OPTIMIZING COMPUTATIONAL KERNELS IN QUANTUM CHEMISTRY

A Dissertation
Presented to
The Academic Faculty

By

Matthew C. Schieber

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Computational Science and Engineering

Georgia Institute of Technology

February 2018

OPTIMIZING COMPUTATIONAL KERNELS IN QUANTUM CHEMISTRY

Approved by:

Dr. Burdell, Advisor
School of Myths
Georgia Institute of Technology

Dr. Two
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Three
School of Electrical Engineering
Georgia Institute of Technology

Dr. Four
School of Computer Science
Georgia Institute of Technology

Dr. Five
School of Public Policy
Georgia Institute of Technology

Dr. Six
School of Nuclear Engineering
Georgia Institute of Technology

Date Approved: January 11, 2000

A great quote to start the thesis

George P. Burdell

To my mom and dad.

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

TABLE OF CONTENTS

Acknowledgments v

List of Tables viii

List of Figures ix

Chapter 1: Introduction and Background 1

Chapter 2: Utilizing Spatial Sparsity 5

2.1 Sparsity Masks 5

2.2 Integral Construction 7

2.3 Integral Transformation 8

2.4 Proof of Implementation 9

Chapter 3: Optimizing Integral Transformations 13

3.1 A note on disk-bound blocking 13

Chapter 4: Discussion 15

Chapter 5: Conclusion 16

Appendix A: Experimental Equipment 18

Appendix B: Data Processing 19

References 19

Vita 20

LIST OF TABLES

2.1 Characteristics of benzene stack systems. N and N_{aux} refer to the number of primary and auxiliary basis functions, respectively. Mask sparsity refers to the percentage of significant AO function pairs in the sparsity mask. Mask sparsity increases with additional benzenes added to the stack. 10

2.2 Speedups obtained from sparsity screening at ten benzenes from data in Figure 2. 11

LIST OF FIGURES

2.1 Schwarz sparsity masks used for benzene stacks containing 1, 5, and 10 benzenes spaced 3 apart. Sparsity masks were obtained by evaluating $\sqrt{\mu(i)\mu(j)\nu(i)\nu(j)} < \tau$. 6

2.2 Comparison of execution times using sparsity screening (blue) against no sparsity screening (orange). Execution time is plotted against number of benzenes in a benzene stack from one to ten benzenes. Transformations involved computing $(ib|Q)$, where i and b denote occupied and virtual indices, respectively. Cutoff refers to the Schwarz screening threshold. (a) Computing the integrals. (b) Contracting AO integrals with the fitting metric. (c) first transformation times only. (d) total transformation times. 11

SUMMARY

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Electron repulsion integrals (ERIs) and operations involving them pose a fundamental computational problem for quantum chemistry. These 2-electron, 4-center integrals take the form:

$$(\mu\nu|\lambda\sigma) = \int\int \mu(\mathbf{r}_1)\nu(\mathbf{r}_1)r_{12}^{-1}\lambda(\mathbf{r}_2)\sigma(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{1.1}$$

Where $\mu, \nu, \lambda, \sigma$ are atomic orbital (AO) basis functions and $\mathbf{r}_1$ and $\mathbf{r}_2$ are electron coordinates. For simplicity, we have assumed the orbitals are real functions, as is usually the case in practice. Immediately, one should note the $\mathcal{O}(N_{AO}^4)$ storage costs (where N_{AO} is the number of AO basis functions), which quickly limits the size of in-core investigations. Moreover, the two fundamental operations involving ERIs are also costly. The first operation involves their evaluation to build the Coulomb and exchange matrices:

$$J[D_{\lambda\sigma}]_{\mu\nu} = (\mu\nu|\lambda\sigma)D_{\lambda\sigma} \tag{1.2}$$

$$K[D_{\lambda\sigma}]_{\mu\nu} = (\mu\lambda|\nu\sigma)D_{\lambda\sigma} \tag{1.3}$$

Where $J[D_{\lambda\sigma}]_{\mu\nu}$, $K[D_{\lambda\sigma}]_{\mu\nu}$, and $D_{\lambda\sigma}$ are the Coulomb, exchange, and density matrices, respectively. Evaluating the 4-center ERIs according to (1.2) and (1.3) requires $\mathcal{O}(N_{AO}^4)$ operations, which is a heavy computational burden that plagues even the most basic quantum chemistry procedures. The second fundamental operation when using ERIs involves their transformation from the AO basis to the molecular orbital (MO) basis. This operation

is written as:

$$(pq|rs) = C_{p\mu}C_{q\nu}(\mu\nu|\lambda\sigma)C_{r\lambda}C_{s\sigma}, \tag{1.4}$$

where p, q, r, s denote MO indices and orbital matrices, C, were introduced. These integral transformations scale as $\mathcal{O}(N_{AO}^4 N_p)$ (where N_p is the size of the denoted MO basis) and serve as a major bottleneck for perturbative correlation methods like second-order Møller-Plesset perturbation theory (MP2).

To overcome the drastic cost of evaluation, transforming, and processing the 4-center ERIs, various approximations and screening techniques have been developed. Density fitting is one method for alleviating these costs which has been gaining popularity in electronic structure theory. By utilizing the resolution of the identity technique, density fitting reduces the rank of the 4-center ERIs and approximates them using a 3-center form:

$$(\mu\nu|P) = \int\int \mu(\mathbf{r}_1)\nu(\mathbf{r}_1)r_{12}^{-1}P(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{1.5}$$

$$[J]_{PQ} = \int\int P(\mathbf{r}_1)r_{12}^{-1}Q(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{1.6}$$

$$(\mu\nu|\lambda\sigma) \approx (\mu\nu|P)[J]_{PQ}^{-1}(Q|\lambda\sigma) \tag{1.7}$$

Here we have introduced an auxiliary basis P, Q and the Coloumb fitting metric $[J]_{PQ}^{-1}$. In practice, permuational symmetry is almost always used so that only one set of these 3-center integrals need be built:

$$B_{\mu\nu}^P = [J]_{PQ}^{-\frac{1}{2}}(\mu\nu|Q) \tag{1.8}$$

$$(\lambda\sigma|\mu\nu) \approx B_{\mu\nu}^P B_{\lambda\sigma}^P \tag{1.9}$$

The B tensors indicate integrals that have been contracted with the fitting metric. The rank-reduced 3-center ERIs yield a space complexity of $\mathcal{O}(N_{aux}N_{AO}^2)$, which allows many

more investigations to be run in-core. Using the 3-center ERIsi, the Coulomb and exchange matrices are evaluated as:

$$J[D_{\lambda\sigma}]_{\mu\nu} = B^P_{\mu\nu} B^P_{\lambda\sigma} D_{\lambda\sigma} \tag{1.10}$$

$$K[D_{\lambda\sigma}]_{\mu\nu} = B^P_{\mu\lambda} B^P_{\nu\sigma} D_{\lambda\sigma} \tag{1.11}$$

Using (1.8), the Coulomb matrix is evaluated in $\mathcal{O}(N_{aux} N_{AO}^2)$ operations, which is a great success for reducing computational complexity. On the other hand, (1.9) still requires $\mathcal{O}(N_{aux} N^3)$ operations. Since N_{aux} is always larger than N_{AO}, evaluating (1.13) is actually more expensive than evaluating (1.10). This leaves exchange matrix evaluation as the major bottleck for most density-fitted procedures. One method for alleviating this bottleneck involves employing knowledge of the form of the density matrix:

$$D_{\lambda\sigma} = C_{p\lambda} C_{p\sigma} \tag{1.12}$$

Since MO spaces are often much smaller than AO spaces, it is possible to lower the complexity prefactor when building the exchange matrix. Decomposing the density matrix and rewriting the Coulomb and exchange matrix evaluations, we get:

$$J[D_{\lambda\sigma}]_{\mu\nu} = B^P_{\mu\nu} B_{\lambda\sigma} C_{p\lambda} C_{p\sigma} \tag{1.13}$$

$$K[D_{\lambda\sigma}]_{\mu\nu} = B^P_{\mu\lambda} B_{\nu\sigma} C_{p\lambda} C_{p\sigma} \tag{1.14}$$

Evaluating both (1.13) and (1.14) requires $\mathcal{O}(N_{aux} N_{AO}^2 N_p)$ operations. This technique is not benificial for Coulomb matrix evaluations, since the complexity of (1.13) is actually higher than that of (1.10). However, (1.14) reduces the complexity of exchange matrix evaluation by $\frac{N_{aux}}{p}$, which provides moderate speedups in practice.

Moreover, density fitting also improves the cost of integral transformations. Applying

(1.8) to (1.4), we have:

$$B_{pq}^{Q} = B_{\mu\nu}^{Q} C_{p\mu} C_{q\nu} \tag{1.15}$$

$$(pq|rs) \approx B_{pq}^{Q} B_{rs}^{Q} \tag{1.16}$$

The formal computational scaling of this operations is $\mathcal{O}(N_{aux} N_p N_q N_r N_s)$. However, in practice, algorithms based on density fitting often involve contracting the transformed 3-index integrals, B_{pq}^{Q}, with other terms, rather than explicity ofrmulation of $(pq|rs)$ as actual intermediates. Therefore, optimizing these procedures will primarily focus on optimizing the computation of B_{pq}^{Q}.

Although density fitting makes considerable progress towards reducing the space and computation complexity when using the ERIs, there is still work to be done. Importantly, (1.14) and (1.15) still serve as major computational bottlenecks for iterative and perturbative methods, respectively. In my research, I have sought out and implemented various optimizations to further improve the density fitting regime. The essense of my work involves the marriage of sparsity screening, parallel scaling, minimizing disk IO, and optimal contraction paths. This thesis details the progress made via my research. In Chapter 2, I discuss the spatial sparsity of the 3-index integrals and the utilization of such sparsity for integral construction, metric contractions, and integral transformations. In Chapter 3, I focus on (1.15) by discussing its implementation, blocking procedures, disk implications, and detail how the context in which transformations occur dictate which of two unique algorithms is superior. Finally, Chapter 4 focuses on (1.14) and displays the impact of my work for improving the state of the art.

CHAPTER 2

UTILIZING SPATIAL SPARSITY

For all but the smallest molecules, spatial sparsity is very important for achieving computational savings in quantum chemistry. There exists two primary forms of spatial sparsity in (1.3). The first form results from the Gaussian product $\mu(\mathbf{r}_1)\nu(\mathbf{r}_1)$ diminishing with the overlap of the two AO Gaussians. This product diminishes rapidly with distance between the Gaussian function centers and is also highly sensitive to their degree of locality. If the overlap between two AO Gaussians is insignificant, then it becomes unecessary to compute any shell triplet featuring this AO Gaussian pair. Utilizing this pairwise sparsity requires only an estimation of the significant AO Gaussian pairs. Once this spatial sparsity is applied and insignificant AO function pairs are screened, the complexity of computing the 3-center integrals becomes $\mathcal{O}(N_{aux}N_{AO}^{1-2})$, where the lower bound is achieved for sufficiently sparse systems. Another type of spatial sparsity in (1.3) involves the auxiliary function, which is more complicated to take advantage of. see Ref. [**Werner:2003:8149**]. In this thesis, I focus on the former, pairwise sparsity which is featured in the product $\mu(\mathbf{r}_1)\nu(\mathbf{r}_1)$. Utilizing this sparsity not only speeds up the compoutation of the three-index integrals, but stored in a sparse format, then this sparsity can be utilized in later operations such as the fitting metric contraction and basis transformations.

2.1 Sparsity Masks

The pairwise sparsity available to the 3-center integrals can be determined via an estimation of significant AO function pairs. An implementation will require a two dimensional sparsity mapping structure which keeps track of the significant pairs. We refer to these structures as sparsity masks, which are N_{AO} by N_{AO} boolean matrices denoted as $S^b_{\mu\nu}$. The superscript b reminds the reader that the structure is of boolean type. The element $S^b_{\mu\nu}$ will be true if

the product $\mu(r)\nu(r)$ is estimated to be significant and false otherwise.

For efficiency, it is common to compute batches of integrals in *shell triples*:

$$(MN|Q) = \{(\mu\nu|P), \mu \in M, \nu \in N, Q \in P\} \tag{2.1}$$

A shell pair (M, N) is significant if

$$\max \sqrt{\mu(M)\mu(N)\nu(M)\nu(N)} < \tau. \tag{2.2}$$

Where τ is the screening threshold. This is known as Schwarz screening. Moreover, additional sparsity can be achieved if the screening is applied to function pairs rather than entire shell pairs. Then, a function pair (i, j) is significant if

$$\sqrt{\mu(i)\mu(j)\nu(i)\nu(j)} < \tau. \tag{2.3}$$

Using this strategy, a sparsity mask can be easily constructed. Figure 1 includes illustrations of sparsity masks for benzene stacks with 1, 5, and 10 benzenes spaced 3 Å apart. As sparsity increases, the number of significant AO functin pairs contained in these masks becomes $\mathcal{O}(N_{AO}^{1-2})$, where the lower bound is obtained for sufficiently sparse systems.

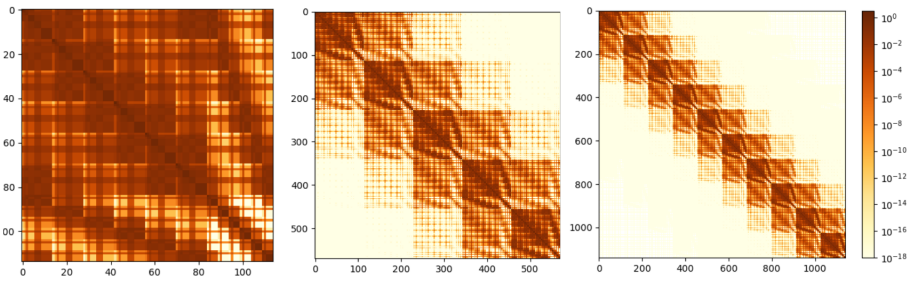

Figure 2.1: Schwarz sparsity masks used for benzene stacks containing 1, 5, and 10 benzenes spaced 3 apart. Sparsity masks were obtained by evaluating $\sqrt{\mu(i)\mu(j)\nu(i)\nu(j)} < \tau$.

2.2 Integral Construction

Now, we are equipped to discuss the resulting algorithms when applying the integral sparsity to important operations such as integral construction and transformation. To write more readible algorithms, we employ some tensor notation and write $A^P_{\mu\nu}$ to represent the pre-contracted 3-center integrals $(\mu\nu|P)$. To prune these integrals using sparsity, the following algorithm can be employed:

Algorithm 1 Prune $A^P_{\mu\nu}$ using sparsity

Require: AO integrals: $A^P_{\mu\nu}$, screening mask: $S^b_{\mu\nu}$
  for $\mu = 0$ to $\mu = N_{AO} - 1$ **do**
    $A^P_{\mu\nu} S^b_{\mu\nu} \rightarrow A^P_{\mu\nu^\mu}$
  end for
  return $A^P_{\mu\nu^\mu}$

Here we have introduced the symbol ν^μ, which indicates that ν is restricted to AO functions which are spatially close enough to μ to survive the Schwarz screening process. The superscript in ν^μ indicates a dependence of ν according to μ. Algorithm 1 is purely pedagogical, as one would never build the full $A^P_{\mu\nu}$ integrals and then prune them for sparsity. Rather, sparsity screening would be applied as the integrals are constructed and insignificant function triplets are never computed. The computation and storage of $A^P_{\mu\nu^\mu}$ scales as $\mathcal{O}(N_{aux} N_{AO}^{1-2})$. Moroever, the costly metric contraction becomes

$$B^Q_{\mu\nu^\mu} = A^P_{\mu\nu^\mu} [J]^{-1}_{PQ} \tag{2.4}$$

and scales as $\mathcal{O}(N^2_{aux} N_{AO}^{1-2})$, where the lower bound is achieved for sufficiently sparse systems.

2.3 Integral Transformation

The transformation of $A^P_{\mu\nu}$ (or $B^Q_{\mu\nu^\nu}$) from the AO basis to the MO basis is an essential operation for many quantum chemistry procedures. The operation requires two MO matrices: $C_{\mu p}, C_{\nu q}$, where p, q are general MO space indices. A first contraction $A^P_{p\nu} = A^P_{\mu\nu} C_{\mu p}$ will half-transform the integrals and costs $\mathcal{O}(N_{aux} N^2_{AO} N_p)$. The second contraction $A^P_{pq} = A^P_{p\nu} C_{\nu q}$ will cost $\mathcal{O}(N_{aux} N_{AO} N_p N_q)$. Note the first contraction should involve the smaller of N_p, N_q in order to reduce complexity. Also, the first contraction is comparably more expensive than the latter since the size of the AO space is larger than most MO spaces. Therefore, reducing the cost of the first contraction would alleviate a bottleneck overall. Thankfully, we can exploit the sparsity of $A^P_{\mu\nu^\mu}$. To do so, we carry out a looping DGEMM through the μ index and apply the sparsity mask to the orbital matrix $C_{\mu p}$. Algorithm 2 illustrates the resulting process:

Algorithm 2 Transform sparse integrals $A^P_{\mu\nu^\mu}$ to MO spaces.

Require: Sparse AO integrals: $A^P_{\mu\nu^\mu}$, orbital matrices: $C_{\mu p}, C_{\nu q}$, screening mask: $S^b_{\mu\nu}$

for $\mu = 0$ to $\mu = N_{AO} - 1$ **do**

$C_{\nu p} S^b_{\mu\nu} \rightarrow C_{\nu^\mu p}$

$A^P_{\mu\nu^\mu} C_{\nu^\mu p} \rightarrow A^P_{\mu p}$

end for

return $A^P_{\mu q}$

$A^P_{\mu q} C_{\mu p} \rightarrow A_{pq}$

return A^P_{pq}

The cost of the first contraction becomes $\mathcal{O}(N_{aux} N^{1-2}_{AO} N_p)$. The remaining sparsity in the half-transformed $A^P_{\nu q}$ is unrelated to the original sparsity mask.

2.4 Proof of Implementation

All methods were implemented in the PSI4 electronic structure software package. The parallelism in PSI4 relies on the shared memory programming model using OpenMP and carries out matrix multiplications using Intel's Math Kernel Library.

We first demonstrate the performance of our Schwarz screening implementation by measuring the performance when computing the 3-center integrals, contracting the fitting metric, and transforming the integrals into an MO basis. The experiment employed an ideal sparse system: stacked benzenes. Execution times were recorded for each successive benzene added to the stack, from one to ten benzenes, with each benzene spaced 5Å apart. Transformation time involved the wall time required to carry out the common occupied-virtual transformation, as would be required by density-fitted MP2:

$$(ib|Q) = (\lambda\sigma|Q)C_{\sigma i}C_{\lambda b} \tag{2.5}$$

Where i, j and a, b denote occupied and virtual spaces, respectively. Algorithm 4 was implemented and used to carry out these transformations. The cc-pVTZ and cc-pVTZ-jkfit basis sets were used for primary and auxiliary basis sets, respectively. The characteristics of each system are listed in Table 2. The mask sparsity listed in Table 2 refers to the percentage of non-significant AO function pairs appearing in the sparsity mask. The experiment was carried out using one node consisting of an Intel Core i7-5930K processor (6 cores at 3.50GHz) and 50GB DRAM. The results are plotted in Figure 2. Figure 2 (a), (b), (c), and (d) involve time to compute the 3-index integrals, contract the fitting metric, perform the first transformation step: $(\mu\nu|Q)C_{i\mu} \rightarrow (i\nu|Q)$, and total transform time, respectively. Note that our novel contribution involves the application of the sparsity screening in the transformation step.

Table 2.1: Characteristics of benzene stack systems. N and N_{aux} refer to the number of primary and auxiliary basis functions, respectively. Mask sparsity refers to the percentage of significant AO function pairs in the sparsity mask. Mask sparsity increases with additional benzenes added to the stack.

Benzenes	N	N_{aux}	**Mask Sparsity (%)**
1	264	654	2.6
2	528	1308	24.7
3	792	1962	43.6
4	1056	2616	55.3
5	1320	3270	63.1
6	1584	3924	68.6
7	1848	4578	72.7
8	2112	5232	75.9
9	2376	5886	78.4
10	2640	6540	80.4

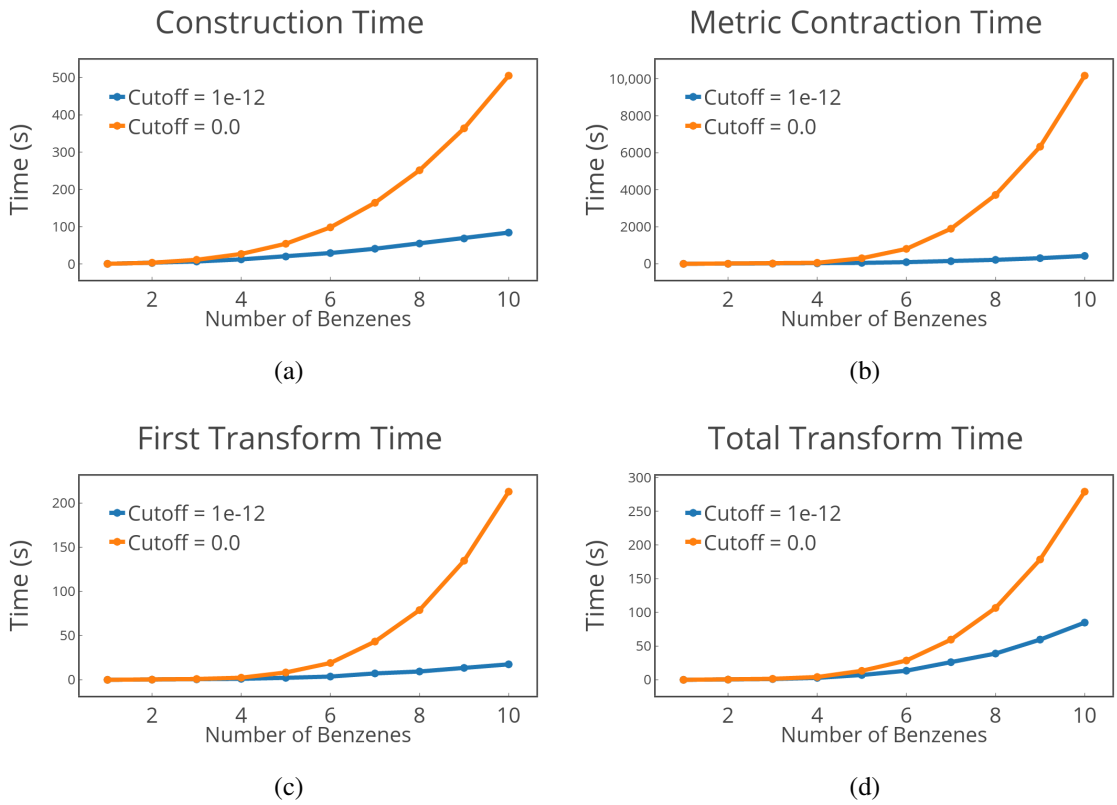

Figure 2.2: Comparison of execution times using sparsity screening (blue) against no sparsity screening (orange). Execution time is plotted against number of benzenes in a benzene stack from one to ten benzenes. Transformations involved computing $(ib|Q)$, where i and b denote occupied and virtual indices, respectively. Cutoff refers to the Schwarz screening threshold. (a) Computing the integrals. (b) Contracting AO integrals with the fitting metric. (c) first transformation times only. (d) total transformation times.

Table 2.2: Speedups obtained from sparsity screening at ten benzenes from data in Figure 2.

Operation	**Speedup at 10 benzenes**
Construction	10.6
Metric Contraction	37.8
First Transformation	25.8
Total Transformation	5.5

Clearly, Figure 2 reveals significant time reductions for all procedures measured. Table 3 reinforces that operations with higher complexity scaling have the largest time reduction,

with the caveat that total transformation time includes portions without sparsity utilization. In the integral computations (Figure 2 (a)), we construct the sparse integrals using function screening; however, we still compute them in shell triplets for effeciency. Therefore the acquired speedup is slightly less dramatic compared to the metric contraction and transformation procedures (Figure 2 (b) and (c), respectively) due to select functions being screened in cases where the entire shell is not screened. Note that the metric contraction is by far the most expensive operation, which should in one part highlight the boon of sparsity utilization and in another part illustrate the pertinence of our workflow investigation later in this paper. Last, our proposal outlined in Algorithm 2 for applying sparsity screening to integral transformations is proven viable by Figure 2 (c). Note that significant reductions are obtained for the first transformation; however, the sparsity thereafter is unrelated to the initial sparsity mask. With sparsity utilization, the first step scales as $\mathcal{O}(N_{aux}N_{AO}^{1-2}p)$, whereas the second transformation step will still scale as $\mathcal{O}(N_{aux}N_pN_q)$. In this work we have not gone on to consider sparsity of the MO indices, which would typically require transformation to local orbitals.

CHAPTER 3

OPTIMIZING INTEGRAL TRANSFORMATIONS

3.1 A note on disk-bound blocking

The size of tensors grow rapidly in quantum chemistry. The 3-center integrals of density fitting are no exception. Often, as the size of the AO three-index integrals will exceed 50GB of RAM for systems as small as 40 atoms when a moderate basis set such as aug-cc-pVQZ is used, even with sparsity screening. At that point, it is necessary for any implementation to begin reading and writing these tensors to and from disk-based memory. For any field this can be a major slowdown, but it is especially critical for performance when high dimensional data is involved. To illustrate this issue, I will introduce an adapted tensor notation which better indicates memory layout.

We denote an n-dimensional tensor as $T_{ab...n}$, where the indices from left to right go from the slowest-running to the fastest-running indices. Here, a is the slowest-running index, b is the next slowest-running index, and n is the fastest-running index. The choice of memory layout plays a crucial role when indices are being accessed. Iterating through the slowest-running index, a, would require the largest memory strides whereas the elements of the fastest-running index, n, are contiguous in memory.

Now, we can consider two possible forms for the 3-index integrals: $A_{P\mu\nu}$ or $A_{\mu P\mu}$. If these tensors are too large to fit into memory, we must read and write pieces of them to and from disk-based memory. To accomplish this, we must choose an index to block accross. Primarily, this will involve either the P or μ index. For example, if we choose to block accross the P index, then we will partition the basis of P into discrete blocks:

$$a; lkds \tag{3.1}$$

Then, we will read and write only those blocks of P along with all of μ and ν. The latency of these operations is bound by the movement of a physical read-write head, so it is critically important to ensure that read and writes are as contiguous as possible. In the case of P blocking, the $A_{P\mu\nu}$ tensor is far superior to $A_{\mu P\mu}$ since the former will yield entirely contiguous operations whereas the latter will require strided operations.

CHAPTER 4

DISCUSSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum..

CHAPTER 5
CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Appendices

APPENDIX A
EXPERIMENTAL EQUIPMENT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

APPENDIX B

DATA PROCESSING

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

VITA

Vita may be provided by doctoral students only. The length of the vita is preferably one page. It may include the place of birth and should be written in third person. This vita is similar to the author biography found on book jackets.