

Tensor-structured coupled cluster theory

Roman Schutski,¹ Jinmo Zhao,¹ Thomas M. Henderson,^{1,2} and Gustavo E. Scuseria^{1,2}

¹Department of Chemistry, Rice University, Houston, Texas 77251-1892, USA

²Department of Physics and Astronomy, Rice University, Houston, Texas 77251-1892, USA

(Received 20 July 2017; accepted 26 October 2017; published online 14 November 2017)

We derive and implement a new way of solving coupled cluster equations with lower computational scaling. Our method is based on the decomposition of both amplitudes and two electron integrals, using a combination of tensor hypercontraction and canonical polyadic decomposition. While the original theory scales as $O(N^6)$ with respect to the number of basis functions, we demonstrate numerically that we achieve sub-millihartree difference from the original theory with $O(N^4)$ scaling. This is accomplished by solving directly for the factors that decompose the cluster operator. The proposed scheme is quite general and can be easily extended to other many-body methods. *Published by AIP Publishing.* <https://doi.org/10.1063/1.4996988>

I. INTRODUCTION

Many basic building blocks of quantum theories are tensors. Examples include the one- and two-electron integrals defining the Hamiltonian or the cluster operators of coupled cluster (CC) theory, which instead define the wave function. Unfortunately, algebraic manipulations with tensors have a significant numerical cost, which tends to grow exponentially with the dimension d of the tensors and often makes these manipulations the computational bottleneck of the theories.

The cost of tensor manipulations can be significantly reduced by some form of tensor decomposition in which a d -dimensional tensor is expressed in terms of lower dimensional objects. For example, the resolution of identity (RI, see Ref. 1 and references therein) can be used to decompose the two-electron integrals. More recently, the tensor hypercontraction²⁻⁷ (THC) scheme of Hohenstein, Parrish, and Martínez has been introduced. There, a fourth-order tensor is represented by a contraction of five factor matrices, some of which can be the same if one wants to enforce symmetries of the original tensor. Related to THC is the canonical polyadic decomposition^{8,9} (CPD), which as we will explain later can be regarded as its building block.

These tensor decompositions have been used in various ways to introduce low-scaling versions of conventional electronic structure methods. Tensor hypercontraction has been applied by Hohenstein and Kokkila¹⁰ to the CC2 method, where it was used to represent electron interaction potential. Shenvi *et al.* did the same in their reduced density matrix algorithm.¹¹ Benedict *et al.* used polyadic decomposition of amplitudes and electron interaction integrals in the coupled cluster doubles and full configuration interaction methods.^{12,13} While working on this manuscript we also learned about the recent work of Hummel *et al.*,¹⁴ who showed that by using the THC of the electron interaction in the context of the distinguishable cluster doubles or linearized coupled cluster singles and doubles methods, one can achieve a reduction of the computational cost from $O(N^6)$ to $O(N^5)$, where N is the number of basis functions.

Here we apply tensor decompositions based on THC to coupled cluster with single and double (CCSD) excitations.^{15,16} The cost of the original CCSD scales as $O(N^6)$, but by using tensor decomposition, we can reduce the cost to scale as $O(N^4)$. In most previous applications, THC was used to decompose the electron repulsion integrals, and grids in real space were employed to build the decomposition. We show how to build the THC algebraically for the full fourth-order tensor in $O(N^5)$ cost, or $O(N^4)$ cost if the resolution of identity¹⁷ is employed, and compare different ways of doing so. We also explain how to optimize all factors of the THC in $O(N^4)$ cost when solving iterative equations with decomposed tensors, such as in the CCSD method. By optimizing all factors of the THC, our implementation achieves the same ~ 0.5 mhartree accuracy as the previous work⁴ which used THC but with ranks which are roughly half as large. However, we should emphasize that our method is general and is not limited to THC; rather, it can be used with *any* suitable tensor decomposition.

We note that there are even more efficient techniques for approximate CC methods in the weakly correlated regime, namely, the linear scaling ($O(N)$) CC methods.^{18,19} These methods tend to be based on the locality of both the electron interaction integrals and the cluster amplitudes.²⁰ If either of these assumptions fails (as might happen in strongly correlated problems, for example, which couple electrons at large distances), linear scaling methods are likely to see a corresponding decrease in performance. Of course coupled cluster theory is not ideally suited to the description of strong correlation anyway, so this is not too significant a drawback. The goal of this work, however, is not only to derive a lower-scaling CC approach but also to present a framework for applying tensor decompositions to other CC-like theories²¹⁻²³ which are not limited to weakly correlated problems.

For completeness, we should note the existence of other coupled cluster-like techniques built on Jastrow Hilbert space approaches where the equivalent of the CC excitation operator is built from occupation number operators in variational quantum Monte Carlo²⁴ and similarity transform approaches.^{25,26}

In these cases, the tensor factorization occurs naturally and can be determined by simply inspecting the numerical amplitude values of the correlator.

II. NOTATION AND TERMINOLOGY

Throughout this work, we will use notation and diagrams which are common in the literature of tensor decompositions but which may be unfamiliar to the quantum chemistry community. A short review of our diagrammatic notation is available in the [Appendix](#); we summarize our notation and terminology here.

One of the most basic properties of a tensor T is its *order*, which is just its dimensionality and corresponds to the number of indices in its basis representation. Thus, a four-index object (if a tensor) corresponds with a fourth-order tensor. We sometimes refer to a first-order tensor as a vector and a second-order tensor as a matrix. Generically we denote matrices and tensors by capital letters and vectors by boldface lowercase letters.

The *rank* of a tensor is the dimension of the auxiliary indices used in a particular tensor decomposition. As there are a great variety of tensor decompositions, the rank of a high-order tensor is not defined as strictly as in the case of matrices and may consist of one number or a set of numbers; for our purposes, if the tensor has more than one rank, it is convenient to require all its ranks to be equal. Different definitions of the tensor rank have significantly different properties; for more information, consult the review of Kolda and Bader.²⁷ It should be clear from the context what dimensions are meant in each particular case in the text.

The Frobenius norm of a tensor T is denoted as $\|T\|$ and is given by

$$\|T\| = \sqrt{\sum_{pqrs\dots} T_{pqrs\dots} T_{pqrs\dots}^*}, \quad (1)$$

where the superscript $*$ denotes complex conjugation; thus, it is simply the square root of the sum of the norm squares of tensor's entries.

We will require a few kinds of tensor product in this work. The Kronecker product is written as \otimes and is defined via

$$C = A \otimes B \Leftrightarrow C_{rp,sq} = A_{p,q} \cdot B_{r,s}. \quad (2)$$

It is also convenient to introduce a column-wise Kronecker product known as the Khatri-Rao²⁸ product; this is denoted by \odot and is defined as

$$D = A \odot B \Leftrightarrow D_{qp,\alpha} = A_{p,\alpha} \cdot B_{q,\alpha}. \quad (3)$$

In the foregoing and throughout this manuscript, indices p, q, r, s, \dots correspond to general orbital labels and Greek letters $\alpha, \beta, \gamma, \dots$ denote indices of the CPD, THC, and singular value decompositions. We follow the traditional notation that the indices i, j, k, l, \dots represent occupied orbitals specifically, while a, b, c, d, \dots represent virtual orbitals. We also use composite indices such as pq which are defined as

$$pq \equiv p + \dim(\{p\}) \cdot (q - 1). \quad (4)$$

Curly braces $\{ \}$ denote sets and $\dim()$ means the number of elements in the set.

The transpose of a matrix M is M^T , and its inverse is M^{-1} ; if M is singular or not square, M^{-1} refers to the pseudoinverse²⁹ of M . We will use $\text{sqrt}()$ for the element-wise square root operation

$$\text{sqrt}(M)_{pq} = \sqrt{M_{pq}}. \quad (5)$$

III. TENSOR DECOMPOSITIONS

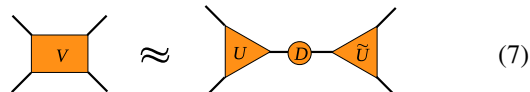
The tensor hypercontraction decomposition can be regarded as a combination of two well-established factorizations: a rank decomposition of a matrix such as the eigenvalue or singular value decomposition (SVD) on the one hand and the canonical polyadic decomposition^{8,9,27} of third order tensors on the other. Thus, we first discuss these two ideas.

A. Resolution of identity and singular value decomposition

The computation of a rank-revealing decomposition for the electron interaction tensor is well studied and is known as the resolution of identity (RI) or density fitting.³⁰⁻³³ By introducing an auxiliary basis, the Coulomb interaction can be written as a contraction of three tensors,

$$V_{pqrs} \approx \sum_{\alpha\alpha'} U_{pq}^{\alpha} D_{\alpha,\alpha'} \tilde{U}_{rs}^{\alpha'}, \quad (6)$$

where V is the Mulliken-ordered interaction, U and \tilde{U} are (possibly different) three index integrals, and D is a generalized overlap.¹⁷ Diagrammatically the same expression is



As one may see, the RI has the same basic form of a singular value or an eigenvalue decomposition of the interaction tensor. It is known that the error in the RI approximation of the Coulomb operator decays exponentially with the auxiliary basis size $r_{\text{RI}} = \dim(\{\alpha\})$, and negligible errors can be reached with the number of auxiliary basis functions scaling as $O(N)$.^{30,33}

We note that for a given rank r_{RI} , the lowest error RI decomposition can be calculated using the singular value decomposition of the matrix $V_{pq,rs}$ and taking D , U , and \tilde{U} to be the singular values and left and right singular vectors, respectively. The optimality of the factorization will then be guaranteed by the Eckart-Young theorem.³⁴ Although this approach is not generally used for practical calculations due to its computational cost, which scales as $O(N^4 \cdot r_{\text{RI}})$, we employed it in some of our test calculations. We also note that a popular practical option in the case of two electron integrals is the use of the Cholesky decomposition,^{1,35,36} but this method is limited to symmetric tensors only.

We have said that V is the Mulliken-ordered interaction tensor. The restriction to Mulliken ordering is important because the order of indices in the original tensor $V_{pq,rs}$ crucially influences the size of the rank r_{RI} for a fixed approximation error. Indeed, while the SVD of the Mulliken-ordered electron interaction $V_{pq,rs}$ yields $O(N)$ non-zero singular values, the matrix $V_{pr,qs}$ formed from a Dirac-ordered interaction

tensor has $O(N^2)$ non-zero singular values. This explains why there is no practical RI-like approximation for Dirac-ordered two-electron integrals (or, equivalently, the exchange contribution in the context of the Hartree-Fock method). This is a natural consequence of the form of the electron interaction operator.

The RI decomposition itself can readily lead to reduced scaling of some quantum chemistry algorithms. If the contractions of the electron interaction involve mostly indices p, q and r, s but not cross combinations between them (e.g., contractions where one tensor has indices pq and rs while another has indices pr and qs), then a reduction of cost can be achieved, such as in the RI-MP2 approach.^{20,37,38} When these cross combinations occur, however, one needs to search for additional structure in the operator tensors. The latter can be achieved by the canonical polyadic decomposition.

B. Canonical polyadic decomposition

A polyad is a rank one tensor expressible, for example, by

$$X_{ijk\dots} = a_i b_j c_k \dots \quad (8)$$

or more abstractly as a series of Kronecker products

$$X = \dots \otimes c \otimes b \otimes a. \quad (9)$$

Note that we multiply factors in inverse order; this is simply to preserve a consistent column-major indexing of tensors.

A polyadic decomposition of a tensor is thus a decomposition as a sum of polyads⁹

$$T_{pqr\dots} = \sum_{\alpha} a_p^{\alpha} b_q^{\alpha} c_r^{\alpha} \dots \quad (10)$$

or, more abstractly,

$$T = \sum_{\alpha} \dots \otimes c_{\alpha} \otimes b_{\alpha} \otimes a_{\alpha}. \quad (11)$$

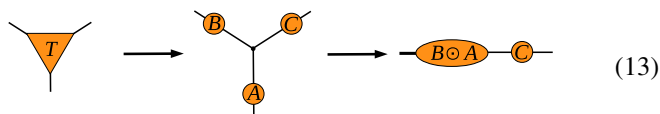
The canonical polyadic decomposition is the polyadic decomposition of lowest rank. It may be seen as one of the generalizations of SVD to third and higher order tensors, and for dimensions greater than 2, the CPD is unique under mild conditions.^{39,40}

As can be seen from the definition of Eq. (11), some matrix factorizations (for example, QR or LU factorizations) can be thought of as a CPD of matrices. In dimensions greater than 2, however, no closed form algorithm to extract the CPD is known, and one must rely on iterative optimization techniques to approximate the CPD.⁴¹ Substantial effort has been made by the mathematical community to develop approaches for doing so. Typical algorithms are the alternating least squares⁴² (ALS), gradient descent by means of, for example, the method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS), and non-linear least squares (NLS) methods.⁴¹ We refer the reader to the corresponding reviews^{27,43} for further details. We have used the Tensorlab⁴⁴ program by Lathauwer *et al.* for calculating the CPD in this work.

The polyadic decomposition can be expressed more conveniently through the Khatri-Rao product. If the vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} of Eq. (11) are stacked together as columns of matrices $A = \{\mathbf{a}\}$, $B = \{\mathbf{b}\}$, and $C = \{\mathbf{c}\}$, then the polyadic decomposition can be written as

$$T_{pqr} = ((B \odot A) \cdot C^T)_{pqr}, \quad (12)$$

which diagrammatically is

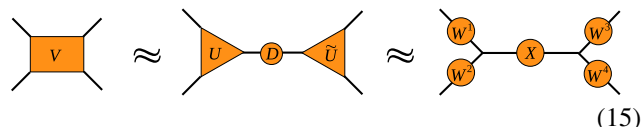


C. Tensor hypercontraction

The THC is a factorization of fourth-order tensors and can be seen as a combination of a singular value decomposition and a canonical polyadic decomposition. The THC approximation can be written as

$$\begin{aligned} V_{pqrs} &= \sum_{\alpha\beta} W_{p,\alpha}^1 W_{q,\alpha}^2 X_{\alpha,\beta} W_{r,\beta}^3 W_{s,\beta}^4 \\ &= ((W^2 \odot W^1) \cdot X \cdot (W^4 \odot W^3)^T)_{pqrs}. \end{aligned} \quad (14)$$

The THC can be viewed as a further approximation over RI, which is clear from the following diagram:

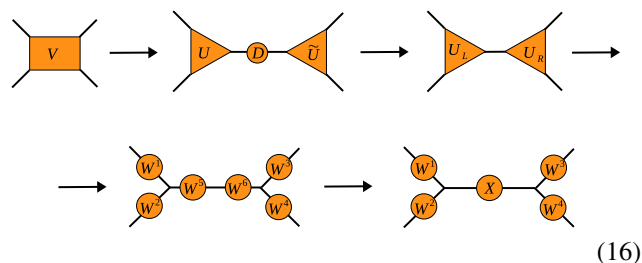


The sizes of the auxiliary indices α and β are the ranks of the decomposition. In all subsequent expressions, $r_{\text{THC}} = \dim(\{\alpha\}) = \dim(\{\beta\})$ for simplicity, although there is no fundamental restriction that $\dim(\{\alpha\}) = \dim(\{\beta\})$. Using the analogy with density fitting, several authors^{3,14} have speculated that the optimal rank of THC scales as $r_{\text{THC}} = O(N)$ in the case of the electron interaction tensor. We confirm this numerically in Sec. IV.

D. Algorithms for tensor hypercontraction

1. Composite method

The diagram in Eq. (15) suggests one possible way to calculate the THC of an order-4 tensor as a combination of the singular value and canonical polyadic decompositions. The following diagram depicts the procedure we call THC-CPD:



First, one can reshape the original tensor V with dimensions $I_1 \times I_2 \times I_3 \times I_4$ into a matrix of shape $I_1 I_2 \times I_3 I_4$ and apply a truncated SVD of rank r_{SVD} to it. We chose to multiply square roots of singular values into left and right singular vectors. Note that this produces matrices U_L and U_R of identical norm. Next, the left and right matrices of shapes $I_1 I_2 \times r_{\text{SVD}}$ and $I_3 I_4 \times r_{\text{SVD}}$ are reshaped into third-order tensors of shapes $I_1 \times I_2 \times r_{\text{SVD}}$ and $I_3 \times I_4 \times r_{\text{SVD}}$, respectively. The CPD of rank r_{CPD} is calculated for each of those tensors separately with any algorithm of choice, with each algorithm limited to n_{it} iterations. Finally, those factors of the CPD which do not have external indices can be merged into a single factor X .

Algorithm 1. Computing the THC using CPD.

```

1: function THC-CPD( $V, r_{\text{SVD}}, r_{\text{CPD}}$ )
2:    $I_1, I_2, I_3, I_4 \leftarrow \text{size}(V)$ 
3:    $V \leftarrow \text{reshape}(V, I_1 \cdot I_2, I_3 \cdot I_4)$ 
4:    $U, D, \tilde{U} \leftarrow \text{svd}(V, r_{\text{SVD}})$   $\triangleright O(N^4 r_{\text{SVD}})$ 
5:    $U_L \leftarrow U \cdot \text{sqrt}(D)$   $\triangleright O(N^2 r_{\text{SVD}}^2)$ 
6:    $U_R \leftarrow \tilde{U} \cdot \text{sqrt}(D^\dagger)$ 
7:    $W^1, W^2, W^5 \leftarrow \text{cpd}(U_L, r_{\text{CPD}})$   $\triangleright O(N^2 r_{\text{SVD}} r_{\text{CPD}} n_{it})$ 
8:    $W^3, W^4, W^6 \leftarrow \text{cpd}(U_R, r_{\text{CPD}})$ 
9:    $X \leftarrow W^5 \cdot W^{6T}$   $\triangleright O(r_{\text{CPD}}^2 \cdot r_{\text{SVD}})$ 
   return  $W^1, W^2, W^3, W^4, X$ 
10: end function

```

Algorithm 1 summarizes the composite method we employ, along with the computational scaling of its steps for a tensor of size $N \times N \times N \times N$, where we used $\text{cpd}()$ to denote a CPD method of choice.

A similar scheme was employed by Hohenstein *et al.* in their initial work on THC.² The only difference in our application is the choice of more efficient optimization methods to solve for CPD factors in steps 7 and 8 and the use of partial SVD in step 4.

The scaling of this algorithm is dominated by the truncated SVD in step 4. If the optimal rank of the SVD is of order $O(N)$, the algorithm is of $O(N^5)$ cost if $r_{\text{SVD}} = O(N)$ and $O(N^6)$ in the worst case. The SVD can be avoided if substitute singular vectors are available for the tensor V . In the case of the electron interaction, such substitutes are given by the 3-index integrals coming from the RI approximation. The auxiliary dimension r_{RI} is of $O(N)$.

A faster Algorithm 2 based on the RI approximation can be formulated as follows. We start with third-order tensors U, \tilde{U} of shapes $I_1 \times I_2 \times r_{\text{RI}}$ and $I_3 \times I_4 \times r_{\text{RI}}$, respectively, and an overlap matrix D of shape $r_{\text{RI}} \times r_{\text{RI}}$. A matrix root $D^{\frac{1}{2}}$ of D is calculated using the SVD or eigenvalue decomposition. This matrix is then multiplied into order-3 tensors U and \tilde{U} , which yields left and right third-order tensors U_L and U_R . If the size of the RI basis is large and U_L equals U_R , as in the case of 3-index integrals, an optional compression step can be applied (Algorithm 3): the auxiliary dimension of U_L and U_R is reduced by a truncated SVD of rank r_{SVD} .

Algorithm 2. Computing the THC using CPD and RI.

```

1: function THC-CPD-RI( $U, \tilde{U}, D, r_{\text{SVD}}, r_{\text{CPD}}$ )
2:    $Q\Lambda\tilde{Q} \leftarrow \text{svd}(D)$   $\triangleright O(r_{\text{RI}}^3)$ 
3:    $D^{\frac{1}{2}} \leftarrow Q \cdot \text{sqrt}(\Lambda)$ 
4:    $U_L \leftarrow U \cdot D^{\frac{1}{2}}$   $\triangleright O(N^2 r_{\text{RI}}^2)$ 
5:    $U_R \leftarrow \tilde{U} \cdot D^{\frac{1}{2}}$ 
6:   if  $U_R = U_L$  then
7:      $U_R \leftarrow \text{compress}(U_R, r_{\text{SVD}})$   $\triangleright$  Optional
8:      $U_L \leftarrow \text{compress}(U_L, r_{\text{SVD}})$   $\triangleright O(N^2 r_{\text{RI}} r_{\text{SVD}})$ 
9:   end if
10:   $W^1, W^2, W^5 \leftarrow \text{cpd}(U_L, r_{\text{CPD}})$   $\triangleright O(N^2 r_{\text{SVD}} r_{\text{CPD}} n_{it})$ 
11:   $W^3, W^4, W^6 \leftarrow \text{cpd}(U_R, r_{\text{CPD}})$ 
12:   $X \leftarrow W^5 \cdot W^{6T}$   $\triangleright O(r_{\text{CPD}}^2 \cdot r_{\text{SVD}})$ 
   return  $W^1, W^2, W^3, W^4, X$ 
13: end function

```

Algorithm 3. Compressing the RI basis.

```

1: function COMPRESS( $U, r_{\text{CPD}}$ )
2:    $I_1, I_2, I_3 \leftarrow \text{size}(U)$ 
3:    $U \leftarrow \text{reshape}(U, I_1 \cdot I_2, I_3)$ 
4:    $Q\Lambda\tilde{Q} \leftarrow \text{svd}(U, r_{\text{SVD}})$   $\triangleright O(I_1 I_2 I_3 r_{\text{SVD}})$ 
5:    $U \leftarrow Q\Lambda$   $\triangleright O(I_1 I_2 r_{\text{SVD}}^2)$ 
6:    $U \leftarrow \text{reshape}(U, I_1, I_2, r_{\text{SVD}})$ 
7:   return  $U$ 
8: end function

```

Finally, a CPD of rank r_{CPD} is calculated for the left and right parts, and the resulting factors with no external indices are merged into a single factor X . The resulting algorithm is listed below.

The overall scaling of Algorithm 2 may be dominated either by the $O(N^2 r_{\text{RI}}^2)$ cost of the SVD and matrix multiplications or by the $O(N^2 r_{\text{SVD}} r_{\text{CPD}} n_{it})$ cost of the iterative algorithm of the CPD. In practical calculations, we found that the latter contribution, despite scaling mildly with the system size N and optimal ranks r_{CPD} and r_{SVD} , is always dominant because of the large number of iterations required by the CPD algorithm. This motivated us to look for an equivalent algorithm to build the THC decomposition directly.

2. Direct method

We follow Sorber *et al.*⁴¹ to build a simple alternating least squares algorithm for the THC. We begin by introducing the approximation of a fourth-order tensor V by its THC decomposition \tilde{V} , which we recall is

$$\tilde{V}_{ijkl} = \sum_{\alpha\beta} W_{p,\alpha}^1 W_{q,\alpha}^2 X_{\alpha,\beta} W_{r,\beta}^3 W_{s,\beta}^4, \quad (17a)$$

$$= [(W^2 \odot W^1) \cdot X \cdot (W^4 \odot W^3)^T]_{ijkl}. \quad (17b)$$

Then we can define an error tensor

$$\Delta_V = V - \tilde{V}, \quad (18)$$

whose Frobenius norm is just

$$f = \|\Delta_V\|^2 = \sum_{pqrs} (V_{pqrs}^* - \tilde{V}_{pqrs}^*) (V_{pqrs} - \tilde{V}_{pqrs}). \quad (19)$$

Diagrammatically, this is

$$f = \text{Diagram 1} - \text{Diagram 2} - \text{Diagram 3} + \text{Diagram 4} \quad (20)$$

Clearly, the best possible THC approximation to V will correspond to a minimum of the cost function f . We note that f is a real-valued analytic function, and hence $\frac{\partial f}{\partial W} = (\frac{\partial f}{\partial W^*})^*$, where $W \in \{W^1, W^2, W^3, W^4, X\}$.

In order to minimize the cost function, we proceed with the calculation of its gradient, which can be easily done using diagram 20. The partial derivative of f with respect to W^1 is

$$\frac{\partial f}{\partial W^1} = - \text{diagram} + \text{diagram} \quad (21)$$

and the full gradient of f can be found in the [supplementary material](#). Noting that $\frac{\partial f}{\partial W}$ is linear in W^* , we contract all factors around W^* into an environment matrix A , as seen in diagram 22, and set the derivative to zero,

$$\text{diagram} = \text{diagram} \quad (22)$$

We end up with a problem

$$A \cdot W^* = B. \quad (23)$$

The solution to Eq. (23) can be obtained by taking the inverse of A (or a pseudoinverse, if A is a rank-deficient matrix). The final expression for W^{1*} is given diagrammatically as

$$W^{1*} = \text{diagram} \quad (24)$$

As made clear by the diagram, if both ranks of the THC decomposition are r_{THC} , then the construction of the environment matrix A scales as $O(r_{\text{THC}}^3)$, as does computing its generalized inverse. If each of the dimensions of V equals N , then the cost of calculating W^{1*} scales as $O(N^4 r_{\text{THC}})$. Updates for the rest of the terms in the THC decomposition can be calculated similarly.

A simple iterative optimization algorithm can be built as follows. First, the THC factors W are initialized randomly. For each factor, an update is calculated as shown on diagram 24, keeping the other factors fixed. The process is iterated until convergence of the factors. The resulting THC-ALS algorithm is listed below.

The calculation of the right-hand side of Eq. (23) dominates in the cost of THC-ALS, scaling as $O(N^4 r_{\text{THC}})$. A simple modification is possible to reduce this cost by one order of magnitude. If an approximation to the singular vectors of the original tensor V is available from the beginning, as in the case of the electron interaction, it can be used in place of V , leading to a faster algorithm. The diagram corresponding to Eq. (23) then becomes

$$W^{1*} = \text{diagram} \quad (25)$$

The cost of the expression above scales as $O(N^2 r_{\text{RI}} r_{\text{THC}})$ because the contraction of a fourth-order tensor V with matrices W is replaced by contractions of two third-order tensors U

Algorithm 4. Alternating least squares.

```

1: function THC-ALS( $V, r_{\text{THC}}, \epsilon$ )
2:    $I_1, I_2, I_3, I_4 \leftarrow \text{size}(V)$ 
3:    $W^1, W^2, W^3, W^4, X$ 
4:    $\text{init\_random}(I_1, I_2, I_3, I_4, r_{\text{THC}})$ 
5:   repeat
6:     for all  $W \in \{W^1, W^2, W^3, W^4, X\}$  do
7:        $A_W \leftarrow \text{get\_environment}(W^1, W^2, W^3, W^4, X)$ 
8:        $B_W \leftarrow \text{get\_rhs}(V, W^1, W^2, W^3, W^4, X)$ 
9:        $\triangleright O(r_{\text{THC}}^3)$ 
10:       $W_{\text{new}} \leftarrow A^{-1} B$ 
11:       $\triangleright O(N^4 r_{\text{THC}})$  or  $O(N^2 r_{\text{SVD}} r_{\text{THC}})$  with RI
12:     $\Delta \leftarrow \max_W \frac{\|W_{\text{new}} - W\|}{\|W\|}$ 
13:     $W \leftarrow W_{\text{new}}$ 
14:  until  $\Delta > \epsilon$  return  $W^1, W^2, W^3, W^4, X$ 
15: end function

```

and \tilde{U} . We only need to modify the function *get_rhs()* to build a lower scaling algorithm, which we refer to as THC-ALS-RI.

Alternating least squares algorithms are simple and often robust⁴⁵ but may take a large number of iterations to converge.⁴² Following an analogy with CPD,⁴¹ we also implemented a quasi-Newton method using limited memory BFGS (L-BFGS) with a dogleg trust region⁴⁶ for THC; this method we refer to as THC-BFGS.

The THC-ALS and THC-BFGS, and their RI variants, are novel direct methods to calculate the THC decomposition based on minimization of the Frobenius norm of the error. Composite methods such as THC-CPD(ALS) and their RI variants have been used previously in earlier work on THC.²

We refer the reader to the [supplementary material](#) for optimized expressions of the THC gradient and objective function. Due to their complexity many of the equations we present (especially the ones related to coupled cluster, see Sec. IV) were generated by a computer algebra system developed in our group,^{47,48} although this can be done by manipulating diagrams as well.

3. Numerical experiments

Here we wish to compare the performance of the composite methods [THC-CPD(ALS), THC-CPD(BFGS), and THC-CPD(NLS)] and direct algorithms (THC-ALS and THC-BFGS) for THC decomposition. Table I shows the scaling per iteration for the various algorithms we consider (see algorithms in the text and also Ref. 41 for further details on the scaling of CPD, which we used in the composite methods). The scaling is given for a full fourth-order tensor with sizes equal N in the first part of the table and for RI-decomposed tensors with rank r_{SVD} in the second part. Recall that the composite methods in the first part of the table require an initial SVD, the cost of which scales as $O(N^4 r_{\text{SVD}})$; this cost is in addition to that of the iterative steps required to converge the CPD.

To summarize the contents of Table I, let us assume that both r_{SVD} and r_{THC} are $O(N)$, as is the case for the electron interaction tensor.³ Then all composite algorithms have a non-iterative $O(N^5)$ step followed by iterative $O(N^4)$ steps, while direct algorithms have $O(N^5)$ cost per iteration. If an RI

TABLE I. Computational scaling per iteration of various algorithms to converge the CPD in composite methods or the THC itself in direct methods. The top half of the table shows scaling for methods which do not use an initial RI, while the bottom half of the table shows scaling for methods which do use an initial RI.

Algorithm	Scaling
THC-CPD(ALS)	$O(N^3 r_{\text{THC}})$
THC-CPD(BFGS)	$O(N^3 r_{\text{THC}})$
THC-CPD(NLS)	$O(N^3 r_{\text{THC}} + r_{\text{THC}}^3 + N^2 r_{\text{THC}}^2)$
THC-ALS	$O(N^4 r_{\text{THC}} + r_{\text{THC}}^3)$
THC-BFGS	$O(N^4 r_{\text{THC}} + r_{\text{THC}}^3)$
THC-CPD-RI(ALS)	$O(N^3 r_{\text{THC}})$
THC-CPD-RI(BFGS)	$O(N^3 r_{\text{THC}})$
THC-CPD-RI(NLS)	$O(N^3 r_{\text{THC}} + r_{\text{THC}}^3 + N^2 r_{\text{THC}}^2)$
THC-ALS-RI	$O(N^2 r_{\text{SVD}} r_{\text{THC}} + r_{\text{THC}}^3)$
THC-BFGS-RI	$O(N^2 r_{\text{SVD}} r_{\text{THC}} + r_{\text{THC}}^3)$

approximation is used, all algorithms have $O(N^4)$ scaling per iteration.

To get a feeling for how these various algorithms perform in practice, we compared the convergence speed of direct and composite methods using the performance metrics proposed by Dolan and Moré.⁴⁹ We generated fifty sets of random THC factors using a uniform distribution, from which we built fifty tensors which had size $4 \times 4 \times 4 \times 4$ and THC ranks 2 and 3. We further generated fifty sets of random initial guesses drawn from the same uniform distribution. This yielded a set P of 2500 (tensor, initial guess) pairs for each tensor rank.

The algorithms in the first part of Table I (i.e., those algorithms that do not use RI) form a set of algorithms S . For each problem p in P , we applied each algorithm s in S . We allowed the algorithms to run for up to 2000 iterations or until converged, where our convergence criterion was $\|V - \tilde{V}\| \leq 10^{-5}$. The number of iterations required for an algorithm s to converge a problem p we denote as $t_{p,s}$. If an algorithm did not converge a given problem, we set $t_{p,s} \rightarrow \infty$.

For direct methods, we stopped the iterative algorithm if $\|V - \tilde{V}\| \leq 10^{-7} \|V\|$ and declared the method to have failed if it did not meet our convergence threshold. For composite methods, we retained singular values larger than 10^{-7} in building the factors U_L and U_R . We declared the CPD converged if $\|U - \tilde{U}\| \leq 10^{-10}$ and stopped the iterations if $\|U - \tilde{U}\| \leq 10^{-14} \|U\|$. In all cases the threshold for the pseudoinverse was set to 10^{-14} . We emphasize that for both direct and composite methods, the definition of success was accurate decomposition of V , e.g., the magnitude of absolute error had to be less than the threshold $\|V - \tilde{V}\| \leq 10^{-5}$.

Having applied each algorithm s to each problem p , we use as a performance metric

$$\rho_s(\tau) = \frac{|\{p \in P : t_{p,s} \leq 2^\tau \cdot \min_{s \in S} \{t_{p,s}\}\}|}{|P|}. \quad (26)$$

In other words, $\rho_s(\tau)$ is the fraction of problems that algorithm s solved within 2^τ times the best algorithm for each problem. We would like $\rho_s(\tau)$ to approach one for large enough τ , indicating that the algorithm converged all problems that could be converged, and we would like $\rho_s(\tau)$ to grow toward one as rapidly as possible, indicating that the algorithm converged

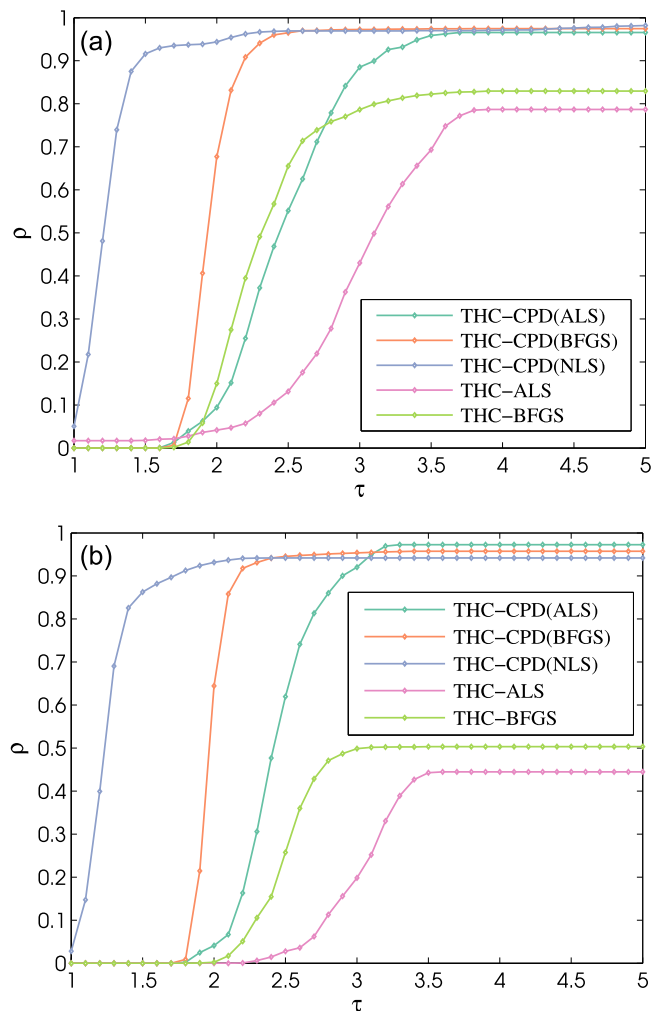


FIG. 1. Performance metric $\rho_s(\tau)$ for various THC decomposition algorithms. (Top) $r_{\text{THC}} = 2$. (Bottom) $r_{\text{THC}} = 3$.

relatively quickly. Results are shown in Fig. 1 where the top panel shows results for rank two tensors and the bottom panel shows results for rank three tensors.

As one can see, composite methods THC-CPD outperform our direct THC decomposition. The difference in performance is more prominent for $r_{\text{THC}} = 3$ than it is for $r_{\text{THC}} = 2$. For example, THC-ALS converges for less than 50% of the possible problems when $r_{\text{THC}} = 3$, compared with about 80% for $r_{\text{THC}} = 2$. We believe the poor performance of the direct algorithm is because the THC factors are not unique (as our numerical experimentation indicated), whereas the factors in the CPD are unique under mild conditions.⁴⁰ This non-uniqueness results in gradient vectors which are close to zero in certain directions, and optimization algorithms then require many more iterations to minimize the objective function.

Overall, the best method for THC seems to be the composite THC-CPD(NLS), which we recall uses a nonlinear least squares solver for CPD.^{41,46} We will thus use THC-CPD(NLS) for subsequent THC decompositions in this work.

We should note that no method was able to solve all problems in our setup, though the composite methods succeeded in the very large majority of cases. Similar behavior for random test factors was previously observed for CPD.⁴¹ This did not pose a problem in our practical applications. We should also

note that our results here should be considered with some caution, simply because metrics generated with random factors may not be representative for the tensors encountered in quantum chemistry, which generally have more structure. However, our results most likely show the worst case behavior for the proposed algorithms.

IV. TENSOR-STRUCTURED COUPLED CLUSTER

While the direct algorithms proposed in Sec. III D 2 were not particularly good for the decomposition of random tensors, we introduced them because they find new life in our tensor-decomposed coupled cluster methods, as we discuss below. Let us begin, however, with a quick overview of the restricted CCSD (RCCSD) method. We define a cluster operator

$$\hat{T} = {}^1\hat{T} + {}^2\hat{T}, \quad (27)$$

where the individual operators ${}^i\hat{T}$ are excitation operators

$${}^1\hat{T} = \sum_{ai} {}^1T_i^a \hat{E}_i^a, \quad (28a)$$

$${}^2\hat{T} = \sum_{abij} {}^2T_{ij}^{ab} \hat{E}_i^a \hat{E}_j^b. \quad (28b)$$

Here

$$\hat{E}_i^a = \hat{a}_{a,\uparrow}^\dagger \hat{a}_{i,\uparrow} + \hat{a}_{a,\downarrow}^\dagger \hat{a}_{i,\downarrow} \quad (29)$$

are spin adapted excitations or unitary group generators,¹⁶ and iT are order $2i$ amplitude tensors. With these cluster operators, we construct a similarity-transformed Hamiltonian \bar{H} as

$$\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}, \quad (30)$$

from which the energy can be extracted as

$$E_{\text{CCSD}} = \langle 0 | \bar{H} | 0 \rangle, \quad (31)$$

where $|0\rangle$ is a closed shell single determinant (usually a Hartree-Fock state). The excitation amplitudes are usually obtained by projecting the similarity-transformed Hamiltonian on the left against a set of excited determinants to form residuals which are set to zero,

$${}^1R_i^a = \langle 0 | \hat{a}_{i,\uparrow}^\dagger \hat{a}_{a,\uparrow} \bar{H} | 0 \rangle = 0, \quad (32a)$$

$${}^2R_{ij}^{ab} = \langle 0 | \hat{a}_{i,\uparrow}^\dagger \hat{a}_{j,\downarrow}^\dagger \hat{a}_{b,\downarrow} \hat{a}_{a,\uparrow} \bar{H} | 0 \rangle = 0. \quad (32b)$$

These result in polynomial equations of the amplitude tensors which can be transformed to the form

$${}^1T_i^a = {}^1D_i^a {}^1G_i^a, \quad (33a)$$

$${}^2T_{ij}^{ab} = {}^2D_{ij}^{ab} {}^2G_{ij}^{ab}, \quad (33b)$$

which can be solved by iterations until a fixed point is found. Here, 1D and 2D are orbital energy denominator tensors built from diagonal elements of the Fock matrix F

$${}^1D_i^a = \frac{1}{F_a^a - F_i^i}, \quad (34a)$$

$${}^2D_{ij}^{ab} = \frac{1}{F_a^a + F_b^b - F_i^i - F_j^j}. \quad (34b)$$

The tensors 1G and 2G are built from contractions of the amplitude tensors with the Hamiltonian.

A. Least squares coupled cluster theories

The logic used to derive the ALS algorithm for THC decomposition can be readily applied in the coupled cluster context. Here, we will use coupled cluster doubles (for which one neglects ${}^1\hat{T}$) as an example, with expressions for CCSD shown in the [supplementary material](#).

We begin by imposing the THC structure on the doubles amplitudes. We approximate the amplitude tensor 2T with its THC decomposition ${}^2\tilde{T}$. The difference between original and approximated amplitudes is

$$\begin{aligned} \Delta_T &= {}^2T - {}^2\tilde{T} \\ &= {}^2T - (Y^2 \odot Y^1) \cdot Z \cdot (Y^4 \odot Y^3)^T, \end{aligned} \quad (35)$$

where Y^i and Z are factors in the THC decomposition of 2T . We wish to minimize the squared norm of the error tensor Δ_T , which is the minimization of the corresponding cost function f_T ,

$$f_T = |\Delta_T|^2 = ({}^2T^* - {}^2\tilde{T}^*)({}^2T - {}^2\tilde{T}). \quad (36)$$

Setting partial derivatives of f_T with respect to the decomposition factors to zero, we obtain a new set of equations

$$\frac{\partial f_T}{\partial Y} = -{}^2T^* \frac{\partial {}^2\tilde{T}}{\partial Y} + {}^2\tilde{T}^* \frac{\partial {}^2T}{\partial Y} = 0, \quad (37)$$

where $Y \in \{Y^1, Y^2, Y^3, Y^4, Z\}$. Again, as f_T is real and analytic, only one set of derivatives (either with respect to Y or Y^*) is sufficient to find its minimum.

Now we use Eq. (33b) to replace ${}^2T^*$ with ${}^2D^2G$. The idea is to thus minimize the difference between a decomposed tensor ${}^2\tilde{T}$ and a solution of the CCD amplitude equations. The resulting amplitude equations are

$$\tilde{T}^* \frac{\partial \tilde{T}}{\partial Y} = {}^2G^2D \frac{\partial \tilde{T}}{\partial Y}. \quad (38)$$

This is the analog of Eq. (23) in THC-ALS and can be solved in the least-squares sense (i.e., with the help of the pseudoinverse) as the left-hand side is linear in Y^* . Diagrammatically, we have

$$\text{Diagrammatic representation of Eq. (38)} \quad (39)$$

Note that we have written the energy denominator 2D as an order-8 diagonal tensor ${}^2D_{ijj'j'}^{abab'}$, instead of an order-4 dense tensor ${}^2D_{ij}^{ab}$ as in Eq. (33b), which we do only so that we can separate 2D from 2G in preparation for the decomposition of 2D . Indeed, these equations can be further factorized if one employs a CPD of 2D to disentangle particle and hole indices. A low-rank decomposition of denominator tensors can be built using an exponential parametrization⁵⁰ (also known as Laplace transformation)⁵¹ as, for example,

$${}^2D_{ij}^{ab} = \sum_{\omega} C_{\omega} e^{A_{\omega} F_i^a} e^{A_{\omega} F_j^b} e^{-A_{\omega} F_a^a} e^{-A_{\omega} F_b^b} \quad (40a)$$

$$= \sum_{\omega} {}^2D_{i,\omega}^1 {}^2D_{j,\omega}^2 {}^2D_{a,\omega}^3 {}^2D_{b,\omega}^4. \quad (40b)$$

We have used the parameters from Ref. 50, which provide absolute accuracy of better than 10^{-12} with ranks of order ≈ 15 , which do not depend on the system size N .

The final form of our ALS-type coupled cluster doubles equations is thus

$$(41)$$

The explicit form of these equations and analogous expressions for ALS-type CCSD are shown in the [supplementary material](#).

We chose to apply THC to 2T amplitudes ordered in the same way as Mulliken ordered two-electron integrals, i.e.,

$${}^2T_{ij}^{ab} = \sum_{\alpha\beta} Y_{a,\alpha}^1 Y_{i,\alpha}^2 Z_{\alpha\beta} Y_{b,\beta}^3 Y_{j,\beta}^4. \quad (42)$$

In contrast with the decomposition of the interaction tensor, we found that this parametrization of the amplitudes does not significantly influence the rank needed for a good approximation. It does, however, lead to simpler intermediate quantities.

After defining proper intermediates, which we did using our automatic algebraic system,⁴⁸ the cost of these equations has quartic scaling in r_{THC} and N per iteration. We provide those fully factorized equations in the [supplementary material](#) along with the source code for the contractions. Most of the numerical experiments in Sec. IV B were done with a simpler code which had $O(N^5)$ scaling because it made less sophisticated use of intermediate quantities; however, the $O(N^4)$

and $O(N^5)$ implementations differ only in the order in which contractions were carried out.

Equation (41) and its analogs for all other factors in the decomposition of 2T constitute what we call THC-RCCSD and are the main result of this paper. It must be stressed that the proposed scheme is generic and can be applied to any factorization of amplitudes and the Hamiltonian. We use THC here and leave the exploration of other possibilities for subsequent work.

B. Test calculations

To assess the performance of our tensor-structured CCSD, we present calculations on a variety of small-to medium-sized molecules. All calculations used the cc-pVDZ basis from the EMSL database,⁵² and the corresponding cc-pVDZ-RI was used in the RI approximation.

For smaller systems, the THC-CPD(NLS) algorithm was used to obtain the THC approximation to the full two-electron integrals in the AO basis. We set the relative convergence threshold for CPD iterations to 10^{-14} , as we did in our numerical experiments in Sec. III. Singular values larger than 10^{-12} were retained in obtaining U_L and U_R . The maximum number of iterations allowed during the decomposition of the integrals was 1000. The subsequent coupled cluster calculations were stopped either after the energy was converged to within 10^{-9} Hartree or a limit of 250 iterations was reached. Thresholds for pseudoinverse calculations were set to 10^{-14} .

For larger systems, listed in Table II, THC-CPD(NLS) was applied to RI-decomposed two-electron integrals. Other parameters were as described above, except we decreased

TABLE II. CCSD correlation energies (E_c), errors in correlation energies (ΔE_c), number of THC-RCCSD iterations (n_{iter}), and the norm of doubles residuals ($|{}^2R_{ij}^{ab}|$) for several small molecules.

System	E_c (mH)	ΔE_c (mH)		n_{iter}		$ {}^2R_{ij}^{ab} $	
		N_{RI}	$1.5N_{\text{RI}}$	N_{RI}	$1.5N_{\text{RI}}$	N_{RI}	$1.5N_{\text{RI}}$
Acetic acid	-666.510	-0.579	-0.453	50	31	0.041	0.033
Aniline	-997.193	-1.177	-0.471	111	64	0.051	0.032
Diboron tetrafluoride	-909.944	-0.702	-0.716	15	17	0.053	0.034
Benzene	-823.101	-0.985	-0.450	111	62	0.048	0.030
Butadiene	-581.340	-0.710	-0.274	41	42	0.041	0.025
Cyclobutane	-621.099	-0.895	-0.290	77	50	0.039	0.028
Dimethylsulfoxide	-661.870	0.195	-0.624	287	39	0.056	0.025
Furan	-736.463	-0.865	-0.454	73	50	0.046	0.033
Isobutane	-652.505	-0.876	-0.263	78	49	0.035	0.025
Methylformate	-666.805	-0.586	-0.455	62	35	0.042	0.032
Methylnitrite	-708.990	-0.476	-0.492	68	47	0.047	0.033
Phenol	-1005.727	-0.887	-0.514	120	59	0.051	0.032
Pyridine	-842.453	-1.045	-0.475	104	61	0.047	0.032
Pyrrole	-727.051	-0.855	-0.407	84	52	0.045	0.032
Thiophene	-695.593	-1.013	-0.657	97	52	0.039	0.032
Toluene	-980.030	-1.270	-0.461				
MUE ^a		0.820	0.466				
Max ^b		1.270	0.716				
RMS ^c		0.861	0.482				

^aMean unsigned error.

^bMaximum unsigned error.

^cRoot-mean-square error.

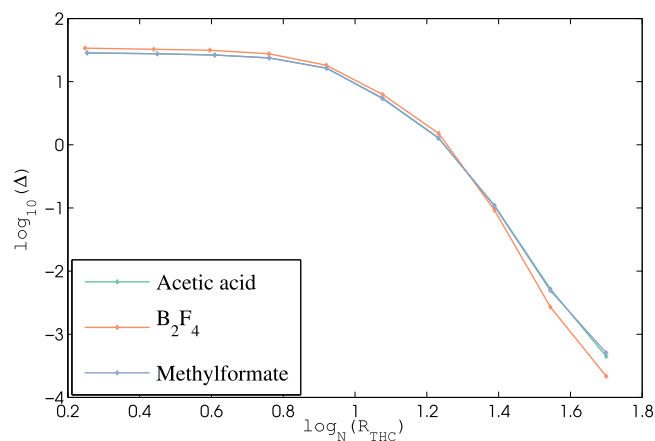


FIG. 2. Frobenius norm of error in decomposed two electron integrals.

the number of iterations allowed during decomposition of the integrals to 500.

1. Decomposition of two-electron integrals

The accuracy of the THC decomposition of the two-electron integrals governs the accuracy of the energy in subsequent calculations. Thus, we first wish to check the dependence on the error in the decomposition of two-electron integrals on THC rank. Figure 2 plots this error in a double logarithmic scale for three small molecules. We note that the decomposition is computationally useful if the rank r_{RHC} is close to the number of basis functions N . As the figure shows, the error in the two-electron integrals decays exponentially with respect to the THC rank. We found that this trend holds for every system tested and depends only slightly on whether the two-electron integrals are decomposed in the atomic orbital or molecular orbital basis.

To see how the decomposition affects subsequent energies, we checked the error in the second-order Møller-Plesset (MP2) correlation energy, as shown in Fig. 3. The combination of MP2 and THC was first proposed by Hohenstein *et al.*³ and scales as $O(N^4)$. These authors used a version of THC with the restriction that all factors W were the same, which we did not impose in our work. The error in the MP2 correlation energy follows the trend seen in the decomposition of the two-electron integrals. Results within 0.1 mH of the exact

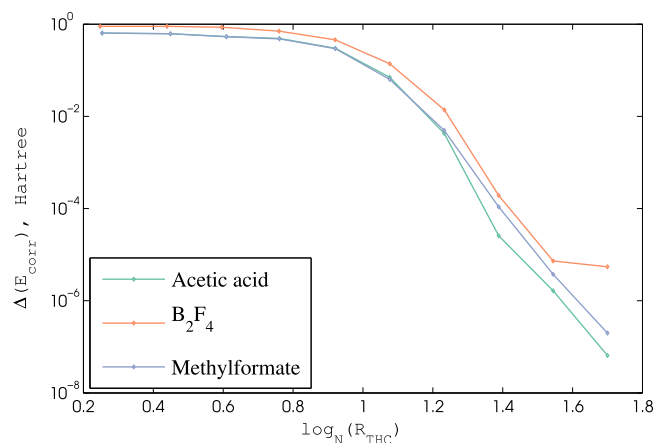


FIG. 3. Absolute error in the MP2 correlation energy.

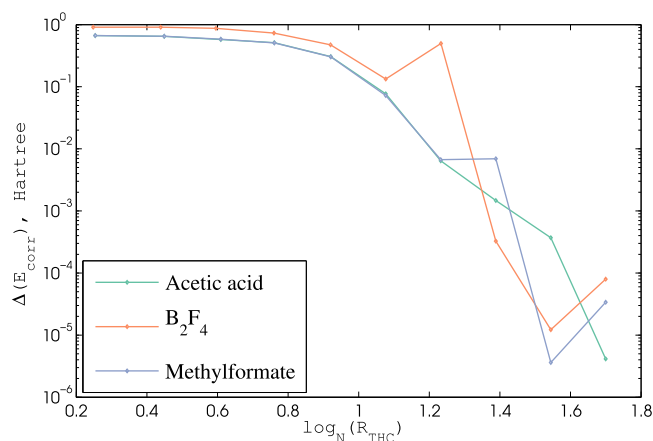


FIG. 4. Absolute error in the RCCSD correlation energy.

MP2 correlation energy are already achieved with $r_{\text{THC}} \sim N^{1.2} - N^{1.4}$. We expect that the THC would work better for larger and more extended systems as the two-electron integrals become sparser, and a lower rank decomposition would correspondingly become more accurate. Note that this is true even in strongly correlated systems for which the atomic orbital basis two-electron integrals remain sparse.

2. Restricted coupled cluster with singles and doubles

Finally, we demonstrate the behavior of the THC-decomposed RCCSD method (THC-RCCSD), seen in Fig. 4. We chose the rank of the THC decomposition of the amplitudes and two-electron integrals to be the same. The error in the RCCSD correlation energy has a non-monotonic dependence on the THC rank but follows the same basic trends as seen in Figs. 2 and 3. As with MP2, errors on the order of 0.1 mH are achieved with $r_{\text{THC}} \sim N^{1.2} - N^{1.4}$.

It is interesting to see what part of the error in energy can be attributed to the approximation of the Hamiltonian, especially because building the decomposition of the Hamiltonian contributed $\sim 95\%$ of the total computational cost. For this reason, we calculated the correlation energy with converged THC-RCCSD amplitudes but exact two-electron integrals. As Fig. 5 shows, using the exact two-electron integrals decreases the error in energy, as one would expect, but does not remove its

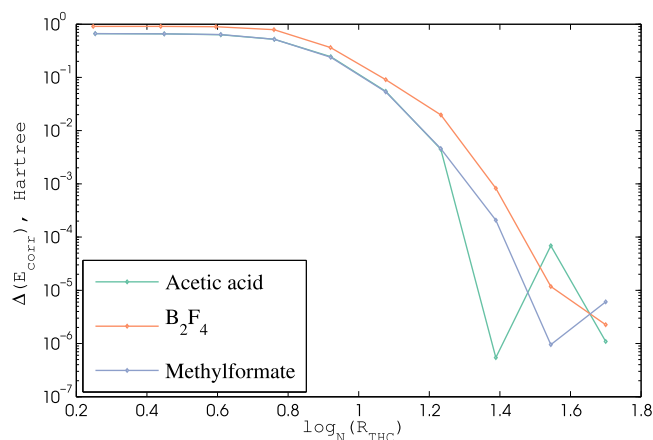


FIG. 5. Absolute error in the RCCSD correlation energy with exact two-electron integrals.

non-monotonic dependence on the THC rank. We attribute this behavior to the nonlinear nature of the coupled cluster equations, which can be quite sensitive to changes in the parameters of the Hamiltonian.

3. Convergence of THC-RCCSD algorithm

Let us take a moment to discuss the convergence of the THC-RCCSD. In this work, we used simple fixed point iteration without any convergence acceleration to solve THC-RCCSD equations and stopped iterating when a specified difference in energy between subsequent steps was reached. The convergence of the resulting algorithm is highly dependent on the rank chosen for the THC approximation. Figure 6 shows the number of iterations required to converge the energy to 1 microhartree.

For smaller and larger THC ranks, the convergence of THC-RCCSD is comparable or better than the regular RCCSD if both techniques use simple fixed-point iteration. In the intermediate regime, however, the algorithm may need a large number of iterations to converge. Convergence in this regime depends strongly on the choice of initial parameters, and several random trials are needed to find good ones.

We rationalize this behavior by noting that THC-RCCSD in our formulation is a combination of two iterative algorithms: ALS and standard coupled cluster iterations. There is some competition between these two approaches in determining the updates of the parameters. When the THC rank is small, the updates should be mostly determined by ALS, as the 2T amplitudes are poorly approximated with any set of parameters Y . In contrast, with larger ranks, the CC equations govern the update because 2T is well approximated at each step. In the intermediate regime, the update is effective neither for CC nor for ALS.

This odd convergence behavior of our hybrid algorithms requires further study so that we can definitely avoid it. For the remainder of this work, we simply used relatively large THC ranks, where THC-RCCSD performs similarly to regular RCCSD.

In this work, we tested convergence by tracking the coupled cluster energy. We checked that when the energy became stationary, the coupled cluster residuals are small. The T amplitudes also are converged, but the individual factor

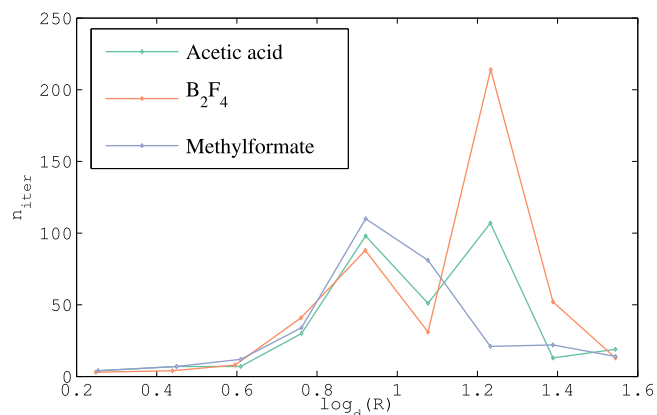


FIG. 6. Number of iterations of THC-RCCSD to reach $10^{-6} H$ difference in energy.

matrices Y^n and Z of Eq. (42) do not necessarily converge uniquely. Presumably this is inherited from the flexibility of the CPD. Note that, for example, if we scale one factor by λ and another factor by $1/\lambda$, the reconstituted tensor is unaffected. This same freedom occurs in, for example, the eigen decomposition, where the normalization of the eigenvectors is arbitrary. Additional constraints may be imposed to remove this flexibility, and we will explore doing so in future work.

4. Larger systems

Having seen how the THC-RCCSD method performs for various THC decomposition ranks, we tested the method on a set of small- and medium-sized molecules introduced in previous work on THC.⁴ Technical details of the calculations, including molecular geometries and reference energies, are provided in the [supplementary material](#). We chose the ranks of the THC decomposition of the amplitudes and integrals to be similar to the number of functions N_{RI} in the basis used in the RI approximation. Energies, differences with respect to regular coupled cluster, the number of CC iterations, and norm of final residuals are presented in Table II. We used RI for all these calculations.

Let us highlight some results in Table II. As THC-RCCSD is an approximation to the regular CC equations, the full residuals may not be zero because the number of CC equations is usually much larger than the number of parameters of THC-RCCSD. Interestingly, while the norm of the doubles residuals is not negligible in THC-RCCSD, the error in the energy is quite low. This contrasts with the conventional CC, where the difference of energy with the final value during iterations is proportional to the norm of the residuals. Also, we observed a generically faster convergence of energy in THC-RCCSD for larger ranks.

We note that our results are on par with calculations of Hohenstein *et al.*,⁴ but similar errors are achieved with ranks which are roughly half as large. Presumably this is because in previous work, most of the factors in the THC decomposition of the amplitudes were kept fixed (except Z), whereas our scheme optimizes all factors, therefore providing greater flexibility and reaching the exact decomposition quicker. It should be noted that most of the time in our calculations was spent on the decomposition of the Hamiltonian rather than on the solution of approximated CC equations. Quadrature-based methods for building THC, such as those described in Refs. 2 and 53, may be much faster than iterative approaches and reach the same accuracy as ALS-based THC with about $3\times$ more factors.⁵³ Hence we recommend a hybrid scheme for future development of THC-based methods: one may build the THC of the Hamiltonian with real space quadratures, as shown in Ref. 3, and later solve for decomposed cluster amplitudes as described here. Again, we emphasize that our scheme is not limited to THC and can be applied to many other decompositions, which is the topic of ongoing investigation.

5. Comparison with other methods

As the use of tensor decompositions in quantum chemistry has become more popular recently, here we aim to compare

our approach with existing work. Our technique extends the approach of Hohenstein *et al.*,²⁻⁴ who first introduced the THC. Our Algorithm 1, used for the two-electron integrals, is similar to the one introduced in Ref. 2 up to implementation details. These authors abandoned the optimization-based THC decomposition in favor of real space quadratures, first introduced in Ref. 3, and presented a quadrature-based $O(N^4)$ CCSD method in Ref. 4. The method of Hohenstein *et al.* used fixed factors W in the decomposition of the integrals [see Eq. (14) for the definition of W], and likewise amplitudes. Our contribution extends their approach by allowing all factors in the THC decomposition of amplitudes to be optimized. Additionally, our scheme is not tied to a particular form of the decomposition, which should stimulate the development of other similar methods.

Related is recent work of Hummel *et al.*¹⁴ Those authors developed a symmetric ALS algorithm to decompose the electron interaction tensor in their periodic CCSD theory. Subsequently, they demonstrated that by using THC decomposed interaction for a single term, it is possible to reduce the scaling of distinguishable CC doubles or linearized CC theories to $O(N^5)$ with very low approximation errors. The difference with our approach is the use of the factorization of amplitudes along with the factorization of integrals.

Lastly, we wish to discuss closely related studies of Benedikt *et al.*^{12,13} These authors used CP decomposition for both the two-electron interaction and the cluster amplitudes. Although the idea of their CCD method is quite similar to ours (apart from the choice of the decompositions), the actual method of solving for the 2T amplitudes is very different. These authors build 2T amplitudes in the CP format in the traditional way and then reduce the rank of the resulting tensor by an iterative algorithm at each CC iteration. This, presumably, is computationally expensive, as their CP-decomposed CCD method was applied only to small test systems. Our scheme can, in principle, be applied to the CP format as well but does not have the overhead of a recompression step.

V. CONCLUSIONS

Systematically dependable quantum chemical methods rely on solving the Schrödinger equation, but unfortunately do so at a significant and often impractical computational cost. For many-body methods such as coupled cluster theory, the cost can be explained simply: the various objects of the theory are high-order tensors which must be contracted with one another, and the contraction of two high-order tensors is computationally costly. Tensor decompositions lower the cost by writing high-order tensors as sums of products of low-order objects and are one of the most promising ways to apply many-body theories to large systems.

In this work, we have shown how the combination of tensor hypercontraction and canonical polyadic decomposition allows us to solve the closed-shell CCSD equations with $O(N^4)$ scaling by solving directly for the factors which decompose the cluster operator [Eq. (41)]. By increasing the dimensions of these factors (i.e., by increasing the rank), we can approach the exact CCSD result in a more or less systematic fashion and can achieve results within 0.1 *mH* of the exact CCSD answer

with ranks on the order of the size of the basis. Our alternating least squares method improves over previous studies of THC in coupled cluster theories^{4,10} where fixed real-space quadratures were used to build the decomposition of cluster amplitudes and provide more accurate results for smaller ranks. The proposed scheme, however, is general and can be applied to any decomposition, as well as readily extended to more sophisticated coupled cluster theories. Among other possibilities, we plan extensions to the unrestricted CC and our own symmetry-projected CC theories.^{22,23,54} Lastly, we should mention that coupled cluster methods with decomposed amplitudes are much more suitable for parallelization than are the traditional ones because the communication becomes much cheaper. While our work along the aforementioned lines is still in the early stages, we hope that these low-scaling coupled cluster methods will help make large-scale CCSD calculations essentially routine.

SUPPLEMENTARY MATERIAL

See [supplementary material](#) for the THC gradient expressions, complete specification of test systems, and least squares coupled cluster expressions.

ACKNOWLEDGMENTS

This work was supported as part of the Center for the Computational Design of Functional Layered Materials, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under Award No. DE-SC0012575. G.E.S. is a Welch Foundation chair (C-0036).

APPENDIX: WIRING DIAGRAMS

We have made extensive use of wiring diagrams to simplify the representation and manipulation of complex tensor expressions. This graphical notation is similar to the usual diagrammatic notation used in many-body theory but not identical. For completeness, we here describe the basic semantics of our diagrams.

In our notation, tensors are represented by shapes. Typically a d -order tensor is represented by a polygon with d corners (and a second-order tensor by a circle), though we have not followed this convention universally. Indices are denoted by lines; a line connecting multiple tensors is to be summed over and open lines correspond to free indices. If a particular element of a tensor expression is required, we label the open lines.

To be concrete, a matrix product would be represented by

$$\begin{array}{c} \text{---} \text{A} \text{---} \text{B} \text{---} \\ | \quad | \\ p \quad q \end{array} \longrightarrow \begin{array}{c} \text{---} \text{C} \text{---} \\ | \\ p \quad q \end{array} \quad \sum_r A_{pr} B_{rq} = C_{pq} \quad (\text{A1})$$

and a more general contraction of a fourth-order tensor with a third-order tensor can be drawn as

$$\begin{array}{c} \text{---} \text{V} \text{---} \text{U} \text{---} \\ | \quad | \\ p \quad r \quad t \end{array} \longrightarrow \begin{array}{c} \text{---} \text{W} \text{---} \\ | \\ p \quad r \quad t \end{array} \quad \sum_{qs} V_{pqrs} U_{qst} = W_{prt} \quad (\text{A2})$$

Diagrams can be used to readily estimate the cost of contractions (and other operations). The cost Ω of contracting two tensors over L indices of size $\{\lambda\}_1^L$ to a tensor with M indices of size $\{\mu\}_1^M$ scales with respect to N as

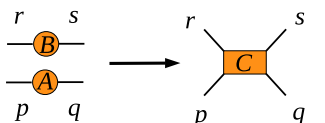
$$\Omega = \mathcal{O}(N^{\sum_{l=1}^L \log_N \dim(\{\lambda\}_l) - \sum_{m=1}^M \log_N \dim(\{\mu\}_m)}). \quad (\text{A3})$$

One can simply estimate the scaling of a contraction by multiplying the dimensions of each open line in the result together with those of each closed line. For example, a contraction of two third-order tensors of size $N \times N \times N$ over two indices of size N scales as $\mathcal{O}(N^4)$,



$$\mathcal{O}(N^4) \quad (\text{A4})$$

Other operations that can be represented pictorially are of an outer product type. This situation corresponds to merging the nodes together and leaving all lines in the final structure



$$\mathcal{O}(N^4) \quad (\text{A5})$$

Note that if one reshapes the fourth-order tensor above into a matrix with combined indices rp and sq , then the result will coincide with the usual Kronecker product of matrices, where we recall that the Kronecker product is

$$C = A \otimes B \Leftrightarrow C_{rp,sq} = A_{p,q} \cdot B_{r,s}. \quad (\text{A6})$$

The cost of product-type operations is

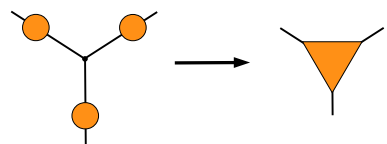
$$\Omega = \mathcal{O}(N^{\sum_{m=1}^M \log_N \dim(\{\mu\}_m)}), \quad (\text{A7})$$

where $\{\mu\}_1^M$ are M free indices in the resulting tensor.

For our purposes, we slightly extended the diagrammatic notation by introducing summations over an index shared by more than two terms. We denote such indices by branching lines with a dot at the branching point. This dot can be interpreted either as an index of the summation itself or as a fully diagonal tensor whose elements are contractions of Kronecker deltas, e.g.,

$$K_{p,q,r,\dots} = \sum_{\alpha} \delta_p^{\alpha} \delta_q^{\alpha} \delta_r^{\alpha} \dots \quad (\text{A8})$$

The latter interpretation means that all contractions in the diagrams can be thought pairwise as in the normal case. Although not quite standard, this extension has been used before in the tensor network literature.⁵⁵ Using our new notation, contracting a canonical polyadic decomposition of a third order-tensor back to a full tensor can be denoted as



$$\mathcal{O}(N^4) \quad (\text{A9})$$

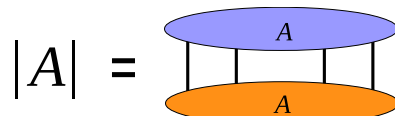
$$T_{pqr} = \sum_{\alpha} A_{p\alpha} B_{q\alpha} C_{r\alpha}$$

If the dimensions of this tensor are $N \times N \times N$ and the rank of the decomposition (the dimension of the auxiliary index α) is N , then the cost of rebuilding the original tensor from its decomposed form will scale as $\mathcal{O}(N^4)$. We note that Eq. (A3) holds in this case just the same way as with normal pairwise contractions.

Let us also list diagrammatic representations of common matrix operations. The Frobenius norm of a tensor, which we recall is

$$\|A\| = \sqrt{\sum_{pqrs\dots} A_{pqrs\dots} A_{pqrs\dots}^*} \quad (\text{A10})$$

is given diagrammatically as the square root of a tensor fully contracted with its own conjugate,



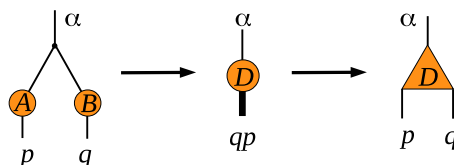
$$|A| = \quad (\text{A11})$$

We have used a darker color to denote complex conjugation here.

The column-wise Khatri-Rao product is

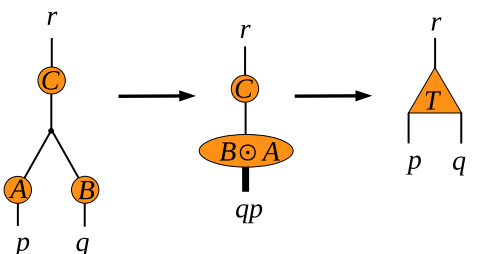
$$D = A \odot B \Leftrightarrow D_{qp,\alpha} = A_{p,\alpha} \cdot B_{q,\alpha}. \quad (\text{A12})$$

Note that A and B should have the same number of columns to be compatible. The resulting matrix D can be reshaped to a third-order tensor with indices p , q , and α . Diagrammatically, the Khatri-Rao product is



$$\quad (\text{A13})$$

Here we used a thick line to denote a combined index qp . Note also that the canonical polyadic decomposition can be conveniently expressed through the Khatri-Rao product, which is also reflected by the diagrams



$$\quad (\text{A14})$$

$$T_{pqr} = \sum_{\alpha} A_{p\alpha} B_{q\alpha} C_{r\alpha} = (A \odot B \cdot C^T)_{pqr}$$

Finally, we point out that wiring diagrams provide an easy way to calculate derivatives. A partial derivative of a tensor network with respect to one of its component tensors is simply the network with that tensor removed.

¹F. Weigend, M. Kattannek, and R. Ahlrichs, *J. Chem. Phys.* **130**, 164106 (2009).

²E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, *J. Chem. Phys.* **137**, 044103 (2012).

³R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, *J. Chem. Phys.* **137**, 224106 (2012).

⁴E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martínez, *J. Chem. Phys.* **137**, 221101 (2012).

⁵S. I. Kokkila Schumacher, E. G. Hohenstein, R. M. Parrish, L.-P. Wang, and T. J. Martínez, *J. Chem. Theor. Comput.* **11**, 3042 (2015).

⁶R. M. Parrish, C. D. Sherrill, E. G. Hohenstein, S. I. Kokkila, and T. J. Martínez, "Communication: Acceleration of coupled cluster singles and doubles via orbital-weighted least-squares tensor hypercontraction," *J. Chem. Phys.* **140**, 181102 (2014).

- ⁷E. G. Hohenstein, S. I. Korkkila, R. M. Parrish, and T. J. Martínez, *J. Phys. Chem. B* **117**, 12972 (2013).
- ⁸F. L. Hitchcock, *Stud. Appl. Math.* **6**, 164 (1927).
- ⁹L. De Lathauwer, *SIAM J. Matrix Anal. Appl.* **28**, 642 (2006).
- ¹⁰E. G. Hohenstein, S. I. Korkkila, R. M. Parrish, and T. J. Martínez, *J. Chem. Phys.* **138**, 124111 (2013).
- ¹¹N. Shenvi, H. Van Aggelen, Y. Yang, W. Yang, C. Schwerdtfeger, and D. Mazziotti, *J. Chem. Phys.* **139**, 054110 (2013).
- ¹²U. Benedikt, A. A. Auer, M. Espig, and W. Hackbusch, *J. Chem. Phys.* **134**, 054118 (2011).
- ¹³U. Benedikt, K.-H. Böhm, and A. A. Auer, *J. Chem. Phys.* **139**, 224101 (2013).
- ¹⁴F. Hummel, T. Tsatsoulis, and A. Grüneis, *J. Chem. Phys.* **146**, 124105 (2017).
- ¹⁵G. D. Purvis III and R. J. Bartlett, *J. Chem. Phys.* **76**, 1910 (1982).
- ¹⁶G. E. Scuseria, C. L. Janssen, and H. F. Schaefer III, *J. Chem. Phys.* **89**, 7382 (1988).
- ¹⁷G. R. Ahmadi and J. Almlöf, *Chem. Phys. Lett.* **246**, 364 (1995).
- ¹⁸F. Neese, A. Hansen, and D. G. Liakos, *J. Chem. Phys.* **131**, 064103 (2009).
- ¹⁹C. Riplinger and F. Neese, *J. Chem. Phys.* **138**, 034106 (2013).
- ²⁰P. Y. Ayala and G. E. Scuseria, *J. Chem. Phys.* **110**, 3660 (1999).
- ²¹M. Degroote, T. M. Henderson, J. Zhao, J. Dukelsky, and G. E. Scuseria, *Phys. Rev. B* **93**, 125124 (2016).
- ²²J. A. Gomez, T. M. Henderson, and G. E. Scuseria, *Mol. Phys.* **115**, 2673 (2017).
- ²³Y. Qiu, T. M. Henderson, J. Zhao, and G. E. Scuseria, *J. Chem. Phys.* **147**, 064111 (2017).
- ²⁴E. Neuscamman, *J. Chem. Phys.* **139**, 181101 (2013).
- ²⁵J. Wahlen-Strothman, C. A. Jiménez-Hoyos, T. M. Henderson, and G. E. Scuseria, *Phys. Rev. B* **91**, 041114(R) (2015).
- ²⁶J. Wahlen-Strothman and G. E. Scuseria, *J. Phys.: Condens. Matter* **28**, 485502 (2016).
- ²⁷T. G. Kolda and B. W. Bader, *SIAM Rev.* **51**, 455 (2009).
- ²⁸S. Liu and G. Trenkler, *Int. J. Inf. Syst. Sci.* **4**, 160 (2008).
- ²⁹J. C. A. Barata and M. S. Hussein, *Braz. J. Phys.* **42**, 146 (2012).
- ³⁰N. H. Beebe and J. Linderberg, *Int. J. Quantum Chem.* **12**, 683 (1977).
- ³¹O. Vahtras, J. Almlöf, and M. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).
- ³²L. Boman, H. Koch, and A. Sánchez de Merás, *J. Chem. Phys.* **129**, 134107 (2008).
- ³³M. Sierka, A. Hogekamp, and R. Ahlrichs, *J. Chem. Phys.* **118**, 9136 (2003).
- ³⁴C. Eckart and G. Young, *Psychometrika* **1**, 211 (1936).
- ³⁵H. Koch, A. Sánchez de Merás, and T. B. Pedersen, *J. Chem. Phys.* **118**, 9481 (2003).
- ³⁶H. Harbrecht, M. Peters, and R. Schneider, *Appl. Numer. Math.* **62**, 428 (2012).
- ³⁷H.-J. Werner, F. R. Manby, and P. J. Knowles, *J. Chem. Phys.* **118**, 8149 (2003).
- ³⁸A. F. Izmaylov and G. E. Scuseria, *Phys. Chem. Chem. Phys.* **10**, 3421 (2008).
- ³⁹J. B. Kruskal, *Linear Algebra Appl.* **18**, 95 (1977).
- ⁴⁰N. D. Sidiropoulos and R. Bro, *J. Chemom.* **14**, 229 (2000).
- ⁴¹L. Sorber, M. Van Barel, and L. De Lathauwer, *SIAM J. Optim.* **23**, 695 (2013).
- ⁴²P. Comon, X. Luciani, and A. L. De Almeida, *J. Chemom.* **23**, 393 (2009).
- ⁴³N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, *IEEE Trans. Signal Proc.* **65**, 3551 (2017).
- ⁴⁴N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, URL: <http://www.tensorlab.net>.
- ⁴⁵A. Uschmajew, *SIAM J. Matrix Anal. Appl.* **33**, 639 (2012).
- ⁴⁶L. Sorber, M. V. Barel, and L. D. Lathauwer, *SIAM J. Optim.* **22**, 879 (2012).
- ⁴⁷J. Zhao and G. E. Scuseria, "Generic and efficient canonicalization of combinatorial objects," <http://jz21.web.rice.edu/drudge/> (unpublished).
- ⁴⁸J. Zhao and G. E. Scuseria, "Efficient optimization of tensor contractions, parts i and ii," <http://jz21.web.rice.edu/gristmill/> (unpublished).
- ⁴⁹E. D. Dolan and J. J. Moré, *Math. Program.* **91**, 201 (2002).
- ⁵⁰D. Braess and W. Hackbusch, *IMA J. Numer. Anal.* **25**, 685 (2005).
- ⁵¹J. Almlöf, *Chem. Phys. Lett.* **181**, 319 (1991).
- ⁵²K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li, and T. L. Windus, *J. Chem. Inf. Model.* **47**, 1045 (2007).
- ⁵³R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, *J. Chem. Phys.* **138**, 194107 (2013).
- ⁵⁴Y. Qiu, T. M. Henderson, and G. E. Scuseria, *J. Chem. Phys.* **146**, 184105 (2017).
- ⁵⁵L. Ying, *Multiscale Model. Simul.* **15**, 1423 (2017).