Asymptotic-Behavior–Driven Algorithm Design and Discrimination: From Scalable Computation to Convergence Metrics

A Dissertation Presented

by

Hongji Gao

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

December 2025

Stony Brook University

The Graduate School

Hongji Gao

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Xiangmin Jiao – Dissertation Advisor
Professor, Department of Applied Mathematics and Statistics

Robert Harrison – Chairperson of Defense
Professor, Institute for Advanced Computational Scinece

Benjamin Levine
Professor, Department of Chemistry

Name of Outside Member
Position
Institution

This dissertation is accepted by the Graduate School.

Eric Wertheimer
Dean of the Graduate School

Abstract of the Dissertation

Asymptotic-Behavior–Driven Algorithm Design and Discrimination: From Scalable Computation to Convergence Metrics

by

Hongji Gao

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2025

text of abstract

Contents

List of Figures

Acknowledgements

text of acknowledgement

Chapter 1

Introduction

Asymptotic behavior is a unifying lens through which a vast range of algorithms are conceived, analyzed, and compared. On the design side, asymptotics provides a principled target: reduce the leading-order time and memory costs by exploiting structural regularities of the data and operators rather than relying on constant-factor optimizations. On the assessment side, asymptotics supplies metrics that distinguish convergence rates and stability properties beyond implementation details, enabling scale-aware judgments about when and why one method should be preferred to another. This dissertation brings these two perspectives together in a single theme—algorithm design and algorithmic discrimination guided by asymptotic behavior—while deliberately spanning three largely independent contributions: two system-level, structure-exploiting algorithms in electronic structure theory, and a general convergence framework for multivariate iterative methods.

In large-scale scientific computing, achieving true scalability almost always hinges on uncovering and exploiting latent structure. Beyond classical sparsity, many operators and tensors in high-dimensional settings exhibit data sparsity: even though entries are not individually zero, far-field interactions admit low-rank surrogates. Hierarchical matrix techniques convert this observation into concrete data structures and algorithms. By recursively partitioning index sets and replacing admissible far-field blocks with controlled low-rank approximations, hierarchical matrices (H-/H^2-matrices) reduce asymptotic storage and matrix–vector multiplication from quadratic or worse to nearly linear or linearithmic, depending on rank growth and admissibility criteria. In H^2 variants, nested bases tie together multiple levels of the hierarchy, share subspace information across blocks, and further compress the representation in a way that directly benefits downstream solvers whose inner loops are dominated by operator applications.

These ideas are especially consequential in electronic structure theory, where the four-index electron–repulsion integral (ERI) tensor constitutes a central bottleneck. When reshaped into an $N^2 \times N^2$ matrix (with N the number of one-electron basis functions), the ERI exhibits strong locality among near-field orbital pairs and low-rank structure among well-separated groups. Prior work has shown that embedding hierarchical formats into Hartree–Fock (HF) and related self-consistent field (SCF) procedures can simultaneously compress ERI storage and accelerate the construction of Fock-like intermediates, thereby reducing both memory footprint and runtime at controlled accuracy. These developments confirm a general premise that is central to this dissertation: global algorithmic gains follow from local, quantifiable structure, provided that the data layout and computational kernels are co-designed to expose and reuse that structure across scales.

Building on this background, the first part of the dissertation adapts an H^2-matrix representation of the ERI tensor to the spin-opposite-scaled second-order Møller–Plesset (SOS-MP2) correlation framework. The design couples three elements: a hierarchical partition that separates near- and far-field interactions, nested-basis reuse to share low-dimensional subspaces across levels, and an atomic-orbital (AO) Laplace-style factorization that turns the dominant contractions into batches of matrix–vector and small matrix–matrix operations. This pipeline reduces the theoretical work in the dominant stages from the canonical $O(N^3)$ baseline toward $O(N^2 \log N)$ while achieving a corresponding decline in memory complexity due to the nested representation. Accuracy is managed through blockwise tolerances and rank caps, enabling smooth trade-offs between wall-clock time and correlation-energy error. In practice, data-locality-aware traversals, level-synchronized batching of far-field blocks, and parallel scheduling on the hierarchical frontier yield robust performance across molecular geometries and basis set sizes; the end result is a system-level method whose asymptotic profile makes problem sizes feasible that would be out of reach under unstructured dense algebra.

The second algorithmic contribution targets strongly correlated systems, where configuration-interaction (CI) methods remain a gold standard but are often memory-limited by the size of the coefficient tensor (or its matricizations). Empirically, such tensors frequently display a pronounced corner concentration under natural orderings: information density is highest near a principal corner and decays away from it. To capitalize on this anisotropy, the dissertation introduces a corner-hierarchical (CH) matrix format that aligns the recursive partition with the observed distribution of information. The associated CH-based approximated configuration interaction (CHACI) algorithm greedily refines the hierarchy according to blockwise information density and error tolerance, switching between dense storage and truncated-SVD representations as needed. By privileging the corner and its neighborhoods throughout the multilevel partition, CHACI captures more signal per degree of freedom than a single global SVD of the same unfolding. In representative active spaces and accuracy targets, this strategy yields about a 90% reduction in storage relative to a global SVD at comparable error levels, while preserving the algebraic interfaces needed by downstream CI kernels. Theoretically, storage bounds are provided in terms of block ranks and structural parameters of corner-dominant models, and practically, the method includes tolerance-setting and error-monitoring procedures that map directly onto chemical-accuracy requirements.

Complementing these two instances of asymptotic *design*, the third contribution focuses on asymptotic *discrimination*: how to compare and certify the convergence of iterative algorithms with sufficient sharpness and norm independence. Classical Q- and R-orders offer valuable but sometimes coarse lenses; gaps remain when one needs norm-agnostic statements, fine-grained distinctions within broad "linear" or "superlinear" classes, or principled handling under weak smoothness (e.g., Hölder continuity) and multistep couplings. To address these needs, the dissertation develops the P-order framework, which quantifies convergence by asymptotic comparison to power-like gauges and associated classes, thereby enabling a calibrated spectrum that includes fractional-power, linearithmic, and anti-linearithmic regimes. Two operational subfamilies—QUP-order and UP-order—are introduced to make the framework immediately usable in proofs and diagnostics. Within this setting, one obtains norm-independent, precision-aware comparisons among algorithms; sharp criteria for transitions between linear and superlinear behavior under minimal differentiability; and explicit rate characterizations for K-point methods under $C^{K-1,\nu}$ regularity. Conceptually, P-order provides a dimensionally consistent language for convergence that mirrors how complexity theory classifies work and storage, closing several long-standing gaps in rate analysis that previously relied on stronger smoothness or problem-specific norms.

Although the three parts of the dissertation address different problems and employ distinct techniques, they are joined by a common methodology: use asymptotics to guide both how we compute and how we judge computations. The hierarchical algorithms exemplify asymptotic design: identify structure, reorganize data and kernels to reduce leading-order costs, and prove the resulting bounds; the convergence framework exemplifies asymptotic discrimination: define scale-sensitive, norm-free gauges that separate algorithms by their rates under weak assumptions. Taken together, these contributions advance scalability, reliability, and analytic clarity in high-dimensional scientific computing. On the algorithmic side, they show that compressive representations—when matched to intrinsic structure such as far-field low rank or corner concentration—can move canonical workloads from cubic toward near-quadratic–logarithmic regimes while preserving accuracy through local error control. On the analytic side, they show that convergence can be expressed in a uniform, granular vocabulary that accommodates weak smoothness, multistep couplings, and nonstandard rate families without dependence on a particular norm.

The remainder of the dissertation develops each component in detail and connects the theoretical insights with reproducible implementations. For the hierarchical SOS-MP2 pipeline, we describe the H^2 data structures, admissibility conditions, rank-adaptive kernels, and AO/Laplace factorization that expose the right algebraic primitives, together with end-to-end complexity and accuracy analyses and empirical scaling on representative molecular series. For the corner-hierarchical compression of FCI tensors, we define the CH format, derive storage and error bounds, and present numerical studies that demonstrate substantial memory savings and stable accuracy under increasing active-space size. For the P-order framework, we formalize the definitions, establish norm independence and comparability properties, give sharp results under weak smoothness assumptions, and apply the framework to canonical single- and multi-point schemes to illustrate how the analysis refines classical classifications. The dissertation concludes by outlining directions in which these threads can interact—for example, adaptive strategies that couple hierarchical compression with P-order diagnostics to co-optimize accuracy and complexity—and by highlighting opportunities for broader application of asymptotic design and discrimination in other areas of scientific computing.

Chapter 2

Preliminaries and Related Work

2.1 Asymptotic Behavior and Convergence Analysis

Throughout this work we will study quantities that depend on a size or iteration parameter, such as the system size N, the number of basis functions, ranks of low-rank approximations, or the iteration index k of an iterative scheme. Their long-term behavior as $N \to \infty$ or $k \to \infty$ will be captured using a unified language of *asymptotic behavior*. In this section we first recall basic notions such as $\mathcal{O}(N^k)$ and related asymptotic notation, and explain how they will be used throughout the paper. We then view convergence orders of iterative methods—in particular the classical Q-order and the Ortega–Rheinboldt R-order—as specific applications of this asymptotic analysis to error sequences.

2.1.1 Asymptotic behavior of sequences and functions

Let $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$ be real sequences with $b_n > 0$ for all sufficiently large n. We say that

- $a_n = \mathcal{O}(b_n)$ as $n \to \infty$ if there exist constants $C > 0$ and $n_0 \in \mathbb{N}$ such that
$$|a_n| \leq C\, b_n \quad \forall\, n \geq n_0.$$
Intuitively, a_n grows no faster than b_n up to a constant factor.

- $a_n = o(b_n)$ as $n \to \infty$ if
$$\lim_{n\to\infty} \frac{|a_n|}{b_n} = 0,$$
i.e., a_n is asymptotically negligible compared to b_n.

- $a_n = \Theta(b_n)$ as $n \to \infty$ if there exist $c_1, c_2 > 0$ and $n_0 \in \mathbb{N}$ such that

$$c_1 b_n \leq |a_n| \leq c_2 b_n \quad \forall\, n \geq n_0.$$

  In this case a_n and b_n have the same growth rate up to constant factors.

Analogous definitions apply to functions $a(t)$ and $b(t)$ as $t \to \infty$ or $t \to 0^+$ by considering t as a continuous parameter. We will freely switch between sequence and function notation depending on the context.

A particularly important family of asymptotic behaviors in this work is *polynomial* and *exponential* scaling with respect to a size parameter $N \in \mathbb{N}$. We say that a quantity has

- polynomial or algebraic complexity of order k if it scales as $\mathcal{O}(N^k)$ for some $k > 0$;
- (sub)exponential complexity if it scales as $\exp(\mathcal{O}(N^k))$ or faster for some $k > 0$.

For example, the storage cost of a dense $N \times N$ matrix is $\Theta(N^2)$, while the cost of enumerating all determinants in a full configuration interaction (FCI) expansion grows combinatorially with N and is often described informally as "exponential" in the size of the active space.

In the context of iterative methods, we are interested in the asymptotic decay of an *error sequence* $\{\xi_k\}$ as $k \to \infty$. Typical behaviors include

- geometric or exponential decay, e.g. $\xi_k = \mathcal{O}(\rho^k)$ with $0 < \rho < 1$ ("linear" convergence);
- polynomial decay, e.g. $\xi_k = \mathcal{O}(k^{-\alpha})$ with $\alpha > 0$ (sublinear convergence);
- faster-than-linear decay, such as $\xi_k = \mathcal{O}(\rho^{k^2})$ or $\xi_k = \mathcal{O}(\rho^k / \log k)$ (various forms of superlinear convergence).

Throughout the paper we will use this asymptotic language at several levels:

- *Convergence analysis.* Convergence rates of iterative methods are expressed in terms of asymptotic decay rates of error sequences (Sections 2.1.2–2.1.3).
- *Complexity analysis.* Storage and arithmetic costs of low-rank and hierarchical matrix formats will be described using $\mathcal{O}(\cdot)$ and $\Theta(\cdot)$ notation in terms of system size, basis size, and ranks.

- *Approximation error.* Truncation errors in singular value decompositions (SVD/TSVD), strong rank-revealing QR (SRRQR), hierarchical matrices, and wavefunction/tensor compressions will be quantified using $\mathcal{O}(\cdot)$ notation in the relevant norms.

By fixing this asymptotic viewpoint at the outset, we can treat convergence rates, algorithmic complexity, and approximation errors within a unified framework of asymptotic behavior.

2.1.2 Q-order of convergence

We now recall how asymptotic behavior can be used to describe the convergence of iterative methods. Consider a generic fixed-point iteration

$$x_{k+1} = g(x_k), \qquad k = 0, 1, 2, \ldots, \tag{2.1}$$

aiming to compute a fixed point x^* satisfying $x^* = g(x^*)$, or equivalently a root of a nonlinear system $f(x^*) = 0$ after writing $g(x) = x - A(x)f(x)$. Let $e_k = x_k - x^*$ and $\xi_k = \|e_k\|$. The asymptotic behavior of the sequence $\{\xi_k\}$ as $k \to \infty$ characterizes the convergence of the method.

Following Traub [1], the sequence $\{x_k\}$ is said to converge to x^* with *Q-order* $r \geq 1$ if there exists a constant $C \in (0, \infty)$ such that

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^r} = C. \tag{2.2}$$

If $r = 1$ and $0 < C < 1$, the convergence is *Q-linear* with asymptotic rate factor C; if $r > 1$ the convergence is *Q-superlinear*, and if $r = 2$ it is *Q-quadratic*. In terms of asymptotic notation, Q-linear convergence corresponds roughly to $\|e_k\| = \Theta(\rho^k)$ for some $\rho \in (0, 1)$, whereas higher Q-orders describe faster regimes, such as $\Theta(\rho^{k^2})$.

Q-order is simple and has been widely used in the analysis of Newton-type methods and related algorithms; see, e.g., [2–4]. However, it has two well-known drawbacks [2, 5–7]. First, it is *norm dependent*: the rate factor C in (2.2) depends on the chosen norm, which complicates comparisons between different problems and different vector norms. Second, the definition implicitly assumes a form of asymptotic uniformity—roughly speaking, the local error reduction must behave like a fixed power of $\|e_k\|$ in the limit. As a result, Q-order can be ill-suited for non-monotone or highly non-uniform error sequences, which are common in modern iterative schemes with globalization strategies, adaptive steps, or stochastic perturbations.

These limitations motivate more flexible notions of convergence order that are better aligned with the asymptotic behavior viewpoint of Section 2.1.1.

2.1.3 R-order of convergence

To relax the asymptotic uniformity requirement and to support non-monotone sequences, Ortega and Rheinboldt introduced the *R-order* ("root order" of convergence) [3]. The idea is to compare the error to an auxiliary scalar sequence with prescribed Q-order. A sequence $\{x_k\}$ is said to converge to x^* with R-order r if there exists a nonnegative scalar sequence $\{\xi_k\}$ such that

$$\|x_k - x^*\| \le \xi_k, \qquad \lim_{k\to\infty} \frac{\xi_{k+1}}{\xi_k^r} = C_r \in (0, \infty), \tag{2.3}$$

for some $r \ge 1$. In this way, $\{\xi_k\}$ provides an upper bound for the error and has the desired Q-order-r behavior. This definition is naturally norm independent and can accommodate non-monotone error sequences.

Based on this construction, one can define *R-linear*, *R-superlinear*, and *R-quadratic* convergence, which are widely used in nonlinear equation solving and optimization [3, 4, 8, 9]. From the asymptotic viewpoint, R-order describes how fast the error can be bounded above by a scalar sequence with a given exponential-type decay.

However, R-order also has important limitations that have been discussed in detail in [5–7]. Because R-order is based on an upper bounding sequence, it often fails to distinguish between different sublinear rates: many qualitatively different behaviors (such as $\mathcal{O}(k^{-1})$ versus $\mathcal{O}(k^{-1/2})$) can share the same R-order. Moreover, the notion of "R-order-1" does not coincide with "R-linear" in general. For instance, sequences of the form

$$\|x_k - x^*\| = 0.5^k \, \log k \quad \text{or} \quad \|x_k - x^*\| = \frac{0.5^k}{\log k} \tag{2.4}$$

can both be classified as R-order-1, yet the former is superlinear and the latter is sublinear in the conventional sense. Similar ambiguities occur for $r > 1$ when one tries to distinguish R-order-r from truly rth-order convergence.

Consequently, while Q-order and R-order provide a useful first level of classification for the asymptotic behavior of error sequences, they do not always offer the fine-grained resolution needed to compare modern iterative methods, especially in the intermediate regimes between strictly linear and strictly quadratic convergence. This motivates the development of more refined convergence frameworks, which will be introduced and applied later in this work.

2.2 Linear Algebra Tools for Low-Rank and Hierarchical Approximation

Many of the algorithms considered in this work rely on low-rank and hierarchical approximations of large matrices and tensors. In this section we review the basic linear algebra tools underlying such approximations: singular value decomposition (SVD) and truncated SVD, strong rank-revealing QR (SRRQR) factorization, and hierarchical matrix formats including $\mathcal{H}$- and $\mathcal{H}^2$-matrices. Throughout, we emphasize their asymptotic storage and computational complexities in the sense of Section 2.1.1.

2.2.1 Singular value decomposition and truncated SVD

Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. The singular value decomposition (SVD) of A is

$$A = U\Sigma V^\top, \tag{2.5}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal with nonnegative entries

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,$$

called the singular values of A. The classical Eckart–Young theorem states that for any $1 \leq k \leq n$, the best rank-k approximation of A in either the spectral or Frobenius norm is obtained by truncating the SVD:

$$A_k := U_{(:,1:k)}\Sigma_{1:k,1:k}V_{(:,1:k)}^\top. \tag{2.6}$$

The approximation error satisfies

$$\|A - A_k\|_2 = \sigma_{k+1}, \qquad \|A - A_k\|_F^2 = \sum_{j>k}\sigma_j^2, \tag{2.7}$$

see, e.g., [10, 11]. Thus the decay of the singular values directly controls the asymptotic behavior of the truncation error as a function of k.

If A is dense with $m \geq n$, a full SVD costs $\mathcal{O}(mn^2)$ operations and requires $\Theta(mn)$ storage, which is prohibitive for very large problems. In many applications, however, the effective numerical rank of A is much smaller than $\min\{m, n\}$, and it suffices to compute a low-rank approximation of rank $k \ll \min\{m, n\}$. In this case one uses a *truncated SVD* (TSVD), computed by algorithms that target only the leading k singular triplets. Dense algorithms, Krylov subspace methods, and randomized algorithms can reduce the cost to roughly $\mathcal{O}(mnk)$, up to logarithmic factors, while maintaining near-optimal accuracy [12, 13].

From the point of view of asymptotic behavior, if the singular values of A decay polynomially, say $\sigma_j = \mathcal{O}(j^{-\alpha})$ for some $\alpha > 1/2$, then the TSVD error in the Frobenius norm satisfies

$$\|A - A_k\|_F = \Theta\Big(\big(\sum_{j>k} j^{-2\alpha}\big)^{1/2}\Big) = \mathcal{O}(k^{1/2-\alpha}),$$

while exponential decay of σ_j leads to exponentially small truncation errors. These relations will be used later to interpret low-rank tensor and matrix approximations in terms of their asymptotic error behavior.

2.2.2 Strong rank-revealing QR factorization

While SVD-based low-rank approximations are optimal in a least-squares sense, they can be relatively expensive and may not preserve sparsity. Rank-revealing QR (RRQR) factorizations provide a cheaper alternative that approximates the behavior of the SVD while relying only on orthogonal transformations and column permutations.

Given $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, a column-pivoted QR factorization has the form

$$A\Pi = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}, \tag{2.8}$$

where Π is a permutation matrix, Q is orthogonal, and R is upper triangular. A *strong rank-revealing QR* (SRRQR) factorization, as proposed by Gu and Eisenstat [14], imposes additional bounds on $\|R_{11}^{-1}R_{12}\|$ and $\|R_{22}\|$, ensuring that the leading columns selected by Π form a numerically stable basis for the dominant column space of A. For a target rank k, SRRQR yields an approximation of the form

$$A \approx A_{(:,J)}X, \tag{2.9}$$

where J indexes k selected columns and X is a well-conditioned interpolation matrix.

The cost of computing an SRRQR factorization of an $m \times n$ dense matrix is comparable to that of a standard pivoted QR factorization, $\mathcal{O}(mn^2)$ in the worst case, but with significantly improved rank-revealing properties. In practice, SRRQR is often applied to many moderate-size blocks (for example, in a hierarchical matrix structure), so that the overall complexity scales as a sum of local costs. From the asymptotic viewpoint, SRRQR provides low-rank approximations whose error is within a moderate factor of the optimal TSVD error, but at a lower total cost when applied blockwise.

2.2.3 Hierarchical matrices

Hierarchical matrices ($\mathcal{H}$-matrices) are data-sparse representations for large, structured matrices arising in applications such as elliptic partial differential equations, boundary integral equations, and kernel methods [15–17]. The main idea is to exploit the fact that matrix blocks corresponding to well-separated index clusters are often numerically low rank, while blocks corresponding to near-field interactions are not.

More precisely, one first organizes the index set $\mathcal{I} = \{1, \ldots, N\}$ into a cluster tree based on geometric or algebraic information, and then constructs a block cluster tree that partitions the $N \times N$ matrix into blocks. An admissibility condition—typically expressing that the underlying index clusters are well separated—is used to classify blocks as *far-field* or *near-field*. Near-field blocks are stored explicitly as dense matrices, while far-field blocks are compressed by low-rank approximations, often obtained by SVD, RRQR, or SRRQR.

Under mild assumptions on the underlying problem and the admissibility condition, $\mathcal{H}$-matrices achieve almost linear storage and arithmetic complexity. For example, for a second-order elliptic operator in three dimensions, one can typically store the corresponding stiffness matrix in $\mathcal{O}(N \log N)$ memory and apply it to a vector in $\mathcal{O}(N \log^2 N)$ time, where N is the number of degrees of freedom [17]. In the language of Section 2.1.1, this represents a substantial improvement over the $\Theta(N^2)$ storage and $\Theta(N^2)$ matrix-vector multiplication cost of a dense matrix.

The hierarchical organization is particularly flexible: blocks can be further subdivided, admissibility can be adapted to the problem at hand, and different low-rank approximation techniques can be used in different parts of the matrix. This flexibility makes $\mathcal{H}$-matrices a natural tool for compressing large matrices and tensors in many-body problems.

2.2.4 $\mathcal{H}^2$-matrices

While $\mathcal{H}$-matrices already reduce storage and computation costs significantly, they store separate low-rank bases for different far-field blocks, which leads to redundancy. $\mathcal{H}^2$-matrices refine this idea by introducing *nested* cluster bases that are shared across blocks [17, 18]. In an $\mathcal{H}^2$-matrix, each cluster in the index tree is associated with a basis that spans the relevant far-field interactions, and bases at coarse levels are represented in terms of bases at finer levels via small transfer matrices.

This nested structure eliminates much of the redundancy present in standard $\mathcal{H}$-matrices. Under suitable assumptions on the kernel or operator, $\mathcal{H}^2$-matrices can achieve linear or nearly linear complexity:

$$\mathcal{O}(N) \quad \text{for storage,} \qquad \mathcal{O}(N) \quad \text{for matrix-vector multiplication,} \tag{2.10}$$

again up to logarithmic factors and assuming bounded ranks. From the asymptotic viewpoint, this represents an improvement from $\mathcal{O}(N \log N)$ or $\mathcal{O}(N \log^2 N)$ to (essentially) $\mathcal{O}(N)$, while maintaining a prescribed accuracy in the relevant matrix norm.

$\mathcal{H}^2$-matrices are particularly effective for discretizations of translation-invariant or asymptotically smooth kernels, such as Green's functions of elliptic operators and Coulomb-like interactions. In later chapters we will make use of hierarchical and $\mathcal{H}^2$-matrix ideas to compress large matrices and tensors arising in electronic-structure theory, and we will analyze their storage, computational complexity, and approximation errors using the asymptotic framework introduced in Section 2.1.

2.3 Quantum Chemistry Background

We now review the quantum chemistry background relevant to this work, focusing on the tensor structures and electronic-structure methods that will be used in later sections. We first introduce the full configuration interaction (FCI) tensor and the electron repulsion integral (ERI) tensor, and then briefly review the Hartree–Fock (HF) method and second-order Møller–Plesset perturbation theory (MP2). Throughout, we emphasize the asymptotic behavior of the computational cost in terms of the number of basis functions and electrons.

2.3.1 Full configuration interaction tensor

Consider an N-electron system in a finite one-particle basis of M spin-orbitals $\{\chi_p\}_{p=1}^{M}$. Within this basis, the electronic wave function can be expanded in a basis of Slater determinants $\{\Phi_I\}$ constructed by occupying N spin-orbitals out of M:

$$\Psi = \sum_I C_I \Phi_I, \qquad \Phi_I = \frac{1}{\sqrt{N!}} \det\big[\chi_{i_1}(1)\, \chi_{i_2}(2)\, \ldots\, \chi_{i_N}(N)\big], \tag{2.11}$$

where $I = (i_1, \ldots, i_N)$ indexes the occupied spin-orbitals in determinant Φ_I and C_I are configuration interaction (CI) coefficients. In full configuration interaction (FCI), the sum runs over all determinants consistent with the specified number of electrons and spin symmetry, leading to a formally exact solution of the nonrelativistic electronic Schrödinger equation within the chosen basis [19, 20].

The dimension of the determinant basis is

$$N_{\text{det}} = \binom{M}{N_\alpha}\binom{M}{N_\beta}, \tag{2.12}$$

where N_α and N_β are the numbers of α- and β-spin electrons, respectively. As M grows, N_{det} increases combinatorially, which leads to an exponential scaling of the FCI cost with respect to system size. In the asymptotic language of Section 2.1.1, the storage required for the CI coefficients is $\Theta(N_{\text{det}})$, which behaves roughly like $\exp(\mathcal{O}(M))$ for fixed electron fraction.

It is often convenient to view the collection of CI coefficients as a tensor, the *FCI tensor*. For example, in an occupation-number representation the FCI tensor has one index per spin-orbital,

$$C_{n_1 n_2 \dots n_M}, \qquad n_p \in \{0, 1\}, \tag{2.13}$$

subject to the constraint $\sum_p n_p = N$. Alternatively, one can group spin-orbitals into α- and β-strings and fold the CI vector into a matrix with indices (I_α, I_β), where each index labels an occupation pattern of α- or β-spin orbitals. Regardless of the specific indexing, the resulting FCI tensor is extremely high-dimensional and dense, and serves as a prototypical example of an object with exponential storage requirements.

Because of this unfavorable asymptotic behavior, practical electronic-structure calculations rely on truncated CI expansions (CISD, CISDTQ, etc.), coupled-cluster theory, selected CI, density-matrix renormalization group (DMRG), tensor network states, and various low-rank or sparsity-exploiting wave function approximations; see, e.g., [19–23] for reviews. In later sections we will revisit the FCI tensor as a motivating example for hierarchical and low-rank tensor approximations.

2.3.2 Electron repulsion integral tensor

Let $\{\varphi_\mu\}_{\mu=1}^{N_{\text{bas}}}$ denote a set of spatial basis functions (typically Gaussian-type orbitals). The two-electron Coulomb interaction in this basis is encoded by the electron repulsion integral (ERI) tensor

$$(\mu\nu|\lambda\sigma) = \iint \varphi_\mu(\mathbf{r}_1)\varphi_\nu(\mathbf{r}_1)\frac{1}{\|\mathbf{r}_1 - \mathbf{r}_2\|}\varphi_\lambda(\mathbf{r}_2)\varphi_\sigma(\mathbf{r}_2)\, d\mathbf{r}_1\, d\mathbf{r}_2. \tag{2.14}$$

Here $\mu, \nu, \lambda, \sigma \in \{1, \ldots, N_{\text{bas}}\}$ index basis functions. The ERI tensor is a rank-4 object with $\Theta(N_{\text{bas}}^4)$ elements, and it possesses several permutation symmetries, such as

$$(\mu\nu|\lambda\sigma) = (\nu\mu|\lambda\sigma) = (\mu\nu|\sigma\lambda) = (\lambda\sigma|\mu\nu). \tag{2.15}$$

By grouping pairs of indices into compound indices, e.g. $(\mu\nu)$ and $(\lambda\sigma)$, one can reshape the ERI tensor into a $N_{\text{bas}}^2 \times N_{\text{bas}}^2$ matrix, which is convenient for numerical linear algebra operations and for applying low-rank and hierarchical approximations.

The ERI tensor plays a central role in Hartree–Fock, post-HF correlation methods, and density functional theory. Direct storage of all ERIs scales as $\Theta(N_{\text{bas}}^4)$, and the naive computation of all integrals has a similar or worse cost, depending on the integral algorithm [24, 25]. This motivates a wide range of integral screening and compression techniques, including density fitting (resolution of the identity) [26, 27], Cholesky decomposition of the ERI matrix [28], and tensor hypercontraction [29]. These methods seek to reduce both the storage cost and the asymptotic complexity of forming Coulomb and exchange contributions by exploiting approximate low-rank structure and the locality of basis functions.

2.3.3 Hartree–Fock method

The Hartree–Fock (HF) method provides a mean-field approximation to the electronic ground state by assuming that the N-electron wave function is a single Slater determinant built from N spin-orbitals. In its spin-restricted, closed-shell form, HF minimizes the electronic energy with respect to a set of occupied spatial orbitals $\{\phi_i\}$, subject to orthonormality constraints [19, 20]. This leads to the Roothaan–Hall equations

$$\mathbf{FC} = \mathbf{SC}\varepsilon, \tag{2.16}$$

where $\mathbf{F}$ is the Fock matrix, $\mathbf{S}$ is the overlap matrix, $\mathbf{C}$ is the coefficient matrix of molecular orbitals in the atomic orbital (AO) basis, and ε is the diagonal matrix of orbital energies.

In an AO basis $\{\varphi_\mu\}$, the Fock matrix elements are given by

$$F_{\mu\nu} = h_{\mu\nu} + \sum_{\lambda\sigma} P_{\lambda\sigma} \left[2(\mu\nu|\lambda\sigma) - (\mu\lambda|\nu\sigma)\right], \tag{2.17}$$

where $h_{\mu\nu}$ are one-electron integrals (kinetic energy and electron–nuclear attraction), $P_{\lambda\sigma}$ is the density matrix constructed from occupied orbitals, and

$(\mu\nu|\lambda\sigma)$ are electron repulsion integrals (ERIs). The HF procedure iterates between building the Fock matrix from a trial density matrix and solving the generalized eigenvalue problem until self-consistency is reached (self-consistent field, SCF).

The computational bottleneck of conventional HF lies in the construction of the Coulomb and exchange contributions to the Fock matrix. Without any screening or compression, the number of significant ERIs scales as $\Theta(N_{\text{bas}}^4)$, and each SCF iteration has a formal cost of $\mathcal{O}(N_{\text{bas}}^4)$ or higher, where N_{bas} is the number of basis functions [19, 20]. Various techniques have been developed to reduce this cost, including integral prescreening, density fitting (resolution of the identity) [26, 27], Cholesky decomposition of the ERI matrix [28], and local or linear-scaling HF methods that exploit the decay of the density matrix with distance and the nearsightedness of electrons in insulators [30, 31]. Many of these techniques are compatible with low-rank and hierarchical representations of the ERI tensor, which further reduce the asymptotic complexity of HF calculations.

Hierarchical block low-rank and $\mathcal{H}^2$-based ERI representations

A more recent line of work by Xing, Huang, and Chow combines the Hartree–Fock framework with hierarchical block low-rank and $\mathcal{H}^2$-matrix techniques to obtain near-linear-scaling algorithms for constructing the Coulomb and exchange matrices.

In their J. Chem. Phys. paper, Xing, Huang, and Chow introduced a *hierarchical block low-rank representation* of the ERI tensor [32]. By reshaping the ERI tensor into a matrix with compound indices $(\mu\nu)$ and $(\lambda\sigma)$, they treat it as a kernel matrix associated with the Coulomb interaction between products of AO basis functions. The AO indices are organized into a hierarchical cluster tree, and a block-cluster tree is used to partition the ERI matrix into near-field and far-field blocks. Near-field blocks are stored explicitly, while far-field blocks are approximated in low-rank form using hierarchical block low-rank techniques closely related to $\mathcal{H}^2$-matrices. This leads to a data-sparse representation of the ERI tensor in which both storage and matrix-vector multiplications scale linearly with the matrix dimension (up to logarithmic factors), rather than quadratically as in the dense case [32].

To efficiently construct such $\mathcal{H}^2$-type representations, Huang, Xing, and Chow developed the H2Pack library for kernel matrices [33]. H2Pack uses a hybrid analytic–algebraic compression strategy based on the proxy point method to build $\mathcal{H}^2$-matrix representations with linear complexity in the number of points. Storage and matrix-vector multiplication costs are both $\mathcal{O}(N)$, where N is the matrix dimension, under standard assumptions on the kernel and the geometry of the points [33]. Although H2Pack is a general-purpose kernel matrix package, its underlying ideas can be directly applied to the ERI matrix viewed as a Coulomb kernel matrix over AO products.

Within the HF context, the hierarchical block low-rank ERI representation enables the Coulomb and exchange contributions to the Fock matrix to be constructed using only matrix-vector and small dense-matrix operations with the compressed ERI representation. As a result, the asymptotic cost of building the Fock matrix can be reduced from $\Theta(N_{\text{bas}}^4)$ to nearly linear in N_{bas} for large three-dimensional systems, while controlling the approximation error through the ranks and tolerance parameters in the hierarchical compression [32]. From the perspective of this work, these results illustrate how hierarchical and $\mathcal{H}^2$-based low-rank representations of the ERI tensor can substantially improve the asymptotic behavior of mean-field electronic-structure calculations, and they provide an important point of comparison for the tensor and hierarchical approaches developed later in this paper.

2.3.4 Second-order Møller–Plesset perturbation theory (MP2)

Second-order Møller–Plesset perturbation theory (MP2) is one of the simplest and most widely used post-HF correlation methods. Starting from a converged HF reference, MP2 treats the residual electron correlation as a second-order perturbation [34]. In the canonical molecular orbital (MO) basis, the MP2 correlation energy can be written as

$$E_{\text{MP2}} = \sum_{ijab} \frac{(ij||ab)^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}, \tag{2.18}$$

where i, j index occupied MOs, a, b index virtual MOs, ε_p are orbital energies, and $(ij||ab)$ are antisymmetrized two-electron integrals in the MO basis:

$$(ij||ab) = (ij|ab) - (ij|ba). \tag{2.19}$$

A straightforward evaluation of E_{MP2} scales as $\mathcal{O}(N_{\text{occ}}^2 N_{\text{vir}}^2 N_{\text{bas}})$ in floating point operations and requires $\Theta(N_{\text{occ}} N_{\text{vir}} N_{\text{bas}}^2)$ storage if all relevant ERIs are precomputed, where N_{occ} and N_{vir} are the numbers of occupied and virtual MOs, respectively, and N_{bas} is the number of basis functions. In terms of a single size parameter N representing system or basis size, the overall scaling is typically described as $\mathcal{O}(N^5)$ in time and $\mathcal{O}(N^4)$ in memory [19, 20].

Because of this unfavorable asymptotic behavior, a large literature has focused on reducing the scaling of MP2. Existing approaches include local correlation methods, which exploit the spatial locality of occupied orbitals and electron pairs [35, 36], density fitting and Cholesky decomposition of the ERI tensor [28, 37, 38], Laplace-transformed MP2 [37, 39], and tensor-factorization techniques such as resolution-of-the-identity MP2 (RI-MP2) and tensor hypercontraction [29]. Many of these methods can achieve effective scaling closer to $\mathcal{O}(N^4)$ or even lower for large, insulating systems, while maintaining chemical accuracy.

In this work, MP2 primarily serves as a representative example of a correlated wave function method whose computational cost is dominated by operations involving the ERI tensor. The asymptotic behavior of the MP2 energy evaluation is therefore closely tied to the structure and compression of the ERI tensor introduced in Section 2.3.2.

Chapter 3

Corner Hierarchically Approximated Configuration Interaction

In this chapter we summarize the main ideas and results of the corner hierarchically approximated configuration interaction (CHACI) approach for wave function compression, based on Berard *et al.* [40]. CHACI combines the tensor structure of full configuration interaction (FCI) wave functions with a corner hierarchical (CH) matrix format to obtain efficient and scalable compression of strongly correlated wave functions.

Building on the preliminaries in Chapters 2.1–2.3, we organize this chapter into three parts:

1. Why corner hierarchical matrices are well suited for FCI/CI tensors.

2. The CHACI algorithm: CH blocking, blockwise TSVD, and information-density-based rank selection.

3. Numerical performance: model systems, error-versus-storage behavior, and ablation studies.

3.1 Corner Hierarchical Matrices for FCI and CI Tensors

3.1.1 From FCI tensor to CI matrix and corner dominance

Consider a CASCI or FCI wave function written in an active space of M_α α-strings and M_β β-strings, as in Section 2.3.1. We can write

$$\Psi = \sum_{I_\alpha=1}^{M_\alpha} \sum_{I_\beta=1}^{M_\beta} C_{I_\alpha I_\beta} \, \Phi_{I_\alpha I_\beta}, \tag{3.1}$$

where $C_{I_\alpha I_\beta}$ are CI coefficients and $\Phi_{I_\alpha I_\beta}$ are determinants labeled by α- and β-string indices. Arranging $C_{I_\alpha I_\beta}$ into a matrix

$$\mathbf{C} = \left(C_{I_\alpha I_\beta}\right) \in \mathbb{R}^{M_\alpha \times M_\beta}, \tag{3.2}$$

we obtain the *CI matrix* representation of the FCI tensor.

For strongly correlated systems such as acenes, numerical evidence shows that the largest CI coefficients can be concentrated into a relatively small sub-block of **C** after a suitable reordering of rows and columns (e.g., sorting by row/column norms or natural-orbital occupation). We observe that after sorting, the important coefficients cluster near the upper-left corner of **C**, while the rest of the matrix becomes increasingly sparse or low-magnitude as one moves away from this corner [40]. This behavior is referred to as *corner dominance*.

A schematic illustration is given in Figure 3.1. In the unsorted CI matrix, significant coefficients are scattered. After sorting, the density of large entries is strongly localized in the upper-left corner, suggesting a natural focus region for refinement and a surrounding region that can be approximated more aggressively.

3.1.2 Why standard H-matrices are not ideal

As reviewed in Section 2.2.3, traditional hierarchical ($\mathcal{H}$) matrices are tailored for *diagonally dominant* matrices, in which strong interactions are located near the main diagonal and far-field blocks away from the diagonal are approximately low rank. The block structure for a typical H-matrix is sketched in Figure 3.2: near-diagonal blocks (gray) are kept dense, while far-field blocks (white) are represented in low-rank form. For CI matrices with corner domi-

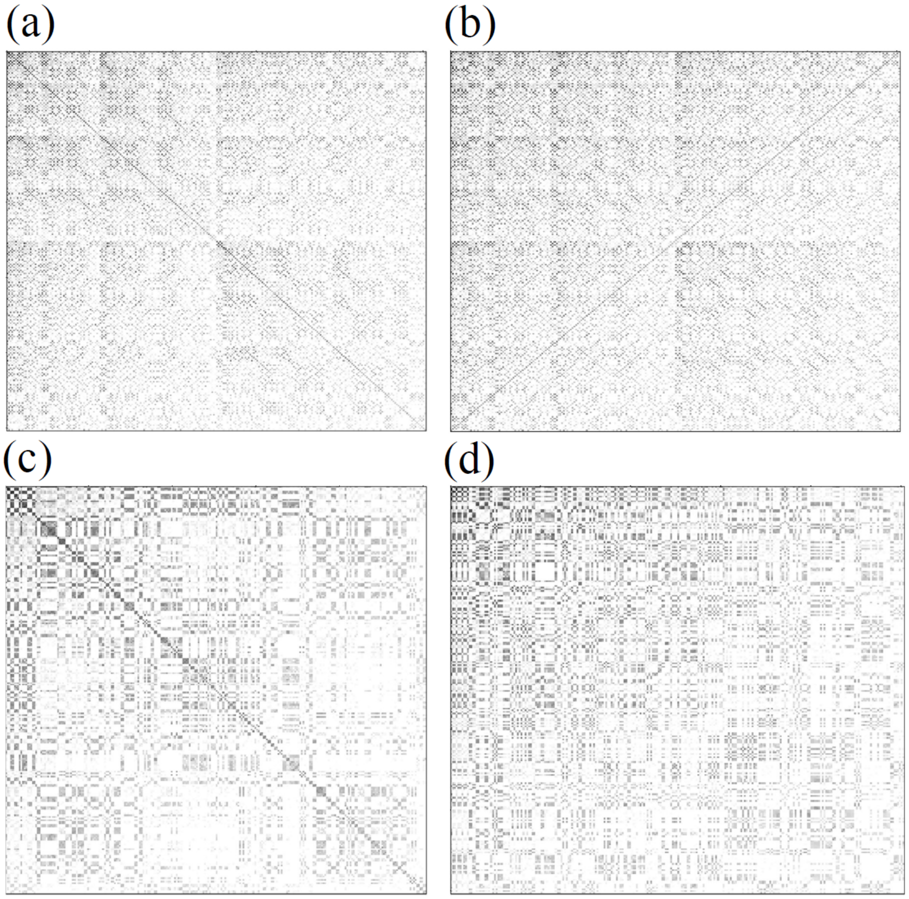

Figure 3.1: A heat map representation of the CI vector of 12-acene, computed with a 10-10 active space. To form the heat map, we take the logarithm (base 10) of the absolute value of the singlet and triplet CI vector coefficients. The color scale (white to black) ranges from 10^{-6} to 1. Panels (a) and (b) correspond to the unsorted CI vector (with strings indexed according to Duch[**?**]), while panels (c) and (d) are reordered according to the CHACI algorithm. Panels (a) and (c) show the singlet wave function, while panels (b) and (d) show the triplet wave function.

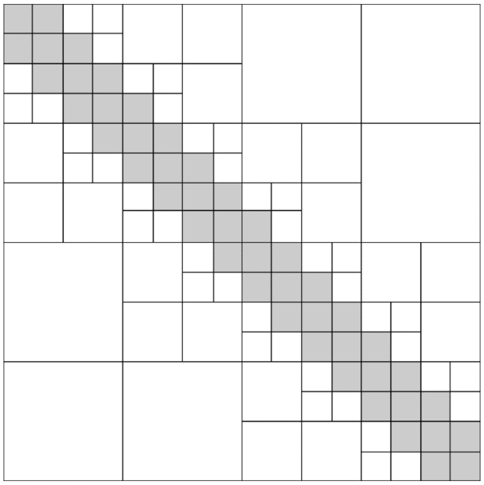

Figure 3.2: Block representation of an H-matrix. White blocks are stored as low-rank approximations, while gray blocks are stored as dense matrices.

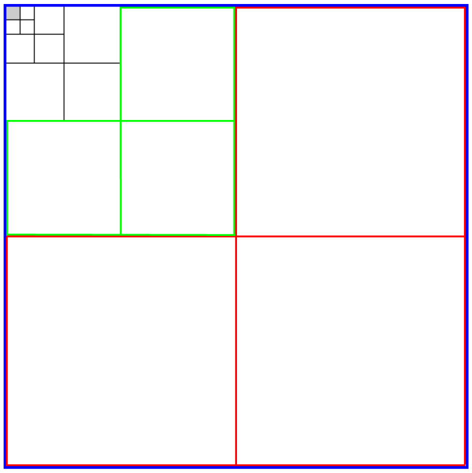

Figure 3.3: Schematic of the corner-hierarchical matrix structure. The upper-left corner is compressed hierarchically and the other blocks are compressed adaptively.

nance in Figure 3.3, however, the strongest coefficients are not aligned along a diagonal. Instead, they cluster in an upper-left corner whose boundary does not generally coincide with a diagonal band. Applying a standard H-matrix blocking to such a matrix leads to:

- many near-diagonal blocks that contain mostly small coefficients and are therefore inefficiently stored as dense blocks;
- off-diagonal blocks that mix important and unimportant coefficients, making it difficult to find low-rank approximations that capture the relevant information without increasing rank.

Numerical experiments in Ref. [40] confirm that H-matrix compression of CI matrices is significantly less efficient than the corner hierarchical approach described below.

3.1.3 Corner hierarchical (CH) matrix format

To better exploit corner dominance, we introduce the *corner hierarchical* (CH) matrix format [40]. The key idea is to refine the upper-left corner of the matrix hierarchically, rather than a diagonal band. A CH matrix is constructed as follows:

1. Start from the full $M_\alpha \times M_\beta$ CI matrix $\mathbf{C}$ (after sorting).

2. Recursively split the current block into four subblocks: upper-left, upper-right, lower-left, and lower-right.

3. Refine (subdivide) only the *upper-left* subblock at each level. The other three blocks are treated as candidate blocks for dense or low-rank approximation.

4. Repeat this for p levels. At the end, there is one fully refined upper-left block and 3^p other blocks of varying sizes.

This CH structure is tailored to the observed pattern of CI coefficients:

- The upper-left corner contains the most important determinants and is kept in relatively high resolution (smaller blocks, possibly dense).

- As we move away from the corner, blocks contain progressively less important coefficients and can be stored in compressed form (blockwise truncated SVD) or dropped entirely.

Because the CH blocking follows the physical structure of the CI matrix, it allows CHACI to achieve much higher compression ratios than global TSVD or H-matrix compression at the same accuracy.

3.2 The CHACI Algorithm

3.2.1 Overall workflow

Given a CASCI wave function, the CHACI workflow is:

1. **Build and sort CI matrix.** Construct the CI matrix $\mathbf{C}$ from the CI vector in the (I_α, I_β) representation and sort its rows and columns by some measure of importance (e.g., row/column norms). This step enhances corner dominance.

2. **CH blocking.** Partition $\mathbf{C}$ into $3^p + 1$ blocks using the CH blocking scheme described in Section 3.1.3, for a chosen number of levels p.

3. **Blockwise compression.** For each block, decide among three options: (i) drop the block, (ii) store it in dense form, or (iii) store a truncated SVD (TSVD) approximation with an adaptively chosen rank.

4. **Assemble CHACI wave function.** The compressed wave function is represented as the collection of dense blocks and TSVD factors $(U_i, \Sigma_i, V_i^\dagger)$ across the CH structure.

The central algorithmic ingredients are the TSVD of individual blocks and the information-density-based rank selection, described below.

3.2.2 TSVD for individual blocks

Let $B \in \mathbb{C}^{m\times n}$ be a block from the CH partition of $\mathbf{C}$. Its singular value decomposition (SVD) is

$$B = U\Sigma V^\dagger, \tag{3.3}$$

where U and V are unitary matrices and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min\{m,n\}})$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. For a rank-$k$ approximation, we retain only the largest k singular values and corresponding singular vectors:

$$B \approx \widetilde{B}' = U_T\Sigma_T V_T^\dagger, \tag{3.4}$$

with $U_T \in \mathbb{C}^{m\times k}$, $V_T \in \mathbb{C}^{n\times k}$ and $\Sigma_T = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$.

The Frobenius-norm error of this truncation is

$$\|B - \widetilde{B}'\|_F^2 = \sum_{i>k} \sigma_i^2, \tag{3.5}$$

so the truncation error is entirely controlled by the neglected singular values. Following Ref. [40], CHACI introduces an additional scalar factor

$$a = \frac{\|B\|_F}{\|\widetilde{B}'\|_F}, \tag{3.6}$$

and stores a together with $(U_T, \Sigma_T, V_T^\dagger)$ so that the block reconstruction is $B \approx a\widetilde{B}'$. This prevents global renormalization from shifting probability density between blocks.

3.2.3 Information density and rank selection

To decide how many singular values to retain in each block, CHACI introduces an *information density* measure for each singular vector pair. Let n_{row} and n_{col} denote the number of rows and columns of the block. The information density of the ith singular pair is defined as

$$\rho_i = \frac{\sigma_i^2}{n_{\text{row}} + n_{\text{col}} + 1}. \tag{3.7}$$

The numerator measures the contribution of the ith singular pair to the squared Frobenius norm of B, while the denominator approximates its storage cost (two vectors plus one scalar).

Given a user-specified threshold ρ, CHACI keeps only those singular pairs for which $\rho_i > \rho$:

$$\rho_i > \rho \quad \Longrightarrow \quad \text{keep the } i\text{th singular pair.} \tag{3.8}$$

The number k of singular values retained for a particular block is thus chosen adaptively based on the decay of σ_i. Blocks with rapidly decaying singular values will have small k; blocks with slow decay may be either stored densely or with larger k.

Blocks whose all $\rho_i \leq \rho$ are dropped entirely (not stored). This adaptive, blockwise strategy approximately maximizes the overall captured Frobenius norm of the CI matrix under a global storage budget.

3.2.4 Near-optimal CHACI algorithm

The near-optimal CHACI algorithm can be summarized in the pseudocode in Algorithm 1, adapted from Algorithm 1 in Ref. [40].

Algorithm 1 Near-Optimal CHACI

```
Input: The sorted CI vector, number of partitioning levels $p$, and information
    density threshold $\rho$
Output: The CHACI format and storage information for each block
 1: Split the CI vector into a corner hierarchical structure with $3p + 1$ blocks
 2: Store the upper-left block in dense format
 3: for each of the remaining $3p$ blocks do
 4:     Let $n_{\text{row}}$ and $n_{\text{col}}$ be the numbers of rows and columns of the block
 5:     $k \leftarrow 0$, $k_{\max} \leftarrow 1$, and perform rank-1 TSVD of the block
 6:     while $k < \min\{n_{\text{row}}, n_{\text{col}}\}/2$ do
 7:         if $k + 1 > k_{\max}$ then
 8:             $k_{\max} \leftarrow \min\{2k_{\max}, n_{\text{row}}/2, n_{\text{col}}/2\}$
 9:             Update the TSVD of this block to rank $k_{\max}$
10:         end if
11:         if $\sigma_{k+1}^2 \leq \rho(n_{\text{row}} + n_{\text{col}} + 1)$ then
12:             break
13:         end if
14:         $k \leftarrow k + 1$
15:     end while
16:     if $k = 0$ then
17:         Do not store the block
18:     else if $k = \min\{n_{\text{row}}, n_{\text{col}}\}/2$ then
19:         Store the block in dense format
20:     else
21:         Store the block in TSVD format with rank $k$
22:     end if
23: end for
```

The doubling strategy for $k_{\max}$ (Lines 7–9) ensures that the cost of updating TSVDs grows only logarithmically with the final rank per block. In practice, TSVDs are computed by iterative methods that exploit the moderate size of individual blocks.

3.2.5 CHACI variants and baselines

To disentangle the contributions of different ingredients in CHACI, we compare several variants and baselines. These are summarized in Table 3.1, adapted from Table I in Ref. [40].

- **CHACI** is the full method described above.

Table 3.1: Wave function compression schemes compared in Ref. [40]. "Blocking" describes the block structure; "Sorting" indicates whether rows/columns of **C** are sorted prior to compression; "Optimal rank" indicates whether ranks are adaptively chosen using information density.

Scheme	Blocking	Sorting?	Optimal rank?
CHACI	Corner hierarchical	Yes	Yes
SR-CHACI	Corner hierarchical	Yes	No
U-CHACI	Corner hierarchical	No	Yes
H-matrix	Diagonally dominant	No	No
Truncated SVD	None	No	No

- **SR-CHACI** uses CH blocking and sorting, but with a static rank per block rather than information-density-based adaptive ranks.

- **U-CHACI** uses CH blocking and adaptive ranks but does not sort **C**.

- **H-matrix** uses a standard diagonally dominant hierarchical blocking without sorting or adaptive ranks.

- **TSVD** applies a single global truncated SVD to **C**.

Comparisons among these schemes in the next section show that all three components—sorting, corner blocking, and adaptive rank selection—are crucial for the best performance.

3.3 Numerical Performance

3.3.1 Accuracy of CHACI Compression Versus TSVD

We start by comparing the performance of CHACI compression to TSVD compression for the 14-14 active space. Figure 3.4 presents the error in the singlet-triplet gap as a function of total storage. Clearly, CHACI outperforms TSVD in this case. The CHACI error is always less than 0.07 eV, even when only 28 kdoubles are stored (compared to 11,778 kdoubles for dense storage). Truncated SVD also achieves substantial compression, but errors of 0.2 eV remain even with 220 kdoubles stored.

Figures 3.5a and b present errors in the singlet and triplet absolute energies, respectively. Again, the performance of CHACI is far superior to TSVD, especially in the small-storage regime, where the TSVD errors are a substantial fraction of an eV. Figure 3.5c and d present spin error. Here CHACI again

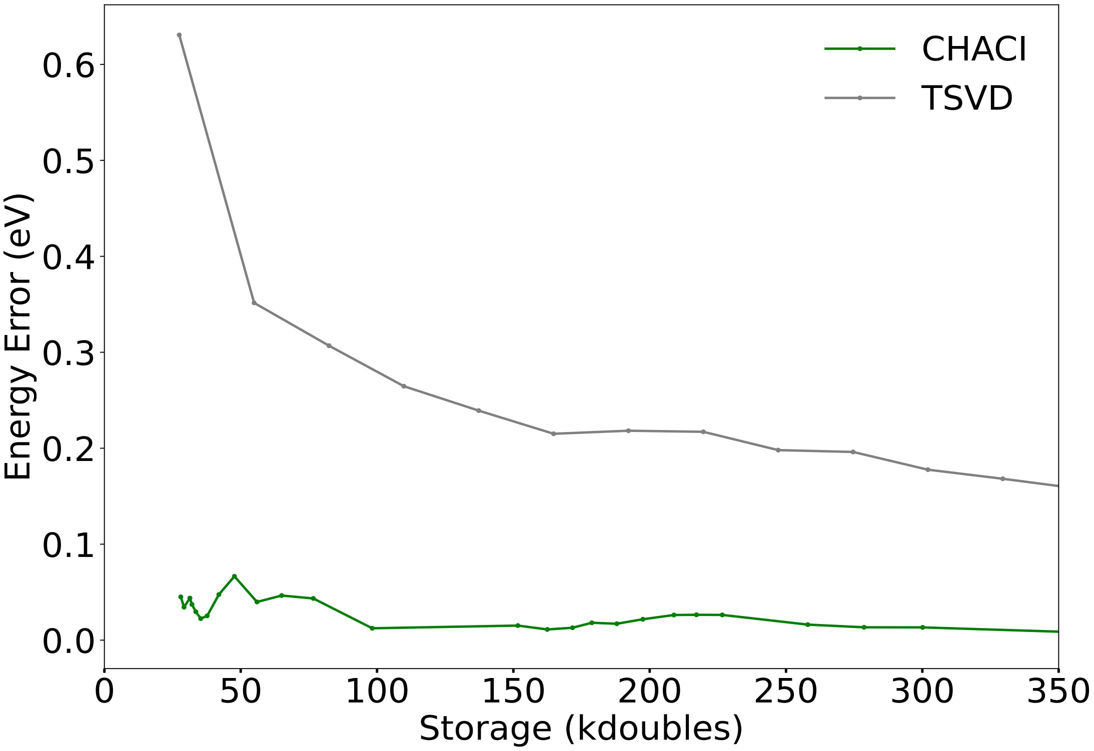

Figure 3.4: The error in the singlet-triplet gap of 12-acene as a function of total storage, computed with a 14-14 active space. The gray and green lines represent the error corresponding to the compression of the wave function using TSVD and CHACI, respectively.

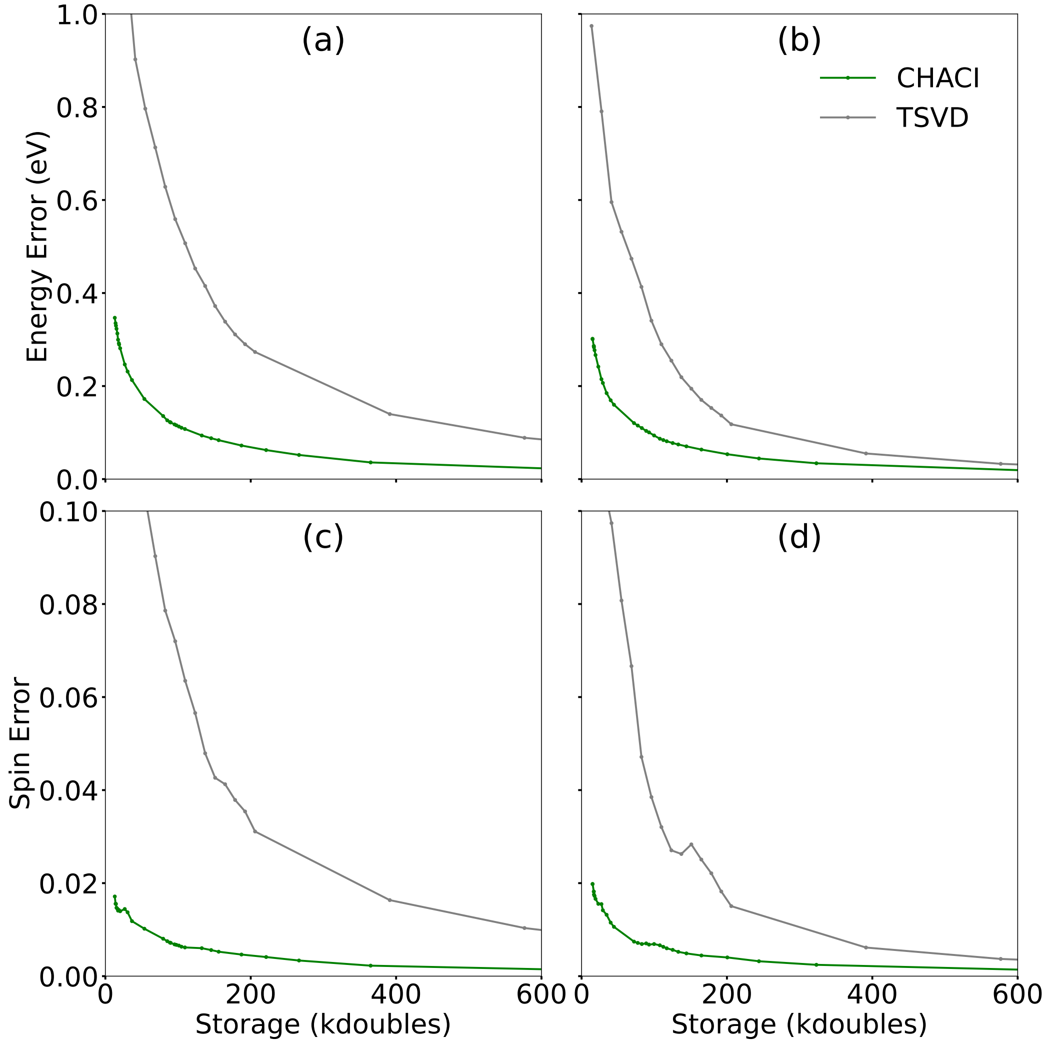

Figure 3.5: Absolute energy (a and b) and spin (c and d) errors as a function of the storage for 12-acene with a 14-14 active space. Panel (a) and (c) correspond to the singlet wave function, while (b) and (d) correspond to the triplet wave function. The gray and green lines correspond to TSVD and CHACI compression, respectively.

outperforms TSVD. Even with very modest (28 kdoubles) storage, the error in $\langle S^2 \rangle$ is 0.02, and the error drops rapidly toward zero with additional storage. In contrast, TSVD spin errors for the singlet case do not drop below 0.02 until nearly 400 kdoubles are stored.

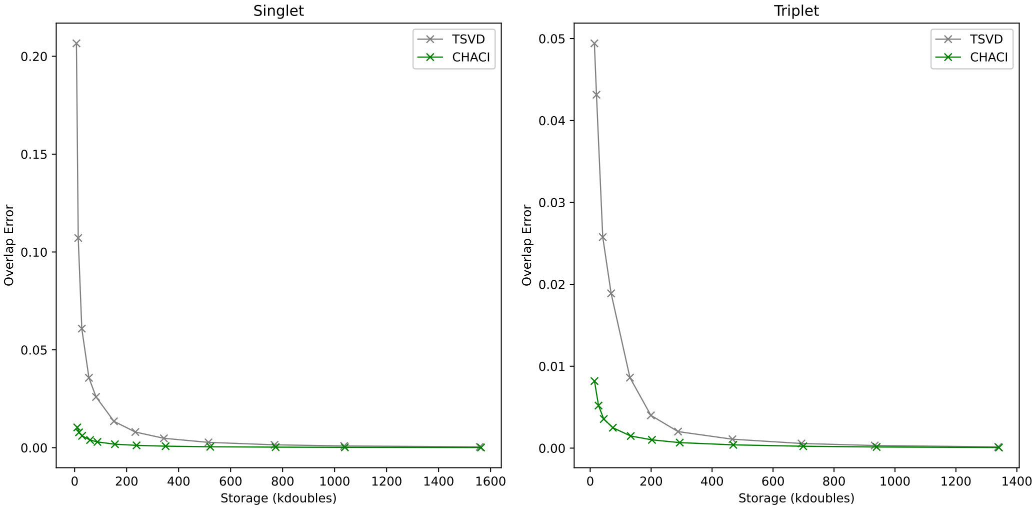

Figure 3.6: The overlap error, as defined in the text, for the 14-14 singlet (left) and triplet (right) wave functions of 12-acene as a function of storage. CHACI and TSVD results are shown in green and gray, respectively.

As an additional metric of the accuracy of the wave function, we computed what we call the overlap error, defined as $1 - \mathrm{Tr}(\mathbf{C}^T\tilde{\mathbf{C}})$. In physical terms, it is the deviation of the overlap of the compressed and exact wave functions from one. Overlap errors are reported as a function of storage in Figure 3.6. Clearly CHACI compression provides a much more accurate reproduction of the wave function at a given storage cost. Even at the lowest storage values tested, overlap with the exact wave function is ∼0.99 for both the singlet and triplet.

To investigate whether the convergence of states defined in different configuration spaces is balanced, we compare the convergence behavior of two rigorously degenerate states with very different CI vectors: the $m_s = 0$ and $m_s = 1$ components of the triplet (Fig. 3.7). The CHACI approach outperforms TSVD in all cases. However, the $m_s = 0$ wave function converges considerably more quickly with increasing storage than the $m_s = 1$ wave function in both cases. In addition, the discrepancy for CHACI is very large at some points, with the error in the absolute energy of the $m_s = 1$ component being several times larger than that of the $m_s = 0$ component. This larger error is observed despite the fact that the $m_s = 1$ CI vector is actually 24%

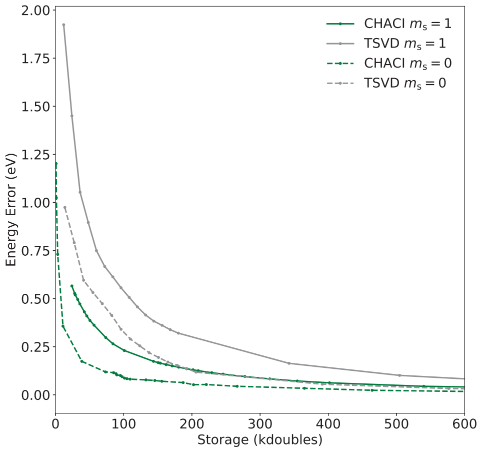

Figure 3.7: The error in the absolute energies of two m_s components of the triplet of 12-acene, computed with a 14-14 active space. Green and gray lines correspond to CHACI and TSVD, respectively. Dashed and solid lines correspond to $m_s = 0$ and $m_s = 1$.

shorter than the $m_s = 0$ vector. It seems that achieving a balanced treatment of states defined in different configuration spaces is challenging.

Figures 3.8 and 3.9 demonstrate that the difference in performance between CHACI and TSVD increases when the active space size is increased from 14-14 to 16-16. In Figure 3.8, it can be seen that errors in the singlet-triplet gap are at or below 0.1 eV for all cases, when CHACI is employed, even when only 59 kdoubles are stored (compared to 165,637 kdoubles for dense storage). Errors decrease rapidly with additional storage. In contrast, TSVD errors are above 0.2 eV for all cases. Similarly large differences in performance are observed for errors in absolute energy and spin in Figure 3.9. As in the 14-14 case, CHACI does a much better job of maintaining the spin symmetry of the wave function than TSVD.

3.3.2 Extrapolation of Performance to Larger Active Spaces

Ultimately, our goal is not to compute dense wave functions for subsequent compression. Our goal is to solve for large CI wave functions using a hierarchically compressed basis. Thus, in this section, we consider the behavior of CHACI compression as a function of active space size. In Figure 3.10, we consider several active spaces, comparing the convergence of several measures

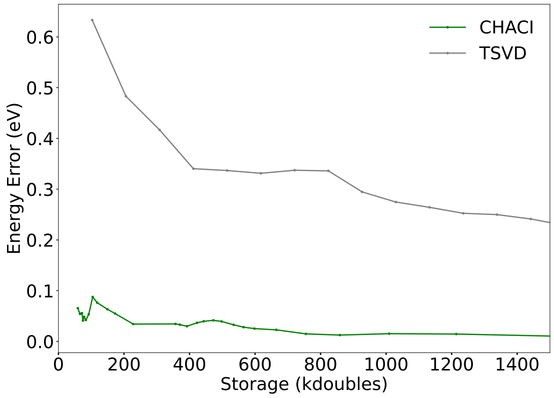

Figure 3.8: The error in the singlet-triplet gap of 12-acene as a function of total storage, computed with a 16-16 active space. The gray and green lines represent the error incurred by compression of the wave function using TSVD and CHACI, respectively.

of the accuracy as a function of the percentage of dense storage used (the compression ratio). We find that as the size of the active space increases, the accuracies of both absolute energy and spin converge faster with increasing compression ratio.

To quantify this convergence behavior, we plot the compression ratio at which absolute energies of 0.2 eV accuracy are achieved as a function of the number of active orbitals/electrons in Figure 3.11. Both the singlet and triplet compression ratios converge quickly with increasing active space. Of the two, the triplet energy converges more slowly, thus we fit it to an exponential in order to extrapolate to larger active spaces. We find that the required compression ratio decays proportional to

$$f(N_{\mathrm{MO}}) \propto e^{-0.561 N_{\mathrm{MO}}}. \tag{3.9}$$

Extrapolating to larger active spaces, we estimate that a 24-24 active space could converge to 0.2 eV accuracy at a storage cost of 77,370 kdoubles, which is less than the cost of the dense storage of a 16-16 active space (165,637 kdoubles). Though the convergence behavior is likely to be system-dependent, this result certainly encourages further study.

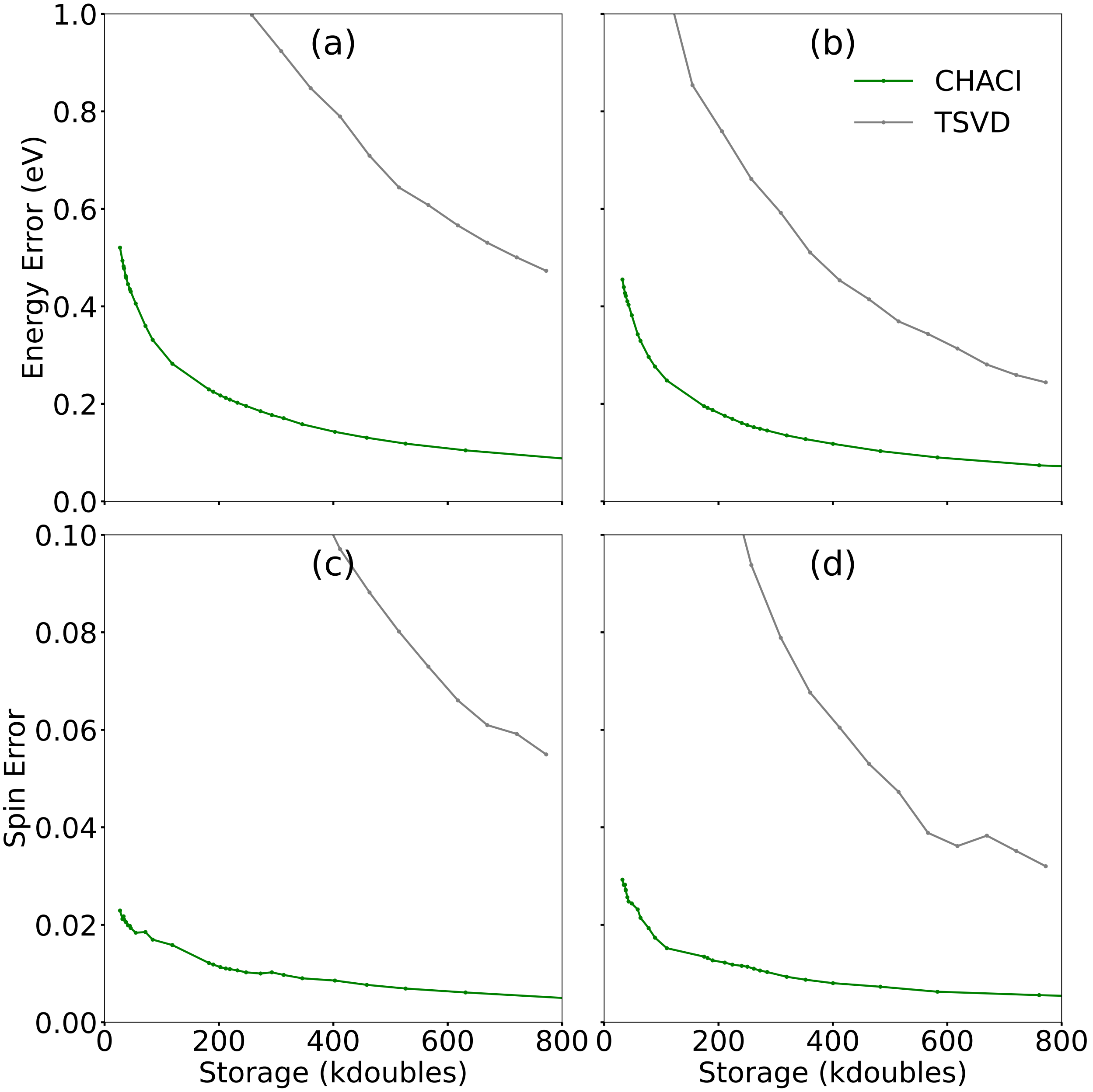

Figure 3.9: Absolute energy (a and b) and spin (c and d) errors as a function of the storage for 12-acene with a 16-16 active space. Panel (a) and (c) correspond to the singlet wave function, while (b) and (d) correspond to the triplet wave function. The gray and green lines correspond to TSVD and CHACI compression, respectively.

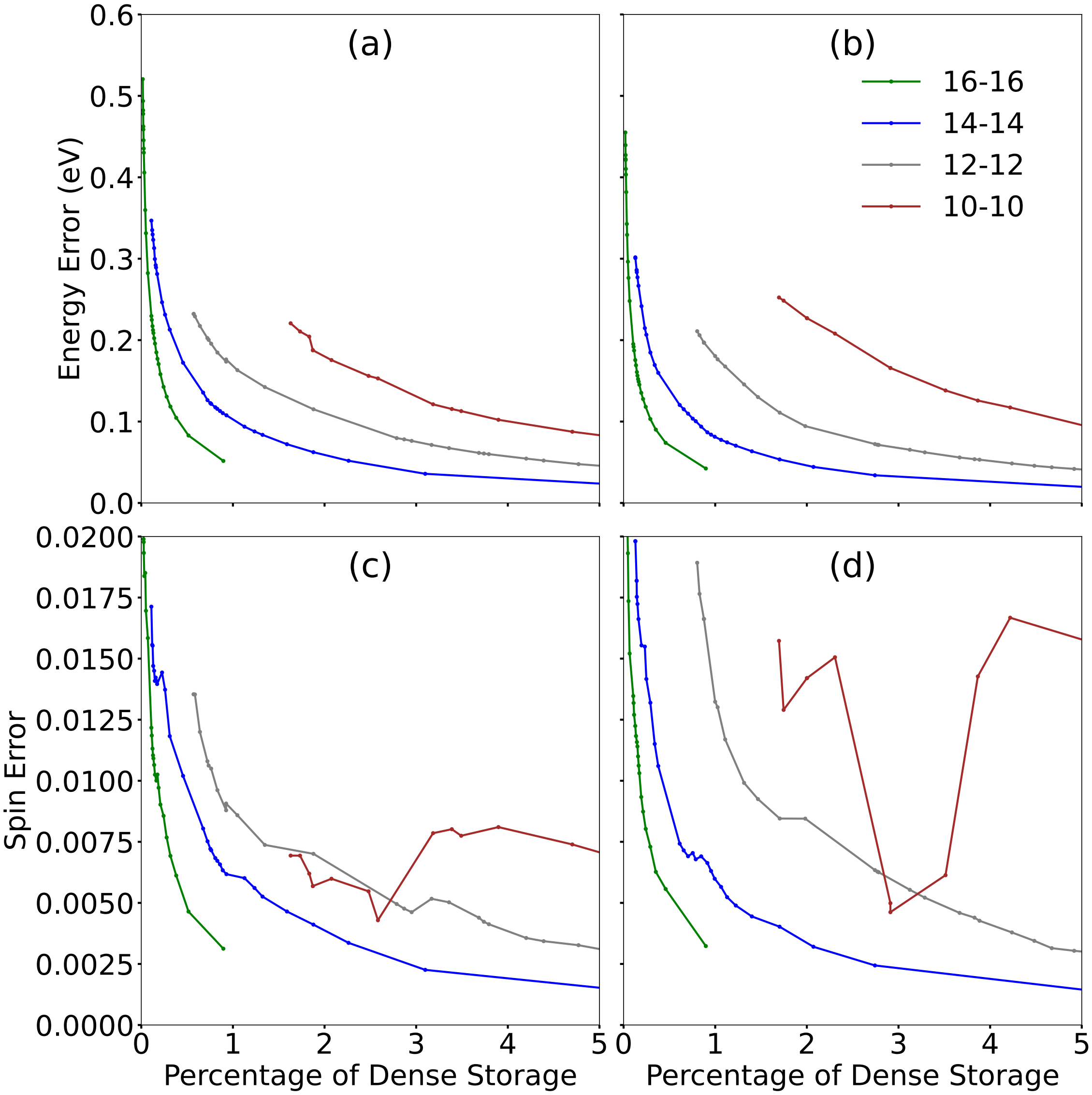

Figure 3.10: The absolute energy (a and b) and spin errors (c and d) as a function of the percentage of dense storage for 12-acene with 10-10, 12-12, 14-14, and 16-16 active spaces. Panels (a) and (c) correspond to the singlet wave function, while panels (b) and (d) correspond to the triplet spin wave function.

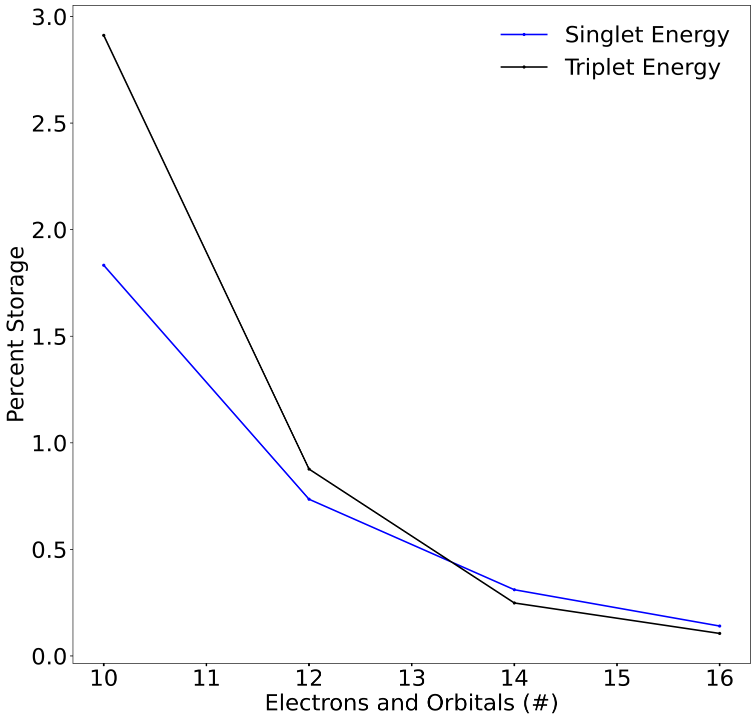

Figure 3.11: The storage ratio required to achieve < 0.2 eV accuracy in absolute energies as a function of the active space size. Blue and black lines correspond to the singlet and triplet wave functions, respectively.

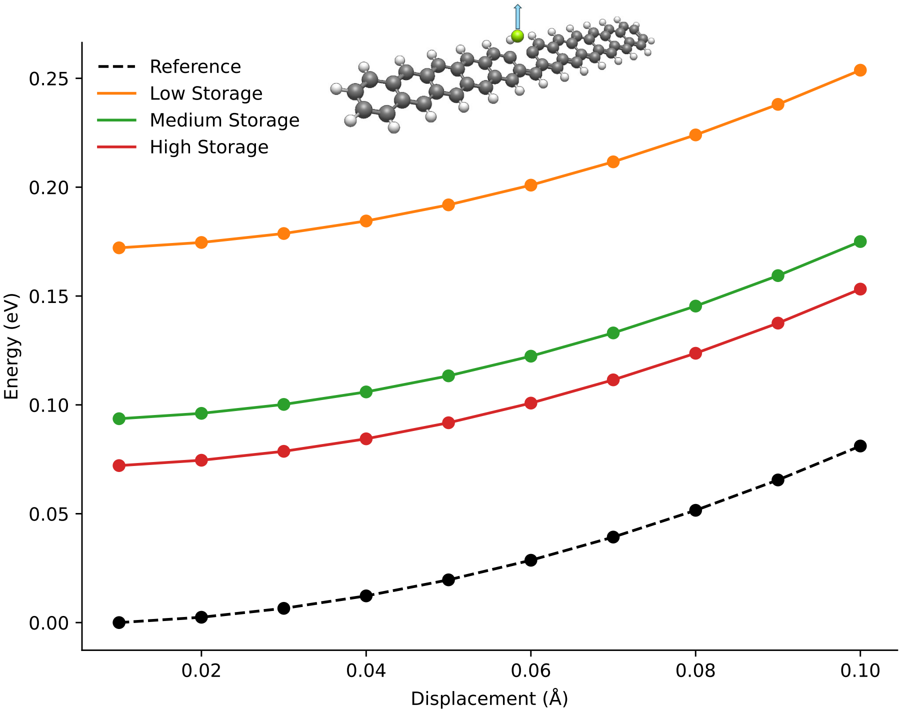

Figure 3.12: The PES of 12-acene as a function of the displacement of a single carbon atom out of plane, computed with a 14-14 active space. Results in the low, medium, and high storage regimes (∼50, ∼130, ∼190 kdoubles) are compared to exact CASCI results.

3.3.3 Effect of Compression on the Potential Energy Surface

It is interesting to consider how a potential energy surface (PES) is effected by compression for two reasons. First, one might expect that compression might introduce non-parallelity error. Second, as one moves from point to point along the PES, one might expect discontinuities as specific singular vector pairs are added or dropped.

To investigate this possibility, we have computed the PES as a function of the out-of-plane displacement of a single carbon atom in 12-acene in three different storage regimes (Fig. 3.12). No discontinuities are observed on the scale of 0.01 eV in any of the plots. Comparing the energies at the first and last points of each curve, we observe very small deviations from parallelity: 5×10^{-4}, 3×10^{-4}, and 4×10^{-5} eV in the low, medium, and high storage regimes, respectively.

3.3.4 Effect of Using Optimal Rank on Compression

In order to assess the necessity of the optimal rank procedure, we compare SR-CHACI (which uses a static rank) to CHACI and TSVD. Figure 3.13 presents the error in the singlet-triplet gap as a function of the total storage for the 14-14 active space. Excepting one fortunate point at 160 kdoubles, the accuracy of SR-CHACI is significantly worse than that of CHACI, but still better than TSVD. However, considering the error in the absolute energies of the singlet and triplet states separately (Figure 11a and b, respectively), it is clear that this is due to error cancellation. Errors in the absolute energy of the singlet state derived from the SR-CHACI wave function are similar to those of TSVD, and much greater than those of CHACI. Further, errors in the SR-CHACI triplet state are slightly larger than TSVD.

Analysis of spin errors tells a similar story. CHACI is much superior to SR-CHACI at reproducing the spin of the original wave function, and SR-CHACI has similar (and sometimes larger) spin errors compared to TSVD. Taken together, we conclude that optimal rank is an essential component of the CHACI algorithm.

3.3.5 Effect of Sorting on Compression

Here we assess the necessity of another feature of the CHACI compression algorithm: the sorting of rows/columns of the **C** matrix prior to compression. To this end, we compare U-CHACI, in which the rows/columns remain unsorted, to CHACI and TSVD. Figure 3.1 presents a heat-map of the uncompressed

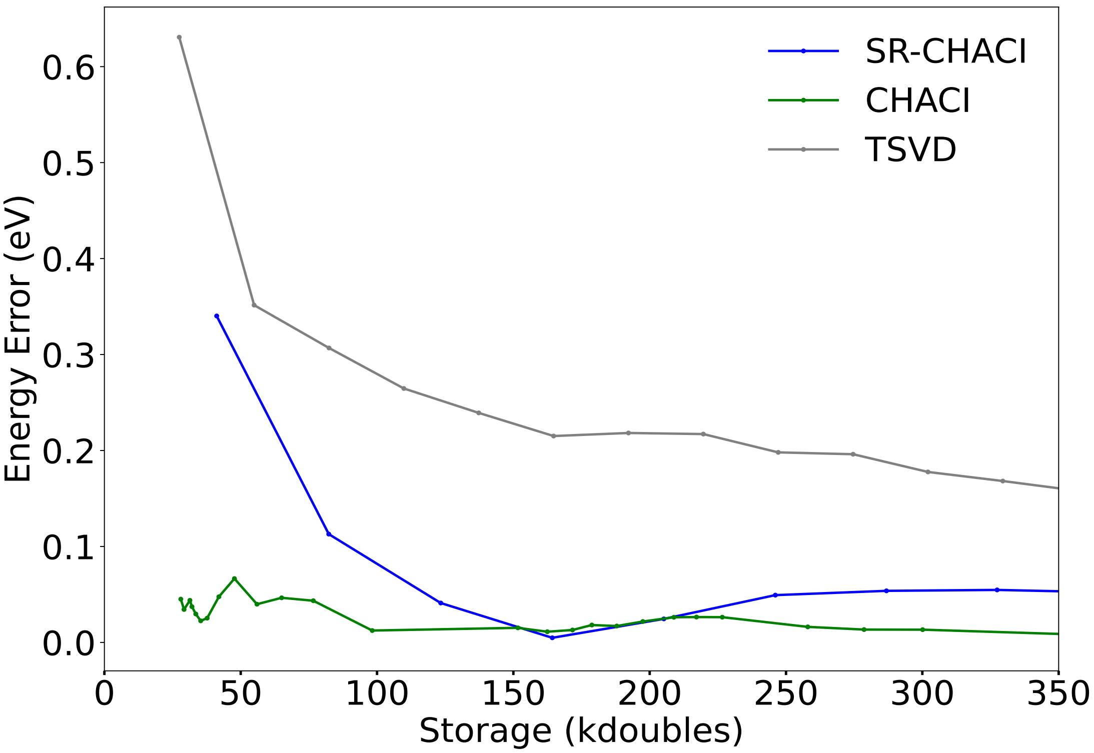

Figure 3.13: The SR-CHACI error (blue) in the singlet-triplet gap of 12-acene as a function of total storage, computed with a 14-14 active space. The TSVD and CHACI errors (gray and green, respectively) are shown for comparison.

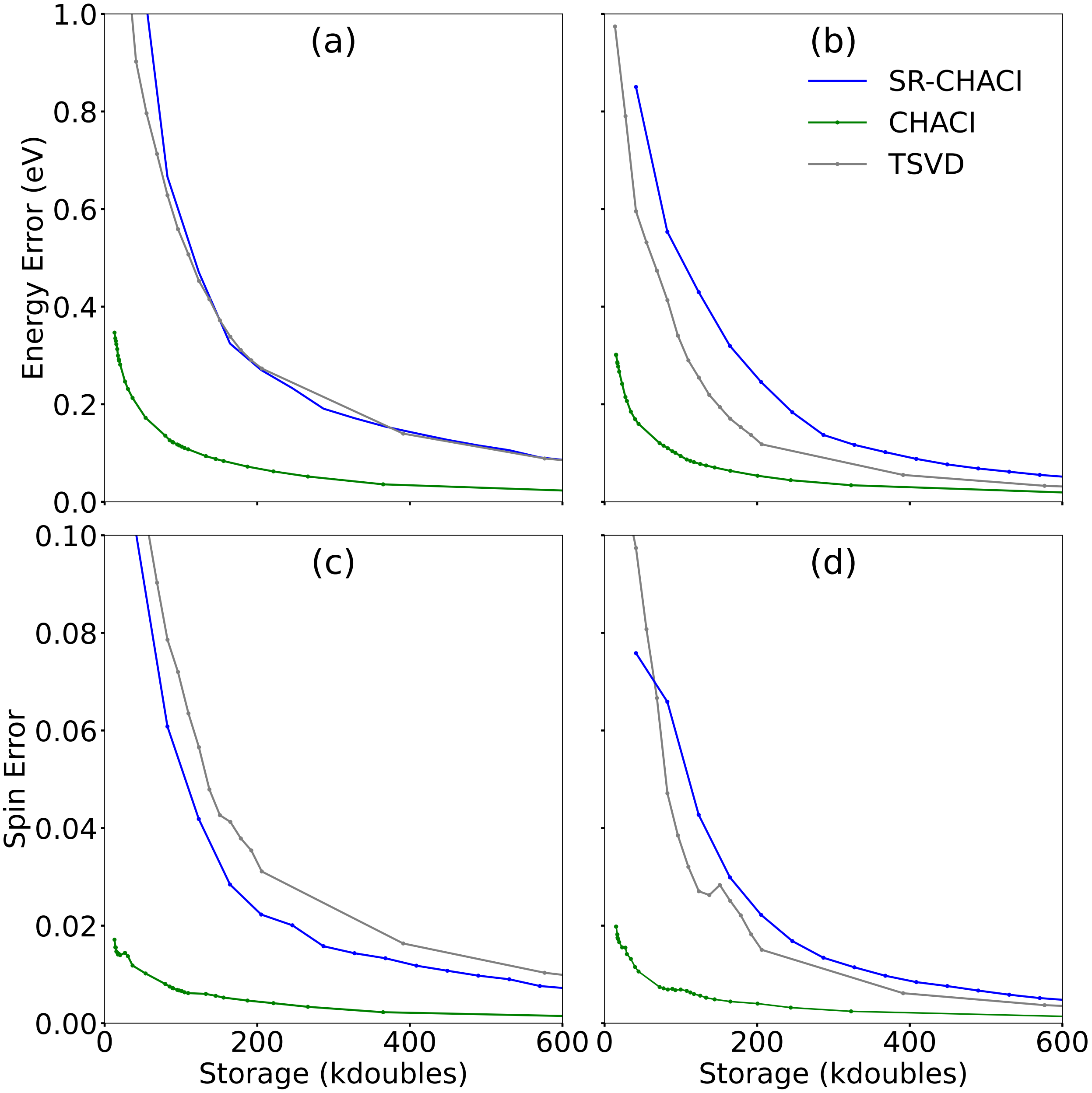

Figure 3.14: The SR-CHACI errors (blue) in the absolute energy (a and b) and spin (c and d) as a function of the storage for 12-acene (14-14 active space). Panels (a) and (c) correspond to the singlet wave function, while (b) and (d) correspond to the triplet wave function. The TSVD (gray) and CHACI (green) errors are shown for comparison.

C matrix of the singlet (panels a and c) and triplet (b and d) wave functions before (a and b) and after (c and d) sorting. Note that sorting concentrates larger elements into the upper-left corner of the matrix.

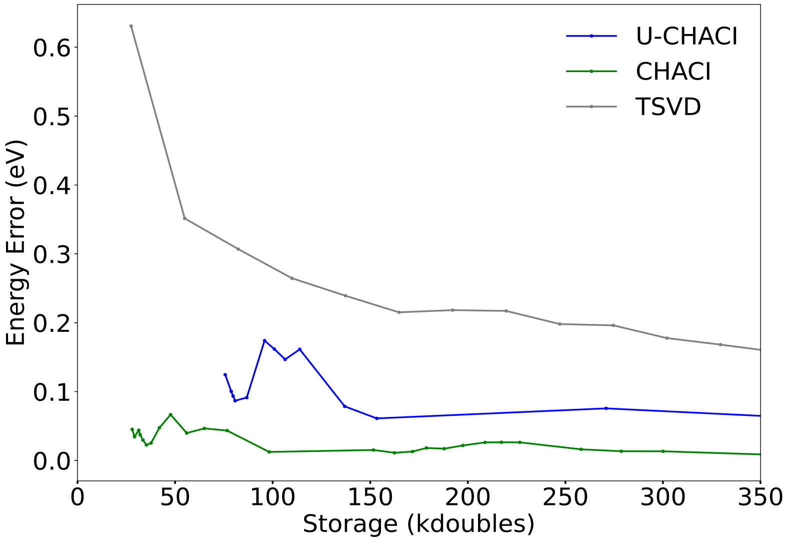

Figure 3.15: The U-CHACI error (blue) in the singlet-triplet gap of 12-acene as a function of total storage, computed with a 14-14 active space. The TSVD and CHACI errors (gray and green, respectively) are shown for comparison.

Figure 3.15 compares the U-CHACI singlet-triplet errors to those of CHACI and TSVD. Though U-CHACI appears to be more accurate for predicting the relative energy than TSVD, it remains inferior to CHACI. Considering the errors in the absolute singlet and triplet energies (Figure 3.14a and b), we see that U-CHACI errors are on the order of the same size as those of TSVD, and considerably larger than those of CHACI. That being said, U-CHACI is solidly between CHACI and TSVD in its ability to accurately describe the total spin angular momentum (Figure 3.14c and d).

Taking this data together, we conclude that sorting is an essential component of the CHACI algorithm. However, our ultimate goal is not to compute the full wave function and subsequently compress it, and the type of *a posteriori* sorting that we use in CHACI would not be possible if we were to directly solve for the wave function in compressed form. But given that the Duch ordering of spin strings does not allow for efficient compression, the determination of an *a priori* scheme by which strings may be ordered for efficient compression remains an important open question.

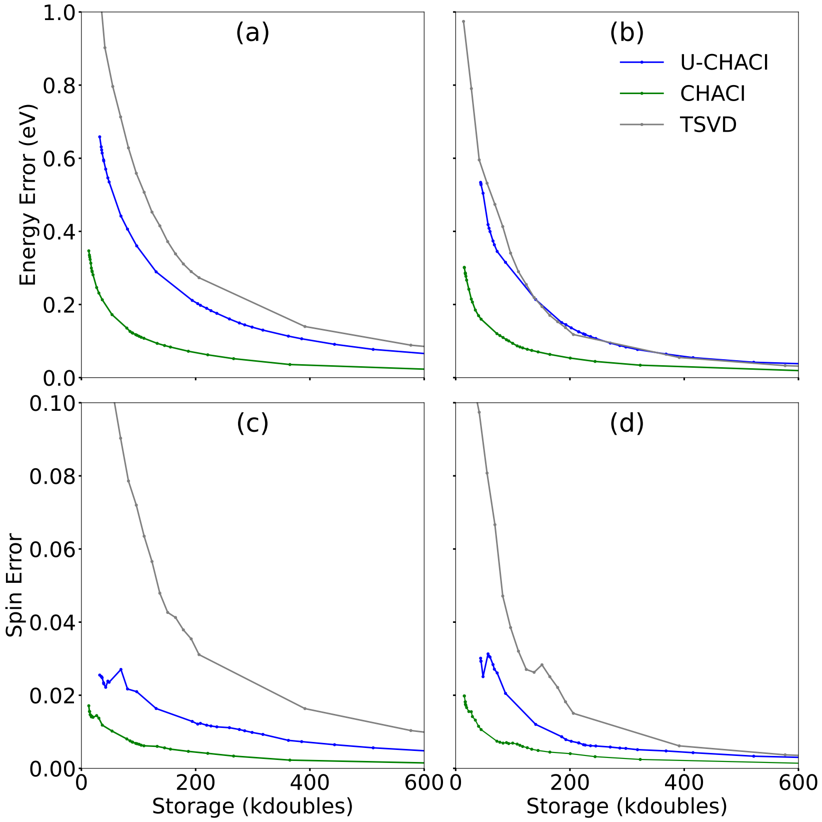

Figure 3.16: The U-CHACI errors (blue) in the absolute energy (a and b) and spin (c and d) errors as a function of the storage for 12-acene (14-14 active space). Panels (a) and (c) correspond to the singlet wave function, while (b) and (d) correspond to the triplet wave function. The TSVD (gray) and CHACI (green) errors are shown for comparison.

3.3.6 Effect of Upper Quadrant vs H-matrix Compression

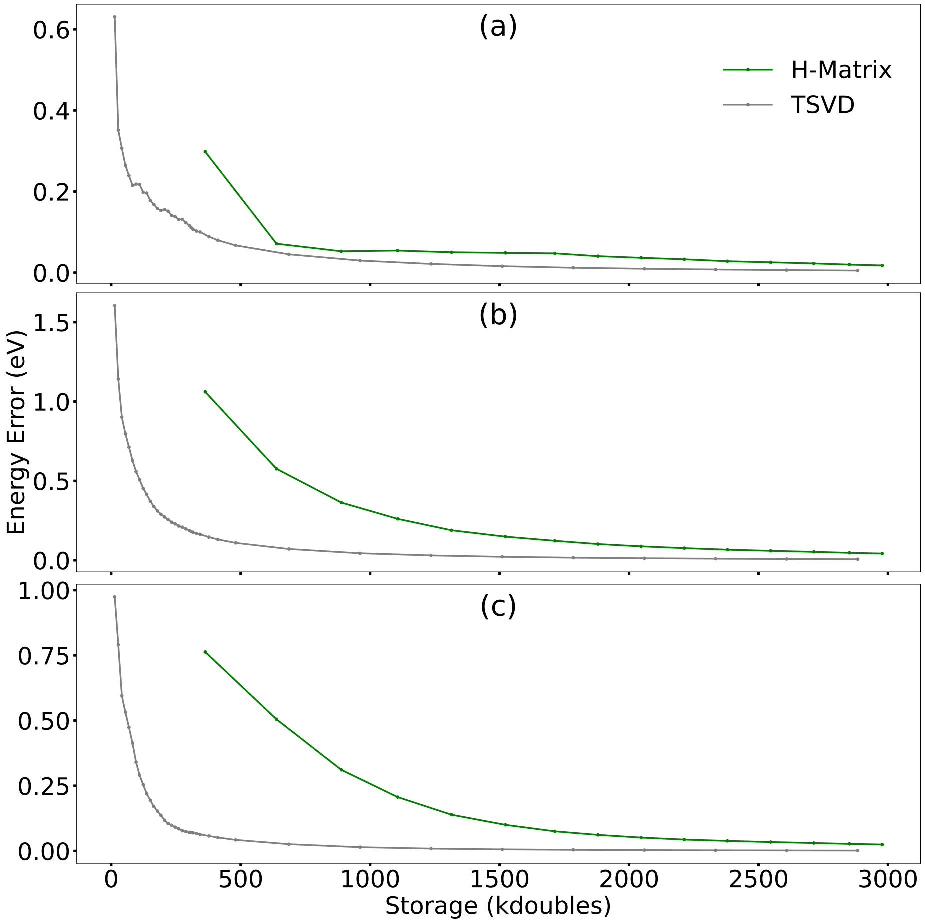

Figure 3.17: Panels (a), (b), and (c) show the errors in the singlet-triplet gap, singlet, and triplet energies, respectively, of 12-acene (14-14 active space) as a function of the required storage for H-matrix and TSVD compression (green and gray).

Lastly, we consider compression of **C** into the H-matrix format,[41] which is designed to leverage the diagonally dominant nature of the matrix, in contrast to the CH blocking scheme used in CHACI. Figure 3.17 shows the error in the singlet-triplet gap, singlet absolute energy, and triplet absolute energy of the 14-14 wave function compressed into H-matrix format. H-matrix compression is inferior to TSVD, thus we conclude that the CH blocking scheme is an essential component of the CHACI scheme.

Chapter 4

Hierarchically Compressed SOS-MP2 Algorithm

In this chapter we describe a hierarchically compressed spin-opposite-scaled (SOS-) MP2 algorithm based on the work of Gao, Jiao, and Levine [42]. The method combines the atomic-orbital (AO) Laplace-transformed formulation of the MP2 energy with an $\mathcal{H}^2$ representation of the electron repulsion integral (ERI) tensor. The key idea is to exploit both the data-sparsity of the ERI tensor in $\mathcal{H}^2$ form and the element-wise sparsity of the energy-weighted density and complementary density matrices.

Since the AO Laplace MP2 formulation and $\mathcal{H}^2$-matrix basics have already been reviewed in previous chapters, we begin directly with the methodology. Following the structure of Ref. [42], we first introduce the partitioning of the ERI tensor, then describe the short- and long-range index transformations, and finally explain how the SOS-MP2 Coulomb-like term is evaluated together with a brief complexity and error analysis.

4.1 Methodology

Here we describe the key steps in our hierarchical SOS-MP2 algorithm, which leverages both the data-sparsity of the ERI tensor in the $\mathcal{H}^2$ format and the element-wise sparsity of the energy-weighted density matrices. The method involves three main stages: partitioning the ERI tensor, performing index transformations on its short- and long-range components, and finally computing the SOS-MP2 energy. The overall procedure is presented in Algorithm 2.

~~Algorithm 2: Hierarchical SOS-MP2 algorithm~~

```
Input: The atomic orbital ERI tensor $W$ in $\mathcal{H}^2$ format. The coefficient matrix
    $C$ and occupied and virtual molecular orbitals $\epsilon_i$. The quadrature weights
    and abscissa $\{w_\alpha, t_\alpha\}$
Output: MP2 Coulomb-like term energy $E_2^{\text{SOS-MP2}}$
for $\alpha$ in quadrature points do
    Calculate $X^\alpha_{\mu\nu} = \sum_i^{\text{occ}} C_{\mu i} C_{\nu i} e^{\epsilon_i t_\alpha}$ and $Y^\alpha_{\mu\nu} = \sum_a^{\text{vir}} C_{\mu a} C_{\nu a} e^{-\epsilon_a t_\alpha}$
    Do $X$ and $Y$ index transformations on $W_s$ to get transformed short-range part $T_s$
    for Block $B_i$ in completely low-rank format do
        for Column $\kappa\epsilon$ in the block do
            for $\lambda$ where $X_{\kappa\underline{\lambda}}$ is the truncated significant elements do
                Compare the block containing column $\underline{\lambda}\epsilon$ with $B_i$
                if They are in the same level then
                    Directly multiply the column basis set using (4.13)
                else if the transformed block is of lower level then
                    Do ancestor index transformation using (4.14)
                else
                    Do descendant index transformation using (4.15)
                end if
            end for
        end for
    end for
    Get the half index transformed low-rank part ERI tensor $(\mu\nu|\underline{\lambda}\epsilon)$
    Similarly do $Y$ index transformation to get transformed long-range part $T_l = (\mu\nu|\underline{\lambda}\overline{\sigma})$
    Loop over all the elements of $T_s$ to compute $\mathrm{Tr}(T_s^2 + 2T_sT_l)$.
    Loop over all the blocks of $T_l$ to compute $\mathrm{Tr}(T_l^2)$.
    Summarize this quadrature point: $e_2^\alpha = -2\,\mathrm{Tr}\left[(T_s^2) + (T_l^2) + 2T_s \cdot T_l\right]$
end for
return $E_2^{\text{SOS-MP2}} = \sum_\alpha w_\alpha e_2^\alpha$
```

4.1.1 Partitioning of the ERI Tensor

The SCF procedure involves the multiplication of the ERI tensor by the density matrix, which can be understood as a matrix-vector product in the space of basis function pairs. Such an operation is well-suited to the $\mathcal{H}^2$-matrix representation. By contrast, the SOS-MP2 method involves operations that resemble a matrix-matrix product, which is less ideal for the direct application of the $\mathcal{H}^2$ format. Nonetheless, for finite systems, the electron density decays exponentially with distance, and insulator-type systems exhibit the fastest decay.[43, 44] As a result, the density and density complementary matrices are expected to be sparse, which allows us to avoid a full, complex matrix-

matrix multiplication. Furthermore, substantial cancellations observed in MP2 calculations suggest that accurate results can be achieved by focusing on the dominant elements in one matrix, thereby further simplifying the computation.

In what follows, we assume that the SCF phase of the calculation is complete, and therefore the orbital coefficient matrices, C, are already available. With C, the energy-weighted density and density complementary matrices, X and Y, can be computed as

$$X_{\mu\nu}^{\alpha} = \sum_{i}^{\text{occ}} C_{\mu i} C_{\nu i} e^{\epsilon_i t_\alpha} \tag{4.1}$$

and

$$Y_{\mu\nu}^{\alpha} = \sum_{a}^{\text{vir}} C_{\mu a} C_{\nu a} e^{-\epsilon_a t_\alpha}. \tag{4.2}$$

Both calculations have a time complexity of $\mathcal{O}(N^3)$. However, these computations are performed only once, and the associated prefactor is sufficiently small that the overall time complexity is negligible for the systems considered in this work. X and Y are both sparse in larger systems. We therefore store them in compressed sparse row (CSR) format. Elements below a user-chosen threshold, η, are approximated as zero.

As described in Section **??**, the Coulomb-like term of the MP2 energy can be expressed as a weighted sum over quadrature points, $E_2 = -\sum_{\alpha}^{\tau} w_\alpha e_2^\alpha$. For clarity, we focus on the calculation of e_2^α at a single quadrature point and omit the superscript α in the derivation that follows. In Section 4.2, we employ the quadrature rules developed by Braess and Hackbusch.[45, 46] Typically, seven or eight quadrature points are used.

The Coulomb-like term of the MP2 energy can be written explicitly as

$$e_2 = -2 \sum_{\mu,\nu,\lambda,\overline{\sigma},\gamma,\delta,\kappa,\epsilon} (\mu\nu|\lambda\sigma) X_{\mu\gamma} Y_{\nu\delta} (\gamma\delta|\kappa\epsilon) X_{\kappa\lambda} Y_{\epsilon\sigma}. \tag{4.3}$$

Using W to denote the ERI tensor $(\mu\nu|\lambda\sigma)$, we can express the equation more compactly as

$$e_2 = -2 \operatorname{Tr}\left((W(X \otimes Y))^2 \right), \tag{4.4}$$

where Tr denotes the trace and $\otimes$ denotes the Kronecker product. Defining

$$T = W(X \otimes Y), \tag{4.5}$$

the energy expression simplifies further to

$$e_2 = -2\,\mathrm{Tr}(T^2). \tag{4.6}$$

We refer to the operation of multiplying W by $X \otimes Y$ as *index transformation*. Here, W denotes the original ERI tensor, while T denotes the index-transformed ERI tensor. As discussed in Section **??**, the ERI tensor, represented as an $\mathcal{H}^2$-matrix, can be decomposed into a short-range component, W_s, which includes diagonal and near-diagonal blocks, and a long-range component, W_l, including the remainder of the matrix. Owing to the linearity of matrix-matrix multiplication,

$$T = T_s + T_l \tag{4.7}$$

where

$$T_s = W_s(X \otimes Y) \tag{4.8}$$

and

$$T_l = W_l(X \otimes Y). \tag{4.9}$$

It is important to note that T_s and T_l do not denote the short- and long-range parts of the index-transformed ERI tensor. Instead, they are the index-transformed versions of the short- and long-range parts of the original ERI tensor. The Coulomb-like term of the MP2 energy then takes the form

$$e_2 = -2\,\mathrm{Tr}\left(T_s^2 + T_l^2 + 2T_sT_l\right). \tag{4.10}$$

4.1.2 Index Transformation of the Short-Range Component

Here we describe the transformation of the short-range component of the ERI tensor, Eq. (4.8). Both the original W_s and the ultimate index-transformed T_s are stored in CSR matrix format in order to leverage their sparsity, with elements whose absolute values fall below a second sparsification threshold, ζ, are approximated as zero. In practice, the transformation is carried out in two steps, applying X and Y to the ERI tensor separately. First, the X index transformation is computed as

$$(\mu\nu|\underline{\lambda}\epsilon) = \sum_{\kappa}(\mu\nu|\kappa\epsilon)X_{\kappa\underline{\lambda}}. \tag{4.11}$$

Following this transformation, the resulting intermediate ERI tensor is sparsified by applying the threshold ζ. The Y index transformation is then performed analogously

$$(\mu\nu|\underline{\lambda}\overline{\sigma}) = \sum_{\epsilon}(\mu\nu|\underline{\lambda}\epsilon)Y_{\epsilon\sigma}. \tag{4.12}$$

4.1.3 Index Transformation of the Long-Range Component

Now we discuss the index transformation of the long-range component, Eq. (4.9). The long-range part of the ERI, W_l, comprises a set of low-rank matrix blocks in an $\mathcal{H}^2$ representation. Since this index transformation applies a right multiplication to W_l, the nested property of the row basis sets is preserved, while the nested property in the column basis sets is lost. Nevertheless, the row basis sets of blocks in the same row of blocks remain connected via transfer matrices.

We begin by converting the $\mathcal{H}^2$ matrix into a more general hierarchical matrix, no longer enforcing the nested property of the column basis. This is done by retaining the row basis sets and multiplying the intermediate matrices by the column basis sets. This step ensures that the nested property of the row basis sets is maintained across different blocks.

Next, we perform the X index transformation on the hierarchical matrix by evaluating

$$(\mu\nu|\underline{\lambda}\epsilon) = \sum_{\kappa}(\mu\nu|\kappa\epsilon)X_{\kappa\underline{\lambda}},$$

and store $(\mu\nu|\underline{\lambda}\epsilon)$ in a *completely low-rank hierarchical matrix* format. Such a matrix is partitioned in the same manner as in the original hierarchical matrix. However, all blocks, including the diagonal and neighboring blocks, are stored in low-rank format. Because right multiplication does not affect the row basis, each block's row basis set in $(\mu\nu|\underline{\lambda}\epsilon)$ matches that in $(\mu\nu|\kappa\epsilon)$. We then iterate over all blocks in $(\mu\nu|\underline{\lambda}\epsilon)$, including both diagonal and neighboring blocks, to compute their column basis sets. Each column basis represents a basis function pair $\underline{\lambda}\epsilon$. When determining the influence of the row elements, $X_{\kappa\underline{\lambda}}$, on the column basis sets, we consider three cases based on the relationship between the row basis sets:

Same-level Index Transformation The row basis set of the block containing $\kappa\epsilon$ is identical to that of the block containing $\underline{\lambda}\epsilon$.

Ancestor Index Transformation The row basis set of the block containing $\kappa\epsilon$ is a subset of the row basis set of the block containing $\underline{\lambda}\epsilon$.

Descendant Index Transformation The row basis set of the block containing $\kappa\epsilon$ is a superset of the row basis set of the block containing $\underline{\lambda}\epsilon$.

Let y denote the column basis vector $\underline{\lambda}\epsilon$ to be computed, and let x denote the column basis set in $\kappa\epsilon$. When the block containing $\kappa\epsilon$ lies in the short-range part, it is treated as an empty block, as this part has already been computed in Section 4.1.2.

In the same-level transformation, we directly multiply the value $X_{\kappa\underline{\lambda}}$ with the column basis vector x to obtain the contribution to y, since the row basis sets of both blocks are identical. This can be written as

$$y = X_{\kappa\underline{\lambda}} x. \tag{4.13}$$

This step is implemented in line 9 of Algorithm 2.

In the ancestor transformation, we trace the sequence of ancestors n_i in the row tree from the block containing $\kappa\epsilon$ to the block containing $\underline{\lambda}\epsilon$. The row basis set of the block containing $\kappa\epsilon$ is effectively the recursive product of the transfer matrices along this ancestor sequence. We express this as

$$y = X_{\kappa\underline{\lambda}} \prod_i R_{n_i} x. \tag{4.14}$$

This step is implemented in line 11 of Algorithm 2.

In the descendant transformation, the procedure is more involved. First, we identify the blocks containing the required column $\kappa\epsilon$, denoted by a set of blocks, B_i, with corresponding row basis sets U_i and columns x_i. For each U_i, we then trace a sequence of ancestors, n_{ij}, in the row tree from the block containing $\underline{\lambda}\epsilon$ down to the block containing U_i. This sequence is the reverse of that in the ancestor transformation, moving from descendant to ancestor rather than vice versa. We write

$$y = \sum_i X_{\kappa\underline{\lambda}} \prod_j R_{n_{ij}}^{-1} x_i, \tag{4.15}$$

where $R_{n_{ij}}^{-1}$ denotes the pseudo-inverse of the transfer matrix, which is precomputed.

The Y index transformation step proceeds analogously to the X transformation step. The key difference is that diagonal and neighboring blocks are no longer treated as empty but instead are represented as low-rank matrices sharing the same row basis sets as their neighbors.

In practice, the error introduced by the long-range index transformation is found to be smaller than that of the short-range transformation, which allows a higher threshold to be used when sparsifying X and Y compared to the threshold used for the short-range part of the index transformation.

4.1.4 Computation of MP2 Energy

With the CSR matrix, T_s, and the completely low-rank hierarchical matrix, T_l, representing the short- and long-range components of the index-transformed ERI tensor, the Coulomb-like term of the MP2 energy is given by (4.10). The terms $\mathrm{Tr}(T_\mathrm{s}^2)$ and $\mathrm{Tr}(T_\mathrm{s}T_\mathrm{l})$ are evaluated directly by iterating over all elements of T_s and identifying the corresponding elements in the other matrix to compute their contributions to the total trace.

For the term $\mathrm{Tr}(T_\mathrm{l}^2)$, due to the symmetric block structure of the hierarchical matrix, it is sufficient to select each block and compute the trace of the product of the block with its mirror image across the diagonal.

4.1.5 Time and Space Complexity

We analyze costs *per quadrature point*, with the understanding that the number of points τ is a small constant ($\sim$7–8) and hence contributes only a constant factor overall. We adopt the following assumptions (standard for insulating finite systems and $\mathcal{H}^2$ compression): (A1) off-diagonal entries of the density and complementary matrices decay exponentially with the distance between basis-function centers;[43, 44] (A2) after thresholding at levels η_s and η_l, the *expected* number of significant entries per row/column of X and Y is $\mathcal{O}(1)$; (A3) the $\mathcal{H}^2$ representation of W_l uses admissible blocks with numerical rank bounded by a geometry-dependent constant $k_{\max}$; (A4) let m be the number of basis-function pairs retained after Schwarz screening ($m \leq N^2$), the hierarchical partition over these m pairs has $\mathcal{O}(\log m)$ levels with constant-size leaves; (A5) transfer matrices are well-conditioned so their (pseudo-)inverses are stable and bounded in size; (A6) formation of X and Y is treated as given for the purpose of asymptotic bounds (e.g., produced during or after SCF using standard linear-scaling routines), i.e., it does not dominate the costs reported below.

Short-range index transformation. A naive dense application of X and Y would be $\mathcal{O}(N^4)$, but by (A2) only $\mathcal{O}(1)$ entries per row/column of X and Y are retained in expectation. Applying these to W_s yields T_s with $\mathcal{O}(m)$ nonzeros and cost $\mathcal{O}(m)$; storage for T_s in CSR is $\mathcal{O}(m)$.

Long-range index transformation. Converting W_l from $\mathcal{H}^2$ to a general hierarchical form by folding column bases into intermediates costs $\mathcal{O}(m \log m)$. The X- and Y-steps each require assembling column bases across all levels. By nearsightedness and the level-wise geometry of the trees, the expected number of significant interactions per target is $\mathcal{O}(1)$, while there are $\mathcal{O}(m)$ targets per level and $\mathcal{O}(\log m)$ levels. Hence the total expected work is $\mathcal{O}(m \log m)$ per step; storage for the completely low-rank result is also $\mathcal{O}(m \log m)$. A detailed derivation of this $\mathcal{O}(m \log m)$ bound for the long-range step is provided in the Supplementary Material.

Energy accumulation. Evaluating $\mathrm{Tr}(T_s^2)$ and $\mathrm{Tr}(2T_sT_l)$ by iterating over the $\mathcal{O}(m)$ nonzeros of T_s costs $\mathcal{O}(m)$. The term $\mathrm{Tr}(T_l^2)$ involves level-wise block products with $\mathcal{O}(m)$ work per level and $\mathcal{O}(\log m)$ levels, i.e., $\mathcal{O}(m \log m)$. Combining the above, the per-quadrature *work* and *storage* are

$$\#(\text{operations}) = \mathcal{O}(m \log m), \qquad \#(\text{storage}) = \mathcal{O}(m \log m), \tag{4.16}$$

which implies the worst-case bounds $\mathcal{O}(N^2 \log N)$ since $m \leq N^2$.

4.1.6 Error Analysis

Now we turn our attention to analyzing the sources of numerical error in our algorithm. The errors in the index transformation procedure primarily arise from neglecting small elements of X and Y to enable sparse storage. Let X_r and Y_r denote the truncated matrices. The resulting error in T_s comprises the terms

$$T_\mathrm{sX} = W_\mathrm{d}(X_\mathrm{r} \otimes Y), \tag{4.17}$$

$$T_\mathrm{sY} = W_\mathrm{d}(X \otimes Y_\mathrm{r}), \tag{4.18}$$

and

$$T_\mathrm{sXY} = W_\mathrm{d}(X_\mathrm{r} \otimes Y_\mathrm{r}), \tag{4.19}$$

where subscript r indicates the residual after sparsification. Analogous considerations apply to T_l. This error behaves similarly to rounding error and is controlled by the thresholds η (for sparsifying X/Y) and ζ (for sparsifying the short-range transforms). If T_sX and T_sY are $\mathcal{O}(\epsilon)$, then T_sXY is $\mathcal{O}(\epsilon^2)$ and typically negligible. In principle, one could reduce the net error from $\mathcal{O}(\epsilon)$ to $\mathcal{O}(\epsilon^2)$ by evaluating $\mathrm{Tr}[T_\mathrm{s}(T_\mathrm{sX} + T_\mathrm{sY})]$, which would add only $\mathcal{O}(m)$ work. However, in our experiments, the observed error is already sufficiently small and comparable to the intrinsic error in $\mathrm{Tr}(T_\mathrm{s}^2)$, which cannot be corrected as efficiently, so we omit this step.

Errors from the low-rank approximation and the low-rank index transformation are dominated by the descendant transformation, where pseudo-inverses of transfer matrices are used. With well-conditioned transfer matrices and bounded numerical ranks, this contribution is negligible. The Laplace transformation itself introduces additional numerical error that is well understood and typically much smaller than the MP2 model error.

4.1.7 Potential for Parallel Implementation

In quantum chemistry, along with physical approximations and efficient numerical methods, parallelization is a key strategy to extend the size and complexity of systems that may be studied.[47–57] That hierarchical matrices naturally map to massively parallel computer architectures is a significant advantage that will be exploited in future work. In Laplace transform MP2, the computation of the MP2 energy at each quadrature point is independent, making the algorithm highly parallelizable. Furthermore, each row computation in T_s and each block computation in T_l are also independent, enabling parallelization of the short- and long-range parts of the index transformation, respectively. This same property applies to the MP2 energy computation step. In other words, the Hierarchical SOS-MP2 algorithm could theoretically achieve constant time complexity with an infinite number of processors. However, since parallelization requires considerable memory, we implemented parallelization only for the quadrature points in this proof-of-concept paper. We apply OpenMP with 8 threads on this parallelization because there are usually 7 or 8 number of quadrature points in the Laplace transformation. Future work will involve parallelizing other parts of the algorithm on massively parallel computers.

4.2 Results

All computations were carried out on the Seawulf cluster at Stony Brook University. We selected two model systems to study: an alkane chain system, which is representative of one-dimensional systems, and a water cluster system, which is representative of fully three-dimensional systems. All SOS-MP2 calculations were performed using the cc-pVDZ basis set. The threshold used for sparsification of both the short-range part of the ERI tensor and the index-transformed ERI tensor was set to $\zeta = 1 \times 10^{-6}$.

Different thresholds for sparsification of X and Y are used in the short- and long-range transformations, labeled η_s and η_l, respectively. Except as noted below, we set the short-range truncation threshold to $\eta_s = 10^{-5}$ for the alkane chain systems and $\eta_s = 3 \times 10^{-4}$ for the water cluster. For the long-range transformation, thresholds are $\eta_l = 10^{-4}$ for the alkane system and $\eta_l = 3 \times 10^{-4}$ for the water cluster.

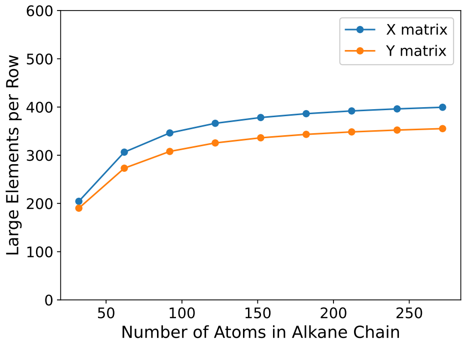

Figure 4.1: The number of significant values per row or column in the energy-weighted density matrices in the series of linear alkane chains.

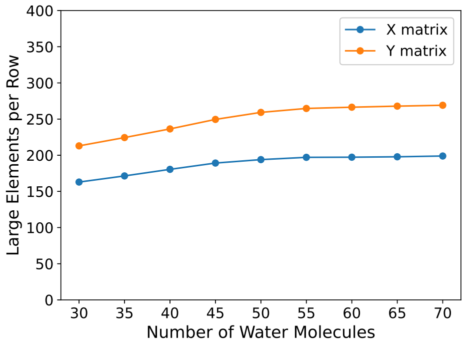

Figure 4.2: The number of significant values per row or column in the energy-weighted density matrices in the series of water clusters.

4.2.1 The Sparsity of Density and Density Complementary Matrices

The performance of our algorithm is contingent on the sparsity of the energy-weighted density matrices X and the density complementary matrices Y, thus we quantify that sparsity here. We report the number of significant elements per row in Figure 4.1 for the alkane chain and Figure 4.2 for the water cluster. Values were computed across all quadrature points, and the maximum value was retained. For the alkane system, we observe rapid growth in the number of significant values from $C_{10}H_{32}$ to $C_{30}H_{92}$. Beyond this range, growth slows and eventually saturates, consistent with the expected scaling of $\mathcal{O}(N)$ total significant density matrix element for large systems. The water cluster exhibits similar behavior—as the system size increases, the number of significant values per row reaches a plateau. These results confirm the asymptotic sparsity of both the density and complementary matrices in both 1-D and 3-D systems.

4.2.2 Time Complexity

The measured times-to-solution for a series of linear alkane chains is shown in Figure 4.3. All timing results are averaged over three runs. Molecular structures were rendered using Jmol.[58]

The first four data points correspond to systems $C_{10}H_{22}$ through $C_{40}H_{82}$, in which the X and Y matrices have not yet reached asymptotic sparsity. Beyond this point, the number of nonzero elements per row of X and Y stabilizes. To illustrate the role of sparsity, we include two trend lines. For sufficiently

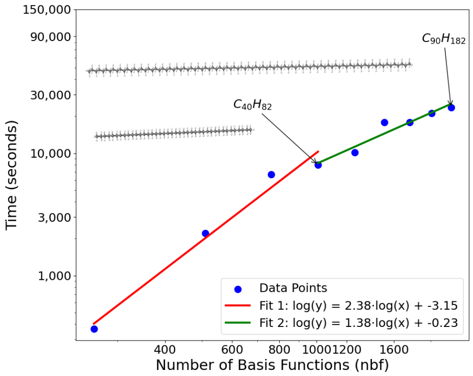

Figure 4.3: The time-to-solution versus number of basis functions for the series of linear alkane chains.

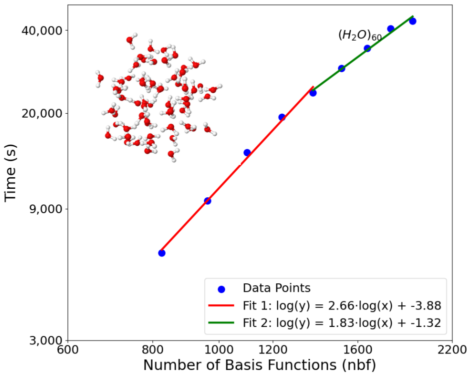

Figure 4.4: The time-to-solution versus number of basis functions for the series of water clusters.

large systems, we expect time complexity to scale as $\mathcal{O}(N^2 \log N)$, though the observed performance is closer to $\mathcal{O}(N^{1.38})$. The superior performance is likely due to Schwarz screening. For smaller systems, complexity grows more quickly, which is consistent with expectations. The observed $\mathcal{O}(N^{2.38})$ scaling is still better than the theoretical $\mathcal{O}(N^3 \log^2 N)$ bound. A discontinuity is observed between $\mathrm{C}_{50}\mathrm{H}_{102}$ and $\mathrm{C}_{60}\mathrm{H}_{122}$, corresponding to an increase in the number of levels in the hierarchical tree. Despite this, the increase in time-to-solution remains nearly linear.

Times-to-solution for the water cluster system are shown in Figure 4.4. For clusters with 30–50 molecules (first five points), time complexity scales approximately as $\mathcal{O}(N^{2.66})$. For clusters with 50–70 molecules (last five points), scaling improves to $\mathcal{O}(N^{1.83})$. The absolute time cost is higher for the water clusters than for the alkanes, as expected for a 3-D system, in which a larger fraction of the interactions are short range.

It is worth noting that, for the largest water clusters, we modified the hierarchical block-splitting algorithm relative to what was used for smaller clusters and alkane chains. In 3-D systems, the number of interacting block pairs grows rapidly with system size. To mitigate this, we cap the maximum depth of the block tree to control the number of interactions. This change improves performance and preserves the desired asymptotic behavior.

We also tested different values of η_s and η_l. We find that the threshold also does not affect the asymptotic growth of the time complexity of the method.

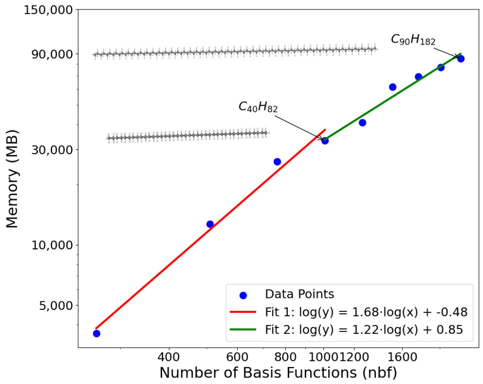

Figure 4.5: The memory versus number of basis functions for the series of linear alkane chains.

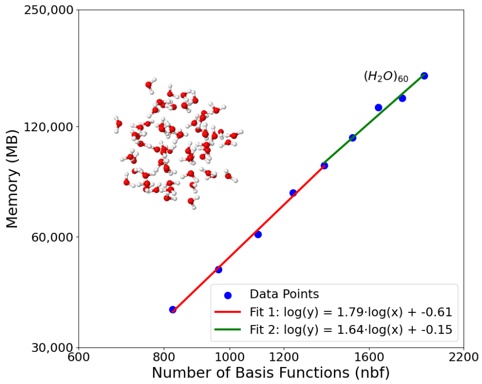

Figure 4.6: The memory versus number of basis functions for the series of water clusters.

4.2.3 Space Complexity

The total memory storage needed for the alkane chain systems is shown in Figure 4.5. Their behavior closely parallels the time complexity results, though the rate of growth is even slower. For systems smaller than $\mathrm{C}_{40}\mathrm{H}_{82}$, where X and Y have not yet reached asymptotic sparsity, storage scales as $\mathcal{O}(N^{1.68})$. For larger systems, the theoretical scaling is $\mathcal{O}(N^2 \log N)$, but observed performance is $\mathcal{O}(N^{1.22})$, approaching linear. As in the time complexity case, we observe a step increase between $\mathrm{C}_{50}\mathrm{H}_{102}$ and $\mathrm{C}_{60}\mathrm{H}_{122}$, again due to an increase in the number of levels in the hierarchical tree.

Figure 4.6 presents the memory storage for the water clusters. For clusters with 30–50 molecules, the observed scaling is approximately $\mathcal{O}(N^{1.79})$, while storage scales as $\mathcal{O}(N^{1.64})$ for clusters with 50–70 molecules. In contrast to the alkane chain system, this scaling does not flatten substantially with increasing system size, but remains within acceptable bounds and significantly better than the theoretical bound.

We attribute the slowdown in the alkane chain's asymptotic growth to Schwarz screening. In 1-D systems, the number of atoms within a given radius of a particular atom is small, and the majority of atom pairs interact weakly. Schwarz screening is therefore highly effective in eliminating small terms, yielding reduced memory costs. In 3-D systems, more pairs of atoms are close enough to interact strongly, leaving fewer terms that can be ignored. As a result, memory cost does not flatten as dramatically.

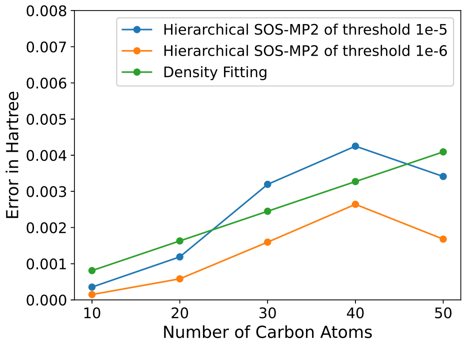

Figure 4.7: The error for alkane chain, computed with default thresholds ($\eta_s = 10^{-5}$, $\eta_l = 10^{-4}$) and tighter thresholds ($\eta_s = 10^{-6}$, $\eta_l = 10^{-5}$), are shown in blue and orange, respectively.

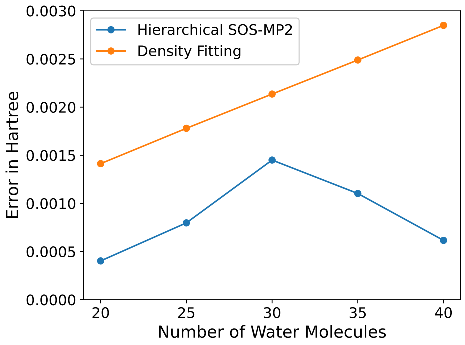

Figure 4.8: The error for water cluster using default thresholds.

4.2.4 Energetic Accuracy

We have carried out error analysis only on small systems. To validate our method, we compute the Coulomb-like component of the MP2 energy using Molpro.[59–61] These reference calculations were performed both exactly and with density fitting with default parameters and compared to the results from our hierarchical SOS-MP2 algorithm. Total error in energy relative to exact MP2 as a function of system size is presented for our alkane and water cluster models in Figures 4.7 and 4.8. The observed error relative to exact MP2 is comparable to that of density fitting (DF), but achieved at significantly lower cost in both time and memory. Interestingly, the error in our results is not observed to grow linearly with system size. The dominant source of error in our implementation is due to thresholding of the X and Y matrices. This error behaves more like numerical rounding error and typically scales as $\mathcal{O}(\sqrt{N})$. Although this introduces some mild and irregular oscillations in the total error, it nonetheless grows more slowly than linearly.

The thresholds used for sparsification of X and Y can be adjusted to improve accuracy. For the time and space complexity studies presented in Sections 4.2.3 and 4.2.2, we used the default thresholds listed above, which are already sufficiently small for most practical applications. However, as shown in Figure 4.7, this error can be reduced further by tightening the threshold by a factor of ten ($\eta_s = 10^{-6}$, $\eta_l = 10^{-5}$). Varying the threshold does not affect the qualitative trend in the sparsity growth of the X and Y matrices; instead, it simply shifts the onset of convergence.

For water cluster systems, we performed additional tests on smaller examples and found that the error remained quite small in practice. As illustrated in Figure 4.8, the error is notably smaller than that of DF—even though we employed relatively loose truncation thresholds for the X and Y matrices. We attribute this improved accuracy to cancellation effects within the sparsified matrices. Although each individual truncation introduces error, these errors tend to cancel out when summed over the full ERI contraction, resulting in significantly lower total error.

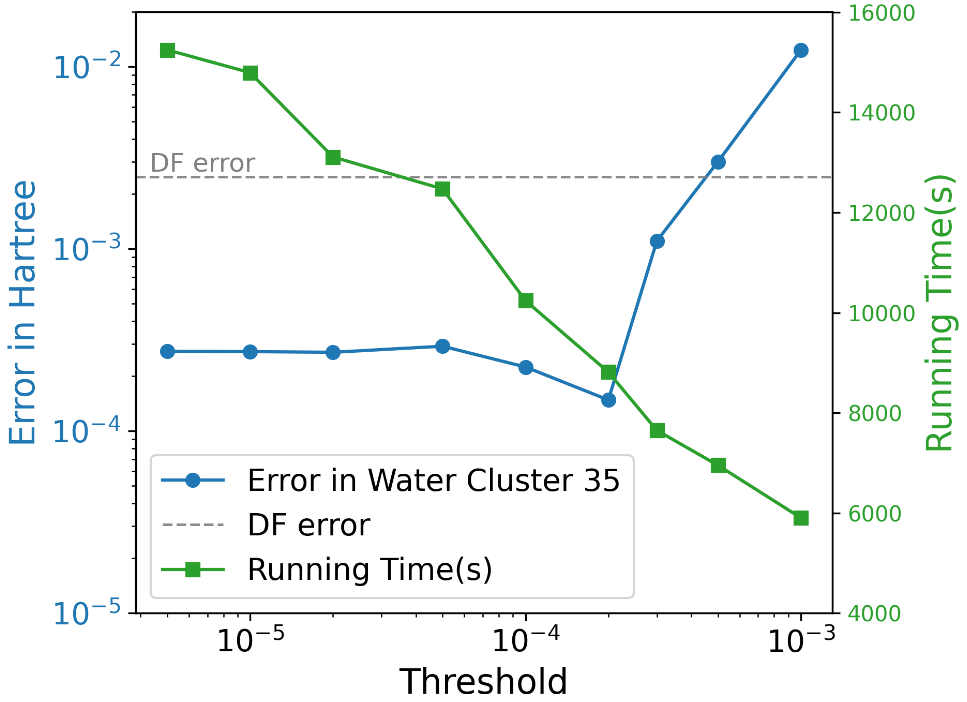

Figure 4.9: The error and time-to-solution for a water cluster system with 35 water molecules as a function of both thresholds, which are set equal to one another, $\eta_s = \eta_l$. The grey dashed line indicates the accuracy of DF-MP2 with default parameters.

Our experiments across multiple systems and threshold values reveal a sharp drop in threshold near $\eta_s = 1 \times 10^{-4}$. Beyond that point, the impact of the threshold on the error slows considerably. By choosing a sparsification threshold of $\zeta = 1 \times 10^{-8}$ Hartree for the short-range part of the ERI tensor, and $\eta_s = 1 \times 10^{-8}$ for the X and Y matrices, the total error can be pushed below 1×10^{-5} Hartree—comparable to the inherent numerical error of the Laplace transformation itself. However, in most cases, an error of 1×10^{-4} Hartree (roughly 0.06 kcal/mol) is entirely acceptable, especially given that this is far smaller than the intrinsic error of MP2 itself. Accordingly, we opted for a balanced choice of thresholds (detailed in Section 4.2.1) when plotting and memory costs. Figure 4.9 illustrates how the error and running time behave as a function of threshold ($\eta_s = \eta_l$) for a 35-molecule water cluster. When the threshold is relaxed, the error remains almost unchanged below a certain value, but increases rapidly once this critical point is passed. By contrast, the running time decreases gradually.

While these empirical findings are promising, we currently lack a complete theoretical understanding of the underlying cancellation behavior. It is possible that some deeper symmetry in the structure of the ERI tensor and the X and Y matrices drives this error suppression. We intend to explore this question further, as it may reveal new opportunities for improved performance and accuracy.

4.3 Attempts on the exchange-like terms

This part is planned for future work.

4.4 Comparison with tensor hyper-contraction

It is instructive to contrast the present $\mathcal{H}^2$-compressed SOS-MP2 formulation with tensor hyper-contraction (THC), particularly least-squares THC (LS-THC) in the AO basis [62]. Both approaches ultimately exploit locality, but they do so in conceptually different ways and therefore lead to different computational bottlenecks and tunable approximations. In our algorithm, the central quantity at each Laplace quadrature point is the operator application

$$T = W(X \otimes Y), \qquad e_2^{\alpha} = -2\,\mathrm{Tr}\big[(T_s^{\alpha})^2 + (T_l^{\alpha})^2 + 2\,T_s^{\alpha}T_l^{\alpha}\big]\,, \tag{4.20}$$

where W is the AO ERI tensor viewed as a matrix over AO pairs, and efficiency is obtained by (i) representing W as a hierarchically compressed operator, (ii) splitting it into short- and long-range parts $W_s + W_l$, and (iii) applying W to element-sparse, Laplace-weighted density-like objects X and Y using sparse index transformations and hierarchical basis transfers. In other words, we leave the ERIs conceptually intact and accelerate the *action* of the Coulomb operator on structured inputs.

THC, by contrast, replaces the four-index ERIs by an explicit low-order tensor factorization and reorganizes the MP2 contractions around an auxiliary index set. In its canonical form, THC approximates ERIs as

$$(pq|rs) \approx \sum_{P,Q} X^{\mathrm{THC}}_{Pp}\, X^{\mathrm{THC}}_{Pq}\, Z_{PQ}\, X^{\mathrm{THC}}_{Qr}\, X^{\mathrm{THC}}_{Qs}, \tag{4.21}$$

so the "difficulty" of the four-index object is shifted into the auxiliary core matrix Z and the collocation factors X^{THC}. In LS-THC, X^{THC} is prescribed on a molecular grid as (weighted) orbital values, and Z is determined by a least-squares fit through a metric on the auxiliary space. When implemented in the AO basis, the practical advantage is that X^{THC} can be made sparse due to the locality of atom-centered Gaussian functions, and one can screen small collocation values using a user threshold (often denoted X_{thre}). This moves much of the remaining work into matrix multiplications over the auxiliary indices, which is highly compatible with GPU-accelerated linear algebra.

This difference in viewpoint leads to a different cost profile. In our $\mathcal{H}^2$ approach, the dominant operations are the repeated sparse index transformations through W_s and the hierarchical long-range transformations through W_l, followed by trace accumulation using the split $T = T_s + T_l$. In AO-THC/LS-THC, the *energy* evaluation can be organized to be low-scaling once X^{THC} and Z are available, but the precomputation of Z introduces an additional global step: forming and (pseudo-)inverting an auxiliary-space metric (and, in practice, applying eigenvalue truncation for stability). Thus, compared to our method—which controls accuracy primarily through hierarchical low-rank truncations in W_l and sparsification thresholds for X, Y, and intermediates—THC controls accuracy primarily through the quality/size of the auxiliary grid and the numerical stabilization choices made in constructing Z.

Finally, the two methods favor different implementation styles. The present algorithm is naturally expressed as a sequence of operator applications and tree-based traversals with irregular sparsity patterns, which maps well onto CPUs and task-based parallelism and benefits from the separability of quadrature points and hierarchical blocks. THC intentionally reorganizes work into BLAS-like contractions (sparse–dense and dense–dense) over auxiliary indices, which is particularly attractive on GPUs, at the expense of additional preprocessing and storage associated with the auxiliary representation. These distinctions clarify that the two approaches are not merely notational variants: THC changes the representation of the ERIs via a fitted factorization (4.21), whereas our method changes the way ERIs are *applied* by combining hierarchical operator compression with double sparsity in the Laplace-weighted density objects.

Chapter 5

Preliminaries and Related Work

5.1 Introduction and Background

Iterative methods are foundational in numerical computation and optimization. Many algorithms can be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k), \tag{5.1}$$

where the goal is to approach a fixed point $\boldsymbol{x}_*$ satisfying $\boldsymbol{g}(\boldsymbol{x}_*) = \boldsymbol{x}_*$. Classical convergence characterizations—such as Q-order and R-order—are well established for linear and superlinear behaviors [**? ? ? ? ? ?**]. However, these traditional notions become less expressive when encountering:

- **Nonstandard sublinear/superlinear rates** involving logarithmic corrections or fractional powers;
- **Weak regularity** (e.g., only Hölder-continuous derivatives) where "fractional-order" convergence arises;
- **Comparability issues** between different iterative families under mixed scaling.

To address these, we propose the *P-order (Power-order)* framework, which introduces a general scale function $\psi(k)$ to unify different convergence regimes:

$$\|\boldsymbol{x}_k - \boldsymbol{x}_*\| \approx (C_\psi)^{\psi(k)}, \qquad C_\psi \in (0, 1).$$

This representation allows geometric, polynomial, and even logarithmically corrected behaviors to coexist under one analytical language.

—

5.2 P-Order Framework

5.2.1 Definition

[P-Order] Given $\boldsymbol{x}_k \to \boldsymbol{x}_*$ and $\xi_k = \|\boldsymbol{x}_k - \boldsymbol{x}_*\|$, if there exists a function $\psi : \mathbb{N} \to (0, \infty)$ with $\psi(k) \to \infty$ and a constant $C_\psi \in (0, 1)$ such that

$$\limsup_{k\to\infty} \xi_k^{1/\psi(k)} = C_\psi,$$

then $\{\boldsymbol{x}_k\}$ converges to $\boldsymbol{x}_*$ with *P-order-ψ*.

[Quasi-Uniform and Uniform P-Orders] If the limit above exists, the sequence is said to have *QUP-order-ψ*. If there exist constants $m, M > 0$ such that

$$mC_\psi^{\psi(k)} \leq \xi_k \leq MC_\psi^{\psi(k)}$$

for all large k, we say it has *UP-order-ψ*.

5.2.2 Key Property: Relationship to Q-Order

[Q-Superlinear ⇒ UP-Superlinear] If $\lim_{k\to\infty} \xi_{k+1}/\xi_k^q = Q_q \in (0, \infty)$ for some $q > 1$, then $\xi_k = \Theta(C_\psi^{q^k})$ for $C_\psi = e^{-s} \in (0, 1)$, meaning the sequence converges UP-order-ψ with $\psi(k) = q^k$.

Proof. Define $f(k) = -\ln \xi_k$. Then $f(k + 1) = qf(k) + d(k)$ with $d(k) = -\ln(\xi_{k+1}/\xi_k^q) \to -\ln Q_q$. Dividing by q^{k+1}, summing, and bounding the tail series shows $f(k) = sq^k + \mathcal{O}(1)$ with $s > 0$. Hence $\xi_k = e^{-f(k)} = \Theta(e^{-sq^k}) = \Theta(C_\psi^{q^k})$. □

—

5.3 Refined Convergence Analysis of Fixed-Point Iterations

Consider the iteration $\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$ where $\boldsymbol{g}(\boldsymbol{x}_*) = \boldsymbol{x}_*$. When $\nabla \boldsymbol{g}(\boldsymbol{x}_*) = 0$, the method may achieve superlinear or higher-order convergence.

[UP- and QUP-Higher-Order Convergence] Let $\boldsymbol{g}$ be q-times differentiable near $\boldsymbol{x}_*$, satisfying

$$\partial_{\boldsymbol{v}}^{(k)} \boldsymbol{g}(\boldsymbol{x}_*) = 0 \quad (k = 1, \ldots, q - 1),$$

and $\partial_{\boldsymbol{v}}^{(q)} \boldsymbol{g}(\boldsymbol{x}_*) \neq 0$ for some unit $\boldsymbol{v}$. Then for $\boldsymbol{x}_0$ sufficiently close to $\boldsymbol{x}_*$, the iteration converges with UP-order q. If derivatives are only one-sided, the rate is QUP-order-q.

Sketch. Using Taylor expansion,

$$\boldsymbol{\xi}_{k+1} = \frac{1}{q!} \partial_{\boldsymbol{\xi}_k}^{(q)} \boldsymbol{g}(\boldsymbol{x}_*) + \mathcal{O}(\|\boldsymbol{\xi}_k\|^{q+\epsilon}),$$

thus $\xi_{k+1} = \Theta(\xi_k^q)$. The leading term ensures bounded ratio ξ_{k+1}/ξ_k^q, implying UP-order-q convergence. □

—

5.4 Sublinear and Fractional-Power Convergence

[Fractional-Power Sublinear Behavior] Suppose near $\boldsymbol{x}_*$ there exist constants $C_0 > 0$, $s > 0$ such that

$$\boldsymbol{u}^\top(\boldsymbol{g}(\boldsymbol{x}_* + \boldsymbol{\xi}) - \boldsymbol{x}_*) = \boldsymbol{u}^\top \boldsymbol{\xi}\big(1 - C_0(-\ln(\boldsymbol{u}^\top \boldsymbol{\xi}))^{-s} + o((-\ln(\boldsymbol{u}^\top \boldsymbol{\xi}))^{-s})\big),$$

while other directions contract linearly. Then $\boldsymbol{x}_k \to \boldsymbol{x}_*$ sublinearly with a fractional-power type rate (capturable via an appropriate $\psi(k)$).

Proof. Let $u_k = \boldsymbol{u}^\top(\boldsymbol{x}_k - \boldsymbol{x}_*)$ and $w_k = -\ln u_k$. Then $w_{k+1} = w_k + C_0 w_k^{-s} + o(w_k^{-s})$, implying $w_k \sim (\text{const})\, k^{1/(1+s)}$. Thus $u_k = e^{-w_k}$ decays sublinearly, matching a fractional-power P-order. □

This mechanism explains phenomena observed in gradient methods under weak smoothness, where rates slower than $\mathcal{O}(1/k)$ occur but still fit the P-order pattern [**?** **?** **?**].

—

5.5 Multipoint Iterative Methods under Hölder Continuity

Multipoint schemes use several function evaluations to achieve high local order [**?** **?**]. If derivatives are only Hölder continuous, one replaces classical Taylor expansions with a $C^{k,\nu}$ version:

[Hölder–Taylor Expansion] If $f \in C^{k,\nu}$ near x_* $(0 < \nu \leq 1)$, then

$$f(x) = \sum_{j=0}^{k} \frac{f^{(j)}(x_*)}{j!}(x - x_*)^j + \mathcal{O}(|x - x_*|^{k+\nu}).$$

Using this, multipoint methods under $C^{k,\nu}$ assumptions satisfy

$$|e_{k+1}| \leq A|e_k|^p + B|e_k|^{p+\nu},$$

yielding effective order between p and $p+\nu$. P-order expresses this continuum uniformly via $\psi(k)$.

—

5.6 Conclusions and Outlook

P-order provides a unified formalism that subsumes linear, superlinear, and sublinear convergence behaviors within a single scale–base pair (ψ, C_ψ). It clarifies ambiguous intermediate cases, accommodates weak regularity, and offers a common metric for comparing iterative schemes.

Future work may extend this framework to stochastic or noisy settings [**?** **?**], link C_ψ quantitatively to problem conditioning, and explore optimal rate frontiers for multipoint methods under mixed smoothness assumptions.

—

Chapter 6

Conclusion

bla

6.1 Part 1

blabla

Bibliography

[1] Joseph F. Traub. *Iterative Methods for the Solution of Equations.* Prentice-Hall, Englewood Cliffs, NJ, 1964.

[2] Alexander M. Ostrowski. *Solution of Equations and Systems of Equations.* Academic Press, New York, 1966.

[3] James M. Ortega and Werner C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[4] John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* SIAM, Philadelphia, 1996.

[5] Florian A. Potra and Vlastimil Pták. *Nondiscrete Induction and Iterative Processes.* Pitman, Boston, 1984.

[6] Florian A. Potra. On q-order and r-order of convergence. *Journal of Optimization Theory and Applications*, 63(3):415–431, 1989.

[7] L. O. Jay. A note on q-order of convergence. *BIT Numerical Mathematics*, 42(2):370–378, 2002.

[8] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, New York, 2 edition, 2006.

[9] Wei Sun and Ya xiang Yuan. *Optimization Theory and Methods: Nonlinear Programming.* Springer, Berlin, 2006.

[10] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[11] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 4 edition, 2013.

[12] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion.* SIAM, Philadelphia, 1998.

[13] Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

[14] Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

[15] Wolfgang Hackbusch and Stefanie Börm. Data-sparse approximation of boundary element matrices. *SIAM Journal on Scientific Computing*, 24 (4):1107–1133, 2002.

[16] Mario Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems.* Springer, Berlin, 2008.

[17] Wolfgang Hackbusch. *Hierarchical Matrices: Algorithms and Analysis.* Springer, Cham, 2015.

[18] Stefanie Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003.

[19] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory.* McGraw-Hill, New York, 1989.

[20] Trygve Helgaker, Poul Jørgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory.* Wiley, Chichester, 2000.

[21] C. David Sherrill and Henry F. Schaefer III. The configuration interaction method: Advances in highly correlated approaches. *Advances in Quantum Chemistry*, 34:143–269, 1999.

[22] Garnet Kin-Lic Chan and Martin Head-Gordon. Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group. *Journal of Chemical Physics*, 116(11):4462–4476, 2002.

[23] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[24] S. F. Boys. Electronic wave functions. i. a general method of calculation for the stationary states of any molecular system. *Proceedings of the Royal Society of London A*, 200:542–554, 1950.

[25] Martin Head-Gordon and John A. Pople. A method for two-electron gaussian integral and integral derivative evaluation using recurrence relations. *Journal of Chemical Physics*, 89(9):5777–5786, 1988.

[26] James L. Whitten. Coulombic potential energy integrals and approximations. *Journal of Chemical Physics*, 58(10):4496–4501, 1973.

[27] Nelson H. F. Beebe and Biørner Linderberg. Simplifications in the generation and transformation of two-electron integrals in molecular calculations. *International Journal of Quantum Chemistry*, 12(4):683–705, 1977.

[28] Francesco Aquilante, Lina Boman, Jesper Boström, et al. Cholesky decomposition techniques in electronic structure theory. *Molecular Physics*, 108(22):2593–2604, 2010.

[29] Egon G. Hohenstein and Todd J. Martinez. Tensor hypercontraction in quantum chemistry. *Journal of Chemical Physics*, 137(4):044103, 2012.

[30] Weitao Yang. Direct calculation of electron density in density-functional theory. *Physical Review Letters*, 66(11):1438–1441, 1991.

[31] Stefan Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71(4):1085–1123, 1999.

[32] Xin Xing, Hua Huang, and Edmond Chow. A linear scaling hierarchical block low-rank representation of the electron repulsion integral tensor. *Journal of Chemical Physics*, 153(8):084119, 2020. doi: 10.1063/5.0010732.

[33] Hua Huang, Xin Xing, and Edmond Chow. H2Pack: High-performance $\mathcal{H}^2$ matrix package for kernel matrices using the proxy point method. *ACM Transactions on Mathematical Software*, 47(1):1–29, 2020. doi: 10.1145/3412850.

[34] Christian Møller and Milton S. Plesset. Note on an approximation treatment for many-electron systems. *Physical Review*, 46(7):618–622, 1934.

[35] Sigbjørn Sæbø and Peter Pulay. Local treatment of electron correlation. *Annual Review of Physical Chemistry*, 44:213–236, 1993.

[36] Martin Schütz and Hans-Joachim Werner. Low-order scaling local electron correlation methods. iv. linear scaling local mp2. *Journal of Chemical Physics*, 114(15):661–681, 2001.

[37] Christof Hättig. Optimization of auxiliary basis sets for ri-mp2 and ricc2 calculations: Core–valence and quintuple-ζ basis sets for h to ar and qzvpp basis sets for li to kr. *Physical Chemistry Chemical Physics*, 7(1): 59–66, 2005.

[38] Florian Weigend, Ansgar Köhn, and Christoph Hättig. Efficient use of the correlation consistent basis sets in resolution of the identity mp2 calculations. *Journal of Chemical Physics*, 116(8):3175–3183, 2002.

[39] Philippe Y. Ayala and Gustavo E. Scuseria. Linear scaling perturbation theory with localized molecular orbitals and density fitting. *Journal of Chemical Physics*, 110(15):3660–3671, 1999.

[40] Kenneth O. Berard, Hongji Gao, Alexander Teplukhin, Xiangmin Jiao, and Benjamin G. Levine. Efficient and scalable wave function compression using corner hierarchical matrices. *The Journal of Chemical Physics*, 161 (20):204106, 2024. doi: 10.1063/5.0231409.

[41] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. part i: Introduction to H-matrices. *Computing*, 62(2):89–108, 1999.

[42] Hongji Gao, Xiangmin Jiao, and Benjamin G. Levine. Accelerating correlated wave function calculations with hierarchical matrix compression of the two-electron integrals. *The Journal of Chemical Physics*, 163(13): 134107, 2025. doi: 10.1063/5.0286521.

[43] W Kohn. Density functional theory for systems of very many atoms. *International Journal of Quantum Chemistry*, 56(4):229–232, 1995.

[44] Walter Kohn. Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters*, 76(17):3168, 1996.

[45] Dietrich Braess and Wolfgang Hackbusch. Approximation of 1/x by exponential sums in $[1, \infty)$. *IMA Journal of Numerical Analysis*, 25(4): 685–697, 2005.

[46] Akio Takatsuka, Seiichiro Ten-No, and Wolfgang Hackbusch. Minimax approximation for the decomposition of energy denominators in laplace-transformed møller–plesset perturbation theories. *The Journal of Chemical Physics*, 129(4), 2008.

[47] Justus A. Calvin, Chong Peng, Varun Rishi, Ashutosh Kumar, and Edward F. Valeev. Many-body quantum chemistry on massively parallel computers. *Chemical Reviews*, 121(3):1203–1231, FEB 10 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.0c00006.

[48] DE Bernholdt and RJ Harrison. Large-scale correlated electronic structure calculations: The ri-mp2 method on parallel computers. *Chemical Physics Letters*, 250(5-6):477–484, MAR 8 1996. ISSN 0009-2614. doi: 10.1016/0009-2614(96)00054-1.

[49] GD Fletcher, MW Schmidt, and MS Gordon. Developments in parallel electronic structure theory. In I Prigogine and SA Rice, editors, *Advances in Chemical Physics, Vol 110*, volume 110 of *Advances in Chemical Physics*, pages 267–294. John Wiley & Sons, Inc., 1999. ISBN 0-471-33180-5. doi: 10.1002/9780470141694.ch4.

[50] C Hättig, A Hellweg, and A Köhn. Distributed memory parallel implementation of energies and gradients for second-order moller-plesset perturbation theory with the resolution-of-the-identity approximation. *Physical Chemistry Chemical Physics*, 8(10):1159–1169, MAR 14 2006. ISSN 1463-9076. doi: 10.1039/b515355g.

[51] Ivan S. Ufimtsev and Todd J. Martinez. Quantum chemistry on graphical processing units.: 1.: Strategies for two-electron integral evaluation. *Journal of Chemical Theory and Computation*, 4(2):222–231, FEB 2008. ISSN 1549-9618. doi: 10.1021/ct700268q.

[52] B. S. Fales and B. G. Levine. Nanoscale multireference quantum chemistry: Full configuration interaction on graphical processing units. *Journal of Chemical Theory and Computation*, 11(10):4708–4716, 2015.

[53] Edmond Chow, Xing Liu, Mikhail Smelyanskiy, and Jeff R Hammond. Parallel scalability of hartree–fock calculations. *The Journal of Chemical Physics*, 142(10):104103, 2015.

[54] Hans-Joachim Werner, Gerald Knizia, Christine Krause, Max Schwilk, and Mark Dornbach. Scalable electron correlation methods i.: Pno-lmp2 with linear scaling in the molecular size and near-inverse-linear scaling in the number of processors. *Journal of Chemical Theory and Computation*, 11(2):484–507, FEB 2015. ISSN 1549-9618. doi: 10.1021/ct500725e.

[55] Chong Peng, Justus A. Calvin, Fabijan Pavosevic, Jinmei Zhang, and Edward F. Valeev. Massively parallel implementation of explicitly correlated

coupled-cluster singles and doubles using tiledarray framework. *Journal of Physical Chemistry*, 120(51):10231–10244, DEC 29 2016. ISSN 1089-5639. doi: 10.1021/acs.jpca.6b10150.

[56] Karol Kowalski, Raymond Bair, Nicholas P. Bauman, Jeffery S. Boschen, Eric J. Bylaska, Jeff Daily, Wibe A. de Jong, Thom Dunning, Jr., Niranjan Govind, Robert J. Harrison, Murat Keceli, Kristopher Keipert, Sriram Krishnamoorthy, Suraj Kumar, Erdal Mutlu, Bruce Palmer, Ajay Panyala, Bo Peng, Ryan M. Richard, T. P. Straatsma, Peter Sushko, Edward F. Valeev, Marat Valiev, Hubertus J. J. van Dam, Jonathan M. Waldrop, David B. Williams-Young, Chao Yang, Marcin Zalewski, and Theresa L. Windus. From nwchem to nwchemex: Evolving with the computational chemistry landscape. *Chemical Reviews*, 121(8):4962–4998, APR 28 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.0c00998.

[57] Hang Hu, Shiv Upadhyay, Lixin Lu, Andrew J. Jenkins, Tianyuan Zhang, Agam Shayit, Stefan Knecht, and Xiaosong Li. Small tensor product distributed active space (stp-das) framework for relativistic and non-relativistic multiconfiguration calculations: Scaling from 10¡sup¿9¡/sup¿ on a laptop to 10¡sup¿12¡/sup¿ determinants on a supercomputer. *Chemical Physics Reviews*, 5(4):041404, DEC 2024. doi: 10.1063/5.0227122.

[58] Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/. Accessed: 2025-06-19.

[59] Hans-Joachim Werner, Peter J Knowles, Gerald Knizia, Frederick R Manby, and Martin Schütz. Molpro: a general-purpose quantum chemistry program package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2(2):242–253, 2012.

[60] H.-J. Werner, P. J. Knowles, P. Celani, W. Györffy, A. Hesselmann, D. Kats, G. Knizia, A. Köhn, T. Korona, D. Kreplin, R. Lindh, Q. Ma, F. R. Manby, A. Mitrushenkov, G. Rauhut, M. Schütz, K. R. Shamasundar, T. B. Adler, R. D. Amos, S. J. Bennie, A. Bernhardsson, A. Berning, J. A. Black, P. J. Bygrave, R. Cimiraglia, D. L. Cooper, D. Coughtrie, M. J. O. Deegan, A. J. Dobbyn, K. Doll, M. Dornbach, F. Eckert, S. Erfort, E. Goll, C. Hampel, G. Hetzer, J. G. Hill, M. Hodges, T. Hrenar, G. Jansen, C. Köppl, C. Kollmar, S. J. R. Lee, Y. Liu, A. W. Lloyd, R. A. Mata, A. J. May, B. Mussard, S. J. McNicholas, W. Meyer, T. F. Miller III, M. E. Mura, A. Nicklass, D. P. O'Neill, P. Palmieri, D. Peng, K. A. Peterson, K. Pflüger, R. Pitzer, I. Polyak, M. Reiher, J. O. Richardson, J. B. Robinson, B. Schröder, M. Schwilk, T. Shiozaki, M. Sibaev, H. Stoll,

A. J. Stone, R. Tarroni, T. Thorsteinsson, J. Toulouse, M. Wang, M. Welborn, and B. Ziegler. Molpro, version , a package of ab initio programs, 2020. see https://molpro.net/.

[61] Hans-Joachim Werner, Frederick R Manby, and Peter J Knowles. Fast linear scaling second-order møller-plesset perturbation theory (mp2) using local and density fitting approximations. *The Journal of Chemical Physics*, 118(18):8149–8160, 2003.

[62] Chenchen Song and Todd J Martínez. Atomic orbital-based SOS-MP2 with tensor hypercontraction. II. local tensor hypercontraction. *The Journal of Chemical Physics*, 146(3):034104, 2017.