Lecture 5: Tensor-Network Renormalization Group (TNRG)

Naoki KAWASHIMA

ISSP, U. Tokyo

May 13, 2019

In this lecture, we see ...

- The MKRG was manageable, but is rather crude an approximation. Even worse, we do not know when we can expect this approximation to be good or how we can improve 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.

[5-1] 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 both the faults in the real-space RG.

What is a tensor network?

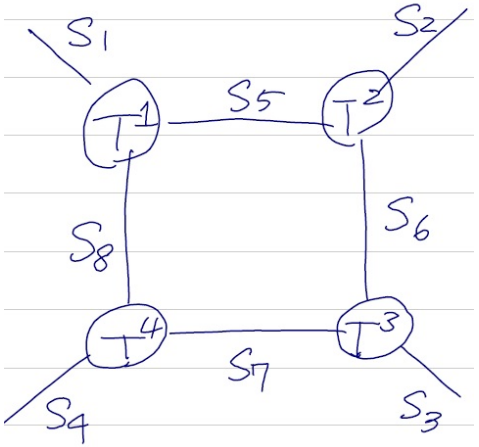
- When an object is expressed as the result of (full or partial) contraction of tensor-product of multiple tensors, we call such an expression a "tensor-network". An expression such as

$$\mathrm{Cont}\left(\prod_k T^k\right) \equiv \sum_{(s_i)_{i\in\Omega}} \prod_k T^k_{s_{i_1^k}, s_{i_2^k}, \cdots, s_{i_{n_k}^k}} \tag{1}$$

  is a tensor-network, where Ω is a subset of all indices, $\{i_\alpha^k\}$, appearing multiple times (typically twice) in the summand.

- Example:

$$T_{s_1,s_2,s_3,s_4} = \sum_{s_6,s_7,s_8,s_9} T^1_{s_1,s_8,s_5}\, T^2_{s_2,s_5,s_6}\, T^3_{s_3,s_6,s_7}\, T^4_{s_4,s_7,s_8}$$

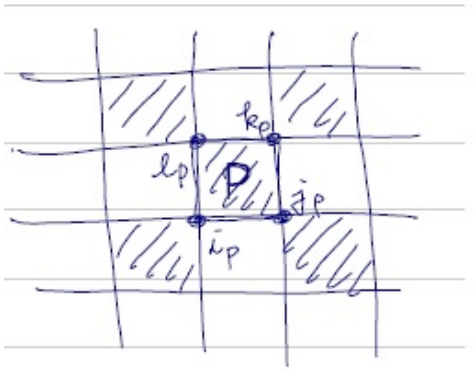


Statistical-mechanical models are tensor networks

The partition function of the Ising model on the square lattice can be expressed as

$$Z = \sum_{S} \prod_{p\,:\,\text{shaded square}} W(S_{i_p}, S_{j_p}, S_{k_p}, S_{l_p})$$

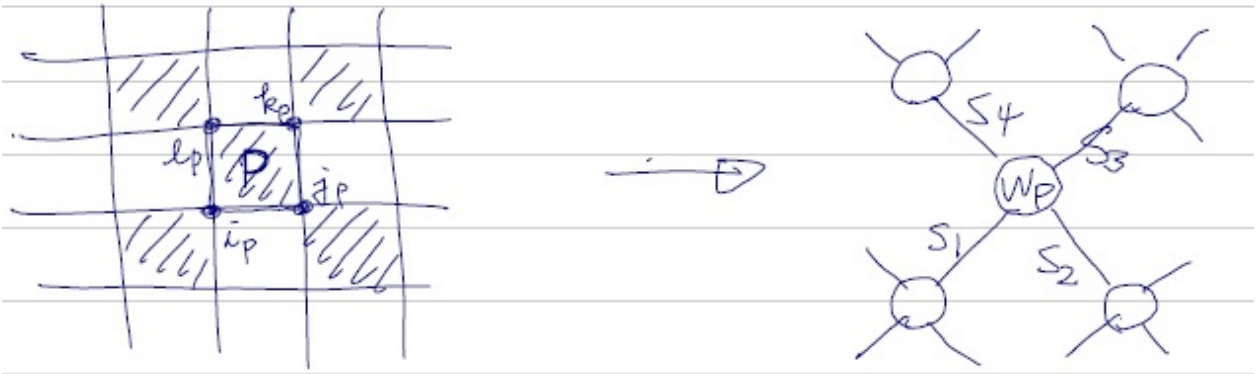
$$W(S_1, S_2, S_3, S_4) \equiv e^{K(S_1S_2 + S_2S_3 + S_3S_4 + S_4S_1)}$$



- We can regard $W(S_1, S_2, S_3, S_4)$ as a degree-4 tensor.
- Then, the above equation is a tensor network representation of the partition function.

Graphical notation

- In TN-related discussions, we use more diagrams than equations because it is often much easier to grasp the idea.
-



- For tensors, we often use bulkier symbols than just dots such as circles, triangles, squares, etc, while we use simple lines for indices. (This is more natural from the information-scientific point of view because tensors are the carriers of most of the information.)

Wave function can be represented as TN (1)

- Consider a quantum many-body system defined on a lattice.
- A local quantum degree of freedom, say S_i, is defined on each site.
- Accordingly, we have a local Hilbert space $H_i \equiv \{|S_i\rangle_i\}$ associated with each site, e.g., H_i is 2-dimensional for $S = 1/2$ spin models.
- 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

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

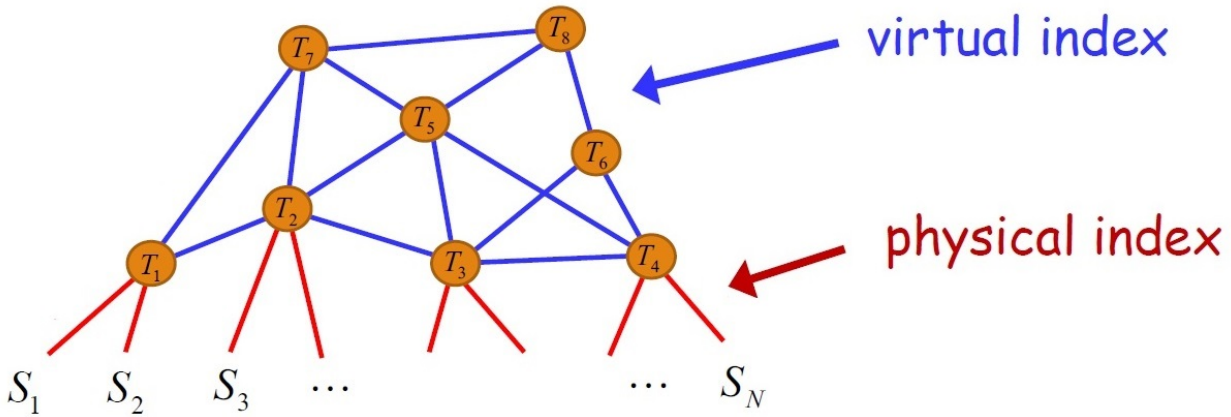
  where $|S_1, S_2, \cdots, S_N\rangle \equiv |S_1\rangle_1 \otimes |S_2\rangle_2 \otimes \cdots \otimes |S_N\rangle_N$.
- Now, $C_{S_1,S_2,\cdots,S_N}$ can be viewed as a degree-N tensor. It may be approximated by some tensor network, i.e,

$$C_{\mathbf{S}} \approx \mathrm{Cont}\left(\prod_k T^k\right)$$

Wave function can be represented as TN (2)

$$C_{\mathbf{S}} \approx \mathrm{Cont}\left(\prod_k T^k\right)$$

$=$



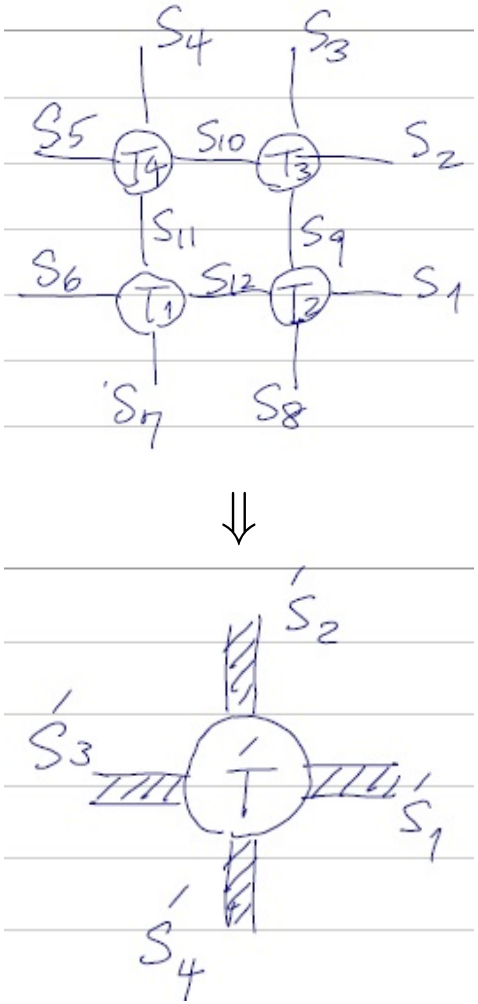
Note that $C_{\mathbf{S}}$ has d^N parameters ($d = 2$ for $S = 1/2$ spin systems), whereas the tensor network can be specified by only $O(N)$ number of parameters. By the tensor network representation, we may be able to reduce the computation for large N down to a manageable level.

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:

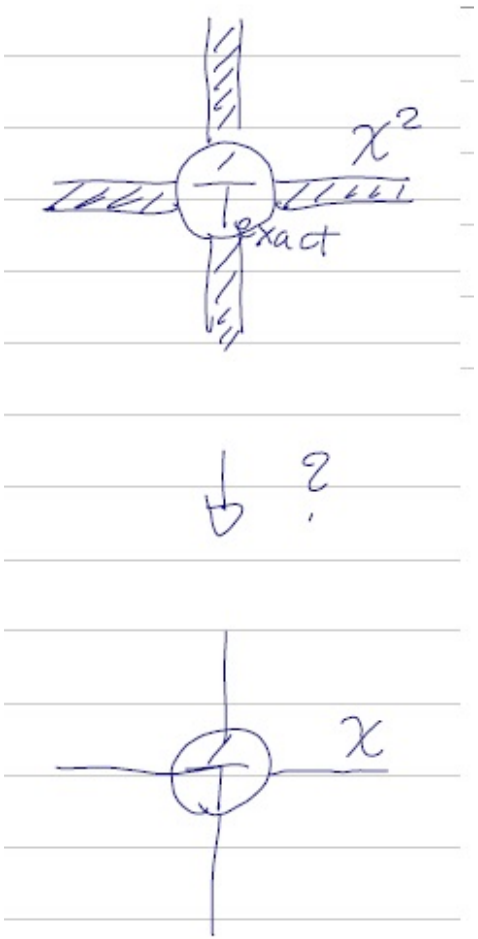
$$\acute{T}_{\acute{s}_1,\acute{s}_2,\acute{s}_3,\acute{s}_4} \equiv \sum_{s_9,s_{10},s_{11},s_{12}} T^1_{s_1,s_9,s_{12},s_8} \times T^2_{s_2,s_3,s_{10},s_9} T^3_{s_{10},s_4,s_5,s_{11}} T^4_{s_{12},s_{11},s_6,s_7}$$

where $\acute{s}_1 \equiv (s_1, s_2)$, $\acute{s}_2 \equiv (s_3, s_4), \cdots$



What's wrong with trivial TNRG?

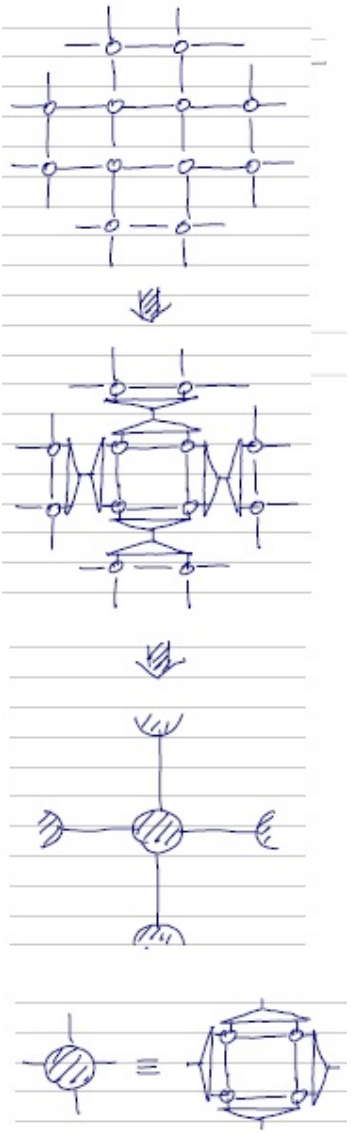
- Using $\acute{T}$, we can exactly express the original partition function with lattice constant twice larger than the original, which is good.
- However, the dimension of each index of the new tensor is χ^2 where χ is the index dimension of the original tensor.
- To be more specific, to handle an $L \times L$ system we end up with a big tensor with χ^L-dimensional indices. (We cannot go to so large L.)
- To make the whole procedure practically useful for larger systems, we need to make the index dimension back to χ in each iteration.



Data compression is necessary!

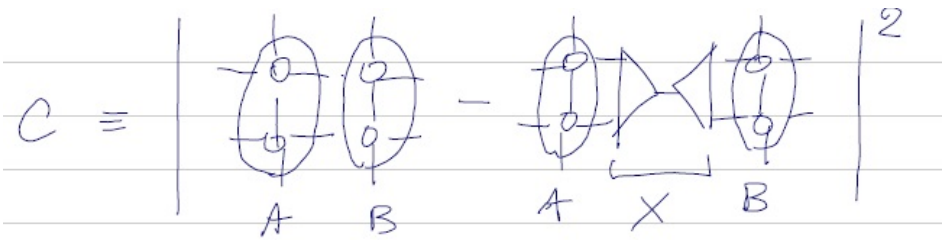
Rank-reducer

- What we need is a 'rank-reducer'.
- A rank-reducer is a tensor whose rank (when viwed as a matrix) is χ in stead of χ^2, and whose insertion keep things unchanged.
- If such a thing exists, we can define triangle operators as illustrated in the figure by SVD.
- Then, by cutting the network at the reduced indices, we can define the renormalized tensor with index dimension χ, as we disired.
- Now, we must ask whether such a magical rank-reducer exists or not, and if it does, how we can compute it.



Low-rank approximation (LRA)

- How can we optimize the rank-reducer X for the given rank χ?
- For the cost function, we 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

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

Solution to LRA problem (1)

- We want the rank-l matrix X that minimizes

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

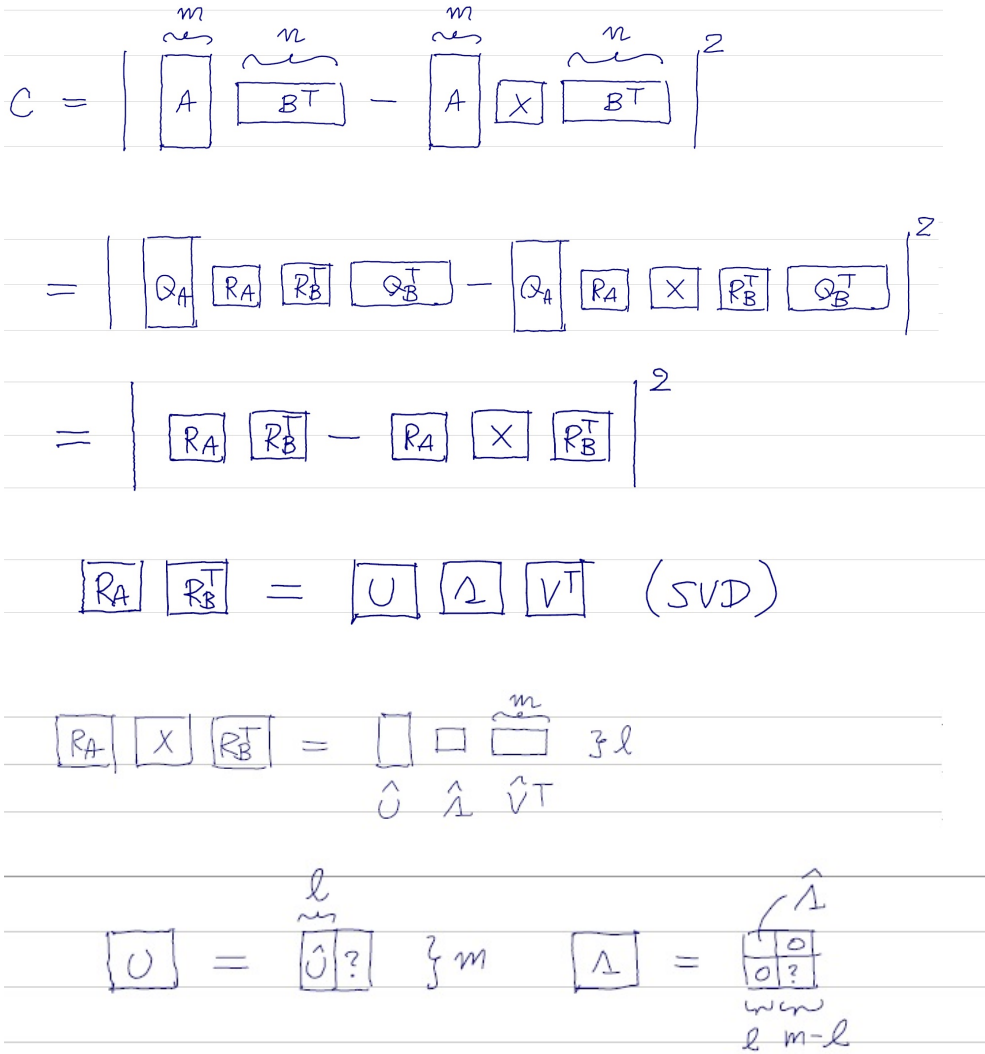
- Consider the QR-decomposition,

$$A = Q_A R_A, \quad B = Q_B R_B.$$

- Then, $C \equiv |R_A R_B^{\mathsf{T}} - R_A X R_B^{\mathsf{T}}|^2$
- Consider SVD: $R_A R_B^{\mathsf{T}} = U \Lambda V^{\mathsf{T}}$.
- If X satisfies

$$R_A X R_B^{\mathsf{T}} = \hat{U} \hat{\Lambda} \hat{V}^{\mathsf{T}}, \qquad (3)$$

it is optimal. $(*)$ (Here, $\hat{U}$, $\hat{\Lambda}$, and $\hat{V}$ are truncated matrices at the l-th row and/or column.)



Solution to LRA problem (2)

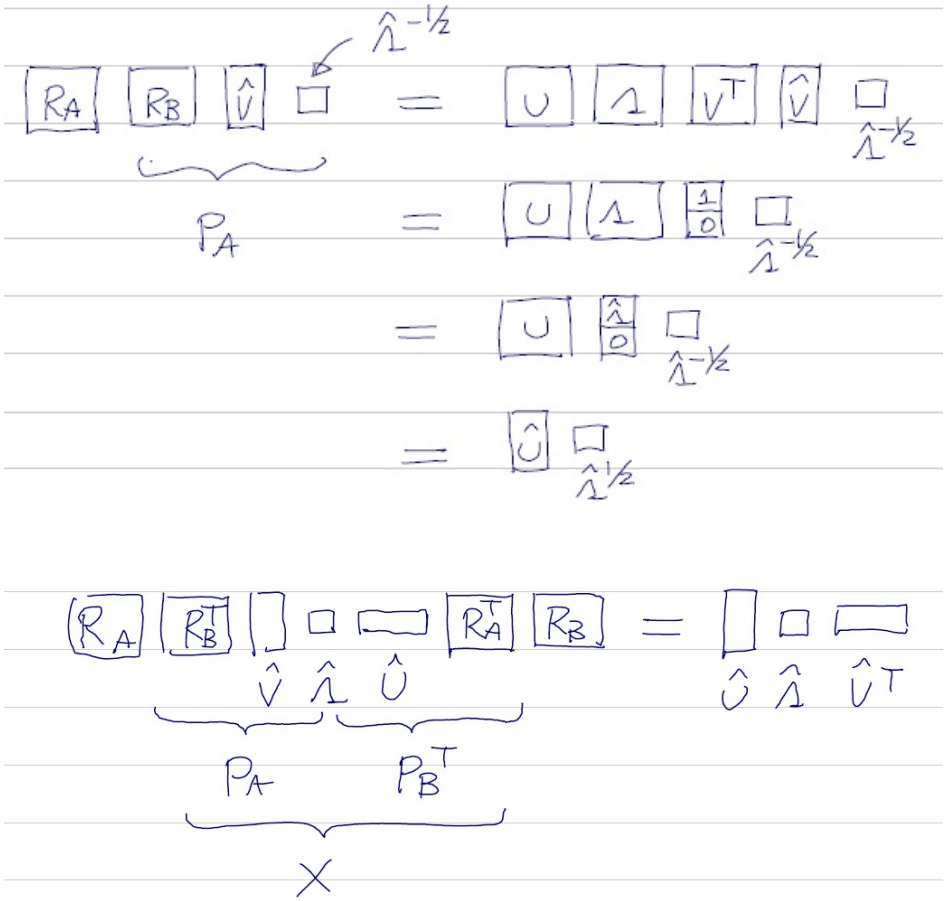
- Now, let us define "triangule operators," P_A and P_B, by

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

- 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 Eq.(3), $R_A X R_B^{\mathsf{T}} = \hat{U} \hat{\Lambda} \hat{V}^{\mathsf{T}}$, and therefore is the optimal rank-reducer.



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,

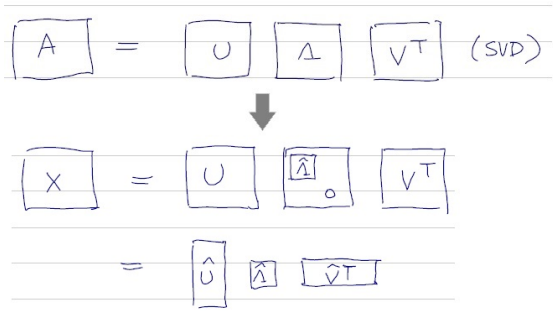
$$E^2 \geq \lambda_{l+1}^2 + \lambda_{l+2}^2 + \cdots + \lambda_m^2$$

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.

$\Lambda \equiv$

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

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



Supplement: LRA used in the derivation

Theorem 2 (LRA)

Consider an $m \times m$ matrix Y expressed as $Y = R_A R_B^T$ with R_A and R_B, and consider its SVD, $Y = U\Lambda V^{\mathsf{T}}$. Then, Y's optimal LRA of the form $R_A X R_B^{\mathsf{T}}$ with rank l ($l < m$) matrix X is obtained when $R_A X R_B^{\mathsf{T}} = \hat{U}\hat{\Lambda}\hat{V}^{\mathsf{T}}$

.

Proof: When the condition of the theorem is satisfied,

$$|Y - R_A X R_B^{\mathsf{T}}|^2 = |U\Lambda V^{\mathsf{T}} - \hat{U}\hat{\Lambda}\hat{V}^{\mathsf{T}}|^2$$

$$= |U\Lambda V^{\mathsf{T}} - U\tilde{\Lambda}V^{\mathsf{T}}|^2 = |\Lambda - \tilde{\Lambda}|^2 = \sum_{k=l+1}^{m} \lambda_k^2,$$

where $\tilde{\Lambda}$ is Λ with singular values λ_k ($k > l$) replaced by 0. Therefore, $R_A X R_B^{\mathsf{T}}$ saturates the inequality of the EYM theorem.

Summary of the TNRG procedure

1. QR-decomposition of A and B matrices.

   $$A = Q_A R_A, \;\; B = Q_B R_B$$

2. SVD. $\quad R_A R_B^{\mathsf{T}} = U \Lambda V^{\mathsf{T}}$

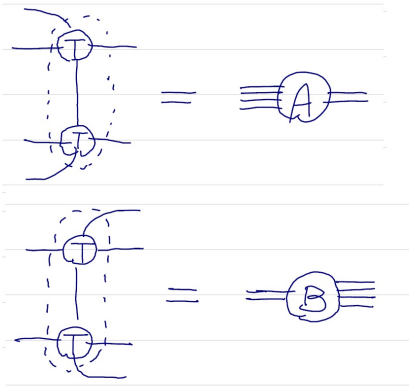
3. Compute the "triangle operators".

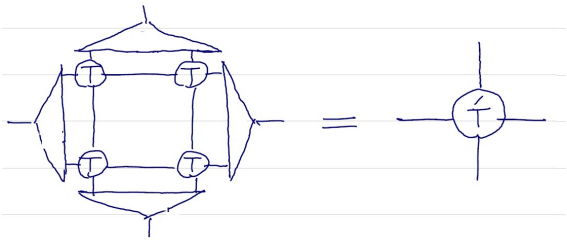
   $$P_A \equiv R_B^{\mathsf{T}} \hat{V} \hat{\Lambda}^{-\frac{1}{2}},$$
   $$P_B \equiv R_A^{\mathsf{T}} \hat{U} \hat{\Lambda}^{-\frac{1}{2}}$$

4. Do the same for other directions.

5. Using the triangular operators, contract four original tensors to obtain the new element tensor $\acute{T}$.

6. Repeat these till the desired system size has been reached.



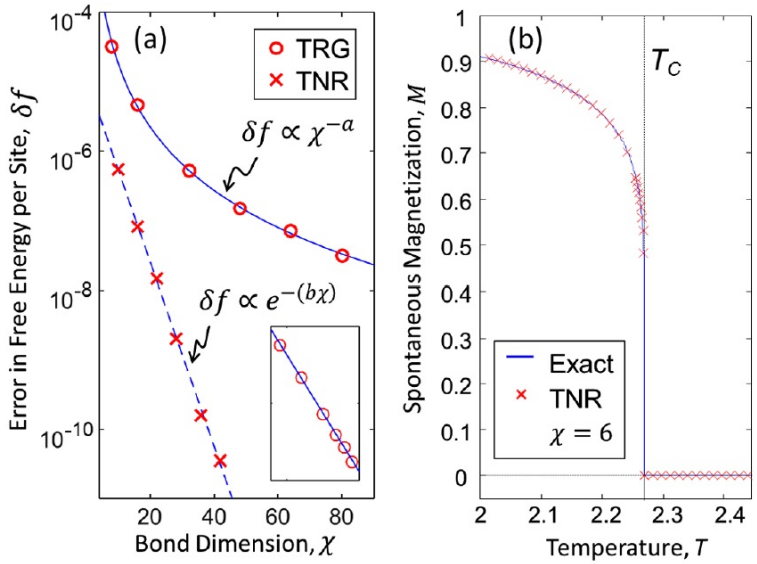


TNRG provides accurate estimates

- 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",

  $$T^{(0)} \equiv T, \quad T^{(n)} \equiv 0 \ (n > 1)$$
  $$T^{(1)}_{S_1 S_2 S_3 S_4} \equiv T_{S_1 S_2 S_3 S_4} \times m(S_1, S_2, S_3, S_4)$$

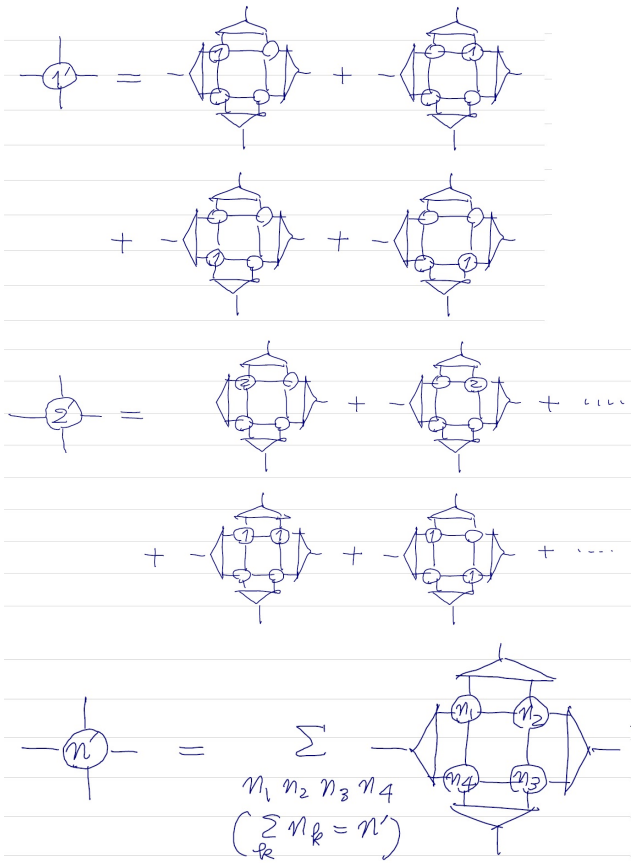
  where $m = (S_1 + S_2 + S_3 + S_4)/2$.

- Define "renormalized impurity tensors":

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

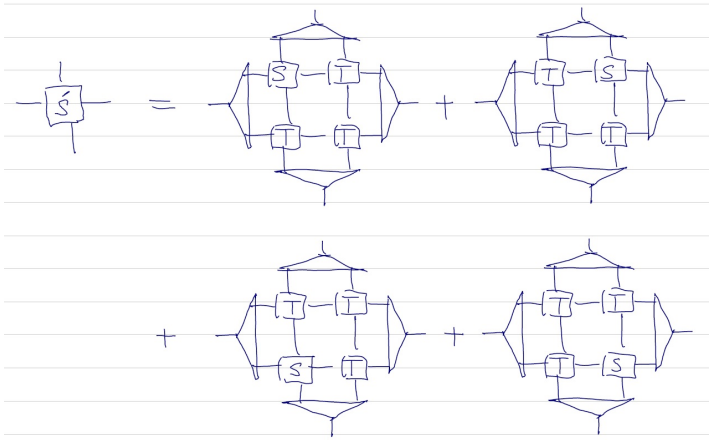
- At the end of all iterations,

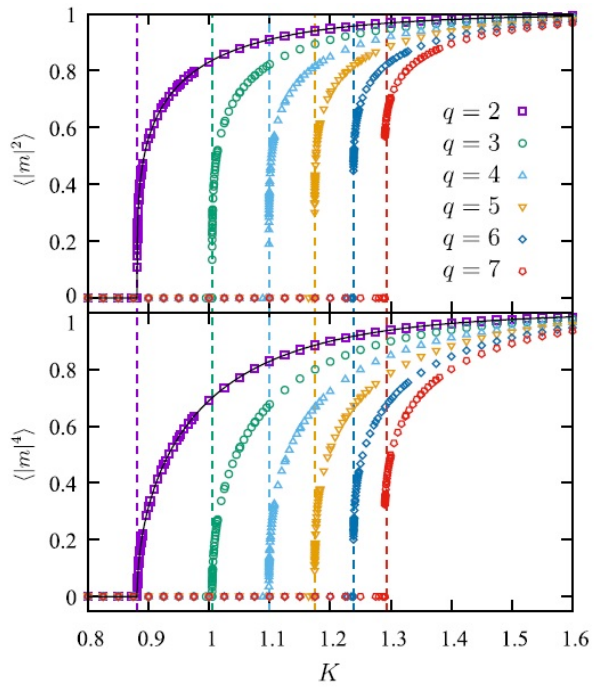
  $$\langle M^n \rangle = \sum_{S_1 S_2} T^{(n)}_{S_1 S_2 S_1 S_2} / \sum_{S_1 S_2} T^{(0)}_{S_1 S_2 S_1 S_2}$$



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') $\chi = 48$.



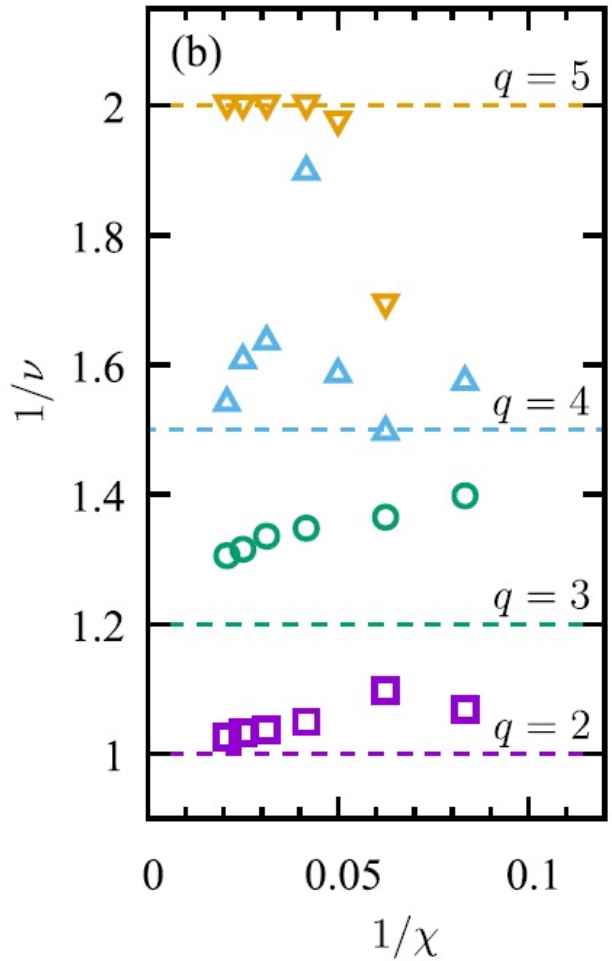


Application of TNRG to q-state Potts model (2)

- 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).]

Summary

- Tensor-network RG (TNRG) is a scheme that realizes **"data compression"** at every length scale.
- 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.
- While we have seen just one way of implementing the idea, there are many proposals for realizing TNRG. (MERA, TRG, TNR, loop-TNR, etc)