Tensor Contraction Engine: Abstraction and Automated Parallel Implementation of Configuration-Interaction, Coupled-Cluster, and Many-Body Perturbation Theories

So Hirata*

William R. Wiley Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, P.O.Box 999, Richland, Washington 99352

Received: March 7, 2003; In Final Form: July 16, 2003

We have developed a symbolic manipulation program and program generator (tensor contraction engine or TCE) that abstracts and automates the time-consuming, error-prone processes of deriving the working equations of a well-defined model of second-quantized many-electron theories and synthesizing efficient parallel computer programs on the basis of these equations. Provided an ansatz of a many-electron theory model, TCE performs valid contractions of creation and annihilation operators according to Wick's theorem, consolidates identical terms, and reduces the expressions into the form of multiple tensor contractions acted upon by permutation operators. It subsequently determines the binary contraction order for each multiple tensor contraction with the minimal operation and memory cost, factorizes common binary contractions (defines intermediate tensors), and identifies reusable intermediates. The resulting ordered list of binary tensor contractions, additions, and index permutations is translated into an optimized program that is combined with the NWCHEM and UTCHEM computational chemistry software packages. The programs synthesized by TCE take advantage of spin symmetry (within the spin−orbital formalisms), real Abelian point-group symmetry, and index permutation symmetry at every stage of the calculations to minimize the number of arithmetic operations and storage requirement, adjust the peak local memory usage by index-range tiling, and support parallel I/O interfaces and dynamic load balancing for parallel executions. We demonstrate the utility of TCE through automatic derivation and implementation of parallel programs for a range of predictive computational methods—configuration-interaction theory (CISD, CISDT, CISDTQ), generalized many-body perturbation theory [MBPT(2), MBPT(3), MBPT(4)], and coupled-cluster theory (LCCD, CCD, LCCSD, CCSD, QCISD, CCSDT, and CCSDTQ), some for the first time—and discuss the performance of the implemented programs.

1. Introduction

Electronic computers have enabled a complex sequence of arithmetic operations in quantum mechanical calculations of many-electron systems. Equally complex are the symbolic manipulation processes of deriving the working equations of many-electron theories and implementing efficient computer programs on the basis of these equations, which are also subject to abstraction and automation by computers. The objective of this study is to develop a general-purpose computer program that performs both symbolic manipulation processes—a program that manipulates second-quantized operators and derives the working equations of any well-defined second-quantized many-electron theory, analyzes these equations, and translates them into a thoroughly optimized parallel program. The significance of such a program is evident. (1) It expedites time-consuming and error-prone derivation and computer implementation of various many-electron theory models, (2) it facilitates parallelization and other laborious optimization of synthesized programs, which may be tailored to a particular computer architecture, (3) it enhances the portability, maintainability, and extensibility of synthesized programs, and (4) it helps design and test a new many-electron theory model or implements models that are too complex to be hand-coded.

A number of researchers have capitalized on computer-aided formula derivation and program synthesis in the past, and some symbolic manipulation programs with a varied degree of sophistication have been developed.[1−7] A pioneering and perhaps the most thorough study was conducted by Janssen and Schaefer,[3] who built a computer program that automated the derivation and computer implementation of coupled-cluster models for open-shell systems. There have also been some studies that aimed at performing calculations of various many-electron theory models in a single algorithmic framework, sometimes at the expense of efficiency. The prime examples are determinant-based general-order many-body perturbation[8,9] and coupled-cluster theories[10−12] and the remarkable string-based general-order coupled-cluster theory of Kállay and Surján.[13] These and the computer-aided formula derivation and program synthesis are closely related in the sense that both approaches require a high degree of abstraction of the equations and constituent quantities of many-electron theories.[14]

We have developed a symbolic manipulation program of second-quantized operators and a program generator, which we call a tensor contraction engine or TCE,[15] adopting the design philosophy of Janssen and Schaefer. TCE inherits various techniques invented by these and other authors,[3,5−7,13,14] but its applicability is broadened and its capabilities in equation analysis and program optimization are significantly enhanced to the extent that the computer-synthesized programs can compete with hand-coded programs in terms of operation and memory cost.

TCE is based on the second-quantized representation of many-electron theories, which is general and covers a wide spectrum of models ranging from configuration-interaction (CI) theory, many-body perturbation theory (MBPT), and coupled-cluster

* E-mail: so.hirata@pnl.gov.

10.1021/jp034596z CCC: $25.00
Published on Web 10/28/2003

(CC) theory.[16−19] Given a definition of a many-electron theory as the quasi-vacuum expectation values of normal-ordered second-quantized operators, TCE performs valid contractions of creation and annihilation operators according to Wick's theorem, consolidates identical terms, and reduces the expressions into the form of multiple tensor contractions acted upon by permutation operators. TCE subsequently performs strength reduction (determines the binary contraction order for each multiple tensor contraction that has the minimal operation and memory cost), factorization (eliminates common binary contractions and defines intermediate tensors), and common subexpression elimination (intermediate reuse). The resulting ordered list ("operation tree") of binary tensor contractions, additions, and index permutations is translated into an optimized program that is combined with a high-performance quantum chemistry program package tailored to parallel computer environments. The programs synthesized by TCE take advantage of spin, spatial, and index permutation symmetries at every stage of the calculations to reduce the operation cost and storage requirement, adjust the peak local memory usage by index-range tiling, and support multiple parallel I/O interfaces and dynamic load balancing for parallel executions.

In this article, we describe the machinery of TCE and discuss the characteristics of the equations and intermediate quantities of many-electron theories upon which TCE is based. To render TCE widely applicable and to advance its program optimization capabilities, we answer the following questions: (1) What are the adequate representations of tensors, second-quantized operators, permutation operators, and so forth that permit rapid pattern-matching operations? (2) What is the rational definition of intermediate tensors that have desirable index permutation symmetry? (3) How can we take advantage of spin, spatial, and index permutation symmetries simultaneously to minimize the number of arithmetic operations? (4) How can we adjust the peak memory usage without significantly increasing the operation cost? (5) What is the universal storage scheme for tensors compressed by the use of spin, spatial, and index permutation symmetries? (6) How can we effectively parallelize the entire calculation? We demonstrate the computer-aided implementation of high-performance parallel programs for various models of many-electron theories that include configuration-interaction theory (CISD, CISDT, CISDTQ), many-body perturbation theory [MBPT(2), MBPT(3), MBPT(4)], and coupled-cluster theory (LCCD, CCD, LCCSD, CCSD, QCISD, CCSDT, and CCSDTQ).

2. Machinery of Tensor Contraction Engine

2.1. Derivation of Working Equations. The vast majority of modern many-electron theories for electron correlation problems are defined in terms of the expectation values of the nonrelativistic electronic Hamiltonian and other operators for wave functions expanded by Slater determinants. The expectation values, in their simplest forms, are usually tensor algebraic expressions in which the tensors represent certain physical interactions.[14] Therefore, the derivation of working equations amounts to evaluating the expectation values of operators for determinantal wave functions and reducing the resulting algebraic expressions into the simplest form of tensor contractions and additions. This may be accomplished in various ways,[16] but three widely used approaches are the one based on Slater's rules, the method of second quantization, and the diagrammatic approach. TCE adopts the method of second quantization that appears to blend the applicability and expediency in the most adequate balance for the purpose of building a universal and efficient symbolic manipulation program.

The process of evaluating operator expectation values with second quantization can be significantly simplified and accelerated by Wick's theorem, which states that a string of creation and annihilation operators is a sum of all possible partial contractions of the string in the normal order. A normal-ordered string of operators (enclosed by a pair of braces) relative to a quasi-vacuum is defined as the rearrangement of the operators such that all hole annihilation and particle creation operators are to the left of all hole creation and particle annihilation operators. The significance of Wick's theorem is that the quasi-vacuum expectation value of a normal-ordered string of operators vanishes unless it is fully contracted.

An initial task of TCE (an interpreted, interactive, object-oriented program written in Python programming language) is therefore to perform all valid contractions of second-quantized operators, provided a definition of a many-electron theory model is expressed in terms of quasi-vacuum expectation values of a product of normal-ordered strings of operators, such as

$$+\frac{1}{128}v^{g_5g_6}_{g_7g_8}t^{p_9p_{10}}_{h_{11}h_{12}}t^{p_{13}p_{14}}_{h_{15}h_{16}}\langle 0|\{h_1^\dagger h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8g_7\}\{p_9^\dagger p_{10}^\dagger h_{12}h_{11}\}\{p_{13}^\dagger p^\dagger{}_{14}h_{16}h_{15}\}|\,0\rangle \quad (1)$$

which is a part of the CCSD T_2 amplitude equation. Here, h_n denotes a hole index, p_n denotes a particle index, g_n denotes either a hole or a particle index, $v^{g_5g_6}_{g_7g_8}$ represents an antisymmetrized two-electron integral, $t^{p_{13}p_{14}}_{h_{15}h_{16}}$ represents an excitation amplitude, and the Einstein convention of implied summation is employed. A valid contraction is the one between a hole creation and a hole annihilation operator and between a particle annihilation and a particle creation operator across different normal-ordered strings (i.e., excluding internal contractions). TCE performs this procedure iteratively, with each cycle of iteration consisting of the following steps. First, TCE performs all possible contractions of an operator. Because the order of contraction is immaterial, TCE elects to contract the leftmost operator with another operator. After a single contraction of the leftmost operator, eq 1 becomes

$$-\frac{1}{128}v^{g_5g_6}_{g_7h_1}t^{p_9p_{10}}_{h_{11}h_{12}}t^{p_{13}p_{14}}_{h_{15}h_{16}}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_7\}\{p_9^\dagger p_{10}^\dagger h_{12}h_{11}\}\{p_{13}^\dagger p_{14}^\dagger h_{16}h_{15}\}|0\rangle$$

$$+\frac{1}{128}v^{g_5g_6}_{h_1g_8}t^{p_9p_{10}}_{h_{11}h_{12}}t^{p_{13}p_{14}}_{h_{15}h_{16}}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8\}\{p_9^\dagger p_{10}^\dagger h_{12}h_{11}\}\{p_{13}^\dagger p_{14}^\dagger h_{16}h_{15}\}|0\rangle$$

$$-\frac{1}{128}v^{g_5g_6}_{g_7g_8}t^{p_9p_{10}}_{h_{11}h_1}t^{p_{13}p_{14}}_{h_{15}h_{16}}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8g_7\}\{p_9^\dagger p_{10}^\dagger h_{11}\}\{p_{13}^\dagger p_{14}^\dagger h_{16}h_{15}\}|0\rangle$$

$$+\frac{1}{128}v^{g_5g_6}_{g_7g_8}t^{p_9p_{10}}_{h_1h_{12}}t^{p_{13}p_{14}}_{h_{15}h_{16}}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8g_7\}\{p_9^\dagger p_{10}^\dagger h_{12}\}\{p_{13}^\dagger p_{14}^\dagger h_{16}h_{15}\}|0\rangle$$

$$-\frac{1}{128}v^{g_5g_6}_{g_7g_8}t^{p_9p_{10}}_{h_{11}h_{12}}t^{p_{13}p_{14}}_{h_{15}h_1}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8g_7\}\{p_9^\dagger p_{10}^\dagger h_{12}h_{11}\}\{p_{13}^\dagger p_{14}^\dagger h_{15}\}|0\rangle$$

$$+\frac{1}{128}v^{g_5g_6}_{g_7g_8}t^{p_9p_{10}}_{h_{11}h_{12}}t^{p_{13}p_{14}}_{h_1h_{16}}\langle 0|\{h_2^\dagger p_4p_3\}\{g_5^\dagger g_6^\dagger g_8g_7\}\{p_9^\dagger p_{10}^\dagger h_{12}h_{11}\}\{p_{13}^\dagger p_{14}^\dagger h_{16}\}|0\rangle \quad (2)$$

Even when a contraction gives rise to nonvanishing terms, the inspection of the number of remaining creation and

annihilation operators and their types in the strings can indicate that they eventually lead to only vanishing contributions when fully contracted. TCE identifies and erases partially contracted strings that will vanish when fully contracted. It is critical to erase them early to maintain the number of partially contracted strings at a manageable level, which otherwise tends to grow exponentially. In the above example (eq 2), the first two terms vanish when fully contracted.

A contraction can often give rise to the equivalent terms multiple times, which differs merely in the order of tensor parts of operators (e.g., integrals and excitation amplitudes), the order of permutable indices (e.g., indices of an integral matrix), or the labels of common (summation) indices. A simple and efficient way to consolidate these equivalent terms is to recast the expressions in a canonical form[3] so that the expressions of the two equivalent terms become identical by character and are rapidly merged. The definition of a canonical form is rather arbitrary, and TCE adopts the following: the tensor parts of the operators are in alphabetical order, then in the order of their ranks, then in the order of labels of the indices that are not among the common indices. All common indices are subsequently relabeled and sorted in a unique order. Equation 2 has only two distinct nonvanishing terms and is hence rewritten in the canonical form as

$$-\frac{1}{64}t^{p_5p_6}_{h_7h_1}t^{p_8p_9}_{h_{10}h_{11}}v^{g_{12}g_{13}}_{g_{14}g_{15}}\langle 0|\{h_2^\dagger p_4 p_3\}\{g_{12}^\dagger g_{13}^\dagger g_{15}g_{14}\}\{p_5^\dagger p_6^\dagger h_7\}\{p_8^\dagger p_9^\dagger h_{11}h_{10}\}|\,0\rangle$$

$$-\frac{1}{64}t^{p_5p_6}_{h_7h_1}t^{p_8p_9}_{h_{10}h_{11}}v^{g_{12}g_{13}}_{g_{14}g_{15}}\langle 0|\{h_2^\dagger p_4 p_3\}\{g_{12}^\dagger g_{13}^\dagger g_{15}g_{14}\}\{p_8^\dagger p_9^\dagger h_{11}h_{10}\}\{p_5^\dagger p_6^\dagger h_7\}|\,0\rangle \quad (3)$$

For an ansatz containing $2n$ second-quantized operators, n cycles of an iterative contraction procedure lead to fully contracted expressions

$$\begin{aligned}&-\frac{1}{2}t^{p_5p_3}_{h_1h_2}t^{p_6p_4}_{h_7h_8}v^{h_7h_8}_{p_5p_6}+\frac{1}{2}t^{p_5p_4}_{h_1h_2}t^{p_6p_3}_{h_7h_8}v^{h_7h_8}_{p_5p_6}-\frac{1}{2}t^{p_3p_4}_{h_5h_1}t^{p_6p_7}_{h_8h_2}v^{h_5h_8}_{p_6p_7}\\&+\frac{1}{2}t^{p_3p_4}_{h_5h_2}t^{p_6p_7}_{h_8h_1}v^{h_5h_8}_{p_6p_7}-t^{p_5p_4}_{h_6h_1}t^{p_7p_3}_{h_8h_2}v^{h_6h_8}_{p_5p_7}+t^{p_5p_3}_{h_6h_1}t^{p_7p_4}_{h_8h_2}v^{h_6h_8}_{p_5p_7}\\&+\frac{1}{4}t^{p_5p_6}_{h_1h_2}t^{p_3p_4}_{h_7h_8}v^{h_7h_8}_{p_5p_6}+\frac{1}{4}t^{p_3p_4}_{h_1h_2}t^{p_5p_6}_{h_7h_8}v^{h_7h_8}_{p_5p_6}\end{aligned} \quad (4)$$

for the above example. They need to be further simplified by virtue of the topological properties of the expressions akin to those of corresponding diagrammatic representations. First, the tensor contraction expressions that correspond to disconnected diagrams may optionally be deleted. The connectedness of tensor contraction expressions can be inferred straightforwardly by chasing the tensor indices. In eq 4, the last term is disconnected because the only contraction takes place between tensor v and one of the two tensors t. Examining the connectedness may also identify "cyclic" tensor contraction expressions, such as

$$+\frac{1}{8}(t^{p_1p_2}_{h_3h_4})^*t^{p_5p_6}_{h_3h_4}v^{p_1p_2}_{p_5p_6} \quad (5)$$

whose corresponding diagram contains a closed loop formed by more than two vertices. A cyclic tensor contraction typically arises when deexcitation operators are employed in the ansatz. The expressions of this type cannot be handled effectively by the canonicalization technique. When TCE detects a cyclic tensor contraction expression, it performs a more rigorous comparison of two expressions by permuting the common indices in all possible ways to examine whether the two are equivalent.

Subsequently, TCE examines index permutation symmetries among the tensor contraction expressions. When there are two or more tensor contraction expressions that are related to each other by index permutation, TCE consolidates them into one tensor contraction expression acted upon by a sum of operators that permute just the indices of output tensors (a permutation of common indices merely gives rise to equivalent expressions). Equation 4 (after the disconnected term is deleted) can thus be simplified to

$$\begin{aligned}&+\frac{1}{2}(1-P^{p_3p_4h_1h_2}_{p_4p_3h_1h_2})t^{p_5p_4}_{h_1h_2}t^{p_6p_3}_{h_7h_8}v^{h_7h_8}_{p_5p_6}-\frac{1}{2}(1-P^{p_3p_4h_1h_2}_{p_3p_4h_2h_1})t^{p_3p_4}_{h_5h_1}t^{p_6p_7}_{h_8h_2}v^{h_5h_8}_{p_6p_7}\\&+\frac{1}{4}t^{p_5p_6}_{h_1h_2}t^{p_3p_4}_{h_7h_8}v^{h_7h_8}_{p_5p_6}-(1-P^{p_3p_4h_1h_2}_{p_4p_3h_1h_2})t^{p_5p_4}_{h_6h_1}t^{p_7p_3}_{h_8h_2}v^{h_6h_8}_{p_5p_7}\end{aligned} \quad (6)$$

where permutation operators are also expressed in the tensor notation (subscript particle indices of the permutation operator are replaced by the corresponding superscript particle indices, and superscript hole indices are replaced by the corresponding subscript hole indices in the above equation). Occasionally, an index permutation of output tensors acting upon a tensor contraction expression results in an expression equivalent to the original expression. This occurs when there are two or more equivalent tensors in the expression (i.e., the tensors of the same type and rank contracted in the same topological manner). In this situation, TCE prefers to rewrite the expression in a more symmetric form with a permutation operator, for reasons that will become apparent. The final tensor contraction expressions for eq 1 are

$$\begin{aligned}&+\frac{1}{2}(1-P^{p_3p_4h_1h_2}_{p_4p_3h_1h_2})t^{p_5p_4}_{h_1h_2}t^{p_6p_3}_{h_7h_8}v^{h_7h_8}_{p_5p_6}-\frac{1}{2}(1-P^{p_3p_4h_1h_2}_{p_3p_4h_2h_1})t^{p_3p_4}_{h_5h_1}t^{p_6p_7}_{h_8h_2}v^{h_5h_8}_{p_6p_7}\\&+\frac{1}{4}t^{p_5p_6}_{h_1h_2}t^{p_3p_4}_{h_7h_8}v^{h_7h_8}_{p_5p_6}-\frac{1}{2}(1-P^{p_3p_4h_1h_2}_{p_4p_3h_1h_2}+P^{p_3p_4h_1h_2}_{p_4p_3h_2h_1}-P^{p_4p_3h_1h_2}_{p_4p_3h_2h_1})t^{p_5p_4}_{h_6h_1}t^{p_7p_3}_{h_8h_2}v^{h_6h_8}_{p_5p_7}\end{aligned} \quad (7)$$

Note that the permutation operator of the last term of eq 6 is symmetrized.

2.2. Generation of an Operation Tree. Although multiple tensor contraction expressions, such as eq 7, may be implemented for the purpose of verifying the expressions themselves, they are premature for the synthesis of a high-performance program. They must first undergo (1) the canonicalization of permutation operator expressions, (2) the strength reduction (which determines the order of contractions), (3) the canonicalization of binary tensor contraction expressions, (4) the factorization, and (5) the common subexpression elimination (intermediate reuse). The result of these processes is an operation tree consisting of binary tensor contractions and additions.

The canonicalization of permutation operator expressions serves the dual purposes of guiding the program generator to invoke index permutation symmetry and facilitating the subsequent optimizations. The use of the index permutation symmetry is crucial in a many-electron theory calculation because it dramatically reduces both the storage and operation costs of a tensor contraction and it also enhances the stability of the calculation. Consider the following example taken from the CCSDT T_3 amplitude equation:

$$\chi^{p_4p_5p_6}_{h_1h_2h_3}=+\frac{1}{2}(1-P^{p_4p_5p_6h_3h_1h_2}_{p_4p_5p_6h_2h_1h_3}-P^{p_4p_5p_6h_3h_2h_1}_{p_4p_5p_6h_1h_2h_3})t^{p_4p_5p_6}_{h_7h_8h_3}v^{h_7h_8}_{h_1h_2} \quad (8)$$

By virtue of the index permutation symmetry of input and output tensors

$$t^{p_4p_5p_6}_{h_7h_8h_3} = -t^{p_4p_5p_6}_{h_8h_7h_3} = -t^{p_5p_4p_6}_{h_7h_8h_3} = \cdots = t^{p_6p_5p_4}_{h_3h_8h_7} \tag{9}$$

$$v^{p_7p_8}_{h_1h_2} = -v^{p_7p_8}_{h_2h_1} = -v^{p_8p_7}_{h_1h_2} = v^{p_8p_7}_{h_2h_1} \tag{10}$$

we may store just the nonredundant elements of each tensor, which is equivalent to restricting the ranges of indices as $t^{p_4<p_5<p_6}_{h_7<h_8<h_3}$, $v^{h_7<h_8}_{p_1<p_2}$, and $\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3}$. This not only reduces the storage space (by a factor of ca. 36 for a T_3 amplitude tensor) but also ensures the antisymmetry of the wave function and prohibits the calculation from erroneously converging to a nonphysical symmetric or asymmetric wave function. The operation cost of a tensor contraction can also be reduced by restricting the ranges of the indices of the output tensor (the external indices) and common indices. In eq 8, we can recover all of the nonredundant elements of the output tensor $\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3}$ by performing the contraction within the restricted ranges of $p_4 < p_5 < p_6$ and $h_1 < h_2$ and by subsequently applying the permutation operator to the result. Likewise, we may vary the common indices within the triangular range of $h_7 < h_8$ and multiply the result by an appropriate scalar factor to compensate for the reduction in the index ranges. We may therefore rewrite eq 8 as

$$\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3} = +(1 - P^{p_4p_5p_6h_3h_1h_2}_{p_4p_5p_6h_2h_1h_3} - P^{p_4p_5p_6h_3h_2h_1}_{p_4p_5p_6h_1h_2h_3}) t^{p_4<p_5<p_6}_{h_7<h_8h_3} v^{h_7<h_8}_{h_1<h_2} \tag{11}$$

to highlight the use of index permutation symmetry (the operation cost of this contraction is reduced by a factor of ca. 24). This can be transformed further into a more transparent expression by rearranging the indices of the permutation operators into a "canonical" form:

$$\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3} = +(1 - P^{h_1<h_2h_3p_4<p_5<p_6}_{h_1<h_3h_2p_4<p_5<p_6} + P^{h_1<h_2h_3p_4<p_5<p_6}_{h_2<h_3h_1p_4<p_5<p_6}) t^{p_4<p_5<p_6}_{h_7<h_8h_3} v^{h_7<h_8}_{h_1<h_2} \tag{12}$$

or equivalently

$$\xi^{p_4<p_5<p_6}_{h_1<h_2h_3} = t^{p_4<p_5<p_6}_{h_7<h_8h_3} v^{h_7<h_8}_{h_1<h_2} \tag{13}$$

$$\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3} = \xi^{p_4<p_5<p_6}_{h_1<h_2h_3} - \xi^{p_4<p_5<p_6}_{h_1<h_3h_2} + \xi^{p_4<p_5<p_6}_{h_2<h_3h_1} \tag{14}$$

A canonical permutation operator maps the index ranges of the tensor it acts upon back onto the original index ranges as it permutes the indices. Consequently, as eq 14 illustrates, it is not necessary to lift the restrictions on the index ranges of ξ, and hence it offers the most compact way of performing tensor contractions with index permutation symmetry (see also section 2.3). This is contrasted with the noncanonical permutation operators in eq 11, which entail partially extended intermediate storage for ξ.

A permutation operator can be brought to a canonical form by first rearranging the columns of indices in ascending order of the origin indices (i.e., covariant (subscript) particle indices and contravariant (superscript) hole indices in the above example) and then by sorting the destination indices (i.e., contravariant (superscript) particle indices and covariant (subscript) hole indices) whose corresponding origin indices are in a "permutable set." A permutable set is a subset of the external indices, any permutation of which leaves the tensor that the permutation operator acts upon unchanged, apart from its parity. In eq 11, the permutable sets are $\{p_4, p_5, p_6\}$ and $\{h_1, h_2\}$. The canonical expression of a permutation operator is unique and is hence a convenient representation for the subsequent strength reduction, factorization, and common subexpression elimination processes.

Strength reduction refers to the process of finding the order of contractions in a multiple tensor contraction with the minimal number of arithmetic operations. Owing to the associativity and commutativity of a tensor contraction, the outcome of a multiple tensor contraction is invariant to the contraction order, whereas the operation cost can be strongly dependent on this parameter. There are $(m - 1)!$ distinct contraction orders of an m-fold multiple-tensor contraction,[20] and TCE determines the best order by comparing the peak operation cost, then the peak memory cost, then the aggregate operation cost, and then the aggregate memory cost of all possible contraction orders. The operation and memory cost are measured in terms of a polynomial of the ranges of hole and particle indices, and it is assumed that the range of a particle index is marginally greater than that of a hole index. The strength reduction must also be applied to permutation operators. An overall permutation operator of a multiple tensor contraction can be expressed as a product of component permutation operators in a number of ways. However, for a given contraction order, there is only one sensible way of decomposing the associated overall permutation operator that ensures that the intermediate tensors defined by the binary tensor contractions acted upon by the component permutation operators will have the desired index permutation symmetry. For instance, a quadruple tensor contraction that appears in the CCSDT T_3 amplitude equation

$$\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3} = -P_9 t^{p_7}_{h_1} t^{p_6}_{h_8} t^{p_9<p_4<p_5}_{h_{10}<h_2<h_3} v^{h_8<h_{10}}_{p_7<p_9} \tag{15}$$

can be executed stepwise as

$$\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3} = -(1 + P^{h_1<h_2<h_3p_5<p_6p_4}_{h_1<h_2<h_3p_4<p_5p_6} - P^{h_1<h_2<h_3p_4<p_6p_5}_{h_1<h_2<h_3p_4<p_5p_6}) \xi^{h_8p_4<p_5}_{h_1<h_2<h_3} t^{p_6}_{h_8} \tag{16}$$

$$\xi^{h_8p_4<p_5}_{h_1<h_2<h_3} = +(1 - P^{h_1h_2<h_3p_4<p_5p_6}_{h_2h_1<h_3p_4<p_5p_6} + P^{h_1h_2<h_3p_4<p_5p_6}_{h_3h_1<h_2p_4<p_5p_6}) \kappa^{h_8<h_{10}}_{h_1p_9} t^{p_4<p_5<p_9}_{h_2<h_3<h_{10}} \tag{17}$$

$$\kappa^{h_8<h_{10}}_{h_1p_9} = +v^{h_8<h_{10}}_{p_7<p_9} t^{p_7}_{h_1} \tag{18}$$

The overall permutation operator (P_9 in abbreviated notation) is factorized such that intermediate tensors (ξ and κ) become antisymmetric to the interchange of any pair of contravariant or covariant indices of the output tensor χ (the external indices) and to the interchange of any pair of the remaining contravariant or covariant indices (the internal indices). The intermediate $\xi^{h_8p_4<p_5}_{h_1<h_2<h_3}$, for instance, has two contravariant external indices $\{p_4, p_5\}$, one contravariant internal index $\{h_8\}$, and three covariant external indices $\{h_1, h_2, h_3\}$, and its desired index permutation symmetry is $p_4 < p_5$ and $h_1 < h_2 < h_3$. The contraction of $\kappa^{h_8<h_{10}}_{h_1p_9}$ and $t^{p_4<p_5<p_9}_{h_2<h_3<h_{10}}$ brings together the three covariant external indices $h_1 < h_2 < h_3$ to form $\xi^{h_8p_4<p_5}_{h_1<h_2<h_3}$ at the expense of $h_8 < h_{10}$ and $p_5 < p_9$. Therefore, for the intermediate $\xi^{h_8p_4<p_5}_{h_1<h_2<h_3}$ to be antisymmetric to permutations among $\{h_1, h_2, h_3\}$, we must antisymmetrize the contraction with the permutation operator that interchanges h_1 with h_2 and h_1 with h_3. Likewise, the intermediate $\chi^{p_4<p_5<p_6}_{h_1<h_2<h_3}$ is endowed with index permutation symmetry among $\{p_4, p_5, p_6\}$ by the permutation operator that interchanges p_4 with p_6 and p_5 with p_6.

As this example illustrates, the appropriate form of a component permutation operator is dictated by the form of the binary tensor contraction. Therefore, TCE performs strength reduction by generating component permutation operators on the basis of the binary tensor contractions and subsequently examining the compatibility between the component permutation operators and the overall permutation operator. When the overall permutation operator is not equal to the product of the component permutation operators, TCE is unable to process the equations any further. This might appear to be a limitation of TCE, but this is not the case. Incompatibility occurs only when the ansatz supplied to TCE intrinsically violates the antisymmetry of wave functions and hence the mechanism prevents a faulty ansatz from going undetected.

Judging the identity of two permutation operators deserves caution. For example, apparently distinct permutation operators $P^{h_1h_2h_3p_5p_6p_4}_{h_1h_2h_3p_4p_5p_6}$ and $-P^{h_1h_2h_3p_6p_5p_4}_{h_1h_2h_3p_4p_5p_6}$ are equivalent to each other in the context of eq 16 because $\xi^{h_8p_4<p_5}_{h_1<h_2<h_3}$ is antisymmetric to the interchange of p_4 and p_5. The canonicalization of permutation operators expedites this judgment by bringing the expressions of two equivalent permutation operators into an identical form. The symmetrization of permutation operators mentioned in section 2.1 also concerns this process. Because the products of component permutation operators are symmetric with respect to the interchange of two equivalent tensors, overall permutation operators must be symmetrized beforehand.

Let us consider the following example of binary tensor contractions and additions that result from the strength reduction process:

$$\chi^{p_4p_5p_6}_{h_1h_2h_3} = P_9\xi^{h_7p_4}_{h_1h_3}t^{p_5p_6}_{h_2h_7} - \frac{1}{2}P_9\kappa^{h_7p_6}_{h_1h_2}t^{p_4p_5}_{h_3h_7} \tag{19}$$

$$\xi^{h_7p_4}_{h_1h_3} = P_2t^{p_4p_8}_{h_3h_9}v^{h_7h_9}_{h_1p_8} \tag{20}$$

$$\kappa^{h_7p_6}_{h_1h_2} = t^{p_8p_9}_{h_1h_2}v^{h_7p_6}_{p_8p_9} \tag{21}$$

Equation 19 is factorizable, and the two binary tensor contractions and an addition can be converted into a binary tensor addition and a contraction by virtue of the distributive and commutative nature of tensor algebra. However, in the above representation of the tensor expressions, the factorizability is inconspicuous, owing to the apparent mismatch of the indices in the common tensors ($t^{p_5p_6}_{h_2h_7}$ and $t^{p_4p_5}_{h_3h_7}$ in the above equation). The mismatch arises partially from the degrees of freedom in representing a set of tensor contractions that are related to each other by index permutation symmetry. Therefore, prior to factorization, TCE canonicalizes binary tensor contraction expressions so that the two tensors that are contracted are sorted in a unique order and the external indices are in ascending order across the two tensors (the corresponding permutation operator is also re-expressed accordingly). Equations 19−21 are rewritten as

$$\begin{aligned}\chi^{p_4p_5p_6}_{h_1h_2h_3} &= -P_9t^{p_4p_5}_{h_1h_7}\xi^{h_7p_6}_{h_2h_3} - \frac{1}{2}P_9t^{p_4p_5}_{h_1h_7}\kappa^{h_7p_6}_{h_2h_3} \\ &= -P_9t^{p_4p_5}_{h_1h_7}\left(\xi^{h_7p_6}_{h_2h_3} + \frac{1}{2}\kappa^{h_7p_6}_{h_2h_3}\right)\end{aligned} \tag{22}$$

$$\xi^{h_7p_4}_{h_1h_2} = -P_2t^{p_4p_8}_{h_1h_9}v^{h_7h_9}_{h_2p_8} \tag{23}$$

$$\kappa^{h_7p_4}_{h_1h_2} = t^{p_8p_9}_{h_1h_2}v^{h_7p_4}_{p_8p_9} \tag{24}$$

in the canonical form that exposes the common tensors and lends itself to rapid factorization. (It also manifests the common loop indices and expedites loop fusion and space-time tradeoff optimization,[21] which are, however, beyond the scope of this study.) TCE identifies every factorizable pair of binary tensor contractions and replaces each of them by an addition and a contraction, under the tacit assumption that the strength reduction and the factorization are decoupled. Finding an optimal contraction sequence of a coupled strength reduction and factorization problem involves an intractably large search space, and the simple exhaustive search algorithm adopted here for the decoupled problem will not be adequate, although an effective approach to finding a near-optimal solution does seem to exist (e.g., the recursive intermediate factorization scheme of Kucharski and Bartlett).[4] Nevertheless, the gain in performance on going from an optimal sequence of the decoupled problem to that of the coupled one will not be substantial (see section 3).

TCE also supports the common subexpression elimination that permits the precomputation and reuse of the results of equivalent contractions (persistent intermediates as opposed to volatile intermediates) of two input (nonintermediate) tensors at the cost of increased storage space. This optimization, despite its appeal, will not necessarily be invoked in the applications discussed in this article because the reduction in the operation cost by the optimization is relatively insignificant and is largely canceled by increased memory and I/O operation cost in practice.

Tables 1−3 are examples of operation trees generated by TCE, which may be viewed as intermediate codes that describe the operations, the order of execution, and the data dependencies. Each operation that appears in the trees is either a unary tensor substitution, a binary tensor contraction, or a tensor addition by virtue of the strength reduction. Many of the volatile intermediates ξ are defined as a sum of tensors as a consequence of the factorization, and the intermediates represented by Ξ are persistent intermediates that are reused several times. The order of entries in the operation trees reflects the data flow. Internally, TCE stores an operation tree as a data structure analogous to that of a directed acyclic graph (DAG),[22] each node of the tree representing a binary tensor contraction or a unary tensor substitution and an edge connecting two adjacent nodes corresponding to a tensor addition. The leaf nodes of the tree are binary tensor contractions or unary tensor substitutions of input tensors or persistent intermediate tensors. The definition of the

TABLE 1: Operation Tree for the CCSD Energy Equation

$$(\xi_1)^{h_6}_{p_5} = +f^{h_6}_{p_5} + \frac{1}{2}t^{p_3}_{h_4}v^{h_4h_6}_{p_3p_5}$$

$$e = +t^{p_5}_{h_6}(\xi_1)^{h_6}_{p_5} + \frac{1}{4}t^{p_1p_2}_{h_3h_4}v^{h_3h_4}_{p_1p_2}$$

TABLE 2: Operation Tree for the CCSD T_1 Amplitude Equation

$$(\Xi_1)^{h_7}_{p_3} = -t^{p_5}_{h_6}v^{h_6h_7}_{p_3p_5}$$

$$(\xi_{22})^{h_7}_{p_3} = +f^{h_7}_{p_3} + (\Xi_1)^{h_7}_{p_3}$$

$$(\xi_2)^{h_7}_{h_1} = +f^{h_7}_{h_1} + t^{p_3}_{h_1}(\xi_{22})^{h_7}_{p_3} - t^{p_4}_{h_5}v^{h_5h_7}_{h_1p_4} - \frac{1}{2}t^{p_3p_4}_{h_1h_5}v^{h_5h_7}_{p_3p_4}$$

$$(\xi_3)^{p_2}_{p_3} = +f^{p_2}_{p_3} - t^{p_4}_{h_5}v^{h_5p_2}_{p_3p_4}$$

$$(\xi_5)^{h_8}_{p_7} = +f^{h_8}_{p_7} + (\Xi_1)^{h_8}_{p_7}$$

$$(\xi_6)^{h_4h_5}_{h_1p_3} = +v^{h_4h_5}_{h_1p_3} - t^{p_6}_{h_1}v^{h_4h_5}_{p_3p_6}$$

$$r^{p_2}_{h_1} = +f^{p_2}_{h_1} - t^{p_2}_{h_7}(\xi_2)^{h_7}_{h_1} + t^{p_3}_{h_1}(\xi_3)^{p_2}_{p_3} - t^{p_3}_{h_4}v^{h_4p_2}_{h_1p_3} + t^{p_2p_7}_{h_1h_8}(\xi_5)^{h_8}_{p_7} - \frac{1}{2}t^{p_2p_3}_{h_4h_5}(\xi_6)^{h_4h_5}_{h_1p_3} - \frac{1}{2}t^{p_3p_4}_{h_1h_5}v^{h_5p_2}_{p_3p_4}$$

TABLE 3: Operation Tree for the CCSD T_2 Amplitude Equation

$$(\Xi_1)_{h_1p_9}^{h_7h_{10}} = +t_{h_1}^{p_5} v_{p_5p_9}^{h_7h_{10}}$$
$$(\Xi_2)_{h_1h_2}^{h_{10}h_{11}} = -\frac{1}{2} t_{h_1h_2}^{p_7p_8} v_{p_7p_8}^{h_{10}h_{11}}$$
$$(\Xi_3)_{h_1p_5}^{h_{10}p_3} = -\frac{1}{2} t_{h_1}^{p_6} v_{p_5p_6}^{h_{10}p_3}$$
$$(\Xi_4)_{p_5}^{h_{10}} = -t_{h_7}^{p_6} v_{p_5p_6}^{h_7h_{10}}$$
$$(\xi_{222})_{h_1p_5}^{h_{10}h_{11}} = +v_{h_1p_5}^{h_{10}h_{11}} + \frac{1}{2}(\Xi_1)_{h_1p_5}^{h_{10}h_{11}}$$
$$(\xi_{22})_{h_1h_2}^{h_{10}h_{11}} = -v_{h_1h_2}^{h_{10}h_{11}} + P_2 t_{h_1}^{p_5} (\xi_{222})_{h_2p_5}^{h_{10}h_{11}} + (\Xi_2)_{h_1h_2}^{h_{10}h_{11}}$$
$$(\xi_{23})_{h_1p_5}^{h_{10}p_3} = +v_{h_1p_5}^{h_{10}p_3} + (\Xi_3)_{h_1p_5}^{h_{10}p_3}$$
$$(\xi_{24})_{p_5}^{h_{10}} = +f_{p_5}^{h_{10}} + (\Xi_4)_{p_5}^{h_{10}}$$
$$(\xi_{25})_{h_1p_9}^{h_7h_{10}} = +v_{h_1p_9}^{h_7h_{10}} + (\Xi_1)_{h_1p_9}^{h_7h_{10}}$$
$$(\xi_2)_{h_1h_2}^{h_{10}p_3} = +v_{h_1h_2}^{h_{10}p_3} + \frac{1}{2} t_{h_{11}}^{p_3} (\xi_{22})_{h_1h_2}^{h_{10}h_{11}} - P_2 t_{h_1}^{p_5} (\xi_{23})_{h_2p_5}^{h_{10}p_3}$$
$$- t_{h_1h_2}^{p_3p_5} (\xi_{24})_{p_5}^{h_{10}} + P_2 t_{h_1h_7}^{p_3p_9} (\xi_{25})_{h_2p_9}^{h_7h_{10}} + \frac{1}{2} t_{h_1h_2}^{p_5p_6} v_{p_5p_6}^{h_{10}p_3}$$
$$(\xi_3)_{h_1p_5}^{p_3p_4} = +v_{h_1p_5}^{p_3p_4} - \frac{1}{2} t_{h_1}^{p_6} v_{p_5p_6}^{p_3p_4}$$
$$(\xi_{42})_{p_8}^{h_9} = +f_{p_8}^{h_9} + (\Xi_4)_{p_8}^{h_9}$$
$$(\xi_4)_{h_1}^{h_9} = +f_{h_1}^{h_9} + t_{h_1}^{p_8} (\xi_{42})_{p_8}^{h_9} - t_{h_7}^{p_6} v_{h_1p_6}^{h_7h_9} - \frac{1}{2} t_{h_1h_8}^{p_6p_7} v_{p_6p_7}^{h_8h_9}$$
$$(\xi_5)_{p_5}^{p_3} = +f_{p_5}^{p_3} - t_{h_7}^{p_6} v_{p_5p_6}^{h_7p_3} - \frac{1}{2} t_{h_7h_8}^{p_3p_6} v_{p_5p_6}^{h_7h_8}$$
$$(\xi_{62})_{h_1p_8}^{h_9h_{11}} = +v_{h_1p_8}^{h_9h_{11}} + \frac{1}{2}(\Xi_1)_{h_1p_8}^{h_9h_{11}}$$
$$(\xi_6)_{h_1h_2}^{h_9h_{11}} = -v_{h_1h_2}^{h_9h_{11}} + P_2 t_{h_1}^{p_8} (\xi_{62})_{h_2p_8}^{h_9h_{11}} + (\Xi_2)_{h_1h_2}^{h_9h_{11}}$$
$$(\xi_7)_{h_1p_5}^{h_6p_3} = +v_{h_1p_5}^{h_6p_3} + 2(\Xi_3)_{h_1p_5}^{h_6p_3} - \frac{1}{2} t_{h_1h_8}^{p_3p_7} v_{p_5p_7}^{h_6h_8}$$
$$r_{h_1h_2}^{p_3p_4} = +v_{h_1h_2}^{p_3p_4} - P_2 t_{h_{10}}^{p_3} (\xi_2)_{h_1h_2}^{h_{10}p_4} - P_2 t_{h_1}^{p_5} (\xi_3)_{h_2p_5}^{p_3p_4} - P_2 t_{h_1h_9}^{p_3p_4} (\xi_4)_{h_2}^{h_9}$$
$$+ P_2 t_{h_1h_2}^{p_3p_5} (\xi_5)_{p_5}^{p_4} - \frac{1}{2} t_{h_9h_{11}}^{p_3p_4} (\xi_6)_{h_1h_2}^{h_9h_{11}} - P_4 t_{h_1h_6}^{p_3p_5} (\xi_7)_{h_2p_5}^{h_6p_4} + \frac{1}{2} t_{h_1h_2}^{p_5p_6} v_{p_5p_6}^{p_3p_4}$$

persistent intermediate tensors is attached to the tree as an auxiliary data structure.

2.3. Synthesis of a Computer Program. To minimize the number of arithmetic operations and the amount of storage space by the use of index permutation symmetry, we need to know how an intermediate tensor transforms upon the permutation of indices. There does not seem to be a simple answer to this question in general situations, but under the conditions[23] that (1) there is no deexcitation amplitude tensor, (2) there is not more than one integral tensor (a tensor that originates from the Hamiltonian operator), (3) the strength reduction seeks an operation minimal contraction sequence for an individual multiple tensor contraction, and (4) an overall permutation operator is factorized in the manner prescribed above, we can write an intermediate tensor in the following general form:[13]

$$\xi_{e_{a+1}<\cdots<e_b, i_c+1<\cdots<i_d}^{e_1<\cdots<e_a, i_1<\cdots<i_c} \tag{25}$$

Here the indices of the intermediate tensor are categorized into four disjointed sets (i.e., contravariant external indices $e_1 < \cdots < e_a$ (the superscript indices that appear in the output tensor of the multiple tensor contraction), covariant external indices $e_{a+1} < \cdots < e_b$, contravariant internal indices $i_1 < \cdots < i_c$ (the superscript indices that are not external indices), and covariant internal indices $i_{c+1} < \cdots < i_d$), and any pair of indices in each category is permutable. Input tensors (excitation amplitude tensors, integral tensors, etc.) and output tensors can also be cast in this form.

This general form of an intermediate tensor can be deduced by examining the process of a binary tensor contraction that takes place under the above listed conditions. The conditions preclude any contraction except for that between an excitation amplitude tensor and an integral tensor and that between an excitation amplitude tensor and an intermediate tensor. Because an integral tensor is encompassed by eq 25, we can confine our analysis to the binary tensor contraction of the form

$$P \xi_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e<c_{e+1}<\cdots<c_f}^{e_1<\cdots<e_a, i_1<\cdots<i_c<c_{c+1}<\cdots<c_d} t_{e_{g+1}<\cdots<e_h<c_c+1<\cdots<c_d}^{e_{b+1}<\cdots<e_g<c_{e+1}<\cdots<c_f} \tag{26}$$

where P is a sum of permutation operators that makes the tensor it acts upon antisymmetric to the interchange of any pair of contravariant or covariant external indices. Notice that the common (summation) indices are among the internal indices of the intermediate tensor ξ and that an index of the excitation amplitude tensor t is either a common index or an external index. The contraction of eq 26 is performed in the following three distinct steps. First, we "decompress" the tensors by lifting some of the restrictions on the index ranges

$$\xi_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e<c_{e+1}<\cdots<c_f}^{e_1<\cdots<e_a, i_1<\cdots<i_c<c_{c+1}<\cdots<c_d} \rightarrow \xi_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e, c_{e+1}<\cdots<c_f}^{e_1<\cdots<e_a, i_1<\cdots<i_c, c_{c+1}<\cdots<c_d} \tag{27}$$

$$t_{e_{g+1}<\cdots<e_h<c_{c+1}<\cdots<c_d}^{e_{b+1}<\cdots<e_g<c_{e+1}<\cdots<c_f} \rightarrow t_{e_{g+1}<\cdots<e_h, c_{e+1}<\cdots<c_d}^{e_{b+1}<\cdots<e_g, c_{e+1}<\cdots<c_f} \tag{28}$$

such that the common indices can vary independently from the other indices. The restricted ranges among the common indices, however, merely give rise to a scalar factor of $(d - c)!(f - e)!$. Second, we carry out the summation over the common indices and obtain

$$\eta_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e, e_{g+1}<\cdots<e_h}^{e_1<\cdots<e_a, i_1<\cdots<i_c, e_{b+1}<\cdots<e_g} = (d - c)!(f - e)!$$
$$\xi_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e, c_{e+1}<\cdots<c_f}^{e_1<\cdots<e_a, i_1<\cdots<i_c, c_{c+1}<\cdots<c_d} t_{e_{g+1}<\cdots<e_h, c_{c+1}<\cdots<c_d}^{e_{b+1}<\cdots<e_g, c_{e+1}<\cdots<c_f} \tag{29}$$

Third, we "compress" the intermediate η by introducing the restrictions on the index ranges with the aid of the permutation operator

$$P \eta_{e_{a+1}<\cdots<e_b, i_{d+1}<\cdots<i_e, e_{g+1}<\cdots<e_h}^{e_1<\cdots<e_a, i_1<\cdots<i_c, e_{b+1}<\cdots<e_g} \rightarrow$$
$$\eta_{e_{a+1}<\cdots<e_b<e_{g+1}<\cdots<e_h, i_{d+1}<\cdots<i_e}^{e_1<\cdots<e_a<e_{b+1}<\cdots<e_g, i_1<\cdots<i_c} \tag{30}$$

The intermediate η consequently recovers the general form of intermediate tensors as expressed by eq 25.

In addition to the index permutation symmetry, we must also take account of spin (within the spin−orbital formalisms) and the real Abelian point-group symmetry of orbitals (i.e., tensor indices) simultaneously to minimize the number of arithmetic operations and storage space. This amounts to exploiting the fact that the tensor elements are nonvanishing only when the following conditions are satisfied:

$$\sum_g^{\text{covariant}} \sigma_g = \sum_g^{\text{contravariant}} \sigma_g \tag{31}$$

$$\prod_g^{\text{covariant}} \Gamma_g = \prod_g^{\text{contravariant}} \Gamma_g \tag{32}$$

where σ_g and Γ_g are the spin momentum and irreducible representation, respectively, associated with tensor index g.

```
 1 PARALLELIZED LOOP over p4t, p5t ≤ p6t, h1t ≤ h2t ≤ h3t
 2   IF (X is nonzero by spin and spatial symmetry) THEN
 3     LOOP over p7t ≤ p8t
 4       IF (V is nonzero by spin and spatial symmetry) THEN
 5         IF (p8t < p4t) GET +T(p7t,p8t,p4t,h1t,h2t,h3t)
 6         IF (p7t < p4t) & (p4t ≤ p8t) GET -T(p7t,p4t,p8t,h1t,h2t,h3t)
 7         IF (p4t ≤ p7t) GET +T(p4t,p7t,p8t,h1t,h2t,h3t)
 8         SORT to T′(p7t,p8t,p4t,h1t,h2t,h3t)
 9         GET +V(p5t,p6t,p7t,p8t)
10         SORT to V′(p7t,p8t,p5t,p6t)
11         IF (p8t = p7t) factor = 0.5
12         IF (p8t ≠ p7t) factor = 1.0
13         X′(p4t,h1t,h2t,h3t,p5t,p6t) = factor × T′× V′
14         IF (p4t ≤ p5t) SORT & ADD +X(p4t,p5t,p6t,h1t,h2t,h3t)
15         IF (p5t ≤ p4t) & (p4t ≤ p6t) SORT & ADD -X(p5t,p4t,p6t,h1t,h2t,h3t)
16         IF (p6t ≤ p4t) SORT & ADD +X(p5t,p6t,p4t,h1t,h2t,h3t)
17       END IF
18     END LOOP
19   END IF
20 END LOOP
21 SYNCHRONIZE
```

Figure 1. Overview of a subroutine automatically generated by TCE for tensor contraction $\chi^{p_4p_5p_6}_{h_1h_2h_3} = {}^1/_2 P_3 t^{p_4p_7p_8}_{h_1h_2h_3} v^{p_5p_6}_{p_7p_8}$. Sorted and unsorted tiles of tensors are distinguished by primes.

Executing conditionals 31 and 32 or imposing the index-range restrictions element by element in the process of tensor contraction is inefficient owing to the substantial overhead of having IF statements in a performance-critical section (i.e., matrix multiplications and matrix sorts) of the calculation. Alternatively, one might decompress tensors (cf. eqs 27 and 28) and rearrange their elements according to the spin and spatial symmetry to expose a long uninterrupted sequence of arithmetic operations. However, such an algorithm tends to require a large memory space to accommodate the decompressed tensor that contains many redundant elements. To overcome this algorithmic dilemma, we resort to index-range tiling (also called index-range blocking) that partitions indices into tiles (blocks) with each tile consisting of indices with the same hole or particle type, spin symmetry, and spatial symmetry, and we exploit these symmetries at the tile level (rather than at the element level).

For example, an excitation amplitude tensor $t^{p_4<p_5<p_6}_{h_1<h_2<h_3}$ is stored as $t^{p_{4t}\le p_{5t}\le p_{6t}}_{h_{1t}\le h_{2t}\le h_{3t}}$ where h_{1t} (p_{4t}),... is a group of hole (particle) indices sharing the same spin and spatial symmetry symbol. The range restrictions imposed on the tile indices to a large extent minimize the storage space for the tensor by the use of index permutation symmetry. Redundant elements occur only within the "diagonal" tiles that are characterized by equalities in the tile index-range restrictions. The proportion of redundant elements diminishes as the tile size decreases, and it vanishes in the extreme where each tile consists of just one index. Conditionals 31 and 32 can be executed only once for each tile to decide whether the tile is zero. Therefore, the tiling allows the conditionals associated with spin, spatial, and index permutation symmetry to be moved to outside of the performance-critical section of the calculation, and it also permits these conditionals to map the tiles of a tensor to their symmetrically unique tiles and to retrieve (store) them directly from (to) storage without explicitly decompressing (compressing) the tensor. Furthermore, the tiling also exposes an adequate granularity of parallelism (the tile-level matrix multiplications and matrix element sorts can be performed in parallel) and offers a means to control the peak memory usage by adjusting the tile sizes at run time.

Figure 1 illustrates a program automatically synthesized by TCE for tensor contraction $\chi^{p_4p_5p_6}_{h_1h_2h_3} = {}^1/_2 P_3 t^{p_4p_7p_8}_{h_1h_2h_3} v^{p_5p_6}_{p_7p_8}$. The nested outer loops (line 1) run over the tiles of tensor χ with the tile range restrictions $p_{5t} \le p_{6t}$ and $h_{1t} \le h_{2t} \le h_{3t}$. Although tensor χ is stored with the full restrictions of $p_{4t} \le p_{5t} \le p_{6t}$ and $h_{1t} \le h_{2t} \le h_{3t}$, we cannot impose $p_{4t} \le p_{5t}$ in the contraction stage because $-t^{p_{5t}p_{7t}\le p_{8t}}_{h_{1t}\le h_{2t}\le h_{3t}} v^{p_{4t}\le p_{6t}}_{p_{7t}\le p_{8t}}$ and $t^{p_{6t}p_{7t}\le p_{8t}}_{h_{1t}\le h_{2t}\le h_{3t}} v^{p_{4t}\le p_{5t}}_{p_{7t}\le p_{8t}}$ as well as $t^{p_{4t}p_{7t}\le p_{8t}}_{h_{1t}\le h_{2t}\le h_{3t}} v^{p_{5t}\le p_{6t}}_{p_{7t}\le p_{8t}}$ contribute to $\chi^{p_{4t}\le p_{5t}\le p_{6t}}_{h_{1t}\le h_{2t}\le h_{3t}}$ by virtue of the permutation operator. Immediately within the loops, we execute a conditional of spin and spatial symmetry to know whether the tile of tensor χ has any nonvanishing element (line 2). When the tile is nonzero, the operation enters the inner nested loops over the common indices $p_{7t} \le p_{8t}$ (line 3), which are followed by a conditional (line 4) to test whether the tile of tensor v is nonzero by spin and spatial symmetry (when the tiles of χ and v are nonzero, the tile of t is also nonzero). At this stage, we retrieve the tiles of tensors t and v from storage. Because the loop indices p_{4t}, p_{7t}, and p_{8t} do not conform to the full tile index-range restrictions of tensor t, the tile associated with the loop indices may not be located in storage. In that case, the nonoverlapping conditionals in lines 5−7 map the tile to an equivalent tile (apart from its parity) in storage. For tensor v, the tile can be obtained directly from storage (line 9) because the restrictions on the loop indices $p_{5t} \le p_{6t}$ and $p_{7t} \le p_{8t}$ coincide with those of the indices of tensor v in storage. Subsequently, the arrays containing the tiles of t and v are sorted (lines 8 and 10) such that the loop over a composite common index has unit stride in the tile-level tensor contraction (matrix multiplication) in line 13. The resulting tile of tensor χ is multiplied by an extra factor of 2 when $p_{7t} < p_{8t}$ to compensate the bypassed common index ranges of $p_{7t} > p_{8t}$ (lines 11 and 12). Finally, the tile of tensor χ is mapped to a symmetrically unique tile and is added to storage (lines 14−16). Note that this mapping corresponds to the reverse operation of the canonicalized permutation and also that the conditionals (lines

```
1 CALL OFFSET_ccsd_e_1_1(file_i1,offset_i1,size_i1)
2 CALL FILENAME('ccsd_e_1_1_i1',filename)
3 CALL CREATEFILE(filename,file_i1,size_i1)
4 CALL ccsd_e_1_1(file_f1,offset_f1,file_i1,offset_i1)
5 CALL ccsd_e_1_2(file_t1,offset_t1,file_v2,offset_v2,file_i1,offset_i1)
6 CALL RECONCILEFILE(file_i1,size_i1)
7 CALL ccsd_e_1(file_t1,offset_t1,file_i1,offset_i1,file_i0,offset_i0)
8 CALL DELETEFILE(file_i1)
9 CALL ccsd_e_2(file_t2,offset_t2,file_v2,offset_v2,file_i0,offset_i0)
```

Figure 2. Caller subroutine for the CCSD energy equation generated by TCE.

14−16) overlap with one another because the permutation applies to diagonal tiles also.

The nested outer loops (line 1) are parallelized, and the computational workload (the tile retrieval, tile-level tensor sorts and contractions, and tile accumulation) is balanced between processors by dynamical workload allocation. Therefore, each processor performs asynchronously the workload dynamically assigned to it within its local memory space that accommodates the tiles of the tensors, and synchronization between processors occurs only at the conclusion of the tensor contraction process (line 21). Dynamical load balancing is essential because tile sizes can be significantly different from one another and it is hard to distribute the workload statically evenly across processors within the present algorithmic framework. Although they are inconspicuous, interprocessor interactions exists in the tile retrieval and accumulation processes as overlapping I/O operations to the tensor storage.

TCE generates subroutines (Figure 1) for unary tensor substitution and binary tensor contraction and also caller subroutines (Figure 2). We have interfaced these subroutines with the computational chemistry program suite UTCHEM[24] (in Fortran90) for sequential executions and with NWCHEM[25] (in Fortran77) for sequential and parallel executions on the basis of the following parallel I/O schemes: (1) a shared file algorithm based on a global file system using the shared file library,[26] (2) a replicated file algorithm based on a distributed or global file system using the exclusive access file library,[26] and (3) an incore algorithm using the global array library.[26] In the shared file algorithm, each tensor is stored in a file on a global file system accessed by all processors, whereas in the replicated algorithm each processor has its own copy of a file and updates the copy asynchronously. In the latter algorithm, the copies of the file must be reconciled at the end of tensor contraction so that each copy contains a complete intermediate tensor. In the incore algorithm, a global memory space residing across all computer nodes is employed in lieu of a file system, and hence I/O operations are replaced by interprocessor communications.

Tensors are stored as a 1D array of symmetrically unique tiles, and the elements of each tile are stored consecutively. The positions (offsets) of the tiles in the storage are precomputed and stored in memory for each intermediate tensor by a subroutine generated by TCE. Figure 2 illustrates a sequence of operations for evaluating the CCSD energy equation (Table 1). In lines 1−3, the program precomputes the offsets, size, and name of a file that reserves the storage space for intermediate ξ_1. Subsequently, it issues subroutine calls for unary tensor substitution $(\xi_1)_{p_5}^{h_6} = f_{p_5}^{h_6}$ (line 4) ($f_{p_5}^{h_6}$ denotes a Fock matrix element) and binary tensor contraction $(\xi_1)_{p_5}^{h_6} = (\xi_1)_{p_5}^{h_6} + {}^1/_2 t_{h_4}^{p_3} v_{p_3 p_5}^{h_4 h_6}$ (line 5). When a replicated file algorithm is employed, tensor ξ_1, written in fragment in exclusive access files, is consolidated into a complete tensor, which is then broadcast to each processor (line 6). Tensor ξ_1 is used as one of the input tensors in the subsequent tensor contraction $e = t_{h_6}^{p_5}(\xi_1)_{p_5}^{h_6}$ (line 7) and is destroyed immediately after its use (line 8).

3. Applications of the Tensor Contraction Engine

As an initial application, we have employed TCE to derive the equations of various models of CI, CC, MBPT automatically and implement them into parallel computer programs within the computational chemistry program suites of NWCHEM[25] and UTCHEM.[24] The models that have been implemented were spin-unrestricted CISD, CISDT, and CISDTQ, spin-unrestricted CCD, LCCD, CCSD, LCCSD, QCISD, CCSDT, and CCSDTQ, and noncanonical spin-unrestricted MBPT(2), MBPT(3), and MBPT(4). For details of these models, see refs 17−19.

The ansatz of CI theory (e.g., that of configuration-interaction single, double, and triple substitutions (CISDT)) may be given as

$$C = C_1 + C_2 + C_3 \tag{33}$$

$$E = Q_0 H_N (1 + C) Q_0 \tag{34}$$

$$0 = Q_1 H_N (1 + C) Q_0 - E Q_1 C Q_0 \tag{35}$$

$$0 = Q_2 H_N (1 + C) Q_0 - E Q_2 C Q_0 \tag{36}$$

$$0 = Q_3 H_N (1 + C) Q_0 - E Q_3 C Q_0 \tag{37}$$

where Q_n is a projection operator onto the manifold of n-tuply substituted determinants from the reference wave function, C_n is an n-fold excitation operator, H_N is the normal-ordered Hamiltonian operator, and E is the correlation energy. Equation 34 can be expressed in an explicit normal-ordered second-quantized form as

$$\begin{aligned} E = {} & + f_{g_2}^{g_1} c_{h_4}^{p_3} \langle 0 | \{ g_1^\dagger g_2 \} \{ p_3^\dagger h_4 \} | 0 \rangle \\ & + \frac{1}{4} v_{g_3 g_4}^{g_1 g_2} c_{h_6}^{p_5} \langle 0 | \{ g_1^\dagger g_2^\dagger g_4 g_3 \} \{ p_5^\dagger h_6 \} | 0 \rangle \\ & + \frac{1}{4} f_{g_2}^{g_1} c_{h_5 h_6}^{p_3 p_4} \langle 0 | \{ g_1^\dagger g_2 \} \{ p_3^\dagger p_4^\dagger h_6 h_5 \} | 0 \rangle \\ & + \frac{1}{16} v_{g_3 g_4}^{g_1 g_2} c_{h_7 h_8}^{p_5 p_6} \langle 0 | \{ g_1^\dagger g_2^\dagger g_4 g_3 \} \{ p_5^\dagger p_6^\dagger h_8 h_7 \} | 0 \rangle \\ & + \frac{1}{36} f_{g_2}^{g_1} c_{h_6 h_7 h_8}^{p_3 p_4 p_5} \langle 0 | \{ g_1^\dagger g_2 \} \{ p_3^\dagger p_4^\dagger p_5^\dagger h_8 h_7 h_6 \} | 0 \rangle \\ & + \frac{1}{144} v_{g_3 g_4}^{g_1 g_2} c_{h_8 h_9 h_{10}}^{p_5 p_6 p_7} \langle 0 | \{ g_1^\dagger g_2^\dagger g_4 g_3 \} \{ p_5^\dagger p_6^\dagger p_7^\dagger h_{10} h_9 h_8 \} | 0 \rangle \end{aligned} \tag{38}$$

Equations such as this can be parsed and processed by TCE, which subsequently synthesizes parallel computer programs. The synthesized programs can then be compiled and executed

without any manual modification once appropriate interface programs to a computational chemistry program package are in place.

The ansatz of CC theory such as the following (coupled-cluster singles and doubles or CCSD)

$$T = T_1 + T_2 \tag{39}$$

$$E = Q_0\left[H_N\left(1 + T + \frac{1}{2!}T^2\right)\right]_C Q_0 \tag{40}$$

$$0 = Q_1\left[H_N\left(1 + T + \frac{1}{2!}T^2 + \frac{1}{3!}T^3\right)\right]_C Q_0 \tag{41}$$

$$0 = Q_2\left[H_N\left(1 + T + \frac{1}{2!}T^2 + \frac{1}{3!}T^3 + \frac{1}{4!}T^4\right)\right]_C Q_0 \tag{42}$$

can be processed in the same manner by TCE except that immediately after the contraction step disconnected terms need to be eliminated ($[\cdots]_C$ means that all terms in the second-quantized form of H_N and T must be connected by common indices, and T_n is an n-fold excitation operator). The operation trees obtained from the CCSD ansatz are shown in Tables 1−3.

In this article, we confine ourselves to "noncanonical" or "generalized" MBPT that is invariant to unitary transformations within the just-occupied and just-virtual spaces without resorting to Wigner's $2n + 1$ rule. Noncanonical MBPT in its tensor formulation is considered to be the most fundamental representation of the theory[27] and is essential for linear scaling algorithms using local molecular orbitals.[14] As usual, we partition the Hamiltonian operator as a sum of the Fock operator F and the fluctuation potential V and apply Rayleigh−Schrödinger perturbation theory. Hence, the ansatz of second-order many-body perturbation theory [MBPT(2)] is

$$E^{(2)} = Q_0 V(T_1 + T_2)Q_0 \tag{43}$$

$$0 = Q_1 F(T_1 + T_2)Q_0 + Q_1 V Q_0 \tag{44}$$

$$0 = Q_2 F(T_1 + T_2)Q_0 + Q_2 V Q_0 \tag{45}$$

where T_1 and T_2 are the first-order single- and double-excitation operators, respectively. When evaluated by TCE, this ansatz leads to a tensor formulation of MBPT(2) (see also ref 14) that reads

$$E^{(2)} = +\frac{1}{4} t_{h_3h_4}^{p_1p_2} v_{p_1p_2}^{h_3h_4} \tag{46}$$

$$0 = -f_{h_9}^{h_2} t_{h_2}^{p_1} + f_{p_2}^{p_1} t_{h_9}^{p_2} + f_{p_3}^{h_2} t_{h_2h_9}^{p_3p_1} \tag{47}$$

$$0 = -(1 - P_{p_2p_1h_9h_{10}}^{p_1p_2h_9h_{10}} - P_{p_2p_1h_{10}h_9}^{p_2p_1h_9h_{10}} + P_{p_2p_1h_{10}h_9}^{p_1p_2h_9h_{10}}) f_{h_9}^{p_2} t_{h_{10}}^{p_1} - (1 - P_{p_1p_2h_{10}h_9}^{p_1p_2h_9h_{10}}) f_{h_9}^{h_3} t_{h_3h_{10}}^{p_1p_2} - (1 - P_{p_2p_1h_9h_{10}}^{p_1p_2h_9h_{10}}) f_{p_3}^{p_2} t_{h_9h_{10}}^{p_3p_1} + v_{h_9h_{10}}^{p_1p_2} \tag{48}$$

which may be solved iteratively. It may be noticed that because a Hartree−Fock (HF) reference wave function satisfies the Brillouin theorem ($f_{h_3}^{p_1} = f_{p_2}^{h_4} = 0$) T_1 is zero. Employing this fact, we can write the ansatz of MBPT(3) as

$$E^{(3)} = Q_0 V(U_1 + U_2)Q_0 \tag{49}$$

$$0 = Q_1 F(U_1 + U_2)Q_0 + Q_1 V T_2 Q_0 \tag{50}$$

$$0 = Q_2 F(U_1 + U_2)Q_0 + Q_2 V T_2 Q_0 \tag{51}$$

where U_1 and U_2 are the second-order single- and double-excitation operators. Provided this ansatz, TCE generates the following coupled equations:

$$E^{(3)} = +\frac{1}{4} u_{h_3h_4}^{p_1p_2} v_{p_1p_2}^{h_3h_4} \tag{52}$$

$$0 = -f_{h_9}^{h_2} u_{h_2}^{p_1} + f_{p_2}^{p_1} u_{h_9}^{p_2} + \frac{1}{2} t_{h_3h_4}^{p_2p_1} v_{h_9p_2}^{h_3h_4} + \frac{1}{2} t_{h_4h_9}^{p_2p_3} v_{p_2p_3}^{h_4p_1} \tag{53}$$

$$\begin{aligned} 0 = &-(1 - P_{p_1p_2h_{10}h_9}^{p_1p_2h_9h_{10}}) f_{h_9}^{h_3} u_{h_3h_{10}}^{p_1p_2} \\ &-(1 - P_{p_2p_1h_9h_{10}}^{p_1p_2h_9h_{10}}) f_{p_3}^{p_2} u_{h_9h_{10}}^{p_3p_1} + \frac{1}{2} t_{h_3h_4}^{p_1p_2} v_{h_9h_{10}}^{h_3h_4} \\ &+ (1 - P_{p_1p_2h_10h_9}^{p_2p_1h_{10}h_9} - P_{p_1p_2h_{10}h_9}^{p_1p_2h_9h_{10}} + P_{p_1p_2h_9h_{10}}^{p_2p_1h_{10}h_9}) t_{h_4h_{10}}^{p_3p_1} v_{h_9p_3}^{h_4p_2} \\ &+ \frac{1}{2} t_{h_9h_{10}}^{p_3p_4} v_{p_3p_4}^{p_1p_2} \end{aligned} \tag{54}$$

The unrestricted CI and CC models can in principle take any reference wave function such as a restricted HF (RHF), an unrestricted HF (UHF), a restricted open-shell HF (ROHF), and a restricted/unrestricted Kohn−Sham wave function, whereas the unrestricted MBPT models are based either on an RHF or a UHF reference wave function. When an RHF wave function is used as a reference of a closed-shell system, the programs generated by TCE map β orbitals to the α orbitals having the same spatial part and avoid any redundant computation and storage associated with the all-β spin-components of tensors. For example, among the three independent spin-components of the T_2 tensor, $t_{\alpha\alpha}^{\alpha\alpha}$, $t_{\alpha\beta}^{\alpha\beta}$, and $t_{\beta\beta}^{\beta\beta}$, the CCD program generated by TCE processes and stores only the $t_{\alpha\alpha}^{\alpha\alpha}$ and $t_{\alpha\beta}^{\alpha\beta}$ spin components for an RHF reference. (However, this must be distinguished from spin-adapted CCD, which deals with just the $t_{\alpha\beta}^{\alpha\beta}$ spin components.) The coupled equations of the CI, CC, and MBPT models are solved iteratively by a common driver subroutine that updates excitation amplitude tensors by a combination of the Jacobi iteration and the DIIS (direct inversion in the iterative subspace) extrapolation.

It may be instructive to compare the theoretical operation counts of the CCSD program automatically generated by TCE with an equivalent program hand-coded by a group of experts in the field. According to Stanton et al.,[28] the number of arithmetic operations of their unrestricted CCSD program is approximately $(^5/_4)O^2V^4 + 20O^3V^3 + (^5/_2)O^4V^2$ in the leading order, where $O \approx O_\alpha \approx O_\beta$ is the number of α- or β-spin occupied orbitals and $V \approx V_\alpha \approx V_\beta$ is the number of α- or β-spin virtual orbitals. The operation count of the TCE-generated CCSD code (Table 3) is approximately $(^5/_4)O^2V^4 + (^{45}/_2)O^3V^3 + (^{25}/_2)O^4V^2$. The prefactors of the O^2V^4 and O^3V^3 terms, which usually dominate the operation count because $V \gg O$ in most CCSD applications, are the same or only slightly greater in the TCE-generated program than in the program of Stanton et al. Therefore, the TCE-generated CCSD program can compete with, albeit not outperform, the equivalent, carefully hand-coded program. The greater operation count of the former than that of the latter is caused by the decoupling of the strength reduction and factorization processes in TCE.

Table 4 compares the CPU time for a single CC iteration performed on a Hewlett-Packard Longs Peak Linux cluster consisting of Intel Itanium-2 1-GHz dual processors. A comparison of the first two rows of the Table attests to the tremendous speedup of the CCSDT calculation brought about by the use of point-group symmetry. The speedup that is actually observed usually does not exceed 70% of the theoretical maximum speedup of h^2, with h being the order of the point

TABLE 4: CPU Time (min) for a Single Coupled-Cluster Iteration on 1, 2, 4, 8, and 16 Intel Itanium-2 1-GHz Dual Processors of a Hewlett-Packard Longs Peak Linux Cluster

				number of processors				
molecule	symmetry	theory	I/O	1	2	4	8	16
CH_2	C_1	CCSDT/cc-pVTZ	global array	6.9	3.7	2.0	1.5	1.1
CH_2	C_{2v}	CCSDT/cc-pVTZ	global array	1.6	0.63	0.37	0.24	0.17
CH_2	C_{2v}	CCSDT/cc-pVTZ	replicated	1.6	0.68	0.35	0.20	0.13
$C_{10}H_8^{+\,a}$	C_{2v}	CCSD/cc-pVDZ	replicated	7.2	4.2	2.6	2.1	1.7
$NC_4H_5^{+\,b}$	C_{2v}	CCSDT/cc-pVDZ	replicated	550	320	190	140	110

[a] The azulene radical cation. [b] The pyrrole radical cation.

group,[28] primarily because the partitioning of orbitals according to irreducible representations is irregular. We consider the speedup achieved by the TCE-generated program to be satisfactory.

Stanton et al.[28] also pointed out that point-group symmetry naturally subdivides a tensor contraction (matrix multiplication) into *h* independent operations that can be performed in parallel, although they did not actually parallelize their CCSD program. This is realized in the TCE-generated programs by virtue of the tiling algorithm and the dynamic load balancing scheme. The tiles can either coincide with symmetry blocks of orbitals or subdivide them and are therefore adjustable to available memory resources. The tile-level matrix multiplications and matrix sorts are performed in parallel within the local memory space of each processor with vectorized kernels (the DGEMM subroutine of the BLAS library). Although parallel performance falls off gradually as the number of processors increases, both the global array and replicated algorithms exhibit reasonable scalability for methylene. Expectedly, the replicated algorithm sustains slightly better parallel performance than the global array algorithm, owing to the greater communication overhead in the latter. The global array algorithm, however, is expected to be scalable with respect to problem size, but the replicated algorithm is not. If the fall off were primarily a consequence of small serial components within the CC iteration, then the scalability would improve for a more time-consuming calculation because the calculation would have an increased proportion of parallel components. The fact that the scalability does not improve for the azulene or pyrrole radical cation may therefore indicate that the overhead arising from the overlapping I/O operations is not negligible or the dynamical setting for the workload balancing is not entirely successful in decomposing the parallel portion of the problem evenly. It is not known at this moment whether stagnant scalability is inherent to many-electron theories that suffer from complex data dependency or if it can be overcome by an alternative parallel strategy that also maintains the universal applicability of TCE.

The results of the applications of the electron correlation methods implemented by TCE to the problem of the singlet−triplet separation of methylene are available in the Supporting Information.

4. Concluding Remarks

This article has described the symbolic manipulation program TCE that enables automatic derivation and program synthesis for general many-electron theories and has demonstrated its utility through its applications to the core theories (CI, CC, MBPT) of quantum chemistry.

The questions raised in the Introduction are answered as follows: (1) The canonicalization of expressions performed at various stages of the symbolic manipulation process ensures rapid pattern matching. (2, 5) Under certain conditions imposed on the ansatz of many-electron theories, we can assume that an intermediate tensor has two disjoint permutable sets of covariant indices and two disjoint permutable sets of contravariant indices. (3−6) The index-range tiling algorithm achieves the best compromise between the conflicting demand from the minimization of the number of arithmetic operations and the minimization of the memory requirement. It moves the conditionals to exploit spin, spatial, and index permutation symmetries to the suburb of the performance-critical section of the calculation without necessitating large temporary memory space to accommodate decompressed tensors. It also exposes an adequate granularity of parallelism and offers a means to control the peak memory usage by adjusting the tile sizes.

Despite its proven practical usefulness, the TCE program presented here is an initial prototype, and its capabilities are currently being actively enhanced in a multi-disciplinary, multi-institution project that involves quantum chemists and computer scientists. A list of enhancements that are being considered for TCE's program generator includes storage-space minimization via the loop fusion and space−time tradeoff techniques[21] and the development of a universal interface to various quantum chemistry software packages. Needless to say, TCE offers an expedient, exact, and perhaps even pedagogical means to derive and implement existing and new models of many-electron theories for the accurate theoretical treatment of electronic, magnetic, mechanical, and spectroscopic properties of atomic and molecular systems, which may include but are not limited to equation-of-motion CC and MBPT for excited states, relativistic correlation theories, analytical energy derivatives, combined CC and MBPT such as CCSD(T), and multireference CI, CC, and MBPT.

Acknowledgment. I thank the members of the TCE project—Professor Robert J. Harrison, Professor Marcel Nooijen, Dr. Alexander A. Auer, Dr. David E. Bernholdt, Dr. Venkatesh Choppella, Professor P. Sadayappan, Professor Gerald Baumgartner, Dr. Daniel Cociorva, Professor Russell Pitzer, and Professor J. Ramanujam—for valuable suggestions and encouragements throughout the study. I am indebted to Dr. Jarek Nieplocha and Dr. Theresa L. Windus for insightful discussions on the parallel I/O strategies, Dr. Takeshi Yanai and Professor Kimihiko Hirao for generously providing the UTCHEM program system, Dr. Michel Dupuis for a critical reading of the manuscript prior to publication, and Professor Rodney J. Bartlett for introducing me to advanced second-quantization techniques and diagrammatics for many-electron theories and many useful suggestions on the manuscript. This work has been funded by the U.S. Department of Energy, Office of Basic Energy Science and Office of Biological and Environmental Research, Office of Science under contract DE-AC06-76RLO 1830 with Battelle Memorial Institute. This work used the Molecular Science Computing Facility in the William R. Wiley Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, which has been supported by the Office of Biological and Environmental Research.

Supporting Information Available: The results of the applications of the electron correlation methods implemented by TCE to the problem of the singlet−triplet separation of methylene. This material is available free of charge via the Internet at http://pubs.acs.org.

References and Notes

(1) (a) Wong, H. C.; Paldus, J. *Comput. Phys. Commun.* **1973**, *6*, 1. (b) Wong, H. C.; Paldus, J. *Comput. Phys. Commun.* **1973**, *6*, 9.

(2) (a) Bartlett, R. J.; Purvis, G. D. *Int. J. Quantum Chem.* **1978**, *14*, 561. (b) Purvis, G. D., III; Bartlett, R. J. *J. Chem. Phys.* **1982**, *76*, 1910.

(3) Janssen, C. L.; Schaefer, H. F., III. *Theor. Chim. Acta* **1991**, *79*, 1.

(4) Kucharski, S. A.; Bartlett, R. J. *Theor. Chim. Acta* **1991**, *80*, 387.

(5) Li, X.; Paldus, J. *J. Chem. Phys.* **1994**, *101*, 8812.

(6) Harris, F. E. *Int. J. Quantum Chem.* **1999**, *75*, 593.

(7) (a) Nooijen M.; Lotrich, V. *J. Chem. Phys.* **2000**, *113*, 494. (b) Nooijen, M.; Lotrich, V. *J. Chem. Phys.* **2000**, *113*, 4549. (c) Nooijen, M.; Lotrich, V. *J. Mol. Struct.: THEOCHEM* **2001**, *547*, 253.

(8) Laidig, W. D.; Fitzgerald, G.; Bartlett, R. J. *Chem. Phys. Lett.* **1985**, *113*, 151.

(9) Knowles, P. J.; Somasundram, K.; Handy, N. C.; Hirao, K. *Chem. Phys. Lett.* **1985**, *113*, 8.

(10) Hirata, S.; Bartlett, R. J. *Chem. Phys. Lett.* **2000**, *321*, 216.

(11) Kállay, M.; Surján, P. R. *J. Chem. Phys.* **2000**, *113*, 1359.

(12) Olsen, J. *J. Chem. Phys.* **2000**, *113*, 7140.

(13) Kállay, M.; Surján, P. R. *J. Chem. Phys.* **2001**, *115*, 2945.

(14) Head-Gordon, M.; Maslen, P. E.; White, C. A. *J. Chem. Phys.* **1998**, *108*, 616.

(15) Hirata, S. *Tensor Contraction Engine: A Symbolic Manipulation Program for Quantum-Mechanical Many-Electron Theories*, version 1.0; Pacific Northwest National Laboratory: Richland, WA, 2002.

(16) Shavitt, I.; Bartlett, R. J. *Many-Body Methods in Quantum Chemistry*, unpublished.

(17) Bartlett, R. J. *Annu. Rev. Phys. Chem.* **1981**, *32*, 359.

(18) Bartlett, R. J.; Stanton, J. F. *Rev. Comput. Chem.* **1994**, *5*, 65.

(19) Bartlett, R. J. *Modern Electronic Structure Theory, Part II*; Yarkony, D. P., Ed.; World Scientific: Singapore, 1995; p 1047.

(20) It is assumed that the first binary contraction involves an integral tensor and that the subsequent contractions occur sequentially; ABCDE (with A being an integral tensor) may be evaluated as (((AB)C)D)E, (((AD)E)C)B... but not as (((BC)A)D)E, (AB)((CD)E)... This assumption is usually correct for an operation cost minimal solution to the strength reduction of an individual multiple tensor contraction, when the conditions discussed in section 2.3 are met.

(21) Baumgartner, G.; Bernholdt, D. E.; Cociorva, D.; Harrison, R.; Hirata, S.; Lam, C.-C.; Nooijen, M.; Pitzer, R.; Ramanujam, J.; Sadayappan, P. *Proceedings of Supercomputing 2002*; Baltimore, MD, 2002.

(22) Dowd, K.; Severance, C. *High Performance Computing*, 2nd ed.; O'Reilly & Associates: Sebastopol, CA, 1998.

(23) The conditions imposed are readily satisfied by CC and CI theories, but they may become a severe constraint on MBPT, in which a deexcitation tensor enters the formalisms when Wigner's $2n + 1$ rule is invoked. It is possible to relax these conditions at the expense of some of the restrictions in the index ranges or the flexibility in the strength reduction. This issue is deferred for future investigations and will not be discussed any further in this article.

(24) Yanai, T.; et al. *UTChem: A Program Suite for Ab Initio Electronic Structure Theories, A Development Version*; University of Tokyo: Tokyo, Japan, 2002.

(25) Harrison, R. J.; et al. *NWChem: A Computational Chemistry Package for Parallel Computers, A Development Version*; Pacific Northwest National Laboratory: Richland, WA, 2002.

(26) Nieplocha, J.; et al. *ParSoft: Parallel Computing Libraries and Tools Software, A Development Version*; Pacific Northwest National Laboratory: Richland, WA, 2002.

(27) Lauderdale, W. J.; Stanton, J. F.; Gauss, J.; Watts, J. D.; Bartlett, R. J. *Chem. Phys. Lett.* **1991**, *187*, 21.

(28) Stanton, J. F.; Gauss, J.; Watts, J. D.; Bartlett, R. J. *J. Chem. Phys.* **1991**, *94*, 4334.