

RESEARCH ARTICLE | AUGUST 12 2024

## FORTE: A suite of advanced multireference quantum chemistry methods **FREE**

Special Collection: **Modular and Interoperable Software for Chemical Physics**

Francesco A. Evangelista  ; Chenyang Li  ; Prakash Verma  ; Kevin P. Hannon  ; Jeffrey B. Schriber  ; Tianyuan Zhang  ; Chenxi Cai; Shuhe Wang; Nan He; Nicholas H. Stair  ; Meng Huang  ; Renke Huang  ; Jonathon P. Misiewicz  ; Shuhang Li  ; Kevin Marin  ; Zijun Zhao  ; Lori A. Burns 

 Check for updates

*J. Chem. Phys.* 161, 062502 (2024)  
<https://doi.org/10.1063/5.0216512>



 Nanotechnology & Materials Science

 Optics & Photonics

 Impedance Analysis

 Scanning Probe Microscopy

 Sensors

 Failure Analysis & Semiconductors

 **Unlock the Full Spectrum.**  
From DC to 8.5 GHz.  
Your Application. Measured.  
[Find out more](#)

# FORTE: A suite of advanced multireference quantum chemistry methods

Cite as: *J. Chem. Phys.* **161**, 062502 (2024); doi: 10.1063/5.0216512

Submitted: 29 April 2024 • Accepted: 24 June 2024 •

Published Online: 12 August 2024



View Online



Export Citation



CrossMark

Francesco A. Evangelista,<sup>1,a)</sup> Chenyang Li,<sup>2</sup> Prakash Verma,<sup>1</sup> Kevin P. Hannon,<sup>1</sup> Jeffrey B. Schriber,<sup>1,3</sup> Tianyuan Zhang,<sup>1</sup> Chenxi Cai,<sup>1</sup> Shuhe Wang,<sup>1</sup> Nan He,<sup>1</sup> Nicholas H. Stair,<sup>1</sup> Meng Huang,<sup>1</sup> Renke Huang,<sup>1</sup> Jonathon P. Misiewicz,<sup>1</sup> Shuhang Li,<sup>1</sup> Kevin Marin,<sup>1</sup> Zijun Zhao,<sup>1</sup> and Lori A. Burns<sup>4</sup>

## AFFILIATIONS

<sup>1</sup> Department of Chemistry and Cherry Emerson Center for Scientific Computation, Emory University, Atlanta, Georgia 30322, USA

<sup>2</sup> Key Laboratory of Theoretical and Computational Photochemistry, Ministry of Education, College of Chemistry, Beijing Normal University, Beijing 100875, China

<sup>3</sup> Department of Chemistry and Biochemistry, Iona University, New Rochelle, New York 10801, USA

<sup>4</sup> Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry, Georgia Institute of Technology, Atlanta, Georgia 30332-0400, USA

**Note:** This paper is part of the JCP Special Topic on Modular and Interoperable Software for Chemical Physics.

**a)** Author to whom correspondence should be addressed: [francesco.evangelista@emory.edu](mailto:francesco.evangelista@emory.edu)

## ABSTRACT

FORTE is an open-source library specialized in multireference electronic structure theories for molecular systems and the rapid prototyping of new methods. This paper gives an overview of the capabilities of FORTE, its software architecture, and examples of applications enabled by the methods it implements.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0216512>

## I. INTRODUCTION

Software development plays a critical role in advancing quantum chemistry,<sup>1</sup> enabling the exploration of new fundamental theoretical ideas and modeling systems of ever-increasing complexity.<sup>2–22</sup> In the past decade, the availability of quantum chemistry packages that use modular designs and provide application programming interfaces (APIs) has enabled the creation of specialized software plugins, enhancing the capabilities of the original codes.<sup>10,23–29</sup> The availability of well-documented APIs is particularly beneficial in the context of academic scientific software development because it reduces the entry barrier for new developers and shields them from the complexities of large software projects.

A prime example of these benefits is FORTE, an open-source plugin developed in our group for the quantum chemistry package Psi4.<sup>19,23,24</sup> Initially a C++ plugin for Psi3,<sup>30</sup> FORTE has transformed into a sophisticated software suite with a Python interface, illustrating the evolution and integration facilitated by modern APIs. The rich quantum chemistry functionality provided by Psi4's API and the continued improvements made by its dedicated developer

community have allowed the developers of FORTE to concentrate efforts on pioneering new quantum chemistry methods, accelerating its evolution.

FORTE specializes in the implementation of multireference electronic structure theories,<sup>31–33</sup> focusing on methods that can be applied to a variety of challenging problems, including predicting global potential energy surfaces and modeling open-shell spin-coupled systems and multi-electron excited states. Multireference methods assume a correlated reference wave function as a starting point for describing a target electronic state. This reference state is assumed to account for static electron correlation effects, which can be captured only by a linear combination of Slater determinants. A major challenge in the electronic structure theory is creating multireference approaches that can achieve high accuracy, especially in scenarios where single-reference methods, such as coupled-cluster theory, fall short.<sup>32,34,35</sup> Issues with the numerical stability and generality of the resulting methods have been significant obstacles in virtually all attempts to generalize coupled-cluster theory to the multireference case. Perhaps, the most important distinguishing feature

of FORTE is its implementation of a promising class of multireference methods based on the driven similarity renormalization group (DSRG).<sup>36</sup> The DSRG methods implemented in FORTE mitigate the numerical instabilities connected with the intruder state problem in perturbation theory, are systematically improvable, yield continuous potential energy surfaces, have lower computational requirements than similar methods, and can be scaled to large active spaces. In addition to implementing DSRG methods, FORTE also offers a variety of complementary tools, ranging from standard and adaptive active space solvers, multireference quantum embedding methods, orbital transformations, and a convenient framework for the rapid prototyping of new many-body methods.

In this article, we provide a concise review of the theoretical methods underpinning FORTE (Sec. II), describe its software design principles and capabilities (Sec. III), and showcase representative applications (Sec. IV). We conclude by discussing the current limitations and future development plans (Sec. V).

## II. METHODS IMPLEMENTED IN FORTE

An overview of the main methods and functionalities implemented in FORTE is presented in Table I. FORTE specializes in

implementing multireference quantum chemistry methods based on the active space formalism and its generalizations.<sup>37</sup> The basic assumption of active space methods is that of the spin orbital basis  $\{\psi_p\}$  (taken to be orthonormal), only a subset of spin orbitals  $\mathbb{A}$  termed “active” are essential to approximate the set of electronic states targeted. The remaining spin orbitals are assumed to play a secondary role and are taken to be either occupied by an electron (core,  $\mathbb{C}$ ) or unoccupied (virtual,  $\mathbb{V}$ ). It should be noted that this partitioning need not be based on the orbital energy. In the following, we denote the size of the spin orbital basis as  $N$ , and the size of the core, active, and virtual subsets as  $N_{\mathbb{C}}$ ,  $N_{\mathbb{A}}$ , and  $N_{\mathbb{V}}$ , respectively.

FORTE is a code for molecular computations, and as such, it can handle general fermionic Hamiltonians with up to two-body interactions of the form

$$\hat{H} = V + \sum_{pq} h_p^q \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4} \sum_{pqrs} v_{pq}^{rs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r, \quad (1)$$

where  $V$  is a scalar term,  $h_p^q = \langle \psi_p | \hat{h} | \psi_q \rangle$  is a matrix element of the one-electron kinetic plus potential operator ( $\hat{h}$ ), and  $v_{pq}^{rs} = \langle pq | rs \rangle = \langle \psi_p(1) \psi_q(2) | r_{12}^{-1} | \psi_r(1) \psi_s(2) \rangle - \langle \psi_p(1) \psi_q(2) | r_{12}^{-1} | \psi_s(1) \psi_r(2) \rangle$  is an antisymmetrized matrix element of the Coulomb operator.

**TABLE I.** Methods and features implemented in FORTE.

Methods	Notes <sup>a</sup>	References
FCI	14–16 active orbitals	
CASCI/CASSCF	14–16 active orbitals	
GASCI/GASSCF	Supports up to six spaces	89
ACI/ACISCF	20–30 active orbitals	48 and 51
PCI	20–100 active orbitals	53
DMRG	Supported via interface to CHEMPS2 and Block2	90 and 91
DSRG methods	All the DSRG methods support reference relaxation, spin adaptation, and state-averaging	65
DSRG-MRPT2	~2000 AOs with DF. Analytic energy gradients available for the unrelaxed energy	92 and 93
DSRG-MRPT3	~750 AOs with DF, ~1450 AOs with DF and FNO	66
LDSRG(2)/sq-LDSRG(2)	~550 AOs with DF and NIVO, ~1050 AOs with DF, NIVO, and FNO	67 and 68
ASET(mf)	Environment treated at the mean-field level (CASSCF)	94
Relativistic Hamiltonian	One-electron scalar relativistic Hamiltonian (1e-X2C) supported for all methods via Psi4	19 and 95
AVAS	Supports the selection of atoms, AOs, and planes and average planes for $\pi$ orbitals	96 and 97
Integrals	Conventional, density fitting, Cholesky decomposition	
Frozen natural orbitals	Computed from the DSRG-MRPT2 second-order unrelaxed 1-RDM	98
Orbital localization	Pipek–Mezey supported via an interface to Psi4	19

<sup>a</sup>The ranges given in this column are the expected maximum capability of FORTE expressed in terms of the number of atomic orbitals (AOs) or active orbitals.

## A. Active space solvers

Active space solvers are one of the fundamental categories of methods implemented in FORTE. Active space solvers primarily provide a zeroth-order description of multideterminantal electronic states and account for static electron correlation within the active orbitals; however, they may also be used to perform high-accuracy computations on systems with few electrons/orbitals.

### 1. Complete and occupation restricted configuration interaction

Most active space solvers implemented by FORTE are based on configuration interaction (CI) and expand the eigenstates in a determinant basis  $M = \{\Phi_\mu, \mu = 1, \dots, d\}$  termed model space. A generic Slater determinant may be written as a product of creation operators labeled by core and active indices acting on the vacuum state ( $|-\rangle$ ),

$$|\Phi_\mu\rangle = \hat{a}_u^\dagger \hat{a}_v^\dagger \cdots \prod_{m \in \mathbb{C}} \hat{a}_m^\dagger |-\rangle, \quad (2)$$

where  $\hat{a}_u^\dagger \hat{a}_v^\dagger \cdots$  is one of the  $d$  ways of creating a given number of electrons in the active orbitals, such that  $\Phi_\mu$  belongs to the desired spin and point group irreducible representation of the target electronic state. In the occupation number representation, each model space determinant is mapped onto a vector representing the occupation number ( $n_{u_i} \in \{0, 1\}$ ) of each active spin orbital,

$$|\Phi_\mu\rangle \rightarrow \underbrace{|n_{u_1} n_{u_2} \cdots|}_{N_A}. \quad (3)$$

A general CI state  $\Psi_\alpha$  is a linear combination of model space determinants with expansion coefficients  $c_{\mu\alpha}$ ,

$$|\Psi_\alpha\rangle = \sum_{\Phi_\mu \in M} c_{\mu\alpha} |\Phi_\mu\rangle. \quad (4)$$

FORTE implements configuration interaction using the string-based approach of Handy<sup>38</sup> and the vectorized Bendazzoli–Evangelisti algorithm to build the  $\sigma$  vector.<sup>39</sup> In string-based codes, each determinant is expressed as the tensor product of two “strings” of alpha and beta occupation numbers ( $I_\alpha / I_\beta$ ),

$$|\Phi_\mu\rangle = |I_\alpha\rangle \otimes |I_\beta\rangle. \quad (5)$$

When expressed in a basis of alpha and beta strings, the CI coefficient vector for a given state may be considered as a matrix  $c_\mu \equiv c_{I_\alpha, I_\beta}$ , labeled by the indices of the two strings. The efficient implementation of string-based CI relies on functionality to associate a unique numerical index (without gaps) to each string and pre-computed lists of determinants connected by one- and two-electron substitution operators.<sup>40</sup> While efficient graphical approaches for addressing strings and determinants exist,<sup>41</sup> FORTE instead relies on a hash table that maps strings to their address, using the C++ `std::unordered_map` container.

FORTE also implements occupation-restricted configuration interaction. This is particularly useful to reduce the size of the configuration interaction space by eliminating “deadwood”<sup>42</sup> or to study electronic states characterized by specific orbital occupations (e.g., core-excited states). If the active space is further partitioned into disjoint spaces,  $\mathbb{A} = \cup_{k=1}^n \mathbb{A}_k$ , it is possible to truncate the model space to

determinants that satisfy a specific set of occupation restrictions. In FORTE, these restrictions are imposed both at the level of  $\alpha/\beta$  strings and on the allowed combinations of such strings. The resulting code is capable of handling restricted active spaces (RAS),<sup>41</sup> generalized active spaces (GAS),<sup>41,43,44</sup> and occupation-restricted multiple active spaces (ORMAS).<sup>45</sup> For all the CI codes, it is possible to enable projection of the CI wave function on a subspace with well-defined spin quantum numbers for the total  $\hat{S}^2$  and  $\hat{S}_z$  spin operators. The procedure employed by FORTE transforms the CI expansion from the Slater determinant basis into the configuration state function basis<sup>46</sup> and performs the Davidson–Liu diagonalization procedure in this basis. This approach has been recently revisited by others.<sup>46,47</sup>

The string-based CI codes implement algorithms to compute  $k$ -body reduced density matrices ( $k$ -RDMs) for all the CI methods, where  $k$  ranges from one to three, either as spinful or spin-free quantities. Given a CI state  $\Psi_\alpha$ , in the spinful case, these quantities are denoted by  $\gamma_k^\alpha$  and their elements are defined as

$$(\gamma_1^\alpha)_u^x = \langle \Psi_\alpha | \hat{a}_x^\dagger \hat{a}_u | \Psi_\alpha \rangle, \quad (6)$$

$$(\gamma_2^\alpha)_{uv}^{xy} = \langle \Psi_\alpha | \hat{a}_x^\dagger \hat{a}_y^\dagger \hat{a}_v \hat{a}_u | \Psi_\alpha \rangle, \quad (7)$$

$$(\gamma_3^\alpha)_{uvw}^{xyz} = \langle \Psi_\alpha | \hat{a}_x^\dagger \hat{a}_y^\dagger \hat{a}_z^\dagger \hat{a}_w \hat{a}_v \hat{a}_u | \Psi_\alpha \rangle, \quad (8)$$

where the indices  $u, v, w, x, y, z \in \mathbb{A}$  run over the active orbitals. FORTE also supports the evaluation of the one-body transition density matrix ( $\gamma_1^{\alpha\beta}$ ), defined as

$$(\gamma_1^{\alpha\beta})_u^x = \langle \Psi_\alpha | \hat{a}_x^\dagger \hat{a}_u | \Psi_\beta \rangle \quad u, x \in \mathbb{A}, \quad (9)$$

which is used to evaluate transition properties between two states  $\Psi_\alpha$  and  $\Psi_\beta$  expressed in a common spin orbital basis.

### 2. Determinantal and selected configuration interaction

A second set of active solvers is based on general CI expansions and employs a determinant-based approach, whereby a state is expressed directly in terms of determinants selected from the FCI space. FORTE implements the adaptive configuration interaction (ACI),<sup>48</sup> a variant of the CIPSI method,<sup>49</sup> which aims to identify a model space  $M$  such that the variational energy of a CI expansion in this space ( $E_M$ ) approximately matches the CASCI energy to within a user-provided parameter  $\sigma$ ,

$$|E_{\text{CASCI}} - E_M| \approx \sigma. \quad (10)$$

In ACI, the model space is identified by iteratively growing a set of reference determinants ( $P$ ) and screening determinants  $\Phi_\mu$  in its first-order interacting space ( $F$ ), using an estimate of the energetic contributions of each new determinant  $e(\Phi_\mu)$ . This energy estimate is obtained by diagonalizing a  $2 \times 2$  Hamiltonian matrix built from the current CI state  $\Psi_\alpha$  and the determinant under consideration  $\Phi_\mu$ . In the screening process, error control is enforced by excluding determinants with the smallest energy estimates, such that the sum of these perturbative corrections approximately equals the energy threshold  $\sigma$ . Specifically, at each iteration of the ACI procedure, we

enforce the following condition on the determinants excluded from selection:

$$\sum_{\Phi_\mu}^{\text{excluded}} |\epsilon(\Phi_\mu)| \leq \sigma. \quad (11)$$

As customary in selected CI, the ACI energy can be improved by including the estimated energy contribution of those determinants not included in the reference space. Using ACI, we have been able to push CI computations up to active spaces with as many as 30 electrons in 30 orbitals.<sup>50</sup> Extensions of ACI for computing excited states<sup>51</sup> and modeling the ultrafast dynamics of electrons have also been implemented in FORTE.<sup>52</sup> FORTE also implements the projector configuration interaction (PCI) method,<sup>53</sup> a deterministic version of the full configuration interaction quantum Monte Carlo,<sup>54</sup> which applies determinant selection to the imaginary-time propagator.

## B. Orbital optimization and state averaging

FORTE implements a general algorithm for the self-consistent orbital optimization of multideterminantal wave functions. In the most general form, orbital optimization may be formulated as a minimization problem, where the central quantity to optimize is the weighted average of the energy of a set of states  $S = \{\Psi_\alpha, \alpha = 1, \dots, n\}$ . FORTE implements a general state averaging procedure that allows the user to specify a set of states  $S$  with different electron number, spin/point-group symmetry, and restrictions on the occupation of each space  $\mathbb{A}_k$ . The corresponding Lagrangian is given by the average energy plus constraints for orbital orthonormality [ $\mathcal{W}(\{C\})$ ], variational CI conditions [ $\mathcal{X}(\{c^\alpha\}, C)$ ], and CI orthonormality conditions [ $\mathcal{Y}(\{c^\alpha\})$ ],

$$\mathcal{L}(\{c^\alpha\}, C) = \sum_{\alpha=1}^n \omega_\alpha \langle \Psi_\alpha | \hat{H} | \Psi_\alpha \rangle + \mathcal{W}(\{C\}) + \mathcal{X}(\{c^\alpha\}, C) + \mathcal{Y}(\{c^\alpha\}), \quad (12)$$

where  $\omega_\alpha \geq 0$  is the weight of state  $\Psi_\alpha$  and the weights add up to one.

In FORTE, the orbital optimization is performed via the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm, initialized using the modified diagonal Hessian suggested in Ref. 55.

To compute the averaged electronic energy, orbital gradients, and the diagonal Hessian, the inactive and active Fock matrices and the two-electron integrals of the form  $\langle px|uy \rangle$  are required. We obtain these integrals using Psi4's atomic-orbital (AO) version of the Coulomb and exchange (JK) build algorithm, which is also used for building the Fock matrix in standard self-consistent-field procedures implemented in Psi4. For every  $(x, y)$  pair, we first evaluate the matrix  $J_{\mu\nu}^{(xy)} = \langle \mu x | \nu y \rangle$  using the Coulomb (J) builder, followed by a two-index integral transformation via the matrix multiplication  $\langle px|uy \rangle = \sum_{\mu\nu}^{\text{AO}} C_{\mu\mu} J_{\mu\nu}^{(xy)} C_{\nu\nu}$ . For MCSCF computations, the number of L-BFGS iterations is limited to 6–8 since strict convergence at each iteration is unnecessary. It should be noted that the Lagrangian may be expressed in terms of contractions of the one- and two-electron integrals with one- and two-RDMs averaged over the states. Therefore, due to the decoupling of the CI and orbital coefficients in the optimization, all active solvers that generate RDMs up to rank two may be interfaced with the optimizer to obtain variational orbitals.

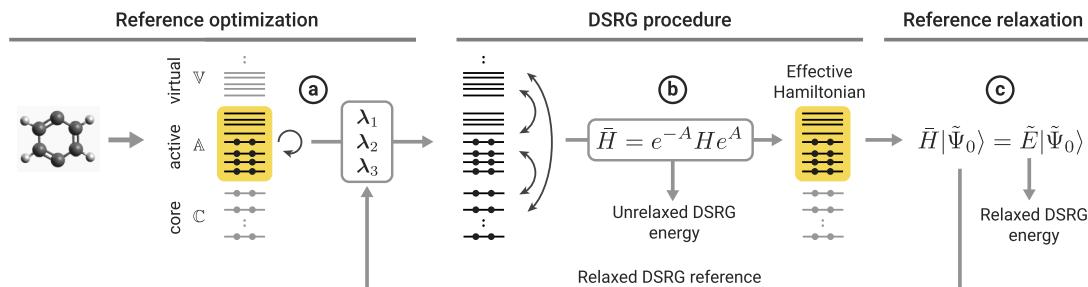
## C. Multireference theories of dynamical correlation

Active space solvers provide approximate reference states that do not account for dynamical correlation effects. A feature unique to FORTE is implementing a wide range of dynamical correlation solvers based on the driven similarity renormalization group (DSRG).<sup>36</sup> These solvers can be combined with any active space solver to account for the combined effect of static and dynamic electron correlation. An overview of the various steps of a multireference DSRG (MR-DSRG) computation is shown in Fig. 1.

The DSRG approach is formally related to unitary multireference coupled cluster theory<sup>57</sup> and analogous approaches.<sup>58–60</sup> In the DSRG, the second-quantized (bare) Hamiltonian [Eq. (1)] is brought to a block-diagonal form by a unitary transformation that depends on a time-like parameter  $s$ ,

$$\hat{H} \mapsto \bar{H} = e^{-\hat{A}(s)} \hat{H} e^{\hat{A}(s)}. \quad (13)$$

The unitary transformation is written as the exponential of the anti-Hermitian operator  $\hat{A}(s)$ , preserving the Hermitian character of the bare Hamiltonian in the resulting transformed Hamiltonian  $\bar{H}$ . The parameter  $s$  controls the extent to which the Hamiltonian is



**FIG. 1.** Outline of the steps involved in MR-DSRG computations. The first step (a) optimizes the correlated reference state  $\Psi_0$  and evaluates its reduced density cumulants ( $\lambda_k$ ,  $k = 1, 2, 3$ ). The cumulants are then passed to the MR-DSRG algorithm (b) to produce the effective Hamiltonian ( $\bar{H}$ ) and its energy expectation value ( $\langle \Psi_0 | \bar{H} | \Psi_0 \rangle$ ). The reference state can be relaxed by diagonalizing  $\bar{H}$  to obtain a new reference state  $\tilde{\Psi}_0$  (c). The relaxed reference can be used as a starting point for a new MR-DSRG computation, and this procedure may be iterated until self-consistency is reached.

transformed. For  $s > 0$ , the DSRG method is designed to gradually suppress those excitations with energy denominators smaller than the energy cutoff  $\Lambda = s^{-1/2}$ , providing a way to mitigate the intruder state problem.

The anti-Hermitian operator is expressed in terms of an  $s$ -dependent cluster operator  $\hat{T}(s)$  as

$$\hat{A}(s) = \hat{T}(s) - \hat{T}^\dagger(s), \quad (14)$$

where  $\hat{T}(s)$  is a sum of many-body operators,  $\hat{T}(s) = \hat{T}_1(s) + \hat{T}_2(s) + \dots$ , and a generic  $k$ -body term  $\hat{T}_k(s)$  is written in terms of  $s$ -dependent cluster amplitudes  $t_{ab\dots}^{ij\dots}(s)$  and normal-ordered second quantized operators ( $\{\hat{a}_{ij\dots}^{ab\dots}\} = \{\hat{a}_a^\dagger \hat{a}_b^\dagger \dots \hat{a}_j \hat{a}_i\}$ ) as

$$\hat{T}_k(s) = \frac{1}{(k!)^2} \sum_{ij\dots} \sum_{ab\dots} t_{ab\dots}^{ij\dots}(s) \{\hat{a}_{ij\dots}^{ab\dots}\}. \quad (15)$$

The cluster amplitudes are antisymmetric with respect to separate permutations of upper or lower indices, and internal excitations (labeled only by active indices) are implicitly omitted from the definition of  $\hat{T}_k(s)$  since they are equivalent to unitary rotations among the model space determinants.

The general goal of the DSRG is to remove those terms in  $\bar{H}$  that send model space determinants into the orthogonal complement of  $M$ .<sup>61–63</sup> If we express the transformed Hamiltonian in normal-ordered form as

$$\bar{H}(s) = \bar{H}_0(s) + \sum_{pq} \bar{H}_p^q(s) \{\hat{a}_q^p\} + \frac{1}{4} \sum_{pqrs} \bar{H}_{pq}^{rs}(s) \{\hat{a}_{rs}^{pq}\} + \dots, \quad (16)$$

then the DSRG aims to remove terms such as, for example,  $\bar{H}_e^u(s) \{\hat{a}_u^e\}$  that replace an active orbital  $\psi_u$  with a virtual orbital  $\psi_e$ , transforming a model space determinant [see Eq. (2)] into a new determinant not contained in the model space. Here, normal ordering of the operators is assumed with respect to a correlated state  $\Psi_\alpha$  following the approach of Mukherjee,<sup>64</sup> so that the condition  $\langle \Psi_\alpha | \{\hat{a}_{rs}^{pq\dots}\} | \Psi_\alpha \rangle = 0$  is satisfied for any operator  $\hat{a}_{rs}^{pq\dots}$ . In the DSRG, the amplitudes that enter into  $\hat{A}(s)$  are obtained by solving the following set of many-body equations:<sup>65</sup>

$$\bar{H}_{ab\dots}^{ij\dots}(s) = r_{ab\dots}^{ij\dots}(s), \quad (17)$$

where  $r_{ab\dots}^{ij\dots}(s)$  is a function parametrized to match the first-order transformed Hamiltonian elements from the single-reference similarity renormalization group.<sup>65</sup> After solving the DSRG amplitude equations [Eq. (17)], the energy can be evaluated by taking the expectation value of  $\bar{H}$  with respect to the reference state  $\Psi_\alpha$ ,

$$E_\alpha(s) = \langle \Psi_\alpha | \bar{H}(s) | \Psi_\alpha \rangle. \quad (18)$$

Alternatively, it is possible to evaluate a relaxed energy ( $E'_\alpha$ ) and wave function ( $\Psi'_\alpha$ ) by diagonalizing  $\bar{H}$  in the basis of model space determinants,

$$\bar{H}(s) | \Psi'_\alpha(s) \rangle = E'_\alpha(s) | \Psi'_\alpha(s) \rangle, \quad (19)$$

yielding the relaxed reference state,

$$| \Psi'_\alpha(s) \rangle = \sum_{\Phi_\mu \in M} c'_{\mu\alpha} | \Phi_\mu \rangle. \quad (20)$$

In FORTE, the relaxation process may be performed once or alternated to the solution of the amplitude equations until self-consistency is reached.

FORTE implements three truncation levels for the DSRG. Two are approximations obtained by truncating the correlation energy to second- and third-order perturbation theory (DSRG-MRPT2/3). These approaches are non-iterative and have a scaling analogous to that of single-reference perturbation theories. In particular, the dominant cost in evaluating the second-order correlation energy is proportional to  $N_C^2 N_V^2$  (ignoring the cost of transforming the two-electron integrals), while the third-order procedure instead scales as  $N_C^2 N_V^4$ . For both the methods, the term with the steepest scaling with respect to the active space size is of order  $N_A^6 N_V$ . For these perturbative approaches, a single reference relaxation step can be performed at a negligible extra cost, and it is already sufficient to capture the largest improvements to the correlation energy. Given its favorable accuracy/cost ratio, this is the recommended procedure for running perturbative MR-DSRG computations.<sup>66</sup>

A nonperturbative approximation in which the  $\hat{A}(s)$  operator is truncated to one- and two-body terms [i.e.,  $\hat{A}(s) \approx \hat{A}_1(s) + \hat{A}_2(s)$ ] is also implemented. This approach, termed MR-LDSRG(2),<sup>67</sup> truncates each commutator in the Baker–Campbell–Hausdorff (BCH) expansion of  $\bar{H}$  to only the zero-, one-, and two-body components,<sup>59</sup>

$$[\cdot, \hat{A}(s)] \approx \sum_{k=0}^2 [\cdot, \hat{A}(s)]_k, \quad (21)$$

where  $[\cdot, \hat{A}(s)]_k$  is the  $k$ -body component of the commutator. This approximation is then applied recursively to all the terms that arise from the BCH expansion. The cost to evaluate a single commutator [Eq. (21)] scales as  $O(N_C^2 N_V^2 N^2) = O(N_C^2 N_V^4 + N_C^3 N_V^3 + \dots)$ , with the leading order being identical to the scaling of coupled-cluster with singles and doubles (CCSD) [ $O(N_C^2 N_V^4)$ ]. In typical problems, converging the Frobenius norm of  $\bar{H}$  to within  $10^{-12} E_h$  requires ten evaluations of the commutator  $[\cdot, \hat{A}(s)]$ , implying that the MR-LDSRG(2) method is about ten times more expensive than CCSD. In addition, one must store large intermediate tensors of the size of  $O(N^4)$ , limiting computation on a single computer node to 200–300 basis functions. To overcome the limitations of the MR-LDSRG(2) method, the non-interacting virtual orbital (NIVO) approximation is introduced for  $[\cdot, \hat{A}(s)]_2$  in Eq. (21), whereby two-body intermediates labeled by three and four virtual indices are neglected.<sup>68</sup>

FORTE also implements a sequential version of the MR-LDSRG(2) approximation [sq-MR-LDSRG(2)],<sup>68</sup> based on the following stepwise transformation of the Hamiltonian:

$$\bar{H}_{sq} = e^{-\hat{A}_2(s)} e^{-\hat{A}_1(s)} \hat{H} e^{\hat{A}_1(s)} e^{\hat{A}_2(s)}. \quad (22)$$

The advantage of this approach is that the inner  $\hat{A}_1$ -dressed Hamiltonian,  $e^{-\hat{A}_1(s)} \hat{H} e^{\hat{A}_1(s)}$ , can be computed exactly and amounts to a transformation of the one- and two-electron integrals.<sup>69,70</sup> When combined with density-fitted or Cholesky-decomposed integrals, the cost of this inner transformation scales as  $O(KN^3)$ , where  $K$  is the number of auxiliary basis functions, an effort negligible compared to the cost of evaluating the commutator. All the approximate MR-DSRG schemes implemented in FORTE require as input the one- and two-electron integrals and reduced density matrices of

the reference state up to rank three, offering an advantage over other multireference methods that require higher-order RDMs.<sup>71–74</sup> We have also explored the possibility of truncating the three-body RDM in DSRG-MRPT2 to further reduce the storage and computational cost. Our results show that cumulant truncation schemes are particularly useful in applications with large active spaces ( $N_A > 10–20$ ).<sup>75</sup>

Extensive benchmarks of the various DSRG methods implemented in FORTE have been reported in the literature.<sup>56,66,67,76–78</sup> In Fig. 2, we show some of the results reported in Ref. 56, which examined error statistics for equilibrium bond distance ( $r_e$ ) and the dissociation energy ( $D_0$ , including zero-point vibrational corrections) of 33 open- and closed-shell diatomic molecules computed with spin-adapted MR-DSRG methods. These computations employ a full-valence CASSCF reference state as well as the cc-pVQZ and cc-pCVQZ basis sets<sup>79,80</sup> and are compared to coupled-cluster theory with singles and doubles (CCSD),<sup>81</sup> CCSD with non-iterative triples corrections [CCSD(T)],<sup>82</sup> and various multireference approaches.<sup>83–85</sup> These results generally indicate that the accuracy of the spin-free MR-DSRG methods follows the order: DSRG-MRPT2 < DSRG-MRPT3 < sq-MR-LDSRG(2)  $\lesssim$  MR-LDSRG(2). The DSRG-MRPT3, sq-MR-LDSRG(2), and MR-LDSRG(2) methods yield bond distances that are comparable in accuracy with those from CCSD(T), while dissociation energies predicted with the DSRG methods are slightly superior in accuracy.

#### D. Excited states and spin-adapted implementation

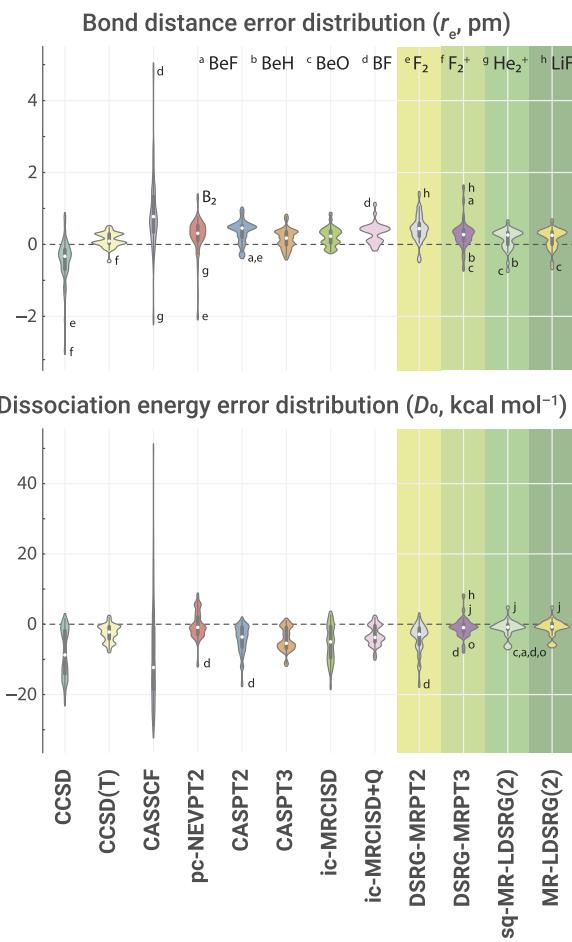
The DSRG formalism presented thus far corresponds to a state-specific approach with amplitudes optimized for one state at a time. FORTE also implements a state-averaged (or ensemble) generalization of the MR-DSRG that optimizes transformed Hamiltonian for multiple electronic states at a time.<sup>86</sup> This generalization is useful to (1) describe multiple electronic states and (2) to produce open-shell states of the correct spin symmetry. In the state-averaged MR-DSRG approach, operator normal ordering with respect to a single state is replaced by a more general normal ordering with respect to a density matrix ( $\hat{\rho}$ ) of all the target states,<sup>62,87,88</sup>

$$\hat{\rho} = \sum_{\alpha} \omega_{\alpha} |\Psi_{\alpha}\rangle\langle\Psi_{\alpha}|. \quad (23)$$

In this generalization for ensembles, a normal-ordered operator satisfies the condition  $\langle\{\hat{O}\}\rangle_{\hat{\rho}} \equiv \text{Tr}(\hat{\rho}\langle\hat{O}\rangle) = 0$ . The only practical difference between the state-specific and state-averaged versions of the MR-DSRG is that in the latter, all the reduced density matrices are replaced by averaged counterparts,

$$\hat{\gamma}_{rs\cdots}^{pq\cdots} = \sum_{\alpha=1}^n \omega_{\alpha} [\gamma_{\alpha}]_{rs\cdots}^{pq\cdots}. \quad (24)$$

When state-averaging is performed on a multiplet of spin states,<sup>56</sup> it is possible to implement spin-free versions of the MR-DSRG equations that have a smaller computational prefactor and reduced memory requirements. The spin-free MR-DSRG approach produces a transformed Hamiltonian that yields degenerate spin multiplet energies consistent with the state-specific approach and is applicable to states with arbitrary spin quantum numbers.



**FIG. 2.** Error distributions for the spectroscopic constants of 33 diatomic molecules. Each violin plot depicts the median (white dot), the interquartile range (thick bar in the center), the upper and lower adjacent values (line in the center), and the probability distribution (width). Molecules with errors lying outside three halves of the interquartile range are labeled. The cc-pCVQZ basis set was employed for Li and Be, while the cc-pVQZ basis set was used for all other atoms. The figure is adapted from Ref. 56.

#### E. Energy gradients and properties

Analytic energy gradients with respect to the nuclear coordinates and response properties are implemented for the unrelaxed DSRG-MRPT2 method.<sup>100</sup> These can be evaluated either with conventional two-electron integrals or density-fitted ones. The analytic energy gradients are derived using the method of Lagrange multipliers, where the constraints include the DSRG-MRPT2 energy and amplitude equations augmented with an additional set of conditions that enforce the use of semicanonical CASSCF orbitals. Because the DSRG-MRPT2 energy is not stationary, neither with respect to orbitals nor the CI coefficients, a linear system of equations that couple orbital and CI coefficients must be solved to account for relaxation effects. We note that the analytic energy gradients of SA-DSRG-PT2 have been derived by Park<sup>101</sup> and an implementation is available in the software package BAGEL.<sup>102</sup>

## F. Embedding

When dealing with large chemical systems where the properties or chemical changes of interest are spatially localized, computations may be accelerated by embedding a costly high-level computation within a low-level method.<sup>103–109</sup> FORTE implements active space embedding theory (ASET),<sup>94,99</sup> an embedding approach that uses as a starting point an MCSCF state for the entire system. In ASET, the system under consideration is partitioned into high- and low-level fragments, referred to as A and B here. The core and virtual orbitals are then rotated to maximize their overlap with the atomic orbitals centered onto these two fragments, following an approach used in the density-matrix embedding theory.<sup>110</sup> In contrast, active orbitals are left unchanged to preserve the energy of the starting MCSCF state. The partitioning of the orbitals in ASET is controlled by an AO overlap threshold  $\tau$  and by the projector  $\hat{P}^A = \sum_{\chi_\mu, \chi_\nu \in A} |\chi_\mu\rangle \langle \chi_\nu| (S_{\mu\nu}^A)^{-1} \langle \chi_\nu|$ , where the basis functions  $\chi_\mu$  and  $\chi_\nu$  are centered on fragment A and  $S_{\mu\nu}^A = \langle \chi_\mu | \chi_\nu \rangle$  is the AO overlap matrix. The overlap of an orbital  $\phi_p'$  with the fragment A is measured by the norm of the projected orbital, that is,  $n_p^A = \langle \hat{P}^A \phi_p' | \hat{P}^A \phi_p' \rangle$ . Rotated orbitals whose value of  $n_p^A$  is greater than  $\tau$  are assigned to A, otherwise to B. Alternatively, the user can select a fixed number of fragment orbitals from those with the largest overlap with fragment A. FORTE implements the lowest level of ASET, termed ASET(mf), which assumes that the total wave function for the entire system ( $\Psi_{A+B}$ ) is decomposable in terms of a correlated wave function for A ( $\Psi_A$ ) and a mean-field state for B ( $\Phi_B$ , a Slater determinant),

$$\Psi_{A+B} \approx \Psi_A \otimes \Phi_B. \quad (25)$$

This approach treats the interaction between the high- and low-level fragments of the system at the mean-field level, capturing electrostatic and exchange effects. The ASET(mf) Hamiltonian used in the high-level computation is given by

$$\hat{G} = E_B + \sum_{pq}^A \hat{h}_p^q \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4} \sum_{pqrs}^A v_{pq}^{rs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r, \quad (26)$$

where  $E_B$  is the energy contribution from the occupied orbitals of the low-level part, while  $\hat{h}_p^q$  are effective one-electron integrals that

account for the interaction of the electrons in A with the electrons in B, where the latter is the mean-field state. The generation of the fragment-localized orbitals and the ASET(mf) Hamiltonian corresponds to an orbital transformation and subsequent transformation of the integrals (dropping the occupied and unoccupied orbitals of B). The ASET(mf) energy is given by the energy of the entire system (A + B) computed at the MCSCF level ( $E_{\text{MCSCF}}^{A+B}$ ) plus the correlation energy of A from a high-level multireference computation on the ASET(mf) Hamiltonian ( $\delta E_{\text{MR}}^A$ ),

$$E_{\text{ASET}(\text{mf})} = E_{\text{MCSCF}}^{A+B} + \delta E_{\text{MR}}^A. \quad (27)$$

Since the cost of ASET(mf) is determined by the projection step, which scales as  $N^3$ , its cost is negligible compared to that of the MCSCF step.

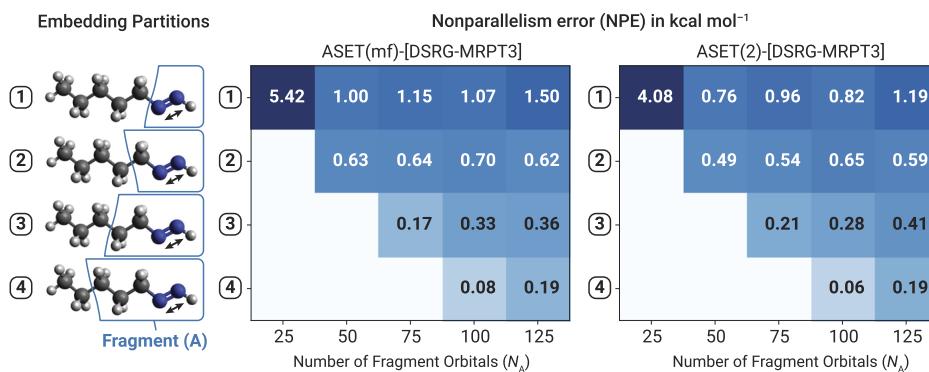
ASET can be systematically improved, for example, by treating interactions among fragments to second order, like in the ASET(2) method.<sup>99</sup> In practice, the increase in accuracy achieved by ASET(2) does not usually justify the extra cost necessary to go beyond a mean-field treatment. This point is shown in Fig. 3, where we report the nonparallelism error (NPE)—a measure of the overall deviation of a potential energy surface from a reference one—for the double N = N bond dissociation in pentyldiazene, a test case introduced in Ref. 111. For this example, the high-level computation uses DSRG-MRPT3, while the embedding Hamiltonian is generated with ASET(mf) and ASET(2). This plot shows that ASET(2) brings marginal improvements over ASET(mf). These results also suggest that given a computational budget expressed as the maximum number of fragment orbitals, then the most accurate results are obtained by choosing the largest fragment compatible with such a restriction.

## G. Orbital transformations

FORTE implements several types of orbital transformations useful in multireference computations. These are implemented using a general interface for performing orthogonal transformations of orbitals  $\{\psi_q\}$  into a new set of orbitals  $\{\psi'_p\}$  of the form

$$|\psi'_p\rangle = \sum_q |\psi_q\rangle Q_{qp}, \quad (28)$$

where Q is an orthogonal matrix ( $Q^{-1} = Q^T$ ).



**FIG. 3.** Comparison of the ASET(mf) and ASET(2) methods. Nonparallelism error (NPE) for the pentyldiazene ( $\text{CH}_3(\text{CH}_2)_4-\text{N}=\text{NH}$ ) bond dissociation curve computed with DSRG-MRPT3 using different numbers of orbitals in A and different fragment definitions. All the computations here use DSRG flow parameter  $s = 0.5 E_{\text{h}}^{-2}$  and the cc-pVQZ basis set. The figure is adapted from Ref. 99.

One type of orbital transformation useful when performing multireference computations is the generation of initial guess orbitals with a well-defined valence character. This transformation can help improve the convergence of MCSCF computations and avoid the problem of multiple solutions.<sup>112–114</sup> FORTE implements the atomic valence active space (AVAS) method,<sup>97</sup> which projects a starting set of orbitals—from Hartree–Fock or density functional theory—onto a set of atomic-like valence orbitals. This projection is analogous to the one used in ASET, except that the orbitals can be specified by the user via a much more general criterion. FORTE implements a general syntax to let the user specify the target AVAS orbitals. This allows the flexibility to select orbitals using the element symbol, atom number, orbital shell, or even a specific atomic-like orbital. For example, the following input selects the 3d shell of all Fe atoms and the 2p<sub>z</sub> orbital of all C atoms:

```
["Fe(3d)", "C(2pz)"]
```

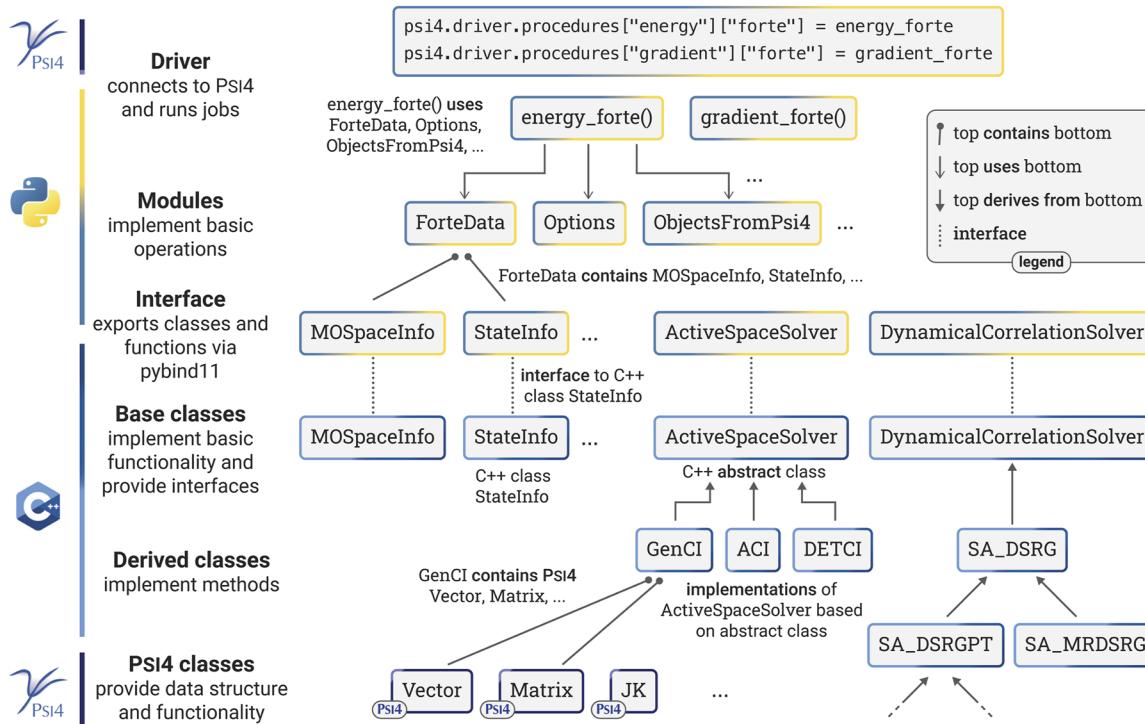
The FORTE implementation of AVAS also includes a more recent development in AVAS<sup>97,115</sup> that allows the selection of  $\pi$  orbitals by specifying a local orthogonal plane to each local p orbital.

More recently, frozen natural orbitals (FNOs) truncated schemes<sup>70,116,117</sup> for multireference computations based on second-order perturbation theory were implemented in FORTE.<sup>98</sup> This technique has been used to truncate the size of the virtual orbital space, reducing the computational cost of methods that scale steeply with

respect to the number of virtual orbitals. In FORTE, we first compute the second-order unrelaxed one-RDM at the DSRG-MRPT2 level and then rotate the virtual orbitals to make the virtual block diagonal. Once transformed to this basis, the orbitals are sorted according to their natural occupation number, and the virtual frozen orbitals are selected using a truncation scheme based on cumulative occupation numbers. The retained virtual orbitals are then transformed to the semicanonical basis before subsequent higher-level computations are performed. The FNO scheme implemented in FORTE even allows the user to specify two different values of the flow parameter, one for the DSRG-MRPT2 and one for the subsequent high-level computation.

### III. OVERVIEW OF THE STRUCTURE OF FORTE

An overview of the structure of FORTE is shown in Fig. 4. The main philosophy behind the design of FORTE is to create a library of highly performant functions written using a low-level language, which is exposed to a higher-level language to provide a flexible way to express general computational workflows. Hence, FORTE consists of a collection of C++ classes and methods, a Python interface that exports the C++ functionalities based on PYBIND11, and Python code that implements several computational workflows using a modular class system. FORTE relies on PSI4 in several ways. Many of the C++ methods in FORTE use classes and functions



**FIG. 4.** Structure of FORTE. The C++ code consists of base classes that provide basic functionality (e.g., managing orbital symmetry) and define interfaces to complex functionality via abstract classes. The derived classes implement active space solvers and dynamical correlation methods. Both the base and derived classes contain classes and use functions implemented in PSI4. A PYBIND11 interface exposes the base C++ classes to Python. A higher-level Python layer uses the base classes to implement a modular class system that executes computations. A Python driver layer connects FORTE to PSI4.

provided by Psi4, including vectors, matrices, and JK builders. The computational workflow of FORTE is implemented in Python to leverage Psi4's driver and to delegate many aspects of a computation (e.g., basis set handling, initial generation of mean-field orbitals, and geometry optimization).

### A. Base classes

The FORTE C++ code is organized around a few base classes used to implement data containers and interfaces for classes that implement computational procedures. The base classes implemented in FORTE are presented in Table II. These consist of classes that store information about the computation and the target electronic states (`ForteOptions`, `MOSpaceInfo`, `StateInfo`, and `RDMs`) and abstract interfaces for computational methods (`ActiveSpaceSolver` and `DynamicCorrelationSolver`).

The `MOSpaceInfo` class is designed to facilitate the handling of symmetry in FORTE, centralizing and simplifying book-keeping operations. Throughout FORTE, orbitals are assumed to be ordered in the "Pitzer" format, that is, orbitals are grouped according to their irreducible representation (irrep) and, within an irrep, are ordered according to their energy.

Classes that implement active space solvers are derived from the abstract `ActiveSpaceMethod` class, which has the scope of computing the energy of a manifold of states of the same symmetry (number of electrons,  $M_S$ , S, and point group) and restrictions on the occupation of each active space. The `ActiveSpaceSolver` class is then responsible for creating instances of `ActiveSpaceMethod` for each different target state. This allows FORTE significant flexibility in generating multiple reference states simultaneously, together with their corresponding averaged reduced density matrices. This feature enables, for example, state-averaged computations of electronic states of different point group symmetry and spin. The possibility of averaging over states with different orbital restrictions also enables specialized computations on core-excited states, where occupation restrictions on core orbitals may be used to target specific transitions.<sup>89,118</sup> Similarly, the `DynamicCorrelationSolver` class provides a standard interface for dynamical electron correlation methods. All the DSRG methods are derived from this

class, and sub-classes are defined to implement different levels of theory.

### B. Integral classes

The base class `ForteIntegrals` provides a standard interface for computing and accessing integrals in the molecular orbital basis and handles removing doubly occupied and empty orbitals that are excluded from treatments of the correlation energy. `ForteIntegrals` exposes standard one-electron integrals,

$$h_{pq} = \langle \phi_p | \hat{h} | \phi_q \rangle, \quad (29)$$

and antisymmetrized two-electron integrals,

$$\langle pq \| rs \rangle = \langle \phi_p \phi_q | \frac{1}{r_{12}} | \phi_r \phi_s \rangle - \langle \phi_p \phi_q | \frac{1}{r_{12}} | \phi_s \phi_r \rangle. \quad (30)$$

The `ForteIntegrals` class implements several integral types available from Psi4. FORTE implements conventional two-electron integrals and approximate two-electron integrals in the Cholesky<sup>119–122</sup> and density-fitted (DF) form,<sup>123–128</sup> whereby each element is given by the contraction of 2 three-index tensors ( $L_{pr}^P$ ) over an auxiliary index  $P$ ,

$$\langle pq | rs \rangle \approx \sum_p^K L_{pr}^P L_{qs}^P. \quad (31)$$

For density-fitted integrals, FORTE implements both in-core and disk integral classes, enabling routine applications to systems with 2000–3000 basis functions on a typical single-CPU node. Furthermore, the `CustomIntegrals` class handles user-provided integrals that may be read from an external file (e.g., FCIDUMP),<sup>129</sup> from PySCF,<sup>10</sup> or can be passed via the Python interface.

In addition, `ForteIntegrals` provides methods to evaluate generalized Fock matrices of the form

$$f_{pq} = h_{pq} + \sum_{rs} \langle pr \| qs \rangle \gamma_s^r, \quad (32)$$

which are used throughout the code by dynamical correlation solvers and orbital optimizers. The `ForteIntegrals` class also automates

TABLE II. Base classes implemented in FORTE.

Class name	Description
<code>ForteOptions</code>	User-specified options
<code>MOSpaceInfo</code>	Store size and symmetry information of orbitals spaces
<code>StateInfo</code>	Store information about a target state (electron number, symmetry, and spin)
<code>RDMs</code>	Store reduced density matrices and cumulants
<code>ActiveSpaceSolver</code>	Manages active space solver computations on multiple target states
<code>ActiveSpaceMethod</code>	Base class for active space solvers [FCI, GASCI, selected CI, and DMRG]
<code>DynamicCorrelationSolver</code>	Base class for dynamical correlation solvers [DSRG-MRPT2/3 and LDSRG(2)]

updating the molecular orbital coefficients (e.g., during orbital optimization) and the corresponding transformation of the one- and two-electron integrals.

### C. Support for manipulating determinants, states, and operators

The Determinant class in FORTE represents Slater determinants using the occupation number representation and finds use in the CI codes. This class uses a binary representation of the determinant based on an array of 64-bit unsigned integers, with each bit representing the occupation number of a spin orbital. For an array of  $k$  64-bit unsigned integers, this class stores the occupation numbers in the following way:

$$\underbrace{\left( \underbrace{n_{63}^1 \cdots n_0^1}_{\text{64-bit integer}}, \dots, \underbrace{n_{63}^k \cdots n_0^k}_{\text{64-bit integer}} \right)}_{\text{array of 64-bit integers}} = \underbrace{\left| n_0^1 \cdots n_{63}^1 \cdots n_0^k \cdots n_{63}^k \right\rangle}_{\text{element of Fock-space basis}}, \quad (33)$$

where the occupation number of a spin orbital  $\psi_p$  is such that  $n_p \in \{0, 1\}$ . Operations on determinants (getting/setting bits, masking, and partial sums of bits) are implemented using efficient bitwise operations.

FORTE also implements functions to manipulate general states and operators, a feature that enables the rapid implementation of new many-body theories. In FORTE, this functionality is supported by two classes: `SparseState` and `SparseOperator`. A `SparseState` object can represent a general state expressed as a linear combination of determinants that may span the entire Fock space associated with the one-electron basis,

$$|\Psi\rangle = \sum_{\mu \in \mathcal{S}} c_{\mu} |\mathbf{n}_{\mu}\rangle, \quad (34)$$

where the index  $\mu$  labels the elements of a general set  $\mathcal{S}$  of occupation number vectors  $|\mathbf{n}_{\mu}\rangle$ . For example, the superposition of determinants (using a notation that reflects the occupation of a spatial orbital),

$$|\Psi\rangle = \frac{1}{\sqrt{2}} (|20\rangle + |00\rangle), \quad (35)$$

can be produced in FORTE via the following Python code:

```
1 from forte import SparseState, det
2 c = 1./ math.sqrt(2.0)
3 psi = SparseState({ det('20') : c,
4                      det('00') : c})
```

This code relies on the utility function `det` to create `Determinant` objects and insert them in a dictionary passed to the constructor of the `SparseState` class.

Similarly, the class `SparseOperator` can manipulate general second quantized operators of the form

$$\hat{T} = \sum_{\mu} t_{\mu} \hat{\tau}_{\mu}, \quad (36)$$

where  $t_{\mu}$  is a scalar and  $\hat{\tau}_{\mu}$  is a product (string) of second quantized operators. The symbol  $\mu$  denotes a multi-index that collects all the creation and annihilation operators. Due to efficiency considerations, an operator string  $\hat{\tau}_{\mu}$  is stored in the following canonical form:

$$\hat{\tau}_{\mu} = \underbrace{\hat{a}_{p_1\alpha}^{\dagger} \hat{a}_{p_2\alpha}^{\dagger} \cdots}_{\alpha \text{ creation}} \underbrace{\hat{a}_{q_1\beta}^{\dagger} \hat{a}_{q_2\beta}^{\dagger} \cdots}_{\beta \text{ creation}} \underbrace{\cdots \hat{a}_{s_2\beta}^{\dagger} \hat{a}_{s_1\beta}^{\dagger} \cdots}_{\beta \text{ annihilation}} \underbrace{\hat{a}_{r_2\beta}^{\dagger} \hat{a}_{r_1\beta}^{\dagger}}_{\alpha \text{ annihilation}}, \quad (37)$$

where the indices are assumed to be sorted in ascending order, e.g.,  $p_i < p_{i+1}$ . Internally, FORTE represents a general string of second quantized operators in the `SQOperatorString` class, which holds two sets of  $k$  64-bit unsigned integers, with each bit representing the presence or absence of the corresponding creation or annihilation operator. General operators may be easily defined by creating a `SparseOperator` object and adding terms, as shown in the following example:

```
1 from forte import SparseOperator
2 op = SparseOperator()
3 op.add_term_from_str(' [1a+ 2b+ 0b- 0a-] ', 1.0)
4 op.add_term_from_str(' [2a+ 1b+ 0b- 0a-] ', -1.0)
```

This code produces the following operator:

$$\text{op} = \hat{a}_{1\alpha}^{\dagger} \hat{a}_{2\beta}^{\dagger} \hat{a}_{0\beta} \hat{a}_{0\alpha} - \hat{a}_{2\alpha}^{\dagger} \hat{a}_{1\beta}^{\dagger} \hat{a}_{0\beta} \hat{a}_{0\alpha}. \quad (38)$$

Functionality is also implemented to apply an arbitrary `SparseOperator` object  $\hat{T}$  to a general `SparseState` object  $|\Psi\rangle$ ,

$$|\Psi'\rangle \leftarrow \hat{T} |\Psi\rangle. \quad (39)$$

In addition, FORTE implements functionality to apply the exponential of  $\hat{T}$ ,

$$|\Psi'\rangle \leftarrow e^{\hat{T}} |\Psi\rangle, \quad (40)$$

via the corresponding Taylor series. To support the realization of methods relevant to quantum computing, FORTE also implements functionality to evaluate the product of exponentials of single *nilpotent* Fermionic operators (such that  $\hat{\tau}_{\mu}^2 = 0$ ),

$$|\Psi'\rangle \leftarrow \prod_{\mu} e^{t_{\mu} \hat{\tau}_{\mu}} |\Psi\rangle = \prod_{\mu} (1 + t_{\mu} \hat{\tau}_{\mu}) |\Psi\rangle. \quad (41)$$

When the operator strings entering  $\hat{T}$  do not commute ( $[\hat{\tau}_{\mu}, \hat{\tau}_{\nu}] \neq 0$  for some  $\mu$  and  $\nu$ ), different orderings of the exponential factors lead to different states. Therefore, FORTE uses a specialized class (`SparseOperatorList`) to store and manipulate lists of operators ( $t_1 \hat{\tau}_1, t_2 \hat{\tau}_2, \dots$ ), preserving the order of the operators and allowing for repetitions. For each function described above, a corresponding version that applies the anti-Hermitian operator  $\hat{T} - \hat{T}^{\dagger}$  is also available to support the implementation of unitary formalisms.



**FIG. 5.** Mid-layer Python modules used in FORTE. (a) The ForteData class stores Psi4 and FORTE base objects that are created and modified during a computation. (b) Example Python code used in FORTE that creates a ForteData object and uses the OptionsFactory, ObjectsFromPsi4, and ActiveSpaceIntegrals modules. (c) Illustration of the execution flow and how the modules change the ForteData object from being unallocated (light gray) to allocated (dark gray).

#### D. Python modules

The next level of abstraction in FORTE consists of a set of mid-level Python modules that provide core functionality, such as setting up FORTE objects, performing aspects of a computation, or simplifying access to the Python API. This layer makes extensive use of the PYBIND11 interface to C++ types and classes, and it is easily extended. Some of the features of this layer are shown in Fig. 5.

Modules define a standard interface for creating steps in a computation and for passing the input and storing the output of a computation. To standardize their behavior, modules are derived from a base abstract class (Module). Modules provide functionality but do not store the input and output data of a computation. FORTE uses the ForteData class to store all the Psi4 and FORTE objects involved in a computation. ForteData is implemented as a Python dataclass, and after initialization, it is empty.

Using modules requires performing two steps:

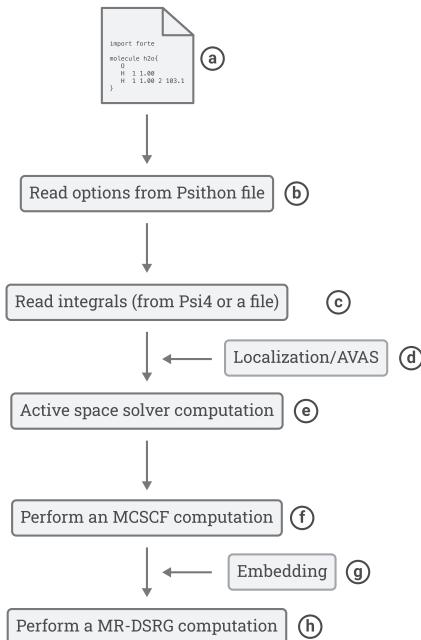
1. **Module initialization.** In the first step, the module object is created, and parameters are passed to control the behavior of the module:

```
module = Module(args)
```

2. **Module execution.** In the next step, the member function run is called passing the ForteData object

```
data = module.run(data)
```

For example, the module ObjectsFromPsi4 initializes the objects used by FORTE from Psi4 and is invoked in the FORTE drivers as



**FIG. 6.** Standard FORTE workflow. Starting from an input Python file (a), the options controlling the computation are read (b). In the next step, integrals are read from Psi4 or an external file (c). Optionally, the orbitals are transformed to a localized or AVAS basis (d). The main computation consists of an initial active space solver computation (e), followed by simultaneous orbital and CI coefficient optimization (f) and a final MR-DSRG computation (g). If the user requests an ASET(mf) embedding computation, the orbitals are localized prior to performing the MR-DSRG computation (g).

```
data = ObjectsFromPsi4(**kwargs).run(data)
```

The keyword arguments (`kwargs`) variable is passed in the constructor of this object to propagate arguments passed by those used in the `Psi4` function `psi4.energy()`.

## E. Running FORTE: Standard workflow and Python interface

The simplest way to run FORTE computations is via its interface to `Psi4`. This requires the user to write an input file, which

**LISTING 1.** A sample FORTE input file for a spin-adapted DSRG-MRPT2 computation on the HF molecule using the cc-pVDZ basis set. Lines 6–11 define the molecular geometry and basis set. Lines 13–21 set the options for FORTE. The `frozen_docc`, `restricted_docc`, and `active` options specify the number of frozen doubly occupied, the restricted doubly occupied, and active orbitals, respectively, in each irreducible representation of the  $C_{2v}$  point group ( $A_1, A_2, B_1, B_2$ ).

```
1 # Spin-adapted DSRG-MRPT2 computation on HF using
2 # a CASSCF(2,2) reference state
3
4 import forte
5
6 molecule {
7   F
8   H 1 1.5
9 }
10
11 set basis cc-pvdz
12
13 set forte {
14   # calculation options
15   active_space_solver fci
16   correlation_solver sa-mrdsrg
17   corr_level pt2
18   dsrg_s 1.0
19   # orbital subspace dimensions (A1,A2,B1,B2)
20   frozen_docc [1,0,0,0]
21   restricted_docc [1,0,1,1]
22   active [2,0,0,0]
23 }
24
25 energy('forte')
```

is then executed by calling `psi4` on the command line. When executed in this way, FORTE follows a standard workflow for multireference computations shown in Fig. 6. The main steps in this workflow are an active space solver computation, followed by orbital optimization (MCSCF) and an MR-DSRG computation. The user can optionally transform the initial guess orbitals prior to the MCSCF computation or enable an embedded computation in the MR-DSRG step. FORTE also supports external active space solvers, either through direct integration on the C++ side or via the general `ExternalActiveSpaceMethod` class. This feature has enabled new types of computations, such as the ones reported in Ref. 130, where the active space solver was a hardware quantum computation based on the variational quantum eigensolver.<sup>131</sup>

This standard workflow encompasses a variety of user case scenarios, allowing any combination of integrals, active space solvers, and dynamical correlation solvers. This is possible due to FORTE's extensive use of abstract interfaces and polymorphism, which enables using a single workflow code that is easy to maintain. Within this workflow, a computational procedure is expressed in terms of

**LISTING 2.** A sample Python script to run an FCI computation using FORTE's API. Lines 6–9 define the molecular geometry, while Lines 11–17 define the options of this computation. FORTE-specific options are prefixed with "forte\_". We skip the MCSCF step by setting the `mcscf_reference` option to false. Line 19 runs the standard FORTE workflow.

```
1 # FCI computation on Li2 using RHF orbitals
2
3 import psi4
4 import forte
5
6 mol = psi4.geometry("""
7 Li
8 Li 1 1.6
9 """
10
11 psi4.set_options(
12   {
13     'basis': 'cc-pVDZ',
14     'scf_type': 'pk',
15     'forte__active_space_solver': 'fci',
16     'forte__mcscf_reference' : False
17   }
18 )
19
20 psi4.energy('forte', molecule=mol)
```

**LISTING 3.** A sample FORTE input file for a spin-integrated DSRG-MRPT2 computation on the HF molecule using its interface for PySCF. Lines 5–14 run a PySCF CASSCF computation. Lines 16–33 set the options for FORTE. The `frozen_docc`, `restricted_docc`, and `active` options specify the number of frozen doubly occupied, the restricted doubly occupied, and active orbitals, respectively, in each irreducible representation of the  $C_{2v}$  point group ( $A_1, A_2, B_1, B_2$ ).

```
1 import forte
2 from pyscf import gto, scf, mcscf
3
4 # run a CASSCF(2,2) computation with pyscf
5 molecule = """
6     F
7     H 1 1.5
8 """
9 mol = gto.M(atom = molecule, basis = 'cc-pvdz', symmetry = 'c2v')
10 mf = scf.RHF(mol)
11 mf.kernel()
12 mc = mcscf.CASSCF(mf, ncas=2, nelecas=(1,1))
13 mc.conv_tol = 1e-12
14 mc.mc2step()
15
16 molecule {
17     F
18     H 1 1.5
19 }
20
21 set forte {
22     # calculation options
23     int_type          pyscf
24     active_space_solver fci
25     correlation_solver  dsrg-mrpt2
26     dsrg_s            1.0
27     # orbital subspace dimensions (A1,A2,B1,B2)
28     frozen_docc        [1,0,0,0]
29     restricted_docc    [1,0,1,1]
30     active             [2,0,0,0]
31     # skip MCSCF
32     mcscf_reference    false
33 }
34
35 energy('forte', pyscf_obj = mc)
```

base classes, and the various possible combinations of methods are obtained by instantiating such base classes with derived classes, as specified in the input.

The input for a FORTE computation using this standard workflow relies on Psithon, a Python syntax customized for Psi4 that simplifies certain aspects of specifying the input to a computation. In Listing 1, we show the input file for a basic state-specific DSRG-MRPT2 computation on the hydrogen fluoride molecule using a CASSCF reference state. The main options specified in this input consist of the type of active space (FCI = fci) and the type of correlation solver (spin-adapted MR-DSRG = sa-mrdsrg, truncated to second-order = pt2). The spin-adapted implementation of the DSRG-MRPT2 is the most computationally efficient and generally recommended for standard computations. In this example, the  $1a_1$  orbital of HF is optimized in the MCSCF procedure, but it is dropped from the correlated MR-DSRG computation.

An alternative way to run FORTE is through the Python API directly, as shown in Listing 2. This example illustrates how the geometry and options may be defined via the Psi4 Python API and FORTE executed by directly calling the `psi4.energy` function. In the future, we plan to develop code that will enable to run FORTE using its Python module infrastructure to enable the user to specify more general workflows than the standard one currently implemented.

FORTE can also be used through its PYSCF interface, as shown in Listing 3. This example illustrates how the integrals may be generated via PYSCF and used by using FORTE by specifying `int_type pyscf` and defining a `pyscf_obj`.

## F. Distribution, documentation, tutorials, and external tools

The recommended way for users to obtain FORTE is by downloading prepackaged binaries made available via [conda-forge.org](https://conda-forge.org), a community-driven repository of conda packages. After downloading conda, installation of the latest version of FORTE and its dependencies can be performed via the command

```
conda install -c conda-forge forte
```

FORTE is an open-source project, and its codes are distributed for free via GitHub (version 0.4.1 is available at <https://github.com/evangelistalab/forte>) through an LGPL-3.0 license. Compilation of FORTE requires Psi4 (version 1.9.1 or later, available at <https://github.com/psi4/psi4>) and the tensor library AMBIT (version 0.7.1 or later, available at <https://github.com/jturney/ambit>).

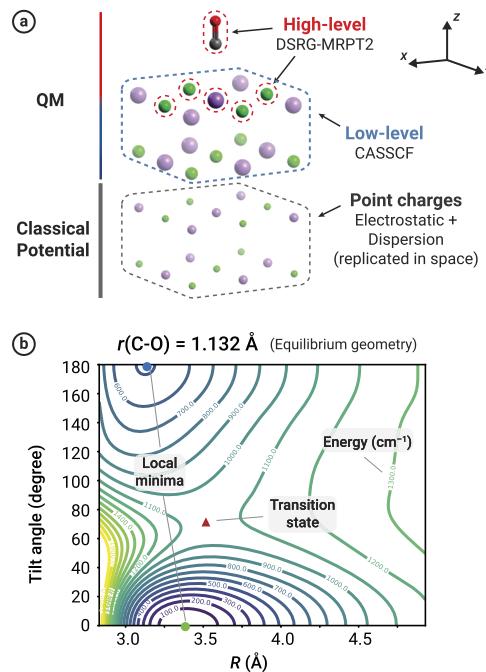
Compilation instructions for FORTE are available at the GitHub site, while general documentation is available online through the ReadtheDocs website (<https://forte.readthedocs.io>). Supplementing the FORTE documentation is a suite of more than 250 test cases for the methods implemented that demonstrate how to run various types of computations and the relevant user options. The FORTE repository includes a set of tutorials that introduce users to some of the features of the software. These tutorials cover basic aspects (e.g., running FORTE via Jupyter and implementing CCSD using the sparse operator infrastructure), useful functionality (e.g., orbital visualization), and how to implement methods using quantities computed in FORTE.

In parallel to the development of FORTE, we have created the Python package `fortecubeview` (<https://github.com/evangelistalab/fortecubeview>, installable via pip), which may be used to display molecular orbitals and normal modes computed with FORTE and Psi4. Another related development is the QFORTE package,<sup>132</sup> which implements a quantum computer emulator and a library of quantum algorithms and can consume integrals generated by FORTE.

## IV. EXAMPLE APPLICATIONS OF FORTE

### A. Modeling molecules in complex chemical environments

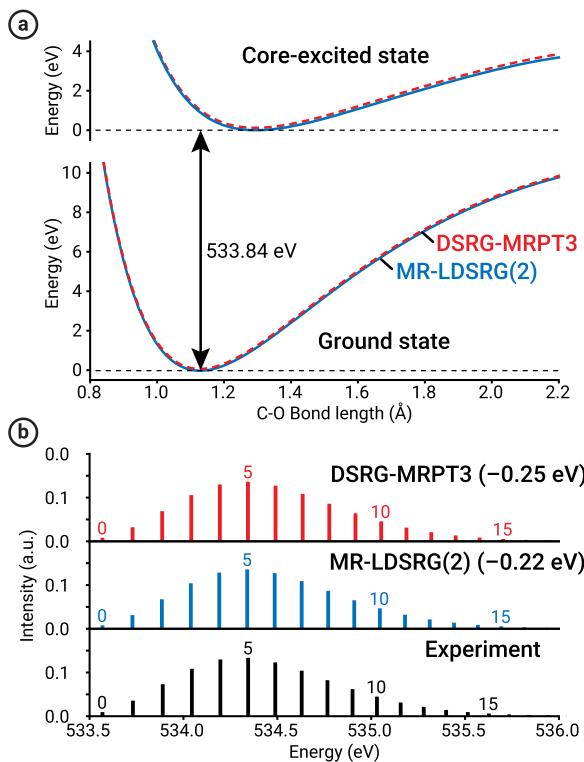
An example of how the tools implemented in FORTE may be combined to enable new types of computations is given by our recent multireference study of physisorbed molecules.<sup>133</sup> This work focused on the recent experimental observation of the isomerization of vibrationally excited CO molecules on an NaCl(100) surface.<sup>134</sup> To accurately describe the highly vibrationally excited



states of CO, we performed multireference computations that can correctly describe the ground state of CO from the equilibrium to the dissociated limit. As shown in Fig. 7, we modeled this system by computing the potential energy surface of a single CO molecule on NaCl(100). The starting point of this computation was a CASSCF computation on the cluster [CO-Na<sub>9</sub>Cl<sub>9</sub>] using AVAS guess orbitals that span the full valence space of CO. This CASSCF computation included a classical external potential to account for electrostatic interactions between the cluster and the Na<sup>+</sup> and Cl<sup>-</sup> ions outside the cluster. ASET(mf) was then used to compute dynamical correlation energy corrections due to the fragment [CO-NaCl<sub>4</sub>]<sup>3-</sup> at the DSRG-MRPT2 level. Classical additive corrections were also included to account for dispersion interactions between CO and the Na<sup>+</sup> and Cl<sup>-</sup> ions. The initial CASSCF step was the main bottleneck in these computations. To map the potential energy surface of CO-NaCl(100), we performed 5376 single-point computations using 288 CPU cores, which ran in less than 50 h.

## B. Simulating x-ray absorption spectra

Another example of application enabled by FORTE is simulating x-ray absorption spectroscopy through our GAS-DSRG



**FIG. 8.** (a) Potential energy surface for the ground state and 1s  $\rightarrow \pi^*$  core-excited state of CO computed using GAS-DSRG and the cc-pCVQZ-DK basis. All the curves are shifted so that the values of the MR-LDSRG(2) potential energy in both traces at each minimum are zero. (b) Spectrum of CO simulated from the GAS-DSRG potentials and the experimentally determined Morse constants. The theoretical spectra are shifted to align the 0-0 transition with experiment. Each transition is labeled with the vibrational quantum number  $v'$  of the upper state. The figure is adapted from Ref. 89.

scheme.<sup>89</sup> This approach starts by describing core-excited states using GASSCF reference states, which capture the dominant orbital relaxation effects. Following this step, a multireference DSRG computation is used to account for dynamical electron correlation. An illustrative example of the potential application of this approach is our recent computation of the vibrationally resolved O K-edge 1s  $\rightarrow \pi^*$  transition of CO. In Fig. 8, we show the simulated spectrum with Franck-Condon factors calculated using

**LISTING 4.** Python function that uses FORTE's sparse classes to evaluate the coupled cluster energy and residual vector. The arguments of this function are a SparseOperator object ( $T$ ) that represents the operator  $\hat{T}$ , a SparseState object ( $\Phi$ ) that represents the reference state  $|\Phi\rangle$ , a SparseHamiltonian object ( $H$ ) that represents the Hamiltonian operator ( $\hat{H}$ ), and a Boolean variable ( $is\_antiherm$ ) to specify the type of similarity transformation requested.

```

1 def cc_residual(T, Phi, H, is_antiherm):
2     """Evaluate the CC energy and residual equations"""
3     exp = forte.SparseExp()
4
5     if is_antiherm:
6         # apply S = T - T^dagger
7         exp_S = exp.apply_antiherm
8     else:
9         # apply S = T
10        exp_S = exp.apply_op
11
12    # compute exp(S)|Phi>
13    wfn = exp_S(T, Phi)
14
15    # compute H exp(S)|Phi>
16    Hwfn = H.apply(wfn)
17
18    # compute exp(-S) H exp(S)|Phi>
19    R = exp_S(T, Hwfn, scaling_factor=-1.0)
20
21    # compute <Phi|tau^dagger_mu exp(-S) H exp(S)|Phi>
22    residual = forte.get_projection(T, Phi, R)
23
24    # compute <Phi|exp(-S) H exp(S)|Phi>
25    energy = 0.0
26
27    for det, coeff in Phi.items():
28        energy += coeff * R[det]
29
30    return (residual, energy)

```

one-dimensional potential energy surfaces of both ground and core-excited states obtained from GAS-DSRG and an experimentally determined Morse potential.<sup>135</sup> Our multireference computations accurately describe core-excited CO molecules up to the dissociation limit, as evidenced by the excellent agreement in the vibrational features of CO in the x-ray absorption spectrum and the small energy shifts (less than 0.25 eV) required to align the origin of the spectrum to the experimental value.

### C. Rapid prototyping of many-body methods with FORTE

Our second example application of FORTE focuses on rapidly prototyping many-body theories. The `SparseState` and `SparseOperator` objects allow the implementation of virtually any form of the many-body theory expressible in second quantization. To demonstrate this point, we show how these classes may be used to create a pilot implementation of *arbitrary-order* traditional and unitary coupled-cluster theory using a projective formalism.

At the core of these methods is the evaluation of the residual vector ( $r_\mu$ ) defined in terms of a reference determinant  $\Phi$  and a cluster operator  $\hat{S}$ . The definition of the operator  $\hat{S}$  depends on the type of similarity transformation:

$$\hat{S} = \begin{cases} \hat{T} & \text{traditional,} \\ \hat{T} - \hat{T}^\dagger & \text{unitary,} \end{cases} \quad (42)$$

where  $\hat{T}$  is a sum of particle-hole excitation operators  $\hat{t}_\mu$   $\equiv \hat{a}_a^\dagger \hat{a}_b^\dagger \cdots \hat{a}_j \hat{a}_i$  with corresponding amplitude  $t_\mu$  defined as

$$\hat{T} = \sum_\mu t_\mu \hat{t}_\mu. \quad (43)$$

Listing 4 shows that with less than 25 lines of code, FORTE can evaluate the residual equation

$$r_\mu = \langle \Phi | \hat{t}_\mu^\dagger e^{-\hat{S}} \hat{H} e^{\hat{S}} | \Phi \rangle = 0, \quad (44)$$

for both the traditional and unitary case using the functionality provided by the `SparseState` and `SparseOperator` Operator.

A more complex example application of this functionality can be found in Ref. 136 where benchmark projective quantum eigen-solver computations on  $H_{10}$  systems (corresponding to 20 qubits) were made possible by FORTE's sparse state and add operator classes.

## V. CURRENT LIMITATIONS AND FUTURE DEVELOPMENT PLANS

This paper summarizes the major developments in FORTE over the past decade, highlighting its extensive implementation of DSRG-based methods. In the future, we plan to improve several aspects of FORTE and add desirable functionality currently lacking in the program. In regards to the functionality supported by FORTE, several areas deserve improvement, including the implementation of gradients and static properties beyond state-specific DSRG methods and the implementation of nonadiabatic couplings. Another set of desirable features includes DSRG-based methods for determining a manifold of excited states. Our group is currently exploring how to formulate equation-of-motion methods

based on the DSRG, a preliminary step required to guide the development of production-level implementations of these new theories. Another set of functionalities under exploration are two- and four-component relativistic extensions of the DSRG methods.<sup>137</sup> While conceptually straightforward under the no-pair approximation, implementing these methods would require generalizing the code to handle complex tensors.

At the software level, a major plan is to make FORTE more interoperable with other quantum chemistry packages to leverage functionality not available in Psi4. This development will be facilitated by FORTE's Python layer, which allows interfacing it with other Python-based packages and writing adaptors to read the information required to perform a multireference computation (e.g., integrals and MO coefficients). Our group is also working on increasing interoperability, including creating a lightweight library for tensor storage that supports allocating and exporting tensors across C++ and Python without copying data and that supports both real and complex types. This effort aims to enable the use of existing optimized numerical libraries, such as NUMPY, for tensor operations.<sup>138</sup> Future work will fully integrate this new functionality in FORTE.

## VI. CONCLUSION

By leveraging the power of open-source quantum chemistry codes, the past ten years have seen FORTE develop from a simple plugin to Psi3 to a suite of advanced multireference electronic structure theories that have pushed the boundaries of quantum chemistry. We hope that FORTE will become a valuable asset to the quantum chemistry community, enabling researchers to perform advanced multireference computations and prototype new theories. Further collaborations and contributions from the community will be essential in refining FORTE's functionalities and expanding its reach to address open challenges in quantum chemistry.

## ACKNOWLEDGMENTS

The developers of FORTE are thankful for the various funding sources that have supported its development. These include the U.S. Department of Energy (under Awards Nos. DE-SC0016004, DE-SC0019374, and DE-SC0024532), the U.S. National Science Foundation (under Awards Nos. CHEM-1900532, CHEM-2038019, and CHEM-2312105), a Research Fellowship of the Alfred P. Sloan Foundation, a Camille Dreyfus Teacher-Scholar Award (No. TC-18-045), and start-up funds provided by Emory University.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

**Francesco A. Evangelista:** Funding acquisition (lead); Project administration (lead); Software (lead); Writing – original draft (lead); Writing – review & editing (lead). **Chenyang Li:** Software (lead); Writing – review & editing (equal). **Prakash Verma:** Software (equal). **Kevin P. Hannon:** Software (equal). **Jeffrey B. Schriber:**

Software (equal). **Tianyuan Zhang**: Software (equal). **Chenxi Cai**: Software (equal). **Shuhe Wang**: Software (equal). **Nan He**: Software (equal). **Nicholas H. Stair**: Software (equal). **Meng Huang**: Software (equal). **Renke Huang**: Software (equal). **Jonathon P. Misiewicz**: Software (equal). **Shuhang Li**: Software (supporting). **Kevin Marin**: Software (supporting). **Zijun Zhao**: Software (supporting). **Lori A. Burns**: Software (equal).

## DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## REFERENCES

- 1 C. D. Sherrill, *J. Chem. Phys.* **132**, 110902 (2010).
- 2 D. A. Matthews, L. Cheng, M. E. Harding, F. Lipparini, S. Stopkowicz, T.-C. Jagau, P. G. Szalay, J. Gauss, and J. F. Stanton, *J. Chem. Phys.* **152**, 214108 (2020).
- 3 S. D. Folkestad, E. F. Kjønstad, R. H. Myhre, J. H. Andersen, A. Balbi, S. Coriani, T. Giovannini, L. Goletto, T. S. Haugland, A. Hutcheson, I.-M. Höyvik, T. Moitra, A. C. Paul, M. Scavino, A. S. Skeidsvoll, A. H. Tveten, and H. Koch, *J. Chem. Phys.* **152**, 184103 (2020).
- 4 C. Peng, C. A. Lewis, X. Wang, M. C. Clement, K. Pierce, V. Rishi, F. Pavošević, S. Slattery, J. Zhang, N. Teke, A. Kumar, C. Masteran, A. Asadchev, J. A. Calvin, and E. F. Valeev, *J. Chem. Phys.* **153**, 044120 (2020).
- 5 F. Aquilante, J. Autschbach, A. Biaardi, S. Battaglia, V. A. Borin, L. F. Chibotaru, I. Conti, L. D. Vico, M. Delcey, I. F. Galván, N. Ferré, L. Freitag, M. Garavelli, X. Gong, S. Knecht, E. D. Larsson, R. Lindh, M. Lundberg, P. A. Malmqvist, A. Nenov, J. Norell, M. Odelius, M. Olivucci, T. B. Pedersen, L. Pedraza-González, Q. M. Phung, K. Pierloot, M. Reiher, I. Schapiro, J. Segarra-Martí, F. Segatta, L. Seijo, S. Sen, D.-C. Sergentu, C. J. Stein, L. Ungur, M. Vacher, A. Valentini, and V. Veryazov, *J. Chem. Phys.* **152**, 214117 (2020).
- 6 G. Li Manni, I. F. Galván, A. Alavi, F. Aleotti, F. Aquilante, J. Autschbach, D. Avagliano, A. Biaardi, J. J. Bao, S. Battaglia, L. Birnoschi, A. Blanco-González, S. I. Bokarev, R. Broer, R. Cacciari, P. B. Calio, R. K. Carlson, R. C. Couto, L. Cerdáñ, L. F. Chibotaru, N. F. Chilton, J. R. Church, I. Conti, S. Coriani, J. Cueellar-Zuquin, R. E. Daoud, N. Dattani, P. Decleva, C. de Graaf, M. G. Delcey, L. D. Vico, W. Dobrutz, S. S. Dong, R. Feng, N. Ferreé, M. Filatov(Gulak), L. Gagliardi, M. Garavelli, L. González, Y. Guan, M. Guo, M. R. Hennefarth, M. R. Hermes, C. E. Hoyer, M. Huix-Rotllant, V. K. Jaiswal, A. Kaiser, D. S. Kaliakin, M. Khamesian, D. S. King, V. Kochetov, M. Krosnicki, A. A. Kumaar, E. D. Larsson, S. Lehtola, M.-B. Lepetit, H. Lischka, P. L. Riéos, M. Lundberg, D. Ma, S. Mai, P. Marquetand, I. C. D. Merritt, F. Montorsi, M. Moürchen, A. Nenov, V. H. A. Nguyen, Y. Nishimoto, M. S. Oakley, M. Olivucci, M. Oppel, D. Padula, R. Pandharkar, Q. M. Phung, F. Plasser, G. Raggi, E. Rebolini, M. Reiher, I. Rivalta, D. Roca-Sanjuán, T. Romig, A. A. Safari, A. Saénchez-Mansilla, A. M. Sand, I. Schapiro, T. R. Scott, J. Segarra-Martí, F. Segatta, D.-C. Sergentu, P. Sharma, R. Shepard, Y. Shu, J. K. Staab, T. P. Straatsma, L. K. Sorensen, B. N. C. Tenorio, D. G. Truhlar, L. Ungur, M. Vacher, V. Veryazov, T. A. Voß, O. Weser, D. Wu, X. Yang, D. Yarkony, C. Zhou, J. P. Zobel, and R. Lindh, *J. Chem. Theory Comput.* **19**, 6933 (2023).
- 7 K. Guther, R. J. Anderson, N. S. Blunt, N. A. Bogdanov, D. Celand, N. Dattani, W. Dobrutz, K. Ghanem, P. Jeszenszki, N. Liebermann, G. Li Manni, A. Y. Lozovoi, H. Luo, D. Ma, F. Merz, C. Overy, M. Ramp, P. K. Samanta, L. R. Schwarz, J. J. Shepherd, S. D. Smart, E. Vitale, O. Weser, G. H. Booth, and A. Alavi, *J. Chem. Phys.* **153**, 034107 (2020).
- 8 E. Aprà, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. W. Aquino, R. Atta-Fynn, J. Autschbach, N. P. Bauman, J. C. Becca, D. E. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Boschen, J. Brabec, A. Bruner, E. Caué, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. J. O. Deegan, T. H. Dunning, M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Frücht, L. Gagliardi, J. Garza, N. Gawande, S. Ghosh, K. Glaesemann, A. W. Götz, J. Hammond, V. Helms, E. D. Hermes, K. Hirao, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jónsson, R. A. Kendall, M. Klemm, R. Kobayashi, V. Konkov, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, R. J. Littlefield, A. J. Logsdail, K. Lopata, W. Ma, A. V. Marenich, J. M. del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, T. Nakajima, D. R. Nascimento, J. A. Nichols, P. J. Nichols, J. Nieplocha, A. O. de-la Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, R. M. Richard, P. Sadayappan, G. C. Schatz, W. A. Shelton, D. W. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. S. Thomas, V. Tippuraju, D. G. Truhlar, K. Tsemekhman, T. V. Voorhis, A. Vázquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Wolinski, A. T. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison, *J. Chem. Phys.* **152**, 184102 (2020).
- 9 G. M. J. Barca, C. Bertoni, L. Carrington, D. Datta, N. D. Silva, J. E. Deustua, D. G. Fedorov, J. R. Gour, A. O. Gunina, E. Guidez, T. Harville, S. Irle, J. Ivanic, K. Kowalski, S. S. Leang, H. Li, W. Li, J. J. Lutz, I. Magoulas, J. Mato, V. Mironov, H. Nakata, B. Q. Pham, P. Piecuch, D. Poole, S. R. Pruitt, A. P. Rendell, L. B. Roskop, K. Ruedenberg, T. Sattasathuchana, M. W. Schmidt, J. Shen, L. Slipchenko, M. Sosonkina, V. Sundriyal, A. Tiwari, J. L. G. Vallejo, B. Westheimer, M. Włoch, P. Xu, F. Zahariev, and M. S. Gordon, *J. Chem. Phys.* **152**, 154102 (2020).
- 10 Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Erikson, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Kováč, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkkin, W. Purwanto, P. J. Robinson, E. Ronca, E. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whittfeld, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Sokolov, and G. K.-L. Chan, *J. Chem. Phys.* **153**, 024109 (2020).
- 11 S. Seritan, C. Bannwarth, B. S. Fales, E. G. Hohenstein, S. I. L. Kokkila-Schumacher, N. Luehr, J. W. Snyder, C. Song, A. V. Titov, I. S. Ufimtsev, and T. J. Martínez, *J. Chem. Phys.* **152**, 224110 (2020).
- 12 M. J. T. Oliveira, N. Papirer, Y. Pouillon, V. Blum, E. Artacho, D. Caliste, F. Corsetti, S. de Gironcoli, A. M. Elena, A. García, V. M. García-Suárez, L. Genovese, W. P. Huhn, G. Huhs, S. Kokott, E. Küçükbenli, A. H. Larsen, A. Lazzaro, I. V. Lebedeva, Y. Li, D. López-Durán, P. López-Tarifa, M. Lüders, M. A. L. Marques, J. Minar, S. Mohr, A. A. Mostofi, A. O'Cais, M. C. Payne, T. Ruh, D. G. A. Smith, J. M. Soler, D. A. Strubbe, N. Tancogne-Dejean, D. Tildesley, M. Torrent, and V. W. Yu, *J. Chem. Phys.* **153**, 024117 (2020).
- 13 T. Saue, R. Bast, A. S. P. Gomes, H. J. A. Jensen, L. Visscher, I. A. Aucar, R. D. Remigio, K. G. Dyall, E. Eliav, E. Fasshauer, T. Fleig, L. Halbert, E. D. Hedegård, B. Helmich-Paris, M. Iliaš, C. R. Jacob, S. Knecht, J. K. Laerdahl, M. L. Vidal, M. K. Nayak, M. Olejniczak, J. M. H. Olsen, M. Pernpointner, B. Senjean, A. Shee, A. Sunaga, and J. N. P. van Stralen, *J. Chem. Phys.* **152**, 204104 (2020).
- 14 H. Lischka, R. Shepard, T. Müller, P. G. Szalay, R. M. Pitzer, A. J. A. Aquino, M. M. A. do Nascimento, M. Barbatti, L. T. Belcher, J.-P. Blaudeau, I. Borges, S. R. Brozell, E. A. Carter, A. Das, G. Gidofalvi, L. González, W. L. Hase, G. Kedziora, M. Kertesz, F. Kososki, F. B. C. Machado, S. Matsika, S. A. do Monte, D. Nachtigalová, R. Nieman, M. Oppel, C. A. Parish, F. Plasser, R. F. K. Spada, E. A. Stahlberg, E. Ventura, D. R. Yarkony, and Z. Zhang, *J. Chem. Phys.* **152**, 134110 (2020).
- 15 H.-J. Werner, P. J. Knowles, F. R. Manby, J. A. Black, K. Doll, A. Heßelmann, D. Kats, A. Köhn, T. Korona, D. A. Kreplin, Q. Ma, T. F. Miller, A. Mitrushchenkov, K. A. Peterson, I. Polyak, G. Rauhut, and M. Sibaev, *J. Chem. Phys.* **152**, 144107 (2020).
- 16 M. Kállyay, P. R. Nagy, D. Mester, Z. Rolik, G. Samu, J. Csontos, J. Csóka, P. B. Szabó, L. Gyevi-Nagy, B. Hégly, I. Ladjánszki, L. Szegedy, B. Laďóczki, K. Petrov, M. Farkas, P. D. Mezei, and A. Ganyecz, *J. Chem. Phys.* **152**, 074107 (2020).
- 17 F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, *J. Chem. Phys.* **152**, 224108 (2020).
- 18 S. G. Balasubramani, G. P. Chen, S. Coriani, M. Diedenhofen, M. S. Frank, Y. J. Franzke, F. Furche, R. Grotjahn, M. E. Harding, C. Hättig, A. Hellweg, B. Helmich-Paris, C. Holzer, U. Huniar, M. Kaupp, A. M. Khah, S. K. Khani, T. Müller, F. Mack, B. D. Nguyen, S. M. Parker, E. Perlt, D. Rappoport, K. Reiter, S. Roy, M. Rückert, G. Schmitz, M. Sierka, E. Tapavicza, D. P. Tew, C. van Wüllen, V. K. Voora, F. Weigend, A. Wodyński, and J. M. Yu, *J. Chem. Phys.* **152**, 184107 (2020).
- 19 D. G. A. Smith, L. A. Burns, A. C. Simmonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. D. Remigio, A. Alenaiyan, A. M. James, S. Lehtola, J. P. Misiewicz, M. Scheurer, R. A. Shaw, J. B. Schriber, Y. Xie, Z. L. Glick, D. A. Sirianni, J. S. O'Brien, J. M. Waldrop, A. Kumar, E. G. Hohenstein, B. P. Pritchard, B. R. Brooks, H. F. Schaefer III, A. Y. Sokolov, K. Patkowski, A. E. DePrince III,

U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, *J. Chem. Phys.* **152**, 184108 (2020).

<sup>20</sup>K. Nakano, C. Attaccalite, M. Barborini, L. Capriotti, M. Casula, E. Coccia, M. Dagrada, C. Genovese, Y. Luo, G. Mazzola, A. Zen, and S. Sorella, *J. Chem. Phys.* **152**, 204121 (2020).

<sup>21</sup>E. Mutlu, A. Panyala, N. Gawande, A. Bagussetty, J. Glabe, J. Kim, K. Kowalski, N. P. Bauman, B. Peng, H. Pathak, J. Brabec, and S. Krishnamoorthy, *J. Chem. Phys.* **159**, 024801 (2023).

<sup>22</sup>M. Chan, T. Verstraelen, A. Tehrani, M. Richer, X. D. Yang, T. D. Kim, E. Vöhringer-Martinez, F. Heidar-Zadeh, and P. W. Ayers, *J. Chem. Phys.* **160**, 162501 (2024).

<sup>23</sup>J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, *WIREs Comput. Mol. Sci.* **2**, 556 (2012).

<sup>24</sup>R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince III, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. D. Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer III, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, *J. Chem. Theory Comput.* **13**, 3185 (2017).

<sup>25</sup>D. G. A. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. M. James, J. B. Schriber, T. Zhang, B. Zhang, A. S. Abbott, E. J. Berquist, M. H. Lechner, L. A. Cunha, A. G. Heide, J. M. Waldrop, T. Y. Takeshita, A. Alenaizan, D. Neuhauser, R. A. King, A. C. Simmonett, J. M. Turney, H. F. Schaefer, F. A. Evangelista, A. Eugene DePrince III, T. D. Crawford, K. Patkowski, and C. D. Sherrill, *J. Chem. Theory Comput.* **14**, 3504 (2018).

<sup>26</sup>Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, *WIREs Comput. Mol. Sci.* **8**, e1340 (2018).

<sup>27</sup>S. Lehtola, *J. Chem. Phys.* **159**, 180901 (2023).

<sup>28</sup>M. G. Delcey, *WIREs Comput. Mol. Sci.* **13**, e1675 (2023).

<sup>29</sup>C. B. Hicks and T. J. Martinez, *J. Chem. Phys.* **160**, 142501 (2024).

<sup>30</sup>T. D. Crawford, C. D. Sherrill, E. F. Valeev, J. T. Fermann, R. A. King, M. L. Leininger, S. T. Brown, C. L. Janssen, E. T. Seidl, J. P. Kenny, and W. D. Allen, *J. Comput. Chem.* **28**, 1610 (2007).

<sup>31</sup>P. G. Szalay, T. Müller, G. Gidofalvi, H. Lischka, and R. Shepard, *Chem. Rev.* **112**, 108 (2012).

<sup>32</sup>D. I. Lyakh, M. Musiał, V. F. Lotrich, and R. J. Bartlett, *Chem. Rev.* **112**, 182 (2012).

<sup>33</sup>D. A. Mazziotti, *Chem. Rev.* **112**, 244 (2012).

<sup>34</sup>A. Köhn, M. Hanauer, L. A. Mück, T.-C. Jagau, and J. Gauss, *WIREs Comput. Mol. Sci.* **3**, 176 (2013).

<sup>35</sup>F. A. Evangelista, *J. Chem. Phys.* **149**, 030901 (2018).

<sup>36</sup>C. Li and F. A. Evangelista, *Annu. Rev. Phys. Chem.* **70**, 245 (2019).

<sup>37</sup>B. O. Roos, P. R. Taylor, and P. E. Siegbahn, *Chem. Phys.* **48**, 157 (1980).

<sup>38</sup>N. Handy, *Chem. Phys. Lett.* **74**, 280 (1980).

<sup>39</sup>G. L. Bendazzoli and S. Evangelisti, *J. Chem. Phys.* **98**, 3141 (1993).

<sup>40</sup>C. D. Sherrill and H. F. Schaefer, *Adv. Quantum Chem.* **34**, 143 (1999).

<sup>41</sup>J. Olsen, B. O. Roos, P. Jørgensen, and H. J. A. Jensen, *J. Chem. Phys.* **89**, 2185 (1988).

<sup>42</sup>J. Ivanic and K. Ruedenberg, *Theor. Chem. Acc.* **106**, 339 (2001).

<sup>43</sup>T. Fleig, J. Olsen, and C. M. Marian, *J. Chem. Phys.* **114**, 4775 (2001).

<sup>44</sup>D. Ma, G. Li Manni, and L. Gagliardi, *J. Chem. Phys.* **135**, 044128 (2011).

<sup>45</sup>J. Ivanic, *J. Chem. Phys.* **119**, 9364 (2003).

<sup>46</sup>J. Olsen, *J. Chem. Phys.* **141**, 034112 (2014).

<sup>47</sup>B. S. Fales and T. J. Martínez, *J. Chem. Phys.* **152**, 164111 (2020).

<sup>48</sup>J. B. Schriber and F. A. Evangelista, *J. Chem. Phys.* **144**, 161106 (2016).

<sup>49</sup>B. Huron, J. P. Malrieu, and P. Rancurel, *J. Chem. Phys.* **58**, 5745 (1973).

<sup>50</sup>J. B. Schriber, K. P. Hannon, C. Li, and F. A. Evangelista, *J. Chem. Theory Comput.* **14**, 6295 (2018).

<sup>51</sup>J. B. Schriber and F. A. Evangelista, *J. Chem. Theory Comput.* **13**, 5354 (2017).

<sup>52</sup>J. B. Schriber and F. A. Evangelista, *J. Chem. Phys.* **151**, 171102 (2019).

<sup>53</sup>T. Zhang and F. A. Evangelista, *J. Chem. Theory Comput.* **12**, 4326 (2016).

<sup>54</sup>G. H. Booth, A. J. W. Thom, and A. Alavi, *J. Chem. Phys.* **131**, 054106 (2009).

<sup>55</sup>G. Chaban, M. W. Schmidt, and M. S. Gordon, *Theor. Chem. Acc.* **97**, 88 (1997).

<sup>56</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **155**, 114111 (2021).

<sup>57</sup>M. R. Hoffmann and J. Simons, *J. Chem. Phys.* **88**, 993 (1988).

<sup>58</sup>S. R. White, *J. Chem. Phys.* **117**, 7472 (2002).

<sup>59</sup>T. Yanai and G. K.-L. Chan, *J. Chem. Phys.* **124**, 194106 (2006).

<sup>60</sup>D. A. Mazziotti, *Phys. Rev. Lett.* **97**, 143002 (2006).

<sup>61</sup>M. Nooijen, *J. Chem. Phys.* **104**, 2638 (1996).

<sup>62</sup>D. Datta, L. Kong, and M. Nooijen, *J. Chem. Phys.* **134**, 214116 (2011).

<sup>63</sup>M. Nooijen, O. Demel, D. Datta, L. Kong, K. R. Shamasundar, V. Lotrich, L. M. Huntington, and F. Neese, *J. Chem. Phys.* **140**, 081102 (2014).

<sup>64</sup>D. Mukherjee, *Chem. Phys. Lett.* **274**, 561 (1997).

<sup>65</sup>F. A. Evangelista, *J. Chem. Phys.* **141**, 054109 (2014).

<sup>66</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **146**, 124132 (2017).

<sup>67</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **144**, 164114 (2016).

<sup>68</sup>T. Zhang, C. Li, and F. A. Evangelista, *J. Chem. Theory Comput.* **15**, 4399 (2019).

<sup>69</sup>H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker, *Chem. Phys. Lett.* **228**, 233 (1994).

<sup>70</sup>A. E. DePrince III and C. D. Sherrill, *J. Chem. Theory Comput.* **9**, 2687 (2013).

<sup>71</sup>D. Zgid, D. Ghosh, E. Neuscamman, and G. K.-L. Chan, *J. Chem. Phys.* **130**, 194107 (2009).

<sup>72</sup>K. Chatterjee and A. Y. Sokolov, *J. Chem. Theory Comput.* **16**, 6343 (2020).

<sup>73</sup>Y. Guo, K. Sivalingam, and F. Neese, *J. Chem. Phys.* **154**, 214111 (2021).

<sup>74</sup>Y. Guo, K. Sivalingam, C. Kollmar, and F. Neese, *J. Chem. Phys.* **154**, 214113 (2021).

<sup>75</sup>S. Li, J. P. Misiewicz, and F. A. Evangelista, *J. Chem. Phys.* **159**, 114106 (2023).

<sup>76</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **148**, 079903 (2018).

<sup>77</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **148**, 079902 (2018).

<sup>78</sup>M. Wang, W.-H. Fang, and C. Li, *J. Chem. Theory Comput.* **19**, 122 (2023).

<sup>79</sup>T. H. Dunning, *J. Chem. Phys.* **90**, 1007 (1989).

<sup>80</sup>D. E. Woon and T. H. Dunning, *J. Chem. Phys.* **103**, 4572 (1995).

<sup>81</sup>G. D. Purvis and R. J. Bartlett, *J. Chem. Phys.* **76**, 1910 (1982).

<sup>82</sup>K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).

<sup>83</sup>K. Andersson, P. A. Malmqvist, B. O. Roos, A. J. Sadlej, and K. Wolinski, *J. Phys. Chem.* **94**, 5483 (1990).

<sup>84</sup>C. Angel, R. Cimiraglia, S. Evangelisti, T. Leininger, and J. P. Malrieu, *J. Chem. Phys.* **114**, 10252 (2001).

<sup>85</sup>H.-J. Werner and P. J. Knowles, *J. Chem. Phys.* **89**, 5803 (1988).

<sup>86</sup>C. Li and F. A. Evangelista, *J. Chem. Phys.* **148**, 124106 (2018).

<sup>87</sup>W. Kutzelnigg and D. Mukherjee, *J. Chem. Phys.* **107**, 432 (1997).

<sup>88</sup>J.-N. Boyn and D. A. Mazziotti, *J. Chem. Phys.* **154**, 134103 (2021).

<sup>89</sup>M. Huang, C. Li, and F. A. Evangelista, *J. Chem. Theory Comput.* **18**, 219 (2022).

<sup>90</sup>S. Wouters, W. Poelmans, P. W. Ayers, and D. Van Neck, *Comput. Phys. Commun.* **185**, 1501 (2014).

<sup>91</sup>H. Zhai, H. R. Larsson, S. Lee, Z.-H. Cui, T. Zhu, C. Sun, L. Peng, R. Peng, K. Liao, J. Tölle, J. Yang, S. Li, and G. K.-L. Chan, *J. Chem. Phys.* **159**, 234801 (2023).

<sup>92</sup>C. Li and F. A. Evangelista, *J. Chem. Theory Comput.* **11**, 2097 (2015).

<sup>93</sup>K. P. Hannon, C. Li, and F. A. Evangelista, *J. Chem. Phys.* **144**, 204111 (2016).

<sup>94</sup>N. He and F. A. Evangelista, *J. Chem. Phys.* **152**, 094107 (2020).

<sup>95</sup>P. Verma, W. D. Derricotte, and F. A. Evangelista, *J. Chem. Theory Comput.* **12**, 144 (2016).

<sup>96</sup>E. R. Sayfutyarova and S. Hammes-Schiffer, *J. Chem. Theory Comput.* **15**, 1679 (2019).

<sup>97</sup>E. R. Sayfutyarova, Q. Sun, G. K.-L. Chan, and G. Knizia, *J. Chem. Theory Comput.* **13**, 4063 (2017).

<sup>98</sup>C. Li, S. Mao, R. Huang, and F. A. Evangelista, *J. Chem. Theory Comput.* **20**, 4170 (2024).

<sup>99</sup>N. He, C. Li, and F. A. Evangelista, *J. Chem. Theory Comput.* **18**, 1527 (2022).

<sup>100</sup>S. Wang, C. Li, and F. A. Evangelista, *J. Chem. Theory Comput.* **17**, 7666 (2021).

<sup>101</sup>J. W. Park, *J. Chem. Theory Comput.* **18**, 2233 (2022).

<sup>102</sup>T. Shiozaki, *WIREs Comput. Mol. Sci.* **8**, e1331 (2017).

<sup>103</sup>M. Svensson, S. Humbel, R. D. J. Froese, T. Matsubara, S. Sieber, and K. Morokuma, *J. Phys. Chem.* **100**, 19357 (1996).

<sup>104</sup>R. M. Richard and J. M. Herbert, *J. Chem. Phys.* **137**, 064113 (2012).

<sup>105</sup>F. R. Manby, M. Stella, J. D. Goopaster, and T. F. Miller, *J. Chem. Theory Comput.* **8**, 2564 (2012).

<sup>106</sup>M. Isegawa, B. Wang, and D. G. Truhlar, *J. Chem. Theory Comput.* **9**, 1381 (2013).

<sup>107</sup>F. Libisch, C. Huang, and E. A. Carter, *Acc. Chem. Res.* **47**, 2768 (2014).

<sup>108</sup>Q. Sun and G. K.-L. Chan, *Acc. Chem. Res.* **49**, 2705 (2016).

<sup>109</sup>A. Wasserman and M. Pavanello, *Int. J. Quantum Chem.* **120**, e26495 (2020).

<sup>110</sup>G. Knizia and G. K.-L. Chan, *J. Chem. Theory Comput.* **9**, 1428 (2013).

<sup>111</sup>H. Q. Pham, V. Bernales, and L. Gagliardi, *J. Chem. Theory Comput.* **14**, 1960 (2018).

<sup>112</sup>A. S. D. Meraés, M. B. Lepetit, and J. P. Malrieu, *Chem. Phys. Lett.* **172**, 163 (1990).

<sup>113</sup>A. Zaitsevskii and J. P. Malrieu, *Chem. Phys. Lett.* **228**, 458 (1994).

<sup>114</sup>A. Marie and H. G. A. Burton, *J. Phys. Chem. A* **127**, 4538 (2023).

<sup>115</sup>E. R. Sayfutyarova, *J. Chem. Theory Comput.* **20**, 79 (2024).

<sup>116</sup>A. G. Taube and R. J. Bartlett, *J. Chem. Phys.* **128**, 164101 (2008).

<sup>117</sup>A. Landau, K. Khistyayev, S. Dolgikh, and A. I. Krylov, *J. Chem. Phys.* **132**, 014109 (2010).

<sup>118</sup>M. Huang and F. A. Evangelista, *J. Chem. Phys.* **158**, 124112 (2023).

<sup>119</sup>N. H. F. Beebe and J. Linderberg, *Int. J. Quantum Chem.* **12**, 683 (1977).

<sup>120</sup>I. Røeggen and E. Wisløff-Nilssen, *Chem. Phys. Lett.* **132**, 154 (1986).

<sup>121</sup>H. Koch, A. S. de Merás, and T. B. Pedersen, *J. Chem. Phys.* **118**, 9481 (2003).

<sup>122</sup>F. Aquilante, R. Lindh, and T. B. Pedersen, *J. Chem. Phys.* **127**, 114107 (2007).

<sup>123</sup>J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).

<sup>124</sup>B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 4993 (1979).

<sup>125</sup>O. Vahtras, J. Almlöf, and M. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).

<sup>126</sup>M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).

<sup>127</sup>F. Weigend, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).

<sup>128</sup>F. Weigend, M. Kattannek, and R. Ahlrichs, *J. Chem. Phys.* **130**, 164106 (2009).

<sup>129</sup>P. J. Knowles and N. C. Handy, *Comput. Phys. Commun.* **54**, 75 (1989).

<sup>130</sup>R. Huang, C. Li, and F. A. Evangelista, *PRX Quantum* **4**, 020313 (2023).

<sup>131</sup>A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, *Nat. Commun.* **5**, 4213 (2014).

<sup>132</sup>N. H. Stair and F. A. Evangelista, *J. Chem. Theory Comput.* **18**, 1555 (2022).

<sup>133</sup>N. He, M. Huang, and F. A. Evangelista, *J. Phys. Chem. A* **127**, 1975 (2023).

<sup>134</sup>J. A. Lau, A. Choudhury, C. Li, D. Schwarzer, V. B. Verma, and A. M. Wodtke, *Science* **367**, 175 (2020).

<sup>135</sup>R. Püttner, I. Dominguez, T. J. Morgan, C. Cisneros, R. F. Fink, E. Rotenberg, T. Warwick, M. Domke, G. Kaindl, and A. S. Schlachter, *Phys. Rev. A* **59**, 3415 (1999).

<sup>136</sup>N. H. Stair and F. A. Evangelista, *PRX Quantum* **2**, 030301 (2021).

<sup>137</sup>Z. Zhao and F. A. Evangelista, *J. Phys. Chem. Lett.* **15**, 7103 (2024).

<sup>138</sup>C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. D. Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Nature* **585**, 357 (2020).