Title
Fast Algorithms for Exact Exchange

Permalink
https://escholarship.org/uc/item/553571b1

Author
Manzer, Samuel

Publication Date
2016

Peer reviewed|Thesis/dissertation
Fast Algorithms for Exact Exchange

by

Samuel Fenton Manzer

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Chemistry in the Graduate Division of the University of California, Berkeley

Committee in charge:
Professor Martin Head-Gordon, Chair
Professor William H. Miller
Professor James Demmel

Spring 2016
Fast Algorithms for Exact Exchange

Copyright 2016
by
Samuel Fenton Manzer
Abstract

Fast Algorithms for Exact Exchange

by

Samuel Fenton Manzer

Doctor of Philosophy in Chemistry

University of California, Berkeley

Professor Martin Head-Gordon, Chair

This thesis describes several new theoretical developments that facilitate the computation of the exact exchange energy, a vital component of accurate molecular simulations. The primary technique on which these developments are based is the resolution of the identity approximation, particularly the pair atomic resolution of the identity approximation (PARI). We prove that computation of exact exchange using the PARI approximation is variationally stable, and provide benchmarks of the performance and accuracy of our implementation. We then show that the most commonly used SCF convergence acceleration algorithm, DIIS, enables the design of a new fast exchange algorithm that we designate as occ-RI-K. Next, we combine the preceding occ-RI-K algorithm with the PARI approximation to create a linear-scaling exact exchange algorithm for the specific case of large weakly-interacting systems. Finally, we discuss our development of a high-level, object-oriented software library for block-sparse tensor operations. This library provides the underlying implementation for all of the algorithms discussed in this thesis.
To Claire, without whom I would never have completed this Sisyphean task.
Contents

List of Figures iv
List of Tables vi

1 Introduction 1
   1.1 The Born-Oppenheimer Approximation ................................. 1
   1.2 The Variational Principle .............................................. 4
   1.3 The Hartree-Fock Method ................................................ 4
   1.4 Electron Correlation ..................................................... 7
   1.5 Density Functional Theory .............................................. 12
   1.6 High-Performance Algorithms for Exact Exchange ..................... 17
   1.7 Thesis Outline .......................................................... 20

2 The PARI Approximation: Discussion and Efficient Implementation 22
   2.1 Introduction ............................................................. 22
   2.2 Theory ................................................................. 27
   2.3 Results .................................................................. 32
   2.4 Conclusions .............................................................. 38
   2.5 Acknowledgements ....................................................... 39

3 The occ-RI-K Algorithm 40
   3.1 Introduction ............................................................. 40
   3.2 Theory ................................................................. 43
   3.3 Results and Discussion .................................................. 47
   3.4 Conclusions .............................................................. 52
   3.5 Acknowledgements ....................................................... 52

4 Efficient Exact Exchange for Intermolecular Interactions 53
   4.1 Introduction ............................................................. 53
   4.2 Theory ................................................................. 55
   4.3 Results and Discussion .................................................. 61
5 **Fully General Sparse Tensor Framework for Electronic Structure Theory** 67

5.1 Introduction .......................... 67
5.2 Theory .................................. 69
5.3 Results and Discussion ............... 81
5.4 Conclusions ........................... 84
5.5 Acknowledgements ....................... 84

References .............................. 85
List of Figures

2.1 Hydrogen-terminated 3x6 graphene lattice upon which QZ timings were performed. 34
2.2 Wall time for the two K builds in the second SCF iteration for a hydrogen-terminated 3x6 graphene lattice with the range-separated ωB97X-V functional. Speedups relative to the integral-driven K build for the various RI-methods are shown inset. The first iteration is approximately 15% more expensive as of this writing due to initialization costs, but only for smaller systems. RI-K and ARI-K timings were calculated by timing one K build and scaling the result by 2, as the cost of the short and long-range K builds is essentially identical for these methods. 35
2.3 Wall time for the two K builds in the second SCF iteration for various N×6 graphene lattices, performed in the cc-pVTZ basis. RI-K and ARI-K timings were calculated by timing one K build and scaling the result by 2, as the cost of the short and long-range K builds is essentially identical for these methods. 36
2.4 Sparsity in of $D_{iQ}^{\mu}$ tensor for a series of linear alkanes using the cc-pVTZ basis set. Screening of the AO-basis RI fit coefficients was performed using a threshold of $10^{-6}$. 37
2.5 Wall time for a single second SCF iteration K build for several linear alkanes, performed in the cc-pVTZ basis. An integral threshold of $10^{-12}$ was used in conjunction with a fit coefficient threshold of $10^{-6}$. 37
2.6 Comparison of varying dimensions of tensors in the rate determining step for AO and MO algorithms. [NB2] was computed using a screening threshold of $10^{-12}$. 38
3.1 Histogram of change in number of SCF iterations for the calculations in the test set between occ-RI-K and exact integral-direct K. 49
3.2 Timings for the second K build of various depths of n-by-6 graphene sheets in the cc-pVQZ basis set using a single thread. An integral screening threshold of $10^{-10}$ was used for all calculations. 50
3.3 Timings for the second K build of various size octahedral diamondoids in the cc-pVDZ basis set using a single thread. The points correspond to $n = 2, 3, 4, 5$ in the compound series $C_{(4n^3-n)/3}H_{4n^2}$ used by Strout and Scuseria[25]. An integral screening threshold of $10^{-10}$ was used for all calculations. 51
4.1 Density matrix breadth as a function of alkane chain length and screening threshold in the 6-31G* basis set. ................................................................. 57
4.2 Linear-scaling algorithm for prediction of significant output blocks of $C$, the output from contraction of a sparse tensor $A$ with a dense matrix $B$, such as $\sigma^{ij}$. 62
4.3 The ALMO-PARI-K algorithm. All steps are asymptotically linear scaling, with the exception of the final increment of $K$, which is trivial. ...................... 63
4.4 127-molecule water cluster used in timing benchmarks .................................. 65
4.5 One of the stacked benzene systems used as a scaling benchmark in this work. 66
4.6 Scaling benchmarks performed on increasing length stacks of benzene rings. The aug-cc-pVTZ basis set was used with the cc-pVTZ-JK RI basis set augmented with the diffuse functions from the rimp2-aug-cc-pVTZ RI basis set. An integral threshold of $10^{-11}$ was used. ........................................ 66

5.1 An example of the block-major sparse tensor storage format used in our library. The tensors correspond to a fictional system in which only the index tuples (0,1,0), (1,1,1), and (2,0,1) correspond to non-zero blocks. The numbers 5,10, and 15 are example index ranges that are used to define the sizes of the three 3 non-zero blocks illustrated on the right. ........................................ 70
5.2 The conceptual representation of our sparse tree data structure and its literal multi-dimensional representation. ...................................................... 73
5.3 Permutation of subspaces 1 and 2 (i.e. second and third columns) of a sparse tree 73
5.4 Fusion of two lower-dimensional sparse trees which share a common index to form a higher-dimensional tree. In this example, two two-dimensional tensors with a common index type (lowest level of tree A; highest level of tree B) are fused to make a three-dimensional sparse tensor. ........................................ 74
5.5 Contraction of the second subspace of a tree to form a lower-dimensional tree. 75
5.6 Example of advantage of multi-dimensional sparsity relative to pair-wise sparsity. The red element is erroneously traversed in nested loops over the combined pair-wise sparsity, but not over the multi-dimensional original. 78
5.7 Conversion of symbolic tensor contraction expression to C++ code, as deployed in our development version of Q-Chem. ........................................ 79
5.8 Algorithm for performing fusion of sparsity across all tensors to form canonical set of trees. ................................................................. 80
5.9 Algorithm for grouping loops over indices by sparse coupling. ...................... 80
5.10 Nested loop execution algorithm for general sparse tensors. ...................... 80
5.11 A 46-molecule water cluster used in our performance benchmarks. .......... 82
5.12 Timings for all three contractions from Eqs. 5.13-5.15. Benzene stack calculations used the aug-cc-pVTZ[199] basis set with an integral threshold of $10^{-10}$. Reported times are for the first SCF iteration. ...................... 83
5.13 Stacked benzene molecules used to benchmark linear-scaling contractions 83
List of Tables

2.1 MO-basis algorithm for exchange matrix formation. The second column gives the operation cost for each step in terms of $[NB2]$ (number of significant orbital-basis function pairs, which is asymptotically linear in system size), $X$ (mean number of auxiliary basis functions per atom, independent of system size), $X$ (number of auxiliary basis functions), $o$ (number of occupied orbitals), and $[NBX]$ (number of significant orbital-basis to aux-basis function pairs, which is also asymptotically linear). 30

2.2 Errors in atomization energies (AE), ionization potentials (IP), electron affinities (EA), and proton affinities (PA) relative to no RI approximation for a subset of the G3-05 test set. All data is in kcal mol$^{-1}$. 32

2.3 Errors in counterpoise-corrected binding energies for the S66 set relative to no RI approximation, decomposed by interaction type. We use Řezáč’s original classification of H: Hydrogen-bonded, D: Dispersion, and O: Others[96]. All data is in kcal mol$^{-1}$. 33

3.1 The occ-RI-K algorithm, omitting the trivial final matrix multiplication steps. The second column gives the operation cost for each step in terms of $N$ (number of primary basis functions), $[NB2]$ (number of significant primary-basis function pairs, which is asymptotically linear in system size), $X$ (number of auxiliary basis functions), and $o$ (number of occupied orbitals). The computational cost scaling is given as a power of the size of the molecule, $M$, assuming a uniform choice of the AO and auxiliary basis sets. Memory demand is indicated in the same way. 46

3.2 occ-RI-K vs. PARI-K Accuracy comparison over a compilation of test sets, using the B3LYP/def2-QZVPPD level of the theory. Data in kcal/mol. 48

3.3 Contribution of various steps to overall occ-RI-K walltime for two systems and basis sets. Disk reads/writes are included in overall times for various steps, though their contribution is minor. An integral threshold of $10^{-10}$ was employed in both calculations. The total number of basis functions were 3510 and 4088, respectively. 51
4.1 Binding energy errors for the S66 set relative to SCF-MI with no PARI-K/occ-RI-K approximation. The column header acronyms are MAE = Mean Absolute Error, MSE = Mean Signed Error, and MAXE = Maximum Absolute Error. The last two columns denote the average absolute and percentage change in the number of SCF iterations, respectively. All units are in kcal mol$^{-1}$.  

4.2 Wall times in hours for the first K build of an SCF-MI calculation, using a guess of superimposed fragment densities. The aug-cc-pVTZ basis set was used with an integral screening threshold of $10^{-11}$ and the augmented cc-pVTZ-JK auxiliary basis described above.  

5.1 Percentage of non-zero elements in the half-transformed PARI fit coefficients $D_{(\mu,j)}$ defined in Eq. 5.6, illustrated in terms of different sparse trees. Percentages are based on block tuples, not on elements. The discrepancy between the first column and the “pair-wise” column is the sparsity lost by utilizing only a pair-wise representation. The aug-cc-pVTZ[199] basis set was used in conjunction with the cc-pVTZ-JK RI fitting basis set[39], and the half-transformed coefficients were screened with a threshold of $10^{-6}$. 
Acknowledgments

I thank Donghyun Lee and Alex Shearer for being steadfast friends. I also wish to thank my advisor, Martin Head-Gordon, for giving me the opportunity to learn many new things. Finally, I am thankful to all members of the Head-Gordon group for being a wonderful community, particularly everyone who attended my wedding: Eric Sundstrom, David Stück, Paul Horn, Narbe Mardirossian, Richard Julian Azar, Yuezhi Mao, and Franziska Bell. We sure had some good times.
Chapter 1

Introduction

Computational chemistry, broadly defined, seeks to predict and characterize chemical phenomena using computers. The applications of computational chemistry potentially span the entirety of the chemical sciences, from determining reaction mechanisms to increasing reaction yields and selectivity. Electronic structure theory, also often referred to simply as “quantum chemistry,” is a branch of computational chemistry concerned with the calculation of the quantum mechanical wave functions of molecular systems. This objective is often achieved by solving some approximation of the molecular Schrödinger equation. We shall now briefly detail the different layers that compose modern approximations to this equation.

1.1 The Born-Oppenheimer Approximation

If relativistic effects are neglected, as is appropriate for systems composed of light atoms, then the Hamiltonian operator for an arbitrary molecule (in atomic units) is:

\[ \hat{H} = \hat{T}_N + \hat{V}_{NN} + \hat{T}_e + \hat{V}_{Ne} + \hat{V}_{ee} \]  \hspace{1cm} (1.1)

where each term is defined as follows:
\[ \hat{T}_N = -\sum_A^{N_A} \frac{1}{2M_A} \nabla^2 R_A \]  
(Nuclear Kinetic Energy) \hspace{1cm} (1.2)

\[ \hat{V}_{NN} = \sum_A^{N_A} \sum_{B>A}^{N_A} \frac{Z_A Z_B}{\|R_A - R_B\|} \]  
(Nuclear Repulsion Energy) \hspace{1cm} (1.3)

\[ \hat{T}_e = -\sum_i^{N_e} \frac{1}{2} \nabla^2 r_i \]  
(Electronic Kinetic Energy) \hspace{1cm} (1.4)

\[ \hat{V}_{Ne} = -\sum_i^{N_e} \sum_A^{N_A} \frac{Z_A}{\|r_i - R_A\|} \]  
(Electron-Nuclear Attraction Energy) \hspace{1cm} (1.5)

\[ \hat{V}_{ee} = \sum_i^{N_e} \sum_{j>i}^{N_e} \frac{1}{\|r_i - r_j\|} \]  
(Electron-Electron Repulsion Energy) \hspace{1cm} (1.6)

Here, \( N_A \) is the number of nuclei, \( N_e \) is the number of electrons, and each \( R_A \) and \( r_i \) corresponds to the appropriate nuclear and electronic position vector, respectively. The molecular Schrödinger equation thus yields solutions that are functions of both the electronic and nuclear coordinates:

\[ \hat{H} \Psi (\{r_i\}, \{R_A\}) = E_{tot} \Psi (\{r_i\}, \{R_A\}) \] \hspace{1cm} (1.7)

The vast majority of electronic structure calculations utilize the Born-Oppenheimer approximation\[1\], which simplifies the molecular Schrödinger equation by decoupling the electronic and nuclear degrees of freedom. The Born-Oppenheimer approximation uses the following definition of the electronic Hamiltonian:

\[ \hat{H}^{el} = \hat{T}_e + \hat{V}_{Ne} + \hat{V}_{ee} + \hat{V}_{NN} \] \hspace{1cm} (1.8)

We shall label the eigenstates of this Hamiltonian as \( \{\psi_k^{el}(r; R)\} \), where \( R \) is now merely a parameter defining the particular molecular geometry to which these eigenstates correspond. The \( \{\psi_k^{el}(r; R)\} \) form a basis for our wave function Hilbert space, and thus for a particular fixed value of \( R \), our exact wave function may be expanded as:

\[ \Psi (r; R) = \sum_k c_k \psi_k^{el} (r; R) \] \hspace{1cm} (1.9)

For each different value of \( R \), the coefficients (and basis functions) in the expansion of \( \Psi (\{r_i\}, \{R_A\}) \) in Equation 1.9 change; thus, we may designate the coefficients as functions of \( R \) and write the wave function as:
\[ \Psi (r, R) = \sum_{k} c_k (R) \psi_k (r; R) \quad (1.10) \]

Beginning with the electronic Schrödinger equation:

\[ \hat{H}_{el} \psi_{el i} (r; R) = E_{el i} (R) \psi_{el i} (r; R) \quad (1.11) \]

we proceed to substitute the expansion from Equation 1.10 into the molecular Schrödinger equation. The action of the nuclear kinetic energy operator \( \hat{T}_N \) upon the expansion given in Equation 1.10 yields three derivative terms per entry in the sum:

\[ \hat{T}_N [c_k (R) \psi_k (r; R)] = c_k'' (R) \psi_k (r; R) + 2 c_k' (R) \psi_k' (r; R) + c_k (R) \psi_k'' (r; R) \]

The Born-Oppenheimer approximation is defined mathematically by the neglect of the latter two terms. Making this approximation allows us to obtain a second modified Schrödinger equation:

\[ \left[ \hat{T}_N + E_{el i} (R) \right] c_i (R) = E_{tot i} c_i (R) \quad (1.12) \]

As it is now obvious that, within this approximation, the coefficients \( \{ c_i (R) \} \) may be regarded as the eigenstates of a nuclear Hamiltonian, we designate them as “nuclear wave functions” \( \{ \chi_i (R) \} \). The solution to the full molecular Schrödinger equation within the Born-Oppenheimer approximation satisfies both the nuclear and electronic Schrödinger equations, and it thus has a simple product form:

\[ \Psi_{BO}^{i} (r, R) = \chi_i (R) \psi_i (r; R) \quad (1.13) \]

The nuclear wave functions are usually solved for using the harmonic approximation for \( E_{el i} (R) \); we shall not discuss them further. Our concern therefore lies fully with the solution of the electronic Schrödinger equation. Except in the case of very simple systems, we must always apply further judiciously chosen approximations in order to obtain electronic wave functions of sufficient accuracy at affordable computational cost. We now discuss a method by which such approximations are developed and validated.
1.2 The Variational Principle

Clearly, it is desirable to have a benchmark by which we may judge the accuracy of whatever approximations we choose to employ, even in cases in which experimental benchmark data may be unavailable. Such a benchmark is provided by the variational principle, which guarantees that the exact ground state energy for any Hamiltonian is always a lower bound to the approximate energy, provided that the approximate energy is computed as the normalized Hamiltonian expectation value corresponding to some approximate wave function:

\[ E_{0}^{\text{exact}} \leq E_{0} = \frac{\langle \hat{H} \rangle_{\Psi_{0}}}{\langle \Psi_{0} | \Psi_{0} \rangle} \]  

(1.14)

The variational principle provides us with a useful and general prescription for designing approximate electronic structure methods. First, we choose a particular mathematical form for our approximate wave function. Second, we minimize the variational energy expression with respect to all free parameters in our wave function. We shall now discuss the simplest of these approximate wave functions in common use - the Slater determinant. The process of variational optimization of a single Slater determinant wave function is known as the Hartree-Fock method, and forms the foundation of much of quantum chemistry.

1.3 The Hartree-Fock Method

The Hartree-Fock method[2] may be regarded as a mathematically rigorous formulation of molecular orbital theory. The fundamental approximation of Hartree-Fock theory is to describe each electron separately in terms of its own molecular orbital (MO) probability distribution. The simplest instantiation of such an approximation would be a Hartree product wave function:

\[ \Psi(x_{1}, x_{2} \ldots x_{n}) \approx \chi_{1}(x_{1}) \chi_{2}(x_{2}) \ldots \chi_{n}(x_{n}) \]  

(1.15)

where we have made the customary combination of the space and spin coordinates by defining spin orbitals as the product of a spatial orbital and a spin eigenfunction:

\[ \chi_{i}(x) = \psi_{i}(r) \alpha(\omega) \]  

(1.16)

The Slater determinant wave function is a transformation of the Hartree product to incorporate antisymmetry with respect to exchange of electronic spatial and spin coordinates, a formal property of the exact wave functions of fermionic systems. The Slater determinant for a set of molecular orbitals \( \{\chi_{i}\} \) is defined as:
\begin{align}
\Psi (x_1, x_2 \ldots x_n) &= \frac{1}{\sqrt{N!}} \det \begin{bmatrix}
\chi_1(x_1) & \chi_2(x_1) & \cdots & \chi_n(x_1) \\
\chi_1(x_2) & \chi_2(x_2) & \cdots & \chi_n(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
\chi_1(x_n) & \chi_2(x_n) & \cdots & \chi_n(x_n) 
\end{bmatrix} 
\end{align}

(1.17)

The Hartree-Fock method seeks the molecular orbitals (MOs), \{\chi_i\}, that constitute the minimum energy Slater determinant. Calculations with an even number of electrons are typically run in a “restricted” fashion, requiring each electron to pair with an opposite-spin counterpart in the same spatial orbital:

\[ \chi_i(x) = \psi_i(r) \alpha(\omega) \]
\[ \chi_{i+1}(x) = \psi_i(r) \beta(\omega) \]

In calculations on molecular systems, the spatial molecular orbitals are typically expanded using a set of atom-centered functions (most commonly Gaussians):

\[ \psi_i(r) = \sum_{\mu} C_{\mu i} \phi_\mu(r) \]

(1.18)

The functions \{\phi_\mu\} are referred to as the “atomic orbital” (AO) basis set. As is standard, we shall assume that they are real functions. Finding the minimum energy Slater determinant for a given basis set requires finding the optimal set of molecular spin orbitals \{\chi_i\} by minimizing the energy with respect to all of the \(C_{\mu i}\). For convenience, the optional constraint that the MOs remain orthonormal is often imposed on this optimization:

\[ \int dx \chi_i^*(x) \chi_j(x) = \delta_{ij} \]

(1.19)

Solving this constrained optimization problem yields the condition that the optimal spin orbitals are eigenfunctions of the Fock operator of the system:

\[ \hat{f}(x) = \hat{H}^{\text{core}}(x) + \sum_i \hat{J}_i(x) - \hat{K}_i(x) \]

(1.20)

where \(\hat{H}^{\text{core}}\) incorporates the kinetic energy and nuclear potential energy terms, the Coulomb operator \(\hat{J}_i\) corresponds to the classical Coulomb repulsion induced by the electron density of a particular orbital, and the exchange operator \(\hat{K}_i\) accounts for non-classical energetic effects of wave function antisymmetry:
where $r^{-1}_{12}$ is the Coulomb operator. The Fock operator has the form of an approximate Hamiltonian in which the explicit interactions between electrons have been “averaged out.” Hence, the Hartree-Fock method is also often referred to as “mean-field theory.” As the molecular orbitals appear both in the construction of the Fock matrix and in the solution to the eigenvalue problem that it defines, the Hartree-Fock equations must be solved in an iterative self-consistent fashion. Hartree-Fock calculations are therefore often referred to as self-consistent field (SCF) calculations.

The optimal MO coefficient matrix $C$ can be solved for by expressing the Fock operator eigenvalue problem in the atomic orbital basis, resulting in the the Roothaan equations:

$$\mathbf{FC} = \mathbf{SC}\varepsilon$$

where $S$ is the AO overlap matrix:

$$S_{\mu\nu} = \int d\mathbf{r} \phi_{\mu}(\mathbf{r}) \phi_{\nu}(\mathbf{r})$$

Solving the generalized eigenvalue problem (1.23) yields (for a linearly independent AO basis) $N_{basis}$ orthogonal functions. The lowest energy of these are occupied orbitals, out of which the final Slater determinant is constructed. The remaining high-energy orbitals are called “virtual orbitals,” and are employed in higher accuracy wave functions that are discussed later in this work.

Calculating the elements of $\mathbf{F}$ requires calculating the matrix elements of $\hat{J}$ and $\hat{K}$ in the atomic orbital basis, which are given by the following expressions:

$$J_{\mu\nu} = \sum_{\lambda\sigma} \left[ \int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_{\mu}(\mathbf{r}_1) \phi_{\nu}(\mathbf{r}_1) r_{12}^{-1} \phi_{\lambda}(\mathbf{r}_2) \phi_{\sigma}(\mathbf{r}_2) \right] P_{\lambda\sigma}$$

$$K_{\mu\nu} = \sum_{\lambda\sigma} \left[ \int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_{\mu}(\mathbf{r}_1) \phi_{\lambda}(\mathbf{r}_1) r_{12}^{-1} \phi_{\nu}(\mathbf{r}_2) \phi_{\sigma}(\mathbf{r}_2) \right] P_{\lambda\sigma}$$

where $\mathbf{P}$ is the density matrix in the AO basis:

$$P_{\lambda\sigma} = \sum_i C_{\lambda i} C_{\sigma i}^*$$
CHAPTER 1. INTRODUCTION

For compactness, we introduce the standard “chemist’s notation” for the two-electron four-center integrals appearing in the above expressions:

\[(\mu \lambda | \nu \sigma) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \phi_\mu (\mathbf{r}_1) \phi_\lambda (\mathbf{r}_1) \mathbf{r}_{12}^{-1} \phi_\nu (\mathbf{r}_2) \phi_\sigma (\mathbf{r}_2)\]  

(1.28)

These two-electron integrals are incredibly expensive to calculate due to their great number, and avoiding this problem will be the central theme of this thesis, which we shall develop in more detail later.

Hartree-Fock theory provides reasonably accurate estimates of molecular energies in absolute numerical terms, but not nearly accurate enough for making reliable chemical predictions. The reason for this may be seen by examining the result yielded by Hartree-Fock for the joint probability of finding an electron at a point \(\mathbf{r}_1\) given the presence of an electron at \(\mathbf{r}_2\). Integrating the Slater determinant yielded by Hartree-Fock theory for an opposite-spin two-electron system produces the following:

\[P_r(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{2} \int d\omega_1 \int d\omega_2 [\psi_1 (\mathbf{r}_1) \alpha (\omega_1) \psi_2 (\mathbf{r}_2) \beta (\omega_2) - \psi_1 (\mathbf{r}_2) \alpha (\omega_2) \psi_2 (\mathbf{r}_1) \beta (\omega_1)]^2\]

This is an uncorrelated result, as would be obtained if the electrons were entirely non-interacting. This is clearly unphysical; Coulomb repulsion ensures that electronic motions will correlate to avoid one another. To correct this problem, it is necessary to incorporate electron correlation effects into our approximate wave function.

1.4 Electron Correlation

We shall now discuss the nature of the exact wave function, which shall provide us with a blueprint for how to correct the Hartree-Fock wave function by adding in electron correlation effects. We begin by considering the expansion of a single coordinate of the exact wave function, with the dependence on the remaining coordinates relegated to the coefficients as in our treatment of the Born-Oppenheimer approximation:

\[\Psi (\mathbf{r}_1, \mathbf{r}_2 \ldots \mathbf{r}_n) = \sum_{\mu} c_\mu (\mathbf{r}_1, \mathbf{r}_2 \ldots \mathbf{r}_{n-1}) \phi_\mu (\mathbf{r}_n)\]  

(1.29)

where we have neglected spin degrees of freedom for the present. In the limit of an infinite basis set of appropriate functions (such as Gaussians or plane waves), this expansion becomes mathematically exact. The Hartree-Fock molecular orbitals (both the occupied and the higher energy virtual orbitals) form an orthogonal basis that spans the same space as the
original AO basis functions. We may thus re-express our initial expansion in terms of these orbitals:

\[ \Psi (r_1, r_2 \ldots r_n) = \sum_k c_k (r_1, r_2 \ldots r_{n-1}) \psi_k (r_n) \]  

(1.30)

We may now sequentially expand the basis function coefficients in terms of the same set of orbital basis functions, eventually yielding the fully expanded expression:

\[ \Psi (r_1, r_2 \ldots r_n) = \sum_{k_1} \sum_{k_2} \cdots \sum_{k_n} c_{k_1 k_2 \ldots k_n} [\psi_{k_1} (r_1) \psi_{k_2} (r_2) \ldots \psi_{k_n} (r_n)] \]  

(1.31)

The above wave function is not antisymmetric. It can be made antisymmetric by imposing appropriate symmetry on coefficients whose indices are permutations of one another; for example, in a two-particle case, we must set \( c_{ij} = -c_{ji} \) and \( c_{ii} = 0 \). If the wave function is antisymmetrized in this manner, our expansion takes the form of a succession of determinants. Each of these determinants is typically written in a chemically intuitive way as a set of single-orbital “excitations” with respect to a ground state determinant; for example, if the ground state determinant is \( |\Psi_0\rangle \), a determinant incorporating an excitation of orbitals \( i \) and \( j \) to \( a \) and \( b \) would be written \( |\Psi_{ij}^{ab}\rangle \). The full expansion, then, consists of all possible excitations from the ground state:

\[ |\Psi\rangle = |\Psi_0\rangle + \sum_i \sum_a c_i^a |\Psi_i^a\rangle + \sum_{i<j} \sum_{a<b} c_{ij}^{ab} |\Psi_{ij}^{ab}\rangle \ldots \]  

(1.32)

Each one of these determinants is referred to as a “configuration” of \( n \) electrons within \( n \) orbitals. The linear variational method guarantees that the minimum energy wave function of this form may be obtained by diagonalizing the Hamiltonian operator in the determinant basis. The off-diagonal elements of this Hamiltonian matrix take the form of one configuration interacting with another; hence, methods in which the wave function is approximated as a linear combination of some number of determinants are referred to as Configuration Interaction (CI) methods. Diagonalization in the full basis of all determinants that may be formed from a given set of molecular orbitals is known as Full Configuration Interaction (FCI), and represents the best possible variational wave function that can be formed from a given atomic orbital basis set. FCI incorporates the maximal amount of electron correlation that can represented by the chosen AO basis set. As such, the correlation energy of a system in a given basis is defined as the difference between the FCI and HF results:

\[ E_{corr} = E_{FCI} - E_{HF} \]  

(1.33)
As both FCI and HF are variational methods, this difference is always negative, and may be interpreted as the energy lowering obtained by allowing the electrons greater freedom to avoid one another. Unfortunately, the number of determinants used in the FCI method is:

\[ N_{\text{det}} = \binom{N_{\text{basis}}}{n} \]
\[ = \frac{N_{\text{basis}}!}{n!(N_{\text{basis}} - n)!} \]  

This renders the method intractable for all but very small systems. A natural course of action to extend the range of applicability of CI might be to truncate the determinant expansion in a systematic way. For example, one could include only the ground state and determinants formed by single or double excitations. Truncating the determinant expansion reduces the computational cost, but also destroys the accuracy of the method. To see why the accuracy is so adversely impacted, it is necessary to define the property of size-extensivity. A method is size-extensive if, when applied to two infinitely separated fragments, it correctly yields the sum of the fragment energies as the energy of the full system:

\[ E_{AB}^\infty = E_A + E_B \]  

Given that the Hamiltonian for the system of infinitely separated fragments is \( \hat{H}_{AB} = \hat{H}_A + \hat{H}_B \), it is apparent that a size-extensive method must yield a supersystem wave function that is the product of the optimized wave functions for the individual fragments:

\[ (\hat{H}_A + \hat{H}_B) [|\Psi_A\rangle \otimes |\Psi_B\rangle] = (E_A + E_B) [|\Psi_A\rangle \otimes |\Psi_B\rangle] \]  

It is clear that this wave function is not accessible to truncated CI. If the truncation is carried out at \( n \) excitations, then the product of the optimal fragment wave functions as computed using the same truncated CI method will contain excitations of order \( 2n \); as these excitations are too high order to be found in the expansion of the full system wave function, this method is not size-extensive. Without being size-extensive, an electronic structure method is likely to be unable to accurately model any dissociation process. This issue prevents truncated CI from being useful in practice. Nevertheless, as FCI is the most accurate possible method for a given AO basis set, it is still often used as a benchmark for other methods in small test cases.

There are many alternative non-CI approaches to including electron correlation in electronic structure calculations. The coupled cluster (CC) family of methods can obtain very accurate results, provided that they are applied to systems small enough to make their steep computational cost bearable\[3, 4\]. Coupled cluster wave functions are constructed using products of excitation operators acting upon the Hartree-Fock ground state:
CHAPTER 1. INTRODUCTION

\[
|\Psi_{CC}\rangle = \left[ \prod_{q}^{N_{ex}} (1 + t_{q} \hat{\tau}_{q}) \right] |\Psi_{HF}\rangle \tag{1.38}
\]

Each of these excitation operators \( \hat{\tau}_{q} \) excites some set of occupied orbitals to a corresponding set of virtual orbitals. For example, we could define the following double excitation operator:

\[
\hat{\tau}_{ij}^{ab} |\Psi_{HF}\rangle = |\Psi_{ij}^{ab}\rangle \tag{1.39}
\]

Products of excitation operators create a combined excitation:

\[
\hat{\tau}_{ij}^{cb} \hat{\tau}_{ij}^{ab} |\Psi_{HF}\rangle = |\Psi_{ijk}^{ab}\rangle \tag{1.40}
\]

The unique power of coupled cluster methods comes from the fact that, while we may only include excitations up to a certain order, including singles excitations alone is sufficient to produce an expansion containing the complete set of determinants present in FCI. This seemingly magical feat is accomplished at affordable computational cost because the number of truly free parameters in this model is much lower than in FCI; while in FCI, the expansion coefficient of each individual determinant constitutes a variational parameter, only the much small number of \( t_{q} \) amplitudes is optimized in the coupled cluster framework.

The size-extensivity of the coupled cluster approach becomes evident when one considers an alternate form of the product formulation of the CC wave function. It is useful to note that applying the same excitation to the Hartree-Fock ground state twice produces a “vacuum state,” the equivalent of a zero wave function:

\[
\hat{\tau}_{ij}^{ab} \hat{\tau}_{ij}^{ab} |\Psi_{HF}\rangle = |0\rangle \tag{1.41}
\]

The following is therefore true for any excitation operator:

\[
1 + \hat{\tau}_{q} = e^{\hat{\tau}_{q}} \tag{1.42}
\]

As the different excitation operators commute with one another, we may therefore write the coupled cluster wave function in its standard exponentiated form:

\[
|\Psi_{CC}\rangle = \exp \left( \hat{T} \right) |\Psi_{HF}\rangle \tag{1.43}
\]

where \( \hat{T} \) is the combined excitation operator:

\[
\hat{T} = \sum_{q} t_{q} \hat{\tau}_{q} \tag{1.44}
\]
The size-extensivity of the method becomes clear when we consider that, for an infinitely separated pair of fragments, the total excitation operator for the system will simply be $\hat{T}_{AB} = \hat{T}_A + \hat{T}_B$, as inter-fragment excitations will always result in zero determinants at infinite separation. Due to the commutation of the subsystem excitation operators, we may write $\lvert \Psi_{CC}^{AB} \rangle = \left[ \exp \left( \hat{T}_A \right) \lvert \Psi_{HF}^A \rangle \otimes \exp \left( \hat{T}_B \right) \lvert \Psi_{HF}^B \rangle \right]$. This wave function clearly has the required product form to fulfill the size-extensivity condition given in equation 1.37. While there are many other details involved in the formal proof of size-extensivity, as well as in the actual optimization of the coupled cluster wave function, the above analysis suffices to illustrate the key point. While coupled cluster methods have many attractive properties, in practice their computational scaling is at least $O(N^6)$, limiting their applicability to relatively small systems, at least without further approximations.

A more economical wave function treatment of electron correlation is second-order Møller-Plesset perturbation theory (MP2)[5]. The MP2 method is derived within the framework of Rayleigh-Schrödinger perturbation theory[6], which is defined by the partitioning of the system Hamiltonian into an easily soluble zeroth-order piece $\hat{H}_0$ and a “perturbation” scaled by a parameter $\lambda$:

$$\hat{H} = \hat{H}_0 + \lambda \hat{V} \quad (1.45)$$

The eigenvectors and eigenvalues of $\hat{H}$ are then expanded as a MacLaurin series in $\lambda$:

$$\left[ \hat{H}_0 + \lambda \hat{V} \right] \left[ \lvert \Psi^{(0)} \rangle + \lambda \lvert \Psi^{(1)} \rangle + \ldots \right] = \left[ E^{(0)} + \lambda E^{(1)} + \ldots \right] \left[ \lvert \Psi^{(0)} \rangle + \lambda \lvert \Psi^{(1)} \rangle + \ldots \right] \quad (1.46)$$

The series of correction terms $E^{(1)}, E^{(2)}$ etc. can be derived by equating terms of the same order in $\lambda$. For example, the first order energy expression is simply:

$$E^{(1)} = \left\langle \Psi^{(0)} \mid \hat{V} \mid \Psi^{(0)} \right\rangle \quad (1.47)$$

The MP2 method is formulated by choosing the Fock operator to be the zeroth-order Hamiltonian:

$$\hat{H}_0 = \sum_i \hat{f} (x_i) \quad (1.48)$$

$$= \sum_i \hat{H}_{\text{core}} (x_i) + \sum_{ij} \hat{J}_j (x_i) - \hat{K}_j (x_i) \quad (1.49)$$

The perturbation $\hat{V}$ is then the difference between the exact Coulomb operator and the mean-field electronic interactions:
\[ V = \sum_{i<j} \frac{1}{r_{ij}} - \left[ \sum_{ij} J_j (x_i) - K_j (x_i) \right] \] (1.50)

To first order in the perturbation parameter \( \lambda \), the energy simply takes the form of the Hartree-Fock energy; thus, there is no such thing as first-order Möller-Plesset perturbation theory. However, the second-order correction has a useful form:

\[ E^{(2)}_{MP2} = -\sum_{i<j} \sum_{a<b} \frac{[(ia|jb) - (ib|ja)]^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \] (1.51)

MP2 can capture long-range electron correlation effects present in hydrogen bonding and van der Waals interactions with varying degrees of accuracy. The integrals \((ia|jb)\) require \(O(M^5)\) operations to compute, making MP2 cheaper than coupled cluster but still too expensive for use on very large molecules.

We have surveyed three types of methods for capturing electron correlation; there are many others. Only one approach, however, is sufficiently economical to see common use among experimentalists and theorists alike: density functional theory. We shall now discuss this field in detail.

### 1.5 Density Functional Theory

Given the high cost of obtaining accurate molecular energies using correlated wave functions, researchers have naturally sought more economical approaches. Density functional theory (DFT) provides the ability to treat electron correlation for very large systems, and constitutes the most widely-used class of electronic structure methods. The core principle of DFT is the abandonment of the cumbersome many-particle wave function that formed the basis of all methods in the previous sections of this work. In place of this wave function, DFT works with the much simpler electron density:

\[ \rho (r_1) = N \int d\omega_1 \int d\mathbf{x}_2 \ldots \int d\mathbf{x}_n |\Psi (x_1, x_2 \ldots x_n)|^2 \] (1.52)

The first and second Hohenberg-Kohn theorems (HK1 and HK2) provide the fundamental justification for DFT calculations[7]. HK1 establishes the existence of a 1-to-1 mapping between the ground state density and the external potential \( \hat{V}_{ext} \) (and thus the full Hamiltonian, ground state wave function, and ground state energy) of a system. HK2 provides a prescription for finding such a density in the form of the following variational principle:

\[ E_0 [\rho_{exact}] \leq T[\rho] + \int d\rho \rho (r) V_{ext} (r) + E_{ee} [\rho] \] (1.53)
where $T[\rho]$ is the kinetic energy corresponding to the density $\rho$ and $E_{ee}[\rho]$ is the electron repulsion energy. The HK2 variational principle, unfortunately, only enables energy minimization over the space of so-called v-representable densities; densities that correspond to a ground state wave function of a Hamiltonian with some external potential. In general, it is unclear how to assess whether a given trial density $\rho$ meets this condition, and thus whether the above variational principle applies to the trial energy. It is therefore desirable to loosen the variational search criteria from v-representable densities to N-representable densities, a category defined as densities corresponding to any N-particle antisymmetric wave function. This is accomplished by the introduction of the Levy constrained search formulation\cite{8}.

We now seek a means of decomposing the electron-repulsion energy term $E_{ee}[\rho]$ into “classical” and “non-classical” effects. This is accomplished by consideration of its explicit form in terms of the wave function corresponding to our trial density:

$$E_{ee}[\rho] = \sum_{i<j} \langle \Psi[\rho] | \hat{r}_{ij}^{-1} | \Psi[\rho] \rangle$$

(1.54)

We restrict ourselves to consideration of only one of the pair-wise Coulomb repulsion terms, $\hat{r}_{12}^{-1}$. It is useful to define the pair density $\rho_2(r_1, r_2)$ corresponding to expected number of electron pairs of any spin located at $r_1$ and $r_2$ This has the following expression:

$$\rho_2(r_1, r_2) \equiv N(N-1) \left[ \int d\omega_1 \int d\omega_2 \int dx_3 \ldots \int dx_n |\Psi(x_1, x_2 \ldots x_n)|^2 \right]$$

(1.55)

The Coulomb repulsion energy between electrons 1 and 2 has a simple expression in terms of the pair-density:

$$\langle \Psi[\rho] | \hat{r}_{12}^{-1} | \Psi[\rho] \rangle = \int dr_1 \int dr_2 \rho_2(r_1, r_2)$$

(1.56)

The conditional probability of finding an electron at $r_2$ may also be expressed in terms of this pair density:

$$Pr(r_2|r_1) = \frac{\rho_2(r_1, r_2)}{\rho(r_1)}$$

(1.57)

The classical picture of uncorrelated electronic motions yields a simpler expression for the conditional probability:

$$Pr^{cl}(r_2|r_1) = \rho(r_2)$$

(1.58)

The difference between the true conditional probability and the classical picture can be described as the “hole” cut out by the presence of the electron at $r_1$, and is customarily referred to as the “exchange-correlation hole”: 

$x$
\[ h_{xc} (r_2 | r_1) = Pr (r_2 | r_1) - \rho (r_2) \]  

(1.59)

Using the exchange-correlation hole definition, we may finally achieve our desired decomposition of the electronic interaction into classical and non-classical portions:

\[
\langle \Psi[\rho] | \hat{r}_{12}^{-1} | \Psi[\rho] \rangle = \int dr_1 \int dr_2 \frac{\rho (r_1) \rho (r_2)}{r_{12}} + \int dr_1 \int dr_2 \frac{\rho (r_1) h_{xc} (r_2 | r_1)}{r_{12}}
\]

(1.60)

\[
= J [\rho] + E_{xc} [\rho]
\]

(1.61)

where \( J [\rho] \) now embodies the classical Coulomb interactions and we have sequestered the difficult quantum behavior into \( E_{xc} [\rho] \). We may now rewrite the energy expression for our trial density in standard form:

\[
E_0 [\rho] = T [\rho] + E_{ext} [\rho] + J [\rho] + E_{xc} [\rho]
\]

(1.62)

The determination of \( T [\rho] \) represents a particularly vexing challenge, as the exact form of the kinetic energy as an explicit functional of the density remains unknown. The Kohn-Sham approach to DFT provides an ingenious solution to this problem\[9\]. The defining insight of Kohn-Sham approach is that mean-field wave functions from Hartree-Fock theory actually provide a reasonable estimate of the kinetic energy. In DFT, such wave functions are referred to as “non-interacting,” due to their lack of electron correlation. The exact kinetic energy expression for a non-interacting wave function is:

\[
T_{NI}[\{\chi_i\}] = \langle \Psi_{NI}[\{\chi_i\}] | \hat{T} | \Psi_{NI}[\{\chi_i\}] \rangle
\]

(1.63)

\[
= -\frac{1}{2} \sum_i \int d \mathbf{x} \ \chi_i^* (\mathbf{x}) \left[ \nabla^2 \chi_i (\mathbf{x}) \right]
\]

(1.64)

However, this expression is given in terms of orbitals, not the electron density. The gap between the density and wave function views of the kinetic energy may be bridged in the following way. We suppose that there exists a Slater determinant wave function that yields the exact density for our system upon integration. We then define a mean-field Hamiltonian in which explicit electron-electron repulsion has been removed (this is analogous to the Fock operator in Hartree-Fock theory), for which this Slater determinant is the ground state. This Hamiltonian defines a “non-interacting reference system.” The Kohn-Sham approach approximates the true kinetic energy as the kinetic energy of the non-interacting reference system:

\[
T_{NI}[\rho] = \langle \Psi_{NI} [\rho] | \hat{T} | \Psi_{NI} [\rho] \rangle
\]

(1.65)
The quantity $T_{NI}[\rho]$ is introduced into the DFT energy expression via a repartitioning of the terms:

$$E_0[\rho] = T_{NI}[\rho] + E_{ext}[\rho] + J[\rho] + E_{xc}[\rho] \quad (1.66)$$

where we have now redefined $E_{xc}[\rho]$ to also contain whatever errors occur in our kinetic energy treatment:

$$E_{xc}[\rho] = T[\rho] - T_{NI}[\rho] + \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{\rho(\mathbf{r}_1) h_{xc}(\mathbf{r}_2|\mathbf{r}_1)}{r_{12}} \quad (1.67)$$

We began this exposition of the Kohn-Sham approach by moving from the exact ground-state density to the wave function for a non-interacting reference system. In practice, this procedure is now inverted - the exact density is accessed by solving for the Slater determinant corresponding to an appropriately constructed non-interacting Hamiltonian. The path is thus $|\Psi_0[\rho_0]\rangle$, then $\rho_0(\mathbf{r})$ and finally $E_0[\rho_0]$, our end goal. In the evaluation of the energy, orbitals are used where they are needed, and densities are used elsewhere. The optimization of the Slater determinant is accomplished by iteratively solving the Kohn-Sham equations[10] for the molecular orbitals composing $|\Psi_0[\rho_0]\rangle$:

$$\left[\hat{T} + \hat{V}_{ext} + \hat{J} + \hat{V}_{XC}\right] \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r}) \quad (1.68)$$

where the hidden complexity of $E_{xc}[\rho]$ has been propagated by defining $\hat{V}_{xc} = \frac{\delta E_{xc}[\rho]}{\delta \rho}$.

While the Kohn-Sham framework has provided us with a satisfactory result for $T[\rho]$, we now face the remaining problem of determining $E_{xc}[\rho]$. Over the past several decades, an extraordinary variety of schemes for capturing the electron exchange and correlation effects within $E_{xc}[\rho]$ have been developed. Perdew’s “Jacob’s Ladder” categorization scheme provides an elegant ranking of different exchange-correlation functional classes in terms of their sophistication and computational cost[11]. At the bottom run of the ladder are functionals based on the local density approximation (LDA) or local spin density approximation (LSDA); these are simple functions of $\rho$. On the second rung reside the generalized gradient approximation (GGA) family of functionals, which are defined by their dependence on the density gradient $\nabla \rho$. The GGA class includes such popular functionals as BLYP[12, 13] and PBE[14], which are widely employed for the many problems in which functionals from higher rungs on the ladder are too expensive. The third rung on the ladder consists of the “hybrid” density functionals, which incorporate some weighted contribution from the exact exchange energy, a quantity that we shall define momentarily. This class contains the most popular density functionals for molecular problems, including B3LYP[12, 13, 15] and the Minnesota family of functionals[16, 17], as well as the recently developed $\omega$B97X-V functional[18]. While there are other groups in Perdew’s classification, we shall focus our
attention now upon the hybrids, as it is with these functionals that this thesis is principally concerned.

The construction of hybrid density functionals is justified by an interesting theoretical result: the adiabatic connection formula. A concise derivation of this formula has been presented by Becke\[19\]. The adiabatic connection formula is derived by considering a continuum of systems between the non-interacting reference system of Kohn-Sham theory and the fully interacting true physical system. Such a set of systems may be parameterized by a parameter $\lambda$:

$$\hat{H}^\lambda = \hat{T} + \hat{V}_{ext}(\lambda) + \lambda \sum_{i<j} \hat{r}_{ij}^{-1}$$

This continuum of systems may be exploited to replace the (unknown) exchange-correlation hole expression for the fully interacting system in terms of an integral over the exchange-correlation holes for all systems in the continuum:

$$\tilde{h}_{xc}(r_2| r_1) = \int d\lambda \ h_{xc}^\lambda (r_2| r_1)$$

yielding a useful expression for the exchange-correlation energy:

$$E_{xc}[\rho] = \int d\bf{r}_1 \int d\bf{r}_2 \frac{\rho(r_1) \tilde{h}_{xc}(r_2| r_1)}{r_{12}}$$

While the treatment of the fully interacting system remains as difficult as ever, we have now reduced its weight in the overall energy expression - it is only one point in the quadrature of the integral over $\lambda$ values. How exactly to perform such a quadrature is unclear; however, it is reasonable to guess that any accurate quadrature should include a contribution from the first endpoint, $\lambda = 0$. The function $h_{xc}^{\lambda=0}$ may be calculated exactly, and corresponds to the exchange hole created by Pauli repulsion. The energetic contribution from $h_{xc}^{\lambda=0}$, typically referred to as “exact exchange” or “Hartree-Fock exchange”, is just the Hartree-Fock exchange energy computed using the Kohn-Sham orbitals:

$$E_{K}^{\text{exact}} = -\frac{1}{2} \sum_{ij} (ij|ij)$$

Density functionals that incorporate some weighted contribution from exact exchange are referred to as “hybrid density functionals.” Hybrids have been shown to dramatically outperform non-hybrid “local” functionals for a wide variety of chemical properties, in particular molecular thermochemistry and kinetics\[18, 20, 21\]. This improvement in performance comes at a significant price.
The inclusion of this energy term in the self-consistent orbital optimization of the Kohn-Sham procedure requires calculation of the AO-basis exchange matrix:

\[ K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma} \]  

(1.73)

This quantity is of central importance because of the great expense required to calculate it, and forms the major subject of this thesis.

### 1.6 High-Performance Algorithms for Exact Exchange

We shall now briefly discuss why the computation of the exact exchange term forms the bottleneck of virtually all hybrid DFT calculations in realistic basis sets. For any density functional, evaluation of kinetic energy and nuclear potential terms involves only one-electron integrals, and thus is computationally trivial relative to the terms involving two-electron integrals. The components of \( E_{XC}[\rho] \), with the exception of exact exchange, are typically evaluated by numerical quadrature over one or two electronic coordinates. The atom-centered numerical integration scheme of Becke[22], combined with intelligent economization of the integration grid[23], prevents the evaluation of these components from being the bottleneck step of typical hybrid DFT calculations. One is then left with the evaluation of the exact exchange and Coulomb repulsion terms, both of which require the evaluation of the two-electron integrals:

\[ J_{\mu\nu} = \sum_{\lambda\sigma} (\mu\nu|\lambda\sigma) P_{\lambda\sigma} \]  

(1.74)

\[ K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma} \]  

(1.75)

Naive evaluation of the two-electron integrals scales as the fourth power of molecular size with a very large prefactor, making it untenable for large systems. Fortunately, many approximations have been developed to make computation of the two-electron integrals feasible. We shall now discuss several selected examples of such techniques.

**Integral Screening**

The two electron integral tensor in a Gaussian basis set is sparse, and intelligently predicting and utilizing this sparsity is the key to developing efficient electronic structure methods. This sparsity derives from the Gaussian overlap theorem, from which it can be seen that the product of two Gaussians contains a prefactor that decays exponentially with the distance between the two functions:
\[ \phi_\mu (\mathbf{r}; \mathbf{R}_A) \phi_\nu (\mathbf{r}; \mathbf{R}_B) \propto \exp \left[ -\| \mathbf{R}_A - \mathbf{R}_B \|^2 \right] \] (1.76)

The number of significant \( \mu \lambda \) or \( \nu \sigma \) pairs thus scales linearly for systems of sufficient size, and the total number of numerically significant two-electron integrals thus scales asymptotically quadratically with molecular size[24, 25]. For computational efficiency, determination of significant pairs of Gaussians is typically performed in terms of “shells” rather than individual elements, where a shell is a group of Gaussians sharing a common center and contraction scheme. The result of the screening process is therefore a set of significant \( \mu \lambda \) “shell pairs” to be evaluated. The screening process can be further extended to the determination of significant shell quartets by the employment of the following Schwarz inequality[26]:

\[ | (\mu \lambda | \nu \sigma) | \leq | (\mu \lambda | \mu \lambda) |^{\frac{1}{2}} | (\nu \sigma | \nu \sigma) |^{\frac{1}{2}} \]

While the number of numerically significant integrals scales quadratically, it is possible within the context of an SCF calculation to accomplish linear scaling evaluation of the two-electron integrals. This is accomplished in different ways depending on whether the integrals in question are used to evaluate the Coulomb or the exchange term. The Coulomb matrix expression may be interpreted as the Coulomb repulsion energy between two sets of charge distributions, one of which is scaled in magnitude by the appropriate density matrix elements. It is thus natural to consider its linear scaling evaluation by the fast multipole method[27], with appropriate modifications to account for the use of Gaussian charge distributions rather than point charges. White et al. have implemented such an algorithm, which they have designated as the “Continuous Fast Multipole Method,” or CFMM[28]. A variety of other highly efficient approaches specifically for the treatment of the Coulomb-repulsion term have also been developed[29, 30].

Linear scaling algorithms for the computation of the exchange matrix rely on the occurrence of sparsity in the density matrix. The decay of density matrix elements with molecular size was conjectured by Kohn[31], and has since been numerically verified[32]. Linear scaling integral screening algorithms have been derived that exploit this decay in conjunction with Gaussian overlap sparsity, and have been shown to provide acceptable performance for small basis sets[33, 34]. However, as we shall now discuss, the use of larger basis sets required for high accuracy calculations necessitate the use of alternate families of algorithms in order to obtain good performance.

**Resolution of the Identity Approximations**

The many accuracy benefits of hybrid density functionals are only fully realized if these functionals are employed in conjunction with triple-zeta or higher quality basis sets[20, 21]. Unfortunately, significant performance problems arise for exact exchange calculations when the quality of the basis set is increased. The effectiveness of integral screening algorithms
decreases drastically in large and diffuse basis sets, as the Gaussian overlap sparsity upon which the screening procedure relies becomes much less pronounced. Additionally, the use of larger basis sets generally increases the condition number of the overlap matrix, on which the sparsity of the density matrix depends\[35\]. This makes integral screening even less productive. In order to render large-basis set calculations on larger and larger systems tractable, new algorithmic approaches must be employed.

The most common approach to reduce the burden of large basis set exchange matrix calculations is the resolution of the identity (RI) approximation, so called because it may be viewed as the introduction of a projector that approximates a true identity operator in the limit of an infinite basis\[36\]. RI methods rely on the expansion of pairs of Gaussians, such as those appearing in the two-electron integrals, using an “auxiliary basis set” of additional Gaussian functions:

\[
\phi_\mu (r) \phi_\lambda (r) = \sum_Q C_\mu^\lambda_Q \chi_Q (r) \tag{1.77}
\]

RI approximations are also commonly referred to using the term “density fitting.” The RI fit coefficients, \(C_\mu^\lambda_Q\), appearing in Equation 3.3 are typically determined by minimizing the Coulomb repulsion energy of the fitting residual\[37\]:

\[
\eta_{\mu\lambda} (r) = \phi_\mu (r) \phi_\lambda (r) - \sum_Q C_\mu^\lambda_Q \chi_Q (r) \tag{1.78}
\]

\[
\frac{\partial (\eta_{\mu\lambda}|\eta_{\nu\sigma})}{\partial C_\mu^\lambda_Q} = 0 \tag{1.79}
\]

The \(C_\mu^\lambda_Q\) that satisfy this condition are given by\[38\]:

\[
C_\mu^\lambda_Q = \sum_P (\mu|P) (P|Q)^{-1} \tag{1.80}
\]

This “Coulomb metric RI” expansion thus amounts to projection of the original Gaussian pair into the auxiliary basis using the Coulomb metric. The core performance advantage provided by RI methods is the replacement of the pairs of functions appearing in the bra and ket of the two-electron integrals with single functions, thus reducing the dimensionality of the AO-basis tensors that must be computed. This, in turn, allows the algorithm steps to be ordered in a manner that compresses the integral tensors as early as possible, reducing the scaling with basis set size and improving the prefactor of the calculation.

Unfortunately, the naive algorithm for computing HF exchange using the RI approximation scales as the fourth power of molecular size, due to the necessity of recombining the three-center quantities and the lack of sparsity in the canonical molecular orbitals:\[39\]
While fourth-order scaling algorithms with a low prefactor are competitive with linear scaling approaches for quite large systems due to the slow onset of sparsity in large basis sets,[40] it is obviously desirable to obtain RI-based algorithms with lower scaling. However, further approximations are necessary to attain linear scaling for RI methods, and we shall discuss those approximations in detail later in this work.

1.7 Thesis Outline

This thesis contains a description of several new algorithms to compute exact exchange for hybrid DFT, as well as the description of new software tools that have enabled the development of these algorithms and, it is hoped, will enable other new and different work in the future. It is organized as follows:

Chapter 2: The PARI Approximation: Discussion and Efficient Implementation

We discuss the pair resolution of the identity approximation (PARI), a local density fitting method that has recently drawn significant interest. We address concerns regarding the stability and accuracy of the approximation, and discuss how it may be used to formulate a high-performance algorithm for exact exchange. We also present an efficient implementation of the algorithm, and discuss its advantages relative to competing implementations. This research has been published in *J. Chem. Theory Comput.*, 2015, 518-527.

Chapter 3: Fast, accurate evaluation of exact exchange: The occ-RI-K algorithm

We discuss how the most commonly used algorithm for converging SCF calculations, direct inversion of the iterative subspace (DIIS), actually requires only an economized form of the exchange matrix to reproduce the exact energy and DIIS error vector. We show how this technique may be exploited to formulate a powerful new RI algorithm for exchange, and provide results showing dramatic improvements in performance. We demonstrate that this algorithm preserves accuracy and efficient convergence for the overall SCF calculation. This work has been published in *J. Chem. Phys.*, 2015, 143, 024113.
Chapter 4: Efficient Exact Exchange for Intermolecular Interactions

Weakly-interacting systems are amenable to a wave function description containing additional advantageous sparsity. The SCF for Molecular Interactions (SCF MI) method incorporates such sparsity by restricting the amount of charge transfer allowed between weakly-interacting fragments, simplifying the structure of the wave function. Utilizing the techniques from the previous two chapters, we provide a new algorithm to compute exact exchange with linear scaling within the SCF MI approximation. We demonstrate that our new algorithm provides significant speedups relative to existing linear scaling approaches.

Chapter 5: Fully General Sparse Tensor Framework for Electronic Structure Theory

Block-sparse tensors appear in many contexts in electronic structure theory, and the lack of robust software infrastructure to represent these objects leads to brittle, difficult-to-maintain implementations of many methods. We provide an object-oriented, symbolic C++ library to treat block-sparse tensor operations. We demonstrate that our library captures more sparsity than competing implementations due to its support for sparse index couplings of arbitrary dimensionality, and show that it obtains high floating point performance even for implementations of linear scaling methods.
Chapter 2

The PARI Approximation: Discussion and Efficient Implementation

2.1 Introduction

Modern Kohn-Sham Density Functional Theory\cite{7, 9} has become the most widely used electronic structure method. A crucial contribution to this success was Becke’s insight into the role of exact exchange in approximations to the adiabatic connection formula\cite{41}. This discovery led to the creation of the class of methods known as “hybrid” functions that include some (usually empirically determined) weighted contribution from exact exchange\cite{12, 15, 42}, as exemplified by the widely used B3LYP functional. The use of hybrid functionals has yielded improved results for thermochemical properties\cite{41}, molecular geometries\cite{43}, and vibrational frequencies\cite{44, 45}. More recently developed hybrid functionals such as M06-2X have yielded significant additional improvements in accuracy\cite{16, 17}.

The role of exact-exchange has since been further refined with the introduction of range-separated hybrid functionals\cite{46, 47}, which hold particular promise for applications in the area of time-dependent density functional theory\cite{48, 49}. Local functionals and standard hybrid functionals suffer from incorrect long-range behavior of charge transfer excited states due to self-interaction error of the transferred electron\cite{50, 51}. Range-separated hybrid functionals partition the Coulomb operator into short and long-range components in order to smoothly increase the weight of exact-exchange towards unity in the long-range limit, thus recovering the correct asymptotic behavior. The development of systematically optimized functionals such as the ωB97X-D functional\cite{52} led to significant improvements in accuracy over first generation range-separated hybrids. Further noteworthy improvements were recently reported in the ωB97X-V functional\cite{18}.

Unfortunately, large basis sets are required to obtain the maximal accuracy which modern hybrid and range-separated hybrid functionals can provide. Many studies have demonstrated that a basis set of at least triple-zeta quality is required to obtain results reasonably close to the basis set limit with hybrid functionals\cite{20, 21, 53, 54}. At least triple, and if possible
quadruple zeta basis sets are needed for accurate thermochemistry, while at least augmented triple zeta basis sets are required for non-covalent interactions if no counterpoise corrections are performed[18]. In this scenario, computation of exact-exchange typically consumes the vast majority of the computation time for a hybrid DFT calculation.

The $K$ matrix required for the computation of exact exchange is computed via the expression:

$$K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma}$$  \hspace{1cm} (2.1)

The number of two-electron integrals appearing in (2.1) scales as the fourth power of molecular size. Schwarz-inequality-based integral screening algorithms can reduce this scaling to nearly $O(M^2)$ for sufficiently extended systems[24, 25]. In fact, linear scaling algorithms can be obtained by exploiting the density matrix sparsity that occurs in extended insulating systems, as is done in the SONX[55] and LinK[33, 56] methods. However, such linear scaling behavior is in practice seldom obtained for small-gap systems and/or with larger basis sets which can lead to overlap ill-conditioning and resulting loss of density matrix sparsity.

Even with the employment of screening algorithms, the scaling of the evaluation of (2.1) remains fourth order with respect to increasing the average number of basis functions per atom while molecule size is fixed. High-accuracy calculations in large basis sets are therefore most severely inhibited by the computational demands associated with exact exchange. With the introduction of still-more sophisticated treatments of electron correlation such as double-hybrid functionals[57–59], the need to compute efficiently with larger basis sets has only become more pressing.[60]

A variety of approaches have been used to improve the efficiency of constructing $K$ in large basis sets, the most common of which is the application of the resolution of the identity (RI) approximation, also frequently referred to as the density fitting or DF approximation. In the RI approximation, products of orbital basis functions are further expanded in an auxiliary basis:

$$|\mu\lambda\rangle \approx \sum_{Q} C_{Q}^{\mu\lambda} |Q\rangle$$  \hspace{1cm} (2.2)

This reduces the dimensionality of the integral tensors that must be calculated from four to three. The use of the RI approximation to compute exact exchange was introduced by Frücht[61], while the first direct implementation is due to Weigend[39]. Unfortunately, without further approximation the RI-K algorithm scales fourth order with molecular size, so performance deteriorates relative to conventional integral formation for extended systems. Hence, the application of RI-SCF has historically been limited to compact systems with large basis sets[39, 61, 62].

It has been suggested that RI methods may be better described as “inner projection” methods, as in the absence of a complete auxiliary basis the RI approximation does not amount to an insertion of a resolution of the identity, but rather the insertion of an inner
projection operator[63, 64]. The Cholesky decomposition (CD) of the two-electron integrals has been shown to be equivalent to the introduction of an inner projection operator onto the space spanned by the orthogonalized Cholesky vectors[65]. In this sense, the Cholesky decomposition may be seen as generating an ideal auxiliary basis for the particular problem at hand[64]. CD methods encompass many efficient algorithms. Since its introduction to the electronic structure community by Beebe and Lindberg[65], CD has been used to formulate a variety of efficient electronic structure algorithms, including the construction of the Fock matrix [63, 66] and the evaluation of the CCSD(T)[67] and SOS-MP2[68] energies. CD methods are of great interest because they produce an auxiliary basis that does not depend on the model chemistry for which it was optimized, and thus can deliver consistent performance across a variety of theoretical methods[69]. Additionally, CD methods allow for fine-grained control of the accuracy of the approximation simply by varying the decomposition threshold δ. This is in contrast to the RI method’s reliance on pre-optimized auxiliary basis sets, which limits its accuracy to that attainable by the largest auxiliary basis set trained for the given theoretical method. However, as currently available RI auxiliary basis sets yield fitting errors in the tens of µHartree per atom for absolute energies[39, 70–72], which is negligible compared to orbital basis set errors, the practical advantage provided by CD in terms of accuracy is often minimal.

CD methods have historically suffered from the inclusion of two-center functions in the resulting auxiliary basis, and therefore required the expensive evaluation of four-center integrals[69]. The situation improves if the CD basis is restricted to contain only one-center auxiliary functions; examples of such methods include the one-center Cholesky decomposition[69] (1C-CD) and the atomic Cholesky decomposition (aCD)[73]. The aCD and 1C-CD approaches have been used to obtain more compact sets of Cholesky vectors, with a size ≈5 times that of the orbital basis set found to yield acceptable accuracy[69, 73]. For comparison, the RI approximation typically requires an auxiliary basis set of 3-5 times the size of the orbital basis set to obtain accuracy in the µHartree range[39, 70–72]. Rigorous efficiency comparisons of RI and CD methods are scarce; in one such study, Weigend has shown that 1C-CD is outperformed significantly by RI-J[74]. However, more recent work on so-called “method-specific” CDs, in which Hadamard products involving the two-electron integrals are decomposed instead of the two-electron integrals themselves, has been shown to require a drastically smaller auxiliary basis[64], and has been found to be competitive with RI for the computation of exact exchange[74].

The unfavorable scaling of RI-SCF can be mitigated through the use of several types of local approximations. The RI expansion coefficients are typically obtained in the Coulomb metric by minimizing the Coulomb repulsion of the density residual[37]. The fitting coefficients thus obtained are highly non-local; the expansion of a product on a given center contains many numerically significant contributions from auxiliary functions on distant atomic centers. A root cause of this is local incompleteness of the auxiliary basis set, which leads to the inclusion of off-center auxiliary functions in order to obtain a better fit[73, 75]. The locality of the fit coefficients can be improved, while still retaining accuracy in the fit, by minimizing the residual repulsion based on an alternative local operator, a so-called “local
metric.” Perhaps the simplest and most drastic local metric approach to density fitting is the overlap metric first introduced by Baerends et al. [76]. The overlap metric was subsequently employed by Vahtras et al. in SCF calculations and was found to give errors an order of magnitude larger than the Coulomb metric[38].

In an attempt to combine the locality of the overlap metric with the accuracy of the Coulomb metric, Jung et al. introduced an attenuated Coulomb metric with rapid decay with interelectronic distance and similar RI errors to the full Coulomb metric[75]. The attenuated Coulomb metric has subsequently been employed by Reine in conjunction with localized MOs to implement reduced-scaling RI-SCF[77]. In an analogous development for CD methods, Aquilante et al. have shown that aCD methods yield a set of fit coefficients with spatial locality that increases as the decomposition threshold is decreased[73]. This is a consequence of a more complete CD auxiliary basis reducing the need to include off-center auxiliary functions in order to represent on-center products.

An alternative ubiquitous approach to constructing \( K \) economically is a family of RI approximations employing local fitting domains. Instead of expanding each orbital basis function product using a fitting basis set spanning the entire molecule, a subset of the full fitting basis is chosen for each orbital product. There have been numerous applications of local fitting domains to RI-SCF [62, 78–80]. In the first development in this area, Polly et al. developed a linear scaling algorithm using local atomic fitting domains in conjunction with localized orbitals [78]. Sodt and Head-Gordon have developed the ARI-K algorithm, wherein each pair of orbital basis functions is expanded using the auxiliary basis functions on the atoms in a certain radius around their parent atoms[62]. Very recently, Mejía-Rodríguez et al. have developed the LDF-HF algorithm, which builds upon the work of Polly et al. with localization of the molecular orbitals at each SCF cycle in order to obtain very compact local fitting domains[81].

Yet another unique approach to improving the locality and scaling of exchange matrix construction is the pseudospectral approach of Neese et al., in which one coordinate in the integrals from (2.1) is integrated by numerical quadrature[82]. The locality of the grid points coupling to the AO basis is exploited to obtain a conditionally linear scaling algorithm. This approach has been shown to yield equal or superior speedups to RI-K with comparable accuracy[83]. However, to our knowledge its performance has not been benchmarked relative to local density fitting methods.

The focus of this work is the construction of \( K \) using a drastically local form of the RI approximation, which we shall designate by the name used by Merlot et al. as the Pair Atomic Resolution of the Identity (PARI) approximation[79]. In this approach, a given pair of orbital basis functions is expanded using only auxiliary basis functions on either of the parent atoms. This fitting approach was explored in early work by Baerends et al.[84] and subsequently by several others[85–87]. When used with Dunlap’s robust fit functional[88], the PARI approximation has been found to give surprisingly accurate results[79, 80]. However, it has been demonstrated that the use of the PARI approximation for the construction of \( J \) can lead to unphysical “attractive electron” states[79, 80]. Merlot et al. have explained this behavior on the basis of the loss of a positive semidefinite integral tensor, and have
shown that this problem manifests for any local RI approximation employing the Dunlap functional. They attempted to correct this problem through local completion of the auxiliary basis using the Cholesky decomposition, but this approach resulted in the loss of all performance improvements derived from the PARI approximation[79]. Hollman et al. have taken a different approach by computing subsets of the two-electron integrals analytically, and employing the PARI approximation for the remainder[80]. We circumvent this variational stability issue entirely by using PARI-K alone, without PARI-J. We will analytically demonstrate that the use of PARI-K alone is variationally stable, and empirically demonstrate that PARI-K yields negligible errors compared to exact integral evaluation. The \( J \) matrix can in turn be constructed by one of several pre-existing efficient highly efficient algorithms[39, 89].

Merlot et al. have used the PARI approximation and the Dunlap functional[88] to create a new algorithm, PARI-K, for the construction of \( K[79] \). These authors report an impressive speedups of between five and seven-fold for their PARI-K algorithm relative to the LinK algorithm, measured on systems of up to 42 atoms in a triple-zeta basis set. However, their algorithm involves contraction of three-center RI integrals over the full auxiliary basis with density-contracted fit coefficients:

\[
K_{\mu\nu}^+ = \sum_{Q\sigma} d^\mu_{\sigma Q}(Q|\nu\sigma)
\] (2.3)

The auxiliary index \( Q \) is coupled by sparsity to the AO-basis index \( \mu \) as follows: for a given \( \mu \), all \( Q \) auxiliary basis functions on atoms with at least one basis function that has non-negligible overlap with \( \mu \) is significant. The scaling of this term is thus asymptotically quadratic, just like direct evaluation of the two electron integrals, which is highly desirable. However, it suffers from the same unfavorable scaling of \( O(N^4) \) with respect to basis set size for fixed molecular size, but with an improved prefactor based on the number of FLOPs required to sum over \( Q \) versus the FLOPs required to form a given two-electron integral directly. We shall demonstrate that dramatic improvements in performance can be obtained via an MO basis algorithm. In our algorithm, the two and three-center quantities evaluated separately in Merlot’s formulation are evaluated simultaneously in the MO basis. Unlike Merlot’s algorithm, the scaling of our algorithm remains fourth order; however, we demonstrate by numerical experiments that our algorithm performs fewer operations even for very extended 1D systems. We obtain speedups of up to 19x over conventional \( K \) construction for systems of up to 3570 basis functions. We also demonstrate that auxiliary basis sets no larger than those used in conventional RI-SCF are required when the PARI-K approximation is used to obtain chemical accuracy for thermodynamic properties and intermolecular interactions, despite the drastically local nature of the approximation.
2.2 Theory

Variational stability of RI-K and PARI-K

We shall abbreviate RI quantities using a tilde as follows:

$$|\tilde{\mu}\lambda\rangle = \sum_{Q} C_{Q}^{\mu\lambda}|Q\rangle$$ (2.4)

Dunlap has shown that the following “robust” approximation of the two-electron integrals yields errors that are quadratic in the error resulting from the RI approximation on the individual products[88]:

$$(\mu\lambda|\nu\sigma) = (\tilde{\mu}\tilde{\lambda}|\nu\sigma) + (\mu\lambda|\nu\sigma) - (\mu\lambda|\nu\sigma)$$ (2.5)

Merlot et al. have shown that the use of this robust formulation in conjunction with local RI fitting domains and/or non-Coulomb fitting metrics results in a two-electron integral tensor that is not positive semidefinite[79]. However, regardless of this complication, the following equality holds for the Dunlap formulation of any fitting method[88]:

$$(\mu\lambda|\nu\sigma) - (\tilde{\mu}\tilde{\lambda}|\nu\sigma) = (\mu\lambda - \tilde{\mu}\tilde{\lambda}|\nu\sigma - \tilde{\nu}\tilde{\sigma})$$ (2.6)

The difference of the exact and fitted exchange energies may thus be written as:

$$\Delta E_{K} = -\frac{1}{2} \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}(\mu\lambda - \tilde{\mu}\tilde{\lambda}|\nu\sigma - \tilde{\nu}\tilde{\sigma}) P_{\lambda\sigma}$$ (2.7)

Transforming to the MO basis, we have:

$$\Delta E_{K} = -\frac{1}{2} \sum_{ij} (ij - \tilde{ij}|ij - \tilde{ij})$$ (2.8)

By an early result of Slater[90], each term in the above sum is strictly positive. The exchange energy thus can only increase by the application of robust fitting, and densities that are negative eigenvalues of the two-electron integral matrix will be avoided by the SCF convergence algorithm. Local fit domains and non-Coulomb fitting metrics can therefore be applied to the computation of $K$ without risk of variational instability.

We now prove a stronger result concerning the relationship between standard Coulomb-metric RI using a global fit domain and that of any local fit approximation. Denoting coefficients for the MO pair $ij$ computed using global RI by $C^{G}_{ij}$ and those with local RI by $C^{L}_{ij}$, we may express their difference by:

$$C^{L}_{ij} = C^{G}_{ij} + \Delta$$ (2.9)
We now consider the expansion of a single exchange integral using the Dunlap functional and local RI fit coefficients, using $I_{ij}$ to denote the three-center integral vector for this MO pair and $V$ to denote the auxiliary basis metric:

$$
(ij|ij)_L = 2C^L_{ij} \cdot I_{ij} - C^L_{ij} \cdot VC^L_{ij}
$$

$$
= 2C^G_{ij} \cdot I_{ij} - C^G_{ij} \cdot VC^G_{ij} + 2\Delta \cdot (I_{ij} - VC^G_{ij}) - \Delta \cdot V\Delta
$$

The combination of the first two terms in the above expression yields the integral computed using global RI with the Coulomb metric. The next term vanishes by the inverse present in the Coulomb metric fit coefficients, and we thus obtain:

$$
(ij|ij)_L = (ij|ij)_G - \Delta \cdot V\Delta
$$

The error term is non-negative, as $V$ is positive semidefinite. We therefore see that the exchange energy computed with local RI methods is greater than or equal to both the exact and global Coulomb RI variants:

$$
E^L_K \geq E^G_K \geq E^{\text{exact}}_K
$$

It is well known that exchange integrals are bounded by their corresponding Coulomb integrals:

$$
0 \leq (ij|ij) \leq (ii|jj)
$$

This inequality holds when both integrals are expanded using a global auxiliary basis. Our treatment expands the Coulomb term in the global auxiliary basis, with the exchange term treated in the pair basis. Our above result proving that $E^L_K \geq E^G_K$ therefore guarantees that the total electronic energy of the system cannot become negative.

Efficient MO basis PARI-K algorithm

Merlot et al. have applied Dunlap’s robust formulation in conjunction with the PARI approximation, in which orbital basis products between functions lying on the atoms $A$ and $B$ are expanded only using auxiliary basis functions on those two atoms:

$$
\tilde{|\mu A\lambda B\rangle} = \sum_Q C^\mu \lambda Q |Q\rangle
$$
matrix in the following convenient form:

\[ K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma} \]

\[ = \sum_{\lambda\sigma} \left[ (\mu\bar{\lambda}|\nu\sigma) + (\mu\lambda|\nu\bar{\sigma}) - (\mu\bar{\lambda}|\nu\bar{\sigma}) \right] P_{\lambda\sigma} \]

\[ = \sum_{\lambda\sigma} \left[ (\mu\bar{\lambda}|\nu\sigma) - \frac{1}{2} (\mu\lambda|\nu\bar{\sigma}) \right] P_{\lambda\sigma} + \sum_{\lambda\sigma} \left[ (\nu\bar{\lambda}|\mu\sigma) - \frac{1}{2} (\nu\lambda|\mu\bar{\sigma}) \right] P_{\lambda\sigma} \]

We then make the following definition:

\[ L_{\mu\nu} = \sum_{\lambda\sigma} \left[ (\mu\bar{\lambda}|\nu\sigma) - \frac{1}{2} (\mu\lambda|\nu\bar{\sigma}) \right] P_{\lambda\sigma} \] (2.16)

to yield the resulting expression for the exchange matrix:

\[ K_{\mu\nu} = L_{\mu\nu} + L_{\nu\mu} \] (2.17)

The problem of efficiently computing the exchange matrix now reduces to the efficient computation of \( L \). We now introduce the RI approximation, presuming the use of the PARI fit coefficients in equation (5.5):

\[ L_{\mu\nu} = \sum_{\lambda\sigma} \left[ \sum_{Q} C_{Q}^{\mu\lambda} (\nu\sigma|Q) - \frac{1}{2} \sum_{QR} C_{Q}^{\mu\lambda} C_{R}^{\nu\sigma} (R|Q) \right] P_{\lambda\sigma} \] (2.18)

Grouping common terms yields:

\[ L_{\mu\nu} = \sum_{\lambda\sigma} \sum_{Q} C_{Q}^{\mu\lambda} \left[ (\nu\sigma|Q) - \frac{1}{2} \sum_{R} C_{R}^{\nu\sigma} (R|Q) \right] P_{\lambda\sigma} \]

Reverting to the MO basis and rearranging yields an intelligently grouped expression:

\[ L_{\mu\nu} = \sum_{\sigma} \sum_{i} \left[ \sum_{\lambda} C_{Q}^{\mu\lambda} M_{i}^{\lambda} \right] \left[ \sum_{\sigma} M_{i}^{\sigma} \left\{ (\nu\sigma|Q) - \frac{1}{2} \sum_{R} C_{R}^{\nu\sigma} (R|Q) \right\} \right] \] (2.19)

Here the MO coefficient matrix is \( M \). This expression is evaluated using the steps presented in Table 3.1.

Our algorithm is very similar to the LDF-HF algorithm of Mejía-Rodríguez et al., but without MO localization and with additional steps to account for the explicitly robust Dunlap formulation. Additionally, when all tensors cannot be held in memory, we batch over auxiliary functions rather than over molecular orbitals, allowing us to compute the three-center AO integrals only once per SCF cycle. For large systems, the step \( L_{\mu\nu} = \sum_{Qi} D_{i}^{\mu\lambda} H_{i}^{\nu\sigma} \)
### Table 2.1: MO-basis algorithm for exchange matrix formation

The second column gives the operation cost for each step in terms of [NB2] (number of significant orbital-basis function pairs, which is asymptotically linear in system size), \( \overline{X} \) (mean number of auxiliary basis functions per atom, independent of system size), \( X \) (number of auxiliary basis functions), \( o \) (number of occupied orbitals), and [NBX] (number of significant orbital-basis to aux-basis function pairs, which is also asymptotically linear).

<table>
<thead>
<tr>
<th>Step</th>
<th>Operations</th>
<th>Scaling</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before first SCF iteration:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate ((R</td>
<td>Q))</td>
<td>( X^2 )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>Calculate (C^{\mu\lambda}<em>{Q} = \sum</em>{R} (\mu\nu</td>
<td>R) (R</td>
<td>Q)^{-1})</td>
<td>[NB2] ( \overline{X} )</td>
</tr>
<tr>
<td>For each SCF iteration:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loop over atomic batches of auxiliary functions (Q):</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D_{i}^{\mu Q} = \sum_{\lambda} M^{\lambda}<em>{i} C^{\mu\lambda}</em>{Q})</td>
<td>( o \ [NB2] \overline{X} )</td>
<td>( M^2 )</td>
<td>( M^1 )</td>
</tr>
<tr>
<td>(E_{Q}^{\nu\sigma} = \sum_{R} C^{\nu\sigma}_{R} \left[ \frac{1}{2} (R</td>
<td>Q) \right])</td>
<td>[NB2] ( \overline{X} X )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>Calculate integral batch ((\nu\sigma</td>
<td>Q))</td>
<td>[NB2] ( X )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>(G_{Q}^{\nu\sigma} = (\nu\sigma</td>
<td>Q) - E_{Q}^{\nu\sigma})</td>
<td>[NB2] ( X )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>(H_{i}^{\nu Q} = \sum_{\sigma} M_{i}^{\sigma} G_{Q}^{\nu\sigma})</td>
<td>( o \ [NB2] \overline{X} )</td>
<td>( M^3 )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>(L_{\mu\nu} = \sum_{Q_{i}} D_{i}^{\mu Q} H_{i}^{\nu Q})</td>
<td>( oN \ [NBX] )</td>
<td>( M^3 )</td>
<td>( M^2 )</td>
</tr>
<tr>
<td>(K_{\mu\nu} = L_{\mu\nu} + L_{\nu\mu})</td>
<td>( N^2 )</td>
<td>( M^2 )</td>
<td>( M^2 )</td>
</tr>
</tbody>
</table>

**dominates the overall computation time**. As we demonstrate for a series of linear alkanes, the onset of sparsity in the MO-contracted fit coefficients \(D_{i}^{\mu Q}\) is sufficiently slow even for linear systems that it is generally not economical to utilize sparsity in this step, and the algorithm is thus effectively fourth order scaling, but with a significantly smaller prefactor than RI-K. The two fourth-order steps in RI-K are given by:

\[
B_{Q}^{\mu i} = \sum_{R} (\mu i|R) (R|Q)^{-\frac{1}{2}}
\]

\[
K_{\mu\nu} = \sum_{iQ} B_{Q}^{\mu i} B_{Q}^{\nu i}
\]

The formation of the \(B\) matrix is more expensive than its subsequent contraction by a factor of \(X/N\). The fourth-order step in our algorithm has the same cost as the latter fourth...
order step of RI-K, and the asymptotic speedup for our algorithm relative to RI-K is thus $1 + X/N$. It should be noted that our algorithm is capable of treating much larger systems on an economical timescale than the largest presented in this paper; for the purposes of this study, we were limited by the feasibility of timing the integral-driven code for comparison.

The Dunlap formulation converts first order error in the fitted products to second-order error in the approximate integral. We therefore may further economize our fit coefficients by using much looser screening criteria in their evaluation than in the overall calculation. Specifically, we may obtain the same effect as a given integral screening threshold by neglecting the fit coefficients of pairs whose integral estimates are less than the square root of said threshold. Thus, if the desired integral screening threshold for the calculation is $10^{-12}$, this level of accuracy can be preserved while setting the fit coefficients of products with integral estimates of approximately $10^{-6}$ uniformly to zero. Our results demonstrate that this approximation causes negligible loss of accuracy.

Additional Benefits for Range-Separated Hybrids

For range-separated functionals, both a short and long-range K build are required. For these functionals, we re-use the fit coefficients formed with the full Coulomb operator, and transfer the short and long-range operator dependence exclusively to the two and three-center integrals. Our expressions for each $L$ matrix are therefore:

\[
L_{\mu\nu}^{SR} = \sum_i \sum_Q \left[ \sum_{\lambda} C_{\mu\lambda}^{\mu\lambda} M_{\sigma}^{\mu} \right] \left[ \sum_{\sigma} M_{\sigma}^{\mu} \right] \left\{ (\nu\sigma|Q)^{SR} - \frac{1}{2} \sum_R \sum_{\lambda} C_{\nu\sigma}^{\mu\lambda} (R|Q)^{SR} \right\}
\]

\[
L_{\mu\nu}^{LR} = \sum_i \sum_Q \left[ \sum_{\lambda} C_{\mu\lambda}^{\mu\lambda} M_{\sigma}^{\mu} \right] \left[ \sum_{\sigma} M_{\sigma}^{\mu} \right] \left\{ (\nu\sigma|Q)^{LR} - \frac{1}{2} \sum_R \sum_{\lambda} C_{\nu\sigma}^{\mu\lambda} (R|Q)^{LR} \right\}
\]

It should be noted that a robust fit is still obtained due to the explicit use of Dunlap’s robust ansatz, and our results below demonstrate that the independent use of two sets of fit coefficients is unnecessary. To obtain similar accuracy with global RI-K using a single set of fit coefficients would require the explicit use of the Dunlap functional, at a significant performance penalty. We exploit this advantage to compress the evaluation of both exchange contributions into a single $K$ build as follows:

\[
(\nu\sigma|Q)^{C} = c_{SR} (\nu\sigma|Q)^{SR} + c_{LR} (\nu\sigma|Q)^{LR}
\]

\[
(R|Q)^{C} = c_{SR} (R|Q)^{SR} + c_{LR} (R|Q)^{LR}
\]

\[
L_{\mu\nu}^{C} = \sum_i \sum Q \left[ \sum_{\lambda} C_{\mu\lambda}^{\mu\lambda} M_{\sigma}^{\mu} \right] \left[ \sum_{\sigma} M_{\sigma}^{\mu} \right] \left\{ (\nu\sigma|Q)^{C} - \frac{1}{2} \sum_R \sum_{\lambda} C_{\nu\sigma}^{\mu\lambda} (R|Q)^{C} \right\}
\]

As may be seen from the above equations, only steps which are non-rate determining are duplicated. In the limit of large systems, our algorithm will therefore outperform RI-K by an
CHAPTER 2. THE PARI APPROXIMATION: DISCUSSION AND EFFICIENT IMPLEMENTATION

<table>
<thead>
<tr>
<th></th>
<th>AE</th>
<th>IP</th>
<th>EA</th>
<th>PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Absolute Error</td>
<td>.03</td>
<td>.0026</td>
<td>.052</td>
<td>.01</td>
</tr>
<tr>
<td>Mean Signed Error</td>
<td>-.03</td>
<td>.0004</td>
<td>-.052</td>
<td>-.009</td>
</tr>
<tr>
<td>Max Error</td>
<td>.22</td>
<td>.0108</td>
<td>.17</td>
<td>.09</td>
</tr>
</tbody>
</table>

Table 2.2: Errors in atomization energies (AE), ionization potentials (IP), electron affinities (EA), and proton affinities (PA) relative to no RI approximation for a subset of the G3-05 test set. All data is in kcal mol$^{-1}$.

additional factor of two for range-separated hybrids, leading to a net speedup of $2(1 + X/N)$. We demonstrate that this limit is reached for relatively small systems.

2.3 Results

All calculations were performed with a development version of the Q-Chem program[92, 93].

Accuracy

Hollman and Merlot have evaluated the accuracy of the PARI approximation for HF and B3LYP - we now wish to assess its accuracy in the context of modern density functionals. The $\omega$B97X-V functional was chosen due to its excellent and highly transferable performance for a variety of types of chemical interactions[18]. Our calculations were performed in the aug-cc-pVTZ orbital basis set using the corresponding RI auxiliary basis for exchange.[39] For an orbital basis set of this size, it is expected that the user will also wish to use an RI approach for the formation of the Coulomb matrix, and we therefore employ the RI-J algorithm in conjunction with PARI-K. In order to benchmark performance for thermochemical properties, a subset of the G3-05 test set[94, 95] was selected consisting of all compounds for which the auxiliary basis was supported. The errors for a variety of thermochemical properties are tabulated in Table 2.2. Larger than expected errors were found when employing the RI-J approximation for the calculation of electron affinities, so this approximation was not used for these table entries. An integral screening threshold of $10^{-12}$ was utilized, with the screening threshold for fit coefficients set to $10^{-6}$ as discussed above.

In order to assess the errors introduced by the PARI-K approximation when applied to intermolecular interactions, we have also calculated counterpoise corrected binding energies for the S66 test set[96], which are given in Table 2.3.

The results in Tables 2.2 and 2.3 demonstrate that PARI-K can be employed with the same size of auxiliary basis set used for standard RI-JK calculations with negligible loss of accuracy. It should be noted that the $\omega$B97X-V functional scales the short-range exchange energy by a factor of 0.167 and the long range by 1.0, thus reducing the error resulting from applying the PARI approximation relative to full exact exchange. Functionals with
CHAPTER 2. THE PARI APPROXIMATION: DISCUSSION AND EFFICIENT IMPLEMENTATION

<table>
<thead>
<tr>
<th>H</th>
<th>D</th>
<th>O</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Absolute Error</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>Mean Signed Error</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>Max Error</td>
<td>0.04</td>
<td>0.04</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 2.3: Errors in counterpoise-corrected binding energies for the S66 set relative to no RI approximation, decomposed by interaction type. We use Řezàč’s original classification of H: Hydrogen-bonded, D: Dispersion, and O: Others[96]. All data is in kcal mol$^{-1}$.

larger contributions from exact exchange will experience somewhat larger errors when used in conjunction with PARI-K.

Stability

All of our calculations were converged successfully to a threshold of $10^{-8}$. An initial guess for all systems was computed using the B97-D[97] LDA functional in the same basis set. We were able to successfully converge all systems in our truncated G3 set, including the systems for which Merlot et al. and Hollman et al. observe convergence problems (hexafluorobenzene, chloro-pentafluorobenze, 3-butyn-2-one, and 2-butyn). As expected from our proof of variational stability, we do not observe the presence of any “attractive-electron” solutions, as these arise from the computation of $J$ in the PARI approximation. The non positive-semidefinite integral tensor raises the possibility of the SCF encountering densities that lead to repulsive exchange interactions, but this is much less likely to pose a problem in practice as these densities will be actively avoided by the SCF optimizer. On the basis of our results, we assert that the PARI-K method may be reliably used with an appropriate guess.

Performance

We consider first the performance of our algorithm for a system to which it is “optimally suited” - a molecule of moderate spatial extent treated with a large basis set. The chosen example is a hydrogen-terminated three ring by six ring graphene lattice (shown in Figure 3.2) using the $\omega$B97X-V functional with the cc-pVQZ basis set. Our method is compared against the following competing approaches implemented in Q-Chem: an integral-driven K build using the LinK algorithm[33] (optimal for very large systems), the ARI-K algorithm[62] (optimal for mid-size systems in larger basis sets), and the RI-K algorithm[39] (optimal for compact systems in very large basis sets). The results of this comparison are shown in Figure 2.2. Our method outperforms all examined alternatives, obtaining a 19 times speedup over the conventional integral-driven K build and a 2.7 times speedup over RI-K. The speedup relative to RI-K will increase with increasing system size; however, our RI-K code is at present unable to handle the memory requirements of jobs larger than those presented here.
Figure 2.1: Hydrogen-terminated 3x6 graphene lattice upon which QZ timings were performed.

It should be noted that the RI-K and ARI-K timings were produced by timing a single \( K \) build and scaling the result by 2, the integral-driven timings are for a single \( K \) build without scaling, as the short and long range \( K \) builds can be combined by transferring all of the operator dependence to the fundamental integrals prior to contraction and application of recurrence relations.

We now assess the performance of our algorithm for as a function of system size in a smaller (TZ) basis set, where one would expect better performance from the lower-scaling integral-driven \( K \) build. We begin with acene-5 and extend the graphene sheet in two dimensions with additional chains of six rings along the \( y \)-axis. The results of this timing experiment are summarized in Figure 2.3. We are limited to a four by six aromatic lattice for RI-K due to the memory usage of our benchmark RI-K implementation. Our PARI-K implementation obtains a 6.4 times speedup relative to exact integrals for the largest lattice, a system with 2348 basis functions. This compares favorably with the speedups obtained by Merlot et al. for various smaller acenes. The superior efficiency of our MO algorithm thus appears to effectively compensate for its asymptotically higher scaling. Our algorithm outperforms RI-K by an increasing margin as the system size is increased, reaching a 3.4 times speedup for the largest lattice.

Our combined \( K \) build provides a dramatic advantage even for the smallest system in
Figure 2.2: Wall time for the two K builds in the second SCF iteration for a hydrogen-terminated 3x6 graphene lattice with the range-separated ωB97X-V functional. Speedups relative to the integral-driven K build for the various RI-methods are shown inset. The first iteration is approximately 15% more expensive as of this writing due to initialization costs, but only for smaller systems. RI-K and ARI-K timings were calculated by timing one K build and scaling the result by 2, as the cost of the short and long-range K builds is essentially identical for these methods.

Figure 2.3: A single K build performed for the 1x6 graphene sheet requires 312 s, whereas the combined short and long-range K build takes only 371 s. It should be recalled that the performance of RI-K and ARI-K will be twice as good for functions that are not range-separated hybrids.

Sparsity and Performance Considerations for PARI Algorithms

We shall now demonstrate empirically that the loss of sparsity in the RI fit coefficients caused by contracting them with dense matrices leads to severe difficulties in practically attaining the low scaling that is expected for such local fit domains. We focus on the sparsity of the most important sparse tensor appearing in our algorithm, the MO-transformed RI fit coefficients \( D^\mu_Q \), as a function of system size. As can be seen from the sparsity of this tensor for a series of linear alkanes, shown in Figure 2.4, the onset of sparsity in \( D^\mu_Q \) is much slower than might be expected given the immense locality of the PARI fit coefficients. The tensor is approximately 83 percent sparse for the longest alkane chain. As a consequence of this slow
Figure 2.3: Wall time for the two K builds in the second SCF iteration for various $N \times 6$ graphene lattices, performed in the cc-pVTZ basis. RI-K and ARI-K timings were calculated by timing one K build and scaling the result by 2, as the cost of the short and long-range K builds is essentially identical for these methods.

onset, exploitation of sparsity does not become appreciably favorable even for very extended systems. This leads to the timings shown in Figure 2.5. Note that even as the sparsity of $D_{\mu Q}^i$ increases, this gain for the sparse implementation is offset by increasing performance of the matrix-multiply routine called by the dense implementation as the matrix size grows. Thus, even for these ideal systems, the advantage gained by the formally lower-scaling algorithm which exploits sparsity is minimal. We acknowledge that a higher-performance sparse linear algebra implementation would yield a more favorable comparison. However, the tensor contraction expression in this step fundamentally takes the form of either a single matrix multiply in the dense case, or a series of what are essentially vector-matrix multiplies in the sparse case. The floating point performance difference is therefore bound to reflect the (usually large) difference in the performance of these two classes of operations for a given BLAS implementation.

It is important to recall that this unfortunate sparsity behavior is not a consequence of the MO-basis transformation per se; the density-matrix contracted fit coefficients appearing in the analogous step of Merlot’s algorithm will suffer the exact same slow onset of sparsity, assuming that density-matrix sparsity cannot be efficiently utilized. Rather, the lack of sparsity in the contracted fit coefficients may be viewed as a fundamental non-local behavior of the PARI approximation. Assuming a fully dense MO coefficient matrix (or density matrix for the analogous AO problem), an entry $D_{\mu Q}^i$ will be non-zero if even a single orbital-basis AO function on the atom of $Q$ has non-negligible overlap with $\mu$. This will hold for every single function $Q$ on the given atom. The decay of $D_{\mu Q}^i$ is thus controlled exclusively by the most diffuse AO functions in the orbital basis, and is slowed significantly by the fact that auxiliary functions are grouped in large atom blocks. Finally, we also note that our
CHAPTER 2. THE PARI APPROXIMATION: DISCUSSION AND EFFICIENT IMPLEMENTATION

Figure 2.4: Sparsity in of $D_{i}^{\mu Q}$ tensor for a series of linear alkanes using the cc-pVTZ basis set. Screening of the AO-basis RI fit coefficients was performed using a threshold of $10^{-6}$.

Figure 2.5: Wall time for a single second SCF iteration K build for several linear alkanes, performed in the cc-pVTZ basis. An integral threshold of $10^{-12}$ was used in conjunction with a fit coefficient threshold of $10^{-6}$.

algorithm outperforms linear-scaling integral evaluation even for very long alkane chains, thus suggesting that widely-held reservations about the use of RI-SCF methods due to their higher formal scaling may be somewhat misplaced.

Similar sparsity arguments are relevant to a discussion of the merits of an AO versus MO implementation. A reduction in the formal scaling of the algorithm can be obtained by utilizing the gaussian product sparsity present in the AO basis representation of the relevant tensors, as is done in Merlot’s PARI-K algorithm. However, based on our data for linear alkanes, we believe that utilizing AO sparsity would actually increase the raw operation cost of this algorithm, even for very extended systems. We illustrate this point by considering the relative sizes of the MO-transformed three-center integrals and the original AO three-center
integrals, accounting for sparsity. The ratio between these two quantities represents the ratio between the rate determining step of our algorithm and an AO algorithm such as Merlot’s. This ratio reduces to a comparison the quantities $oN$ and $[NB2]$. A plot of each of these quantities with respect to alkane length for the cc-pVTZ basis is shown in Figure 2.6. The permutational symmetry of the gaussian product indices is accounted for in this figure. On the basis of the relative size of these quantities, we propose that the AO-basis algorithm, despite formally scaling lower than our method, will in fact require more operations even for very extended systems, in addition to suffering from a significant efficiency penalty due to the high performance of dense linear algebra operations. This serves as a reminder that the tremendous compactness of the occupied MO basis and the associated high efficiency of dense linear algebra libraries must be properly accounted for when assessing the efficacy of competing lower-scaling implementations. Many similar points about the relative merits of lower-scaling methods in contrast to lower-prefactor, higher-scaling competitors are stated eloquently in a review by Neese[98]. On the basis of these concerns, it seems to us that the best path toward a lower-scaling PARI-K implementation is the use of localized occupied molecular orbitals.

### 2.4 Conclusions

Our MO-based implementation of the PARI-K approximation has been shown to significantly accelerate large scale range-separated hybrid DFT calculations. Benchmark results for thermochemistry and intermolecular interactions indicate that impressive accuracy can be obtained with the modern $\omega$B97X-V functional, while requiring an auxiliary basis set no larger than that used in conventional RI-HF. Our algorithm has been shown to out-perform commonly available alternatives for extended and compact systems in TZ and QZ basis sets.
2.5 Acknowledgements

Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, and Basic Energy Sciences. We also acknowledge partial support of this work from Q-Chem Inc through NIH SBIR grant no. GM096678. MHG and EE are part-owners of Q-Chem Inc. SFM acknowledges Patrick Merlot for helpful discussions, and acknowledges Claire Manzer for support in all things.
Chapter 3

The occ-RI-K Algorithm

3.1 Introduction

The widespread adoption of Kohn-Sham density functional theory[7, 9] (DFT) has allowed for electronic structure calculations of reasonable accuracy to be performed on systems of unprecedented size[99–101]. Perdew’s famous “Jacob’s Ladder” analogy elegantly captures the different components of the ideal high-accuracy density functional[11]. A crucial component of this hierarchy are the so-called “hybrid functionals”, containing a weighted contribution from the exact exchange energy. The importance of including exact exchange was initially recognized by Becke[12, 41], and exact exchange has since become a core component of a huge number of widely used modern density functionals[18, 102–105].

Unfortunately, exact exchange requires the computation of the numerous and expensive four-center two-electron repulsion integrals (ERIs):

$$(\mu\lambda|\nu\sigma) = \int \int d\mathbf{r}_1 d\mathbf{r}_2 \phi_\mu^*(\mathbf{r}_1) \phi_\lambda(\mathbf{r}_1) \phi_\nu^{-1}(\mathbf{r}_2) \phi_\sigma(\mathbf{r}_2)$$

(3.1)

to construct the exchange matrix, $K$:

$$K_{\mu\nu} = \sum_{\lambda\sigma} (\mu\lambda|\nu\sigma) P_{\lambda\sigma}$$

(3.2)

This is in contrast to highly efficient linear scaling evaluation of the classical Coulomb interactions, where four center ERIs can be rigorously avoided by multipole expansions of long-range interactions[28], and rigorous reformulation of the short-range interactions (the “J-matrix engine”[29, 106]). Naive construction of the full two-electron integral tensor scales as the fourth power of molecular size. However, it was demonstrated by Almlöf early in the development of the direct SCF method that the number of numerically significant two-electron integrals scales asymptotically only quadratically with molecular size, as there are linear number of overlapping pairs of Gaussian atomic orbitals[107]. This locality is most commonly exploited via the use of Schwarz-inequality based integral bounds[26, 108], though
CHAPTER 3. THE OCC-RI-K ALGORITHM

tighter bounds on numerically significant integrals have since been further developed[109–111].

In addition, the density matrix, $\rho(\mathbf{r}, \mathbf{r}')$ itself decays exponentially with distance, $|\mathbf{r} - \mathbf{r}'|$, in the position representation for systems with a non-negligible HOMO-LUMO gap[31, 112, 113]. For electronic structure theory based on localized atomic orbital (AO) basis functions, this translates into sparsity of the density matrix in the AO representation, as well as potential localization of the molecular orbitals (MOs). This sparsity is basis-set dependent and has been studied empirically[32, 114]. For appropriate insulating systems and compact basis sets, significant sparsity is present and may be exploited to formulate efficient conditionally linear-scaling integral-driven algorithms[33, 34, 115]. Such methods make hybrid DFT calculations possible on very large insulating systems, provided the basis set is compact.

However, the situation worsens dramatically in larger basis sets for two reasons. First, the number of significant two-electron integrals scales as the fourth power of the mean number of basis functions on a single atomic center, leading to a tremendous pre-factor increase. Second, as larger basis sets contain additional diffuse functions, the condition number of the overlap matrix becomes larger, leading to decreased sparsity as well, and thus loss of effectiveness of linear scaling. The combination of these two factors makes large basis exact exchange calculations almost prohibitively more expensive than small basis set calculations. Unfortunately, hybrid functionals must be used with basis sets of at least triple-zeta quality to obtain adequately converged thermodynamic quantities[20, 21]. In smaller basis sets there is no accuracy advantage to hybrid functionals. For the same reason, it has been shown that in order to obtain relatively basis-set independent functional parameters, density functionals should be trained (and thus used) with triple-zeta or larger basis sets[54]. Functionals trained close to the basis set limit yield significantly inferior performance in smaller basis sets[18, 58, 116]. This paper attempts to make progress on the challenging problem of constructing the $K$ matrix in these large basis sets.

Methods based on the resolution of the identity (RI) approximation (also frequently referred to as the “density fitting” or DF approximation) offer the best-established avenue to reducing the computational cost of large basis set DFT calculations. These methods expand the atomic orbital “primary basis” products as linear combinations of “auxiliary basis” functions:

$$
\phi^*_{\mu}(\mathbf{r}_1)\phi_{\lambda}(\mathbf{r}_1) = \sum_Q C^{n\lambda}_Q \chi_Q(\mathbf{r}_1)
$$

Highly efficient RI algorithms have been developed for the Coulomb problem[89, 117], along with the auxiliary Coulomb fitting basis sets necessary to use such algorithms[118–120]. Turning to exchange, for which appropriate auxiliary basis sets are also available[39, 72], the RI approximation results in a much more tractable expression for the $K$ matrix in which
only three and two-center quantities are present:

$$K_{\mu\nu} = \sum_{\lambda\sigma} \sum_{PQ} C_{P\lambda}^\mu C_{\nu\sigma}^Q (P|Q) P_{\lambda\sigma}$$

(3.4)

By reducing the dimensionality of the AO basis quantities, these methods attain significant improvements in performance for large basis sets. This comes at a formal cost; the subsequent recombination of these lower-dimensional quantities exhibits fourth-order scaling. However, the fact that the relevant quantities can be compressed in earlier steps by transformation to the MO basis allows for a significant advantage in the prefactor of the calculation\[39, 121\]. It should be noted that while RI-based algorithms for constructing $K$ are necessarily significantly more complex than those used for $J$, the underlying approximation is the same.

Extensive effort has been devoted to lowering the scaling of RI-SCF. These approaches may be broadly classified into two categories: local fit metrics\[38, 75–77\] and local fit domains\[62, 79–81, 121\]. In the local fit metric approach, the fit coefficients are determined by minimizing the self-repulsion between the fitted and exact densities as measured by a rapidly decaying local analog of the Coulomb operator, rather than the more typical full Coulomb metric. Local fit domain approximations are applied by expanding a given primary basis product only with the auxiliary basis functions within a predefined spatial domain; they differ from local metrics in that locality is achieved explicitly through a constraint rather than implicitly through the decay of the metric.

Two alternative approaches to the problem of large basis set $K$ matrix calculations should be mentioned. The first is the Cholesky decomposition, which has been shown to be equivalent to RI expansion of the two-electron integrals in a customized auxiliary basis\[65\]. It may thus be regarded as an idealization of the RI approximation in some sense, and the algorithms based on CD and RI are often closely related\[122\]. A second related approach is the pseudospectral or “semi-numerical” approach introduced to quantum chemistry by Friesner\[123\] and substantially improved by Neese et al.\[82\] Pseudospectral methods apply numerical quadrature to one of the two electronic coordinates in the two-electron integral formula (3.1). In essence, this replaces a given two-electron integral with a large number of one-electron integrals resembling nuclear attraction integrals. When initially compared to the RI-K method, the pseudospectral approach was found to yield comparable speedups and accuracy\[83\], while subsequent work has found superior speedups for large systems\[124\].

While efforts to reduce the scaling of RI-K are necessary in the long run, in this work we focus on reducing the prefactor of the calculation. Combined with the already significant prefactor advantage of standard RI-K, we obtain a method that outperforms linear scaling integral evaluation and existing alternative RI methods for systems of up to 4400 basis functions in quadruple-zeta quality basis sets, and even outperforms conventional algorithms in small double-zeta basis sets for a surprisingly large size regime. We achieve this by introducing a framework in which one constructs a much smaller “economized” exchange matrix that nonetheless yields energies, orbital first variations, and DIIS error vectors that are exact. Within this framework, we then adapt the RI-K method to yield a new algorithm, termed the occ-RI-K method, which is intrinsically more efficient than RI-K. While our new
approach can be combined with local RI approximations through the use of localized MOs, here we concentrate on the standard global Coulomb RI approximation. We demonstrate that the resulting algorithm is both faster and more accurate than our previous PARI-K implementation[121], as well as much faster and just as accurate as RI-K itself.

3.2 Theory

**Economization of the K matrix**

Using standard spin-orbital notation, the exchange energy, $E_K$, which enters a global hybrid functional in scaled form (i.e. $0.54E_K$ for M06-2X or $0.2E_K$ for B3LYP) can be written in terms of $K$, defined in Eq. 3.2, as follows:

$$E_K = -\sum_{\mu\nu} c_{\mu i} K_{\mu\nu} c_{\nu i}$$

(3.5)

where we have expanded the AO density matrix $P_{\mu\nu} = \sum_i c_{\mu i} c_{\nu i}$ in terms of the MO coefficients $c_{\mu i}$. Evidently to evaluate the energy, it is possible to simply evaluate the diagonal elements of the exchange matrix in the occupied block, $K_{ii}$, rather than the full matrix in the AO representation, $K_{\mu\nu}$.

However, more information is required to evaluate the orbital gradient of $E_K$, as needed for performing an SCF calculation. Variations of the MO coefficients away from our current set are given as a unitary transformation, $U$, which is parameterized in terms of the exponential of a skew-symmetric matrix composed of mixings, $\Delta_{ai}$, between occupied orbitals, $i$, and empty orbitals, $a$.

$$U = \exp [\Delta - \Delta^T]$$

(3.6)

It is then straightforward to show[125] that

$$\frac{\partial E_K}{\partial \Delta_{ai}} = 2K_{ai}$$

(3.7)

Evidently the gradient of the exact exchange energy requires only the occupied-virtual block, $K_{ia}$, rather than the full matrix in the AO representation, $K_{\mu\nu}$. Together, the energy and the gradient require only $K_{ii}$, rather than the full $K_{\mu\nu}$.

The Direct Inversion of the Iterative Subspace (DIIS) method introduced by Pulay is perhaps the most widely used convergence acceleration method[126, 127]. For SCF convergence acceleration, the standard form of DIIS[127] accumulates a sequence of error vectors defined as the AO-basis representation of the commutator $[\hat{f}, \hat{\rho}]$. The commutator, which is proportional to the SCF orbital gradient (cf Eq. 3.7 for the exact exchange gradient), vanishes at SCF convergence. The AO basis commutator at the $m^{th}$ iteration is:

$$e_m = F_m P_m S - SP_m F_m$$

(3.8)
where $S$ is the AO overlap matrix. DIIS extrapolation to obtain a new candidate solution is performed by minimizing a weighted linear combination of error vectors, $\sum_m w_m e_m$, subject to the normalization constraint, $\sum_m w_m = 1$. The extrapolated Fock matrix, $F_{\text{extrap}} = \sum_m w_m F_m$ is then diagonalized to define the updated density matrix, $P_{m+1}$, and thus $F_{m+1}$, etc.

Since the Fock matrix appears in Eq. 3.8 only when multiplied by the density matrix, a sequence of DIIS iterates can be correctly evaluated using only the occupied row-block of the $K$ matrix. This yields exactly the same set of DIIS coefficients as if the full $K$-matrix were used to evaluate the error vector. Hence construction of the full AO-basis K-matrix is potentially wasteful, as only the entries $K_{\mu\nu}$ are required to obtain an exact DIIS error vector and exchange energy. In addition, it is trivial to show that the occupied eigenvalues are exactly reproduced. This is all that is typically required in a standard DFT calculation, and therefore exact exchange algorithms can and should be developed to exploit this fact. There are two caveats to note. First, diagonalization steps that neglect the virtual-virtual part of $K$ will not be identical with steps that include it. Second, the virtual eigenvalues will not be exactly reproduced, and thus a single full K-build is required for post-SCF purposes such as double hybrid density functional theory, or the random phase approximation. We note that the Eq. 3.8 suggests that all other components of the Fock matrix, not just the exchange matrix, may be utilized in half-transformed form; however, this would likely be a much more drastic approximation, as the virtual-virtual block of the Fock matrix is qualitatively correct when only the virtual-virtual contribution from $K$ is neglected. Assessment of this more aggressive approach is beyond the scope of this work.

The compressed K-matrix, $K_{\mu\nu}$ can be employed to build an AO basis quantity as follows. We partition the exchange operator into terms involving projectors onto the occupied subspace (the projector is $\hat{\rho}$ of course) and the virtual subspace (the projector is $\hat{q} = (1 - \hat{\rho})$):

$$
\hat{K} = (\hat{\rho} + \hat{q}) \hat{K} (\hat{\rho} + \hat{q})
= \hat{\rho} \hat{K} \hat{\rho} + \hat{q} \hat{K} \hat{q} + \hat{q} \hat{K} \hat{q}
$$

We neglect the final term in Eq. (3.10) and recombine the projection operators in the remaining terms to obtain a modified exact exchange operator:

$$
\hat{K}' = \hat{\rho} \hat{K} + \hat{K} \hat{\rho} - \hat{\rho} \hat{K} \hat{\rho}
$$

Transforming this expression into the AO basis yields our final working expression for $\hat{K}'$:

$$
K'_{\mu\nu} = \sum_{\lambda i} S_{\mu\lambda} c_{\lambda i} K_{i\nu} + \sum_{i\lambda} K_{\mu i} c_{\lambda i}^* S_{\lambda \nu} - \sum_{ij \lambda \sigma} S_{\mu \lambda} c_{\lambda i} K_{ij} c_{\sigma j}^* S_{\sigma \nu}
$$

The second term is clearly just the transpose of the first, and the final term may be obtained easily from the common intermediate, $K_{i\nu}$. The basic idea of computing just $K_{i\nu}$,
or, for that matter, just $F_{\nu\nu}$ is of course extremely simple and undoubtedly has been thought of before. It has been employed in the iterative updates of plane-wave DFT calculations[128], where the full Fock matrix is enormous and thus evaluating a narrow rectangular matrix is far preferable to an enormous square matrix. Indeed, the idea of constructing this economized version of $K$ was mentioned briefly by Aquilante et al.[129], but we have found no prior or subsequent discussion in the literature. The important point that has not yet been accomplished is to demonstrate how formation of an economized $K$ matrix can be used for computational advantage with AO basis sets.

The RI Approximation

One approach to exploiting the gains in efficiency made possible by the economization of the $K$ matrix, is to introduce the RI approximation. This approximation is applied by expanding a given pair of gaussians in terms of an atom-centered gaussian auxiliary basis set, as shown in Eq. (3.3). The expansion coefficients are typically determined by minimizing the self-repulsion of the residual between the exact gaussian product and its fitted expansion[38]. This formulation yields the following standard formulation of the coefficients in global Coulomb-metric RI[38]:

$$C_{\mu\lambda}^\nu_{Q} = \sum_{P} (\mu\lambda|P) (P|Q)^{-1}$$

(3.13)

Global Coulomb RI can be shown to be equivalent to the Dunlap functional for the two-electron integral, and thus yields error in the final integral quadratic in the expansion errors of the individual products[88].

The occ-RI-K Algorithm

It is apparent from the above discussion that the problem of the efficient computation of the $K$ matrix now reduces to the problem of computing $K_{\nu\nu}$. Application of the RI approximation and expressing the density matrix in the MO basis yields the following expression for the $K_{\nu\nu}$:

$$K_{\nu\nu} = \sum_{\mu j \lambda \sigma Q} C_{\mu\lambda Q}^{\nu\lambda} (Q|\nu\sigma) c_{\mu i} c_{\lambda j} c_{\sigma j}$$

(3.14)

In the following discussion of performance, we use the following notation for dimension quantities: $N$ (number of primary basis functions), $[NB2]$ (number of significant primary-basis function pairs, which is asymptotically linear in system size), $X$ (number of auxiliary basis functions), and $o$ (number of occupied orbitals). In order to avoid an expensive $O(X^2)$ step, we pretransform the three-center integrals prior to the formation of the Coulomb metric RI fit coefficients. For calculations run with realistic basis sets and thresholds, $[NB2] >> o^2$, and thus pretransforming the integrals renders the $O(X^2)$ step relatively unimportant. The
CHAPTER 3. THE OCC-RI-K ALGORITHM

<table>
<thead>
<tr>
<th>Step</th>
<th>Operations</th>
<th>Scaling</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>At the start of first SCF iteration: Calculate $(P</td>
<td>Q)^{-1}$</td>
<td>$X^3$</td>
<td>$M^3$</td>
</tr>
<tr>
<td>For each SCF iteration:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loop over atomic batches of auxiliary functions $P$:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate $(\mu\lambda</td>
<td>P)$ in batch</td>
<td>$[\text{NB2}]X$</td>
<td>$M^2$</td>
</tr>
<tr>
<td>$(\mu j</td>
<td>P) = \sum_{\lambda} (\mu\lambda</td>
<td>P) c_{\lambda}^j$</td>
<td>$o[\text{NB2}]X$</td>
</tr>
<tr>
<td>Write $(\mu j</td>
<td>P)$ to disk.</td>
<td>$oNX$</td>
<td>$M^3$</td>
</tr>
<tr>
<td>$(ij</td>
<td>P) = \sum_{\mu} (\mu j</td>
<td>P) c_{\mu}^i$</td>
<td>$o^2NX$</td>
</tr>
<tr>
<td>$C_{ij}^{\mu} + = \sum_{\mu} (ij</td>
<td>P)(P</td>
<td>Q)^{-1}$</td>
<td>$o^2X^2$</td>
</tr>
<tr>
<td>Loop over atomic batches of auxiliary functions $Q$:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read $(\nu j</td>
<td>Q)$ batch from disk.</td>
<td>$oNX$</td>
<td>$M^3$</td>
</tr>
<tr>
<td>$K_{\mu\nu} + = \sum_{Q} C_{ij}^{\mu}(\nu j</td>
<td>Q)$</td>
<td>$o^2NX$</td>
<td>$M^4$</td>
</tr>
</tbody>
</table>

Table 3.1: The occ-RI-K algorithm, omitting the trivial final matrix multiplication steps. The second column gives the operation cost for each step in terms of $N$ (number of primary basis functions), $[\text{NB2}]$ (number of significant primary-basis function pairs, which is asymptotically linear in system size), $X$ (number of auxiliary basis functions), and $o$ (number of occupied orbitals). The computational cost scaling is given as a power of the size of the molecule, $M$, assuming a uniform choice of the AO and auxiliary basis sets. Memory demand is indicated in the same way.

resulting algorithm for the formation of $K_{\mu\nu}$ is shown in Table. 3.1. With $K_{\mu\nu}$ in hand, the three terms in Eq. (4.7) can be formed trivially via fast matrix multiplications.

If the $oNX$ storage requirement becomes prohibitive, the algorithm can be trivially modified to add an additional level of batching over the occupied orbital index $j$. This will then necessitate recomputation of the AO three-center integrals once per batch of $j$ values. We have constructed such an implementation, but do not find it necessary to use for the timing experiments presented here. In practice, the disk I/O wall time cost is a negligible fraction of the algorithmic runtime.

We can anticipate the performance gains that can be achieved using occ-RI-K by comparing the three quartic steps in Table 3.1 against the two quartic steps of the standard RI-K method. In standard RI-K[39], one forms $B_{\mu\nu}^Q = (\mu j|P)(P|Q)^{-1/2}$ which requires $oNX^2$ operations, followed by $K_{\mu\nu} + = B_{\mu j}^Q B_{\nu j}^Q$ which requires $oN^2X$ operations. The speedup given by occ-RI-K is therefore approximately:

$$s = \frac{N(X+N)}{o(X+2N)}$$ (3.15)
If we (conservatively) assume that \( X \approx 2N \), then the approximate speedup is \( s = \frac{3}{4} (N/o) \), suggesting that occ-RI-K can yield a speedup over RI-K which is 75% of the compression factor associated with calculating \( K_i^\nu \) rather than \( K_{\mu \nu} \). For a basis set of cc-pVQZ size, this suggests an asymptotic speedup of roughly a factor of 15 (presuming the cubic steps are negligible, as in the very large molecule limit). The next step is to obtain real-world timings on the basis of an implementation for finite-size molecules, and compare against other methods for forming the \( K \) matrix.

### 3.3 Results and Discussion

All calculations were performed with a development version of Q-Chem 4.2\cite{130}

#### Accuracy

At SCF convergence, the energy errors associated with occ-RI-K are formally equivalent to those of RI-K. However, as RI-K has historically received only minimal accuracy benchmarking, primarily in the context of training auxiliary basis sets\cite{72}, we wish to provide an expanded survey of the accuracy of this approximation. We also wish to provide an in-depth comparison to our recent implementation\cite{121} of the (also sparsely benchmarked)\cite{79, 80} PARI-K algorithm. We now demonstrate that occ-RI-K yields superior accuracy to PARI-K, causing negligible errors with respect to unapproximated integral-direct algorithms for a wide variety of energetic properties. Our test suite is based on that of Mardirossian et al.\cite{116}, and consists of a compilation of many separate sets of energetic reference data spanning thermochemistry and intermolecular interactions. Calculations performed at the B3LYP/def2-QZVPPD\cite{12, 13, 15, 131–133} level of theory yield the results summarized in Table 3.2. While PARI-K yields impressive accuracy in its own right, the accuracy of occ-RI-K is clearly superior. The errors per electron in the absolute energies of all molecules in the test set are quite acceptable, with a mean of 0.7 cal mol\(^{-1}\) electron\(^{-1}\) and a maximum of 3.2 cal mol\(^{-1}\) electron\(^{-1}\). This is significantly lower than the corresponding mean of 2.0 cal mol\(^{-1}\) electron\(^{-1}\) and max of 11.8 cal mol\(^{-1}\) electron\(^{-1}\) for PARI-K. Note that in these tests the Coulomb interactions are evaluated by exact methods.

#### Convergence

At no point in the course of SCF iteration does the occ-RI-K method explicitly construct the exact Fock matrix; rather, we construct a modified Fock matrix that yields the identical orbital gradient and DIIS error vectors, and thus converges to the same result as a calculation using the conventional RI-K method. However, the results of the diagonalization updates are not identical, as these depend upon the unoccupied eigenvalues which are not obtained exactly via occ-RI-K. So while the final result (presuming convergence) is not in doubt, the convergence properties of occ-RI-K are an open question. We empirically demonstrate
<table>
<thead>
<tr>
<th>Dataset</th>
<th>occ-RI-K Error</th>
<th>PARI-K Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A21x12[134]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>A24[135]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AE18[136]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>AlkAtom19[137]</td>
<td>0.01</td>
<td>0.14</td>
</tr>
<tr>
<td>AlkIsomer11[137]</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>AlkIsod14[137]</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>Bauza30[138, 139]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Butanedio65[140]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>BzDC215[141]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CT20[142]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DIE60[143]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DS14[144]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>EA13[145]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>EIE22[146]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>FmH2O10[147, 148]</td>
<td>0.01</td>
<td>0.16</td>
</tr>
<tr>
<td>ACONF[149, 150]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>BHPERI[150, 151]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CYCONF[150, 152]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>G21EA[150, 153]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>G21IP[150, 153]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>NBPRC[150]</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>WATER27[150, 154]</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>NHTBH38[150, 155]</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>HTB38[150, 156]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>BH76RC[150]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DBH24[151]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>H2OBind8[148, 157]</td>
<td>0.00</td>
<td>0.05</td>
</tr>
<tr>
<td>HBS15[158]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>HSG[159, 160]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>HW30[161]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>HW6Cl[147, 148]</td>
<td>0.00</td>
<td>0.04</td>
</tr>
<tr>
<td>HW6F[147, 148]</td>
<td>0.00</td>
<td>0.04</td>
</tr>
<tr>
<td>IP13[145]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>NBC10[160]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>NC15[162]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Pentane14[163]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>RG10[164]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>S22[160, 165]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>S66[96, 166]</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>S66x8[96]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Shields38[167]</td>
<td>0.01</td>
<td>0.06</td>
</tr>
<tr>
<td>Styrene45[168]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>SW49Rel345[169]</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>SW49Rel345[169]</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>SW49Rel6[169]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>SW49Rel6[169]</td>
<td>0.00</td>
<td>0.04</td>
</tr>
<tr>
<td>TA13[170]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>BDE99MR[171]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>BDE99nonMR[171]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>HAT707nonMR[171]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>HAT707MR[171]</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>ISOMERIZATION20[171]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SN13[171]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TAE140nonMR[171]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>TAE140MR[171]</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>X40[172]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>XB31S[173]</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>XB51[173]</td>
<td>0.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 3.2: occ-RI-K vs. PARI-K Accuracy comparison over a compilation of test sets, using the B3LYP/def2-QZVPPD level of the theory. Data in kcal/mol.
CHAPTER 3. THE OCC-RI-K ALGORITHM

that convergence using occ-RI-K is essentially identical to that of standard DIIS. Over the entire test set, the average number of iterations to convergence with and without occ-RI-K is identically 9.3 iterations. A histogram of the change in SCF iteration count for all calculations in the test set is shown in Fig. (3.1) Convergence for the vast majority of jobs is completely unaffected, while 752 jobs converge faster with occ-RI-K compared to 702 that converge slower. No job requires more than six additional iterations to converge. We can thus state definitively that occ-RI-K convergence is well-behaved, and on the basis of these statistics, we can simply assess performance based on timing a single K-build.

Timings

We now demonstrate that occ-RI-K outperforms our recently developed PARI-K algorithm and traditional RI-K. The timings for conventional integral evaluation are shown to reiterate the dramatic advantage of these methods for large basis sets. The systems chosen for comparison are n-by-6 hydrogen-terminated graphene sheets of successively higher depth; we prefer these “somewhat two-dimensional” benchmarks as their size can be increased in a linear fashion but they yield arguably more realistic AO locality than linear systems. The basis set used was the Dunning cc-pVQZ basis set.[174] The results of this timing experiment are shown in Fig. (3.2).

For the largest system treated with the conventional integral code and RI-K, occ-RI-K provides a 14× speedup, compared to 7× for PARI and 4× for RI. We note that the difference between this PARI-K speedup and that reported previously[121] for the same system/basis set is likely due to a looser screening threshold ($10^{-10}$ here vs $10^{-12}$ in previous work), which accelerates the conventional code relative to PARI and occ-RI-K. The use of a different architecture (AMD Opteron 6376 vs. AMD Opteron 2376 previously) could also play a role.
Figure 3.2: Timings for the second $K$ build of various depths of $n$-by-6 graphene sheets in the cc-pVQZ basis set using a single thread. An integral screening threshold of $10^{-10}$ was used for all calculations.

For the largest system, occ-RI-K outperforms PARI-K by a factor of two. For the largest system treated with RI-K, occ-RI-K yields a speedup of 3.3x relative to RI-K. This factor is considerably smaller than our predicted speedup relative to RI-K of about 15 in cc-pVQZ. This reflects the fact that quartic steps do not entirely dominate the calculation for molecules of this size, contrary to what we assumed in the speedup analysis. As a result the speedups over RI-K will increase further with yet larger size molecules. We demonstrate this trend by considering two linear alkanes, $C_{30}H_{62}$ and $C_{70}H_{142}$, in the basis sets cc-pVQZ and cc-pVTZ, respectively. For the smaller alkane in the larger basis set, the computational cost is dominated by the lower scaling steps, as shown in Table 3.3. When the system size is increased and the basis set size is reduced, the higher-scaling MO basis steps become much more significant, illustrating the trend of larger speedups with respect to RI-K with increasing system size.

We shall now demonstrate that occ-RI-K can be applied to quite large systems even in small basis sets, while still outperforming conventional SCF. While occ-RI-K has unfavorable asymptotic scaling compared to conventional SCF, in practice this shortcoming is drastically mitigated by two factors. First, the much higher floating point efficiency of standardized linear algebra libraries relative to AO integral evaluation codes coupled with the replacement of extended primary basis function indices with more compact occupied orbital indices yields a much lower prefactor for RI methods that is not overcome by lower-scaling methods until very long length scales. Second, the favorable scaling of conventional $K$ construction is very difficult to attain in practice, as both more diffuse basis sets and tighter integral thresholds than those used in timing benchmark studies must often be used in practical applications.

As an illustration of these challenges, we compare the performance of occ-RI-K to that of LinK[33]. An integral threshold of $10^{-10}$ is often recommended[25, 98] to converge SCF energies to $10^{-6}$. The systems on which these comparisons are performed are the octahedral diamondoid model systems used by Strout and Scuseria to model SCF scaling in a previous publication,[25] treated using the cc-pVDZ basis set[174]. Due to the absence of a standard exchange auxiliary fitting basis for this basis set, the double zeta RI-MP2 auxiliary basis set of Weigend was used instead[70]. The results are shown in Fig. (3.3).
Table 3.3: Contribution of various steps to overall occ-RI-K walltime for two systems and basis sets. Disk reads/writes are included in overall times for various steps, though their contribution is minor. An integral threshold of $10^{-10}$ was employed in both calculations. The total number of basis functions were 3510 and 4088, respectively.

Figure 3.3: Timings for the second $K$ build of various size octahedral diamondoids in the cc-pVDZ basis set using a single thread. The points correspond to $n = 2, 3, 4, 5$ in the compound series $C_{(4n^3-n)/3}H_{4n^2}$ used by Strout and Scuseria[25]. An integral screening threshold of $10^{-10}$ was used for all calculations.
While occ-RI-K is eventually surpassed by LinK, the first system in which LinK performs better is $C_{165}H_{100}$. Thus, even for this very small, compact basis set, a significant size regime exists where occ-RI-K is competitive. We thus assert that occ-RI-K is likely to accelerate a significant subset of everyday quantum chemistry calculations.

### 3.4 Conclusions

The occ-RI-K method is a new scheme for building the $\mathbf{K}$ matrix partially in the MO basis that, combined with the RI approximation, leads to an extremely efficient algorithm that typically outperforms current alternatives. The occ-RI-K method yields extremely accurate energetics (identical to conventional RI-K) and well-behaved convergence (essentially unchanged relative to conventional methods). It yields speedups over conventional integrals and conventional RI-K that increase with the size of the AO basis set. Occ-RI-K thus helps to make larger basis set hybrid DFT calculations more feasible, which is desirable for improved accuracy with modern functionals. Whilst we have not treated the case explicitly here as it is a very simple generalization, we note that range-separated hybrid functionals evaluated using occ-RI-K yield an essentially equal speedup over RI-K. Finally, there are very intriguing prospects for employing the economized $\mathbf{K}$ framework in lower scaling (conditionally linear scaling) algorithms, using localized orbitals, which will be valuable for the largest molecules.

### 3.5 Acknowledgements

Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, and Basic Energy Sciences. We also acknowledge partial support of this work from Q-Chem Inc through NIH SBIR grant no. GM096678. MHG is a part-owner of Q-Chem Inc.
Chapter 4

Efficient Exact Exchange for Intermolecular Interactions

4.1 Introduction

Weakly-interacting molecular clusters form a very large class of chemically interesting systems; in particular, the explicit treatment of solvation requires very large scale calculations on clusters of solvent molecules. Electronic structure calculations on systems of such scale inevitably require linear-scaling algorithms in order to be practical. While many general linear-scaling electron structure methods are available\[28, 33, 82, 175\], significant savings can be gained by exploiting the specific weakly-interacting nature of these systems. One classic approach to treating such systems is to associate each molecular orbital with a local subset of the atomic orbital basis; for example, in a large cluster of water molecules the molecular orbitals for a given molecule might be expanded using an atomic orbital domain restricted to that molecule. This approach has been described under many names in the literature over its long history\[176–182\]; we shall use the nomenclature coined by Gianinetti et al. and refer to the method as SCF for Molecular Interactions (SCF-MI)\[179\]. The orbitals in SCF-MI are non-orthogonal, and we shall refer to them using the nomenclature of Khaliullin and Head-Gordon as Absolutely Localized Molecular Orbitals (ALMOs)\[182\].

A highly desirable property of SCF-MI is that despite its chemically intuitive locality, it still yields a valid antisymmetric variational trial wave function, and may be applied in conjunction with any density functional of the users choice. The SCF-MI method is also free of basis set superposition error (BSSE)\[183\] by construction.

While initially conceived as a means to improve SCF performance\[176\], the SCF-MI method has found extensive use in the field of Energy Decomposition Analysis (EDA) methods\[184–188\], which seek to break down the full intermolecular interaction energy into physically intuitive components. One of the earliest methods in this class that remains in wide use to this day is symmetry-adapted perturbation theory (SAPT), in which the parts of the Hamiltonian corresponding to intermolecular interactions are treated as perturbations within
the framework of Rayleigh-Schrödinger perturbation theory[189]. The low-order terms in the SAPT energy expression may be interpreted as corresponding to physically intuitive interactions such as frozen electrostatic interactions, polarization, and dispersion effects[190]. Unfortunately, the SAPT method is very computationally expensive; alternate EDA methods with mean-field cost are therefore desirable in order to treat larger systems. An early example of a mean-field EDA method was developed by Kitaura and Morokuma, who developed an EDA in which different energy components are obtained by solving a series of modified Hartree-Fock equations with different Fock matrix components neglected depending on the type of interaction. It is desirable for an EDA to separate the energetic contributions from monomer polarization and charge transfer between monomers. The SCF-MI method is naturally suited to this task, as by construction it allows for molecular polarization but not inter-fragment charge transfer. An EDA using SCF-MI was developed by Mo et al.[184], and a similar scheme was later reinvented by Khaliullin and Head-Gordon[186].

In this work, however, we shall focus on the use of SCF-MI as a faster alternative to traditional SCF for weakly-interacting systems. While SCF-MI alone is not accurate enough for use in situations in which charge transfer is important[181, 182, 191], perturbative corrections are available to resolve this problem[182, 192]. For water clusters, SCF-MI can recover 96-97% of the unconstrained SCF binding energy if large basis sets and appropriate perturbative corrections are used[182]. The cost of such perturbative corrections is comparable to diagonalizing the Fock matrix[182], which contributes a trivial portion of the overall calculation time for most large basis-set calculations that are feasible at present. Thus, the central question remains how SCF-MI can be exploited to reduce the cost of the SCF calculation. Payne recognized very early that the SCF-MI method greatly decreases the cost of the Fock matrix diagonalization for large systems[193]; unfortunately, it was pointed out by Stoll et al. that this yields no net benefit because the global Fock matrix must still be formed prior to projection[194]. However, SCF-MI can also accelerate the formation of the global Fock matrix insofar as it can, in some circumstances, facilitate density-based integral screening algorithms. Khaliullin et al. reported moderate speedups for a series of water clusters when the SCF-MI density was used as input for the LinK[33] exchange matrix construction algorithm[182]. The ability of SCF-MI to significantly accelerate large non-hybrid DFT calculations was recently demonstrated by Khaliullin et al.[195], and in this work we shall demonstrate that it can also accelerate more accurate DFT calculations using hybrid functionals, even for large basis sets and tight integral thresholds. While many properties of weakly-interacting systems can often be treated with non-hybrid density functionals at an acceptable level of accuracy[116], hybrid functionals and large basis sets are required for high-accuracy treatment of many types of properties[20, 21], particularly of kinetics data[18]. A high-performance treatment using these functionals is therefore desirable.

Applying SCF-MI in conjunction with hybrid density functionals and large basis sets introduces additional difficulty as the construction of the exact exchange matrix becomes the bottleneck of the calculation. An SCF-MI implementation may of course utilize all of the same integral-screening-based linear scaling algorithms used in traditional SCF[28, 33], and may gain varying degrees of performance improvement from the potential for increased
sparsity in the density matrix. However, we shall demonstrate that traditional approaches to linear scaling exchange matrix construction discard much of the advantageous sparsity present in the ALMO coefficient matrix. In order to more properly utilize this sparsity to make larger performance gains, we shall present a new exchange-matrix construction algorithm combining two recently discussed approximations: the pair resolution of the identity approximation for exchange (PARI-K)[121], and the economization of the K matrix through construction in a mixed occupied MO/AO basis (occ-RI-K)[40]. We combine these two approximations, together with extensive exploitation of the sparsity introduced by the ALMO coefficient matrix, to create a new fast algorithm that we shall call ALMO-PARI-K. We demonstrate that our new method yields very large speedups compared to traditional integral-driven SCF-MI, and is capable of attaining linear scaling. We also show that our method adds negligible additional error to the SCF-MI method.

4.2 Theory

The SCF-MI Method

In the SCF-MI formalism, the chemical system of interest is divided into one or more non-bonded fragments. The occupied orbitals, $|\psi_i\rangle$, corresponding to electrons on a given fragment are expanded only using the atomic orbitals, $|\phi_\mu\rangle$, on that fragment:

$$|\psi_i\rangle = \sum_{\mu \in F_i} T^{\mu \bullet}_{i} |\phi_\mu\rangle$$

These orbitals are referred to as Absolutely Localized Molecular Orbitals (ALMOs). Due to this AO expansion restriction, ALMOs on one fragment are not orthogonal to the ALMOs from a different fragment. We therefore use the standard subscript/superscript tensor notation to denote biorthogonal covariant and contravariant quantities, using the • symbol to add clarity to index ordering when both subscripted and superscripted indices are present[196]:
CHAPTER 4. EFFICIENT EXACT EXCHANGE FOR INTERMOLECULAR INTERACTIONS

\[ S_{\mu\nu} \equiv (S^{-1})_{\mu\nu} \]
\[ \langle \phi^\mu | = \sum_\nu S_{\mu\nu} \langle \phi_\nu | \]
\[ \langle \phi^\mu | \phi_\nu \rangle = \delta_{\mu\nu} \]

\[ \sigma_{ij} \equiv \langle \psi_i | \psi_j \rangle \]
\[ \sigma^{ij} \equiv (\sigma^{-1})_{ij} \]
\[ \langle \psi^i | = \sum_j \sigma^{ij} \langle \psi_j | \]
\[ \langle \psi^i | \psi_j \rangle = \delta_{ij} \]

where we have made the customary definition of \( \sigma \) as the MO-basis overlap matrix. Without loss of generality, we require the ALMOs on a single fragment to be orthogonal to one another. At each new SCF-MI iteration, new ALMOs for each fragment are obtained through diagonalization of the corresponding fragment-projected Fock operator. In this work, we use Stoll’s fragment Fock operator[177]:

\[ \hat{f}^x = (\hat{1} - \hat{\rho} + \hat{s}^x) \hat{f} (\hat{1} - \hat{\rho} + \hat{s}^x) \quad (4.2) \]

where the fragment occupied projection operator \( \hat{s}^x \) is simply[182]:

\[ \hat{s}^x = \sum_{i \in F_x} |\psi^i \rangle \langle \psi^i | \quad (4.3) \]

SCF-MI therefore removes the need for diagonalization of the global Fock matrix, replacing it with a set of fragment-dimension Fock matrix diagonalizations. This removes the potential \( O(N^3) \) diagonalization bottleneck for very large systems. However, there remains the problem of the construction of the global Fock operator appearing in Eq. 4.2. We shall now discuss the efficient evaluation of this quantity for methods involving exact exchange.

The Inadequacy of Traditional Integral Screening

The combination of density-matrix and integral screening provides an asymptotically linear-scaling algorithm for computation of exact exchange. However, the performance of integral-screening based algorithms is severely reduced in large basis sets, which are necessary to attain the full accuracy potential of modern density functionals. There are two reasons behind this unfortunate fact. First, Gaussian overlap sparsity is greatly reduced in larger, more diffuse basis sets. Second, the density matrix does not exhibit appreciable sparsity in
these basis sets. The reason that large, diffuse basis sets destroy density matrix sparsity is contained in the work of Baer and Head-Gordon[35], who showed that the number of non-zero elements in a given column of the AO-basis density matrix is proportional to the condition number of the overlap matrix. In a large, somewhat linearly-dependent basis set, this condition number is sufficiently large to make the density matrix practically dense except for very extended systems. This fact is not merely true for SCF-MI but for regular SCF as well. We provide numerical illustration of this fact in Fig. 4.1, which displays density matrix sparsity as a function of alkane chain length and threshold. Even for this small basis set (6-31G*), the onset of density matrix sparsity is very slow unless a quite aggressive threshold is used.

When one considers the ALMO density matrix, an alternate interpretation of the result of Baer and Head-Gordon for this particular case becomes apparent:

\[ P = T \sigma^{-1} T^\dagger \]  

(4.4)

In the SCF-MI approximation, the MO coefficient matrix is block-diagonal, providing highly advantageous sparsity. However, this sparsity is difficult to exploit in traditional AO-based linear-scaling codes because the inverse metric appearing in the density matrix expression destroys much of the sparsity. While Khaliullin et al. observed moderate speedups for certain systems when SCF-MI was employed in conjunction with LinK[182], new algorithms are needed to take full advantage of the sparsity inherent in this paradigm. Our subsequent performance benchmarks will demonstrate that very significant performance gains may be
attained if the ALMO coefficient matrix sparsity is properly utilized, rather than being sacrificed through formation of the density matrix.

The occ-RI-K and PARI-K Algorithms

The use of density fitting has for some time been a standard technique to accelerate exchange matrix calculations in large basis sets[39]. We shall now apply techniques from this area to the unique case of the SCF-MI exchange matrix. As our goal is the development of a linear-scaling algorithm, we must necessarily employ some type of local fitting domain approximation. We elect to use the PARI approximation, which has been shown to be quite accurate for exchange[79, 80, 121]. In the PARI approximation, our fit coefficients are expanded in terms of auxiliary functions residing on the two atoms on which the functions in the relevant AO pair reside:

\[
C_{\mu\lambda Q} = \sum_{A \cup B} (\mu \lambda | P) (P|Q)_{A \cup B}^{-1}
\]  

(4.5)

In order to provide satisfactory accuracy, it is usually necessary to correct the intrinsic large errors of the PARI approximation by combining it with the Dunlap robust fit functional, as originally suggested by Merlot et al.[79] The Dunlap functional[88] elegantly produces an approximate integral from any RI expansion method with error that is quadratic in the error of the original method:

\[
(\mu \lambda | \nu \sigma) \approx (\tilde{\mu} \tilde{\lambda} | \nu \sigma) + (\mu \lambda | \tilde{\nu} \tilde{\sigma}) - (\tilde{\mu} \tilde{\lambda} | \tilde{\nu} \tilde{\sigma})
\]  

(4.6)

It should be noted, however, that recent work suggests that a very large, high-angular momentum auxiliary basis set may obviate the need for the Dunlap functional in the PARI approximation, providing the potential for much larger speedups, depending on the relative sizes of auxiliary basis sets needed[197].

In addition to the PARI approximation, we incorporate the central “economization” technique of our occ-RI-K algorithm, which allows us to significantly reduce the prefactor of our calculation through early transformation of an additional index into the occupied subspace of the MO basis. This technique is enabled by the fact that the exchange energy and its gradient (and thus the DIIS error vector[127]) depend only upon \( K_{i\nu} \), rather than \( K_{\mu\nu} \). Specifically, we have shown that the following modified exchange matrix expression may be used to compute these quantities with no loss of accuracy:

\[
K'_{\mu\nu} = \sum_{\lambda i} S_{\mu\lambda} c_{\lambda i} K_{i\nu} + \sum_{i\lambda} K_{\mu i} c^*_{\lambda i} S_{\lambda\nu} \\
- \sum_{ij} \sum_{\lambda\sigma} S_{\mu\lambda} c_{\lambda i} K_{ij} c^*_{\sigma j} S_{\sigma\nu}
\]  

(4.7)
CHAPTER 4. EFFICIENT EXACT EXCHANGE FOR INTERMOLECULAR INTERACTIONS

Once $K_{i\nu}$ is formed, evaluation of $K'_{i\mu}$ is trivial. When the Dunlap PARI-K formulation is applied to the formation of $K_{i\nu}$, we obtain the following expressions:

$$K_{i\nu} = \sum_j (i^j | \nu_j)$$

$$= \sum_j \left[ (i^j | \nu_j) + (i^j | \nu_j') - (i^j | \nu_j) \right]$$

$$= \sum_{jQ} \left[ (D'')_{i\nu}^j H_{\nu jQ} + (H'')_{i\nu}^j D_{\nu jQ} - \sum_P (D'')_{i\nu}^j (P|Q) D_{\nu jQ} \right]$$

where the transformed fit coefficient and integral tensors are given by the following formulas:

$$D_{\mu kQ} = \sum_{\lambda} C_{\mu\lambda Q} T_{\lambda k}^\lambda$$ (4.8)

$$(D')_{ikQ} = \sum_{\mu} D_{\mu kQ} T_{i\mu}^\mu$$ (4.9)

$$(D'')_{i\nu}^j = \sum_k (D')_{ikQ}^j \sigma^{kj}$$ (4.10)

$$H_{\mu kQ} = \sum_{\lambda} (\mu\lambda|Q) T_{\lambda k}^\lambda$$ (4.11)

$$(H')_{ikQ} = \sum_{\mu} H_{\mu kQ} T_{i\mu}^\mu$$ (4.12)

$$(H'')_{i\nu}^j = \sum_k (H')_{ikQ}^j \sigma^{kj}$$ (4.13)

Linear-Scaling RI Implementation of SCF-MI

We shall now discuss the several components of our approach that are necessary to attain linear scaling at each step in our algorithm. Many of the steps in our algorithm exhibit trivial linear scaling by virtue of the appearance of both ALMO and PARI locality in the expressions. However, the formation of the three-center AO integrals $(\mu\lambda|Q)$ appearing in the first two Dunlap terms would exhibit quadratic scaling without the application of additional screening. To approach this issue, we emulate traditional AO-based linear scaling approaches to the formation of the $K$ matrix, which rely principally upon AO integral screening. The LinK algorithm of Ochsenfeld and Head-Gordon[33] is representative of such an approach. The LinK algorithm attains linear scaling by only considering integrals with density-weighted Schwarz estimates with values greater than a specified significance threshold:

$$(\mu\lambda|\mu\lambda) (\nu\sigma|\nu\sigma) |P_{\lambda\sigma}| \geq \text{thresh}$$
In practice, one wishes to use the largest-magnitude ("loosest") possible threshold that satisfies accuracy requirements and does not inhibit SCF convergence. A rigorous understanding of the integral thresholds required to ensure well-behaved SCF convergence across different applications is at present still being developed. Several authors have suggested that an integral threshold of approximately $10^{-10}$ is appropriate to ensure microhartree convergence\[25, 98]. We use this value except in cases in which a tighter threshold is necessary to ensure that the AO overlap matrix does not manifest spurious negative eigenvalues. We wish to construct an algorithm that obeys integral threshold requirements analogous to those employed in such “traditional” methods. This will help to ensure that our performance comparisons are fair, and provide reasonable confidence that convergence will be unaffected by the application of the threshold. We obtain expressions compatible with this aim by transforming our fit coefficient tensors into the AO basis:

$$D_{\nu\sigma Q} = \sum_{k_j} D_{\mu k Q} T_{\bullet j}^\sigma \sigma^{kj}$$  \hspace{1cm} (4.14)

$$D''_{i\sigma Q} = \sum_{k_j} (D'')_{i\bullet j}^\sigma T_{\bullet j}^\sigma$$  \hspace{1cm} (4.15)

$$D''_{i\sigma Q} = \sum_{k_j} (D'')_{i\bullet j}^\sigma T_{\bullet j}^\sigma$$  \hspace{1cm} (4.16)

With these new fit coefficients at our disposal, we may rewrite our first two Dunlap terms in a manner amenable to AO integral screening:

$$(L1)_{i\nu} = \sum_{\sigma Q} \mathcal{D}''_{i\sigma Q} (\nu\sigma|Q)$$  \hspace{1cm} (4.17)

$$(L2)_{i\nu} = \sum_{\lambda Q} (i\lambda|Q) \mathcal{D}_{\nu\lambda Q}$$  \hspace{1cm} (4.18)

We thus arrive at two different criteria for the significant integrals needed to accurately evaluate each term:

$$\max_i |\mathcal{D}''_{i\sigma Q}| (\nu\sigma|\nu\sigma) |(Q|Q)^{\frac{1}{2}}| \geq \text{thresh}$$  \hspace{1cm} (4.19)

$$\max_i |T_{\bullet i}^\sigma| \max_{\nu} |\mathcal{D}_{\nu\sigma Q}| (\mu\lambda|\mu\lambda) |(Q|Q)^{\frac{1}{2}}| \geq \text{thresh}$$  \hspace{1cm} (4.20)

We then obtain the combined list of unique integrals from our two respective terms and proceed to evaluate the entire list. Finally, it should be noted that as the third Dunlap term does not require expensive three-center integrals, its evaluation can be accomplished without an analogous integral screening step.

A unique challenge arises in the treatment of contractions of sparse tensors with the inverse metric matrix $\sigma^\nu$. While this matrix does exhibit decay of the off-diagonal elements
with molecular size, its decay is drastically slower than that of the corresponding covariant metric. This makes determining the sparse structure of the output of contractions between the inverse metric and other tensors a non-trivial problem. While one could imagine choosing a threshold for discarding inverse metric elements and using the resulting imposed sparse structure to define the corresponding sparse structure of the output, the slow decay of the matrix elements renders the choice of such a threshold difficult. Threshold values of comparable magnitude to “best practice” AO integral thresholds result in the retention of virtually all elements and the corresponding loss of all output sparsity, while choosing very aggressive cutoff values would necessitate further study to determine whether this practice affects SCF convergence robustness. Given these challenges, we instead opt for a numerically rigorous approach; we bound the magnitude of output blocks using the well-known Frobenius norm identity:

$$\|AB\|_F \leq \|A\|_F \|B\|_F$$ (4.21)

This output block screening procedure was recently used by Lewis et al. in a similar context[198]. The tightness of the predicted bounds is of particularly crucial importance to the formation of the integral screening coefficient tensor for the second term, $D_{\mu\sigma Q}$ in Eq. (4.14), as three AO indices appear in that step. Forming an excessive number of elements of this tensor would result in deteriorating performance for large basis sets. Unfortunately, implementing this screening procedure in a linear-scaling fashion presents its own set of difficulties; we must take care to exit the screening loops early where appropriate in order to maintain linear-scaling. We have therefore developed the algorithm shown in Fig. 4.2 to predict the significant blocks of the output tensor corresponding to the contraction of a linear-size sparse tensor with the dense $\sigma^{ij}$ matrix.

In practice, our screening of inverse metric contractions is sufficiently tight that all of steps in which it appears together constitute a minor portion of the runtime. Our ALMO-PARI-K algorithm, which exploits the combination of PARI locality in the RI fit coefficients and ALMO locality in the MO coefficient matrix to obtain linear scaling, is presented in Fig. 4.3.

4.3 Results and Discussion

Convergence: The Effect of Conjugation

We have found that using occ-RI-K in conjunction with PARI-K can lead to convergence difficulties that are not present when either method is used separately. Even when a very good SCF guess (for weakly interacting systems) is used, such as the superposition of monomer coefficient matrices, the initial Roothaan step taken after the first Fock build may land on a density that yields very large PARI errors. In such cases, the exchange energy may even become positive. Such results are possible due to the fact that the exact exchange energy is a lower bound to the PARI exchange energy, and the problem seems to correlate with an
For each row block of $B$, sort column blocks in descending order by their Frobenius norm.

Loop significant blocks of $A$:

If uncontracted indices in $A$ block unchanged from previous iteration:

Compute and store Frobenius norm of $A$ block into [norm list].

Store contracted index value into [contracted block list].

Continue.

Else:

Select maximal $A$ block Frobenius norm from [norm list], $\|A\|_F^{max}$.

Loop over $j' = 1 : [\# \text{ of } B \text{ column blocks}]$

Set $keep\_going$ to False

Loop over $i \in [\text{contracted block list}]$

If $\|A\|_F^{max} \cdot \|B\text{ block}\|_F \geq \text{thresh}$

Set $keep\_going$ to True

Add $j$, $B$ column index value corresponding to $(i, j')$, to [output block list]

If $keep\_going$ is False, break

Loop $j$ in [output block list]:

Append combined $A$ uncontracted indices and $j$ to list of significant $C$ indices

Clear [norm list]

Clear [contracted block list]

Clear [output block list]

Figure 4.2: Linear-scaling algorithm for prediction of significant output blocks of $C$, the output from contraction of a sparse tensor $A$ with a dense matrix $B$, such as $\sigma^{ij}$

insufficient auxiliary basis set. We have found that, for calculations run in the aug-cc-pVTZ basis set[199], a simple solution to this problem is to augment the cc-pVTZ-JK auxiliary basis set[39] with the diffuse functions present in the RI-MP2 aug-cc-pVTZ auxiliary basis set[200]. This causes only a small increase in cost, as the most expensive steps of our algorithm scale only first order in the size of the auxiliary basis set. While augmenting the auxiliary basis set enables convergence for all systems, a significant increase in the required number of iterations is observed for conjugated systems in the S66 set[96]. This is in accord with earlier findings describing more severe PARI fitting errors for conjugated systems[79, 80], though in our case the accuracy of the final energy is unaffected. As will be shown in the subsequent section on the performance of the algorithm, the performance gains are so large that even for the worst convergence cases a net speedup is expected. The reason why the use of occ-RI-K causes the SCF to explore these pathological PARI densities that are not typically encountered by PARI-K alone is presently unclear, and presents an interesting
\[
C_{\mu\lambda Q} = \sum_P (\mu\lambda|P)(P|Q)^{-1}_{A\cup B}
\]
\[
D_{\mu k Q} = \sum C_{\mu\lambda Q} T^{\lambda^*}_{k}
\]
\[
(D')_{ik Q} = \sum_{\lambda} D_{\mu k Q} T^{\mu^*}_{i}
\]
\[
(D'')^{*j}_{i Q} = \sum_{k} (D'')^{*j}_{ik Q} \sigma^{kj}
\]

Loop over atomic batches of auxiliary functions Q:
\[
\overline{D}_{\nu \sigma Q} = \sum_{kj} D_{\mu k Q} \sigma^{kj} T^{\mu^*}_{j}
\]
\[
\overline{D''}_{i \sigma Q} = \sum_{kj} (D'')^{*j}_{i \sigma Q} T^{\sigma^*}_{j}
\]

Compute \((\mu\lambda|Q)\) based on \(\overline{D}_{\nu \sigma Q}, \overline{D''}_{i \sigma Q}\)
\[
H_{\mu k Q} = \sum_{\lambda} (\mu\lambda|Q) T^{\lambda^*}_{k}
\]
\[
(H')_{ik Q} = \sum_{\lambda} H_{\mu k Q} T^{\mu^*}_{i}
\]
\[
(H'')^{*j}_{i Q} = \sum_{k} (H')_{ik Q} \sigma^{kj}
\]
\[
(L1)_{i \nu} = (D'')^{*j}_{i \nu} H_{vj Q}
\]
\[
(L2)_{i \nu} = (H'')^{*j}_{i \nu} D_{vj Q}
\]
\[
E^{*j}_{i \nu} = (D'')^{*j}_{i \nu} (P|Q)
\]
\[
(L3)_{i \nu} = E^{*j}_{i \nu} D_{vj Q}
\]
\[
K_{i \nu} = (L1)_{i \nu} + (L2)_{i \nu} - (L3)_{i \nu}
\]

Figure 4.3: The ALMO-PARI-K algorithm. All steps are asymptotically linear scaling, with the exception of the final increment of \(K\), which is trivial.
CHAPTER 4. EFFICIENT EXACT EXCHANGE FOR INTERMOLECULAR INTERACTIONS

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>MSE</th>
<th>MAXE</th>
<th>Δ iterations</th>
<th>Δ iterations %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjugated</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>5.1</td>
<td>78</td>
</tr>
<tr>
<td>Non-Conjugated</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.5</td>
<td>7.4</td>
</tr>
<tr>
<td>All</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>2.8</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 4.1: Binding energy errors for the S66 set relative to SCF-MI with no PARI-K/occ-RI-K approximation. The column header acronyms are MAE = Mean Absolute Error, MSE = Mean Signed Error, and MAXE = Maximum Absolute Error. The last two columns denote the average absolute and percentage change in the number of SCF iterations, respectively. All units are in kcal mol\(^{-1}\).

prospect for further investigation.

Accuracy

In order to assess the accuracy of our ALMO-PARI-K method relative to standard integral-driven SCF-MI, we compare the performance of the two approaches for the binding energies of the S66 non-covalent dimer test set \[96\]. We use the density functional \(\omega B97X-V\)[18], which has been shown to perform well on a broad range of chemical problems, in conjunction with aug-cc-pVTZ basis set and the augmented cc-pVTZ-JK RI auxiliary basis set discussed in the previous section. For this range-separated functional, we employ the tactic from our previous PARI-K work \[121\] of combining the evaluation of the short and long-range \(K\) matrices. We do so through the use of a single set of (Coulomb metric) fit coefficients for both terms, which is nonetheless a formally robust fitting procedure due to our use of the Dunlap functional. In a somewhat interesting development, we have found that this technique yields both improved performance and improved numerical stability, as linear dependence in the long-range two-center auxiliary integral matrix can render the inversion step numerically unstable.

The energetic accuracy results that we have obtained are given in Table 4.1. It is readily apparent that the introduction of the PARI-K and occ-RI-K approximations, as well as the screening specific to the ALMO-PARI-K algorithm, introduce negligible additional errors into the binding energies. As SCF-MI is free of BSSE by construction, we have no need to employ counterpoise corrections.

Performance

We begin by assessing the speedup that our method attains relative to traditional integral-driven SCF-MI, as exemplified by the LinK algorithm. As our test systems, we choose a suitable SCF-MI application target, large water clusters, an example of which is shown in Fig. 5.11.

The wall-times for the \(K\) matrix build for three such clusters are summarized in Table 4.2. It is readily apparent that our implementation attains a very significant speedup with
CHAPTER 4. EFFICIENT EXACT EXCHANGE FOR INTERMOLECULAR INTERACTIONS

Figure 4.4: 127-molecule water cluster used in timing benchmarks

<table>
<thead>
<tr>
<th># of Basis Functions</th>
<th>46 H2O</th>
<th>71 H2O</th>
<th>127 H2O</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinK Wall Time (hrs)</td>
<td>26.9</td>
<td>79.5</td>
<td>290.6</td>
</tr>
<tr>
<td>ALMO-PARI-K Wall Time (hrs)</td>
<td>2.4</td>
<td>7.5</td>
<td>37.3</td>
</tr>
<tr>
<td>Speedup</td>
<td>11.4x</td>
<td>10.6x</td>
<td>7.8x</td>
</tr>
</tbody>
</table>

Table 4.2: Wall times in hours for the first K build of an SCF-MI calculation, using a guess of superimposed fragment densities. The aug-cc-pVTZ basis set was used with an integral screening threshold of $10^{-11}$ and the augmented cc-pVTZ-JK auxiliary basis described above.

respect to LinK, due to its use of lower-dimensional AO quantities and rapid contraction into the MO basis. While these water clusters, as non-conjugated systems, could likely be run even faster by omitting the remedial augmentation of the auxiliary basis set, we wish to demonstrate the performance of our method in its “black box” configuration.

We now wish to assess the asymptotic scaling of our method, and do so by considering a series of benzene ring $\pi-\pi$ stacks with increasing numbers of stacked rings. While the nature of this system is admittedly artificial, we have chosen it primarily because it is one of the few systems in which we have observed that the integral-driven LinK algorithm used for comparison is even capable of attaining linear scaling for reasonable system sizes in the aug-cc-pVTZ basis set. It is thus ideal for comparing the scaling between our method and LinK in the large-system limit. An example of such a stacked benzene system is shown in Fig. 4.5.

Our timing results are shown in Fig. 4.6. LinK attains perfect linear scaling when going from ten benzenes to twenty. Despite the appearance of the graph, our method does not quite attain linear scaling by thirty benzenes due to the necessity of including augmented auxiliary functions to improve convergence, as discussed earlier. From a log-log plot, we obtain an effective scaling of $O(N^{1.35})$. Nonetheless, the absolute speedup with respect to LinK is so large (14x for twenty benzenes) that this difference in effective scaling is unlikely to make LinK a better choice for any realistic system.
CHAPTER 4. EFFICIENT EXACT EXCHANGE FOR INTERMOLECULAR INTERACTIONS

Figure 4.5: One of the stacked benzene systems used as a scaling benchmark in this work.

Figure 4.6: Scaling benchmarks performed on increasing length stacks of benzene rings. The aug-cc-pVTZ basis set was used with the cc-pVTZ-JK RI basis set augmented with the diffuse functions from the rimp2-aug-cc-pVTZ RI basis set. An integral threshold of $10^{-11}$ was used.

4.4 Conclusions

We have applied the occ-RI-K and PARI-K approximations to create a fast, linear-scaling algorithm for the SCF-MI method. Our implementation dramatically accelerates large calculations in realistic basis sets, with very significant performance gains for several large water clusters. We achieve these gains with negligible loss of accuracy, although an increase in the SCF iteration count is observed for conjugated systems. We hope that our method will enable SCF-MI with hybrid density functionals to be applied to much larger systems in conjunction with appropriately accurate basis sets.
Chapter 5

Fully General Sparse Tensor Framework for Electronic Structure Theory

5.1 Introduction

Block-sparse tensors are central mathematical objects in electronic structure theory, the quintessential example of which is the four-center, two-electron integral tensor. The sparsity of the two-electron integral tensor and its blocked structure arise from the application of common integral screening algorithms, the cornerstone of modern SCF calculations\cite{26, 108}. More expensive correlated electronic structure methods transform the two-electron integrals into the molecular orbital basis, requiring additional approximations to preserve sparsity in the computation\cite{201–203}. Such codes can become very cumbersome to develop and maintain; consider, for example, a universally sparse transformation of the three-center atomic orbital basis integrals into a localized occupied\cite{204, 205} and virtual molecular orbital\cite{206, 207} basis, a common step in local correlation methods:

\[
(i\alpha|Q) = \sum_{\mu \in \{\mu\}} \sum_{\nu \in \{\nu\}} (\mu\nu|Q) c_{\mu i} c_{\nu a}
\]  

(5.1)

When a code consists of a long series of such sparse operations, including many different coupled sparse indices with varying types of sparsity, performing such simple refactoring operations as changing the storage order of the indices in several of the contractions becomes an extremely difficult task. While the resolution of the identity (RI) approximation is applied in many contexts in order to reduce the dimensionality of the integral tensors\cite{36, 38}, the developer is still forced to work with tensors containing at least three indices. The increasing sophistication of modern electronic structure methods makes the development of software infrastructure to efficiently handle these objects ever more necessary. While historically most
electronic structure methods have been laboriously implemented by manually composing loops and matrix operations, interest in more suitable software paradigms has been steadily growing.

The early work of Windus and Pople\cite{208} is perhaps the first recognizably modern approach to tensors in electronic structure software. In more recent times, the development of tensor infrastructure has been dominated by the Tensor Contraction Engine (TCE) project\cite{209}. The capabilities of the TCE span the entire process of tensor expression derivation, factorization, and code generation\cite{209, 210}. The TCE has been used to implement extremely complex electronic structure methods, with a particular emphasis on high-order coupled cluster methods\cite{209, 211, 212}. In an alternate line of development, Epifanovsky et al. have developed the libtensor project, a high-level symbolic C++ library to describe block tensor operations\cite{213}. A similar library project with a greater emphasis on concurrency is the Cyclops Tensor Framework of Solomonik et al.\cite{214}

Until recently, the tensor library literature focused almost exclusively on dense tensors. Unfortunately, this neglected the pressing need to software tools to treat sparse tensors, which appear in virtually all reduced-scaling implementations of electron correlation methods, and also in recently developed high-performance mean-field methods\cite{79, 121}. Early efforts to provide rapid software prototyping capabilities for certain classes of sparse problems came from Parkhill and Head-Gordon\cite{215}, and later from Kats and Manby\cite{216}. Recently, there has been a surge of interest in the area of sparse tensor tools. Pinski et al. have formalized the nomenclature and provided an implementation of several commonly used sparse operations\cite{217}. Lewis et al. have described a block-sparse tensor library that also derives cost savings from exploitation of rank sparsity within individual blocks\cite{198}. Finally, Solomonik et al. have extended the Cyclops Tensor Framework to handle a degree of sparsity, and demonstrated this capability in the case of massively parallel contraction of a sparse tensor with a dense tensor\cite{218}.

In this work, we shall describe our block-sparse tensor library, an extension of the libtensor project\cite{213}. We shall provide the first detailed description of an implementation capable of handling fully-general multi-dimensional sparsity in tensor contractions. Our library has the unique ability to represent universally sparse contractions, in which the output and all input tensors have uniquely defined sparse structures. This capability is provided with the added benefit of a high-level, symbolic C++ implementation. We shall present concrete numerical evidence that the ability to handle multi-dimensional sparsity provides significant cost reductions relative to utilizing only pair-wise sparsity, and that the low-overhead of our implementation allows very high floating-point performance to be obtained. In addition, we will discuss the algorithms by which our library resolves an inline symbolic tensor expression into nested loops that perform the minimal number of arithmetic operations in a black-box, fully automated fashion.
5.2 Theory

Block-Sparse Tensor Storage Format

The most straightforward approach to implementing sparse linear algebra routines is the use of element-wise sparsity representations such as the compressed-sparse-row\[219] storage formats. There are many efficient implementations of fundamental linear algebra operations that are built from such data structures\[219–222\]. In many cases, however, performance is greatly improved by using a sparsity representation consisting of small dense blocks rather than individual elements, even if this representation involves storing additional explicit zeros\[223–227\]. Automatic sparse kernel tuning methodologies allow such block-sparse approaches to yield very high floating point performance on a variety of architectures\[225, 228, 229\].

In this work, the dimensions of the dense blocks used in the block-sparse representation of our data structures will be drawn from underlying physical phenomena; tuning such block sizes to obtain optimal performance for specific tensors and architectures presents an interesting avenue for future research. Unfortunately, implementation of block-sparse tensor contractions has historically required time-consuming and error-prone manual coding of each individual contraction. However, several recent high-level library projects have utilized block sparsity in some form\[198, 216\], and our work continues in this vein. In order to obtain high floating point efficiency through the use of existing highly optimized matrix multiply routines, our library stores tensors using the well-established “block-major order.” \[198, 208, 213, 216\] This storage format is defined for a given tensor as follows. First, the subspaces of the tensor are subdivided into blocks; for example, in our implementation of the PARI-K algorithm\[121\], subspaces corresponding to atomic-orbital indices are subdivided by shell (a set of Gaussians with the same center and contraction scheme), those corresponding to auxiliary basis indices are subdivided by atom, and occupied molecular-orbital basis indices are not subdivided at all and referenced as a single block. A given block of the tensor is specified by a tuple of block indices, hereafter referred to as an “index tuple.” Elements of the tensor whose absolute indices fall within the bounds of a given index tuple are stored contiguously in memory. Only non-zero blocks are stored. By convention, the distance between tensor elements corresponding to consecutive block index values of a given subspace decreases with increasing subspace index (this is analogous to “row-major” order for matrices). An example of a tensor stored in this format is shown in Fig. 5.1.

It is somewhat more cumbersome to compute offsets into block-major tensors than typical “row-major” or “column-major” arrays, and this difficulty is compounded when said tensors are sparse. We find it useful to partition the subspaces of a given tensor into “index groups”. An index group is either a single dense index or any set of indices coupled by sparsity to one another. We enclose the indices constituting a sparse index group in parentheses to indicate that they are all members of a single index group, with the exception that we drop this convention for tensors with superscripted indices. For example, the three-center integrals, in which the first two AO indices $\mu$ and $\nu$ are coupled by overlap sparsity, may be denoted
Figure 5.1: An example of the block-major sparse tensor storage format used in our library. The tensors correspond to a fictional system in which only the index tuples (0,1,0),(1,1,1), and (2,0,1) correspond to non-zero blocks. The numbers 5,10, and 15 are example index ranges that are used to define the sizes of the three 3 non-zero blocks illustrated on the right.

by \( I_{(\mu \nu)Q} \), where \( Q \) is the auxiliary basis index. When iterating over tensor blocks within a series of nested index loops, the current index loop may touch a variety of index groups across the different sparse tensors involved in our current operation. For a particular tensor, we calculate the offset of the current block using the formulas given in Eq. 5.2. For an index tuple entry corresponding to a given index group, we define the “[outer size]” to be the product of the block sizes of the outer dimensions, while the “[inner size]” for an index group is the product of the full dimensions of the inner index groups. Thus the [outer size] is a property of an individual index tuple/index group combination, while the inner size is a property of the index group itself. The notion of inner size is essentially equivalent to that of loop “stride”; however, it is important to grasp that incrementing a loop that sets a given index group does not simply move the array references by the appropriate stride, as it also changes the outer size values seen by all further nested loops. Note that for a sparse index group, the dimension that it contributes to the inner size of an index group is the number of elements corresponding to the non-zero index tuples in the group, not the product of the dimensions of the individual subspaces.
[block offset] = \sum_{i \in \text{index groups}} [\text{outer size}]_i [\text{offset}]_i [\text{inner size}]_i \quad (5.2)

[\text{outer size}]_i = \prod_{j < i} [\text{block size}]_j \quad (5.3)

[\text{inner size}]_i = \prod_{j > i} [\text{index dimension}]_j \quad (5.4)

Using Eq. 5.2, we may seamlessly address blocks in both sparse and dense tensors with very heterogeneous block sizes. It is helpful to reconsider the blocks shown in Fig. 5.1 as an example of how Eq. 5.2 is used in practice. Consider the index tuple \((2, 0, 1)\). With the subspace blocking scheme used in Fig. 5.1, the outer size values corresponding to each subspace of the index tuple are 1, 5, and 10, respectively. In this example, all indices constitute a single sparse index group, and as such the inner size of that index group is simply 1. As we shall demonstrate, any apparent peculiarities of this block-major storage format are validated by the high floating point performance that it enables.

**Tensor Sparsity**

Historically, the exploitation of sparsity in electronic structure has focused on pair-wise sparse couplings between indices - for example, the overlap sparsity between two atomic orbital indices \(\mu \lambda\) in the two-electron integrals[25], or the locality of the expansion coefficients \(c(\mu)\) of local molecular orbitals[32]. Pinski et al.[217] and Riplinger et al.[230] have discussed methods for tracking the sparsity relationships between two indices, using a “sparse map” data structure which they describe as two nested lists[230]. While the infrastructure they propose is powerful, its pair-wise nature leads to certain inevitable limitations. For example, in order to formulate a local RI expansion using coefficients \(C_{\alpha\mu\pi\rho}\) of the localized occupied-virtual pair \(i\mu\), Pinski et al. use a pair-wise sparse map between \(i\) and the auxiliary basis index \(Q\). However, it would be more natural in this context to map the \(i\pi\) pair itself to a unique local subset of auxiliary basis functions. We shall discuss why it is necessary in many contexts to utilize a higher-dimensional description, which we shall refer to as “tensor sparsity,” in order to capture the full measure of sparsity in many of the tensors appearing in electronic structure. As a stronger example, we consider the fit coefficients arising in the pair resolution of the identity approximation (PARI)[79], in which a product of Gaussians is expanded in terms of the union of the sets of auxiliary basis functions centered on the respective atoms, \(A\) and \(B\), of the two functions:

\[ |\mu_A\lambda_B\rangle \approx \sum_{Q} A_{\mu\lambda}^{\alpha\pi\rho}(Q) |Q\rangle \quad (5.5) \]
Clearly, the sparse structure of this tensor cannot be described simply by pair-wise maps; the range of the \( Q \) index depends upon both other indices. One might expect that issues of this nature could be handled within the existing pair-wise frameworks by relabeling the pair of indices \((\mu \nu)\) as a single “combined pair index.” However, this does not yield convenient data structures for use in tensor operations in which only one index from the specified index pair couples to the sparsity present in a different tensor. We may readily obtain an example of such a problem by considering the transformation of one index of the PARI fit coefficients into a basis of localized molecular orbitals with the MO coefficients \( c(\lambda j) \):

\[
D(\mu jQ) = \sum_{\lambda} C_{Q}^{\mu \lambda} c(\lambda j) \tag{5.6}
\]

For this and many other problems, we propose the use of a sparse tree data structure, an example of which is shown in Fig. 5.2. In this data structure, each non-zero index tuple corresponds to a specific lowest-level node of the tree. The individual block index values for the different subspaces are nested at each appropriate level; thus the index tuples \((0, 3, 6)\) and \((0, 4, 2)\) nest under the common node corresponding to block index 0 of the first subspace. Each significant value of a given subspace index corresponds to a separate subtree. The “list of lists” utilized by Pinski et al.[217] and Riplinger et al.[230] may be regarded as a two-dimensional special case of this data structure in some sense. However, we have found that in order to obtain high performance it is necessary to minimize the number of independent memory allocations involved in the construction of a sparse tree. As such, while we find that conceptualizing our data structure as a tree and referring to it as such yields a better understanding of its fundamental operations, the actual implementation stores the tree as a single contiguous allocation of block index tuples, as shown in Fig. 5.2.

In a subsequent section, we shall define a set of (nearly) linear-scaling operations on these sparse tree data structures. Using these operational definitions, we shall then prove that no combination of pair-wise sparse maps can recover the full sparsity present in a tensor description; only the tensor description can minimize the number of arithmetic operations in a sparse problem. We shall provide numerical examples of the size of the cost reduction for practical calculations in our Results section.

**Sparse Operations**

We now specify a standardized set of operations that may be performed on these sparse tree data structures. The first operation we wish to discuss is “permutation”, in which we permute the order of the levels in the tree. This may be trivially implemented with \(O(N \log N)\) scaling by swapping the corresponding entries in each index tuple, followed by re-sorting the index tuples into lexicographic order using standard sorting algorithms. The “inversion” operation of Pinski et al., in which the order of the subspace indices in a pair-wise map is reversed, may be regarded as a special case of tree permutation. An example of the permutation operation is shown in Fig. 5.3. This operation is used in many situations,
Figure 5.2: The conceptual representation of our sparse tree data structure and its literal multi-dimensional representation.

Figure 5.3: Permutation of subspaces 1 and 2 (i.e. second and third columns) of a sparse tree

particularly to bring all sparsity information in a tensor contraction into agreement with a common loop ordering.

The second operation that we shall discuss is “fusion”, in which the sparsity of two tensors may be combined to yield a higher-dimensional object. The following example illustrates the utility of this operation. Consider a standard sparse matrix product:

\[ C_{(ij)} = \sum_k A_{(ik)} B_{(kj)} \]  

(5.7)
CHAPTER 5. FULLY GENERAL SPARSE TENSOR FRAMEWORK FOR ELECTRONIC STRUCTURE THEORY

Figure 5.4: Fusion of two lower-dimensional sparse trees which share a common index to form a higher-dimensional tree. In this example, two two-dimensional tensors with a common index type (lowest level of tree A; highest level of tree B) are fused to make a three-dimensional sparse tensor.

All indices in the above equation refer to blocks rather than elements. What is the range of the $k$ index in the inner loop of the contraction in Eq. 5.7? Clearly, it depends upon the particular iteration of the outer loops over the $i$ and $j$ indices. We may obtain the appropriate set of $k$ index values for a given set of outer indices by *fusing* the trees describing $A$ and $B$, meaning that we join the lowest level of tree $A$ to the highest level of tree $B$, as shown in Fig. 5.4.

While this operation bears a resemblance to the “chaining” operation of Pinski et al., in which two pair-wise sparse maps with a common index are used to generate a pair-wise map between their un-shared indices, our operation preserves the full multi-dimensional character of all input sparsity, yielding a higher-dimensional output reflecting the new combined coupling. The space of loop iterations traversed in Eq. 5.7 corresponds to the elements of the resulting fused tree, and the set of $k$ index blocks touched for given values of the outer loop indices is given by the appropriate set of leaf nodes.

Finally, we wish to discuss the “contraction” operation. Contraction involves the removal of a given subspace from a tree, yielding a lower dimensional tree. It is implemented in a near-linear scaling fashion in several stages. First, the appropriate entry is deleted from each index tuple. The index tuples are then resorted, and the duplicate entries which may have been introduced are removed. An example of the contraction operation is given in Fig. 5.5. This operation is extremely helpful for the implementation of electronic structure codes, though we principally reference it in this work in the context of generating pair-wise maps for comparison to higher-dimensional sparsity. We caution that Pinski et al. utilize the term “contract” in the context of sparse data with a very different meaning than that which we give here[217].
CHAPTER 5. FULLY GENERAL SPARSE TENSOR FRAMEWORK FOR ELECTRONIC STRUCTURE THEORY

Figure 5.5: Contraction of the second subspace of a tree to form a lower-dimensional tree.

Emergent Sparsity

We now wish to highlight an interesting special case of the exploitation of sparsity in the context of electronic structure theory, though it likely arises in other contexts also. If a given tensor (Tensor 1) is contracted with another tensor (Tensor 2) sharing two or more of its indices, it is clear that we should utilize the sparse coupling between those shared indices, as present in Tensor 2, not only in the latter contraction but also “retroactively” in the formation of Tensor 1. If we do not, we would be forming elements of Tensor 1 which, while they might be numerically significant, would be multiplied by zero values when they were actually utilized in the subsequent contraction. It is thus often advantageous when possible to impose a sparse structure on the output of a tensor contraction (in this example, Tensor 1) that is somewhat independent of the sparse structure of its input tensors.

We shall now provide a real-world example of this phenomenon. In a separate publication, we discuss the concurrent use of the occ-RI-K[40] and PARI-K[121] approximations in the context of SCF for molecular interactions (SCF-MI)[182]. The SCF-MI method is defined by the construction of a minimum-energy Slater determinant from absolutely localized molecular orbitals (ALMOs)[182]. Each ALMO is expanded only in terms of the atomic orbitals residing on its respective molecular fragment; this results in a block-diagonal MO coefficient matrix, and gives many of the tensors in the method additional block sparse structure. The expression for the exchange matrix in the occ-RI-K approximation using the non-orthogonal ALMOs is given by:

$$ K_{ij} = \sum_j \langle \phi^i | \nu j \rangle $$

where we have used the standard superscript notation to indicate that the first occurrence of the $j$ index is contravariant[196]. Our new method utilizes the Dunlap robust fit functional
to remove the first order fitting errors associated with the PARI-K approximation[88]:

\[(\mu \lambda | \nu \sigma) \approx (\tilde{\mu} \tilde{\lambda} | \nu \sigma) + (\mu \lambda | \tilde{\nu} \tilde{\sigma}) - (\tilde{\mu} \lambda | \tilde{\nu} \sigma)\] (5.9)

In the context of evaluating the exchange matrix, the expression corresponding to the third term in the Dunlap functional is therefore:

\[
(\tilde{\nu}^i_{\bullet} | \tilde{\nu}^j_j) = \sum \limits_P (D''_{\bullet \star})^i_{j \star} (P|Q) D(\nu_{jQ})
\] (5.10)

In this expression, \((D''_{\bullet \star})^i_{j \star}\) corresponds to the fully MO transformed fit coefficients (with one of the MO indices contravariant due to the non-orthogonality of the absolutely localized molecular orbitals)[182], while \(D(\nu_{jQ})\) corresponds to the half-transformed fit coefficients. The combined expression is evaluated in two steps. In the first step, the fully MO transformed RI fit coefficients are contracted with the auxiliary basis Coulomb integrals:

\[
E^i_{\bullet \star} = \sum \limits_P (D''_{\bullet \star})^i_{j \star} (P|Q)
\] (5.11)

Second, the resulting tensor is contracted with the half-transformed fit coefficients and used to decrement the economized exchange matrix:

\[
K_{\nu j} = \sum \limits_{jQ} E^i_{\bullet \star} D(\nu_{jQ})
\] (5.12)

The auxiliary basis Coulomb integrals may be assumed to always be fully dense, given the extremely slow decay of the Coulomb operator. In the absence of other factors, this would in general result in an output \(E^i_{\bullet \star}\) that was quadratic in size. However, the sparse structure of the half-transformed fit coefficients ensures that for every \(j\) value, only a constant number of \(Q\) blocks will be significant. We therefore impose this sparse coupling between \(j\) and \(Q\) on \(E^i_{\bullet \star}\), rendering the first contraction linear-scaling. Due to cases such as this, it desirable that any sparse tensor framework allow the imposition of arbitrary sparse structure on the output of any tensor contraction.

**Tensor Sparsity vs. Pair-Wise Sparsity**

Utilizing multi-dimensional sparsity brings additional code complexity if not handled appropriately. We therefore wish to provide definitive proof that exploiting sparsity in the full multi-dimensional form can reduce the operation count beyond what is possible through consideration of pair-wise sparsity alone, and at worst will recover the performance of the
pair-wise implementation. Once again, we consider as an example the local density fitting expansion for the atomic orbital product $\mu \nu$. The most economical expansion naturally takes the form of some set of auxiliary functions $K$ centered near the $\mu \nu$ product. However, to avoid the additional complexity of dealing with high-dimensional sparsity, many local fitting codes such as the ARI-K algorithm of Sodt and Head-Gordon[62] simply associate the fit domain of $K$ values with a single AO index, either $\mu$ or $\nu$. The most economical pair-wise approximate description of the multi-dimensional sparsity inherent in our RI coefficient tensor is obtained by contracting the appropriate excluded tree index using the definition of the contraction operation from the previous section. For example, to obtain the $(\mu, K)$ pair-wise map, we may contract the $\nu$ index of the $(\mu, \nu, K)$ tree. The use of any pair-wise map that does not contain all elements resulting from this contraction operation amounts to making further physical approximations with respect to the multi-dimensional case - approximations that may or may not be justified.

We consider now the problem of looping over the significant elements of the fit coefficient tensor $C_{(\mu \nu K)}$ corresponding to the sparse tree shown in Fig. 5.6. While in this simplified example the simplest solution would be to loop over the tensor elements in order without consideration of sparsity at all, this is clearly not possible if another tensor must be accessed concurrently as, for example, in the case of contraction with another tensor $T_{\nu j}$. If we are restricted to using pair-wise maps within our nested loops, then the minimum-operation solution that retains the same physical content as the multi-dimensional loop is to run one outer loop over its full index range and restrict every inner loop by the applicable pair-wise sparse map(s) obtained by contracting the appropriate subspace(s) of the full multi-dimensional tree. We may recover the space of iterations performed by such nested pair-wise loops using our previously defined tree fusion operation. For example, given the loop nesting order $\mu, \nu, K$, we first contract each subspace of our tree individually to obtain the pair-wise maps $(\mu, \nu)$, $(\mu, K)$, and $(\nu, K)$. Then we fuse the $(\mu, \nu)$ and $(\nu, K)$ trees to produce a new tree suitable for looping over all three indices. We then prune the resulting tree by performing two-index fusion between it and our $(\mu, K)$ tree. In this case, the two-index fusion amounts to an “intersection” between the two trees, in the same sense as defined by Pinski et al. The resulting tree, the final tree in Fig. 5.6, represents the full space of iterations that would be performed by a set of nested loops over the pair-wise sparse maps. We have highlighted the red element to indicate an increase in the number of operations relative to the multi-dimensional case that we have started with. We further note that any fused tree composed from pair-wise maps that corresponds to a set of nested loops over all indices in the tensor must contain all entries from the original multi-dimensional tree. The combination of these two facts proves that the cost of the pair-wise implementation is an upper bound to the cost of the multi-dimensional implementation. We shall give concrete numerical examples of the cost savings associated with exploitation of multi-dimensionality in the Results section of this work.
Initial Tree

Contract Subspace 1

Contract Subspace 2

Contract Subspace 0

Fuse Subspace 1

Intersect Subspaces 0 and 2

Figure 5.6: Example of advantage of multi-dimensional sparsity relative to pair-wise sparsity. The red element is erroneously traversed in nested loops over the combined pair-wise sparsity, but not over the multi-dimensional original.

Automatic Loop Generation: A Fully Symbolic Library

In the interest of providing easily extensible, maintainable implementations of electronic structure methods, we wish to provide a library that does not require the explicit coding of any loops over indices or cumbersome machine generated code. The libtensor[213] library has provided such functionality for some time for dense tensors; in the present work, we have extended this library to treat block-sparse tensors. Our library utilizes C++ operator overloading to enable parsing of in-code symbolic sparse tensor expressions, allowing problems such as a fully sparse integral transformation to be converted into a single line of C++ code as shown in Fig. 5.7.

Our algorithm begins by choosing a loop order that traverses all indices in the problem. The problem of optimally ordering sparse nested loops to maximize floating point performance for a given operation is very complex. In the interest of simplicity of the initial implementation, we only make one potential rearrangement to the index order given in the contraction expression; namely, we decide whether or not the loops over contracted or uncontracted indices should be the outer loops. We make a crude estimate of the performance
CHAPTER 5. FULLY GENERAL SPARSE TENSOR FRAMEWORK FOR ELECTRONIC STRUCTURE THEORY

Symbolic Expression:
\[(\nu k|Q) = \sum_{\sigma} (\nu \sigma|Q) c_{\sigma k}\]

C++ Code:
\[H(\nu|Q|k) = \text{contract}(\sigma,I(\nu|\sigma|Q),C_{\text{mo}}(\sigma|k));\]

Figure 5.7: Conversion of symbolic tensor contraction expression to C++ code, as deployed in our development version of Q-Chem.

The tradeoff between these two choices by the comparing the product of the contracted dimensions with the uncontracted dimensions. This will favor reuse of input tensor elements in cases in which many indices are contracted, meaning that the output tensor is much lower order than the input tensors. This procedure is neither sophisticated nor optimal, and a more advanced treatment is desirable. However, we find that we are able to obtain good performance with the present implementation.

Our automatic loop generation code must handle an arbitrary number of indices across both sparse and dense tensors. In order to generate a canonical set of loops over all indices, we must first fuse all coupled sparse trees in the input and output tensors. The algorithm performing this process is summarized in Fig. 5.8. A given loop will traverse either a single sparse tree produced by the above loop fusion step (and therefore possibly traverse many indices at once) or traverse a single index that is dense in its occurrences in all tensors in the expression. While one could represent dense tensors as sparse trees with all block combinations included, this would introduce undesirable overhead.

Having obtained a canonical set of sparse trees, we prepare our nested loops to be run in two phases. First, we merge loops over indices that are coupled as members of the same sparse tree (note that they need not all be members of the same tensor) into a “loop group.” Universally dense indices each constitute a loop group unto themselves. Once the loop groups have been formed, we match each loop group to the list of all (tensor,index group) pairs that it touches. Finally, we generate the list of offsets into each (tensor,index group) pair for every iteration of the loop group. The loop grouping process is summarized in Fig. 5.9. It should be noted that our design does not allow interspersing loops over dense indices between loops over coupled sparse indices. This is an undesirable limitation on our library, and one which we are currently working to correct.

Finally, we proceed to execute the nested loops themselves. Our implementation uses a recursive algorithm in which each loop group corresponds to a unique recursion level. Our loop list execution routine takes a [block kernel] parameter, allowing the same loop traversal algorithm to implement many different block-level operations such as permutation and contraction. The algorithm by which we traverse all tensors and perform the requested operations in terms of individual block operations is summarized in Fig. 5.10.
Initialize $[\text{tree list}]$, list of all sparse trees in all tensors.
Permute all trees in $[\text{tree list}]$ to match index loop ordering.

Loop over indices $i$:
  Form $[\text{trees for loop}]$, list of $[\text{tree list}]$ entries touched by $i$ loop
  Loop over $\text{tree}$ in $[\text{trees for loop}]$:
    Fuse $\text{tree}$ to $[\text{tree list}]$ entry indexed by $[\text{trees for loop}]$ first element
    Remove $\text{tree}$ from $[\text{tree list}]$

Figure 5.8: Algorithm for performing fusion of sparsity across all tensors to form canonical set of trees.

Loop over indices $i$:
  If $i$ belongs to a $\text{tree}$ in $[\text{tree list}]$:
    Add $i$ loop to loop group for $\text{tree}$
  Else:
    Create new loop group for dense index $i$
    Add all (tensor,index group) pairs including $i$ to loop group

Loop over loop groups $g$:
  Loop over index tuples entries of $g$:
    Loop over (tensor,index group) pairs touched by $g$:
      Set appropriate entries in offset arrays

Figure 5.9: Algorithm for grouping loops over indices by sparse coupling.

Loop over offset list entries for current loop group:
  Loop over (tensor,index group) pairs touched by loop group:
    Scale $[\text{outer size}]$ and set $[\text{offset}]$ for (tensor,index group)
  If inner loop in loop group:
    Apply Eq. 5.2 to compute block offsets
    Call $[\text{block kernel}]$ on block pointers
  Else:
    Recurse down to next loop group

Figure 5.10: Nested loop execution algorithm for general sparse tensors.
An unfortunate side-affect of this implementation is that the trees representing coupled sparsity across multiple tensors are held explicitly in memory and looped over. These trees can become quite large due to their high dimensionality, in some cases comparable in size to some of the input tensors themselves. We have nonetheless successfully run jobs with over 10000 basis functions in augmented basis sets on nodes with 256-512GB of RAM, demonstrating that our implementation may be used in practical calculations. Work on an alternate formulation of the fusion steps that avoids this relatively high memory usage is in progress.

5.3 Results and Discussion

Multi-Dimensional Sparsity: Numerical Performance

We shall now provide numerical examples of the advantages of utilizing the full multidimensional index coupling present in higher-order tensors. We examine a series of globular water clusters, an example of which is shown in Fig. 5.11. For each of these clusters, we compute the half-transformed PARI fit coefficient tensor $D_{(\mu jQ)}$ given in Eq. 5.6. We then zero all blocks of $D_{(\mu jQ)}$ in which no element of a given block exceeds a threshold of $10^{-6}$ (based on the second order errors of the Dunlap robust fit[121] we may use such a loose threshold to screen RI fit coefficients). We now compare the operation count of a set of loops traversing the significant blocks of $D_{(\mu jQ)}$ using the full three-dimensional tree, versus the operation count yielded by using nested pair-wise sparse maps. As discussed in the Theory section, the nested loops over pair-wise sparse maps traverse a set of index tuples corresponding to the entries of a sparse tree created by a two step process; first, we generate three different two-dimensional sparse trees by contracting the full $D_{(\mu jQ)}$ tree over all three different subspaces. These trees formed by contracting subspaces 0, 1, and 2 are labeled $(jQ), (\mu Q),$ and $(\mu j)$, respectively. We then fuse these trees together in sequence over the appropriate shared indices to recreate a new “pair-wise” tree. As the numerical results in Table 5.1 demonstrate, the result is that we lose a very large factor of sparsity. Maintaining a fully multi-dimensional description of tensor sparsity is therefore necessary in order to keep arithmetic operations to a minimum.

Tensor Contraction Performance

We have already employed the sparse tensor library in this work in the context of implementing two high-performance applications[40, 121], and are in the process of reporting further applications in a separate publication. In order to highlight the generality of our library, we consider examples of three types of contractions: $[\text{sparse}] = [\text{sparse}] \ast [\text{sparse}]$, $[\text{sparse}] = [\text{sparse}] \ast [\text{dense}]$, and $[\text{dense}] = [\text{sparse}] \ast [\text{sparse}]$ Our chosen examples of these contraction types are chosen from the proceeding SCF-MI equations, and are summarized in Eqs. 5.13-5.15 in the order specified above.
Figure 5.11: A 46-molecule water cluster used in our performance benchmarks.

| System       | $D_{(\mu,j|Q)}$ % dense | $(\mu|j)$ % dense | $(j|Q)$ % dense | $(\mu|Q)$ % dense | pair-wise % dense |
|--------------|--------------------------|-------------------|-----------------|-------------------|-------------------|
| Water 46     | 1.0                      | 68                | 96              | 34                | 28                |
| Water 71     | 0.6                      | 55                | 88              | 28                | 19                |
| Water 127    | 0.3                      | 41                | 71              | 20                | 10                |

Table 5.1: Percentage of non-zero elements in the half-transformed PARI fit coefficients $D_{(\mu,j|Q)}$ defined in Eq. 5.6, illustrated in terms of different sparse trees. Percentages are based on block tuples, not on elements. The discrepancy between the first column and the “pair-wise” column is the sparsity lost by utilizing only a pair-wise representation. The aug-cc-pVTZ[199] basis set was used in conjunction with the cc-pVTZ-JK RI fitting basis set[39], and the half-transformed coefficients were screened with a threshold of $10^{-6}$.

$$D_{(\mu,j|Q)} = \sum_{\lambda} C_{Q}^{\mu}\lambda c_{(\lambda|j)}$$  \hspace{1cm} (5.13)

$$E^{*}_{i|Q} = \sum_{P} (D^{P})^{*}_{i|P} (P|Q)$$  \hspace{1cm} (5.14)

$$K_{i\nu} = -\sum_{jQ} E^{*}_{i|Q} D_{(\nu|jQ)}$$  \hspace{1cm} (5.15)

As a further illustration of the benefits yielded by the abstract symbolic nature of our library, our implementation uses only the following literal C++ code to implement these three contractions with full exploitation of sparsity:

```cpp
D_full(mu|Q |k) = contract(lambda,C_perm(mu|Q |lambda),C_mo(lambda |k));
E(ij|Q) = contract(R,Dpp_truncated_full(i|R|j),V(Q|R));
L3(i|nu) = contract(j|Q,E(ij|Q),D_truncated_full_perm(nu|i|Q));
```
Figure 5.12: Timings for all three contractions from Eqs. 5.13-5.15. Benzene stack calculations used the aug-cc-pVTZ[199] basis set with an integral threshold of $10^{-10}$. Reported times are for the first SCF iteration.

Figure 5.13: Stacked benzene molecules used to benchmark linear-scaling contractions

As a test system for benchmarking these tensor contractions, we have chosen linear stacked benzene ring structures such as that shown in Fig. 5.13. The spacing between consecutive benzene rings is 3.8 Å. We avoid the traditional linear benchmarks such as alkanes in this case because ALMOs are traditionally defined only for non-bonded fragments. For stack lengths of ten, twenty, and thirty rings, we time the first SCF iteration starting from a guess ALMO coefficient matrix consisting of the block-diagonal combination of the monomer MO coefficient matrices. Our timing results are shown in Fig. 5.12. The floating point performance of each contraction for the largest stack was measured as 1.3 GFLOPS, 4.3 GFLOPS, and 2.0 GFLOPS, respectively. These rates correspond respectively to 25%, 80%, and 38% and of single-core peak floating point performance (5.4 GFLOPS, as gauged by the performance of a 10000x10000 dense matrix multiplication using the Intel MKL library) for the single AMD Opteron 6376 processor on which these calculations were run. Thus, our tensor library captures the linear scaling behavior inherent in the sparse nature of these structures, and does so while retaining high floating point performance. This leaves the SCF-MI calculations dominated by the three-center AO integral calculation, the optimization of which we shall discuss in future work.
5.4 Conclusions

We have described a highly abstract, high performance implementation of common block-
 sparse tensor operations. Our implementation is capable of utilizing truly arbitrary sparsity
to reduce operation count without sacrificing floating-point performance. We have demon-
strated the capabilities of this library in the context of a high-performance SCF-MI code fully
described in a separate publication. The source code for our block-sparse library is available 
on GitHub as part of the libtensor project at URL https://github.com/epifanovsky/libtensor.
Work on adding greater concurrency support to the library is also in progress. We hope that
the capabilities of our library assist in the rapid development of more complex local electronic
structure methods.

5.5 Acknowledgements

This work was supported through the Scientific Discovery through Advanced Computing
(SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced
References


REFERENCES


(16) Zhao, Y.; Truhlar, D. G. The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals. *Theor. Chem. Acc.* **2008**, *120*, 215.


REFERENCES


REFERENCES


