The Unreasonable Usefulness of Approximation by Linear Combination

Cannada A. Lewis

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

Daniel T. Crawford

John R. Morris

Diego Troya

April 13, 2018

Blacksburg, Virginia

Keywords: Electronic Structure, Tensor Factorization, Linear Scaling

The Unreasonable Usefulness of Approximation by Linear Combination

Cannada A. Lewis

(ABSTRACT)

Through the exploitation of data-sparsity —a catch all term for savings gained from a variety of approximations— it is possible to reduce the computational cost of accurate electronic structure calculations to linear. Meaning, that the total time to solution for the calculation grows at the same rate as the number of particles that are correlated. Multiple techniques for exploiting data-sparsity are discussed, with a focus on those that can be systematically improved by tightening numerical parameters such that as the parameter approaches zero the approximation becomes exact. These techniques are first applied to Hartree-Fock theory and then we attempt to design a linear scaling massively parallel electron correlation strategy based on second order perturbation theory.

The Unreasonable Usefulness of Approximation by Linear Combination

Cannada A. Lewis

(GENERAL AUDIENCE ABSTRACT)

The field of Quantum Chemistry is highly dependent on a vast hierarchy of approximations; all carefully balanced, so as to allow for fast calculation of electronic energies and properties to an accuracy suitable for quantitative predictions. Formally, computing these energies should have a cost that increases exponentially with the number of particles in the system, but the use of approximations based on local behavior, or nearness, of the particles reduces this scaling to low order polynomials while maintaining an acceptable amount of accuracy. In this work, we introduce several new approximations that throw away information in a specific fashion that takes advantage of the fact that the interactions between particles decays in magnitude with the distance between them (although sometimes very slowly) and also exploits the smoothness of those interactions, by factorizing their numerical representation into a linear combination of simpler items. These factorizations, while technical in nature, have benefits that are hard to obtain by merely ignoring interactions between distant particles. Through the development of new factorizations and a careful neglect of interactions between distant particles, we hope to be able to compute properties of molecules in such a way that accuracy is maintained, but that the cost of the calculations only grows at the same rate as the number of particles. It seems that very recently, circa 2015, that this goal may actually soon become a reality, potentially revolutionizing the ability of quantum chemistry to make quantitative predictions for properties of large molecules.

For Marjorie and Michael the two most important people in my life.

Acknowledgments

- I would first like to acknowledge my advisor Edward F. Valeev for being of great assistance in both the generation of the ideas in this document as well being a great help in overcoming the inevitable obstacles faced when doing research. After six years (a not insignificant portion of which was probably spent trying to find ways to avoid having to talk to me) Edward deserves some peace and quiet.

- Next, I would like to thank my committee professors: Daniel Crawford, John Morris, and Diego Troya for helping to guide my studies to their completion.

- The Valeev and Crawford groups were very helpful for their numerous discussions in particular: Justus Calvin, Fabijan Pavošivić, Chong Peng, Andrey Asadchev, Harley McAlexander, Ashutosh Kumar, and Xiao Wang.

- My parents John and Helen for their support in my 20+ years of being in school.

- My wife Marjorie for editing everything that I have written in graduate school and for translating what I write into English with punctuation and some semblance of structure.

- My son Michael for not disowning me due to my absence in the final few months of graduate school.

- Finally, to anyone and everyone who had to deal with the fact that I spent the majority of my time and their time talking about anything and everything. You were all very polite and I am sure I was very distracting.

Attribution

Chapters 2 has the coauthor Justus Calvin and chapters 2, 3 and 4 have the coauthor Edward Valeev. Justus Calvin's main contribution to Chapter 2 was the development of the tensor algebra library TiledArray and his support in the use of this library. All calculations, figures, and the majority of the text for all three chapters was contributed by myself. Edward's main contribution was in the idea generation for the topic of all three chapters and for editing, for both accuracy and clarity, the introduction and background sections of chapters 2 and 4.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

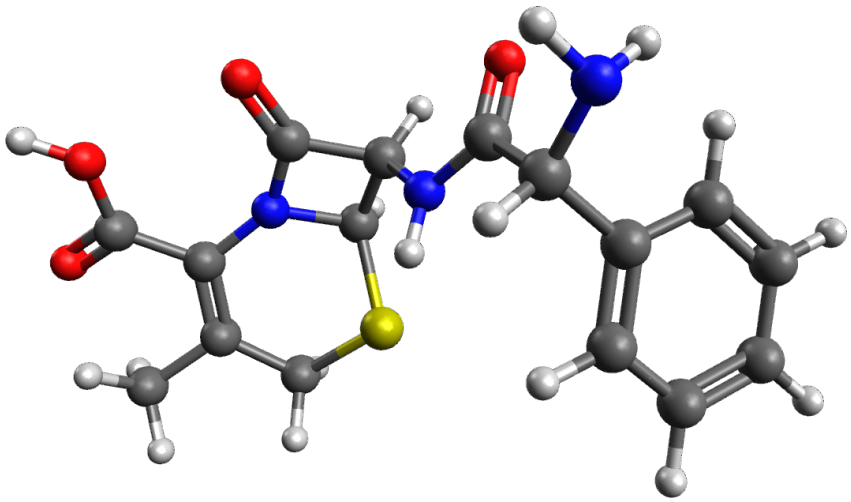

Figure 1.1: Cephalexin, an antibiotic that your author was taking while writing this section.

Quantum chemistry and specifically electronic structure, or the study of the quantum nature of electrons in molecules, can provide deep insights into the inner workings of chemical processes in fields ranging from astro-biology to drug design. A particularly compelling example for your author (who happened to have a slight infection while working on this section) is the mechanism of the antibiotic Cephalexin[1] shown in Figure 1.1. Cephalexin, a β-lactam antibiotic, is a bactericidal molecule that acts by inhibiting the synthesis of bacterial cell walls. Unfortunately, antibiotic resistance is starting to become an issue for these and many other types of antibiotics,[2] but drug design and discovery are costly and difficult processes. So many theorists have proposed using quantum chemistry to reduce the cost inherent in the design of drug molecules.[3] Using quantum chemistry we can understand how drug molecules interact with their targets, how the bacteria are countering that mechanism, and finally, how to design new antibiotics that circumvent the bacteria's resistance. It is like a deadly serious version of spy versus spy.

To understand how the study of electronic structure can be used to help elucidate the detailed working of drug interactions, it is helpful to start at the very beginning, the Schrödinger equation, followed by the modern techniques of how it is approximated to study the behavior of electrons in molecules. The remainder of this chapter is organized as follows, first we will discuss how the Schroödinger equation can be applied to simple molecules, then use those ideas to discuss what is typically thought of as the most basic quantum mechanical approximation to the wavefunction, the Hartree-Fock approximation. We will detail an example of how Hartree-Fock fails and then discuss ways in which to improve upon our initial approximation with an emphasis on Møller-Plesset perturbation theory. Then we will discuss one tool of the trade that is used to make the implementation of these methods more computationally friendly. Finally, we will make an unusual analogy between electronic structure methods and computer graphics that should help to put the types of approximations made in a context that non-experts might find more familiar.

1.1 The Electronic Schrödinger Equation

Before beginning this section I would like to point the reader to the ever helpful source *Modern Quantum Chemistry* by Attila Szabo and Neil S. Ostlund.[4] This book was not only indispensable (multiple copies had to be replaced after falling apart from use), but provides most of the background for this section. To avoid redundancy, and also prevent what is almost surely a worse explanation, at points in this section I will refer the reader to the appropriate section of this book.

The non-relativistic time-independent Schrödinger equation written in the compact form

$$\mathcal{H}\Psi = \mathcal{E}\Psi, \tag{1.1}$$

shows that the application of the Hamiltonian operator, $\mathcal{H}$, to the wavefunction, Ψ, returns the energy, $\mathcal{E}$. The wavefunction is a function of $3N$ variables, where N is the number of particles in the system

$$\Psi \equiv \Psi\left(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m, \mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_w\right), \tag{1.2}$$

where $\mathbf{r}_1$ refers to the coordinates of electron 1 and $\mathbf{R}_1$ the coordinates of nucleus 1. For the remainder of this chapter the use of bold face will denote a vector so that $\mathbf{r}_1$ defines the vector $\{r_1(1), r_1(2), r_1(3)\}$. The Hamiltonian can be written out as

$$\begin{aligned}\mathcal{H} \equiv &-\sum_i \frac{1}{2}\nabla_i^2 - \sum_A \frac{1}{2M_A}\nabla_A^2 - \sum_i \sum_A \frac{Z_A}{r_{iA}} \\ &+ \sum_i \sum_{j>i} \frac{1}{r_{ij}} + \sum_A \sum_{B>A} \frac{Z_A Z_B}{r_{AB}},\end{aligned} \tag{1.3}$$

where M_A is the ratio of the mass of nucleus A, to the mass of the electron, Z_A is the atomic number of nucleus A, ∇_i^2 is the kinetic energy operator for electron i, similarly ∇_A^2 is the kinetic energy operator of nucleus A and r_{ij}, r_{iA}, and r_{AB} are the distances between electron i and electron j, electron i and nucleus A, and nucleus A and nucleus B respectfully. With Equation (1.3), which makes use of atomic units to simplify the expression, one can then attempt to solve for Ψ, but no known analytical solutions exist for the general case.

To simplify the search for an approximate solution to Ψ, a common approach is to separate or factorize (a term we will see again) Ψ into an electronic part that depends only on fixed nucleus coordinates and a nuclear part that depends only on the average position of the electrons

$$\Psi \approx \Phi_{\text{elec}}(\mathbf{r}_1, \ldots \mathbf{r}_m, \{\mathbf{R}_A\})\Phi_{\text{nucl}}(\mathbf{R}_1, \ldots \mathbf{R}_w, \{\mathbf{r}_i\}), \tag{1.4}$$

Table 1.1: The symbols used to represent different types of functions in the text.

Ψ	The exact electronic and nuclear wavefunction
Φ	The exact electronic wavefunction
$\tilde{\Phi}$	An approximate electronic wavefunction
$\bar{\Phi}$	A single Slater determinant
χ_k	Molecular orbital k
ψ_k	The spatial part of molecular orbital k
ϕ_k	The kth function in a basis set
ϕ_μ	The μth function in an atomic orbital basis set

Table 1.2: The notation used for various types of functions, the meaning and origin of the indices will be explained in the text.

$i, j, k, \ldots$	Occupied molecular orbitals
$a, b, c, \ldots$	Unoccupied molecular orbitals
$X, Y, \ldots$	Auxiliary basis indices
$\mu, \nu, \rho, \ldots$	Atomic orbital indices

The approximation of Equation (1.4) is typically called the Born-Oppenheimer approximation, although perhaps it should really be called the clamped nucleus approximation.[5] For the remainder of this work we will focus only on the electronic portion, Φ_{elec}, and the electronic parts of the Hamiltonian:

$$\mathcal{H}_{\text{elec}} \equiv -\sum_i \frac{1}{2}\nabla_i^2 - \sum_i \sum_A \frac{Z_A}{r_{iA}} + \sum_i \sum_{j>i} \frac{1}{r_{ij}}, \tag{1.5}$$

where the positions of the atoms centers are merely inputs to Φ_{elec} and the quantum mechanical behavior of the electrons can be solved for independently of the nuclear wavefunction.

1.2 Hartree-Fock and the Factorization of Φ_{elec}

To start with we will define some notation used for the remainder of the introduction in Table 1.1 and Table 1.2. After application of the clamped nucleus approximation has been

applied our electronic wavefunction can be thought of as a function of $3M$ parameters where M is the number of electrons

$$\Phi_{\text{elec}} \equiv \Phi_{\text{elec}}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m). \tag{1.6}$$

Historically Φ was approximated via trial functions (here labeled as $\tilde{\Phi}$) using a Rayleigh quotient (from this point forward we will drop the electronic designation):

$$\mathcal{E} = \frac{\left\langle \tilde{\Phi} \middle| \mathcal{H} \middle| \tilde{\Phi} \right\rangle}{\left\langle \tilde{\Phi} \middle| \tilde{\Phi} \right\rangle}, \tag{1.7}$$

where $\langle \, | \, \rangle$ is Dirac's bra-ket notation. Equation (1.7) allows the variational optimization of any parameters that enter into $\tilde{\Phi}$ with the guarantee that the computed energy will always be an upper bound to the exact energy. For the smallest of molecules like H_2 and helium it is reasonable to create trial wavefunctions that depend explicitly on the r_1, r_2, and r_{12} distances, where the first two are the distances of electron 1 and electron 2 from the nucleus and r_{12} is the distance from electron 1 to electron 2. One of the most successful forms of trial function are those that depend explicitly on the distance between the electrons. A simple example is the Hylleraas trial function

$$\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2) = e^{-\zeta r_1} e^{-\zeta r_2} (1 + c r_{12}), \tag{1.8}$$

which can be variationally optimized to give an energy of -2.8913 Hartrees, for the helium atom, within 0.5% of the exact energy.[6, 7] While trial functions that explicitly account for the interaction between particles can be used to achieve very high accuracy results, they have a serious drawback: as the number of particles grows the number of interactions grows extremely rapidly such that their computation becomes difficult.

Instead of dealing explicitly with particle interactions, we can attempt to build a trial function where Φ is approximated as products of one particle functions

$$\tilde{\Phi} = \psi_1(\mathbf{r}_1)\psi_2(\mathbf{r}_2)\dots\psi_m(\mathbf{r}_m), \tag{1.9}$$

which is known as a Hartree product. The Hartree product can be thought of as a factorization of a high dimensional function into a product of one dimensional functions. If a Hartree product or a small sum of Hartree products is a good approximation to Φ then for this work we will say that Φ is factorizable, where factorizable means that some function of dimension N can be represented as a small linear combination of products of functions with dimension less than N. Perhaps the simplest test case for the type of factorization in Equation (1.9) is the ground state of the helium atom. We can test one of the simplest variationally optimizable Hartree products:

$$\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\zeta}{\pi} e^{-\zeta(r_1+r_2)}, \tag{1.10}$$

which when $\zeta = \frac{27}{16}$ gives an energy of -2.84766 Hartrees.[7] While this result is much worse that $\tilde{\Phi}$ from Equation (1.8) it only has a single variational parameter. To treat more general cases than single states of the helium atom we will need to account for the fact that electrons are fermions and thus $\tilde{\Phi}$ must be anti-symmetric with respect to the swapping of any two electrons. The most natural way to anti-symmetrize a Hartree product is to adapt it into a

Slater determinant:

$$\bar{\Phi}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m) = \frac{1}{\sqrt{M!}} \begin{vmatrix} \chi_1(\mathbf{r}_1) & \chi_2(\mathbf{r}_1) & \ldots & \chi_m(\mathbf{r}_1) \\ \chi_1(\mathbf{r}_2) & \chi_2(\mathbf{r}_2) & \ldots & \chi_m(\mathbf{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{r}_m) & \chi_2(\mathbf{r}_m) & \ldots & \chi_m(\mathbf{r}_m) \end{vmatrix}, \tag{1.11}$$

where $\bar{\Phi}$ is used to represent a single Slater determinant, it is possible to have $\tilde{\Phi} = \bar{\Phi}$ if we approximate Φ as a single Slater determinant. A Slater determinant places M electrons into M spin orbitals $\{\chi_i\}$, also called molecular orbitals, and its form is basically a linear combination of Hartree products such that the swapping of any two electrons changes the sign of $\bar{\Phi}$. Next we must ask, what is the form of these χ functions? For calculations on light atoms, where the effects relativity are small χ can be factorized (there is that word again) into a spatial part and a spin part such that $\chi_i(\mathbf{r}_i) = \psi_i(\mathbf{r}_i)\alpha(\omega_i)$ where the $\alpha(\omega_i)$ designates the spin of particle i. Then the question becomes what is the form of the $\{\psi_i\}$? It turns out that once again for general systems there is no analytical form for these ψ and we must instead approximate them. This calls for the introduction of some basis —a set of functions— in which the set of ψ can be approximated. Given a set of basis functions $\{\phi_j\}$ we can approximate each ψ_i as a linear combination of the ϕ functions via:

$$\psi_i(\mathbf{r}_1) \approx \sum_k C_{ki}\phi_k(\mathbf{r}_1), \tag{1.12}$$

where the C_{ki} parameters are to be determined via optimization. The most common form of basis set for typical electronic structure calculations are those based on the solutions to the hydrogen atom, further discussion of basis sets is beyond the scope of this introduction, but insight into how they are designed can be found in reference [8]. The use of these atomic like basis sets for the molecular orbitals is usually termed the linear combination of

atomic orbital approximation or LCAO. Once the $\{\phi_\mu\}$ —lower case Greek letters designate atomic orbital basis functions— have been chosen almost all wavefunction based electron structure methods determine the C_{ki} parameters via the Hartree-Fock method. For more background on Hartree-Fock and a derivation of the following close shell equations the reader is referred to chapters 2 and 3 of *Modern Quantum Chemistry*. After some amount of work it can be shown that best set of C_{ki} for each ψ_i is determined by taking the $\frac{M}{2}$ eigenvectors corresponding to the $\frac{M}{2}$ smallest eigenvalues of the following generalized eigenvalue equation:

$$\mathbf{FC} = \lambda \mathbf{SC}, \tag{1.13}$$

where $S_{\mu\nu} = \langle \phi_\mu \,|\, \phi_\nu \rangle$ is the overlap of the basis set and $\mathbf{F}$, in closed shell form where every orbital holds two electrons and the spin component of the $\{\chi_i\}$ has been integrated out, is given by:

$$F_{\mu\nu} = T_{\mu\nu} + V_{\mu\nu} + \sum_{\rho\sigma} \left(2(\mu\nu|\rho\sigma) - (\mu\rho|\nu\sigma)\right) D_{\rho\sigma}, \tag{1.14}$$

where $\mathbf{T}$ is the kinetic energy operator in the basis $\{\phi_\mu\}$, $\mathbf{V}$ is the nuclear attraction felt by the electrons and the sum over ρ and σ represents the repulsion felt by the electron from the average of all the other electrons in the system, $\mathbf{D}$ is the density given by:

$$D_{\mu\nu} = \sum_{i=1}^{M} C_{\mu i} C_{\nu i}, \tag{1.15}$$

and finally, $(\mu\nu|\rho\sigma)$ represents the integral

$$(\mu\nu|\rho\sigma) = \iint \phi_\mu^*(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1) r_{12}^{-1} \phi_\rho^*(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2) \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2. \tag{1.16}$$

The astute reader will realize that $\mathbf{F}$ depends on $\mathbf{D}$ and $\mathbf{D}$ depends on $\mathbf{F}$. This means in

practice these equations must be solved self consistently where a guess is made for **D** and then a new **F** is generated and that **F** is used to construct a new **D**. Only once **F** and **D** experience changes below some threshold do we consider the problem solved. For this reason sometimes Hartree-Fock is also called SCF, which stands for self consistent field method, a reference to how the equations are solved in practice.

It turns out that while Hartree-Fock is able to account for the vast majority of the total energy of a molecule, it is not without failures. Even when using extrapolation schemes that estimate the lowest possible Hartree-Fock energy we only get an energy of -2.86168344 Hartrees, for the helium atom, which is off by 1.4% from the exact value.[7] A poor result relative to the very simple trial function of Equation (1.8) given that the Hartree-Fock solution needs hundreds of variational parameters for this calculation while Equation (1.8) needs only 2. Clearly, approximating the wavefunction as a simple product of well defined orbitals prevents it from achieving the accuracy of even very simple trial functions. For example, even the trial function:

$$\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2) = N(e^{-\zeta_1 r_1} e^{-\zeta_2 r_2} + e^{-\zeta_2 r_1} e^{-\zeta_1 r_2}), \tag{1.17}$$

where N is a normalization factor is more accurate than the Hartree-Fock basis set limit for the helium atom. Figure 1.2 shows my personal favorite example of the failure of Hartree-Fock as well as the difficulty in choosing a proper basis set. The hydrogen anion is a bound system, but Hartree-Fock and even the exact —what exact means will be covered in the next section— solution within a small non-diffuse basis set does not show this. The selection of appropriate basis sets and their extrapolation to the complete basis set limit is beyond the scope of this introduction, but it is evident that the Hartree-Fock approximation is woefully inadequate for the prediction of chemical properties. But there is hope and Equation (1.17) suggests to us the form of a possible improvement, we can construct $\tilde{\Phi}$ not as a simple

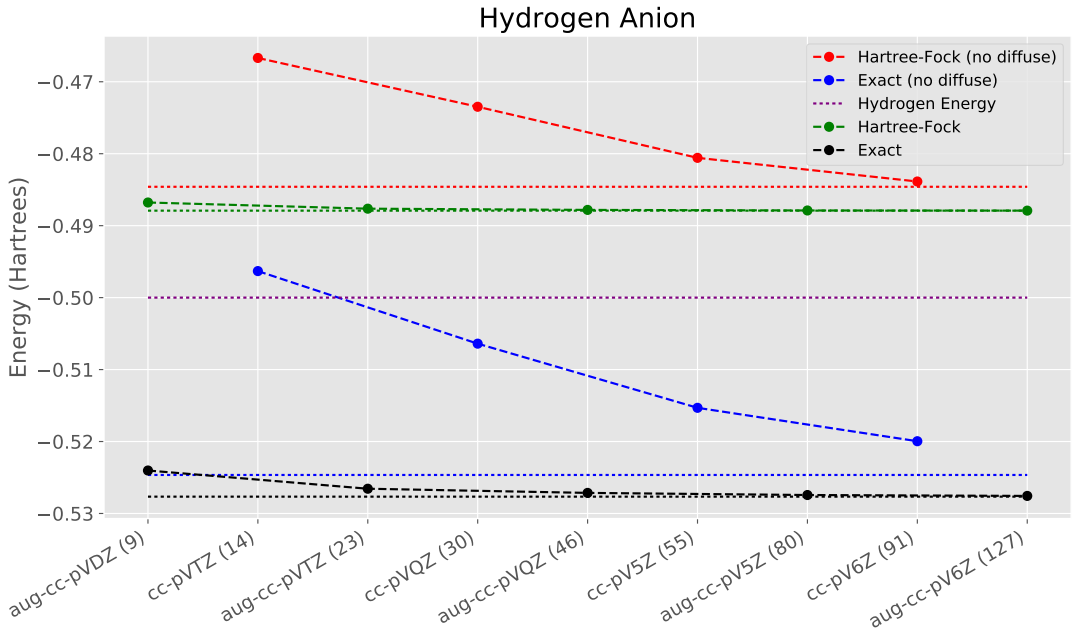

Figure 1.2: The energies for the hydrogen anion for Hartree-Fock and the Exact energy (exact solutions within the given basis set were computed with CCSD) in two different types of basis sets. The basis sets with diffuse functions (prefixed with aug-) show much better convergence to the basis set limit (the basis set limit is displayed via dashed lines with no markers, where the colors correspond to the limit for the matching method in the legend) as would be expected with an anion. Most importantly though is that Hartree-Fock fails to show that the hydrogen anion is lower in energy than the hydrogen atom itself.

product of one-electron functions, or as is actually the case a single Slater determinant ($\bar{\Phi}$), but instead include a linear combination of Slater determinants to approximate Φ. In the next section we will see how it is possible to construct a linear combination of Slater determinants that with the proper basis sets and extrapolation techniques (and infinite computer memory and time) will give the exact energy of a molecule within the clamped nucleus and non-relativistic approximations.

1.3 Post Hartree-Fock Methods

Based on the final trial function in the previous section (Equation (1.17)), we see a possible solution to our problems, Φ can be approximated as a linear combination of Slater determinants (hopefully the reader is beginning to notice a pattern here):

$$|\Phi\rangle = \sum_z C_z \left|\bar{\Phi}_z\right\rangle, \tag{1.18}$$

where each $\left|\bar{\Phi}\right\rangle$ represents a single Slater determinant. Given a complete basis of Slater determinants $|\Phi\rangle$ can be exactly represented and the energy of the system can be found by diagonalizing the following matrix,

$$H_{y,z} = \left\langle \bar{\Phi}_y \middle| \mathcal{H} \middle| \bar{\Phi}_z \right\rangle, \tag{1.19}$$

with $\mathbf{H}$'s most negative eigenvalue corresponding to the ground state energy and its corresponding eigenvector giving the values of the C_z coefficients from Equation (1.18). The above approach, called configuration interaction, only requires two things: the first is the ability to compute the integrals in Equation (1.20) the second, an approximation, is that some suitable basis set of Slater determinants is used for the expansion. If both integrals

and a good bases of Slater determinants are available then $|\Phi\rangle$ can be approximated to arbitrary accuracy. Unfortunately, both of these requirements are non-trivial to fulfill, integrals between Slater determinants are complicated, unless certain conditions are enforced and in practice we end up needing a fairly large basis of Slater determinants for high accuracy calculations.[9] Thankfully, for the study of most molecules, in their equilibrium geometries and in their ground electronic states, there is a systematic way to converge towards the exact solution while also simplifying the integrals between determinants.

When computing the Hartree-Fock wavefunction for a closed shell system, we calculate as many orbitals as their are basis functions (let us call N the number of basis functions) in the calculation, but we only end up using $M/2$ of those orbitals, where M is the number of electrons. Thus at the completion of the SCF procedure we have $V = N - \frac{M}{2}$ extra orbitals. These orbitals can be used to construct valid Slater determinants leading to a natural basis of Slater determinants for our expansion of $|\Phi\rangle$, where we can move each electron from one of our $\frac{M}{2}$ (occupied) orbitals to one of our V (virtual) orbitals to form a new Slater determinant. Leading to the following expansion of $|\Phi\rangle$:

$$|\Phi\rangle = C_0 \left|\bar{\Phi}_0\right\rangle + \sum_{ia} C_a^i \left|\bar{\Phi}_a^i\right\rangle + \sum_{\substack{i<j \\ a<b}} C_{ab}^{ij} \left|\bar{\Phi}_{ab}^{ij}\right\rangle + \ldots, \tag{1.20}$$

where $\left|\bar{\Phi}\right\rangle$ is the minimal energy Hartree-Fock Slater determinant and $\left|\bar{\Phi}_{ab}^{ij}\right\rangle$ is the determinant formed by removing electrons from orbitals i and j and placing them in orbitals a and b respectfully. The quality of this expansion will depend heavily on the quality of the basis set used for the Hartree-Fock calculation [1], but if all possible substitutions are made and the appropriate basis set is chosen then the exact $|\Phi\rangle$ can be approximated very accurately (at great cost) and the systems ground state energy can be extrapolated to what is essentially

[1] see Figure 1.2 for the exact solution without diffuse functions for an example where this Slater determinant basis fails

the exact value. Unfortunately, this approach leads to an astronomical increase in costs as the Hartree-Fock basis becomes larger, necessitating further approximation. Usually, this approximation takes the form of truncating the expansion at a certain rank of substitution (also called excitation) such that all terms larger than X are dropped from the sum in Equation (1.20) where X is usually D,T,Q,... Meaning that all terms including double, triple, quadruple, ... replacements are kept. Once the basis of determinants has been chosen we need to know how to compute matrix elements between determinants in this basis, thankfully this particular basis we have chosen naturally leads to simplified integrals such that $\left\langle \bar{\Phi}^{abc\ldots}_{ijk\ldots} \middle| \mathcal{H} \middle| \bar{\Phi}^{def\ldots}_{lmn\ldots} \right\rangle$ can be readily computed by Slater-Condon rules,[4] the details of which are widely available. Perhaps the largest benefit of this approach is that any two determinants that differer by more than two orbitals will not contribute to $\mathbf{H}$ due to the Slater-Condon rules, limiting the number of determinants that any specific determinant will interact with. Historically the way the coefficients were determined was via CI,[4] but eventually the coupled cluster (CC) approximation became dominate within the field. References [10] and [11] provide ample background on the coupled cluster approach.

1.3.1 Møller-Plesset Perturbation Theory

In addition to configuration interaction and the coupled cluster approximation there is one other technique commonly used for determining the expansion of $|\Phi\rangle$ in terms of Slater determinants, Møller-Plesset perturbation (MP) theory. Unlike CC and CI we will take the time to discuss MP theory since a reduced scaling approximation to the second order variant is the focus of Chapter 4. The main idea of perturbation theory is that we split our Hamiltonian $\mathcal{H}$ into two pieces such that $\mathcal{H} = \mathcal{H}_0 + \mathcal{V}$, where $\mathcal{H}_0$ is a part of the Hamiltonian that we have eigenvalues and eigenfunctions for and $\mathcal{V}$ is a perturbation. When $\mathcal{H}_0$ is a good approximation for $\mathcal{H}$ then the inclusion of $\mathcal{V}$ should only have a small effect that can

be approximated. The main choice in electronic structure theory for $\mathcal{H}_0$ and $\mathcal{V}$ is the one used in Møller-Plesset (MP) perturbation theory, named after Christian Møller and Milton Plesset.[12]

Rayleigh-Schrödinger perturbation theory, the theory on which MP is based, takes our separated Hamiltonian from the previous paragraph and adds an ordering parameter λ to $\mathcal{V}$:

$$\mathcal{H} = \mathcal{H}_0 + \lambda\mathcal{V}. \tag{1.21}$$

Then the exact energy and the exact wavefunction can be expanded in a Taylor series in λ,

$$\begin{aligned} \mathcal{E} &= E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \dots \\ \Phi &= \tilde{\Phi}^{(0)} + \lambda\tilde{\Phi}^{(1)} + \lambda^2\tilde{\Phi}^{(2)} + \dots, \end{aligned} \tag{1.22}$$

where $\tilde{\Phi}^{(0)}$ is the lowest energy eigenfunction of $\mathcal{H}_0$ and $E^{(0)}$ is the energy of $\tilde{\Phi}^{(0)}$. By applying the Hamiltonian in the form $\mathcal{H}_0 + \lambda\mathcal{V}$ and using Equation (1.22) we can group terms that have the same powers in lambda to obtain a series of equations:

$$\begin{aligned} \mathcal{H}_0 \left|\tilde{\Phi}^{(0)}\right\rangle &= E^{(0)} \left|\tilde{\Phi}^{(0)}\right\rangle \\ \mathcal{H}_0 \left|\tilde{\Phi}^{(1)}\right\rangle + \mathcal{V}\left|\tilde{\Phi}^{(0)}\right\rangle &= E^{(0)} \left|\tilde{\Phi}^{(1)}\right\rangle + E^{(1)} \left|\tilde{\Phi}^{(0)}\right\rangle \\ \mathcal{H}_0 \left|\tilde{\Phi}^{(2)}\right\rangle + \mathcal{V}\left|\tilde{\Phi}^{(1)}\right\rangle &= E^{(0)} \left|\tilde{\Phi}^{(2)}\right\rangle + E^{(1)} \left|\tilde{\Phi}^{(1)}\right\rangle + E^{(2)} \left|\tilde{\Phi}^{(0)}\right\rangle \end{aligned} \tag{1.23}$$

By choosing $\tilde{\Phi}^{(n)}$ $(n \neq 0)$ to be orthogonal to $\tilde{\Phi}^{(0)}$ and projecting on the left by $\left\langle\tilde{\Phi}^{(0)}\right|$ in

Equation (1.23) we can solve for the corrections to the energies:

$$\begin{aligned} E^{(0)} &= \left\langle \tilde{\Phi}^{(0)} \middle| \mathcal{H}_0 \middle| \tilde{\Phi}^{(0)} \right\rangle \\ E^{(1)} &= \left\langle \tilde{\Phi}^{(0)} \middle| \mathcal{V} \middle| \tilde{\Phi}^{(0)} \right\rangle \\ E^{(2)} &= \left\langle \tilde{\Phi}^{(0)} \middle| \mathcal{V} \middle| \tilde{\Phi}^{(1)} \right\rangle \\ E^{(3)} &= \left\langle \tilde{\Phi}^{(0)} \middle| \mathcal{V} \middle| \tilde{\Phi}^{(2)} \right\rangle . \end{aligned} \tag{1.24}$$

Our task then becomes determining the $\tilde{\Phi}^{(n)}$ in terms of the eigenfunctions of $\mathcal{H}_0$. Coming back to MP perturbation theory where the choice of $\mathcal{H}_0$ is the Fock operator meaning that $\left|\tilde{\Phi}^{(0)}\right\rangle \equiv \left|\bar{\Phi}^{(0)}\right\rangle$ and $\mathcal{V}$ becomes:

$$\mathcal{V} = \sum_{i<j} \frac{1}{r_{ij}} - \sum_i v^{\mathrm{HF}}(i), \tag{1.25}$$

where v^{HF} is the potential from Hartree-Fock. Given $\mathcal{V}$, and using the same basis of determinants as before (the basis where determinants differ by which orbitals from Hartree-Fock they include) we can now approximate $\tilde{\Phi}^{(n)}$ by expanding it in this basis. Within MP perturbation theory the first energy that is an improvement on Hartree-Fock is the second order treatment and as it turns out $\tilde{\Phi}^{(1)}$ can be expanded only in the determinants where two orbitals have been replaced. Since by design $\bar{\Phi}^{(0)}$ is orthogonal to all other determinants in our Slater basis, we can project $\tilde{\Phi}^{(1)}$ into the basis of all determinants that are not the ground state one forcing it to be orthogonal to $\bar{\Phi}^{(0)}$

$$\tilde{\Phi}^{(1)} = \sum_{\substack{abc\ldots \\ ijk\ldots}} \left|\bar{\Phi}^{abc\ldots}_{ijk\ldots}\right\rangle \left\langle \bar{\Phi}^{abc\ldots}_{ijk\ldots} \middle| \tilde{\Phi}^{(1)} \right\rangle, \tag{1.26}$$

where we assume the summation only goes over unique determinants. In reality, the Brillouin condition (which says that matrix elements of the type $\left\langle \bar{\Phi} \middle| \mathcal{H} \middle| \bar{\Phi}^a_i \right\rangle$ are zero) and the fact

that $\frac{1}{r_{ij}}$ can couple determinants that differ by at most 2 orbitals allows us to expand $\tilde{\Phi}^{(1)}$ only in terms of double replacements.

$$\tilde{\Phi}^{(1)} = \sum_{\substack{ab \\ ij}} \left|\bar{\Phi}_{ij}^{ab}\right\rangle \left\langle \bar{\Phi}_{ij}^{ab} \middle| \tilde{\Phi}^{(1)} \right\rangle, \tag{1.27}$$

By making use of the projection of Equation (1.27) and the second term from Equation (1.23) we can eventually arrive at the following expression for the $E^{(2)}$, in closed shell form:[4]

$$E^{(2)} = \sum_{ijab} \frac{(2(ia|jb) - (ib|ja))(ia|jb)}{\epsilon_{\mathrm{i}} + \epsilon_{\mathrm{j}} - \epsilon_{\mathrm{a}} - \epsilon_{\mathrm{b}}}, \tag{1.28}$$

where

$$(ia|jb) = \iint \psi_i^*(\mathbf{r}_1)\psi_j^*(\mathbf{r}_2)\tfrac{1}{r_{12}}\psi_a(\mathbf{r}_1)\psi_b(\mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2, \tag{1.29}$$

and ϵ_{a} is the eigenvalue of eigenfunction to the Fock operator, ψ_a. The calculation of $E^{(2)}$ is commonly called the MP2 method and although it is not sufficiently accurate for quantitative study it is an improvement on the Hartree-Fock energy and does often give qualitatively correct results as can be seen in Figure 1.3.

1.3.2 Brief Introduction of Big O

In order to discuss the relative cost of different algorithms in this section, we will define a notation that loosely differentiates between algorithms with different orders of scaling. Any algorithm will have some cost associated with it that is a function of the size of the input to that algorithm. For example, the number of mathematical operations that must be

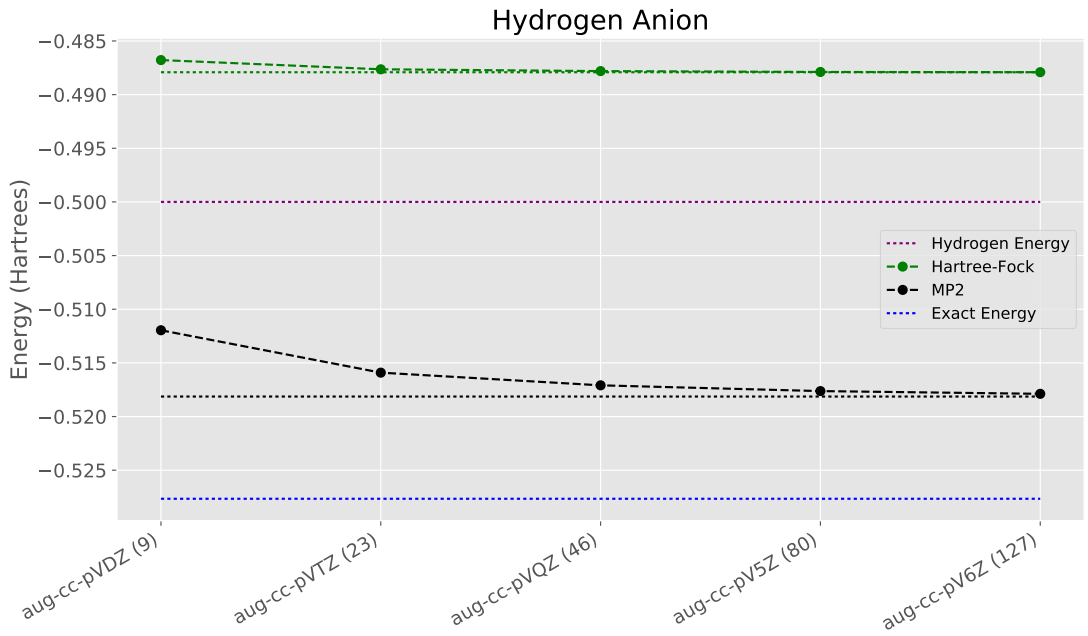

Figure 1.3: A calculation very similar to Figure 1.2, except that the non-diffuse basis sets have been removed and MP2 is now compared with Hartree-Fock. It is clear that while MP2 is quite aways off from the exact —CCSD— energy for this system, it does qualitatively predict that the hydrogen anion will be bound. The dashed black line represents the basis set limit for MP2.

undertaken in a matrix multiplication can be determined from:

$$C_{m,n} = C_{m,n} + \sum_{k} A_{m,k} * B_{k,n}. \tag{1.30}$$

Equation (1.30) shows us that the number of operations needed to update a single element of $\mathbf{C}$ is k multiplications and k additions. Since $\mathbf{C}$ has $m \times n$ elements that means that the cost of computing all elements of $\mathbf{C}$ must be $m \times n \times k$ multiplications and $m \times n \times k$

additions. We can write out the cost of updating $\mathbf{C}$ as

$$\text{Cost}(\mathbf{C}) = mnk + mnk = 2mnk \tag{1.31}$$

and if $m = n = k$ then the cost is $2m^3$. In order to avoid the requirement of listing out the exact cost of every algorithm we will adopt a shorthand called Big O notation.[2] Big O notation designated by a $\mathcal{O}$ gives us a fast way to see what the asymptotic scaling of an algorithm is, although it does not always tell us what the most expensive step for a specific input might be. To apply Big O to the cost of our matrix update we will keep only the fastest growing terms and then we will drop the prefactor of those terms, so the Big O of Equation (1.30) becomes $\mathcal{O}(m^3)$. This description of Big O is far from complete, but should help the reader for the remainder of this document and hopefully it captures the informal way in which Big O is actually used within the electronic structure community.

1.3.3 Relative Cost of Electronic Structure Methods

While post Hartree-Fock methods can offer significant improvement in the energy and wavefunction, they do have their downsides. The time to apply these corrections (with the exception of exact methods which grow as $\mathcal{O}(n!)$) at best grows as large polynomials with an increase in either the number of electrons treated or as the size of the basis set is increased. For a large enough system these methods require anywhere from slightly more computer time, than Hartree-Fock, to what might as well be an infinite amount of computer time to compute. In fact Hartree-Fock itself is formally $\mathcal{O}(n^4)$, although in practice, for three dimensional systems and accurate basis sets, approximations allow for calculations on

[2] Computer scientist may object that what we are going to call Big O should really be called Big Θ notation, but unfortunately the use of Big O is so well established in many fields that trying to switch to more formal notation would likely be more confusing than helpful.

Table 1.3: The formal asymptotic computational complexity of commonly used electronic structure methods with respect to the number of particles.[11]

MP2	$\mathcal{O}(n^5)$
CCSD	$\mathcal{O}(n^6)$
CCSD(T)	$\mathcal{O}(n^7)$
CCSDT	$\mathcal{O}(n^8)$
CCSDT(Q)	$\mathcal{O}(n^9)$

large molecules that are dominated by a high-prefactor $\mathcal{O}(n^2)$ step. The formal cost of some of the most commonly used corrections to Hartree-Fock are listed in Table 1.3

One cost that all of the above methods share is the $\mathcal{O}(n^5)$ integral transformation of the four-center two-electron integrals from the atomic orbital space to the molecular orbital space. This step shown in Equation (1.32), while only a computational bottleneck for MP2, is expensive and deserves mentioning. Upon first glance, it may appear that the transformation to molecular orbital integrals is $\mathcal{O}(n^8)$

$$(ia|jb) = \sum_{\mu\nu\rho\sigma} C_{\mu i} C_{\nu a} (\mu\nu|\rho\sigma) C_{\rho j} C_{\sigma b}, \tag{1.32}$$

where $\mathbf{C}$ is the coefficient matrix that describes the molecular orbitals in terms of the atomic orbital functions. But it is possible to rearrange the sums and use temporary storage to reduce this cost to four $\mathcal{O}(n^5)$ operations:

$$\begin{aligned}
(\mu\nu|\rho b) &= \sum_{\sigma} (\mu\nu|\rho\sigma) C_{\sigma b} \\
(\mu\nu|jb) &= \sum_{\rho} (\mu\nu|\rho b) C_{\rho j} \\
(\mu a|jb) &= \sum_{\nu} C_{\nu a} (\mu\nu|jb) \\
(ia|jb) &= \sum_{\mu} C_{\mu i} (\mu a|jb).
\end{aligned} \tag{1.33}$$

Some methods also require the more costly transform that generates the $(ab|cd)$ integrals,[3] although it has the same $\mathcal{O}(n^5)$ complexity —notice the imprecise nature of Big O notation and how it can only be used to compare methods with different scaling. The integral transformation of Equation (1.33) is almost always the first step for any post Hartree-Fock calculation and since it is shared among many methods the next section will investigate a commonly used approximation that reduces the time required to compute these integrals, but does not change their scaling.

1.4 Density Fitting

The previous sections contain all of the information needed to compute accurate electronic energies and wavefunctions, but they make no mention about the how to compute them efficiently. In this section, we will discuss one approximation that while not new, has recently seen rapid adoption for post Hartree-Fock methods and some adoption for Hartree-Fock (although the benefits are less pronounced). This approximation at the most basic level takes the integrals of the type:

$$(\mu\nu|\rho\sigma) = \iint \frac{\phi_\mu^*(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\phi_\rho^*(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)}{r_{12}} \mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2, \tag{1.34}$$

and asks the question: can the product of terms for a single particle (think $\mathbf{r}_1$) be expressed not as a product of functions, but as a linear combination of one particle functions. More precisely is the following approximation a good one:

$$\phi_\mu^*(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1) \approx \sum_X C_X^{\mu\nu}\phi_X(\mathbf{r}_1)? \tag{1.35}$$

[3] Since the size of the virtual space is almost always greater that $\frac{M}{2}$ these integrals are more expensive in absolute terms

The answer of course will depend on many things such as how many X are need for sufficient accuracy, the form of the $\{\phi_X\}$, and how difficult is it to solve for the $C_X^{\mu\nu}$ coefficients. Too attempt to solve these equations we might try projecting on the left by $\iint \phi_Y(\mathbf{r}_2)\mathcal{M}(\mathbf{r}_1,\mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2$:

$$\iint \phi_Y(\mathbf{r}_2)\mathcal{M}(\mathbf{r}_1,\mathbf{r}_2)\phi_\mu^*(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 = \sum_X C_X^{\mu\nu}\iint \phi_Y(\mathbf{r}_2)\mathcal{M}(\mathbf{r}_1,\mathbf{r}_2)\phi_X(\mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 \tag{1.36}$$

$$(Y|\mathcal{M}|\mu\nu) = \sum_X C_X^{\mu\nu}(Y|\mathcal{M}|X) \tag{1.37}$$

where Equation (1.37) is a concise short hand for these integrals and $\mathcal{M}$ is some positive definite operator, usually $\frac{1}{r_{12}}$. Realizing then that Equation (1.37) is a system of equations then it can be solved by inverting the matrix $M_{Y,X} = (Y|\mathcal{M}|X)$ and using $\mathbf{M}^{-1}$ to solve for the $C_X^{\mu\nu}$ coefficients:

$$C_X^{\mu\nu} = \sum_Y M_{Y,X}^{-1}(Y|\mathcal{M}|\mu\nu). \tag{1.38}$$

Once these coefficients have been determined the integral from Equation (1.34) can be approximated:

$$(\mu\nu|\rho\sigma) \approx \sum_{XY} C_X^{\mu nu} C_Y^{\rho\sigma}\iint \frac{\phi_X(\mathbf{r}_1)\phi_Y(\mathbf{r}_2)}{r_{12}}\mathrm{d}\mathbf{r}_1\mathbf{r}_2 \tag{1.39}$$

$$(\mu\nu|\rho\sigma) \approx C_X^{\mu\nu}(X|Y)C_Y^{\rho\sigma}, \tag{1.40}$$

where $\mathcal{M}(\mathbf{r}_1,\mathbf{r}_2) \equiv \frac{1}{r_{12}}$ and in Equation (1.40) the summation over X and Y is assumed. Now that we have an approximation for $(\mu\nu|\rho\sigma)$ that splits particles 1 and 2 we will see how it helps with equation Equation (1.33), assuming that the number of X is linear in the

number of particles in the calculation:

$$C_X^{\prime\mu a} = C_X^{\prime\mu\nu} C_{\nu a} \qquad \mathcal{O}\left(n^4\right) \tag{1.41}$$

$$C_X^{\prime ia} = C_{\mu i} C_X^{\prime\mu a} \qquad \mathcal{O}\left(n^4\right) \tag{1.42}$$

$$B_Y^{ia} = (Y|X) C_X^{\prime ia} \qquad \mathcal{O}\left(n^4\right) \tag{1.43}$$

$$(ia|jb) \approx B_Y^{ia} C_Y^{\prime jb} \qquad \mathcal{O}\left(n^5\right), \tag{1.44}$$

where $\mathbf{C}'$ is the coefficients from Equation (1.38), unfortunately $\mathbf{C}$ is often used for the fitting coefficients as well as the molecular orbital coefficients in the literature. Equations (1.41) to (1.44) take a process that would have four expensive $\mathcal{O}\left(n^5\right)$ steps and approximates it using an approach that only has one $\mathcal{O}\left(n^5\right)$ step. This is also an example where Big O notation fails to capture the true cost of the operations; due to the nature of modern computer architectures Equation (1.44) is typically more efficient than any of the $\mathcal{O}\left(n^5\right)$ steps in Equation (1.33).

As the title of this section suggests, it this type of factorization is often called density fitting, but sometimes it is abbreviated RI after resolution of the identity. Density fitting plays an important role in all of the remaining chapters, and a more in-depth discussion of it along with many historical references is given in Chapter 3

1.5 LCAO Electronic Structure by Analogy

To make what I think is a useful analogy for non-chemists we will compare electronic structure theory to the bust of President Obama in Figure 1.4. Let us think of Figure 1.4 as a good approximation for President Obama himself. In that sense, then we can think of the bust as being $\tilde{\Phi}$ or the approximation to President Obama. For this analogy to work, it is important

Figure 1.4: Bust of the 44th President of the United States.[13]

to know how this bust was made. This bust of the former President was constructed not by a skilled sculptor but instead is a 3D printed bust where the 3D model was created using the light and camera array in Figure 1.5. At this point you —the reader— are probably asking yourself what could this possibly have to do with electronic structure and why are we comparing the bust to $\tilde{\Phi}$. Well dear reader we are going to potentially over do an analogy (really the only *right* way to do analogies) and compare the construction of this bust to the picture we have painted of approximations within electronic structure theory all the way down to the deepest of levels. As I have already stated let us think of the bust as $\tilde{\Phi}$ and then outline a list of all the other comparisons: section.

- The 3D model for the bust is made of combinations of pictures much like $\tilde{\Phi} \approx \sum_k C_k \left|\bar{\Phi}_k\right\rangle$ where each $\left|\bar{\Phi}_k\right\rangle$ is a single Slater determinant. This means in our analogy a Slater determinant must be similar to one 2D picture. This is a nice idea in the sense that each determinant captures some low dimensional slice of the behavior of the true wavefunction, a picture captures a 2D slice of some 3D object.

Figure 1.5: Array of lights and cameras used to capture the 3D information of President Obama's face. That information was then used to create the bust in Figure 1.4 (taken by Pete Souza).[14]

- A Slater determinant must be constructed using some basis though, in our case this is usually atomic like orbitals since we work in the LCAO framework. Similarly, a digital pictures can be though of as being represented by a collection of pixels. Just like how Slater determinants get more accurate as the number of basis functions increase, so too pictures become a more faithful representation of reality as the number of pixels used in their representation increases.

- Finally, one detail which I neglected to mention earlier in this section is that usually those atomic functions that we use to form our basis sets are not truly atomic functions, for technical reasons it is difficult to integrate atomic functions so they are approximated as a linear combination of Gaussian functions. Eerily pixels are also not actually little specks of pure color, but instead the color of each pixel is generated based on a combination of a red, a green, and a blue light (at least for RGB monitors, also interestingly at least circa 2018 digital cameras actually mainly detect these three colors). So the final part of our analogy is that our atomic basis functions are actually approximated just like the color of each pixel in an image is really a combination of three simple colors.

The remainder of this section will discuss the last two analogies in greater detail.

1.5.1 Atomic Orbital Functions are Like Pixels

Let us start with a regular 2D picture of President Obama given in Figure 1.6. This portrait of President Obama is made up of 340500 pixels (500×681) and it does a reasonably good job of representing the President, but Figure 1.7 and Figure 1.8 show that if too few pixels are used then important information is lost and we can imagine that a bust made from pictures of their quality would be a poor approximation indeed. Very similar to how too few

Figure 1.6: An official portrait of the 44th President of the United States.[15]

Figure 1.7: The portrait from Figure 1.6 compared to one using only 3400 pixels (50×68). At this point I think that it is still possible to tell who is in the picture on the right, but clearly a significant amount of detail has been lost.

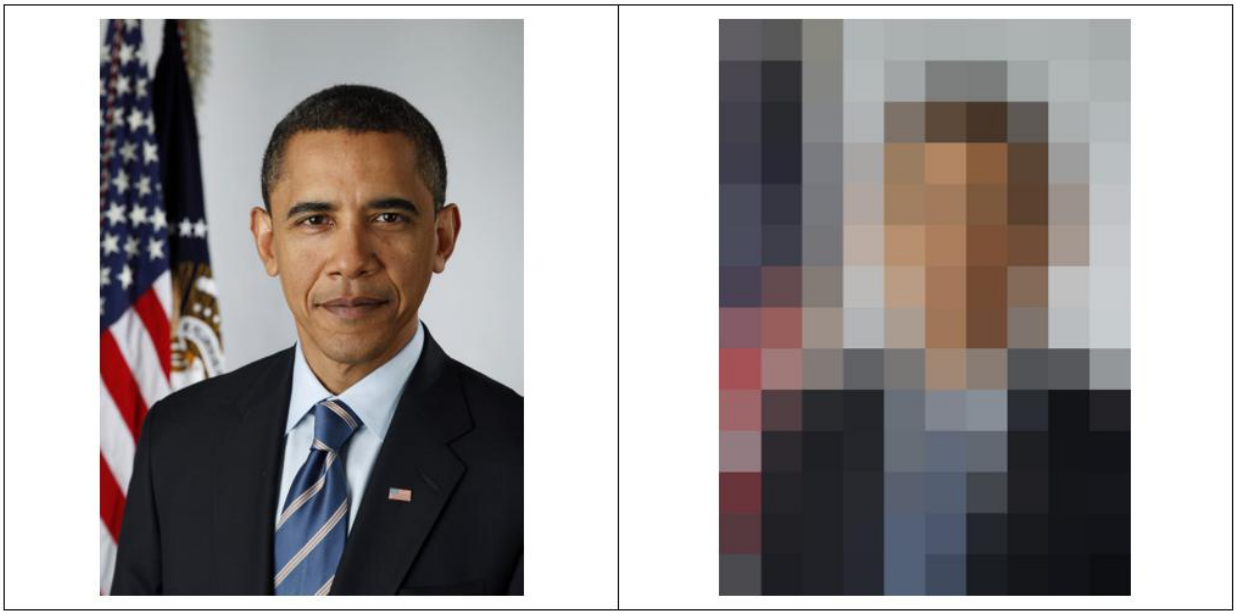

Figure 1.8: The portrait from Figure 1.6 compared to one using only 140 pixels (10×14). At this point it is really impossible to tell what this is a picture of except maybe to suggest it might be a representation of a person.

pixels leads to a very approximate representation of an exact 2D slice of President Obama's face, using too few atomic orbitals to expand our Slater determinants leads to a poor and only qualitative description of our wavefunction. Notice for example, how in Figure 1.3 the smallest basis (aug-cc-pVDZ) is noticeably worse for both Hartree-Fock and MP2 than the other larger basis sets. It is only by increasing the size of the atomic orbital basis that we can obtain accurate approximations to our wavefunctions.

1.5.2 Gaussian Functions are Like RGB Channels

As was briefly mentioned not all Atomic orbitals are created equally, due to the fact that integrals for the functions that mirror the solutions to the hydrogen atom are hard to compute. In fact, most calculations today are undertaken by approximating these functions as a sum of Gaussians. This approximation is not unlike how RGB subpixels are used to create colors

Figure 1.9: Representation of Figure 1.6 using only 2 shades for each of color channels in each pixel leading to 8 unique colors.

in image pixels. And just like with an image, where if too few shades of red, green, and blue are available the colors suffer, if the sum over Gaussians is not a good representation for the desired atomic orbital the calculation accuracy will suffer. Figure 1.9 and Figure 1.10 show what happens to our presidential portrait when too few colors are used to represent each pixel. Similarly in LCAO calculations if each atomic orbital is not sufficiently accurate then accuracy of the calculation will suffer. A similar effect can be seen in Figure 1.2 where the basis sets without diffuse functions not only require more functions to achieve a given level of accuracy than the ones with the diffuse functions, but also converge to the incorrect limit. This last point is slightly more in line with the idea that a certain color channel was removed from our subpixel array but the other channels have a sufficient number of shades. Let us say that we just did not have a blue subpixel, then colors that had a significant fraction of blue in them would be very poorly represented, similarly in the hydrogen anion the second electron is very weakly held and likely has a significant probability of being found far

Figure 1.10: Representation of Figure 1.6 using only 4 shades for each of color channels in each pixel leading to 64 unique colors.

away from the nucleus, for this reason it is necessary to include diffuse functions that have significant tails far away from the nucleus as well. Just like how our representation of say purple would suffer without a blue subpixel, so does our representation of this weakly held electron suffer without the long range diffuse functions.

Putting it all together we see that to make the bust of Figure 1.4 we needed high resolution pictures, and although the bust is colorless, we can imagine that if the bust was to have color we would have needed each picture to also capture a significant amount of RGB information such that the real colors were closely approximated. This is similar to how we construct accurate wavefunction using perturbation theory or coupled cluster theory. All of the post Hartree-Fock methods depend on first obtaining a good basis of Slater determinant for expansion of our wavefunction. But for that basis to lead to high accuracy we must include an ample number of Gaussian basis functions and those functions that we include must have both the correct structure and must accurately represent the atomic functions they are

approximating. Just as the presidential bust would suffer if too few images had been used or if those images had been of poor quality, so to will our wavefunction suffer if it is expanded in too few Slater determinants or if the quality of those determinants is not high enough.

Hopefully this analogy has been useful for both experts and non-experts alike. I personally find that analogies like this help us not only to understand what is happening in a complicated field such as electronic structure, but can also help lead us to new ways of thinking about things and potentially maybe even lead to new ideas. Also it is easy to get bogged down in the details of our complicated field and sometimes analogies like this can help by providing a mental map of where all the approximations lie and provide a nice visual example of how approximations at one stage of a calculation can effect the steps further down in the hierarchy.

1.6 Conclusions

The main purpose of this introduction has been to introduce the framework within which we compute electronic structure for well behaved molecules, where $\tilde{\Phi} = \bar{\Phi}$ is a reasonable but insufficient approximation. This section has been rather concise but that is largely because a correct treatment of this topic would require your author to reproduce the tome *Molecular Electronic-Structure Theory*[9], which incidentally is the source I recommend if for both a broad and in-depth coverage of the majority of topics in electronic structure. To end this section we will simply summarize the remaining chapters:

1. Chapter 2 discusses a novel technique for the factorization of blocks of the atomic orbital integrals used to make the computation of the exchange contribution to the Fock matrix, $\mathbf{F}$, more efficient

2. Chapter 3 discusses a similar idea for the linear scaling exchange, except instead of our factorization from Chapter 2 it uses local density fitting

3. Chapter 4 discusses a massively parallel linear scaling implementation of MP2 that makes use of pair natural orbitals

4. Finally, Chapter 5 presents a summary of the work and ties it all together.

These chapters each touch on how to take the ideas from this section and make further approximations to them in order to decrease the time it takes to compute energies and wavefunctions for Hartree-Fock and MP2. They are each self-contained and can be read independently of each other.

Chapter 2

Clustered Low Rank Tensor Approximation

2.1 Introduction

Despite the exponential complexity of many-body quantum mechanics —a manifestation of "the curse of dimensionality"— many important classes of problems, such as the electronic structure in chemistry and materials science, have robust polynomial solutions that become exact for some practical purposes.[16, 17, 18, 19, 20, 21] However, such solutions are limited by the high-order polynomial complexity in data and operations. For example, the straightforward implementation of CCSD[22] —the coupled-cluster[23] model with 1-body and 2-body correlations— has $\mathcal{O}(N^6)$ operation and $\mathcal{O}(N^4)$ data complexities. This is significantly more expensive than the corresponding $\mathcal{O}(N^4)$ and $\mathcal{O}(N^2)$ complexities of hybrid Kohn-Sham density functional theory (KS DFT) that dominates chemistry applications. Thankfully, fast algorithms improve on these figures.

Fast numerical algorithms trade precision and/or small-N cost for improved asymptotic scaling. A classic example is Strassen's algorithm for matrix multiplication[24] that has a higher operation count than the naïve algorithm for small matrices but is faster for large matrices, namely $\mathcal{O}\left(N^{\log_2 7}\right)$ vs. $\mathcal{O}(N^3)$. The operation count of Strassen-based implementation of CCSD is therefore $\mathcal{O}\left(N^{2\log_2 7}\right) \approx \mathcal{O}(N^{5.6})$. While Strassen's algorithm is formally exact, typically fast algorithms trade-off precision: A notable example of particular importance to the field of electronic structure is the Fast Multipole Method (FMM),[25, 26] which applies an integral (e.g. Coulomb) operator in $\mathcal{O}(N)$ operations, instead of $\mathcal{O}(N^2)$, for any finite precision ϵ.

In molecular electronic structure, fast algorithms traditionally take advantage of the sparse structure of operators and states. Such structure typically takes form of *element* or *block* sparsity. For example, the one-electron density matrix is conjectured[27, 28, 29] to decay exponentially in insulators when expressed in real space or in a localized, AO or Wannier,

basis (for noninteracting electrons the exponential decay can be shown exactly[30]). This result is key to fast density matrix formulation of one-particle theories and also rationalizes the exponential decay of the "exact" (Hartree-Fock) exchange operator appearing in hybrid DFT and many-body methods. Linear scaling methods based on direct density minimization (i.e. avoiding solving for orbitals) have been demonstrated by multiple groups.[29, 31, 32, 33, 34, 35] While the element-sparsity-based strategy is appropriate for LCAO in small basis sets and low-dimensional systems, the sparsity of the density matrix in three-dimensional systems is remarkably low,[36, 37, 38] especially in realistic (triple- and higher-ζ) basis sets, that are necessary for many-body methods and even hybrid DFT; e.g., the density matrix of a 32000-molecule water cluster is only 83% sparse (!) even when expressed in a double-ζ basis.[38] Similar conclusions can be drawn from the early attempts to develop practical *many-body* methods solely by using sparsity of correlation operators in the AO basis.[39] It is clear that the element sparsity alone is hardly sufficient for practical fast electronic structure. Another strategy used in fast electronic structure methods exploits *rank* sparsity. For example, the Coulomb operator, $\hat{V} f(x) \equiv \int \mathrm{d}x\, f(x')/|x - x'|$, has a dense matrix representation due to the slow decay of the integral kernel, but the rank of the off-diagonal blocks when expressed in localized basis is low, due to the smoothness of the kernel; of course, this is the basis of FMM.[25] The key lesson here is that while globally the Coulomb operator has no non-trivial rank-sparsity, ranks of the off-diagonal blocks are low in its localized matrix representation. Remarkably, similar local rank-sparsity is seen in other areas of electronic structure, *not* merely as a direct consequence of the integral operator properties. For example, blocks of many-body wave functions have low rank when expressed in localized basis; such rank-sparsity is the foundation of fast many-body methods based on local pair-natural orbitals (PNO)[40] recently demonstrated in linear-scaling form.[41, 42]

In this work we propose a practical approach to recovering element and rank sparsities

(termed here *data sparsity*) using a general tensor format framework called Clustered Low-Rank (CLR). Some existing approaches, such as PNO-based many-body methods, can be viewed as a specialized application of CLR. However, we demonstrate in this paper that CLR has powerful applications beyond this particular context.

Just as the motivation for the CLR format is familiar, the mathematical structure of the CLR approach is related to the existing ideas in rank-structured matrix algebras, e.g. semiseparable matrices [43], $\mathcal{H}$-matrices[44, 45], $\mathcal{H}^2$-matrices[46, 47], and hierarchically semiseparable matrices[48, 49]. These ideas have occurred in various forms for many decades, and interested reader is referred to Ref. [43] for a long list of references; only a very brief recap is given here. Most relevant to us is the connection of these matrix algebras with the rank-sparse matrices arising in integral boundary problems, studied first by Rokhlin[50] and Hackbusch[51]. Later, Tyrtyshnikov generalized this type of approximation to matrices arising in integral equations via a method he calls the mosaic-skeleton approximation (MSA). While more general, MSA still mandates that the integral kernels first satisfy a necessary set of conditions.[52] The main tenet of MSA is that matrices fitting these conditions contain a non-overlapping blocking called a mosaic in which many of the blocks have reduced rank.[53, 54] Shortly after Hackbusch premiered the $\mathcal{H}$-matrix concept, the $\mathcal{H}$ stands for hierarchical, much like MSA, approximations previously specific to certain integrals are cast into a general matrix framework. The main difference between MSA and $\mathcal{H}$-matrix is $\mathcal{H}$-matrices use a hierarchical block structure while MSA does not. In this regard CLR can be viewed as a refinement of MSA for general tensor data, with domain-specific geometric prescriptions for blocking the tensor dimensions; the lack of hierarchy is deemed important for compatibility with standard algorithms for matrix and tensor algebra in high-performance computing where tensor blocks are distributed across regular cartesian grid of processors.

The rest of the paper is organized as follows: in Section 2.2 we will discuss the details of

CLR, in Section 2.3 we discuss an application of CLR to density fitting approximation in the context of the Hartree-Fock (exact) exchange (CLR-DF-K) evaluation, and in Section 4.6 we discuss our findings as well as future applications of CLR in electronic structure and generally in numerical tensor computation.

2.2 Clustered Low Rank Approach

CLR may be thought of as a general-purpose tensor representation *framework*, which divides a tensor into sub-blocks (tiles) that are stored in either low-rank or full-rank (dense) form. For example, consider an order-k tensor $\mathbf{T}$ over ring $\mathcal{R}$, with dimension sizes $\{n_0, \dots n_{k-1}\}$:[1]

$$\mathbf{T} : \mathbf{x} \to \mathcal{R}, \quad \forall \mathbf{x} \equiv \{x_0 \dots x_{k-1}\}, x_i \in [0, n_i) \equiv \{0, 1, ..n_i - 1\}, i \in [0, k), \tag{2.1}$$

In other words, a tensor is a map from a k-dimensional integer *index* $\mathbf{x}$ in domain $[0, n_0) \times \cdots \times [0, n_{k-1})$ to a value in $\mathcal{R}$. All possible values of tensor can be represented as an n_0 by n_1 by ... n_{k-1} array. The elements of this array are indexed by k-dimensional indices $\mathbf{x}$ and are denoted as $\mathbf{T}_\mathbf{x}$.

The CLR representation of tensor $\mathbf{T}$ is defined as follows:

1. Each dimension i of the index domain is *tiled* by dividing the integer interval $[0, n_i)$ into ν_i nonoverlapping intervals, or *tiles*. Tiles are encoded by their boundaries $\delta^{(i)}_\chi$ as $[\delta^{(i)}_0, \delta^{(i)}_1), [\delta^{(i)}_1, \delta^{(i)}_2), \dots [\delta^{(i)}_{\nu_i}, \delta^{(i)}_{\nu_i+1})$, with $\delta^{(i)}_0 = 0$ (to reduce complexity, Greek letters will henceforth refer to tile quantities, and roman letters will refer to dimensions and element quantities). Tiles in each dimension i are indexed by tile indices $\chi \in [0, \nu_i)$

[1] Without sacrificing generality, we only consider real-valued tensors ($\mathcal{R} \equiv \mathbb{R}$ in Eq. (2.1)) and do not distinguish between contravariant and covariant indices.

and are denoted as $\tau_\chi^{(i)}$. $\beta_\chi^{(i)}$ denotes the size of tile χ in dimension i, $\beta_\chi^{(i)} \equiv \tau_{\chi+1}^{(i)} - \tau_\chi^{(i)}$. $\boldsymbol{\tau}^{(i)} \equiv \{\tau_0^{(i)}, \ldots \tau_{\nu_i-1}^{(i)}\}$ denotes the set of tiles for dimension i and will be termed its *tiling*.

2. A tiling of the k-dimensional tensor domain is obtained as a tensor product of tilings for each dimension, $\boldsymbol{\tau}^{(0)} \otimes \ldots \boldsymbol{\tau}^{(k-1)}$. The k-dimensional tile index $\boldsymbol{\chi} \equiv \{\chi_0, \ldots \chi_{k-1}\}$ refers to the k-dimensional index interval $\tau_{\chi_0}^{(0)} \otimes \cdots \otimes \tau_{\chi_{k-1}}^{(k-1)}$. The tensor *tile* is the set of tensor elements addressed by the element indices in tile $\boldsymbol{\chi}$. It can be viewed as a k-dimensional array with dimensions $\{\beta_{\chi_0}^{(0)} \ldots, \beta_{\chi_{k-1}}^{(k-1)}\}$.

3. Tensor tile $\mathbf{T}_{\boldsymbol{\chi}}$ is set to zero and not stored if *shape* predicate $z(\boldsymbol{\chi})$ evaluates to false. In this work $z(\boldsymbol{\chi})$ is true if the *per-tile-element* Frobenius norm of $\mathbf{T}_{\boldsymbol{\chi}}$ exceeds model threshold ϵ_{sp} (see Eq. (2.22)).

4. Each non-zero tensor block $\mathbf{T}_{\boldsymbol{\chi}}$ is compressed using an approximate low-rank tensor decomposition, whose precision is determined by model threshold ϵ_{lr}.

It is evident from the above description that CLR is similar to $\mathcal{H}$-matrices and MSA with respect to the structure of the matrix representations. However, the choice of tiling, shape, and low-rank decomposition schemes as well as the specific application chemical knowledge differentiates these methods. In the following Sections, we will specialize CLR for a particular set of tensors, but first a brief discussion of the design elements of CLR framework is in order.

- To maximize the computational savings from CLR the tensor indices must be ordered so that the functions that are "near" each other in the Hilbert space have nearby indices. In all applications in this paper this is possible by a priori ordering basis for each dimension according to geometric positions of atomic orbital and localized molecular orbital functions, using the *k-means* algorithm adapted to the atomistic context. (see

Section 2.2.1 for details). The index clustering improves not only the extent of data sparsity, but also is essential to maximize the data locality in the context of parallel computing[55].

- The *shape* predicate defines the block-sparse structure of CLR matrices, dictating whether a specific tile is kept and used in subsequent computation. In this work the shape predicate is based on vector norms of tiles (or the norm estimates); however, in electronic structure shape predicates are often imposed by the model and not determined by the actual tensor data. The imposed sparsity usually leads to reduced costs by skipping work *a priori*, but can introduce uncontrolled or suboptimally-controlled errors.

- The choice of tile compression methods must necessarily be problem-specific since the low rank is a consequence of the physical properties of tensors. Since tensor rank is not uniquely defined, there is no optimal decomposition for tensors, thus carefully-chosen heuristics are important. Even for matrices, SVD is usually not the optimal choice due to its high computational cost; rank-revealing decompositions are usually preferred in practice.

- Another factor in the choice of decomposition is its suitability for use in algebraic operations, where frequent recompressions will often be necessary. For example, for matrices SVD provide optimal low-rank representation for a given accuracy, but usually trading optimality for speed will be desired.

To demonstrate the utility of the CLR tensor format in practical applications, we applied it to a reduced-complexity construction of the Hartree-Fock, or exact, exchange (usually labeled by K) using density fitting (DF-K) as an initial example. DF (also known as resolution of the identity) technology reduces the computational cost of the Fock operator construc-

tion by decreasing its prefactor, especially in triple- or higher-ζ basis sets. Yet, relative to conventional methods for constructing the exchange matrix that requires $\mathcal{O}(N^2)$ storage and is straightforwardly evaluated in $\mathcal{O}(N)$ cost, DF-K has $\mathcal{O}(N^3)$ storage and $\mathcal{O}(N^4)$ operation complexities. As we demonstrate here, CLR-based formulation of DF-K readily outperforms DF-K already for small systems, all with robust control of precision. When applied to medium-sized molecules with large basis sets CLR-DF-K also outperforms the standard LinK algorithm[56], which is one of several well-known $\mathcal{O}(N)$ approaches to the exchange build in LCAO representation[56, 57, 58, 59]. Although the apparent complexity of CLR-DF-K for the largest systems considered here exceeds that of LinK, the CLR-DF-K approach should be useful in the context of more advanced schemes for the exchange build with the properly-linear complexity with the system size,[60, 61, 62, 63] as well as in the context of reduced-scaling many-body methods such as coupled-cluster.[64, 65]

2.2.1 Basis Clustering

It is well recognized that clustering of basis functions helps to reveal the element and rank-sparse structure of the data. In the context of "flat" (i.e. hierarchy-free) tensor data in atomistic quantum mechanics this is usually achieved by locality-preserving basis function reordering using space-filling curves. [66, 67] In this work we use a much simpler approach to clustering based on the standard k-means algorithms[68, 69, 70] augmented with additional heuristics for the atomistic context.

Given a set of Cartesian coordinates $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ (either atomic coordinates when clustering atoms/AOs, or the expectation values of the position operator $\hat{\mathbf{r}}$ when clustering localized

MOs) and the target number of clusters m, k-means seeks clusters that minimize

$$\sum_{i=1}^{m} \sum_{\mathbf{r} \in \mathcal{X}_i} ||\mathbf{r} - \mathbf{R}_i||^2, \tag{2.2}$$

where $\mathcal{X}_i$ represents the ith cluster (in our case, cluster is a group of atoms or orbitals) and $\mathbf{R}_i$ is the centroid of cluster $\mathcal{X}_i$. It is known that finding the global minimum for a given number of clusters is NP hard, but standard k-means algorithms[68] can be used to find acceptable clusterings relatively quickly. In practice k-means is both accurate and fast, and the clustering never takes a significant portion of the computation cost; see Ref. [71] for a deeper analysis of k-means.

The clusters of atoms are seeded using the *k-means++* algorithm[69], except the hydrogen atoms are always placed in the same cluster as its nearest heavy atom. Clusters are then optimized with the standard k-means in which the cluster center is defined by its center of mass rather than the centroid. We stop when the center of mass of each clusters reaches a (local) minima, or after 100 iterations. The procedure is repeated 50 times with random starting seeds (note that the use of pseudo-random number generators allows to ensure reproducibility of clustering, if so desired). From these 50 clusterings we use the one that has the lowest value for the objective function, Eq. (2.2). The clusters of atoms are then used to determine the clusters of AOs.

To determine the clusters of molecular orbitals, we initially seed our clusters with random orbital centers and perform 5 iterations of standard k-means. Because MO clustering must happen on every single SCF iteration we use a less robust initialization procedure than in the atom clustering in order to save time. Initializing clusters with random centers is computationally efficient, but may lead to less well balanced clusters than the k-means++ algorithm.

2.2.2 Block Sparsity in CLR Tensors

The DF-K method involves at most order-3 tensors, e.g. the so-called three-center two-electron Coulomb integral:

$$(\kappa|\mu\nu) \equiv \iint \phi_\kappa(\mathbf{r}_1)\frac{1}{r_{12}}\phi^*_\mu(\mathbf{r}_2)\phi_\nu(\mathbf{r}_2)\, d\mathbf{r}_1 d\mathbf{r}_2. \tag{2.3}$$

To reduce the computational complexity of the exchange term we seek to exploit both the data sparsity of this tensor and tensors derived from it. Due to the local nature of the atomic orbitals this tensor has only $\mathcal{O}(N^2)$ nonzero tiles. To avoid computing zero tiles it is mandatory to be able to estimate the tile norms accurately. For the Coulomb 3-index tensor (2.3) in an AO basis, blocks can be readily screened by using AO shell-wise integral estimators, *e.g.* ref [72]. An extension to atom and multi-atom blocks is straightforward. Tile-norm estimates define the sparsity of computed tensors (see Section 2.2.4 for details). For simplicity, we did not skip the computation of any blocks a priori, but did avoid computation of negligibly-small shell sets whose Schwartz estimate of Frobenius norm is below 10^{-12}. Tensor blocks were removed if the per-element Frobenius norm of the block is below a predetermined threshold, ϵ_{sp}.

2.2.3 Low-Rank Block Representation and Arithmetic

For each block of a CLR tensor an attempt is made to compress it to a low-rank *matrix* form. The choice to represent the tensors as matrices internally is largely due to the existence of robust and efficient low-rank factorizations for matrices; investigation of practical tensor factorizations is left to future work.[2]

[2] As of April 13th, 2018 we have not attempted to use tensor factorization within the CLR framework. Current work in the group on tensor factorizations may soon make this possible though.

Representing a tensor as matrix is commonly known as *matricization*, or *flattening*. For the order-3 tensors in density fitting the auxiliary basis index is always used as either a row or column index of the matrix while the other two indices, either both AO or one AO and one MO, are fused to create the remaining index. Using this approach a tile of the three center two electron integrals can be represented in low rank form:

$$(Q|\mu\nu) = \sum_r S_{Q,r} T_{\mu\nu,r}. \tag{2.4}$$

Before discussing multiplication and addition of CLR matrices, we must first consider the algebra of tiles in the low-rank representation.

Low-Rank Matrix Approximation

Matrix $\mathbf{A}_r \in \mathbb{R}^{n\times m}$ has rank $r \leq \min(n, m)$ if it can be represented *exactly* as a sum of outer products of no fewer than r linearly independent vectors

$$\mathbf{A}_r = \sum_{i=1}^{r} \mathbf{s}_i^{\mathbf{A}} \left(\mathbf{t}_i^{\mathbf{A}}\right)^{\dagger} \tag{2.5}$$

$$\equiv \mathbf{S}^{\mathbf{A}} \left(\mathbf{T}^{\mathbf{A}}\right)^{\dagger}, \tag{2.6}$$

where $\mathbf{S}^{\mathbf{A}} \equiv \left[\mathbf{s}_1^{\mathbf{A}}, \mathbf{s}_2^{\mathbf{A}}, \ldots \mathbf{s}_r^{\mathbf{A}}\right]$ is a matrix of column vectors $\mathbf{s}_i^{\mathbf{A}}$ and, similarly, $\mathbf{T}^{\mathbf{A}} \equiv \left[\mathbf{t}_1^{\mathbf{A}}, \mathbf{t}_2^{\mathbf{A}}, \ldots \mathbf{t}_r^{\mathbf{A}}\right]$. We seek to approximate a full-rank matrix $\mathbf{A}$ with a rank-r matrix $\mathbf{A}_r$ which satisfies

$$||\mathbf{A} - \mathbf{A}_r||_F \leq \epsilon_{\text{lr}} \tag{2.7}$$

and permits efficient arithmetic in decomposed form. In this work, we approximate matrices using the rank-revealing QR decomposition (RRQR) since it is more efficient than SVD and approximates the SVD (optimal) ranks well.[73, 74, 75] The column-pivoted RRQR

decomposition of an $m \times n$ matrix $\mathbf{A}$ (we assume without loss of generality that $m \leq n$) is represented as

$$\mathbf{AP} = \mathbf{QR}, \tag{2.8}$$

where $\mathbf{P}$ is an $n \times n$ permutation matrix, $\mathbf{Q}$ is an $m \times m$ unitary matrix, and $\mathbf{R}$ is an $m \times n$ upper-triangular matrix. Selecting first r columns of matrix $\mathbf{R}$ partitions it into

$$\mathbf{R} \equiv \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{pmatrix}, \tag{2.9}$$

where $\mathbf{R}_{11}$ is an $r \times r$ upper-triangular matrix, $\mathbf{R}_{12}$ is an $r \times (n-r)$ matrix, and $\mathbf{R}_{22}$ is an $(m-r) \times (n-r)$ upper-triangular matrix. The maximum singular value of $\mathbf{R}_{22}$ is an upper bound to the $r+1$-st singular value of $\mathbf{A}$; furthermore it is easy to come up with a practical bound that only requires vector (Frobenius) norm computation:

$$\sigma_{r+1}(\mathbf{A}) \leq ||\mathbf{R}_{22}||_2 \leq ||\mathbf{R}_{22}||_F. \tag{2.10}$$

Thus to estimate the rank of $\mathbf{A}$ for given precision ϵ_{lr} (Eq. (2.7)) we compute full $\mathbf{R}$ (using the DGEQP3 LAPACK function) and numerically find r such that the $||\mathbf{R}_{22}^r||_F \leq \epsilon_{\text{lr}}$. While this method is not optimal, it is more efficient than SVD.

A given block $\mathbf{A}$ is stored in low-rank form only if the total storage of $\mathbf{S}^{\mathbf{A}}$ and $\mathbf{T}^{\mathbf{A}}$ is below that of $\mathbf{A}$. For example, a square $m \times m$ matrix $\mathbf{A}$ is stored in low-rank form if its rank $r < \frac{m}{2}$.

Multiplication of Low-Rank Matrices

Clearly, $\mathbf{C} = \mathbf{AB}$ can be directly computed in low-rank form $\mathbf{C} = \mathbf{S^C}\left(\mathbf{T^C}\right)^\dagger$ using low-rank representations of $\mathbf{A}$ and $\mathbf{B}$ with rank $\min(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}))$:

$$\begin{aligned}\mathbf{S^C}\left(\mathbf{T^C}\right)^\dagger =& \mathbf{S^A}\left(\mathbf{T^A}\right)^\dagger \mathbf{S^B}\left(\mathbf{T^B}\right)^\dagger && (2.11)\\ =& \begin{cases}\mathbf{S^A}\left(\left(\left(\mathbf{T^A}\right)^\dagger \mathbf{S^B}\right)\left(\mathbf{T^B}\right)^\dagger\right) & r_A < r_B\\ \left(\mathbf{S^A}\left(\left(\mathbf{T^A}\right)^\dagger \mathbf{S^B}\right)\right)\left(\mathbf{T^B}\right)^\dagger & r_A \geq r_B\end{cases}, && (2.12)\end{aligned}$$

in which the order of evaluation and hence the definitions of $\mathbf{S^C}$ and $\mathbf{T^C}$ are specified by the parentheses. Note that the multiplication never increases matrix rank.

The special cases where either $\mathbf{A}$ or $\mathbf{B}$ is full rank are also straightforward.

Addition of Low-Rank Matrices

Unlike the multiplication, addition of low-rank matrices may increase rank as can be immediately seen:

$$\begin{aligned}\mathbf{C} =& \mathbf{S^A}\left(\mathbf{T^A}\right)^\dagger + \mathbf{S^B}\left(\mathbf{T^B}\right)^\dagger\\ =& \mathbf{S^{A+B}}\left(\mathbf{T^{A+B}}\right)^\dagger && (2.13)\end{aligned}$$

where $\mathbf{S^{A+B}} = \left[\mathbf{S^A}, \mathbf{S^B}\right]$ and $\mathbf{T^{A+B}} = \left[\mathbf{T^A}, \mathbf{T^B}\right]$. The maximum rank of $\mathbf{C}$ in (2.13) is the sum of the ranks of $\mathbf{A}$ and $\mathbf{B}$ if columns of $\mathbf{S^{A+B}}$ and/or $\mathbf{T^{A+B}}$ are linearly independent. Of course it is often the case that the ranks of these matrices can be reduced further with controlled error. We can reduce the ranks of $\mathbf{S^{A+B}}$ and $\mathbf{T^{A+B}}$ as follows. Starting with QR

decompositions of these matrices,

$$\mathbf{S}^{\mathbf{A}+\mathbf{B}} = \mathbf{Q}^{\mathbf{S}}\mathbf{R}^{\mathbf{S}} \tag{2.14}$$

$$\mathbf{T}^{\mathbf{A}+\mathbf{B}} = \mathbf{Q}^{\mathbf{T}}\mathbf{R}^{\mathbf{T}}, \tag{2.15}$$

Eq. (2.13) is rewritten

$$\mathbf{C} = \mathbf{Q}^{\mathbf{S}}\mathbf{R}^{\mathbf{S}}\left(\mathbf{R}^{\mathbf{T}}\right)^{\dagger}\left(\mathbf{Q}^{\mathbf{T}}\right)^{\dagger}, \tag{2.16}$$

Intermediate $\mathbf{M} \equiv \mathbf{R}^{\mathbf{S}}\left(\mathbf{R}^{\mathbf{T}}\right)^{\dagger}$ is then RRQR decomposed resulting in reduced rank $r \leq \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B})$.

$$\mathbf{M} = \widetilde{\mathbf{Q}}\widetilde{\mathbf{R}} \overset{\epsilon_{\text{lr}}}{\approx} \widetilde{\mathbf{Q}}_r\widetilde{\mathbf{R}}_r \tag{2.17}$$

where for $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{R}}$ the tilde indicates the use of RRQR as opposed to traditional QR.

The final result for the low-rank form of $\mathbf{C}$ is

$$\mathbf{C} = \left(\mathbf{Q}^{\mathbf{S}}\widetilde{\mathbf{Q}}_r\right)\left(\widetilde{\mathbf{R}}_r\left(\mathbf{Q}^{\mathbf{T}}\right)^{\dagger}\right) \equiv \mathbf{S}^{C}\left(\mathbf{T}^{C}\right)^{\dagger}, \tag{2.18}$$

where

$$\mathbf{S}^{C} = \mathbf{Q}^{\mathbf{S}}\widetilde{\mathbf{Q}}_r \tag{2.19}$$

$$\left(\mathbf{T}^{C}\right)^{\dagger} = \widetilde{\mathbf{R}}_r\left(\mathbf{Q}^{\mathbf{T}}\right)^{\dagger}. \tag{2.20}$$

if after compression $\text{rank}(\mathbf{C}) \geq \frac{\text{fullrank}}{2}$ then $\mathbf{C}$ is converted to its full rank representation since no space savings is available.

When performing addition if exactly one of $\mathbf{A}$ or $\mathbf{B}$ is in the full rank representation, the addition is efficiently expressed as a single generalized matrix-matrix multiplication (GEMM) available as part of a standard-compliant implementation of BLAS. For example if $\mathbf{A}$ is low rank and $\mathbf{B}$ is full rank the addition proceed as follows:

$$\mathbf{C} = \mathbf{S}^{\mathbf{A}} \left(\mathbf{T}^{\mathbf{A}}\right)^{\dagger} + \mathbf{B}, \tag{2.21}$$

where $\mathbf{C}$ is in its full rank representation. The use of GEMM avoids the storage overhead of constructing a temporary full-rank form of $\mathbf{A}$.

2.2.4 Block-Sparse Arithmetic with CLR Tensors

An arithmetic operation on matrices/tensors composed of CLR-format blocks can be performed as a standard operations on dense matrices/tensors with standard block operations replaced by their CLR specializations. For the sake of computational efficiency, however, it is necessary to screen arithmetic expressions involving CLR-format blocks to avoid laboriously computing blocks that have small norms. Therefore we represent CLR matrices/tensors as their block-sparse counterparts, in which only some blocks are deemed to have non-zero norms; the surviving blocks are represented using CLR format. This allows us to exploit both block-sparsity and block-rank-sparsities. In the limit $\epsilon_{\text{lr}} \to 0$, CLR matrices/tensors become their ordinary block-sparse counterparts. Here we describe how the block-sparse structure of CLR tensors is utilized in arithmetic operations.

Block $\mathbf{A}_{\boldsymbol{\xi}}$ of CLR tensor $\mathbf{A}$ that is the result of an arithmetic operation is *nonzero* (hence, it will be computed) if its Frobenius norm satisfies

$$\|\mathbf{A}_{\boldsymbol{\xi}}\|_{\text{F}} \geq \epsilon_{\text{sp}} \operatorname{volume} \mathbf{A}_{\boldsymbol{\xi}} \tag{2.22}$$

where ϵ_{sp} is a non-negative threshold, and volume $\mathbf{A}_{\boldsymbol{\xi}}$ is the block *volume*, i.e. the number of elements in the block. Because the volume of each block within a tensor varies with the tiling details, the choice of a basis, etc., taking into account the block area defines the sparsity in a manner that is more robust than the Frobenius norm alone.

The sub-multiplicative property of the Frobenius norm is utilized to estimate the norm in addition and contraction operations, e.g. for addition

$$\|\mathbf{A}_{\boldsymbol{\xi}} + \mathbf{B}_{\boldsymbol{\xi}}\|_{\text{F}} \leq \|\mathbf{A}_{\boldsymbol{\xi}}\|_{\text{F}} + \|\mathbf{B}_{\boldsymbol{\xi}}\|_{\text{F}} \tag{2.23}$$

the upper bound provided by the RHS is used to estimate whether computing the block is warranted according to (2.22). Similarly, the norm of a contraction of two tensor blocks is bounded as

$$\begin{aligned} &\|\mathbf{A}_{\xi_i \ldots \xi_k \ldots} \mathbf{B}_{\xi_k \ldots \xi_j \ldots}\|_{\text{F}} \\ &\qquad \leq \|\mathbf{A}_{\xi_i \ldots \xi_l \ldots}\|_{\text{F}} \|\mathbf{B}_{\xi_l \ldots \xi_j \ldots}\|_{\text{F}} \end{aligned}, \tag{2.24}$$

where $\xi_i \ldots$ and $\xi_j \ldots$ are the outer block indices of the left- and right-hand tensors, respectively, and $\xi_k \ldots$ is the set of block summation indices. Eqs. (2.23), and (2.24) are combined to estimate the norm of the result blocks in a contraction operations:

$$\|\mathbf{C}_{\xi_i \ldots \xi_j \ldots}\| \leq \sum_k \|\mathbf{A}_{\xi_i \ldots \xi_k \ldots}\|_{\text{F}} \|\mathbf{B}_{\xi_k \ldots \xi_j \ldots}\|_{\text{F}} \tag{2.25}$$

where $\mathbf{C}_{i \ldots j \ldots}$ is equal to $\sum_{k \ldots} \mathbf{A}_{i \ldots k \ldots} \mathbf{B}_{k \ldots j \ldots}$. Note that we do not screen *individual* contributions to the result blocks, although others have done so.[76] In our case all contributions are computed when the estimated result block norm satisfies Eq. (2.22). Additionally, we avoid explicit computation of norms of low-rank blocks by using the multiplicative property

of Frobenius norm:

$$\|\mathbf{A}_{\boldsymbol{\xi}}\| \approx \|\mathbf{S}^{\mathbf{A}}_{\boldsymbol{\xi}}(\mathbf{T}^{\mathbf{A}}_{\boldsymbol{\xi}})^{\dagger}\|_{\mathrm{F}} \leq \|\mathbf{S}^{\mathbf{A}}_{\boldsymbol{\xi}}\|_{\mathrm{F}}\|\mathbf{T}^{\mathbf{A}}_{\boldsymbol{\xi}}\|_{\mathrm{F}}. \tag{2.26}$$

It is clear that computing norm estimates in complex expressions using upper bounds can quickly lead to poor norm estimates. Therefore, it is sometimes necessary to recompute the block norms and/or truncate *zero* blocks that are the result of computation. Similarly, after a CLR tensor contraction or addition operation it is possible that the rank of certain blocks are overestimated. Therefore in this work we recompute block norms and recompress after a sequence of tensor contractions to minimize the ranks and reduce the storage whenever beneficial.

2.3 Application: CLR-based Density Fitting Exchange

2.3.1 Density Fitting and Hartree-Fock Method

Density fitting (also known as the resolution-of-the-identity)[77, 78, 79, 80, 81] in the context of electronic structure involves fitting the electron density or more often any product of (one-electron) functions as a linear combination of basis functions from a fixed (auxiliary) basis set to minimize some functional. This allows us to compute standard 4-center 2-electron

integrals in terms of 2- and 3-center integrals:

$$(\mu\nu|\rho\sigma) \approx \sum_{QP} E_{Q,\mu\nu} \left(\mathbf{V}^{-1}\right)_{Q,P} E_{P,\rho\sigma} \tag{2.27}$$

$$= \sum_{X} B_{X,\mu\nu} B_{X,\rho\sigma}, \tag{2.28}$$

$$B_{X,\mu\nu} \equiv \sum_{Q} E_{Q,\mu\nu} \left(\mathbf{V}^{-1/2}\right)_{Q,X} \tag{2.29}$$

where $E_{Q,\mu\nu} \equiv (Q|\mu\nu)$ (defined in Eq. (2.3)) and $V_{P,Q} \equiv (P|Q)$ are the three- and two-center Coulomb integrals. In practice, instead of $\mathbf{V}^{-1/2}$ we use the inverse of Cholesky decomposition of $\mathbf{V}$; this is more efficient than computing the square root inverse and incidentally leads to better CLR compression in $\mathbf{B}$.

There are many different strategies for DF-K, here we use the standard MO-based approach[82] using Boys-Foster localized MOs[83], rather than the density-based (AO-only) algorithm:

1. Compute and store tensor $\mathbf{B}$ defined in Eq. (2.29).

2. Compute an intermediate tensor $\mathbf{W}$ with one index transformed from the AO to MO space as follows

$$W_{X,\mu i} = \sum_{\sigma} B_{X,\mu\sigma} C_{\sigma i} \tag{2.30}$$

   where the column vectors of $\mathbf{C}$ are the occupied molecular orbitals.

3. Form the exchange contribution to the Fock matrix

$$K_{\mu\nu} = \sum_{Xi} W_{X,\mu i} W_{X,\nu i}. \tag{2.31}$$

The two most expensive steps in computing the exchange matrix, for a given SCF iteration,

are the formations of $\mathbf{W}$ and $\mathbf{K}$, each of which requires approximately $2\mathcal{N}n^2o$ floating point operations where $\mathcal{N}$ is the size of the auxiliary basis, n is the size of regular basis, and o is the number of occupied orbitals in the system. The occ-RI algorithm can improve the times for $\mathbf{K}$ by computing only the occupied contribution to the exchange matrix[84], but was not used in this work.

Although the use of density fitting for Hartree-Fock exchange has the same formal complexity as the conventional Fock operator construction, namely $\mathcal{O}(N^4)$, great increase in efficiency is observe for small and medium-size systems in large orbital basis (triple-ζ and larger) due to avoiding the relatively expensive and repeated (once-per-iteration) computation of 4-center integrals. These advantages are exacerbated by the poor suitability of highly-irregular integral kernels to modern wide-SIMD architectures; in contrast, the dense matrix arithmetic of Eq. (2.29) can attain closer to peak performance on most modern architectures.

Despite the cost advantages of DF, its cubic storage requirements and quartic cost prohibit direct application to large systems. *Local* density fitting reduces the storage and computational complexity of density fitting by expanding localized products in terms of auxiliary basis functions that are "nearby" in some sense (geometric or otherwise)[61, 62, 85, 86, 87, 88, 89, 90]

Local DF approximations can be difficult to make robust *and* accurate, thus our hope for the CLR-based DF-K algorithm was to serve as a black-box reduced-scaling density fitting methodology without ad hoc imposition of local density fitting approximations.

2.3.2 Linear Scaling Exchange

In addition to comparison against the traditional $\mathcal{O}(N^4)$ DF-K, we compared timings and accuracy with the $\mathcal{O}(N)$ LinK algorithm[56]. LinK achieves linear scaling by neglecting individual shell contributions to the Fock matrix smaller than threshold ϵ using standard

Schwarz bounds for the Coulomb integrals[91] and careful reformulation of shell loop nests to avoid the $\mathcal{O}(N^2)$ overhead of loop iterations in the conventional shell-pair-list approach. The LinK algorithm was implemented in a modified version of the Libint library's four-center direct Hartree-Fock test program.[92] For simplicity, our implementation uses a $\mathcal{O}(N^2)$ pre-screening step, but even though it constitutes a minuscule fraction of the total time, we exclude it from the reported timings. The linear computational complexity was confirmed by performing exchange matrix computation for n-alkanes $C_{50}H_{102}$ and $C_{100}H_{202}$ in def2-SVP and cc-pVDZ basis sets; the estimated computational complexity of the algorithm is $N^{1.2}$.

2.3.3 Results

Computational Details

The CLR-DF-K method was implemented with the help of the open-source TiledArray parallel tensor framework[93]. We tested the performance and precision of the CLR-DF-K method by computing Hartree-Fock energies of quasi-one-dimensional (n-alkanes) and 3-D (water clusters) systems. Reference calculations for CLR-DF-K were computed using $\epsilon_{\text{sp}} = 10^{-16}$ and $\epsilon_{\text{lr}} = 0$. For non-reference calculations in compact basis sets $\epsilon_{\text{sp}} = 10^{-11}$ and in diffuse basis sets $\epsilon_{\text{sp}} = 10^{-12}$. Precision of the HF orbitals was further tested by computing Hartree-Fock dipole moments (see Figures S1 and S2 in Supporting Information) and canonical second order Møller-Plesset (MP2) energies for select systems.

Cartesian geometries for n-alkanes were generated using the Open Babel toolbox.[94] Cartesian coordinates for the water clusters were taken from ErgoSCF's public repository;[95] these clusters are random snapshots of molecular dynamics trajectories and are representative of liquid water structure at ambient conditions.[36]

Dunning's correlation consistent basis sets and the matching auxiliary basis sets were utilized

throughout this work.[8, 96] We also tested the def2-SVP[97] and cc-pVDZ-F12[98] basis sets.

The AO and MO dimensions were tiled as follows:

- Orbital AO basis was tiled *naturally* to contain one monomer (H_2O molecule or $CH_{2/3}$ unit) per tile; such tiling arises automatically when our k-means-based clustering is applied to $(H_2O)_n$ / C_nH_{2n+2} with the target number of clusters set to n.
- The auxiliary/density-fitting AO basis was tiled via k-means to produce half as many tiles as the orbital AO basis;
- The target number of tiles in the occupied MOs for $(H_2O)_n$ / C_nH_{2n+2} was set to n.

The details of the k-means-based clustering were given in Section 2.2.1.

To simplify the timing comparisons the low-rank addition (Section 2.2.3) was limited to the computation of the $\mathbf{B}$ tensor (Eq. (2.29)); the only exception was the parallel scaling tests, in which the low-rank addition was used throughout.

All computations were carried out on Virginia Tech Advanced Research Computing's NewRiver cluster where each node has 2 Intel Xeon E5-2680v3 2.5ghz CPUs, for a total of 24 cores and are available with either 128 GB or 512 GB of memory. Each node has a theoretical peak performance of 960 Gflops/s with a measured peak of 780 Gflops/s in Intel MKL's multithreaded implementation of DGEMM.

Our implementation supports massive distributed memory parallelism in the Fock matrix build due to the use of the TILEDARRAY framework.[93, 99] For technical reasons, two versions of TILEDARRAY used in our work. For the most faithful comparison against the serial LinK method implementation we used a copy of TILEDARRAY that does not use the Intel Thread Building Blocks (TBB) runtime. The rest of the computations (parallel scaling and block-size testing) used Intel TBB for task scheduling as it yields higher parallel efficiency

on modern multicore processors. The default configuration of TiledArray enables the use of Intel TBB if it is available.

The single threaded build of the CLR-DF-K code (not using TBB) was compiled with gcc 5.2.0 using single threaded Intel MKL 11.2.3 for linear algebra. The LinK test code was compiled using Intel icpc version 15.0.3. For parallel scaling and block size tests the CLR-DF-K code was compiled using icpc version 15.0.3 with single threaded Intel MKL 11.2.3 for linear algebra, Intel TBB 4.3.5 for task based parallelism, and the Intel impi 5.0 library for node based parallelism.

Impact on Storage Requirements of DF-K

To demonstrate the storage reduction attained by the CLR representation for the key order-3 tensors in the density-fitting-based exchange build, we reported the memory footprint of the **B** (compressed) and **W** (not compressed) tensors as a function of the orbital basis set size and the low-rank threshold $\epsilon_{\rm lr}$ in Figs. 2.1, 2.2, and 2.3.

As expected, even without the low-rank compression ($\epsilon_{\rm lr}$=0), the observed scaling of storage **B** with compact (non-diffuse) basis sets is better than $\mathcal{O}(N^3)$ of the fully-dense tensor representation due to savings from the block-sparsity ($\epsilon_{\rm sp} > 0$).[3] Even for the 3D systems the observed scaling is only slightly worse than the expected $\mathcal{O}(N^2)$ asymptote. For example, in cc-pVTZ basis the use of block-sparsity alone results in 36.7% storage savings for $(H_2O)_{32}$; for linear systems the savings are even more dramatic, e.g. 82.3% for $C_{50}H_{102}$. In diffuse basis sets the savings from sparsity are greatly reduced and the observed scaling of storage is close to cubic. Note that the onset of sparsity with a diffuse basis set is artificially protracted since the use of atom-based (i.e. not shell-based) clustering causes each AO tile to contain

[3] Note that due to the lack of support for permutational symmetry in TiledArray (the feature is under current development) the storage size for **B** is twice what it should be in practice.

at least one diffuse orbital; alternative clusterings that group diffuse shells together can help to increase the amount of sparsity but are not considered here.

The use of low-rank compression ($\epsilon_{\text{lr}} > 0$) together with the block-sparsity further reduces the storage for tensor $\mathbf{B}$ in CLR representation, especially with compact bases. E.g., with the cc-pVTZ basis the storage savings of $\epsilon_{\text{lr}} = 10^{-6}$ over $\epsilon_{\text{lr}} = 0$ (block-sparsity alone) were 85.6% for $(H_2O)_{32}$ and 83.2% for $C_{50}H_{102}$. For the largest 1-D and 3-D systems considered the storage savings from the low-rank compression are *more than an order of magnitude* relative to the block-sparse ($\epsilon_{\text{lr}} = 0$) representation. Furthermore, with the low-rank compression the observed storage *complexity* is significantly better than quadratic in compact basis sets. The benefit of the low-rank compression in CLR is also noticeable with diffuse basis sets, where it allows nearly quadratic scaling in 3D systems.

Note that we did not utilize compression in $\mathbf{W}$ because its memory footprint is much smaller than that of $\mathbf{B}$ by a factor of $\frac{n}{o}$ where n is the number of AOs and o is the number of occupied MOs. This is because for triple-zeta and larger basis sets the storage for $\mathbf{W}$ is completely negligible compared to $\mathbf{B}$ in the traditional dense representation. However, the low-rank compression of $\mathbf{B}$ in compact bases is so efficient that even in cc-pVTZ basis the CLR footprint of $\mathbf{B}$ becomes smaller than the storage needed for block-sparse uncompressed $\mathbf{W}$. This result suggests that the CLR approach can help improve the traditional density-fitting formulations which often trade-off speed for space by recomputing three-index 2-e integrals every SCF iteration. In the CLR-based algorithm it appears possible to reduce the memory footprint of $\mathbf{B}$ to where it can be stored in memory for systems with hundreds of atoms and thus avoid the need to use the integral-direct formulation (note that $\mathbf{B}$ does not depend on orbitals, hence can be computed once and used every SCF iteration). It is however clear that compression of $\mathbf{W}$ via rounded addition will be necessary to treat even larger systems; for simplicity we do not investigate this approach here as it would complicate the analysis

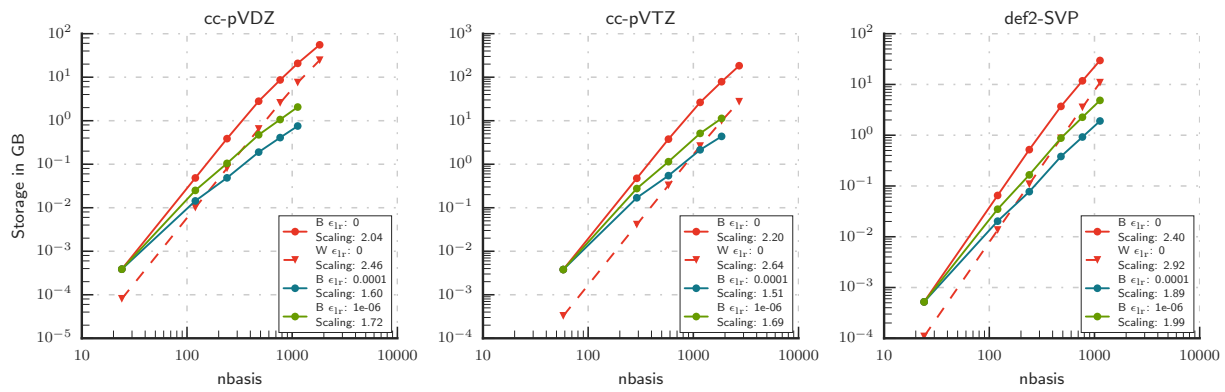

Figure 2.1: Variation of the order-3 tensor storage in CLR-DF-K with respect to number of basis functions and the compression threshold ϵ_{lr} for water clusters in tight basis sets.

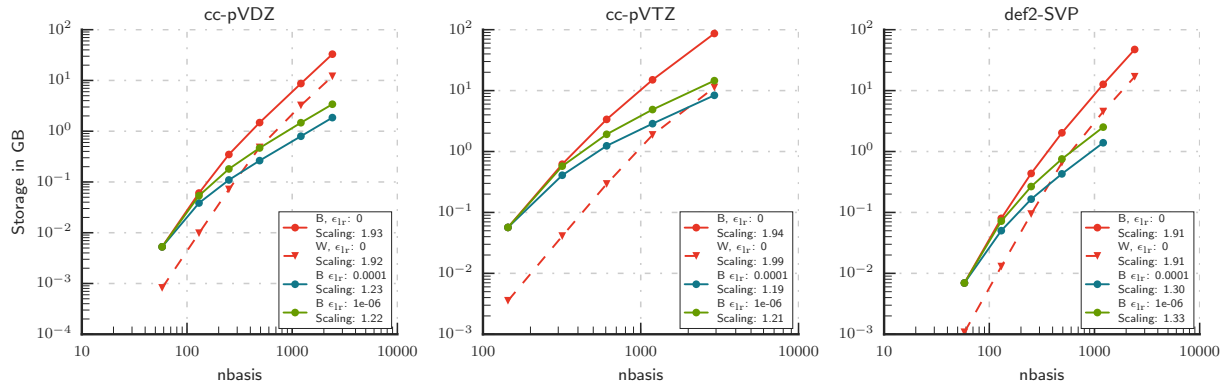

Figure 2.2: Variation of the order-3 tensor storage in CLR-DF-K with respect to number of basis functions and the compression threshold ϵ_{lr} for n-alkanes in tight basis sets.

of the errors introduced by the low-rank compression.

CLR-DF-K vs LinK: Performance and Precision

Computational cost and precision of the CLR-DF-K method was compared against the highly-robust LinK algorithm for computing the exchange matrix. The computational cost was defined as the wall time needed to compute the exchange matrix in serial (single-threaded) fashion to simplify the performance analysis. Of course, it only makes sense

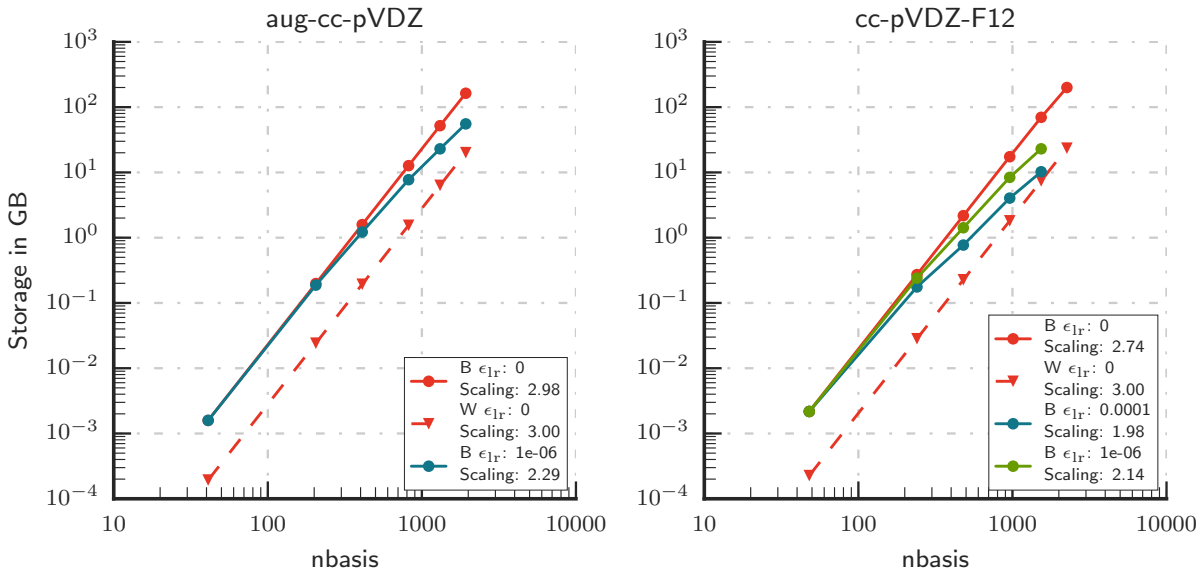

Figure 2.3: Variation of the order-3 tensor storage in CLR-DF-K with respect to number of basis functions and the compression threshold ϵ_{lr} for water clusters in diffuse basis sets.

to discuss speed in the context of precision, since both methods compute the exchange matrix approximately, with precision governed by one or more parameters. Hence our goal here is to compare the speed of a given CLR-DF-K computation against the LinK computation with comparable precision. Precision of a given LinK computation is thereby defined relative to the near-exact LinK method with the truncation threshold for exchange set to machine precision, or 2.2×10^{-16} (to save computational resources the Coulomb contribution to the Fock matrix was always evaluated with the truncation threshold of 10^{-12}, rather than machine precision). Similarly, precision of CLR-DF-K is defined here relative to the (near)exact DF-K method where $\epsilon_{\text{sp}} = 10^{-16}$ and $\epsilon_{\text{lr}} = 0$ (such definition excludes the error introduced by the density-fitting approach itself, since it is well-behaved and largely cancels for relative energy differences and other molecular properties).

With compact basis sets (cc-pVDZ, cc-pVTZ, and def2-SVP, see Figs. 2.4 and 2.5) the apparent computational complexity of CLR-DF-K is below the $\mathcal{O}(N^4)$ figure of the standard

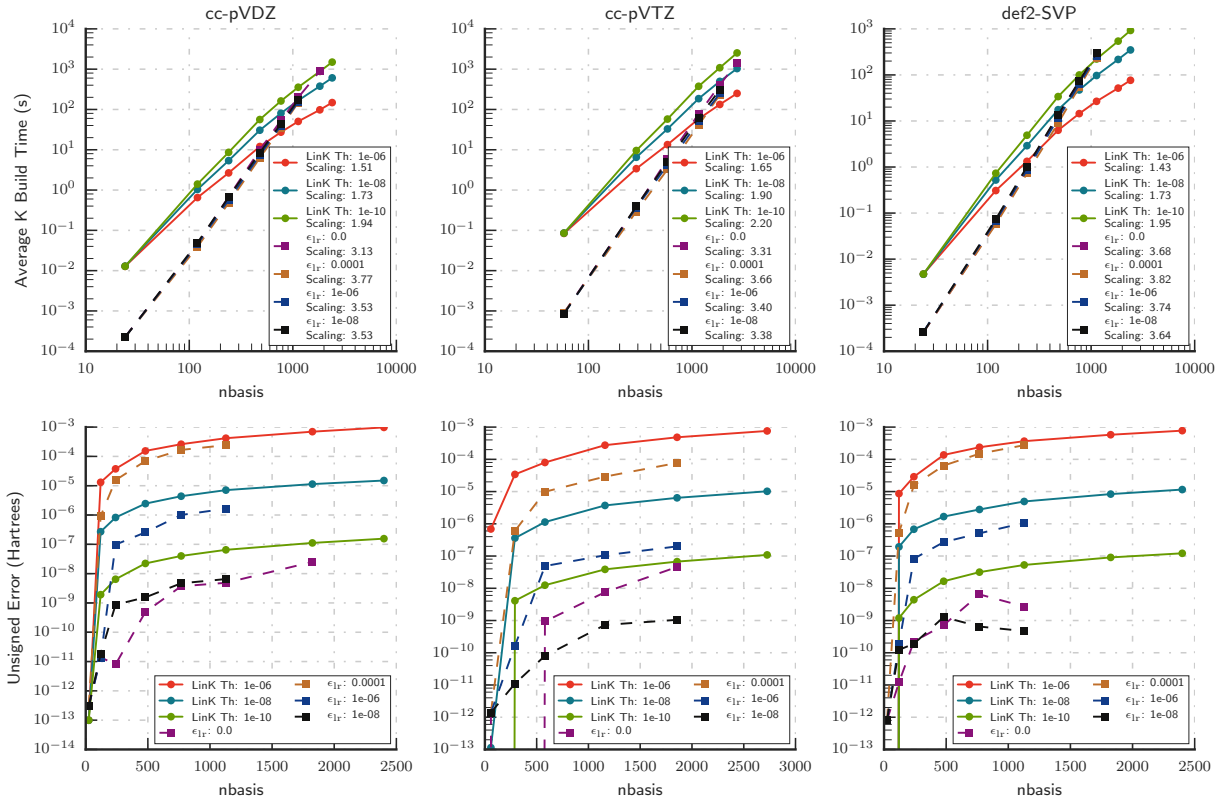

Figure 2.4: Computational cost and precision of CLR-DF-K and LinK exchange matrix builds for water clusters in tight basis sets.

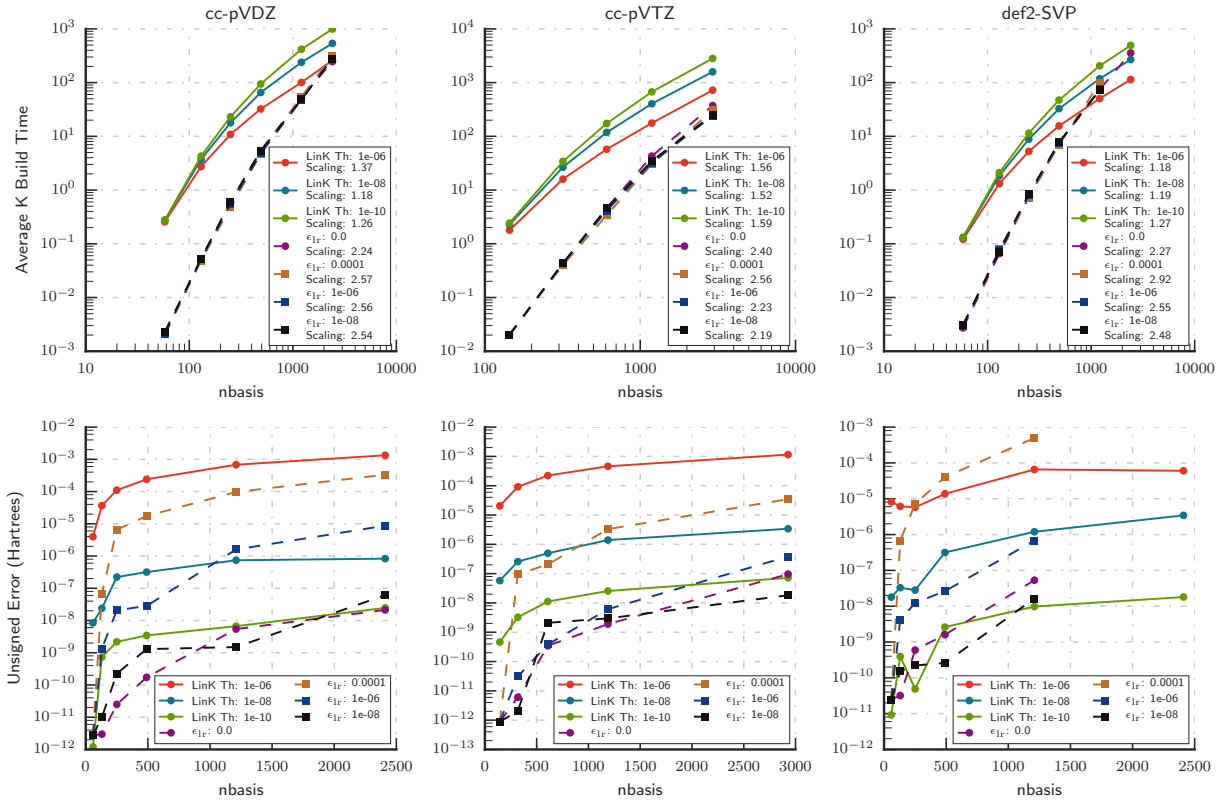

Figure 2.5: Computational cost and precision of CLR-DF-K and LinK exchange matrix builds for n-alkanes in tight basis sets.

DF-K approach for both n-alkanes and water clusters. Some of this reduction is purely due to the use of block-sparsity alone: the observed complexity is subquartic even when no low-rank compression is used (ϵ_{lr}=0). When using CLR compression in cc-pVTZ with $\epsilon_{\text{lr}} = 10^{-6}$, additional performance gains, as large as 38.6% and 35.3% over block-sparsity alone, were obtained for $(H_2O)_{32}$ and $C_{50}H_{102}$, respectively. Since the errors in absolute energies introduced by the low-rank compression are robustly controlled by the ϵ_{lr}, the use of CLR-DF-K method seems warranted as a black-box alternative to DF-K.

Of course, the complexity of CLR-DF-K is still too high to compete with the highly-optimized LinK approach. With compact basis sets the apparent complexity of LinK is subquadratic in most cases, and approaches near-linear for 1D systems in double-zeta basis. Thus although for smaller systems CLR-DF-K is significantly faster than LinK, the latter becomes faster for sufficiently large systems with compact bases. Although there is significant room to improve the performance of CLR-DF-K by optimizing tile sizes (see Fig. 2.9), it is clear that CLR-DF-K is unlikely to be useful as an alternative to LinK for compact basis sets.

The relative performance changes dramatically when diffuse basis sets are used (see Fig. 2.6) In all computations CLR-DF-K was faster than LinK, usually by a significant margin. E.g., for $(H_2O)_{32}$ in the aug-cc-pVDZ basis with $\epsilon_{\text{lr}} = 10^{-6}$ CLR-DF-K was 10 times faster than LinK using a threshold of 10^{-8}. Since the complexity of CLR-DF-K is still higher than that of LinK, we expect that eventually LinK will become faster than CLR-DF-K, however it is clear that for a significant range of molecular applications that call for the use of diffuse basis sets, such as in computations of molecular anions, response properties, and explicitly-correlated many-body methods the use of CLR-DF-K will be an economical alternative to LinK and DF-K.

To demonstrate that the accuracy of CLR-DF-K is not limited to energies, we used the Hartree-Fock orbitals computed with the CLR-DF-K method to evaluate the electric dipole

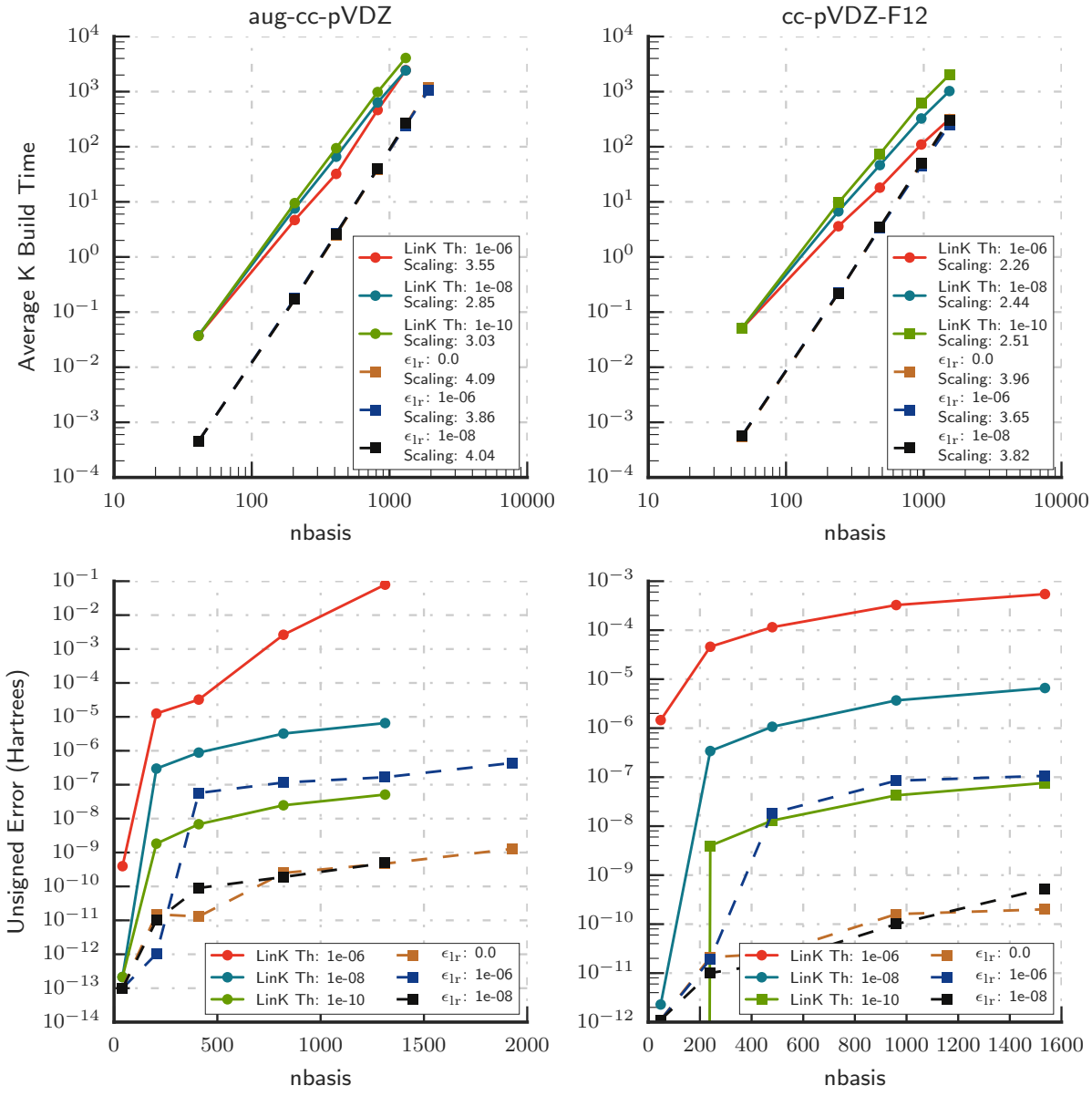

Figure 2.6: Computational cost and precision of CLR-DF-K and LinK exchange matrix builds for water clusters in diffuse basis sets.

moments and DF-MP2 energies (see Figs. S1, S2 and 2.7, respectively). Precision of these properties, just like that of the Hartree-Fock energy, is robustly controlled by the two CLR threshold parameters.

Performance Analysis

The performance comparison between LinK and CLR-DF-K must be judged cautiously since different factors will influence their computational efficiencies. The performance of LinK is largely determined by the computational expedience of the 4-center ERI evaluation, and our ERI engine is reasonably well optimized for at least single-thread execution. Compare this to CLR-DF-K which is dominated by small-matrix linear algebra in serial regime. Optimization of small-matrix linear algebra only recently received attention from the CS community, and mainstream linear algebra libraries perform poorly in this regime. Thus our implementation of CLR-DF-K should be considered highly preliminary, with significant potential for further optimization.

To understand the performance of the current CLR-DF-K implementation we compared it to the theoretical peak of traditional $\mathcal{O}(N^4)$ DF-K for which it is easy to build an accurate performance model by counting the number of floating point operations and converting to the wall time using the measured peak performance of DGEMM on our hardware, 780 GFlops per-node (see Fig. 2.8). Unsurprisingly, with the cc-pVDZ basis the block-sparsity alone is sufficient to best the performance of the hypothetical DF-K calculation, while with the cc-pVTZ basis we require the computational savings obtained by computing in the low-rank representation to best the hypothetical dense calculation. Thus even our preliminary implementation is sufficiently efficient to beat the *optimal* DF-K implementation for a sufficiently large system.

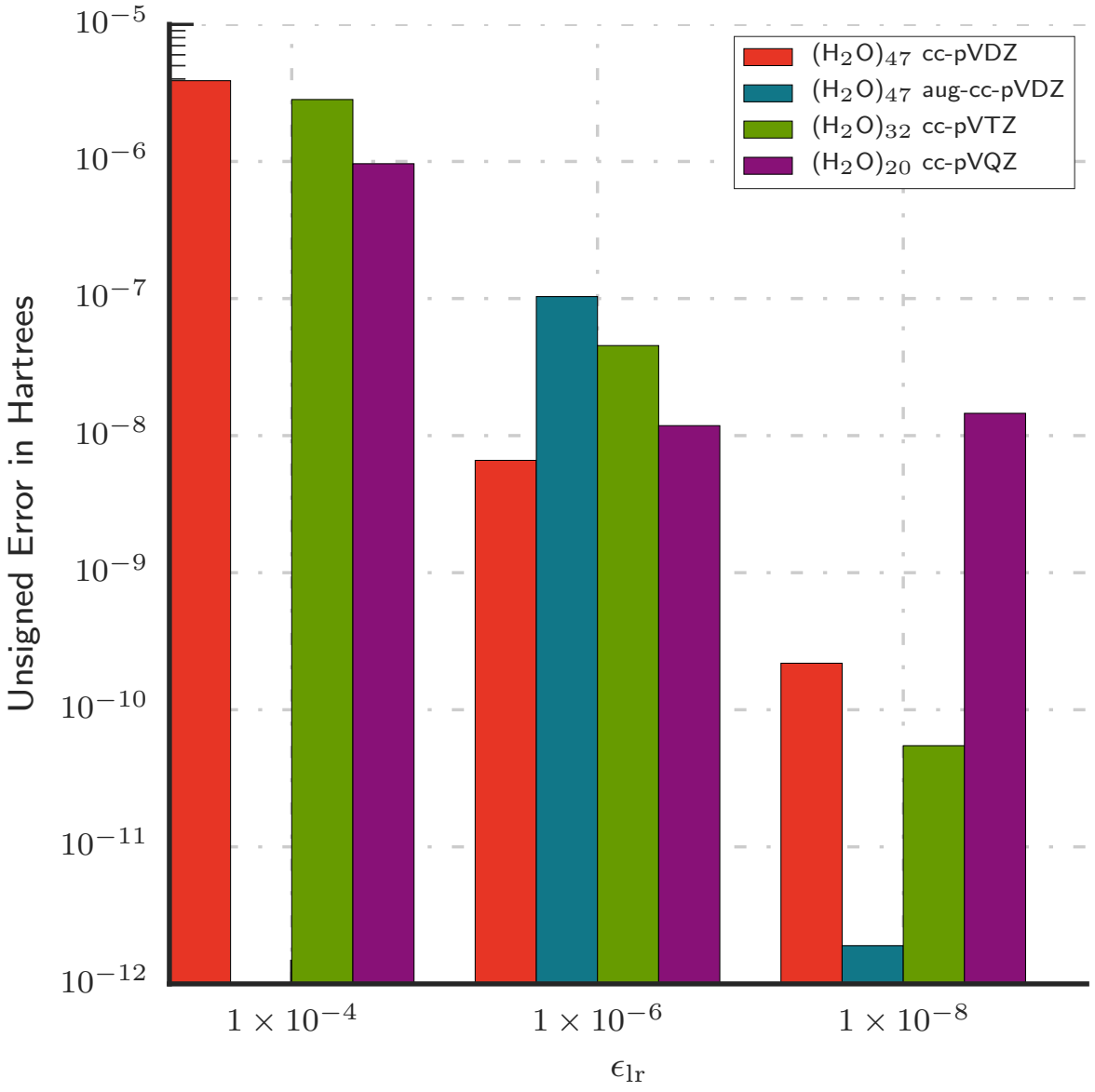

Figure 2.7: Errors in DF-MP2 energy for various water clusters. $\epsilon_{\mathrm{sp}} = 10^{-15}$ for all calculations, and the reference energy used $\epsilon_{\mathrm{lr}} = 0$. There was no CLR approximation used in the DF-MP2 calculation, thus the error comes from error in the CLR-DF-K Fock matrix. The $\epsilon_{\mathrm{lr}} = 10^{-4}$ $(H_2O)_{47}$ aug-cc-pVDZ data point is omitted since the SCF did not converge.

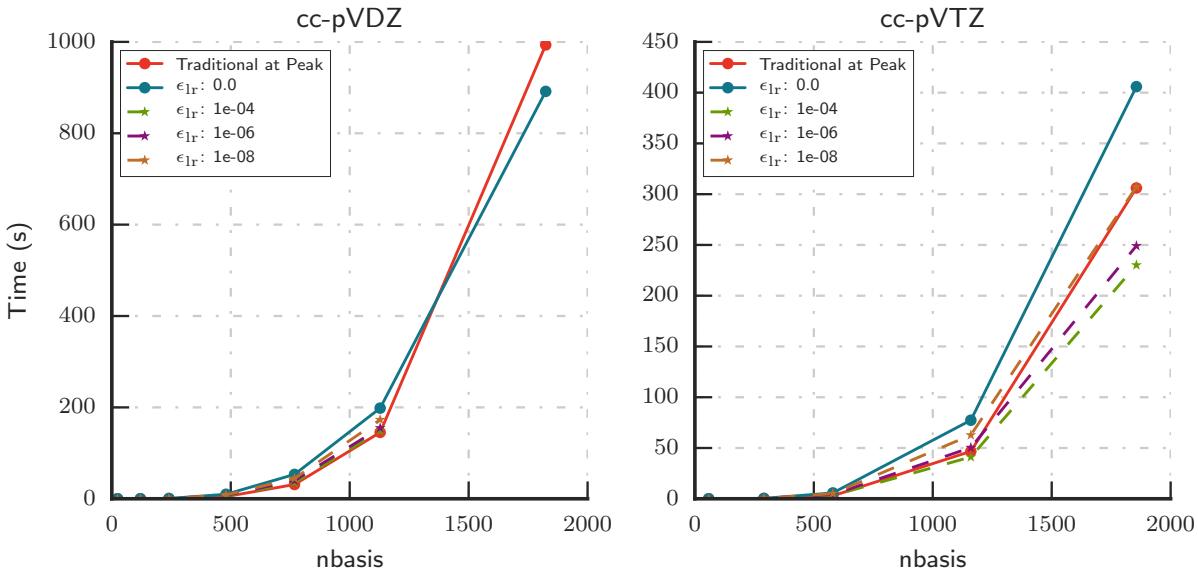

Figure 2.8: Time of CLR-DF-K for water clusters versus a hypothetical traditional $\mathcal{O}(N^4)$ DF-K running at a measured machine peak of 32.5 Gflops for a single core. $\epsilon_{\text{sp}} = 10^{-11}$.

The next most important factor to consider is the tile size. Varying the tile size will not only change the amount of data sparsity in the tensors, but also severely affect the actual throughput of linear algebra operations on the tiles. Due to the complexity of the analysis, we did not attempt to optimize the tile size here and leave it to the future. Nevertheless, to convince readers that there is substantial room for optimization by varying tile size we performed a simple test in which the number of blocks in each dimension is varied, shown in Fig. 2.9. The data demonstrated that varying the blocking scheme can have a large effect on the performance of this CLR-DF-K implementation.

Another performance consideration is the amenability of an algorithm to parallel execution on multiple CPUs. We tested the scalability of our CLR-DF-K implementation with respect to the number of threads on a single multicore node, shown in Fig. 2.10, and on multiple nodes of a distributed memory cluster, shown in Fig. 2.11. The computations were performed on 20- and 32-water clusters in the aug-cc-pVTZ basis, respectively. This larger basis was

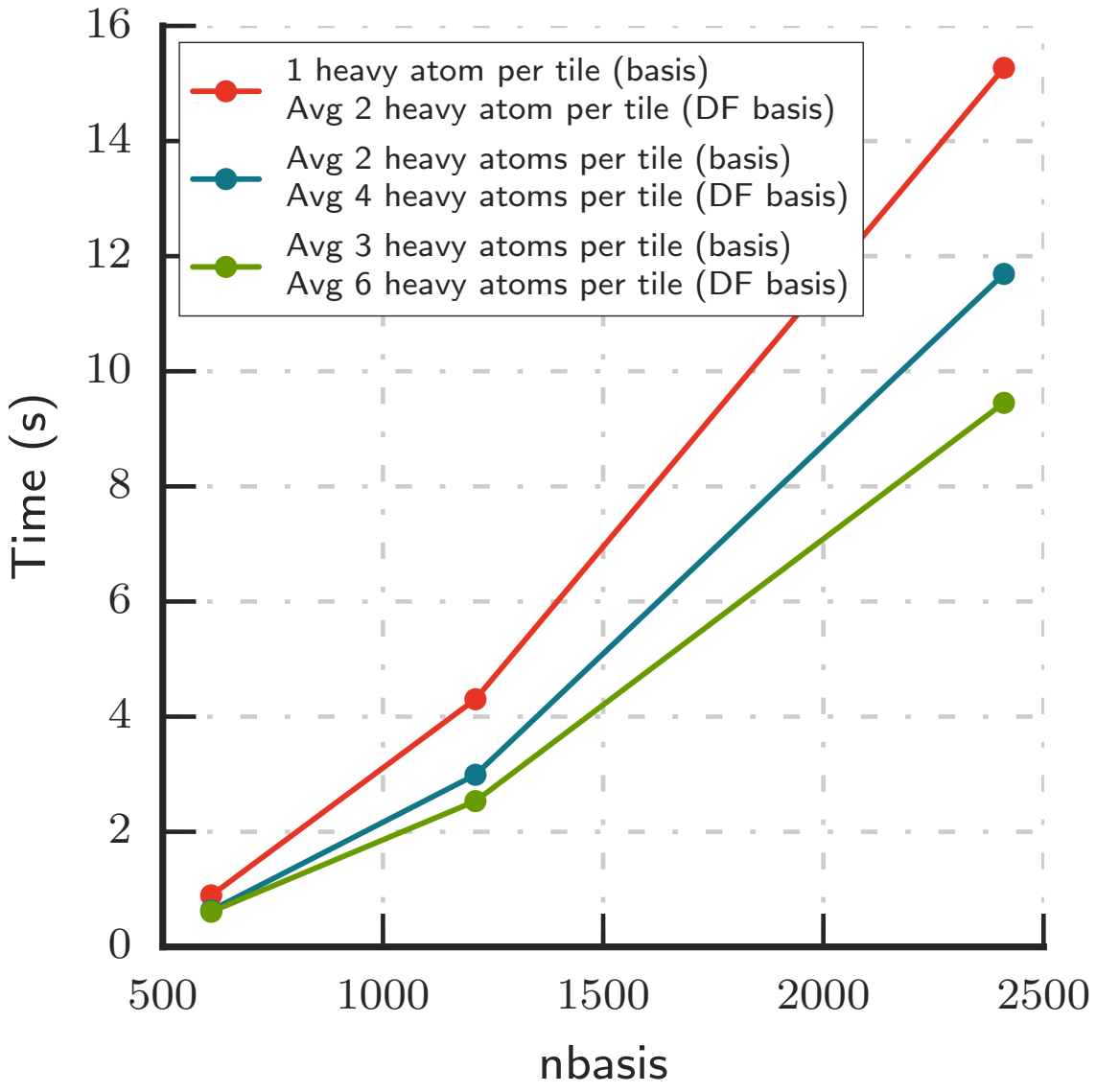

Figure 2.9: Comparison between the number of heavy atoms per block for the orbital and auxiliary (DF) basis for n-alkanes in cc-pVDZ. These calculations were computed with 24 threads using $\epsilon_{\text{sp}} = 10^{-10}$ and $\epsilon_{\text{lr}} = 0$. When there is 1 heavy atom per block the blocking is uniform. When there is more than one heavy atom the blocking is non-uniform so the average number of heavy atoms per block is reported.

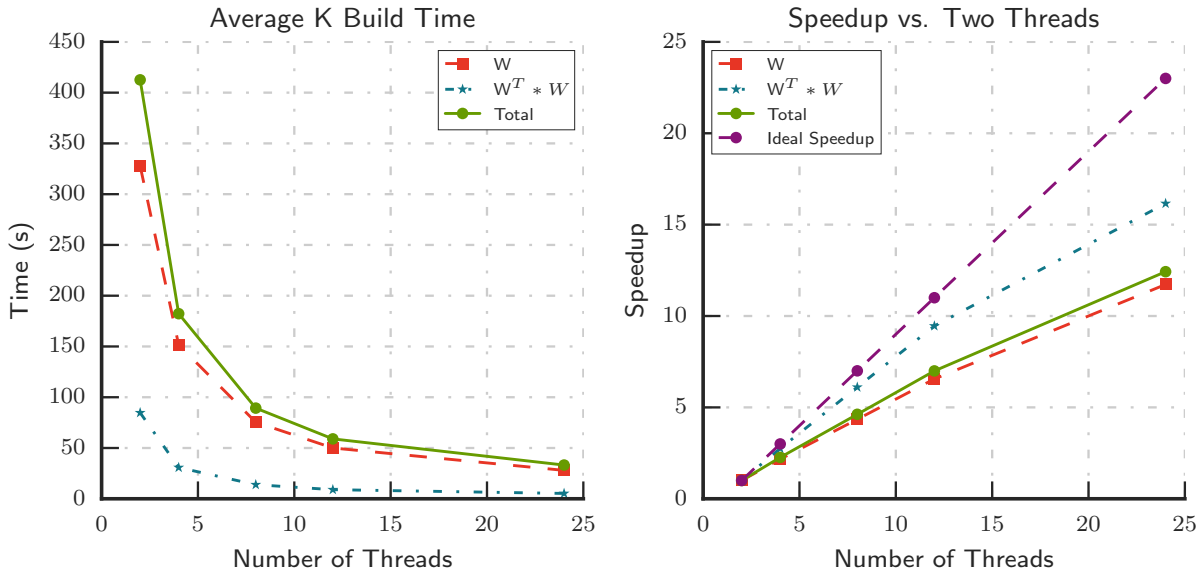

Figure 2.10: Performance versus number of threads for 20 waters in aug-cc-pVTZ basis. Note that the thread scaling begins at 2 (see text). The calculations were run using $\epsilon_{\text{sp}} = 10^{-11}$ and $\epsilon_{\text{lr}} = 0$.

chosen not to artificially improve scaling performance, but instead to provide performance estimates for traditionally expensive calculations.

Parallelism was obtained from the TiledArray library, using TBB for intranode tasks and the message passing interface (MPI) for internode communication in an MPI+X fashion. For the single node scaling calculation, we use 2 threads as a baseline because we have assumed a model where 1 thread is exclusively used to schedule work, with perfect task scaling this would lead to a maximum of 23 times speedup when using 24 threads versus 2 threads. For the multiple node calculations a 2 node baseline is used because the 32 water aug-cc-pVTZ calculation without CLR compression could not fit in 512 GB of memory. The strong scaling in Fig. 2.11 shows that communication overhead does not lead to negative performance until we reach one node per water molecule! Finally, the improved speedup when CLR compression is used, seen in Fig. 2.11, can be attributed to the decrease in internode communication due

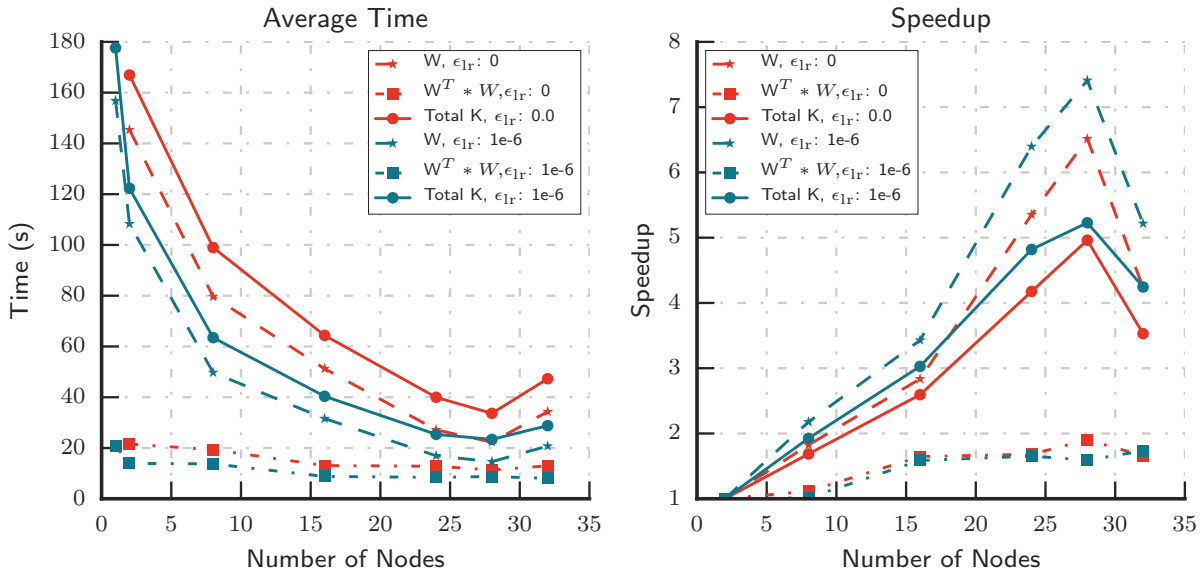

Figure 2.11: Multiple node times and speedup of 32 waters in aug-cc-pVTZ. The calculations were run using $\epsilon_{\text{sp}} = 10^{-11}$. When $\epsilon_{\text{lr}} \neq 0$ rounded addition was used in computation of $\mathbf{W}$.

to block compression.

2.4 Conclusions and Perspective

In this work we introduced the Clustered Low Rank (CLR) framework for block-sparse and block-low-rank tensor representation and computation. Use of the CLR format for the order-2 and order-3 tensors that appear in the context of density-fitting-based Hartree-Fock exchange significantly reduced the storage and computational complexities below their standard $\mathcal{O}(N^3)$ and $\mathcal{O}(N^4)$ figures. *Even for relatively small systems* CLR-DF-K becomes more efficient than the standard DF approach while negligibly affecting molecular energies and properties. Although the apparent cost complexity of CLR-DF-K is higher than that of the standard LinK method for computing the exchange matrix, CLR-DF-K is faster for small and medium size systems, especially when higher-zeta and diffuse basis sets are required.

Making CLR-DF-K asymptotically competitive with LinK should be possible by combining it with local density-fitting ideas.

The entire computation framework that we described here depends on 2 parameters that control precision. One controls the block rank truncation and the other controls screening of small contributions in arithmetic operations on CLR tensors; as they approach 0 the CLR arithmetic becomes exact. There are no other ad-hoc heuristics, such as domains.

This is an initial application of the CLR format, and many significant optimization opportunities remain to be explored. Nevertheless, the efficiency of CLR-DF-K method immediately makes it useful on its own and as a building block for other reduced scaling methods. For example, the fast CLR-DF-K methodology should also be immediately applicable in other contexts where density fitting is key, e.g. the reduced scaling electron correlation methods. We should note that the CLR framework should be naturally beneficial for massive parallelism necessary for ab initio dynamics, since the CLR data compression should reduce the traffic through the memory hierarchy, whether between memory tiers in the next generation of "accelerators" or through the network in a cluster. Previously we demonstrated that CLR can improve scaling with respect to the number of processors used relative to the dense formulation by computing inverse square roots of the Coulomb Metric.[99]

Although it was not demonstrated in this work, it is trivial to also use CLR for construction of the Coulomb contribution to the Fock matrix (CLR-DF-J). CLR-DF-J was only avoided in order to isolate errors to the exchange term, making a direct comparison to LinK possible.

Last, but not least, it is exciting to imagine uses of the CLR framework for exploiting the data sparsity in other tensors that appear in electronic structure and related fields, such as the wave function projections (e.g. cluster amplitudes), density matrices and Green's functions. Some work along these lines is already underway.

2.5 Acknowledgements

We would like to thank Mr. Fabijan Pavošević for useful discussions. The authors also acknowledge Advanced Research Computing at Virginia Tech (www.arc.vt.edu) for providing computational resources and technical support that have contributed to the results reported within this paper. Finally, we acknowledge the support by the U.S. National Science Foundation (grants CHE-1362655 and ACI-1450262).

Chapter 3

Linear Scaling Concentric Atomic Density Fitting

3.1 Introduction

The Hartree-Fock approximation is the basic building block of modern wavefunction methods in electronic-structure theory. The Fock matrix, in closed shell form, is computed, with formal complexity $\mathcal{O}(n^4)$, via the equation:

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\rho\sigma} D_{\rho\sigma} \left[2(\mu\nu|\rho\sigma) - (\mu\rho|\nu\sigma)\right], \tag{3.1}$$

with $\mathbf{D}$ as the Hartree-Fock density and $(\mu\nu|\rho\sigma)$ as the four-center two-electron integrals. Over the years, significant effort has lead to efficient implementations based on avoiding, or screening, computation of small magnitude four-center two-electron integrals reducing the computational complexity of the Fock matrix formation to the often cited $\mathcal{O}(n^2)$.[91, 100] Further work has succeeded in reducing this complexity to $\mathcal{O}(n)$ by splitting the formation of the Fock matrix into a Coulomb ($\mathbf{J}$) part:

$$J_{\mu\nu} = \sum_{\rho\sigma} (\mu\nu|\rho\sigma) D_{\rho\sigma}, \tag{3.2}$$

that can take advantage of fast multipole methods.[101] And an Exchange ($\mathbf{K}$) part:

$$K_{\mu\nu} = \sum_{\rho\sigma} (\mu\rho|\nu\sigma) D_{\rho\sigma}, \tag{3.3}$$

with the most well known $\mathcal{O}(n)$ approaches being Linear scaling exchange (LinK)[56] and Order-N exchange (ONX).[57, 102]

An alternative approach for the efficient construction of the Fock matrix is to factorize the four-center integrals into two three-index tensors. This strategy, most commonly referred to as resolution of the identity (RI) or density fitting (DF), takes the integral ket $|\rho\sigma)$, which

contains products having basis functions on up to two centers and fits it within a basis of single center functions:

$$|\rho\sigma) \approx \sum_X C^X_{\rho\sigma}|X), \tag{3.4}$$

where $\mathbf{C}$ is the tensor of fitting coefficients and the X functions come from some auxiliary basis. Approximations of this type have been in use since the 1950s,[77, 78, 103, 104, 105, 106] but became more mainstream with the development of optimized auxiliary basis sets[96, 107, 108, 109] and demonstrations that the performance of both SCF methods[80] and correlation methods[79] could be improved with acceptable loss of accuracy. We can solve for the fitting coefficients via projection onto the auxiliary basis:

$$(Y|\rho\sigma) = \sum_X C^X_{\rho\sigma}(Y|X), \tag{3.5}$$

and by rewriting this as a tensor expression (summation notation will be used from this point forward), with $E^Y_{\rho\sigma} = (Y|\rho\sigma)$ and $M_{YX} = (Y|X)$ we find

$$E^Y_{\rho\sigma} = C^X_{\rho\sigma} M_{YX} \tag{3.6}$$

$$M^{-1}_{YX} E^Y_{\rho\sigma} = C^X_{\rho\sigma}. \tag{3.7}$$

Usually, $\mathbf{M}^{-1}$ is decomposed such that $\mathbf{M}^{-1} = \mathbf{Z}\mathbf{Z}^\dagger$, where $\mathbf{Z}$ is either the Cholesky decomposition or the square root of $\mathbf{M}^{-1}$. The reconstruction of the $(\mu\nu|\rho\sigma)$ tensor then becomes:

$$(\mu\nu|\rho\sigma) \approx E^X_{\mu\nu} M^{-1}_{XW} M_{WR} M^{-1}_{RY} E^Y_{\rho\sigma} \tag{3.8}$$

$$\approx E^X_{\mu\nu} Z_{XQ} Z^\dagger_{QW} M_{WR} Z_{RP} Z^\dagger_{PY} E^Y_{\rho\sigma} \tag{3.9}$$

$$\approx E^X_{\mu\nu} Z_{XS} Z^\dagger_{SY} E^Y_{\rho\sigma} \tag{3.10}$$

This decomposition of $(\mu\nu|\rho\sigma)$ allows for a small prefactor $\mathcal{O}(n^3)$ implementation of Equation (3.2) (DF-J). But Equation (3.3) (DF-K) formally remains $\mathcal{O}(n^4)$. Even so, in small molecules with large and/or diffuse basis sets the density fitting approach to exchange (DF-K) is competitive with LinK.

To decrease both the memory and computational complexity of DF-K the number of coefficients used to fit each $\rho\sigma$ product must become constant with respect to the number of atoms in the system. This is accomplished by constructing $\mathbf{C}$ from functions that are nearby or local to the $\rho\sigma$ product, called local density fitting (LDF). In LDF $\mathbf{C}$ is constructed from a restricted set of functions in a way similar to Equation (3.4):

$$(Y_{\text{local}}|\rho\sigma) = C_{\rho\sigma}^{X_{\text{local}}} M_{X_{\text{local}}Y_{\text{local}}}, \tag{3.11}$$

where the subset of functions in the auxiliary indices X_{local} and Y_{local} depends in some approximation specific way on ρ and σ.

It is well known that fitting with the Coulomb metric yields errors in $(\mu\nu|\rho\sigma)$ that are second order with respect to errors in $\mathbf{C}$,[81] but the use of the Coulomb metric leads to slow decay with respect to the distance between centers in $\mathbf{E}$, but not necessarily in $\mathbf{C}$.[110, 111] This slow decay makes reducing the computational complexity of Equation (3.7) below $\mathcal{O}(n^3)$ difficult. To overcome this limitation a number of techniques have been explored: the use of metrics with more rapid decay,[60, 78, 110, 112] using the Coulomb metric with auxiliary functions coming from a local domain,[85, 90, 112] and an extreme version of domains that limits the auxiliary functions to the atoms on which functions ρ and σ sit ,[62, 78, 87, 113, 114, 115, 116, 117, 118] herein called concentric atomic density fitting (CADF) and elsewhere known as pair atomic resolution of the identity (PARI).

In CADF, $\mathbf{C}$ becomes sparse by construction and is found by solving:

$$(Y_{\in\{a,b\}}|\rho_a\sigma_b) = C_{\rho_a\sigma_b}^{X_{\in\{a,b\}}} M_{X_{\in\{a,b\}}Y_{\in\{a,b\}}}, \tag{3.12}$$

where a designates the center for function ρ and b for function σ. Constructing $\mathbf{C}$ using the CADF definition of locality proves to be a very inaccurate approximation[62] necessitating the use of Dunlap's robust fitting:[81]

$$(\mu\nu|\rho\sigma) \approx E_{\mu\nu}^{X} C_{\rho\sigma}^{X_{cd}} + C_{\mu\nu}^{Y_{ab}} E_{\rho\sigma}^{Y} - C_{\mu\nu}^{R_{ab}} M_{RP} C_{\rho\sigma}^{P_{cd}}, \tag{3.13}$$

where subscripts ab and cd designate a local fit (from this point forward we will assume that robust fitting is used). The accuracy of CADF is well established[62, 114, 116] and ref. [117] shows errors in the range of $1\frac{\text{cal}}{\text{mol}}$ per electron for the Rx25 data set in a cc-pVQZ basis, when used for exchange only. But unfortunately robust fitting can suffer from a loss of positive semi-definiteness in the approximate $(\mu\nu|\rho\sigma)$ allowing the SCF procedure to converge to unphysical densities and attractive electrons.[114] Some options to mitigate this behavior are discussed in references [114] and [62]. Alternatively, CADF can be used only for the computation of Hartree-Fock exchange (CADF-K), avoiding these issues.[116] With only basic integral screening the computational complexity of CADF-K is $\mathcal{O}(n^3)$ in each SCF iteration. While CADF-K has a reduced prefactor relative to traditional four-center exchange builds, it requires further approximation to achieve complexity $\mathcal{O}(n)$. There has been some effort to devise a $\mathcal{O}(n)$-CADF-K algorithm in our group, ref. [118], and others, ref. [115], in this work we will take an approach similar to the latter that rapidly achieves reduced scaling at only a small loss of accuracy relative to $\mathcal{O}(n^3)$ CADF-K. Because the tensors involved in CADF-K are generally sparse, except for the three-center two-electron integrals, we will use techniques developed for sparse tensor algebra to achieve reduced scaling. To handle the

lack of sparsity in the three-center integrals we will use an approach akin to the sparse maps of linear scaling correlation methods[41, 42, 119, 120, 121] and the methods discussed in ref. [115].

Exploiting approximate element and block-wise sparsity is an active area of research in quantum chemistry. For the remainder of the work, when we refer to sparsity we will assume block-wise sparsity, exploiting element-wise sparsity has been attempted in the context of electronic-structure,[122, 123, 124] but for performance reasons the majority of work focuses on the block-wise variant. A dizzying array of approaches for sparse matrix multiplication (SpGEMM), an operation which is sufficient to compute tensor contractions, are currently under development, including but not limited to: no output truncation (no approximation is made for the given inputs),[124, 125, 126] truncation of blocks with small norms based on a threshold parameter,[66] radius based criteria for functions with well defined locality,[35, 127, 128, 129] output norm estimation to avoid computing blocks that will be known to have small norm,[99, 130] avoiding small contributions to significant result blocks,[131, 132, 133] an idea that appears in integral driven construction of the Fock matrix.[56, 91, 100, 134] Multiple approaches, such as skipping small contributions and using a radius, can be combined[135] and also these techniques can be used in conjunction with advanced data structures[67, 136, 137, 138, 139] to reduce the computational complexity required even further. Other strategies, many of which arise naturally in reduced scaling correlation methods, define tensor sparsity using a variety of criteria.[42, 115, 140, 141, 142] Finally, novel methods to gain performance have been explored.[130, 143]

Our sparse tensor library described in Section 4.3 uses a threshold ϵ_{sp} such that a block $\mathbf{B}$ is treated as zero if:

$$\epsilon_{\text{sp}}\text{Vol}(\mathbf{B}) > ||\mathbf{B}||_F, \tag{3.14}$$

where Vol($\mathbf{B}$) is the number of elements in block $\mathbf{B}$ and $||||_F$ is the Frobenius norm. A more thorough overview of our sparse tensor library is outside the scope of this work, its general performance is discussed in references [130], [99], and [144]. In this work, we will only discuss performance with respect to CADF-K.

The remainder of the paper is organized as follows: in Section 3.2 we discuss our algorithm for $\mathcal{O}(n)$ CADF-K, Section 4.3 provides some implementation details, Section 4.5 provides both timings and accuracy for our method relative to traditional CADF-K, and finally Section 4.6 relays our comments on the successes and failures of our approach.

3.2 Concentric Atomic Density Fitting Exchange

3.2.1 Algorithm

We tested an algorithm for $\mathcal{O}(n)$-CADF-K bases on that of Manzer et al.[116]

Algorithm 1 CADF-K using localized MOs

procedure CADF-K($\mathbf{C}$, $\bar{\mathbf{C}}$) ▷ Compute $\mathbf{K}$ from the fitting coefficients and the LMOs, $\bar{\mathbf{C}}$

$Z^X_{\mu i} = C^X_{\mu\rho}\bar{C}_{\rho i}$

$F^X_{\nu i} = \mathcal{F}\left(E^X_{\nu\sigma}\bar{C}_{\sigma i}, \mathbf{Z}\right) - \mathcal{F}\left(0.5 M_{XY} Z^Y_{\nu i}, \mathbf{Z}\right)$

$L_{\mu\nu} = Z^X_{\mu i} F^X_{\nu i}$

$K_{\mu\nu} = \frac{1}{2}\left(L_{\mu\nu} + L_{\nu\mu}\right)$

return K

Our changes include the use of localized molecular orbitals (LMO) and a screening function $\mathcal{F}$, which can be set to the identity function to reproduce the original algorithm.

Algorithm 2 Identity function for $\mathcal{F}$

1: **procedure** $\mathcal{F}(\mathbf{Q}, \mathbf{Z})$ ▷ **Q** is an expression for a three-index tensor, which will be screened by **Z**
2: **for all** a, b, c **do**
3: Compute and store Q^a_{bc} in a normal fashion
4: **return Q**

Manzer's procedure simplifies the construction of **K** yielding promising results relative to LinK for linear n-alkanes in the cc-pVTZ basis set. The downside is that this approach cannot easily scale below $\mathcal{O}(n^2)$ due to term 3 in Algorithm 1 where the three-centered integrals are needed, and in practice they report approximately $\mathcal{O}(n^3)$ for their algorithm using canonical molecular orbitals.

3.2.2 Contraction Screening

If we set $\mathcal{F}$ to the identity (Algorithm 2) then in Algorithm 1 the most expensive step is the construction of **F**. In order to achieve reduced scaling we need to find a way to compute only a linear number of blocks of **F**. Since sparse tensor algebra alone cannot achieve this, due to the contraction of **E** with the LMOs, we must look at the only tensor expression in which **F** is used,

$$L_{\mu\nu} = Z^X_{\mu i} F^X_{\nu i}, \tag{3.15}$$

to try and reduce the cost of construction of **F**. Here we propose a method to limit the construction of **F** to only compute blocks in which X and ν are "near" i. It turns out that the natural sparsity of **F** is sufficient to ensure that ν is close to molecular orbital i, but

due to the $\frac{1}{r}$ decay between X and νi in $\mathbf{E}$ the number of significant blocks of $\mathbf{F}$ will decay slowly. On the other hand $\mathbf{Z}$ only includes X near i by construction, thus we propose using the structure of $\mathbf{Z}$ to screen which blocks of $\mathbf{F}$ are computed. To do so we make lists of all significant X and μ blocks for a given group of i in $\mathbf{Z}$ according to:

$$||Z^X_{\mu i}||_F \geq \epsilon_{\mathrm{f}}. \tag{3.16}$$

We then only compute a block $F^X_{\nu i}$ if either X or ν is in our list of indices close to i. Using this idea, our $\mathcal{O}(n)$ algorithm for $\mathcal{F}$ is given in Algorithm 3.

Algorithm 3 Screening function for $\mathcal{F}$

procedure $\mathcal{F}(\mathbf{Q}, \mathbf{Z})$ ▷ $\mathbf{Q}$ is an expression for a three-index tensor, which will be screened by $\mathbf{Z}$
  $x = \{\}$ ▷ Set for important blocks in X
  $n = \{\}$ ▷ Set for important blocks in ν
  for all c **do**
    for all a, b **do**
      if $||Z^a_{bc}||_F \geq \epsilon_{\mathrm{f}}$ **then**
        Add a to x
        Add b to n
  for all c **do** ▷ Construct $\mathbf{Q}$
    for all a, b **do**
      if $a \in x$ or $b \in n$ **then**
        Construct block Q^a_{bc}
      else
        Skip Q^a_{bc}
  return Q

While the motivation for this approach is based on selecting only functions "near" to a shared index, in practice it is much easier to explain algebraically. Looking at Equation (3.15) we see that our idea to screen the construction of $\mathbf{F}$ based on "nearness" is the same as only computing those blocks of $F^X_{\nu i}$ which are multiplied by at least one block of $\mathbf{Z}$ with norm greater than ϵ_{f}. Essentially, we avoid computation of blocks of $\mathbf{F}$ that only interact with nearly negligible blocks of $\mathbf{Z}$. While this procedure does not provide rigorous bounds on the norm error for $\mathbf{L}$, because the magnitude of the avoided blocks in $\mathbf{F}$ is never taken into account, in practice choosing a sufficiently small ϵ_{f} should lead to good accuracy and as ϵ_{f} approaches 0 this method becomes exact within the CADF-K framework.

Finally, we make one additional approximation, we apply an additional truncation parameter to the LMOs, where if the Frobenius norm of a block of the LMOs is smaller than ϵ_{c} that block is treated as zero.

3.3 Implementation

3.3.1 Software

Our implementation of these ideas took place in a developmental version of the Massively Parallel Quantum Chemistry (MPQC) package,[1] which takes advantage of the sparse tensor algebra provided by the library developed in our group TILEDARRAY.[93] A version of LinK was developed for comparison based on the hartree-fock++ example from the Libint2 integral library.[92, 145]

In principle the algorithm that we outlined in Section 3.2 would compute the significant blocks of $\mathbf{E}$ on demand, which would minimize storage at the cost of some computation on

[1] An older version of the project can be found at https://github.com/ValeevGroup/mpqc

every SCF iteration. Due to a technical limitation of the TiledArray library with regards to direct computation we could not easily implement a $\mathcal{O}(n)$ direct version of our algorithm. To overcome this limitation we made use of the Clustered Low Rank (CLR) tensor compression technique, described in ref. [130], for storage of $\mathbf{E}$. CLR allows for $\mathbf{E}$ to be computed once, at computational cost $\mathcal{O}(n^2)$, and stored in a compressed fashion that depends on the blocking and on a parameter ϵ_{lr}. While reducing our per-iteration prefactor the substitution of CLR should not change the scaling of our method relative to direct computation.

Throughout all $\mathcal{O}(n)$-CADF-K calculations we used the sparse threshold $\epsilon_{\text{sp}} = 10^{-11}$, Our block were built by taking the functions from each hydrogen and combined them with the functions from the nearest heavy atom (LMOs were blocked to have the same number of blocks as the AOs), and we used Foster-Boy's localization for the LMOs.[83, 146] Although any localization providing sparse orbitals should work. It is likely that we could have obtained better performance by using a slightly looser ϵ_{sp} and by searching for the ideal block size for each tensor and dimension, but these optimizations will be left for future publications.

3.3.2 Hardware

All computations were carried out on Virgina Tech's NewRiver system where each node has 2 Xeon E5-2680v3 2.5ghz Intel processors, with 24 cores and available with either 128 GB or 512 GB of RAM. Each node has a theoretical peak performance of 960 Gflops/s with a measured peak of 780 Gflops/s in Intel MKL's DGEMM routine.

3.4 Results

3.4.1 Molecules and Basis Sets

To test our algorithm we chose to use linear n-alkanes, Cartesian geometries generated with OPEN BABEL,[94] and water clusters, Cartesian geometries —taken from random snapshots of molecular dynamics trajectories at ambient conditions[36]— were obtained from ERGOSCF's public repository.[95] Dunning's correlation consistent basis sets were used,[8] with augmented diffuse functions for cc-pVDZ.[8, 147] [2] matching auxiliary basis sets were used for fitting.[96] Unlike some works we did not increase the ζ of the auxiliary basis to help correct errors arising from the local fitting.

3.4.2 CLR Compression

CLR compression, of the three-center two-electron integrals allows for both efficient storage and contraction of the tensor $(X|\mu\nu)$. Figures 3.1 and 3.2 show significant compression of the these integrals, with 92% improvement over block sparsity alone for $C_{120}H_{242}$ in cc-pVDZ with $\epsilon_{\text{lr}} = 10^{-6}$. CLR also provides compression in three dimensional systems with tight and even diffuse basis sets. For $(H_2O)_{47}$ a compression of 67% was obtained in aug-cc-pVDZ and a compression of 95% for $(H_2O)_{139}$ in cc-pVDZ with $\epsilon_{\text{lr}} = 10^{-6}$, relative to block sparsity alone.[3]

[2] We experienced convergence issues with the large n-alkanes in aug-cc-pVDZ, so only tested them on water clusters

[3] CLR is able to achieve higher compression percents for water clusters over linear alkanes because of the comparison to the storage required for block-sparsity. If the comparison was the full size of the dense tensor versus data-sparsity then the alkanes would have a significantly larger compression percentage.

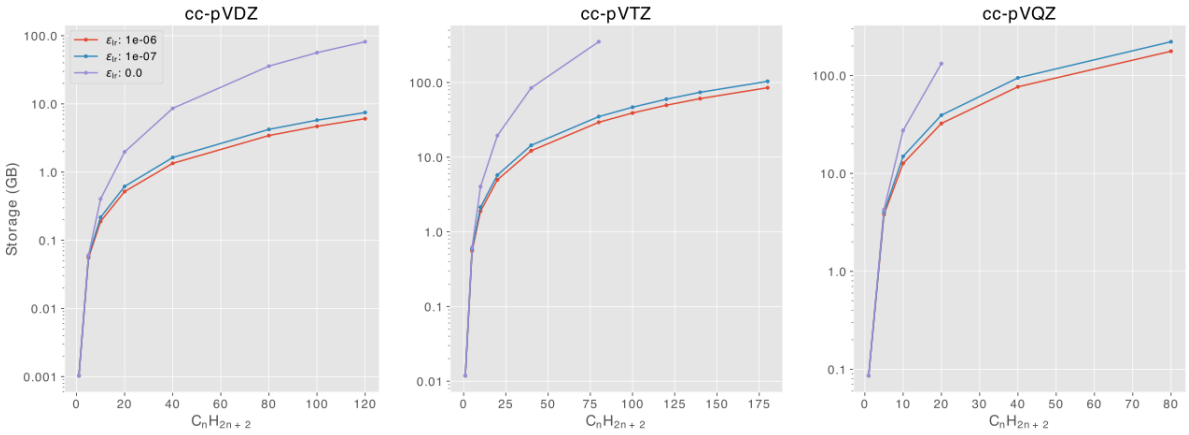

Figure 3.1: n-alkanes CLR storage for $(X|\mu\nu)$ integrals

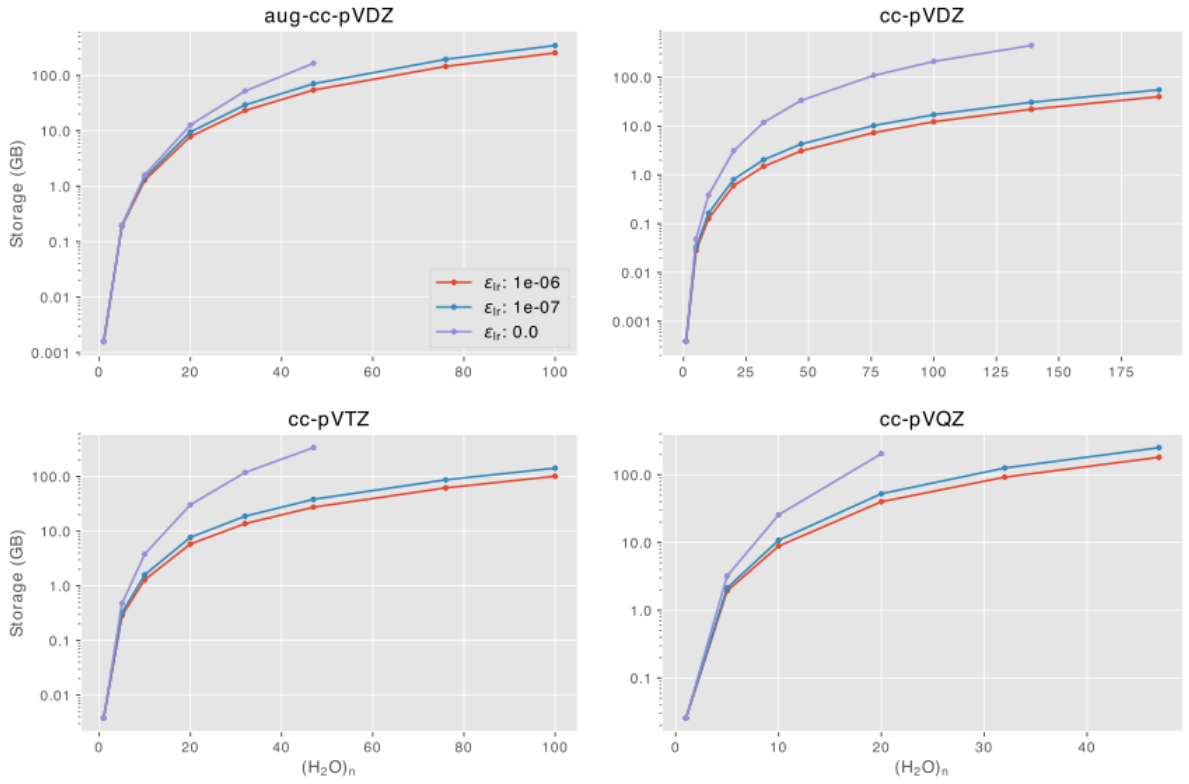

Figure 3.2: Water cluster CLR storage for $(X|\mu\nu)$ integrals

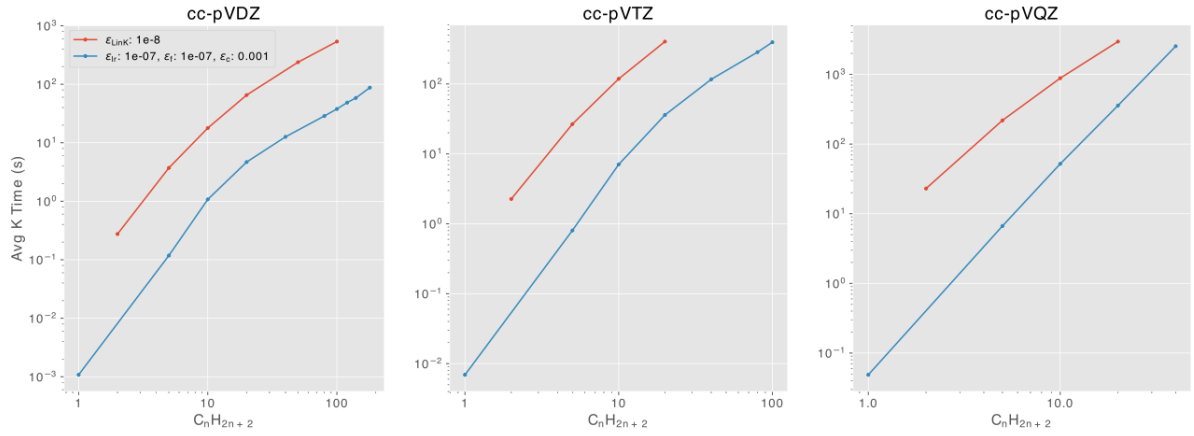

Figure 3.3: n-alkanes timings for $\mathcal{O}(n)$-CADF-K versus LinK. Both methods were run using a single thread

3.4.3 Timings

Single-Thread

Figure 3.3 shows that $\mathcal{O}(n)$-CADF-K achieves linear scaling at approximately the same system sizes as LinK, but does so with a significantly reduced prefactor. For $C_{100}H_{202}$ in the cc-pVDZ basis our $\mathcal{O}(n)$-CADF-K implementation had an average iteration time of 38 seconds and a scaling exponent of 1.3, while LinK, with a screening threshold of 10^{-8}, had an average iteration time of 536 seconds and a scaling exponent of 1.17. The effect is similar for cc-pVTZ.

Figure 3.4 shows that $\mathcal{O}(n)$-CADF-K also nearly matches LinK's scaling with improved prefactor for water clusters. For $(H_2O)_{47}$ in cc-pVTZ $\mathcal{O}(n)$-CADF-K had an average iteration time of 459 seconds with a scaling exponent of 1.75, while LinK, with a threshold of 10^{-8}, had an average iteration time of 1034 seconds and a scaling exponent of 1.90. Similarly, our average iteration times are better than LinK for aug-cc-pVDZ, and cc-pVDZ while having

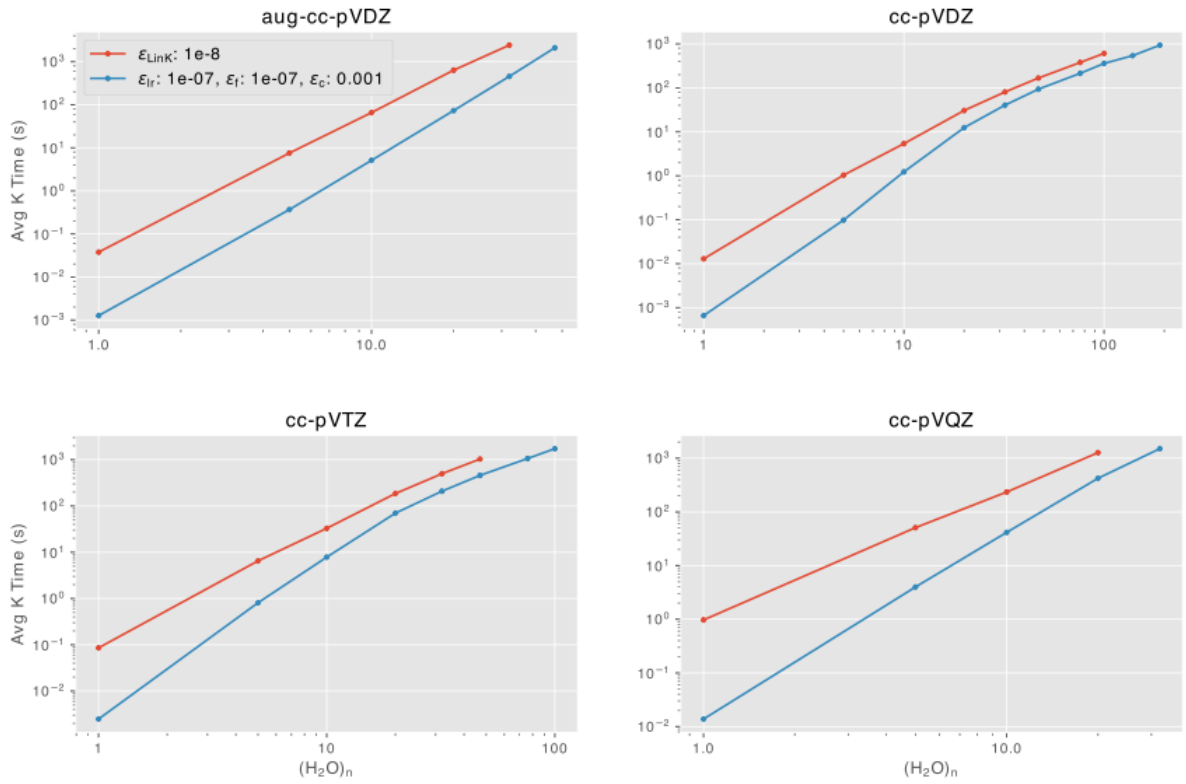

Figure 3.4: Water cluster timings for $\mathcal{O}(n)$-CADF-K versus LinK. Both methods were run using a single thread

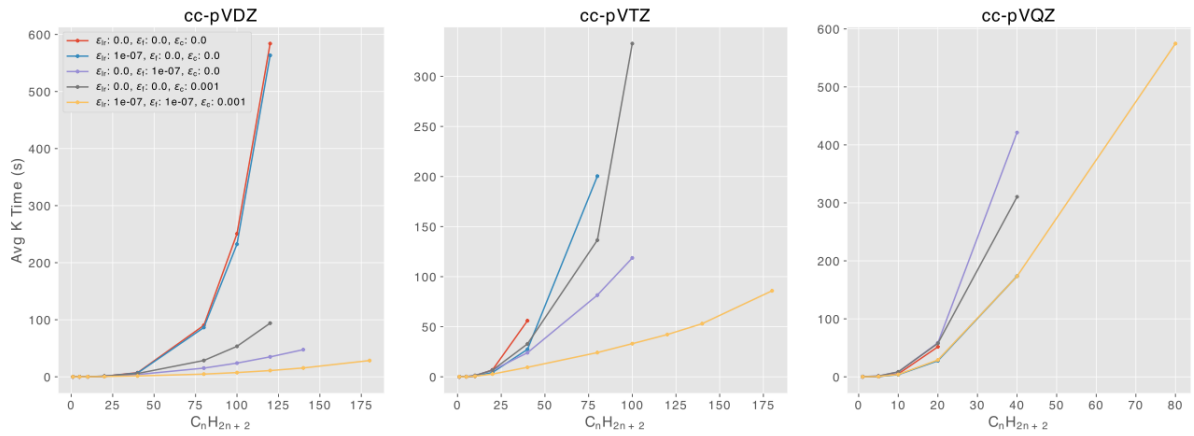

Figure 3.5: n-alkane timings for $\mathcal{O}(n)$-CADF-K at various approximation levels, using 24 threads

roughly the same scaling exponents.

Multi-Thread

Figure 3.5 shows the multi-threaded performance of our $\mathcal{O}(n)$-CADF-K method for n-alkanes at different levels of approximation. Showing that CLR doesn't significantly contribute to the reduced scaling. Almost all of the scaling improvements come from a combination of truncating the LMOs and the special screening of terms —that is the subject of this paper. Figure 3.6 shows that water clusters have similar behavior to the n-alkanes, except that reduced scaling onset is delayed. In cc-pVDZ and cc-pVTZ it is clear that we have obtained near linear behavior and in cc-pVQZ there is a significant reduction of cost, with larger calculations needed to determine when reduced scaling arises. Additionally, Figure 3.6 shows that while we are able to run larger systems in diffuse basis sets we do not approach linear scaling, approximate four-center methods also exhibit this delayed onset of reduced scaling behavior.

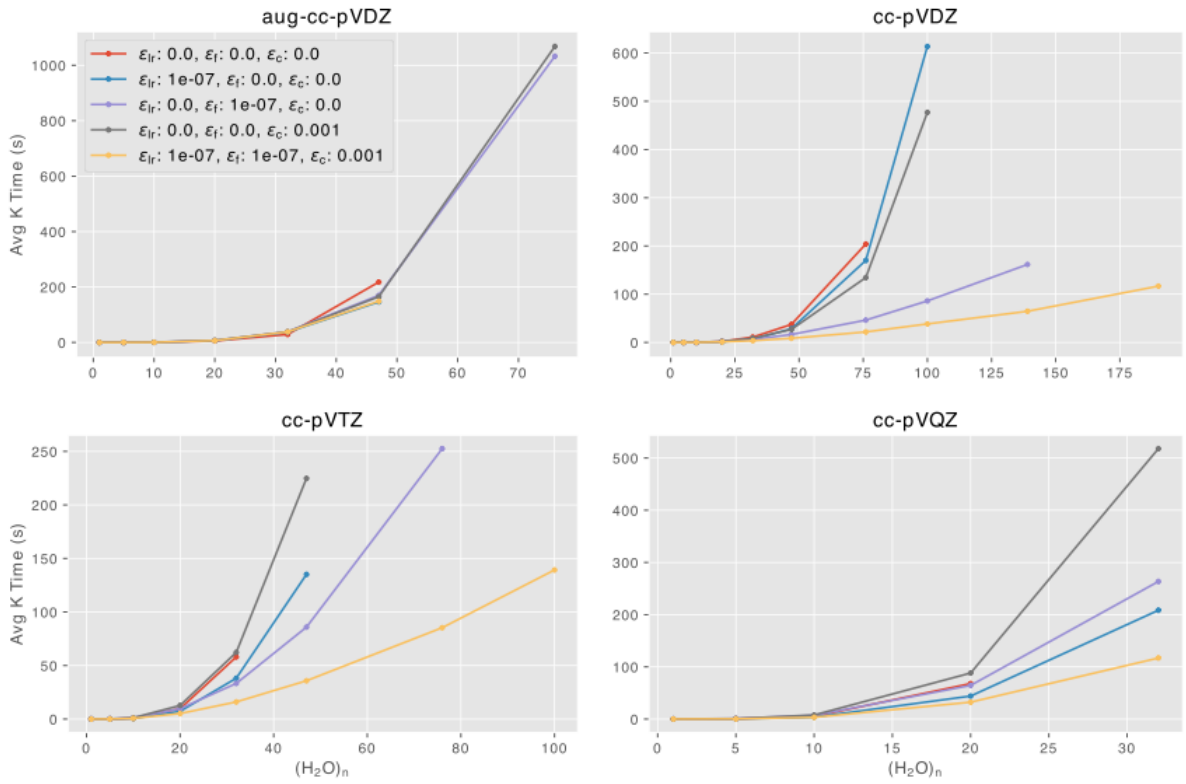

Figure 3.6: Water cluster timings for $\mathcal{O}(n)$-CADF-K at various approximation levels, using 24 threads

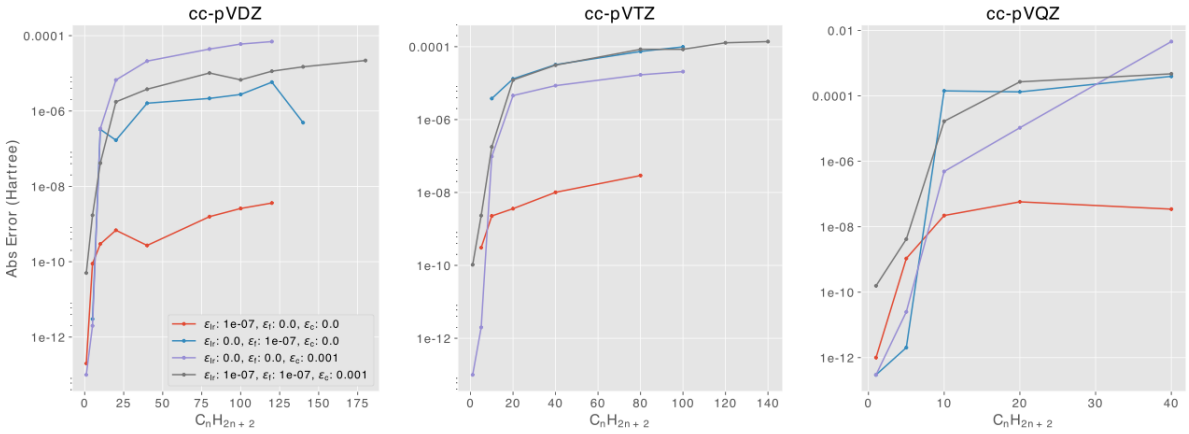

Figure 3.7: n-alkane $\mathcal{O}(n)$-CADF-K errors from the different approximations used

3.4.4 Errors

All errors are computed relative to CADF-K where the only approximations made are Schwarz screening of the three-center two-electron integrals with a threshold of 10^{-12} and application of our sparse tensor library with $\epsilon_{\text{sp}} = 10^{-11}$. Figure 3.7 shows the errors for n-alkanes from each of the approximations we make to CADF-K. For linear alkanes it is obvious that the errors arising from the CLR approximation of $(X|\mu\nu)$ are significantly smaller than the other errors in our reduced scaling method justifying its use in avoiding repeated integral computation. The errors from the other approximations —truncation of the LMOs and our screening approach— are not as easy to untangle. In cc-pVDZ it appears that most of the error comes from truncation of the LMOs, but in cc-pVTZ the errors appear to come from skipping terms in the formation of $\mathbf{F}$. The combined error is not always a simple sum of the other errors suggesting that there is some cancellation of error between the LMO truncation and the screening. Figure 3.8 shows that in water clusters we see a similar effect to the linear alkanes, with the additional observation that the errors grow more rapidly with respect to system size when using aug-cc-pVDZ. Once again a similar trend appears with

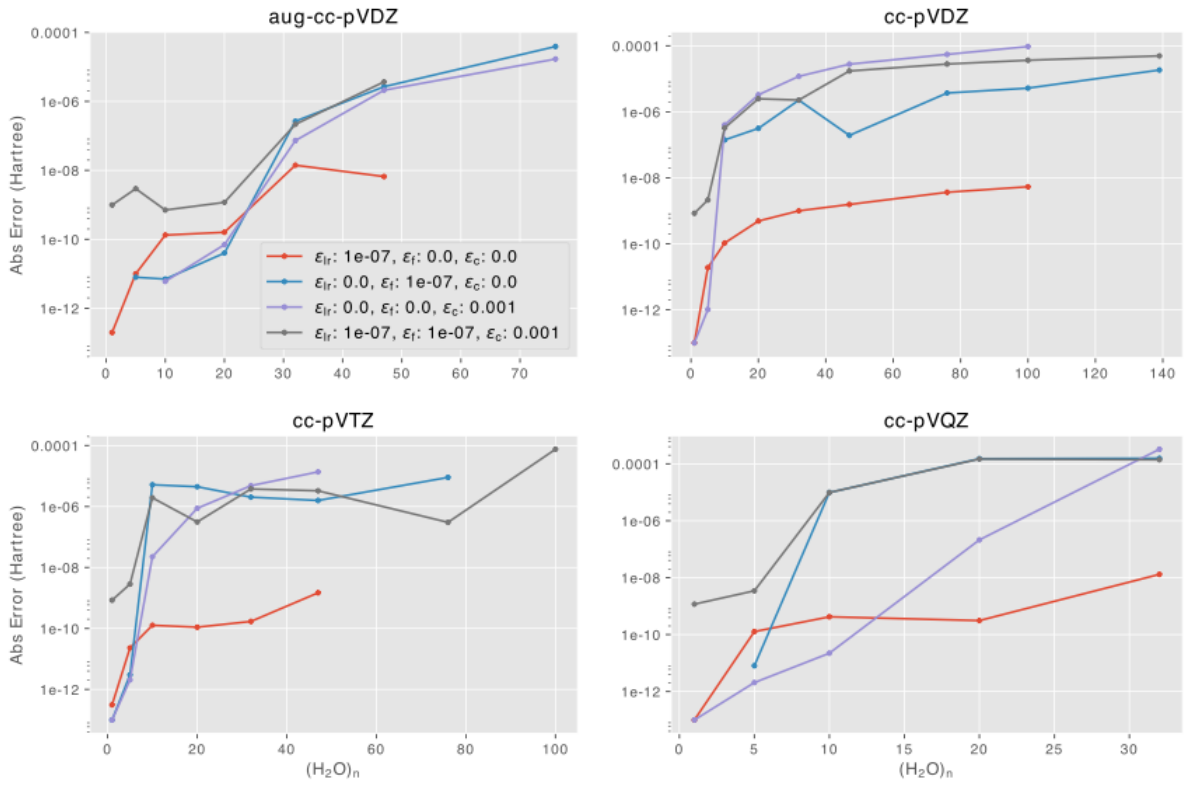

Figure 3.8: Water cluster $\mathcal{O}(n)$-CADF-K errors from the different approximations used

the LMO truncation being larger for cc-pVDZ and it is unclear which effect is larger for cc-pVTZ, although error cancellation still keeps the combined error smaller than the sum of its parts. Finally with the chosen parameters of $\epsilon_{\mathrm{lr}} = 10^{-7}$, $\epsilon_{\mathrm{f}} = 10^{-7}$, and $\epsilon_{\mathrm{c}} = 10^{-3}$, the errors appear to fall around 10^{-4} Hartrees which is less than the errors introduced by the CADF-K method. Also fortuitously the errors appear to be smaller for the smaller basis sets, this is helpful because in these basis sets the rank reduction, obtained from using LMOs, that gives an advantage over density based methods is not as large as in the higher ζ basis sets, necessitating looser thresholds to be competitive with LinK.

3.5 Conclusions and Perspective

In summary, we have developed a linear scaling implementation of Hartree-Fock exchange using the CADF approximation. By combining localized molecular orbitals with the imposed sparsity of the CADF fitting coefficients and screening intermediate contractions we have demonstrated similar scaling to LinK with a significantly reduced prefactor. Errors, while large, are below the method errors in CADF-K and are also improvable by tightening the approximation parameters. CLR compression for the three-center two-electron integrals was used in this study to demonstrate that the CLR approximation can usefully allow for the storage of large integral tensors and also provides some efficiency improvement in their contraction, but the use of CLR was not necessary to achieve linear scaling—a reasonable direct integral approach would also have worked.

Future work entails testing the method on larger systems in high-zeta ($\zeta \geq 4$) basis sets and attempting to improve the errors from truncation of the LMOs. Ideas developed in this paper are now being used for efficient construction of exchange in periodic systems and in construction of approximate atomic orbital basis integrals in reduced scaling electron

correlation methods. We plan on publishing these results soon.

3.6 Acknowledgements

We would like to thank Fabijan Pavošević and Xiao Wang for useful discussions. Also the authors acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: http://www.arc.vt.edu

Chapter 4

Linear Scaling Parallel Pair Natural Orbital MP2

4.1 Introduction

Density functional theory (DFT) is the computational workhorse in the domains of chemistry and material science due to its acceptable quantitative accuracy and low cost.[148] However, the lack of systematic improvability of DFT prevents its use for high accuracy predictions. Many-body electronic structure methods, such as the coupled-cluster method[10], are, on the other hand, systematically improvable but suffer from unacceptably high complexity with respect to size and precision, which limit their applications to small molecules. For example, the coupled cluster singles, doubles, and perturbative triples (CCSD(T)), routinely described as the gold standard of quantum chemistry, has a $\mathcal{O}\left(n^7\epsilon^{-1/4}\right)$ computational complexity in LCAO representation, where n is proportional to system size and ϵ is the target precision. While the use of explicitly correlated (F12) LCAO formulations[149, 150, 151] have largely alleviated the precision problem, the steep size scaling still limits the use of CCSD(T) method to systems with up to 20 atoms on a single computer, and up to 100 atoms on a supercomputer.

Reduction of cost and scaling with size has been an area of much recent progress. Complexity reduction approaches can be broadly classified in one of three groups. The first group includes so-called divide-and-conquer methods, in which the grand problem is split into subproblems by partitioning the original problem's space. Examples include the divide-and-conquer method,[152] various embedding techniques (QM/MM,[153, 154] QM/QM[155, 156]) , and various local correlation techniques;[142, 157, 158, 159, 160] each of these generic realizations are further specialized by the numerous technical details, such as how the subproblems are coupled to each other, whether the partitioning is refined adaptively, etc. Another group of approaches approximate the energy and other properties as a truncated many-body expansion, i.e. as a sum of 1- and higher-body terms. Again a number of realizations of this idea have appeared over the years; an incomplete list of examples includes the fragment molecu-

lar orbital (FMO) method,[161] the incremental correlation scheme[162, 163, 164], and the Many-Body-Expansion approach.[165]Lastly, it is possible to reduce complexity by changing the numerical representation and employing fast algorithms for application of operators. For example, the use of real-space representation makes possible to reduce the complexity of the second-order Møller-Plesset (MP2) to $\mathcal{O}(n^3)$ from the traditional $\mathcal{O}(n^5)$ figure in the LCAO representation.[166, 167]

The current paper builds on recent successful efforts in complexity reduction based on the concepts of local correlation and truncated pair-natural orbitals (PNO). Although linear and reduced scaling realizations of traditional many-body methods up to CCSD(T) were realized some time ago, their combination with pair-natural orbitals (and other related ideas) greatly reduce the cost to achieve supremacy to canonical implementations for systems with 20 atoms (note that introduction of PNOs alone into CCSD(T) also reduces the scaling with system size[168, 169]). Combination with the explicit correlation[120, 121, 168, 170, 171] leads to robust methods that provide nearly numerically exact for most application purposes and applicable to three-dimensional systems with hundreds of atoms on a single workstation.[41, 42, 119, 120, 121, 172] Further reductions of time-to-solution are possible by harnessing the massively-parallel computer hardware of today, as demonstrated systematically by the Werner group.[170, 171, 173, 174, 175]

As one of the simplest correlated methods to derive and implement a large amount of effort has gone into the development of linear scaling second order Møller-Plesset perturbation theory (MP2), including Laplace transform techniques,[176, 177, 178] projected atomic orbitals,[142, 157, 179, 180, 181], and PNO based methods that often are built on top of reduced sized intermediate bases.[41, 173, 182] This work will build upon these techniques and combined with the idea that groups or clusters of occupied orbitals can share an intermediate basis in which to look for PNOs.

The goal of this work is to present an improvement to the linear-scaling formulation of the PNO approach by employing *clustered* basis of orbitals. The use of a clustered basis not only leads to reduced cost, but also improves the utilization of data parallelism which is crucial to efficient execution on modern and future computer architectures with wide vector units, such as all CPUs, general-purpose graphical processing units (GPGPUs), and accelerators like Intel Xeon Phi. To demonstrate the utility of clustered orbitals, in this work, we report on a massively parallel implementation of PNO-MP2 within the Massively Parallel Quantum Chemistry (MPQC) package that takes advantage of said clustering.

The remainder of the paper is organized into 5 more sections, Section 4.2 discusses the tools and tricks used to achieve linear scaling PNO construction, Section 4.3 discusses our specific implementation along with the idea of using pairs of clustered orbitals instead of just pairs of orbitals, Section 4.4 briefly discusses computational resources and molecular examples chosen, Section 4.5 shows the performance of our implementation, and finally Section 4.6 wraps up and discusses ideas for the future.

4.2 Formalism

Our approach here builds on the basic formalism for linear-scaling PNO evaluation of MP2 energy established by Pinski and co-workers (including one of us) [41] and by Werner and co-workers.[173] These formulations, which differ in technical details, largely build on the same ideas put forth much earlier by many others.[40, 142, 157, 179, 180, 181, 183, 184, 185, 186, 187] To introduce the idea of clustered orbitals, and how they advance the state-of-the-art, here we briefly recap the essential ingredients of the linear-scaling PNO methods. Additional technical details are given in Section 4.3. Interested readers are referred to the literature cited above for the more complete discussion of the technical details.

Table 4.1: Index notation used in this paper.

$\mu, \nu, \rho, \ldots$	Atomic orbitals (AOs)
$i, j, k, \ldots$	Localized occupied orbitals (LOOs)
$a, b, c, \ldots$	Canonical virtual orbitals
$w, x, y, \ldots$	Density-fitting (auxiliary) atomic orbitals (DFAOs)
$\tilde{\mu}, \tilde{\nu}, \tilde{\rho}, \ldots$	Projected atomic orbitals (PAO)
$\tilde{a}_i, \tilde{b}_i, \tilde{c}_i, \ldots$	Orthonormal orbital-specific PAOs
$\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{c}_{ij}, \ldots$	Orthonormal pair-specific PAOs
$a_i, b_i, c_i, \ldots$	Orbital-specific virtuals (OSVs) for orbital i
$\bar{a}_{ij}, \bar{b}_{ij}, \bar{c}_{ij}, \ldots$	OSV basis for pair ij
$a_{ij}, b_{ij}, c_{ij}, \ldots$	Pair natural orbitals (PNOs) for pair ij
$M, N, R, \ldots$	Clusters of AOs
$I, J, K, \ldots$	Clusters of LOOs
$W, X, Y, \ldots$	Clusters of DFAOs
$\tilde{A}_{IJ}, \tilde{B}_{IJ}, \tilde{C}_{IJ}, \ldots$	Orthonormal clustered-pair-specific PAOs
$\bar{A}_{IJ}, \bar{B}_{IJ}, \bar{C}_{IJ}, \ldots$	Clustered-pair-specific unionized OSVs

4.2.1 Local correlation

Standard expression for the MP2 energy with a closed-shell spin-restricted reference reads:

$$E_{\mathrm{MP2}} = \tilde{g}_{ij}^{ab} T_{ab}^{ij} \tag{4.1}$$

where $g_{rs}^{pq} \equiv \langle \phi_r(1)\phi_s(2) | \, r_{12}^{-1} \, | \phi_p(1)\phi_q(2) \rangle$, $\tilde{O}_{rs}^{pq} \equiv 2O_{rs}^{pq} - O_{sr}^{pq}$, and Einstein summation convention is used. The index space notation used throughout this paper is specified in Table 4.1. Amplitudes T_{ab}^{ij} are determined by solving the MP1 equations:

$$0 = R_{ab}^{ij} \equiv g_{ab}^{ij} + F_a^c T_{cb}^{ij} + F_b^c T_{ac}^{ij} - F_k^i T_{ab}^{kj} - F_k^j T_{ab}^{ik}. \tag{4.2}$$

The $\mathcal{O}(n^5)$ cost of computing the MP1 residual R_{ab}^{ij} dominates the $\mathcal{O}(n^3)$ cost of the standard Hartree-Fock LCAO solvers applicable in realistic basis sets. It is possible to reduce the complexity of MP2 to that of Hartree-Fock purely by switching to a representation which

permits fast application of operators, i.e. real space/grids.[166, 167, 188] However, this greatly increases the verbosity of the representation, especially in the high-precision limit. Hence to reduce the complexity while preserving the compactness of LCAO representation, it is necessary to switch to localized bases for both the occupied and unoccupied orbitals in Eqs. (4.1) and (4.2). Techniques for the localization of occupied states, leading to compact wave function representations with reduced formal complexity, were developed around the same time as the robust modern many-body methods.[189] The basis of occupied orbitals can be directly localized using a number of available approaches,[83, 146, 190, 191, 192, 193] whereas direct localization of unoccupied (virtual) states is more difficult. Pulay and others[142, 157, 179, 183, 184, 194] proposed the use of projected atomic orbitals (PAOs) as the building block for the localized basis of unoccupied orbitals. PAOs are linear combinations of atomic orbitals that span the unoccupied space, i.e.

$$|\tilde{\mu}\rangle \equiv \sum_{\nu} |\nu\rangle \, C_{\nu}^{\tilde{\mu}}, \tag{4.3}$$

where for a closed-shell system PAO coefficients are defined as

$$\mathbf{C} = \mathbf{I} - \frac{1}{2}\mathbf{DS}, \tag{4.4}$$

where $\mathbf{I}$ is the identity matrix in AO basis, $\mathbf{D}$ is the density matrix, and $\mathbf{S}$ is the overlap matrix. For systems with nonzero gap, PAOs are exponentially localized and thus lead to sparse representations; e.g. $g_{\tilde{\mu}\tilde{\nu}}^{ij}$ has $\mathcal{O}(n^2)$ significant elements when i and j are localized, and thus can be sparsified by thresholding. PAO $\tilde{\mu}$ was produced from AO μ, hence PAOs preserve the atom-centered structure of the AO basis. This is crucial for defining sets of PAOs (a.k.a. *domains*) that are closed under unitary transformations of atom blocks; PAO domain associated with a given localized occupied orbital i is a set of PAOs, $\{\tilde{\mu}_i\}$, that form

"significant" products $i\tilde{\mu}$; the measure of significance is defined using geometric/topological criteria[173] or using Hilbert space distance between i and $\tilde{\mu}$.[41] This strategy is generalized to any *tuple* of orbitals by taking unions of domains for the constituent orbitals; e.g. for pair ij $\{\tilde{\mu}_{ij}\} = \{\tilde{\mu}_i\} \oplus \{\tilde{\mu}_j\}$. However, PAOs are nonorthogonal and rank-deficient ($n_{\mathrm{PAO}} = n_{\mathrm{AO}} > n_{\mathrm{v}}$, where n_v is the number of virtual orbitals). Thus for each tuple-specific domain of PAOs we must compute its rank and remove the linearly dependent vectors. Before making PAOs orthogonal they are sparsified, then the resulting sparsified PAOs are orthogonalized. The standard construction of orthonormalized domain PAOs, due to Boughton and Pulay, is described in Ref. [183]; in this work we follow the somewhat simplified procedure of Ref. [41].

In terms of PAOs Eqs. (4.1) and (4.2) become

$$E_{\mathrm{MP2}} = \tilde{g}_{ij}^{\tilde{a}_{ij}\tilde{b}_{ij}} T_{\tilde{a}_{ij}\tilde{b}_{ij}}^{ij} \tag{4.5}$$

$$0 = R_{\tilde{a}_{ij}\tilde{b}_{ij}}^{ij} \equiv g_{\tilde{a}_{ij}\tilde{b}_{ij}}^{ij} + F_{\tilde{a}_{ij}}^{\tilde{c}_{ij}} T_{\tilde{c}_{ij}\tilde{b}_{ij}}^{ij} + F_{\tilde{b}_{ij}}^{\tilde{c}_{ij}} T_{\tilde{a}_{ij}\tilde{c}_{ij}}^{ij} - F_k^i T_{\tilde{a}_{kj}\tilde{b}_{kj}}^{kj} S_{\tilde{a}_{ij}}^{\tilde{a}_{kj}} S_{\tilde{b}_{ij}}^{\tilde{b}_{kj}} - F_k^j T_{\tilde{a}_{ik}\tilde{b}_{ik}}^{ik} S_{\tilde{a}_{ij}}^{\tilde{a}_{ik}} S_{\tilde{b}_{ij}}^{\tilde{b}_{ik}}. \tag{4.6}$$

Note the appearance of the overlaps between PAO domains in the last two terms of Eq. (4.6), due to the fact that PAOs for pair ij and pairs $\{ik, kj\}$ are not orthogonal. If the gap is nonzero, PAO MP2 energy costs $\mathcal{O}(n^2)$ without considering the cost of the integrals; the key to this scaling is the exponential decay of the exchange component of the Fock operator. In practice, the cost of PAO MP2 is usually dominated by the cost of computing the integrals, which is also $\mathcal{O}(n^2)$ assuming exponential localization (hence nonzero gap). The use of local density fitting approximation for the integrals, which allows to reduce the prefactor (but not the complexity) of their computation, was popularized by Manby, Werner, and co-workers.[181, 195]

Note that PAO MP2 has lower complexity than real space formulation of MP2,[167, 188] but

only by assuming exponential localization of orbitals. Further reduction of the complexity of PAO MP2 to $\mathcal{O}(n)$ requires multiscale treatment of electron correlation, e.g. by truncating pair interactions past certain distance range, or by employing hybrid quantum-classical approach where correlation energies of weak pairs are treated classically.[179, 196]

4.2.2 Pair-Natural Orbitals

Pair natural orbitals, introduced under the name pseudonatural orbitals by Edmiston and Krauss[186, 197], were explored extensively[185, 198, 199, 200, 201, 202, 203] as a compact basis in which to express electronic wave functions. PNOs are pair-specific orbitals orthogonal to the occupied subspace, defined as eigenstates of the corresponding pair's contribution to the unoccupied block of the 1-particle reduced density matrix (or, simply, pair density), which for a closed-shell spin-restricted reference state can be constructed from 2-body correlation amplitudes $(\mathbf{T}^{ij})_{ab} \equiv \{T^{ij}_{ab}\}$ as

$$\mathbf{D}^{ij} = \frac{2}{1+\delta_{ij}} \left((\tilde{\mathbf{T}}^{ij})^\dagger \mathbf{T}^{ij} + \tilde{\mathbf{T}}^{ij} (\mathbf{T}^{ij})^\dagger \right), \tag{4.7}$$

with $\tilde{\mathbf{T}}^{ij} \equiv 2\mathbf{T}^{ij} - (\mathbf{T}^{ij})^\dagger$. Eigenvectors $\mathbf{U}^{ij}$ of $\mathbf{D}^{ij}$,

$$\mathbf{D}^{ij} = \mathbf{U}^{ij} \mathbf{d} \left(\mathbf{U}^{ij} \right)^\dagger, \tag{4.8}$$

are the coefficients of PNOs:

$$|a_{ij}\rangle = \sum_a |a\rangle \, U^{ij}_{aa_{ij}}. \tag{4.9}$$

PNOs are truncated by only keeping PNOs whose corresponding eigenvalues ("occupation numbers") are greater than threshold τ_{PNO}. Once truncated PNOs have been determined,

amplitudes in the truncated PNO basis are determined by solving the corresponding equations. PNO equations are identical to those in the PAO basis, e.g. PNO MP1 amplitudes are obtained by solving Eq. 4.6 where all PAO indices $\tilde{a}_{ij}$ are replaced by their PNO counterparts $\tilde{a}_{ij}$. Note that PNOs are typically canonicalized to make quasi-Newton solvers for amplitude equations more robust.

To determine PNOs it is necessary to have a good guess of 2-body amplitudes in Eq. (4.7). In the context of ground-state CC-type methods exact MP1 amplitudes can be used as the guess amplitudes for PNO formation. However, following Neese[40] approximate (semicanonical) MP1 amplitudes are typically used as the PNO guess:

$$(T^{\text{scMP1}})^{ij}_{ab} = \frac{g^{ij}_{ab}}{F^a_a + F^b_b - F^i_i - F^j_j}. \tag{4.10}$$

in which only the unoccupied orbitals are assumed canonical (i.e. due to localization F^i_j is not diagonal). scMP1 PNOs were found by Neese to be nearly as good as older renormalized perturbative choices, however recently we found that substantial improvement is possible in the context of higher-order methods like CCSD.[204]

The use of PNOs alone (without additional approximations) affords a modest reduction of size complexity of CCSD, from $\mathcal{O}(n^6)$ to $\mathcal{O}(n^4)$, and greatly increased costs of the integral transformation (which in the *untruncated* PNO basis costs $\mathcal{O}(n^7)$ compared to the standard $\mathcal{O}(n^5)$ figure). Thus in practice PNO-based methods must leverage ideas of local correlation and local density fitting. To reduce the cost of computing integrals in the PNO basis Neese[40, 187] pioneered the use of local density fitting technology, which had already been used in the context of PAO-based reduced-scaling many-body methods by Werner and co-workers.[181, 205] Although LDF does not reduce the complexity of PNO CCSD, it is crucial to making PNO-based methods practical by reducing the overall costs such that the

supremacy to conventional methods is achieved for systems with 20 atoms.

To reduce formal scaling further it is necessary to express PNOs in terms of intermediate local bases, such as PAOs, i.e. Eq. (4.9) is replaced by

$$|a_{ij}\rangle = \sum_{\tilde{a}_{ij}} |\tilde{a}_{ij}\rangle \, U^{ij}_{\tilde{a}_{ij} a_{ij}} \tag{4.11}$$

in which $U^{ij}_{\tilde{a}_{ij} a_{ij}}$ are the eigenvectors of $\mathbf{D}^{ij}$ evaluated in the basis of orthogonal PAOs for pair ij. The use of PAOs as the intermediate basis reduces the cost of PNO construction from $\mathcal{O}(n^5)$ (or $\mathcal{O}(n^4)$, if using an iterative eigensolver) to $\mathcal{O}(n^2)$ and reduces the cost of the integral transformation. PAO-based PNO coupled-cluster methods were first simulated (implemented in a canonical CCSD program) by Krause and Werner.[206] First production implementation was reported by Riplinger and Neese for up to CCSD(T)[65, 207] and further improved by Neese, Valeev, and co-workers[41, 42, 119, 120, 121, 172]; implementation of PNO methods by Werner and co-workers[170, 171, 173, 174, 175] also supports PAO-based expansion of PNOs.

Another intermediate basis for expanding PNOs are orbital-specific virtuals (OSVs). First proposed by Yang *et al.*[208] as a standalone basis for compressed representation of cluster operator, OSVs were almost immediately proposed as the intermediate basis for PNO construction[182] OSVs are typically (but not always)[209] defined as eigenstates of pair densities in Eq. (4.7) of the *diagonal* pairs ii derived from MP1 amplitudes; OSVs are subsequently truncated in the same manner as PNOs, but using a separate (and tighter) criterion τ_{OSV}. The use of OSVs in PNO methods allows to reduce the $\mathcal{O}(n^5)$ cost of PNO construction to $\mathcal{O}(n^4)$ (or even to $\mathcal{O}(n^3)$ by using an iterative eigensolver and sparse integral

transformations[210]). However, using PAOs as the intermediate basis for OSV expansion,

$$|a_i\rangle = \sum_{\tilde{a}_i} |\tilde{a}_i\rangle \, U^{ii}_{\tilde{a}_i a_i} \tag{4.12}$$

in combination with local density fitting allows to reduce the cost of OSV construction to $\mathcal{O}(n)$. Although the PAO $\rightarrow$ OSV $\rightarrow$ PNO strategy does not change complexity relative to the simpler PAO $\rightarrow$ PNO strategy, its advantages are substantially cheaper PNO construction and integral transformation.

The ultimate reduction of complexity of PAO-(OSV)-PNO MP2 from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ is only possible through mixed classical-quantum treatment of electron correlation. One such approximate treatment proposed by Werner in Ref. [173], called the OSV-SC-DIP approach, involves a very simple dipole-dipole approximation to estimate the pair energy of weak pairs. That energy estimation is given as:

$$E^{\text{est}}_{ij} = -\frac{8}{R^6_{ij}} \sum_{\bar{a}_i} \sum_{\bar{b}_j} \frac{\left[\langle i \,|\, \mathbf{r} \,|\, \bar{a}_i\rangle \cdot \langle j \,|\, \mathbf{r} \,|\, \bar{b}_j\rangle\right]^2}{\epsilon^i_a + \epsilon^j_b - f_{ii} - f_{jj}}, \tag{4.13}$$

where R_{ij} is the distance between the centers of orbitals i and j and ϵ^i_a signifies the quasi-canonical energies for the OSV a for orbital i. This energy estimation is used both to estimate weak pairs and can also to estimate the contribution of the correlation energy from pairs that will thrown away during the full PNO-MP2 procedure.

4.2.3 Orbital-Clustered PNO MP2

To date, the use of orbital-based domains has focused on the most fine-grained construction possible, where every single orbital (or orbital tuple) receives its own unique set of virtuals. The main contribution of this work will be to show that a more coarse-grained partitioning,

specific to clusters and pairs of clusters of orbitals, will lead to significant savings in the context of PNO-MP2 based methods. While not shown, the same approach will also lead to savings in PAO-based PNO coupled-cluster methods as well.

For example, consider computing integrals $g^{ij}_{\tilde{a}_{ij}\tilde{b}_{ij}}$, which is one of the most expensive steps in the formation of the PNOs. In this construction, the orthogonalization and removal of linear dependencies of the PAOs in the domains of pair ij (taking the $\{\tilde{\mu}\} \in ij$ to the $\{\tilde{a}_{ij}\}$) takes considerable effort since it requires solving a fixed size, but large, eigenproblem for every single pair ij. But solving all of these eigenproblems likely leads to a large amount of redundant work, since for many pairs $\{\tilde{\mu}\}_{ij}$ is similar to $\{\tilde{\mu}\}_{kl}$, assuming that pairs kl and ij are "similar". In order to reduce the amount of redundant work, we propose the clustering of localized occupied orbitals into spatially close groups and the sharing of a domain of PAOs between all of the orbitals in the cluster. Then instead of needing to solve an eigenproblem for each pair of orbitals ij we can solve one for each pair of orbital clusters IJ. Assuming the size of the domains of the clusters is not significantly larger than the size of the domains of the individual pairs, the savings can be significant. If each cluster of orbitals contains exactly four orbitals then we will be able to solve one eigenproblem for 16 pairs at a time.

Moreover, the clustering of orbitals can make the transformation from AOs to PAOs significantly more efficient by allowing for more effective data parallelism within the linear algebra routines used for the integral transformations. This clustering is not without costs though. The most obvious objection will be that the size of the PAO domains needs to grow, and that it is not entirely obvious how to apply an intermediate basis of OSVs when this approach is used. To address the question of how to go from $g^{IJ}_{\tilde{A}_{IJ}\tilde{B}_{IJ}}$ to $g^{IJ}_{\bar{A}_{IJ}\bar{B}_{IJ}}$ we formed OSVs not for the diagonal pairs ii but instead formed OSVs for the diagonal clusters II. We did this by summing all of the pair densities contained in the cluster II and then diagonalizing the resulting matrix to determine a basis in which to represent the orbitals in the cluster

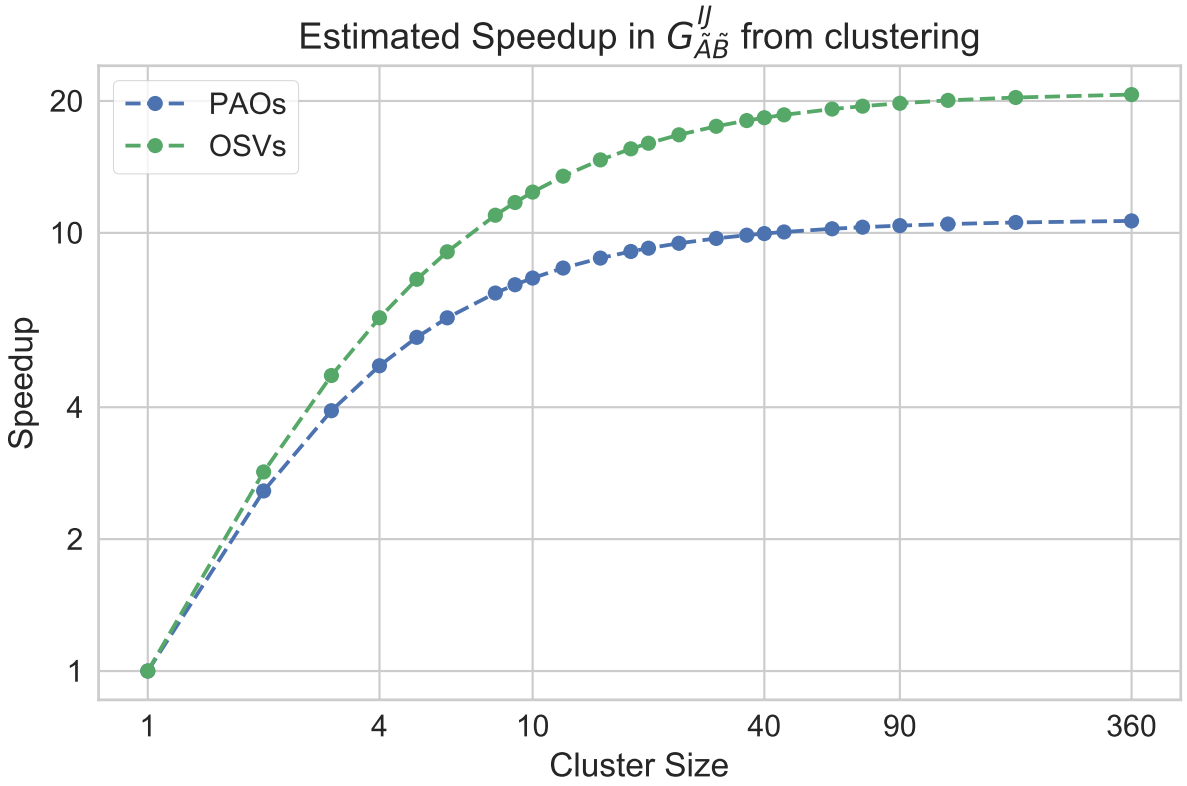

Figure 4.1: Estimated performance improvements (approximate number of operations) in the formation of $g^{IJ}_{\tilde{A}\tilde{B}}$ and $g^{IJ}_{\bar{A}\bar{B}}$ from clustering. Domain sizes were fixed such that pair ij would have the same number of PAO and OSV functions as cluster pair IJ.

I. While this does slightly increase the cost of forming OSVs, it should not be large when the sizes of the clusters are reasonable. In this work we were able to demonstrate significant savings in time to solution by using clusters of LOOs even though the time needed to form OSVs was slightly increased.

To estimate the performance gains that we could obtain in the formation of $g^{IJ}_{\tilde{A}\bar{B}_{IJ}}$ and $g^{IJ}_{\bar{A}_{IJ}\bar{B}_{IJ}}$ we made a simple performance model,[1] for 360 orbitals. In this performance model we added up the cost of all major operations (Eigenvalue decompositions, Cholesky solver, and matrix multiplications), ignoring prefactors and efficiency. This means that in our model

[1] A Mathematica notebook to generate the data for Figure 4.1 is given in the supporting information

an eigenvalue decomposition on an $n \times n$ matrix has the same cost as a matrix multiplication between two $n \times n$ matrices. The results of our model can be seen in Figure 4.1 and show that for relatively small orbital clusters there is a rapid improvement in performance, but that as the sizes of the clusters grow the benefit tapers off. Of course in reality as the size of the I clusters grow it is clear that the number of atoms in the domain of IJ must also grow to retain accuracy. But for small orbital clusters, say less than 5 orbitals per cluster, it should be possible for the sizes of the domains to increase only modestly or not at all.

Although there are many ways one might think to cluster orbitals, we made use of a method that we had already used in a previous work (ref [130]). Where orbitals were clustered by first determining their centers from the integrals $\langle i \,|\, \mathbf{r} \,|\, j \rangle$ and then using a simple k-means procedure to spatially group the orbitals. Technically, this procedure scales as $\mathcal{O}\left(n^2\right)$, but the time to form the integrals and to compute the clusters is insignificant relative to the total time for PNO-MP2.

Finally, we did preliminary testing of clustered PNOs by summing the densities of every pair ij in the cluster pair IJ, but the number of PNOs needed grew rapidly with increase in cluster size (we also tested with averaged densities, which helped but did not eliminate the issue). That is not to say that we never saw speedups relative to unclustered calculations —we did— but we quickly realized that the majority of the speed up was due to the benefits of sharing PAO domains between orbitals. Because of this we elected to switch exclusively to forming PNOs only for individual pairs. Once an integral had been constructed (either $g^{IJ}_{\tilde{A}_{IJ}\tilde{B}_{IJ}}$ or $g^{IJ}_{\bar{A}_{IJ}\bar{B}_{IJ}}$) we unclustered it and found the PNOs using $g^{ij}_{\tilde{A}_{IJ}\tilde{B}_{IJ}}$ (or the OSV variant). Essentially, after preliminary testing we determined that clustering was highly effective for forming intermediate bases in which to find PNOs, but that, at least for MP2, PNOs really do work best for pairs.

Throughout the paper we have used the general cluster notation of capital letters to refer to

clustered indices, but in reality for the calculations performed in this work all indices that were not LMO indices were clusters of hydrogen attached heavy atoms. This simplified the work needed to determine thresholds and remains very similar to the atom based clustering (domains) used in the vast majority of PAO and LDF work. We leave it to future work to determine if clustering of other indices besides I and J is appropriate.

4.3 Implementation

In this section we will outline the actual implementation details of the work in this paper. The development of our parallel $\mathcal{O}(n)$ PNO-MP2 code was largely based on references [41] and [173], the majority of the implementation details are well handled in those papers, here we will only discuss a basic outline of our strategy.

4.3.1 Sparse Maps

The concept of sparse maps, while falling under the domain of LDF, deserves a special mention here. While the idea is certainly not unique, the term sparse map was coined in the original set of DLPNO papers,[41, 42, 119, 120, 121] a sparse map between LMOs and PAOs, $\mathcal{S}_{I\to\tilde{A}}$, is nothing more than a boolean matrix of dimension number of orbital clusters I by number of PAO clusters $\tilde{A}$ with value 1 if $\tilde{A}$ is important for orbitals I and 0 if not. In the sparse map $\mathcal{S}_{I\to\tilde{A}}$ each row represents a single I and each column a single $\tilde{A}$ providing a simple way to determine if cluster A is in the domain of I. In a slight violation of our usual notation we will represent indices of sparse maps not by subscripts, but by parenthesis such that the element between orbital cluster J and PAO cluster $\tilde{B}$ in the map, $\mathcal{S}_{I\to\tilde{A}}$, is given by $\mathcal{S}_{I\to\tilde{A}}(J,\tilde{B})$. This way of thinking about sparse maps allows for easy construction of the

PAO domains for pairs IJ as unions by simply summing the pairs of orbital rows of $\mathcal{S}_{I\to\tilde{A}}$

$$\mathcal{S}_{IJ\to\tilde{A}}(KL,\tilde{B}) = \mathcal{S}_{I\to\tilde{A}}(K,\tilde{B}) + \mathcal{S}_{I\to\tilde{A}}(L,\tilde{B}). \tag{4.14}$$

Similarly we can construct new sparse maps from two other maps that share an index. A sparse map for auxiliary blocks X and PAO blocks $\tilde{A}$ can be constructed as:

$$\mathcal{S}_{X\to\tilde{A}}(Y,\tilde{B}) = \sum_{J} \mathcal{S}_{I\to X}(J,Y) * \mathcal{S}_{I\to\tilde{A}}(J,\tilde{B}), \tag{4.15}$$

by contracting over the J index. Map $\mathcal{S}_{X\to\tilde{A}}$ tells us which combinations of auxiliary function blocks and PAO blocks were connected by at least 1 orbital block, information that is useful the for the $\mathcal{O}(n)$ construction of LDF integrals. We see that Equation (4.15) is really just a matrix multiplication of $\mathcal{S}_{I\to X}$ and $\mathcal{S}_{I\to\tilde{A}}$,

$$\mathcal{S}_{X\to\tilde{A}} = \mathcal{S}^{\dagger}_{I\to X}\mathcal{S}_{I\to\tilde{A}} \tag{4.16}$$

This realization opens the door to a generalization of sparse maps to weighted maps where each element is not a boolean, but instead an integer or floating point number representing the importance of the connection between two indices. This generalization was not explored in this work but should be looked into in future work. Lastly, we should not fail to mention that others have worked with similar ideas.[115, 173]

4.3.2 Sparse Map Construction

There are several sparse maps which are needed for the $\mathcal{O}(n)$ construction of LDF integrals, discussed in Section 4.3.3, they are $\mathcal{S}_{I\to\tilde{A}}$ the orbital PAO map, $\mathcal{S}_{I\to X}$ the orbital auxiliary map, $\mathcal{S}_{I\to M}$ the orbital to AO map, $\mathcal{S}_{I\to J}$ a map that defines the list of significant pairs

—here computed with the OSV-SC-DIP approach using threshold tcutIJ—, $\mathcal{S}_{M\to N}$ a map of AO block pairs, and $\mathcal{S}_{X\to M}$ a map between auxiliary and AO blocks. Lastly, there were two maps $\mathcal{S}^U_{I\to\tilde{A}}$ and $\mathcal{S}^U_{I\to X}$ that represent the unions of all important $\tilde{A}$ and X for every J in I or more formally:

$$\mathcal{S}^U_{I\to\tilde{A}}(K,\tilde{B}) = \sum_L \mathcal{S}_{I\to J}(K,L)\mathcal{S}_{I\to\tilde{A}}(L,\tilde{B}), \tag{4.17}$$

where $\mathcal{S}^U_{I\to X}$ was formed in the exact same way.

$\mathcal{S}_{I\to\tilde{A}}$ and $\mathcal{S}_{I\to X}$ were formed based on Mulliken populations via the matrix $\mathbf{\Lambda}$ constructed as:

$$\Lambda_{MI} = L_{RI}\sum_N S_{MN}L_{NI}, \tag{4.18}$$

where $\mathbf{L}$ is the matrix of LMOs and $\mathbf{S}$ is the AO overlap. Once $\mathbf{\Lambda}$ has been constructed domains for $\mathcal{S}_{I\to\tilde{A}}$ were chosen if the norm of Λ_{MI} was larger than a threshold tcutIA and $\mathcal{S}_{I\to X}$ domains were chosen the same way except that a threshold tcutIX was used. We require that $\tilde{A}$, M, and X all have the same clustering structure. In this work each was clustered by hydrogen attached heavy atoms. This was mainly done because the limited number of basis functions that typically appear on hydrogen makes linear algebra over the hydrogen only blocks inefficient.

Next, we truncated the blocks of LMO coefficients that had norms less than a parameter tcutC and the structure of $\mathcal{S}_{I\to M}$ was initialized based on the remaining blocks of the LMOs. Next we formed a map $\mathcal{S}_{M\to\tilde{A}}$ based on the sparse block structure of the PAO projector and

farther pruned $\mathcal{S}_{I\to M}$ elements according to the following:

$$\mathcal{S}_{I\to M}(J,N) = \begin{Bmatrix} \mathcal{S}_{I\to M}(J,N), & \text{If } 0 < \sum_{\tilde{B}} \mathcal{S}^{U}_{I\to\tilde{A}}(J,\tilde{B})\mathcal{S}_{M\to\tilde{A}}(N,\tilde{B}) \\ 0, & \text{If } 0 = \sum_{\tilde{B}} \mathcal{S}^{U}_{I\to\tilde{A}}(J,\tilde{B})\mathcal{S}_{M\to\tilde{A}}(N,\tilde{B}) \end{Bmatrix} \tag{4.19}$$

In this way a block connection in $\mathcal{S}_{I\to M}$ was only important if it was important in the LMOs and if for the orbital block J there was a PAO block that interacted with AO block M, unlike in other works we did not truncate our PAO projector beyond the default block truncation of 10^{-11} used by TiledArray.

$\mathcal{S}_{M\to N}$ was initialized based on the estimated norm of the $(MN|MN)$ integrals obtained by Schwarz screening, using a threshold tcutRS, and then its intersection was taken with the result of $\mathcal{S}^{\dagger}_{I\to M}\mathcal{S}_{I\to M}$.

Finally, the map $\mathcal{S}_{X\to M}$ was formed from the product $(\mathcal{S}^{U}_{I\to X})^{\dagger}\mathcal{S}_{I\to M}$. Once all of these maps have been defined it is rather easy to construct the integrals needed for LDF in a linear scaling fashion.

4.3.3 $\mathcal{O}(n)$ LDF

To construct the $(Y|I\tilde{A})$ integrals in $\mathcal{O}(n)$ complexity we took advantage of TiledArray's sparse GEMM capability and a feature that allows for construction of only selected output blocks. Given the sparse maps from the previous section is is rather trivial to determine the needed output blocks. Our construction followed the next three steps:

- Construct all blocks $(Y|RS)$ where $\mathcal{S}_{X\to M}(Y,S) * \mathcal{S}_{X\to M}(Y,R) * \mathcal{S}_{M\to N}(R,S) > 0$
- Next perform the contraction $Z^{Y}_{JR} = \sum_{S}(Y|RS)L_{SJ}$ constructing output blocks only when $\mathcal{S}^{U}_{I\to X}(J,Y) * \mathcal{S}_{I\to M}(J,R) * \mathcal{S}_{X\to M}(Y,R) > 0$

- Finally, perform the contraction $Z^Y_{J\tilde{B}} = \sum_R Z^Y_{JR} P_{R\tilde{B}}$, where $\mathbf{P}$ is the PAO projector, forming output blocks only when $\mathcal{S}^U_{I\to\tilde{A}}(J,\tilde{B}) * \mathcal{S}^U_{I\to X}(J,Y) > 0$, we did not try to use a map $\mathcal{S}_{X\to\tilde{A}}$ because forming it from $\mathcal{S}^U_{I\to\tilde{A}}$ and $\mathcal{S}^U_{I\to X}$ would be redundant.

The screening of the above three contractions using the sparse maps as described is sufficient to construct all $(Y|I\tilde{A})$ blocks needed to form the $(I\tilde{A}|J\tilde{B})$ integrals used for PNO construction with $\mathcal{O}(n)$ complexity.

The distribution and parallelization of integral construction is handled via TILEDARRAY. The sparse maps algebra is formally $\mathcal{O}(n^2)$ and serial, but it is extremely efficient and was not noticeable in any of our calculations.

Finally, the integrals $g^{IJ}_{\tilde{A}\tilde{B}}$ are constructed in the standard way using the Cholesky decomposition of $(Y_{KL}|Z_{KL})$ with domains $\mathcal{S}_{I\to X}(K,W) + \mathcal{S}_{I\to X}(L,W)$. The distribution of the final integral formation step is done jointly with the formation of PNOs and is described in Section 4.3.4.

4.3.4 Pair Distribution

Once the $(Y|I\tilde{A})$ integrals are available we chose to form PNOs for $i \geq j$ in a round robin fashion over nodes, where each node would request the blocks of $(Y|I\tilde{A})$ that were needed for the pairs that it had been assigned. Other matrices that were needed: the PAO overlap, the PAO Fock matrix, and $(X|Y)$ were replicated across all nodes to avoid communication. To avoid repeated request for integrals each node kept a cache of blocks that it had already requested. Once the pair specific data had been computed for each pair had been computed all the relevant terms ($g^{ij}_{a_{ij}b_{ij}}$, eigenvalues of $F_{a_{ij}b_{ij}}$, etc.) were shipped to the final destination of that pair.

Once the PNOs were formed the terms needed for Equation (4.6) were distributed by orbital index j. We also distribute all ij pairs this way, not just $i \geq j$, allowing each node to compute all of the overlaps that and most of the residual terms it needs without data movement.

4.4 Computational Details

4.4.1 Software

Our implementation of these ideas took place in a developmental version of the Massively Parallel Quantum Chemistry (MPQC) package,[2] which takes advantage of the sparse tensor algebra provided by the library developed in our group TiledArray.[93]

The Hartree-Fock reference for all MP2 calculations was computed using the direct RI-RHF wavefunction avalible in MPQC and a TiledArray sparse threshold of 10^{-11} was used throughout for all sparse tensor contractions.

The code used was compiled using the g++ 6.1.0 compiler with compiler flags '-O3 -DNDEBUG -std=c++14'. The 2017.0.098 serial version of MKL was used for BLAS and Lapack calls.

4.4.2 Hardware

All computations were carried out on Virgina Tech's NewRiver system where each node has 2 Xeon E5-2680v3 2.5ghz Intel processors, with 24 cores and is available with either 128 GB or 512 GB of RAM. Each node has a theoretical peak performance of 960 Gflops/s with a measured peak of 780 Gflops/s in Intel MKL's DGEMM routine.

[2] An older version of the project can be found at https://github.com/ValeevGroup/mpqc

4.5 Results

4.5.1 Basis Sets and Molecules

In this work, we tested our $\mathcal{O}(n)$ PNO-MP2 implementation on linear n-alkanes generated using OPENBABEL[94] and three dimensional water clusters, the construction of which is outlined in reference [36], and which are publicly available online.[95] Finally, we tested our method on the 10 largest (in number of atoms) molecules in the Rx200 database, from reference [62].

4.5.2 Proof of $\mathcal{O}(n)$

For proof of linear scaling we used thresholds of tcutIX $= 10^{-4}$, tcutIA $= 2 \times 10^{-4}$, tcutIJ $= 10^{-6}$, and tcutPNO $= 10^{-8}$ which were sufficient to achieve approximately 99.9% accuracy for both linear n-alkanes and water clusters. For n-alkanes Figure 4.2 shows that we rapidly achieve a near linear scaling of $\mathcal{O}(n^{1.1})$ measured from $C_{50}H_{102}$ to $C_{80}H_{162}$. Waters do not scale as well, but Figure 4.3 shows that we achieve $\mathcal{O}(n^{2.74})$ complexity going from $(H_2O)_{47}$ to $(H_2O)_{76}$, which is significantly more performant than RI-MP2.

4.5.3 Parallel Efficiency

To test the distributed-memory performance of our code we tested n-alkanes from 1 to 24 nodes in Figure 4.4 and the overall scaling was pretty poor with only about a 2 times increase in performance from 1 to 24 nodes for $C_{80}H_{162}$ in def2-TZVP. But importantly the PNO formation part of the code scaled acceptably well achieving about 50% parallel efficiency, and only taking about 6 seconds for the 16 node calculation. The parts of the code that do

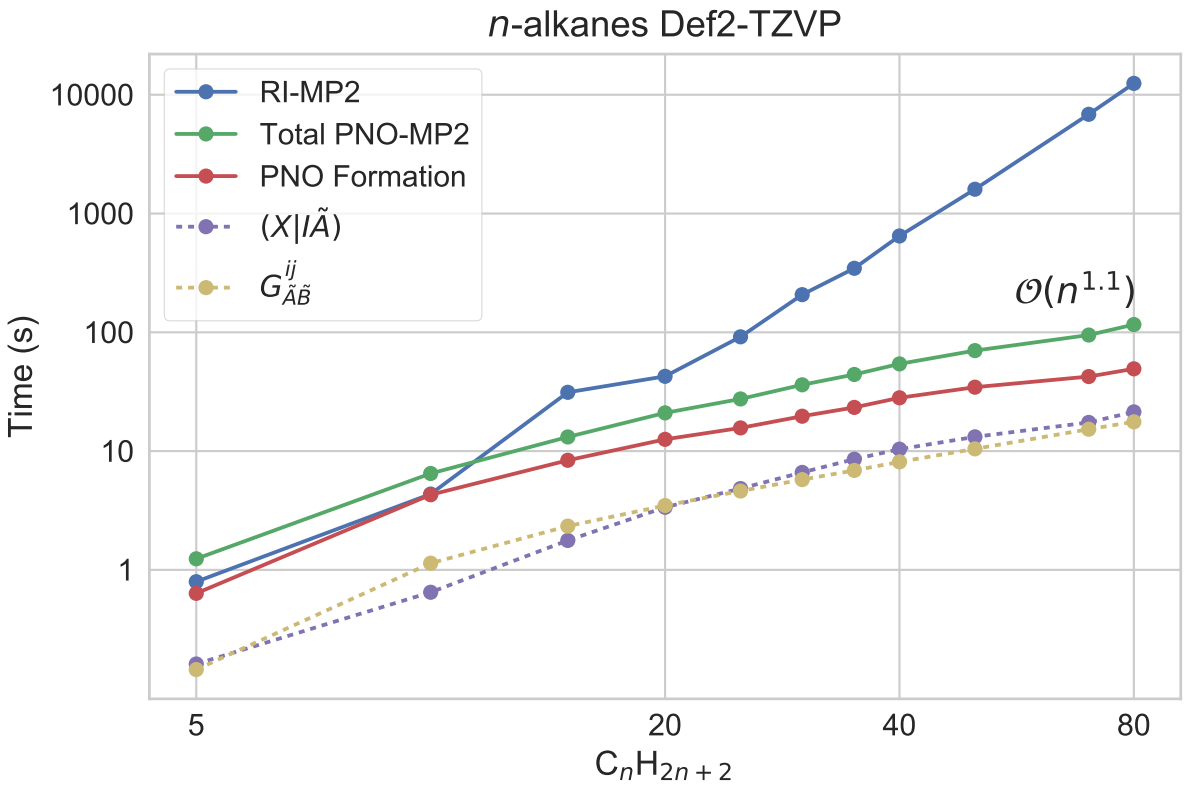

Figure 4.2: Comparison of timings for our $\mathcal{O}(n)$ PNO-MP2 versus RI-MP2 for n-alkanes. The formation of $G^{ij}_{\tilde{A}\tilde{B}}$ was accumulated in tasks and then averaged over the number of threads. PNO formation includes both the time to form $G^{ij}_{\tilde{A}\tilde{B}}$ and to diagonalize the pair densities.

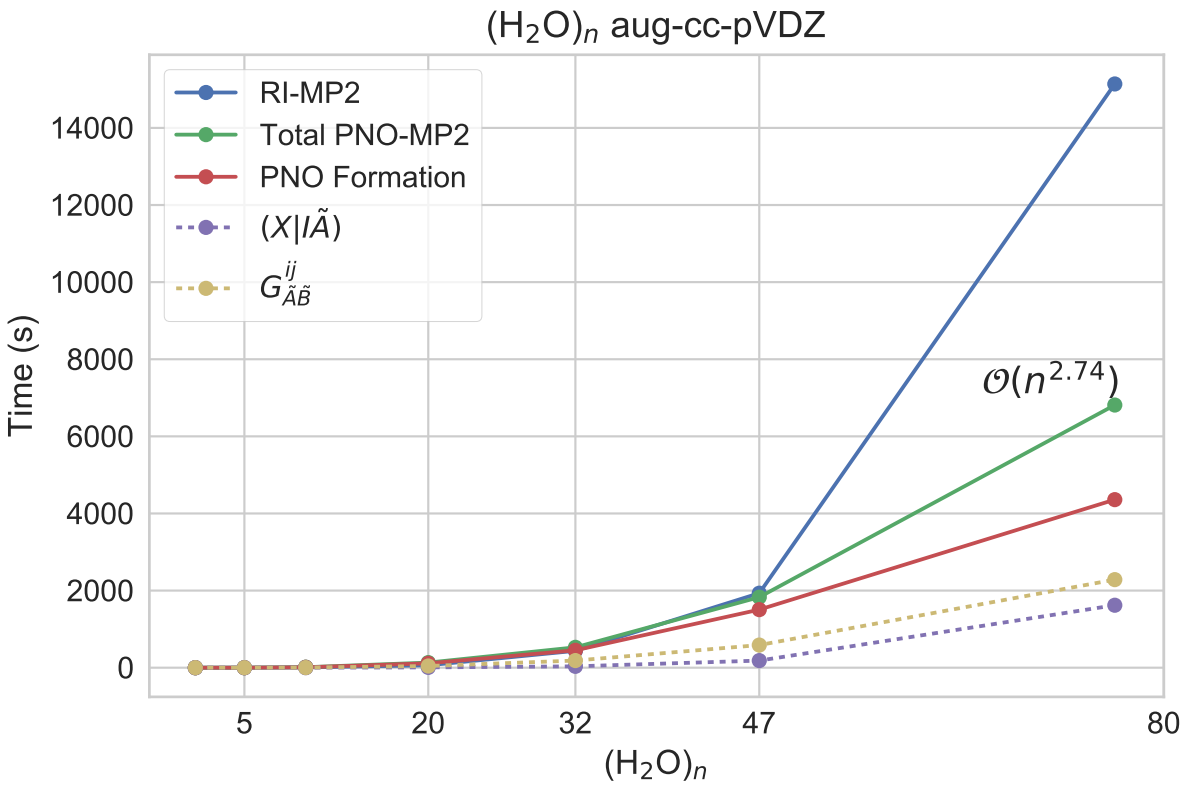

Figure 4.3: Comparison of timings for our $\mathcal{O}(n)$ PNO-MP2 versus RI-MP2 for water clusters. The formation of $G^{ij}_{\tilde{A}\tilde{B}}$ was accumulated in tasks and then averaged over the number of threads. PNO formation includes both the time to form $G^{ij}_{\tilde{B}\tilde{B}}$ and to diagonalize the pair densities.

not scale well, or at all, such as the formation of the $\mathcal{S}_{M\to N}$ and the $\mathcal{S}_{I\to J}$ start to become bottlenecks for linear systems such as n-alkanes. Water clusters on the other hand, shown in Figure 4.5, exhibit decent parallel scaling for all important steps. The two most costly steps, the formation of PNOs and the computation of $(Y|I\tilde{A})$ both achieved greater than 50% efficiency when going from 1 to 16 nodes and still has high efficiency all the way to 32 nodes. Currently we make no effort to optimize the distribution of work except to distribute the pairs of ij by orbital index j for the residual equations. The PNO formation is distributed by round robin and integrals are distributed with TiledArray's default distribution. Neither of these is likely the ideal distribution to minimize communication of our data. An obvious strategy to improve the scaling of both the $(Y|I\tilde{A})$ formation and the PNO formation is to attempt to minimize the communication needed between PNO pairs for both steps. Two such strategies could be hierarchical clustering of indices or the use of graph partitioning algorithms. We suspect that either of these approaches would show a large increase in the scalability for linear molecules where the bandwidth of the $\mathcal{S}_{I\to J}$ is easily minimized, but the effect for three dimensional systems should be less pronounced.

4.5.4 Effect of Clustering LMOs

During the development of our $\mathcal{O}(n)$ PNO-MP2 code we noticed that many ij pairs have very similar PAO domains. Given our experience using TiledArray, which already requires clustering for efficiency, it was relatively easy to adapt our code to allow for clustering of occupied orbitals i into groups of orbitals I.

To test the real world performance of clustering we choose our clusters to have an average of 4 orbitals per cluster and we choose new parameters for tcutIA, tcutIX, and tcutIJ, while tcutPNO $= 10^{-8}$ was kept constant throughout all calculations. This was done to keep

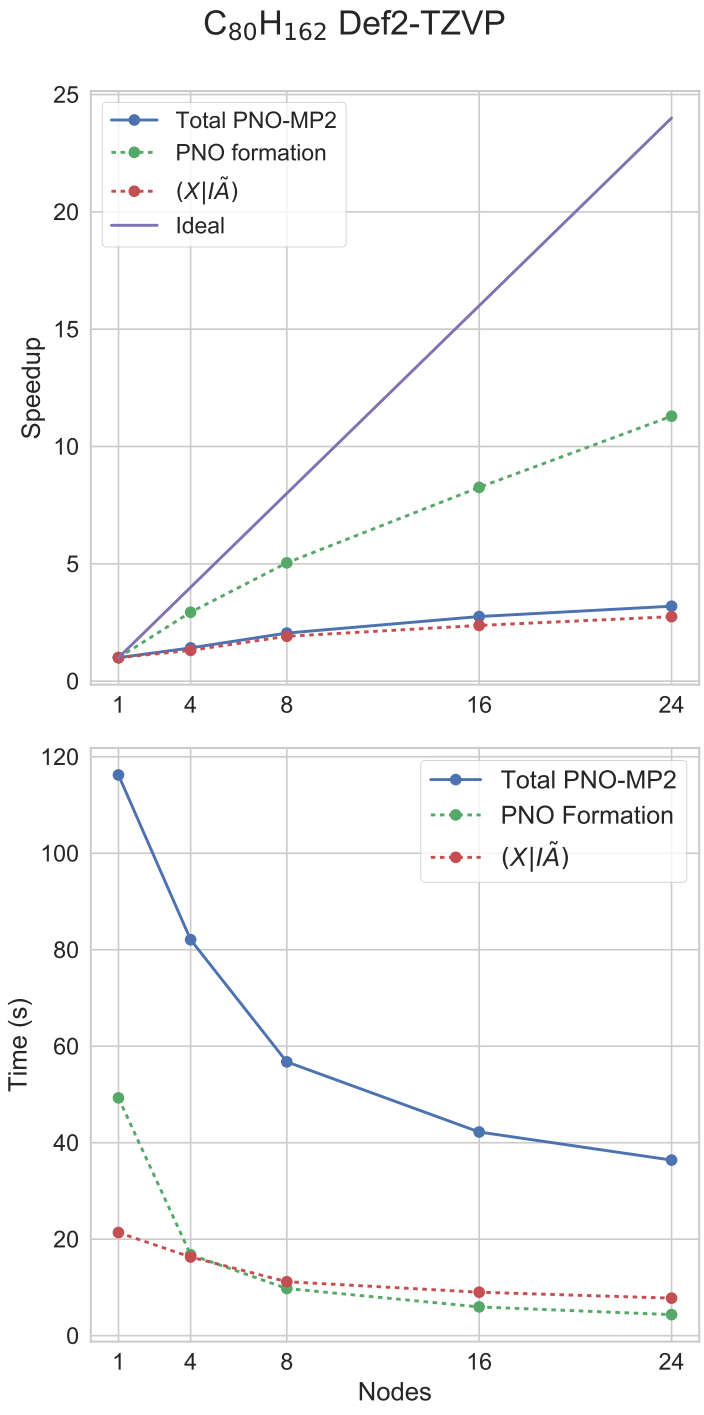

Figure 4.4: Speedup and actual times relative to the number of nodes (each node has 24 processors) for $C_{80}H_{162}$ in def2-TZVP

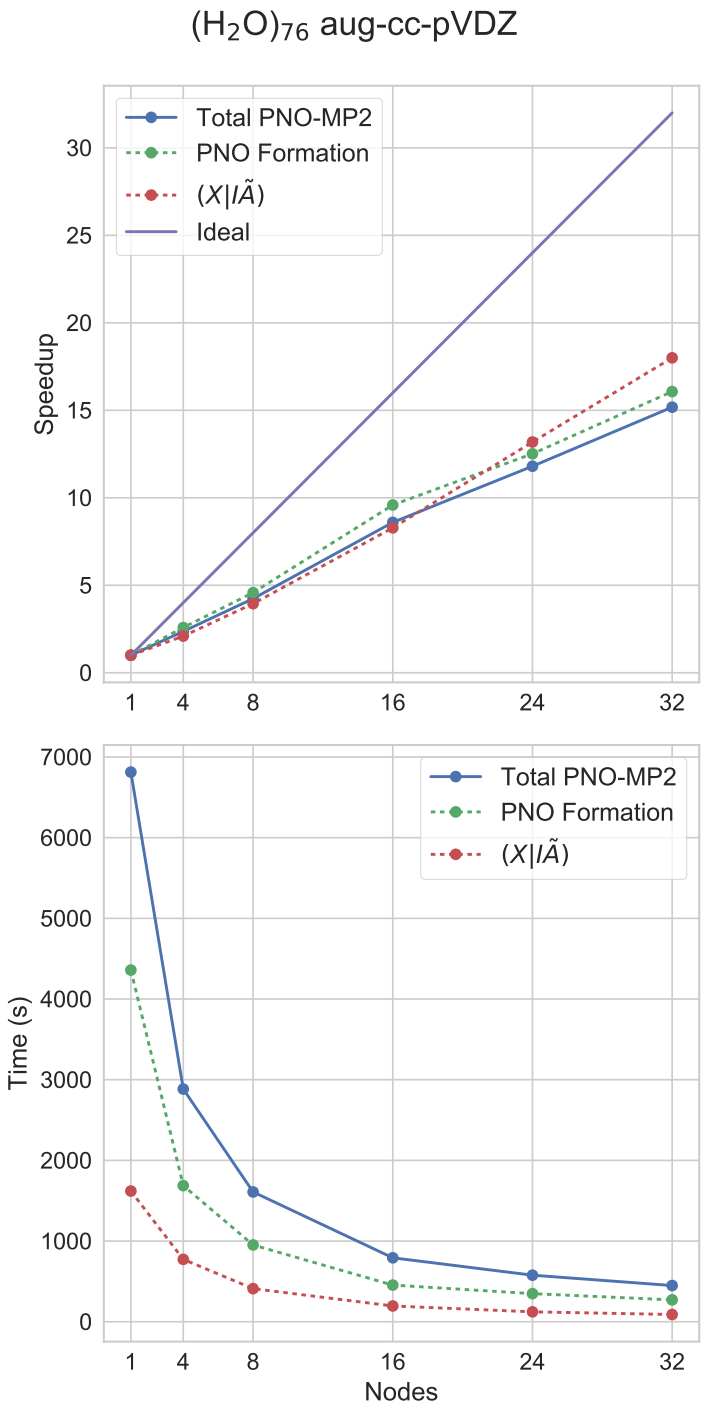

Figure 4.5: Speedup and actual times relative to the number of nodes (each node has 24 processors) for $(H_2O)_{76}$ in aug-cc-pVDZ.

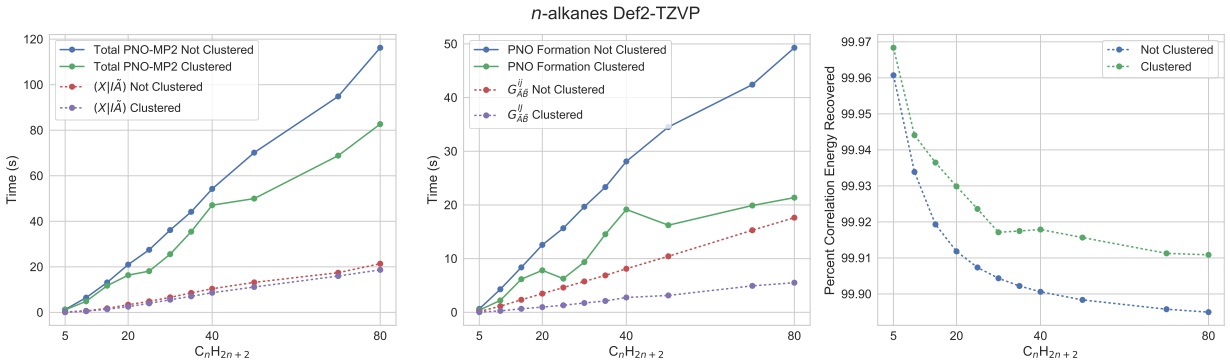

Figure 4.6: Effect of clustering occupied orbitals for n-alkanes with an average of 4 orbitals per cluster.

the final number of ij pairs, the number of domains, and the accuracy similar. Although not a perfect comparison; no calculations reported are below 99.8% error and usually the calculations with clustered LMOs have very similar errors. The parameters for the clustered calculations were tcutIA $= 6 \times 10^{-4}$, tcutIX $= 10^{-4}$, and tcutIJ $= 3 \times 10^{-5}$. For water clusters we also tested an even looser set of parameters tcutIA $= 8 \times 10^{-4}$, tcutIX $= 10^{-3}$, and tcutIJ=4×10^{-5}.

To start, we look at the effect of clustering on n-alkanes in Figure 4.6 and see that while not drastic, clustering does decrease the time to solution and makes a large —but not quite the estimated 5 times— improvement in the cost of computing the semi-canonical $g^{IJ}_{\tilde{A}_{IJ}\tilde{B}_{IJ}}$ integrals and thus reduces the time needed to compute PNOs by more than half. But it appears that clustering did not help improve the integral computation time, we believe this is because the integrals are already very fast for n-alkanes and that a significant portion of the integral time is actually preparing the sparse maps and performing Schwarz screening, both steps that do not gain from clustering. In the case of n-alkanes the parameters that were chosen actually increased the accuracy of the calculations while improving the performance. For waters the effect is even more pronounced as can be seen in Figure 4.7, with the largest loose parameter clustered calculation taking a total of 2236 seconds versus

about 6800 seconds for the calculation in which clusters were not used. Our calculations that do not take advantage of clustering only become faster than our implementation of RI-MP2 around $(H_2O)_{47}$, but both the clustered calculations cross over with the canonical method before $(H_2O)_5$. While impressive, it is likely that the block sizes chosen to make RI-MP2 efficient at large systems sizes are too large for maximally efficient small molecule RI-MP2, still though this early crossover was unexpected and welcome. For the more accurate calculations using orbital clusters we observed a speed up in the formation of $\mathbf{g}$ of 3.4 and a speedup in overall PNO formation of 2.1. In order to try and get closer to the theoretical 5 times speed up that 4 orbital clusters could bring to the semi-canonical integral formation we chose the loose parameters for the water clusters to try and bring the average number of waters in the maps $\mathcal{S}_{IJ\to X}$ and $\mathcal{S}_{IJ\to\tilde{A}}$ to be as close as possible to the calculations without clusters. To that end the clustered calculations with loose parameters spent a total of 8042 seconds (accumulated over tasks) in the formation of $g^{IJ}_{\tilde{A}_{IJ}\tilde{B}_{IJ}}$ while the calculations that didn't use clusters took about 54820 seconds a 6.8 times improvement! We are able to best the performance improvement of our model likely because the model ignores operation prefactor, but in reality the eigenvalue decompositions used for semi-canonicalization, which all $ij \in IJ$ can share, have a much higher prefactor than the contractions that form $\mathbf{g}$. It is clear that trying to match the domain sizes does result in a small loss of accuracy, but as the n-alkanes calculations show it is possible to maintain accuracy while at the same time improving the performance. Finally, for water clusters we see that the computation of $(Y|I\tilde{A})$ improved by about 2.2 times, which we attribute to more performance friendly block sizes in the integral contraction steps.

It is clear that clustering can lead to large performance improvements in $\mathbf{g}$ formation. Perhaps more importantly clustering of LMOs can improve the efficiency of $(Y|I\tilde{A})$ formation by allowing more optimal blocksizes for GEMM operations. Clustering is the ideal way to

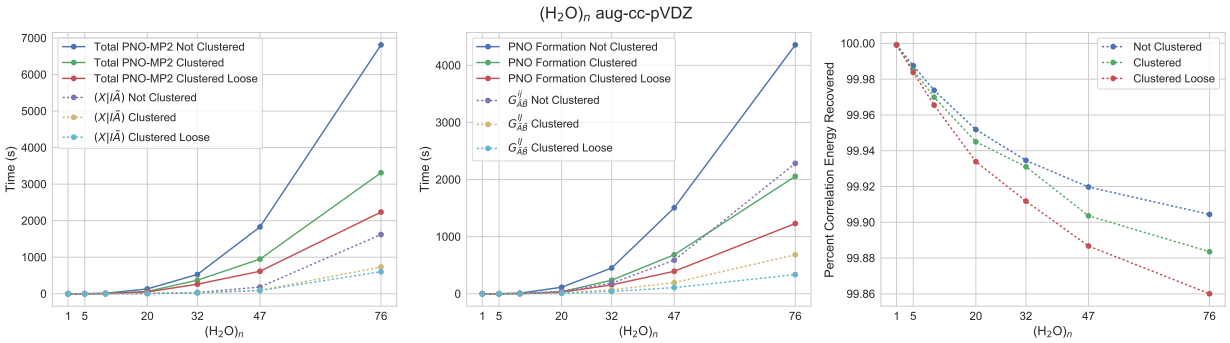

Figure 4.7: Effect of clustering occupied orbitals for water clusters with an average of 4 orbitals per cluster. The definition of Loose is defined in the text.

take advantage of the fact that many ij pairs, while having different PNO spaces share a very similar PAO space thus groups of pairs from clustered orbitals can share semi-canonical transformations as demonstrated here. Distributed-memory scaling performance of clustered PNOs is not shown here because at present the clustering makes exploitation of the fine grain task parallelism more complicated leading to their being too few tasks to effectively parallelize over hundreds of threads. This should be relatively easy to fix in future work by changing the formation of PNOs for all pairs in IJ from a single task to individual tasks for all pairs $ij \in IJ$, giving about the same amount of parallelism that is available in the code that does not cluster.

To show that the idea of orbital clustering and specifically the use of an average of 4 orbitals per cluster is not somehow unreasonably useful for water clusters we applied it to the 10 largest molecules in the Rx200 set of commonly prescribed drug molecules (here called Rx10) the results are shown in Figure 4.8 demonstrating that is is relatively easy to achieve about a 2 time improvement in the total PNO-MP2 time just by orbital clustering.

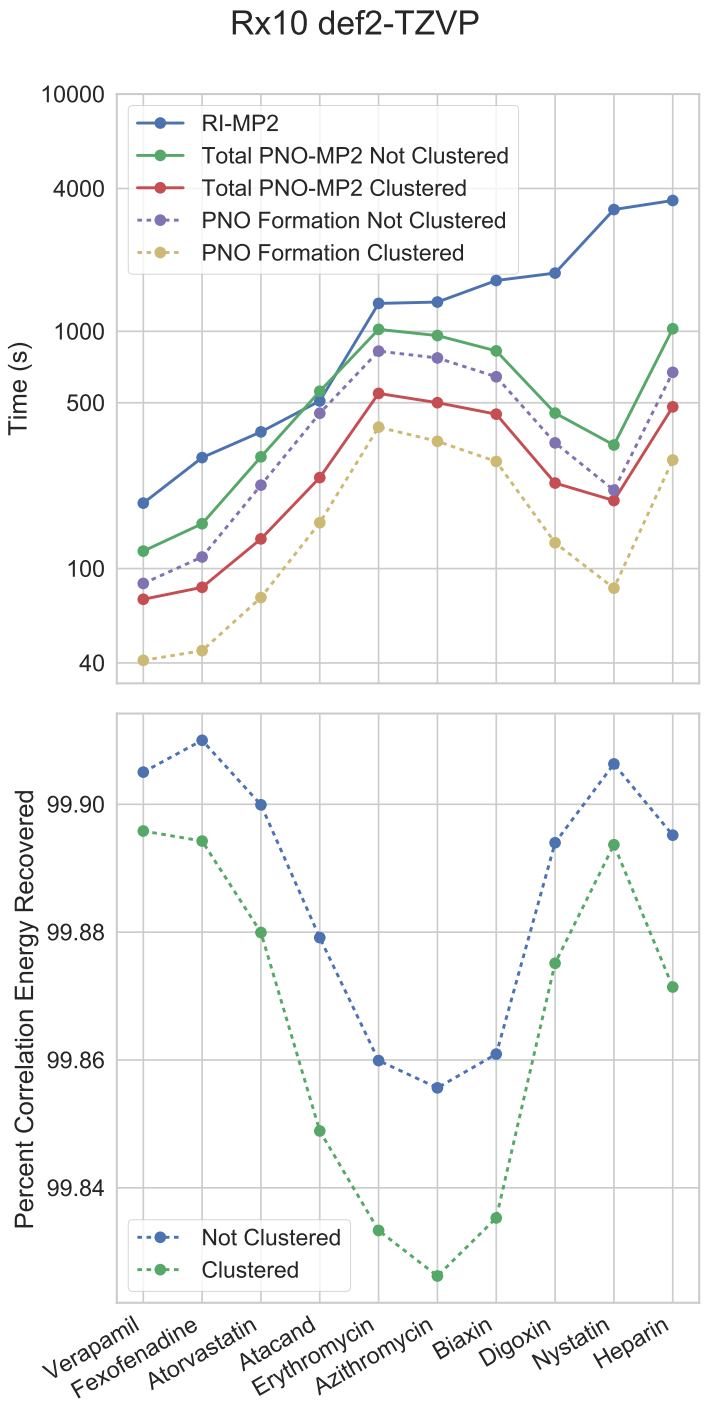

Figure 4.8: Performance and accuracy of our $\mathcal{O}(n)$ PNO-MP2 for the ten largest molecules in the Rx200 molecule set.

4.6 Conclusions

In this work we demonstrate a distributed-memory $\mathcal{O}(n)$ PNO-MP2 implementation that can exploit the similarity of auxiliary and PAO domains for small clusters of LMO orbitals. The use of clustered orbitals for 76 waters in an augmented double zeta basis allowed us to improve the total PNO-MP2 time by 3, the time for forming $(X|I\tilde{A})$ by 2.7, the time spent forming PNOs by 3.5, and the time spent semi-canonicalizing and forming PAO integrals by 6.8, all while maintaining a similar level of accuracy. It is clear that LMO clustering is a valid strategy to improve the efficiency of both PNO formation as well as the efficiency of the integral transform step required to form PNOs. All calculations shown in this work matched their RI-MP2 energies to greater than 99.8%, without the use of any corrections, such as a dipole dipole estimate for weak pairs, to the PNO-MP2 energy. Thus we could further improve the accuracy of our methods with these corrections. A simple performance model also showed that our clustering strategy should apply equally well to methods that use OSVs as an intermediate basis, although it would require OSVs for the orbital clusters II as opposed to just single orbitals ii. We do not foresee this being a large issue since our code currently computes these II OSVs for the construction of $\mathcal{S}_{I\rightarrow J}$, and in the case of 76 waters the time to compute II OSVs (with an average of 4 orbitals per cluster) was only about twice as expensive as constructing OSVs for ii.

Future development will consist of the following steps:

- improve the distributed-memory performance of clustered LMOs by adding more fine grain parallelism.
- add the option of using OSVs as an intermediate basis
- internally adapt the threshold parameters with respect to the clustering of LMOs,

instead of requiring input modification

- continue the development of PNO-CCSD in MPQC
- implement linear scaling F12 corrections in MPQC

Chapter 5

Summary

5.1 Recap

This dissertation has really been a study in approximation via factorization within the LCAO electronic structure framework. We started out in Chapter 1 by realizing that Hartree-Fock theory is a particular type of factorization of the true wavefunction Ψ, and is in some sense the best rank 1 approximation of Ψ. It turns out that while Hartree-Fock is able to obtain the majority of the total energy for a molecule, it is not accurate enough for quantitative prediction —really it is not even always accurate enough for qualitative prediction either. It turns out that we can improve the quality of our approximation by taking linear combinations of Slater determinants and thus increasing the rank of the approximation. We briefly covered several methods for determining this expansion including configuration interaction, coupled cluster, and MP theory.

Next we talked about another approximation called density fitting which tries to represent the product of two functions as a linear combination of single functions. We found in Chapter 1 that density fitting of the four-center two-electron integrals was an effective way of reducing the cost of the molecular orbital transformation, in Chapter 2 that it could lead to a reduced prefactor in Hartree-Fock, in Chapter 3 that further approximations could even lead to $\mathcal{O}(n)$ scaling of Hartree-Fock, and finally in Chapter 4 that local density fitting is an important part of linear scaling correlation treatment. I suppose this paragraph really makes a good case that this thesis is really the story of density fitting. It is only the main focus of Chapter 3, but it really is the tool that makes all of the work in this document possible. CLR with four-center integrals has a massive compression percent, but the amount of memory needed is still much too large to store in RAM. Linear scaling computation of the exchange term in Hartree-Fock is now over 20 years old, but in practice four center integrals are so costly that often density fitting, let alone reduced scaling density fitting, still pays off from medium sized molecules in realistic basis sets. Finally, the difficulties in

molecular integral construction when using PNOs are largely alleviated when density fitting is used, making PNO based correlation methods useful for large systems. Density fitting and especially local density fitting are not perfect though, they require special basis sets, depend heavily on error cancellation, and it is still an open problem on how to best achieve $\mathcal{O}(n)$ scaling of Hartree-Fock and even post Hartree-Fock methods when using it. For example, in $\mathcal{O}(n)$-PNO correlated methods eventually, for large enough systems, computation of local density fitting integrals will become the bottleneck.

5.2 Ideas and Future Directions

Ultimately the use of $\mathcal{O}(n)$ correlation methods will open up areas of research that until now have been the domain of either semi-classical treatment or parameterized methods. Given the large number of calculations on water clusters presented, it should come as no shock that one area where highly efficient methods can make a difference is in the study of the behavior of bulk water. To the lay observer it might seem surprising that there is still much we do not understand about the nature of possibly the most important substance, for life, on our planet —water. But even in 2018 we are still hard at work trying to pin down the exact details of the smallest of water clusters, the dimer![211] But to make quantitative predictions and develop a general model for bulk water we must go larger than just the water dimer, for example, the water hexamer is the first cluster to exhibit pronounced 3D structure making it an obvious staring point on the quest to understand the properties of bulk water.[212] It is well known that the potential energy surface of water clusters is both complicated and full of local minimum, necessitating the use of high accuracy methods if we want to determine the true structure of larger water clusters.[213] But these larger clusters have too many atoms and arrangements to be treated by traditional accurate wavefunction methods

(such as CCSD(T)). To cope with this one strategy is to create models that combine highly accurate calculations, on very small water clusters, with many body methods.[214] But these methods require at least two things from ab initio calculations: 1) that the potential energy surface of small clusters can be simulated by many thousands of individual calculations on different geometries and 2) that the results of the model can be validated against ab initio methods for small to medium sized clusters.

Finally, we see where our $\mathcal{O}(n)$ correlated methods come into this. While perhaps one day ab initio methods may allow for on-the-fly calculations of the interactions of a large number of waters possible, a more near term goal is to allow for the parameterization of models not with dimers and trimers, but with hexamers and larger n-mers. Due to the 3D nature of these larger clusters their use could allow for a more accurate models. Also massively parallel $\mathcal{O}(n)$ methods will allow for the benchmarking of these models on systems that more closely resemble the bulk water they intend to represent.

Many developments are needed before this type of undertaking is possible though. To accurately describe large water clusters we need more accurate methods than MP2, CCSD and CCSD(T) being good examples. Next, we would need the development of analytic force computations. Both of these tasks are non-trivial, but the work of this document coupled with the work of references [41, 42, 119, 120, 121, 172] and [170, 171, 173, 174, 175] provide a path toward achieving these goals. Efficient $\mathcal{O}(n)$ methods are still in their infancy, but one day make make the calculations described in this section actually worth attempting.

Now that I have outlined one possible future use of $\mathcal{O}(n)$ methods, I will outline some ideas that I did not attempt or see to completion as a graduate student, but that I think are good ideas or are obvious follow-ups to the work in this document:

- Potentially the most pressing task should be a massively parallel implementation of

$\mathcal{O}(n)$ PNO-CCSD, while not really interesting anymore from a flag planting standpoint since other implementations already exist, a truly massively parallel version would allow for CCSD calculations on molecules having potentially thousands of atoms. This coupled with linear scaling F12 corrections has the potential to allow wavefunction based methods to be used for systems that previously were exclusively within the purview of density functional theory.

- A truly efficient and accurate linear scaling CCSD(T) or CCSDT implementation. Unlike CCSD this is somewhat of an unsolved problem. It seems that we are getting close, though. The groups of Werner and Neese are both hard at work on this problem. The real benefit to including the triples correction is that suddenly true quantitative accuracy becomes possible for many systems of chemical interest.

- Semi-exact integral approximations, this is an idea that is very similar to $\mathcal{H}$-matrix and $\mathcal{H}^2$-matrix approaches, where certain blocks or regions of a matrix or tensor are not approximated. A former postdoc in our group David Hollman explored this idea where certain blocks of the four-center two-electron integrals were computed exactly and the rest were computed using CADF. It turns out that it works exceptionally well ... until it does not. Edward and I tried something very similar which was to compute certain blocks of the density in Hartree-Fock with exact four-center integrals and then compute the other blocks with CADF. We found amazing results, where for some basis sets, without diffuse functions, we could achieve better errors than density fitting at roughly the cost of linear scaling CADF, the best of both worlds if you will. There was a hidden danger though, we were subtly altering the spectrum of $(\mu\nu|\rho\sigma)$ and once diffuse functions entered the basis SCF convergence sometimes catastrophically failed so we abandoned the idea. I think this avenue might still be the best way to achieve a truly reduced scaling Hartree-Fock implementation, but it is not yet clear to us how

to maintain the positive definiteness of the integrals in an efficient way.

- The CLR contraction implementation is far from optimized and I think my biggest regret is not having put more effort into making it efficient. Also, CLR is very performant for DF-J in Hartree-Fock, but because J is rarely a bottleneck it is hard to get excited about this.

- Finally, we really could use better distributed memory linear algebra that does not depend on uniform blocksizes, but that is a pipe dream that I would not hold my breath for.

Bibliography

[1] W. E. Wick, Appl. Microbiol. **15**, 765 (1967).

[2] J. F. Fisher, S. O. Meroueh, and S. Mobashery, Chem. Rev. **105**, 395 (2005).

[3] A. J. Mulholland, Drug Discovery Today **10**, 1393 (2005).

[4] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Courier Corporation, 2012).

[5] B. T. Sutcliffe, in *Methods in Computational Molecular Physics*, NATO ASI Series (Springer, Boston, MA, 1992) pp. 19–46.

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

[7] D. A. McQuarrie, *Quantum Chemistry* (University Science Books, 2008).

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

[9] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (John Wiley & Sons, 2014).

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

[11] T. D. Crawford and H. F. Schaefer, Reviews in Computational Chemistry **14**, 33 (2000).

[12] C. Møller and M. S. Plesset, Phys. Rev. **46**, 618 (1934).

[13] (2014), created by Smithsonian 3D Digitization Program, Obtained from https://newsdesk.si.edu/sites/default/files/photos/DPO_ObamaBust_3views_1.jpg on March 19, 2018, and used according to the terms at https://www.si.edu/termsofuse.

[14] (2014), obtained from https://obamawhitehouse.archives.gov/sites/whitehouse.gov/files/images/Frontiers_photocaption.jpg on March 22, 2018. This photo was taken by Pete Souza and is part of the public domain according to https://obamawhitehouse.archives.gov/copyright.

[15] (2009), obtained from https://commons.wikimedia.org/wiki/File:Official_portrait_of_Barack_Obama.jpg on March 22, 2018. This photo was taken by Pete Souza and according to the url it was obtained from it was originally hosted at http://change.gov/newsroom/entry/new_official_portrait_released/ under a Creative Commons Attribution 3.0 License, although that link is now dead.

[16] A. Tajti, P. G. Szalay, A. G. Császár, M. Kállay, J. Gauss, E. F. Valeev, B. A. Flowers, J. Vázquez, and J. F. Stanton, J. Chem. Phys. **121**, 11599 (2004).

[17] Y. J. Bomble, J. Vázquez, M. Kállay, C. Michauk, P. G. Szalay, A. G. Császár, J. Gauss, and J. F. Stanton, J. Chem. Phys. **125**, 064108 (2006).

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

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

[20] G. H. Booth, A. Grüneis, G. Kresse, and A. Alavi, Nature **493**, 365 (2013).

[21] J. Yang, W. Hu, D. Usvyat, D. Matthews, M. Schütz, and G. K.-L. Chan, Science **345**, 640 (2014).

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

[23] R. J. Bartlett, Annu. Rev. Phys. Chem. **32**, 359 (1981).

[24] P. V. Strassen, Numer. Math. **13**, 354 (1969).

[25] L. Greengard and V. Rokhlin, J. Comp. Phys. **73**, 325 (1987).

[26] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).

[27] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).

[28] X. P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).

[29] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

[30] J. Cloizeaux, Phys. Rev. **135**, A685 (1964).

[31] C. Ochsenfeld and J. Kussmann, Rev. Comp. Chem. **23**, 1. (2007).

[32] A. Niklasson and C. J. Tymczak, J. Chem. Phys. **118**, 8611 (2003).

[33] Y. Shao, C. Saravanan, and M. Head-Gordon, J. Chem. Phys. **118**, 6144 (2003).

[34] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).

[35] A. H. R. Palser and D. E. Manolopoulos, Phys. Rev. B **58**, 12704 (1998).

[36] E. Rudberg, E. H. Rubensson, and P. Sałek, J. Chem. Phys. **128**, 184106 (2008).

[37] E. Rudberg, E. H. Rubensson, and P. Sałek, J. Chem. Theory Comput. **7**, 340 (2011).

[38] J. VandeVondele, U. Borštnik, and J. Hutter, J. Chem. Theory Comput. **8**, 3565 (2012).

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

[40] F. Neese, F. Wennmohs, and A. Hansen, J. Chem. Phys. **130**, 114108 (2009).

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

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

[43] R. Vandebril, M. Van Barel, G. Golub, and N. Mastronardi, Calcolo **42**, 249 (2005).

[44] W. Hackbusch, Computing **62**, 89 (1999).

[45] W. Hackbusch and B. N. Khoromskij, Computing **64**, 21 (2000).

[46] W. Hackbusch, B. Khoromskij, and S. A. Sauter, in *Lectures on Applied Mathematics* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2000) pp. 9–29.

[47] W. Hackbusch and S. Börm, Computing **69**, 1 (2002).

[48] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A. J. van der Veen, and D. White, SIAM. J. Matrix Anal. & Appl. **27**, 341 (2005).

[49] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, Numer. Linear Algebra Appl. **17**, 953 (2010).

[50] V. Rokhlin, J. Comp. Phys. **60**, 187 (1985).

[51] W. Hackbusch and Z. P. Nowak, Numer. Math. **54**, 463 (1989).

[52] E. Tyrtyshnikov, Calcolo **33**, 47 (1996).

[53] E. Tyrtyshnikov, in *Numerical Analysis and Its Applications*, Lecture Notes in Computer Science, Vol. 1196, edited by L. Vulkov, J. Waśniewski, and P. Yalamov (Springer Berlin Heidelberg, 1997) pp. 505–516.

[54] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, Linear Algebra and its Applications **261**, 1 (1997).

[55] V. Weber, T. Laino, A. Pozdneev, I. Fedulova, and A. Curioni, J. Chem. Theory Comput. **11**, 3145 (2015).

[56] C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).

[57] E. Schwegler and M. Challacombe, J. Chem. Phys. **105**, 2726 (1996).

[58] J. C. Burant, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **105**, 8969 (1996).

[59] F. Neese, F. Wennmohs, A. Hansen, and U. Becker, Chem. Phys. **356**, 98 (2009).

[60] A. Sodt, J. E. Subotnik, and M. Head-Gordon, J. Chem. Phys. **125**, 194109 (2006).

[61] P. Merlot, T. Kjærgaard, and T. Helgaker, J. Comput. Chem **34**, 1486 (2013).

[62] D. S. Hollman, H. F. Schaefer, and E. F. Valeev, J. Chem. Phys. **140**, 064109 (2014).

[63] D. S. Hollman, H. F. S. III, and E. F. Valeev, "Fast construction of the exchange operator in atom-centered basis with concentric atomic density fitting." http://arxiv.org/abs/1410.4882 (2014).

[64] H.-J. Werner and M. Schütz, J. Chem. Phys. **135**, 144116 (2011).

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

[66] M. Challacombe, Comput. Phys. Commun. **128**, 93 (2000).

[67] N. Bock and M. Challacombe, SIAM J. Sci. Comput. **35**, C72 (2013).

[68] S. P. Lloyd, IEEE Transactions on Information Theory **28**, 129 (1982).

[69] D. Arthur and S. Vassilvitskii, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics, 2007) pp. 1027–1035.

[70] A. K. Jain, Pattern Recognition Letters **31**, 651 (2010).

[71] D. Arthur, B. Manthey, and H. Röglin, in *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on* (2009) pp. 405–414.

[72] D. S. Hollman, H. F. Schaefer, and E. F. Valeev, J. Chem. Phys. **142**, 154106 (2015).

[73] T. F. Chan, Linear Algebra and its Applications **88-89**, 67 (1987).

[74] G. Quintana-Ortí, X. Sun, and C. H. Bischof, SIAM J. Sci. Comput. **19**, 1486 (1998).

[75] C. H. Bischof and G. Quintana-Ortí, ACM Transactions on Mathematical Software (TOMS) **24**, 226 (1998).

[76] U. Borštnik, J. VandeVondele, V. Weber, and J. Hutter, Parallel Computing **40**, 47 (2014).

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

[78] E. J. Baerends, D. E. Ellis, and P. Ros, Chem. Phys. **2**, 41 (1973).

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

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

[81] B. I. Dunlap, J. Mol. Struct.: THEOCHEM **529**, 37 (2000).

[82] R. A. Kendall and H. A. Früchtl, Theoret. Chim. Acta **97**, 158 (1997).

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

[84] S. Manzer, P. R. Horn, N. Mardirossian, and M. Head-Gordon, J. Chem. Phys. **143**, 024113 (2015).

[85] R. T. Gallant and A. St-Amant, Chem. Phys. Lett. **256**, 569 (1996).

[86] C. F. Guerra, J. G. Snijders, and G. te Velde, Theor. Chem. Acc. **99**, 391 (1998).

[87] M. A. Watson, N. C. Handy, and A. J. Cohen, J. Chem. Phys. **119**, 6475 (2003).

[88] C. Köppl and H.-J. Werner, J. Chem. Theory Comput. **12**, 3122 (2016).

[89] A. Sodt, J. E. Subotnik, and M. Head-Gordon, J. Chem. Phys. **125**, 194109 (2006).

[90] A. Sodt and M. Head-Gordon, J. Chem. Phys. **128**, 104106 (2008).

[91] M. Häser and R. Ahlrichs, J. Comput. Chem. **10**, 104 (1989).

[92] E. F. Valeev and J. T. Fermann, "Libint: machine-generated library for efficient evaluation of molecular integrals over Gaussians (version 2.2.0)," `http://github.com/evaleev/libint/` (2016).

[93] J. A. Calvin and E. F. Valeev, "Tiledarray: A massively-parallel, block-sparse tensor library written in C++ ," https://github.com/valeevgroup/tiledarray/ (2015).

[94] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, J. Cheminf. **3**, 33 (2011).

[95] http://www.ergoscf.org/xyz/h2o.php, accessed : 2015-08-10.

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

[97] F. Weigend and R. Ahlrichs, Phys. Chem. Chem. Phys. **7**, 3297 (2005).

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

[99] J. A. Calvin, C. A. Lewis, and E. F. Valeev, in *Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms*, IA^3 '15 (ACM, New York, NY, USA, 2015) pp. 4:1–4:8.

[100] J. Almlöf, K. Faegri, and K. Korsell, J. Comput. Chem. **3**, 385 (1982).

[101] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).

[102] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).

[103] P.-O. Löwdin, J. Chem. Phys. **21**, 374 (1953).

[104] J. A. Jafri and J. L. Whitten, J. Chem. Phys. **61**, 2116 (1974).

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

[106] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, J. Chem. Phys. **71**, 4993 (1979).

[107] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs, Chem. Phys. Lett. **240**, 283 (1995).

[108] K. Eichkorn, F. Weigend, O. Treutler, and R. Ahlrichs, Theor Chem Acta **97**, 119 (1997).

[109] F. Weigend, Phys. Chem. Chem. Phys. **4**, 4285 (2002).

[110] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon, PNAS **102**, 6692 (2005).

[111] P. M. W. Gill, A. T. B. Gilbert, S. W. Taylor, G. Friesecke, and M. Head-Gordon, J. Chem. Phys. **123**, 061101 (2005).

[112] S. Reine, E. Tellgren, A. Krapp, T. Kjærgaard, T. Helgaker, B. Jansik, S. Høst, and P. Salek, J. Chem. Phys. **129**, 104101 (2008).

[113] M. Krykunov, T. Ziegler, and E. v. Lenthe, Int. J. Quantum Chem. **109**, 1676 (2009).

[114] P. Merlot, T. Kjærgaard, T. Helgaker, R. Lindh, F. Aquilante, S. Reine, and T. B. Pedersen, J. Comput. Chem. **34**, 1486 (2013).

[115] S. Manzer, E. Epifanovsky, A. I. Krylov, and M. Head-Gordon, J. Chem. Theory Comput. **13**, 1108 (2017).

[116] S. F. Manzer, E. Epifanovsky, and M. Head-Gordon, J. Chem. Theory Comput. **11**, 518 (2015).

[117] E. Rebolini, R. Izsák, S. S. Reine, T. Helgaker, and T. B. Pedersen, J. Chem. Theory Comput. **12**, 3514 (2016).

[118] D. S. Hollman, H. F. Schaefer, and E. F. Valeev, Mol. Phys. **115**, 2065 (2017).

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

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

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

[122] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

[123] A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).

[124] E. Solomonik and T. Hoefler, arXiv:1512.00066 [cs] (2015), arXiv: 1512.00066.

[125] A. Buluç and J. Gilbert, SIAM J. Sci. Comput. **34**, C170 (2012).

[126] A. Buluc and J. R. Gilbert, in *2008 37th International Conference on Parallel Processing* (2008) pp. 503–510.

[127] E. Hernández, M. J. Gillan, and C. M. Goringe, Phys. Rev. B **53**, 7147 (1996).

[128] X.-P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).

[129] Y. Shao, C. Saravanan, M. Head-Gordon, and C. A. White, J. Chem. Phys. **118**, 6144 (2003).

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

[131] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).

[132] A. Lazzaro, J. VandeVondele, J. Hutter, and O. Schuett, arXiv:1705.10218 [cs] , 1 (2017), arXiv: 1705.10218.

[133] I. Bethune, A. Gloess, J. Hutter, A. Lazzaro, H. Pabst, and F. Reid, arXiv:1708.03604 [cs] (2017), arXiv: 1708.03604.

[134] D. Cremer and J. Gauss, J. Comput. Chem. **7**, 274 (1986).

[135] C. Saravanan, Y. Shao, R. Baer, P. N. Ross, and M. Head–Gordon, J. Comput. Chem. **24**, 618 (2003).

[136] E. H. Rubensson and E. Rudberg, J. Comput. Chem. **32**, 1411 (2011).

[137] E. H. Rubensson and E. Rudberg, Parallel Computing **57**, 87 (2016).

[138] E. H. Rubensson, E. Rudberg, and P. Sałek, J. Chem. Phys. **128**, 074106 (2008).

[139] E. H. Rubensson, E. Rudberg, and P. Sałek, J. Comput. Chem. **28**, 2531 (2007).

[140] A. M. N. Niklasson, S. M. Mniszewski, C. F. A. Negre, M. J. Cawkwell, P. J. Swart, J. Mohd-Yusof, T. C. Germann, M. E. Wall, N. Bock, E. H. Rubensson, and H. Djidjev, J. Chem. Phys. **144**, 234101 (2016).

[141] D. Kats and F. R. Manby, J. Chem. Phys. **138**, 144101 (2013).

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

[143] M. McCourt, B. Smith, and H. Zhang, SIAM J. Matrix Anal. & Appl. **36**, 90 (2015).

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

[145] C. A. Lewis and E. F. Valeev, Fork located at `http://github.com/calewis/libint/`, commit 2eef4545ecd6b30adb4f09480ab87ab452e76c66 was used for results (2016).

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

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

[148] W. Koch and M. C. Holthausen, *A Chemist's Guide to Density Functional Theory* (John Wiley & Sons, 2015).

[149] W. Kutzelnigg, Theoret. Chim. Acta **68**, 445 (1985).

[150] S. Ten-no and J. Noga, WIREs Comput Mol Sci **2**, 114 (2012).

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

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

[153] A. Warshel and M. Levitt, Journal of Molecular Biology **103**, 227 (1976).

[154] R. A. Friesner and V. Guallar, Annu. Rev. Phys. Chem. **56**, 389 (2005).

[155] F. R. Manby, M. Stella, J. D. Goodpaster, and T. F. Miller, III, J. Chem. Theory Comput. **8**, 2564 (2012).

[156] T. A. Wesolowski, S. Shedge, and X. Zhou, Chem. Rev. **115**, 5891 (2015).

[157] P. Pulay and S. Saebø, Theoret. Chim. Acta **69**, 357 (1986).

[158] W. Li and P. Piecuch, J. Phys. Chem. A **114**, 6721 (2010).

[159] W. Li and P. Piecuch, J. Phys. Chem. A **114**, 8644 (2010).

[160] Z. Rolik and M. Kállay, J. Chem. Phys. **135**, 104111 (2011).

[161] D. G. Fedorov and K. Kitaura, J. Chem. Phys. **121**, 2483 (2004).

[162] H. Stoll, Phys. Rev. B **46**, 6700 (1992).

[163] B. Paulus, P. Fulde, and H. Stoll, Phys. Rev. B **51**, 10572 (1995).

[164] H. Stoll, B. Paulus, and P. Fulde, J. Chem. Phys. **123**, 144108 (2005).

[165] R. M. Richard and J. M. Herbert, J. Chem. Phys. **137**, 064113 (2012).

[166] F. A. Bischoff, R. J. Harrison, and E. F. Valeev, J. Chem. Phys. **137**, 104103 (2012).

[167] N. Mardirossian, J. D. McClain, and G. K.-L. Chan, J. Chem. Phys. **148**, 044106 (2018).

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

[169] G. Schmitz and C. Hättig, J. Chem. Theory Comput. **13**, 2623 (2017).

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

[171] Q. Ma, M. Schwilk, C. Köppl, and H.-J. Werner, J. Chem. Theory Comput. **13**, 4871 (2017).

[172] P. Pinski and F. Neese, J. Chem. Phys. **148**, 031101 (2018).

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

[174] M. Schwilk, Q. Ma, C. Köppl, and H.-J. Werner, J. Chem. Theory Comput. **13**, 3650 (2017).

[175] Q. Ma and H.-J. Werner, J. Chem. Theory Comput. **14**, 198 (2018).

[176] M. Häser and J. Almlöf, J. Chem. Phys. **96**, 489 (1992).

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

[178] D. Kats, D. Usvyat, and M. Schütz, Phys. Chem. Chem. Phys. **10**, 3430 (2008).

[179] G. Hetzer, P. Pulay, and H.-J. Werner, Chem. Phys. Lett. **290**, 143 (1998).

[180] G. Hetzer, M. Schütz, H. Stoll, and H.-J. Werner, J. Chem. Phys. **113**, 9443 (2000).

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

[182] C. Hättig, D. P. Tew, and B. Helmich, J. Chem. Phys. **136**, 204105 (2012).

[183] J. W. Boughton and P. Pulay, J. Comput. Chem. **14**, 736 (1993).

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

[185] W. Meyer, J. Chem. Phys. **58**, 1017 (1973).

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

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

[188] F. A. Bischoff and E. F. Valeev, J. Chem. Phys. **139**, 114106 (2013).

[189] R. Ahlrichs, F. Driessler, H. Lischka, V. Staemmler, and W. Kutzelnigg, J. Chem. Phys. **62**, 1235 (1975).

[190] G. Knizia, J. Chem. Theory Comput. **9**, 4834 (2013).

[191] J. Pipek and P. G. Mezey, J. Chem. Phys. **90**, 4916 (1989).

[192] C. Edmiston and K. Ruedenberg, Rev. Mod. Phys. **35**, 457 (1963).

[193] C. Edmiston and K. Ruedenberg, J. Chem. Phys. **43**, S97 (1965).

[194] G. Rauhut, P. Pulay, and H.-J. Werner, J. Comput. Chem. **19**, 1241 (1998).

[195] M. Schütz and F. R. Manby, Phys. Chem. Chem. Phys. **5**, 3349 (2003).

[196] M. Schütz, G. Hetzer, and H.-J. Werner, J. Chem. Phys. **111**, 5691 (1999).

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

[198] W. Meyer, Theoret. Chim. Acta **35**, 277 (1974).

[199] W. Meyer, in *Methods of Electronic Structure Theory*, Modern Theoretical Chemistry (Springer, Boston, MA, 1977) pp. 413–446.

[200] P. R. Taylor, G. B. Bacskay, N. S. Hush, and A. C. Hurley, Chem. Phys. Lett. **41**, 444 (1976).

[201] R. Ahlrichs and F. Driessler, Theoret. Chim. Acta **36**, 275 (1975).

[202] R. Ahlrichs and W. Kutzelnigg, Theoret. Chim. Acta **10**, 377 (1968).

[203] M. Jungen and R. Ahlrichs, Theoret. Chim. Acta **17**, 339 (1970).

[204] M. C. Clement, J. Zhang, C. A. Lewis, C. Yang, and E. F. Valeev, arXiv:1803.09135 [physics] (2018), arXiv:1803.09135 [physics] .

[205] H.-J. Werner and F. R. Manby, J. Chem. Phys. **124**, 054114 (2006).

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

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

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

[209] Y. Kurashige, J. Yang, G. K.-L. Chan, and F. R. Manby, J. Chem. Phys. **136**, 124106 (2012).

[210] G. Schmitz, B. Helmich, and C. Hättig, Mol. Phys. **111**, 2463 (2013).

[211] A. Mukhopadhyay, S. S. Xantheas, and R. J. Saykally, Chem. Phys. Lett. **700**, 163 (2018).

[212] R. J. Saykally and D. J. Wales, Science **336**, 814 (2012).

[213] G. Di Liberto, R. Conte, and M. Ceotto, J. Chem. Phys. **148**, 104302 (2018).

[214] Y. Wang, X. Huang, B. C. Shepler, B. J. Braams, and J. M. Bowman, J. Chem. Phys. **134**, 094509 (2011).