Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing

Ho N. Phien,[1] Johann A. Bengua,[1] Hoang D. Tuan,[1] Philippe Corboz,[2] and Román Orús[3]

[1]*Centre for Health Technologies, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, New South Wales 2007, Australia*

[2]*Institute for Theoretical Physics, University of Amsterdam, Science Park 904 Postbus 94485, 1090 GL Amsterdam, Netherlands*

[3]*Institute of Physics, Johannes Gutenberg University, 55099 Mainz, Germany*

The infinite projected entangled pair states (iPEPS) algorithm [J. Jordan *et al.*, Phys. Rev. Lett. **101**, 250602 (2008)] has become a useful tool in the calculation of ground-state properties of two-dimensional quantum lattice systems in the thermodynamic limit. Despite its many successful implementations, the method has some limitations in its present formulation which hinder its application to some highly entangled systems. The purpose of this paper is to unravel some of these issues, in turn enhancing the stability and efficiency of iPEPS methods. For this, we first introduce the *fast full update* scheme, where effective environment and iPEPS tensors are both *simultaneously* updated (or evolved) throughout time. As we shall show, this implies two crucial advantages: (i) dramatic computational savings and (ii) improved overall stability. In addition, we extend the application of the *local gauge fixing*, successfully implemented for finite-size PEPS [M. Lubasch *et al.*, Phys. Rev. B **90**, 064425 (2014)], to the iPEPS algorithm. We see that the gauge fixing not only further improves the stability of the method but also accelerates the convergence of the alternating least-squares sweeping in the (either "full" or "fast full") tensor update scheme. The improvement in terms of computational cost and stability of the resulting "improved" iPEPS algorithm is benchmarked by studying the ground-state properties of the quantum Heisenberg and transverse-field Ising models on an infinite square lattice.

I. INTRODUCTION

Recently, tensor networks have emerged as a powerful tool to understand quantum many-body systems [1]. From the point of view of numerical simulations there have been a number of novel algorithms developed, whose inner workings are deeply rooted in the theory of quantum entanglement. In all these algorithms quantum many-body states are conveniently represented by tensor networks that efficiently capture the natural structure of quantum correlations in the system (such as the so-called *entanglement area law* [2–6]). Many important classes of states can be accurately approximated by a tensor network with a number of parameters that depend only polynomially on the size of the system. Such properties, among others, have enabled tensor network methods to break the *curse of dimensionality*, namely, the fact that the Hilbert-space dimension of a quantum many-body system is exponentially large in the number of particles. Thanks to this, it is now possible to efficiently simulate quantum many-body systems by targeting the relevant tiny corner of quantum states (e.g., those satisfying an area law) inside of the exponentially large Hilbert space [7–9].

The so-called matrix product state (MPS) [10,11] is a typical tensor network ansatz representing the state of one-dimensional (1D) gapped quantum lattice systems. It is also well known that MPS is the class of variational wave functions at the root of the density matrix renormalization group (DMRG) method [12,13], widely used in the study of 1D systems [14–20]. Subsequently, MPS methods to study time evolution of 1D systems have also been put forward, such as time-evolving block decimation (TEBD) [7,8], time-dependent DMRG [21,22], and, more recently, algorithms based on the time-dependent variational principle [23]. One way of generalizing MPS methods to

higher-dimensional systems [24] is using projected entangled pair states (PEPS) [25–27], sometimes also called tensor product states (TPS) [28–30]. There has been a lot of progress in both conceptual and algorithmic developments for PEPS (see, e.g., Refs. [31–47]). From the numerical perspective, PEPS have been used to study ground-state properties as well as dynamics of two-dimensional (2D) lattice systems of both finite and infinite size. Moreover, motivated by the success of 1D methods in the thermodynamic limit such as infinite TEBD (iTEBD) [48,49], the so-called infinite PEPS (iPEPS) algorithm [32,33] was put forward to study infinite-size 2D quantum lattice systems.

So far, the iPEPS algorithm has been quite successful in the study of ground-state properties of a growing number of 2D quantum lattice systems (see, e.g., Ref. [1] and references therein). In general terms, results obtained by using iPEPS can be competitive when compared to the ones derived from quantum Monte Carlo [33,50]. More important, the iPEPS algorithm is not hampered by the *sign* problem (unlike quantum Monte Carlo) in the study of fermionic and frustrated spin systems. Recent applications of iPEPS to such systems include calculations for the $t - J$ model of fermions on the square [36,51,52] and honeycomb lattices [53], as well as the $J_1 - J_2$ frustrated Heisenberg model on the square lattice [54], the Shastry-Sutherland model [55], and the kagome Heisenberg antiferromagnet [56].

One of the main drawbacks of the iPEPS algorithm is its high computational cost as a function of the bond dimension D, which controls the accuracy of the method. This is particularly true when trying to obtain good accuracies in physical regimes where entanglement is large (e.g., close to a quantum critical point or in the presence of many nearly degenerate quantum states), which requires a large D. The computational bottleneck of the method is the calculation of

the so-called *effective environment*, i.e., the effective description of the tensor network surrounding a given site. Technically, effective environments can be computed using different approaches, such as Tensor Renormalization Group/Second Renormalization Group (TRG/SRG) and Higher Order Tensor Renormalization Group/Higher Order Second Renormalization Group (HOTRG/HOSRG) [31,35,40,57], iTEBD [32], corner transfer matrices (CTM) [33,58–62], or, more recently, the tensor network renormalization method [63]. Independent of the chosen approach for their calculation, accurate effective environments are required for the so-called *full update* (FU), which is the accurate scheme proposed to find the iPEPS tensors throughout an (imaginary) time evolution. Because of the large computational cost, only relatively small bond dimensions can be afforded when using the FU scheme.

Alternative update schemes have been developed trying to overcome this problem, but with only partial success. A very popular approach is the so-called *simple update* (SU) [34,39], which relies on a mean-field approximation of the effective environment and thus is very efficient. Such a scheme allows us to reach large bond dimensions in the iPEPS, but not surprisingly, it does not produce accurate results when systems are very strongly correlated. Intermediate approaches interpolating between the FU and the SU schemes have also been put forward as an alternative [43,44].

So here is the dilemma: accurate update schemes like the FU are too costly, whereas efficient update schemes such as the SU are not accurate enough. The question then is, Can one somehow "accelerate" the FU, making it more efficient while keeping its accuracy?

In this paper we give a positive answer to this question. We do this by constructing an update scheme, which we call "fast full update" (FFU), that significantly reduces the computational cost of iPEPS algorithms while still being accurate. More specifically, in this new strategy at every time step the tensors of the effective environment are updated by a single iteration step (in a sense to be made specific later) and simultaneously with those of the iPEPS. Importantly, we find that applying this strategy to iPEPS algorithms not only reduces the computational cost by a large factor (as expected) but also contributes to stabilizing the algorithm. The reason for this is that the successively updated environment helps to maintain the compatibility between the related tensors throughout the time evolution, as we shall explain later.

In addition, we show that incorporating the *local gauge-fixing* scheme proposed in Ref. [45] (and successfully applied to finite PEPS) can make the iPEPS algorithm even faster and more stable. As described in Ref. [45], the idea for the gauge fixing of PEPS tensors is inspired by the case of MPS, for which tensors can always be represented in a *canonical form* during their update by means of local gauge fixing. In the canonical form many of the tensor manipulations of an MPS get simplified (or directly canceled out), implying a much better conditioning and stability of related algorithms. However, unlike in the MPS case, there is no exact canonical form for PEPS in the same sense. A recent attempt along this direction is the so-called *quasicanonical form* for iPEPS [41,42,46]. This has been shown to lead to some computational advantages but unfortunately does not fully capture the effect of quantum correlations spreading throughout the 2D lattice.

A different approach was considered in Refs. [45,46], where it was shown that by considering the effect of the entire 2D lattice, a local gauge choice of the tensors can also produce a well-conditioned environment, which in turn improves the stability of the subsequent calculations. Here we apply the same local gauge fixing as in Ref. [45] to the iPEPS algorithm and show that it not only improves the stability but also accelerates the convergence of the alternating least-squares sweeping in the tensor-update scheme.

To show the validity of these approaches, we provide benchmarking calculations for the "improved" iPEPS algorithm with the two improvements mentioned above (FFU and gauge fixing). In particular, we analyze the computational cost and the stability of the algorithm for ground-state calculations of the Heisenberg and transverse-field Ising models on an infinite square lattice. We shall see quantitatively that the improvements both accelerate and stabilize the overall numerical calculations.

This paper is structured as follows. Some background material on PEPS, iPEPS, and the iPEPS algorithm is presented in Sec. II. In Sec. III we introduce the improvements (FFU and gauge fixing). Benchmarking calculations are presented in Sec. IV. Finally, conclusions are presented in Sec. V.

II. BACKGROUND

A. PEPS and iPEPS

1. Generalities

For completeness, we briefly review the notation and fundamental properties of projected entangled pair states [25–27]. To this end, let us consider a 2D quantum lattice system consisting of N sites, each of which is described by a local Hilbert space $\mathbb{C}^d$. The full Hilbert space of the system is thus $\mathcal{H} = (\mathbb{C}^d)^{\otimes N}$. As the dimension of the full Hilbert space grows exponentially with the size of the system, the problem quickly becomes intractable already for moderately low values of N. In order to avoid this curse of dimensionality, an option is to use a PEPS to represent a pure state of the system. Generally speaking, a PEPS is a state defined by a 2D lattice of interconnected tensors, i.e.,

$$|\Psi\rangle = \sum_{\{s_{\vec{r}_i}\}_{i=1}^N}^{d} F\left(A_{s_{\vec{r}_1}}^{[\vec{r}_1]}, \ldots, A_{s_{\vec{r}_N}}^{[\vec{r}_N]}\right) |s_{\vec{r}_1}, \ldots, s_{\vec{r}_N}\rangle,$$

(1)

where $|s_{\vec{r}_i}\rangle$ is the local basis of the site located at $\vec{r}_i = (x_i, y_i)$. Depending on the geometry of the lattice pattern, the tensor $A_{s_{\vec{r}_i}}^{[\vec{r}_i]}$ at each lattice site $\vec{r}_i$ contains $n_{\vec{r}_i}$ *bond* indexes taking up to D values ($n_{\vec{r}_i}$ is typically the number of nearest neighbors of the lattice site $\vec{r}_i$) and a *physical* index taking up to d values. The operation F contracts all the tensors $A_{s_{\vec{r}_i}}^{[\vec{r}_i]}$ along the bond indices. Conventionally, D is usually referred to as the bond dimension, which plays the role of a parameter quantifying both the size of the tensors in the PEPS and also the amount of entanglement in the wave function [64]. Thus, the larger D is, the better the PEPS can represent the state of the system (since there are more variational parameters). As an example, in Fig. 1(a) we illustrate a graphical representation of the PEPS for a 5×5 square lattice. In this case the number of
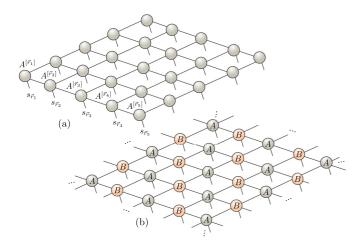
FIG. 1. (Color online) (a) Graphical representation of a PEPS on a 5×5 square lattice. Each ball is a tensor, and lines correspond to tensor indices. Lines from one tensor to another correspond to common summed indices (or *contracted* indices). The free index, or open leg, in each tensor is called the *physical index* and corresponds to the local degrees of freedom of the local Hilbert space at every site. (b) An infinite PEPS with a two-tensor unit cell. The two tensors, A and B, are repeated on the infinite 2D lattice.

complex parameters describing the PEPS quantum state is $O(NdD^4)$, thus being polynomial in N, D, and d, in contrast to the exponential dependence on N for an arbitrary state in the Hilbert space. For the sake of simplicity, from now on we shall always assume that we have a 2D square lattice [65].

2. Properties

PEPS have some important properties that make them an appropriate representation for 2D quantum lattice systems. First, and similar to MPS, PEPS satisfy the entanglement area law [2–6]. More specifically, the scaling of entanglement entropy of an $L \times L$ block of a PEPS is $O(L \log D)$. Accordingly, PEPS can describe very well the entanglement structure of many interesting 2D quantum systems, including low-energy eigenstates of many 2D Hamiltonians with local interactions. Second, PEPS can, in principle, represent systems with both *finite* and *infinite* correlation lengths [27]. This is to be contrasted with the case of MPS, where only a finite correlation length is possible. Third, and also unlike for MPS, given the loops present on 2D lattices there is no obvious canonical form for a PEPS (although some proposals have been recently put forward along this direction [41,42,46]). Last, but not least, PEPS can be used to represent systems in the *thermodynamic limit* by using a small number of tensors under the assumption of shift invariance. More precisely, a unit cell of tensors is repeated all over the 2D lattice to construct an arbitrarily large shift-invariant PEPS. For an infinite system, this is the so-called infinite PEPS, or iPEPS [see Fig. 1(b) for a graphical illustration of an iPEPS with a two-site unit cell].

3. Numerical application

PEPS can be used to study both ground-state properties and dynamics of 2D quantum lattice systems. For ground states, one can either (i) variationally optimize the PEPS tensors so

as to minimize the expectation value of the 2D corresponding Hamiltonian (as done in DMRG in one dimension) or (ii) evolve the system in imaginary time until a fixed point (ground state) is reached (as done in TEBD in one dimension). The second method can also be applied to study real-time evolution of the system. This approach is also easy to extend to the thermodynamic limit, called the iPEPS algorithm [32,33], which we go over again briefly in the next section.

B. The iPEPS algorithm

1. Generalities

For a given Hamiltonian H, the ground state of the system can be obtained by evolving an initial state $|\Psi_0\rangle$ in imaginary time β as described by

$$|\Psi_{\mathrm{GS}}\rangle = \lim_{\beta \to \infty} \frac{e^{-H\beta}|\Psi_0\rangle}{||e^{-H\beta}|\Psi_0\rangle||}. \quad (2)$$

Moreover, the real-time evolution is described via the solution of the Schrödinger equation, which for a time-independent Hamiltonian H reads

$$|\Psi(t)\rangle = e^{-iHt}|\Psi_0\rangle. \quad (3)$$

From now on, let us consider the evolution of an iPEPS in imaginary time (the extension to real-time evolution is technically straightforward).

We assume that the Hamiltonian contains only translationally invariant nearest-neighbor interactions, i.e.,

$$H = \sum_{\langle \vec{r}, \vec{r}' \rangle} h^{[\vec{r}, \vec{r}']}, \quad (4)$$

where the sum is performed over all the nearest neighbors $\langle \vec{r}, \vec{r}' \rangle$. For a Hamiltonian that is invariant under translations, we can use an iPEPS also with translation symmetry to represent the wave function $|\Psi\rangle$ of the system. This could be achieved by repeating the same tensor on each lattice site all over the lattice. However, our update scheme (described below) requires neighboring tensors to be different, which is why we typically use the two-site translationally invariant iPEPS shown in Fig. 1(b), depending only on two tensors, A and B. If translational symmetries are spontaneously broken in the thermodynamic limit, then a larger unit cell of tensors (compatible with the structure of the ground state) can be used instead [51].

In order to compute the ground state of the system by an evolution in imaginary time, one first decomposes the time-evolution operator into a product of so-called *two-body gates*. To this end, the Hamiltonian H is rewritten as

$$H = H_l + H_r + H_u + H_d, \quad (5)$$

where each term $H_i = \sum_{\langle \vec{r}, \vec{r}' \rangle \in i} h^{[\vec{r}, \vec{r}']}, i \in (l, r, u, d)$, is the sum of mutually commuting Hamiltonian terms for links labeled as (left, right, up, down). Notice, however, that the commutator $[H_i, H_j]$ is, in general, different from zero whenever $i \neq j$. Applying the first-order Suzuki-Trotter decomposition [66], we can write the time-evolution operator as

$$\begin{aligned} e^{-H\beta} &= (e^{-H\delta})^m \\ &\approx (e^{-H_l\delta}e^{-H_r\delta}e^{-H_u\delta}e^{-H_d\delta})^m, \end{aligned} \quad (6)$$

where δ is an infinitesimal time step and $m \equiv \beta/\delta \in \mathbb{N}$ is the number of "time steps" that need to evolve the system, so as to reach total evolution time β. Since each term H_i is a sum of mutually commuting terms, we can further write each term $e^{-H_i\delta}$ in Eq. (6) as a product of two-body gates, i.e.,

$$e^{-H_i\delta} = \prod_{\langle \vec{r}, \vec{r}' \rangle \in i} g^{[\vec{r}, \vec{r}']}, \tag{7}$$

with $g^{[\vec{r}, \vec{r}']} \equiv e^{-h^{[\vec{r}, \vec{r}']}\delta}$.

For a given time step of the evolution, let us consider the action of the term $e^{-H_i\delta}$ on an infinite PEPS $|\Psi_{AB}\rangle$ with two tensors A and B and bond dimension D. To this aim we focus on *one* link on the lattice and, *initially*, disregard the effect of the gates $g^{[\vec{r}, \vec{r}']}$ on the rest of the links (which is approximately correct since $\delta \ll 1$ and thus $g^{[\vec{r}, \vec{r}']} \sim \mathbb{I}$). Let us consider the case of an r link, for concreteness. After applying the gate, we obtain a new iPEPS $|\Psi_{A'B'}\rangle$ which is characterized by tensors A and B everywhere except for the two tensors connected by the link where the gate acted. More precisely, $|\Psi_{A'B'}\rangle$ and $|\Psi_{AB}\rangle$ differ from each other by only two tensors. Because of the effect of the gate, the bond dimension of the affected index changes to $D' \leqslant d^2 D$ and thus increases, corresponding to a change of the entanglement in the tensor network. Because of this, the iPEPS bond dimension quickly increases exponentially fast after a few gate applications, making the simulation intractable. To overcome this problem, the infinite PEPS $|\Psi_{A'B'}\rangle$ is approximated by a new PEPS $|\Psi_{\tilde{A}\tilde{B}}\rangle$ by replacing tensors A' and B' by $\tilde{A}$ and $\tilde{B}$, where these two last tensors have again bond dimension D for the affected index. This is done in a way such that the state $|\Psi_{\tilde{A}\tilde{B}}\rangle$ is close to the exact state $|\Psi_{A'B'}\rangle$ and thus introduces only a small error. Such a procedure is called the *tensor update* of the iPEPS.

A possibility to implement the tensor update is to look for new tensors $\tilde{A}$ and $\tilde{B}$ that minimize the squared distance between the exact and the approximating states, i.e.,

$$\min_{\tilde{A},\tilde{B}} |||\Psi_{A'B'}\rangle - |\Psi_{\tilde{A}\tilde{B}}\rangle||^2 = \min_{\tilde{A},\tilde{B}} d(\tilde{A}, \tilde{B}), \tag{8}$$

with

$$d(\tilde{A}, \tilde{B}) = \langle \Psi_{A'B'}|\Psi_{A'B'}\rangle + \langle \Psi_{\tilde{A}\tilde{B}}|\Psi_{\tilde{A}\tilde{B}}\rangle$$
$$- \langle \Psi_{\tilde{A}\tilde{B}}|\Psi_{A'B'}\rangle - \langle \Psi_{\tilde{A}\tilde{B}}|\Psi_{A'B'}\rangle. \tag{9}$$

To solve the problem in Eq. (8), two main tasks are needed. These are (i) the effective environment calculation and (ii) the tensor update. Since these pieces of the method will be fundamental for the improvements to be later explained, we review them in detail in the following.

2. Effective environment calculation

To properly evaluate $d(\tilde{A}, \tilde{B})$ one needs to take into account the effect of the whole tensor network surrounding the affected link, i.e., the *environment*. Such a tensor network of infinitely many tensors is conveniently approximated by an *effective environment*, consisting of only a small number of tensors. The effective environment can be computed using various approaches, such as TRG/SRG, HOTRG/HOSRG [31,35,40,57], tensor network renormalization (TNR) [63], iTEBD [32], and CTM methods. This last approach will be our choice in this paper. We shall not explain CTM methods
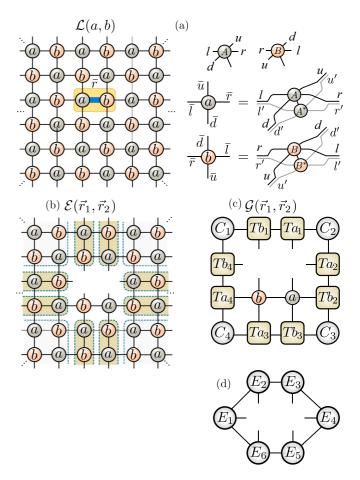

FIG. 2. (Color online) (a) Left: Two-dimensional lattice of tensors formed from a and b; right: contractions to obtain tensors a and b. (b) Environment of a given link on the lattice (here an r link). (c) Effective environment of a given link on the lattice (here an r link). (d) Six-tensor representation of the effective environment around the link.

in full detail here, and we refer the reader to the extensive existing literature on the topic, such as Refs. [33,58–62], for technicalities. However, since these methods will also be important at a later stage of this paper, we review briefly some notations and conventions.

We first construct an infinite square lattice $\mathcal{L}(a,b)$ by contracting the physical indexes of $|\Psi_{AB}\rangle$ and $\langle\Psi_{AB}|$ [see Fig. 2(a)], where

$$a_{\bar{l}\bar{r}\bar{u}\bar{d}} = \sum_s A^s_{lrud} (A^s_{l'r'u'd'})^*, \tag{10}$$

$$b_{\bar{l}\bar{r}\bar{u}\bar{d}} = \sum_s B^s_{lrud} (B^s_{l'r'u'd'})^*, \tag{11}$$

where $\bar{l}, \bar{r}, \bar{u}, \bar{d}$ are combined indices, e.g., $\bar{l} = (l,l')$.

The exact environment $\mathcal{E}(\vec{r}_1, \vec{r}_2)$ of two sites at $\vec{r}_1$ and $\vec{r}_2$ is shown in Fig. 2(b). In CTM methods, this is approximated by an effective tensor network $\mathcal{G}(\vec{r}_1, \vec{r}_2)$, the effective environment, which comprises a set of four $\chi \times \chi$ corner transfer matrices $\{C_1, C_2, C_3, C_4\}$, eight half transfer row/column tensors $\{Ta_1, Ta_2, Ta_3, Ta_4, Tb_1, Tb_2, Tb_3, Tb_4\}$, and two tensors, a and b [see Fig. 2(c)]. A further simple contraction of the tensor network produces an effective
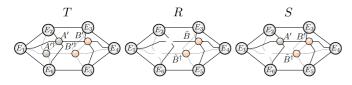
FIG. 3. (Color online) Tensor network diagrams showing how to compute T, R, and S in Eq. (12).

environment for the considered link in terms of only six tensors, $\{E_1, E_2, E_3, E_4, E_5, E_6\}$ [see Fig. 2(d)].

3. iPEPS tensor full update

Equation (9) implies that the squared distance $d(\tilde{A}, \tilde{B})$ is a quadratic function of tensors $\tilde{A}$ and $\tilde{B}$. Thus, once the effective environment is obtained, one can apply an alternating least-squares method to optimize tensors $\tilde{A}$ and $\tilde{B}$. This is the so-called full update, and it follows these steps:

(i) We fix tensor $\tilde{B}$ to some initial tensor [67] or to the tensor obtained from the previous iteration. In order to find $\tilde{A}$, rewrite Eq. (9) as a quadratic scalar expression for the tensor,

$$d(\tilde{A}, \tilde{A}^\dagger) = \tilde{A}^\dagger R \tilde{A} - \tilde{A}^\dagger S - S^\dagger \tilde{A} + T. \quad (12)$$

In this equation, $\tilde{A}$ is understood as a reshaped vector with $D^4 d$ components, and matrices R, S, T can be obtained from the appropriate tensor contractions including the effective environment around the r link (see Fig. 3).

(ii) We find the minimum of $d(\tilde{A}, \tilde{A}^\dagger)$ in Eq. (12), with respect to $\tilde{A}^\dagger$, which is given by $\tilde{A} = R^{-1} S$.

(iii) Next, we fix tensor $\tilde{A}$ and find $\tilde{B}$ using the same procedure as in steps (i) and (ii) above.

The above steps are iterated until the cost function $d(\tilde{A}, \tilde{B})$ converges to a sufficiently small value. A possibility to check convergence is, e.g., to check the value of this cost function between two successive iterations and compare it to some small tolerance. Last, but not least, once the optimal tensors are found, they are replaced over the entire 2D lattice, approximating the effect of all the gates $g^{[\vec{r}, \vec{r}']}$ acting over the infinitely many links of the same type. Such a procedure defines the updated infinite PEPS $|\Psi_{\tilde{A}\tilde{B}}\rangle$ in terms of the two new tensors. Finally, the same procedure is repeated for the l, u, and d links to complete one time step. This is iterated until the desired real running time is achieved or until the iPEPS converges to a fixed-point approximation of the ground state for imaginary-time evolutions.

As such, the computational cost of the procedure above is quite high due to the calculation of the inverse of matrix R in step (ii). Specifically, this cost is $O(D^8)$ if recurrent methods are applied for computing the inverse (e.g., biconjugate gradient). Otherwise, the cost would be $O(D^{12})$ for an exact inverse calculation.

4. Reduced-tensor full update

As an efficient and convenient alternative to this method, one can apply the "revised" FU scheme discussed in Ref. [36], where, instead of updating tensors $\tilde{A}$ and $\tilde{B}$, one updates some lower-rank *subtensors* related to them (sometimes they are also called *reduced tensors* [45]). These subtensors are denoted $\tilde{a}_R$ and $\tilde{b}_L$, respectively. The main idea of this optimization
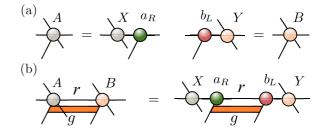

FIG. 4. (Color online) (a) Subtensors for iPEPS tensors A and B. Decompose $A = X a_R$ and $B = b_L Y$ by means of decompositions such as QR or SVD. (b) The action of the two-body gate g on the iPEPS tensors A and B connected by an r link is equivalent to its action on only the subtensors a_R and b_L, leaving X and Y unaffected.

scheme is based on the observation that applying the two-body gate g on two sites A and B connected by a specific link changes the properties of *only* that link, while the others remain unchanged. For instance, for an r link this results in modifying the bond dimension for this link from D to $D' > D$, while the size of the other indices is unaffected.

Thus, by means of the QR or singular-value decomposition (SVD), we decompose tensors A and B such that $A = X a_R$ and $B = b_L Y$, where a_R and b_L are connected by the bond index on the r link and also contain the physical indices [see Fig. 4(a)]. When applying the two-body gate g, we now contract it with a_R and b_L in order to update them directly. Once a_R and b_L are updated to, say, $\tilde{a}_R$ and $\tilde{b}_L$, we can easily get updated iPEPS tensors $\tilde{A}$ and $\tilde{B}$ by using $\tilde{A} = X \tilde{a}_R$ and $\tilde{B} = \tilde{b}_L Y$. The update scheme for $\tilde{a}_R$ and $\tilde{b}_L$ can be performed in a way similar to what we explained for $\tilde{A}$ and $\tilde{B}$, following steps (i)–(iii) above. More precisely, in these steps we replace $\tilde{A}$ and $\tilde{B}$ by $\tilde{a}_R$ and $\tilde{b}_L$, which are alternatively optimized according to a similar figure of merit defined as

$$d(\tilde{a}_R, \tilde{b}_L) = \langle \Psi_{a'_R b'_L} | \Psi_{a'_R b'_L} \rangle + \langle \Psi_{\tilde{a}_R \tilde{b}_L} | \Psi_{\tilde{a}_R \tilde{b}_R} \rangle$$
$$- \langle \Psi_{\tilde{a}_R \tilde{b}_L} | \Psi_{a'_R b'_L} \rangle - \langle \Psi_{\tilde{a}_R \tilde{b}_L} | \Psi_{a'_R b'_L} \rangle. \quad (13)$$

Equation (12) for $\tilde{a}_R$ is now rewritten as

$$d(\tilde{a}_R, \tilde{a}_R^\dagger) = \tilde{a}_R^\dagger R \tilde{a}_R - \tilde{a}_R^\dagger S - S^\dagger \tilde{a}_R + T, \quad (14)$$

and the cost function for variable $\tilde{b}_L$ is defined in a similar way. Note that the tensors $\{R, S, T\}$ in Eq. (14) are different from the ones in Eq. (12) (see Fig. 5).

The computational cost of this update scheme is $O(d^3 D^6)$, where the inverse of the matrix R can now be computed *exactly*. Due to this huge advantage in computational cost compared to the direct update of the iPEPS tensors A and B, the present scheme is able to deal with iPEPS of larger bond dimension D, with the corresponding advantages.
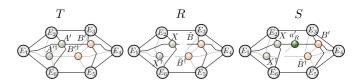

FIG. 5. (Color online) Tensor network diagrams showing how to compute T, R, and S in Eq. (14).

The rest of the update follows as explained in the previous section. That is, the reduced tensors are optimized until the cost function is sufficiently small, and then, they are replaced over the entire 2D lattice. As explained before, this last step approximates the effect of all the gates acting on all the links of the lattice of the same type. As before, the procedure defines an updated infinite PEPS $|\Psi_{\tilde{A}\tilde{B}}\rangle$ in terms of two new tensors. After this, one repeats the same procedure for the l, u, and d links to complete one time step of the evolution. This will be iterated until the iPEPS converges to a fixed point approximating the ground state (for imaginary time) or until the desired real-time evolution is completed.

III. IMPROVING THE iPEPS ALGORITHM

One of the main limitations of the FU iPEPS algorithm in its present formulation is that the effective environment has to be recomputed from scratch at every time step. Obtaining a converged effective environment requires N_{CTM} iterations of the CTM algorithm, with N_{CTM} depending on the amount of entanglement in the system (for not too entangled systems, typically, $N_{\mathrm{CTM}} \sim 10\text{--}20$, but it can be considerably larger in strongly entangled systems). Since these CTM iterations are the computational bottleneck of the method, it is desirable to keep N_{CTM} as low as possible. However, if one uses an N_{CTM} which is too small, then the errors introduced in the effective environment can lead to instability problems of the method.

Here we propose two improvements to the algorithm which enhance the stability of the method and make it more efficient. First, we explain how to implement what we call a fast full update, where the accuracy of the FU is preserved while substantially reducing its computational cost and improving its stability. Second, we discuss the application of a local gauge fixing, as in Ref. [45], which naturally improves stability and also accelerates the convergence of the overall method. This is explained in what follows, and benchmarking numerical calculations shall be provided in the forthcoming sections.

A. Fast full update

In this update scheme we reduce the computational cost by using the following idea: instead of recomputing the environment tensors from scratch, we can update the environment tensors *simultaneously* with those of the iPEPS at each step, just by *a single CTM iteration step*. This reduces the computational cost by a large factor (see results below). The crucial technical point of this FFU scheme is to make sure that the environment tensors remain compatible with the updated iPEPS tensors throughout the (imaginary) time evolution.

The details of the FFU scheme for iPEPS tensors A and B plus the effective environment tensors are shown in Fig. 6. In the following we explain how to proceed at each time step when the two-body gates g are successively applied on the r, l, u, and d links:

(a) *r link update.* Suppose that the effective environment around four links of the iPEPS is characterized by tensors $\{C_i, Ta_i, Tb_i\}_{i=1}^4$ [see Fig. 6(a-i)]. Applying the gate g to the r link will modify the properties of this link, so we need to update the tensors A and B using the update scheme from the previous section. Specifically, the effective environment
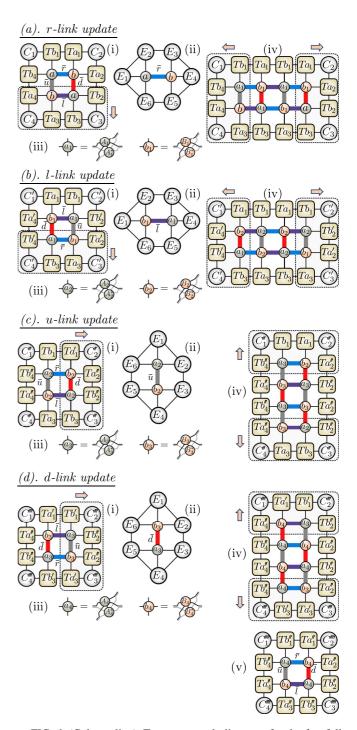

FIG. 6. (Color online) Tensor network diagrams for the fast full update. Details are explained in the main text.

of an r link is obtained by first absorbing a row of tensors $\{Ta_4, b, a, Tb_2\}$ in the bottom edge and then contracting the tensors appropriately, such that the effective environment is represented by six tensors $\{E_i\}_{i=1}^6$ as in Fig. 6(a-ii). We then apply the tensor update scheme explained previously in order to find some updated tensors A_1 and B_1.

In order to prepare for the next update of the l link, we now update the environment tensors. To this end, we insert two columns of tensors, $\{Ta_1, b_1, a_1, Tb_3\}$ and $\{Tb_1, a_1, b_1, Ta_3\}$, in the middle of the tensor network [see Figs. 6(a-iii) and 6(a-iv)].

Note that the two tensors a and b connected by link $\bar{r}$ are now replaced by the updated ones, a_1 and b_1. But importantly, there are still two sites where we use the old tensors, a and b, since the corresponding r link is connected to the old environment and thus was not "formally" updated.

Next, the columns of tensors $\{Ta_1, b_1, a, Tb_3\}$ and $\{Tb_1, a_1, b, Ta_3\}$ are absorbed at the right and left edges, respectively [see Fig. 6(a-iv)]. After this absorption, one finds appropriate isometries to renormalize the environment tensors as usual, in such a way that their bond dimensions do not grow [33]. The new environment tensors of the l link are denoted as in Fig. 6(b-i).

(b) *l link update.* For the action of a gate on the l link, we proceed in the same way as with the r link, which is shown graphically in Fig. 6(b). The iPEPS is now updated and represented by two new tensors, A_2 and B_2. The new environment tensors for updating the u link are denoted as in the Fig. 6(c-i).

(c) *u link update.* From the tensor network in Fig. 6(c-i), we obtain the environment tensors for the u link as shown in Fig. 6(c-ii). An update of the tensors will now produce two new tensors, A_3 and B_3. We then compute the effective environment for the update of the d link as shown in Fig. 6(c-iv)

(d) *d link update.* Now we follow the same procedure as for the u link in order to update the iPEPS tensors as well as the effective environment. This is shown in Fig. 6(d). Finally, we obtain the new iPEPS tensors, A_4 and B_4, and the new effective environment of the four links is represented by tensors $\{C_i'''', Ta_i'', Tb_i''\}_{i=1}^4$ [see Fig. 6(d-v)].

The above update scheme is illustrated for only one time step. One then needs to iterate this scheme in order to evolve the system up to the desired time. As in the update approaches explained in the previous section, for the FFU it is also a good idea to choose properly the initial state in ground-state calculations (e.g., converged iPEPS obtained from the SU scheme or from the FU scheme with a smaller bond dimension). Evidently, for the FFU this choice also helps in the stability and fast convergence of the algorithm.

The FFU that we just presented has two key advantages: first, one keeps an environment at every step that is perfectly compatible with the tensors in the iPEPS in all bond indices. This is the reason why, e.g., in Fig. 6(a-iv) we still have two tensors, a and b, at two sites. Such a property naturally improves the stability. Second, the environment is not reconverged for every link at every step [68]. As a result, this reduces the required computational time considerably.

B. Gauge fixing

In contrast to MPS, a PEPS does not have a canonical form. Therefore, it is difficult to fix the gauge of the PEPS tensors in some appropriate way during the time evolution. Despite this, local gauge-fixing schemes have proven useful in improving the stability of the algorithm [41,42,45,46]. In this paper, we use the local gauge-fixing proposed in Ref. [45] (applied there to finite PEPS) in the context of the iPEPS algorithm. We shall see that this helps not only to improve the stability of the method but also to accelerate its convergence.
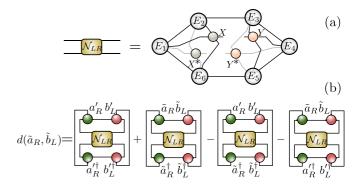

FIG. 7. (Color online) (a) Contraction producing the norm tensor. The leading computational cost of this contraction is $O(d^4 D^4 \chi^2) + O(d^2 D^6 \chi^2) + O(d^2 D^3 \chi^3)$ [45]. (b) Diagrammatic representation of the cost function defined in Eq. (15).

To this end, we consider the tensor update applied to the reduced tensors. We rewrite Eq. (13) as follows:

$$d(\tilde{a}_R, \tilde{b}_L) = a_R'^\dagger b_L'^\dagger \mathcal{N}_{LR} a_R' b_L' + \tilde{a}_R^\dagger \tilde{b}_L^\dagger \mathcal{N}_{LR} \tilde{a}_R \tilde{b}_R$$
$$- \tilde{a}_R^\dagger \tilde{b}_L^\dagger \mathcal{N}_{LR} a_R' b_L' - \tilde{a}_R^\dagger \tilde{b}_L \mathcal{N}_{LR} a_R' b_L', \quad (15)$$

where $\mathcal{N}_{LR}$ is the "norm" tensor obtained by computing the overlap of $\langle \Psi_{\tilde{a}_R \tilde{b}_L} | \Psi_{\tilde{a}_R \tilde{b}_R} \rangle$ while leaving out tensors $\tilde{a}_R, \tilde{b}_L$ and their complex conjugates. Specifically, this can be done by contracting the tensor network shown in Fig. 7(a). Also, the cost function is represented diagrammatically in Fig. 7(b). In order to update the subtensors one needs, thus, to compute the norm tensor $\mathcal{N}_{LR}$.

Tensor $\mathcal{N}_{LR}$ has the following properties: first, it is impossible to choose a gauge in such a way that this tensor is the identity matrix for all links at the same time. The reason is that an iPEPS does not have a canonical form in the same sense as MPS. Second, this tensor needs to be computed using approximations, which implies that, generally, it is neither strictly Hermitian nor positively defined. Although one could always apply some approximation methods that preserve positivity explicitly (such as the single-layer method [38]), the tensors obtained from such approaches do not produce results as accurate as those of other methods that do not enforce positivity (such as CTM methods) [69]. It is well known that the (normally small) negative part of the norm tensor often causes some ill-posed conditions in updating the iPEPS tensors. To circumvent this problem, we can make it Hermitian and positively defined using the following steps [45]: first, we approximate the norm tensor as $\tilde{\mathcal{N}}_{LR} = (\mathcal{N}_{LR} + \mathcal{N}_{LR}^\dagger)/2$, which is Hermitian. Next, we approximate $\tilde{\mathcal{N}}_{LR}$ by its positive part. To achieve this, we apply the eigenvalue decomposition $\tilde{\mathcal{N}}_{LR} = W \Sigma W^\dagger$ and replace the (small) negative eigenvalues in Σ by zero. This is the so-called *positive approximant*. The approximate eigenvalues are now denoted Σ_+. Moreover, $\tilde{\mathcal{N}}_{LR} = Z Z^\dagger$, where $Z = W \Sigma_+^{1/2}$ [see Fig. 8(a)].

The gauge fixing that we apply here is explained in Fig. 8. After fixing the local gauge in the norm tensor, we replace $\tilde{\mathcal{N}}_{LR}$ and compatible subtensors in Eq. (15) and then start the variational update of subtensors for the iPEPS. As shown in Ref. [45], this choice of gauge greatly improves the conditioning of the norm tensor and thus greatly increases the
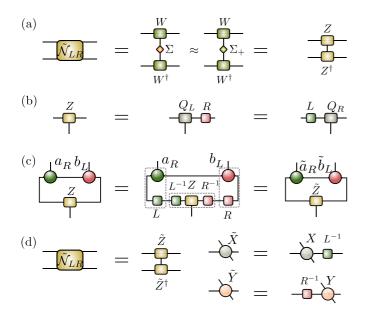
FIG. 8. (Color online) (a) Positive approximant for the norm tensor $\tilde{\mathcal{N}}_{LR}$. (b) Apply the QR decomposition for the left and the right bond indices of tensor Z, so that $Z = Q_L R = L Q_R$. (c) Insert the identities $I = L^{-1}L = R^{-1}R$ into the left and right bond indices of the tensor Z to obtain $\tilde{Z} = L^{-1}Z R^{-1}$ and the new subtensors $\tilde{a}_R = La_R, \tilde{b}_L = b_L R$. (d) A new norm tensor is obtained as $\tilde{\mathcal{N}}_{LR} = \tilde{Z}\tilde{Z}^\dagger$. In order to keep the compatibility once the subtensors have been updated, we need to multiply the tensors X and Y by matrices L^{-1}, R^{-1}, so that $\tilde{X} = XL^{-1}$ and $\tilde{Y} = R^{-1}Y$, respectively. This is done before recovering the tensors $\tilde{A} = \tilde{X}\tilde{a}_R$ and $\tilde{B} = \tilde{b}_L \tilde{Y}$.

stability of the update. In particular, when this gauge fixing is combined with the FFU, the resulting iPEPS algorithm is remarkably fast, stable, and accurate.

IV. RESULTS

We have benchmarked the improved iPEPS algorithm, including both the FFU and the gauge fixing, by studying ground-state properties of two models on an infinite 2D square lattice. The first model is the ferromagnetic quantum Ising model with transverse magnetic field,

$$H = -\sum_{\langle \vec{r}, \vec{r}' \rangle} \sigma_z^{[\vec{r}]} \sigma_z^{[\vec{r}']} - h \sum_{\vec{r}} \sigma_x^{[\vec{r}]}, \qquad (16)$$

where $\sigma_i^{[\vec{r}]}$ is the $i = (x,z)$ Pauli matrix for site $\vec{r}$, h is the transverse magnetic field, and $\langle \vec{r}, \vec{r}' \rangle$ represent nearest-neighbor sites. The second example is the spin-1/2 antiferromagnetic Heisenberg model,

$$H = \sum_{\langle \vec{r}, \vec{r}' \rangle} \vec{S}^{[\vec{r}]} \vec{S}^{[\vec{r}']}, \qquad (17)$$

where $\vec{S}^{[\vec{r}]} = (\sigma_x^{[\vec{r}]}, \sigma_y^{[\vec{r}]}, \sigma_z^{[\vec{r}]})/2$.

In our simulations, we have represented the ground state of the system by a two-site translationally invariant iPEPS made up of two tensors, A and B. In order to approximate the ground state of the system, we have applied imaginary-time evolution together with a second-order Suzuki-Trotter decomposition using δ down to 10^{-3}–10^{-4}. To update the tensors, we used

the alternating-least-squares (ALS) sweep for the subtensors, as explained in Sec. II B 4, combined with the FFU. At every update, we have also fixed the gauge of the tensors according to the gauge-fixing described above. Let us stress here that the leading order of the computational cost is the same for both FU and FFU plus gauge-fixing schemes, but the prefactor and the subleading correction are different. These turn out to produce a big difference in practical running times, as we shall see.

In order to assess the advantage of the local gauge fixing for the norm matrix $\tilde{\mathcal{N}}_{LR}$ used in the ALS sweep, we first compare the mean condition numbers of this matrix obtained with and without the gauge. As a rule of thumb, the larger the condition number of $\tilde{\mathcal{N}}_{LR}$ is, the less accuracy we get in solving the system of linear equations at every step of the ALS. The result is shown in Table I for different models and bond dimensions. Overall, we see that the condition number of the norm matrix in the case of gauge fixing is improved by several orders of magnitude when compared to the case without gauge fixing. For completeness, we also shown in Table I the condition numbers of matrices L and R. The gauge fixing thus improves the stability of the iPEPS algorithm. This result is very similar to what has been obtained for finite PEPS in a similar context [45].

In addition, we observe that the gauge fixing in the norm matrix also accelerates the convergence in the ALS sweeping. More concretely, in Fig. 9 we show the convergence of the relative change of the cost function defined in Eq. (15), as well as the local fidelity of the subtensors between two iterations u and $u + 1$ defined as

$$f_{ab}^{u+1} = \frac{\left(a_R^{u+1} b_L^{u+1} \big| a_R^u b_L^u\right)}{\sqrt{\left(a_R^{u+1} b_L^{u+1} \big| a_R^{u+1} b_L^{u+1}\right)\left(a_R^u b_L^u \big| a_R^u b_L^u\right)}}, \qquad (18)$$

where, e.g., $|a_R^u b_L^u)$ is to be understood as tensors a_R^u and b_L^u with their indices reshaped as a vector. We observe that, when the gauge fixing is applied, both cost function and local fidelity tend to converge faster than in the case without gauge fixing (see Fig. 9). We also noted that speedup in convergences becomes more significant as the bond dimension D of the iPEPS increases (not shown).

We also compute the order parameter $m_z \equiv \langle \sigma_z \rangle$ for the quantum Ising model and compare it for the cases with and without gauge fixing (see Fig. 10). The results for both cases are quantitatively very similar deep in the gapped phases. However, when close to the quantum critical point, the results obtained with the iPEPS plus gauge fixing are better, and again, we see that this effect becomes more relevant for larger bond dimensions (e.g., we almost see no difference for $D = 2$, whereas for $D = 3$ we already see a clear difference). In addition, in Fig. 11 we have plotted the correlation function $S_{zz}(x) = \langle \sigma_z^l \sigma_z^{l+x} \rangle - \langle \sigma_z^l \rangle \langle \sigma_z^{l+x} \rangle$ at the critical point $h = 3.044$. We see that the simulation with gauge fixing captures the correlation of the system better for the case of bond dimensions $(D, \chi) = (3,30)$.

We have also applied the improved iPEPS algorithm to study the ground state of the Heisenberg model for different bond dimensions (D, χ). For small bond dimensions D, we do not see much difference for the results with and without gauge fixing. But for large bond dimensions we observe that the ground-state energy obtained using gauge

TABLE I. Mean condition number of the norm matrix $\tilde{\mathcal{N}}_{LR}$ as well as of matrices L and R with their standard deviations. The numbers are for the case of the positive approximant, with and without fixing the gauge, for the subtensor FFU. We show different bond dimensions (D,χ) for an iPEPS algorithm with time step $\delta = 0.02$. The mean is computed over ten time steps for Heisenberg and Ising models. The transverse field for the Ising model is $h = 3.01$, which is close to criticality.

(D,χ)	No gauge	Gauge	R	L
Heisenberg model				
(2, 20)	$(9.33 \pm 1.50) \times 10^2$	1.46 ± 0.01	5.49 ± 0.23	5.29 ± 0.39
(3, 30)	$(1.6 \pm 0.17) \times 10^6$	$(0.15 \pm 0.00) \times 10^2$	$(0.41 \pm 0.01) \times 10^2$	$(0.45 \pm 0.01) \times 10^2$
w(4, 40)	$(8.36 \pm 0.30) \times 10^4$	$(3.02 \pm 0.02) \times 10^1$	$(1.78 \pm 0.00) \times 10^1$	$(1.80 \pm 0.02) \times 10^1$
(5, 50)	$(3.46 \pm 0.16) \times 10^6$	$(1.04 \pm 0.08) \times 10^3$	$(5.04 \pm 0.06) \times 10^1$	$(5.03 \pm 0.06) \times 10^1$
(6, 70)	$(1.01 \pm 0.30) \times 10^7$	$(2.21 \pm 0.03) \times 10^4$	$(5.12 \pm 0.15) \times 10^1$	$(5.05 \pm 0.22) \times 10^1$
Ising model				
(2, 20)	$(3.44 \pm 1.85) \times 10^4$	5.97 ± 0.09	$(1.29 \pm 0.13) \times 10^1$	$(1.29 \pm 0.14) \times 10^1$
(3, 30)	$(4.28 \pm 0.59) \times 10^6$	$(1.95 \pm 0.24) \times 10^3$	$(5.00 \pm 0.34) \times 10^1$	$(5.01 \pm 0.32) \times 10^1$
(4, 30)	$(1.28 \pm 1.52) \times 10^{19}$	$(5.81 \pm 11.88) \times 10^{17}$	$(5.39 \pm 0.13) \times 10^2$	$(5.47 \pm 0.14) \times 10^2$

fixing is better [see Fig. 12(a)]. To quantify the overall error, we compare our results with the best result obtained from quantum Monte Carlo [70], $\epsilon_0 = -0.669437(5)$. In our case, with gauge fixing we obtain $\epsilon_g = -0.669309(2)$, and without gauge we get $\epsilon_{ng} = -0.669243(1)$ for $(D,\chi) = (7,70)$. We also compute the staggered magnetization m as a function of the bond dimension, shown in Fig. 12(b). Again, for large bond dimensions the calculations with the gauge fixing become better than those without the gauge. Our best values were obtained for $(D,\chi) = (7,70)$ and are $m_g = 0.33490$ with

gauge and $m_{ng} = 0.33662$ without gauge. This should be compared with the Monte Carlo result, $m_0 = 0.30703$ [70].

Finally, even if not mentioned explicitly at every calculation, we recall that our results are obtained by using the FFU, which is remarkably faster than the costly FU update because the environment is not recomputed from scratch at every step. In Fig. 13 we compare the actual running times for the FU and the FFU (plus gauge fixing) schemes. The speedup factor increases with increasing D since more CTM steps are required to reach convergence of the environment tensors if the state is more entangled. In the present example we obtained a
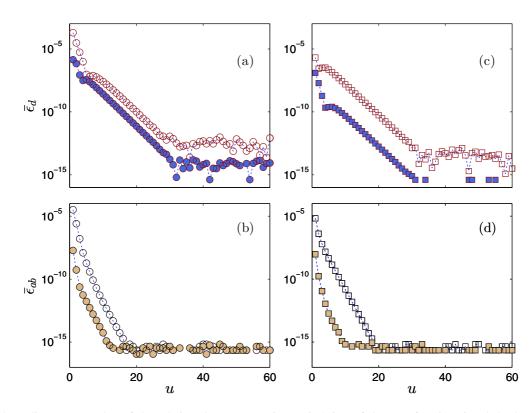

FIG. 9. (Color online) Mean value of the relative change $\bar{\epsilon}_d = |d_{u+1} - d_u|/|d_{\text{init}}|$ of the cost function d and the local fidelity $\bar{\epsilon}_{ab} = |f_{ab}^{u+1} - f_{ab}^u|/|f_{ab}^{\text{init}}|$ in the iPEPS algorithm with gauge (solid symbols) or without gauge (open symbols) for (a) and (b) the Ising model with transverse field and (c) and (d) the Heisenberg model, respectively. For the Ising model the magnetic field is $h = 3.01$ and $(D,\chi) = (3,30)$, whereas for the Heisenberg model $(D,\chi) = (5,50)$. The time step is $\delta = 0.005$. The mean value is taken over all steps in imaginary time.
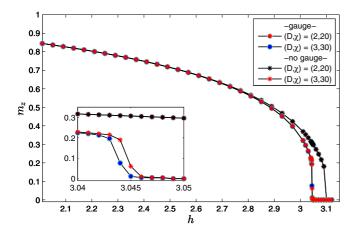
FIG. 10. (Color online) Local magnetization m_z as a function of the transverse field h in the quantum Ising model, with gauge (solid circles) and without gauge (stars). The inset is a zoom around criticality.

speedup factor of up to ~ 30. This factor will be even larger in more strongly entangled systems (e.g., in fermionic systems). In combination with the gauge fixing the FFU approach is even better: we have seen that the overall improved algorithm is remarkably stable, as well as substantially faster than the old version.

One more comment is in order: one could naively expect that after several updates with the FFU scheme the environment will have drifted away, so that it may need to be reinitialized by fully converging it. However, we find that the new update scheme is self-correcting for the considered values of δ and for the studied models (indeed, if δ is small enough, the changes in the tensors are also expected to be small). One may expect, however, that for larger δ the method is no longer self-correcting. This is a possibility that needs to be taken into account when implementing the algorithm in practice. But in any case, δ decreases throughout the evolution of the algorithm, so the self-correction happens naturally, and in practice we never need to restart. For the models analyzed in our paper we never encounter such a situation, but we can imagine that for
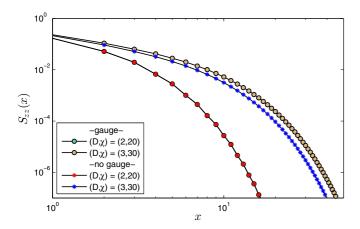

FIG. 11. (Color online) Two-point correlator $S_{zz}(x)$ of the quantum Ising model close to the critical point $h = 3.044$, computed with gauge (solid circles) and without gauge (stars).
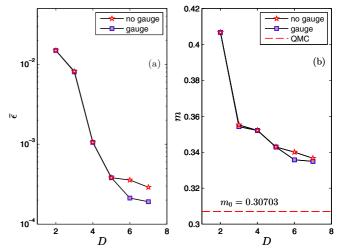

FIG. 12. (Color online) (a) Relative error of the energy per link $\bar{\epsilon} = (\epsilon_0 - \epsilon)/\epsilon_0$ for the Heisenberg model as a function of the iPEPS bond dimension D. Here $\epsilon_0 = -0.669437(5)$ is the quantum Monte Carlo result [70]. (b) Staggered magnetization m of the Heisenberg model as a function of the bond dimension D, compared to the Monte Carlo result $m_0 = 0.30703$ [70].

more complex systems one may need to restart the environment from time to time in order to improve the results. This may also be an important difference between imaginary- and real-time evolutions. For imaginary evolutions, the algorithm is naturally self-correcting since in the limit $\delta \to 0$ the actual time steps behave, in practice, as convergence steps for the environment. However, this property may be lost for real-time evolutions if δ is not small enough, so that several iterations may be needed at every step. But in any case, for real-time evolutions it is also important to recycle environment tensors at every iteration in order to save time, as well as to take care of correctly matching all bond indices as done explicitly in the FFU.
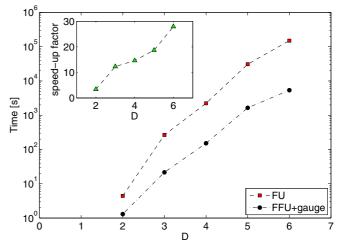

FIG. 13. (Color online) Total running time (in seconds) for five time steps in imaginary-time evolution with the Heisenberg Hamiltonian, with the FU and FFU (plus gauge fixing) updates. The speedup factor of the new update scheme with respect to the old one is shown in the inset.

V. CONCLUSIONS

In this paper we have presented how to improve the stability and efficiency of the iPEPS algorithm. We have discussed two improvements, namely, the fast full update and the gauge fixing in the ALS sweep. In the FFU scheme the tensors in the effective environment are updated at every step while keeping them compatible with the updated iPEPS tensors. This implies a speedup by a large factor (up to ∼30 in the present examples and even larger in more strongly entangled systems) compared to the previous FU approach, where the environment tensors are recomputed from scratch at each time step. The gauge fixing improves the conditioning in the ALS sweep at every update step in the same way as already shown for finite PEPS calculations [45], leading to better stability and faster convergence.

We have benchmarked the improved iPEPS algorithm with calculations for the quantum Ising and Heisenberg models on an infinite 2D square lattice, where we have seen that similar or slightly better accuracies can be obtained substantially faster and with more stable evolutions than in previous iPEPS calculations. This is particularly true for large bond dimensions and in highly entangled systems, such as in the vicinity of quantum critical points.

Technically, we have demonstrated the improved iPEPS algorithm for systems on an infinite square lattice and a two-site unit cell, but the extension to other 2D lattices and bigger unit cells is straightforward. The method can also be extended to Hamiltonians with longer-range interactions. We expect that these improvements will be a significant step forward towards powerful tensor network calculations for challenging 2D systems.

[1] R. Orús, Ann. Phys. (N.Y.) **349**, 117 (2014); Eur. Phys. J. B **87**, 280 (2014); J. Eisert, Model. Simul. **3**, 520 (2013); N. Schuch, *Quantum Information Processing*, Lecture Notes of the 44th IFF Spring School (Forschungszentrum, Jülich, 2013); J. I. Cirac, and F. Verstraete, J. Phys. A **42**, 504004 (2009); F. Verstraete, J. I. Cirac, and V. Murg, Adv. Phys. **57**, 143 (2008).

[2] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, Phys. Rev. Lett. **90**, 227902 (2003).

[3] P. Calabrese and J. Cardy, J. Stat. Mech (2004) P06002.

[4] M. Srednicki, Phys. Rev. Lett. **71**, 666 (1993).

[5] M. B. Plenio, J. Eisert, J. Dreißig, and M. Cramer, Phys. Rev. Lett. **94**, 060503 (2005).

[6] A. Riera and J. I. Latorre, Phys. Rev. A **74**, 052326 (2006).

[7] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[8] G. Vidal, Phys. Rev. Lett. **93**, 040502 (2004).

[9] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, Phys. Rev. Lett. **106**, 170501 (2011).

[10] M. Fannes, B. Nachtergaele, and R. F. Werner, Commun. Math. Phys. **144**, 443 (1992).

[11] S. Ostlund and S. Rommer, Phys. Rev. Lett. **75**, 3537 (1995).

[12] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[13] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[14] H. Takasaki, T. Hikihara, and T. Nishino, J. Phys. Soc. Jpn. **68**, 1537 (1999).

[15] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005).

[16] D. Pérez-García, F. Verstraete, M. M. Wolf, and J. I. Cirac, Quantum Inf. Comput. **7**, 401 (2007).

[17] I. P. McCulloch, J. Stat. Mech. (2007) P10014.

[18] I. P. McCulloch, arXiv:0804.2509.

[19] G. M. Crosswhite, A. C. Doherty, and G. Vidal, Phys. Rev. B **78**, 035116 (2008).

[20] U. Schollwöck, Ann. Phys. (N.Y.) **326**, 96 (2011).

[21] S. R. White and A. E. Feiguin, Phys. Rev. Lett. **93**, 076401 (2004).

[22] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal, J. Stat. Mech. (2004) P04005.

[23] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pizorn, H. Verschelde, and F. Verstraete, Phys. Rev. Lett. **107**, 070601 (2011).

[24] Other generalizations are possible using, e.g., tree tensor networks [71] and the multiscale entanglement renormalization ansatz [72].

[25] F. Verstraete and J. I. Cirac, arXiv:cond-mat/0407066.

[26] V. Murg, F. Verstraete, and J. I. Cirac, Phys. Rev. A **75**, 033605 (2007).

[27] F. Verstraete, M. M. Wolf, D. Pérez-García, and J. I. Cirac, Phys. Rev. Lett. **96**, 220601 (2006).

[28] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, and Y. Akutsu, Nucl. Phys. B **575**, 504 (2000).

[29] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, Y. Akutsu, and A. Gendiar, Prog. Theor. Phys. **105**, 409 (2001).

[30] A. Gendiar, N. Maeshima, and T. Nishino, Prog. Theor. Phys. **110**, 691 (2003).

[31] M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).

[32] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **101**, 250602 (2008).

[33] R. Orús and G. Vidal, Phys. Rev. B **80**, 094403 (2009).

[34] H. C. Jiang, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **101**, 090603 (2008).

[35] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **103**, 160601 (2009).

[36] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Phys. Rev. B **81**, 165104 (2010).

[37] L. Wang, I. Pižorn, and F. Verstraete, Phys. Rev. B **83**, 134421 (2011).

[38] I. Pižorn, L. Wang, and F. Verstraete, Phys. Rev. A **83**, 052321 (2011).

[39] W. Li, J. von Delft, and T. Xiang, Phys. Rev. B **86**, 195137 (2012).

[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] H. Kalis, D. Klagges, R. Orús, and K. P. Schmidt, Phys. Rev. A **86**, 022317 (2012).

[42] R. Orús, H. Kalis, M. Bornemann, and K. P. Schmidt, Phys. Rev. A **87**, 062312 (2013).

[43] M. Lubasch, J. Ignacio Cirac, and M.-C. Bañuls, New J. Phys. **16**, 033014 (2014).

[44] L. Wang and F. Verstraete, arXiv:1110.4362.

[45] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Phys. Rev. B **90**, 064425 (2014).

[46] H. N. Phien, I. P. McCulloch, and G. Vidal, Phys. Rev. B **91**, 115137 (2015).

[47] B. Bauer, P. Corboz, R. Orús, and M. Troyer, Phys. Rev. B **83**, 125106 (2011).

[48] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).

[49] R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008).

[50] B. Bauer, G. Vidal, and M. Troyer, J. Stat. Mech. (2009) P09006.

[51] P. Corboz, S. R. White, G. Vidal, and M. Troyer, Phys. Rev. B **84**, 041108 (2011).

[52] P. Corboz, T. M. Rice, and M. Troyer, Phys. Rev. Lett. **113**, 046402 (2014).

[53] Z.-C. Gu, H.-C. Jiang, D. N. Sheng, H. Yao, L. Balents, and X.-G. Wen, Phys. Rev. B **88**, 155112 (2013).

[54] L. Wang, Z.-C. Gu, F. Verstraete, and X.-G. Wen, arXiv:1112.3331.

[55] P. Corboz and F. Mila, Phys. Rev. B **87**, 115144 (2013); Y. H. Matsuda, N. Abe, S. Takeyama, H. Kageyama, P. Corboz, A. Honecker, S. R. Manmana, G. R. Foltin, K. P. Schmidt, and F. Mila, Phys. Rev. Lett. **111**, 137204 (2013); P. Corboz and F. Mila, *ibid.* **112**, 147203 (2014).

[56] M. Ziegler and R. Orús (unpublished).

[57] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang, Phys. Rev. B **81**, 174411 (2010).

[58] R. Orús, Phys. Rev. B **85**, 205117 (2012).

[59] R. J. Baxter, J. Math. Phys. **9**, 650 (1968).

[60] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **65**, 891 (1996).

[61] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **67**, 3066 (1998).

[62] P. Corboz, J. Jordan, and G. Vidal, Phys. Rev. B **82**, 245119 (2010).

[63] G. Evenbly and G. Vidal, arXiv:1412.0732.

[64] $D = 1$ is a product state, whereas larger $D > 1$ correspond to entangled states. The larger D is, the larger the amount of entanglement in the wave function is, as quantified by the prefactor to the entanglement area law [1].

[65] Other 2D lattices can also be easily considered.

[66] M. Suzuki, Phys. Lett. A **146**, 319 (1990); J. Math. Phys. (Melville, N.Y.) **32**, 400 (1991); For higher order expansions, see also A. T. Sornborger and E. D. Stewart, (Melville, N.Y.) **60**, 1956 (1999).

[67] A good initialization of $\tilde{A}$ and $\tilde{B}$ can be obtained by applying the following procedure. First, contract the tensors A and B along the link of interest together with the two-body gate g. Second, apply a singular value decomposition of the obtained tensor separating the links associated with each lattice site. Finally, truncate the common link up to D values, which leads to a good candidate for $\tilde{A}$ and $\tilde{B}$.

[68] Formally speaking, this means that the FFU is not the "correct" update in the sense that the environment tensors are not fully converged. In fact, it does not even correspond to the full first-order correction in perturbation theory for the environment. Still, the point of this paper is to see that the FFU is already sufficient for practical purposes.

[69] In fact, even more fundamental, it has also been shown that some positive matrix product operators with finite bond dimension do not have a representation in terms of a matrix product density operator with finite bond dimension (see Ref. [73]).

[70] A. W. Sandvik, Phys. Rev. B **56**, 11678 (1997).

[71] L. Tagliacozzo, G. Evenbly, and G. Vidal, Phys. Rev. B **80**, 235127 (2009).

[72] G. Evenbly and G. Vidal, Phys. Rev. Lett. **102**, 180406 (2009).

[73] G. de las Cuevas, N. Schuch, D. Pérez-García, and J. I. Cirac, New J. Phys. **15**, 123021 (2013).