A Strictly Single-Site DMRG Algorithm with Subspace Expansion

C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf

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

Centre for Engineered Quantum Systems, School of Physical Sciences, The University of Queensland, Brisbane, Queensland 4072, Australia

(Dated: 23rd January 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 optimisation, similarly 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 speedup of $\approx (d + 1)/2$, with $d$ the physical site dimension. Further speedups result from cheaper auxiliary calculations and an often improved convergence rate per DMRG sweep. Runtime to convergence improves by up to a factor of 4 on the Fermi-Hubbard model. The method is compatible with real-space parallelisation and non-abelian symmetries.

I. INTRODUCTION

Since its introduction in 1993\cite{White1993,White1992} the Density Matrix Renormalisation Group method (DMRG) has seen tremendous use in the study of one-dimensional systems\cite{White1992,White1993}. Various improvements such as real-space parallelisation\cite{Schollwock2005}, the use of abelian and non-abelian symmetries\cite{McCulloch2007,Hubig2008}, multi-grid methods\cite{Schollwock2005}, and the switch from two-site DMRG as originally proposed to one-site DMRG with specific enrichment/perturbation steps acting on the reduced density matrix\cite{Kollath2005} have been proposed.

Nevertheless, despite some progress\cite{Hubig2008,McCulloch2007} (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: In one dimension, entanglement and hence matrix dimensions in DMRG are essentially size-independent for 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: In one dimension, entanglement and hence matrix dimensions in DMRG are essentially size-independent for 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.

As a result, the part of the Hilbert space considered by DMRG during its ground state search increases dramatically, resulting mainly in three problems: firstly, the DMRG algorithm becomes numerically more challenging as the size of matrices involved grows (we will assume matrix-matrix multiplications to scale as $O(m^3)$ throughout the paper). Secondly, the increased search space size makes it more likely to get stuck in local minima. Thirdly, while sequential updates work well in 1-D chains with short-range interactions, nearest-neighbour sites in the 2-D 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 a method developed in the tensor train/numerical linear algebra community to develop a strictly single-site DMRG algorithm that works without accessing the (full) reduced density matrix. Compared to the existing centermatrix wavefunction formalism (CWF)\cite{Hubig2008} we achieve a speedup of $\approx (d + 1)/2$ during each application of $\hat{H}$ to $|\Psi\rangle$ in the eigensolver during the central optimisation 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 Section V alongside a comparison with the existing CWF. As both the original perturbation method and the heuristic subspace expansion require a mixing factor $\alpha$ Section VI describes how to adaptively choose $\alpha$ for fastest convergence. Numerical comparisons and examples will be given in Section VII.

II. DMRG BASICS

The notation established here closely follows the review article Ref.\cite{McGuire2014} Consider a state $|\Psi\rangle$ of a system of $l$ spins. Each site has a physical state dimension $d_i$, e.g. $\forall i : d_i = 3$, $l = 50$ for a system of 50 spins $S = 1$ spins:

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

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

$$|\Psi\rangle = \sum_{\sigma_1 \ldots \sigma_l} M_1^{\sigma_1} \cdots M_l^{\sigma_l} |\sigma_1 \ldots \sigma_l\rangle . \quad (2)$$

The maximal dimension $m = \max_i \{ m_i \}$ is called the MPS bond dimension. In typical one-dimensional calcu-
we can rewrite multiple times. During each sweep, each site tensor for any possibly one optimisation step via e.g. a sparse eigensolver and updated sequentially. unambiguous.

If we then define the contractions 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) normalised MPS tensor:

\[ \sum_{\sigma_i} A_i^{\tau_1 \sigma_1} A_i^{\tau_2 \sigma_2} = \mathbb{I} \]

\[ \sum_{\sigma_i} B_i^{\sigma_i \tau_1} B_i^{\sigma_i \tau_2} = \mathbb{I} . \]

If we then define the contractions

\[ l_i = \left( A_i^{\sigma_1} \cdots A_i^{\sigma_{i-1}} M_i^{\sigma_i} \right) \in (d_1, \ldots, d_i, m_i) \]

\[ r_i = \left( M_i^{\sigma_i} B_i^{\sigma_{i+1}} \cdots B_i^{\tau_2} \right) \in (d_1, \ldots, d_i, m_i) \]

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

\[ |\Psi\rangle = \sum_{\sigma_1 \cdots \sigma_i} l_i |\sigma_1 \cdots \sigma_i\rangle \otimes |\sigma_{i+1} \cdots \sigma_l\rangle . \]

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

\[ \langle \Psi | \hat{H} | \Psi \rangle = L_{i-1} M_i^l W_i M_i R_{i+1} \]

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

DMRG then works by sweeping over the system multiple times. During each tensor $M_i$ is sequentially updated once with each update consisting of one optimisation 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 CenterMatrix Wavefunction Formalism (CWF)

#### A. Convergence Problems of Single-Site DMRG

During single-site DMRG, only a single MPS tensor $M_i$ on site $i$ is optimised at once. Compared to two-site DMRG, the search space is reduced by a factor of $d \approx 2 \cdots 5$, leading to a speedup of at least $O(d)$ per iteration. 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 that allow 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 optimiser, as its norm will be zero identically. A concrete example of this issue is given in Section VIIA.

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 (2005)\[ (3) \]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 optimisation of the tensor $M_i$, the reduced density matrix

\[ \rho_{i, R} = l_{i-1} M_i^l l_{i-1} \]

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 $(i, i+1)$.

\[ \rho_{i, R} \text{ is then perturbed as } \]

\[ \rho_{i, R} \rightarrow \rho'_{i, R} = \rho_{i, R} + \alpha \text{Tr} \left( L_i \rho_{i, R} L_i^l \right) . \]

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 $\alpha$ is a small scalar used to control the perturbation. A new scheme to find the optimal choice of $\alpha$ is discussed in Section VII.

C. Centermatrix Wavefunction 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 centermatrix wavefunction formalism 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 “centermatrix” $C_{i,R} \in (d_i m_{i-1}, m_i)$ and replaces the original site tensor as

$$M_i \rightarrow A_i \in (d_i, m_{i-1}, d_i m_{i-1}) \; \text{s.t.} \; M_i = A_i C_{i,R}.$$  \hspace{1cm} (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 optimisation step can work on $C_{i,R}$ alone, with $L_i$ built prior to optimisation 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 $O\left(w (d^2 + d) m^3\right)$ per step. After optimisation, 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 $O\left(w (d^3 m^3)\right)$. The bond between $A_i$ and $C_{i,R}$ can then be truncated down to $m$ using $\rho_{i,R}^\prime$ 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 $O\left(w (d^2 + d) m^3\right)$ as well as the large density matrix $\rho \in (d m, d m)$ cause problems if $m$ and $w$ become large.

IV. SUBSPACE EXPANSION

The idea of using subspace expansion instead of density matrix perturbation originates 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

$$|Z_i\rangle \equiv \hat{H}|\Psi\rangle - E|\Psi\rangle = \sum_{\sigma_1..\sigma_l} Z_i^{\sigma_1} \cdots Z_i^{\sigma_l} |\sigma_1..\sigma_l\rangle$$  \hspace{1cm} (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.

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 $\{A_1, \ldots, A_{i-1}, M_i, B_{i+1}, \ldots, B_l\}$. At the bond $(i, i+1)$, we can then decompose the state as a sum over left and right basis states as in Eq. (5). This decomposition is purely for illustrative reasons here and does not occur during the actual algorithm.

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:

$$M_i^{\sigma_i} \rightarrow \tilde{M}_i^{\sigma_i} = [M_i^{\sigma_i} P_i^{\sigma_i}] \; .$$  \hspace{1cm} (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:

$$B_{i+1}^{\sigma_{i+1}} \rightarrow \tilde{B}_{i+1}^{\sigma_{i+1}} = \begin{bmatrix} B_{i+1}^{\sigma_{i+1}} & 0 \end{bmatrix} \; .$$  \hspace{1cm} (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

$$|\Psi\rangle = \sum_{\sigma_1..\sigma_l} \begin{bmatrix} l_{i-1} & P_i \end{bmatrix} \begin{bmatrix} r_{i+1} & 0 \end{bmatrix} |\sigma_1..\sigma_l\rangle \otimes |\sigma_{i+1}..\sigma_l\rangle$$  \hspace{1cm} (16)

where $P$ is the result of multiplying $l_{i-1}$ and $P_i$, with the 0 in the second expression similarly resulting from the 0 in $B_{i+1}$. While the state $|\Psi\rangle$ remains unchanged, the local optimiser 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 optimisation 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 $M_i$ down from $m_i + n_P$ 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 $B_{i+1}$:

$$M_i^{\sigma_i} \rightarrow \tilde{M}_i^{\sigma_i} = \begin{bmatrix} M_i^{\sigma_i} \\ P_i^{\alpha_i} \end{bmatrix}$$

$$A_{i-1}^{\sigma_{i-1}} \rightarrow A_{i-1}^{\sigma_{i-1}} = \begin{bmatrix} A_{i-1}^{\sigma_{i-1}} & 0 \end{bmatrix}.$$  

B. Expansion Term

Using the exact residual as the expansion term is computationally expensive: The term $H |\Psi\rangle$ can be updated locally and is mostly unproblematic, but the subtraction of $E |\Psi\rangle$ and subsequent re-orthonormalisation is costly and has to be done after each local optimisation, 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

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

(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 $\alpha$. 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: (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 $m_i + n_P$ to $m_i$ is then possible without inducing errors.

Numerically, it seems possible to choose $\alpha$ arbitrarily large without hindering convergence or perturbing the state too much in simple (one-dimensional) problems. In more complicated problems, however, not taking $\alpha$ to zero eventually blocks the calculation from converging. This could have two reasons: Firstly, it is possible that in these calculations, $m$ is not large enough to faithfully represent the ground state. If the above understanding is correct, $P_i$ would then always add new states and disturb the result of the eigensolver which is optimal at this specific value of $m$ but not an eigenstate of $H$ yet. Secondly, if the above understanding is not correct, $P_i$ would (in general) be linearly independent of $M_i$ even in the ground state and therefore not allow error-free truncation.

The cost of a single subspace expansion is $O(dm^3 + w^2d^2m^2)$ for the calculation of $P_i$, potentially $O(2dwm^2)$ for the addition to $M_i$ and $B_{i+1}$ respectively and $O(wdm^3 + d^2m^2)$ for the SVD of an $(dm, w)$ matrix formed from $M_i$. If we restrict the SVD to $m$ singular values, then the resulting matrices will be of dimension $(dm, m)$, $(m, m)$ and $(m, w)$ respectively. The first can be reformed into $A_i$ 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 $O(dm^2m^3)$, 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

$$H = S_x^1S_x^2 + S_y^1S_y^2 + S_z^1S_z^2$$

$$= \frac{1}{2} \left( S_x^1S_x^2 - S_y^1S_y^2 + S_z^1S_z^2 \right)$$

(20)

(21)

with MPO-components

$$W_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & S_z \end{bmatrix}$$

$$W_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ S_z & S_z \end{bmatrix}^T.$$  

(22)

(23)

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

$$A_1^\dagger = [a] \quad A_1^\dagger = [\sqrt{1 - a^2}]$$

$$B_2^\dagger = [b] \quad B_2^\dagger = [\sqrt{1 - b^2}]$$

(24)

(25)

where square brackets denote matrices in the MPS bond indices. Due to the standard normalisation 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):

$$P_1^\dagger = \sum_{\sigma_1} W_1 \sigma_1 A_1^\dagger$$

$$P_1^\dagger = W_1^\dagger A_1^\dagger + W_1^\dagger A_1^\dagger$$

$$= \begin{bmatrix} \frac{\sqrt{1 - a^2}}{\sqrt{2}} & 0 & a \end{bmatrix}$$

(26)

(27)

(28)

$$P_1^\dagger = W_1^\dagger A_1^\dagger + W_1^\dagger A_1^\dagger$$

$$= \begin{bmatrix} 0 & a \sqrt{2} & -\sqrt{1 - a^2} \end{bmatrix}$$

(29)

(30)
resulting in $A'_1$ and $B'_2$ directly after the expansion:

$$A'_1 = \begin{bmatrix} a & \sqrt{1-a^2} & 0 & 0 \end{bmatrix} \, \lambda \, B'_{21}$$

$$B'_2 = \begin{bmatrix} b & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{1-b^2} & 0 & 0 & 0 \end{bmatrix}$$

Normalising $A'_1$ via a singular value decomposition as $A'_1 \to A''_1$ SV$^\dagger$ and multiplying $SV^\dagger B'_2 \to B''_2$ gives:

$$A'_{21} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$A''_{11} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$SV^\dagger = \begin{bmatrix} a/a & \sqrt{1-a^2} & 0 & 0 \\ \sqrt{1-a^2} & 0 & a/a & -\sqrt{1-a^2} \end{bmatrix}$$

$$B''_{21} = \begin{bmatrix} ab & 0 & 0 & 0 \\ \sqrt{1-a^2}b & 0 & 0 & 0 \end{bmatrix}$$

$$B''_{21} = \begin{bmatrix} a\sqrt{1-b^2} & 0 & 0 & 0 \\ \sqrt{1-a^2}b & 0 & 0 & 0 \end{bmatrix}$$

As expected, the final state $|\Psi\rangle = \sum_{\sigma_1} A'_{11} B''_{21}$ is still entirely unchanged, but there is now a one-to-one correspondence between the four entries of $B''_2$ and the coefficients $c_{\{\tau, \tau\}, \{\tau, \tau\}}$ in the computational basis, making the optimisation 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 Section [4] 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 consists of the following steps:

1. Optimise the tensor $M_i$: Use an eigensolver targeting the smallest eigenvalue to find a solution $(\hat{M}_i^*, \lambda^*)$ to the eigenvalue problem

$$L_{i-1}R_{i+1}W_i M_i = \lambda M_i$$

$\lambda^*$ is the new current energy estimate. This first step dominates the computational cost.

2. Build $\alpha P_i$ 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 $M_i^* \to \hat{M}_i^*$ with $\alpha P_i$ and $B_{i+1}$ with $0_{i+1}$.

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

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

6. Build $L_i$ from $A'_i, L_{i-1}$ and $W_i$.

7. Calculate a new energy value after truncation based on $L_i, B_{i+1}, W_i$ and $R_{i+1}$. Use this energy value and $\lambda$ to adapt the current value of $\alpha$ (cf. Section [VI]).

8. Continue on site $i + 1$.

Right-to-left sweeps work analogously.

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 parallelised DMRG [12] or the use of nonabelian symmetries [11] or multi-grid methods [2].

To analyse 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 $((L_{i-1}M_i)W_i)R_{i+1}$:

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

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 $O(md \cdot wm \cdot m = m^3 d w)$.

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 $m$, this gives a speedup in the eigensolver multiplications of $(d+1)/2$ over the CWF approach, which takes $O(m^3 wd + d^2 m^3 w)$.

In addition to this speedup, 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. Section [VII]) and sometimes even decreases.
VI. ADAPTIVE CHOICE OF MIXING FACTOR

Both density matrix perturbation and subspace expansion generally require some small mixing factor $\alpha$ to moderate the contributions of the perturbation terms. The optimal choice of this $\alpha$ 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 $\alpha$ hinder convergence by destroying the improvements made by the local optimiser, whereas too small values lead to the calculation being stuck in local minima with vital states not added for the reasons given in Section IIIIB. The correct choice of $\alpha$ hence affects calculations to a large degree, but is also difficult to estimate before the start of the calculation.

Fig. VI displays the individual steps within a single update from the energy perspective: Let $\Delta E_O$ denote the gain in energy during the optimisation step and let $\Delta E_T$ 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 $\Delta E_T$ via $\alpha$, which leads to a simple adaptive and computationally cheap algorithm:

If $\Delta E_T$ was very small or even negative (after changing the optimised state by expansion of its right basis) during the current update, we can increase $\alpha$ 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 optimisation step, we should reduce the value of $\alpha$ at the next iteration to avoid making this mistake again.

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 $\alpha$, 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 $\Delta E_O$ (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 $\alpha$ and the ratio of $\Delta E_T/\Delta E_O$ or whether both simply correlate with a proceeding DMRG calculation: at the beginning, gains in energy are large and $\alpha$ is optimally chosen large, whereas later on, energy decreases more slowly and smaller values of $\alpha$ 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 wavefunction representing the ground state, the calculation of a new $\alpha$ 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 $\alpha$ is of course absolutely necessary.

VII. NUMERICAL EXAMPLES

A. DMRG Stuck in a Local Minimum

In this sub-section, we will give a short example of how DMRG can get stuck in a local minimum even on a very small system. Consider 20 $S = \frac{1}{2}$ 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 3 sites on the very right at $S_3 = 0.5$ 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 $E^{\alpha\neq0} = -8.6824724$. The quantum
Heisenberg spin chain with squares) and once with subspace expansion enabled (circles. Two DMRG calculations have then been done on $S$ are forced in the $U$ site. Hamiltonian as a simple link between the first and last boundary conditions implemented on the level of the stored. right parts of the system and numbers are now evenly distributed between the left- and right parts of the system and $\pm S_z$ symmetry is also restored.

B. $S = 1, l = 100$ Heisenberg Chain

As a first comparative example, 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. $U(1)$ symmetries are exploited and the calculations are forced in the $S_z = 0$ sector. This example is identical to one of those used in 2005 by White. For reasons of simplicity, we will also use the same states configuration, namely four sweeps each at $m = 50$, 100, 200, 300, 400 and 600, even if this configuration may not be optimal for one or both of the algorithms. $\alpha$ is set to 1 initially and then adaptively corrected by both the CWF and DMRG3S. All calculations are done without any parallelisation on a single core of a Xeon E5-2650.

The error in energy between the result reported by the eigenvalue solver and the reference value $E_0 = -140.148 403 897$ (cf. Ref. [8]) is plotted as a function of the CPU time used and sweep number in Fig. 3. The minor differences in energies obtained are entirely due to numerical inaccuracies; if $\langle CWF_{\text{final}} \rangle$ and $\langle \text{DMRG3S}_{\text{final}} \rangle$ denote the two states as produced by the algorithms at the end of the 24th sweep, then $1 - \sqrt{\langle \text{CWF}_{\text{final}} \rangle \langle \text{DMRG3S}_{\text{final}} \rangle} = 8 \times 10^{-5}$; the two states should hence be considered identical.

At the same time, DMRG3S finishes after 1710 s, whereas the CWF algorithm requires 2598 s, resulting in a total speedup of $p \approx 1.5$.

Under the assumption that the term $d^2 w/m$ is small, the maximal speedup during eigensolver iterations is $(d + 1)/2 = 2$. For this system, $d = 3$ and $w = 4$ and hence $m \gg 36$ is necessary to achieve the $p_{\text{ideal}} = 2$ speedup. An analysis of the speedup during each of the six stages supports this: $p_{m=50} = 0.85, p_{m=100} = 1.08, p_{m=200} = 1.10, p_{m=300} = 1.36, p_{m=400} = 1.51, p_{m=600} = 1.61$. The deviation from the maximal value of 2 even for high values of $m$ mostly results from the other steps (enrichment, truncation etc.), which do not see the same speedup, in particular not at small values of $d$ and $w$.

If aiming for the lowest possible energy, the most conservative approach is usually to sweep as long as possible with the maximum bond dimensions from the beginning. We hence repeat calculations on the previous system, but allow each algorithm $m = 800$ states. Energy as a function of sweeps and CPU time used is shown in Fig. 4. Here, the advantage of $(d + 1)/2$ during eigensolver it-
Both algorithms converge to the same energy value \( E_0 = -140.148403 \). Due to the larger MPS bond dimension, the speedup is more clearly visible here than in Fig. 3.

C. Dilute Bosons on an Optical Lattice

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

\[
\hat{H} = -\sum_{i=1}^{159} \left\{ c_i^\dagger c_{i+1} + \text{h.c.} \right\} \\
+ \sum_{i=1}^{160} \left\{ \cos^2 \left( 2\pi 16 \frac{i-0.5}{10} \right) + n_i (n_i - 1) \right\} .
\] (40)

The state is initialised with \( n = 80 \) bosons in total. The calculations are run at \( m = 200 \) from the beginning. Both algorithms converge to the same energy value \( E = -103.646757 \), which is reproduced by calculations at \( m = 250 \). Fig. 5 gives the error in energy from this value as a function of sweeps and CPU time. Due to the relatively low number of states, the theoretical possible speedup is not achieved, but there is no slow-down either when comparing the two algorithms.

D. Fermi-Hubbard Model with 100 sites

As a last example, calculations are done for a Fermi-Hubbard model of 100 sites with periodic boundary conditions. The Hamiltonian for this model is given by

\[
\hat{H} = \sum_{i=1}^{100} \left\{ -\sum_{\sigma = \uparrow, \downarrow} \left[ c_i^\dagger c_{i+1, \sigma} + \text{h.c.} \right] + n_{i, \uparrow} n_{i, \downarrow} \right\} .
\] (41)

Both \( U(1)_{\text{charge}} \) and \( U(1)_{\text{spin}} \) symmetries are employed, with 50 fermions and \( S_z^{\text{total}} = 0 \) enforced through the choice of initial state. As the most conservative approach, we again allow \( m = 1200 \) states from the start.

Both algorithms converge to the energy value \( E = -84.2555254 \), albeit after vastly different times. Since \( \langle H^2 \rangle - \langle H \rangle^2 \approx 10^{-6} \) for this state, we will use that value as the reference value \( E_0 \). Two different effects can be observed from Fig. 6. Up until the error in energy is approximately \( 10^{-6} \), DMRG3S takes roughly four sweeps more than the CWF algorithm. The observed speedup \( p \approx 2.7 \) until then is therefore entirely due to faster multiplications and other auxiliary calculations, well in line with the prediction of \( (d+1)/2 = 2.5 \). Later, the convergence rate per sweep also improves, leading to an overall speedup of \( p \approx 4 \) for energy errors below \( 10^{-8} \).
VIII. CONCLUSIONS

The new strictly single-site DMRG (DMRG3S) algorithm results in a theoretical speedup of \((d+1)/2\) during the optimisation steps compared to the centermatrix wavefunction formalism. In addition, auxiliary calculations (enrichment, normalisation, etc.) are improved and memory requirements are relaxed. Convergence rates per sweep are mostly comparable to the centermatrix wavefunction formalism. Numerical experiments confirm the expected speedup.

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 Initiative Munich. F. A. Wolf acknowledges support by the research unit FOR 1807 of the DFG.