

 Open access • Journal Article • DOI:10.1103/PHYSREVB.91.155115

Strictly single-site DMRG algorithm with subspace expansion — [Source link](#)

[Claudius Hubig](#), [Ian P. McCulloch](#), [Ulrich Schollwöck](#), [F. A. Wolf](#)

Institutions: [Ludwig Maximilian University of Munich](#), [University of Queensland](#)

Published on: 10 Apr 2015 - [Physical Review B](#) (American Physical Society)

Topics: [Energy \(signal processing\)](#) and [Dimension \(graph theory\)](#)

Related papers:

- [The density-matrix renormalization group in the age of matrix product states](#)
- [Density matrix formulation for quantum renormalization groups](#)
- [Density-matrix algorithms for quantum renormalization groups.](#)
- [The density-matrix renormalization group](#)
- [From density-matrix renormalization group to matrix product states](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/strictly-single-site-dmrg-algorithm-with-subspace-expansion-4d4ge5moqh>

Strictly single-site DMRG algorithm with subspace expansion

C. Hubig,^{1,*} I. P. McCulloch,² U. Schollwöck,¹ and F. A. Wolf¹

¹*Department of Physics and Arnold Sommerfeld Center for Theoretical Physics, Ludwig-Maximilians-Universität München, Theresienstrasse 37, 80333 München, Germany*

²*Centre for Engineered Quantum Systems, School of Physical Sciences, The University of Queensland, Brisbane, Queensland 4072, Australia*
(Received 26 January 2015; revised manuscript received 26 March 2015; published 10 April 2015)

We introduce a strictly single-site DMRG algorithm based on the subspace expansion of the alternating minimal energy (AMEn) method. The proposed new MPS basis enrichment method is sufficient to avoid local minima during the optimization, similar to the density matrix perturbation method, but computationally cheaper. Each application of \hat{H} to $|\Psi\rangle$ in the central eigensolver is reduced in cost for a speed-up of $\approx (d+1)/2$, with d the physical site dimension. Further speed-ups result from cheaper auxiliary calculations and an often greatly improved convergence behavior. Runtime to convergence improves by up to a factor of 2.5 on the Fermi-Hubbard model compared to the previous single-site method and by up to a factor of 3.9 compared to two-site DMRG. The method is compatible with real-space parallelization and non-Abelian symmetries.

DOI: [10.1103/PhysRevB.91.155115](https://doi.org/10.1103/PhysRevB.91.155115)

PACS number(s): 05.10.Cc, 02.70.-c, 71.27.+a

I. INTRODUCTION

Since its introduction in 1993 [1,2], the density matrix renormalization group method (DMRG) has seen tremendous use in the study of one-dimensional systems [3,4]. Various improvements such as real-space parallelization [5], the use of Abelian and non-Abelian symmetries [6], and multigrid methods [7] have been proposed. Most markedly, the introduction [8] of density matrix perturbation steps allowed the switch from two-site DMRG to single-site DMRG in 2005, which provided a major speed-up and improved convergence in particular for systems with long-range interactions.

Nevertheless, despite some progress [9–11], (nearly) two-dimensional systems, such as long cylinders, are still a hard problem for DMRG. The main reason for this is the different scaling of entanglement due to the area law [12,13]: in one dimension, entanglement and hence matrix dimensions in DMRG are essentially size-independent for ground states of gapped systems, whereas in two dimensions, entanglement grows linearly and matrix dimensions roughly exponentially with system width.

As a result, the part of the Hilbert space considered by DMRG during its ground state search increases dramatically, resulting mainly in three problems as follows. First, the DMRG algorithm becomes numerically more challenging as the sizes of matrices involved grow [we will assume matrix-matrix multiplications to scale as $O(m^3)$ throughout the paper]. Second, the increased search space size makes it more likely to get stuck in local minima. Third, while sequential updates work well in one-dimensional (1D) chains with short-range interactions, nearest-neighbor sites in the 2D lattice can be separated much farther in the DMRG chain. Therefore, improvements to the core DMRG algorithm are still highly worthwhile.

In this paper, we will adopt parts of the AMEn method [14] developed in the tensor train/numerical linear algebra community to construct a strictly single-site DMRG algorithm that works without accessing the (full) reduced density

matrix. Compared to the existing *center matrix wave function formalism* (CWF) [15], we achieve a speed-up of $\approx (d+1)/2$ during each application of \hat{H} to $|\Psi\rangle$ in the eigensolver during the central optimization routine, where d is the dimension of the physical state space on each site.

The layout of this paper is as follows. Section II will establish the notation. Section III will recapitulate the density matrix perturbation method and the CWF. Section IV will introduce the subspace expansion method and the heuristic expansion term with a simple two-spin example. The strictly single-site DMRG algorithm (DMRG3S) will be presented in Sec. V alongside a comparison with the existing CWF. As both the original perturbation method and the heuristic subspace expansion require a *mixing factor* α [8], Sec. VI describes how to adaptively choose α for fastest convergence. Numerical comparisons and examples will be given in Sec. VII.

II. DMRG BASICS

The notation established here closely follows the review article in Ref. [4]. Consider a state $|\Psi\rangle$ of a system of l sites. Each site has a physical state dimension d_i , e.g., $\forall i : d_i = 3$, $l = 50$ for a system of 50 $S = 1$ spins:

$$|\Psi\rangle = \sum_{\sigma_1 \dots \sigma_l} c_{\sigma_1 \dots \sigma_l} |\sigma_1 \dots \sigma_l\rangle. \quad (1)$$

In practice, the dimension of the physical basis is usually constant, $\forall i : d_i = d$, but we will keep the subscript to refer to one specific basis on site i where necessary.

It is then possible to decompose the coefficients $c_{\sigma_1 \dots \sigma_l}$ as a series of rank-3 tensors M_1, \dots, M_l of size (d_i, m_{i-1}, m_i) , respectively, with $m_0 = m_l = 1$. The coefficient $c_{\sigma_1 \dots \sigma_l}$ can then be written as the matrix product of the corresponding matrices in M_1, \dots, M_l :

$$|\Psi\rangle = \sum_{\sigma_1 \dots \sigma_l} \underbrace{M_1^{\sigma_1} \dots M_l^{\sigma_l}}_{c_{\sigma_1 \dots \sigma_l}} |\sigma_1 \dots \sigma_l\rangle. \quad (2)$$

The maximal dimension $m = \max_i \{m_i\}$ is called the *MPS bond dimension*. In typical one-dimensional calculations, $m = 200$, but for, e.g., 32×5 cylinders, $m > 5000$ is often

*c.hubig@physik.uni-muenchen.de

necessary. It is in these numerically demanding cases that our improvements are of particular relevance.

Similarly, a Hamiltonian operator can be written as a *matrix product operator* (MPO), where each tensor W_i is now of rank 4, namely (d_i, d_i, w_{i-1}, w_i) :

$$\hat{H} = \sum_{\substack{\sigma_1 \dots \sigma_l \\ \tau_1 \dots \tau_l}} W_1^{\sigma_1 \tau_1} \dots W_l^{\sigma_l \tau_l} |\sigma_1 \dots \sigma_l\rangle \langle \tau_1 \dots \tau_l|. \quad (3)$$

$w = \max_i \{w_i\}$ is called the *MPO bond dimension*. We will usually assume that for most i , $m_i = m$ and $w_i = w$. In practice, this holds nearly everywhere except at the ends of the chain, where the m_i grow exponentially from 1 to m . The basis of M_i (W_i) of dimension m_{i-1} (w_{i-1}) is called the left-hand side (LHS) basis, whereas the basis of dimension m_i (w_i) is the right-hand side (RHS) basis of this tensor. For simplicity, m_i , d_i , and w_i can also refer to the specific basis (and not only its dimension) when unambiguous.

Instead of M_i , we will also write A_i (B_i) for a left (right) normalized MPS tensor:

$$\sum_{\sigma_i} A_i^{\sigma_i \dagger} A_i^{\sigma_i} = \mathbb{I}, \quad (4)$$

$$\sum_{\sigma_i} B_i^{\sigma_i} B_i^{\sigma_i \dagger} = \mathbb{I}. \quad (5)$$

If we then define the contractions

$$l_i = (A_1^{\sigma_1} \dots A_{i-1}^{\sigma_{i-1}} M_i^{\sigma_i}) \in (d_1, \dots, d_i, m_i), \quad (6)$$

$$r_i = (M_i^{\sigma_i} B_{i+1}^{\sigma_{i+1}} \dots B_l^{\sigma_l}) \in (m_{i-1}, d_i, \dots, d_l), \quad (7)$$

we can rewrite $|\Psi\rangle$ from (2) as

$$|\Psi\rangle = \sum_{\sigma_1 \dots \sigma_l} l_i r_{i+1} |\sigma_1 \dots \sigma_i\rangle \otimes |\sigma_{i+1} \dots \sigma_l\rangle. \quad (8)$$

That is, when only considering one specific bond $(i, i+1)$, the left and right MPS bases at this bond are built up from the states generated by the MPS tensor chains to the left and right of the bond. Individual elements of an MPS basis are therefore called “state.”

Furthermore, define $L_0 = 1$ and $L_i = L_{i-1} A_i^\dagger W_i A_i$ with summation over all possible indices. Similarly, $R_{l+1} = 1$ and $R_i = R_{i+1} B_i^\dagger W_i B_i$. With these contractions, it is possible to write

$$\langle \Psi | \hat{H} | \Psi \rangle = L_{i-1} M_i^\dagger W_i M_i R_{i+1} \quad (9)$$

for any $i \in [0, l]$.

DMRG then works by *sweeping* over the system multiple times. During each sweep, each site tensor M_i is sequentially *updated* once with each update consisting of one optimization step via, e.g., a sparse eigensolver and possibly one *enrichment* step during which the left or right MPS basis of M_i is changed in some way. Depending on the exact implementation, updates may work on one (single-site DMRG) or two sites (two-site DMRG) at a time. The enrichment step may be missing or implemented via density matrix perturbation or subspace expansion.

III. PERTURBATION STEP AND CENTER MATRIX WAVE FUNCTION FORMALISM (CWF)

A. Convergence problems of single-site DMRG

During single-site DMRG, only a single MPS tensor M_i on site i is optimized at once. Compared to two-site DMRG, the search space is reduced by a factor of $d \approx 2 \dots 5$, leading to a speed-up of at least $O(d)$ per iteration [8]. However, since the left and right bases of the tensors M_i are fixed and defined by the environment (l_{i-1} and r_{i+1}), this approach is likely to get stuck. While also occurring if there are no symmetries implemented on the level of the MPS, this issue is most easily visible if one considers $U(1)$ symmetries [4]: assume that all basis states to the right of the RHS bond of M_i transform as some quantum number s_z . If we now target a specific sector, e.g., $S_z = 0$ overall, then on the LHS of this bond (i.e., from the left edge up to and including M_i), all states must transform as $-s_z$. In this configuration, it is impossible for a local change of M_i to add a new state that transforms as, say, s'_z to its right basis states, as there would be no corresponding state $-s'_z$ to the right of that bond, rendering the addition of the state moot from the perspective of the local optimizer, as its norm will be zero identically. A concrete example of this issue is given in Sec. VII A.

DMRG is a variational approach on the state space available to MPS of a given bond dimension. As such, the algorithm must converge into either the global or a local minimum of the energy in this state space. Hence, we will call all cases where DMRG converges on an energy substantially higher than the minimal energy achievable with the allowed MPS bond dimension cases where DMRG is stuck in *local minima*.

B. Density matrix perturbation

This convergence problem has been solved by White [8]. In the following, we will assume a left-to-right sweep; sweeping in the other direction works similarly, but on the left rather than right bonds. After the local optimization of the tensor M_i , the reduced density matrix

$$\rho_{i,R} = l_{i-1} M_i M_i^\dagger l_{i-1}^\dagger \quad (10)$$

is built on the next bond. This is the reduced density matrix resulting from tracing out the part of the system to the left of bond $(i, i+1)$.

$\rho_{i,R}$ is then perturbed as

$$\rho_{i,R} \rightarrow \rho'_{i,R} = \rho_{i,R} + \alpha \text{Tr}(L_i \rho_{i,R} L_i^\dagger). \quad (11)$$

The new $\rho'_{i,R}$ is then used to decide on a new set of basis states on the RHS of M_i , with the inverse mapping from the new to the old basis being multiplied into each component of B_{i+1} . The mixing factor α is a small scalar used to control the perturbation. A new scheme to find the optimal choice of α is discussed in Sec. VI.

C. Center matrix wave function formalism (CWF)

In a standard single-site DMRG calculation, the reduced density matrix $\rho_{i,R}$ is never used. More importantly, even building $\rho_{i,R}$ on a given bond $(i, i+1)$ will not yield a density matrix that can be used in (11), as it only contains the m_i states

existing on that bond already without knowledge of the m_{i-1} states on the bond one step to the left. In other words, it is not possible to choose the optimal set \tilde{m}_i based only on m_i ; rather, one requires also d_i and m_{i-1} .

The center matrix wave function formalism [15] was developed to cope with this problem. Given a site tensor $M_i \in (d_i, m_{i-1}, m_i)$ on a left-to-right sweep, it introduces a ‘‘center matrix’’ $C_{i,R} \in (d_i m_{i-1}, m_i)$ and replaces the original site tensor as

$$M_i \rightarrow A_i \in (d_i, m_{i-1}, d_i m_{i-1}) \text{ so that } M_i = A_i C_{i,R}. \quad (12)$$

A_i is constructed to be left orthogonal and is essentially an identity matrix mapping the left basis m_{i-1} and the physical basis d_i onto a complete basis containing all $d_i m_{i-1}$ states on its right. The new basis is ‘‘complete’’ in the sense that all states reachable from the left bond basis m_{i-1} and the local physical basis d_i are contained within it.

The contents of M_i are placed in $C_{i,R}$ accordingly and the original state remains unchanged. The reduced density matrix is then $\rho_{i,R} = C_{i,R} C_{i,R}^\dagger$ and has access to all $d_i m_{i-1}$ states, as required above. A perturbation of $\rho_{i,R}$ according to (11) hence allows the introduction of new states.

The DMRG optimization step can work on $C_{i,R}$ alone, with L_i built prior to optimization of $C_{i,R}$ from the expanded A_i . During each eigensolver step, the effective Hamiltonian on site i has to be applied onto $C_{i,R}$. The application is done by contraction of $L_i \in (w, d_i m_{i-1}, d_i m_{i-1})$, $R_{i+1} \in (w, m_i, m_i)$, and $C_{i,R} \in (d_i m_{i-1}, m_i)$ at cost $O(w(d^2 + d)m^3)$ per step. After optimization, the perturbation is added. Its computational cost is dominated by the calculation of $\alpha \text{Tr}\{L_i \rho_{i,R} L_i^\dagger\}$ at $O(wd^3 m^3)$. The bond between A_i and $C_{i,R}$ can then be truncated down to m using $\rho'_{i,R}$ and the remaining parts of $C_{i,R}$ are multiplied into B_{i+1} to the right.

The resulting algorithm converges quickly for one-dimensional problems and performs reasonably well for small cylinders. However, both the cost of the applications of \hat{H} to $|\Psi\rangle$ as $O(w(d^2 + d)m^3)$ as well as the large density matrix $\rho \in (dm, dm)$ cause problems if m and w become large.

IV. SUBSPACE EXPANSION

The idea of using *subspace expansion* instead of density matrix perturbation originates [14,16] in the tensor train/numerical linear algebra community. There, a stringent proof was given regarding the convergence properties of this method when the local tensor Z_i of the residual

$$|Z\rangle \equiv \hat{H}|\Psi\rangle - E|\Psi\rangle = \sum_{\sigma_1 \dots \sigma_l} Z_1^{\sigma_1} \dots Z_l^{\sigma_l} |\sigma_1 \dots \sigma_l\rangle \quad (13)$$

is used as the expansion term. Here, we will only use the method of subspace expansion and substitute a numerically much more cheaply available expansion term.

The following section is divided into three parts as follows. First, we will explain the concept of subspace expansion acting on two neighboring MPS tensors M_i, M_{i+1} . Second, the expansion term employed in DMRG3S is introduced and motivated. Third, a simple example is described.

A. Subspace expansion with an arbitrary expansion term

In the following, we will describe subspace expansion of the RHS basis of the current working tensor, as it would occur during a left-to-right sweep.

Assume a state $|\Psi\rangle$ described by a set of tensors $\{A_1, \dots, A_{i-1}, M_i, B_{i+1}, \dots, B_l\}$. At the bond $(i, i+1)$, we can then decompose the state as a sum over left and right basis states as in Eq. (8).

Now we *expand* the tensor $M_i \in (d, m_{i-1}, m_i)$ by some expansion term $P_i \in (d, m_{i-1}, m_{p_i})$ for each individual physical index component:

$$M_i^{\sigma_i} \rightarrow \tilde{M}_i^{\sigma_i} = \begin{bmatrix} M_i^{\sigma_i} & P_i^{\sigma_i} \end{bmatrix}. \quad (14)$$

This effectively expands the RHS MPS basis of M_i from m_i to $m_i + m_{p_i}$. Similarly, expand the components of $B_{i+1} \in (d, m_i, m_{i+1})$ with zeros:

$$B_{i+1}^{\sigma_{i+1}} \rightarrow \tilde{B}_{i+1}^{\sigma_{i+1}} = \begin{bmatrix} B_{i+1}^{\sigma_{i+1}} \\ 0 \end{bmatrix}. \quad (15)$$

The appropriately sized block of zeros only multiplies with the expansion term $P_i^{\sigma_i}$. In terms of a decomposition as in (8), this is equivalent to

$$|\Psi\rangle = \sum_{\sigma_1, \dots, \sigma_l} [l_i \ p] \begin{bmatrix} r_{i+1} \\ 0 \end{bmatrix} |\sigma_1 \dots \sigma_i\rangle \otimes |\sigma_{i+1}, \dots, \sigma_l\rangle \quad (16)$$

where p is the result of multiplying l_{i-1} and P_i , with the zero in the second expression similarly resulting from the zero in B_{i+1} . While the state $|\Psi\rangle$ remains unchanged, the local optimizer on the new site B_{i+1} can now choose the initially zero components differently if so required: the necessary flexibility in the left/right basis states to escape local minima has been achieved without referring to the density matrix.

Note that while orthonormality of B_{i+1} is lost, we do not need it between the enrichment step on site i and the optimization step on site $i+1$. The orthonormality of M_i can be restored via singular value decomposition as usual. Furthermore, it is usually necessary to truncate the RHS basis of \tilde{M}_i down from $m_i + m_{p_i}$ to m immediately following the expansion: this preserves the most relevant states of the expansion term while avoiding an exponential explosion of bond dimensions.

When sweeping from right to left, the left rather than right MPS basis of the current working tensor is expanded, with the left tensor A_{i-1} being zero padded as opposed to the right tensor B_{i+1} :

$$M_i^{\sigma_i} \rightarrow \tilde{M}_i^{\sigma_i} = \begin{bmatrix} M_i^{\sigma_i} \\ P_i^{\sigma_i} \end{bmatrix}, \quad (17)$$

$$A_{i-1}^{\sigma_{i-1}} \rightarrow \tilde{A}_{i-1}^{\sigma_{i-1}} = \begin{bmatrix} A_{i-1}^{\sigma_{i-1}} & 0 \end{bmatrix}. \quad (18)$$

B. Expansion term

Using the exact residual as the expansion term is computationally expensive: the term $\hat{H}|\Psi\rangle$ can be updated locally and is mostly unproblematic, but the subtraction of $E|\Psi\rangle$ and subsequent reorthonormalization is costly and has to be done after each local optimization, as the current value of E changes. This exact calculation is hence only possible for $m \approx 100$, which is far too small to tackle difficult two-dimensional problems.

Instead, we propose the very cheaply available terms

$$P_i = \alpha L_{i-1} M_i W_i \in (d_i, m_{i-1}, w_i m_i) \quad (19)$$

to be used during left-to-right sweeps and $P_i = \alpha R_{i+1} M_i W_i$ for use during right-to-left sweeps with some scalar mixing factor α . In the regime where the exact residual can be computed, these terms work essentially equally well.

This expression for P_i can be heuristically motivated as follows: Eq. (19) is equivalent to the partial projection of $H|\Psi\rangle$ onto $|\Psi\rangle$ to the left of the current bond. Hence, in the ground state and ignoring numerical errors, the RHS basis of this P_i is identical to that of M_i . Truncation from $m_i + m_{p_i}$ to m_i is then possible without inducing errors.

Numerically, it seems possible to choose α arbitrarily large without hindering convergence or perturbing the state too much in simple (one-dimensional) problems. However, if the chosen maximal bond dimension m is insufficient to faithfully capture the ground state of the given system, α has to be taken to zero eventually to allow convergence. Otherwise, P_i will continuously add new states and disturb the result of the eigensolver, which is optimal at this specific value of m but not an eigenstate of \hat{H} yet.

The cost of a single subspace expansion is $O(wdm^3 + w^2d^2m^2)$ for the calculation of P_i , potentially $O(2dwm^2)$ for the addition to M_i and B_{i+1} , respectively, and $O(dw^2m^3 + d^2m^2)$ for the SVD of an (dm, wm) matrix formed from \tilde{M}_i . If we restrict the SVD to m singular values, then the resulting matrices will be of dimension (dm, m) , (m, m) , and (m, wm) , respectively. The first can be reformed into \tilde{A}_i at cost $O(dm^2)$ and the second and third multiplied into B_{i+1} at cost $O(m^3dw + m^3d)$. The total cost of this step is dominated by the cost of the SVD at $O(dw^2m^3)$, which is still cheaper than the calculation of the perturbation term in (11), not considering the other costs associated to using the density matrix for truncation.

C. Subspace expansion at the example of a $d = l = 2$ spin system

In the following, we will demonstrate and illustrate the method of subspace expansion at the simple example of a system of two spins with $S = \frac{1}{2}$ from $m = 1$ to $m = 2$ as it would occur during a left-to-right sweep.

Assume the Hamiltonian

$$H = S_x^1 S_x^2 + S_y^1 S_y^2 + S_z^1 S_z^2 \quad (20)$$

$$= \frac{1}{2} \{S_+^1 S_-^2 + S_-^1 S_+^2\} + S_z^1 S_z^2 \quad (21)$$

with MPO components

$$W_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} S_+ & \frac{1}{\sqrt{2}} S_- & S_z \end{bmatrix}, \quad (22)$$

$$W_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} S_- & \frac{1}{\sqrt{2}} S_+ & S_z \end{bmatrix}^T. \quad (23)$$

Let the initial state be an $m = 1$ MPS, described by components

$$A_1^\uparrow = [a], \quad A_1^\downarrow = [\sqrt{1-a^2}], \quad (24)$$

$$B_2^\uparrow = [b], \quad B_2^\downarrow = [\sqrt{1-b^2}], \quad (25)$$

where square brackets denote matrices in the MPS bond indices. Due to the standard normalization constraints, there are only two free scalar variables here, a and b .

Subspace expansion of A_1 is straightforward (keep in mind that $L_0 \equiv 1$ for convenience):

$$P_1^{\tau_1} = \sum_{\sigma_1} W_1^{\tau_1 \sigma_1} A_1^{\sigma_1}, \quad (26)$$

$$P_1^\uparrow = W_1^{\uparrow\uparrow} A_1^\uparrow + W_1^{\uparrow\downarrow} A_1^\downarrow \quad (27)$$

$$= \begin{bmatrix} \frac{\sqrt{1-a^2}}{\sqrt{2}} & 0 & a \end{bmatrix}, \quad (28)$$

$$P_1^\downarrow = W_1^{\downarrow\uparrow} A_1^\uparrow + W_1^{\downarrow\downarrow} A_1^\downarrow \quad (29)$$

$$= \begin{bmatrix} 0 & \frac{a}{\sqrt{2}} & -\sqrt{1-a^2} \end{bmatrix}, \quad (30)$$

resulting in A'_1 and B'_2 directly after the expansion:

$$A_1'^\uparrow = \begin{bmatrix} a & \frac{\sqrt{1-a^2}}{\sqrt{2}} & 0 & a \end{bmatrix}, \quad (31)$$

$$A_1'^\downarrow = \begin{bmatrix} \sqrt{1-a^2} & 0 & \frac{a}{\sqrt{2}} & -\sqrt{1-a^2} \end{bmatrix}, \quad (32)$$

$$B_2'^\uparrow = \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B_2'^\downarrow = \begin{bmatrix} \sqrt{1-b^2} \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (33)$$

Normalizing A'_1 via a singular-value decomposition as $A'_1 \rightarrow A''_1 S V^\dagger$ and multiplying $S V^\dagger B'_2 \rightarrow B''_2$ gives

$$A_1''^\uparrow = [1 \quad 0], \quad (34)$$

$$A_1''^\downarrow = [0 \quad 1], \quad (35)$$

$$S V^\dagger = \begin{bmatrix} a & \frac{\sqrt{1-a^2}}{\sqrt{2}} & 0 & a \\ \sqrt{1-a^2} & 0 & \frac{a}{\sqrt{2}} & -\sqrt{1-a^2} \end{bmatrix}, \quad (36)$$

$$B_2''^\uparrow = \begin{bmatrix} ab \\ \sqrt{1-a^2} b \end{bmatrix}, \quad (37)$$

$$B_2''^\downarrow = \begin{bmatrix} a\sqrt{1-b^2} \\ \sqrt{1-a^2}\sqrt{1-b^2} \end{bmatrix}. \quad (38)$$

As expected, the final state $|\Psi\rangle = \sum_{\sigma_1 \sigma_2} A_1''^{\sigma_1} B_2''^{\sigma_2}$ is still entirely unchanged, but there is now a one-to-one correspondence between the four entries of B_2'' and the coefficients $c_{\{\uparrow, \downarrow\}, \{\uparrow, \downarrow\}}$ in the computational basis, making the optimization towards $c_{ii} = 0, c_{i \neq j} = \frac{1}{\sqrt{2}}$ trivial.

V. STRICTLY SINGLE-SITE DMRG

We can now combine standard single-site DMRG (e.g., Ref. [4], p. 67) with the subspace expansion method as a way to enrich the local state space, leading to a strictly single-site DMRG implementation (DMRG3S) that works without referring to the density matrix at any point.

With the notation from Sec. II, the steps follow mostly standard single-site DMRG. In an outermost loop, the algorithm sweeps over the system from left-to-right and right-to-left until convergence is reached. Criteria for convergence are, e.g., diminishing changes in energy or an overlap close to 1 between the states at the ends of subsequent sweeps.

The inner loop sweeps over the system, iterating over and updating the tensors on each site sequentially. Each local update during a left-to-right sweep (right-to-left sweeps work analogously) consists of the following steps.

(1) Optimize the tensor M_i : use an eigensolver targeting the smallest eigenvalue to find a solution (M_i^*, λ^*) to the eigenvalue problem

$$L_{i-1} R_{i+1} W_i M_i = \lambda M_i. \quad (39)$$

λ^* is the new current energy estimate. This first step dominates the computational cost.

(2) Build αP_i according to (19) using M_i^* . Build an appropriately sized zero block 0_{i+1} after the dimensions of P_i are known.

(3) Subspace-expand $M_i^* \rightarrow \tilde{M}_i^*$ with αP_i and B_{i+1} with 0_{i+1} .

(4) Apply a SVD to \tilde{M}_i^* and truncate its right basis to m_i again, resulting in \tilde{A}_i^* .

(5) Multiply the remainder of the SVD (SV^\dagger) into $B_{i+1} \rightarrow \tilde{B}_{i+1}$.

(6) Build L_i from \tilde{A}_i^* , L_{i-1} , and W_i .

(7) Calculate a new energy value after truncation based on L_i , \tilde{B}_{i+1} , W_{i+1} , and R_{i+1} . Use this energy value and λ^* to adapt the current value of α (cf. Sec. VI).

(8) Continue on site $i + 1$.

Of these, steps (2) and (3) implement the actual subspace expansion, whereas all others are identical to standard single-site DMRG.

It is important to note that the only change from standard single-site DMRG is the addition of an enrichment step via subspace expansion. Therefore, this method does not interfere with, e.g., real-space parallelized DMRG [5,17], the use of non-Abelian symmetries [6,15], or multigrid methods [7].

To analyze the computational cost, we have to take special care to ensure optimal ordering of the multiplications during each eigensolver iteration in (39). The problem is to contract $L_{i-1} R_{i+1} W_i M_i$, with L_{i-1} and $R_{i+1} \in (w, m, m)$, $W_i \in (d, d, w, w)$, and $M_i \in (d, m, m)$. The optimal ordering is then $((L_{i-1} M_i) W_i) R_{i+1}$ as follows.

(1) Contract L_{i-1} and M_i over the left MPS bond at cost $O(mw \cdot m \cdot dm = m^3 wd)$.

(2) Multiply in W_i over the physical bond of M_i and the left MPO bond at cost $O(m^2 \cdot wd \cdot dw = m^2 d^2 w^2)$.

(3) Finally contract with R_{i+1} over the right MPO and MPS bonds at cost $O(md \cdot wm \cdot m = m^3 dw)$.

The total cost of this procedure to apply \hat{H} to $|\Psi\rangle$ is $O(2m^3 wd + d^2 m^2 w^2)$. Assuming large $d^2 w/m$ is small, this gives a speed-up in the eigensolver multiplications of $(d + 1)/2$ over the CWF approach, which takes $O(m^3 wd(d + 1))$.

In addition to this speed-up, the subspace expansion is considerably cheaper than the density matrix perturbation. Since the perturbation/truncation step can often take up to 30% of total computational time, improvements there also have a high impact. At the same time, the number of sweeps at large m

needed to converge does not seem to increase compared to the CWF approach (cf. Sec. VII) and sometimes even decreases.

VI. ADAPTIVE CHOICE OF MIXING FACTOR

Both density matrix perturbation and subspace expansion generally require some small mixing factor α to moderate the contributions of the perturbation terms. The optimal choice of this α depends on the number of states available and those required to represent the ground state, as well as the current speed of convergence. Too large values for α hinder convergence by destroying the improvements made by the local optimizer, whereas too small values lead to the calculation being stuck in local minima with vital states not added for the reasons given in Sec. III B. The correct choice of α hence affects calculations to a large degree, but is also difficult to estimate before the start of the calculation.

Figure 1 displays the individual steps within a single update from the energy perspective: let ΔE_O denote the gain in energy during the optimization step and let ΔE_T denote the subsequent rise in energy during the truncation following the enrichment step. $\Delta E_T \neq 0$ only occurs if some enrichment (either via density matrix perturbation or subspace expansion) has occurred; otherwise, there would be no need for any sort of truncation. We can hence control the approximate value of ΔE_T via α , which leads to a simple adaptive and computationally cheap algorithm.

If ΔE_T was very small or even negative (after changing the optimized state by expansion of its right basis) during the current update, we can increase α during the next update step on the next site. If, on the other hand, $|\Delta E_T| \approx |\Delta E_O|$, that is, if the error incurred during truncation nullified the gain in energy during the optimization step, we should reduce the value of α at the next iteration to avoid making this mistake again.

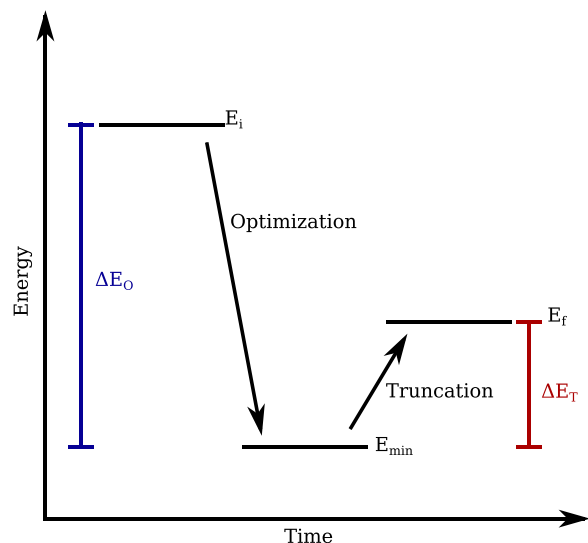


FIG. 1. (Color online) Energies of the state at different points during a single update: before optimization, the state has some initial energy E_i . Local optimization via the eigensolver takes this energy down by ΔE_O to E_{\min} . Subsequent truncation causes a rise in energy by ΔE_T with the final value at the end of this update being E_f .

In practice, it seems that keeping $\Delta E_T \approx -0.3\Delta E_O$ gives the fastest convergence. Given the order-of-magnitude nature of α , it is furthermore best to increase/decrease it via multiplication with some factor greater/smaller than 1 as opposed to adding or subtracting fixed values.

Some special cases for very small ΔE_O (stuck in a local minimum or converged to the ground state?) and $\Delta E_T > 0$ or $\Delta E_T < \Delta E_O$ have to be considered, mostly depending on the exact implementation.

It is unclear whether there is a causal relation between the optimal choice of α and the ratio of $\Delta E_T/\Delta E_O$ or whether both simply correlate with a preceding DMRG calculation: at the beginning, gains in energy are large and α is optimally chosen large, whereas later on, energy decreases more slowly and smaller values of α are more appropriate.

It is important to note that this is a tool to reach convergence more quickly. If one is primarily interested in a wave function representing the ground state, the calculation of a new α at each iteration comes at essentially zero cost. If, however, the aim is to extrapolate in the truncation error during the calculation, then a fixed value for α is of course absolutely necessary.

VII. NUMERICAL EXAMPLES

A. DMRG stuck in a local minimum

In this subsection, we will give a short example of how DMRG can get stuck in a local minimum even on a very small system. Consider $20S = \frac{1}{2}$ spins with isotropic antiferromagnetic interactions and open boundary conditions. The $U(1)$ symmetry of the system is exploited on the MPS basis, with the overall S_z forced to be zero. The initial state is constructed from 20 linearly independent states, all with three sites on the very right at $S_z = 0.5$ and $m = 20$ in total. The quantum number distribution at each bond is plotted in Fig. 2 as black circles.

DMRG3S is run with subspace expansion disabled, i.e., $\alpha = 0$, throughout the calculation. The algorithm “converges” to some high-energy state at $E^{\alpha=0} = -6.35479$. The resulting quantum number distribution (red squares in Fig. 2) shows clear asymmetry both between the left and right parts of the system and the $+S_z$ and $-S_z$ sectors at any given bond. It is also visible that while some states are removed by DMRG3S without enrichment, it cannot add new states: the red squares only occur together with the black filled circles from the input state.

If we enable enrichment via subspace expansion, i.e., take $\alpha \neq 0$, DMRG3S quickly converges to a much better ground state at $E^{\alpha \neq 0} = -8.6824724$. The quantum numbers are now evenly distributed between the left and right parts of the system and $\pm S_z$ symmetry is also restored.

B. Application to physical systems

In the following subsections, we will compare the two single-site DMRG algorithms CWF and DMRG3S when applied to four different physical systems: a $S = 1$ Heisenberg spin chain with periodic boundary conditions, a bosonic system with an optical lattice potential, a Fermi-Hubbard model at $U = 1$ and quarter-filling, and a system of free fermions at half-filling.

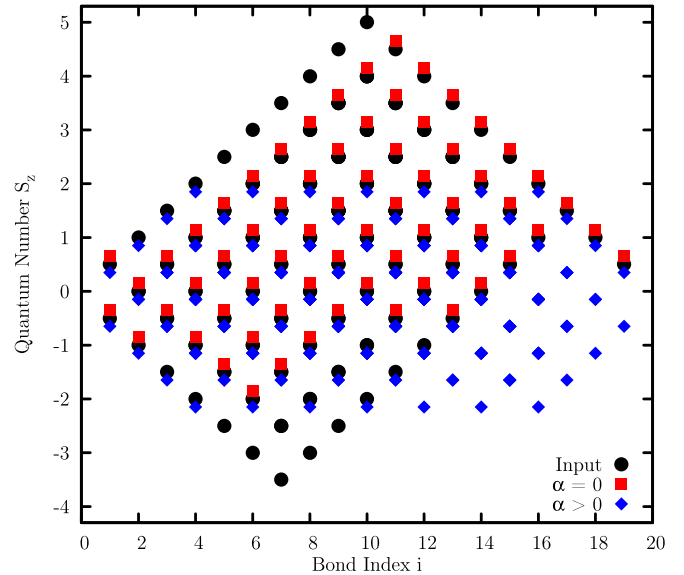


FIG. 2. (Color online) Quantum number distribution as counted from the right at each bond of a $l = 20$ system with $S = \frac{1}{2}$ and $S_z^{\text{total}} = 0$. The artificial input state is shown with black circles. Two DMRG calculations have then been done on this input state, once with no enrichment term ($\alpha = 0$, red squares) and once with subspace expansion enabled ($\alpha \neq 0$, blue diamonds). It is clearly visible that without enrichment, DMRG3S can reduce some weights to zero, but cannot add new states—red only occurs together with black. As soon as enrichment is enabled, DMRG3S restores $\pm S_z$ symmetry and reflective symmetry over the 10th bond and finds a much better ground state.

Each algorithm is run at three different values of $m = m_{\text{max}}, m_{\text{max}}/2, m_{\text{max}}/4$ from the same initial state and run to convergence. This way, it is possible to observe the behavior of the methods at both low and high accuracies.

The usual setup in DMRG calculations of starting at small m and increasing m slowly while the calculation progresses makes it unfortunately very difficult to compare between the three methods. This is because different methods require different configurations to converge optimally. We therefore restrict ourselves to fixed m throughout an entire calculation, even though all methods could be sped up further by increasing m slowly during the calculation.

Errors in energy compared to a numerically exact reference value E_0 are plotted as a function of sweeps and CPU time. It should be stressed that this error in energy is not directly comparable to the truncation error traditionally used in two-site DMRG or the variance $\langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2$ sometimes considered in single-site DMRG. Even small differences in energy can lead to vastly different physical states and reaching maximal accuracy in energy is crucial to ensure that the true ground state has been reached.

Furthermore, a traditional two-site DMRG (2DMRG) calculation without perturbations is done and its error in energy and runtime to convergence is compared to the two single-site algorithms. Here, *convergence* is defined as a normalized change in energy less than 10^{-9} (for $m = m_{\text{max}}$) (10^{-8}) (for $m < m_{\text{max}}$). The *runtime to convergence* is the CPU time used

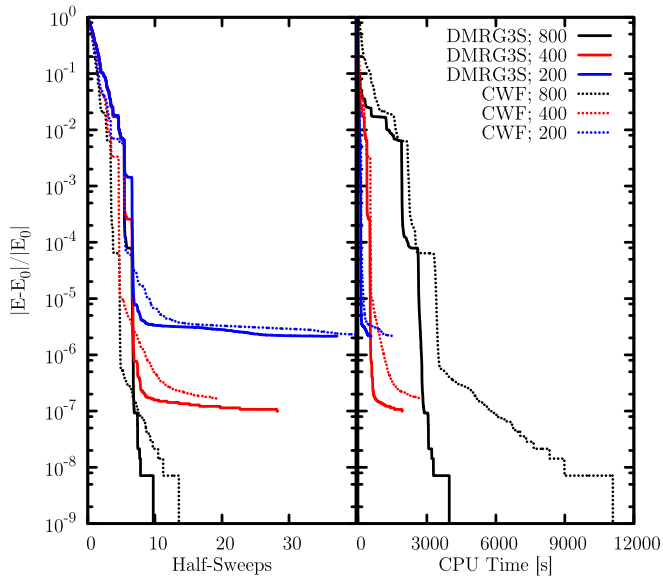


FIG. 3. (Color online) Spin chain Eq. (40): normalized error in energy as a function of sweeps (left) and CPU time used (right) of the two single-site algorithms at different $m = 200, 400, 800$. DMRG3S shows both a speed-up and an improved convergence per sweep compared to CWF, with a long tail of slow convergence very visible for CWF at high accuracies.

until that energy was output by the eigensolver for the first time.

All calculations were performed on a single core of a Xeon E5-2650.

1. $S = 1$ Heisenberg chain

First, we consider a $S = 1$ Heisenberg spin chain with $l = 100$ sites and periodic boundary conditions implemented on the level of the Hamiltonian as a simple link between the first and last site:

$$\hat{H} = \sum_{i=1}^{100} \hat{S}_i \cdot \hat{S}_{(i+1)\%100}. \quad (40)$$

$U(1)$ symmetries are exploited and the calculations are forced in the $S_z = 0$ sector.

This system is of particular interest as, first, it is one of the standard benchmarking systems with well-known analytic values for the ground-state energy. Second, it is a one-dimensional system where the case of periodic boundary conditions can still be tackled by DMRG. The larger MPO bond dimension resulting from these PBC similarly arises during the simulation of quasi-two-dimensional systems as cylinders. The same applies to the non-nearest-neighbor interactions in this system (between the first and last site) and cylindrical systems.

Figure 3 compares the error in energy with respect to the reference value $E_0 = -140.148404$ for DMRG3S and CWF for $m = 200, 400, 800$ as a function of sweeps and computation time.

During the first three to four sweeps, DMRG3S exhibits a smaller convergence rate per sweep; however, compared to the first sweeps of CWF, they also cost negligible CPU time. Afterwards, DMRG3S offers comparable (at medium

TABLE I. Spin chain Eq. (40): normalized error in energy at convergence and runtime to convergence of all three methods. DMRG3S is consistently faster than CWF, whereas the energies provided by 2DMRG are not comparable in accuracy.

	$m = 200$	$m = 400$	$m = 800$
DMRG3S energy error	2.1×10^{-6}	1.0×10^{-7}	7.1×10^{-9}
CWF energy error	2.8×10^{-6}	1.7×10^{-7}	7.1×10^{-9}
2DMRG energy error	1.1×10^{-5}	8.6×10^{-7}	1.0×10^{-7}
DMRG3S runtime	583 s	1935 s	3990 s
CWF runtime	1519 s	2695 s	11133 s
2DMRG runtime	762 s	3181 s	21963 s

accuracies) or much improved (at high accuracies) convergence rate per sweep as compared to CWF together with a still reduced average runtime per sweep. Combined, these effects lead to a speed-up of 2.6, 1.3, and 2.7 for $m = 200, 400$, and 800, respectively, between CWF and DMRG3S when considering the runtime to convergence.

In comparison, the 2DMRG algorithm does not handle the periodic boundary conditions well and yields energies higher than the single-site algorithms with perturbations (cf. Table I). Runtime to convergence is hence not comparable.

2. Dilute bosons on an optical lattice

We carry on to study bosons in a modulated potential of 10 unit cells, each with 16 sites. The cutoff for local occupation numbers is $n_{\max} = 5$, resulting in a local site dimension of $d = 6$. The Hamiltonian is given as

$$\hat{H} = + \sum_{i=1}^{160} \hat{n}_i \left\{ \cos^2 \left(2\pi \frac{i-0.5}{16} \right) + (\hat{n}_i - 1) \right\} - \sum_{i=1}^{159} \{ \hat{c}_i^\dagger \hat{c}_{i+1} + \text{H.c.} \}. \quad (41)$$

This system should be fairly easy for DMRG to handle, as there are only nearest-neighbor interactions. However, the large-scale order due to the modulated potential and a very small energy penalty paid for an uneven distribution of bosons was observed to cause badly converged results [7]. Manual checks of the states returned by each method were hence done to ensure a proper, equal distribution of bosons throughout the whole system.

The state is initialized with $n = 80$ bosons in total. We allow $m = 50, 100, 200$ states and use the energy reference value $E_0 = -103.646757$. All algorithms converge to this value at $m = 200$.

Figure 4 compares CWF and DMRG3S, whereas Table II additionally lists 2DMRG. Since the bond dimensions are relatively small, we do not expect a speed-up from faster numerical operations. Instead, the improved convergence behavior per sweep is responsible for the speed-up of 2 of DMRG3S over CWF at small m . At larger m , CWF converges better, but numerical operations also become cheaper for DMRG3S for a speed-up of 2 again.

As there are no long-range interactions, 2DMRG also fares well with regard to energy accuracy. However, it takes longer

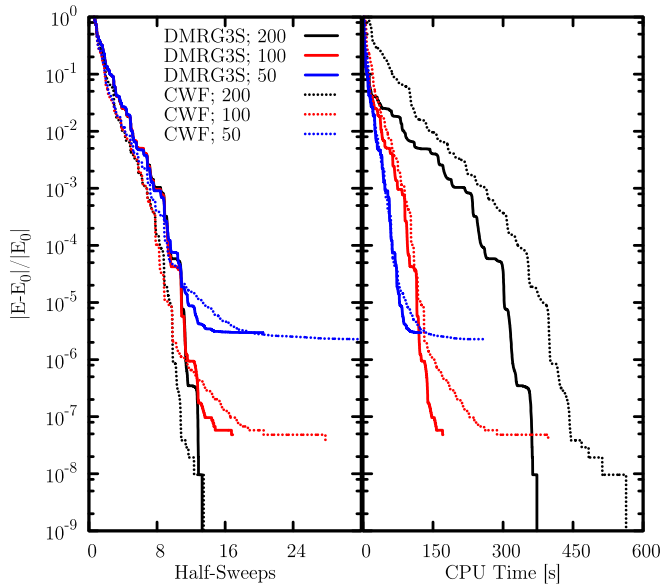


FIG. 4. (Color online) Bosonic system Eq. (41): normalized error in energy from CWF and DMRG3S as a function of sweeps (left) and CPU time used (right) for $m = 50, 100, 200$. Again, an improved convergence behavior at high accuracies can be observed, in particular at smaller values of m . The small bond dimensions lead to a smaller speed-up due to faster numerical operations, which only becomes visible at $m = 200$.

to converge than the single-site methods especially at large m , mainly because the eigenvalue problem in two-site DMRG is of dimension d larger than in single-site DMRG. A comparison between DMRG3S and 2DMRG leads to a speed-up of up to 3.3 for the case of $m = 200$.

3. Fermi-Hubbard model

As a third example, substantially more expensive calculations are carried out for a substantially stronger entangled Fermi-Hubbard model of 100 sites with Hamiltonian

$$\hat{H} = \sum_{i=1}^{100} \left\{ - \sum_{\sigma=\uparrow,\downarrow} [\hat{c}_{i,\sigma}^\dagger \hat{c}_{i+1,\sigma} + \text{H.c.}] + \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow} \right\}. \quad (42)$$

Both $U(1)_{\text{charge}}$ and $U(1)_{S_z}$ symmetries are employed, with 50 fermions and $S_z^{\text{total}} = 0$ enforced through the choice of initial state. Together with the free fermions from the next

TABLE II. Bosonic system Eq. (41): normalized error in energy at convergence and run time to convergence of all three methods. DMRG3S is again the fastest method with a very constant speed-up of 2 over CWF and up to 3.3 over 2DMRG.

	$m = 50$	$m = 100$	$m = 200$
DMRG3S energy error	2.9×10^{-6}	4.8×10^{-8}	$< 10^{-9}$
CWF energy error	2.3×10^{-6}	3.9×10^{-8}	$< 10^{-9}$
2DMRG energy error	1.9×10^{-6}	2.8×10^{-8}	$< 10^{-9}$
DMRG3S runtime	124 s	171 s	469 s
CWF runtime	260 s	397 s	951 s
2DMRG runtime	210 s	462 s	1550 s

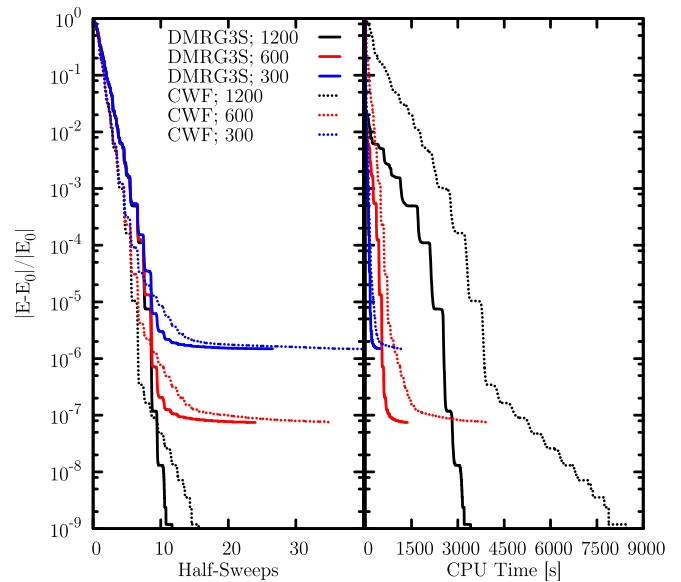


FIG. 5. (Color online) Fermi-Hubbard Eq. (42): normalized error in energy from DMRG3S and CWF as a function of sweeps (left) and CPU time used (right) for different bond dimensions $m = 300, 600, 1200$. The same basic behavior as for the previous systems is repeated, with both improved convergence behavior at high accuracies and faster numerical operations.

section, we can use this system to study how criticality and increased entanglement affect the three methods.

Calculations are done for $m = 300, 600, 1200$. All methods converge to the same value $E_0 = -84.255\,525\,4$ at large m .

Figure 5 compares the two single-site methods, while Table III summarizes all three DMRG implementations. Since the system only exhibits local interactions, 2DMRG fares well and all methods generally provide comparable energies. The difference is therefore in the runtime needed to achieve these energies. Compared to CWF, DMRG3S achieves a speed-up of ≈ 2.6 consistently at all m , as the smallest $m = 300$ is already large enough to justify the assumption $d^2 w \ll m$ in the speed-up of numerical operations. In particular, it continues to converge quickly at high accuracies, whereas CWF develops a long tail of slow convergence. The speed-up compared to 2DMRG is smaller at lower values of m , but increases to 3.9 at $m = 1200$.

TABLE III. Fermi-Hubbard Eq. (42): normalized error in energy at convergence and runtime to convergence of all three methods. Accuracies are comparable between the different methods, but runtimes vary greatly.

	$m = 300$	$m = 600$	$m = 1200$
DMRG3S energy error	1.5×10^{-6}	7.5×10^{-8}	$< 10^{-9}$
CWF energy error	1.5×10^{-6}	7.6×10^{-8}	$< 10^{-9}$
2DMRG energy error	1.3×10^{-6}	6.4×10^{-8}	$< 10^{-9}$
DMRG3S runtime	474 s	1367 s	3955 s
CWF runtime	1215 s	3917 s	10122 s
2DMRG runtime	727 s	2950 s	15596 s

TABLE IV. Free fermions Eq. (43): normalized error in energy at convergence and runtime to convergence of all three methods.

	$m = 300$	$m = 600$	$m = 1200$
DMRG3S energy error	5.0×10^{-6}	2.8×10^{-7}	$< 10^{-9}$
CWF energy error	3.8×10^{-6}	2.8×10^{-7}	$< 10^{-9}$
2DMRG energy error	3.7×10^{-6}	2.6×10^{-7}	$< 10^{-9}$
DMRG3S runtime	533 s	1452 s	4643 s
CWF runtime	863 s	2590 s	9586 s
2DMRG runtime	794 s	4584 s	29698 s

4. Free fermions

Finally, we consider a model of free fermions on a chain of 100 sites with Hamiltonian

$$\hat{H} = - \sum_{i=1}^{100} \sum_{\sigma=\uparrow,\downarrow} [\hat{c}_{i,\sigma}^\dagger \hat{c}_{i+1,\sigma} + \text{H.c.}] \quad (43)$$

The maximally delocalized wave function found in the ground state of this system is notoriously difficult for MPS formats in general to reproduce faithfully. At the same time, most other parameters are identical (d , l , m) or very close (w) to those in the Fermi-Hubbard model from Sec. VIIB3. The calculation is done using $U(1)_{\text{charge}}$ and $U(1)_{S_z}$ symmetries at half-filling with $N = 100$ fermions and $S_z^{\text{total}} = 0$. The choice of m is the same as for the Fermi-Hubbard system, namely $m = 300, 600, 1200$. We used $E_0 = -126.602376$ as the reference value, since all methods converged to this ground-state energy at $m = 1200$.

The results in Table IV and Fig. 6 mostly follow the previous results for locally interacting systems: accuracies of all methods are essentially identical, whereas time to convergence varies between the methods. At small m , there are some speed-ups of DMRG3S over CWF, largely due to better convergence behavior per sweep, whereas a significant advantage of DMRG3S becomes visible at larger m , when numerical operations become cheaper compared to the CWF method. Correspondingly, the speed-up from CWF to DMRG3S increases from 1.6 at $m = 300$ to 2 at $m = 1200$.

Similarly, the larger numerical cost of two-site DMRG becomes more noticeable at larger m , with the speed-up between 2DMRG and DMRG3S increasing from 1.5 at $m = 300$ to more than 6 at $m = 1200$.

Compared to the noncritical Fermi-Hubbard system from Sec. VIIB3, we observe larger errors in energy at fixed m , as expected. Correspondingly, as more eigenvalues contribute significantly, convergence of both the eigenvalue solver and the singular value decompositions becomes slower, leading to a slow-down of all three methods.

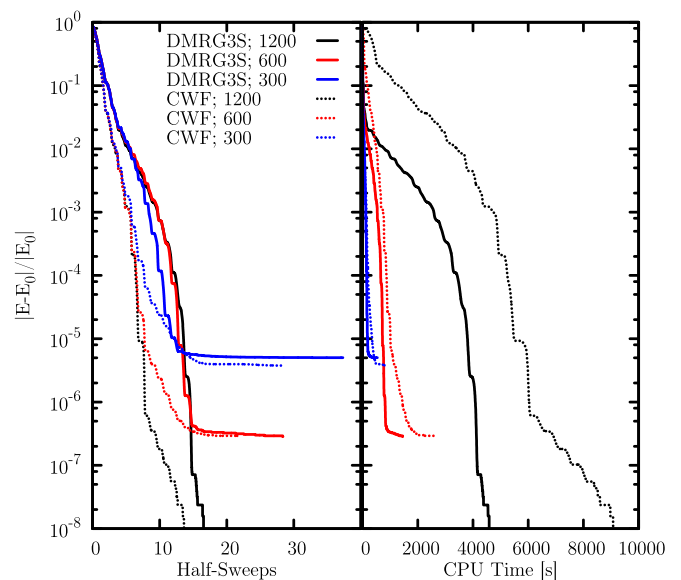


FIG. 6. (Color online) Free fermions Eq. (43): normalized error in energy from CWF and DMRG3S as a function of sweeps (left) and CPU time used (right) at $m = 300, 600, 1200$. CWF again exhibits a long tail of slow convergence, while DMRG3S converges quickly at all m and all accuracies.

VIII. CONCLUSIONS

The new strictly single-site DMRG (DMRG3S) algorithm results in a theoretical speed-up of $\sim (d+1)/2$ during the optimization steps compared to the center matrix wave function formalism (CWF), provided that d^2w/m is small. Further, convergence rates per sweep are improved in the important and computationally most expensive high-accuracy/large- m phase of the calculation. In addition, auxiliary calculations (enrichment, normalization, etc.) are sped up and memory requirements are relaxed.

Numerical experiments confirm a speed-up within the theoretical expectations compared to the CWF method. The efficiency of single-site DMRG in general compared to the traditional two-site DMRG was substantiated further by a large speed-up at comparable accuracies in energy.

ACKNOWLEDGMENTS

We would like to thank S. Dolgov, D. Savostyanov, and I. Kuprov for very helpful discussions. C.H. acknowledges funding through the ExQM graduate school and the Nanosystems Initiate Munich. F.A.W. acknowledges support by the research unit FOR 1807 of the DFG.

[1] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
 [2] S. R. White, *Phys. Rev. B* **48**, 10345 (1993).
 [3] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
 [4] U. Schollwöck, *Ann. Phys. (NY)* **326**, 96 (2011).
 [5] E. M. Stoudenmire and S. R. White, *Phys. Rev. B* **87**, 155137 (2013).

[6] I. P. McCulloch and M. Gulácsi, *Europhys. Lett.* **57**, 852 (2002).
 [7] M. Dolfi, B. Bauer, M. Troyer, and Z. Ristivojevic, *Phys. Rev. Lett.* **109**, 020604 (2012).
 [8] S. R. White, *Phys. Rev. B* **72**, 180403 (2005).
 [9] S. Yan, D. A. Huse, and S. R. White, *Science* **332**, 1173 (2011).

- [10] S. Depenbrock, I. P. McCulloch, and U. Schollwöck, *Phys. Rev. Lett.* **109**, 067201 (2012).
- [11] E. Stoudenmire and S. R. White, *Annu. Rev. Condens. Matter Phys.* **3**, 111 (2012).
- [12] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, *Phys. Rev. Lett.* **90**, 227902 (2003).
- [13] J. Eisert, M. Cramer, and M. B. Plenio, *Rev. Mod. Phys.* **82**, 277 (2010).
- [14] S. Dolgov and D. Savostyanov, *SIAM J. Sci. Comput.* **36**, A2248 (2014).
- [15] I. P. McCulloch, *J. Stat. Mech.* (2007) P10014.
- [16] S. V. Dolgov and D. V. Savostyanov, *Lect. Notes Comput. Sci. Eng.* **103**, 335 (2015).
- [17] S. Depenbrock, Ph.D. thesis, Ludwig-Maximilians-Universität München, 2013.