rangle$ may be the simultaneous eigenstates of the z-components of all the spin operators, i.e., $\hat{S}_i^z |\boldsymbol{S}\rangle = S_i |\boldsymbol{S}\rangle$ where $\hat{S}_i^z$ is the spin operator at i.
- Now, $C_{S_1 S_2 \cdots S_N}$ can be viewed as a degree-N tensor.
- It may be approximated by some tensor network, i.e, $C_{\boldsymbol{S}} \approx \mathrm{Cont}\left(\prod_k T_k\right)$.

TN is a compressed representation of a wave function

$$C_{\boldsymbol{S}} \approx \mathrm{Cont}\left(\prod_k T_k\right)$$

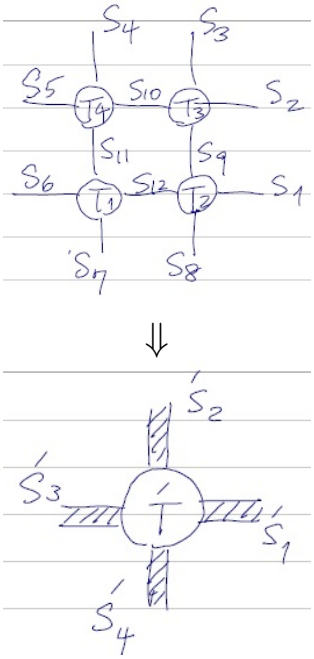
$=$



Note that $C_{\boldsymbol{S}}$ has d^N parameters ($d = 2$ for $S = 1/2$ spin systems), whereas the tensor network can be specified by only $O(N)$ number of parameters. By the tensor network representation, we may be able to reduce the computation for large N down to a manageable level.

Trivial Tensor-network RG

- Let us condider classical systems, and ask how we can use the tensor network for RG.
- How can we replace the original tensor lattice into something similar but with the unit cell bigger than the original?
- Let us solve this problem starting from the trivial TNRG:

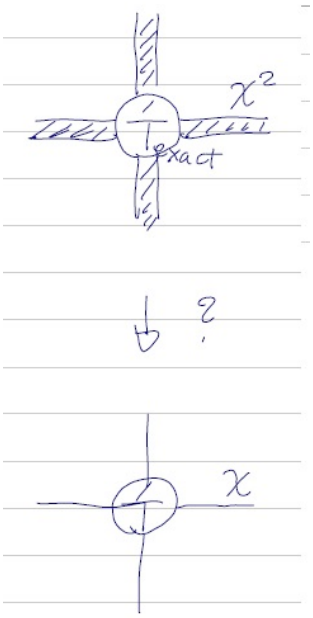


$$T'_{S'_1,S'_2,S'_3,S'_4} \equiv \sum_{S_9,S_{10},S_{11},S_{12}} (T_1)_{S_1,S_9,S_{12},S_8} \times (T_2)_{S_2,S_3,S_{10},S_9} (T_3)_{S_{10},S_4,S_5,S_{11}} (T_4)_{S_{12},S_{11},S_6,S_7}$$

where S'_i is simply "bundled spins", i.e., $S'_1 \equiv (S_1, S_2),\ S'_2 \equiv (S_3, S_4), \cdots$

What's good and bad with trivial TNRG?

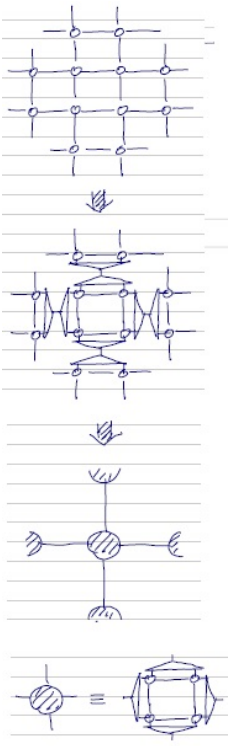
- Using T', we can exactly express the original partition function with lattice constant twice larger than the original, which is good.
- However, the dimension of each index of T' is χ^2 where χ is the index dimension (called "bond dimension" in many papers) of T_i.
- To renormalize a $b \times b$ cluster into a single tensor, its index dimension is χ^b. If we repeat this, it will cause exponential divergence.
- To make the whole procedure practically useful for large systems, we need to keep the index dimension below some constant.



Data compression is necessary!

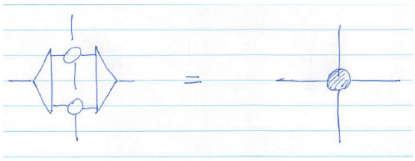
Rank-reducer

- What we need is a 'rank-reducer'.
- A rank-reducer is a matrix with rank χ, instead of χ^b, that approximates the identity matrix.
- If a good rank-reducer exists, we can singular-value-decompose (SVD) it and insert the SVDed rank-reducer as illustrated.
- Then, by cutting the network at the reduced indices and tracing out all the internal degrees of freedom within each cluster, we can define the renormalized tensor with index dimension χ, as we desired.
- Now, we must ask whether such a magical rank-reducer exists or not, and if it does, how we can compute it.

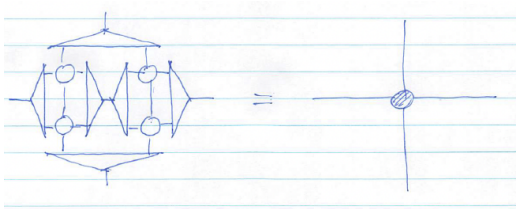


A practical remark

In the previous slide, we compressed 2x2 tensors into one coarse-grained tensor. This is conceptually simpler but technically more expensive than handling each direction one by one. In practice, the single-direction coarse-graining is applied to each direction sequentially. (The following illustrates the "y-direction" coarse-graining.)

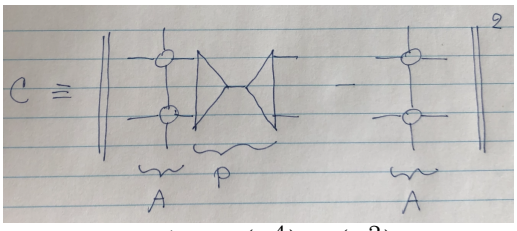


After applying the coarse-graining to y and then x direction, we obtain



The optimal projector

We consider a projector P that satisfies $P^2 = P$ and minimizes the "error" caused by its insertion. Our problem is* to minimize $C \equiv ||AP - A||^2$ with a rank-χ matrix P. With the singular value decomposition $A = V^{\mathsf{T}}\Lambda U$ with $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots)$ $(\lambda_1 \geq \lambda_2 \geq \cdots \geq 0)$, we have



The matrix A is a $(\chi^4) \times (\chi^2)$ matrix whereas P is $(\chi^2) \times (\chi^2)$ with rank-χ.

$$C = \mathrm{Tr}((AP - A)(AP - A)^{\mathsf{T}}) = \mathrm{Tr}(\Lambda^2(1 - U^{\mathsf{T}}PU)).$$

This suggests** that the condition for the optimal P, $U^{\mathsf{T}}PU = 1_\chi$, where 1_χ is the truncated identify matrix with rank χ. Thus we have obtained the optimal projector $P = U1_\chi U^{\mathsf{T}} = \hat{U}\hat{U}^{\mathsf{T}}$ where $\hat{U}$ is the truncated U.

* This is just a compromise between the accuracy and the cost. (It would be better to minimize the error of the whole system, not just A.) Also, preservation of the symmetries, such as reflection, may be also important, though we omit the issue here.

** A more rigorous proof can be made by the EYM theorem.

Supplement: Theorem for Low-Rank Approximation (LRA)

Theorem 1 (Eckhart-Young-Mirsky)

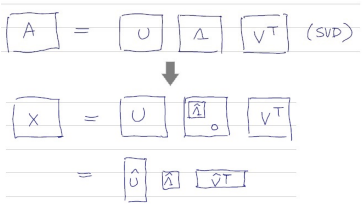
For a given $n \times m$ matrix A, consider its approimation by a rank-l ($l \leq m \leq n$) matrix X and its error $E^2 = |A - X|^2$ where $|A|^2 \equiv \mathrm{Tr}\, A^\mathsf{T} A$. Let $A = U\Lambda V^\mathsf{T}$ be the singular value decomposition (SVD) of A with an $n \times m$ diagonal matrix Λ and n and m dimensional unitaries, U and V, respectively. Then,

$$E^2 \geq \lambda_{l+1}^2 + \lambda_{l+2}^2 + \cdots + \lambda_m^2$$

where λ_i is the i-th largest singular value. The lower bound is attained by $X \equiv \hat{U}\hat{\Lambda}\hat{V}^\mathsf{T}$ where '$\hat{}$' represents truncation at the l-th row/column.

$$\Lambda \equiv \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_m \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}.$$

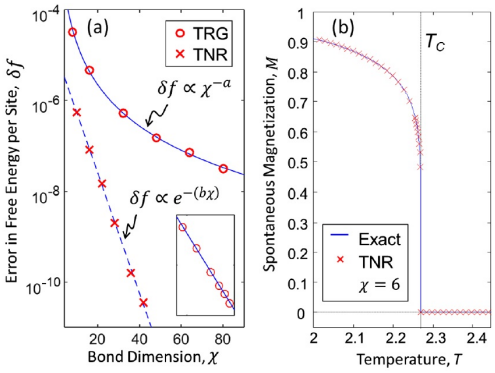
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$$



TNRG provides accurate estimates

- The free energy can be obtained to the accuracy of nearly 8 digits. ("TRG" in the figure.)

  ("TRG" is essentially the same, but technically different way of realizing TNRG from the one discussed in this lecture. See Levin and Nave, Phys. Rev. Lett. 99, 120601 (2007) for details.)
- An improvement ("TNR") pushes it even up to 10 digits.



[Evenbly and Vidal, Physical Review Letters 115, 180405 (2015)]

How we can compute other quantities

From the method described so far, we can obtain F, E, S and C. What about the magnetization, M, χ, and the Binder ratio?

- Define "impurity tensors",

  $T^{(0)} \equiv T, \quad T^{(n)} \equiv 0 \ (n > 1)$

  $T^{(1)}_{S_1 S_2 S_3 S_4} \equiv T_{S_1 S_2 S_3 S_4} \times m(S_1, S_2, S_3, S_4)$

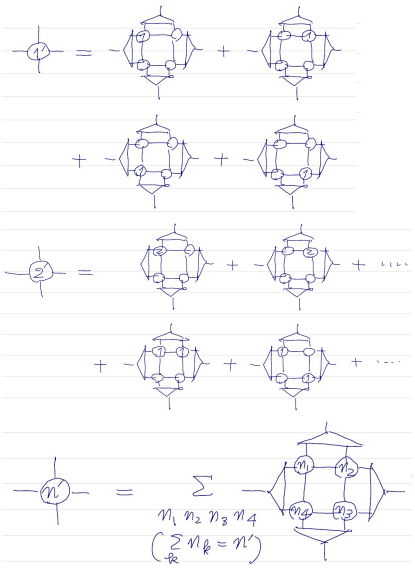
  where $m = (S_1 + S_2 + S_3 + S_4)/2$.

- Define "renormalized impurity tensors":

  $$\acute{T}^{(n)} \equiv \sum_{\substack{n_1 n_2 n_3 n_4 \\ (\sum_k n_k = n)}} \mathrm{Cont}(T^{(n_1)} T^{(n_2)} T^{(n_3)} \times T^{(n_4)} \times (\text{triangle tensors}))$$

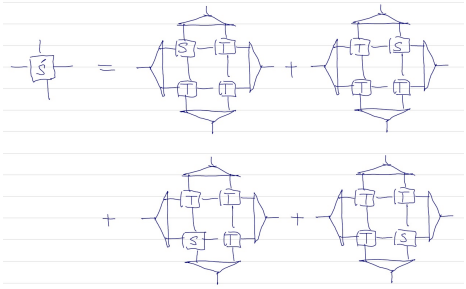
- At the end of all iterations,

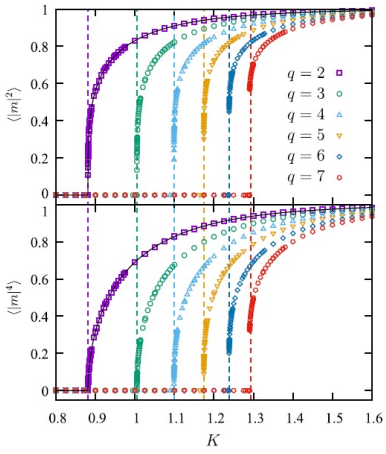
  $$\langle M^n \rangle = \sum_{S_1 S_2} T^{(n)}_{S_1 S_2 S_1 S_2} / \sum_{S_1 S_2} T^{(0)}_{S_1 S_2 S_1 S_2}$$



Application of TNRG to q-state Potts model (1)

- q-state Potts model in 2D.
  [S. Morita and N.K., Computational Physics Communications, 236 65-71 (2019).]
- n-th moments of magnetization are computed (e.g., magnetization ($n = 1$), susceptibility ($n = 2$), Binder ratio ($n = 4$), etc)
- The result of 20 RG iterations (i.e., $L = 2^{20} \approx 10^6$) was obtaind for $q = 2, 3, \cdots, 7$ for the truncation dimension ('bond-dimension') $\chi = 48$.



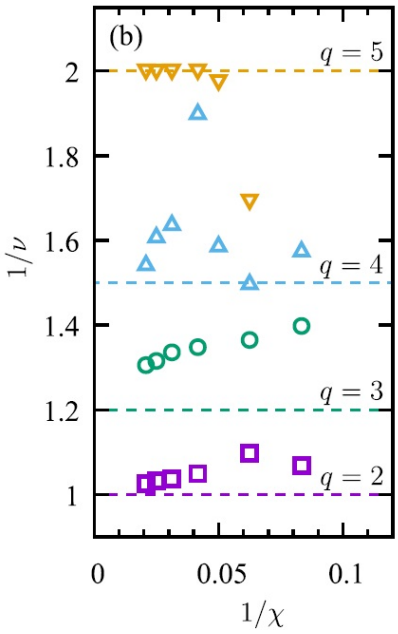


Application of TNRG to q-state Potts model (2)

- According to the finite-size scaling (FSS), which we will discuss later, the Binder ratio is defined as $U_4 \equiv \langle M^4 \rangle / \langle M^2 \rangle^2$ depends on T and L as

$$\left(\frac{dU_4}{dT} \right)_{T=T_c} = \frac{1}{\nu} \log L + a + bL^{-\omega} + \cdots$$

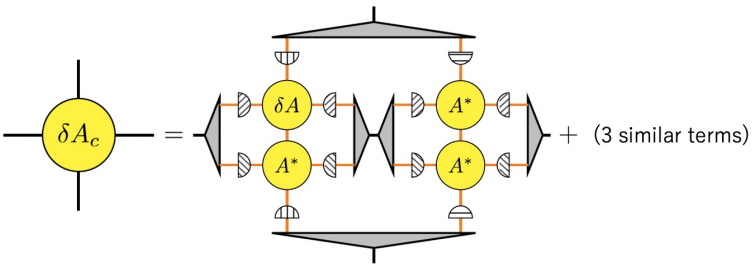
- For first-order transitions, $1/\nu = d$ is expected.
- The 1st order nature of the transition of 5-state Potts model has been confirmed. (CF: $\xi \approx 2500$ at T_c).



[S. Morita and N.K., Comp. Phys. Comm. 236, 65-71 (2019).]

Superoperator and its linearization

The linearized RGT can be expressed in terms of the T^* itself and the projection operators as follows. The scaling dimensions can be obtained by diagonalizing this operator as in the general case.



The diagram means

$$\delta A' = R\, \delta A$$

By diagonalizing R,

$$\phi'_\mu = \lambda_\mu \phi_\mu$$
$$\lambda_\mu = 2^{x_\mu}$$

The linearized TRG. The shaded half circles are the projection operators reducing the dimensions of indices. (Not discussed in the lecture.) The half circles and the triangles are computed for, A^*, the fixed point tensor (T^* in this lecture note). While there are four terms on the right hand side, only one is shown explicitly. For more details, see [Lyu et al, PRR **3** 023048 (2021)].

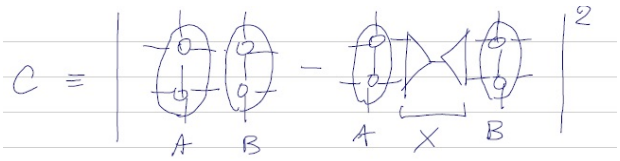
Summary

- Tensor-network RG (TNRG) is a scheme that realizes **"data compression"** at every length scale (by the low-rank approximation of the identity matrix).
- With TNRG, we can systematically improve the real-space RG by adjusting the compression level, i.e., by increasing the cut-off dimension χ (often called "bond-dimension").
- TNRG provides us with rather accurate estimates of various quantities and critical indices.
- We have seen one particular implementation of TNRG, which is called HOTRG. There are many other proposals for realizing TNRG, such as MERA, TRG, TNR, loop-TNR, etc. (See Tao Xiang, "Density matrix and tensor network renormalization", Cambridge 2023)

Appendices

Low-rank approximation (no reflection symmetry)

- How can we optimize the rank-reducer X for the given rank χ?
- For the cost function, we may take the amplitude of the local disturbance caused by the insertion of X, i.e.,



- Let us regard A and B as $\chi^4 \times \chi^2$ matrices and the rank-reducer X as a $\chi^2 \times \chi^2$ matrix whose rank is χ (or less).

Low-rank approximation problem

Suppose 3 integers, l, m, n, that satisfy $l < m < n$. For two given $n \times m$ matrices A and B, find a rank-l, $m \times m$ matrix X that minimizes

$$C(X) \equiv |AB^{\mathsf{T}} - AXB^{\mathsf{T}}|^2. \tag{1}$$

Solution to LRA problem I

- We want the rank-l matrix X minimize

$$C \equiv |AB^{\mathsf{T}} - AXB^{\mathsf{T}}|^2. \tag{2}$$

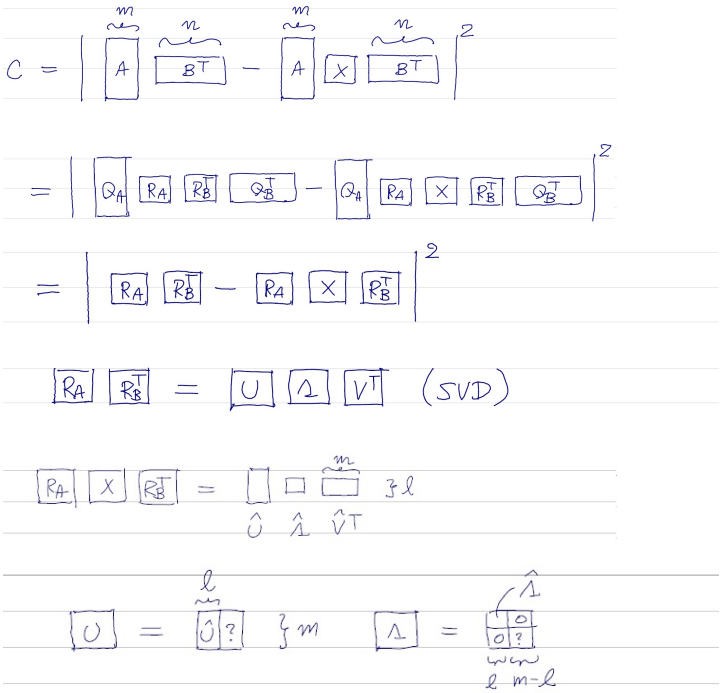
- By the QR-decomposition, $A = Q_A R_A$, $B = Q_B R_B$ with Q_X and R_X the relevant parts of orthogonal and upper triangle matrices, respectively,

$$C \equiv |R_A R_B^{\mathsf{T}} - R_A X R_B^{\mathsf{T}}|^2 \tag{3}$$

- Consider SVD: $R_A R_B^{\mathsf{T}} = U \Lambda V^{\mathsf{T}}$.
- If X satisfies

$$R_A X R_B^{\mathsf{T}} = \hat{U} \hat{\Lambda} \hat{V}^{\mathsf{T}}, \tag{4}$$

it is optimal. (See supplement)



Solution to LRA problem II

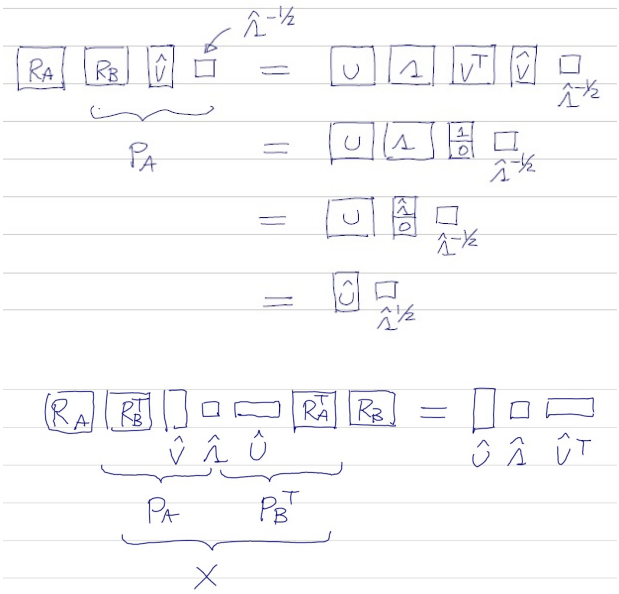
- Now, let us define "triangle operators," P_A and P_B, by

$$P_A \equiv R_B^{\mathsf{T}} \hat{V} \hat{\Lambda}^{-\frac{1}{2}},$$
$$P_B \equiv R_A^{\mathsf{T}} \hat{U} \hat{\Lambda}^{-\frac{1}{2}}$$

- Then, because $R_A R_B^{\mathsf{T}} = U \Lambda V^{\mathsf{T}}$,

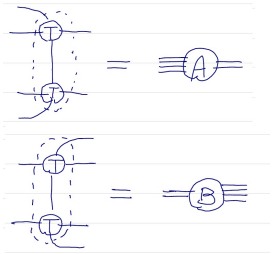
$$R_A P_A = \hat{U} \hat{\Lambda}^{\frac{1}{2}},$$
$$R_B P_B = \hat{V} \hat{\Lambda}^{\frac{1}{2}}.$$

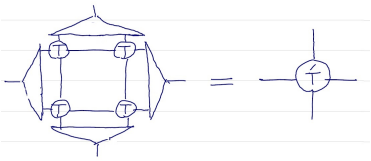
- Therefore, $X \equiv P_A P_B^{\mathsf{T}}$, satisfies (4), $R_A X R_B^{\mathsf{T}} = \hat{U} \hat{\Lambda} \hat{V}^{\mathsf{T}}$, so it yields the optimal value of (3). This means that we can take P_A and P_B as "triangle operators".



Summary of the TNRG procedure (no reflection symmetry)

1. QR-decompose A and B matrices.
$$A = Q_A R_A, \; B = Q_B R_B$$
2. SVD. $R_A R_B^{\mathsf{T}} = U \Lambda V^{\mathsf{T}}$
3. Compute the "triangle operators".
$$P_A \equiv R_B^{\mathsf{T}} \hat{V} \hat{\Lambda}^{-\frac{1}{2}},$$
$$P_B \equiv R_A^{\mathsf{T}} \hat{U} \hat{\Lambda}^{-\frac{1}{2}}$$
4. Do the same for other $d-1$ directions.*
5. Contract the cluster of b^d original operators together with the $2d$ triangle operators and obtain the new element tensor $\acute{T}$.
6. Repeat these till the desired system size has been reached.





* Computationally, it is less expensive to bunch up one direction first before computing the triangular operators in the other directions.