PROCEEDINGS OF SCIENCE

Initial tensor construction for the tensor renormalization group

Katsumasa Nakayama[a,*] **and Manuel Schneider**[b]

[a] *RIKEN Center for Computational Science, Kobe, 650-0047, Japan*

[b] *Institute of Physics, National Yang Ming Chiao Tung University, 1001 University Road, Hsinchu 30010, Taiwan*

E-mail: katsumasa.nakayama@riken.jp

We propose a method to construct the initial tensor representation of partition functions and observables for the tensor renormalization group (TRG). The TRG is a numerical calculation technique that utilizes a tensor network representations of physical quantities to investigate physical properties without encountering the sign problem. To apply the TRG, it is essential to construct a locally connected tensor network suitable for recursive coarse-graining. We present a systematic approach for translating a general tensor representation of the partition function to this form. Furthermore, we show the dependence of TRG algorithms on the choice of the initial tensor network representation and propose an improvement of TRG algorithms in this respect.

The 41st International Symposium on Lattice Field Theory (LATTICE2024)
28 July - 3 August 2024
Liverpool, UK

*Speaker

PoS(LATTICE2024)043

1. Introduction

Tensor network representations are widely used to calculate physical quantities [1, 2]. If observables are expressed in the form of a tensor network, they can be calculated with tensor renormalization group (TRG) methods [3] that contract all indices of the tensor network. Since these TRG methods do not sample from a distribution function like Monte Carlo methods, they can be applied to systems that would otherwise suffer from a sign problem. Examples are systems at finite densities or with a topological charge [4, 5]. The TRG has been successfully applied to quantum field theories with gauge fields [6–9]. Therefore, TRG methods provide a promising toolbox for studying high-energy physics where Monte Carlo cannot be applied.

The prerequisite for applying TRG methods is that the partition function or other observables are written in the form of a locally connected tensor network. In this proceeding, we introduce a generic method for finding such a representation. We start from the partition function that is expressed by the product of Boltzmann factors with a summation over physical degrees of freedom. In the two-dimensional Ising model, for example, the spin degrees of freedom are summed and the partition function can be expressed as

$$Z = \sum_{\sigma=\pm 1} \prod_{x,y=1}^{N} e^{\frac{\beta g}{2}\sigma_{x,y}(\sigma_{x+1,y}+\sigma_{x,y+1})} = \sum_{\sigma=\pm 1} \prod_{x,y=1}^{N} K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}\sigma_{x,y+1}}. \tag{1}$$

Here, β is inverse temperature, g is the coupling constant, and σ are the spin variables at the lattice points $\{x, y\}$. The partition function is expressed as a product of the Boltzmann factors $K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}\sigma_{x,y+1}}$. This is used as a starting point to construct a locally connected tensor network representation.

Note that the Boltzmann factor representation is not suitable for the application of TRG methods since it is not a tensor network. Although $K^{(\mathrm{Ising})}$ are tensors of rank three, their indices are not contracted pairwise. A tensor network representation implies that an index that is summed over appears exactly twice in the product. We can therefore represent it graphically as a line that connects two tensors, or it could connect a tensor with itself. In the previous partition function, each index is shared by three tensors: $\sigma_{x,y}$ is an index of the tensors $K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}\sigma_{x,y+1}}$, $K^{(\mathrm{Ising})}_{\sigma_{x-1,y}\sigma_{x,y}\sigma_{x-1,y+1}}$, and $K^{(\mathrm{Ising})}_{\sigma_{x,y-1}\sigma_{x+1,y-1}\sigma_{x,y}}$ in the product.

Common methods for constructing a tensor network representation use expansions such as the Taylor expansion, character expansion, or expansion by orthogonal functions [10]. For the example of the Ising model, a tensor network representation derived from a Taylor expansion [11, 12] is

$$K^{(\mathrm{exp})}_{l_{x,y},l_{x+1,y},m_{x,y},m_{x,y+1}} = \sum_{\alpha} W_{\alpha,l_{x,y}} W_{\alpha,l_{x+1,y}} W_{\alpha,m_{x,y}} W_{\alpha,m_{x,y+1}}, \tag{2}$$

where the matrix W is defined as

$$W = \begin{pmatrix} \sqrt{\cosh(\beta g/2)} & \sqrt{\sinh(\beta g/2)} \\ \sqrt{\cosh(\beta g/2)} & -\sqrt{\sinh(\beta g/2)} \end{pmatrix}. \tag{3}$$

These constructions are model-specific and make use of certain properties of the Boltzmann weights. For example, the property $\sigma^2 = 1$ is used for the Ising model to find a finite-size tensor $K^{(\mathrm{exp})}$. Consequently, these previous approaches cannot be generalized to arbitrary physical models.

In this proceeding, we present an initial tensor construction method for general Boltzmann factor representations, without any expansions or decompositions. The applicability of the method to gauge theories was shown for the $\mathbb{Z}_2$ model [8], where the free energy and specific heat were calculated in good agreement with previous TRG calculations and Monte Carlo simulations [6].

2. Initial tensor construction for the Ising model

We explain our initial tensor construction for the Ising model with periodic boundary conditions in two-dimensional space-time in the following. The starting point is the tensor representation of the partition function in terms of the Boltzmann factors $K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}\sigma_{x,y+1}}$ from eq. (1). We introduce a new index a and use the identity

$$K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}\sigma_{x,y+1}} = \sum_{a=\pm 1} K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y+1}} \delta_{\sigma_{x,y+1}a_{x,y+1}}. \tag{4}$$

Note that this transformation does not require specific details of the Boltzmann factors K or any numerical calculation. The partition function becomes

$$Z = \sum_{a,\sigma=\pm 1} \prod_{x,y=1}^{N} K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y+1}} \delta_{\sigma_{x,y+1}a_{x,y+1}} \tag{5}$$

$$= \sum_{a,\sigma=\pm 1} \prod_{x,y=1}^{N} K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y+1}} \delta_{\sigma_{x,y}a_{x,y}}. \tag{6}$$

In the second line we shifted the indices of the delta functions by one site and made use of periodic boundary conditions. With the definition of new tensors

$$K^{(\mathrm{delta})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y}a_{x,y+1}} \equiv K^{(\mathrm{Ising})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y+1}} \times \delta_{\sigma_{x,y}a_{x,y}} \tag{7}$$

the partition function can be written as

$$Z = \sum_{a,\sigma=\pm 1} \prod_{x,y=1}^{N} K^{(\mathrm{delta})}_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y}a_{x,y+1}}. \tag{8}$$

This is a locally connected tensor network, since each index appears on exactly two nearest neighboring tensors $K^{(\mathrm{delta})}$. It is therefore a suitable starting point for the TRG method. Figure 1 shows a schematic picture of the index shift using a delta matrix for the Ising model.

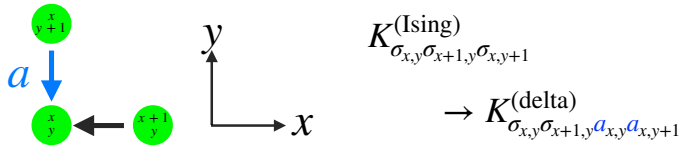

Figure 1: Schematic of the tensor network construction for the Ising model in the two dimensions. Green nodes represent the spin indices of a Boltzmann factor $K^{(\mathrm{Ising})}$ in eq. (1). The blue vertical arrow represents the introduction of a new index $a_{x,y+1}$ as in eq. (4), and a translation $a_{x,y+1} \to a_{x,y}$ as from eq. (5) to eq. (6). The black vertical arrow marks the nearest neighboring two unchanged spin variables in the resulting tensor $K^{(\mathrm{delta})}$.

3. Initial tensor construction for general models

For general models, the same index will not only be shared by three consecutive Boltzmann factors as in the two-dimensional Ising model. The initial tensor construction method can, however, be generalized. The previous construction consisted of two steps:

1. Insert a delta function, which creates a new index
2. Shift the delta function to a different site and combine it with the tensor at that site to create a new tensor that defines the partition function.

Although the partition function remains unchanged by these steps, the new tensor no longer depends on the index that was separated by the delta function. The dependence was replaced by creating new variables. However, these new indices each appear on two neighboring tensors only, as required for a tensor contraction.

We formulate the concrete steps to construct a locally connected tensor network from any given Boltzmann weight as a pseudo-algorithm:

Algorithm 1 Create a locally connected tensor network

Input: Boltzmann weights K

Output: Locally connected tensors for TRG

1: Connect all sites that K depends on by arrows; these should form a tree with arrows pointing from the roots to the origin
2: **while** arrows left in tree **do**
3: Pick a root with an arrow pointing from $\hat{r}-\hat{\mu}$ to $\hat{r}$ in $\hat{\mu}$-direction
4: Insert a delta function: $K_{\ldots,\sigma_{\hat{r}-\hat{\mu}}} = \sum_{a_{\hat{\nu}}} K_{\ldots,a_{\hat{\nu}}} \delta_{\sigma_{\hat{r}-\hat{\mu}},a_{\hat{\nu}}}$
5: Shift the delta function to a neighboring tensor: $K'_{\ldots,\sigma_{\hat{r}},a_{\hat{\nu}},a_{\hat{\nu}+\hat{\mu}}} \equiv K_{\ldots,a_{\hat{\nu}}} \delta_{\sigma_{\hat{r}},a_{\hat{\nu}+\hat{\mu}}}$
6: Replace K by the new K' and remove the arrow from the tree
7: **end while**

The newly introduced indices $a_{\hat{\nu}}$ have a site-index $\hat{\nu}$, which should be chosen such that the new tensor $K'^{(\hat{x})}_{\ldots,a_{\hat{\nu}},a_{\hat{\nu}+\hat{\mu}}}$ based at site $\hat{x}$ depends only on $a_{\hat{x}}$ and $a_{\hat{x}+|\hat{\nu}|}$.[1] For two dimensions, this explicitly means:

→ Right arrow pointing from $(x'-1,y')$ to (x',y') in direction $\hat{\mu}=\hat{x}$ new index: $a_{\hat{\nu}} = a_{x,y}$
decomposition: $K^{(x,y)}_{\ldots,\sigma_{x'-1,y'}} = \sum_{a_{x,y}} K_{\ldots,a_{x,y}} \delta_{\sigma_{x'-1,y'},a_{x,y}}$
new tensor: $K'^{(x,y)}_{\ldots,\sigma_{x',y'},a_{x,y},a_{x+1,y}} \equiv K^{(x,y)}_{\ldots,a_{x,y}} \delta_{\sigma_{x',y'},a_{x+1,y}}$

← Left arrow pointing from $(x'+1,y')$ to (x',y') in direction $\hat{\mu}=-\hat{x}$; new index: $a_{\hat{\nu}} = a_{x+1,y}$
decomposition: $K^{(x,y)}_{\ldots,\sigma_{x'+1,y'}} = \sum_{a_{x+1,y}} K_{\ldots,a_{x+1,y}} \delta_{\sigma_{x'+1,y'},a_{x+1,y}}$
new tensor: $K'^{(x,y)}_{\ldots,\sigma_{x',y'},a_{x+1,y},a_{x,y}} \equiv K^{(x,y)}_{\ldots,a_{x+1,y}} \delta_{\sigma_{x',y'},a_{x,y}}$

↑ Up arrow pointing from $(x',y'-1)$ to (x',y') in direction $\hat{\mu}=\hat{y}$; new index: $a_{\hat{\nu}} = a_{x,y}$
new tensor: $K'^{(x,y)}_{\ldots,\sigma_{x',y'},a_{x,y},a_{x,y+1}} \equiv K^{(x,y)}_{\ldots,a_{x,y}} \delta_{\sigma_{x',y'},a_{x,y+1}}$

[1] In this convention, the final tensor will depend on indices at sites (x,y) and $(x,y)+|\hat{\mu}|$ for all directions $\hat{\mu}$. Other conventions such as a dependence on (x,y) and $(x,y)-|\hat{\mu}|$ would also be possible.

↓ Down arrow pointing from $(x', y'+1)$ to (x', y') in direction $\hat{\mu} = -\hat{y}$; new index: $a_{\hat{y}} = a_{x,y+1}$
new tensor: $K'^{(x,y)}_{\ldots,\sigma_{x',y'},a_{x,y+1},a_{x,y}} \equiv K^{(x,y)}_{\ldots,a_{x,y+1}} \delta_{\sigma_{x',y'},a_{x,y}}$

We note that the steps in lines 3 to 6 can be skipped for one final arrow pointing to the origin at $\hat{x} = (x, y)$. In diagrams, we draw the arrow pointing from $(x+1, y)$ to (x, y) in black, indicating that we omit the transformation and are left with a dependence on the original indices $\sigma_{x,y}$ and $\sigma_{x+1,y}$.

In line 4, we label the new indices $a_{\hat{y}}$. In subsequent iterations, new names should be chosen, for example $a, b, c, \ldots$.

To illustrate the procedure in more general cases, we discuss two additional examples. In the first case, we start from a Boltzmann factor representation which contains non-local interactions:

$$Z = \sum_{\sigma=\pm1} \prod_{x,y=1}^{N} K_{\sigma_{x,y}\sigma_{x+1,y+1}\sigma_{x+2,y+1}\sigma_{x+1,y+2}}. \tag{9}$$

The tensor $K_{\sigma_{x,y}\sigma_{x+1,y+1}\sigma_{x+2,y+1}\sigma_{x+1,y+2}}$ has four different indices and can represent a variety of interaction terms [8]. The diagram in fig. 2 shows the directed tree that connects all sites involved in the interaction to the origin of the Boltzmann weight at (x, y).

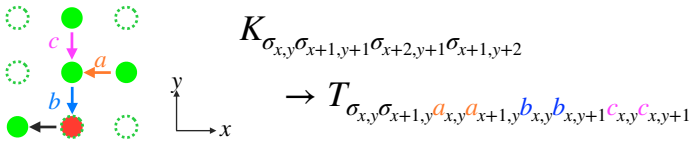

Figure 2: Schematic picture of the tensor network construction. The initial Boltzmann weight K is transformed into tensors T, which are the building blocks of a locally connected tensor network for TRG coarse-graining. Filled green nodes represent the original degrees of freedom that K depends on. Arrows are introduced to create a directed tree connecting all green nodes to the origin at (x, y). Red nodes are additional points that need to be included in the tree. Each arrow introduces a new variable in T according to algorithm 1. The final black arrow pointing to the origin leaves the spin variables unchanged.

In order to construct the tensor network, we replace the arrow $\sigma_{x+2,y+1} \to \sigma_{x+1,y+1}$ and introduce the new index a. This is followed by a replacement of the arrow $\sigma_{x+1,y+1} \to \sigma_{x+1,y}$, which introduces the index c. In a last step, the arrow $\sigma_{x+1,y+2} \to \sigma_{x+1,y+1}$ creates index b.

The final tensor T, expressed in terms of the original Boltzmann weights K becomes

$$T_{\sigma_{x,y}\sigma_{x+1,y}a_{x,y}a_{x+1,y}b_{x,y}b_{x,y+1}c_{x,y}c_{x,y+1}} \equiv K_{\sigma_{x,y}b_{x,y+1}a_{x+1,y}c_{x,y+1}} \delta_{b_{x,y+1}a_{x,y}} \delta_{b_{x,y+1}c_{x,y}} \delta_{\sigma_{x+1,y}b_{x,y}}. \tag{10}$$

Although the Boltzmann factor K does not have an index $\sigma_{x+1,y}$, we have to add the point $(x+1, y)$ in the sequential index shifts to connect the other points to the origin. This is indicated by the red node in fig. 2 which is part of the tree structure.

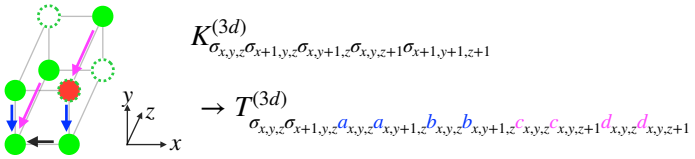

Figure 3: Schematic picture of the tensor network construction in three dimensions.

As a second example, we consider a three-dimensional system as depicted in fig. 3 with a Boltzmann factor

$$Z^{(3\mathrm{d})} = \sum_{\sigma=\pm 1} \prod_{x,y,z=1}^{N} K^{(3\mathrm{d})}_{\sigma_{x,y,z}\sigma_{x+1,y,z}\sigma_{x,y+1,z}\sigma_{x,y,z+1}\sigma_{x+1,y+1,z+1}}. \tag{11}$$

The structure can incorporate nearest-neighbor interactions as well as an interaction on the diagonal of a cubic unit cell. The index shifts are indicated in fig. 3. The final tensor T has a larger size of d^{10} if d is the dimension of an initial degree of freedom σ. The resource scaling is discussed in more detail in [8].

The initial tensor construction does not assume any specific properties of the Boltzmann weights. Because of this, algorithm 1 can be used in very general cases to construct the locally connected tensor network representation of a partition function.

4. The Steiner tree problem

For the initial tensor network construction according to algorithm 1, we have to connect all the sites on which the Boltzmann weight depends by a tree. This is the initial line 1 of the algorithm and is depicted in figs. 1 to 3. Every arrow in the tree increases the bond dimension of the final tensor. Therefore, the optimal choice minimizes the number of arrows in the tree. This is known as the rectilinear Steiner tree problem, which is a generalized minimum spanning tree problem [13]. The latter is the problem of finding line segments of minimum length which connect every given point. The Steiner tree problem extends this by allowing one to add new points which also have to be connected by line segments. For square and cubic lattices we have the additional constraints that nodes must be on lattice points, and lines are only allowed between nearest neighbors.

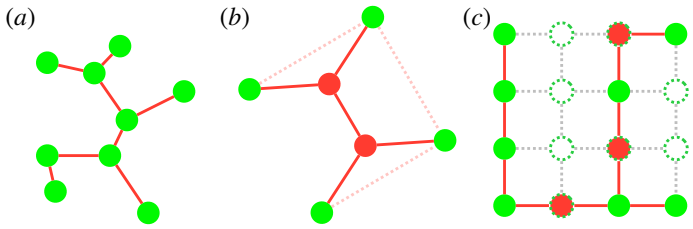

Figure 4: Examples of (a) the minimum spanning tree problem, (b) the Steiner tree problem, and (c) the rectilinear Steiner tree problem. Green dots are the original nodes, and red dots are Steiner points that were added. In (b), the dotted lines are the solution of the minimum spanning tree problem, which has a longer total line length than the Steiner tree solution. In (c), nodes and line segments are only allowed on a rectangular lattice of equal spacing (dotted nodes and lines).

Figure 4 shows the minimum spanning tree and Steiner tree problems. The Steiner tree problem is NP-hard, and the rectilinear Steiner tree problem is NP-complete. Thus, the problem becomes very challenging if a large number of non-local interactions in the Boltzmann weights is given. However, relevant physical systems only include a small number of sites on which the Boltzmann weight depends. In these cases, the rectilinear Steiner problem can still be solved exactly by hand.

5. Initial tensor dependence of tensor renormalization methods

The partition function can be represented in many ways by different initial tensors. In this section, we use the example of the two-dimensional Ising model, where the tensors obtained from our method in eq. (7) differ from the initial tensor in eq. (2). We discuss how the accuracy of TRG algorithms with truncations depend on the form of the initial tensors. We suggest a way to remove this dependence by improvements using the boundary TRG method [14, 15].

We calculate the free energy $F = -\frac{1}{\beta V}\ln Z$ for a volume $V = 2^{20}$ at the critical temperature $\beta_c = \ln(1+\sqrt{2})$ and compare to the exact solution [16]. We use the higher-order TRG (HOTRG) method [12]. Figure 5 shows the initial tensor dependence. The Taylor expansion method with initial tensors $K^{(\mathrm{exp})}$ leads to better precision compared to our method with $K^{(\mathrm{delta})}$.

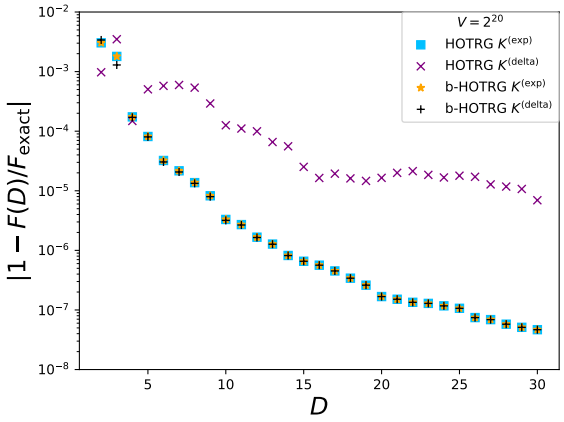

Figure 5: Error of the free energy of the Ising model at the critical temperature from HOTRG and boundary-HOTRG. Different initial tensors are used to represent the partition function, see eqs. (2) and (7).

In order to remove the dependence on the initial tensors, we apply the ideas of the boundary TRG method [14] to HOTRG. All TRG methods introduce a truncation, which minimizes a cost function. The cost functions of HOTRG and boundary HOTRG are shown in fig. 6. While HOTRG uses only the left or right isometries of a truncated singular value decomposition (SVD), the boundary HOTRG method includes both of them in so called squeezers P which are used for the truncation [15]. This also corresponds to a larger cell that is taken into account in the cost function.

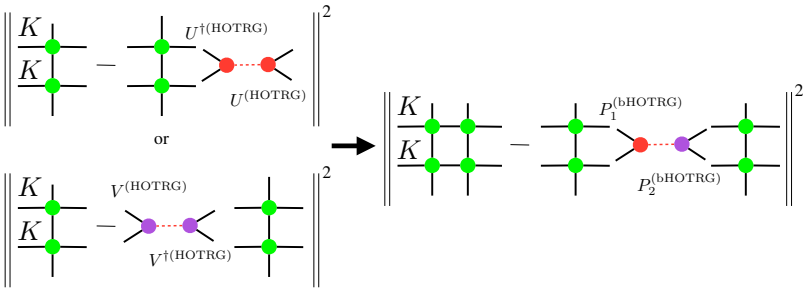

Figure 6: The minimized cost function in the HOTRG (left) and boundary HOTRG (right).

Specifically, the isometry $U^{(\mathrm{HOTRG})}$ is obtained by a truncated SVD of KK with singular values $(\lambda^{(U)})$. It connects to the right indices of the product KK and can be used to truncate

the indices and create coarse-grained tensors. Similarly, the isometry $V^{(\mathrm{HOTRG})}$ truncates the left indices, with corresponding singular values $(\lambda^{(V)})$. HOTRG uses either U or V for the coarse-graining and thus introduces an asymmetry in the truncation step. This is the reason why HOTRG depends strongly on the symmetry properties of the initial tensors [8]. In the boundary HOTRG method, we instead take into account both $U^{(\mathrm{HOTRG})}$ and $V^{(\mathrm{HOTRG})}$. For this, we consider the additional SVD $\lambda^{(U)}U^{(\mathrm{HOTRG})}V^{(\mathrm{HOTRG})}\lambda^{(V)} = U\Lambda V$. The squeezers $P_1^{(\mathrm{bHOTRG})}$ and $P_2^{(\mathrm{bHOTRG})}$ that are used in the truncation are then defined as $P_1^{(\mathrm{bHOTRG})} \equiv V^{(\mathrm{HOTRG})}\lambda^{(V)}V^\dagger/\sqrt{\Lambda}$, $P_2^{(\mathrm{bHOTRG})} \equiv (1/\sqrt{\Lambda})U^\dagger\lambda^{(U)}U^{(\mathrm{HOTRG})}$. Further details can be found in [8].

When the boundary HOTRG method is applied, the initial tensors $K^{(\mathrm{delta})}$ and $K^{(\mathrm{exp})}$ lead to the same accuracy of the free energy, as shown in fig. 5. The boundary HOTRG method can reproduce the previous result of HOTRG with $K^{(\mathrm{exp})}$. Our initial tensor construction thus leads to similarly good results as previous, problem-specific methods. In addition, the boundary TRG improvement is not only applicable to HOTRG but can be used in any TRG method [8, 15, 17, 18]. Replacing isometries by squeezers removes the dependence on the form of the initial tensors.

6. Conclusions

We have shown a general method to find the tensor network representation of a partition function. These initial tensors can be used in tensor renormalization group algorithms. Even though we assumed translational-invariant systems, the method is applicable to inhomogeneous systems. It can, thus, be used not only for the partition function but for general observables, which can include impurity tensors.

The initial tensor construction starts from the Boltzmann weights and replaces original indices by newly introduced ones, which are shifted in the direction of a given path. This path is given as a solution of the Steiner tree problem. Although the latter problem is difficult to solve in general, we can typically find the solution easily for physically relevant systems with a limited number of non-local interactions. The solution of the Steiner tree problem directly relates to the size of the initial tensors, which is important for the accuracy of the coarse-graining. Thus, we can estimate the applicability and numerical demands of TRG methods in advance.

We further discussed the dependence of TRG methods on the form of the initial tensors. Algorithms like HOTRG lead to inaccurate results for non-symmetric initial tensors. However, this dependence can be removed by introducing squeezers in the coarse-graining step, similarly to the boundary TRG method. In this way, common TRG methods can be improved, and the initial tensor construction presented in this work leads to accuracies similar to those of previous constructions.

We assumed periodic boundary conditions in order to move the delta functions from one tensor to its nearest neighbor. The method can, however, also be used for other boundary conditions. In this case, the boundary tensors differ from the tensors in the bulk.

Acknowledgments

This work is based on the publication [8] and was supported by JSPS KAKENHI Grant No. 24K17059, and Taiwanese NSTC Grant No. 113-2119-M-007-013.

References

[1] Y. Meurice, R. Sakai and J. Unmuth-Yockey, *Tensor lattice field theory for renormalization and quantum computing*, *Rev. Mod. Phys.* **94** (2022) 025005 [2010.06539].

[2] M.C. Bañuls, *Tensor Network Algorithms: A Route Map*, *Ann. Rev. Condensed Matter Phys.* **14** (2023) 173 [2205.10345].

[3] M. Levin and C.P. Nave, *Tensor renormalization group approach to two-dimensional classical lattice models*, *Phys. Rev. Lett.* **99** (2007) 120601 [cond-mat/0611687].

[4] X. Luo and Y. Kuramashi, *Tensor renormalization group approach to* $(1+1)$*-dimensional su(2) principal chiral model at finite density*, *Phys. Rev. D* **107** (2023) 094509 [2208.13991].

[5] K. Nakayama, L. Funcke, K. Jansen, Y.-J. Kao and S. Kühn, *Phase structure of the CP(1) model in the presence of a topological* θ*-term*, *Phys. Rev. D* **105** (2022) 054507 [2107.14220].

[6] Y. Kuramashi and Y. Yoshimura, *Three-dimensional finite temperature* Z_2 *gauge theory with tensor network scheme*, *JHEP* **08** (2019) 023 [1808.08025].

[7] A. Yosprakob and K. Okunishi, *Tensor renormalization group study of the three-dimensional SU(2) and SU(3) gauge theories with the reduced tensor network formulation*, 2406.16763.

[8] K. Nakayama and M. Schneider, *Initial tensor construction and dependence of the tensor renormalization group on initial tensors*, *Phys. Rev. D* **110** (2024) 094501 [2407.14226].

[9] K.H. Pai, S. Akiyama and S. Todo, *Grassmann tensor renormalization group approach to* $(1+1)$*-dimensional two-color lattice QCD at finite density*, 2410.09485.

[10] Y. Liu, Y. Meurice, M.P. Qin, J. Unmuth-Yockey, T. Xiang, Z.Y. Xie et al., *Exact blocking formulas for spin and gauge models*, *Phys. Rev. D* **88** (2013) 056005 [1307.6543].

[11] H.H. Zhao, Z.Y. Xie, Q.N. Chen, Z.C. Wei, J.W. Cai and T. Xiang, *Renormalization of tensor-network states*, *Phys. Rev. B* **81** (2010) 174411 [1002.1405].

[12] Z.Y. Xie, J. Chen, M.P. Qin, J.W. Zhu, L.P. Yang and T. Xiang, *Coarse-graining renormalization by higher-order singular value decomposition*, *Phys. Rev. B* **86** (2012) 045139 [1201.1144].

[13] N.A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Springer New York (1999), 10.1007/b116436.

[14] S. Iino, S. Morita and N. Kawashima, *Boundary tensor renormalization group*, *Phys. Rev. B* **100** (2019) 035449 [1905.02351].

[15] D. Adachi, T. Okubo and S. Todo, *Anisotropic tensor renormalization group*, *Phys. Rev. B* **102** (2020) 054432 [1906.02007].

[16] B. Kaufman, *Crystal statistics. ii. partition function evaluated by spinor analysis*, *Phys. Rev.* **76** (1949) 1232.

[17] D. Kadoh and K. Nakayama, *Renormalization group on a triad network*, 1912.02414.

[18] K. Nakayama, *Randomized higher-order tensor renormalization group*, 2307.14191.