Unifying projected entangled pair state contractions

Michael Lubasch, J Ignacio Cirac and Mari-Carmen Bañuls

Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Straße 1, D-85748 Garching, Germany

E-mail: michael.lubasch@mpq.mpg.de

Abstract

The approximate contraction of a tensor network of projected entangled pair states (PEPS) is a fundamental ingredient of any PEPS algorithm, required for the optimization of the tensors in ground state search or time evolution, as well as for the evaluation of expectation values. An exact contraction is in general impossible, and the choice of the approximating procedure determines the efficiency and accuracy of the algorithm. We analyze different previous proposals for this approximation, and show that they can be understood via the form of their environment, i.e. the operator that results from contracting part of the network. This provides physical insight into the limitation of various approaches, and allows us to introduce a new strategy, based on the idea of clusters, that unifies previous methods. The resulting contraction algorithm interpolates naturally between the cheapest and most imprecise and the most costly and most precise method. We benchmark the different algorithms with finite PEPS, and show how the cluster strategy can be used for both the tensor optimization and the calculation of expectation values. Additionally, we discuss its applicability to the parallelization of PEPS and to infinite systems.

Keywords: computational physics, condensed matter: electrical, magnetic and optical, quantum information and quantum mechanics

1. Introduction

In the past few years, tensor network states (TNS) have been revealed as a very promising choice for the numerical simulation of strongly correlated quantum many-body systems. A sustained effort has led to significant conceptual and technical advancement of these methods, e.g. in [1–22].

In the case of one-dimensional systems, matrix product states (MPS) are the variational class of TNS underlying the density matrix renormalization group (DMRG) [23]. Insight gained from quantum information theory has allowed the understanding of DMRG's enormous success at approximating ground states of spin chains, and the extension of the technique to dynamical problems [1–6] and lattices of more complex geometry [7–9].

Projected entangled pair states (PEPS) [7] constitute a family of TNS that naturally generalizes MPS to spatial dimensions larger than 1 and arbitrary lattice geometry. Like MPS, PEPS incorporate the area law by construction, which makes them a very promising variational ansatz for strongly correlated systems which might not be tractable by other means, e.g. frustrated or fermionic states where quantum Monte Carlo methods suffer from the sign problem. Although originally defined for spin systems, PEPS have been subsequently formulated for fermions [24–27], and their potential in the numerical simulation of fermionic phases has been demonstrated [27–30]. But in contrast to the case for MPS, even local expectation values cannot be computed exactly in the case of PEPS. This is because the exact evaluation of the TN that represents the observables has an exponential cost in the system size. The same difficulty affects the contraction of the TN that surrounds a given tensor, the so-called *environment*, required for the local update operations in the course of optimization algorithms. It is nevertheless possible to perform an approximate TN contraction with controlled error, albeit involving a much higher computational cost than in the case of MPS. This limits the feasible PEPS simulations to relatively small tensor dimensions.

Lately, several algorithmic proposals have emerged [21, 31–33] that make larger tensors accessible by using new approximations in the PEPS contraction. Although these approaches allow the manipulation of a larger set of PEPS, their assumptions have an impact on the accuracy of the ground state approximation, and this accuracy is not always directly related to the maximum bond dimension that the algorithm can accommodate. It is nevertheless possible to analyze the various approximations from the unifying point of view of how they treat the environment contraction, which in turn has a physical meaning. This allows us to understand how a given strategy may attain only a limited precision approximation to the ground state, even when its computational cost allows for large bond dimensions.

Contraction strategies proposed in the literature include the original PEPS method [7], the 'Simple Update' [21][1] and the 'Single-Layer' [31] algorithm. In this work we investigate these algorithms from the unifying perspective introduced above, and present a new contraction scheme that naturally interpolates between the cheapest and most imprecise method and the most expensive and precise one. We illustrate our findings with finite-size PEPS with open boundary conditions. A finite PEPS, in which each tensor contributes independent variational parameters, is less biased than its infinite counterpart iPEPS, in which a unit cell of variational

---

[1] In the original proposal [21] the Simple Update denotes not a contraction strategy but a tensor update procedure for imaginary time evolution. However, the environment used in this update corresponds also to a certain contraction method, as we will show later.

tensors is replicated infinitely often and the form of that unit cell can have an effect on the observed order. However, all our results apply also to iPEPS, and, as we will argue, provide the basis for a new promising approach in that context.

This article is structured as follows. In section 2 we briefly introduce the basic PEPS concepts and original algorithms. Section 3 reviews the Simple Update method introduced in [21], and analyzes its performance with finite PEPS. We find that the resulting ground state energies can be less accurate than those of the original algorithm when the environment form assumed in the method is far from the true one. The Single-Layer algorithm proposed in [31] can be seen as a first, conceptual generalization of the Simple Update, and we investigate it in section 4. We show that the error introduced by this method exhibits a strong system size dependence, in contrast to the original algorithm. With the insight gained, in section 5 we formulate and investigate a new strategy for the environment approximation based on the idea of clusters, that is applicable to both the tensor update and the computation of expectation values. Furthermore, we discuss how this cluster strategy is also beneficial to the parallelization of PEPS as well as to the infinite case, i.e. iPEPS. Finally in section 6 we briefly summarize our results.

2. PEPS: basic concepts and algorithms

We consider a system of N quantum particles with Hilbert spaces of dimensions d_l, for $l = 1, \ldots, N$, spanned by individual bases $\{| s_l \rangle\}$, with $s_l = 1, \ldots, d_l$. Projected entangled pair states (PEPS) [7] are states for which the coefficients in the product basis are given by the contraction of a tensor network,

$$|\psi^{PEPS}\rangle := \sum_{s_1, s_2, \ldots, s_N} \mathcal{F}\left(A_1^{s_1} A_2^{s_2} \ldots A_N^{s_N}\right) |s_1 s_2 \ldots s_N\rangle,$$

with one tensor A_l per physical site. The tensors A_l are arranged in a certain lattice geometry and connected to neighboring sites by shared indices, such that the coordination number, c, of a certain lattice site coincides with the number of connecting indices. The latter are called virtual, and each tensor A_l possesses, apart from them, one physical index s_l, standing for the physical degree of freedom of the quantum particle on lattice site l. The function $\mathcal{F}$ represents the contraction of all virtual indices. Each of them ranges up to the parameter D which is named the bond dimension. D determines the number of variational parameters of each tensor, namely dD^c.[2] The bond dimension sets an upper bound to the entanglement entropy of the state, in fulfillment of the area law. In particular, if we consider a subsystem delimited by a regular shape of side length ℓ, the entropy of its reduced density matrix, ρ_ℓ, is upper bounded by $S(\rho_\ell)_{\max} \propto \ell^{\dim-1} \log(D)$, where dim denotes the system's dimensionality. Throughout this work, we consider PEPS on two-dimensional square lattices of size $N = L \times L$ with side length L and open boundary conditions. An example is shown in figure 1.

For general PEPS, the computation of an expectation value or even the norm is known to be hard [34], like the evaluation of a two-dimensional classical partition function [35]. Hence only an approximate contraction is possible for already moderate lattice sizes. The originally

---

[2] In the case of open boundary conditions the tensors on the boundaries have fewer virtual indices and variational parameters.
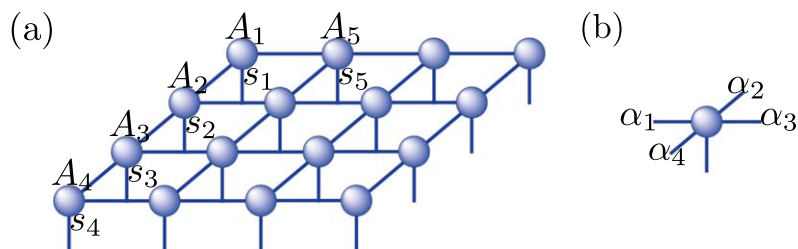
Figure 1. (a) A 4×4 PEPS $|\psi^{PEPS}\rangle := \sum_{s_1,s_2,\ldots,s_{16}} \mathcal{F}(A_1^{s_1} A_2^{s_2} \ldots A_{16}^{s_{16}}) |s_1 s_2 \ldots s_{16}\rangle$ on a square lattice. (b) A tensor from the interior with four virtual indices α_1 to α_4. The function $\mathcal{F}$ performs the contraction of the tensor network by summing over connected virtual indices.
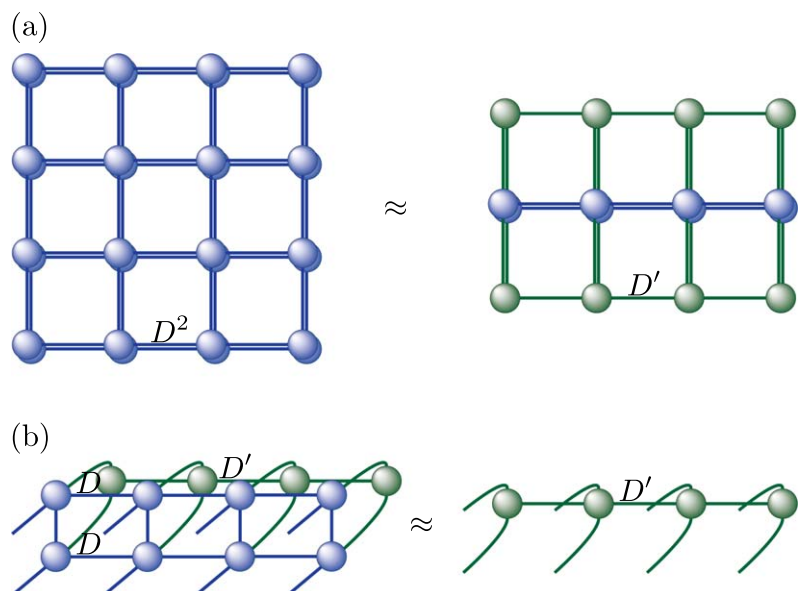
Figure 2. Sandwich contraction of the norm $\langle\psi|\psi\rangle$. (a) The TN corresponding to $\langle\psi|\psi\rangle$ results from figure 1(a) as a ket and as bra and a contraction over the physical indices (left). The hard computation of this two-dimensional PEPS sandwich is approximated by an efficiently contractible one-dimensional MPS expectation value (right). (b) This is done by successively approximating the action of a bulk row of the sandwich, of bond dimension D, on a boundary MPO, of bond dimension D' (left), by a new boundary MPO (right). The latter can be determined with computational cost $O(D^4 D'^3) + O(dD^6 D'^2)$.

proposed algorithm [7] approximates the two-dimensional TN of an expectation value $\langle\psi|\hat{O}|\psi\rangle$ or the norm $\langle\psi|\psi\rangle$ by means of a succession of one-dimensional MPS contractions, as sketched in figure 2(a) for the norm. In the following we refer to this original method by the term *sandwich contraction*. The procedure starts by identifying two opposite sides of the TN, e.g. the uppermost and bottommost rows, with MPS, and each of the intermediate rows with a matrix product operator (MPO) [36]. Beginning from one of the edges, the contraction of the last row with the immediately neighboring one is then an MPS–MPO product, which can be optimally

approximated by a MPS of fixed bond dimension, D'. By repeating the procedure from both opposite sides, successive MPS–MPO approximations lead to a representation of both halves of the TN by MPS. Finally the row in the center is contracted between the two MPS to give the approximate expectation value or norm.

At each point of this procedure, the MPS obtained approximates the boundary between the contracted part of the network and the rest. This MPS can be interpreted as an operator that maps the virtual indices of the ket boundary to the bra and thus we will refer to it as the *boundary MPO*, shown in figure 2(b).[3] The approximate contraction of the norm has the leading cost $O(D^4D'^3) + O(dD^6D'^2)$, and thus both cost and error are determined solely by the bond dimension D' of the boundary MPO. Although in principle D' could scale exponentially with the number of rows, in practice it typically scales as $D' \propto D^2$ independently of the system size, such that the original sandwich contraction has the total computational cost $O(D^{10})$.

For certain problems, this observed mild scaling can be given more rigorous grounds. Indeed, the boundary MPO can be interpreted as the thermal state of a Hamiltonian defined on the virtual degrees of freedom of the boundary. This boundary Hamiltonian is obtained by identifying its excitation spectrum with the entanglement spectrum of the state [37]. Such a construction is very natural in the framework of PEPS and establishes a holographic principle [38]. While PEPS are expected to represent the low energy sector of local Hamiltonians well, it has not been proven when expectation values can be computed efficiently with them. However, if the boundary Hamiltonian is local, as evidence suggests for gapped models [38], the corresponding thermal state will be efficiently approximated by a MPO [39].

In the following, we obtain the PEPS approximation to the ground state of a certain Hamiltonian by means of imaginary time evolution. It is based on the idea that $e^{-t\hat{H}}\left|\psi_0\right\rangle$ converges to the ground state of H exponentially fast with t, as long as the ground state is not degenerate and has non-vanishing overlap with the initial state, $\left|\psi_0\right\rangle$. In the context of TNS [3], the initial state is chosen within the appropriate TNS family, and a Suzuki–Trotter decomposition of the evolution operator $U(t) = e^{-t\hat{H}} = (e^{-\tau\hat{H}})^n$ is applied to local Hamiltonians, such that each step of the evolution, $\tau = t/n$, is approximated by a product of local *Trotter gates*. The resulting state after each gate or set of gates is again approximated by an adequate TNS. In particular, the action of a certain operator $\hat{O}$ on a PEPS $|\phi\rangle$ can be approximated by a new PEPS $|\psi\rangle$ by minimizing the cost function $d(|\psi\rangle) = \||\psi\rangle - \hat{O}|\phi\rangle\|^2$. We perform this minimization for each gate via an alternating least squares (ALS) scheme, optimizing one tensor at a time while the others are fixed, and sweeping only over the tensors on which the Trotter gate acts. The optimal tensor at position l is the solution of a system of linear equations $N_l A_l = b_l$, where the norm matrix N_l is defined from the tensor network $\langle\psi|\psi\rangle$ by leaving out the tensor A_l in the ket and A_l^* in the bra, and the vector b_l results from the tensor network $\langle\psi|\hat{O}|\phi\rangle$ by removing A_l^* from the bra.

The environment of a tensor at site l is the open TN that results when this tensor and its adjoint are removed from the norm of the state. Contracting the environment is necessary for

---

[3] If the contraction was exact, this boundary MPO would always be positive in the case of the norm computation, due to the bra–ket structure of all the rows involved, but this positive character is in general lost in the truncation.
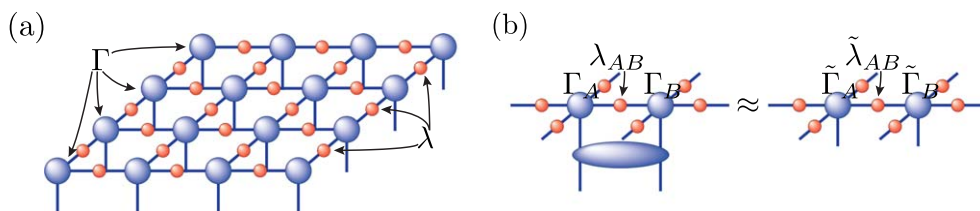
Figure 3. (a) A 4×4 PEPS $|\psi^{PEPS,SU}\rangle := \sum_{s_1,s_2,\ldots,s_{16}} \mathcal{F}(\Gamma_1^{s_1} \lambda_1 \ldots \Gamma_{16}^{s_{16}})|s_1 s_2 \ldots s_{16}\rangle$ of the Simple Update (SU) form, composed of Γ tensors and λ matrices. (b) Assuming nearest neighbor Trotter gates, the six λ matrices surrounding a tensor pair Γ_A and Γ_B are sufficient for the update of this pair and its λ_{AB}.

evaluating N_l and b_l, which determine the local equation for A_l. Such contraction can only be carried out approximately, and the approximation strategy is decisive both for the accuracy and for the computational cost of the algorithm[4]. Because we process the Trotter gates one after another and modify only the tensors on which the gate directly acts, in the following we focus on the contraction of the norm TN around a Trotter gate. The importance of the environment approximation has been recognized also in other works, e.g. [40], and in the different context of tensor renormalization group algorithms [20] where a more precise environment representation led to significant improvements [22].

3. Simple Update

The Simple Update method (SU) [21] directly generalizes the one-dimensional TEBD [1–4, 41] and proposes for the PEPS tensors the decomposition

$$|\psi^{PEPS,SU}\rangle := \sum_{s_1,s_2,\ldots,s_N} \mathcal{F}(\Gamma_1^{s_1} \lambda_1 \Gamma_2^{s_2} \lambda_2 \ldots \Gamma_N^{s_N})|s_1 s_2 \ldots s_N\rangle,$$

formally analogous to the canonical form for MPS [1], where the Γ_l are tensors with the same dimensions as the original A_l, and the λ_l are $D \times D$ diagonal and positive matrices; see figure 3. Although in the case of PEPS, the λ matrices do not have the clear meaning of their one-dimensional counterparts, the SU has proven to be a successful strategy in the context of iPEPS, starting with [21]. This success can be attributed, on one hand, to the low computational cost of the tensor update, which is why large values of D can be reached easily; indeed, the SU rule requires only the λ matrices that are closest to a tensor pair and as a consequence has the computational cost $O(D^5)$. On the other hand, all parts of the algorithm are well-conditioned. These positive aspects arise at the expense of an oversimplified representation of the environment as separable and local, that, in general, can only be a rough approximation of the true environment.

In order to illustrate its performance, we employ the SU to find ground states of the quantum Ising Hamiltonian with transverse field

---

[4] In general, also the tensor update operations contribute to the final cost. If we restrict the variational parameters to the reduced tensor [27, 32], the linear equations can be solved with $O(D^6)$ operations.
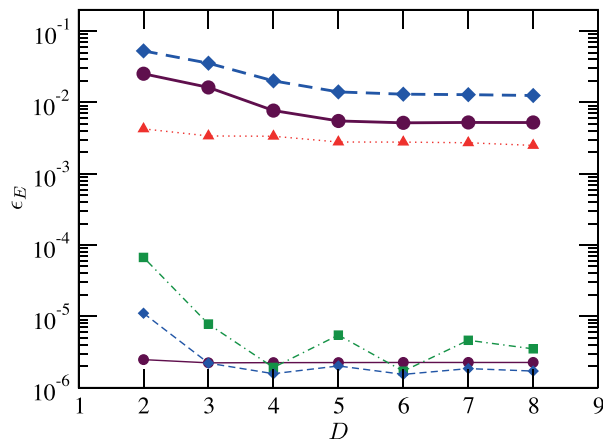
Figure 4. Relative energy error $\epsilon_E := \left| E(D) - E_0 \right| / \left| E_0 \right|$ of the SU, $E(D)$, with respect to the exact ground state energy, E_0. We consider the Ising model (thin lines) on a 4×4 lattice with $B = 1.0$ (solid), 2.0 (dotted), 3.0 (dash–dotted), and 4.0 (dashed), and the Heisenberg model (thick lines) on a 4×4 (dashed) and 10×10 (solid) lattice.

$$\hat{H} = - \sum_{\langle l,m \rangle} \sigma_l^z \otimes \sigma_m^z - B \sum_l \sigma_l^x \tag{1}$$

and of the Heisenberg model

$$\hat{H} = \sum_{\langle l,m \rangle} \vec{S_l} \cdot \vec{S_m}, \tag{2}$$

where $\langle l, m \rangle$ denotes pairs of neighboring sites l and m. In the context of finite PEPS considered here, to the best of our knowledge, the SU has not yet been used. We determine the ground state of a particular problem by evolving an initial state long enough in imaginary time and successively decreasing the time step τ until convergence is reached.

Figure 4 compares SU results to exact ground state energies. The scheme performs remarkably well for the Ising model at $B = 1.0$, 3.0, and 4.0, where the relative energy error is below 10^{-5} already with $D = 3$. But at $B = 2.0$ and for the Heisenberg Hamiltonian, we observe that the energy does not improve significantly beyond a certain value of the bond dimension, and remains far from the exact value. We identify this as a limitation, not of the ansatz, but of the update procedure, since the original PEPS algorithm [7] achieves already for the Heisenberg model on a 4×4 lattice with $D = 3$ lower energy than any of the SU values from the figure, and with $D = 4$ it attains an energy per site -0.5739 already very close to the exact value -0.5743. Although we observed that the SU result can depend on the initial state, in particular for the larger bond dimensions[5], this dependence appeared less strongly when we increased the bond dimension successively during imaginary time evolution.

---

[5] This became evident from running the algorithm with various values of D separately, each run starting from a product state in which the tensors' zeros were replaced by random numbers from the interval $[-0.01, 0.01]$. In this setting we observed that the SU can lead to a final $D = 8$ energy above the $D = 7$ value.
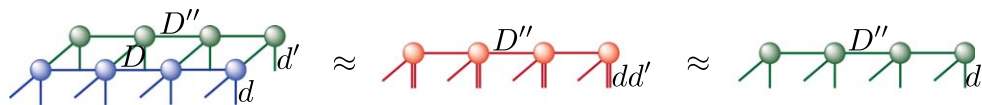
Figure 5. SL contraction of the norm $\langle\psi|\psi\rangle$. As explained in the text, the approximation of figure 2 is achieved by means of operations in the ket alone.

4. Single-Layer

The Single-Layer (SL) algorithm for the computation of the norm $\langle\psi|\psi\rangle$ was presented in [31]. This method takes into account the bra–ket structure of the sandwich, and maintains it and hence the positive character of the environment while the contraction of the network progresses from one edge. To achieve this structure, the approximate contraction proceeds by successive MPO–MPS operations, like the original algorithm, but this time performed on a single layer of the sandwich TN. Then the boundary, i.e. the already contracted part of the network, is always approximated by a purification MPO [5], namely the result of tracing out a part of the physical indices at every site of a MPS. This MPS is assumed to have some maximum bond dimension, D'', and physical dimensions $D \times d'$, where d' is the dimension of the traced-out degrees of freedom, which we call the *purification bond*. In this way, the local and separable environment defined by the λ matrices in the SU is generalized by means of purification MPS that can better capture non-local and non-separable boundary correlations. Moreover, the boundary purification MPO is always a positive operator, and it allows one to devise a stable tensor update procedure for imaginary time evolution [31].

The SL operations take place in the two steps shown in figure 5. First, the ket part of a PEPS row is applied as a MPO to the MPS of the boundary purification. The result is truncated to a MPS with bond dimension D'' and increased purification bond, dd'. Then the purification bond is reduced from dd' to d', by imposing the canonical form [1] and projecting the reduced density matrix of each site onto the space spanned by its d' largest eigenvectors. The computational cost of the first step, which proceeds via the standard ALS scheme, scales as $O(dd'D^4D''^2) + O(dd'D^2D''^3)$, while the leading cost of the second step is $O(d^2d'^2DD''^2)$, negligible only when d' is small. Because the purification bond satisfies $d' \leqslant DD''^2$, the maximum cost can at most grow as $O(dD^5D''^4) + O(dD^3D''^6)$ when d' takes on its largest possible value. In [31, 33] it was proposed to set $d' = D = D''$, in which case the number of operations scales only like $O(D^7)$, and a clear computational gain compared to the original contraction can be expected.

In order to analyze the performance of the SL procedure, we study the accuracy of the norm contraction, $\mathcal{N} \equiv \langle\psi|\psi\rangle$, as a function of the truncation parameters, D'' and d', for a set of different PEPS, and compare the results to those from the original algorithm. In particular, we consider $D = 2$–4 PEPS ground state approximations from the SU for the Ising model (1) with transverse fields $B = 1.0, 2.0, 3.0,$ and 4.0, on lattices with side lengths $L = 11$ and 21. In all cases, the exact norm was estimated by means of the sandwich contraction with bond dimension $D' = 100$, large enough to make the error negligible.

In the case of the original algorithm, the relative error always decreases exponentially with the bond dimension of the general boundary MPO, D'. Moreover, for a fixed bond dimension of the PEPS, D, this error shows no system size dependence. In the SL algorithm, for fixed
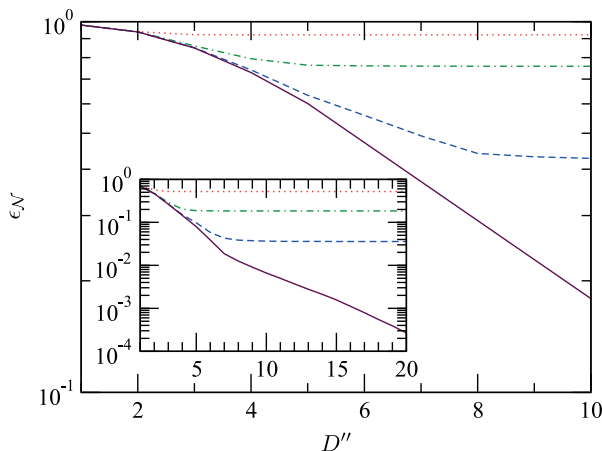
Figure 6. Relative norm error $\epsilon_\mathcal{N}$ in the SL contraction of a $D = 2$ SU ground state approximation of the Ising model with $B = 3.0$ on 11×11 (inset) and 21×21 (main) lattices. We consider $d' = 2$ (dotted), $d' = 4$ (dash–dotted), and $d' = 8$ (dashed), and the maximum possible $d' = DD''^2$ (solid). We defined $\epsilon_\mathcal{N} := |\mathcal{N}(D'') - \mathcal{N}_{ref}|/|\mathcal{N}_{ref}|$ where the reference value $\mathcal{N}_{ref}$ was computed with the sandwich contraction using $D' = 100$.

purification bond d', the contraction error converges quickly as a function of D'' to a final value that is entirely determined by d'. Even when that purification bond dimension takes on its maximum value, $d' = DD''^2$, this error lies many orders of magnitude above the one from the sandwich contraction with the same $D' = D''$. It is worth noticing that for large $D' = D'' \gg D$ the computational cost of the original method is actually lower than that of the SL algorithm with maximum $d' = DD''^2$.

The differences between the original and the SL contraction become even more apparent when the lattice size is increased to $N = 21 \times 21$, because the SL algorithm depends strongly on the system size as can be gathered from figure 6. In that case, given $d' = DD''^2$ and $D'' = 10$, the norm error grows from $\epsilon_\mathcal{N} \approx 0.007$ in the 11×11 lattice to $\epsilon_\mathcal{N} \approx 0.1$ in the 21×21 lattice, in marked contrast to $\epsilon_\mathcal{N} \approx 10^{-11}$ in the sandwich contraction with $D' = 10$ obtained for both lattice sizes. And we observe a similar scaling for PEPS with larger bond dimensions. For instance, the SL value for $D = 4$, $d' = 8$, and $D'' = 10$ grows from $\epsilon_\mathcal{N} \approx 0.06$ in the 11×11 lattice to $\epsilon_\mathcal{N} \approx 0.6$ in the 21×21 lattice, which has to be compared to $\epsilon_\mathcal{N} \approx 10^{-5}$ in the sandwich contraction with $D' = 10$ achieved for both lattice sizes.

From this analysis we conclude that the choice $d' = D = D''$, which ensures the advantageous computational cost $O(D^7)$, is in general too restrictive for getting a precision comparable to that of the original algorithm. Moreover, because the required values of the parameters d' and D'' for a certain contraction precision depend strongly on the system size, one cannot make a general statement about the cost scaling of the SL algorithm. This is different from the situation in the original algorithm, where the parameter D' controlling the cost can typically be chosen as $D' \propto D^2$ with a prefactor that seems to depend not on the system size but only on the state.

The environment approximation in the SL scheme, despite being positive, does not correspond to the most general boundary purification, a fact that provides some insight into the limitations of the method. The purification of a mixed state is only defined up to an isometry on the traced-out degrees of freedom. But the optimization in the SL algorithm does allow for local isometries, i.e. for tensor products of isometries each acting on a single site of the boundary only. It is possible to devise an ALS algorithm that searches for the optimal general boundary purification, at the expense of a cost function for each site which is no longer quadratic but now quartic in the local tensor variables. The minimum of such a cost function is no longer the solution of linear equations but that of nonlinear equations, finding which is numerically much more demanding than the simple QR decomposition that gives the optimal general boundary MPO in the original algorithm. Therefore such a strategy may result in an undesirable slowing down of the algorithm. Notice also that, while a given purification can always be written efficiently as a positive MPO, namely via contraction of the tensors at each site over their purification bond, the reverse statement is not true, since there exist positive MPO that cannot be written efficiently as purifications [42]. We conclude that, for the problems considered here, it is more advisable to work with a general boundary MPO upon which positivity is not explicitly imposed[6], and, on the basis of it, formulate contraction algorithms in which cost and precision grow simultaneously, as we shall do in the following section.

5. Clusters

The most precise environment approximation is achieved by the original algorithm, in the form of a MPO with sufficiently large bond dimension D'. On the opposite extreme of the spectrum, the lowest computational cost corresponds to the SU, where the environment is represented by a tensor product of matrices each acting on a single virtual bond only. Here, we aim at a contraction scheme that allows us to systematically tune the environment precision together with the cost and that interpolates between the SU and the original algorithm.

This goal is achieved with the help of *clusters*. In a state with short-range correlations, we expect the major contribution to the environment of a given tensor to come from the closest sites. If such sites are not correlated with further ones, or among themselves, the environment will actually be a product, similar to the SU approximation. Correlations in the state cause the environment to be non-separable in general, and to incorporate relevant contributions from faraway sites. Hence, by progressively taking into account the contribution of sites at longer distances, we would improve the environment description.

In our PEPS algorithm, we are interested in the environment of a row (or column), which is required for the update of all the tensors in it. We therefore define a *cluster* of size δ around a certain row as all the surrounding rows which are separated from it by a distance smaller than or equal to δ (and similarly for columns). The environment contribution from sites outside the cluster can be roughly approximated by a product in the spirit of the SU, while the contribution from sites inside the cluster is taken into account with more precision, as in the original algorithm. This defines a new contraction scheme that we call Cluster Update by analogy to [33], and that we abbreviate as CU_δ for cluster size δ. The limiting cases of this strategy are

---

[6] Although when D' is chosen too small, the negative eigenvalues of the environment can lead to instabilities in the tensor update, when D' is large enough, the tensor update is stable and then more accurate.

$\delta = L - 1$, when the environment reverts to that of the original algorithm, and $\delta = 0$, which is closely related to the SU.

### 5.1. Cluster size $\delta = 0$: a generalized Simple Update

The particular case $\delta = 0$ (CU_0) leads to the environment approximation of a certain row, or column, as a product MPO, illustrated in figure 7. This can be found by optimizing the boundary MPO with $D' = 1$, where the standard MPO–MPS ALS scheme can now yield a positive MPO. Indeed, if each of the local tensors of the MPO is positive, this positivity is maintained during the update procedure, since for each local optimization the norm matrix in $N_l A_l = b_l$ is proportional to the identity, and the TN for b_l is positive, as explained in figure 8. Starting the ALS sweeping from an initial positive MPO, which can be trivially constructed from positive local tensors (e.g. of the form $X^\dagger X$ with random X), ensures then a positive environment. Moreover, all contractions can be performed with $O(dD^5)$ operations, so the computation of the optimal separable environment does not exceed the leading cost of the SU.

Imaginary time evolution based on a positive separable boundary MPO leads to an algorithm which is very similar to the SU. The two schemes are characterized by the same computational cost[7] and make use of a separable environment, but the CU_0 method proposed here optimizes the approximate boundary over all possible separable MPO, and hence can be interpreted as a generalized Simple Update.

In order to elucidate the connection between the two algorithms, we study how imaginary time evolution with CU_0 changes PEPS ground state approximations from the SU for the Ising model (1) with various magnetic fields. In our quantitative comparison we consider a specific virtual bond between two neighboring sites in the center of the lattice, and focus on the corresponding λ matrix generated by the SU after convergence, λ^{SU}. The diagonal of that matrix can be directly compared to the converged singular values emerging in CU_0 every time a gate is applied to this particular pair of sites, λ^{CU_0}. As shown in table 1 for an 11×11 PEPS with $D = 2$, the relative difference between the entries of these two λ matrices is below $\approx 10^{-2}$.

We can analyze the similarities between the two algorithms in more detail by looking at the role of the λ matrices in the environment for the update operations. In the SU, the entries in λ^{SU} are determined after applying one gate to the relevant pair of tensors, but (in the case considered here, of nearest neighbor interactions) they are not affected by gates which involve only one member of the pair. For the latter tensor updates, the λ^{SU} matrix enters the environment unchanged, even after the Γ tensors of the pair have been modified. In contrast, in the CU, the environment for a given update operation depends on the surrounding tensors, and changes every time they are updated. In the case of CU_0, a role similar to that of λ^{SU} is played by the eigenvalues of the local tensor in the boundary MPO at the site corresponding to this particular virtual bond, Σ^{CU_0}. For nearest neighbor interactions and the bond that we are considering, there are six Trotter gates in each time step (see the figure in table 1) that involve only one of the tensors of the pair. The Σ^{CU_0} entries change only slightly, $\approx 10^{-2}$, for each such tensor update, as can be appreciated in table 1. And their difference from the corresponding λ^{SU} is of the same order.

---

[7] Just like the environment approximation, the tensor update in a separable environment can be performed with $O(D^5)$ operations.
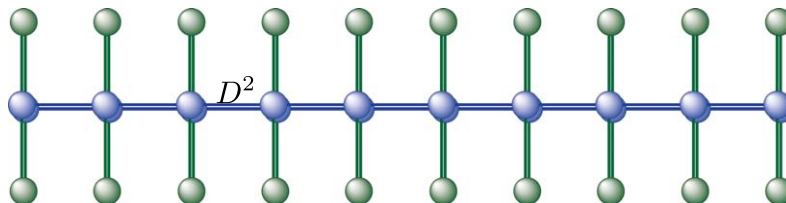
Figure 7. Separable environment with $D' = 1$.
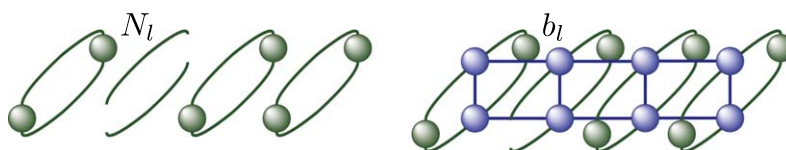

Figure 8. Tensor contractions during the ALS sweeping in the computation of the environment for CU_0. For each local update, the norm matrix (left) is the identity times a positive constant, such that the solution is simply proportional to b_l (right), which is given by a positive TN if each of the local tensors is positive.

Table 1. We apply the SU with $D = 2$ to an 11×11 Ising model with different magnetic fields B. All λ matrices have converged to machine precision and we report the final second entry λ_2^{SU} on the vertical virtual bond at row 5 and column 6. The resulting PEPS is further evolved with CU_0 until convergence is reached. We show the second singular value $\lambda_2^{CU_0}$ emerging in the tensor update on the bond considered and the second eigenvalue $\Sigma_2^{CU_0}$ of the boundary MPO matrix at that place whenever it enters a tensor update. This happens at the six different positions relative to a tensor pair defined in the figure on the right, during the approximation of the four sets of Trotter gates in one time evolution step. We adopt the normalization in which each first λ entry, singular value and eigenvalue is always 1.

	$B = 1.0$	$B = 2.0$	$B = 3.0$	$B = 4.0$
λ_2^{SU}	0.006 007	0.026 032	0.078 572	0.071 486
$\lambda_2^{CU_0}$	0.006 022	0.026 252	0.078 953	0.071 210
$\Sigma_2^{CU_0}$(i)	0.006 008	0.026 049	0.078 399	0.071 216
$\Sigma_2^{CU_0}$(ii)	0.006 008	0.026 048	0.078 421	0.071 216
$\Sigma_2^{CU_0}$(iii)	0.005 784	0.025 236	0.077 678	0.071 391
$\Sigma_2^{CU_0}$(iv)	0.005 784	0.025 237	0.077 828	0.071 391
$\Sigma_2^{CU_0}$(v)	0.005 567	0.024 434	0.076 890	0.071 566
$\Sigma_2^{CU_0}$(vi)	0.005 567	0.024 434	0.077 094	0.071 566
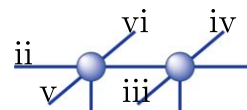
Additionally, we computed the separable boundary MPO for the SU PEPS and compared the eigenvalues of the local tensors to the corresponding Σ^{CU_0}, to find a similar agreement. We observed the same behavior in larger lattices, with larger bond dimensions, as well as on different virtual bonds.

(a)

(b)
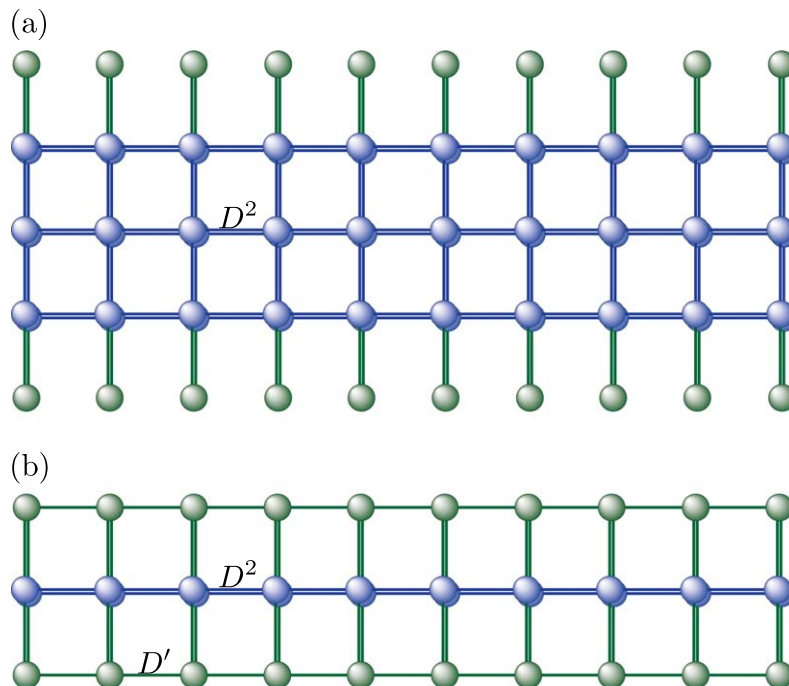
Figure 9. Environment for the CU_1 of a middle row. (a) Outside the cluster, the approximate contraction of the sandwich is performed via a positive separable boundary MPO. (b) The contraction continues inside the cluster via a general boundary MPO with bond dimension D'.

Our observations provide an explanation for the functioning of the SU: because the latter scheme applies the same λ^{SU} matrix to the tensor updates of all four sets of Trotter gates, this λ^{SU} can be seen as a mean value for the six Σ^{CU_0} from the optimal positive separable environment, and the SU does indeed converge it to that mean value.

The CU with $\delta = 0$ always uses the best separable environment in each tensor update, and therefore depends less on the initial state and can produce energies slightly below the ones from the SU. However, the final energies of the two methods are very close to each other (compare figures 4 and 10).

Although the SU is an algorithm completely formulated in the SL, our study in the double-layer picture provided crucial insight into it, thus reinforcing the idea that the boundary should be described as a general MPO.

5.2. From Simple Update to Full Update

By considering clusters of size $\delta \geqslant 1$ we can systematically take into account more of the correlations in the environment approximation. Outside the cluster, the environment is represented by a separable MPO and determined as in the CU_0 described above. Then the cluster tensors are contracted row by row with this boundary, as in the original algorithm, to produce a new boundary MPO with larger bond dimension. The approximation is controlled by the cluster size and the bond dimension D' used in the contractions within the cluster.
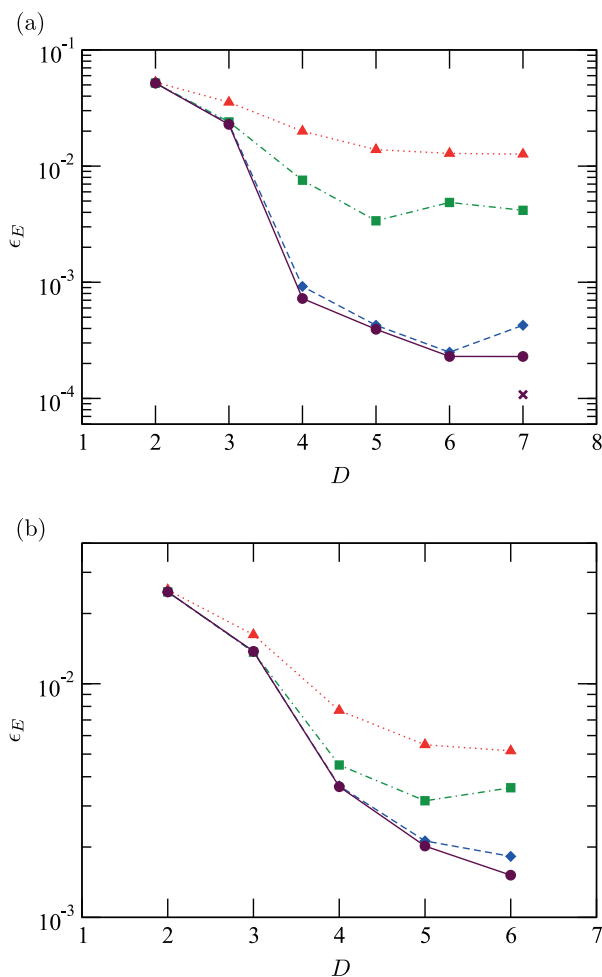
Figure 10. Relative energy error ϵ_E as in figure 4 for a 4×4 (a) and a 10×10 (b) Heisenberg model. We consider the CU_0 (dotted), the CU_1 with $D' = D^2$ (dash–dotted), and the FU with $D' = D^2$ (dashed), $D' = 2D^2$ (solid), and $D' = 130$ (cross).

Figure 9 shows the smallest non-trivial cluster, for CU_1, in which only the two rows adjacent to the one to be updated enter the cluster contraction. In this case, the optimal boundary MPO with bond dimension D' for the update of a row is computed from the action of a bulk row on a separable boundary MPO with $O(dD^5D'^2)$ operations. This is the dominant cost in the environment approximation of CU_1, given the fact that the separable MPO outside the cluster is obtained with only $O(dD^5)$ operations. Hence, the environment approximation for clusters of size $\delta = 1$ is computationally cheaper than the full contraction with cost $O(D^4D'^3) + O(dD^6D'^2)$.

To examine the usefulness of this cluster strategy, we compare its performance to that of the SU via their ground state accuracies for the Heisenberg model (2). Starting from converged SU PEPS, we ran the CU imaginary time evolution with several cluster sizes for various bond dimensions D and D' on 4×4 and 10×10 lattices. Figure 10 contains our cluster results for $\delta = 0$, 1, and $L - 1$, as a function of D, such that they can be compared directly to the SU results of figure 4. The convergence of the CU with cluster size δ as well as with bond
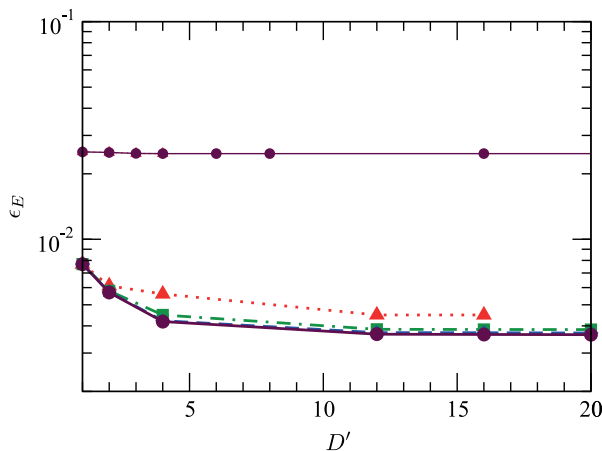
Figure 11. Relative energy error ϵ_E as in figure 4 for a 10×10 Heisenberg model, obtained with various fixed values of the bond dimension D' of the boundary MPO. We propagated $D = 2$ (upper thin lines) and $D = 4$ (lower thick lines) PEPS with the CU with cluster size $\delta = 1$ (dotted), 2 (dash–dotted), and 3 (dashed), and with the FU (solid).

dimension D' can be gathered from figure 11 for 10×10 PEPS with $D = 2$ and 4. We refer to the CU with maximum cluster size $\delta = L - 1$ as *Full Update* (FU), a notion taken from iPEPS (see e.g. [27]). The FU is not identical to the original PEPS algorithm [7], because in the CU the action of single Trotter gates is approximated sequentially, such that for each gate the only tensors updated are those on which the gate directly acts. Thanks to this procedure, the FU requires just the approximate contraction of the norm sandwich, and is therefore computationally less demanding than the original algorithm, in which, additionally, the PEPS sandwich with a full set of Trotter gates acting on all of the state has to be contracted.

We find that the CU_0 produces very similar energies to the SU, with slight improvement for small systems or for large bond dimensions. The difference between the two methods is most apparent in the case of the smaller 4×4 lattice where the CU_0 gives lower energies for bond dimensions $D \geqslant 4$. This can be understood taking into account that the effect of the system boundary, better captured by the environment approximation in CU_0, is more important for smaller systems. We observe then that the CU_1 improves the SU energies considerably. Finally, the FU reduces the energy further significantly when $D \geqslant 4$, and its effect appears more pronounced with increasing bond dimension D. For $D = 2$ and 3, the FU improves upon the CU_1 only in the case of the smaller 4×4 system. Notice that the tables A1–A6 contain the precise energy values that were used in this analysis.

From the arguments above it is apparent that a better representation of the environment is crucial for an improved PEPS approximation of the true ground state. We also infer that larger bond dimensions D require more precise environment representations in the tensor update. Within the CU, this improvement can be achieved systematically by gradually increasing, firstly, the cluster size δ and, secondly, for each fixed δ, the boundary bond dimension D'. Indeed, we can see in figure 11 for each fixed cluster size that with growing D' the energy decreases consistently, as the precision of the environment representation in the tensor update increases. The energy converges at a certain value D'_{max} that depends both on the bond
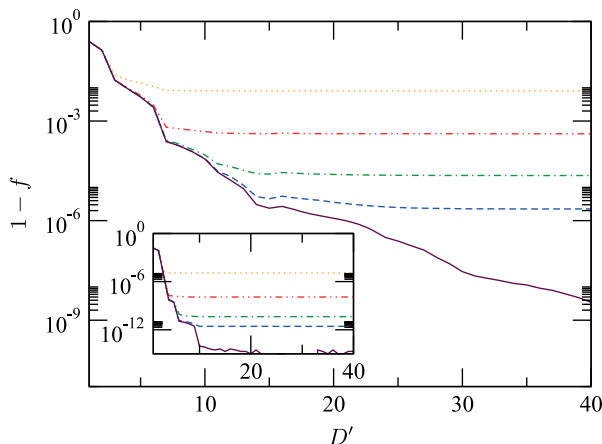
Figure 12. Distance to the exact boundary MPO, $1 - f$ with $f := |\mathrm{tr}\,(\rho_{ref}^{\dagger}\rho\,(D'))|$, for the leftmost column boundary MPO $\rho(D')$ of a $D = 2$ (inset) and a $D = 4$ (main) SU Heisenberg PEPS on a 20×20 lattice. We compare the cluster contraction based on clusters of size $\delta = 1$ (dotted), 2 (dash–double-dotted), 3 (dash–dotted), and 4 (dashed), to the full contraction (solid). The reference boundary MPO ρ_{ref} comes from the full contraction with $D' = 100$, and we adopt the normalization $\mathrm{tr}\,(\rho^{\dagger}\rho) = 1$.

dimension D of the PEPS considered and on the cluster size. While for $D = 2$ the lowest energy is already attained with CU_1, for $D = 4$ the energy improves when larger clusters are used and the FU value is obtained with CU_4.

This behavior agrees with our previous observation that larger bond dimensions benefit more from accurate environment representations. We can gain further insight into this feature by looking at the convergence of a boundary MPO as a function of D' for different cluster sizes. In figure 12 the environment MPO for the leftmost column, computed with different cluster sizes, is compared to the full contraction of the $L - 1$ right columns with large enough D', for PEPS with bond dimensions $D = 2$ and 4 on a 20×20 lattice. We find that for each cluster size δ there exists a maximum value of D' beyond which the distance to the reference boundary MPO no longer decreases, and that this value is smaller than the largest possible $D' = D^{2\delta}$. Considering a sufficiently large fixed D', the distance drops exponentially with increasing cluster size until the value of the full contraction is reached. Beyond this, larger clusters have no effect. Finally, we can directly see that, in order to have the same distance, the $D = 4$ PEPS requires larger clusters and larger boundary bond dimensions than the $D = 2$ PEPS, which explains why it responds more strongly to a better environment representation.

5.3. Computation of expectation values

Although we introduced clusters in the specific context of environment approximations for the tensor update, figure 12 suggests that, in fact, the reduced density matrix of any part of a PEPS can be accurately approximated by a cluster around that part, with a precision determined by the cluster size. Therefore the cluster strategy could also be applied to the evaluation of (local) expectation values, without the need for an accurate contraction of the full TN. To validate this idea, we computed the energy of PEPS with $D = 2$ and 4 on a 20×20 lattice using clusters of
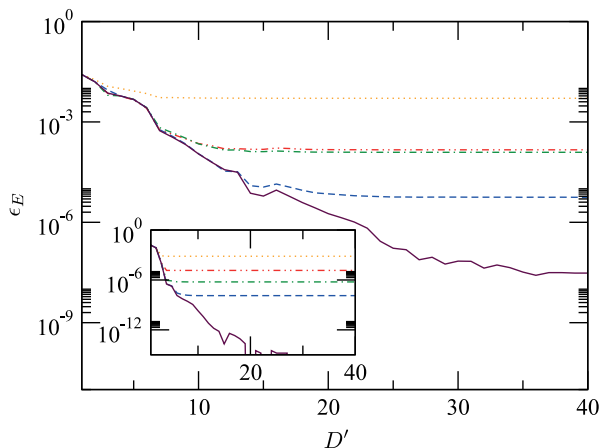
Figure 13. Relative energy error $\epsilon_E := |E(D') - E_{ref}|/|E_{ref}|$ of the $D = 2$ (inset) and $D = 4$ (main) PEPS from figure 12, for the same setting. The reference value E_{ref} comes from the full contraction with $D' = 100$, and the clusters are formed around the individual terms of the Hamiltonian independently of each other.

different sizes around the local terms of the Hamiltonian, shown in figure 13. We observe that, analogously to the case for figure 12, for each cluster size the energy error converges for a certain value of D', and that the larger bond dimensions require larger clusters and larger values of D'. Most remarkably, we find again that for large enough fixed D' the error drops exponentially with the cluster size.

5.4. Applicability to a parallel PEPS code and to iPEPS

In the context of finite PEPS, considered in this study, the computational cost of the environment approximation for CU_δ is lower than that for the full contraction only when $\delta = 0$ ($\mathrm{O}(dD^5)$) and when $\delta = 1$ ($\mathrm{O}(dD^5D'^2)$). Indeed, if the boundary MPO has bond dimension D_0', contracting it with a PEPS row and approximating the result by a new boundary with bond dimension D_1' needs $\mathrm{O}(dD^6D_0'D_1') + \mathrm{O}(D^4D_0'^2D_1') + \mathrm{O}(D^4D_0'D_1'^2)$ operations. If $D_0' = D_1'$, we recover the scaling of the full contraction, so the CU only results in a cheaper scheme if the environment bond dimensions D_i' decrease as we increase the distance i to the row (or column) to be updated. Moreover, after every update of a row, the complete cluster surrounding the next row has to be contracted, without being able to reuse the previously obtained cluster boundary MPO. This situation is different from that for the FU, where, when moving to the update of a new row, only one new boundary MPO has to be determined, as the previously computed and properly stored boundaries can be reused. In the CU, the only boundary MPO that can be reused are the previously obtained separable ones, and $2\delta + 1$ new MPO have to be computed for the update of the next row. Of those, one is separable and thus determined with computational cost $\mathrm{O}(dD^5)$, and two require $\mathrm{O}(dD^5D'^2)$ operations, which we can neglect, such that $2\delta - 2$ new boundary MPO have to be found with cost $\mathrm{O}(dD^6D'^2) + \mathrm{O}(D^4D'^3)$. On the other hand, the CU takes up less memory than the FU. The separable boundary MPO do not have to be written to hard disk but can be stored in main memory since they take up much less memory than MPO with bond dimensions $D' > 1$, and then the cluster boundary MPO are computed on the fly.
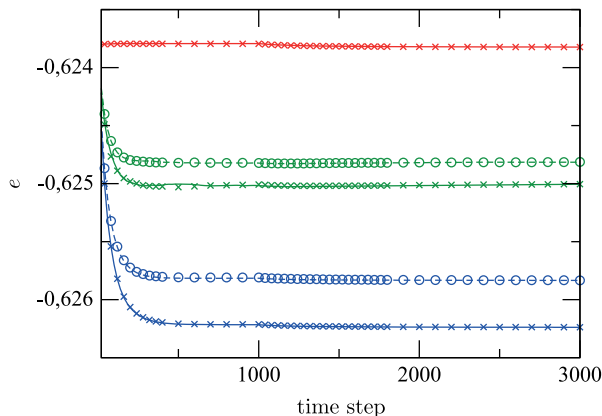
Figure 14. Energy per site e of the Heisenberg model on a 10×10 lattice during imaginary time evolution of a $D = 4$ PEPS via the CU with parallel (lines) and sequential (symbols) tensor updates. The boundary MPO have fixed bond dimension $D' = 1$ (top red lines and symbols), $D' = 2$ (middle green lines and symbols), and $D' = 16$ (bottom blue lines and symbols), and we further distinguish cluster size 1 (dashed lines and circles) and 2 (solid lines and crosses). The state propagates 1000 time steps with $\tau = 0.01$ and then 2000 time steps with $\tau = 0.001$.

Although the CU_δ with cluster sizes $\delta > 1$ does not reduce the computational cost of a sequential algorithm, in which one tensor is updated after another, it can reduce the cost of a parallel algorithm, in which different rows or columns are updated simultaneously on different processors. Assuming that the time for the optimization of a boundary MPO (for a middle row) is t_B on average, and that the update of all the tensors in a row or column is achieved in the time t_U, then the sequential FU requires $2(L - 2) \cdot t_B + L \cdot t_U$ for one set of Trotter gates. In contrast, each row or column update with the CU_δ for $\delta \geqslant 1$ necessitates the computation of only $2(\delta - 1)$ boundary MPO and thus has the cost $2(\delta - 1) \cdot t_B + t_U$. We conclude that a parallel CU algorithm can attain an L/δ speed-up. Since δ does not depend on the system size, L, but only on the bond dimension, D, it can be chosen much smaller than L, such that this speed-up factor may be large. This estimation neglects all computations with sub-leading costs $O(dD^5)$ and $O(dD^5D'^2)$ and the communication between the parallel processors. Although the latter will have an impact on the final performance of the algorithm, we expect the speed-up to be still significant, given the fact that just the small individual tensors of separable boundary MPO have to be exchanged between different processors after each set of Trotter gates.

The success of this parallelization strategy relies heavily on the simultaneous update of tensors in different rows. As described in section 2, each tensor update is based on solving a system of linear equations that arises from the minimization of a cost function for the whole PEPS by utilizing an ALS scheme. In this scheme one sweeps over the tensors and for each one minimizes the cost function under the assumption that all the others are fixed. This guarantees a non-increasing cost function only when the tensors are updated sequentially. An important question is then that of whether the convergence of the energy in imaginary time is as fast with the independent updates as with the sequential ones. That this is indeed the case can be gathered from figure 14. The plot demonstrates an impressive agreement, which can be attributed to a minor modification of the tensor when the action of a time evolution gate is approximated in a

sequential update. We conclude that imaginary time evolution with the CU constitutes a natural basis for a parallel ground state search algorithm based on PEPS. An agreement similar to that in figure 14 cannot be expected in direct energy minimization, where a tensor is changed significantly during an update[8].

Although we carried out our analysis in the framework of finite PEPS, it is clear that the CU_δ can also be applied to iPEPS, to replace the costly computation of the environment via the dominant boundary eigenvector with $D' > 1$ [18], fixed point corner transfer matrices [19], or second-renormalized environment [22]. A CU procedure would only require the search for the dominant boundary eigenvector with $D' = 1$, which needs $O(dD^5)$ operations, followed by a cluster contraction as in the finite case. Then, the cost and precision of both tensor update and expectation value computation would be determined by the cluster size and the bond dimension D' employed in the contraction of the cluster. Furthermore, it is always possible to evaluate clusters by means of Monte Carlo sampling [15, 16]. This method requires only the contraction of the PEPS coefficients, computationally less costly than the contraction of the PEPS sandwich, for different sampled values of the physical indices s_1, s_2, ..., depicted in figure 1. While a full infinite PEPS cannot be sampled, since this would necessitate determining infinitely many classical spin values, clusters open the door to variational Monte Carlo in the realm of iPEPS. For the sampling of an observable as well as of an energy gradient, a cluster would be formed around the tensors considered and then only the physical indices of that cluster would have to be sampled. Larger clusters necessitate longer sampling times such that the error of a finite cluster could be adjusted together with the Monte Carlo error according to the available computational resources.

6. Conclusions

In this article, we have analyzed the environment representation in previous proposals, namely the Simple Update and the Single-Layer algorithm. We have shown how the different approximations applied to the environment explain the limitations of each method in the achievable ground state accuracy, an issue that we have studied quantitatively in the context of finite PEPS. On the basis of this deeper understanding, we have formulated a new proposal, the cluster strategy, that allows a systematic increase of the environment precision from the simplest possible representation, in the SU, to the most accurate full contraction, in the FU.

In its simplest form, CU_0 provides an explanation for the Simple Update in terms of a separable boundary approximation, and constitutes a slightly improved version of the latter for the models analyzed here, characterized by the same computational cost. The first non-trivial Cluster Update, CU_1, produces significantly better ground state energies than the SU, and has a lower computational cost than the FU. In general, CU_δ interpolates naturally between the SU and the FU. We have shown that increasing the cluster size improves the precision of the environment approximation exponentially. This improvement applies directly to the computation of local observables, which can always be accelerated with the help of clusters.

---

[8] In direct energy minimization, the energy $\langle \psi | \hat{H} | \psi \rangle / \langle \psi | \psi \rangle$ is minimized directly by sweeping over the tensors with an ALS scheme. This algorithm converges typically within far fewer sweeps over the PEPS than imaginary time evolution, when all sweeps for all time steps are taken into account, and therefore modifies a tensor considerably in an update.

Our analysis of the computational costs of the CU revealed that in the sequential update of finite PEPS any cluster size $\delta > 1$ exceeds the cost of the FU, which can reuse intermediate calculations more efficiently. However, the CU_δ forms the basis of a very promising parallel PEPS algorithm, with a prospective large gain in computational time also for larger clusters. Although our numerical studies have all been carried out in the framework of finite PEPS, we have also argued how the CU_δ is straightforwardly useful for the infinite iPEPS ansatz.

In summary, we have shown that the environment approximation is a key ingredient to the precision of any PEPS contraction, whether we are interested in the norm, or in some expectation value. The CU_δ provides the means to control this approximation accuracy and can be used in any contraction. It is then reasonable to think of its potential applicability to other PEPS algorithms.

Acknowledgments

Appendix A

Here, we list explicitly a selection of precise values, as they are used in the main text (see tables A1–A6).

Regarding the reference energies, for $L = 4$ the exact energy values come from exact diagonalization. The exact Heisenberg ground state energy per site on a 4×4 lattice reads $-0.574\,325\,44$. For $L = 10$, we computed the exact values with the quantum Monte Carlo loop algorithm from ALPS [43–45], and we use the result for temperature $T = 0.0001$, where we have checked consistency with $T = 0.01$ and 0.001. That energy per site for a 10×10 system reads $-0.628\,655(2)$.

Appendix B

We used the following setup for time evolution and energy computation, if not explicitly stated otherwise.

We initialize imaginary time evolution with a separable $D = 2$ PEPS in which the zeros are replaced by noise as uniformly distributed random numbers from $[-0.01, 0.01]$. This state is evolved for N_1 steps with τ_1, followed by N_2 steps with τ_2, and so on, which we abbreviate to the short notation $(N_1 \times \tau_1, N_2 \times \tau_2, \ldots)$ for fixed bond dimension D. In order to specify a successively growing value of D, we introduce the recursive notation $(D_{i+1} = D_i^\tau + 1, N_1 \times \tau_1, N_2 \times \tau_2, \ldots)$. It defines the next PEPS for the propagation with bond dimension D_{i+1} as the final state of the previous evolution with bond dimension D_i and time step τ_i with a bond dimension incremented by 1. In the case of the Cluster Update and that of the Full Update, the additional parameter D' is typically chosen as $D' = 1, 2$, and, related to D, as

Table A1. Energy per site e of the Heisenberg model on a 4×4 and on a 10×10 lattice, obtained by means of the SU, presented in figure 4.

D	4×4	10×10
2	−0.54404(1)	−0.61281(1)
3	−0.55396(2)	−0.61846(2)
4	−0.56281(1)	−0.62382(1)
5	−0.56628(2)	−0.62520(2)
6	−0.56684(3)	−0.62541(2)
7	−0.56696(2)	−0.62537(2)
8	−0.56715(3)	−0.62538(2)

Table A2. Energy per site e of the Heisenberg model on a 4×4 and on a 10×10 lattice, obtained by means of the CU_0, presented in figure 10.

D	4×4	10×10
2	−0.54404(3)	−0.61280(2)
3	−0.55397(2)	−0.61846(2)
4	−0.56287(5)	−0.62382(2)
5	−0.56637(2)	−0.62521(2)
6	−0.56694(2)	−0.62541(2)
7	−0.56706(3)	

Table A3. Energy per site e of the Heisenberg model on a 4×4 and on a 10×10 lattice, obtained by means of the CU_1 with $D' = D^2$, presented in figure 10.

D	4×4	10×10
2	−0.54458(2)	−0.61310(2)
3	−0.5605(3)	−0.62007(1)
4	−0.56999(3)	−0.62583(2)
5	−0.57238(7)	−0.62667(2)
6	−0.57153(7)	−0.6264(2)
7	−0.57194(1)	

$D' = D, D^2$, and so on. The final PEPS obtained with a certain value of D' is always the initial state for increased D with that D'.

Regarding the energy computation, all energies are evaluated with $D' = 100$ for the final PEPS corresponding to the smallest time step. We define the energy error as the difference between the energy of this final state and the energy of an intermediate state. The latter is either the PEPS obtained after half of the evolution or the final PEPS corresponding to the immediately larger time step, depending on whether or not the propagation was also performed with this larger time step.

Figure 4: we propagate the initial $D = 2$ PEPS 1000 time steps with $\tau = 0.1$, then 2000 time steps with $\tau = 0.01$, then 8000 time steps with $\tau = 0.001$, and then according to the configuration ($D_{i+1} = D_i^{\tau=0.01} + 1$, $2000 \times \tau = 0.01$, $8000 \times \tau = 0.001$).

Table A4. Energy per site e of the Heisenberg model on a 4×4 and on a 10×10 lattice, obtained by means of the FU, presented in figure 10.

D	D'	4×4	10×10
2	4	−0.54458(2)	−0.61310(2)
	8	−0.54458(2)	−0.61310(2)
3	9	−0.56101(2)	−0.62002(2)
	18	−0.5612(1)	−0.62000(2)
4	16	−0.5738(3)	−0.62636(3)
	32	−0.5739(2)	−0.62637(2)
5	25	−0.57408(1)	−0.62732(4)
	50	−0.57410(3)	−0.62739(1)
6	36	−0.57418(2)	−0.62751(2)
	72	−0.57419(1)	−0.62770(7)
7	49	−0.57408(1)	
	98	−0.57419(1)	
	130	−0.57426(1)	

Table A5. Energy per site e of the Heisenberg model on a 10×10 lattice, from $D = 2$, presented in figure 11.

D'	CU_1	FU
1	−0.61280(2)	−0.61280(2)
2	−0.61290(2)	−0.61289(1)
3	−0.61307(2)	−0.61307(1)
4	−0.61310(2)	−0.61310(2)
100		−0.61310(2)

Table A6. Energy per site e of the Heisenberg model on a 10×10 lattice, from $D = 4$, presented in figure 11.

D'	CU_1	CU_2	CU_3	CU_4	FU
1	−0.62382(2)	−0.62382(2)	−0.62382(2)	−0.62382(2)	−0.62382(2)
2	−0.62481(1)	−0.62501(1)	−0.62506(5)	−0.62504(1)	−0.62508(4)
4	−0.62513(3)	−0.62583(4)	−0.62600(1)	−0.62607(2)	−0.62602(1)
12	−0.62583(2)	−0.62623(2)	−0.62631(2)	−0.62634(3)	−0.62635(2)
16	−0.62583(2)	−0.62623(2)	−0.62632(3)	−0.62635(3)	−0.62636(3)
20		−0.62624(2)	−0.62632(2)	−0.62635(2)	−0.62636(2)
32					−0.62637(3)

Figure 6: we propagate the initial $D = 2$ PEPS, each time, 10 000 steps first with $\tau = 0.1$, then with $\tau = 0.01$, then with $\tau = 0.001$, then 20 000 steps with $\tau = 0.0001$, and finally 50 000 steps with $\tau = 0.000\,01$. The $D = 3$ and 4 results, used in the analysis, were obtained by evolving the final $D = 2$ PEPS from $\tau = 0.0001$ further according to the configuration $(D_{i+1} = D_i^{\tau=0.0001} + 1, 20000 \times \tau = 0.0001)$.

Table 1: we apply the SU to an 11×11 Ising model with different magnetic fields B, and propagate the initial $D = 2$ PEPS 10 000 time evolution steps with $\tau = 0.1$ and then 10 000 steps

with $\tau = 0.01$. The resulting PEPS is further evolved with the CU_0 for 10 000 time steps with $\tau = 0.01$. All numbers shown are converged to machine precision.

Figure 10 (a) $N = 4 \times 4$; the initial state of the imaginary time evolution is the converged $D = 2$ SU ground state approximation to time step $\tau = 0.01$. We propagate this state 1000 time steps with $\tau = 0.01$, then 2000 time steps with $\tau = 0.001$, and then according to the configuration $(D_{i+1} = D_i^{\tau=0.01} + 1, 1000 \times \tau = 0.01, 2000 \times \tau = 0.001)$ up to bond dimension $D = 5$. Then, we continue with $(D_{i+1} = D_i^{\tau=0.01} + 1, 500 \times \tau = 0.01, 1000 \times \tau = 0.001)$. In the case of the FU with $D = 7$ and $D' = 130$, the state propagates 100 time steps with $\tau = 0.01$.

Figure 10 (b) $N = 10 \times 10$; the initial state of the imaginary time evolution is the converged $D = 2$ SU ground state approximation to time step $\tau = 0.01$. We propagate this state 1000 time steps with $\tau = 0.01$, then 2000 time steps with $\tau = 0.001$, and then according to the configuration $(D_{i+1} = D_i^{\tau=0.01} + 1, 1000 \times \tau = 0.01, 2000 \times \tau = 0.001)$. In the cases of the CU_0 and the CU_1 we use this time evolution configuration up to $D = 5$, and for $D = 6$ propagate the states 500 time steps with $\tau = 0.01$ and then 500 time steps with $\tau = 0.001$. In the case of the FU we use this configuration up to $D = 4$, and for $D = 5$ evolve the states 500 time steps with $\tau = 0.01$ and then 1000 time steps with $\tau = 0.001$, and for $D = 6$ we propagate the states 500 time steps with $\tau = 0.01$.

Figure 11: the initial state of the imaginary time evolution is the converged SU ground state approximation to time step $\tau = 0.01$ with the bond dimension considered. In the case of $D = 2$, we propagate this state 1000 time steps with $\tau = 0.01$, then 2000 time steps with $\tau = 0.001$. In the case of $D = 4$, for CU_1 we use $(1000 \times \tau = 0.01, 2000 \times \tau = 0.001)$, for CU_2 we use $(1000 \times \tau = 0.01, 1000 \times \tau = 0.001)$, and for CU_3, CU_4, and FU we use $(500 \times \tau = 0.01, 1000 \times \tau = 0.001)$.

References

[1] Vidal G 2003 Efficient classical simulation of slightly entangled quantum computations *Phys. Rev. Lett.* **91** 147902

[2] Daley A J, Kollath C, Schollwöck U and Vidal G 2004 Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces *J. Stat. Mech.: Theor. Exp.* P04005

[3] Vidal G 2004 Efficient simulation of one-dimensional quantum many-body systems *Phys. Rev. Lett.* **93** 040502

[4] White S R and Feiguin A E 2004 Real-time evolution using the density matrix renormalization group *Phys. Rev. Lett.* **93** 076401

[5] Verstraete F, García-Ripoll J J and Cirac J I 2004 Matrix product density operators: simulation of finite-temperature and dissipative systems *Phys. Rev. Lett.* **93** 207204

[6] Zwolak M and Vidal G 2004 Mixed-state dynamics in one-dimensional quantum lattice systems: a time-dependent superoperator renormalization algorithm *Phys. Rev. Lett.* **93** 207205

[7] Verstraete F and Cirac J I 2004 Renormalization algorithms for quantum many-body systems in two and higher dimensions arXiv:cond-mat/0407066v1 [cond-mat.str-el]

[8] Shi Y-Y, Duan L-M and Vidal G 2006 Classical simulation of quantum many-body systems with a tree tensor network *Phys. Rev.* A **74** 022320

[9] Vidal G 2008 Class of quantum many-body states that can be efficiently simulated *Phys. Rev. Lett.* **101** 110501

[10] Verstraete F, Porras D and Cirac J I 2004 Density matrix renormalization group and periodic boundary conditions: a quantum information perspective *Phys. Rev. Lett.* **93** 227205

[11] Murg V, Verstraete F and Cirac J I 2005 Efficient evaluation of partition functions of inhomogeneous many-body spin systems *Phys. Rev. Lett.* **95** 057206

[12] Murg V, Verstraete F and Cirac J I 2007 Variational study of hard-core bosons in a two-dimensional optical lattice using projected entangled pair states *Phys. Rev.* A **75** 033605

[13] Murg V, Verstraete F and Cirac J I 2009 Exploring frustrated spin systems using projected entangled pair states *Phys. Rev.* B **79** 195119

[14] Bañuls M-C, Pérez-García D, Wolf M M, Verstraete F and Cirac J I 2008 Sequentially generated states for the study of two-dimensional systems *Phys. Rev.* A **77** 052306

[15] Sandvik A W and Vidal G 2007 Variational quantum Monte Carlo simulations with tensor-network states *Phys. Rev. Lett.* **99** 220602

[16] Schuch N, Wolf M M, Verstraete F and Cirac J I 2008 Simulation of quantum many-body systems with strings of operators and Monte Carlo tensor contractions *Phys. Rev. Lett.* **100** 040501

[17] Verstraete F, Murg V and Cirac J I 2008 Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems *Adv. Phys.* **57** 143

[18] Jordan J, Orús R, Vidal G, Verstraete F and Cirac J I 2008 Classical simulation of infinite-size quantum lattice systems in two spatial dimensions *Phys. Rev. Lett.* **101** 250602

[19] Orús R and Vidal G 2009 Simulation of two-dimensional quantum systems on an infinite lattice revisited: corner transfer matrix for tensor contraction *Phys. Rev.* B **80** 094403

[20] Levin M and Nave C P 2007 Tensor renormalization group approach to two-dimensional classical lattice models *Phys. Rev. Lett.* **99** 120601

[21] Jiang H C, Weng Z Y and Xiang T 2008 Accurate determination of tensor network state of quantum lattice models in two dimensions *Phys. Rev. Lett.* **101** 090603

[22] Xie Z Y, Jiang H C, Chen Q N, Weng Z Y and Xiang T 2009 Second renormalization of tensor-network states *Phys. Rev. Lett.* **103** 160601

[23] Östlund S and Rommer S 1995 Thermodynamic limit of density matrix renormalization *Phys. Rev. Lett.* **75** 3537

[24] Barthel T, Pineda C and Eisert J 2009 Contraction of fermionic operator circuits and the simulation of strongly correlated fermions *Phys. Rev.* A **80** 042333

[25] Kraus C V, Schuch N, Verstraete F and Cirac J I 2010 Fermionic projected entangled pair states *Phys. Rev.* A **81** 052338

[26] Pižorn I and Verstraete F 2010 Fermionic implementation of projected entangled pair states algorithm *Phys. Rev.* B **81** 245110

[27] Corboz P, Orús R, Bauer B and Vidal G 2010 Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states *Phys. Rev.* B **81** 165104

[28] Pineda C, Barthel T and Eisert J 2010 Unitary circuits for strongly correlated fermions *Phys. Rev.* A **81** 050303(R)

[29] Corboz P, Jordan J and Vidal G 2010 Simulation of fermionic lattice models in two dimensions with projected entangled-pair states: next-nearest neighbor Hamiltonians *Phys. Rev.* B **82** 245119

[30] Corboz P, White S R, Vidal G and Troyer M 2011 Stripes in the two-dimensional *t-J* model with infinite projected entangled-pair states *Phys. Rev.* B **84** 041108(R)

[31] Pižorn I, Wang L and Verstraete F 2011 Time evolution of projected entangled pair states in the single-layer picture *Phys. Rev.* A **83** 052321

[32] Wang L, Pižorn I and Verstraete F 2011 Monte Carlo simulation with tensor network states *Phys. Rev.* B **83** 134421

[33] Wang L and Verstraete F 2011 Cluster update for tensor network states arXiv:1110.4362v1 [cond-mat.str-el]

[34] Schuch N, Wolf M M, Verstraete F and Cirac J I 2007 Computational complexity of projected entangled pair states *Phys. Rev. Lett.* **98** 140506

[35] Nishino T 1995 Density matrix renormalization group method for 2D classical models *J. Phys. Soc. Japan* **64** 3598

[36] Pirvu B, Murg V, Cirac J I and Verstraete F 2010 Matrix product operator representations *New J. Phys.* **12** 025012

[37] Li H and Haldane F D M 2008 Entanglement spectrum as a generalization of entanglement entropy: identification of topological order in non-abelian fractional quantum Hall effect states *Phys. Rev. Lett.* **101** 010504

[38] Cirac J I, Poilblanc D, Schuch N and Verstraete F 2011 Entanglement spectrum and boundary theories with projected entangled-pair states *Phys. Rev.* B **83** 245134

[39] Hastings M B 2006 Solving gapped Hamiltonians locally *Phys. Rev.* B **73** 085115

[40] Evenbly G and Pfeifer R N C 2013 Improving the efficiency of variational tensor network algorithms arXiv:1310.8023v2 [cond-mat.str-el]

[41] Vidal G 2007 Classical simulation of infinite-size quantum lattice systems in one spatial dimension *Phys. Rev. Lett.* **98** 070201

[42] De las Cuevas G, Schuch N, Pérez-García D and Cirac J I 2013 Purifications of multipartite states: limitations and constructive methods *New J. Phys.* **15** 123021

[43] Todo S and Kato K 2001 Cluster algorithms for general-*S* quantum spin systems *Phys. Rev. Lett.* **87** 047203

[44] Albuquerque A F *et al* 2007 The ALPS project release 1.3: open-source software for strongly correlated systems *J. Magn. Magn. Mater.* **310** 1187

[45] Bauer B *et al* 2011 The ALPS project release 2.0: open source software for strongly correlated systems *J. Stat. Mech.: Theor. Exp.* P05001