

Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm

Naoki Nakatani and Garnet Kin-Lic Chan

Citation: *J. Chem. Phys.* **138**, 134113 (2013); doi: 10.1063/1.4798639

View online: <http://dx.doi.org/10.1063/1.4798639>

View Table of Contents: <http://aip.scitation.org/toc/jcp/138/13>

Published by the [American Institute of Physics](#)



**COMPLETELY
REDESIGNED!**

**PHYSICS
TODAY**

Physics Today Buyer's Guide
Search with a purpose.

Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm

Naoki Nakatani and Garnet Kin-Lic Chan

Department of Chemistry, Princeton University, Frick Chemistry Laboratory, Princeton, New Jersey 08544, USA

(Received 30 January 2013; accepted 14 March 2013; published online 4 April 2013)

We investigate tree tensor network states for quantum chemistry. Tree tensor network states represent one of the simplest generalizations of matrix product states and the density matrix renormalization group. While matrix product states encode a one-dimensional entanglement structure, tree tensor network states encode a tree entanglement structure, allowing for a more flexible description of general molecules. We describe an optimal tree tensor network state algorithm for quantum chemistry. We introduce the concept of half-renormalization which greatly improves the efficiency of the calculations. Using our efficient formulation we demonstrate the strengths and weaknesses of tree tensor network states versus matrix product states. We carry out benchmark calculations both on tree systems (hydrogen trees and π -conjugated dendrimers) as well as non-tree molecules (hydrogen chains, nitrogen dimer, and chromium dimer). In general, tree tensor network states require much fewer renormalized states to achieve the same accuracy as matrix product states. In non-tree molecules, whether this translates into a computational savings is system dependent, due to the higher prefactor and computational scaling associated with tree algorithms. In tree like molecules, tree network states are easily superior to matrix product states. As an illustration, our largest dendrimer calculation with tree tensor network states correlates 110 electrons in 110 active orbitals. © 2013 American Institute of Physics. [<http://dx.doi.org/10.1063/1.4798639>]

I. INTRODUCTION

Currently, there is much effort devoted to finding efficient numerical techniques for strongly correlated electrons. Amongst several approaches, the Density Matrix Renormalization Group (DMRG)^{1,2} has provided many new insights in challenging systems. In recent years, efficient DMRG implementations have also appeared for quantum chemistry.^{3–9} Originally, the DMRG was formulated as an algorithm in the language of Wilson's numerical renormalization group. However, more recently, attention has expanded to the underlying class of wavefunctions optimized by the DMRG, which are the Matrix Product States (MPS).

MPS provides a compact description of entanglement in 1D systems, thus DMRG calculations for chain-like molecules are very efficient.^{10,11} However, when used in 2D and 3D systems, much larger bond-dimensions, usually denoted by M and referred to as the number of renormalized states in DMRG calculations, must be used to reach a target accuracy. The need for larger M reflects the sub-optimal representation of 2D and 3D entanglement by the MPS wavefunction structure.

The generalization of MPS to tensor network states (TNS) provides a natural way to compactly describe 2D and 3D entanglement. There are several families of TNS that differ in the way the entanglement is encoded, and a brief overview of the classes of TNS is given in Refs. 12–17. While the formal properties of general TNS are appealing, efficient computation with these states lags far behind computation with MPS. In this work, we explore efficient computation for quantum chemistry with Tree Tensor

Network States (TTNS),^{12,18–21} one of the simplest families of TNS.

TTNS encodes a tree entanglement structure, as illustrated in Figure 1. A tree of maximal degree Z has at most Z neighbours at any site. MPS is a special case of TTNS with $Z = 2$. The absence of cycles in a TTNS simplifies many algorithms. In particular, the variational minimization of TTNS wavefunctions can be carried out by generalizing the DMRG algorithm used for MPS, from a two-block formulation, to a Z -block formulation. Multi-block DMRG algorithms have been studied for some time,^{22–27} although to the best of our knowledge, their interpretation in terms of the underlying class of TTNS variational states first appeared in Ref. 19. Thus, Ref. 19 can be considered as a starting point for the current work.

In Ref. 19 the authors considered a prototype application of TTNS to quantum chemistry with a minimal basis beryllium atom calculation. Although this provided evidence that for a given M , TTNS of degree $Z > 2$ capture more entanglement than the corresponding MPS with the same M , for practical computation many questions were left unanswered. For example, although TTNS is more flexible than MPS for describing entanglement, there is an implicit trade-off of higher computational scaling with M . Furthermore, the calculation in Ref. 19 used an unrealistically small $M = 3$, whereas typical DMRG calculations use $M = 1000$ – 10000 . Thus, the relevance of TTNS for quantum chemistry calculations remains to be established.

In the current work, we address this question by describing and implementing an efficient TTNS algorithm that can

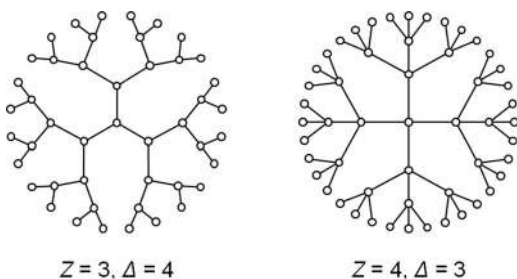


FIG. 1. Examples of trees. The left panel shows a tree with degree $Z = 3$ and depth $\Delta = 4$, and the right panel shows a tree with degree $Z = 4$ and depth $\Delta = 3$.

be used for realistic calculations. Our algorithm is formulated to have an optimal computational scaling for quantum chemistry Hamiltonians. We achieve a much lower cost than earlier formulations through a “half”-renormalization transformation which exactly maps the multi-block DMRG to a conventional, efficient, two-block DMRG. We also address the issue of orbital ordering on trees which is necessary to use TTNS in chemistry. Using our efficient implementation, we assess the performance of TTNS relative to MPS used in quantum chemistry DMRG calculations. We compare TTNS and MPS on several benchmark molecular systems, including ideal hydrogen trees and chains, and benchmark molecules such as the nitrogen and chromium dimers. Finally, to demonstrate the power of TTNS, we carry out complete active space calculations in π -conjugated dendrimer molecules, correlating up to 110 electrons in 110 orbitals.

II. OVERVIEW OF THE DMRG ALGORITHM BASED ON MPS

We first present a brief overview of MPS wavefunctions and the DMRG algorithm that optimizes their energy. This will allow us to establish basic notation which will be used to discuss TTNS in Sec. III. However, as the presentation is not entirely self-contained, for further details we refer to additional articles and reviews.^{1,4,14,28,29}

We use the following notation for MPS: n_i is the many-body basis at site i , k is number of sites, and M is the number of renormalized states. The (one-site) MPS wavefunction is obtained by expressing the coefficient of the determinant $|n_1 \dots n_k\rangle$ as a product of matrices for each occupancy $n_1 \dots n_k$,

$$|\Psi\rangle = \sum_{n_1 \dots n_k} \mathbf{L}^{n_1} \dots \mathbf{L}^{n_{i-1}} \boldsymbol{\psi}^{n_i} \mathbf{R}^{n_{i+1}} \dots \mathbf{R}^{n_k} |n_1 \dots n_k\rangle. \quad (1)$$

For a MPS with M renormalized states, the matrices are of maximum dimension $M \times M$, except for the first and last, which are of maximum dimension $1 \times M$ and $M \times 1$, respectively. Note that the MPS is invariant to a number of transformations of the matrices.^{5,29} We remove this invariance by choosing a MPS *canonical form*. In the canonical form at site i , the rotation matrices to the left of site i are constrained to satisfy orthonormality conditions $\sum_{n_i} \mathbf{L}^{n_i \dagger} \mathbf{L}^{n_i} = \mathbf{1}$, while those to the right of site i are constrained to satisfy $\sum_{n_i} \mathbf{R}^{n_i} \mathbf{R}^{n_i \dagger} = \mathbf{1}$.

The left and right rotation matrices allow us to define left and right renormalized many-body states, $|l_{i-1}\rangle$ and $|r_i\rangle$,

respectively. These renormalized representations are used to construct the computational intermediates (renormalized operators) in the DMRG algorithm. Carrying out the matrix multiplications from \mathbf{L}^{n_1} through $\mathbf{L}^{n_{i-1}}$, and from \mathbf{R}^{n_k} through $\mathbf{R}^{n_{i+1}}$, we obtain

$$|l_{i-1}\rangle = \sum_{n_1 \dots n_{i-1}} \mathbf{L}^{n_1} \dots \mathbf{L}^{n_{i-1}} |n_1 \dots n_{i-1}\rangle, \quad (2)$$

$$|r_i\rangle = \sum_{n_{i+1} \dots n_k} \mathbf{R}^{n_{i+1}} \dots \mathbf{R}^{n_k} |n_{i+1} \dots n_k\rangle. \quad (3)$$

The orthonormality conditions on the rotation matrices \mathbf{L} and \mathbf{R} imply that the renormalized bases are orthonormal

$$\langle l_{i-1} | l'_{i-1} \rangle = \delta_{ll'}, \quad (4)$$

$$\langle r_i | r'_i \rangle = \delta_{rr'}. \quad (5)$$

The DMRG (canonical) form of the MPS wavefunction is obtained by rewriting the MPS wavefunction (1) in terms of the renormalized many-body bases $|l_{i-1}\rangle$ and $|r_i\rangle$,

$$|\Psi\rangle = \sum_{l_{i-1} n_i r_i} \psi_{l_{i-1} n_i r_i}^{n_i} |l_{i-1} n_i r_i\rangle. \quad (6)$$

In this interpretation, $\boldsymbol{\psi}^{n_i}$ is viewed as a wavefunction coefficient vector in a Hilbert space spanned by the renormalized product states, $|l_{i-1} n_i r_i\rangle = |l_{i-1}\rangle |n_i\rangle |r_i\rangle$.

Computations using MPS involve tensor-tensor contractions. To express such operations, it is helpful to use a graphical representation, shown in Figure 2. Each vertex is a tensor and the number of edges connected to the vertex determines the tensor rank. In the case of a MPS, each \mathbf{L}^{n_i} or \mathbf{R}^{n_i} is a rank-3 tensor, represented by a vertex with three edges. Here, we always choose the vertical index to represent n_i . The MPS wavefunction is obtained by contracting the horizontal edges of all the tensors (Fig. 2(b)), leading to Eq. (1). The computation of the overlap of two MPS is shown in Fig. 2(c).

We optimize the MPS energy, by minimizing the Lagrangian $\langle \Psi | \hat{H} | \Psi \rangle - \lambda (\langle \Psi | \Psi \rangle - 1)$ with respect to the tensors in the MPS. In the (one-site) DMRG sweep algorithm, this minimization is carried out with respect to a single tensor at a time. In step i of the DMRG sweep, the MPS is expressed in the DMRG form (6), and the coefficient vector $\boldsymbol{\psi}$ is optimized, holding the rotation matrices to the left and right of the site which define the bases $|l_{i-1}\rangle$, $|r_i\rangle$, fixed. Because the energy is a quadratic form in $\boldsymbol{\psi}$, minimization leads to a standard eigenvalue problem,

$$\sum_{l'_{i-1} n'_i r'_i} \langle l_{i-1} n_i r_i | \hat{H} | l'_{i-1} n'_i r'_i \rangle \psi_{l'_{i-1} r'_i}^{n'_i} - \lambda \psi_{l_{i-1} r_i}^{n_i} = 0. \quad (7)$$

In subsequent steps of the sweep, the MPS is transformed to the DMRG form at successive sites, and the coefficient vectors at these sites are optimized.

The most expensive operation in the sweep is computing $\langle \Psi | \hat{H} | \Psi \rangle$ and performing $\hat{H} \boldsymbol{\psi}$ to solve the eigenvalue

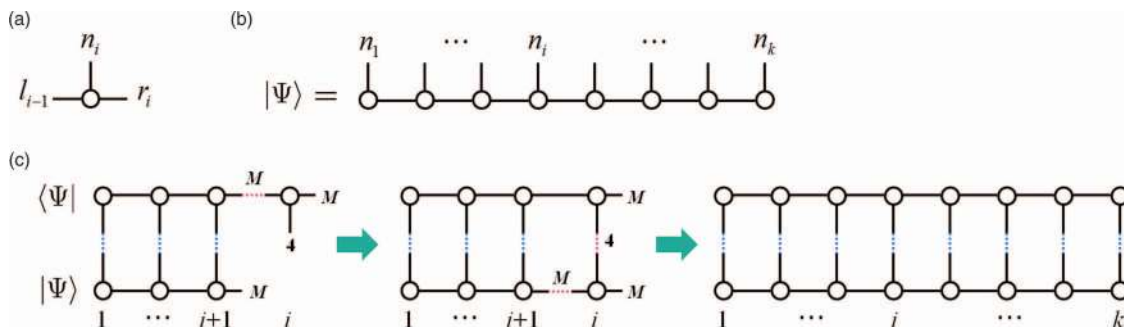


FIG. 2. Graphical representations of MPS. (a) one-site coefficient vector ψ^{n_i} is a rank-3 tensor, (b) the MPS wavefunction is given by contracting all horizontal bonds, (c) the overlap of two MPS wavefunctions can be efficiently computed recursively.

problem (7). This is because the second-quantized Hamiltonian contains a large number of terms,

$$\hat{H} = \sum_{ij} h_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \sum_{ijkl} v_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k \hat{a}_l, \quad (8)$$

and for each term we need its matrix representation in the basis $\{|l_{i-1} n_i r_i\rangle\}$. There are two generic strategies to handle the large number of terms. The first, used in existing quantum chemistry DMRG implementations, is through *Complementary Operators*.^{3,4,30} The second uses the more recent concept of *Matrix Product Operators*.³¹⁻³⁵

Complementary operators are a way to maximize the reuse of intermediates. For example, there are $\mathcal{O}(k^4)$ terms in the summation (8), and each expectation value individually is of $\mathcal{O}(M^3)$ cost, leading to a naive scaling of $\mathcal{O}(M^3 k^4)$ for the energy. However, much information can be reused between terms. For example, the two terms $\langle a_1^\dagger a_2^\dagger a_3 a_4 \rangle$ and $\langle a_1^\dagger a_2^\dagger a_6 a_7 \rangle$ involve the same partial expectation value over $a_1^\dagger a_2^\dagger$. Complementary operators reuse and combine such partial traces. At site i in the DMRG sweep, we partition the Hilbert space into two subspaces: A, containing the left block of sites (sites $1 \dots i-1$) and site i , spanned by renormalized states $|l_{i-1} n_i\rangle$, and B, containing the right block of sites $i+1 \dots k$, and spanned by renormalized states $|r_i\rangle$ (see Fig. 3). \hat{H} is correspondingly partitioned as

$$\hat{H} = \hat{H}_A + \hat{H}_B + \hat{H}_{AB}, \quad (9)$$

\hat{H}_A and \hat{H}_B are acting locally on A and B, respectively, and have non-trivial expectation values with only $|l_{i-1} n_i\rangle$ and $|r_i\rangle$ separately. \hat{H}_{AB} describes the interactions between A and B, and is given by a sum of products acting separately on the two

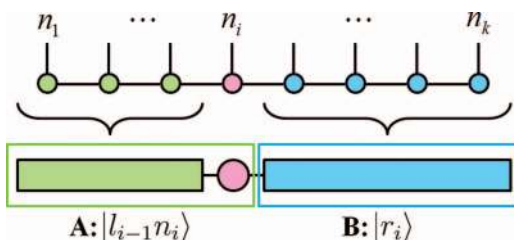


FIG. 3. Graphical representations of DMRG wavefunction. Canonical form of MPS wavefunction (top) can be re-written as a block diagram in DMRG language (bottom).

spaces

$$\begin{aligned} \hat{H}_{AB} = & \sum_{i \in A, j \in B} (\hat{a}_i^{A\dagger} \hat{S}_i^B + \hat{a}_j^{B\dagger} \hat{S}_j^A + \hat{a}_i^{A\dagger} \hat{a}_i^A \hat{Q}_{ii}^B) \\ & + \sum_{i > j \in A} (\hat{a}_i^A \hat{a}_j^A \hat{P}_{ij}^{B\dagger} + \hat{a}_i^{A\dagger} \hat{a}_j^A \hat{Q}_{ij}^B) + \text{adjoint}, \quad (10) \end{aligned}$$

where \hat{P}_{ij} , \hat{Q}_{ij} , and \hat{S}_i are the *Complementary Operators*,

$$\hat{P}_{ij}^B = \sum_{kl \in B} v_{ijkl} \hat{a}_k \hat{a}_l, \quad (11)$$

$$\hat{Q}_{ij}^B = \sum_{kl \in B} (v_{ikjl} - v_{ijkl}) \hat{a}_k^\dagger \hat{a}_l, \quad (12)$$

$$\hat{S}_i^B = \sum_{j \in B} t_{ij} \hat{a}_j + \sum_{jkl \in B} v_{ijkl} \hat{a}_j^\dagger \hat{a}_k \hat{a}_l. \quad (13)$$

Using complementary operators, the total complexity of evaluating $\langle \Psi | H | \Psi \rangle$ and performing $\hat{H} \Psi$ is reduced to $\mathcal{O}(M^3 k^3 + M^2 k^4)$ per sweep, including the cost of constructing the components of \hat{H} in the partitioned form (renormalization).⁴ This is the standard procedure to evaluate the energy in the DMRG algorithm.

An alternative approach to handle complicated Hamiltonians is through *Matrix Product Operators* (MPO). MPO's provide a convenient way to reason about operators in MPS algorithms and have been employed extensively in time-dependent MPS simulations.³¹ Here we provide a brief analysis of this approach for quantum chemistry. The basic idea in a MPO is extend the matrix product like representation to operators. Writing the Hamiltonian in the occupation number basis as a general rank- $2k$ tensor, $[\hat{H}]^{n_1 \dots n_k n'_1 \dots n'_k}$, it is decomposed into a set of tensors analogously to a MPS,

$$[\hat{H}]^{n_1 \dots n_k n'_1 \dots n'_k} = \mathbf{W}^{[1]n_1 n'_1} \dots \mathbf{W}^{[i]n_i n'_i} \dots \mathbf{W}^{[k]n_k n'_k}. \quad (14)$$

This is illustrated graphically in Figure 4 where $\mathbf{W}^{[i]}$ is a rank-4 tensor and the contraction of horizontal edges describes the quantum “entanglement” of the Hamiltonian operator. To decompose \hat{H} exactly, the dimension of the horizontal edge of $\mathbf{W}^{[i]}$ needs to be exactly the same as the number of complementary operators, that is $\mathcal{O}(k^2)$. Consequently, the cost of computing the expectation value $\langle \Psi | H | \Psi \rangle$ with a MPO representation becomes $\mathcal{O}(M^3 k^3 + M^2 k^5)$. Note that this cost is *larger* than in the complementary operator approach. The

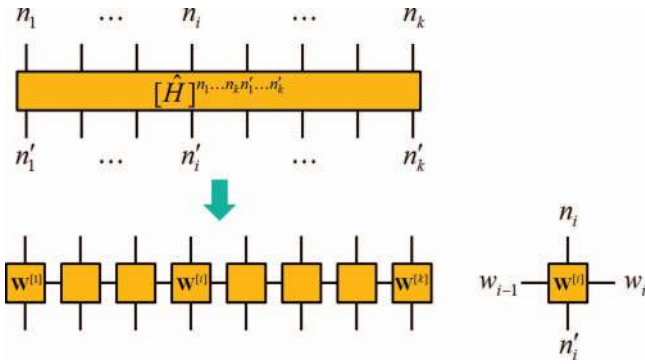


FIG. 4. Graphical representations of MPO. The Hamiltonian operator is a rank- $2k$ tensor (top panel). This can be divided into a contracted product of site-independent tensors similar to a MPS, leading to a Matrix Product Operator representation (MPO, bottom panel), in which each site tensor is a rank-4 tensor.

difference arises because we have not considered the sparsity of the individual $\mathbf{W}^{[l]}$ tensors that arise in the Hamiltonian decomposition. However, incorporating element-wise sparsity into a MPO algorithm eliminates much of the conceptual and algorithmic simplicity of the MPO approach. Consequently, in our view, the complementary operator algorithm is a more practical and efficient route for MPS computations with quantum chemistry Hamiltonians. The relative benefit of using complementary operators versus tensor product operators (TPO's) is even greater for TTNS than for MPS, thus we focus on the complementary operator approach when considering trees.

III. TREE TENSOR NETWORK STATES (TTNS)

Tensor Network States (TNS) are mathematical generalizations of the MPS that can code more general entanglement networks. The form of a TNS wavefunction is directly analogous to a MPS wavefunction,

$$|\Psi\rangle = \sum_{n_1 \dots n_k} \text{tr}[\mathbf{A}^{n_1} \cdot \dots \mathbf{A}^{n_i} \cdot \dots \mathbf{A}^{n_k}] |n_1 \dots n_k\rangle, \quad (15)$$

the only difference being that \mathbf{A}^{n_i} is now a tensor, rather than a rotation matrix as in the MPS, and the multiplication operator together with tr denotes a general contraction over tensor indices. The flexibility of TNS wavefunctions arises from the fact that whereas the matrices in a MPS can only be contracted along a 1D-lattice, there are many different ways to connect general tensors together to form a network of entanglement.

Tree tensor network states (TTNS) are a special class of TNS where the tensors are connected as a tree as shown in Figure 5. A tree is a graph that has no loops, which leads to many simplifying mathematical properties that parallel those of a MPS (indeed, a MPS is simply a tree with $Z = 2$ legs). For example, at a given site i in the tree, we can define renormalized bases for each of the Z legs connected to the site. This allows us to rewrite (15) as

$$|\Psi\rangle = \sum_{b_1^1 \dots b_i^Z n_i} \psi_{b_1^1 \dots b_i^Z}^{n_i} |b_1^1 \dots b_i^Z n_i\rangle, \quad (16)$$

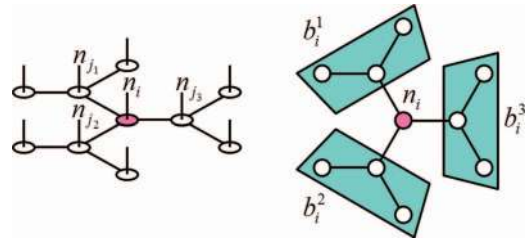


FIG. 5. Graphical representation of TTNS. Left panel shows overall structure of TTNS as described in Eq. (15), and right panel shows the one-site wavefunction spanned by the renormalized basis, as described in Eq. (16). Note that physical indices (vertical bonds) are omitted in the right panel.

where $|b_i^\alpha\rangle$ is the renormalized basis in the α th branch of site i . This basis is defined by recursively contracting tensors in the branch from the leaves up to site i . Thus $|b_i^\alpha\rangle$ is obtained as

$$|b_i^\alpha\rangle = \sum_{b_j^1 \dots b_j^{Z-1} n_j} A_{b_j^1 \dots b_j^{Z-1} b_i^\alpha}^{n_j} |b_j^1 \dots b_j^{Z-1} n_j\rangle, \quad (17)$$

where sites j are adjacent to i in the branch. Note that for a Z degree TTNS with M renormalized states, the tensor $A_{b_j^1 \dots b_j^{Z-1} b_i^\alpha}^{n_j}$ has $\mathcal{O}(M^Z)$ elements.

Analogous to the rotation matrices in MPS, the tensors \mathbf{A}^{n_i} in a given branch around site i can be chosen to satisfy orthonormality constraints, rendering the TTNS in canonical form,

$$\sum_{b_j^1 \dots b_j^{Z-1} n_j} A_{b_j^1 \dots b_j^{Z-1} b_i^\alpha}^{n_j*} A_{b_j^1 \dots b_j^{Z-1} b_i^\alpha}^{n_j} = \delta_{b_i^\alpha b_i^\alpha}. \quad (18)$$

As a result, the renormalized basis states $|b_i^\alpha\rangle$ are orthonormal.

The above mathematical properties make a DMRG energy optimization algorithm for TTNS very similar to that for MPS. Similarly to the DMRG algorithm for MPS, we optimize one site at a time. The TTNS is expressed in canonical form around site i , then the coefficient tensor ψ^{n_i} is optimized, and the sites of the tree are traversed during the sweep. The computational challenge is once again how to efficiently compute the representation of the Hamiltonian in the renormalized bases, namely $\langle b_i^1 \dots b_i^Z n_i | \hat{H} | b_i^1 \dots b_i^Z n_i \rangle$, and its action on the coefficient vector ψ^{n_i} , $\hat{H} \psi$. As we discussed for the case of MPS, for quantum chemistry Hamiltonians the complementary operator approach is most natural.

We rewrite the Hamiltonian in complementary operator form by partitioning into $Z + 1$ blocks, A, B, C, D, and so on, corresponding to the Z branches around site i , and site i itself. The Hamiltonian is re-expressed in terms of operators acting on each of the blocks separately,

$$\begin{aligned} \hat{H} = & \hat{H}_A + \hat{H}_B + \hat{H}_C + \hat{H}_D + \dots \\ & + \hat{H}_{AB} + \hat{H}_{AC} + \hat{H}_{BC} + \dots \\ & + \hat{H}_{ABC} + \hat{H}_{ACD} + \hat{H}_{BCD} + \dots \\ & + \hat{H}_{ABCD} + \dots, \end{aligned} \quad (19)$$

where,

$$\hat{H}_{AB} = \sum_{i \in A, j \in B} (\hat{a}_i^{A\dagger} \hat{S}_i^B + \hat{a}_j^{B\dagger} \hat{S}_j^A + \hat{a}_i^{A\dagger} \hat{a}_i^A \hat{Q}_{ii}^B) + \sum_{i > j \in A} (\hat{a}_i^A \hat{a}_j^A \hat{P}_{ij}^{B\dagger} + \hat{a}_i^{A\dagger} \hat{a}_j^A \hat{Q}_{ij}^B) + \text{adjoint}, \quad (20)$$

$$\hat{H}_{ABC} = \sum_{i \in A, j \in B} (\hat{a}_i^A \hat{a}_j^B \hat{P}_{ij}^{C\dagger} + \hat{a}_i^{A\dagger} \hat{a}_j^B \hat{Q}_{ij}^C) + \sum_{i \in A, j \in C} (\hat{a}_i^A \hat{a}_j^C \hat{P}_{ij}^{B\dagger} + \hat{a}_i^{A\dagger} \hat{a}_j^C \hat{Q}_{ij}^B) + \sum_{i \in B, j \in C} (\hat{a}_i^B \hat{a}_j^C \hat{P}_{ij}^{A\dagger} + \hat{a}_i^{B\dagger} \hat{a}_j^C \hat{Q}_{ij}^A) + \text{adjoint}, \quad (21)$$

$$\hat{H}_{ABCD} = \frac{1}{2} \sum_{\substack{i \in A, j \in B \\ k \in C, l \in D}} v_{ijkl} \hat{a}_i^{A\dagger} \hat{a}_j^{B\dagger} \hat{a}_k^C \hat{a}_l^D + \text{permutation}. \quad (22)$$

As in the case of MPS, the full matrix representation of \hat{H} is never built explicitly as the storage requirements would be immense. Instead only $\hat{H}\psi$ is computed in the Davidson algorithm. Note that when computing $\hat{H}\psi$ the order of multiplication of the various terms in the complementary operator decomposition of \hat{H} is important. For example, the term $\hat{a}_i^A \hat{a}_j^B \hat{P}_{ij}^{C\dagger} \psi$ in which A and C are large blocks with $\mathcal{O}(k)$ sites and M renormalized states (indices b_1 and b_2 below) and B is the site being optimized, with 4 states (index n_i below) should be computed as

$$\sum_{i \in A, j \in B} [\hat{a}_i^A] \otimes [\hat{a}_j^B] \otimes [\hat{P}_{ij}^{C\dagger}] \psi^{n_i} = \sum_{i \in A, b_1} [\hat{a}_i^A]^{b_1} \left(\sum_{b_2, n_i} \psi_{b_1 b_2 \dots b_Z}^{n_i} \left(\sum_{j \in B} [\hat{a}_j^B]^{n_i} [\hat{P}_{ij}^{C\dagger}]^{b_2 b_2} \right) \right). \quad (23)$$

In the above form, $\sum_{j \in B} [\hat{a}_j^B] \otimes [\hat{P}_{ij}^{C\dagger}]$ is of $\mathcal{O}(16M^2k)$ cost, $[\hat{R}_i^{BC\dagger}] \psi^{n_i}$ is of $\mathcal{O}(64M^{Z+1}k)$ cost, and finally $\sum_{i \in A} [\hat{a}_i^A] [\hat{R}_i^{BC\dagger} \psi]$ is of $\mathcal{O}(4M^{Z+1}k)$ cost.

We have carefully optimized the order of tensor contractions for each of the terms in (20)–(22), obtaining the complexities shown in Table I. For a general Z degree tree, the total cost of a single $\hat{H}\psi$ (sigma vector) computation is $\mathcal{O}(M^{Z+1}k^3 + M^Zk^5)$ per sweep. To this must be added the construction of the matrix representations of the components of \hat{H} in the different blocks (renormalization steps), which costs $\mathcal{O}(M^{Z+1}k^3 + M^Zk^5)$ per sweep (The renormalization is described explicitly in the Appendix). In the case of $Z = 2$ (MPS) we have already noted that the cost per sweep is lower than in the general case ($\mathcal{O}(M^3k^3 + M^2k^4)$). This is due to the absence of several terms in Table I. The same is true for the $Z = 3$ tree, where the total cost of a sweep is $\mathcal{O}(M^4k^3 + M^2k^4)$ ($\hat{H}\psi$) and $\mathcal{O}(M^4k^3 + M^2k^5)$ (renormalization). Because of the special efficiency of the $Z = 3$ tree, our later computations focus on this kind of tree.

TABLE I. Complexity of optimal tensor contractions for $\hat{H}\psi$ per site. Multiplying by $\mathcal{O}(k)$ gives complexity per sweep.

Terms in \hat{H}	Complexity
$\hat{a}_i^A \hat{S}_i^{B\dagger}$	$\mathcal{O}(M^{Z+1}k)$
$\hat{a}_i^A \hat{a}_j^A \hat{P}_{ij}^{B\dagger}$	$\mathcal{O}(M^{Z+1}k^2)$
$\hat{a}_i^{A\dagger} \hat{a}_j^A \hat{Q}_{ij}^B$	$\mathcal{O}(M^{Z+1}k^2)$
$\hat{a}_i^{A\dagger} \hat{a}_j^{B\dagger} \hat{P}_{ij}^C$	$\mathcal{O}(M^{Z+1}k^2 + M^{Z+1}k)$
$\hat{a}_i^{A\dagger} \hat{a}_j^B \hat{Q}_{ij}^C$	$\mathcal{O}(M^{Z+1}k^2 + M^{Z+1}k)$
$\hat{a}_i^{A\dagger} \hat{a}_j^{B\dagger} \hat{a}_k^C \hat{a}_l^D$	$\mathcal{O}(M^{Z+1}k^2 + M^{Z+1}k + M^Zk^4)$

A. Half-renormalization

We have found additionally that it is possible to significantly reduce the computational prefactor of a TTNS DMRG calculation through an additional step we call *half-renormalization* (Fig. 6). Half-renormalization involves first constructing an *exact* mapping of the TTNS onto an equivalent $Z = 2$ MPS, then carrying out the $\hat{H}\psi$ operations in this simpler representation. Although the mapping is itself

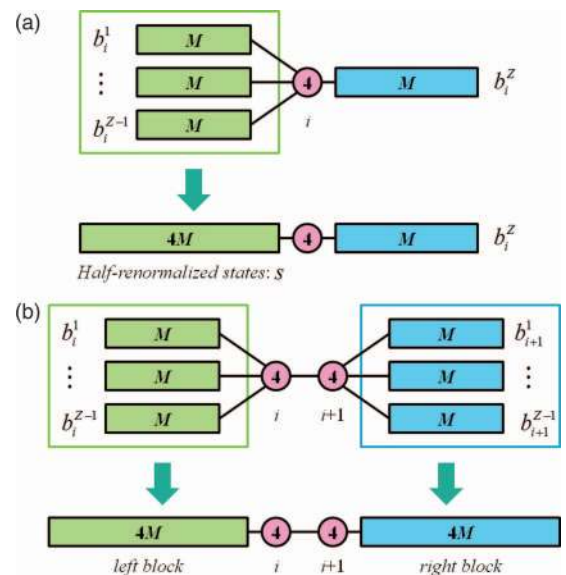


FIG. 6. Half-renormalization (HR) algorithm on TTNS. (a) One-site algorithm: $Z - 1$ system blocks $\{|b_i^\alpha\rangle\}$ are mapped into one system block. (b) Two-site algorithm: additionally, $Z - 1$ environment blocks $\{|b_{i+1}^\beta\rangle\}$ are mapped into one environment block. The half renormalized block contains only $4M$ states for any $Z > 2$.

expensive (and retains the full computational scaling of the TTNS sweep described above), it needs only be carried out once per site, while the $\hat{H}\psi$ operations typically need to be carried out many times per site during a Davidson diagonalization. To map a TTNS onto a MPS, we consider the coefficient tensor at site i , $\psi_{b^1\dots b^Z}^{n_i}$. Through a singular value decomposition (SVD), this tensor can be exactly decomposed into a rank-3 tensor and a residual tensor,

$$\begin{aligned} A_{b^1\dots b^Z}^{n_i(\text{TTNS})} &= \sum_{s s'} U_{b^1\dots b^{Z-1}s} S_{s s'} V_{s' b^Z}^{n_i \dagger} \\ &= \sum_s U_{b^1\dots b^{Z-1}s} \psi_{s b^Z}^{n_i(\text{MPS})}, \end{aligned} \quad (24)$$

where U is the residual tensor and SV^\dagger is a MPS-like pseudo-coefficient tensor $\psi_{s b^Z}^{n_i(\text{MPS})}$. Note that S is a diagonal matrix with only $4 \times M$ non-zero singular values, thus $\psi_{s b^Z}^{n_i(\text{MPS})}$ has $O(M^2)$ values, similar to a MPS coefficient tensor with M renormalized states. $U_{b^1\dots b^{Z-1}s}$ defines the half-renormalization mapping from the states on $Z-1$ branches, $\alpha = 1 \dots Z-1$ of a general tree to a *single* set of $4M$ renormalized states $|b_i^\alpha\rangle$ on a single effective branch. These new ‘‘half-renormalized’’ states, together with the states on the remaining branch $\alpha = Z$, define the left and right states of an effective MPS, together with the pseudo-coefficient tensor $\psi_{s b^Z}^{n_i(\text{MPS})}$. In the case of $Z = 3$, the mapping costs $O(M^4 k^3 + M^2 k^5)$ per sweep which leads to the same scaling as the one-site algorithm without half-renormalization. However, the $\hat{H}\psi$ operations now carry the MPS cost, namely $O(M^3 k^3)$ per sweep, leading to a significant savings in computation time. As the mapping is exact, there is no approximation involved, although one drawback is that we typically see slower convergence, as at a given site we optimize only the pseudo-coefficient tensor rather than the full coefficient tensor of the TTNS.

The full DMRG sweep on a tree is carried out using depth-first search with backtracking. Applying this to a $Z = 2$ tree, as in a MPS, recovers the usual DMRG sweep algorithm. Figure 7(a) shows the details of the one-site sweep algorithm on TTNS.

- (1) Define any site as the root (depth = 0). Construct the TTNS in canonical form at the site, construct the renormalized states and operators by contracting from the leaves to the root site.

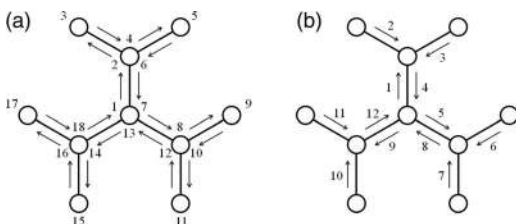


FIG. 7. Sweep algorithm on a tree ($Z = 3$), by depth-first search with backtracking, where the labels indicate the order of the search. (a) One-site algorithm: starting from the center site, 1 sweep contains 18 one-site optimization steps, and (b) Two-site algorithm: starting from the center site and one adjacent site, 1 sweep contains 12 two-site optimization steps.

- (2) Optimize the wavefunction at site i with Davidson diagonalization. Renormalize to construct the canonical form at a neighbouring site.
- (3) Continue (2) by carrying out a depth-first search with backtracking to optimize the entire tree.
- (4) Continue (2) and (3) until the energy is converged to a target accuracy.

B. Two-site TTNS algorithm

Although our discussion of MPS focused for simplicity on the one-site MPS and the corresponding one-site DMRG algorithm, it is well known from practical experience that one-site DMRG calculations tend to get stuck in local minima and suffer from poor convergence characteristics. Thus, the two-site MPS and DMRG algorithm are more commonly used.^{4,28,36} In the case of MPS, the two-site MPS wavefunction is obtained by modifying the coefficient tensor to span two sites,

$$|\Psi\rangle = \sum_{l_{i-1} n_i n_{i+1} r_{i+1}} \psi_{l_{i-1} n_i n_{i+1} r_{i+1}}^{n_i n_{i+1}} |l_{i-1} n_i n_{i+1} r_{i+1}\rangle. \quad (25)$$

A Z -branch two-site TTNS is modified in a similar way

$$\sum_{\substack{n_i n_{i+1} \\ \mathbf{b}_i \mathbf{b}_{i+1}}} \psi_{b_i^1 \dots b_i^{Z-1} b_{i+1}^1 \dots b_{i+1}^{Z-1}}^{n_i n_{i+1}} |b_i^1 \dots b_i^{Z-1} n_i n_{i+1} b_{i+1}^1 \dots b_{i+1}^{Z-1}\rangle. \quad (26)$$

An important difference however, between a general two-site tree and two-site MPS is that whereas around sites $i, i+1$ in a MPS we can define left and right renormalized basis, just as for an one-site MPS, in the case of a TTNS, there are $2Z-2$ branches around sites $i, i+1$ (Fig. 8), rather than the Z branches around a single-site. Consequently, the naive cost of the $\hat{H}\psi$ operations in a two-site TTNS DMRG sweep using complementary operators is much higher than that for an one-site TTNS sweep, with a prohibitive cost of $O(M^{2Z-1} k^3 + M^{2Z-2} k^5)$. However, by employing two sets of half-renormalization steps, we can map the two-site TTNS onto a two-site MPS, reducing the cost of the $\hat{H}\psi$ operations to only $O(M^3 k^3)$ per sweep with an additional half-renormalization cost of $O(M^{Z+1} k^3 + M^Z k^5)$ per sweep for general Z degree trees, or $O(M^4 k^3 + M^2 k^5)$ per sweep in the case of $Z = 3$. This is the same cost as an one-site TTNS sweep with half-renormalization, and thus the two-site TTNS

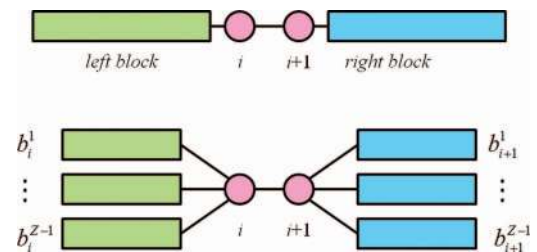


FIG. 8. Block structure of the two-site algorithm. Top panel shows the two-site MPS which contains one system block, system site, environment site, and one environment block. Bottom panel shows the two-site TTNS which contains $Z-1$ system blocks, system site, environment site, and $Z-1$ environment blocks.

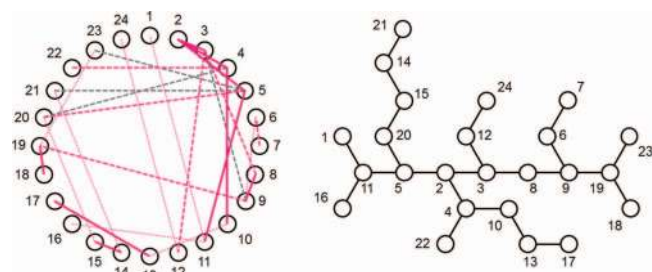


FIG. 9. Approximate degree-fixed minimum spanning tree (MST, $Z = 3$) for 24 orbitals from a RHF/cc-pVDZ calculation of the water molecule. Left panel shows a representation of the exchange integral K_{ij} , where solid lines denote $K_{ij} \geq 0.10$, dashed lines denote $K_{ij} \geq 0.07$, and dotted lines denote $K_{ij} < 0.07$. Colored lines are connected lines in the right panel and gray lines are ignored interactions. For $K_{ij} < 0.07$, only connected lines are shown. Labels in the right panel indicate the MOs (as indexed by energy).

sweep becomes practical. We compare the cost and convergence characteristics of the one-site and two-site TTNS sweeps with half-renormalization in our later calculations.

C. Tree shape and site ordering

When carrying out a quantum chemistry DMRG calculation using a MPS, it is necessary to choose a mapping of the sites to the 1D lattice. Generally, this should be done to minimize entanglement between distant sites on the lattice,^{37–39} but computing the entanglement and carrying out an exact minimization are costly procedures. In practice, an approximate proxy for the entanglement between orbitals is constructed, and it is approximately minimized.⁴ One example of such a proxy which has been used in prior DMRG studies is a weighted exchange integral $K_{ij}D_{ij}^n$, where K_{ij} is the exchange integral between orbitals i and j , D_{ij} is the separation on the lattice, and n is an adjustable parameter.

In the case of ordering for trees, we need to consider not only the mapping of the orbitals onto a given tree, but also the shape of the tree as well, even if we restrict ourselves to trees of a fixed maximum degree. Since the computational cost of a

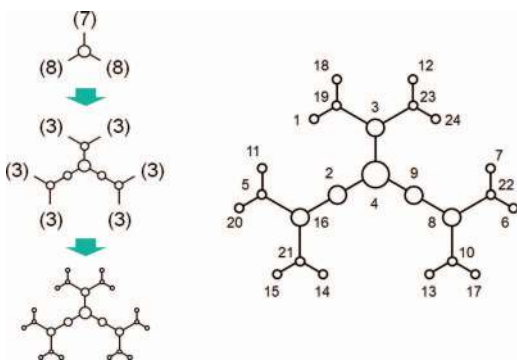


FIG. 10. Minimum-entangled tree (MET, $Z = 3$) for 24 orbitals from a RHF/cc-pVDZ calculation of the water molecule. Left panel shows the stepwise construction of MET. First, 24 sites are divided into 1 center site and 3 branches containing 7, 8, and 8 sites. Second, 7 sites are divided into 1 site and 2 branches containing 3 sites for each, and 8 sites are divided into 2 sites and 2 branches containing 3 sites for each. Finally, we used a genetic algorithm to map the orbitals to the tree sites as shown in right panel. Labels in right panel indicate MO indices (as indexed by energy).

TABLE II. The ground state energies of water molecule calculated with two different trees.

M	MST		MET	
	$E/a.u.$	CPU time/s sweep	$E/a.u.$	CPU time/s sweep
100	-76.242805	307.9	-76.242242	220.6
200	-76.243652	1209.9	-76.243491	826.5

sweep also depends on the tree shape (changing the prefactor associated with the cost of the sweep) the tree which gives the lowest energy for a given M is not necessarily the most efficient tree to use in practice.

To illustrate these issues we consider the water molecule in a cc-pVDZ basis set, which has 24 orbitals. We have considered two shapes of trees: a degree-fixed Minimum-Spanning Tree (MST) shown in Figure 9, and a Minimum-Entangled Tree (MET). The MET is defined as the shape of tree, for a given degree, where the number of renormalized states required to achieve an *exact* calculation is minimized. Its construction is shown in Figure 10. Because of its balanced nature, the MET also minimizes the prefactor of the cost of the TTNS calculation for a given M .

For each of these trees, we mapped the orbitals onto the sites by a genetic algorithm⁴⁰ that minimized the cost function $\sum_{ij} K_{ij}D_{ij}^2$, where D_{ij} is the counting distance between the sites i and j in the tree.

The ground state energies computed with the two different trees and two different M (100 and 200) are summarized in Table II. From these calculations, we see that the MST gives a slightly better energy than that the MET, but the MST sweep is 50% slower than the MET sweep for the same M , due to the unbalanced nature of the MST. For larger M , because the MET minimizes the number of renormalized states for the

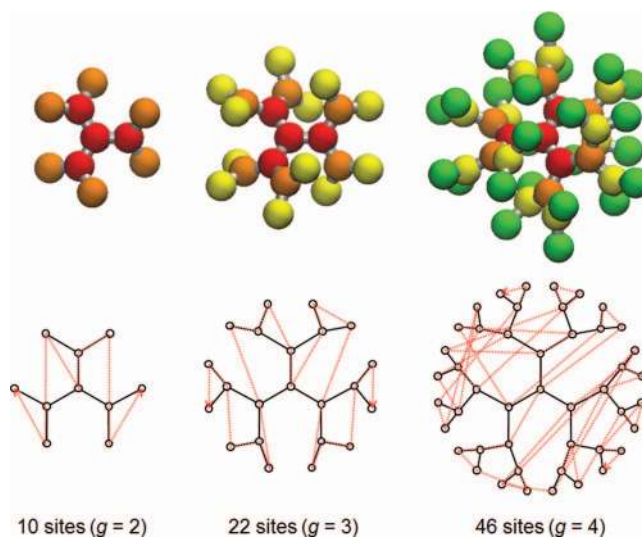


FIG. 11. Hydrogen atoms on Cayley-trees ($Z = 3$, $g = 2$ (10 sites), 3 (22 sites), and 4 (46 sites)). Top panel shows the actual structures of the hydrogen trees, in which red atoms are the core region, orange atoms are for $g = 2$, yellow atoms are for $g = 3$, and green atoms are for $g = 4$. The distance between adjacent hydrogens is 2.0 bohrs and torsional angle between different generations is 30° . Bottom panel shows corresponding Cayley-tree diagrams. The red-dotted line denotes the site ordering used in the MPS.

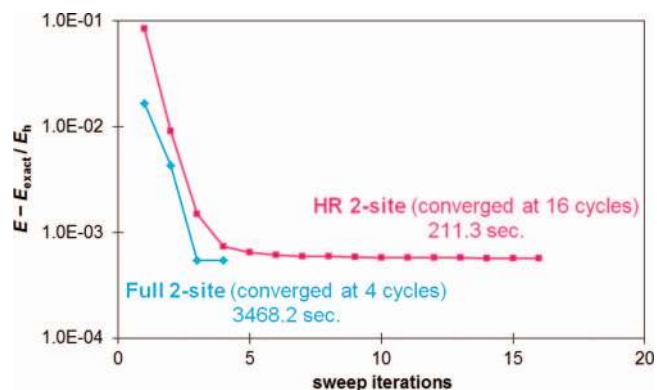


FIG. 12. Energy convergence and total CPU time computed with normal two-site TTNS (denoted full) and half-renormalized two-site TTNS (denoted HR-TTNS).

exact calculation, we expect it to eventually give a lower energy for a given M than the corresponding MST. Although these results are system dependent, they indicate the importance of tree shape in determining the cost of practical TTNS calculations. We adopt the MET in our calculations unless an-

other tree shape is trivially indicated by the molecular structure, e.g., in a tree shaped molecule such as a dendrimer.

IV. ILLUSTRATIVE CALCULATIONS

We now present several benchmark applications to molecular systems to understand the performance of TTNS and to compare with that of MPS. For the MPS calculations, we used our TTNS code with $Z = 2$. This was to allow a fair comparison of timings using the same implementation.

To start, we consider an idealized system: hydrogen atoms on Cayley-trees in an (orthogonalized) minimal STO-3G basis. This is a model system where TTNS is expected to work very well. To avoid nearly overlapping hydrogens which would arise in a planar geometry, the tree structures are taken to be slightly twisted as shown in Figure 11. All geometries are provided in the supplementary material.⁴⁵

We first illustrate the importance of the half-renormalization algorithm by comparing the normal two-site TTNS algorithm and the HR-TTNS two-site algorithm on a small 10 site hydrogen tree. The energy convergence along the sweeps and total process time (CPU time) is shown in Figure 12. Though the convergence per sweep is slower

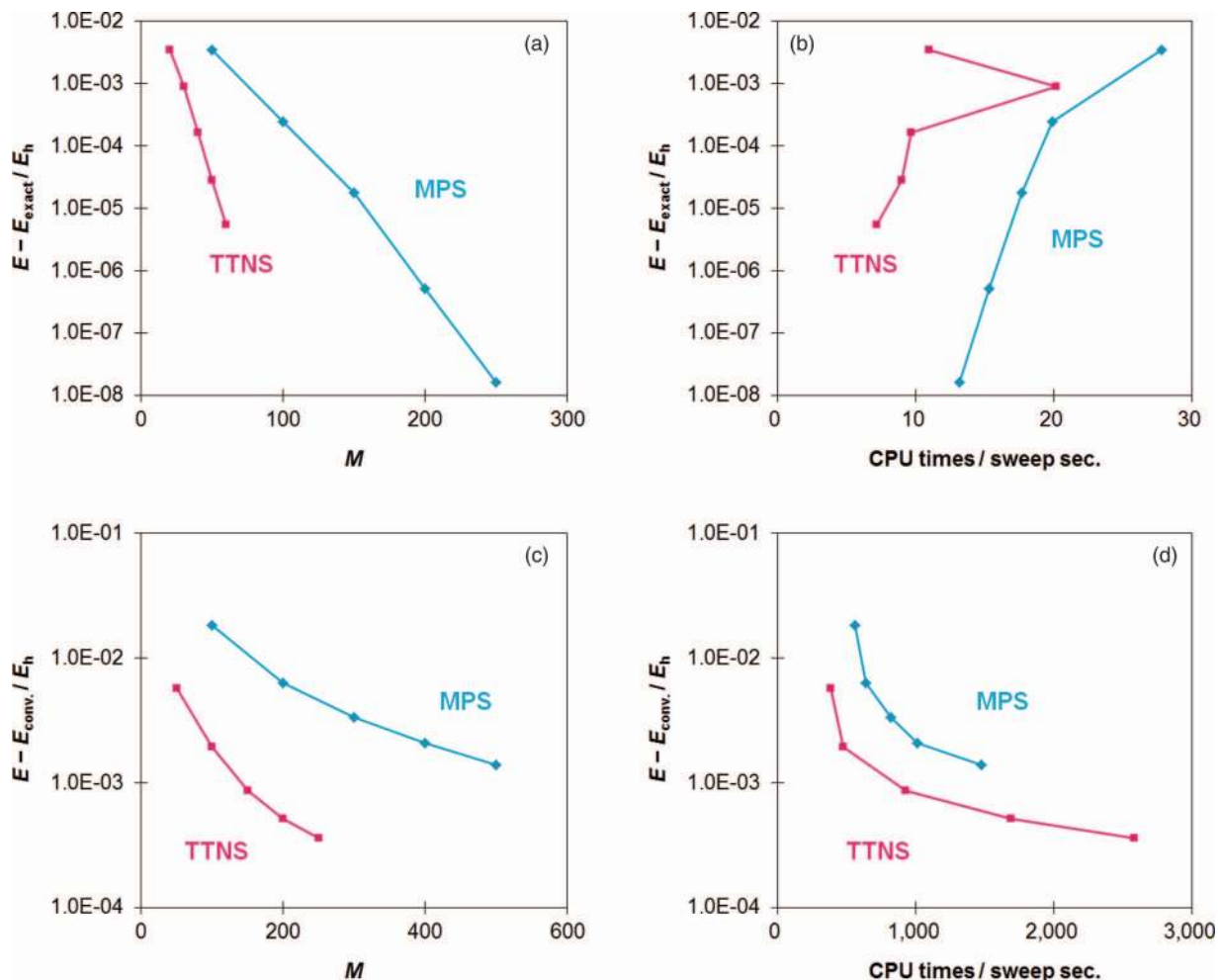


FIG. 13. Energy convergence of hydrogen trees plotted with respect to the number of renormalized states M and CPU time (s) per one sweep; (a) and (b) show energy versus M and CPU time per sweep, respectively, for 10 sites, and (c) and (d) show energy versus M and CPU time per sweep, respectively, for 22 sites. All calculations are for the triplet ground state energy in an orthogonalized STO-3G basis.

in the HR-TTNS algorithm, the total CPU time is much smaller than in the normal two-site TTNS algorithm due to the much less expensive $\hat{H}\psi$ operation. Thus, the remaining calculations have been done using the HR-TTNS algorithm, which we henceforth shorten to TTNS.

Next, we compare the performance of the MPS and TTNS on three hydrogen trees of different sizes: 10 sites, 22 sites, and 46 sites. We focus on two aspects: energy convergence with M , and CPU time per sweep. In general, the energy convergence of the TTNS is much faster than the MPS, both with increasing M and both as a function of CPU time, as seen in Figure 13 for the 22 site tree. (In the 10 site tree, the strange behaviour of the CPU time, where larger M required smaller CPU time, is an artifact of our Davidson diagonalization implementation, which required a large number of iterations when M was very small.) In the 46 site tree, we only carried out calculations with small M because of the large computational cost. For the CPU time per sweep, we found that a calculation with MPS with $M = 200$ (6182 s/sweep), and TTNS with $M = 100$ (6839 s/sweep) were comparable. However, the corresponding energies were computed to be $-22.705014E_h$ and $-22.958219E_h$, respectively, and thus the TTNS gave a much better energy than the MPS at

a comparable computational cost. These results demonstrate that TTNS is more cost effective than MPS when the system is tree-shaped. Moreover, the relative benefits of the TTNS increase as the tree size increases.

We now consider a model system designed to mimic more difficult molecular structures for MPS and TTNS. In the previous tree structures we used a local atomic basis. This, together with the underlying tree connectivity of the molecule, allowed the TTNS to completely exploit the local nature of correlation in the system. Though the system was not linear, the MPS still benefitted from the local basis in the calculations, as not every orbital was correlated with every other. (For similar reasons, in DMRG calculations localized molecular orbitals are often used to minimize long-range entanglement.) However, in some situations, it may be unavoidable for a MPS or TTNS to describe some long-range entanglement, either due to the underlying physics, or due to a poor mapping of the molecular structure onto a chain or tree. To mimic this situation we carry out MPS and TTNS calculations in a *canonical* molecular orbital basis. Because these orbitals are delocalized, each orbital can be considered to interact with each other. We choose as our model system a set of hydrogen chains in a minimal STO-3G

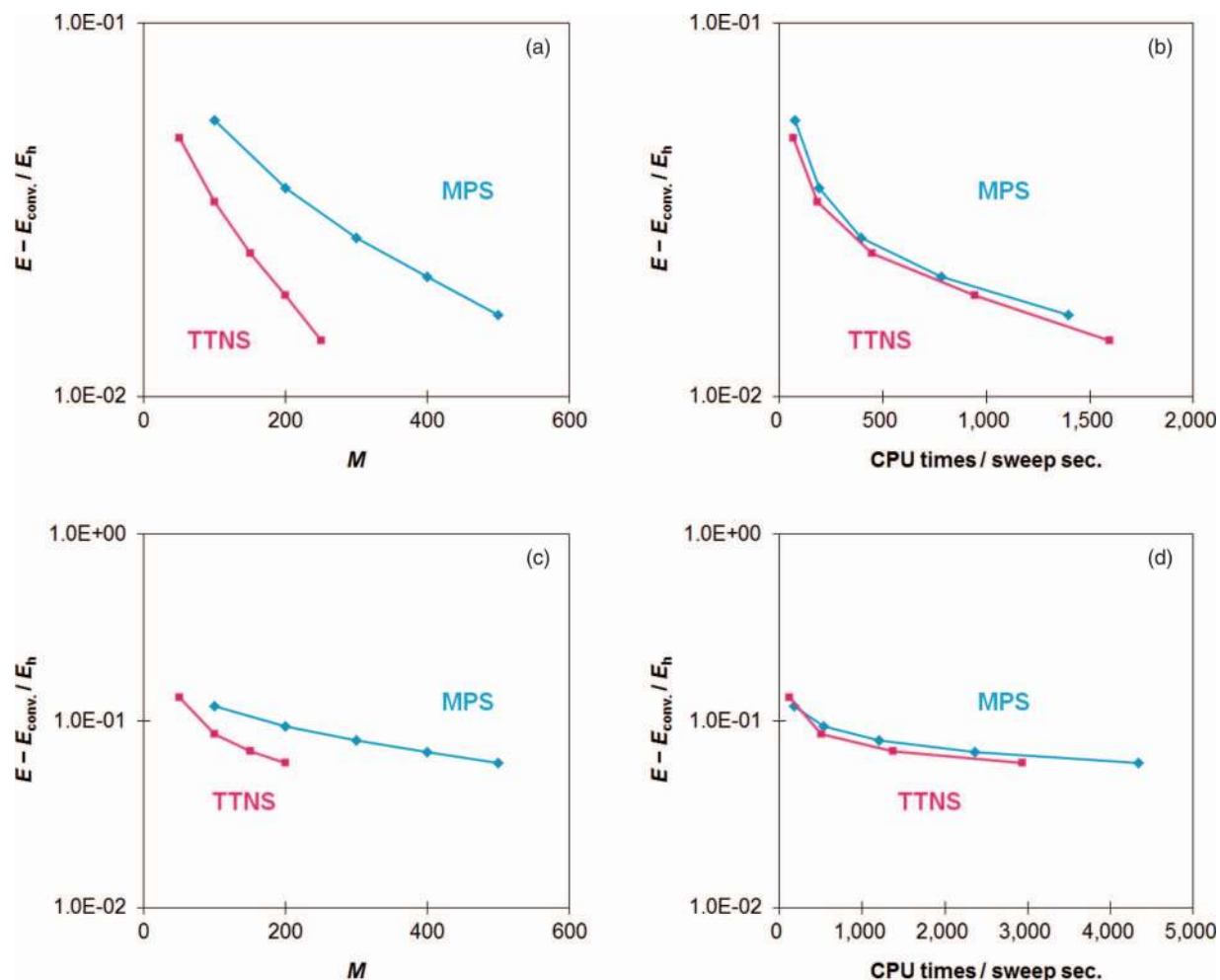


FIG. 14. Energy convergence of hydrogen chains with canonical molecular orbitals (CMOs) plotted by number of renormalized states M and CPU time (s) per one sweep; (a) and (b) showed those of M and CPU time per sweep, respectively, for 20 sites, and (c) and (d) showed those of M and CPU time per sweep, respectively, for 30 sites. Singlet ground state energy was computed with STO-3G basis sets.

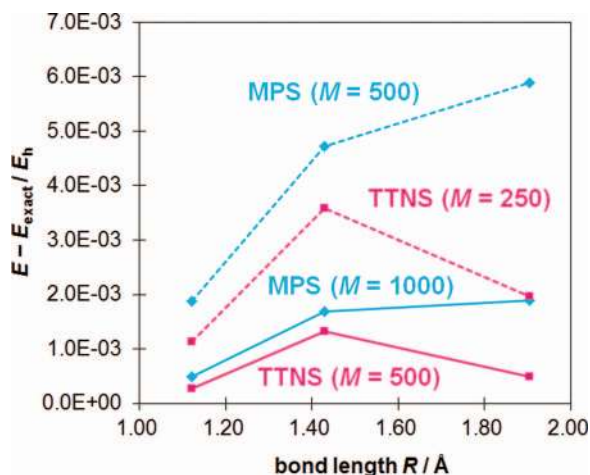


FIG. 15. Energy errors for bond dissociation of nitrogen dimer using (10e, 26o) active space and cc-pVDZ basis sets. Reference energies were from previous full-CI work in Ref. 41.

basis. Although these are chain molecules, the use of canonical molecular orbitals means that this is not an ideal system for the MPS.

Figure 14 shows the energy convergence as a function of M and CPU time per sweep. Both MPS and TTNS exhibit very slow energy convergence as a function of M due to the use of canonical molecular orbitals, but it is clear that the convergence with M is always better in TTNS than in MPS by a factor of 2 or more. As a function of CPU time, we find that the decreased M means that TTNS performs better in comparison with MPS, although the improvement is slight due to the higher scaling of TTNS with M .

These results provide some promise that in general molecules without a tree or chain structure, TTNS can perform better than MPS, although this involves a delicate balance between the decreased number of renormalized states used by the TTNS, and the higher computational scaling with M . To examine this in a more realistic setting, we now consider calculations on two benchmark diatomics, the nitrogen dimer and the chromium dimer.

TABLE III. CPU times per sweep in s for MPS and TTNS calculations for the nitrogen dimer using (10e, 26o) active space and cc-pVDZ basis sets.

$R/\text{\AA}$	MPS		TTNS	
	$M = 500$	$M = 1000$	$M = 250$	$M = 500$
1.1208	706	2998	794	5088
1.4288	1027	4684	1337	7510
1.9050	1155	5053	992	5945

The bond dissociation curve of nitrogen dimer is often used as a good benchmark to evaluate whether a method can describe strong electron correlation correctly. We evaluated ground state energies at three points on the bond dissociation of nitrogen dimer, 1.1208(R_c), 1.4288, and 1.9050 Å. A frozen core active space (10e, 26o) with a cc-pVDZ basis set was employed as used in previous DMRG calculations⁴ and full-CI (configuration interaction) calculations⁴¹ (see supplementary material for the tree graphs and the site orderings⁴⁵). Figure 15 shows the energy errors from full-CI results⁴¹ as a function of bond length and CPU times per sweep are summarized in Table III. In TTNS, half the M can be used at $R = 1.1208$ and 1.4288 Å and a quarter the M can be used at $R = 1.9050$ Å as compared with MPS. TTNS with half the M of the corresponding MPS required approximately the same CPU time per sweep. This indicates that the TTNS gives comparable and/or slightly better performance at short bond-lengths but much better performance at long bond-lengths compared to the MPS in this molecule. We conclude that the TTNS works better in the case of the nitrogen dimer than the MPS.

Since the chromium dimer has an unusual multiple bond, its ground state is very complicated and difficult to describe by conventional methods. Recently, DMRG calculations of the chromium dimer have been carried out for relatively large active spaces.^{9,42} We performed MPS and TTNS calculation at 1.5 Å using the same basis sets and the same active space (24e, 30o) as in earlier DMRG calculations⁴² (see supplementary material for the tree graph and the site ordering⁴⁵). Figure 16 shows the energy convergence as a function of M

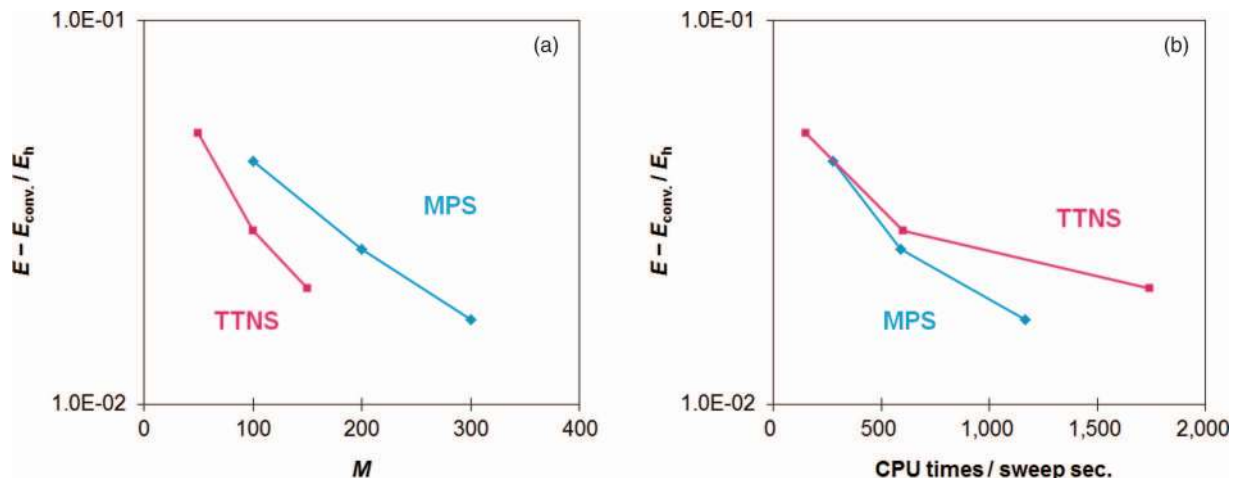


FIG. 16. Energy convergence of the chromium dimer at 1.5 Å using a (24e, 30o) active space and Ahlrichs' split-valence plus polarization (SVP) basis set. Plotted as a function of M and (b) as a function of CPU time per sweep. Reference energies were from previous DMRG work in Ref. 42.

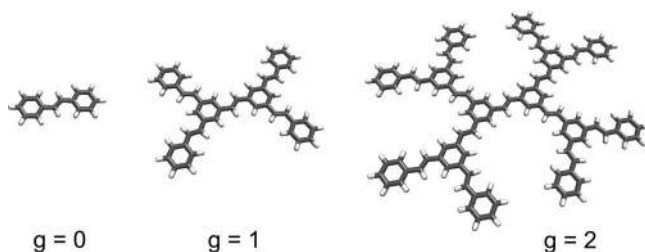


FIG. 17. Structures of stilbenoid dendrimers $g = 0, 1,$ and 2 . Geometries were optimized at the B3LYP/cc-pVDZ level of theory.

and CPU time per sweep. Since the previous DMRG calculation employed very large M (up to 10 000), the energy reported here is far from convergence because we only employed M up to 150 in the TTNS. Nonetheless compared to our MPS benchmarks, we find that smaller M can be used in the TTNS. In comparing CPU time per sweep, however, we see that MPS gives much better performance than the TTNS in this molecule. Thus, we conclude that the MPS works better in the case of the chromium dimer than the TTNS.

These two benchmark calculations on the nitrogen dimer and chromium dimer show that the performance of the MPS and TTNS in *general* molecules depends sensitively on the electronic structure and the nature of the quantum entanglement of the molecule, thus their relative merits must be determined on a molecule by molecule basis.

Finally, to demonstrate the power of TTNS, we consider a TTNS calculation on more realistic tree-shaped molecules. Stilbenoid dendrimers are prototypical π -conjugated dendrimers, with each unit being a stilbene fragment.⁴³ These dendrimers are potentially attractive for chemistry because

photo-induced electron-transfer or exciton-transfer may proceed from the leaves to the core, mimicking a biological antenna system (we note that dendrimer systems have also been previously studied using semi-empirical DMRG^{26,44}). Although it is the excited states and dynamical properties of these systems that are of primary interest, here we focus on the ground-state energy for benchmarking and reserve the study of excited states to future work.

We consider three different sizes of stilbenoid dendrimers denoted by the generation g as shown in Figure 17. We carried out π -full valence MPS and TTNS calculations with STO-3G and 6-31G basis sets. Single valence calculations were performed for $g = 0, 1,$ and 2 with (14e, 14o), (46e, 46o), and (110e, 110o) active-spaces, respectively, and double valence calculations were performed for $g = 0$ and 1 with (14e, 28o) and (46e, 92o) active-spaces, respectively. To construct the tree graph, π -orbitals computed from a RHF calculation were localized for occupied and unoccupied spaces separately. Localized MOs were grouped for each ethylene and benzene fragment, and were ordered on the tree according to the underlying dendritic structure, as shown in Figure 18. The orderings within each fragment were determined to put strongly interacting pairs (evaluated by K_{ij}) on neighboring sites.

For the $g = 0$ dendrimer (stilbene), the MPS and TTNS gave similar energy convergence as a function of M in the single-valence (STO-3G) calculation, and the MPS gave somewhat better performance with respect to CPU time per sweep (see Figures 19(a) and 19(b)). In the double-valence (6-31G) calculation, the TTNS gave an improved energy convergence with M compared to MPS, and the MPS and TTNS gave almost the same performance with respect to CPU time

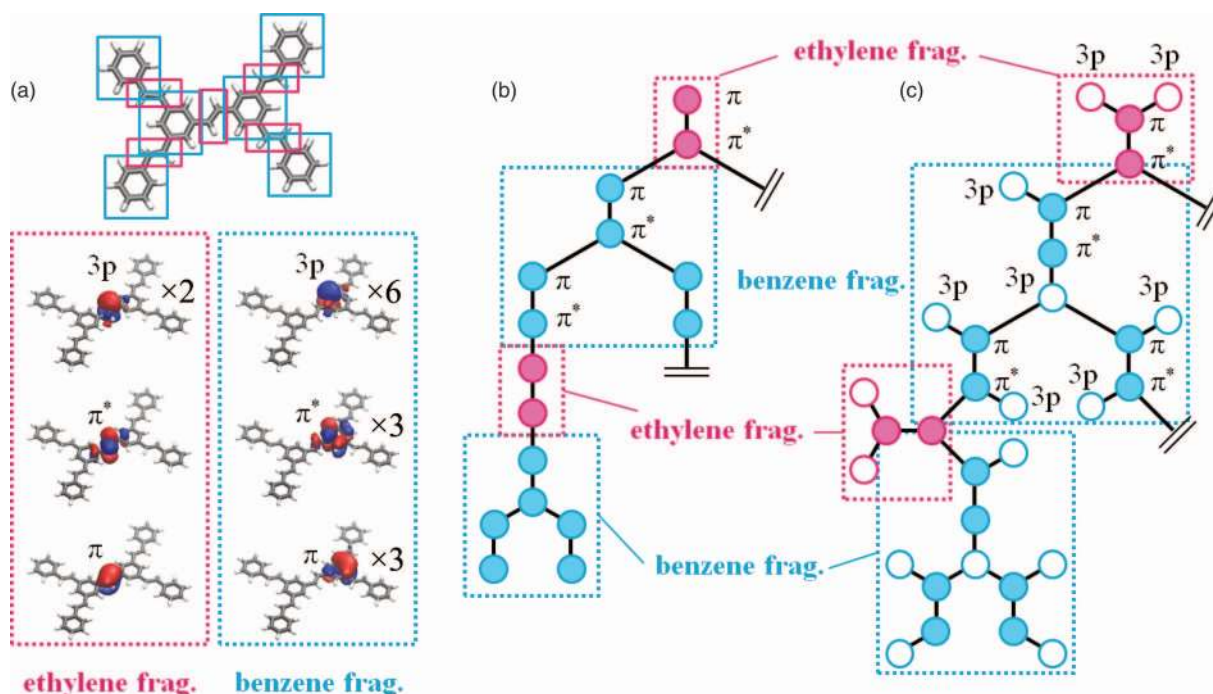


FIG. 18. Tree graphs for TTNS calculation of stilbenoid dendrimer ($g = 1$). (a) Selected molecular orbitals of the localized ethylene and benzene fragments, (b) Tree graph of the single-valence (STO-3G) calculation, and (c) tree graph of the double-valence (6-31G) calculation where the filled circles represent local π and π^* orbitals and empty circles represent local $3p$ orbitals. Note that identical orbitals are omitted in the tree graphs.

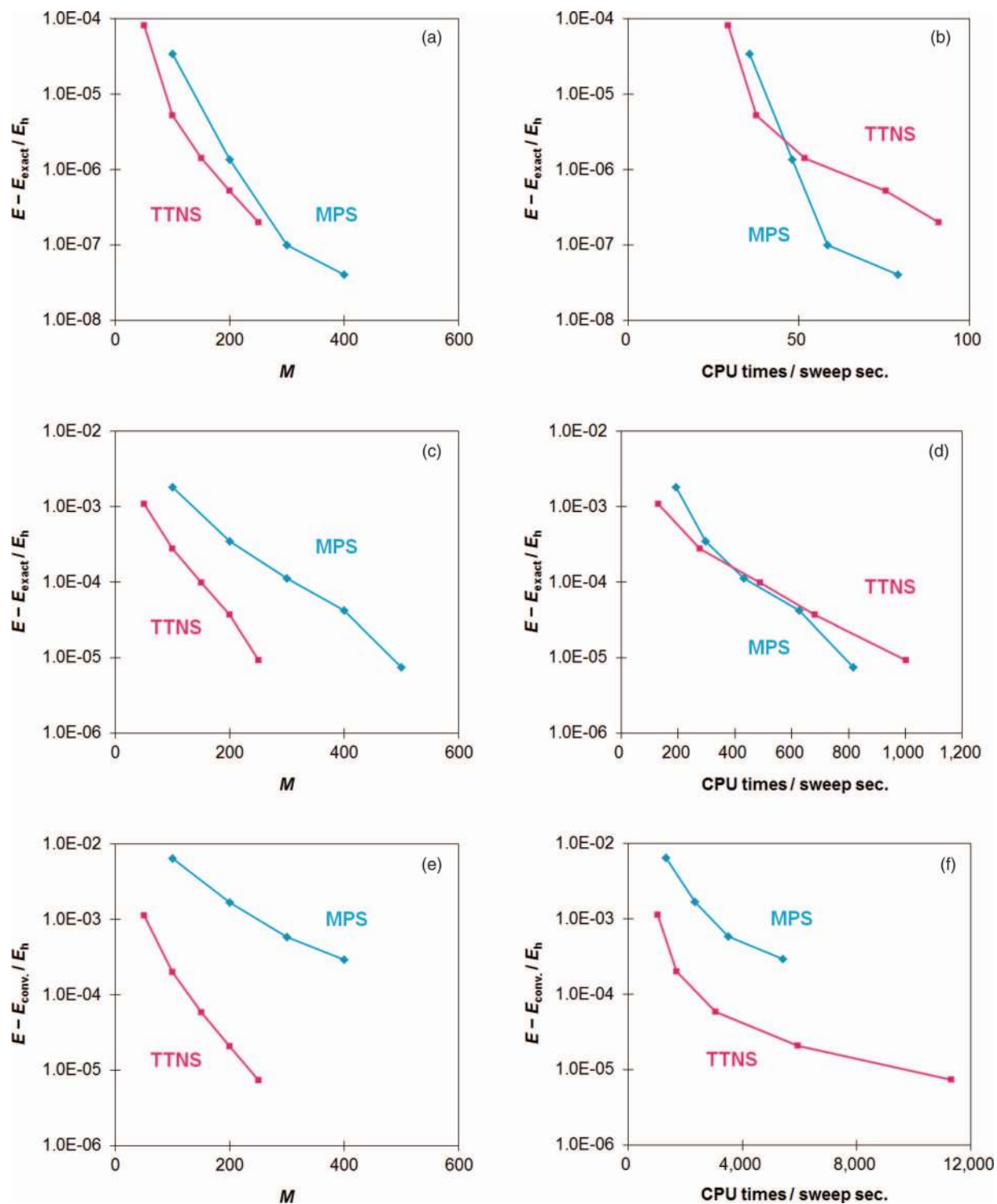


FIG. 19. Energy convergence of stilbenoid dendrimers $g = 0$ and $g = 1$ plotted by number of renormalized states M and CPU time (sec.) per sweep; (a) and (b) shows energy versus M and CPU time per sweep, respectively, for $g = 0$, single valence (14e, 14o), (c) and (d) shows energy versus M and CPU time per sweep, respectively, for $g = 0$, double valence (14e, 28o), (e) and (f) shows energy versus M and CPU time per sweep, respectively for $g = 1$, single valence (46e, 46o). All energies are for the singlet ground-state energy.

per sweep (see Figures 19(c) and 19(d)). The competitive performance of MPS relative to TTNS in the $g = 0$ dendrimer reflects the very small size of the system, which is almost linear in nature and thus nearly ideal for MPS.

For the larger $g = 1$ dendrimer, in the single-valence space the TTNS required a quarter the M to obtain the

same energy accuracy as the MPS. The computational cost of the TTNS for this accuracy was also much lower than that of the MPS (see Figures 19(e) and 19(f)). The double-valence calculations were very expensive, hence we only performed calculations with $M = 100$ for MPS and with $M = 50$ for TTNS. Although the resulting energies were not converged to

chemical accuracy, the correlation energies ($E - E_{\text{HF}}$) of the MPS, $-0.644736E_h$, and the TTNS, $-0.678678E_h$, indicate that the TTNS is once again much more accurate.

Finally, for the $g = 2$ dendrimer, even the single-valence active space calculation was quite expensive with our implementation, consequently, we only performed single-valence calculations using $M = 100$ for MPS and $M = 50$ for TTNS. These calculations had comparable timings. The computed correlation energies were $-1.943651E_h$ and $-2.021378E_h$ with MPS and TTNS, respectively, indicating once again that the TTNS works much better than the MPS in this large dendritic molecule. Although we could not reach fully converged energies in this work, the ability to even approximately target such systems with TTNS demonstrates the promise of the technique for complex systems.

V. CONCLUSIONS

In this work we investigated tree tensor network states (TTNS) for quantum chemistry. We formulated an efficient tree tensor network algorithm that is analogous to the density matrix renormalization group (DMRG) algorithm in quantum chemistry for matrix product states (MPS). We introduced the additional step of half-renormalization that greatly reduced the computation cost. We found that our TTNS calculations were competitive with MPS and DMRG calculations in general molecules, requiring significantly fewer renormalized states for the same accuracy, although this did not always translate into a savings in computational time. In tree like molecules, TTNS was clearly superior to MPS requiring both fewer renormalized states and less time to reach the same accuracy. This bodes well for the application of TTNS to study a wide class of interesting optically active systems based on dendritic structures, as illustrated in our calculations on stilbenoid dendrimers.

Tree tensor network states are one of the simplest generalizations of the matrix product states, because the tree network structure has no cycles. More complex tensor networks which describe even more general entanglement require the treatment of cycles. The improvements observed with tree tensor networks here suggests that investigating these more complex classes will be fruitful, particularly to describe quantum chemistry of larger systems, where the molecules or materials acquire an extended two-dimensional and three-dimensional structure.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) through Grant No. NSF-OCI-1148287 and NSF-CHE-1213933.

APPENDIX: OPTIMAL TENSOR CONTRACTION FOR TTNS RENORMALIZATION

During the TTNS renormalization procedure, it is necessary to construct matrix representations of operators act on $b_i^1 \dots b_i^{Z-1} n_i = |n_i\rangle |b_i^{Z-1}\rangle \dots |b_i^1\rangle$. Because of the quartic terms

in the Hamiltonian, this can involve products of up to four operators. The corresponding product formulae follow.

The matrix representation of an operator \hat{O}_α which acts only on renormalized states in block α is given by

$$\begin{aligned} [\hat{O}_\alpha]^{b_i^1 \dots b_i^{Z-1} n_i, b_i^1 \dots b_i^{Z-1} n_i} &= \langle b_i^1 | \dots \langle b_i^{Z-1} | \langle n_i | \hat{O}_\alpha | n_i \rangle | b_i^{Z-1} \rangle \dots | b_i^1 \rangle \\ &= [\hat{O}_\alpha]^{b_i^\alpha b_i^\alpha} \times \hat{P}(\langle b_i^{\alpha+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\alpha), \end{aligned} \quad (\text{A1})$$

where $[\hat{O}_\alpha]^{b_i^\alpha b_i^\alpha}$ is the matrix representation of \hat{O}_α spanned by states $|b_i^\alpha\rangle$ and \hat{P} is a parity operator which gives $+1$ or -1 depending on the particle numbers of $\langle b_i^{\alpha+1} | \dots \langle b_i^{Z-1} | n_i \rangle$ and \hat{O}_α . Similarly, those of operators $\hat{O}_\alpha \hat{O}_\beta$, $\hat{O}_\alpha \hat{O}_\beta \hat{O}_\gamma$, and $\hat{O}_\alpha \hat{O}_\beta \hat{O}_\gamma \hat{O}_\delta$ are computed as follows,

$$\begin{aligned} [\hat{O}_\alpha \hat{O}_\beta]^{b_i^1 \dots b_i^{Z-1} n_i, b_i^1 \dots b_i^{Z-1} n_i} &= [\hat{O}_\alpha]^{b_i^\alpha b_i^\alpha} [\hat{O}_\beta]^{b_i^\beta b_i^\beta} \\ &\times \hat{P}(\langle b_i^{\alpha+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\alpha) \\ &\times \hat{P}(\langle b_i^{\beta+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\beta), \end{aligned} \quad (\text{A2})$$

$$\begin{aligned} [\hat{O}_\alpha \hat{O}_\beta \hat{O}_\gamma]^{b_i^1 \dots b_i^{Z-1} n_i, b_i^1 \dots b_i^{Z-1} n_i} &= [\hat{O}_\alpha]^{b_i^\alpha b_i^\alpha} [\hat{O}_\beta]^{b_i^\beta b_i^\beta} [\hat{O}_\gamma]^{b_i^\gamma b_i^\gamma} \\ &\times \hat{P}(\langle b_i^{\alpha+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\alpha) \\ &\times \hat{P}(\langle b_i^{\beta+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\beta) \\ &\times \hat{P}(\langle b_i^{\gamma+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\gamma), \end{aligned} \quad (\text{A3})$$

$$\begin{aligned} [\hat{O}_\alpha \hat{O}_\beta \hat{O}_\gamma \hat{O}_\delta]^{b_i^1 \dots b_i^{Z-1} n_i, b_i^1 \dots b_i^{Z-1} n_i} &= [\hat{O}_\alpha]^{b_i^\alpha b_i^\alpha} [\hat{O}_\beta]^{b_i^\beta b_i^\beta} [\hat{O}_\gamma]^{b_i^\gamma b_i^\gamma} [\hat{O}_\delta]^{b_i^\delta b_i^\delta} \\ &\times \hat{P}(\langle b_i^{\alpha+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\alpha) \\ &\times \hat{P}(\langle b_i^{\beta+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\beta) \\ &\times \hat{P}(\langle b_i^{\gamma+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\gamma) \\ &\times \hat{P}(\langle b_i^{\delta+1} | \dots \langle b_i^{Z-1} | \langle n_i |, \hat{O}_\delta). \end{aligned} \quad (\text{A4})$$

Here, the representation of operator $[\hat{O}]^{b_i^1 \dots b_i^{Z-1} n_i, b_i^1 \dots b_i^{Z-1} n_i}$ is a $4M^Z \times 4M^Z$ matrix. In the

TABLE IV. Complexity of optimal tensor contractions for TTNS renormalization per site. Multiplying by $\mathcal{O}(k)$ gives complexity per sweep.

Complementary operator	Complexity
\hat{H}	$\mathcal{O}(M^{Z+1}k^2 + M^Zk^4)$
\hat{a}_i	$\mathcal{O}(M^{Z+1}k)$
\hat{S}_i	$\mathcal{O}(M^{Z+1}k^2 + M^Zk^4)$
$\hat{a}_i \hat{a}_j$	$\mathcal{O}(M^{Z+1}k^2)$
$\hat{a}_i^\dagger \hat{a}_j$	$\mathcal{O}(M^{Z+1}k^2)$
\hat{P}_{ij}	$\mathcal{O}(M^{Z+1}k^2 + M^2k^4)$
\hat{Q}_{ij}	$\mathcal{O}(M^{Z+1}k^2 + M^2k^4)$

renormalization step, this is reduced to $M \times M$ matrix $[\hat{O}]^{b_i^Z b_i^Z}$, spanned by renormalized states $|b_i^Z\rangle$ using a tensor A^{n_i} as

$$\sum_{\substack{b_i^1 \dots b_i^{Z-1} n_i \\ b_i^1 \dots b_i^{Z-1} n_i}} A_{b_i^1 \dots b_i^{Z-1}}^{n_i^\dagger} [\hat{O}]_{b_i^1 \dots b_i^{Z-1} n_i^1 b_i^1 \dots b_i^{Z-1} n_i} A_{b_i^1 \dots b_i^Z}^{n_i} \quad (\text{A5})$$

It is necessary to minimize the cost of tensor contractions for TTNS renormalization (A5) is just as for $\hat{H}\psi$, as described in the text. To compute representations of complementary operators, each complementary operator can be further divided into $Z - 1$ renormalized blocks and the site i itself. We carefully minimized these tensor contraction costs for each complementary operator and their complexities per site are summarized in Table IV for a general tree. Thus, for a general tree, the computational complexity of a TTNS sweep is $\mathcal{O}(M^{Z+1}k^3 + M^Zk^5)$. As discussed in the text, the complexity can be further reduced to $\mathcal{O}(M^3k^3 + M^2k^4)$ for $Z = 2$ (MPS) and $\mathcal{O}(M^4k^3 + M^2k^5)$ for $Z = 3$.

¹S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).

²S. R. White, *Phys. Rev. B* **48**, 10345 (1993).

³S. R. White and R. L. Martin, *J. Chem. Phys.* **110**, 4127 (1999).

⁴G. K.-L. Chan and M. Head-Gordon, *J. Chem. Phys.* **116**, 4462 (2002).

⁵G. K.-L. Chan and S. Sharma, *Annu. Rev. Phys. Chem.* **62**, 465 (2011).

⁶Ö. Legeza, J. Röder, and B. A. Hess, *Phys. Rev. B* **67**, 125114 (2003).

⁷G. Moritz and M. Reiher, *J. Chem. Phys.* **126**, 244109 (2007).

⁸K. H. Marti and M. Reiher, *Z. Phys. Chem.* **224**, 583 (2010).

⁹Y. Kurashige and T. Yanai, *J. Chem. Phys.* **130**, 234114 (2009).

¹⁰J. Hachmann, W. Cardoen, and G. K.-L. Chan, *J. Chem. Phys.* **125**, 144101 (2006).

¹¹D. Ghosh, J. Hachmann, T. Yanai, and G. K.-L. Chan, *J. Chem. Phys.* **128**, 144117 (2008).

¹²Y.-Y. Shi, L.-M. Duan, and G. Vidal, *Phys. Rev. A* **74**, 022320 (2006).

¹³G. Vidal, *Phys. Rev. Lett.* **99**, 220405 (2007).

¹⁴F. Verstraete, V. Murg, and J. I. Cirac, *Adv. Phys.* **57**, 143 (2008).

¹⁵K. H. Marti and M. Reiher, *Phys. Chem. Chem. Phys.* **13**, 6750 (2011).

¹⁶Ö. Legeza, T. Rohwedder, and R. Schneider, "Numerical approaches for high-dimensional PDE's for quantum chemistry," in *Encyclopedia of Applied and Computational Mathematics*, edited by B. Engquist (Springer, 2012).

¹⁷G. K.-L. Chan, *Comput. Mol. Sci.* **2**, 907 (2012).

¹⁸L. Tagliacozzo, G. Evenbly, and G. Vidal, *Phys. Rev. B* **80**, 235127 (2009).

¹⁹V. Murg, F. Verstraete, O. Legeza, and R. M. Noack, *Phys. Rev. B* **82**, 205105 (2010).

²⁰W. Li, J. Delft, and T. Xiang, *Phys. Rev. B* **86**, 195137 (2012).

²¹H. J. Changlani, S. Ghosh, C. L. Henley, and A. Läuchli, *Phys. Rev. B* **87**, 085107 (2013).

²²H. Otsuka, *Phys. Rev. B* **53**, 14004 (1996).

²³B. Friedman, *J. Phys.: Condens. Matter* **9**, 9021 (1997).

²⁴R. J. Bursill, *Phys. Rev. B* **60**, 1643 (1999).

²⁵M. B. Lepetit, M. Cousy, and G. M. Pastor, *Eur. Phys. J. B* **13**, 421 (2000).

²⁶M. A. Martin-Delgado, J. Rodriguez-Laguna, and G. Sierra, *Phys. Rev. B* **65**, 155116 (2002).

²⁷M. Kumar, S. Ramasesha, and Z. G. Soos, *Phys. Rev. B* **85**, 134415 (2012).

²⁸U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).

²⁹U. Schollwöck, *Ann. Phys.* **326**, 96 (2011).

³⁰T. Xiang, *Phys. Rev. B* **53**, 10445 (1996).

³¹F. Verstraete, J. J. Garcia-Ripoll, and J. I. Cirac, *Phys. Rev. Lett.* **93**, 207204 (2004).

³²I. P. McCulloch, *J. Stat. Mech.: Theory Exp.* **2007**, P10014.

³³G. M. Crosswhite and D. Bacon, *Phys. Rev. A* **78**, 012356 (2008).

³⁴B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New J. Phys.* **12**, 025012 (2010).

³⁵F. Frowis, V. Nebendahl, and W. Dur, *Phys. Rev. A* **81**, 062337 (2010).

³⁶S. R. White, *Phys. Rev. B* **72**, 180403 (2005).

³⁷Ö. Legeza and J. Sólyom, *Phys. Rev. B* **68**, 195116 (2003).

³⁸J. Rissler, R. M. Noack, and S. R. White, *Chem. Phys.* **323**, 519 (2006).

³⁹G. Barcza, Ö. Legeza, K. H. Marti, and M. Reiher, *Phys. Rev. A* **83**, 012508 (2011).

⁴⁰S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithm* (Springer-Verlag, Berlin/Heidelberg, 2010).

⁴¹H. Larsen, J. Olsen, P. Jørgensen, and O. Christiansen, *J. Chem. Phys.* **113**, 6677 (2000).

⁴²S. Sharma and G. K.-L. Chan, *J. Chem. Phys.* **136**, 124121 (2012).

⁴³H. Meier and M. Lehmann, *Angew. Chem., Int. Ed.* **37**, 643 (1998).

⁴⁴S. Mukhopadhyay, B. J. Topham, Z. G. Soos, and S. Ramasesha, *J. Phys. Chem. A* **112**, 7271 (2008).

⁴⁵See supplementary material at <http://dx.doi.org/10.1063/1.4798639> for the geometries of hydrogen trees, and the tree graphs of nitrogen dimer and chromium dimer.