Low communication high performance ab initio density matrix renormalization group algorithms

Huanchen Zhai[a] and Garnet Kin-Lic Chan[b]
Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, CA 91125, USA

(Dated: 23 March 2021)

There has been recent interest in the deployment of *ab initio* density matrix renormalization group computations on high performance computing platforms. Here, we introduce a reformulation of the conventional distributed memory *ab initio* DMRG algorithm that connects it to the conceptually simpler and advantageous sum of sub-Hamiltonians approach. Starting from this framework, we further explore a hierarchy of parallelism strategies, that includes (i) parallelism over the sum of sub-Hamiltonians, (ii) parallelism over sites, (iii) parallelism over normal and complementary operators, (iv) parallelism over symmetry sectors, and (v) parallelism over dense matrix multiplications. We describe how to reduce processor load imbalance and the communication cost of the algorithm to achieve higher efficiencies. We illustrate the performance of our new open-source implementation on a recent benchmark ground-state calculation of benzene in an orbital space of 108 orbitals and 30 electrons, with a bond dimension of up to 6000, and a model of the FeMo cofactor with 76 orbitals and 113 electrons. The observed parallel scaling from 448 to 2800 CPU cores is nearly ideal.

I. INTRODUCTION

The Density Matrix Renormalization Group (DMRG) algorithm[1,2] is established as a method to obtain highly accurate low-energy eigenstates of *ab initio* quantum chemistry Hamiltonians[3–15]. While multiple techniques can now solve for low-energy eigenstates to high precision in problems that are formally beyond the reach of full configuration interaction[16–20], DMRG provides a unique capability to treat problems with a large number of open shells.[21,22] Consequently it is particularly useful in active space problems where a large fraction of the orbitals have open shell character, for example, as found in molecular clusters with multiple open-shell transition metal centers.[23–28] In many such problems, teasing out the relevant chemistry requires not only a single ground-state energy calculation, but also the characterization of many competing low-energy states. For such applications, improving the scalabity and efficiency of current DMRG implementations is highly desirable.

Over the last two decades, many different strategies have been proposed to parallelize the DMRG algorithm in quantum chemistry. These include:

(i) Parallelism over dense matrix multiplications[29,30]. This is a fine-grained parallelism which is effective when the size of the dense matrices is sufficiently large (namely, when a large Matrix Product State (MPS) bond dimension M is used). It can be implemented simply by linking the code to a parallelized shared-memory math library.

(ii) Parallelism over symmetry sectors,[11,30] which is available when DMRG is implemented with symmetry restrictions. Typically, particle number, total spin or projected spin, and spatial symmetry are used in *ab initio* DMRG implementations.

(iii) Parallelism over normal and complementary operators[31,32]. This is often considered the largest source of parallelism for typical *ab initio* problems.

(iv) Parallelism over a sum of sub-Hamiltonians[33]. This is a coarse-grained parallelism with very low communication cost, and is easy to express in a Matrix Product Operator (MPO) description of DMRG.

(v) Parallelism over sites[34]. For a large number of sites, this is an additional source of coarse grained parallelism. Such an implementation relies on the transformation of the MPS to a form with multiple canonical centers.

Recently, Brabec et. al. reported a non-spin-adapted massively parallel implementation of DMRG for quantum chemistry using strategies (ii) and (iii).[35] We also note promising recent progress in GPU accelerated parallel DMRG algorithms.[36–38] However, to the best our knowledge, there has not been an implementation that utilizes all 5 sources of parallelism in a scalable DMRG code for *ab initio* problems. This may be partly ascribed to the fact that strategies (iv) and (v) are most conveniently implemented in a DMRG code[39,40] that is structured using an MPO/MPS formalism,[33,41] while many other efficient *ab initio* DMRG implementations[42,43] using strategies (i), (ii) and (iii) are organized around the construction and transformation of renormalized operators.[6]

In this work, we first reformulate strategy (iii) for a distributed memory model using the sum of sub-Hamiltonians language. This demonstrates that a low communication version of strategy (iii) can in fact also be viewed as a variant of strategy (iv). This analysis constitutes Section II A and Section II B. In Section II C, we discuss how the load-imbalance that arises in strategy (v) can be alleviated via the dynamical determination of connection sites. In Section II D to Section II F we briefly introduce the shared memory parallelism strategies (i), (ii) and (iii) used in this work. Next, in Section III, we illustrate the computational performance of our new implementation of parallel DMRG for a recent ground-state

[a] Electronic mail: hczhai@caltech.edu
[b] Electronic mail: gkc1000@gmail.com

benzene benchmark[20] in a polarized valence double zeta basis.[44] Although not a correlated or open-shell system that is particularly suited to DMRG, the size of the calculation serves to illustrate the scalability of our algorithm. For a correlated electron problem with many open shells that is more suited for DMRG, we also consider a model of the FeMo cofactor system[45], and observe that a similar scaling can be achieved. Finally, the conclusions are given in Section IV.

II. THEORY

Rather than reintroduce the DMRG formalism here, we summarize the background theory and notation for the serial DMRG algorithm[1,2] and the SU(2) (spin-adapted) *ab initio* DMRG algorithm[32,42,46] in Appendix A and Appendix B, respectively. We encourage readers unfamiliar with the standard DMRG algorithm and terminology to first consult these appendices.

A. Parallelism over renormalized operators

In most parallel implementations of *ab initio* DMRG, the most important source of distributed memory parallelism comes from distributing the left-right renormalized operator decomposition of the Hamiltonian, as discussed in Ref. 31. In this approach, "normal" and "complementary" renormalized operators (see Appendix B for definitions) are assigned to different processors according to their orbital indices.

The leading communication cost per sweep in the approach described in Ref. 31 is $O(16M^2K^2 \log P_{\text{hamil}})$ from the blocking step, where M is the MPS bond dimension, K is the number of sites, and P_{hamil} is the total number of processors (processor cores) at this parallelism level. The sub-leading term in the communication cost is $O(M^2K^2 \log P_{\text{hamil}})$ from the transformation (rotation) step.

In order to achieve better scalability, it is desirable to reduce the communication cost. For this purpose, we note that the leading and sub-leading terms in the communication cost in the above approach mainly come from the accumulation of the $R_i^{L/R[\frac{1}{2}]}$ operators (defined in Eq. (B7)). Therefore, the communication cost can be greatly reduced by never accumulating $R_i^{L/R[\frac{1}{2}]}$. Namely, we can arrange for each processor to compute and store a partial contribution to $R_i^{L/R[\frac{1}{2}]}$ for all indices i. Compared to the original scheme, this new scheme only needs to communicate when accumulating the wavefunction, with a communication cost of $O(16M^2K \log P_{\text{hamil}})$ per sweep. However, since all partial components of the $R_i^{L/R[\frac{1}{2}]}$ operators have to enter into the solving (Davidson) step, the computational cost for the solving step increases from $O(M^3(K^3 + K^2)/P_{\text{hamil}})$ to $O(M^3K^3/P_{\text{hamil}} + M^3K^2)$ per sweep. The total disk storage cost also increases from $O(M^2(K^3 + K^2))$ to $O(M^2K^3 + M^2K^2P_{\text{hamil}})$. For the typical case where $P_{\text{hamil}} \not\gg K$, the increase in the subleading term of the storage is not a large concern.

We note that in this new scheme, the communication of renormalized operators is completely removed. In other words, each processor performs blocking, solving, and transformation steps for a part of the Hamiltonian - i.e. a sub-Hamiltonian - independently, and only wavefunctions from the solving step are communicated. This motivates a more general picture where we can develop low communication algorithms that are formulated in terms of sub-Hamiltonians, rather than the left-right decomposition of the Hamiltonian.

B. Parallelism over sub-Hamiltonians

For this purpose, we write the *ab initio* Hamiltonian Eq. (B1) as

$$\hat{H} = \hat{H}^{(1)} + \hat{H}^{(2)} + \cdots + \hat{H}^{(P_{\text{hamil}})} \tag{1}$$

where $\hat{H}^{(p)}$ is the sub-Hamiltonian assigned to processor p. To describe this assignment, we can write

$$\hat{H}^{(p)} = \frac{1}{2}\sum_{ij,\sigma}\Big[\text{proc}(p,i) + \text{proc}(p,j)\Big] t_{ij,\sigma}\ a^\dagger_{i\sigma}a_{j\sigma} + \frac{1}{2}\sum_{ijkl,\sigma\sigma'} \text{proc}(p,i,j,k,l)\ v_{ijkl,\sigma\sigma'}\ a^\dagger_{i\sigma}a^\dagger_{k\sigma'}a_{l\sigma'}a_{j\sigma} \tag{2}$$

where $\text{proc}(p,\cdots)$ defines the mapping from orbital indices to processor rank p ($p = 1, 2, \cdots, P_{\text{hamil}}$). There is clearly much freedom in choosing the definition of these mappings.

A possible definition of $\text{proc}(p,\cdots)$ is

$$\text{proc}(p,i) = \begin{cases} 1 & p \equiv i \pmod{P_{\text{hamil}}} \\ 0 & \text{otherwise} \end{cases}$$
$$\text{proc}(p,i,j) = \begin{cases} 1 & p \equiv \frac{(j-1)j}{2} + i \pmod{P_{\text{hamil}}} \text{ and } i \le j \\ 1 & p \equiv \frac{(i-1)i}{2} + j \pmod{P_{\text{hamil}}} \text{ and } i > j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and $\text{proc}(p,i,j,k,l)$ has the same value for any permutation of parameters i,j,k,l, namely

$$\text{proc}(p,i,j,k,l) = \text{proc}(p, \text{sorted: } i,j,k,l) \tag{4}$$

As discussed above, we can think of a modified version of the normal-complementary operator parallelism as arising from a particular decomposition into sub-Hamiltonians. In particular, the NC renormalized operator partition (Eq. (B5)) corresponds to

$$\text{proc}(p, \text{sorted: } i,j,k,l) = \begin{cases} \text{proc}(p,j) & j = k \\ \text{proc}(p,i,j) & \text{otherwise} \end{cases} \tag{5}$$

while the CN renormalized operator partition (Eq. (B6)) corresponds to

$$\text{proc}(p, \text{sorted: } i,j,k,l) = \begin{cases} \text{proc}(p,j) & j = k \\ \text{proc}(p,k,l) & \text{otherwise} \end{cases} \quad (6)$$

Here the notation $\text{proc}(p, \text{sorted: } i,j,k,l)$ means that $i \leq j \leq k \leq l$. Based on the above definitions, the symmetry condition $\text{proc}(p,i,j) = \text{proc}(p,j,i)$ is satisfied. This is important for efficiency, since the operator symmetry conditions used for efficient DMRG algorithms (see Eq. (B8)) can still be used on each processor without any communication.

To see how this assignment of Hamiltonian terms gives the correct scaling for multiple processors, we note that in the NC partition in Eq. (B5) the summation over two-index operators is over indices in the left block of sites L, which are the small indices i,j in the tuple i,j,k,l, and thus in Eq. (5) the indices i,j are used for the processor assignment. For similar reasons, the large indices k,l are used in the CN partition case. This ensures that the total number of terms in the left-right decomposition of the effective Hamiltonian on each processor is roughly $O(K^2/P_{\text{hamil}})$ (if $P_{\text{hamil}} \not\gg K$).

It is worth noting that Eq. (1) is in the same spirit as the sum of MPOs formulation first introduced in Ref. 33. This is often considered a different strategy from the strategy of parallelism over renormalized operators. The two methods indeed have a very different origin and motivation. However, our new formulation of the low communication version of the parallelism over renormalized operators establishes a clear connection between the two methods. In addition, we find that this new formulation inherits the most important advantages from both methods:

(i) Low communication time.[33] Since each sub-Hamiltonian can be manipulated completely independently, only the communication of the (small) wavefunction obtained from the DMRG solving step is required.

(ii) Simple implementation. To parallelize a serial *ab initio* DMRG code, one only needs to start with a distributed integral file, where for each processor some integrals t_{ij} and v_{ijkl} are set to zero according to Eq. (2). A single communication step then needs to be added to accumulate wavefunctions from all processors. No other part of the code needs to be changed significantly.

(iii) Compatibility with both the exact renormalized operator and compressed MPO DMRG formalisms. Because the description of our algorithm does not rely on specific definitions and choices of normal and complementary operators,[3] one has great freedom to decompose each sub-Hamiltonian. For example, we have presented examples that correspond to the conventional exact NC or CN renormalized operator partitions[6] and their corresponding MPOs, but other MPO representations of the sub-Hamiltonians, including compressed representations[33] can be used. In this work, we will only use exact MPO representations of the sub-Hamiltonians.

(iv) Compatibility with index-symmetry conditions. We note that the previous description of sub-Hamiltonians in Ref. 33 was based on splitting the Hamiltonian based on single site-indices. This has the disadvantage that it becomes difficult to use the two-index symmetry conditions Eq. (B8) to reduce the computational cost associated with each sub-Hamiltonian in a distributed setting, because a single site-index based processor assignment can easily assign index-symmetry related operators to different processors. The current two-index based splitting does not have this problem since $\hat{O}_{ij}$ and $\hat{O}_{ji}$ are always assigned to the same processor.

(v) Load balance between processors. The two-index based assignment assigns roughly equal amounts of work to different processors, if $P_{\text{hamil}} \not\gg K$.

(vi) Compatibility with the sum of sub-Hamiltonians and automated MPO construction. In Ref. 33, it was demonstrated that by expressing the *ab initio* Hamiltonian as a sum of K sub-Hamiltonians, we can work with K MPOs each with bond dimension $O(K)$, instead of one MPO with bond dimension $O(K^2)$. The advantage is that this captures the primary sparsity within the single MPO representation of the Hamiltonian. This means that a simple MPO construction of the sub-Hamiltonians, which uses only dense matrices, produces the correct serial computational cost of $O(M^3K^3 + M^2K^4)$, rather than the naive (and incorrect) cost of $O(M^3K^3 + M^2K^5)$ arising from single dense MPO representation of H, making the correct scaling of the *ab initio* implementation very easy to achieve. In particular, this is attractive when combined with various automated MPO construction approaches, which then do not need to implement sparse tensor algebra.[33,40,41,47] The two-index based sum of sub-Hamiltonians retains this attractive feature, but has the further advantage that the computational prefactor (e.g. from the sub-MPO bond dimensions) is smaller, when compared with the previous one-index decomposition.

In this work, we combine the low communication scheme based on sub-Hamiltonians with a mixed NC/CN partition[11] to achieve high efficiency. The mixed NC/CN partition introduces additional costs for computation and communication at the middle site of the sweep. These details are discussed in Section III A.

C. Parallelism over sites

A more recent approach to coarse-grained parallelism in DMRG is the "real space parallel DMRG" approach introduced by Stoudenmire and White,[34,48] which has been shown to give near ideal scaling in some calculations with model Hamiltonians and very recently for quantum chemistry Hamiltonian.[49] An implementation of this approach for (non-spin-adapted) quantum chemistry Hamiltonians can also be found in the QCMaquis code.[40]

The approach relies on a representation of the MPS with multiple canonical centers.[34] Each extra canonical center can be introduced by first performing a SVD on

the effective wavefunction (given in Eq. (A11)) at the original canonical center k

$$\mathbf{\Psi}[k]^{\text{eff}} = \mathbf{L}[k]\mathbf{S}[k]\mathbf{R}[k] \tag{7}$$

Then we can write

$$\mathbf{\Psi}[k]^{\text{eff}} = \mathbf{\Psi}_1[k]^{\text{eff}}\mathbf{S}[k]^{-1}\mathbf{\Psi}_2[k+1]^{\text{eff}} \tag{8}$$

where

$$\begin{aligned} \mathbf{\Psi}_1[k]^{\text{eff}} &= \mathbf{L}[k]\mathbf{S}[k] \\ \mathbf{\Psi}_2[k+1]^{\text{eff}} &= \mathbf{S}[k]\mathbf{R}[k] \end{aligned} \tag{9}$$

are the two new canonical centers at site k and $k+1$. Once we have two canonical centers in the MPS, two partial DMRG sweeps, namely, a backward sweep starting from site k and a forward sweep starting from site $k+1$, can be performed simultaneously by separate processors. The above approach can be invoked iteratively to generate P_{site} canonical centers in the MPS, where P_{site} is the total number of (groups of) processors at this level of parallelism. Matrix $\mathbf{S}[k]^{-1}$ (termed the connection matrix) is used after a round of forward and backward partial sweeps to merge the updated $\mathbf{\Psi}_1^{\text{new}}[k]^{\text{eff}}$ and $\mathbf{\Psi}_2^{\text{new}}[k]^{\text{eff}}$ to yield an approximation to the updated $\mathbf{\Psi}^{\text{new}}[k]^{\text{eff}}$

$$\mathbf{\Psi}^{\text{new}}[k]^{\text{eff}} = \mathbf{\Psi}_1^{\text{new}}[k]^{\text{eff}}\mathbf{S}[k]^{-1}\mathbf{\Psi}_2^{\text{new}}[k+1]^{\text{eff}} \tag{10}$$

Merging the two separately optimized portions of the MPS using this connection matrix does not change the MPS when the MPS has reached its variational optimum. The two partial sweeps over sites $\cdots, k$ and $k+1, \cdots$ cannot update the MPS bond between the sites k and $k+1$. Therefore, a sweep iteration at the connection site is performed, where Eq. (A12) is solved for the merged wavefunction $\mathbf{\Psi}^{\text{new}}[k]^{\text{eff}}$. The solution of Eq. (A12), denoted as $\mathbf{\Psi}'^{\text{new}}[k]^{\text{eff}}$, is then split according to Eq. (8) to generate the updated connection matrix $\mathbf{S}^{\text{new}}[k]^{-1}$.

In a typical *ab initio* application, the amount of computation is not distributed homogeneously among different groups of sites (see Fig. 1) because of the boundary effects of the MPS and the different block sizes from different truncations at different sites. In addition, the total number of sites available for this level of parallelism is limited. If the same number of sites is assigned to different processors, one observes a significant load imbalance, which negatively impacts the scalability.[49]

To alleviate this problem, similar to the dynamic boundary strategy used in Ref. 49, we have added an additional step to dynamically determine the position of the canonical center (connection site) to improve load balancing. After all processors finish their partial sweeps, the total computational cost is measured for each processor for the partial sweep and all its sweep iterations. From this, it is possible to estimate whether changing the connection site from k to $k+2$ (for example) reduces the cost discrepancy between the two processors connected at site k. If this is the case, then the connection site is moved to $k+2$, with the hope that this helps to reduce the degree of load imbalance during the next sweep.

We note that changing the position of the connection site between sweeps is not an operation with negligible cost, since not only the MPS tensors, but also the renormalized operators, need to be transformed (see Eq. (A15)). Consequently, in our implementation, we have limited the distance between the old connection site and the new connection site to at most two sites. This ensures that the operation itself does not consume a significant amount of time. In practice, we can start the DMRG algorithm with an arbitrary set of connection sites. After several sweeps with dynamical adjustment of the connection sites, we observe that we can often achieve a stable set of connection sites and a well-balanced workload amongst the processors. The performance of the parallelism over sites with the dynamical adjustment of connection sites is discussed in Section III C.

Comparing to the strategy very recently introduced in Ref. 49, our approach does not directly reduce the waiting time of the current sweep; instead, the performance statistics of the current sweep are accumulated, to determine the position of the connection sites for the next sweep. In contrast, the approach introduced in Ref. 49 completely removes the waiting time at each sweep, but introduces an extra projection error in the wavefunction initial guess (Eq. (10)) when the connection site is changed (which is larger for *ab initio* systems compared to spin systems, according to Ref. 49). This extra error in the wavefunction transformation may increase the number of Davidson iterations.

In order to improve single-node performance, we have also considered the fine-grained strategies for shared memory parallelism.[29] Most of them can be easily implemented in an *ab initio* DMRG code with minor modifications. These are now discussed.

D. Shared memory parallelism over normal and complementary operators

The left-right decomposition of Hamiltonian (Eq. (B5) and Eq. (B6)) is a sum of products of normal and complementary operators. For the *ab initio* sub-Hamiltonians, there are $O(K^2/P_{\text{hamil}} + K)$ terms in the summation. Therefore, for the matrix-vector multiplication

$$|\Psi'[k]^{\text{eff}}\rangle = \hat{H}[k]^{\text{eff}}|\Psi[k]^{\text{eff}}\rangle \tag{11}$$

invoked during the Davidson procedure, we can divide the work among T_{op} threads, namely

$$\hat{H}[k]^{\text{eff}} = \hat{H}[k]^{\text{eff}}_{(1)} + \hat{H}[k]^{\text{eff}}_{(2)} + \cdots + \hat{H}[k]^{\text{eff}}_{(T_{\text{op}})} \tag{12}$$

The partial contribution to $|\Psi'[k]^{\text{eff}}\rangle$ is computed on every thread t as

$$|\Psi'[k]^{\text{eff}}_{(t)}\rangle = \hat{H}[k]^{\text{eff}}_{(t)}|\Psi[k]^{\text{eff}}\rangle \tag{13}$$

Finally, a reduction step is performed to obtain $|\Psi'[k]^{\text{eff}}\rangle$, as

$$|\Psi'[k]^{\text{eff}}\rangle = |\Psi'[k]^{\text{eff}}_{(1)}\rangle + |\Psi'[k]^{\text{eff}}_{(2)}\rangle + \cdots + |\Psi'[k]^{\text{eff}}_{(T_{\text{op}})}\rangle \quad (14)$$

with a small additional computation cost of $O(16M^2K \log T_{\text{op}})$ per sweep.

E. Shared memory parallelism over symmetry sectors

In addition, every term in Eq. (13) is implemented as a block-sparse matrix-matrix multiplication, which can be further decomposed into dense matrix-matrix multiplications over independent symmetry sectors. Instead of using nested threaded parallelism over normal and complementary operators and symmetry sectors, we can collapse the two thread parallelism levels to one level,[35] to achieve a better load balance and reduce the overhead from creating threads.

F. Shared memory parallelism over dense matrix multiplication

Thread-level parallelism for dense matrix multiplication can be easily introduced by using a threaded math library.[29] The effectiveness of this lowest level of parallelism is analyzed in Section III B.

III. RESULTS

As a first benchmark, we assess our parallel DMRG implementation in a ground-state energy calculation of benzene using a cc-pVDZ basis[44] with an orbital space comprising 108 orbitals and 30 electrons.[20] Although the benzene system is a closed shell system and thus does not showcase the strengths of the DMRG algorithm, it nonetheless serves as an example in the literature where a DMRG calculation with a large bond dimension and a relatively large number of orbitals has been recently reported.

For the benzene calculation, we use particle number, SU(2) (spin) and C_s point group symmetry to reduce the overall cost of the calculation. The same orbitals, integrals and orbital ordering as in Ref. 20 were used in this work. The DMRG correlation energy at $M = 6000$ (plus approximately 200 states to represent the low-weight quantum numbers) obtained in this work is -859.1 mE_H. Given the differences in implementation that gives rise to small differences in bond truncations across many sweeps, this is in excellent agreement with the DMRG correlation energy (-859.2 mE_H) reported in Ref. 20 at $M = 6000$.

In addition, we demonstrate the performance of our DMRG implementation in a calculation on the FeMo cofactor system, using a model with 76 orbitals and 113 electrons in the active space recently proposed by Li et. al. in Ref. 45. This is an example of a system with multiple transition metal centers where the strengths of the DMRG algorithm can in principle be demonstrated. We use the integral file provided in Ref. 45 without any further orbital reordering. The state with total spin $S = 3/2$ is targeted.

All calculations in this work use the two-site DMRG algorithm with perturbative noise.[50] Five sweeps were performed at each MPS bond dimension M. To measure the wall time per sweep, we used the average wall time for the last four sweeps for each M. For the benzene system, to alleviate the problem of losing quantum numbers, we kept at least one state for each quantum number after the normal decimation process.[31] This makes the bond dimension M in the calculation slightly larger than its specified target value. For example, when M was set to 6000, the observed actual M was typically about 6200.

We denote different parallelism schemes by a set of numbers $P_{\text{site}}, P_{\text{hamil}}, T_{\text{op}}$ and T_{dense} indicating the number of groups of processors, processors, or threads used in the four levels of parallelism. Namely, P_{site} denotes the parallelism over sites; P_{hamil} denotes the distributed parallelism over sub-Hamiltonians; T_{op} is for the joint shared memory parallelism over normal and complementary operators and symmetry sectors; and T_{dense} is for the thread-level parallelism in the dense matrix multiplications. The total number of CPU cores for a specific parallelism scheme is given by $N_{\text{core}} = P_{\text{site}}P_{\text{hamil}}T_{\text{op}}T_{\text{dense}}$.

The calculations were executed on nodes with 28-core Intel Cascadelake 8276 CPUs (2.20 GHz), made available via the Caltech high-performance computing facility. Each node has 56 CPU cores and 384 GB memory.

A. Mixed NC/CN approach

As discussed in Appendix B, in conventional DMRG implementations, there are two possible ways to write the left-right decomposition of the *ab initio* Hamiltonian at each site k. The NC scheme corresponds to an MPO with tensor dimensions that increase from left to right, while the CN scheme corresponds to an MPO with tensor dimensions that decrease from left to right.[33] Typically, efficient DMRG implementations use a mixed NC/CN approach,[11] where the NC decomposition is used for sites $k < K/2$ and the CN decomposition is used for sites $k \geq K/2$, which gives a significantly smaller "MPO" bond dimension. However, a transformation from the normal to complementary (two-index) operators is required near the middle site in this mixed NC/CN approach. The time complexity for this transformation is $O(K^4M^2)$. Additionally, for parallelism over sub-Hamiltonians, since we use different processor assignments for the NC and CN schemes, an extra reduction step for all the two-index complementary operators is required. The communication cost is $O(K^2M^2 \log P_{\text{hamil}})$. The extra computation and communication cost means that the middle site of the sweep is significantly more

expensive than the other sites. Consequently, for parallelism over sites, we consider only odd P_{site} and use a non-uniform division of the sweep ranges, so that the high cost of computation at the middle site (included in the sweep range of processor group $p = \lceil \frac{1}{2} P_{\text{site}} \rceil$) is amortized among all P_{site} groups of processors. In this work, we tested $P_{\text{site}} = 1, 3, 5$.

In Fig. 1 we show the distribution of MPO bond dimensions (for each MPO tensor) and the corresponding wall time cost at each site, for the NC and mixed NC/CN partitions of the benzene system. For a spin-adapted DMRG algorithm, the bond dimension of the MPO tensor at site k is $2 + 2K + 6k^2$ (blue dashed line), if the NC scheme is used without any optimization or additional simplifications. Using the symmetry conditions Eq. (B8), the bond dimension can be reduced to approximately $2+2K+2k^2$ (green dashed line), and the sudden decrease of the MPO bond dimension near the rightmost site in the figure is due to the removal of complementary operators with vanishing integrals. The mixed NC/CN approach gives a much better distribution of bond dimensions (black dashed line), with the maximal value $D = 5996$ appearing near the middle site of the test system. From Fig. 1 we can see that the time cost near the middle site of the mixed approach is approximately two times as large as that of the NC approach, but the mixed approach gives a much smaller total wall time per sweep (with $M = 4000$ and $P_{\text{hamil}} = 16$, $t_{\text{mixed}} = 13071$ sec) as compared to the NC approach ($t_{\text{nc}} = 19356$ sec), mainly due to the smaller MPO bond dimensions for $k \geq K/2$. The speed-up $t_{\text{nc}}/t_{\text{mixed}}$ is approximately 148% based on the data in Fig. 1. Due to this, subsequent calculations in this work all use the mixed approach.

B. Parallelism over dense matrix multiplication

As discussed in earlier studies,[29] using thread parallelism in the dense matrix multiplications is not very effective in DMRG, compared with the other parallel strategies. Table I shows that this is also true for our implementation of *ab initio* DMRG. For $M = 2500$ and 3000, parallelism schemes with $T_{\text{dense}} = 4$ are approximately 60% to 70% slower than the scheme with $T_{\text{dense}} = 1$. Therefore, for production calculations in this work, this level of parallelism was not utilized (i.e., we used only $T_{\text{op}} = 28$ and $T_{\text{dense}} = 1$).

TABLE I. Wall time per sweep (in seconds) in the benzene calculation for MPS bond dimension $M = 2500$ and 3000 using parallelism schemes with different T_{dense}.

parallelism scheme				Wall time per sweep (sec)	
P_{site}	P_{hamil}	T_{op}	T_{dense}	$M = 2500$	$M = 3000$
5	14	28	1	893	1291
5	14	7	4	1521	2106
5	7	14	4	1493	2079

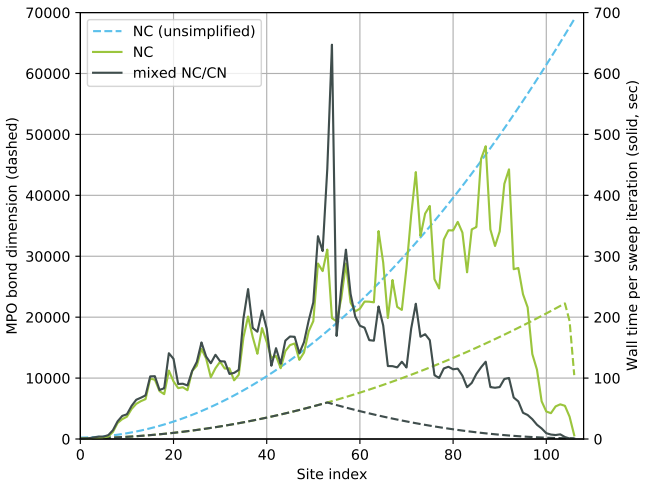

FIG. 1. MPO tensor bond dimensions (dashed lines) and wall time cost (solid lines) at each site for the NC and mixed NC/CN approaches for the benzene system. (Unsimplified refers to the bond dimension obtained without accounting for zero-integrals and symmetry conditions, given by $2 + 2K + 6k^2$). The performance data is from a $M = 4000$ calculation with the parallelism scheme $P_{\text{site}} = 1, P_{\text{hamil}} = 16, T_{\text{op}} = 28$, and $T_{\text{dense}} = 1$.

C. Parallelism over sites

It is sometimes argued that when parallelism over sites is used, the convergence of the DMRG energy as a function of the number of sweeps is slower than that of the standard DMRG approach, if the same sweep schedule is used.[34] In Fig. 2 we compare parallelism schemes with different P_{site} for MPS bond dimensions up to $M = 6000$ (data for $M = 6000$ with $P_{\text{site}} = 3$ and $P_{\text{hamil}} = 8$ could not be obtained due to memory constraints) for the benzene system. We can see that in our test system, convergence is only slightly affected by increasing P_{site} from 1 to 5. When five sweeps were performed for each M, the energy obtained from the last sweep for each M was almost the same with different P_{site}, up to $M = 5000$. Although we started the calculation from the same initial MPS (with a single canonical center) for different P_{site}, for $P_{\text{site}} = 3$ and 5 the initial MPS is re-canonicalized to introduce extra canonical centers. During this canonicalization step some low-weight single-state quantum numbers were discarded, which makes the $P_{\text{site}} = 3$ and 5 DMRG energy at $M = 2500$ (artificially) higher in Fig. 2.

To examine the effect of the dynamical connection sites, we have compared the estimated performance using dynamical, fixed and uniform connection sites in Fig. 3. For our test benzene system with 108 sites and $P_{\text{site}} = 5$, four connection sites are required. From the dotted lines in Fig. 3, we see selecting connection sites based on a uniform division of sweep ranges (namely, $K_{\text{conn}} = \{21, 43, 64, 86\}$) gives a large load imbalance among the five processors. At the last sweep, the longest processor task consumed 388% more time than the short-

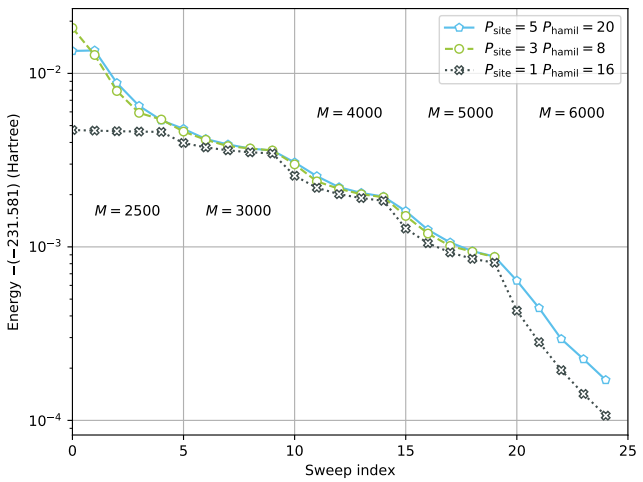

FIG. 2. Sweep energies for parallelism schemes with different P_{site} and different MPS bond dimensions M for the benzene system. For each M, five sweeps are performed.

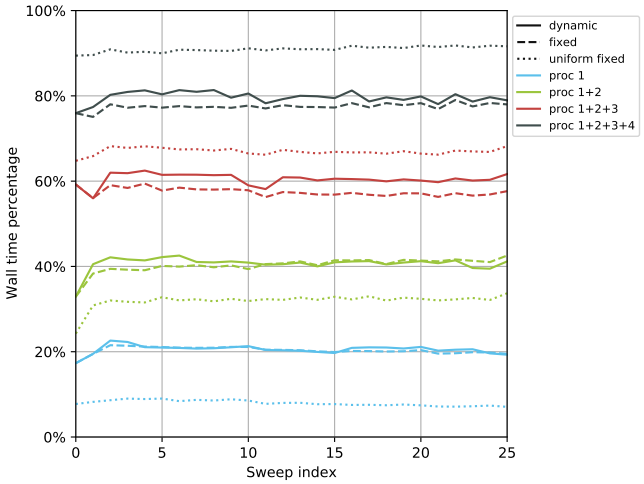

FIG. 3. Estimated wall time per processor for parallelism over sites (as a percentage of the sum of wall times for all processors), when the connection centers are dynamically adjusted (solid lines), fixed (dashed lines), uniformly distributed (dotted lines). The performance data is from the $P_{\text{site}} = 5$ and $P_{\text{hamil}} = 20$ benzene calculation with the MPS bond dimension increasing from $M = 2500$ to $M = 6000$.

est processor task, which can be mainly attributed to the highly non-uniform distribution of computational effort among sites (see solid black line in Fig. 1). In this work, we found that $K_{\text{conn}} = \{33, 49, 57, 73\}$ (obtained from using dynamical connection sites for small bond dimensions) gave much better performance. This corresponds to the dashed lines in Fig. 3. With this fixed set of connection sites, the longest task consumed 53% more time than the shortest task. If we allow the set of connection sites to be dynamically adjusted between the sweeps, we end up with a slightly altered set of connection sites $K_{\text{conn}} = \{33, 48, 59, 74\}$. Using this, in the last sweep the longest task then consumed only 26% more time than the shortest task.

D. Parallel Scaling

In Table II we list the average wall time per sweep with MPS bond dimensions from $M = 2500$ to $M = 6000$ for the benzene system, when parallelism schemes with different P_{site} and P_{hamil} are used. The speed-up relative to the $P_{\text{site}} = 1$ and $P_{\text{hamil}} = 16$ ($N_{\text{core}} = 448$) case is plotted in Fig. 4.

In Fig. 4 we see that, when $P_{\text{site}} = 3$, increasing P_{hamil} from 12 to 18 only reduces the wall time slightly, while nearly ideal speed-up is observed for different $(P_{\text{site}}, P_{\text{hamil}})$ when increasing from (1, 16) to (3, 12) and from (3, 12) to (5, 14). This illustrates that a combination of different DMRG parallelism strategies is essential to achieve good scaling across thousands of CPU cores. The better-than-ideal speed-up for $(P_{\text{site}}, P_{\text{hamil}})$ when increasing from (1, 16) to (3, 8) is also due to the change of parallelism strategy. Ideally, the $P_{\text{site}} = P_{\text{hamil}} = 1$ case should be used as the reference point for computing the speed-up. However, this is not feasible in our test

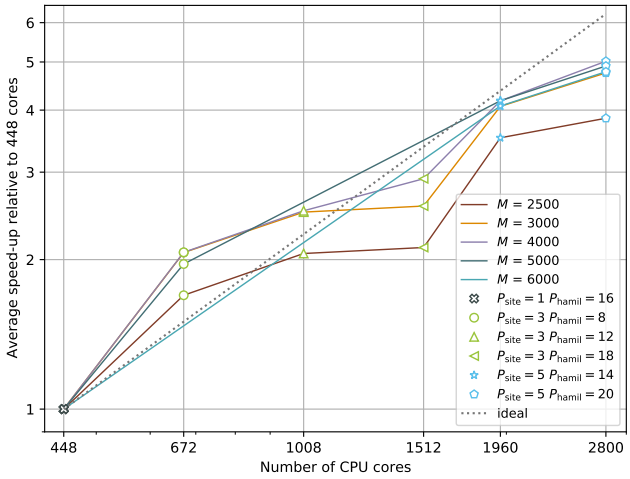

FIG. 4. Speed-up of average wall time per sweep relative to the $N_{\text{core}} = 448$ case for different MPS bond dimensions using parallelism schemes with different P_{site} and P_{hamil}. $T_{\text{op}} = 28$ and $T_{\text{dense}} = 1$ are used for all parallelism schemes.

system due to the large maximum MPO bond dimension $D = 5996$ and when using a large MPS bond dimension $M = 6000$. Here, we need $P_{\text{hamil}} \geq 10$ to ensure that the memory cost per node is less than 384 GB. This is an important reason to use larger P_{hamil} rather than P_{site} in certain systems, since increasing P_{site} does not reduce the memory cost per processor group. Finally, we note that the speed-up for $M = 2500$ appears to be significantly less than the other cases with the larger M. This is likely related to the fact that for the $P_{\text{site}} = 3$ and $P_{\text{site}} = 5$ cases, an initial $M = 2500$ MPS with an

TABLE II. Average wall time per sweep (in seconds) of the benzene calculation for different MPS bond dimensions using parallelism schemes with different P_{site} and P_{hamil}. $T_{\text{op}} = 28$ and $T_{\text{dense}} = 1$ were used for all parallelism schemes.

parallelism scheme		N_{core}	Average wall time per sweep (sec)				
P_{site}	P_{hamil}		$M = 2500$	$M = 3000$	$M = 4000$	$M = 5000$	$M = 6000$
1	16	448	3145	5253	12740	22212	35451
3	8	672	1855	2542	6158	11335	
	12	1008	1529	2107	5079		
	18	1512	1487	2049	4379		
5	14	1960	894	1291	3051	5317	8696
	20	2800	816	1105	2539	4526	7419

TABLE III. Average wall time per sweep (in seconds) of the FeMo cofactor calculation for different MPS bond dimensions using parallelism schemes with different P_{site} and P_{hamil}. $T_{\text{op}} = 28$ and $T_{\text{dense}} = 1$ were used for all parallelism schemes.

parallelism scheme		N_{core}	Average wall time per sweep (sec)		
P_{site}	P_{hamil}		$M = 2000$	$M = 2500$	$M = 3000$
1	16	448	10596	19464	50677
3	8	672	6380	12191	31496
5	16	2240	3262	5156	12499

artificially higher energy was used (see Fig. 2).

For the largest calculation considered in this work with $P_{\text{site}} = 5, P_{\text{hamil}} = 20$ and $M = 6000$ for the benzene system, the average communication and idle time among the P_{hamil} processors constituted approximately 15% of the total wall time for each group of P_{hamil} processors and the average idle time among the P_{site} groups of processors was approximately 10% of the total wall time. The Davidson step (including communication) constituted 60% to 70% of the total wall time for each processor. Reading/writing disk files cost approximately 5% of the total wall time.

Finally, in Table III we show that a similar scaling can be observed for the FeMo cofactor system. When increasing $(P_{\text{site}}, P_{\text{hamil}})$ from $(1, 16)$ to $(5, 16)$, for a sufficiently large MPS bond dimension ($M = 3000$) we obtain a speed-up of 4.05, which is close to the ideal speed-up (5). Note that the worse-than-cubic scaling with respect to M for the $M = 2500$ and $M = 3000$ cases shown in Table III is mainly due to the difference in the Davidson convergence criteria used for different M.

IV. CONCLUSIONS

In this work, we introduced a modification of the conventional strategy for distributed memory parallelism in *ab initio* DMRG algorithms that reduces the computation to the manipulation of independent sub-Hamiltonians, together with a small wavefunction communication step. This formulation thus combines the conceptual advantages of the sum of sub-Hamiltonians approach introduced in earlier work, with the greater parallelizability and lower prefactor of the conventional distributed memory DMRG algorithm. In addition to this, we carried out a comprehensive examination and implementation of four other sources of parallelism in DMRG, introducing techniques for load balancing via dynamic connection sites in site-based parallelism, and collapsing tasks to maximize thread efficiency in the shared memory parallelism. Finally, we showed that the combination of different DMRG parallelism strategies using both distributed and shared memory models was essential to achieve near-ideal speed-ups for a benchmark calculation with 108 orbitals and a DMRG bond dimension of $M = 6000$, scaling from 448 to 2800 CPU cores. The DMRG implementation in the Block2 code used in this work is open-source and can be freely obtained.[51]

ACKNOWLEDGMENTS

This work was supported by the US National Science Foundation (NSF) via grant CHE-1665333. HZ thanks Seunghoon Lee for providing the integrals and reference DMRG outputs for the benzene system, and Henrik R. Larsson, Zhi-Hao Cui and Tianyu Zhu for helpful discussions. The computations presented in this work were conducted on the Caltech High Performance Cluster, partially supported by a grant from the Gordon and Betty Moore Foundation.

Appendix A: The serial DMRG algorithm

To establish notation for the DMRG algorithm, consider a quantum lattice system with K sites. Each site is associated with a Hilbert space spanned by a site-basis $\{|n_k\rangle\}$. A complete basis of the system Hilbert space can be defined as the tensor product of K site-bases

$$\{|n_1\ n_2\ \cdots\ n_K\rangle\} = \{|n_1\rangle \otimes |n_2\rangle \otimes \cdots \otimes |n_K\rangle\} \quad \text{(A1)}$$

The goal of the DMRG algorithm is to optimize a variational wavefunction in this Hilbert space, whose amplitudes can be written as a product of matrices

$$|\Psi\rangle = \sum_{\{n\}} \mathbf{A}[1]^{n_1}\mathbf{A}[2]^{n_2}\cdots\mathbf{A}[K]^{n_K}|n_1\ n_2\ \cdots\ n_K\rangle \quad \text{(A2)}$$

where each $\mathbf{A}[k]^{n_k}$ $(k = 2,\cdots,K-1)$ is an $M\times M$ matrix, and the leftmost and rightmost matrices are $1\times M$ and $M\times 1$ vectors, respectively. The dimension M is known as the bond-dimension of the MPS $|\Psi\rangle$.

Within the MPS ansatz, variational minimization of the energy, formally written as

$$E_0 = \min_{|\Psi\rangle}\frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \quad \text{(A3)}$$

where $\hat{H}$ is the system Hamiltonian and E_0 is the ground-state energy, can be performed iteratively by optimizing the parameters of a single matrix at a time in the MPS, while the parameters in the remaining matrices are kept constant. This corresponds to the 1-site DMRG algorithm. A common variant, designed to improve the ability to escape local minima, optimizes a single larger matrix $A[k]^{n_k n_{k+1}}$ that describes the variational space of 2-sites at a time. This formally takes one outside of the single-site MPS variational space and thus the solution must be decimated back to the standard MPS form. This corresponds to the 2-site DMRG algorithm. The same idea can be generalized to p sites, but in this work we mainly consider the $p=2$ case.

The iterative process in a serial DMRG algorithm is structured as a series of *sweeps* along a fixed one-dimensional ordering of the K sites. Each sweep alternates between the forward and backward directions, consisting of $K+1-p$ *sweep iterations*. In the k-th $(k=1,\cdots,K+1-p)$ sweep iteration of a forward sweep, the parameters in the current matrix being optimized (associated with d adjacent sites, $\mathbf{A}[k]^{n_k\cdots n_{k+d-1}}$) are updated, while in a backward sweep the matrices are updated in reverse order. The lattice can then be conveniently divided into $2+d$ *blocks* (or sets of sites) $\{L_{k-1}, S_k, \cdots, S_{k+d-1}, R_{k+d}\}$ in the k-th sweep iteration (of a forward sweep, for example): a left block (or the *system*) L_{k-1} for sites $1,\cdots,k-1$; the individual sites whose matrices are being optimized S_k ... S_{k+d-1}; and the right block (or the *environment*) R_{k+d} for sites $k+d,\cdots,K$ (see Fig. 5).

In each sweep iteration, we consider a left-right decomposition of the system Hamiltonian as the sum of tensor products of operators defined in blocks L_k and R_{k+1}

$$\hat{H}[k] = \hat{H}^{L_k}\otimes\hat{1}^{R_{k+1}} + \hat{1}^{L_k}\otimes\hat{H}^{R_{k+1}} + \sum_i \hat{h}_i^{L_k}\hat{h}_i^{R_{k+1}} \quad \text{(A4)}$$

where a bipartition of the lattice $\{L_k, R_{k+1}\}$ has been used. A convenient way to construct this left-right decomposition for any k is to first write the system Hamiltonian in a so-called MPO form

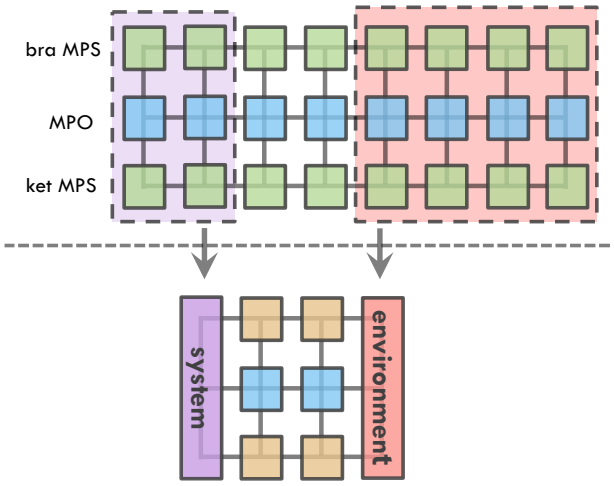

FIG. 5. The left block (system), right block (environment) and the individual sites being optimized in a given sweep iteration of the 2-site DMRG algorithm.

$$\hat{H} = \sum_{\{n,n'\}} \mathbf{W}[1]^{n_1 n_1'}\mathbf{W}[2]^{n_2 n_2'}\cdots\mathbf{W}[K]^{n_K n_K'} \times |n_1\ n_2\ \cdots\ n_K\rangle\langle n_1'\ n_2'\ \cdots\ n_K'| \quad \text{(A5)}$$

where each $\mathbf{W}[k]^{n_k n_k'}$ $(k=2,\cdots,K-1)$ is a $D\times D'$ matrix, and the leftmost and rightmost matrices are $1\times D'$ and $D\times 1$ vectors, respectively. The maximal dimension D among these matrices will be called the bond-dimension of the MPO.

The left-right decomposition of the MPS can be defined as (in 2-site DMRG, for example)

$$|\Psi[k]\rangle = \sum_{\alpha_{k-1}\alpha_k\alpha_{k+1},n_k n_{k+1}} A[k]^{n_k}_{\alpha_{k-1}\alpha_k}A[k+1]^{n_{k+1}}_{\alpha_k\alpha_{k+1}} \times |\alpha^L_{k-1}\rangle\otimes|n_k\ n_{k+1}\rangle\otimes|\alpha^R_{k+1}\rangle \quad \text{(A6)}$$

where the left and right renormalized basis vectors are

$$\begin{aligned}|\alpha_k^L\rangle &= \sum_{\{n_1\cdots n_k\}}\Big[\mathbf{A}[1]^{n_1}\cdots\mathbf{A}[k]^{n_k}\Big]_{\alpha_k}|n_1\ \cdots\ n_k\rangle \\ |\alpha_k^R\rangle &= \sum_{\{n_{k+1}\cdots n_K\}}\Big[\mathbf{A}[k+1]^{n_{k+1}}\cdots\mathbf{A}[K]^{n_K}\Big]_{\alpha_k} \\ &\qquad\times|n_{k+1}\ \cdots\ n_K\rangle\end{aligned} \quad \text{(A7)}$$

Using the MPO form, the decomposition Eq. (A4) can be constructed as

$$\hat{H}[k] = \sum_{\beta_k}\hat{H}[k]^L_{\beta_k}\otimes\hat{H}[k]^R_{\beta_k} \quad \text{(A8)}$$

where

$$\hat{H}[k]^L_{\beta_k} = \sum_{\{n_1\cdots n_k, n'_1\cdots n'_k\}} \Big[\mathbf{W}[1]^{n_1 n'_1}\cdots\mathbf{W}[k]^{n_k n'_k}\Big]_{\beta_k} \times |n_1\cdots n_k\rangle\langle n'_1\cdots n'_k|$$
$$\hat{H}[k]^R_{\beta_k} = \sum_{\{n_{k+1}\cdots n_K, n'_{k+1}\cdots n'_K\}} \Big[\mathbf{W}[k+1]^{n_{k+1}n'_{k+1}}\cdots \times \mathbf{W}[K]^{n_K n'_K}\Big]_{\beta_k} |n_{k+1}\cdots n_K\rangle\langle n'_{k+1}\cdots n'_K| \quad \text{(A9)}$$

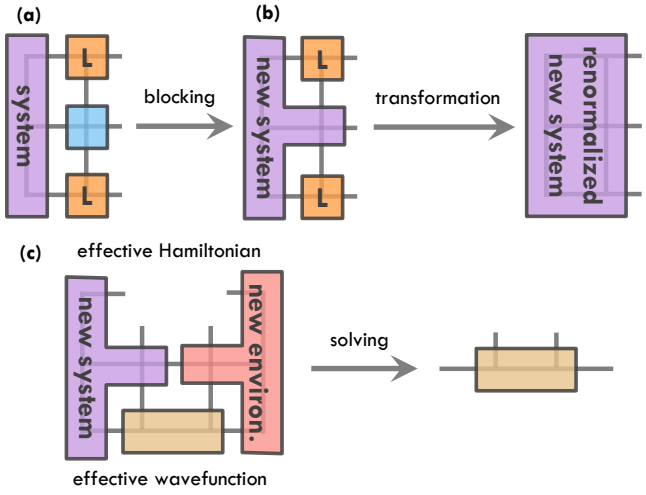

FIG. 6. The (a) blocking, (b) transformation, and (c) solving steps in each sweep iteration of the 2-site DMRG algorithm.[33]

Each sweep iteration of the 2-site DMRG algorithm is divided into three main steps (see Fig. 6):[6]

(i) blocking, where we compute the matrix representation of $\hat{H}[k]^L_{\beta_k}$ and $\hat{H}[k]^R_{\beta_k}$ Eq. (A9)) in bases $|\alpha^L_{k-1}\ n_k\rangle$ and $|n_{k+1}\ \alpha^R_{k+1}\rangle$ from the renormalized operators represented in bases $|\alpha^L_{k-1}\rangle$ and $|\alpha^R_{k+1}\rangle$, respectively

$$\begin{aligned}
&\langle\alpha^L_{k-1}\ n_k|\hat{H}[k]^L_{\beta_k}|\alpha'^L_{k-1}\ n'_k\rangle \\
&= \sum_{\beta_{k-1}} W[k]^{n_k n'_k}_{\beta_{k-1}\beta_k}\langle\alpha^L_{k-1}|\hat{H}[k-1]^L_{\beta_{k-1}}|\alpha'^L_{k-1}\rangle \\
&\langle n_{k+1}\ \alpha^R_{k+1}|\hat{H}[k]^R_{\beta_k}|n'_{k+1}\ \alpha'^R_{k+1}\rangle \\
&= \sum_{\beta_{k+1}} W[k+1]^{n_{k+1}n'_{k+1}}_{\beta_k\beta_{k+1}}\langle\alpha^R_{k+1}|\hat{H}[k+1]^R_{\beta_{k+1}}|\alpha'^R_{k+1}\rangle
\end{aligned} \quad \text{(A10)}$$

(ii) solving, where we update the wavefunction in the renormalized basis $|\alpha^L_{k-1}\ n_k\rangle\otimes|n_{k+1}\ \alpha^R_{k+1}\rangle$ (Eq. (A6)), given by

$$\Psi[k]^{\text{eff}}_{\alpha_{k-1}n_k,n_{k+1}\alpha_{k+1}} = \sum_{\alpha_k} A[k]^{n_k}_{\alpha_{k-1}\alpha_k} A[k+1]^{n_{k+1}}_{\alpha_k\alpha_{k+1}} \quad \text{(A11)}$$

by solving the eigenvalue problem

$$\mathbf{H}[k]^{\text{eff}}\mathbf{\Psi}[k]^{\text{eff}} = E[k]\mathbf{\Psi}[k]^{\text{eff}} \quad \text{(A12)}$$

where the matrix elements of the effective Hamiltonian $\mathbf{H}[k]^{\text{eff}}$ are given by

$$\begin{aligned}
&H[k]^{\text{eff}}_{\alpha_{k-1}n_k,n_{k+1}\alpha_{k+1};n'_{k+1}\alpha'_{k+1},\alpha'_{k-1}n'_k} \\
&= \sum_{\beta_k}\langle\alpha^L_{k-1}\ n_k|\hat{H}[k]^L_{\beta_k}|\alpha'^L_{k-1}\ n'_k\rangle \\
&\quad\times\langle n_{k+1}\ \alpha^R_{k+1}|\hat{H}[k]^R_{\beta_k}|n'_{k+1}\ \alpha'^R_{k+1}\rangle
\end{aligned} \quad \text{(A13)}$$

Since the Hamiltonian is sparse, the eigenvalue problem is normally solved using an iterative method such as the Davidson algorithm.[52]

(iii) decimation and transformation. Once the optimized wavefunction $\mathbf{\Psi}[k]^{\text{eff}}$ is determined, the new $\mathbf{A}[k]$ and $\mathbf{A}[k+1]$ can be found by decomposing the wavefunction using the density matrix, or via a singular value decomposition (SVD). After the decomposition, the matrix dimensions of $\mathbf{A}[k]$ and $\mathbf{A}[k+1]$ are truncated to bond dimension M by discarding small singular values or eigenvalues. The truncated $\mathbf{A}[k]$ and $\mathbf{A}[k+1]$ are then used to construct new renormalized bases $|\alpha^L_k\rangle$ and $|\alpha^R_k\rangle$, in a forward and backward sweep iteration, respectively, as

$$\begin{aligned}
|\alpha^L_k\rangle &= \sum_{\alpha_{k-1}} A[k]^{n_k}_{\alpha_{k-1}\alpha_k}|\alpha^L_{k-1}\ n_k\rangle \\
|\alpha^R_k\rangle &= \sum_{\alpha_{k+1}} A[k+1]^{n_{k+1}}_{\alpha_k\alpha_{k+1}}|n_{k+1}\ \alpha^R_{k+1}\rangle
\end{aligned} \quad \text{(A14)}$$

The operators formed in the blocking step (Eq. (A10)) are also transformed to the new renormalized basis

$$\begin{aligned}
&\langle\alpha^L_k|\hat{H}[k]^L_{\beta_k}|\alpha'^L_k\rangle \\
&= \sum_{\alpha_{k-1}n_k;\alpha'_{k-1}n'_k} A[k]^{n_k}_{\alpha_{k-1}\alpha_k}A[k]^{n'_k}_{\alpha'_{k-1}\alpha'_k} \\
&\quad\times\langle\alpha^L_{k-1}\ n_k|\hat{H}[k]^L_{\beta_k}|\alpha'^L_{k-1}\ n'_k\rangle \\
&\langle\alpha^R_k|\hat{H}[k]^R_{\beta_k}|\alpha'^R_k\rangle \\
&= \sum_{n_{k+1}\alpha_{k+1};n'_{k+1}\alpha'_{k+1}} A[k+1]^{n_{k+1}}_{\alpha_k\alpha_{k+1}}A[k+1]^{n'_{k+1}}_{\alpha'_k\alpha'_{k+1}} \\
&\quad\times\langle n_{k+1}\ \alpha^R_{k+1}|\hat{H}[k]^R_{\beta_k}|n'_{k+1}\ \alpha'^R_{k+1}\rangle
\end{aligned} \quad \text{(A15)}$$

Appendix B: Notation for SU(2) spin-adapted ab initio DMRG

For the *ab initio* DMRG implemented in this work, we associate each site k ($k = 1, 2, \cdots, K$) with a spatial orbital. The *ab initio* Hamiltonian is written as[32]

$$\hat{H} = \sum_{ij,\sigma} t_{ij,\sigma}\ a^\dagger_{i\sigma}a_{j\sigma} + \frac{1}{2}\sum_{ijkl,\sigma\sigma'} v_{ijkl,\sigma\sigma'}\ a^\dagger_{i\sigma}a^\dagger_{k\sigma'}a_{l\sigma'}a_{j\sigma} \quad \text{(B1)}$$

where

$$t_{ij,\sigma} = \int \mathrm{d}\mathbf{x}\ \phi^*_{i\sigma}(\mathbf{x}) \left(-\frac{1}{2}\nabla^2 - \sum_a \frac{Z_a}{r_a}\right) \phi_{j\sigma}(\mathbf{x})$$
$$v_{ijkl,\sigma\sigma'} = \int \mathrm{d}\mathbf{x}_1 \mathrm{d}\mathbf{x}_2\ \frac{\phi^*_{i\sigma}(\mathbf{x}_1)\phi^*_{k\sigma'}(\mathbf{x}_2)\phi_{l\sigma'}(\mathbf{x}_2)\phi_{j\sigma}(\mathbf{x}_1)}{r_{12}} \tag{B2}$$

with the following symmetry conditions

$$\begin{aligned} t_{ij,\sigma} &= t_{ji,\sigma} \\ v_{ijkl,\sigma\sigma'} &= v_{jikl,\sigma\sigma'} = v_{ijlk,\sigma\sigma'} = v_{klij,\sigma'\sigma} \end{aligned} \tag{B3}$$

With SU(2) spin symmetry we additionally have[42]

$$\begin{aligned} t_{ij} &= t_{ij,\alpha} = t_{ij,\beta} \\ v_{ijkl} &= v_{ijkl,\alpha\alpha} = v_{ijkl,\alpha\beta} = v_{ijkl,\beta\alpha} = v_{ijkl,\beta\beta} \end{aligned} \tag{B4}$$

In conventional *ab initio* DMRG, the left-right decomposition of the Hamiltonian (Eq. (A4)) is written in terms of *normal* and *complementary* operators.[3] One can choose to use two-index complementary operators only with the right block (the Normal/Complementary (NC) partition) or only with the left block (the Complementary/Normal (CN) partition). The SU(2) spin-adapted left-right decomposition of the Hamiltonian using the NC and CN partition is respectively[42]

$$\begin{aligned} \hat{H}[k]^{\mathrm{NC}[0]} &= \hat{H}^{L[0]} \otimes_{[0]} \hat{1}^{R[0]} + \hat{1}^{L[0]} \otimes_{[0]} \hat{H}^{R[0]} \\ &+ 2\sum_{i\in L} \left(a_i^{\dagger[\frac{1}{2}]} \otimes_{[0]} \hat{R}_i^{R[\frac{1}{2}]} + a_i^{[\frac{1}{2}]} \otimes_{[0]} \hat{R}_i^{R\dagger[\frac{1}{2}]}\right) \\ &+ 2\sum_{i\in R} \left(\hat{R}_i^{L\dagger[\frac{1}{2}]} \otimes_{[0]} a_i^{[\frac{1}{2}]} + \hat{R}_i^{L[\frac{1}{2}]} \otimes_{[0]} a_i^{\dagger[\frac{1}{2}]}\right) \\ &- \frac{1}{2}\sum_{ij\in L} \left(\hat{A}_{ij}^{[0]} \otimes_{[0]} \hat{P}_{ij}^{R[0]} + \sqrt{3}\ \hat{A}_{ij}^{[1]} \otimes_{[0]} \hat{P}_{ij}^{R[1]} \right. \\ &\left. + \hat{A}_{ij}^{\dagger[0]} \otimes_{[0]} \hat{P}_{ij}^{R\dagger[0]} + \sqrt{3}\ \hat{A}_{ij}^{\dagger[1]} \otimes_{[0]} \hat{P}_{ij}^{R\dagger[1]}\right) \\ &+ \sum_{ij\in L} \left(\hat{B}_{ij}^{[0]} \otimes_{[0]} \hat{Q}_{ij}^{R[0]} + \sqrt{3}\ \hat{B}_{ij}^{[1]} \otimes_{[0]} \hat{Q}_{ij}^{R[1]}\right) \end{aligned} \tag{B5}$$

and

$$\begin{aligned} \hat{H}[k]^{\mathrm{CN}[0]} &= \hat{H}^{L[0]} \otimes_{[0]} \hat{1}^{R[0]} + \hat{1}^{L[0]} \otimes_{[0]} \hat{H}^{R[0]} \\ &+ 2\sum_{i\in L} \left(a_i^{\dagger[\frac{1}{2}]} \otimes_{[0]} \hat{R}_i^{R[\frac{1}{2}]} + a_i^{[\frac{1}{2}]} \otimes_{[0]} \hat{R}_i^{R\dagger[\frac{1}{2}]}\right) \\ &+ 2\sum_{i\in R} \left(\hat{R}_i^{L\dagger[\frac{1}{2}]} \otimes_{[0]} a_i^{[\frac{1}{2}]} + \hat{R}_i^{L[\frac{1}{2}]} \otimes_{[0]} a_i^{\dagger[\frac{1}{2}]}\right) \\ &- \frac{1}{2}\sum_{ij\in R} \left(\hat{P}_{ij}^{L[0]} \otimes_{[0]} \hat{A}_{ij}^{[0]} + \sqrt{3}\ \hat{P}_{ij}^{L[1]} \otimes_{[0]} \hat{A}_{ij}^{[1]} \right. \\ &\left. + \hat{P}_{ij}^{L\dagger[0]} \otimes_{[0]} \hat{A}_{ij}^{\dagger[0]} + \sqrt{3}\ \hat{P}_{ij}^{L\dagger[1]} \otimes_{[0]} \hat{A}_{ij}^{\dagger[1]}\right) \\ &+ \sum_{ij\in R} \left(\hat{Q}_{ij}^{L[0]} \otimes_{[0]} \hat{B}_{ij}^{[0]} + \sqrt{3}\ \hat{Q}_{ij}^{L[1]} \otimes_{[0]} \hat{B}_{ij}^{[1]}\right) \end{aligned} \tag{B6}$$

where the superscript and subscript $[S]$ are used to indicate the total spin quantum number for the spin tensor operator and the resulting spin tensor operator obtained from the tensor product, respectively, and the block Hamiltonian $\hat{H}^{L/R[0]}$, normal operators $\hat{A}_{ij}^{[S]}$, $\hat{B}_{ij}^{[S]}$, and complementary operators $\hat{R}_i^{L/R[\frac{1}{2}]}$, $\hat{P}_{ij}^{L/R[S]}$, $\hat{Q}_{ij}^{L/R[S]}$ are defined by

$$\begin{aligned} \hat{R}_i^{L/R[\frac{1}{2}]} &= \frac{\sqrt{2}}{4}\sum_{j\in L/R} t_{ij}\ a_j^{[\frac{1}{2}]} \\ &\quad + \sum_{jkl\in L/R} v_{ijkl}\Big(a_k^{\dagger[\frac{1}{2}]} \otimes_{[0]} a_l^{[\frac{1}{2}]}\Big) \otimes_{[\frac{1}{2}]} a_j^{[\frac{1}{2}]}, \\ \hat{A}_{ij}^{[0/1]} &= a_i^{\dagger[\frac{1}{2}]} \otimes_{[0/1]} a_j^{\dagger[\frac{1}{2}]}, \\ \hat{B}_{ij}^{[0/1]} &= a_i^{\dagger[\frac{1}{2}]} \otimes_{[0/1]} a_j^{[\frac{1}{2}]}, \\ \hat{P}_{ik}^{L/R[0/1]} &= \sum_{jl\in L/R} v_{ijkl}\ a_j^{[\frac{1}{2}]} \otimes_{[0/1]} a_l^{[\frac{1}{2}]}, \\ \hat{Q}_{ij}^{L/R[0]} &= \sum_{kl\in L/R} \big(2v_{ijkl} - v_{ilkj}\big) a_k^{\dagger[\frac{1}{2}]} \otimes_{[0]} a_l^{[\frac{1}{2}]}, \\ \hat{Q}_{ij}^{L/R[1]} &= \sum_{kl\in L/R} v_{ilkj}\ a_k^{\dagger[\frac{1}{2}]} \otimes_{[1]} a_l^{[\frac{1}{2}]} \end{aligned} \tag{B7}$$

with the following symmetry conditions (when $i \neq j$)[11]

$$\begin{aligned} \hat{A}_{ij}^{[S]} &= (-1)^S \hat{A}_{ji}^{[S]} \\ \hat{B}_{ij}^{[S]} &= (-1)^S \big(\hat{B}_{ji}^{[S]}\big)^\dagger \\ \hat{P}_{ij}^{[S]} &= (-1)^S \hat{P}_{ji}^{[S]} \\ \hat{Q}_{ij}^{[S]} &= (-1)^S \big(\hat{Q}_{ji}^{[S]}\big)^\dagger \end{aligned} \tag{B8}$$

The corresponding MPO for the NC and CN partitions can be constructed based on the blocking formulae for the spin tensor operators, and Eq. (B5) and Eq. (B6), respectively. Since these operators have at most two spatial orbital indices, the MPO bond dimension $D \sim K^2$. The blocking formulae explicitly yield only the non-zero elements of the MPO, and thus using the blocking formulae in the DMRG algorithm can be viewed as implementing sparse tensor contraction with the MPO.

Alternatively, there are procedures to automatically construct the elements of the MPO tensors by matrix decomposition (and other algorithms) simply given the list of two-electron integrals. Examples of these automated MPO construction approaches are the fork-merge approach,[40] the SVD approach,[33] the delinearization approach,[41] and the bipartite approach.[47] Note that some of these procedures work with a dense matrix representation of the MPO tensors (even if the matrices have exact zeros). As discussed in the main text, the sum of sub-Hamiltonians allows for the correct scaling of implementations which use such MPO construction techniques without explicit implementation of sparse tensor algebra. Thus the strategies in this work, when used with automated MPO construction techniques, achieve both the

correct serial cost as well as have a low communication overhead for parallel scaling.

[1]White, S. R. Density matrix formulation for quantum renormalization groups. *Physical review letters* **1992**, *69*, 2863.

[2]White, S. R. Density-matrix algorithms for quantum renormalization groups. *Physical Review B* **1993**, *48*, 10345.

[3]White, S. R.; Martin, R. L. Ab initio quantum chemistry using the density matrix renormalization group. *The Journal of chemical physics* **1999**, *110*, 4127–4130.

[4]Mitrushenkov, A. O.; Fano, G.; Ortolani, F.; Linguerri, R.; Palmieri, P. Quantum chemistry using the density matrix renormalization group. *The Journal of Chemical Physics* **2001**, *115*, 6815–6821.

[5]Mitrushenkov, A.; Linguerri, R.; Palmieri, P.; Fano, G. Quantum chemistry using the density matrix renormalization group II. *The Journal of chemical physics* **2003**, *119*, 4148–4158.

[6]Chan, G. K.-L.; Head-Gordon, M. Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group. *The Journal of chemical physics* **2002**, *116*, 4462–4476.

[7]Legeza, Ö.; Röder, J.; Hess, B. QC-DMRG study of the ionic-neutral curve crossing of LiF. *Molecular Physics* **2003**, *101*, 2019–2028.

[8]Chan, G. K.-L.; Kállay, M.; Gauss, J. State-of-the-art density matrix renormalization group and coupled cluster theory studies of the nitrogen binding curve. *The Journal of chemical physics* **2004**, *121*, 6110–6116.

[9]Moritz, G.; Reiher, M. Construction of environment states in quantum-chemical density-matrix renormalization group calculations. *The Journal of chemical physics* **2006**, *124*, 034103.

[10]Hachmann, J.; Cardoen, W.; Chan, G. K.-L. Multireference correlation in long molecules with the quadratic scaling density matrix renormalization group. *The Journal of chemical physics* **2006**, *125*, 144101.

[11]Kurashige, Y.; Yanai, T. High-performance ab initio density matrix renormalization group method: Applicability to large-scale multireference problems for metal compounds. *The Journal of chemical physics* **2009**, *130*, 234114.

[12]Marti, K. H.; Reiher, M. New electron correlation theories for transition metal chemistry. *Physical Chemistry Chemical Physics* **2011**, *13*, 6750–6759.

[13]Chan, G. K.-L.; Sharma, S. The density matrix renormalization group in quantum chemistry. *Annual review of physical chemistry* **2011**, *62*, 465–481.

[14]Fertitta, E.; Paulus, B.; Barcza, G.; Legeza, Ö. Investigation of metal–insulator-like transition through the ab initio density matrix renormalization group approach. *Physical Review B* **2014**, *90*, 245129.

[15]Olivares-Amaya, R.; Hu, W.; Nakatani, N.; Sharma, S.; Yang, J.; Chan, G. K.-L. The ab-initio density matrix renormalization group in practice. *The Journal of chemical physics* **2015**, *142*, 034102.

[16]Holmes, A. A.; Tubman, N. M.; Umrigar, C. Heat-bath configuration interaction: An efficient selected configuration interaction algorithm inspired by heat-bath sampling. *Journal of chemical theory and computation* **2016**, *12*, 3674–3680.

[17]Sharma, S.; Holmes, A. A.; Jeanmairet, G.; Alavi, A.; Umrigar, C. J. Semistochastic heat-bath configuration interaction method: Selected configuration interaction with semistochastic perturbation theory. *Journal of chemical theory and computation* **2017**, *13*, 1595–1604.

[18]Booth, G. H.; Thom, A. J.; Alavi, A. Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space. *The Journal of chemical physics* **2009**, *131*, 054106.

[19]Blunt, N.; Booth, G. H.; Alavi, A. Density matrices in full configuration interaction quantum Monte Carlo: Excited states, transition dipole moments, and parallel distribution. *The Journal of chemical physics* **2017**, *146*, 244105.

[20]Eriksen, J. J.; Anderson, T. A.; Deustua, J. E.; Ghanem, K.; Hait, D.; Hoffmann, M. R.; Lee, S.; Levine, D. S.; Magoulas, I.; Shen, J., et al. The ground state electronic energy of benzene. *The journal of physical chemistry letters* **2020**, *11*, 8922–8929.

[21]Kurashige, Y.; Yanai, T. Second-order perturbation theory with a density matrix renormalization group self-consistent field reference function: Theory and application to the study of chromium dimer. *The Journal of chemical physics* **2011**, *135*, 094104.

[22]Guo, S.; Li, Z.; Chan, G. K.-L. Communication: An efficient stochastic algorithm for the perturbative density matrix renormalization group in large active spaces. *The Journal of chemical physics* **2018**, *148*, 221104.

[23]Marti, K. H.; Ondík, I. M.; Moritz, G.; Reiher, M. Density matrix renormalization group calculations on relative energies of transition metal complexes and clusters. *The Journal of chemical physics* **2008**, *128*, 014104.

[24]Kurashige, Y.; Chan, G. K.-L.; Yanai, T. Entangled quantum electronic wavefunctions of the Mn 4 CaO 5 cluster in photosystem II. *Nature chemistry* **2013**, *5*, 660–666.

[25]Kurashige, Y.; Chalupskỳ, J.; Lan, T. N.; Yanai, T. Complete active space second-order perturbation theory with cumulant approximation for extended active-space wavefunction from density matrix renormalization group. *The Journal of chemical physics* **2014**, *141*, 174111.

[26]Chalupsky, J.; Rokob, T. A.; Kurashige, Y.; Yanai, T.; Solomon, E. I.; Rulisek, L.; Srnec, M. Reactivity of the binuclear non-heme iron active site of Δ9 desaturase studied by large-scale multireference ab initio calculations. *Journal of the American Chemical Society* **2014**, *136*, 15977–15991.

[27]Sharma, S.; Sivalingam, K.; Neese, F.; Chan, G. K.-L. Low-energy spectrum of iron–sulfur clusters directly from many-particle quantum mechanics. *Nature chemistry* **2014**, *6*, 927–933.

[28]Li, Z.; Guo, S.; Sun, Q.; Chan, G. K.-L. Electronic landscape of the P-cluster of nitrogenase as revealed through many-electron quantum wavefunction simulations. *Nature chemistry* **2019**, *11*, 1026–1033.

[29]Hager, G.; Jeckelmann, E.; Fehske, H.; Wellein, G. Parallelization strategies for density matrix renormalization group algorithms on shared-memory systems. *Journal of Computational Physics* **2004**, *194*, 795–808.

[30]Levy, R.; Solomonik, E.; Clark, B. K. Distributed-memory DMRG via sparse and dense parallel tensor contractions. *arXiv preprint arXiv:2007.05540* **2020**,

[31]Chan, G. K.-L. An algorithm for large scale density matrix renormalization group calculations. *The Journal of chemical physics* **2004**, *120*, 3172–3178.

[32]Wouters, S.; Van Neck, D. The density matrix renormalization group for ab initio quantum chemistry. *The European Physical Journal D* **2014**, *68*, 272.

[33]Chan, G. K.-L.; Keselman, A.; Nakatani, N.; Li, Z.; White, S. R. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of chemical physics* **2016**, *145*, 014102.

[34]Stoudenmire, E.; White, S. R. Real-space parallel density matrix renormalization group. *Physical review B* **2013**, *87*, 155137.

[35]Brabec, J.; Brandejs, J.; Kowalski, K.; Xantheas, S.; Legeza, Ö.; Veis, L. Massively parallel quantum chemical density matrix renormalization group method. *arXiv preprint arXiv:2001.04890* **2020**,

[36]Nemes, C.; Barcza, G.; Nagy, Z.; Legeza, Ö.; Szolgay, P. The density matrix renormalization group algorithm on kilo-processor architectures: Implementation and trade-offs. *Computer Physics Communications* **2014**, *185*, 1570–1581.

[37]Chen, F.-Z.; Cheng, C.; Luo, H.-G. Improved hybrid parallel strategy for density matrix renormalization group method. *Chinese Physics B* **2020**, *29*, 070202.

[38]Li, W.; Ren, J.; Shuai, Z. Numerical assessment for accuracy and GPU acceleration of TD-DMRG time evolution schemes. *The Journal of Chemical Physics* **2020**, *152*, 024127.

[39]Li, Z.; Chan, G. K.-L. Spin-projected matrix product states: Ver-

satile tool for strongly correlated systems. *Journal of chemical theory and computation* **2017**, *13*, 2681–2695.

[40]Keller, S.; Dolfi, M.; Troyer, M.; Reiher, M. An efficient matrix product operator representation of the quantum chemical Hamiltonian. *The Journal of chemical physics* **2015**, *143*, 244118.

[41]Hubig, C.; McCulloch, I.; Schollwöck, U. Generic construction of efficient matrix product operators. *Physical Review B* **2017**, *95*, 035129.

[42]Sharma, S.; Chan, G. K.-L. Spin-adapted density matrix renormalization group algorithms for quantum chemistry. *The Journal of chemical physics* **2012**, *136*, 124121.

[43]Wouters, S.; Poelmans, W.; Ayers, P. W.; Van Neck, D. CheMPS2: A free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry. *Computer Physics Communications* **2014**, *185*, 1501–1514.

[44]Dunning Jr, T. H. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *The Journal of chemical physics* **1989**, *90*, 1007–1023.

[45]Li, Z.; Li, J.; Dattani, N. S.; Umrigar, C.; Chan, G. K.-L. The electronic complexity of the ground-state of the FeMo cofactor of nitrogenase as relevant to quantum simulations. *The Journal of chemical physics* **2019**, *150*, 024302.

[46]Keller, S.; Reiher, M. Spin-adapted matrix product states and operators. *The Journal of chemical physics* **2016**, *144*, 134101.

[47]Ren, J.; Li, W.; Jiang, T.; Shuai, Z. A general automatic method for optimal construction of matrix product operators using bipartite graph theory. *The Journal of Chemical Physics* **2020**, *153*, 084118.

[48]Secular, P.; Gourianov, N.; Lubasch, M.; Dolgov, S.; Clark, S. R.; Jaksch, D. Parallel time-dependent variational principle algorithm for matrix product states. *Physical Review B* **2020**, *101*, 235123.

[49]Chen, F.-Z.; Cheng, C.; Luo, H.-G. Real-space parallel density matrix renormalization group with adaptive boundaries. *Chinese Physics B* **2021**,

[50]White, S. R. Density matrix renormalization group algorithms with a single center site. *Physical Review B* **2005**, *72*, 180403.

[51]Zhai, H.; Larsson, H. R. block2: Efficient MPO implementation of quantum chemistry DMRG. 2021; `https://github.com/block-hczhai/block2-preview`.

[52]Davidson, E. R. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *Journal of Computational Physics* **1975**, *17*, 87–94.