JCTC Journal of Chemical Theory and Computation

Article

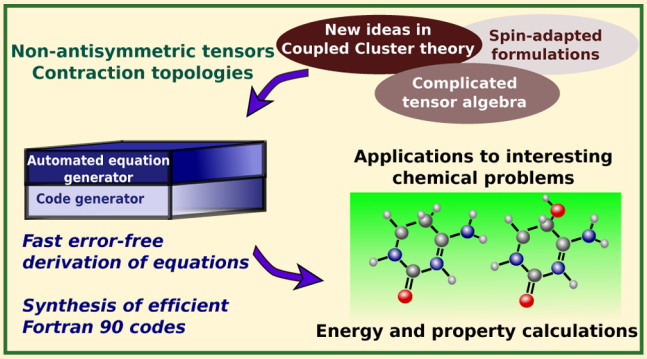
pubs.acs.org/JCTC

A Non-antisymmetric Tensor Contraction Engine for the Automated Implementation of Spin-Adapted Coupled Cluster Approaches

Dipayan Datta* and Jürgen Gauss*

Institut für Physikalische Chemie, Johannes Gutenberg-Universität Mainz, Duesbergweg 10-14, D-55128 Mainz, Germany

ABSTRACT: We present a symbolic manipulation algorithm for the efficient automated implementation of rigorously spin-free coupled cluster (CC) theories based on a unitary group parametrization. Due to the lack of antisymmetry of the unitary group generators under index permutations, all quantities involved in the equations are expressed in terms of non-antisymmetric tensors. Given two tensors, all possible contractions are first generated by applying Wick's theorem. Each term is then put down in the form of a non-antisymmetric Goldstone diagram by assigning its contraction topology. The subsequent simplification of the equations by summing up equivalent terms and their factorization by identifying common intermediates is performed via comparison of these contraction topologies. The definition of the contraction topology is completely general for non-antisymmetric Goldstone diagrams, which enables our algorithm to deal with noncommuting excitations in the cluster operator that arises in the unitary group based CC formulation for open-shell systems. The resulting equations are implemented in a new code, in which tensor contractions are performed by successive application of matrix−matrix multiplications. Implementation of the unitary group adapted CC equations for closed-shell systems and for the simplest open-shell case, i.e., doublets, is discussed, and representative calculations are presented in order to assess the efficiency of the generated codes.



1. INTRODUCTION

Coupled cluster theory,[1−3] in particular in its CCSD(T)[4,5] variant with a perturbative treatment of triple excitations on top of a CC singles-doubles (CCSD) calculation, is one of the most reliable tools for the accurate calculation of molecular energies and properties. State-of-the-art computer programs have been developed for CCSD,[6−9] and also for CC methods involving higher excitations, e.g., CCSDT,[10,11] and CCSDTQ.[12] Recently, there has been growing interest in including even higher excitations in the CC equations, e.g., 5- or 6-fold excitations. Unlike CCSD and CCSDT, for which the equations may be manually derived by using, for example, diagrammatic techniques (see, e.g., ref 2), an error-free derivation of the CC equations including higher excitations goes beyond the level of human perception. A number of computer-aided tools have been developed during the past two decades,[13−18] which allow the automated derivation of the CC equations and the synthesis of efficient computer codes. While these tools offer a general algorithm for including arbitrary rank excitations in the CC equations within the most widely used framework of CC theory, there still remain important aspects, which might lead to a rather nonstandard structure of the equations. One of these aspects is spin-adaptation, which may require a parametrization of the cluster operator in terms of spin-free noncommuting excitations. Most of the general CC algorithms work with the spin−orbital based formulation of CC theory, which leads to spin-contamination of the wave function for nonsinglet species.[19−21] It is therefore important to develop a new symbolic manipulation algorithm that can handle more general formulations in CC theory, especially the spin-adapted formulation, which we propose in this paper.

Among the automated implementation techniques that have appeared in the literature, the developments by Olsen,[13] Kállay and Surján,[14] and Hirata and Bartlett[22] are based on the use of determinant-based techniques, in which the CC equations are cast into determinantal expectation values by applying a projection (de-excitation) operator onto the transformed Hamiltonian. An alternative strategy is to exploit the many-body structure of the CC equations without using projection operators. Kállay and Surján explored this many-body formulation in their further development,[15] which provides codes with correct scaling for solving the CC equations in contrast to the determinant-based algorithms. The method of Kállay and Surján[15] combines the diagrammatic representation of the CC equations with a string-based algorithm and is one of the most efficient general order CC algorithms proposed so far. Another important symbolic manipulation tool is the "tensor contraction engine" (TCE) developed by Hirata[16] together with several other groups,[23] which automates the derivation and factorization of the CC equations and implements the equations into an efficient parallel program. Other noteworthy developments include those by Nooijen and Lotrich,[17] and the most recent general CC algorithm by Hanrath and Engels-

Received: March 18, 2013

   dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Putzka,[18] which employs genetic algorithms to achieve a cost-minimal factorization of the CC equations. Further approaches by Kong[24] [known as the "automatic program generator" (APG)] and Hanauer and Köhn[25] primarily focus on the implementation of multireference CC methods. As already mentioned, most symbolic manipulation algorithms are based on a spin−orbital formulation of CC theory and use antisymmetrized tensors to represent integrals and cluster amplitudes. The only exception is the APG,[24] which accommodates both spin-free as well as spin−orbital based formulations.

Spin adaptation in CC theory is an important field of research. Although the extent of spin-contamination in spin−orbital based CC theory is small in most cases[21] and therefore may not greatly affect molecular energies, it may have a significant impact on spin-dependent molecular properties. Computation of properties for high-spin open-shell systems using spin−orbital based CC theory is well established in the literature.[20,26−30] The effect of spin-contamination, however, has yet to be investigated.

Important developments in the formulation of spin-adapted CC theory include the partially spin-adapted approaches,[31−33] the spin-restricted CC theory,[34] and its extension to a fully spin-adapted CC theory.[35] The spin-restricted approach ensures that the expectation value of the $\hat{S}^2$ operator is preserved in the CC treatment. Rigorous spin-adaptation of the wave function may preferably be achieved by defining the cluster operator in terms of spin-free generators of the unitary group, which are defined in terms of spatial orbitals.[31,36] Powers of the cluster operator are thus expressed in terms of products of the generators of the unitary group, and hence the cluster ansatz becomes spin-free. The high-spin open-shell CC approach of Janssen and Schaefer[31] and the unitary group based CC approach of Li and Paldus[36,37] use this type of parametrization. Unlike in spin−orbital based CC theory, the components of the spin-free cluster operator $\hat{T}$ can contract among themselves and are generally noncommuting. This imparts substantial complexity to the amplitude equations: the equations have a nonterminating structure.

Very recently, Datta and Mukherjee developed a combinatoric cluster expansion ansatz for the unitary group based spin-free cluster operator.[38,39] It was argued, based on an expansion of the wave operator in orders of perturbation,[40] that the structure of the cluster ansatz does not remain exponential if the cluster operator has noncommuting excitations in its definition. The new ansatz, which was termed the "combinatoric open-shell CC" (COSCC) ansatz, has the same factor $1/n!$ for $\hat{T}^n$ as in an exponential if the excitations in $\hat{T}^n$ commute. But in case they do not commute, the factor $1/n!$ is replaced by $1/f_n$, where f_n is the number of permutations of the n operators in which they are contracted among themselves such that the composite excitation is the same. The factor f_n was termed the "automorphic factor" of the composite. It was shown that the use of the COSCC ansatz leads to a manifest factorization of the amplitude equations,[38,39] such that the final equations are at most quartic in the cluster amplitudes.

The basic motivation for us to develop a new automated implementation tool is to accomplish the derivation and code generation for the spin-adapted CC approaches, in particular the unitary group based formalisms. Thus, our algorithm (a) must use the spin-free non-antisymmetric tensors, owing to the lack of antisymmetry of the unitary group generators under index permutations, and (b) for open-shell CC theory must be able to deal with noncommuting excitations in the cluster operator. (c) For a general implementation of the COSCC ansatz, our algorithm should be able to determine the automorphic factors associated with the contracted composites of operators. (d) We intend to implement a general procedure that can deal with both commuting and noncommuting operators on the same footing. (e) We also want to achieve reasonable efficiency in the CC codes by allowing a proper factorization of the equations. We should emphasize that our aim is not to develop an algorithm that can handle higher excitations in the CC ansatz; i.e., we may not go beyond triple excitations. We rather aim at expanding the scope of the automated implementation tools to handle noncommuting operators and nonexponential ansätze.

We have developed our symbolic manipulation tool by using Python.[41] It consists of two components, an expression generator and a code generator. The expression generator derives the equations and factorizes them. Similar to ref 15, we use a many-body approach, which gives us a proper scaling of the computational cost, e.g., $n_o^2 n_v^4$ for closed-shell CCSD, where n_o and n_v indicate the numbers of occupied and virtual orbitals, respectively. The cluster operator and the integrals are represented by *non-antisymmetric tensors*. Given two tensors, terms with all possible valid contractions are first generated by applying Wick's theorem.[42] This step is algebraic, akin to the TCE or APG. Each term is then specified with the types of contractions that each tensor possesses with the other tensors in the term. We call this collective information, represented by lists of integers, the "contraction topology" of the term. In other words, each term is expressed as a *non-antisymmetric Goldstone diagram*.[43] The subsequent steps of simplifying the expressions by summing up equivalent terms and their factorization by identifying common intermediates are performed entirely by comparing the contraction topologies of the terms. Thus, we avoid putting the terms in a "canonical form."[16,24] We have developed factorization schemes for the equations, which involve defining intermediates by recursive elimination of tensors with common contraction topologies, similar to the algorithm of Kállay and Surján.[15] We have also developed a set of generalized tensor contraction subroutines in Fortran 90[44] that perform tensor contractions via linear operations of matrix−matrix multiplications.

2. THEORETICAL BACKGROUND

2.1. Spin-Adapted CC Theory for Closed-Shell Systems. For closed-shell systems, spin adaptation of the CC wave function is trivial, and spin−orbital based formulations already provide spin-eigenfunctions. Thus, the unitary group based CC theory in this case follows the same pathway as the spin−orbital based CC theory. The correlated wave function is parametrized in terms of an exponential ansatz

$$|\Psi_{CC}\rangle = e^{\hat{T}}|\Phi_0\rangle \tag{1}$$

where Φ_0 is the reference determinant, e.g., the restricted Hartree−Fock (RHF) determinant. The cluster operator $\hat{T}$ is defined as

$$\begin{aligned}\hat{T} &= \hat{T}_1 + \hat{T}_2 + \ldots + \hat{T}_k + \ldots \\ &= t_{1\,p_1}^{h_1}\hat{E}_{h_1}^{p_1} + \frac{1}{2!}t_{2\,p_1p_2}^{h_1h_2}\hat{E}_{h_1h_2}^{p_1p_2} + \ldots + \frac{1}{k!}t_{k\,p_1p_2\ldots p_k}^{h_1h_2\ldots h_k}\hat{E}_{h_1h_2\ldots h_k}^{p_1p_2\ldots p_k} + \ldots\end{aligned} \tag{2}$$

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

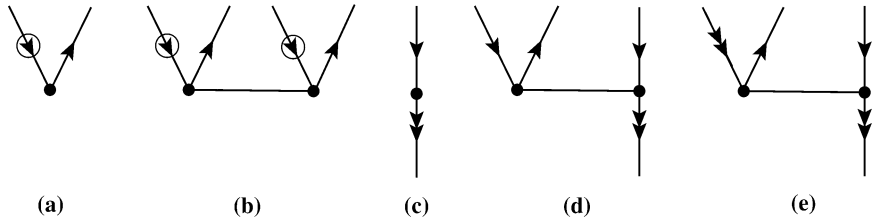


Figure 1. The spin-free singles−doubles excitation operators in the unitary group adapted cluster operator $\hat{T}$ of eq 7 for the doublet case. (a) $t_{1\,p_1}^{\;\;h_1}\hat{E}_{h_1}^{p_1}$, (b) $1/2!t_{2\,p_1p_2}^{\;\;h_1h_2}\hat{E}_{h_1h_2}^{p_1p_2}$, (c) $t_{1e\,u_1}^{(1)\;\;i_1}\hat{E}_{i_1}^{u_1}$, (d) $t_{2e\;\;p_1u_1}^{(1)\;i_1i_2}\hat{E}_{i_1i_2}^{p_1u_1}$, and (e) $t_{2x\,p_1u_1}^{(1)\;\;u_1i_1}\hat{E}_{u_1i_1}^{p_1u_1}$. The arrows enclosed within open circles indicate general hole labels (doubly and singly occupied), while the single arrows pointing downword in (c), (d), and (e) strictly indicate inactive (doubly occupied) hole labels. The double-headed arrows indicate the active or open-shell orbital u_1.

where $\hat{T}_k$ brings in the k-fold excitation by substituting k (spatial) orbitals h_1, h_2, ..., h_k occupied in Φ_0 (holes) with k virtual or unoccupied (spatial) orbitals p_1, p_2, ..., p_k (particles) weighted by the cluster amplitude t_k. The Einstein convention of implied summation is employed in eq 2. The operators $\hat{E}_{h_1}^{p_1}$, $\hat{E}_{h_1h_2}^{p_1p_2}$, etc. represent the unitary group generators, which are defined as the spin-free or spin-singlet excitation operators by summing over the spin−orbital based excitations as

$$\hat{E}_{h_1}^{\;\;p_1} = \sum_{\sigma_1=\alpha,\beta} \hat{a}_{h_{1\sigma_1}}^{\;\;\;p_{1\sigma_1}},\ \hat{E}_{h_1h_2}^{\;\;\;p_1p_2} = \sum_{\sigma_1,\sigma_2=\alpha,\beta} \hat{a}_{h_{1\sigma_1}h_{2\sigma_2}}^{\;\;\;\;\;p_{1\sigma_1}p_{2\sigma_2}} \tag{3}$$

where the orbital indices carrying the subscripts $\sigma_i = \alpha,\beta$ indicate the corresponding spin−orbital indices. The important distinction between the spin-free unitary group generators with the corresponding spin−orbital based excitation operators is the lack of their antisymmetry under index permutations. While the following permutational antisymmetry is allowed for spin−orbital excitations

$$\hat{a}_{RS}^{\;\;PQ} = -\hat{a}_{SR}^{\;\;PQ} = -\hat{a}_{RS}^{\;\;QP} = \hat{a}_{SR}^{\;\;QP} \tag{4}$$

with the upper case letters P, Q, R, and S indicating general spin−orbital indices (holes or particles), the only permutational symmetry allowed for the unitary group generators is

$$\hat{E}_{rs}^{\;\;pq} = \hat{E}_{sr}^{\;\;qp} \neq \hat{E}_{sr}^{\;\;pq} \tag{5}$$

where the lower case letters p, q, r, and s indicate general (spatial) orbital indices. This *lack of permutational antisymmetry of the unitary group generators requires one to express the spin-free cluster amplitudes t_k as non-antisymmetric tensors.*

The energy E and the equations determining the t_k amplitudes are given by

$$\langle\Phi_0|e^{-\hat{T}}\hat{H}_N e^{\hat{T}}|\Phi_0\rangle = E$$
$$\langle\Phi_k|e^{-\hat{T}}\hat{H}_N e^{\hat{T}}|\Phi_0\rangle = 0,\ \forall\ k \tag{6}$$

where $|\Phi\rangle \equiv \hat{E}_{h_1h_2\ldots h_k}^{p_1p_2\ldots p_k}|\Phi_0\rangle$ and $\hat{H}_N$ is the normal-ordered Hamiltonian defined with respect to Φ_0 as the vacuum.

The diagrammatic representation of the terms involved in eqs 6 is well-known in the literature.[1−3] In the unitary group adapted formulation, the spin-free cluster amplitudes and the components of the Hamiltonian are represented by *non-antisymmetric Goldstone diagrams.*[43] The use of non-antisymmetric tensors, however, increases the number of terms in eqs 6 compared to the use of antisymmetrized diagrams. For example, the number of antisymmetrized diagrams in the CCSD doubles amplitude equation is 31 for a non-HF reference, whereas the number of non-antisymmetric diagrams is 55.[2]

2.2. Spin-Adapted CC Theory for Open-Shell Systems Using the COSCC Ansatz. Unlike for closed-shell systems, the unitary group based CC theory for open-shell systems involves new features in the formulation, which are not present in the corresponding spin−orbital based formulation. The reference function is generally taken to be a restricted open-shell HF (ROHF) determinant, which is a spin-eigenfunction and has a number of singly occupied orbitals with either α or β spin electrons. In order to define the excitations in the cluster operator $\hat{T}$ in terms of spatial orbitals similar to eq 2, one necessarily requires the use of a closed-shell vacuum state Φ_0, in which the singly occupied (open-shell) orbitals in the ROHF determinant are either doubly filled or empty.[45] Therefore, the vacuum state used to define the cluster operator (and the Hamiltonian) in second quantization differs from the reference determinant.

For the doublet case, for example, the ROHF determinant Φ_u has a single open-shell orbital, say u_1, and the vacuum state Φ_0 may be chosen as the determinant in which u_1 is doubly occupied. Thus, Φ_u is obtained by annihilating an electron in orbital u_1 in Φ_0: $\Phi_u \equiv u_1|\Phi_0\rangle$. The orbitals are classified as the "inactive holes" (doubly occupied orbitals in Φ_0, denoted by i_1, i_2, ...), "particles" (unoccupied orbitals in Φ_0, denoted by p_1, p_2, ...), and the open-shell or "active" orbital u_1, while h_1, h_2, ... indicate general hole labels (inactive and active). In the current choice of Φ_0, u_1 is a hole-type active orbital. However, the formulation does not change for the choice of a particle-type active orbital. The spin-free cluster operator $\hat{T}$ within the singles−doubles approximation is defined for the doublet case as[38]

$$\begin{aligned}\hat{T} &= \hat{T}^{(0)} + \hat{T}^{(1)} \\ &= t_{1\,p_1}^{\;\;h_1}\hat{E}_{h_1}^{p_1} + \frac{1}{2!}t_{2\,p_1p_2}^{\;\;h_1h_2}\hat{E}_{h_1h_2}^{p_1p_2} + t_{1e\,u_1}^{(1)\;\;i_1}\hat{E}_{i_1}^{u_1} + t_{2e\;\;p_1u_1}^{(1)\;i_1i_2}\hat{E}_{i_1i_2}^{p_1u_1} \\ &\quad + t_{2x\,p_1u_1}^{(1)\;\;u_1i_1}\hat{E}_{u_1i_1}^{p_1u_1}\end{aligned} \tag{7}$$

Thus, $\hat{T}$ consists of closed-shell-like hole−particle excitations (denoted in eq 7 by $\hat{T}^{(0)}$, the "zero-valence" cluster operator) and operators that involve excitations into the active orbital u_1 (denoted by $\hat{T}^{(1)}$, the "one-valence" cluster operator). The excitations in $\hat{T}$ defined in eq 7 are shown diagrammatically in Figure 1. The terminology of an n-valence operator finds its origin in Fock-space CC theory.[46] For the single hole−particle excitation $i_1 \rightarrow u_1$, there are now two linearly independent cluster amplitudes:[36] (a) the zero-valence amplitude $t_{1p_1}^{\;\;\;h_1}$ and (b) the one-valence "exchange spectator" amplitude $t_{2x\,p_1u_1}^{(1)\;\;u_1i_1}$.

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Since the exchange spectators have the same active label appearing as both creation and annihilation, contractions are possible among the components of $\hat{T}$; i.e., the excitations in $\hat{T}$ do not commute. Therefore, the CC amplitude equations do not terminate at quartic power, in contrast to the spin-adapted CC equations for the closed-shell case.

Datta and Mukherjee recently developed the COSCC ansatz[38−40] which is defined as

$$\{\{\exp(\hat{T})\}\} = 1 + \{\hat{T}^{(0)}\} + \{\hat{T}^{(1)}\} + \frac{1}{2!}\{\hat{T}^{(0)}\hat{T}^{(1)}\} + \frac{1}{2!}\{\hat{T}^{(1)}\hat{T}^{(0)}\} + \sum_{i,j}\frac{1}{f_{ij}}\{\overline{\hat{T}_i^{(1)}\hat{T}_j^{(1)}}\} + \sum_{i,j,k}\frac{1}{f_{ijk}}\{\overline{\hat{T}_i^{(1)}\hat{T}_j^{(1)}\hat{T}_k^{(1)}}\} + \frac{1}{2!}\sum_{i,j,k,l}\frac{1}{f_{ij}f_{kl}}\{\overline{\hat{T}_i^{(1)}\hat{T}_j^{(1)}}\cdot\overline{\hat{T}_k^{(1)}\hat{T}_l^{(1)}}\} + \ldots \tag{8}$$

where an overline represents a contraction, and the curly braces indicate normal ordering with respect to Φ_0. The COSCC ansatz represents a polynomial that has different factors associated with the powers of the cluster operator compared to an exponential ansatz. However, the factors are different only for the contracted composites. For a composite of n noncommuting operators, the factor is taken to be $1/f_n$ instead of $1/n!$ in an exponential, where f_n has been termed the automorphic factor of the composite. The components of the $\hat{T}^{(0)}$ operator are assumed to commute with all other excitations in $\hat{T}$ in the definition of the COSCC ansatz.[39]

The rationale for the choice of these automorphic factors is as follows. Not all $n!$ permutations of n excitation operators necessarily lead to the same composite excitation when contractions are possible. The reason is two-fold: contractions may not occur in all permutations, and in general the contracted composites formed in different permutations lead to nonidentical composite excitations. The meaning of "nonidentical excitations" will be elaborated in section 3.2. The counting of the automorphic factor f_n should therefore not be based on the total number of permutations n unlike in an exponential ansatz but rather on the number of permutations (out of $n!$) in which the composite excitations are identical. Thus, the factor $1/f_n$ provides the same book-keeping for the noncommuting excitations in $\hat{T}^n$ as does the factor $1/n!$ for n commuting excitations in an exponential ansatz; viz., each distinct excitation generated by $\hat{T}^n$ is included only once in the ansatz.

Unlike the factor $1/n!$ for n excitation operators in an exponential, the factor $1/f_n$ is factorizable. Let us consider that k excitations out of n commute among themselves. Then there would be a factor $1/k!$ for this composite. If among the rest of the excitations, l and m operators $(k + l + m = n)$ form separate contracted composites such that f_l and f_m are the number of ways of generating identical excitations from each composite, then there would be two independent automorphic factors $1/f_l$ and $1/f_m$ for these two composites. This implies

$$\frac{1}{f_n} = \frac{1}{k!} \times \frac{1}{f_l} \times \frac{1}{f_m} \tag{9}$$

A rigorously spin-adapted CC theory was developed using the COSCC ansatz for the simplest open-shell systems, e.g., doublet states. In this paper, we focus on the automated implementation of the COSCC amplitude equations. For the theoretical details, we refer to the previous papers.[38−40] The various terms appearing in the amplitude equations were classified into two types: (a) the "strongly connected" composites, in which each t amplitude (zero-valence and one-valence) is contracted to the Hamiltonian vertices through at least one nonactive orbital index, e.g., an inactive hole or a particle, and (b) the "weakly connected" composites, in which the one-valence $t^{(1)}$ amplitudes are joined to each other in a chain-like fashion without any contraction to the Hamiltonian. The presence of these weakly connected terms leads to the nonterminating structure of the amplitude equations, when an exponential ansatz is employed.[31,36] In the COSCC equations, on the other hand, the strongly and weekly connected composites appear with their own "local" automorphic factors, which stems from the factorizability of the automorphic factors as shown in eq 9. This leads to a manifest factorization and cancellation of the weakly connected composites of operators.[38−40]

The final simplified COSCC amplitude equations for the doublet case[38] read

$$\langle\Phi_u^k|(\{\{\hat{H}\exp(\hat{T})\}\}_s - \{\exp(\hat{T}^{(1)})\}_h\hat{H}_{\text{eff}})|\Phi_u\rangle = 0, \forall\, k \tag{10}$$

where $|\Phi_u^k\rangle = \hat{E}_k|\Phi_u\rangle$, with k denoting any excitation defined in eq 7. The subscript "s" denotes strongly connected composites, whereas the subscript "h" in the second term indicates that each $t^{(1)}$ amplitude in $\{\exp(\hat{T}^{(1)})\}_h$ is individually contracted to the effective Hamiltonian operator $\hat{H}_{\text{eff}}$ through the active index. The $\hat{H}_{\text{eff}}$ operator provides the final energy E and is a strongly connected operator defined by

$$E = \langle\Phi_u|\hat{H}_{\text{eff}}|\Phi_u\rangle = \langle\Phi_u|\{\{\hat{H}\exp(\hat{T})\}\}_s|\Phi_u\rangle \tag{11}$$

The first term of eq 10 is henceforth referred to as the "direct term" and the second term as the "folded term." Since the Hamiltonian contains at most two-body operators, the maximum number of strongly connected cluster amplitudes in the direct term is four, akin to closed-shell CC theory. There are, however, some terms which cannot be factorized and therefore survive in the final equations.[39] The number of such terms in the case of doublets is only four for COS-CCSD, and thus they may be safely ignored without affecting the numerical results significantly. In the implementation presented in this paper, these terms are not included. Termination of the folded term in eq 10 depends on the valence rank of the $\hat{H}_{\text{eff}}$ operator. For the doublet states, i.e., the one-valence case, the maximum number of cluster amplitudes in the folded terms is four. Since the zero-valence cluster amplitudes are not allowed to contract with the one-valence cluster amplitudes, eq 10 may be factorized by introducing the partially transformed Hamiltonian operator $\hat{\bar{H}} = e^{-\hat{T}(0)}\,\hat{H}_N\,e^{\hat{T}(0)}$ as

$$\langle\Phi_u^k|(\hat{\bar{H}} + \{\{\hat{\bar{H}}\exp(\hat{T}^{(1)})\}\}_s - \{\exp(\hat{T}^{(1)})\}_h \times [\hat{\bar{H}}_{\text{eff}} + \{\{\hat{\bar{H}}\exp(\hat{T}^{(1)})\}\}_{\text{eff}}])|\Phi_u\rangle = 0, \forall\, k \tag{12}$$

Therefore, amplitudes of the $\hat{\bar{H}}$ operator may be used as intermediates to construct the terms involving the $t^{(1)}$ amplitudes. Since the equations for all amplitudes are coupled, these intermediates are to be updated iteratively.

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

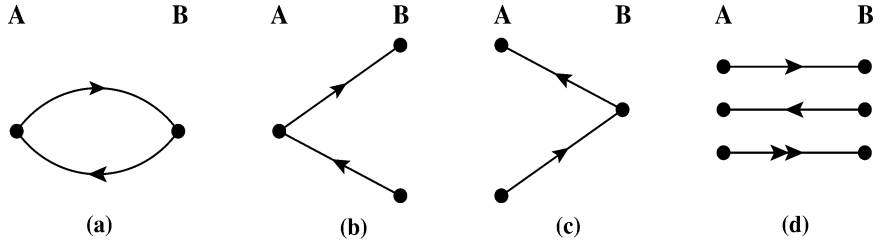


Figure 2. The possible contraction patterns between two non-antisymmetric tensors A and B: (a) loop, (b and c) exchange, and (d) single hole, particle, and active contractions.

3. THE AUTOMATED EXPRESSION AND CODE GENERATOR

3.1. Generation of Equations. The first step is the generation of equations by the expression generator. We employ Wick's theorem[42] to generate terms with all possible valid contractions. Each term is defined as a product of a number of tensors and is symbolically represented as a python class object named "term" with the following attributes:

•Operator: A python class object representing a unitary group generator. It is defined in terms of a list object consisting of vertices. Each vertex consists of a pair of orbital indices. The first is a creation operator, and the second is an annihilation operator. For example, the unitary group generator $\hat{E}^{pq}_{rs}$ is represented in terms of the list of vertices $[(p^{\dagger},r),(q^{\dagger},s)]$.

•Matrix elements: A python list object consisting of the tensors present in the term. Each tensor in turn is represented by a class object with two attributes. The first is the name of the tensor, e.g., f, v, t_1, t_2, etc., where f and v represent the Fock matrix and the non-antisymmetric two-electron integral, and t_1, t_2, etc. represent the one and two-body cluster amplitudes. The second attribute is again a list of vertices similar to the class "operator."

•Coefficient: An integer that is a product of two quantities. The first is the sign and loop factor, $2^{n_l} \times -1^{(n_h+n_l)}$, determined by the total number of contracted hole indices (n_h) in the term and the number of closed loops (n_l) consisting of a hole and a particle index. Each loop gets a factor of 2 in a spin-summed Goldstone diagram. The second factor constituting the coefficient is the automorphic factor of the term, which indicates the number of equivalent ways this term can be formed in different permutations of the tensors present in the term, and under permutation of vertices of each individual tensor.

•Contraction topology: This consists of two sets of information. One is the vertex connections which specifies to which tensors each tensor vertex is contracted. The other information consists of the contraction patterns, which specifies the types of contractions of each tensor with the others in the term. We shall shortly return to a more elaborate description of the contraction topology.

Let us consider two terms, $A \equiv 1/2 v^{g_3g_4}_{g_1g_2} \hat{E}^{g_1g_2}_{g_3g_4}$ and $B \equiv 1/2 t_2{}^{h_1h_2}_{p_1p_2} \hat{E}^{p_1p_2}_{h_1h_2}$, where the automorphic factor 1/2 in each term arises from the pairwise permutation symmetry of the vertices of the corresponding tensors. The two-electron integral v is labeled by general indices g_1, g_2, etc., while the cluster amplitude t_2 is labeled by specified hole labels h_1 and h_2 and particle labels p_1 and p_2. The $\hat{E}$ operators are defined in normal order with respect to the Fermi vacuum Φ_0. The contraction pattern for each term consists of blank lists to begin with.

Given the above two terms, a composite term is first constructed that has the matrix elements $[v^{g_3g_4}_{g_1g_2}, t_2{}^{h_1h_2}_{p_1p_2}]$ and the operator $\hat{E}^{g_1g_2p_1p_2}_{g_3g_4h_1h_2}$. The sign and loop factor initially is 1, and the automorphic factor is $1/2 \times 1/2 = 1/4$. Wick's theorem is then applied on the two strings of creation and annihilation operators $g_1{}^{\dagger}g_3g_2{}^{\dagger}g_4$ and $p_1{}^{\dagger}h_1p_2{}^{\dagger}h_2$, which are constructed from the two input operators $\hat{E}^{g_1g_2}_{g_3g_4}$ and $\hat{E}^{p_1p_2}_{h_1h_2}$, respectively. Contractions are generated recursively in the current algorithm, e.g., single contractions, followed by double contractions, and so on. Contractions are generated *only across the strings*, avoiding self-contractions. For our choice of the Fermi vacuum, a valid hole contraction is the one between a creation on the left and an annihilation on the right, whereas for particle labels, a valid contraction requires the order to be annihilation on the left and creation on the right.

At each step of recursion, a number of pairs of contracted indices, i.e., Kronecker deltas, is generated. These deltas are then used to determine the matrix elements and the operators for the new terms. The deltas are also used to construct the contraction topologies of the new terms in the following way. The vertices of the two tensors $v^{g_3g_4}_{g_1g_2}$ and $t_2{}^{h_1h_2}_{p_1p_2}$ are numbered 1−4 in the same order as they appear on the composite operator $\hat{E}^{g_1g_2p_1p_2}_{g_3g_4h_1h_2}$. The algorithm determines the indices on which two vertices are contracted, and the type of contraction that exists between them, e.g., hole, particle, or active. In the recursive generation of contractions, this information gets accumulated. The number of contractions at each vertex can be at most two. Thus, at each recursion step, we obtain detailed, updated information about the hole, particle, and active contractions that exist between each pair of the four vertices. Of course, there would be no contraction, for example, among the vertices 1 and 2, or 3 and 4, as they belong to the same tensors.

At each step of recursion, the algorithm finds out the resulting "contraction patterns" using the above information. The possible contraction patterns, each of which is represented by an integer, are diagrammatically depicted in Figure 2.

(1) Loop: Both indices on one vertex in A are contracted to the same vertex in B (Figure 2a).
(2) Exchange: One vertex in A is joined to two different vertices in B or vice versa (Figure 2b, c).
(3) Single index contractions: single hole, particle, or active contractions (Figure 2d).

The above three contraction patterns suffice to specify the contraction possibilities between two non-antisymmetric tensors. However, there might exist more than one contraction pattern. In such cases, the contraction pattern is represented by a sum of the integers representing each pattern.

Let us consider, as an example, some terms with double contractions between A and B as shown in Figure 3. The

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

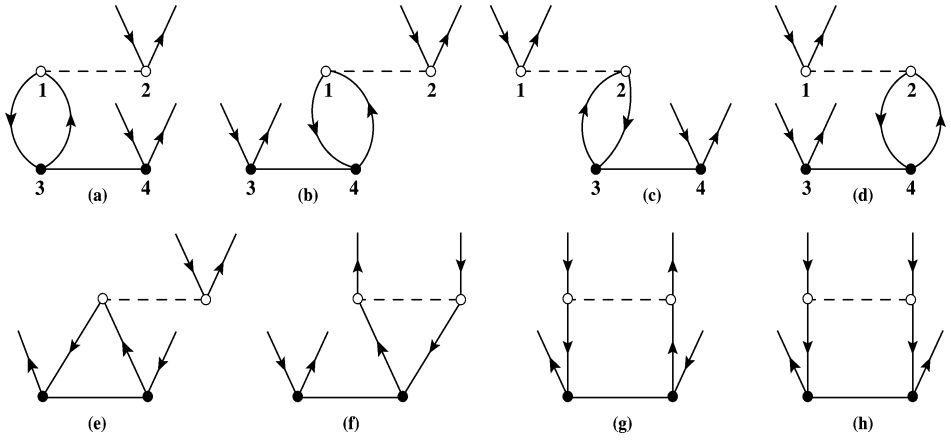


Figure 3. Non-antisymmetric Goldstone diagrams generated from the double contractions between the $v_{g_1g_2}^{g_3g_4}$ and $t_2{}_{p_1p_2}^{h_1h_2}$ tensors. The dashed line with open circles indicates the two-electron integral v. Diagrams (a)−(d) are topologically equivalent, while (e)−(h) are topologically distinct. The vertices in diagrams (a)−(d) have been numbered for clarifying the correspondence between them and the terms of eq 13.

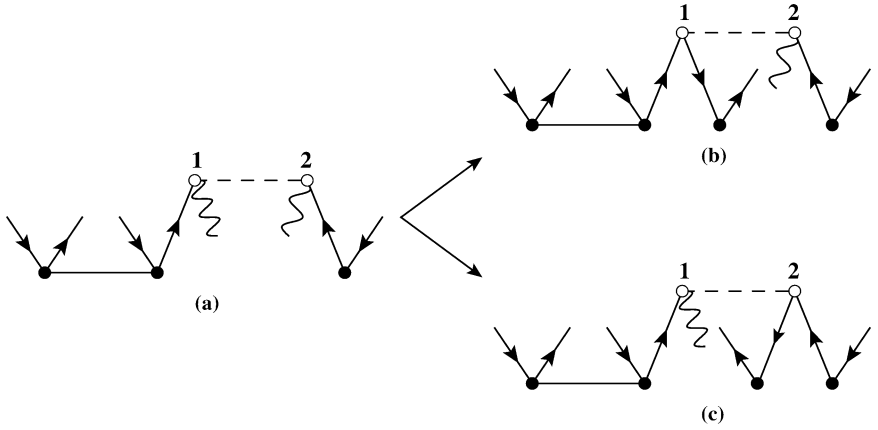


Figure 4. Two topologically distinct cubic terms (b) and (c) generated from the quadratic term (a) upon contraction with a t_1 tensor. Each t tensor has an identical contraction pattern in both (b) and (c). Therefore, they cannot be distinguished by comparing the contraction patterns. However, contractions at the two vertices numbered 1 and 2 of v are distinct in (b) and (c), which needs to be compared along with the contraction patterns.

algebraic expressions for the diagrams of Figure 3a−d, as obtained by applying Wick's theorem, are

$$2 \times \frac{1}{4} v_{h_1g_2}^{p_1g_4} t_2{}_{p_1p_2}^{h_1h_2} \hat{E}_{g_4h_2}^{g_2p_2},\ 2 \times \frac{1}{4} v_{h_2g_2}^{p_2g_4} t_2{}_{p_1p_2}^{h_1h_2} \hat{E}_{g_4h_1}^{g_2p_1},$$
$$2 \times \frac{1}{4} v_{g_1h_1}^{g_3p_1} t_2{}_{p_1p_2}^{h_1h_2} \hat{E}_{g_3h_2}^{g_1p_2},\ 2 \times \frac{1}{4} v_{g_1h_2}^{g_3p_2} t_2{}_{p_1p_2}^{h_1h_2} \hat{E}_{g_3h_1}^{g_1p_1} \quad (13)$$

where the factor 2 is the sign and loop factor for these terms. This factor is determined individually for each term in a *nonrecursive* fashion. The above four terms are equivalent under permutation of the contracted vertices on the two tensors. Our algorithm determines this equivalence in a straightforward manner by comparing the contraction patterns. For all four terms in eq 13, it recognizes that there is a loop contraction. Therefore, our algorithm identifies the terms as the diagrams of Figure 3. Thus, all four terms are found to be "*topologically equivalent*" irrespective of which two vertices on the two tensors are contracted. Then using a selection criterion, as stated below, only one term in eq 13 is picked up, and the others are deleted. The automorphic factor of this term is recomputed by summing up the automorphic factors of the four terms, which then becomes $1/4 \times 4 = 1$. Among the topologically equivalent terms, the one having the maximum number of particle contractions on the leftmost vertices of each tensor is selected. In the absence of particle contractions, other types of contractions are considered. Thus, for the terms in eq 13, the leftmost term is the preferred one. However, we should mention that practically any term among a set of equivalent terms may be chosen as the final form, as long as the code generator can find a proper permutation of indices required for evaluating the contractions by matrix multiplication. Thus, using our algorithm, eq 13 simplifies to

$$2 v_{h_1g_2}^{p_1g_4} t_2{}_{p_1p_2}^{h_1h_2} \hat{E}_{g_4h_2}^{g_2p_2} \quad (14)$$

On the basis of the contraction patterns, our algorithm recognizes the terms of Figure 3e−h as topologically distinct terms. Thus, the final expression will have all topologically distinct terms with correct coefficients. The above simplification of the equations is done when terms with all possible contractions have been generated, and not in each step of the recursion. This keeps the counting of the automorphic factors straightforward.

For the linear terms, comparing the contraction patterns suffices to identify all topologically distinct terms. The case of nonlinear terms is somewhat more involved, which arises *only in the case of non-antisymmetric Goldstone diagrams*. Nonlinear terms are constructed sequentially in our algorithm in the order linear, quadratic, cubic, etc. Every time a new tensor is brought

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

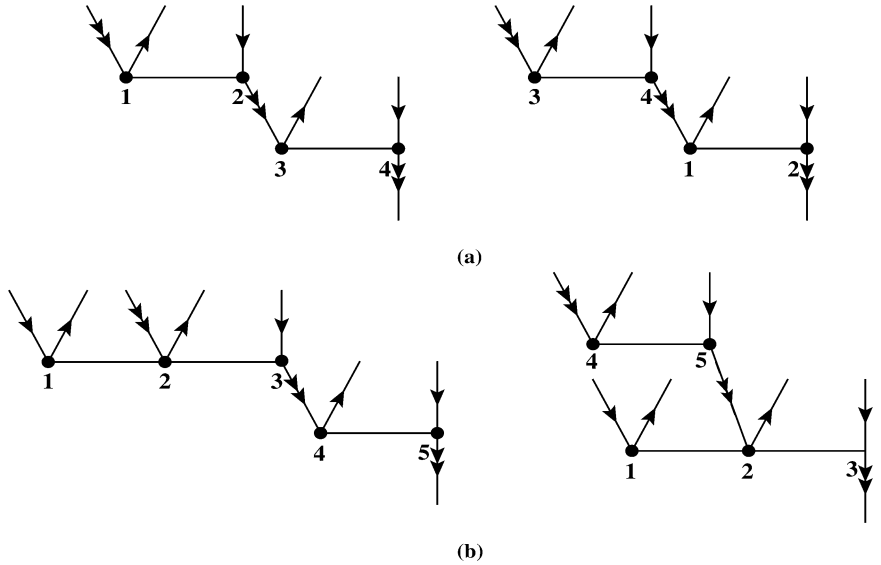


Figure 5. (a) Contracted composites of two $t_{2x}^{(1)}$ tensors in two permutations. Vertex connections for the right diagram are $t_{2x}^{(1)}$, [vertex-1 (−), vertex-2 ($t_{2x}^{(1)}$)] and $t_{2x}^{(1)}$, [vertex-3 ($t_{2x}^{(1)}$), vertex-4 (−)] and for the left diagram are $t_{2x}^{(1)}$, [vertex-3 (−), vertex-4 ($t_{2x}^{(1)}$)]; $t_{2x}^{(1)}$, [vertex-1 ($t_{2x}^{(1)}$), vertex-2 (−)]. Since both tensors are identical, viz., $t_{2x}^{(1)}$, the vertex connections become identical if the numberings are omitted, which is done just for the sake of convenience. Hence, the two contracted composites are *topologically equivalent*. (b) Contracted composites of $t_{2x}^{(1)}$ and $t_{3x}^{(1)}$ tensors in two permutations. In this case, the vertex connections of $t_{2x}^{(1)}$ and $t_{3x}^{(1)}$ are compared individually. Vertex connections for $t_{3x}^{(1)}$ on the right diagram are [vertex-1 (−), vertex-2 (−), vertex-3 ($t_{2x}^{(1)}$)], and on the left diagram are [vertex-1 (−), vertex-2 ($t_{2x}^{(1)}$), vertex-3 (−)]. Since the vertices-2 and 3 of $t_{3x}^{(1)}$ are nonequivalent, the vertex connections of $t_{3x}^{(1)}$ are not identical in the two permutations. The same is true also for $t_{2x}^{(1)}$. Therefore, the two contracted composites are *topologically inequivalent*.

in, it is contracted with an effective operator that has been formed at the previous level. For a term consisting of v and n t amplitudes, for example, the list of contraction patterns would consist of n integers, where each integer specifies the contraction pattern for each t amplitude. Let us emphasize that our algorithm works precisely in the same way if a new tensor is brought in either from the right or from left of a term (which may have one or more tensors in it).

Let us now consider the term depicted diagrammatically in Figure 4a. The wavy lines without arrows indicate general indices on v. The two vertices of v labeled 1 and 2 are inequivalent, since they are contracted to two different tensors, viz., t_1 and t_2. Therefore, the two cubic terms of Figure 4b and c, obtained by contracting another t_1 amplitude, through a hole index in both cases, are topologically distinct. This is, however, not reflected in the contraction patterns of the t tensors. In both terms, each t tensor has precisely the same contraction pattern. On the other hand, if we compare the vertex connections of v in these terms, we find the following:

Term-b: vertex-1 (t_1, t_2). vertex-2 (t_1).

Term-c: vertex-1 (t_2). vertex-2 (t_1, t_1).

which implies that in the term of Figure 4b, vertex-1 of v is connected to t_1 and t_2 amplitudes and vertex-2 is connected only to t_1. These two terms can therefore be distinguished by comparing the vertex connections of v. Thus, the information of vertex connections is additionally required in order to identify terms with distinct topologies. Since in the above example there are no contractions among the t tensors, it suffices to compare the vertex connections of v only. In the case of noncommuting operators, however, information about the vertex connections of all the tensors are needed. We should mention here that since the two vertices of v are symmetric under permutation when no tensor is contracted to it, our algorithm checks the similarity of vertex connections in both permutations of the two vertices. For example, the two terms obtained by replacing the t_2 by a t_1 tensor in Figure 4b and c (i.e., the vt_1^3 terms) would be identified as topologically equivalent. On the other hand, if a tensor has inequivalent vertices to begin with, e.g., the $t_{2e}^{(1)}$ amplitude of Figure 1d, only contractions on the equivalent vertices are compared.

The information about contraction patterns that each tensor possesses with the other tensors in a term and the information about vertex connections of all the tensors together constitute the necessary and sufficient criterion to completely specify its topology. These two sets of information are stored in the attribute "contraction topology" of the class "term."

The topology based algorithm has several advantages over the canonicalization scheme employed in TCE[16] and APG.[24] In the latter scheme, all terms are rewritten in a "canonical form"[31] by renumbering the contracted indices and sorting them in a unique order. The equivalent terms then become identical and thus can be readily summed up. The definition of the canonical form is arbitrary, and one needs to define a set of rules for this purpose (see, e.g., ref 16). In our algorithm, on the other hand, the contraction topologies of the terms are determined once they are generated by applying Wick's theorem and stored as an attribute of the class "term." The contraction topologies then serve as the "fingerprints" of the terms and are used in all subsequent steps, e.g., simplification of the equations and, as will be discussed in section 3.3, their factorization. In none of these steps does one need to go through the actual orbital indices on the tensors or the operator in a term. Instead, it suffices to compare the information contained in the contraction topologies, which being lists of integers, are easily comparable.

3.2. Derivation of the COSCC Ansatz and the Amplitude Equations. The determination of the automorphic factors in the COSCC ansatz is quite straightforward with

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

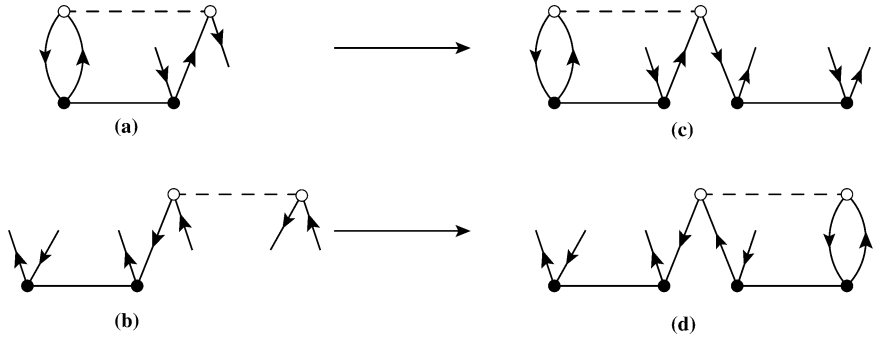


Figure 6. Two topologically distinct linear terms (a) and (b) generating two topologically equivalent quadratic terms (c) and (d), respectively. In (c) and (d), the orders of multiplications of the two t_2 tensors are different.

our expression generator. We adopt the following strategy. Given a set of n noncommuting operators, all $n!$ permutations are first generated. Wick's theorem is then applied to each permutation, and the contraction topologies of the resulting terms are compared. Each term in the resulting expression is weighted by the inverse of the number of its topologically equivalent forms generated in different permutations.

Let us illustrate the generation of some quadratic products in the COSCC ansatz. For a product consisting of $t_{1e}^{(1)}$ and $t_{2x}^{(1)}$ amplitudes, the above procedure obtains

$$t_{1e\,u_1}^{(1)\,i_1}\hat{E}_{i_1}^{\ u_1}\cdot t_{2x\,p_1u_1}^{(1)\,u_1i_2}\hat{E}_{u_1i_2}^{\ p_1u_1} \rightarrow t_{1e\,u_1}^{(1)\,i_1}\hat{E}_{i_1}^{\ u_1}\cdot t_{2x\,p_1u_1}^{(1)\,u_1i_2}\hat{E}_{u_1i_2}^{\ p_1u_1} + t_{2x\,p_1u_1}^{(1)\,u_1i_2}\hat{E}_{u_1i_2}^{\ p_1u_1}\cdot t_{1e\,u_1}^{(1)\,i_1}\hat{E}_{i_1}^{\ u_1} \tag{15}$$

$$\text{Wick's theorem} \rightarrow \frac{1}{2}\{t_{1e\,u_1}^{(1)\,i_1}t_{2x\,p_1u_1}^{(1)\,u_1i_2}\}\hat{E}_{i_1u_1i_2}^{\ u_1p_1u_1} - \{\overline{t_{1e\,u_1}^{(1)\,i_1}t_{2x\,p_1u_1}^{(1)\,u_1i_2}}\}\hat{E}_{i_1i_2}^{\ p_1u_1} + \frac{1}{2}\{t_{2x\,p_1u_1}^{(1)\,u_1i_2}t_{1e\,u_1}^{(1)\,i_1}\}\hat{E}_{u_1i_2i_1}^{\ p_1u_1u_1} \tag{16}$$

where the minus sign in the second term arises from the active index contraction between the $t_{1e}^{(1)}$ and $t_{2x}^{(1)}$ amplitudes, and the first and the third terms represent normal-ordered products without contractions. The first two terms in eq 16 are generated from the first permutation of eq 15 by application of Wick's theorem. The second permutation of eq 15 generates only the third term in eq 16 and does not generate any contracted product. Therefore, the factor associated with the contracted product is 1 instead of 1/2. Note that if the same procedure is followed for the commuting operators, then all $n!$ permutations of n operators would generate products with the same contraction topology (viz., no contraction), and therefore the automorphic factor of each product would be $1/n!$.

For the product of a pair of $t_{2x}^{(1)}$ amplitudes, the final expression is

$$\frac{1}{2}\{t_{2x\,p_1u_1}^{(1)\,u_1i_1}t_{2x\,p_2u_1}^{(1)\,u_1i_2}\}\hat{E}_{u_1i_1u_1i_2}^{\ p_1u_1p_2u_1} - \frac{1}{2}\{\overline{t_{2x\,p_1u_1}^{(1)\,u_1i_1}t_{2x\,p_2u_1}^{(1)\,u_1i_2}}\}\hat{E}_{u_1i_1i_2}^{\ p_1p_2u_1} + \frac{1}{2}\{t_{2x\,p_2u_1}^{(1)\,u_1i_2}t_{2x\,p_1u_1}^{(1)\,u_1i_1}\}\hat{E}_{u_1i_2u_1i_1}^{\ p_2u_1p_1u_1} - \frac{1}{2}\{\overline{t_{2x\,p_2u_1}^{(1)\,u_1i_2}t_{2x\,p_1u_1}^{(1)\,u_1i_1}}\}\hat{E}_{u_1i_2i_1}^{\ p_2p_1u_1} \tag{17}$$

In this case, contracted products are generated in both permutations, and they both have the same contraction topology, or more explicitly, the same contraction pattern, viz., active contraction, and the same vertex connections for each $t_{2x}^{(1)}$ tensor, as illustrated in Figure 5a. Thus, each contracted product gets a factor of 1/2. Let us consider a more complicated case: the product of $t_{2x}^{(1)}$ and $t_{3x}^{(1)}$ amplitudes. The latter is the three-body exchange spectator and would appear in the COS-CCSDT ansatz. The contracted composites in the two permutations of these amplitudes are shown in Figure 5b. In both permutations, the contraction patterns are the same. But if we compare the vertex connections of the $t_{2x}^{(1)}$ amplitude in the two composites, we see that the same type of contraction (viz., active contraction) appears on two different vertices. Since, by definition, the two vertices of $t_{2x}^{(1)}$ are nonsymmetric under permutation, the same type of contraction on the two vertices leads to different contraction topologies. The same is true also for $t_{3x}^{(1)}$: the vertices labeled 1 and 2 are equivalent, but neither is equivalent to vertex-3. Therefore, the two contracted composites of Figure 5b are found to be *topologically distinct*, and each composite gets a factor of 1.

The above discussion reveals the generality of our topology based algorithm. The same definition of contraction topology that works to generate all the topologically distinct terms in the closed-shell CC equations using the commuting cluster operator leads also to the correct determination of the automorphic factors in the COSCC ansatz with noncommuting excitations. In other words, *our definition of contraction topology applies generally to non-antisymmetric Goldstone diagrams*. The reason for first generating the terms with all possible contractions using Wick's theorem, instead of directly generating the topologically distinct terms as in ref 15, is also evident. For the COSCC ansatz, the determination of the automorphic factors requires one to generate each composite of excitation operators with a specific contraction topology in all possible permutations of the operators.

For the derivation of the COSCC amplitude equations, we first generate the products of two, three, and four cluster amplitudes in all permutations with the topology-dependent automorphic factors as discussed above. These products are then contracted either to the right of the Hamiltonian to construct the direct term in eq 10 or from the left of the $\hat{H}_{\text{eff}}$ operator to generate the folded terms. This implies that the nonlinear terms are not generated sequentially, unlike in the case of CC equations using the commuting cluster amplitudes. This nonsequential algorithm is necessary for COSCC, since the automorphic factors for the products of noncommuting excitation operators are not known beforehand. The use of a sequential algorithm instead would require these automorphic factors to be determined directly from the COSCC equations, which is not straightforward, and is not recommended either, because the definition of the COSCC ansatz should be independent of the structure of the amplitude equations.

The strongly, weakly, and h-connected amplitudes in a term can be easily distinguished by tracing the vertex connections of

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

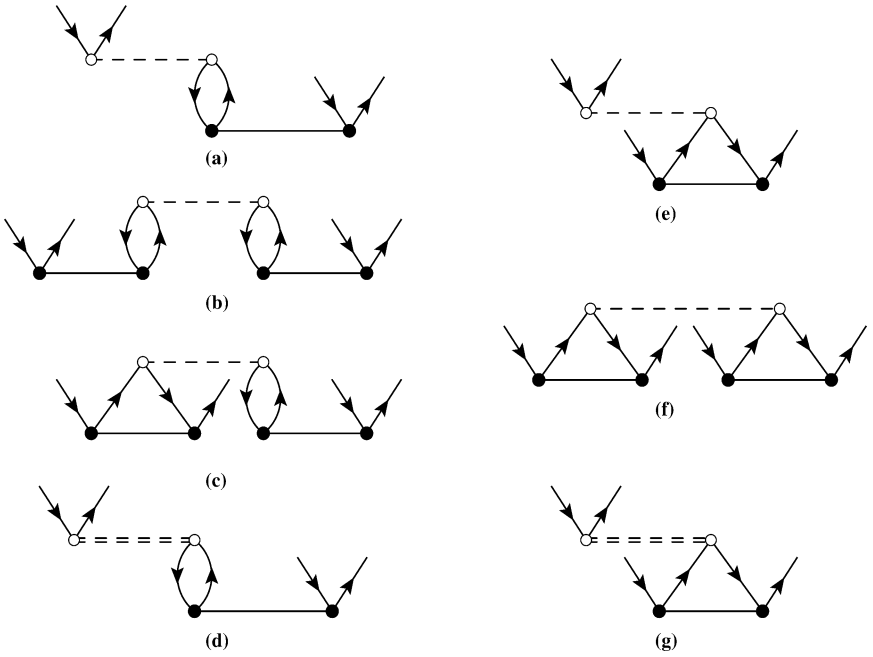


Figure 7. Some intermediates involved in the CCSD doubles amplitude equations. The terms (a)−(c) are constructed as the quasilinear term (d) using the intermediate, whose vertices are indicated by double broken lines. The terms (e) and (f) are constructed using the intermediate in (g). These two intermediates are identical, but they consist of different sets of diagrams and therefore cannot be reused to construct all terms.

the amplitudes and the types of indices through which they are contracted (hole, particle, or active) to the other tensors. For example, a strongly connected term is the one in which each cluster amplitude is joined to the f or v tensor through at least one inactive hole or particle index. For implementation purposes, it suffices to select only the strongly connected terms (up to quartic) that contribute to the final, simplified COSCC amplitude equations. However, we emphasize that a simplification of the equations, which was discussed in great detail in ref 39, can actually be achieved using our automated expression generator.

3.3. Factorization of Equations. Any general term in the CC equations is numerically evaluated in a sequence of binary tensor multiplications, i.e., by defining suitable intermediates. The efficiency of a CC program depends strongly on the choice of the proper sequences. To determine the cost-minimal sequence, the so-called "strength reduction,"[16] one needs to consider two important aspects: (a) computational costs and (b) memory requirements for storing the intermediates.

For commuting operators, we adopt the following strategy. Since the nonlinear terms are constructed sequentially in the order linear, quadratic, etc., at every step of this sequence, topologically equivalent nonlinear terms are generated by Wick's theorem. Let us consider, for example, the two linear terms of Figure 6a and b. These are topologically distinct, and hence they are both generated with correct coefficients. When a second t_2 is contracted, the terms of Figure 6c and d are generated from a and b, respectively. These two terms are topologically equivalent but have different orders of multiplication of the two t_2 tensors. While choosing the final form for this term, our algorithm compares the computational costs of the two linear intermediates of Figure 6a and b and picks up the term c, since it involves minimal cost. Thus, the order of multiplication of the tensors in each term gets optimized at every step of the sequence of generating the nonlinear terms. Typically, for a nonlinear term, e.g., $vt_at_bt_c$, our algorithm stores it in a form, say, $vt_bt_at_c$, in which the number of contracted indices between the v and t tensors decreases from the leftmost t to the rightmost. This term is then constructed in the following sequence

$$\mathcal{W}_1 = v \otimes t_b;\ \mathcal{W}_2 = \mathcal{W}_1 \otimes t_a;\ vt_bt_at_c = \mathcal{W}_2 \otimes t_c \qquad (18)$$

with $\otimes$ indicating a binary tensor multiplication. Among the types of contracted indices, the particle indices are given the maximum weight factor, such that the number of uncontracted particle indices in the intermediates involved is as low as possible. This optimizes their storage cost at the same time. For the COSCC equations, since the nonlinear terms are not generated sequentially, we employ an alternative algorithm of comparing the computational costs in all possible permutations of the cluster amplitudes.

Further factorization of the equations may be achieved by identifying common intermediates to construct a number of terms. This reduces the number of binary tensor multiplications. We greatly benefit from our contraction topology based algorithm to achieve this. We employ the following recursive factorization algorithm, which is similar in spirit to the algorithm of Kállay and Surján.[15]

All terms in the CC amplitude equations are stored in a form in which the leftmost tensor is always a Hamiltonian component, viz., f or v. Given an equation, the algorithm compares the contraction topologies of the terms present therein and determines in which terms the rightmost t tensor has the same contraction pattern (indicated by the same integer), and the composites obtained by deleting this t tensor are identical. The entire equation is thus split into several subgroups of terms. A quasilinear term is then defined for each subgroup as a binary tensor product of an intermediate ($\mathcal{W}$) and the rightmost t tensor, such that the equation becomes

$$R = \sum_{i=1}^{n_g} \mathcal{W}_i \otimes t_i \qquad (19)$$

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

where n_g denotes the number of subgroups, and the residual R corresponds to a specific CC amplitude equation. The intermediates W_i may consist of a single term or a group of terms, i.e., a subexpression. In the latter case, the same procedure is repeated with the subexpression to find out further factorization possibilities. Thus, each subgroup i in eq 19 is subject to a recursive factorization procedure, which continues until all intermediates reduce to single terms, which means no further factorization is possible. The algebraic expressions for all the intermediates are not required to be stored. As soon as the number of terms in an intermediate reduces to one, the expression for the corresponding quasilinear term is converted to a Fortran 90 subroutine call using the code generator. While determining the common intermediates in this way, the order of multiplication of the tensors in each term, which was optimized in the previous step, is not altered.

Another factorization scheme, which defines effective amplitudes by combining a set of cluster amplitudes, e.g.,

$$\begin{aligned} \tau_{2\,p_1p_2}^{h_1h_2} &= t_{2\,p_1p_2}^{h_1h_2} + t_{1\,p_1}^{h_1} t_{1\,p_2}^{h_2} \\ \tau'_{2\,p_1p_2}^{h_1h_2} &= 2t_{2\,p_1p_2}^{h_1h_2} - t_{2\,p_1p_2}^{h_2h_1} \end{aligned} \tag{20}$$

$$\sigma_{2p_1u_1}^{(1)i_1i_2} = t_{2e\,p_1u_1}^{(1)i_1i_2} - \{\overline{t_{1e\,u_1}^{(1)i_1} t_{2x\,p_1u_1}^{(1)u_1i_2}}\} \tag{21}$$

is also employed in our implementation.

The above schemes, however, do not produce the most efficiently factorized equations. For the non-antisymmetric Goldstone diagrams, the recursive factorization is somewhat less effective compared to a similar factorization of the antisymmetrized diagrams as described in ref 15. In the Appendix, we provide explicit expressions for the singles and doubles amplitude equations for closed-shell CCSD along with the definitions of the intermediates. For CCSD, the term involving the two-electron integrals with four virtual indices is computationally the most expensive term. This term is evaluated as a quasilinear term using the τ_2 intermediate of eq 20 and has the correct computational scaling, viz., $n_o^2 n_v^4$. The relative inefficiency in the factorization arises from the fact that the intermediates with two hole and two virtual indices, viz., $\mathcal{W}1$, $\mathcal{W}2$, $\mathcal{X}1$, $\mathcal{X}2$, $\mathcal{Y}1$, and $\mathcal{Y}2$ etc. defined in the Appendix, are different, because they either consist of different sets of terms or the same terms have different coefficients. One example of this situation is illustrated in Figure 7. This is not the case for an algorithm that uses antisymmetrized Goldstone diagrams, e.g., the factorized CCSD amplitude equations presented in ref 8. The reuse of intermediates is therefore somewhat restricted.

For an automated factorization scheme, one needs to specify certain criteria to choose the optimal order of multiplication of the tensors in each term, and also to determine how the common subexpressions are eliminated. In the current scheme, the search for common intermediates is limited within the optimally ordered products, which are not altered once determined. The equations in the Appendix indicate that intermediates with one hole and three virtual indices are required within the current factorization scheme, which may become quite expensive as the size of the basis set increases. It may be possible to choose a different set of criteria to optimize the order of multiplication of the tensors in each term, such that the use of this particular intermediate may be reduced. A more effective factorization may be achieved by producing all possible permutations in the algorithm and simultaneously identifying the common intermediates in various permutations, in the spirit of the genetic search algorithm recently proposed by Hanrath and Engels-Putzka.[18] In our opinion, the current factorization schemes lead to reasonably efficient codes, although there are possibilities for further improvements.

3.4. Synthesis of Fortran Codes. The final step in the automated implementation is the conversion of the Python expressions to Fortran 90 subroutines by the code generator. The code generator is a Python script that converts the algebraic expression for each binary tensor product to a Fortran 90 subroutine call by determining the index permutations on the two tensors involved in the binary product from the contraction topology.

The tensors are stored as matrices in our codes. No restriction on the indices is used, which implies that the t_2 tensor is stored as a matrix $\mathbb{T}_2(n_v^2,n_o^2)$ with all pairs of hole and particle indices. This type of storage is necessary for matrix multiplications, but as indicated in ref 15, the storage requirements rapidly increase with the excitation rank of the t tensors. As already emphasized, our aim is to implement rigorously spin-adapted CC theories within the standard singles−doubles approximation, or possibly with triple excitations, for which the current storage of matrices is a reasonable working strategy. Second, we do not exploit point-group symmetry in our current code to reduce the dimensions of the matrices. We have developed a series of generalized tensor contraction subroutines which perform tensor contractions by matrix−matrix or matrix−vector multiplications using the basic linear algebra subprograms (BLAS)[47] DGEMM and DGEMV, respectively. The tensor contraction subroutines are general in the sense that they can perform arbitrary permutations of the indices, which are required to align the summation indices on the input tensors for matrix multiplication, and also that they work with all types of orbital indices (hole, particle, and active).

The intermediate matrices in our codes are defined as temporary arrays with dynamic memory allocation and are stored only until they are required. The two-electron integrals involving four virtual indices are not stored in the main memory. The tensor contractions involving these integrals are performed as matrix−vector multiplications by reading one row of the integral matrix at a time from disc into the memory. The generated codes are implemented into a new program, which requires as input the one- and two-electron integrals in the spatial orbital basis. For the calculations presented in this paper, the integrals were generated using the GAMESS-US program package.[48]

4. REPRESENTATIVE CALCULATIONS

4.1. Implementation of the COSCC Amplitude Equations for the Doublet Case. Within the factorized form of eq 12, the t_1 and t_2 amplitude equations for the one active orbital (doublet) case have the following forms:[38]

$$\begin{aligned} \bar{h}_{p_1}^{h_1} - \bar{h}_{p_1u_1}^{h_1u_1} - g_{p_1u_1}^{h_1u_1} &= 0 \\ \bar{h}_{p_1p_2}^{h_1h_2} - \bar{h}_{p_1p_2u_1}^{h_1h_2u_1} - g_{p_1p_2u_1}^{h_1h_2u_1} &= 0 \end{aligned} \tag{22}$$

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

while the equations determining the $t^{(1)}$ amplitudes are

$$\begin{aligned} \bar{h}^{i_1}_{u_1} + g^{i_1}_{u_1} - t^{(1)i_1}_{1e\,u_1}(\bar{h}^{u_1}_{u_1} + g^{u_1}_{u_1}) &= 0 \\ \bar{h}^{i_1 i_2}_{p_1 u_1} + g^{i_1 i_2}_{p_1 u_1} - t^{(1)i_1 i_2}_{2e\,p_1 u_1}(\bar{h}^{u_1}_{u_1} + g^{u_1}_{u_1}) &= 0 \\ \bar{h}^{u_1 i_1}_{p_1 u_1} + g^{u_1 i_1}_{p_1 u_1} &= 0 \end{aligned} \tag{23}$$

In the above equations, $\bar{h}$ and g denote the amplitudes of the $\hat{\bar{H}}$ and $\{\{\hat{\bar{H}}\ \exp\ (\hat{T}^{(1)})\}\}_s$ operators, respectively. In every iteration, the $\bar{h}^{h_1}_{p_1}$ and $\bar{h}^{h_1 h_2}_{p_1 p_2}$ amplitudes of eq 22 and the other one- and two-body $\bar{h}$ amplitudes are constructed using the t amplitudes known from the previous iteration, and they carry the general hole and particle indices. The corresponding matrices are then divided into submatrices which are labeled explicitly by the inactive holes and the active label(s). Since the g amplitudes consist only of the strongly connected terms with each $t^{(1)}$ amplitude in them being contracted to the $\bar{h}$ tensor through its inactive hole and particle indices, the $\mathcal{W}$ intermediates generated in an optimized sequence of binary tensor multiplications similar to eq 18 are always labeled by one or more active indices. Since the number of active or open-shell orbitals (n_s) is much smaller than the number of inactive holes (n_h), e.g., $n_s = 1$ for the doublet case, the dimensions of the corresponding intermediate matrices are rather small, leading to low storage requirements and faster multiplication compared to the intermediate matrices involved in the evaluation of the $\bar{h}^{h_1}_{p_1}$ and $\bar{h}^{h_1 h_2}_{p_1 p_2}$ amplitudes. Thus, despite having terms with $t^{(1)} - t^{(1)}$ contractions contributing to the g amplitudes, the numerical evaluation of these terms is quite efficient. Use of the $\sigma^{(1)}_2$ intermediates of eq 21 effectively reduces the number of operations, which also adds to the efficiency.

The hole−particle excitation amplitudes $\bar{h}^{h_1}_{p_1}$ and $\bar{h}^{h_1 h_2}_{p_1 p_2}$ of eq 22 have the same computational scaling as closed-shell CCSD, viz., $n_o^2 n_v^4$ with $n_o = n_h + n_s$. The leading computational cost for the terms involved in the three-body amplitudes $\bar{h}^{h_1 h_2 u_1}_{p_1 p_2 u_1}$ and $g^{h_1 h_2 u_1}_{p_1 p_2 u_1}$ is $n_o^2 n_v^3$ apart from a prefactor of $n_s = 1$. Similarly, for the terms contributing to the two-body g amplitudes of eq 23, the computational cost never exceeds $n_o^2 n_v^3$ (apart from the same prefactor) with our scheme of optimizing the order of tensor multiplications in each term. We gain advantage from the fact that the active orbital u_1 is of the hole type in our formulation, so that there is at most one virtual index in the definition of $\hat{T}^{(1)}$. For the $g^{h_1 h_2 u_1}_{p_1 p_2 u_1}$ amplitudes, contributions from terms up to quadratic in the one-valence cluster amplitudes are included. Therefore, in terms of the occupied (both singly and doubly) and virtual orbitals, the scaling of the computational cost for COS-CCSD for the one active orbital case (doublets) is $\mathcal{N}^6$, with the leading computational cost being $n_o^2 n_v^4$, which is the same as CCSD.

4.2. Numerical Examples. Energy calculations have been performed using our COSCC codes on the ground states of a number of doublet radicals and cations in order to assess the efficiency of our codes. We provide a thorough analysis of the CPU time required for a single COSCC amplitude iteration and the fractions of iteration time involved in computing the various terms involved in eqs 22 and 23. To assess the relative CPU time per iteration compared to the spin−orbital based CCSD, ROHF-CCSD calculations were carried out on the same doublet systems using the CFOUR program package.[49] All calculations were performed using Dunning's cc-pVTZ basis[50] on an Intel Xeon CPU having a processor speed of 2.93 GHz. The radicals/cations studied in this paper include the CH_2N radical, the allyl radical (C_3H_5), the radical cation of *trans*-1,3-butadiene ($C_4H_6^{\bullet+}$), the cation of pyrrole ($C_4H_4NH^+$), and the adducts of the cytosine molecule with H and OH radicals. The last two systems are of genuine chemical and biological interest as they are known to be involved in the degradation of DNA.[51,52] These adducts have been studied previously in the literature using multiconfigurational self-consistent field and density-functional theories,[53] and also recently by equation-of-motion CC calculations.[54]

All electrons were correlated in all calculations except those for the two adducts of cytosine, for which the 1s electrons on C, N, and O atoms were kept frozen. The ROHF orbitals optimized for the radicals/cations were employed. Since point-group symmetry is not used in the current implementation of COSCC, the ROHF-CCSD calculations were performed using the C_1 point group for an unbiased comparison of the iteration time. The two cytosine adducts are, however, nonsymmetric anyway. We optimized the geometries of these two adducts using second-order Moller−Plesset perturbation theory based on the unrestricted HF reference (UHF-MP2) and the POL1 basis set[55] taking the geometries reported in ref 54 as the starting point. No electrons were kept frozen in the geometry optimization, similar to ref 54. The optimized structures of these two adducts are shown in Figure 8.

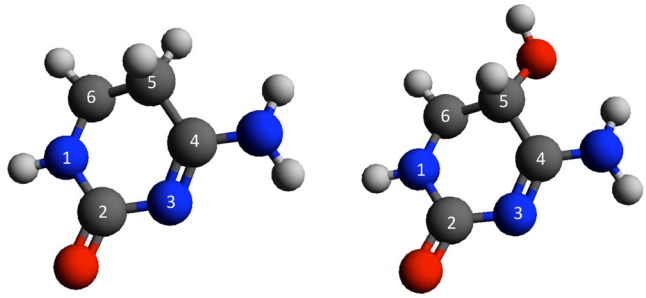


Figure 8. Optimized geometries for the H and OH adducts of the cytosine molecule obtained at the UHF-MP2 level of theory using the POL1 basis set. In both cases, the odd electron is localized at the carbon atom numbered 6.

Let us first compare the iteration times for the current closed-shell CCSD code with the CCSD code implemented in CFOUR. We compare the iteration times for two closed-shell molecules, viz., *trans*-1,3-butadiene and pyrrole. The closed-shell CCSD residuals consist only of the $\bar{h}^{h_1}_{p_1}$ and $\bar{h}^{h_1 h_2}_{p_1 p_2}$ amplitudes of eq 22. The iteration times are compared in Table 1, which indicates that the closed-shell CCSD code is ~3.7 times slower than the implementation in CFOUR, which

Table 1. Comparison of the Iteration Times of the Closed-Shell CCSD Codes

				CPU time/iteration (min.)	
molecule	n_{corr}[a]	n_{MO}[b]	energy (hartree)	CFOUR	this work
trans-C_4H_6	30	204	−155.727093	3.14	11.49
C_4H_4NH	36	220	−209.827128	4.95	18.55

[a]Number of correlated electrons. [b]Number of correlated molecular orbitals in cc-pVTZ basis.

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Table 2. Comparison of Energies and Iteration Times of ROHF-CCSD and COS-CCSD for Various Doublet Radicals in Their Ground States Using the cc-pVTZ Basis

			energy (hartree)		CPU time/iteration (min.)	
radical/cation	n_{corr} [a]	n_{MO} [b]	ROHF- CCSD[c]	COS-CCSD	ROHF- CCSD[c]	COS-CCSD
CH_2N	15	88	−93.835872	−93.836255	0.23	0.19
C_3H_5	23	160	−117.057459	−117.062025	2.32	3.91
trans-$C_4H_6^{\bullet+}$	29	204	−155.400165	−155.408854	5.46	14.91
$C_4H_4NH^+$	35	220	−209.533382	−209.540271	11.43	26.65
cytosine(5)H	43	316	−394.775756	−394.776605	86.33	141.11
cytosine(5)OH	49	345	−469.890081	−469.890822	142.59	259.11

[a]Number of correlated electrons. [b]Number of correlated molecular orbitals in cc-pVTZ basis. [c]The ROHF-CCSD calculations were performed using the CFOUR program package.[49]

is based on the factorized equations of ref 8. As explained in section 3.3, the computational scaling of the current CCSD code is $n_o^2 n_v^4$, which is the same as the implementation in CFOUR. It is the grouping of terms to identify the common intermediates which is somewhat less efficient, such that intermediates involving three virtual indices are important as well. Further improvements of the factorization schemes will be tested in the near future.

In Table 2, we compare the CPU times per iteration for the COSCC implementation discussed in section 4.1 and the spin−orbital based ROHF-CCSD code implemented in CFOUR. The $\bar{h}_{p_1}^{h_1}$ and $\bar{h}_{p_1p_2}^{h_1h_2}$ amplitudes of eq 22 are the same as the singles and doubles amplitude equations in the closed-shell CCSD case. There are additional terms in the COSCC amplitude equations involving the $t^{(1)}$ amplitudes, which include terms with $t^{(1)} - t^{(1)}$ contractions. In the ROHF-CCSD calculation, the number of independent amplitudes increases roughly by three times. Therefore, despite using the same factorization scheme as in the closed-shell code, the CPU time per iteration increases. Table 2 indicates that the iteration times of the COSCC and ROHF-CCSD codes are somewhat closer compared to the closed-shell codes, with the former being approximately 2 times slower than the latter. It is also important to note that the COSCC energies are slightly lower than the ROHF-CCSD energies. This observation precisely parallels the trend of the results reported in ref 39, where the differences were attributed to the inclusion of implicit triple excitations through the terms with $t^{(1)} - t^{(1)}$ contractions.

The additional complexity of the COSCC equations compared to the standard formulation of CC theory, both closed-shell CC theory as well as the spin−orbital based open-shell CC theory, arises from the terms involving $t^{(1)} - t^{(1)}$ contractions. It is therefore very important to analyze how much time in every iteration is spent to compute such terms. In Table 3, we compare the fractions of time (in %) spent in the three major steps involved in a COSCC amplitude iteration: (a) computation of the $\bar{h}_{p_1}^{h_1}$ and $\bar{h}_{p_1p_2}^{h_1h_2}$ amplitudes of eq 22 (τ_{ccsd}), (b) computation of the $\bar{h}$ intermediates contributing to the g amplitudes of eqs 22 and 23 ($\tau_{\bar{h}}$) and the sorting of the $\bar{h}$ amplitudes to various submatrices carrying explicit inactive and active labels, and (c) computation of the g amplitudes (up to three-body) which involve terms with up to quartic power of $t^{(1)}$ amplitudes (τ_g), where the nonlinear terms necessarily involve $t^{(1)} - t^{(1)}$ contractions. A graphical representation of these data is presented in Figure 9.

It is evident from both Table 3 and Figure 9 that the terms with no contractions among the $t^{(0)}$ amplitudes are the most

Table 3. Analysis of the Fractions of CPU Time per Iteration (in %) Required to Construct the Various Terms in the COS-CCSD Amplitude Equations

radical/cation	CPU time/iteration (min)	τ_{ccsd} [a]	$\tau_{\bar{h}}$ [b]	τ_g [c]
CH_2N	0.19	58.7	20.7	19.6
C_3H_5	3.91	71.2	14.1	14.0
trans-$C_4H_6^{\bullet+}$	14.91	76.1	12.4	11.0
$C_4H_4NH^+$	26.65	76.6	12.6	10.4
cytosine(5)H	141.11	77.7	12.3	9.7
cytosine(5)OH	259.11	78.8	11.7	9.3

[a]Time required to construct the $\bar{h}_{p_1}^{h_1}$ and $\bar{h}_{p_1p_2}^{h_1h_2}$ amplitudes of eq 22. [b]Time required to construct the $\bar{h}$ intermediates contributing to the g amplitudes. [c]Total time required to construct all g amplitudes contributing to eqs 22 and 23.

time-consuming ones. Of less importance is step b, which is much faster than step a. Most interestingly, step c, which embodies all the additional complexities of the COSCC ansatz, requires the least amount of CPU time in the current implementation. As already mentioned in section 4.1, this simplicity emerges from the fact that the COSCC amplitude equations consist only of the strongly connected terms, such that each cluster amplitude in a nonlinear term is contracted to the Hamiltonian vertices through the inactive indices, thus leaving the active indices as the external ones in the intermediates that are formed at various stages of constructing a nonlinear term. The use of submatrices in the current implementation labeled explicitly by the active and inactive indices has the advantage that the intermediate matrices labeled by one or more active labels have much smaller dimensions compared to the intermediate matrices involved in step a, which allows quicker matrix multiplications. Table 3 also indicates that with an increasing number of correlated electrons and basis size, step a becomes more time-consuming, while the computational time for step c diminishes. For the two largest calculations considered in this paper, viz., the H and OH adducts of cytosine, step c is found to require less than 10% of the total iteration time.

The above analysis leads to a rather promising picture about a production level implementation of the COSCC ansatz. There are two issues pertaining to the COSCC approach, which might be considered to be difficult and somewhat undesirable: (a) the derivation and simplification of the COSCC equations involving contractions among the $t^{(1)}$ amplitudes, and the determination of the automorphic factors, and (b) the numerical evaluation of the terms involving $t^{(1)} - t^{(1)}$ contractions. The automated expression generator developed in the present work overcomes the difficulty of

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

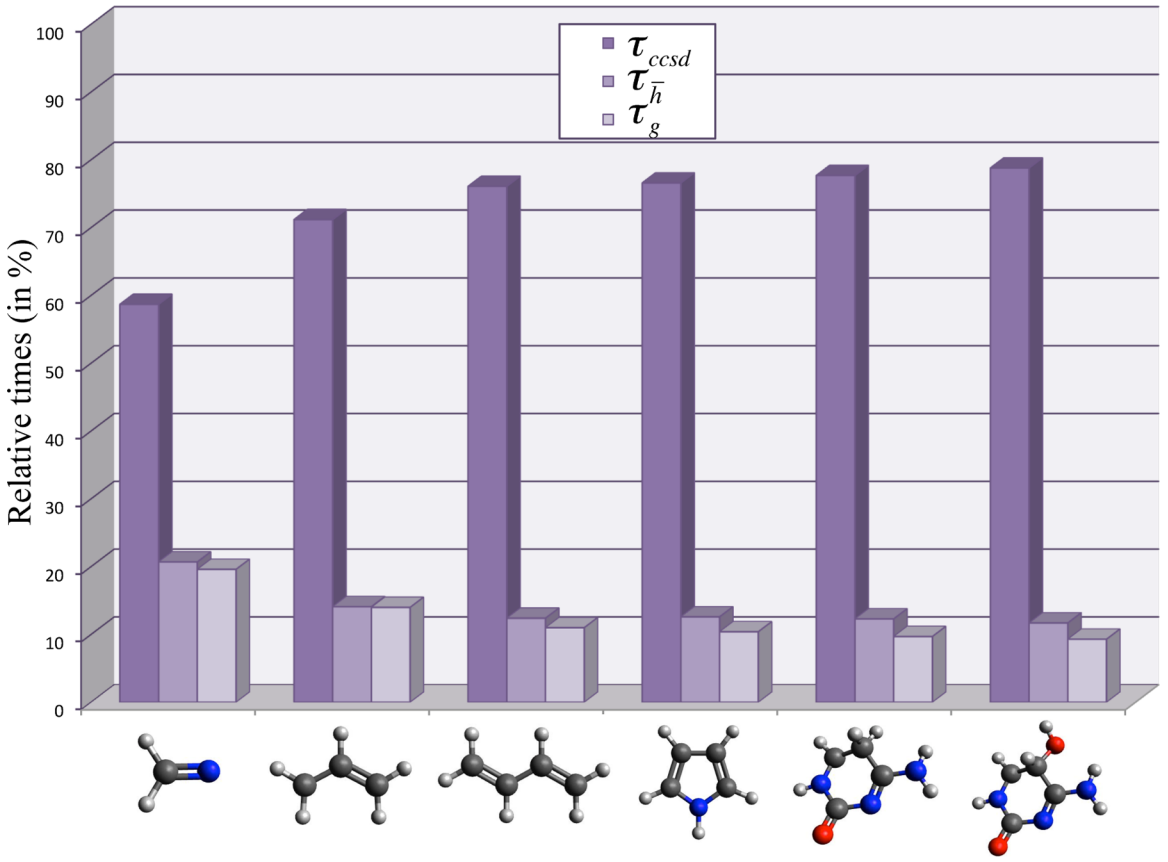


Figure 9. Bar plots indicating the fractions of CPU time per iteration (in %) required to construct the various terms in the COS-CCSD amplitude equations for various radicals and cations.

derivation very efficiently, and also the current implementation of COSCC equations clearly indicates that the numerical evaluation of the terms with $t^{(1)}$–$t^{(1)}$ contractions is the least time-consuming step. The presence of the latter terms in the COSCC equations is therefore an issue of no specific concern. The above arguments will hold equally well for the implementation of the COSCC ansatz for two-valence cases, e.g., for triplet and open-shell singlet states.

The relative iteration times for COSCC and ROHF-CCSD indicate that the additional computational cost that one pays to achieve rigorous spin-adaptation is not prohibitively high. The efficiency of the current COSCC code may be improved further by adopting alternative factorization schemes for the computation of $\bar{h}_{p_1}^{h_1}$ and $\bar{h}_{p_1p_2}^{h_1h_2}$ amplitudes, i.e., by improving the efficiency of the closed-shell CCSD code. Although in our opinion, the computation of the g amplitudes of eqs 22 and 23 is already quite efficient. It was emphasized in refs 38−40 that the $t^{(1)} - t^{(1)}$ contractions in the COSCC ansatz mimic the products and powers of cluster amplitudes as in an exponential ansatz with commuting excitations in the cluster operator. These contractions are important for including all orders of correlation and orbital relaxation effects, which are naturally present in the exponential ansatz of the spin−orbital based CC theory. The current implementation of COSCC for the doublet states and our applications to radicals and cations of real chemical interest clearly indicate that rigorous spin-adaptation in CC theory using the unitary group parametrization, while including precisely the same physical effects in the ansatz as in the spin−orbital based CC ansatz and yet generating a set of working equations that are at most quartic in cluster amplitudes akin to closed-shell CCSD, can in practice be achieved in a rather efficient way.

5. CONCLUSIONS

In this paper, we have proposed a general algorithm for the automated derivation and code synthesis for the spin-adapted CC approaches using a unitary group parametrization. We have reported a contraction topology based algorithm that applies generally to the non-antisymmetric Goldstone diagrams, which appear in the unitary group based CC formulations. We have illustrated that our algorithm generates CC amplitude equations with all topologically distinct terms having the correct coefficients, for both commuting as well as non-commuting excitations in the cluster operator.

We have proposed a general algorithm for counting the factors associated with the powers of the cluster operator, e.g., $\hat{T}^n$. All $n!$ permutations of the excitations in $\hat{T}^n$ are generated first. Then, Wick's theorem is applied to each permutation, and finally the contraction topologies of the generated terms are compared. If the excitations in $\hat{T}$ are commuting, this algorithm obtains the factor $1/n!$ as in an exponential, whereas for the noncommuting excitations, the same procedure yields the automorphic factor f_n of the COSCC ansatz. This implies that the automorphic factors of the COSCC ansatz may be deduced by employing the usual rules that specify the contraction topologies of the non-antisymmetric Goldstone diagrams.

The automated implementation tool developed by us effectively overcomes the potential difficulties of the COSCC theory, viz., understanding the structure of the equations, their derivation, and synthesis of efficient computer codes. The $t^{(1)} - t^{(1)}$ contractions

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

in the COSCC ansatz allow all orders of orbital relaxation and correlation effects to be included precisely as in the exponential ansatz of the closed-shell CC theory or the spin−orbital based open-shell CC theory. The availability of the implementation tool strengthens the reason for further investigating its usefulness for general open-shell systems and in real chemical applications. Implementations of the COSCC equations for triplets and open-shell singlets are underway and will be published in the near future. It is worthwhile to mention in this context that the property of having noncommuting excitations in the cluster operator is shared also by the internally contracted multireference CC ansatz.[25,56,57] A contraction topology based algorithm as proposed in this paper may become useful in this context as well.

The numerical applications to assess the efficiency of our COSCC codes clearly indicate that despite having terms with $t^{(1)} - t^{(1)}$ contractions, it is possible to implement the COSCC equations in a rather efficient way, such that applications to real chemical problems are feasible. In our current implementation, these terms require the least amount of relative time in a COSCC amplitude iteration. Furthermore, our applications indicate that the iteration time of COSCC differs from that of ROHF-CCSD by a factor of 2, which can be considered well acceptable. The computational cost for both methods scales as $\mathcal{N}^6$. The relative inefficiency of the COSCC code stems from (a) the presence of a larger number of terms owing to the use of non-antisymmetric Goldstone diagrams and (b) the comparatively less efficient factorization of the equations involving the commuting t_1 and t_2 amplitudes, which limits the use of common intermediates to some extent. While issue b may be improved further, which will be attempted in the near future, issue a is an inherent property of the unitary group adapted formalisms. Nevertheless, we are convinced that improving the factorization of the terms will reduce this difference even further, and issue a will not cause any significant difficulty in the exploration of the COSCC approach to chemical problems.

APPENDIX

Explicit Expressions for the Factorized Singles and Doubles Amplitude Equations for Closed-Shell CCSD in the Case of a Non-HF Reference

The factorized equation for the t_2 amplitudes using our current factorizations schemes is

$$\begin{aligned} t_2{}^{h_1h_2}_{p_1p_2} D_2 = {} & \frac{1}{2} v^{h_1h_2}_{p_1p_2} + \mathcal{F}^{h_1}_{h_3} t_2{}^{h_3h_2}_{p_1p_2} + \mathcal{F}^{p_3}_{p_1} t_2{}^{h_1h_2}_{p_3p_2} + \mathcal{W}^{h_1h_2}_{p_1h_3} t_1{}^{h_3}_{p_2} \\ & + \mathcal{W}^{h_1p_3}_{p_1p_2} t_1{}^{h_2}_{p_3} + \mathcal{W}^{h_1h_2}_{h_3h_4} t_2{}^{h_3h_4}_{p_1p_2} + \frac{1}{2} v^{p_3p_4}_{p_1p_2} \tau_2{}^{h_1h_2}_{p_3p_4} \\ & + \mathcal{X}1^{h_1p_3}_{p_1h_3} t_2{}^{h_3h_2}_{p_3p_2} + \mathcal{X}2^{h_1p_3}_{p_1h_3} t_2{}^{h_2h_3}_{p_3p_2} + \mathcal{Y}1^{h_1p_3}_{h_3p_1} t_2{}^{h_3h_2}_{p_3p_2} \\ & + \mathcal{Y}2^{h_1p_3}_{h_3p_2} t_2{}^{h_3h_2}_{p_1p_3} - v^{p_3p_4}_{h_4h_3} t_2{}^{h_3h_1}_{p_3p_1} t_2{}^{h_4h_2}_{p_4p_2} \end{aligned} \tag{24}$$

and the factorized equation for the t_1 amplitudes is

$$\begin{aligned} t_1{}^{h_1}_{p_1} D_1 = {} & f^{p_1}_{h_1} + \mathcal{F}^{h_1}_{h_2} t_1{}^{h_2}_{p_1} + \mathcal{F}^{p_2}_{p_1} t_1{}^{h_1}_{p_2} + \mathcal{F}^{p_2}_{h_2} \tau'_2{}^{h_1h_2}_{p_1p_2} - v^{h_1p_2}_{h_3h_2} \tau'_2{}^{h_3h_2}_{p_1p_2} \\ & + v^{p_2p_3}_{h_2p_1} \tau'_2{}^{h_1h_2}_{p_3p_2} + \mathcal{X}3^{h_1p_2}_{p_1h_2} t_1{}^{h_2}_{p_2} + \mathcal{Y}1^{h_1p_2}_{h_2p_1} t_1{}^{h_2}_{p_2} \\ & - 2 v^{p_2p_3}_{h_3h_2} t_2{}^{h_2h_1}_{p_2p_1} t_1{}^{h_3}_{p_3} \end{aligned} \tag{25}$$

where the last two terms in eqs 24 and 25 are constructed in sequences similar to eq 18. The various one- and two-body intermediates are defined as follows:

$$\mathcal{F}^{p_1}_{h_1} = -f^{p_1}_{h_1} - 2 v^{p_1p_2}_{h_1h_2} t_1{}^{h_2}_{p_2} + v^{p_1p_2}_{h_2h_1} t_1{}^{h_2}_{p_2} \tag{26}$$

$$\begin{aligned} \mathcal{F}^{h_2}_{h_1} = {} & -(1 - \delta^{h_1}_{h_2}) f^{h_2}_{h_1} + \mathcal{F}^{p_1}_{h_1} t_1{}^{h_2}_{p_1} - 2 v^{h_2p_1}_{h_1h_3} t_1{}^{h_3}_{p_1} + v^{h_2p_1}_{h_3h_1} t_1{}^{h_3}_{p_1} \\ & - v^{p_1p_2}_{h_3h_1} \tau_2{}^{h_3h_2}_{p_1p_2} \end{aligned} \tag{27}$$

$$\mathcal{F}^{p_2}_{p_1} = (1 - \delta^{p_1}_{p_2}) f^{p_2}_{p_1} + 2 v^{p_3p_2}_{h_1p_1} t_1{}^{h_1}_{p_3} - v^{p_2p_3}_{h_1p_1} t_1{}^{h_1}_{p_3} + v^{p_3p_2}_{h_1h_2} \tau'_2{}^{h_1h_2}_{p_3p_1} \tag{28}$$

and the two-body intermediates

$$\mathcal{W}^{h_3p_1}_{h_1h_2} = v^{h_3p_1}_{h_1h_2} + \frac{1}{2} v^{p_3p_1}_{h_1h_2} t_1{}^{h_3}_{p_3} \tag{29}$$

$$\mathcal{W}^{h_3h_4}_{h_1h_2} = \frac{1}{2} v^{h_3h_4}_{h_1h_2} + \mathcal{W}^{h_3p_1}_{h_1h_2} t_1{}^{h_4}_{p_1} + \frac{1}{2} v^{p_1p_2}_{h_1h_2} t_2{}^{h_3h_4}_{p_1p_2} \tag{30}$$

$$\mathcal{W}1^{h_1p_2}_{p_1h_2} = -v^{h_1p_2}_{p_1h_2} - v^{p_3p_2}_{h_3h_2} \tau'_2{}^{h_1h_3}_{p_1p_3} + v^{p_3p_2}_{h_2h_3} t_2{}^{h_1h_3}_{p_1p_3} \tag{31}$$

$$\mathcal{X}1^{h_1p_2}_{p_1h_2} = 2 v^{h_1p_2}_{p_1h_2} + 2 v^{p_3p_2}_{h_3h_2} t_2{}^{h_1h_3}_{p_1p_3} - 2 v^{p_3p_2}_{h_3h_2} t_2{}^{h_3h_1}_{p_1p_3} \tag{32}$$

$$\mathcal{X}2^{h_1p_2}_{p_1h_2} = -v^{h_1p_2}_{p_1h_2} + \frac{1}{2} v^{p_3p_2}_{h_3h_2} t_2{}^{h_3h_1}_{p_1p_3} \tag{33}$$

$$\mathcal{X}3^{h_1p_2}_{p_1h_2} = 2 v^{h_1p_2}_{p_1h_2} + 2 v^{p_3p_2}_{h_3h_2} \tau'_2{}^{h_1h_3}_{p_1p_3} \tag{34}$$

$$\mathcal{W}2^{h_2p_2}_{h_1p_1} = -v^{h_2p_2}_{h_1p_1} - v^{p_3p_2}_{h_1p_1} t_1{}^{h_2}_{p_3} + v^{p_3p_2}_{h_1h_3} t_2{}^{h_2h_3}_{p_3p_1} \tag{35}$$

$$\mathcal{Y}1^{h_2p_2}_{h_1p_1} = -v^{h_2p_2}_{h_1p_1} + v^{p_3p_2}_{h_1h_3} t_2{}^{h_2h_3}_{p_3p_1} \tag{36}$$

$$\mathcal{Y}2^{h_2p_2}_{h_1p_1} = -v^{h_2p_2}_{h_1p_1} + \frac{1}{2} v^{p_3p_2}_{h_1h_3} t_2{}^{h_2h_3}_{p_3p_1} \tag{37}$$

$$\begin{aligned} \mathcal{W}^{h_1h_2}_{h_3p_1} = {} & -v^{h_3p_1}_{h_1h_2} + \mathcal{F}^{p_2}_{h_3} t_2{}^{h_1h_2}_{p_2p_1} + \mathcal{W}1^{h_2p_3}_{p_1h_3} t_1{}^{h_1}_{p_3} + \mathcal{W}2^{h_1p_3}_{h_3p_1} t_1{}^{h_2}_{p_3} \\ & + \mathcal{W}^{h_1h_2}_{h_3h_4} t_1{}^{h_4}_{p_1} + v^{h_1p_2}_{h_4h_3} t_2{}^{h_4h_2}_{p_2p_1} + v^{h_2p_2}_{h_4h_3} t_2{}^{h_1h_4}_{p_2p_1} \\ & - v^{h_1p_2}_{h_3h_4} \tau'_2{}^{h_4h_2}_{p_2p_1} - v^{p_2p_3}_{h_3p_1} t_2{}^{h_1h_2}_{p_2p_3} \end{aligned} \tag{38}$$

$$\mathcal{W}^{h_1p_3}_{p_1p_2} = v^{p_1p_2}_{h_1p_3} - v^{p_3p_4}_{h_2p_2} t_2{}^{h_1h_2}_{p_1p_4} - v^{p_3p_4}_{h_2p_1} t_2{}^{h_2h_1}_{p_2p_4} + v^{p_4p_3}_{h_2p_2} \tau'_2{}^{h_2h_1}_{p_4p_1} \tag{39}$$

The intermediates τ_2 and τ'_2 appearing in the above equations are defined by eq 20. The denominators D of eqs 24 and 25 are defined as

$$\begin{aligned} D_2 &= f^{h_1}_{h_1} + f^{h_2}_{h_2} - f^{p_1}_{p_1} - f^{p_2}_{p_2} \\ D_1 &= f^{h_1}_{h_1} - f^{p_1}_{p_1} \end{aligned} \tag{40}$$

AUTHOR INFORMATION

Corresponding Author

*E-mail: datta@uni-mainz.de; gauss@uni-mainz.de.

Notes

The authors declare no competing financial interest.

dx.doi.org/10.1021/ct400216h | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

■ ACKNOWLEDGMENTS

D.D. is indebted to Professor Marcel Nooijen for stimulating discussions on the automated derivation of equations for various CC approaches. Thanks are due also to Dr. Liguo Kong for his kind help in understanding the details of the APG developed by him, the knowledge of which helped in developing the automated implementation tool presented in this paper. The postdoctoral fellowship of the Alexander von Humboldt Foundation to D.D. and financial support from the Deutsche Forschungsgemeinschaft (DFG, GA 370/5-1) to J.G. are gratefully acknowledged.

■ REFERENCES

(1) (a) Čížek, J. *J. Chem. Phys.* **1966**, *45*, 4256−4266. (b) Čížek, J. *Adv. Chem. Phys.* **1969**, *14*, 35−89. (c) Čížek, J.; Paldus, J. *Int. J. Quantum Chem.* **1971**, *5*, 359−379.
(2) Musiał, M.; Bartlett, R. J. *Rev. Mod. Phys.* **2007**, *79*, 291−352.
(3) Shavitt, I.; Bartlett, R. J. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge University Press: Cambridge, U. K., 2009.
(4) (a) Urban, M.; Noga, J.; Cole, S. J.; Bartlett, R. J. *J. Chem. Phys.* **1985**, *83*, 4041−4046. (b) Bartlett, R. J.; Watts, J. D.; Kucharski, S. A.; Noga, J. *Chem. Phys. Lett.* **1990**, *165*, 513−522.
(5) Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. *Chem. Phys. Lett.* **1989**, *157*, 479−483.
(6) Purvis, G. D.; Bartlett, R. J. *J. Chem. Phys.* **1981**, *75*, 1284−1292.
(7) Scuseria, G. E.; Janssen, C. L.; Schaefer, H. F., III. *J. Chem. Phys.* **1988**, *89*, 7382−7387.
(8) Stanton, J. F.; Gauss, J.; Watts, J. D.; Bartlett, R. J. *J. Chem. Phys.* **1991**, *94*, 4334−4345.
(9) Hampel, C.; Peterson, K. A.; Werner, H.-J. *Chem. Phys. Lett.* **1992**, *190*, 1−12.
(10) (a) Lee, Y. S.; Kucharski, S. A.; Bartlett, R. J. *J. Chem. Phys.* **1984**, *81*, 5906−5912. (b) Noga, J.; Bartlett, R. J. *J. Chem. Phys.* **1987**, *86*, 7041−7050.
(11) Scuseria, G. E.; Schaefer, H. F., III. *Chem. Phys. Lett.* **1988**, *152*, 382−386.
(12) (a) Kucharski, S. A.; Bartlett, R. J. *Theor. Chim. Acta.* **1991**, *80*, 387−405. (b) Kucharski, S. A.; Bartlett, R. J. *J. Chem. Phys.* **1992**, *97*, 4282−4288.
(13) Olsen, J. *J. Chem. Phys.* **2000**, *113*, 7140−7148.
(14) Kállay, M.; Surján, P. R. *J. Chem. Phys.* **2000**, *113*, 1359−1365.
(15) Kállay, M.; Surján, P. R. *J. Chem. Phys.* **2001**, *115*, 2945−2954.
(16) Hirata, S. *J. Phys. Chem. A* **2003**, *107*, 9887−9897.
(17) (a) Nooijen, M.; Lotrich, V. *J. Chem. Phys.* **2000**, *113*, 4549−4557. (b) Nooijen, M.; Lotrich, V. *J. Mol. Struct. (THEOCHEM)* **2001**, *547*, 253−267.
(18) Hanrath, M.; Engels-Putzka, A. *J. Chem. Phys.* **2011**, *134*, 124106.
(19) Rittby, M.; Bartlett, R. J. *J. Phys. Chem.* **1988**, *92*, 3033−3036.
(20) Watts, J. D.; Gauss, J.; Bartlett, R. J. *J. Chem. Phys.* **1993**, *98*, 8718−8733.
(21) Stanton, J. F. *J. Chem. Phys.* **1994**, *101*, 371−374.
(22) Hirata, S.; Bartlett, R. J. *Chem. Phys. Lett.* **2000**, *321*, 216−224.
(23) Auer, A. A.; Baumgartner, G.; Bernholdt, D. E.; Bibireata, A.; Choppella, V.; Cociorva, D.; Gao, X.; Harrison, R. J.; Hartono, A.; Krishnamoorthy, S.; Krishnan, S.; Lam, C.; Lu, Q.; Nooijen, M.; Pitzer, R. M.; Ramanujam, J.; Sadayappan, P.; Sibiryakov, A. *Mol. Phys.* **2006**, *104*, 211−228.
(24) Kong, L. *Internally Contracted Multireference Coupled Cluster Method and Normal-Order-Based Automatic Code Generator*, Ph.D. thesis, University of Waterloo, Waterloo, Canada, 2009.
(25) Hanauer, M.; Köhn, A. *J. Chem. Phys.* **2011**, *134*, 204111.
(26) Gauss, J.; Lauderdale, W. J.; Stanton, J. F.; Watts, J. D.; Bartlett, R. J. *Chem. Phys. Lett.* **1991**, *182*, 207−215.
(27) Gauss, J.; Stanton, J. F.; Bartlett, R. J. *J. Chem. Phys.* **1991**, *95*, 2623−2638.
(28) Watts, J. D.; Gauss, J.; Bartlett, R. J. *Chem. Phys. Lett.* **1992**, *200*, 1−7.
(29) Szalay, P. G.; Gauss, J.; Stanton, J. F. *Theor. Chim. Acta* **1998**, *100*, 5−11.
(30) Gauss, J.; Kállay, M.; Neese, F. *J. Phys. Chem. A* **2009**, *113*, 11541−11549.
(31) Janssen, C. L.; Schaefer, H. F., III. *Theor. Chim. Acta* **1991**, *79*, 1−42.
(32) Knowles, P. J.; Hampel, C.; Werner, H.-J. *J. Chem. Phys.* **1993**, *99*, 5219−5227.
(33) Neogrády, P.; Urban, M.; Hubač, I. *J. Chem. Phys.* **1994**, *100*, 3706−3716.
(34) Szalay, P. G.; Gauss, J. *J. Chem. Phys.* **1997**, *107*, 9028−9038.
(35) Heckert, M.; Heun, O.; Gauss, J.; Szalay, P. G. *J. Chem. Phys.* **2006**, *124*, 124105.
(36) Li, X.; Paldus, J. *J. Chem. Phys.* **1994**, *101*, 8812−8826.
(37) Li, X.; Paldus, J. In *Recent Advances in Computational Chemistry*; Bartlett, R. J., Ed.; World Scientific: Singapore, 1997; Vol 3, pp 183−219.
(38) Datta, D.; Mukherjee, D. *Int. J. Quantum Chem.* **2008**, *108*, 2211−2222.
(39) Datta, D.; Mukherjee, D. *J. Chem. Phys.* **2009**, *131*, 044124.
(40) Datta, D.; Mukherjee, D. *J. Chem. Phys.* **2011**, *134*, 054122.
(41) Lutz, M.; Ascher, D. *Learning Python*, 2nd ed.; O'Reilly & Associates, Inc.: Sebastopol, CA, 2003.
(42) Wick, G. C. *Phys. Rev.* **1950**, *80*, 268−272.
(43) Goldstone, J. *Proc. R. Soc. London, Ser. A* **1957**, *239*, 267−279.
(44) Brainerd, W. S.; Goldberg, C. H.; Adams, J. C. *Programmer's Guide to Fortran 90*, third ed.; Springer-Verlag: New York, 1996.
(45) Nooijen, M.; Bartlett, R. J. *J. Chem. Phys.* **1996**, *104*, 2652−2668.
(46) (a) Lindgren, I.; Mukherjee, D. *Phys. Rep.* **1987**, *151*, 93−127. (b) Mukherjee, D.; Pal, S. *Adv. Quantum Chem.* **1989**, *20*, 291−373.
(47) Lawson, C. L.; Hanson, R. J.; Kincaid, D. R.; Krogh, F. T. *ACM Trans. Math. Soft.* **1979**, *5*, 308−323.
(48) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. J.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. *J. Comput. Chem.* **1993**, *14*, 1347−1363.
(49) Stanton, J. F.; Gauss, J.; Harding, M. E.; Szalay, P. G.; Auer, A. A.; Bartlett, R. J.; Benedikt, U.; Bernholdt, D. B.; Berger, C.; Cheng, L.; Christiansen, O.; Heckert, M.; Heun, O.; Huber, C.; Jagau, T.-C.; Jonsson, D.; Jusélius, J.; Klein, K.; Lauderdale, W. J.; Matthews, D.; Metzroth, T.; O'Neill, D. P.; Price, D. R.; Prochnow, E.; Ruud, K.; Schiffmann, F.; Stopkowicz, S.; Schwalbach, W.; Tajti, A.; Varner, M. E.; Vázquez, J.; Watts, J. D., Wang, F. *CFOUR: Coupled cluster techniques for computational chemistry*, with the integral packages MOLECULE (Almlöf J.; Taylor, P. R.), PROPS (Taylor, P. R.), ABACUS (Helgaker, T.; Jensen, H. J. A.; Jørgensen, P.; Olsen, J.), and ECP routines by Mitin, A. V.; van Wüllen C. For the current version, see http://www.cfour.de.
(50) Dunning, T. H. *J. Chem. Phys.* **1989**, *90*, 1007−1023.
(51) Beckman, K. B.; Ames, B. N. *Physiol. Rev.* **1998**, *78*, 547−581.
(52) Marnett, L. J. *Carcinogenesis* **2000**, *21*, 361−370.
(53) (a) Krauss, M.; Osman, R. *J. Phys. Chem. A* **1997**, *101*, 4117−4120. (b) Tureček, F.; Yao, C. *J. Phys. Chem. A* **2003**, *107*, 9221−9231.
(54) Kuś, T.; Lotrich, V.; Bartlett, R. J. *J. Chem. Phys.* **2009**, *130*, 124122.
(55) Sadlej, A. J. *Theor. Chim. Acta* **1992**, *79*, 123−140.
(56) Evangelista, F. A.; Gauss, J. *J. Chem. Phys.* **2011**, *134*, 114102.
(57) Datta, D.; Kong, L.; Nooijen, M. *J. Chem. Phys.* **2011**, *134*, 214116.