Chapter 3
Two-Dimensional Tensor Networks and Contraction Algorithms

Abstract In this section, we will first demonstrate in Sect. 3.1 that many important physical problems can be transformed to 2D TNs, and the central tasks become to compute the corresponding TN contractions. From Sects. 3.2 to 3.5, we will then present several paradigm contraction algorithms of 2D TNs including TRG, TEBD, and CTMRG. Relations to other distinguished algorithms and the exactly contractible TNs will also be discussed.

3.1 From Physical Problems to Two-Dimensional Tensor Networks

3.1.1 Classical Partition Functions

Partition function, which is a function of the variables of a thermodynamic state such as temperature, volume, and etc., contains the statistical information of a thermodynamic equilibrium system. From its derivatives of different orders, we can calculate the energy, free energy, entropy, and so on. Levin and Nave pointed out in Ref. [1] that the partition functions of statistical lattice models (such as Ising and Potts models) with local interactions can be written in the form of TN. Without losing generality, we take square lattice as an example.

Let us start from the simplest case: the classical Ising model on a single square with only four sites. The four Ising spins denoted by s_i $(i = 1, 2, 3, 4)$ locate on the four corners of the square, as shown in Fig. 3.1a; each spin can be up or down, represented by $s_i = 0$ and 1, respectively. The classical Hamiltonian of such a system reads

$$H_{s_1 s_2 s_3 s_4} = J(s_1 s_2 + s_2 s_3 + s_3 s_4 + s_4 s_1) - h(s_1 + s_2 + s_3 + s_4), \tag{3.1}$$

with J the coupling constant and h the magnetic field.

S.-J. Ran et al., *Tensor Network Contractions*, Lecture Notes in Physics 964,
https://doi.org/10.1007/978-3-030-34489-4_3

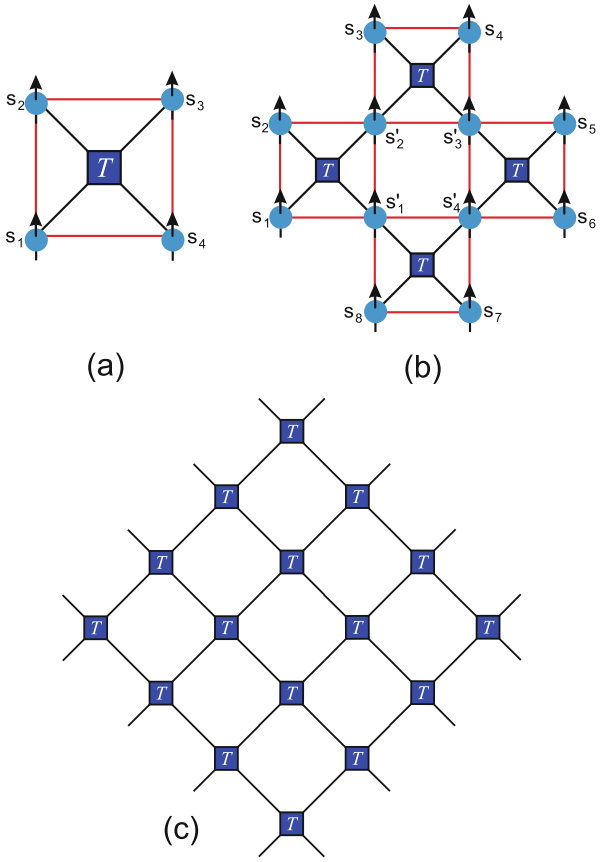

Fig. 3.1 (**a**) Four Ising spins (blue balls with arrows) sitting on a single square, and the red lines represent the interactions. The blue block is the tensor T (Eq. (3.2)), with the black lines denoting the indexes of T. (**b**) The graphic representation of the TN on a larger lattice with more than one square. (**c**) The TN construction of the partition function on infinite square lattice

When the model reaches the equilibrium at temperature T, the probability of each possible spin configuration is determined by the Maxwell–Boltzmann factor

$$T_{s_1 s_2 s_3 s_4} = e^{-\beta H_{s_1 s_2 s_3 s_4}}, \tag{3.2}$$

with the inverse temperature $\beta = 1/\mathrm{T}$.[1] Obviously, Eq. (3.2) is a fourth-order tensor T, where each element gives the probability of the corresponding configuration.

[1] In this paper, we set Boltzmann constant $k_B = 1$ for convenience.

The partition function is defined as the summation of the probability of all configurations. In the language of tensor, it is obtained by simply summing over all indexes as

$$Z = \sum_{s_1 s_2 s_3 s_4} T_{s_1 s_2 s_3 s_4}. \tag{3.3}$$

Let us proceed a little bit further by considering four squares, whose partition function can be written in a TN with four tensors (Fig. 3.1b) as

$$Z = \sum_{\{ss'\}} T_{s_1 s_2 s_2' s_1'} T_{s_2' s_3 s_4 s_3'} T_{s_4' s_3' s_5 s_6} T_{s_8 s_1' s_4' s_7}. \tag{3.4}$$

Each of the indexes $\{s'\}$ inside the TN is shared by two tensors, representing the spin that appears in both of the squares. The partition function is obtained by summing over all indexes.

For the infinite square lattice, the probability of a certain spin configuration $(s_1, s_2, \cdots)$ is given by the product of infinite number of tensor elements as

$$e^{-\beta H_{\{s\}}} = e^{-\beta H_{s_1 s_2 s_3 s_4}} e^{-\beta H_{s_4 s_5 s_6 s_7}} \cdots = T_{s_1 s_2 s_3 s_4} T_{s_4 s_5 s_6 s_7} \cdots \tag{3.5}$$

Then the partition function is given by the contraction of an infinite TN formed by the copies of T (Eq. (3.2)) as

$$Z = \sum_{\{s\}} \prod_n T_{s_1^n s_2^n s_3^n s_4^n}, \tag{3.6}$$

where two indexes satisfy $s_j^n = s_k^m$ if they refer to the same Ising spin. The graphic representation of Eq. (3.6) is shown in Fig. 3.1c. One can see that on square lattice, the TN still has the geometry of a square lattice. In fact, such a way will give a TN that has a geometry of the dual lattice of the system, and the dual of the square lattice is itself.

For the Q-state Potts model on square lattice, the partition function has the same TN representation as that of the Ising model, except that the elements of the tensor are given by the Boltzmann weight of the Potts model and the dimension of each index is Q. Note that the Potts model with $q = 2$ is equivalent to the Ising model.

Another example is the eight-vertex model proposed by Baxter in 1971 [2]. It is one of the "ice-type" statistic lattice model, and can be considered as the classical correspondence of the Z_2 spin liquid state. The tensor that gives the TN of the partition function is also $(2 \times 2 \times 2 \times 2)$, whose non-zero elements are

$$T_{s_1, \cdots, s_N} = \begin{cases} 1, & s_1 + \cdots + s_N = even, \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

We shall remark that there are more than one ways to define the TN of the partition function of a classical system. For example, when there only exist nearest-neighbor couplings, one can define a matrix $M_{ss'} = e^{-\beta H_{ss'}}$ on each bond and put on each site a *super-digonal* tensor I (or called copy tensor) defined as

$$I_{s_1,\cdots,s_N} = \begin{cases} 1, & s_1 = \cdots = s_N; \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

Then the TN of the partition function is the contraction of copies of M and I, and possesses exactly the same geometry of the original lattice (instead of the dual one).

3.1.2 Quantum Observables

With a TN state, the computations of quantum observables as $\langle\psi|\hat{O}|\psi\rangle$ and $\langle\psi|\psi\rangle$ are the contraction of a scalar TN, where $\hat{O}$ can be any operator. For a 1D MPS, this can be easily calculated, since one only needs to deal with a 1D TN stripe. For 2D PEPS, such calculations become contractions of 2D TNs. Taking $\langle\psi|\psi\rangle$ as an example, the TN of such an inner product is the contraction of the copies of the local tensor (Fig. 3.1c) defined as

$$T_{a_1a_2a_3a_4} = \sum_s P^*_{s,a_1''a_2''a_3''a_4''} P_{s,a_1'a_2'a_3'a_4'}, \tag{3.9}$$

with P the tensor of the PEPS and $a_i = (a_i', a_i'')$. There are no open indexes left and the TN gives the scalar $\langle\psi|\psi\rangle$. The TN for computing the observable $\langle\hat{O}\rangle$ is similar. The only difference is that we should substitute some small number of $T_{a_1a_2a_3a_4}$ in original TN of $\langle\psi|\psi\rangle$ with "impurities" at the sites where the operators locate. Taking one-body operator as an example, the "impurity" tensor on this site can be defined as

$$\widetilde{T}^{[i]}_{a_1a_2a_3a_4} = \sum_{s,s'} P^*_{s,a_1''a_2''a_3''a_4''} \hat{O}^{[i]}_{s,s'} P_{s',a_1'a_2'a_3'a_4'}. \tag{3.10}$$

In such a case, the single-site observables can be represented by the TN contraction of

$$\frac{\langle\psi|\hat{O}^{[i]}|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\mathrm{tTr}\ \ \widetilde{T}^{[i]} \prod_{n\neq i} T}{\mathrm{tTr}\ \ \prod_{n=1}^{N} T}. \tag{3.11}$$

For some non-local observables, e.g., the correlation function, the contraction of $\langle\psi|\hat{O}^{[i]}\hat{O}^{[j]}|\psi\rangle$ is nothing but adding another "impurity" by

$$\langle\psi|\hat{O}^{[i]}\hat{O}^{[j]}|\psi\rangle = \mathrm{tTr}\ \widetilde{T}^{[i]}\widetilde{T}^{[j]}\prod_{n\neq i,j}^{N} T. \tag{3.12}$$

3.1.3 Ground-State and Finite-Temperature Simulations

Ground-state simulations of 1D quantum models with short-range interactions can also be efficiently transferred to 2D TN contractions. When minimizing the energy

$$E = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}, \tag{3.13}$$

where we write $|\psi\rangle$ as an MPS. Generally speaking, there are two ways to solve the minimization problem: (1) simply treat all the tensor elements as variational parameters; (2) simulate the imaginary-time evolution

$$|\psi_{gs}\rangle = \lim_{\beta\to\infty}\frac{e^{-\beta\hat{H}}|\psi\rangle}{\parallel e^{-\beta\hat{H}}|\psi\rangle\parallel}. \tag{3.14}$$

The first way can be realized by, e.g., Monte Carlo methods where one could randomly change or choose the value of each tensor element to locate the minimal of energy. One can also use the Newton method and solve the partial-derivative equations $\partial E/\partial x_n = 0$ with x_n standing for an arbitrary variational parameter. Anyway, it is inevitable to calculate E (i.e., $\langle\psi|\hat{H}|\psi\rangle$ and $\langle\psi|\psi\rangle$) for most cases, which is to contraction the corresponding TNs as explained above.

We shall stress that without TN, the dimension of the ground state (i.e., the number of variational parameters) increases exponentially with the system size, which makes the ground-state simulations impossible for large systems.

The second way of computing the ground state with imaginary-time evolution is more or less like an "annealing" process. One starts from an arbitrarily chosen initial state and acts the imaginary-time evolution operator on it. The "temperature" is lowered a little for each step, until the state reaches a fixed point. Mathematically speaking, by using Trotter-Suzuki decomposition, such an evolution is written in a TN defined on $(D+1)$-dimensional lattice, with D the dimension of the real space of the model.

Here, we take a 1D chain as an example. We assume that the Hamiltonian only contains at most nearest-neighbor couplings, which reads

$$\hat{H} = \sum_n \hat{h}_{n,n+1}, \tag{3.15}$$

with $\hat{h}_{n,n+1}$ containing the on-site and two-body interactions of the n-th and $n+1$-th sites. It is useful to divide $\hat{H}$ into two groups, $\hat{H} = \hat{H}^e + \hat{H}^o$ as

$$\hat{H}^e \equiv \sum_{even\ n} \hat{h}_{n,n+1}, \quad \hat{H}^o \equiv \sum_{odd\ n} \hat{h}_{n,n+1}. \tag{3.16}$$

By doing so, each two terms in $\hat{H}^e$ or $\hat{H}^o$ commutes with each other. Then the evolution operator $\hat{U}(\tau)$ for infinitesimal imaginary time $\tau \to 0$ can be written as

$$\hat{U}(\tau) = e^{-\tau \hat{H}} = e^{-\tau \hat{H}^e} e^{-\tau \hat{H}^o} + O\left(\tau^2\right)\left[\hat{H}^e, \hat{H}^o\right]. \tag{3.17}$$

If τ is small enough, the high-order terms are negligible, and the evolution operator becomes

$$\hat{U}(\tau) \simeq \prod_n \hat{U}(\tau)_{n,n+1}, \tag{3.18}$$

with the two-site evolution operator $\hat{U}(\tau)_{n,n+1} = e^{-\tau \hat{H}_{n,n+1}}$.

The above procedure is known as the first-order Trotter-Suzuki decomposition [3–5]. Note that higher-order decomposition can also be adopted. For example, one may use the second-order Trotter-Suzuki decomposition that is written as

$$e^{-\tau \hat{H}} \simeq e^{-\frac{\tau}{2} \hat{H}^e} e^{-\tau \hat{H}^o} e^{-\frac{\tau}{2} \hat{H}^e}. \tag{3.19}$$

With Eq. (3.18), the time evolution can be transferred to a TN, where the local tensor is actually the coefficients of $\hat{U}(\tau)_{n,n+1}$, satisfying

$$T_{s_n s_{n+1} s'_n s'_{n+1}} = \langle s'_n s'_{n+1} | \hat{U}(\tau)_{n,n+1} | s_n s_{n+1} \rangle. \tag{3.20}$$

Such a TN is defined in a plain of two dimensions that corresponds to the spatial and (real or imaginary) time, respectively. The initial state is located at the bottom of the TN ($\beta = 0$) and its evolution is to do the TN contraction which can be efficiently solved by TN algorithms (presented later).

In addition, one can readily see that the evolution of a 2D state leads to the contraction of a 3D TN. Such a TN scheme provides a straightforward picture to understand the equivalence between a $(d+1)$-dimensional classical and a d-dimensional quantum theory. Similarly, the finite-temperature simulations of a quantum system can be transferred to TN contractions with Trotter-Suzuki decomposition. For the density operator $\hat{\rho}(\beta) = e^{-\beta \hat{H}}$, the TN is formed by the same tensor given by Eq. (3.20).

3.2 Tensor Renormalization Group

In 2007, Levin and Nave proposed TRG approach [1] to contract the TN of 2D classical lattice models. In 2008, Gu et al. further developed TRG to handle 2D quantum topological phases [6]. TRG can be considered as a coarse-graining contraction algorithm. To introduce the TRG algorithm, let us consider a square TN formed by infinite number of copies of a fourth-order tensor $T_{a_1a_2a_3a_4}$ (see the left side of Fig. 3.2).

Contraction and Truncation The idea of TRG is to iteratively "coarse-grain" the TN without changing the bond dimensions, the geometry of the network, and the translational invariance. Such a process is realized by two local operations in each iteration. Let us denote the tensor in the t-th iteration as $T^{(t)}$ (we take $T^{(0)} = T$). For obtaining $T^{(t+1)}$, the first step is to decompose $T^{(t)}$ by SVD in two different ways (Fig. 3.2) as

$$T^{(t)}_{a_1a_2a_3a_4} = \sum_b U_{a_1a_2b} V_{a_3a_4b}, \tag{3.21}$$

$$T^{(t)}_{a_1a_2a_3a_4} = \sum_b X_{a_4a_1b} Y_{a_2a_3b}. \tag{3.22}$$

Note that the singular value spectrum can be handled by multiplying it with the tensor(s), and the dimension of the new index satisfies $dim(b) = \chi^2$ with χ the dimension of each bond of $T^{(t)}$.

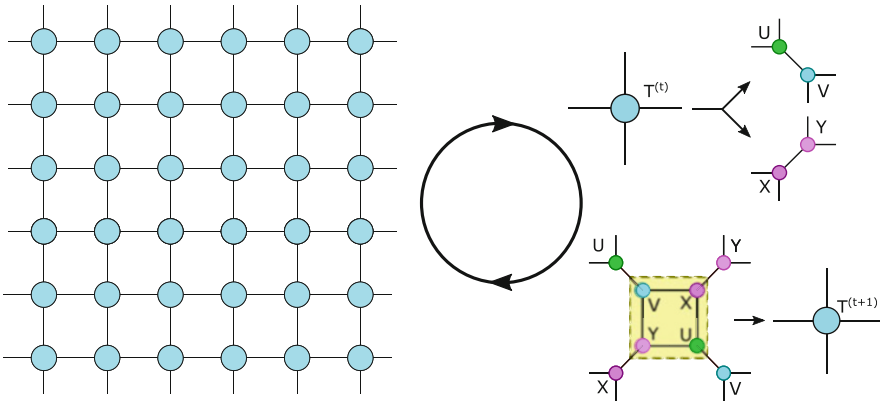

Fig. 3.2 For an infinite square TN with translational invariance, the renormalization in the TRG algorithm is realized by two local operations of the local tensor. After each iteration, the bond dimensions of the tensor and the geometry of the network keep unchanged

The purpose of the first step is to deform the TN, so that in the second step, a new tensor $T^{(t+1)}$ can be obtained by contracting the four tensors that form a square (Fig. 3.2) as

$$T^{(t+1)}_{b_1b_2b_3b_4} \leftarrow \sum_{a_1a_2a_3a_4} V_{a_1a_2b_1} Y_{a_2a_3b_2} U_{a_3a_4b_3} X_{a_4a_1b_4}. \tag{3.23}$$

We use an arrow instead of the equal sign, because one may need to divide the tensor by a proper number to keep the value of the elements from being divergent. The arrows will be used in the same way below.

These two steps define the contraction strategy of TRG. By the first step, the number of tensors in the TN (i.e., the size of the TN) increases from N to $2N$, and by the second step, it decreases from $2N$ to $N/2$. Thus, after t times of each iterations, the number of tensors decreases to the $\frac{1}{2^t}$ of its original number. For this reason, TRG is an *exponential contraction algorithm*.

Error and Environment The dimension of the tensor at the t-th iteration becomes χ^{2^t}, if no truncations are implemented. This means that truncations of the bond dimensions are necessary. In its original proposal, the dimension is truncated by only keeping the singular vectors of the χ-largest singular values in Eq. (3.22). Then the new tensor $T^{(t+1)}$ obtained by Eq. (3.23) has exactly the same dimension as $T^{(t)}$.

Each truncation will absolutely introduce some error, which is called the *truncation error*. Consistent with Eq. (2.7), the truncation error is quantified by the discarded singular values λ as

$$\varepsilon = \frac{\sqrt{\sum_{b=\chi}^{\chi^2-1} \lambda_b^2}}{\sqrt{\sum_{b=0}^{\chi^2-1} \lambda_b^2}}. \tag{3.24}$$

According to the linear algebra, ε in fact gives the error of the SVD given in Eq. (3.22), meaning that such a truncation minimizes the error of reducing the rank of $T^{(t)}$, which reads

$$\varepsilon = |T^{(t)}_{a_1a_2a_3a_4} - \sum_{b=0}^{\chi-1} U_{a_1a_2b} V_{a_3a_4b}|. \tag{3.25}$$

One may repeat the contraction-and-truncation process until $T^{(t)}$ converges. It usually only takes $\sim$10 steps, after which one in fact contract a TN of 2^t tensors to a single tensor.

The truncation is optimized according to the SVD of $T^{(t)}$. Thus, $T^{(t)}$ is called the *environment*. In general, the tensor(s) that determines the truncations is called the environment. It is a key factor to the accuracy and efficiency of the algorithm. For those that use local environments, like TRG, the efficiency is relatively high since the truncations are easy to compute. But, the accuracy is bounded since the truncations are only optimized according to some local information (like in TRG the local partitioning $T^{(t)}$).

One may choose other tensors or even the whole TN as the environment. In 2009, Xie et al. proposed the second renormalization group (SRG) algorithm [7]. The idea is in each truncation step of TRG, they define the global environment that is a fourth-order tensor $\mathscr{E}^{}_{a_1^{\tilde{n}}a_2^{\tilde{n}}a_3^{\tilde{n}}a_4^{\tilde{n}}} = \sum_{\{a\}} \prod_{n\neq\tilde{n}} T^{(n,t)}_{a_1^n a_2^n a_3^n a_4^n}$ with $T^{(n,t)}$ the n-th tensor in the t-th step and $\tilde{n}$ the tensor to be truncated. $\mathscr{E}$ is the contraction of the whole TN after getting rid of $T^{(\tilde{n},t)}$, and is computed by TRG. Then the truncation is obtained not by the SVD of $T^{(\tilde{n},t)}$, but by the SVD of $\mathscr{E}$. The word "second" in the name of the algorithm comes from the fact that in each step of the original TRG, they use a second TRG to calculate the environment. SRG is obviously more consuming, but bears much higher accuracy than TRG. The balance between accuracy and efficiency, which can be controlled by the choice of environment, is one main factor to consider while developing or choosing the TN algorithms.

3.3 Corner Transfer Matrix Renormalization Group

In the 1960s, the corner transfer matrix (CTM) idea was developed originally by Baxter in Refs. [8, 9] and a book [10]. Such ideas and methods have been applied to various models, for example, the chiral Potts model [11–13], the 8-vertex model [2, 14, 15], and to the 3D Ising model [16]. Combining CTM with DMRG, Nishino and Okunishi proposed the CTMRG [17] in 1996 and applied it to several models [17–27]. In 2009, Orús and Vidal further developed CTMRG to deal with TNs [28]. What they proposed to do is to put eight *variational tensors* to be optimized in the algorithm, which are four corner transfer matrices $C^{[1]}$, $C^{[2]}$, $C^{[3]}$, $C^{[4]}$ and four row (column) tensors $R^{[1]}$, $R^{[2]}$, $R^{[3]}$, $R^{[4]}$, on the boundary, and then to contract the tensors in the TN to these variational tensors in a specific order shown in Fig. 3.3. The TN contraction is considered to be solved with the variational tensors when they converge in this contraction process. Compared with the boundary-state methods in the last subsection, the tensors in CTMRG define the states on both the boundaries and corners.

Contraction In each iteration step of CTMRG, one choses two corner matrices on the same side and the row tensor between them, e.g., $C^{[1]}$, $C^{[2]}$, and $R^{[2]}$. The update of these tensors (Fig. 3.4) follows

$$\tilde{C}^{[1]}_{\tilde{b}_2 b_1'} \leftarrow \sum_{b_1} C^{[1]}_{b_1 b_2} R^{[1]}_{b_1 a_1 b_1'}, \tag{3.26}$$

$$\tilde{R}^{[2]}_{\tilde{b}_2 a_4 \tilde{b}_3} \leftarrow \sum_{a_2} R^{[2]}_{b_2 a_2 b_3} T_{a_1 a_2 a_3 a_4}, \tag{3.27}$$

$$\tilde{C}^{[2]}_{\tilde{b}_3 b_4'} \leftarrow \sum_{b_4} C^{[2]}_{b_3 b_4} R^{[3]}_{b_4 a_3 b_4'}, \tag{3.28}$$

where $\tilde{b}_2 = (b_2, a_1)$ and $\tilde{b}_3 = (b_3, a_1)$.

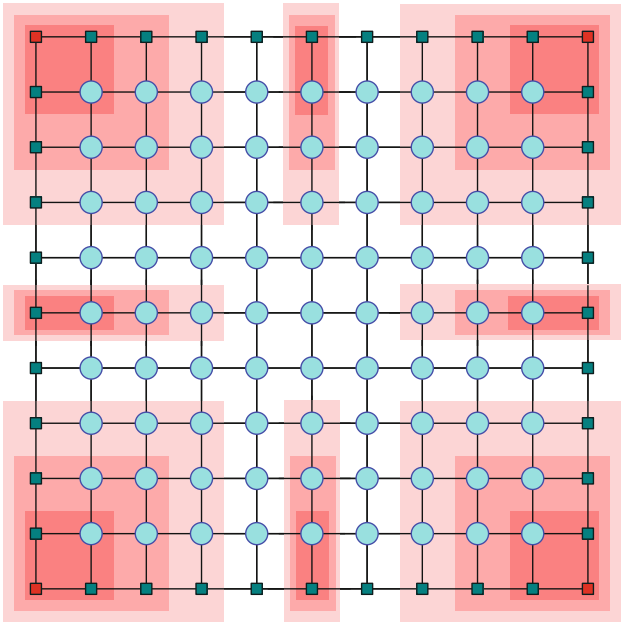

Fig. 3.3 Overview of the CTMRG contraction scheme. The tensors in the TN are contracted to the variational tensors defined on the edges and corners

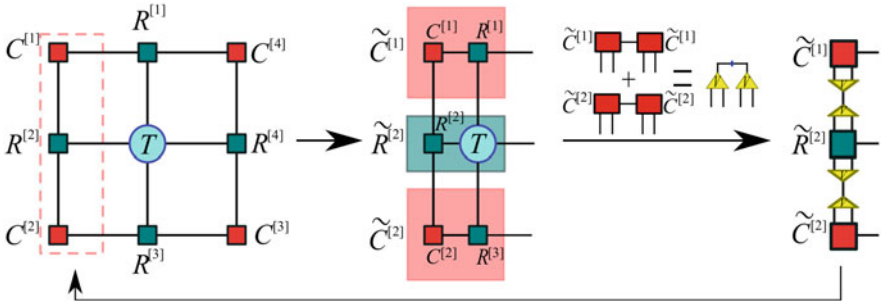

Fig. 3.4 The first arrow shows absorbing tensors $R^{[1]}$, T, and $R^{[3]}$ to renew tensors $C^{[1]}$, $R^{[2]}$, and $C^{[2]}$ in left operation. The second arrow shows the truncation of the enlarged bond of $\tilde{C}^{[1]}$, $\tilde{R}^{[2]}$, and $\tilde{C}^{[2]}$. Inset is the acquisition of the truncation matrix Z

After the contraction given above, it can be considered that one column of the TN (as well as the corresponding row tensors $R^{[1]}$ and $R^{[3]}$) is contracted. Then one chooses other corner matrices and row tensors (such as $\tilde{C}^{[1]}$, $C^{[4]}$, and $R^{[1]}$) and implement similar contractions. By iteratively doing so, the TN is contracted in the way shown in Fig. 3.3.

Note that for a finite TN, the initial corner matrices and row tensors should be taken as the tensors locating on the boundary of the TN. For an infinite TN, they can be initialized randomly, and the contraction should be iterated until the preset convergence is reached.

CTMRG can be regarded as a *polynomial contraction scheme*. One can see that the number of tensors that are contracted at each step is determined by the length of the boundary of the TN at each iteration time. When contracting a 2D TN defined on a $(L \times L)$ square lattice as an example, the length of each side is $L-2t$ at the t-th step. The boundary length of the TN (i.e., the number of tensors contracted at the t-th step) bears a linear relation with t as $4(L-2t)-4$. For a 3D TN such as cubic TN, the boundary length scales as $6(L-2t)^2-12(L-2t)+8$, thus the CTMRG for a 3D TN (if exists) gives a polynomial contraction.

Truncation One can see that after the contraction in each iteration step, the bond dimensions of the variational tensors increase. Truncations are then in need to prevent the excessive growth of the bond dimensions. In Ref. [28], the truncation is obtained by inserting a pair of isometries V and $V^\dagger$ in the enlarged bonds. A reasonable (but not the only choice) of V for translational invariant TN is to consider the eigenvalue decomposition on the sum of corner transfer matrices as

$$\sum_b \tilde{C}^{[1]\dagger}_{\tilde{b}b}\tilde{C}^{[1]}_{\tilde{b}'b} + \sum_b \tilde{C}^{[2]\dagger}_{\tilde{b}b}\tilde{C}^{[1]}_{\tilde{b}'b} \simeq \sum_{b=0}^{\chi-1} V_{\tilde{b}b}\Lambda_b V^*_{\tilde{b}'b}. \tag{3.29}$$

Only the χ largest eigenvalues are preserved. Therefore, V is a matrix of the dimension $D\chi \times \chi$, where D is the bond dimension of T and χ is the dimension cut-off. We then truncate $\tilde{C}^{[1]}$, $\tilde{R}^{[2]}$, and $\tilde{C}^{[2]}$ using V as

$$C^{[1]}_{b'_1 b_2} = \sum_{\tilde{b}_2} \tilde{C}^{[1]}_{\tilde{b}_2 b'_1} V^*_{\tilde{b}_2 b_2}, \tag{3.30}$$

$$R^{[2]}_{b_2 a_4 b_3} = \sum_{\tilde{b}_2,\tilde{b}_3} \tilde{R}^{[2]}_{\tilde{b}_2 a_4 \tilde{b}_3} V_{\tilde{b}_2 b_2} V^*_{\tilde{b}_3 b_3}, \tag{3.31}$$

$$C^{[2]}_{b_3 b'_4} = \sum_{\tilde{b}_3} \tilde{C}^{[2]}_{\tilde{b}_3 b'_4} V_{\tilde{b}_3 b_3}. \tag{3.32}$$

Error and Environment Same as TRG or TEBD, the truncations are obtained by the matrix decompositions of certain tensors that define the environment. From Eq. (3.29), the environment in CTMRG is the loop formed by the corner matrices and row tensors. Note that symmetries might be considered to accelerate the computation. For example, one may take $C^{[1]} = C^{[2]} = C^{[3]} = C^{[4]}$ and $R^{[1]} = R^{[2]} = R^{[3]} = R^{[4]}$ when the TN has rotational and reflection symmetries ($T_{a_1 a_2 a_3 a_4} = T_{a'_1 a'_2 a'_3 a'_4}$ after any permutation of the indexes).

3.4 Time-Evolving Block Decimation: Linearized Contraction and Boundary-State Methods

The TEBD algorithm by Vidal was developed originally for simulating the time evolution of 1D quantum models [29–31]. The (finite and infinite) TEBD algorithm has been widely applied to varieties of issues, such as criticality in quantum many-body systems (e.g., [32–34]), the topological phases [35], the many-body localization [36–38], and the thermodynamic property of quantum many-body systems [39–45].

In the language of TN, TEBD solves the TN contraction problems in a linearized manner, and the truncation is calculated in the context of an MPS. In the following, let us explain the infinite TEBD (iTEBD) algorithm [31] (Fig. 3.5) by still taking the infinite square TN formed by the copies of a fourth-order tensor T as an example. In each step, a row of tensors (which can be regarded as an MPO) are contracted to an MPS $|\psi\rangle$. Inevitably, the bond dimensions of the tensors in the MPS will increase exponentially as the contractions proceed. Therefore, truncations are necessary to prevent the bond dimensions diverging. The truncations are determined by minimizing the distance between the MPSs before and after the truncation. After the MPS $|\psi\rangle$ converges, the TN contraction becomes $\langle\psi|\psi\rangle$, which can be exactly and easily computed.

Contraction We use is two-site translational invariant MPS, which is formed by the tensors A and B on the sites and the spectrum Λ and Γ on the bonds as

$$\sum_{\{a\}} \cdots \Lambda_{a_{n-1}} A_{s_{n-1},a_{n-1}a_n} \Gamma_{a_n} B_{s_n,a_n a_{n+1}} \Lambda_{a_{n+1}} \cdots . \tag{3.33}$$

In each step of iTEBD, the contraction is given by

$$A_{s,\tilde{a}\tilde{a}'} \leftarrow \sum_{s'} T_{sbs'b'} A_{s',aa'}, \quad B_{s,\tilde{a}\tilde{a}'} \leftarrow \sum_{s'} T_{sbs'b'} B_{s',aa'}, \tag{3.34}$$

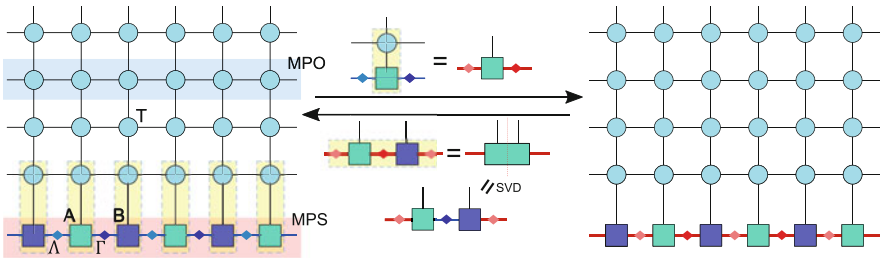

Fig. 3.5 The illustration of the contraction and truncation of the iTEBD algorithm. In each iteration step, a row of tensors in the TN are contracted to the MPS, and truncations by SVD are implemented so that the bond dimensions of the MPS keep unchanged

where the new virtual bonds are entangled, satisfying $\tilde{a} = (b, a)$ and $\tilde{a}' = (b', a')$. Meanwhile, the spectrum is also updated as

$$\Lambda_{\tilde{a}} \leftarrow \Lambda_a \mathbf{1}_b, \quad \Gamma_{\tilde{a}'} \leftarrow \Gamma_{a'} \mathbf{1}_{b'}, \tag{3.35}$$

where $\mathbf{1}$ is a vector with $\mathbf{1}_b = 1$ for any b.

It is readily to see that the number of tensors in iTEBD will be reduced linearly as tN, with t the number of the contraction-and-truncation steps and $N \to \infty$ the number of the columns of the TN. Therefore, iTEBD (also finite TEBD) can be considered as a *linearized contraction algorithm*, in contrast to the exponential contraction algorithm like TRG.

Truncation Truncations are needed when the dimensions of the virtual bonds exceed the preset dimension cut-off χ. In the original version of iTEBD [31], the truncations are done by local SVDs. To truncate the virtual bond $\tilde{a}$, for example, one defines a matrix by contracting the tensors and spectrum connected to the target bond as

$$M_{s_1\tilde{a}_1,s_2\tilde{a}_2} = \sum_{\tilde{a}} \Lambda_{\tilde{a}_1} A_{s_1,\tilde{a}_1\tilde{a}} \Gamma_{\tilde{a}} B_{s_2,\tilde{a}\tilde{a}_2} \Lambda_{\tilde{a}_2}. \tag{3.36}$$

Then, perform SVD on M, keeping only the χ-largest singular values and the corresponding basis as

$$M_{s_1\tilde{a}_1,s_2\tilde{a}_2} = \sum_{a=0}^{\chi-1} U_{s_1,\tilde{a}_1 a} \Gamma_a V_{s_2,a\tilde{a}_2}. \tag{3.37}$$

The spectrum Γ is updated by the singular values of the above SVD. The tensors A and B are also updated as

$$A_{s_1,\tilde{a}a} = (\Lambda_{\tilde{a}})^{-1} U_{s_1,\tilde{a}a}, \quad B_{s_2,a\tilde{a}} = V_{s_2,a\tilde{a}} (\Lambda_{\tilde{a}})^{-1}. \tag{3.38}$$

Till now, the truncation of the spectrum Γ and the corresponding virtual bond have been completed. Any spectra and virtual bonds can be truncated similarly.

Error and Environment Similar to TRG and SRG, the environment of the original iTEBD is M in Eq. (3.37), and the error is measured by the discarded singular values of M. Thus, iTEBD seems to only use local information to optimize the truncations. What is amazing is that when the MPO is unitary or near unitary, the MPS converges to a so-called *canonical form* [46, 47]. The truncations are then optimal by taking the whole MPS as the environment. If the MPO is far from being unitary, Orús and Vidal proposed the *canonicalization* algorithm [47] to transform the MPS into the canonical form before truncating. We will talk about this issue in detail in the next section.

Boundary-State Methods: Density Matrix Renormalization Group and Variational Matrix Product State The iTEBD can be understood as a boundary-state method. One may consider one row of tensors in the TN as an MPO (see Sect. 2.2.6 and Fig. 2.10), where the vertical bonds are the "physical" indexes and the bonds shared by two adjacent tensors are the geometrical indexes. This MPO is also called the *transfer operator* or *transfer MPO* of the TN. The converged MPS is in fact the dominant eigenstate of the MPO.[2] While the MPO represents a physical Hamiltonian or the imaginary-time evolution operator (see Sect. 3.1), the MPS is the ground state. For more general situations, e.g., the TN represents a 2D partition function or the inner product of two 2D PEPSs, the MPS can be understood as the *boundary state* of the TN (or the PEPS) [48–50]. The contraction of the 2D infinite TN becomes computing the boundary state, i.e., the dominant eigenstate (and eigenvalue) of the transfer MPO.

The boundary-state scheme gives several non-trivial physical and algorithmic implications [48–52], including the underlying resemblance between iTEBD and the famous infinite DMRG (iDMRG) [53]. DMRG [54, 55] follows the idea of Wilson's NRG [56], and solves the ground states and low-lying excitations of 1D or quasi-1D Hamiltonians (see several reviews [57–60]); originally it has no direct relations to TN contraction problems. After the MPS and MPO become well understood, DMRG was re-interpreted in a manner that is more close to TN (see a review by Schollwöck [57]). In particular for simulating the ground states of infinite-size 1D systems, the underlying connections between the iDMRG and iTEBD were discussed by McCulloch [53]. As argued above, the contraction of a TN can be computed by solving the dominant eigenstate of its transfer MPO. The eigenstates reached by iDMRG and iTEBD are the same state up to a gauge transformation (note the gauge degrees of freedom of MPS will be discussed in Sect. 2.4.2). Considering that DMRG mostly is not used to compute TN contractions and there are already several understanding reviews, we skip the technical details of the DMRG algorithms here. One may refer to the papers mentioned above if interested. However, later we will revisit iDMRG in the clue of multi-linear algebra.

Variational matrix product state (VMPS) method is a variational version of DMRG for (but not limited to) calculating the ground states of 1D systems with periodic boundary condition [61]. Compared with DMRG, VMPS is more directly related to TN contraction problems. In the following, we explain VMPS by solving the contraction of the infinite square TN. As discussed above, it is equivalent to solve the dominant eigenvector (denoted by $|\psi\rangle$) of the infinite MPO (denoted by $\hat{rho}$) that is formed by a row of tensors in the TN. The task is to minimize $\langle\psi|\hat{\rho}|\psi\rangle$ under the constraint $\langle\psi|\psi\rangle = 1$. The eigenstate $|\psi\rangle$ written in the form of an MPS.

The tensors in $|\psi\rangle$ are optimized on by one. For instance, to optimize the n-th tensor, all other tensors are kept unchanged and considered as constants. Such a local minimization problem becomes $\hat{H}^{eff}|T_n\rangle = \mathscr{E}\hat{N}^{eff}|T_n\rangle$ with $\mathscr{E}$ the eigenvalue.

[2]For simplicity, we assume the MPO gives an Hermitian operator so that its eigenstates and eigenvalues are well-defined.

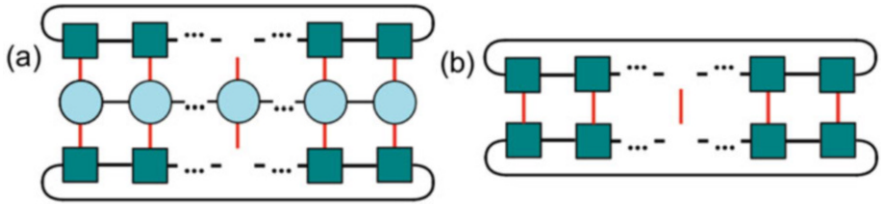

Fig. 3.6 The illustration of (**a**) $\hat{H}^{eff}$ and (**b**) $\hat{N}^{eff}$ in the variational matrix product state method

$\hat{H}^{eff}$ is given by a sixth-th order tensor defined by contracting all tensors in $\langle\psi|\hat{\rho}|\psi\rangle$ except for the n-th tensor and its conjugate (Fig. 3.6a). Similarly, $\hat{N}^{eff}$ is also given by a sixth-th order tensor defined by contracting all tensors in $\langle\psi|\psi\rangle$ except for the n-th tensor and its conjugate (Fig. 3.6b). Again, the VMPS is different from the MPS obtained by TEBD only up to a gauge transformation.

Note that the boundary-state methods are not limited to solving TN contractions. An example is the time-dependent variational principle (TDVP). The basic idea of TDVP was proposed by Dirac in 1930 [62], and then it was cooperated with the formulation of Hamiltonian [63] and action function [64]. For more details, one could refer to a review by Langhoff et al. [65]. In 2011, TDVP was developed to simulate the time evolution of many-body systems with the help of MPS [66]. Since TDVP (and some other algorithms) concerns directly a quantum Hamiltonian instead of the TN contraction, we skip giving more details of these methods in this paper.

3.5 Transverse Contraction and Folding Trick

For the boundary-state methods introduced above, the boundary states are defined in the real space. Taking iTEBD for the real-time evolution as an example, the contraction is implemented along the time direction, which is to do the time evolution in an explicit way. It is quite natural to consider implementing the contraction along the other direction. In the following, we will introduce the transverse contraction and the folding trick proposed and investigated in Refs. [67–69]. The motivation of transverse contraction is to avoid the explicit simulation of the time-dependent state $|\psi(t)\rangle$ that might be difficult to capture due to the fast growth of its entanglement.

Transverse Contraction Let us consider to calculate the average of a one-body operator $o(t) = \langle\psi(t)|\hat{o}|\psi(t)\rangle$ with $|\psi(t)\rangle$ that is a quantum state of infinite size evolved to the time t. The TN representing $o(t)$ is given in the left part of Fig. 3.7, where the green squares give the initial MPS $|\psi(0)\rangle$ and its conjugate, the yellow diamond is $\hat{o}$, and the TN formed by the green circles represents the evolution operator $e^{it\hat{H}}$ and its conjugate (see how to define the TN in Sect. 3.1.3).

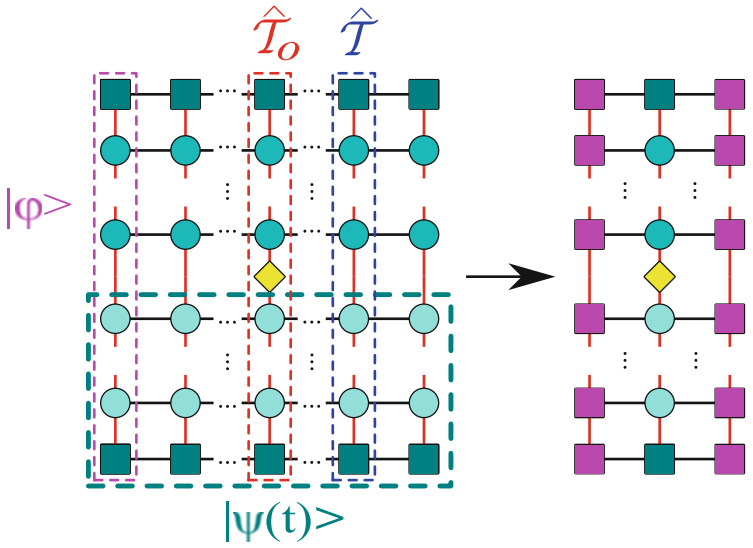

Fig. 3.7 Transverse contraction of the TN for a local expectation value $\langle O(t)\rangle$

To perform the transverse contraction, we treat each column of the TN as an MPO $\hat{\mathcal{T}}$. Then as shown in the right part of Fig. 3.7, the main task of computing $o(t)$ is to solve the dominant eigenstate $|\phi\rangle$ (normalized) of $\hat{\mathcal{T}}$, which is an MPS illustrated by the purple squares. One may solve this eigenstate problems by any of the boundary-state methods (TEBD, DMRG, etc.). With $|\phi\rangle$, $o(t)$ can be exactly and efficiently calculated as

$$o(t) = \frac{\langle\psi(t)|\hat{o}|\psi(t)\rangle}{\langle\psi(t)|\psi(t)\rangle} = \frac{\langle\phi|\hat{\mathcal{T}}_o|\phi\rangle}{\langle\phi|\hat{\mathcal{T}}|\phi\rangle}, \tag{3.39}$$

with $\hat{\mathcal{T}}_o$ is the column that contains the operator $\hat{o}$. Note that the length of $|\phi\rangle$ (i.e., the number of tensors in the MPS) is proportional to the time t, thus one should use the finite-size versions of the boundary-state methods. It should also be noted that $\hat{\mathcal{T}}$ may not be Hermitian. In this case, one should not use $|\phi\rangle$ and its conjugate, but compute the left and right eigenstates of $\hat{\mathcal{T}}$ instead.

Interestingly, similar ideas of the transverse contraction appeared long before the concept of TN emerged. For instance, transfer matrix renormalization group (TMRG) [70–73] can be used to simulate the finite-temperature properties of a 1D system. The idea of TMRG is to utilize DMRG to calculate the dominant eigenstate of the transfer matrix (similar to $\mathcal{T}$). In correspondence with the TN terminology, it is to use DMRG to compute $|\phi\rangle$ from the TN that defines the imaginary-time evolution. We will skip of the details of TMRG since it is not directly related to TN. One may refer the related references if interested.

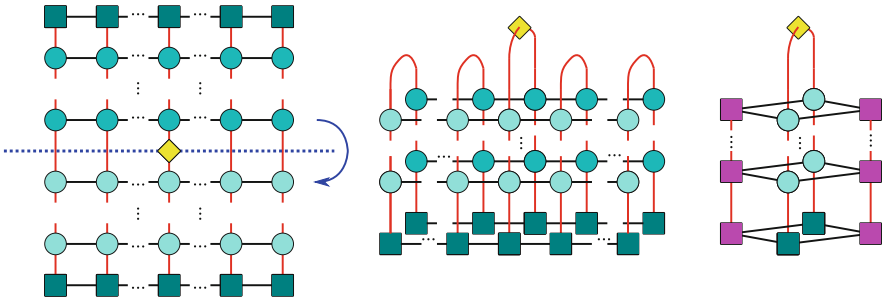

Fig. 3.8 The illustration of the folding trick

Folding Trick The main bottleneck of a boundary-state method concerns the entanglement of the boundary state. In other words, the methods will become inefficient when the entanglement of the boundary state grows too large. One example is the real-time simulation of a 1D chain, where the entanglement entropy increases linearly with time. Solely with the transverse contraction, it will not essentially solve this problem. Taking the imaginary-time evolution as an example, it has been shown that with the dual symmetry of space and time, the boundary states in the space and time directions possess the same entanglement [69, 74].

In Ref. [67], the folding trick was proposed. The idea is to "fold" the TN before the transverse contraction (Fig. 3.8). In the folded TN, each tensor is the tensor product of the original tensor and its conjugate. The length of the folded TN in the time direction is half of the original TN, and so is the length of the boundary state.

The previous work [67] on the dynamic simulations of 1D spin chains showed that the entanglement of the boundary state is in fact reduced compared with that of the boundary state without folding. This suggests that the folding trick provides a more efficient representation of the entanglement structure of the boundary state. The authors of Ref. [67] suggested an intuitive picture to understand the folding trick. Consider a product state as the initial state at $t - 0$ and a single localized excitation at the position x that propagates freely with velocity v. By evolving for a time t, only $(x \pm vt)$ sites will become entangled. With the folding trick, the evolutions (that are unitary) besides the $(x \pm vt)$ sites will not take effects since they are folded with the conjugates and become identities. Thus the spins outside $(x \pm vt)$ will remain product state and will not contribute entanglement to the boundary state. In short, one key factor to consider here is the entanglement structure, i.e., the fact that the TN is formed by unitaries. The transverse contraction with the folding trick is a convincing example to show that the efficiency of contracting a TN can be improved by properly designing the contraction way according to the entanglement structure of the TN.

3.6 Relations to Exactly Contractible Tensor Networks and Entanglement Renormalization

The TN algorithms explained above are aimed at dealing with contracting optimally the TNs that cannot be exactly contracted. Then a question arises: Is a classical computer really able to handle these TNs? In the following, we show that by explicitly putting the isometries for truncations inside, the TNs that are contracted in these algorithms become eventually exactly contractible, dubbed as exactly contractible TN (ECTN). Different algorithms lead to different ECTN. That means the algorithm will show a high performance if the TN can be accurately approximated by the corresponding ETNC.

Figure 3.9 shows the ECTN emerging in the plaquette renormalization [75] or higher-order TRG (HOTRG) algorithms [76]. Take the contraction of a TN (formed by the copies of tensor T) on square lattice as an example. In each iteration step, four nearest-neighbor Ts in a square are contracted together, which leads to a new square TN formed by tensors ($T^{(1)}$) with larger bond dimensions. Then, isometries (yellow

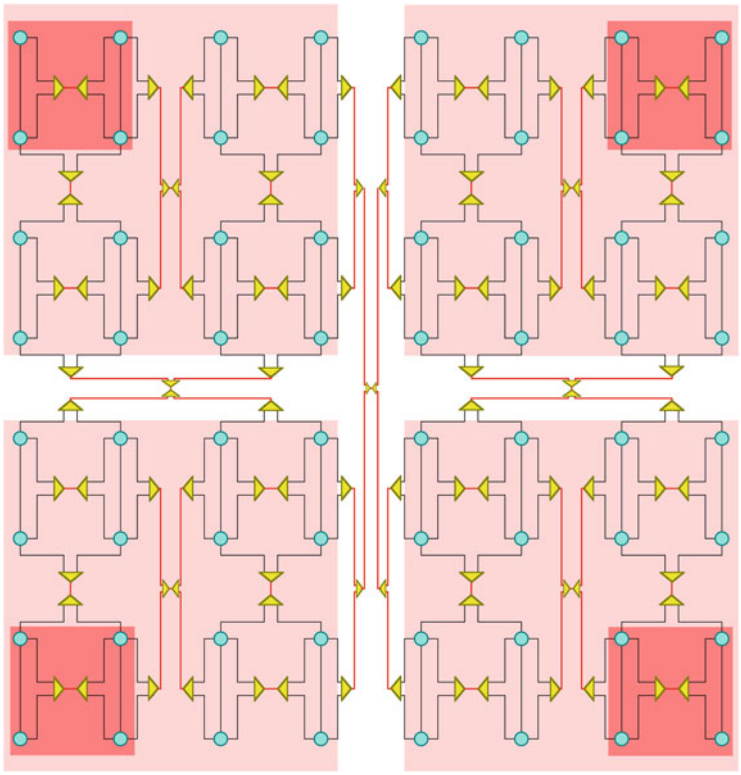

Fig. 3.9 The exactly contractible TN in the HOTRG algorithm

triangles) are inserted in the TN to truncate the bond dimensions (the truncations are in the same spirit of those in CTMRG, see Fig. 3.4). Let us not contract the isometries with the tensors, but leave them there inside the TN. Still, we can move on to the next iteration, where we contract four $T^{(1)}$'s (each of which is formed by four T and the isometries, see the dark-red plaques in Fig. 3.9) and obtain more isometries for truncating the bond dimensions of $T^{(1)}$. By repeating this process for several times, one can see that tree TNs appear on the boundaries of the coarse-grained plaques. Inside the 4-by-4 plaques (light red shadow), we have the two-layer tree TNs formed by three isometries. In the 8-by-8 plaques, the tree TN has three layers with seven isometries. These tree TNs separate the original TN into different plaques, so that it can be exactly contracted, similar to the fractal TNs introduced in Sect. 2.3.6.

In the iTEBD algorithm [29–31, 47] (Fig. 3.10), one starts with an initial MPS (dark-blue squares). In each iteration, one tensor (light blue circles) in the TN is contracted with the tensor in the MPS and then the bonds are truncated by isometries (yellow triangles). Globally seeing, the isometries separate the TN into many "tubes" (red shadow) that are connected only at the top. The length of the tubes equals to the number of the iteration steps in iTEBD. Obviously, this TN is exactly contractible. Such a tube-like structure also appears in the contraction algorithms based on PEPS.

For the CTMRG algorithm [28], the corresponding ECTN is a little bit complicated (see one quarter of it in Fig. 3.11). The initial row (column) tensors and the corner transfer matrices are represented by the pink and green squares. In each

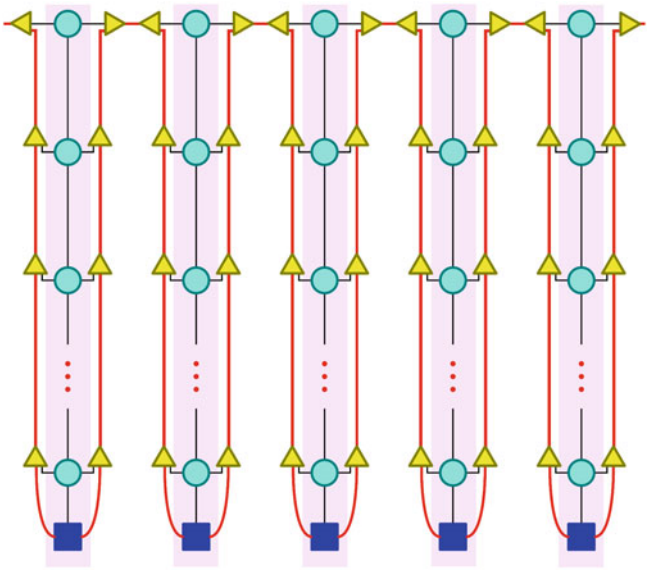

Fig. 3.10 The exactly contractible TN in the iTEBD algorithm

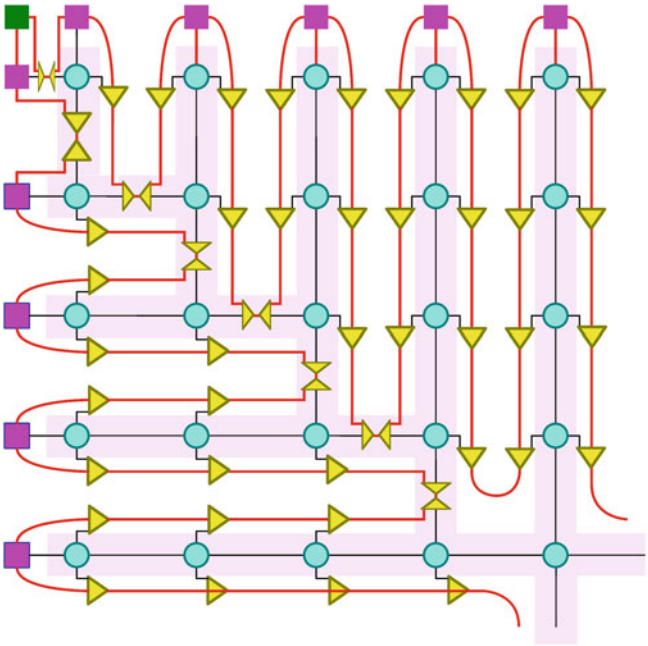

Fig. 3.11 A part of the exactly contractible TN in the CTMRG algorithm

iteration step, the tensors (light blue circles) located most outside are contracted to the row (column) tensors and the corner transfer matrices, and isometries are introduced to truncate the bond dimensions. Globally seeing the picture, the isometries separate the TN into a tree-like structure (red shadow), which is exactly contractible.

For these three algorithms, each of them gives an ECTN that is formed by two part: the tensors in the original TN and the isometries that make the TN exactly contractible. After optimizing the isometries, the original TN is approximated by the ECTN. The structure of the ECTN depends mainly on the contraction strategy and the way of optimizing the isometries depends on the chosen environment.

The ECTN picture shows us explicitly how the correlations and entanglement are approximated in different algorithms. Roughly speaking, the correlation properties can be read from the minimal distance of the path in the ECTN that connects two certain sites, and the (bipartite) entanglement can be read from the number of bonds that cross the boundary of the bipartition. How well the structure suits the correlations and entanglement should be a key factor of the performance of a TN contraction algorithm. Meanwhile, this picture can assist us to develop new algorithms by designing the ECTN and taking the whole ECTN as the environment for optimizing the isometries. These issues still need further investigations.

The unification of the TN contraction and the ECTN has been explicitly utilized in the TN renormalization (TNR) algorithm [77, 78], where both isometries and

unitaries (called *disentangler*) are put into the TN to make it exactly contractible. Then instead of tree TNs or MPSs, one will have MERAs (see Fig. 2.7c, for example) inside which can better capture the entanglement of critical systems.

3.7 A Shot Summary

In this section, we have discussed about several contraction approaches for dealing with 2D TNs. Applying these algorithms, many challenging problems can be efficiently solved, including the ground-state and finite-temperature simulations of 1D quantum systems, and the simulations of 2D classical statistic models. Such algorithms consist of two key ingredients: contractions (local operations of tensors) and truncations. The local contraction determines the way how the TN is contracted step by step, or in other words, how the entanglement information is kept according to the ECTN structure. Different (local or global) contractions may lead to different computational costs, thus optimizing the contraction sequence is necessary in many cases [67, 79, 80]. The truncation is the approximation to discard less important basis so that the computational costs are properly bounded. One essential concept in the truncations is "environment," which plays the role of the reference when determining the weights of the basis. Thus, the choice of environment concerns the balance between the accuracy and efficiency of a TN algorithm.

References

1. M. Levin, C.P. Nave, Tensor renormalization group approach to two-dimensional classical lattice models. Phys. Rev. Lett. **99**, 120601 (2007)
2. R.J. Baxter, Eight-vertex model in lattice statistics. Phys. Rev. Lett. **26**, 832–833 (1971)
3. H.F. Trotter, On the product of semi-groups of operators. Proc. Am. Math. Soc. **10**(4), 545–551 (1959)
4. M. Suzuki, M. Inoue, The ST-transformation approach to analytic solutions of quantum systems. I general formulations and basic limit theorems. Prog. Theor. Phys. **78**, 787 (1987)
5. M. Inoue, M. Suzuki, The ST-transformation approach to analytic solutions of quantum systems. II: transfer-matrix and Pfaffian methods. Prog. Theor. Phys. **79**(3), 645–664 (1988)
6. Z.C. Gu, M. Levin, X.G. Wen, Tensor-entanglement renormalization group approach as a unified method for symmetry breaking and topological phase transitions. Phys. Rev. B **78**, 205116 (2008)
7. Z.Y. Xie, H.C. Jiang, Q.N. Chen, Z.Y. Weng, T. Xiang, Second renormalization of tensor-network states. Phys. Rev. Lett. **103**, 160601 (2009)
8. R.J. Baxter, Dimers on a rectangular lattice. J. Math. Phys. **9**, 650 (1968)
9. R.J. Baxter, Variational approximations for square lattice models in statistical mechanics. J. Stat. Phys. **19**, 461 (1978)
10. R.J. Baxter, *Exactly Solved Models in Statistical Mechanics* (Elsevier, Amsterdam, 2016)
11. R.J. Baxter, Corner transfer matrices of the chiral Potts model. J. Stat. Phys. **63**, 433–453 (1991)

12. R.J. Baxter, Chiral Potts model: corner transfer matrices and parametrizations. Int. J. Mod. Phys. B **7**, 3489–3500 (1993)
13. R.J. Baxter, Corner transfer matrices of the chiral Potts model. II. The triangular lattice. J. Stat. Phys. **70**, 535–582 (1993)
14. R.J. Baxter, Corner transfer matrices of the eight-vertex model. I. Low-temperature expansions and conjectured properties. J. Stat. Phys. **15**, 485–503 (1976)
15. R.J. Baxter, Corner transfer matrices of the eight-vertex model. II. The Ising model case. J. Stat. Phys. **17**, 1–14 (1977)
16. R.J. Baxter, P.J. Forrester, A variational approximation for cubic lattice models in statistical mechanics. J. Phys. A Math. Gen. **17**, 2675–2685 (1984)
17. T. Nishino, K. Okunishi, Corner transfer matrix renormalization group method. J. Phys. Soc. Jpn. **65**, 891–894 (1996)
18. T. Nishino, Y. Hieida, K. Okunishi, N. Maeshima, Y. Akutsu, A. Gendiar, Two-dimensional tensor product variational formulation. Prog. Theor. Phys. **105**(3), 409–417 (2001)
19. T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, Y. Akutsu, Self-consistent tensor product variational approximation for 3D classical models. Nucl. Phys. B **575**(3), 504–512 (2000)
20. T. Nishino, K. Okunishi, A density matrix algorithm for 3D classical models. J. Phys. Soc. Jpn. **67**(9), 3066–3072 (1998)
21. K. Okunishi, T. Nishino, Kramers-Wannier approximation for the 3D Ising model. Prog. Theor. Phys. **103**(3), 541–548 (2000)
22. T. Nishino, K. Okunishi, Numerical latent heat observation of the $q = 5$ Potts model (1997). arXiv preprint cond-mat/9711214
23. T. Nishino, K. Okunishi, Corner transfer matrix algorithm for classical renormalization group. J. Phys. Soc. Jpn. **66**(10), 3040–3047 (1997)
24. N. Tsushima, T. Horiguchi, Phase diagrams of spin-3/2 Ising model on a square lattice in terms of corner transfer matrix renormalization group method. J. Phys. Soc. Jpn. **67**(5), 1574–1582 (1998)
25. K. Okunishi, Y. Hieida, Y. Akutsu, Universal asymptotic eigenvalue distribution of density matrices and corner transfer matrices in the thermodynamic limit. Phys. Rev.E **59**(6) (1999)
26. Z.B. Li, Z. Shuai, Q. Wang, H.J. Luo, L. Schülke, Critical exponents of the two-layer Ising model. J. Phys. A Math. Gen. **34**(31), 6069 (2001)
27. A. Gendiar, T. Nishino, Latent heat calculation of the three-dimensional q= 3, 4, and 5 Potts models by the tensor product variational approach. Phys. Rev.E **65**(4), 046702 (2002)
28. R. Orús, G. Vidal, Simulation of two-dimensional quantum systems on an infinite lattice revisited: corner transfer matrix for tensor contraction. Phys. Rev. B **80**, 094403 (2009)
29. G. Vidal, Efficient classical simulation of slightly entangled quantum computations. Phys. Rev. Lett. **91**, 147902 (2003)
30. G. Vidal, Efficient simulation of one-dimensional quantum many-body systems. Phys. Rev. Lett. **93**, 040502 (2004)
31. G. Vidal, Classical simulation of infinite-size quantum lattice systems in one spatial dimension. Phys. Rev. Lett. **98**, 070201 (2007)
32. L. Tagliacozzo, T. de Oliveira, S. Iblisdir, J.I. Latorre, Scaling of entanglement support for matrix product states. Phys. Rev. B **78**, 024410 (2008)
33. F. Pollmann, S. Mukerjee, A.M. Turner, J.E. Moore, Theory of finite-entanglement scaling at one-dimensional quantum critical points. Phys. Rev. Lett. **102**, 255701 (2009)
34. F. Pollmann, J.E. Moore, Entanglement spectra of critical and near-critical systems in one dimension. New J. Phys. **12**(2), 025006 (2010)
35. F. Pollmann, A.M. Turner, Detection of symmetry-protected topological phases in one dimension. Phys. Rev. B **86**(12), 125441 (2012)
36. D. Delande, K. Sacha, M. Płodzień, S.K. Avazbaev, J. Zakrzewski, Many-body Anderson localization in one-dimensional systems. New J. Phys. **15**(4), 045021 (2013)
37. J.H. Bardarson, F. Pollmann, J.E. Moore, Unbounded growth of entanglement in models of many-body localization. Phys. Rev. Lett. **109**(1), 017202 (2012)

38. P. Ponte, Z. Papić, F. Huveneers, D.A. Abanin, Many-body localization in periodically driven systems. Phys. Rev. Lett. **114**(14), 140401 (2015)
39. F. Pollmann, J.E. Moore, Entanglement spectra of critical and near-critical systems in one dimension. New J. Phys. **12**(2), 025006 (2010)
40. B. Pozsgay, M. Mestyán, M.A. Werner, M. Kormos, G. Zaránd, G. Takács, Correlations after Quantum Quenches in the XXZ spin chain: failure of the generalized Gibbs ensemble. Phys. Rev. Lett. **113**(11), 117203 (2014)
41. P. Barmettler, M. Punk, V. Gritsev, E. Demler, E. Altman, Relaxation of antiferromagnetic order in spin-1/2 chains following a quantum quench. Phys. Rev. Lett. **102**(13), 130603 (2009)
42. M. Fagotti, M. Collura, F.H.L. Essler, P. Calabrese, Relaxation after quantum quenches in the spin-1 2 Heisenberg XXZ chain. Phys. Rev. B **89**(12), 125101 (2014)
43. P. Barmettler, M. Punk, V. Gritsev, E. Demler, E. Altman, Quantum quenches in the anisotropic spin-Heisenberg chain: different approaches to many-body dynamics far from equilibrium. New J. Phys. **12**(5), 055017 (2010)
44. F.H.L. Essler, M. Fagotti, Quench dynamics and relaxation in isolated integrable quantum spin chains. J. Stat. Mech. Theory Exp. **2016**(6), 064002 (2016)
45. W. Li, S.J. Ran, S.S. Gong, Y. Zhao, B. Xi, F. Ye, G. Su, Linearized tensor renormalization group algorithm for the calculation of thermodynamic properties of quantum lattice models. Phys. Rev. Lett. **106**, 127202 (2011)
46. D. Pérez-García, F. Verstraete, M.M. Wolf, J.I. Cirac, Matrix Product State Representations. Quantum Inf. Comput. **7**, 401 (2007)
47. R. Orús, G. Vidal, Infinite time-evolving block decimation algorithm beyond unitary evolution. Phys. Rev. B **78**, 155117 (2008)
48. J.I. Cirac, D. Pérez-García, N. Schuch, F. Verstraete, Matrix product density operators: renormalization fixed points and boundary theories. Ann. Phys. **378**, 100–149 (2017)
49. N. Schuch, D. Poilblanc, J.I. Cirac, D. Pérez-García, Topological order in the projected entangled-pair states formalism: transfer operator and boundary Hamiltonians. Phys. Rev. Lett. **111**, 090501 (2013)
50. J.I. Cirac, D. Poilblanc, N. Schuch, F. Verstraete, Entanglement spectrum and boundary theories with projected entangled-pair states. Phys. Rev. B **83**, 245134 (2011)
51. S.-J. Ran, C. Peng, W. Li, M. Lewenstein, G. Su, Criticality in two-dimensional quantum systems: Tensor network approach. Phys. Rev. B **95**, 155114 (2017)
52. S. Yang, L. Lehman, D. Poilblanc, K. Van Acoleyen, F. Verstraete, J.I. Cirac, N. Schuch, Edge theories in projected entangled pair state models. Phys. Rev. Lett. **112**, 036402 (2014)
53. I.P. McCulloch, Infinite size density matrix renormalization group, revisited (2008). arXiv preprint:0804.2509
54. S.R. White, Density matrix formulation for quantum renormalization groups. Phys. Rev. Lett. **69**, 2863 (1992)
55. S.R. White, Density-matrix algorithms for quantum renormalization groups. Phys. Rev. B **48**, 10345–10356 (1993)
56. K.G. Willson, The renormalization group: critical phenomena and the Kondo problem. Rev. Mod. Phys. **47**, 773 (1975)
57. U. Schollwöck, The density-matrix renormalization group in the age of matrix product states. Ann. Phys. **326**, 96–192 (2011)
58. E.M. Stoudenmire, S.R. White, Studying two-dimensional systems with the density matrix renormalization group. Annu. Rev. Condens. Matter Phys. **3**, 111–128 (2012)
59. U. Schollwöck, The density-matrix renormalization group. Rev. Mod. Phys. **77**, 259–315 (2005)
60. G.K.-L. Chan, S. Sharma, The density matrix renormalization group in quantum chemistry. Ann. Rev. Phys. Chem. **62**(1), 465–481 (2011). PMID: 21219144
61. F. Verstraete, D. Porras, J.I. Cirac, Density matrix renormalization group and periodic boundary conditions: a quantum information perspective. Phys. Rev. Lett. **93**, 227205 (2004)
62. P.A.M. Dirac, Note on exchange phenomena in the Thomas atom, in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 26(3), (Cambridge University Press, Cambridge, 1930), pp. 376–385

63. A.K. Kerman, S.E. Koonin, Hamiltonian formulation of time-dependent variational principles for the many-body system. Ann. Phys. **100**(1), 332–358 (1976)
64. R. Jackiw, A. Kerman, Time-dependent variational principle and the effective action. Phys. Lett. A **71**(2), 158–162 (1979)
65. P.W. Langhoff, S.T. Epstein, M. Karplus, Aspects of time-dependent perturbation theory. Rev. Mod. Phys. **44**, 602–644 (1972)
66. J. Haegeman, J.I. Cirac, T.J. Osborne, I. Pižorn, H. Verschelde, F. Verstraete, Time-dependent variational principle for quantum lattices. Phys. Rev. Lett. **107**, 070601 (2011)
67. M.C. Bañuls, M.B. Hastings, F. Verstraete, J.I. Cirac, Matrix product states for dynamical simulation of infinite chains. Phys. Rev. Lett. **102**, 240603 (2009)
68. A. Müller-Hermes, J.I. Cirac, M.-C. Bañuls, Tensor network techniques for the computation of dynamical observables in one-dimensional quantum spin systems. New J. Phys. **14**(7), 075003 (2012)
69. M.B. Hastings, R. Mahajan, Connecting entanglement in time and space: improving the folding algorithm. Phys. Rev. A **91**, 032306 (2015)
70. R.J. Bursill, T. Xiang, G.A. Gehring, The density matrix renormalization group for a quantum spin chain at non-zero temperature. J. Phys. Condens. Matter **8**(40), L583 (1996)
71. X.-Q. Wang, T. Xiang, Transfer-matrix density-matrix renormalization-group theory for thermodynamics of one-dimensional quantum systems. Phys. Rev. B **56**(9), 5061 (1997)
72. N. Shibata, Thermodynamics of the anisotropic Heisenberg chain calculated by the density matrix renormalization group method. J. Phys. Soc. Jpn. **66**(8), 2221–2223 (1997)
73. T. Nishino, Density matrix renormalization group method for 2d classical models. J. Phys. Soc. Jpn. **64**(10), 3598–3601 (1995)
74. E. Tirrito, L. Tagliacozzo, M. Lewenstein, S.-J. Ran, Characterizing the quantum field theory vacuum using temporal matrix product states (2018). arXiv:1810.08050
75. L. Wang, Y.-J. Kao, A.W. Sandvik, Plaquette renormalization scheme for tensor network states. Phys. Rev. E **83**, 056703 (2011)
76. Z.-Y. Xie, J. Chen, M.-P. Qin, J.-W. Zhu, L.-P. Yang, T. Xiang, Coarse-graining renormalization by higher-order singular value decomposition. Phys. Rev. B **86**, 045139 (2012)
77. G. Evenbly, G. Vidal, Tensor network renormalization. Phys. Rev. Lett. **115**, 180405 (2015)
78. G. Evenbly, G. Vidal, Tensor network renormalization yields the multiscale entanglement renormalization ansatz. Phys. Rev. Lett. **115**, 200401 (2015)
79. G. Evenbly, R.N.C. Pfeifer, Improving the efficiency of variational tensor network algorithms. Phys. Rev. B **89**, 245118 (2014)
80. R.N.C. Pfeifer, J. Haegeman, F. Verstraete, Faster identification of optimal contraction sequences for tensor networks. Phys. Rev. E **90**, 033315 (2014)