

Fast convergence of imaginary time evolution tensor network algorithms by recycling the environment

Ho N. Phien,^{1,2} Ian P. McCulloch,² and Guifré Vidal^{2,3}

¹*Centre for Health Technologies, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney 2007, Australia*

²*Centre for Engineered Quantum Systems, School of Mathematics and Physics, University of Queensland, Brisbane 4072, Australia*

³*Perimeter Institute for Theoretical Physics, Waterloo, Ontario, N2L 2Y5, Canada*

(Received 25 November 2014; revised manuscript received 17 February 2015; published 25 March 2015)

We propose an environment recycling scheme to speed up a class of tensor network algorithms that produce an approximation to the ground state of a local Hamiltonian by simulating an evolution in imaginary time. Specifically, we consider the time-evolving block decimation (TEBD) algorithm applied to infinite systems in 1D and 2D, where the ground state is encoded, respectively, in a matrix product state (MPS) and in a projected entangled-pair state (PEPS). An important ingredient of the TEBD algorithm (and a main computational bottleneck, especially with PEPS in 2D) is the computation of the so-called *environment*, which is used to determine how to optimally truncate the bond indices of the tensor network so that their dimension is kept constant. In current algorithms, the environment is computed at each step of the imaginary time evolution, to account for the changes that the time evolution introduces in the many-body state represented by the tensor network. Our key insight is that close to convergence, most of the changes in the environment are due to a change in the choice of gauge in the bond indices of the tensor network, and not in the many-body state. Indeed, a consistent choice of gauge in the bond indices confirms that the environment is essentially the same over many time steps and can thus be re-used, leading to very substantial computational savings. We demonstrate the resulting approach in 1D and 2D by computing the ground state of the quantum Ising model in a transverse magnetic field.

DOI: [10.1103/PhysRevB.91.115137](https://doi.org/10.1103/PhysRevB.91.115137)

PACS number(s): 03.67.-a, 03.65.Ud, 02.70.-c, 05.30.Fk

I. INTRODUCTION

The study of interacting quantum many-body systems remains a central problem in modern physics. Understanding how the microscopic degrees of freedom organize themselves collectively is key to explaining the large variety of complex, emergent phenomena exhibited by quantum many-body systems. The study of such systems, for instance on a lattice, faces a major computational challenge, given the exponential growth of the Hilbert space dimension with the lattice size. There are several numerical approaches available, including exact diagonalization of the Hamiltonian on small lattices or quantum Monte Carlo, based on sampling an imaginary time evolution in Hamiltonians that are free from the sign problem. Among these and other methods, tensor network approaches stand up for their ability to approximating the full ground-state wave function, which is efficiently stored in a tensor network state.

The matrix product states (MPS) [1,2] is the best known example of a tensor network state. It is the basis of the density matrix renormalization group (DMRG) method [3,4] to efficiently represent the ground state of a 1D quantum system in both finite and infinite [5–11] lattices. The MPS is also the basis of methods to simulate time evolution, including the time evolving block decimation (TEBD) algorithm [12,13] and its DMRG-like formulation [14,15], which have been recently applied to study dynamical properties of 1D quantum systems [16–18], as well as the more recent algorithms based on the time-dependent variational principle [19]. Here, we will focus on the TEBD algorithm, originally proposed for finite systems and subsequently extended to address systems in the thermodynamic limit, where it is referred to as the *infinite* TEBD algorithm (iTEBD) [20,21]. In the thermodynamic

limit, the many-body state is represented by an *infinite* MPS (iMPS), which consists of a finite unit cell of MPS tensors that is repeated throughout the infinite chain. On the other hand, the projected entangled-pair state (PEPS) [22–26] or tensor product state (TPS) [27–29] was introduced as a natural generalization of the MPS for 2D systems. In the thermodynamic limit, one can again use an infinite PEPS (iPEPS) [30,31], which has a finite unit cell of tensors repeated throughout the infinite 2D lattice.

In both 1D and 2D, there are two main strategies to find a good approximation to the ground state of a local Hamiltonian H : one can either minimize the expectation value of H , succinctly referred to as *energy minimization*, or one can simulate the action, on some initial state, of the evolution operator e^{-tH} in imaginary time in the long-time limit, referred to as *imaginary time evolution*. In a 1D lattice, energy minimization (e.g., through the DMRG algorithm) is preferred over imaginary time evolution, because it generally converges faster to the ground state. In contrast, in 2D systems, most current implementations are based on imaginary time evolution using the TEBD algorithm supplemented with a large variety of algorithms to contract the tensor network, including schemes based on a boundary MPS [30], the *corner transfer matrix* [31–36], and the *tensor renormalization group* and its various important generalizations [37–40]. Among the most exciting explorations of ground states in challenging 2D systems by means of imaginary time evolution on iPEPS one can find, e.g., recent calculations of the t - J model on the square lattice [41,42] and honeycomb lattice [43], and of the $J_1 - J_2$ frustrated Heisenberg model on the square lattice [44].

In order to simulate the action of the imaginary time evolution operator e^{-tH} on some initial state, the TEBD algorithm proceeds by breaking this evolution operator into the

product of a large number of operators $e^{-\delta H}$, each representing a small time step, $\delta \ll 1$, and by then applying each of these operator $e^{-\delta H}$ successively. Each application of a small time step $e^{-\delta H}$ results in the growth of the bond dimension of the tensors in the tensor network, which in turn results in a significant increase in computational costs. In order to keep the computational cost in check, it is thus important to truncate the bond index back to its original size. An optimal truncation of the bond index of a tensor (that is, a truncation that minimizes the error introduced in the many-body state), requires the computation of the so-called *environment*, which contains information on how the truncation affects the rest of the many-body state.

In a 1D system with open boundary conditions (OBC), the cost of computing the environment is similar to the cost of applying the time evolution $e^{-\delta H}$. However, in 2D, the cost of computing the environment is much larger than the cost of applying $e^{-\delta H}$, and in fact computing the environment is the major bottleneck of the whole TEBD algorithm. So much so, that it has become a standard practice to bypass the computation of the environment altogether, and use instead just local information as a guide to perform the truncation of the bond indices, in what is known as the *simplified update* (SU) [45,46]. In spite of its wide use within the PEPS community, the SU is not well-justified, and it is known to sometimes lead to poor PEPS approximations of the ground state. In other words, the SU, with its lower computational cost, allows to use a PEPS with larger bond dimension; however, it does not exploit this larger bond dimension in an optimal way, potentially producing worse approximations to the ground state than what one would have obtained with a properly optimized PEPS with smaller bond dimension. Therefore the so-called *full update* (FU), which requires the computation of the environment, is still the most reliable option in spite of its prohibitive cost.

In this paper, we thus focus on simulating imaginary time evolution using TEBD with FU, and propose a strategy to reduce its computational cost. Our strategy aims mainly at speeding up PEPS computations of 2D ground states, but it is also be useful in MPS calculations in 1D systems with periodic boundary conditions (PBC) as opposed to the OBC case alluded to above, where computing the environment is also significantly more expensive than just applying $e^{-\delta H}$. The key observation is that while at the beginning of the imaginary time evolution the many-body state is changing rapidly, in later iterations of the TEBD algorithm the changes are much less significant. This may not be directly apparent in the environment, which might still be subject to big changes. However, these big changes correspond mostly to a random choice in the gauge freedom existing in the tensor network. Specifically, on each bond index connecting two tensors of the MPS or PEPS, one always introduce the product of an invertible matrix and its inverse without affecting the represented many-body state. Moreover, it is important to remember that the environment is only used to determine the right choice of truncation of the bond dimension, and that a small change in the environment will not change this choice significantly. Therefore, if we manage to fix the gauge freedom in the environment so that an equivalent gauge is chosen over several time steps, then we may be able to compute the

environment only *once* and then recycle or reuse it to guide the truncation during *many* time steps. As we will see, this strategy actually leads to very significant computational savings.

In order to be able to recycle the environment, we first need to bring the tensor network into a canonical form that fixes the gauge degrees of freedom in the bond indices. For an MPS with OBC, there is already a natural canonical form as well as a well-understood procedure to obtain it. For an MPS with PBC and a PEPS, there is no obvious choice of a canonical form and it is not *a priori* clear that a canonical form even exists. Here we propose an iterative procedure that, when applied to an MPS with OBC, has the usual canonical form as its fixed point. The same iterative procedure can then be applied to an MPS with PBC and (after suitable generalization) to a PEPS, and we verify numerically that a unique fixed-point is reached in each case. We then define the canonical form of an MPS with PBC and of a PEPS to be this fixed-point. For the purpose of incorporating the environment recycling schemes into the TEBD algorithm, we introduce an iterative method to obtain the canonical form of an MPS and a PEPS [47]. We notice that this type of canonical form of a PEPS is similar to the “quasicanonical” form used in Refs. [48,49] to study the ground state of a 2D system by means of the SU.

For concreteness, in this work, we discuss the recycling of the environment in a ground state computation by imaginary time evolution in an infinite system both in 1D and 2D, using iMPS and iPEPS, respectively. However, we emphasize that environment recycling can also be used in a more broader class of algorithms. On the one hand, one can recycle the environment when simulating imaginary time evolution on finite systems, even though such algorithms are not currently very popular. On the other hand, the very same ideas can also be applied to the computation of a classical partition function in 2D and 3D, using a scheme where a tensor network (e.g., an iMPS and an iPEPS, respectively) is used to encode the dominant eigenvector of a (1D or 2D, respectively) transfer matrix by a power method, where now this transfer matrix replaces the imaginary time evolution operator $e^{-\delta H}$. Indeed, after applying this transfer matrix a few times, the tensor network starts to approach the dominant eigenvector; from that moment on, the corresponding environment will no longer change significantly over iterations and it can be recycled.

Let us next summarize the structure of the paper. In Sec. II, we briefly introduce MPS and PEPS. In Sec. III, we revise the canonical form of an iMPS and introduce an alternative iterative method to obtain it. We also provide methods to fix the remaining gauge degrees of freedom (mostly, complex phases) within a canonical iMPS. Then in Sec. IV we propose a canonical form for iPEPS and a scheme to fix all its gauge degrees of freedom. In Sec. V, we present an environment recycling scheme for TEBD on iMPS and some benchmark results for 1D quantum Ising model with transverse magnetic field. Next, in Sec. VI, the environment recycling scheme is developed for TEBD on iPEPS and applied to study 2D quantum Ising model with the transverse magnetic field. Finally, Sec. VII contains our conclusions.

In this paper, we use the terms MPS and PEPS to denote tensor networks in 1D and 2D, respectively, and the term TEBD to denote an algorithm that simulates a time evolution (in imaginary time) according to e^{-tH} in order to obtain an

MPS/PEPS approximation to the ground state of H . In the literature one often finds that the term “iPEPS algorithm” is used to denote what we here call instead the TEBD algorithm for iPEPS. In this paper, we find it important to distinguish between the tensor network being used (e.g., MPS, PEPS, or even iMPS, iPEPS) and the algorithm that is employed to optimize it (DMRG, TEBD, etc.).

II. TENSOR NETWORK STATES

In this section, we briefly revise the matrix product state and the projected entangled-pair state employed to represent the many-body state of 1D and 2D quantum lattice systems respectively.

A. Matrix product states

Let us consider a 1D quantum lattice system consisting of N sites. Each lattice site has an internal degree of freedom, for example, a spin, and is represented by a local d -dimensional Hilbert space $\mathcal{H}_d = \mathbb{C}^d$. The N -site lattice is then described in the Hilbert space $\mathcal{H} = (\mathcal{H}_d)^{\otimes N}$. A general pure state of the system can be written in the local basis as

$$|\Psi\rangle = \sum_{s_1 s_2 \dots s_N=1}^d c_{s_1 s_2 \dots s_N} |s_1, s_2, \dots, s_N\rangle, \quad (1)$$

where $|s_1, s_2, \dots, s_N\rangle = |s_1\rangle \otimes |s_2\rangle \otimes \dots \otimes |s_N\rangle$, and $|s_i\rangle$ (for $i = 1, \dots, N$ and $s_i = 1, \dots, d$) is the orthonormal basis of the local d -dimensional Hilbert space at site i , and $c_{s_1 s_2 \dots s_N}$ is a complex component of a rank- N tensor c . The total number of components $c_{s_1 s_2 \dots s_N}$ is d^N , and hence proliferates exponentially with the number of lattice sites.

We can decompose the rank- N tensor c into a product of lower-rank tensors without changing the state of the system. More precisely, if the lattice has OBC, the pure state in Eq. (1) can be rewritten as follows:

$$|\Psi\rangle = \sum_{s_1 s_2 \dots s_N=1}^d A^{s_1} A^{s_2} \dots A^{s_N} |s_1, s_2, \dots, s_N\rangle, \quad (2)$$

where A^{s_1} and A^{s_N} corresponding to each value of s_1 and s_N are the $1 \times \chi$ row and $\chi \times 1$ column vectors, respectively, and each A^{s_k} (for $k = 2, \dots, N-1$) corresponding to each value of s_k is a $\chi \times \chi$ matrix. The pure state represented by Eq. (2) is called as a *matrix product state*; see Fig. 1(a). This MPS is parameterized by $(N-2)\chi^2 d + 2\chi d$ variational

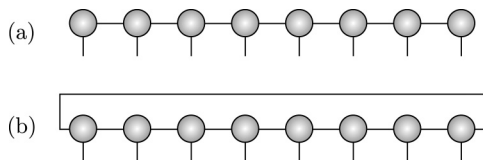


FIG. 1. Diagrammatic representations of an MPS corresponding to different boundary conditions: (a) The MPS with OBC. (b) The MPS with PBC. In these representations, a tensor is represented by a circle which has *legs* corresponding to the indices of the tensor. The links connecting tensors together represent the contraction over bond indices. The open legs of the tensors correspond to the physical indices, which are not shared between tensors in the network.

parameters, and accordingly the state is represented by a number of parameters, which increases polynomially with N . The coefficient $\chi \geq 1$ is defined as the *bond dimension* and understood as a *refinement* parameter. More precisely, χ plays an important role in qualifying the MPS representation as the larger the χ , the better the MPS representation. If $\chi = 1$, the MPS represents a product state otherwise the MPS describes an entangled state. We can always choose an upper bound of χ to represent exactly the general state of N -site lattice, say $\chi = d^{N/2}$, but this will result in poor use of computational resources because the MPS is no longer an efficient representation.

In case that the lattice obeys PBC, the MPS expressed by Eq. (2) can still represent correctly the state of the system. This is straightforward because the tensor c decomposed into a product of tensors $\{A^{s_k}\}_{k=1}^N$ does not impose any specific boundary condition on the lattice. However, it is not a feasible representation in the sense that there is no connection between the tensors representing the first and the last sites of the lattice. Therefore the correlation between these two sites is not transmitted through the link of the system. Alternatively, we can generalize this representation where all the tensors including the first and the last ones have two bond indices. The generalized MPS is then written as follows:

$$|\Psi\rangle = \sum_{s_1 s_2 \dots s_N=1}^d \text{Tr}(A^{s_1} A^{s_2} \dots A^{s_N}) |s_1, s_2, \dots, s_N\rangle. \quad (3)$$

Note that the *trace* (Tr) is taken to ensure that after contracting all the matrices $\{A^{s_k}\}_{k=1}^N$ we have a scalar quantity. A diagrammatic representation of the MPS with PBC is shown in Fig. 1(b), where the trace is taken by connecting the left bond of tensor A^{s_1} with the right bond of tensor A^{s_N} .

In reality, the representation of an MPS with PBC often causes some computational difficulties in simulation. More explicitly, a simulation using MPS with PBC is much more computationally expensive than employing MPS with OBC. As the cyclic structure of the MPS with PBC, it requires larger computational resources to capture the amount of entanglement in the system. The computational cost scales as $\mathcal{O}(\chi^5)$ [50] comparing to $\mathcal{O}(\chi^3)$ when simulating using MPS with OBC. However, this scaling sometimes can be reduced to $\mathcal{O}(\chi^3)$ [51,52].

Note that MPS can also represent the state of an infinite-size homogeneous system using only a small number of parameters by employing its invariance under translations. This results in what is called *infinite* MPS (iMPS). An iMPS is normally constructed by repeatedly assigning a unit cell made up of a finite set of tensors along the lattice chain. For example, a one-site iMPS is represented as follows:

$$|\Psi\rangle = \sum_{\dots s \dots=1}^d \dots \lambda \Gamma^s \dots | \dots, s, \dots \rangle. \quad (4)$$

In this iMPS, the unit cell contains a pair of tensors $\{\Gamma, \lambda\}$ where the tensor Γ is positioned at each lattice site and the diagonal matrix λ lies on the link between each two adjacent sites. The iMPS is represented by only $\chi^2 d + \chi$ variational parameters. This is the most striking advantage of using translational invariant iMPS because we only need a small

number of parameters to represent the ground state of a lattice system in the thermodynamic limit. Consequently, using an iMPS to represent a state when simulating a 1D infinite system obviously reduces the computational complexity as well as the computer memory.

B. Projected entangled-pair states

Consider a 2D quantum system on a lattice of N sites where each lattice site is represented by a local Hilbert space $\mathcal{H}_d \cong \mathbb{C}^d$. To represent the system in terms of a TN, one option is to use the so-called *projected entangled-pair state* [22–24], which is generally defined as

$$|\Psi\rangle = \sum_{s_{\vec{r}_1}, \dots, s_{\vec{r}_N}=1}^d F(A^{[\vec{r}_1]s_{\vec{r}_1}}, \dots, A^{[\vec{r}_N]s_{\vec{r}_N}}) |s_{\vec{r}_1}, \dots, s_{\vec{r}_N}\rangle. \quad (5)$$

This PEPS is represented by tensors $\{A^{[\vec{r}_i]s_{\vec{r}_i}}\}_{i=1}^N$ (where $\vec{r}_i = (x_i, y_i)$ is the coordinate of the lattice site i), which are connected to some set of neighbor sites according to the geometry of the lattice. Each tensor of the PEPS has n (the coordination number, or the number of nearest neighbors of one lattice site) bond indices of dimension D and one physical index of dimension d . The choice of n in the TN depends on the geometry of the lattice, and can in principle be chosen arbitrarily (although this affects the computational complexity of contracting the TN). The function F contracts all the tensors $\{A^{[\vec{r}_i]s_{\vec{r}_i}}\}_{i=1}^N$ according to this pattern and then performs the trace to obtain a scalar quantity. In Fig. 2, we show diagrammatically several PEPSs for the systems corresponding to different geometries with OBC. In case the geometry of a system is a square lattice pattern with PBC ($n = 4$), the PEPS consists of tensors that have four bond indices and one physical index. Overall, the PEPS depends on $\mathcal{O}(ND^4d)$ variational parameters. Note that the bond dimension D determines the quality of PEPS, and if D is chosen to be large enough, any state of a 2D lattice system can be well represented by a PEPS.

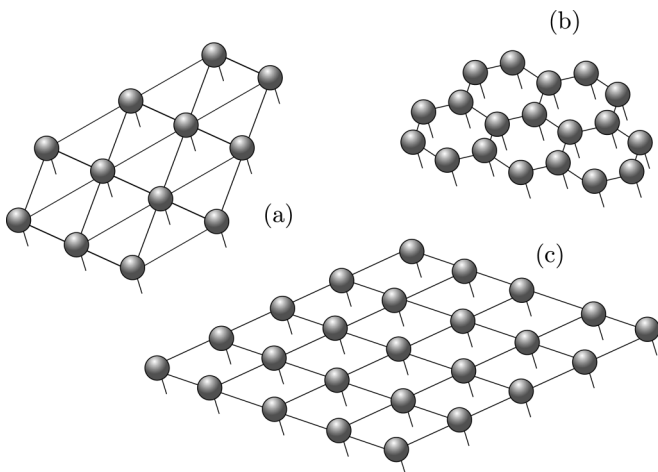


FIG. 2. Diagrammatic representations of PEPSs that correspond to different lattice patterns: (a) a triangular lattice, (b) a hexagonal lattice, and (c) a square lattice.

As a generalization of the MPS representation to 2D, PEPS is a good representation for a class of variational wave functions when studying the ground state of 2D quantum lattice systems. The rationale behind this is that a PEPS is constructed naturally to capture the geometry of the lattice pattern, such that the scaling of entanglement entropy of a subregion is proportional to the surface area of the subregion, it fulfills the *area law* [53–55]. More precisely, the entanglement entropy of an $l \times l$ block scales as the number of bonds that are cut in the TN, which is $\mathcal{O}(l)$ [56]. Thus a PEPS can represent efficiently the ground state of a 2D quantum lattice system with a small number of parameters, which scales at worst polynomially with the system size. Furthermore, it can also effectively capture the quantum correlations existing in the system. There is some prospect of capturing also dynamical properties, as long as the dynamical growth in entanglement entropy is not too large. Besides, as similar to MPS, we can also employ an *infinite* PEPS (iPEPS) to represent a state of a 2D homogeneous system, which is parameterized by a fairly small number of parameters.

III. CANONICAL FORM OF AN IMPS AND GAUGE DEGREES OF FREEDOM FIXING

In this section, we revise the canonical form of an iMPS and introduce an iterative method to obtain it. This method as we will see can be generalized to apply to MPS with either OBC or PBC. Besides, some methods applied to fix the gauge degrees of freedom in a canonical iMPS will also be introduced explicitly.

A. Canonical form of an iMPS

For simplicity, let us consider an infinite lattice chain represented by a one-site translationally invariant iMPS in Eq. (4). This iMPS is said to be in its canonical form if λ contains the Schmidt coefficients and therefore we can always write it in the Schmidt decomposition at any bond of the chain as follows:

$$|\Psi\rangle = \sum_{\alpha=1}^{\chi} \lambda_{\alpha} |\phi^{\alpha}\rangle_L |\phi^{\alpha}\rangle_R, \quad (6)$$

with $|\phi^{\alpha}\rangle_L$ and $|\phi^{\alpha}\rangle_R$ being orthonormal bases. Equivalently, this iMPS must satisfy the orthonormality constraints defined as

$$\sum_s \Gamma^{s\dagger} \rho_L \Gamma^s = \mathbb{I}, \quad (7)$$

$$\sum_s \Gamma^s \rho_R \Gamma^{s\dagger} = \mathbb{I}, \quad (8)$$

where $\rho_L = \rho_R = \lambda^2$ and \mathbb{I} are reduced density and identity matrices respectively.

To transfer an arbitrary iMPS represented by Eq. (4) into a canonical form, one can apply the scheme introduced in Ref. [21]. Here, as an alternative, we propose an iterative method that can also handle well this task. The key idea of this iterative method is quite similar to the one in Ref. [21]. The main difference is that we do not need the left- and

right-dominant eigenvectors of a transfer matrix to find the resolutions but instead we will find the resolutions locally.

More concretely, our scheme consists of the following fundamental steps:

(i) Compute the matrices V_L and V_R whose coefficients are defined as follows:

$$V_L = \sum_s (\lambda \Gamma^s)^\dagger (\lambda \Gamma^s), \quad (9)$$

$$V_R = \sum_s (\Gamma^s \lambda) (\Gamma^s \lambda)^\dagger, \quad (10)$$

which are shown graphically in Fig. 3(i). As V_L and V_R are Hermitian and positive matrices, we can decompose $V_L = Y^\dagger Y$ and $V_R = X X^\dagger$ by means of the eigenvalue decomposition. For instance, we can decompose $V_L = W S W^\dagger$ and then assign $Y^\dagger = W \sqrt{S}$ and $Y = \sqrt{S} W^\dagger$. Similarly, we can easily obtain $X = W' \sqrt{S'}$ and $X^\dagger = \sqrt{S'} W'^\dagger$ from the decomposition $V_R = W' S' W'^\dagger$.

(ii) Insert two resolutions $\mathbb{I} = (Y^T)^{-1} Y^T$ and $\mathbb{I} = X X^{-1}$ into the bonds of the iMPS as illustrated in Fig. 3(ii). Then we get a new $\tilde{\lambda}$ which contain the singular values from taking the SVD of $Y^T \lambda X$ such that $U_1 \tilde{\lambda} V_1 = Y^T \lambda X$.

(iii) A new tensor $\tilde{\Gamma}$ is obtained by contracting all the tensors including $V_L, X^{-1}, \Gamma, (Y^T)^{-1}$ and U_1 together; see Fig. 3(iii). Reassign $\Gamma \equiv \tilde{\Gamma}$ and $\lambda \equiv \tilde{\lambda}$ then go back to step (i).

The above steps are repeated until the spectrum consisting of singular values $\tilde{\lambda}$ converge to a fixed point. Then the iMPS represented by $\{\tilde{\Gamma}, \tilde{\lambda}\}$ fulfills the conditions in Eqs. (7) and (8) and, thus, is in the canonical form.

We have empirically observed that the above algorithm always converged to the canonical form. Note also that because the canonical form of an iMPS is obtained locally without taking into account the information of the whole system, we can also apply this scheme to find the canonical form for MPS with PBC when it is translationally invariant under shifts of a

certain unit cell of sites. The canonical form of the MPS with PBC seems to be closely related to the one defined in Ref. [7] for the case of translational invariant state. However, we are unable to confirm that they are exactly the same since in our case it can be also defined for the nontranslational invariant state as well.

B. Fixing gauge degrees of freedom in an iMPS

We now introduce two possible methods to fix the gauge degrees of freedom between two different canonical iMPSs that represent the same physical state $|\Psi\rangle$. Suppose that the canonical iMPSs are one-site translationally invariant and denoted as $|\Psi_A\rangle$ and $|\Psi_B\rangle$. In this notation, $|\Psi_A\rangle$ and $|\Psi_B\rangle$ are characterized by single tensors A and B , respectively. For simplicity, let us assume that $|\Psi_A\rangle$ and $|\Psi_B\rangle$ are normalized. Due to the canonical form representations in these iMPSs, we come up with the relation $A = U^\dagger B U$, where U can be a unitary matrix or a diagonal matrix consisting of complex phases. To fix the gauge degrees of freedom between $|\Psi_A\rangle$ and $|\Psi_B\rangle$, we need to search for the matrix U . In the following, with the assumption that U is unitary, we will explain in details how to obtain U with two different methods.

1. Direct method

Suppose that X is the left-dominant eigenvector of the transfer matrix created by two tensors A and B with corresponding left-dominant eigenvalue μ . We have the following equation:

$$\sum_{s=1}^d \sum_{\alpha', \alpha=1}^X (B_{\beta' \alpha'}^s)^* X_{\alpha' \alpha} A_{\alpha \beta}^s = \mu X_{\beta' \beta}. \quad (11)$$

Note that for a clear demonstration, we will incorporate graphical representations for some expressions in the text. For example, Eq. (11) is shown graphically as

Let us replace $A_{\alpha\beta}^s = \sum_{\gamma, \delta=1}^X U_{\alpha\gamma}^* B_{\gamma\delta}^s U_{\delta\beta}$ into the above equation, we have

Multiplying both sides of Eq. (12) with U^\dagger yields

which is simplified as

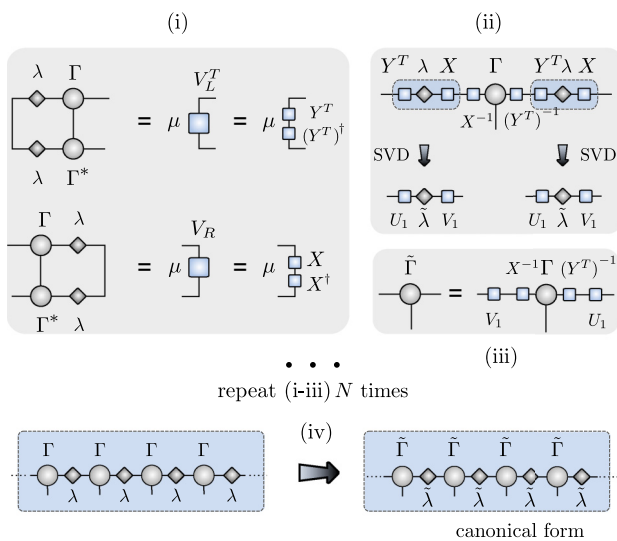


FIG. 3. (Color online) (i)–(iii) Fundamental steps applied to transfer an iMPS $\{\Gamma, \lambda\}$ into its canonical form. These steps are iterated until the Schmidt coefficients converge. (iv) Two iMPSs $\{\Gamma, \lambda\}$ and $\{\tilde{\Gamma}, \tilde{\lambda}\}$ are different from each other but they represent the same physical state $|\Psi\rangle$.

We realize that XU^\dagger is also an eigenvector of the transfer matrix created by tensors B and B^\dagger with corresponding eigenvalue μ . However, as $|\Psi_B\rangle$ is defined in its canonical form, a possible solution of Eq. (14) is $XU^\dagger = \mathbb{I}$. Thus we can easily extract $U = X$ and $\mu = 1$. The matrix U needed for fixing the gauge degrees of freedom between the two iMPSs $|\Psi_A\rangle$ and $|\Psi_B\rangle$ has been found exactly and this corresponds to the dominant eigenvector of the transfer matrix created by tensors A and B .

2. Iterative method

We can also determine the matrix U iteratively. To do that, let us define a *local fidelity* as follows:

$$\begin{aligned} F &= \sum_{s=1}^d \sum_{\alpha,\beta,\gamma,\alpha'=1}^{\chi} U_{\alpha\beta} A_{\beta\gamma}^s (U_{\gamma\alpha'})^* (B_{\alpha'\alpha}^s)^*, \\ &= \text{Tr}(U A U^\dagger B^\dagger) \\ &= \text{Tr}(U M). \end{aligned} \quad (15)$$

This equation is visualized as

$$F = \begin{array}{c} \text{---} U \text{---} A \text{---} U^\dagger \text{---} \\ | \quad | \quad | \\ \text{---} B^* \text{---} \end{array} = \begin{array}{c} \text{---} U \text{---} A \text{---} U^\dagger \text{---} \\ | \quad | \quad | \\ \text{---} B^* \text{---} \end{array} = \begin{array}{c} \text{---} U \text{---} \\ | \\ \text{---} M \text{---} \end{array}$$

where we have defined $M = AU^\dagger B^\dagger$. Our task is to find the matrix U that maximizes the fidelity F . More specifically, we first initialize an arbitrary U , e.g., $U = \mathbb{I}$ and then iterate the following steps: (i) apply the SVD to decompose the matrix M such that $M = W S V^\dagger$; (ii) assign $U = V W^\dagger$ and substitute it back into Eq. (15); (iii) compute the fidelity, which is now defined as $F = \text{Tr}(S)$; (iv) recompute the matrix $M = AU^\dagger B^\dagger$ and then go back to step (i). The above steps are iterated until the fidelity converges to a fixed point, i.e., $F = 1$.

IV. CANONICAL FORM OF AN IPEPS AND GAUGE DEGREES OF FREEDOM FIXING

Similar to an iMPS, an iPEPS representing a pure state $|\Psi\rangle$ of a 2D lattice system is not unique, and there exists a lot of different iPEPSs representing the same state physical $|\Psi\rangle$. These iPEPSs can be distinguished by some gauge degrees of freedom. For instance, for a one-site translationally invariant iPEPS $|\Psi_A\rangle$, characterized by tensor A , one can always insert $M^{-1}M = \mathbb{I}$ and $K^{-1}K = \mathbb{I}$ (M and K are arbitrary invertible matrices) into *horizontal* and *vertical* links of the iPEPS $|\Psi_A\rangle$, respectively, without changing its physical properties. More concretely, we define a new iPEPS $|\Psi_B\rangle$ characterized by tensor B given by

$$B_{lrdu}^s = \sum_{l'r'd'u'} M_{ll'} K_{dd'} A_{l'r'd'u'}^s M_{r'r'}^{-1} K_{u'u'}^{-1}, \quad (16)$$

which describes a same physical state as $|\Psi_A\rangle$. The existence of gauge degrees of freedom sometimes causes lots of numerical difficulties in dealing with iPEPS representation. Therefore, we need to fix the gauge degrees of freedom in every iPEPS. This can be done by defining a new type of *canonical form* for an iPEPS in what follows.

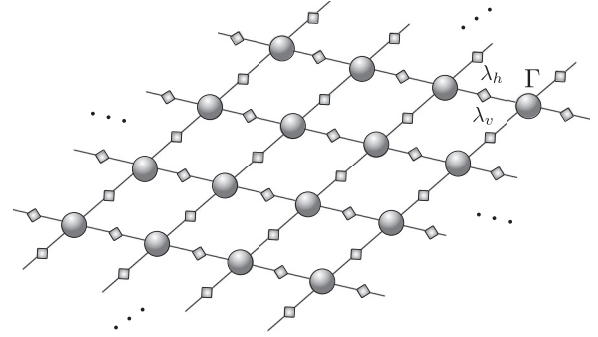


FIG. 4. Diagrammatic representation of an infinite one-site translationally invariant iPEPS, which consists of tensors $\{\lambda_h, \lambda_v, \Gamma\}$.

A. Canonical form of an iPEPS

For the sake of simplicity, we consider a one-site translationally invariant iPEPS, which is represented by tensors $\{\Gamma, \lambda_h, \lambda_v\}$ and is visualised as in Fig. 4. We can also represent the iPEPS with a single tensor A by contracting the matrices λ_h and λ_v to tensor Γ such that $A = \lambda_h \lambda_v \Gamma$.

Note that for a TN with closed loops such as an iPEPS, it is impossible to define a canonical form based on the orthonormality of a bipartite Schmidt decomposition. However, in what follows, we define a new type of canonical form for an iPEPS in the sense that the tensors in the unit cell of the iPEPS satisfy some particular constraints. This definition for a canonical form iPEPS is completely different from the one given in Ref. [57] where the canonical form of a *finite* PEPS is termed from characterizing the existence of symmetries given by the fact that two representations of the same *injective* PEPS are related by unique invertible matrices. In what follows, we define the canonical form based on the orthogonality constraints of local tensors of the iPEPS.

Definition. An iPEPS is defined in its canonical form if it is represented by the tensors $\{\Gamma, \lambda_h, \lambda_v\}$ satisfying the following constraints:

$$\sum_{s,l,d,u} L_{lrdu}^s (L_{l'r'd'u'}^s)^* = \delta_{rr'}, \quad (17)$$

$$\sum_{s,r,d,u} R_{lrdu}^s (R_{l'r'd'u'}^s)^* = \delta_{ll'}, \quad (18)$$

$$\sum_{s,l,r,d} \mathcal{D}_{lrdu}^s (\mathcal{D}_{l'r'd'u'}^s)^* = \delta_{uu'}, \quad (19)$$

$$\sum_{s,l,r,u} U_{lrdu}^s (U_{l'r'd'u'}^s)^* = \delta_{dd'}, \quad (20)$$

where we have defined

$$L_{lrdu}^s = \sum_{l',d',u'} \Gamma_{l'r'd'u'}^s (\lambda_h)_{ll'} (\lambda_v)_{dd'} (\lambda_v)_{u'u'}, \quad (21)$$

$$R_{lrdu}^s = \sum_{r',d',u'} \Gamma_{l'r'd'u'}^s (\lambda_h)_{r'r'} (\lambda_v)_{dd'} (\lambda_v)_{u'u'}, \quad (22)$$

$$\mathcal{D}_{lrdu}^s = \sum_{l',r',d'} \Gamma_{l'r'd'u'}^s (\lambda_h)_{ll'} (\lambda_v)_{dd'} (\lambda_h)_{r'r'}, \quad (23)$$

$$U_{lrdu}^s = \sum_{l',r',u'} \Gamma_{l'r'd'u'}^s (\lambda_h)_{ll'} (\lambda_v)_{u'u'} (\lambda_h)_{r'r'}. \quad (24)$$

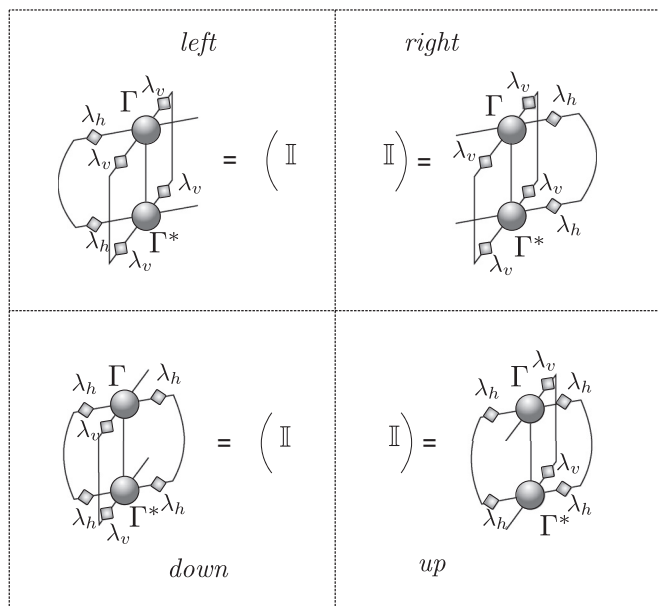


FIG. 5. Diagrammatic representation of constraints for the canonical iPEPS.

The conditions defined by Eqs. (17)–(20) can be named as the *left*, *right*, *down*, and *up* canonical constraints for the canonical form of the iPEPS and are shown diagrammatically in Fig. 5.

By generalizing the iterative method applied to transfer an iMPS into its canonical form, we can obtain the canonical form for an arbitrary iPEPS. In particular, we iterate the following steps, which are also illustrated diagrammatically in Fig. 6.

(i) Compute matrices V_L , V_R , V_D , and V_U to find the necessary resolutions; see Fig. 6(i). As all of these matrices are Hermitian and positive matrices, we can decompose $V_L = Y_h^\dagger Y_h$, $V_R = X_h X_h^\dagger$, $V_D = Y_v^\dagger Y_v$, $V_U = X_v X_v^\dagger$ by employing, for example, the eigenvalue decomposition. Specifically, we can construct $V_L = W_h D_h W_h^\dagger$, $V_R = Q_h T_h Q_h^\dagger$, $V_D = W_v D_v W_v^\dagger$ and $V_U = Q_v T_v Q_v^\dagger$ then assign $Y_h = \sqrt{D_h} W_h^\dagger$, $X_h = Q_h \sqrt{T_h}$, $Y_v = \sqrt{D_v} W_v^\dagger$, and $X_v = Q_v \sqrt{T_v}$, respectively.

(ii) Insert simultaneously four resolutions $(Y_h^T)^{-1} Y_h^T$, $X_h X_h^{-1}$, $(Y_v^T)^{-1} Y_v^T$, and $X_v X_v^{-1}$, which are all identity matrices, into the bonds of the iPEPS as illustrated in Fig. 6(ii). Two new diagonal matrices $\tilde{\lambda}_{1h}$ and $\tilde{\lambda}_{1v}$ are then obtained by taking the SVDs of $Y_h^T \lambda_h X_h$ and $Y_v^T \lambda_v X_v$, respectively.

(iii) Define a new tensor $\tilde{\Gamma}$ which is obtained by contracting all the remaining tensors V_h , X_h^{-1} , V_v , X_v^{-1} , Γ , $(Y_h^T)^{-1}$, U_h , $(Y_v^T)^{-1}$, and U_v together; see Fig. 6(iii). Reassign $\tilde{\Gamma} \equiv \Gamma$, $\tilde{\lambda}_h \equiv \tilde{\lambda}_{1h}$ and $\tilde{\lambda}_v \equiv \tilde{\lambda}_{1v}$ then go back to step (i).

The above steps are repeated until the fixed point of the spectrum described by diagonal tensors $\tilde{\lambda}_h, \tilde{\lambda}_v$ is reached. Then the iPEPS represented by tensors $\{\tilde{\Gamma}, \tilde{\lambda}_h, \tilde{\lambda}_v\}$ is in its canonical form and satisfies the conditions defined in Eqs. (17)–(20). In practice, we empirically see that the iteration for obtaining the canonical form always converges.

B. Fixing gauge degrees of freedom in an iPEPS

One of the most significant benefits of working with the canonical form of an iPEPS is that we can fix its gauge degrees

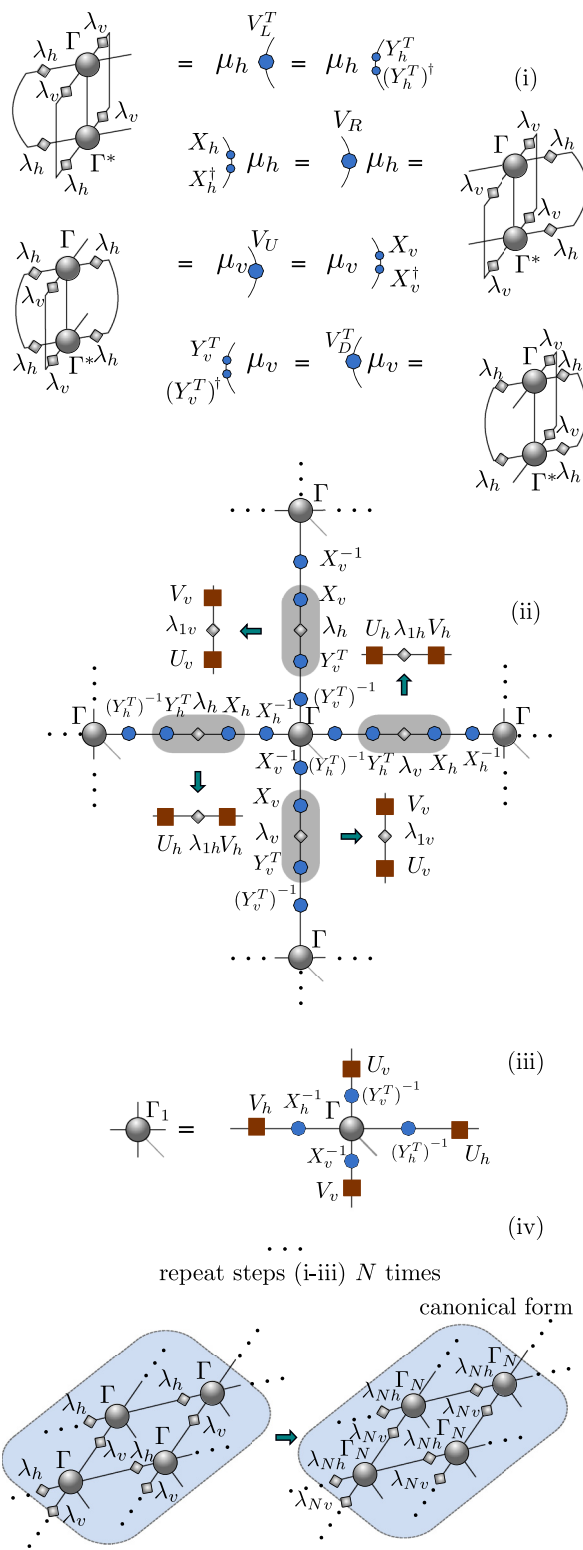


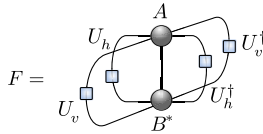
FIG. 6. (Color online) Diagrammatic illustration of how to transfer an iPEPS into its canonical form.

of freedom that can reduce many numerical difficulties. Even if an iPEPS is in the canonical form, there is still a gauge freedom to choose the complex phases of each index of the tensors. Here, by assuming that two different iPEPSs describing the same quantum state are in the canonical forms, we suggest an

iterative method to fix the gauge degrees of freedom between them. In particular, let us call these two iPEPSs as $|\Psi_A\rangle$ and $|\Psi_B\rangle$, which are characterized by tensors $\{\Gamma_A, \lambda_{Ah}, \lambda_{Av}\}$ and $\{\Gamma_B, \lambda_{Bh}, \lambda_{Bv}\}$, respectively. We define two new tensors $A = \lambda_{Ah}^{1/2} \lambda_{Av}^{1/2} \Gamma_A \lambda_{Ah}^{1/2} \lambda_{Av}^{1/2}$ and $B = \lambda_{Bh}^{1/2} \lambda_{Bv}^{1/2} \Gamma_B \lambda_{Bh}^{1/2} \lambda_{Bv}^{1/2}$, which satisfy the normalization $\text{Tr}(A^\dagger A) = \text{Tr}(B^\dagger B) = 1$. Correspondingly, these two tensors also characterize $|\Psi_A\rangle$ and $|\Psi_B\rangle$ and are different from each other by phases or unitary matrices. More precisely, they are related by $A = U_h^\dagger U_v^\dagger B U_h U_v$, where we assume that U_h and U_v are unitary matrices. Fixing the gauge degrees of freedom between $|\Psi_A\rangle$ and $|\Psi_B\rangle$ requires determining matrices U_h and U_v . By generalizing the iterative method proposed in Sec. III B 2, we can find U_h and U_v such that they maximize the local fidelity, defined as

$$F = \text{Tr}(U_h U_v A U_h^\dagger U_v^\dagger B^\dagger), \quad (25)$$

which is graphically represented as



More precisely, U_h and U_v are obtained iteratively as follows. To start, we initialize them arbitrarily, for example, we could start from identity matrices (or some random matrices), and repeat the following steps.

(i) Fix matrices U_v , U_h^\dagger , and U_v^\dagger to search U_h by computing the matrix M_h such that

$$M_h = \text{Tr}(U_h U_v A U_h^\dagger U_v^\dagger B^\dagger) \quad (26)$$

(ii) Find the matrix U_h that might maximize the local fidelity (25). Similar to the 1D case, U_h can be determined via decomposing the matrix M_h using SVD such that $M_h = W_h S_h V_h^\dagger$. We then set $U_h = V_h W_h^\dagger$.

(iii) Update the new matrices U_h and U_h^\dagger .

(iv) Fix the matrices U_h , U_h^\dagger , and U_v^\dagger to search U_v in a similar way by computing the matrix M_v as follows:

$$M_v = \text{Tr}(U_h U_v A U_h^\dagger U_v^\dagger B^\dagger) \quad (27)$$

We then obtain $U_v = V_v W_v^\dagger$ from $M_v = W_v S_v V_v^\dagger$.

(v) Compute the fidelity F and check whether $F = 1$, up to some acceptable small tolerance. If not, go back to step (i).

V. ENVIRONMENT RECYCLING IN THE TEBD ALGORITHM FOR IMPS (1D)

In this section, we explain the environment recycling scheme in the TEBD algorithm for infinite 1D systems. For concreteness, we describe the scheme for the simple quantum Ising model with transverse magnetic field, where the time evolution operator is broken into gates representing the

interaction and magnetic field. The infinite system described by this model can be represented by a one-site translationally invariant iMPS. The scheme can be generalized to arbitrary Hamiltonians and larger unit cells.

A. Imaginary-time evolution for the 1D quantum Ising model

Consider an infinite spin-1/2 chain described by the Hamiltonian of the quantum ferromagnetic Ising model defined as

$$H = - \sum_i \sigma_i^z \sigma_{i+1}^z - h \sum_i \sigma_i^x, \quad (28)$$

where σ^z, σ^x are Pauli matrices, h is the amplitude of the transverse magnetic field.

In order to find the ground state of the system employing the iTEBD algorithm, we firstly rewrite the Hamiltonian as

$$H = \sum_i H_I^{[i,i+1]} + \sum_i H_F^{[i]}, \quad (29)$$

where $H_I^{[i,i+1]} = -\sigma_i^z \sigma_{i+1}^z$ and $H_F^{[i]} = -h \sigma_i^x$ describe the nearest-neighbor interaction and magnetic field terms, respectively. The commutation relations $[H_I^{[i,i+1]}, H_I^{[i',i'+1]}] = 0$ and $[H_F^{[i]}, H_F^{[i']}] = 0$ are satisfied whereas $[H_I^{[i,i+1]}, H_F^{[i']}] \neq 0$ possibly. The imaginary-time evolution operator at each time step δ can be decomposed by employing the second-order Suzuki-Trotter decomposition [58] such that

$$\begin{aligned} e^{-H\delta} &= e^{-\sum_i H_I^{[i]} \delta/2} e^{-\sum_i H_I^{[i,i+1]} \delta} e^{-\sum_i H_F^{[i]} \delta/2} + O(\delta^3) \\ &= \prod_i g_F^{[i]} \prod_{\text{oddi}} g_I^{[i,i+1]} \prod_{\text{eveni}} g_I^{[i,i+1]} \prod_i g_F^{[i]} + O(\delta^3), \end{aligned} \quad (30)$$

where we have defined

$$g_F^{[i]} = e^{-H_F^{[i]} \delta/2}, \quad g_I^{[i,i+1]} = e^{-H_I^{[i,i+1]} \delta}. \quad (31)$$

As it is expected to find the ground state of the system that is represented by a one-site translationally invariant iMPS, we need to express the time evolution operator in terms of the matrix product operator (MPO) representation [11, 59, 60]. As illustrated in Fig. 7, at each link between two lattice sites, we employ the SVD to decompose the interaction term as

$$g_I^{[i,i+1]} = U^{[i]} S^{[i]} V^{[i+1]}, \quad (32)$$

with the same contribution between the interaction and magnetic field terms at each site, we define the MPO as follows:

$$g^{[i]} = g_F^{[i]} V^{[i]} S^{[i]} U^{[i]} g_F^{[i]}. \quad (33)$$

The infinite MPO (iMPO) is then constructed by

$$g_\infty = \bigotimes_i g^{[i]}, \quad (34)$$

which is translationally invariant under a single rank-four tensor g of dimensions $\kappa \times \kappa \times d \times d$. To obtain the ground state of the system represented by an iMPS, we apply successively the iMPO to an initial state $|\Psi_0\rangle$ until the

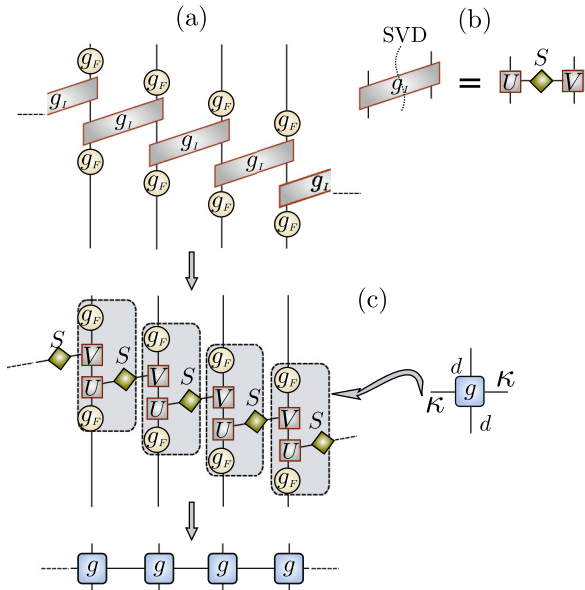


FIG. 7. (Color online) Diagrammatic illustration for constructing an iMPO. (a) The imaginary-time evolution operator is decomposed into a product of nearest-neighbor interaction terms denoted by g_I and magnetic field terms named g_F . (b) Perform the SVD of the interaction operator $g_I = USV$. (c) The iMPO is obtained by contracting all the tensors g_F, U, V, S together.

convergence is achieved. Mathematically, this is described by

$$\begin{aligned}
 |\Psi_{\text{GS}}\rangle &= \lim_{T \rightarrow \infty} \frac{e^{-HT} |\Psi_0\rangle}{\|e^{-HT} |\Psi_0\rangle\|} \\
 &= \lim_{m \rightarrow \infty} \frac{g^m |\Psi_0\rangle}{\|g^m |\Psi_0\rangle\|} \\
 &= \lim_{m \rightarrow \infty} \frac{(\bigotimes_i g^{[i]})^m |\Psi_0\rangle}{\|(\bigotimes_i g^{[i]})^m |\Psi_0\rangle\|}, \quad (35)
 \end{aligned}$$

where $m = T/\delta$ and is the number of iterations in iTEBD algorithm.

B. 1D environment recycling

We now describe the environment recycling scheme and how to incorporate it into the iTEBD algorithm to accelerate the convergence. We first need to determine quantities which characterize the environment in each update of the iTEBD algorithm. We assume that the system is represented by a canonical iMPS $|\Psi_{A_0}\rangle$ at time t_0 of the evolution. This iMPS is characterized by a tensor A_0 with bond dimension χ . In the next update, at time $t_0 + \delta$, the iMPS is updated to $|\Psi_{\Theta}\rangle$ with bond dimension $\kappa\chi$ where $\Theta = gA_0$. The iMPS $|\Psi_{\Theta}\rangle$ is then transferred into its canonical to perform the bond dimension truncation to obtain a new iMPS $|\Psi_{\tilde{A}}\rangle$ with bond dimension χ . This truncation is equivalent to finding matrices P and Q which dimensions are $\kappa\chi \times \chi$ and $\chi \times \chi\kappa$ respectively such that

$$\tilde{A} = Q\Theta P. \quad (36)$$

Note that the tensors P and Q are the objects that contain the information of the environment that we want to recycle.

We assume that the environment characterized by tensors P and Q changes slightly for the next few time steps of the evolution, and thus these tensors can be recycled instead of being re-calculated from the scratch at every iteration. In order to propose the environment recycling scheme, we first need to construct a so-called *renormalized gate* denoted as G which is obtained by following the fundamental steps.

(i) Transfer the iMPS $|\Psi_{\tilde{A}}\rangle$ into its canonical form, which is described by tensors $\{\tilde{\Gamma}_1, \tilde{\lambda}_1\}$. We obtain new iMPS $|\Psi_{\tilde{A}_1}\rangle$ with $\tilde{A}_1 = \sqrt{\tilde{\lambda}_1} \tilde{\Gamma}_1 \sqrt{\tilde{\lambda}_1}$. There is also a relation between \tilde{A}_1 and \tilde{A} such that $\tilde{A}_1 = Q_c \tilde{A} P_c$, which is shown graphically as

$$\tilde{A}_1 = Q_c \tilde{A} P_c \quad (37)$$

where Q_c and P_c are $\chi \times \chi$ matrices and can be obtained easily from the relation $Q_c \tilde{A} P_c = \sqrt{\tilde{\lambda}_1} \tilde{\Gamma}_1 \sqrt{\tilde{\lambda}_1}$.

(ii) Assume that $|\Psi_{A_0}\rangle$ and $|\Psi_{\tilde{A}_1}\rangle$ represent a similar physical state, and hence, there is a unitary transformation such that $\tilde{A}_1 \approx UA_0U^\dagger$, which is graphically shown as

$$\tilde{A}_1 \approx UA_0U^\dagger \quad (38)$$

where the unitary matrix U can be found using the methods mentioned in Sec. III B.

(iii) From the tensors $\{U, Q_c, Q, g, P, P_c\}$, we construct a tensor G such that $G = U^\dagger Q_c Q g P P_c U$, which is visualized as

$$G = U^\dagger Q_c Q g P P_c U \quad (39)$$

G is called the renormalized gate.

If we apply the renormalized gate G on the initial iMPS $|\Psi_{A_0}\rangle$, a new iMPS $|\Psi_{A_1}\rangle$ is obtained where A_1 has the same bond dimension with A_0 and the gauge degrees of freedom between them are fixed. At this point the environment recycling scheme can be applied by keeping applying the renormalized gate G successively to tensor A_0 for several times to get a new state $|\Psi_{A_N}\rangle$ before computing a new environment for the next environment recycling process.

C. 1D benchmark results

We have incorporated the environment recycling scheme into the iTEBD algorithm for iMPS to study the ground state of the 1D Ising model with the transverse magnetic field $h = 1.05$. In Fig. 8, we show the relative errors during the convergence of the ground state energy per link and local magnetization per site versus the number of times that the environment is calculated, denoted as i_{Re} , for different numbers of recyclings N_{Re} 's. The total number of time steps is $N_{ts} = N_{\text{Re}} i_{\text{Re}}$. The larger the N_{ts} the closer to the ground state. Therefore, in the plots, we see that with the same i_{Re} , the larger the N_{Re} smaller the relative error. Besides, note that unless N_{Re} is extremely large, it may take the same number of time steps N_{ts} to obtain the same ground state of the system for different N_{Re} 's. However, when $N_{\text{Re}} > 1$, the computational effort is already reduced because the number of environment

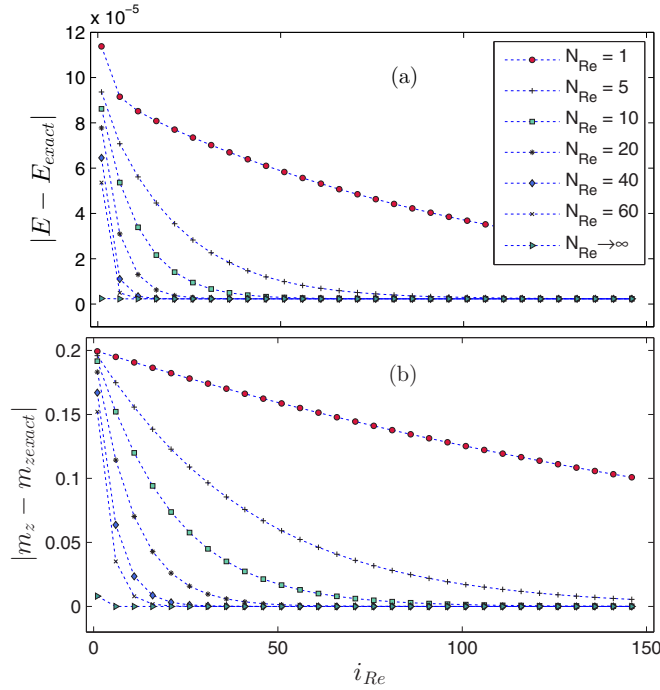


FIG. 8. (Color online) Plots of relative errors in the local expectation values regarding the number of times i_{Re} that the environment is calculated. The upper plot (a) shows the errors in the energy per link where $E_{\text{exact}} \approx -1.306856$. The bottom plot (b) shows the errors in the local magnetization per site where $m_{z,\text{exact}} = 0$. The results for different N_{Re} 's are obtained by simulating with the same initial iMPS of bond dimension $\chi = 50$, i_{Re} is counted from one, the transverse magnetic field $h = 1.05$, time step $\delta = 0.05$. The $N_{\text{Re}} \rightarrow \infty$ lines correspond to the cases where the dominant eigenvalue of the renormalized gate is found.

calculations is reduced. Therefore, the larger the N_{Re} , the faster the ground-state convergence.

A question naturally arises: can we keep recycling a particular environment as many times as we want? In principle, the answer is yes and the maximum of N_{Re} can be reached when the dominant eigenvector of the renormalized gate is found. Roughly, this can be understood as recycling the environment with an infinite number of times, i.e., $N_{\text{Re}} \rightarrow \infty$. However, if N_{Re} is chosen to be very large, the scheme can be unstable as the state of the system might fall into the local minima. Therefore, it depends on the problem, we can perform numerical calculation to choose an appropriate value for N_{Re} .

VI. ENVIRONMENT RECYCLING IN THE TEBD ALGORITHM FOR IPEPS (2D)

In the previous section we have shown that the environment recycling scheme works quite well for 1D systems using iMPS. Although it is not a crucial improvement of the TEBD with iMPS with OBC (recycling the environment in that case does not reduce significantly the computational cost), it is a good start to build the environment recycling scheme for 2D systems. In this section, we will present in detail a scheme for 2D infinite systems using iPEPS. For concreteness, we propose the scheme to study the ground state of an infinite square

lattice system described by the 2D quantum Ising model with a transverse magnetic field.

A. Imaginary-time evolution for the 2D quantum Ising model

The ferromagnetic Ising Hamiltonian for a quantum lattice system in 2D has the following form:

$$H = - \sum_{(\vec{r}, \vec{r}')} \sigma_{\vec{r}}^z \sigma_{\vec{r}'}^z - h \sum_{\vec{r}} \sigma_{\vec{r}}^x, \quad (40)$$

where (\vec{r}, \vec{r}') represents the nearest-neighbor sites. For a square lattice, each lattice site has four nearest neighbors. We then group the nearest-neighbor interactions into two terms, for interactions along the horizontal and vertical directions as follows:

$$H = - \sum_{i,j} \sigma_{(i,j)}^z \sigma_{(i+1,j)}^z - \sum_{i,j} \sigma_{(i,j)}^z \sigma_{(i,j+1)}^z - h \sum_{i,j} \sigma_{(i,j)}^x, \quad (41)$$

where we have used indices i and j to represent horizontal and vertical directions in the square lattice respectively, i.e., $\vec{r} = (i, j)$. The Hamiltonian can be also written as

$$H = \sum_{i,j} H_{I_h}^{[i,i+1]j} + \sum_{i,j} H_{I_v}^{[i,j,j+1]} + \sum_{i,j} H_F^{[i,j]}, \quad (42)$$

where $H_{I_h}^{[i,i+1]j} = -\sigma_{(i,j)}^z \sigma_{(i+1,j)}^z$ and $H_{I_v}^{[i,j,j+1]} = -\sigma_{(i,j)}^z \sigma_{(i,j+1)}^z$ are the horizontal and vertical nearest-neighbor interactions, respectively, and $H_F^{[i,j]} = -h \sigma_{(i,j)}^x$ is the one-site magnetic field. Notice that the commutation relations $[H_{I_h}^{[i,i+1]j}, H_{I_h}^{[i',i'+1]j'}] = 0$, $[H_{I_h}^{[i,i+1]j}, H_{I_v}^{[i',j',j'+1]}] = 0$, $[H_{I_h}^{[i,i+1]j}, H_{I_v}^{[i',j',j'+1]}] = 0$ and $[H_F^{[i,j]}, H_F^{[i',j']}] = 0$ are satisfied whereas in general $[H_{I_h}^{[i,i+1]j}, H_F^{[i',j']}] \neq 0$, $[H_{I_v}^{[i,j,j+1]}, H_F^{[i',j']}] \neq 0$.

In order to obtain the ground state of the system, we again employ the imaginary-time evolution. To start, we need to express the imaginary-time evolution operator in terms of the projected entangled-pair operator (PEPO). This can be done by decomposing the imaginary-time evolution operator as follows:

$$\begin{aligned} e^{-H\delta} &= e^{-\sum_{i,j} H_F^{[i,j]}\delta/2} e^{-\sum_{i,j} H_{I_h}^{[i,i+1]j}\delta} e^{-\sum_{i,j} H_{I_v}^{[i,j,j+1]}\delta} \\ &\quad \times e^{-\sum_{i,j} H_F^{[i,j]}\delta/2} + O(\delta^3) \\ &= \prod_{i,j} g_F^{[i,j]} \prod_{j,\text{oddi}} g_{I_h}^{[i,i+1]j} \prod_{j,\text{eveni}} g_{I_h}^{[i,i+1]j} \prod_{i,\text{oddi}} g_{I_v}^{[i,j,j+1]} \\ &\quad \times \prod_{i,\text{evenj}} g_{I_v}^{[i,j,j+1]} \prod_{i,j} g_F^{[i,j]} + O(\delta^3), \end{aligned} \quad (43)$$

where we have defined

$$\begin{aligned} g_F^{[i,j]} &= e^{-H_F^{[i,j]}\delta/2}, g_{I_h}^{[i,i+1]j} = e^{-H_{I_h}^{[i,i+1]j}\delta}, \\ g_{I_v}^{[i,j,j+1]} &= e^{-H_{I_v}^{[i,j,j+1]}\delta}. \end{aligned}$$

We construct the PEPO in a similar way with what we did above for the MPO. The scheme is explained diagrammatically in Fig. 9. More concretely, we apply SVD to decompose the interaction components in both horizontal and vertical

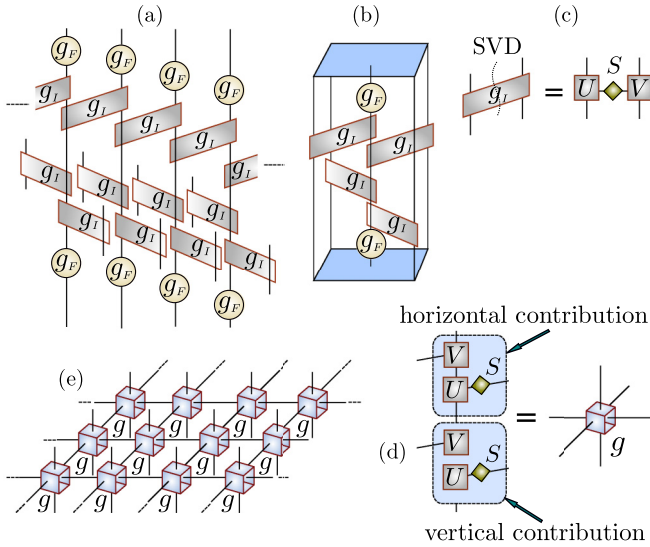


FIG. 9. (Color online) Diagrammatic illustration for building an iPEPO. (a) The imaginary-time evolution operator is decomposed into two terms: the interaction terms denoted by g_I in horizontal and vertical directions and the on-site magnetic field terms g_F . (b) A PEPO will be created from the horizontal and vertical contributions and the on-site magnetic field terms. (c) Perform the SVD of the interaction operators $g_I = USV$. (d) PEPO obtained from contracting tensors g_F, U, V, S . Each PEPO is a tensor consisting of two physical indices of dimension d and four virtual indices of dimension κ . (e) An iPEPO.

directions of the lattice as follows:

$$g_{I_h}^{[i,i+1]j} = U_h^{[i,j]} S_h^{[i,j]} V_h^{[i+1,j]} \quad (44)$$

and

$$g_{I_v}^{[i,j+1]} = U_v^{[i,j]} S_v^{[i,j]} V_v^{[i,j+1]}, \quad (45)$$

respectively; see Fig. 9(c). With the equal contributions of the interaction terms (from both directions) and magnetic field term, the PEPO at each lattice site is constructed by

$$g^{[i,j]} = g_F^{[i,j]} V_h^{[i,j]} U_h^{[i,j]} S_h^{[i,j]} V_v^{[i,j]} U_v^{[i,j]} S_v^{[i,j]} g_F^{[i,j]}, \quad (46)$$

as illustrated in Fig. 9(d). In Fig. 9(e), the iPEPO is then defined as a tensor product of identical local terms, which is given by

$$g_\infty = \bigotimes_{i,j} g^{[i,j]}. \quad (47)$$

The ground state of system will be obtained by acting on the iPEPS with the iPEPO gate as many times as needed until it is converged. We keep in mind that, as a result of acting the iPEPO gate on the iPEPS at each update, the bond dimension of this iPEPS will also increase by κ . Thus one needs to perform a truncation so that the bond dimension will remain the same after each update and the obtained iPEPS is also the best approximated ground state. At this point, we can take advantage of transferring the iPEPS into its canonical form to perform the truncation. This scheme is also known as the SU [45,46,48,49], applied here for a one-site translationally invariant system. The advantage of the SU is obvious as the computational cost is reduced essentially as compared to the FU. However, in most of the cases, results obtained from this scheme are less accurate, especially when applied to a critical

system where the correlations are long-range. In these cases, we need the FU, which involves calculating the environment and will be explained next.

B. Full update for a one-site translationally invariant iPEPS

The FU is typically implemented using two-site translationally invariant iPEPS [30,31,35]. In what follows, we will describe the FU for a one-site translationally invariant iPEPS.

At each time step of the evolution, after applying the iPEPO gate to the iPEPS, we get a new state $|\Psi_g\rangle = g_\infty |\Psi_A\rangle$ which is represented by tensor Θ with bond dimension κD . We need to find four optimal tensors $\{P_h, Q_h, P_v, Q_v\}$ to insert into four bonds of the tensor Θ simultaneously such that the updated iPEPS $|\Psi_{\tilde{A}}\rangle$ represented by $\tilde{A} = Q_h Q_v \Theta P_h P_v$ with bond dimension D is well-approximated to $|\Psi_g\rangle$ and this can be achieved by employing the FU.

The FU involves the two following fundamental processes. *Environment calculation.* There are a lot of feasible ways to compute the environment. For instance, one can utilize the coarse-graining tensor renormalization group methods, like TRG/SRG and HOTRG/HOSRG [37–40]. Alternatively, the environment can be also computed by means of the iTEBD algorithm applied to a boundary iMPS [30]. Here we apply the corner transfer matrix method (CTM) [31–36] to compute the environment. This method is known to be more efficient than the MPS schemes when studying the system in the vicinity of the phase transition [31]. In our case, an infinite square lattice $\mathcal{L}(a)$, where $a_{(l_1 l_2), (r_1 r_2), (d_1 d_2), (u_1 u_2)} = \sum_s \Theta_{l_1 r_1 d_1 u_1}^s (\Theta_{l_2 r_2 d_2 u_2}^s)^*$, is constructed by contracting the physical indices of $|\Psi_g\rangle$ and $\langle \Psi_g|$. We decompose $\mathcal{L}(a)$ into two contiguous regions \mathcal{A} and \mathcal{B} ; see Fig. 10(a). \mathcal{B} is chosen to contain four sites where the tensors are connected to each other by the links denoted as r (right), d (down), l (left), and u (up). \mathcal{A} plays a role as an environment around \mathcal{B} and is required to be computed. Applying the CTM method, this environment is computed approximately and characterized by a set of four corner tensors $\{C_1, C_2, C_3, C_4\}$ and eight edge tensors $\{T a_1, T a_2, T a_3, T a_4, T b_1, T b_2, T b_3, T b_4\}$, see Fig. 10(a). Note that we can further contract the network to obtain the environment represented by only six tensors $\{E_1, E_2, E_3, E_4, E_5, E_6\}$ around a specific link, see Fig. 10(b).

In order to obtain optimal tensors $\{P_h, Q_h, P_v, Q_v\}$, we need to variationally optimize the pairs of tensors (P_r, Q_r) , (P_d, Q_d) , (P_l, Q_l) , and (P_u, Q_u) so that they optimally truncate the bond dimensions of the tensor Θ corresponding to the links r , d , l , and u , respectively. This can be done via the standard variational method explained below.

Standard variational method. Let us define a cost function as follows:

$$f(\zeta) = |||\Psi_g\rangle - |\Psi_\zeta\rangle||^2 = \langle \Psi_g | \Psi_g \rangle + \langle \Psi_\zeta | \Psi_\zeta \rangle - \langle \Psi_g | \Psi_\zeta \rangle - \langle \Psi_\zeta | \Psi_g \rangle, \quad (48)$$

where $\zeta = (P_r, Q_r, P_d, Q_d, P_l, Q_l, P_u, Q_u)$. Note that $f(\zeta)$ is a quadratic function for every variable in ζ , and hence we can use a standard variational method to find ζ that minimizes $f(\zeta)$. More concretely, let us fix all tensors in ζ except for the first two tensors (P_r, Q_r) . We then find P_r and Q_r that minimize the cost function by following the following steps. (i) Fix Q_r with some arbitrary initial tensor or the tensor obtained from

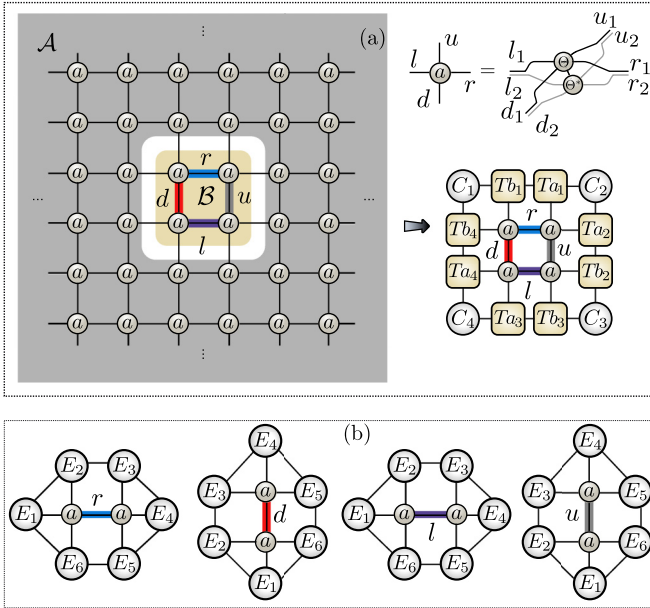


FIG. 10. (Color online) Compute the environment around a specific link using the CTM method. (a) An infinite square lattice $\mathcal{L}(a)$ is decomposed into two contiguous parts \mathcal{A} and \mathcal{B} . The environment \mathcal{A} around \mathcal{B} is characterized by corner tensors $\{C_1, C_2, C_3, C_4\}$ and edge tensors $\{T_{a1}, T_{a2}, T_{a3}, T_{a4}, T_{b1}, T_{b2}, T_{b3}, T_{b4}\}$. (b) The environment around a specific link is represented by six tensors $\{E_1, E_2, E_3, E_4, E_5, E_6\}$.

previous iteration and find P_r . Then we can write Eq. (48) as a quadratic scalar expression as follows:

$$f(P_r, P_r^\dagger) = P_r^\dagger R P_r - P_r^\dagger S - S^\dagger P_r + T, \quad (49)$$

where P_r is understood as a reshaped vector with $D^2\kappa$ components and matrices R, S, T can be obtained from contracting appropriate TNs including the contraction of the whole environment around the link r . (ii) From Eq. (49), the minimum of $f(P_r, P_r^\dagger)$ with respect to P_r is obtained only if $P_r = R^{-1}S$. (iii) Fix the tensor P_r and find Q_r with the same procedure as steps (i) and (ii) above. Update the new tensors P_r and Q_r in ζ and then turn into the links d, l , and u to update (P_d, Q_d) , (P_l, Q_l) , and (P_u, Q_u) , respectively.

The above process are iterated until all the tensors in ζ converge to fixed tensors. The convergence can be recognized by tracing of the cost function computed in the form of Eq. (48). Note also that instead of applying the optimization scheme described above, one could also apply other methods such as the conjugate gradient algorithm to optimize these tensors.

Once the tensors $(P_r, Q_r, P_d, Q_d, P_l, Q_l, P_u, Q_u)$ are converged, we can set $P_h = (P_r + P_l)/2$, $Q_h = (Q_r + Q_l)/2$, $P_v = (P_d + P_u)/2$, and $Q_v = (Q_d + Q_u)/2$. This is due to the links r and l are equivalent and so are the links d and u . The new iPEPS is represented by $\tilde{A} = Q_h Q_v \Theta P_h P_v$.

If the state is evolved by a large number of time steps, where at each time step one apply the iPEPO to the iPEPS combined with the bond dimension truncation using the above FU scheme, a converged iPEPS is obtained which represents approximately the ground state of the system. Note that the computational cost of this FU is quite expensive, as the cost

roughly scales as $\mathcal{O}(\chi^3 D^6 \kappa^6)$. This dominant cost is due to the environment calculation. Compared to the FU for a two-site translationally invariant iPEPS, where the dominant cost is $\mathcal{O}(\chi^3 D^6)$ as stated in Ref. [31], this one-site translationally invariant scheme is more expensive by a factor κ^6 , and hence, is much less efficient. Fortunately, we can get rid of the factor κ^6 by truncating the tensor Θ before sandwiching it to tensor a to calculate the environment. The truncation can be performed by using tensors $\{P_h, Q_h, P_v, Q_v\}$, which can be obtained from the previous update or the SU. In our simulation, we use the latter and it works very well. With this optimization, we reduce the computational cost down to $\mathcal{O}(\chi^3 D^6)$, which is the same as for the case of using two-site translationally invariant iPEPS.

C. 2D environment recycling scheme

The above FU provides a method to study the ground state of a one-site translationally invariant iPEPS, with the same computational cost as the usual two-site algorithm. However, this is still computationally very expensive and more importantly it requires a long time to converge to the ground state. The main aim of this paper is to show how we can further improve this drawback. This can be done by incorporating the environment recycling scheme into the iTEBD algorithm for iPEPS to speed up the convergence.

The 2D recycling environment scheme is constructed similarly with the case of 1D. More precisely, suppose that at time t_0 of the imaginary-time evolution, the iPEPS $|\Psi_{A_0}\rangle$ is in the canonical form and characterized by tensor A_0 with bond dimension D . At time $t_0 + \delta$, we apply the iPEPO to this iPEPS by contracting tensor g with A_0 . This results in tensor Θ of bond dimension κD . To keep the bond dimension of the iPEPS fixed at D , we employ the FU described above to get a new tensor $\tilde{A} = Q_h Q_v \Theta P_h P_v$, which is visualized as

$$\tilde{A} = Q_h Q_v \Theta P_h P_v \quad (50)$$

Note that the tensors $\{Q_h, Q_v, P_h, P_v\}$ indirectly contain the environment. As the environment is assumed to change slightly for the next few time steps of the evolution, we can recycle them. Similar to the case 1D, the recycling environment scheme includes two phases: define the *renormalized gate* G and successively apply G a few time steps on A_0 .

In order to determine G , we proceed with the following fundamental steps. (i) Transfer the iPEPS $|\Psi_{\tilde{A}}\rangle$ into its canonical form, which is characterized by tensors $\{\tilde{\Gamma}, \tilde{\lambda}_h, \tilde{\lambda}_v\}$. We then assign $\tilde{A}_1 = \sqrt{\tilde{\lambda}_h} \tilde{\Gamma} \sqrt{\tilde{\lambda}_v}$ for the iPEPS $|\Psi_{\tilde{A}_1}\rangle$. Next, we can easily find the matrices $\{Q_{hc}, Q_{vc}, P_{hc}, P_{vc}\}$ such that $\tilde{A}_1 = Q_{hc} Q_{vc} \tilde{A} P_{hc} P_{vc}$, which can be seen graphically as

$$\tilde{A}_1 = Q_{hc} \tilde{A} P_{vc} P_{hc} \quad (51)$$

(ii) Assume that $|\Psi_{\tilde{A}}\rangle$ and $|\Psi_{\tilde{A}_1}\rangle$ represent a similar physical state, then we can find the unitary matrices to fix the gauge

between two tensors A_0 and \tilde{A}_1 such that $\tilde{A}_1 \approx U_h U_v A_0 U_h^\dagger U_v^\dagger$ which is visualized as

$$\tilde{A}_1 \approx U_h U_v A_0 U_h^\dagger U_v^\dagger \quad (52)$$

(iii) Construct the renormalized gate defined as $G = U_h^\dagger U_v^\dagger Q_{hc} Q_h Q_{vc} Q_v g P_h P_{hc} Q_v P_v P_{vc} U_h U_v$, which is graphically shown as follows:

$$G = U_h^\dagger U_v^\dagger Q_{hc} Q_h Q_{vc} Q_v g P_h P_{hc} Q_v P_v P_{vc} U_h U_v \quad (53)$$

The recycling process is implemented by applying successively the renormalized gate G to state $|\Psi_{A_0}\rangle$ for several times to get a new state $|\Psi_{A_N}\rangle$. Next, we reassign state $|\Psi_{A_0}\rangle = |\Psi_{A_N}\rangle$ before applying another environment recycling process.

D. 2D benchmark results

In Fig. 11, we plot the local magnetization $m_z(h)$ per site, which is defined as

$$m_z(h) = \frac{\langle \Psi_h | \sigma^z | \Psi_h \rangle}{\langle \Psi_h | \Psi_h \rangle}, \quad (54)$$

where $|\Psi_h\rangle$ is the ground state pertaining to the magnetic field h . The plot also shows the comparisons between different update schemes: SU and FU. We can see that the result obtained from the FU are much better than the results of SU, especially in the vicinity of the criticality.

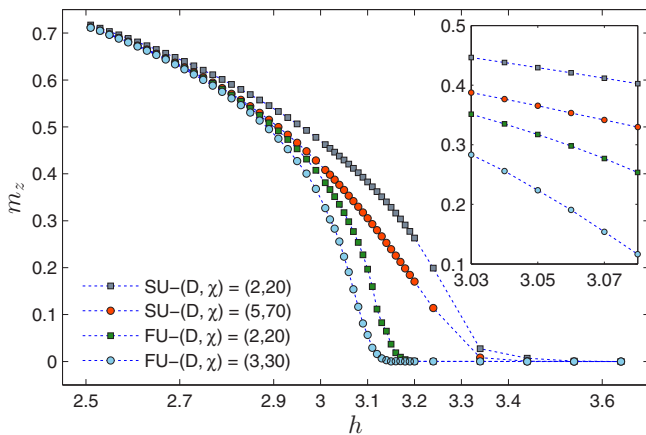


FIG. 11. (Color online) Expectation value for the local magnetization per site m_z as a function of the transverse magnetic field h . The result is obtained by using a one-site translationally invariant iPEPS simulating an imaginary time evolution with the iTEBD algorithm, both with SU and FU. The results close to the criticality are shown in the inset.

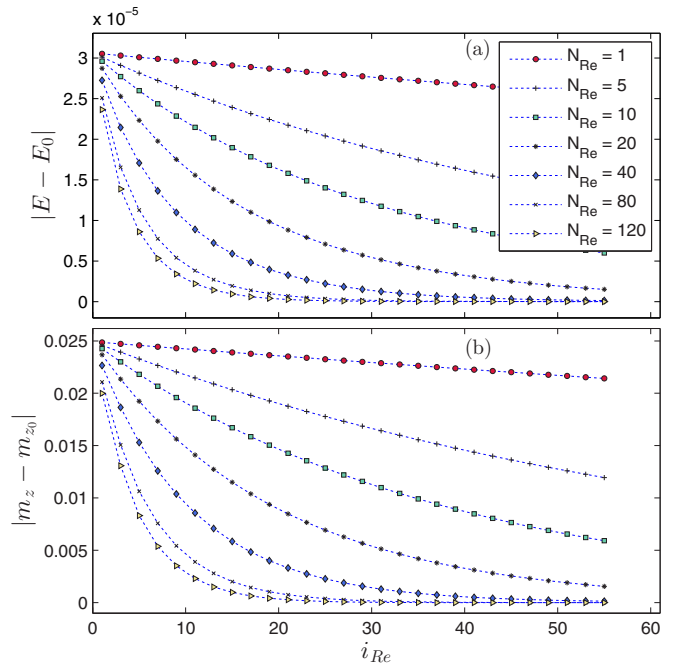


FIG. 12. (Color online) Plots of relative errors in the local expectation values regarding the number of times i_{Re} that the environment is calculated. The upper plot (a) shows the errors in the energy per link where $E_0 \approx -1.619581$. The bottom plot (b) shows the errors in the local magnetization per site where $m_{z_0} \approx 0.240717$. The results for different N_{Re} 's are obtained by simulating with the same initial iPEPS of bond dimension $D = 3$, $\chi = 30$, i_{Re} is counted from one, the transverse magnetic field $h = 3.05$ and time step $\delta = 0.005$.

Results of environment recycling scheme for the 2D quantum Ising model are shown in Fig. 12 using iPEPS with bond dimensions $D = 3$ and $\chi = 30$ is used for computing the environment with CTM. We observe how fast the algorithm drives the system to the ground state by calculating the energy per link $E(h)$ and magnetization per site $m_z(h)$, with different numbers of environment recyclings near the critical point $h = 3.05$. In both plots we realize that the larger the number of environment recyclings the faster the convergence into ground state we obtain. We show extra plots in Fig. 13 to compare the time consumed when the system is evolved with the same amount of computational time $T = N_{ts}\delta$ using different numbers of environment recyclings. The plots are illustrated for different bond dimensions of the iPEPS, i.e., $(D, \chi) = (2, 20)$ and $(D, \chi) = (3, 30)$. The plots show that the larger the N_{Re} , the smaller the time elapsed to evolve the system up to time T . We also fit the data and observe that the time measured almost decays exponentially with the number of environment recyclings.

Note also that as in the case of 1D environment recycling scheme, it is not always allowed to use a freely large number of recyclings although the larger it is the faster convergence to ground state. This is also applied for the case 2D. The reason is that instability appears if the same environment used to update the state that might be significantly different from the initial state, and hence it is contrary to the assumption that the states between successive time evolution update steps are close enough to each other. Therefore we need to wisely

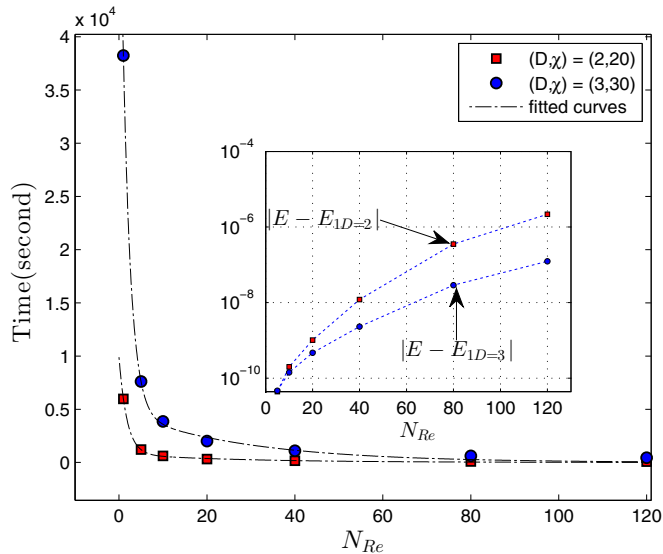


FIG. 13. (Color online) Plots of the elapsed time measured in terms of the number of environment recyclings N_{Re} . The system is evolved with the same amount of time $T = 12.05$ at time step $\delta = 0.01$ and $h = 3.01$. The plots are shown for different iPEPS bond dimensions where the exponential fitting curves are added to observe the abrupt decay in computational time when N_{Re} increases. The inset shows the relative errors in the energy per link obtained after T compared between $N_{Re} = 1$ and the others in each case of bond dimension where $E_{1D=2} \approx -1.601426$ and $E_{1D=3} \approx -1.601809$.

choose an appropriate number of recyclings to stabilize the algorithm when dealing with a specific problem.

VII. CONCLUSIONS

In this paper, we have first introduced a canonical form for both MPS (with PBC) in 1D and PEPS in 2D, as the fixed point of an iterative procedure, and used this canonical form as the basis to completely fix the gauge freedom in the MPS and PEPS. Then we have introduced a strategy to recycle the environment during a simulation of imaginary time evolution using the TEBD algorithm with iMPS and iPEPS.

By recycling the environment, we have obtained a significant reduction of computational costs while using the FU approach. The speed-up is accomplished because we only need to compute the environment once and can apply it for several time steps of the imaginary time evolution. To validate the schemes, we have studied the ground state of the quantum Ising model with transverse magnetic field in 1D and 2D and shown that the convergence into ground state is greatly accelerated—the computational time required to evolving the system for a given amount of imaginary time decays abruptly. However, the incorporation of environment recycling scheme into the TEBD algorithm with iPEPS sometimes causes numerical instability. For instance, if the initial state on which the time evolution is applied is very different than the ground state of the system, we find it is best to first use a FU scheme without recycling of the environment for a few time steps, and only use recycling once the many-body state has stopped changing dramatically at each time step. Similarly, the number N_{Re} of time steps during which the environment is recycled needs to be adjusted to ensure that the environment does not become too out-dated.

In summary, we have proposed an environment recycling strategy that significantly accelerates ground state computations using iPEPS. We envisage that this strategy will help boost the application of PEPS to challenging 2D problems, such as the resolution of t - J and Hubbard models [41,42].

ACKNOWLEDGMENTS

This research was supported in part by the ARC Centre of Excellence in Engineered Quantum Systems (EQuS), Project No. CE110001013 and the Discovery Projects funding scheme, Project No. DP1092513. H. N. Phien would like to thank the Centre for Health Technologies at The University of Technology Sydney for hospitality when completing this research. G. Vidal thanks the EQuS for hospitality at the University of Queensland, the NSERC (discovery grant), the John Templeton Foundation (Emergence grant) and the Simons Foundation (Many Electron Collaboration). Research at Perimeter Institute is supported by the Government of Canada through Industry Canada and by the Province of Ontario through the Ministry of Research and Innovation.

-
- [1] M. Fannes, B. Nachtergaele, and R. F. Werner, *Commun. Math. Phys.* **144**, 443 (1992).
 - [2] S. Ostlund and S. Rommer, *Phys. Rev. Lett.* **75**, 3537 (1995).
 - [3] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
 - [4] S. R. White, *Phys. Rev. B* **48**, 10345 (1993).
 - [5] H. Takasaki, T. Hikihara, and T. Nishino, *J. Phys. Soc. Jpn.* **68**, 1537 (1999).
 - [6] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
 - [7] D. Pérez-García, F. Verstraete, M. M. Wolf, and J. I. Cirac, *Quant. Inf. Comput.* **7**, 401 (2007).
 - [8] I. P. McCulloch, *J. Stat. Mech.: Theory Exp.* (2007) P10014.
 - [9] I. P. McCulloch, [arXiv:0804.2509v1](https://arxiv.org/abs/0804.2509v1) [cond-mat.str-e1].
 - [10] G. M. Crosswhite, A. C. Doherty, and G. Vidal, *Phys. Rev. B* **78**, 035116 (2008).
 - [11] U. Schollwöck, *Annals of Physics* **326**, 96 (2011).
 - [12] G. Vidal, *Phys. Rev. Lett.* **91**, 147902 (2003).
 - [13] G. Vidal, *Phys. Rev. Lett.* **93**, 040502 (2004).
 - [14] S. R. White and A. E. Feiguin, *Phys. Rev. Lett.* **93**, 076401 (2004).
 - [15] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal, *J. Stat. Mech.: Theory Exp.* (2004) P04005.
 - [16] H. N. Phien, G. Vidal, and I. P. McCulloch, *Phys. Rev. B* **86**, 245107 (2012).
 - [17] H. N. Phien, G. Vidal, and I. P. McCulloch, *Phys. Rev. B* **88**, 035103 (2013).
 - [18] V. Zauner, M. Ganahl, H. G. Everts, and T. Nishino, [arXiv:1207.0862](https://arxiv.org/abs/1207.0862).
 - [19] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pizorn, H. Verschelde, and F. Verstraete, *Phys. Rev. Lett.* **107**, 070601 (2011).

- [20] G. Vidal, *Phys. Rev. Lett.* **98**, 070201 (2007).
- [21] R. Orús and G. Vidal, *Phys. Rev. B* **78**, 155117 (2008).
- [22] F. Verstraete and J. I. Cirac, [arXiv:cond-mat/0407066](https://arxiv.org/abs/cond-mat/0407066).
- [23] V. Murg, F. Verstraete, and J. I. Cirac, *Phys. Rev. A* **75**, 033605 (2007).
- [24] F. Verstraete, M. M. Wolf, D. Pérez-García, and J. I. Cirac, *Phys. Rev. Lett.* **96**, 220601 (2006).
- [25] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, *New J. Phys.* **16**, 033014 (2014).
- [26] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, *Phys. Rev. B* **90**, 064425 (2014).
- [27] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, and Y. Akutsu, *Nucl. Phys. B* **575**, 504 (2000).
- [28] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, Y. Akutsu, and A. Gendiar, *Prog. Theor. Phys.* **105**, 409 (2001).
- [29] A. Gendiar, N. Maeshima, and T. Nishino, *Pro. Theor. Phys.* **110**, 691 (2003).
- [30] J. Jordan, R. Orus, G. Vidal, F. Verstraete, and J. I. Cirac, *Phys. Rev. Lett.* **101**, 250602 (2008).
- [31] R. Orús and G. Vidal, *Phys. Rev. B* **80**, 094403 (2009).
- [32] R. J. Baxter, *J. Math. Phys.* **9**, 650 (1968).
- [33] T. Nishino, K. Okunishi, *J. Phys. Soc. Jpn.* **65**, 891 (1996).
- [34] T. Nishino and K. Okunishi, *J. Phys. Soc. Jpn.* **67**, 3066 (1998).
- [35] P. Corboz, J. Jordan, and G. Vidal, *Phys. Rev. B* **82**, 245119 (2010).
- [36] R. Orús, *Phys. Rev. B* **85**, 205117 (2012).
- [37] M. Levin and C. P. Nave, *Phys. Rev. Lett.* **99**, 120601 (2007).
- [38] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, *Phys. Rev. Lett.* **103**, 160601 (2009).
- [39] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang, *Phys. Rev. B* **81**, 174411 (2010).
- [40] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, *Phys. Rev. B* **86**, 045139 (2012).
- [41] P. Corboz, S. R. White, G. Vidal, and M. Troyer, *Phys. Rev. B* **84**, 041108 (2011).
- [42] P. Corboz, T. M. Rice, and M. Troyer, *Phys. Rev. Lett.* **113**, 046402 (2014).
- [43] Z.-C. Gu, H.-C. Jiang, D. N. Sheng, H. Yao, L. Balents, and X.-G. Wen, *Phys. Rev. B* **88**, 155112 (2013).
- [44] L. Wang, Z.-C. Gu, F. Verstraete, and X.-G. Wen, [arXiv:1112.3331](https://arxiv.org/abs/1112.3331) [cond-mat.str-el].
- [45] H. C. Jiang, Z. Y. Weng, and T. Xiang, *Phys. Rev. Lett.* **101**, 090603 (2008).
- [46] W. Li, J. von Delft, and T. Xiang, *Phys. Rev. B* **86**, 195137 (2012).
- [47] The work related to defining a new type of canonical form for an iPEPS has been done independently with the quasicanonical form defined in Refs. [48, 49]. We also presented a poster related to this work at the workshop “Networking tensor networks: many-body systems and simulations” in Benasque, May 2012.
- [48] H. Kalis, D. Klagges, R. Orús, and K. P. Schmidt, *Phys. Rev. A* **86**, 022317 (2012).
- [49] R. Orús, H. Kalis, M. Bornemann, and K. P. Schmidt, *Phys. Rev. A* **87**, 062312 (2013).
- [50] F. Verstraete, D. Porras, and J. I. Cirac, *Phys. Rev. Lett.* **93**, 227205 (2004).
- [51] A. W. Sandvik and G. Vidal, *Phys. Rev. Lett.* **99**, 220602 (2007).
- [52] P. Pippin, S. R. White, and H. G. Evertz, *Phys. Rev. B* **81**, 081103 (2010).
- [53] J. D. Bekenstein, *Phys. Rev. D* **7**, 2333 (1973).
- [54] M. Srednicki, *Phys. Rev. Lett.* **71**, 666 (1993).
- [55] J. Eisert, M. Cramer, and M. B. Plenio, *Rev. Mod. Phys.* **82**, 277 (2010).
- [56] F. Verstraete and J. I. Cirac, *Phys. Rev. A* **70**, 060302(R) (2004).
- [57] D. Pérez-García, M. Sanz, C. E. Gonzalez-Guillen, M. M. Wolf, and J. I. Cirac, *New J. Phys.* **12**, 025010 (2010).
- [58] M. Suzuki, *Phys. Lett. A* **146**, 319 (1990); *J. Math. Phys. (N. Y.)* **32**, 400 (1991); For higher order expansions, see also A. T. Sornborger and E. D. Stewart, *Phys. Rev. A* **60**, 1956 (1999).
- [59] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New J. Phys.* **12**, 025012 (2010).
- [60] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, [arXiv:1407.1832](https://arxiv.org/abs/1407.1832) [cond-mat.str-el].