

Novel strategy to implement active-space coupled-cluster methods

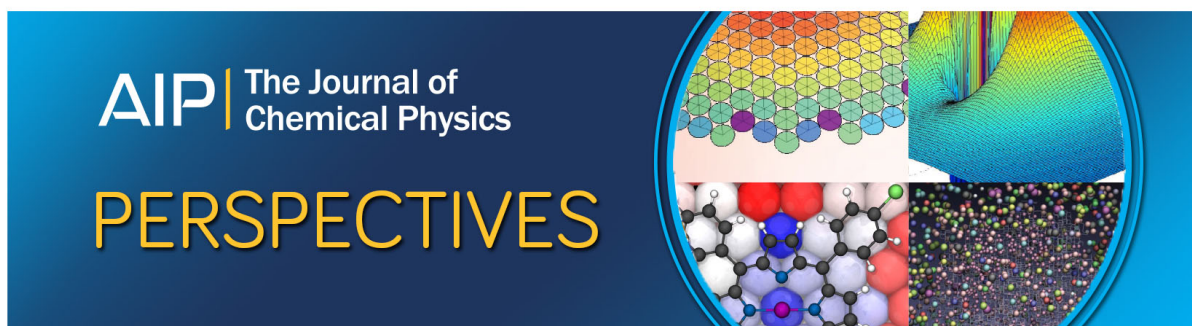
Zoltán Rolik, and Mihály Kállay

Citation: *The Journal of Chemical Physics* **148**, 124108 (2018); doi: 10.1063/1.5004971

View online: <https://doi.org/10.1063/1.5004971>

View Table of Contents: <http://aip.scitation.org/toc/jcp/148/12>

Published by the *American Institute of Physics*



Novel strategy to implement active-space coupled-cluster methods

Zoltán Rolik and Mihály Kállay

MTA-BME “Lendület” Quantum Chemistry Research Group, Department of Physical Chemistry and Materials Science, Budapest University of Technology and Economics, H-1521 Budapest, Hungary

(Received 15 September 2017; accepted 5 March 2018; published online 26 March 2018)

A new approach is presented for the efficient implementation of coupled-cluster (CC) methods including higher excitations based on a molecular orbital space partitioned into active and inactive orbitals. In the new framework, the string representation of amplitudes and intermediates is used as long as it is beneficial, but the contractions are evaluated as matrix products. Using a new diagrammatic technique, the CC equations are represented in a compact form due to the string notations we introduced. As an application of these ideas, a new automated implementation of the single-reference-based multi-reference CC equations is presented for arbitrary excitation levels. The new program can be considered as an improvement over the previous implementations in many respects; e.g., diagram contributions are evaluated by efficient vectorized subroutines. Timings for test calculations for various complete active-space problems are presented. As an application of the new code, the weak interactions in the Be dimer were studied. *Published by AIP Publishing.* <https://doi.org/10.1063/1.5004971>

I. INTRODUCTION

Since the coupled-cluster (CC) method was first applied in quantum chemistry in the pioneering work of Čížek,^{1,2} it has become the principal tool for electron correlation calculations. After the implementation and the initial applications of the CC with singles and doubles (CCSD)³ method, the importance of higher excitations became clear soon and the CC with singles, doubles, and triples (CCSDT)^{4,5} and with quadruples (CCSDTQ)^{6–8} methods were formulated and implemented.

Although the higher-order CC corrections are able to describe molecular systems with multi-reference (MR) character, for efficient description, usually MRCC methods are needed. In spite of the many attempts, the proper low-cost generalization of the single-reference CC (SRCC) models to the MR case is still an open problem.

Most of the MRCC studies published recently focus on state-selective approaches due to the ability to avoid the intruder state problem. Some of these methods are based on the Jeziorski–Monkhorst ansatz⁹ (JM), where a set of cluster amplitudes is assigned to each model space determinant. Typical examples for the JM-based MRCC methods are the Brillouin–Wigner CC theory, the MRexpT, or Mukherjee’s state-specific MRCC (SSMRCC) theory, where the first is not extensive, the second is only core-extensive, and the third is inaccurate and suffers from numerical instability. Recently, the most actively developed branch of the MRCC models is the internally contracted MRCC methods,^{10–18} where the interacting subspace is defined by excitation operators with respect to the contracted MR reference function. Naturally, the resulting functions are not orthogonal and generally form a redundant basis set. This difficulty is the source of the complexity of these approaches, which leads to theories where the efficient implementation is hard, the working equations must be truncated after a certain power (second or third) of cluster

operators, and due to the numerical elimination of redundancies, the calculated potential energy surfaces (PESs) may contain discontinuities.

For us the most important class of the MRCC approaches is that of the active-space CC models, where the application of restricted higher-order cluster amplitudes is the characteristic feature. These models use a CAS space to select the orbitals essential to describe the MR attribute of the wave function. The prototype of such methods is the single-reference-based multi-reference coupled-cluster method (SRMRCC), which was originally proposed by Oliphant and Adamowicz,⁸ further developed by Piecuch *et al.*,¹⁹ and fully implemented by us.²⁰ As it is well-known, there are two drawbacks of the SRMRCC method which restricts its applicability. First, the SRMRCC energy depends on the choice of the reference determinant. For those cases where the leading determinant of the wave function is changing, one gets discontinuity on the PES. Alternatively, smooth PES’s can be obtained using a spin and symmetry breaking principal determinant, where, however, the slow convergence of the CC equations can cause additional difficulty. Second, it is computationally demanding due to the high excitation levels in the cluster operator. We addressed the first problem in a previous paper,²¹ where a quasi-particle-based formalism was presented (QMRCC). This approach reduces the problem as the reference function is a multi-reference CAS space function, but the definition of the excitation space still remains reference function dependent. In this paper, we address the second problem and a novel implementation of the SRMRCC method is presented. The motivation behind the new implementation is partly to test the new diagrammatic approach presented in this paper and also to develop a proper bases which will be suitable for the implementation of the quasiparticle-based MRCC model. In addition, utilizing the new framework a novel implementation of high-order CC methods is also considered.

Due to the highly-excited cluster amplitudes of the active-space approaches, the efficient implementation of these methods is challenging. The arising problems are common in many points with that of the high-order SRCC^{22–25} implementations, but in some respect—because of the heterogeneous nature of tensor indices—they are more severe. The common points are the application of tensors with high ranks, the difficult optimization and implementation of equations due to the large number of terms, the index reordering steps needed for the efficient calculation of the contractions, etc. To tackle with the large number of terms, automated derivation of working equations^{26,27} and code generation^{28–33} have become a standard tool of quantum chemistry for a long time ago.³⁴ From our point of view, the work of Engels-Putzka and Hanrath²⁵ is especially significant, who presented an antisymmetric tensor contraction engine where the diagram contractions were converted into matrix products, and it was applied to the implementation of general-order SRCC methods. They also introduced a cost minimal factorization algorithm³⁵ for the derivation of the CC equations at arbitrary excitation level. We should also mention the Tensor Contraction Engine symbolic computing system (TCE) of Hirata, which automates both the equation derivation and the efficient implementation of high-order CC methods and many other models.^{36–39} A determinant-based implementation of the general CC equations was also published by Olsen,²² which could not achieve the correct scaling of the computation time with the system size. This implementation was applied to test the general active-space (GAS) CC method. Later Sørensen *et al.* published an optimal scaling general-order CC implementation,^{40,41} which is also applicable in the four-component relativistic framework and with the GAS CC concept. This implementation used a commutator-based algorithm for the evaluation of the Baker-Cambell-Hausdorff expansion of the CC equations. Lyakh *et al.* also published a diagram-based SRMRCC implementation,⁴² where both the derivation of the equations and the code generation is automated. This code was used for demonstrative purposes and not fully optimized.

For the efficient evaluation of the CC diagrams, sophisticated tensor contraction routines are needed. In the recent years, significant progress has been made to develop general-order efficient tensor contraction libraries,^{36,43–48} which are suitable for the implementation of the post-Hartree-Fock methods. We have already mentioned the TCE and Hanrath's implementations, but other highly efficient general purpose tensor contraction libraries have been also published, such as the libtensor tensor algebra library,⁴³ which have been designed for shared-memory systems, the TiledArray block-sparse tensor framework,⁴⁴ and the Cyclops Tensor Framework,⁴⁵ which support general tensor sparsity and distributed-memory systems.

The SRMRCC program, implemented previously by one of us,²⁰ was developed using the string-based general-order CC technique.²⁴ The basic elements of the implementation are the strings, that is, ordered lists of one-particle indices, which can be used to address the independent elements of a high-rank antisymmetric tensor in a compact form. As the contraction of arbitrary antisymmetric tensors can be formulated

in the string framework, it provides a proper background for the implementation of the CC equations for general excitation levels.

Although the novel implementation of SRMRCC presented here varies from the previous one²⁰ (here and after called previous MRCC⁴⁹ code) at several points, it keeps the string representation of operators and amplitudes where it is practical. Derivation of equations was based on a diagrammatic approach^{24,26}—as in the previous version—rather than an algebraic derivation using the second-quantized formalism.^{25,36} The main difference is that in the new approach, the strings are used to describe the active fixed indices only, but the contraction lines and the inactive open lines are labeled by one-particle indices. This allows us to formulate the tensor contractions in a vectorized form and perform the contractions more efficiently using the elements of the Basic Linear Algebra Subprograms (BLAS) package, as it was found advantageous in previous high-order CC implementations.^{25,36,41} The CC equations are formulated in a compact form, which enables the use of a dedicated subroutine for each contraction pattern instead of using a general one. This supports a more efficient implementation, as subroutines can be independently improved. Due to this relative compactness, the SRMRCC equations are also hard-coded for arbitrary excitation levels using a code generator computer program.

To illustrate the applicability of the new code, we used it to study the weak interaction of two Be atoms. It has been shown for a long time ago^{50–52} that, for the correct description of this system, application of large basis sets and the proper description of the dynamical electron correlation are essential. As it is demonstrated in this paper, the PES of Be₂ can be correctly described by active-space CC methods, and an accurate estimate for the correlation contribution of the core electrons to the dissociation energy is also presented.

The paper is organized as follows. After the theory of active-space CC models is briefly summarized in Sec. II, a proper ordering of amplitude and operator indices is introduced, and the corresponding diagram representation is presented in Sec. III A. These diagrams have string lines, and for the proper algebraic interpretation the diagram rules need some extension, which is presented in Sec. III B. The derivation of the corresponding equations, the implementation of the subroutines for the evaluation of the tensor contractions, and the main elements and the loop structures of the new code are shown in Secs. III C, III D, and III E. In Sec. IV, timings for test calculations are presented, and the performance is compared with that of the previous implementation. In Sec. V, the results for the Be dimer are reported. The paper is closed by the summary in Sec. VI.

II. ACTIVE-SPACE CC THEORIES

The qualitatively accurate description of a molecular system with MR character is most commonly obtained from a CAS reference. At the singles and doubles truncation level, highly excited amplitudes can appear in the active-space CC methods where—in addition to the at most two inactive occupied and virtual indices—the amplitudes can hold labels restricted to the active orbitals. In the SRMRCC theory, the

number of active indices is limited only by the number of active orbitals. To significantly reduce the calculation cost, approximate methods were developed, such as the CCSD with active triples⁵³ (CCSDt) and CCSD with active triples and quadruples (CCSDtq) methods,^{54,55} where the overall excitation level is restricted to three and four, respectively. For these methods, the completely renormalized (CR) CC methods were generalized^{56–58} providing corrections for the original theory; e.g., in the CR-CC(t,3) approach, the contribution of the triple excitations missing from the CCSDt is approximated using the method of moments.⁵⁹ To increase the flexibility of the active-space CC methods, the general active-space-concept-based MRCC methods were also introduced^{22,60} and implemented for arbitrary excitation levels for relativistic calculations.^{40,41} We can also mention the SSMRCC with active triples and quadruples (SSMRCCSDtq)⁶¹ developed in our group, where the restricted higher-order amplitudes are used to increase the accuracy of the original SSMRCCSD theory. Finally, the quasiparticle-based MRCC model²¹ developed by us is also an active-space CC theory, where the reference function is a CAS state represented by a determinant of the quasiparticles, and the Hamiltonian has more-than-two-body terms. Various applications of the active-space methods were also published previously,^{62–65} including applications to excited states^{60,66,67} and relativistic calculations.⁴¹

The common points of the active-space CC methods are the special parametrization of the cluster amplitudes and the relatively complex structure of the connected terms of the Hamiltonian and the amplitudes. In this section, we confine ourselves to the brief description of the SRMRCC theory, as it is a simple representative of the active-space CC theories.

The definition of SRMRCC theory, as its name suggests, is closely related to the SRCC formalism, as the cluster expansion of the wave function ansatz starts with a single determinant $|0\rangle$. It is also supposed that this function, the so-called principal determinant, is an element of a CAS. The wave function is parameterized in the usual fashion as

$$|\Psi\rangle = e^{\hat{T}}|0\rangle, \quad (1)$$

where operator \hat{T} is the cluster operator containing the wave function parameters as coefficients of the allowed excitation operators. Without restrictions on \hat{T} , it is an exact parameterization of the wave function in the given one-electron basis. To reduce the computational costs, the excitation levels are usually restricted leading, e.g., to the CCSD method. The multi-reference nature of SRMRCC and other active-space CC approaches comes from the special constraints on the cluster operator, which is parametrized as

$$\hat{T} = \frac{1}{p!q!r!s!} \sum t_{i_1, \dots, i_r, j_1, \dots, j_s}^{a_1, \dots, a_p, A_1, \dots, A_q} \times \{\hat{a}_1^+ \dots \hat{a}_p^+ \hat{A}_1^+ \dots \hat{A}_q^+ \hat{I}_s^- \dots \hat{I}_1^- \hat{I}_r^-, \dots, \hat{I}_1^-\}, \quad p \leq 2, \quad r \leq 2, \quad (2)$$

at the singles and doubles level of theory where, from now on, a_p, b_p , etc., are inactive virtual orbitals and i_r, j_r , etc., are inactive occupied orbitals, I_s, J_s , etc., and A_q, B_q , etc., are active occupied and virtual orbitals, respectively, and only the

number of inactive operators is restricted. Where \hat{T} acts on the reference determinant, all the singly and doubly excited determinants appear with respect to the CAS determinants. These determinants are used to obtain equations for the cluster amplitudes by projecting the stationary Schrödinger equation with the above CC ansatz onto these determinants, i.e., the first-order interacting subspace. If determinant $|0_{i_1, \dots, i_r, j_1, \dots, j_s}^{a_1, \dots, a_p, A_1, \dots, A_q}\rangle$ with the $p \leq 2, r \leq 2$ restrictions is an element of this subspace, the projected equations,

$$\langle 0_{i_1, \dots, i_r, j_1, \dots, j_s}^{a_1, \dots, a_p, A_1, \dots, A_q} | \left(\hat{H}_N e^{\hat{T}} \right)_c | 0 \rangle = 0, \quad (3)$$

are called the CC or residual equations, and

$$\langle 0 | \left(\hat{H}_N e^{\hat{T}} \right)_c | 0 \rangle = E \quad (4)$$

is the CC energy. In the aforementioned relations, \hat{H}_N is the normal-ordered Hamiltonian and the c symbol in the subscripts indicates that just the connected part of the $\hat{H}_N e^{\hat{T}}$ product should be taken into account. As it will be frequently used, please keep in mind that the one-particle indices labeling the projecting determinant Eq. (3) are called the fixed indices.

For the derivation of the CC equations, the contracted terms in Eq. (3) should be evaluated. To increase the computational efficiency, equivalent contributions should be gathered, which can be done in a transparent way using the diagram technique. In this paper, we will use the Goldstone-type anti-symmetrized diagram notations as the starting point. Adapting the diagram technique, the task is to draw each and every independent connected diagram of the Hamiltonian and the amplitudes to build the residuals and the energy contributions. In the discussions of Sec. III, lines playing various roles in the diagrams are classified as contraction lines, free lines, or fixed lines. A contraction line connects the interaction and the amplitudes. Lines of diagrams belonging to an integral or an intermediate vertex which can be later connected to an amplitude vertex are called free lines. These are the lines of interaction vertices which are below the vertex. Lines above the vertices starting from either an interaction or an amplitude vertex of a residual diagram and not involved in contractions are called fixed lines.

Note that, although in Sec. III the discussion is mainly restricted to the SRMRCC theory, with slight modifications the presented approach can be used with the other active-space methods as well.

III. THE NEW APPROACH

A. Representation of amplitudes and intermediates

As in the SRMRCC theory there is no restriction on the number of active creation and annihilation operators, highly-excited cluster amplitudes easily appear. For example, for a 4×4 CAS problem, six-fold excitations have to be treated. Increasing the CAS size, the number of amplitude diagrams and the terms in the CC equations also increase. One can simplify the complexity of the equations by recognizing that the equivalent fixed diagram lines (lines starting from either an interaction or an amplitude vertex of a residual diagram and

not involved in contractions) can be represented by a single symbol. An example for such a compact notation is shown in Fig. 1, where the fixed lines of CCSD diagrams are replaced by string lines leading to a compact diagrammatic description. One can also notice that with the string-based diagrams the CC equations can be represented in a uniform manner for every excitation levels. This idea can be slightly altered to fit better to the active-space CC methods, where the fixed and contraction lines (lines connecting two vertices) can be restricted according to their active or inactive nature. In this subsection, this approach will be presented in more detail. First, a set of algebraic notations will be introduced, which will be handy in the discussion of the computational algorithm, and then the graphical representation will be also shown.

In the case of the active-space CC theories, the occupied and virtual lines can be further classified as active and inactive lines. For our purpose, it is convenient to introduce

multiindices for the active fixed lines. These lines are labeled by integers, and the increasingly ordered lists of these numbers form strings. Following the convention of Ref. 20, according to the occupied or virtual nature of these strings, the **I**, **J** or **A**, **B** notations will be used, respectively, which can be extended by a subscript referring to the length of the string and a spin index whenever it is needed. For a string $\mathbf{A}_p = \{A_1 A_2 \dots A_p\}$, an operator $\hat{A}_p^+ = \hat{A}_1^+ \dots \hat{A}_p^+$ and its adjoint operator \hat{A}_p^- are also defined. The $\mathbf{A} * \mathbf{A}'$ notation will be used for a string obtained by merging string **A** and string **A'**. To help the readability of many index quantities, inactive multiindices are also introduced in a similar fashion, e.g., $\mathbf{i}_q = \{i_1 i_2 \dots i_q\}$ and $\hat{\mathbf{i}}_q^+ = \hat{i}_1^+ \hat{i}_2^+ \dots \hat{i}_q^+$. Finally, $\bar{i}, \bar{j}, \dots (\bar{a}, \bar{b}, \dots)$ are arbitrary (active or inactive) occupied (virtual) orbitals.

Using these notations, a general intermediate appearing in our discussion reads as

$$\begin{aligned} S_{\bar{b}_1 \dots \bar{b}_r j_1 \dots j_q I_1 \dots I_s}^{\bar{i}_1 \dots \bar{i}_m a_1 \dots a_n A_1 \dots A_p} \{ \hat{a}_1^+ \dots \hat{a}_m^+ \hat{a}_1^+ \dots \hat{a}_n^+ \dots \hat{A}_1^+ \dots \hat{A}_p^+ \hat{I}_s^- \dots \hat{I}_q^- \hat{j}_1^- \dots \hat{j}_r^- \dots \hat{b}_1^- \} \\ = S_{\bar{b}_1 \dots \bar{b}_r j_1 \dots j_q}^{\bar{i}_1 \dots \bar{i}_m a_1 \dots a_n} (\mathbf{A}_p, \mathbf{I}_s) \{ \hat{a}_1^+ \dots \hat{a}_m^+ \hat{a}_1^+ \dots \hat{a}_n^+ \hat{A}_p^+ \hat{I}_s^- \hat{j}_q^- \dots \hat{j}_1^- \hat{b}_r^- \dots \hat{b}_1^- \} \\ = S_{\bar{b}_r j_q}^{\bar{i}_m a_n} (\mathbf{A}_p, \mathbf{I}_s) \{ \hat{\mathbf{i}}_m^+ \hat{a}_n^+ \hat{A}_p^+ \hat{I}_s^- \hat{j}_q^- \hat{\mathbf{b}}_r^- \}, \end{aligned} \quad (5)$$

where with the notation we emphasize the special role of the active fixed strings by putting them into parentheses. With the above form it is also indicated how these quantities are treated in the computation, namely, partly by one-particle indices and partly by string indices. For the sake of brevity, these notations are further simplified by introducing a single symbol **Y** for the set $\{\mathbf{i}_m, \mathbf{a}_n, \mathbf{b}_r, \mathbf{j}_q\}$, i.e., $S(\mathbf{Y}, \mathbf{A}_p, \mathbf{I}_s) = S_{\bar{b}_1 \dots \bar{b}_r j_1 \dots j_q}^{\bar{i}_1 \dots \bar{i}_m a_1 \dots a_n} (\mathbf{A}_p, \mathbf{I}_s)$, or using a more compact notation, $S(\mathbf{Y}, \mathbf{X})$, where **X** and **Y** refer to the whole set of active fixed indices and to the remaining indices, respectively.

Similar notations will be used for the cluster amplitudes, i.e.,

$$\begin{aligned} T_{j_1 \dots j_q I_1 \dots I_s}^{a_1 \dots a_n A_1 \dots A_p} \{ \hat{a}_1^+ \dots \hat{a}_n^+ \hat{A}_1^+ \dots \hat{A}_p^+ \hat{I}_s^- \dots \hat{I}_q^- \hat{j}_1^- \dots \hat{j}_1^- \} \\ = T_{j_1 \dots j_q}^{a_1 \dots a_n} (\mathbf{A}_p, \mathbf{I}_s) \{ \hat{a}_1^+ \dots \hat{a}_n^+ \hat{A}_p^+ \hat{I}_s^- \hat{j}_q^- \dots \hat{j}_1^- \} \end{aligned} \quad (6)$$

and $T(\mathbf{Y}, \mathbf{X}) = T_{j_1 \dots j_q}^{a_1 \dots a_n} (\mathbf{A}_p, \mathbf{I}_s)$. Again, our aim with the separation of active fixed indices is to achieve a representation where the active fixed indices can be treated with a string-based algorithm.

In our approach the active indices of amplitudes are organized to be either contraction indices and labeled by one-particle labels or to be fixed indices and described by a string. The active fixed indices of the interaction or intermediate vertices are always treated by strings. When the contraction of an intermediate and an amplitude is processed, the active part of the resulting diagram is the union of the active fixed strings of the connected vertices.

To implement the contractions, our first step is to prepare an index ordering for the amplitudes, which fits for the evaluation of contractions. Due to the definition of cluster amplitudes in the active-space CC theories, the diagrams including an amplitude vertex connected to an intermediate can be characterized by the restrictions on contractions. For a class of such diagrams, the contractions are either occupied or virtual contractions without further restrictions, e.g.,

$$\sum_{\bar{a}\bar{i}} S_{\bar{a}\bar{k}}^{\bar{i}c} T_{ij\mathbf{K}}^{\bar{a}b\mathbf{C}}. \quad (7)$$

Restrictions on contractions appear where the amplitude vertices have more than two one-particle indexed occupied or virtual lines. Following our convention, the fixed one-particle lines must be inactive lines. As an example, consider a diagram where an amplitude has three one-particle virtual indices. At least one of them is a fixed index which must be labeled by inactive indices. In this example, the amplitude cannot connect to the intermediate by more than one inactive virtual index. If there is a second virtual contraction, it is restricted to the active virtual orbitals, e.g.,

$$\sum_{\bar{A}\bar{B}\bar{i}} S_{\bar{B}\bar{A}\bar{I}_1}^{\bar{i}A_2} T_{i\bar{J}_2}^{\bar{B}aA}. \quad (8)$$

In this sense, to get a compact set of diagrams for the active-space CC problems, one can classify the amplitude-intermediate composite diagrams by the number of the restricted active occupied and virtual contraction lines.

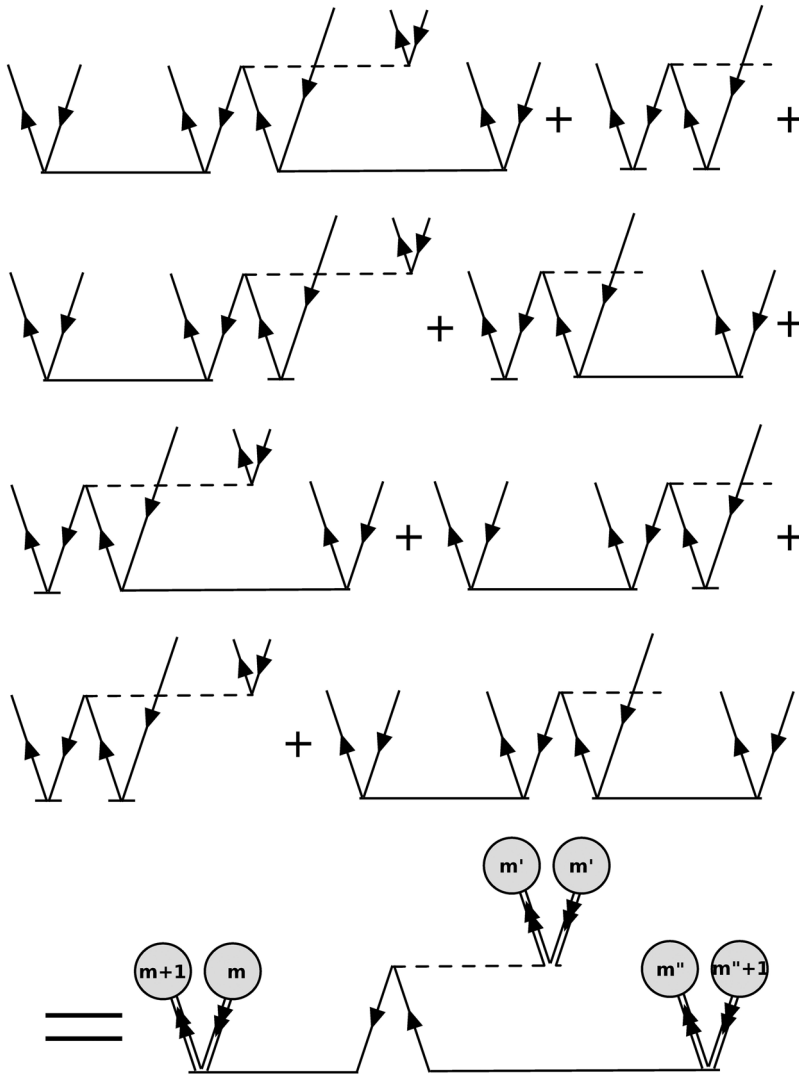


FIG. 1. Compressed diagram notation for diagrams which are different in the fixed indices. The group of fixed indices are signed by double lines; numbers in circles indicate the number of represented lines ($m = 0, 1$; $m' = 0, 1$; $m'' = 0, 1$).

Accordingly, two representations of the amplitudes will be used here. The first one is written as

$$T_{\vec{i}_1 \dots \vec{i}_l}^{\vec{a}_1 \dots \vec{a}_q}(\mathbf{A}_s, \mathbf{I}_n), \quad q \leq 2, \quad l \leq 2, \quad (9)$$

which suits for the situations exemplified by Eq. (7), where the number of virtual contraction and fixed inactive lines together are no more than two, and the same holds for the occupied orbitals. The second representation is useful when the number of virtual (or occupied) contraction and fixed inactive lines together is more than two as, e.g., in Eq. (8). The most general form can be written as

$$T_{i_1 \dots i_k \vec{j}_1 \dots \vec{j}_l I_1 \dots I_m}^{a_1 \dots a_p \vec{b}_1 \dots \vec{b}_q A_1 \dots A_r}(\mathbf{A}_s, \mathbf{I}_n), \quad p + q \leq 2, \quad k + l \leq 2. \quad (10)$$

As in our approach the CC diagrams contain no active fixed orbitals as one-particle index, in each diagram where the above amplitudes appear the active one-particle indices must be contraction indices. For the representation in Eq. (10), the restriction $p + q + r > 2$ or/and $k + l + m > 2$ is also used, otherwise Eq. (10) would become a subset of Eq. (9).

The elementary step of the diagram generation is to form a composite diagram from an intermediate (or an interaction) vertex and an amplitude vertex. As the next step, the role of active fixed strings will be investigated. Using the notations introduced earlier, a particular composite diagram obtained from an elementary intermediate and amplitude diagram can be written as

$$\begin{aligned} & \bar{S}_{\vec{b}\vec{j}}^{\vec{a}}(\mathbf{A}, \mathbf{I}) T_{\vec{i}\vec{j}'\vec{j}'}^{\vec{b}'\vec{a}'\vec{B}'}(\mathbf{A}', \mathbf{I}') \{ \{ \hat{\vec{i}}^+ \hat{\vec{a}}^+ \hat{\vec{A}}^+ \hat{\vec{I}}^+ \hat{\vec{j}}^- \hat{\vec{b}}^- \} \{ \hat{\vec{b}}'^+ \hat{\vec{a}}'^+ \hat{\vec{B}}'^+ \hat{\vec{A}}'^+ \hat{\vec{I}}'^+ \hat{\vec{j}}'^- \hat{\vec{I}}'^- \} \}_{\{ \hat{\vec{k}}^+ \hat{\vec{c}}^+ \hat{\vec{A}}^+ \hat{\vec{I}}^+ \hat{\vec{I}}^- \hat{\vec{I}}^- \hat{\vec{d}}^- \}} \\ & = \bar{S}_{\vec{a}\vec{l}}^{\vec{k}\vec{c}}(\mathbf{A} * \mathbf{A}', \mathbf{I} * \mathbf{I}') \{ \hat{\vec{k}}^+ \hat{\vec{c}}^+ \hat{\vec{A}}^+ \hat{\vec{I}}^+ \hat{\vec{I}}^- \hat{\vec{I}}^- \hat{\vec{d}}^- \}, \end{aligned} \quad (11)$$

where the Einstein convention is used; i.e., a summation is supposed for repeated indices. The subscript in the above relation indicates that only those contractions should be

evaluated which produce $\{ \hat{\vec{k}}^+ \hat{\vec{c}}^+ \hat{\vec{A}}^+ \hat{\vec{I}}^+ \hat{\vec{I}}^- \hat{\vec{I}}^- \hat{\vec{d}}^- \}$ apart from a sign factor. Of course, one can replace the $\mathbf{A} * \mathbf{A}' (\mathbf{I} * \mathbf{I}')$ symbol with a single string, let us say $\mathbf{A}'' (\mathbf{I}'')$, and perform the

summation for \mathbf{A}'' . Keeping the independent summation for \mathbf{A} and \mathbf{A}' (\mathbf{I} and \mathbf{I}') transparently shows that one can detach the active fixed strings from both sides of the equation, and—apart from a sign factor (see Sec. III B)—Eq. (11) can be implemented without the explicit knowledge of the active fixed strings using subroutines where the inputs are the $S_{bj}^{\bar{\mathbf{a}}}$ and $T_{ij'j'}^{\bar{\mathbf{b}}'\mathbf{a}'\mathbf{B}'}$ quantities depending on \mathbf{A} , \mathbf{I} and \mathbf{A}' , \mathbf{I}' , respectively, while the output is $\bar{\mathbf{S}}_{\mathbf{a}\mathbf{l}}^{\bar{\mathbf{k}}\mathbf{c}}$ labeled by the \mathbf{A}' , \mathbf{I}'' strings. It means that such an implementation is excitation level independent. Note that at the rhs of Eq. (11) there is no active fixed index outside the parenthesis.

To illustrate how this idea works, consider the example where a composite diagram is obtained by formally contracting A and A' , \bar{i} and \bar{i}' , and finally the \bar{b} and \bar{b}' index pairs as

$$\begin{aligned} & S_{\bar{b}A}^{\bar{i}}(\mathbf{A}_2, \mathbf{I}_1) T_{\bar{i}'\bar{i}}^{\bar{b}'a'A'}(\mathbf{A}'_0, \mathbf{I}'_2) \\ & \times \{ \{ \hat{i}^+ \hat{\mathbf{A}}_2^+ \hat{\mathbf{I}}_1^- \hat{\mathbf{A}}^- \hat{b}^- \} \{ \hat{b}^+ \hat{a}^+ \hat{\mathbf{A}}'^+ \hat{\mathbf{I}}_2^- \hat{i}'^- \} \}_{\{ \hat{a}^+ \hat{\mathbf{A}}_2^+ \hat{\mathbf{I}}_2^- \hat{\mathbf{I}}_1^- \}} \\ & = \bar{S}^{a'}(\mathbf{A}_2, \mathbf{I}_1 * \mathbf{I}'_2) \{ \hat{a}^+ \hat{\mathbf{A}}_2^+ \hat{\mathbf{I}}_2^- \hat{\mathbf{I}}_1^- \}. \end{aligned} \quad (12)$$

Instead of the algorithm suggested by the above relation, i.e., to perform all the contractions with the result indicated in the subscript, we follow another route, where first the explicit form of operators belonging to active fixed indices is eliminated from the relation and for the other operators the contractions are evaluated using the conventional diagram rules. That is,

intermediate

$$\bar{S}^{a'} = - \sum_{ibA} S_{bA}^{\bar{i}}(\mathbf{A}_2, \mathbf{I}_1) T_{\bar{i}}^{\bar{b}a'A}(\mathbf{A}'_0, \mathbf{I}'_2) \quad (13)$$

is introduced, where the sign from the diagram rules is considered (two loops and one hole contraction result in a minus sign), and then the calculated $\bar{S}^{a'}$ is corrected by another sign factor obtained as the contributions of the eliminated active fixed operators, $\bar{S}^{a'}(\mathbf{A}_2, \mathbf{I}_1 * \mathbf{I}'_2) = \pm \bar{S}^{a'}$ (see Sec. III B). Note that when the active fixed operators are formally eliminated, we still keep in mind that each index in the tensor part of Eq. (12) has a vertical pair⁶⁸ even if its index value is not given explicitly. It is an important point when the correct index order of the composite intermediate is determined.

This concept can be introduced from another point of view using the graphical form of diagrams. Diagrams where the active fixed strings are detached can be represented by the graphs presented in Fig. 2. In these diagrams the double lines with circles are the active fixed labels. Upgoing arrows represent virtual, downgoing arrows represent occupied indices, single arrows are inactive indices, double arrows are active indices, and the lines having no active/inactive restriction hold both a single and a double arrow.

As it is discussed, in the SRMRCC theory two types of diagram contraction appear, namely, restricted to active indices or not restricted at all. To compute those diagrams where none of the contractions are restricted, the amplitudes without active and inactive restrictions can be used [e.g., (a1), (a2), (b1), etc.], while for diagrams with restricted contraction indices, amplitudes such as (a3), (b2), and (f1) are used, where the

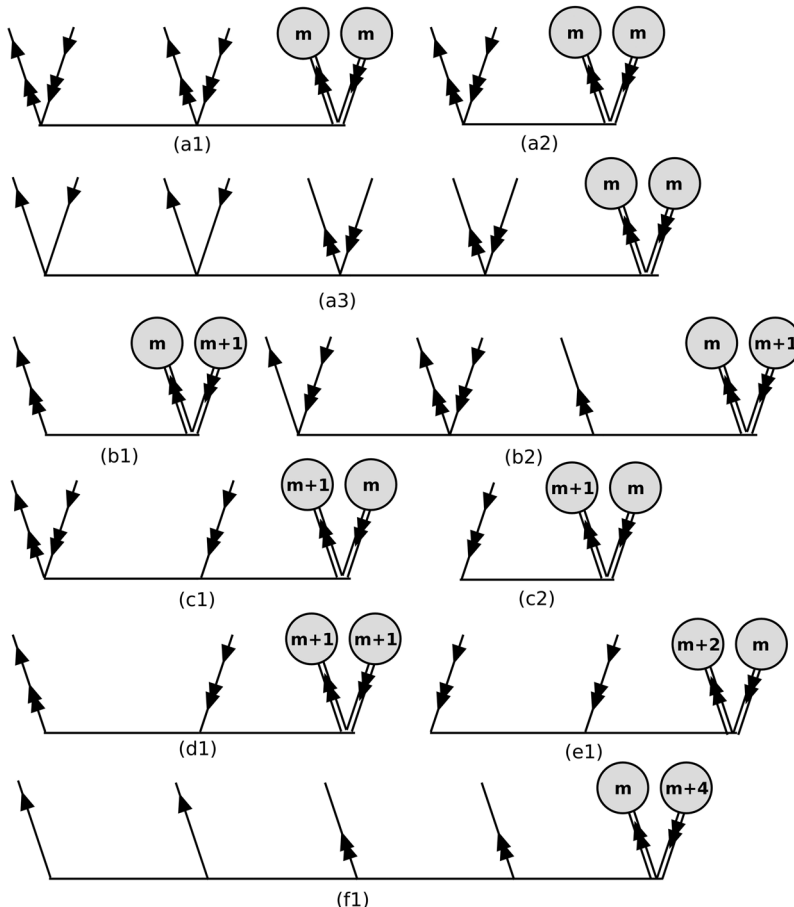


FIG. 2. Examples for amplitude diagrams. The upgoing active fixed lines are described by a string index, the rest of the lines are labeled by one-particle indices. Single, double, and single plus double arrows are used to indicate the lines with inactive, active, and arbitrarily active or inactive indices, respectively.

purely active indices are always contraction indices, while the purely inactive indices are never contraction indices. Note that formally particular lines of diagrams in Fig. 2 have no pairs. In fact, the pairs of these lines are included into the active fixed strings, and when diagrams with single lines are processed using the diagram rules, these lines should be considered as a part of a loop.

The separate treatment of active fixed strings goes in a similar way for the integrals and intermediates. A couple of examples for these diagrams can be found in Fig. 3. The lines at the top of these diagrams are either restricted to inactive orbitals or included into the active fixed strings. It is important to note that there is no need to use intermediates with more than four inactive indices.

Diagrams in Figs. 2 and 3 are our elementary objects. Using these building blocks, all diagrams of the SRMRCC theory can be constructed. The number of the elementary amplitude diagrams considering the spin is 350—most of them have active contraction indices—which is much larger than that in the ordinary diagram representation of the SRMRCC theory. Even though the number of various intermediate diagrams is also larger in our case (due to the diagrams with one and three inactive labels), a relative compactness partly comes from the diagram contraction rules, namely, from the fact that the active one-particle indices of amplitudes must be contracted. One can also notice that the diagram representation does not depend

on the excitation level of the theory. Increasing the excitation level increases the number of strings but not the number of the string-labeled diagrams. As another source of compactness, we should also mention that in the new approach, where it is possible, the contraction lines are not restricted to the active or inactive indices, while in the previous²⁰ implementation each contraction line was restricted to either the active or the inactive orbitals, which is also a small improvement with respect to our previous MRCC implementation.

The important point is that for the implementation of these diagrams, there is no need to explicitly know the form of the active fixed strings.

B. Diagram rules

One of the advantages of our diagram representation is that the topological information is enough to evaluate the diagram contributions. As in our case, the diagram representation is slightly modified by introducing strings; the diagram rules must be updated accordingly. The original CC diagram rules⁶⁸ are not repeated here; only the differences will be highlighted.

Each of our diagrams can contain one-particle lines and active fixed strings, where the active strings are never involved in diagram contractions. The diagrams should be evaluated for all those string combinations which are compatible with the

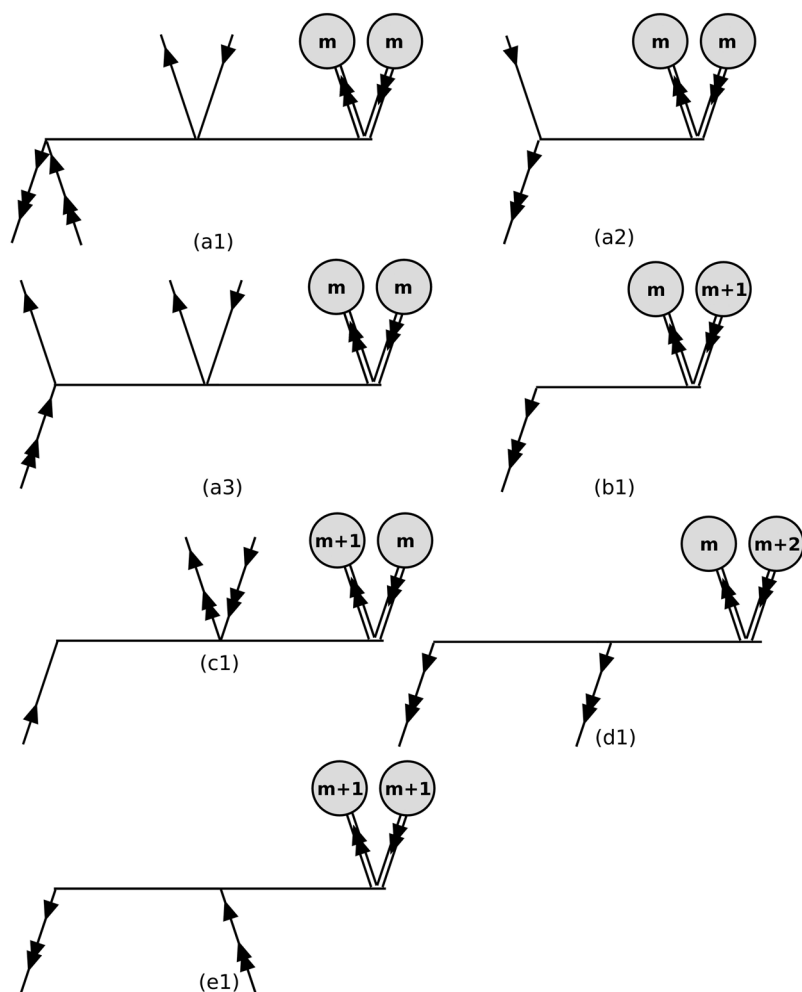


FIG. 3. Examples for the representation of integrals and intermediates. The upgoing active fixed lines are described by a string index; the rest of the lines are labeled by one-particle indices. Single, double, and single plus double arrows are used to indicate the lines with inactive, active, and arbitrarily active or inactive indices, respectively.

rest of the diagram. To indicate these stipulations in the notation, namely, the allowed number of lines in the fixed strings, a number in a circle is used (see Figs. 2 and 3). The string part of the composite objects is obtained by merging the strings appearing on the vertices of the corresponding amplitude and interaction. According to the diagram rules, all permutations of the non-equivalent fixed lines should be taken into account with the proper sign. This condition for the active fixed lines is automatically satisfied due to the anti-symmetry of strings.

Classically the sign of diagrams is calculated from the number of loops (l) and the number of hole lines (h) by the $(-1)^{l+h}$ factor. As a consequence of the additional string lines, it has to be extended by two new sign factors. First, when a composite diagram is constructed from elementary intermediates and amplitudes, the resultant diagram should keep the canonical order of indices as shown in Eq. (5). For this reason, the active fixed strings of the connected vertices should be merged leading to an additional sign factor to be taken into account. At the construction of the merged string, we start with the string of the intermediate (\mathbf{A} and \mathbf{I}) and add the string of the amplitude from the right (\mathbf{A}' and \mathbf{I}'). The sign is calculated from the number of required index interchanges, $\mathcal{P}_{\mathbf{A}\mathbf{A}'}$, as

$$(-1)^{(\mathcal{P}_{\mathbf{A}\mathbf{A}'\alpha} + \mathcal{P}_{\mathbf{A}\mathbf{A}'\beta} + \mathcal{P}_{\mathbf{I}\mathbf{I}'\alpha} + \mathcal{P}_{\mathbf{I}\mathbf{I}'\beta})}. \quad (14)$$

Second, to get the correct sign a further contribution is needed. Figure 4 helps to understand its source, where the contraction of an intermediate and an amplitude is drawn. The diagram contains paired lines⁶⁸ [lines with labels vertically paired in Eq. (5)]. A couple of these paired lines are homogeneously labeled by the active fixed labels, but others start from an active fixed line and end in an inactive line (hereafter called inhomogeneous pairs). Using the ordinary diagram rules, the sign of the composite diagram corresponding to intermediate $\tilde{S}_{\mathbf{I}'\mathbf{I}_k\mathbf{I}_l}^{\mathbf{a}_i\mathbf{A}_k\mathbf{A}'_l}$ can be correctly determined for the given index order. As in this example, when string $\mathbf{I}' (= \mathbf{I}'_i * \mathbf{I}'_l)$ is longer than \mathbf{A}' , indices of string \mathbf{I}' appear in the inhomogeneous pairs and \mathbf{I}'_i is separated from the rest of the fixed occupied active indices (\mathbf{I}'_l) of the amplitude. Before merging the active fixed occupied and

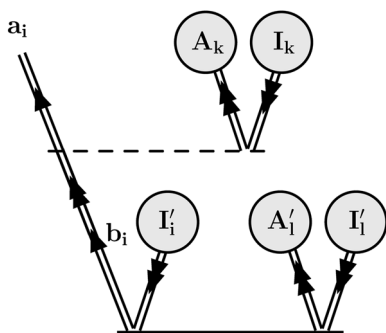


FIG. 4. Illustration for the sign rules. A composite diagram is plotted where both the amplitude and the initial intermediate contribute to the active strings of the composite intermediate. Writing strings \mathbf{I}'_i and \mathbf{I}'_l next to each other form the string of the amplitude. Additional sign factor, $(-1)^{i^*k}$, comes when the fixed occupied active indices of the composite diagram $\tilde{S}_{\mathbf{I}'\mathbf{I}_k\mathbf{I}_l}^{\mathbf{a}_i\mathbf{A}_k\mathbf{A}'_l}$ are rearranged to get the $(-1)^{i^*k} \tilde{S}_{\mathbf{I}_k\mathbf{I}'_l\mathbf{I}'_i}^{\mathbf{a}_i\mathbf{A}_k\mathbf{A}'_l}$ form. Note that merging \mathbf{I}_k , \mathbf{I}'_i , and \mathbf{I}'_l and merging \mathbf{A}_k and \mathbf{A}'_l provides further sign factors. See text for more details.

virtual strings, in our example, \mathbf{I}'_i and \mathbf{I}'_l should be reunited. In the presented example, it needs $(L_{\mathbf{I}'} - L_{\mathbf{A}'})L_{\mathbf{I}}$ index interchanges leading to an additional sign factor of $(-1)^{(L_{\mathbf{I}'} - L_{\mathbf{A}'})L_{\mathbf{I}}}$, where L is the length of the strings. We obtain a similar factor if the length of \mathbf{A}' is longer than that of \mathbf{I}' . Note that these sign factors must be calculated independently for the alpha and beta parts of the diagrams. The sign factor of this kind reads, in general, as

$$(-1)^{\left(\sum_{\sigma=\alpha,\beta} \max(0, L_{\mathbf{I}'\sigma} - L_{\mathbf{A}'\sigma}) L_{\mathbf{I}\sigma} + \max(0, L_{\mathbf{A}'\sigma} - L_{\mathbf{I}'\sigma}) L_{\mathbf{A}\sigma} \right)}. \quad (15)$$

Finally, in the original CC diagram representation, the equivalent amplitudes result in a factor of half. In the original theory, those amplitudes are considered equivalent, which are connected to the Hamiltonian by identical lines and have identical fixed lines as well. As our diagrams are formally evaluated for every string combination, amplitudes with identical contraction lines and identical fixed inactive lines should be considered equivalent, and such a diagrams should be multiplied by $\frac{1}{2}$.

C. Derivation of the SRMRCC equations

The derivation of the SRMRCC equations presented here is based on our diagram representation and diagram rules. To tackle the large number of terms in the SRMRCC formalism, it is performed using a computer code.

For a given element of the residual vector (or for the energy), many diagrams give contribution. Calculation of these diagrams one by one is inefficient; therefore, a proper factorization of diagram contributions is needed, and where possible, reusable intermediate quantities should be introduced. As a first step, each elementary diagram (containing a single Hamiltonian vertex) is constructed in as many ways as it is possible from the given number of amplitude vertices. For each contraction route, the computation expenses are estimated calculating the required number of multiplications (with hypothetical dimensions: 200 virtual orbitals, 20 electrons, 4×4 CAS problem). Here we supposed that the number of active strings is small with respect to the number of inactive orbitals, and the length of the strings is not considered in the cost calculations. In this way, we got system independent equations, which may not be optimal for large CAS problems. This might be seen as a strong restriction, but considering that the SRMRCC method is not the method of choice for large CAS problems, this can be a reasonable compromise.

After determining the computation prices for each calculation route, the elementary diagrams with four amplitudes (a single Hamiltonian connected to four T 's) are investigated and separated into a single T and an intermediate with three T 's (hereafter shortly referred to as the three- T intermediates) choosing the cheapest option. This process starts from the most expensive four- T diagram and proceeds in the direction of the cheapest one. Once an intermediate is chosen, it will be calculated and available for the computation of a subsequent four- T diagram; thus, its computation price will be

set to zero thereafter. At the next step the elementary three- T contributions to the residual and the three- T intermediates chosen earlier are investigated in a similar fashion. Following this strategy, elementary intermediates with one, two, and three amplitude vertices are defined. At the last step, where it is possible, the topologically equivalent elementary intermediates are summed up to form the final form of intermediate quantities.

The equation derivation routine applied here keeps the drawback of our previous approach,²⁰ where for each diagram the factorization is defined first, then the intermediates are defined. It is clearly not the most cost-efficient procedure for the evaluation of the equations. A more sophisticated algorithm can be found in Ref. 35.

The system independent code was generated in the FORTRAN programming language.

D. Computation of contractions

The core of the CC code is a set of tensor contraction subroutines (TCS's). Each topologically different diagram with a particular combination of spin and active/inactive labels appearing in the SRMRCC equations is handled by a separate TCS. The TCS's work with the one-particle index part of the operators; the only contribution from the active fixed strings is the sign factor considered earlier (see Sec. III B). For the SRMRCC method, the number of the independent TCS's is 3062. This number slightly depends on the parameters (i.e., the CAS size, the number of occupied and virtual orbitals) of the program deriving the CC equations but does not depend on the dimensions or the excitation level of the actual calculation. Because the number of TCS's is large, we are bound to use a computer code to generate these subroutines as well.

Each of these subroutines contains an implementation of the given diagram based on the *dgemm* routine, a very efficient component of the BLAS library, to evaluate matrix products. For cases when small matrix products have to be evaluated, an implementation based on the ordinary loop structures is also generated.

As it is discussed in Ref. 25 in more detail, to be able to use the *dgemm* routine, the tensor contractions should be converted into matrix products. In some cases, the amplitude and intermediate index orders fit to this requirement, and then the matrix product can be performed directly. In other cases, indices of these tensors must be reorganized. Reducing the potential cost of index reordering, the first indices of intermediate arrays are free virtual indices (virtual indices of an intermediate vertex which can be connected to an amplitude vertex) followed by the free occupied, fixed virtual, and fixed occupied indices. As there is no index order convention which is suitable for every diagram, the reordering of indices is frequently unavoidable. For that purpose, a general index reordering routine is also implemented, which also handles the various index restrictions. These index restrictions have two sources, namely, in our approach the fixed one-particle lines are always inactive indices, and due to the diagram rules we can utilize the $q_1 < q_2$ type of constraint for equivalent contraction and equivalent fixed index pairs.

As it is discussed earlier, particular amplitudes have active lines which are always contraction lines. For the efficient computation of their diagrams, these indices (occupied and virtual with alpha or beta spin) are addressed by a single string index. This string index is always the leading index of these amplitudes (in FORTRAN the leading index is the fastest one). When such contraction is performed, the indices of the intermediate are ordered and constrained to fit to the ordering and the index restrictions of the string index. This compact treatment of active contraction strings allows a more economical memory usage as well, which is important, as couple of the diagrams have eight lines (see, for instance, diagram a3 in Fig. 2).

Apart from the contraction of tensors, the permutation of non-equivalent inactive indices is also performed here using a loop structure-based implementation.

At this point, we should also discuss the possible form of the parallel implementation of the TCS's introduced earlier. Due to the strategy based on the string description of the fixed indices, the tensors appearing in the TCS are relatively low dimensional quantities. The number of inactive indices is at most four, which can be extended in the case of amplitudes by a single string index of the active contraction indices. Most of these diagrams easily fit into the memory for the accessible basis set sizes. It is not necessarily the case for diagrams having three or four virtual indices. These diagrams are treated specially by slicing them up along an inactive fixed index (as it is discussed in Sec. III E) to obtain sufficiently small arrays. Another important characteristic of the program is the large number of terms needed to be processed, of which even the most expensive ones have only a few percent contribution to the calculation time (see Sec. IV). It seems that the main bottleneck of the code is rather the CPU time than the memory. Under these circumstances, the most practical approach for the parallelization is to use the OpenMP technology to improve the efficiency of the TCS's while the Message Passing Interface (MPI) can be used to distribute the processing of diagrams among the available nodes. The latter is discussed in more detail in Sec. III E. In the recent implementation, the diagram contractions are automatically parallelized through the *dgemm* routines. The OpenMP implementation of the index reordering routine remains an important task for the future.

E. Algorithmic details

The basic idea behind the code structure was to perform the contraction of one-particle labels in the innermost part of the loop structures in a highly efficient way, while the slower string manipulations are carried out outside.

It is supposed that the whole set of amplitudes, the residual vectors, and the integrals with less than three inactive virtual indices can be kept in the memory. As some of the one-particle labels of amplitudes in Fig. 2 are not restricted to the inactive labels, the representation shown in Fig. 2 is strongly redundant, and for larger basis sets or CAS spaces, keeping all of them in memory would strongly limit the applicability of the code; thus, a less redundant form of the amplitudes is stored in the memory. These are shown in Fig. 5. The active fixed strings are characterized by a string index, and the inactive

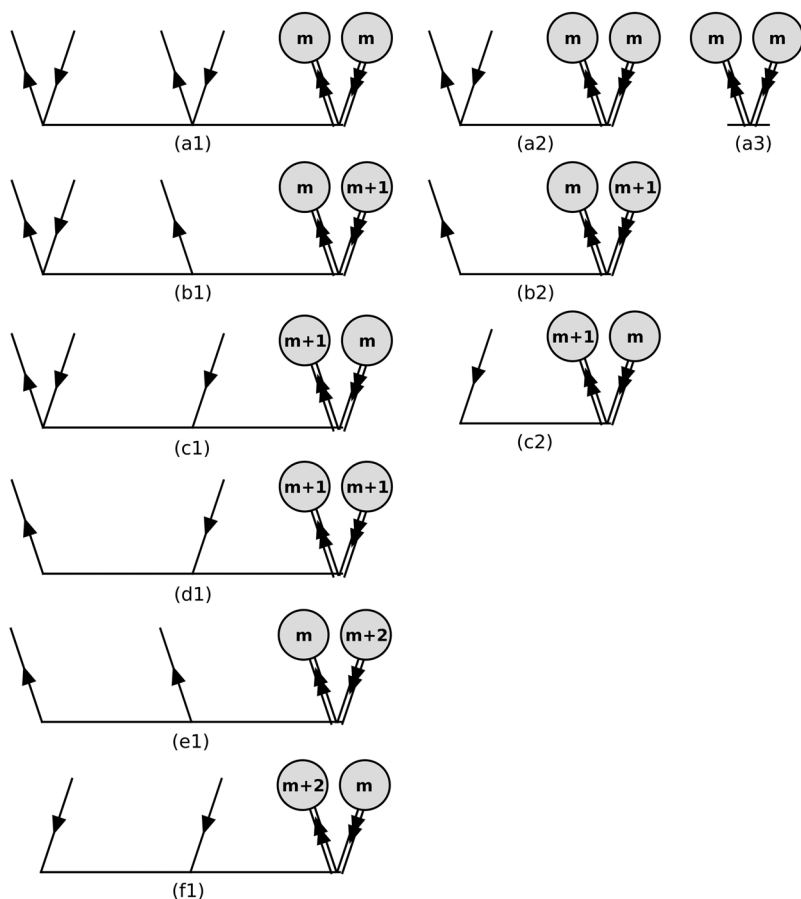


FIG. 5. Compact representation of amplitudes. The active fixed lines are described by a string index; inactive indices are labeled by one-particle indices.

part, by ordinary one-particle indices. Where the less compact form (amplitudes in Fig. 2) is needed, it is constructed on the fly. This step is referred to as the amplitude gathering. Though the scaling of this step is low, the amplitude gathering is still a relatively expensive transformation, as it is difficult to perform it in a vectorized manner. In the design of the loop structure, this point is kept in mind, and we minimized the number of amplitude gatherings as much as possible. For this reason the gathered amplitudes are stored in the memory as long as there is available free space and reused when it is needed.

The size of the intermediate quantities is also worth considering. As the one- T intermediates, where it is possible, contain one or two virtual contraction lines and where there is a single remaining free virtual index of a one- T intermediate it is contracted with a second amplitude, there is no two- T intermediate with more than two inactive virtual lines. Therefore, we can expect that the size of the intermediates containing two amplitude vertices is smaller. As we have roughly 600 one- T intermediates (without exploiting the spin symmetry) and just slightly more, 700 two- T intermediates, we can suppose that the full memory requirement of the one- T intermediates is significantly larger. Accordingly, these intermediates are never stored on the hard drive but used on the fly.

The diagrams of the interaction and the intermediates with at least three virtual lines may be too large to be kept in memory. To tackle this problem, these are separately treated in an integral driven approach, which allows us to slice up the large integral arrays into blocks of acceptable size. Each large

interaction diagram is divided into blocks according to an inactive virtual fixed index, and the resultant blocks are labeled by a fragment index. Note that all the one- T intermediates with three inactive virtual indices originate from the integral driven block of the code and can be also treated in a fragmented form. Figure 6 shows the recent realization of this part of the code.

The outermost loops of the integral driven algorithm run over diagram fragment index M . Loop at lines 2 and 3 together run through the active fixed strings of the first amplitude to be attached to the interaction diagram. After initializing the one- T intermediates and residuals for a given active fixed string \mathbf{X} , the diagram contractions are performed using the TCS's. Here, $H(\mathbf{Y}'', M)$ refers to the interaction diagram defined by \mathbf{Y}'' , M in the arguments of $S_1(\mathbf{Y}, \mathbf{X}, M)$ and $H(\mathbf{Y}'', M)$ symbolizes the index restrictions, and the $\{H(\mathbf{Y}'', M)T(\mathbf{Y}', \mathbf{X})\}_{\mathbf{Y}}$ refers to the diagram contractions of $H(\mathbf{Y}'', M)$ and $T(\mathbf{Y}', \mathbf{X})$ producing $S_1(\mathbf{Y}, \mathbf{X}, M)$. The required amplitudes which are not available in the memory are gathered at line 6 and, if it is possible, kept in the memory. The computed intermediates are used on the fly. Loops at lines 9 and 10 run over the active fixed strings $\bar{\mathbf{X}}$ of the second amplitudes. Amplitudes needed for the two- T diagrams and not stored in memory are gathered at line 12. Two- T residuals and intermediates are computed at lines 13 and 15. If the given two- T intermediates are evaluated previously, these are read from disk and updated at line 14 using the TCS's.

For those diagrams where the integrals can be kept in memory, a different strategy—a one- T intermediate driven

```

1  loop for fragment index M
2  loop for  $p, q, r, s$  length of strings
3  loops for active strings of intermediates  $\mathbf{X} = (\mathbf{A}_p^\alpha, \mathbf{A}_q^\beta, \mathbf{I}_r^\alpha, \mathbf{I}_s^\beta)$ 
4  initialize intermediates and residuals  $S_1(\mathbf{Y}, \mathbf{X}, M)$  for all  $\mathbf{Y}$ 
5  loop for intermediate  $S_1$  and residual diagrams  $R$ 
6  if not available gather  $T(\mathbf{Y}', \mathbf{X})$  for all  $\mathbf{Y}'$ 
7   $S_1(\mathbf{Y}, \mathbf{X}, M) = S_1(\mathbf{Y}, \mathbf{X}, M) + \{H(\mathbf{Y}'', M)T(\mathbf{Y}', \mathbf{X})\}_{\mathbf{Y}}$ 
8   $R(\mathbf{Y}, \mathbf{X}, M) = R(\mathbf{Y}, \mathbf{X}, M) + \{H(\mathbf{Y}'', M)T(\mathbf{Y}', \mathbf{X})\}_{\mathbf{Y}}$  for all  $\mathbf{Y}'$  and  $\mathbf{Y}''$ 
9  loop for  $\bar{p}, \bar{q}, \bar{r}, \bar{s}$ 
10 loop for active fixed strings  $\bar{\mathbf{X}} = (\mathbf{A}_{\bar{p}}^{\alpha\alpha}, \mathbf{A}_{\bar{q}}^{\alpha\beta}, \mathbf{I}_{\bar{r}}^{\nu\alpha}, \mathbf{I}_{\bar{s}}^{\nu\beta})$ 
11 loop for intermed.  $S_2$ , energy, and residual diags.  $R$ 
12 if not available gather  $T(\mathbf{Y}', \bar{\mathbf{X}})$  for all  $\mathbf{Y}'$ 
13  $R(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M) = R(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M) + \{S_1(\mathbf{Y}'', \mathbf{X}, M)T(\mathbf{Y}', \bar{\mathbf{X}})\}_{\mathbf{Y}}$  for  $\forall \mathbf{Y}', \mathbf{Y}''$ 
14 read required  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M)$ 
15  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M) = S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M) + \{S_1(\mathbf{Y}'', \mathbf{X}, M)T(\mathbf{Y}', \bar{\mathbf{X}})\}_{\mathbf{Y}}$ 
16 for  $\forall \mathbf{Y}', \mathbf{Y}''$ 
17 write  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}, M)$  to disk

```

FIG. 6. Loop structure of the integral driven algorithm. It is used for diagrams where the Hamiltonian has more than two virtual lines. Notation $\{H(\mathbf{Y}'', M)T(\mathbf{Y}', \mathbf{X})\}_{\mathbf{Y}}$ indicates those diagrams of $H(\mathbf{Y}'', M)$ and $T(\mathbf{Y}', \mathbf{X})$ where the contractions of \mathbf{Y}'' and \mathbf{Y}' produce \mathbf{Y} . See text for more details.

algorithm—can be advantageous. As the number and the memory requirement of the one- T intermediates are large, it is worth constructing them one by one, and instead of storing them on the hard drive, they are used to calculate the new intermediates and residuals on the fly.

The loop structure of the one- T intermediate driven algorithm is briefly presented in Fig. 7. The loops at lines 1 and 2 define the active fixed lines of the one- T intermediates. Between lines 4 and 9, those integral and amplitude elements are collected, which together hold the specified active fixed line string \mathbf{X} . At lines 7 and 8, diagrams with string \mathbf{X} are evaluated. Once these are ready, the indices of a next set of amplitudes are defined by the loops at lines 11 and 12, and the two- T energy and residual contributions are updated for string $\mathbf{X} * \bar{\mathbf{X}}$, then the two- T intermediates are calculated for the given string.

The implementation of the three- T and four- T diagrams follows a similar philosophy. The next part of the code is a two- T driven algorithm. The two- T intermediates are read from the disk once, and we calculate their contributions for all the three- T intermediates and the residuals. The three- T intermediates are stored on disk and read when needed. At the last step, the three- T intermediates are read from disk, and the residual contributions are calculated.

We should also mention that the most I/O demanding operations are the read and write operations at lines 13 and 18 in Fig. 7, where the two- T intermediates are updated. As the number of these update operations depends on the length of the strings, these steps can be expensive for large CAS spaces. There is a similar I/O step in the integral driven algorithm in Fig. 6, but only a few two- T intermediates are treated there, which has negligible cost.

```

1  loop for  $p, q, r, s$  string lengths
2  loops for active strings of intermediates  $\mathbf{X} = (\mathbf{A}_p^\alpha, \mathbf{A}_q^\beta, \mathbf{I}_r^\alpha, \mathbf{I}_s^\beta)$ 
3  initialize all  $S_1(\mathbf{Y}, \mathbf{X})$  for all  $\mathbf{Y}$ 
4  loops for substrings  $\mathbf{X}' = (\mathbf{A}_{p'}^\alpha \in \mathbf{A}_p^\alpha, \mathbf{A}_{q'}^\beta \in \mathbf{A}_q^\beta, \mathbf{I}_{r'}^\alpha \in \mathbf{I}_r^\alpha, \mathbf{I}_{s'}^\beta \in \mathbf{I}_s^\beta)$ 
5  loop for intermed.  $S_1$ , energy, and residual diags.  $R$ 
6  if not available gather  $T(\mathbf{Y}', \mathbf{X}')$  for all  $\mathbf{Y}'$ 
7   $S_1(\mathbf{Y}, \mathbf{X}) = S_1(\mathbf{Y}, \mathbf{X}) + \{H(\mathbf{Y}'', \mathbf{X}'')T(\mathbf{Y}', \mathbf{X}')\}_{\mathbf{Y}}$  for all  $\mathbf{Y}''$  and  $\mathbf{Y}'$ 
8   $R(\mathbf{Y}, \mathbf{X}) = R(\mathbf{Y}, \mathbf{X}) + \{H(\mathbf{Y}'', \mathbf{X}'')T(\mathbf{Y}', \mathbf{X}')\}_{\mathbf{Y}}$  for all  $\mathbf{Y}''$  and  $\mathbf{Y}'$ 
9  where  $\mathbf{X} = \mathbf{X}' * \mathbf{X}''$  for all  $\mathbf{Y}''$ 
10 end of loops on substrings
11 loop for  $\bar{p}, \bar{q}, \bar{r}, \bar{s}$  string lengths
12 loop for active fixed strings  $\bar{\mathbf{X}} = (\mathbf{A}_{\bar{p}}^{\alpha\alpha}, \mathbf{A}_{\bar{q}}^{\alpha\beta}, \mathbf{I}_{\bar{r}}^{\nu\alpha}, \mathbf{I}_{\bar{s}}^{\nu\beta})$ 
13 read  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}})$  for all  $\mathbf{Y}$ 
14 loop for intermed.  $S_2$ , energy, and residual diags.  $R$ 
15 if not available gather  $T(\mathbf{Y}', \bar{\mathbf{X}})$  for all  $\mathbf{Y}'$ 
16  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}) = S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}) + \{S_1(\mathbf{Y}'', \mathbf{X})T(\mathbf{Y}', \bar{\mathbf{X}})\}_{\mathbf{Y}}$  for all  $\mathbf{Y}''$  and  $\mathbf{Y}'$ 
17  $R(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}) = R(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}}) + \{S_1(\mathbf{Y}'', \mathbf{X})T(\mathbf{Y}', \bar{\mathbf{X}})\}_{\mathbf{Y}}$  for all  $\mathbf{Y}''$  and  $\mathbf{Y}'$ 
18 write  $S_2(\mathbf{Y}, \mathbf{X} * \bar{\mathbf{X}})$  to disk

```

FIG. 7. Loop structure of the one- T intermediates driven algorithm. It is used for diagrams where the Hamiltonian has no more than two virtual lines. See the text and the caption of Fig. 6 for more details.

Finally, the parallelization of the above code structure should also be considered. Both the integral driven and the one- T intermediate driven algorithm can be easily split up into independent pieces which could be efficiently evaluated using MPI. In the case of the integral driven algorithm of Fig. 6, an interval of fragment index M values can be assigned to each node to compute. The contributions to the residual calculated on the nodes can be supposed to have manageable size and can be gathered by the master node. The size of the computed two- T intermediates obtained from the integral driven algorithm is also limited and can be also gathered. The size of the intermediates of the one- T intermediate driven algorithm strongly depends on the size of the active space. If the active space is small, the outermost loop at Fig. 7 can be parallelized, and the residuals and the two- T intermediates can be easily gathered. In this case, the master handles all the two- T intermediates. At the next step, the three- T and four- T diagrams are calculated in a two- T driven algorithm, which can be easily distributed among the nodes. The bottleneck of this approach is the gathering step of the two- T intermediates, where data proportional to the number of nodes times the size of the two- T intermediates information is communicated to the master. For large CAS problems, it is preferable to keep the two- T intermediates at the local nodes, process them on the spot, and copy only the contributions to residual back to the master. It means that the same three- T and four- T diagrams are processed by each node using the locally calculated two- T intermediate contributions. In this solution, a limited amount of CPU time is sacrificed, as the computation of the three- T and four- T diagrams is inefficient, but considering that the contribution of the three- T and

four- T diagrams for the complete calculation time is relatively small (less than 15% for the investigated systems), such an approach can be still justifiable.

IV. TEST RESULTS

A 2×2 , a 4×4 , and a 6×6 CAS problem with various basis sets^{69,70} were chosen to investigate how these ideas work in practice. For the 2×2 problem, the HF molecule is used correlating eight electrons, the 4×4 CAS example is the H₂O molecule with 8 electrons, and the 6×6 model system is the N₂ molecule correlating 10 electrons.

The calculations were performed using the new program and the previous SRMRCC code which is the part of the Mrcc program package.⁴⁹ The comparison of the measured timings can be found in Table I. The calculations were performed on a single thread of a 3.5 GHz Intel Xeon E5-1650 processor with 125 Gbyte memory. In the presented calculations the wall clock and CPU times were close to each other. For this reason, where it is not indicated otherwise, wall clock times are presented.

For the 2×2 and the 4×4 systems, we found considerable speedup with respect to the original code already with moderate sized basis sets. For example, using the Mrcc code one iteration step for the HF model took 546 s with the aug-cc-pVQZ basis set, while the new code completed the task in 13 s. In the measured interval the speedup for the H₂O system is smaller. Using the cc-pV5Z basis set, which was the largest investigated one for H₂O, Mrcc needed 9924 s, while the new code required 772 s for a single iteration step.

TABLE I. Wall clock times (in seconds) for a single iteration step are presented for the HF, H₂O, and N₂ molecules with various basis sets. The number of basis functions (Nb), the timings for the previous and the new code, and the speedups are presented together with the contributions of the amplitude sorting (AS) and tensor contraction subroutines (TCS's) to the CPU time with respect to the entire CPU time. The ratio of CPU times belonging to the *dgemm*-based and loop-based TCS's is also reported where the CPU time of the index reordering is also included into the CPU time of the *dgemm*-based TCS's.

| | Nb | Prev. code | New code | Speedup | CPU(AS)/ CPU(full) | CPU(TCS)/ CPU(full) | CPU(<i>dgemm</i>)/ CPU(loop) |
|-------------------------------------|-----|---------------|-------------|---------|-----------------------|------------------------|-----------------------------------|
| HF (0 \times 0 CAS) | | | | | | | |
| aug-cc-pVQZ | 125 | 83 | 7.4 | 11 | 0 | 0.27 | 0.22 |
| cc-pV5Z | 145 | 144 | 9.5 | 15 | 0 | 0.27 | 0.22 |
| aug-cc-pV5Z | 206 | 740 | 28.0 | 26 | 0 | 0.27 | 0.24 |
| HF (2 \times 2 CAS) | | | | | | | |
| cc-pVQZ | 84 | 112 | 4 | 28 | <0.01 | 0.71 | 0.31 |
| aug-cc-pVQZ | 125 | 546 | 13 | 42 | <0.01 | 0.71 | 0.29 |
| cc-pV5Z | 145 | 963 | 23 | 42 | <0.01 | 0.71 | 0.29 |
| aug-cc-pV5Z | 206 | 3 946 | 84 | 47 | <0.01 | 0.71 | 0.29 |
| H ₂ O (4 \times 4 CAS) | | | | | | | |
| cc-pVTZ | 57 | 135 | 20 | 6.8 | <0.01 | 0.59 | 0.38 |
| cc-pVQZ | 114 | 1 327 | 127 | 10.4 | <0.01 | 0.73 | 0.29 |
| aug-cc-pVQZ | 171 | 5 273 | 467 | 11.3 | <0.01 | 0.81 | 0.27 |
| cc-pV5Z | 200 | 9 924 | 772 | 12.9 | <0.01 | 0.83 | 0.27 |
| N ₂ (6 \times 6 CAS) | | | | | | | |
| cc-pVTZ | 58 | 2 114 | 1 216 | 1.7 | 0.02 | 0.41 | 0.47 |
| cc-pVQZ | 108 | 11 207 | 5 555 | 2.0 | 0.16 | 0.46 | 0.34 |
| aug-cc-pVQZ | 158 | 36 411 | 16 331 | 2.2 | 0.19 | 0.53 | 0.31 |
| cc-pV5Z | 180 | 55 476 | 23 307 | 2.4 | 0.21 | 0.54 | 0.30 |

We could not achieve such relatively large speedups for the N_2 molecule. Moving from the cc-pVTZ to the cc-pV5Z basis set, the achieved speedup increases from the value of 1.7 to 2.4. As the timings in Table I clearly indicate the measured speedups increase with the basis set size for all the investigated CAS problems and for a given basis set size, the speedup is smaller with increasing CAS dimensions. The scaling with the number of occupied orbitals has been also investigated using the 2×2 , 4×4 , and 6×6 CAS reference spaces. For the 2×2 and 4×4 cases, 150 virtual functions, while for the 6×6 CAS calculations 53 virtual functions were used, respectively, and the number of occupied orbitals was increased from the value of 8. As the results presented in Fig. 8 show that, even if the iteration time of the previous SRMRCC code rises steeper with the increasing electron number than that for the new code, for the investigated interval the speedup values slightly decrease. The reason behind this phenomenon is the offset in the computation time of the previous code caused by the overhead of the integer operations related to the string manipulations. As it can be seen in Fig. 8, the achieved speedups for the 2×2 and 4×4 problems are still significant.

The zero CAS limit, i.e., CCSD, for the HF molecule is also compared. To that end, the new SRMRCC code was used without further optimization. The results presented in Table I reveal that the obtained speedups are rather large but still smaller than those in the 2×2 CAS case. Note that for the CCSD case, due to the lack of active orbitals, there is no need for the amplitude gathering.

The most important contribution to the computation time comes from the TCS's and, for the large CAS problems, from

the amplitude gathering. In Table I we presented the ratio of the CPU time spent in the TCS's and the amplitude gathering with respect to the full CPU time. For the tested model systems, in the larger basis sets, roughly 20%-30% of the CPU time is consumed by the collection of intermediates, array initialization, and other data processing tasks, which are not detailed in Table I. For the 2×2 and 4×4 test systems, the cost of the amplitude gathering is negligible, and the computation time is dominated by the work performed by the TCS's. In the case of the 4×4 and 6×6 CAS calculations, the CPU times are sensitive to the basis set size and increase with the number of virtual orbitals. The most remarkable observation is the relatively small portion of the CPU time spent in the TCS's for N_2 , where, for example, it is only 41% for the cc-pVTZ and 54% for the largest investigated cc-pV5Z basis set. Besides the time consuming work of the TCS's, the amplitude gathering has also significant contribution to the computation time for the N_2 calculations (see Table I). These two components together cover more than 75% of the CPU time in the large basis set limit.

An important point of the new implementation is the use of the *dgemm* routine to perform the diagram contractions. For the calculations presented earlier, the *dgemm*-based algorithm was used in all the TCS's instead of the loop structure-based one. As, in general, the application of the *dgemm* routine requires the index reordering of the amplitude and intermediate indices, it is not obvious whether the use of the *dgemm* routine is economical or not. To investigate this crucial point, calculations were performed for the test systems with the loop-based implementation as well. According to the results in the last column

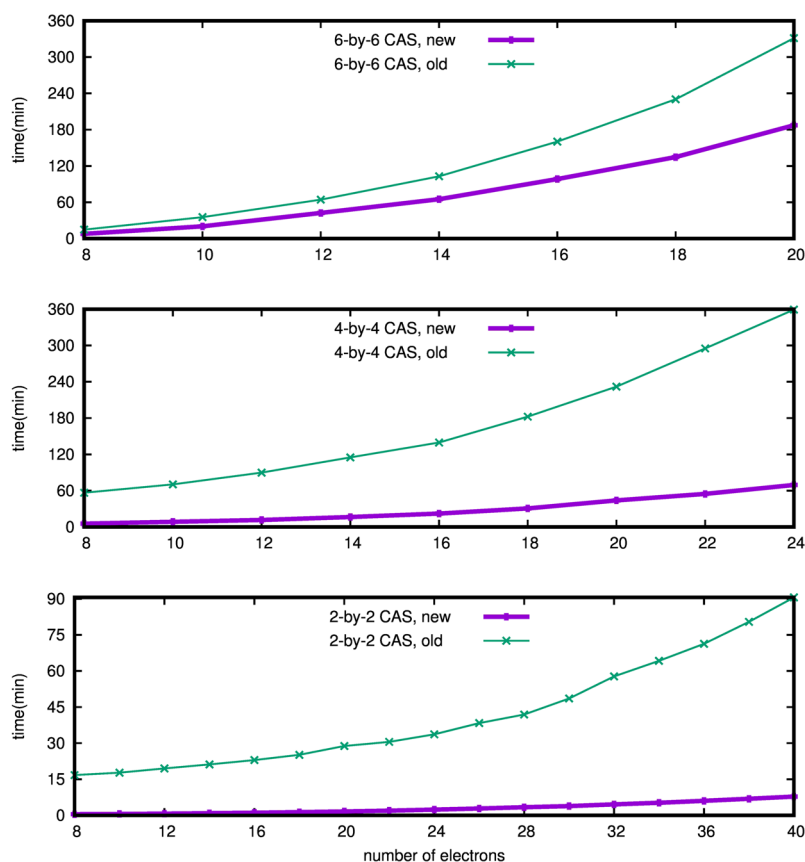


FIG. 8. Running time of the SRMRCC calculations with the previous and the new code as a function of the number of electrons. For the 2×2 and 4×4 CAS-based calculations 150 virtual functions, while for the 6×6 problem 53 virtual functions were used. See text for more details.

of Table I, the use of *dgemm* together with the data reorganization step is roughly three times faster than the loop-based approach, and the speedup factor increases with the size of the basis set.

Though the use of *dgemm*-based TCS's instead of the loop-based ones is obviously beneficial, for some TCS's the loop-based routines are faster. For the N_2 molecule in the cc-pVQZ basis, we also compared term by term the execution times of the *dgemm*- and loop-based calculations for the TCS's and compiled the source code with the faster option for each subroutine. With this optimized version, the CPU time could be reduced by 5%. As it is not obvious when the loop-based implementation would improve the performance in general, the use of the *dgemm*-based implementation in all the TCS's seems to be reasonable.

Finally, we investigated the scaling of the new code. The theoretical scaling of computation time with the number of virtual (n_v) and occupied (n_o) orbitals is $N_{act}n_v^4n_o^2$, where N_{act} is the number of active determinants in the CAS space, and we suppose that the computation time is dominated by the calculation of the ladder diagrams. To check this assumption, we measured the contribution of the ladder diagrams to the full CPU time and found that only 21.1%, 17.6%, and 4.2% of the CPU time were spent to calculate these most expensive diagrams for the HF, H_2O , and N_2 systems, respectively, for the largest investigated basis sets (see Table I). Though for HF with the studied basis sets the ladder diagrams still do not dominate the computation time, we found that the main contributions come from the diagrams scaling as n_v^4 as it is presented in Fig. 9. Here the computation times divided by $N_{act}n_v^4n_o^2$ were plotted against the number of basis functions. As one can see, the lines for the HF models are roughly horizontal, while for the H_2O and the N_2 calculations it has a $\frac{1}{x}$ -like character, which indicates that the large number of diagrams scaling as n_v^3 (or even n_v^2) have a strong contribution to the calculation time. This non-linearity is more pronounced for the N_2 calculations and observable with the previous and the new code as well. It seems that for the large CAS space limit, below a certain number of virtual orbitals not the few diagrams with the highest scaling

but the large number of lower-order terms becomes the main obstacle.

It seems that for N_2 with the investigated basis set sizes we are still below this level, which can be a reason behind the low speedup obtained for this system. The relatively large CPU requirement of the amplitude gathering is also partly responsible for the lower speedups, and of course, the derived SRMRCC equations are not necessarily optimal for large CAS problems (see Sec. III C), and probably it has its own negative effect.

V. APPLICATION: WEAK INTERACTIONS OF TWO BERYLLIUM ATOMS

As a demonstration of the applicability of the SRMRCC method and our new code, we investigated the weak interaction of two beryllium atoms. Though the beryllium dimer is a small system, it is surprisingly difficult to describe its properties using theoretical methods due to the weak interaction and the MR nature of the equilibrium state. Although it is extensively studied by various methods^{50–52,71–76}—to our knowledge—no previous SRMRCC study has been published for this system.

For the qualitatively correct description of the beryllium dimer, at least triple- ζ basis sets and the consideration of the dynamic electron correlation are needed.⁵² In the previous studies to obtain accurate dissociation energy, basis sets with at least g function were also included.^{52,71}

The latest experimental data were published by Merritt *et al.*,⁷⁷ who reported 929.7 cm^{-1} for the dissociation energy. Using these experimental data, Meshkov *et al.*⁷⁸ estimated the dissociation energy as 934.9 cm^{-1} from a direct potential fit. Most of the published theoretical data were calculated using MR configuration interaction (MRCI), size-consistency corrected MRCI, and frozen-core full configuration interaction (FCI) methods. Among the many theoretical studies performed on Be_2 , we should mention the study of Gdanitz,⁷³ who used a large 19s 11p 6d 4f 3g 2h basis set and a valence-CAS reference space for the explicitly correlated averaged coupled pair functional (r_{12} -MR-ACPF) calculations correlating all electrons and obtained 898 cm^{-1} for the dissociation energy. Large basis MRCI results were published by Mitin,⁷⁴ who obtained 822 cm^{-1} using the t-aug-cc-pV7Z basis set. More recently, Kalemos⁷⁹ published MRCI results for Be_2 using a large active space (2s, 2p, 3s, 3p, and 3d orbitals were involved) and the aug-cc-pVQZ basis and found 804 cm^{-1} for the dissociation energy. We should also mention the study of Füsti-Molnár and Szalay, who reported an accurate all electron MR averaged quadratic coupled-cluster (MR-AQCC) study, where a modified WMR basis set⁸⁰ extended by g functions was used. Since in the dissociation limit the 2p orbitals need to be equally treated, the active space should contain the 2s and all the 2p orbitals as well, i.e., a 8×4 CAS should be used. In their study Füsti-Molnár and Szalay⁷¹ found that applying the MR-AQCC method with a symmetry breaking 4×4 CAS, where the active space contained the symmetric and anti-symmetric combination of the 2s and the $2p_z$ atomic orbitals, provides an accurate description of the Be_2 system. They estimated the dissociation energy by the 864 cm^{-1} value. A more extensive

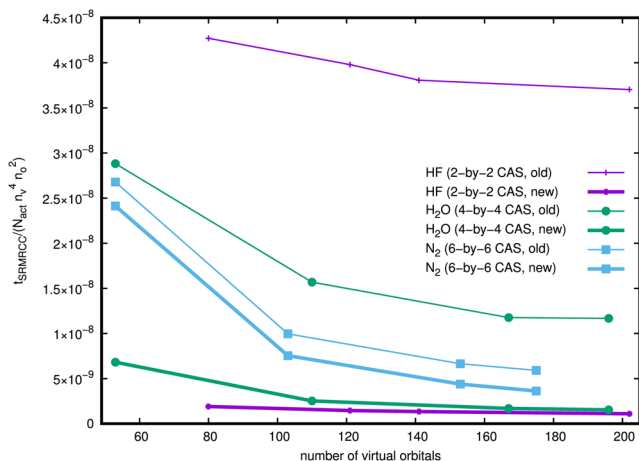


FIG. 9. Running time of the SRMRCC calculations with the previous and the new code divided by $N_{act}n_v^4n_o^2$. The result is expected to be a horizontal line in the large basis set limit. See text for more details.

review of the previous results can be found in Ref. 76 and the references therein.

Applying the frozen core approximation, where the application of large basis sets is possible, the Be dimer can be the target of FCI calculations.^{50–52} The highest level of frozen core FCI calculation available for the Be dimer was published by Helal *et al.*,⁷⁶ who reported FCI results for the lowest singlet and triplet states using the cc-pV6Z basis set.

For our study, we chose the cc-pCVQZ basis set to describe the Be₂ dimer with the all electron SRMRCC calculations. Following the observation of Füsti-Molnár and Szalay, the first SRMRCC calculations were performed using the above 4×4 CAS reference space obtained from an MCSCF calculation. Though the PES is described qualitatively well at the CCSDtqph level, the obtained dissociation energy is poor, 572 cm⁻¹, being far too low compared with the experimental 929.7 (934.9) cm⁻¹ value.^{77,78} Test calculations using the cc-pCVTZ basis set indicated that the large discrepancy comes from the symmetry breaking description of the dissociated system. To describe the 2p orbitals on an equal footing, the 2p_x and 2p_y orbitals were also correlated in the CAS space. As the determinant with the symmetric and antisymmetric combinations of the 1s and 2s atomic orbitals has the largest contribution to the wave function everywhere along the PES, this determinant was a convenient choice for the Fermi-vacuum. To keep the computation price at a moderate level with the 8×4 CAS reference space, the restricted pentuple and hexuple cluster amplitudes were eliminated, i.e., the CCSDtq approximation were used.

The potential surface is plotted in Fig. 10. The calculated bond distance is 2.46 Å, which is in a good agreement with the experimental value (2.45 Å),⁷⁷ and the dissociation energy is 828.3 cm⁻¹ containing the Boys–Bernardi counterpoise (CP) correction⁸¹ (−3.0 cm⁻¹).

As the results show that the CCSDtq method with the cc-pCVQZ basis set provides a good approximation to the dissociation energy. Some of the missing contributions can be estimated using the cc-pCVTZ basis set. To estimate the effect of neglecting the restricted pentuple and hexuple amplitudes, we compared the dissociation energy at the CCSDtq and CCSDtqph levels of theory and found that eliminating the

“small” p and h amplitudes lowers the dissociation energy by 0.8 cm⁻¹. Using the smaller basis set we also computed the difference of the dissociation energies obtained from the CCSDTQ and the CCSDtq calculations, which can be used as an approximation of the correlation energy not included at the CCSDtq level. Its value is 21.7 cm⁻¹. Including these corrections the best approximation to the dissociation energy is 850.8 cm⁻¹. In the cc-pCVQZ basis set, this value should be close to the FCI limit. Probably the main source of the remaining error with respect to the FCI is the effect of the three- and four-fold cluster amplitudes not included in CCSDtq and estimated with the cc-pCVTZ basis set. As the restricted pentuple and hexuple excitations have little contributions to the dissociation energy in the cc-pCVTZ basis, we expect that the excitations above the quadruples altogether have a small effect.

We can also use these results to estimate the contribution of the core electrons to the dissociation energy. To that end the MCSCF core orbitals were frozen in the cc-pCVQZ basis set, and the dissociation energy was recomputed using the calculated equilibrium distance. The previously published values for the core correlation contribution are in the 65–89 cm⁻¹ interval (see Ref. 76). The evaluated core correlation contribution (calculated as the difference of the CP corrected all electron and frozen core calculations) is 70.4 cm⁻¹ using the cc-pCVQZ basis set, which is in a good agreement with the value of 70 cm⁻¹ obtained by Schmidt *et al.*⁷⁵ from an MR-CISDTQ calculation using the similar cc-pwCVQZ basis set.

Applying the cc-pV6Z basis set and an extrapolation of frozen-core FCI results to the infinite basis set limit, Helal *et al.*⁷⁶ estimated the dissociation energy as 850.4 cm⁻¹. Adding our core correlation contribution to Helal’s result, we end up with 922.9 cm⁻¹, which is quite close to the experimental value [929.7 (934.9) cm⁻¹].

In conclusion, we have been able to use the CCSDtq method to describe the PES of the Be₂. We could also compute the core correlation contribution to the dissociation energy, which together with Helal’s frozen-core FCI result accurately estimates the experimental result.

VI. SUMMARY AND OUTLOOK

A new diagrammatic approach suitable for the efficient implementation of high-order CC methods and its initial application are presented. For the development of the approach, diagrams with string lines were introduced, and the diagram rules were derived for these objects. The new representation provides a compact form of the CC equations, and on this basis, excitation level independent active-space CC codes can be written. Using the new approach, we implemented the SRMRCC method where we could demonstrate that the new concept significantly increases the applicability of the active-space MRCC methods. Due to the applied simplifications, the new SRMRCC code is optimal for small CAS problems, while for larger CAS spaces the improvement is moderate. Accordingly, as the test results show, the new code is currently rather efficient for small, 2×2 and 4×4 CAS problems but not significantly better than the previous implementation for larger CAS calculations.

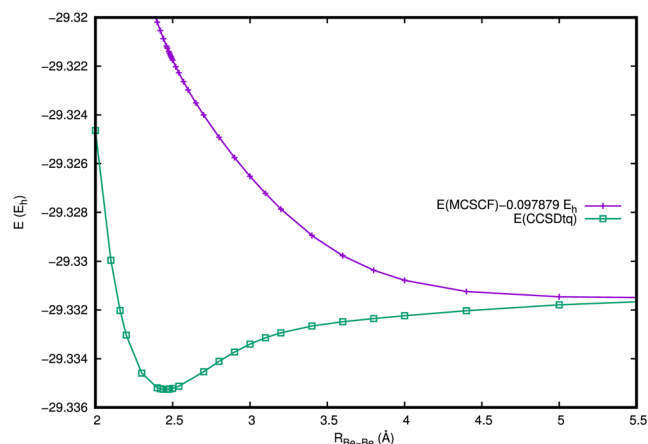


FIG. 10. Potential energy surface of Be₂ obtained from 8×4 MCSCF and CCSDtq calculations using the cc-pCVQZ basis set.

Though the active-space CC methods have obvious limitations due to the high computational price, for small CAS problems these methods can be considered as useful alternatives of the MRCI methods. To demonstrate it, a brief study of the Be dimer is also presented, where it is shown that the PES of Be₂ can be accurately described by the CCSDtq method.

Finally, considering the perspectives of our partly string-based approach, we believe that it has the potential to be applicable for the higher-order SRCC calculations as well. Although the new CC code presented is able to perform SRCC calculations at arbitrary excitation levels supposing that each orbital is active, in its recent form it is inefficient. In the current implementation, we supposed that the number of strings is relatively short, which can be reasonable in active-space MRCC applications, but for the efficient application of the new framework for the high-order SRCC case this condition needs to be relaxed. To that end the presented algorithm should be modified at two points, namely, at the derivation of the CC equations the length of the strings should be taken into account and the recent form of I/O operations belonging to the two-*T* intermediates should be optimized. More details will be presented in a forthcoming paper.

ACKNOWLEDGMENTS

Financial support has been provided by the Hungarian Scientific Research Fund (OTKA), Grant No. PD108451.

- ¹J. Čížek, *J. Chem. Phys.* **45**, 4256 (1966).
- ²J. Čížek, *Adv. Chem. Phys.* **14**, 35 (1969).
- ³G. D. Purvis III and R. J. Bartlett, *J. Chem. Phys.* **76**, 1910 (1982).
- ⁴Y. S. Lee, S. A. Kucharski, and R. J. Bartlett, *J. Chem. Phys.* **81**, 5906 (1984).
- ⁵J. Noga and R. J. Bartlett, *J. Chem. Phys.* **86**, 7041 (1987).
- ⁶S. A. Kucharski and R. J. Bartlett, *Theor. Chim. Acta* **80**, 387 (1991).
- ⁷S. A. Kucharski and R. J. Bartlett, *J. Chem. Phys.* **97**, 4282 (1992).
- ⁸N. Oliphant and L. Adamowicz, *J. Chem. Phys.* **94**, 1229 (1991).
- ⁹B. Jeziorski and H. J. Monkhorst, *Phys. Rev. A* **24**, 1668 (1981).
- ¹⁰A. Banerjee and J. Simons, *Int. J. Quantum Chem.* **19**, 207 (1981).
- ¹¹D. Datta, L. Kong, and M. Nooijen, *J. Chem. Phys.* **134**, 214116 (2011).
- ¹²M. Hanauer and A. Köhn, *J. Chem. Phys.* **134**, 204111 (2011).
- ¹³F. A. Evangelista and J. Gauss, *J. Chem. Phys.* **134**, 114102 (2011).
- ¹⁴P. K. Samanta, D. Mukherjee, M. Hanauer, and A. Köhn, *J. Chem. Phys.* **140**, 134108 (2014).
- ¹⁵M. Hanauer and A. Köhn, *J. Chem. Phys.* **136**, 204107 (2012).
- ¹⁶W. Liu, M. Hanauer, and A. Köhn, *Chem. Phys. Lett.* **565**, 122 (2012).
- ¹⁷M. Hanauer and A. Köhn, *J. Chem. Phys.* **137**, 131103 (2012).
- ¹⁸F. A. Evangelista, M. Hanauer, A. Köhn, and J. Gauss, *J. Chem. Phys.* **136**, 204108 (2012).
- ¹⁹P. Piecuch, N. Oliphant, and L. Adamowicz, *J. Chem. Phys.* **99**, 1875 (1993).
- ²⁰M. Kállay, P. G. Szalay, and P. R. Surján, *J. Chem. Phys.* **117**, 980 (2002).
- ²¹Z. Rolik and M. Kállay, *J. Chem. Phys.* **141**, 134112 (2014).
- ²²J. Olsen, *J. Chem. Phys.* **113**, 7140 (2000).
- ²³S. Hirata and R. J. Bartlett, *Chem. Phys. Lett.* **321**, 216 (2000).
- ²⁴M. Kállay and P. R. Surján, *J. Chem. Phys.* **115**, 2945 (2001).
- ²⁵M. Hanrath and A. Engels-Putzka, *J. Chem. Phys.* **133**, 064108 (2010).
- ²⁶J. Paldus and H. C. Wong, *Comput. Phys. Commun.* **6**, 1 (1973).
- ²⁷Z. Csépes and J. Pipek, *J. Comput. Phys.* **77**, 1 (1988).
- ²⁸C. L. Janssen and H. F. Schaefer III, *Theor. Chim. Acta* **79**, 1 (1991).
- ²⁹X. Li and J. Paldus, *J. Chem. Phys.* **101**, 8812 (1994).
- ³⁰M. Nooijen and V. Lotrich, *J. Mol. Struct.: THEOCHEM* **547**, 253 (2001).
- ³¹M. K. MacLeod and T. Shiozaki, *J. Chem. Phys.* **142**, 051103 (2015).
- ³²B. Vlaisavljevich and T. Shiozaki, *J. Chem. Theory Comput.* **12**, 3781 (2016).
- ³³M. Krupicka, K. Sivalingam, L. Huntington, A. A. Auer, and F. Neese, *J. Comput. Chem.* **38**, 1853 (2017).
- ³⁴S. Hirata, *Theor. Chim. Acta* **116**, 2 (2006).
- ³⁵A. Engels-Putzka and M. Hanrath, *J. Chem. Phys.* **134**, 124106 (2011).
- ³⁶S. Hirata, *J. Phys. Chem. A* **107**, 9887 (2003).
- ³⁷S. Hirata, *J. Chem. Phys.* **121**, 51 (2004).
- ³⁸K. Kowalski, S. Hirata, M. Włoch, P. Piecuch, and T. L. Windus, *J. Chem. Phys.* **123**, 074319 (2005).
- ³⁹P. Piecuch, S. Hirata, K. Kowalski, P. D. Fan, and T. L. Windus, *Int. J. Quantum Chem.* **106**, 79 (2006).
- ⁴⁰L. K. Sørensen, J. Olsen, and T. Fleig, *J. Chem. Phys.* **134**, 214102 (2011).
- ⁴¹L. K. Sørensen and J. Olsen, *Mol. Phys.* **115**, 90 (2017).
- ⁴²D. I. Lyakh, V. V. Ivanov, and L. Adamowicz, *J. Chem. Phys.* **122**, 024108 (2005).
- ⁴³E. Epifanovsky, M. Wormit, T. Kuś, A. Landau, D. Zuev, K. Khistyayev, P. Manohar, I. Kaliman, A. Dreuw, and A. I. Krylov, *J. Comput. Chem.* **34**, 2293 (2013).
- ⁴⁴J. A. Calvin, C. A. Lewis, and E. F. Valeev, in Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms-IA³ 15 (ACM Press, 2015), Article No. 4.
- ⁴⁵E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel, *J. Parallel Distrib. Comput.* **74**, 3176 (2014).
- ⁴⁶E. Deumens, V. F. Lotrich, A. Perera, M. J. Ponton, B. A. Sanders, and R. J. Bartlett, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **1**, 895 (2011).
- ⁴⁷D. Kats and F. R. Manby, *J. Chem. Phys.* **138**, 144101 (2013).
- ⁴⁸J. A. Parkhill and M. Head-Gordon, *Mol. Phys.* **108**, 513 (2010).
- ⁴⁹MRCC, a quantum chemical program suite written by M. Kállay, Z. Rolik, J. Csontos, P. Nagy, G. Samu, D. Mester, I. Ladjánszki, L. Szegedy, B. Ladóczki, K. Petrov, M. Farkas, and B. Hégyel, www.mrcc.hu.
- ⁵⁰R. J. Harrison and N. C. Handy, *Chem. Phys. Lett.* **98**, 97 (1983).
- ⁵¹G. A. Petterson and W. A. Shirley, *Chem. Phys. Lett.* **160**, 494 (1989).
- ⁵²S. Evangelista, G. L. Bendazzoli, and L. Gagliardi, *Chem. Phys.* **185**, 47 (1994).
- ⁵³P. Piecuch, S. A. Kucharski, and R. J. Bartlett, *J. Chem. Phys.* **110**, 6103 (1999).
- ⁵⁴P. Piecuch and L. Adamowicz, *J. Chem. Phys.* **100**, 5792 (1994).
- ⁵⁵L. Adamowicz, P. Piecuch, and K. B. Ghose, *Mol. Phys.* **94**, 225 (1998).
- ⁵⁶J. Shen and P. Piecuch, *J. Chem. Phys.* **136**, 144104 (2012).
- ⁵⁷J. Shen and P. Piecuch, *J. Chem. Theory Comput.* **8**, 4968 (2012).
- ⁵⁸J. Shen and P. Piecuch, *Chem. Phys.* **401**, 180 (2012).
- ⁵⁹K. Kowalski and P. Piecuch, *J. Chem. Phys.* **113**, 18 (2000).
- ⁶⁰A. Köhn and J. Olsen, *J. Chem. Phys.* **125**, 174110 (2006).
- ⁶¹S. Das, M. Kállay, and D. Mukherjee, *J. Chem. Phys.* **133**, 234110 (2010).
- ⁶²D. I. Lyakh, V. V. Ivanov and L. Adamowicz, *Mol. Phys.* **105**, 1335 (2007).
- ⁶³V. V. Ivanov, L. Adamowicz, and D. I. Lyakh, *Int. J. Quantum Chem.* **106**, 2875 (2006).
- ⁶⁴D. I. Lyakh, V. V. Ivanov, and L. Adamowicz, *Theor. Chim. Acta* **116**, 427 (2006).
- ⁶⁵J. Csontos, B. Nagy, and M. Kállay, *J. Phys. Chem. A* **117**, 5518 (2013).
- ⁶⁶V. V. Ivanov, L. Adamowicz, and D. I. Lyakh, *J. Chem. Phys.* **124**, 184302 (2006).
- ⁶⁷D. I. Lyakh, V. V. Ivanov, and L. Adamowicz, *J. Chem. Phys.* **128**, 074101 (2008).
- ⁶⁸I. Shavitt and R. J. Bartlett, *Many-Body Methods in Chemistry and Physics* (University Press, Cambridge, 2009).
- ⁶⁹T. H. Dunning, Jr., *J. Chem. Phys.* **90**, 1007 (1989).
- ⁷⁰R. A. Kendall, T. H. Dunning, Jr., and R. J. Harrison, *J. Chem. Phys.* **96**, 6796 (1992).
- ⁷¹L. Füsti-Molnár and P. G. Szalay, *Chem. Phys. Lett.* **258**, 400 (1996).
- ⁷²U. S. Mahapatra and S. Chattopadhyay, *J. Phys. B: At., Mol. Opt. Phys.* **44**, 105102 (2011).
- ⁷³R. J. Gdanitz, *Chem. Phys. Lett.* **312**, 578 (1999).
- ⁷⁴A. V. Mitin, *Int. J. Quantum Chem.* **111**, 2560 (2011).
- ⁷⁵M. W. Schmidt, J. Ivanic, and K. Ruedenberg, *J. Phys. Chem. A* **114**, 8687 (2010).
- ⁷⁶W. Helal, S. Evangelista, T. Leininger, and A. Monari, *Chem. Phys. Lett.* **568**, 49 (2012).
- ⁷⁷J. M. Merritt, V. E. Bondybey, and M. C. Heaven, *Science* **324**, 1548 (2009).
- ⁷⁸V. V. Meshkov, A. V. Stolyarov, M. C. Heaven, C. Haugen, and R. J. LeRoy, *J. Chem. Phys.* **140**, 064315 (2014).
- ⁷⁹A. Kalamos, *J. Chem. Phys.* **145**, 214302 (2016).
- ⁸⁰P.-O. Widmark, P.-Å. Malmquist, and B. O. Roos, *Theor. Chim. Acta* **77**, 291 (1990).
- ⁸¹S. F. Boys and F. Bernardi, *Mol. Phys.* **19**, 553 (1970).