

# Generic construction of efficient matrix product operators

C. Hubig,<sup>1,\*</sup> I. P. McCulloch,<sup>2</sup> and U. Schollwöck<sup>1</sup><sup>1</sup>*Department of Physics and Arnold Sommerfeld Center for Theoretical Physics, Ludwig-Maximilians-Universität München, Theresienstrasse 37, 80333 München, Germany*<sup>2</sup>*Centre for Engineered Quantum Systems, School of Physical Sciences, The University of Queensland, Brisbane, Queensland 4072, Australia*  
(Received 8 November 2016; revised manuscript received 2 January 2017; published 18 January 2017)

Matrix product operators (MPOs) are at the heart of the second-generation density matrix renormalization group (DMRG) algorithm formulated in matrix product state language. We first summarize the widely known facts on MPO arithmetic and representations of single-site operators. Second, we introduce three compression methods (rescaled SVD, deparallelization, and delinearization) for MPOs and show that it is possible to construct efficient representations of arbitrary operators using MPO arithmetic and compression. As examples, we construct powers of a short-ranged spin-chain Hamiltonian, a complicated Hamiltonian of a two-dimensional system and, as proof of principle, the long-range four-body Hamiltonian from quantum chemistry.

DOI: [10.1103/PhysRevB.95.035129](https://doi.org/10.1103/PhysRevB.95.035129)

## I. INTRODUCTION

Since its introduction in 1992, the density matrix renormalization group [1] (DMRG) algorithm has been extremely successful at the solution of one-dimensional quantum mechanical problems [2]. Following the connection [3] between the original DMRG algorithm and the variational class of matrix product states (MPS), a series of second-generation DMRG algorithms has been developed [4–9], which explicitly build on the underlying tensor structure. In these second-generation algorithms, both the current variational state as well as the Hamiltonian operator are represented as tensor networks, namely MPS and matrix product operators (MPO).

As such, the correct construction of the MPO representation of the Hamiltonian at hand is the starting point of any DMRG calculation. This construction can be done fairly easily by hand for short-range Hamiltonians, if necessary with the help of a finite-state machine [6,10,11], which generates the required terms in the MPO. However, these finite-state machines can very quickly become extremely complicated (see, e.g., Figs. 7, 9, and 10 for automata to generate interactions on a two-dimensional cylinder in Ref. [12]). Other analytical approaches [13,14] to construct the MPO representation of in particular quantum chemistry Hamiltonians require individual treatment of each system and type of interaction by hand.

In this paper, we will present a generic method to construct arbitrary MPOs based solely on (a) the definition of appropriate single-site operators (such as  $c_i^\dagger$  or  $s_i^z$ ) and (b) the implementation of a model-independent MPO arithmetic. We will show that using these two ingredients, it is possible to efficiently construct the optimal representations of small powers of one-dimensional Hamiltonians and of medium-range Hamiltonians on two-dimensional cylinders. We further provide a proof-of-principle that the constructive approach is also able to generate the optimal representation for the four-body quantum chemistry Hamiltonian with long-range interactions.

The outline of the paper is as follows. In Sec. II, we define MPOs as widely used in the literature. Sections III

and IV summarize and supplement the existing works on the construction of fundamental single-site operators such as  $c_i^\dagger$  in MPO form as well as the addition and multiplication of arbitrary MPOs. After such an addition or multiplication, compression using one of the three compression methods specifically adapted to MPOs as laid out in Sec. V brings the operator representation back into its most efficient form. We give examples of the resulting MPOs in Sec. VII for a spin-chain with nearest-neighbor interactions, the Fermi-Hubbard model on a cylinder in hybrid real- and momentum space and the full quantum chemistry Hamiltonian. Section VIII details an algorithm to reduce numerical errors while calculating the variance  $\langle O^2 \rangle - \langle O \rangle^2$  of an MPO—of particular interest here is the Hamiltonian  $\hat{H}$  represented as an MPO. Finally, we conclude in Sec. IX.

## II. MATRIX PRODUCT OPERATORS (MPO)

For a detailed introduction to the DMRG and in particular the second-generation algorithms based on MPS and MPOs, we refer to an existing review [6] as well as a DMRG-centered overview of the implementation [15]. Here, we will only define the basic structure of matrix product operators.

Given a set of  $L$  local Hilbert spaces  $\mathcal{H}_i, i \in [1, L]$  of dimension  $d_i$  each and an operator  $\hat{H}$  which acts on the tensor product space  $\mathcal{H} = \otimes_i \mathcal{H}_i$ , we can write the operator  $\hat{H}$  as

$$\hat{H} = \sum_{\sigma\tau} c^{\sigma\tau} |\tau\rangle \langle \sigma|, \quad (1)$$

where  $|\tau\rangle$  enumerates the (product) basis states of  $\mathcal{H}$  and  $\langle \sigma|$  enumerates the basis states of the dual space of  $\mathcal{H}$ . We can decompose each basis vector  $|\tau\rangle$  as the tensor product of basis vectors on the individual local spaces as

$$|\tau\rangle = \bigotimes_{i=1}^L |\tau_i\rangle = |\tau_1 \dots \tau_L\rangle, \quad (2)$$

which leads to

$$\hat{H} = \sum_{\sigma_1 \tau_1} \dots \sum_{\sigma_L \tau_L} c_{\tau_1 \dots \tau_L}^{\sigma_1 \dots \sigma_L} |\tau_1 \dots \tau_L\rangle \langle \sigma_1 \dots \sigma_L|. \quad (3)$$

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

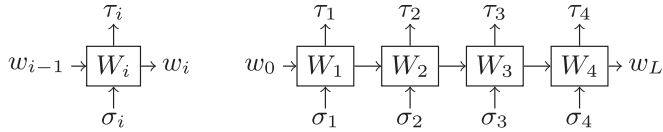


FIG. 1. (Left) Graphical representation of one component tensor  $W_i$  of an MPO. (Right) Contraction corresponding to the matrix-products in Eq. (4) of multiple component tensors. Labels on the legs denote the basis on this leg. Arrows indicate whether the leg is incoming or outgoing and are largely only relevant when implementing quantum number conservation. In this convention, MPOs act on matrix product states from above, with the latter having outgoing physical indices.

This form is still entirely generic [2]. The coefficient  $c \in \mathbb{C}^{\prod_i d_i^2}$  may now be decomposed as a set of *matrix products*. That is, on each site  $i$  and for every combination of local states  $\{|\tau_i\rangle, \langle\sigma_i|\}$ , we introduce a set of matrices  $(W_i^{\sigma_i \tau_i})_{w_{i-1}, w_i}$  with the property that their matrix-matrix product equals a specific element of the  $c$  tensor:

$$\sum_w (W_1^{\sigma_1 \tau_1})_{w_0, w_1} (W_2^{\sigma_2 \tau_2})_{w_1, w_2} \cdots (W_L^{\sigma_L \tau_L})_{w_{L-1}, w_L} = c_{\tau_1 \dots \tau_L}^{\sigma_1 \dots \sigma_L}. \quad (4)$$

The tensor  $W_i$  is then a rank-4 tensor with two *physical indices*  $\sigma_i$  and  $\tau_i$ , while the two matrix indices above are now called *MPO bond indices* and will be labeled  $w_{i-1}$  and  $w_i$ . In order for the above product of matrices to result in a scalar value for a given set of  $\tau$  and  $\sigma$ , we need  $w_0$  and  $w_L$  to be one-dimensional dummy indices. Each tensor  $W_i$  can be represented graphically by a square with four legs, cf. Fig. 1. Connecting legs of two tensors corresponds to a tensor contraction over the associated indices.

The relevant insight is that for a large class of operators, including all Hamiltonians with short-range interactions in one dimension, the required *MPO bond dimension*, i.e., the size of matrices  $W_i^{\sigma_i \tau_i}$  in Eq. (4), to reproduce the original tensor  $c$ , is both small ( $\approx 5$ ) and constant in the size  $L$  of the system. For long-range interactions, the size of the matrices usually only grows polynomially in the range of the interaction.<sup>1</sup> Constructing the set of tensors  $W_i$  which faithfully reproduce the desired operator  $\hat{H}$  at minimal MPO bond dimensions  $w_i$  will be discussed in this paper.

### III. CONSTRUCTION OF SINGLE-SITE OPERATORS

The representation of single-site operators as MPOs is relatively straightforward in general and mostly already widely known. In this section we summarize the existing, though not necessarily published, results in this area.

To construct the MPO representation of a single-site operator, we will first focus on a homogeneous  $S = 1$  spin chain. Section III A contains the transformation of fermionic operators using a Jordan-Wigner string. In Sec. III B, we explain how to handle nonhomogeneous systems, such as

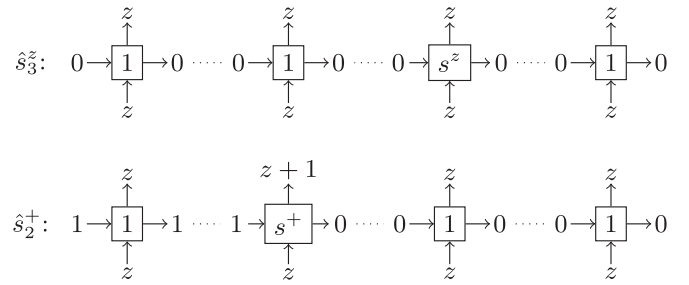


FIG. 2. Graphical representation of MPOs for  $\hat{s}_3^z$  and  $\hat{s}_2^+$  on a four-site system. Numbers and letters  $z$  denote the incoming and outgoing  $S^z$  quantum numbers on each tensor leg. By convention, the leftmost MPO bond index  $w_0$  transforms the same as the represented operator, while the rightmost MPO bond index  $w_L$  always transforms as the vacuum of the system. The MPO acts on the MPS below it, mapping states with  $S^z = z$  to those with  $S^z = z + 1$  in the second example on the second site. Dotted lines indicate the contractions which would result in the full coefficient tensor  $c$  from Eq. (4).

chains of alternating  $S = 1$  spins with  $S = 1/2$  spins at the ends or mixed fermion-boson systems.

Let us start with the construction of  $\hat{s}_i^z$  for an  $S = 1$  spin chain. The first ingredient is the representation of  $\hat{s}^z$  as a matrix on a local Hilbert space. This is straightforwardly given as  $s^z = \text{diag}(1, 0, -1)$ . Secondly, we need the matrix representation  $1_3$  of the identity operator  $\hat{1}$  on this local Hilbert space.

For a given fixed  $i$ , the explicit form of the single-site operator as  $\hat{s}_i^z = \hat{1}_1 \otimes \hat{1}_2 \cdots \hat{1}_{i-1} \otimes \hat{s}_i^z \otimes \hat{1}_{i+1} \cdots \hat{1}_L$  (that is, the identity operator acting on sites 1 through  $i - 1$  and  $i + 1$  through  $L$ ) then corresponds closely to the MPO representation of  $\hat{s}_i^z$  as

$$W_{<i} = 1_3 \quad W_i = s^z \quad W_{>i} = 1_3, \quad (5)$$

where the MPO bond indices are 1-dimensional dummy indices and do not affect the shape of the tensors. The MPO representation of  $\hat{s}_3^z$  is graphically given in Fig. 2. For trivially transforming operators, such as  $\hat{s}^z$  or  $\hat{n}$ , which do not change the quantum numbers of the state, it is entirely sufficient to store the identity MPO component  $1_{d_i}$  and the local representation (e.g.,  $s^z$ ) of the operator in question (e.g.,  $s^z$ ) as rank-4 tensors of size  $(1, 1, d_i, d_i)$ . One can then construct the MPO representation on the fly.

Operators, which do change a quantum number, such as  $\hat{s}_i^+$  or  $\hat{c}_i^\dagger$ , are more complicated. Since each tensor has to locally preserve symmetries and hence quantum numbers, the additional quantum number must be carried from the active site  $i$  to the left edge of the system. In turn, the chain of identity operators to the left of the active site must allow for this quantum number on their MPO bond indices, while those on the right of the active site only carry the vacuum quantum numbers (cf. Fig. 2). Therefore it is necessary to store different identity operator tensor representations for the left and right halves of the system. It is advisable to simply always store left and right identities together with the active site tensor, as the memory requirements of these small tensors are negligible and there is no need for logic differentiating trivially transforming and nontrivially transforming operators.

<sup>1</sup>Some specific long-range interaction terms can be encoded efficiently in constant space, but this is not generally the case.

Note that the case where no quantum numbers are used (either because they are not preserved by the system or not supported by the implementation) is identical to each operator and state transforming trivially and each leg only carrying a single, appropriately sized vacuum sector. In this case, the left and right identity operator tensor representations are again identical.

### A. Fermionic operators

The implementation of anticommutation relations for fermionic operators can also occur at the level of MPO representations of single-site operators [16]. Proper anti-commutation *within* the local state space of a single site,  $c_{\uparrow,i}^\dagger c_{\downarrow,i}^\dagger = -c_{\downarrow,i}^\dagger c_{\uparrow,i}^\dagger$ , is contained in the correct definition of the local site tensor.

Nonlocal anticommutation between operators on different sites requires a defined ordering of all fermionic operators. There is a natural ordering of operators along the MPO chain from the left to the right. It then suffices to replace the identities in the previous section either to the left or to the right of the active site by parity operators, which give a phase of  $-1$  if there is an odd number of fermions on the respective sites.

As an example, consider a product state  $|\psi\rangle = \prod_{i=1}^L [(c_{i\uparrow}^\dagger)^{n_{i\uparrow}} (c_{i\downarrow}^\dagger)^{n_{i\downarrow}}] |\text{vacuum}\rangle$ . An operator  $c_{\downarrow,j}^\dagger$  applied to  $|\psi\rangle$  has to be commuted past all operators with  $i < j$ . For each  $n_{i\sigma} \neq 0$ , it picks up a minus sign. Each of these signs can be implemented as the application of the local parity operator  $\hat{p}_i = (-1)^{n_i}$ . The MPO is then constructed as a chain of parity tensors  $p$ , the active site tensor  $c_{\uparrow}^\dagger$  and then a chain of right MPO identity components  $1_{d_i}$ , graphically represented in Fig. 3. Constructed in such a way, fermionic MPOs can be treated exactly the same as bosonic MPOs in all applications that follow.

### B. Nonhomogeneous systems

It is possible to simulate nonhomogeneous systems using MPS and MPO. Such a nonhomogeneity could be different spin sizes in a spin chain or the presence of both fermionic and bosonic sites in the system (the case of nonhomogeneous hopping between otherwise identical sites will be handled later in Sec. IV). The former case of nonhomogeneity can be used to represent some experimental systems with alternating  $S = 1$  and  $1/2$  spins as well as reduce finite-size effects in  $S = 1$  spin chains by placing  $S = 1/2$  spins at the two edges. The latter case might be helpful in simulating physical systems with bosonic and fermionic species, as they commonly occur in experiments with ultracold atoms.

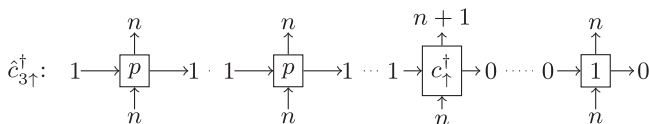


FIG. 3. Graphical representation of the fermionic creation operator  $c_{3\uparrow}^\dagger$  on a four-site system. Labels correspond to the fermionic particle quantum numbers.

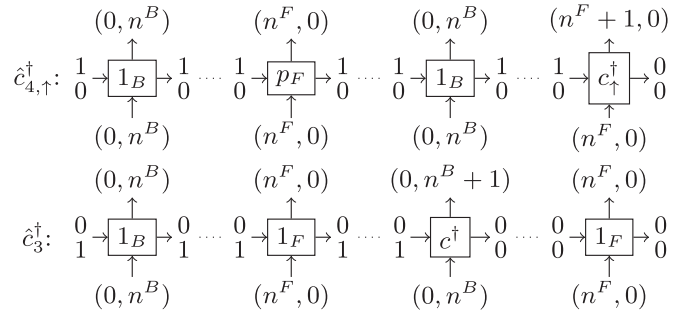


FIG. 4. Graphical representation of  $\hat{c}_{4\uparrow}^\dagger$  and  $\hat{c}_{3\uparrow}^\dagger$  in a nonhomogeneous system. Sites 1 and 3 may contain bosons, while sites 2 and 4 may contain up to four fermions. The creation operator  $c_{i(\sigma)}^\dagger$  is taken to create either a fermion or boson, depending on the type of the site  $i$ . The parity tensor  $p_F = (-1)^{n^F}$  has been used on the second site in place of the usual identity to implement fermionic anticommutation rules for  $c_{4\uparrow}^\dagger$ . Labels denote the  $N^F$  and  $N^B$  quantum numbers on the corresponding indices.

Suppose we have two types of sites in our system. Even sites may contain zero, one or two fermions, while odd sites may contain up to a certain number of bosons.

If we then wish to construct the fermionic creation operator  $c_{2i\uparrow}^\dagger$ , we have to ensure that the identities used to its left and right match the corresponding physical basis on those sites. Further, if we use  $U(1)_{N^F} \times U(1)_{N^B}$  quantum numbers for fermion and boson number conservation, the identities to the left need MPO bond indices transforming as  $N^F = 1$  and  $N^B = 0$ . In contrast, if we apply a bosonic creation operator  $c_{2i+1}^\dagger$ , the bond indices of those identities have to transform as  $N^F = 0, N^B = 1$ .<sup>2</sup> Figure 4 gives examples of those creation operators.

This has two implications. First, for every type  $t'$  of sites in the system, we need to define an appropriate active tensor representing (say)  $c_{\uparrow}^\dagger$  acting on a site of this type  $t'$ . Second, for every active site type  $t'$  on which the operator acts, we also need to store an appropriate left and right identity tensors for all types of sites.

Thus, if we have  $T$  different types of sites in our system, we need to store up to  $T + 2T^2$  rank-4 tensors per single-site operator. However, since these tensors are still only of size  $(1, 1, d_i, d_i)$ , and the number of different types  $T$  is typically also small, this is not a concern in practice.

Consider the example of a spin chain with  $S = 1$  spins in the bulk and two  $S = 1/2$  spins at the boundaries. To construct  $s_i^+$  on the fly, we need to store ten rank-4 tensors: first, we need to store two tensors representing  $s_i^+$  acting on sites with  $S = 1$  and  $1/2$ . Second, for each of these two, we need to store two left-identities which we place on sites with  $S = 1$  and  $S = 1/2$ , respectively, to the left of site  $i$ . Similarly, we need a total of four right-identities to be placed on sites to the right of site  $i$  with  $S = 1$  and  $1/2$ , respectively, for a total of ten tensors of

<sup>2</sup>In this specific case, it might be reasonable to set  $c_{2i+1}^\dagger$  equal to zero and call the bosonic creators  $a^\dagger$ . For spin systems, it is, however, entirely reasonable to have  $\hat{s}^z$  both on  $S = 1/2$  sites (at the edge) and  $S = 1$  sites (in the bulk) of the system.

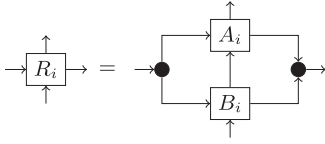


FIG. 5. Product of two MPOs for the tensors on a single site  $i$ . The product is built the same way on all sites  $i \in [1, L]$  of the system. Matching physical indices are contracted and the two left and right MPO bond indices merged into one on each side.

size  $(1, 1, 2S + 1, 2S + 1)$ ; in this specific case, requiring the storage of 55 scalar values in total.<sup>3</sup>

#### IV. ARITHMETIC OPERATIONS WITH MATRIX PRODUCT OPERATORS

The implementation of arithmetic operations with MPOs is well-known already [6,17] and is entirely independent of the specific form of the operands. In particular, the implementation can handle single-site operators as constructed in the previous section and MPOs resulting from earlier arithmetic operations on equal footing.

##### A. Products of matrix product operators

Given two operators  $\hat{A}$ ,  $\hat{B}$  and their MPO representation tensors  $\{A_i\}$  and  $\{B_i\}$ , the product  $\hat{R} = \hat{A}\hat{B}$  (read from right to left,  $\hat{B}$  is applied first) can be built on each site individually. It is graphically represented in Fig. 5. The lower physical index of each  $A_i$  is contracted with the upper physical index of the corresponding  $B_i$ . The left and right MPO indices of the tensors are merged into one fat index. This procedure results in an MPO with bond dimensions  $w'_i = w_i^a \cdot w_i^b$ . Specifically, the product of two single-site operators (MPO bond dimension 1) is again an MPO with bond dimension 1. The scalar products of operators occurring during the implementation of non-Abelian symmetries in tensor networks can similarly be implemented independently of the operator at hand.

##### B. Sums of matrix product operators

The sum of two operators  $\hat{A} + \hat{B} = \hat{R}$ , represented by MPO components  $\{A_i\}$ ,  $\{B_i\}$ , and  $\{R_i\}$  can also be constructed. Considering only the MPO bond indices, i.e., treating  $\{A_i\}$  as *matrices* of operators, the components of the resulting MPO are built as follows:

$$R_1 = (A_1 \ B_1), \quad (6)$$

$$R_{1 < i < L} = \begin{pmatrix} A_i & 0 \\ 0 & B_i \end{pmatrix}, \quad (7)$$

$$R_L = \begin{pmatrix} A_L \\ B_L \end{pmatrix}. \quad (8)$$

<sup>3</sup>Five of the tensors are of size  $(1, 1, 2, 2)$  for  $S = 1/2$  sites, requiring four numbers each. The other five are of size  $(1, 1, 3, 3)$  for  $S = 1$  sites with nine scalar entries. We hence need to store  $5 \times 4 + 5 \times 9$  scalar values to represent  $s_i^+$  on any site  $i$ .

For example, for a  $L = 3$  MPO, it is easy to verify that this results in the desired form representing  $\hat{A} + \hat{B}$ . The sum of two MPOs has a bond dimension  $w'_i = w_i^a + w_i^b$ .

#### V. MATRIX PRODUCT OPERATOR COMPRESSION

When constructing the MPO representation of a single-site operator as described in Sec. III, the resulting operator will have bond dimension 1 and will be in its most efficient representation. Products of such single-site operators (such as  $\hat{c}_i^\dagger \hat{c}_k^\dagger \hat{c}_l \hat{c}_j$ ) will keep the bond dimension at 1. However, the bond dimension will grow linearly in the number of such terms that are added together. Naively, a four-term interaction MPO representing  $\sum_{ijkl} \hat{c}_i^\dagger \hat{c}_k^\dagger \hat{c}_l \hat{c}_j$  will have maximal bond dimensions  $w_{L/2} = O(L^4)$ . The leading term in the computational cost of DMRG typically scales linearly in the maximal  $w_i$  and linearly in  $L$ , but there are subleading terms of quadratic order in  $w_i$ . Hence some way to avoid this quintic or even decic scaling is absolutely necessary.

*Compressing* an MPO will in general reduce its bond dimension to the bare minimum. For example, the sum of two identical MPOs will have a doubled bond dimension, which is obviously not necessary—a prefactor of 2 multiplied into the first tensor would correspond to the same operator. Similarly, two addends with long strings of identities to the left and right of the active sites, such as  $\hat{n}_i + \hat{n}_{i+1}$  can easily “share” these strings such that the most efficient MPO has bond dimension 1 everywhere but on bond  $(i, i + 1)$ , where 2 is the minimum required.

The compression methods presented here for MPOs are based on the same idea as those for MPS; for a given MPO, which has components  $W_i$  and  $W_{i+1}$  on sites  $i$  and  $i + 1$ , it is possible to rewrite

$$W_i \rightarrow W'_i := W_i p, \quad (9)$$

$$W'_{i;w_{i-1}w'_i} := \sum_{w_i} W_{i;w_{i-1}w_i} p_{w_i w'_i}; \quad (10)$$

$$W_{i+1} \rightarrow W'_{i+1} := p^{-1} W_{i+1}, \quad (11)$$

$$W'_{i+1;w'_i w_{i+1}} := \sum_{w_i} p^{-1}_{w'_i w_i} W_{i+1;w_i w_{i+1}} \quad (12)$$

without changing the MPO itself. For some MPO components, it is possible to find matrices  $p \in \mathbb{C}^{w_i \times w'_i}$  with  $w'_i < w_i$ . The new tensors  $W'$  then have a smaller bond dimension  $w'_i$  while representing the same original MPO, as only the matrix product of  $W_i$  and  $W_{i+1}$  or  $W'_i$  and  $W'_{i+1}$  is relevant for the operator. This is entirely analogous to the MPS, which also offer this gauge freedom and where it is also possible to use it in order to compress the size of the MPS.

It must be stressed that the compression methods presented here work iteratively on a bond-by-bond basis and cannot find a globally different (but better) MPO representation. However, for MPOs investigated here, we are still able to recover the optimal representation in most cases and a near-optimal representation even for extremely difficult problems. For the latter, it would be possible to combine the compression methods here with others, such as an iterative fitting method [10].



### A. Rescaling singular value decomposition

The singular value decomposition of MPOs has been proposed before [6,10], and in infinite-precision arithmetic, it would work exactly the same as for an MPS. Given a tensor  $W_{i;w_{i-1}w_i}^{\sigma_i\tau_i}$ , the indices  $w_{i-1}, \sigma_i$  and  $\tau_i$  are combined into a larger index  $\gamma$ , yielding the matrix  $M_{\gamma w_i}$ . This matrix is decomposed via SVD as  $M_{\gamma w_i} = U_{\gamma w'_i} \cdot S_{w'_i w_i} \cdot V_{w'_i w_i}$ . Columns of  $U_{\gamma w'_i}$  and rows of  $V_{w'_i w_i}$ , which correspond to negligible singular values in  $S_{w'_i w_i}$  are removed.  $U_{\gamma w'_i}$  is reshaped into the compressed tensor  $W'_{i;w_{i-1}w'_i}^{\sigma_i\tau_i}$ . The product  $S_{w'_i w_i} \cdot V_{w'_i w_i}$  acts as a transfer matrix and is multiplied into the next tensor on the right, compressing the dimension of the MPO on bond  $(i, i+1)$ . Sweeping left to right and right to left through the MPO compresses all bonds.

Unfortunately, a straightforward SVD yields extremely large singular values. The issue can be observed in Fig. 6 (labels “standard”). Given an uncompressed Fermi-Hubbard Hamiltonian with some finite-range interaction on systems of length  $L$  equal to 20, 40, and 80, we compress the left and right halves and then calculate the singular value spectrum in the center of the system. With increasing system size, we observe singular values growing as large as  $10^{26}$ . At the same time, the numerical noise, singular values normally discarded, grows as large as  $10^{12}$ !

The magnitude of the singular values is linked to the fact that while we can always rescale an MPS to have Frobenius norm 1, an operator will in principle have a system-size dependent Frobenius norm. Normalizing all tensors but one, as is common for SVD, implies that only this one tensor will carry the full norm of the operator.

This leads to two problems. First, there is difficulty in deciding which singular values should be kept, as even the

singular values strictly associated to numerical noise become extremely large. Second, compared to the normalized tensors to its left and right, the entries of the singular value tensor will have a grossly different order of magnitude, resulting in great precision loss during subsequent operations.

To avoid such large singular values, we can *rescale* the singular value tensor  $S$  by a scalar value. While this destroys the orthonormality of the resulting MPO bond basis, it preserves the orthogonality. Further, since all basis vectors are still of the same length, compression can proceed as usual, either based on a sharp cutoff or on a dynamic detection of the drop-off in magnitude of singular values (cf. Fig. 6). Lastly, properly chosen, such a rescaling can most often ensure that the norm of the operator is evenly distributed throughout its length, rather than concentrated in a single place.

In practice, we found it helpful to calculate the arithmetic average  $a_S$  of the singular values in the tensor  $S$  and rescale  $S \rightarrow \frac{1}{a_S} S$  such that this average is of order one. The tensor  $W'_i$  is multiplied with the inverse of the scaling factor to preserve the overall norm. To minimize numerical instabilities, it is advisable to choose the power of two closest to  $a_S$  as the scaling factor, since such multiplications are exact with IEEE-754 floating point numbers.

With this rescaling after each SVD during the compression sweeps, we observe singular values of magnitudes between 1 and 100 independent of the system length and numerical noise, clearly recognizable as such of magnitude  $10^{-14}$  or smaller. However, there are still caveats and counterindications against using SVD in specific cases, primarily concerning MPO representations of projectors or sums of operators involving projectors. First, when attempting to compress a suboptimal representation of a projector, SVD even with rescaling often struggles to properly distribute the norm throughout the system. For example, given the projector

$$\hat{P}_{\downarrow} = \prod_{i=1}^L \left( \frac{\hat{1}_i}{2} - \hat{s}_i^z \right) \quad (13)$$

on the  $S = 1/2$  Heisenberg chain of length  $L$ , an SVD compression will lead to exponentially large terms in the first and last tensor, with the (otherwise properly compressed) terms in the bulk all carrying a prefactor  $1/2$ .

Second, when attempting to evaluate sums of operators with greatly varying Frobenius norms, SVD will often entirely discard the smaller operator. This is not a concern for most Hamiltonians, as they are built from few-body interaction terms all with roughly the same order of magnitude. However, when evaluating  $\hat{1} - \hat{P}_{\downarrow}$  in the above system, the result from SVD is simply  $\hat{1}$ . This can be understood since  $\|\hat{1}\|_{\text{Frob}} = \sqrt{d^L} = 2^{L/2}$  while  $\|\hat{P}_{\downarrow}\|_{\text{Frob}} = 1$ . In a similar fashion, if SVD were tasked with the compression of the sum of two MPS, one of norm  $\sqrt{d^L}$  and the other of norm 1, the result would also simply be the larger of the two states, as soon as the difference in the two is lost in the numerical noise of order  $10^{-16}$ .

Both problems can be detected reliably. For the first, it is sufficient to compare, e.g., the norm of each MPO component: if one or two (i.e., at the edges of the system) is much greater than in the bulk, SVD failed to properly distribute the norm.

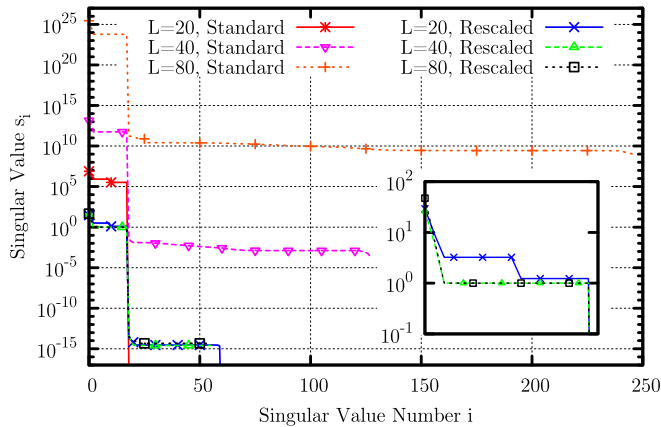


FIG. 6. Typical singular value distribution observed during the compression of a large Hamiltonian MPO with (rescaled) and without (standard) rescaling for different system sizes. There is a sharp drop in magnitude of the singular values between those relating to components of the operator and those made redundant by the SVD. Without rescaling, the overall magnitude of singular values grows exponentially with system size, leading to numerical errors. With rescaling, all relevant singular values have magnitude independent of the system size. (Inset) Zoom-in on the relevant first 18 singular values with rescaling, which are of roughly constant magnitude independent of the system size.

For the second, it is sufficient to compare the Frobenius norms of addends before operator addition, if in doubt. As a rule of thumb, the Frobenius norm is exponential in the number of identity MPO components. It is hence possible to sum few-body interaction terms together (as they typically occur in Hamiltonians or correlators) or *alternatively* sum “few-identity” terms (such as projectors) together. However, this rule only applies to sums of MPOs, not products of MPOs. Special care must be taken in those cases and it must be checked carefully whether the errors introduced by SVD are acceptable relative to the problem at hand.

### B. Deparallelization

The SVD method has the disadvantage that it destroys the extreme sparsity of the usual MPO tensors and relies on a robust and small window of singular values encountered in the MPO. Employing quantum number labels reduces this destruction of sparsity to the scope of individual blocks, which will often be implemented as dense tensors in any case. However, in particular, for simple homogenous operators, it is desirable to keep the sparse, natural structures of MPO. It is furthermore sometimes also necessary to compress MPOs with greatly varying singular values.

The much simpler deparallelization method avoids both issues entirely. Sparsity is largely conserved and the compression does not rely on singular values. It furthermore does not rescale most elements of the tensor, keeping the norm distributed in the same way as before. It was first presented in Ref. [4] and can be considered a slight generalization from the *fork-merge* method presented in Ref. [13], from “forking” and “merging” only identity operators to arbitrary strings of operators.

The algorithm is presented in detail in Appendix B. The basic idea is again to re-shape each site tensor  $W_{i;\sigma_i\tau_i}^{\sigma_i\tau_i}$  into a matrix  $M_{\gamma w_{i+1}}$ . Then, columns of  $M$ , which are entirely parallel to any previous column are removed, with the respective proportionality factor stored in the transfer matrix to be multiplied into the next site tensor.

This procedure results in a MPO that is often optimal for spatially homogeneous operators and retains the advantageous structure of analytically constructed MPO tensors. For more difficult Hamiltonians, it often results in suboptimal representations.

### VI. DELINEARIZATION

The delinearization method aims to combine the advantages of the SVD and the deparallelization. It is suitable to compress any MPO, including the previously mentioned sums of projectors and Hamiltonians as well as complicated Hamiltonians. In most cases, it results in an optimal MPO dimension. For extremely large MPOs, the resulting bond dimensions tend to be slightly larger than with SVD compression. However, the original sparsity of the MPO is largely preserved, even in the dense subblocks<sup>4</sup> labeled by quantum numbers. Wherever

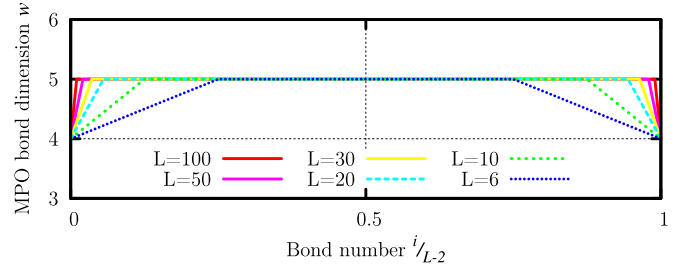


FIG. 7. MPO bond dimension of the representation of Eq. (14) over the length of the chain for different chain lengths, here at  $h = 1$ ,  $J_x = 1/2$ ,  $J_y = 1/3$ , and  $J_z = 1/5$  to illustrate the generic case. The leftmost and rightmost bonds have dimension four, whereas the bulk bond dimension is five as in the analytical solution.

possible, it attempts to ensure that no spurious small terms can occur in the Hamiltonian.

The algorithm is presented in full detail in Appendix C. Similar to the deparallelization, we attempt to remove columns from the  $M_{\gamma w_{i+1}}$  matrix, but now allow for linear combinations of previously kept columns to replace the column in question, whenever possible under the constraint that no cancellation to exactly zero can occur (this avoids the spurious small terms).

### VII. EXAMPLES

We will present three examples of MPO generation using the above construction method. First, we show that it works well for the simple example of nearest-neighbor interactions on spin chains and even for small powers of the Hamiltonian. Second, we explain that it is very easy to generate the Hamiltonian for the Fermi-Hubbard model on a cylinder in hybrid real- and momentum space. Third, we present data that the construction method also correctly sums up partial terms in a toy model for the full quantum chemistry Hamiltonian.

#### A. Spin chains with nearest-neighbour interactions

We consider the Hamiltonian with nearest-neighbor interactions on a spin chain:

$$\hat{H} = \sum_{i=1}^L h \hat{S}_i^z + \sum_{a=x,y,z} \sum_{i=1}^{L-1} J_a S_i^a S_{i+1}^a. \quad (14)$$

We can construct analytically an optimal representation of this Hamiltonian [6], which has MPO bond dimension 5. In comparison, we can plot the dimension of each bond of the numerically constructed representation for various system sizes (cf. Fig. 7). As is clearly visible, the bond dimension quickly saturates at five and stays constant  $O(1)$  independent of the system length. The algorithm even finds an improvement over the usual analytic solution, as only one  $\hat{S}^z$  term is necessary at the boundary. In the bulk, it completely reproduces the analytic

<sup>4</sup>When quantum numbers (both from Abelian and non-Abelian symmetries) are employed in tensor networks, it becomes possible to

split the original, dense tensor into smaller dense tensor blocks which transform uniquely under these quantum numbers. The size of these blocks, e.g., the number of states with total spin  $S = 0$ , is the most relevant scaling dimension.

TABLE I. Bond dimensions  $w_{L/2}$  in the center of an  $L = 100$  chain of powers of the nearest-neighbor spin-chain Hamiltonian (14) with SVD and delinearization compression. Relative sparsity of the resulting MPO is included for the delinearization method (SVD does not preserve sparsity at all). We compare with the results of Fröwis *et al.* [10] for the XXZ Hamiltonian constructed with an iterative fitting procedure, which could also be combined with our construction method for MPOs.

| Order $\hat{H}^n$     | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|
| SVD: $w_{L/2}$        | 5   | 9   | 16  | 32  | 51  | 64  | 92  |
| DLN: $w_{L/2}$        | 5   | 9   | 16  | 32  | 51  | 81  | 126 |
| DLN: Sparsity         | 81% | 84% | 82% | 89% | 88% | 88% | 85% |
| Fitting method:       | 5   | 9   | 16  | 32  | 51  | 79  | 110 |
| Ref. [10], $w_{\max}$ |     |     |     |     |     |     |     |

solution, here at the example of  $J_x = J_y = J_z = h = 1$ :

$$W_{\text{bulk}} = \begin{pmatrix} 1 & s^z & s^z & s^y & s^x \\ 0 & 1 & 0 & 0 & 0 \\ 0 & s^z & 0 & 0 & 0 \\ 0 & s^y & 0 & 0 & 0 \\ 0 & s^z & 0 & 0 & 0 \end{pmatrix}. \quad (15)$$

Further, we can construct powers of the Hamiltonian  $\hat{H}$ , here specifically with coefficients  $h = 1$ ,  $J_x = 1/2$ ,  $J_y = 1/3$ , and  $J_z = 1/5$ . The procedure is to first generate  $\hat{H}$  using only deparallelization, which leads to the near-analytic solution at bond dimension 5. We then multiply the MPO with itself to generate  $\hat{H}^2$  and compress the operator using SVD or Delinearization. Multiplying with  $\hat{H}$  repeatedly, we construct up to the seventh power of  $\hat{H}$  and compare the bond dimensions with those resulting from an iterative fitting procedure (cf. Table I).

For small powers  $n$ , the resulting bond dimensions from the three compression methods coincide. For higher powers, the SVD method results in somewhat lower bond dimensions. This could be both due to numerical inaccuracies in either method (e.g., erroneously discarding small but relevant singular values) or the fitting approach getting stuck in a local minimum. To numerical accuracy, the error resulting from the SVD compression is zero, however.

In comparison, the delinearization method encounters cyclic linear dependencies it cannot break when attempting to compress the higher-power MPO representations. This results in a larger bond dimension. However, the original sparsity of the MPO, calculated as the relative number of exactly zero entries in the dense sub-blocks of the MPO, is largely preserved at over 80% zero entries while no such entries were found after SVD compression.

## B. Fermi-Hubbard Hamiltonian on a cylinder in hybrid space

The Fermi-Hubbard model in two dimensions is a problem of ongoing research. When attempting a solution of two-dimensional problems with DMRG, the usual course of action is the mapping onto a cylinder [18]. This allows for periodic boundary conditions along the cylinder width, while keeping the ends of the cylinder open, as is advantageous for DMRG. With coordinates  $x$  along the length  $L$  of the cylinder and

coordinates  $y$  along its width  $W$ , we can then write the Hamiltonian as

$$\begin{aligned} \hat{H} = & - \sum_{y=1}^W \sum_{x=1}^L \hat{c}_{x,y}^\dagger \cdot \hat{c}_{x,y+1} + \text{H.c.} \\ & - \sum_{y=1}^W \sum_{x=1}^{L-1} \hat{c}_{x,y}^\dagger \cdot \hat{c}_{x+1,y} + \text{H.c.} \\ & + \frac{U}{2} \sum_{y=1}^W \sum_{x=1}^L (\hat{c}_{x,y}^\dagger \cdot \hat{c}_{x,y})^2 - \hat{c}_{x,y}^\dagger \cdot \hat{c}_{x,y}. \end{aligned} \quad (16)$$

The first two lines are the usual kinetic term containing nearest-neighbor hopping along the width and the length of the cylinder. The third term is the on-site interaction with coefficient  $U$ , already written in anticipation of the following Fourier transformation. The objects  $c^\dagger$  and  $c$  are  $\text{SU}(2)$ -invariant operators, the scalar product  $c^\dagger \cdot c$  could be expanded out to the usual form  $c_\uparrow^\dagger c_\uparrow + c_\downarrow^\dagger c_\downarrow$ . This model has explicit  $\text{SU}(2)_{\text{Spin}}$  symmetry with quantum number  $S$  and  $\text{U}(1)_{\text{Charge}}$  symmetry with quantum number  $N$ .

Following Ref. [12], we can perform a Fourier transformation along the width of the cylinder with the identities

$$\hat{c}_{x,y} = \frac{1}{\sqrt{W}} \sum_{k=1}^W e^{2\pi i \frac{k}{W} y} \hat{c}_{x,k}, \quad (17)$$

$$\hat{c}_{x,y}^\dagger = \frac{1}{\sqrt{W}} \sum_{k=1}^W e^{-2\pi i \frac{k}{W} y} \hat{c}_{x,k}^\dagger, \quad (18)$$

to achieve the form

$$\begin{aligned} \hat{H} = & - \sum_{x=1}^L \sum_{k=1}^W 2 \cos\left(2\pi \frac{k}{W}\right) \hat{c}_{x,k}^\dagger \cdot \hat{c}_{x,k} \\ & - \sum_{x=1}^{L-1} \sum_{k=1}^W \hat{c}_{x,k}^\dagger \cdot \hat{c}_{x+1,k} + \text{H.c.} \\ & + \frac{U}{2} \sum_{x=1}^L \sum_{k=1}^W \left[ \sum_{l=1}^W \frac{1}{W} (\hat{c}_{x,k}^\dagger \cdot \hat{c}_{x,l}) (\hat{c}_{x,m}^\dagger \cdot \hat{c}_{x,k-l+m}) \right] \\ & - \hat{c}_{x,k}^\dagger \cdot \hat{c}_{x,k}. \end{aligned} \quad (19)$$

The individual terms are again the widthwise hopping, now diagonal, then the lengthwise hopping, which did not change, and the interaction term. The interaction term is now a proper four-body interaction spanning every ring of the cylinder. For very wide cylinders, this would be prohibitively expensive, but since DMRG is exponentially bound in the width of the cylinders due to the entanglement structure regardless of the choice of real- or momentum space basis, this is not a major concern.

Again following Ref. [12], we can also exploit the  $\mathbb{Z}_W$  momentum conservation symmetry to increase the sparsity of both the MPO and the MPS by attaching an additional quantum number  $K$  to each local basis state. This requires a nonhomogenous basis, as the one-electron state on site  $(x,k)$  has to transform as  $K = k$  (while of course keeping its spin and charge quantum numbers  $S$  and  $N$ , respectively).

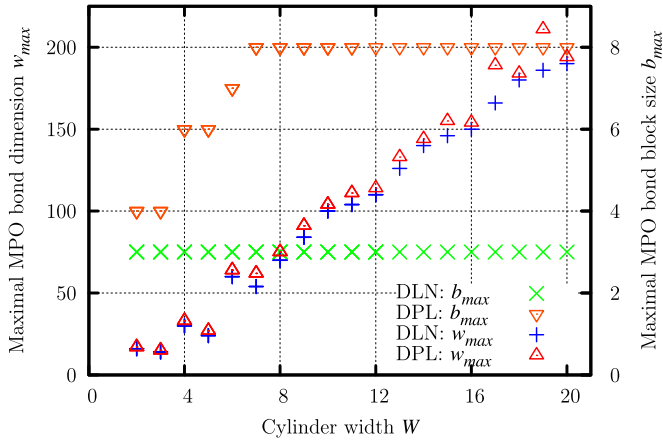


FIG. 8. Maximal MPO bond dimension (left axis) and maximal block size (right axis) for the Fermi-Hubbard Hamiltonian on a cylinder with nearest-neighbor hopping and (real-space) on-site interactions. The maximal bond dimension occurs in the middle of each ring of the cylinder and is independent of the cylinder length. This is also the bond on which the size of the largest quantum number block becomes maximal if only deparallelization (DPL) compression is employed. On this problem, SVD and delinearization (DLN) result in the same bond dimensions. With either of the two, the largest quantum number block is fairly uniformly three, only dropping down to two on the inter-ring connections.

Similarly, the two-electron state on this site has to transform as  $K = 2k$ . For the same reason, the operator  $\hat{c}_{x,k}^\dagger$  and its MPO representation not only transform as  $S = 1/2$  and  $N = 1$  but also as  $K = k$ , which depends on the site on which the operator acts.

There is additional freedom in the choice of ordering of the two-dimensional pairs  $(x, k)$  on the one-dimensional DMRG chain. Here, we choose a Z-like pattern, connecting the last site of each ring to the first site of the next ring. Further, we have the freedom to re-arrange the momentum sites  $k$  within the ring, ideally to minimize both the bond dimension of  $\hat{H}$  as well as to reduce the entanglement of the resulting MPS ground state. Expecting antiferromagnetic correlations, we place momenta separated by  $\pi$  next to each other, e.g., for eight sites, the ordering within a ring is  $k = 0, \pi, \pi/4, 5\pi/4, \pi/2, 3\pi/2, 3\pi/4$ , and  $7\pi/4$ .

The most important scaling is given by the maximal bond dimension of the MPO. The size of this parameter affects the runtime needed by DMRG. Figure 8 shows the maximal bond dimension of the MPO representation constructed using this method for  $U = 2$ . This maximal bond dimension is independent of the cylinder length  $L$  and occurs in the middle of each ring. In comparison with the simple deparallelization, the total bond dimensions resulting from either the SVD compression or the delinearization compression are only slightly smaller. However, both the SVD and delinearization method reduce the size of the largest dense blocks in the tensors from 8 to 3.

Inspecting those dense blocks on, e.g., a  $16 \times 4$  lattice, we find that with the delinearization method, 8% of the stored values are exactly zero, 14% exactly  $-1$ , and 32% exactly  $+1$ . In comparison, after SVD compression, an exactly zero value

never occurs and the two most common values are  $\approx \pm \sqrt{4/3}$  at 4% and 14%, respectively. Hence even in such small blocks of size at most  $3 \times 3$ , the delinearization method preserves sparsity and a relatively simple tensor structure to a noticeable degree.

Independent of the compression method, we observe largely linear growth of the maximal bond dimension with the cylinder width. This can be explained by the momentum conservation in the interaction term: given two fixed operators on one half of the system and a third operator on the other half, there is only one valid location for the fourth operator. Hence we get overall  $O(L)$  scaling.

### C. Full electronic randomized Fermi-Hubbard representation

Contrary to the fairly homogenous problems in solid-state physics, the application of MPO-based algorithms in quantum chemistry is more difficult [13]. In particular, there are often long-range four-body interactions with different coupling coefficients. As a toy model for such a Hamiltonian, we consider the operator

$$\hat{H} = \sum_{\sigma\tau=\uparrow\downarrow} \sum_{ijkl} V_{ijkl} \hat{c}_{i\sigma}^\dagger \hat{c}_{k\tau}^\dagger \hat{c}_{l\tau} \hat{c}_{j\sigma} \quad (20)$$

with  $V_{ijkl} = V_{jilk}$ ,  $|V_{ijkl}| < 2$  but coefficients otherwise random. Construction of this Hamiltonian with the presented method is extremely expensive— $L^4$  MPO-MPO additions have to be evaluated and the intermediate sums have to be continuously compressed to avoid quartic growth of bond dimensions. Nevertheless, we are able to construct the MPO representation of the Hamiltonian for  $L$  up to  $\approx 34$ . With more advanced techniques and a modest amount of preprocessing, which are outside the scope of this paper, it would be possible to also construct the Hamiltonian for larger systems.

For this  $\hat{H}$ , the maximal bond size always occurs in the middle of the system at bond  $L/2$  (or bonds  $L/2 - 1$  and  $L/2$  for odd  $L$ ). It is possible to sum up partial terms in (20) to the left and right of a given bond so that there are only  $O(L^2)$  terms remaining on either side [13].

Figure 9 shows the maximal bond dimension for a given system length after construction from single-site operators via multiplication, addition and compression by deparallelization (every  $L$  steps) and SVD (every  $L^2$  steps). Using the delinearization method instead of the SVD compression, the resulting bond dimensions increase slightly at larger system sizes, as the delinearization cannot always break cyclic linear dependencies in its input columns. However, the bond dimensions as returned by delinearization still have decidedly quadratic scaling.

The main advantage of the method is then in its flexibility. Adding another type of interaction, changing coefficients or changing the system size can be done independently of the implementation of the compression methods as well as the definition of the single site operators.

The parameters of the optimal result can be explained as follows. First, there are always two identity terms which correspond to summands with all  $i, j, k$ , and  $l$  to the left or right of the center bond, resulting in the two constant terms. Further, there are  $L$  contributions each for one out of



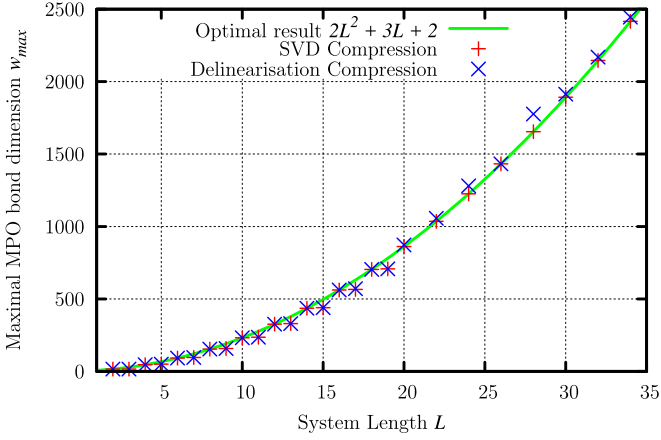


FIG. 9. Maximal MPO bond dimension in the middle of the chain for the representation of the full quantum chemistry Hamiltonian in (20). The maximal bond dimension after SVD compression for even lengths behaves exactly as  $w_{\max}(L) = 2L^2 + 3L + 2$ , which is the optimal result. We included some data for odd lengths  $L$  for completeness; the increase in bond dimension from  $L = 2n$  to  $L = 2n + 1$  is consistently four with SVD. With delinearization compression, the result is exactly the same as with SVD for  $L \leq 16$ , for larger system sizes, the optimal representation is not always found, but the scaling is still decidedly quadratic.

$i, j, k$ , and  $l$  to the left and three to the right (and vice versa) as well as  $L$  contributions for  $i = j$  or  $k = l$ . Finally, there are  $L^2$  ways each to distribute two out of  $i, j, k$ , and  $l$  on the left or the right of the system. This is in agreement with recent results by Chan *et al.* [14] who also find a leading term  $2L^2$ . The SVD compression therefore leads to the optimal MPO representation with scaling  $O(L^2)$ .

### VIII. CALCULATION OF HIGHER MOMENTS

The calculation of higher moments is an obvious application of MPO techniques. For example, the energy variance  $\sigma^2 = \langle H^2 \rangle - \langle H \rangle^2 = \langle (H - E)^2 \rangle$  is a robust alternative to calculating the truncation error that has many advantages [4]. However, the naive computation of the variance as the difference between the second moment  $\langle H^2 \rangle$  and the square of the energy is extremely prone to catastrophic cancellation [19] due to subtraction of two numbers that have a large magnitude, whereas the result has typically a small magnitude. In double-precision floating point numerics that are typical for MPS calculations, there are approximately 16 decimal digits of precision available, so if one wants to be able to resolve a variance of, say  $10^{-10}$ , this implies that the total energy of the system can be no larger than  $10^3$ . In practice that is a gross overestimate, since roundoff errors will account for at least a digit or two as well, which means that in typical calculations one encounters numerical problems evaluating the variance when the system size gets to around  $L \sim 100$  or so sites. The solution to recovering numerical precision is to construct an MPO representation of  $(H - E)^2$  directly, thereby distributing the constant energy term across each site of the MPO. The intermediate sums formed when contracting the

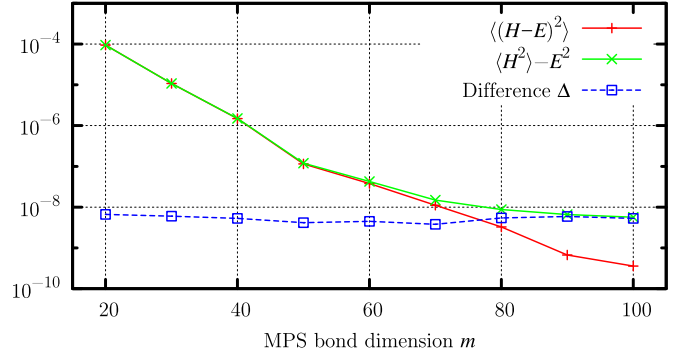


FIG. 10. The variance  $\langle (H - E)^2 \rangle$  of the MPS approximation to the ground state of the  $S = 1$  Heisenberg chain, as a function of bond dimension  $m$ . The naive calculation of  $\langle H^2 \rangle - E^2$  is subject to catastrophic cancellation and cannot be obtained accurately. The properly constructed MPO representation for  $(H - E)^2$  is well-conditioned and obtains full numerical precision. The difference  $\Delta = (\langle H^2 \rangle - E^2) - \langle (H - E)^2 \rangle$  is consistently of order  $10^{-8}$ , i.e., relevant as soon as the variance becomes sufficiently accurate.

MPO are then bounded to be  $O(1)$ —the only component of the summation that diverges with system size is the variance itself, which is only linear in  $L$ . The MPO representation  $(H - E)^2$  is straightforward to construct, by starting from the MPO representation of  $H$  and subtracting the local contribution to the energy at each site,

$$W_i^{H-E} = W_i^H - E_i, \quad (21)$$

where each  $E_i$  is the contribution to the energy due to site  $i$ . Unless a better value is available,  $E_{\text{total}}/L$  can be used here. The squared MPO can then be generated straightforwardly, and it is important that the MPO compression scheme preserves the structure of the MPO, which is true at least for the parallel compression algorithm. This loss of accuracy through catastrophic cancellation is demonstrated in Fig. 10. This is obtained for a spin  $S = 1$  Heisenberg chain with  $L = 100$  sites. With the uncontrolled algorithm, catastrophic cancellation limits the accuracy of the variance calculation to  $O(10^{-8})$ , but with proper construction of the MPO, the variance can be calculated to full accuracy. This difference in how the variance is computed depends drastically on the system size, since the variance is linear in system size but the cancellation of terms in  $\langle H^2 \rangle - E^2$  is  $O(L^2)$ . Higher moments are affected even more drastically, since the  $k^{\text{th}}$  moment involves subtraction of terms of order  $L^k$ . The generalization to higher moments is best viewed in terms of the cumulant expansion, since each cumulant is linearly extensive in the system size and they exactly capture the numerical divergences of the moment expansion. The first cumulant  $\kappa_1$  is just the energy itself, and the second cumulant  $\kappa_2 = \sigma^2$  is the variance. The third cumulant, which characterizes the skewness of the distribution, is given by  $\kappa_3 = \langle H^3 \rangle - 3\langle H^2 \rangle \langle H \rangle + 2\langle H \rangle^3$ , and is obtained from the MPO representation

$$W^{[3]} = ((H - \kappa_1)^2 - \kappa_2)(H - \kappa_1). \quad (22)$$

If the MPO is properly constructed in this way then there are no intermediate terms that grow with the system size as the MPO is contracted, and there is essentially no practical limit to the accuracy of evaluating higher order moments (the exponential time cost from the dimension of the MPO is the limiting factor, not accuracy of computation).

This structure is implicit in the triangular MPO formulation for infinite systems [20]. In determining the expectation values of higher moments of an iMPO, the recursive formulation presented in Ref. [21] keeps control over the numerical precision without explicitly removing the energy contributions, due to the particular triangular structure (Jordan form) of the expectation values. The contributions to the moment that diverge with each power of the system size are obtained separately as the coefficients of a polynomial expansion. Efficient compression of translationally invariant infinite MPOs has some distinct features compared with finite MPOs, and this will be described in detail in a future publication [22].

## IX. CONCLUSIONS & OUTLOOK

This paper presents a generic construction method to generate an efficient MPO representation of arbitrary operators in the context of second-generation DMRG algorithms. The method only requires the definition of single-site operator tensors and the implementation of arithmetic operations on MPOs as well as compression of an MPO. Any operator can then be expressed in a few loops of any object-oriented programming language. In turn, the method facilitates the study of varying and complex systems, as the amount of work to be done up front prior to DMRG calculations is substantially lowered.

The simplest compression method presented (deparallelization) can handle most nearest-neighbor Hamiltonians, while for more complicated MPOs, either the SVD or the delinearization should be used (cf. Table II).

The resulting MPO either exactly reproduces the optimal analytical solution (for spatially homogenous short-range operators) or is the optimal representation which would be difficult to construct analytically (for medium-range Hamiltonians in two dimensions as well as powers of short-range Hamiltonians). In principle, it is also possible to apply the method to the quantum chemistry Hamiltonian, but computational costs of a naive implementation become too large to be feasible. It would be possible to combine the compression techniques presented here with other compression methods in the future, if necessary. The implementation is compatible with non-Abelian spin and charge symmetries enforced on the tensor level. The generalization to tree-tensor networks is also straightforward.

TABLE II. Overview comparison of the three compression methods presented in this paper. Computational cost is a rough statement regarding the relative costs of the methods, as all three scale cubically in the bond dimension of the MPO.

| Method            | Optimal $w_{\max}$ | Sparsity  | Spurious terms | Implementation Complexity | Computational Cost |
|-------------------|--------------------|-----------|----------------|---------------------------|--------------------|
| Deparallelization | Easy MPOs          | Preserved | None           | Simple                    | Cheap              |
| Rescaled SVD      | Always             | Lost      | Yes            | Simple (with LAPACK)      | Expensive          |
| Delinearization   | Most MPOs          | Preserved | Nearly none    | Medium                    | Medium             |

## ACKNOWLEDGMENTS

C. Hubig acknowledges funding through the ExQM graduate school and the Nanosystems Initiative Munich. I. McCulloch acknowledges support from the Australian Research Council (ARC) Centre of Excellence for Engineered Quantum Systems, grant CE110001013. and the ARC Future Fellowships scheme, FT140100625.

## APPENDIX A: SUGGESTED MPO COMPRESSION PROCEDURE

MPO compression of an arbitrary operator should occur in three stages: (1) performing one full sweep using the deparallelization method, (2) performing sweeps using the strict delinearization method until bond dimensions stay constant, and (3) performing sweeps using the relaxed delinearization method until bond dimensions stay constant. The motivation for this sequence is to firstly reduce the bond dimension as much as possible with the fairly cheap deparallelization, then move on to the more costly delinearization and finally, if a cyclic dependency occurs which cannot be broken without allowing cancellation to zero, use the relaxed delinearization. Note that if the MPO is already optimal, the last step will not introduce such small terms.

Independent of the compression method, each full sweep iterates twice over the full system, once from left to right and then from right to left. On each site  $i$ , the local tensor  $W_{i;w_{i-1}w_i}^{\sigma_i\tau_i}$  is reshaped into a matrix  $M_{\gamma w_i}$  ( $M_{\gamma w_{i-1}}$ ) during left-to-right (right-to-left) sweeps. The matrix  $M$  is then decomposed as  $M = \tilde{M} \cdot T$ .  $\tilde{M}$  is reshaped into the new site tensor  $\tilde{W}_{i;w_{i-1}\tilde{w}_i}^{\sigma_i\tau_i}$  ( $\tilde{W}_{i;\tilde{w}_{i-1}w_i}^{\sigma_i\tau_i}$ ) with the transfer matrix  $T$  being multiplied into the next site tensor  $W_{i+1}$  ( $W_{i-1}$ ) during left-to-right (right-to-left) sweeps. The decomposition  $M \rightarrow \tilde{M} \cdot T$  is described in the following sections for the deparallelization and delinearization methods.

## APPENDIX B: DEPARALLELIZATION ALGORITHM

*Input:* Matrix  $M_{ab}$ .

*Output:* Matrices  $\tilde{M}_{a\beta}$ ,  $T_{\beta b}$  so that  $M_{ab} = \sum_{\beta} \tilde{M}_{a\beta} T_{\beta b}$  and  $\tilde{M}$  has at most as many columns as  $M$  and no two columns that are parallel to each other.

*Procedure:*

- (1) Let  $K$  be the set of kept columns, empty initially.
- (2) Let  $T$  be the dynamically resized transfer matrix.
- (3) For every column index  $j \in [1, b]$ :
  - (3.1) For every kept index  $i \in [1, |K|]$ :
    - (3.1) If the  $j$ th column  $M_{:j}$  is parallel to column  $K_i$ :
      - (3.1) set  $T_{i,j}$  to the prefactor between the two columns.

(3.2) Otherwise:

(3.1) add  $M_{:,j}$  to  $K$ , set  $T_{|K|,j} = 1$ .

(4) Construct  $\tilde{M}$  by horizontally concatenating the columns stored in  $K$ .

(5) Return  $\tilde{M}$  and  $T$ .

The check for parallelism is ideally done on an elementwise basis by finding the first nonzero element of either column, calculating the factor between it and the corresponding element of the other column and then ensuring that all other elements agree on that prefactor. Zero columns should be removed with a corresponding zero column stored in  $T$ .

### APPENDIX C: DELINEARIZATION ALGORITHM

*Input:* Matrix  $M_{ab}$ , threshold matrix  $\Delta_{ab}$ .

*Output:* Matrices  $\tilde{M}_{ab}$ ,  $T_{bb}$  so that  $M_{ab} = \sum_{\beta} M_{a\beta} T_{\beta b}$  and  $\tilde{M}$  has at most as many columns as  $M$  and all columns in  $\tilde{M}$  are linearly independent.

*Remark.* Initially, the threshold matrix  $\Delta_{ab}$  is constructed from  $W_{i;\tau_i w_i}$  as  $\Delta_{(\sigma_i \tau_i w_{i-1}) w_i} = \sum_{\sigma'_i \tau'_i} |W_{i;\tau'_i w_i}| \cdot \varepsilon_D$ , i.e., each element is the 1-norm of the original operator to which it belongs multiplied by a small threshold.

*Procedure:*

(1) If relaxed delinearization: set all elements  $\Delta_{ab} \equiv 0$  to  $\varepsilon_D$ .

(2) Deparallelize the rows of  $M_{ab}$ :

$$M_{ab} \rightarrow R_{a\alpha} M_{\alpha b}^p, \quad (C1)$$

$$\Delta_{ab} \rightarrow R_{a\alpha} \Delta_{\alpha b}^p, \quad (C2)$$

where the elements of  $\Delta^p$  are chosen as the smallest elements in that column from nonzero rows that were parallel to the kept row.

(3) Sort the columns of  $M^p$  according to the following criteria, resulting in  $M^{pp}$ ,  $\Delta^{pp}$  and a permutation matrix  $P$ . Sorting criteria are

(3.1) The number of exactly zero values in the column,

(3.2) if tied, the number of exactly zero thresholds in the same column of  $\Delta^p$ ,

(3.3) if tied, the number of exactly zero values from the bottom of the column,

(3.4) if tied, the number of exactly zero thresholds from the bottom of the same column. of  $\Delta^p$

(4) For every column  $\mu$  and associated threshold column  $\delta$  in  $M^{pp}$  and  $\Delta^{pp}$

(4.1) Attempt to solve

$$Ax = \mu, \quad (C3)$$

where  $A$  is the matrix from eligible previously kept columns. A column is eligible for inclusion in  $A$  if it has no nonzero entry in a row where  $\delta$  is exactly zero.

The coefficients  $x$  are found via QR decomposition with column scaling (by their respective norms). Rows of  $R$  and  $Q^H \mu$  are scaled so that the right-hand side is either 1 or 0 prior to solution by backwards substitution.

(4.2) If any coefficients in  $x$  have absolute value less than  $\varepsilon_t$ , remove the associated column from the eligible set to build  $A$  and repeat.

(4.3) If any coefficients in  $x$  are close to  $\pm 1$ , replace them by  $\pm 1$ .

(4.4) If each element  $(Ax - c)_i$  of the residual is smaller than  $\delta_i \times \text{cols}(A)$ :

(4.1) store the coefficients  $x$ ;

(4.5) Else

(4.1) add the column to the set of kept columns and store a coefficient of 1 in the appropriate place.

(5) Collect all kept columns into  $M^{pC}$ , associated columns from  $\Delta^{pP}$  into  $\Delta^{pC}$  and construct the transfer matrix  $T^C$  from the stored coefficients times the permutation matrix  $P$ .

(6) Multiply the row-deparallelization transfer matrix  $R$  back into  $M^{pC}$  and  $\Delta^{pC}$ , yielding  $M^C$  and  $\Delta^C$ .

(7) If the number of columns in  $M^C$  is equal to the number of columns in  $M$ , replace  $M^C = M$ ,  $\Delta^C = \Delta$ ,  $T^C = \mathbf{1}$ .

(8) Repeat steps (2) through (6) for  $M^{C\dagger}$  and  $\Delta^{C\dagger}$  (i.e., delinearize the rows of  $M^C$ ):

$$M^{C\dagger} = M^{CR} T^R, \quad (C4)$$

$$M^C = T^{R\dagger} M^{CR\dagger}. \quad (C5)$$

(9) If neither  $T^{R\dagger}$  nor  $M^C$  have fewer columns than  $M$

(9.1) return  $\tilde{M} = M$  and  $T = \mathbf{1}$ .

(10) Else if  $T^{R\dagger}$  has fewer columns than  $M^C$ ,

(10.1) return  $\tilde{M} = T^{R\dagger}$ ,  $T = M^{CR\dagger} \cdot T^C$ ,

(11) Else

(11.1) return  $\tilde{M} = M^C$ ,  $T = T^C$ .

*Remark.* During matrix-matrix products  $R_{ij} = \sum_k A_{ik} B_{kj}$ , it is helpful and often necessary to set elements of  $R$  for which  $|R_{ij}| < \sum_k |A_{ik}| |B_{kj}| \varepsilon_Z$  is true to zero. This ensures that where we allow cancellation to zero, we do not introduce additional terms whenever possible.

Step 1 removes the requirement that we cannot allow cancellation to zero. Step 2 usually halves the number of rows of  $M$ , as there are often many zero rows or rows parallel to previous ones, making the subsequent QR decompositions both faster and more accurate. Step 3 sorts columns such that those with few nonzero entries are considered first while attempting to keep an upper-triangular form. The former helps to find optimal noncancelling linear superpositions, while the latter attempts to restore the usually preferred triangular form whenever possible. Steps 7 and 9 reduce numerical errors by reverting to the input matrix if no improvements have been found. Finally, steps 8 and 10 often help to break cyclic dependencies and achieve optimal compression.

### APPENDIX D: NUMERICAL THRESHOLD VALUES

The relevant three threshold values are, with the machine precision  $n_\varepsilon \approx 10^{-16}$ .

(1)  $\varepsilon_D$ . During delinearization, a new column has to be equal to the original one to within this value, relative to operator norms. In practice, we found  $\sqrt{n_\varepsilon} \approx 10^{-8}$  to be a suitable value, as columns are usually either completely dependent (with very small error) or differ substantially (with very large error). Too small a threshold will lead to failure to optimize in some cases, as numerical noise may become relatively large during a long calculation.

(2)  $\varepsilon_Z$ . The delinearization method is able to work with operators of very different orders of magnitude in the same MPO. In turn, this means that small terms are not automatically discarded as with SVD. This implies that during the various matrix-matrix products encountered during MPO compression, special care has to be taken to avoid introducing artificial small terms. In practice, we found  $10^5 n_\varepsilon \approx 10^{-11}$  to work.

(3)  $\varepsilon_t$ . This threshold serves to avoid small coefficients in the transfer matrix, which would lead to valid small coefficients in the next tensor. For most sensible operators, coefficients should be of order one and if this is not possible, it may well be desired to keep the components separate rather than conflating them into a single column. Our implementation uses a value of  $10^{-5}$  here.

- 
- [1] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
  - [2] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
  - [3] S. Östlund and S. Rommer, *Phys. Rev. Lett.* **75**, 3537 (1995).
  - [4] I. P. McCulloch, *J. Stat. Mech.* (2007) P10014.
  - [5] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New J. Phys.* **12**, 025012 (2010).
  - [6] U. Schollwöck, *Ann. Phys.* **326**, 96 (2011).
  - [7] S. R. White, *Phys. Rev. B* **72**, 180403 (2005).
  - [8] M. Dolfi, B. Bauer, M. Troyer, and Z. Ristivojevic, *Phys. Rev. Lett.* **109**, 020604 (2012).
  - [9] E. M. Stoudenmire and S. R. White, *Phys. Rev. B* **87**, 155137 (2013).
  - [10] F. Fröwis, V. Nebendahl, and W. Dür, *Phys. Rev. A* **81**, 062337 (2010).
  - [11] G. M. Crosswhite, A. C. Doherty, and G. Vidal, *Phys. Rev. B* **78**, 035116 (2008).
  - [12] J. Motruk, M. P. Zaletel, R. S. K. Mong, and F. Pollmann, *Phys. Rev. B* **93**, 155139 (2016).
  - [13] S. Keller, M. Dolfi, M. Troyer, and M. Reiher, *J. Chem. Phys.* **143**, 244118 (2015).
  - [14] G. Kin-Lic Chan, A. Kesselman, N. Nakatani, Z. Li, and S. R. White, *arXiv:1605.02611*.
  - [15] C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf, *Phys. Rev. B* **91**, 155115 (2015).
  - [16] M. Dolfi, B. Bauer, S. Keller, A. Kosenkov, T. Ewart, A. Kantian, T. Giamarchi, and M. Troyer, *Comput. Phys. Commun.* **185**, 3430 (2014).
  - [17] E. M. Stoudenmire and S. R. White, *New J. Phys.* **12**, 055026 (2010).
  - [18] E. Stoudenmire and S. R. White, *Annu. Rev. Condens. Matter Phys.* **3**, 111 (2012).
  - [19] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002), p. 81.
  - [20] I. P. McCulloch, *arXiv:0804.2509*.
  - [21] L. Michel and I. P. McCulloch, *arXiv:1008.4667*.
  - [22] I. P. McCulloch (unpublished).