MASTER THESIS

Low-Rank Representation of Two-Electron Integrals

Theoretical Overview and Implementation

Submitted by

Niklas Paulicks (Mat. Nr. 7611688)
niklas.paulicks@studium.uni-hamburg.de
Department of Informatics of the MIN Faculty
Universität Hamburg

in the study program

Master of Science Bioinformatics

on

February 19, 2026

Examiners:

Dr. Johannes Tölle

Prof. Dr. Andrew Torda

List of Contents

1 Introduction 7
2 Tensor decompositions 8
2.1 Fundamental Tensor Operations 8
2.2 Tensor Decompositions 9
2.3 Eigen-, Schur- and Singular Value Decomposition 10
2.4 QR Decomposition 11
2.5 Cholesky Decomposition 14
2.6 Interpolative Decomposition 17
2.7 CP Decomposition 18
3 Tensors in Electronic Structure Theory 21
3.1 Hartree-Fock Theory 23
3.2 Møller-Plesset Perturbation Theory 24
3.3 Coupled Cluster Methods 28
4 Low-Rank Approximations of the ERI Tensor 30
4.1 Properties of Electron Pair Repulsion 30
4.2 Density Fitting / Resolution of the Identity 31
4.3 Tensor Hypercontraction 33
4.4 Least-squares THC 36
4.5 Interpolative Separable Density Fitting 38
4.6 Improvements upon LS-THC 40
4.7 Block Tensor Decomposition 42
4.8 CP Decomposition 43
4.9 Closing Remarks 44
5 Implementation 46
6 Experimental Validation 52
6.1 Parameter Search 53
6.2 Full Comparative Benchmark on Small Systems 56
6.3 Calculating Bigger Systems 61
7 Discussion 65
7.1 Limitations by the Test-Data 65
7.2 Limitations by Density Fitting 65
7.3 Limitations of the Grid 67
7.4 Limitations from Underlying Assumptions 69
7.5 Practical Considerations 71
8 Conclusion and Future Work 72
References 73
A Detailed Benchmarking Results 83
B Grid Pruning on Different Molecules 88
C Used Software and Versions 91

List of Figures

Figure 1 Molecule sets used for testing 52
Figure 2 Parameter Search - LS-THC 53
Figure 3 Parameter Search - CD-LS-THC 54

Figure 4 Parameter Search - ISDF-KM ... 54
Figure 5 Parameter Search - ISDF-QR ... 55
Figure 6 Parameter Search - DF-THC ... 56
Figure 7 Benchmark – Comparison ... 57
Figure 8 Benchmark error metrics MP2 ... 57
Figure 9 Benchmark error metrics HF-JK ... 58
Figure 10 Grid pruning ratios ... 58
Figure 11 Empirical scaling of MP2 algorithms approximation methods ... 61
Figure 12 Detailed runtimes of steps in THC-Laplace MP2 ... 62
Figure 13 DNA fragment ... 64
Figure 14 Relative approximation errors with full ERI fitting ... 66
Figure 15 Effect of different auxiliary basis sets on MP2 approximation errors ... 66
Figure 16 AO and MO wavefunctions and products of ethane ... 68
Figure 17 AO and MO wavefunctions and products of ethane with ghost atom ... 68
Figure 18 Energy ratios of THC approximations to canonical values ... 70
Figure 19 Energy ratios of ISDF-KM HF-JK compared to DF ... 70
Figure A.1 Benchmark time scaling - LS-THC ... 83
Figure A.2 Benchmark error metrics - LS-THC ... 83
Figure A.3 Benchmark time scaling - CD-LS-THC ... 84
Figure A.4 Benchmark error metrics - CD-LS-THC ... 84
Figure A.5 Benchmark time scaling - ISDF-QR ... 85
Figure A.6 Benchmark error metrics - ISDF-QR ... 85
Figure A.7 Benchmark time scaling - ISDF-KM ... 86
Figure A.8 Benchmark error metrics - ISDF-KM ... 86
Figure A.9 Benchmark time scaling - DF-THC ... 87
Figure A.10 Benchmark error metrics - DF-THC ... 87
Figure B.1 Pruned grid alkane - CD-LS-THC ... 88
Figure B.2 Pruned grid alkane - ISDF-QR ... 88
Figure B.3 Pruned grid alkane - ISDF-KM ... 89
Figure B.4 Pruned grid naphthalene - CD-LS-THC ... 89
Figure B.5 Pruned grid naphthalene - ISDF-QR ... 90
Figure B.6 Pruned grid naphthalene - ISDF-KM ... 90

List of Tables

Table 1 Chosen method parameters ... 56
Table 2 Empirical scaling of ERI approximation methods ... 58
Table 3 Error metrics of ERI approximation methods ... 59
Table 4 Empirical scaling of MP2 algorithms approximation methods ... 61
Table 5 Empirical scaling of build steps in the THC-Laplace MP2 algorithm. ... 62
Table 6 Relative errors for ethane with denser grids ... 68
Table C.1 Used software versions ... 91

List of Algorithms

Algorithm 1 Row-wise PivotedCholesky with trace error minimization ... 15
Algorithm 2 CP-ALS Algorithm ... 19

Algorithm 3 Laplace Transform MP2 Energy using the THC ERI 36
Algorithm 4 Weighted K-Means Clustering 39
Algorithm 5 Least Squares THC 46
Algorithm 6 Cholesky Decomposition pruned LS-THC 47
Algorithm 7 ISDF QRCP 47
Algorithm 8 ISDF with K-Means Clustering 48
Algorithm 9 Density Fitting THC 51

Glossary

ALS ..... Alternating Least Squares
AO ..... Atomic Orbital
BTD ..... Block Tensor Decomposition
CC ..... Coupled Cluster
CCS ..... Coupled Cluster Singles
CCSD ..... Coupled Cluster Singles Doubles
CD ..... Cholesky Decomposition
CP-THC ..... Canonic Polyadic Tensor Hypercontraction
CPD ..... CANDECOMP/PARAFAC Decomposition
CVT ..... Centroidal Voronoi Tesselation
DF ..... Density Fitting
DFT ..... Density Functional Theory
ERI ..... Electron Repulsion Integral
FFT ..... Fast Fourier Transform
FMM ..... Fast Multipole Method
HF ..... Hartree-Fock Theory
ID ..... Interpolative Decomposition
ISDF ..... Interpolative Separable Density Fitting
LCAO ..... Linear Combination of Atomic Orbitals
LS ..... Least Squares
LS-THC ..... Least Squares Tensor Hypercontraction
MAE ..... Mean Average Error
MBPT ..... Many Body Perturbation Theory
MO ..... Molecular Orbital
MP2 ..... Møller–Plesset Perturbation Theory (Second Order)
QCISD ..... Quadratic Configuration Interaction Singled Doubles
QRCP ..... QR with Column Pivoting
RI ..... Resolution of the Identity
RPA ..... Random Phase Approximation

SAD ..... Superposition of Atomic Densities
SCF ..... Self Consistent Field
SCS ..... Spin Component Scaled
SOS ..... Scaled opposite-spin
SVD ..... Singular Value Decomposition
THC ..... Tensor Hypercontraction
psd ..... positive-semidefinite

1 Introduction

Studying the properties of matter in the framework of quantum mechanics has been a main concern of physics and chemistry since the early 20th century. Problems in quantum chemistry usually involve finding approximate solutions to the Schrödinger equation, which describes the behavior of a system's electronic wavefunction. Methods like Kohn-Sham Density Functional Theory (DFT), Hartree-Fock Theory (HF), Coupled Cluster (CC) methods, Many Body Perturbation Theory (MBPT) and many others have been developed that provide solutions in the context of different approximations of the wavefunction or the Hamiltonian operator[1,2].

The invention and steady improvement in the field of computerized numerics has highly accelerated the progress made in theoretical quantum chemistry. Recent reviews[3] show that computations involving many thousands of atoms are now attainable using DFT. Wavefunction-based methods that consider electronic correlation effects on the other hand become impractical for much smaller systems, due to the size of the integrals needed to model the electron-electron interactions. These Electron Repulsion Integrals (ERIs) exhibit steep polynomial growth with the system's number of basis functions, which means that storing the full two-electron repulsion integral for the largest system considered in this work (3,528 basis functions) would need ≈ 140 TiB[F.1] of system memory.

While these integrals are difficult to avoid, some calculations might not need their full information content when used on specific kinds of chemical systems. Therefore, researchers developed methods like Density Fitting (DF)[4,5], the Tensor Hypercontraction (THC)[6,7] or the CANDECOMP/PARAFAC Decomposition (CPD)[8] that project them into a lower dimensional space. This allows approximate computations for system sizes that would otherwise be costly or entirely intractable on current hardware. Those methods derive from various techniques developed in the field of low-rank tensor approximations.

Unfortunately, the research field of low-rank ERI representations is fractured. The existing solutions are scattered throughout different quantum chemistry codes and benchmarked in ways that do not allow a fair comparison. This is especially true for the THC method-family where multiple derivations have been proposed, but no single approach or stable implementation dominates. This is contrasted by DF, which is well-established in the literature and has seen wide adoption in production software, and the CPD, whose direct application to the ERI remains limited by computational inefficiency and non-linear convergence challenges.

In this thesis we will therefore establish a theoretical foundation and unified notation (Section 2, Section 3 and Section 4); construct a reference implementation of five distinct ways to build the THC (Section 5); benchmark those implementations on a unified set of molecular systems (Section 6) and discuss the results (Section 7). With this work, we hope to aid future developments in this area of research by improving clarity and accessibility, providing a solid platform for the rigorous evaluation of next-generation approximation methods.

[F.1] Assuming a scaling of N_{basis}^4 with 8-fold permutational symmetry and double precision floating point numbers.

2 Tensor decompositions

A main concern of the field of linear algebra in computer science is analyzing properties of tensors. The notion of what a tensor is as well as most of the notation used in this thesis, follows the work of Kolda, T. G.; Bader, B. W. [9], which provides an exceptional overview over the field of tensor decomposition and is often referenced throughout literature[6,10–12]. In the context of this thesis tensors are viewed as multidimensional arrays of numbers that can be used to model connections between other algebraic objects. This should be distinguished from tensors in the theory of relativity, which provide more structure and are referred to by some mathematicians as "tensor-fields". Every tensor can have 0 to N dimensions, which are referred to as orders or modes. While order-0 tensors are just scalar values, order-1 tensors are called vectors and order-2 tensors are called matrices. For any higher-order tensor, the dimensionality will be explicitly stated (order-3, order-4 …).

2.1 Fundamental Tensor Operations

Since operations including tensors up to order-2 are well studied we will take all standard operations (e.g., scalar-, vector- and matrix products, inner- and outer-products, transposes, the inverse, orthogonality…) as given as far as they are covered in books meant for teaching such as the ones by Axler, S. [13] and Strang, G. [14]. The study of higher order tensors usually involves some more complex operations which we will now introduce.

Tensor contractions can be viewed as a generalization of matrix-multiplication[F.2] that merge an order-N tensor with another of order-M. If the order is finite (if we do not consider the case of a generic tensor), the dimensions can be named and added as indices ($\boldsymbol{T}_{abcd}$ would describe an order-4 tensor). In this thesis we will mostly be confronted with finite dimensional tensors and therefore use the Einstein summation convention, which simplifies tensor-tensor contractions like $\boldsymbol{R}^{cd} = \sum_{ab} \boldsymbol{T}_{abcd}\boldsymbol{M}_{ab}$ by omitting the summation sign making summation over shared indices implicit. In this way the given formula can be shortened to $\boldsymbol{R}^{cd} = \boldsymbol{T}_{abcd}\boldsymbol{M}_{ab}$. In some cases we will use explicit sums, if contractions are carried out partially. As a convention all tensors of order-2 or higher will be denoted by bold font capital letters, while vectors will be written in bold-font lower-case and scalars will be marked by lower-case letters. Note that upper- or lower indices do not denote covariance and contravariance as is customary in the theory of relativity.

Unfolding describes the process of re-arranging a given tensor $\boldsymbol{T}$ into a different shape. This process is also called matricization since it can be used to convert a higher order tensor into a matrix. A special case is the "mode-n" unfolding in which a given mode of a tensor $\boldsymbol{T}$ is used to form the columns of a matrix. In that spirit the tensor $\boldsymbol{T}_{abc} \in \mathbb{R}^{3\times4\times4}$ has a mode-1 unfolding matrix denoted $\boldsymbol{T}_{(1)}$ or $\boldsymbol{T}^{a}_{\overline{bc}}$ with

$$\boldsymbol{T}_{(1)} \in \mathbb{R}^{3\times(4\cdot4=16)}. \tag{2.1}$$

The **Hadamard product** of two tensors of the *same order* is defined as the element-wise product of their respective dimensions. For order-2 tensors $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$ this means that

[F.2] We acknowledge that in some contexts "contraction" only refers to operations that reduce the order of the input object. We will however use contraction to refer to any multiplication and summation operation on two tensors that yield another tensor.

$$\boldsymbol{A} * \boldsymbol{B} = \begin{bmatrix} a_{11}b_{11} & \dots & a_{1n}b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & \dots & a_{mn}b_{mn} \end{bmatrix}. \tag{2.2}$$

Note that the field over which the tensors are defined is not necessarily restricted to the real numbers $\mathbb{R}$, but can in theory be any other field such as the complex numbers $\mathbb{C}$.

The **Kronecker product** of two tensors $\boldsymbol{A} \in \mathbb{R}^{i \times j}$ and $\boldsymbol{B} \in \mathbb{R}^{k \times l}$ is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix} \tag{2.3}$$

where $a_{11}\boldsymbol{B}$ denotes the scalar-matrix product.

The face-splitting or **Khatri-Rao product** of two tensors $\boldsymbol{A} \in \mathbb{R}^{i \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{j \times k}$ is defined as the matching, column-wise Kronecker product of $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \dots & \boldsymbol{a}_k \otimes \boldsymbol{b}_k \end{bmatrix}^T. \tag{2.4}$$

Additionally, the symbol $\circ$ is used to denote the outer product. This means an order-3 tensor $\boldsymbol{T}$ can be constructed from vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ like

$$\boldsymbol{T}_{abc} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}. \tag{2.5}$$

2.2 Tensor Decompositions

The field of linear algebra is broad and dates back many decades. While the term "matrix" was coined in 1848, the interest in studying their numerical properties increased drastically after World War II with the advent of modern computers. Since then, many ways of gathering information about a given tensor by means of decomposing it were developed[15]. **Tensor decomposition** techniques usually try to express a tensor $\boldsymbol{A}$ as a contraction of other tensors, which possess some known properties. By studying those component tensors, insights might be gained about the structure of $\boldsymbol{A}$. This chapter will try to outline some tensor decompositions and their computational properties. As there are many techniques for decomposing different classes of tensors, we do not aim to provide a holistic analysis of the field, but present only some selected approaches that will be useful in Section 4. Also, we will not comprehensively outline all the ways to compute those decomposition methods, but only sketch the most prominent ones which are employed in the software libraries used in this thesis.

Strang, G. [14] lists the **rank** of a matrix as one of the central ideas in linear algebra and defines it as the "[…] count of genuinely independent rows in a matrix $\boldsymbol{A}$" and the "[…] dimension of the column space [of $\boldsymbol{A}$]". A matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is full-rank if $\text{rank}(\boldsymbol{A}) = \min\{m, n\}$ and rank-deficient if $\text{rank}(\boldsymbol{A}) < \min\{m, n\}$. When viewing $\boldsymbol{A}$ as a geometric transformation acting on a vector $\boldsymbol{x}$ its rank determines whether $\boldsymbol{x}$ retains its full dimensionality. In the case of a rank-deficient $\boldsymbol{A}$, the transformation $\boldsymbol{A}\boldsymbol{x}$ will collapse $n - \text{rank}(\boldsymbol{A})$ dimensions of $\boldsymbol{x}$ to $\boldsymbol{0}$. The idea of a **low-rank approximation** of $\boldsymbol{A}$ is to find a matrix $\hat{\boldsymbol{A}}$ with $\text{rank}\left(\hat{\boldsymbol{A}}\right) < \text{rank}(\boldsymbol{A})$ that, retains most of the transformational properties of $\boldsymbol{A}$ such that $\hat{\boldsymbol{A}}\boldsymbol{x} \approx \boldsymbol{A}\boldsymbol{x}$.

This intuitive notion of **rank** does not transfer to tensors of order > 2. Kolda, T. G.; Bader, B. W. [9] defines the rank of a generic tensor $\boldsymbol{T}$ as the minimal number of rank-one tensors that generate $\boldsymbol{T}$ as their sum. A rank-one tensor construction is shown in Eq. (2.5). We will elaborate further on

the concept of tensor-rank in Section 2.7. The main problem with ranks of higher order tensors is that there is no straightforward way to compute them. In fact, it is proven to be NP-hard, which means that we might not be able to determine the correct rank in polynomial time and that even checking if a given answer is correct might not be possible in polynomial time depending on the chosen field[16].

In the following sections we will explore different method of analyzing tensors that lead to ways of constructing low-rank approximations.

2.3 Eigen-, Schur- and Singular Value Decomposition

When analyzing order-2 tensors, a powerful tool to gain insight about the transformational properties of the matrix is given by analyzing its eigenvalues and eigenvectors. The eigenvectors can be thought of as spanning a space that the matrix projects vectors on to, whereas the eigenvalues tell us how much a vector is "stretched" or "rotated" along each (real- or complex) eigenvector.

Eigen-decomposition: The spectral theorem states that every real, symmetric matrix $\boldsymbol{A}$ is similar (has the same eigenvalues) to a diagonal matrix, so it can be decomposed into

$$\boldsymbol{A} = \boldsymbol{Q}^T \boldsymbol{\Lambda} \boldsymbol{Q} \tag{2.6}$$

where the columns of $\boldsymbol{Q}$ are the orthogonal eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{A}$. In the complex case where $\boldsymbol{A}$ is hermitian, $\boldsymbol{Q}$ is usually called $\boldsymbol{U}$ and is a unitary matrix.[14]

Schur-decomposition: Although originating in group theory, Schur's lemma implies that any square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ can be decomposed into an orthogonal matrix $\boldsymbol{U}$ and an upper triangular matrix $\boldsymbol{T}$, which holds the eigenvalues of matrix $\boldsymbol{A}$ on its diagonal

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^T. \tag{2.7}$$

Singular Value Decomposition: Another decomposition similar to the eigendecomposition can be found for **any** matrix (not just square matrices) if $\boldsymbol{Q}$ is allowed to differ from $\boldsymbol{Q}^T$. The Singular Value Decomposition (SVD) of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\boldsymbol{A}) \leq \min\{m, n\}$ is usually written as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{2.8}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$, $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$[14]. If we suppose that $\boldsymbol{\Sigma}$ is diagonal and holds the square roots of the eigenvalues and that $\boldsymbol{U}$ and $\boldsymbol{V}$ are distinct but still orthogonal, we can reconstruct Eq. (2.6) for the square positive-semidefinite (psd) matrix $\boldsymbol{A}^T\boldsymbol{A}$

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{A} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T) \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^T\cancel{\boldsymbol{U}^T\boldsymbol{U}}\boldsymbol{\Sigma}\boldsymbol{V}^T \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})\boldsymbol{V}^T. \end{aligned} \tag{2.9}$$

The elements on the diagonal of $\boldsymbol{\Sigma}$ are called singular values of matrix $\boldsymbol{A}$ and are usually denoted as σ_i.[14]

As stated in the Eckart-Young-Mirsky theorem[17], the SVD provides the optimal low-rank approximation in any unitarily invariant norm (like $\|.\|_F$ or $\|.\|_2$). Therefore, the matrix $\hat{\boldsymbol{A}}$ resulting from the truncated SVD

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = [\boldsymbol{U}_1 \;\; \boldsymbol{U}_2] \left[\begin{array}{c|c} \boldsymbol{\Sigma}_1 & 0 \\ \hline 0 & \boldsymbol{\Sigma}_2 \end{array}\right] [\boldsymbol{V}_1 \;\; \boldsymbol{V}_2]^T \\ \hat{\boldsymbol{A}} &= \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^T \end{aligned} \tag{2.10}$$

where $(\Sigma)_1 \in \mathbb{R}^{k \times k}$, will be the best rank-$k$ minimizer of the problem[18]

$$\arg\min_{\hat{\boldsymbol{A}}} \quad \left\| \boldsymbol{A} - \hat{\boldsymbol{A}} \right\|. \tag{2.11}$$

Another main application for the SVD lies in computing the generalized inverse of singular matrices. When viewed from a geometric perspective, a singular transformation matrix $\boldsymbol{A}$ collapses one or multiple of the dimensions of $\boldsymbol{x}$ when acting on it. A proper matrix inverse can not exist, since the information of this dimension is mapped to $\mathbf{0}$. A way to still construct a "weaker" form of inverse matrix called Moore-Penrose **pseudoinverse** $\boldsymbol{A}^+$ is given by

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \Rightarrow \boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T \tag{2.12}$$

where $\boldsymbol{\Sigma}^+$ denotes the reciprocal of the diagonal matrix $\boldsymbol{\Sigma}$ containing the singular values. This pseudoinverse can be shown to possess properties close to a proper inverse[14].

2.4 QR Decomposition

The QR Decomposition of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ is the factorization into two component matrices $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ such that

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R} \tag{2.13}$$

where $\boldsymbol{R}$ is upper triangular. If $m = n$ and $\boldsymbol{A} \in \mathbb{R}$ then $\boldsymbol{Q}$ is orthogonal ($\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}^{-1}\boldsymbol{Q} = \mathbb{I}$). In the complex case $\boldsymbol{Q}$ is unitary. The columns of $\boldsymbol{Q}$ form an orthonormal basis for the geometrical space spanned by $\boldsymbol{A}$. If we require that $\boldsymbol{R}_{ii} \geq 0$ the decomposition is unique. A detailed description of the properties and uses of the QR Decomposition can be found in Goodall, C. R. [19]. In this chapter we will mainly focus on the case of real, symmetric matrices.

There are many algorithms that compute the QR Decomposition. They offer different tradeoffs in terms of efficiency and numeric stability. We will now outline some of them.

Strang, G. [14] shows how to derive the QR decomposition from the Gram-Schmidt procedure. In algorithm given some vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ that span a space in $\mathbb{R}^3$, we compute the vectors $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ [F.3] such that they are orthonormal

$$\begin{aligned} \|\boldsymbol{q}_i\| &= 1 \\ \boldsymbol{q}_i^T \boldsymbol{q}_j &= 0. \end{aligned} \tag{2.14}$$

This can be achieved by sequentially projecting the input vectors into the space spanned by the already computed orthonormal vectors and normalizing them

[F.3] This procedure also extends to problems with more than three vectors.

$$
\begin{aligned}
\boldsymbol{q}_1 &= \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \\
\boldsymbol{q}_2 &= \frac{\boldsymbol{b} - (\boldsymbol{q}_1^T \boldsymbol{b})\boldsymbol{q}_1}{\left\|\boldsymbol{b} - (\boldsymbol{q}_1^T \boldsymbol{b})\boldsymbol{q}_1\right\|} \\
\boldsymbol{q}_3 &= \frac{\boldsymbol{c} - (\boldsymbol{q}_1^T \boldsymbol{c})\boldsymbol{q}_1 - (\boldsymbol{q}_2^T \boldsymbol{c})\boldsymbol{q}_2}{\left\|\boldsymbol{c} - (\boldsymbol{q}_1^T \boldsymbol{c})\boldsymbol{q}_1 - (\boldsymbol{q}_2^T \boldsymbol{c})\boldsymbol{q}_2\right\|}.
\end{aligned}
\tag{2.15}
$$

The matrix can then be factored into

$$
\boldsymbol{A} = [\boldsymbol{a} \mid \boldsymbol{b} \mid \boldsymbol{c}] = [\boldsymbol{q}_1 \mid \boldsymbol{q}_2 \mid \boldsymbol{q}_3] \begin{bmatrix} \boldsymbol{q}_1^T \boldsymbol{a} & \boldsymbol{q}_1^T \boldsymbol{b} & \boldsymbol{q}_1^T \boldsymbol{c} \\ & \boldsymbol{q}_2^T \boldsymbol{b} & \boldsymbol{q}_2^T \boldsymbol{c} \\ & & \boldsymbol{q}_3^T \boldsymbol{c} \end{bmatrix} = \boldsymbol{Q}\boldsymbol{R}. \tag{2.16}
$$

Although this process is easy to understand and implement, it is numerically unstable due to rounding errors that accumulate quickly in the process of calculating the $\boldsymbol{q}$ vectors. This issue can be mitigated by modifying the Gram-Schmidt procedure, or by using another construction method like Givens rotations or Householder transformations[19,20].

The Householder transformation of a given vector $\boldsymbol{v}$ can be represented by a matrix

$$
\boldsymbol{H} = \mathbb{I} - 2\frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}} \tag{2.17}
$$

that if multiplied with a vector $\boldsymbol{x}$ can be geometrically interpreted as reflecting $\boldsymbol{x}$ over the hyperplane spanned by $\boldsymbol{v}$. Since a reflection can be applied twice to yield the original vector the Householder matrix is orthogonal.

To explain the Householder QR algorithm we define $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and an orthogonal $\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ such that

$$
\begin{aligned}
\boldsymbol{A} &= [\boldsymbol{a}_1 \mid \boldsymbol{a}_2 \mid \ ... \ \mid \boldsymbol{a}_n] \\
\boldsymbol{Q} &= [\boldsymbol{e}_1 \mid \boldsymbol{e}_2 \mid \ ... \ \mid \boldsymbol{e}_n].
\end{aligned}
\tag{2.18}
$$

The strategy is to use Householder transformations to reflect the column $\boldsymbol{a}_1$ onto the orthogonal columns $\boldsymbol{e}_1$, which will produce a vector of the form $\boldsymbol{a'}_1 = [a'_{11} \ 0 \ 0 \ \dots \ 0]^T$. To calculate this projection, we can use geometric reasoning and find the formula for the $\boldsymbol{v}$ vector of our Householder transformation

$$
\begin{aligned}
\boldsymbol{a}'_1 &= (\mathbb{I} - 2\boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{a} \\
\boldsymbol{v} &= -\frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} \\
\boldsymbol{b} &= \boldsymbol{a} - (-\text{sign}(\boldsymbol{a}_{11}))\|a\|\boldsymbol{e}_1
\end{aligned}
\tag{2.19}
$$

which lets us compute the first iteration Householder matrix

$$
\boldsymbol{Q}^{(1)} = \mathbb{I} - 2\boldsymbol{v}\boldsymbol{v}^T. \tag{2.20}
$$

Applying this orthogonal matrix on $\boldsymbol{A}$ yields

$$\boldsymbol{Q}^{(1)}\boldsymbol{A} = \boldsymbol{A}^{(1)} = \left[\begin{array}{c|ccc} a'_{11} & a'_{12} & \dots & a'_{1i} \\ \hline 0 & a'_{22} & \dots & a'_{2i} \\ 0 & \vdots & & \vdots \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{a}'_{11} & \boldsymbol{A}'_{12} \\ \hline 0 & \boldsymbol{A}'_{22} \end{array}\right]. \tag{2.21}$$

We now iterate this scheme with the submatrix $\boldsymbol{A}'_{22}$, which results in another smaller Householder matrix $\boldsymbol{Q}^{(2)}$ which we need to bring into the same form as $\boldsymbol{Q}^{(1)}$ by "embedding" it into the identity matrix

$$\boldsymbol{Q}'^{(2)} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \boldsymbol{Q}'^{(2)} \end{array}\right]. \tag{2.22}$$

We now embed the resulting matrix $\boldsymbol{A}^{(2)}$ into the original $\boldsymbol{A}$. This scheme is iterated until all entries in $\boldsymbol{A}$ are processed, resulting in

$$\boldsymbol{A} = \underbrace{\left(\boldsymbol{Q}^{(1)}\boldsymbol{Q}'^{(2)}...\boldsymbol{Q}'^{(n)}\right)}_{\boldsymbol{Q}}\underbrace{\boldsymbol{A}^{(n)}}_{\boldsymbol{R}} \tag{2.23}$$

where all $\boldsymbol{Q}^{(n)}$ are Householder matrices and $\boldsymbol{A}^{(n)}$ is upper triangular and the columns in $\boldsymbol{A}$ were reflected on the orthogonal $\boldsymbol{e}$ columns. This process is numerically more stable than the Gram-Schmidt procedure since the only repeated action is the application of the Householder orthogonal transformation. This procedure is implemented by standard software libraries like LAPACK[21] and can be computed in $\approx mn^2$ operations[19,22].

Despite its simplicity, the QR Decomposition is a widely used tool in linear algebra. The QR Decomposition is useful especially for solving Least Squares (LS) problems, since many sub-problems that involve pseudoinversion can be computed more efficiently and in a numerically stable fashion using the triangular and orthogonal $\boldsymbol{R}$ and $\boldsymbol{Q}$ matrices[19].

A historically important application for the QR decomposition is computing the eigenvalues of a dense matrix. First devised by Francis, J. G. F. [23], the QR eigenvalue algorithm exploits the fact that the matrix $\boldsymbol{R}$ computed by the QR decomposition is similar to the matrix $\boldsymbol{A}$ as the factored out $\boldsymbol{Q}$ is orthogonal and therefore purely "rotational". Given a real matrix $\boldsymbol{A}^{(1)}$, the algorithm computes the QR decomposition

$$\boldsymbol{A}^{(1)} = \boldsymbol{Q}^{(0)}\boldsymbol{R}^{(0)}. \tag{2.24}$$

Next, we can reverse the order of the products to reconstruct a formula similar to the Schur decomposition defined in Eq. (2.7)

$$\boldsymbol{A}^{(1)} = \boldsymbol{Q}^{(0)}\boldsymbol{R}^{(0)} = \boldsymbol{Q}^{(0)T}\boldsymbol{A}^{(0)}\boldsymbol{Q}^{(0)}. \tag{2.25}$$

If this process is repeated with the newly constructed $\boldsymbol{A}^{(1)}$ matrix

$$\boldsymbol{A}^{(k+1)} = \underbrace{\boldsymbol{Q}^{(k)}...\boldsymbol{Q}^{(0)}}_{\boldsymbol{U}}\boldsymbol{A}^{(k)}\underbrace{\boldsymbol{Q}^{(0)T}...\boldsymbol{Q}^{(k)T}}_{\boldsymbol{U}^T} \tag{2.26}$$

it can be shown that under a set of assumptions $\boldsymbol{A}$ and $\boldsymbol{U}$ will converge to the Schur decomposition. This algorithm is simple to implement and very elegant, but seldomly used in practice since a costly QR decomposition is needed for each iteration. Instead, in the stable implementation provided by LAPACK[21], the matrix is first reduced to a tridiagonal form, i.e., only some diagonals alongside the main diagonal contain values. This is done by a sequence of Householder transformations. From this tridiagonal matrix the eigenvalues and eigenvectors can be computed with fewer computations (but

still in $\mathcal{O}(n^3)$ formal complexity) using the Divide and Conquer algorithm[24] devised by Cuppen, J. J. M. [25].

In the case that $\boldsymbol{A}$ is rank-deficient or close to it, the QR decomposition can be used in its "pivoted" form to gain insight about the numeric rank of $\boldsymbol{A}$ by assessing the amount of singular values in $\boldsymbol{A}$ greater than a tolerance value ε. This is crucial to construct low-rank approximations of $\boldsymbol{A}$. The pivoted form of the QR decomposition is written as

$$\boldsymbol{A\Pi} = \boldsymbol{QR} \tag{2.27}$$

where $\boldsymbol{\Pi}$ is a permutation matrix. One of most widely used ways to find the correct permutation $\boldsymbol{\Pi}$ was suggested by Businger, P.; Golub, G. H. [26]. In their variant of the Householder QR algorithm, they identify the column in the submatrix $\boldsymbol{A}'_{(m)}$ from Eq. (2.21) which would maximize the new diagonal element in $\boldsymbol{A}_{(m)}$ and move that column to the front of the submatrix. This can be done by searching for the remaining column with the biggest euclidean norm, which unfortunately leads to high computational complexity and high communication cost on large matrices[27]. In this way, we can construct a low-rank representation of $\boldsymbol{A}$, by choosing an ε and partitioning the result of the QR decomposition such that

$$\boldsymbol{A\Pi} = [\boldsymbol{Q}_1 \;\; \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ 0 & \boldsymbol{R}_{22} \end{bmatrix} \quad [\boldsymbol{R}_{22}]_{11} < \varepsilon. \tag{2.28}$$

Since the pivoting moved all linearly independent columns to the upper left quadrant, an approximation of $\boldsymbol{A}$ can be constructed using

$$\boldsymbol{A} \approx \boldsymbol{Q}_1 [\boldsymbol{R}_{11} \;\; \boldsymbol{R}_{12}] \boldsymbol{\Pi}^T. \tag{2.29}$$

Although this version of QR with Column Pivoting (QRCP) is widely used and numerically stable, its added n^2 operations incurred in the column pivoting makes the approach too costly for most big matrices[27]. Therefore, many publications in the last decades have focused on improving this process in many ways.

More recent publications are concerned with using randomized projections to reduce the row-count of tall matrices, thus speeding up the pivot selection[27,28]. Others made the algorithm better suited for optimization by implementing them with block-wise operations[29]. Some explored distributing the computation on multi-node systems, providing means for higher parallelization and better usage of computer memory[30] or employed low-level optimization on modern CPUs to speed up calculations[31].

2.5 Cholesky Decomposition

The Cholesky Decomposition (CD) is a powerful technique to represent a square, symmetric, positive definite (i.e., with strictly positive eigenvalues) matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ as

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T \tag{2.30}$$

where $\boldsymbol{L} \in \mathbb{R}^{m \times m}$ is a triangular matrix. For complex matrices, instead of transposing $\boldsymbol{L}$, the conjugate transpose $\boldsymbol{L}^\dagger$ is used. Analogous to the QR Decomposition, the CD can be computed by a simple recursive scheme on $\boldsymbol{A}$.[14]

$\boldsymbol{A}$ can be partitioned into quadrants

$$\boldsymbol{A} = \left[\begin{array}{c|c} a_{11} & \star \\ \hline \boldsymbol{a}_{21} & \boldsymbol{A}_{22} \end{array}\right] = \left[\begin{array}{c|c} \lambda_{11} & 0 \\ \hline \boldsymbol{l}_{21} & \boldsymbol{A}_{22} \end{array}\right] \left[\begin{array}{c|c} \lambda_{11} & 0 \\ \hline \boldsymbol{l}_{21} & \boldsymbol{A}_{22} \end{array}\right]^T = \left[\begin{array}{c|c} \lambda_{11}^2 & \star \\ \hline \lambda_{11}\boldsymbol{l}_{21} & \boldsymbol{l}_{21}\boldsymbol{l}_{21}^T + \boldsymbol{L}_{22}\boldsymbol{L}_{22}^T \end{array}\right] \tag{2.31}$$

where the $\star$ symbol marks a submatrix that we do not have to calculate. From this description, we can derive expressions for $\boldsymbol{L}$ in terms of $\boldsymbol{A}$

$$\begin{aligned} \boldsymbol{A} &= \left[\begin{array}{c|c} a_{11} = \lambda_{11}^2 & \star \\ \hline \boldsymbol{a}_{21} = \lambda_{11}\boldsymbol{l}_{21} & \boldsymbol{A}_{22} = \boldsymbol{l}_{21}\boldsymbol{l}_{21}^T + \boldsymbol{L}_{22}\boldsymbol{L}_{22}^T \end{array}\right] \\ \Rightarrow \boldsymbol{L} &= \left[\begin{array}{c|c} \lambda_{11} = \sqrt{a_{11}} & \star \\ \hline \boldsymbol{l}_{21} = \frac{\boldsymbol{a}_{21}}{\lambda_{11}} & \boldsymbol{L}_{22} = \mathrm{Chol}(\boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{l}_{21}^T) \end{array}\right]. \end{aligned} \tag{2.32}$$

From this expression it becomes clear that the diagonal elements of $\boldsymbol{A}$ have to be positive as we have to compute $\lambda_{11} = \sqrt{a_{11}}$. This is true for positive definite matrices, by Sylvester's law of inertia[14]. Also Eq. (2.32) already reveals that the CD can offer computational benefits over the QR Decomposition: The QR algorithm needs to compute a $(n-k+1)^2$ Householder matrix in the kth iteration and apply it to the $\boldsymbol{A}'_{(k)}$ lower-right submatrix, giving

$$\boldsymbol{A}'_{(k+1)} = (\mathbb{I} - 2\boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{A}'_{(k)} = \boldsymbol{A}'_{(k)} - 2\boldsymbol{v}\left(\boldsymbol{v}^T\boldsymbol{A}'_{(k)}\right) = \boldsymbol{A}'_{(k)} - 2\boldsymbol{v}\boldsymbol{w}^T \tag{2.33}$$

which leads to a similar update as the $\boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{l}_{21}^T$ update in the Cholesky algorithm, but involves one additional matrix-vector product.

To understand the usefulness of the CD in computing low-rank approximations, we need to consider the case when $\boldsymbol{A}$ is only positive-semidefinite. We can encounter diagonal entries $a_{ii} = 0$, which would lead to division by zero in the computation of the off-diagonal elements $\frac{\boldsymbol{a}_{ji}}{\sqrt{a_{ii}}}$. Since this computation is not possible, we need to introduce a column pivoting similar to the pivoted QR algorithm

$$\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{\Pi}^T = \boldsymbol{L}\boldsymbol{L}^T. \tag{2.34}$$

As many applications in science involve symmetric interactions, which can be close to zero (e.g., gravitational forces between bodies, the coulomb interaction of particles etc.) the pivoted CD is useful in many applications in the natural sciences. If we assume that $\boldsymbol{A}$ has at least one diagonal entry $a_{ii} > 0$, we can modify the algorithm in Eq. (2.32) such that in each iteration, we use the maximum element of $\mathrm{diag}(\boldsymbol{A}_{22})$ as a pivot and swap it to the front. As outlined by Harbrecht, H.; Peters, M.; Schneider, R. [32], this greedy-approach produces a rank-k approximation $\hat{\boldsymbol{A}}$ in $k^2 \cdot n$ operations where k is the numerical rank of $\hat{\boldsymbol{A}}$. The procedure is outlined in Algorithm 1, which performs the pivoted cholesky algorithm row-by-row while using the trace of the residual $\boldsymbol{A}_{22}$ as measure of error. This allows the user to define a rigorous error bound by defining one parameter.

Algorithm 1 also reveals the second computational advantage of the CD over QRCP: In addition to the diagonal, we only ever need to access one row of $\boldsymbol{A}$ per iteration. Both take n operations to compute. If we assume that our numerical rank $k \ll n$ and that we can calculate rows of $\boldsymbol{A}$ on the fly, the algorithm only uses $(n \cdot k)$ memory instead of the full n^2 that would be needed by the naive implementation.

The pivoted Cholesky Decomposition has found recent adoptions in the field of machine learning, where practitioners often employ so-called kernel methods. A kernel defines a metric or similarity measure on a given set of data point's and can be used to approach interpolation, regression and clustering problems. As the kernel matrix describes a distance measure, it must be symmetric and

Algorithm 1: Row-wise PivotedCholesky with trace error minimization

procedure PivotedCholesky($\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\varepsilon > 0$)
  $\boldsymbol{d} \leftarrow \mathrm{diag}(\boldsymbol{A})$
  $\boldsymbol{F} \leftarrow \boldsymbol{0} \in \mathbb{R}^{n \times n}$
  $\mathrm{err} \leftarrow \sum |\boldsymbol{d}|$
  $\boldsymbol{\pi} \leftarrow (1, 2, ..., N)$
  $m \leftarrow 1$
  while $(m \leq n) \wedge (\varepsilon \leq \mathrm{err})$ **do**
    $s_i \leftarrow$ select a pivot from $\boldsymbol{d}_{\boldsymbol{\pi}(\boldsymbol{m}:\boldsymbol{n})}$ ▷ $\pi(m)$ denotes π_m
    $\pi_m, \pi_i \leftarrow \pi_i, \pi_m$ ▷ Swap the new pivot to the cursor
    $\boldsymbol{F}_{m,\pi(m)} \leftarrow \sqrt{s_i}$ ▷ Calculate the diagonal element
    $\boldsymbol{c} \leftarrow \pi(m+1:n)$ ▷ The remaining columns
    $\boldsymbol{F}_{m,\boldsymbol{c}} \leftarrow \left[\boldsymbol{A}_{\pi(m),\boldsymbol{c}} - \boldsymbol{F}_{:m,\pi(m)} \boldsymbol{F}^T_{:m,\boldsymbol{c}}\right] / \boldsymbol{F}_{m,\pi(m)}$ ▷ Update $\boldsymbol{F}$ with the cholesky vectors

    $\boldsymbol{d} \leftarrow \boldsymbol{d} - \left[\boldsymbol{F}_{m,\boldsymbol{c}}\right]^2$ ▷ Correct remaining diagonal
    $\mathrm{err} \leftarrow \sum \boldsymbol{d}_{\boldsymbol{c}}$ ▷ Update error with residual trace
  end

  return $\boldsymbol{F}, \boldsymbol{\pi}$
end

positive-semidefinite. Since computing those matrices also scales quadratically with the size of the dataset, finding low-rank substitutes is highly desirable. Therefore, a multitude of improvements and optimizations for the CD algorithm was suggested by researchers[33–36].

A topic that has often been discussed in the literature is the pivot selection mechanism. The mentioned greedy approach makes the assumption that the diagonal size is a good indicator for whether the column will substantially minimize the trace error. This is not necessarily true for all matrices, which led to the suggestion of uniform random sampling on the diagonal[37]. It was shown by Chen, Y.; Epperly, E. N.; Tropp, J. A.; Webber, R. J. [33] that the uniform sampling and the greedy approach are different ends of a spectrum. If we think about pivot finding as sampling the diagonal using a Gibbs-distribution with differing temperature parameters, the greedy approach represents zero temperature while uniform sampling would have infinite temperature. The researchers developed the Randomly Sampled Cholesky (RPCholesky) algorithm that represents a middle ground by drawing the new pivot from a probability distribution over the current residual's diagonal

$$\mathbb{P}\{s_i = j\} = \frac{\boldsymbol{A}^{(i-1)}_{j,j}}{\mathrm{tr}\, \boldsymbol{A}^{(i-1)}}. \tag{2.35}$$

This new pivoting strategy is easy to integrate into Algorithm 1. The probability distribution can be computed like from $\mathbb{P} = \boldsymbol{d}_{\pi(m:)} / \left(\sum \boldsymbol{d}_{\pi(m:)}\right)$. For drawing a pivot, we can then employ any random sampling function from the numerical computation library in use. RPCholesky was shown to accelerate a multitude of kernel methods and to converge with almost optimal speed. The authors also provide the outline for a blocked version of the algorithm, which can be used to

maximize CPU utilization. By sampling a set of pivots (size b) and using the full block to project out the associated columns.

Another improvement idea was put forward by Epperly, E. N.; Tropp, J. A.; Webber, R. J. [36]. The researchers observed that on certain datasets the blocked RPCholesky can still fail to minimize the trace error fast enough, which leads to a bigger k than could be achieved with standard RPCholesky. Based on this finding, the authors propose to add another sampling step on a selected pivot block, where all pivots in a block will be assigned a lower acceptance probability if they would be redundant after application of the already accepted pivots. This algorithm was named accelerated RPCholesky.

2.6 Interpolative Decomposition

The Interpolative Decomposition (ID) is yet another way to decompose a matrix $\boldsymbol{A}$ into two matrices that offers some unique benefits when compared with the previously discussed approaches. The goal in the ID is to represent $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ as

$$\boldsymbol{A} = \boldsymbol{C}\boldsymbol{Z} \tag{2.36}$$

where $\boldsymbol{C} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{Z} \in \mathbb{R}^{k \times n}$. The following additional restrictions have to hold:

- All k columns of $\boldsymbol{C}$ are a subset of $\boldsymbol{A}$.
- Some $k \times k$ subset of $\boldsymbol{Z}$'s columns forms the identity $\mathbb{I} \in \mathbb{R}^{k \times k}$.
- All absolute values in $\boldsymbol{Z}$ are smaller than 2: $\left|\boldsymbol{Z}_{ij}\right| < 2$.

The problem can also be formulated row-wise, by just considering $\boldsymbol{A}^T$. To give a less abstract description, consider $\boldsymbol{A} \in \mathbb{R}^{3 \times 3}$ where $\operatorname{rank}(\boldsymbol{A}) = 2$

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix}}_{\boldsymbol{C}} \cdot \underbrace{\begin{bmatrix} 0 & 1 & z_{13} \\ 1 & 0 & z_{33} \end{bmatrix}}_{\boldsymbol{Z}}. \tag{2.37}$$

In comparison to the other method discussed in this thesis, the ID offers the benefit that we can reuse columns of our matrix $\boldsymbol{A}$ to form $\boldsymbol{C}$ and only need to store the required column indices as well as the non-identity parts of $\boldsymbol{Z}$. The matrices $\boldsymbol{C}$ and $\boldsymbol{Z}$ are often called a "skeleton" and interpolation matrix respectively[38,39].

The ID is often computed using a QRCP. This decomposition enables computing the $\boldsymbol{Z}$ matrix using least-squares minimization. The derivation as provided in the literature[39] starts by stating the following minimization in the Frobenius norm

$$\arg\min_{\boldsymbol{Z}} \quad \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{Z}\|_F. \tag{2.38}$$

An analytic solution is given by

$$\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{Z} = \boldsymbol{C}^T\boldsymbol{A} \tag{2.39}$$

which can be simplified using the QR decomposition of the psd matrix $\boldsymbol{C}^T\boldsymbol{C}$

$$\begin{aligned} \boldsymbol{R}^T\cancel{\boldsymbol{Q}^T\boldsymbol{Q}}\boldsymbol{R}\boldsymbol{Z} &= \boldsymbol{C}^T\boldsymbol{A} \\ \boldsymbol{R}^T\boldsymbol{R}\boldsymbol{Z} &= \boldsymbol{C}^T\boldsymbol{A}. \end{aligned} \tag{2.40}$$

In the last step, a triangular solver is employed to find a solution for $\boldsymbol{Z}$. The most expensive operation in this algorithm is the QRCP. Therefore, researchers have applied randomization and sampling techniques to the ID. A recent publication by Dong, Y.; Chen, C.; Martinsson, P.-G.; Pearce, K. [38] provides an overview of the landscape and classifies existing algorithms by the way they approach pivoting in the skeleton-selection step. They identify the following categories:

- **Greedy pivoting**: Methods that have no random component like the one using QRCP as described above. While they are rank-adaptive, i.e., they do not require prior knowledge of the matrix rank, they are sequential and therefore computationally infeasible for big inputs.
- **Sampling**: Methods that only use sampling of random distributions (like the squared Euclidean norm) to select columns for the skeleton. In context of ID those approaches can fail to reveal input rank and might lead to lackluster results.
- **Sketchy pivoting**: Other publications[39–41] employed techniques, where the input is first multiplied with a randomized sketching matrix reducing its size. Subsequently, a sequential greedy pivoting can be applied. Those algorithms can be fast on modern hardware, but often require prior knowledge about the inputs numerical rank, which is often itself the subject of investigation.
- **Random pivoting**: Similar to the randomly pivoted CD algorithms from Section 2.5 Dong, Y.; Chen, C.; Martinsson, P.-G.; Pearce, K. [38] develop an algorithm that uses blocked random pivoting to construct the ID. Specifically the algorithm uses a vector of squared column-norms of the residual matrix in a way similar to the diagonal in RPCholesky. From this vector blocks of pivots are sampled based on a probability distribution. Thereafter, the QRCP is used to reject possible redundant columns in the block. Finally, the ID is updated using efficient matrix-matrix operations. The algorithm terminates when it reaches a user defined threshold in the remaining column norms. This process was shown to yield high quality approximations, while automatically adapting to the rank of the input matrix.

2.7 CP Decomposition

The CANDECOMP/PARAFAC Decomposition can be traced back to a publication from 1927 by Hitchcock, F. L. [42]. The CPD is the first tensor factorization described here for tensors of orders larger than two. It is closely related to the SVD and can be viewed as its higher-order version. We can illustrate this by recalling Eq. (2.8) and re-expressing it as a sum of rank-1 contributions

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + ... + \sigma_n \boldsymbol{u}_n \boldsymbol{v}_n^T. \tag{2.41}$$

If we change $\boldsymbol{A}$ to be an order-3 tensor $\boldsymbol{T}_{abc}$, we might arrive at a good representation for it using only the weighted sum of some vector outer-products

$$\boldsymbol{T}_{abc} \approx \lambda_1 \boldsymbol{a}_1 \circ \boldsymbol{b}_1 \circ \boldsymbol{c}_1 + ... + \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \tag{2.42}$$

which, if written using summation notation, is the exact definition of the CPD

$$\boldsymbol{T}_{abc} \approx \sum_{r=1}^{R} \boldsymbol{\lambda}_r \cdot \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r. \tag{2.43}$$

R denotes the rank of the tensor. The rank of a higher order tensor is not as simple as the matrix rank described in Section 2, where we saw that the problem of determining the rank of a tensor (which then is equivalent to finding an optimal CPD) is NP-hard. Therefore, there are already order-3 tensors for which a rank could not be determined yet[9].

There exist multiple algorithms to compute the CPD. The most studied and implemented way uses an Alternating Least Squares (ALS) strategy to solve the non-linear optimization problem. The ALS algorithm starts from an initial state of the factor matrices $\boldsymbol{A}_r$, $\boldsymbol{B}_r$ and $\boldsymbol{C}_r$, assumes all but one factor to be fixed and solves the least-squares minimization problem

$$\arg\min_{\boldsymbol{A}_r} = \left\| \boldsymbol{T}_{(1)} - \boldsymbol{A}_r \circ \boldsymbol{B}_r \circ \boldsymbol{C}_r \right\|^2. \tag{2.44}$$

The weights $\boldsymbol{\lambda}_r$ are contracted into the factor matrices and $\boldsymbol{T}_{(1)}$ denotes the mode-1 unfolding of $\boldsymbol{T}$ defined in Eq. (2.1). It can be shown that under the assumption of a fixed $\boldsymbol{A}_r$ the problem has an analytic solution of the form

$$\hat{\boldsymbol{A}} = \boldsymbol{T}_{(1)} \left[(\boldsymbol{C} \odot \boldsymbol{B})^T \right]^+. \tag{2.45}$$

Here, $+$ denotes the Moore-Penrose pseudoinverse. This expression can be rewritten such that it only involves the pseudoinversion of an $R \times R$ matrix using properties of the Khatri-Rao product defined in Eq. (2.4)

$$\hat{\boldsymbol{A}} = \boldsymbol{T}_{(1)} (\boldsymbol{C} \odot \boldsymbol{B}) (\boldsymbol{C}^T \boldsymbol{C} * \boldsymbol{B}^T \boldsymbol{B})^+. \tag{2.46}$$

After solving the problem for $\boldsymbol{A}$, we fix $\boldsymbol{B}$ and $\boldsymbol{C}$ in turn and repeat the process until a convergence criterion is reached, i.e., the norm difference is sufficiently minimized, or we reach a maximum number of iterations. The full algorithm for order-n tensors is outlined in Algorithm 2.

We can list some of the shortcoming of the standard ALS approach of computing the CPD:

- The algorithm is not guaranteed to converge to a global minimum and is heavily reliant upon the initial guess[9].
- It was shown to be numerically unstable when the input is ill-conditioned[43].
- The algorithm takes the approximate rank R as a parameter, which means that

Algorithm 2: CP-ALS Algorithm

procedure CP-ALS(input tensor $\boldsymbol{X} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$, rank $R \in \mathbb{N}^+$)
  for $n \in$ modes($\boldsymbol{X}$) **do**
    $\boldsymbol{A}^{(n)} \leftarrow$ random matrix $\in \mathbb{R}^{I_n \times R}$
  end

  while fit insufficient **do**
    for $n \in$ modes($\boldsymbol{X}$) **do**
      $\boldsymbol{V} \leftarrow \boldsymbol{A}^{(1)T}\boldsymbol{A}^{(1)} * \ldots * \boldsymbol{A}^{(n-1)T}\boldsymbol{A}^{(n-1)} * \boldsymbol{A}^{(n+1)T}\boldsymbol{A}^{(n+1)} * \ldots * \boldsymbol{A}^{(N)T}\boldsymbol{A}^{(N)}$
      $\boldsymbol{A}^{(n)} \leftarrow \boldsymbol{X}^{(n)}\big(\boldsymbol{A}^{(N)} \odot \ldots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \ldots \odot \boldsymbol{A}^{(1)}\big)\boldsymbol{V}^+$
      $\boldsymbol{\lambda} \leftarrow$ column-norms of $\boldsymbol{A}^{(n)}$
    end
  end

  return $\boldsymbol{\lambda}, \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \ldots, \boldsymbol{A}^{(N)}$
end

in order to reveal the inputs numerical rank the algorithm needs to be run multiple times until a high enough rank ceases to increase the fit. Therefore, the algorithm is not rank-adaptive as defined by Dong, Y.; Chen, C.; Martinsson, P.-G.; Pearce, K. [38].

A multitude of publications exist that improve upon the standard ALS approach or find alternative solutions:

One such approach that primarily focuses on improving numeric stability is the ALS-CP-QR algorithm[44]. It eliminates the need for a pseudoinversion by computing the QR Decomposition of the of each individual factor matrix and the subsequent Khatri-Rao products. It was shown to achieve higher numeric stability on ill-conditioned inputs, while maintaining good performance at least for low-rank tensors. The performance was shown to be improvable for example by introducing a Dimension Tree that helps to reutilize temporary products[43].

In keeping with the theme of the other decomposition methods discussed, researchers[45–47] applied randomization and sampling techniques to the CPD. The general outline of those algorithms is similar to the sketching ideas mentioned in Section 2.6. A randomized sketching matrix is used to transform a large input into a smaller sample that is easier to decompose using the standard ALS method.

Another technique that completely replaces ALS is using Gauss-Newton optimization techniques to directly solve the non-linear least-squares problem of the CPD step. While not changing the formal runtime complexity of the algorithm, a Gauss-Newton implementation using the conjugate gradient method was shown to outperform ALS based CPD in convergence rates as well as quality of the approximation[48].

3 Tensors in Electronic Structure Theory

Quantum mechanics as theorized in the 20th century postulates that matter exhibits wave-like behavior on small scales. In this regime classical particles are described by wave functions on which measurements act through operators. Physically measured quantities (observables) are given by the eigenvalues of their associated operator $\hat{\boldsymbol{O}}$. Since observables have to be physically meaningful, they have to be real numbers. Those operators can therefore be modelled by Hermitian matrices, for which we know that (1) all Eigenvalues are real and (2) all Eigenvectors can be normalized to an orthonormal basis (being perpendicular with length 1). Real Hermitian matrices are just symmetric and for complex cases they are equal to their own conjugate transpose ($\hat{\boldsymbol{O}} = \hat{\boldsymbol{O}}^\dagger$)[14].

The wavefunction of a system can not be observed directly. A measurement result of an operator action on a wavefunction can be written in terms of expectation values manifesting as integrals of the squared amplitude. In Dirac-Notation this can be written as

$$\langle \Psi | v | \Psi \rangle = \left\langle \Psi \middle| \hat{\boldsymbol{O}} \middle| \Psi \right\rangle = \int_{\mathbb{R}^3} \Psi^*(\boldsymbol{x})\ \hat{\boldsymbol{O}}\ \Psi(\boldsymbol{x}) \mathrm{d}^3\boldsymbol{x} \tag{3.1}$$

where the wavefunction can be normalized to Ψ' such that the operators expectation values becomes a simple scalar

$$\langle \Psi' | \Psi' \rangle = \int \Psi'^* \Psi' \, \mathrm{d}r = 1 \quad \Rightarrow \left\langle \Psi' \middle| \hat{\boldsymbol{O}} \middle| \Psi' \right\rangle = v. \tag{3.2}$$

The core problem in much of electronic structure theory lies in finding solutions to the non-relativistic, time-independent Schrödinger equation, which was derived by Erwin Schrödinger from the tenets of Hamiltonian wave-mechanics

$$\widehat{\boldsymbol{H}} | \Psi \rangle = E | \Psi \rangle \tag{3.3}$$

In the Schrödinger equation, $\widehat{\boldsymbol{H}}$ denotes the Hamiltonian operator, E a scalar value and Ψ the wavefunction. The observable belonging to the $\widehat{\boldsymbol{H}}$ operator is the energy of the system. The eigenvalue E denotes an eigenenergy for a certain eigenstate of the system. The time-independent version of the Schrödinger equation is especially important in chemistry, as it is used to describe steady states like the electronic ground state of an atom or molecule.

In the following chapter, we will explore how to construct the molecular electronic Hamiltonian and some approaches to solve it approximately. We will use Dirac-Notation and other fundamental concepts that are explained in the introductory chapters of Szabo, A.; Ostlund, N. S. [1], Jensen, F. [2] and Leach, A. R. [49]. We will use the following notational conventions and identities:

- Ψ and Φ refer to full molecular wavefunctions
- ψ_p refers to Molecular Orbital (MO) wavefunctions where
- p, q, r, s refer to generic MOs
- i, j, k, l refer to occupied orbitals
- a, b, c, d refer to unoccupied (virtual) orbitals
- φ_μ refers to Atomic Orbital (AO) wavefunctions where
- $\mu, \nu, \lambda, \sigma$ will be used to index AOs
- Indices can appear alone: $\left\langle i \middle| \hat{\boldsymbol{O}} \middle| j \right\rangle = \left\langle \psi_i \middle| \hat{\boldsymbol{O}} \middle| \psi_j \right\rangle$
- For entities other than wavefunctions, $i, j, k, l...$ can still be used as dummy indices

The molecular Hamiltonian operator can be defined by a sum of kinetic ($\hat{\boldsymbol{T}}$) and potential energy ($\hat{\boldsymbol{V}}$) contributions from electrons and nuclei

$$\widehat{\boldsymbol{H}} = \hat{\boldsymbol{T}}_n + \hat{\boldsymbol{T}}_e + \hat{\boldsymbol{V}}_{ne} + \hat{\boldsymbol{V}}_{ee} + \hat{\boldsymbol{V}}_{nn}. \tag{3.4}$$

In order to simplify the Hamiltonian the Born-Oppenheimer-Approximation can be used: Since nuclei have a far greater mass than electrons and therefore move on much slower timescales, we can assume their kinetic energy to be zero and their interaction potential to be constant, leading to the electronic Hamiltonian

$$\begin{aligned} \widehat{\boldsymbol{H}}_\mathrm{e} &= \hat{\boldsymbol{T}}_e && + \hat{\boldsymbol{V}}_{ne} && + \hat{\boldsymbol{V}}_{ee} && + \underbrace{\hat{\boldsymbol{V}}_{nn}}_{\text{const}} \\ &= -\frac{1}{2}\sum_i \nabla_i^2 && - \sum_{i,I} \frac{Z_I}{\|\boldsymbol{r}_i - \boldsymbol{r}_I\|} && + \sum_{i<j} \frac{1}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|} && + \sum_{I<J} \frac{Z_I Z_J}{\|\boldsymbol{r}_I - \boldsymbol{r}_J\|} \end{aligned} \tag{3.5}$$

where Z_I denotes the charge of nucleus I, $\boldsymbol{r}$ denotes the position in space of the indexed entity and ∇^2 is the Laplacian operator describing the second partial derivative in all spatial dimensions. The electronic Hamiltonian is presented here in atomic units, which we will use throughout if not explicitly stated otherwise. $\widehat{\boldsymbol{H}}_e$ is a multi-particle operator, for it contains the electron-electron repulsion term.

In multielectron systems the electronic wave function must be modelled to obey the Pauli exclusion principle. The principle states that two fermionic wavefunctions must be antisymmetric under exchange and that two fermions can never occupy the same spin-orbital. This can be achieved by using a matrix determinant to describe the wavefunction. For a three electron system the so-called "Slater determinant" can be expressed as

$$\Psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \frac{1}{\sqrt{3!}} \begin{vmatrix} \psi_1(\boldsymbol{x}_1) & \psi_2(\boldsymbol{x}_1) & \psi_3(\boldsymbol{x}_1) \\ \psi_1(\boldsymbol{x}_2) & \psi_2(\boldsymbol{x}_2) & \psi_3(\boldsymbol{x}_2) \\ \psi_1(\boldsymbol{x}_3) & \psi_2(\boldsymbol{x}_3) & \psi_3(\boldsymbol{x}_3) \end{vmatrix}. \tag{3.6}$$

The mathematical properties of the determinant ensure that under exchange of coordinates (i.e., swapping two rows) the wavefunction changes its sign

$$\Psi(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = -\Psi(\boldsymbol{x}_2, \boldsymbol{x}_1, \boldsymbol{x}_3) \tag{3.7}$$

and becomes zero if two electrons occupy the same spin orbital. ψ_i denote the MO wavefunctions which are usually not known. The Linear Combination of Atomic Orbitals (LCAO) approach can be employed to model the MOs in a given Slater determinant. In LCAO the MO is modelled by a weighted sum of AO wavefunctions

$$\psi = \sum_{\mu=1}^{N_\mathrm{basis}} c_\mu \varphi_\mu. \tag{3.8}$$

Since the variational principle states that any parametrized LCAO wavefunction will yield a higher or equal energy compared to the real-system, we can vary the AO coefficients in a way that minimizes the energy and be certain that the resulting orbitals will approach the real solution. The LCAO approach is fundamental since it allows us to discretize the given problem of continuous functions and operators. The building blocks of this discretization are called *basis functions* which are arranged in a *basis set*. Many function classes can be used to model the AOs (e.g., Exponential, Gaussian, polynomial plane). Since this thesis is only concerned with molecular systems (as

opposed to periodic systems) we will employ Gaussian functions contracted in a way that provides a sharp peak at the origin to model AOs.

3.1 Hartree-Fock Theory

The "ansatz" (i.e., the fundamental mathematical assumption) in Hartree-Fock Theory is that the wavefunction can be modeled by a single Slater determinant. Using this ansatz HF theory provides a way to determine the energy of this determinant in the context of the Born-Oppenheimer approximation and using the LCAO approach. This can be achieved by introducing a new effective one-electron operator that can be used to approximate the many-particle electronic Hamiltonian. The first step as presented by Jensen, F. [2] is to split the Hamiltonian into one- and two-particle contributions

$$\widehat{\boldsymbol{H}}_e = \underbrace{-\frac{1}{2}\sum_i \nabla_i^2 \quad -\sum_{i,I}\frac{Z_I}{\|\boldsymbol{r}_i - \boldsymbol{r}_I\|}}_{\hat{\boldsymbol{h}}} \quad \underbrace{+\sum_{i<j}\frac{1}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}}_{\hat{\boldsymbol{g}}} \quad + V_{nn}. \tag{3.9}$$

When letting this operator act on a wavefunction and using the orthogonality properties of the MOs, the output of $\hat{\boldsymbol{g}}_{ij}$ can be split into the Coulomb $\boldsymbol{J}_{ij}$ and Exchange $\boldsymbol{K}_{ij}$ integrals

$$\begin{aligned} E &= \left\langle \Psi \middle| \widehat{\boldsymbol{H}}_e \middle| \Psi \right\rangle \\ &= \left\langle \Psi \middle| \hat{\boldsymbol{h}} \middle| \Psi \right\rangle + \langle \Psi | \hat{\boldsymbol{g}} | \Psi \rangle + \left\langle \Psi \middle| \hat{\boldsymbol{V}}_{nn} \middle| \Psi \right\rangle \\ &= \sum_i^{N_{\text{elec}}} \boldsymbol{h}_i + \frac{1}{2}\sum_{ij}^{N_{\text{elec}}} (\boldsymbol{J}_{ij} - \boldsymbol{K}_{ij}) + V_{nn} \end{aligned} \tag{3.10}$$

where we multiply the Coulomb term by $\frac{1}{2}$ to correct for the symmetric over-counting of repulsion contributions. This energy expression can now be minimized using Lagrangian multipliers under the constraint of maintaining orthonormal MOs. This process yields a new **one-particle operator**, called the Fock operator $\hat{\boldsymbol{f}}$

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{h}} + \sum_j^{N_{\text{elec}}/2} \left(\hat{\boldsymbol{J}}_j - \widehat{\boldsymbol{K}}_j \right). \tag{3.11}$$

This leads to the physical interpretation that the eigenfunctions of the Fock operator are the orbitals that render the total energy stationary. From these we choose the $\frac{N_{\text{elec}}}{2}$ functions with the lowest eigenvalue (i.e., the lowest energy) to construct the ground state determinant. This means, we need to find solutions to the eigenvalue problem

$$\hat{\boldsymbol{f}}\psi_i' = \varepsilon_i \psi_i' \tag{3.12}$$

where $\psi_{i'}$ denotes so-called canonical orbitals, on which a unitary transformation $\boldsymbol{U}$ was applied such that we can find ε_i by diagonalizing the matrix representation of the Fock operator

$$\boldsymbol{U}^\dagger \hat{\boldsymbol{f}} \boldsymbol{U}\psi = \boldsymbol{U}^\dagger \boldsymbol{\Sigma} \boldsymbol{U}\psi. \tag{3.13}$$

In order to solve the equation, the other occupied orbitals must be known. Therefore, we must use an approach that works from an initial guess of the orbitals that iteratively solves the Fock operator for each orbital in turn until a convergence criterion is reached. This approach is called the Self Consistent Field (SCF) algorithm.

When applied to the LCAO functions from Eq. (3.8) the problem can be expressed as a series of matrix operations which form the Roothaan-Hall equations

$$\begin{aligned} \boldsymbol{FC} &= \boldsymbol{SC}\boldsymbol{\varepsilon} \\ \boldsymbol{F}_{\mu\nu} &= \left\langle \varphi_\mu \middle| \hat{\boldsymbol{f}} \middle| \varphi_\nu \right\rangle \\ \boldsymbol{S}_{\mu\nu} &= \left\langle \varphi_\mu | \varphi_\nu \right\rangle \end{aligned} \tag{3.14}$$

where $\boldsymbol{C}$ is the matrix of weighting factors for the AO functions and $\boldsymbol{\varepsilon}$ is a diagonal matrix containing the energies. The elements of $\boldsymbol{F}$ can be expanded into

$$\begin{aligned} \left\langle \varphi_\mu \middle| \hat{\boldsymbol{f}} \middle| \varphi_\nu \right\rangle &= \left\langle \mu \middle| \hat{\boldsymbol{h}} \middle| \nu \right\rangle + \sum_k^{N_{\text{occ}}} [\langle \mu k | \hat{\boldsymbol{g}} | \nu k \rangle - \langle \mu k | \hat{\boldsymbol{g}} | k \nu \rangle] \\ &= \left\langle \mu \middle| \hat{\boldsymbol{h}} \middle| \nu \right\rangle + \sum_k^{N_{\text{occ}}} \sum_{\lambda\sigma}^{N_{\text{basis}}} c_{k\lambda} c_{k\sigma} [\langle \mu\lambda | \hat{\boldsymbol{g}} | \nu\sigma \rangle - \langle \mu\lambda | \hat{\boldsymbol{g}} | \sigma\nu \rangle] \\ &= \left\langle \mu \middle| \hat{\boldsymbol{h}} \middle| \nu \right\rangle + \sum_{\lambda\sigma}^{N_{\text{basis}}} \mathbf{P}_{\lambda\sigma} \langle \mu\nu \| \lambda\sigma \rangle \end{aligned} \tag{3.15}$$

where $\mathbf{P}$ denotes the density matrix, which is the only variable part left in the equation. Throughout this work we will write the Electron Repulsion Integral in chemists notation or Mulliken notation as

$$\boldsymbol{G}_{\mu\nu\lambda\sigma} = (\mu\nu|\lambda\sigma) = \iint \varphi_\mu^*(\boldsymbol{x}_1)\varphi_\nu(\boldsymbol{x}_1) \frac{1}{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|} \varphi_\lambda^*(\boldsymbol{x}_2)\varphi_\sigma(\boldsymbol{x}_2)\, \mathrm{d}\boldsymbol{x}_1\, \mathrm{d}\boldsymbol{x}_2. \tag{3.16}$$

and the antisymmetrized version of the ERI in physicists notation as

$$\langle \mu\lambda \| \nu\sigma \rangle = (\mu\nu|\lambda\sigma) - (\mu\sigma|\lambda\nu) \tag{3.17}$$

in the AO basis and

$$\langle ij \| ab \rangle = (ia|jb) - (ib|ja) \tag{3.18}$$

in the MO basis respectively. The expression in Eq. (3.15) can be written as a tensor contraction between the ERI and the density matrix

$$\boldsymbol{F}_{\mu\nu} = \boldsymbol{h}_{\mu\nu} + \mathbf{P}_{\lambda\sigma}\boldsymbol{G}_{\mu\nu\lambda\sigma} - \mathbf{P}_{\lambda\sigma}\boldsymbol{G}_{\mu\lambda\nu\sigma}. \tag{3.19}$$

These contractions are the computationally most expensive parts in the SCF algorithm, needing N_{basis}^4 operations per iteration, where N_{basis} denotes the number of parametrizable basis functions φ_μ.

The theme of costly contractions of many electron integrals like the ERI is common in quantum mechanical calculations and often limits the applicability of ab-initio methods to small systems and/or small-basis sets. Therefore, many researchers are interested in using mathematical tools like the ones presented in Section 2 to improve upon this situation. We will discuss some approaches in Section 4 after presenting some other relevant ab-initio calculations suffering from the same problem.

3.2 Møller-Plesset Perturbation Theory

In 1934 Møller, C.; Plesset, M. S. [50] derived a systematic way to add electron correlation effects to the uncorrelated HF ansatz. The derivation is described in many books that cover the foundation

of quantum chemistry[1,2]. We start by partitioning the HF Hamiltonian $\widehat{\boldsymbol{H}}$ into an unperturbed part and a perturbation (or fluctuation potential)

$$\widehat{\boldsymbol{H}} = \widehat{\boldsymbol{H}}_0 + \boldsymbol{\mathcal{V}}. \tag{3.20}$$

Here $\widehat{\boldsymbol{H}}_0$ is just the sum of the one-electron Fock operators $\hat{\boldsymbol{f}}$ for each electron k

$$\widehat{\boldsymbol{H}}_0 = \sum_k \hat{\boldsymbol{f}}(k). \tag{3.21}$$

We can derive an expression for the difference of the HF mean field Hamiltonian and the "real" many electron operator

$$\begin{aligned}
\langle \Psi_0 | \boldsymbol{\mathcal{V}} | \Psi_0 \rangle &= \left\langle \Psi_0 \middle| \widehat{\boldsymbol{H}} - \widehat{\boldsymbol{H}}_0 \middle| \Psi_0 \right\rangle \\
&= \left\langle \Psi_0 \middle| \sum_i^{N_\text{elec}} \left[\cancel{\hat{\boldsymbol{h}}}_i + \frac{1}{2} \sum_{ij}^{N_\text{elec}} \frac{1}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|} \right] - \sum_i^{N_\text{elec}} \left[\cancel{\hat{\boldsymbol{h}}}_i + \sum_j^{N_\text{elec}} \left(\hat{\boldsymbol{J}}_j - \widehat{\boldsymbol{K}}_j \right) \right] \middle| \Psi_0 \right\rangle \\
&= \frac{1}{2} \sum_{ij}^{N_\text{elec}} (\boldsymbol{J}_{ij} - \boldsymbol{K}_{ij}) - \sum_{ij}^{N_\text{elec}} (\boldsymbol{J}_{ij} - \boldsymbol{K}_{ij}) \\
&= \frac{1}{2} \sum_{ij}^{N_\text{elec}} \langle ij \| ij \rangle - \sum_{ij}^{N_\text{elec}} \langle ij \| ij \rangle = -\frac{1}{2} \sum_{ij}^{N_\text{elec}} \langle ij \| ij \rangle.
\end{aligned} \tag{3.22}$$

which is expressed here in the MO basis. We can see that the first correction to the ground state energy is just the electron repulsion term we already derived when discussing HF theory

$$\begin{aligned}
E_\text{MP1} &= \left\langle \Psi_0 \middle| \widehat{\boldsymbol{H}}_0 \middle| \Psi_0 \right\rangle + \langle \Psi_0 | \boldsymbol{\mathcal{V}} | \Psi_0 \rangle \\
&= \sum_i^{N_\text{elec}} \left\langle \Psi_0 \middle| \hat{\boldsymbol{f}}_i \middle| \Psi_0 \right\rangle - \frac{1}{2} \sum_{ij}^{N_\text{elec}} \langle ij \| ij \rangle \\
&= \sum_i^{N_\text{elec}} \varepsilon_i - \frac{1}{2} \sum_{ij}^{N_\text{elec}} \langle ij \| ij \rangle \\
&= E_0^{(0)} + E_0^{(1)} = E_\text{HF}.
\end{aligned} \tag{3.23}$$

To derive the second order correction, we need to compute the interaction of the fluctuation potential with all possible excited states K in

$$\Psi_0^{(1)} = \sum_{K \neq 0}^{N_\text{elec}} c_K \Psi_K^{(0)} = \sum_{K \neq 0}^{N_\text{elec}} \frac{\left\langle \Psi_K^{(0)} \middle| \boldsymbol{\mathcal{V}} \middle| \Psi_0^{(0)} \right\rangle}{E_0^{(0)} - E_K^{(0)}} \Psi_K^{(0)}. \tag{3.24}$$

We can omit triply and higher excited states since $\boldsymbol{\mathcal{V}}$ only contains 2-electron contributions. We also know K can not be a singly excited state since Brillouin's theorem states that

$$\left\langle \Psi_0 \middle| \widehat{\boldsymbol{H}} \middle| \Psi_i^a \right\rangle = 0 \tag{3.25}$$

where Ψ_0 and Ψ_i^a are orthogonal so the contribution cancels

$$\begin{aligned}\langle\Psi_0|\mathcal{V}|\Psi_i^a\rangle &= \left\langle\Psi_0\middle|\widehat{\boldsymbol{H}}\middle|\Psi_i^a\right\rangle - \left\langle\Psi_0\middle|\widehat{\boldsymbol{H}}_0\middle|\Psi_i^a\right\rangle \\ &= \underbrace{\cancel{\left\langle\Psi_0\middle|\widehat{\boldsymbol{H}}\middle|\Psi_i^a\right\rangle}}_{0} - \sum_{i=1}^{N_{\text{elec}}} \varepsilon_i \underbrace{\cancel{\langle\Psi_0|\Psi_i^a\rangle}}_{0}.\end{aligned} \tag{3.26}$$

Therefore, the second order energy correction only includes interactions of the ground state wavefunctions with doubly excited states. The energy formula can be derived by using a Taylor expansion on the Eigenvalues and expanding the first-order wavefunction

$$\begin{aligned}E_0^{(2)} &= \sum_{\substack{i<j\\a<b}} \frac{\left|\left\langle\Psi_0\middle|\mathcal{V}\middle|\Psi_{ij}^{ab}\right\rangle\right|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \\ &= \sum_{\substack{i<j\\a<b}} \frac{|\langle ij\|ab\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \\ &= \frac{1}{4}\sum_{ijab} \langle ij\|ab\rangle^2 \; \boldsymbol{\Delta}_{ij}^{ab} = E_{\text{MP2}}.\end{aligned} \tag{3.27}$$

We separated the denominator into the

$$\boldsymbol{\Delta}_{ij}^{ab} = \left(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b\right)^{-1} \tag{3.28}$$

tensor containing the reciprocal energy differences, and we dropped the requirement for an absolute value since we assume real valued integrals throughout this work. This energy correction can be further split into contributions from same and opposite spin electron pairs

$$E_{\text{MP2}} = E_{\text{SS}}^C + E_{\text{OS}}^C. \tag{3.29}$$

In the case of a closed shell system, we can expand the numerator in our MP2 expression to achieve another simplification

$$E_{\text{MP2}} = \frac{1}{4}\sum_{ijab} \langle ij\|ab\rangle^2 \boldsymbol{\Delta}_{ij}^{ab} = \frac{1}{4}\sum_{ijab} ((ia|ab)^2 - 2(ia|jb)(ib|ja) + (ib|ja)^2)\boldsymbol{\Delta}_{ij}^{ab}. \tag{3.30}$$

The repulsion and exchange like terms of the electron quadruplets can be split into two same-spin and two opposite-spin cases which can be written using α and β to differentiate spin states

$$\begin{aligned}&\text{s.s.} \quad \alpha\alpha \to \alpha\alpha \quad \beta\beta \to \beta\beta \\ &\text{o.s.} \quad \alpha\beta \to \alpha\beta \quad \beta\alpha \to \beta\alpha.\end{aligned} \tag{3.31}$$

The simplification arises from the fact that the o.s. terms will not contribute any exchange-like energy terms, since the electrons are already distinguished by their respective spins. This means the spin-integrated o.s. contributions to the energy correction are given by

$$E_{\text{OS}} = 2 \cdot \frac{1}{4}\sum_{ijab} \left((ia|jb)^2 - \underbrace{\cancel{2(ia|jb)(ib|ja)}}_{0} + \underbrace{\cancel{(ib|ja)^2}}_{0} \right) \boldsymbol{\Delta}_{ij}^{ab} \tag{3.32}$$

while for the s.s. contributions the whole expression from Eq. (3.30) has to be evaluated. Adding both cases yields the closed shell MP2 energy expression

$$
\begin{aligned}
E_{\text{MP2-RHF}} &= \frac{1}{4}\sum_{ijab}[2(ia|jb)^2 + 2((ia|jb)^2 - 2(ia|jb)(ib|ja) + (ib|ja)^2)]\boldsymbol{\Delta}_{ij}^{ab} \\
&= \frac{1}{4}\sum_{ijab}(4(ia|jb)^2 - 4(ia|jb)(ib|ja) + 2(ib|ja)^2)\boldsymbol{\Delta}_{ij}^{ab} \\
&= \sum_{ijab}\left((ia|jb)^2 - (ia|jb)(ib|ja) + \frac{1}{2}(ib|ja)^2\right)\boldsymbol{\Delta}_{ij}^{ab}.
\end{aligned}
\tag{3.33}
$$

This can be simplified further by integrating out the spin components and considering the integrals' symmetry[1]. Formulated analogous to the repulsion and exchange like terms in HF

$$
\begin{aligned}
E_{\text{MP2-RHF}} &= \sum_{ijab}[(ia|jb)(2(ia|jb) - (ib|ja))]\boldsymbol{\Delta}_{ij}^{ab} \\
&= \boldsymbol{J}_{ij}^{ab}\left(2\boldsymbol{J}_{ij}^{ab} - \boldsymbol{K}_{ij}^{ab}\right)\boldsymbol{\Delta}_{ij}^{ab}
\end{aligned}
\tag{3.34}
$$

this restricted expression is useful when working with alternative forms of the ERI. Since this thesis only focuses on closed shell systems, Eq. (3.34) will be used as the MP2 energy expression throughout.

In this form the MP2 energy correction is easy to compute given the MO energies and the ERI tensor. The calculation will nevertheless be expensive, mainly due to the fact that the ERI tensor must to converted from the AO to the MO basis

$$
(ia|jb) = \sum_{\mu}^{N_{\text{basis}}}\sum_{\nu}^{N_{\text{basis}}}\sum_{\lambda}^{N_{\text{basis}}}\sum_{\sigma}^{N_{\text{basis}}} \boldsymbol{c}_{\mu i}\boldsymbol{c}_{\nu j}\boldsymbol{c}_{\lambda a}\boldsymbol{c}_{\sigma b}(\mu\lambda|\nu\sigma) \tag{3.35}
$$

which naively implemented needs N_{basis}^8 operations. It can be reduced to $4 \cdot N_{\text{basis}}^5$ manipulating the order of contractions (also called a "contraction pattern")

$$
(ij|ab) = \left(\sum_{\mu}^{N_{\text{basis}}} \boldsymbol{c}_{\mu i}\left(\sum_{\nu}^{N_{\text{basis}}} \boldsymbol{c}_{\nu j}\left(\sum_{\lambda}^{N_{\text{basis}}} \boldsymbol{c}_{\lambda a}\left(\sum_{\sigma}^{N_{\text{basis}}} \boldsymbol{c}_{\sigma b}(\mu\lambda|\nu\sigma)\right)\right)\right)\right). \tag{3.36}
$$

There are therefore two major problems that MP2 calculations are usually associated with:

1. For its computation the order-4 tensors $\boldsymbol{J}$, $\boldsymbol{K}$ and $\boldsymbol{\Delta}$ have to be contracted over all of their indices.
2. The cost of the AO to MO transformation requiring $\mathcal{O}(N_{\text{basis}}^5)$ operations leading to high costs for large systems.
3. The chemical accuracy of the second order correction is limited. The contributions of unpaired electrons are systematically overestimated, which leads MP2 to often underperform when measured against DFT energy calculations[51].

The improvement of the MP2 by introducing empirical corrections as well as approximating parts of the calculation is often discussed in the literature. We will now discuss some ideas, which are specifically useful in the context of ERI decomposition. A noteworthy improvement by approximation was proposed by Almlöf, J. [52] who used the Laplace transform to derive a separable version of the $\boldsymbol{\Delta}_{ij}^{ab}$ denominator. It uses the fact that any reciprocal can be represented as an integral in the Laplace domain

$$\frac{1}{x} = -\frac{1}{x}(0-1) = \int_0^\infty e^{-xt}\,\mathrm{d}t \quad \text{for } x > 0. \tag{3.37}$$

Using this transformation in conjunction with a projected form of Gauss-Legendre quadrature with logarithmically spaced quadrature points, the integral can be approximated by a set of matrix contractions

$$\int_0^\infty e^{\varepsilon_a t} \cdot e^{\varepsilon_b t} \cdot e^{-\varepsilon_i t} \cdot e^{-\varepsilon_j t}\,\mathrm{d}t \approx \sum_v^{N_\text{grid}} \boldsymbol{w}_v \boldsymbol{\tau}_v^a \boldsymbol{\tau}_v^b \boldsymbol{\tau}_v^i \boldsymbol{\tau}_v^j. \tag{3.38}$$

where $\boldsymbol{w}$ contains the weights associated with each grid point. A small set of grid points (5-15) already leads to approximations of the denominator that offer accuracy in the milli-Hartree range.

To further improve the chemical accuracy of MP2 calculations a simple scaling method called Spin Component Scaled (SCS) MP2 was proposed by Grimme, S. [51]. SCS MP2 uses the spin-splitting of the energy contributions shown in Eq. (3.29). In order to counteract the imbalance in energy contributions from paired- and unpaired electrons the publication adds a simple scaling by empirically derived coefficients

$$E_\text{SCS-MP2} = c_\text{OS} E_\text{OS} + c_\text{SS} E_\text{SS}. \tag{3.39}$$

Conducting extensive benchmarking involving single-point as well as reaction energy calculations, the author suggests the scaling factors $c_\text{OS} = \frac{6}{5}$ and $c_\text{SS} = \frac{1}{3}$ which led to a reduction in the Mean Average Error (MAE) of the MP2 correlation energy compared to the Quadratic Configuration Interaction Singled Doubles (QCISD) correlation energy. The MAE decreased from 4.6 to 2.3 kcal/mol. Based on this contribution, a multitude of methods were devised by publications that scale the energy contributions in increasingly evolved ways producing better energies for different chemical systems[53].

One specific method that should be singled out here is the Scaled opposite-spin (SOS) MP2 method devised by Jung, Y.; Lochan, R. C.; Dutoi, A. D.; Head-Gordon, M. [54] who proposed just omitting E_SS completely and instead scaling up E_OS by a factor of $c_\text{OS} = 1.3$. This approach brings the added benefit that the only computation needed is

$$\begin{aligned} E_\text{SOS MP2} &= c_\text{OS} \sum_{ijab} (ia|jb)^2 \boldsymbol{\Delta}_{ij}^{ab} \\ &= c_\text{OS} \sum_{ijab} \Big(\big[\boldsymbol{J}_{ij}^{ab} \boldsymbol{J}_{ij}^{ab} \big] \boldsymbol{\Delta}_{ij}^{ab} \Big). \end{aligned} \tag{3.40}$$

In Section 4 we will describe how this expression can be used in conjunction with the Laplace-transformed denominator to yield a formally $\mathcal{O}\big(N_\text{basis}^3\big)$ scaling version of MP2.

3.3 Coupled Cluster Methods

Expensive tensor operations also limit other calculations in electronic structure theory. One such example can be found in Coupled Cluster methods. The main idea of CC is finding a pattern to include excited slater determinants in the description of the wavefunction, leading to a fundamentally different ansatz than the single determinant in HF. An in-depth discussion of CC concepts can was provided by Crawford, T. D.; Schaefer III, H. F. [55]. Formally, CC introduces the cluster operator $\hat{\boldsymbol{T}}$ that correlates different orbitals with each other

$$\hat{\boldsymbol{T}}_n = \left(\frac{1}{n!}\right)^2 \sum_{ij\ldots ab\ldots}^{n} \boldsymbol{t}_{ij\ldots}^{ab\ldots} \hat{\boldsymbol{a}}_a^\dagger \hat{\boldsymbol{a}}_b^\dagger \ldots \hat{\boldsymbol{a}}_j \hat{\boldsymbol{a}}_i. \tag{3.41}$$

$\hat{\boldsymbol{a}}_a^\dagger$ and $\hat{\boldsymbol{a}}_i$ denote the second-quantization creation and annihilation operators that add orbitals a, b to the slater determinant and remove the orbitals i, j. They are weighted by the coefficient in $\boldsymbol{t}$. With these operators the wavefunction can be expanded in the exponential ansatz, which includes many slater determinants in the description of the wavefunction

$$|\Psi\rangle = e^{\hat{\boldsymbol{T}}}|\Phi_0\rangle = e^{\left(\hat{\boldsymbol{T}}_1 + \hat{\boldsymbol{T}}_2 + \ldots\right)}|\Phi_0\rangle. \tag{3.42}$$

The exponential does not imply an actual numerical power but describes the possibility of expanding the cluster operator using a Taylor series

$$e^{\hat{\boldsymbol{T}}_1}|\Phi_0\rangle = \left(1 + \hat{\boldsymbol{T}} + \frac{\hat{\boldsymbol{T}}_2}{2!} + \frac{\hat{\boldsymbol{T}}_3}{3!} + \ldots\right)|\Phi_0\rangle. \tag{3.43}$$

Φ_0 denotes a reference wavefunction that can be chosen to be the wavefunction from HF. The expansion lends itself to truncation giving rise to the Coupled Cluster Singles (CCS) and Coupled Cluster Singles Doubles (CCSD) variants who model the wavefunction as

$$|\Psi_{\text{CCS}}\rangle = e^{\hat{\boldsymbol{T}}_1}|\Phi_0\rangle \quad \text{and} \quad |\Psi_{\text{CCSD}}\rangle = e^{\hat{\boldsymbol{T}}_1 + \hat{\boldsymbol{T}}_2}|\Phi_0\rangle. \tag{3.44}$$

From this ansatz, an energy expression can be derived by expanding a similarity transformed version of the Hamiltonian projected onto the reference wavefunction

$$E_{\text{CC}} = \left\langle \Phi_0 \middle| e^{-\hat{\boldsymbol{T}}} \widehat{\boldsymbol{H}} e^{\hat{\boldsymbol{T}}} \middle| \Phi_0 \right\rangle. \tag{3.45}$$

This version of the Hamiltonian, also written as $\bar{\boldsymbol{H}}$ can be projected on the excited wavefunction, which gives rise to the amplitude equations, which must be solved iteratively to obtain the so-called "cluster amplitudes" i.e., the weights $\boldsymbol{t}$ for the different excited states. The amplitude equations for the single and double excitations can be expanded as

$$\begin{aligned}
0 &= \left\langle \Phi_i^a \middle| \bar{\boldsymbol{H}} \middle| \Phi_0 \right\rangle = \boldsymbol{f}_{ai} + \sum_c \boldsymbol{f}_{ac}\boldsymbol{t}_i^c - \boldsymbol{f}_{ki}\boldsymbol{t}_k^a + \sum_{kc} \langle ka \| ci \rangle \boldsymbol{t}_k^c + \frac{1}{2}\sum_{kcd} \langle ka \| cd \rangle \boldsymbol{t}_{ki}^{cd} + \ldots \\
0 &= \left\langle \Phi_{ij}^{ab} \middle| \bar{\boldsymbol{H}} \middle| \Phi_0 \right\rangle = \langle ab \| ij \rangle + \sum_c \left(\boldsymbol{f}_{bc}\boldsymbol{t}_{ij}^{ac} - \boldsymbol{f}_{ac}\boldsymbol{t}_{ij}^{bc} \right) + \frac{1}{2}\sum_{kl} \langle kl \| ij \rangle \boldsymbol{t}_{kl}^{ab} + \ldots
\end{aligned} \tag{3.46}$$

where $\boldsymbol{f}_i^a$ are elements of the Fock matrix. Eq. (3.46) involves a kind of tensor contractions with the antisymmetrized ERI that we already saw in Section 3.1. These contractions drive the high computational complexity of CC methods like CCSD.

4 Low-Rank Approximations of the ERI Tensor

Reducing the complexity of handling the ERI tensor by employing low-rank approximation techniques has been a topic of exploration in electronic structure theory from the 1970s onward. This chapter provides a theoretical overview that should function as a guideline for developing a reference implementation of some of the presented ERI approximation methods. Due to the limited scope of this thesis, the chapter does not aim to provide a comprehensive overview of the field. We will only focus on finite molecular systems, as opposed to periodic ones. For a more in-depth exploration of this topic, the reader is referred to Qin, X.; Hu, W.; Yang, J. [56]. We will also focus on the case of Gaussian type basis functions that give rise to real-valued orbitals.

4.1 Properties of Electron Pair Repulsion

The ERI in AO basis over spatial coordinates given real-valued orbitals can be written as

$$\boldsymbol{G}^{\mu\nu}_{\lambda\sigma} = (\mu\nu|\lambda\sigma) = \iint \varphi_\mu(\boldsymbol{r}_1)\varphi_\nu(\boldsymbol{r}_1)\frac{1}{\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|}\varphi_\lambda(\boldsymbol{r}_2)\varphi_\sigma(\boldsymbol{r}_2)\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}\boldsymbol{r}_2 \tag{4.1}$$

where it is common practice to merge the orbital product into so-called "co-densities"

$$(\mu\nu|\lambda\sigma) = \iint \boldsymbol{\rho}_{\mu\nu}(\boldsymbol{r}_1)\frac{1}{\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|}\boldsymbol{\rho}_{\lambda\sigma}(\boldsymbol{r}_2)\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}\boldsymbol{r}_2 = \langle\rho_1|\hat{\boldsymbol{g}}|\rho_2\rangle = (\rho_1|\rho_2). \tag{4.2}$$

The ERI exhibits some specific properties that we will now explore:

Inner product: Since we model AOs with atom-centered Gaussian functions and the product of two Gaussians is another Gaussian centered in between the original functions, we can see that the co-densities must be Gaussian functions. The integral built using the Coulomb operator $\hat{\boldsymbol{g}}_{12} = \|\boldsymbol{r}_1 - \boldsymbol{r}_2\|^{-1}$ acting on the co-densities can be shown to define an inner product on the space spanned by the involved orbitals[57]. This implies that

$$\begin{aligned}
&(1) \quad (\rho_1|\rho_2) = (\rho_2|\rho_1)\\
&(2) \quad (\lambda\rho_1|\rho_2) = \lambda(\rho_1|\rho_2)\\
&(3) \quad (\rho_1 + \rho_3|\rho_2) = (\rho_1|\rho_2) + (\rho_3|\rho_2)\\
&(4) \quad (\rho_1|\rho_1) \geq 0\\
&(5) \quad (\rho_1|\rho_1) = 0 \Leftrightarrow \rho_1 = 0\\
&(6) \quad (.|.) \text{ is positive definite.}
\end{aligned} \tag{4.3}$$

Symmetry: If we assume real orbitals, the ERI possesses 8-fold permutational symmetry. This means that additionally to the symmetry defined by the symmetry of the co-densities the following statement holds

$$(\mu\nu|\lambda\sigma) = (\lambda\sigma|\mu\nu) = (\nu\mu|\sigma\lambda) = (\sigma\lambda|\nu\mu) = (\nu\mu|\lambda\sigma) = (\sigma\lambda|\mu\nu) = (\mu\nu|\sigma\lambda) = (\sigma\lambda|\nu\mu). \tag{4.4}$$

This means that the full amount of floating point numbers that we need to describe the ERI amounts to $\approx \frac{1}{8}N^4_{\text{basis}}$.

Sparseness: If the centers of the co-densities ρ are far apart, the contribution to the integral tensor will be near zero. Depending on the system size this means that most of the entries of the ERI can be assumed to be zero. These contributions can be ignored in calculations. This process is called **integral screening**, and has been studied in the literature[57,58]. The main concern is to

find properties for screening the ERI without fully evaluating it. One such method is using the Cauchy-Schwarz inequality for inner products $\langle .,. \rangle$[14,59]

$$|\langle a, b \rangle| \leq \|a\| \|b\| \tag{4.5}$$

to derive the Schwarz estimate for the ERI

$$|(\mu\nu|\lambda\sigma)| \leq (\mu\nu|\mu\nu)^{\frac{1}{2}} (\lambda\sigma|\lambda\sigma)^{\frac{1}{2}}. \tag{4.6}$$

This inequality implies that if we evaluate all self-repulsions $(\mu\nu|\mu\nu)$ (which we can do in N_{basis}^2 operations), we only have to evaluate entries that involve indices with a self-repulsion above a given cutoff ε. There exist a range of estimates that extend this concept to different types of orbitals and operators. A somewhat recent review of these estimates was provided by Thompson, T. H.; Ochsenfeld, C. [57].

Nearsightedness: A standard assumption in electronic structure theory as laid out by Kohn, W. [60] is that the Coulomb interaction for equilibrium states is "nearsighted". This means that far-apart regions of the system interact weakly. This implies that if the system has sufficient size, we might be able to devise methods that calculate said interaction with only linearly increasing complexity by separating near-field from far-field contributions. This is done for the Coulomb integral by works employing Fast Multipole Methods (FMMs)[61–63]. FMMs use hierarchical boxing algorithms to separate the systems into regions. For each of the regions the near-field interaction with the immediate neighbors is computed. These are subsequently merged to a single "multipole", which then interacts with the other regions multipoles. This simplification unfortunately does not extend to the exchange-like contributions in contractions like the one shown in Eq. (3.19). These calculations "mix" the co-densities and therefore become non-local.

4.2 Density Fitting / Resolution of the Identity

Density Fitting or sometimes called Resolution of the Identity is one of the first methods proposed for using low-rank approximations on the ERI tensor. The initial description of this method can be found in the 1970s in publications from Beebe, N. H. F.; Linderberg, J. [5] as well as Baerends, E. J.; Ellis, D. E.; Ros, P. [64]. The idea behind DF is to find a way to represent the co-densities such that the Coulomb kernel can be expressed in a lower dimensional space.

One way to compute this approximation that is historically significant works by projecting the ERI onto a set of auxiliary basis functions. The derivation of this "auxiliary basis" method, can be found in almost all publications[6,63,65–69] concerned with DF. Although varying in notation and level of detail, the starting point is always the approximation of the co-density as a linear combination of simpler functions from the function class of the original basis set

$$\boldsymbol{\rho}_{\mu\nu}(\boldsymbol{r}) \approx \sum_n \boldsymbol{c}_n \chi_n(\boldsymbol{r}) = \boldsymbol{C}_{\mu\nu}^n \chi_n(\boldsymbol{r}). \tag{4.7}$$

The auxiliary basis should grow linearly with the size of the original basis and is often bigger than the original by a factor of 2 to 5. From Eq. (4.7) we can build an error expression

$$\boldsymbol{\varepsilon}_{\mu\nu,\lambda\sigma} = \left\langle \boldsymbol{\rho}_{\mu\nu} - \boldsymbol{C}_{\mu\nu}^L \chi_L \middle| \mathcal{K} \middle| \boldsymbol{\rho}_{\lambda\sigma} - \boldsymbol{C}_{\lambda\sigma}^K \chi_K \right\rangle \tag{4.8}$$

where we dropped the spatial coordinate $\boldsymbol{r}$ for the sake of readability. In the expression $\mathcal{K}$ denotes a kernel of our choosing. The kernel determines the metric we will measure the error in. It can be shown that minimizing the error in the Coulomb metric yields the highest accuracy approximation,

by producing a robust (quadratically decreasing error) fit in the electric field[66,70]. We can use matrix algebra to derive an analytic solution for the coefficients that minimize this error

$$\begin{aligned} \frac{\partial \boldsymbol{e}_{\mu\nu,\lambda\sigma}}{\partial \boldsymbol{C}^K_{\lambda\sigma}} &= \left(\boldsymbol{\rho}_{\mu\nu} - \boldsymbol{C}^L_{\mu\nu}\chi_L \mid \chi_K\right) = 0 \\ \boldsymbol{C}^L_{\mu\nu} &= \boldsymbol{G}^K_{\mu\nu}\left[\boldsymbol{G}^{-1}\right]_{KL}. \end{aligned} \tag{4.9}$$

Here, $\boldsymbol{G}$ denotes the tensor consisting of the Coulomb interaction of its indexed entities like

$$\begin{aligned} \boldsymbol{G}^K_{\mu\nu} = (\mu\nu|K) &= \iint \varphi_\mu(\boldsymbol{r}_1)\varphi_\nu(\boldsymbol{r}_2)\frac{1}{\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|}\chi_K(\boldsymbol{r}_2)\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}\boldsymbol{r}_2 \\ \boldsymbol{G}_{KL} = (K|L) &= \iint \chi_K(\boldsymbol{r}_1)\frac{1}{\|\boldsymbol{r}_1 - \boldsymbol{r}_2\|}\chi_L(\boldsymbol{r}_2)\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}\boldsymbol{r}_2. \end{aligned} \tag{4.10}$$

From Eq. (4.9) we can derive our ERI approximation by additionally introducing the uninverted Coulomb matrix

$$\begin{aligned} (\mu\nu|\lambda\sigma) &\approx \boldsymbol{C}^K_{\mu\nu}\boldsymbol{G}_{KL}\boldsymbol{C}^L_{\lambda\sigma} \\ &= (\mu\nu|K)\cancel{\left[\boldsymbol{G}^{-1}\right]_{KL}\boldsymbol{G}_{LM}}\left[\boldsymbol{G}^{-1}\right]_{MN}(N|\lambda\sigma) \\ &= (\mu\nu|K)\left[\boldsymbol{G}^{-1}\right]_{KL}(L|\lambda\sigma). \end{aligned} \tag{4.11}$$

The runtime of the DF ERI calculation is dominated by the computation of the $(\mu\nu|K)$ integrals, which can be built in $N^2_{\text{basis}} \cdot N_{\text{aux}} \approx N^3_{\text{basis}}$ operations. As $N_{\text{aux}} \propto N_{\text{basis}}$, the overall asymptotic complexity of the DF build is $\mathcal{O}\big(N^3_{\text{basis}}\big)$. For sufficiently big systems this is less than the N^4_{basis} operations needed to compute the full ERI. The DF formulation is also advantageous in tensor contractions. If we take the expensive AO to MO transformation from Eq. (3.36) and replace the full ERI with the DF ERI, we can derive a new contraction pattern

$$\begin{aligned} \langle ij|ab\rangle &= \left(\sum_\mu^{N_{\text{basis}}} \boldsymbol{c}_{\mu i}\left(\sum_\nu^{N_{\text{basis}}} \boldsymbol{c}_{\nu j}\left(\sum_\lambda^{N_{\text{basis}}} \boldsymbol{c}_{\lambda a}\left(\sum_\sigma^{N_{\text{basis}}} \boldsymbol{c}_{\sigma b}\boldsymbol{G}^K_{\mu\nu}\left[\boldsymbol{G}^{-1}\right]_{KL}\boldsymbol{G}^L_{\lambda\sigma}\right)\right)\right)\right) \\ &= \sum_{KL}\left[\left[\boldsymbol{G}^{-1}\right]_{KL}\left(\sum_\mu \boldsymbol{c}_{\mu i}\left(\sum_\nu \boldsymbol{c}_{\nu j}\boldsymbol{G}^K_{\mu\nu}\right)\right)\left(\sum_\lambda \boldsymbol{c}_{\lambda a}\left(\sum_\sigma \boldsymbol{c}_{\sigma b}\boldsymbol{G}^L_{\lambda\sigma}\right)\right)\right]. \end{aligned} \tag{4.12}$$

In this contraction, the most expensive operation will be the multiplication of the 3-center integrals with the corresponding coefficient matrices $\boldsymbol{c}_{\nu j}\boldsymbol{G}^K_{\mu\nu}$. These can be computed in N^4_{basis} operations. This leads to a reduction of complexity by a full power when compared to the full ERI calculation.

Auxiliary basis Density Fitting has been implemented in many quantum chemistry codes like Gaussian[71], ORCA[72] or PySCF[73] where it is used to increase the speed of quantum chemistry calculations like Hartree-Fock Theory SCF[64,70], MBPT[74] and Coupled Cluster methods[75]. Nevertheless, the structure of the DF approximation can be problematic for some use cases which is mainly due to:

Cost: Even though storing the $(\mu\nu|K)$ tensor requires less space than storing the full ERI, the required $\approx (N^2_{\text{basis}} \cdot N_{\text{aux}})/s$ storage requirement (s being a symmetry factor) can still be prohibitively expensive for larger systems. The same problem is present in the contractions built with the DF integrals. While using $\approx N^4_{\text{basis}}$ operations for the AO to MO transformation is better than contractions with higher powers, it still becomes problematic quite fast even on modern hardware.

Pinned indices: Since the DF formulation results in the λ and σ indices being contracted into the $(\mu\nu|K)$ integral tensor, it is not directly applicable to calculations that require a contraction of the ERI tensor with swapped indices like the exchange term in HF that was shown in Eq. (3.19).

Choice of basis: Using the auxiliary basis method requires the user to choose a set of fitting functions. The method also can be viewed as taking an educated guess at the rank of the ERI by introducing a space to project onto that is chemically motivated. Since we saw in Section 2.1 that determining the rank of a tensor is generally a hard problem, this manual choice can introduce biases and does not adapt to all systems. Early publications[64,65] already acknowledge this problem and throughout the last decades many ways were devised to employ the auxiliary basis method on a variety of systems by manually selecting the "right" fitting basis[76]. There were also efforts to automatically determine the fitting basis[77–79], which employ heuristic algorithms to construct an auxiliary basis.

Another method for computing the DF approximation that addresses the "choice of basis" problem uses the Cholesky Decomposition (Section 2.5). This method was already suggested in one of the first publications on DF by Beebe, N. H. F.; Linderberg, J. [5] which did not gain many citations after its publication, but has been the focus of a number of recent studies[67,69,80]. We can show the connection to the CD by splitting the auxiliary basis formulation of DF from Eq. (4.11)

$$\begin{aligned}(\mu\nu|\lambda\sigma) \approx \quad & (\mu\nu|K)\left[\boldsymbol{G}^{-\frac{1}{2}}\right]_{KL}(L|\lambda\sigma) \\ = & \sum_{KL}\left(\sum_{K}(\mu\nu|K)(K|M)^{-\frac{1}{2}}\right)\left(\sum_{L}(M|L)^{-\frac{1}{2}}(L|\lambda\sigma)\right) \\ = \quad & \boldsymbol{L}^{M}_{\mu\nu} \qquad\qquad\qquad \boldsymbol{L}^{M}_{\lambda\sigma}.\end{aligned} \tag{4.13}$$

If we now unfold the ERI into a matrix by introducing combined indices for the co-densities

$$(\mu\nu|\lambda\sigma) = \boldsymbol{G}^{\overline{\mu\nu}}_{\overline{\lambda\sigma}} = \boldsymbol{G}^{i}_{j} \in \mathbb{R}^{N^2\times N^2} \tag{4.14}$$

the DF approximation is just the pivoted Cholesky Decomposition from Eq. (2.34)

$$\boldsymbol{\Pi}^{T}\ \boldsymbol{G}_{ij}\ \boldsymbol{\Pi} \approx \boldsymbol{L}_{Mi}\boldsymbol{L}_{Mj} = \boldsymbol{L}\boldsymbol{L}^{T}. \tag{4.15}$$

The benefit of this approach is, that the pivoted CD provides an automatic way to approximate the numeric rank of the ERI. The CD method can avoid introducing bias by sacrificing the "chemically intuitive" auxiliary basis in favor of a more abstract space formed by the selected pivots, and it allows the user to define an error threshold that will adapt to any given system. As was outlined by Pedersen, T. B.; Lehtola, S.; Fdez. Galván, I.; Lindh, R. [69], this process can be implemented to match the computational complexity of the auxiliary basis algorithm: The key insight is that for most systems the rank of the ERI grows only linearly. From Algorithm 1 we learned that only the diagonal with N^2_{basis} entries needs to be stored. For the formation of the Cholesky vectors only some parts of the input have to be computed, which can be done on the fly. Therefore, we can achieve an implementation of overall $\mathcal{O}\big(N^3_{\text{basis}}\big)$ computational complexity. This was already postulated by Beebe, N. H. F.; Linderberg, J. and thereafter verified by many implementations[69].

4.3 Tensor Hypercontraction

In 2012 Hohenstein, E. G.; Parrish, R. M.; Martínez, T. J. [6] introduced the Tensor Hypercontraction, which proposed that the ERI may be approximated by a matrix product of the form

$$\boldsymbol{G}_{\mu\nu}^{\lambda\sigma} \approx \boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{P}\boldsymbol{Z}^{PQ}\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\sigma}^{Q} \tag{4.16}$$

where $P, Q, R, S...$ denote the abstract space used to project the ERI into. The THC approximation is closely related to the pseudospectral family of methods[81], which expand one side of the DF approximation using numerical quadrature on a grid. To compute this factorization the publication uses the CPD (Section 2.7). The starting point is the observation by Vahtras, O.; Almlöf, J.; Feyereisen, M. W. [70] that the auxiliary basis expansion error shown in Eq. (4.8) can also be expanded in other metrics than the Coulomb metric. If one uses the overlap as a measure of fit, the ERI can be also approximated by

$$(\mu\nu|\lambda\sigma) \approx (\mu\nu K)\big[\boldsymbol{S}^{-1}\big]_{KL}(L|M)\big[\boldsymbol{S}^{-1}\big]_{MN}(N\lambda\sigma) \tag{4.17}$$

where

$$\begin{aligned}(\mu\nu K) &= \int \varphi_{\mu}(\boldsymbol{r})\varphi_{\nu}(\boldsymbol{r})\chi_{K}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \\ \boldsymbol{S}_{KL} &= \int \chi_{K}(\boldsymbol{r})\chi_{L}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}.\end{aligned} \tag{4.18}$$

Canonic Polyadic Tensor Hypercontraction (CP-THC) factorizes the three-center integral matrices into three component matrices with an approximate rank P. The rank should be proportional to N_{basis} much like the amount of auxiliary basis functions employed by the DF approach

$$(\mu\nu K) \approx \boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{P}\boldsymbol{Y}_{K}^{P} \tag{4.19}$$

where the weight matrix $\boldsymbol{\lambda}$ from Eq. (2.43) was contracted into the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices. From this factorization the THC formulation of the ERI can be written as

$$\begin{aligned}(\mu\nu|\lambda\sigma) &\approx \big[\boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{P}\boldsymbol{Y}_{K}^{P}\big]\big[\boldsymbol{S}^{-1}\big]_{KL}(L|M)\big[\boldsymbol{S}^{-1}\big]_{MN}\big[\boldsymbol{Y}_{N}^{Q}\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\sigma}^{Q}\big] \\ &= \boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{P}\Big(\boldsymbol{Y}_{K}^{P}\big[\boldsymbol{S}^{-1}\big]_{KL}(L|M)\big[\boldsymbol{S}^{-1}\big]_{MN}\boldsymbol{Y}_{N}^{Q}\Big)\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\sigma}^{Q} \\ &= \boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{P}\boldsymbol{Z}^{PQ}\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\sigma}^{Q}.\end{aligned} \tag{4.20}$$

The storage required by the ERI can be drastically reduced using THC. We only need to store the $\boldsymbol{X} \in \mathbb{R}^{N_{\text{basis}} \times N_{\text{THC}}}$ and $\boldsymbol{Z} \in \mathbb{R}^{N_{\text{THC}}^2}$ matrices. If we construct the THC such that $N_{\text{THC}} \propto N_{\text{basis}}$ the asymptotic storage requirement reduces to $\mathcal{O}(N_{\text{basis}}^2)$.

The "pinned indices" problem from DF is also addressed. Each $\mu, \nu, \lambda, \sigma$ index is now indexing the same $\boldsymbol{X}$ matrix, while preserving the ERI symmetries. This simplifies contractions of the form

$$\begin{aligned}\boldsymbol{K}_{\mu\nu} &= \mathbf{P}_{\lambda\sigma}\ \boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\sigma}^{P}\boldsymbol{Z}^{PQ}\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\nu}^{Q} \\ &= \Big(\mathbf{P}_{\lambda\sigma}\boldsymbol{X}_{\lambda}^{Q}\boldsymbol{X}_{\sigma}^{P}\Big)\boldsymbol{X}_{\mu}^{P}\boldsymbol{X}_{\nu}^{Q}\boldsymbol{Z}^{PQ}.\end{aligned} \tag{4.21}$$

This in turn allows us to lower the computational complexity of the electron-electron repulsion calculation

$$\begin{aligned}E_{ee} &= \frac{1}{2}\mathbf{P}_{\mu\nu}\big(\boldsymbol{J}_{\mu\nu} - \boldsymbol{K}_{\mu\nu}\big) \\ &= \frac{1}{2}\mathbf{P}_{\mu\nu}(\mathbf{P}_{\lambda\sigma}(\mu\nu|\lambda\sigma) - \mathbf{P}_{\lambda\sigma}(\mu\sigma|\lambda\nu)).\end{aligned} \tag{4.22}$$

With the THC approximation these can be computed with $\mathcal{O}\big(N_{\text{basis}}^4\big)$ to $\mathcal{O}\big(N_{\text{basis}}^3\big)$.

Hohenstein, E. G.; Parrish, R. M.; Martínez, T. J. [6] proposed a scheme to compute the closed-shell Møller–Plesset Perturbation Theory (Second Order) (MP2) energy correction with $\mathcal{O}\big(N_{\text{basis}}^4\big)$ asymptotic complexity using the THC. Since all the basis function indices are separate now, we can lower the complexity of the AO to MO transformation from Eq. (3.36). This is done by contracting the $\boldsymbol{X}$ matrix with the SCF weights

$$\boldsymbol{X}_i^P = \boldsymbol{c}_{\mu i} \boldsymbol{X}_\mu^P \tag{4.23}$$

which requires $\approx N_{\text{basis}}^2 \cdot N_{\text{THC}}$ operations. We introduce the convention here, that the $\boldsymbol{X}$ matrix when indexed with MO indices (i, j, a, b) will refer to the MO transformed matrix as to avoid multiple different ways to write $\boldsymbol{X}$. Using these matrices, we can put the THC ERI into the Laplace transformed closed-shell MP2 expression from Eq. (3.34)

$$\begin{aligned} E_{\text{RHF-MP2}} &= \sum_{i,j,a,b} \underbrace{-2(ia|jb)^2 \Delta_{ij}^{ab}}_{J} + \underbrace{(ia|jb)(ib|ja)\Delta_{ij}^{ab}}_{K} \\ &\approx -2 \cdot \boldsymbol{\tau}_v^a \boldsymbol{\tau}_v^b \boldsymbol{\tau}_v^i \boldsymbol{\tau}_v^j \cdot \boldsymbol{X}_i^P \boldsymbol{X}_a^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_j^Q \boldsymbol{X}_b^Q \cdot \boldsymbol{X}_i^R \boldsymbol{X}_a^R \boldsymbol{Z}^{RS} \boldsymbol{X}_j^S \boldsymbol{X}_b^S \\ &\quad + \boldsymbol{\tau}_v^a \boldsymbol{\tau}_v^b \boldsymbol{\tau}_v^i \boldsymbol{\tau}_v^j \cdot \boldsymbol{X}_i^P \boldsymbol{X}_a^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_j^Q \boldsymbol{X}_b^Q \cdot \boldsymbol{X}_i^R \boldsymbol{X}_b^R \boldsymbol{Z}^{RS} \boldsymbol{X}_j^S \boldsymbol{X}_a^S \end{aligned} \tag{4.24}$$

where the weights of the Laplace transformed integral are merged into the $\boldsymbol{\tau}$ matrices. After some rearranging, we can find a contraction pattern that allows for the J and K term to be computed with $\mathcal{O}\big(N_{\text{basis}}^3\big)$ and $\mathcal{O}\big(N_{\text{basis}}^4\big)$ asymptotic complexity respectively. This contraction pattern is shown in Algorithm 3 together with the corresponding amount of operations needed. To improve readability, the operation costs neglect the small constant amount of Laplace integration grid points. In Algorithm 3 the "contract" function is a stand-in for tensor contraction functions like the ones provided by TBLIS[82], which will contract all indices in the parameter tensor in a way that produces the indices on the desired output while building optimal intermediates. A similar derivation for the MP3 correction was also proposed by Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83]. In order to arrive at the SOS approximation (Section 3.2), it suffices to compute the J term. This is then multiplied with the empiric scaling factor.

Algorithm 3: Laplace Transform MP2 Energy using the THC ERI

procedure THC-MP2-Laplace$\big(\boldsymbol{X} \in \mathbb{R}^{N_{\text{THC}} \times N_{\text{basis}}}, \boldsymbol{Z} \in \mathbb{R}^{N_{\text{THC}} \times N_{\text{THC}}}, \boldsymbol{\tau} \in \mathbb{R}^{N_{\text{basis}} \times N_{\text{laplace}}}\big)$
- ▷ Calculate J
- $\boldsymbol{V}_v^{PR} \leftarrow \text{contract}(\boldsymbol{\tau}_v^a \boldsymbol{X}_a^P \boldsymbol{X}_a^R)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{virt}}$
- $\boldsymbol{O}_v^{PR} \leftarrow \text{contract}\big(\boldsymbol{\tau}_v^i \boldsymbol{X}_i^P \boldsymbol{X}_i^R\big)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{occ}}$
- $\boldsymbol{L}_v^{PR} \leftarrow \boldsymbol{V} * \boldsymbol{O}$ ▷ N_{THC}^2
- $J \leftarrow \text{contract}\big(\boldsymbol{L}_v^{PR} \boldsymbol{L}_v^{QS} \boldsymbol{Z}^{PQ} \boldsymbol{Z}^{RS}\big)$ ▷ N_{THC}^3 (optimized)
- ▷ Calculate K
- $\boldsymbol{B}_{ia}^P \leftarrow \text{contract}\big(\boldsymbol{Z}^{PQ} \boldsymbol{X}_i^Q \boldsymbol{X}_a^Q\big)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{occ}} \cdot N_{\text{virt}}$
- $\boldsymbol{E}_{vi}^{PR} \leftarrow \text{contract}\big(\boldsymbol{\tau}_v^a \boldsymbol{X}_a^R \boldsymbol{B}_{ia}^P\big)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{occ}} \cdot N_{\text{virt}}$
- $\boldsymbol{H}_v^{PR} \leftarrow \text{contract}\big(\boldsymbol{\tau}_v^i \boldsymbol{E}_{vi}^{PR} \boldsymbol{E}_{vi}^{PR}\big)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{occ}}$
- $\boldsymbol{G}_v^{PR} \leftarrow \text{contract}\big(\boldsymbol{\tau}_v^i \boldsymbol{X}_i^P \boldsymbol{X}_i^R\big)$ ▷ $N_{\text{THC}}^2 \cdot N_{\text{occ}}$
- $K \leftarrow \text{contract}(\boldsymbol{G}_v^{PR} \boldsymbol{H}_v^{PR})$ ▷ N_{THC}^2
- **return** $-2 \cdot J + K$

end

It also should be noted that the way Algorithm 3 calculates the K term is not optimal for practical scenarios and can require a high amount of memory. A better optimized method was proposed by Bangerter, F. H.; Glasbrenner, M.; Ochsenfeld, C. [84]. This pattern will be used in later chapters.

The THC and especially the method of its construction via the CPD also comes with some theoretical and practical drawbacks: Like DF, it requires the user to choose an auxiliary basis set, as they are needed to compute the central $\boldsymbol{Z}^{PQ}$ matrix. Additionally, the construction performance is usually limited by the ALS algorithm that computes the CPD. The authors acknowledge that the ALS procedure itself formally possesses $\mathcal{O}(N_{\text{basis}}^4)$ computational complexity and only converged slowly in their tests with $\approx 10^5$ ALS cycles needed to determine the factor matrices.

The reduction in complexity for the Laplace transformed MP calculations was validated empirically on a set of linear growing alkane molecules and water clusters in a correlation consistent double zeta basis set (cc-pVDZ)[85]. In this setup the CP-THC method produced MP2 and MP3 energies that only deviated from the canonical value in the milli-Hartree range. However, the publication does not include wall-time comparison of CP-THC MP2 and DF MP2, but only show produced timings in an MP3 calculation including the THC build phase. These already takes several minutes for systems with ~ 100 basis functions, which can be computed by DF MP2 milliseconds.

This sets up one of the main problems when exploring these kinds of approximations: Determining the **cost-benefit ratio** of a given method. It is easy to see that in the asymptotic case, building the DF approximation will be faster than constructing the THC, at least using the CPD. On the other hand, the higher speed in the desired calculation, might add up to a net gain in efficiency for certain systems. This topic is hard to address since it depends on many variables: The specific implementations and degree of optimization used, the hardware employed, the test-data etc. Because of this complexity it is difficult for inventors of new approximation methods to properly evaluate their method, and it is challenging for users to find the method that best fits their use case.

Therefore, for the rest of this thesis we will try to address this issue by
1. describing other ways to achieve a low-rank approximation that were put forward
2. developing a reference implementation in a unified software framework for a chosen subset
3. benchmarking the implementations on a unified set of molecules.

4.4 Least-squares THC

To address the issues in CP-THC arising from its use of the ALS algorithm Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] established a direct method of computing the THC factorization using a Least Squares minimization approach. The key difference between Least Squares Tensor Hypercontraction (LS-THC) and CP-THC is the use of a molecular quadrature grid, which is also adapted from the pseudospectral method family. A common choice in the context of molecular systems is to use a Becke-type grid[86]. This approach employs atom-centered grids with user defined angular and radial densities delivers, such that the grid points might be used in a Chebyshev-Gauss numerical integration procedure.

In order to shorten the subsequent equations where we introduce the following expressions

$$\begin{aligned} \boldsymbol{X}_\mu^P &= \sqrt[4]{w_P} \cdot \varphi_\mu(\boldsymbol{r}_P) \\ \boldsymbol{X}_{\mu\nu}^P &= \boldsymbol{X}_\mu^P \odot \boldsymbol{X}_\nu^P \\ \boldsymbol{S}^{PQ} &= \boldsymbol{X}_{\mu\nu}^P \boldsymbol{X}_{\mu\nu}^Q = \left[\boldsymbol{X}_\mu^P \boldsymbol{X}_\mu^P\right]\left[\boldsymbol{X}_\nu^Q \boldsymbol{X}_\nu^Q\right]. \end{aligned} \tag{4.25}$$

where w_P is the weight associated with the grid point $\boldsymbol{r}_P$.

The LS-THC algorithm poses the optimization problem

$$\arg\min_{\boldsymbol{Z}^{PQ}} \left[O = \frac{1}{2} \left\| (\mu\nu|\lambda\sigma) - \boldsymbol{X}_\mu^P \boldsymbol{X}_\nu^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_\lambda^Q \boldsymbol{X}_\sigma^Q \right\|^2 \right] \tag{4.26}$$

which has an analytic solution in

$$\frac{\partial O}{\partial \boldsymbol{Z}^{P'Q'}} = -\boldsymbol{X}_{\mu\nu}^{P'} \left[(\mu\nu|\lambda\sigma) - \boldsymbol{X}_{\mu\nu}^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_{\lambda\sigma}^Q \right] \boldsymbol{X}_{\lambda\sigma}^{Q'} = 0. \tag{4.27}$$

From this we can derive an analytic expression for the $\boldsymbol{Z}^{PQ}$ matrix

$$\begin{aligned} \boldsymbol{Z}^{PQ} &= \left[\boldsymbol{X}_{\mu'\nu'}^P \boldsymbol{X}_{\mu'\nu'}^{P'} \right]^{-1} \boldsymbol{X}_{\mu\nu}^{P'} (\mu\nu|\lambda\sigma) \boldsymbol{X}_{\lambda\sigma}^{Q'} \left[\boldsymbol{X}_{\lambda'\sigma'}^{Q'} \boldsymbol{X}_{\lambda'\sigma'}^Q \right]^{-1} \\ &\approx \left[\boldsymbol{S}^{PP'} \right]^+ \left(\boldsymbol{X}_{\mu\nu}^P (\mu\nu|\lambda\sigma) \boldsymbol{X}_{\lambda\sigma}^Q \right) \left[\boldsymbol{S}^{Q'Q} \right]^+ \\ &= \left[\boldsymbol{S}^{PP'} \right]^+ \boldsymbol{E}^{PQ} \left[\boldsymbol{S}^{Q'Q} \right]^+. \end{aligned} \tag{4.28}$$

Fitting with the ERI is the most computationally complex step in this THC building method. Even if fit on the DF version of the ERI, it contains a contraction of the form

$$\boldsymbol{X}_{\mu\nu}^P \boldsymbol{L}_{\mu\nu}^M \tag{4.29}$$

which involves $N_{\text{basis}}^2 \cdot N_{\text{aux}} \cdot N_{\text{THC}}$ which leads to $\mathcal{O}\left(N_{\text{basis}}^4\right)$ asymptotic complexity or $\mathcal{O}\left(N_{\text{basis}}^5\right)$ if fit with the full ERI.

Another problematic and time-consuming step of the algorithm is the inversion of the grid metric matrix $\boldsymbol{S}^{PQ}$. As it is usually rank-deficient, a proper inverse might not exist. So Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] employ a pseudoinversion technique in which the eigendecomposition of the matrix is computed and subsequently the pseudoinverse is derived from the eigenvalues which lie above a user defined cutoff ε

$$\left[\boldsymbol{S}^{PP'} \right]^+ = \boldsymbol{U}_{P_i} \boldsymbol{\lambda}_i^{-1} \boldsymbol{U}_{P_i'} \quad \lambda_i > \varepsilon \lambda_1. \tag{4.30}$$

This involves solving the eigenvalue problem for a Hermitian matrix for which LAPACK[21] uses the Divide and Conquer algorithm mentioned in Section 2.4. This imposes a $\mathcal{O}\left(N_{\text{grid}}^3\right)$ computational complexity. Depending on the choice of grid this can lead a big computational pre-factor. If the grid is dense (i.e., > 1000 points per atom) this inversion can make LS-THC expensive when compared to DF, for systems that have only few atoms.

The LS-THC method was shown to accurately reproduce the MP2 energies of alkane chains, water clusters, linear acenes and alanine polypeptides in the cc-pVDZ basis set with errors in the milli-Hartree region. When fit with the full ERI, LS-THC was shown to approximate the MP2 energies better than DF. The publications tests were run using a sparse grid of ≈ 150 points per atom. Unfortunately, the publication does not include runtime comparison of DF, CP-THC and LS-THC. A later study by Matthews, D. A. [87] provided insights into the runtime behavior of LS-THC, finding a practical cost-to-benefit crossover point at system sizes of ≈ 240 electrons when comparing to DF MP3.

After the inception of LS-THC, more authors suggested ways to improve upon the methodology in different ways. Studies that apply the THC to electronic structure calculations can be found for open-shell and restricted MP2 and MP3 theories[84,88,89]; MP2 for periodic systems[90]; Coupled

Cluster methods[12,91–93]; Hartree-Fock Theory and Density Functional Theory[94–96] as well as other methods that were out of the scope of this thesis like GW and the Random Phase Approximation (RPA)[97,98]. Since the choice of grid is important to the applicability of the method, many later publications are concerned with optimization of molecular grids for the THC construction in order to lower the computational pre-factor. Some propose different types of integration grids but keep the manual choice of grid[99] while many others devise automatic grid optimization schemes[100–105]. Some approaches mentioned even provide means to lower the overall asymptotic scaling of the THC build to $\mathcal{O}\big(N_{\text{basis}}^3\big)$[96,103,105]. We will examine the most promising of these approaches in the following chapters.

4.5 Interpolative Separable Density Fitting

In order to reduce the complexity of building the THC representation of the ERI, Lu, J.; Ying, L. [7] introduced the Interpolative Separable Density Fitting (ISDF) procedure. The core idea of ISDF is to use the Interpolative Decomposition described in Section 2.6 in order to find a low-rank approximation of the ERI co-densities

$$\varphi_\mu(\boldsymbol{r})\varphi_\nu(\boldsymbol{r}) \approx \sum_{\tilde{P}} \varphi_\mu(\boldsymbol{r}_{\tilde{P}})\varphi_\nu(\boldsymbol{r}_{\tilde{P}})\xi_{\tilde{P}}(\boldsymbol{r}) \tag{4.31}$$

where $\boldsymbol{r}_P$ are points of a molecular grid and $\xi_{\tilde{P}}(\boldsymbol{r})$ is a set of interpolation functions. To cast this equation into matrix-form we must assume the co-densities to be discretized over a molecular integration grid

$$\boldsymbol{X}^P_{\mu\nu} \approx \boldsymbol{X}^{\tilde{P}}_{\mu\nu}\boldsymbol{\xi}^P_{\tilde{P}} \tag{4.32}$$

where P denotes points in the full grid and $\tilde{P}$ indexes a chosen subset of interpolation points. From this we can see the similarity to Eq. (2.36) making $\boldsymbol{X}^{\tilde{P}}_{\mu\nu}$ and $\boldsymbol{\xi}^P_{\tilde{P}}$ the skeleton and the interpolation matrices respectively. In order to lower the amount of co-densities that need to be evaluated, the ISDF procedure applies what amounts to a "sketchy" pivoting technique mentioned in Section 2.6.

The procedure starts by computing a "mixing" matrix $\boldsymbol{M} \in \mathbb{R}^{N^2_{\text{basis}} \times N_{\text{grid}}}$ from an unfolded version of the co-density tensor. The transformation is then computed by first multiplying the co-density row-wise with a random complex number of unit-length and then applying a Fast Fourier Transform (FFT) to each column. This mixes co-densities of the AO function pairs on each grid point. Subsequently, ISDF takes a random sample of size k from the now mixed AO pairs, which reduces $\boldsymbol{M} \in \mathbb{C}^{N^2_{\text{basis}} \times N_{\text{grid}}}$ to the $\widetilde{\boldsymbol{M}} \in \mathbb{C}^{k \times N_{\text{grid}}}$ sketch of the co-density matrix. To view the process more clearly, Matthews, D. A. [106] writes the mixing and sampling operations as matrix operators $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{S}}$ in

$$\widetilde{\boldsymbol{M}} = \boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{P}}\boldsymbol{X}. \tag{4.33}$$

In order to compute the interpolation points aka. the skeleton matrix, QRCP decomposition described in Section 2.4 is applied on $\boldsymbol{M}$ yielding

$$\widetilde{\boldsymbol{M}}\boldsymbol{\Pi} = \boldsymbol{Q}\boldsymbol{R} \tag{4.34}$$

where size of the diagonal element in $\boldsymbol{R}$ provides an indicator for the relevance of the grid in context of the co-density. From $\boldsymbol{R}$ we select N_{THC} columns using an error cutoff ε applied on the diagonal. From this "pruned" version of the grid, the interpolation functions can be computed

$$\boldsymbol{\xi}^P_{\tilde{P}} = \Big[\boldsymbol{R}_{1:N_{\text{THC}},1:N_{\text{THC}}}\Big]^{-1}\boldsymbol{R}_{1:N_{\text{THC}},:}\boldsymbol{\Pi}^{-1}. \tag{4.35}$$

Once the interpolation functions are known, the ERI can be reconstructed from the interpolation functions like

$$\begin{aligned}(\mu\nu|\lambda\sigma) &\approx \boldsymbol{X}_{\mu\nu}^{\tilde{P}} \left[\iint \frac{\boldsymbol{\xi}_{\tilde{P}}(\boldsymbol{r}_P)\boldsymbol{\xi}_{\tilde{Q}}(\boldsymbol{r}_Q)}{\|\boldsymbol{r}_P - \boldsymbol{r}_Q\|} \,\mathrm{d}\boldsymbol{r}_P \,\mathrm{d}\boldsymbol{r}_Q \right]^{\tilde{P}\tilde{Q}} \boldsymbol{X}_{\lambda\sigma}^{\tilde{Q}} \\ &= \boldsymbol{X}_{\mu\nu}^{\tilde{P}} \boldsymbol{Z}^{\tilde{P}\tilde{Q}} \boldsymbol{X}_{\lambda\sigma}^{\tilde{Q}}\end{aligned} \tag{4.36}$$

which is very similar to the THC formulation. We just renamed the interpolation points to match the previously used formulations. The expression is clearly separable in terms of the basis function pairs, which means that it can be contracted with $\mathcal{O}\big(N_{\text{basis}}^3\big)$ computational complexity. When using a periodic basis, the matrix can be evaluated using the FFT like

$$\boldsymbol{Z}^{PQ} = \frac{4\pi}{\Omega} \sum_{\boldsymbol{g}\neq 0} \frac{\boldsymbol{\xi}_P(\boldsymbol{g})\boldsymbol{\xi}_Q(\boldsymbol{g})}{\boldsymbol{g}^2} \tag{4.37}$$

where Ω denotes the volume of the unit cell and $\boldsymbol{g}$ the reciprocal lattice vector[56]. In the context of molecular systems this calculation is challenging. Since the diagonal elements of $\boldsymbol{Z}^{PQ}$ contain an integrable singularity for the case $\boldsymbol{r}_P = \boldsymbol{r}_Q$, the problem is numerically ill-conditioned. A solution was proposed by Lee, J.; Lin, L.; Head-Gordon, M. [103] who employed an auxiliary basis set (indexed by$K, L, M, N, ...$) to derive an alternate expression for the Coulomb kernel matrix

$$\begin{aligned}\boldsymbol{Z}^{PQ} &= \boldsymbol{D}_K^P \boldsymbol{D}_K^Q \\ \boldsymbol{D}_K^P &= \left[\boldsymbol{G}^{-\frac{1}{2}}\right]_{KL} (L|\xi_K) \\ &= \left[\boldsymbol{G}^{-\frac{1}{2}}\right]_{KL} \boldsymbol{G}_{\mu\nu}^L \boldsymbol{X}_{\mu\nu}^Q [\boldsymbol{S}^+]_{PQ}.\end{aligned} \tag{4.38}$$

Although leading to correct results, the contraction of the 3-center Coulomb integrals with the co-density tensor implies that the algorithm will again possess $\mathcal{O}\big(N_{\text{basis}}^4\big)$ computational complexity. The publication suggests integral screening methods could be used to decrease the complexity to a value closer to $\mathcal{O}\big(N_{\text{basis}}^3\big)$. A full solution to this problem in molecular systems in the ISDF framework was not yet found. This fact is presented in a review paper by Qin, X.; Hu, W.; Yang, J. [56], which provides an excellent albeit only theoretical overview of ISDF.

A major improvement for the process of finding interpolation points was made by Dong, K.; Hu, W.; Lin, L. [102], who suggested using a Centroidal Voronoi Tesselation (CVT) procedure to identify relevant grid points. The CVT procedure when given a set of weighted points and an amount of centroids computes a placement for the centroids such that the weighted distance for each gridpoint to the next centroid is minimized

$$\underset{\{\mathcal{C}_k, \boldsymbol{c}_k\}}{\arg\min} \quad \sum_k^N \sum_{\boldsymbol{r}_l \in \mathcal{C}_k} w(\boldsymbol{r}_l) \|\boldsymbol{r}_l - \boldsymbol{c}_k\|^2 \tag{4.39}$$

where N is a user-defined amount of centroids, w is the weighting function, $\mathcal{C}_k$ is a cluster and $\boldsymbol{c}_k$ is the cluster's centroid. This is a non-linear optimization problem. The authors address this by using the weighted K-Means clustering procedure, which starts with a random guess at the centroid placements; assigns all points to the nearest centroid; computes the new centroids as the cluster averages and repeats until convergence. The procedure is outlined in Algorithm 4.

An in-depth exploration of molecular ISDF using the K-Means clustering approach was conducted by Lee, J.; Lin, L.; Head-Gordon, M. [103]. The publication used two dedicated benchmarking

Algorithm 4: Weighted K-Means Clustering

procedure K-Means$\left(\{\boldsymbol{r}_i\}^{N_{\text{grid}}},\ w(\boldsymbol{r}),\ \text{Initial Centroids}\ \left\{\boldsymbol{c}_k^{(0)}\right\}^{N_C}\right)$
  $t \leftarrow 0$
  while not converged or maximum steps reached **do**
    ▷ Assign grid points to clusters
    for $i = 1$ to N_{grid} **do**
      assign$\left(\mathcal{C}_k^{(t)}, \boldsymbol{r}_i\right)$ if $\boldsymbol{r}_i$ is closest to $\boldsymbol{c}_k^{(t)}$
    end

    ▷ Update cluster centroids
    for $k = 1$ to N_c **do**
      $\boldsymbol{c}_k^{(t+1)} \leftarrow \left[\sum_{\boldsymbol{r}_j \in \mathcal{C}_k^{(t)}} \boldsymbol{r}_j\ w(\boldsymbol{r}_j)\right] / \left[\sum_{\boldsymbol{r}_j \in \mathcal{C}_k^{(t)}} w(\boldsymbol{r}_j)\right]$
    end
  end

  return $\left\{\boldsymbol{c}_k^{(t)}\right\}$
end

datasets[107,108] and showed that a number of interpolation points $\approx 3-5$ times the size of the auxiliary basis was necessary to achieve milli-Hartree accuracy in HF, MP2 and MP3 calculations. The authors note that the convergence behavior depends on the size of the basis set. Bigger basis sets will generally need more interpolation points. The implementation used Schwarz-type integral screening and could be shown to outperform DF on systems of around 600 basis functions. The central THC matrix for this system could be built in under 10 minutes. The authors also estimate the method to outperform DF in non-screened MP2 calculations at system sizes between 1000 and 2000 electrons. All the calculations were conducted using the Q-Chem package[109] on a 32 core CPU.

4.6 Improvements upon LS-THC

Another automated grid pruning scheme was proposed by Matthews, D. A. [106]. The derivation adapts ideas from ISDF and the QRCP decomposition of the co-density as a starting point

$$\begin{aligned} \boldsymbol{S}^{PQ} &= \boldsymbol{X}^P_{\mu\nu}\boldsymbol{X}^Q_{\mu\nu} = \boldsymbol{X}^T\boldsymbol{X} \approx \boldsymbol{M}^T\boldsymbol{M} \\ &= (\boldsymbol{\Pi}\boldsymbol{R}^T\boldsymbol{Q}^T)(\boldsymbol{Q}\boldsymbol{R}\boldsymbol{\Pi}^T). \end{aligned} \tag{4.40}$$

Since $\boldsymbol{Q}$ is orthogonal, $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{R}$ is upper triangular, we can recover the formula for the pivoted Cholesky Decomposition

$$\begin{aligned} \boldsymbol{S} &= \boldsymbol{\Pi}\boldsymbol{R}^T(\boldsymbol{Q}^T\boldsymbol{Q})\boldsymbol{R}\boldsymbol{\Pi}^T \\ &= \boldsymbol{\Pi}\boldsymbol{R}^T\mathbb{I}\boldsymbol{R}\boldsymbol{\Pi}^T \\ \boldsymbol{\Pi}^T\boldsymbol{S}\boldsymbol{\Pi} &= \boldsymbol{R}^T\boldsymbol{R}. \end{aligned} \tag{4.41}$$

This implies that we can find a low-rank approximation for $\boldsymbol{S}^{PQ}$ using the pivoted CD. We can use Eq. (4.28) to derive an expression for the Coulomb matrix, where the fitting matrix $\boldsymbol{E}$ is constructed with the pruned version of the colocation matrix $\boldsymbol{S}^{\tilde{P}\tilde{Q}} \in \mathbb{R}^{k\times k}$

$$\begin{aligned}
\boldsymbol{Z}^{\tilde{P}\tilde{Q}} &= \left[\boldsymbol{S}^{-1}\right]^{\tilde{P}\tilde{R}}\left[\boldsymbol{X}^{\tilde{R}}_{\mu\nu}(\mu\nu|\lambda\sigma)\boldsymbol{X}^{\tilde{S}}_{\lambda\sigma}\right]\left[\boldsymbol{S}^{-1}\right]^{\tilde{Q}\tilde{S}} \\
&= \left[\boldsymbol{S}^{-1}\right]^{\tilde{P}\tilde{R}}\boldsymbol{E}_{\tilde{R}\tilde{S}}\left[\boldsymbol{S}^{-1}\right]^{\tilde{Q}\tilde{S}} \\
&= \left[\left(\boldsymbol{L}^T\boldsymbol{L}\right)^{-1}\right]^{\tilde{P}\tilde{R}}\boldsymbol{E}_{\tilde{R}\tilde{S}}\left[\left(\boldsymbol{L}^T\boldsymbol{L}\right)^{-1}\right]^{\tilde{Q}\tilde{S}}.
\end{aligned} \tag{4.42}$$

With this expression, we can either decide to build the direct inversion or a pseudoinversion of $\boldsymbol{L}^T\boldsymbol{L}$. We can also use triangular forward solves, which might increase numeric stability. Starting from

$$\boldsymbol{Z} = \boldsymbol{L}^{-1}\left[\boldsymbol{L}^T\right]^{-1}\boldsymbol{E}\boldsymbol{L}^{-1}\left[\boldsymbol{L}^T\right]^{-1}. \tag{4.43}$$

the center matrix can be found in 4 solve steps

$$\begin{aligned}
\boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{A} &= \boldsymbol{E} && \text{solve for } \boldsymbol{A} \\
\boldsymbol{L}^T\boldsymbol{B}^T &= \boldsymbol{A}^T && \text{solve for } \boldsymbol{B}^T \\
\boldsymbol{L}\boldsymbol{C} &= \boldsymbol{B} && \text{solve for } \boldsymbol{C} \\
\boldsymbol{L}\boldsymbol{Z}^T &= \boldsymbol{C}^T && \text{solve for } \boldsymbol{Z}^T.
\end{aligned} \tag{4.44}$$

This grid pruning technique was tested on a benzene molecule in differently sized basis sets. It was shown to reliably decrease the needed grid points by $\approx 60\%$ even for very large cc-pV6Z basis sets. This is highly desirable since all contractions in the THC build step (especially the expensive fitting with the ERI) are more expensive the bigger N_{THC} is. The program was implemented using the CFOUR library[110]. Unfortunately, the authors did not disclose the hardware used in their benchmark.

Another similar approach was put forward by Bangerter, F. H.; Glasbrenner, M.; Ochsenfeld, C. [89] who used the DF expansion of the ERI in fitting matrix $\boldsymbol{E}$ to construct known intermediates

$$\begin{aligned}
\boldsymbol{E}^{RS} &= \boldsymbol{X}^R_{\mu\nu}(\mu\nu|K)(K|L)(L|\lambda\sigma)\boldsymbol{X}^S_{\lambda\sigma} \\
&= \left[\boldsymbol{X}^R_{\mu\nu}(\mu\nu|K)\right](K|L)\left[(L|\lambda\sigma)\boldsymbol{X}^S_{\lambda\sigma}\right] \\
&= \boldsymbol{Y}^R_K(K|L)\boldsymbol{Y}^S_L
\end{aligned} \tag{4.45}$$

which when used to form the center matrix can form another set of intermediates

$$\begin{aligned}
\boldsymbol{Z}^{PQ} &= \left(\left[\boldsymbol{S}^{-1}\right]^{PR}\boldsymbol{Y}^R_K\right)(K|L)\left(\boldsymbol{Y}^S_L\left[\boldsymbol{S}^{-1}\right]^{QS}\right) \\
&= \boldsymbol{\Lambda}^P_K(K|L)\boldsymbol{\Lambda}^Q_L.
\end{aligned} \tag{4.46}$$

We can use the pivoted Cholesky Decomposition of the metric matrix $\boldsymbol{S}$

$$\begin{aligned}
\boldsymbol{S} &= \boldsymbol{\Pi}\boldsymbol{L}\boldsymbol{L}^T \\
\boldsymbol{\Lambda} &= \left(\boldsymbol{\Pi}\boldsymbol{L}\boldsymbol{L}^T\right)^{-1}\boldsymbol{Y} \\
\left(\boldsymbol{\Pi}\boldsymbol{L}\boldsymbol{L}^T\right)\boldsymbol{\Lambda} &= \boldsymbol{Y}
\end{aligned} \tag{4.47}$$

to derive $\boldsymbol{\Lambda}$ using only two triangular solves while avoiding any pseudoinversion

$$\begin{aligned} \boldsymbol{\Pi}^T \boldsymbol{Y} &= \boldsymbol{L}\boldsymbol{W} \quad \text{solve for } \boldsymbol{W} \\ \boldsymbol{W} &= \boldsymbol{L}^T \boldsymbol{\Lambda} \quad \text{solve for } \boldsymbol{\Lambda}. \end{aligned} \tag{4.48}$$

The authors combined this improved handling of the metric matrix with advanced integral screening methods and reported that in empirical tests they were able to determine the MP2 SOS energies from Eq. (3.40) for systems with up to 9000 basis functions in under 12 hours using a 16-thread CPU. It should be noted that they also employed some other optimization techniques that are not in the scope of this thesis like Cholesky Orbitals and using DF with the so-called "erfc" kernel.

In two recent publications Hillers-Bendtsen, A. E.; Martínez, T. J. [95,96] proposed a method that is again or the LU Decomposition (see Section 2.3) similar to the ones described above. Their main goal was to lower the complexity of the SCF algorithm. Under the assumption that the auxiliary Coulomb integral $(K|L)$ captures most of the information contained in the ERI, they derive a least-squares fitted expression

$$\begin{aligned} &\arg\min_{\tilde{\boldsymbol{Z}}^{PQ}} \quad \frac{1}{2}\left\| (K|L) - \boldsymbol{X}_K^P \tilde{\boldsymbol{Z}}^{PQ} \boldsymbol{X}_K^Q \right\|_F^2 \\ &\Rightarrow \boldsymbol{Z}^{PQ} \approx \tilde{\boldsymbol{Z}}^{PQ} = \left(\boldsymbol{X}_K^P \boldsymbol{X}_K^R\right)^{-1} \boldsymbol{X}_K^R (K|L) \boldsymbol{X}_L^S \left(\boldsymbol{X}_L^S \boldsymbol{X}_L^Q\right)^{-1}. \end{aligned} \tag{4.49}$$

Since this step still involves the pseudoinversion of the metric matrix $\boldsymbol{X}_K^P \boldsymbol{X}_K^R$, the authors propose using the Cholesky Decomposition to decompose the auxiliary basis Coulomb matrix

$$(K|L) = \boldsymbol{L}_K^n \boldsymbol{L}_L^n \tag{4.50}$$

which can be used to reformulate the optimization problem to

$$\arg\min_{\boldsymbol{Y}_n^P} \quad \left\| \boldsymbol{L}_K^n - \boldsymbol{X}_K^P \boldsymbol{Y}_n^P \right\|_F^2. \tag{4.51}$$

Given this problem, a solution can be determined using

$$\boldsymbol{X}_K^P \boldsymbol{Y}_n^P = \boldsymbol{L}_K^n \quad \text{solve for } \boldsymbol{Y}. \tag{4.52}$$

This method to avoid pseudoinversion of the metric matrix is extremely similar to the method deployed by Bangerter, F. H.; Glasbrenner, M.; Ochsenfeld, C. [89], with the difference that it only involves integrals derived from auxiliary basis functions. Though this method's formal complexity amount to $\mathcal{O}\left(N_{\text{basis}}^3\right)$, the authors acknowledge that the resulting ERI approximation is numerically imprecise. For this reason they propose using a Superposition of Atomic Densities (SAD) initial guess[111] at the density matrix and then only use the THC ERI to converge the SCF calculation in the density-difference framework[94]. This does not require as high numeric precision. The authors report impressive results, being able to construct the THC for water clusters with up to 2500 basis function in under 10 seconds using an Nvidia A100 GPU and hand-optimized compute kernels with no integral screening. The method was built in the TerraChem quantum chemistry package[112].

4.7 Block Tensor Decomposition

In a recent paper Zhang, Y.; Xiong, X.; Wu, W.; Su, P. [105] proposed a new approach to compute the THC with formal $\mathcal{O}\left(N_{\text{basis}}^3\right)$ computational complexity, which is not derived from the LS-THC equations. Instead, the approach draws ideas from FMMs and merges them with the CD grid pruning procedures. They start with a dense integration grid (indexed by g, h). From this they derive a sparser grid (indexed by A, B). The Block Tensor Decomposition (BTD) uses a CD based

pruning but improves the scalability by using a Hilbert curve to group close points and form clusters by chunking. For each cluster, the highest pivot is determined by a pivoted CD. This provides a sparse grid, while maintaining the relationship from grid points to the chosen "cluster center".

Using this dual-grid approach, the THC center matrix can be expanded as

$$\begin{aligned} \boldsymbol{Z}^{PQ} &= \boldsymbol{K}_M^P \boldsymbol{K}_M^Q \\ \boldsymbol{K}_M^P &= [\boldsymbol{S}^+]_{KL} \left(\boldsymbol{X}_\mu^g \boldsymbol{X}_\mu^L\right)^2 \boldsymbol{B}_M^g \end{aligned} \tag{4.53}$$

where the electrostatic integral $\boldsymbol{B}$ is the separable DF approximation of the full-grid to full-grid Coulomb interaction

$$\boldsymbol{V}_{gh} = \frac{1}{\|\boldsymbol{r}_g - \boldsymbol{r}_h\|} \approx \boldsymbol{B}_M^g \boldsymbol{B}_M^h = (M|N)^{-\frac{1}{2}} \int \frac{\chi_N(\boldsymbol{r})}{\|\boldsymbol{r} - \boldsymbol{r}_g\|} \,\mathrm{d}\boldsymbol{r}. \tag{4.54}$$

In this building procedure, the most expensive contraction involves $N_{\text{basis}} \cdot N_{\text{grid}} \cdot N_{\text{aux}}$ operations, which implies a formal complexity of $\mathcal{O}\left(N_{\text{basis}}^3\right)$. The authors demonstrated the validity by combining it with a Schwarz-like integral screening and calculating SOS MP2 energy estimates. The authors reported that their implementation was able to build the THC representation for water clusters with up to ≈ 1000 basis functions in about one minute using only one CPU core. We would like to note that the results of the paper were not yet reproduced by other publications and the authors did not provide the implementation.

4.8 CP Decomposition

As described in Section 2.7, the rank of a tensor and its CANDECOMP/PARAFAC Decomposition are closely linked. It therefore seems quite natural that the CPD should be a good way to arrive at a low-rank representation of the ERI. In the paper proposing LS-THC, Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] note that the THC can be understood as a restricted form of the CPD. If we were to drop all constraints held on the $\boldsymbol{X}$ matrices, we could find a THC such that the center matrix becomes the identity matrix

$$(\mu\nu|\lambda\sigma) \approx \boldsymbol{X}_\mu^P \boldsymbol{X}_\nu^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_\lambda^Q \boldsymbol{X}_\sigma^Q = \boldsymbol{A}_\mu^r \boldsymbol{A}_\nu^r \,\mathbb{I}\, \boldsymbol{B}_\lambda^r \boldsymbol{B}_\sigma^r. \tag{4.55}$$

From this, we can recover the CPD with the weighting matrix contracted into the component matrices. In recent years, some researchers (notably Karl Pierce and colleagues) have been applying forms of the CPD. Notably, they applied the CPD not to the ERI directly but to tensor networks i.e., the full contraction that is desired in particular terms from the CCSD amplitude equations[8,10] as well as MP2[113].

The CPD is relevant in context of MP2 calculations since it can yield a contraction pattern for the K term that is more efficient than the one using the THC (see Section 4.3)[113]. From Eq. (4.24) we can see that the K term can be expressed as

$$K_{\text{RHF-MP2}} = \tau_v^a \tau_v^b \tau_v^i \tau_v^j \cdot \boldsymbol{X}_i^P \boldsymbol{X}_a^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_b^Q \boldsymbol{X}_j^Q \cdot \boldsymbol{X}_i^R \boldsymbol{X}_b^R \boldsymbol{Z}^{RS} \boldsymbol{X}_j^S \boldsymbol{X}_a^S. \tag{4.56}$$

If we swap one of the component THC expressions with the CPD, we can derive a contraction pattern that does not include any operations with higher complexity than $\mathcal{O}\left(N_{\text{basis}}^3\right)$

$$K_{\text{RHF-MP2}} = \boldsymbol{\tau}_v^a \boldsymbol{\tau}_v^b \boldsymbol{\tau}_v^i \boldsymbol{\tau}_v^j \cdot \boldsymbol{X}_i^P \boldsymbol{X}_a^P \boldsymbol{Z}^{PQ} \boldsymbol{X}_j^Q \boldsymbol{X}_b^Q \cdot \boldsymbol{A}_i^r \boldsymbol{A}_b^r \boldsymbol{B}_j^r \boldsymbol{B}_a^r$$
$$= \sum_{PQvr} \left[\boldsymbol{Z}^{PQ} \left(\sum_i \boldsymbol{\tau}_v^i \boldsymbol{X}_i^P \boldsymbol{A}_i^r \right) \left(\sum_a \boldsymbol{\tau}_v^a \boldsymbol{X}_a^P \boldsymbol{B}_a^r \right) \left(\sum_j \boldsymbol{\tau}_v^j \boldsymbol{X}_j^Q \boldsymbol{B}_j^r \right) \left(\sum_b \boldsymbol{\tau}_v^b \boldsymbol{X}_b^Q \boldsymbol{A}_b^r \right) \right] \tag{4.57}$$

While this is a desirable outcome, computing the CPD efficiently is still a major issue, due to the non-linear nature of the problem. THC reduces this issue to defining a projection space and finding the center matrix. This can often be done using a single operation. For the CPD one needs to employ heuristic, iterative techniques like ALS. Additionally, the THC works by importing a certain bias like the choice of grid or the auxiliary basis if fit with DF. While this can be problematic (see Section 4.2), it also allows us to never construct the full ERI.

Some publications[8,10,113,114] try to mitigate this problem by using substitutes to the full ERI in their fitting procedure, and by implementing their ALS optimization in a "matrix-free" way. This means that the CPD optimization problem from Eq. (2.44) for the ERI becomes

$$\underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min} \quad \frac{1}{2} \left\| \boldsymbol{L}_{\mu\nu}^K \boldsymbol{L}_{\lambda\sigma}^K - \boldsymbol{A}_i^r \boldsymbol{A}_a^r \boldsymbol{B}_j^r \boldsymbol{B}_b^r \right\|^2 \tag{4.58}$$

which can be solved $\approx \left(r \cdot N_{\text{basis}}^4\right)$ operations using ALS, since each ALS cycle needs a contraction over the DF factors $\boldsymbol{L}$. This computational burden can be lowered to $\mathcal{O}\left(r \cdot N_{\text{basis}}^3\right)$ if the fit is constructed against the THC build by a cubic-scaling algorithm. The publications show that the error propagation of this chaining approach is quite low. We can even construct a more expensive robust fitting procedure[8] that yields more faithful fits to the chosen ERI stand-in. The performance of those methods is lower-bounded by the underlying ERI approximation. Therefore, there either must be a fast way to construct another approximation like the THC or one must hope that the better contraction properties outweigh the longer build-time. The authors did not test their implementation with an actual cubic-scaling THC implementation, but "simulated" this using a LS-THC procedure. It should also be noted that formally the CPD does not conserve the symmetry properties of the ERI, though it was shown that the error margin is quite small in practical applications.

4.9 Closing Remarks

At this point it should have become obvious that the field of low-rank approximations for problems involving the ERI is wide and nuanced. Different approaches were developed, usually by researchers trying to improve the runtime efficiency of one or many specific simulation methods. This led to a multitude of angles from which the problem was attacked and produced remarkable progress.

It also created a landscape of methods that lacks clarity and comparability. Each new idea seems to be implemented in a different software library, which means testing newly proposed methods might require adopting a new software, which takes time and mental effort. Often, the implementations are not even publicly available anymore, or they are written using different programming languages or software libraries. Additionally, while the benchmarking conducted by the researchers testing their new implementations is mostly quite thorough, they usually lack information and were conducted in hugely differing environments. Especially differences in hardware (i.e., how many CPUs were used), additional software (i.e., the DF implementation that was used in comparisons) as well as other optimizations that might or might not have been applied make it almost impossible to compare two methods based on literature review.

Those problems might be attributed to the time ranges that the research was conducted over. It follows logically that an older method can not compare itself to something that was developed later in time, but as we saw in the literature review: Researchers seldomly compare a newly developed method against other "competing" ideas, which might be due to the fragmentation issues outlined above. The only baseline that seems to exist is the DF/RI method, since it is already widely adopted in the field and available in most quantum chemistry libraries.

This observation was the motivation for this thesis: There should exist a comparative review of the distinctive low-rank approximation approaches involving the ERI. Therefore, the following chapters will discuss the implementation and comparative benchmarking of some of the described methods. The implementation should be as accessible as possible and will therefore be developed using the open source Python framework PySCF[73] as well as standard numerical computing utilities. We will not use any additional optimizations like integral screening, to facilitate a "fair" comparison. All methods will be tested on a unified set of molecules.

5 Implementation

To address the issues outlined in the previous chapters, we developed the software library `libthc`. It implements the basic infrastructure to build ERI approximations as well as five of the THC building approaches presented in Section 4. The basis for the implementations is provided by the quantum chemistry package PySCF[73], which was chosen because its code is available under a permissive Apache-2.0 license, and it uses the accessible Python programming language. We chose to implement the following methods, as we aimed to compare the mathematically distinct ways to compute the THC approximation:

- **LS-THC**: The standard least-squares derivation approach (Section 4.4)
- **CD-LS-THC**: LS-THC with Cholesky Decomposition based grid pruning (Section 4.6)
- **ISDF-QR**: ISDF with sketchy pivoting using QRCP (Section 4.5)
- **ISDF-KM**: ISDF using K-Means clustering (Section 4.5)
- **DF-THC**: Density fitting approximated THC (Section 4.6)

In the following sections, we will delineate the implementations in pseudocode. We will also outline the parameters of each method. This should provide a baseline for other research teams to validate and reproduce the results achieved in this thesis. The actual implementations use PySCF, NumPy[115] and SciPy[116]. For all integrals, we used the PySCF module wrapping libcint[117]. All tensor contractions were computed with the PyTBLIS library[82].

It should be noted that we did not implement DF, BTD or CPD based approximations. For DF, we used the mature implementation in PySCF. Some numerical experiments were conducted with BTD, but the calculation of the electrostatic integral is not supported in libcint yet. While CPD based improvements for quantum calculations seem promising, the fastest ways to derive the CPD are based upon the THC and DF approximations. Therefore, we decided that arriving at a "production-ready" THC implementation provides the biggest immediate benefit to the field.

LS-THC: As one of the simplest ways to compute the THC, we implemented the method suggested by Hohenstein, E. G.; Fales, B. S.; Parrish, R. M.; Martínez, T. J. [92]. Algorithm 5 outlines the procedure. We start by constructing the Becke grid. The actual implementation uses the DFT grid module from PySCF to construct the grid. The most expensive operation of Algorithm 5 is the contraction with the ERI or the DF factors. The procedure is displayed for the AO case, but can easily be adapted to the MO basis by only contracting $\boldsymbol{X}$ with the MO coefficients and fitting against the MO transformed ERI (or the respective DF factors). The only tunable parameter used in the original publication was the threshold in the pseudoinversion procedure. Since the procedure is always solving the full eigenvalue problem on $\boldsymbol{S}$, this parameter can be set to a low value like $1 \cdot 10^{-10}$; therefore, we will not discuss it further. Alternatively, the algorithm might be tuned by using specialized grids with lower density.

CD-LS-THC: The improved version of LS-THC suggested by Matthews, D. A. [106] is conceptually very similar to the procedure from Algorithm 5. The added step consists of the Cholesky Decomposition of the metric matrix $\boldsymbol{S}$. This is achieved in our implementation using the Accelerated-RPCholesky procedure proposed by Epperly, E. N.; Tropp, J. A.; Webber, R. J. [36] (see sec-cholesky). The actual code does not instantiate the full $\boldsymbol{S}$ matrix but uses a submatrix function which computes parts of $\boldsymbol{S}$ on the fly. This leads to a very low empirical computational pre-factor and linearly growing memory usage. The parameter λ controls the variance cutoff on the diagonal, which will be used in the pivoting. The CD procedure provides the Cholesky factors as well as the numerical rank and sequence of the pivot columns. From here we can select a new "pruned"

Algorithm 5: Least Squares THC

procedure LS-THC(Basis functions φ_μ)
  $\boldsymbol{g}, \boldsymbol{w} \leftarrow$ build Becke grid and weights
  $\boldsymbol{X}^P_\mu \leftarrow \sqrt[4]{w_P \cdot \varphi_\mu(g_P)}$ ▷ Multiply with MO factors if needed
  $\boldsymbol{X}^P_{\mu\nu} \leftarrow \text{contract}\left(\boldsymbol{X}^P_\mu \boldsymbol{X}^P_\nu\right)$
  $\boldsymbol{S}^{PQ} \leftarrow \text{contract}\left(\boldsymbol{X}^P_{\mu\nu} \boldsymbol{X}^Q_{\mu\nu}\right)$
  $\boldsymbol{S}^{+} \leftarrow \text{pseudoinvert}(\boldsymbol{S})$

  if density fitting enabled **then**
    $\boldsymbol{L}^K_{\mu\nu} \leftarrow$ get DF cholesky factors ▷ Transform to MO basis if needed
    $\boldsymbol{Y}^P_K \leftarrow \text{contract}\left(\boldsymbol{L}^K_{\mu\nu} \boldsymbol{X}^P_{\mu\nu}\right)$
    $\boldsymbol{E}^{PQ} \leftarrow \text{contract}\left(\boldsymbol{Y}^P_K \boldsymbol{Y}^Q_K\right)$
  else
    $\boldsymbol{E}^{PQ} \leftarrow \text{contract}\left(\boldsymbol{X}^P_{\mu\nu} (\mu\nu|\lambda\sigma) \boldsymbol{X}^Q_{\lambda\sigma}\right)$ ▷ Alternatively use MO transformed ERI
  end

  $\boldsymbol{Z}^{PQ} \leftarrow \text{contract}(\boldsymbol{S}^{+} \boldsymbol{E} \boldsymbol{S}^{+})$

  return $\boldsymbol{X}, \boldsymbol{Z}$
end

version of the $\boldsymbol{X}$ matrix we call $\boldsymbol{X}'$. This matrix is used throughout the rest of the algorithm. The full procedure is outlined in Algorithm 6.

ISDF-QR: The ISDF with QRCP procedure in Algorithm 7 represents a mixture of the method described by Lu, J.; Ying, L. [7] and the adapted version by Lee, J.; Lin, L.; Head-Gordon, M. [103]. As a first step, the sketching matrix is built the same way that was suggested by the original publication, with the addition of including the grid weights. The sketching matrix is decomposed using a QRCP procedure, which yields the interpolation points as its pivots. Similar to CD-LS-THC, we derive a pruned version of the $\boldsymbol{X}$ matrix and work with that. From there we pseudoinvert the resulting pruned version of the metric matrix and use it to calculate the $\boldsymbol{Z}$ matrix by fitting on the coulomb integrals of an auxiliary basis set. ISDF-QR takes two parameters: The pruning error cutoff ε is used to identify significant grid points, which have a higher diagonal value in $\boldsymbol{R}$. The r value determines the degree of truncation imposed by the sketching. The process will retain $N_{\text{basis}} \cdot r$ rows in the sketching matrix, reducing the dimensions of $\boldsymbol{M}$ from N^2_{basis} to $N_{\text{basis}} \cdot r$. It should be noted that we could select any number of rows from the sketching matrix. The scaling with N_{basis} ensures that the parameter grows with system size and does not require manual intervention.

ISDF-KM: In the improved version of ISDF proposed by Lee, J.; Lin, L.; Head-Gordon, M. [103], the QRCP method is replaced by the K-Means clustering procedure suggested by Dong, K.; Hu, W.; Lin, L. [102]. The algorithm is sketched out in Algorithm 8. The algorithm determines the cluster centers for each atom's Becke grid in turn. Unfortunately, we have to connect a weight to the grid point, and we can not trivially compute a new Becke weight that is consistent with the rest of the existing grid. Therefore, we chose the interpolation points as the subset of points in the parent grid

Algorithm 6: **Cholesky Decomposition pruned LS-THC**

procedure CD-LS-THC(Basis functions φ_μ, Cholesky threshold λ)
  $\boldsymbol{g}, \boldsymbol{w} \leftarrow$ build Becke grid and weights
  $\boldsymbol{X}_\mu^P \leftarrow \sqrt[4]{w_P \cdot \varphi_\mu(g_P)}$ ▷ Multiply with MO factors if needed
  $\boldsymbol{X}_{\mu\nu}^P \leftarrow \text{contract}\big(\boldsymbol{X}_\mu^P \boldsymbol{X}_\nu^P\big)$
  $\boldsymbol{S}^{PQ} \leftarrow \text{contract}\big(\boldsymbol{X}_{\mu\nu}^P \boldsymbol{X}_{\mu\nu}^Q\big)$

  ▷ Returns $\boldsymbol{S} \approx \boldsymbol{R}^T\boldsymbol{R}$, permutation matrix $\boldsymbol{\Pi}$ and numerical rank k
  $\boldsymbol{R}, \boldsymbol{\Pi}, k \leftarrow$ Accelerated-RPCholesky($\boldsymbol{S}, \lambda$)
  $\boldsymbol{X}' \leftarrow (\boldsymbol{\Pi}\boldsymbol{X})_\mu^{1:k}$

  if density fitting enabled **then**
    $\boldsymbol{L}_{\mu\nu}^K \leftarrow$ get DF cholesky factors ▷ Transform to MO basis if needed
    $\boldsymbol{Y}_K^P \leftarrow \text{contract}\big(\boldsymbol{L}_{\mu\nu}^K \boldsymbol{X}_{\mu\nu}'^P\big)$
    $\boldsymbol{E}^{PQ} \leftarrow \text{contract}\Big(\boldsymbol{Y}_K^P \boldsymbol{Y}_K^Q\Big)$
  else
    $\boldsymbol{E}^{PQ} \leftarrow \text{contract}\Big(\boldsymbol{X}_{\mu\nu}'^P (\mu\nu|\lambda\sigma) \boldsymbol{X}_{\lambda\sigma}'^Q\Big)$ ▷ Alternatively use MO transformed ERI
  end

  $\boldsymbol{A} \leftarrow \text{solve}(\boldsymbol{R}^T, \boldsymbol{E})$
  $\boldsymbol{B}^T \leftarrow \text{solve}(\boldsymbol{R}^T, \boldsymbol{A}^T)$
  $\boldsymbol{C} \leftarrow \text{solve}(\boldsymbol{R}, \boldsymbol{B})$
  $\boldsymbol{Z}^T \leftarrow \text{solve}(\boldsymbol{R}, \boldsymbol{C}^T)$

  return $\boldsymbol{X}', \boldsymbol{Z}$
end

that are closest to the cluster centroids. To efficiently solve this problem, we employ a k-d-Tree structure[118]. This binary tree can be constructed in $\mathcal{O}(n \cdot (k + \log(n)))$ ($n = N_{\text{grid}}, k = 3$), occupies $\mathcal{O}(n \cdot k)$ memory and allows querying for nearest neighbors in $\mathcal{O}(\log n)$ time. For the weighted K-Means procedure, we compute the normalized squared amplitude of the AO wave-functions. We do not use the Becke weights directly like the original publication, as it led to ill-behaved calculations. We also decided against using the same parameter as the original publication. They use the size of the auxiliary basis, split evenly between atoms as the amount of centroids. In preliminary tests, this approach produced high error rates for systems with an uneven ratio of elements (i.e., more hydrogen than carbon). We attribute this failure to the oversampling of the hydrogen atoms, which stems from all atom types being assigned the same number of cluster centers. We therefore employed a different strategy, where the number of clusters for an atom is determined by the size of the atom grid (PySCF uses different densities for each element) scaled by a value k.

DF-THC: As put forward by Hillers-Bendtsen, A. E.; Martínez, T. J. [96], the THC can also be computed by approximating the $\boldsymbol{Z}$ matrix with the center DF Coulomb integral $(K|L)$. We named this approach Density Fitting THC (DF-THC) and the pseudocode can be found in Algorithm 9. The method is fairly straightforward, especially when compared to the more complex ISDF

Algorithm 7: ISDF QRCP

procedure ISDF-QR(Basis functions φ_μ, Pruning threshold ε, Sketching rows r)
  $\boldsymbol{g}, \boldsymbol{w} \leftarrow$ build Becke grid and weights
  $\boldsymbol{X}_\mu^P \leftarrow \sqrt[4]{w_P \cdot \varphi_\mu(g_P)}$
  $\boldsymbol{K}_{\overline{\mu\nu}}^P \leftarrow \text{contract}\left(\boldsymbol{X}_\mu^P \boldsymbol{X}_\nu^P\right)$
  $\eta_{\overline{\mu\nu}} \leftarrow$ random-unit-complex()
  $\widetilde{\boldsymbol{K}} \leftarrow \eta \boldsymbol{K}$ ▷ Multiply each row by a random complex number
  $\boldsymbol{F} \leftarrow \text{FFT}\left(\widetilde{\boldsymbol{K}}\right)$ ▷ Column-wise FFT
  $\boldsymbol{M} \leftarrow$ take $N_{\text{basis}} \cdot r$ rows randomly from $\boldsymbol{F}$

  $\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{\Pi} \leftarrow \text{QRCP}(\boldsymbol{M})$
  $\boldsymbol{d} \leftarrow \text{diag}(\boldsymbol{\Pi})$
  $k \leftarrow \text{len}(\boldsymbol{d})$ where $\boldsymbol{d} > (\boldsymbol{d}_1 \cdot \varepsilon))$ ▷ Determine numerical rank
  $\boldsymbol{X}_\mu'^P \leftarrow (\boldsymbol{\Pi}\boldsymbol{X})_\mu^{1:k}$ ▷ Find interpolation points
  $\boldsymbol{S}^{PQ} \leftarrow \text{contract}\left(\boldsymbol{X}_{\mu\nu}'^P \boldsymbol{X}_{\mu\nu}'^Q\right)$
  $\boldsymbol{S}^+ \leftarrow \text{pseudoinvert}(\boldsymbol{S})$

  Build auxiliary basis χ_K
  $\boldsymbol{G}_{KL} \leftarrow$ compute 2-center integrals $(K|L)$
  $\boldsymbol{G}_{\mu\nu}^K \leftarrow$ compute 3-center integrals $(\mu\nu|K)$ ▷ Transform to MO basis if needed
  $\boldsymbol{B}_K^P \leftarrow \text{contract}\left(\boldsymbol{G}_{\mu\nu}^K \boldsymbol{X}_{\mu\nu}'^Q [\boldsymbol{S}^+]^{PQ}\right)$
  $\boldsymbol{D}_L^P \leftarrow \text{contract}\left(\boldsymbol{B}_K^P \left[\boldsymbol{G}^{-\frac{1}{2}}\right]_{KL}\right)$
  $\boldsymbol{Z}^{PQ} \leftarrow \boldsymbol{D}_L^P \boldsymbol{D}_L^Q$

  return $\boldsymbol{X}', \boldsymbol{Z}$
end

procedures. This approach would be quite costly (it is one of the expensive operations in LS-THC). We avoid the inversion by using the Cholesky Decomposition of the Coulomb integral. This allows us to use linear least-squares solving techniques, which are numerically stable and readily available in NumPy. The parameters that can be tuned are the choice of the auxiliary basis set and the threshold for the pivoted CD.

Algorithm 8: ISDF with K-Means Clustering

procedure ISDF-KM(Basis functions φ_μ, Cluster multiplier k)
  $\boldsymbol{X}' \leftarrow \emptyset$
  for $a \in N_{\text{atoms}}$ **do**
    $\boldsymbol{g}, \boldsymbol{w} \leftarrow$ build atomic Becke grid
    $\mathcal{T} \leftarrow$ k-d-Tree(g)
    $\boldsymbol{X}_\mu^P \leftarrow \sqrt[4]{w_P \cdot \varphi_\mu(g_P)}$
    $\boldsymbol{\rho}^P \leftarrow \sum_\mu \left(\boldsymbol{X}_\mu^P\right)^2$
    $\boldsymbol{\rho}^P \leftarrow \boldsymbol{\rho}^P / \sum_P \boldsymbol{\rho}^P$
    $n_{\text{cluster}} \leftarrow k \cdot \text{len}(g)$

    ▷ Perform clustering on grid coordinates with density as weight
    $\boldsymbol{c} \leftarrow$ K-Means$(n_{\text{clusters}}, \boldsymbol{g}, \boldsymbol{d})$
    $\boldsymbol{g}' \leftarrow$ Nearest-Neighbors$(\mathcal{T}, \boldsymbol{c})$ ▷ Query for points closest to cluster centroids
    $\boldsymbol{X}' \leftarrow \boldsymbol{X}' \cup \left(\boldsymbol{X}_\mu^P \text{ where } P \in g'\right)$
  end

  $\boldsymbol{S}^{PQ} \leftarrow \text{contract}\left(\boldsymbol{X}'^P_{\mu\nu} \boldsymbol{X}'^Q_{\mu\nu}\right)$
  $\boldsymbol{S}^+ \leftarrow \text{pseudoinvert}(\boldsymbol{S})$

  Build auxiliary basis χ_K
  $\boldsymbol{G}_{KL} \leftarrow$ compute 2-center integrals $(K|L)$
  $\boldsymbol{G}^K_{\mu\nu} \leftarrow$ compute 3-center integrals $(\mu\nu|K)$ ▷ Transform to MO basis if needed
  $\boldsymbol{B}^P_K \leftarrow \text{contract}\left(\boldsymbol{G}^K_{\mu\nu} \boldsymbol{X}'^Q_{\mu\nu} [\boldsymbol{S}^+]^{PQ}\right)$
  $\boldsymbol{D}^P_L \leftarrow \text{contract}\left(\boldsymbol{B}^P_K \left[\boldsymbol{G}^{-\frac{1}{2}}\right]_{KL}\right)$
  $\boldsymbol{Z}^{PQ} \leftarrow \boldsymbol{D}^P_L \boldsymbol{D}^Q_L$

  return $\boldsymbol{X}', \boldsymbol{Z}$
end

Algorithm 9: Density Fitting THC

procedure DF-THC$\big($Basis functions φ_μ, Cholesky threshold $\lambda\big)$
  $\boldsymbol{g}, \boldsymbol{w} \leftarrow$ build Becke grid and weights
  Build auxiliary basis χ_K

  $\boldsymbol{X}^P_\mu \leftarrow \sqrt[4]{w_P \cdot \varphi_\mu(g_P)}$ ▷ Multiply with MO factors if needed
  $\boldsymbol{G}_{KL} \leftarrow$ compute 2-center integrals $(K|L)$
  $\boldsymbol{A}^P_K \leftarrow \sqrt{w_P \cdot \chi_K(g_P)}$

  $\boldsymbol{L}, \boldsymbol{\Pi}, k \leftarrow$ Cholesky-Decompose$(\boldsymbol{G}, \lambda)$
  $\boldsymbol{Y}^T \leftarrow$ solve$\big(\boldsymbol{A}^T, (\boldsymbol{\Pi}\boldsymbol{L})\big)$
  $\boldsymbol{Z}^{PQ} \leftarrow$ contract$\Big(\boldsymbol{Y}^P_C \boldsymbol{Y}^Q_C\Big)$

  return $\boldsymbol{X}, \boldsymbol{Z}$
end

6 Experimental Validation

One of the stated goals of this thesis was to fairly compare the respective approximation methods in a unifying benchmark. For this purpose, we derived a benchmarking method inspired by the validation techniques used by Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] and Matthews, D. A. [106] that should fulfill the following goals:

1. Test a varied range of systems (see Figure 1)
2. Check the numerical and chemical validity of the approximations
3. Show the empirical runtime scaling of the different methods.

As test-data molecules we chose sets of alkane chains, linear acenes or fused benzene rings and water clusters. These structures involve the common elements carbon, hydrogen, and oxygen. They can be extended systematically, which makes them perfectly suited for analyzing the computational complexity our method exhibits with growing system size. They also possess different geometries, the alkanes being stretched along one dimension, the acenes being somewhat two-dimensional and the water clusters forming dense three-dimensional clusters. We generated sets of molecules for different system sizes, which we will denote with N (i.e., an alkane chain with $N = 1$ is just methane). The alkanes and acenes were built using the chemo-informatics framework RDKit[F.4]. The structures were optimized using a classical force field. The water clusters were generated using PACKMOL[119]. The packing algorithm was configured to keep a steady density in the cluster. All software versions used are specified in Appendix C.

For each of the chosen systems, we start the benchmark by running the SCF algorithm to form a converged density matrix. From this we call the method under test and build the THC matrices. To verify the validity of our approximations, we chose to calculate the electron-electron repulsion term from Eq. (4.22) which we will call HF-JK, as well as the MP2 energy correction. This allows us to test approximations both in the AO and MO basis. It should also provide more insight about the chemical relevance of the approximation compared to a more brute-force metric like the ERI norm-distance. We targeted the correlation-consistent double zeta basis cc-pVDZ[85]. It is big enough for calculations like MP2, and it is used by most of the other publications examined in Section 4. As auxiliary basis we used cc-pVDZ-RI[120] for all MP2 calculations and cc-pVDZ-JKfit[121] for the HF-JK calculations. Each basis is optimized for the targeted calculation. All computations used double precision (64 bit) floating point numbers.

We compare energies resulting from our calculations to canonical energies computed using PySCF. We used the PySCF implementation of DF, MP2 and HF throughout all tests. We will explicitly state if we are comparing our measured result against the canonical value or a result computed using DF. For the error measurements, we will use absolute value in Hartree as well as relative



Figure 1: Alkane chains, linear acenes (fused benzene rings), water clusters used for testing.

[F.4] https://www.rdkit.org/

error percentages. Since the absolute values of HF-JK and MP2 vary by 3 to 4 orders of magnitude, the relative error often works better to judge approximation performance. Wall times will be given in seconds and are calculated using the Python built-in utilities. All graphics were created using Pandas[122] as well as the plotnine[F.5] library.

For this work the HPC-cluster Hummel-2 at University of Hamburg was used. The cluster was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 498394658. All calculations were run on a 96-Core AMD EPYC 9654 CPU having access to 384GB of RAM. We configured all NumPy, SciPy, TBLIS and scikit-learn procedures to use the full amount of available cores wherever it was possible to do so.

6.1 Parameter Search

As we outlined in Section 5, most of the algorithms we want to test provide parameters to tune them to the desired calculation. The first task at hand is therefore to find a set of parameters that will be used in the course of the benchmark. Since most of the parameters are numbers, an exhaustive search for the optimal parameter value is impossible and one must resort to sampling. We chose value ranges based on the suggested values in the literature (if they exist) and calculated our target metrics (HF-JK and MP2 energies) for a single benzene molecule (linear acene $N = 1$). We aim to find parameter values that properly converge the errors for our target cc-pVDZ basis. We also included calculations for the bigger cc-pVTZ basis in order to verify that the observed parameter behavior is independent of the basis set used. To judge whether the approximation is successful, we calculated reference values with and without applying DF. Since we use the DF integrals for fitting, we expected to see the methods converging at a similar error level as the DF approximation. All error plots in the following section will indicate the error made by DF-MP2 or DF-HF-JK as a dashed line. Comparisons are made against the canonical value.

LS-THC: While there is no direct numerical parameter for this method, we should still be able to increase the approximation quality by using a denser grid. For this reason, we implemented multiple grid pruning strategies that decrease the density of the PySCF grid, which is very dense even on the lowest setting (≈ 500 points per atom). In Figure 2 we show how a denser grid affects the observed energies. Since we fit to the DF ERI it is clear that the default pruned grid (P1) as well as the grid with no pruning at all (P0) provided good approximations in both calculations. As implementing manual pruning is a delicate process, and we would like to observe the effectiveness of automated grid pruning, we will use the default pruning strategy of PySCF (P1) during our tests.

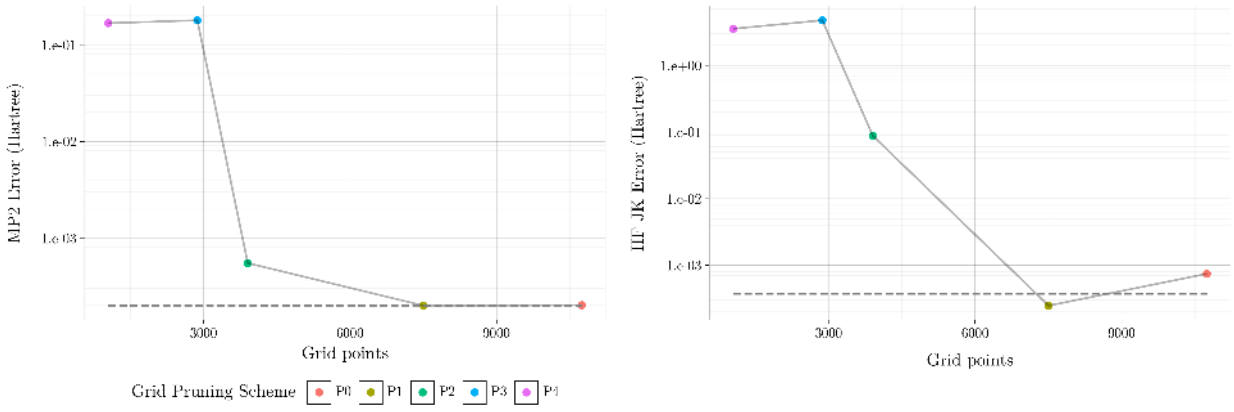


Figure 2: Effects of the grid density on the LS-THC error metrics for a benzene molecule (cc-pVDZ).

[F.5] https://plotnine.org/

CD-LS-THC: The only tunable parameter is the pruning threshold of the pivoted CD. From Figure 3 we can see that the smaller we set the value, the smaller the approximation error becomes. This is the expected result since smaller values will lead to more grid points being selected, increasing the accuracy of the quadrature. Therefore, we chose $\lambda = 1 \cdot 10^{-8}$ as the threshold value. It seemed to offer satisfactory convergence in cc-pVDZ. This value does not fully agree with the $\lambda = 1 \cdot 10^{-6}$ suggested by Matthews, D. A. [106], the probable reason being that the publication did not consider the AO basis and the suggested value also converged cc-pVDZ in the MP2 calculation. The cc-pVTZ results imply that for bigger basis sets a less aggressive pruning parameter should be chosen.

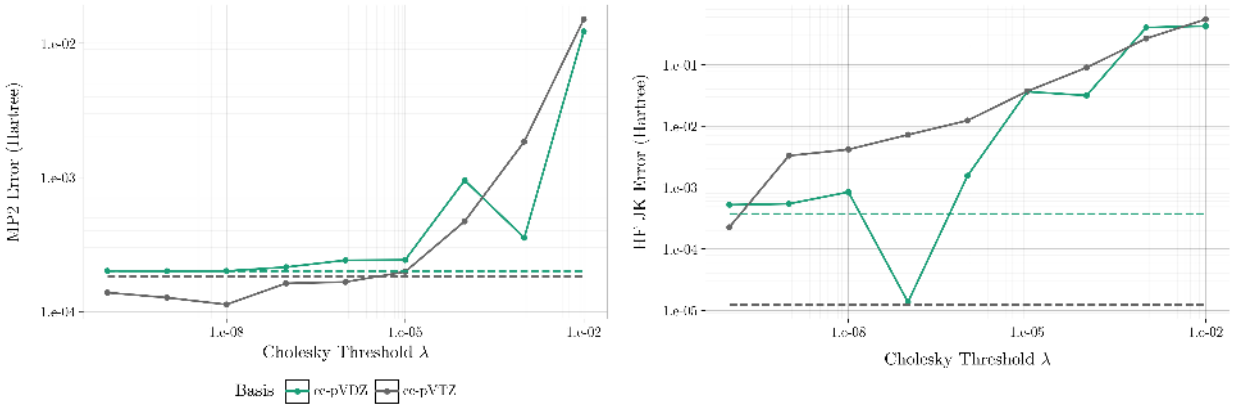


Figure 3: Effects of the CD-LS-THC cholesky pruning threshold λ on the error metrics for a benzene molecule (cc-pVDZ, cc-pVTZ).

ISDF-KM: For the cluster multiplier parameter, we can see a clear effect on the approximation quality. The PySCF atomic grids without pruning are dense ($\approx 10,000$ points per carbon atom) and the algorithm must invert a $n_{\text{centroid}} \times n_{\text{centroid}}$ matrix. Therefore, we chose a low sample range of only $0.1\% - 5\%$ of the parent atomic grid. This parameter range yielded metric matrices of decent size as well as sufficient approximation performance. For our benchmarks, we chose $k = 2\%$ for both the MO and the AO basis, as we saw the method converge the error in cc-pVDZ using this setting (Figure 4).

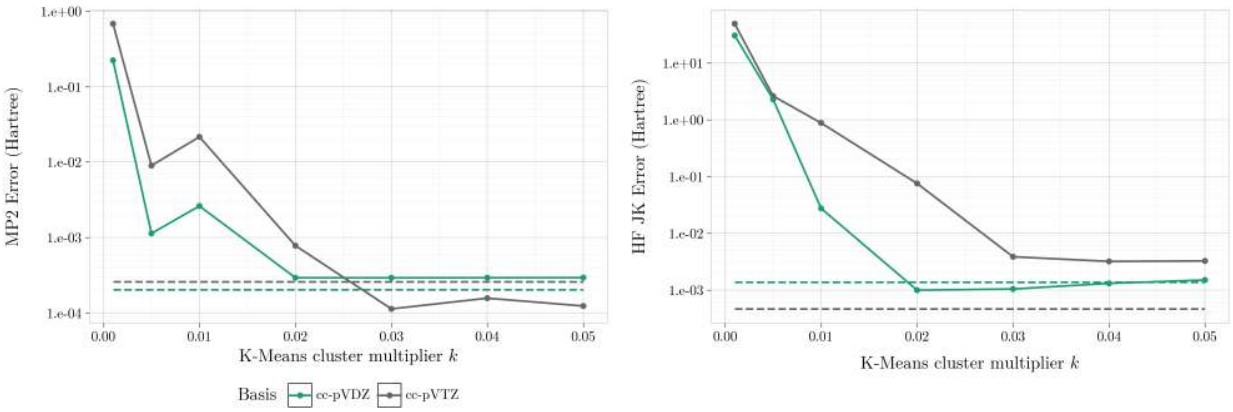


Figure 4: Effects of the ISDF-KM cluster multiplier k on the error metrics for a benzene molecule (cc-pVDZ, cc-pVTZ).

ISDF-QR: This algorithm possesses multiple tunable parameters. This makes the parameter search more difficult, since we would have to consider the product space of both the QR pruning threshold ε and the sketching parameter r. Therefore, we considered the parameters to be independent of one another and started from values provided by literature (Lu, J.; Ying, L. [7] suggest $\varepsilon = 1 \cdot 10^{-6}$ as well as $r = 20$). In our tests we found both values to work well for our calculations.

The resulting error metrics are displayed in Figure 5. We can see that both parameters have a clear impact on the quality of our approximations, which diminishes at a certain point. We therefore chose to use $\varepsilon = 1 \cdot 10^{-6}$ and $r = 20$ for our benchmark.

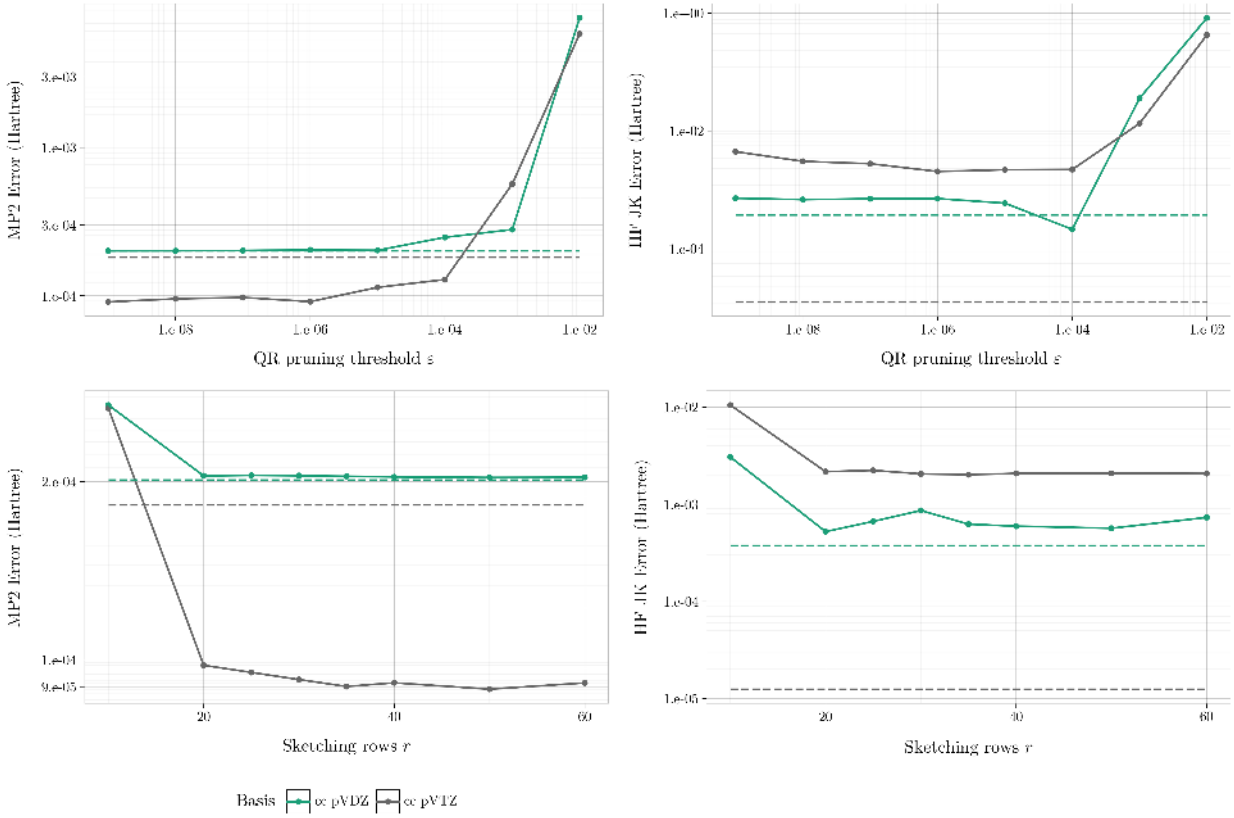


Figure 5: Effects of the ISDF-QR sketching and pruning parameters the error metrics for a benzene molecule (cc-pVDZ, cc-pVTZ). For each of the measurements, the second parameter was set to $\varepsilon = 1 \cdot 10^{-6}$ and $r = 35$ respectively.

DF-THC: We wanted to observe whether we could improve our approximation without tuning the grid density. Therefore, we altered the auxiliary basis set that was used to approximate the center matrix. We tested correlation consistent basis sets of increasing size including augmented basis sets that add more diffuse functions. We can see from Figure 6 that while the error decreased with more auxiliary basis functions, they seem to be detrimental to the HF-JK calculation. We can also confirm the observation of Hillers-Bendtsen, A. E.; Martínez, T. J. [96] that it produces results of lower fidelity than our other methods, especially in the AO basis. For the following tests, we chose the aug-cc-pVDZ-RI basis set. It is one of the smallest sets we tested, which should lower the computational burden in the triangular solve, and it yielded one of the lowest errors in the MP2 and HF-JK calculations. Since the method also involves a pivoted CD, we also tested several threshold values and found the error converging from vales of $1 \cdot 10^{-6}$ onward. We still decided to use a higher value $\lambda = 1 \cdot 10^{-10}$ since we wanted to come as close as possible to the original publication which used a full (non-pivoted) CD.

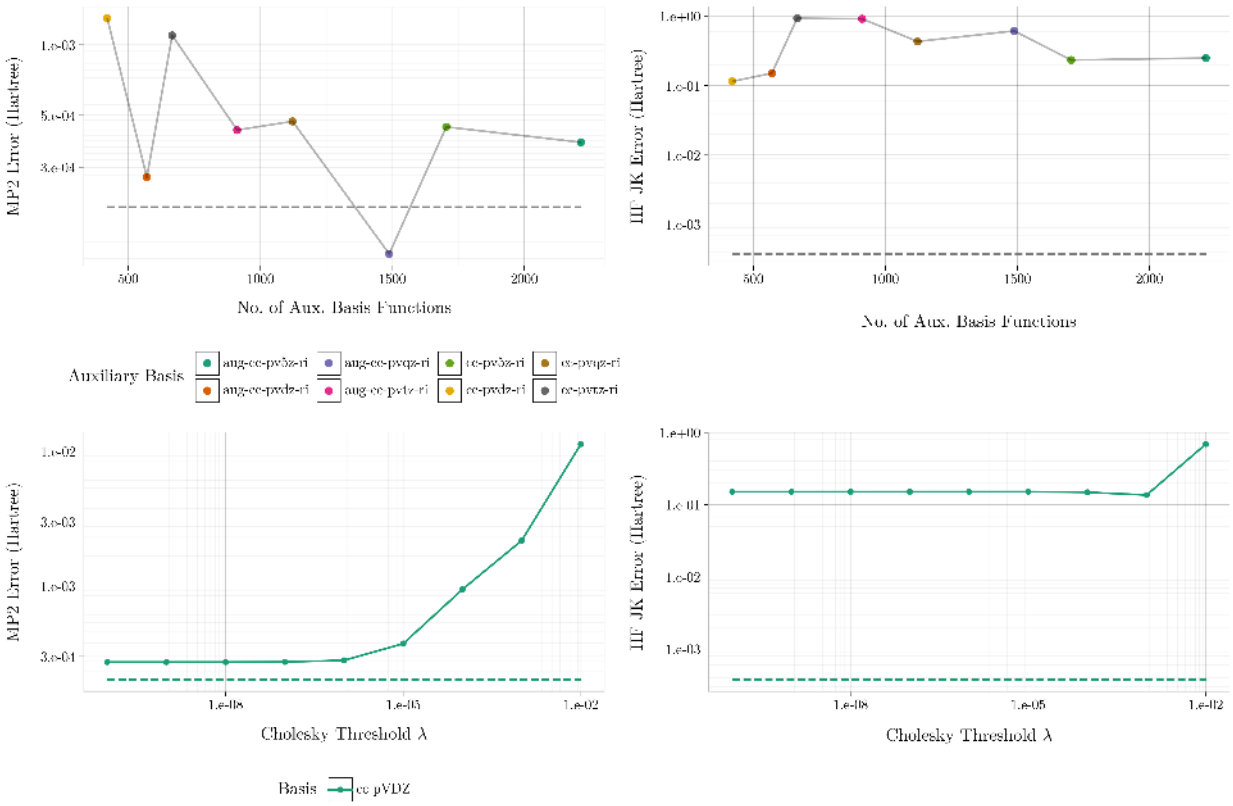


Figure 6: Effects of the DF-THC parameters set on the error metric for a benzene molecule (cc-pVDZ).

Overall, we can conclude that all approximations (except DF-THC) seem to be capable of producing approximations with a similar fidelity as DF. This is an encouraging result, since DF is widespread and its error levels are low enough for many real-world applications. All chosen parameters are presented in Table 1.

Table 1: Parameters chosen in the parameter search that will be used in the method benchmarking.

Approximation Method	Parameter	Value
LS-THC	Grid density	Default pruning
CD-LS-THC	Cholesky Threshold λ	$1 \cdot 10^{-8}$
ISDF-KM	Cluster multiplier k	0.02
ISDF-QR	Pruning threshold ε	$1 \cdot 10^{-6}$
	Sketching rows r	20
DF-THC	Auxiliary Basis	aug-cc-pVDZ-ri
	Cholesky Threshold λ	$1 \cdot 10^{-10}$

6.2 Full Comparative Benchmark on Small Systems

With the chosen set of parameters we computed the approximations and the energy metrics on our test-data. We wanted to measure the overall runtime of the procedures as well as their accuracy. We limited our test-data to systems that would amount to $N_{\text{basis}} < 700$ functions in the cc-pVDZ basis. This amount was enough to cover a variety of geometries while converging the scaling close to the empirical asymptotic regimes we expected for the methods. In the following paragraphs we will present the results for each method as well as a comparative overview. For each method we

collected data on the errors made compared to the canonical and DF versions of the calculation as well as the full time it took to compute the THC approximation. Our timings do not include the runtime of the SCF algorithm. To derive an estimated exponent for the observed empirical scaling, we used a linear regressor on the logarithmized dataset. We used a weighted L2-regularized regression model with the normalized amount of basis functions as weights. This was done in order to compensate for the non-representative nature of small systems. On the following pages we will present the results of this benchmark. We will only show the comparative plots here. Plots for each of the methods with the fitting curve derived by the regressor can be found in Appendix A. The MP2 energies were computed without Laplace quadrature to isolate the error made by THC.

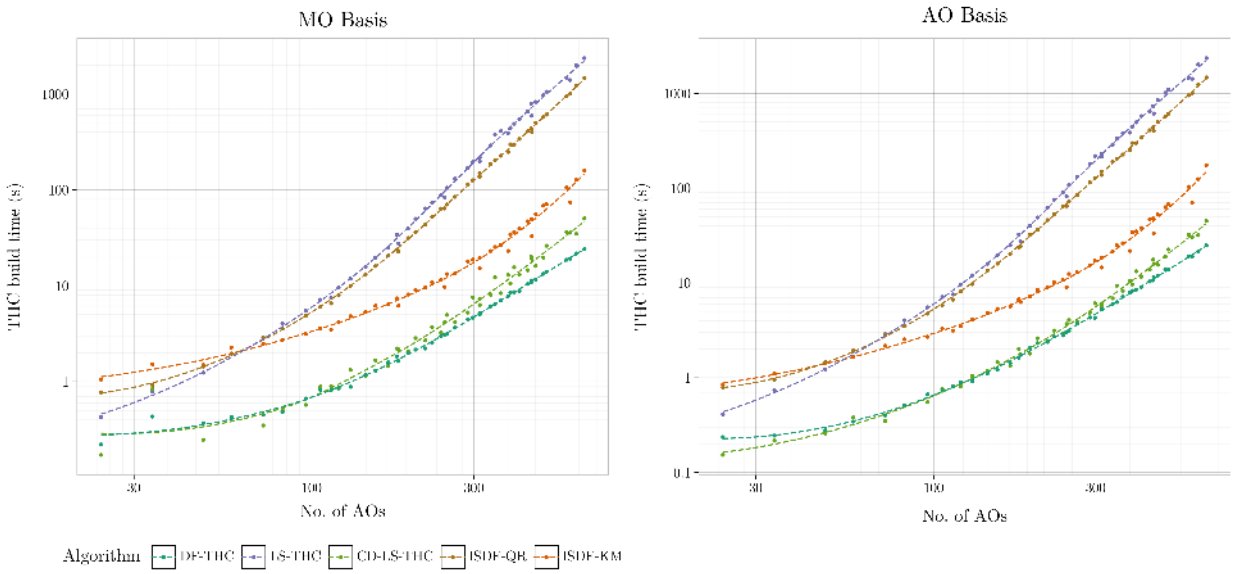


Figure 7: Runtimes of the approximation methods on the test set.

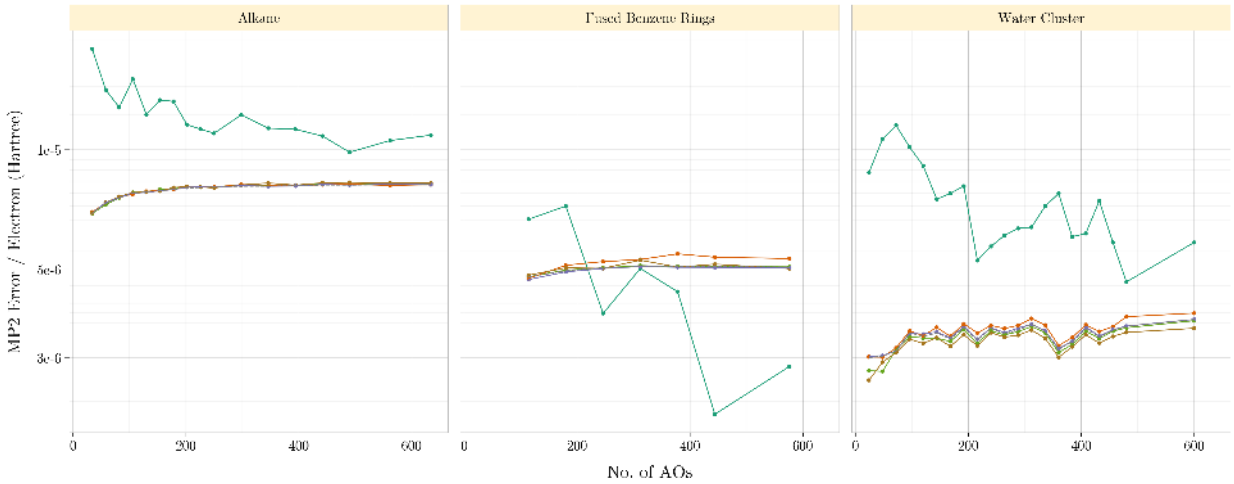


Figure 8: MP2 errors of the approximation methods on the test set.

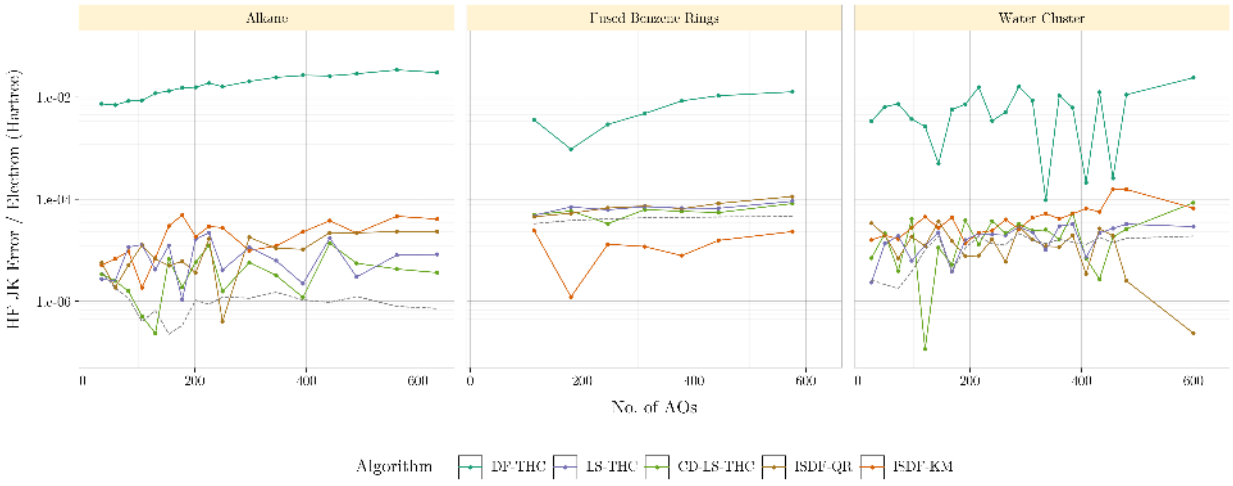


Figure 9: HF-JK errors of the approximation methods on the test set.

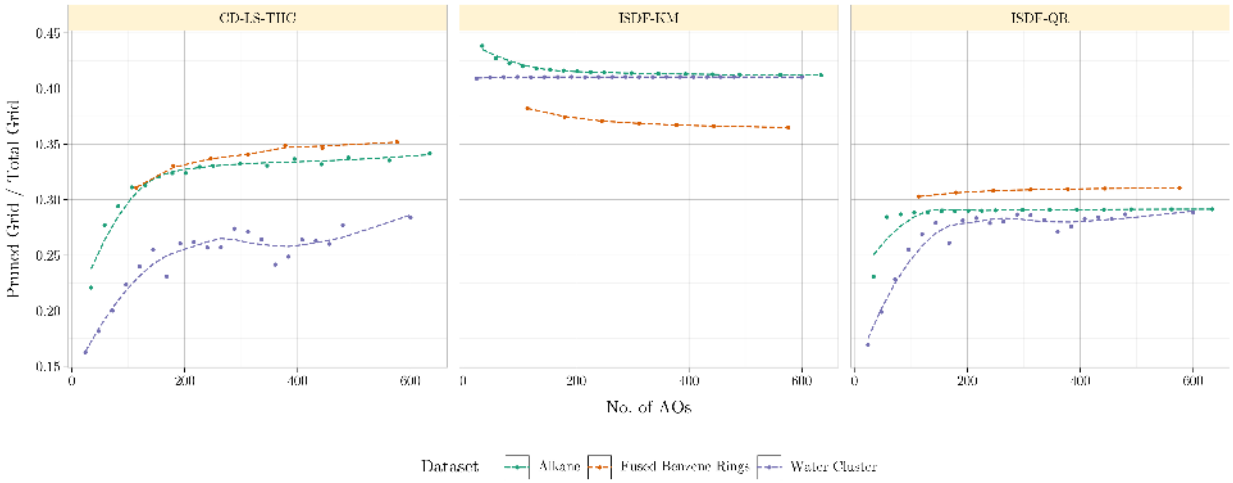


Figure 10: Ratios of pruned to parent grids for the approximation methods that implement grid pruning.

Table 2: Empirical runtime scaling with the input system size of the tested ERI approximation methods as well as average THC build times per atom. The exponents were estimated with a weighted linear regression model.

Method	Avg. Build Time / Atom (AO)	Avg. Build Time / Atom (MO)	Exponent (AO)	Exponent (MO)
LS-THC	6.95s	6.90s	3.33	3.33
CD-LS-THC	0.18s	0.20s	2.37	2.35
ISDF-QR	4.44s	4.42s	3.15	3.16
ISDF-KM	0.56s	0.59s	2.17	2.12
DF-THC	0.13s	0.13s	2.08	2.08

Table 3: Average and maximum relative errors of the approximation methods when compared to canonical MP2 calculations.

Method	Avg. Rel. Error (HF-JK)	Avg. Rel. Error (MP2)	Max. Rel. Error (HF-JK)	Max. Rel. Error (MP2)
DF/RI	$1.412 \cdot 10^{-6}\%$	$2.830 \cdot 10^{-4}\%$	$5.684 \cdot 10^{-6}\%$	$4.475 \cdot 10^{-4}\%$
LS-THC	$2.645 \cdot 10^{-6}\%$	$2.833 \cdot 10^{-4}\%$	$9.034 \cdot 10^{-6}\%$	$4.490 \cdot 10^{-4}\%$
CD-LS-THC	$2.463 \cdot 10^{-6}\%$	$2.822 \cdot 10^{-4}\%$	$8.447 \cdot 10^{-6}\%$	$4.501 \cdot 10^{-4}\%$
ISDF-QR	$2.814 \cdot 10^{-6}\%$	$2.816 \cdot 10^{-4}\%$	$1.172 \cdot 10^{-5}\%$	$4.539 \cdot 10^{-4}\%$
ISDF-KM	$3.871 \cdot 10^{-6}\%$	$2.869 \cdot 10^{-4}\%$	$1.319 \cdot 10^{-5}\%$	$4.507 \cdot 10^{-4}\%$
DF-THC	$1.474 \cdot 10^{-3}\%$	$4.632 \cdot 10^{-4}\%$	$3.826 \cdot 10^{-3}\%$	$1.086 \cdot 10^{-3}\%$

In general, we can see that all methods were able to achieve results with error margins that should be usable in real-world scenarios. While DF-THC offers the fastest results, it also delivers the lowest quality approximations. This is especially true for the HF-JK calculation where DF-THC delivers results almost three orders of magnitude less precise than the other methods. The second-fastest method is CD-LS-THC which offers a lot higher quality results (on par with DF in many cases) although with a higher empirical scaling, already visible on our small systems. The generally better runtime performance in the AO basis through most of the methods is largely due to the required AO to MO transformation. We also observed that relative errors in the HF-JK calculation were about two orders of magnitude lower for almost all methods including DF. The error per electron of the THC approximation seems to be additive to the error made by DF. Across all methods (except DF-THC) we observed very tight fits to the DF errors. This confirms observations made by Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] and Matthews, D. A. [87], who attribute this additive behavior to the localized nature of the Coulomb-operator. We will now collect some observations we made about the behavior of each of the tested approximation methods.

LS-THC: As this method presented the baseline for our test, it was quite encouraging to see it reproduce error rates that were close to the observations made by Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83]. We observed steady error rates per electron, which seem to be tightly fit the DF error. Similar to the original publication, the method performed best in the MP2 calculation on water clusters, which the authors attributed to the sparser nature of the ERI in those systems. Another notable difference lies in the elevated error rates produced when used in the AO ERI calculations of the alkane chains, which the original publication did not compute. Our theory is that this shows the limitations of the grid based approach, and we will elaborate on this further in Section 7. On the test set, we observed the method converging to an empirical scaling upward of N_{basis}^3 (see Table 2), which implies that the systems were not yet big enough to converge the method to its formal $\mathcal{O}\big(N_{\text{basis}}^4\big)$ complexity scaling. Overall, the method took the longest to compute approximations, which can be mainly attributed to the large THC projection space (the unpruned grid), which leads to expensive contractions with the DF ERI as well as a high pre-factor due to the metric matrix inversion. In our tests, we found that the pseudoinversion of the grid metric matrix took considerably more time in the AO basis than in the MO basis, while selecting a higher number of eigenvalues in the MO basis. Since the rest of the algorithm was identical, this likely shows that the metric matrix in the AO basis is numerically more ill-conditioned than in the MO basis. While we were able to mitigate this effect by setting matrix elements smaller than $1 \cdot 10^{-15}$ to zero, the effect can still be seen from the higher average runtime

of LS-THC (see Table 3). The worse runtime of the pseudoinversion even outweighed the AO to MO transformation.

CD-LS-THC: While being similar to LS-THC, the method achieved greater performance. In most of our test cases it delivered approximations that matched LS-THC closely. It even outperformed LS-THC in precision on some HF-JK calculations. Those calculations do not seem to correlate as much with the grid density. The considerable gains in speed of more than an order of magnitude can be explained by it overcoming most limitations inherent in LS-THC: The grid pruning leads to a reduced projection space, and it exchanges the expensive metric matrix inversion for a series of triangular solving steps, which are in turn sped up by the pruned grid. The most expensive steps in the method are the contraction with the DF ERI as well as its AO to MO transformation. The Cholesky Decomposition seems to only add a small contribution to the computational pre-factor. The Accelerated-RPCholesky method was able to calculate decompositions even for vast grids ($> 100,000$ points) in a matter of minutes, while fully utilizing available CPU cores. The amount of grid points chosen seems to be sensitive to the geometry of the system. Figure 10 shows that the CD procedure selected $\approx 25 - 30\%$ of the parent grid for water clusters while using another $\approx 5 - 10\%$ more points for the less-dense systems.

ISDF-QR: As the method is different from the LS-THC approach and does not share the same derivation, we were surprised that it generally achieved comparable error levels. The most time in the procedure was taken for computing the QRCP. This is aligned with the observations collected by Qin, X.; Hu, W.; Yang, J. [56] and Dong, K.; Hu, W.; Lin, L. [102] who describe the QRCP as the computational expensive step in the algorithm. ISDF-QR was also the algorithm that required the most memory. Since for the QRCP, a full $N_{\text{basis}}^2 \times N_{\text{grid}}$ needs to be stored (at least when used with the LAPACK implementation of QRCP). The QRCP selected a similar amount of grid points as the Cholesky Decomposition in CD-LS-THC (see Figure 10). ISDF-QR is the method with its empirical runtime scaling closest to the formal $\mathcal{O}(N_{\text{basis}}^4)$ complexity. For this method we can also see a very slightly decrease in runtime when used to fit the MO ERI. As the grid pruning strategy only uses the AO basis values, we attribute this behavior to the pruning dominating the runtime overshadowing the time taken for the AO to MO transformation by fluctuations in runtime.

ISDF-KM: As expected, eliminating the costly QRCP decomposition from the ISDF procedure enabled it to produce results up to an order of magnitude faster than ISDF-QR. It is clear though, that the method trades computational speed for higher error margins. In all our test sets (except the fused benzene rings in the HF-JK calculation) the error made by ISDF-KM is higher than the one made by ISDF-QR. The higher accuracy of the QRCP pruning might be the explanation: While the K-Means algorithm depends on a set of heuristics (i.e., the cluster multiplier parameter and random initial guess of the centroids), the QRCP processes the full co-density and chooses accurate pivots by their columns Euclidean norm. It is therefore not surprising that the pivots of ISDF-QR will generally yield a "better" sub-grid. The K-Means clustering process was slower in our tests than the pivoted CD in CD-LS-THC. The small overhead in the fitting of the MO basis ERI was similar to CD-LS-THC and can be attributed to the AO to MO transformation.

DF-THC: This method was able to deliver results faster than all other methods we tested and shows quadratic empirical scaling with N_{basis}. We already expected that it would yield the lowest fidelity approximations as it was described as less precise by Hillers-Bendtsen, A. E.; Martínez, T. J. [95]. Surprisingly enough, in absolute terms, the method was able to achieve reasonable errors (in the milli-Hartree regime) in the MO basis MP2 calculations. Its accuracy was more aligned to the other methods, especially on the fused benzene systems where it came closer to the canonical value than the other methods. In the AO basis it produced relative and absolute errors that were

at least two orders of magnitude higher than the ones produced by the other methods as well as DF. We should also note that the output of the method in our current implementation is still suboptimal for subsequent calculations since the result $\boldsymbol{X}$ matrix spans a full, unpruned version of the grid. We were hesitant to lower the grid density though as the error margins produced by the algorithm were still high. As we expected, calculations in the AO basis were just as fast as in the MO basis, since the algorithm fully shifts the problem into the auxiliary basis.

Overall, it should be noted that our implementations of the approximation methods do not yet make optimal use of CPU and RAM. While we tried to efficiently use the system's resources, we heavily relied on the automatic contraction path determination in TBLIS. We found in our testing that while the program generally saturated the available CPU cores, some contractions like the one used for building the $\boldsymbol{H}$ tensor in Algorithm 3 would present computational bottlenecks even with TBLIS determining an "optimal" contraction path. Due to time constraints we also could not optimize all methods for proper RAM usage, by fully exploiting the symmetry properties inherent in the co-density and ERI objects. While the 384GB of RAM was more than enough to process systems with $N_{\text{basis}} < 700$, our implementations regularly used more than this when used on bigger systems.

6.3 Calculating Bigger Systems

In order to properly evaluate the achievable performance and therefore the practical feasibility of THC based ERI approximations, we conducted a second round of benchmarking, only using the CD-LS-THC algorithm. We chose it because it was one of the fastest algorithms in our previous test-set and produced reasonably low error rates as well as a much reduced projection space. We implemented an improved version of Algorithm 6 which computes the expensive DF ERI contraction with the co-density in blocks. The new implementation also takes advantage of the 4-fold symmetry of the co-density and ERI matrices (see Section 4.1). This allows for proper RAM utilization while not over-allocating memory. With this implementation, we were able to calculate bigger systems of water clusters up to $N = 65$ ($N_{\text{basis}} \approx 1500$). As a canonical calculation of MP2 would be very expensive for systems of this size (at least when done with PySCF), we only calculated the DF-MP2 results to provide a comparison. For this benchmark, we used an implementation of the Laplace-quadrature MP2 procedure shown in Algorithm 3 (THC-LT-MP2), which also employs block-based computations to conserve RAM usage. We used 10 quadrature points, which provided calculated values that matched the DF-MP2 energies to milli-Hartree margins. We also measured the time taken to compute the J term since we could stop the algorithm early and return the SOS approximation to the MP2 energy (THC-LT-SOS-MP2). The results are presented in Figure 11 and Figure 12 together with their estimated runtime scaling exponents in Table 4 and Table 5.

By analyzing Figure 11, we can clearly see a crossover point at which our implementation of THC-LT-DF-MP2 outperforms PySCF's DF-MP2. The crossover already happens for systems with $N_{\text{basis}} < 300$. For the biggest systems we tested (a water cluster with 65 molecules amounting to $N_{\text{basis}} = 1560$) the full calculation of DF-MP2 took 130 minutes, while THC-LT-MP2 provided a result in 40 minutes and the THC-LT-SOS-MP2 result was obtained in just under 12 minutes. The empirical scaling exponents displayed in Table 4 also broadly confirm our expectations. While we would expect DF-MP2 to exhibit $\mathcal{O}\big(N_{\text{basis}}^5\big)$ complexity scaling, we did not converge into its asymptotic runtime behavior. For the other methods we can also observe exponents that broadly conform with theory: $\mathcal{O}\big(N_{\text{basis}}^4\big)$ and $\mathcal{O}\big(N_{\text{basis}}^3\big)$ for THC-LT-MP2 and THC-LT-SOS-MP2 respectively.

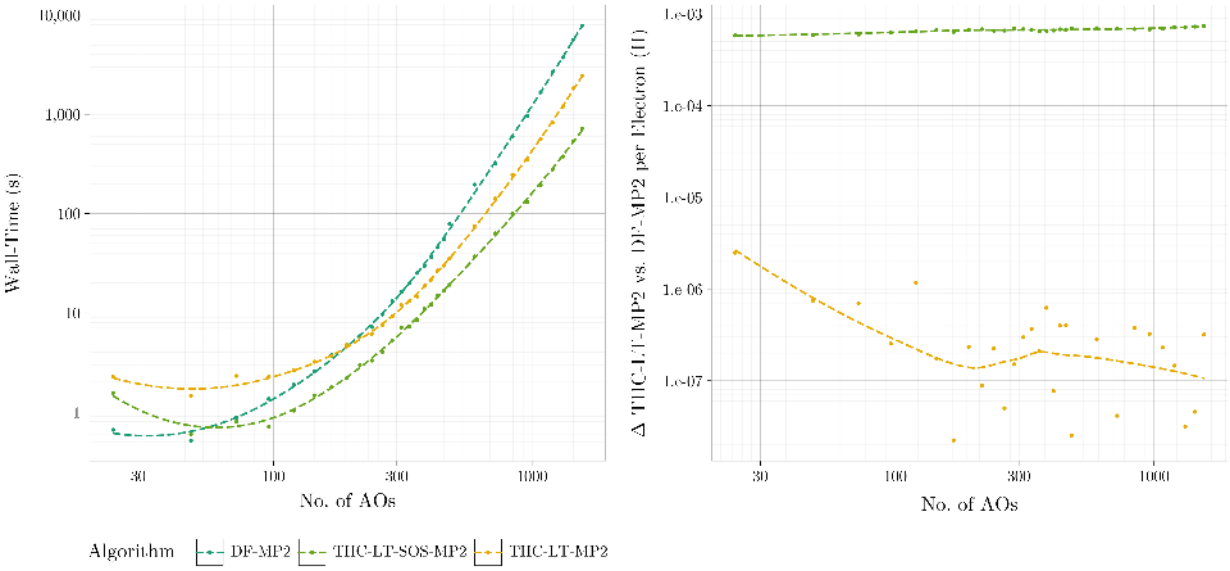


Figure 11: Runtime of the MP2 algorithms measures versus the input system size.

Table 4: Empirical runtime scaling with the input system size of the MP2 algorithms. Measured with a weighted linear regression model.

Method	Scaling Exponent
DF-MP2	3.79
THC-LT-MP2	3.27
THC-LT-SOS-MP2	2.84

We expected THC-LT-SOS-MP2 to produce a high error margin when compared to DF-MP2. Since SOS MP2 tries to better fit the actual energy of the system with empirical scaling parameters, the DF-MP2 energy is not a fair comparison. THC-LT-MP2 reproduces the DF-MP2 values accurately, often up to levels close to machine precision ($\approx 1 \cdot 10^{-8}$). The error also does not seem to increase with system size. On the contrary, the error per electron slightly decreased for bigger systems, while staying underneath $1 \cdot 10^{-6}$ Hartree per electron.

To provide a more detailed analysis, we measured the runtime of various steps in the THC-LT-MP2 procedure. The timings and the estimated asymptotic scaling exponents are displayed in Figure 12 and Table 5. The main runtime determining component is the K-build step. Interestingly enough, this is only one of the formally $\mathcal{O}(N_{\text{basis}}^4)$ scaling steps in the algorithm, the second being the THC-build. It seems as if the size of our test systems did not suffice to even converge the CD-LS-THC algorithm to above cubic empirical scaling, while the K-build step almost exhibits quartic scaling for the biggest water clusters. The displayed timings of the Cholesky grid pruning also show the efficiency of the Accelerated-RPCholesky algorithm. We already pointed out that the parent grid was a lot denser than the basis (often up to $N_{\text{grid}} \approx 1000 \cdot N_{\text{basis}}$), but the algorithm achieved its pruning with such a low computational pre-factor that it did not even exceed cubic empirical scaling with N_{basis}. This is likely due to the low-rank nature of the provided metric matrices. We also observed the memory footprint of the CD procedure to be negligible for all systems.

We decided to calculate even bigger systems. We therefore generated a DNA fragment involving 5 base pairs, converted it into coordinate files using the Open Babel program suite[123] and ran the THC-LT-SOS-MP2 algorithm. We did not calculate a DF-MP2 energy as a smaller system with only 3 base pairs in the cc-pVDZ basis ($N_{\text{basis}} \approx 2100$) took PySCF's DF MP2 implementation

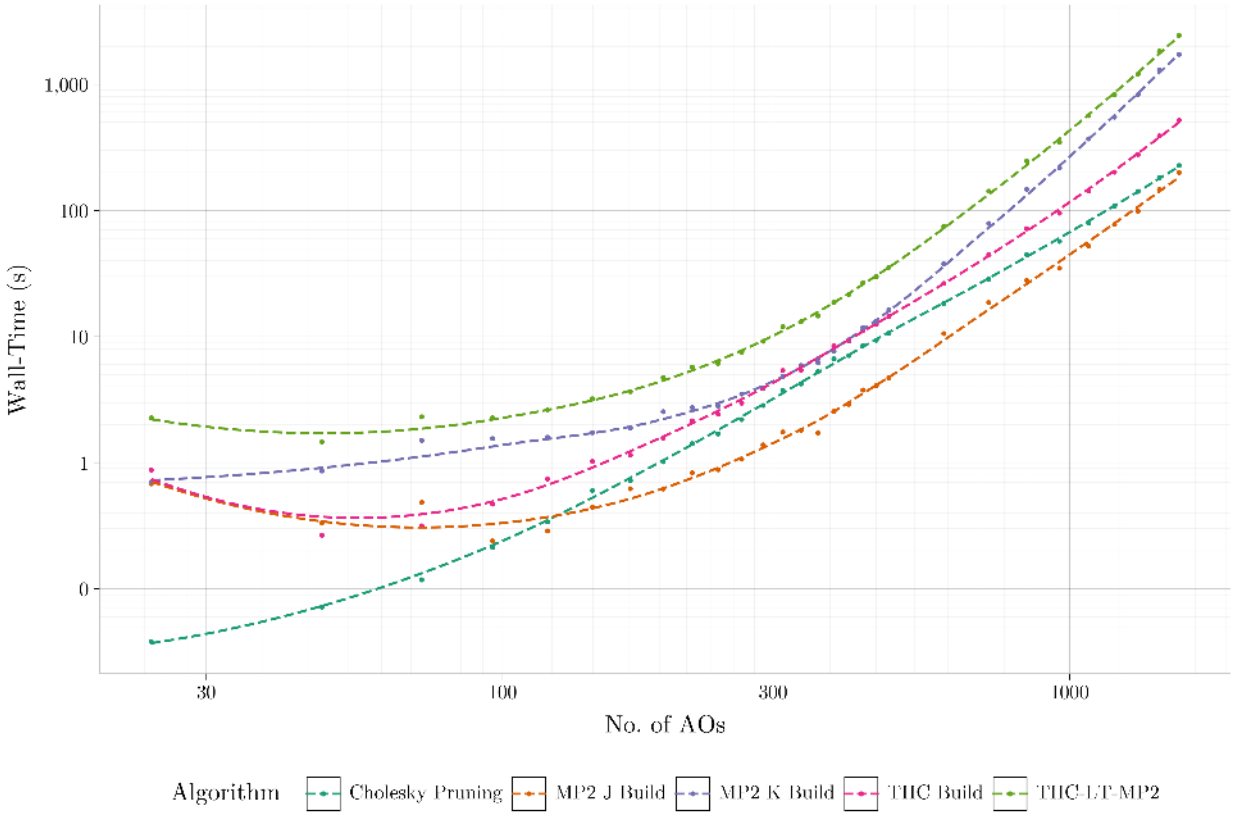
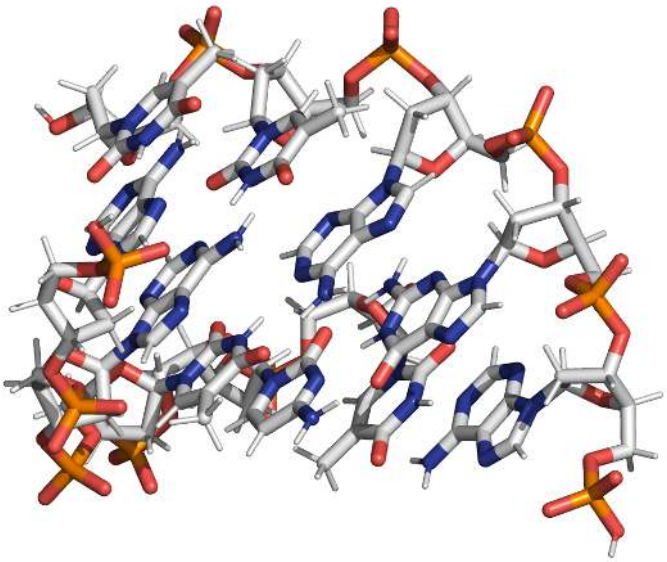


Figure 12: Detailed runtimes of steps in THC-Laplace MP2.

more than 11 hours to compute. Due to these limitations, we could not evaluate the accuracy of our SOS MP2 results on the DNA fragments. The 5-base pair fragment as well as the resulting metrics are displayed in Figure 13. For even bigger systems we eventually faced limitations from the PySCF implementation of DF, which writes the full $\boldsymbol{L}^K_{\mu\nu}$ tensor from Eq. (4.13) to disk. This amounts to more than 1TB for systems bigger than the 5-base pair fragment.

Table 5: Empirical runtime scaling of build steps in the THC-Laplace MP2 algorithm. Measured with a weighted linear regression model.

Method	**Scaling Exponent**
Cholesky Pruning	2.55
THC Build	2.82
MP2 J Build	2.89
MP2 K Build	3.63
$\sum$ **THC-Laplace MP2**	3.27



Basis	cc-pVDZ
No. of AOs	3,528
Parent grid size	229,128
Pruned grid size	55,670
Cholesky Pruning	00:37:00h
Full THC build time	03:22:00h
SOS MP2 build time	00:48:00h
THC-LT-SOS-MP2 time	04:10:00h

Figure 13: DNA fragment with 5 base pairs that was used for testing the SOS MP2 implementation.

7 Discussion

In the course of our benchmarks, most of our expectations were confirmed, but there were some points that we found worth exploring to a deeper level. We will now discuss possible limitations inherent in our chosen research method.

7.1 Limitations by the Test-Data

Our tests only measured errors in single-point energies. While this technique is used in many publications that also conduct tests of different ERI approximations[6,7,83,99] other publications also measure energy differences between reactants and products of chemical reactions[100,103]. The approach of measuring single-point energies is useful to determine the runtime behavior of a method, since we can use systematically extendable systems. The reaction-based measurement, on the other hand, might give a more nuanced picture of the error that would occur when used in real-world chemical scenarios. This is mainly due to the error cancellation that might be observed in the relative energy calculation.

7.2 Limitations by Density Fitting

All our tests so far used the DF ERI to fit against. We decided not to use the full ERI for fitting, since the DF approximation is widely considered to deliver reasonable results and for bigger systems (which are the main target for THC based calculations) the instantiation of the full ERI is the exact problem we would like to avoid. Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83] already showed that using LS-THC with the full ERI outperforms DF in terms of approximation quality by multiple orders of magnitude. We already observed during Section 6.1 that all our methods would converge on an error margin extremely close to the one of the DF approximation. Our data also shows this pattern, with the notable exception of some of the randomized method producing lower errors than DF in AO basis HF-JK calculations. Since we know all methods to be mathematically limited to the error level made by DF, we attribute those cases to error cancellation or noise from the randomized steps involved. Overall, this means that the error margins we measured should ideally be viewed not in direct comparison to the full ERI method, but rather in comparison to the DF error.

We wanted to quantify how much the DF fitting limited our calculations and whether the same trends we saw between the different systems would reappear if we used the full ERI for fitting. We therefore calculated all systems with less than 300 basis functions using LS-THC and CD-LS-THC. The results are displayed in Figure 14. The biggest difference to the systems we calculated in Section 6.2 lies in the results of the MP2 calculation, which now outperforms DF by multiple orders of magnitude. This matches the observations made by the original LS-THC publication. In the HF-JK calculation both approximations now converge to a similar error margin. We conclude that the fitting against the DF ERI is the accuracy bottleneck of our THC based methods in the MP2 calculations. In the HF-JK calculations, the error did not substantially decrease when using the full ERI, which means that our method might be limited by other factors like the integration grid itself. We will elaborate on those shortcomings in the next section.

We also explored how much the DF limits our MP2 calculation and how much the THC can be improved by improving the DF fit. For this we calculated the MP2 error of benzene using different auxiliary basis sets with increasing size. We displayed the results in Figure 15. While the quality of DF approximately doubled, the THC based approximations tracked this change closely. This

strengthens our verdict that the THC approximation is of very high fidelity, but is only as good as the input used for fitting.

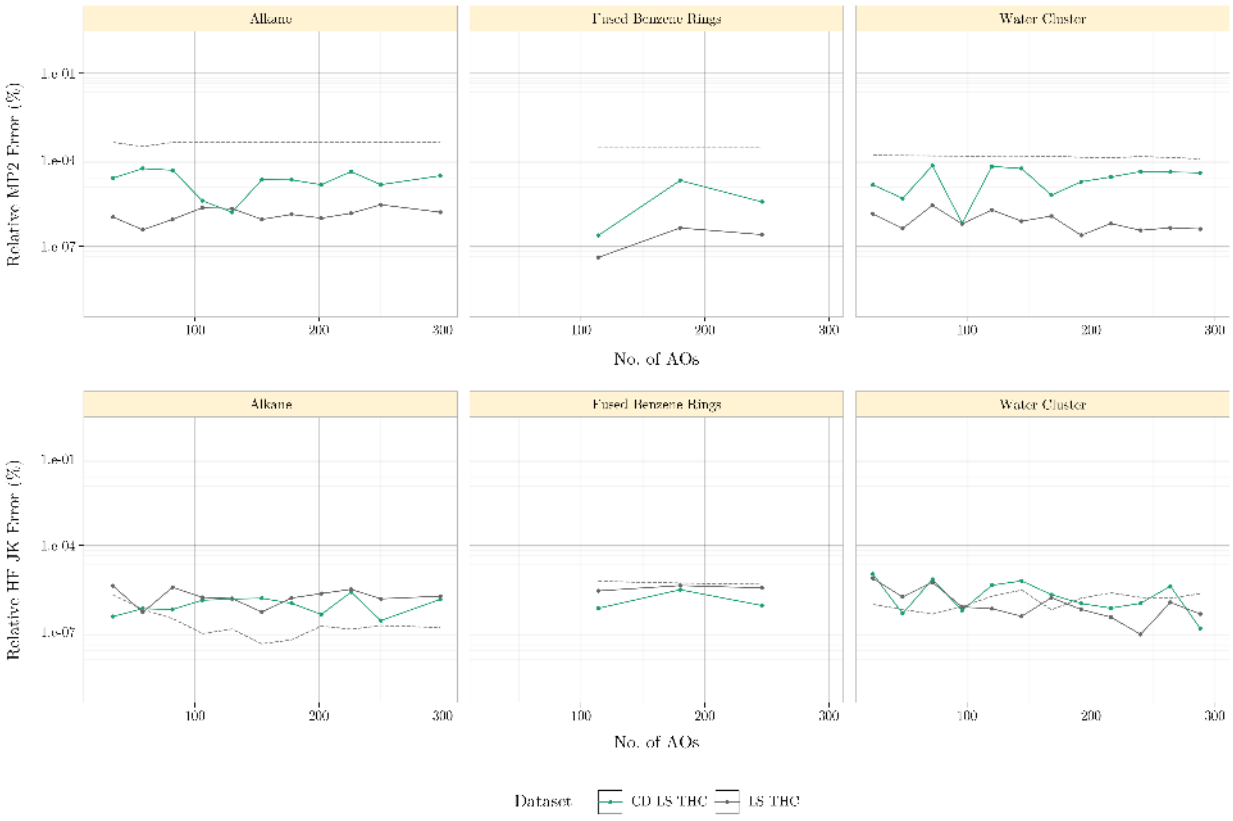


Figure 14: Relative errors of LS-THC and CD-LS-THC compared against DF (dashed line) on systems with $N_{\text{basis}} \leq 300$ using full ERI fitting.

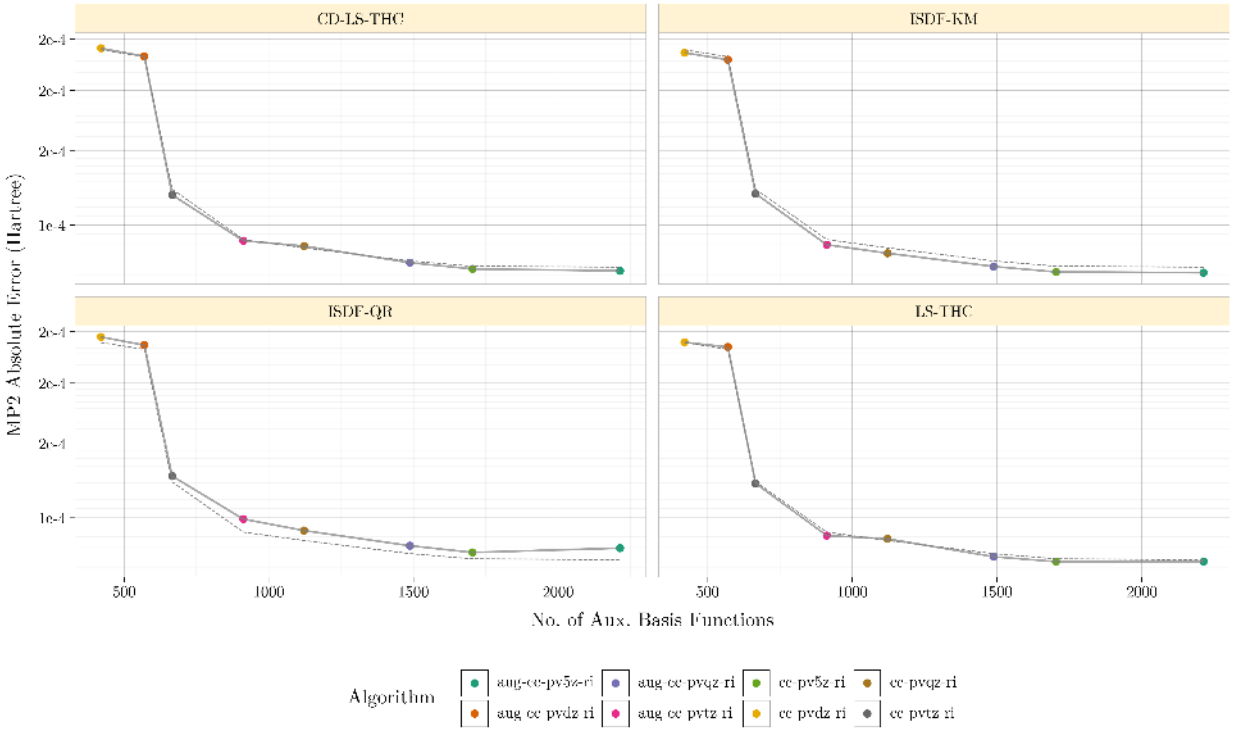


Figure 15: Effect of different auxiliary basis sets on MP2 approximation errors measured on Benzene (cc-pVDZ).

7.3 Limitations of the Grid

All our tested approximation techniques rely on real-space grids to discretize the information of the co-density interactions. A clear trend that we saw in our comparative benchmark was that the choice of this grid is extremely important to the quality of the approximation as well as the time it can be computed in. As described by Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [83], the THC approximation should become exact when the grid size approaches N^2_{basis}. Unfortunately, this size would make methods like LS-THC infeasible for all but the smallest systems. Our tests clearly show that approximations of similar fidelity can be achieved orders of magnitude faster by employing clever ways to select a sub-grid.

We wanted to explore how much the chosen parent grid was limiting our calculations. We therefore plotted the AO and MO wavefunctions as well as their pairwise products for ethane projected along the bond-axis (Figure 16). We chose this molecule because the geometry was easy enough to visualize in a two-dimensional plot. We also added the density of the grid projected onto the bond-axis to the visualization. From Figure 16, it becomes quite clear that the product space of the molecule in the AO basis is more localized than the occupied-virtual orbital products. The virtual orbitals are less local and show more oscillatory behavior (they possess more nodes). Simultaneously we can see that the Becke grid is sampling the regions around the CH_3 center of mass the most, which likely leads to undersampling of the inter atomic and outer regions of the molecule. This could mean that the higher error margins observed in the MP2 calculations were also due to the initial Becke grid limiting the overall fidelity. A publication that acknowledges this problem and proposes potential solutions can be found in Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. [99], who proposed a discrete variable scheme to achieve better sampling, while lowering the grid density needed.

To test this hypothesis, we added an additional hydrogen atom at the origin of the ethane system using PySCF's ghost-atom feature. This added another set of grid points in the bond-region of the molecule. For the ghost atom we used a level 1 grid, which is one density level higher than the parent grid. We removed the additionally added basis functions of the ghost atom as to not skew our calculation by altering the co-density landscape. The resulting system is displayed in Figure 17. We then calculated the MP2 and HF-JK energies using LS-THC and the full-denominator MP2 implementation and compared them to the regular ethane system. We used the full ERI for fitting, since using the DF ERI would add a lower bound to the quality of our approximation. From Table 6 we can see that the additional atom forces denser grid coverage in the bonding region, which decreased the error in both calculations by about one order of magnitude. This confirms our hypothesis that the Becke grid used in our computations might be too sparse in the relevant inter-atomic regions.

We also observed higher variances in the HF-JK calculations, especially when compared to DF, which provided very steady absolute and relative error rates. A possible explanation could also be provided by the more localized nature of the AO co-densities. It is easy to imagine that sampling with a grid that is too sparse could easily skip high-amplitude regions if they are strongly localized. Therefore, the initial setup of the grid in conjunction with the geometry of the systems determines if the grid is dense enough to capture all relevant regions. Since our grid building was only considering single atoms and not the whole system, we might encounter proper sampling in some cases and be limited by the parent grid in others yielding a higher overall variance. This effect would be less pronounced in densely packed systems, since the atom centered grids would overlap more and therefore provide a higher sampling rate per volume of space.

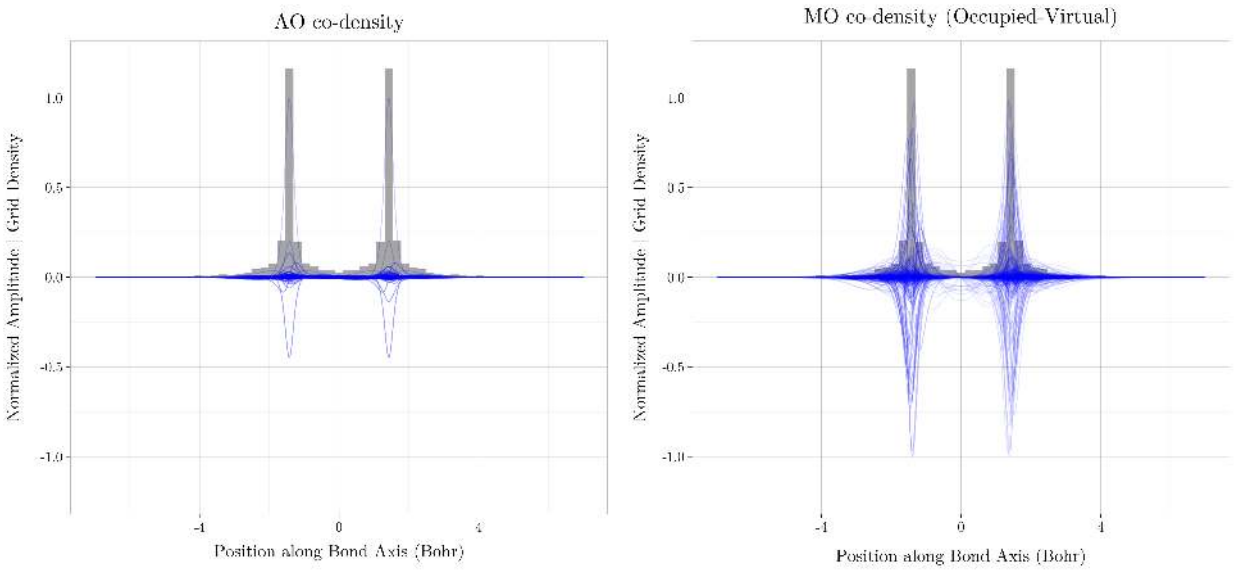


Figure 16: AO and MO wavefunctions, AO products and MO occupied-virtual orbital products for ethane for one spatial dimension on the bond-axis combined with a histogram of the integration grid density.

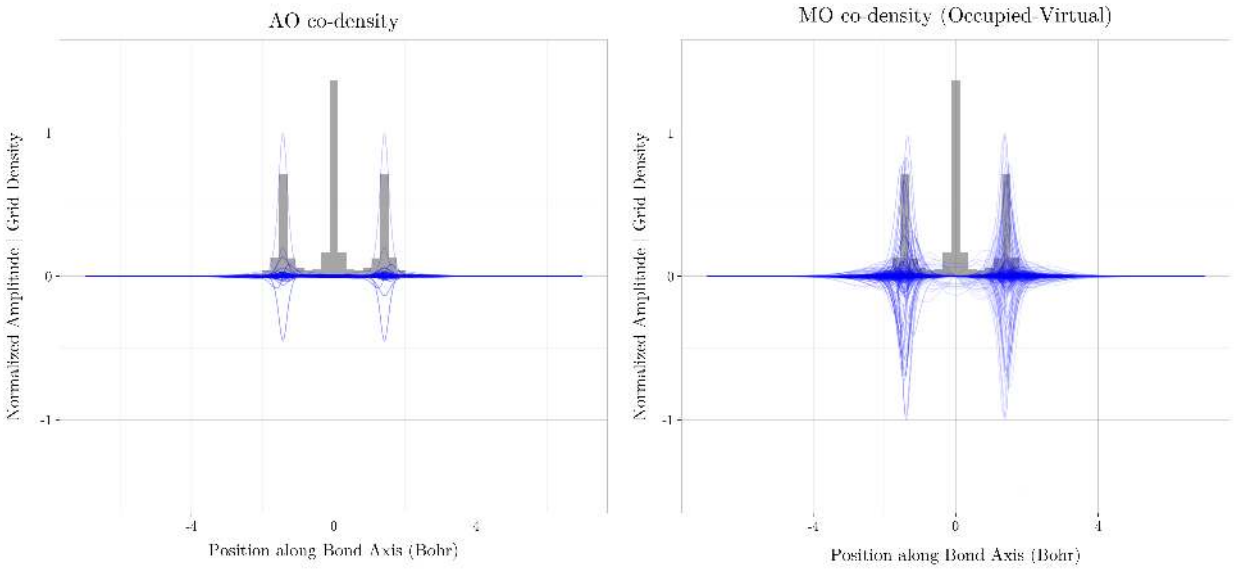


Figure 17: AO and MO wavefunctions, AO products and MO occupied-virtual orbital products for Ethane with an added ghost atom for one spatial dimension on the bond-axis combined with a histogram of the integration grid density.

Table 6: Relative errors for ethane and ethane with an added ghost hydrogen to force higher grid sampling.

System	**Rel. HF-JK Error**	**Rel. HF-JK Error DF**	**Rel. MP2 Error**	**Rel. MP2 Error DF**
Ethane	$3.815 \cdot 10^{-6}\%$	$5.968 \cdot 10^{-7}\%$	$1.179 \cdot 10^{-6}\%$	$4.027 \cdot 10^{-4}\%$
Ethane with ghost atom	$2.463 \cdot 10^{-7}\%$	$5.968 \cdot 10^{-7}\%$	$1.586 \cdot 10^{-7}\%$	$4.027 \cdot 10^{-4}\%$

We conclude that the choice of the parent grid is just as important as the employed pruning strategy. Grid pruning is not a universal solution but is always limited by the informational content captured by the parent grid. We saw that for calculations that include virtual-orbitals, the atom centered integration grids we used were not optimal, as they failed to account for high-amplitude

components in inter-atomic regions. Parent grids that apply a higher sampling rate to regions of interest could yield results of even higher fidelity.

7.4 Limitations from Underlying Assumptions

The underlying assumption of THC as well as DF is that the co-density of the wavefunctions contains enough redundant information to be spanned by an auxiliary basis set of size $\approx c \cdot N_{\text{basis}}$ where c is a small constant. This means, that the information contained in the co-density only scales linearly with system size and that the interaction of the approximated co-densities are a good stand-in for the actual 4-center interaction. The low error rates attained by the DF approximation in our tests validate this assumption, as do several experiments described in the literature, which we discussed in Section 4.2.

Some of our methods diverged from this assumption. The non-deterministic clustering step in ISDF-KM for example, gives higher significance to grid points with an overall high density. We already explained in Section 7.3 how density is not in all cases a suitable proxy for co-density since it is strictly positive and carries none of the phase information (i.e., sign changes or oscillatory behavior). To further investigate the exact reason for the lower fidelity of the K-Means clustering, we plotted the relative deviation from the canonical value that all our approximations made (Figure 18). Ideally we would expect most of our methods to exhibit the same systematic biases as DF due to the fitting process. We can see this expected behavior for all methods except ISDF-KM. It should be noted that we did not include DF-THC in Figure 18. Since it deviates from both the DF results and the canonical value by a much higher margin, it would have been challenging to visually differentiate the other methods. For ISDF-KM we can see it overestimating energies in the HF-JK calculation on the alkanes as well as on the fused benzene rings. This overestimation leads to the lower distance from the canonical value in the fused benzene systems compared to DF. DF seemingly underestimated the energy while ISDF-KM overshoots when approximating DF, leading to cancellation effects.

In order to quantify this error, we calculated the J and K contributions of ISDF-KM and DF and compared them to the canonical value (Figure 19). The method steadily overestimates the J (repulsion) contribution more than the K (exchange) contribution. Since the J term is much higher than K in absolute numbers (on average by factor of 6 on our test systems), we can attribute most of the error to the J term. The overestimation seems to only occur on the alkane chains and fused benzene rings. This overestimation is clearly related to the density based clustering, as we did not observe this behavior in ISDF-QR. A possible explanation could lie in the clustering oversampling regions near the atomic centers and in the covalent bonding regions, due to the higher electron density. This could lead to an over representation of these regions in the subsequent least-squares problem, which could bias the solution towards higher electron repulsion contributions, as they are more pronounced in regions with higher density. This claim might be supported by visual inspection comparing the pruned grids displayed in Appendix B. Overall we note that while ISDF-KM provided the lowest fidelity results of our tested methods (except DF-THC), the absolute errors are still in a reasonable range and the substantially lower runtime might justify its use over ISDF-QR in many cases.

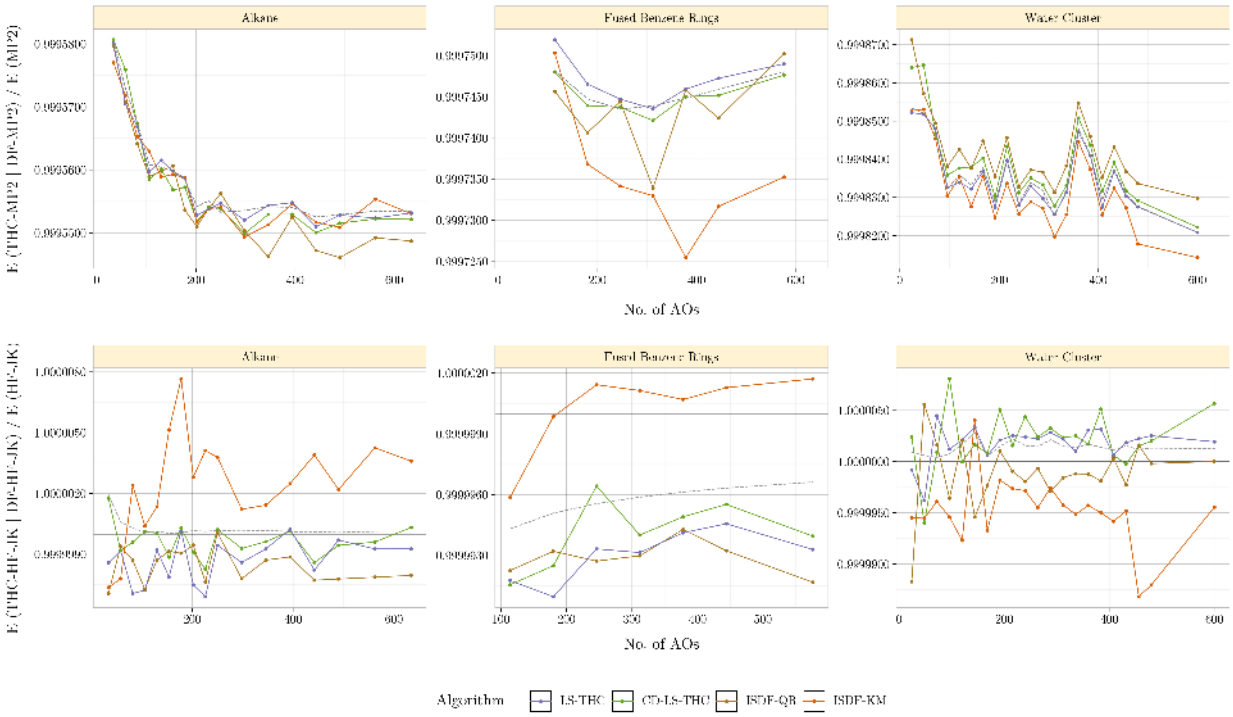


Figure 18: Energy ratios of THC approximations and DF (dashed line) to the canonical value.

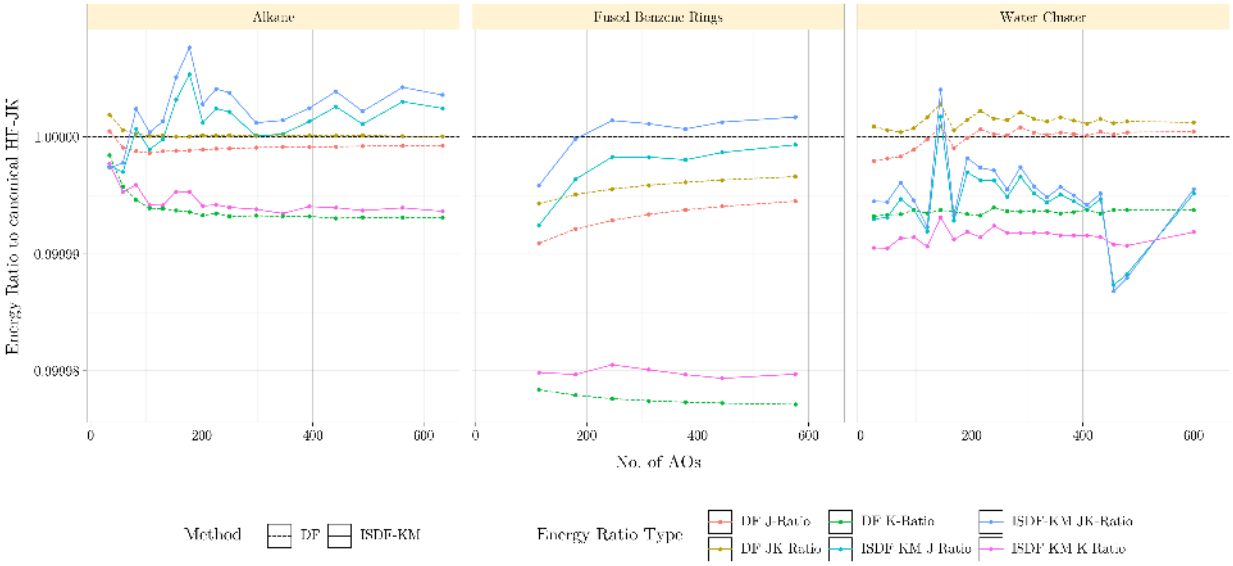


Figure 19: Energy ratios of the J and K contributions of ISDF-KM HF-JK compared to DF together with the $\frac{J-K}{J_{\text{ref}}-K_{\text{ref}}}$ values.

More problematic than the error margins in ISDF-KM was the fidelity delivered by DF-THC. We clearly saw all other methods outperforming it in terms of quality by usually more than one order of magnitude. This confirms that the auxiliary basis functions is likely not flexible enough to capture the information contained in the THC center matrix. We expected this result, as it was already suggested in the original publication[96]. It should be noted that the error made by the approximation could still be small enough to be useful. As our test set only compared absolute energies, we cannot estimate the method's fidelity when used to calculate reaction energies, where some error cancellation is to be expected. Therefore, DF-THC might be useful in a different context

than was used in our test-setup or with system sizes that are so big that the formal $\mathcal{O}(N^3)$ scaling of the method outweighs the lower fidelity compared to methods like CD-LS-THC.

7.5 Practical Considerations

In the past chapter, we mainly discussed the overall empirical runtime scaling of our methods. This is a very relevant metric in method development, as the goal is to develop algorithms that ideally can be scaled to systems exceeding 10,000 basis functions. Comparing algorithms on this basis, might not be ideal for most practical use cases. For all the methods presented in this thesis this pre-factor heavily depends on the properties of the system in question as well as on the concrete implementation of the algorithm. Algorithms that are highly optimized for the hardware in use might yield an approximation so fast that this outweighs lower fidelity. One such example is DF-THC as discussed by Hillers-Bendtsen, A. E.; Martínez, T. J. [96]. In the publication the algorithm was implemented on GPUs. This THC approximation is uniquely useful in this case, since it involves no memory expensive order-3 tensors and host-device memory transfer is often a limiting factor for GPU based computations. Paired with a technique to mitigate the higher error margins the authors could still achieve chemically accurate results for systems with many thousands of basis functions in matter of seconds.

This makes the quantitative comparison together with the reference implementation achieved by this thesis all the more relevant. It provides a starting point for future method development: A novel idea to compute the THC (or any other ERI approximation) can be seamlessly integrated and benchmarked in the existing library, reusing the fundamental testing infrastructure. This addresses the problem described in Section 4.9.

8 Conclusion and Future Work

In this thesis, we described the complexity bottleneck imposed in most quantum chemistry calculations by the handling of Electron Repulsion Integrals. We explored the diverse landscape of approaches to lower this complexity by introducing approximations. We derived the need for a differential and fair comparison, which we implemented by developing reference implementations of LS-THC, CD-LS-THC, ISDF-QR, ISDF-KM and DF-THC and testing them on a unified set of molecular systems. This work delivers the most detailed comparison that we know of to-date. We observed that all methods, with the notable exception of DF-THC, could be tuned to reproduce Hartree-Fock electron repulsion energies as well as MP2 energy corrections with only a marginal error added to the one introduced by the widely-used DF approximation. While testing, we observed stark differences in the time needed to compute the THC matrices between the methods. While all methods fitting the DF ERI retain a formal $\mathcal{O}\big(N_{\text{basis}}^4\big)$ scaling, there is a lot of potential for improvement in real-world calculations by lowering the computational pre-factor. This is achievable by employing sophisticated pruning strategies on the molecular integration grid in use.

In our tests, the Cholesky Decomposition based CD-LS-THC method offered an especially low computational pre-factor in conjunction with low error rates. When tested against PySCF's implementation of DF-MP2, it achieved a runtime crossover on small systems with less than 300 basis functions which made us able to compute the MP2 energy correction for systems with over 3000 basis functions on a 96-core CPU. This makes us hopeful that with a more optimized implementation, THC based approximations will be able to compete with highly optimized DF-RI codes in the future. Although we might add that while all operations involved in the algorithms used in this work are quite simple, the sheer size of the input can make hardware usage optimization a difficult task. Therefore, future works could explore other paths of lowering the computational complexity further. As we did not use any sparsity considerations in our implementations, this would be one direction to explore. The time taken to compute the DF approximation is the lower bound for most of our algorithms, using a faster implementation of DF than the one provided by PySCF could also lower runtimes as well as using integration grids that are less dense and more suited for the co-density approximation.

The scope of this thesis was limited to molecular systems and the HF-JK and MP2 calculations. Therefore, exploring the performance of our THC implementations on periodic systems as well as calculations like GW and RPA[124] could also be considered. Future studies should also explore energies along reaction pathways, to allow better judgment of the error introduced by the THC. We could also imagine adding more and novel approaches to compute the THC like the Block Tensor Decomposition (see Section 4.7). The software package presented in this work already contains all the basic infrastructure to compute the THC together with software modules for automated benchmarking. This should considerably aid the integration of either new approximation approaches or new quantum chemical calculations, making method development easier and laying the foundation for future research projects.

References

(1) Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, 1. publ., unabr., unaltered republ. of the 1. ed., New York 1989.; Dover Publications: Mineola, New York, 1996.

(2) Jensen, F. *Introduction to Computational Chemistry*, Third edition.; John Wiley & Sons: Chichester, UK Hoboken, NJ, 2017.

(3) Blum, V.; Asahi, R.; Autschbach, J.; Bannwarth, C.; Bihlmayer, G.; Blügel, S.; Burns, L. A.; Crawford, T. D.; Dawson, W.; Jong, W. A. de; Draxl, C.; Filippi, C.; Genovese, L.; Giannozzi, P.; Govind, N.; Hammes-Schiffer, S.; Hammond, J. R.; Hourahine, B.; Jain, A.; Kanai, Y.; Kent, P. R. C.; Larsen, A. H.; Lehtola, S.; Li, X.; Lindh, R.; Maeda, S.; Makri, N.; Moussa, J.; Nakajima, T.; Nash, J. A.; Oliveira, M. J. T.; Patel, P. D.; Pizzi, G.; Pourtois, G.; Pritchard, B. P.; Rabani, E.; Reiher, M.; Reining, L.; Ren, X.; Rossi, M.; Schlegel, H. B.; Seriani, N.; Slipchenko, L. V.; Thom, A.; Valeev, E. F.; Van Troeye, B.; Visscher, L.; Vlček, V.; Werner, H.-J.; Williams-Young, D. B.; Windus, T. L. Roadmap on Methods and Software for Electronic Structure Based Simulations in Chemistry and Materials. *Electronic Structure* **2024**, *6* (4), 42501. https://doi.org/10.1088/2516-1075/ad48ec.

(4) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. On the Applicability of LCAO-Xα Methods to Molecules Containing Transition Metal Atoms: The Nickel Atom and Nickel Hydride. *International Journal of Quantum Chemistry* **1977**, *12* (S11), 81–87. https://doi.org/10.1002/qua.560120813.

(5) Beebe, N. H. F.; Linderberg, J. Simplifications in the generation and transformation of two-electron integrals in molecular calculations. *International Journal of Quantum Chemistry* **1977**, *12* (4), 683–705. https://doi.org/10.1002/qua.560120408.

(6) Hohenstein, E. G.; Parrish, R. M.; Martínez, T. J. Tensor Hypercontraction Density Fitting. I. Quartic Scaling Second- and Third-Order Møller-Plesset Perturbation Theory. *The Journal of Chemical Physics* **2012**, *137* (4), 44103. https://doi.org/10.1063/1.4732310.

(7) Lu, J.; Ying, L. Compression of the Electron Repulsion Integral Tensor in Tensor Hypercontraction Format with Cubic Scaling Cost. *Journal of Computational Physics* **2015**, *302*, 329–335. https://doi.org/10.1016/j.jcp.2015.09.014.

(8) Pierce, K.; Rishi, V.; Valeev, E. F. Robust Approximation of Tensor Networks: Application to Grid-Free Tensor Factorization of the Coulomb Interaction. *Journal of Chemical Theory and Computation* **2021**, *17* (4), 2217–2230. https://doi.org/10.1021/acs.jctc.0c01310.

(9) Kolda, T. G.; Bader, B. W. Tensor Decompositions and Applications. *SIAM Review* **2009**, *51* (3), 455–500. https://doi.org/10.1137/07070111X.

(10) Pierce, K. Toward Using Matrix-free Tensor Decompositions to Systematically Improve Approximate Tensor-Networks. *Journal of Chemical Theory and Computation* **2025**. https://doi.org/10.1021/acs.jctc.5c00413.

(11) Kwon, T.; Ko, J.; Jung, J.; Shin, K. *TensorCodec: Compact Lossy Compression of Tensors without Strong Data Assumptions*. https://doi.org/10.48550/arXiv.2309.10310.

(12) Schutski, R.; Zhao, J.; Henderson, T. M.; Scuseria, G. E. Tensor-Structured Coupled Cluster Theory. *The Journal of Chemical Physics* **2017**, *147* (18), 184113. https://doi.org/10.1063/1.4996988.

(13) Axler, S. *Linear Algebra Done Right*; Undergraduate Texts in Mathematics; Springer International Publishing: Cham, 2024. https://doi.org/10.1007/978-3-031-41026-0.

(14) Strang, G. Linear Algebra and Its Applications (Fourth Edition). **2006**, *1* (4).

(15) Tucker, A. The Growing Importance of Linear Algebra in Undergraduate Mathematics. *THE COLLEGE MATHEMATICS JOURNAL* **1993**, *24* (1).

(16) Hillar, C. J.; Lim, L.-H. Most Tensor Problems Are NP-Hard. *J. ACM* **2013**, *60* (6), 45:1–45:39. https://doi.org/10.1145/2512329.

(17) Eckart, C.; Young, G. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika* **1936**, *1* (3), 211–218. https://doi.org/10.1007/BF02288367.

(18) Markovsky, I. *Low Rank Approximation: Algorithms, Implementation, Applications*; Communications and Control Engineering; Springer: London, 2012. https://doi.org/10.1007/978-1-4471-2227-2.

(19) Goodall, C. R. 13 Computation Using the QR Decomposition. *Handbook of Statistics*, 1993, *9*, 467–508. https://doi.org/10.1016/S0169-7161(05)80137-3.

(20) Householder, A. S. Unitary Triangularization of a Nonsymmetric Matrix. *Journal of the ACM* **1958**, *5* (4), 339–342. https://doi.org/10.1145/320941.320947.

(21) Anderson, E.; Bai, Z.; Bischof, C.; Blackford, L. S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK Users' Guide*, Third.; Society for Industrial and Applied Mathematics, 1999. https://doi.org/10.1137/1.9780898719604.

(22) E. Anderson,; Z. Bai,; C. Bischof,; J. Demmel,; J. Dongarra,; J. Du Croz,; A. Greenbaum,; S. Hammarling,; A. McKenney,; S. Ostrouchov,; D. Sorensen. *LAPACK Benchmark*. https://netlib.org/lapack/lug_old/node70.html (accessed 2025-06-27).

(23) Francis, J. G. F. The QR Transformation A Unitary Analogue to the LR Transformation —Part 1. *The Computer Journal* **1961**, *4* (3), 265–271. https://doi.org/10.1093/comjnl/4.3.265.

(24) Gu, M.; Eisenstat, S. C. A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem. *SIAM Journal on Matrix Analysis and Applications* **1995**, *16* (1), 172–191. https://doi.org/10.1137/S0895479892241287.

(25) Cuppen, J. J. M. A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem. *Numerische Mathematik* **1980**, *36* (2), 177–195. https://doi.org/10.1007/BF01396757.

(26) Businger, P.; Golub, G. H. Linear Least Squares Solutions by Householder Transformations. *Numerische Mathematik* **1965**, *7* (3), 269–276. https://doi.org/10.1007/BF01436084.

(27) Xiao, J.; Gu, M.; Langou, J. Fast Parallel Randomized QR with Column Pivoting Algorithms for Reliable Low-Rank Matrix Approximations. In *2017 IEEE 24th International Conference on High Performance Computing (HiPC)*; 2017; pp 233–242. https://doi.org/10.1109/HiPC.2017.00035.

(28) Duersch, J. A.; Gu, M. Randomized Projection for Rank-Revealing Matrix Factorizations and Low-Rank Approximations. *SIAM Review* **2020**, *62* (3), 661–682. https://doi.org/10.1137/20M1335571.

(29) Quintana-Ortí, G.; Sun, X.; Bischof, C. H. A BLAS-3 Version of the QR Factorization with Column Pivoting. *SIAM Journal on Scientific Computing* **1998**, *19* (5), 1486–1494. https://doi.org/10.1137/S1064827595296732.

(30) Mijić, N.; Kaushik, A.; Davidović, D. *QR Factorization of Ill-Conditioned Tall-and-Skinny Matrices on Distributed-Memory Systems*. https://doi.org/10.48550/arXiv.2405.04237.

(31) Melnichenko, M.; Murray, R.; Killian, W.; Demmel, J.; Mahoney, M. W.; Luszczek, P.; Gates, M. *Anatomy of High-Performance Column-Pivoted QR Decomposition*. https://doi.org/10.48550/arXiv.2507.00976.

(32) Harbrecht, H.; Peters, M.; Schneider, R. On the Low-Rank Approximation by the Pivoted Cholesky Decomposition. *Applied Numerical Mathematics* **2012**, *62* (4), 428–440. https://doi.org/10.1016/j.apnum.2011.10.001.

(33) Chen, Y.; Epperly, E. N.; Tropp, J. A.; Webber, R. J. *Randomly Pivoted Cholesky: Practical Approximation of a Kernel Matrix with Few Entry Evaluations*. https://doi.org/10.48550/arXiv.2207.06503.

(34) Epperly, E. N.; Moreno, E. Kernel Quadrature with Randomly Pivoted Cholesky. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*; NIPS '23; Curran Associates Inc.: Red Hook, NY, USA, 2023; pp 65850–65868.

(35) Vishwakarma, P. K. *Cholesky Decomposition for Symmetric Matrices over Finite Fields*. https://doi.org/10.48550/arXiv.2508.04657.

(36) Epperly, E. N.; Tropp, J. A.; Webber, R. J. *Embrace Rejection: Kernel Matrix Approximation by Accelerated Randomly Pivoted Cholesky*. https://doi.org/10.48550/arXiv.2410.03969.

(37) Williams, C.; Seeger, M. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems*; MIT Press, 2000; Vol. 13.

(38) Dong, Y.; Chen, C.; Martinsson, P.-G.; Pearce, K. *Robust Blockwise Random Pivoting: Fast and Accurate Adaptive Interpolative Decomposition*. https://doi.org/10.48550/arXiv.2309.16002.

(39) Advani, R.; O'Hagan, S. *Efficient Algorithms for Constructing an Interpolative Decomposition*. https://doi.org/10.48550/arXiv.2105.07076.

(40) Liberty, E.; Woolfe, F.; Martinsson, P.-G.; Rokhlin, V.; Tygert, M. Randomized Algorithms for the Low-Rank Approximation of Matrices. *Proceedings of the National Academy of Sciences* **2007**, *104* (51), 20167–20172. https://doi.org/10.1073/pnas.0709640104.

(41) Woolfe, F.; Liberty, E.; Rokhlin, V.; Tygert, M. A Fast Randomized Algorithm for the Approximation of Matrices. *Applied and Computational Harmonic Analysis* **2008**, *25* (3), 335–366. https://doi.org/10.1016/j.acha.2007.12.002.

(42) Hitchcock, F. L. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics* **1927**, *6* (1–4), 164–189. https://doi.org/10.1002/sapm192761164.

(43) Xie, W.; Xu, J.; Peng, Z.; Wang, Q. *Efficient QR-Based CP Decomposition Acceleration via Dimension Tree and Extrapolation*. https://doi.org/10.48550/arXiv.2503.18759.

(44) Minster, R.; Viviano, I.; Liu, X.; Ballard, G. *CP Decomposition for Tensors via Alternating Least Squares with QR Decomposition*. https://doi.org/10.48550/arXiv.2112.10855.

(45) Erichson, N. B.; Manohar, K.; Brunton, S. L.; Kutz, J. N. *Randomized CP Tensor Decomposition*. https://doi.org/10.48550/arXiv.1703.09074.

(46) Bharadwaj, V.; Malik, O. A.; Murray, R.; Buluç, A.; Demmel, J. Distributed-Memory Randomized Algorithms for Sparse Tensor CP Decomposition. In *Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures*; SPAA '24; Association for Computing Machinery: New York, NY, USA, 2024; pp 155–168. https://doi.org/10.1145/3626183.3659980.

(47) Battaglino, C.; Ballard, G.; Kolda, T. G. A Practical Randomized CP Tensor Decomposition. *SIAM Journal on Matrix Analysis and Applications* **2018**, *39* (2), 876–901. https://doi.org/10.1137/17M1112303.

(48) Singh, N.; Ma, L.; Yang, H.; Solomonik, E. *Comparison of Accuracy and Scalability of Gauss-Newton and Alternating Least Squares for CP Decomposition*. https://doi.org/10.48550/arXiv.1910.12331.

(49) Leach, A. R. *Molecular Modelling : Principles and Applications*; Harlow, England ; New York : Prentice Hall, 2001.

(50) Møller, C.; Plesset, M. S. Note on an Approximation Treatment for Many-Electron Systems. *Physical Review* **1934**, *46* (7), 618–622. https://doi.org/10.1103/PhysRev.46.618.

(51) Grimme, S. Improved Second-Order Møller–Plesset Perturbation Theory by Separate Scaling of Parallel- and Antiparallel-Spin Pair Correlation Energies. *The Journal of Chemical Physics* **2003**, *118* (20), 9095–9102. https://doi.org/10.1063/1.1569242.

(52) Almlöf, J. Elimination of Energy Denominators in Møller—Plesset Perturbation Theory by a Laplace Transform Approach. *Chemical Physics Letters* **1991**, *181* (4), 319–320. https://doi.org/10.1016/0009-2614(91)80078-C.

(53) Grimme, S.; Goerigk, L.; Fink, R. F. Spin-Component-Scaled Electron Correlation Methods. *WIREs Computational Molecular Science* **2012**, *2* (6), 886–906. https://doi.org/10.1002/wcms.1110.

(54) Jung, Y.; Lochan, R. C.; Dutoi, A. D.; Head-Gordon, M. Scaled Opposite-Spin Second Order Møller–Plesset Correlation Energy: An Economical Electronic Structure Method. *The Journal of Chemical Physics* **2004**, *121* (20), 9793–9802. https://doi.org/10.1063/1.1809602.

(55) Crawford, T. D.; Schaefer III, H. F. An Introduction to Coupled Cluster Theory for Computational Chemists. *Reviews in Computational Chemistry*, 2000, 33–136. https://doi.org/10.1002/9780470125915.ch2.

(56) Qin, X.; Hu, W.; Yang, J. Interpolative Separable Density Fitting for Accelerating Two-Electron Integrals: A Theoretical Perspective. *Journal of Chemical Theory and Computation* **2023**, *19* (3), 679–693. https://doi.org/10.1021/acs.jctc.2c00927.

(57) Thompson, T. H.; Ochsenfeld, C. Distance-Including Rigorous Upper Bounds and Tight Estimates for Two-Electron Integrals over Long- and Short-Range Operators. *The Journal of Chemical Physics* **2017**, *147* (14), 144101. https://doi.org/10.1063/1.4994190.

(58) Whitten, J. L. Coulombic Potential Energy Integrals and Approximations. *The Journal of Chemical Physics* **1973**, *58* (10), 4496–4501. https://doi.org/10.1063/1.1679012.

(59) Häser, M.; Ahlrichs, R. Improvements on the Direct SCF Method. *Journal of Computational Chemistry* **1989**, *10* (1), 104–111. https://doi.org/10.1002/jcc.540100111.

(60) Kohn, W. Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms. *Physical Review Letters* **1996**, *76* (17), 3168–3171. https://doi.org/10.1103/PhysRevLett.76.3168.

(61) White, C. A.; Head-Gordon, M. Derivation and Efficient Implementation of the Fast Multipole Method. *The Journal of Chemical Physics* **1994**, *101* (8), 6593–6605. https://doi.org/10.1063/1.468354.

(62) Irmler, A.; Pauly, F. Multipole-Based Distance-Dependent Screening of Coulomb Integrals. *The Journal of Chemical Physics* **2019**, *151* (8), 84111. https://doi.org/10.1063/1.5111054.

(63) Neese, F.; Colinet, P.; DeSouza, B.; Helmich-Paris, B.; Wennmohs, F.; Becker, U. The "Bubblepole" (BUPO) Method for Linear-Scaling Coulomb Matrix Construction with or Without Density Fitting. *The Journal of Physical Chemistry A* **2025**, *129* (10), 2618–2637. https://doi.org/10.1021/acs.jpca.4c07415.

(64) Baerends, E. J.; Ellis, D. E.; Ros, P. Self-Consistent Molecular Hartree—Fock—Slater Calculations I. The Computational Procedure. *Chemical Physics* **1973**, *2* (1), 41–51. https://doi.org/10.1016/0301-0104(73)80059-X.

(65) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. On Some Approximations in Applications of Xα Theory. *The Journal of Chemical Physics* **1979**, *71* (8), 3396–3402. https://doi.org/10.1063/1.438728.

(66) Manby, F. R. Density Fitting in Second-Order Linear-R12 Møller–Plesset Perturbation Theory. *The Journal of Chemical Physics* **2003**, *119* (9), 4607–4613. https://doi.org/10.1063/1.1594713.

(67) Pedersen, T. B.; Aquilante, F.; Lindh, R. Density Fitting with Auxiliary Basis Sets from Cholesky Decompositions. *Theoretical Chemistry Accounts* **2009**, *124* (1), 1–10. https://doi.org/10.1007/s00214-009-0608-y.

(68) Varga, Š. On Robust Density Fitting in Molecules and Extended Systems. *Journal of Mathematical Chemistry* **2011**, *49* (1), 1–5. https://doi.org/10.1007/s10910-010-9736-9.

(69) Pedersen, T. B.; Lehtola, S.; Fdez. Galván, I.; Lindh, R. The Versatility of the Cholesky Decomposition in Electronic Structure Theory. *WIREs Computational Molecular Science* **2023**, *14* (1), e1692. https://doi.org/10.1002/wcms.1692.

(70) Vahtras, O.; Almlöf, J.; Feyereisen, M. W. Integral Approximations for LCAO-SCF Calculations. *Chemical Physics Letters* **1993**, *213* (5), 514–518. https://doi.org/10.1016/0009-2614(93)89151-7.

(71) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams-Young, D.; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.;

Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. Gaussian 16 Revision C.01, 2016.

(72) Neese, F. The ORCA Program System. *WIRES Comput. Molec. Sci.* **2012**, *2* (1), 73–78. https://doi.org/10.1002/wcms.81.

(73) Sun, Q.; Zhang, X.; Banerjee, S.; Bao, P.; Barbry, M.; Blunt, N. S.; Bogdanov, N. A.; Booth, G. H.; Chen, J.; Cui, Z.-H.; Eriksen, J. J.; Gao, Y.; Guo, S.; Hermann, J.; Hermes, M. R.; Koh, K.; Koval, P.; Lehtola, S.; Li, Z.; Liu, J.; Mardirossian, N.; McClain, J. D.; Motta, M.; Mussard, B.; Pham, H. Q.; Pulkin, A.; Purwanto, W.; Robinson, P. J.; Ronca, E.; Sayfutyarova, E. R.; Scheurer, M.; Schurkus, H. F.; Smith, J. E. T.; Sun, C.; Sun, S.-N.; Upadhyay, S.; Wagner, L. K.; Wang, X.; White, A.; Whitfield, J. D.; Williamson, M. J.; Wouters, S.; Yang, J.; Yu, J. M.; Zhu, T.; Berkelbach, T. C.; Sharma, S.; Sokolov, A. Y.; Chan, G. K.-L. Recent Developments in the PySCF Program Package. *The Journal of Chemical Physics* **2020**, *153* (2), 24109. https://doi.org/10.1063/5.0006074.

(74) Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. RI-MP2: Optimized Auxiliary Basis Sets and Demonstration of Efficiency. *Chemical Physics Letters* **1998**, *294* (1), 143–152. https://doi.org/10.1016/S0009-2614(98)00862-8.

(75) Hampel, C.; Werner, H. Local Treatment of Electron Correlation in Coupled Cluster Theory. *The Journal of Chemical Physics* **1996**, *104* (16), 6286–6297. https://doi.org/10.1063/1.471289.

(76) Weigend, F.; Köhn, A.; Hättig, C. Efficient Use of the Correlation Consistent Basis Sets in Resolution of the Identity MP2 Calculations. *The Journal of Chemical Physics* **2002**, *116* (8), 3175–3183. https://doi.org/10.1063/1.1445115.

(77) Yang, R.; Rendell, A. P.; Frisch, M. J. Automatically Generated Coulomb Fitting Basis Sets: Design and Accuracy for Systems Containing H to Kr. *The Journal of Chemical Physics* **2007**, *127* (7), 74102. https://doi.org/10.1063/1.2752807.

(78) Stoychev, G. L.; Auer, A. A.; Neese, F. Automatic Generation of Auxiliary Basis Sets. *Journal of Chemical Theory and Computation* **2017**, *13* (2), 554–562. https://doi.org/10.1021/acs.jctc.6b01041.

(79) Antonini, N.; Ronca, E.; Storchi, L.; Belpassi, L. *Automatic Generation of Density Fitting Auxiliary Basis Sets for All Electron Dirac-Kohn-Sham Calculations*. https://doi.org/10.48550/arXiv.2506.15631.

(80) Weigend, F.; Kattannek, M.; Ahlrichs, R. Approximated Electron Repulsion Integrals: Cholesky Decomposition Versus Resolution of the Identity Methods. *The Journal of Chemical Physics* **2009**, *130* (16), 164106. https://doi.org/10.1063/1.3116103.

(81) Friesner, R. A. Solution of Self-Consistent Field Electronic Structure Equations by a Pseudospectral Method. *Chemical Physics Letters* **1985**, *116* (1), 39–43. https://doi.org/10.1016/0009-2614(85)80121-4.

(82) Matthews, D. A. High-Performance Tensor Contraction Without Transposition. *SIAM Journal on Scientific Computing* **2018**, *40* (1), C1–C24. https://doi.org/10.1137/16M108968X.

(83) Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. Tensor Hypercontraction. II. Least-squares Renormalization. *The Journal of Chemical Physics* **2012**, *137* (22), 224106. https://doi.org/10.1063/1.4768233.

(84) Bangerter, F. H.; Glasbrenner, M.; Ochsenfeld, C. Tensor-Hypercontracted MP2 First Derivatives: Runtime and Memory Efficient Computation of Hyperfine Coupling Constants. *Journal of Chemical Theory and Computation* **2022**, *18* (9), 5233–5245. https://doi.org/10.1021/acs.jctc.2c00118.

(85) Dunning, T. H., Jr. Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron Through Neon and Hydrogen. *The Journal of Chemical Physics* **1989**, *90* (2), 1007–1023. https://doi.org/10.1063/1.456153.

(86) Becke, A. D. A Multicenter Numerical Integration Scheme for Polyatomic Molecules. *The Journal of Chemical Physics* **1988**, *88* (4), 2547–2553. https://doi.org/10.1063/1.454033.

(87) Matthews, D. A. A Critical Analysis of Least-Squares Tensor Hypercontraction Applied to MP3. *The Journal of Chemical Physics* **2021**, *154* (13), 134102. https://doi.org/10.1063/5.0038764.

(88) Zhao, T.; Simons, M.; Matthews, D. A. Open-Shell Tensor Hypercontraction. *Journal of Chemical Theory and Computation* **2023**, *19* (13), 3996–4010. https://doi.org/10.1021/acs.jctc.3c00392.

(89) Bangerter, F. H.; Glasbrenner, M.; Ochsenfeld, C. Low-Scaling Tensor Hypercontraction in the Cholesky Molecular Orbital Basis Applied to Second-Order Møller–Plesset Perturbation Theory. *Journal of Chemical Theory and Computation* **2021**, *17* (1), 211–221. https://doi.org/10.1021/acs.jctc.0c00934.

(90) Nejad, A.; Zhu, A.; Sorathia, K.; Tew, D. P. *DLPNO-MP2 with Periodic Boundary Conditions*. https://doi.org/10.48550/arXiv.2507.09810.

(91) Hohenstein, E. G.; Parrish, R. M.; Sherrill, C. D.; Martínez, T. J. Communication: Tensor Hypercontraction. III. Least-squares Tensor Hypercontraction for the Determination of Correlated Wavefunctions. *The Journal of Chemical Physics* **2012**, *137* (22), 221101. https://doi.org/10.1063/1.4768241.

(92) Hohenstein, E. G.; Fales, B. S.; Parrish, R. M.; Martínez, T. J. Rank-Reduced Coupled-Cluster. III. Tensor Hypercontraction of the Doubles Amplitudes. *The Journal of Chemical Physics* **2022**, *156* (5), 54102. https://doi.org/10.1063/5.0077770.

(93) Datar, A.; Matthews, D. A. Robust Tensor Hypercontraction of the Particle-Particle Ladder Term in Equation-of-Motion Coupled Cluster Theory. *Journal of Chemical Theory and Computation* **2024**, *20* (2), 708–720. https://doi.org/10.1021/acs.jctc.3c00892.

(94) Parrish, R. M.; Liu, F.; Martínez, T. J. Communication: A Difference Density Picture for the Self-Consistent Field Ansatz. *The Journal of Chemical Physics* **2016**, *144* (13), 131101. https://doi.org/10.1063/1.4945277.

(95) Hillers-Bendtsen, A. E.; Martínez, T. J. Lowering the Scaling of Self-Consistent Field Methods by Combining Tensor Hypercontraction and a Density Difference Ansatz. *The Journal of Physical Chemistry Letters* **2025**, *16* (19), 4734–4739. https://doi.org/10.1021/acs.jpclett.5c00405.

(96) Hillers-Bendtsen, A. E.; Martínez, T. J. *Accelerating Hartree-Fock and Density Functional Theory Calculations Using Tensor Hypercontraction*. https://doi.org/10.48550/arXiv.2508.19212.

(97) Pokhilko, P.; Yeh, C.-N.; Morales, M. A.; Zgid, D. Tensor Hypercontraction for Fully Self-Consistent Imaginary-Time GF2 and GWSOX Methods: Theory, Implementation, And Role of the Green's Function Second-Order Exchange for Intermolecular Interactions. *The Journal of Chemical Physics* **2024**, *161* (8), 84108. https://doi.org/10.1063/5.0215954.

(98) Yeh, C.-N.; Morales, M. A. Low-Scaling Algorithms for GW and Constrained Random Phase Approximation Using Symmetry-Adapted Interpolative Separable Density Fitting. *Journal of Chemical Theory and Computation* **2024**, *20* (8), 3184–3198. https://doi.org/10.1021/acs.jctc.4c00085.

(99) Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. Discrete Variable Representation in Electronic Structure Theory: Quadrature Grids for Least-Squares Tensor Hypercontraction. *The Journal of Chemical Physics* **2013**, *138* (19), 194107. https://doi.org/10.1063/1.4802773.

(100) Kokkila Schumacher, S. I. L.; Hohenstein, E. G.; Parrish, R. M.; Wang, L.-P.; Martínez, T. J. Tensor Hypercontraction Second-Order Møller–Plesset Perturbation Theory: Grid Optimization and Reaction Energies. *Journal of Chemical Theory and Computation* **2015**, *11* (7), 3042–3052. https://doi.org/10.1021/acs.jctc.5b00272.

(101) Smyser, K. E.; White, A.; Sharma, S. *Use of Multigrids to Reduce the Cost of Performing Interpolative Separable Density Fitting*. https://doi.org/10.48550/arXiv.2404.09373.

(102) Dong, K.; Hu, W.; Lin, L. *Interpolative Separable Density Fitting through Centroidal Voronoi Tessellation With Applications to Hybrid Functional Electronic Structure Calculations*. https://doi.org/10.48550/arXiv.1711.01531.

(103) Lee, J.; Lin, L.; Head-Gordon, M. Systematically Improvable Tensor Hypercontraction: Interpolative Separable Density-Fitting for Molecules Applied to Exact Exchange, Second- and Third-Order Møller-Plesset Perturbation Theory. *Journal of Chemical Theory and Computation* **2020**, *16* (1), 243–263. https://doi.org/10.1021/acs.jctc.9b00820.

(104) Matthews, D. A. Improved Grid Optimization and Fitting in Least Squares Tensor Hypercontraction: Supplemental Information.

(105) Zhang, Y.; Xiong, X.; Wu, W.; Su, P. *Block Tensor Decomposition: A Dual Grid Scheme with Formal O(N3) for THC Decomposition of Molecular Systems*. https://doi.org/10.48550/arXiv.2506.19392.

(106) Matthews, D. A. Improved Grid Optimization and Fitting in Least Squares Tensor Hypercontraction. *Journal of Chemical Theory and Computation* **2020**, *16* (3), 1382–1385. https://doi.org/10.1021/acs.jctc.9b01205.

(107) Karton, A.; Daon, S.; Martin, J. M. L. W4-11: A High-Confidence Benchmark Dataset for Computational Thermochemistry Derived from First-Principles W4 Data. *Chemical Physics Letters* **2011**, *510* (4), 165–178. https://doi.org/10.1016/j.cplett.2011.05.007.

(108) Řezáč, J.; Hobza, P. Describing Noncovalent Interactions Beyond the Common Approximations: How Accurate Is the "Gold Standard," CCSD(T) at the Complete Basis Set Limit?. *Journal of Chemical Theory and Computation* **2013**, *9* (5), 2151–2155. https://doi.org/10.1021/ct400057w.

(109) Shao, Y.; Gan, Z.; Epifanovsky, E.; Gilbert, A. T.; Wormit, M.; Kussmann, J.; Lange, A. W.; Behn, A.; Deng, J.; Feng, X.; Ghosh, D.; Goldey, M.; Horn, P. R.; Jacobson, L. D.; Kaliman, I.; Khaliullin, R. Z.; Kuś, T.; Landau, A.; Liu, J.; Proynov, E. I.; Rhee, Y. M.;

Richard, R. M.; Rohrdanz, M. A.; Steele, R. P.; Sundstrom, E. J.; Woodcock III, H. L.; Zimmerman, P. M.; Zuev, D.; Albrecht, B.; Alguire, E.; Austin, B.; Beran, G. J. O.; Bernard, Y. A.; Berquist, E.; Brandhorst, K.; Bravaya, K. B.; Brown, S. T.; Casanova, D.; Chang, C.-M.; Chen, Y.; Chien, S. H.; Closser, K. D.; Crittenden, D. L.; Diedenhofen, M.; DiStasio Jr., R. A.; Do, H.; Dutoi, A. D.; Edgar, R. G.; Fatehi, S.; Fusti-Molnar, L.; Ghysels, A.; Golubeva-Zadorozhnaya, A.; Gomes, J.; Hanson-Heine, M. W.; Harbach, P. H.; Hauser, A. W.; Hohenstein, E. G.; Holden, Z. C.; Jagau, T.-C.; Ji, H.; Kaduk, B.; Khistyaev, K.; Kim, J.; Kim, J.; King, R. A.; Klunzinger, P.; Kosenkov, D.; Kowalczyk, T.; Krauter, C. M.; Lao, K. U.; Laurent, A. D.; Lawler, K. V.; Levchenko, S. V.; Lin, C. Y.; Liu, F.; Livshits, E.; Lochan, R. C.; Luenser, A.; Manohar, P.; Manzer, S. F.; Mao, S.-P.; Mardirossian, N.; Marenich, A. V.; Maurer, S. A.; Mayhall, N. J.; Neuscamman, E.; Oana, C. M.; Olivares-Amaya, R.; O'Neill, D. P.; Parkhill, J. A.; Perrine, T. M.; Peverati, R.; Prociuk, A.; Rehn, D. R.; Rosta, E.; Russ, N. J.; Sharada, S. M.; Sharma, S.; Small, D. W.; Sodt, A.; Stein, T.; Stück, D.; Su, Y.-C.; Thom, A. J.; Tsuchimochi, T.; Vanovschi, V.; Vogt, L.; Vydrov, O.; Wang, T.; Watson, M. A.; Wenzel, J.; White, A.; Williams, C. F.; Yang, J.; Yeganeh, S.; Yost, S. R.; You, Z.-Q.; Zhang, I. Y.; Zhang, X.; Zhao, Y.; Brooks, B. R.; Chan, G. K.; Chipman, D. M.; Cramer, C. J.; Goddard III, W. A.; Gordon, M. S.; Hehre, W. J.; Klamt, A.; Schaefer III, H. F.; Schmidt, M. W.; Sherrill, C. D.; Truhlar, D. G.; Warshel, A.; Xu, X.; Aspuru-Guzik, A.; Baer, R.; Bell, A. T.; Besley, N. A.; Chai, J.-D.; Dreuw, A.; Dunietz, B. D.; Furlani, T. R.; Gwaltney, S. R.; Hsu, C.-P.; Jung, Y.; Kong, J.; Lambrecht, D. S.; Liang, W.; Ochsenfeld, C.; Rassolov, V. A.; Slipchenko, L. V.; Subotnik, J. E.; Van Voorhis, T.; Herbert, J. M.; Krylov, A. I.; Gill, P. M.; Head-Gordon, M. Advances in Molecular Quantum Chemistry Contained in the Q-Chem 4 Program Package. *Molecular Physics* **2015**, *113* (2), 184–215. https://doi.org/10.1080/00268976.2014.952696.

(110) Matthews, D. A.; Cheng, L.; Harding, M. E.; Lipparini, F.; Stopkowicz, S.; Jagau, T.-C.; Szalay, P. G.; Gauss, J.; Stanton, J. F. Coupled-Cluster Techniques for Computational Chemistry: The CFOUR Program Package. *The Journal of Chemical Physics* **2020**, *152* (21), 214108. https://doi.org/10.1063/5.0004837.

(111) Van Lenthe, J. H.; Zwaans, R.; Van Dam, H. J. J.; Guest, M. F. Starting SCF Calculations by Superposition of Atomic Densities. *Journal of Computational Chemistry* **2006**, *27* (8), 926–932. https://doi.org/10.1002/jcc.20393.

(112) Seritan, S.; Bannwarth, C.; Fales, B. S.; Hohenstein, E. G.; Kokkila-Schumacher, S. I. L.; Luehr, N.; Snyder, J. W.; Song, C.; Titov, A. V.; Ufimtsev, I. S.; Martínez, T. J. TeraChem: Accelerating Electronic Structure and Ab Initio Molecular Dynamics with Graphical Processing Units. *The Journal of Chemical Physics* **2020**, *152* (22), 224110. https://doi.org/10.1063/5.0007615.

(113) Pierce, K.; Morales, M. *Using Matrix-Free Tensor-Network Optimizations to Construct a Reduced-Scaling and Robust Second-Order Møller-Plesset Theory.* https://doi.org/10.48550/arXiv.2502.17683.

(114) Pierce, K.; Valeev, E. F. Efficient Construction of Canonical Polyadic Approximations of Tensor Networks. *Journal of Chemical Theory and Computation* **2023**, *19* (1), 71–81. https://doi.org/10.1021/acs.jctc.2c00861.

(115) Harris, C. R.; Millman, K. J.; Walt, S. J. van der; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; Kerkwijk, M. H. van; Brett, M.; Haldane, A.; Río, J. F. del; Wiebe, M.; Peterson, P.; Gérard-Marchant,

P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2.

(116) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; Walt, S. J. van der; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; Mulbregt, P. van. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17* (3), 261–272. https://doi.org/10.1038/s41592-019-0686-2.

(117) Sun, Q. Libcint: An Efficient General Integral Library for Gaussian Basis Functions. *Journal of Computational Chemistry* **2015**, *36* (22), 1664–1671. https://doi.org/10.1002/jcc.23981.

(118) Bentley, J. L. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* **1975**, *18* (9), 509–517. https://doi.org/10.1145/361002.361007.

(119) Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A Package for Building Initial Configurations for Molecular Dynamics Simulations. *Journal of Computational Chemistry* **2009**, *30* (13), 2157–2164. https://doi.org/10.1002/jcc.21224.

(120) Hättig, C. Optimization of Auxiliary Basis Sets for RI-MP2 and RI-CC2 Calculations: Core–Valence and Quintuple-Z Basis Sets for H to Ar and QZVPP Basis Sets for Li to Kr. *Physical Chemistry Chemical Physics* **2005**, *7* (1), 59–66. https://doi.org/10.1039/B415208E.

(121) Weigend, F. A Fully Direct RI-HF Algorithm: Implementation, Optimised Auxiliary Basis Sets, Demonstration of Accuracy and Efficiency. *Physical Chemistry Chemical Physics* **2002**, *4* (18), 4285–4291. https://doi.org/10.1039/B204199P.

(122) Pandas Team. *Pandas-Dev/Pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134.

(123) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *Journal of Cheminformatics* **2011**, *3* (1), 33. https://doi.org/10.1186/1758-2946-3-33.

(124) Yeh, C.-N.; Morales, M. A. *Low-Scaling Algorithms for GW and Constrained Random Phase Approximation Using Symmetry-Adapted Interpolative Separable Density Fitting*. https://doi.org/10.48550/arXiv.2401.12308.

A Detailed Benchmarking Results

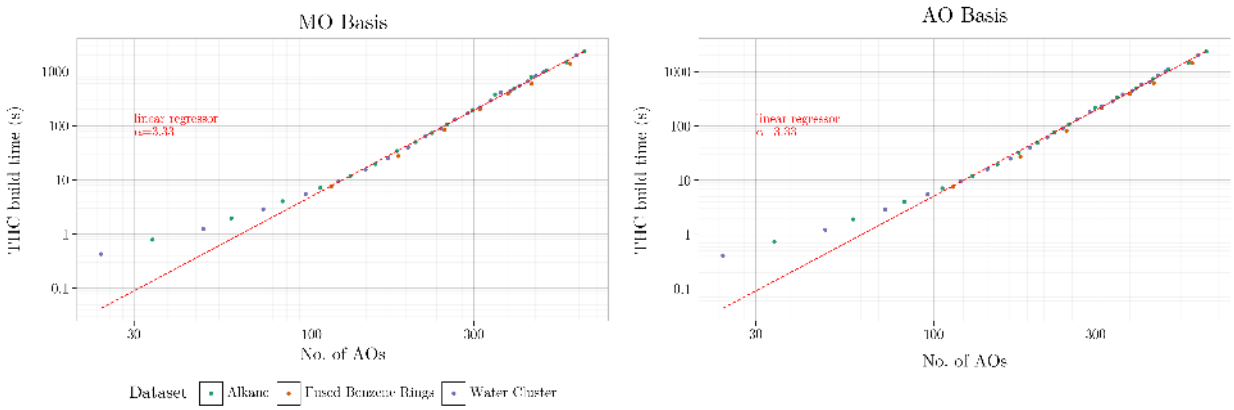


Figure A.1: Performance of the LS-THC method on the test set.

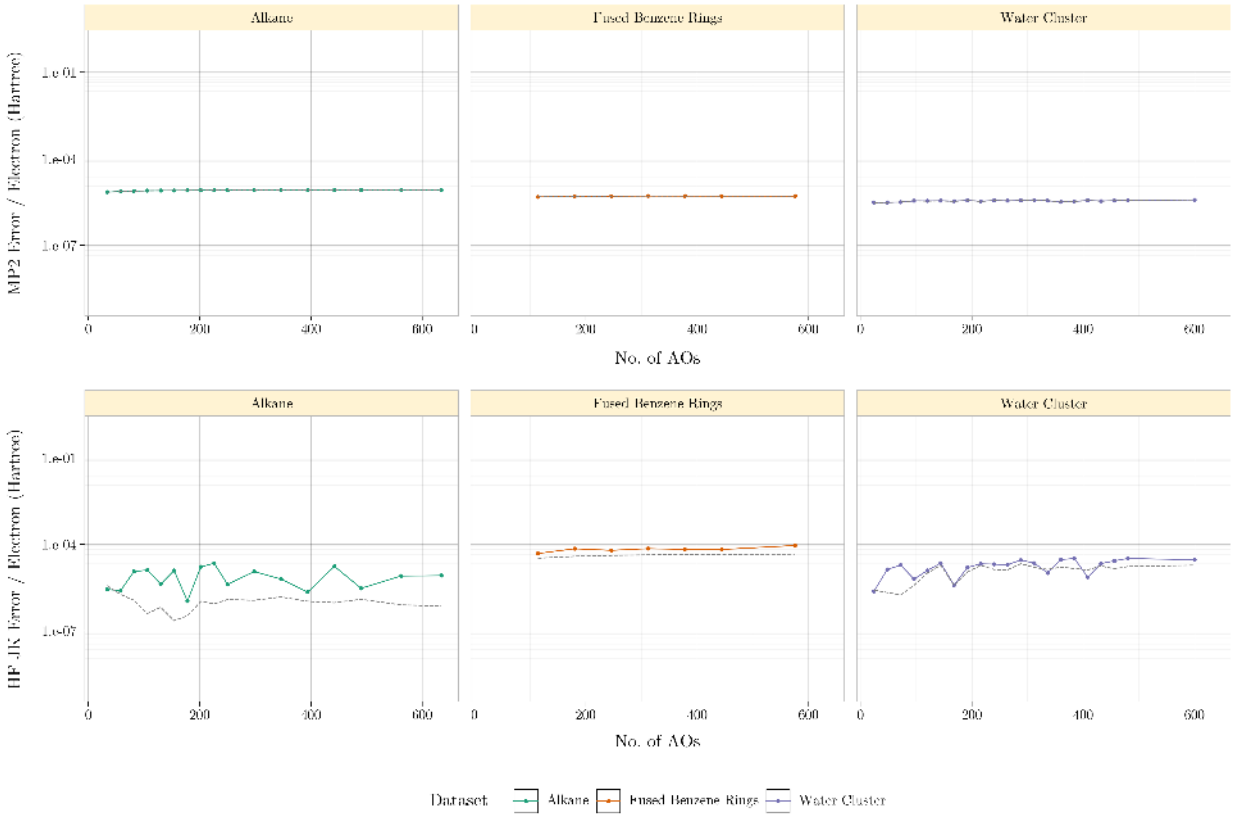


Figure A.2: Error metrics of the LS-THC method on the test set.

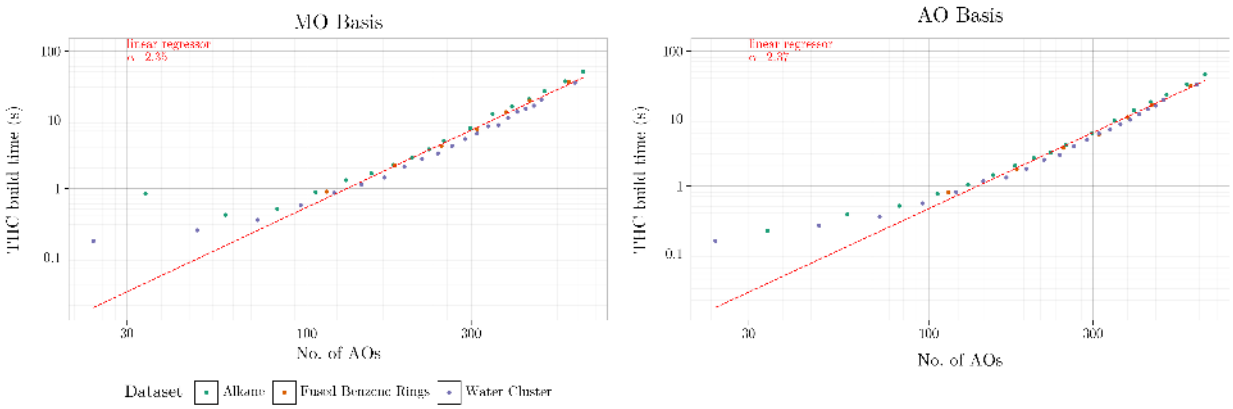


Figure A.3: Performance of the CD-LS-THC method on the test set.

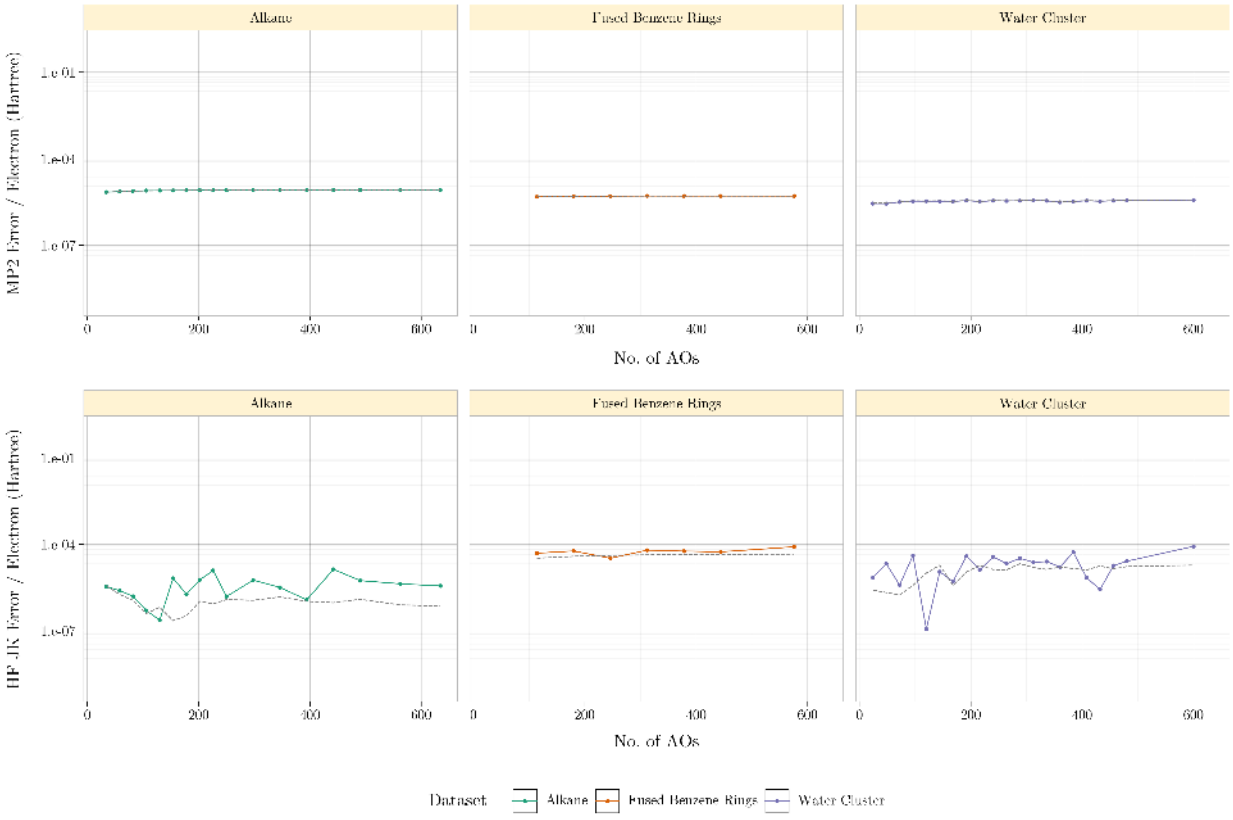


Figure A.4: Error metrics of the CD-LS-THC method on the test set.

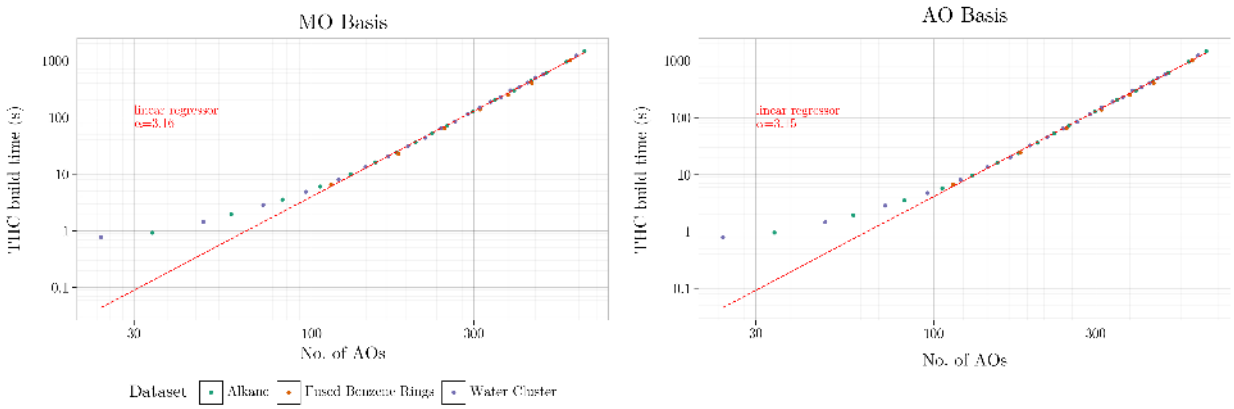


Figure A.5: Performance of the ISDF-QR method on the test set.

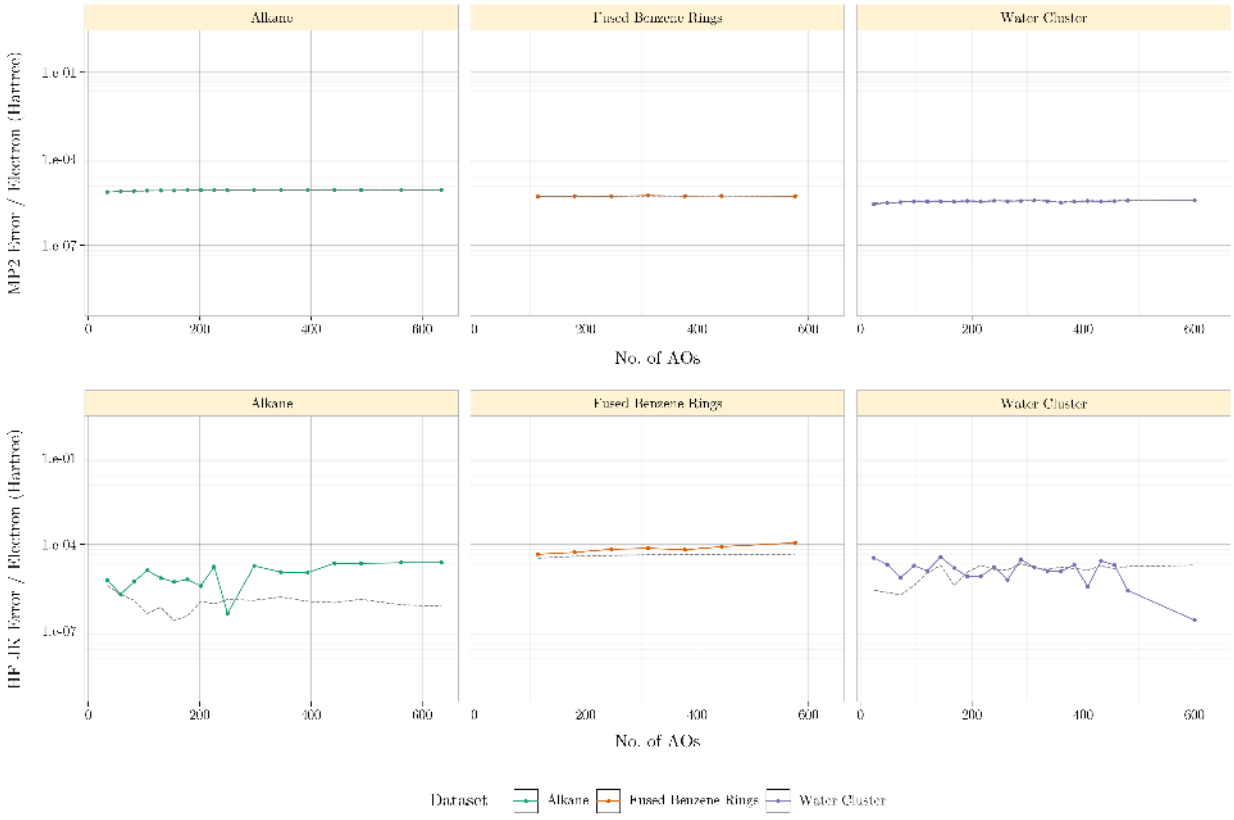


Figure A.6: Error metrics of the ISDF-QR method on the test set.

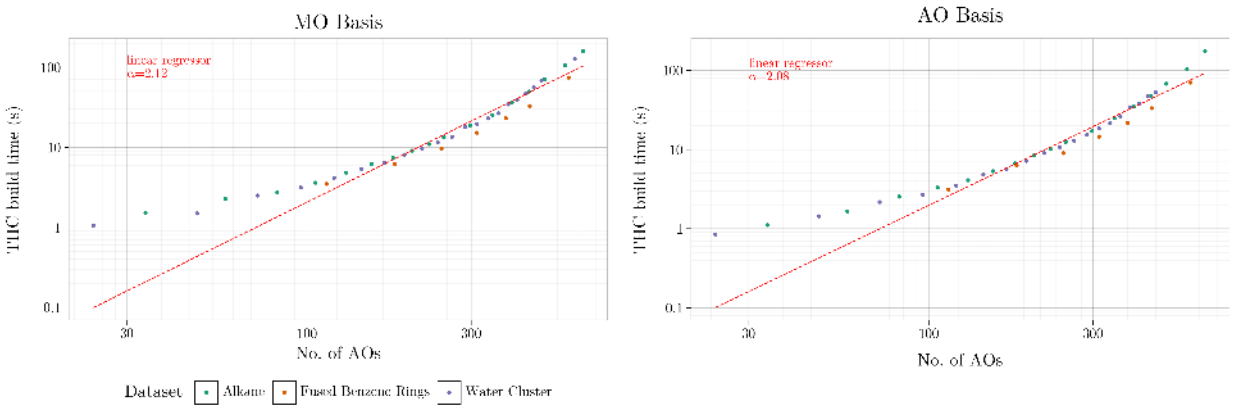


Figure A.7: Performance of the ISDF-KM method on the test set.

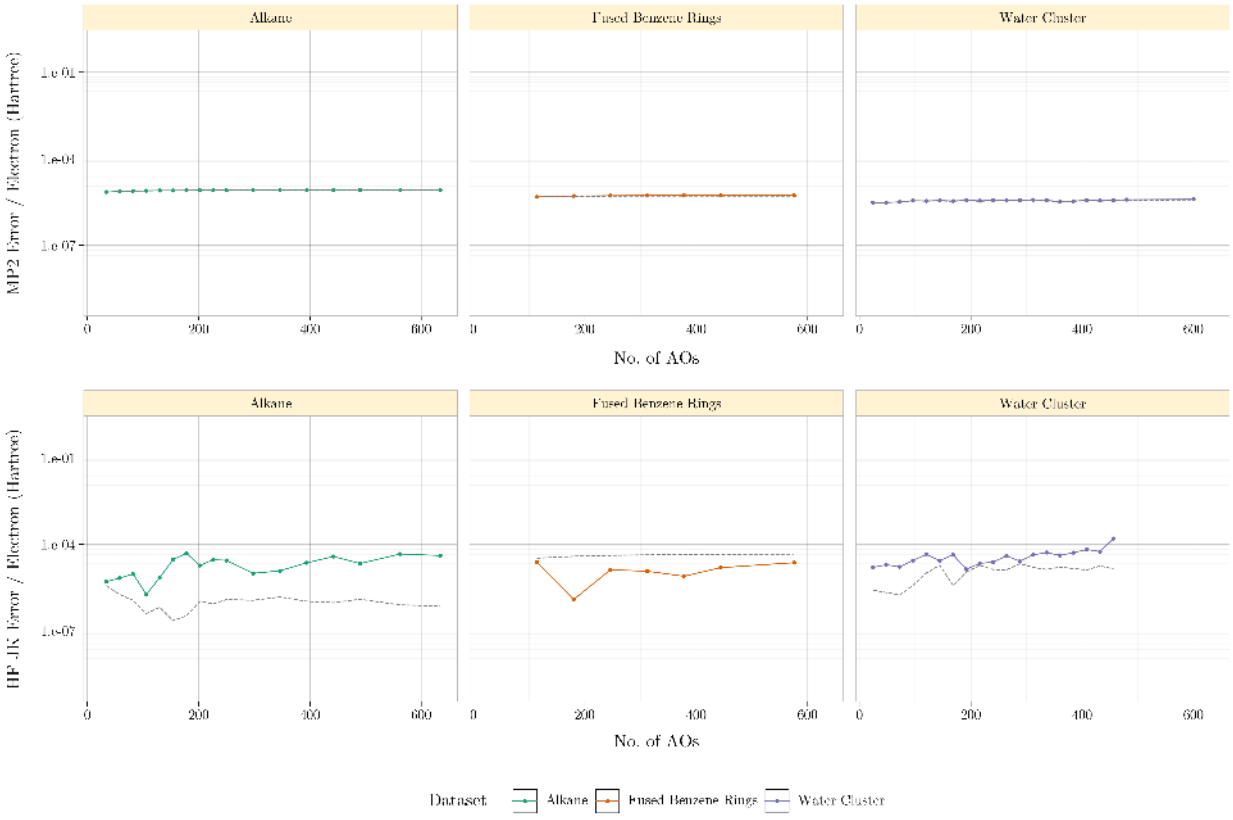


Figure A.8: Error metrics of the ISDF-KM method on the test set.

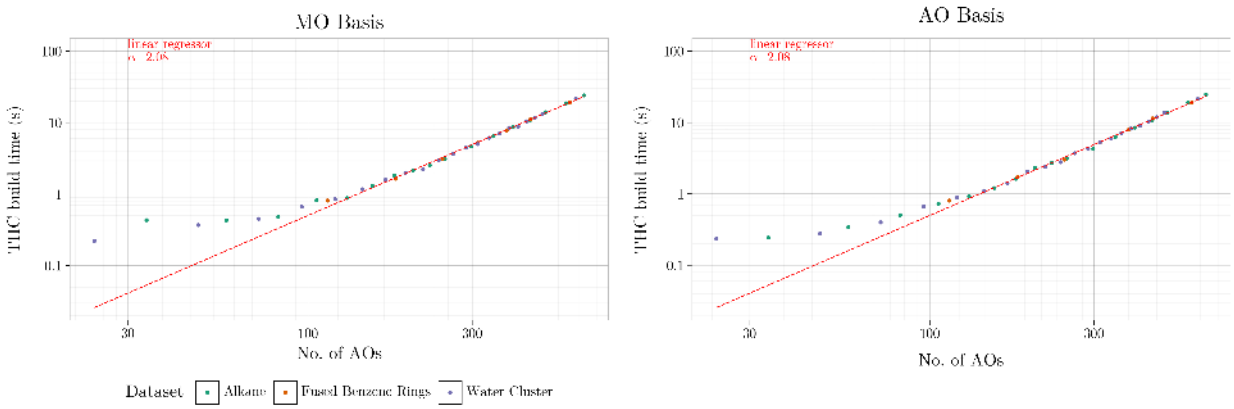


Figure A.9: Performance of the DF-THC method on the test set.

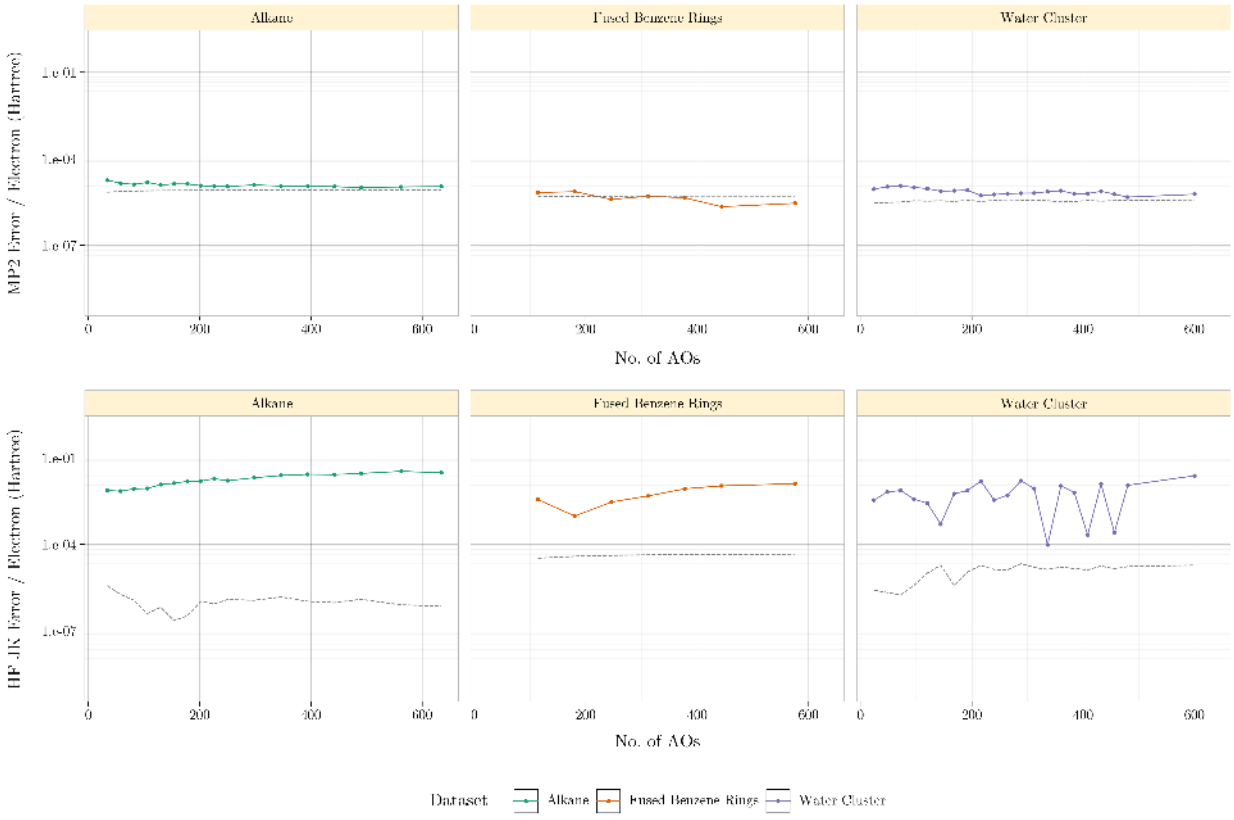


Figure A.10: Error metrics of the DF-THC method on the test set.

B Grid Pruning on Different Molecules

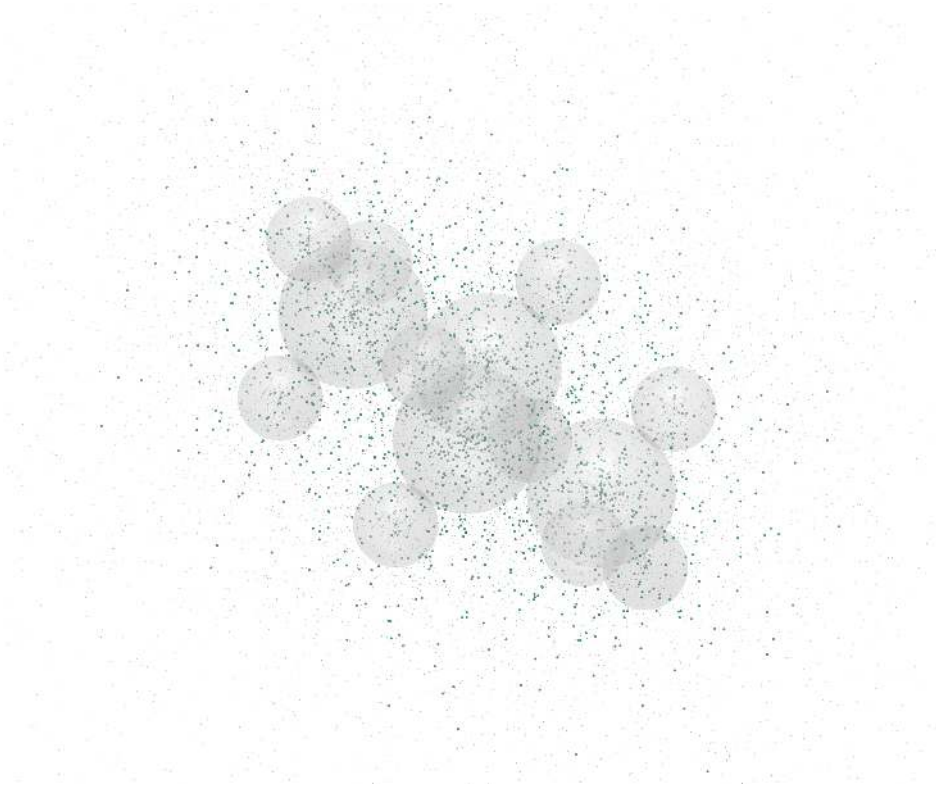


Figure B.1: Pruned integration grid on butane computed by CD-LS-THC.

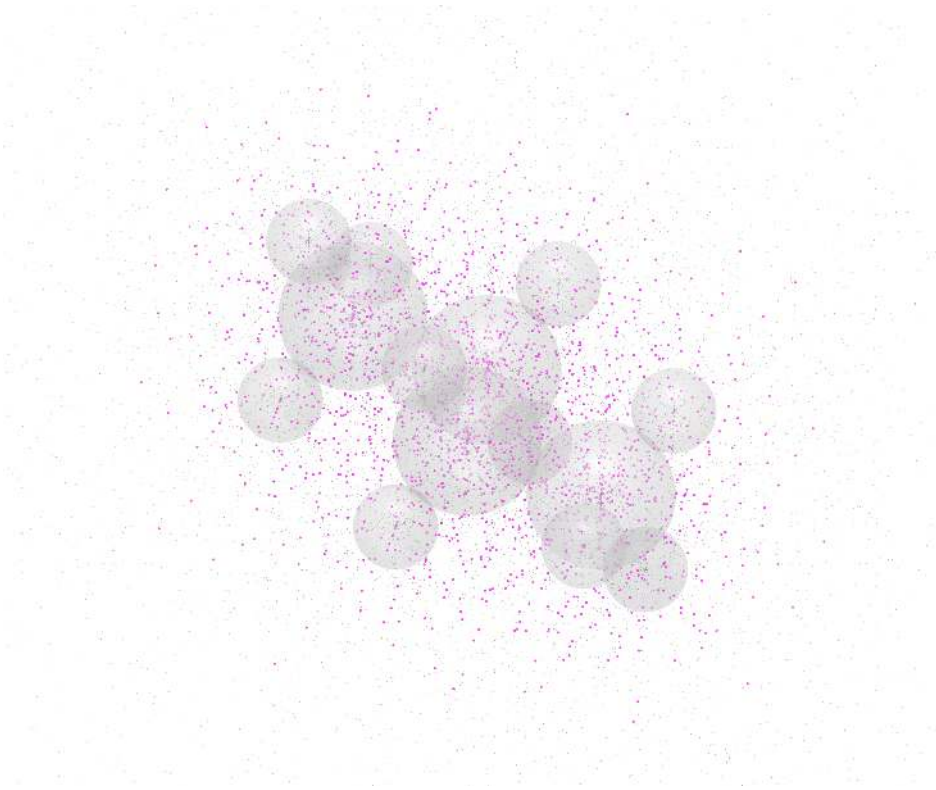


Figure B.2: Pruned integration grid on butane computed by ISDF-QR.

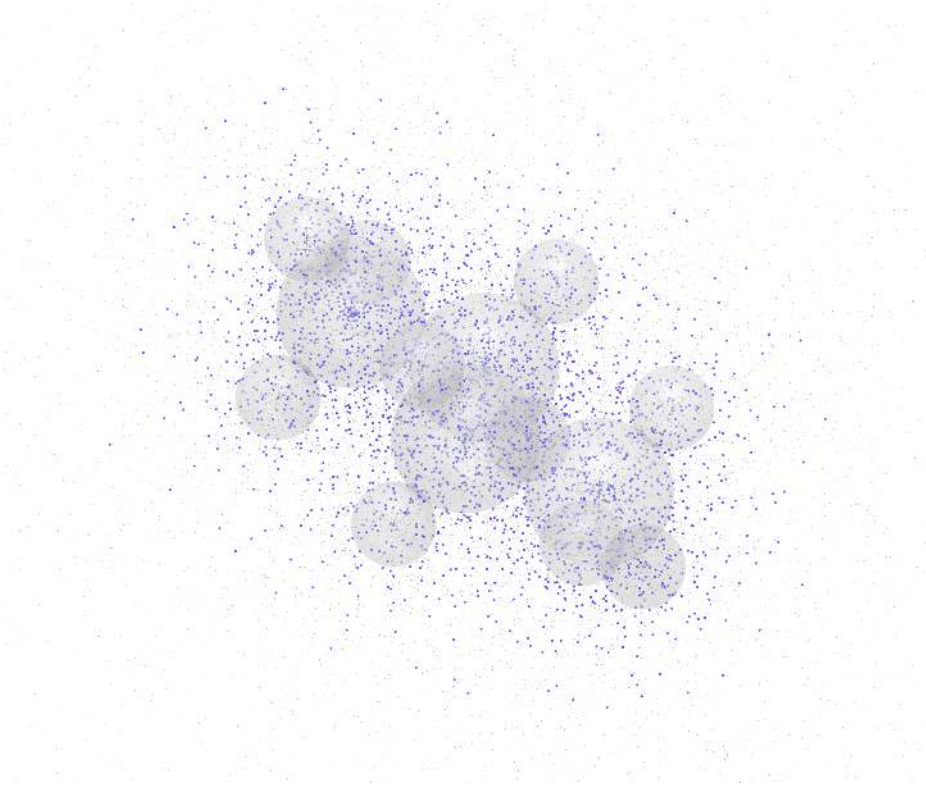


Figure B.3: Pruned integration grid on butane computed by ISDF-KM.

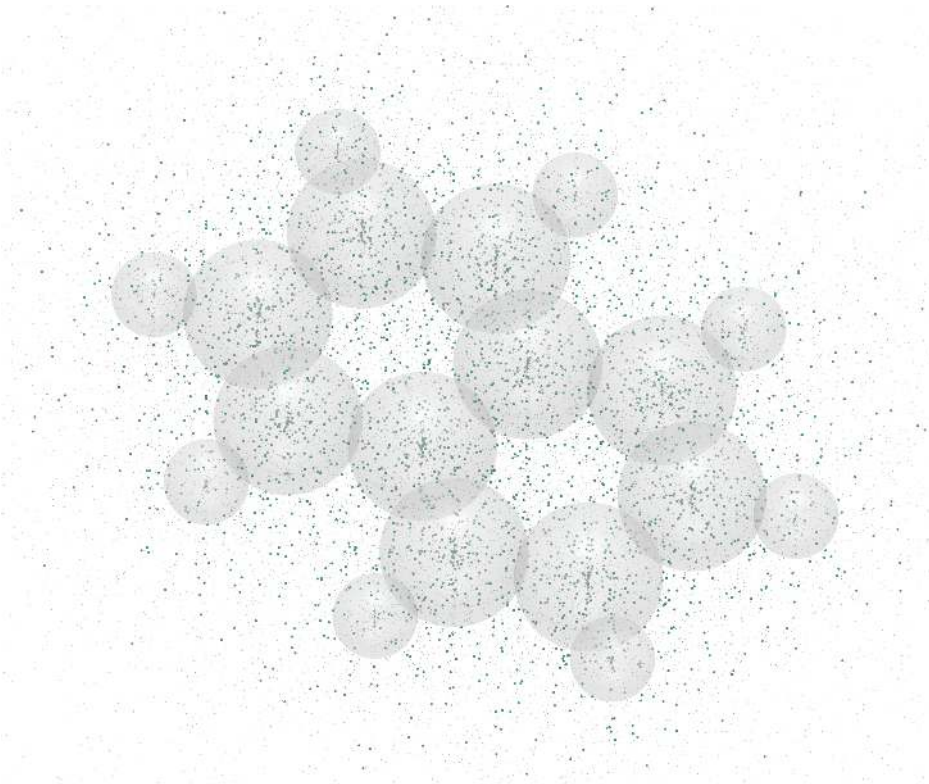


Figure B.4: Pruned integration grid on naphthalene computed by CD-LS-THC.

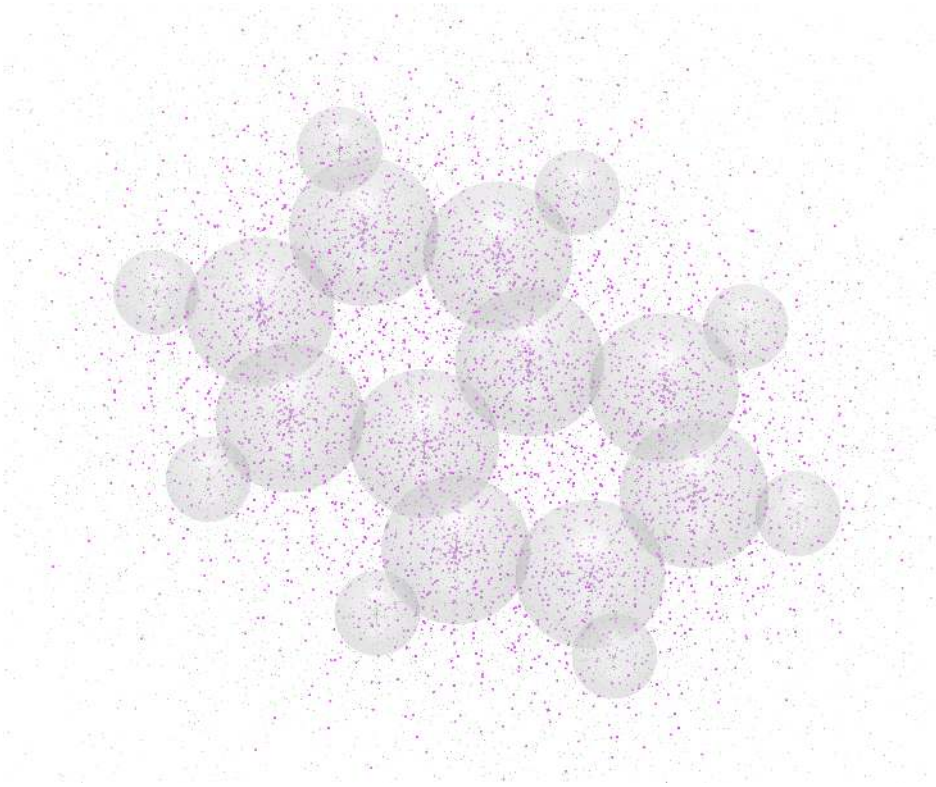


Figure B.5: Pruned integration grid on naphthalene computed by ISDF-QR.

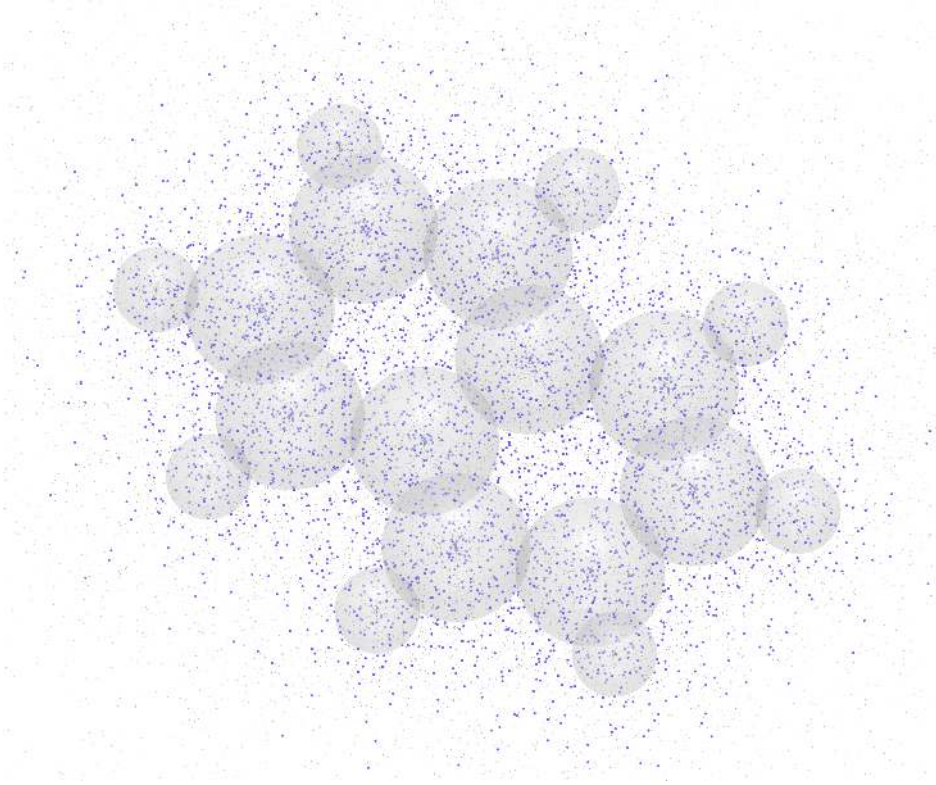


Figure B.6: Pruned integration grid on naphthalene computed by ISDF-KM.

C Used Software and Versions

All Python dependencies were installed using the `uv` package manager[F.6] and can be downloaded from the PyPi[F.7] Python package index. PACKMOL was downloaded from GitHub.

Table C.1: Used software versions

Software	Version
PACKMOL	20.3.5
Blender	4.5.2 LTS
duckdb	1.3.2
numba	2.2.5
numexpr	2.14.1
numpy	2.2.5
pandas	2.2.3
plotnine	0.15.0
PyMol	3.1.0
pyscf	2.9.0
pytblis	0.0.5
opt-einsum	3.4.0
rdkit	2025.3.3
scikit-learn	1.7.1
scipy	1.15.2

[F.6] https://docs.astral.sh/uv/
[F.7] https://pypi.org/

Erklärung über Eigenständigkeit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Bioinformatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe. Sofern im Zuge der Erstellung der vorliegenden Abschlussarbeit generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten Wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Hamburg den 19.02.2026



Niklas Paulicks