Tensor-decomposed vibrational coupled-cluster theory: Enabling large-scale, highly accurate vibrational-structure calculations

Niels Kristian Madsen, Ian H. Godtlielsen, Sergio A. Losilla, and Ove Christiansen

Citation: *J. Chem. Phys.* **148**, 024103 (2018); doi: 10.1063/1.5001569

View online: <https://doi.org/10.1063/1.5001569>

View Table of Contents: <http://aip.scitation.org/toc/jcp/148/2>

Published by the [American Institute of Physics](#)

Articles you may be interested in

[Hartree–Fock symmetry breaking around conical intersections](#)

The Journal of Chemical Physics **148**, 024109 (2018); 10.1063/1.5010929

[Lowering of the complexity of quantum chemistry methods by choice of representation](#)

The Journal of Chemical Physics **148**, 044106 (2018); 10.1063/1.5007779

[Perspective: Ab initio force field methods derived from quantum mechanics](#)

The Journal of Chemical Physics **148**, 090901 (2018); 10.1063/1.5009551

[On the difference between variational and unitary coupled cluster theories](#)

The Journal of Chemical Physics **148**, 044107 (2018); 10.1063/1.5011033

[Uniform magnetic fields in density-functional theory](#)

The Journal of Chemical Physics **148**, 024101 (2018); 10.1063/1.5007300

[Non-iterative triple excitations in equation-of-motion coupled-cluster theory for electron attachment with applications to bound and temporary anions](#)

The Journal of Chemical Physics **148**, 024104 (2018); 10.1063/1.5006374

PHYSICS TODAY

WHITEPAPERS

ADVANCED LIGHT CURE ADHESIVES

Take a closer look at what these environmentally friendly adhesive systems can do

READ NOW

PRESENTED BY
MASTERBOND
ADHESIVES | SEALANTS | COATINGS

Tensor-decomposed vibrational coupled-cluster theory: Enabling large-scale, highly accurate vibrational-structure calculations

Niels Kristian Madsen,^{a)} Ian H. Godtliobsen,^{b)} Sergio A. Losilla,^{c)} and Ove Christiansen^{d)}
Department of Chemistry, Aarhus University, 8000 Aarhus C, Denmark

(Received 25 August 2017; accepted 19 December 2017; published online 9 January 2018)

A new implementation of vibrational coupled-cluster (VCC) theory is presented, where all amplitude tensors are represented in the canonical polyadic (CP) format. The CP-VCC algorithm solves the non-linear VCC equations without ever constructing the amplitudes or error vectors in full dimension but still formally includes the full parameter space of the VCC[n] model in question resulting in the same vibrational energies as the conventional method. In a previous publication, we have described the non-linear-equation solver for CP-VCC calculations. In this work, we discuss the general algorithm for evaluating VCC error vectors in CP format including the rank-reduction methods used during the summation of the many terms in the VCC amplitude equations. Benchmark calculations for studying the computational scaling and memory usage of the CP-VCC algorithm are performed on a set of molecules including thiadiazole and an array of polycyclic aromatic hydrocarbons. The results show that the reduced scaling and memory requirements of the CP-VCC algorithm allows for performing high-order VCC calculations on systems with up to 66 vibrational modes (anthracene), which indeed are not possible using the conventional VCC method. This paves the way for obtaining highly accurate vibrational spectra and properties of larger molecules. *Published by AIP Publishing.*
<https://doi.org/10.1063/1.5001569>

I. INTRODUCTION

Describing chemistry from first principles requires a quantum-mechanical description of both electrons and nuclei. Invoking the Born-Oppenheimer approximation (BOA)¹ allows for constructing a molecular *potential-energy surface* (PES) from electronic-structure calculations² which becomes the starting point for describing the nuclear motion and the associated vibrational properties. A mean-field description of the vibrational wave function can be obtained from a *vibrational self-consistent field* (VSCF) calculation,^{3–6} which serves as a reference for further, correlated vibrational-structure calculations such as *vibrational configuration interaction* (VCI),^{4,7–9} *vibrational Møller-Plesset perturbation* (VMP) *theory*,^{10–12} and *vibrational coupled cluster* (VCC) *theory*.^{9,13–16} We will here focus exclusively on our VSCF-reference-based VCC method. This VCC approach is fundamentally different from boson coupled-cluster approaches applied to the vibrational problem.^{17,18}

VCC has proven to be a highly accurate method for calculating vibrational spectra and properties of small- to medium-sized molecules at reasonable cost due to its fast convergence with respect to excitation order.^{19–21} High-order VCC calculations show steep polynomial scaling with respect to the number of vibrational modes M and the number of one-mode basis functions per mode N . The bottleneck of VCC calculations is the manipulation of multidimensional arrays (or tensors)

which constitute the free wave-function parameters (the cluster amplitudes) and the residuals of the non-linear VCC equations. With the emerging possibilities of constructing accurate PESs for larger molecules,^{22,23} it becomes increasingly important to lower the computational scaling of VCC calculations.

Earlier work from our group emphasized the potential of tensor decomposition with respect to representing the same VCC or VCI wave function with fewer parameters.^{24,25} In order to lower the scaling and reduce the memory requirements of the VCC algorithms, we have in a recent publication²⁶ introduced a new non-linear-equation solver, which solves the VCC equations with all tensors decomposed to the *canonical polyadic* (CP)^{27–29} tensor format. In this work, we present the general algorithm for calculating the VCC error vector required in the equation solver in CP format without constructing any tensors in full dimension at any stage of the calculation, and we benchmark the scaling behavior and memory requirements of the CP-VCC algorithm. We stress that the goal of the CP-VCC algorithm is to deliver results identical to the ones obtained from conventional, full-tensor VCC calculations within the numerical accuracy of the latter. Error control is essential during the CP-VCC calculations in order to obtain correct results and also for solving the non-linear equations,²⁶ and we describe how accurate results are obtained with CP decomposition. The performance of the CP-VCC algorithm depends strongly on the choice of the CP-decomposition algorithm and the method for choosing the rank of the representation.^{30–35} We will in this work present a thorough description of our tensor-decomposition algorithms.

The CP format (also known in the literature as CAN-DECOMP/PARAFAC³⁰) has been applied in many contexts, e.g., analysis of 3-dimensional fluorescence spectra,^{36,37} and

^{a)}Electronic mail: nielskm@chem.au.dk

^{b)}Electronic mail: ian@chem.au.dk

^{c)}Electronic mail: sergio@chem.au.dk

^{d)}Electronic mail: ove@chem.au.dk

it has recently gained interest in the field of quantum chemistry.^{38–45} It has been applied to VCC calculations^{24,25} and later to other types of vibrational-structure calculations^{46–48} as well as for fitting PESs.^{49,50} In our previous work of Refs. 24 and 25, we showed the perspectives of representing the VCC wave function in terms of CP tensors and that sufficient accuracy could be obtained. The new CP-VCC implementation keeps all tensors in CP format throughout the entire calculation which enables us to reduce the computational cost and memory consumption of VCC calculations. Devising an algorithm which is able to capitalize on these computational benefits is, however, highly non-trivial and we will in this paper describe the details of our CP-VCC implementation. The main underlying principle in the CP-VCC algorithm is that the rank of a given tensor is determined as needed for representing the different components (amplitudes, error vectors, intermediates, etc.) to the necessary accuracy. This differs from other approaches where the rank is given as input and the effect on error is investigated.⁴⁷ The accuracies needed in the CP-VCC algorithm depend on the convergence thresholds for the non-linear-equation solver and not on the molecule in question which makes the CP-VCC algorithm black-box in nature.

The paper is structured as follows. Section II provides an overview of the VCC model and the bottlenecks of full-tensor VCC calculations. Section III introduces the concept of tensor decomposition in the context of VCC, and Sec. IV describes the rank-reduction algorithms used in the CP-VCC algorithm. The computational details and the results of our benchmark calculations are presented in Sec. V, and finally, Sec. VI provides a summary and future outlook.

II. VIBRATIONAL-STRUCTURE THEORY

The vibrational motion of a non-linear molecule with N atoms is described in terms of $M = 3N - 6$ degrees of freedom denoted vibrational *modes*. These are represented by the coordinates $q_1, \dots, q_M$. Introducing a one-mode basis set $\{\phi_{r^m}^m(q_m)\}$ with $r^m = 0, 1, \dots, N^m - 1$ for each mode allows for parameterizing the exact wave function in terms of Hartree products of one-mode functions, $\Phi_{\mathbf{r}}(q_1, q_2, \dots, q_M) = \prod_{m=1}^M \phi_{r^m}^m(q_m)$. The one-mode functions are denoted *modals* and are analogous to orbitals in electronic-structure theory. The Hartree-product ansatz does not include any symmetrization of the wave function with respect to permutation of the nuclei. This fundamental symmetry is neither enforced nor broken, and as in many other molecular-dynamics methods, we consider the system as being composed of M distinguishable degrees of freedom.^{3–5,51} Thus, we should not expect any permutational symmetry of the VCC wave-function parameters (unlike the case of cluster amplitudes of electronic-structure theory).

In *second quantization* (SQ), Hartree products are represented as *occupation-number vectors* (ONVs),⁹ and the state with no occupation is denoted as the *vacuum state* $|\text{vac}\rangle$. All states and operators are then expressed in terms of creation and annihilation operators, $a_{r^m}^{m\dagger}$ and $a_{r^m}^m$, which add or remove occupation from the modal r^m of mode m and satisfy the commutation relations,

$$[a_{r^m}^{m\dagger}, a_{s^{m'}}^{m'\dagger}] = [a_{r^m}^m, a_{s^{m'}}^{m'}] = 0, \quad (1a)$$

$$[a_{r^m}^m, a_{s^{m'}}^{m'\dagger}] = \delta_{mm'} \delta_{r^m s^{m'}}. \quad (1b)$$

The vibrational states that describe the nuclear motion of the molecule are obtained by solving the time-independent Schrödinger equation for the vibrational Hamiltonian $H = T_n + V$, where T_n is the nuclear-kinetic-energy operator which can include corrections from the Watson Hamiltonian,⁵² and $V = E^e(\{q_m\})$ is the potential-energy surface (PES). In order to reduce the computational complexity of optimizing the vibrational wave function, the Hamiltonian is represented as a *sum-over-products* (SOPs) of one-mode operators $h^{mo^{mm}}$

$$= \sum_{r^m s^m} h_{r^m s^m}^{mo^{mm}} a_{r^m}^{m\dagger} a_{s^m}^m,$$

$$H = \sum_t c_t \prod_{m \in \mathbf{m}_t} h^{mo^{mm}}, \quad (2)$$

where the o^{mm} index denotes the operator type (e.g., $q_m, q_m^2, q_m^3, \dots$) of mode m in term t . This reduces operations with the Hamiltonian to an array of one-index transformations, which can be evaluated very efficiently if the wave-function parameters are represented as CP tensors as described in Sec. III B.

A. The VCC model

The VCC wave-function ansatz is given in terms of the cluster operator T and a reference state $|\Phi_i\rangle = \prod_{m=1}^M a_{i^m}^{m\dagger} |\text{vac}\rangle$ which is usually obtained from a VSCF calculation,

$$|\text{VCC}\rangle = \exp(T)|\Phi_i\rangle, \quad (3)$$

where i^m indicates the occupied modal for mode m . The cluster operator is a linear combination of excitation operators $\tau_{\mu^m}^{\mathbf{m}}$

$$= \prod_{m \in \mathbf{m}} a_{\mu^m}^{m\dagger} a_{i^m}^m,$$

$$T = \sum_{\mathbf{m} \in \text{MCR}[T]} T^{\mathbf{m}} = \sum_{\mathbf{m} \in \text{MCR}[T]} \sum_{\mu^{\mathbf{m}}} t_{\mu^{\mathbf{m}}}^{\mathbf{m}} \tau_{\mu^{\mathbf{m}}}^{\mathbf{m}}, \quad (4)$$

where the coefficients $t_{\mu^{\mathbf{m}}}^{\mathbf{m}}$ are denoted as the VCC *amplitudes* and $\mu^{\mathbf{m}}$ is a compound index containing the indices of the excited modals defining the excitation for a given *mode combination* (MC) $\mathbf{m}$, e.g., $\mu^{\{m_1 m_2 m_3\}} = \{a^{m_1}, a^{m_2}, a^{m_3}\}$.

The cluster operator is a sum of cluster operators for each MC included in the *mode-combination range* (MCR) of the cluster operator T . We follow the convention that i^m refers to the modal occupied in the reference VSCF state, while a^m, b^m refer to virtual modals and r^m, s^m are modals with unspecified occupancy. Since there is always one occupied modal per mode in the VSCF reference, we define the number of virtual modals $N_{\text{vir}}^m = N^m - 1$ for later reference.

Requiring the VCC wave-function ansatz to satisfy the time-independent Schrödinger equation results in an expression for the energy as well as a set of non-linear equations for determining the amplitudes,¹³

$$E_{\text{VCC}} = \langle \Phi_i | \exp(-T) H \exp(T) | \Phi_i \rangle = \langle \Phi_i | H \exp(T) | \Phi_i \rangle, \quad (5)$$

$$e_{\mu^{\mathbf{m}}}^{\mathbf{m}} \equiv \langle \mu^{\mathbf{m}} | \exp(-T) H \exp(T) | \Phi_i \rangle = 0. \quad (6)$$

Truncating the MCR of the cluster operator at n -mode excitations defines the VCC[n] hierarchy of approximate wave-function methods. In addition, efficient approximate

treatments of 3- and 4-mode excitations can be obtained from a perturbational analysis leading to the VCC[2pt3]⁵³ and VCC[3pt4]⁵⁴ models.

The VCC amplitude equations [Eq. (6)] are solved using iterative techniques such as Newton-Raphson or quasi-Newton methods. In this paper, we use the *conjugate residual with optimal trial vectors* (CROP) algorithm⁵⁵ as described in Ref. 26, which allows both full tensors and decomposed tensors to be used.

The main bottleneck of any VCC calculation is the calculation of the error vector $e_{\mu^{\mathbf{m}}}^{\mathbf{m}}$ for a given set of amplitudes. Because the vibrational Hamiltonian is not limited to two-mode couplings like its electronic counterpart,² the exact expressions for calculating the error vector contain many terms and are thus in our framework derived and evaluated automatically.¹⁴ The challenge in the CP-VCC implementation is to handle these thousands of terms efficiently using the CP tensor format. This will be discussed further in Sec. III, but first we need to describe a few important elements of the VCC implementation presented in Ref. 14 which will be denoted here as the full-tensor VCC (FT-VCC) algorithm.

B. The FT-VCC algorithm

The algorithm for evaluating the VCC error vector consists of an outer loop that runs over the MCs included in the cluster operator. For each MC, $\mathbf{m} \in \text{MCR}[T]$ the terms that contribute to $e^{\mathbf{m}}$ are evaluated and added to the result. Each of these terms gives rise to a series of contractions and direct products, and the computational scaling of a given VCC[n] model is determined by the scaling of the most expensive term. The terms are matrix elements of operator products between the Hamiltonian and cluster operators for different MCs [Eq. (39) of Ref. 14],

$$e_{\mu^{\mathbf{m}}}^{\mathbf{m}} \leftarrow c_{\text{BCH}} \langle \mu^{\mathbf{m}} | \prod_k^{N^L} T_{n_k^L} H_{n_H} \prod_l^{N^R} T_{n_l^R} | \Phi_i \rangle. \quad (7)$$

The important thing to notice in Eq. (7) is that evaluating the error-vector terms is a matter of applying the vibrational Hamiltonian to states generated by the cluster operator acting

on the reference state,

$$|t^{\mathbf{m}}\rangle = T^{\mathbf{m}}|\Phi_i\rangle = \sum_{\mu^{\mathbf{m}}} t_{\mu^{\mathbf{m}}}^{\mathbf{m}} |\mu^{\mathbf{m}}\rangle. \quad (8)$$

The terms that need to be evaluated can be divided into types based on four basic contraction types. These are defined by splitting the sum in each one-mode operator of the Hamiltonian into the following parts:^{14,16}

$$\begin{aligned} h^{mo^{\mathbf{m}}} &= \left(h_{i^{\mathbf{m}}i^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{i^{\mathbf{m}}}^{m\dagger} a_{i^{\mathbf{m}}}^m \right) + \left(\sum_{a^{\mathbf{m}}} h_{a^{\mathbf{m}}i^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{a^{\mathbf{m}}}^{m\dagger} a_{i^{\mathbf{m}}}^m \right) \\ &+ \left(\sum_{a^{\mathbf{m}}} h_{i^{\mathbf{m}}a^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{i^{\mathbf{m}}}^{m\dagger} a_{a^{\mathbf{m}}}^m \right) + \left(\sum_{a^{\mathbf{m}}b^{\mathbf{m}}} h_{a^{\mathbf{m}}b^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{a^{\mathbf{m}}}^{m\dagger} a_{b^{\mathbf{m}}}^m \right) \\ &= h_p^{mo^{\mathbf{m}}} + h_u^{mo^{\mathbf{m}}} + h_d^{mo^{\mathbf{m}}} + h_f^{mo^{\mathbf{m}}}. \end{aligned} \quad (9)$$

Applying the four different one-mode operators to the state in Eq. (8) defines four types of contractions: *passive*, *up*, *down*, and *forward*. Since the Hamiltonian is given as a SOP of one-mode operators, each term in H [Eq. (2)] gives rise to a series of different contractions, e.g.,

$$\begin{aligned} &\left(h_f^{m_0o^{m_0}} h_f^{m_2o^{m_2}} h_d^{m_3o^{m_3}} \right) (T^{m_0m_1} T^{m_2m_3}) |\Phi_i\rangle \\ &= \left(h_f^{m_0o^{m_0}} T^{m_0m_1} \right) \left(h_f^{m_2o^{m_2}} h_d^{m_3o^{m_3}} T^{m_2m_3} \right) |\Phi_i\rangle. \end{aligned} \quad (10)$$

This example corresponds to performing a forward contraction on mode m_0 of the amplitude tensor $t^{\{m_0,m_1\}}$ as well as a down contraction on mode m_3 followed by a forward contraction on mode m_2 on the tensor $t^{\{m_2,m_3\}}$.

1. VCC tensor contractions

The different types of contractions are only non-zero in the VCC equations when certain conditions are fulfilled by the MC $\mathbf{m}$ of the $|t^{\mathbf{m}}\rangle$ state and the mode m of the one-mode operator $h^{mo^{\mathbf{m}}}$. Table I lists the four types of contractions together with their computational cost, non-zero conditions, and optimal evaluation order. Furthermore, the up, down, and forward contractions are illustrated in Fig. 1 where $\times_i$ denotes contraction along mode i . The passive contraction is not depicted as this simply corresponds to multiplying all elements of the

TABLE I. Properties of the four types of VCC contractions. For the up, down, and forward contractions, the evaluation order is important since up and down contractions change the dimensionality of the $t^{\mathbf{m}}$ tensor and thereby the cost of the following operations. The scalings represent the computational cost of performing the contraction in the FT-VCC algorithm.

Contraction	Operator	Condition	Scaling	Evaluation order	Description
Passive	$h_p^{mo^{\mathbf{m}}} = h_{i^{\mathbf{m}}i^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{i^{\mathbf{m}}}^{m\dagger} a_{i^{\mathbf{m}}}^m$	$m \notin \mathbf{m}$	N^D	...	Multiply $t^{\mathbf{m}}$ with the $h_{i^{\mathbf{m}}i^{\mathbf{m}}}^{mo^{\mathbf{m}}}$ integral.
Down	$h_d^{mo^{\mathbf{m}}} = \sum_{a^{\mathbf{m}}} h_{i^{\mathbf{m}}a^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{i^{\mathbf{m}}}^{m\dagger} a_{a^{\mathbf{m}}}^m$	$m \in \mathbf{m}$	$2N^D$	1	De-excitation. Always evaluated first since the dimensionality of $t^{\mathbf{m}}$ is lowered.
Forward	$h_f^{mo^{\mathbf{m}}} = \sum_{a^{\mathbf{m}}b^{\mathbf{m}}} h_{a^{\mathbf{m}}b^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{a^{\mathbf{m}}}^{m\dagger} a_{b^{\mathbf{m}}}^m$	$m \in \mathbf{m}$	$2N^{D+1}$	2	Move occupation between virtual modals.
Up	$h_u^{mo^{\mathbf{m}}} = \sum_{a^{\mathbf{m}}} h_{a^{\mathbf{m}}i^{\mathbf{m}}}^{mo^{\mathbf{m}}} a_{a^{\mathbf{m}}}^{m\dagger} a_{i^{\mathbf{m}}}^m$	$m \notin \mathbf{m}$	N^{D+1}	3	Excitation. Always evaluated last since the dimensionality of $t^{\mathbf{m}}$ is increased.

FIG. 1. Contractions used in the FT-VCC algorithm. A change of colour denotes that the tensor elements have been modified. (a) Down contraction, (b) forward contraction, and (c) up contraction.

tensor by a scalar. As an example, the forward contraction of Fig. 1(b) corresponds to the operation,

$$r_{a^{m_1} a^{m_2} b^{m_3}}^{\mathbf{m}} = \sum_{a^{m_3}=1}^{N_{\text{vir}}^{m_3}} h_{b^{m_3} a^{m_3}}^{m_3} t_{a^{m_1} a^{m_2} a^{m_3}}^{\mathbf{m}}, \quad (11)$$

for the MC $\mathbf{m} = \{m_1, m_2, m_3\}$, which in this 3rd-order case requires $N^3(2N-1)$ floating-point operations (for $N_{\text{vir}}^{m_1} = N_{\text{vir}}^{m_2} = N_{\text{vir}}^{m_3} = N$). It is seen in Fig. 1 how all contraction types require operations on all elements of $t^{\mathbf{m}}$ when using full tensors. We will see in Sec. III that this is avoided if the VCC amplitudes are represented as CP tensors.

Another important operation related to the contractions is direct products between tensors. These are needed in order to construct the final error vectors from individually transformed amplitude tensors of lower order.^{14,15} Using full tensors, these scale as N^D where D is the dimensionality of the resulting tensor.

2. Automatic identification of intermediates

The general VCC implementation includes automatic identification of intermediates that can lower the computational scaling of the VCC models using an elaborate scheme detailed further in Ref. 14. These intermediates are instrumental in obtaining scalings which are comparable to the VCI calculation of a corresponding excitation level. The algorithm finds the optimal intermediates by removing sum restrictions, generating all possible intermediates, and comparing their scaling. In the VCC algorithm of Ref. 14, the M scaling is considered most important followed by the O (number of one-mode operators per mode in the Hamiltonian) and N scalings. Apart from intermediates that include the Hamiltonian coefficients, the algorithm is also able to store and reuse the so-called t intermediates, which are amplitude tensors that have been down or forward contracted. The algorithm only stores intermediates that have been down contracted at least once (such that they are of lower dimensionality) and have not been forward contracted (otherwise, there would be many).

III. TENSOR DECOMPOSITION AND THE CP-VCC ALGORITHM

Reducing the computational cost of the VCC contractions and direct products as well as the memory requirements of the intermediates is instrumental for applying high-order VCC models to larger systems. Both of these bottlenecks can be addressed by representing the amplitudes and error vectors in the CP tensor format.

A. The CP tensor format

The VCC amplitudes and error vectors can be viewed as a set of individual tensors, i.e., $\mathbf{t} = \{t^{\mathbf{m}}\}$ and $\mathbf{e} = \{e^{\mathbf{m}}\}$, where $t^{\mathbf{m}}$ is the amplitude tensor of the MC $\mathbf{m}$.^{24–26} The VCC wave function is thus parametrized in terms of many small tensors instead of one large tensor of coefficients for each state as used in other approaches.⁴⁶ The order or dimensionality of the tensor is equal to the number of modes in the MC, and the cost of the VCC tensor contractions and direct products scales very steeply with respect to the order of the amplitude tensors. The same is true for the memory requirements of the VCC algorithm—especially for large molecules, as the number of n th-order amplitude tensors is equal to $\binom{M}{n}$.

By decomposing all tensors to the CP format, both the computational cost and the storage requirements can be reduced. The CP format represents a tensor $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{D^{\mathcal{F}}}}$ of order $D^{\mathcal{F}}$ with extents $I_1, I_2, \dots, I_{D^{\mathcal{F}}}$ as a sum of vector outer products^{27–30} as illustrated in Fig. 2,

$$\mathcal{F} \approx \sum_{r=1}^{R^{\mathcal{F}}} f_r^1 \otimes f_r^2 \otimes \dots \otimes f_r^{D^{\mathcal{F}}}. \quad (12)$$

The element-wise expression is given as

$$f_{i_1 i_2 \dots i_{D^{\mathcal{F}}}} \approx \sum_{r=1}^{R^{\mathcal{F}}} f_{i_1 r}^1 f_{i_2 r}^2 \dots f_{i_{D^{\mathcal{F}}} r}^{D^{\mathcal{F}}} = \sum_{r=1}^{R^{\mathcal{F}}} \prod_{d=1}^{D^{\mathcal{F}}} f_{i_d r}^d, \quad (13)$$

and the so-called *mode matrices* of $\mathcal{F}$ are defined as $\mathbf{F}^d = [f_1^d, \dots, f_{R^{\mathcal{F}}}^d]$ with elements $f_{i_d r}^d$. The value of $R^{\mathcal{F}}$ for which

FIG. 2. The CP representation of a full 3rd-order tensor as a sum of outer products.

the CP representation is exact is denoted as the *rank* of $\mathcal{F}$. Determining the rank of a tensor, i.e., finding $R^{\mathcal{F}}$ for which Eq. (12) is exact, is a non-deterministic polynomial-time (NP)-hard problem.^{30,56} However, in order to solve the VCC equations using the CP-VCC algorithm, we are only interested in determining low-rank approximations to a specific (absolute) accuracy T_{CP} , i.e., fitting a tensor $\mathcal{G}$ to a low-rank approximation $\mathcal{F}$ such that $\|\mathcal{F} - \mathcal{G}\| < T_{\text{CP}}$ (see Sec. IV), where by $\|\mathcal{F}\|$ we mean the Frobenius norm of the tensor $\mathcal{F}$. The structure of the VCC amplitudes and error vectors allows us to choose the accuracies individually for each MC as discussed in Ref. 26.

When the tensors can be fitted to low ranks, the N scaling of tensor contractions and direct products as well as the memory consumption is reduced significantly as discussed later in Sec. III B. VCC is a size-consistent wave-function model, which implies that the amplitudes for the interactions between non-interacting subsystems are exactly zero resulting in rank-0 amplitude and error-vector tensors. Consider a system of two non-interacting subsystems A and B with modes $\{m_1^A, m_2^A\}$ and $\{m_1^B, m_2^B\}$. In the VCC wave function, the amplitude tensors that couple the two subsystems are exactly zero and thereby of rank 0, e.g., $t^{m_1^A m_1^B} = \mathbf{0}$. The same applies to higher-order amplitude tensors where simultaneous excitations in both subsystems are described exactly by direct products between lower-order tensors. As an example, $t^{m_1^A m_2^A m_1^B} = \mathbf{0}$, while $t^{m_1^A m_2^A} \otimes t^{m_1^B} \neq \mathbf{0}$ and $t^{m_1^A} \otimes t^{m_2^A} \otimes t^{m_1^B} \neq \mathbf{0}$.

As we turn on interactions between the subsystems, as in real molecular systems, matters become more complicated, but the norms of the amplitude tensors describing weak interactions are small compared to the norms of tensors representing important couplings. This means that the approximate ranks of amplitude tensors describing weak and spatially separated interactions are small when the tensors are fitted to an absolute threshold as also shown numerically in Ref. 24. In the CP-VCC algorithm, the computational effort is thus adapted dynamically to the strength of the physical interactions between the vibrational modes. As the number and ratio of weak mode couplings is expected to increase with the size of the molecule due to spacial separation, there is much to be gained in computational efficiency when going to larger systems. This opens for the possibility of effectively reducing the M scaling of the VCC models, both in terms of memory and computational cost, and thereby for performing large-scale VCC calculations.

TABLE II. Scalings of the four contraction types using full tensors and CP tensors. N is the size of the modal basis, and D and R are the dimensionality and rank of the cluster amplitude tensor, respectively.

Contraction	Scaling	
	Full tensors	CP tensors
Passive	N^D	RN^a
Down	$2N^D$	$2RN^a$
Forward	$2N^{D+1}$	$2RN^2$
Up	N^{D+1}	0^b

^aFor convenience, we sometimes distribute the norm of the tensors on all mode vectors after performing the contraction, which scales as RND .

^bWith CP tensors, up contractions only require copying data.

B. Computational benefits of the CP format

The computational benefits of the CP format originate from the fact that the full multi-dimensional array $\mathcal{F}$ is decomposed to a set of $D^{\mathcal{F}}$ matrices, each corresponding to a mode (or index) of the tensor. Thereby, the storage requirements scale linearly instead of exponentially with respect to the tensor order (assuming constant $R^{\mathcal{F}}$). Because of the separation of indices, general multi-dimensional contractions are reduced to a set of matrix multiplications and the scalings of the VCC contractions described in Sec. II B 1 are reduced as shown in Table II. The contractions needed when performing CP-VCC are illustrated in Fig. 3. Returning to the example presented in Eq. (11), the forward contraction illustrated in Fig. 3(b) corresponds to

$$r_{a^{m_1} a^{m_2} b^{m_3}}^{\mathbf{m}} = \sum_{a^{m_3}=1}^{N_{\text{vir}}^{m_3}} h_{b^{m_3} a^{m_3}}^{m_3} \sum_{r=1}^R t_{a^{m_1} r}^{\mathbf{m}, m_1} t_{a^{m_2} r}^{\mathbf{m}, m_2} t_{a^{m_3} r}^{\mathbf{m}, m_3} = \sum_{r=1}^R t_{a^{m_1} r}^{\mathbf{m}, m_1} t_{a^{m_2} r}^{\mathbf{m}, m_2} \left(\sum_{a^{m_3}=1}^{N_{\text{vir}}^{m_3}} h_{b^{m_3} a^{m_3}}^{m_3} t_{a^{m_3} r}^{\mathbf{m}, m_3} \right) \quad (14)$$

requiring $NR(2N - 1)$ floating-point operations. It is clear that the computational complexity of the forward contraction is reduced since the forward contraction with CP tensors corresponds to a standard matrix multiplication. The up and down contractions illustrated in Figs. 3(c) and 3(a) are performed as

$$r_{a^{m_1} a^{m_2} a^{m_3}}^{\mathbf{m}} = h_{a^{m_3} i^{m_3}}^{m_3} \sum_{r=1}^R t_{a^{m_1} r}^{\mathbf{m}, m_1} t_{a^{m_2} r}^{\mathbf{m}, m_2} = \sum_{r=1}^R t_{a^{m_1} r}^{\mathbf{m}, m_1} t_{a^{m_2} r}^{\mathbf{m}, m_2} r_{a^{m_3} r}^{\mathbf{m}, m_3}, \quad (15)$$

FIG. 3. Contractions used in the CP-VCC algorithm. The $\langle \cdot | \cdot \rangle$ notation denotes a dot product between mode vectors. A change of colour denotes that the tensor elements have been modified. (a) Down contraction, (b) forward contraction, and (c) up contraction.

with $r_{a^{m_3}r}^{\mathbf{m},m_3} = h_{a^{m_3}i^{m_3}}^{m_3 o^{m_3}} \forall r$, and

$$\begin{aligned} r_{a^{m_1}a^{m_2}}^{\mathbf{m}} &= \sum_{a^{m_3}=1}^{N_{vir}^{m_3}} h_{i^{m_3}a^{m_3}}^{m_3 o^{m_3}} \sum_{r=1}^R t_{a^{m_1}r}^{\mathbf{m},m_1} t_{a^{m_2}r}^{\mathbf{m},m_2} t_{a^{m_3}r}^{\mathbf{m},m_3} \\ &= \sum_{r=1}^R \left(\omega_r^{1/2} t_{a^{m_1}r}^{\mathbf{m},m_1} \right) \left(\omega_r^{1/2} t_{a^{m_2}r}^{\mathbf{m},m_2} \right), \end{aligned} \quad (16)$$

with $\omega_r = \sum_{a^{m_3}=1}^{N_{vir}^{m_3}} h_{i^{m_3}a^{m_3}}^{m_3 o^{m_3}} t_{a^{m_3}r}^{\mathbf{m},m_3}$, respectively. The passive contraction is simply performed by multiplying all elements of all mode matrices by $(h_{i^{m_3}i^{m_3}}^{m_3 o^{m_3}})^{1/D}$. It is seen from Table II that the most expensive contraction scales as $\mathcal{O}(N^2)$ when using CP tensors (for a fixed rank R) which especially for high-order VCC models is a major reduction. The scaling is linear with respect to R , and it is therefore important to keep the ranks of the $t^{\mathbf{m}}$ tensors as low as possible.

C. The general CP-VCC algorithm

The CP-VCC algorithm has been implemented in the general VCC framework described in Sec. II A and solves the same equations to the same accuracy as the FT-VCC algorithm (within the numerical thresholds of the non-linear-equation solver). In this section, we discuss the important differences between the algorithms as well as a few notable details of the implementation. The main points are summarized in Table III, and an overview of the algorithm is given in Table IV.

The matrix elements of Eq. (7) which constitute the terms of the VCC equations are evaluated by performing tensor contractions and direct products. The resulting contributions to the VCC error vector are then given in terms of either CP tensors or direct products of CP tensors,

$$\mathcal{G} = \bigotimes_j \mathcal{G}_j, \quad (17)$$

where the rank is equal to the product of individual ranks $R^{\mathcal{G}} = \prod_j R^{\mathcal{G}_j}$. Using full tensors, the distinction between tensors and direct products is irrelevant since the direct products are simply evaluated explicitly (scaling as $\mathcal{O}(N^D)$) and added to the result. In the CP-VCC algorithm, the contributions to

TABLE IV. Overview of the main steps of calculating the error vector in the CP-VCC algorithm.

• Loop over MCs $\mathbf{m} \in \text{MCR}[T]$.   1. Evaluate the terms of the VCC equations [Eq. (7)] that contribute to $e^{\mathbf{m}}$. 2. Append the terms to a tensor sum.   (a) Recompress direct products before adding them to the sum (Fig. 4). (b) Ignore terms with norms smaller than $T_{\text{VCC}}^{\text{screen}}$. 3. Recompress the sum when $R > R_{\text{VCC}}^{\text{max}}$ (Fig. 5).

the error vector are appended to a sum of CP tensors which increases the rank of the resulting error vector after each term. This makes it necessary to perform rank reductions (or recompressions) in order to keep the memory consumption low. Figures 4 and 5 show the two types of recompressions used in the CP-VCC algorithm: direct-product recompression and sum recompression. Both operations fit the tensor to a low-rank CP tensor and the algorithms are specialized to the different tensor formats as discussed in Sec. IV. In Fig. 4, a simple example of a direct-product recompression is shown. Here, the first tensor is simply a vector ($R_1 = 1$) and thus the rank of the direct product would be quite small before the recompression. Direct products of higher dimension may contain several matrices and higher-order tensors resulting in large product ranks. In those cases, the ranks can be reduced quite significantly by the recompression of the direct product. For recompression of the tensor sum, we have implemented a dynamic scheme where a rank reduction is performed as soon as the accumulated rank exceeds a pre-defined maximum allowed rank, $R_{\text{VCC}}^{\text{max}}$. If the allowed rank is small, a large number of recompressions need to be performed, which can become quite expensive. However, if $R_{\text{VCC}}^{\text{max}}$ is too large (the extreme would be to only recompress the sum after evaluating all terms in the error vector), the memory consumption explodes and the individual recompressions become more expensive. In our experience, $R_{\text{VCC}}^{\text{max}} \sim 1000$ performs well in all test cases, but the exact choice is not critical as long as the extremes are avoided (see Sec. V C for

TABLE III. Important similarities and differences between the FT-VCC algorithm and the CP-VCC algorithm.

Similarities	
<ul style="list-style-type: none"> • The VCC equations are not changed and the energy and wave function are obtained to the same accuracy within the numerical thresholds of the non-linear-equation solver. 	
Differences	
FT-VCC	CP-VCC
<ul style="list-style-type: none"> • The bottlenecks are contractions and direct products. • All tensors are represented exactly. • Direct products are performed explicitly. • The error-vector terms are added explicitly. 	<ul style="list-style-type: none"> • The bottleneck is recompression of the error-vector sums. • The accuracies of the tensors can be chosen individually during the iterations of the VCC solver to lower the computational cost of recompressions. • Direct products are performed implicitly via recompression. • The terms are appended to a general tensor sum for later recompression.

FIG. 4. Recompression of direct products of CP tensors. Note that the rank of the direct product is equal to the product of the ranks of the individual CP tensors. This number can become quite large for high-order tensors where the sub-tensors are not only vectors and matrices.

FIG. 5. Recompression of sums of CP tensors.

numerical results). Recall that the maximum allowed rank is per mode combination. For a single MC, a rank-1000 tensor (of typical order $D \in [3, 6]$) containing $1000 \times D \times N_{vir}^m$ elements with $N_{vir}^m \sim 10$) is rather small compared to a full-dimensional CP tensor ($1000 \times M \times N_{vir}^m$) or full tensor ($(N_{vir}^m)^M$). There are of course many MCs in the VCC error vector, but they are not all treated simultaneously. In order to slow down the rank growth further and enhance the numerical stability of the rank-reduction algorithms, we have also introduced a screening based on the norm of the individual terms. If the norm is smaller than a given threshold T_{VCC}^{screen} , the contribution is neglected. In this manner, contributions that have no influence on the result (at this stage of the calculation) will also not contribute to the considerable time used in recompressions.

We choose to recompress all direct products before adding them to the sum which means that the error-vector sum contains only CP tensors as depicted in Fig. 5, but we could also choose to include direct products in the sum and perform recompressions on a general sum containing different tensor formats. The main reason for this choice is that it enables us to use the *canonical-to-Tucker* (C2T) rank-reduction algorithm⁵⁷ described in Sec. IV A for recompressing the sum, and we have also seen that it provides a speedup on its own compared to including direct products in the sum. This is probably due to the fact that direct products are often fitted to quite low ranks which slow down the rank growth of the total tensor sum. By using this scheme for evaluating all direct products, we effectively replace the comparatively expensive direct-product evaluation of the VCC algorithm with a rank reduction.

Because the scalings of the VCC contractions are different with CP tensors compared to full tensors (see Sec. III B), one could try to identify intermediates based on the reduced scalings. In this context, it is important to note, however, that the bottleneck of the CP-VCC algorithm is shifted from performing contractions and direct products to recompressing the error vectors. Therefore, it is not straightforward (if at all possible) to devise an improved set of criteria for determining the optimal intermediates and we choose to use the intermediates obtained from the VCC framework of Ref. 14.

IV. CP DECOMPOSITION AND RECOMPRESSION

In the CP-VCC algorithm, the main bottleneck is the recompression of the sum of terms during the calculation of the error vector. Therefore, the performance of the rank-reduction algorithm both in terms of accuracy, speed, and final tensor ranks is crucial to the performance of our implementation. We have implemented an extensive tensor-decomposition framework (see the [supplementary material](#)), which includes an array of different methods for finding the best rank- R approximation, fitting a tensor to a given accuracy, constructing accurate starting guesses for the optimizations, etc. The algorithms have been optimized for recompressions of CP tensors and direct products of CP tensors.

In order to solve the VCC equations using the CROP algorithm, the error vectors and amplitudes in each iteration need a certain accuracy $\|e_{CP}^m - e^m\| < T_{CP}^m$ as discussed in Ref. 26. The accuracy to which we represent the VCC amplitudes in a given iteration is determined in a dynamic way relative to the step length of the non-linear-equation solver such that the noise is per default at least two orders of magnitude smaller than the norm of the amplitude update. The accuracies of the individual error-vector tensors are then determined such that the amplitude update becomes as accurate as the amplitudes. We therefore need an efficient algorithm for recompressing a tensor to a given accuracy, i.e., solving the *extended approximation problem*,³²

$$J(\mathcal{F}_\epsilon) \leq \epsilon, \quad (18a)$$

$$J(\mathcal{F}_\epsilon) = \min_{\mathcal{F}} J(\mathcal{F}), \quad (18b)$$

for the objective function,

$$J(\mathcal{F}) = \frac{1}{2} \|\mathcal{F} - \mathcal{G}\|^2, \quad (19)$$

where the *fitting tensor* $\mathcal{F}$ is a low-rank approximation to the *target tensor* $\mathcal{G}$. $\mathcal{F}$ can be optimized to satisfy the optimality condition for a fixed rank [Eq. (18b)] by searching for a stationary point of $J(\mathcal{F})$. The gradient of the objective function is given as

$$\begin{aligned}
\frac{\partial J(\mathcal{F})}{\partial f_{\tilde{r}}^d} &= \frac{\partial}{\partial f_{\tilde{r}}^d} \left(\frac{1}{2} \|\mathcal{F}\|^2 + \frac{1}{2} \|\mathcal{G}\|^2 - \langle \mathcal{F} | \mathcal{G} \rangle \right) \\
&= \left\langle \frac{\partial \mathcal{F}}{\partial f_{\tilde{r}}^d} \middle| \mathcal{F} \right\rangle - \left\langle \frac{\partial \mathcal{F}}{\partial f_{\tilde{r}}^d} \middle| \mathcal{G} \right\rangle \\
&= [\mathbf{F}^{\tilde{d}} \mathbf{\Gamma}^{\tilde{d}} - \mathbf{Q}^{\tilde{d}}]_{\tilde{r}}, \tag{20}
\end{aligned}$$

where we denote $\mathbf{\Gamma}^d$ as the *gamma matrix* and $\mathbf{Q}^d$ as the *right-hand side* (RHS) of mode d . The gamma matrix is calculated from the fitting tensor as

$$[\mathbf{\Gamma}^{\tilde{d}}]_{\tilde{r}r} = \prod_{d \neq \tilde{d}} \langle f_{\tilde{r}}^d | f_r^d \rangle, \tag{21}$$

and the structure of the RHS matrix depends on the target-tensor format (see Sec. IV B 1). Using the full gradient tensor, $\mathcal{F}$ can be optimized using a variety of *CP non-linear conjugate gradient* (CP-NCG) algorithms.^{31,33} The performance of the CP-NCG algorithm depends strongly on the choice of the NCG method (see Ref. 58 for a comprehensive survey) and line-search algorithm.^{59–61} Generally, these methods require a number of gradient evaluations per iteration in order to determine the step length, which can become computationally costly. This can be avoided by using alternating algorithms which optimize one mode matrix of $\mathcal{F}$ at a time. The most widespread method is the *CP alternating least squares* (CP-ALS) method which sets the gradient block of mode $\tilde{d}$ equal to zero and solves the linear least-squares problem for $\mathbf{F}^{\tilde{d}T}$ (using the fact that $\mathbf{\Gamma}^{\tilde{d}}$ is symmetric),

$$\mathbf{\Gamma}^{\tilde{d}} \mathbf{F}^{\tilde{d}T} = \mathbf{Q}^{\tilde{d}T}, \tag{22}$$

to update the $\tilde{d}$ -th mode matrix of $\mathcal{F}$. This is done successively for each mode, and the whole process is repeated until convergence. Another alternating method is the *CP pivotised alternating steepest descent* (CP-PASD) method introduced in Ref. 40, which computes the full gradient and then performs a steepest-descent step with an exact line search on the mode with the largest gradient norm. We include the possibility of using a diagonal preconditioner with the steepest-descent step. This approach is equivalent to taking one (preconditioned) steepest-descent iteration in solving the linear system of Eq. (22), and it thereby avoids solving the full least-squares problem which scales cubically with the rank of the fitting tensor $R^{\mathcal{F}}$.

Our tensor-decomposition framework includes implementations of CP-NCG, CP-ALS, and CP-PASD as well as a pivotised version of CP-ALS. We have found that for CP-VCC calculations, the CP-ALS algorithm offers the best balance between speed and accuracy, and we therefore describe this implementation in more detail in Sec. IV B. In order for $\mathcal{F}$ to satisfy Eq. (18) while keeping the rank as low as possible, it is necessary to devise a scheme for increasing the rank of the fitting tensor if the error of the optimized tensor is too large. This can be done in several ways, and we describe our method of choice in Sec. IV C.

An alternative to rank reduction has been suggested in Ref. 62 in the context of solving eigenvalue equations where in iteration n a tensor $\sigma^{(n)}$ is projected on $\sigma^{(n-1)}$ in case the

overlap $\langle \sigma^{(n)} | \sigma^{(n-1)} \rangle$ is sufficiently large. This approach is not necessarily well-suited for recompressions during the VCC-error-vector calculation and will not be considered further in this work, but it may become relevant later in the context of VCC-response eigenvalue calculations.

A. Reducing the extents of the target tensor

The scaling of the CP-ALS algorithm depends linearly on the extents of the target tensor. We therefore wish to reduce the extents by removing linear dependency between the mode vectors of the target tensor $\mathcal{G}$ which also enhances the numerical stability of the recompression algorithm. Our approach is based on the C2T algorithm of Ref. 57, where the CP tensor is converted to a Tucker-like format,

$$g_{i_1 i_2 \dots i_{D\mathcal{G}}} = \sum_{v_1=1}^{\tilde{I}_1} \sum_{v_2=1}^{\tilde{I}_2} \dots \sum_{v_{D\mathcal{G}}=1}^{\tilde{I}_{D\mathcal{G}}} \tilde{g}_{v_1 v_2 \dots v_{D\mathcal{G}}} u_{i_1 v_1}^1 u_{i_2 v_2}^2 \dots u_{i_{D\mathcal{G}} v_{D\mathcal{G}}}^{D\mathcal{G}}, \tag{23}$$

where

$$\tilde{g}_{v_1 v_2 \dots v_{D\mathcal{G}}} = \sum_{r=1}^{R^{\mathcal{G}}} \tilde{g}_{v_1 r}^1 \tilde{g}_{v_2 r}^2 \dots \tilde{g}_{v_{D\mathcal{G}} r}^{D\mathcal{G}} \tag{24}$$

are elements of the *core tensor* $\tilde{\mathcal{G}}$ and $\mathbf{U}^d$ is the *side matrix* of mode d . If $\mathcal{G}$ is a direct product of CP tensors, the core tensor will also be given in direct-product format $\tilde{\mathcal{G}} = \bigotimes_j \tilde{\mathcal{G}}_j$, where the elements of the $\tilde{\mathcal{G}}_j$ tensors are given in CP format as in Eq. (24). The extents of the core tensor are smaller than (or equal to) the extents of $\mathcal{G}$, i.e., $\tilde{I}_d \leq I_d$. The Tucker-like format is obtained by performing truncated *singular value decompositions* (SVDs) on the mode matrices of $\mathcal{G}$ [known as *high-order SVD* (HOSVD)³⁰], i.e., $\mathbf{G}^d = \mathbf{U}^d \mathbf{\Sigma}^d \mathbf{V}^{dT}$. In contrast to Ref. 57 and previous work from our group,⁴⁵ we do not perform the *high-order orthogonal iterations* (HOOI) on top of the SVDs as this involves constructing the tensor in full dimension but obtain the mode matrices of the core tensor simply as

$$\tilde{\mathbf{G}}^d = \mathbf{U}^{dT} \mathbf{G}^d. \tag{25}$$

After reducing the rank of the core tensor by fitting it to a low-rank approximation $\tilde{\mathcal{F}}$ such that $\|\tilde{\mathcal{F}} - \tilde{\mathcal{G}}\| < T_{\text{CP}}$, we transform back to the original basis by

$$\mathbf{F}^d = \mathbf{U}^d \tilde{\mathbf{F}}^d. \tag{26}$$

B. The CP-ALS implementation

Our implementation of the CP-ALS algorithm is shown in Algorithm 1. The $\mathbf{\Gamma}^d$ and $\mathbf{Q}^d$ matrices of the least-squares system [Eq. (22)] as well as the difference norm $e = \|\mathcal{F} - \mathcal{G}\|$ can be calculated from different types of intermediates as described in Sec. IV B 1.

The stopping conditions for the optimization procedure are based on the absolute error change $\Delta e^{(i)} = |e^{(i-1)} - e^{(i)}|$ for the amplitude tensors and on the relative distance decrease $\Delta d^{(i)} = \frac{|d^{(i-1)} - d^{(i)}|}{d^{(i)}}$ with $d = \sqrt{1 + e^2}$ for recompression during the error-vector calculation. The distance-decrease check is able to stop the iterations early if the error of the fit is large, even if the absolute error change is above the threshold value. This can be seen from the fact that for large errors, Δd becomes

Algorithm 1: CP-ALS.

Input: Target tensor, $\mathcal{G}$
Fitting tensor, $\mathcal{F}$
Output: Optimized rank- $R^{\mathcal{F}}$ tensor, $\ast \mathcal{F}$

```

1 begin CP-ALS
2   • Initialize intermediates for the  $\Gamma^d$  and  $Q^d$  matrices
3   for  $i = 1$  to  $N_{\text{maxiter}}$  do
4     // Optimize the mode matrices of the fitting
      tensor
5     for  $\tilde{d} = 1$  to  $D^{\mathcal{G}}$  do
6       • Construct  $\Gamma^{\tilde{d}}$  and  $Q^{\tilde{d}T}$  from intermediates
7       • Regularize the  $\Gamma^{\tilde{d}}$  matrix
8       • Solve linear system:  $\Gamma^{\tilde{d}} X^{\tilde{d}T} = Q^{\tilde{d}T}$ 
9       • Update mode matrix:  $F^{\tilde{d}} \leftarrow X^{\tilde{d}}$ 
10      • Update intermediates of mode  $\tilde{d}$ 
11    • Calculate  $e^{(i)} = \|\mathcal{F} - \mathcal{G}\|$  from intermediates and  $\|\mathcal{G}\|^2$ 
12    // Check distance decrease or error change
13    If converged then
14      •  $\ast \mathcal{F} \leftarrow \mathcal{F}$ 
15      break

```

a relative measure while it is an absolute measure if the error norm is small,

$$\Delta d^{(i)} \approx \begin{cases} \frac{|e^{(i-1)} - e^{(i)}|}{e^{(i)}} & , e \gg 1 \\ \frac{1}{2} |e^{(i-1)2} - e^{(i)2}| & , e \ll 1 \end{cases}, \quad (27)$$

where the last identity comes from performing a 1st-order Taylor expansion. We use the distance-decrease check for recompressions during the error-vector calculation where speed is more important than the final ranks of the tensors. For the recompression of the amplitude tensors, we use the absolute error change in order to obtain as low ranks as possible at the cost of spending more iterations on each tensor fit. The maximum allowed number of iterations for each

optimization is typically set to $N_{\text{maxiter}}^e = 10$ for the error vectors and $N_{\text{maxiter}}^t = 100$ for the amplitudes.

We add a regularization term to the objective function which helps in balancing the norms of the individual rank-1 tensors in the CP representation,

$$J_N(\mathcal{F}) = J(\mathcal{F}) + \frac{\lambda^{\mathcal{F}}}{2} \sum_{r=1}^{R^{\mathcal{F}}} \prod_{d=1}^{D^{\mathcal{F}}} \|\mathbf{f}_r^d\|^2. \quad (28)$$

This helps us to avoid problems with *degeneracy*, where the best rank- R approximation does not exist.^{30,32,63} The regularization modifies the gamma matrix of the least-squares system as

$$[\Gamma^{\tilde{d}}]_{\tilde{r}\tilde{r}} = \left(\prod_{d \neq \tilde{d}} \langle \mathbf{f}_{\tilde{r}}^d | \mathbf{f}_{\tilde{r}}^d \rangle \right) (1 + \lambda^{\mathcal{F}} \delta_{\tilde{r}\tilde{r}}). \quad (29)$$

The regularization parameter $\lambda^{\mathcal{F}}$ is chosen such that the regularization term is small compared to the requested accuracy $\sqrt{2(J_N(\mathcal{F}) - J(\mathcal{F}))} < T_{\text{CP}}$ as discussed in Ref. 32. Furthermore, in order to enhance numerical stability, we balance the norms of the mode vectors after recompressing the amplitude tensors such that $\|\mathbf{f}_r^d\| = \|\mathbf{f}_r^{d'}\|$ for all r .

1. Intermediates for the recompression algorithm

Together with solving the least-squares system of Eq. (22), the most expensive operations in the CP-ALS algorithm is the construction of the gamma and right-hand-side matrices. Since the CP-ALS algorithm only updates one mode matrix at a time, we are able to reuse the intermediates shown in Table V between iterations. Both the gamma and right-hand-side intermediates are essentially dot products between mode vectors of the $\mathcal{F}$ and $\mathcal{G}$ tensors in the cases where the target tensor is a CP tensor or a direct product of CP tensors. If, on the other hand, $\mathcal{G}$ was a full tensor, it would not be possible to identify right-hand side intermediates, and the evaluation of the Q^d matrices would be much more expensive.

The Γ intermediates $\Gamma_{\tilde{r}\tilde{r}}^d = [\Gamma \mathbf{I}^d]_{\tilde{r}\tilde{r}} = \langle \mathbf{f}_{\tilde{r}}^d | \mathbf{f}_{\tilde{r}}^d \rangle$ are easily identified from the simple structure of the gamma matrix,

TABLE V. Γ and Q intermediates for recompression of a target tensor in either the CP or the direct-product format.

Type	Target format	Intermediate	Description
Q	CP	$Q_{\tilde{r}\tilde{r}}^d = [Q \mathbf{I}^d]_{\tilde{r}\tilde{r}} = \langle \mathbf{f}_{\tilde{r}}^d \mathbf{g}_{\tilde{r}}^d \rangle$	Dot products between mode vectors. $Q \mathbf{I}^d = \mathbf{F}^{dT} \mathbf{G}^d$.
	Direct product	$1 \mathbf{t}_{\tilde{r}\tilde{r}j}^j = \begin{bmatrix} \mathbf{t}_{\tilde{r}j}^{\mathcal{G}} \\ \mathbf{t}_{\tilde{r}j}^{\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{\tilde{r}}^{\mathcal{G}} \\ \mathbf{t}_{\tilde{r}}^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{t}_{j,r_j}^{\mathcal{G}} \\ \mathbf{t}_{j,r_j}^{\mathcal{F}} \end{bmatrix}$ $2 \mathbf{t}_{\tilde{r}}^j = [2 \mathbf{t}^j]_{\tilde{r}} = \sum_{r_j=1}^{R^{\mathcal{G}}} \prod_{d_j=1}^{D^{\mathcal{G}}} \left(\mathbf{f}_{\tilde{r}}^{\mathcal{G}} \middle \mathbf{t}_{\tilde{r}}^{\mathcal{G}} \right) \left(\mathbf{g}_{j,r_j}^{\mathcal{G}} \middle \mathbf{t}_{j,r_j}^{\mathcal{G}} \right)$ $= \sum_{r_j=1}^{R^{\mathcal{G}}} \prod_{d_j=1}^{D^{\mathcal{G}}} 1 \mathbf{t}_{\tilde{r}r_j}^j$	Level-1 intermediates. Dot products between mode vectors. Level-2 intermediates. Reused between updating modes coming from different tensors in the direct product.
Γ	All	$\Gamma_{\tilde{r}\tilde{r}}^d = [\Gamma \mathbf{I}^d]_{\tilde{r}\tilde{r}} = \langle \mathbf{f}_{\tilde{r}}^d \mathbf{f}_{\tilde{r}}^d \rangle$	Dot products between mode vectors. $\Gamma \mathbf{I}^d = \mathbf{F}^{dT} \mathbf{F}^d$.

$$[\Gamma^{\tilde{d}}]_{\tilde{r}\tilde{r}} = \prod_{d \neq \tilde{d}} \langle f_{\tilde{r}}^d | f_r^d \rangle = \prod_{d \neq \tilde{d}} \Gamma \iota_{\tilde{r}r}^d \quad (30)$$

which allows us to only re-calculate $\Gamma \mathbf{I}^d = \mathbf{F}^{dT} \mathbf{F}^d$ after updating the d -th mode matrix of $\mathcal{F}$. The Q intermediates used for calculating the $\mathbf{Q}^d$ matrix depend on the format of the target tensor $\mathcal{G}$. If the target tensor is given in CP format, the matrix elements of $\mathbf{Q}^d$ are given as

$$[\mathbf{Q}^d]_{\tilde{r}\tilde{r}} = \left\langle \frac{\partial \mathcal{F}}{\partial f_{\tilde{r}}^d} \middle| \mathcal{G} \right\rangle = \sum_{r=1}^{R^G} \prod_{d \neq \tilde{d}} \langle f_{\tilde{r}}^d | \mathbf{g}_r^d \rangle g_{\tilde{r}r}^d = \sum_{r=1}^{R^G} \prod_{d \neq \tilde{d}} Q \iota_{\tilde{r}r}^d g_{\tilde{r}r}^d, \quad (31)$$

where we can store the Q intermediates $\mathbf{Q} \mathbf{I}^d = \mathbf{F}^{dT} \mathbf{G}^d$ between iterations.

For recompression of direct products, the structure of $\mathcal{G} = \bigotimes_j \mathcal{G}_j$ allows for identifying two levels of intermediates. We define index sets for the $\mathcal{G}_j$ tensors as $\mathbf{i}^{\mathcal{G}_j} = \{i_1^{\mathcal{G}_j}, i_2^{\mathcal{G}_j}, \dots, i_{D^{\mathcal{G}_j}}^{\mathcal{G}_j}\}$ where the union of all index sets corresponds to the indices of the $\mathcal{G}$ tensor, $\mathbf{i}^{\mathcal{G}} = \cup_j \mathbf{i}^{\mathcal{G}_j} = \{1, 2, \dots, D^{\mathcal{G}}\}$. Thus, a direct product of CP tensors has the structure

$$\mathcal{G} = \bigotimes_j \sum_{r_j=1}^{R^{\mathcal{G}_j}} \mathbf{g}_{j,r_j}^{i_1^{\mathcal{G}_j}} \otimes \mathbf{g}_{j,r_j}^{i_2^{\mathcal{G}_j}} \otimes \dots \otimes \mathbf{g}_{j,r_j}^{i_{D^{\mathcal{G}_j}}^{\mathcal{G}_j}}. \quad (32)$$

We furthermore define $j^{\tilde{d}}$ as the index of the tensor $\mathcal{G}_{j^{\tilde{d}}}$ which includes the $\tilde{d}$ -th mode matrix of the $\mathcal{G}$ tensor, i.e., $j^{\tilde{d}} = j | \tilde{d} \in \mathbf{i}^{\mathcal{G}_j}$. For a general direct product, the RHS matrix is defined as

$$\left\langle \frac{\partial \mathcal{F}}{\partial f_{\tilde{r}}^{\tilde{d}}} \middle| \bigotimes_j \mathcal{G}_j \right\rangle = \left[\prod_{j \neq j^{\tilde{d}}} \left\langle f_{\tilde{r}}^{i_1^{\mathcal{G}_j}} \otimes \dots \otimes f_{\tilde{r}}^{i_{D^{\mathcal{G}_j}}^{\mathcal{G}_j}} \middle| \mathcal{G}_j \right\rangle \right] \times \left[\left\langle f_{\tilde{r}}^{i_1^{j^{\tilde{d}}}} \otimes \dots \otimes \mathbf{1}_i^{\tilde{d}} \otimes \dots \otimes f_{\tilde{r}}^{i_{D^{j^{\tilde{d}}}}^{\mathcal{G}_{j^{\tilde{d}}}}} \middle| \mathcal{G}_{j^{\tilde{d}}} \right\rangle \right], \quad (33)$$

where $\mathbf{1}_i^{\tilde{d}}$ denotes a unit vector in mode $\tilde{d}$ with the $\tilde{r}$ th element equal to 1. We see that the first part of Eq. (33) is constant when updating modes included in the tensor $\mathcal{G}_{j^{\tilde{d}}}$ which will enable us to identify intermediates for specific types of $\mathcal{G}_j$. Furthermore, in the CP-VCC algorithm, all direct products contain only CP tensors which in the end allow us to identify two types of Q intermediates, level-1 and level-2. These are identified by explicitly inserting Eq. (32) into Eq. (33),

$$\begin{aligned} \left\langle \frac{\partial \mathcal{F}}{\partial f_{\tilde{r}}^{\tilde{d}}} \middle| \bigotimes_j \mathcal{G}_j \right\rangle &= \left[\prod_{j \neq j^{\tilde{d}}} \left(\sum_{r_j=1}^{R^{\mathcal{G}_j}} \prod_{d_j=1}^{D^{\mathcal{G}_j}} \left\langle f_{\tilde{r}}^{i_{d_j}^{\mathcal{G}_j}} \middle| \mathbf{g}_{j,r_j}^{i_{d_j}^{\mathcal{G}_j}} \right\rangle \right) \right] \\ &\times \left[\sum_{r_{j^{\tilde{d}}}=1}^{R^{\mathcal{G}_{j^{\tilde{d}}}}} \prod_{d_{j^{\tilde{d}}} \neq \tilde{d}} \left\langle f_{\tilde{r}}^{i_{d_{j^{\tilde{d}}}}^{\mathcal{G}_{j^{\tilde{d}}}}} \middle| \mathbf{g}_{j^{\tilde{d}},r_{j^{\tilde{d}}}}^{i_{d_{j^{\tilde{d}}}}^{\mathcal{G}_{j^{\tilde{d}}}}} \right\rangle g_{j^{\tilde{d}},\tilde{r}}^{\tilde{d}} \right] \\ &= \left[\prod_{j \neq j^{\tilde{d}}} 2 \iota_{\tilde{r}}^j \right] \left[\sum_{r_{j^{\tilde{d}}}=1}^{R^{\mathcal{G}_{j^{\tilde{d}}}}} \prod_{d_{j^{\tilde{d}}} \neq \tilde{d}} 1 \iota_{\tilde{r}r_{j^{\tilde{d}}}}^{d_{j^{\tilde{d}}}} g_{j^{\tilde{d}},\tilde{r}}^{\tilde{d}} \right]. \quad (34) \end{aligned}$$

The level-2 intermediates are reused between updating modes coming from different tensors in the direct product, whereas the level-1 intermediates can also be used when updating modes in the same tensor. Note that the level-2 intermediates can be calculated from the level-1 intermediates.

The Γ and Q intermediates are also used when calculating the difference norms between the target and fitting tensor,

$$\|\mathcal{F} - \mathcal{G}\| = \sqrt{\|\mathcal{F}\|^2 + \|\mathcal{G}\|^2 - 2\langle \mathcal{F} | \mathcal{G} \rangle}, \quad (35)$$

which are needed for checking convergence. The norm of the fitting tensor can be obtained from the Γ intermediates $\Gamma \mathbf{I}^d$, while the overlap $\langle \mathcal{F} | \mathcal{G} \rangle$ is calculated easily from the Q intermediates. The norm of the target tensor $\|\mathcal{G}\|^2$ is calculated once at the beginning of the recompression.

C. Obtaining accurate low-rank approximations

Devising an efficient algorithm for solving Eq. (18) with the lowest possible rank is complicated and involves several rank- R fittings. We use what we call the FindBestCP algorithm as shown in Algorithm 2 which simply fits the target tensor to CP tensors of higher and higher ranks until a good enough fit is obtained. The bottleneck of this algorithm is the fitting of the last tensor which scales cubically with the final rank. We choose the upper limit of the fitting rank as

$$R_{\text{bound}} = \frac{\prod_{d=1}^D I_d}{\max_d I_d}, \quad (36)$$

which for a 3rd-order tensor is the theoretical maximum rank.³⁰

1. Choosing the rank increment

The performance of the FindBestCP algorithm depends on the magnitude of the rank increment ΔR . A small value of ΔR results in lower ranks but can require a large number of

Algorithm 2: FindBestCP recompression algorithm.

Input: Target tensor, $\mathcal{G}$
Threshold, T_{CP}
Maximum allowed rank, R_{max}
Starting rank, R_{start}
Rank increment, ΔR
Guesser and guess updater types

Output: Optimized fitting tensor, $\mathcal{F}$
Final rank, $R^{\mathcal{F}}$
Absolute error, $\|\mathcal{F} - \mathcal{G}\|$

```

1 begin (Decompose)
2   • Reduce extents of the target using C2T (Sec. IV A):  $\tilde{\mathcal{G}}^d = \mathbf{U}^{dT} \mathcal{G}^d$ 
3   • Set  $\tilde{\mathcal{F}}$  to a starting guess of rank  $R_{\text{start}}$ 
4   • Calculate maximum theoretical rank  $R_{\text{bound}}$  using Eq. (36)
5   • Set  $r_{\text{max}} = \min\{R_{\text{max}}, R_{\text{bound}}\}$ 
6   for  $r = R_{\text{start}}$  to  $r_{\text{max}}$  do
7     if  $R^{\tilde{\mathcal{F}}} < r$  then
8       Update  $\tilde{\mathcal{F}}$  to rank  $r$ 
9       • Run CP-ALS fitting of  $\tilde{\mathcal{F}}$ 
10      if  $\|\tilde{\mathcal{F}} - \tilde{\mathcal{G}}\| < T_{\text{CP}}$  then
11        break
12      •  $r \leftarrow r + \Delta R$ 
13   • Transform the fitting tensor back to the original basis:  $\mathbf{F}^d = \mathbf{U}^d \tilde{\mathbf{F}}^d$ 

```

CP fittings which becomes expensive. On the other hand, a large value of ΔR results in higher ranks. We have therefore devised a black-box dynamic scheme for determining ΔR for a higher-order tensor ($D > 2$) based on the order and extents of the target tensor (or the core tensor when the C2T algorithm is used),

$$\Delta R = \min_d I_d \times (D - 2). \quad (37)$$

This choice of ΔR becomes small when one of the extents of the tensor is small which is often the case when using the C2T algorithm. The rank increments also depend linearly on the order of the tensor to avoid too many fittings of high-order tensors.

We have also tried other ways of setting the rank increment and used either $\Delta R = \Delta R_{in}$ or $\Delta R = \Delta R_{in}(D^G - 2)$ where ΔR_{in} is an input parameter. These approaches are, however, less black-box in nature. Their performance compared to the dynamic scheme of Eq. (37) is examined in Sec. V D.

As default, we use the dynamic scheme for recompressions during the error-vector calculation, but for recompression of the VCC amplitudes, we simply set $\Delta R = 1$ to obtain the lowest possible ranks.

2. Choosing an accurate starting guess

The performance of the CP-ALS algorithm depends heavily on the starting guess, and we have therefore devised different strategies for generating accurate guesses. One popular way of initializing the rank- R guess is by performing truncated SVDs on matrix unfoldings of the full tensor and using the R leading left-singular vectors as the mode matrices of $\mathcal{F}$.³⁰ This approach, however, requires the construction of the matrix unfoldings in full dimension and is therefore not well-suited for recompression. Other approaches such as the *successive cross approximation* (SCA) have been suggested for recompression.^{32,64} In our experience, however, initializing the guess with random elements and scaling by the norm of the target tensor results in the best performance of the recompressions and the overall CP-VCC algorithm.

When using the FindBestCP method presented in Algorithm 2, we also need a scheme for obtaining a rank- $(R + \Delta R)$ guess if the rank- R fit is not accurate enough. Because the best rank- R approximation is not necessarily included in the best rank- $(R + 1)$ approximation,³⁰ a reasonable solution would be to simply replace $\mathcal{F}$ with a fresh starting guess at each rank. However, we observe much better performance of the FindBestCP algorithm (in terms of computational cost, final ranks, and accuracy) if we reuse the information from the previous fittings. This can be done by adding ΔR random vectors to each mode matrix of $\mathcal{F}$,⁶⁵ but the best performance is obtained when we fit a rank- ΔR tensor $\Delta\mathcal{F}$ to the residual $\mathcal{R} = \mathcal{G} - \mathcal{F}$ and add it to the fitting tensor $\mathcal{F} \leftarrow \mathcal{F} + \Delta\mathcal{F}$ as suggested in Ref. 32.

V. RESULTS

A. Computational details

The CP-VCC method has been implemented in the MidasCpp program package⁶⁶ which contains implementations of all the vibrational-structure methods described in

Sec. II as well as tools for automatic PES generation and the tensor-decomposer framework described in Sec. IV and in the [supplementary material](#).

The test calculations have been performed on a set of molecules: thiadiazole (15 modes) and an array of polycyclic aromatic hydrocarbons (PAHs) including naphthalene (48 modes) and anthracene (66 modes) using the PESs of Refs. 24 and 6, respectively. The thiadiazole PES contains up to 3-mode couplings, while the PAH PESs contain up to 2-mode couplings.

All calculations are converged based on the maximum norm $e_{\max} = \max_{\mathbf{m}} ||e^{\mathbf{m}}||$ to $T_{\text{VCC}} = 10^{-6}$ and on the energy change to a threshold of $T_{\Delta E} = 10^{-8}$ a.u. using the CROP(3) algorithm presented in our previous work.²⁶ An overview of the internal thresholds of the CP-VCC algorithm together with our default settings is given in Table VI. Most thresholds are in our default setup related directly to T_{VCC} in order to minimize the number of user-defined parameters. The thresholds for screening and maximum accuracy of amplitudes and error vectors ($T_{\text{VCC}}^{\text{screen}}$, $T_{\text{CP},t}^{\text{min}}$, and $T_{\text{CP},e}^{\text{min}}$) are chosen to be smaller than T_{VCC} in such a way that they do not affect the results within the expected accuracy for the chosen convergence threshold. In essence, only the convergence threshold T_{VCC} needs to be set and the rest is according to defaults in a typical calculation, and we will in the results show how T_{VCC} determines the accuracy of the results for both CP-VCC and FT-VCC calculations.

B. Accuracy of the CP-VCC solutions

As a preliminary study, we wish to compare the solutions obtained from the CP-VCC algorithm using the threshold settings of Table VI to the FT-VCC solution vectors in order to show that the same states are found. Figure 6 shows the accuracy of CP-VCC[3] and FT-VCC[3] solution vectors and energies compared to a reference calculation, which is a tightly converged FT-VCC state (using $T_{\text{VCC}} = 10^{-12}$ and $T_{\Delta E} = 10^{-14}$). The results show that both the CP-VCC and FT-VCC solution vectors and energies converge towards the reference as T_{VCC} is tightened. It is seen that the average errors in the solution vectors obtained from the CP-VCC and FT-VCC algorithms are quite similar and in all cases smaller than T_{VCC} . The errors are a little smaller for FT-VCC because of the accumulated error of the CP representation over all MCs. When looking at the maximum error, the differences between FT-VCC and CP-VCC are a little less pronounced, and in one case, the CP-VCC state actually has a smaller maximum error than the FT-VCC state. Note also that the energy errors are in all cases smaller than the value of $T_{\Delta E}$. We thus conclude that the numerical paths to solution are different for the FT-VCC and CP-VCC algorithms, but the results obtained are similar within reasonable variations due to the different numerical treatments, and in both cases, the error is controlled by the given threshold.

C. Conditions for recompression

We now turn to examine the effect of the maximum allowed rank $R_{\text{VCC}}^{\max}$. Figure 7 shows the relative time spent on solving the VCC[4] equations for thiadiazole with

TABLE VI. Numerical thresholds in the CP-VCC algorithm. The relation between the thresholds is shown for our default setup.

Threshold	Definition	Input parameter	Default input setting	Description
CP-VCC algorithm				
T_{VCC}	...	T_{VCC}	10^{-6}	Error-vector threshold for the non-linear-equation solver.
$T_{\Delta E}$	$T_{\Delta E} = C_{\Delta E} \times T_{\text{VCC}}$	$C_{\Delta E}$	10^{-2}	Energy-change threshold for the non-linear-equation solver.
$T_{\text{VCC}}^{\text{screen}}$	$T_{\text{VCC}}^{\text{screen}} = C_{\text{VCC}}^{\text{screen}} \times T_{\text{VCC}}$	$C_{\text{VCC}}^{\text{screen}}$	10^{-4}	Screening threshold.
FindBestCP algorithm				
$T_{\text{CP},t}^{(n) \text{ a}}$	$T_{\text{CP},t}^{(n)} = C_t \ \Delta \mathbf{t}^{(n)}\ $	C_t	5×10^{-3}	Accuracy of the VCC amplitudes in iteration n .
$T_{\text{CP},e}^{\mathbf{m},(n) \text{ a}}$	$T_{\text{CP},e}^{\mathbf{m},(n)} = T_{\text{CP},t}^{(n)} \frac{\ \mathbf{e}^{\mathbf{m},(n-1)}\ }{\ [\mathbf{A}_0^{-1}]^{\mathbf{m}} \mathbf{e}^{\mathbf{m},(n-1)}\ }$	...	...	Accuracy of recompressions during the calculation of $\mathbf{e}^{\mathbf{m}}$ in iteration n .
$T_{\text{CP},t}^{\min}$	$T_{\text{CP},t}^{\min} = C_{\text{CP},t}^{\min} \times T_{\text{VCC}}$	$C_{\text{CP},t}^{\min}$	10^{-1}	Highest allowed accuracy of amplitude tensors (see Ref. 26).
$T_{\text{CP},e}^{\min}$	$T_{\text{CP},e}^{\min} = C_{\text{CP},e}^{\min} \times T_{\text{VCC}}$	$C_{\text{CP},e}^{\min}$	10^{-3}	Highest allowed accuracy of error vectors (see Ref. 26).
CP-ALS algorithm				
$T_{\text{CP-ALS}}$	$T_{\text{CP-ALS}} = C_{\text{CP}}^{\text{rel}} \times T_{\text{CP}}$	$C_{\text{CP}}^{\text{rel}}$	10^{-3} (error change) 10^{-6} (distance decrease)	Convergence threshold for CP-ALS algorithm (see Sec. IV B and the supplementary material).

^aThe accuracies are adapted to the step size in the non-linear-equation solver as described in Ref. 26. $[\mathbf{A}_0^{-1}]^{\mathbf{m}}$ is the block of the inverse 0th-order VCC Jacobian corresponding to MC $\mathbf{m}$.

$N^m = 8$ using the CP-VCC algorithm with different choices of $R_{\text{VCC}}^{\max}$. The results show that choosing a small maximum rank results in a large computational overhead arising from the sheer

number of recompressions. Large values of $R_{\text{VCC}}^{\max}$ also result in a small computational overhead, but more importantly, the memory consumption increases significantly as we increase

FIG. 6. Accuracies of solution vectors and energies of CP-VCC and FT-VCC calculations for different convergence thresholds T_{VCC} . (a) Average error in the solution vector, (b) maximum error in the solution vector, and (c) energy errors.

FIG. 7. The time to solution (relative to the fastest time) of CP-VCC[4] calculations on thiadiazole with $N^m = 8$ using different values of $R_{\text{VCC}}^{\text{max}}$. The deviation from the FT-VCC energy (in cm^{-1}) is also shown.

$R_{\text{VCC}}^{\text{max}}$. Choosing $R_{\text{VCC}}^{\text{max}} \in [750, 2000]$ is thus the optimal choice, and we choose $R_{\text{VCC}}^{\text{max}} = 1000$ as our default setting in order to save memory compared to using a higher maximum rank. As expected, the errors in the final VCC energies are not at all affected by the choice of maximum rank. The fluctuations seen in Fig. 7 are of the order of 10^{-5} cm^{-1} , and the deviation from the FT-VCC energy is for all $R_{\text{VCC}}^{\text{max}}$ smaller than the convergence threshold $T_{\Delta E} = 10^{-8} \text{ a.u.} = 2.2 \times 10^{-3} \text{ cm}^{-1}$.

D. FindBestCP rank increments

We now wish to examine the performance of the three schemes for choosing the rank increment of the FindBestCP algorithm described in Sec. IV C 1. Figure 8(a) shows the time of CP-VCC[4] calculations on thiadiazole with $N^m = 8$ for different choices of the input rank increment ΔR_{in} , while Fig. 8(b) shows the average rank of the converged 4th-order error vector. The timings are normalized with respect to the time of the calculation using the dynamic scheme of Eq. (37). The results show that the dynamic scheme outperforms the two others with respect to speed regardless of the choice of ΔR_{in} . The effect on the computation time is in many cases around $\sim 10\%$. Furthermore, the dynamic scheme also makes the algorithm more black-box since it does not require the user to specify the input rank increment. It is seen from Fig. 8(b) that the final average ranks are lowest for low values of ΔR_{in} as would be expected.

TABLE VII. Ground-state energies in cm^{-1} for CP-VCC calculations on thiadiazole using different excitation orders and modals per mode.

	$N^m = 6$	$N^m = 8$	$N^m = 10$	$N^m = 12$	$N^m = 14$
VCC[2]	9333.88	9333.87	9333.87	9333.87	9333.87
VCC[2pt3]	9323.12	9323.10	9323.10	9323.10	9323.10
VCC[3]	9322.95	9322.93	9322.93	9322.93	9322.93
VCC[3pt4]	9322.70	9322.68	9322.67	9322.67	9322.67
VCC[4]	9322.69	9322.67	9322.67	9322.67	9322.67
VCC[5]	9322.66	9322.64	9322.63	9322.63	9322.63
VCC[6]	9322.66	9322.63	9322.63	9322.63	9322.63

E. Benchmark of high-order VCC models

The CP-VCC algorithm allows for performing high-order VCC calculations that have not been available before using the FT-VCC algorithm. It is therefore of interest to examine the convergence of the zero-point vibrational energy of thiadiazole with respect to excitation order and modal-basis size. The energies of VCC[n] calculations using the CP-VCC algorithm with $n \in [2, 6]$ as well as the perturbative VCC[2pt3] and VCC[3pt4] models are shown in Table VII. All results have been compared to FT-VCC calculations of the same level of theory (except for the VCC[5] and VCC[6] calculations which could not be performed with full tensors), and the maximum energy difference between the results was seen to be $\Delta E_{\text{max}} = 1.9 \times 10^{-3} \text{ cm}^{-1}$ which is well below the expected accuracy for the convergence thresholds of the non-linear-equation solver. The results show that the energy converges quite fast with respect to the number of one-mode basis functions. The energies are converged below 10^{-2} cm^{-1} with respect to modal-basis size for $N^m = 8$, except for VCC[3pt4] and VCC[5] where basis-set convergence is reached at $N^m = 10$. The largest effects on the energy are seen when introducing 3-mode excitations in the wave function, i.e., going from VCC[2] to VCC[2pt3] and VCC[3]. Introducing 4-mode excitations lowers the energy by $\sim 0.3 \text{ cm}^{-1}$, while the 5-mode and 6-mode excitations only have minor effects. Considering that the errors in the PES may be of the order of $\sim 1 \text{ cm}^{-1}$, VCC[3pt4] or VCC[4] with $N^m = 8$ seems to be sufficient for most applications. However, it is important to note that these results are only for the ground-state energies. When calculating excitation energies using VCC response theory, it can be necessary to go to higher-order VCC models in order to get converged results.

FIG. 8. Comparison of the three schemes for choosing the rank increment ΔR described in Sec. IV C 1 for CP-VCC[4] calculations on thiadiazole with $N^m = 8$. (a) Time relative to the dynamic scheme of Eq. (37). (b) Average ranks of 4th-order error vectors.

FIG. 9. Number of parameters in the converged solution vectors of a set of CP-VCC[3] and CP-VCC[6] calculations on thiadiazole. The formal number of parameters in the corresponding FT-VCC solutions is also shown.

FIG. 10. Number of parameters in the converged solution vectors of a set of CP-VCC[3] and CP-VCC[4] calculations on PAHs with $N^m = 8$. The formal number of parameters in the corresponding FT-VCC solutions is also shown.

F. Compression of the VCC wave function

Figure 9 shows the number of stored elements in the converged VCC[3] and VCC[6] solution vectors from CP-VCC calculations on thiadiazole using different numbers of one-mode basis functions. These numbers are compared to the formal number of parameters in the calculation, and it is evident that the CP format allows for substantial storage savings. This is especially the case for high-order VCC models such as VCC[6] where we have been able to perform calculations with more than 10^{11} formal parameters. Such calculations are completely inaccessible using the FT-VCC algorithm due to the memory requirements of the wave-function parameters and

intermediates as well as the computational cost of the VCC contractions.

Figure 10 shows the number of parameters in the converged CP-VCC[3] and CP-VCC[4] solution vectors for different PAHs. Also here, order-of-magnitude reductions in the parameter spaces are observed. It is seen that the M scaling of the number of parameters in the wave function is reduced slightly by using the CP-VCC algorithm compared to FT-VCC. However, it is important to note that the calculations have been performed in normal coordinates which are delocalized over the entire molecule. Performing CP-VCC calculations using the localized FALCON⁶⁷ or HOLC⁶⁸ coordinates is an interesting future perspective for

FIG. 11. Rank distributions of all three-mode couplings in the converged VCC[3] wave functions with $N^m = 8$ for benzene, naphthalene, anthracene, and tetracene.

FIG. 12. Convergence of the data-compression factors [Eq. (38)] during a CP-VCC[6] calculation on thiadiazole. The maximum mode-combination error $e_{\max} = \max_{\mathbf{m}} ||\mathbf{e}^{\mathbf{m}}||$ is also shown. Note that the points with compression factors equal to zero (all ranks are zero) are not shown due to the logarithmic scale.

FIG. 13. Convergence of the data-compression factors [Eq. (38)] during a CP-VCC[4] calculation on anthracene. The maximum mode-combination error $e_{\max} = \max_{\mathbf{m}} ||\mathbf{e}^{\mathbf{m}}||$ is also shown. Note that the points with compression factors equal to zero (all ranks are zero) are not shown due to the logarithmic scale.

studying the ranks of interactions between distant vibrational modes.

In Sec. III A, it was argued that the relative number of weak mode couplings increases with the size of the molecule. Figure 11 shows the distribution of ranks of all three-mode couplings in the converged VCC[3] wave functions of benzene, naphthalene, anthracene, and tetracene. The number of three-mode couplings is much larger than the number of lower-order couplings for all four molecules, and thereby the rank distribution is determined almost fully by the ranks of the 3rd-order amplitude tensors. It is clearly seen that the number of unimportant mode couplings increases significantly for the larger molecules. The observed evolution of ranks with system size is particularly noteworthy given that the systems considered are described in terms of very delocalised normal coordinates. There is a general shift towards higher ratios of lower-rank tensors and reduced ratios of higher-rank tensors. The number of tensors with $R > 5$ exhibits a remarkably flat scaling with respect to system size and actually decreases when going from anthracene to tetracene. For tetracene, the largest fraction of three-mode tensors (27%) is actually of (approximate) rank 0 when fitted to an absolute threshold, meaning that those couplings can be neglected completely.

Another important question is whether the ranks of the CP-VCC amplitudes converge to a maximum value during the course of iteration of the non-linear-equation solver and how much the tensors of the different excitation orders are compressed. Figures 12 and 13 show the compression factors of the cluster amplitudes for a given excitation level $t_n = \{\mathbf{t}^{\mathbf{m}_n}\} = \{\mathbf{t}^{\mathbf{m}} | \dim(\mathbf{m}) = n\}$ (singles, doubles, triples, etc.),

$$\text{comp}(t_n) = \frac{\sum_{\mathbf{m}_n} R^{\mathbf{m}_n} \sum_{d=1}^{D^{\mathbf{m}_n}} I_d}{\sum_{\mathbf{m}_n} \prod_{d=1}^{D^{\mathbf{m}_n}} I_d} = \frac{\text{CP } N_{\text{params}}^{t_n}}{\text{full } N_{\text{params}}^{t_n}}, \quad (38)$$

together with the maximum MC error $e_{\max} = \max_{\mathbf{m}} ||\mathbf{e}^{\mathbf{m}}||$ for each iteration in a CP-VCC[6] calculation on thiadiazole and a CP-VCC[4] calculation on anthracene, respectively. It is clearly seen that the compression factors converge during the iterations and that the higher-order amplitudes are compressed more than the low-order excitations. The only exception is the doubles amplitudes where the CP decomposition is equivalent to a singular-value decomposition (SVD) of a matrix which only results in data compression if $R < N_{\text{vir}}^m/2$. It is also important to note that we observe a higher rate of compression of the 3- and 4-mode amplitudes for anthracene than for thiadiazole. This can be explained from an increased number of weakly interacting modes.

G. Computational scaling of the CP-VCC models

Figures 14 and 15 show the N scaling of VCC[3] and VCC[4] calculations on thiadiazole and naphthalene using the FT-VCC and CP-VCC algorithms. The average time per iteration t_{it} is plotted against the number of virtual modals (the extents of the tensors) on two logarithmic axes in order to see the polynomial scaling of the FT-VCC methods. It is clear that the computational scaling is reduced significantly for both molecules when using CP-VCC. As expected from the theoretical scalings of the contractions (Table II), the benefits of using the CP-VCC method are much more clear for VCC[4] than for VCC[3] since the VCC[4] contractions are more costly. When 4-mode couplings are introduced, the CP-VCC

FIG. 14. Time per iteration for a series of FT-VCC and CP-VCC calculations on thiadiazole with different numbers of basis functions. The iteration times are the total time-to-solution divided by the number of iterations and normalized with respect to the smallest FT-VCC calculation.

FIG. 15. Time per iteration of a series of FT-VCC and CP-VCC calculations on naphthalene with different numbers of basis functions. The iteration times are the total time-to-solution divided by the number of iterations and normalized with respect to the smallest FT-VCC calculation.

algorithm outperforms the FT-VCC algorithm quite significantly. This is even the case for quite small basis sets $N^m = 8$ ($N_{vir}^m = 7$) where the largest tensors are only of dimensions $7 \times 7 \times 7 \times 7$. For both thiadiazole and naphthalene, it was not possible to perform the calculations with the largest basis sets using the FT-VCC[4] algorithm (on a node with two 14-core, 2.4 GHz Intel central processing units (CPUs) and 256 GB of memory). This showcases the significant benefits of using the CP-VCC algorithm for performing high-order VCC calculations on larger molecules.

VI. SUMMARY AND OUTLOOK

We have described the new implementation of VCC theory introduced in Ref. 26 where all tensors are kept in CP format during the entire calculation in order to harvest the computational benefits of the decomposed format. The similarities and differences between the CP-VCC and FT-VCC algorithms have been presented, and the computational scalings of the VCC tensor contractions and direct products as well as the memory requirements of the two methods have been discussed. The rank-reduction algorithms for CP tensors and direct products of CP tensors which are crucial to the performance of the CP-VCC algorithm have been described in some detail focusing on ways to optimize speed and numerical stability.

Our benchmark calculations show that the number and ratio of low-rank and rank-0 tensors in the VCC wave function increase significantly for larger molecules which is important in order to lower the M scaling of VCC calculations. Furthermore, the average ranks of the VCC amplitude tensors converge during the course of iteration of the non-linear-equation solver, and tensors describing high-order excitations are compressed more than the low-order tensors as expected. The computational scaling and memory requirements are reduced significantly in the CP-VCC method which enables us to perform high-order VCC calculations on larger molecules and thereby to see the convergence of the VCC energy with respect to excitation order and basis size for thiadiazole. This benchmark shows that the energy errors of the VCC[4] and VCC[3pt4] models are only $\sim 5 \times 10^{-2} \text{ cm}^{-1}$ compared to the VCC[6] results which is much less than the expected error of the PES. Using the CP-VCC algorithm, we have been able to calculate vibrational zero-point energies at the VCC[4] level of theory for systems with up to 66 vibrational modes (anthracene). A CP-VCC[3pt4] calculation on tetracene (84 modes) with $N^m = 6$ has also been performed.

A relevant future investigation is the application of the CP-VCC algorithm with FALCON or HOLC coordinates, where the locality of the vibrational modes may give rise to lower ranks of interactions between far-distant modes and thereby a further reduction of the M scaling. Furthermore, this investigation can be combined with the MC-screening measures described in Ref. 69.

Another important perspective is the extension of the present implementation to VCC response theory in order to calculate excitation energies as well as other vibrational properties. The current CP-VCC implementation is flexible enough to allow for VCC-Jacobian transformations which is the main ingredient in performing VCC response calculations. The next step will be to further develop the tensor-decomposed eigenvalue solver of our earlier work to include automatic recompression of the trial vectors as well as devising an efficient scheme for obtaining the diagonal Davidson preconditioner in CP format. These future developments may allow for calculating highly accurate vibrational spectra and properties of larger molecules with more than 20 atoms such as PAHs with 3-4 rings.

SUPPLEMENTARY MATERIAL

See [supplementary material](#) for a thorough description of the MidasCpp tensor-decomposition framework.

ACKNOWLEDGMENTS

The authors would like to thank Mads Bøttger Hansen and Gunnar Schmitz for thorough reading of this paper as well as for their valuable comments. We are grateful to Mike Espig for discussions regarding the recompression algorithms. O.C. acknowledges support from the Lundbeck Foundation, the Danish e-infrastructure Cooperation (DeiC), and the Danish Council for Independent Research through a Sapere Aude III Grant (No. DFF-4002-00015).

¹M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," *Ann. Phys.* **389**(20), 457–484 (1927).

²T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, 2000).

³J. M. Bowman, "Self-consistent field energies and wavefunctions for coupled oscillators," *J. Chem. Phys.* **68**(2), 608 (1978).

⁴J. M. Bowman, "The self-consistent-field approach to polyatomic vibrations," *Acc. Chem. Res.* **19**(7), 202–208 (1986).

⁵S. Carter, S. J. Culik, and J. M. Bowman, "Vibrational self-consistent field method for many-mode systems: A new approach and application to the vibrations of CO adsorbed on Cu(100)," *J. Chem. Phys.* **107**(24), 10458–10469 (1997).

- ⁶M. B. Hansen, M. Sparta, P. Seidler, D. Toffoli, and O. Christiansen, "New formulation and implementation of vibrational self-consistent field theory," *J. Chem. Theory Comput.* **6**(1), 235–248 (2010).
- ⁷G. Rauhut, "Configuration selection as a route towards efficient vibrational configuration interaction calculations," *J. Chem. Phys.* **127**(18), 184109 (2007).
- ⁸M. Neff and G. Rauhut, "Toward large scale vibrational configuration interaction calculations," *J. Chem. Phys.* **131**(12), 124129 (2009).
- ⁹O. Christiansen, "A second quantization formulation of multimode dynamics," *J. Chem. Phys.* **120**(5), 2140 (2004).
- ¹⁰L. S. Norris, M. A. Ratner, A. E. Roitberg, and R. B. Gerber, "Møller-Plesset perturbation theory applied to vibrational problems," *J. Chem. Phys.* **105**(24), 11261–11267 (1996).
- ¹¹J. O. Jung and R. B. Gerber, "Vibrational wave functions and spectroscopy of (H₂O)_n, n = 2, 3, 4, 5: Vibrational self-consistent field with correlation corrections," *J. Chem. Phys.* **105**(23), 10332–10348 (1996).
- ¹²O. Christiansen, "Møller-Plesset perturbation theory for vibrational wave functions," *J. Chem. Phys.* **119**(12), 5773 (2003).
- ¹³O. Christiansen, "Vibrational coupled cluster theory," *J. Chem. Phys.* **120**(5), 2149 (2004).
- ¹⁴P. Seidler and O. Christiansen, "Automatic derivation and evaluation of vibrational coupled cluster theory equations," *J. Chem. Phys.* **131**(23), 234109 (2009).
- ¹⁵P. Seidler, M. B. Hansen, and O. Christiansen, "Towards fast computations of correlated vibrational wave functions: Vibrational coupled cluster response excitation energies at the two-mode coupling level," *J. Chem. Phys.* **128**(15), 154113 (2008).
- ¹⁶P. Seidler and O. Christiansen, "Vibrational coupled cluster theory," in *Recent Progress in Coupled Cluster Methods: Theory and Applications*, edited by P. Čársky, J. Paldus, and J. Pittner (Springer Netherlands, Dordrecht, 2010), pp. 491–512.
- ¹⁷S. Banik, S. Pal, and M. D. Prasad, "Study of molecular vibration by coupled cluster method: Bosonic approach," *AIP Conf. Proc.* **1642**(1), 227–230 (2015).
- ¹⁸J. A. Fauchaux and S. Hirata, "Higher-order diagrammatic vibrational coupled-cluster theory," *J. Chem. Phys.* **143**(13), 134105 (2015).
- ¹⁹B. Thomsen, M. B. Hansen, P. Seidler, and O. Christiansen, "Vibrational absorption spectra from vibrational coupled cluster damped linear response functions calculated using an asymmetric Lanczos algorithm," *J. Chem. Phys.* **136**(12), 124101 (2012).
- ²⁰I. H. Godtlielsen and O. Christiansen, "A band Lanczos approach for calculation of vibrational coupled cluster response functions: Simultaneous calculation of IR and Raman anharmonic spectra for the complex of pyridine and a silver cation," *Phys. Chem. Chem. Phys.* **15**(25), 10035 (2013).
- ²¹I. H. Godtlielsen and O. Christiansen, "Calculating vibrational spectra without determining excited eigenstates: Solving the complex linear equations of damped response theory for vibrational configuration interaction and vibrational coupled cluster states," *J. Chem. Phys.* **143**(13), 134108 (2015).
- ²²C. König and O. Christiansen, "Linear-scaling generation of potential energy surfaces using a double incremental expansion," *J. Chem. Phys.* **145**(6), 064105 (2016).
- ²³F. Richter, P. Carbonniere, and C. Pouchan, "Toward linear scaling: Locality of potential energy surface coupling in valence coordinates," *Int. J. Quantum Chem.* **114**(20), 1401–1411 (2014).
- ²⁴I. H. Godtlielsen, B. Thomsen, and O. Christiansen, "Tensor decomposition and vibrational coupled cluster theory," *J. Phys. Chem. A* **117**, 7267–7279 (2013).
- ²⁵I. H. Godtlielsen, M. B. Hansen, and O. Christiansen, "Tensor decomposition techniques in the solution of vibrational coupled cluster response theory eigenvalue equations," *J. Chem. Phys.* **142**(2), 024105 (2015).
- ²⁶N. K. Madsen, I. H. Godtlielsen, and O. Christiansen, "Efficient algorithms for solving the non-linear vibrational coupled-cluster equations using full and decomposed tensors," *J. Chem. Phys.* **146**(13), 134110 (2017).
- ²⁷F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.* **6**(1-4), 164–189 (1927).
- ²⁸F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *J. Math. Phys.* **7**(1-4), 39–79 (1928).
- ²⁹J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika* **35**(3), 283–319 (1970).
- ³⁰T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.* **51**(3), 455–500 (2009).
- ³¹E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemom.* **25**(2), 67–86 (2011).
- ³²M. Espig and W. Hackbusch, "A regularized Newton method for the efficient approximation of tensors represented in the canonical tensor format," *Numerische Math.* **122**(3), 489–525 (2012).
- ³³M. Espig, W. Hackbusch, T. Rohwedder, and R. Schneider, "Variational calculus with sums of elementary tensors of fixed rank," *Numerische Math.* **122**(3), 469–488 (2012).
- ³⁴G. Beylkin and M. J. Mohlenkamp, "Numerical operator calculus in higher dimensions," *Proc. Natl. Acad. Sci. U. S. A.* **99**(16), 10246–10251 (2002).
- ³⁵G. Beylkin and M. J. Mohlenkamp, "Algorithms for numerical analysis in high dimensions," *SIAM J. Sci. Comput.* **26**(6), 2133–2159 (2005).
- ³⁶C. M. Andersen and R. Bro, "Practical aspects of PARAFAC modeling of fluorescence excitation-emission data," *J. Chemom.* **17**(4), 200–215 (2003).
- ³⁷J.-P. Royer, N. Thirion-Moreau, P. Comon, R. Redon, and S. Mounier, "A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3D fluorescence spectroscopy: Regularized NNCPPD and preprocessing for 3D fluorescence," *J. Chemom.* **29**(4), 253–265 (2015).
- ³⁸U. Benedikt, A. A. Auer, M. Espig, and W. Hackbusch, "Tensor decomposition in post-Hartree-Fock methods. I. Two-electron integrals and MP2," *J. Chem. Phys.* **134**(5), 054118 (2011).
- ³⁹U. Benedikt, K.-H. Böhm, and A. A. Auer, "Tensor decomposition in post-Hartree-Fock methods. II. CCD implementation," *J. Chem. Phys.* **139**(22), 224101 (2013).
- ⁴⁰K.-H. Böhm, A. A. Auer, and M. Espig, "Tensor representation techniques for full configuration interaction: A Fock space approach using the canonical product format," *J. Chem. Phys.* **144**(24), 244102 (2016).
- ⁴¹F. A. Bischoff and E. F. Valeev, "Low-order tensor approximations for electronic wave functions: Hartree-Fock method with guaranteed precision," *J. Chem. Phys.* **134**(10), 104104 (2011).
- ⁴²F. A. Bischoff, R. J. Harrison, and E. F. Valeev, "Computing many-body wave functions with guaranteed precision: The first-order Møller-Plesset wave function for the ground state of helium atom," *J. Chem. Phys.* **137**(10), 104103 (2012).
- ⁴³E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, "Tensor hypercontraction density fitting. I. Quartic scaling second- and third-order Møller-Plesset perturbation theory," *J. Chem. Phys.* **137**(4), 044103 (2012).
- ⁴⁴R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, "Tensor hypercontraction. II. Least-squares renormalization," *J. Chem. Phys.* **137**(22), 224106 (2012).
- ⁴⁵G. Schmitz, N. K. Madsen, and O. Christiansen, "Atomic-batched tensor decomposed two-electron repulsion integrals," *J. Chem. Phys.* **146**(13), 134112 (2017).
- ⁴⁶A. Leclerc and T. Carrington, "Calculating vibrational spectra with sum of product basis functions without storing full-dimensional vectors or matrices," *J. Chem. Phys.* **140**(17), 174111 (2014).
- ⁴⁷P. S. Thomas and T. Carrington, "Using nested contractions and a hierarchical tensor format to compute vibrational spectra of molecules with seven atoms," *J. Phys. Chem. A* **119**(52), 13074–13091 (2015).
- ⁴⁸P. S. Thomas and T. Carrington, "An intertwined method for making low-rank, sum-of-product basis functions that makes it possible to compute vibrational spectra of molecules with more than 10 atoms," *J. Chem. Phys.* **146**(20), 204110 (2017).
- ⁴⁹L. Ostrowski, B. Ziegler, and G. Rauhut, "Tensor decomposition in potential energy surface representations," *J. Chem. Phys.* **145**(10), 104103 (2016).
- ⁵⁰P. Rai, K. Sargsyan, H. Najm, M. R. Hermes, and S. Hirata, "Low-rank canonical-tensor decomposition of potential energy surfaces: Application to grid-based diagrammatic vibrational Green's function theory," *Mol. Phys.* **115**, 2120 (2017).
- ⁵¹M. H. Beck, A. Jackle, G. A. Worth, and H. D. Meyer, "The multiconfiguration time-dependent Hartree (MCTDH) method: A highly efficient algorithm for propagating wavepackets," *Phys. Rep.* **324**(1), 1–105 (2000).
- ⁵²J. K. G. Watson, "Simplification of the molecular vibration-rotation Hamiltonian," *Mol. Phys.* **100**(1), 47–54 (2002).
- ⁵³P. Seidler, E. Matito, and O. Christiansen, "Vibrational coupled cluster theory with full two-mode and approximate three-mode couplings: The VCC[2pt3] model," *J. Chem. Phys.* **131**(3), 034115 (2009).
- ⁵⁴A. Zocante, P. Seidler, M. B. Hansen, and O. Christiansen, "Approximate inclusion of four-mode couplings in vibrational coupled-cluster theory," *J. Chem. Phys.* **136**(20), 204118 (2012).
- ⁵⁵M. Ziolkowski, V. Weijo, P. Jørgensen, and J. Olsen, "An efficient algorithm for solving nonlinear equations with a minimal number of trial vectors:

- Applications to atomic-orbital based coupled-cluster theory,” *J. Chem. Phys.* **128**(20), 204105 (2008).
- ⁵⁶J. Håstad, “Tensor rank is NP-complete,” *J. Algorithms* **11**(4), 644–654 (1990).
- ⁵⁷V. Khoromskaia and B. N. Khoromskij, “Tensor numerical methods in quantum chemistry: From Hartree-Fock to excitation energies,” *Phys. Chem. Chem. Phys.* **17**, 31491–31509 (2015).
- ⁵⁸W. W. Hager and H. Zhang, “A survey of nonlinear conjugate gradient methods,” *Pac. J. Optim.* **2**(1), 35–58 (2006).
- ⁵⁹J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. (Springer, 2006).
- ⁶⁰J. J. Moré and D. J. Thuente, “Line search algorithms with guaranteed sufficient decrease,” *ACM Trans. Math. Software* **20**(3), 286–307 (1994).
- ⁶¹W. W. Hager and H. Zhang, “A new conjugate gradient method with guaranteed descent and an efficient line search,” *SIAM J. Optim.* **16**(1), 170–192 (2005).
- ⁶²K.-H. Böhm, “Anwendung von tensorapproximationen auf die full configuration interaction methode,” PhD thesis, TU Chemnitz, 2016.
- ⁶³M. Bachmayr, R. Schneider, and A. Uschmajew, “Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations,” *Found. Comput. Math.* **16**(6), 1423–1472 (2016).
- ⁶⁴M. Espig, L. Grasedyck, and W. Hackbusch, “Black box low tensor-rank approximation using Fiber-crosses,” *Constructive Approximation* **30**(3), 557–597 (2009).
- ⁶⁵M. J. Reynolds, A. Doostan, and G. Beylkin, “Randomized alternating least squares for canonical tensor decompositions: Application to a PDE with random data,” *SIAM J. Sci. Comput.* **38**(5), A2634–A2664 (2016).
- ⁶⁶O. Christiansen, I. H. Godtliebsen, E. Matito Gras, W. Györfy, M. Böttger Hansen, M. Bo Hansen, J. Kongsted, E. Lund Klinting, C. König, S. A. Losilla, D. Madsen, N. Kristian Madsen, P. Seidler, K. Sneskov, M. Sparta, B. Thomsen, D. Toffoli, and A. Zocante, MidasCpp (Molecular Interactions, Dynamics and Simulation Chemistry Program Package in C++), University of Aarhus, 2016, www.chem.au.dk/midas.
- ⁶⁷C. König, M. B. Hansen, I. H. Godtliebsen, and O. Christiansen, “Falcon: A method for flexible adaptation of local coordinates of nuclei,” *J. Chem. Phys.* **144**(7), 074108 (2016).
- ⁶⁸E. L. Klinting, C. König, and O. Christiansen, “Hybrid optimized and localized vibrational coordinates,” *J. Phys. Chem. A* **119**(44), 11007–11021 (2015).
- ⁶⁹C. König and O. Christiansen, “Automatic determination of important mode–mode correlations in many-mode vibrational wave functions,” *J. Chem. Phys.* **142**(14), 144115 (2015).