Coupled-Cluster Methods for Large Molecular Systems Through Massive Parallelism and Reduced-Scaling Approaches

Chong Peng

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Chemistry

Edward F. Valeev, Chair

T. Daniel Crawford

John R. Morris

Diego Troya

February 8th, 2018

Blacksburg, Virginia

Keywords: Quantum Chemistry, Electronic Structure Theory, Explicitly Correlated Coupled-Cluster Methods, Parallel Computing

Coupled-Cluster Methods for Large Molecular Systems Through Massive Parallelism and Reduced-Scaling Approaches

Chong Peng

(ABSTRACT)

Accurate correlated electronic structure methods involve a significant amount of computations and can be only employed to small molecular systems. For example, the coupled-cluster singles, doubles, and perturbative triples model (CCSD(T)), which is known as the "gold standard" of quantum chemistry for its accuracy, usually can treat molecules with 20-30 atoms. To extend the reach of accurate correlated electronic structure methods to larger molecular systems, we work towards two directions: parallel computing and reduced-cost/scaling approaches. Parallel computing can utilize more computational resources to handle systems that demand more substantial computational efforts. Reduced-cost/scaling approaches, which introduce approximations to the existing electronic structure methods, can significantly reduce the amount of computation and storage requirements.

In this work, we introduce a new distributed-memory massively parallel implementation of standard and explicitly correlated (F12) coupled-cluster singles and doubles (CCSD) with canonical $\mathcal{O}\left(N^6\right)$ computational complexity (C. Peng, J. A. Calvin, F. Pavošević, J. Zhang,

and E. F. Valeev, *J. Phys. Chem. A* 2016, **120**, 10231.), based on the TiledArray tensor framework. Excellent strong scaling is demonstrated on a multi-core shared-memory computer, a commodity distributed-memory computer, and a national-scale supercomputer. We also present a distributed-memory implementation of the density-fitting (DF) based CCSD(T) method. (C. Peng, J. A. Calvin, and E. F. Valeev, *in preparation for submission*) An improved parallel DF-CCSD is presented utilizing lazy evaluation for tensors with more than two unoccupied indices, which makes the DF-CCSD storage requirements always smaller than those of the non-iterative triples correction (T). Excellent strong scaling is observed on both shared-memory and distributed-memory computers equipped with conventional Intel Xeon processors and the Intel Xeon Phi (Knights Landing) processors. With the new implementation, the CCSD(T) energies can be evaluated for systems containing 200 electrons and 1000 basis functions in a few days using a small size commodity cluster, with even more massive computations possible on leadership-class computing resources. The inclusion of F12 correction to the CCSD(T) method makes it converge to basis set limit much more rapidly. The large-scale parallel explicitly correlated coupled-cluster program makes the accurate estimation of the coupled-cluster basis set limit for molecules with 20 or more atoms a routine. Thus, it can be used rigorously to test the emerging reduced-scaling coupled-cluster approaches.

Moreover, we extend the pair natural orbital (PNO) approach to excited states through the equation-of-motion coupled cluster singles and doubles (EOM-CCSD) method. (C. Peng, M. C. Clement, and E. F. Valeev, *submitted*) We simulate the PNO-EOM-CCSD method using

an existing massively parallel canonical EOM-CCSD program. We propose the use of state-averaged PNOs, which are generated from the average of the pair density of excited states, to span the PNO space of all the excited states. The doubles amplitudes in the CIS(D) method are used to compute the state-averaged pair density of excited states. The issue of incorrect states in the state-averaged pair density, caused by an energy reordering of excited states between the CIS(D) and EOM-CCSD, is resolved by simply computing more states than desired. We find that with a truncation threshold of 10^{-7}, the truncation error for the excitation energy is already below 0.02 eV for the systems tested, while the average number of PNOs is reduced to 50-70 per pair. The accuracy of the PNO-EOM-CCSD method on local, Rydberg and charge transfer states is also investigated.

This work received the financial support of the U.S. National Science Foundation.

Acknowledgments

I would like to acknowledge the following for their help through the graduate school:

- Prof. Edward F. Valeev, my Ph.D. advisor, for accepting me as a member of the Valeev Research Group, being patient on my slow progress in research in my early Ph.D. life, investing his time and effort to teach me, and offering me help at any time.
- Prof. T. Daniel Crawford, Prof. John R. Morris and Prof. Dr. Diego Troya, my committee members, for their guidance and support.
- Huichao Peng and Ying Gao, my parents, for being supportive of my choice of pursuing a Ph.D. in the US.
- Xi Chen, my girlfriend, for being with me through the last three years of my Ph.D. life, without whom my time at Blacksburg will be much less meaningful.
- Justus Calvin for his contribution in developing the TiledArray framework.
- Colleagues in the theoretical chemistry groups including Fabijan Pavošević, Drew Lewis, Jinmei Zhang, Ashutosh Kumar and Andrey Asadchev, for useful discussions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scientists are interested in many chemical processes that involve both the ground and excited states of large molecular systems. Quantum chemistry provides useful information to study these processes through computational simulation, which can explain or even predict experimental results. To examine these problems computationally, we must describe the electronic structure of ground and excited states of atoms and molecules accurately. It has been more than ninety years since the first introduction of time-independent Schrödinger equation in 1926:[2]

$$\hat{H}\Psi = E\Psi. \tag{1.1}$$

The obstacles to its application in chemistry remain in two categories: i) developing approaches that obtain an accurate solution to the Schrödinger equation for systems with strong electron correlation, and ii) applying those methods to large molecular systems. Solving these two problems is among the most prominent challenges for quantum chemistry. This work focuses on the second challenge that aims at applying highly accurate electronic structure theory to large molecular systems. The extreme computational demand of accurate electronic structure methods always limits their application to relatively small systems. In our opinion, there are two directions to surmount this obstacle. Firstly, introducing approximations to a current existing algorithm can reduce the computational expense and extend the existing algorithm to larger scale system. Secondly, using techniques of parallel computing can allow utilization of more computational resources to treat more extensive systems.

In this work, we discuss the research in both directions, which includes massively parallel implementation of accurate quantum chemistry methods and exploration of reduced-scaling approaches in excited-state methods. In Chapter 1, we first give a brief introduction on conventional correlated and explicitly correlated wave function methods in electronic structure theory. In Chapter 2, we introduce a parallel implementation of the explicitly correlated coupled-cluster method. We test the performance of this application on various super-computing platforms. This new implementation allows us to revisit the basis set limit for the CCSD contribution to the binding energy of π-stacked uracil dimer, a challenging paradigm of π-stacking interactions from the S66 benchmark database.[3] In Chapter 3, we describe a parallel implementation of the density-fitting (DF) based CCSD(T) program. The performance is tested on computers with Intel Xeon and Intel Xeon Phi (Knights Landing) processors. In Chapter 4, we explore the effects of truncating the Pair Natural Orbitals (PNOs) on the excitation energies from EOM-CCSD. We test the accuracy of the PNO-

EOM-CCSD approach on the low-lying excitation energies of a benchmark dataset of 28 molecules.[4]

1.1 Hartree-Fock Method

In quantum chemistry, the Schrödinger equation in Eq. 1.1 for the molecular system is written as:

$$\left[-\sum_{i}\frac{1}{2}\nabla_i^2-\sum_{A}\frac{1}{2M_A}\nabla_A^2-\sum_{A,i}\frac{Z_A}{r_{Ai}}+\sum_{A<B}\frac{Z_AZ_B}{r_{AB}}+\sum_{i>j}\frac{1}{r_{ij}}\right]\Psi=E\Psi, \tag{1.2}$$

where i,j are the index of electrons and A,B are the index of nuclei, r is the distance and Z is the nuclear charge. The exact solution to the Schrödinger equation cannot be obtained for all but simplest chemical species such as one-electron system. Hence, construction of an approximate solution to this equation is necessary. The Hartree-Fock (HF)[5] theory, based on the Born-Oppenheimer approximation,[6] aims at solving the time-independent electronic Schrödinger equation. The Born-Oppenheimer approximation assumes that the nuclei are fixed since the electrons move much faster than the heavy nuclei. Hence, the electronic Schrödinger equation for the molecular system becomes

$$\left[-\sum_{i}\frac{1}{2}\nabla_i^2-\sum_{A,i}\frac{Z_A}{r_{Ai}}+\sum_{i>j}\frac{1}{r_{ij}}\right]\Psi_{el}=E_{el}\Psi_{el}, \tag{1.3}$$

where the kinetic energies of nuclei $-\sum_A\frac{1}{2M_A}\nabla_A^2$ is zero, and the nuclear repulsion energy $\sum_{A<B}\frac{Z_AZ_B}{r_{AB}}$ is a constant.

The Slater determinant in spin-orbitals is used to represent the ground state Hartree-Fock wave function $|\Psi_0\rangle$,[7] which fulfills the antisymmetric property of the wave function:

$$\begin{aligned}|\Psi_0\rangle &= |\chi_1(\mathbf{x}_1)\chi_2(\mathbf{x}_2)\ldots\chi_n(\mathbf{x}_N)\rangle \\ &= \frac{1}{\sqrt{N!}}\begin{vmatrix}\chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \cdots & \chi_n(\mathbf{x}_1)\\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \cdots & \chi_n(\mathbf{x}_2)\\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x}_n) & \chi_2(\mathbf{x}_n) & \cdots & \chi_n(\mathbf{x}_n)\end{vmatrix},\end{aligned} \tag{1.4}$$

where $\chi(\mathbf{x})$ stands for the spin-orbital and $\mathbf{x}$ contains both the spatial and spin coordinates $\mathbf{r}, w$:

$$\chi(\mathbf{x}) = \psi(\mathbf{r})\alpha(w) \quad or \quad \psi(\mathbf{r})\beta(w). \tag{1.5}$$

The $\alpha(w)$ and $\beta(w)$ are two orthonormal functions that represent the spin up and spin down of the electron. While the Rayleigh quotient,

$$\rho = \frac{\langle\Psi_0|\hat{H}|\Psi_0\rangle}{\langle\Psi_0|\Psi_0\rangle}, \tag{1.6}$$

is always an upper bound to the exact energy, we can approximate the exact wave function by minimizing the Rayleigh quotient based on variational principle. The HF energy can be solved by minimizing the expectation value of $\hat{H}$ with respect to the HF wave function under the constraint that the spin orbitals are orthonormal. In Dirac bra-ket notation, the Hartree-Fock energy is expressed using one and two-electron integrals:

$$E_{\mathrm{HF}} = \langle\Psi_0|\hat{H}_{el}|\Psi_0\rangle = \sum_i \langle i|h|j\rangle + \frac{1}{2}\sum_{ij}\left(\langle ij|ij\rangle - \langle ij|ji\rangle\right) \tag{1.7}$$

where i and j are indices for spin orbitals χ_i and χ_j. Table 1.1 shows the indices used in this chapter that represents orbitals unless otherwise mentioned. Because of the constraint that the spin orbitals are orthonormal,

$$\langle\chi_p|\chi_q\rangle = \delta_{pq} \tag{1.8}$$

Table 1.1: Notations used in Chapter 1

Indices	Meaning
i,j,k,l	occupied molecular orbitals
a,b,c,d	unoccupied molecular orbitals
p,q,r,s	molecular orbitals
μ, ν, ρ, σ	atomic orbitals

the Lagrange's method is used to minimize the Hartree-Fock energy with respect to the orbital coefficients.

$$\mathcal{L}\left[\{\chi_i\}\right] = E_0[\{\chi_i\}] - \sum_{ij} \epsilon_{ij} \left(\langle i|j\rangle - \delta_{ij}\right), \quad \delta\mathcal{L} = 0 \tag{1.9}$$

where E_0 is the expectation value of Ψ_0 in Eq. 1.7. In canonical orbital form, the Hartree-Fock equations are

$$\hat{f}_i \chi_i(\mathbf{x}) = \epsilon_i \chi_i(\mathbf{x}) \tag{1.10}$$

Hartree-Fock equations are then converted into eigenvalue problem by introducing an atomic orbital basis set, and expanding the spin orbital as a linear combination of atomic orbitals,

$$\chi_p(\mathbf{x}) = w_p(\omega) \sum_{\mu=1}^{K} C_{\mu p} \phi_\mu(\mathbf{r}) \tag{1.11}$$

where ϕ_μ is an atomic orbital, $C_{\mu p}$ is the corresponding coefficient and w_p is the spin eigen function. This transforms Eq. 1.10 to generalized eigenvalue problem. In the matrix form, the equation will be

$$\mathbf{FC} = \mathbf{SC}\boldsymbol{\epsilon} \tag{1.12}$$

where $\mathbf{F}$ is denoted as the Fock matrix and $\mathbf{S}$ is the overlap matrix. Since $\mathbf{F}$ depends on its own solution **C**, the process must be done iteratively.[5] The actual computational procedure for obtaining Hartree-Fock wave functions is a self-consistent-field procedure.

1.2 Electron Correlation

The HF wave function approximates the exact wave function for all but one-electron systems. The electron-electron repulsions are approximated by the average repulsion of each electron from the average potential of all other electrons. This approximation causes unavoidable errors in the wave function since the HF method omits the instantaneous Coulomb interactions between electrons. While the HF wave function is represented as a single Slater determinant, the exact wave function for the ground state or excited states can be expressed as a linear combination of all the possible Slater determinants from spin-orbitals in the complete basis set.

$$|\Psi_{\text{exact}}\rangle = c_0|\Psi_0\rangle + \sum_{i,a} c_i^a|\Psi_i^a\rangle + \sum_{i<j,a<b} c_{ij}^{ab}|\Psi_{ij}^{ab}\rangle + \sum_{i<j<k,a<b<c} c_{ijk}^{abc}|\Psi_{ijk}^{abc}\rangle + ... \tag{1.13}$$

The $|\Psi_i^a\rangle$ is a singly excited Slater determinant with an electron in occupied orbital χ_i been excited to unoccupied orbital χ_a:

$$|\Psi_i^a\rangle = \hat{a}_i^a|\Psi_0\rangle, \tag{1.14}$$

where $\hat{a}_i^a$ is the excitation operator which includes the annihilation operator a_i and the creation operator $a_a^\dagger$. The $|\Psi_{ij}^{ab}\rangle$ and $|\Psi_{ijk}^{abc}\rangle$ are the doubly and triply excited determinants and so on. The procedure of adding excited determinants to the wave function is the configuration interaction (CI) method since every excited determinant is a particular configuration formation from spin orbitals. In a complete basis, the full CI wave function becomes exact wave function. The difference between HF energy and full CI ground state energy is called the correlation energy, and the mathematical definition of the correlation energy in a complete basis is[5]

$$E_{\text{corr}} = \mathcal{E}_{\text{exact}} - E_{\text{HF}}^{\infty} \tag{1.15}$$

Usually, we refer to correlation energy in a finite basis set as correlation energy, because the complete basis set is not possible to reach. The HF usually recovers over 99% of the total electronic energy of molecular systems. However, the missing 1% is essential to the accurate description of chemical processes. In the HF method, electrons with the same spin are correlated, while electrons with different spin are not. The HF method neglects instantaneous electron-electron repulsion: the dynamical electron correlation. Moreover, when a covalent chemical bond is stretched, the correlation energy usually increases, as demonstrated in Table 1.2 for water molecule at extended geometry. In fact, this is due to the non-dynamic electron correlation (static correlation) that caused by nearly degenerate electron configurations. The HF method also fails to describe so-called static correlation effects, which requires the inclusion of all nearly degenerate electron configurations into

Table 1.2: Electron correlation energy of the water molecule in aug-cc-pVDZ basis

Geometry	Correlation Energy (Hartree)
$\mathbf{R}_e$	- 0.148028
1.5 $\mathbf{R}_e$	- 0.210992
2.0 $\mathbf{R}_e$	- 0.310067

reference wave function.

1.3 Correlated Methods

As discussed in Section 1.2, the electron correlation can be considered by including the excited determinants into the HF wave function. It leads to conventional correlated methods for post-Hartree-Fock methods, such as the configuration interaction (CI) methods, coupled-cluster (CC) methods and many-body perturbation theory (MBPT).

Configuration Interaction Methods

The full CI wave function is a linear combination of all possible Slater determinants in spin-orbital basis, as given in Eq. 1.13. The computational cost of full CI grows factorial with the system size. Hence, in a finite basis set, solving the full CI is not practical for even small molecular systems with small basis set, since the total number of Slater determinants in N-electron system with K spin orbitals is $\binom{2K}{N}$. Diagonalizing the Hamiltonian matrix formed from the $\binom{2K}{N}$ full CI determinants is extremely expensive, and truncation of the full CI expansion is usually necessary. The full CI wave function can be truncated at specific excitation level to reduce the number of possible determinants. It leads to the hierarchy of CI methods such as CI with singles (CIS), CI with singles and doubles (CISD), CI with singles, doubles, and triples (CISDT) and so on. For example, the CISD wave function is written as:

$$|\Psi_{\text{CISD}}\rangle = c_0|\Psi_0\rangle + \sum_{i,a} c_i^a|\Psi_i^a\rangle + \sum_{i<j,a<b} c_{ij}^{ab}|\Psi_{ij}^{ab}\rangle. \tag{1.16}$$

As the excitation level goes higher, the wave function approaches to the full CI wave function, and the computational cost also increases. The CISD approach already has a computational scaling of $\mathcal{O}\left(N^6\right)$ while higher excitation CI methods have limited application due to high polynomial scaling. However, the truncated CI methods suffer from the issue that they are not size-consistent, which means that the energy of two non-interacting fragments will not equal to the sum of the energy of two fragments from truncated CI wave function:

$$E_A + E_B \neq E_{AB}(r_{AB} = \infty). \tag{1.17}$$

This disadvantage of truncated CI methods prevents the application to chemical problems such as bond dissociation process. Therefore, many-body perturbation theory (MBPT) and coupled-cluster (CC) method, which are size-consistent, are becoming more popular nowadays.

Many-Body Perturbation Theory

Perturbation theory, which is size-consistent, is another procedure to recover the correlation energy. In MBPT, the Hamiltonian is partitioned into a zeroth-order term and a perturbation term with the perturbation parameter λ:

$$\hat{H} = \hat{H}^{(0)} + \lambda\hat{H}^{(1)}, \tag{1.18}$$

where the zeroth-order Hamiltonian has an exact solution:

$$\hat{H}^{(0)}\Psi^{(0)} = E^{(0)}\Psi^{(0)}. \tag{1.19}$$

The wave function and energy can be expanded in a Taylor series as follows:

$$\Psi = \Psi^{(0)} + \lambda\Psi^{(1)} + \lambda^2\Psi^{(2)} + \ldots, \tag{1.20}$$

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \ldots. \tag{1.21}$$

By inserting the wave function and energy into the electronic Schrödinger equation and projecting the left side by $\Psi^{(0)}$, the expressions for energies can be obtained from collecting terms with the same order of λ:

$$\begin{aligned} E^{(0)} &= \langle\Psi^{(0)}|\,\hat{H}^{(0)}\,|\Psi^{(0)}\rangle\,, && (1.22)\\ E^{(1)} &= \langle\Psi^{(0)}|\,\hat{H}^{(1)}\,|\Psi^{(0)}\rangle\,, && (1.23)\\ E^{(2)} &= \langle\Psi^{(0)}|\,\hat{H}^{(1)}\,|\Psi^{(1)}\rangle\,, && (1.24)\\ &\vdots\\ E^{(n)} &= \langle\Psi^{(0)}|\,\hat{H}^{(1)}\,|\Psi^{(n-1)}\rangle && (1.25) \end{aligned}$$

The first order wave function is expanded by the eigen functions of zeroth-order Hamiltonian to complete:

$$\Psi^{(1)} = \sum_n c_n^{(1)}\Psi_n^{(0)}, \tag{1.26}$$

where $c_p^{(1)}$ are undetermined coefficients. To improve the Hartree-Fock energy by recovering electron correlation using perturbation theory, the Hamiltonian can be partitioned such that the Hartree-Fock Hamiltonian is the zeroth-order Hamiltonian:

$$\hat{H}^{(0)} = \hat{F}, \tag{1.27}$$

$$\hat{H}^{(1)} = \hat{H} - \hat{H}^{(0)}, \tag{1.28}$$

which is known as the Møller-Plesset perturbation theory (MPPT). Plug in this Hamiltonian to the equations for energy, we can find that:

$$E^{(0)} = \langle \Psi^{(0)} | \hat{H}^{(0)} | \Psi^{(0)} \rangle = \sum_i e_i, \tag{1.29}$$

$$E^{(1)} = \langle \Psi^{(0)} | \hat{H}^{(1)} | \Psi^{(0)} \rangle = \langle \Psi^{(0)} | \hat{H} - \hat{H}^{(0)} | \Psi^{(0)} \rangle = E_{HF} - E^{(0)}. \tag{1.30}$$

The summation of the zeroth- and first-order energy is the Hartree-Fock energy, while higher order corrections accounts for the electron correlation energy. As for the second-order Møller-Plesset perturbation theory, the first-order wave function is given by

$$\Psi^{(1)} = \frac{1}{4} \sum_{ijab} t^{ij}_{ab} \, | \Psi^{ab}_{ij} \rangle , \tag{1.31}$$

and the second-order energy is

$$E^{(2)} = \frac{1}{4} \sum_{ijab} \frac{|\bar{g}^{ab}_{ij}|^2}{e_i + e_j - e_a - e_b}. \tag{1.32}$$

In the equation above, $\bar{g}^{pq}_{rs} \equiv g^{pq}_{rs} - g^{pq}_{sr}$ and g^{pq}_{rs} is the Coulomb integral

$$g^{pq}_{rs} \equiv \langle rs | \frac{1}{r_{12}} | pq \rangle . \tag{1.33}$$

MPPT has its shortcomings such as erratic behavior and divergent behavior of the MPn series.[8] The second-order energy correction in MP2 is the first contribution to electron correlation, which accounts for around 80-90% of the correlation energy. MP2 can be used for quick estimation of electron correlation due to its computational efficiency (conventional scales $\mathcal{O}\left(N^5\right)$ but can be reduced).

Coupled-Cluster Methods

The coupled-cluster (CC) wave function is obtained by acting the exponentiated cluster operator $\hat{T}$ on the reference wave function $|\Psi_0\rangle$:

$$\begin{aligned} |\Psi_{\mathrm{CC}}\rangle &= e^{\hat{T}} |\Psi_0\rangle , && (1.34)\\ &= (1 + \hat{T} + \frac{1}{2!}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + ...) |\Psi_0\rangle , && (1.35) \end{aligned}$$

where $\hat{T}$ includes all possible excitations in N-electron system.

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + ... + \hat{T}_N, \tag{1.36}$$

$$\hat{T}_N = (\frac{1}{N})^2 \sum_{ij...ab...}^{n} T^{ij...}_{ab...} \hat{a}^{ab...}_{ij...}. \tag{1.37}$$

The electronic Schrödinger equation for CC wave function is

$$\hat{H} e^{\hat{T}} |\Psi_0\rangle = E e^{\hat{T}} |\Psi_0\rangle . \tag{1.38}$$

By introducing the similarity-transformed Hamiltonian

$$\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}, \tag{1.39}$$

the CC Schrödinger equation can be rewritten as

$$\bar{H} |\Psi_0\rangle = E |\Psi_0\rangle . \tag{1.40}$$

The ground state CC energy E is obtained by projecting the reference wave function from the left side to the equation above:

$$\langle\Psi_0|\,\bar{H}\,|\Psi_0\rangle = E, \tag{1.41}$$

while the amplitudes in Eq. 1.37 can be solved by projecting excited determinants from the left side:

$$\langle\Psi_{ijk...}^{abc...}|\,\bar{H}\,|\Psi_0\rangle = 0. \tag{1.42}$$

For N-electron system, the CC wave function become exact if the cluster operator $\hat{T}$ includes all the possible excitation operators till $\hat{T}_N$. Same as full CI, the full CC wave function is impossible to solve for realistic systems and truncation on cluster operator $\hat{T}$ is usually used. It gives the hierarchy of CC methods such as CC with singles and doubles (CCSD), CC with singles, doubles, and triples (CCSDT) and so on. For example, CCSD limits the cluster operator to singles and doubles excitations:

$$\hat{T}_{\mathrm{CCSD}} = \hat{T}_1 + \hat{T}_2. \tag{1.43}$$

While the CCSD method has relative high computational cost, which scales $\mathcal{O}\left(N^6\right)$, it is still not accurate enough for many applications, and it would require the inclusion of triples excitations for better accuracy. However, the computational cost increases fast with the full inclusion of triples excitation (CCSDT), which scales $\mathcal{O}\left(N^8\right)$. Approximations to the CC amplitudes have been introduced by using Møller-Plesset theory. For example, the CC2 model approximates the CCSD doubles amplitudes by keeping terms that only contribute to the lowest order, and the CC3 model approximates the CCSDT triples amplitudes. Moreover, the CCSD(T) method approximates the CCSDT triples amplitudes by adding a perturbative triples correction. The computational and storage scaling of different CC methods are listed in Table 1.3. The CCSD(T) method can give very accurate result in energies, interaction energies and geometries compared to experiment, which is known as the "gold-standard of quantum chemistry".

Table 1.3: Computational and storage scaling for CC methods

Methods	Computation	Storage
CC2	$\mathcal{O}\left(N^5\right)$	$\mathcal{O}\left(N^4\right)$
CCSD	$\mathcal{O}\left(N^6\right)$	$\mathcal{O}\left(N^4\right)$
CCSD(T)	$\mathcal{O}\left(N^7\right)$	$\mathcal{O}\left(N^4\right)$
CC3	$\mathcal{O}\left(N^7\right)$	$\mathcal{O}\left(N^6\right)$
CCSDT	$\mathcal{O}\left(N^8\right)$	$\mathcal{O}\left(N^6\right)$

The similarity-transformed Hamiltonian is not Hermitian, which means that the left-hand CC wave function is not conjugate of the right-hand CC wave function. In the CC analytic gradient theory, the left-hand wave function is required by solving the lambda equations.[9] The left-hand CC wave function is written as:

$$\langle\Psi_{CC}| = \langle\Psi_0|\,(1+\hat{\Lambda})e^{-\hat{T}}, \tag{1.44}$$

where $\hat{\Lambda}$ is a sum of de-excitation of cluster operator

$$\hat{\Lambda} = \hat{\Lambda}_1 + \hat{\Lambda}_2 + ... + \hat{\Lambda}_N, \tag{1.45}$$

$$\hat{\Lambda}_N = (\frac{1}{N})^2 \sum_{ij...ab...}^{n} \lambda_{ab...}^{ij...} \hat{a}_{ij...}^{ab...\dagger} \tag{1.46}$$

The lambda amplitudes can be solved by projecting the excited determinant from the right hand side

$$\langle\Psi_0| (1+\Lambda)(\bar{H}-E_{cc}) |\Psi_{ijk...}^{abc...}\rangle = 0. \tag{1.47}$$

The CC methods can also be extended to excited states through equation of motion CC (EOM-CC) theory, where excited-state CC wave function is a linear expansion of the ground-state CC wave function

$$\bar{H}\hat{R}_{(k)} |\Psi_0\rangle = E_{(k)}\hat{R}_{(k)} |\Psi_0\rangle , \tag{1.48}$$

where k is the index for excited states and $\hat{R}$ is an excitation cluser operator,

$$\hat{R}_{(k)} = \hat{R}_{0(k)} + \hat{R}_{1(k)} + \hat{R}_{2(k)} + ... + \hat{R}_{N(k)}, \tag{1.49}$$

$$\hat{R}_{N(k)} = (\frac{1}{N})^2 \sum_{ij...ab...}^{n} R_{ab...}^{ij...} \hat{a}_{ij...}^{ab...}. \tag{1.50}$$

Solving the eigenvalue problem above gives the right-hand wave function and energy for excited state k. Similarly, the left-hand eigen value problem can be represented as

$$\langle\Psi_0| \hat{L}_{(m)}\bar{H} = \langle\Psi_0| \hat{L}_{(m)}E_m. \tag{1.51}$$

Multi-configuration SCF

Multiple electronic configurations can dominate the electronic wave functions at the same time. This property causes difficulty for single-configuration HF wave function, which is incapable of describing systems containing several configurations, to give a good description. In the same way, methods developed to recover dynamical correlation, such as MP2 and coupled-cluster methods, are also not suitable for these systems. The Multi-configuration SCF (MCSCF) represents a flexible solution to the multi-configuration problem. It may be used either as a wave function to describe the electronic system or as a reference for more accurate treatments of the electronic structure. In MCSCF, the wave function of a particular state is built using a linear combination of multiple determinants.

$$|\Psi_{\mathrm{MCSCF}}\rangle = \sum_m C_m |\Psi_m\rangle \tag{1.52}$$

$|\Psi_m\rangle$ stands for the Slater determinant for mth configuration. The molecular spin orbitals in $|\Psi_m\rangle$ are expressed as linear combination of AO basis sets. The MCSCF energy is obtained by minimizing $\langle\Psi_{\mathrm{MCSCF}}|\hat{H}|\Psi_{\mathrm{MCSCF}}\rangle$ with respect to the configuration coefficients C_m and $|\Psi_m\rangle$ variationally.[10] However, it is different from CI in that the coefficients in configurations and molecular orbitals are both optimized. The MCSCF wave function works well for systems that involved degenerate or near degenerate configurations, where static electron correlation is essential. The selection of configurations is essential to the quality of the wave function and energy. Methods to select reference configurations were also developed, such as the complete active space SCF (CASSCF) and restricted active space SCF (RASSCF) method.[11,12]

Multi-reference Perturbation Theory

The Møller-Plesset perturbation theory works well in the single-reference (SR) framework. However, a simple translation of this idea to multi-configuration wave functions is not readily applicable because the choice of the zeroth-order state is not apparent. Therefore, a general perturbation theory, which takes the CASSCF wave function as the zeroth-order state, was developed.[13] In the case of CASSCF, the Fock matrix is not in diagonal form because the CASSCF wave function is not an eigenfunction of the Fock operator. There are different approaches to define the zeroth-order Hamiltonian.[14–16] Usually, the partitioning is based on the Fock Hamiltonian. To make the CASSCF wave function an eigenfunction of the zeroth order Hamiltonian, the projector

$$\hat{P} = 1 - |\Psi^{(0)}\rangle\langle\Psi^{(0)}|, \tag{1.53}$$

annihilates the zeroth order wave function component. The zeroth order Hamiltonian operator $\hat{H}^{(0)}$ is

$$\hat{H}^{(0)} = E^{(0)}|\Psi^{(0)}\rangle\langle\Psi^{(0)}| + \hat{P}\hat{f}\hat{P}, \tag{1.54}$$

where $E^{(0)}$ is the zeroth order energy $\langle\Psi^{(0)}|\hat{f}|\Psi^{(0)}\rangle$. The zeroth order Hamiltonian satisfy

$$\hat{H}^{(0)}|\Psi^{(0)}\rangle = E^{(0)}|\Psi^{(0)}\rangle. \tag{1.55}$$

The first order wave function is expressed as linear combination of single and double excitations. This approach is also known as “diagonalize then perturb”, which uses the wave function from MCSCF calculation as zeroth-order wave function (in this case CASSCF wave function) and then applies Møler-Plesset perturbation theory to it.[11] The second order energy can be solved as

$$E^{(2)} = \langle\Psi^{(1)}|\hat{H}^{(1)}|\Psi^{(0)}\rangle \tag{1.56}$$

Roos introduced his version of CASSCF perturbation theory following the “diagonalize then perturb” approach.[17] It becomes the most popular one, which is named as CASPT2. CASPT2 is size consistent, and it recovers almost eighty percent of correlation energy.

Besides the Fock partitioned Hamiltonian, there are other approaches to define zeroth-order Hamiltonian. Alternatively, Dyall proposed the Dyall Hamiltonian to define zero-order Hamiltonian in multi-reference perturbation theory, which includes two-electron terms.[18] The introduction of full active space in zeroth-order Hamiltonian can cure the intruder states problem.[18] New multi-reference perturbation approach, which uses the Dyall Hamiltonian, called "n-electron valence states perturbation theory" (NEV-PT) was introduced by Malrieu et al.[19]

Multi-reference Configuration Interaction

The multi-reference configuration interaction method (MRCI) is developed based The multi-reference CI method uses multi-configuration reference functions to generate excited determinants. For example, in the uncontracted MRCI case, the single excitations are generated by applying single excitation operator to the reference functions,

$$|\Psi_{m}{}_{i}^{a}\rangle \equiv \hat{a}_{i}^{a}|\Psi_{m}\rangle; ... \qquad \text{for } m = 1, 2 ... N_{ref} \tag{1.57}$$

where m is the index of reference function. Using the same idea in CI, the wave function of uncontracted MRCI method can be written as a linear combination of excited determinants on reference functions.

$$\Psi_{\text{MRCI}} = \sum_{m} C_m|\Psi_m\rangle + \sum_{m}\sum_{i,a} C_{m}{}_{a}^{i}|\Psi_{m}{}_{i}^{a}\rangle + \sum_{m}\sum_{a>b,i>j} C_{m}{}_{ab}^{ij}|\Psi_{m}{}_{ij}^{ab}\rangle + ... \tag{1.58}$$

where C are coefficients for each determinant. To reduce the amount of variables to optimize, many methods are used to simplify MRCI, such as the contraction method.[11] For example, the internally contracted MRCI (ic-MRCI) builds the wave function by acting excitation operators on reference with multiple configurations while fixes the reference. In ic-MRCI, the reference is a multi-configuration function:[20]

$$|\tilde{\Psi}_0\rangle = \sum_m C_m |\Psi_m\rangle \tag{1.59}$$

The coefficients C_m are optimized in MCSCF and kept fixed. The single excitation from occupied orbital to unoccupied orbital is:

$$|\tilde{\Psi}_i^a\rangle \equiv \hat{a}_i^a |\tilde{\Psi}_0\rangle \tag{1.60}$$

Then the wave function is built as linear combination of excited determinant based on $\tilde{\Psi}_0$.

$$|\Psi\rangle = \tilde{C}_0|\tilde{\Psi}_0\rangle + \sum_{i,a} \tilde{C}_a^i |\tilde{\Psi}_i^a\rangle + \sum_{i>j,a>b} \tilde{C}_{ab}^{ij} |\tilde{\Psi}_{ij}^{ab}\rangle + \dots \tag{1.61}$$

The ic-MRCI method the parameters that need to be optimized are decreased. Therefore, it can reduce the computation cost efficiently.

1.4 Slow Convergence of Electron Correlation

As mentioned in 1.2, the full CI energy converges to the exact energy as the basis reach to a complete basis (CBS). The energy difference between the energy in a finite basis and the energy in a complete basis is the basis set incompleteness error (BSIE):

$$\Delta E_{\text{BSIE}} = E_{\text{CBS}} - E. \tag{1.62}$$

To obtain a good accuracy for the electronic energy of a molecular system, it would require a balanced choice of both basis set and the level of excited determinants included in the wave function. For example, a full CI energy in minimal basis would not be accurate due to the large BSIE. Unfortunately, the BSIE in correlation energy converges towards zero slowly, which is inversely proportional to the basis set size, as the basis set size increases. Moreover, the computational cost also rises steeply as the basis set size increases. For example, the cost of evaluating two-electron integral scales $\mathcal{O}\left(K^4\right)$ (K is the size of basis), which becomes 16 times more expensive as the basis set size doubles. The computational cost increases even faster for higher polynomial scaling methods, such as CCSD(T) ($\mathcal{O}\left(N^7\right)$) and CCSDT ($\mathcal{O}\left(N^8\right)$). The reason for this issue is the use of the Slater determinants, which are constructed with products of one-electron orbitals. When the distance between two electrons approaches zero, the Coulomb potential will reach infinity, and the wave function will become singular. The wave functions of many-body systems are continuous and have first-order partial derivatives except at the Coulomb singular points.[21] This leads to the electron-electron cusp condition:

$$\left(\frac{\delta\Psi(r_{12})}{\delta r_{12}}\right)_{r_{12}\to 0} = \frac{1}{2}\Psi(r_{12}=0). \tag{1.63}$$

The electron-electron cusp condition cannot be described accurately with one-electron products. As a result, the electron correlation energy converges slowly with respect to basis set size, and it would require the inclusion of high angular momenta basis functions to describe the short-range electron-electron interactions.

A solution to this problem is to include the inter-electronic distance r_{12} explicitly into the wave function. One example is the Hylleraas-type wave function for two-electron systems:[22]

$$\Psi_N(s,t,u) = e^{-\alpha s} \sum_i c_i s^{n_i} t^{l_i} u^{m_i}, \tag{1.64}$$

where $n_i + l_i + m_i \leq N$ and $s = r_1 + r_2$, $t = r_1 - r_2$, $u = r_{12}$. Figure 1.1 plots the explicitly correlated wave function (Hylleraas wave function), and the standard correlated wave function (CI wave function) of the Helium atom with both electrons fixed on the circle. The convergence of the explicitly correlated Hylleraas wave function is much more rapidly than the conventional CI wave function. The Hylleraas wave function with a maximum quantum number of three can already give a much better description of the exact wave function than

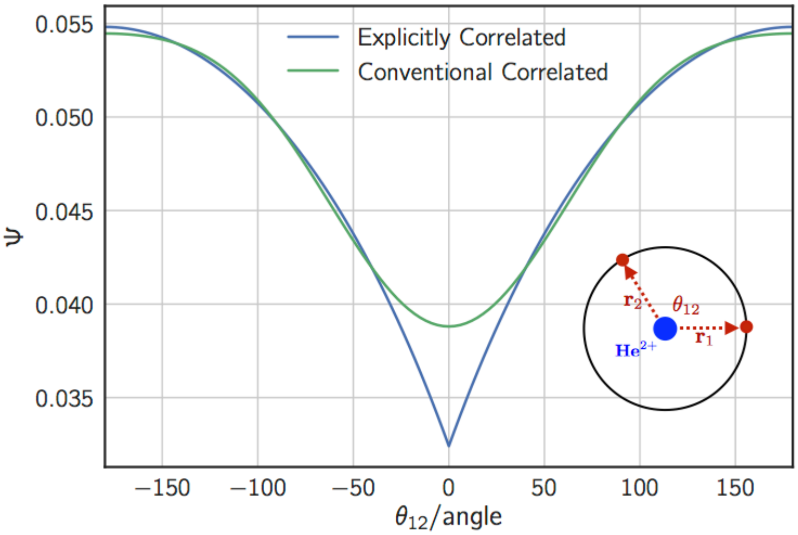

Figure 1.1: Ground state wave function of the Helium atom with both electrons fixed on the circle of 1 Bohr radius. The maximum principle quantum number of the explicitly correlated wave function (Hylleraas wave function) and the conventional correlated wave function (CI wave function) is $N = 3$ and 6, respectively.

CI wave function with a maximum quantum number of six. Moreover, the ground-state energy computed with Hylleraas wave function also converges much more rapidly than energy calculated with CI wave function.

1.5 Explicitly Correlated Methods

The use of the explicitly correlated wave function in many-electron systems traces back to 1929 by Hylleraas.[22] The explicitly correlated wave function includes the inter-electronic distance (r_{12}) and can give a better description when the inter-electronic distance is very small. This new approach shows a substantial advantage over the conventional wave function methods in the convergence of electron correlation energy. However, the use of the r_{12} functions in wave functions introduces the need for the evaluation of the expensive many-electron integrals, which is a significant obstacle in developing explicitly correlated methods for many-electron systems. Over the past years, there are many approaches to deal with the many-electron integrals in explicitly correlated methods. For example, the analytical evaluation of Gaussian integrals,[23] the trans-correlated methods,[24] the quantum Monte Carlo methods,[25] and the resolution-of-the-identity (RI) in the R12 methods.[26] The R12 methods (or F12 methods) are among the most successful explicitly correlated methods, which is first proposed by Kutzelnigg in 1985.[27] In Kutzelnigg's R12 method, the conventional wave function is appended with a reference wave function scaled by the r_{12} factor, which gives better description of the electron-electron cusp condition. The RI technique is also used to factorize the many-electron integrals into two-electron integrals. The R12 wave function can be combined with the conventional correlated methods such as MBPT and CC methods.

In the MP2-R12 method, the first order wave function of MP2-R12 is constructed as a combination of the MP2 first order wave function with explicitly correlated terms:

$$|\Psi^{(1)}_{\mathrm{MP}}\rangle = \frac{1}{4}\sum_{ijab} t^{ij}_{ab}\hat{a}^{ab}_{ij}|\Psi_0\rangle, \tag{1.65}$$

$$|\Psi^{(1)}_{\mathrm{MP-R12}}\rangle = |\Psi^{(1)}_{MP}\rangle + \hat{R}\,|\Psi_0\rangle\,, \tag{1.66}$$

$$\hat{R} = \frac{1}{2}\sum_{ijkl} t^{ij}_{kl}\hat{R}^{kl}_{ij}, \tag{1.67}$$

$$\hat{R}^{kl}_{ij} = \frac{1}{4}\sum_{\alpha\beta} \bar{R}^{kl}_{\alpha\beta}\hat{a}^{\alpha\beta}_{ij}, \tag{1.68}$$

where α, β ... are indices for virtual orbitals in the complete basis. The $\bar{R}^{kl}_{\alpha\beta}$ is the antisymmetrized form of integral:

$$R^{kl}_{\alpha\beta} = \langle\alpha\beta|\,\hat{Q}_{12}f(r_{12})\,|ij\rangle\,. \tag{1.69}$$

The projector $\hat{Q}_{12}$[28] makes sure that the R12 functions are strongly orthogonal to the MP2 part:

$$\hat{Q}_{12} = (1-\hat{O}_1)(1-\hat{O}_2) - \hat{V}_1\hat{V}_2. \tag{1.70}$$

The t^{kl}_{ij} are the geminal amplitudes, which can be variationally optimized or fixed by using the SP method.[29] The SP approach determines the geminal amplitudes from the singlet and triplet cusp conditions. Therefore, the $\hat{R}$ can be expressed using fixed geminal amplitudes:

$$\hat{R} = \frac{1}{2}\sum_{ijkl}\left(\frac{1}{2}\hat{P}_0 + \frac{1}{4}\hat{P}_1\right)\hat{R}^{kl}_{ij}, \tag{1.71}$$

where $\hat{P}_0$ and $\hat{P}_1$ are the singlet and triplet spin projectors. The $f(r_{12})$ is the geminal correlation factor. In the early times of R12 methods, the linear r_{12} term is used as the correlation factor. The linear r_{12} function gives a good description of the wave function at small inter-electronic distances, but not at long-range distances. Improvements had been done by using non-linear correlation factors, including the Slater-type geminal[30] $e^{-\zeta r_{12}}$ and the Gaussian-type geminals[31] $e^{-\zeta r_{12}^2}$, which are referred as the F12 methods. The accuracy of various correlation factors has been compared using MP2-F12 calculations,[32,33] and the Slater-type geminal appears to have the best performance. Usually, the Slater-type geminal is fitted with a linear combination of Gaussian-type geminals,

$$e^{-\zeta r_{12}} \approx \sum_{i}^{N} c_i e^{-a_i r_{12}^2}, \tag{1.72}$$

to make it easier to evaluate the F12 integrals.

The final MP2-R12 energy is the combination of the standard MP2 energy with the R12 correction:

$$E_{\text{MP2-R12}} = E_{\text{MP2}} + E_{\text{R12}}. \tag{1.73}$$

In the evaluation of the MP2-R12 energy, it will encounter intermediates (V, X, and B) that require evaluation of many-electron integrals. The intermediates V and X involve three-electron integrals and intermediate B requires up to four-electron integrals.[34] To reduce the computational cost, the RI method is introduced to convert the costly integrals into two-electron integrals:

$$1 \approx \hat{P}' = \sum_{p'} |p'\rangle\langle p'|, \tag{1.74}$$

where $\{p'\}$ is the RI basis that approximates the complete basis set. Taking the V intermediate as an example, the matrix element which requires three-electron integral to evaluate can be reduced to two-electron integral using the RI approach. The elements of V intermediate V_{ij}^{kl} can be written as

$$\begin{aligned}
V_{ij}^{kl} &= \langle ij| \frac{1}{r_{12}} \hat{Q}_{12} f_{12} |kl\rangle \\
&= \langle ij| \frac{1}{r_{12}} \Big((1-\hat{O}_1)(1-\hat{O}_2) - \hat{V}_1\hat{V}_2\Big) f_{12} |kl\rangle \\
&= \langle ij| \frac{f_{12}}{r_{12}} |kl\rangle + \sum_{mn} \langle ij| \frac{1}{r_{12}} |mn\rangle \langle mn| f_{12} |kl\rangle - \sum_{ab} \langle ij| \frac{1}{r_{12}} |ab\rangle \langle ab| f_{12} |kl\rangle \\
&- \langle ij| \frac{1}{r_{12}} \hat{O}_1 f_{12} |kl\rangle - \langle ij| \frac{1}{r_{12}} \hat{O}_2 f_{12} |kl\rangle .
\end{aligned} \tag{1.75}$$

In the expression above, the first three terms only require evaluation of two-electron integrals, while the last two terms require three-electron integrals. Using the RI approximation, these two terms can be reduced to products of two-electron integrals:

$$\begin{aligned}
\langle ij|\frac{1}{r_{12}}\hat{O}_1 f(r_{12})|kl\rangle &\approx \langle ij|\frac{1}{r_{12}}\hat{O}_1\hat{P'}_2 f(r_{12})|kl\rangle \\
&= \sum_{mp'}\langle ij|\frac{1}{r_{12}}|mp'\rangle\langle mp'|f(r_{12})|kl\rangle \\
&= g_{ij}^{mp'} r_{mp'}^{kl},
\end{aligned} \tag{1.76}$$

The choices of the RI basis $\{p'\}$ include orthonormal orbital basis set (OBS),[35] auxiliary basis set (ABS)[36] and complementary ABS (CABS).[28]

The R12 methods are also extended to coupled-cluster (CC) theory. Noga proposed the CC-

R12 method by including the r_{12} term into the exponential expansion of the wave function:[37]

$$|\Psi_{\text{CC-R12}}\rangle = e^{\hat{S}} |\Psi_0\rangle , \tag{1.77}$$

$$\hat{S} = \hat{T} + \hat{R}. \tag{1.78}$$

The Schrödinger equation for the CC-R12 wave function can be written as:

$$e^{-\hat{S}} \hat{H} e^{\hat{S}} |\Psi_0\rangle = E |\Psi_0\rangle \tag{1.79}$$

Similarly, the energy and amplitude equations can be obtained by projecting the excited determinants from the left side:

$$E = \langle\Psi_0| e^{-\hat{S}} \hat{H} e^{\hat{S}} |\Psi_0\rangle , \tag{1.80}$$

$$0 = \langle\Psi_0| \hat{a}_{ijk...}^{abc...\dagger} e^{-\hat{S}} \hat{H} e^{\hat{S}} |\Psi_0\rangle , \tag{1.81}$$

$$0 = \langle\Psi_0| \hat{R}_{ij}^{kl\dagger} e^{-\hat{S}} \hat{H} e^{\hat{S}} |\Psi_0\rangle . \tag{1.82}$$

Taking CCSD-R12 as an example, the operator $\hat{T}$ is truncated to only include singles and doubles excitations. The expression of the CCSD-R12 energy equation is:

$$\begin{aligned} E_{\text{CCSD-R12}} &= \langle\Psi_0| \bar{H}_{\text{CCSD}} + \left[\bar{H}_{\text{CCSD}}, \hat{R}\right] |\Psi_0\rangle \\ &= \langle\Psi_0| \bar{H}_{\text{CCSD}} |\Psi_0\rangle + \langle\Psi_0| \left[\hat{W}, \hat{R}\right] |\Psi_0\rangle \end{aligned} \tag{1.83}$$

The CCSD-R12 singles and doubles amplitude can be obtained by solving the equations below:

$$0 = \langle\Psi_0| \hat{a}_i^{a\dagger} \left(\bar{H}_{\text{CCSD}} + \left[\bar{H}_{\text{CCSD}}, \hat{R}\right]\right) |\Psi_0\rangle , \tag{1.84}$$

$$0 = \langle\Psi_0| \hat{a}_{ij}^{ab\dagger} \left(\bar{H}_{\text{CCSD}} + \left[\bar{H}_{\text{CCSD}}, \hat{R}\right] + \frac{1}{2}\left[\left[\bar{H}_{\text{CCSD}}, \hat{R}\right], \hat{R}\right]\right) |\Psi_0\rangle , \tag{1.85}$$

$$0 = \langle \Psi_0 | \hat{R}_{ij}^{kl\dagger} \left(\bar{H}_{\mathrm{CCSD}} + \left[\bar{H}_{\mathrm{CCSD}}, \hat{R} \right] + \frac{1}{2} \left[\left[\bar{H}_{\mathrm{CCSD}}, \hat{R} \right], \hat{R} \right] \right) | \Psi_0 \rangle . \tag{1.86}$$

The Eq. 1.86 introduces additional R12 intermediates that requires analytical evaluation of many-electron integrals, which raises the computational cost by a significant amount. Though the CC-R12 methods are much more complicated than conventional CC methods due to the inclusion of r_{12} terms, there are existing implementations of CC-R12 methods using an auto-generated code, such as CCSD-R12, CCSDT-R12, and CCSDTQ-R12.[38–40]

Because of the high complexity and computational cost of the CC-R12 methods, various approximations have been introduced to CC-R12 methods, which significantly simplified the formalism of CC-R12 models. Klopper *et al.* proposed the CCSD(R12) method by only keeping the linear terms and omitting the $\left[\hat{W}, \hat{R} \right]$ term in the CC-R12 amplitudes equations .[41] In their approach, the expressions for double and geminal amplitudes equations in Eq. 1.85 and Eq. 1.86 are simplified to:

$$0 = \langle \Psi_0 | \hat{a}_{ij}^{ab\dagger} \left(\bar{H}_{\mathrm{CCSD}} + \left[\bar{H}_{\mathrm{CCSD}}, \hat{R} \right] \right) | \Psi_0 \rangle , \tag{1.87}$$

$$0 = \langle \Psi_0 | \hat{R}_{ij}^{kl\dagger} \left(\bar{H}_{\mathrm{CCSD}} + \left[\hat{F}, \hat{R} \right] \right) | \Psi_0 \rangle . \tag{1.88}$$

Later, Hättig *et al.* continued to simplify the CCSD(R12) method by introducing the CCSD[F12] and CCSD(F12*) methods within the SP ansatz.[42] The CCSD[F12] method ignores the F12 terms higher than the third order in energy, while the CCSD(F12*) keeps some critical high-order terms. The cost of the CCSD[F12] method is only slightly higher than conventional CCSD method, and the cost of the CCSD(F12*) method is similar to CCSD(F12). Werner *et al.* introduced the CCSD-F12x and CCSD(T)-F12x methods within the SP ansatz, where $x = a, b$.[43,44] The CC-F12x ignored the $\left[\left[\hat{W}, \hat{T}_2\right], \hat{R}\right]$ term in the doubles amplitudes equation of the CCSD(F12) method:

$$0 = \langle \Psi_0 | \, \hat{a}_{ij}^{ab\dagger} \left(\bar{H}_{\mathrm{CCSD}} + \left[\hat{H}, \hat{R}\right] \right) | \Psi_0 \rangle \,, \tag{1.89}$$

Moreover, the summations over CABS indices of F12 intermediates that do not appear in MP2-F12 theory are neglected. Valeev and co-workers proposed perturbative approximations to the CC-F12 framework, which are dubbed as CCSD(2)$_{\overline{\mathrm{F12}}}$ and CCSD(T)$_{\overline{\mathrm{F12}}}$ methods.[45–47] In these approaches, the CCSD amplitudes are not modified, and the final energy includes the contribution from F12 intermediates. The approximations only introduce very small errors and increase the computational cost by a small amount.

Multi-reference methods also suffer from the problem of slow convergence of electron correlation. The explicitly correlated wave function is also used in multi-reference methods. Torheyden and Valeev published their R12 method, which works for any arbitrary wave functions, such as CASPT2 and MRCI, as long as the one and two-electron density matrices are available. This approach is denoted as $[2]_{\mathrm{R12}}$. The spin-free version of this approach was published later.[48] The CABS singles energy method is used to correct the basis set error for the reference wave functions,[49] which is denoted as $[2]_{\mathrm{S}}$. Shiozaki and Werner developed F12 approaches to MRCI and CASPT2: MRCI-F12[50] and CASPT2-F12.[51] Recently, an explicitly correlated NEVPT2 approach (NEVPT2-F12) was published.[52]

1.6 Conclusions

We introduced the conventional correlated wave function methods in electronic structure theory and pointed out the slow convergence of electron correlation energy with respect to the basis functions size of conventional correlated methods. The explicitly correlated wave function methods (F12 methods) reduce the basis set incompleteness error significantly by inclusion of explicitly correlated geminals in the wave function. The explicitly correlated coupled-cluster methods (CC-F12) were introduced by Noga,[37] and approximations that reduce the complexity of the original CC-F12 methods were also developed.[41–43,45] The approximate CC-F12 methods have been implemented in various quantum chemistry packages, such as DALTON,[53] TURBOMOLE,[54] MPQC[55] and MOLPRO.[56] Most of the existing implementation of CC-F12 methods are designed to shared-memory systems but not distributed memory systems. The CC-F12 methods have the same high computational complexity as the conventional CC methods. For example, the $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ method[45,46] and $\mathrm{CCSD(T)}_{\overline{\mathrm{F12}}}$ method[47] developed by Valeev scale $\mathcal{O}\left(N^6\right)$ and $\mathcal{O}\left(N^7\right)$, respectively. Therefore, the CC-F12 methods are usually limited to systems with less than 30 atoms with shared-memory parallelized softwares.

In this work, we present the research we have done to extend the CC methods and the CC-F12 methods to larger molecular systems. In Chapter 2, we introduce a distributed-memory implementation of the $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ method. In Chapter 3, we describe a distributed-memory implementation of the density-fitting (DF) based perturbative correction to CCSD, which leads to the complete implementation of the CCSD(T)$_{\overline{\mathrm{F12}}}$ method.[47] In Chapter 4, we try to combine the Pair Natural Orbitals (PNOs) with coupled-cluster methods for excited states (PNO-EOM-CCSD), which will reduce the computational complexity of EOM-CCSD significantly.

Chapter 2

Massively Parallel Implementation of Explicitly Correlated Coupled-Cluster Singles and Doubles Using TiledArray Framework

Reproduced in part with permission from C. Peng and E. F. Valeev, *J. Phys. Chem. A*, 2016, **120**, 10231. Copyright 2016 American Chemical Society

2.1 Introduction

The coupled-cluster (CC) ansatz[57,58] for the electronic wave function is a robust model of a correlated n-electron system that has rapid (geometric) convergence to the exact solution of the Schrödinger equation for ground- and excited-states of small molecules. Even at low truncation ranks it provides sufficient accuracy for many molecular applications, as demonstrated by the gold-standard coupled-cluster singles doubles with perturbative triples (CCSD(T)).[59] However, the utility of CCSD(T) and higher-order analogs is hampered by two factors:

- large errors in some molecular properties (most notably, energy) when small basis sets are used, with slow asymptotic convergence to the exact numerical limit, and
- the high computational complexity of the conventional (naive) implementations of such theories, e.g. the cost of CCSD(T) is $\mathcal{O}\left(N^7\right)$ with system size N.

Thus precise numerical computation of the coupled-cluster wave functions is limited to ~ 10 nonhydrogen atoms.

The basis set problem of the traditional coupled-cluster can be addressed from first principles by the introduction of the cluster operators, which is explicitly dependent on the interelectronic distances, in the wave function ansatz. The R12/F12 formalism of the explicitly correlated coupled-cluster was first explored in the original form by Noga, Kutzelnigg, and others,[60,37] and later CCSD and higher-order variants of F12 were realized in modern form with the help of specialized computer algebra systems.[38–40,61] Due to the complexity and great expense of the rigorous formulation of the CCSD-F12 relative to that of CCSD, practical introduction of F12 terms into the coupled-cluster framework even at the singles and doubles level must involve approximations. *Iterative* approximations to CCSD-F12, such as CCSD(F12),[41] CCSD-F12{a,b},[43] CCSD(F12*), and CCSD[F12],[42] include the geminal terms – only the most essential ones – in the conventional amplitude equations. *Perturbative* approximations to CCSD-F12 are constructed by a low-order perturbative expansion with respect to CCSD as the zeroth-order.[45] Both styles of approximations have similar costs (although CCSD(F12) is significantly more expensive than others) and their performance is comparable. For prediction of relative energies the explicitly correlated CCSD energy obtained with a basis of cardinal number X is roughly equivalent to the conventional CCSD counterpart with the cardinal number $X + 2$, resulting in savings of 1 to 2 orders of magnitude.[62,63]

To apply explicitly correlated coupled-cluster to larger systems, we must reduce the prohibitive, $\mathcal{O}(N^6)$, operation count. One of several established reduced-scaling frameworks can be used for this purpose. Some reduced-scaling frameworks treat only wave functions of small fragments rather than the whole system; the property of the whole system is then patched up from the contributions of its fragments. This group of methods includes the divide-and-conquer method,[64] the fragment molecular orbital (FMO),[65] the incremental scheme,[66] the divide-expand-consolidate (DEC),[67] and others. Another strategy is to explicitly compute the wave function of the whole system by employing sparse representation for the wave function and the Hamiltonian. These approaches are efficient (by avoiding the redundancy of most fragment approaches) but are technically elaborate due to the use of more complex data-sparse representations. The reduced-complexity representation predicates the use of a basis localized in spatial or other sense, e.g. the local correlation approach of Pulay and Saebø[68,69] uses spatially localized occupied orbital and projected atomic orbitals (PAOs) as the localized unoccupied orbitals. Similarly, atomic orbital representations was pioneered by Almlöf in the context of the second-order Møller-Plesset energy[70] and extended to the coupled-cluster context by Scuseria and co-workers.[71] Combined use of localized occupied and atomic orbitals for the unoccupied space is also possible.[72]

The coupled-cluster approaches based on the ideas of Pulay recently emerged as the robust candidates for routine chemical applications. The foundation for the recent progress was set up by the demonstration of MP2 and CCSD(T) in linear-scaling form by the Werner group.[73,74] A crucial step towards robust application of such methods is the use of the pair-natural orbitals[75] (PNO) as the basis in which the coupled-cluster equations are solved, which was first demonstrated by Neese an co-workers.[76] *Robust* linear-scaling implementation of many-body methods based on the use of PNOs and efficient block-sparse tensor formalisms were recently demonstrated (MP2, MP2-F12, CCSD(T) and NEVPT2 by Neese, Valeev, and co-workers[77–80] and MP2 and MP2-F12 by Werner and co-workers[81,82]). Polynomial reduced-scaling explicitly correlated CCSD methods were demonstrated in 2014 by Valeev and Neese[83] and by Hättig and Tew.[84]

Despite the recent emergence of the robust reduced-scaling variants of the coupled-cluster methods, high-performance conventional CC formulations are still highly relevant. One reason is that they are necessary as the benchmarks for the development of reduced-scaling formalisms. The efficient reduced-scaling formulations available now are not adaptive, i.e. their precision is a complex function of numerous truncation thresholds whose values are fixed throughout the computation. As these parameters approach zero the reduced-scaling method become equivalent to the full-scaling counterpart. However, due to the steep growth of the computational cost with these thresholds, the limit cannot be approached in practice except for the smallest systems. In fact the reduced-scaling PNO-based formulations become far more expensive than the traditional formulations as the truncation thresholds approach zero, e.g. without truncations the integral transformation in PNO-based CCSD costs $\mathcal{O}(N^7)$, and dominates the $\mathcal{O}(N^6)$ cost of standard CCSD and the $\mathcal{O}(N^5)$ cost of its integral transformations. Thus, to design robust reduced-scaling formulations we need the help of efficient conventional formulations.

Another reason to strive for an efficient conventional implementation of the coupled-cluster methods is that the existing reduced-scaling approaches, which are based on the real-space truncation, may be less appropriate for some applications, e.g. computation of nonresonant field response of the wave function. Lastly, for some applications the precision that is desired will be too high to warrant a reduced-scaling computation. A high-efficiency full-scaling implementation of the explicitly correlated CC will be again highly desirable in such circumstances.

With these objectives in mind, we developed a high-performance massively parallel program of our explicitly correlated CCSD method. The goal of this paper is to describe the implementation as well as to demonstrate its suitability to provide high-precision benchmarks for reduced-scaling coupled-cluster approaches.

To establish the context for our work we must briefly review the existing *distributed-memory* parallel implementations of the coupled-cluster methods. To the best of our knowledge, there are no other distributed-memory parallel implementations of explicitly correlated coupled-cluster. In our discussion we only limit ourselves to the published accounts of parallel CCSD implementations aimed at strong scaling. Thus we exclude implementations that aim at small-scale setups which typically fully replicate the coupled-cluster amplitudes, most notably in Molpro and CFOUR.[85]

The first massively parallel implementation of the conventional closed-shell CCSD energy was reported in 1992 by Rendell, Lee, and Lindh,[86] based on the spin-adapted CCSD formulation of Scuseria, Janssen, and Schaefer.[87] The authors utilized a novel semi-direct strategy in which the contributions to the doubles amplitude equations from integrals with 3 and 4 unoccupied (*virtual*) indices were computed in an integral-direct fashion, using AO integrals computed on the fly; the rest of the terms were evaluated in the molecular orbital basis. In 1997 Kobayashi and Rendell reported a similar formulation of closed-shell CCSD in the NWChem[88] program, using the Global Array (GA) toolkit to store data across the distributed machine as well as to perform distributed matrix multiplication.[89] Excellent strong scaling was demonstrated on Cray T3D; 70% parallel efficiency was maintained upon increasing the processor count from 16 to 256. This implementation was recently improved[90] to reduce the communication in the integral-direct computation by replicating the (symmetrized) doubles amplitudes; at the cost of increased memory requirement per node this permits to eliminate all remote reads (but remote accumulation is still needed). A strong-scaling test of the improved CCSD code demonstrated a 30% parallel efficiency upon increasing the node count from 1100 to 20,000 on a Cray XE6 cluster (the "Blue Waters" supercomputer at NCSA).

NWChem has another implementation of the conventional coupled-cluster methods based on the Tensor Contraction Engine (TCE).[91,92] The TCE code is produced by an integrated many-body algebra compiler that derives and optimizes equations (e.g. by common subexpression elimination, strength reduction, etc.). The generated low-level FORTRAN code uses Global Array runtime for one-sided distributed memory operations. The TCE module implements a variety of spin-orbital CC methods (for closed-shell systems the spin symmetry is only partially utilized) for ground and excited states, all in MO basis (i.e. not AO integral-driven). The TCE EOM-CC program has been applied to several large systems with hundreds of electrons and basis functions.[93–95] Recent improvements of the TCE compiler focused on coarse-grained parallel evaluation of the tensor expressions[94] dramatically improved the strong scaling of the code; the CCSD program demonstrated 84% parallel efficiency upon increase in core count from 768 to 3072 on a commodity infiniband cluster.[95]

The Parallel Quantum Solutions (PQS) package of Pulay and co-workers implemented closed-shell CCSD and CCSD(T) methods using an atomic orbital-driven algorithm[96,97] based on a closed-shell spin-adapted CCSD formulation of Hampel, Peterson, and Werner[98] (closely related to that of Scuseria *et al.*[87] by using the same generator-state formalism[99]). This implementation used Array Files(AF) middleware[100] to store data on each node's local disk; by providing each compute process with one-sided access to data the implementation of disk-resident data was greatly simplified. The QCISD program, which is similar to CCSD, could handle calculations as large as benzene dimer with 1512 basis functions without symmetry. 90% parallel efficiency was observed for the CCSD program upon scaling from 2 to 16 nodes[97]! Another interesting feature of the implementation is the use of AO integral sparsity in the direct computation of CCSD doubles residual (this is useful for computation on large molecules with small basis sets).

The original parallel CCSD and CCSD(T) implementations[101] developed by the Gordon group in the GAMESS package utilized the hybrid AO-MO approach, similar to that in NWChem, but based on the closed-shell spin-adapted formulation of Piecuch et al.[102] The program utilized hybrid parallelism (message passing + interprocess UNIX System V communication), with the distributed data in memory managed using the Distributed Data Interface (DDI) that allowed more efficient data sharing between processes on the same node. The parallel efficiency was modest, limited by the scalability of the MO terms in CCSD. An alternative implementation was developed as a standalone library (later integrated into GAMESS) by Asadchev and Gordon.[103] The key innovation of this work was to consider explicitly disk (local or otherwise) as an additional level of memory hierarchy, similar to the PQS implementation, but also used Global Arrays for fast one-sided distributed memory operations. Unlike all previous implementations, the new program utilized thread programming explicitly for intranode parallelism; it can take advantage of NVidia GPUs for matrix multiplication. On a small commodity cluster the CCSD program illustrated as high as $\sim 83\%$ parallel efficiency upon core increase from 24 to 96; on a high-end Cray XE6 supercomputer the efficiency was $\sim 93\%$ upon core increase from 256 to 1024.

ACES III program is a massively parallel re-engineering of the ACESII program of Bartlett *et al.* and includes a full suite of ground- and excited-state coupled-cluster methods, e.g. CCSD and CCSD(T) energy and gradient are available for both restricted and unrestricted HF reference wave functions. All methods were implemented in terms of a domain-specific language, the super instruction assembly language(SIAL), which is processed by the super instruction processor(SIP) interpreter.[104,105] The program includes explicit I/O statements, but parallelism is implicit and achieved by distributing "super-instructions" (tasks) to execution agents on the nodes of the parallel machine. Excellent strong scaling, usually perfect or superlinear, was demonstrated for CCSD[106] and other related methods.[107] Around 80% of parallel efficiency was obtained on Cray XT5 cluster (Jaguar at ORNL) by increasing number of processors from 2000 to 8000.[104]

Several implementations of coupled-cluster appeared recently based on the Cyclops Tensor Framework (CTF) of Solomonik,[108] most notably Aquarius[109] of Devin Matthews as well as Q-Chem.[110] CTF is a distributed-memory framework for dense and element-sparse tensor arithmetic implemented in terms of MPI and OpenMP. CTF employs state-of-the-art communication-optimal algorithms for tensor contractions (so-called 2.5D variant[111] of the popular SUMMA algorithm[112]). The Aquarius program implemented a number of ground-

and excited-state coupled-cluster methods, up to CCSDTQ, using MO-only algorithms, with all tensors fully distributed. Excellent strong scaling of the CCSD in Aquarius was demonstrated on the high-end Cray XC30 supercomputer at NERSC,[109] far outperforming the TCE implementation of CCSD in NWChem (it is not clear if this comparison utilized the improvements of the TCE code reported in Ref. 94). For example, for a cluster of 15 water molecules in cc-pVDZ basis parallel efficiency is conservatively estimated at $\sim$ 50% upon node count increase from 16 to 256. The sequential performance of Aquarius was worse than NWChem, and the scaling with the number of threads was modest.

In this paper we describe a new parallel implementation of the explicitly correlated closed-shell coupled-cluster singles and doubles energy. Our implementation distributes all data greater than $\mathcal{O}(N)$ in memory and implements conventional and AO integral-direct approaches, with and without density fitting. The rest of the paper is structured as follows. Section 2.2.1 briefly describes the TiledArray tensor framework and its key features for the present work. Section 2.2.2 and 2.2.3 recap the explicitly correlated coupled-cluster formalism and its implementation in the MPQC[113] (version 4) program. Section 2.3.1 documents the performance of the code on a single multicore computer against state-of-the-art freely available CCSD implementations in Psi4[114] and ORCA[115] programs. The parallel performance of the program on a commodity cluster is shown and compared to NWChem's TCE CCSD implementation. Finally, the parallel performance is demonstrated a national-scale supercomputer. Section 2.3.2 demonstrates the utility of our implementation to establish the numerical CCSD limit for the binding energy of uracil dimer by performing explicitly correlated CCSD energy computations in a large quadruple-zeta basis and thus resolve the discrepancy between two recent predictions for this system obtained with reduced-scaling CCSD F12 approaches. Section 2.4 summarizes the essential findings.

2.2 Methods

2.2.1 TiledArray Tensor Framework

TiledArray is an open-source framework for distributed-memory parallel implementation of dense and block-sparse tensor arithmetic. Although the development of TiledArray in our group has been driven by the needs of the reduced-scaling electronic structure theory, the framework is a domain-neutral toolkit that can be deeply customized for general-purpose computation with flat and hierarchical tensor data. Efficient application to particular domains of application is nevertheless possible due to careful design of zero-runtime-cost abstractions. An in-depth description of TiledArray will be published elsewhere; here we only present a short synopsis of TiledArray's concepts and discuss its key features that are relevant to this work.

TiledArray's central concept is a distributed tensor, represented by class `DistArray`.[1] Tensors of arbitrary *orders* (i.e., the number of dimensions) are supported. Each dimension of

[1] TiledArray is written in the C++ language (the 2011 ISO standard) to allow powerful composition and efficiency; interfaces to other languages can be designed straightforwardly, but with loss of expressiveness and power.

the tensor is *tiled*, i.e., divided into one or more contiguous blocks of arbitrary size. The tiling of the tensor is a Cartesian product of the dimension tilings. The tile data is by default represented as a dense row-major tensor represented by class `Tensor<T>` (`T` can be any (not necessarily commutative) ring type, such as `int`, `double`, `std::complex<float>`, or `boost::math::quaternion<double>`). User can customize the representation for the tile data by providing a custom tensor class as a template parameter to `DistArray`.

`DistArray`'s data is distributed among worker processes. TiledArray uses the MADWorld runtime[116] to support distributed computation. MADWorld builds upon a Message Passing Interface (MPI) library to provide message passing (including active messages), global namespace, task scheduler, and other facilities that support high-level composition of parallel applications. `DistArray` are distributed objects that exist in a (sub)set of MPI processes. The instance of a given `DistArray` on a given process holds only some tiles; the map from tile index to the process rank is governed by a `Pmap` (process map) object that is a member of each `DistArray`. Several types of process maps are provided by TiledArray (replicated, blocked, round-robin, 2-D block-cyclic) and custom maps can be easily constructed by the user.

`DistArray` can represent dense tensors, in which every tile is explicitly present, and block-sparse, in which some or all tiles are omitted. These types of array structure are described by the policy template parameter to `DistArray`; dense and block-sparse arrays correspond to `DensePolicy` and `SparsePolicy`, respectively (users can provide custom policy classes, if needed). Which tiles in a block-sparse array are nonzero is described by its member object of the associated Shape class. In this work we only consider "boolean" shapes (tile is nonzero or zero); in general, shapes return tile norms that, when defined appropriately, can be used to estimate shapes of the arithmetic results from the shapes of arguments.

In addition to the DistArray itself, TiledArray provides high-performance implementation of arithmetic operations on such tensors. This includes unary operations, such as scale, permute indices, and element reductions (e.g. norm computation), and binary operations such as add, subtract, Hadamard product, and contractions. More complex algorithms on tensor data can be implemented seamlessly using the domain-specific language (DSL) for tensor arithmetic. A short demonstration of the TiledArray tensor DSL for computing MP2 energy is presented in the Supporting Information.

Efficiency of tensor contraction is the foremost concern for applications in the domain of the many-body quantum physics. Just like is the case for the well-understood matrix multiplication problem (a special case of tensor contraction), minimizing the data movement across the levels of memory hierarchy is key to high performance, whether the setting is a single-core processor with multi-level cache, or a distributed-memory parallel machine. The tensor contraction in TiledArray is implemented as a distributed-memory matrix multiplication, with appropriate permutations applied to the arguments and the result (if needed). A generic block-sparse distributed matrix multiplication is implemented in TiledArray as a task-based variant of the 2D SUMMA algorithm of van de Geijn and Watts.[112] 2D SUMMA algorithm is popular because it is simple, general, and efficient; in fact, it is *communication optimal* under some conditions.[117]

TiledArray implements SUMMA as a graph of fine-grained tasks scheduled by the MADWorld task scheduler. The task-based formulation allows to hide communication latency and deal with the load imbalance, both factors that limit the strong scaling. This is achieved by overlapping execution of tasks and processing of messages (MADWorld runtime dedicates a thread to process messages in addition to the threads deployed by the MPI library). This is also done by maximally avoiding global synchronization such as collective operations. Thus synchronization between tasks is (almost purely) driven by the data flow. Such fine-grained (at the level of a single tile) decomposition of computation allows to execute in parallel not only tile operations that result from a single arithmetic expression, but also to schedule multiple independent arithmetic operations at the same time, e.g. two products in $a \times b + c \times d$ can be executed simultaneously. The fine-grained task-based design of TiledArray thus permits incorporation of coarse-grained task approaches used to improve parallelism in coupled-cluster implementation in NWChem.[94]

Note that the permutational symmetry of tensors can be incorporated into SUMMA-style algorithms with near-perfect load balance[108] (this is the key innovation of CTF); however task-based formulation helps to alleviate the load imbalance that arises due to sparsity or non-uniform blocking mandated by physics (such as in AO-basis CCSD algorithms as shown later). Excellent strong scaling of TiledArray's task-based matrix multiplication for dense and block-rank-sparse matrices (where some tiles are missing or kept in low-rank form) was demonstrated in Ref. 118.

2.2.2 Formalism

The CCSD(2)$_{\overline{\mathrm{F12}}}$ method[45,46] is an approximation to the explicitly correlated CCSD-F12 method[60,38,61] obtained by second-order perturbation from the standard CCSD. Thus our discussion starts with a recap of the closed-shell CCSD method; the reader is referred to the existing reviews[119,120] and monograph[121] on the coupled-cluster theory.

The coupled-cluster singles and doubles wave function is obtained from the single-determinant reference $|0\rangle$ by the exponentiated cluster operator $\hat{T}$:

$$|\Psi_{\mathrm{CC}}\rangle = e^{\hat{T}} |0\rangle \,. \tag{2.1}$$

The CCSD cluster operator is limited to spin-free single and double substitutions (or, excitations), conveniently defined for closed-shell references in terms of spin-free one- and two-particle replacers E^a_i and E^{ab}_{ij}

$$\hat{T}_{\mathrm{CCSD}} \equiv \hat{T}_1 + \hat{T}_2, \tag{2.2}$$

$$\hat{T}_1 \equiv T^i_a E^a_i, \tag{2.3}$$

$$\hat{T}_2 \equiv \frac{1}{2} T^{ij}_{ab} E^{ab}_{ij}, \tag{2.4}$$

with

$$E^a_i \equiv \sum_{\tau=\alpha,\beta} a^\dagger_{a_\tau} a_{i_\tau}, \tag{2.5}$$

$$E^{ab}_{ij} \equiv \sum_{\tau,\sigma=\alpha,\beta} a^\dagger_{a_\tau} a^\dagger_{b_\sigma} a_{j_\sigma} a_{i_\tau}. \tag{2.6}$$

where operator $a_{p_\tau}/a^\dagger_{p_\tau}$ annihilates/creates an electron in orbital p and spin projection τ ($\tau = \alpha, \beta$ corresponds to $m_s = +1/2, -1/2$). Einstein summation convention is employed throughout (any symbol that appears once as a contravariant and once as a covariant index is summed over). [2]

In our work we follow the generator-state formulation of closed-shell CCSD.[99] The energy and amplitudes equations of CCSD are defined as follows:

$$E_{\mathrm{CCSD}} \equiv \langle 0|\, \bar{H}_{\mathrm{CCSD}}\, |0\rangle = \langle 0|\, (\hat{H} - E_0)(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2)\, |0\rangle \tag{2.7}$$

$$0 = \langle {}^{\overline{a}}_{i}|\, \bar{H}_{\mathrm{CCSD}}\, |0\rangle = \langle {}^{\overline{a}}_{i}|\, (\hat{H} - E_0)(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2\hat{T}_1 + \frac{1}{3!}\hat{T}_1^3)\, |0\rangle \tag{2.8}$$

$$0 = \langle {}^{\overline{ab}}_{ij}|\, \bar{H}_{\mathrm{CCSD}}\, |0\rangle = \langle {}^{\overline{ab}}_{ij}|\, (\hat{H} - E_0)(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2\hat{T}_1 + \frac{1}{3!}\hat{T}_1^3 + \frac{1}{2}\hat{T}_2\hat{T}_1^2 + \frac{1}{4!}\hat{T}_1^4)\, |0\rangle \tag{2.9}$$

where $\bar{H}_{\mathrm{CCSD}} \equiv \exp(-\hat{T}_{\mathrm{CCSD}})\hat{H}\exp(\hat{T}_{\mathrm{CCSD}})$, $\langle {}^{\overline{a}}_{i}| \equiv \langle 0|\,(E^a_i)^\dagger$, $\langle {}^{\overline{ab}}_{ij}| \equiv \langle 0|\,(2E^{ab}_{ij} - E^{ba}_{ij})^\dagger$, and $E^{(0)} \equiv \langle 0|\,\hat{H}\,|0\rangle$. Implementation of the CCSD equations largely follows Scuseria, Janssen, and Schaefer,[87] with the differences described in Section 2.2.3.

The CCSD(2)$_{\overline{\mathrm{F12}}}$ approach[45] corrects the basis set error of CCSD wave function and energy perturbatively. The first-order explicitly correlated correction to the CCSD wave function is defined as

$$|1_{\mathrm{F12}}\rangle \equiv \frac{1}{2}\tilde{R}^{ij}_{\alpha\beta}E^{\alpha\beta}_{ij}\,|0\rangle \tag{2.10}$$

[2] i, j, k, l denote orbitals that are occupied in $|0\rangle$ and are *active* in CCSD. m, n denote *all* occupied orbitals; together with the unoccupied orbitals a, b, c, d they form the full set of orbitals denoted by p, q, r, s (in this work these are simply canonical Hartree-Fock orbitals). $\kappa, \lambda, \mu, \nu$ denote the formal complete set of orbitals that includes the $\{p\}$ set. The orbitals of $\{\kappa\}$ that are not occupied are denoted α, β, γ. Orbitals that approximate α, β, γ in a finite (auxiliary) AO basis and span $\{p\}$ exactly are denoted A', B', C'; the complement to $\{p\}$ is spanned by complementary auxiliary basis set orbitals[28] a', b', c', d'.

where $\tilde{R}$ are:

$$\tilde{R}^{ij}_{\alpha\beta} \equiv \langle \alpha\beta | \hat{Q}_{12} \hat{P}_{12} f(r_{12}) | ij \rangle . \tag{2.11}$$

Projector $\hat{Q}_{12} \equiv 1 - \hat{V}_1 \hat{V}_2$ ensures orthogonality of the geminal basis to the conventional double excitations.[28] Projector $\hat{P}_{12}$ takes care of the spin-dependence of the electron-electron cusp conditions;[29,63] resolving the projector produces

$$\tilde{R}^{pq}_{rs} \equiv \frac{1}{2}(C_0 + C_1) R^{pq}_{rs} + \frac{1}{2}(C_0 - C_1) R^{qp}_{rs} = \frac{3}{8} R^{pq}_{rs} + \frac{1}{8} R^{qp}_{rs}, \tag{2.12}$$

where $C_{0,1} = 1/2, 1/4$ are the cusp coefficients for spin-singlet and spin-triplet pairs[21,122] and R are the matrix elements of projected geminal

$$R^{ij}_{\alpha\beta} \equiv \langle \alpha\beta | \hat{Q}_{12} f(r_{12}) | ij \rangle . \tag{2.13}$$

The standard Ten-no geminal, $f(r_{12}) = -\exp(-\gamma r_{12})/\gamma$,[29] is used in this work; it efficiently models the universal behavior of the electronic wave function at short inter-electronic distances. The second-order correction to the CCSD energy is obtained from $|1_{\mathrm{F12}}\rangle$ as a simplified Hylleraas functional:[46]

$$E^{(2)}_{\mathrm{F12}} \equiv \sum_{i<j} \left(2\mathcal{V}^{ij}_{ij} + \mathcal{B}^{ij}_{ij} \right) \tag{2.14}$$

with

$$\mathcal{V}^{ij}_{ij} \equiv \bar{V}^{ij}_{ij} + (\bar{V}^{ab}_{ij} + \bar{C}^{ab}_{ij}) T^{ij}_{ab} + \hat{S}_{ij} \bar{V}^{ia}_{ij} T^{j}_{a}, \tag{2.15}$$

$$\mathcal{B}^{ij}_{ij} \equiv \bar{B}^{ij}_{ij} - (F^i_i + F^j_j) \bar{X}^{ij}_{ij} \tag{2.16}$$

T^{ij}_{ab} and T^{j}_{a} in Eq. (2.15) are the converged CCSD amplitudes, and tensors V and C are given by

$$V^{pq}_{ij} \equiv \tilde{R}^{\alpha\beta}_{ij} G^{pq}_{\alpha\beta}, \tag{2.17}$$

$$C^{ab}_{ij} \equiv F^{a}_{\alpha} \tilde{R}^{\alpha b}_{ij} + F^{b}_{\alpha} \tilde{R}^{a\alpha}_{ij}, \tag{2.18}$$

where G is the integral of Coulomb operator. Tensors B and X in Eq. (2.16) are defined as

$$B^{ij}_{ij} \equiv \hat{S}_{ij} \tilde{R}^{ij}_{\alpha\beta} F^{\beta}_{\gamma} \tilde{R}^{\alpha\gamma}_{ij}, \tag{2.19}$$

$$X^{ij}_{ij} \equiv \tilde{R}^{ij}_{\alpha\beta} \tilde{R}^{\alpha\beta}_{ij}. \tag{2.20}$$

Bars over tensors denote the closed-shell analog of anti-symmetrization: $\bar{O}^{pq}_{rs} \equiv 2O^{pq}_{rs} - O^{qp}_{rs}$. $\hat{S}_{ij}$ is the index symmetrizer: $\hat{S}_{ij} f(i,j) \equiv f(i,j) + f(j,i)$. Robust approximate evaluation of tensors V, C, X and B in terms of two-electron integrals follows standard CABS approach[28] for V, X, and C, whereas for B CABS-based approximation C is used,[123] as well as the recently proposed approximation D.[80] Detailed expressions have been given elsewhere,[124, 63] and here we only demonstrate the CABS approximation for intermediate V which will be important later:

$$V^{pq}_{ij} \overset{\text{CABS}}{\approx} (\tilde{GR})^{pq}_{ij} - G^{pq}_{rs} \tilde{R}^{rs}_{ij} - \hat{S}_{ma'} G^{pq}_{ma'} \tilde{R}^{ma'}_{ij}, \tag{2.21}$$

where GR is the integral of $f(r_{12})/r_{12}$. The computational complexity of $E^{(2)}_{\text{F12}}$ is governed by $\mathcal{O}(N^6)$ cost of computing $\bar{V}^{ab}_{ij}$, although in practice the cost of AO integral evaluation and transformation is also significant. Finally, the total CCSD(2)$_{\overline{\text{F12}}}$ energy has three contributions

$$E_{\text{CCSD(2)}_{\overline{\text{F12}}}} \equiv E_{\text{CCSD}} + E^{(2)}_{\text{F12}} + E^{(2)}_{\text{S}}. \tag{2.22}$$

$E_{\mathrm{S}}^{(2)}$ is the "CABS singles" contribution that correct the basis set error from Hartree-Fock energy:[43]

$$E_{\mathrm{S}}^{(2)} \equiv 2t_{A'}^{i} F_{i}^{A'} \tag{2.23}$$

in which the singles amplitudes $t_{A'}^{i}$ are obtained by solving a system of linear equations

$$F_{i}^{A'} + t_{i}^{B'} F_{B'}^{A'} - t_{j}^{A'} F_{i}^{j} = 0. \tag{2.24}$$

2.2.3 Implementation

The conventional CCSD and CCSD$(2)_{\overline{\mathrm{F12}}}$ approaches are implemented using TiledArray to represent all AO and MO tensors. All tensors with more than 1 index are fully distributed across the entire MPI process group that was used to initialize the computation.

AO basis tensors are implemented as TiledArray's `DistArray` tensors parametrized by a custom tile. AO basis is tiled by groups of atoms (see Ref. 125 for details of how molecules are clustered). Each tile is evaluated locally on the node on which it resides; tile's compute method computes shell-blocks of one- and two-electron AO-basis integrals using the Libint library (version 2.1.0).[126]

MO basis tensors are implemented as TiledArray's `DistArray` tensors with standard implementation of in-memory tiles (class `TA::Tensor`). MO dimensions are uniformly-tiled using user-specified tile sizes for occupied and unoccupied dimensions (the last tile of each dimension may be smaller or greater than the user-specified tile size). Integral transformation of

2-electron integral tensors from AO to (canonical) MO basis is performed in conventional (using order-4 AO tensors) or density fitting manner. Both methods for computing MO integrals cost $\mathcal{O}\left(N^5\right)$, but the density fitting route affords significant computational savings for basis sets with high angular momenta by reducing the cost of computing AO integrals. Furthermore, modern reduced-scaling coupled-cluster approaches utilize density fitting throughout to reduce the cost of integral computation and transformation; a density fitting canonical CC formulation is thus necessary to be able to compare directly to the reduced-scaling CC counterparts.

The CCSD equations were evaluated following the formulation of Scuseria, Janssen, and Schaefer,[87] using the standard Jacobi solver with DIIS acceleration.[127] Evaluation of the doubles amplitude residual (Eq. (2.9)) has a $\mathcal{O}\left(N^6\right)$ cost; specifically, the $\mathcal{O}\left(N^6\right)$ floating-point operation count is $\frac{1}{4}O^2V^4 + 4O^3V^3 + \frac{1}{2}O^4V^2$ where O and V stands for number of occupied and unoccupied orbitals, respectively (the convention in the electronic structure literature treats one floating-point multiply and one addition as a single floating-point operation (FPO)). Our implementation in MPQC does not take advantage of the permutational symmetry, hence the total operation cost of the doubles residual is $O^2V^4 + 6O^3V^3 + O^4V^2$. With density fitting approximation, one O^3V^3 term is reduced. The total cost becomes $O^2V^4 + 5O^3V^3 + O^4V^2$ FPO per iteration. Hence, the operation cost of our CCSD implementation is greater than optimal. However, despite this drawback the high efficiency and strong scalability of our implementation makes it competitive to existing optimized implementation, as we demonstrate in Section 2.3.1.

The conventional CCSD approach requires computing MO basis integrals with more than two unoccupied indices. Since typically $V >> O$, the size for these integrals greatly exceed the size of the doubles amplitudes and thus dominate the storage requirement of CCSD. To reduce the storage we utilize a *hybrid* scheme (denoted as AO-CCSD in this paper) that combines some elements of the AO-integral-direct approach, introduced by Rendell *et al.*,[86] with the density fitting factorization of MO integrals to avoid storage of MO integrals with 3 and 4 unoccupied indices. In the AO-integral-driven approach of Rendell *et al.*[86] the following intermediates are evaluated in AO-basis using Coulomb integrals G computed and consumed on-the-fly:

$$(U_1)^{ij}_{\rho\sigma} \equiv (T^{ij}_{cd} + T^i_c T^j_d)(C^c_\mu C^d_\nu) G^{\mu\nu}_{\rho\sigma} \tag{2.25}$$

$$(U_2)^{ij}_{\rho\sigma} \equiv (T^i_c C^c_\mu C^j_\nu) G^{\mu\nu}_{\rho\sigma} \tag{2.26}$$

$$(U_3)^{ij}_{\nu\sigma} \equiv (T^i_c C^c_\mu C^j_\rho) G^{\mu\nu}_{\rho\sigma} \tag{2.27}$$

By utilizing these three intermediates, MO integrals with more than two unoccupied indices can be avoided. For example, term $G^{cj}_{ab} T^i_c$ can be evaluated via U_2:

$$G^{cj}_{ab} T^i_c = C^\rho_a C^\sigma_b (U_2)^{ij}_{\rho\sigma} \tag{2.28}$$

In our implementation, the U_1 term is computed using 4-index AO integrals, just like in the approach of Rendell *et al.*. The U_2 and U_3 terms are avoided via the density fitting approximation, e.g. Eq. (2.28) becomes:

$$G^{cj}_{ab} T^i_c = X^\Lambda_{jb} (X^\Lambda_{ac} T^i_c) \tag{2.29}$$

where X is the half transformed three center integral and Λ is the density fitting basis:

$$X^{\Lambda}_{pq} = C^{\rho}_{p}C^{\sigma}_{q}(\rho\sigma|\Gamma)(\Gamma|\Lambda)^{-\frac{1}{2}} \tag{2.30}$$

This leads to total cost to $O^2N^4 + 5O^3V^3 + O^4V^2$, where N is the number of total molecular orbitals (this excludes the cost of the AO integral evaluation). Usually the O^2N^4 term dominates the cost of the whole calculation. All the other intermediates in CCSD are computed from stored MO integrals with 0, 1, and 2 unoccupied indices, evaluated by density fitting.

Evaluation of the second-order F12 correction to the CCSD energy involves 4 index integrals (or intermediates) with no more than 2 unoccupied indices; the lone exceptions are the V^{ab}_{ij} and V^{ia}_{ij} intermediates whose evaluation involves integrals with 3 and 4 unoccupied indices. To avoid storage of these integrals we use the hybrid AO-DF described above. For example, the $V^{ab}_{ij}T^{ab}_{ij}$ contribution to Eq. (2.15),

$$V^{ab}_{ij}T^{ij}_{ab} \equiv ((\tilde{GR})^{ab}_{ij} - G^{ab}_{pq}\tilde{R}^{pq}_{ij} - \hat{S}_{ij}G^{ab}_{ma'}\tilde{R}^{ma'}_{ij})T^{ij}_{ab} \tag{2.31}$$

Storage of G^{ab}_{pq} can be avoided by using the integral-direct evaluation, similar to Eq. (2.25). The total cost to evaluate this term would be O^2N^4 FPO.

2.3 Results & Discussion

2.3.1 Performance

The CCSD and CCSD$(2)_{\overline{\mathrm{F12}}}$ energy evaluation was implemented in a developmental version of Massively Parallel Quantum Chemistry package (MPQC), version 4. The performance was measured on a standalone multicore workstation, a medium-size commodity cluster, and a national-scale supercomputer.

Shared-Memory Multiprocessor

We tested the multiprocessor performance on a single node of the "BlueRidge" cluster hosted by Virginia Tech Advanced Research Computing (VT ARC). Each node of BlueRidge has two Intel (Sandy Bridge) Xeon E5-2670 CPUs (total of 16 physical cores), for a theoretical peak of 332 GFLOPS, and 64 GB of RAM.

The performance was compared to two popular freely-available software packages that implement CCSD (Psi4 1.0) as well as the explicitly correlated CCSD (ORCA 3.0). Shared-memory parallelism in Psi4's CCSD codes is only available by using threaded version of BLAS, whereas ORCA on a shared memory system uses MPI (hence in the ORCA context the number of threads will mean the number of MPI processes). MPQC and TiledArray were compiled using Intel Parallel Studio XE 15.3, with serial BLAS from MKL 11.2.3 and Pthreads-based task queue implementation in MADWorld. Psi4 was compiled using Intel

Parallel Studio XE 15.3 along with parallel MKL 11.2.3 and Intel OpenMP. ORCA was compiled with Intel Parallel Studio XE 13.1 using parallel MKL 11 and OpenMPI 1.6.5.

Figure 2.1 presents the CCSD energy performance vs. the number of threads for a $(H_2O)_{10}$ cluster with the cc-pVDZ basis (frozen core, $O = 40$, $V = 190$). The block-size was set as 4 for occupied and 36 for unoccupied. MPQC is the slowest traditional (no DF, MO-only) CCSD code with 1 thread: 1.4 times slower than ORCA and 2.3 times slower than Psi4. This is expected because MPQC is doing at least twice more work than ORCA and Psi4 without using permutation symmetry (ORCA's CCSD uses the formulation of Hampel *et al.*,[98] whereas Psi4's original codes follows the formulation of Stanton, Gauss, Watts, and Bartlett[128] both are properly spin-adapted and use proper advantage of the permutational symmetry). Our implementation demonstrates superlinear scaling with the number of threads: a 20.7x speedup is observed with 16 threads relative to 1 thread. The standard CCSD code in Psi4 scales poorly, with only 3.4 times speedup at 16 threads. MPQC CCSD is already faster than Psi4 at 8 cores, and 2.7 times faster than Psi4 at 16 threads. On contrast, ORCA has 18 times speedup and is still 1.2 times faster than MPQC at 16 threads.

Psi4 also has a separate implementation of DF-based CCSD[129] which follows Ref. 130. The DF-CCSD Psi4 code has excellent thread scaling (14.5x speedup on 16 threads) and is the most efficient code for this test case. Our DF-based implementation of CCSD is 1.3 times slower than Psi4 on 1 thread, but is nearly as fast as Psi4 on 16 threads, with a 16.9x times speedup(MPQC DF-CCSD takes 66 seconds per iteration while Psi4 takes 60 seconds). Both DF-CCSD codes are substantially faster than their conventional counterparts.

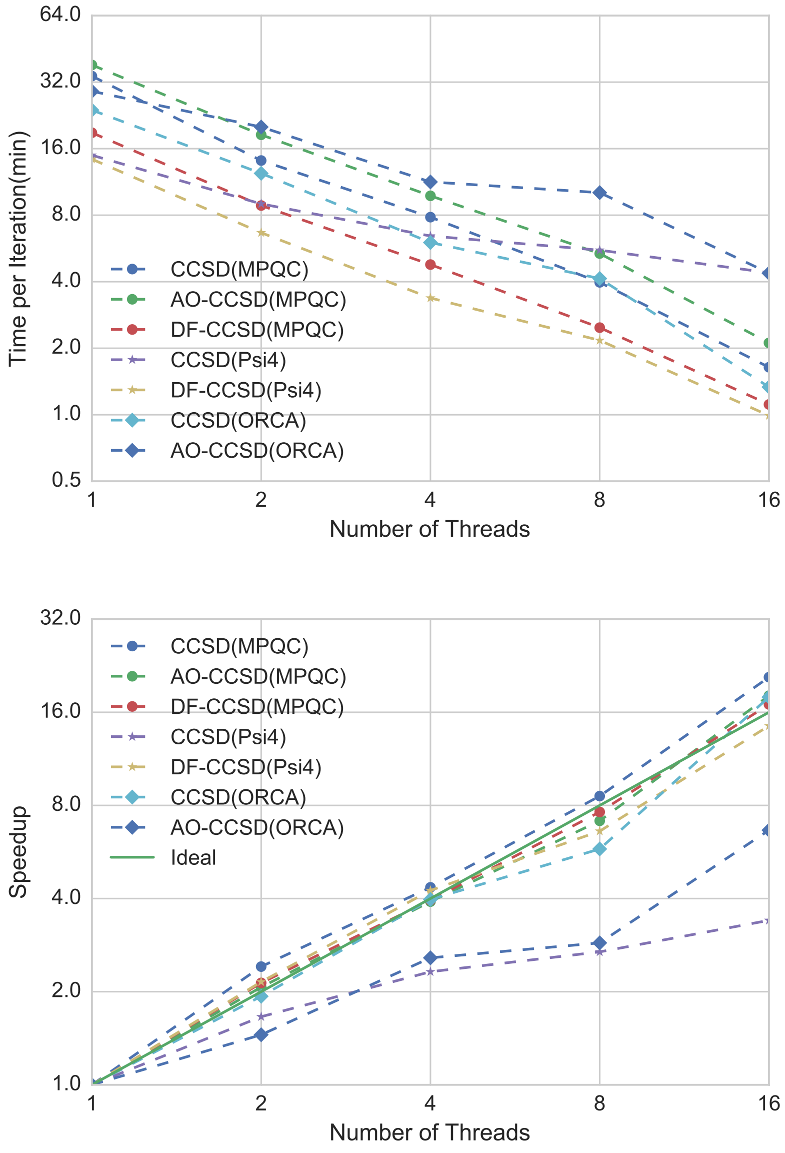

Figure 2.1: Performance comparison of the present and existing implementations of CCSD (frozen-core CCSD on a $(H_2O)_{10}$ cluster in cc-pVDZ basis).

The AO-CCSD implementation in MPQC shows a 18.0x speedup using 16 threads, while ORCA's AO-CCSD implementation (which unlike ours does not use density fitting anywhere) shows a speedup of 6.6x. The integral-direct implementation in MPQC is 2.1 times faster than ORCA on 16 threads. Due to the increase cost and expense of computing two electron integral at each iteration, the AO-CCSD approach in MPQC is slightly slower than the conventional CCSD. As we will show later, for larger systems the time spent on computing integrals will contribute less to the total time, since its $\mathcal{O}(N^4)$ scaling (without accounting for sparsity) is lower than $\mathcal{O}(N^6)$ scaling of CCSD. This approach with greatly reduced storage requirements will warrant its use in large-system computations.

The computational cost of the explicitly correlated terms is small relative to that of CCSD itself, hence the scaling of the $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ implementation is largely determined by the scaling of the CCSD. Nevertheless, we compared the single-node performance of the F12-only contributions to the $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ energy in MPQC and ORCA. This test was performed on a $(H_2O)_6$ cluster using the cc-pVDZ-F12 basis set specifically designed for explicitly correlated computations[131] and the matching CABS basis set[132] (288 basis functions and 660 CABS functions). The occupied block-size was set to 8 and the unoccupied block-size was set to 32. The total time and speedup of F12 part were plotted against the number of threads, as shown in Figure 2.2. The MPQC implementation again exhibits a superlinear 22x speedup when increasing the number of threads from 1 to 16, while ORCA shows an excellent speedup of 10.4x. Although the MPQC is 2.3 times slower than ORCA on 1 thread, their performance is close on 16 threads, with MPQC only 10% times slower than ORCA, despite the higher

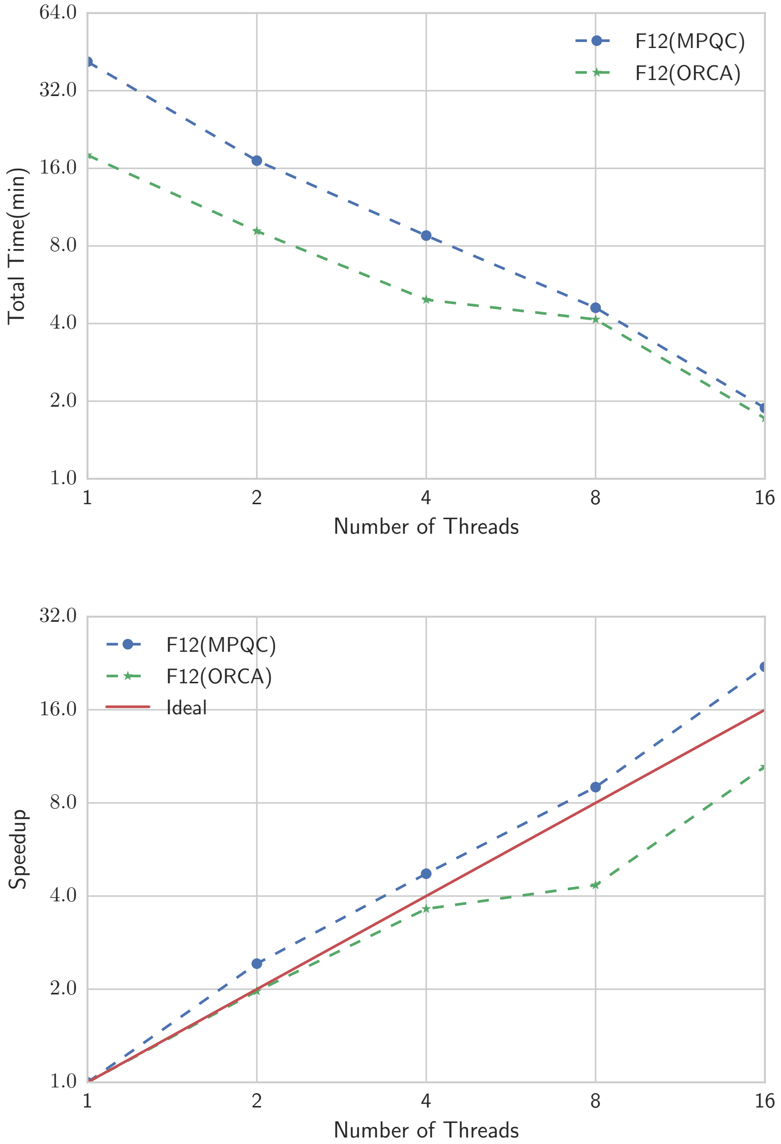

Figure 2.2: Parallel performance of the F12 part of the $CCSD(2)_{\overline{F12}}$ computation on a $(H_2O)_6$ cluster (frozen-core, cc-pVDZ-F12 basis).

operation count of MPQC.

Despite the lack of support for permutational symmetry, the CCSD and CCSD$(2)_{\overline{\text{F12}}}$ implementations in MPQC are competitive to the existing state-of-the-art implementations on a single-node computer. Having established high base efficiency of our implementations, we will next examine their parallel scalability on distributed-memory systems.

Commodity Cluster

VT ARC's Blueridge is a "commodity" cluster in the sense that the nodes (dual-socket x86-based blades), network (Infiniband), and software are all widely available and used. This platform is therefore a representative test setting available to a typical research group. In this benchmark, MPQC and TiledArray were compiled with Intel Parallel Studio XE 15.3 and Intel MPI 5.0. Serial version of BLAS in MKL 11.2.3 were used. MADWorld queue implementation used Intel Thread Building Blocks (TBB) 4.3.3 library. All MPQC calculations used 16 active compute threads on each node. Due to significant memory requirements of the examples in this section, all 1-node computations in this section used "fat" nodes with 128 GB of RAM, whereas the rest of computations used the standard (64 GB) nodes.

Figure 2.3 describes the performance of CCSD$(2)_{\overline{\text{F12}}}$ energy computation for a $(\text{H}_2\text{O})_{10}$ cluster vs. the number of nodes. The cc-pVDZ-F12 basis set was used (N = 480, O = 40, V = 430). The matching cc-pVDZ-F12/OptRI CABS basis set has 1100 functions, and the aug-cc-pVDZ-RI basis set was used for density fitting (1180 basis functions). Occupied block-size was set to 8 while unoccupied block-size was set to 32. Integral-direct formulation was utilized in the CCSD and F12 parts of the computation. The CCSD implementation demonstrates good strong scaling: 10.4x speedup is achieved on 16 nodes, relative to 1 node, and 16.4x on 32 nodes ($\sim 65\%$ and $\sim 51\%$ parallel efficiency, respectively). The F12 part does not scale as well as CCSD: a 5.0x speedup is attained on 16 nodes (relative to 1 node) and 5.7x on 32 nodes. Since the noniterative F12 energy contribution is much cheaper than the iterative CCSD calculation, the overall scaling of $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ is dominated by CCSD; $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ exhibits a 58% parallel efficiency on 16 nodes and 42% on 32 nodes.

It is useful to compare the performance of the standard CCSD algorithm in MPQC with the state-of-the-art implementation of the TCE CCSD in NWChem. Although NWChem does not have an implementation of the explicitly-correlated CCSD, it currently offers a much broader set of functionality than MPQC. NWChem 6.6 was compiled with the same compiler and MPI library as MPQC; however, a multithreaded MKL 11.2.3 implementation of BLAS was used. Each node executed 16 NWChem processes. NWChem used tilesize 40 in all computations, whereas MPQC used blocksize 8 for the occupied space and 32 for the unoccupied space.

Table 2.1 presents the measured performance of MPQC and NWChem for valence CCSD computations on a 10-water cluster with the cc-pVDZ basis (the relatively small system and basis had to be used since the NWChem TCE CCSD implementation assumes that all integrals can be stored in memory). For this relatively small example MPQC's traditional CCSD code is significantly faster than the NWChem TCE CCSD code. Some of the difference is probably due to the use of spin-free CCSD formulation in MPQC, while NWChem's TCE generated CCSD program uses the spin-orbital formulation with partial utilization of the spin symmetry. Note that the performance we obtained for the NWChem is in reasonable agreement with the performance data reported at `http://www.nwchem-sw.org/index.php/Benchmarks` .

Table 2.1: Parallel performance of valence CCSD (seconds/iteration) in MPQC and NWChem for $(H_2O)_{10}$ in the cc-pVDZ basis on the BlueRidge cluster.

Nodes	1	2	4	8
MPQC	50	34	27	18
NWChem	242	229	131	113

It is also useful to examine the weak scalability of our implementation. An algorithm has perfect weak scaling if increasing the problem size and the number of processing units by a factor of x results in no change in the execution time. Due to the lack of global instrumentation for operation counting in TiledArray, we performed a weak scaling analysis of the floating operation throughput in the O^2V^4 term in the CCSD doubles equations (evaluation of this term is usually the rate-limiting step in CCSD). The transformed integrals were stored in memory (i.e. the integral-direct algorithm was not utilized). We performed a series of frozen-code cc-pVDZ CCSD computations on water clusters with $10+2n$ molecules ($n = 0..5$) using $N = 2^n$ nodes. The measured effective floating point operation (FLOP) throughput per node are shown in Figure 2.4. On 1 node, MPQC is able to reach around 85% of the machine peak, which is close to the practical maximum performance of the vendor-provided

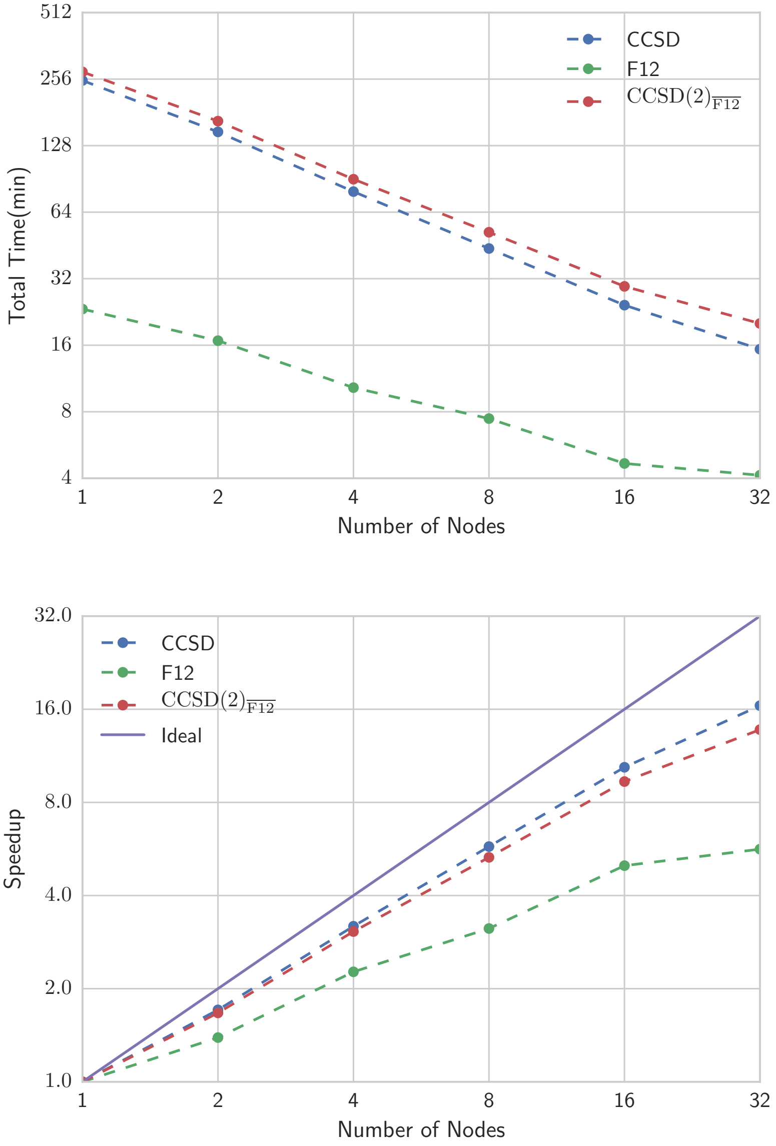

Figure 2.3: $\text{CCSD(2)}_{\overline{\text{F12}}}$ total time and strong scaling of $(H_2O)_{10}$ cluster on the Blueridge cluster.

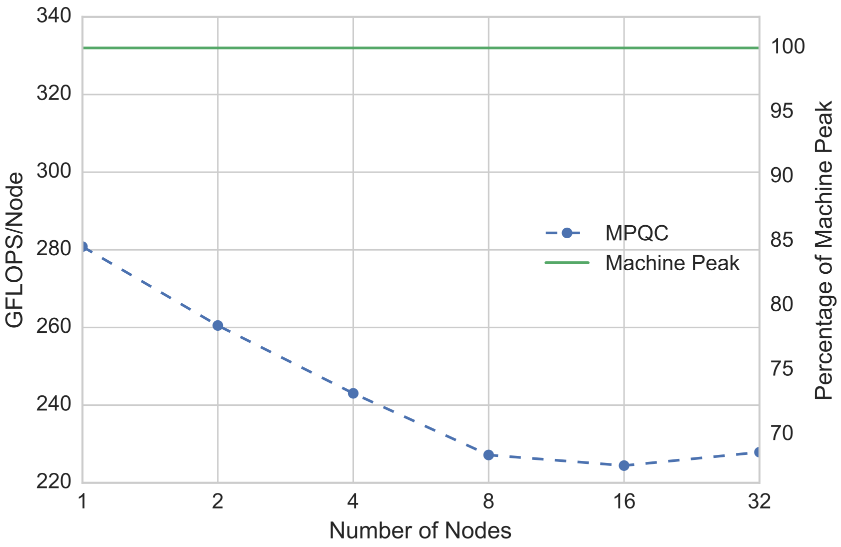

Figure 2.4: Weak scaling of the O^2V^4 term in the CCSD doubles equations on the BlueRidge cluster. See the text for details.

BLAS DGEMM routine on this cluster. Excellent weak scaling is observed as the problem size grows from $(H_2O)_{10}$ (1 node) to $(H_2O)_{20}$ (32 nodes), as the throughput decreases from 85 % to 70% of the machine peak.

Table 2.2: CCSD performance data for different systems on the Blueridge cluster

Molecule	Basis Set	Act Occ [a]	Total Orbitals	Nodes	Time/Iteration(min)
Uracil dimer	cc-pVDZ-F12	42	552	12	5
	cc-pVTZ-F12		992	12	40
	cc-pVQZ-F12		1664	32	110
Tamoxifen	aug-cc-pVDZ	75	928	24	48
β-Carotene	mTZ[b]	108	1032	32	100

[a] Active occupied orbitals [b] 6-31G basis functions for H atom and cc-pVTZ basis functions for C atom with f-component removed.[95]

Despite the use of RAM for storage of all quantities and the current lack of permutational symmetry handling, the ability to avoid of integrals with 3 and 4 unoccupied indices by the hybrid AO-integral-direct and density fitting formulation allows fairly large $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ computations to be performed on modest computational resources. Table 2.2 demonstrates representative CCSD timings (which dominate the overall $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ cost) for a few systems with as few as 24 and as many as 96 atoms. The largest computation with respect to the basis set size, on uracil dimer with the cc-pVQZ-F12 basis set ($N = 1664$), would demand 6.7 TB to store G^{ab}_{cd} with the full utilization of permutational symmetry, but can be carried out easily on 32 nodes with 64 GB of RAM each. The time per iteration is 110 minutes on 32 nodes and the whole calculation can be done in one day. The largest computation with respect to the number of atoms, on β-carotene, is identical to the NWChem TCE-based CCSD computation reported by Hu *et al.* in Ref. 95. The reported NWChem timing on 768 cores was 6877 sec per iteration, whereas our computation on 512 cores takes $\sim$ 6000 seconds per iteration (coincidentally, our work and Ref. 95 used identical CPU cores), despite our use of AO-integral-direct formulation.

National-Scale Supercomputer

To be able to test reduced-scaling coupled-cluster methods on largest possible molecules, it is necessary to deploy conventional implementation to the largest machines of today and tomorrow. Therefore we tested the scalability of our $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ implementation on the IBM BlueGene/Q computers hosted by Argonne Leadership Computing Facility. Each BG/Q node has a single PowerPC A2 processor with 16 1.6 GHz compute cores, for a total of 204.8 GFLOPS peak per node, and 16 GB memory. The strong scaling study was performed on "Cetus", a 4096-node testing platform, which allows computations on fewer than 512 nodes. Performance highlight computations utilized "Mira", the 49152-node production machine at ALCF. The bgclang 3.9.0, a version of LLVM/Clang customized for BG/Q system, was used to compile MPQC and TiledArray. The default compiler driver were used. The serial version of IBM's ESSL 5.1 was used for MPQC and TiledArray. 16 threads/node were used, unless noted.

We have tested strong scaling of CCSD$(2)_{\overline{\rm F12}}$ on a relatively small system, a $(H_2O)_{12}$ cluster with the cc-pVDZ-F12 basis ($N = 576$, $O = 48$, $V = 516$, 1320 CABS functions, 1416 density fitting functions). The block-size was set as 8 for occupied orbitals and 16 for unoccupied orbitals. The performance of the conventional density fitting and AO-based CCSD$(2)_{\overline{\rm F12}}$ on 128, 256, 512, and 1024 nodes are shown in Figure 2.5. Both approaches maintain 50% parallel efficiency on 512 nodes (67% and 52%) relative to the 128-node computation. The parallel efficiency drops below 50% on 1024 nodes (16,384 cores).

Table 2.3: CCSD Performance data for different systems on Mira

Molecule	Basis Set	Act Occ [a]	Total Orbitals	Nodes	Time/Iteration(min)
C_{60}	cc-pVDZ	180	840	512	41
				1024	18
C_{84}	cc-pVDZ	168	1092	512	144
				1024	75

[a] Active occupied orbitals

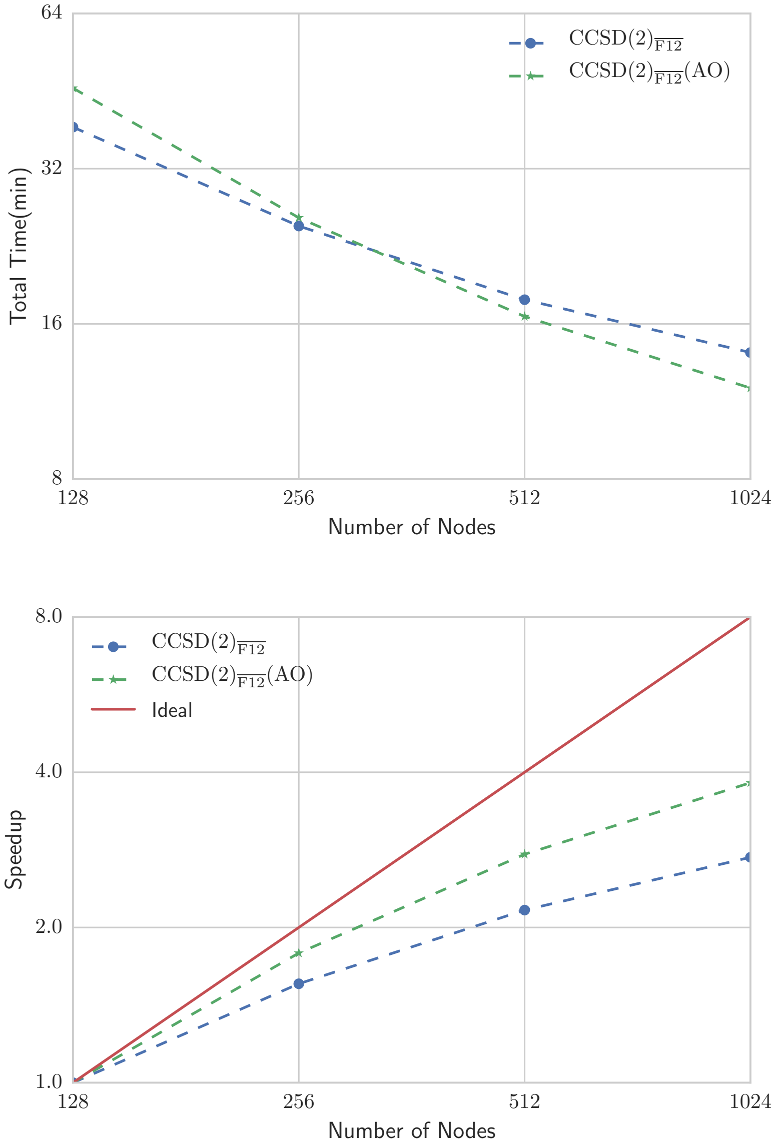

Figure 2.5: CCSD(2)$_{\overline{\mathrm{F12}}}$ total time and strong scaling of $(\mathrm{H_2O})_{12}$ cluster on Mira

Computations on much larger systems can be carried out on Mira with our implementation (see Table 2.3). To demonstrate this we performed CCSD computations on buckminsterfullerene C_{60} and a nanotube fragment C_{84}. With the cc-pVDZ basis set these have $O = \{180, 168\}$ and $V = \{660, 924\}$ (core orbitals were frozen in the nanotube computation). In both cases near-ideal speedups were observed from 512 nodes to 1024 nodes.

2.3.2 Reference Application

Non-covalent interactions play a crucial role in physics, chemistry, and biology as they govern structure and properties of liquids, molecular solids, and biopolymers. Among the variety of non-covalent interactions,[133] the π-π interactions, involving the attraction between conjugated π systems stacked next to each other, are particularly challenging for computational description due to the delicate balance of multiple physics. Accurate description of the energetics of π-π interactions requires high levels *ab initio* treatment(CCSD(T)) of the electronic structure *and* extended basis sets approaching the complete basis set limit.[134]

To demonstrate the utility of the strongly-scalable CCSD$(2)_{\overline{\mathrm{F12}}}$ implementation we computed a precise reference value for the CCSD contribution to the binding energy for the π-stacked uracil dimer, one of the most challenging systems in the standard S66 benchmark dataset of noncovalent interactions.[3] The new reference allowed us to resolve a discrepancy between two reduced-scaling CCSD F12 computations for the binding energy in this system. The standard *composite* approach is to approximate the complete basis set limit of a property computed with the reference high-order method X (e.g., X=CCSD(T)) as a sum of the CBS MP2 property estimate (typically obtained by extrapolation) and the X-MP2 difference computed in an affordable (small) basis. The S66 database[3,135] estimated the MP2 CBS limit for the binding energy by extrapolation from aug-cc-pVTZ and aug-cc-pVQZ basis sets, with the CCSD(T)-MP2 difference computed in aug-cc-pVDZ basis;[3] all computations utilized counterpoise correction. The CCSD binding energy for the uracil dimer was reported as -8.12 kcal/mol (this value corresponds to the CBS estimate obtained according to Ref. 135) of which the correlation CCSD contribution is -8.50 kcal/mol.

To establish the basis set limit for the CCSD correlation contribution to the binding energy of the uracil dimer we used our new CCSD implementation to compute binding energies with quadruple-zeta basis sets (to our knowledge, this is the first time this is done for the π-stacked systems in the S66 data set). All calculations used density fitting basis set aug-cc-pV(X+1)Z-RI, where X is the cardinal number of conventional basis set used. Table 2.4 presents the CCSD CBS estimates obtained with a composite approach in which the MP2 CBS limit approximated by the MP2-F12 cc-pVQZ-F12 value and the CCSD-MP2 difference is computed explicitly with a given basis (all computations use frozen core approximation). The CCSD CBS limit estimates obtained from counterpoise-corrected energies with aug-cc-pVXZ and cc-pVXZ-F12 basis set families agree to 0.01 kcal/mol (the aug-cc-pVQZ and cc-pVQZ-F12 estimates are -8.292 and -8.283 kcal/mol, respectively). However, without

[3] The CCSD(T)-MP2 contribution was later improved[135] by employing extrapolated haug-cc-pV{D,T}Z extrapolated estimate (where 'haug' denotes augmented functions on non-hydrogen atoms only); unfortunately the CCSD contribution was not reported separately.

the counterpoise corrections the two families are not consistent with each other, and the estimates differ from the CP-based estimates by more than 0.2 kcal/mol!

Table 2.4: CCSD correlation binding energy(kcal/mol) of uracil dimer with and without counterpoise correction

Method	Basis Set	CCSD	CCSD(CBS)
CP	aug-cc-pVDZ	-7.171	-8.518
	aug-cc-pVTZ	-7.802	-8.320
	aug-cc-pVQZ	-8.000	-8.292
	cc-pVDZ-F12	-6.758	-8.522
	cc-pVTZ-F12	-7.655	-8.309
	cc-pVQZ-F12	-7.988	-8.283
nonCP	aug-cc-pVDZ	-11.234	-8.339
	aug-cc-pVTZ	-9.680	-8.158
	aug-cc-pVQZ	-8.708	-8.029
	cc-pVDZ-F12	-8.921	-8.506
	cc-pVTZ-F12	-8.486	-8.201
	cc-pVQZ-F12	-8.363	-8.114

Table 2.5: $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ correlation binding energy(kcal/mol) of uracil dimer

Basis Set	$\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$	$\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}\mathrm{(CBS)}$
cc-pVDZ-F12	-8.225	-8.312
cc-pVTZ-F12	-8.331	-8.301

To unequivocally resolve the discrepancy between the CBS CCSD estimates, we computed explicitly the $\mathrm{CCSD(2)}_{\overline{\mathrm{F12}}}$ binding energies for uracil dimer without the counterpoise correction (use of such corrections with the explicitly correlated methods often increases the basis set errors; for a more nuanced discussion see a recent paper by Brauer *et al.*[136]). In all explicitly correlated calculations, we used the geminal exponents 0.9 for cc-pVDZ-F12, 1.0 for cc-pVTZ-F12 and 1.1 for cc-pVQZ-F12, as recommended in Ref 137. The corresponding complementary auxiliary(CABS) basis set and density fitting basis set(aug-cc-pV(X+1)Z-RI) were used. The explicitly correlated CCSD correlation contributions are reported in Table 2.5. To gauge the basis set error of these predictions we corrected the residual basis set incompleteness of the explicitly correlated CCSD energies in the spirit of the composite scheme by adding the MP2 basis set incompleteness estimated against the MP2-F12 cc-pVQZ-F12 reference. Both cc-pVDZ-F12 and cc-pVTZ-F12 CCSD$(2)_{\overline{\text{F12}}}$ binding energies only differ by $\sim$ 0.1 kcal/mol, and after compensating for the basis set incompleteness the estimates agree with each other to 0.01 kcal/mol. The basis set incompleteness of the cc-pVTZ-F12 CCSD$(2)_{\overline{\text{F12}}}$ binding energy appears to be smaller than 0.03 kcal/mol. Most importantly, all estimates agree with the prior estimate obtained with CP-corrected non-F12 CCSD to within 0.02 kcal/mol. Thus our best estimate of the CCSD contribution to the binding energy of uracil dimer (at the S66 geometry) is $\mathbf{-8.30 \pm 0.02}$ kcal/mol. By combining with the $\mathbf{+0.377}$ kcal/mol HF CBS value obtained from an essentially exact multiresolution HF computation,[138] we obtain the best estimate for the CCSD binding energy of $\mathbf{-7.92 \pm 0.02}$ kcal/mol.

Note that the use of explicitly correlated CCSD provides significant cost savings relative to the standard CCSD, e.g. CCSD$(2)_{\overline{\text{F12}}}$ with the cc-pVTZ-F12 basis set is more than 5 times cheaper, has a smaller basis set error, and requires much less memory than CCSD with the cc-pVQZ-F12 basis set.

The definitive estimate of the CCSD binding energy is useful to evaluate the published predictions obtained with two non-CP-corrected reduced-scaling explicitly correlated CCSD methods. In 2014 some of us predicted the CCSD contribution to be -8.30 kcal/mol using the LPNO-CCSD$(2)_{\overline{\mathrm{F12}}}$ method with cc-pVDZ-F12 basis set.[83] This result is in a good (and, probably, somewhat fortuitous) agreement with the canonical (full-scaling) cc-pVDZ-F12 CCSD$(2)_{\overline{\mathrm{F12}}}$ value reported here (-8.225 kcal/mol, Table 2.5). Correcting the cc-pVDZ-F12 LPNO-CCSD$(2)_{\overline{\mathrm{F12}}}$ value for the residual basis set incompleteness by adding the (cc-pVQZ-F12 - cc-pVDZ-F12) MP2-F12 difference we obtain the CBS limit estimate of -8.39 kcal/mol. Later in 2014 Schmitz, Hättig and Tew predicted the CCSD correlation binding energy to be -7.99 kcal/mol with their PNO-CCSD[F12] method,[84] with aug-cc-pV$\{D, T\}$Z basis set extrapolation. The authors suggested that the S66 CCSD binding energy estimate, -8.50 kcal/mol, was in error and recommended their predictions for the CCSD binding energy as the new reference. Our new estimate, -8.30 kcal/mol, suggests that neither the original estimate of Řezáč *et al.*[135] nor the PNO-CCSD[F12] value of Schmitz *et al.* are within 0.2 kcal/mol of the numerical CCSD limit.

As this paper was being finalized for publication, Brauer *et al.*[139] published an extensive revision of the S66x8 and S66 database sets using explicitly correlated coupled-cluster methods, with basis sets as large as cc-pVTZ-F12. It is not possible to compare to their data directly as they reevaluated the equilibrium geometries using the new CBS CCSD(T) estimates for the S66x8 data set. They reported a value for the CCSD binding energy of uracil dimer at the S66x8($1r_e$) of -7.96 kcal/mol obtained with the CCSD-F12b method[43] with a cc-pVTZ-F12 basis set, which is in an excellent agreement with our prediction of −7.92 kcal/mol at the S66 geometry.

2.4 Conclusions

We presented a new massively parallel implementation of standard and explicitly correlated (F12) coupled-cluster singles and doubles (CCSD) with canonical $\mathcal{O}\left(N^6\right)$ computational complexity. The implementation is based on a flexible TiledArray tensor framework that allows natural high-level composition of tensor expressions. Generic design of TiledArray allows implementation of conventional, density fitting, and integral-direct variants of standard and explicitly correlated CCSD with modest effort. The biggest disadvantage of our implementation is the lack of support for permutational and point group symmetries (implementation of the support for symmetry, including non-Abelian, is relatively straightforward and is underway). Our implementation aims at high-end parallel scalability, thus all data greater than $\mathcal{O}\left(N\right)$ is distributed in memory. Nevertheless, to extend the range of systems that can be studied on modest clusters we combined density fitting and AO-integral-driven formulations to avoid storage of tensors with 3 and 4 unoccupied indices.

The performance on a shared-memory computer was found to be competitive with the freely-available CCSD implementations in ORCA and Psi4. Excellent strong scaling was demonstrated on a multicore shared-memory computer, a commodity distributed-memory computer, and a national-scale supercomputer. Large-scale parallel explicitly correlated coupled-cluster implementation makes routine of accurate estimation of the coupled-cluster basis set limit for molecules with 20 or more atoms. Thus, this can provide valuable benchmarks for the merging reduced-scaling coupled-cluster approaches. For example, we re-estimated the basis set limit for the CCSD contribution to the binding energy of π-stacked uracil dimer, a challenging paradigm of π-stacking interactions from the S66[3] benchmark database, using quadruple-zeta basis sets. The revised value for the CCSD correlation binding energy, -8.30 kcal/mol, is significantly different from the S66 reference value, -8.50 kcal/mol, as well as an estimate by Schmitz *et al.* obtained from a reduced-scaling explicitly correlated CCSD approach.[84]

Chapter 3

Efficient Massively Parallel Coupled-Cluster Singles, Doubles, and Perturbative Triples Correction with Density Fitting Approximation

3.1 Introduction

The coupled-cluster method[140,141] is a benchmark molecular electronic structure approach that can attain unparalleled accuracy rivaling that of precise experimental techniques[142] for molecular electronic energies and properties at relatively low orders of truncation. The coupled-cluster singles and doubles with perturbative triples method, or CCSD(T),[59, 143, 144] is often referred to as the "gold standard" of quantum chemistry by balancing the $\mathcal{O}\left(N^7\right)$ operation and $\mathcal{O}\left(N^4\right)$ space complexities while – in the weak correlation regime – substantially improving on CCSD, which has the same storage but only a $\mathcal{O}\left(N^6\right)$ operation complexity. Although for many problems CCSD(T) is not sufficiently accurate,[145, 146] it is a qualitative improvement on the density functional theory for vast areas of molecular simulation and thus remains a key target for efficient realization, especially in an explicitly correlated form.[62, 43, 47, 147] For example, recently CCSD(T)-F12 has been realized in a reduced-scaling form by several groups, including ours.[148, 149] To test the reduced-scaling CCSD(T), we need to obtain the full-scaling (canonical) CCSD(T) for systems that are sufficiently large for the near-sightedness of electron correlation effects to become apparent. The motivation for this work is to provide a reference scalable canonical CCSD(T)-F12 implementation capable of treating molecules as large as presently feasible. We recently reported a massively-scalable implementation of explicitly correlated CCSD,[150] so here we focus primarily on the (T) correction.

Some distributed-memory parallel algorithms have been devised for the (T) correction of the CCSD(T) method (a recent review of the existing distributed-memory parallel implementations of CCSD can be found in Ref. 150). Rendell *et al.*reported the first implementation of a CCSD(T)-like method on distributed parallel computers.[151] Only the so-called fourth-order terms (in the Møller-Plesset sense) of the (T) energy correction were included. Nevertheless, the parallelization strategy and implementation are directly applicable to the full CCSD(T) correction. The (T) energy correction can be viewed as a set of binary reductions of order-6 tensors (with three occupied indices of rank O and three unoccupied/virtual indices of rank V, with typically $O << V$) whose elements/blocks are evaluated on the fly. Rendell *et al.* used what they referred to as the *abcijk* strategy, in which the reduction over three unoccupied indices (*abc*) uses task parallelism, as well as the *aijkbc* strategy, in which task parallelism is over one unoccupied and one occupied index (*ai*). The *abcijk* algorithm was nearly perfectly scalable if all integrals are replicated in each node's memory (the largest set of integrals has OV^3 elements). For all but the smallest computations, it was necessary to store integrals on a parallel file system and read them in batches. Due to the $\mathcal{O}\left(OV^4\right)$ I/O complexity the disk-based *abcijk* algorithm was not scalable, whereas the *aijkbc* algorithm had excellent strong scaling thanks to the I/O requirement substantially reduced to $\mathcal{O}\left(O^2V^3\right)$. Note that the *aijkbc* algorithm has an operation count greater than the optimal by a factor of 4. The local (per-node) memory requirements of the *aijkbc* algorithm are $\mathcal{O}\left(OV^2\right)$, i.e., rather modest. The first complete distributed-memory implementation of closed-shell CCSD(T) was developed in NWChem by Kobayashi and Rendell,[89] which followed the *aijkbc* algorithm of Rendell *et al.*[151] Unlike the disk-based implementation,[151] the implementation in NWChem stored integrals in memory using Global Arrays (GA) abstraction for one-sided data access. Perfect scaling of the (T) energy was demonstrated for glycine ($C_2O_2NH_5$; $O = 20$, $V = 80$) in Cray T3E from 8 to 64 processors. The perturbative triples code was later improved by reducing communication and memory consumption.[152] The improved code was applied to study the binding energy of liquid water cluster. For $(H_2O)_{18}$ in mTZ basis (918 basis functions), the total CCSD(T) evaluation (including CCSD) exhibited a speed-up of ~ 2.2 from 30,000 to 90,000 processors of the Cray XT5 supercomputer. For $(H_2O)_{20}$ in mTZ basis (1020 basis functions), this new code used 100TB memory and 96,000 cores of ORNL's Jaguar computer to evaluate the (T) energy, which provided 487 TFLOPS performance (55% of theoretical peak) for 2 hours.[152]

Another implementation of CCSD(T) in NWChem is based on the Tensor Contraction Engine (TCE),[91] which implements each tensor operation as a statically-scheduled sequence of tile tasks with blocking one-sided data movement provided by the GA toolkit. This implementation is significantly different from most other applications in that the reduction over (block of) all six indices, rather than 2 or 3, are fully distributed among nodes. The trade-off of such fine-grained parallelization strategy is greater communication requirements. The efficiency of TCE approach has been demonstrated with CR-EOMCCSD(T) calculation on FBP-f-coronene/AVTZ (780 basis functions), where the non-iterative triples correction shown 84% parallel efficiency from 60,000 cores to 210,000 cores.[94] Later, the TCE based implementation was ported to GPU hardware[153] as well as Intel Xeon Phi co-processor.[154]

The original parallel CCSD(T) implementation in GAMESS uses both distributed and shared memory was described by Bentz and Ryan *et al.*[155,101] The data distribution is handled by the Distributed Data Interface(DDI) library. This (T) algorithm corresponds to the $ijkabc$ strategy, originally described in Ref. 156, in which the reduction over $i \geq j \geq k$ are distributed among nodes. The integrals were fetched through the one-sided GET operation in DDI. A parallel efficiency of 72% was obtained at 24 processors on the T-shaped benzophenolbenzene dimer ($O = 33, V = 313$) using an IBM SMP cluster.[155] Later, another closed-shell CCSD(T) implementation was done by Asadchev and Gordon[103] within the libcchem library embedded in GAMESS. This (T) algorithm used the *abcijk* strategy in which the reduction over *abc* is distributed among nodes (this was first used in a parallel (T) program by Janowski and Pulay;[97] see below) to minimize the per-node memory requirements. Parallel speedup of 4 was demonstrated for $C_8H_{10}N_4O_2$/cc-pVTZ from 4 nodes to 16 nodes on a commodity cluster; similarly, a speedup of 3.7 was obtained for $SiH_4B_2H_6$/aug-cc-pVQZ from 256 cores to 1024 cores on a Cray XE6 system.

The massively parallel ACES III[106] is a ground-up redesign of the ACES II coupled-cluster suite of programs from Quantum Theory Project. The CCSD(T) energy and gradient are implemented for both restricted and unrestricted HF reference using the super instruction assembly language (SIAL) domain-specific language. The (T) energy implementation in ACES III appears to be similar to the NWChem TCE implementation, by entirely distributing the tasks down to the tile (block) level. The strong scaling of UHF CCSD(T) code was demonstrated through the Ar_6/aug-cc-pVTZ molecule. The total execution time was reduced from 784 minutes to 131 minutes by increasing the number of processors from 32 to 256 on a Sun cluster, which exhibited approximately 75% parallel efficiency.

The (T) energy implementation in Parallel Quantum Solution (PQS) package[97] is heavily optimized to use clusters of nodes with local disks by building on the Array Files middleware. This algorithm uses the *abcijk* strategy in which reduction over *abc* is distributed among nodes. The advantage of this strategy is that the per-node memory requirement is only $\mathcal{O}\left(O^3\right)$, which does not depend on the number of virtual orbitals and is suited for large basis set calculations. A (T) test case on aspirin/6-311G** demonstrated that a factor of 27.8 times speedup was obtained by increasing the number of processors from 1 to 32. They also provided timing for (T) calculations on benzene dimer/aug-cc-pVQZ, which contains 1512 basis functions. By taking advantage of the point group symmetry of benzene dimer (C_{2h} symmetry), the (T) calculation took 57 hours using 32 nodes.

The density-fitting or resolution-of-the-identity approximation approximates a four-index two-electron integral tensor in terms of two- and three-index tensors, e.g. a two-electron Coulomb integral is approximated as:[157–160]

$$(\rho\sigma|\mu\nu) \overset{\mathrm{DF}}{\approx} B^{\Lambda}_{\rho\sigma} B^{\Lambda}_{\mu\nu}, \tag{3.1}$$

where

$$B^{\Lambda}_{\rho\sigma} =(\rho\sigma|K)(\mathbf{V}^{-1/2})_{K\Lambda}, \tag{3.2}$$

$$V_{K\Lambda} \equiv(K|\Lambda), \tag{3.3}$$

and standard Mulliken (chemists) bra-ket notation for 2-electron Coulomb integrals has used.[1] DF-based MP2 method[161–163] has reduced computational cost relative to the conventional MP2 approaches that utilize the 4-center AO integrals by reducing the cost of the integral evaluation (which is particularly significant for large basis sets with high-angular

[1] Throughout the document lowercase Greek letters (μ, ν, ρ, σ) will denote the Atomic Orbital (AO) functions used to expand the electronic states, uppercase Greek letters (K, Λ) will denote the AO functions used for the density fitting, and lowercase Roman letters will denote occupied (i, j, k, l) and unoccupied (a, b, c, d) Hartree-Fock states.

momentum AOs) and by reducing the prefactor of the $\mathcal{O}\left(N^5\right)$ step, from ON^4 to O^2V^2X. In the coupled-cluster methods, the density fitting approximation doesn't reduce the computational cost that much. However, it is a standard technique to compute four-index molecular integrals with density fitting and avoid storing them in a shared-memory program to reduce the I/O and disk storage requirements. This approach has been shown to improve the efficiency of the CCSD program by a significant amount as well as to obtain high accuracy at the same time.[164, 165, 129] Massively parallel DF-based MP2 implementation has been reported before,[166–169] however, a massively parallel DF-based CCSD(T) implementation has not been reported yet. All existing distributed-memory implementations of the (T) energy utilize conventional (4-center) integrals. The primary reason to use the density fitting approximation is that it is essential to all reduced-scaling CC methods based on pair-natural orbital (PNOs);[170–172] DF dramatically reduces the computational expense of computing the Hamiltonian in the PNO framework by decoupling the (chemists') bra and ket. Scalable canonical DF-based implementations are therefore highly desired for testing the emerging reduced-scaling PNO-CC approaches. In this paper, we describe an implementation that uses density fitting (DF) approximation for all two-electron integrals. We implement a new parallel DF-based closed-shell CCSD(T) method inside Massively Parallel Quantum Chemistry package version 4 (MPQC4).[113] This implementation utilized TiledArray framework to distribute all data in memory and perform tensor contractions. The (T) correction is parallelized by manual partitioning of data and computation. The detail of the feature of TiledArray and original CCSD implementation has been discussed in our previous work.[150]

In Section 3.2, the implementation detail of the DF-CCSD(T) will be discussed. Section 3.3 explains the details of computing resources used in this work as well as the information about compiling and running MPQC4. Section 3.4 demonstrates the performance of the DF-CCSD(T) on shared-memory and distributed-memory computers, and comparison of performance to NWChem.

3.2 Methods

Before turning our attention to the DF-based (T) energy algorithm, we will discuss the prerequisite improvements of the massively parallel DF-CCSD implementation in Ref. 150.

3.2.1 CCSD

The original massively parallel implementation[150] of CCSD in MPQC4 incorporated two formulations of CCSD that employed density fitting approximation for the four-center integrals. The *standard* formulation of DF-CCSD used MO-based 3-index integrals to reconstruct the MO-basis 4-index integrals as needed. Although DF does not change the arithmetic complexity of the method, modest savings are possible as one of several $\mathcal{O}(N^6)$ terms in the CCSD doubles equation can be reduced to $\mathcal{O}(N^5)$.[150] DF also does not change the $\mathcal{O}(N^4)$ space complexity of CCSD due to the need to store MO-basis 4-index integrals; the overall storage is typically dominated by the MO integrals and intermediates with four unoccupied indices. Since in the standard DF-CCSD approach all 4-index MO integrals are stored persistently in memory, storage can become the limiting factor on modest clusters.[150] However, we were able to reduce the storage requirements by avoiding all integrals with with 3 and 4 unoccupied indices by evaluating some $\mathcal{O}\left(N^6\right)$ terms in the CCSD doubles residual in AO basis using the standard integral-direct approach. Such combination of the DF approximation with the integral-direct AO formalism was referred to as the *hybrid* DF-AO CCSD algorithm.

Although the hybrid DF-AO-CCSD absolute correlation energies differed from the standard DF-CCSD counterparts, the differences in relative energies were negligible and the hybrid approach was the default choice of CCSD when a DF basis was provided by the user. Nevertheless, it is not possible to compare directly the absolute DF-AO-CCSD energies with the reduced-scaling DLPNO-CCSD energies since the latter uses density fitting throughout the coupled-cluster procedure. Thus we started our work by re-formulating the standard massively parallel DF-CCSD approach to reduce the memory storage without resorting to the use of AO 4-index integrals. The idea is simple: to avoid storing integrals and intermediates with more than two unoccupied indices we evaluate one batches 4-center integrals with lazy evaluation of the largest 4-index MO integrals via DF. E.g., consider the typically most expensive term in the doubles amplitude equation:

$$\begin{aligned} W^{ab}_{cd}\tau^{cd}_{ij} &= (G^{ab}_{cd} - G^{ak}_{cd}t^b_k - G^{kb}_{cd}t^a_k)\tau^{cd}_{ij} \qquad (3.4) \\ &\approx P^{ab}_{ij}\left\{B^{\Lambda}_{ac}(\frac{1}{2}B^{\Lambda}_{bd} - B^{\Lambda}_{kd}t^k_b)\tau^{cd}_{ij}\right\} \qquad (3.5) \\ &= P^{ab}_{ij}\left\{(B^{\Lambda}_{ac}\bar{B}^{\Lambda}_{bd})\tau^{cd}_{ij}\right\}, \qquad (3.6) \end{aligned}$$

where G^{pq}_{rs} are 4-index two-electron integrals and τ is defined as:

$$\tau^{ab}_{ij} = t^{ab}_{ij} + t^a_i t^b_j. \tag{3.7}$$

Unlike conventional shared-memory program, not storing intermediates but computing them on the fly through density fitting will actually increase the amount of computation time for each CCSD iteration, especially when using large density fitting basis set. The direct evaluation of W was done by constructing it as `DistArray` of `DirectTile` in TiledArray. Please refer our previous publications for detail descriptions on template tensor classes in TiledArray framework.[118,150] The `DirectTile` does not explicitly store the data, however, it can be converted to `Tile` through a customizable builder. In the case of Eq. 3.4, whenever a `DirectTile` in tensor W is needed in SUMMA iteration to contract with τ, it will be converted to `Tile` by calling builder to evaluate this `Tile` of integral using density fitting. This requires the builder to have access to data of three-index tensor B and $\bar{B}$, and to compute particular block of W by looping over density fitting auxiliary basis Λ. An trivial solution to this is to replicate tensor B and $\bar{B}$ on each node, so that all processors have access to it. However, this increases the amount of local memory requirement dramatically when the number of unoccupied orbitals become larger and larger. For example, for a system with 1000 unoccupied orbitals and 3000 auxiliary basis functions, tensor B^{Λ}_{ac} will cost 24 GB to store without utilizing permutation symmetry. Hence, distributing the tensor B and $\bar{B}$ in memory among all nodes is the only way to make it scalable for large systems with more than one thousand basis functions. The builder of the `Tile` object will have access to the distributed tensor B and $\bar{B}$, and will fetch all the blocks needed through communication to form the corresponding `Tile` object. This introduces more cost (computation and communication), but it will reduce the amount of memory requirement significantly. The comparison of the performance of the new and old DF-CCSD code is demonstrated in Section 3.4.

3.2.2 (T)

We have implemented a distributed-memory parallel the (T) correction for the coupled-cluster singles and doubles for closed-shell systems. In the spin-adapted formulation, the energy expression for the (T) correction is given as follow:

$$E_{(T)} = \sum_{a \geq b \geq c} (2-\delta_{ab}-\delta_{bc}) \sum_{ijk} (W_{ijk}^{abc}+V_{ijk}^{abc})(4W_{ijk}^{abc}+W_{jki}^{abc}+W_{kij}^{abc}-2W_{jik}^{abc}-2W_{ikj}^{abc}-2W_{kji}^{abc})/D_{ijk}^{abc}, \tag{3.8}$$

where W, V and D are defined as:

$$W_{ijk}^{abc} = P_{ijk}^{abc}(G_{if}^{ab}t_{kj}^{cf} - G_{ij}^{al}t_{lk}^{bc}), \tag{3.9}$$

$$V_{ijk}^{abc} = t_i^a G_{jk}^{bc} + t_j^b G_{ik}^{ac} + t_k^c G_{ij}^{ab}, \tag{3.10}$$

$$D_{ijk}^{abc} = f_a^a + f_b^b + f_c^c - f_i^i - f_j^j - f_k^k. \tag{3.11}$$

In the equation above, t_i^a and t_{ij}^{ab} are the converged CCSD amplitudes and f is the Fock matrix. The cost to form W is the most expensive part, which is $O^4V^3 + O^3V^4$ and scales $\mathcal{O}(N^7)$, where O and V stand for the number of active occupied and unoccupied orbitals. In the (T) energy Eq 3.9 and 3.10 , DF does not provide computation reduction through factorization. Also, the $\mathcal{O}(N^7)$ contraction step becomes the computation bottleneck and storing the largest integral G^{ab}_{if} is usually not the issue. Hence, integral G^{ab}_{if}, which is not computed in DF-CCSD, will be computed with DF and distributed in memory. This DF-based (T) implementation leads to the same cost and parallel algorithm as the conventional (T) approach.

In general, the algorithms to parallel (T) is to distribute the loops over (ijk) index or (abc) index among different MPI processes. As discussed by Rendel *et al.*[156] as well as Janowski and Pulay,[97] the former approach would require more local memory ($\mathcal{O}\left(V^3\right)$) than the latter ($\mathcal{O}\left(O^3\right)$), such that the latter method is suited for large systems. In this work, we loop over blocks of unoccupied orbital $b_a \geq b_b \geq b_c$ and distribute the loops to different MPI processes using the Round-robin algorithm.

Listing 3.1: (T) algorithm for the W term

```
iter = 0;
// loop over blocks in a
for block_a in a {
// loop over blocks in b <= a
for block_b <= block_a {
// loop over blocks in c <= b
for block_c <= block_b {
iter++;
if (iter % MPI_SIZE != MPI_RANK) continue;
// fetch all data needed for block_a, block_b and block_c
...
// compute W for block_a , block_b and block_c
...
// accumulate contribution to E(T)
...
}
}
}
```

All the data required to perform (T) calculation will be distributed in memory, which is managed by TiledArray. The MPI process assigned with particular indices b_a, b_b and b_c will ask for the corresponding blocks of data from TiledArray. The communication happens when those data, which are distributed to different MPI processes, are sent to current MPI process that asks for them. TiledArray uses task-based parallel runtime from MADNESS that allows overlap between communication and computation, which provides excellent parallel scaling in this process. This means that the working MPI process does not need to wait for all the data to arrive to start the computation. Within each MPI process, the computation of $W_{ijk}^{b_a b_b b_c}$ looks like below:

$$W_{ijk}^{b_a b_b b_c} = P_{ijk}^{abc}(G_{if}^{b_a b_b} t_{kj}^{b_c f} - G_{ij}^{b_a l} t_{lk}^{b_b b_c}) \tag{3.12}$$

This leads to the blocked algorithm similar to Ref. 103, which has local storage requirement of $O^3 b_v^3$ for $W_{ijk}^{b_a b_b b_c}$ and $V_{ijk}^{b_a b_b b_c}$, where b_v is the block size in unoccupied space. Instead of using a threaded matrix multiplication to parallelize the computation of the $W_{ijk}^{b_a b_b b_c}$ term, multiple

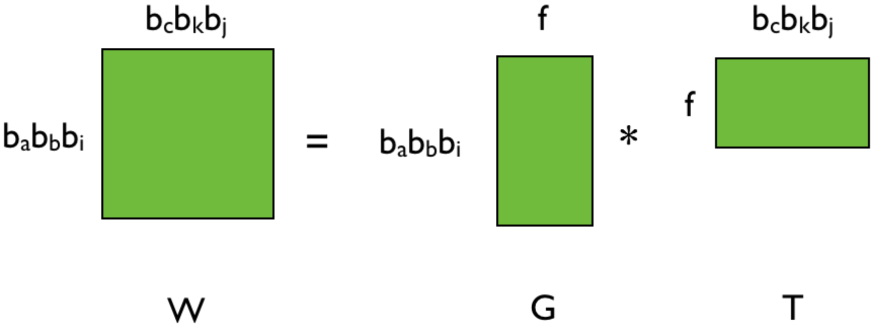

Figure 3.1: Contribution to first term in $W_{ijk}^{b_a b_b b_c}$ as a matrix product

sequential matrix multiplications was used by blocking over the i, j and k indices in Eq. 3.12. This gives a total number of $n_{b_o}^3$ matrix multiplications, where n_{b_o} stands for the number of blocks in the occupied space. The formation of the $W_{ijk}^{b_a b_b b_c}$ term within each MPI process is paralleled by submitting these serial matrix multiplications to different threads, which is also handled by TiledArray. Figure 3.1 illustrates how each matrix multiplication looks like in the first term of Eq. 3.12. Notice that the dimension f is not blocked. This is because that the size of $W_{b_i b_j b_k}^{b_a b_b b_c}$ matrix ($b_a b_b b_i$ by $b_c b_k b_j$) can be much larger than the contraction dimension f (V), and matrix closer to square form has higher efficiency in performing matrix multiplication. This introduces the problem of different blocking for the same space, which can be solved through the approach described later in Eq. 3.13 by re-blocking. The algorithm to evaluate the second term in Eq. 3.12 is the same as the algorithm to evaluate the first term.

Since six-dimension tensors are encountered in the (T) part, the tiling of six-dimension tensors needs to be much smaller than the CCSD. For example, four-dimension tensor with tile size 30 will cost around 6.5 MB to store, but 5.8 GB to store in six-dimension tensor with the same tile size. Since the local storage requirement $(O^3 b_v^3)$ depends on the tile size of the unoccupied space, large tile size in unoccupied space will raise the local storage requirement. Moreover, tensors that have large tile size will become less efficient to communicate from one node to another. Therefore, the tile size of the integrals and CCSD amplitudes will be updated to fit the requirement of tile sizes for the (T) correction. In MPQC4, updating the tile size is done by multiply the original array with identity matrix with different tile sizes, as shown in Eq. 3.13:

$$t^{a_{\bar{b}_v} b_{\bar{b}_v}}_{i_{\bar{b}_o} j_{\bar{b}_o}} = t^{a_{b_v} b_{b_v}}_{i_{b_o} j_{b_o}} I^{i_{b_o}}_{i_{\bar{b}_o}} I^{j_{b_o}}_{j_{\bar{b}_o}} I^{a_{\bar{b}_v}}_{a_{b_v}} I^{b_{\bar{b}_v}}_{b_{b_v}}, \tag{3.13}$$

where b_o and b_v are original tiling for occupied and unoccupied space while $\bar{b}_o$ and $\bar{b}_v$ are new tilings. All the integrals and CCSD amplitudes were transformed in this way before starting the (T) step. This introduces extra computational cost, but it is only a small fraction of the total computation time since it only scales $\mathcal{O}(N^4)$.

3.3 Computational Details

The DF-CCSD(T) calculations were performed using a developmental version of Massively Parallel Quantum Chemistry package version 4 (MPQC4). All calculations were performed on the “Blueridge” cluster at Virginia Tech and the “Theta” supercomputer at the Argonne National Laboratory. Blueridge has two Intel Xeon E5-2670 CPUs (332 GFLOPS) and 64 GB of RAM on each node. On Blueridge, both MPQC4 and TiledArray were compiled using GCC 5.3.0 with Intel MPI 5.0 and serial MKL library 11.2.3. All MPQC4 calculations performed on Blueridge launched 1 MPI process per node and 16 threads per MPI process using tile size 20 for the CCSD and 8 for the (T). The compute nodes on Theta are equipped with Intel Xeon Phi Knights Landing 7230 Processor (2660 GFLOPS), which contains 64 cores, 192 GB DDR4 memory and 16 GB MCDRAM memory. On Theta, MPQC4 and TiledArray were compiled with GCC6.3.0, using Cray MPICH/7.6.0 and serial MKL 2017.0 library. Shared-memory calculations on Theta used 64 threads on one node, and distributed computations on Theta launched 4 MPI processes per node and 16 threads per MPI process. All MPQC4 calculations on Theta used the quad-cache mode and used tile size 30 for the CCSD and 10 for the (T) calculations. Frozen core approximation was used throughout all examples. In the GC-dDMP-B calculation, the atomic integrals were computed with the precision of 10^{-12} Hartree, which is the same as used by NWChem calculation.[90]

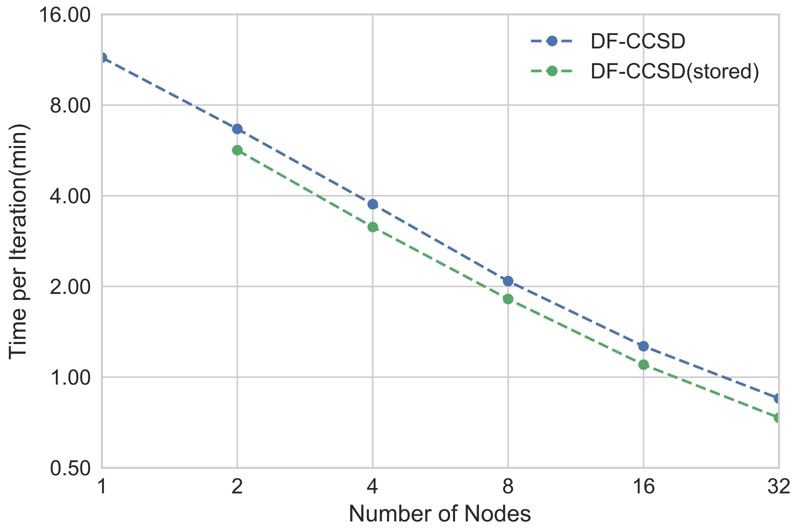

Figure 3.2: Strong scaling performance of the DF-CCSD implementation of uracil trimer in the cc-pVDZ basis on Blueridge.

3.4 Results & Discussion

3.4.1 CCSD

Figure 3.2 compares the performance of two different versions of DF-CCSD implementation on uracil trimer/cc-pVDZ. The DF-CCSD stands for the new implementation that computes the W_{cd}^{ab} intermediate on the fly using DF, while the DF-CCSD(stored) stands for the old approach that stores and distributes all intermediates in memory. Both approaches have the same strong scaling, reaching $\sim 40\%$ parallel efficiency on 32 nodes. The DF-CCSD code is slower than the other, because the W_{cd}^{ab}intermediate is computed on the fly at each iteration. However, this approach requires less memory, and it can run on one node. Table 3.1 provides the DF-CCSD timings for different systems. We noticed that the DF-CCSD performance is slower than our previous implementation which uses the *hybrid* DF-AO algorithm. It is because evaluating the W_{cd}^{ab} term on the fly becomes expensive with large prefactor when using large auxiliary basis sets. However, when the system size becomes larger, the $\mathcal{O}(N^5)$ step should become negligible to the $\mathcal{O}(N^6)$ cost in the CCSD calculation. The uracil trimer/cc-pVDZ example was also used to test the multi-thread performance on the Intel Xeon Phi Knights Landing processor using one node of the Theta. There are a total number of 64 cores on Intel Xeon Phi Knights Landing processor, so a good threading performance is essential for this chip. Figure 3.3 shows that the DF-CCSD code has excellent performance with 27.5 and 41.0 times speedup on 32 and 64 threads compared to time at 1 thread. Moreover, the time on one node of Theta (150s) is about 4.5 times faster than the time obtained on one node of Blueridge (690s). Figure 3.4 compares the parallel performance of the DF-CCSD implementation on Blueridge and Theta. The performance on Theta shows only a factor of 2 faster than the performance on Blueridge, but the parallel speedup on 16 and 32 nodes are similar on both platforms.

Table 3.1: The DF-CCSD performance for different systems on Blueridge

Molecule	Basis Set	Atoms	Act Occ [a]	Basis/DF Basis	Nodes	Time/Iter [b]
Uracil dimer	aug-cc-pVTZ	24	42	920/2064	32	56
$(H_2O)_{20}$	cc-pVTZ	60	80	1160/2820	32	126
Pentacene dimer	aug-cc-pVDZ	72	102	1264/3812	64	191

[a] Active occupied orbitals [b] All times are in minutes

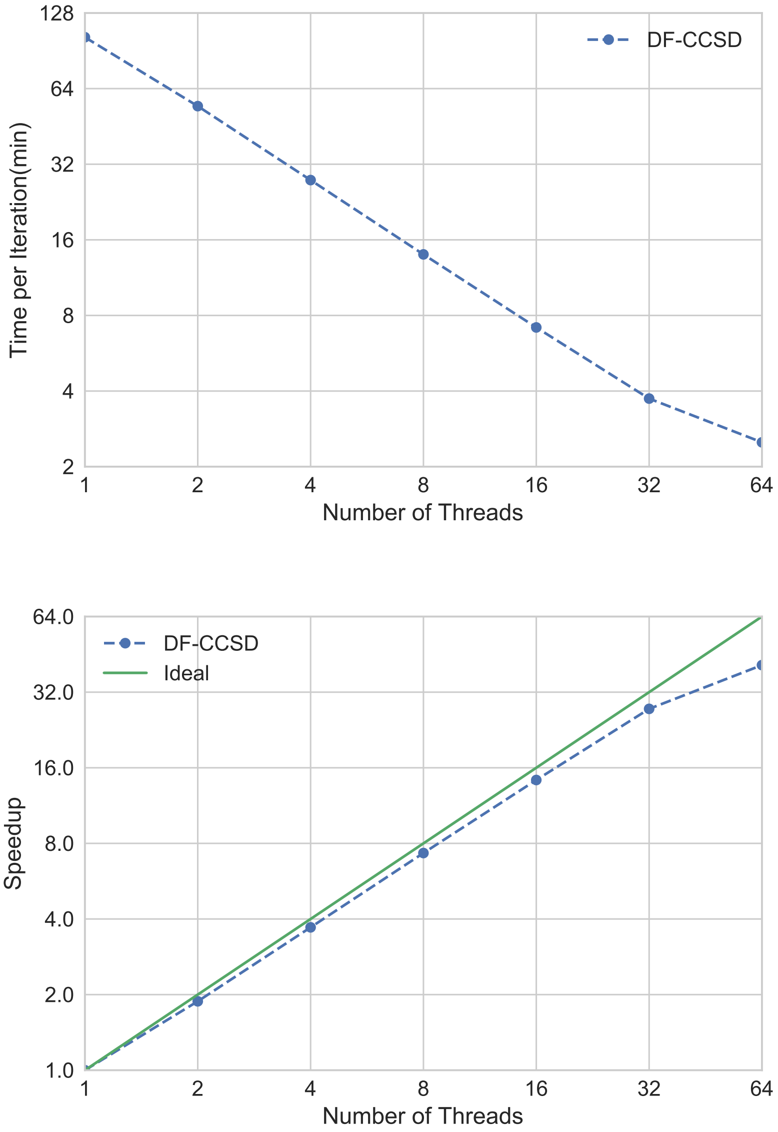

Figure 3.3: The multi-thread performance of the DF-CCSD implementation of uracil trimer in the cc-pVDZ basis on Theta.

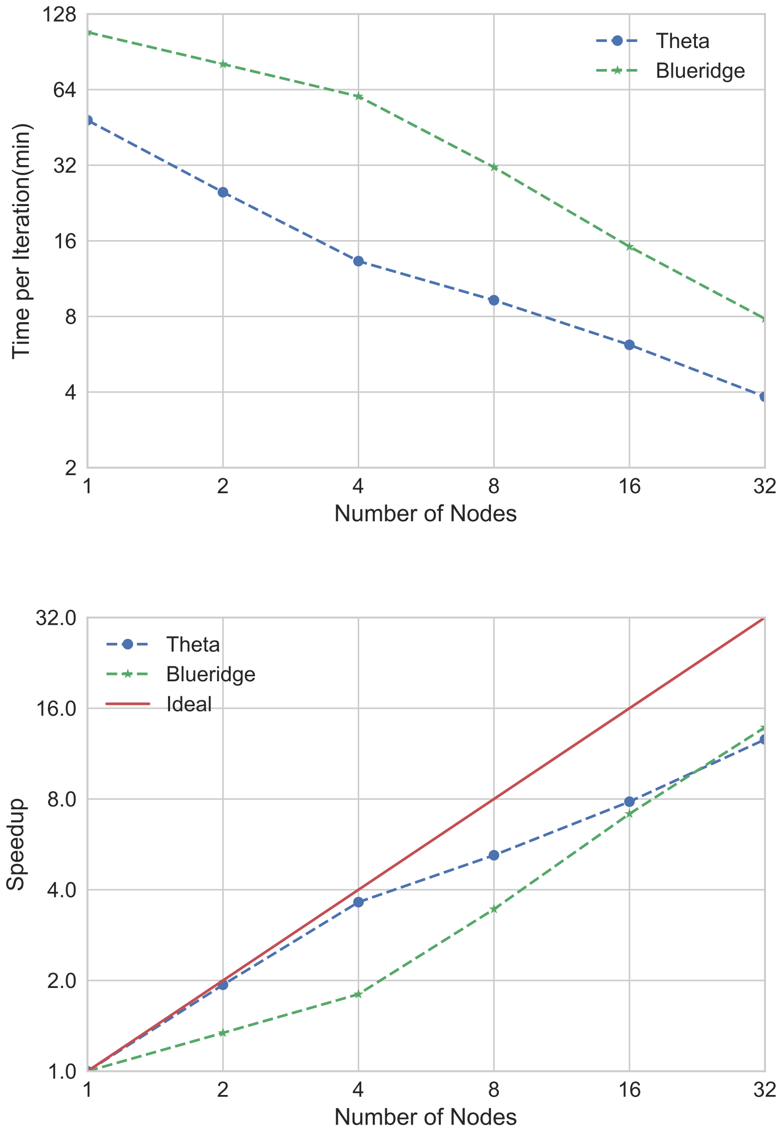

Figure 3.4: Strong scaling performance of the DF-CCSD implementation of uracil trimer in the aug-cc-pVDZ basis on Blueridge and Theta

3.4.2 (T)

The multi-thread performance of the (T) implementation on a single node was examined using the Blueridge cluster on a $(H_2O)_{10}$/cc-pVDZ cluster, which has 40 occupied and 190 unoccupied orbitals. In this case, there are 5 blocks in occupied space, which provides a total number of 250 matrix multiplications to compute the W term inside the loop in List 3.1. It guarantees good intra-node scaling with respect to the number of threads. Figure 3.5 demonstrates that the (T) execution time is reduced from 659.8 minutes to 41.6 minutes using 16 threads compared to 1 thread, which is almost linear scaling with a factor of 15.9 times speedup. The strong scaling performance of the (T) implementation was tested on both Blueridge and Theta. Figure 3.6 shows the performance for a $(H_{20})_{14}$ cluster with respect to the number of nodes. The cc-pVDZ basis set was used in this benchmark, and this system has 56 active occupied orbitals and 266 unoccupied orbitals. Excellent strong scaling on Blueridge is demonstrated in Figure 3.6: a factor of 14.5x and 25.7x speedup are observed on 16 nodes and 32 nodes comparing to 1 node, which is $\sim 90\%$ and $\sim 80\%$ parallel efficiency, respectively. On Theta, the calculation starts from 2 nodes, and we assumed the parallel efficiency on two nodes is 100% compared to 1 node. A factor of 11.2x and 18.0x speedup is maintained on 16 nodes and 32 nodes comparing to 1 node, respectively, which is slightly lower that Blueridge. This might be caused by the fact that 4 MPI processes per node are launched on Theta, which introduced more intra-node communication compared to using 1 MPI process per node on Blueridge. Moreover, the Theta performance is around 2x faster than Blueridge, which is in accordance with the result in Figure 3.4. It indicates

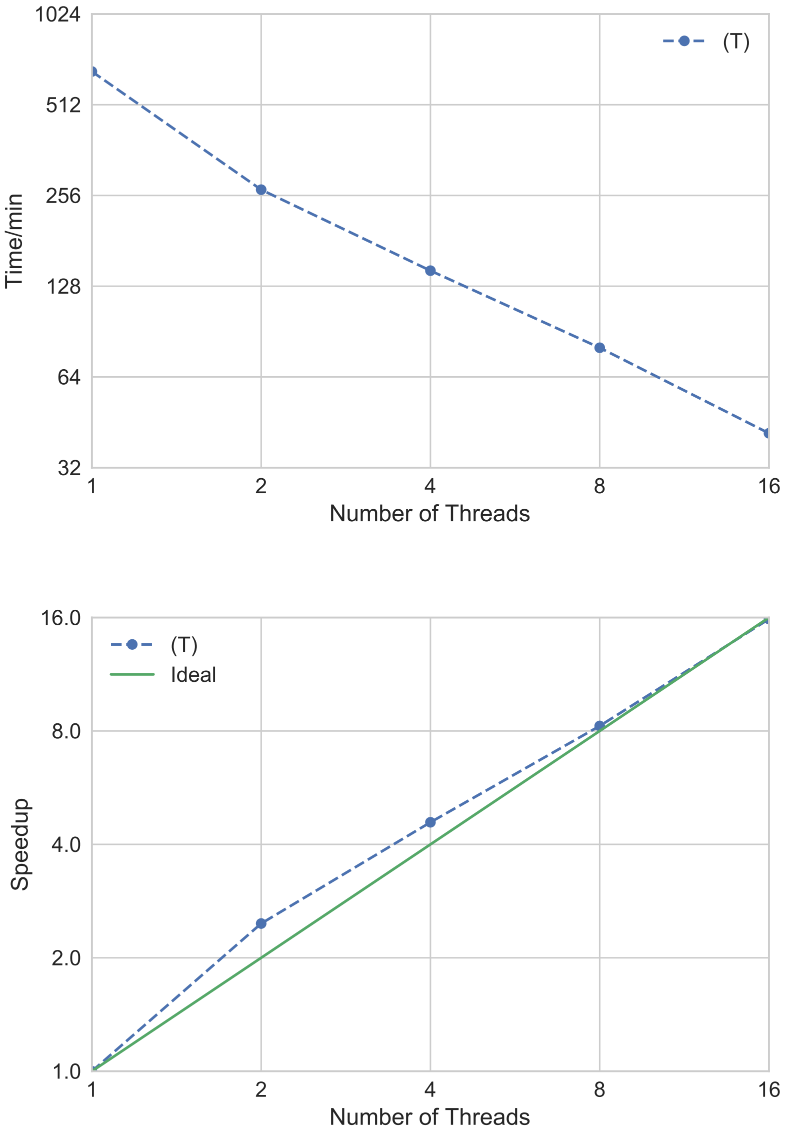

Figure 3.5: Threading performance of the (T) implementation of $(H_2O)_{10}$ cluster in cc-pVDZ basis on Blueridge.

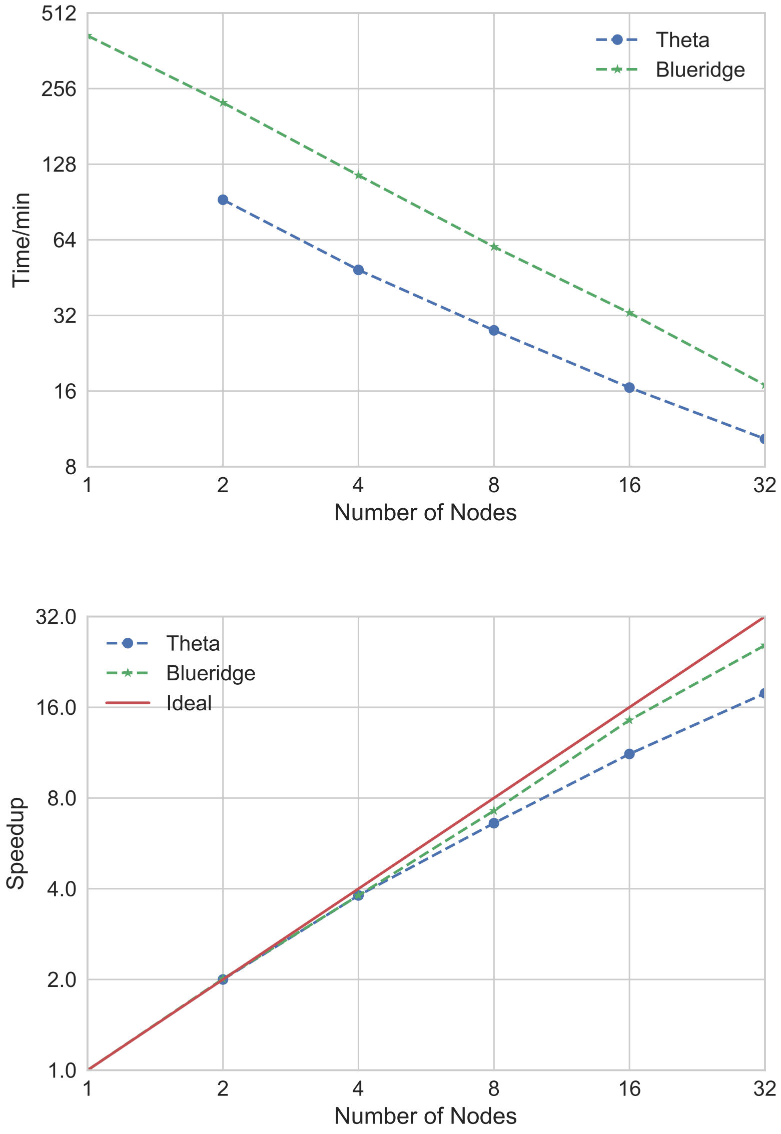

Figure 3.6: Strong scaling performance of the (T) implementation of a $(H_2O)_{14}$ cluster in cc-pVDZ basis on Blueridge and Theta.

that the performance of GEMM is low for small matrices on the Knights Landing. In the case of the (T) calculation, the contraction dimensions in Eq. 3.12 are small: $O = 56$ and $V = 266$. We expect that the performance will improve for larger systems on Theta. Table 3.2 shows the (T) calculation time for different molecules from as few as 12 atoms to as many as 72 atoms on Blueridge. Among these calculations, the uracil dimer/cc-pVTZ-F12 contains the largest number of basis functions (992), and the calculation of the (T) part took less than 10 hours with 24 nodes (384 cores). The most expensive calculation in Table 3.2 is pentacene dimer/cc-pVDZ, which has a total number of 72 atoms and 756 basis functions, and the (T) correction took $\sim$ 25 hours using 32 nodes (512 cores). We have compared the CCSD(T) performance of MPQC4 with NWChem performance on GC-dDMP-B/6-311++G** (103 occupied orbitals and 1042 basis functions) reported previously,[90] which is the largest conventional CCSD(T) calculation reported as far as we know. GC-dDMP-B stands for guanine-cytosine deoxydinucleotide monophosphate in B-conformation neutralized by a sodium cation, which has a total number of 63 atoms. The NWChem calculation used 20,000 nodes of the Blue Waters supercomputer, which has two AMD 6276 Interlagos

Table 3.2: (T) performance data for different systems on the Blueridge cluster

Molecule	Basis Set	Atoms	Act Occ [a]	Basis	Nodes	(T) Time [b]
Uracil	cc-pVDZ-F12	12	21	276	1	16
	cc-pVTZ-F12			496	1	147
Uracil dimer	cc-pVDZ-F12	24	42	552	8	152
	cc-pVTZ-F12			992	24	582
Pentacene	cc-pVDZ	36	51	378	1	507
Pentacene dimer		72	102	756	32	1485

[a] Active occupied orbitals [b] All times are in minutes

Processors (313.6 GFLOPS) and 64 GB memory on each node, and it is very similar to the Blueridge compute node we used in terms of total memory and peak performance. We reproduced the CCSD(T) calculation on GC-dDMP-B/6-311++G** with the same geometry using 64 nodes of Blueridge and the timings are reported in Table 3.3. Due to the tremendous computational cost of this benchmark, we had to ask for extension of total runtime on Blueridge to enable this calculation. Hence, the CCSD part was limited to run four iterations to save computing time, and the final result is not converged, which is not a problem in this case since only performance will be compared. It is remarkable that the MPQC4 CCSD time per iteration on 64 nodes is faster than the NWChem time on 1100 nodes and only 3.3 times slower than the NWChem time on 20,000 nodes. It indicates that the performance of the NWChem CCSD part is not efficient, which can also be proved from NWChem's data: the CCSD computation time is longer than the (T) time. The total time of MPQC4 spent in the CCSD part is estimated by multiplying the average CCSD time per iteration (43 minutes) with a factor of 18, which is the total number of iteration to converge to 10^{-6} according to NWChem's data, and adding the time for initialization ($20 + 18 * 43 = 792$). The total time spent in the (T) part of MPQC4 is 47.4 hours and is 33.8 times slower than NWChem, which is still great considering that NWChem is using over 300 times more nodes. At last,

Table 3.3: Comparison of GC-dDMP-B/6-311++G** CCSD(T) performance

Software	Nodes	CCSD Iteration Time	CCSD Time	(T) Time	CCSD(T) Time
NWChem	1,100	72 min	-	-	-
	20,000	13 min	4.0 h	1.4 h	5.4 h
MPQC4	64	43 min	13.2 h	47.4 h	60.6 h

the total CCSD(T) time of MPQC4 is only 11.2 times slower than NWChem by using much less computational resources that are readily available to most researchers. Through this test case, we have proved that the CCSD(T) implementation in MPQC4 is efficient and scalable.

3.5 Conclusions

In this work, we presented an efficient and scalable DF-based CCSD(T) implementation in MPQC4, which has excellent strong scaling on both shared-memory and distributed memory computers with Intel Xeon and Xeon Phi processors. We have shown the ability of this program to perform large CCSD(T) calculations with a limited amount of computing resources by comparing to NWChem performance on the GC-dDMP-B/6-311++G** system. The TiledArray framework was proved to be an efficient tool to implement massively parallel electronic structure methods. We believe that MPQC4 is a robust package to develop new electronic structure methods that can handle more massive chemical problems than conventional shared-memory quantum chemistry software.

Chapter 4

Exploration of Reduced Scaling Formulation of Equation of Motion Coupled-Cluster Singles and Doubles Based on State-Averaged Pair Natural Orbitals

4.1 Introduction

Accurate description of electronic spectra of medium (< 100 atoms) and large (> 100 atoms) molecular systems has always been a challenge for quantum chemistry. The time-dependent density functional theory (TDDFT) is the most popular method for the analysis of excited states due to its computational efficiency, capable of treatment of systems with hundreds and thousands of atoms. Although TDDFT provides medium accuracy for one-electron excitations, the accuracy of TDDFT can be limited for certain types of excited states (e.g. Rydberg or charge transfer)[173] and in general its accuracy depends strongly on the density functional.[174] In contrast to TDDFT, multiconfiguration/multireference (MR) wave function models, such as MR perturbation theory methods (e.g. complete active space perturbation theory (CASPT2)[14] and n-electron valence states perturbation theory (NEVPT2)[19]) and MR configuration interaction[175,176] can recover both static and dynamic electron correlation, can treat multiple electronic states on equal footing, and attain high accuracy, albeit for rather small systems.[4] Among the challenges of the MR approaches is the need to select the active space, and the exponential growth of complexity with the size of active space. Although the latter can be avoided for certain types of systems by numerical approximations such as density matrix renormalization group[177,178] and other tensor network approaches, the MR methods are generally difficult to use for nonspecialists. Accurate treatment of dynamical electron correlation in the context of MR methodologies is an ongoing direction of research.

In this work we focus on the treatment of excited states by the coupled-cluster method. The highly robust coupled-cluster hierarchy provides unparalleled accuracy for the ground states by systematically including two-, three- and higher-body correlation effects from a single determinant reference. The CC ansatz can be extended to excited states through the use of the linear-response (LR) theory,[179] the symmetry-adapted cluster configuration interaction (SAC-CI) method,[180,181] or the equation of motion coupled-cluster (EOM-CC) method.[182,183] However, the high-order scaling of the coupled-cluster methods limits its application to small molecules. Even with truncation to singles and doubles excitations, the excited state CCSD methods still have polynomial scaling with large factor $\mathcal{O}\left(N^6\right)$ and are constrained to systems containing only 20-30 atoms without access to campus-level or national computing resources.

Recently, the development of *reduced scaling* variants of the coupled-cluster methods has been reinvigorated by Neese's introduction[76] of Pair Natural Orbitals (PNOs) in the context of local correlation formalisms of CC initiated by Pulay[69] and pursued by Werner[73,184] and others.[185,186] PNOs were originally proposed in the 1960s under the name Pseudo-natural Orbitals.[170,171] Truncation of PNOs significantly reduces the number of unoccupied orbitals while only introducing small errors in correlation energies in post-Hartree-Fock calculations. However, the demanding computational cost of the pair-specific integral transformation to the PNO space, which scales $\mathcal{O}\left(N^7\right)$ if there is no truncation of the PNO space, prevents the development of PNO-based electronic structure theories. In 2009, Neese *et al.* revived local PNOs (LPNOs) for the CEPA[76] and CCSD[172] methods, making use of density fitting approximations to accelerate the integral transformation process. It makes large-scale coupled-cluster possible for systems with up to 100 atoms using a small workstation. The LPNO approach was improved by imposing block sparsity into cluster operator amplitudes via domains of projected atomic orbitals (PAOs).[187,188] The DLPNO-CC method was subsequently improved via the linear-scaling density fitting[189,190] and the introduction of F12 explicit correlation to reduce the basis set error,[148, 80, 83] culminating in a linear scaling explicitly correlated CCSD(T) method for ground states. These developments were pursued in parallel by several other groups, with polynomial scaling PNO-CCSD(T) code demonstrated by Hättig and co-workers[149] and a scalable implementation of a linear-scaling PNO-CCSD(T)-F12 demonstrated by Werner and co-workers.[191]

The key ideas of modern reduced-scaling coupled-cluster methods apply not only to the ground states but also to the excited states. Two competing visions of how to formulate reduced-scaling excited state methodology have been explored. Hättig and Helmich have explored excited-state coupled-cluster methods by introducing the $\mathcal{O}\left(N^4\right)$ scaling PNO-EOM-CC2 with state-specific PNOs,[192] as well as PNO based CIS(D)[193] and ADC(2).[194] The common ideas to these developments is the use of state-specific PNOs to compress the cluster operator (computed in the ground state CC equation) and the excited state wave operators for each state, with excited states computed one at a time. Thus the total number of PNOs grows linearly with the number of excited states. Recently, Dutta *et al* presented a PNO-based coupled-cluster method for excited states utilizing the similarity-transformed EOM (STEOM) CCSD framework.[195, 196] In their approach the use of PNOs is limited to the ground state only, with DLPNO-CCSD amplitudes subsequently transformed to the canonical basis and used to evaluate the bt-PNO-STEOM-CCSD energies of manifolds of states, at a $\mathcal{O}\left(N^6\right)$ complexity but possible to reach $\mathcal{O}\left(N^5\right)$ with additional improvements. However, this approach back-transformed the PNOs to the canonical space in the equation of motion CCSD calculations, thus limiting the size of the system can be considered. We should also note that the use of local correlation ideas (namely, PAO domains) without the PNO-style compression, have been explored in the context of coupled-cluster methods for excitation energies, such as local EOM-CC2[197] and local EOM-CCSD.[185,184]

In this work, we present a PNO-based approach suitable for robust treatment of manifolds of excited states with the EOM-CCSD methods. The key idea is to use state-averaged PNOs similar to those used in the ground-state PNO coupled-cluster methods through state-averaged guess pair densities averaged over the target excited state manifold. To quickly explore the performance of our approach we simulated it using a massively parallel EOM-CCSD implementation. The new massively parallel EOM-CCSD was implemented in the Massively Parallel Quantum Chemistry (MPQC) package[113] using the TiledArray[198] framework, based on the ground-state CCSD implementation described previously.[150] The new implementation exhibits good strong-scaling parallel performance and allows the calculation of excitation energy for systems with more than 50 atoms and more than 1000 basis functions; this is crucial to the exploration of the state-averaged PNO ansatz for systems of realistic size. In Section 4.2, the theory and implementation of state-averaged PNOs are discussed. Section 4.3 describes the computational details as well as the computing resources used. Section 4.4 demonstrates the performance of the parallel EOM-CCSD code and the accuracy of state-averaged PNOs.

4.2 Methods

The coupled-cluster ground-state wave function,

$$\Psi_{(0)} \equiv e^{\hat{T}} \left|0\right\rangle , \tag{4.1}$$

where the $\left|0\right\rangle$ stands for the zeroth order reference wave function (usually a Hartree-Fock determinant), is determined by projection of the Schrödinger equation against excited determinants

$$0 = \left\langle {}^{\overline{a}}_{i}\right| \bar{H} \left|0\right\rangle , \tag{4.2}$$

$$0 = \left\langle {}^{\overline{ab}}_{ij}\right| \bar{H} \left|0\right\rangle . \tag{4.3}$$

with $\bar{H} \equiv e^{-\hat{T}} H e^{\hat{T}}$ the usual similarity-transformed Hamiltonian. Within the equation of motion coupled-cluster method[182,183] the excited-state wave functions are obtained in a CI fashion, by the action of a linear excitation operator acting on the ground-state CC wave function:

$$\Psi_{(k)} \equiv \hat{R}_{(k)} e^{\hat{T}} \left|0\right\rangle , \tag{4.4}$$

$\hat{R}_{(k)}$ and the corresponding energies $E_{(k)}$ are obtained by diagonalizing the similarity-transformed Hamiltonian:

$$\bar{H}\hat{R}_{(k)} \left|0\right\rangle = E_{(k)}\hat{R}_{(k)} \left|0\right\rangle . \tag{4.5}$$

In practice the ground and excited states are represented in terms of single and double excitations only:

$$\hat{T} \overset{\text{CCSD}}{\equiv} \hat{T}_1 + \hat{T}_2, \tag{4.6}$$

$$\hat{T}_1 \equiv T^i_a E^a_i, \tag{4.7}$$

$$\hat{T}_2 \equiv \frac{1}{2} T^{ij}_{ab} E^{ab}_{ij}. \tag{4.8}$$

$$\hat{R}_{(k)} \overset{\text{CCSD}}{\equiv} \delta_{k0} + \hat{R}_{1(k)} + \hat{R}_{2(k)}, \tag{4.9}$$

$$\hat{R}_{1(k)} \equiv B^i_{a(k)} E^a_i, \tag{4.10}$$

$$\hat{R}_{2(k)} \equiv \frac{1}{2} B^{ij}_{ab(k)} E^{ab}_{ij}. \tag{4.11}$$

The storage and operation costs of the CCSD and EOM-CCSD methods are $\mathcal{O}(N^4)$ and $\mathcal{O}(N^6)$, respectively.

Two-body amplitude tensors, T^{ij}_{ab} and B^{ij}_{ab}, are efficiently rank-compressed by transforming each ij block into the ij-specific subspace. For the ground-state amplitudes the optimal pair-specific subspaces are robustly approximated by the truncated singular subspace of the corresponding ground-state pair densities, $\mathbf{D}^{ij}_{(0)}$, computed from guess amplitudes:

$$\mathbf{D}^{ij}_{(0)} = \frac{2}{1+\delta_{ij}} (\mathbf{T}^{ij} \tilde{\mathbf{T}}^{ij\dagger} + \mathbf{T}^{ij\dagger} \tilde{\mathbf{T}}^{ij}), \tag{4.12}$$

where $\left(\mathbf{T}^{ij}\right)_{ab} \equiv T^{ij}_{ab}$ and $\tilde{\mathbf{T}}^{ij} \equiv 2\mathbf{T}^{ij} - \mathbf{T}^{ij\dagger}$. (Semi)canonical MP1 amplitudes,

$$(T^{(1)})^{ij}_{ab} \equiv \frac{G^{ij}_{ab}}{f^i_i + f^j_j - f^a_a - f^b_b}, \tag{4.13}$$

are typically used as the guess[172] (In Eq. (4.13) f are the matrix elements of the Fock operator, and G stands for the two-electron integral). PNOs are the basis for the singular subspace of $\mathbf{D}^{ij}_{(0)}$; they are obtained by solving the eigensystem:

$$\mathbf{D}^{ij}_{(0)}\mathbf{U}^{ij}_{(0)} = \mathbf{U}^{ij}_{(0)}\mathbf{n}^{ij}_{(0)}. \tag{4.14}$$

where $\mathbf{n}^{ij}_{(0)}$ are the PNO occupation numbers. PNOs with occupation numbers less than user-provided threshold T_{CutPNO} are omitted, hence the number of PNOs per pair is independent of the system size (i.e. $\mathcal{O}(1)$). One-body amplitudes, T^i_a and B^i_a, are compressed similarly to the two-body counterparts by transforming into the basis of orbital-specific virtuals (OSVs).[199] OSVs are traditionally defined to be identical to the PNOs of diagonal pairs but truncated according to a different threshold, T_{CutOSV}.

As pointed out by Hättig and Helmich,[193] the optimal singular subspaces for the ground and excited state amplitudes differ; as a result, the PNOs and OSVs must be constructed separately for the ground and excited states. Hättig and Helmich proposed the use of state-specific PNOs, where the PNOs for each state are constructed using CIS(D) doubles amplitudes with respect to that state:[192]

$$B^{ij}_{ab(k)} = \frac{K^{ij}_{ab(k)}}{\omega_{(k)} + f^i_i + f^j_j - f^a_a - f^b_b}, \tag{4.15}$$

$$K^{ij}_{ab(k)} = B^i_{c(k)}G^{cj}_{ab} + B^j_{c(k)}G^{ic}_{ab} - B^l_{a(k)}G^{ij}_{lb} - B^l_{b(k)}G^{ij}_{al}, \tag{4.16}$$

where $\mathbf{B}^i_{a(k)}$ and $\mathbf{B}^{ij}_{ab(k)}$ are the CIS singles amplitudes and CIS(D) doubles amplitudes for excited state k, and $\omega_{(k)}$ is the CIS excitation energy. The state-specific PNOs for excited states can be obtained from the state-specific pair density using the CIS(D) doubles amplitudes similar to the approach used in ground state:

$$\mathbf{D}^{ij}_{(k)} = \frac{2}{1+\delta_{ij}}(\mathbf{B}^{ij}_{(k)}\tilde{\mathbf{B}}^{ij\dagger}_{(k)} + \mathbf{B}^{ij\dagger}_{(k)}\tilde{\mathbf{B}}^{ij}_{(k)}), \tag{4.17}$$

Such definition of excited state PNOs yields good accuracy in the context of PNO-EOM-CC2 method.[192] However, there are several factors that prompted us to look beyond the state-specific PNOs. First, and foremost, the cost of PNO construction and integral transformation grow linearly with the number of states. This is particularly notable since the cost of PNO-based methods is often dominated by the cost of the integral transformation, even when domain approximations are employed.[189,190] Second, state-specific PNOs make it difficult to deal with degenerate state manifolds (which ideally need to be expressed in the same basis). Lastly, the use of state-specific PNOs increases the complexity of formalism and implementation.

Thus we decided to investigate PNO-EOM-CCSD that uses one set of PNOs for all excited states, in particular, we propose to use the *state-averaged* PNOs. The state-averaged PNOs are defined as the eigenvectors of averaged pair densities over an N-state manifold:

$$\mathbf{D}^{ij} = \frac{1}{N}\sum_{k}^{N} \mathbf{D}^{ij}_{(k)}. \tag{4.18}$$

(State-averaged OSVs will be defined in this work the PNOs of the diagonal pairs, in complete analogy with the construction of the ground-state OSVs).

Although the work is underway to develop a production implementation of reduced-scaling CC, here our goal is more modest: we aim to evaluate the proposed state-averaged PNO formulation in the context of EOM-CCSD. Hence we initially implemented a simulation for PNO-EOM-CCSD based on a newly-developed massively parallel canonical (i.e., $\mathcal{O}\left(N^6\right)$) EOM-CCSD program in the MPQC code. Note that simulation has been used previously for initial evaluation of locally-correlated PAO-based EOM-CCSD by Russ and Crawford[185,186] and by Korona and Werner.[184] Werner *et al.* and Crawford *et al.* also used simulation to compare PAO-, OSV-, and PNO-based formulations of CCSD.[200,201] It should also be noted that Hättig and Helmich have demonstrated *production*-level $\mathcal{O}\left(N^4\right)$ PNO-EOM-CC2 methods,[192] but PNO-EOM-CCSD has not yet been reported at the time of writing this manuscript.

The canonical closed-shell EOM-CCSD program in MPQC was implemented on top of the TiledArray tensor framework following the formalism of Bartlett and Stanton.[183] The implementation details generally follow the ground-state explicitly correlated CCSD implementation reported previously.[150] All amplitudes and intermediates are distributed in memory, and contractions are evaluated using the communication-optimal implementation of the distributed-memory scalable universal matrix multiplication algorithm (SUMMA) implemented in TiledArray.[202,203] Similarly to the ground-state CCSD, the largest intermediate needed to compute in the EOM-CCSD is the W^{cd}_{ab} term with four virtual indices:

$$W^{cd}_{ab} \equiv G^{cd}_{ab} - T^i_b G^{cd}_{ai} - T^i_a G^{cd}_{ib} + (T^{ij}_{ab} + T^i_a T^j_b) G^{cd}_{ij}. \tag{4.19}$$

When contracting with B^{ij}_{ab}, this intermediate can be avoided through a back-transformed intermediate:

$$W^{cd}_{ab} B^{ij}_{cd} = X^{ij}_{\rho\sigma} C^\rho_a C^\sigma_b - X^{ij}_{\sigma\rho} C^\rho_k C^\sigma_a T^k_b - X^{ij}_{\rho\sigma} C^\rho_k C^\sigma_b T^k_a + X^{ij}_{\rho\sigma} C^\rho_k C^\sigma_l (T^{kl}_{ab} + T^k_a T^l_b), \tag{4.20}$$

$$X^{ij}_{\rho\sigma} = (B^{ij}_{cd} C^c_\mu C^d_\nu) G^{\mu\nu}_{\rho\sigma}. \tag{4.21}$$

Computing intermediate X requires evaluating atomic two-electron integral on the fly. In this way, the storage requirements of the EOM-CCSD program have been reduced, allowing us to carry out calculations on systems with over 1000 basis functions. The same technique has been used by Kuś *et al.* in ACES III.[107]

The ground-state PNO-CCSD simulation was implemented with modification to the Jacobi update in the following manner:

1. After the CCSD amplitude residuals $\mathbf{R}_1$ and $\mathbf{R}_2$ are computed, $\mathbf{R}_1$ is transformed into a semi-canonical OSV basis and $\mathbf{R}_2$ is transformed into a semi-canonical PNO basis:

$$\bar{\mathbf{R}}^i = \mathbf{U}^{i\dagger}\mathbf{R}^i, \tag{4.22}$$

$$\bar{\mathbf{R}}^{ij} = \mathbf{U}^{ij\dagger}\mathbf{R}^{ij}\mathbf{U}^{ij}, \tag{4.23}$$

   where $\mathbf{R}^i/\mathbf{R}^{ij}$ are the corresponding orbital/pair blocks of $\mathbf{R}_1/\mathbf{R}_2$ residuals, and $\mathbf{U}^i/\mathbf{U}^{ij}$ are the ground-state OSV/PNO bases.

2. The residuals are updated through a Jacobi update in the OSV and PNO space:

$$\bar{\Delta}^i_{a_i} = \frac{\bar{R}^i_{a_i}}{f^i_i - \bar{f}^{a_i}_{a_i}}, \tag{4.24}$$

$$\bar{\Delta}^{ij}_{a_{ij}b_{ij}} = \frac{\bar{R}^{ij}_{a_{ij}b_{ij}}}{f^i_i + f^j_j - \bar{f}^{a_{ij}}_{a_{ij}} - \bar{f}^{b_{ij}}_{b_{ij}}}, \tag{4.25}$$

   where a_i and a_{ij} are unoccupied orbitals in the truncated OSV and PNO basis, respectively.

3. The updated residuals are extrapolated with DIIS and back-transformed into the canonical basis:

$$\mathbf{\Delta}^i = \mathbf{U}^i\bar{\mathbf{\Delta}}^i, \tag{4.26}$$

$$\mathbf{\Delta}^{ij} = \mathbf{U}^{ij}\bar{\mathbf{\Delta}}^{ij}\mathbf{U}^{ij\dagger}. \tag{4.27}$$

4. The new CCSD amplitudes are formed as an update to the current amplitudes:

$$\mathbf{T}^{i}_{n+1} = \mathbf{T}^{i}_{n} + \mathbf{\Delta}^{i}, \tag{4.28}$$

$$\mathbf{T}^{ij}_{n+1} = \mathbf{T}^{ij}_{n} + \mathbf{\Delta}^{ij}, \tag{4.29}$$

   where n stands for the number of current iteration.

5. The CCSD residuals are recomputed using the new amplitudes, and this process is repeated from step 1 until convergences is reached.

Similarly, the state-averaged PNO simulation in EOM-CCSD can be done with modification in the Davidson solver:

1. The residuals produced by the Davidson algorithm are transformed into the OSV and PNO bases:

$$\bar{\mathbf{R}}^{i}_{(k)} = \mathbf{U}^{i\dagger}\mathbf{R}^{i}_{(k)}, \tag{4.30}$$

$$\bar{\mathbf{R}}^{ij}_{(k)} = \mathbf{U}^{ij\dagger}\mathbf{R}^{ij}_{(k)}\mathbf{U}^{ij}. \tag{4.31}$$

2. A preconditioner is applied to the residuals in the OSV and PNO spaces:

$$\bar{B}^{i}_{a_i(k)} = \frac{\bar{R}^{i}_{a_i(k)}}{\omega_{(k)} + f^{i}_{i} - \bar{f}^{a_i}_{a_i}}, \tag{4.32}$$

$$\bar{B}^{ij}_{a_{ij}b_{ij}(k)} = \frac{\bar{R}^{ij}_{a_{ij}b_{ij}(k)}}{\omega_{(k)} + f^{i}_{i} + f^{j}_{j} - \bar{f}^{a_{ij}}_{a_{ij}} - \bar{f}^{b_{ij}}_{b_{ij}}}, \tag{4.33}$$

   where $\omega_{(k)}$ is the eigenvalue of state k.

3. The updated trial vectors are projected back into the canonical space:

$$\mathbf{B}^{i}_{(k)} = \mathbf{U}^{i}\bar{\mathbf{B}}^{i}_{(k)}, \tag{4.34}$$

$$\mathbf{B}^{ij}_{(k)} = \mathbf{U}^{ij}\bar{\mathbf{B}}^{ij}_{(k)}\mathbf{U}^{ij\dagger}. \tag{4.35}$$

4. The new trial vectors are added to the next iteration of the Davidson algorithm to update the subspace, and the process is continued from step 1 until it reached convergence.

4.3 Computational Details

The canonical EOM-CCSD code was implemented and tested in the developmental version of the MPQC program.[113] All computations were performed on a commodity cluster at Virginia Tech, each node of which has two Intel Xeon E5-2670 CPUs (332 GFLOPS) and 64 GB of RAM. MPQC was compiled using GCC 5.3.0 with Intel MPI 5.0 and the serial Intel MKL version 11.2.3. All computations launched 1 MPI process per node with 16 threads per MPI process, with the orbital block size set to 20.

The state-averaged PNO-EOM-CCSD simulation code was also implemented in MPQC. To simplify the definition of PNOs and OSVs truncation, we assumed T_{CutOSV}=0.1$*T_{\mathrm{CutPNO}}$ for both ground and excited states. Neither domain nor weak pair approximations were utilized. The occupied MOs were localized in all calculations via the Foster-Boys algorithm.[204,205] The density-fitting (resolution-of-identify) approximation[163,159] and frozen core approximation were used for all the calculations performed in this work. We have used the cc-pVTZ[206] and aug-cc-pVD/TZ[207] atomic orbital basis sets in our calculations, with the corresponding auxiliary basis sets cc-pVTZ-RI and aug-cc-pVD/ TZ-RI[208] for density-fitting. In Section 4.4, the geometries of the methylated uracil dimer with water and the phenolate form of the anionic chromophore of the photoactive yellow protein were obtained from Ref. 165. The structure of 11-cis-retinal protonated Schiff base was obtained from 195. The structures of benzonitrile and acetamide were obtained from Ref. 4. In Section 4.4.3, a total number of 10 excited-states were computed for the benchmark dataset of 28 organic molecules by Thiel.[4]

4.4 Results & Discussion

4.4.1 Parallel Performance of EOM-CCSD

The new canonical EOM-CCSD code can attain high efficiency and good parallel scalability as illustrated in Fig. 1 and Fig. 2 for realistic computations (with aug-cc-pVDZ and cc-pVTZ basis sets) on excited-states of the methylated uracil dimer with water and 11-cis-retinal protonated Schiff base, respectively. The data in Fig. 4.1 corresponds to a ~ 5 speedup from 16 to 128 nodes with the aug-cc-pVDZ basis and a ~ 3 speedup from 32 to 128 nodes with the cc-pVTZ basis. The data in Fig. 4.2 corresponds to a ~ 5.5 speedup from 16 to 128 nodes with the aug-cc-pVDZ basis and a ~ 1.6 speedup from 64 to 128 nodes with the cc-pVTZ basis. The demonstrated strong scaling is not as impressive as that of the

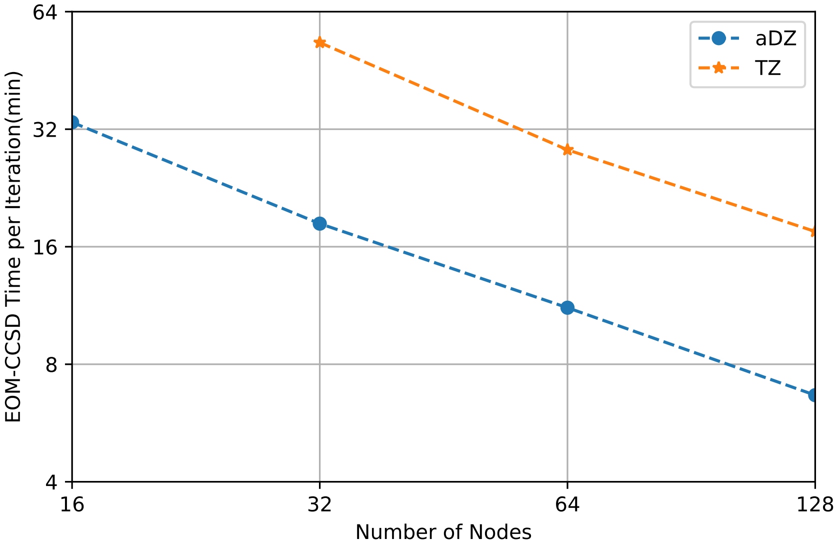

Figure 4.1: Parallel performance of EOM-CCSD on 4 states of the methylated uracil dimer with water (39 atoms) with the aug-cc-pVDZ (645) and cc-pVTZ (882) basis sets

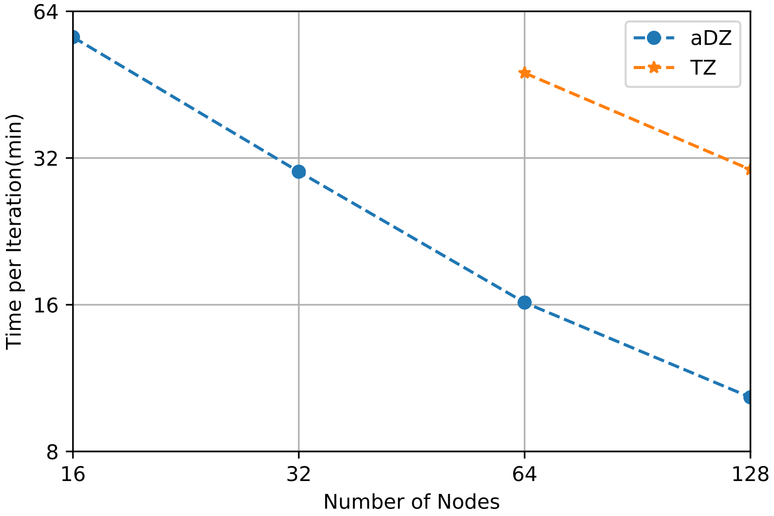

Figure 4.2: Parallel performance of EOM-CCSD on 4 states of the 11-cis-retinal protonated Schiff base (51 atoms) with the aug-cc-pVDZ (753) and cc-pVTZ (1050) basis sets

ground-state CCSD program,[150] but additional improvements are planned. The performance of our code is already sufficient to be able to treat multiple states of a system with 50-100 atoms and 1000-1500 basis functions.

4.4.2 Accuracy of State-Averaged PNOs

To quantify the performance of state-averaged PNOs we computed errors in excitation energies relative to the canonical EOM-CCSD values introduced by the truncation of PNOs (and the corresponding truncation of OSVs). Table 4.1 lists the PNO truncation errors for benzonitrile in the cc-pVTZ basis for a fixed value of the T_{CutPNO} parameter, as a function of the number of computed states.

As expected, the average number of excited-state PNOs (ESnPNO) increases with the total number of states. However, the rate of increase is rather modest: raising the number of states from 1 to 20 increases the number of PNOs only by a factor of ~ 2. Clearly, the total number of state-averaged PNOs grows with the number of states far slower than the linear growth of the total number of state-specific PNOs used by Hättig and co-workers. This is not entirely surprising; since the low-energy states in molecules to zeroth order have many occupied orbitals in common; correlation effects will be largely similar among the states. Clearly, the use of state-averaged PNOs should offer substantial savings in the costs of the integral transformation. The errors in excitation energies also decrease as the total number of states increases because of the concomitant increase in the number of PNOs. On average

Table 4.1: Truncation errors in excitation energy (eV) of benzonitrile (cc-pVTZ,V=283 [a]) with respect to total number of states at T_{CutPNO}=10^{-8}

nStates ESnPNO	1 63	2 79	4 91	6 95	8 101	10 104	20 133
S_1	0.0072	0.0016	0.0014	0.0003	-0.0001	-0.0001	-0.0011
S_2		0.0043	0.0023	0.0008	0.0005	0.0004	-0.0005
S_3			0.0235	0.0039	0.0019	0.0019	0.0005
S_4			0.0263	0.0022	0.0022	0.0022	0.0009
S_5				0.0161	0.0173	0.0171	0.0040
S_6				0.0069	0.0049	0.0042	0.0004
S_7					0.0876	0.0235	0.0050
S_8					0.1834	0.0557	0.0012
S_9						0.0201	-0.0001
S_{10}						0.0058	0.0044
MAE [b]	0.0072	0.0030	0.0134	0.0050	0.0372	0.0131	0.0018
MAX [c]	0.0072	0.0043	0.0263	0.0161	0.1834	0.0557	0.0050

[a] Total number of unoccupied orbitals

[b] Mean absolute error

[c] Max absolute error

our approach to PNO construction is rather accurate: the mean absolute errors are below 0.02 eV for all cases except nStates=8, which has a mean absolute error of 0.037 eV. These errors are small relative to the average accuracy of the EOM-CCSD model even for states with single excitation character.

Note that the mean absolute errors do not smoothly decrease as nStates increases. This is correlated with sporadic increases in the maximum absolute errors as the number of states is increased, such as in the case of states 3 and 4 when nStates is at 4 and states 7 and 8 when nStates is at 8. However, these errors are significantly reduced when nStates is increased to 6 and 10, respectively. This indicates that the highest excited states in the computed manifold sometimes have more significant errors with state-averaged PNOs, which can be observed from the nStates = 4,8 data. The reason for this behavior is that the composition of N lowest EOM-CCSD states may not be similar to that of CIS, either due to pure root flipping or, more generally, due to nonperturbative effects of dynamical correlation on the excited state character and ordering. Analysis of the excited states in this example suggests that CIS states 3, 4, 5, and 6 become EOM-CCSD states 5, 6, 3, and 4. Therefore accurate description of EOM-CCSD states 3 and 4 will require including pair densities from CIS(D) states 5 and 6. This is not a serious issue since in excited state computations to increase the probability that N lowest-energy target states have been reproduced is to compute $M > N$ states. Hence, a slightly larger error in a few of the highest excited states would not be an issue since typically the number of *computed* states is always greater than the number of *target* states.

Lastly, note that the T_{CutPNO} threshold is kept constant in Table 1. Therefore the averaged errors decrease as the number of states increases, at the cost of increasing the average number of state-averaged PNOs per pair. Clearly, if we wanted to keep the average error per state constant we could loosen the T_{CutPNO} threshold as the number of states is increased. This would further alleviate the modest increase of the total number of PNOs with the number of states. Dependence of the error on T_{CutPNO} will be examined next.

Tables 4.2, 4.3 and 4.4 illustrate the correlation between the T_{CutPNO} parameter and the errors in the excitation energies of the 4 lowest singlet excited states of the phenolate form of the anionic chromophore of the photoactive yellow protein (PYPb). Since T_{CutPNO} affects the EOM-CCSD excitation energies through both ground-state ($\hat{T}$) and excited-state ($\hat{R}$) operators, we examined its effects separately on the ground-state cluster operators only (Table 4.2), excited-state operators (Table 4.3) and both (Table 4.4). As expected (see Table 4.2) truncating the ground-state PNOs only lowers the excitation energies (the errors in excited states are all negative) since the energy of the ground state becomes higher due to a decrease in the amount of correlation energy that is recovered. Similarly, truncating the excited-state PNOs raises the excitation energy since the calculated energy of the excited states is now higher as a result of recovering less of the correlation energy (Table 4.3). When both the ground and excited state PNOs are truncated, the opposite signs of the two sources of error partially cancel each other out, leading to smaller errors in the excitation

Table 4.2: Truncation error in excitation energy (eV) of PYPb (aug-cc-pVDZ, V=296 [a]) by only truncating the ground-state PNOs

T_{CutPNO} GSnPNO	10^{-6} 13	10^{-7} 25	10^{-8} 45	10^{-9} 74	10^{-10} 110
S_1	-0.0872	-0.0289	-0.0092	-0.0030	-0.0011
S_2	-0.0892	-0.0301	-0.0098	-0.0033	-0.0012
S_3	-0.0855	-0.0287	-0.0093	-0.0031	-0.0012
S_4	-0.0875	-0.0296	-0.0098	-0.0033	-0.0013

[a] Total number of unoccupied orbitals

Table 4.3: Truncation error in excitation energy (eV) of PYPb (aug-cc-pVDZ, V=296 [a]) by only truncating the excited-state PNOs

T_{CutPNO}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
ESnPNO	6	15	32	58	94
S_1	0.1323	0.0564	0.0211	0.0053	0.0015
S_2	0.1215	0.0525	0.0197	0.0047	0.0011
S_3	0.1354	0.0635	0.0256	0.0078	0.0024
S_4	0.1296	0.0611	0.0261	0.0092	0.0035

[a] Total number of unoccupied orbitals

Table 4.4: Truncation error in excitation energy (eV) of PYPb (aug-cc-pVDZ, V=296 [a]) by truncating both the ground and excited-state PNOs

T_{CutPNO}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
GSnPNO	13	25	45	74	110
ESnPNO	6	15	32	58	94
S_1	0.0457	0.0276	0.0119	0.0022	0.0003
S_2	0.0332	0.0226	0.0098	0.0013	-0.0002
S_3	0.0500	0.0348	0.0163	0.0046	0.0012
S_4	0.0422	0.0315	0.0163	0.0058	0.0022

[a] Total number of unoccupied orbitals

energy, as can be seen in Table 4.4. However, this error cancellation may lead to occasional non-monotonic convergence.

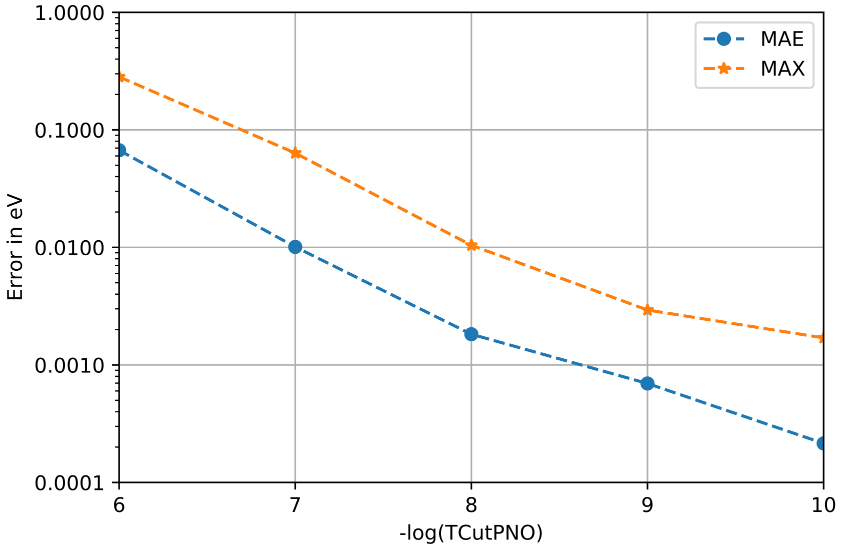

Figure 4.3: Mean absolute (MAE) and maximum (MAX) PNO truncation errors (in eV) of PNO-EOM-CCSD/cc-pVTZ excitation energies for the 28-molecule benchmark set.

4.4.3 Error Analysis

To further test the performance of state-averaged PNOs, we used the PNO-EOM-CCSD method to compute the excitation energies of the lowest six singlet excited states of 28 organic molecules in the benchmark dataset from Thiel *et al.*[4] Variation of statistical measures of the errors with the T_{CutPNO} parameter are presented in Fig. 4.3 and Fig. 4.4 for cc-pVTZ and aug-cc-pVTZ basis sets, respectively. The corresponding average numbers of ground-state and excited-state PNOs are shown in Fig. 4.5 and Fig. 4.6, respectively.

The average excitation energy errors become smaller than 0.1 eV already with T_{CutPNO}=10^{-6}, further reduce to below 0.02 eV with T_{CutPNO}=10^{-7}, and further decrease monotonically with T_{CutPNO} for both basis sets. The maximum errors also decrease monotonically, but

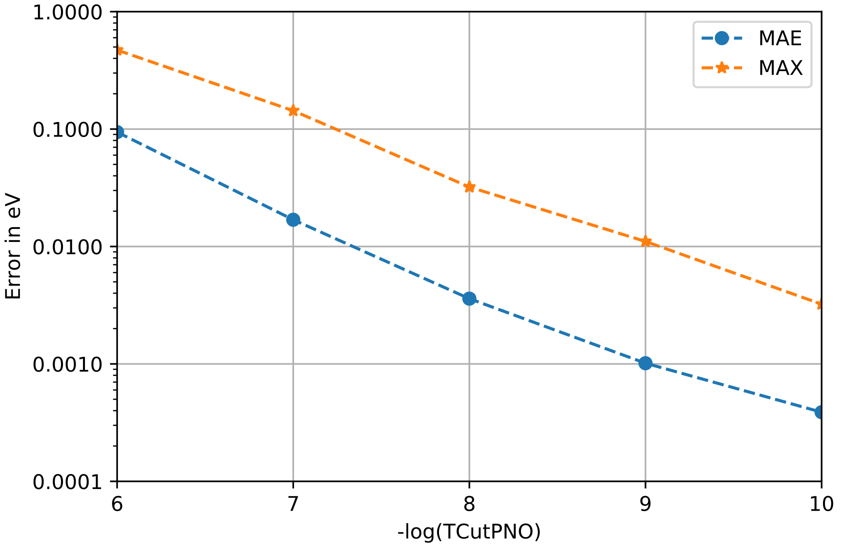

Figure 4.4: Mean absolute (MAE) and maximum (MAX) PNO truncation errors (in eV) of PNO-EOM-CCSD/aug-cc-pVTZ excitation energies for the 28-molecule benchmark set.

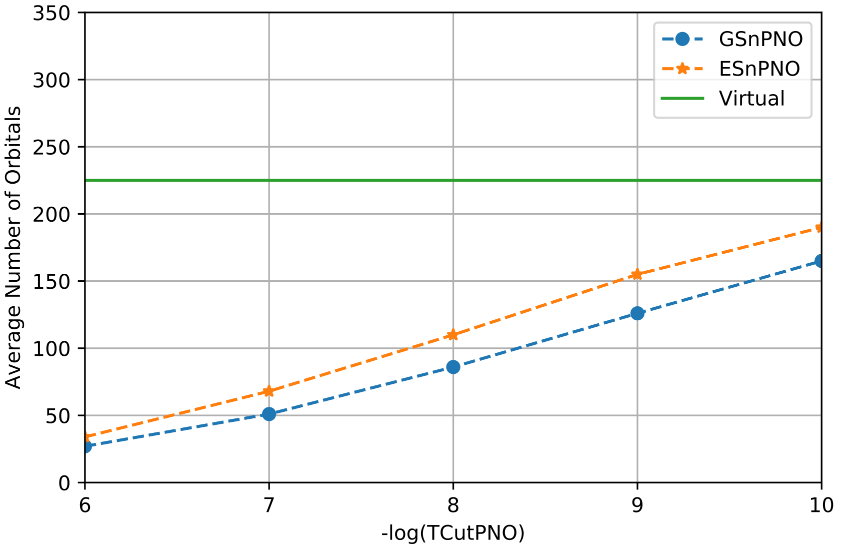

Figure 4.5: Convergence of average number of PNOs per pair per molecule in ground state (GSnPNO) and excited states (ESnPNO) of PNO-EOM-CCSD/cc-pVTZ

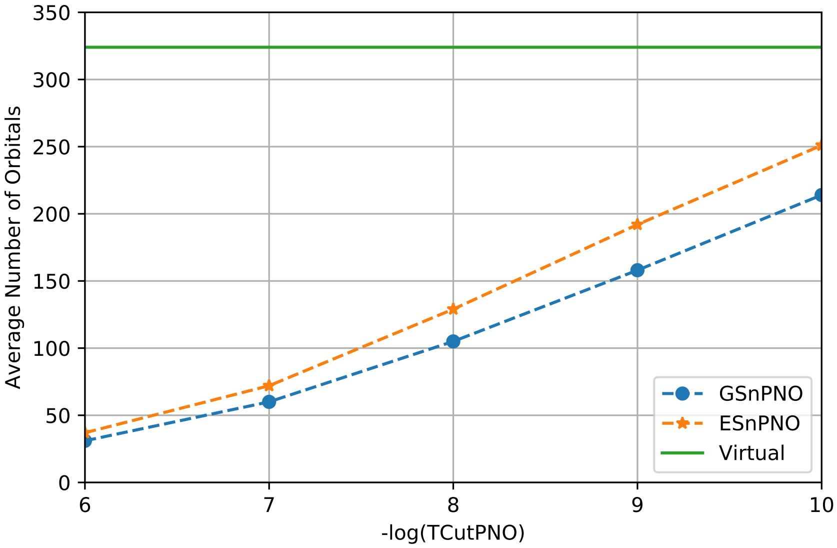

Figure 4.6: Convergence of average number of PNOs per pair per molecule in ground state (GSnPNO) and excited states (ESnPNO) of PNO-EOM-CCSD/aug-cc-pVTZ

require tighter truncation, T_{CutPNO}=$\{10^{-7}, 10^{-8}\}$ for the {cc-pVTZ,aug-cc-pVTZ} basis set, to reduce below 0.1 eV. Hence, a threshold of 10^{-7} is suitable for general applications while a threshold of 10^{-8} should be sufficient for high-accuracy applications. With these thresholds, the average numbers of ground/excited state PNOs are ∼50/70 and ∼100/120 for cc-pVTZ and aug-cc-pVTZ basis, respectively, which is a significant decrease over the average number of total virtual orbitals. For all values of T_{CutPNO} the number of excited state PNOs is only 20%-30% higher than the corresponding number of ground state PNOs.

4.4.4 Rydberg and Charge Transfer States

To study the accuracy of state-averaged PNO-EOM-CCSD on excited states with Rydberg and charge transfer character we selected two prototypical examples: states S_1 and S_4 of acetamide (Table 4.5) and states S_1 and S_2 of the ethylene-tetrafluoroethylene ($C_2H_4-C_2F_4$) model[1] (Table 4.6). The latter model was also used to test other PNO-based excited state methods, by Hättig and Helmich[192] and by Dutta *et al.*[195]

Table 4.5: Truncation errors (eV) of the aug-cc-pVTZ [a] PNO-EOM-CCSD excitation energies of the four lowest singlet states of acetamide . States S_1 and S_4 have strong Rydberg character. Excited-state PNOs were averaged over lowest 10 states.

T_{CutPNO}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
ESnPNO	32	69	125	191	245
S_1	0.0699	0.0052	-0.0017	-0.0009	-0.0002
S_2	0.0038	-0.0058	-0.0032	-0.0012	-0.0005
S_3	0.0185	-0.0001	-0.0028	-0.0011	-0.0003
S_4	0.0464	0.0012	-0.0023	-0.0004	0.0000

[a] Total number of unoccupied orbitals = 283.

The truncation errors for the two Rydberg states (S_1 and S_4) were found to be somewhat larger than the errors for the valence states (S_2 and S_3) with $T_{\text{CutPNO}}=10^{-6}$ but they are comparable with $T_{\text{CutPNO}}=10^{-7}$ or tighter. Overall, no significant differences in the performance of excited state PNOs is observed for Rydberg and non-Rydberg states.

The truncation errors for the two charge transfer states were found to be substantially larger

Table 4.6: Truncation errors (eV) of the aug-cc-pVDZ [a] PNO-EOM-CCSD excitation energies of the four lowest singlet states of the C_2H_4-C_2F_4 dimer separater by 10 a.u. (Ref. 1). States S_1 and S_2 have charge-transfer character. Excited-state PNOs were averaged over lowest 10 states.

T_{CutPNO}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
ESnPNO	8	19	37	63	93
S_1	0.1330	0.0321	0.0087	0.0022	0.0006
S_2	0.2059	0.0513	0.0100	0.0014	0.0001
S_3	0.0153	0.0013	0.0004	0.0003	0.0001
S_4	0.0164	0.0005	-0.0001	-0.0000	0.0000

[a] Total number of unoccupied orbitals = 188.

than the valence states at all truncation thresholds. T_{CutPNO}=10^{-7} was required to reduce the errors to below 0.1 eV, and T_{CutPNO}=10^{-8} is sufficient to reduce the errors to 0.01 eV for charge transfer excitations. This is in agreement with the findings of Hättig and Helmich,[193] who pointed out that it requires more PNOs to get the same accuracy for charge transfer excitations. They attributed this to the use of the semicanonical CIS(D) amplitudes in constructing the excited-state PNOs (Eq. (4.15)); with localized occupied orbitals the off-diagonal matrix elements of the Fock operator are substantial and cannot be neglected. Nevertheless, the performance of semicanonical PNOs is still acceptable.

4.5 Conclusions

We proposed the use of state-averaged PNO ansatz for efficient and simple treatment of manifolds of excited states in the context of reduced-scaling excited-state many-body methods. We evaluated the performance of the state-averaged PNO ansatz in the context of PNO-EOM-CCSD method for prediction of excitation energies. The PNO-EOM-CCSD implementation is based on a new massively parallel canonical implementation of EOM-CCSD in the MPQC program. The state-averaged PNO-EOM-CCSD approach has been tested on the first six excited states of 28 organic molecules, yielding an average truncation error below 0.020 eV at T_{CutPNO}=10^{-7} for both the cc-pVTZ and aug-cc-pVTZ basis sets. With this truncation threshold, the number of state-averaged PNOs is reduced by more than 70% for cc-pVTZ and 80% for aug-cc-pVTZ. Overall, the state-averaged PNOs provide excellent accuracy for low lying valence and Rydberg states, but more PNOs are required to achieve the same accuracy for charge transfer states. These results encourage further development in this area, including implementing a production-level PNO-EOM-CCSD code as well as implementing the reduced-scaling DLPNO-EOM-CCSD method.

Chapter 5

Summary and Outlook

This work concentrates on extending the reach of the coupled-cluster methods to larger molecular systems. Two directions have been pursued to achieve this goal:

- using techniques of parallel computing on modern super-computers to extend the application range of explicitly correlated coupled-cluster methods, including the CCSD(2)$_{\overline{\text{F12}}}$ and CCSD(T)$_{\overline{\text{F12}}}$ methods,
- using a tensor compression technique (truncation of PNOs) to examine the reduced-scaling algorithm for the excited-state coupled-cluster model (EOM-CCSD).

In the first part of the research, reducing the amount of memory required is critical to treat large molecular systems. Even with distributed memory storage, it would still be difficult to store the intermediates in the CCSD calculations. The CCSD program uses two approaches to reduce the amount of memory requirement i) the *hybrid* DF-AO algorithm in

Chapter 2 and ii) the lazy evaluation of intermediates using density fitting in Chapter 3. The TiledArray tensor framework is used throughout the program to handle data management in distributed memory and parallel tensor operations. The inclusion of perturbative correction to basis set incompleteness error from explicitly correlated methods significantly improves the convergence of the coupled-cluster methods towards the complete basis set limit. However, the explicitly correlated coupled-cluster methods currently available in MPQC4 now only calculate the energies of closed-shell molecular systems. Part of the future work would be extending the current application to handle open-shell systems as well as to compute coupled-cluster properties. Moreover, there are still plenty rooms to improve the efficiency of performance of the current implementation. Firstly, the current CCSD implementation in MPQC4 does not take advantage of permutation symmetry in the doubles amplitudes and integrals, which increases the CPU time of the calculation by a factor of 2-3 based on our benchmark. It would require fundamental changes in the design of the TiledArray tensor framework to support tensor operations that can utilize permutation symmetry. Secondly, the current implementation would have considerable potential for acceleration if it can take advantage of hardware accelerators such as graphics processing units (GPUs) and field-programmable gate arrays (FPGAs), which requires the TiledArray tensor framework to have support for these hardware accelerators.

In the second part of the research, choosing the appropriate PNOs for excited states is essential to the accuracy of the PNO-EOM-CCSD method. Unlike the previous approaches that use the state-specific PNOs for excited states, we propose to use the state-averaged PNOs, which significantly simplifies the PNO-EOM-CCSD formulation and provides excellent accuracy for excitation energies. Even though using the state-specific PNOs for excited states leads to higher number of average PNOs than the ground-state PNOs, it can still reduce the size of the unoccupied space by a significant amount. However, the current research only explored the effect of truncating the state-averaged PNOs. It would be fascinating to introduce the domain approximation to the PNO-EOM-CCSD model (DLPNO-EOM-CCSD). Moreover, the current implementation of the PNO-EOM-CCSD method is a simulation code based on an existing canonical EOM-CCSD program in MPQC4, which does not provide any computation savings by truncating the state-averaged PNOs. Since this project has proven that the state-averaged PNOs are very useful in representing excited states amplitudes, it is worth to implement a production-level PNO-EOM-CCSD program, which can be used to study the excited states of larger molecular systems than current software can do, using the state-averaged PNOs.

At last, the ultimate approach to extend the coupled-cluster methods to large molecular systems will be combining the two strategies used in this work: massively parallel reduced-scaling coupled-cluster methods. The reduced-scaling coupled-cluster methods implemented to work on a single workstation can already treat much larger systems than conventional approaches. For example, the DLPNO-CCSD(T) method can manage the entire Crambin protein, which contains 644 atoms and over 6400 basis functions, with about 30 days using a single core.[188] Introducing parallel computing to the reduced-scaling coupled-cluster methods can efficiently reduce the amount of computation time and extend the reach of reduced-scaling methods to even bigger molecular systems.

Appendix A

Publication List

Published:

1. **C. Peng**, J. A. Calvin, F. Pavošević, J. Zhang and E. F. Valeev, "Massively parallel implementation of explicitly correlated coupled-cluster singles and doubles using TiledArray framework" *J. Phys. Chem. A* 2016, **120**, 10231.

2. F. Pavošević, **C. Peng**, P. Pinski, C. Riplinger, F. Neese and E. F. Valeev, "SparseMaps - A systematic infrastructure for reduced scaling electronic structure methods. V. Linear scaling explicitly correlated coupled-cluster method with pair natural orbitals" *J. Chem. Phys.* 2017, **146**, 174108.

3. F. Pavošević, **C. Peng**, J. V. Ortiz and E. F. Valeev, "Explicitly correlated formalism for second-order single-particle Green's function" *J. Chem. Phys.* 2017, **147**, 121101.

Submitted:

4. **C. Peng**, M. C. Clement and E. F. Valeev, "Exploration of reduced scaling formulation of equation of motion coupled-cluster singles and doubles based on state-averaged pair natural orbitals" *(submitted)*

In preparation:

52. **C. Peng**, J. A. Calvin and E. F. Valeev, "Efficient massively parallel coupled-cluster singles, doubles and perturbative triples correction with density fitting approximation" *(in preparation)*

Bibliography

[1] A. Dreuw, J. L. Weisman, and M. Head-Gordon, *J. Chem. Phys.* **119**, 2943 (2003).

[2] E. Schrödinger, *Annalen der Physik* **385**, 437 (1926).

[3] J. Řezáč, K. E. Riley, and P. Hobza, *J. Chem. Theory Comput.* **7**, 2427 (2011).

[4] M. Schreiber, M. R. Silva-Junior, S. P. Sauer, and W. Thiel, *J. Chem. Phys.* **128**, 134110 (2008).

[5] A. Szabo and N. S. Ostlund, *Dover Publications* (1996).

[6] M. Born and R. Oppenheimer, *Annalen der Physik* **389**, 457 (1927).

[7] F. Pacati and S. Boffi, *Phys. Rev. C* **2**, 1205 (1970).

[8] D. Cremer, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **1**, 509 (2011).

[9] A. C. Scheiner, G. E. Scuseria, J. E. Rice, T. J. Lee, and H. F. S. III, *J. Chem. Phys.* **87**, 5361 (1987).

[10] J. Hinze and F. F. Chemie, *Int. J. Quantum Chem., Quantum Chem. Symp.* **15**, 69 (1981).

[11] P. G. Szalay, T. Müller, G. Gidofalvi, H. Lischka, and R. Shepard, *Chem. Rev.* **112**, 108 (2012).

[12] V. Veryazov, P. Å. Malmqvist, and B. O. Roos, *Int. J. Quantum Chem.* **111**, 3329 (2011).

[13] B. O. Roos, *Multiconfigurational quantum chemistry for ground and excited states*, pp. 125–156, Springer Netherlands, Dordrecht, 2008.

[14] K. Andersson, P. Å. Malmqvist, and B. O. Roos, *J. Chem. Phys.* **96**, 1218 (1992).

[15] R. B. Murphy and R. P. Messmer, *Chem. Phys. Lett.* **183**, 443 (1991).

[16] P. Celani and H.-J. Werner, *J. Chem. Phys.* **112**, 5546 (2000).

[17] B. O. Roos, K. Andersson, M. P. Fülscher, P. Å. Malmqvist, L. Serrano-Andrés, K. Pierloot, and M. Merchán, *Advances in Chemical Physics: New Methods in Computational Quantum Mechanics, Volume 93* , 219 (1996).

[18] K. G. Dyall, *J. Chem. Phys.* **102**, 4909 (1995).

[19] C. Angeli, R. Cimiraglia, S. Evangelisti, T. Leininger, and J. P. Malrieu, *J. Chem. Phys.* **114**, 10252 (2001).

[20] K. R. Shamasundar, G. Knizia, and H.-J. Werner, *J. Chem. Phys.* **135**, 054101 (2011).

[21] T. Kato, *Comm. Pure Appl. Math* **10**, 151 (1957).

[22] E. A. Hylleraas, *Z. Physik* **54**, 347 (1929).

[23] W. A. Lester and M. Krauss, *J. Chem. Phys.* **41**, 1407 (1964).

[24] S. F. Boys and N. C. Handy, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **309**, 209 (1969).

[25] W. A. Lester, Jr., L. Mitas, and B. Hammond, *Chem. Phys. Lett.* **478**, 1 (2009).

[26] L. Kong, F. A. Bischoff, and E. F. Valeev, *Chem. Rev.* **112**, 75 (2012).

[27] W. Kutzelnigg, *Theoretica chimica acta* **68**, 445 (1985).

[28] E. F. Valeev, *Chem. Phys. Lett.* **395**, 190 (2004).

[29] S. Ten-no, *J. Chem. Phys.* **121**, 117 (2004).

[30] S. Ten-no, *Chem. Phys. Lett.* **398**, 56 (2004).

[31] B. J. Persson and P. R. Taylor, *J. Chem. Phys.* **105**, 5915 (1996).

[32] D. P. Tew and W. Klopper, *J. Chem. Phys.* **125**, 094302 (2006).

[33] D. P. Tew and W. Klopper, *J. Chem. Phys.* **123**, 074101 (2005).

[34] E. F. Valeev and C. L. Janssen, *J. Chem. Phys.* **121**, 1214 (2004).

[35] W. Kutzelnigg and W. Klopper, *J. Chem. Phys.* **94**, 1985 (1991).

[36] W. Klopper and C. C. Samson, *J. Chem. Phys.* **116**, 6397 (2002).

[37] J. Noga and W. Kutzelnigg, *J. Chem. Phys.* **101**, 7738 (1994).

[38] T. Shiozaki, E. F. Valeev, and S. Hirata, *J. Chem. Phys.* **131**, 044118 (2009).

[39] T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, *J. Chem. Phys.* **130**, 054101 (2009).

[40] T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, *J. Chem. Phys.* **129**, 071101 (2008).

[41] H. Fliegl, W. Klopper, and C. Hättig, *J. Chem. Phys.* **122**, 084107 (2005).

[42] C. Hättig, D. P. Tew, and A. Köhn, *J. Chem. Phys.* **132**, 231102 (2010).

[43] T. B. Adler, G. Knizia, and H.-J. Werner, *J. Chem. Phys.* **127**, 221106 (2007).

[44] G. Knizia, T. B. Adler, and H. J. Werner, *J. Chem. Phys.* **130** (2009).

[45] E. F. Valeev, *Phys. Chem. Chem. Phys.* **10**, 106 (2008).

[46] M. Torheyden and E. F. Valeev, *Phys. Chem. Chem. Phys.* **10**, 3410 (2008).

[47] E. F. Valeev and T. Daniel Crawford, *J. Chem. Phys.* **128**, 244113 (2008).

[48] L. Kong and E. F. Valeev, *J. Chem. Phys.* **135**, 214105 (2011).

[49] J. Noga, S. Kedžuch, and J. Šimunek, *J. Chem. Phys.* **127**, 034106 (2007).

[50] T. Shiozaki, G. Knizia, and H.-J. Werner, *J. Chem. Phys.* **134**, 034113 (2011).

[51] T. Shiozaki and H.-J. Werner, *J. Chem. Phys.* **133**, 141103 (2010).

[52] Y. Guo, K. Sivalingam, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **147**, 064110 (2017).

[53] K. Aidas, C. Angeli, K. L. Bak, V. Bakken, R. Bast, L. Boman, O. Christiansen, R. Cimiraglia, S. Coriani, P. Dahle, E. K. Dalskov, U. Ekstrm, T. Enevoldsen, J. J. Eriksen, P. Ettenhuber, B. Fernndez, L. Ferrighi, H. Fliegl, L. Frediani, K. Hald, A. Halkier, C. Httig, H. Heiberg, T. Helgaker, A. C. Hennum, H. Hettema, E. Hjertens, S. Hst, I.-M. Hyvik, M. F. Iozzi, B. Jansk, H. J. A. Jensen, D. Jonsson, P. Jrgensen, J. Kauczor, S. Kirpekar, T. Kjrgaard, W. Klopper, S. Knecht, R. Kobayashi, H. Koch, J. Kongsted, A. Krapp, K. Kristensen, A. Ligabue, O. B. Lutns, J. I. Melo, K. V. Mikkelsen, R. H. Myhre, C. Neiss, C. B. Nielsen, P. Norman, J. Olsen, J. M. H. Olsen, A. Osted, M. J. Packer, F. Pawlowski, T. B. Pedersen, P. F. Provasi, S. Reine, Z. Rinkevicius, T. A. Ruden, K. Ruud, V. V. Rybkin, P. Saek, C. C. M. Samson, A. S. de Mers, T. Saue, S. P. A. Sauer, B. Schimmelpfennig, K. Sneskov, A. H. Steindal, K. O. Sylvester-Hvid, P. R. Taylor, A. M. Teale, E. I. Tellgren, D. P. Tew, A. J. Thorvaldsen, L. Thgersen, O. Vahtras, M. A. Watson, D. J. D. Wilson, M. Ziolkowski,

and H. GREN, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **4**, 269 (2014).

[54] F. Furche, R. Ahlrichs, C. Httig, W. Klopper, M. Sierka, and F. Weigend, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **4**, 91 (2014).

[55] C. L. Janssen, I. B. Nielsen, M. L. Leininger, E. F. Valeev, J. P. Kenny, and E. T. Seidl, The Massively Parallel Quantum Chemistry Program (MPQC), 2008.

[56] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schtz, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2**, 242 (2012).

[57] F. Coester, *Nucl. Phys.* **7**, 421 (1958).

[58] F. Coester and H. Kümmel, *Nucl. Phys.* **17**, 477 (1960).

[59] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).

[60] J. Noga, W. Kutzelnigg, and W. Klopper, *Chem. Phys. Lett.* **199**, 497 (1992).

[61] A. Köhn, G. W. Richings, and D. P. Tew, *J. Chem. Phys.* **129**, 201103 (2008).

[62] D. P. Tew, W. Klopper, C. Neiss, and C. Hättig, *Phys. Chem. Chem. Phys.* **9**, 1921 (2007).

[63] J. Zhang and E. F. Valeev, *J. Chem. Theory Comput.* **8**, 3175 (2012).

[64] W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

[65] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi, *Chem. Phys. Lett.* **313**, 701 (1999).

[66] J. Friedrich, M. Hanrath, and M. Dolg, *J. Chem. Phys.* **126**, 154110 (2007).

[67] K. Kristensen, M. Ziółkowski, B. Jansík, T. Kjærgaard, and P. Jørgensen, *J. Chem. Theory Comput.* **7**, 1677 (2011).

[68] S. Sæbø and P. Pulay, *Chem. Phys. Lett.* **113**, 13 (1985).

[69] P. Pulay, *Chem. Phys. Lett.* **100**, 151 (1983).

[70] J. Almlöf, *Chem. Phys. Lett.* **181**, 319 (1991).

[71] G. E. Scuseria and P. Y. Ayala, *J. Chem. Phys.* **111**, 8330 (1999).

[72] A. K. Wilson and J. Almlöf, *Theor. Chim. Acta* **95**, 49 (1997).

[73] C. Hampel and H.-J. Werner, *J. Chem. Phys.* **104**, 6286 (1996).

[74] M. Schütz and H.-J. Werner, *J. Chem. Phys.* **114**, 661 (2001).

[75] C. Edmiston and M. Krauss, *J. Chem. Phys.* **42**, 1119 (1965).

[76] F. Neese, A. Hansen, and D. G. Liakos, *J. Chem. Phys.* **131**, 064103 (2009).

[77] P. Pinski, C. Riplinger, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **143**, 034108 (2015).

[78] C. Riplinger, P. Pinski, U. Becker, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **144**, 024109 (2016).

[79] Y. Guo, K. Sivalingam, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **144**, 094111 (2016).

[80] F. Pavošević, P. Pinski, C. Riplinger, F. Neese, and E. F. Valeev, *J. Chem. Phys.* **144**, 144109 (2016).

[81] H.-J. Werner, G. Knizia, C. Krause, M. Schwilk, and M. Dornbach, *J. Chem. Theory Comput.* **11**, 484 (2015).

[82] Q. Ma and H.-J. Werner, *J. Chem. Theory Comput.* **11**, 5291 (2015).

[83] F. Pavošević, F. Neese, and E. F. Valeev, *J. Chem. Phys.* **141**, 054106 (2014).

[84] G. Schmitz, C. Hättig, and D. P. Tew, *Phys. Chem. Chem. Phys.* **16**, 22167 (2014).

[85] M. E. Harding, T. Metzroth, J. Gauss, and A. A. Auer, *J. Chem. Theory Comput.* **4**, 64 (2008).

[86] A. P. Rendell, T. J. Lee, and R. Lindh, *Chem. Phys. Lett.* **194**, 84 (1992).

[87] G. E. Scuseria, C. L. Janssen, and H. F. Schaefer III, *J. Chem. Phys.* **89**, 7382 (1988).

[88] M. Valiev, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, H. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. Windus, and W. de Jong, *Comput. Phys. Commun.* **181**, 1477 (2010).

[89] R. Kobayashi, *Chem. Phys. Lett.* **265**, 1 (1997).

[90] V. M. Anisimov, G. H. Bauer, K. Chadalavada, R. M. Olson, J. W. Glenski, W. T. C. Kramer, E. Aprà, and K. Kowalski, *J. Chem. Theory Comput.* **10**, 4307 (2014).

[91] S. Hirata, *J. Phys. Chem. A* **107**, 9887 (2003).

[92] A. A. Auer, G. Baumgartner, D. E. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Krishnamoorthy, S. Krishnan, C.-C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov, *Mol. Phys.* **104**, 211 (2006).

[93] K. Kowalski, S. Krishnamoorthy, O. Villa, J. R. Hammond, and N. Govind, *J. Chem. Phys.* **132**, 154103 (2010).

[94] K. Kowalski, S. Krishnamoorthy, R. M. Olson, V. Tipparaju, and E. Aprà, Scalable implementations of accurate excited-state coupled cluster theories: Application of high-level methods to porphyrin-based systems, in *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–10, 2011.

[95] H.-S. Hu, K. Bhaskaran-Nair, E. Aprà, N. Govind, and K. Kowalski, *J. Phys. Chem. A* **118**, 9087 (2014).

[96] T. Janowski, A. R. Ford, and P. Pulay, *J. Chem. Theory Comput.* **3**, 1368 (2007).

[97] T. Janowski and P. Pulay, *J. Chem. Theory Comput.* **4**, 1585 (2008).

[98] C. Hampel, K. A. Peterson, and H.-J. Werner, *Chem. Phys. Lett.* **190**, 1 (1992).

[99] P. Pulay, S. Saebø, and W. Meyer, *J. Chem. Phys.* **81**, 1901 (1984).

[100] A. R. Ford, T. Janowski, and P. Pulay, *J. Comput. Chem.* **28**, 1215 (2007).

[101] R. M. Olson, J. L. Bentz, R. A. Kendall, M. W. Schmidt, and M. S. Gordon, *J. Chem. Theory Comput.* **3**, 1312 (2007).

[102] P. Piecuch, S. A. Kucharski, K. Kowalski, and M. Musiał, *Comput. Phys. Commun.* **149**, 71 (2002).

[103] A. Asadchev and M. S. Gordon, *J. Chem. Theory Comput.* **9**, 3385 (2013).

[104] B. A. Sanders, R. Bartlett, E. Deumens, V. Lotrich, and M. Ponton, A block-oriented language and runtime system for tensor algebra with very large arrays, in *2010 ACM/IEEE Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pp. 1–11, IEEE, 2010.

[105] E. Deumens, V. F. Lotrich, A. S. Perera, R. J. Bartlett, N. Jindal, and B. A. Sanders, Chapter 8 The super instruction architecture: A framework for high-productivity parallel implementation of coupled-cluster methods on petascale computers, in *Annu. Rep. Comput. Chem.*, volume 7, pp. 179–191, Elsevier, 2011.

[106] V. Lotrich, N. Flocke, M. Ponton, A. D. Yau, A. Perera, E. Deumens, and R. J. Bartlett, *J. Chem. Phys.* **128**, 194104 (2008).

[107] T. Kuś, V. F. Lotrich, and R. J. Bartlett, *J. Chem. Phys.* **130**, 124122 (2009).

[108] E. Solomonik, D. Matthews, J. Hammond, and J. Demmel, Cyclops Tensor Framework: Reducing communication and eliminating load imbalance in massively parallel contractions, in *2013 IEEE 27th Int. Symp. Parallel Distrib. Process.*, pp. 813–824, IEEE, 2013.

[109] E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel, *J. Parallel Distrib. Comput.* **74**, 3176 (2014).

[110] Y. Shao, Z. Gan, E. Epifanovsky, A. T. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Ku, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. W. III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen,

S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. D. Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. Hanson-Heine, P. H. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. D. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, E. Neuscamman, C. M. Oana, R. Olivares-Amaya, D. P. ONeill, J. A. Parkhill, T. M. Perrine, R. Peverati, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stck, Y.-C. Su, A. J. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, J. Yang, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. Chan, D. M. Chipman, C. J. Cramer, W. A. G. III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. S. III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. V. Voorhis, J. M. Herbert,

A. I. Krylov, P. M. Gill, and M. Head-Gordon, *Mol. Phys.* **113**, 184 (2015).

[111] E. Solomonik and J. Demmel, *Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms*, pp. 90–109, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[112] R. A. Van De Geijn and J. Watts, *Concurr. Pract. Exp.* **9**, 255 (1997).

[113] E. F. Valeev, C. Peng, C. A. Lewis, and J. A. Calvin, The Massively Parallel Quantum Chemistry Program (MPQC), 2016.

[114] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2**, 556 (2012).

[115] F. Neese, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2**, 73 (2012).

[116] R. J. Harrison, G. Beylkin, F. A. Bischoff, J. A. Calvin, G. I. Fann, J. Fosso-Tande, D. Galindo, J. R. Hammond, R. Hartman-Baker, J. C. Hill, J. Jia, J. S. Kottmann, M.-J. Y. Ou, J. Pei, L. E. Ratcliff, M. G. Reuter, A. C. Richie-Halford, N. A. Romero, H. Sekino, W. A. Shelton, B. E. Sundahl, W. S. Thornton, E. F. Valeev, lvaro Vzquez-Mayagoitia, N. Vence, T. Yanai, and Y. Yokoi, *SIAM J. Sci. Comput.* **38**, S123 (2016).

[117] J. Demmel, D. Eliahu, A. Fox, S. Kamil, B. Lipshitz, O. Schwartz, and O. Spillinger, Communication-optimal parallel recursive rectangular matrix multiplication, in *2013 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pp. 261–272, IEEE, 2013.

[118] J. A. Calvin, C. A. Lewis, and E. F. Valeev, Scalable task-based algorithm for multiplication of block-rank-sparse matrices, in *Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms*, IA3 '15, pp. 4:1–4:8, New York, NY, USA, 2015, ACM.

[119] R. J. Bartlett and M. Musiał, *Rev. Mod. Phys.* **79**, 291 (2007).

[120] T. D. Crawford and H. F. Schaefer, *An introduction to coupled cluster theory for computational chemists*, volume 14, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2000.

[121] I. Shavitt and R. J. Bartlett, *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*, Cambridge University Press, 2009.

[122] R. T. Pack, *J. Chem. Phys.* **45**, 556 (1966).

[123] S. Kedžuch, M. Milko, and J. Noga, *Int. J. Quantum Chem.* **105**, 929 (2005).

[124] H. J. Werner, T. B. Adler, and F. R. Manby, *J. Chem. Phys.* **126**, 164102 (2007).

[125] C. A. Lewis, J. A. Calvin, and E. F. Valeev, *J. Chem. Theory Comput.* **12**, 5868 (2016).

[126] E. F. Valeev, A library for the evaluation of molecular integrals of many-body operators over Gaussian functions, http://libint.valeyev.net/, 2014.

[127] G. E. Scuseria, T. J. Lee, and H. F. Schaefer, *Chem. Phys. Lett.* **130**, 236 (1986).

[128] J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett, *J. Chem. Phys.* **94**, 4334 (1991).

[129] a. E. DePrince and C. D. Sherrill, *J. Chem. Theory Comput.* **9**, 2687 (2013).

[130] H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker, *Chem. Phys. Lett.* **228**, 233 (1994).

[131] K. A. Peterson, T. B. Adler, and H.-J. Werner, *J. Chem. Phys.* **128**, 084102 (2008).

[132] K. E. Yousaf and K. A. Peterson, *J. Chem. Phys.* **129**, 184108 (2008).

[133] M. Klaus and P. Hobza, *Chem. Rev.* **100**, 143 (2000).

[134] M. Swart, T. van der Wijst, C. Fonseca Guerra, and F. M. Bickelhaupt, *J. Mol. Model.* **13**, 1245 (2007).

[135] J. Řezáč, K. E. Riley, and P. Hobza, *J. Chem. Theory Comput.* **7**, 3466 (2011).

[136] B. Brauer, M. K. Kesharwani, and J. M. L. Martin, *J. Chem. Theory Comput.* **10**, 3791 (2014).

[137] J. G. Hill, K. A. Peterson, G. Knizia, and H.-J. Werner, *J. Chem. Phys.* **131**, 194105 (2009).

[138] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, *J. Chem. Phys.* **121**, 11587 (2004).

[139] B. Brauer, M. K. Kesharwani, S. Kozuch, and J. M. L. Martin, *Phys. Chem. Chem. Phys.* **18**, 20905 (2016).

[140] J. Čížek, *J. Chem. Phys.* **45**, 4256 (1966).

[141] J. Čížek and J. Paldus, *Int. J. Quantum Chem.* **5**, 359 (1971).

[142] M. E. Harding, J. Vázquez, B. Ruscic, A. K. Wilson, J. Gauss, and J. F. Stanton, *J. Chem. Phys.* **128**, 114111 (2008).

[143] G. E. Scuseria and T. J. Lee, *J. Chem. Phys.* **93**, 5851 (1990).

[144] J. F. Stanton, *Chem. Phys. Lett.* (1997).

[145] Y. J. Bomble, J. F. Stanton, M. Kállay, and J. Gauss, *J. Chem. Phys.* **123**, 054101 (2005).

[146] M. Musiał and R. J. Bartlett, *J. Chem. Phys.* **122**, 224102 (2005).

[147] D. Bokhan, S. Bernadotte, and S. Ten-No, *Chem. Phys. Lett.* **469**, 214 (2009).

[148] F. Pavošević, C. Peng, P. Pinski, C. Riplinger, F. Neese, and E. F. Valeev, *J. Chem. Phys.* **146**, 174108 (2017).

[149] G. Schmitz and C. Hättig, *J. Chem. Phys.* **145**, 234107 (2016).

[150] C. Peng, J. A. Calvin, F. Pavošević, J. Zhang, and E. F. Valeev, *J. Phys. Chem. A* **120**, 10231 (2016).

[151] A. P. Rendell, T. J. Lee, A. Komornicki, and S. Wilson, *Theor. Chim. Acta* **84**, 271 (1993).

[152] E. Aprà, A. P. Rendell, R. J. Harrison, V. Tipparaju, W. a. DeJong, and S. S. Xantheas, *Proc. Conf. High Perform. Comput. Networking, Storage Anal. - SC '09* , 1 (2009).

[153] W. Ma, S. Krishnamoorthy, O. Villa, and K. Kowalski, *J. Chem. Theory Comput.* **7**, 1316 (2011).

[154] E. Apra, M. Klemm, and K. Kowalski, Efficient Implementation of Many-Body Quantum Chemical Methods on the Intel ®Xeon Phi Coprocessor, in *Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pp. 674–684, 2014.

[155] J. L. Bentz, R. M. Olson, M. S. Gordon, M. W. Schmidt, and R. A. Kendall, *Comput. Phys. Commun.* **176**, 589 (2007).

[156] A. P. Rendell, T. J. Lee, and A. Komornicki, *Chem. Phys. Lett.* **178**, 462 (1991).

[157] J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).

[158] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

[159] O. Vahtras, J. Almlöf, and M. W. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).

[160] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon, *Proc. Natl. Acad. Sci. U. S. A.* **102**, 6692 (2005).

[161] H.-J. Werner, F. R. Manby, and P. J. Knowles, *J. Chem. Phys.* **118**, 8149 (2003).

[162] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.* **294**, 143 (1998).

[163] M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).

[164] A. P. Rendell and T. J. Lee, *J. Chem. Phys.* **101**, 400 (1994).

[165] E. Epifanovsky, D. Zuev, X. Feng, K. Khistyaev, Y. Shao, and A. I. Krylov, *J. Chem. Phys.* **139**, 134105 (2013).

[166] D. E. Bernholdt and R. J. Harrison, *Chem. Phys. Lett.* **250**, 477 (1996).

[167] D. E. Bernholdt, *Parallel Comput.* **26**, 945 (2000).

[168] M. Katouda and S. Nagase, *Int. J. Quantum Chem.* **109**, 2121 (2009).

[169] M. Katouda, A. Naruse, Y. Hirano, and T. Nakajima, *J. Comput. Chem.* **37**, 2623 (2016).

[170] C. Edmiston and M. Krauss, *J. Chem. Phys.* **45**, 1833 (1966).

[171] W. Meyer, *Int. J. Quantum Chem.* **5**, 341 (1971).

[172] F. Neese, F. Wennmohs, and A. Hansen, *J. Chem. Phys.* **130** (2009).

[173] A. Dreuw and M. Head-Gordon, *Chem. Rev.* **105**, 4009 (2005).

[174] D. Jacquemin, V. Wathelet, E. A. Perpete, and C. Adamo, *J. Chem. Theory Comput.* **5**, 2420 (2009).

[175] H. J. Werner and P. J. Knowles, *J. Chem. Phys.* **89**, 5803 (1988).

[176] P. G. Szalay, T. Müller, G. Gidofalvi, H. Lischka, and R. Shepard, *Chem. Rev.* **112**, 108 (2012).

[177] G. K.-L. Chan and S. Sharma, *Annu. Rev. Phys. Chem.* **62**, 465 (2011).

[178] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).

[179] H. J. Monkhorst, *Int. J. Quantum Chem.* **12**, 421 (1977).

[180] H. Nakatsuji, *Chem. Phys.* **75**, 425 (1983).

[181] H. Nakatsuji, *Chem. Phys. Lett.* **67**, 329 (1979).

[182] H. Sekino and R. J. Bartlett, *Int. J. Quantum Chem.* **26**, 255 (1984).

[183] J. F. Stanton and R. J. Bartlett, *J. Chem. Phys.* **98**, 7029 (1993).

[184] T. Korona and H. J. Werner, *J. Chem. Phys.* **118**, 3006 (2003).

[185] T. Daniel Crawford and R. A. King, *Chem. Phys. Lett.* **366**, 611 (2002).

[186] N. J. Russ and T. D. Crawford, *Chem. Phys. Lett.* **400**, 104 (2004).

[187] C. Riplinger and F. Neese, *J. Chem. Phys.* **138**, 034106 (2013).

[188] C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, *J. Chem. Phys.* **139**, 134101 (2013).

[189] P. Pinski, C. Riplinger, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **143**, 034108 (2015).

[190] C. Riplinger, P. Pinski, U. Becker, E. F. Valeev, and F. Neese, *J. Chem. Phys.* **144**, 024109 (2016).

[191] Q. Ma and H.-J. Werner, *J. Chem. Theory Comput.* **14**, acs.jctc.7b01141 (2017).

[192] B. Helmich and C. Hättig, *J. Chem. Phys.* **139**, 084114 (2013).

[193] B. Helmich and C. Hättig, *J. Chem. Phys.* **135**, 214106 (2011).

[194] B. Helmich and C. Hättig, *Comput. Theor. Chem.* **1040-1041**, 35 (2014).

[195] A. K. Dutta, F. Neese, and R. Izsák, *J. Chem. Phys.* **145**, 034102 (2016).

[196] A. K. Dutta, M. Nooijen, F. Neese, and R. Izsák, *J. Chem. Theory Comput.* **14**, 72 (2018).

[197] D. Kats, T. Korona, and M. Schütz, *J. Chem. Phys.* **125**, 104106 (2006).

[198] J. A. Calvin and E. F. Valeev, TiledArray: A general-purpose scalable block-sparse tensor framework, 2016.

[199] J. Yang, Y. Kurashige, F. R. Manby, and G. K. Chan, *J. Chem. Phys.* **134**, 044123 (2011).

[200] C. Krause and H.-J. Werner, *Phys. Chem. Chem. Phys.* **14**, 7591 (2012).

[201] H. R. McAlexander and T. D. Crawford, *J. Chem. Theory Comput.* **12**, 209 (2016).

[202] J. A. Calvin and E. F. Valeev, p. 9 (2015).

[203] J. A. Calvin, C. A. Lewis, and E. F. Valeev, *Proc. 5th Work. Irregul. Appl. Archit. Algorithms - IA3 '15* , 1 (2015).

[204] J. M. Foster and S. F. Boys, *Rev. Mod. Phys.* **32**, 300 (1960).

[205] S. F. Boys, *Rev. Mod. Phys.* **32**, 296 (1960).

[206] T. H. Dunning, *J. Chem. Phys.* **90**, 1007 (1989).

[207] R. A. Kendall, T. H. Dunning, and R. J. Harrison, *J. Chem. Phys.* **96**, 6796 (1992).

[208] F. Weigend, A. Köhn, and C. Hättig, *J. Chem. Phys.* **116**, 3175 (2002).