Improved numerical methods for infinite spin chains with long-range interactions

V. Nebendahl and W. Dür

Institut für Theoretische Physik, Universität Innsbruck, Technikerstr. 25, A-6020 Innsbruck, Austria

(Received 5 June 2012; published 7 February 2013)

We present several improvements of the infinite matrix product state (iMPS) algorithm for finding ground states of one-dimensional quantum systems with long-range interactions. As a main ingredient, we introduce the superposed multioptimization method, which allows an efficient optimization of exponentially many MPS of different lengths at different sites all in one step. Here, the algorithm becomes protected against position-dependent effects as caused by spontaneously broken translational invariance. So far, these have been a major obstacle to convergence for the iMPS algorithm if no prior knowledge of the system's translational symmetry was accessible. Further, we investigate some more general methods to speed up calculations and improve convergence, which might be partially interesting in a much broader context, too. As a more special problem, we also look into translational invariant states close to an invariance-breaking phase transition and show how to avoid convergence into wrong local minima for such systems. Finally, we apply these methods to polar bosons with long-range interactions. We calculate several detailed Devil's staircases with the corresponding phase diagrams and investigate some supersolid properties.

DOI: 10.1103/PhysRevB.87.075413 PACS number(s): 02.70.−c, 03.67.−a, 67.85.−d

I. INTRODUCTION

A numerical method for the simulation of large quantum systems needs to meet two requirements: (i) an ansatz suitable for the problem in question, and (ii) efficient algorithms to find the (at least nearly) optimal solution within the chosen ansatz. For one-dimensional quantum systems on lattices, the currently most powerful numerical tools are matrix product states (MPS) based algorithms including the density matrix renormalization group (DMRG).[1–5] Their primary limitation is given by the amount of entanglement they can handle. Several extensions of MPS have been conceived to overcome this restriction as, e.g., Refs. 6–8, but for most practical applications MPS, are still the first choice. This is mainly due to the performance of the underlying optimization routines. Although the general task to find ground states is known to be NP hard,[9,10] commonly used algorithms seem to have no problem to attain optimal MPS solutions within computer precision for a plenitude of physical relevant systems. Nevertheless, for some physical systems of interest, these algorithms still face severe difficulties.

In this paper, we treat such problematic cases given by ground states of infinite spin chains with long-range interactions. An increasing interest in reliable numerical methods for these states is, e.g., triggered by the excellent experimental control of ultracold gases and the possibility to realize systems with long-range dipole-dipole interactions such as Rydberg atoms or polar molecules.[11–13] Although these systems are of finite size, one is often interested in the thermodynamical limit (i.e., infinite systems) for a better insight.

Different strategies are known for the numerical study of ground states in the thermodynamic limit. One might try to extrapolate results from a series of increasingly large finite systems[14–16] or directly construct the infinite state itself. The latter is, e.g., done by the infinite time-evolving block decimation (iTEBD) algorithm,[17] which is based on an explicit translational invariant ansatz. This ansatz is quite elegant for interactions which are restricted to nearest neighbors, while it gets impractical for long-range interactions.

A comfortable way to incorporate long-range interactions is to encode them into a matrix product operator (MPO).[18–21] This concept can be integrated in an infinite matrix product state (iMPS) algorithm.[22,23] The basic idea of this iMPS algorithm is to obtain the ground state of an infinite system as the fixed point of a constantly growing finite state by inserting iteratively new sites into its middle until convergence is reached. A major disadvantage of this approach is that the algorithm generally fails to converge if the ground state has a nontrivial translational symmetry.

In this paper, various extensions to the basic iMPS algorithm are presented. As a central element, the superposed multioptimization (SMO) is introduced (Sec. III B), which provides a remedy for the just mentioned convergence problem. Here, the key idea is to join the optimization of exponentially many MPS in a superposition and solve it efficiently. Due to this superposed optimization, the effective overall problem becomes translational invariant again and poses no longer a hindrance for convergence.

We also introduce several improvements which work independently of the SMO method. As such, we present two different modifications of the MPS optimization routine: On the one hand, we use physical insight for systems close to translational invariance-breaking phase transition to suggest a method to reduce the danger of being trapped into a local minimum of the energy (Sec. III D). On the other hand, we provide a more technical discussion as to how to recycle information from previous optimizations to speed up calculations (Sec. IV C). As a part of these considerations, we provide a simple implementation of a Davidson-type algorithm[24] based on information from previous optimizations (Appendix D). This implementation is not bonded to the MPS framework and hence might be used in much broader context.

All algorithms are described in depth, such that readers who are willing to reproduce our results should find all the information needed for successful programming. As a consequence, readers who just like to understand the crucial ideas might find the amount of algorithmic details far beyond their interest. Having these two types of readers in mind, we

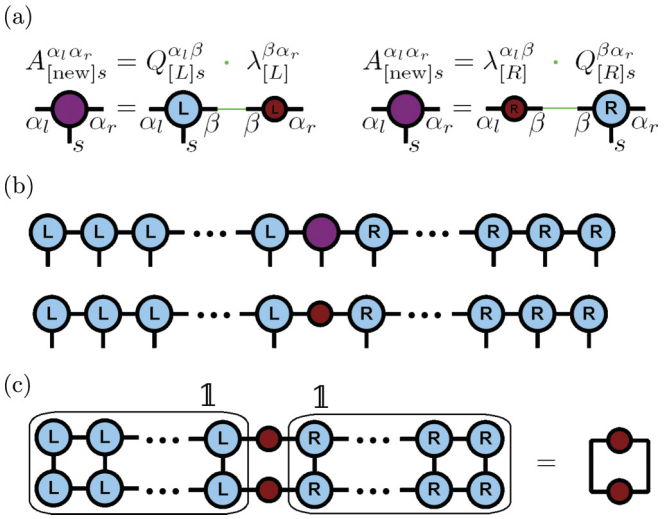

FIG. 1. (Color online) (a) Diagrammatic representation of the two decompositions of A according to Eq. (8). Vertical legs correspond to physical indices, while horizontal legs belong to the auxiliary indices. Connected legs are summed over. (b) Resulting structure of the MPS (11) before and after the decomposition of the last $A_{[\text{new}]}$. (c) MPS version of $\langle\psi|\psi\rangle$, where the turned over MPS symbolizes the complex conjugate. Due to the orthogonal decomposition (10), the tensors in the left and right boxes generate the identity (12), which allows an easy control of the MPS norm (13).

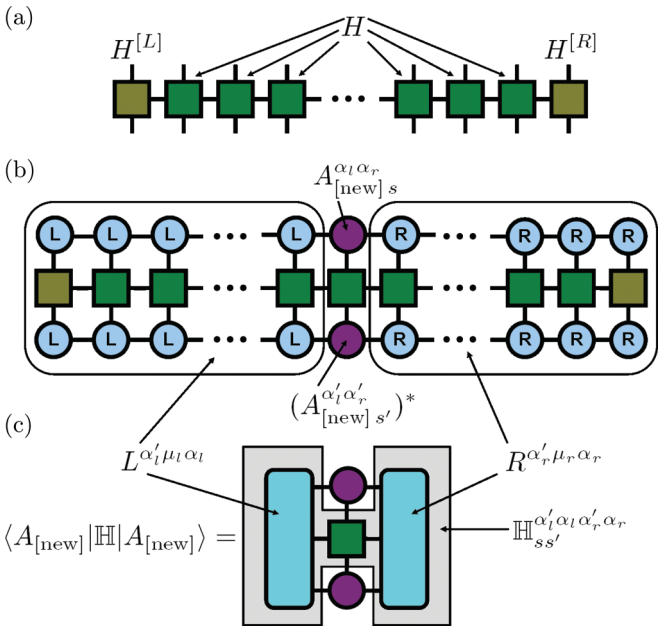

FIG. 2. (Color online) (a) Diagrammatic representation of the Hamiltonian MPO (5). (b) MPS and MPO realization of $\langle\psi|H|\psi\rangle$ (compare with Fig. 1). The boxes indicate the left and right halves $L^{\alpha_l'\mu_l\alpha_l}$ [Eq. (16)] and $R^{\alpha_r'\mu_r\alpha_r}$ [Eq. (17)]. (c) Same object as above with contracted inner indices of $L^{\alpha_l'\mu_l\alpha_l}$ and $R^{\alpha_r'\mu_r\alpha_r}$. The object in the H-shaped box corresponds to the effective operator $\mathbb{H}$ [Eq. (15)].

partitioned the material according to its level of detail into different chapters, such that entire passages can be omitted.

In Sec. II, we review the basic concepts and the iMPS algorithm as presented in Ref. 22. Readers who feel safe to skip this part find in Figs. 1 and 2 a pictorial description of all symbols used in this section. In Sec. III, the main concepts of this paper are presented, above all the SMO method. Changes of the algorithm are kept to a minimum in here, in contrast to Sec. IV where several algorithmic improvements and their numerical realization are presented in detail. Each subsection of Sec. IV contains an individual topic, which is presented in the first lines. These sections can be skipped without danger of losing the ability to understand the rest of the paper. A slight exception might be Sec. IV C, in which the concept of iterative eigenvector solver based on subspace projections is reviewed. Familiarity with this concept is assumed in Secs. IV D, IV E, and Appendix B 1. In Sec. V, we apply our algorithm to a system of polar bosons with long-range interactions. Detailed calculations of Devil's staircases and phase diagrams are shown and a supersolidlike phase is investigated. Finally, the paper is complemented by an appendix, into which several details have been outsourced.

II. BASIC ALGORITHM

In this section, we review fundamental concepts[1] and the iMPS algorithm as presented in Ref. 22. For this algorithm to work, not only the Hamiltonian but also the ground state have to be translational invariant. The extension to ground states with broken translational symmetry will be introduced in Sec. III.

A. MPS and MPO

In this paper, we deal with spin chains. The quantum state of a spin chain is determined by the inner degrees of freedom of its components

$$|\psi\rangle = \sum_{s_1 s_2 \ldots s_n} \mathcal{A}_{s_1 s_2 \ldots s_n} \cdot |s_1\rangle \otimes |s_2\rangle \otimes \ldots \otimes |s_n\rangle. \tag{1}$$

Since the size of the tensor $\mathcal{A}_{s_1 s_2 \ldots s_n}$ grows exponentially with the number of sites, a more economical representation is needed. A matrix product state[1,3,4] (MPS) consists in the ansatz

$$\mathcal{A}_{s_1 s_2 \ldots s_n} = A^{\alpha_1}_{[1]\,s_1} \cdot A^{\alpha_1\alpha_2}_{[2]\,s_2} \cdot A^{\alpha_2\alpha_3}_{[3]\,s_3} \ldots A^{\alpha_{n-2}\alpha_{n-1}}_{[n-1]\,s_{n-1}} \cdot A^{\alpha_{n-1}}_{[n]\,s_n}, \tag{2}$$

where we used the Einstein summation convention. For a general exact quantum state, exponentially growing bond dimensions α_i are needed, but even for infinite systems excellent approximations are possible with moderate bond dimensions if the ground state fulfills an area law for the entanglement entropy.[25]

For Hamiltonians

$$\hat{H} = \sum_{s_1 \ldots s_n, s_1' \ldots s_n'} \mathcal{H}^{s_1' \ldots s_n'}_{s_1 \cdots s_n} \cdot |s_1'\rangle\langle s_1| \otimes \ldots \otimes |s_n'\rangle\langle s_n|, \tag{3}$$

a similar ansatz as for the quantum state (2) leads to the concept of matrix product operators[18–21]

$$\mathcal{H}^{s_1' s_2' \ldots s_n'}_{s_1 s_2 \ldots s_n} = H^{\mu_1}_{[1]\,s_1' s_1} \cdot H^{\mu_1\mu_2}_{[2]\,s_2' s_2} \cdot H^{\mu_2\mu_3}_{[3]\,s_3' s_3} \ldots H^{\mu_{n-2}\mu_{n-1}}_{[n-1]\,s_{n-1}' s_{n-1}} \cdot H^{\mu_{n-1}}_{[n]\,s_n' s_n}. \tag{4}$$

Many relevant Hamiltonians are represented by MPO with relative small bond dimensions. For our purposes, it is

important to mention that this is often also true for Hamiltonians with long-range interaction terms (see, e.g., Ref. 21). A recipe for the explicit construction is explained in Appendix A. In the case of translational invariant Hamiltonians (see remark below), the MPO can be built in such a fashion that all tensors $H^{\mu_{i-1}\mu_i}_{[i]\,s'_i s_i}$ are identical for $2 \leqslant i \leqslant n-1$. This allows us to drop the index in the square brackets except for the leftmost and rightmost tensors:

$$\mathcal{H}^{s'_1 s'_2 \ldots s'_n}_{s_1 s_2 \ldots s_n} = H^{\mu_1}_{[L]\,s'_1 s_1} \cdot H^{\mu_1\mu_2}_{s'_2 s_2} \cdot H^{\mu_2\mu_3}_{s'_3 s_3} \cdot H^{\mu_3\mu_4}_{s'_4 s_4} \ldots H^{\mu_{n-3}\mu_{n-2}}_{s'_{n-2} s_{n-2}} \cdot H^{\mu_{n-2}\mu_{n-1}}_{s'_{n-1} s_{n-1}} \cdot H^{\mu_{n-1}}_{[R]\,s'_n s_n}. \quad (5)$$

Furthermore, all tensors H are independent of the total number of sites. As a consequence, the MPO of a translational invariant Hamiltonian for n sites can easily be augmented to $n+1$ sites by just inserting another tensor H.

Apart from an efficient representation of quantum states and operators, MPS and MPO also provide an efficient way to calculate expectation values. For details, we refer to the literature (as, e.g., Ref. 1).

Remark. In this paper, we apply the term *translational invariant* also to finite systems (with open boundary conditions) and their Hamiltonians which are used in the iMPS algorithm to approach the infinite case.

B. Overview of the iMPS algorithm

Any algorithm which deals with infinite MPS could be addressed as an iMPS algorithm. In this paper, we use this term exclusively for algorithms of the type as presented in Ref. 22. This algorithm aims at finding an MPS representation for the ground state of an infinite one-dimensional quantum system. In its plain version, the iMPS algorithm takes translational invariance for granted such that all sites behave equally. Thus, all we need to construct the entire state is a perfect description of one site and its entanglement features with its environment given by the rest of the system. This environment is dominated by nearby neighbor sites, while the influence of sites far away can be neglected in any one-dimensional system with an asymptotic decay of correlations faster than r^{-1}. Therefore, the environment built up by an infinite system can be simulated with a finite system. Correspondingly, the center site of a sufficiently large but finite system provides a good approximation for its counterpart in the infinite case.

The iMPS algorithm is built upon a finite system, which is iteratively enlarged by inserting new sites into its middle. Since we express quantum states by MPS, each of these sites is represented by an individual tensor $A_{[n]}$ [Eq. (2)]. Before a tensor $A_{[n]}$ is inserted. it is optimized such that the resulting energy

$$E = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \quad (6)$$

is minimized. Hereby, all previously inserted tensors $A_{[j<n]}$ are left unchanged. Of course, these local optimizations of the new tensors $A_{[n]}$ are generally not sufficient to find the lowest-energy state of the *entire* system. What we are supposed to get is a ground state approximation which might be bad at the outer edges, but close to the center, it should become better with each new tensor inserted. This is all we need to obtain an adequate description of the center site's environment since the influence of the outer sites fades away with distance anyway. Therefore, we expect the environment of the center site to converge towards its infinite counterpart, and with that the new tensors $A_{[n]}$ inserted into each round should converge too:

$$A_{[n]} \to A_{[\text{converged}]}. \quad (7)$$

For ground states which violate translational invariance, it is no longer given that all sites behave equally. At this point, our intuitive argumentation breaks down and the algorithm generally fails to converge. We will study this case in Sec. III. For the time being, we assume that the algorithm ends up with a converged tensor $A_{[\text{converged}]}$. With the help of this one tensor, the entire iMPS can be constructed using the rules we will encounter in Sec. II C.

1. Long-range interactions

An important ingredient for a fast computer code is a clever bookkeeping of the interaction terms, which allows us to save many calculations due to recycling. In the case of long-range interactions, this task becomes tricky since the iMPS algorithm permanently splits the MPS and adds new sites. By this means, the distances between sites on the left and right halves change each time and with them all distance-dependent interaction terms. Nonetheless, for translational invariant Hamiltonians, recycling can still be done in a well-arranged fashion by encoding the Hamiltonian into an MPO as in Eq. (5). Every time the MPS is enlarged by a new tensor, the MPO is enlarged by the standard tensor $H^{\mu_l\mu_r}_{s's}$ too. This simple procedure automatically corrects all distance-dependent interaction terms.

C. Constructing the MPS

So far, we just mentioned that the iMPS algorithm constructs the MPS by constantly inserting new tensors in its middle. We will now specify on that. First, it is convenient to treat the MPS as divided into two halves, left and right from the center. Each time the optimization procedure described in Sec. II D provides a new tensor $A^{\alpha_l\alpha_r}_s$, we have to decide into which half $A^{\alpha_l\alpha_r}_s$ is to be absorbed. This is done under the following rules:

(1) According to the half in which $A^{\alpha_l\alpha_r}_s$ is to be absorbed, decompose it [Fig. 1(a)] as

$$A^{\alpha_l\alpha_r}_s = \begin{cases} Q^{\alpha_l\beta}_{[L]\,s} \cdot \lambda^{\beta\alpha_r}_{[L]} & \text{left half,} \\ \lambda^{\alpha_l\beta}_{[R]} \cdot Q^{\beta\alpha_r}_{[R]\,s} & \text{right half.} \end{cases} \quad (8)$$

Q is orthogonalized such that

$$Q^{\alpha_l\beta}_{[L]\,s} \cdot Q^{*\alpha_l\beta'}_{[L]\,s} = \delta^{\beta\beta'} = Q^{\beta\alpha_r}_{[R]\,s} \cdot Q^{*\beta'\alpha_r}_{[R]\,s}, \quad (9)$$

where the asterisk denotes the complex conjugation [see the upcoming Eq. (10) for details].

(2) With each new tensor A overwrite the λ of the tensor before.

Each $A_{[n]}$ is optimized in such a fashion that it compensates for the overwritten $\lambda_{[n-1]}$.

For the orthogonalization (9), we proceed as follows: In a first step, we write the tensor $A^{\alpha_l\alpha_r}_s$ as matrix $A^{m,a}$ where m is a multi-index. If the tensor $A^{\alpha_l\alpha_r}_s$ is to be absorbed into the left MPS half $m = (s,\alpha_l)$, otherwise $m = (s,\alpha_r)$. For the

leftmost and rightmost tensors of the MPS, which have only two indices, $m = s$. The index a corresponds to the leftover index of $A_s^{\alpha_l\alpha_r}$, which is not in m. Next, the matrix $A^{m,a}$ is decomposed into an orthogonal part Q and a "rest" part λ. Different decompositions would fulfill this task, but for the working of the algorithm it is best to resort to a singular value decomposition

$$A = U \cdot D \cdot V^\dagger = \underbrace{U \cdot V^\dagger}_{Q} \cdot \underbrace{V \cdot D \cdot V^\dagger}_{\lambda}. \tag{10}$$

Rewritten as tensors, we end up with Eq. (8).

A consequent application of these rules yields an MPS built of orthogonalized tensors Q and one single matrix λ from the very last $A_{[\mathrm{new}]}$ in the center [Fig. 1(b)]:

$$Q_{[L]\,s_1}^{[1]\,\alpha_1} \cdots Q_{[L]\,s_k}^{[k]\,\alpha_{k-1}\widetilde{\alpha}_k} \cdot \lambda^{\widetilde{\alpha}_k\alpha_k} Q_{[R]\,s_{k+1}}^{[k+1]\,\alpha_k\alpha_{k+1}} \cdots Q_{[R]\,s_n}^{[n]\,\alpha_{n-1}}. \tag{11}$$

This MPS standard form is numerical robust and has an easily calculated norm $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$. To see this, let us multiply $Q_{[L]\,s_1}^{[1]\,\alpha_1} \cdots Q_{[L]\,s_k}^{[k]\,\alpha_{k-1}\alpha_k}$ [the left half of equation (11)] with its complex conjugated

$$\begin{aligned}&\left(Q_{[L]\,s_1}^{[1]\,\alpha_1} \cdot Q_{[L]\,s_2}^{[2]\,\alpha_1\alpha_2} \cdots\right)\left(Q_{[L]\,s_1}^{*[1]\,\beta_1} \cdot Q_{[L]\,s_2}^{*[2]\,\beta_1\beta_2} \cdots\right)\\ &= \underbrace{\underbrace{\left(Q_{[L]\,s_1}^{[1]\,\alpha_1} \cdot Q_{[L]\,s_1}^{*[1]\,\beta_1}\right)}_{\delta^{\alpha_1\beta_1}} \left(Q_{[L]\,s_2}^{[2]\,\alpha_1\alpha_2} \cdot Q_{[L]\,s_2}^{*[2]\,\beta_1\beta_2}\right) \cdots}_{\delta^{\alpha_2\beta_2}}\\ &\qquad \vdots\\ &\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\delta^{\alpha_k\beta_k}}\end{aligned} \tag{12}$$

and analog for the right half. Thanks to Eq. (9), all QQ^* pairs turn into δ functions and the norm of the MPS (11) equals the remaining $\|\lambda\|$ [see Fig. 1 (c)], which also equals the norm of the last inserted tensor $A_{[\mathrm{new}]}$. Thus,

$$\|\mathrm{MPS}\| = \|A_{[\mathrm{new}]}\|. \tag{13}$$

D. Tensor optimization

The iMPS algorithm is an iterative procedure. As described in Sec. II B, new tensors $A_{[n]}$ are constantly inserted into the MPS, which represents the finite state ψ. Each of these new tensors $A_{[n]}$ is optimized such that the energy $E = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$ is minimized. As we will discuss in more detail in the following, this can be written as

$$\min \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \to \min \frac{\langle A_{[n]}|\mathbb{H}_{[n]}|A_{[n]}\rangle}{\langle A_{[n]}|\mathbb{I}_{[n]}|A_{[n]}\rangle}. \tag{14}$$

(i) $|A_{[n]}\rangle$ is the vectorized form of the tensor $A_{[n]}$, that is, $|A_{[n]}\rangle = A_{[n]}^i = A_{[n]\,s}^{\alpha_l\alpha_r}$ with the multi-index $i = (\alpha_l,\alpha_r,s)$.

(ii) $\mathbb{H}_{[n]}$ is an effective operator built of the MPO representation of the Hamiltonian H (5) and all MPS tensors of $\langle\psi|$ and $|\psi\rangle$ except for the two new $A_{[n]}$.

(iii) $\mathbb{I}_{[n]}$ is the identity operation thanks to the orthogonalized standard form of the MPS [see Eq. (12)].

Equation (14) is solved by setting $|A_{[n]}\rangle$ equal to the lowest eigenvector of $\mathbb{H}_{[n]}$.

Remark. We will repeatedly use the notation $|T\rangle$ or $\langle T|$ for a vectorized tensor T.

1. Effective operator $\mathbb{H}$

In Sec. II C, we mentioned that it is convenient to treat the MPS as divided into two halves: left and right from the newest tensor $A_{[n]}$. For the same reason, we decompose $\mathbb{H}_{[n]}$ into a left half $L_{[n]}^{\alpha_l'\mu_l\alpha_l}$ and a right half $R_{[n]}^{\alpha_r'\mu_r\alpha_r}$, which are connected by a single MPO tensor $H_{s_n's_n}^{\mu_l\mu_r}$ [Eq. (5)] corresponding to the new site [see Fig. 2(c)]

$$\mathbb{H}_{[n]}^{i',i} = \mathbb{H}_{[n]\,s_n's_n}^{\alpha_l'\alpha_l\alpha_r'\alpha_r} = L_{[n]}^{\alpha_l'\mu_l\alpha_l} \cdot H_{s_n's_n}^{\mu_l\mu_r} \cdot R_{[n]}^{\alpha_r'\mu_r\alpha_r} \tag{15}$$

with $i = (\alpha_l,\alpha_r,s_n)$.

Since the iMPS algorithm is an iterative procedure, $L_{[n]}$ and $R_{[n]}$ are built up iteratively, as well. Suppose we intend to absorb the $A_{[n-1]}$ of the previous optimization step into the left half. First, we use Eq. (8) to gain the orthogonal tensor $Q_{[L]\,s_{n-1}}^{\alpha_{\bar{l}}\alpha_l}$. With that

$$L_{[n]}^{\alpha_l'\mu_l\alpha_l} = L_{[n-1]}^{\alpha_{\bar{l}}'\mu_{\bar{l}}\alpha_{\bar{l}}} \cdot Q_{[L]\,s'}^{*\,\alpha_{\bar{l}}'\alpha_l'} \cdot H_{s',s}^{\mu_{\bar{l}}\mu_l} \cdot Q_{[L]\,s}^{\alpha_{\bar{l}}\alpha_l}, \tag{16}$$

where the asterisk denotes complex conjugation. In this case, where the tensor $A_{[n-1]}$ is absorbed into the left half, the right half stays unchanged $R_{[n]} = R_{[n-1]}$. Conversely, if we decide to absorb $A_{[n-1]}$ into the right half, the left half stays unchanged and R becomes

$$R_{[n]}^{\alpha_r'\mu_r\alpha_r} = Q_{[R]\,\alpha_r'\alpha_{\bar{r}}'}^{*\,\alpha_r'\alpha_{\bar{r}}'} \cdot H_{s',s}^{\mu_r\mu_{\bar{r}}} \cdot Q_{[R]\,s}^{\alpha_r\alpha_{\bar{r}}} \cdot R_{[n-1]}^{\alpha_{\bar{r}}'\mu_{\bar{r}}\alpha_{\bar{r}}}. \tag{17}$$

E. Algorithm

After having presented the decisive ingredients of the iMPS algorithm, we like to emphasize the steps one actually has to perform on the computer. The algorithm consists of an initializing procedure (see Sec. II E2) and an iteration loop, which is repeated until convergence is reached $A_{[n]} \to A_{[\mathrm{converged}]}$ [Eq. (7)]. The tensor $A_{[\mathrm{converged}]}$ is all we need to calculate expectation values. We do not hold any copy of the MPS we are calculating. The only objects stored (in the purest version of the algorithm) are the actual versions of $L_{[n]}^{\alpha_l'\mu_l\alpha_l}$ [Eq. (16)], $R_{[n]}^{\alpha_r'\mu_r\alpha_r}$ [Eq. (17)], and $A_{[n]\,s}^{\alpha_l\alpha_r}$.

1. Loop

(1) Calculate the new $A_{[n]\,s}^{\alpha_l\alpha_r} = A_{[n]}^i$ with $i = (\alpha_l,\alpha_r,s)$. Therefor,

(a) use Eq. (15) to calculate $\mathbb{H}_{[n]}^{i',i} = \mathbb{H}_{[n]\,s,s'}^{\alpha_l'\alpha_l\alpha_r'\alpha_r}$,

(b) set $A_{[n]}^i$ equal to the lowest eigenvector of $\mathbb{H}_{[n]}^{i',i}$.

(2) Decide whether to absorb $A_{[n]}^i$ into the left or right half (e.g., even steps left, odd steps right) and act accordingly in the following two steps.

(3) Use Eqs. (8) and (10) to decompose $A_{[n]}^i$ and gain $Q_{[L]}$ or $Q_{[R]}$.

(4) Use Eq. (16) or (17) to get $L_{[n+1]}^{\alpha_l'\mu_l\alpha_l}$ and $R_{[n+1]}^{\alpha_r'\mu_r\alpha_r}$.

2. Initialization

At the beginning, we have to initialize the values of $L^{\alpha_l'\mu_l\alpha_l}$ and $R^{\alpha_r'\mu_r\alpha_r}$. This can be done with the help of an exact solution ψ for a small system best with an even number of sites $n = 2k$ (we mostly used $n = 8$ or 10 for two-level sites). The state ψ

is split into a left and right part, which allows us to calculate L and R. In more details, note the following:

(1) Write the Hamiltonian of the small system as matrix with multi-indices $H^{(s'_1\ldots s'_n),(s_1\ldots s_n)}$ and solve for $\psi^{(s_1\ldots s_n)}$.

(2) Split the multi-index in two multi-indices, giving $\psi^{(s_1\ldots s_k),(s_{k+1}\ldots s_n)}$, which can be interpreted as a matrix.

(3) Use a singular value decomposition $\psi = U \cdot D \cdot V^\dagger$ (or Takagi's factorization $\psi = U \cdot D \cdot U^T$ if $\psi = \psi^T$).

(4) Interpret the index structure of U as $U^{i,j} = U^{i,\alpha_k} = U^{(s_1\ldots s_k),\alpha_k}$.

(5) $L^{\alpha'_k\mu_k\alpha_k} = U^{*s'_1\ldots s'_k,\alpha'_k} \cdot H^{s'_1\ldots s'_k,\mu_k}_{s_1\ldots s_k} \cdot U^{s_1\ldots s_k,\alpha_k}$, $H^{s'_1\ldots s'_k,\mu_k}_{s_1\ldots s_k} = H^{[L]\mu_1}_{s'_1 s_1} \cdot H^{\mu_1\mu_2}_{s'_2 s_2} \ldots H^{\mu_{k-1}\mu_k}_{s'_k s_k}$ [Eq. (5)].

(6) Use V analog to U to calculate $R^{\alpha'_r\mu_r\alpha_r}$.

Comparing with (11), we find that

$$U = Q^{[1]\alpha_1}_{[L]s_1} \ldots Q^{[k]\alpha_{k-1}\widetilde{\alpha}_k}_{[L]s_k}, \quad D = \lambda^{\widetilde{\alpha}_k\alpha_k},$$
$$V = Q^{[k+1]\alpha_k\alpha_{k+1}}_{[R]s_{k+1}} \ldots Q^{[n]\alpha_{n-1}}_{[R]s_n}.$$

III. BROKEN TRANSLATIONAL INVARIANCE

In this section, we present some concepts for the iMPS algorithm which arise from the need to deal with spontaneously broken translational invariance. A principal shortcoming of the basic iMPS algorithm is its failure to converge in such cases. To overcome this deficiency, we introduce the superposed multioptimization (SMO) method in Sec. III B. Once convergence is restored, we turn our attention in Sec. III C to the question of how to obtain a specific solution out of the ground state manifold degenerate due to broken translational invariance. In addition, in Sec. III D, we treat the special case of a nondegenerate ground state which is separated by a very small energy gap from a state that breaks translational invariance.

A. Preliminary considerations

Just one tensor $A_{[\text{converged}]}$ suffices to construct an entire iMPS. At first sight, one might therefore think that such an iMPS is only capable of describing states where all sites behave equally, which is no longer true for states which break translational invariance. But still, also these states can be handled. This is due to the construction rules presented in Sec. II C. These result in an iMPS structure given by Eq. (11), where the matrix λ marks a special position (and with that breaks translational invariance) if it can not be commuted to its neighbor sites.

The real problem is to *find* $A_{[\text{converged}]}$. In Sec. II B, we already mentioned that our argumentation in favor of the convergence $A_{[n]} \to A_{[\text{converged}]}$ is no longer valid in the case of broken translational invariance. The iMPS algorithm is grounded on local optimization and therefore it is vulnerable to locally altering states, as they appear on a physical level for states with broken translational invariance. In this case, simple local optimization will not result in a global optimal fixed point $A_{[\text{converged}]}$.

1. Known solutions

When we write about the breakdown of translational invariance, we mean that the state is no longer invariant under the shift of one site. Still, the state can maintain invariance under the shift of k sites. If k is known, one can introduce new supersites where one supersite encompasses k of the old sites. Now, the system is translational invariant with respect to the shift of one supersite. This involves that we have to optimize tensors which represent the supersites. Due to the exponential increase of the physical dimension, this method is practical for very small k only. To avoid the scaling problem, Crosswhite[22] suggested to use an MPS ansatz for the supersites. In practice, this means we insert k old sites at once and optimize the corresponding tensors. We are not aware if this was ever tested successfully. Aside from that, one still needs a prior knowledge of the value k.

Alternatively, one can extend the standard MPS structure with auxiliary tensors,[26,27] which allow us to introduce a symmetry-breaking element for the price of a nonlinear optimization. To our knowledge, this has not been tested for long-range interactions so far.

B. Superposed multioptimization

Our solution of the convergence problem induced by locally altering states does not depend on any prior knowledge and stays within the standard MPS framework. The key idea is to wash out local dependency of the optimization by choosing each new tensor $A_{[n]}$ such that it minimizes the sum of the energy of exponentially many different MPS instead of just one. These MPS are different ground state approximations to the qualitatively same Hamiltonian applied to systems of different sizes. All necessary minimizations can be joined in a superposition and solved by one optimization. The time-relevant steps stay the same as in the single MPS optimizing algorithm presented so far, so there is no noticeable loss in speed.

As explained in the following, after each optimization round, the number of MPS joined in the superposition increases by a factor of 4. Thus, in the nth round, the optimization (14) gets formally extended to

$$\begin{aligned}\min\left(\sum_{i=1}^{4^{n-1}}\langle\psi_i|H_i|\psi_i\rangle\right) &\to \min\left(\sum_{i=1}^{4^{n-1}}\langle A_{[n]}|\mathbb{H}_{[n]i}|A_{[n]}\rangle\right)\\ &= \min\left(\langle A_{[n]}|\sum_{i=1}^{4^{n-1}}\mathbb{H}_{[n]i}|A_{[n]}\rangle\right)\\ &= \min(\langle A_{[n]}|\widetilde{\mathbb{H}}_{[n]}|A_{[n]}\rangle). \qquad (18)\end{aligned}$$

The MPS representing the $|\psi_i\rangle$ are of different length and the position of the tensor $A_{[n]}$ is no longer in the center but varies from MPS to MPS. In the basic algorithm, the tensors $A_{[n]}$ experience quite individualized environments and the optimization adapts to these local circumstances. In the modified algorithm, $A_{[n]}$ faces exponentially many different environments averaging out local effects and emphasizing common global features. This enforces heavily the desired convergence $A_{[n]} \to A_{[\text{converged}]}$.

1. Modification of the algorithm

Formally, the superposition $\widetilde{\mathbb{H}}_{[n]} = \sum_{i=1}^{4^{n-1}} \mathbb{H}_{[n]i}$ in Eq. (18) is based on $2^{2(n-1)} = 4^{n-1}$ different MPS. These MPS are

not hand picked, but indirectly generated by the algorithm. The only modification needed to create such a superposition concerns the left and right halves $L^{\alpha_l'\mu_l\alpha_l}$ and $R^{\alpha_r'\mu_r\alpha_r}$. We still use Eqs. (16) and (17) to perform the iteration steps $L^{\alpha_l'\mu_l\alpha_l}_{[n-1]} \to L^{\alpha_l'\mu_l\alpha_l}_{[n]}$ and $R^{\alpha_r'\mu_r\alpha_r}_{[n-1]} \to R^{\alpha_r'\mu_r\alpha_r}_{[n]}$, but afterwards we add the new and the old results to achieve superpositions

$$L^{\alpha_l'\mu_l\alpha_l}_{[n]} \leftarrow L^{\alpha_l'\mu_l\alpha_l}_{[n-1]} + L^{\alpha_l'\mu_l\alpha_l}_{[n]}, \quad R^{\alpha_r'\mu_r\alpha_r}_{[n]} \leftarrow R^{\alpha_r'\mu_r\alpha_r}_{[n-1]} + R^{\alpha_r'\mu_r\alpha_r}_{[n]}. \tag{19}$$

Since this is a simple addition of two tensors, the structure and size of $L^{\alpha_l'\mu_l\alpha_l}_{[n]}$ and $R^{\alpha_r'\mu_r\alpha_r}_{[n]}$ stay the same and do not entail any computational complications.

In the basic algorithm, we have to decide at each iteration step whether to absorb the tensor $A_{[n-1]}$ into the left half $L^{\alpha_l'\mu_l\alpha_l}_{[n]}$ [Eq. (16)] or into the right half $R^{\alpha_r'\mu_r\alpha_r}_{[n]}$ [Eq. (17)]. Only the half of choice is modified. Now, we symmetrize the algorithm and modify both halves in each step. With this modification, the operator sum $\widetilde{\mathbb{H}}_{[n]}$ is calculated with one single use of Eq. (15). As a further advantage of the symmetrization, the modified iMPS algorithm can now take advantage of mirror-symmetric Hamiltonians. As this subject is a bit off topic, we refer the interested reader to Appendix F.

Since each of the tensors $A_{[1]}, \ldots, A_{[n-1]}$ is absorbed into the left half L and the right half R, the longest MPS encoded in the operator superposition of $\widetilde{\mathbb{H}}_{[n]}$ contains $2(n-1)$ tensors (where we neglect the contribution from the initialization routine explained in Sec. II E2 and the hole for the new tensor $A_{[n]}$). But, in the general MPS, each of the $2(n-1)$ tensors only appears with a probability of 50% due to the addition of the new and old L and R in Eq. (19). The possible tensor combinations give rise to the heralded $2^{2(n-1)}$ different MPS from which $\binom{k}{2(n-1)}$ are of length k. More precisely, $\binom{k_{[L]}}{(n-1)} \cdot \binom{k_{[R]}}{(n-1)}$ of these MPS contain $k_{[L]}$ tensors left and $k_{[R]}$ tensors right from the hole for the new tensor $A_{[n]}$.

2. *Comments*

The crucial observation is that the iteration steps to perform are always the same independent of the tensor position and the size of the MPS. Therefore, all the different MPS can be optimized together combined in a superposition. The reader who is more familiar with finite MPS calculations might wonder about the complete loss of information concerning the single MPS in the superposition, which comes along with Eq. (19). We have to remind ourselves that the main objective of the iMPS algorithm is to get the tensor $A_{[\text{converged}]}$, which suffices to construct the infinite MPS. The finite MPS are just tools to obtain this tensor. Once we have it, the finite MPS are no longer needed.

Another interesting question is whether the operator sum $\widetilde{\mathbb{H}}_{[n]}$ in Eq. (18) constructed via Eq. (19) is really suitable for a variational ansatz. This question has two aspects.

(1) Is every optimal iMPS solution of Eq. (18) also a minimum of the Hamiltonian?

(2) Does the algorithm always converge to this optimal solution?

The intuitive argumentation given in Sec. II B also suggests that the answer to the second question should be yes, but actually even for the well-established DMRG algorithm the answer has to be no, since otherwise NP hard problems could be solved. Nonetheless, it is a matter of fact that DMRG converges extremely well for most practical purposes. The same "practical proof" can be given for Eq. (18) as demonstrated by our applications (Sec. V).

The first question can be answered more formally. Equation (18) represents the sum of exponentially many energy terms. The lowest conceivable value of this sum is reached, if each energy term takes its individual minimal value. If all individual energies are minimized, the obvious answer to the first question is yes. Hence, we have to ask whether it is possible to minimize all individual energies at once, having in mind that all MPS involved are created indirectly via Eq. (19). With finite MPS, this might only be possible up to a certain relative error. But for the limit of infinite MPS this relative error shrinks to zero and the problem is trivially solved by uniform iMPS, i.e., in the case where all tensors stem from the same $A_{[\text{converged}]}$. Then, all iMPS appearing in Eq. (18) look alike and either none or all of them minimize their Hamiltonians. Even in the case of broken translational invariance, one can always find at least one translational invariant ground state, which can be written as uniform iMPS and hence optimizes all terms in the sum at once.

Next, we like to further inspect the numerical consequences of Eq. (19) for the different MPS which are part of the operator sum $\widetilde{\mathbb{H}}_{[n]}$. The tensor $A_{[n-1]}$ was absorbed into exactly half of the superpositions encoded in $L_{[n]}$ and in $R_{[n]}$. Since $L_{[n]}$ and $R_{[n]}$ are the building blocks of $\widetilde{\mathbb{H}}_{[n]}$ [Eq. (15)], four subsets of $\widetilde{\mathbb{H}}_{[n]}$ can be distinguished:

(1) $A_{[n-1]}$ was neither absorbed into $L_{[n]}$ nor into $R_{[n]}$.
(2) $A_{[n-1]}$ was only absorbed into $L_{[n]}$.
(3) $A_{[n-1]}$ was only absorbed into $R_{[n]}$.
(4) $A_{[n-1]}$ was absorbed into both halves $L_{[n]}$ and $R_{[n]}$.

Although we never experienced any practical problems, the cases 1 and 4 are, at least from the theoretical point of view, a bit troublesome. In case 4, the tensor $A_{[n-1]}$ is inserted twice. But, $A_{[n-1]}$ was never optimized for double insertion. Close to the end, when $A_{[n]} \to A_{[\text{converged}]}$ is almost achieved, this should pose no problem. Meanwhile, at an early stage the effect should be more severe. On the other hand, even in the basic algorithm, the MPS description is not perfect, especially not at the beginning.

Case 1 might seem trivial since everything stays the same. Potential difficulties arise in the superposition with the other cases. According to Eq. (8), the tensors A are decomposed into Q and λ and only Q is absorbed. The matrix λ is overwritten with the next A (respectively Q). In the basic algorithm, this is easy to justify: The next $A_{[n]}$ can compensate for $\lambda_{[n-1]}$. But, a perfect compensation can only be achieved for *one*λ, not for many of them. Here is the problem: In case 1, old $\lambda_{[n-2]}, \lambda_{[n-3]}, \ldots$ of the previous steps are conserved, while in cases 2, 3, and 4, a new $\lambda_{[n-1]}$ comes into play. All λ have to be compensated for. The stronger the λ alter, the less adequate is their compensation. The variation of the λ can be reduced by enforcing $\|A_{[n]} - A_{[n-1]}\|$ to be small. Towards the end of the optimization, $\|A_{[n]} - A_{[n-1]}\|$ is small anyway. At an early stage, one might have to resort more strongly to the convergence enforcing method we will discuss in Sec. IV A2.

C. Selecting a specific ground state

We have seen how to ensure convergence in the case of broken translational symmetry. But so far, we have no control to which of the degenerate ground states the algorithm converges. Some of these ground states might be more favorable for our purposes than others, and we now answer the question as to how to obtain them. Any further degeneration aside from broken translational invariance is excluded from this consideration.

Let us look at two fully converged MPS $\mathcal{A}$ and $\mathcal{B}$ where $\mathcal{B}$ is a representation of the wished-for ground state which fits our purposes best, while $\mathcal{A}$ stands for any ground state to which the algorithm actually has converged. According to Eq. (11), both MPS have the following structure:

$$\begin{aligned}\mathcal{A} &= \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0} \cdot \lambda^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]s_2} \ldots, \\ \mathcal{B} &= \ldots q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}} \cdot q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0} \cdot \xi^{\tilde{\alpha}_0\alpha_0} \cdot q^{\alpha_0\alpha_1}_{[R]s_1} \cdot q^{\alpha_1\alpha_2}_{[R]s_2} \ldots .\end{aligned} \tag{20}$$

In Appendix C, we show that it suffices to replace the matrix $\lambda^{\tilde{\alpha}_0\alpha_0}$ in $\mathcal{A}$ by the new matrix $\gamma^{\tilde{\alpha}_0\alpha_0}$ to obtain an MPS which represents exactly the same physical state as $\mathcal{B}$:

$$\mathcal{B} = \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0} \cdot \gamma^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]s_2} \ldots . \tag{21}$$

In other words, we do not need to take care to which ground state the algorithm converges since after it has converged, we are able to transform the obtained solution easily into any other. We do not even have to know $\mathcal{B}$, as long as we have a description such as, e.g., "the ground state with the highest expectation value for the operator $\widehat{X}$." All we have to do is a one-time optimization of the new matrix $\gamma^{\tilde{\alpha}_0\alpha_0}$ under the desired side condition.

1. One-tensor update versus multitensor update

So far, we focused on uniform MPS which result from an algorithm that inserts one new tensor each round. Crosswhite[22] suggested that one might also insert a certain number of q tensors per round in the form of a small MPS. Although we are so far not aware of any successful practical applications of this ansatz, it is worth having a closer look. For the single-site algorithm to work in the presence of broken translational invariance, we introduced the SMO method, which washes out local variations and thereby fortifies convergence. Still, the SMO is a general method and could also be implemented in a q-site algorithm.

As we have seen in the section above, the single-site iMPS algorithm will come up with a solution that encodes all possible ground states. This abundance has its price. Given the situation that we know the periodicity q of the ground state of interest, we could use an iMPS algorithm which inserts q sites at once. This would enable us to find some specific lowly entangled ground states which could be expressed by a nonuniform MPS with a far smaller bond dimension. Since q translationally shifted copies of such a nonuniform MPS always allow us to construct a uniform MPS, the maximal gain in bond dimension is given by a factor q and the maximal difference in the entanglement entropy of the half chain is $\Delta S = \ln_2(q)$. We could confirm this difference for the model studied in the applications (Sec. V) varying the matrix λ [Eq. (20)] over the set of ground states, as described in the section above.

From the perspective of the needed bond dimension, the multitensor update is clearly superior to the single-tensor update for systems with a periodicity $q > 1$. Still, in this paper, we favor the single-tensor update, which needs no prior knowledge of the periodicity and results in a well-converging algorithm, which has proofed its reliability in practical tests.

2. Degenerate tensor

In the case of broken translational invariance, one can jump from one ground state solution to another just by changing the matrix λ, which is part of the bigger tensor $A_{[n]}$ [Eq. (8)]. Hence, different $A_{[n]}$ minimize $\langle A_{[n]}|\widetilde{\mathbb{H}}_{[n]}|A_{[n]}\rangle$, i.e., $A_{[n]}$ is degenerate. The iMPS algorithm aims for the convergence $A_{[n]} \to A_{[\text{converged}]}$. Without precautions, this convergence might be undermined towards the very end by an $A_{[n]}$ which jumps from one solution to another. At first glance, this does not seem troublesome because all solutions $A_{[n]}$ could jump to are good solutions. Nonetheless, due to imperfect numerics, this jumping might also occur into $A_{[n]}$ of minor quality. This effect is not fatal, but it still might turn an otherwise perfect result into a less accurate one.

To suppress this effect, we can resort to the convergence enforcing method we will present in Sec. IV A2. In addition, we will describe in Secs. III D and IV D a numerical method to enforce a translational invariant solution which eliminates the above-mentioned degeneration.

D. translational invariant ground states and local minima

In this section, we consider possible convergence problems due to translational invariance-breaking states which lie closely above the nondegenerate ground state level. In such cases, the infinite system still provides a translational invariant ground state, while for finite systems even small alterations of the energy spectrum due to boundary effects suffice to favor a ground state with broken translational invariance. Since the iMPS algorithm is based on growing finite systems, it might start out converging into a false minimum and get trapped there. Even if the algorithm escapes out of this trap later, it supposedly costs many optimization rounds and significantly slows down convergence.

To avoid these problems, we suggest to modify the algorithm such that it only converges to translational invariant states. This is no limitation: In the case of a unique ground state, the sole solution has to be translational invariant, anyway. If the ground state level is degenerate, one of the solutions is translational invariant and according to Eq. (21), we can still transform it into another type of solution after the algorithm has converged.

Whether the fully converged MPS $\mathcal{A}$,

$$\mathcal{A} = \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0} \cdot \lambda^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]s_2} \ldots, \tag{22}$$

is translational invariant or not depends on its matrix λ. At this point, we should be more precise and write $\lambda_{[L]}$ or $\lambda_{[R]}$, depending on whether λ stems from a left or a right decomposition (8). Actually, as a consequence of decomposition (8), the MPS $\mathcal{A}$ is translational invariant if the

left and right versions of λ are identical:

$$\lambda_{[L]} = \lambda_{[R]} = \lambda \Rightarrow Q_{[L]} \cdot \lambda = A_{[\text{converged}]} = \lambda \cdot Q_{[R]}. \tag{23}$$

In this case, λ can be commuted to any position and hence no longer marks any specific site of the MPS. This is what we are aiming for.

In order to end up with an $A_{[\text{converged}]}$ where $\lambda_{[L]} = \lambda_{[R]}$, we alter the minimization routine which computes the tensors A such that solutions with small differences $\|\lambda_{[L]} - \lambda_{[R]}\|$, i.e., big overlap $\langle\lambda_{[L]}|\lambda_{[R]}\rangle$ are preferred. In the long run, this should accumulate to $\lambda_{[L]} = \lambda_{[R]}$.

As a first straightforward way, we tried to extend the minimization (18) of $\langle A|\widetilde{\mathbb{H}}|A\rangle$ to

$$\min(\langle A|\widetilde{\mathbb{H}}|A\rangle - \gamma_{[\lambda]} \cdot \langle\lambda_{[L]}|\lambda_{[R]}\rangle) \tag{24}$$

with a suitable coupling parameter $\gamma_{[\lambda]}$. This is no longer a simple to solve bilinear problem since one needs to perform the decomposition (10) to get $\lambda_{[L]}$ and $\lambda_{[R]}$. To avoid this complication and restore bilinearity, we tried to resort to the easily calculated approximations $\bar{\lambda}_{[L]}$ and $\bar{\lambda}_{[R]}$ [Eq. (E2) derived in Appendix E], but the results we obtained in this way were not very convincing.

In Sec. IV D we introduce a less conventional approach which turned out to work far more satisfyingly for us. Instead of extending the minimization of $\langle A|\widetilde{\mathbb{H}}|A\rangle$ by a new term as suggested in Eq. (24), we alter the routines of the iterative eigenvector solver we use to solve it. The modus operandi of these solvers is reviewed in Sec. IV C. Until after then, we suspend further explanations.

IV. ENHANCED ALGORITHM

The considerations of the last section were mainly conceptual. The only actual change of the algorithm we performed is given by Eq. (19), which incorporates the SMO method. In this section, we delve far more into numerical details and extend the algorithm by further routines to make it more efficient. A reader not interested in technical details of the algorithm might proceed directly to Sec. V.

A. Enforcing convergence

The goal of the iMPS algorithm is the global convergence $A_{[n]} \to A_{[\text{converged}]}$. This property has to emerge over the long term, while it is not part of the evaluation system of the local minimization from which each $A_{[n]}$ is drawn. As a consequence, small local improvements might be purchased with strong fluctuating $A_{[n]}$ counteracting global convergence. In an unstable scenario of overcompensation, these fluctuations might even inflate in a fatal manner. To prevent this from happening, we extend the algorithm by two methods. The first method (superposition method) aims at attenuating the influence of problematic $A_{[n]}$ on the ongoing calculations, while the second method (gain function method) directly modifies the optimization routine such that excessive variation of the $A_{[n]}$ are suppressed. Both methods are complementary and worked well together in our calculations.

1. Superposition method

The first method takes advantage of the fact that the $\widetilde{\mathbb{H}}_{[n]}$ [Eq. (18)] of the modified algorithm represent superpositions of operators. By decreasing the weight of those contributions to the superpositions which contain problematic $A_{[n]}$, one can ensure that excessive fluctuation of the $A_{[n]}$ do not spread to the level of the $\widetilde{\mathbb{H}}_{[n+1]}$ and with that inhibit a chain of overcompensation. We remind the reader that $A_{[n]}$ is absorbed into $L_{[n+1]}$ [Eq. (16)] and $R_{[n+1]}$ [Eq. (17)] before Eq. (19) is used to build up superpositions. This latter equation is now replaced by

$$\begin{aligned} L_{[n+1]}^{\alpha_l'\mu_l\alpha_l} &\leftarrow L_{[n]}^{\alpha_l'\mu_l\alpha_l} + \xi_{[n]} \cdot L_{[n+1]}^{\alpha_l'\mu_l\alpha_l}, \\ R_{[n+1]}^{\alpha_r'\mu_r\alpha_r} &\leftarrow R_{[n]}^{\alpha_r'\mu_r\alpha_r} + \xi_{[n]} \cdot R_{[n+1]}^{\alpha_r'\mu_r\alpha_r}. \end{aligned} \tag{25}$$

The only new ingredient compared to Eq. (19) is the adjustable weight $1 \geqslant \xi_{[n]} > 0$ calculated as

$$\xi_{[n]} = \min\left(1, \frac{\langle\Delta A\rangle_{[n]}}{\Delta A_{[n]}}\right), \tag{26}$$

where $\Delta A_{[n]}$ measures the deviation of $A_{[n]}$ and $\langle\Delta A\rangle_{[n]}$ is a weighted average of the previous deviations. Each time the deviation $\Delta A_{[n]}$ exceeds the average value $\langle\Delta A\rangle_{[n]}$, $\xi_{[n]}$ gets smaller than 1 and with that the weight of all contributions of $\widetilde{\mathbb{H}}$ which contain $A_{[n]}$ is reduced accordingly.

To measure the deviation $\Delta A_{[n]}$, we need to define a reference tensor $A_{[n]}^{[\text{refer}]}$ such that $\Delta A_{[n]} = \|A_{[n]} - A_{[n]}^{[\text{refer}]}\|$. In order to avoid unnecessary fluctuation of this reference tensor, we use the same trick as above and define $A_{[n]}^{[\text{refer}]}$ iteratively as a weighted average of the previous $A_{[0\leqslant j<n]}$:

$$A_{[n+1]}^{[\text{refer}]} = \frac{1}{N} \cdot \left(A_{[n]}^{[\text{refer}]} + \xi_{[n]} \cdot A_{[n]}\right), \tag{27}$$

with $N = \|A_{[n+1]}^{[\text{refer}]}\|$. The weights $\xi_{[n]}$ used in Eq. (27) are the same as in Eqs. (25) and (26).

For Eq. (26) to work, we still have to define the weighted average $\langle\Delta A\rangle_{[n]}$. Various definitions are possible. As a heuristic choice, we picked the following one:

$$\begin{aligned} \langle\Delta A\rangle_{[n]} = \frac{1}{N} \cdot \min(&0.9 \cdot \langle\Delta A\rangle_{[n-1]} + 0.1 \cdot \Delta A_{[n-1]}, \\ &1.02 \cdot \langle\Delta A\rangle_{[n-1]}), \end{aligned} \tag{28}$$

with $N = 1\text{–}0.9^n$. Obviously, the term $1.02 \cdot \langle\Delta A\rangle_{[n]}$ prevents a too sudden increase of $\langle\Delta A\rangle_{[n]}$ by limiting it to 2% per round. Without this term, we get the clearer expression $\langle\Delta A\rangle_{[n]} \sim \Sigma_{j=0}^{n} 0.9^{n-j} \cdot \Delta A_{[j]}$, i.e., older $\Delta A_{[j]}$ lose in each round 10% of their influence in the weighted average.

Finally, we remark that we end up in a deadlock if $\xi_{[n]} = 0$. To prevent this from happening, we will introduce the parameter $\Delta_{\max}$ in Eq. (30) of the upcoming section.

2. Gain function method

The idea of the gain function method is to manipulate the minimization procedure of $\langle A_{[n]}|\widetilde{\mathbb{H}}_{[n]}|A_{[n]}\rangle$ by adding a gain function, i.e., replacing $\widetilde{\mathbb{H}}_{[n]}$ by $\widetilde{\mathbb{H}}_{[n]}^{[\gamma]}$:

$$\widetilde{\mathbb{H}}_{[n]}^{[\gamma]} = \widetilde{\mathbb{H}}_{[n]} - \gamma \cdot \left|A_{[n]}^{[\text{refer}]}\right\rangle\left\langle A_{[n]}^{[\text{refer}]}\right| \quad \text{with} \quad \gamma \geqslant 0, \tag{29}$$

where $|A^{[\mathrm{refer}]}_{[n]}\rangle$ is the vectorized version of the reference tensor defined in Eq. (27). Let $A^{[\gamma]}_{[n]}$ be the result of the above optimization. Obviously, bigger values for γ favor smaller deviations $\Delta A^{[\gamma]}_{[n]} = \|A^{[\gamma]}_{[n]} - A^{[\mathrm{refer}]}_{[n]}\|$.

In Appendix B, we show how to approximate γ efficiently such that

$$\Delta A^{[\gamma]}_{[n]} \approx \min\left(c_{[n]} \cdot \Delta A^{[\gamma=0]}_{[n]}, \Delta_{\max}\right), \tag{30}$$

where $0 < c_{[n]} \leqslant 1$ and $\Delta_{\max}$ are parameters of our choice. Limiting $\Delta A^{[\gamma]}_{[n]}$ by assigning, e.g., $\Delta_{\max} = 10\langle \Delta A\rangle_{[n]}$ [Eq. (28)] ensures that $\xi_{[n]}$ [Eq. (26)] is lower bounded around 0.1.

Assigning the parameter $0 < c_{[n]} \leqslant 1$ [Eq. (30)] allows us to shorten $\Delta A^{[\gamma]}_{[n]}$ to a chosen fraction of the maximal value $\Delta A^{[\gamma=0]}_{[n]}$. The price to pay for a $c_{[n]} < 1$ is a lesser energy improvement $\Delta \tilde{E}^{[\gamma]}_{[n]}$ which is calculated as the difference between the energy one gets due to choosing $A_{[n]} = A^{[\gamma]}_{[n]}$ instead of just taking $A_{[n]} = A^{[\mathrm{refer}]}_{[n]}$:

$$\begin{aligned}\Delta \tilde{E}^{[\gamma]}_{[n]} &= \left\langle A^{[\gamma]}_{[n]} \middle| \widetilde{\mathbb{H}}_{[n]} \middle| A^{[\gamma]}_{[n]} \right\rangle - \left\langle A^{[\mathrm{refer}]}_{[n]} \middle| \widetilde{\mathbb{H}}_{[n]} \middle| A^{[\mathrm{refer}]}_{[n]} \right\rangle \\ &\approx \Delta \tilde{E}^{[\gamma=0]}_{[n]} \cdot [1 - (1 - c_{[n]})^2]. \end{aligned} \tag{31}$$

Choosing, e.g., a γ which corresponds to $c_{[n]} \approx 0.9$ reduces $\Delta A^{[\gamma]}_{[n]}$ by 10%, while the energy improvement $\Delta \tilde{E}^{[\gamma]}_{[n]}$ is still at 99% of the maximal value $\Delta \tilde{E}^{[\gamma=0]}_{[n]}$.

The parameter $\Delta_{\max}$ [Eq. (30)] and the entire superposition method are designed to intervene only in the case that $\Delta A_{[n]}$ suddenly increases with ongoing n; otherwise, they have no effect. The parameter $c_{[n]}$ on the other hand always effects the calculation if chosen to be smaller than 1. Generally, the $c_{[n]}$ should be chosen in dependence of $\Delta A^{[\gamma=0]}_{[n]}$ (the bigger $\Delta A^{[\gamma=0]}_{[n]}$, the smaller $c_{[n]}$ and vice versa). Just for orientation (not as exclusive choice), we give the value we chose for most of our calculations:

$$c_{[n]} = 1 - \max\left[0.1;\ 0.7 + 0.1 \cdot \log_{10}\left(\Delta A^{[\gamma=0]}_{[n]}\right)\right]. \tag{32}$$

With that, $0.269 < c_{[n]} \leqslant 0.9$ since $\Delta A^{[\gamma]}_{[n]} \leqslant 2$. This formula was found heuristically and worked fine for us, although more adequate choices might exist.

When the iMPS algorithm finally approaches its end, γ becomes very small and its effect might be overruled by numerical imprecision. To prevent this, we recommend defining a lower limit for γ above the limit of the numerical precision.

B. Energy overgrow

If the average energy per site of an infinite state does not equal zero, the total energy of the entire state is $\pm\infty$. Of course, we never have to deal with an infinite value since our numeric is restricted to finite systems. Nonetheless, a problem remains. In the long run, the numeric value of all the information encoded in the tensor $\widetilde{\mathbb{H}}$ stays more or less the same except for the energy, which grows with each new site. The tensor $\widetilde{\mathbb{H}}$ gets more and more ill conditioned since the numeric value of the energy overgrows other information and thereby reduces the achievable precision. To avoid this problem, we advise to subtract from each iteration step the energy $E_{[n]} = \langle A_{[n]}|\widetilde{\mathbb{H}}_{[n]}|A_{[n]}\rangle$ from the system. Simply speaking, we recommend to assign

$$\widetilde{\mathbb{H}}_{[n+1]} \leftarrow \widetilde{\mathbb{H}}_{[n+1]} - E_{[n]} \cdot \mathbb{I}. \tag{33}$$

To be of any use, this simple assignment has to be encoded into $L^{\alpha'_l \mu_l \alpha_l}_{[n+1]}$ and $R^{\alpha'_r \mu_r \alpha_r}_{[n+1]}$ the building blocks of $\widetilde{\mathbb{H}}_{[n+1]}$ [Eq. (15)]. This can be done by modifying the MPO tensor $H^{\mu_l \mu_r}_{s's}$ used in Eqs. (16) and (17). As shown in Appendix A, the MPO tensor $H^{\mu_l \mu_r}_{s's}$ has a slot which represents a local interaction term. To this local interaction we add $-E_{[n]} \cdot \mathbb{I}_{s's}$.

C. Minimization routine and information recycling

With an increasing number of rounds n, the successive minimizations of the different $\langle A_{[n]}|\widetilde{\mathbb{H}}_{[n]}|A_{[n]}\rangle$ become more and more similar, which opens the opportunity to speed up the minimization recycling information from preceding turns. In order to understand these ideas (and also those of Secs. IV D, IV E, and Appendix B 1), we have to review the principles of the iterative eigenvector solvers we use.[28] In the MPS context, these solvers come with the major advantage that $\widetilde{\mathbb{H}}_{[n]}$ never has to be constructed explicitly; it suffices to be able to assemble $\widetilde{\mathbb{H}}_{[n]}|A\rangle$ for any given $|A\rangle$. Further, we do not need to perform the minimization to its very end. For the algorithm to work, it suffices to perform a limited amount of iterations, such that the resulting $|A\rangle$ might not be optimal but still significantly improved. Due to information recycling, these improvements accumulate, such that the optimal solution emerges in the long run.

Iterative eigenvector solvers are very well suited for the outer eigenvalue spectrum. Already with modest effort we can expect to find a good approximation $|e_0\rangle \approx |E_0\rangle$ for the lowest eigenvector of $\widetilde{\mathbb{H}}_{[n]}$. The central idea is to project the problem defined on a huge space of dimension N onto a much smaller subspace of dimension $k \ll N$ and solve it there. For this to work, we have to build up iteratively a small set $\{|\mathfrak{A}_1\rangle, \ldots, |\mathfrak{A}_k\rangle\}$ of k orthonormal vectors which enables us to express the minimizing eigenvector $|A^{[\mathrm{min}]}_{[n]}\rangle = |E_0\rangle$ of $\widetilde{\mathbb{H}}_{[n]}$ as a linear combination

$$\left|A^{[\mathrm{min}]}_{[n]}\right\rangle = |E_0\rangle \approx |e_0\rangle = \sum_{i=1}^{k} |\mathfrak{A}_i\rangle \cdot a^{[\mathrm{min}]}_i, \tag{34}$$

$$\begin{aligned} E_0 &= \langle E_0|\widetilde{\mathbb{H}}_{[n]}|E_0\rangle \approx a^{[\mathrm{min}]\,\dagger}_i \cdot \langle \mathfrak{A}_i|\widetilde{\mathbb{H}}_{[n]}|\mathfrak{A}_j\rangle \cdot a^{[\mathrm{min}]}_j \\ &= a^{[\mathrm{min}]\,\dagger}_i \cdot \mathfrak{H}^{[n]}_{ij} \cdot a^{[\mathrm{min}]}_j = e_0. \end{aligned} \tag{35}$$

We need to solve for $a^{[\mathrm{min}]}_i$, which is obviously the minimizing eigenvector for the $k \times k$ matrix $\mathfrak{H}^{[n]}_{ij} = \langle \mathfrak{A}_i|\widetilde{\mathbb{H}}_{[n]}|\mathfrak{A}_j\rangle$. A possible measure for the accuracy of the approximation (34) is given by the norm of the residual vector

$$|r\rangle = (\widetilde{\mathbb{H}}_{[n]} - e_0)|e_0\rangle. \tag{36}$$

As long as $\|r\|$ is too big, we have to extend the set $\{|\mathfrak{A}_1\rangle, \ldots, |\mathfrak{A}_k\rangle\}$ iteratively by a further vector $|\mathfrak{A}_{k+1}\rangle$. Any form of educated guessing for a suitable new $|\mathfrak{A}_{k+1}\rangle$ is allowed. The Lanczos[29] and Arnoldi[30] algorithms use a different way of calculation but end up with

$$|\mathfrak{A}_{k+1}\rangle = \|r\|^{-1} \cdot |r\rangle, \tag{37}$$

with $|\mathfrak{A}_{k+1}\rangle \perp |\mathfrak{A}_{1 \leqslant j \leqslant k}\rangle$ by construction.

In contrast to the basic iMPS algorithm, which sets $|\mathfrak{A}_1\rangle$ equal to the lowest eigenvector $|e_{[n-1]0}\rangle = |A_{[n-1]}\rangle$ of the last round,[22] we choose

$$|\mathfrak{A}_1\rangle = \left|A^{[\mathrm{refer}]}_{[n]}\right\rangle, \tag{38}$$

with $A^{[\mathrm{refer}]}_{[n]}$ defined in Eq. (27). This small change allows an easy implementation of the method presented in Appendix B 1 and should help to improve global convergence. Both versions are straightforward examples of information recycling since $|A_{[n-1]}\rangle$ as well as $|A^{[\mathrm{refer}]}_{[n]}\rangle$ are already good approximations for $|E_0\rangle$ $(=|E_{[n]0}\rangle)$. In many cases, we could obtain a considerable speedup extending this idea to a few more than just the first vector of the set

$$\begin{aligned}|\mathfrak{A}_{1+j}\rangle &= \left|A^{[\mathrm{refer}]}_{[n-j]}\right\rangle_{\perp} \\ &= \frac{1}{\|\mathfrak{A}_{1+j}\|}\cdot\left(\mathbb{I}-\sum_{i=1}^{j-1}|\mathfrak{A}_i\rangle\langle\mathfrak{A}_i|\right)\cdot\left|A^{[\mathrm{refer}]}_{[n-j]}\right\rangle. \end{aligned}\tag{39}$$

Further, we observe that all $|A^{[\mathrm{refer}]}\rangle$ [Eq. (27)] are derived from the best eigenvectors of the previous rounds. As an additional extension, we also tried to include the next best eigenvectors $|e_{[n-1]j>0}\rangle$ of the last round

$$|\mathfrak{A}_{1+j}\rangle = |e_{[n-1]j}\rangle. \tag{40}$$

The improvements we achieved in this way were relatively poor. A much more promising way to take advantage of the $|e_{[n-1]j}\rangle$ is to use them for an efficient approximation of the inverse operator $\mathfrak{D} = (e_0\cdot\mathbb{I}-\widetilde{\mathbb{H}}_{[n]})^{-1}$ [Eq. (D6)], which allows a handy implementation resembling the Davidson (or Jacobi-Davidson)[24] method. As a result, the update equation (37) is replaced by the more appropriate ansatz (D7). Details are explained in Appendix D.

At the end of this section, we like to caution the reader that the methods presented here might counteract the methods presented in Sec. IV A. Global convergence and improved local minimization often go hand in hand, but not always. If the algorithms indicate to run unstable, one should consider to partially switch off the improvements just presented. This is likely to happen if $|e_0\rangle$ is degenerate. In this case, the recycled knowledge from the past strongly increases the probability that already a shallow optimization suffices to find alternative solutions, which might result in unwanted fluctuations as, e.g., described in Sec. III C2. Usually, this problem is announced in advance. In the Davidson implementation, one should not resort to eigenvectors with eigenvalues too close to the best. Similarly, once the small set of recycled initial values $|A^{[\mathrm{refer}]}_{[n-j]}\rangle$ suffices to get a second best eigenvalue very close to the best, it might be wise to abandon this method and only use $|A^{[\mathrm{refer}]}_{[n]}\rangle$ alone.

D. Enforcing translational invariant ground states

In this section, we demonstrate the algorithmic realization of the considerations put forward in Sec. III D. There, we argued that it is beneficial to push the algorithm towards translational invariant iMPS solutions to avoid getting trapped in local minima. We further showed that translational invariance is assured if the decomposition (8) of the tensor $A_{[\mathrm{converged}]}$ results in $\lambda_{[L]} = \lambda_{[R]}$ [Eq. (23)]. This is what we are aiming for.

The approach we are about to present is not very intuitive. Therefore, we start our explanations with an intermediate step and introduce a less practical but easier to understand procedure which consists of the following steps and has to be performed with each new tensor A after it has been optimized:

(1) Decompose A into $Q_{[L]}\cdot\lambda_{[L]} = A = \lambda_{[R]}\cdot Q_{[R]}$ [Eq. (8)].
(2) Define $\lambda_{[\mathrm{sym}]} = \frac{1}{2}(\lambda_{[L]}+\lambda_{[R]})$.
(3) Set $A \leftarrow \frac{1}{2}(Q_{[L]}\cdot\lambda_{[\mathrm{sym}]}+\lambda_{[\mathrm{sym}]}\cdot Q_{[R]})$.
(4) Go to 1.

Due to line 2, this procedure converges towards a tensor A with $\lambda_{[L]} = \lambda_{[R]}$. Further, we expect $A^{[\mathrm{initial}]} \approx A^{[\mathrm{final}]}$ if already $\lambda^{[\mathrm{initial}]}_{[L]} \approx \lambda^{[\mathrm{initial}]}_{[R]}$. Nonetheless, the changes in A might be too pronounced to be acceptable. To soften this approach, one can ignore line 4 and just go through 1 to 3 once. After that, we generally still have $\lambda_{[L]} \neq \lambda_{[R]}$ but with a reduced distance $\|\lambda_{[L]}-\lambda_{[R]}\|$ compared to the initial value. This is all we need to achieve $\lambda_{[L]} = \lambda_{[R]}$ in the long run. But, the new A is still likely not to qualify for the optimizing tensor we are looking for.

Now, we come to the procedure we really use. Instead of symmetrizing the tensor A *after* its optimization, we integrate the symmetrization into the optimization routine. As recapitulated in Sec. IV C, the optimization routine expresses the vectorized tensor $A_{[n]} = |A\rangle$ as a linear combination

$$|A\rangle = |\mathfrak{A}_i\rangle\cdot a_i \tag{41}$$

of a small set of basis vectors $|\mathfrak{A}_i\rangle$ [Eq. (34)]. The idea is to alter these basis vectors $|\mathfrak{A}_i\rangle$ such that we have a similar effect as the procedure above. At the stage of the optimization, the $Q^{[n]}_{[L/R]}$ are still unknown and we have to approximate them by their precursors $Q^{[n-1]}_{[L/R]}$.

The $|\mathfrak{A}_i\rangle$ are created iteratively. In each iteration step, we first create a new $|\mathfrak{A}_i\rangle$ as we used to do Eqs. (37) and (D8) and then alter it. Therefor, we introduce $|\bar{\mathfrak{A}}_i\rangle$ defined as

$$\begin{aligned}|\bar{\mathfrak{A}}_i\rangle &= \frac{1}{2}|\mathfrak{A}_i\rangle + \frac{1}{4}\left(Q^{[n-1]}_{[L]}\cdot\bar{\lambda}_{[R]i}+\bar{\lambda}_{[L]i}\cdot Q^{[n-1]}_{[R]}\right) \quad \text{with} \\ \bar{\lambda}^{\alpha\beta}_{[R]i} &= |\mathfrak{A}_i\rangle^{\alpha\gamma}_{s}\cdot Q^{[n-1]*\,\gamma\beta}_{[R]s}, \quad \bar{\lambda}^{[n]\,\alpha\beta}_{[L]i} = Q^{[n-1]*\,\alpha\gamma}_{[L]s}\cdot|\mathfrak{A}_i\rangle^{\gamma\beta}_{s}, \end{aligned}\tag{42}$$

where we tensorized the vector $|\mathfrak{A}_i\rangle$ in lines 2 and 3. With that, we replace $|\mathfrak{A}_i\rangle$ by an orthonormal version of $|\bar{\mathfrak{A}}_i\rangle$:

$$|\mathfrak{A}_i\rangle \leftarrow \left(\mathbb{I}-\sum_{j=1}^{i-1}|\mathfrak{A}_j\rangle\langle\mathfrak{A}_j|\right)\cdot|\bar{\mathfrak{A}}_i\rangle, \quad |\mathfrak{A}_i\rangle \leftarrow \frac{1}{\|\mathfrak{A}_i\|}\cdot|\mathfrak{A}_i\rangle. \tag{43}$$

For a better understanding, we insert the $|\bar{\mathfrak{A}}_i\rangle$ in the linear combination (41). As shown in Appendix E, we get

$$\begin{aligned}|\bar{A}\rangle &= |\bar{\mathfrak{A}}_i\rangle\cdot a_i \approx \tfrac{1}{2}(Q_{[L]}\cdot\lambda_{[\mathrm{sym}]}+\lambda_{[\mathrm{sym}]}\cdot Q_{[R]}) \quad \text{with} \\ \lambda_{[\mathrm{sym}]} &= \tfrac{1}{2}(\lambda_{[L]}+\lambda_{[R]}), \end{aligned}\tag{44}$$

which mimics the effect of the procedure presented above. But, in contrast to the procedure above, the story does not end here. The important point to notice is that the algorithm can still adopt to the alteration (43) of the basis vectors $|\mathfrak{A}_i\rangle$ and come up with alternative solutions. More favorable weights a_i than those used in Eq. (44) are presumably to be found. Even the $|\mathfrak{A}_i\rangle$ themselves are likely to be different since they are calculated iteratively according to the needs of the minimization. While there are still enough resources to compensate sufficiently for the negative effects of the enforced alteration, the positive effects should survive since the arguments in their favor are largely independent of the a_i, $|\mathfrak{A}_i\rangle$ chosen by the optimization routine. Still, this alteration is a tradeoff, but we have good reasons to believe that we gain more than we sacrifice.

For practical applications, we only need a few lines of code to implement Eq. (42), which is also easy to turn off for systems where it is not needed, i.e., when the unaltered algorithm shows no tendency to run the risk of being trapped in a local minimum. In such a case, the alteration is likely to slow down the algorithm slightly. For the applications tested by us, the loss in performance was only marginal. On the other hand, we also encountered many cases where the altered algorithm clearly outperformed the unaltered one, which was partially even unable to find the correct ground state within the observed run time.

Although we strongly recommend to implement the alteration (43) as presented, one could also use a compromise and only alter the first basis vector $|\mathfrak{A}_1\rangle = A^{[\mathrm{refer}]}_{[n]}$, which has already a strong impact on the outcome of the optimization. This reduced version does not come with the need to program a new eigenvector solver. Each solver which accepts an initial vector $|\mathfrak{A}_1\rangle$ will do. In any case, the gain function in Eq. (29) is understood to change accordingly to the alteration of $|\mathfrak{A}_1\rangle = A^{[\mathrm{refer}]}_{[n]}$.

E. Length of the MPS

After n optimization steps, even the longest MPS encoded in the superposition created by the SMO method does not surpass the length $l = 2n + l_0$ [where l_0 is the initial length (see Sec. II E 2)]. For some systems with long-range correlations, this might be too short unless n reaches some considerably high number, which would go along with an extended calculation time. To shorten this calculation time, two methods might be of help:

(1) Use a tensor $A^{\alpha_l\alpha_r}_{[n]\,s}$ with small bond dimension $\chi_{[\mathrm{small}]}$ until a certain MPS length is reached, then increase the bond dimension to its final value $\chi_{[\mathrm{big}]}$.

(2) Use fast Krylov subspace methods[28] to insert the same tensor many times (e.g., 10^5) into the MPS.

A simple and comfortable way to increase the bond dimension from $\chi_{[\mathrm{small}]}$ to $\chi_{[\mathrm{big}]}$ is to use an isometric $\chi_{[\mathrm{small}]} \times \chi_{[\mathrm{big}]}$ matrix $u^{\alpha\beta}$ with

$$u^{\alpha\beta} \cdot (u^T)^{\beta\alpha'} = \delta^{\alpha\alpha'} \tag{45}$$

and proceed as follows after $A_{[n]}$ has been optimized but still not been inserted into $L_{[n]}$ [Eq. (16)] and $R_{[n]}$ [Eq. (17)]:

$$\begin{aligned} L^{\beta'_l\mu_l\beta_l}_{[n]} &\leftarrow L^{\alpha'_l\mu_l\alpha_l}_{[n]} \cdot u^{\alpha'_l\beta'_l} \cdot u^{\alpha_l\beta_l}, \\ R^{\beta'_r\mu_r\beta_r}_{[n]} &\leftarrow R^{\alpha'_r\mu_r\alpha_r}_{[n]} \cdot u^{\alpha'_r\beta'_r} \cdot u^{\alpha_r\beta_r}, \\ A^{\beta_l\beta_r}_{[n]\,s} &\leftarrow A^{\alpha_l\alpha_r}_{[n]\,s} \cdot u^{\alpha_l\beta_l} \cdot u^{\alpha_r\beta_r}. \end{aligned} \tag{46}$$

Next, the new tensor $A^{\beta_l\beta_r}_{[n]\,s}$ is inserted into $L^{\beta'_l\mu_l\beta_l}_{[n]}$ and $R^{\beta'_r\mu_r\beta_r}_{[n]}$ as usual but without the superposition building steps [Eqs. (19) and (25), respectively]. To avoid trapping into a local minimum, one might also consider to add a small amount of noise to $A^{\alpha_l\alpha_r}_{[n]\,s}$ before applying Eq. (46).

A possible strategy for the small bond dimension $\chi_{[\mathrm{small}]}$ is to proceed until convergence has been reached $A_{[n]} \to A_{[\mathrm{converged}]}$, but before the bond dimension is increased, many more copies of $A_{[\mathrm{converged}]}$ are inserted into the MPS without any further optimization. These insertions can be done in the standard fashion or generally much faster by projecting the problem onto a small subspace, similar to the way the eigenvector problem is solved (see Sec. IV C). To formalize this method, let us introduce the operator $\mathcal{I}$ which inserts one copy of $A_{[\mathrm{converged}]}$ into $L_{[n]}$ [Eq. (16)], i.e.,

$$\mathcal{I} \cdot L_{[n]} = L_{[n+1]}. \tag{47}$$

With that, we build up the Krylov subspace $\mathcal{K}_r$,

$$\mathcal{K}_r = \mathrm{span}\{L_{[n]}, \mathcal{I} \cdot L_{[n]}, \mathcal{I}^2 \cdot L_{[n]}, \ldots, \mathcal{I}^{r-1} \cdot L_{[n]}\}, \tag{48}$$

and similar with $R_{[n]}$ [Eq. (17)]. As in Sec. IV C, we create an orthonormalized system of basis vectors $|\mathfrak{L}_k\rangle$,

$$|\mathfrak{L}_k\rangle = \frac{1}{\|\mathfrak{L}_k\|} \cdot \left(\mathbb{I} - \sum_{i=0}^{k-1} |\mathfrak{L}_i\rangle\langle\mathfrak{L}_i| \right) \cdot |\mathcal{I}^k \cdot L_{[n]}\rangle, \tag{49}$$

and calculate $\mathfrak{I}_{ij}$, the subspace projection of $\mathcal{I}$:

$$\mathfrak{I}_{ij} = \langle\mathfrak{L}_i|\mathcal{I}|\mathfrak{L}_j\rangle. \tag{50}$$

Keeping in mind that the subspace projection of $L_{[n]}$ is simply given by the vector $l_j = (1\,0\,0\,0\,\ldots)^T$, we find

$$L_{[n+p]} = \mathcal{I}^p \cdot L_{[n]} \approx \mathbf{(}(\mathfrak{I}^p)_{ij} \cdot l_j\mathbf{)} \cdot |\mathfrak{L}_i\rangle. \tag{51}$$

The number of basis vectors $|\mathfrak{L}_i\rangle$ should be chosen such that this approximation is perfect within computer precision. Further errors are introduced by an imperfectly converged $A^{\alpha_l\alpha_r}_{[\mathrm{converged}]\,s}$ and from the energy overgrow effect described in Sec. IV B, which should rule out attempts to go for $p \to \infty$. Still, a small amount of the last two errors are acceptable since they have a similar effect as the aforementioned extra noise to avoid local minima.

V. APPLICATIONS

In the last section, we presented various methods to improve the performance of the iMPS algorithm with long-range interactions. The main subject was to ensure convergence, where special attention was paid to broken translational invariance. This so far troublesome case can now be tackled mainly due to the introduced method of superposed multioptimization (SMO).

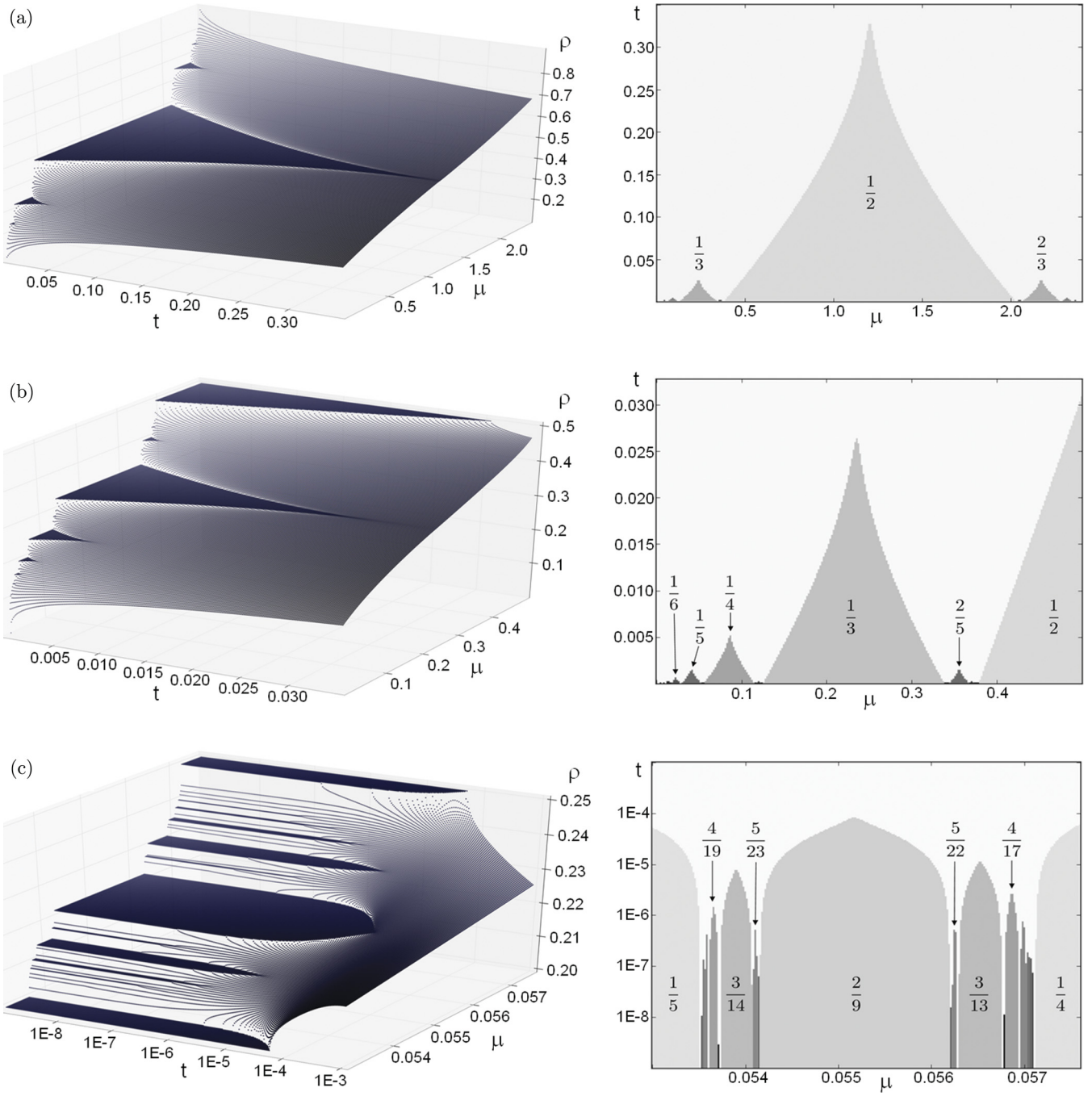

FIG. 3. (Color online) The densities ρ of polar bosons (left) and the corresponding phase diagrams (right) for the ground states of the Hamiltonian (53) for $U \to \infty$ plotted over t and μ in units of $V = 1$. Each plot consists of 66 049 data points calculated with the bond dimensions $\chi_{\mathrm{MPS}} = 32$ for (a) and (b) and $\chi_{\mathrm{MPS}} = 64$ for (c). The fractions associated to selected phases of the phase diagrams denote their p/q values (see main text).

The modified iMPS algorithm has superior convergence properties compared to the basic version but it does not surpass its precision, which is determined by MPS and MPO inherited limitations. In cases where both versions converge, the quality of the results is identical. Readers who are interested in the achievable precision of the iMPS method in comparison with analytical solutions are therefore referred to the literature.[22,23]

We checked our algorithm with different models. For a reliable basic benchmark, we examined, e.g., states with long-range chiral order in the next-nearest-neighbor Heisenberg model and found the expected agreement with the results given in Refs. 31 and 32.

Here, we will present results for a model of polar bosons described by a Bose-Hubbard–type Hamiltonian with a long-range interaction term. In the thermodynamic limit, the ground state of this model exhibits symmetry-breaking crystalline phases as well as incommensurate phases with algebraically decaying long-range correlations. The long-range interaction of the Hamiltonian we consider decays as $\sim r^{-3}$. To model this interaction with an MPO, the decay is approximated as

weighted sum of 20 exponential functions

$$(r)^{-3} \approx \sum_{i=1}^{20} a_i \cdot \lambda_i^{r-1}, \quad r = 1,2,3,\ldots \tag{52}$$

[see also Eq. (A17) and Ref. 19].

A. Bose-Hubbard model with long-range interaction

We study the thermodynamic limit ground states of polar bosons in a one-dimensional optical lattice described by the following effective Hamiltonian[11,33]:

$$\mathcal{H} = V \cdot \sum_{k<j} \frac{1}{(j-k)^3} \cdot \hat{n}_k \cdot \hat{n}_j + \frac{U}{2} \cdot \sum_j \hat{n}_j \cdot (\hat{n}_j - 1) - \mu \cdot \sum_j \hat{n}_j - t \cdot \sum_j (\hat{c}_j^\dagger \cdot \hat{c}_{j+1} + \hat{c}_j \cdot \hat{c}_{j+1}^\dagger), \tag{53}$$

where $\hat{c}_j^\dagger$ and $\hat{c}_j$ are the creation and annihilation operators for a boson on site j and $\hat{n}_j = \hat{c}_j^\dagger \cdot \hat{c}_j$. This model is characterized by a hopping amplitude t, an onsite interaction energy U, a chemical potential μ, and a long-range dipole-dipole coupling V/r^3. For $\mu > 0$, the chemical potential favors as many bosons as possible in the ground state, while the dipole-dipole coupling together with the onsite interaction try to avoid two bosons coming too close to each other. For certain parameter regimes, this interplay allows for translational invariance-breaking crystalline phases with optimized distances between the bosons where q sites accommodate exactly p bosons. We will refer to them as p/q phases. The model is known to host an entire Devil's staircase of crystalline phases for $t = 0$ if the joined potential of onsite interaction and dipole-dipole coupling is convex.[11,34]

In the following, we will investigate two qualitative different regimes of this model: $U \to \infty$ and $U = V$.

1. Devils's staircase for $U \to \infty$

For $U \to \infty$, each site can accommodate at most one boson and the effective dimension d_{eff} of the local Hilbert spaces reduces to $d_{\text{eff}} = 2$. Figure 3 displays the average ground state densities of the bosons and the localization of the corresponding p/q phases. To determine the periodicity q of the phases, we counted the number of eigenvectors of the transfer matrix $T_{[L]}^{ij}$:

$$T_{[L]}^{ij} = T_{[L]}^{(\alpha_l \alpha_l'),(\alpha_r \alpha_r')} = Q_{[L]s}^{\alpha_l \alpha_r} \cdot Q_{[L]s}^{*\,\alpha_l' \alpha_r'} \tag{54}$$

with an absolute eigenvalue of one. Once q is known, p follows from the average density. Figure 3(c) shows a magnification of the area between the $\frac{1}{4}$ and the $\frac{1}{5}$ phases. The biggest phase between these two phases is the $\frac{2}{9}$ phase, which can be understood as primary compromise ($\frac{2}{9} = [1+1]/[4+5]$). In the same fashion, we find, e.g., that the biggest phase between the phases $\frac{2}{9}$ and $\frac{1}{5}$ is the $\frac{3}{14}$ phase. The maximal detectable value of q is given by the bond dimension χ of the MPS, which is 64 in case of Fig. 3(c). However, since the range of μ covered by the different phases diminishes with growing value of q, most phases beyond $q = 30$ escaped our resolution. The highest value we hit was $p/q = \frac{11}{52}$.

2. Devils's staircase for $U = V = 1$

For sufficient small U, the ground states of the Hamiltonian (53) might accommodate more than one boson per site, which allows for new types of Devil's staircases. An example is given by Fig. 4, which shows the densities and phases for $U = V = 1$. Here, simple translational invariance is broken by an underlying occupation pattern given by $\ldots, x_i, 0, x_{i+2}, 0, x_{i+4}, 0, \ldots$ with $x_j = 1$ or 2. Of course, for any nonzero hopping amplitude $t > 0$, we expect fluctuations around this pattern such that a more accurate description might be given by $\ldots, x_i, \varepsilon_{i+1}, x_{i+2}, \varepsilon_{i+3}, \ldots$, which we need in the next section (Sec. V A3). At a certain point, these fluctuations will become so strong that the underlying pattern is destroyed, but this is not the case for the entire region of Fig. 4.

In the lobes of the new Devil's staircase, the sublattice $\ldots, x_i, x_{i+2}, x_{i+4}, \ldots$ crystallizes in regular pattern of single and double occupied sites. These lobes exhibit an approximate symmetry under the exchange of single and double occupied sites. This is a nontrivial symmetry in contrast to the exact particle hole symmetry of Fig. 3(a).

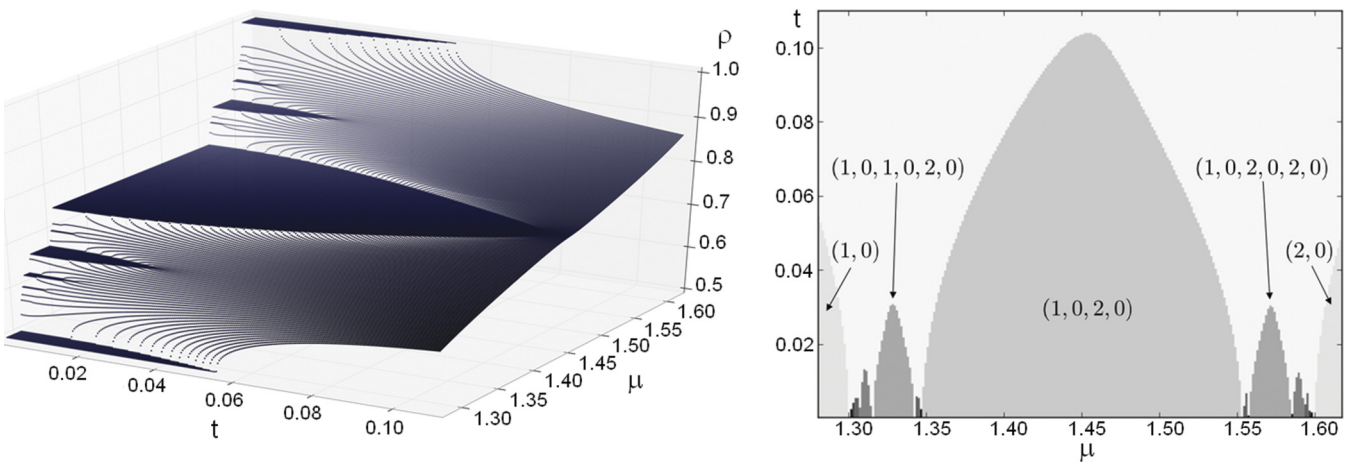

FIG. 4. (Color online) The densities ρ of polar bosons (left) and the corresponding phase diagram (right) for the ground states of the Hamiltonian (53) with $U = V = 1$ plotted over t and μ. Each plot consists of 62 194 data points calculated with the bond dimensions $\chi_{\text{MPS}} = 32$. The bracketed numbers associated to selected phases of the phase diagram describe their periodically repeated occupation pattern (see main text).

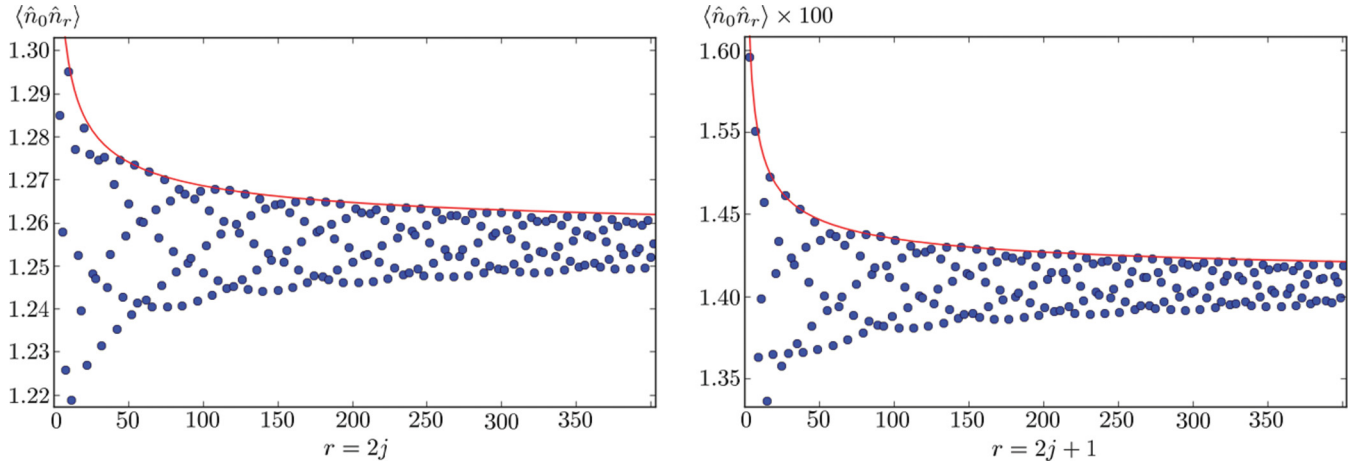

FIG. 5. (Color online) The blue circles mark the values of the correlation function $\langle \hat{n}_0 \hat{n}_r \rangle$ for $U = V = 1$, $\mu = 1.55$, and $t = 0.044$ extracted from an MPS with bond dimension $\chi = 256$. For odd distances $r = 2j + 1$ (right picture), $\langle \hat{n}_0 \hat{n}_r \rangle$ is suppressed by a factor of roughly 100 compared to even distances $r = 2j$ (left picture). To demonstrate the algebraic decay of $\langle \hat{n}_0 \hat{n}_r \rangle$, the function $f(r) = \alpha \cdot r^{-0.5} + \langle \hat{n}_0 \hat{n}_{r\to\infty} \rangle$ with an adequate α is included on top of both plots as a red line.

Outside the crystalline phases, Burnell[11] predicted a supersolidlike phase. In the following, we present numerical evidence which supports this claim.

3. *Supersolids for $U = V = 1$*

A supersolid is characterized as a spatially ordered phase which also exhibits superfluid properties. We already mentioned the spatial order belonging to Fig. 4, which is given by the occupation pattern $\ldots, x_i, \varepsilon_{i+1}, x_{i+2}, \varepsilon_{i+3}, \ldots$. In our numerical studies, we consider translational invariant superpositions of the ground states, where the occupation pattern is still visible in the two-point correlation functions as $\langle \hat{n}_0 \hat{n}_r \rangle$ and $\langle \hat{c}_0^\dagger \hat{c}_r \rangle$ shown in Figs. 5 and 6. Due to the spatial order, both correlation functions are split into two branches, where correlations belonging to odd distances $r = 2j + 1$ are strongly suppressed compared to correlations belonging to even distances $r = 2j$. Furthermore, both branches exhibit an algebraic decay of the same power. For a superfluid, the power of the decay of $\langle \hat{n}_0 \hat{n}_r \rangle$ is supposed to be reciprocal to the power of $\langle \hat{c}_0^\dagger \hat{c}_r \rangle$. Luttinger liquid theory predicts[35]

$$\langle \hat{n}_0 \hat{n}_r \rangle = \langle \hat{n}_0 \hat{n}_{r\to\infty} \rangle + \text{const}_1 \times \cos(2\pi\rho_0 r)\, r^{-2K} + \cdots,$$
$$\langle \hat{c}_0^\dagger \hat{c}_r \rangle = \text{const}_2 \times r^{-\frac{1}{2K}} + \cdots \tag{55}$$

with $\rho_0 = \langle \hat{n}_i \rangle$. These relations are confirmed by our numerical results, as displayed in Fig. 7. Both correlation functions give rise to the same Luttinger liquid parameter K within a small error range.

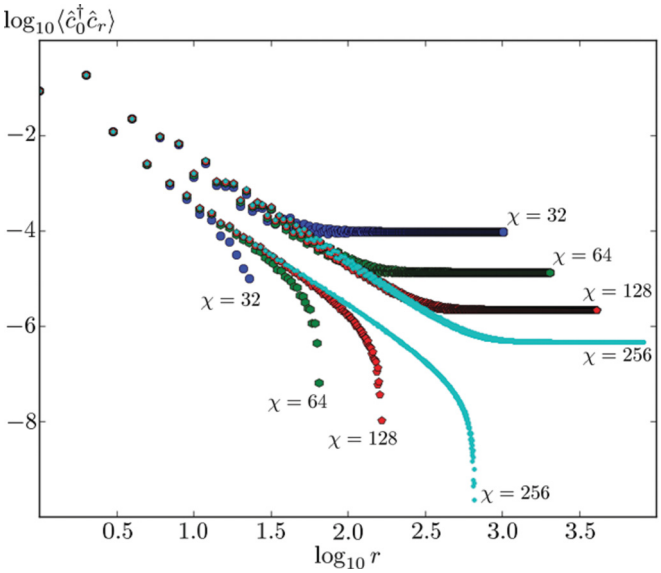

FIG. 6. (Color online) Log-log plot of the correlation function $\langle \hat{c}_0^\dagger \hat{c}_r \rangle$ for $U = V = 1$, $\mu = 1.55$, and $t = 0.044$ extracted from MPS with different bond dimensions χ. Due to the spatial order, $\langle \hat{c}_0^\dagger \hat{c}_r \rangle$ splits into two branches, each decaying $\propto r^{-\alpha} + \text{const}_\chi$, with $\text{const}_\chi < 0$ for the lower branch. This constant is a purely numerical effect, as can be seen by the scaling with the bond dimension χ.

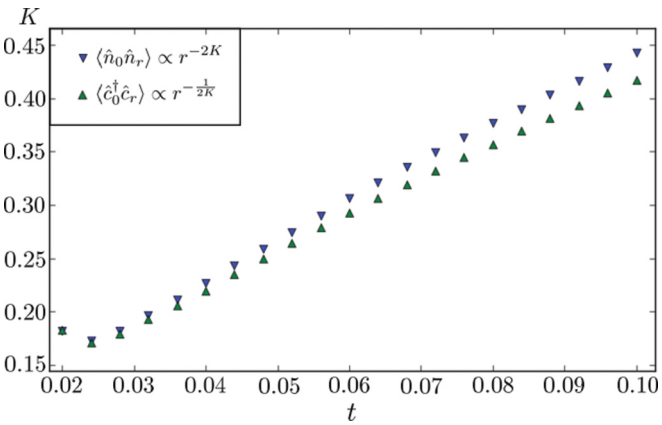

FIG. 7. (Color online) The Luttinger liquid parameter K in dependence of the hopping amplitude t for $U = V = 1$ and $\mu = 1.55$. The parameter K was obtained in two different ways by fitting the first-order term approximation (55) to the numerical values of $\langle \hat{n}_0 \hat{n}_r \rangle$ as well as $\langle \hat{c}_0^\dagger \hat{c}_r \rangle$, both calculated with an MPS bond dimension $\chi = 256$. The fits were performed in the interval $r = 10 \ldots 60$ with only two free parameters, namely, $\text{const}_{1/2}$ and K of Eq. (55).

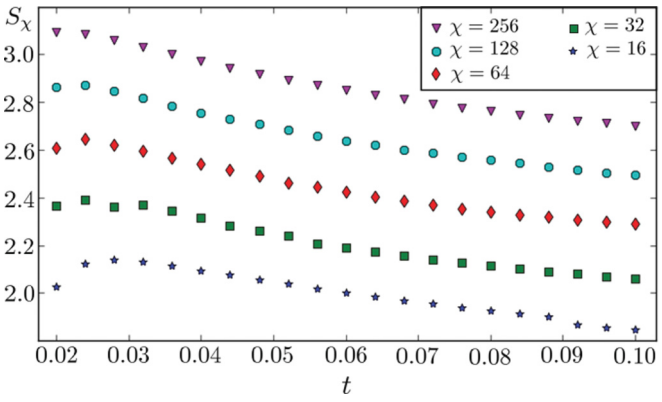

FIG. 8. (Color online) The entanglement entropy of the half chain S_χ extracted from MPS with different bond dimension χ is plotted in dependence of the hopping amplitude t for $U = V = 1$ and $\mu = 1.55$. The different curves are roughly equidistant, as predicted by Eq. (56).

In the thermodynamical limit, algebraically decaying correlation functions go hand in hand with an infinite entanglement entropy of the half chain S, which can not be represented by any MPS with finite bond dimension χ. Nonetheless, it was shown[36,37] that the numerically obtained entropy S_χ for such critical phases shows a predictable scaling as function of the MPS bond dimension χ (Ref. 37):

$$S_{\chi_2} - S_{\chi_1} \approx \left(\sqrt{\frac{12}{c}} + 1\right)^{-1} \ln_2\left(\frac{\chi_2}{\chi_1}\right), \tag{56}$$

where c represents the central charge. A demonstration of the scaling behavior is given in Fig. 8. From this numerical sample, one obtains $S_{\chi=256} - S_{\chi=16} = (0.218 \pm 0.003) \times 4$, which matches $S_{\chi=256} - S_{\chi=16} \approx 0.224 \times 4$ drawn from Eq. (56) for $c = 1$.

VI. CONCLUSION

We have presented several extensions to the basic iMPS algorithm for systems with long-range interactions, some of them with the potential to be useful in a much broader context. A special focus was set on problems arising from broken translational invariance. Here, convergence was ensured by various means, but mainly due to the SMO method which irons out local variations by optimizing exponentially many MPS simultaneously. The algorithm was successfully applied to calculate detailed Devil's staircases and phase diagrams for polar bosons [Eq. (53)] and was also suitable to verify supersolid properties. Theoretical restraints as considered in the comments in Sec. III B2 seem to have negligible influence on practical applications such that this altered version of the iMPS algorithm is a genuine improvement in the sense that it can do all the old version could plus more.

ACKNOWLEDGMENTS

This research was funded by the Austrian Science Fund (FWF): P20748-N16, P24273-N16, SFB F40-FoQus F4012-N16, the European Union (NAMEQUAM), and by the Austrian Ministry of Science BMWF as part of the UniInfrastrukturprogramm of the Research Platform Scientific Computing at the University of Innsbruck. We would further like to thank A. J. Daley, M. Dalmonte, J. Schachenmayer, U, Schollwöck, and especially A. Läuchli for helpful discussions.

APPENDIX A: MPO REPRESENTATION FOR HAMILTONIANS

For self-consistency, we give a short account based on some examples as to how to construct an MPO representation for a given Hamiltonian (see also Refs. 18–21). For finite systems with open boundaries, the Hamiltonian can be written [Eq. (4)] as

$$\mathcal{H}^{s_1' s_2' \ldots s_n'}_{s_1 s_2 \ldots s_n} = H^{[1]\,\mu_1}_{s_1' s_1} \cdot H^{[2]\,\mu_1\mu_2}_{s_2' s_2} \cdot H^{[3]\,\mu_2\mu_3}_{s_3' s_3} \ldots H^{[n-1]\,\mu_{n-2}\mu_{n-1}}_{s_{n-1}' s_{n-1}} \cdot H^{[n]\,\mu_{n-1}}_{s_n' s_n}. \tag{A1}$$

First, we need a neat way to write the explicit form of the fourth-order tensors $H^{\mu_l\mu_r}_{s's}$. We write them as matrices whose entries are matrices too:

$$H^{\mu_l\mu_r}_{s's} = (H^{s's})^{\mu_l\mu_r}. \tag{A2}$$

As an example, we consider the Ising Hamiltonian

$$\mathcal{H} = -\sum_{i=1}^{n-1} \sigma_z^{[i]} \cdot \sigma_z^{[i+1]} - \sum_{i=1}^{n} \sigma_x^{[i]}. \tag{A3}$$

As we will see in the following, a possible choice for all $H^{[k]}$ in Eq. (A1) with $2 \leqslant k \leqslant n-1$ is

$$(H^{s's})^{\mu_l\mu_r} = \begin{pmatrix} \mathbb{I}^{s's} & 0^{s's} & 0^{s's} \\ -\sigma_z^{s's} & 0^{s's} & 0^{s's} \\ -\sigma_x^{s's} & \sigma_z^{s's} & \mathbb{I}^{s's} \end{pmatrix}. \tag{A4}$$

$H^{[1]\,\mu_1}_{s_1' s_1}$ and $H^{[n]\,\mu_{n-1}}_{s_n' s_n}$ are vectors over matrices

$$H^{[1]\,\mu_1}_{s_1' s_1} = \left(-\sigma_x^{s_1' s_1}, \quad \sigma_z^{s_1' s_1}, \quad \mathbb{I}^{s_1' s_1}\right). \tag{A5}$$

In order to get a better understanding, we look at the tensor product of the first k tensors. Below (A7), we show by induction that

$$\begin{aligned} H^{[1\ldots k]} &= H^{[1]\,\mu_1}_{s_1' s_1} \cdot H^{[2]\,\mu_1\mu_2}_{s_2' s_2} \ldots H^{[k]\,\mu_{k-1}\mu_k}_{s_k' s_k} \\ &= \left(\left[-\sum_{i=1}^{k-1} \sigma_z^{[i]} \cdot \sigma_z^{[i+1]} - \sum_{i=1}^{k} \sigma_x^{[i]}\right], \sigma_z^{[k]}, \mathbb{I}\right) \\ &= \left(\mathcal{H}^{[k]}, \sigma_z^{[k]}, \mathbb{I}\right). \end{aligned} \tag{A6}$$

The resulting vector $H^{[1\ldots k]}$ can be seen as an object with three "slots" in which all the relevant information about the first k sites is stored. Of course, the number of slots corresponds to the bond dimension of the MPO. The first slot contains all interaction terms between the first k sites only and local terms. Since the kth site also interacts with the $k+1$th site, the second slot of the vector passes on $\sigma_z^{[k]}$ and finally, the third slot preserves the identity $\mathbb{I} = \mathbb{I}^{s_1' s_1} \otimes \mathbb{I}^{s_2' s_2} \otimes \ldots \otimes \mathbb{I}^{s_k' s_k}$. For $H^{[1]}$ this description is easily checked. The tensor $H^{[k]} = (H^{s's})^{\mu_l\mu_r}$ [Eq. (A4)] is designed such that it performs the

correct induction step

$$\begin{aligned} H^{[1\ldots k]} &= H^{[1\ldots k-1]} \cdot H^{[k]} \\ &= \left(\mathcal{H}^{[k-1]}, \quad \sigma_z^{[k-1]}, \quad \mathbb{I}\right) \begin{pmatrix} \mathbb{I}^{s_k' s_k} & 0^{s_k' s_k} & 0^{s_k' s_k} \\ -\sigma_z^{s_k' s_k} & 0^{s_k' s_k} & 0^{s_k' s_k} \\ -\sigma_x^{s_k' s_k} & \sigma_z^{s_k' s_k} & \mathbb{I}^{s_k' s_k} \end{pmatrix} \\ &= \left(\left[\mathcal{H}^{[k-1]} - \sigma_z^{[k-1]} \cdot \sigma_z^{[k]} - \sigma_x^{[k]}\right], \sigma_z^{[k]}, \mathbb{I}\right) \\ &= \left(\mathcal{H}^{[k]}, \quad \sigma_z^{[k]}, \quad \mathbb{I}\right). \end{aligned} \tag{A7}$$

The final tensor $H^{[n]\mu_{n-1}}_{s_n' s_n}$ is given by

$$H^{[n]\mu_{n-1}}_{s_n' s_n} = \left(\mathbb{I}^{s_n' s_n}, \quad -\sigma_z^{s_n' s_n}, - \quad \sigma_x^{s_n' s_n}\right)^T. \tag{A8}$$

With that, we get

$$\begin{aligned} H^{[1\ldots n]} &= H^{[1\ldots n-1]} \cdot H^{[n]} \\ &= \left(\mathcal{H}^{[n-1]}, \sigma_z^{[n-1]}, \mathbb{I}\right) \begin{pmatrix} \mathbb{I}^{s_n' s_n} \\ -\sigma_z^{s_n' s_n} \\ -\sigma_x^{s_n' s_n} \end{pmatrix} \\ &= \mathcal{H}^{[n-1]} - \sigma_z^{[n-1]} \cdot \sigma_z^{[n]} - \sigma_x^{[n]} \\ &= -\sum_{i=1}^{n-1} \sigma_z^{[i]} \cdot \sigma_z^{[i+1]} - \sum_{i=1}^{n} \sigma_x^{[i]}. \end{aligned} \tag{A9}$$

We described the vector $H^{[1\ldots k]}$ [Eq. (A6)] as an object which contains all relevant information of the sites $1\ldots k$. This description is true not only for Ising interaction. For any Hamiltonian, we have to identify what the relevant information is and design the vector accordingly. As convention, we use the first slot of the vector to store $\mathcal{H}^{[k]}$ the sum of all interaction terms between the first k sites only, including local terms. In the last slot, we pass on the identity. The slots in-between are needed for interaction terms which involve (at least) one of the first k sites and (at least) one of the other sites $k+1\ldots n$. In the case of a Heisenberg chain

$$\begin{aligned} \mathcal{H} = &\sum_{i=1}^{n-1} J_x \cdot \sigma_x^{[i]} \cdot \sigma_x^{[i+1]} + \sum_{i=1}^{n-1} J_y \cdot \sigma_y^{[i]} \cdot \sigma_y^{[i+1]} \\ &+ \sum_{i=1}^{n-1} J_z \cdot \sigma_z^{[i]} \cdot \sigma_z^{[i+1]}, \end{aligned} \tag{A10}$$

the vector $H^{[1\ldots k]}$ needs five slots:

$$H^{[1\ldots k]} = \left(\mathcal{H}^{[k]}, \sigma_x^{[k]}, \sigma_y^{[k]}, \sigma_z^{[k]}, \mathbb{I}\right). \tag{A11}$$

Further slots might be necessary if we do not restrict ourselves to nearest-neighbor interactions.

Once we have identified the structure of $H^{[1\ldots k]}$, it is straightforward to write the first tensor $H^{[1]}$ in vector form. The matrix structure of all the following tensors $H^{[j]}$ is constructed columnwise such that the induction

$$H^{[1\ldots j]} = H^{[1\ldots j-1]} \cdot H^{[j]} \tag{A12}$$

is accomplished as in Eqs. (A7) or (A9) for the final tensor $H^{[n]}$. According to our convention, local terms, as needed in Sec. IV B, are always represented in the bottom left entry of the matrices.

For long-range interaction, the recipe given so far becomes problematic. The longer the range of the interaction, the more information has to be stored in the vector $H^{[1\ldots k]}$, which usually requires more and more slots. But, there are some exceptions (see, e.g., Ref. 21). An exponentially decaying interaction needs only one slot, even for infinite range. As an example, we look at the toy Hamiltonian

$$\mathcal{H} = J \cdot \sum_{i=1}^{n} \sum_{j=1}^{i-1} \lambda^{i-j-1} \cdot \sigma_z^{[j]} \cdot \sigma_z^{[i]}, \tag{A13}$$

where $i-j-1$ is the exponent of λ and not an index. First, we have to identify the structure of $H^{[1\ldots k]}$:

$$H^{[1\ldots k]} = \left(\mathcal{H}^{[k]}, \textstyle\sum_{j=1}^{k} \lambda^{k-j} \cdot \sigma_z^{[j]}, \mathbb{I}\right).$$

The crucial observation is that $\sum_{j=1}^{k} \lambda^{k-j} \cdot \sigma_z^{[j]}$ can be generated iteratively. The following tensors fulfill this task:

$$H^{[1]} = \left(0^{s_1' s_1}, \sigma_z^{s_1' s_1}, \mathbb{I}^{s_1' s_1}\right), \tag{A14}$$

$$H^{[k]} = \begin{pmatrix} \mathbb{I}^{s_k' s_k} & 0^{s_k' s_k} & 0^{s_k' s_k} \\ J \cdot \sigma_z^{s_k' s_k} & \lambda \cdot \mathbb{I}^{s_k' s_k} & 0^{s_k' s_k} \\ 0^{s_k' s_k} & \sigma_z^{s_k' s_k} & \mathbb{I}^{s_k' s_k} \end{pmatrix}, \tag{A15}$$

$$H^{[n]} = \left(\mathbb{I}^{s_n' s_n}, J \cdot \sigma_z^{s_n' s_n}, 0^{s_n' s_n}\right)^T. \tag{A16}$$

In order to encode the polynomial decay as $(r)^{-3}$ into an MPO, we resorted to an approximation as a weighted sum of N_{exp} different exponential terms, i.e.,

$$(r)^{-3} \approx \sum_{i=1}^{N_{\text{exp}}} a_i \lambda_i^{r-1}, \quad r = 1,2,3,\ldots. \tag{A17}$$

In the Appendix of Ref. 19 it is shown how to calculate the optimal a_i and λ_i. For the quality of the approximation in dependence of N_{exp}, see Ref. 22.

APPENDIX B: GAIN FUNCTION

We like to estimate the influence of γ in

$$\widetilde{\mathbb{H}}^{[\gamma]} = \widetilde{\mathbb{H}} - \gamma \cdot |A^{[\text{refer}]}\rangle\langle A^{[\text{refer}]}| \quad \text{with} \quad \gamma \geqslant 0 \tag{B1}$$

on $A^{[\gamma]}$ which is supposed to minimize $\langle A^{[\gamma]}|\widetilde{\mathbb{H}}^{[\gamma]}|A^{[\gamma]}\rangle$. For our numerical purposes, the following simple approximation suffices:

$$A^{[\gamma]} = \sqrt{1-\varepsilon^2(\gamma)} \cdot A^{[\text{refer}]} + \varepsilon(\gamma) \cdot B, \tag{B2}$$

with $\|A^{[\text{refer}]}\| = \|B\| = 1$ and $A^{[\text{refer}]} \perp B = \text{const}$. The vector B is extracted from $A^{[\gamma=0]}$, which has to be calculated first. This might seem inefficient since we have to minimize $\langle A|\widetilde{\mathbb{H}}^{[\gamma]}|A\rangle$ twice, once for $\gamma = 0$ and once for the final value of γ. The solution is to use a minimization routine which projects the minimization onto a small subspace, as explained in Sec. IV C. This projection has to be done only once but can be used twice. Since the projection is the most time-consuming part, the double calculation is done quite cheap. More to this at the end of this section.

Now, we calculate the pseudoenergy E using Eqs. (B1) and (B2):

$$\begin{aligned} E &= \langle A^{[\gamma]}|\widetilde{\mathbb{H}}^{[\gamma]}|A^{[\gamma]}\rangle \\ &= (1-\varepsilon^2)\cdot\left(\langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|A^{[\text{refer}]}\rangle - \gamma\right) + \varepsilon^2\cdot\langle B|\widetilde{\mathbb{H}}|B\rangle \\ &\quad + 2\varepsilon\cdot\sqrt{1-\varepsilon^2}\,\mathrm{Re}\langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|B\rangle. \end{aligned} \tag{B3}$$

In the following, we approximate $\varepsilon\cdot\sqrt{1-\varepsilon^2}\approx\varepsilon$. This approximation is not needed but it keeps the calculations clear. In addition, the formula we will derive from the approximated version is numerically more stable. In our program, we used the exact version (which we will not derive here) only if $\varepsilon(\gamma=0)>0.01$.

With $\varepsilon\cdot\sqrt{1-\varepsilon^2}\approx\varepsilon$, E is a parabola in ε. Assuming that the apex is the minimum, one gets

$$\varepsilon_{\min}(\gamma) = \frac{-\,\mathrm{Re}\langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|B\rangle}{\langle B|\widetilde{\mathbb{H}}|B\rangle - \langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|A^{[\text{refer}]}\rangle + \gamma}. \tag{B4}$$

Since $\|A^{[\gamma]}_{[n]} - A^{[\text{refer}]}_{[n]}\|\approx\varepsilon_{\min}$ for $\varepsilon_{\min}\ll 1$, we choose $\varepsilon_{\min}$ in accordance with Eq. (30) to be

$$\varepsilon_{\min}(\gamma) = \min\left[c\varepsilon_{\min}(\gamma=0),\Delta_{\max}\right]. \tag{B5}$$

From that, γ is calculated to be

$$\begin{aligned} \gamma &= \max(\gamma_{[c]},\gamma_{[\Delta_{\max}]}) \quad \text{with} \\ \gamma_{[c]} &= \frac{1-c}{c}\cdot(\langle B|\widetilde{\mathbb{H}}|B\rangle - \langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|A^{[\text{refer}]}\rangle), \\ \gamma_{[\Delta_{\max}]} &= \langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|A^{[\text{refer}]}\rangle - \langle B|\widetilde{\mathbb{H}}|B\rangle \\ &\quad - \frac{1}{\Delta_{\max}}\cdot\mathrm{Re}\langle A^{[\text{refer}]}|\widetilde{\mathbb{H}}|B\rangle. \end{aligned} \tag{B6}$$

1. Subspace projection and γ

As mentioned, we have to solve $\min\langle A^{[\gamma]}|\widetilde{\mathbb{H}}^{[\gamma]}|A^{[\gamma]}\rangle$ twice: first for $\gamma=0$ and after that for the final value of γ. The idea is to reuse the information gathered in the first minimization for the second run. As described in Sec. IV C, the minimization problem is projected onto a subspace. The first basis vector of this subspace is $|\mathfrak{A}_1\rangle = |A^{[\text{refer}]}\rangle$ [Eq. (38)]. Hence, the only element of the subspace matrix $\mathfrak{H}$ [Eq. (35)] which has to be adopted is $\mathfrak{H}_{1,1}\leftarrow\mathfrak{H}_{1,1}-\gamma$.

So, the update of the subspace matrix $\mathfrak{H}$ is extremely simple to perform, but one might wonder whether the associated basis vectors $|\mathfrak{A}_i\rangle$ are still optimal. For the Lanczos[29] and Arnoldi[30] algorithms, which are built on pure Krylov spaces, it turns out that the influence of γ on the basis vectors gets extinguished, while for the extended algorithm presented in Sec. IV C the optimal choice of the basis vectors shows a slight dependence on γ. Numerically, this is not a serious problem. Nonetheless, since the second optimization is the important one we recommend to use an approximated value of the final γ to construct the basis vectors in the first run. A simple and effective way is to use $\gamma_{[n-1]}$ from the last tensor optimization assuming $\gamma_{[n]}\approx\gamma_{[n-1]}$. Alternatively, one can solve the intermediate subspace matrices $\mathfrak{H}$ and use these intermediate results to calculate approximations of γ as described above.

APPENDIX C: TRANSFORMATION PROOF FOR DEGENERATE GROUND STATES

Let $\mathcal{A}$ and $\mathcal{B}$ represent two different ground states to the same Hamiltonian with a g times degenerate ground state level due to broken translational invariance:

$$\begin{aligned} \mathcal{A} &= \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0}\cdot\lambda^{\tilde{\alpha}_0\alpha_0}\cdot Q^{\alpha_0\alpha_1}_{[R]s_1}\cdot Q^{\alpha_1\alpha_2}_{[R]s_2}\cdots, \\ \mathcal{B} &= \ldots q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0}\cdot\xi^{\tilde{\alpha}_0\alpha_0}\cdot q^{\alpha_0\alpha_1}_{[R]s_1}\cdot q^{\alpha_1\alpha_2}_{[R]s_2}\cdots. \end{aligned} \tag{C1}$$

If degenerations due to further symmetries are involved, $\mathcal{A}$ and $\mathcal{B}$ are supposed to have the same characteristic values for these symmetries. This allows us to operate as if no further symmetries exist since all operations we are about to use leave these characteristic values unchanged. Under this condition, we will prove the existence of a matrix $\gamma^{\tilde{\alpha}_0\alpha_0}$ such that the ground state $\mathcal{B}$ can be expressed using the tensors Q stemming from the iMPS $\mathcal{A}$:

$$\mathcal{B} = \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0}\cdot\gamma^{\tilde{\alpha}_0\alpha_0}\cdot Q^{\alpha_0\alpha_1}_{[R]s_1}\cdot Q^{\alpha_1\alpha_2}_{[R]s_2}\cdots, \tag{C2}$$

where $\gamma^{\tilde{\alpha}_0\alpha_0} = \omega^{\tilde{\alpha}_0\beta}\cdot\xi^{\beta\delta}\cdot\nu^{\delta\alpha_0}$.

Let us start by surveying the elements of the proof. In order to show the claimed equation (C2), we will actually prove the gauge transformation

$$\begin{aligned} \ldots q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0} &= \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot Q^{\alpha_{-1}\beta}_{[L]s_0}\cdot\omega^{\beta\tilde{\alpha}_0}, \\ q^{\alpha_0\alpha_1}_{[R]s_1}\cdot q^{\alpha_1\alpha_2}_{[R]s_2}\cdots &= \nu^{\alpha_0\beta}\cdot Q^{\beta\alpha_1}_{[R]s_1}\cdot Q^{\alpha_1\alpha_2}_{[R]s_2}\cdots. \end{aligned} \tag{C3}$$

This gauge transformation will be proven for the case that the two underlying MPS represent the same physical state, which is not given for $\mathcal{A}$ and $\mathcal{B}$ [Eq. (C1)]. In order to use the gauge proof for our purpose, we need to find one physical state described by two different MPS where the first MPS can be constructed using the Q of $\mathcal{A}$ and the second by the q of $\mathcal{B}$. This one state which allows us to complete our proof is the translational invariant ground state. According to our preliminary remarks, this state is unique. Hence, if we succeed to construct two different MPS which represent a translational invariant ground state, we know that they represent the same physical state as demanded. We will prove the following construction for this state:

$$\begin{aligned} \mathcal{T}_Q &= \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0}\cdot\tau^{\tilde{\alpha}_0\alpha_0}\cdot Q^{\alpha_0\alpha_1}_{[R]s_1}\cdot Q^{\alpha_1\alpha_2}_{[R]s_2}\cdots \\ &= \ldots q^{\alpha_{-2}\alpha_{-1}}_{[L]s_{-1}}\cdot q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]s_0}\cdot\theta^{\tilde{\alpha}_0\alpha_0}\cdot q^{\alpha_0\alpha_1}_{[R]s_1}\cdot q^{\alpha_1\alpha_2}_{[R]s_2}\ldots = \mathcal{T}_q \end{aligned} \tag{C4}$$

with new tensors $\tau^{\tilde{\alpha}_0\alpha_0}$ and $\theta^{\tilde{\alpha}_0\alpha_0}$.

We start by showing the gauge transformation (C3). In order to increase clarity, we define

$$\overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} = Q^{\alpha_1}_{[1]s_1}\cdots Q^{\alpha_{k-1}\alpha_k}_{[n]s_k}, \quad \overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k} = p^{\alpha_1}_{[1]s_1}\cdots p^{\alpha_{k-1}\alpha_k}_{[n]s_k}. \tag{C5}$$

For this specific gauge proof, the different $Q_{[i]}$ do not need to be of the same structure and neither do the $p_{[i]}$. But, in contrast to the $p_{[i]}$, the $Q_{[i]}$ still have to fulfill Eq. (9), i.e.,

$$\begin{aligned} \overrightarrow{\mathbb{Q}}^{*\beta_k}_{s_1\ldots s_k}\cdot\overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} &= \delta^{\beta_k\alpha_k}, \quad \text{while} \\ \overrightarrow{\mathbb{Q}}^{\alpha_k}_{\bar{s}_1\ldots\bar{s}_k}\cdot\overrightarrow{\mathbb{Q}}^{*\alpha_k}_{s_1\ldots s_k} &\neq \delta_{(\bar{s}_1\ldots\bar{s}_k)(s_1\ldots s_k)}. \end{aligned} \tag{C6}$$

Although not an identity, the operator $\overrightarrow{\mathbb{Q}}^{\alpha_k}_{\bar{s}_1\ldots\bar{s}_k}\cdot\overrightarrow{\mathbb{Q}}^{*\alpha_k}_{s_1\ldots s_k}$ acts like such if it is applied from the left on any MPS with the structure

$\overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n}$, where $\overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n}$ represents an arbitrary right side of the MPS. Using the identity (C6), we get

$$\left(\overrightarrow{\mathbb{Q}}^{\beta_k}_{\bar{s}_1\ldots\bar{s}_k} \cdot \overrightarrow{\mathbb{Q}}^{*\beta_k}_{s_1\ldots s_k}\right) \cdot \left(\overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n}\right) = \overrightarrow{\mathbb{Q}}^{\beta_k}_{\bar{s}_1\ldots\bar{s}_k} \cdot \delta^{\beta_k\alpha_k} \cdot \overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n} = \overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n}. \quad \text{(C7)}$$

In the following, we assume that the two MPS

$$\overrightarrow{\mathbb{Q}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{R}}^{\alpha_k}_{s_{k+1}\ldots s_n} = \overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{M}}^{\alpha_k}_{s_{k+1}\ldots s_n} \quad \text{(C8)}$$

represent the same physical state. Now, let us apply the operator $\overrightarrow{\mathbb{Q}}^{\beta_k}_{\bar{s}_1\ldots\bar{s}_k} \cdot \overrightarrow{\mathbb{Q}}^{*\beta_k}_{s_1\ldots s_k}$ on this equation. Due to Eq. (C7), the left side stays unchanged. Hence, the physical state represented by the right side does not change either:

$$\begin{aligned}\overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{M}}^{\alpha_k}_{s_{k+1}\ldots s_n} &= \overrightarrow{\mathbb{Q}}^{\beta_k}_{\bar{s}_1\ldots\bar{s}_k} \cdot \overrightarrow{\mathbb{Q}}^{*\beta_k}_{s_1\ldots s_k} \cdot \overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k} \cdot \overleftarrow{\mathbb{M}}^{\alpha_k}_{s_{k+1}\ldots s_n} \\ &= \overrightarrow{\mathbb{Q}}^{\beta_k}_{\bar{s}_1\ldots\bar{s}_k} \cdot \omega^{\beta_k\alpha_k} \cdot \overleftarrow{\mathbb{M}}^{\alpha_k}_{s_{k+1}\ldots s_n}, \end{aligned} \quad \text{(C9)}$$

with $\omega^{\beta\alpha_k} = \overrightarrow{\mathbb{Q}}^{*\beta_k}_{s_1\ldots s_k} \cdot \overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k}$. The existence of the inverse $\overleftarrow{\mathbb{M}}^{-1}$ with $\overleftarrow{\mathbb{M}}^{\alpha_k}_{s_{k+1}\ldots s_n} \cdot \overleftarrow{\mathbb{M}}^{-1\,\beta_k}_{s_{k+1}\ldots s_n} = \delta^{\alpha_k\beta_k}$ can be assured: if the rank of $\overleftarrow{\mathbb{M}}^{\alpha_k}_{(s_{k+1}\ldots s_n)}$ is smaller than the bond dimension α_k, the value of α_k can be reduced since in that case it turns out to be unnecessarily high. Applying $\overleftarrow{\mathbb{M}}^{-1}$ from the right on Eq. (C9), we end up with

$$\overrightarrow{\mathbb{P}}^{\alpha_k}_{s_1\ldots s_k} = \overrightarrow{\mathbb{Q}}^{\beta_k}_{s_1\ldots s_k} \cdot \omega^{\beta_k\alpha_k}. \quad \text{(C10)}$$

For $k \to \infty$, this covers the first part of the heralded gauge equation (C3). The second part of Eq. (C3) is proved by a straightforward application of the arguments used above on the right side of the MPS.

Now, we have to show the MPS construction of the translational invariant ground state (C4). As above, we assume that the ground state level of the Hamiltonian under consideration is g times degenerate due to broken translational invariance. Let T_L be the operator which shifts all sites of an MPS by one position to the left and $T_R = (T_L)^{-1}$. For the MPS $\mathcal{A}$ representing any of the possible ground states, we get

$$\begin{aligned} T_L \cdot \mathcal{A} \neq \mathcal{A}, \quad (T_L)^g \cdot \mathcal{A} = \mathcal{A}, \\ \mathcal{T} := \sum_{j=0}^{g-1} (T_L)^j \cdot \mathcal{A}, \quad T_L \cdot \mathcal{T} = \mathcal{T}, \end{aligned} \quad \text{(C11)}$$

where $\mathcal{T}$ represents an unnormalized version of the translational invariant ground state, which we like to construct. As an intermediate step, we like to prove Eq. (C15) below. Therefore, we have to look at the effect T_L has on $\mathcal{A}$ [using Eq. (C1) for $\mathcal{A}$]:

$$\begin{aligned} T_L\big(\ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]\,s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]\,s_0} \cdot \lambda^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]\,s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]\,s_2} \ldots\big) \\ = \ldots Q^{\alpha_{-2}\tilde{\alpha}_{-1}}_{[L]\,s_{-1}} \cdot \lambda^{\tilde{\alpha}_{-1}\alpha_{-1}} \cdot Q^{\alpha_{-1}\alpha_0}_{[R]\,s_0} \cdot Q^{\alpha_0\alpha_1}_{[R]\,s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]\,s_2} \cdots. \end{aligned} \quad \text{(C12)}$$

Next, we look at the MPS $(T_R)^g \cdot T_L \cdot \mathcal{A}$, which has the form

$$\ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]\,s_{-1}} Q^{\alpha_{-1}\alpha_0}_{[L]\,s_0} \ldots Q^{\alpha_{g-2}\tilde{\alpha}_{g-1}}_{[L]\,s_{g-1}} \lambda^{\tilde{\alpha}_{g-1}\alpha_{g-1}} Q^{\alpha_{g-1}\alpha_g}_{[R]\,s_g} \ldots. \quad \text{(C13)}$$

Since the two MPS $(T_R)^g \cdot T_L \cdot \mathcal{A} = T_L \cdot \mathcal{A}$ describe the same physical state, we are allowed to apply the gauge transformation (C10) and identify

$$\ldots Q^{\alpha_{-2}\tilde{\alpha}_{-1}}_{[L]\,s_{-1}} \cdot \lambda^{\tilde{\alpha}_{-1}\alpha_{-1}} \cdot Q^{\alpha_{-1}\alpha_0}_{[R]\,s_0} = \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]\,s_{-1}} \cdot Q^{\alpha_{-1}\gamma}_{[L]\,s_0} \cdot \omega^{\gamma\tilde{\alpha}_0}. \quad \text{(C14)}$$

Inserting this expression into Eq. (C12), we arrive at the following description for $T_L \cdot \mathcal{A}$:

$$\begin{aligned} T_L\big(\ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]\,s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]\,s_0} \cdot \lambda^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]\,s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]\,s_2} \ldots\big) \\ = \ldots Q^{\alpha_{-2}\alpha_{-1}}_{[L]\,s_{-1}} \cdot Q^{\alpha_{-1}\tilde{\alpha}_0}_{[L]\,s_0} \cdot \omega^{\tilde{\alpha}_0\alpha_0} \cdot Q^{\alpha_0\alpha_1}_{[R]\,s_1} \cdot Q^{\alpha_1\alpha_2}_{[R]\,s_2} \ldots. \end{aligned} \quad \text{(C15)}$$

As we see, applying the operator T_L on $\mathcal{A}$ has the same effect as the replacement of the matrix $\lambda^{\tilde{\alpha}_0\alpha_0}$ by $\omega^{\tilde{\alpha}_0\alpha_0}$. Since higher powers of T_L can be created by an iteration of the arguments just presented, it follows that the effect of any $(T_L)^j$ on $\mathcal{A}$ can be accounted for by an accordingly calculated $\omega^{\tilde{\alpha}_0\alpha_0}_{[j]}$. Following that construction, the only difference between the various MPS $(T_L)^j \cdot \mathcal{A}$ is their tensor $\omega^{\tilde{\alpha}_0\alpha_0}_{[j]}$ and hence the task of Eq. (C11) to sum up these MPS reduces to a summation of the $\omega^{\tilde{\alpha}_0\alpha_0}_{[j]}$. In other words: Replacing the matrix $\lambda^{\tilde{\alpha}_0\alpha_0}$ in the MPS $\mathcal{A}$ by $\tau^{\tilde{\alpha}_0\alpha_0}$,

$$\tau^{\tilde{\alpha}_0\alpha_0} = \sum_{j=0}^{g-1} \omega^{\tilde{\alpha}_0\alpha_0}_{[j]}, \quad \text{(C16)}$$

results in the translational invariant MPS $\mathcal{T}$. Of course, the same arguments can be applied to the MPS $\mathcal{B}$ giving us $\mathcal{T}_Q = \mathcal{T} = \mathcal{T}_q$ as claimed in Eq. (C4).

Let us review our arguments: By virtue of Eq. (C16), we can transform the MPS $\mathcal{A}$ and $\mathcal{B}$ given in Eq. (C1) such that we end up with the translational invariant ground states $\mathcal{T}_Q$ and $\mathcal{T}_q$ as claimed in Eq. (C4). Since we work under the condition that $\mathcal{T}_Q = \mathcal{T}_q$, we are allowed to use the gauge transformation (C10) to replace the q of $\mathcal{T}_q$ by the Q of $\mathcal{T}_Q$. The same replacement is possible in $\mathcal{B}$ because the q in $\mathcal{B}$ and $\mathcal{T}_q$ are identical (as are the Q in $\mathcal{A}$ and $\mathcal{T}_Q$). This concludes the proof of Eq. (C2) we aimed for.

APPENDIX D: DAVIDSON IMPLEMENTATION

In this section, we introduce a practical implementation resembling the Davidson[24] method based on recycled information of the previous round, which allows us to improve the update equation (37) of the iterative eigenvector solver presented in Sec. IV C. We adopt the same notation as in that section but mostly drop the index $[n]$ to keep the formulas clean.

The best possible new vector $|\mathfrak{A}_{k+1}\rangle$ the iterative eigenvector solver could come up with to replace Eq. (37) is an orthonormalized version of $|\Delta_0\rangle = |E_0\rangle - |e_0\rangle$:

$$\begin{aligned} \widetilde{\mathbb{H}} \cdot |E_0\rangle &= E_0 \cdot |E_0\rangle, \\ \widetilde{\mathbb{H}} \cdot (|e_0\rangle + |\Delta_0\rangle) &= E_0 \cdot (|e_0\rangle + |\Delta_0\rangle), \\ (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}}) \cdot |\Delta_0\rangle &= (\widetilde{\mathbb{H}} - E_0 \cdot \mathbb{I}) \cdot |e_0\rangle, \\ (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}}) \cdot |\Delta_0\rangle &\approx (\widetilde{\mathbb{H}} - e_0 \cdot \mathbb{I}) \cdot |e_0\rangle, \\ (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}}) \cdot |\Delta_0\rangle &= |r\rangle, \\ |\Delta_0\rangle &= (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}})^{-1} \cdot |r\rangle, \end{aligned} \quad \text{(D1)}$$

where we used the definition (36) for $|r\rangle$. The Davidson method requires a workable approximation for the nontrivial operator $\mathfrak{D} = (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}})^{-1}$. At this point, we take advantage of the expectation that the operator $\mathfrak{D}_{[n]}$ calculated in round n should look pretty much the same as $\mathfrak{D}_{[n-1]}$ calculated in round $n-1$:

$$\begin{aligned} \mathfrak{D}_{[n]} &\approx \mathfrak{D}_{[n-1]}, \\ (E_{[n]0} \cdot \mathbb{I} - \widetilde{\mathbb{H}}_{[n]})^{-1} &\approx (E_{[n-1]0} \cdot \mathbb{I} - \widetilde{\mathbb{H}}_{[n-1]})^{-1}. \end{aligned} \tag{D2}$$

Hence, we use the accumulated data at the end of round $n-1$ for an efficient one-time estimation of $\mathfrak{D}_{[n-1]}$, which we will apply in round n.

In order to calculate $\mathfrak{D}$, we need a simplified form of $\widetilde{\mathbb{H}}$ which allows easy inversion. We know k approximated eigenvectors $|e_{i<k}\rangle$ of $\widetilde{\mathbb{H}}$. In order to have an orthonormal basis for $\widetilde{\mathbb{H}}$, we imagine $N-k$ further $|\hat{e}_{k\leqslant i<N}\rangle$, where $\dim(\widetilde{\mathbb{H}}) = N \times N$. With that, we approximate $\widetilde{\mathbb{H}}$ as

$$\begin{aligned} \widetilde{\mathbb{H}} &\approx \sum_{i=0}^{k-1} \widetilde{\mathbb{H}} |e_i\rangle\langle e_i| + \alpha \cdot \sum_{i=k}^{N} |\hat{e}_i\rangle\langle \hat{e}_i| \\ &= \sum_{i=0}^{k-1} (\widetilde{\mathbb{H}} - \alpha)|e_i\rangle\langle e_i| + \alpha \cdot \mathbb{I} \end{aligned} \tag{D3}$$

with an average eigenvalue $\alpha = \text{const}$ for the unknown eigenvectors. One might be tempted to simplify Eq. (D3) using $\widetilde{\mathbb{H}}|e_i\rangle \approx e_i \cdot |e_i\rangle$, but we do not since the eigenvectors are not very well approximated except for $|e_0\rangle$. To be able to perform the inversion in Eq. (D6), it suffices to resort to the exact result

$$\langle e_i | \widetilde{\mathbb{H}} | e_j \rangle = e_i \cdot \delta_{ij} \quad \text{without} \sum_i, \tag{D4}$$

which is a consequence of the construction (35). Further, with the results gathered during the optimization (Sec. IV C), the $\widetilde{\mathbb{H}}|e_i\rangle$ are as quickly calculated as the $|e_i\rangle$.

Next, we take the trace of Eq. (D3) and set α such that both sides are equal:

$$\operatorname{tr}(\widetilde{\mathbb{H}}) = \langle e_i | \widetilde{\mathbb{H}} | e_i \rangle + \alpha(N-k), \quad \alpha = \frac{\operatorname{tr}(\widetilde{\mathbb{H}}) - \sum_{i=0}^{k-1} e_i}{N-k}. \tag{D5}$$

The trace of the exact $\widetilde{\mathbb{H}}$ is efficiently calculated by already tracing over its components $L^{\alpha_l' \mu_l \alpha_l}$ and $R^{\alpha_r' \mu_r \alpha_r}$ before assembling them [Eq. (15)].

Now, we insert the approximated $\widetilde{\mathbb{H}}$ [Eq. (D3)] in $\mathfrak{D}$:

$$\begin{aligned} \mathfrak{D} &= (E_0 \cdot \mathbb{I} - \widetilde{\mathbb{H}})^{-1} \\ &\approx \left((E_0 - \alpha) \cdot \mathbb{I} - \sum_{i=0}^{k-1} (\widetilde{\mathbb{H}} - \alpha)|e_i\rangle\langle e_i| \right)^{-1}. \end{aligned} \tag{D6}$$

The inversion is solved by

$$\mathfrak{D} = (E_0 - \alpha)^{-1} \cdot \left(\mathbb{I} + \sum_{i=0}^{k-1} \frac{\widetilde{\mathbb{H}} - \alpha}{E_0 - e_i} |e_i\rangle\langle e_i| \right), \tag{D7}$$

as can be verified inserting the result in $\mathfrak{D} \cdot \mathfrak{D}^{-1} = \mathbb{I}$.

The final question we have to answer is which value we assign to the unknown exact eigenvalue E_0. The best approximation [which we already used in Eq. (D1)] is $E_0 \approx e_0$, but this produces a singularity in $\mathfrak{D}$. There are two ways out. First, we can always pick E_0 a little bit lower $E_0 := e_0 - \varepsilon$. Second, we should discard the troublesome term $\sim |e_0\rangle\langle e_0|$ in $\mathfrak{D}$ anyway for the following reason: We replace Eq. (37) by

$$|\mathfrak{A}_{k+1}\rangle = \mathfrak{D} \cdot |r\rangle, \tag{D8}$$

where $|r\rangle \perp |\mathfrak{A}_1\rangle = |A_{[n]}^{[\text{refer}]}\rangle \approx |e_0^{[n]}\rangle \approx |e_0^{[n-1]}\rangle$ and with that $|e_0\rangle\langle e_0|r\rangle \approx 0 \cdot |e_0\rangle$. Afterwards, residual parts of the $|e_0\rangle\langle e_0|$ term are exfiltrated again because $|\mathfrak{A}_{k+1}\rangle$ has to be orthogonalized (and normalized)

$$|\mathfrak{A}_{k+1}\rangle \longleftarrow \left(\mathbb{I} - \sum_{i=1}^{k} |\mathfrak{A}_i\rangle\langle \mathfrak{A}_i| \right) \cdot |\mathfrak{A}_{k+1}\rangle. \tag{D9}$$

These considerations are also part of the more elaborated Jacobi-Davidson[24] method to which this implementation can be extended.

We might further consider to omit terms with $e_i \gg e_0$ since their influence shrinks with $(e_0 - e_i)^{-1}$. At the end of the day, the effort to construct $\mathfrak{D}$ as well as the effort for each application scale with N times the number of $|e_i\rangle\langle e_i|$ terms used in $\mathfrak{D}$.

APPENDIX E: ALTERED MINIMIZATION

Here, we derive the missing equations of Sec. III D. First, we search for an approximation of $\lambda_{[L/R]}$ and start by an alternative way of expressing them as

$$\lambda_{[L]}^{\alpha\beta} = Q_{[L]s}^{*\alpha\gamma} \cdot A_s^{\gamma\beta}; \quad \lambda_{[R]}^{\alpha\beta} = A_s^{\alpha\gamma} \cdot Q_{[R]s}^{*\gamma\beta}, \tag{E1}$$

where we used the orthogonality (9) of the Q and the decomposition (8). Next, we use the fact that the algorithm is tuned to produce consecutive tensors $A_{[n-1]}, A_{[n]}$ which are quite similar. Hence, we approximate the yet unknown $Q_{[L/R]}^{*}$ [Eq. (E1)] by their known predecessor of the optimization round before, i.e., $Q_{[L/R]}^{[n]*} \approx Q_{[L/R]}^{[n-1]*}$:

$$\bar{\lambda}_{[L]}^{[n]\alpha\beta} = Q_{[L]s}^{[n-1]*\alpha\gamma} \cdot A_s^{[n]\gamma\beta}, \quad \bar{\lambda}_{[R]}^{[n]\alpha\beta} = A_s^{[n]\alpha\gamma} \cdot Q_{[R]s}^{[n-1]*\gamma\beta}. \tag{E2}$$

Next, we use the same idea and replace the $Q_{[L/R]}^{[n-1]}$ in definition (42) by the $Q_{[L/R]}^{[n]}$:

$$\begin{aligned} |\bar{\mathfrak{A}}_i\rangle &\approx \tfrac{1}{2}|\mathfrak{A}_i\rangle + \tfrac{1}{4}\left(Q_{[L]}^{[n]} \cdot \bar{\lambda}_{[R]i} + \bar{\lambda}_{[L]i} \cdot Q_{[R]}^{[n]}\right) \quad \text{with} \\ \bar{\lambda}_{[R]i}^{\alpha\beta} &\approx |\mathfrak{A}_i\rangle_s^{\alpha\gamma} \cdot Q_{[R]s}^{[n]*\gamma\beta}, \quad \bar{\lambda}_{[L]i}^{[n]\alpha\beta} \approx Q_{[L]s}^{[n]*\alpha\gamma} \cdot |\mathfrak{A}_i\rangle_s^{\gamma\beta}. \end{aligned} \tag{E3}$$

With that, we like to calculate $|\bar{A}\rangle = |\bar{\mathfrak{A}}_i\rangle \cdot a_i$ [Eq. (44)]. Therefore, we first observe

$$\bar{\lambda}_{[L]i}^{[n]} \cdot a_i \approx Q_{[L]s}^{[n]*\alpha\gamma} \cdot |\mathfrak{A}_i\rangle_s^{\gamma\beta} \cdot a_i = Q_{[L]s}^{[n]*\alpha\gamma} \cdot A_s^{\gamma\beta} = \lambda_{[L]}^{[n]\alpha\beta}, \tag{E4}$$

where we used $A = |\mathfrak{A}_i\rangle \cdot a_i$ [Eq. (41)] in line two and Eq. (E1) in line three. Likewise, we find $\bar{\lambda}_{[R]i} \cdot a_i \approx \lambda_{[R]}$ and

$$\begin{aligned}|\bar{\mathfrak{A}}_i\rangle \cdot a_i &\approx \left[\tfrac{1}{2}|\mathfrak{A}_i\rangle + \tfrac{1}{4}(Q_{[L]} \cdot \bar{\lambda}_{[R]i} + \bar{\lambda}_{[L]i} \cdot Q_{[R]})\right] \cdot a_i \\ &\approx \tfrac{1}{2}|A\rangle + \tfrac{1}{4}(Q_{[L]} \cdot \lambda_{[R]} + \lambda_{[L]} \cdot Q_{[R]}) \\ &= \tfrac{1}{4}|A\rangle + \tfrac{1}{4}|A\rangle + \tfrac{1}{4}Q_{[L]} \cdot \lambda_{[R]} + \tfrac{1}{4}\lambda_{[L]} \cdot Q_{[R]} \\ &= \tfrac{1}{4}Q_{[L]} \cdot \lambda_{[L]} + \tfrac{1}{4}Q_{[R]} \cdot \lambda_{[R]} \\ &\quad + \tfrac{1}{4}Q_{[L]} \cdot \lambda_{[R]} + \tfrac{1}{4}\lambda_{[L]} \cdot Q_{[R]} \\ &= \tfrac{1}{2}(Q_{[L]} \cdot \lambda_{[\mathrm{sym}]} + \lambda_{[\mathrm{sym}]} \cdot Q_{[R]}) \quad \text{with} \\ \lambda_{[\mathrm{sym}]} &= \tfrac{1}{2}(\lambda_{[L]} + \lambda_{[R]}),\end{aligned} \tag{E5}$$

as claimed in Eq. (44).

APPENDIX F: MIRROR SYMMETRY

Here, we show that in case of a mirror-symmetricHamiltonian $\mathcal{H}$, i.e., a Hamiltonian that is invariant under inversion of the order of its sites

$$\mathcal{H}^{s_1' \ldots s_n'}_{s_1 \ldots s_n} = \mathcal{H}^{s_n' \ldots s_1'}_{s_n \ldots s_1}, \tag{F1}$$

all tensors $A^{\alpha_l \alpha_r}_{[n]\,s}$ can be chosen mirror symmetrical in their auxiliary indices α_l,α_r, i.e.,

$$A^{\alpha_l \alpha_r}_{[n]\,s} = A^{\alpha_r \alpha_l}_{[n]\,s}. \tag{F2}$$

This allows us to impose an extra constraint on $A_{[n]}$. Commuting the indices α_l,α_r also results in

$$Q^{\alpha_l \alpha_r}_{[L]\,s} = Q^{\alpha_r \alpha_l}_{[R]\,s}, \quad \lambda^{\alpha_l \alpha_r}_{[L]} = \lambda^{\alpha_r \alpha_l}_{[R]}, \tag{F3}$$

as can be seen directly from the decomposition (8). Further, it is possible to construct $L^{\alpha_l' \mu_l \alpha_l}$ and $R^{\alpha_r' \mu_r \alpha_r}$ [Eq. (19)] such that they are identical. But, therefore we need to resort to an alternative MPO construction for the Hamiltonian, such that the MPO tensors of the left half are mirror symmetric to the tensors of the right half. This can be achieved if we include a special interface tensor in the middle where $L^{\alpha_l' \mu_l \alpha_l}$ and $R^{\alpha_r' \mu_r \alpha_r}$ are connected to build $\widetilde{\mathbb{H}}$. This reduces the requirement in storage memory roughly by a factor of 2, but has nearly no effect on the speed. Since storage capacities are usually not a big issue, we do not elaborate this point any further.

One should be aware that Eq. (F1) enforces a mirror-symmetric MPS. In case of a mirror-symmetry-breaking ground state, the MPS will represent a superposition of both chiralities, implying an unfavorably increased requirement in bond dimension. We further remark that our definition of a mirror-symmetric Hamiltonian does not forcedly imply a symmetry in real space. Although in practical application the order of the sites generally coincides with one specific spatial direction, there is no mathematical connection between the direction of space and the chosen order.

Assuming a mirror-symmetric Hamiltonian $\mathcal{H}$ [Eq. (F1)], the claim of the mirror-symmetric tensor $A^{\alpha_l \alpha_r}_{[n]\,s}$ [Eq. (F2)] can be proven iteratively:

(1) If $A^{\alpha_l \alpha_r}_{[i]\,s} = A^{\alpha_r \alpha_l}_{[i]\,s}$ for all $i < n$, then $\widetilde{\mathbb{H}}_{[n]}$ [Eq. (18)] is mirror symmetric, i.e., $\widetilde{\mathbb{H}}^{\alpha_l' \alpha_l \alpha_r' \alpha_r}_{[n]\,ss'} = \widetilde{\mathbb{H}}^{\alpha_r' \alpha_r \alpha_l' \alpha_l}_{[n]\,ss'}$.

(2) If 1 is fulfilled, then $A^{\alpha_l \alpha_r}_{[n]\,s} = A^{\alpha_r \alpha_l}_{[n]\,s}$ [Eq. (F2)].

The very first $\widetilde{\mathbb{H}}_{[1]}$ is constructed via the initialization procedure described in Sec. II E2. If we start with a mirror-symmetric wave function and use Takagi's factorization as suggested in 3 of the initialization procedure, $\widetilde{\mathbb{H}}_{[1]}$ is symmetric and the induction is well grounded.

Proof for 1. All $\widetilde{\mathbb{H}}$ represent a sum of operators according to Eq. (18):

$$\widetilde{\mathbb{H}} = \sum_{i=1}^{4^n} \mathbb{H}_i. \tag{F4}$$

Each of these $\mathbb{H}_i$ contains an entire Hamiltonian MPO as central unit sandwiched by bra and ket MPS, with a hole where the new tensor $A_{[\mathrm{new}]} = A_{[n]}$ is supposed to be inserted. The Hamiltonian is guaranteed to be mirror symmetric, while there is no such condition for the MPS. The different MPS are encoded in the building blocks $L^{\alpha_l' \mu_l \alpha_l}$ and $R^{\alpha_r' \mu_r \alpha_r}$, which are constructed symmetrically [Eq. (19)], i.e., in contrast to the basic algorithm each new tensor is inserted in $L^{\alpha_l' \mu_l \alpha_l}$ and $R^{\alpha_r' \mu_r \alpha_r}$ at equal footing. Hence, for each MPS exists a counterpart which contains exactly the same tensors in inverted order. In general, this alone is not enough because mirror symmetry also exchanges the left and right auxiliary indices. But, since all involved tensors $A^{\alpha_l \alpha_r}_{[i]\,s} = A^{\alpha_r \alpha_l}_{[i]\,s}$ are supposed to be invariant under this kind of exchange {and $Q^{\alpha_l \alpha_r}_{[L]\,s} = Q^{\alpha_r \alpha_l}_{[R]\,s}$ [Eq. (F3)], as needed}, mirror-symmetric counterparts for all involved MPS are guaranteed and with that $\widetilde{\mathbb{H}}$ is mirror symmetric.

Proof for 2. The tensor $A^{\alpha_l \alpha_r}_{[n]\,s}$ is the result of the minimization procedure described in more details in Sec. IV C. Each element in this procedure maintains mirror symmetry if $\widetilde{\mathbb{H}}_{[n]}$ and the initial tensor $A^{[\mathrm{refer}]}_{[n]}$ [Eq. (27)] are mirror symmetric. Since $A^{[\mathrm{refer}]}_{[n]}$ is a superposition of mirror-symmetric tensors, all conditions are met.

Finally, we remark that accumulating numerical errors might undermine the symmetry. Therefore, we recommend to explicitly restore the symmetry of each $A^{\alpha_l \alpha_r}_{[n]\,s}$ during its calculation.

[1] U. Schollwoeck, Ann. Phys. (NY) **326**, 96 (2011).

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

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

[4] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, Phys. Rev. Lett. **59**, 799 (1987).

[5] J. Dukelsky, M. A. Martín-Delgado, T. Nishino, and G. Sierra, Europhys. Lett. **43**, 457 (1998).

[6] G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007).

[7] R. Hübener, C. Kruszynska, L. Hartmann, W. Dür, F. Verstraete, J. Eisert, and M. B. Plenio, Phys. Rev. A **79**, 022317 (2009).

[8] R. Hübener, V. Nebendahl, and W. Dür, New J. Phys. **12**, 025004 (2010).

[9] J. Eisert, Phys. Rev. Lett. **97**, 260501 (2006).

[10] N. Schuch, I. Cirac, and F. Verstraete, Phys. Rev. Lett. **100**, 250501 (2008).

[11]F. J. Burnell, M. M. Parish, N. R. Cooper, and S. L. Sondhi, Phys. Rev. B **80**, 174519 (2009).
[12]J. Schachenmayer, I. Lesanovsky, A. Micheli, and A. J. Daley, New J. Phys. **12**, 103044 (2010).
[13]C. Trefzger, C. Menotti, B. Capogrosso-Sansone, and M. Lewenstein, J. Phys. B: At., Mol. Opt. Phys. **44**, 193001 (2011).
[14]J. Cardy, *Scaling and Renormalization in Statistical Physics* (Cambridge University Press, Cambridge, UK, 1996).
[15]B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo, Phys. Rev. B **86**, 075117 (2012).
[16]A. Gendiar, M. Daniska, Y. Lee, and T. Nishino, Phys. Rev. A **83**, 052118 (2011).
[17]G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).
[18]I. P. McCulloch, J. Stat. Mech. (2007) P10014.
[19]V. Murg, J. Cirac, B. Pirvu, and F. Verstraete, New J. Phys. **12**, 025012 (2010).
[20]G. M. Crosswhite and D. Bacon, Phys. Rev. A **78**, 012356 (2008).
[21]F. Fröwis, V. Nebendahl, and W. Dür, Phys. Rev. A **81**, 062337 (2010).
[22]G. M. Crosswhite, A. C. Doherty, and G. Vidal, Phys. Rev. B **78**, 035116 (2008) .
[23]I. P. McCulloch, arXiv:0804.2509.
[24]G. L. G. Sleijpen and H. A. Van der Vorst, SIAM Rev. **42**, 267 (2000).
[25]M. B. Hastings, J. Stat. Mech. (2007) P08024.
[26]H. Ueda, I. Maruyama, and K. Okunishi, J. Phys. Soc. Jpn. **80**, 023001 (2011).
[27]H. Ueda and I. Maruyama, Phys. Rev. B **86**, 064438 (2012).
[28]Y. Saad, *Iterative Methods for Sparse Linear Systems,* 2nd ed. (SIAM, Philadelphia, 2003).
[29]C. Lanczos, J. Res. Natl. Bur. Stand. (US) **45**, 255 (1951).
[30]W. Arnoldi, Q. Appl. Math. **9**, 17 (1951).
[31]S. Furukawa, M. Sato, S. Onoda, and A. Furusaki, Phys. Rev. B **86**, 094417 (2012).
[32]K. Okunishi, J. Phys. Soc. Jpn. **77**, 114004 (2008).
[33]F. J. Burnell, arXiv:1004.5595.
[34]P. Bak and R. Bruinsma, Phys. Rev. Lett. **49**, 249 (1982).
[35]T. Giamarchi, *Quantum Physics in One Dimension* (Oxford University Press, Oxford, 2003).
[36]L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre, Phys. Rev. B **78**, 024410 (2008).
[37]F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore, Phys. Rev. Lett. **102**, 255701 (2009).