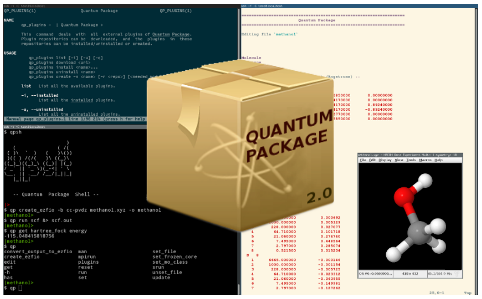
Quantum Package 2.0: An Open-Source Determinant-Driven Suite of Programs

Yann Garniron,[1] Thomas Applencourt,[2] Kevin Gasperich,[2, 3] Anouar Benali,[2] Anthony Ferté,[4] Julien Paquier,[4] Barthélémy Pradines,[4, 5] Roland Assaraf,[4] Peter Reinhardt,[4] Julien Toulouse,[4] Pierrette Barbaresco,[6] Nicolas Renon,[6] Grégoire David,[7] Jean-Paul Malrieu,[1] Mickaël Véril,[1] Michel Caffarel,[1] Pierre-François Loos,[1, a)] Emmanuel Giner,[4, b)] and Anthony Scemama[1, c)]

[1)] *Laboratoire de Chimie et Physique Quantiques (UMR 5626), Université de Toulouse, CNRS, UPS, France*
[2)] *Computational Science Division, Argonne National Laboratory, Argonne, IL 60439, United States*
[3)] *Department of Chemistry, University of Pittsburgh, Pittsburgh, PA 15260, United States*
[4)] *Laboratoire de Chimie Théorique, Sorbonne Université, CNRS, Paris, France*
[5)] *Institut des Sciences du Calcul et des Données, Sorbonne Université, F-75005 Paris, France*
[6)] *CALMIP, Université de Toulouse, CNRS, INPT, INSA, UPS, UMS 3667, France*
[7)] *Aix-Marseille Univ, CNRS, ICR, Marseille, France*

Quantum chemistry is a discipline which relies heavily on very expensive numerical computations. The scaling of correlated wave function methods lies, in their standard implementation, between $\mathcal{O}(N^5)$ and $\mathcal{O}(e^N)$, where N is the number of one-electron basis functions. Therefore, performing accurate calculations on chemically meaningful systems requires i) approximations that can lower the computational scaling, and ii) efficient implementations that take advantage of modern massively parallel architectures. Quantum Package is an open-source programming environment for quantum chemistry specially designed for wave function methods. Its main goal is the development of determinant-driven selected configuration interaction (sCI) methods and multi-reference second-order perturbation theory (PT2). The determinant-driven framework allows the programmer to include any arbitrary set of determinants in the reference space, hence providing greater methodological freedoms. The sCI method implemented in Quantum Package is based on the CIPSI (Configuration Interaction using a Perturbative Selection made Iteratively) algorithm which complements the variational sCI energy with a PT2 correction. Additional external plugins have been recently added to perform calculations with multireference coupled cluster theory and range-separated density-functional theory. All the programs are developed with the IRPF90 code generator, which simplifies collaborative work and the development of new features. Quantum Package strives to allow easy implementation and experimentation of new methods, while making parallel computation as simple and efficient as possible on modern supercomputer architectures. Currently, the code enables, routinely, to realize runs on roughly 2 000 CPU cores, with tenths of millions of determinants in the reference space. Moreover, we have been able to push up to 12 288 cores in order to test its parallel efficiency. In the present manuscript, we also introduce some key new developments: i) a renormalized second-order perturbative correction for efficient extrapolation to the full CI limit, and ii) a stochastic version of the CIPSI selection performed simultaneously to the PT2 calculation at no extra cost.



```
QP_PLUGINS(1)                Quantum Package                QP_PLUGINS(1)

NAME
       qp_plugins - | Quantum Package >

       This  command  deals  with  all  external plugins of Quantum Package.
       Plugin repositories can be  downloaded,  and  the  plugins  in  these
       repositories can be installed/uninstalled or created.

USAGE
          qp_plugins list [-i] [-u] [-q]
          qp_plugins download <url>
          qp_plugins install <name>...
          qp_plugins uninstall <name>
          qp_plugins create -n <name> [-r <repo>] [<needed_mo

       list   List all the available plugins.

       -i, --installed
              List all the installed plugins.

       -u, --uninstalled
              List all the uninstalled plugins.
 Manual page qp_plugins.1 line 1/96 21% (press h for help
```

```
$ qpsh

   -- Quantum  Package  Shell --

|>
$ qp create_ezfio -b cc-pvdz methanol.xyz -o methanol
|methanol>
$ qp run scf &> scf.out
|methanol>
$ qp get hartree_fock energy
-115.048415818756
|methanol>
$ qp
convert_output_to_ezfio  man                     set_file
create_ezfio             mpirun                  set_frozen_core
edit                     plugins                 set_mo_class
get                      reset                   srun
-h                       run                     unset_file
has                      set                     update
|methanol>
$ qp
```

TOC graphical abstract

I. INTRODUCTION

In 1965, Gordon Moore predicted that the number of transistors in an integrated circuit would double about every two years (the so-called Moore's law).[1] Rapidly, this "law" was interpreted as an expected two-fold increase in performance every 18 months. This became an industrial goal. The development of today's most popular electronic structure codes was initiated in the 1990's (or even before). At that time, the increase of computational power from one supercomputer generation to the next was mostly driven by an increase of processors' frequency. Indeed, the amount of random access memory was small, the time to access data from disk was slow, and the energy consumption of the most powerful computer was 236 kW, hence far from being an economical concern.[2] At the very beginning of the 21st century, having increased continuously, both the number of processors and their frequency raised the supercomputer power consumption by two orders of magnitude, inflating accordingly the electricity bill. The only way to slow down this frenetic growth of power consumption while keeping alive Moore's dream was to freeze the processor's frequency (between 1 and 4 GHz), and increase the number of CPU cores. The consequence of such a choice was that *"free lunch"* was over: the programmers now had to parallelize their programs to make them run faster.[3] At the

[a)] Electronic mail: loos@irsamc.ups-tlse.fr
[b)] Electronic mail: emmanuel.giner@lct.jussieu.fr
[c)] Electronic mail: scemama@irsamc.ups-tlse.fr

same time, computer scientists realized that the increase of performance in memory access was slower than the increase in computational power,[4] and that the floating-point operation (or flop) count would soon stop being the bottleneck. From now on, data movement would be the main concern. This paradigm shift was named the *memory wall.* Moore's law is definitely near the end of its life.[5]

The traditional sequential algorithms of quantum chemistry are currently being redesigned and replaced by parallel equivalents by multiple groups around the world.[6–18] This has obviously a significant influence on methodological developments. The most iconic example of this move towards parallel-friendly methods is the recently developed *full configuration interaction quantum Monte Carlo* (FCIQMC) method by Alavi and coworkers.[6] FCIQMC can be interpreted as a Monte Carlo equivalent of older selected configuration interaction (sCI) algorithms[9,10,19–54] such as CIPSI (Configuration Interaction using a Perturbative Selection made Iteratively),[21] that are iterative and thus *a priori* not well adapted to massively parallel architecture. As we shall see here, things turn out differently, and the focus of the present article is to show that sCI methods can be made efficient on modern massively parallel supercomputers.

Quantum Package[55] is an open-source suite of wave function quantum chemistry methods mainly developed at the *Laboratoire de Chimie et Physique Quantiques* (LCPQ) in Toulouse (France), and the *Laboratoire de Chimie Théorique* (LCT) in Paris. Its source code is freely available on GitHub at the following address: https://github.com/QuantumPackage/qp2. Quantum Package strives to allow easy implementation and experimentation of new methods, while making parallel computation as simple and efficient as possible. Accordingly, the initial choice of Quantum Package was to go towards *determinant-driven* algorithms. Assuming a wave function expressed as a linear combination of determinants, a determinant-driven algorithm essentially implies that the outermost loop runs over determinants. On the other hand, more traditional *integral-driven* algorithms have their outermost loop running on the two-electron integrals appearing in the expression of the matrix elements in the determinant basis (see Sec. II B). Determinant-driven algorithms allow more flexibility than their integral-driven counterparts,[56] but they have been known for years to be less efficient than their integral-driven variant for solving electronic structure problems. In high-precision calculations, the number of determinants is larger than the number of integrals, justifying the integral-driven choice. However, today's programming standards impose parallelism, and if determinant-driven calculations prove to be better adapted to parallelism, such methods could regain in popularity.

Quantum Package was used in numerous applications, in particular to obtain reference ground-state energies[34–38,57] as well as excitation energies[44,54,58] for atomic and molecular systems. For example, in Ref. 44, Quantum Package has been used to compute more than hundred very accurate transition energies for states of various characters (valence, Rydberg, $n \to \pi^*$, $\pi \to \pi^*$, singlet, triplet, . . .) in 18 small molecules. The high quality and compactness of the CIPSI wave function was also used for quantum Monte Carlo calculations to characterize the ground state of the water and the FeS molecules,[38,42] and obtained highly accurate excitation energies.[43,59,60] Of course, the technical considerations were not the main concern of the different articles that were produced. Because the present work focused on the actual implementation of the methods at least as much as on the theory behind them, this article is a perfect opportunity to discuss in depth their implementation.

This manuscript is organized as follows. In Sec. II, we briefly describe the main computational methods implemented in Quantum Package as well as newly developed methods and extrapolation techniques. Section III deals with their implementation. In particular, Sec. III A discusses the computation of the Hamiltonian matrix elements using determinant-driven algorithms, while Sec. III B focuses on the acceleration of the Davidson diagonalization, a pivotal point of sCI methods. In Sec. III C, we focus on the determinant selection step used to build compact wave functions. In a nutshell, the principle is to incrementally build a reference wave function by scavenging its external space for determinants that interact with it. To make this step more affordable, we designed a new stochastic scheme which selects *on the fly* the more important determinants while the second-order perturbative (PT2) energy is computed using a hybrid stochastic-deterministic scheme.[10] Therefore, the selection part of this new stochastic CIPSI selection is virtually free as long as one is interested in the second-order perturbative correction, which is crucial in many cases in order to obtain near *full configuration interaction* (FCI) results. The efficiency of the present algorithms is demonstrated in Sec. VI C where illustrative calculations and parallel speedups are reported. Finally, Sec. VII discusses the development philosophy of Quantum Package as well as other relevant technical details. Unless otherwise stated, atomic units are used throughout.

II. METHODS

A. Generalities

The correlation energy is defined as[61]

$$E_c = E_{\text{exact}} - E_{\text{HF}}, \tag{1}$$

where E_{exact} and E_{HF} are, respectively, the exact (non-relativistic) energy and the Hartree-Fock (HF) energy in a complete (one-electron) basis set.

To include electron correlation effects, the wave function associated with the kth electronic state, $|\Psi_k\rangle$, may be expanded in the set of all possible N-electron Slater determinants, $|I\rangle$, built by placing $N_\uparrow$ spin-up electrons in N_{orb} orbitals and $N_\downarrow$ spin-down electrons in N_{orb} orbitals (where $N = N_\uparrow + N_\downarrow$). These so-called molecular orbitals (MOs) are defined as linear combinations of atomic orbitals (AOs)

$$\phi_p(\mathbf{r}) = \sum_{\mu}^{N_{\text{orb}}} C_{\mu p} \chi_\mu(\mathbf{r}). \tag{2}$$

Note that the MOs are assumed to be real valued in the context of this work. The eigenvectors of the Hamiltonian $\hat{H}$ are consequently expressed as linear combinations of Slater determinants, i.e.,

$$|\Psi_k\rangle = \sum_{I}^{N_{\text{det}}} c_{Ik} \, |I\rangle \, , \tag{3}$$

where N_{det} is the number of determinants. For sake of conciseness, we will restrict the discussion to the ground state (i.e. $k = 0$) and drop the subscript k accordingly. Solving the eigenvalue problem in this basis is referred to as FCI and yields, for a given basis set, the exact solution of the Schrödinger equation. Unfortunately, FCI is usually computationally intractable because of its exponential scaling with the size of the basis set (i.e., N_{orb}).

B. Matrix elements of the Hamiltonian

In the N-electron basis of Slater determinants, one expects the matrix elements of $\hat{H}$ to be integrals over $3N$ dimensions. However, given the two-electron nature of the Hamiltonian, and because the MOs are orthonormal, Slater determinants that differ by more than two spinorbitals yield a zero matrix element. The remaining elements can be expressed as sums of integrals over one- or two-electron coordinates, which can be computed at a reasonable cost. These simplifications are known as Slater-Condon's rules, and reads

$$\langle I|\hat{H}|I\rangle = \sum_{i\in|I\rangle} (i|\hat{h}|i) + \frac{1}{2} \sum_{(i,j)\in|I\rangle} (ii||jj), \tag{4a}$$

$$\langle I|\hat{H}|I_p^r\rangle = (p|\hat{h}|r) + \sum_{i\in|I\rangle} (pr||ii), \tag{4b}$$

$$\langle I|\hat{H}|I_{pq}^{rs}\rangle = (pr||qs), \tag{4c}$$

where $\hat{h}$ is the one-electron part of the Hamiltonian (including kinetic energy and electron-nucleus attraction operators),

$$(p|\hat{h}|q) = \int \phi_p(\mathbf{r})\hat{h}(\mathbf{r})\phi_q(\mathbf{r})d\mathbf{r} \tag{5}$$

are one-electron integrals, $i \in |I\rangle$ means that ϕ_i belongs to the Slater determinant $|I\rangle$, $|I_p^r\rangle$ and $|I_{pq}^{rs}\rangle$ are determinants obtained from $|I\rangle$ by substituting orbitals ϕ_p by ϕ_r, and ϕ_p and ϕ_q by ϕ_r and ϕ_s, respectively,

$$(pq|rs) = \iint \phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)r_{12}^{-1}\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{6}$$

are two-electron electron repulsion integrals (ERIs), $r_{12}^{-1} = |\mathbf{r}_1 - \mathbf{r}_2|^{-1}$ is the Coulomb operator, and $(pq||rs) = (pq|rs) - (ps|rq)$ are the usual antisymmetrized two-electron integrals.

C. Two-electron integrals

Within the HF method, Roothaan's equations allow to solve the problem in the AO basis.[62] In this context, one needs to compute the $\mathcal{O}(N_{\text{orb}}^4)$ two-electron integrals $(\mu\nu|\lambda\sigma)$ over the AO basis. Thanks to a large effort in algorithmic development and implementation,[63–70] these integrals can now be computed very fast on modern computers.

However, with post-HF methods, the computation of the two-electron integrals is a potential bottleneck. Indeed, when computing matrix elements of the Hamiltonian in the basis of Slater determinants, ERIs over MOs are required. Using Eq. (2), the cost of computing a single integral

$$(pq|rs) = \sum_{\mu\nu\lambda\sigma} C_{\mu p} C_{\nu q} C_{\lambda r} C_{\sigma s} (\mu\nu|\lambda\sigma) \tag{7}$$

scales as $\mathcal{O}(N_{\text{orb}}^4)$. A naive computation of all integrals in the MO basis would cost $\mathcal{O}(N_{\text{orb}}^8)$. Fortunately, computing all of them can be scaled down to $\mathcal{O}(N^5)$ by transforming the indices one by one:[71]

$$(p\nu|\lambda\sigma) = \sum_{\mu} C_{\mu p}(\mu\nu|\lambda\sigma),$$

$$(p\nu|r\sigma) = \sum_{\lambda} C_{\lambda r}(p\nu|\lambda\sigma), \quad \text{(semi-transformed)}, \tag{8a}$$

$$(pq|r\sigma) = \sum_{\nu} C_{\nu q}(p\nu|r\sigma),$$

$$(pq|rs) = \sum_{\sigma} C_{\sigma s}(pq|r\sigma), \quad \text{(fully transformed)}. \tag{8b}$$

This step is known as the four-index integral transformation. In addition to being very costly, this step is hard to parallelize in a distributed way, because it implies multiple collective communications.[72–75]

D. Selected CI methods

The sCI methods rely on the same principle as the usual configuration interaction (CI) approaches, except that determinants are not chosen *a priori* based on occupation or excitation criteria, but selected among the entire set of determinants based on their estimated contribution to the FCI wave function. Indeed, it has been noticed long ago that, even inside a predefined subspace of determinants, only a small number of them significantly contributes.[33,76] Therefore, an *on-the-fly* selection of determinants is a rather natural idea that has been proposed in the late 1960's by Bender and Davidson[19] as well as Whitten and Hackmeyer.[20] sCI methods are still very much under active development.[51,77] The main advantage of sCI methods is that no *a priori* assumption is made on the type of electronic correlation. Therefore, at the price of a brute force calculation, a sCI calculation is less biased by the user's appreciation of the problem's complexity.

The approach that we have implemented in Quantum Package is based on the CIPSI algorithm developed by Huron, Rancurel and Malrieu in 1973,[21] that iteratively selects external determinants $|\alpha\rangle$ — determinants which are not present in the (reference or variational) zeroth-order wave function

$$|\Psi^{(0)}\rangle = \sum_{I} c_I \, |I\rangle \tag{9}$$

at a given iteration — using a perturbative criterion

$$e_{\alpha}^{(2)} = \frac{\langle \Psi^{(0)} | \hat{H} | \alpha \rangle^2}{E^{(0)} - \langle \alpha | \hat{H} | \alpha \rangle}, \tag{10}$$

where

$$E^{(0)} = \frac{\langle \Psi^{(0)} | \hat{H} | \Psi^{(0)} \rangle}{\langle \Psi^{(0)} | \Psi^{(0)} \rangle} \geq E_{\text{FCI}} \tag{11}$$

is the zeroth-order (variational) energy, and $e_{\alpha}^{(2)}$ the (second-order) estimated gain in correlation energy that would be brought by the inclusion of $|\alpha\rangle$. The second-order perturbative correction

$$E^{(2)} = \sum_{\alpha} e_{\alpha}^{(2)} = \sum_{\alpha} \frac{\langle \alpha | \hat{H} | \Psi^{(0)} \rangle^2}{E^{(0)} - \langle \alpha | \hat{H} | \alpha \rangle} \tag{12}$$

is an estimate of the total missing correlation energy, i.e., $E^{(2)} \approx E_{\text{FCI}} - E^{(0)}$, for large enough expansions.

Let us emphasize that sCI methods can be applied to any determinant space. Although presented here for the FCI space, it can be trivially generalized to a complete active space (CAS), but also to standard CI spaces such as CIS, CISD or MR-CISD. The only required modification is to set to zero the contributions associated with the determinants which do not belong to the target space.

There is, however, a computational downside to sCI methods. In conventional CI methods, the rule by which determinants are selected is known *a priori*, and therefore, one can map a particular determinant to some row or column indices.[78] As a consequence, it can be systematically determined to which matrix element of $\hat{H}$ a two-electron integral contributes. This allows for the implementation of so-called *integral-driven* methods that work essentially by iterating over integrals. On the contrary, in (most) sCI methods, the determinants are selected *a posteriori*, and an explicit list has to be maintained as there is no immediate way to know whether or not a determinant has been selected. Consequently, we must rely on the so-called *determinant-driven* approach in which iterations are performed over determinants rather than integrals. This can be a lot more expensive, since the number of determinants N_{det} is typically much larger than the number of integrals. The number of determinants scales as $\mathcal{O}(N_{\text{orb}}!)$ while the number of integrals scales (formally) as $\mathcal{O}(N_{\text{orb}}^4)$. What makes sCI calculations possible in practice is that sCI methods generate relatively compact wave functions, i.e. wave functions where N_{det} is much smaller (by orders of magnitude) than the size of the FCI space. Furthermore, determinant-driven methods require an effective way to compare determinants in order to extract the corresponding excitation operators, and a way to rapidly fetch the associated integrals involved, as described in Sec. III A.

Because of this high computational cost, approximations have been proposed.[24] Recently, the heat-bath configuration interaction (HBCI) algorithm has taken further the idea of a more approximate but extremely cheap selection.[9,39,53] Compared to CIPSI, the selection criterion is simplified to

$$e_{\alpha}^{\text{HBCI}} = \max_{I} \left(\left| c_I \langle I | \hat{H} | \alpha \rangle \right| \right). \tag{13}$$

This algorithmically allows for an extremely fast selection of doubly-excited determinants by an integral-driven approach. Nonetheless, the bottlenecks of the HBCI are the diagonalization step and the computation of $E^{(2)}$, which remain determinant driven.

As mentioned above, FCIQMC is an alternative approach of stochastic nature recently developed in Alavi's group,[6–8] where signed walkers spawn from one determinant to connected ones, with a probability that is a function of the associated matrix element. The average proportion of walkers on a determinant converges to its coefficient in the FCI wave function. A more "brute force" approach is the purely stochastic selection of Monte Carlo CI (MCCI),[79,80] where determinants are randomly added to the zeroth-order wave function. After diagonalization, the determinants of smaller coefficient are removed, and new random determinants are added.

A number of other variants have been developed but are not detailed here.[9,10,19–21,24–28,30,31,33–54] Although these various approaches appear under diverse acronyms, most of them rely on the very same idea of selecting determinants iteratively according to their contribution to the wave function or energy.

E. Extrapolation techniques

1. *Usual extrapolation procedure*

In order to extrapolate the sCI results to the FCI limit, we have adopted the method recently proposed by Holmes, Umrigar and Sharma[40] in the context of the HBCI method.[9,39,40] It consists in extrapolating the sCI energy, $E^{(0)}$, as a function of the second-order Epstein-Nesbet energy, $E^{(2)}$, which is an estimate of the truncation error in the sCI algorithm, i.e $E^{(2)} \approx E_{\text{FCI}} - E^{(0)}$.[21] When $E^{(2)} = 0$, the FCI limit has effectively been reached. This extrapolation procedure has been shown to be robust, even for challenging chemical situations.[9,40–45,54] Below, we propose an improved extrapolation scheme which *renormalizes* the second-order perturbative correction.

2. *Renormalized PT2*

At a given sCI iteration, the sCI+PT2 energy is given by

$$E = E^{(0)} + E^{(2)}, \tag{14}$$

where $E^{(0)}$ and $E^{(2)}$ are given by Eqs. (11) and (12), respectively. Let us introduce the following energy-dependent second-order self-energy

$$\Sigma^{(2)}[E] = \sum_{\alpha} \frac{\langle \alpha | \hat{H} | \Psi^{(0)} \rangle^2}{E - \langle \alpha | \hat{H} | \alpha \rangle}. \tag{15}$$

Obviously, we have $\Sigma^{(2)}[E^{(0)}] = E^{(2)}$. Now, let us consider the more general problem, which is somewhat related to Brillouin-Wigner perturbation theory, where we have

$$E = E^{(0)} + \Sigma^{(2)}[E], \tag{16}$$

and assume that $\Sigma^{(2)}[E]$ behaves linearly for $E \approx E^{(0)}$, i.e.,

$$\Sigma^{(2)}[E] \approx \Sigma^{(2)}[E^{(0)}] + (E - E^{(0)}) \left. \frac{\partial \Sigma^{(2)}[E]}{\partial E} \right|_{E=E^{(0)}} . \qquad (17)$$

This linear behavior is corroborated by the findings of Nitzsche and Davidson.[81] Substituting Eq. (17) into (16) yields

$$\begin{aligned} E &= E^{(0)} + \Sigma^{(2)}[E^{(0)}] + (E - E^{(0)}) \left. \frac{\partial \Sigma^{(2)}[E]}{\partial E} \right|_{E=E^{(0)}} \\ &= E^{(0)} + Z\, E^{(2)}, \end{aligned} \qquad (18)$$

where the renormalization factor is

$$Z = \left[1 - \left. \frac{\partial \Sigma^{(2)}[E]}{\partial E} \right|_{E=E^{(0)}} \right]^{-1}, \qquad (19)$$

and

$$\left. \frac{\partial \Sigma^{(2)}[E]}{\partial E} \right|_{E=E^{(0)}} = - \sum_{\alpha} \frac{\langle \alpha | \hat{H} | \Psi^{(0)} \rangle^2}{(E^{(0)} - \langle \alpha | \hat{H} | \alpha \rangle)^2} < 0. \qquad (20)$$

Therefore, the renormalization factor fulfills the condition $0 \leq Z \leq 1$, and its actual computation does not involve any additional cost when computed alongside $E^{(2)}$ as they involve the same quantities. This renormalization procedure of the second-order correction, that we have named rPT2, bears obvious similarities with the computation of quasiparticle energies within the G_0W_0 method.[82–85] Practically, the effect of rPT2 is to damp the value of $E^{(2)}$ for small wave functions. Indeed, when N_{det} is small, the sum $E^{(0)} + E^{(2)}$ usually overestimates (in magnitude) the FCI energy, yielding a pronounced non-linear behavior of the sCI+PT2 energy. Consequently, by computing instead the (renormalized) energy $E^{(0)} + Z\, E^{(2)}$, one observes a much more linear behavior of the energy, hence an easier extrapolation to the FCI limit. Its practical usefulness is illustrated in Sec. VI A.

F. Range-separated density-functional theory

Quantum Package also contains the basic tools to use and develop range-separated density-functional theory (RSDFT, see, e.g., Refs. 86 and 87) which allows to perform multi-configurational density-functional theory (DFT) calculations within a rigorous mathematical framework. In the core modules of Quantum Package, single-determinant approximations of RSDFT are available, which fall into the so-called range-separated hybrid[88,89] (RSH) approximation. These approaches correct for the wrong long-range behavior of the usual hybrid approximations thanks to the inclusion of the long-range part of the HF exchange.

Some genuine multi-configurational methods for RSDFT are available as external plugins (see, for example, https://gitlab.com/eginer/qp_plugins_eginer).

1. Range-separated DFT in a nutshell

By using the Levy-Lieb constrained search[90–93] and splitting the electron-electron interaction into long- and short-range contributions, the exact ground-state energy E of a general N-electron system can be written as a minimum over general multi-determinant wave functions Ψ (see, e.g., Refs. 87 and 94):

$$E = \min_{\Psi} \left\{ \langle \Psi | \hat{h} + \hat{W}_{\text{ee}}^{\text{lr},\mu} | \Psi \rangle + \bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n_{\Psi}(\mathbf{r})] \right\}, \qquad (21)$$

where

$$\hat{W}_{\text{ee}}^{\text{lr},\mu} = \sum_{i<j} w_{\text{ee}}^{\text{lr},\mu}(r_{ij}) \qquad (22)$$

is the long-range electron-electron interaction operator,

$$w_{\text{ee}}^{\text{lr},\mu}(r_{12}) = \frac{\text{erf}(\mu r_{12})}{r_{12}} \qquad (23)$$

is the long-range interaction potential, μ is the range-separation parameter (having the unit of the inverse of a distance), $\text{erf}(x)$ is the error function,[95] and $\bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n_{\Psi}(\mathbf{r})]$ is the complementary short-range Hartree-exchange-correlation functional evaluated at $n_{\Psi}(\mathbf{r})$, i.e. the one-electron density of Ψ.

The wave function Ψ^{μ} minimizing the energy in Eq. (21) satisfies the Schrödinger-like equation:

$$\left(\hat{h} + \hat{W}_{\text{ee}}^{\text{lr},\mu} + \hat{\bar{V}}_{\text{Hxc}}^{\text{sr},\mu}[n_{\Psi^{\mu}}(\mathbf{r})] \right) | \Psi^{\mu} \rangle = \mathcal{E}^{\mu} \, | \Psi^{\mu} \rangle \, , \qquad (24)$$

where

$$\hat{\bar{V}}_{\text{Hxc}}^{\text{sr},\mu}[n(\mathbf{r})] = \int \frac{\delta \bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n]}{\delta n(\mathbf{r})} \hat{n}(\mathbf{r}) d\mathbf{r} \qquad (25)$$

is the complementary short-range Hartree-exchange-correlation potential operator [using the density operator $\hat{n}(\mathbf{r})$], which acts as an effective one-electron operator maintaining the exact density. As a two-electron operator appears in Eq. (24), the wave function Ψ^{μ} is intrinsically multi-configurational.

RSDFT includes two special cases: when $\mu \to 0$ one recovers usual Kohn-Sham DFT and when $\mu \to \infty$ one recovers usual wave function theory. If no approximations are made on the wave function Ψ (i.e. full set of Slater determinants in a complete basis set) nor on the short-range functional $\bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n(\mathbf{r})]$, RSDFT provides the exact energy and density of the system, which of course do not depend on the range-separation parameter μ.

The short-range functional $\bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n(\mathbf{r})]$ is usually split in terms of the short-range Hartree and exchange-correlation functionals:

$$\bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n(\mathbf{r})] = E_{\text{H}}^{\text{sr},\mu}[n(\mathbf{r})] + \bar{E}_{\text{xc}}^{\text{sr},\mu}[n(\mathbf{r})], \qquad (26)$$

with

$$E_{\text{H}}^{\text{sr},\mu}[n(\mathbf{r})] = \frac{1}{2} \iint n(\mathbf{r}_1) n(\mathbf{r}_2) w_{\text{ee}}^{\text{sr},\mu}(r_{12}) d\mathbf{r}_1 d\mathbf{r}_2, \qquad (27)$$

where $w_{\text{ee}}^{\text{sr},\mu}(r_{12}) = r_{12}^{-1} - w_{\text{ee}}^{\text{lr},\mu}(r_{12})$ is the short-range interaction potential and, in practice, approximations must be performed for the remaining $\bar{E}_{\text{xc}}^{\text{sr},\mu}[n(\mathbf{r})]$. Nonetheless, the usual semi-local approximations for this functional are better tailored to treat the complementary short-range interaction.[87] Also, the use of an incomplete basis set necessary induces an error in Ψ, but it has been shown that the use of a non-diverging interaction in Eq. (24) considerably improves the basis-set convergence of the wave function part.[96]

2. *Range-separated hybrids*

Assuming a single Slater determinant wave function in Eq. (21), $\Psi \approx \Phi$, one obtains the RSH scheme[89]

$$E_{\text{RSH}}^{\mu} = \min_{\Phi} \langle \Phi | \hat{h} + \hat{W}_{\text{ee}}^{\text{lr},\mu} | \Phi \rangle + \bar{E}_{\text{Hxc}}^{\text{sr},\mu}[n_{\Phi}(\mathbf{r})], \tag{28}$$

and the minimizing single Slater determinant Φ^{μ} satisfies the Schrödinger-like equation:

$$\left(\hat{h} + \hat{W}_{\text{ee}}^{\text{lr},\mu} + \hat{\bar{V}}_{\text{Hxc}}^{\text{sr},\mu}[n_{\Phi^{\mu}}(\mathbf{r})] \right) |\Phi^{\mu}\rangle = \mathcal{E}_{\text{RSH}}^{\mu} |\Phi^{\mu}\rangle . \tag{29}$$

Equation (29) can be reformulated as a Kohn-Sham-like equation as the variational space of a single-Slater determinant reduces to the orbitals. Nonetheless, as the long-range interaction appears in Eq. (29), the long-range part of HF exchange appears in the Kohn-Sham-like equation which improves the long-range behavior with respect to usual hybrid approximations. Also, as the short-range exchange-correlation functional $\bar{E}_{\text{xc}}^{\text{sr},\mu}[n(\mathbf{r})]$ describes one- and two-electron effects induced by the short-range interaction, the usual local or semi-local approximations used in DFT are much more suited to describe these situations.

3. *Implementation within Quantum Package*

Quantum Package contains all necessary integrals to perform RSDFT calculations, including the long-range interaction integrals

$$(pq|w_{\text{ee}}^{\text{lr},\mu}|rs) = \iint \phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1) \frac{\text{erf}(\mu r_{12})}{r_{12}} \phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2, \tag{30}$$

and Hartree-exchange-correlation energies and potentials derived from the short-range version of the local-density approximation (LDA)[97] and a short-range generalized-gradient approximation (GGA) based on the Perdew-Burke-Ernzerhof (PBE) functional.[98]

All numerical integrals are performed using the standard Becke quadrature grid[99] associated with the improved radial grids of Mura et *al.*[100] More evolved schemes based on the RSDFT formalism can be found as external plugins (see, for example, https://gitlab.com/eginer/qp_plugins_eginer).

III. IMPLEMENTATION

In this section, we give an overview of the implementation of the various methods present in Quantum Package. The implementation of the crucial algorithms is explained in details in the PhD thesis of Dr Y. Garniron.[101]

A. Determinant-driven computation of the matrix elements

For performance sake, it is vital that some basic operations are done efficiently and, notably, the computation of the Hamiltonian matrix elements. This raises some questions about the data structures chosen to represent the two-electron integrals and determinants, as well as their consequences from an algorithmic point of view. This section is going to address these questions by going through the basic concepts of our determinant-driven approach.

1. *Storage of the two-electron integrals*

In Quantum Package, the two-electron integrals are kept in memory because they require a fast random access. Considering the large number of two-electron integrals, a hash table is the natural choice allowing the storage of only non-zero values with a data retrieval in near constant time.[102] However, standard hashing algorithms tend to shuffle data to limit the probability of collisions. Here, we favor data locality using the hash function given in Algorithm 1. This hash function i) returns the same value for all keys related by permutation symmetry, ii) keeps some locality in the storage of data, and iii) can be evaluated in 10 CPU cycles (estimated with MAQAO[103]) if the integer divisions by two are replaced by right bit shift instructions.

```
Function HASH($i,j,k,l$): /* Hash function for
  two-electron integrals                         */
    Data: $i,j,k,l$ are the orbital indices
    Result: The corresponding hash
    $p \leftarrow \min(i,k)$ ;
    $r \leftarrow \max(i,k)$ ;
    $t \leftarrow p + r(r-1)/2$ ;
    $q \leftarrow \min(j,l)$ ;
    $s \leftarrow \max(j,l)$ ;
    $u \leftarrow q + s(s-1)/2$ ;
    $v \leftarrow \min(t,u)$ ;
    $w \leftarrow \max(t,u)$ ;
    return $v + w(w-1)/2$ ;
```

Algorithm 1: Hash function that maps any orbital quartet (i,j,k,l) related by permutation symmetry to a unique integer.

The hash table is such that each bucket can potentially store 2^{15} consecutive key-value pairs. The 15 least significant bits of the hash value are removed to give the bucket index $[i_{\text{bucket}} = \lfloor \text{hash}(i,j,k,l)/2^{15} \rfloor]$, and only those 15 bits need

TABLE I. Time to access integrals (in nanoseconds/integral) with different access patterns. The time to generate random numbers (measured as 67 ns/integral) was not counted in the random access results.

Access	Array	Hash table
i,j,k,l	9.72	125.79
i,j,l,k	9.72	120.64
i,k,j,l	10.29	144.65
l,k,j,i	88.62	125.79
l,k,i,j	88.62	120.64
Random	170.00	370.00

to be stored in the bucket for the key storage [hash(i,j,k,l) mod 2^{16}]. Hence, the key storage only requires two bytes per key, and they are sorted in increasing order, enabling a binary search within the bucket. The key search is always fast since the binary search is bounded by 15 misses and the maximum size of the key array is 64 kiB, the typical size of the L1 cache.

The efficiency of the storage as a hash table was measured on a dual socket Intel Xeon E5-2680 v2@2.80GHz processor, taking the water molecule with the cc-pVQZ basis set (115 MOs). The time to access all the integrals was measured by looping over the entire set of ERIs using different loop orderings. The results are given in Table I, the reference being the storage as a plain four-dimensional array.

In the array storage, the value of 170 ns/integral in the random access case is typical of the latency to fetch a value in the RAM modules, telling that the requested integral is almost never present in any level of cache. When the data is accessed with a stride of one (i,j,l,k storage), the hierarchical architecture of the cache levels accelerates the access by a factor of 18, down to 9.71 ns/integral, corresponding mostly to the overhead of the function call, the retrieval of the data being negligible.

With the hash table, the random access is only 2.18 times slower than the random access in the array. Indeed, two random accesses are required: one for the first element of the key bucket to do the search, and one for the value of the integral. The remaining time corresponds to the binary search. The results show that data locality is exploited: when the access is done with a regular access pattern, the data is fetched roughly 3 times faster than using a random access, giving a latency below the latency of a random access in the array.

A CIPSI calculation was run once with the array storage, and once with the hash table storage. With the hash storage, the total wall clock time was increased only by a factor of two. To accelerate the access to the most frequently used integrals and reduce this overhead, we have implemented a software cache. All the integrals involving the 128 MOs closest to the Fermi level are copied in a dense array of 128^4 elements (2 GiB), and benefit from the fastest possible access.

2. *Internal representation of determinants*

Determinants can be conveniently written as a string of creation operators applied to the vacuum state $|\rangle$, e.g., $a_i^\dagger a_j^\dagger a_k^\dagger|\rangle = |I\rangle$. Because of the fermionic nature of electrons, a permutation of two contiguous creation operators results in a sign change $a_j^\dagger a_i^\dagger = -a_i^\dagger a_j^\dagger$, which makes their ordering relevant, e.g., $a_j^\dagger a_i^\dagger a_k^\dagger|\rangle = -|I\rangle$. A determinant can be broken down into two pieces of information: i) a set of creation operators corresponding to the set of occupied spinorbitals in the determinant, and ii) an ordering of the creation operators responsible for the sign of the determinant, known as *phase factor*. Once an ordering operator $\hat{\mathcal{O}}$ is chosen and applied to all determinants, the phase factor may simply be included in the CI coefficient.

The determinants are built using the following order: i) spin-up (↑) spinorbitals are placed before spin-down (↓) spinorbitals, as in the Waller-Hartree double determinant representation[104] $\hat{\mathcal{O}}\,|I\rangle = \hat{I}|\rangle = \hat{I}_\uparrow \hat{I}_\downarrow|\rangle$, and ii) within each operator $\hat{I}_\uparrow$ and $\hat{I}_\downarrow$, the creation operators are sorted by increasing indices. For instance, let us consider the determinant $|J\rangle = a_j^\dagger a_k^\dagger a_{\bar{i}}^\dagger a_i^\dagger|\rangle$ built from the set of spinorbitals $\{i_\uparrow, j_\uparrow, k_\uparrow, i_\downarrow\}$ with $i < j < k$. If we happen to encounter such a determinant, our choice of representation imposes to consider its re-ordered expression $\hat{\mathcal{O}}\,|J\rangle = -a_i^\dagger a_j^\dagger a_k^\dagger a_{\bar{i}}^\dagger|\rangle = -|J\rangle$, and the phase factor must be handled.

The indices of the creation operators (or equivalently the spinorbital occupations), are stored using the so-called *bitstring* encoding. A bitstring is an array of bits; typically, the 64-bit binary representation of an integer is a bitstring of size 64. Quite simply, the idea is to map each spinorbital to a single bit with value set to its occupation number. In other words, 0 and 1 are associated with the *unoccupied* and *occupied* states, respectively.

For simplicity and performance considerations, the occupations of the spin-up and spin-down spinorbitals are stored in different bitstrings, rather than interleaved or otherwise merged in the same one. This allows to straightforwardly map orbital index p to bit index $p-1$ (orbitals are usually indexed from 1, while bits are indexed from 0). This makes the representation of a determinant a tuple of two bitstrings, associated with respectively spin-up and spin-down orbitals.

The storage required for a single determinant is, in principle, one bit per spinorbital, or $2 \times N_{\text{orb}}$ bits. However, because CPUs are designed to handle efficiently 64-bit integers, each spin part is stored as an array of 64-bit integers, the unused space being padded with zeros. The actual storage needed for a determinant is $2 \times 64 \times N_{\text{int}}$ bits, where $N_{\text{int}} = \lfloor (N_{\text{orb}} - 1)/64 \rfloor + 1$ is the number of 64-bit integers needed to store one spin part.

Taking advantage of low-level hardware instructions,[105] we are able, given two arbitrary determinants $|I\rangle$ and $|J\rangle$, to find with a minimal cost the excitation operator $\hat{T}$ such that $|J\rangle = \hat{T}\,|I\rangle$. This is a necessary step to obtain the (i,j,k,l) indices of the two-electron integral(s) involved in the Hamiltonian matrix element between $|I\rangle$ and $|J\rangle$. Then, fetching the values of the integrals can be done quickly using the hash table presented in Sec. III A 1.

Because the data structure used to store determinants implies an ordering of the MOs, we also need to compute a phase factor. Here, we propose an algorithm to perform efficiently the computation of the phase factor. For a determinant $|I\rangle$ that is going to be used repeatedly for phase calculations, we introduce a *phase mask* represented as a bitstring:

$$P_I[i] = 1 \wedge \sum_{k=0}^{i} I[k], \tag{31}$$

where $\wedge$ denotes the *and* bitwise operation, and $I[k]$ is the kth bit of bitstring I, corresponding to the $(k+1)$th spinorbital of determinant $|I\rangle$ (remember that the orbital indices start at 1 and the bit indices start at 0). In other words, the ith bit of the phase mask is set to 1 if the number of electrons occupying the $i+1$ lowest spinorbitals is odd, and 0 otherwise. When an electron of determinant $|I\rangle$ is excited from orbital h to p, the associated phase factor is

$$\begin{cases} +(-1)^{P_I[h-1] \oplus P_I[p-1]}, & \text{if } p > h, \\ -(-1)^{P_I[h-1] \oplus P_I[p-1]}, & \text{if } h > p, \end{cases} \tag{32}$$

where $\oplus$ denotes the exclusive or (*xor*) operation. So if the phase mask is available, the computation of the phase factor only takes a few CPU cycles. Another important aspect is to create efficiently the phase masks. We propose Algorithm 2, which computes it in a logarithmic time with respect to the number of MOs, taking advantage of the associativity of the exclusive *or* operator.

```
Function PhasemaskOfDet($I$):
    Data: $I$ : 64-bit string representation of $|I\rangle$
    Result: $P$ : phase mask associated with $|I\rangle$, as a 64-bit
            string.
    for $\sigma \in \{\uparrow,\downarrow\}$ do
        $r \leftarrow 0$ ;
        for $i \leftarrow 0, N_{int} - 1$ do
            $P_\sigma[i] \leftarrow I_\sigma[i] \oplus (I_\sigma[i] \ll 1)$ ;
            for $d \leftarrow 0, 5$ do
                $P_\sigma[i] \leftarrow P_\sigma[i] \oplus (P_\sigma[i] \ll (1 \ll d))$ ;
            end
            $P_\sigma[i] \leftarrow P_\sigma[i] \oplus r$ ;
            if $(\|I_\sigma[i]\| \wedge 1) = 1$ then
                $r \leftarrow \neg r$ ;
            end
        end
    end
    return $P$ ;
───────────────────────────────────────────
$\|I\|$          : number of bits set to 1 in $I$ (popcnt),
$\wedge$         : bitwise and,
$\oplus$         : bitwise xor,
$(I \ll k)$      : shift $I$ by $k$ bits to the left,
$\neg$           : bitwise negation.
```

Algorithm 2: Function that returns a phase mask as a bitstring.

B. Davidson diagonalization

1. Generalities

Finding the lowest root(s) of the Hamiltonian is a necessary step in CI methods. Standard diagonalization algorithms scale as $\mathcal{O}(N_{\text{det}}^3)$ and $\mathcal{O}(N_{\text{det}}^2)$ in terms of computation and storage, respectively. Hence, their cost is prohibitive as N_{det} is usually, at least, of the order of few millions. Fortunately, not all the spectrum of $\hat{H}$ is required: only the first few lowest eigenstates are of interest. The Davidson diagonalization[106–110] is an iterative algorithm which aims at extracting the first N_{states} lowest eigenstates of a large matrix. This algorithm reduces the cost of both the computation and storage to $\mathcal{O}(N_{\text{states}} N_{\text{det}}^2)$ and $\mathcal{O}(N_{\text{states}} N_{\text{det}})$, respectively. It is presented as Algorithm 3.

```
Function DAVIDSON_DIAG($N_{states}$, $\mathbf{U}$):
    Data: $N_{\text{states}}$: Number of requested states
    Data: $N_{\text{det}}$: Number of determinants
    Data: $\mathbf{U}$: Guess vectors, $N_{\text{det}} \times N_{\text{states}}$
    Result: $N_{\text{states}}$ lowest eigenvalues eigenvectors of $\mathbf{H}$
    converged $\leftarrow$ FALSE ;
    while $\neg$converged do
        Gram-Schmidt orthonormalization of $\mathbf{U}$ ;
        $\mathbf{W} \leftarrow \mathbf{H}\,\mathbf{U}$ ;
        $\mathbf{h} \leftarrow \mathbf{U}^\dagger\,\mathbf{W}$ ;
        Diagonalize $\mathbf{h}$ : eigenvalues $E$ and eigenvectors $\mathbf{y}$ ;
        $\mathbf{U}' \leftarrow \mathbf{U}\,\mathbf{y}$ ;
        $\mathbf{W}' \leftarrow \mathbf{W}\,\mathbf{y}$ ;
        for $k \leftarrow 1, N_{states}$ do
            for $i \leftarrow 1, N_{det}$ do
                $\mathbf{R}_{ik} \leftarrow \frac{E_k \mathbf{U}'_{ik} - \mathbf{W}'_{ik}}{\mathbf{H}_{ii} - E_k}$ ;
            end
        end
        converged $\leftarrow \|\mathbf{R}\| < \epsilon$ ;
        $\mathbf{U} \leftarrow [\mathbf{U}, \mathbf{R}]$ ;
    end
    return $\mathbf{U}$;
```

Algorithm 3: Davidson diagonalization procedure.

Within Quantum Package, the Davidson diagonalization algorithm is implemented in its multi-state version. Algorithmically, the expensive part of the Davidson diagonalization is the computation of the matrix product $\mathbf{H}\,\mathbf{U}$. As mentioned above (see Sec. II), two determinants $|I\rangle$ and $|J\rangle$ are connected via $\mathbf{H}$ (i.e. $\langle I|\hat{H}|J\rangle \neq 0$) only if they differ by no more than two spinorbitals. Therefore, the number of non-zero elements per row in $\mathbf{H}$ is equal to the number of single and double excitation operators, namely $\mathcal{O}(N_\uparrow^2 (N_{\text{orb}} - N_\uparrow)^2)$. As $\mathbf{H}$ is symmetric, the number of non-zero elements per column is identical. This makes $\mathbf{H}$ very sparse. However, for large basis sets, the whole matrix may still not fit in a single node memory, as the number of non-zero entries to be stored is of the order of $N_{\text{det}} N_\uparrow^2 (N_{\text{orb}} - N_\uparrow)^2$. One possibility would be to distribute the storage of $\mathbf{H}$ among multiple compute nodes, and use a distributed library such as PBLAS[111] to perform the matrix-vector operations. Another approach is to use a so-called *direct* algorithm, where the matrix elements are computed *on the fly*, and this is the approach we have chosen in Quantum Package. This effectively means iterating over all pairs of determinants $|I\rangle$ and $|J\rangle$, checking whether $|I\rangle$ and $|J\rangle$ are connected by $\mathbf{H}$ and if they are, accessing the corresponding integral(s) and computing the phase factor. Even though it is possible to compute the excitation degree between two determinants very efficiently,[105] the number of such computations scales as N_{det}^2, which becomes rapidly prohibitively high. To get an efficient determinant-driven implementation it is mandatory to filter out all pairs of determinants that are not connected by $\mathbf{H}$, and iterate only over connected pairs. To reach this goal, we have implemented an algorithm similar to the *Direct Selected Configuration Interaction Using Strings* (DISCIUS) algorithm.[56]

The determinants of the internal space are re-ordered in linear time as explained in Ref. 112, such that the wave function can be expressed as

$$|\Psi^{(0)}\rangle = \sum_{I}^{N_{\text{det}}^{\uparrow}} \sum_{J}^{N_{\text{det}}^{\downarrow}} C_{IJ} \, |I_{\uparrow} J_{\downarrow}\rangle \,, \tag{33}$$

where we take advantage of the Waller-Hartree double determinant representation.[104]

Moving along a row or a column of $\mathbf{C}$ keeps the spin-up or spin-down determinants fixed, respectively. For a given determinant, finding the entire list of same-spin single and double excitations can be performed in $\mathcal{O}(N_{\text{det}}^{\uparrow}) = \mathcal{O}(N_{\text{det}}^{\downarrow}) = \mathcal{O}(\sqrt{N_{\text{det}}})$, while finding the opposite-spin double excitations is done via a two-step procedure. First, we look for all the spin-up single excitations. Then, starting from this list of spin-up single excitations, we search for the spin-down single excitation such that the resulting opposite-spin doubly-excited determinant belongs to $\Psi^{(0)}$. Hence, the formal scaling is reduced to $\mathcal{O}(N_{\text{det}}^{3/2})$. It could be further reduced to $\mathcal{O}(N_{\text{det}})$ at the cost of storing the list of all singly- and doubly-excited determinants for each spin-up and spin-down determinant, but we preferred not to follow this path in order to reduce the memory footprint as much as possible.

C. CIPSI selection and PT2 energy

1. The basic algorithm

The largest amount of work for this second version of Quantum Package has been devoted to the improvement of the CIPSI algorithm implementation.[113] As briefly described in Sec. II, this is an iterative selection algorithm, where determinants are added to the reference wave function according to a perturbative criterion.

The nth CIPSI iteration can be described as follows:

1. The zeroth-order (reference or variational) wave function

$$|\Psi^{(0)}\rangle = \sum_{I \in \mathcal{I}_n} c_I \, |I\rangle \tag{34}$$

is defined over a set of determinants $\mathcal{I}_n$ — characterized as *internal* determinants — from which the lowest eigenvector of $\hat{H}$ are obtained.

2. For all *external* determinants $|\alpha\rangle \notin \mathcal{I}_n$ but connected to $\mathcal{I}_n$, i.e., $\langle \Psi^{(0)} | \hat{H} | \alpha \rangle \neq 0$, we compute the individual perturbative contribution $e_{\alpha}^{(2)}$ given by Eq. (10). This set of external determinants is labeled $\mathcal{A}_n$.

3. Summing the contributions of all the external determinants $\alpha \in \mathcal{A}_n$ gives the second-order perturbative correction provided by Eq. (12) and the FCI energy can be estimated as $E_{\text{FCI}} \approx E^{(0)} + E^{(2)}$.

4. We extract $|\alpha^{\star}\rangle \in \mathcal{A}_n^{\star}$, the subset of determinants $|\alpha\rangle \in \mathcal{A}_n$ with the largest contributions $e_{\alpha}^{(2)}$, and add them to the variational space $\mathcal{I}_{n+1} = \mathcal{I}_n \cup \mathcal{A}_n^{\star}$.

5. Iterate until the desired convergence has been reached.

2. Implementation details

There are multiple ways to compute the $e_{\alpha}^{(2)}$'s. One way is to loop over pairs of internal determinants $|I\rangle$ and $|J\rangle$, generate the list of external determinants $\{|\alpha\rangle\}$ connecting $|I\rangle$ and $|J\rangle$ and increment the corresponding values $e_{\alpha}^{(2)}$ stored in a hash table. Using a hash table to store in memory a list of $|\alpha\rangle$'s without duplicates and their contributions $e_{\alpha}^{(2)}$ is obviously not a reasonable choice since the total number of $|\alpha\rangle$'s scales as $\mathcal{O}(N_{\text{det}} N_{\uparrow}^2 (N_{\text{orb}} - N_{\uparrow})^2)$. To keep the memory growth in check, we must design a function that can build a stream of unique external determinants, compute their contribution $e_{\alpha}^{(2)}$ and retain in memory only the few most significant pairs $(|\alpha\rangle, e_{\alpha}^{(2)})$.

In Quantum Package, we build the stream of unique external determinants as follows. We loop over the list of internal determinants (the *generators*) sorted by decreasing c_I^2. For each generator $|I\rangle$, we generate all the singly- and doubly-excited determinants $\{|\alpha\rangle\}$, removing from this set the internal determinants and the determinants connected to any other generator $|J\rangle$ such that $J < I$. This guarantees that the $|\alpha\rangle$'s are considered only once, without any additional memory requirement.

For each generator $|I\rangle$, before generating its set of $|\alpha\rangle$'s, we pre-compute the diagonal of the Fock matrix associated with $|I\rangle$. This enables to compute the diagonal elements $\langle \alpha | \hat{H} | \alpha \rangle$ involved in Eq. (10) for a few flops.[114] The computation of $\langle \Psi^{(0)} | \hat{H} | \alpha \rangle = \sum_J c_J \langle J | \hat{H} | \alpha \rangle$ is more challenging than the diagonal term since, at first sight, it appears to involve the N_{det} internal determinants. Fortunately, most of the terms amongst this sum vanish due to Slater-Condon's rules. Indeed, we know that the terms where $|J\rangle$ is more than doubly excited with respect to $|\alpha\rangle$ vanish, and these correspond to the determinants $|J\rangle$ which are more than quadruply excited with respect to $|I\rangle$.[114] To compute efficiently $\langle \Psi^{(0)} | \hat{H} | \alpha \rangle$, for each generator $|I\rangle$, we create a filtered wave function $|\Psi_I^{(0)}\rangle$ by projecting $|\Psi^{(0)}\rangle$ on a subset $\mathcal{J}_I$ of internal determinants $\{|J\rangle\}$ where $\langle J|\hat{H}|\alpha\rangle$ is possibly non-zero. This yields $\langle\Psi^{(0)}|\hat{H}|\alpha\rangle = \langle\Psi_I^{(0)}|\hat{H}|\alpha\rangle$, where $\Psi_I^{(0)}$ is a much smaller determinant expansion than $\Psi^{(0)}$. In addition, as we have defined the $|\alpha\rangle$'s in such a way that they do not interact with $|J\rangle$ when $J < I$, all these $|J\rangle$'s can also be excluded from $\mathcal{J}_I$. This pruning process yielding to $|\Psi_I^{(0)}\rangle$ will be referred to as the *coarse-grained* filtering. A *fine-grained* filtering of $|\Psi_I^{(0)}\rangle$ is performed in a second stage to reduce even more the number of determinants, as we shall explain later.

To make the coarse-grained filtering efficient, we first filter out the determinants that are more than quadruply excited in the spin-up and spin-down sectors separately. Using the representation shown in Eq. (33), this filtering does not need to run through all the internal determinants and scales as $\mathcal{O}(N_{\text{det}}^{\uparrow}) = \mathcal{O}(\sqrt{N_{\text{det}}})$. It is important to notice that, at this stage, the size of $\mathcal{J}_I$ is bounded by the number of possible quadruple excitations in both spin sectors, and does not scale any more as $\mathcal{O}(N_{\text{det}})$. Next, we remove the determinants that are i) quadruply excited in one spin sector and excited in the other spin sector, ii) triply excited in one spin sector and more than singly excited in the other spin sector, and iii) all the determinants that are doubly excited in one spin sector and more than doubly excited in the other spin sector.

The external determinant contributions are computed in batches. A batch I_{pq} is defined by a doubly-ionized generator $|I_{pq}\rangle = a_p a_q\,|I\rangle$. When a batch is created, the fine-grained filtering step is applied to $\mathcal{J}_I$ to produce $\mathcal{J}_{I_{pq}}$ and $\Psi_{I_{pq}}^{(0)}$, such that $\langle\Psi_{I_{pq}}^{(0)}|\hat{H}|\alpha\rangle = \langle\Psi_I^{(0)}|\hat{H}|\alpha\rangle$. The efficiency of the filtering steps is presented in Sec. VI B.

Each external determinant produced in the batch I_{pq} is characterized by two indices r and s with $\hat{O}a_r^\dagger a_s^\dagger a_p a_q\,|I\rangle = |I_{pq}^{rs}\rangle$. The contribution associated with each determinant of a given batch will be computed incrementally in a two-dimensional array $A(r,s)$ as follows. A first loop is performed over all the determinants $|J\rangle$ belonging to the filtered internal space $\mathcal{J}_{I_{pq}}$. Comparing $|J\rangle$ to $|I_{pq}\rangle$ allows to quickly identify if $|J\rangle$ will be present in the list of external determinants, and consequently tag the corresponding cell $A(r,s)$ as *banned*. Banned cells will not be considered for the computation of $e_\alpha^{(2)}$ nor the determinant selection, as they correspond to determinants already belonging to the internal space. A second loop over all the $|J\rangle \in \mathcal{J}_{I_{pq}}$ is then performed. During this loop, all the (r,s) pairs where $|I_{pq}^{rs}\rangle$ is connected to $|J\rangle$ are generated, and the corresponding cells $A(r,s)$ are incremented with $c_J\,\langle J|\hat{H}|I_{pq}^{rs}\rangle$. After this second loop, $A(r,s) = \langle\Psi|\hat{H}|I_{pq}^{rs}\rangle$ and all the contributions $e_\alpha^{(2)}$ of the batch can be obtained using $A(r,s)$. The running value of $E^{(2)}$ is then incremented, and the N_{det} most significant determinants are kept in an array sorted by decreasing $|e_\alpha^{(2)}|$.

3. *New stochastic selection*

In the past, CIPSI calculations were only possible in practice thanks to approximations. The first approximation restricts the set $\mathcal{A}_n$ by defining a set of *generators*. Indeed, it is very unlikely that $|\alpha\rangle$ will be selected if it is not connected to any $|I\rangle$ with a large coefficient, so only the determinants with the largest coefficients are generators. A second approximation defines a set of *selectors* in order to reduce the cost of $e_\alpha^{(2)}$ by removing the determinants with the smallest coefficients in the expression of $\Psi^{(0)}$ in $E^{(2)}$. This approximate scheme was introduced in the 80's and is known as *three-class CIPSI*.[24] The downside of these approximations is that the calculation is biased and, consequently, does not strictly converge to the FCI limit. Moreover, similar to the initiator approximation in FCIQMC,[8] this scheme suffers from a size-consistency issue.[115] The stochastic selection that we describe in this section (asymptotically) cures this problem, as there is *no threshold* on the wave function: if the calculation is run long enough, the unbiased FCI solution is obtained.

Recently, some of us developed a hybrid deterministic/stochastic algorithm for the computation of $E^{(2)}$.[116] The main idea is to rewrite the expression of

$$E^{(2)} = \sum_\alpha c_\alpha\,\langle\Psi^{(0)}|\hat{H}|\alpha\rangle \tag{35}$$

into elementary contributions labeled by the determinants of the internal space:

$$E^{(2)} = \sum_I \sum_{\alpha\in\mathcal{A}_I} c_\alpha\,\langle\Psi^{(0)}|\hat{H}|\alpha\rangle = \sum_I \varepsilon_I, \tag{36}$$

where

$$c_\alpha = \frac{\langle\Psi^{(0)}|\hat{H}|\alpha\rangle}{E^{(0)} - \langle\alpha|\hat{H}|\alpha\rangle} \tag{37}$$

is the corresponding coefficient estimated via second-order perturbation theory, and $\mathcal{A}_I$ is the subset of determinants $|\alpha\rangle$ connected to $|I\rangle$ by $\hat{H}$ such that $|\alpha\rangle \notin \cup_{K<I}\mathcal{A}_K$. The sum is decomposed into a stochastic and a deterministic contribution

$$E^{(2)} = \sum_{J\in\mathcal{D}} \varepsilon_J + \sum_{K\in\mathcal{S}} \varepsilon_K, \tag{38}$$

where $\mathcal{D}$ and $\mathcal{S}$ are the sets of determinants included in the deterministic and stochastic components, respectively.

The $|I\rangle$'s are sorted by decreasing c_I^2, and two processes are used simultaneously to compute the contributions ε_I. The first process is stochastic and $|I\rangle$ is drawn according to c_I^2. When a given ε_I has been computed once, its contribution is stored such that if $|I\rangle$ is drawn again later the contribution does not need to be recomputed. The only update is to increment the number of times it has been drawn for the Monte Carlo statistics. In parallel, a deterministic process is run, forcing to compute the contribution ε_I with the smallest index which has yet to be computed. The deterministic component is chosen as the first contiguous set of ε_I. Hence, the computation of $E^{(2)}$

is unbiased, and the exact deterministic value can be obtained in a finite time if the calculation is run long enough. The stochastic part is only a convergence accelerator providing a reliable error bar. The computation of $E^{(2)}$ is run with a default stopping criterion set to $|\delta E^{(2)}/E^{(2)}| = 0.002$, where $\delta E^{(2)}$ is the statistical error associated with $E^{(2)}$. We would like to stress that, thanks to the present semistochastic algorithm, the complete wave function is considered, and that no threshold is required. Consequently, size-consistency will be preserved if a size-consistent perturbation theory is applied.

While performing production runs, we have noticed that the computation of $E^{(2)}$ was faster than the CIPSI selection. Hence, we have slightly modified the routines computing $E^{(2)}$ such that the selection of determinants is performed alongside the computation of $E^{(2)}$. This new on-the-fly CIPSI selection performed during the stochastic PT2 calculation completely removes the conventional (deterministic) selection step, and the determinants are selected with no additional cost. We have observed that, numerically, the curves of the variational energy as a function of N_{det} obtained with either the deterministic or the stochastic selections are indistinguishable, so that the stochastic algorithm does not harm the selection's quality.

For the selection of multiple states, one PT2 calculation is run for each state and, as proposed by Angeli *et al.*,[117] the selection criterion is modified as

$$\tilde{e}_\alpha^{(2)} = \sum_k^{N_{\text{states}}} \frac{c_{\alpha k}}{\max_I |c_{Ik}|} \langle \Psi_k^{(0)} | \hat{H} | \alpha \rangle , \tag{39}$$

with

$$c_{\alpha k} = \frac{\langle \Psi_k^{(0)} | \hat{H} | \alpha \rangle}{\langle \Psi_k^{(0)} | \hat{H} | \Psi_k^{(0)} \rangle - \langle \alpha | \hat{H} | \alpha \rangle} . \tag{40}$$

This choice gives a balanced selection between states of different multi-configurational nature.

IV. SPIN-ADAPTED WAVE FUNCTIONS

Determinant-based sCI algorithms generate wave functions expressed in a truncated space of determinants. Obviously, the selection presented in the previous section does not enforce that $\hat{H}$ commutes with $\hat{S}^2$ in the truncated space. Hence, the eigenstates of $\hat{H}$ are usually not eigenvectors of $\hat{S}^2$, although the situation improves when the size of the internal space tends to be complete. A natural way to circumvent this problem is to work in the basis of *configuration state functions* (CSFs), but this representation makes the direct computation of the Hamiltonian less straightforward during the Davidson diagonalization.

Instead, we follow the same path as the MELD and SCIEL codes,[118–120] and identify all the spatial occupation patterns associated with the determinants.[121] We then generate all associated spin-flipped configurations, and add to the internal space all the missing determinants. This procedure ensures that $\hat{H}$ commutes with $\hat{S}^2$ in the truncated space, and spin-adapted states are obtained by the diagonalization of $\hat{H}$. In addition, we apply a penalty method in the diagonalization by modifying the Hamiltonian as[122]

$$\tilde{\mathbf{H}} = \mathbf{H} + \gamma \left(\mathbf{S}^2 - \mathbf{I} \langle S^2 \rangle_{\text{target}} \right)^2 , \tag{41}$$

where $\mathbf{I}$ is the identity matrix and γ is a fixed parameter set to 0.1 by default. This improves the convergence to the desired spin state, but also separates degenerate states with different spins, a situation that can potentially occurs with Rydberg states. In the Davidson algorithm, this requires the additional computation of $\mathbf{S}^2\,\mathbf{U}$, for which the cost is expected to be the same as the cost of $\mathbf{H}\,\mathbf{U}$ (see Algorithm 3). The cost of computing $\mathbf{H}\,\mathbf{U}$ and $\mathbf{S}^2\,\mathbf{U}$ is mostly due to the search of the connected pairs of determinants, namely the determinants $\langle I|$ and $|J\rangle$ for which $\langle I|\hat{H}|J\rangle$ and $\langle I|\hat{S}^2|J\rangle$ are not zero due to Slater-Condon's rules. We have modified the function computing $\mathbf{H}\,\mathbf{U}$ so that it also computes $\mathbf{S}^2\,\mathbf{U}$ at the same time. Hence, the search of connected pairs is done once for both operations and $\mathbf{S}^2\,\mathbf{U}$ is obtained with no extra computational cost.

Working with spin-adapted wave functions increases the size of the internal space by a factor usually between 2 and 3, but it is particularly important if one is willing to obtain excited states.[42–44,54] Therefore, the default in Quantum Package is to use spin-adapted wave functions.

V. PARALLELISM

In Quantum Package, multiple parallelism layers are implemented: a fine-grained layer to benefit from shared memory, an intermediate layer to benefit from fast communication within a group of nodes, and a coarse-grained layer to interconnect multiple groups of nodes. Fine-grained parallelism is performed with OpenMP[123] in almost every single routine. Then, to go beyond a single compute node, Quantum Package does not use the usual single program/multiple data (SPMD) paradigm. A task-based parallelism framework is implemented with the ZeroMQ library.[124] The single-node instance runs a compute process as well as a task server process, while helper programs can be spawned asynchronously on different (heterogeneous) machines to run a distributed calculation. The helper programs can connect via ZeroMQ to the task server at any time, and contribute to a running calculation. As the ZeroMQ library does not take full advantage of the low latency hardware present in HPC facilities, the helper programs are parallelized also with the message passing interface[125] (MPI) for fast communication among multiple client nodes, typically for fast broadcasting of large data structures.

Hence, we have 3 layers of parallelism in Quantum Package: OpenMP, MPI and ZeroMQ. This allows for an elastic management of resources: a running calculation taking too much time can be dynamically accelerated by plugging in more computing resources, by submitting more jobs in the queue or possibly in the cloud, i.e. outside of the HPC facility.

This scheme has the advantage that it is not necessary to wait for all the nodes to be free to start a calculation, and hence minimizes the waiting time in the batch queue. It also gives the possibility to use altogether different helper programs. For instance, one could use a specific GPU-accelerated helper program on a GPU node while CPU-only helpers run on the CPU-only partition of the cluster. It is also possible to write a helper program that helps only one PT2/selection step and then exit, allowing to gather resources after the PT2/selection has started, and freeing them for the following diagonalization step.

The current limitation of Quantum Package is the memory of the single-node instance. We have not yet considered the possibility to add more compute nodes to increase the available memory, but this can be done by transforming the main program into an MPI program using scattered data structures.

We now describe how the Davidson and PT2/selection steps are parallelized.

1. *Davidson diagonalization*

In the direct Davidson diagonalization method, the computational bottleneck is the matrix product $\mathbf{W} = \mathbf{H}\,\mathbf{U}$, and only this step needs to be distributed. The calculation is divided into independent tasks where each task builds a unique piece of $\mathbf{W}$ containing 40 000 consecutive determinants. Communicating the result of all the tasks scales as $\mathcal{O}(N_{\text{det}})$, independently of the number of parallel processes. On the other hand, $\mathbf{U}$ needs to be broadcast efficiently at the beginning of the calculation to each slave process.

The computation of a task is parallelized with OpenMP, looping in a way that guarantees a safe write access to $\mathbf{W}$, avoiding the need of a lock. When idle, a slave process requests a task to the ZeroMQ task server, computes the corresponding result and sends it to the *collector* thread of the master instance via ZeroMQ. As the OpenMP tasks are not guaranteed to be balanced, we have used a dynamic scheduling, with a chunk size of 64 elements. The reason for this chunk size is to force that multiple threads access to $\mathbf{W}$ at memory addresses far apart, avoiding the so-called *false sharing* performance degradation that occurs when multiple threads write simultaneously in the same cache line.[126] When the task is fully computed, the computed piece of $\mathbf{W}$ is sent back to the master process and a new task is requested, until the task queue is empty.

The $\mathbf{U}$ and $\mathbf{W}$ arrays are shared among threads, as well as all the large constant data needed for the calculation such as the ERIs. Sharing $\mathbf{U}$ also provides the benefit to reduce the amount of communication since $\mathbf{U}$ needs to be fetched only once for each node, independently of the number of cores. To make the broadcast of $\mathbf{U}$ efficient, the slave helper program is parallelized with MPI in a SPMD fashion, and each node runs a single MPI process. The $\mathbf{U}$ matrix is fetched from the ZeroMQ server by the process with rank zero, and then it is broadcast to the other slave processes within the same MPI job via MPI primitives. Then, each MPI process behaves independently and communicates via ZeroMQ with the task server, and with the master node which collects the results. A schematic view of the communication is presented in Fig 1.

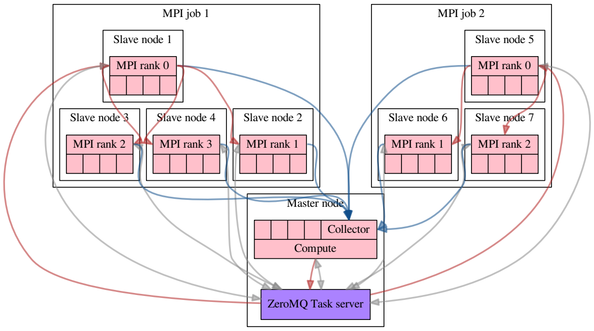


FIG. 1. Communications in the Davidson diagonalization for a calculation with a master node and two helper MPI jobs, each using 4 cores for the computation. Red arrows represent the broadcast of $\mathbf{U}$ starting from the compute process of the master node, gray arrows the exchange of ZeroMQ messages with the task server and blue arrows the collection of the results.

2. *CIPSI selection and PT2 energy*

In the computation of $E^{(2)}$ and the CIPSI selection, each task corresponds to the computation of one ε_J or ε_K in Eq. (38), together with the selection of the associated external determinants. To establish the list of tasks, the Monte Carlo sampling is pre-computed on the master node. We associate to each task the number of drawn Monte Carlo samples such that running averages can be computed when the results of the tasks have been received by the collector thread. When the convergence criterion is reached, the task queue is emptied and the collector waits for all the running tasks to terminate.

As opposed to the Davidson implementation where each task is parallelized with OpenMP, here each OpenMP thread handles independently a task computed on a single core. Hence, there are multiple ZeroMQ clients per node, typically one per core, requesting tasks to the task server and sending the results back to the collector thread (see Fig. 2). Here, all the OpenMP threads are completely independent during the whole selection, and this explains the pleasing scaling properties of our implementation, as shown in Sec. VI C. As in the Davidson distributed scheme, when the helper programs are run with MPI all the common data are communicated once from the ZeroMQ server to the rank-zero MPI process. Then, the data is broadcast to all the other processes with MPI primitives (there is one MPI process per node).

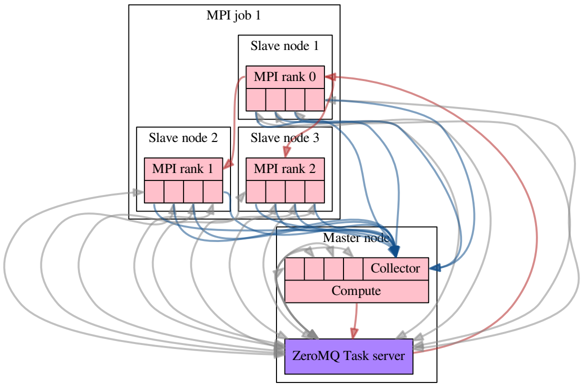


FIG. 2. Communications in the stochastic selection for a calculation with a master node and one helper MPI job, each using 4 cores for the computation. Red arrows represent the broadcast of the common data starting from the compute process of the master node, gray arrows the exchange of ZeroMQ messages with the task server and blue arrows the collection of the results.

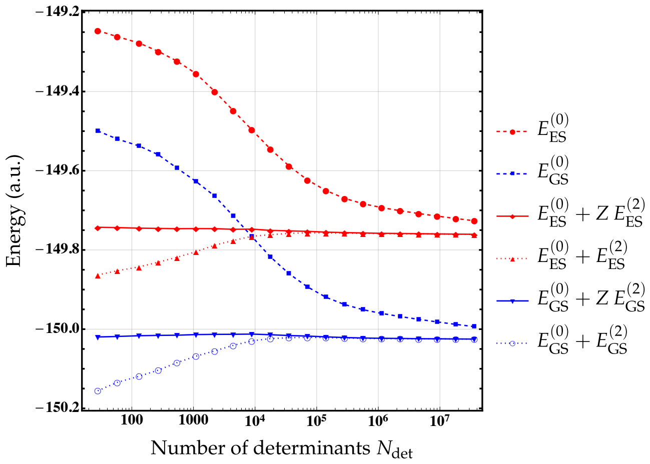


FIG. 3. Energy convergence of the ground state (GS, in blue) and excited state (ES, in red) of CN3 with respect to the number of determinants $N_{\rm det}$ in the reference space. The zeroth-order energy $E^{(0)}$ (dotted) , its second-order corrected value $E^{(0)} + E^{(2)}$ (dashed) as well as its renormalized version $E^{(0)} + ZE^{(2)}$ (solid) are represented. See Table II for raw data.

VI. RESULTS

A. Extrapolation

To illustrate the extrapolation procedure described in Sec. II E, we consider a cyanine dye[127] $H_2N-CH=NH_2{}^+$ (labeled as CN3 in the remaining) in both its ground state and first excited state.[45,128,129] The geometry is the equilibrium geometry of the ground state optimized at the PBE0/cc-pVQZ level.[129] The ground state is a closed shell, well described by a single reference, while the excited state is singly excited and requires, at least, two determinants to be properly modeled. The calculations were performed in the aug-cc-pVDZ basis set with state-averaged natural orbitals obtained from an initial CIPSI calculation. All the electrons were correlated, so the FCI space which is explored corresponds to a CAS(24,114) space. The reference excitation energy, obtained at the CC3/ANO-L-VQZP level is 7.18 eV[128] (see also Ref. 45).

In Fig. 3, we plot the energy convergence of the ground state (GS) and the excited state (ES) as a function of the number of determinants $N_{\rm det}$, with and without the second-order perturbative contribution. From the data gathered in Table II, one can see that, although $E^{(2)}$ is still large (roughly 0.02 a.u.), the sCI+PT2 and sCI+rPT2 excitation energies converge to a value of 7.20 eV compatible with the reference energy obtained in a larger basis set. We have also plotted the sCI+rPT2 energy given by $E^{(0)} + ZE^{(2)}$ (see Sec. II E 2) and we clearly see that this quantity converges much faster than the usual sCI+PT2 energy. Even for very small reference wave function, the energy gap between GS and ES is qualitatively correct. The graph of Fig. 4, which shows the zeroth-order energy $E^{(0)}$ as a function of the second-order energy $E^{(2)}$ (dotted lines) or its renormalization variant $Z\,E^{(2)}$ (solid lines), also indicates that it is practically much easier to extrapolate to the FCI limit using the rPT2 correction.

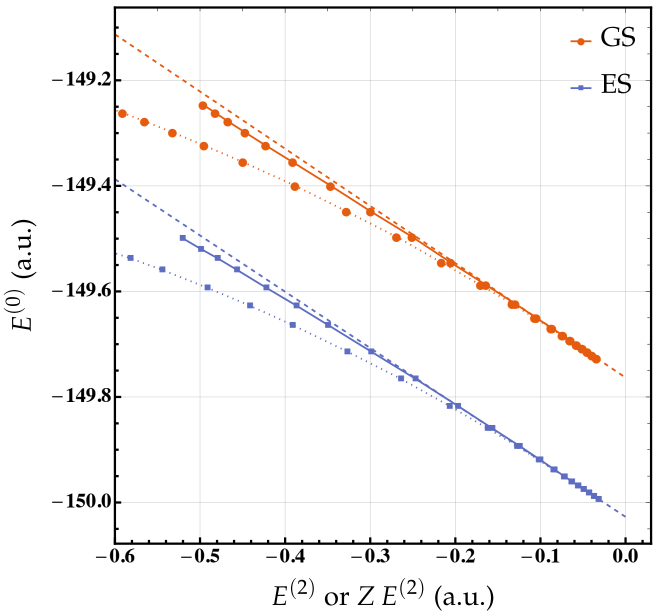


FIG. 4. Zeroth-order energy $E^{(0)}$ as a function of the second-order energy $E^{(2)}$ (dotted lines) or its renormalization variant $Z\,E^{(2)}$ (solid lines). A linear fit (dashed lines) of the last 6 points is also reported for comparison. See Table II for raw data.

TABLE II. Zeroth-order energy $E^{(0)}$, second-order perturbative correction $E^{(2)}$ and its renormalized version $ZE^{(2)}$ (in hartree) of CN3 for increasingly large wave functions. The excitation energy ΔE (in eV) is the energy difference between the ground state (GS) and the excited state (ES). The statistical error, corresponding to one standard deviation, is reported in parenthesis.

	$E^{(0)}$		$E^{(0)}+E^{(2)}$			$E^{(0)}+ZE^{(2)}$		
N_{det}	GS (a.u.)	ES (a.u.)	GS (a.u.)	ES (a.u.)	ΔE (eV)	GS (a.u.)	ES (a.u.)	ΔE (eV)
28	−149.499 574	−149.246 268	−150.155(1)	−149.863(1)	7.95(5)	−150.020(1)	−149.743(1)	7.54(5)
58	−149.519 908	−149.261 390	−150.134(1)	−149.853(1)	7.67(5)	−150.018(1)	−149.744(1)	7.48(5)
131	−149.537 424	−149.277 496	−150.118(1)	−149.842 7(9)	7.52(4)	−150.017(1)	−149.744 9(9)	7.39(4)
268	−149.559 465	−149.298 484	−150.103 5(9)	−149.830 8(9)	7.42(4)	−150.015 8(9)	−149.745 7(9)	7.35(4)
541	−149.593 434	−149.323 302	−150.084 5(8)	−149.818 6(8)	7.24(4)	−150.015 2(8)	−149.746 3(8)	7.32(4)
1 101	−149.627 202	−149.354 807	−150.068 3(8)	−149.804 5(8)	7.18(3)	−150.013 7(8)	−149.746 0(8)	7.28(3)
2 207	−149.663 850	−149.399 522	−150.054 9(7)	−149.787 9(7)	7.26(3)	−150.013 2(7)	−149.746 2(7)	7.27(3)
4 417	−149.714 222	−149.448 133	−150.040 9(6)	−149.776 2(6)	7.20(3)	−150.013 0(6)	−149.747 8(6)	7.22(3)
8 838	−149.765 886	−149.496 401	−150.029 6(5)	−149.765 5(5)	7.19(2)	−150.012 4(5)	−149.747 3(5)	7.21(2)
17 680	−149.817 301	−149.545 048	−150.023 9(4)	−149.761 5(4)	7.14(2)	−150.014 1(4)	−149.750 5(4)	7.17(2)
35 380	−149.859 737	−149.587 668	−150.021 6(3)	−149.758 2(3)	7.17(1)	−150.016 1(3)	−149.751 8(3)	7.19(1)
70 764	−149.893 273	−149.623 235	−150.020 7(2)	−149.756 6(3)	7.18(1)	−150.017 4(2)	−149.753 0(3)	7.19(1)
141 545	−149.919 463	−149.650 109	−150.021 4(2)	−149.757 2(2)	7.189(8)	−150.019 4(2)	−149.755 0(2)	7.196(8)
283 108	−149.937 839	−149.669 735	−150.022 4(2)	−149.757 6(2)	7.206(7)	−150.021 1(2)	−149.756 2(2)	7.209(7)
566 226	−149.950 918	−149.683 278	−150.023 3(1)	−149.758 0(1)	7.217(6)	−150.022 3(1)	−149.757 0(1)	7.219(6)
1 132 520	−149.960 276	−149.693 053	−150.023 8(1)	−149.758 8(1)	7.212(5)	−150.023 1(1)	−149.758 0(1)	7.214(5)
2 264 948	−149.968 203	−149.700 907	−150.024 0(1)	−149.759 0(1)	7.211(4)	−150.023 5(1)	−149.758 4(1)	7.212(4)
4 529 574	−149.975 230	−149.708 061	−150.024 5(1)	−149.759 4(1)	7.215(4)	−150.024 1(1)	−149.758 9(1)	7.216(4)
9 057 914	−149.981 770	−149.714 526	−150.024 63(9)	−149.759 81(8)	7.206(3)	−150.024 34(9)	−149.759 48(8)	7.207(3)
18 110 742	−149.987 928	−149.720 648	−150.024 95(7)	−149.760 25(8)	7.203(3)	−150.024 74(7)	−149.760 00(8)	7.204(3)
36 146 730	−149.993 593	−149.726 253	−150.025 27(6)	−149.760 65(7)	7.198(3)	−150.025 02(6)	−149.760 47(7)	7.198(3)

B. Efficiency of the filtering

Figure 5 shows the number of determinants retained in $\Psi_I^{(0)}$ or $\Psi_{I_{pq}}^{(0)}$ after filtering out disconnected determinants of the ground state of the CN3 molecule with 935 522 determinants (see Sec. III C 2). This example shows that, starting from $\Psi^{(0)}$, the coarse-grained process which consists in removing the determinants more than quadruply excited with respect to the generator $|I\rangle$ produces wave functions $\Psi_I^{(0)}$ with a typical size of 120 000 determinants, a reduction by a factor 8. Then, starting from $\Psi_I^{(0)}$, the fine-grained filtering, specific to the batch generating $\Psi_{I_{pq}}^{(0)}$, reduces even more the number of determinants (by a factor 3), down to a typical size of 40 000 determinants, which represents only 4% of the total wave function $\Psi^{(0)}$.

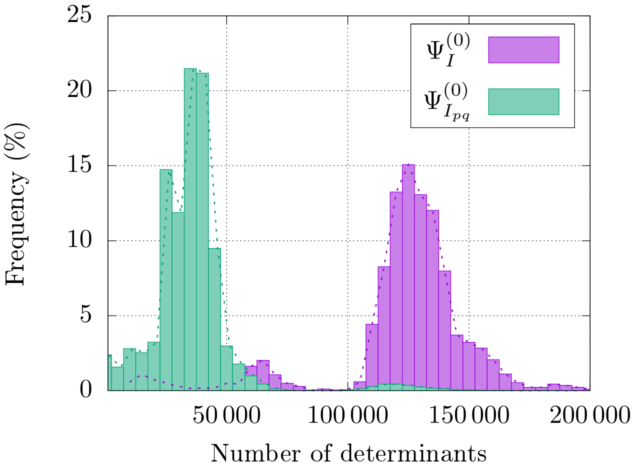


FIG. 5. Histograms representing the number of determinants remaining after the coarse-grained (purple) and fine-grained (green) filtering processes applied to the ground state of the CN3 molecule with $N_{\text{det}} = 935\,522$.

C. Speedup

In this Section, we discuss the parallel efficiency of the algorithms implemented in Quantum Package. The system we chose for these numerical experiments is the benzene molecule C_6H_6 for which we have performed sCI calculations with the 6-31G* basis set. The frozen-core approximation has been applied and the FCI space that we explore is a CAS(30,90). The measurements were made on GENCI's Irene supercomputer. Each Irene's node is a dual-socket Intel(R) Xeon(R) Platinum 8168 CPU@2.70GHz with 192GiB of RAM, with a total of 48 physical CPU cores. Parallel speedup curves are made up to 12 288 cores (i.e. 256 nodes) for i) a single iteration of the Davidson diagonalization, and ii) the hybrid semistochastic computation of $E^{(2)}$ (which includes the CIPSI selection). The speedup reference corresponds to the single node calculation (48 cores).

First, we measure the time required to perform a single Davidson iteration as a function of the number of CPU cores for the two largest wave functions ($N_{\text{det}} = 25 \times 10^6$ and 100×10^6). The timings are reported in Table III while the parallel speedup curve is represented in Fig. 6. The parallel efficiency increases together with N_{det}, as shown in Fig. 6. For the largest wave function, a parallel efficiency of 66% is obtained on 192 nodes (i.e. 9216 cores). We note that the speedup reaches a plateau at 3 072 cores (64 nodes) for $N_{\text{det}} = 25 \times 10^6$. For this wave function, there are 625 tasks computing each 40 000 rows of $\mathbf{W}$. When the number of nodes reaches 64, the number of tasks is too small for the load to be balanced

TABLE III. Wall-clock time (in seconds) to perform a single Davidson iteration and a second-order correction $E^{(2)}$ calculation (which also includes the CIPSI selection) with an increasing number of 48-core compute nodes N_{nodes}. The statistical error obtained on $E^{(2)}$, defining the stopping criterion, is 0.17×10^{-3} a.u.

N_{nodes}	Wall-clock time (in seconds)		
	Davidson for	Davidson for	PT2/selection
	$N_{\text{det}} = 25 \times 10^6$	$N_{\text{det}} = 25 \times 10^6$	$N_{\text{det}} = 100 \times 10^6$
1	3 340	65 915	406 840
32	142	2 168	12 711
48	109	1 497	8 515
64	93	1 181	6 421
96	93	834	4 323
128	93	674	3 287
192	96	522	2 435
256	96	519	1 996

between the nodes, and the computational time is limited by the time taken to compute the longest task. The same situation arises for $N_{\text{det}} = 100 \times 10^6$ with 9 408 cores (192 nodes), with 2 500 tasks to compute.

Second, we analyze the parallel efficiency of the calculation of $E^{(2)}$ for the sCI wave function with $N_{\text{det}} = 25 \times 10^6$. The stopping criterion during the calculation of $E^{(2)}$ is given by a relative statistical error below 2×10^{-3} of the current $E^{(2)}$ value. The speedups are plotted in Fig. 6 (see also Table III). For 192 nodes, one obtains a parallel efficiency of 89%. The present parallel efficiency is not as good as the one presented in the original paper.[116] The reason behind this is a faster computation of $e_\alpha^{(2)}$, which reduces the parallel efficiency by increasing the ratio communication/computation.

VII. DEVELOPING IN QUANTUM PACKAGE

A. The Quantum Package philosophy

Quantum Package is a standalone easy-to-use library for developers. The main goals of Quantum Package are to i) facilitate the development of new quantum chemistry methods, ii) minimize the dependency on external programs/libraries, and iii) encourage the collaborative and educative work through human readable programs. Therefore, from the developer point of view, Quantum Package can be seen as a standalone library containing all important quantities needed to perform quantum chemistry calculations, both involving wave function theory, through the determinant driven algorithms, and DFT methods, thanks to the presence of a quadrature grid for numerical integrations and basic functionals. These appealing features are made more concrete thanks to the organization of Quantum Package in terms of core modules and plugins (see Sec. VII C) together with its programming language (see Sec. VII B), which naturally creates a very modular environment for the programmer.

Although Quantum Package is able to perform all the required steps from the calculation of the one- and two-electron integrals to the computation of the sCI energy, interfacing Quantum Package, at any stage, with other programs is relatively simple. For example, canonical or CASSCF molecular orbitals can be imported from GAMESS,[130] while atomic and/or molecular integrals can be read from text files like FCIDUMP. Thanks to this flexibility, some of us are currently developing plugins for performing sCI calculations for periodic systems.

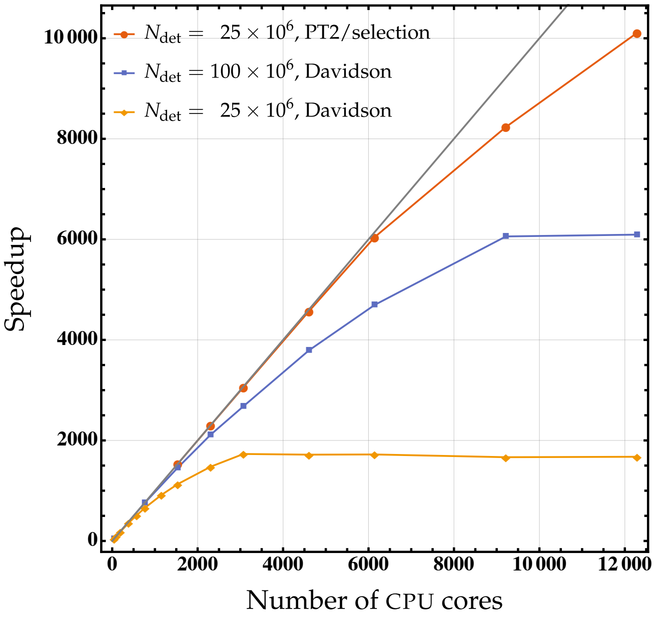


FIG. 6. Speedup obtained for a single Davidson iteration (blue and yellow curves) and the combination of CIPSI selection and PT2 calculation (red curve) as a function of the number of CPU cores. For the Davidson diagonalization, two sizes of reference wave functions are reported ($N_{\text{det}} = 25 \times 10^6$ and 100×10^6), while for the PT2/selection calculation only results corresponding to the smallest wave function ($N_{\text{det}} = 25 \times 10^6$) are reported. See Table III for raw data.

B. The IRPF90 code generator

It is not a secret that large scientific codes written in Fortran (or in similar languages) are difficult to maintain. The program's complexity originates from the inter-dependencies between the various entities of the code. As the variables are more and more coupled, the programs become more and more difficult to maintain and to debug. To keep a program under control, the programmer has to be aware of all the consequences of any source code modification within all possible execution paths. When the code is large and written by multiple developers, this becomes almost impossible. However, a computer can easily handle such a complexity by taking care of all the dependencies between the variables, in a way similar to how GNU Make handles the dependencies between source files.

IRPF90 is a Fortran code generator.[131] Schematically, the programmer only writes computation kernels, and IRPF90 generates the glue code linking all these kernels together to pro-

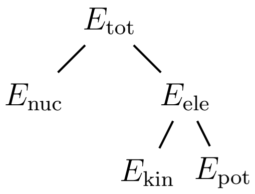


FIG. 7. Production tree of the energy computed by IRPF90.

duce the expected result, handling all relationships between variables. To illustrate in a few words how IRPF90 works, let us consider the simple example which consists in calculating the total energy of a molecular system as the sum of the nuclear repulsion and the electronic energy $E_{\text{tot}} = E_{\text{nuc}} + E_{\text{ele}}$. The electronic energy is the sum of the kinetic and potential energies, i.e., $E_{\text{ele}} = E_{\text{kin}} + E_{\text{pot}}$.

The *production tree* associated with the computation of the total energy is shown in Fig. 7. Within the IRPF90 framework, the programmer writes a *provider* for each *entity*, i.e., a node of the production tree. The provider is a subroutine whose only goal is to compute the value associated with the entity, assuming the values of the entities on which it depends are computed and valid. Hence, when an entity is used somewhere in the program (in a subroutine, a function or a provider), a call to its provider is inserted in the code before it is used such that the corresponding value is guaranteed to be valid.

Quantum Package is a library of providers designed to make the development of new wave function theory and DFT methods simple. Only a few programs using these providers are part of the core modules of Quantum Package, such as the sCI module using the CIPSI algorithm or the module containing the semi-stochastic implementation of the second-order perturbative correction. The main goal of Quantum Package is to be used as a library of providers, and programmers are encouraged to develop their own modules using Quantum Package.

C. The plugin system

External programmers should not add their contributions by modifying directly Quantum Package's core, but by creating their own modules in independent repositories hosted and distributed by themselves. This model gives more freedom to the developers to distribute modules as we do not enforce them to follow any rule. The developers are entirely responsible for their own plugins. This model has the advantage to redirect immediately the users to the right developer for questions, installation problems, bug reports, etc.

Quantum Package integrates commands to download external repositories and integrate all the plugins of these repositories into the current installation of Quantum Package. External plugins appear exactly as if they were part of Quantum Package, and if a plugin is useful for many users, it can be easily integrated in Quantum Package's core after all the coding and documentation standards are respected.

Multiple external plugins were developed by the authors. For instance, one can find a multi-reference coupled cluster program,[116,132] interfaces with the quantum Monte Carlo programs QMC=Chem,[133] QMCPack[134] and CHAMP,[135] an implementation of the shifted-Bk method,[45] a program combining CIPSI with RSDFT,[136] a four-component relativistic RSDFT code,[137] and many others.

VIII. CONCLUSION

Significant improvements were brought to the second version of Quantum Package. Some were single-core optimizations, and others focused on the algorithm adaptation to large-scale parallelism (load balancing in particular). Currently, the code has a parallel efficiency that enables routinely to realize runs on roughly 2 000 CPU cores, with tenths of millions of determinants in the reference space. Moreover, we have been able to push up to 12 288 cores (256 nodes) on GENCI's supercomputer Irene. Such a gain in efficiency has and will lead to many more challenging chemical applications.[34–38,43,44,54,59,60]

The Davidson diagonalization, which is at the center of sCI and FCI methods, suffers from the impossibility to fully store the Hamiltonian in the memory of a single node. The solution we adopted was to resort to *direct methods*, i.e., recomputing *on the fly* the matrix elements at each iteration. While an extremely fast method was already available to detect zero matrix elements,[105] the former implementation still had to search over the $\mathcal{O}(N_{\text{det}}^2)$ matrix elements for interacting determinant pairs. Now, determinants are split in disjoint sets entirely disconnected from each other. Thus, only a small fraction of the matrix elements needs to be explored, and an algorithm with $\mathcal{O}(N_{\text{det}}^{3/2})$ scaling was proposed. While the parallelization of this method was somewhat challenging due to the extremely unbalanced nature of the elementary tasks, a distributed implementation was realized with satisfying parallel speedups (typically 35 for 50 nodes) with respect to the 48-core single-node reference.

Significant improvements were also realized in the computation of the second-order perturbative correction, $E^{(2)}$. A natural idea was to take into account the tremendous amount of tiny contributions via a stochastic Monte Carlo approach. $E^{(2)}$ being itself an approximated quantity used for estimating the FCI energy, its exact value is indeed not required, as long as the value is unbiased and the statistical error is kept under control. Our scheme allows to compute $E^{(2)}$ with a small error bar for a few percents of the cost of the fully deterministic computation.

Similarly, the CIPSI selection is now performed stochastically alongside the PT2 calculation. Therefore, the selection part of the new stochastic CIPSI selection is virtually free as long as one is interested in the second-order perturbative correction.

Finally, efforts have been made to make this software as developer friendly as possible thanks to a very modular architecture that allows any developer to create an its own module and to directly benefit from all pre-existing work.

LICENSE

Quantum Package is licensed under GNU Affero General Public License (AGPLv3).

ACKNOWLEDGMENTS

The authors would like to thank the *Centre National de la Recherche Scientifique* (CNRS) for funding. This work was performed using HPC resources from i) GENCI-TGCC (Grant No. 2018-A0040801738), and ii) CALMIP (Toulouse) under allocations 2018-0510, 2018-18005 and 2019-18005. A.B. was supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division, as part of the Computational Materials Sciences Program and Center for Predictive Simulation of Functional Materials. K.G. acknowledges support from grant number CHE1762337 from the U.S. National Science Foundation.

[1] Moore, G. Cramming More Components onto Integrated Circuits. *Electronics* **1965**, *38*, 114–117.

[2] Lists | TOP500 Supercomputer Sites. 2018; https://www.top500.org/lists/1993/11, [Online; accessed 9. Oct. 2018].

[3] Sutter, H.; Larus, J. Software and the concurrency revolution. *Queue* **2005**, *3*, 54.

[4] Wulf, Wm. A.; McKee, S. A. Hitting the memory wall: implications of the obvious. *SIGARCH Comput. Archit. News* **1995**, *23*, 20–24.

[5] Khan, H. N.; Hounshell, D. A.; Fuchs, E. R. H. Science and research policy at the end of Moore's law. *Nature Electronics* **2018**, *1*, 14–21.

[6] Booth, G. H.; Thom, A. J. W.; Alavi, A. Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space. *J. Chem. Phys.* **2009**, *131*, 054106.

[7] Booth, G. H.; Alavi, A. Approaching chemical accuracy using full configuration-interaction quantum Monte Carlo: A study of ionization potentials. *J. Chem. Phys.* **2010**, *132*, 174104.

[8] Cleland, D.; Booth, G. H.; Alavi, A. Communications: Survival of the fittest: Accelerating convergence in full configuration-interaction quantum Monte Carlo. *J. Chem. Phys.* **2010**, *132*, 041103.

[9] Sharma, S.; Holmes, A. A.; Jeanmairet, G.; Alavi, A.; Umrigar, C. J. Semistochastic Heat-Bath Configuration Interaction Method: Selected Configuration Interaction with Semistochastic Perturbation Theory. *J. Chem. Theory Comput.* **2017**, *13*, 1595–1604.

[10] Garniron, Y.; Scemama, A.; Loos, P.-F.; Caffarel, M. Hybrid stochastic-deterministic calculation of the second-order perturbative contribution of multireference perturbation theory. *J. Chem. Phys.* **2017**, *147*, 034101.

[11] Smith, J. E. T.; Mussard, B.; Holmes, A. A.; Sharma, S. Cheap and Near Exact CASSCF with Large Active Spaces. *J. Chem. Theory Comput.* **2017**, *13*, 5468–5478.

[12] Neuhauser, D.; Rabani, E.; Baer, R. Expeditious Stochastic Approach for MP2 Energies in Large Electronic Systems. *J. Chem. Theory Comput.* **2013**, *9*, 24.

[13] Willow, S. Y.; Hirata, S. Stochastic, real-space, imaginary-time evaluation of third-order Feynman–Goldstone diagrams. *J. Chem. Phys.* **2014**, *140*, 024111.

[14] Willow, S. Y.; Kim, K. S.; Hirata, S. Stochastic evaluation of second-order many-body perturbation energies. *J. Chem. Phys.* **2012**, *137*, 204122.

[15] Johnson, C. M.; Hirata, S.; Ten-no, S. Explicit correlation factors. *Chem. Phys. Lett.* **2017**, *683*, 247.

[16] Johnson, C. M.; Doran, A. E.; Zhang, J.; Valeev, E. F.; Hirata, S. Monte Carlo explicitly correlated second-order many-body perturbation theory. *J. Chem. Phys.* **2016**, *145*, 154115.

[17] Gruneis, A.; Hirata, S.; Ohnishi, Y.-Y.; Ten-no, S. Perspective: Explicitly correlated electronic structure theory for complex systems. *J. Chem. Phys.* **2017**, *146*, 080901.

[18] Doran, A. E.; Hirata, S. Monte Carlo MP2 on Many Graphical Processing Units. *J. Chem. Theory Comput.* **2016**, *12*, 4821.

[19] Bender, C. F.; Davidson, E. R. Studies in Configuration Interaction: The First-Row Diatomic Hydrides. *Phys. Rev.* **1969**, *183*, 23–30.

[20] Whitten, J. L.; Hackmeyer, M. Configuration Interaction Studies of Ground and Excited States of Polyatomic Molecules. I. The CI Formulation and Studies of Formaldehyde. *J. Chem. Phys.* **1969**, *51*, 5584–5596.

[21] Huron, B.; Malrieu, J. P.; Rancurel, P. Iterative perturbation calculations of ground and excited state energies from multiconfigurational zeroth-order wavefunctions. *J. Chem. Phys.* **1973**, *58*, 5745–5759.

[22] Shih, S.; Butscher, R., W ans Buenker; Peyerimhoff, S. Calculation of vertical electronic-spectrum of nitrogen molecule using mrd-ci method. *Chemical Physics* **1978**, *29*, 241–252.

[23] Buenker, R.; Peyerimhoff, S.; Butscher, W. Applicability of multi-reference double-excitation ci (mrd-ci) method to calculation of electronic wavefunctions and comparison with related techniques. *Molecular Physics* **1978**, *35*, 771–791.

[24] Evangelisti, S.; Daudey, J.-P.; Malrieu, J.-P. Convergence of an improved CIPSI algorithm. *Chemical Physics* **1983**, *75*, 91–102.

[25] Cimiraglia, R. Second order perturbation correction to CI energies by use of diagrammatic techniques: An improvement to the CIPSI algorithm. *J. Chem. Phys.* **1985**, *83*, 1746–1749.

[26] Cimiraglia, R.; Persico, M. Recent advances in multireference second order perturbation CI: The CIPSI method revisited. *J. Comput. Chem.* **1987**, *8*, 39–47.

[27] Illas, F.; Rubio, J.; Ricart, J. M. Approximate natural orbitals and the convergence of a second order multireference many-body perturbation theory (CIPSI) algorithm. *J. Chem. Phys.* **1988**, *89*, 6376–6384.

[28] Povill, A.; Rubio, J.; Illas, F. Treating large intermediate spaces in the CIPSI method through a direct selected CI algorithm. *Theor. Chem. Acc.* **1992**, *82*, 229–238.

[29] Engels, B.; Hanrath, M.; Lennartz, C. Individually selecting multi-reference CI and its application to biradicalic cyclizations. *Computers & CHemistry* **2001**, *25*, 15–38.

[30] Abrams, M. L.; Sherrill, C. D. Important configurations in configuration interaction and coupled-cluster wave functions. *Chem. Phys. Lett.* **2005**, *412*, 121–124.

[31] Bunge, C. F.; Carbó-Dorca, R. Select-divide-and-conquer method for large-scale configuration interaction. *J. Chem. Phys.* **2006**, *125*, 014108.

[32] Musch, P.; Engels, B. DIESEL-MP2: A new program to perform large-scale multireference-MP2 computations. *Journal of Computational Chemistry* **2006**, *27*, 1055–1062.

[33] Bytautas, L.; Ruedenberg, K. A priori identification of configurational deadwood. *Chemical Physics* **2009**, *356*, 64–75.

[34] Giner, E.; Scemama, A.; Caffarel, M. Using perturbatively selected configuration interaction in quantum Monte Carlo calculations. *Can. J. Chem.* **2013**, *91*, 879–885.

[35] Caffarel, M.; Giner, E.; Scemama, A.; Ramírez-Solís, A. Spin Density Distribution in Open-Shell Transition Metal Systems: A Comparative Post-Hartree–Fock, Density Functional Theory, and Quantum Monte Carlo Study of the $CuCl_2$ Molecule. *J. Chem. Theory Comput.* **2014**, *10*, 5286–5296.

[36] Giner, E.; Scemama, A.; Caffarel, M. Fixed-node diffusion Monte Carlo potential energy curve of the fluorine molecule F_2 using selected configuration interaction trial wavefunctions. *J. Chem. Phys.* **2015**, *142*, 044115.

[37] Caffarel, M.; Applencourt, T.; Giner, E.; Scemama, A. Using CIPSI nodes in diffusion Monte Carlo. **2016**,

[38] Caffarel, M.; Applencourt, T.; Giner, E.; Scemama, A. Communication: Toward an improved control of the fixed-node error in quantum Monte Carlo: The case of the water molecule. *J. Chem. Phys.* **2016**, *144*, 151103.

[39] Holmes, A. A.; Tubman, N. M.; Umrigar, C. J. Heat-Bath Configuration Interaction: An Efficient Selected Configuration Interaction Algorithm Inspired by Heat-Bath Sampling. *J. Chem. Theory Comput.* **2016**, *12*, 3674–3680.

[40] Holmes, A. A.; Umrigar, C. J.; Sharma, S. Excited states using semistochastic heat-bath configuration interaction. *J. Chem. Phys.* **2017**, *147*, 164111.

[41] Chien, A. D.; Holmes, A. A.; Otten, M.; Umrigar, C. J.; Sharma, S.; Zimmerman, P. M. Excited States of Methylene, Polyenes, and Ozone from Heat-Bath Configuration Interaction. *J. Phys. Chem. A* **2018**, *122*, 2714–2722.

[42] Scemama, A.; Garniron, Y.; Caffarel, M.; Loos, P. F. Deterministic construction of nodal surfaces within quantum Monte Carlo: the case of FeS. *J. Chem. Theory Comput.* **2018**, *14*, 1395.

[43] Scemama, A.; Benali, A.; Jacquemin, D.; Caffarel, M.; Loos, P.-F. Excitation energies from diffusion Monte Carlo using selected configuration interaction nodes. *J. Chem. Phys.* **2018**, *149*, 034108.

[44] Loos, P.-F.; Scemama, A.; Blondel, A.; Garniron, Y.; Caffarel, M.; Jacquemin, D. A Mountaineering Strategy to Excited States: Highly Accurate Reference Energies and Benchmarks. *J. Chem. Theory Comput.* **2018**, *14*, 4360–4379.

[45] Garniron, Y.; Scemama, A.; Giner, E.; Caffarel, M.; Loos, P. F. Selected Configuration Interaction Dressed by Perturbation. *J. Chem. Phys.* **2018**, *149*, 064103.

[46] Evangelista, F. A. Adaptive multiconfigurational wave functions. *J. Chem. Phys.* **2014**, *140*, 124114.

[47] Schriber, J. B.; Evangelista, F. A. Communication: An adaptive configuration interaction approach for strongly correlated electrons with tunable accuracy. *J. Chem. Phys.* **2016**, *144*, 161106.

[48] Schriber, J. B.; Evangelista, F. A. Adaptive Configuration Interaction for Computing Challenging Electronic Excited States with Tunable Accuracy. *J. Chem. Theory Comput.* **2017**,

[49] Liu, W.; Hoffmann, M. R. iCI: Iterative CI toward full CI. *J. Chem. Theory Comput.* **2016**, *12*, 1169–1178.

[50] Per, M. C.; Cleland, D. M. Energy-based truncation of multi-determinant wavefunctions in quantum Monte Carlo. *J. Chem. Phys.* **2017**, *146*, 164101.

[51] Ohtsuka, Y.; ya Hasegawa, J. Selected configuration interaction method using sampled first-order corrections to wave functions. *J. Chem. Phys.* **2017**, *147*, 034102.

[52] Zimmerman, P. M. Incremental full configuration interaction. *J. Chem. Phys.* **2017**, *146*, 104102.

[53] Li, J.; Otten, M.; Holmes, A. A.; Sharma, S.; Umrigar, C. J. Fast semistochastic heat-bath configuration interaction. *J. Chem. Phys.* **2018**, *149*, 214110.

[54] Loos, P. F.; Boggio-Pasqua, M.; Scemama, A.; Caffarel, M.; Jacquemin, D. Reference energies for double excitations. *J. Chem. Theory Comput.* **2019**, *15*, in press.

[55] Quantum Package. 2019; https://github.com/QuantumPackage/qp2, [Online; accessed 11. Feb. 2019].

[56] Povill, À.; Rubio, J. An efficient improvement of the string-based direct selected CI algorithm. *Theor. Chem. Acc.* **1995**, *92*, 305–313.

[57] Scemama, A.; Applencourt, T.; Giner, E.; Caffarel, M. Accurate nonrelativistic ground-state energies of 3d transition metal atoms. *J. Chem. Phys.* **2014**, *141*, 244110.

[58] Giner, E.; Tew, D. P.; Garniron, Y.; Alavi, A. Interplay between Electronic Correlation and Metal–Ligand Delocalization in the Spectroscopy of Transition Metal Compounds: Case Study on a Series of Planar Cu2+ Complexes. *J. Chem. Theory Comput.* **2018**, *14*, 6240–6252, PMID: 30347156.

[59] Dash, M.; Moroni, S.; Scemama, A.; Filippi, C. Perturbatively Selected Configuration-Interaction Wave Functions for Efficient Geometry Optimization in Quantum Monte Carlo. *J. Chem. Theory Comput.* **2018**, *14*, 4176–4182.

[60] Pineda Flores, S. D.; Neuscamman, E. Excited State Specific Multi-Slater Jastrow Wave Functions. *arXiv* **2018**,

[61] Löwdin, P. Correlation Problem in ManyfifftElectron Quantum Mechanics I. Review of Different Approaches and Discussion of Some Current Ideas. *Adv. Chem. Phys.* **1959**, *2*, 207.

[62] Roothaan, C. C. J. New Developments in Molecular Orbital Theory. *Reviews of Modern Physics* **1951**, *23*, 69–89.

[63] Obara, S.; Saika, A. Efficient recursive computation of molecular integrals over Cartesian Gaussian functions. *J. Chem. Phys.* **1986**, *84*, 3963–3974.

[64] Head-Gordon, M.; Pople, J. A. A method for two-electron Gaussian integral and integral derivative evaluation using recurrence relations. *J. Chem. Phys.* **1988**, *89*, 5777–5786.

[65] Ten-no, S. An efficient algorithm for electron repulsion integrals over contracted Gaussian-type functions. *Chem. Phys. Lett.* **1993**, *211*, 259–264.

[66] Gill, P. M. W.; Head-Gordon, M.; Pople, J. A. An efficient algorithm for the generation of two-electron repulsion integrals over gaussian basis functions. *Int. J. Quantum Chem.* **1989**, *36*, 269–280.

[67] Gill, P. M. W.; Pople, J. A. The prism algorithm for two-electron integrals. *Int. J. Quantum Chem.* **1991**, *40*, 753–772.

[68] Valeev, E. F. Libint: A library for the evaluation of molecular integrals of many-body operators over Gaussian functions. http://libint.valeyev.net/, 2018; version 2.5.0-beta.1.

[69] Barca, G. M.; Loos, P.-F. Three-and Four-Electron Integrals Involving Gaussian Geminals: Fundamental Integrals, Upper Bounds, and Recurrence Relations. *J. Chem. Phys.* **2017**, *147*, 024103.

[70] Zhang, J. libreta: Computerized Optimization and Code Synthesis for Electron Repulsion Integral Evaluation. *J. Chem. Theory Comput.* **2018**, *14*, 572–587.

[71] Wilson, S. Four-Index Transformations. In *Methods in Computational Chemistry*; Springer US, 1987; pp 251–309.

[72] Rajbhandari, S.; Rastello, F.; Kowalski, K.; Krishnamoorthy, S.; Sadayappan, P. Optimizing the Four-Index Integral Transform Using Data Movement Lower Bounds Analysis. Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming - PPoPP '17. 2017.

[73] Limaye, A. C.; Gadre, S. R. A general parallel solution to the integral transformation and second-order Mo/ller–Plesset energy evaluation on distributed memory parallel machines. *J. Chem. Phys.* **1994**, *100*, 1303–1307.

[74] Fletcher, G.; Schmidt, M.; Gordon, M. Developments in parallel electronic structure theory. *Advances in chemical physics* **1999**, *110*, 267–294.

[75] Covick, L. A.; Sando, K. M. Four-Index transformation on distributed-memory parallel computers. *J. Comput. Chem.* **1990**, *11*, 1151–1159.

[76] Anderson, J. S.; Heidar-Zadeh, F.; Ayers, P. W. Breaking the curse of dimension for the electronic Schrodinger equation with functional analysis. *Comput. Theor. Chem.* **2018**, *1142*, 66–77.

[77] Coe, J. P. Machine Learning Configuration Interaction. *J. Chem. Theory Comput.* **2018**, *14*, 5739.

[78] Knowles, P.; Handy, N. A new determinant-based full configuration interaction method. *Chem. Phys. Lett.* **1984**, *111*, 315–321.

[79] Greer, J. C. Estimating full configuration interaction limits from a Monte Carlo selection of the expansion space. *J. Chem. Phys.* **1995**, *103*, 1821–1828.

[80] Greer, J. Monte Carlo Configuration Interaction. *J. Comput. Phys.* **1998**, *146*, 181–202.

[81] Nitzsche, L. E.; Davidson, E. R. A perturbation theory calculation on the 1PIPI* state of formamide. *J. Chem. Phys.* **1978**, *68*, 3103–3109.

[82] Onida, G.; Reining, L.; Rubio, A. Electronic Excitations: Density-Functional versus Many-Body Green's-Function Approaches. *Rev. Mod. Phys.* **2002**, *74*, 601–659.

[83] Reining, L. The GW Approximation: Content, Successes and Limitations: The GW Approximation. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2017**, e1344.

[84] Loos, P. F.; Romaniello, P.; Berger, J. A. Green Functions and Self-Consistency: Insights From the Spherium Model. *J. Chem. Theory Comput.* **2018**, *14*, 3071.

[85] Veril, M.; Romaniello, P.; Berger, J. A.; Loos, P. F. Unphysical Discontinuities in GW Methods. *J. Chem. Theory Comput.* **2018**, *14*, 5220.

[86] Savin, A. On Degeneracy, Near Degeneracy and Density Functional Theory. In *Recent Developments of Modern Density Functional Theory*; Seminario, J. M., Ed.; Elsevier: Amsterdam, 1996; pp 327–357.

[87] Toulouse, J.; Colonna, F.; Savin, A. Long-range–short-range separation of the electron-electron interaction in density-functional theory. *Phys. Rev. A* **2004**, *70*, 062505.

[88] Gerber, I. C.; Ángyán, J. G. Hybrid functional with separated range. *Chem. Phys. Lett.* **2005**, *415*, 100 − 105.

[89] Ángyán, J. G.; Gerber, I. C.; Savin, A.; Toulouse, J. van der Waals forces in density functional theory: perturbational long-range electron interaction corrections. *Phys. Rev. A* **2005**, *72*, 012510.

[90] Levy, M. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem. *Proc. Natl Acad. Sci. USA* **1979**, *76*, 6062.

[91] Levy, M. Electron densities in search of Hamiltonians. *Phys. Rev. A* **1982**, *26*, 1200.

[92] Lieb, E. H. Density functionals for coulomb systems. *Int. J. Quantum Chem.* **1983**, *24*.

[93] Cohen, M. H.; Wasserman, A. N-representability and stationarity in time-dependent density-functional theory. *Phys. Rev. A* **2005**, *71*, 032515.

[94] Savin, A. Beyond the Kohn-Sham Determinant. In *Recent Advances in Density Functional Theory*; Chong, D. P., Ed.; World Scientific, 1996; pp 129–148.

[95] Olver, F. W. J., Lozier, D. W., Boisvert, R. F., Clark, C. W., Eds. *NIST handbook of mathematical functions*; Cambridge University Press: New York, 2010.

[96] Franck, O.; Mussard, B.; Luppi, E.; Toulouse, J. Basis convergence of range-separated density-functional theory. *J. Chem. Phys.* **2015**, *142*, 074107.

[97] Paziani, S.; Moroni, S.; Gori-Giorgi, P.; Bachelet, G. B. *Phys. Rev. B* **2006**, *73*, 155111.

[98] Goll, E.; Werner, H.-J.; Stoll, H.; Leininger, T.; Gori-Giorgi, P.; Savin, A. A short-range gradient-corrected spin density functional in combination with long-range coupled-cluster methods: Application to alkali-metal rare-gas dimers. *Chem. Phys.* **2006**, *329*, 276.

[99] Becke, A. D. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.* **1988**, *88*, 2547–2553.

[100] Mura, M. E.; Knowles, P. J. Improved radial grids for quadrature in molecular densityfifftfunctional calculations. *J. Chem. Phys.* **1996**, *104*, 9848–9858.

[101] Garniron, Y. Development and parallel implementation of selected configuration interaction methods. Ph.D. thesis, Université de Toulouse, 2019.

[102] Maurer, W. D.; Lewis, T. G. Hash Table Methods. *ACM Comput. Surv.* **1975**, *7*, 5–19.

[103] Djoudi, L.; Barthou, D.; Carribault, P.; Lemuet, C.; Acquaviva, J.-T.; Jalby, W. MAQAO: Modular assembler quality Analyzer and Optimizer for Itanium 2. Workshop on EPIC Architectures and Compiler Technology San Jose, California, United-States. 2005.

[104] Pauncz, R. The Waller-Hartree double determinant in quantum chemistry. *Int. J. Quantum Chem.* **1989**, *35*, 717–719.

[105] Scemama, A.; Giner, E. An efficient implementation of Slater-Condon rules. *ArXiv [physics.comp-ph]* **2013**, 1311.6244.

[106] Davidson, E. R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.* **1975**, *17*, 87–94.

[107] Liu, B. The simultaneous expansion for the solution of several of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Numerical Algorithms in Chemistry: Algebraic Method, Lawrence Berkeley Laboratory, University of California, California* **1978**, 49–53.

[108] Olsen, J.; Jørgensen, P.; Simons, J. Passing the one-billion limit in full configuration-interaction (FCI) calculations. *Chem. Phys. Lett.* **1990**, *169*, 463–472.

[109] Gadea, F. X. Large matrix diagonalization, comparison of various algorithms and a new proposal. *Chem. Phys. Lett.* **1994**, *227*, 201–210.

[110] Crouzeix, M.; Philippe, B.; Sadkane, M. The Davidson Method. *SIAM Journal on Scientific Computing* **1994**, *15*, 62–76.

[111] Choi, J.; Dongarra, J.; Ostrouchov, S.; Petitet, A.; Walker, D.; Whaley, R. C. *A Proposal for a Set of Parallel Basic Linear Algebra Subprograms*; LAPACK Working Note 100, 1995; LAPACK Working Note #100. UT-CS-95-292, May 1995.

[112] Scemama, A.; Applencourt, T.; Giner, E.; Caffarel, M. Quantum Monte Carlo with very large multideterminant wavefunctions. *J. Comput. Chem.* **2016**, *37*, 1866–1875.

[113] Giner, E. Coupling Configuration Interaction and quantum Monte Carlo methods: The best of both worlds. Theses, Université de Toulouse, 2014.

[114] Cimiraglia, R. Many-body multireference Møller—Plesset and Epstein—Nesbet perturbation theory: Fast evaluation of second-order energy contributions. *Int. J. Quantum Chem.* **1996**, *60*, 167–171.

[115] Ten-no, S. L. Multi-state effective Hamiltonian and size-consistency corrections in stochastic configuration interactions. *J. Chem. Phys.* **2017**, *147*, 244107.

[116] Garniron, Y.; Giner, E.; Malrieu, J.-P.; Scemama, A. Alternative definition of excitation amplitudes in multi-reference state-specific coupled cluster. *J. Chem. Phys.* **2017**, *146*, 154107.

[117] Angeli, C.; Cimiraglia, R.; Persico, M.; Toniolo, A. Multireference perturbation CI I. Extrapolation procedures with CAS or selected zero-order spaces. *Theor. Chem. Acc.* **1997**, *98*, 57–63.

[118] Davidson, E.; Nitzche, L.; McMurchie, L. A modified Ho for Epstein-Nesbet Rayleigh-Schrōdinger pertubation theory. *Chemical Physics Letters* **1979**, *62*, 467–468.

[119] Kozlowski, P. M.; Davidson, E. R. Considerations in constructing a multireference second-order perturbation theory. *J. Chem. Phys.* **1994**, *100*, 3672–3682.

[120] Caballol, R.; Malrieu, J. P.; Daudey, J. P.; Castell, O. SCIEL program. 1998.

[121] Applencourt, T.; Gasperich, K.; Scemama, A. Spin adaptation with determinant-based selected configuration interaction. *arXiv* **2018**,

[122] Fales, B. S.; Hohenstein, E. G.; Levine, B. G. Robust and Efficient Spin Purification for Determinantal Configuration Interaction. *J. Chem. Theory Comput.* **2017**, *13*, 4162–4172.

[123] Dagum, L.; Menon, R. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.* **1998**, *5*, 46–55.

[124] Hintjens, P. *ZeroMQ*; O'Reilly Media, 2013.

[125] Forum, M. P. MPI: A Message-Passing Interface Standard. *University of Tennessee* **1994**,

[126] Bolosky, W. J.; Scott, M. L. False sharing and its effect on shared memory performance. *USENIX Association* **1993**, 3.

[127] Le Guennic, B.; Jacquemin, D. Taking Up the Cyanine Challenge with Quantum Tools. *Acc. Chem. Res.* **2015**, *48*, 530–537.

[128] Send, R.; Valsson, O.; Filippi, C. Electronic Excitations of Simple Cyanine Dyes: Reconciling Density Functional and Wave Function Methods. *J. Chem. Theory Comput.* **2011**, *7*, 444–455.

[129] Boulanger, P.; Jacquemin, D.; Duchemin, I.; Blase, X. Fast and Accurate Electronic Excitations in Cyanines with the Many-Body Bethe-Salpeter Approach. *J. Chem. Theory Comput.* **2014**, *10*, 1212–1218.

[130] Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; et al., General atomic and molecular electronic structure system. *J. Comput. Chem.* **1993**, *14*, 1347–1363.

[131] Scemama, A. IRPF90: a programming environment for high performance computing. *arXiv* **2009**,

[132] Giner, E.; David, G.; Scemama, A.; Malrieu, J. P. A simple approach to the state-specific MR-CC using the intermediate Hamiltonian formalism. *J. Chem. Phys.* **2016**, *144*, 064101.

[133] Scemama, A.; Caffarel, M.; Oseret, E.; Jalby, W. QMC=Chem: A Quantum Monte Carlo Program for Large-Scale Simulations in Chemistry at the Petascale Level and beyond. In *Lecture Notes in Computer Science*; Springer Berlin Heidelberg, 2013; pp 118–127.

[134] Kim, J. et al. QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids. *J. Phys. Cond. Mat.* **2018**, *30*, 195901.

[135] Umrigar, C. J.; Filippi, C.; Moroni, S. CHAMP: Cornell-Holland Ab-initio Materials Package. 2018.

[136] Giner, E.; Pradines, B.; Ferté, A.; Assaraf, R.; Savin, A.; Toulouse, J. Curing basis-set convergence of wave-function theory using density-functional theory: A systematically improvable approach. *J. Chem. Phys.* **2018**, *149*, 194301.

[137] Paquier, J.; Toulouse, J. Four-component relativistic range-separated density-functional theory: Short-range exchange local-density approximation. *J. Chem. Phys.* **2018**, *149*, 174110.