

Tensor hypercontracted ppRPA: Reducing the cost of the particle-particle random phase approximation from $O(r^6)$ to $O(r^4)$

Neil Shenvi, Helen van Aggelen, Yang Yang, and Weitao Yang

Citation: *The Journal of Chemical Physics* **141**, 024119 (2014); doi: 10.1063/1.4886584

View online: <http://dx.doi.org/10.1063/1.4886584>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/141/2?ver=pdfcov>

Published by the [AIP Publishing](#)

Articles you may be interested in

[Excitation energies from particle-particle random phase approximation: Davidson algorithm and benchmark studies](#)

J. Chem. Phys. **141**, 124104 (2014); 10.1063/1.4895792

[Exchange-correlation energy from pairing matrix fluctuation and the particle-particle random phase approximation](#)

J. Chem. Phys. **140**, 18A511 (2014); 10.1063/1.4865816

[Double, Rydberg and charge transfer excitations from pairing matrix fluctuation and particle-particle random phase approximation](#)

J. Chem. Phys. **139**, 224105 (2013); 10.1063/1.4834875

[Benchmark tests and spin adaptation for the particle-particle random phase approximation](#)

J. Chem. Phys. **139**, 174110 (2013); 10.1063/1.4828728

[Equivalence of particle-particle random phase approximation correlation energy and ladder-coupled-cluster doubles](#)

J. Chem. Phys. **139**, 104112 (2013); 10.1063/1.4820556



Tensor hypercontracted ppRPA: Reducing the cost of the particle-particle random phase approximation from $O(r^6)$ to $O(r^4)$

Neil Shenvi,¹ Helen van Aggelen,² Yang Yang,¹ and Weitao Yang¹

¹Department of Chemistry, Duke University, Durham, NC 27708, USA

²Department of Chemistry, Princeton University, Princeton, New Jersey 08544, USA

(Received 8 April 2014; accepted 22 June 2014; published online 14 July 2014)

In recent years, interest in the random-phase approximation (RPA) has grown rapidly. At the same time, tensor hypercontraction has emerged as an intriguing method to reduce the computational cost of electronic structure algorithms. In this paper, we combine the particle-particle random phase approximation with tensor hypercontraction to produce the tensor-hypercontracted particle-particle RPA (THC-ppRPA) algorithm. Unlike previous implementations of ppRPA which scale as $O(r^6)$, the THC-ppRPA algorithm scales asymptotically as only $O(r^4)$, albeit with a much larger prefactor than the traditional algorithm. We apply THC-ppRPA to several model systems and show that it yields the same results as traditional ppRPA to within mH accuracy. Our method opens the door to the development of post-Kohn Sham functionals based on ppRPA without the excessive asymptotic cost of traditional ppRPA implementations. © 2014 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4886584>]

I. INTRODUCTION

The random phase approximation (RPA) was originally derived in the context of a homogeneous electron gas over 60 years ago.^{1,2} Although there are multiple forms of the RPA, research has largely focused on the particle-hole RPA (which is often referred to in the literature simply as the “RPA” without qualification). Despite some initial success, its application to quantum chemistry was largely overshadowed by the successful development of coupled-cluster theory in the 1970s and 1980s. However, the last 15 years have seen a renewal of interest in the RPA.^{3–6} The RPA is attractive in part because it can capture long-range dispersive interactions,^{7,8} has no static correlation error,^{9–12} and does not experience divergence for materials with small band-gaps.⁵ One of the more interesting theoretical developments to emerge from the study of the RPA was the realization that the phRPA could be rewritten in terms of coupled-cluster theory.¹³ This correspondence was used by several authors^{13,14} to develop efficient algorithms for the solution of the phRPA equations.

More recently, our group has examined the particle-particle RPA (ppRPA).^{15–19} Initial results indicate that, like the phRPA, the ppRPA has several interesting properties as a density functional, and is more accurate than the phRPA for a variety of chemical problems.^{15,18} The ppRPA successfully captures the known fractional-spin and fractional charge constraints on the exact functional, as seen in its correct dissociation of H_2^+ , H_2 , and He_2^+ .¹⁵ The ppRPA also does not suffer from instabilities like the phRPA with exchange (RPAX).¹⁹

Like the phRPA, the ppRPA can be rewritten as a coupled-cluster-like equation, in this case one which only includes so-called “ladder” terms.^{16,20} One shortcoming of the ppRPA is that the particular form taken by its ladder-coupled-cluster-like equations make it resistant to the techniques used to achieve an $O(r^4)$ scaling for the phRPA, where r is the size of the basis set. Whether the ppRPA is written in its matrix form, in its coupled-cluster form, or in its linear, double-

integral form,^{15,19} all methods used for its solution thus far require $O(r^6)$ operations, provided that the number of holes and particles both scale as $O(r)$. In this paper, we will show that a recently developed form of tensor compression can reduce this scaling from $O(r^6)$ to $O(r^4)$ with minimal reduction of accuracy.

Tensor compression decomposes high-rank tensors, which are abundant in electronic structure theory, into lower rank tensors. One well-known example of tensor compression is the resolution-of-identity (RI) technique for the compression of the 2-electron repulsion integral (ERI) tensor

$$\epsilon_{kl}^{ij} = \int \int d\mathbf{r} d\mathbf{r}' \phi_i^*(\mathbf{r}) \phi_k(\mathbf{r}) \frac{1}{|\mathbf{r} - \mathbf{r}'|} \phi_j^*(\mathbf{r}') \phi_l(\mathbf{r}'). \quad (1)$$

RI methods such as RI-SVS or RI-V express this rank-4 tensor as the product of rank-3 and rank-2 tensors^{21–25}

$$\epsilon_{kl}^{ij} = \sum_{P,Q=1}^{P_R} v_{ik}^P J_{PQ} v_{jl}^Q. \quad (2)$$

If the number of auxiliary functions $P_R = r(r+1)/2$, then Eq. (2) is an equality; if $P_H < r(r+1)/2$, then the equality in Eq. (2) is only approximate. However, the locality of the atomic orbitals makes it possible to obtain a very good approximation with only a small number of auxiliary functions, usually $P_R \approx 4r$.

Tensor hypercontraction goes one step further than RI techniques in that it expresses high-rank tensors entirely in terms of rank-2 tensors.^{26–28} For example, the THC expression for the ERI tensor is

$$\epsilon_{kl}^{ij} = \sum_{P,Q=1}^{P_H} h_{iP} h_{kP} H_{PQ} h_{jQ} h_{lQ}. \quad (3)$$

As in the case of RI techniques, the use of $P_H = r(r+1)/2$ auxiliary functions will guarantee an exact representation of the ERI tensor. In practice, it has been shown that only

$P_H = O(r)$ auxiliary functions are needed to obtain a good approximation to the ERI tensor, due again to the locality of the atomic orbitals.²⁶

The THC of the ERIs can be obtained in two steps. First, we select a desired atomic basis (e.g., cc-pVDZ) and apply the standard RI-V method to decompose the ERIs in terms of P_R auxiliary functions in the corresponding auxiliary basis (e.g., cc-pVDZ-RI) from the EMSL basis set library,²⁹ yielding a decomposition of the form in Eq. (2). This step can be performed in $O(r^4)$ operations. We then perform a nonlinear fitting procedure to approximate the rank-3 tensor v_{ik}^P as a product of two rank-2 tensors. Neither of these tensors is specified in advance; they are obtained through finding the best least-squares fit to the tensor v_{ik}^P . This step is described in detail in Ref. 30 and also requires $O(r^4)$ operations, so that the entire THC decomposition procedure scales as $O(r^4)$. One added advantage of the THC decomposition is that orbital rotations of the ERI tensor can be performed in $O(r^3)$ operations by simply applying the orbital rotation matrix to the rank-2 tensor h_{iP} .

Tensor hypercontraction can be applied to other objects in electronic structure theory besides the ERI tensor, especially to excitation tensors T_{ij}^{ab} such as those found in CISD,³¹ CCSD,²⁸ CC2,³² p2RDM theory,³⁰ and phRPA theory.¹⁴ In all of these examples, it was found that the excitation tensor could be compressed accurately using only $P_A = O(r)$ auxiliary functions rather than the $P_A = O(r^2)$ auxiliary functions needed for an exact representation of the tensor. When both the excitation tensor and the ERI tensor were compressed, the asymptotic cost of the resulting algorithm was generally reduced by a factor of r^2 . For example, it was found that the CISD, CCSD, and p2RDM algorithms—which all scale as $O(r^6)$ in their traditional implementations—can be reduced in cost to a scaling of $O(r^4)$ by writing both the ERI tensor ϵ_{kl}^{ij} and the excitation tensor T_{ij}^{ab} as a THC. In this paper, we will apply THC to the ppRPA algorithm. Consistent with these previous results, we find that: (1) the excitation tensor T_{ij}^{ab} used in ppRPA can be accurately compressed using only $P_A = O(r)$ auxiliary functions, and (2) the use of THC reduces the asymptotic scaling of the algorithm from $O(r^6)$ to $O(r^4)$.

The remainder of our paper will be organized as follows: in Sec. II, we will describe the THC-ppRPA algorithm. In Sec. III, we will present the results of applying the THC-ppRPA algorithm to several sets of small molecules, hydrogen chains, and alkanes, comparing our results to the exact ppRPA results obtained through traditional methods. We will show that only a linear number of auxiliary functions are needed to obtain good accuracy and that, when a linear number of auxiliary functions is used, the algorithm does indeed scale as $O(r^4)$. In Sec. IV, we will present our conclusions and areas for future work.

II. ALGORITHM

A. Traditional ppRPA

There are several equivalent formulations of the ppRPA equations,¹⁵ but we will use the formulation in terms of a Riccati equation, as described in Refs. 14 and 20. After some

rearrangement, we find that the solution of the ppRPA equations requires us to solve

$$T_{ij}^{ab} = -\frac{1}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} \times (\epsilon_{ij}^{ab} + T_{cd}^{ab} \epsilon_{ij}^{cd} + \epsilon_{kl}^{ab} T_{ij}^{kl} + T_{kl}^{ab} \epsilon_{cd}^{kl} T_{ij}^{cd}). \quad (4)$$

Here, ϵ_a and ϵ_i are the orbital energies of the unoccupied and occupied orbitals a and i in the reference state. Note that this reference state can be chosen to be either the Hartree-Fock state or a Kohn-Sham reference state using some particular DFT functional. The use of Kohn-Sham reference systems can be justified based on the formulation of time-dependent DFT on the pairing field.³³ Although this approach is not common practice in coupled cluster theory, DFT references are often used in Green's function methods, including the RPA. When we solve Eq. (4) for the excitation tensor T_{ij}^{ab} , the ppRPA correlation energy is given by

$$E_c = \frac{1}{4} \sum_{ijab} \epsilon_{ijab}^{ab} T_{ij}^{ab}. \quad (5)$$

When spin is taken into account, the RPA equations for closed-shell singlet systems separate into two different channels: a same-spin $\alpha\alpha$ channel and an opposite-spin $\alpha\beta$ channel. The $\beta\beta$ channel is identical to the $\alpha\alpha$ channel since the α and β spins are equivalent for singlets. Furthermore, the solution ${}^{\alpha\alpha}T$ of the same-spin equations is simply the anti-symmetrized version of the solution ${}^{\alpha\beta}T$ of the opposite-spin equations such that the total correlation energy can be written as

$$E_c = \sum_{ijab} \epsilon_{ijab}^{ab} (2 {}^{\alpha\beta}T_{ij}^{ab} - {}^{\alpha\beta}T_{ij}^{ba}), \quad (6)$$

where the second term captures the effects of exchange and could be dropped to implement an exchange-free DFT functional. Because of this simplification, we will first calculate the solution ${}^{\alpha\beta}T$ of the opposite-spin channel and then use this solution to calculate the energy in both the opposite-spin and same-spin channels. Although there is no reason that our method cannot be applied to non-singlet states, we will assume for the remainder of the paper that we are dealing with closed-shell singlets to avoid the costs and complications of higher spin states.

In the traditional implementation of the algorithm, Eq. (4) can be solved iteratively in much the same way that coupled-cluster equations are solved. Assuming that T_{ij}^{ab} is small, we can adopt the following approach:

1. Set ${}^{(0)}T_{ij}^{ab} = 0$, $n = 1$.
2. Calculate

$${}^{(n)}T_{ij}^{ab} = -\frac{1}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} (\epsilon_{ij}^{ab} + {}^{(n-1)}T_{cd}^{ab} \epsilon_{ij}^{cd} + \epsilon_{kl}^{ab} {}^{(n-1)}T_{ij}^{kl} + {}^{(n-1)}T_{kl}^{ab} \epsilon_{cd}^{kl} {}^{(n-1)}T_{ij}^{cd}). \quad (7)$$

3. If $\|{}^{(n)}T - {}^{(n-1)}T\|$ is sufficiently small or some maximum number of iterations has been reached, then stop. Otherwise, set $n \rightarrow n + 1$ and return to step 2.

Although this method will work, it will unavoidably scale as $O(r^6)$ because step 2 requires the matrix multiplication of ϵ and T , both of which are $O(r^2) \times O(r^2)$ matrices. To achieve a speed-up using THC methodology, we need to avoid explicit evaluation or multiplication of the matrices ϵ and T . Fortunately, it is possible to adapt the iterative methodology outlined above to create an efficient algorithm, which we describe in Sec. II B.

B. THC methodology

To achieve a speed-up for our algorithm, both ϵ and T must be written in THC form. There exist numerous methodologies for writing the ERI tensor ϵ_{kl}^{ij} as a tensor hypercontraction^{26,27,30,34,35} and research into making this process more efficient is ongoing. Details of our methodology for decomposing the ERI tensor are given in Ref. 30; for our current work, we will simply assume that the molecular ERI tensor has been expressed as a THC of the form in Eq. (3) using $P_H = O(r)$ auxiliary functions through one of the available methods.

Next, we have to represent the excitation tensor T_{ij}^{ab} as a THC. We slightly modify the THC structure in Eq. (3) to take into account the symmetry of T_{ij}^{ab}

$$T_{ij}^{ab} = \sum_{P,Q=1}^{P_A} y_{aP} x_{iP} z_{PQ} y_{bQ} x_{jQ}. \quad (8)$$

In Eq. (8), the matrix z_{PQ} is symmetric such that the symmetry $(a, i) \leftrightarrow (b, j)$ is preserved. We also recall that the antisymmetry of $a \leftrightarrow b$ and $i \leftrightarrow j$ need not be taken into account since we are solving only the opposite-spin channel; the correlation energy of the same-spin channel will be included at the end of the calculation when we evaluate Eq. (6).

As we said, in order to achieve an improvement in algorithmic efficiency, we must avoid any explicit evaluation of T_{ij}^{ab} from its THC components; all quantities must be calculated as functions of the THC components x_{iP} , y_{aP} , and z_{PQ} without ever explicitly generating T . To do so, we take a cue

from previous uses of THC and recast our problem as a functional minimization. We can define the cost function J

$$J^{(n)}(T) = \sum_{abij} ((\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)^{(n)} T_{ij}^{ab} - f_{ij}^{ab(n-1)} T)^2, \quad (9)$$

where

$$f_{ij}^{ab(n-1)} T = -(\epsilon_{ij}^{ab} + \epsilon_{cd}^{ab(n-1)} T_{ij}^{cd} + \epsilon_{kl}^{ab(n-1)} T_{ij}^{kl} + {}^{(n-1)} T_{kl}^{ab} \epsilon_{cd}^{kl(n-1)} T_{ij}^{cd}). \quad (10)$$

But if we remember that both T_{ij}^{ab} and ϵ_{kl}^{ij} can be written as tensor hypercontractions, then J can be written as a function of x_{iP} , y_{aP} , and z_{PQ} without any explicit reference to T . Thus, we can minimize Eq. (9) with respect to the THC parameters. If we find a solution for which $J = 0$, then we have solved the iterative Riccati question in Eq. (7).

To make our iterative procedure more stable, we add a damping term to Eq. (9) which tends to lead to smoother convergence. Our final cost function is

$$J^{(n)}(T) = p \sum_{abij} ((\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)^{(n)} T_{ij}^{ab} - f_{ij}^{ab(n-1)} T)^2 + (1-p) \sum_{abij} ((\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)^{(n)} T_{ij}^{ab} - {}^{(n-1)} T_{ij}^{ab})^2. \quad (11)$$

This damping approach is common in Hartree-Fock calculations and adds minimal computational cost to our algorithm. We find that a selection of $p = 0.5$ is effective at damping oscillations that can occur during iterative optimization.

The cost of our algorithm comes from evaluating the value and gradient of J during our minimization procedure. To see that this cost scales as $O(r^4)$, we can evaluate the various terms in J to see that they require the summation over at most four indices. For instance, one of the terms in our cost function will be $\sum_{abijcd} (\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)^{(n)} T_{ij}^{ab} \epsilon_{cd}^{ab(n-1)} T_{ij}^{cd}$. Remembering the $a \leftrightarrow b$, $i \leftrightarrow j$ symmetry of T and expressing ϵ and T in terms of THCs, we obtain

$$\sum_{abijcd} (\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)^{(n)} T_{ij}^{ab} \epsilon_{cd}^{ab(n-1)} T_{ij}^{cd} = 2 \sum_{PQRSTU} 2y_e h_{PR} \cdot x x_{PT} \cdot y h_{QS} \cdot x x_{QU} \cdot y h_{TR} \cdot y h_{US} \cdot z_{PQ} \cdot H_{RS} \cdot z_{TU} - 2 \sum_{PQRSTU} 2y h_{PR} \cdot x e x_{PT} \cdot y h_{QS} \cdot x x_{QU} \cdot y h_{TR} \cdot y h_{US} \cdot z_{PQ} \cdot H_{RS} \cdot z_{TU}, \quad (12)$$

where we have defined

$$x x_{PT} = \sum_i x_{iP} x_{iT}, \quad (13)$$

$$x e x_{PT} = \sum_i x_{iP} \epsilon_i x_{iT}, \quad (14)$$

$$x h_{PR} = \sum_i x_{iP} h_{iR}, \quad (15)$$

$$y h_{PR} = \sum_a y_{aP} h_{aR}, \quad (16)$$

$$y e h_{PR} = \sum_a y_{aP} \epsilon_a h_{aR}. \quad (17)$$

The sum in Eq. (12) can be evaluated in $O(r^4)$ time, provided that the number of auxiliary functions scales as $P_A = O(r)$. In contrast, evaluating Eq. (12) without using tensor hypercon-

TABLE I. The THC-ppRPA correlation energy in mH for six small molecules in the cc-pVDZ basis as a function of the number of auxiliary functions. The final column lists the result of traditional ppRPA. In all cases, only $P_A = 40$ auxiliary functions were required for the THC-ppRPA result to converge to within a few mH of the traditional ppRPA correlation energy.

Molecule	$P_A = 20$	$P_A = 30$	$P_A = 40$	$P_A = 50$	$P_A = 60$	Exact ppRPA
CH ₂ ($r = 25$)	75.85	79.64	79.90	79.92	79.98	79.83
CO ($r = 30$)	198.80	211.68	214.78	214.18	214.59	214.54
H ₂ O ($r = 25$)	148.79	155.33	154.83	154.97	154.97	154.76
HCN ($r = 35$)	178.75	201.77	207.88	208.99	209.03	208.68
N ₂ ($r = 30$)	209.47	224.03	227.84	228.22	227.89	227.69
NH ₃ ($r = 30$)	128.80	139.21	140.60	140.48	140.39	140.21

traction requires the summation over the six indices a, b, c, d, i, j and $O(r^6)$ time. If we similarly expand the other terms in Eq. (11), we find that J and its gradients $dJ/dx, dJ/dy, dJ/dz$ can all be evaluated in $O(r^4)$ time, provided that both P_A and P_H scale as $O(r)$.

Based on our discussion above, the THC-ppRPA algorithm can be outlined as follows:

1. Randomly initialize $(0)x_{iP}$ and $(0)y_{aP}$. Set $(0)z_{PQ} = 0$ and $n = 1$.
2. Minimize the objective function $J^{(n)}(T)$ in Eq. (11) with respect to $(n)x_{iP}$, $(n)y_{aP}$, and $(n)z_{PQ}$.
3. If $\|(n)T - (n-1)T\|$ is sufficiently small, then stop. Otherwise, set $n \rightarrow n + 1$ and return to step 2.
4. Calculate E_c from Eq. (6)

III. RESULTS

We applied our THC-ppRPA algorithm to several molecules in a variety of basis sets. First, we considered the calculated correlation energy of six small molecules in the cc-pVDZ basis as a function of P_A , the size of the auxiliary basis set. Table I shows the results. As the number of auxiliary functions is increased, the correlation energy approaches the traditional ppRPA result. By the time the number of auxiliary functions P_A is less than twice the size of the basis set size r , the correlation energy is within 1 mH of the traditional result, far less than the error of the traditional ppRPA algorithm itself.

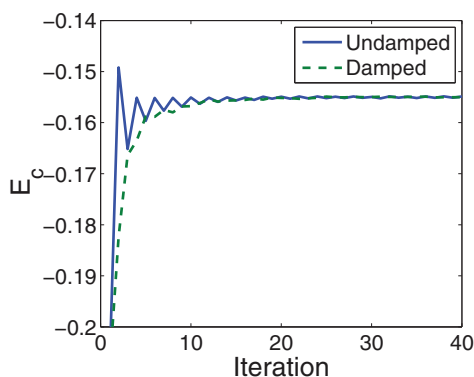


FIG. 1. The convergence of the THC-ppRPA algorithm with and without a damping term for H₂O in the cc-pVDZ basis.

TABLE II. The THC-ppRPA correlation energy in mH for six small molecules in the larger cc-pVTZ basis as a function of the number of auxiliary functions.

Molecule	$P_A = 30$	$P_A = 50$	$P_A = 70$	$P_A = 90$	$P_A = 110$	Exact ppRPA
CH ₂ ($r = 65$)	93.03	104.91	107.87	108.27	108.34	108.49
CO ($r = 70$)	210.79	256.53	278.91	282.59	283.68	283.03
H ₂ O ($r = 65$)	176.11	204.42	207.83	209.89	210.15	209.36
HCN ($r = 85$)	176.35	234.78	258.33	270.04	272.56	272.79
N ₂ ($r = 70$)	232.95	276.52	292.39	295.83	297.08	296.01
NH ₃ ($r = 80$)	154.20	176.52	172.35	185.14	186.59	186.09

We can also examine convergence properties of the THC-ppRPA algorithm with respect to the number of iterations. Figure 1 shows the behavior of the THC-ppRPA algorithm with $P_A = 60$ when applied to H₂O in the cc-pVDZ basis. Because neither the traditional ppRPA algorithm nor the THC-ppRPA algorithm is variational, the THC-ppRPA result will not approach the ppRPA result monotonically. However, the correlation energy does eventually converge to the traditional result. Figure 1 also shows the benefit of including an “anti-damping” term in our cost function, which smoothes out oscillations in the convergence.

We observe the same results when we apply our algorithm to the same six molecules in the larger cc-pVTZ basis set. Table II shows the correlation energies in this larger basis. Once again, auxiliary sets of approximately $P_A = 2r$ functions are sufficient to recover a good approximation to the traditional correlation energy.

These examples are interesting and suggest that the number of auxiliary functions needed to obtain good accuracy is far less than the theoretical limit of $P_A = r(r + 1)/2$. However, they are not clear evidence that the THC-ppRPA algorithm provides an asymptotic speed-up over the traditional approach because the number of electrons in a particular molecule is constant as the basis set size is increased. To obtain evidence of genuine asymptotic scaling improvement, we first examine linear hydrogen chains in the cc-pVDZ basis. In these systems, we can tabulate the fraction of the traditional ppRPA correlation energy recovered by the THC-ppRPA as a function of the number of auxiliary functions. Table III shows our

TABLE III. The percentage of correlation energy recovered by THC-ppRPA algorithm relative to the traditional ppRPA algorithm for hydrogen chains of length N in the cc-pVDZ basis. This table shows that the number of auxiliary functions required to achieve some constant fraction ppRPA energy scales roughly linearly with the size of the system.

Molecule	$P_A = N$	$P_A = 2N$	$P_A = 3N$	$P_A = 4N$	$P_A = 5N$
H ₂	57.8%	90.5%	94.6%	99.8%	100.1%
H ₄	50.8%	79.2%	93.7%	98.1%	99.8%
H ₆	41.2%	75.0%	92.7%	98.2%	99.8%
H ₈	43.3%	73.6%	92.9%	97.6%	99.4%
H ₁₀	43.4%	74.0%	92.3%	97.0%	99.2%
H ₁₂	40.4%	71.0%	91.0%	96.8%	99.2%
H ₁₄	46.8%	74.8%	91.2%	95.9%	98.9%
H ₁₆	44.2%	69.7%	90.5%	95.9%	98.8%

TABLE IV. The percentage of correlation energy recovered by THC-ppRPA algorithm relative to the traditional ppRPA algorithm for alkanes with N carbon atoms in the cc-pVDZ basis. Again, the number of auxiliary functions required to achieve some constant fraction ppRPA energy scales roughly linearly with the size of the system.

Molecule	$P_A = 20N$	$P_A = 40N$	$P_A = 60N$	$P_A = 80N$
CH ₄	87.2%	99.7%	100.1%	100.2%
C ₂ H ₆	89.2%	99.6%	99.9%	100.2%
C ₃ H ₈	90.2%	99.7%	99.8%	100.0%
C ₄ H ₁₀	91.1%	99.4%	100.0%	100.1%

results. Here, we can clearly see that the number of auxiliary functions P_A needed to obtain any particular fraction of the correlation energy (say, 90%) scales linearly with r rather than quadratically. The same result is seen in alkane chains, as shown in Table IV. Once again, the number of auxiliary functions needed scales as r . This result implies that we are achieving an asymptotically significant compression of the excitation tensor T_{ij}^{ab} , since we require only $O(r)$ auxiliary functions to achieve a good approximation instead of the $O(r^2)$ functions which are required in principle to obtain an exact representation.

This linear scaling of P_A with r leads to an overall asymptotic reduction in the complexity of the algorithm, since the evaluation of the cost function and its derivatives require summation over at most 4 indices. We have also verified this scaling numerically. Figure 2 shows the time required for a single evaluation of the cost function and gradient as a function of the size of the hydrogen chain or alkane molecule along with the linear best-fit of the data. The $O(r^4)$ scaling of the algorithm is clearly visible in the case of the hydrogen chain. The slight deviation of the alkane running time scaling from the $O(r^4)$ ideal is likely due to the fact the number of electrons, holes, and basis functions do not grow at a rate exactly proportional to the length of the carbon chain. For instance, if the running time for alkanes were plotted against the number of electrons, a linear best fit would yield a scaling of $r^{4.24}$.

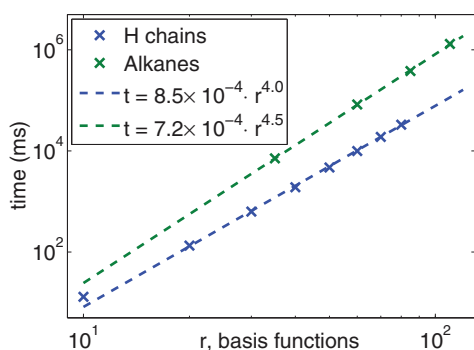


FIG. 2. The computational cost in ms of a single evaluation of the cost function and gradient as a function of the size of the basis set, for hydrogen chains and alkanes. Hydrogen chains show clear $O(r^4)$ scaling. Alkanes show slight deviation, which is primarily due to the fact that the number of electrons and holes are not quite directly proportional to the number of basis functions.

IV. DISCUSSIONS AND CONCLUSIONS

In this article, we have introduced the THC-ppRPA algorithm for improving the asymptotic efficiency of the traditional ppRPA algorithm. Our method relies on the tensor decomposition of both the ERI tensor $^2\epsilon$ and the excitation tensor 2T . Results here and elsewhere confirm that the number of auxiliary functions needed to accurately approximate both of these objects scales as $O(r)$ rather than the theoretical limit of $O(r^2)$, where r is the total number of basis functions in our system. This reduction of order in the number of required auxiliary functions leads to a reduced asymptotic scaling; whereas the traditional ppRPA algorithm scales as $O(r^6)$, our THC-ppRPA algorithm scales as $O(r^4)$.

Two important insights were necessary for the formulation of our algorithm. First, the ppRPA equation had to be rewritten as a coupled-cluster-like equation. We implemented several other variations of THC-ppRPA that relied on double-integral and matrix formulations of the ppRPA equation, but these were unsuccessful in providing a stable, efficient algorithm. Only the coupled-cluster formulation allowed us to solve the ppRPA equations efficiently within a THC expansion. Second, the coupled-cluster equation had to be recast as an iterative, scalar optimization problem. Solving the original Riccati equation in Eq. (4) directly would not lead to a speed-up because it requires us to obtain the full, rank-4 tensor 2T from its THC components, an expansion that requires $O(r^6)$ time. However, we could have converted Eq. (4) into a *non-iterative* scalar optimization problem by minimizing the cost function

$$J(T) = \sum_{abij} ((\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)T_{ij}^{ab} - f_{ij}^{ab}(T))^2. \quad (18)$$

Although this approach would have allowed us to avoid explicitly expanding T , notice that it requires us to calculate the term $\|f_{ij}^{ab}(T)\|^2$ and its derivative. Unfortunately, if we express this term using its THC components, we find that its evaluation carries a cost of $O(r^5)$, which would increase our asymptotic scaling. We circumvented this problem through the iterative approach, which replaced $\|f_{ij}^{ab}(T)\|^2$ with $\|f_{ij}^{ab(n-1)}(T)\|^2$ (see Eq. (11)). This latter term depends on $^{(n-1)}T$ rather than $^{(n)}T$, which is the variable over which we are optimizing. As a result, the derivative of the $\|f_{ij}^{ab(n-1)}(T)\|^2$ term with respect to $^{(n)}T$ is zero. The term can be discarded for the purposes of optimization and the overall $O(r^4)$ scaling of our algorithm can be retained.

At this point, it is helpful to put our work in the context of the broader application of THC to coupled-cluster theory. It was shown in Ref. 28 that CCSD can theoretically be implemented in $O(r^4)$ operations using tensor hypercontraction. Because algorithms like CISD, QCISD,²⁸ and CC2³² can all be obtained through a truncation of terms in the CCSD equations, the result in Ref. 28 proves that the scaling of these other algorithms can also be reduced through the application of THC. What is more, the recent discovery that phRPA and ppRPA can both be written as coupled-cluster-like equations^{16,20} showed that THC-RPA can also be viewed—in principle—as a subset of THC-CCSD. However, in practice,

many of these THC methods must be treated separately from THC-CCSD because they will display quite different scaling behavior than the THC-CCSD algorithm and will make use of different solution techniques.

A good example of the differences that can arise even between closely related THC algorithms is provided by the contrast between phRPA and ppRPA. Both phRPA and ppRPA are—in principle—special cases of the CCSD equations. Both phRPA and ppRPA can also be straightforwardly implemented using exact matrix diagonalization that requires $O(r^6)$ operations.⁵ However, it was shown that the scaling of phRPA could be reduced to $O(r^4 \log r)$ and later to $O(r^3)$ by applying RI³⁶ and THC techniques,¹⁴ respectively. Despite the close connection of the phRPA and ppRPA algorithms, the ppRPA equations cannot be solved as efficiently as the phRPA equations due to the different ways in which the indices are entangled in the two different methods. For example, the most costly term in the Ricatti equations for both phRPA and ppRPA involves the product $T \cdot \epsilon \cdot T$ (see the last term in Eq. (4)). In ppRPA, this term can be written as the THC expansion,

$$\sum_{klcd} t_{kl}^{ab} \epsilon_{cd}^{kl} t_{ij}^{cd} = \sum_{PQTU} y_{aP} \cdot y_{bQ} \cdot x_{iT} \cdot x_{jU} \cdot \left[\sum_{RS} xh_{PR} \cdot xh_{QS} \cdot yh_{TR} \cdot yh_{US} \cdot z_{PQ} \cdot H_{RS} \cdot z_{TU} \right]. \quad (19)$$

Notice that there is no way to evaluate the quantity inside the brackets in fewer than $O(r^5)$ operations. In contrast, the corresponding term in the phRPA equations can be written as

$$\sum_{klcd} t_{ik}^{ac} \epsilon_{cd}^{kl} t_{lj}^{db} = \sum_{PU} y_{aP} \cdot y_{bU} \cdot x_{iP} \cdot x_{jU} \cdot \left[\sum_{QRST} xh_{QR} \cdot yh_{QR} \cdot xh_{TS} \cdot yh_{TS} \cdot z_{PQ} \cdot H_{RS} \cdot z_{TU} \right]. \quad (20)$$

In this case, the quantity inside the brackets can be evaluated in only $O(r^3)$ operations through a clever ordering of summations. This marked difference in complexity is what enables THC-phRPA to be implemented with $O(r^3)$ scaling while the best known scaling for THC-ppRPA is the $O(r^4)$ scaling demonstrated in this paper. This example shows that, despite their similarities, the phRPA and ppRPA equations cannot be solved using the same techniques with the same reduction in cost. The THC-ppRPA algorithm should be developed and studied separately from other THC algorithms, as we have done in this article.

One important question that arises from our work is when the crossover between traditional ppRPA and THC-ppRPA occurs. For what system size r does THC-ppRPA become more efficient than traditional ppRPA? Here, we are hesitant to offer a definitive number for multiple reasons. First, there are many different equivalent approaches for solving the traditional ppRPA equations (matrix diagonalization, double integral formulation, Ricatti formulation) and each will require

different computational resources. For example, our own experience has shown that the solution of the double integral formulation (see Ref. 15) can be tens to hundreds of times more costly than the solution of the matrix formulation. Because the application of the ppRPA method to molecules is relatively new, it is not yet clear which methodology should be viewed as the “standard” one or how it can be implemented most efficiently. The crossover point between traditional ppRPA and THC-ppRPA will therefore depend on which “traditional” ppRPA method is implemented. Second, the cost of the THC-ppRPA algorithm will depend on the number of auxiliary functions P_A and P_H required for an accurate approximation of the excitation tensor and the ERI tensor. Because this number will be strongly system-dependent, the crossover point will likewise be strongly system-dependent. A related concern is that the level of accuracy deemed acceptable will determine the number of auxiliary functions and the degree of convergence required, both of which will affect the THC-ppRPA algorithm’s cost. For all these reasons, while we believe that the crossover point is likely to be large ($r \approx 500$) due to the high prefactor of THC-ppRPA, declaring a single crossover point would depend on too many factors to be of much use. And certainly, future implementations of THC-ppRPA can work on reducing the crossover point without knowing the precise location of the crossover point for a particular system.

Two major areas for future work exist. The most important area for improvement is the large computational prefactor required by our algorithm. While we have shown that our asymptotic scaling is $O(r^4)$, the use of THC introduces a substantial prefactor that, for small values of r , can overwhelm any improvement due to the lower-order scaling. This result is not surprising, given that the same problem is faced by other THC algorithms. Nonetheless, it represents a very important challenge whose solution could significantly improve the practicality of THC-ppRPA. Currently, a *single* evaluation of the THC-ppRPA cost function and its gradient for a problem like H_2O in the cc-pVDZ basis is not significantly more expensive than its solution by traditional ppRPA. However, THC-ppRPA requires many evaluations of the cost function and its gradient, first for the optimization of the cost function J during a single iteration and then for the repeated iterations required to solve the coupled-cluster equation. The overall performance of the algorithm could then be improved either by reducing the number of optimization steps or the number of iterations required for convergence.

A second area of investigation involves the use of THC-ppRPA to correct DFT functionals. While long-range dispersive forces are not captured by many popular DFT functionals, they are captured by RPA.^{5,15,18} The ppRPA also accurately describes the dissociation of H_2^+ , avoiding the delocalization error of DFT, in contrast to phRPA.^{10,12} As a result, we could imagine starting with a DFT calculation and then empirically “mixing in” the results of a ppRPA calculation to qualitatively add in these neglected effects. Because our hybrid scheme would only require inclusion of the qualitative properties of the THC-ppRPA calculation (like its inclusion of dispersion effects), it is possible that a very small number of auxiliary functions could be used to capture these effects, far smaller than the number which

would be needed to obtain a correlation energy that was accurate in some absolute sense. Because the speed of or algorithm depends on the number of auxiliary functions, the use of a very small number of auxiliary functions would lead to a much faster algorithm. Possibly, the auxiliary functions could even be deliberately fixed in such a way that they captured only one particular type of interaction; for instance, we could imagine choosing auxiliary functions that were all strongly localized which were therefore limited to capturing local excitations only. Future research should explore the possibility of correcting popular DFT functionals through the use of THC-ppRPA with a limited auxiliary function set.

Speaking broadly, the applicability of THC to many different electronic structure algorithms is encouraging. It suggests that many if not all electronic structure algorithms can be improved by the application of tensor hypercontraction. Nonetheless, because each algorithm is distinct, it is worthwhile to explicitly develop THC versions of different algorithms to develop a larger toolkit for applying THC to conventional algorithms. For instance, the strategy for applying THC to CISD is different than for applying it to MP2. It is our hope that the extension of THC to the ppRPA algorithm will facilitate further application of THC to areas which have not yet been considered.

ACKNOWLEDGMENTS

N.S. would like to acknowledge support from the UNC EFRC: Center for Solar Fuels, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award No. DE-SC0001011. W.Y. acknowledges support from the National Science Foundation under award CHE-1362927.

¹D. Bohm and D. Pines, *Phys. Rev.* **82**(5), 625 (1951).

²D. Pines and D. Bohm, *Phys. Rev.* **85**(2), 338 (1952).

³F. Furche, *Phys. Rev. B* **64**, 195120 (2001).

⁴A. Heßelmann and A. Görling, *Mol. Phys.* **109**, 2473 (2011).

⁵H. Eshuis, J. E. Bates, and F. Furche, *Theor. Chem. Acc.* **131**, 1084 (2012).

⁶X. Ren, P. Rinke, C. Joas, and M. Scheffler, *J. Mater. Sci.* **47**, 7447 (2012).

⁷A. Szabo and N. S. Ostlund, *J. Chem. Phys.* **67**, 4351 (1977).

⁸H. Eshuis and F. Furche, *J. Chem. Phys. Lett.* **2**, 983 (2011).

⁹M. Fuchs, Y. M. Niquet, X. Gonze, and K. Burke, *J. Chem. Phys.* **122**, 094116 (2005).

¹⁰P. Mori-Sánchez, A. J. Cohen, and W. Yang, *Phys. Rev. A* **85**, 042507 (2012).

¹¹T. M. Henderson and G. E. Scuseria, *Mol. Phys.* **108**, 2511 (2010).

¹²P. Mori-Sánchez, A. J. Cohen, and W. Yang, *Phys. Rev. A* **85**, 042507 (2012).

¹³G. E. Scuseria, T. M. Henderson, and D. C. Sorensen, *J. Chem. Phys.* **129**, 231101 (2008).

¹⁴J. E. Moussa, *J. Chem. Phys.* **140**, 014107 (2014).

¹⁵H. van Aggelen, Y. Yang, and W. Yang, *Phys. Rev. A* **88**, 030501(R) (2013).

¹⁶D. Peng, S. N. Steinmann, H. van Aggelen, and W. Yang, *J. Chem. Phys.* **139**, 104112 (2013).

¹⁷Y. Yang, H. van Aggelen, and W. Yang, *J. Chem. Phys.* **139**, 224105 (2013).

¹⁸Y. Yang, H. van Aggelen, S. N. Steinmann, D. Peng, and W. Yang, *J. Chem. Phys.* **139**, 174110 (2013).

¹⁹H. van Aggelen, Y. Yang, and W. Yang, *J. Chem. Phys.* **140**, 18A511 (2014).

²⁰G. E. Scuseria, T. M. Henderson, and I. W. Bulik, *J. Chem. Phys.* **139**, 104113 (2013).

²¹O. Vahtras, J. Almlöf, and M. W. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).

²²R. Bauernschmitt, M. Häser, O. Treutler, and R. Ahlrichs, *Chem. Phys. Lett.* **264**, 573 (1997).

²³R. A. Kendall and H. A. Früchtl, *Theor. Chem. Acc.* **97**, 158 (1997).

²⁴F. Weigend and M. Häser, *Theor. Chem. Acc.* **97**, 331 (1997).

²⁵X. Ren, P. Rinke, V. Blum, J. Weiferink, A. Tkatchenko, A. Sanfilippo, K. Reuter, and M. Scheffler, *New J. Phys.* **14**, 053020 (2012).

²⁶E. G. Hohenstein, R. M. Parrish, and T. J. Martinez, *J. Chem. Phys.* **137**, 044103 (2012).

²⁷R. M. Parrish, E. G. Hohenstein, T. J. Martinez, and C. D. Sherrill, *J. Chem. Phys.* **137**, 224106 (2012).

²⁸E. G. Hohenstein, R. M. Parrish, C. D. Sherrill, and T. J. Martinez, *J. Chem. Phys.* **137**, 221101 (2012).

²⁹K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li, and T. L. Windus, *J. Chem. Inf. Model.* **47**(3), 1045 (2007).

³⁰N. Shenvi, H. van Aggelen, Y. Yang, W. Yang, C. Schwerdtfeger, and D. A. Mazziotti, *J. Chem. Phys.* **139**, 054110 (2013).

³¹N. Shenvi, H. van Aggelen, and W. Yang, "Using tensor hypercontraction density fitting to achieve an $O(L^4)$ CISD algorithm," preprint [arXiv:1209.2935](https://arxiv.org/abs/1209.2935) [physics.chem-ph] (2012).

³²E. G. Hohenstein, S. I. L. Kokkila, R. M. Parrish, and T. J. Martinez, *J. Chem. Phys.* **138**, 124111 (2013).

³³D. Peng, H. van Aggelen, Y. Yang, and W. Yang, *J. Chem. Phys.* **140**, 18A522 (2014).

³⁴R. M. Parrish, E. G. Hohenstein, T. J. Martinez, and C. D. Sherrill, *J. Chem. Phys.* **138**, 194107 (2013).

³⁵R. M. Parrish, E. G. Hohenstein, N. F. Schunck, C. D. Sherrill, and T. J. Martinez, *Phys. Rev. Lett.* **111**, 132505 (2013).

³⁶H. Eshuis, J. Yarkony, and F. Furche, *J. Chem. Phys.* **132**, 234114 (2010).