Tensor-Structured Techniques for Electronic Structure Calculations

by

Chih-Chuen Lin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering and Scientific Computing)
in the University of Michigan
2022

Doctoral Committee:

Professor Vikram Gavini, Chair
Professor Krishnakumar Garikipati
Professor Shravan Veerapaneni
Associate Professor Paul Zimmerman

Chih-Chuen Lin

iancclin@umich.edu

ORCID iD: 0000-0002-5339-1805

致父與亡母

To my father and late mother.

ACKNOWLEDGMENTS

I would like to take this chance to express my gratitude to all people who have supported me in the past six years. First of all, my PhD advisor, Professor Vikram Gavini. I still remember the first time we met at his office in the EECS building where he gave me a list of useful kickoff readings and suggestions. He gives extreme freedom and extensive supports to his students and has always been providing valuable insights in every discussion. Other than research, he has been a caring mentor who guided me through all the anxieties and stresses during the doctoral study. He made me not only a more mature researcher but also a more mature person.

I thank Professor Krishna Garikipati, Professor Shravan Veerapaneni, and Professor Paul Zimmerman for their generosity in serving on my dissertation committee. I also thank my current and previous labmates Nelson, Bikash, Sambit, Paavai, Vishal, Phani, Krishnendu, and Nathan for their supports and help, especially, Phani's help with setting up various aspects of the tensor-structured algorithm, Bikash's help with proofreading and generating Hartree-Fock orbitals on the tensor-structured mesh, Vishal's help with the code to convert an unstructured mesh to a tensor-structured mesh in DFT-FE, and Sambit's help with modifying DFT-FE code to compute and visualize Kohn-Sham orbitals with higher residual norm tolerance. You guys made this dissertation more complete and my lab life more enjoyable. I also thank Khoi from the Chemistry department for his generous help in proofreading and providing FCI calculations which partly contribute to the last part of this thesis.

I thank all my friends in Taiwan and the US, Dowdow, Yun-Cheng, Magic Roy,

Man-Chun, Dr. Wei-Chung Wen, Dr. Wei-Chu Chuang, Dr. Dian-Ru Li, Hsun-Wei, Dr. Yu-Wei Hwang, Chung-Shuo, Yu-Hsuan, Kuan-Po, Te-Yuh, Kevin, Dr. Michael Chen, Kakaen, and many others I might miss here. It has been a long journey. Your companionship made this journey much more joyous. I also thank Professor Tung-Yu Wu and Yun-Chu Lin for helping me set up my life in the US on the very first day of my arrival. A special thank goes to my previous roommate Gavin for those (literally) countless conscious or unconscious nights we have been through together.

I would also like to extend my acknowledgment with sincere gratitude to my aunt and uncle Tsao. I am grateful for each of your visits and for the caring packages. You made my PhD life fully fed not only intellectually but also physically.

Last but the most important, to my beloved father, late mother, and baby brother. Were it not for your endless supports and unconditional love, I could never make it.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Electronic structure calculations have been one of the most successful scientific fields in the past 50 years. Density functional theory (DFT) with its nice balance between accuracy and efficiency has now become the standard technique for many materials research. On the other hand, the size of systems that wavefunction-based methods can handle has also been improved thanks to the recent developments in both computation power and more advanced methodologies. However, these methods are still suffering from the curse of dimensionality. The cubic scaling of a typical DFT calculation restricts the handleable system size of DFT to only a few thousand electrons. To this end, tensor-structured techniques provide a route to constructing a reduced-order algorithm and hence are promising to improve the computational efficiency. In this dissertation work, we aim at exploring the applications of the tensor-structured techniques on different aspects of electronic structure calculations.

In the first part of this work, we present a tensor-structured algorithm for efficient large-scale DFT calculations by constructing a Tucker tensor basis that is adapted to the Kohn–Sham Hamiltonian and localized in real-space. The proposed approach uses an additive separable approximation to the Kohn–Sham Hamiltonian and an L1 localization technique to generate the 1-D localized functions that constitute the Tucker tensor basis. Numerical results show that the resulting Tucker tensor basis exhibits exponential convergence in the ground-state energy with increasing Tucker rank. Further, the proposed tensor-structured algorithm demonstrated sub-quadratic scaling with system-size for both systems with and without a gap, and involving many thousands of atoms. This reduced-order scaling has also resulted in the proposed

approach outperforming plane-wave DFT implementation for systems beyond 2000 electrons.

In the second part of this work, we present an acceleration scheme for the proposed tensor-structured algorithm. In the proposed algorithm, the discrete Kohn-Sham problem is solved using the Chebyshev filtering subspace iteration method that relies on matrix-matrix multiplications of a sparse symmetric Hamiltonian matrix and a dense wavefunction matrix, expressed in the localized Tucker tensor basis. These matrix-matrix multiplication operations, which constitute the most computationally intensive step of the solution procedure, are GPU accelerated providing 8-fold GPU-CPU speedup for these operations on the largest systems studied. The computational performance of the GPU accelerated code is presented using benchmark studies on aluminum nano-particles and silicon quantum dots with system sizes ranging up to 7,000 atoms.

Finally, we present a computation kernel using tensor-structured techniques to evaluate the one-electron and two-electron integrals, which are the central quantities to construct the Hamiltonian matrix of a full configuration interaction (FCI) calculation, for any molecular orbitals projected on a tensor-structured finite-element mesh. The proposed computation kernel is used to compute the integrals using Hartree-Fock molecular orbitals, which is subsequently used to perform an FCI calculation. In our numerical study, the FCI energy using integrals from the proposed computation kernel is confirmed to be consistent with a standard FCI calculation using the same Hartree-Fock orbitals. The proposed computation kernel provides a useful tool to investigate the behavior of Kohn-Sham orbitals as a basis for FCI calculations in the future, which could be a route to construct a reduced-order basis for an FCI calculation to reduce the computational cost and enable FCI calculation for the previously computationally inaccessible systems.

CHAPTER I

Introduction

Electronic structure calculation is one of the most successful fields in scientific research in the past fifty years. It has since been providing many insights into the quantum mechanical properties of various materials and has been shown to be incredibly useful. The ubiquitous application of electronic structure calculation methods includes studying previously unknown material properties with electronic scale resolution (e.g. semiconductor band structures [1], DNA charge transport [2], metallic dislocations [3], structural properties and phase transition of metals [4]), generating *ab initio* force field for molecular dynamics type calculations [5], understanding of Lithium-ion batteries for new battery design [6], building a materials genome database [7] or developing generic algorithm [8] for new materials discovery [9], and even investigating the crystal structure of iron in the Earth's core [10].

Density functional theory (DFT) [11, 12], owing to the great balance it provides between accuracy and computational efficiency, has emerged as the workhorse of electronic structure calculations [13]. In 2021 alone, there have been more than 46,000 publications on google scholar searches associated with density functional theory. DFT reduces the Schrödinger equation involving the many-body wavefunction in $3N_e$ spatial coordinates (N_e denoting the number of electrons), to an equivalent problem of non-interacting electrons in a mean-field that is dependent on the electron density—a

variable in only 3 spatial coordinates, thus substantially reducing the computational complexity. While DFT is exact in principle, the many-body quantum mechanical interactions are encapsulated in the exchange-correlation (XC) functional whose form is unknown, and approximate models are used to model the XC functional. The development of increasingly accurate XC functionals is an active area of research [14, 15, 16, 17, 18, 19].

Despite the great reduction in the computational complexity in comparison to directly solving Schrödinger equation and the wide adoption of DFT for electronic structure calculations, the computational complexity of DFT calculations—conventionally, $O(MN_e^2)$, where M is the number of the basis functions required to achieve desired chemical accuracy, and is usually proportional to the number of electrons in the system (N_e)—still limits typical DFT calculations to just a few hundred atoms. Thus, to improve the computational efficiency of DFT calculations and enable accurate DFT calculations on large-scale systems, it is highly desirable to develop computational methods that can provide systematic convergence and are scalable to large number of MPI tasks, yet with a small basis set. The plane-wave basis, which is the most widely used basis in DFT calculations [20, 21, 22, 23, 24], provides systematic convergence, and is well suited for periodic calculations. However, the global nature of the plane-wave basis limits the parallel scalability, and its uniform spatial resolution makes it inefficient for non-periodic systems, such as isolated molecules or clusters. Among the real-space basis sets, the finite-element basis has been demonstrated to be highly scalable [25, 26, 3]—with parallel scalability demonstrated on $\sim 200,000$ MPI tasks. However, the number of basis functions required to achieve chemical accuracy is typically much higher than the plane-wave basis. On the other hand, while atomic orbital type basis functions [27, 28, 29, 30] are very efficient—typically involving only a few tens of basis functions per atom—systematic convergence is often a concern, especially in metallic systems. Further, the global nature of the basis functions can

limit the parallel scalability of calculations.

In an attempt to enable DFT calculations on large-scale systems that are critical to understanding many aspects of complex materials phenomena, many efforts over the past three decades have focused on developing reduced-order scaling algorithms for electronic structure calculations [31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]. These approaches have either relied on a localized representation of the single-electron wavefunctions (such as Wannier functions [33]) or the exponential decay of the density-matrix in real-space, and have been demonstrated to provide close to linear-scaling complexity for materials with a gap. However, they have not been widely successful for metallic systems (without a gap) either due to the errors resulting from realizing locality in the wavefunctions in real-space or due to the higher prefactors that make these approaches computationally more expensive than the traditional cubic-scaling algorithms for system-sizes of interest. In the first part of this dissertation work, we present an alternative direction by using tensor-structured ideas to achieve systematically convergent and efficient DFT calculations that exhibit sub-quadratic scaling for systems with and without a gap over system-sizes spanning many thousands of atoms.

Recent progress in using tensor-structured techniques for electronic structure calculations has provided a path forward for developing a reduced-order basis that is systematically improvable, efficient, and exhibits good parallel scalability. In particular, an analysis of various molecules has revealed that the electronic structure, in particular the electron density, admits a low-rank Tucker and canonical decomposition [42, 42, 43]. Further, *a posteriori* results have shown that the rank required to approximate the electronic density is only weakly dependent on the system size [44]. Based on these observations, a tensor-structured basis was proposed for systematically convergent and efficient large-scale DFT calculations [45]. The main ideas included constructing an additive separable approximation of the Kohn-Sham Hamiltonian,

and using the eigenbasis of this approximate Hamiltonian—which has a Tucker tensor format—as a reduced-order basis for DFT calculations. Importantly, being the eigenbasis of a Hermitian operator, the resulting Tucker tensor basis provides systematic convergence. Further, being adapted to the Kohn-Sham Hamiltonian, it was demonstrated to be a more efficient basis than the plane-wave basis, requiring fewer basis functions than the plane-wave basis to achieve similar accuracy. However, the global nature of the Tucker tensor basis resulted in a dense Hamiltonian matrix, which limited the accessible system sizes and parallel scalability of the method.

In order to alleviate this limitation, we proposed an L_1 localization approach to construct a localized Tucker tensor basis as the first part of this dissertation work [46]. In the proposed L_1 localization approach, L_1 localization is used to overcome the aforementioned drawbacks, and we demonstrate systematically convergent, efficient, and reduced-order scaling large-scale DFT calculations using tensor-structured techniques. The L_1 localization constructs a series of 1-D functions that are localized yet closely approximate the function space of interest. The 1-D localized functions that are a close approximation to the eigensubspace of a suitably constructed additive separable approximation of the Kohn-Sham Hamiltonian are used to generate the localized Tucker tensor basis for the DFT problem. The locality of the Tucker tensor basis results in a sparse discrete Kohn-Sham Hamiltonian matrix, which is exploited in the solution of the Kohn-Sham equations using the Chebyshev filtering subspace iteration scheme. The sparsity of the Kohn-Sham Hamiltonian matrix represented in the localized Tucker tensor basis improves both the computational efficiency and the memory footprint. Further, as will be demonstrated, the proposed approach has enabled sub-quadratic scaling DFT calculations on large-scale systems involving many thousands of atoms. The approach is generic and treats both systems with and without a gap on an equal footing. Importantly, this translates to substantial speed-ups over Quantum Espresso, a widely used state-of-the-art plane-wave DFT code [22, 23],

with speed-ups of $\sim$ 8-fold for metallic nano-particles containing $\sim 2,000$ atoms.

In the second part of this dissertation work, we improved the most computationally expensive part of the proposed tensor-structured algorithm using GPU acceleration. During the course of solving a Kohn-Sham equation, we noticed that the solution of Kohn-Sham equations in the localized Tucker tensor basis involves many operations that are amenable to acceleration using graphics processing units (GPU). In this work, we optimize various parts of the tensor-structured algorithm using GPUs, and provide the code base for conducting large-scale DFT calculations using a localized Tucker tensor basis. In particular, we optimize various computationally intensive kernels using CUDA library: (i) the matrix-matrix multiplication between the Kohn-Sham Hamiltonian in the localized Tucker tensor basis and the wavefunction matrix expressed in this basis, which appear in the Chebyshev filtering procedure to compute the occupied subspace of the Kohn-Sham Hamiltonian; (ii) the solution of the Kohn-Sham equations by projecting the problem onto the Chebyshev filtered subspace. Our numerical study shows that the implementation substantially accelerates the Chebyshev filtering step—the most time-consuming part in a many-core CPU-based calculation—by $\sim 7\times$ and substantially reduces the wall-times for DFT calculations. Further, we demonstrate the capability of conducting large-scale DFT calculations, with systems as large as $\sim 7,000$ atoms, on GPUs efficiently.

After the exploration of the application of tensor-structured techniques on providing a reduced-order basis for density functional theory, we shift our focus to the application of tensor-structured techniques on other domains of electronic structure calculations. In contrast to density functional theory, wavefunction based methods provide a hierarchical way for achieving higher accuracy at the price of exponentially growing computational cost with respect to the level of required accuracy [47]. It is thus desirable to have reduced-order molecular orbitals for the wavefunction based methods. To this end, Kohn-Sham orbitals, which intrinsically carry electron correla-

tion information, are postulated to be a good alternative to commonly used Hartree-Fock molecular orbitals, which do not have any electron correlation information at all [48, 49, 50]. However, the applicability of Kohn-Sham orbitals on a full configuration interaction (FCI) calculation still remains unknown due to the high computational cost of FCI. To look closer into this issue, we first developed a computation kernel based on tensor-structured techniques for generating one-electron and two-electron integrals, the central components for constructing matrix for configuration interaction (CI) calculations, from molecular orbitals projected on a tensor-structured finite-element mesh. The computed integrals can subsequently be used to perform an FCI calculation, which can be done using incremental FCI (iFCI) routine [51, 52] in QChem [27]. In this preliminary work, we used a hydrogen molecule with Hartree-Fock orbitals as a benchmark calculation. We have confirmed that the FCI energies of the benchmark system computed using the integrals from the developed computation kernel and from the resolution of identity (RI) approximation [53, 54], which is the subroutine for computing the integrals in a standard FCI calculation, are consistent and the difference in energy per atom is within 0.2 mHa/atom. We note that the proposed computation kernel provides a flexible framework to compute the one-electron and two-electron integrals from any molecular orbitals projected on a tensor-structured finite-element mesh. This step is the key to further studying the efficacy of Kohn-Sham orbitals as a reduced-order basis for FCI calculations in our future work.

The remainder of this thesis is structured as follows. Chapter II reviews the theoretical aspects of density functional theory and the Kohn-Sham formulation. Chapter III introduces the various aspects of tensor decomposition and defines the tensor notations and nomenclatures used in later chapters. Chapter IV discusses the tensor-structured algorithm which generates a reduced-order Tucker tensor basis for Kohn-Sham DFT calculations. The benchmark calculations on the convergence

property and the performance of the proposed algorithm are also presented in the same chapter. Chapter V discusses the GPU acceleration procedure for the proposed tensor-structured algorithm and provides the performance analysis of the accelerated code. Chapter VI gives an introduction on the wavefunction based methods and proposed a computation kernel to generate one-electron and two-electron integrals for any molecular orbitals projected on a tensor-structured finite-element mesh using tensor-structured techniques. Finally, Chapter VII summarizes this dissertation work and provides the future direction to pursue.

CHAPTER II

Density Functional Theory

The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation. – P.A.M. Dirac, 1929 [55]

Almost a century after P.A.M. Dirac made his remarkable quote, people are still in quest of *a practical approximation which can lead to an explanation of the main features of complex atomic systems without too much computation.* The Schrödinger equation, by its form, suffers from the combinatorial increase of dimension. A system soon becomes computationally intractable only after few atoms. Many methods have since been developed to address the issue raised by Dirac. One of the most promising method is Density Functional Theory, which reduces the number of variables from the astronomical $3N_e$, where N_e is the number of electrons in the system, associated with the wavefunction to only three spatial coordinates associated with the electron density. In this chapter, we will give an overview of the theoretical background of the

density functional theory. The remainder of this chapter is organized as follows. The many-body Schrödinger equation is introduced in Section 2.1. Then the Hatree-Fock method is summarized in Section 2.2. Finally, we introduce the density functional theory in Section 2.3. The introduction to DFT will be focusing on (1) Hohenberg-Kohn theorem in Section 2.3.1 to show the existence and the uniqueness of the density functional and (2) the theoretical formulation of the Kohn-Sham method for DFT in Section 2.3.2.

2.1 The many-body Shcrödinger equation

Schrödinger equation, since originally proposed by Erwin Shcrödigner [56] in 1926, has further unraveled the mysteries of our universe. It describes the fundamental properties of quantum mechanical systems within an elegant and concise expression. In the context of non-relativistic quantum mechanics, the time-independent many-body Schrödinger equation is given by

$$\mathcal{H}\Psi = \epsilon\Psi, \tag{2.1}$$

where $\mathcal{H}$ denotes the many-body Hamiltonian operator, ϵ denotes the eigenvalue corresponding to the many-body wavefunction Ψ. Using atomic units [57], the many-body Hamiltonian operator is given by

$$\begin{aligned}\mathcal{H} = &-\frac{1}{2}\sum_{i=1}^{N_e}\nabla_i^2 - \sum_{I=1}^{N_a}\frac{1}{2M_I}\nabla_I^2 - \sum_{i=1}^{N_e}\sum_{I=1}^{N_a}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \\ &+ \sum_{i=1}^{N_e}\sum_{j<i}^{N_e}\frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|} + \sum_{I}^{N_a}\sum_{J<I}^{N_a}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}\end{aligned}, \tag{2.2}$$

where N_e denotes the number of electrons in the system, N_a denotes the number of nuclei in the system, ∇_i^2, ∇_I^2 denote the Laplacian operator of electrons and atoms coordinates respectively, M_I, $\mathbf{R}_I$, and Z_I denote the mass, coordinates, and charge

of the I-th atom respectively, $\mathbf{r}_i$ denotes the position of i electron in the system. We emphasize that as electrons are identical and indistinguishable particles, we use i electron and avoid using i-th electron as a compromise to *not specify* a specific electron in the system. As the time-independent schrödinger equation does not vary with time, it is sometimes also referred to as stationary state Schrödinger equation. The physical meaning of each term in the Hamiltonian operator in Eq. 2.2 is interpreted as:
the operator corresponding to the kinetic energy of electrons

$$\mathcal{T}_e = -\frac{1}{2}\sum_{i=1}^{N_e}\nabla_i^2, \tag{2.3}$$

the operator corresponding to the kinetic energy of nuclei

$$\mathcal{T}_N = -\sum_{I=1}^{N_a}\frac{1}{2M_I}\nabla_I^2, \tag{2.4}$$

the operator corresponding to the attractive electrostatic interaction between electrons and nuclei

$$\mathcal{V}_{eN} = -\sum_{i=1}^{N_e}\sum_{I=1}^{N_a}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}, \tag{2.5}$$

the operator corresponding to the repulsive electrostatic interaction between electrons

$$\mathcal{V}_{ee} = \sum_{i=1}^{N_e}\sum_{j<i}^{N_e}\frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|}, \tag{2.6}$$

and the operator corresponding to the repulsive electrostatic interaction between nuclei

$$\mathcal{V}_{NN} = \sum_{I}^{N_a}\sum_{J<I}^{N_a}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}. \tag{2.7}$$

With Eq. 2.3-2.7, a shorthand representation of the Hamiltonian operator is

$$\mathcal{H} = \mathcal{T}_e + \mathcal{T}_N + \mathcal{V}_{eN} + \mathcal{V}_{ee} + \mathcal{V}_{NN}. \tag{2.8}$$

As can be easily concluded from the expression, $\mathcal{T}_e$ and $\mathcal{V}_{ee}$ are universal for all identical electrons and remain invariant for different systems. The combination of the two terms is hence also called the universal operator.

In Eq. 2.1, the many-body wavefunction is a function taking $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_{N_a}$ and $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_e})$, where $\mathbf{x}_i = \mathbf{x}_i(\mathbf{r}_i, s_i)$ as input variable

$$\Psi = \Psi(\mathbf{x}, \mathbf{R}) \,, \tag{2.9}$$

where s_i is the spin of i electron. In quantum mechanics, wavefunction itself is not an observable and does not carry any physical interpretation. Only the square of wavefunction

$$|\Psi(\mathbf{x}, \mathbf{R})|^2 \tag{2.10}$$

represents the probability density of finding N_e electrons *simultaneously* in the system. As aforementioned, electrons are identical particles hence indistinguishable, this leads to an important statement that interchanging electron coordinates $\mathbf{x}_i$ and $\mathbf{x}_j$ should not change the probability density, or in mathematical expression

$$|\Psi(\mathbf{x}_1...\mathbf{x}_i...\mathbf{x}_j...\mathbf{x}_{N_e}, \mathbf{R})|^2 = |\Psi(\mathbf{x}_1...\mathbf{x}_j...\mathbf{x}_i...\mathbf{x}_{N_e}, \mathbf{R})|^2 \,. \tag{2.11}$$

Meanwhile, as electrons are fermions, it has been shown that the spin of an electron is half integral value $s_i = \pm\frac{1}{2}$ and the Pauli exclusion principle mandates anti-symmetry to the wavefunction, i.e. interchanging coordinates of two electrons leads to sign change of the wavefunction

$$\Psi(\mathbf{x}_1...\mathbf{x}_i...\mathbf{x}_j...\mathbf{x}_{N_e}, \mathbf{R}) = -\Psi(\mathbf{x}_1...\mathbf{x}_j...\mathbf{x}_i...\mathbf{x}_{N_e}, \mathbf{R}). \tag{2.12}$$

Although this expression seems to imply that electrons are distinguishable, as the wavefunction is not an observable, electrons are still identical as long as the observ-

able (i.e. probability density in Eq. 2.11) remains unchanged under the exchange of coordinates of two electrons.

We note that nuclei are much heavier (e.g. $\sim 1,800$ times heavier for a hydrogen atom, $\sim 30,000$ times heavier for an oxygen atom) and hence move much slower than electrons. Electrons can thus be viewed as moving on a potential surface of *frozen* nuclei. As a result, the kinetic contribution from nuclei $\mathcal{T}_N$ can be neglected, the electrostatic interaction between nuclei $\mathcal{V}_{NN}$ is simply a constant, and the $\mathbf{R}$ variables in $\mathcal{V}_{eN}$ can be viewed as parameters. This approximation, commonly known as Born-Oppenheimer approximation or adiabatic approximation[1], removes the nuclei degrees of freedom and significantly reduces the complexity of the system. Under Born-Oppenheimer approximation, the full many-body Hamiltonian in Eq. 2.2 can be recast as many-body electron Hamiltonian

$$\mathcal{H}^{ele} = -\frac{1}{2}\sum_{i=1}^{N_e}\nabla_i^2 - \sum_{i=1}^{N_e}\sum_{I=1}^{N_a}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{i=1}^{N_e}\sum_{j<i}^{N_e}\frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|} + \mathcal{V}_{NN}. \tag{2.13}$$

The full many-body wavefunction is also recast as many-body electron wavefunction parametrized with the nuclei coordinates $\mathbf{R}$ as

$$\Psi^{ele}\Big|_{\mathbf{R}_{I=1\ldots N_a}} = \Psi^{ele}(\mathbf{x};\mathbf{R}) := \Psi^{ele}(\mathbf{x}). \tag{2.14}$$

Typically, we choose to normalize the wavefunction to unity

$$\int\cdots\int |\Psi^{ele}|^2 d\mathbf{x}_1 d\mathbf{x}_2 ... d\mathbf{x}_{N_e} = 1, \tag{2.15}$$

where $d\mathbf{x}_i = d\mathbf{r}_i ds_i$.

The Courant min-max theorem states that the eigenvalue problem Eq. 2.13 can

[1]Born-Oppenheimer approximation is also called adiabatic approximation because the idea to treat much slower nuclei as a negligible perturbation is similar to the adiabatic process in thermal physics. See Ch. 10 in Griffith's book for a more complete discussion [58].

also be recast as a variational problem

$$\begin{aligned}\epsilon_0^{ele} &= \inf_{\Psi^{ele'}\in\mathcal{S}} \int\cdots\int \Psi^{ele'*}\mathcal{H}^{ele}\Psi^{ele'} d\mathbf{x}_1 d\mathbf{x}_2 ... d\mathbf{x}_{N_e} = \inf_{\Psi^{ele'}\in\mathcal{S}} \left\langle \Psi^{ele'} \middle| \mathcal{H}^{ele} \middle| \Psi^{ele'} \right\rangle, \\ \mathcal{S} &= \{\Psi | \langle \Psi \mid \Psi \rangle = 1, \Psi \text{ is anti-symmetric}\},\end{aligned} \tag{2.16}$$

where $\Psi^{ele'}$ is a trial wavefunction, ϵ_0^{ele} is the ground-state energy of the system, $\left\langle \Psi^{ele'} \middle| \mathcal{H}^{ele} \middle| \Psi^{ele'} \right\rangle$ uses Dirac bra-ket notation [59, 60]. For the sake of brevity, we will drop the superscript *ele*. All quantities mentioned hereafter are presumed to be under Born-Oppenheimer approximation and nuclei degrees of freedom are treated as parameters unless otherwise mentioned.

As can be easily seen from the variables of the wavefunction, the computational cost of solving the Schrödinger equation suffers from *the curse of dimensionality*. The dimension of the wavefunction immediately becomes intractable even for a very small system. For example, a three atoms water molecule has only 3 nuclei and 10 electrons. Even if the spatial coordinates are just coarsely discretized into 10 grid points in each direction (assuming the spin-orbital is neglected), it still takes $10^{3\times3+3\times10}$ data points, which is roughly $\mathcal{O}(10^{24})$ petabytes, to simply store the wavefunction. Even with the Born-Oppenheimer approximation, which eliminates the nuclear degrees of freedom, the storage cost is still $\mathcal{O}(10^{15})$ petabytes. Furthermore, the cost of storing the discretized Hamiltonian is quadratic with respect to the size of the discretized wavefunction. A strategy to recast this calculation of astronomical complexity to a more tractable problem is therefore desirable.

2.2 Hartree-Fock method

The Hartree-Fock approximation is one of the early attempts to solve the many-electrons Schrödinger equation and is still being widely used in many applications. The main idea of the Hartree-Fock method is to search the wavefunction that gives the

lowest energy in a subset of $\mathcal{S}$ in Eq. 2.16. The Hartree-Fock method assumes that the wavefunction can be written as a linear combination of the product of single-electron wavefunctions $\xi_i(\mathbf{x}_j)$ and has the form

$$\Psi_0 \approx \Psi_0^{\mathrm{HF}} = \frac{1}{\sqrt{N_e!}} \begin{vmatrix} \xi_1(\mathbf{x}_1) & \xi_2(\mathbf{x}_1) & \cdots & \xi_{N_e}(\mathbf{x}_1) \\ \xi_1(\mathbf{x}_2) & \xi_2(\mathbf{x}_2) & \cdots & \xi_{N_e}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1(\mathbf{x}_{N_e}) & \xi_2(\mathbf{x}_{N_e}) & \cdots & \xi_{N_e}(\mathbf{x}_{N_e}) \end{vmatrix} \tag{2.17}$$

known as a Slater determinant, where the single-electron wavefunction $\xi_i(\mathbf{x}) = \phi_i(\mathbf{r})\sigma_i(s)$, where $\phi_i(\mathbf{r})$ is the spatial part and $\sigma(s)$ is the spin part of the single-electron wavefunction. Given the spin function $\sigma(s)$ defined as $\alpha(s)$ for up-spin or $\beta(s)$ for down-spin, the integration of the spin functions is defined by

$$\begin{cases} \int ds \alpha^*(s)\alpha(s) = \int ds \beta^*(s)\beta(s) = 1\,, \\ \int ds \alpha^*(s)\beta(s) = \int ds \beta^*(s)\alpha(s) = 0\,. \end{cases} \tag{2.18}$$

The prefactor $\frac{1}{\sqrt{N_e!}}$ is the normalization factor for the Slater determinant and it can be easily shown that the Slater determinant naturally satisfies the anti-symmetry requirement of a wavefunction. Thus, $\Psi_0^{\mathrm{HF}} \in \mathcal{S} = \{\Psi |\, \langle \Psi \,|\, \Psi \rangle = 1, \Psi \text{ is anti-symmetric}\}$ in Eq. 2.16 holds. We also introduce a commonly used shorthand notation that will

be useful for later chapters for the Slater determinant Eq. 2.17 as

$$|\xi_1\xi_2\cdots\xi_{N_e}\rangle := \frac{1}{\sqrt{N_e!}}\begin{vmatrix} \xi_1(\mathbf{x}_1) & \xi_2(\mathbf{x}_1) & \cdots & \xi_{N_e}(\mathbf{x}_1) \\ \xi_1(\mathbf{x}_2) & \xi_2(\mathbf{x}_2) & \cdots & \xi_{N_e}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1(\mathbf{x}_{N_e}) & \xi_2(\mathbf{x}_{N_e}) & \cdots & \xi_{N_e}(\mathbf{x}_{N_e}) \end{vmatrix}. \tag{2.19}$$

The Hartree-Fock energy computed using a (trial) wavefunction with the form Eq. 2.17 is

$$\begin{aligned} E^{\mathrm{HF}} &= \left\langle \Psi^{\mathrm{HF}} \right| \mathcal{H} \left| \Psi^{\mathrm{HF}} \right\rangle \\ &= \sum_{i}^{N_e} (i|h|i) + \frac{1}{2}\sum_{i}^{N_e}\sum_{j}^{N_e}\Big\{(ii|jj) - (ij|ji)\Big\}, \end{aligned} \tag{2.20}$$

where, using chemists' notation,

$$(i|h|i) := \int \xi_i^*(\mathbf{x})\Big\{ -\frac{1}{2}\nabla^2 - \sum_{I}^{N_a}\frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}\Big\}\xi_i(\mathbf{x})d\mathbf{x} \tag{2.21}$$

is the one-electron term comprising the contribution from the kinetic term and the electrostatic contribution from nuclei,

$$(ii|jj) := \int\int \frac{\xi_i^*(\mathbf{x})\xi_i(\mathbf{x})\xi_j^*(\mathbf{x}')\xi_j(\mathbf{x}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{x}d\mathbf{x}' \tag{2.22}$$

is referred to as the Coulomb integral, and

$$(ij|ji) := \int\int \frac{\xi_i^*(\mathbf{x})\xi_j(\mathbf{x})\xi_j^*(\mathbf{x}')\xi_i(\mathbf{x}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{x}d\mathbf{x}', \tag{2.23}$$

is referred to as the exchange integral.

From Eq. 2.20, it is obvious that E^{HF} is a functional of the single-electron wavefunctions. The Lagrangian of the Hartree-Fock expression subject to the normality

constraint of the orthogonal single-electron wavefunctions set ξ_i can thus be written as

$$\mathcal{L}[\{\xi_i\}] = E^{\mathrm{HF}}[\{\xi_i\}] - \sum_i^{N_e} \epsilon_i(\langle \xi_i \,|\, \xi_i \rangle - 1). \tag{2.24}$$

We note that while we only show the case that single-electron wavefunctions are chosen to be orthogonal for the sake of simplicity for presentation, this is not a necessary requirement. The derivation can be easily generalized to treat non-orthogonality. More details in the derivation can be found in Ch 3.2 in [61] or refer to Ch. 2.3.2 for a similar derivation for the Kohn-Sham equation. Now, by setting the first variation of Eq. 2.24 to zero, we arrive at the Hartree-Fock equation

$$f\xi_i = \epsilon_i \xi_i \,. \tag{2.25}$$

The Fock operator f in Eq. 2.25 is defined by

$$f\xi_i(\mathbf{x}) = \Big(h(\mathbf{x}) + v^{\mathrm{HF}}(\mathbf{x})\Big)\xi_i(\mathbf{x}), \tag{2.26}$$

where the Hartree-Fock potential v^{HF} is defined by

$$v^{\mathrm{HF}}(\mathbf{x})\xi_i(\mathbf{x}) := \sum_j^{N_e} \Big(\mathcal{J}_j(\mathbf{x}) - \mathcal{K}_j(\mathbf{x})\Big)\xi_i(\mathbf{x}) \,. \tag{2.27}$$

On the right-hand side of Eq. 2.27, the first term corresponds to the Coulomb operator $\mathcal{J}_j(\mathbf{x})$ acting on $\xi_i(\mathbf{x})$

$$\mathcal{J}_j(\mathbf{x})\xi_i(\mathbf{x}) := \int \frac{\xi_j^*(\mathbf{x}')\xi_j(\mathbf{x}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{x}' \xi_i(\mathbf{x}) \,, \tag{2.28}$$

and the second term corresponds to the exchange operator $\mathcal{K}_j(\mathbf{x})$ acting on $\xi_i(\mathbf{x}_i)$

$$\mathcal{K}_j(\mathbf{x})\xi_i(\mathbf{x}) := \int \frac{\xi_j^*(\mathbf{x}')\xi_i(\mathbf{x}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{x}' \xi_j(\mathbf{x}) \,. \tag{2.29}$$

We note that the non-physical self-interacting term in Eq. 2.27 is automatically canceled out in the expression when $i = j$. The physical interpretation of the Hartree-Fock potential is an average repulsive potential experienced by electron i induced by all other $N_e - 1$ electrons. The Hartree-Fock equation can hence be viewed as a system with multiple non-interacting electrons in a mean-field generated from averaging contribution from all other electrons in the system.

While the Hartree-Fock method provides a good starting point to calculate the ground-state energy of the many-electrons Schrödinger equation, it approximates the original many-electrons system as an alternative system with non-interacting electrons in a mean-field with only Coulomb and exchange contribution from other electrons. This approximation nicely handles the quantum mechanical effect due to the Pauli exclusion principle that does not allow electrons with parallel-spin to occupy the same orbital. However, the Hartree-Fock method does not account for the correlational effect coming from instantaneous repulsion between electrons. As a result, electrons in the Hartree-Fock picture tend to be *too close* to each other and give larger (more positive) repulsive energy[2]. Subsequently, the Hartree-Fock ground-state energy ϵ_0^{HF} (also called the Hartree-Fock limit) is always larger than the exact ground-state energy ϵ_0^{exact}.

Many wavefunction based methods have since been developed to improve the Hartree-Fock approximation by introducing the electron correlation, where the electron correlation is accounted for with multi-determinant wavefunctions. These methods can generally be categorized into three main branches—*configuration interaction method* (CI), *many-body perturbation theory* (MBPT), and *coupled-cluster method*

[2]This electronic correlation is also called *dynamic correlation*. Total electronic correlation also includes the *non-dynamic correlation* (or *static correlation*). The non-dynamic correlation comes from the fact that a single Slater determinant fails to approximate the true ground-state properly. This leads to further discussion and is out of the scope of this introductory piece. We refer to pp.14-18 in [62] for a brief introduction, [63] for a discussion of this topic in the context of density functional theory, and [61] for more complete and pedagogical discussion (more in the context of wavefunction based methods.)

(CC)—and called *post Hartree-Fock methods*, *wavefunction (based) methods*, or *electron correlation methods* [47]. On the other hand, density functional theory provides a completely different way of approaching this issue. Thanks to the remarkable contribution of Hohenberg and Kohn, it has been shown that the ground-state energy is uniquely determined by the ground-state electron density of the system. The powerful and elegant statement builds up the foundation of density functional theory, which has now been ubiquitous in the field of electronic structure calculation.

2.3 Desnity functional theory

Density functional theory has been a workhorse in electronic structure calculation since the remarkable paper of Pierre Hohenberg and Walter Kohn appeared on Physical Review in 1964 [11]. The paper lays the cornerstone of the theoretical background of density functional theory. Later on, Kohn and Sham proposed a strategy to treat the many-electrons interacting system as a fictitious system consisting of many non-interacting electrons in an effective mean-field [12]. In the remainder of this section, we will introduce some basic theoretical background of density functional theory.

2.3.1 Hohenberg-Kohn theorem

The Hohenberg-Kohn theorem proves the existence and the uniqueness of the density functional [11]. The proof of the Hohenberg-Kohn theorem is provided here for the sake of completeness, and we refer to the original paper [11] and other textbooks [64, 62] for a more complete discussion on this topic. The theorem is twofold.

2.3.1.1 The first Hohenberg-Kohn theorem

The Hohenberg-Kohn first theorem gives a mathematically sound background that electron density $\rho(\mathbf{r})$ is the basic variable for a many-body electronic structure calculation, where given the electron density operator (physically can be interpreted as

summing over each event that electron i appears at $\mathbf{r}$)

$$\varrho(\mathbf{r}) = \sum_{i}^{N_e} \delta(\mathbf{r} - \mathbf{r}_i) \tag{2.30}$$

and recalling the relation $\mathbf{x} = (\mathbf{r}, s)$, $d\mathbf{x} = d\mathbf{r}ds$ and

$$\int \delta(\mathbf{r} - \mathbf{r}_i) f(\mathbf{x}_i) d\mathbf{x}_i = \int f(\mathbf{r}, s_i) ds_i \,, \tag{2.31}$$

the electron density is derived as

$$\begin{aligned}
\rho(\mathbf{r}) =& \langle \Psi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) | \, \varrho(\mathbf{r}) \, | \Psi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) \rangle \\
=& \langle \Psi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) | \sum_{i}^{N_e} \delta(\mathbf{r} - \mathbf{r}_i) \, | \Psi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) \rangle \\
=& \int \Psi^*(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) \sum_{i}^{N_e} \delta(\mathbf{r} - \mathbf{r}_i) \Psi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e}) d\mathbf{x}_1 d\mathbf{x}_2 \cdots d\mathbf{x}_{N_e} \\
=& N_e \int |\Psi(\mathbf{r}, \mathbf{x}_2, \cdots, \mathbf{x}_{N_e})|^2 ds_1 d\mathbf{x}_2 \cdots d\mathbf{x}_{N_e} \,.
\end{aligned} \tag{2.32}$$

The theorem states that, for a system with non-degenerate ground-state, the ground-state electron density $\rho(\mathbf{r})$, uniquely determines the external potential $v_{\text{ext}}(\mathbf{r})$ to within a constant, and in turn uniquely determines the Hamiltonian. As a result, *the full many body particle ground-state is a unique functional of* $\rho(\mathbf{r})$ (direct quote from the Hohenberg-Kohn 1964 paper [11]).

The proof of the statement uses *reductio ad absurdum*. To begin with, we assume that we have two different external potentials $v_{\text{ext}}^{(1)}$ and $v_{\text{ext}}^{(2)}$ differing by more than a constant shift giving the same electron density $\rho(\mathbf{r})$. This implies that there are two different Hamiltonian constructed from these two external potentials (the nucleus-nucleus repulsion terms are dropped because they are simply constants under Born-

Oppenheimer approximation), *viz.*

$$\begin{aligned}\mathcal{H}^{(1)} &= \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{\text{ext}}^{(1)}\,,\\ \mathcal{H}^{(2)} &= \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{\text{ext}}^{(2)}\,.\end{aligned} \tag{2.33}$$

Now, we have two ground-state wavefunction $\Psi^{(1)}$ and $\Psi^{(2)}$ correspond to the two Hamiltonian $\mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$. Similarly, $\mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$ also correspond to different ground-state energy $\epsilon_0^{(1)}$ and $\epsilon_0^{(2)}$.

By the virtue of variational principle and the definition of the ground-state energy, we have

$$\begin{aligned}\epsilon_0^{(1)} &= \left\langle \Psi^{(1)} \right| \mathcal{H}^{(1)} \left| \Psi^{(1)} \right\rangle \\ &< \left\langle \Psi^{(2)} \right| \mathcal{H}^{(1)} \left| \Psi^{(2)} \right\rangle = \left\langle \Psi^{(2)} \right| \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle + \left\langle \Psi^{(2)} \right| \mathcal{H}^{(1)} - \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle .\end{aligned}$$

With Eq. 2.33, we have

$$\left\langle \Psi^{(2)} \right| \mathcal{H}^{(1)} - \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle = \left\langle \Psi^{(2)} \right| \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{\text{ext}}^{(1)} - \mathcal{T} - \mathcal{V}_{ee} - \mathcal{V}_{\text{ext}}^{(2)} \left| \Psi^{(2)} \right\rangle , \tag{2.34}$$

where, recall Eq. 2.32, the expectation value of the external energy operator $\mathcal{V}_{\text{ext}} = \sum_i^{N_e} v_{\text{ext}}(\mathbf{r}_i)$ can also be written as

$$\begin{aligned}\langle \Psi | \, \mathcal{V}_{\text{ext}} \, | \Psi \rangle &= \int \langle \Psi | \sum_i^{N_e} v_{\text{ext}}(\mathbf{r}) \delta(\mathbf{r} - \mathbf{r}_i) \, | \Psi \rangle \, d\mathbf{r} \\ &= \int v_{\text{ext}}(\mathbf{r}) \, \langle \Psi | \sum_i^{N_e} \delta(\mathbf{r} - \mathbf{r}_i) \, | \Psi \rangle \, d\mathbf{r} = \int \rho(\mathbf{r}) v_{\text{ext}}(\mathbf{r}) d\mathbf{r} \,.\end{aligned} \tag{2.35}$$

Recall that the two external potentials are giving the same electron density $\rho(\mathbf{r})$, by cancelling the universal terms in Eq. 2.34, we obtain

$$\left\langle \Psi^{(2)} \right| \mathcal{H}^{(1)} - \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle = \left\langle \Psi^{(2)} \right| \mathcal{V}_{\text{ext}}^{(1)} - \mathcal{V}_{\text{ext}}^{(2)} \left| \Psi^{(2)} \right\rangle = \int \rho(\mathbf{r}) \left(v_{\text{ext}}^{(1)} - v_{\text{ext}}^{(2)} \right) d\mathbf{r} \,,$$

and hence with $\left\langle \Psi^{(2)} \right| \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle = \epsilon_0^{(2)}$, we can write

$$\epsilon_0^{(1)} < \epsilon_0^{(2)} + \int \rho(\mathbf{r}) \left(v_{\text{ext}}^{(1)} - v_{\text{ext}}^{(2)} \right) d\mathbf{r} \,. \tag{2.36}$$

Similarly, by computing

$$\epsilon_0^{(2)} = \left\langle \Psi^{(2)} \right| \mathcal{H}^{(2)} \left| \Psi^{(2)} \right\rangle < \left\langle \Psi^{(1)} \right| \mathcal{H}^{(2)} \left| \Psi^{(1)} \right\rangle \,,$$

we have

$$\epsilon_0^{(2)} < \epsilon_0^{(1)} + \int \rho(\mathbf{r}) \left(v_{\text{ext}}^{(2)} - v_{\text{ext}}^{(1)} \right) d\mathbf{r} \,. \tag{2.37}$$

Adding the two equations (Eq. 2.36+Eq. 2.37), we have the desired contradiction

$$\epsilon_0^{(1)} + \epsilon_0^{(2)} < \epsilon_0^{(2)} + \epsilon_0^{(1)} \implies 0 < 0 \,. \qquad \blacksquare$$

The proof shows that the ground-state electron density uniquely defined the external potential up to a constant and subsequently specifies the ground-state energy. This implies that the ground-state energy (and its individual components) can be written as the functional of the ground-state electron density, namely

$$\epsilon_0[\rho_0] = T[\rho_0] + E_{ee}[\rho_0] + E_{\text{ext}}[\rho_0] \,, \tag{2.38}$$

where T is the kinetic energy, E_{ee} is the electron-electron repulsion energy, and E_{ext} is the external energy determined by the system and has the form

$$E_{\text{ext}}[\rho_0] = \int \rho_0(\mathbf{r}) v_{\text{ext}} d\mathbf{r} \,. \tag{2.39}$$

We note that on the right-hand side of Eq. 2.38, only the last term is dependent on the actual system. The other two terms are invariant to any systems These two terms

are also called the Hohenberg-Kohn functional

$$F_{HK}[\rho_0] = T[\rho_0] + E_{ee}[\rho_0] = \langle \Psi_0 | \, \mathcal{T}_e + \mathcal{V}_{ee} \, | \Psi_0 \rangle \ , \tag{2.40}$$

which encapsulates all the non-classical and electronic classical effects in a density functional theory calculation. As simple as the expression above may appear to be, the exact form of the Hohenberg-Kohn functional still remains unknown. More than half-century after Hohenberg and Kohn's 1964 paper, finding the exact energy functional of the ground-state electron density is still the holy grail in the study of density functional theory.

2.3.1.2 The second Hohenberg-Kohn theorem

In the first Hohenberg-Kohn theorem, it has been proven that the ground-state electron density uniquely determines the external potential hence establishing the functional relation to the ground-state energy. However, the first theorem does not say anything about how the energy varies when the electron density is not the exact ground-state electron density. This gap is then filled with the second theorem.

The second Hohenberg-Kohn theorem reads that for a trial electron density $\rho(\mathbf{r})$, which satisfies the condition: $\rho(\mathbf{r}) \geq 0$, $\int \rho(\mathbf{r}) d\mathbf{r} = N_e$ and the density is associated with an anti-symmetric wavefunction and some external potential v_{ext} (v-representable), the true ground-state electron density gives the lowest energy, namely

$$\epsilon_0[\rho_0] \leq \epsilon[\rho] = T[\rho] + E_{ee}[\rho] + E_{\text{ext}}[\rho] \, . \tag{2.41}$$

The proof of the second statement is rather simple. In the first theorem, it has been shown that the ground-state electron density gives unique external potential. Meanwhile, as the trial ρ is assumed to be v-representable, the external potential thus can be written as a functional of the trial electron density, namely, $v_{\text{ext}} = v_{\text{ext}}[\rho]$,

and similarly, the corresponding Hamiltonian and the ground-state wavefunction $\Psi = \Psi[\rho]$. Now, we substitute the ground-state wavefunction Ψ corresponding to the trial electron density $\rho(\mathbf{r})$ into the variational principle in Eq. 2.16,

$$\begin{aligned}\langle \Psi[\rho] | \mathcal{H} | \Psi[\rho] \rangle &= \langle \Psi[\rho] | \mathcal{T}_e + \mathcal{V}_{ee} | \Psi[\rho] \rangle + \int \rho(\mathbf{r}) v_{\text{ext}} d\mathbf{r} \\ &= T[\rho] + E_{ee}[\rho] + \int \rho(\mathbf{r}) v_{\text{ext}} d\mathbf{r} = \epsilon[\rho] \geq \epsilon_0[\rho_0] = \langle \Psi_0 | \mathcal{H} | \Psi_0 \rangle \ . \end{aligned} \qquad \blacksquare$$

This shows that any electron density that is not the true ground-state density always gives an upper-bound to the exact ground-state energy. We note that in this section, the proof assumes that the ground-state energy is non-degenerate and the electron density is v-representable. However, the condition for an electron density to be v-representable has not yet been established. To address this issue, the density functional theory can be re-formulated to require only n-representable electron density. In contrast to v-representability, n-representability is a weaker condition that only requires the electron density to be constructed from an anti-symmetric wavefunction. The mathematical formulation for an electron density to be n-representable has been shown to be [65]

$$\rho(\mathbf{r}) \geq 0 \, , \quad \int \rho(\mathbf{r}) d\mathbf{r} = N_e \, , \quad \int |\nabla \sqrt{\rho(\mathbf{r})}|^2 d\mathbf{r} < \infty \, . \tag{2.42}$$

In the subsequent section, we introduce an alternative way of formulating the Hohenberg-Kohn theorem.

2.3.1.3 The Levy constrained-search approach

Recall that the original many-body Schrödinger equation can be recast as the optimization problem in Eq. 2.16, where Eq. 2.16 searches admissible wavefunctions over all space and pick up the one that gives the lowest energy. Similarly, the Levy constrained-search approach [66, 67] first minimizes the energy over all wavefunctions

yielding a specific electron density (see Eq 2.32). Then it searches over all electron densities obtained from the first step for the electron density that gives the lowest energy. That is

$$\begin{aligned}\epsilon_0 &= \min_{\rho \to N_e} \left(\min_{\Psi \to \rho} \langle \Psi | \, \mathcal{T} + \mathcal{V}_{ee} + \mathcal{V}_{\text{ext}} \, | \Psi \rangle \right) \\ &= \min_{\rho \to N_e} \left(\min_{\Psi \to \rho} \langle \Psi | \, \mathcal{T} + \mathcal{V}_{ee} \, | \Psi \rangle + \int \rho(\mathbf{r}) v_{\text{ext}} d\mathbf{r} \right).\end{aligned} \tag{2.43}$$

Therein, the second equation comes from the fact that the external potential is a functional of the electron density. As the first minimization searches over all wavefunctions that give the same electron density, the $\int \rho(\mathbf{r}) v_{\text{ext}} d\mathbf{r}$ term remains the same for all admissible wavefunction in the first minimization. Further, by introducing the universal functional

$$F[\rho] = \min_{\Psi \to \rho} \langle \Psi | \, \mathcal{T}_e + \mathcal{V}_{ee} \, | \Psi \rangle \ , \tag{2.44}$$

Eq. 2.43 can be re-formulated as

$$\epsilon_0 = \min_{\rho \to N_e} \left(F[\rho] + \int \rho(\mathbf{r}) v_{\text{ext}} d\mathbf{r} \right). \tag{2.45}$$

Again, we see that the true ground-state electron density delivers the ground-state energy. We remark that the universal functional $F[\rho]$ is defined for all n-representable electron densities, therefore the v-representable restriction in the original Hohenberg-Kohn theorem is loosened. In the case that the electron density is v-representable, the universal function $F[\rho]$ coincides with the Hohenberg-Kohn functional $F_{HK}[\rho]$ in Eq. 2.40. Furthermore, if the ground-state is degenerate, the Levy constrained-search only picks one wavefunction (in the first minimization step) associated with the ground-state electron density ρ_0. The non-degenerate premise in the original Hohenberg-Kohn theorem is hence also lifted.

2.3.2 Kohn-Sham method

While the Hohenberg-Kohn theorem has proven that the ground-state energy is a functional of the ground-state electron density, the exact form of the functional is still unknown. Some of the attempts have been devoted to constructing a functional of electron density for kinetic energy. A few of those formulations appear even earlier than the introduction of density functional theory, e.g. Thomas-Fermi model for uniform electron gas [68, 69]. These attempts have later been categorized as orbital-free methods for density functional theory and it is still an open research area [70, 71, 72, 73]. While the orbital-free approach is very fast and is theoretically capable of achieving linear scaling with system size, the accuracy of the kinetic energy is regretfully low. The error in the kinetic energy is of the same order as the total ground-state energy. To address this issue, Kohn and Sham come up with a clever way to introduce a fictitious system comprised of non-interacting electrons moving in an effective mean-field [12]. Their method computes the kinetic energy to good accuracy and approximates the non-classical part, whose error is much smaller than the kinetic part, using the *exchange-correlation energy* term. The Kohn-Sham method has since been of central importance in the density functional calculation. We will now provide the formulation of the Kohn-Sham method.

Similar to the idea in the Hartree-Fock method shown in Eq. 2.20, Kohn and Sham introduce Kohn-Sham orbitals Ψ_i^{KS}[3] for a fictitious system with non-interacting electrons moving in an effective mean-field to compute the kinetic energy and reformulate the breakdown of the ground-state energy as

$$E[\rho] = T_s[\rho] + E_{\mathrm{H}}[\rho] + E_{\mathrm{ext}}[\rho] + E_{\mathrm{XC}}[\rho]. \tag{2.46}$$

[3]In the previous section where we discussed the Hartree-Fock method, the orbital is called a single-electron wavefunction. These two terms will be used interchangeably hereafter.

Therein, the T_s defined as

$$T_s = -\frac{1}{2}\sum_{i}^{N_e}\left\langle \Psi_i^{KS}\right| \nabla^2 \left|\Psi_i^{KS}\right\rangle \,, \tag{2.47}$$

is the exact kinetic energy of the non-interacting system. We note that the Kohn-Sham orbitals appear explicitly in the formulation. This is because while T_s must be a functional of electron density, there is no simple formulation for it. $E_{\mathrm{H}}[\rho]$ denotes the classical Coulomb electrostatic interaction defined as

$$E_{\mathrm{H}}[\rho] = \int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' d\mathbf{r} \,, \tag{2.48}$$

which is the only part in Eq. 2.45 that people know the functional expression of electron density except for the already known external potential term. Finally, the mysterious exchange-correlation energy term that encapsulates all terms whose functionals are still unknown is defined as

$$E_{\mathrm{XC}}[\rho] = (T[\rho] - T_s[\rho]) + (E_{ee}[\rho] - E_{\mathrm{H}}[\rho]) \,, \tag{2.49}$$

where $T[\rho] - T_s[\rho]$ is the residual of the true kinetic energy and $E_{ee}[\rho] - E_{\mathrm{H}}[\rho]$ is the non-classical contribution to the electron-electron repulsion.

With the energy functional defined, we now seek to obtain N Kohn-Sham orbitals that give the lowest energy under the orthonormality constraint. To begin with, the Lagrangian of this optimization problem is

$$\mathcal{L}[\Psi_i^{KS}] = E[\rho] - \sum_{i}^{N}\sum_{j}^{N}\epsilon_{ij}\left(\int \Psi_i^{KS*}(\mathbf{x})\Psi_j^{KS}(\mathbf{x})d\mathbf{x} - \delta_{ij}\right). \tag{2.50}$$

Taking variation with Ψ_i^{KS}, Ψ_i^{KS*}, ϵ_{ij} respectively and setting them to zero, we have

$$\begin{cases} \Big(-\frac{1}{2}\nabla^2 + v_{\text{eff}} \Big) \Psi_i^{KS} = \sum_j^N \epsilon_{ij} \Psi_j^{KS} \\ \Big(-\frac{1}{2}\nabla^2 + v_{\text{eff}} \Big) \Psi_i^{KS*} = \sum_j^N \epsilon_{ji} \Psi_j^{KS*} \\ \int \Psi_i^{KS*}(\mathbf{x}) \Psi_j^{KS}(\mathbf{x}) d\mathbf{x} = \delta_{ij} \end{cases} . \tag{2.51}$$

Subtracting the complex conjugate of the second line from the first line in the above equation, we have

$$\sum_j^N (\epsilon_{ij} - \epsilon_{ji}^*) \Psi_j^{KS} = 0 \,. \tag{2.52}$$

As the Kohn-Sham orbitals are linearly independent, the matrix ϵ_{ij} is a Hermitian and hence can be diagonalized by a unitary transformation of the Kohn-Sham orbitals. Finally, we arrive at the Kohn-Sham equation

$$\left(-\frac{1}{2}\nabla^2 + v_{\text{eff}} \right) \Psi_i^{KS} = \epsilon_i \Psi_i^{KS} \,, \tag{2.53}$$

where v_{eff} is the effective potential defined as

$$v_{\text{eff}} = \frac{\delta E_{\text{H}}}{\delta \rho} + \frac{\delta E_{\text{ext}}}{\delta \rho} + \frac{\delta E_{\text{XC}}}{\delta \rho} \,. \tag{2.54}$$

Therein, the first term on the right-hand side denotes the Hartree potential $v_{\text{H}}[\rho]$ defined as

$$v_{\text{H}}[\rho] = \frac{\delta E_{\text{H}}}{\delta \rho} \,. \tag{2.55}$$

The second term is simply the external potential v_{ext}

$$v_{\text{ext}}[\rho] = \frac{\delta E_{\text{ext}}}{\delta \rho} \,. \tag{2.56}$$

The third term is called exchange-correlation potential defined directly as the func-

tional derivative of the exchange-correlation energy with respect to electron density

$$v_{\mathrm{XC}} = \frac{\delta E_{\mathrm{XC}}}{\delta \rho} \,. \tag{2.57}$$

We note that up to this point, the Kohn-Sham method is exact. In the Kohn-Sham method, if we have the true expression for exchange-correlation functional, the many-body problem can be solved exactly and many efforts have been devoted to that direction [18, 19, 74, 75]. The inexactness only comes into play when we approximate the functional expression of this exchange-correlation term.

In quest of a functional expression for the exchange-correlation term, the local density approximation (LDA) is a good starting point. The LDA model assumes that the exchange-correlation energy can be written as

$$E_{\mathrm{XC}}^{LDA}[\rho] = \int \rho(\mathbf{r}) \epsilon_{\mathrm{XC}}(\rho(\mathbf{r})) d\mathbf{r} \,, \tag{2.58}$$

where ϵ_{XC} is the exchange-correlation energy per particle of a uniform electron gas of electron density that comprises the exchange and the correlation part

$$\epsilon_{\mathrm{XC}}^{LDA}(\rho(\mathbf{r}) = \epsilon_{\mathrm{X}}^{LDA}(\rho(\mathbf{r})) + \epsilon_{\mathrm{C}}^{LDA}(\rho(\mathbf{r})) \,, \tag{2.59}$$

and the exchange-correlation potential is formulated as [64]

$$v_{\mathrm{XC}}^{LDA}(\rho(\mathbf{r})) = \frac{\delta E_{\mathrm{XC}}^{LDA}}{\delta \rho} = \epsilon_{\mathrm{XC}}^{LDA}(\rho(\mathbf{r})) + \rho(\mathbf{r}) \frac{\partial \epsilon_{\mathrm{XC}}(\rho(\mathbf{r}))}{\partial \rho(\mathbf{r})} \,. \tag{2.60}$$

In other words, instead of taking the electron density function over the whole space, i.e. $\epsilon_{\mathrm{XC}} = \epsilon_{\mathrm{XC}}[\rho]$, as a variable, the local density approximation takes only the local density value, i.e. $\epsilon_{\mathrm{XC}} = \epsilon_{\mathrm{XC}}(\rho(\mathbf{r}))$.

In the LDA model, the exchange term $\epsilon_{\mathrm{X}}^{LDA}$ is known to be the cubic root of the

electron density up to a constant prefactor. However, the exact form for the correlation term is not known yet. To tackle this, numerical quantum Monte Carlo simulations has been performed to obtain exchange-correlation energy for homogeneous electron gas with different electron density. The results are then parametrized to provide a mapping between the local electron density and the exchange-correlation terms [76]. Some efforts have also been applied to improve the parametrization [77, 78, 79].

Besides LDA, many other exchange-correlation functionals with different hierarchy are available. A generalized gradient approximation functional, e.g. PBE functional [80], takes both electron density ρ and the gradient of the electron density $\nabla\rho$ as variables, i.e. $E_{\mathrm{XC}}^{GGA} = E_{\mathrm{XC}}^{GGA}(\rho, \nabla\rho)$. A meta-GGA functional, e.g. M06L [81], takes ρ, $\nabla\rho$, and $\nabla^2\rho$ as variables, i.e. $E_{\mathrm{XC}}^{meta-GGA} = E_{\mathrm{XC}}^{meta-GGA}(\rho, \nabla\rho, \nabla^2\rho)$. A hybrid functional, e.g. B3LYP [82, 83, 84, 85], takes all variables that GGA takes plus the exact exchange information from the Hartree-Fock theory (cf. Sec. 2.2). This hierarchical structure of the exchange-correlation functional is usually termed as Jacob's ladder in DFT by John Perdew [86].

Finally, while a lot of aspects of density functional theory are omitted in this introductory piece, we will stop the discussion here. We refer to many well-written references for more information [62, 64, 87, 88]. To sum up, in this chapter, we briefly introduced the theoretical background of the density functional theory and the Kohn-Sham method. We will revisit the real-space formulation of the Kohn-Sham method that is relevant to this dissertation work in later chapters.

CHAPTER III

Tensor Decomposition

In this chapter, we introduce various types of tensor decomposition that emerge in the literature and put emphasis on Tucker tensor representation which constitutes the core of this dissertation work. The structure of this chapter is as follows. In Section 3.1, we give an overview of the definitions of basic tensor operations, applications of tensor techniques and different tensor formats as an introduction to the concept of tensor. In Section 3.2, we introduce the canonical decomposition of a tensor. In Section 3.3, we introduce Tucker tensor decomposition. Tucker tensor decomposition compresses a full tensor into a much smaller core tensor and multiple matrices. Various aspects of Tucker tensor decomposition and the corresponding algorithm, higher-order singular value decomposition that is used to obtain Tucker tensor representation in this dissertation work, will be presented. Finally, for the sake of completeness, we summarize and discuss some other tensor decompositions in the last section.

3.1 Nomenclature, definition, and introduction

A tensor is an algebraic object defining a multi-dimensional array. In the context of tensor analysis, a d-dimensional array is also referred to as a d-mode array, d-way array or a d-th order tensor. We first define a tensor product for $\mathbf{u}^j \in \mathbb{F}^{I_j}$, $1 \leq j \leq d$,

as

$$\left(\bigotimes_{j=1}^{d}\mathbf{u}^{j}\right)_{i_1 i_2 \ldots i_d} = \left(\mathbf{u}^{1}\otimes\mathbf{u}^{2}\otimes\ldots\otimes\mathbf{u}^{d}\right)_{i_1 i_2 \ldots i_d} = \prod_{j=1}^{d} u_{i_j}^{j}\,, \tag{3.1}$$

where $\mathbb{F}^{I_j}$, defined on the field $\mathbb{F}$, denotes a vector space on which $\mathbf{u}^j$ is defined, d denotes the dimension, I_j defines a finite index set $I_j = \left\{i_j | i_j = 1, 2, ..., n_j\right\}$, $i_j \in \mathbb{N}$. A d-way tensor A is defined in a tensor space as

$$A \in \bigotimes_{j=1}^{d}\mathbb{F}^{I_j} = \text{span}\left\{\bigotimes_{j=1}^{d}\mathbf{u}^{j} : \mathbf{u}^{j} \in \mathbb{F}^{I_j}\right\}, \tag{3.2}$$

where the tensor elements of A are specified as

$$(A)_{i_1,\ldots,i_j,\ldots,i_d} = a_{i_1\ldots i_j\ldots i_d}\,. \tag{3.3}$$

Other notation

$$A \in \mathbb{F}^{I_1\times I_2\times\ldots\times I_d} \tag{3.4}$$

also appears in literatures and can be proved to be equivalent to $\bigotimes_{j=1}^{d}\mathbb{F}^{I_j}$ by taking all linear combination of $\bigotimes_{j=1}^{d}\mathbf{u}^{j}, \mathbf{u}^{j} \in \mathbb{F}^{I_j}$ [89]. The two notations will be used interchangeably hereafter. We note that for the sake of simplicity and relevance to this work, we will restrict the discussion to a real-valued tensor unless otherwise mentioned, namely

$$A \in \bigotimes_{j=1}^{d}\mathbb{R}^{I_j} = \text{span}\left\{\bigotimes_{j=1}^{d}\mathbf{u}^{j} : \mathbf{u}^{j} \in \mathbb{R}^{I_j}\right\}. \tag{3.5}$$

It is also worth noting that a zeroth-order tensor is simply a scalar. A first-order tensor, namely a one-way or one-dimensional array, defines a vector, and a second-order tensor defines a matrix. Other than these lower-order tensors, a multi-dimensional array whose dimension is higher than three(i.e., $d \geq 3$) is conventionally referred to as a higher-order tensor [90].

For a higher-order tensor, some useful nomenclature, notations, and operations that will be used later are defined as follows.

Fiber:

A fiber in j-dimension is a vector $\mathbf{v} \in \mathbb{R}^{I_j}$ extracted from a tensor with all other dimensions but j fixed. For example, for a three-dimensional tensor $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, a fiber in mode-2 with the indices of the other two modes fixed to $\bar{i_1}$ and $\bar{i_3}$ is a vector $\mathbf{v}$ defined as $(\mathbf{v})_{i_2} = a_{\bar{i_1} i_2 \bar{i_3}}$.

Slice:

A slice in j,k-dimension is a matrix $\mathbf{V} \in \mathbb{R}^{I_j \times I_k}$ extracted from a tensor with all other dimensions but j and k fixed. For example, for a three-dimensional tensor $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, a slice in mode-1 and mode-2 with the index of mode-3 fixed to $\bar{i_3}$ is a matrix $\mathbf{V}$ defined as $(\mathbf{V})_{i_1 i_2} = a_{i_1 i_2 \bar{i_3}}$.

Norm:

The norm of a tensor A is a higher-order analogue to the Frobenius norm of a matrix. The definition goes as

$$\|A\| := \sqrt{\sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \dots \sum_{i_d}^{I_d} a_{i_1 \dots i_j \dots i_d}^2}. \tag{3.6}$$

Hadamard product:

The Hadamard product simply takes the product of the tensor elements element-wisely. The definition of the Hadamard product for two tensors of the same size is

$$(A * B)_{i_1 i_2 \dots i_d} = a_{i_1 i_2 \dots i_d} b_{i_1 i_2 \dots i_d}\,. \tag{3.7}$$

Rank-one tensor:

A rank-one tensor is also called an elementary tensor [89] because the linear combination of such tensors spans the tensor space as shown in Eq. 3.2. A rank-one tensor is defined as

$$A = \bigotimes_{j=1}^{d} \mathbf{u}^j = \mathbf{u}^1 \otimes \mathbf{u}^2 \otimes \dots \otimes \mathbf{u}^d, \tag{3.8}$$

where $\mathbf{u}^j \in \mathbb{R}^{I_j}$, or in terms of its elements

$$a_{i_1 i_2 \ldots i_d} = u^1_{i_1} u^2_{i_2} \ldots u^d_{i_d} \,, \tag{3.9}$$

following the definition of tensor product in Eq. 3.1. A schematic of a rank-one tensor is presented in Fig. 3.1.

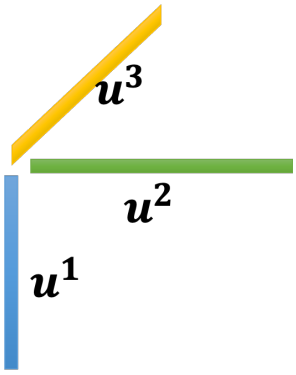


Figure 3.1: Schematic of a rank-one tensor.

Tensor matricization:

Tensor matricization is also called tensor unfolding [90]. j-mode tensor matricization maps the tensor element $(i_1, i_2, ..., i_d)$ to the matrix element (i_j, k). The transformation from the index of a tensor element to a matrix element is defined as

$$\begin{aligned} i_j &= i_j \,, \\ k &= 1 + \sum_{p=1, p\neq j}^{d} (i_p - 1) \prod_{q=1, q\neq j}^{p-1} I_q \,. \end{aligned} \tag{3.10}$$

We note that the j-mode matricization of tensor A is hereafter denoted as $\mathbf{A}_{(j)}$.

Tensor matrix multiplication on j-mode:

A tensor matrix multiplication on j-mode is also called j-mode product. Following Kolda and Bader's notation [90], the tensor matrix multiplication between a tensor

$A \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_j \times \ldots \times I_d}$ and a matrix $\mathbf{X} \in \mathbb{R}^{K \times I_j}$ is written as $A \times_j \mathbf{X}$ and defined as

$$(A \times_j \mathbf{X})_{i_1 i_2 \ldots i_{j-1} k i_{j+1} \ldots i_d} = \sum_{i_j=1}^{I_j} a_{i_1 i_2 \ldots i_j \ldots i_d} x_{i_j k}. \tag{3.11}$$

The operation can also be written as matrix-matrix multiplication in the form of matricized tensor as

$$(A \times_j \mathbf{X})_{(j)} = \mathbf{X}\mathbf{A}_{(j)}. \tag{3.12}$$

Tensor contraction:

Tensor contraction is also referred to as tensor-tensor product or tensor-tensor multiplication. It contracts (sums over) one or more given dimensions of two tensors. For instance, given two tensors $A \in \bigotimes_{j_A=1}^{5} \mathbb{R}^{I_{j_A}}$ and $B \in \bigotimes_{j_B=1}^{4} \mathbb{R}^{I_{j_B}}$, the tensor contraction of A and B along two given modes p and q is defined as

$$c_{abcde} = \sum_p \sum_q a_{abpcq} b_{pdeq}. \tag{3.13}$$

In fact, tensor matrix multiplication in Eq. 3.11 is simply a special case of tensor contraction on mode-j of a higher-order tensor and a matrix. As tensor contraction is of central importance in many applications, e.g. machine learning and wavefunction based methods in quantum chemistry, many computation kernels have thence been developed to facilitate tensor contraction operations or improve the computational efficiency of tensor contractions [91, 92, 93, 94, 95].

With the notations defined, we now turn our attention back to tensor decomposition. The decomposition of a second-order tensor (a matrix) is well-known as singular value decomposition. The generalization of tensor decomposition to a higher-order tensor was first introduced by Hitchcock in 1927 [96, 97]. In the first few decades after being proposed, the concept of tensor decomposition did not receive much attention. It is historically recognized that the application of tensor decomposition

was first introduced to studies in psychometrics by Tucker in the 1960s [98, 99, 100]. Soon after, it was introduced to the field of chemometrics by Appellof and Davidson in the 1980s [101]. Since then, higher-order tensor decomposition has received increasing attention in multiple fields. In the field of signal processing and machine learning, tensor decomposition is used for speech recognition, passive sensing, facial recognition, identification of people connections on social media [102, 103, 104]. A special issue on tensor decomposition for signal processing and machine learning was published by IEEE to highlight the wide applications and importance of tensor decomposition in that field [105]. In the field of data analysis, tensor decomposition is used for compressing high dimensional data whose size could easily exceed many terabytes (TBs) when the dimension of the tensor gets higher. An examination into the use of tensor decomposition on analysis of combustion direct numerical simulation data showed that tensor decomposition is capable of compressing data from the order of terabytes into a few gigabytes, which is handleable by a normal laptop. The data compression makes it possible to analyze and share the representation of the original data with reasonable accuracy [106]. This feature is even more useful in an exascale computing era, where the computed original tensor could be extremely large. The power of tensor decomposition on data analysis and compression also has significant applications in many other fields including biomedical imaging [107, 108, 109, 110], spectroscopy and microscopy [111, 112], environmental modeling [113], and many more. In the context of electronic structure calculations, using tensor decomposition to improve the computational efficiency is an active area of research. The capability of tensor decomposition to decompose a high-dimensional tensor as a linear combination of low-rank/low-dimensional tensor is promising and naturally suitable to tackle the lingering *curse of dimensionality* in electronic structure calculations [114, 115, 116]. Particularly, tensor decomposition has been used to compute the convolutional integrals emerging in the evaluation of electrostatic potentials and

two-electron-integrals in many different quantum mechanical methods [117, 42, 118]. Tensor decomposition techniques have also been used for investigating the property of electron density in a DFT calculation [44], designing an efficient basis for Kohn-Sham DFT calculations [45, 46], and many other applications in electronic structure calculations [115, 119, 120, 121, 122, 123, 124].

While many tensor decomposition techniques are available or under development, canonical decomposition and Tucker decomposition are generally considered as the higher-order analog of principal value analysis or matrix singular value decomposition for a higher-order tensor and therefore receives the most attention and has been of central importance in the study of tensor analysis and applications. Canonical decomposition decomposes a full d-dimensional tensor as a sum of multiple rank-one tensors. On the other hand, Tucker decomposition decomposes a full d-dimensional tensor as d matrices and a smaller tensor of the same dimension of the full tensor. Other than canonical decomposition and Tucker decomposition, some other tensor decomposition techniques are also receiving increasing attention.

Before getting into more details about tensor decomposition, we note that the term *tensor decomposition* is used loosely in this work. Strictly speaking, tensor representation and tensor decomposition have different definitions and can be distinguished rigorously [89]. By definition, tensor representation maps certain components to a d-way tensor. As a result, a tensor representation is not necessarily injective. On the other hand, tensor decomposition is going in the opposite direction. It injectively (essentially[1]) maps a d-way tensor into certain components. For instance, Tucker representation (cf. Ch. 3.3 and Eq. 3.23) maps a core tensor and d factor matrices to

[1]By *essentially* we mean that, by literal definition, it might not be a decomposition in some cases as some other properties could break the injectivity, but we still regard it as a decomposition. For instance, in the case of matrix eigenvalue decomposition, when there is k-fold degeneracy (meaning there are k same eigenvalues), the eigenvectors are not unique, hence *eigenvalue decomposition* is non-injective. Any unitary transformation of the k eigenvectors are still eigenvectors and span the same eigen-subspace of the matrix. By definition, it is not a decomposition, but we still (essentially) regard it as a decomposition in a looser sense.

a d-way tensor $A \in \bigotimes_{j=1}^{d} \mathbb{R}^{I_j}$

$$\text{(core tensor, factor matrices)} \xmapsto{\text{Tucker representation}} A, \tag{3.14}$$

while Tucker decomposition decomposes a d-way tensor as a core tensor and d factor matrices

$$A \xmapsto{\text{Tucker decomposition}} \text{(core tensor, factor matrices)}. \tag{3.15}$$

However, many literatures are loose on these two terms. For instance, while Hackbusch comments that *tensor train format* (cf. Ch. 3.4) is a more appropriate term over *tensor train decomposition* because tensor train format lacks uniqueness (i.e. no injectivity) [89], Oseledets used *tensor train format*, *tensor train decomposition*, *tensor train representation* loosely and somewhat interchangeably in his tensor train decomposition paper [125]. In this dissertation, we do not distinguish strictly between terms for the sake of ease of presentation. Tensor representation, tensor format, and tensor decomposition are used interchangeably throughout unless it is necessary to distinguish and otherwise mentioned. The precise meaning should be clear in the context.

In the remainder of this chapter, we provide a review of various aspects of canonical and Tucker decomposition followed by some comments on other tensor decomposition models and their applications in electronic structure calculations. For a more complete and comprehensive picture of tensor analysis, tensor decomposition, and applications of tensor analysis and methods in different fields, we refer to many nicely written books and review articles. Kolda *et al.*'s 2009 review paper is a good starting point [90]. It defines notations for commonly used tensor operations and thoroughly introduces the tensor decomposition models in the canonical and Tucker family. Smilde *et al.*'s 2004 book [126] has a longer discussion in tensor decomposition that is truncated in Kolda *et al.*'s paper due to limited space and also provides an

overview to the tensor decomposition application in chemometrics, one of the oldest applications of tensor decomposition. Hackbusch's 2012 book [89] has an extensive discussion on various aspects of tensor representation and builds up a more rigorous mathematical foundation, albeit a little advanced to read. An overview of applications of tensor-related methods in quantum mechanical and electronic structure calculations can be found in Khoromskij and Khoromskaia [127, 116]'s books.

3.2 Canonical decomposition

Canonical decomposition (CANDECOMP) was originally termed the polyadic form of a tensor when it was first discovered [96]. It was later on also referred to as parallel factors (PARAFAC) [128], CANDECOMP [129], CANDECOMP/PARAFAC [130], canonical polyadic decomposition (CPD) [131], or simply CP decomposition, which, as a compromise, can be interpreted as initials of CANDECOMP/PARAFAC or an abbreviation of canonical polyadic decomposition.

For a d-way tensor $A \in \bigotimes_{j=1}^{d} \mathbb{R}^{I_j}$, the canonical decomposition of A can be written as the sum of tensor product of rank-one components

$$A = \sum_{r=1}^{R} \bigotimes_{j=1}^{d} \mathbf{u}_r^j, \tag{3.16}$$

where R is the exact rank of A, $\mathbf{u}_r^j$ is the rank-one components corresponding to the j-mode. While $\mathbf{u}_r^j$ need not be normalized, canonical decomposition is usually written in the form that $\mathbf{u}_r^j$ is normalized

$$A = \sum_{r=1}^{R} \lambda_r \bigotimes_{j=1}^{d} \hat{\mathbf{u}}_r^j, \tag{3.17}$$

where $\hat{\mathbf{u}}_r^j$ is the normalized $\mathbf{u}_r^j$ and λ_r can be interpreted as the coefficient corresponding to r-th rank-one tensor $\bigotimes_{j=1}^{d} \hat{\mathbf{u}}_r^j$. The exact rank mentioned in Eq. 3.16 is the

rank such that the equality is held and should not be confused with the (canonical) rank in low-rank canonical decomposition that will be discussed later. The equality in Eq. 3.16 becomes an approximation for a low-rank approximation using canonical decomposition. It is noted that determining the exact rank of a higher-order tensor is non-trivial. Heretofore, there has not been a perfect algorithm [90]. It has even been proven that for a given three-dimensional tensor over any finite field $\mathbb{F}$, determining its exact rank is NP-complete. Furthermore, it is NP-hard for a three-dimensional tensor over $\mathbb{R}$ [132].

For better understanding, we give an example of a real-valued rank-R three-dimensional tensor $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ of size $I_1 \times I_2 \times I_3$ indexed by the set (i_1, i_2, i_3) as

$$(A)_{i_1,i_2,i_3} = a_{i_1 i_2 i_3} \,. \tag{3.18}$$

The canonical decomposition of A reads

$$A = \sum_{r=1}^{R} \mathbf{u}_r^1 \otimes \mathbf{u}_r^2 \otimes \mathbf{u}_r^3, \tag{3.19}$$

where $\mathbf{u}_r^j \in \mathbb{R}^{I_j}$ is rank-one components corresponding to the j-mode. A schematic of the canonical decomposition for a three-dimensional tensor is provided in Fig. 3.2.

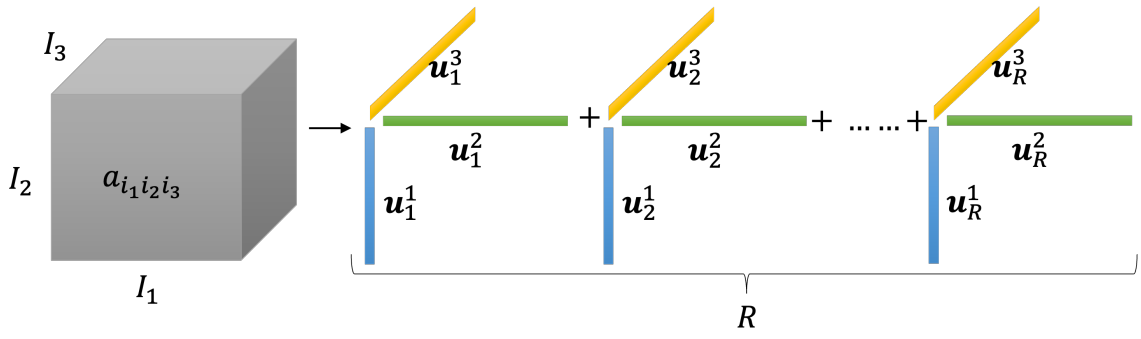


Figure 3.2: Schematic of Canonical decomposition.

Now we turn our attention to low-rank approximation of a tensor using canonical decomposition. In a rank-k low-rank approximation, the equality in Eq. 3.16 is loosen

to an approximation and the exact rank-R is reduced to rank-k as

$$A \approx \tilde{A} = \sum_{r=1}^{k} \bigotimes_{j=1}^{d} \mathbf{u}_r^j, \tag{3.20}$$

or in the case of a three-dimensional tensor

$$A \approx \tilde{A} = \sum_{r=1}^{k} \mathbf{u}_r^1 \otimes \mathbf{u}_r^2 \otimes \mathbf{u}_r^3. \tag{3.21}$$

The best rank-k approximation to a tensor minimizes the norm of the difference between the original tensor and the rank-k approximation

$$\tilde{A} = \arg\min_{A'} \|A - A'\|. \tag{3.22}$$

Different from the singular value decomposition of a matrix, the rank-one components of the best rank-k approximation do not necessarily come from the rank-one components of the previous rank-1 to rank-$(k-1)$ approximation. As a result, the best rank-k canonical decomposition cannot be solved incrementally. Instead, all rank-one components have to be solved simultaneously [133, 126].

In order to compute the best approximation to a tensor in the form of canonical decomposition with a given rank k, the alternating least square (ALS) algorithm is introduced and remains to be the most used algorithm [129, 128]. In the ALS algorithm, all dimensions are fixed but j-dimension, where $j = 1, ..., d$. Then the algorithm finds the factor matrix $\mathbf{U}^j \in \mathbb{R}^{I_j \times k}$, whose column vectors are the collection of all k rank-one components corresponding to j-dimension, that best recover the j-mode matricization of A for $j = 1 : d$ sequentially. The process is repeated in each iteration until the norm of the difference between A and A' is smaller than the given tolerance or the max iteration is reached. For further details, we point out to pp. 113-118 in Smilde *et al.* [126] for a more complete description on how to implement

the algorithm and pp. 262-267 in Hackbusch [89] for discussion in the mathematical properties of the ALS algorithm.

3.3 Tucker decomposition

Similar to canonical decomposition, Tucker decomposition has also appeared in literatures by many different names. Originally termed three-mode factor analysis or Tucker3 decomposition by Tucker and Levin [98, 99, 134] when it was first introduced. Later on, it was also called three-mode principal component analysis (3MPCA) [135], N-mode principal component analysis [136], higher-order singular value decomposition (HOSVD) [137], N-mode singular value decomposition [104].

Tucker tensor decomposition can be regarded as a higher-order generalization of the singular value decomposition or principal component analysis of a d-dimensional tensor. For a d-dimensional tensor, defined in Eq. 3.3, its exact Tucker tensor decomposition can be written as

$$A = \sum_{r_1=1}^{R_1^{\text{exact}}} \sum_{r_2=1}^{R_2^{\text{exact}}} \cdots \sum_{r_d=1}^{R_d^{\text{exact}}} \sigma_{r_1 r_2 \ldots r_d} \bigotimes_{j=1}^{d} \mathbf{u}_{r_j}^{j} \,, \tag{3.23}$$

if the Tucker rank R_j^{exact} recovers the original tensor A exactly, or the low-rank Tucker tensor decomposition which approximates the tensor A with the Tucker rank R_j,

$$A \approx \tilde{A} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_d=1}^{R_d} \sigma_{r_1 r_2 \ldots r_d} \bigotimes_{j=1}^{d} \mathbf{u}_{r_j}^{j} \,, \tag{3.24}$$

where $\sigma_{r_1 r_2 \ldots r_d} \in \bigotimes_{j=1}^{d} \mathbb{R}^{I_j}$ denotes the core tensor of the Tucker tensor decomposition. The core tensor can be seen as the coefficients of the corresponding rank-one components and is typically much more smaller than the original tensor in the case of a low-rank approximations (i.e., $r_j \ll R_j$.) $\bigotimes_{j=1}^{d} \mathbf{u}_{r_j}^{j}$ is a rank-one tensor and $\mathbf{u}_{r_j}^{j} \in \mathbb{R}^{I_j}$ is the r_j rank-one components corresponding to j-dimension. We note

that all R_j rank-one components corresponding to j-dimension comprise the mode-j factor matrix $\mathbf{U}^j$. For Tucker decomposition, the exact n-rank, denoted as $\text{rank}_n(A)$, is defined as the column rank of the matricization of the tensor A in n-dimension, that is,

$$R_n = \text{rank}_n(A) = \text{rank}(\mathbf{A}_{(n)}). \tag{3.25}$$

We note that, instead of using j as in previous sections, n is used here as the free index to specify a mode to follow the convention in the literature [138, 137, 90]. A tensor A is said to be a rank-$(R_1, R_2, ..., R_d)$ tensor if all rank R_j is the exact n-rank corresponding to j-dimension. The Tucker decomposition of rank-$(R_1, R_2, ..., R_d)$ tensor can be easily computed by computing the matrix singular value decomposition of the matricized tensor in each dimension. It is worth noting that in the case that one or more ranks are chosen to be smaller than the n-rank corresponding to that dimension, i.e. $R_j < \text{rank}_j(A)$, the decomposition is called truncated Tucker decomposition. We note that R_j can also be interpreted as the dimension of the vector space spanned by the column vectors of $\mathbf{A}_{(j)}$ and we will simply refer to R_j as mode-j Tucker rank and $\mathbf{R}^t = (R_1, R_2, ...R_d)$, where the superscript t is an abbreviation for Tucker, as Tucker rank from now on.

For better understanding and due to the relevance to this dissertation work, we present Tucker decomposition for a three-dimensional tensor, which is of central interest in this work, in the remainder of this section. The Tucker model for a three-dimensional tensor is originally referred to as $Tucker3$ decomposition [99]. For a three-dimensional tensor $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the $Tucker3$ decomposition with Tucker rank $\mathbf{R}^t = (R_1, R_2, R_3)$ has the form

$$A \approx A^{(\mathbf{R}^t)} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \sigma_{r_1 r_2 r_3} \mathbf{u}^1_{r_1} \otimes \mathbf{u}^2_{r_2} \otimes \mathbf{u}^3_{r_3}, \tag{3.26}$$

where $\sigma \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ denotes the core tensor, $\mathbf{u}^j_{r_j} \in \mathbb{R}^{I_j}, j \in \{1, 2, 3\}$ is the r_j-th

rank-one component in j-dimension. We note that the tensor contraction operations in Eq. 3.26 can also be written as a series of tensor matrix multiplication (defined in Eq. 3.11) between the core tensor σ and the factor matrices. The j factor matrix of the decomposition is denoted as $\mathbf{U}^j$ and is defined as the collection of all R_j $\mathbf{u}^j_{r_j}$ vectors as

$$\mathbf{U}^j = \begin{bmatrix} \mathbf{u}^j_1 & \mathbf{u}^j_2 & ... & \mathbf{u}^j_{R_j} \end{bmatrix} \in \mathbb{R}^{I_j \times R_j}. \tag{3.27}$$

Therefore, Eq. 3.26 can also be written as

$$A \approx A^{(\mathbf{R}^t)} = \sigma \times_1 \mathbf{U}^1 \times_2 \mathbf{U}^2 \times_3 \mathbf{U}^3. \tag{3.28}$$

We note that in the literatures [99], $Tucker1$ decomposition refers to a special case that any one of $\mathbf{U}^j$ is held as an identity matrix $\mathbf{I}^j$, e.g. if mode-3 is set to an identity, $Tucker1$ decomposition has the form

$$A \approx A^{(\mathbf{R}^t)} = \sigma \times_1 \mathbf{U}^1 \times_2 \mathbf{U}^2 \times_3 \mathbf{I}^3. \tag{3.29}$$

Similarly, $Tucker2$ refers to a special case that any two of $\mathbf{U}^j$ are set to identity $\mathbf{I}^j$, e.g. if mode-2 and mode-3 are set to identities, $Tucker2$ decomposition gives

$$A \approx A^{(\mathbf{R}^t)} = \sigma \times_1 \mathbf{U}^1 \times_2 \mathbf{I}^2 \times_3 \mathbf{I}^3. \tag{3.30}$$

A graphical illustration of the Tucker decomposition for a general three-dimensional tensor is presented in Fig. 3.3. It is worth noting that the core tensor can be viewed as the higher-order generalization of singular values storing the coefficients $\sigma_{r_1 r_2 r_3}$ for each rank-one tensor $\mathbf{u}^{r_1}_1 \otimes \mathbf{u}^{r_2}_2 \otimes \mathbf{u}^{r_3}_3$, and $\mathbf{u}^{r_j}_j$ can be interpreted as singular vectors in j-dimension. Accordingly, the mode-j factor matrix can be viewed as a higher-order analogue of left/right singular matrix in a matrix singular value decomposition.

We note that many approaches have been proposed to obtain the Tucker decom-

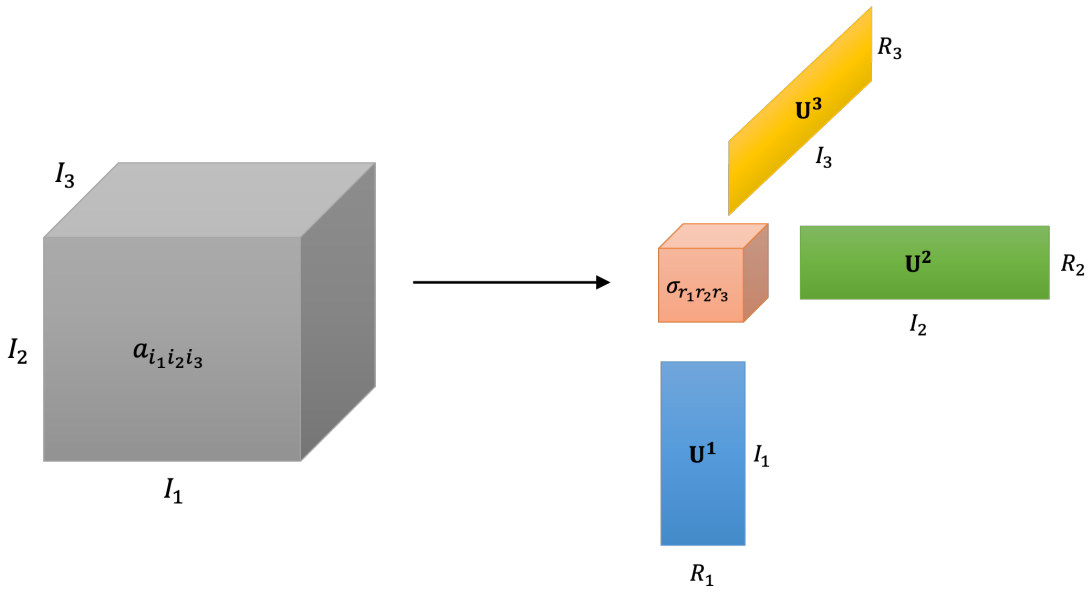


Figure 3.3: Schematic of Tucker decomposition.

position for a given tensor. In this work, we adopt $TuckerMethod1$, which is later on referred to and much more known as truncated higher-order singular value decomposition (HOSVD), for computing rank-(R_1, R_2, R_3) decomposition of the original tensor. As implied by its name, truncated HOSVD computes the singular value decomposition of the tensor A unrolled in each dimension and keeps R_j leading singular vectors of $\mathbf{A}_{(j)}$ for each dimension. The algorithm of truncated HOSVD is summarized in Algorithm 1. While the algorithm is presented in the context of a three-dimensional tensor, its form can be easily generalized to a d-way tensor. It is also noted that code development to improve various aspects of Tucker tensor decomposition is still an active area of research. In particular, as message passing interface (MPI) is heavily used for parallelization in this work, we use TuckerMPI library for computing truncated HOSVD. TuckerMPI is an MPI implementation for parallel Tucker tensor decomposition and various tensor operations in Tucker representation. We refer to [139, 140] for the library, and details of the implementation.

We note that besides truncated HOSVD, many other algorithms are also available for computing Tucker tensor decomposition of a given tensor [135, 136, 141, 142, 143].

Algorithm 1: truncated HOSVD

Input: A, R_1, R_2 R_3
Output: σ, $\mathbf{U}^1$, $\mathbf{U}^2$, $\mathbf{U}^3$
for j=1:3
  $\mathbf{U}^j \leftarrow$ first R_j singular vectors of the $\mathbf{A}_{(j)}$
end
$\sigma \leftarrow A \times_1 \mathbf{U}^1 \times_2 \mathbf{U}^2 \times_3 \mathbf{U}^3$

For example, one of the most used algorithms $TuckerALS$, also know as higher-order orthogonal iteration (HOOI), is the alternating least square algorithm for Tucker tensor decomposition [135]. It uses the solution from truncated HOSVD as an initial guess and tries to improve the quality of the solution by iteratively minimizing the residual norm of the original tensor and the tensor in Tucker decomposition form

$$A^{(\mathbf{R}^t)} = \underset{A'=\sigma' \times_1 \mathbf{U}'^1 \times_2 \mathbf{U}'^2 \times_3 \mathbf{U}'^3}{\arg\min} \|A - \sigma' \times_1 \mathbf{U}'^1 \times_2 \mathbf{U}'^2 \times_3 \mathbf{U}'^3\|. \tag{3.31}$$

After obtaining the Tucker decomposition of a tensor, the full tensor could be needed in some cases. The process of recovering the full tensor from a tensor decomposition format is called (Tucker) tensor reconstruction. We caution that while it seems to be intuitive to compute Tucker tensor reconstruction using for-loop as the multiple sums in Eq. 3.23 seems to suggest so, the computational cost for this operation is $\mathcal{O}(R^3 I^3)$, where $I \coloneqq \max(I_1, I_2, I_3)$ and $R \coloneqq \max(R_1, R_2, R_3)$, which is actually very inefficient. A more efficient way is to treat the tensor reconstruction as a series of tensor-matrix multiplications as shown in Eq. 3.28 and presented in Algorithm 2. The computational cost for this operation is $\mathcal{O}(R^3 I + R^2 I^2 + R I^3)$, which can be orders of magnitude more efficient than naïvely looping over each summation notation.

Finally, we refer to [90, 144, 89] for more details on the various aspects of Tucker tensor decomposition and the implementations of various algorithms for obtaining Tucker tensor decomposition.

Algorithm 2: Tucker tensor reconstruction

Input: σ, $\mathbf{U}^1$, $\mathbf{U}^2$, $\mathbf{U}^3$
Output: A
$P \leftarrow \sigma$
for j=1:3
  $Q \leftarrow P \times_j \mathbf{U}^j$ (done by computing $\mathbf{P}_{(j)}\mathbf{U}^j$ using optimized kernel, e.g. `dgemm`)
  $P \leftarrow Q$
 end
$A \leftarrow P$

3.4 Other tensor decomposition models

Apart from the standard canonical tensor decomposition and Tucker tensor decomposition, many other decomposition models are developed based on canonical and Tucker decomposition and have been used in different application cases. *Canonical decomposition with linear constraints (CANDELINC)* proposed a decomposition model based on Canonical decomposition [145]. The CANDELINC model finds the canonical decomposition of a tensor with one or more factor matrices under linear constraints. It is then also termed Tucker-to-canonical operation [117, 43]. In electronic structure calculations, it can be used to compute classical potentials emerging in many-body electronic structure calculations, i.e. the Newton, Yukawa, and Helmholtz potentials [117]. It is also used to design a multi-grid accelerated tensor approximation model for approximating function related d-way tensors [43]. *Generalized canonical polyadic decomposition (GCP)* is also proposed recently [146]. In standard canonical decomposition, the loss function is defined by the residual norm as shown in Eq. 3.22. While in GCP, a loss function other than residual norms is allowed. The GCP thus provides more flexibility than standard CP to the application of canonical decomposition model on tensors with some special structures, e.g. finding canonical decomposition of a tensor with missing data, binary tensors, and so on. *Parallel profiles with linear dependencies (PARALIND)* combines the ideas of canon-

ical decomposition and Tucker decomposition. PARALIND decomposes a tensor as a sum of multiple low-rank Tucker tensor approximations [147]. *Individual differences in scaling (INDSCAL* [129], *parallel factors for cross products (PARAFAC2)* [148], *decomposition into directional components (DEDICOM)* [149], *shifted Tucker3 (S-T3)* [150], and many more are also available decomposition models. We refer to [90] and [147] for a comprehensive introduction on these different decomposition models based on canonical or Tucker decomposition.

Other than the decomposition based on the canonical/Tucker decomposition, *tensor train format* provides an alternative perspective of representing a tensor. Originally introduced to quantum physics and later termed as *matrix-product systems (MPS)* [151, 152], it was around ten years ago reinvented and referred to as *tensor train* or *TT format* [125, 114]. Tensor train representation for a given tensor has the form represented by its entries as

$$a_{i_1 i_2 \ldots i_d} \approx \sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \ldots \sum_{r_{d-1}}^{R_{d-1}} g^1_{i_1 r_1} g^2_{r_1 i_2 r_2} \quad \ldots \quad g^{d-1}_{r_{d-2} i_{d-1} r_{d-1}} g^d_{r_{d-1} i_d}, \tag{3.32}$$

where g denotes an entry in a three-dimensional tensor train core tensor G^j, R_j denotes the tensor train rank (TT rank).

Finally, as a concluding remark, we note that there have also been other tensor decomposition schemes having different forms from canonical or Tucker decomposition. Particularly, *tree tensor network (tree TN* [2]*)* [153], *hierarchical tensor format (HT)* [154, 155, 156] are also available and used in the context of quantum mechanical or electronic structure calculations. We refer to the citations therein for more details.

[2] Actually, TT originally refers to tree tensor but later on more commonly used to refer to tensor train in Eq. 3.32 [89].

CHAPTER IV

Reduced-Order Tensor-Structured Algorithm for Kohn-Sham Density Functional Theory

In this chapter, we introduce the tensor-structured techniques to construct a reduced-order tensor-structured algorithm for Kohn-Sham DFT calculation. We refer to the previous Chapter II for the theoretical background of the DFT calculation and Chapter III for various aspects of tensor techniques and the nomenclature, especially Tucker tensor decomposition, that will be used extensively in this chapter. This chapter is structured as follows. Section 4.1 introduces the Kohn-Sham DFT formulation in the real-space setting that will be used in this work. Section 4.2 proposes a reduced-order tensor-structured algorithm for Kohn-Sham DFT calculations based on Tucker tensor basis constructed from L_1 localized functions. In Section 4.3, we further investigate the eigenspace representability of the L_1 localized functions to confirm that the localized Tucker-tensor basis preserves the same eigenspace. Section 4.4 shows the convergence study for two benchmark systems, fullerene and tris (bipyridine) ruthenium, using the proposed tensor-structured algorithm. Section 4.5 shows the performance analysis of the proposed tensor-structured algorithm for Kohn-Sham DFT calculation using aluminum nano-particles and silicon quantum dots. Finally, we conclude this work in Section 4.6.

4.1 Real-space Kohn-Sham DFT formulation

Based on the derivation for Kohn-Sham DFT shown in Sec. 2.3.2, we will proceed to show the real-space formulation for the Kohn-Sham method in a non-periodic boundary setting. Due to the relevance to this work, we only consider spinless DFT hereafter unless otherwise specified.

As aforementioned, Kohn-Sham DFT addresses the ground state energy of a quantum mechanical system with N_a atoms and N_e electrons by solving a non-interacting single-particle Schrödinger equation subjected to a mean-field effective potential $v_{\text{eff}}(\rho;\mathbf{R})$

$$\begin{aligned} &\mathcal{H}^{KS}\Psi_i^{KS} = \epsilon_i \Psi_i^{KS}, \qquad i \in \{1, ..., N_{\text{orb}}\} \\ &\mathcal{H}^{KS} = -\frac{1}{2}\nabla^2 + v_{\text{eff}}(\rho;\mathbf{R}) \end{aligned} \tag{4.1}$$

In the above, $\mathcal{H}^{KS}$ denotes the Kohn-Sham Hamiltonian, $\{\epsilon_i, \Psi_i^{KS}\}$ denotes the i-th eigenstate, N_{orb} denotes the number of eigenstates at the lower end of the spectrum that are computed ($N_{\text{orb}} > \frac{N_e}{2}$), and $\mathbf{R}$ denotes the vector with the positions of atoms. The electron density—the central quantity of interest in DFT— is denoted by $\rho = \rho(\mathbf{x})$ in real-space with coordinates $\mathbf{x} = (x_1, x_2, x_3)$, where the spin dependency is dropped as aforementioned. We also choose to drop the KS superscript from now on for the sake of ease of presentation unless otherwise mentioned. We caution that a single-electron wavefunction $\Psi(\mathbf{x})$ appears in this chapter should not be confused with the many-body wavefunction $\Psi(\mathbf{x}_1, \cdots, \mathbf{x}_{N_e})$ in Sec. 2.1. The electron density is related to the Kohn-Sham orbitals by

$$\rho(\mathbf{x}) = 2\sum_{i=1}^{N_{\text{orb}}} f(\epsilon_i;\mu)\left|\Psi_i(\mathbf{x})\right|^2, \tag{4.2}$$

where $f(\epsilon;\mu)$ denotes the orbital occupancy function, and, in the present work, is

represented by the Fermi-Dirac distribution

$$f(\epsilon;\mu) = \frac{1}{1+\exp(\frac{\epsilon-\mu}{k_B T})}\,. \tag{4.3}$$

Here, k_B is the Boltzmann constant, T is the temperature controlling the smearing of the orbital occupancy function, and μ is the Fermi energy that is solved using the the constraint on the total number of electrons given by

$$2\sum_{i=1}^{N_{\text{orb}}} f(\epsilon_i;\mu) = N_e\,. \tag{4.4}$$

As a quick recap from Eq. 2.54, the effective potential in the Kohn-Sham Hamiltonian, $v_{\text{eff}}(\rho)$, is a functional of electron density, and comprises three contributions

$$v_{\text{eff}}(\rho) = \frac{\delta E_{\text{H}}}{\delta\rho} + \frac{\delta E_{\text{XC}}}{\delta\rho} + v_{\text{ext}}(\mathbf{x};\mathbf{R}). \tag{4.5}$$

E_{H} is the Hartree energy, which represents the classical Coulomb electrostatic interaction between electrons and is given by (in a non-periodic setting)

$$E_{\text{H}} = \frac{1}{2}\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{\rho(\mathbf{x})\rho(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}d\mathbf{x}d\mathbf{x}' = \int_{\mathbb{R}^3} \rho(\mathbf{x})v_{\text{H}}(\rho)d\mathbf{x}, \tag{4.6}$$

where $v_{\text{H}}(\rho)$ is the Hartree potential defined by the functional derivative of the Hartree energy

$$v_{\text{H}}(\rho) = \frac{\delta E_{\text{H}}}{\delta\rho} = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|}d\mathbf{x}'. \tag{4.7}$$

E_{XC} is the exchange-correlation energy, which describes all the many-body quantum mechanical interactions between electrons. The functional derivative of E_{XC} is labeled as the exchange-correlation potential

$$v_{\text{XC}}(\rho) = \frac{\delta E_{\text{XC}}}{\delta\rho}, \tag{4.8}$$

In this work, the local density approximation (LDA) in the form of Ceperley-Alder parametrized with Perdew-Zunger data [76, 78] is used for the exchange-correlation functional. The last term in Eq. (4.5), $v_{\text{ext}}(\mathbf{x};\mathbf{R})$, is the electrostatic potential acting on electrons induced by nuclei. Typically, the core electrons do not participate in chemical reactions, hence a pseudopotential approximation is commonly adopted to replace the all-electron Coulomb potential by a smoother potential acting only on valence electrons. The behavior of the pseudopotential operator v_{ext} acting on valence electrons comprises the local part $v_{\text{ext}}^{\text{loc}}$ and the non-local part $v_{\text{ext}}^{\text{nl}}$. In this work, the norm-conserving Troullier-Martin [157] pseudopotential in Kleinman-Bylander [158] form is used. The real space action of the pseudopotential operator acting on the Kohn-Sham orbitals is defined as

$$v_{\text{ext}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) = v_{\text{ext}}^{\text{loc}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) + v_{\text{ext}}^{\text{nl}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}). \tag{4.9}$$

$$v_{\text{ext}}^{\text{loc}}(\mathbf{x};\mathbf{R}) = \sum_{J=1}^{N_a} v_{\text{ext}}^{loc,J}(\mathbf{x}-\mathbf{R}_J)\Psi(\mathbf{x}), \tag{4.10}$$

where $v_{\text{ext}}^{loc,J}(\mathbf{x}-\mathbf{R}_J)$ is the corresponding local potential for the J-th atom, and $\mathbf{R}_J$ is the coordinate of the J-th atom.

$$v_{\text{ext}}^{\text{nl}}(\mathbf{x};\mathbf{R}) = \sum_{J}^{N_a}\sum_{lm} C_{lm}^{J}\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta v_l^J(\mathbf{x}-\mathbf{R}_J), \tag{4.11}$$

where

$$C_{lm}^{J} = \frac{\int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta v_l^J(\mathbf{x}-\mathbf{R}_J)\Psi(\mathbf{x})d\mathbf{x}}{\int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta v_l^J(\mathbf{x}-\mathbf{R}_J)\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)d\mathbf{x}}$$

and

$$\Delta v_l^J(\mathbf{x}-\mathbf{R}_J) = v_l^J(\mathbf{x}-\mathbf{R}_J) - v_{\text{ext}}^{loc,J}(\mathbf{x}-\mathbf{R}_J).$$

Therein, $v_l^J(\mathbf{x})$ is the pseudopotential component of the J-th atom corresponding to the l azimuthal quantum number, $\varphi_{lm}^{J}(\mathbf{x})$ is the single atom pseudo-wavefunction of

the J-th atom corresponding to the azimuthal and magnetic quantum numbers l and m, respectively.

Finally, upon solving Eq. (4.1), Eq. (4.2), and Eq. (4.4) self-consistently in a suitable basis, the ground state energy of the given system can be obtained by

$$E_{\text{tot}} = E_{\text{band}} + E_{\text{XC}} - \int_{\mathbb{R}^3} \rho v_{\text{XC}}(\rho) d\mathbf{x} - \frac{1}{2} \int_{\mathbb{R}^3} \rho v_{\text{H}}(\rho) d\mathbf{x} + E_{\text{ZZ}}\,, \tag{4.12}$$

where

$$E_{\text{band}} = 2 \sum_{i=1}^{N_{\text{orb}}} f(\epsilon_i; \mu) \epsilon_i$$

is the band energy. Finally,

$$E_{\text{ZZ}} = \sum_{I=1}^{N_a} \sum_{J>I}^{N_a} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}\,,$$

is the repulsion energy between nuclei, where Z_I is the valence charge of the I-th atom.

4.2 Tensor-structured algorithm using L_1 localized one-dimensional functions

In the previous work [45], it was suggested that an additive separable approximation to the Kohn-Sham Hamiltonian can be used to construct a Tucker tensor basis that is systematically convergent. In particular, using a tensor-structured cuboidal domain Ω spanned by the tensor product of one-dimensional domains $\omega_{k=1,2,3}$, an additive separable approximation to the Kohn-Sham Hamiltonian ($\mathcal{H}_1(x_1) + \mathcal{H}_2(x_2) + \mathcal{H}_3(x_3) \approx \mathcal{H}(\mathbf{x})$) retains some features of the Hamiltonian, and thus presents a useful operator to generate reduced-order basis functions. To this end, the eigenfunctions of the additive separable approximation to the Hamiltonian, which constitute a Tucker tensor basis formed from the one-dimensional eigenfunctions of the separable

parts of the Hamiltonian ($\mathcal{H}_k$, $k = 1, 2, 3$), are used to solve the Kohn-Sham equations. While an efficient basis, the global nature of the ensuing Tucker tensor basis limits the computational efficiency of the algorithms to solve the Kohn-Sham equations. In the proposed work, in place of the one-dimensional eigenfunctions of $\mathcal{H}_k$, we instead construct compressed modes preserving the subspace spanned by the one-dimensional eigenfunctions using the L_1 localization technique [159]. The obtained one-dimensional localized functions are then used to generate the three-dimensional Tucker tensor basis, which is localized in real-space and allows us to exploit the sparsity of the Kohn-Sham Hamiltonian represented in this basis for both computational efficiency and realizing reduced-order scaling in solving the Kohn-Sham equations. The various aspects of our tensor-structured algorithm are now presented, which include the computation of Kohn-Sham effective potential using tensor-structured algorithm, generation of the additive separable approximation of the Kohn-Sham Hamiltonian, the evaluation of the L_1 localized one-dimensional functions, the construction of the localized Tucker tensor basis, the projection of the Kohn-Sham problem onto the localized Tucker tensor basis, and the solution of the Kohn-Sham equations.

4.2.1 Computation of Khon-Sham effective potential using tensor-structured techniques

Here we elaborate the various aspects of utilizing the low-rank tensor decomposition to evaluate and represent the various components of the Kohn-Sham effective potential. Recall that in Eq. 4.9, the pseudopotential is decomposed as its local and non-local part. As a result, the effective potential can also be decomposed as

$$v_{\text{eff}}(\rho; \mathbf{R}) = v_{\text{eff}}^{\text{loc}}(\rho; \mathbf{R}) + v_{\text{ext}}^{\text{nl}}(\rho; \mathbf{R}) \,,$$

where $v_{\text{eff}}^{\text{loc}}$ denotes the local part of the Kohn-Sham effective potential and $v_{\text{ext}}^{\text{nl}}$ denotes the non-local part, which is simply the non-local part of the pseudopotential. The local part includes the Hartree potential v_{H}, the exchange-correlational functional potential v_{XC}, and the local part of the external pseudopotential $v_{\text{ext}}^{\text{loc}}$, namely

$$v_{\text{eff}}^{\text{loc}}(\rho;\mathbf{R}) = v_{\text{H}}(\rho) + v_{\text{XC}}(\rho) + v_{\text{ext}}^{\text{loc}}(\rho;\mathbf{R})\,. \tag{4.13}$$

4.2.1.1 Hartree Potential

The Hartree potential v_{H} is computed following the tensor-structured approach presented in [160, 161]. We recall that the Hartree potential is given by the convolution integral

$$v_{\text{H}}(\rho) = \int \frac{\rho(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|} d\mathbf{x}'\,. \tag{4.14}$$

We first compute a low-rank Tucker decomposition of the electron density as

$$\tilde{\rho}(\mathbf{x}) = \sum_{r_1,r_2,r_3=1}^{R_\rho} g_{r_1 r_2 r_3} \varrho_1^{r_1}(x_1)\varrho_2^{r_2}(x_2)\varrho_3^{r_3}(x_3)\,, \tag{4.15}$$

In the above, R_ρ represents the Tucker rank associated with the low-rank approximation of the electron density. We note that the approximation error decays exponentially with the Tucker rank. In general, the Tucker rank can be chosen to be different along the three Cartesian directions. However, for the sake of simplicity, the ideas are presented here using a uniform rank along the different Cartesian directions.

The kernel $\frac{1}{|\mathbf{x}-\mathbf{x}'|}$ in Eq. 4.14 is approximated by a series of Gaussian functions to take advantage of the tensor-structured nature as

$$\frac{1}{|\mathbf{x}|} \approx \sum_{k=1}^{K} w_k e^{-\alpha_k(x_1^2+x_2^2+x_3^2)}\,, \tag{4.16}$$

where w_k and α_k are coefficients and K is the number of terms used to expand the

kernel. We will elaborate this part a little bit in the later chapter (Sec. 6.4). We also refer to prior works [161, 117] for the complete derivation and algorithm. The pre-computed coefficients can be found on the webpage [162]. Substituting Eq. 4.15 and Eq. 4.16 into Eq. 4.14 , the Hartree potential can be evaluated using a separable form

$$\begin{aligned}\tilde{v}_{\mathrm{H}}(\mathbf{x}) = \sum_{k=1}^{K} w_k \sum_{r_1,r_2,r_3=1}^{R_\rho} \Bigg(g_{r_1 r_2 r_3} \int \varrho_1^{r_1}(x_1') e^{-\alpha_k (x_1 - x_1')^2} dx_1' \\ \int \varrho_2^{r_2}(x_2') e^{-\alpha_k (x_2 - x_2')^2} dx_2' \int \varrho_3^{r_3}(x_3') e^{-\alpha_k (x_3 - x_3')^2} dx_3' \Bigg) .\end{aligned} \tag{4.17}$$

4.2.1.2 Local part of pseudopotential

The local part of the effective potential is then computed by summing the Hartree potential, exchange-correlation functional and the local part of the external potential. We next compute the Tucker decomposition of the local part of the effective potential to exploit the tensor structure in the computation of the Hamiltonian matrix elements in the Tucker tensor basis (will be elaborated later in Sec. 4.2.5). The Tucker decomposed effective potential is thus represented as

$$\tilde{v}_{\mathrm{eff}}^{\mathrm{loc}}(\mathbf{x}) = \sum_{r_1,r_2,r_3=1}^{R_V} \sigma_{r_1 r_2 r_3}^{V} u_1^{r_1}(x_1) u_2^{r_2}(x_2) u_3^{r_3}(x_3) , \tag{4.18}$$

where R_V is the Tucker rank of the local part of the Kohn-Sham effective potential.

4.2.1.3 Non-local projector of pseudopotential

Now, recall that in Eq. 4.9, the norm-conserving Troullier Martin pseudopotential operator in Kleinmann-Bylander form v_{ext} comprises the local part $v_{\mathrm{ext}}^{\mathrm{loc}}$ and the non-local part $v_{\mathrm{ext}}^{\mathrm{nl}}$. The action of the pseudopotential operator on a Kohn-Sham orbital in real-space is recapitulated here as

$$v_{\mathrm{ext}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) = v_{\mathrm{ext}}^{\mathrm{loc}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) + v_{\mathrm{ext}}^{\mathrm{nl}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) , \tag{4.19}$$

$$v_{\text{ext}}^{\text{loc}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) = \sum_{J=1}^{N_a} v_{\text{ext}}^{loc,J}(\mathbf{x}-\mathbf{R}_J)\Psi(\mathbf{x})\,, \tag{4.20}$$

where

$$v_{\text{ext}}^{\text{nl}}(\mathbf{x};\mathbf{R})\Psi(\mathbf{x}) = \sum_{J}^{N_a}\sum_{lm} C_{lm}^{J}\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta v_l^J(\mathbf{x}-\mathbf{R}_J)\,, \tag{4.21}$$

and

$$\begin{aligned} C_{lm}^{J} &= \frac{\int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta {V_l}^{J}(\mathbf{x}-\mathbf{R}_J)\Psi(\mathbf{x})d\mathbf{x}}{\int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta {V_l}^{J}(\mathbf{x}-\mathbf{R}_J)\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)d\mathbf{x}} \\ &= \frac{1}{\nu_{lm}^{J}}\int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta {V_l}^{J}(\mathbf{x}-\mathbf{R}_J)\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)d\mathbf{x}\,, \end{aligned}$$

where

$$\nu_{lm}^{J} := \int \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta {V_l}^{J}(\mathbf{x}-\mathbf{R}_J)\varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)d\mathbf{x}\,.$$

In order to efficiently compute the action of non-local projector on the Kohn-Sham wavefunction, we introduce an intermediate term $\Lambda_{lm}^{J}(\mathbf{x}) = \varphi_{lm}^{J}(\mathbf{x}-\mathbf{R}_J)\Delta {V_l}^{J}(\mathbf{x}-\mathbf{R}_J)$ and compute its Tucker decomposition denoted as

$$\tilde{\Lambda}_{lm}^{J}(\mathbf{x}) = \sum_{r_1,r_2,r_3=1}^{R_V^{\text{nl}}} \sigma_{r_1r_2r_3}^{\Lambda_{lm}^{J}} \phi_1^{\Lambda_{lm}^{J},r_1}(x_1)\phi_2^{\Lambda_{lm}^{J},r_2}(x_2)\phi_3^{\Lambda_{lm}^{J},r_3}(x_3), \tag{4.22}$$

where R_V^{nl} is the associated Tucker rank, chosen to be the largest among the J atoms and its corresponding quantum numbers l and m.

The non-local part of the external potential computed with the Tucker decomposed quantities is denoted as $\tilde{v}_{\text{ext}}^{\text{nl}}(\mathbf{x})$, and given

$$\tilde{v}_{\text{ext}}^{\text{nl}}(\mathbf{x})\Psi_i(\mathbf{x}) = \sum_{J}^{N_a}\sum_{lm} \tilde{C}_{lm}^{J}\tilde{\Lambda}_{lm}^{J}(\mathbf{x})\,, \tag{4.23}$$

where

$$\tilde{C}_{lm}^{J} = \frac{1}{\nu_{lm}^{J}}\int \tilde{\Lambda}_{lm}^{J}(\mathbf{x})\Psi(\mathbf{x})d\mathbf{x}\,.$$

4.2.2 Construction of separable Hamiltonian

Next, we seek to construct a separable approximation to the Kohn-Sham Hamiltonian $\mathcal{H}_1(x_1) + \mathcal{H}_2(x_2) + \mathcal{H}_3(x_3) \approx \mathcal{H}(\mathbf{x})$ based on a rank-1 approximation of the eigenfunction corresponding to the lowest eigenvalue. To this end, we consider the rank-1 representation for the eigenfunction as $\Psi'(\mathbf{x}) = \psi_1(x_1)\psi_2(x_2)\psi_3(x_3)$. Thus, the problem of computing the smallest eigenvalue of the Kohn-Sham Hamiltonian using the rank-1 approximation is given by the variational problem

$$\min_{\psi_k} \mathcal{L}(\Psi') \qquad \text{subject to: } \langle \Psi' | \Psi' \rangle = 1 \,. \tag{4.24}$$

The Lagrangian, accounting for the normality constraint with the Lagrange multiplier λ, is given by

$$\begin{aligned} \mathcal{L}(\Psi') &= \left\langle \Psi' \right| -\frac{1}{2}\nabla^2 + v_{\text{eff}}(\mathbf{x}) \left| \Psi' \right\rangle \\ &= \int \left[\sum_{\ell=1}^{3} \frac{1}{2} \left| \frac{d\psi_\ell(x_\ell)}{dx_\ell} \right|^2 \prod_{m\neq\ell}^{3} \psi_m^2(x_m) + \left(\tilde{v}_{\text{eff}}^{\text{loc}}(\mathrm{x}) + \lambda \right) \prod_{\ell=1}^{3} \psi_\ell^2(x_\ell) \right. \\ &\quad \left. + \prod_{\ell=1}^{3} \psi_\ell(x_\ell) \tilde{v}_{\text{ext}}^{\text{nl}}(\mathbf{x}) \prod_{\ell=1}^{3} \psi_\ell(x_\ell) \right] d\mathbf{x} \,. \end{aligned} \tag{4.25}$$

Taking the variation of the Lagrangian Eq. 4.25 with respect to ψ_ℓ, we arrive at the simultaneous one-dimensional eigenvalue problems

$$\begin{aligned} &\left(-\frac{1}{2}\frac{d^2}{dx_k^2} + v_k^{\text{loc}}(x_k; \psi_{l\neq k}) + v_k^{\text{nl}}(x_k; \psi_{l\neq k}) \right) \psi_k(x_k) = -(\lambda + a_k)\psi_k(x_k) \,, \\ &k = 1, 2, 3 \,. \end{aligned} \tag{4.26}$$

We note that

$$\mathcal{H}_k := -\frac{1}{2}\frac{d^2}{dx_k^2} + v_k^{\text{loc}}(x_k; \psi_{l\neq k}) + v_k^{\text{nl}}(x_k; \psi_{l\neq k}) \tag{4.27}$$

and

$$\alpha_k = -(\lambda + a_k) \,. \tag{4.28}$$

Therein, we write out the one-dimensional quantities in Eq. 4.26

$$v_k^{\mathrm{loc}}(x_k;\psi_{l\neq k}) = \frac{1}{m_k}\int \tilde{v}_{\mathrm{eff}}^{\mathrm{loc}} \prod_{p=1,p\neq k}^{3} \psi_p^2(x_p) d\hat{\mathbf{x}}_{\mathbf{k}}\,, \tag{4.29}$$

where

$$d\hat{\mathbf{x}}_{\mathbf{k}} = \prod_{p=1,p\neq k}^{3} d\mathbf{x}_{\mathbf{p}}\,, \quad m_k = \int \prod_{p=1,p\neq k}^{3} \psi_p^2 d\hat{\mathbf{x}}_{\mathbf{k}}\,, \tag{4.30}$$

and

$$v_k^{\mathrm{nl}}(x_k;\psi_{l\neq k})\psi_k(x_k) = \frac{1}{m_k}\sum_{J}^{N_a}\sum_{lm} \tilde{C}_{lm}^{J} \int \tilde{\Lambda}_{lm}^{J}(\mathbf{x}) \prod_{p=1,p\neq k}^{3} \psi_p(x_p) d\hat{\mathbf{x}}_{\mathbf{k}}\,, \tag{4.31}$$

$$a_k = \frac{1}{2m_k}\int \sum_{\substack{p,q=1\\p,q\neq k}}^{3} \left|\frac{d\psi_p(x_p)}{dx_p}\right|^2 \psi_q^2(x_q) d\hat{\mathbf{x}}_{\mathbf{k}}\,. \tag{4.32}$$

The minimization problem can thus be written into a set of simultaneous one-dimensional eigenvalue problem, where each eigenvalue problem is parametrized by the solution of the other two directions. The simultaneous eigenvalue problem can be solved using a self-consistent iteration procedure.

Upon achieving self-consistency, the one-dimensional Hamiltonians ($\mathcal{H}_k$) we obtain represent the additive separable approximation of the Kohn-Sham Hamiltonian that we seek. The eigenfunctions of this additive separable approximation to the Hamiltonian, which can be obtained as the tensor product of the one-dimensional eigenfunctions of $\mathcal{H}_k$ ($k = 1, 2, 3$), constitute a complete basis, thus providing systematic convergence as will be demonstrated subsequently.

We note that the proposed approach represents one possibility of systematically constructing an additive separable approximation to the Kohn-Sham Hamiltonian, and other possibilities may exist. We also note that the resulting tensor-structured basis—the eigenbasis of $\mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$—is expected to be better than the plane-wave basis. To elaborate, the plane-wave basis is the eigenbasis of the Laplace op-

erator (which is additive separable), whereas the additive separable approximation obtained via the proposed approach includes both the Laplace operator and an additive separable approximation of the Kohn-Sham potential v_{eff}, thus retaining some additional features of the Kohn-Sham Hamiltonian and providing a better basis than the plane-wave basis. The superior approximation properties of the proposed tensor-structured basis over the plane-wave basis will be demonstrated subsequently via numerical benchmark studies in Sec. 4.5.

We further note that as the electron density evolves during the self-consistent field (SCF) iteration, the Kohn-Sham Hamiltonian changes, and thus the additive separable approximation to Kohn-Sham Hamiltonian also changes with SCF iteration. Here, we study the difference in the approximation properties of the adaptive Tucker tensor basis—one which is constructed as the eiegnbasis of the additive separable approximation to the Kohn-Sham Hamiltonian in every SCF iteration—with those of a fixed Tucker tensor basis where the basis is constructed only in the first SCF iteration (using an input electron density being a superposition of atomic densities) and held fixed during the course of the SCF iteration. To this end, we consider two benchmark systems: (i) Al_{147}; (ii) $Si_{220}H_{144}$. Table 4.1 compares the approximation properties of the adaptive and the fixed Tucker basis. Notably, the difference in the ground-state energies computed using the adaptive and fixed Tucker tensor basis for any given rank is significantly smaller than the basis discretization error corresponding to the rank. To elaborate, the difference in ground-state energies between the adaptive and fixed Tucker tensor basis for rank 70 for Al_{147} is < 1 meV/atom, whereas the basis discretization error corresponding to this rank is ~ 8.5 meV/atom (Reference energy is -56.6274 eV/atom). Similarly, for $Si_{220}H_{144}$, the difference between the adaptive and fixed basis is again < 1 meV/atom for rank 80, whereas the basis discretization error is ~ 7.5 meV/atom (Reference energy is -71.393 eV/atom). These results suggest that it suffices to use a fixed Tucker tensor basis.

rank	**adaptive**	**1 scf fixed**	**diff (meV)**
50	-56.58066	-56.57687	3.7970
60	-56.60263	-56.60099	1.6432
65	-56.6117	-56.61028	1.4181
70	-56.61869	-56.61798	0.7035
75	-56.62425	-56.62381	0.4384
80	-56.62637	-56.62608	0.2838

(a)

rank	**adaptive**	**1 scf fixed**	**diff (meV)**
50	-71.05245	-71.04783	4.6163
60	-71.27783	-71.27434	3.4927
70	-71.36808	-71.36661	1.4734
80	-71.38578	-71.38491	0.8699
90	-71.38866	-71.38821	0.4432
100	-71.39145	-71.39101	0.4314
105	-71.39201	-71.39181	0.1990

(b)

Table 4.1: Comparison of the approximation properties of adaptive and fixed Tucker tensor basis. The per atom ground-state energies are reported in eV for: (a) Al_{147}; (b) $Si_{220}H_{144}$.

4.2.3 Computation of L_1 localized functions

The tensor-structured basis computed using the one-dimensional eigenfunctions of $\mathcal{H}_k$ represents an efficient basis. However, the global nature of the basis limits the computational efficiency and scaling (with system-size) of the solution to the Kohn-Sham equations. To this end, we use the L_1 localization approach [159] to construct a spatially localized tensor-structured basis that is a close approximation to the original tensor-structured basis. The localized basis is obtained by solving the following variational problem (for $k = 1, 2, 3$)

$$\min_{\mathbf{\Psi}'_k \in \mathbb{R}^{n \times N_k}} \frac{1}{\mu} \left| \mathbf{\Psi}'_k \right| + \mathrm{Tr}({\mathbf{\Psi}'_k}^T \mathbf{H}_k \mathbf{\Psi}'_k) \quad \text{s.t. } {\mathbf{\Psi}'_k}^T \mathbf{\Psi}'_k = \mathbf{I}, \tag{4.33}$$

where $\mathbf{H}_k$ is the matrix representation of $\mathcal{H}_k$ in a suitable orthogonal basis with dimension n, $\mathbf{\Psi}'_k$ denotes the representation of N_k trial localized functions in the chosen basis, and μ is a parameter controlling the trade-off between the representability of the original eigensubspace and the locality of the one-dimensional functions, with $|\cdot|$ denoting the L_1 norm of the matrix.

The variational problem in Eq. 4.33 is solved by the splitting orthogonality constraint algorithm (SOC) [163, 159]. The SOC algorithm can be used for finding a set of localized functions which closely approximate the eigenspace, yet preserving the orthogonality of the localized functions. We remark that, in this work, the SOC algorithm is used to construct localized functions that closely approximate the eigensubspace of the separable approximation of the Kohn-Sham Hamiltonian. Since the one-dimensional separable Hamiltonian is computed using a finite-element discretization, which in turn results in a generalized eigenvalue problem, the eigenfunctions are $\mathbf{M}$-orthogonal ($\mathbf{M}$ denoting the overlap matrix). The ${\mathbf{\Psi}'_k}^T \mathbf{\Psi}'_k = \mathbf{I}$ constraint in Eq. 4.33 should be extended to adapt $\mathbf{M}$-orthogonality as ${\mathbf{\Psi}'_k}^T \mathbf{M} \mathbf{\Psi}'_k = \mathbf{I}$. We hereby summarize the SOC algorithm for the $\mathbf{M}$-orthogonality constraint variance.

Consider the generalized one-dimensional eigenvalue problem along direction k given by $\mathbf{H}_k\mathbf{\Psi}_k = \mathbf{M}_k\mathbf{\Psi}_k\mathbf{\Lambda}_k$, where $\mathbf{\Psi}_k \in \mathbb{R}^{n\times N_k}$ with n denoting the dimension of the basis and N_k denoting the number of lowest eigenstates of interest, and $\mathbf{M}_k$ is the overlap matrix of the k-th dimension. Using Cholesky factorization applied on $\mathbf{M}_k$ so that $\mathbf{M}_k = \mathbf{L}^T\mathbf{L}$, the orthogonality constraint is given by $(\mathbf{L\Psi}_k)^T(\mathbf{L\Psi}_k) = \mathbf{I}$. Thus, the original orthogonality constraint for $\mathbf{\Psi}_k$ is thus replaced by $\mathbf{L\Psi}_k$. The SOC algorithm for a generalized eigenvalue problem is summarized in Algorithm 3.

Algorithm 3: SOC

Input: $\mathbf{H}_k$, $\mathbf{M}_k$, μ, η, κ, tol
Output: $\mathbf{\Psi}_k^{\mathrm{L}}$
Chelosky factorization: $\mathbf{M}_k = \mathbf{L}^T\mathbf{L}$
Initialize: $\mathbf{P^0} = \mathbf{L\Psi}_k$, $\mathbf{Q^0} = \mathbf{\Psi}_k$, $\mathbf{b^0} = \mathbf{B^0} = \mathbf{0}$
while $e > tol$ **do**
1. $\mathbf{\Psi}_k^i =$ $\arg\min_{\mathbf{\Psi}'_k} \mathrm{Tr}(\mathbf{\Psi'}_k^T\mathbf{H}_k\mathbf{\Psi}'_k) + \frac{\eta}{2}\|\mathbf{\Psi}'_k - \mathbf{Q}^{i-1} + \mathbf{b}^{i-1}\|_F^2 + \frac{\kappa}{2}\|\mathbf{L\Psi}'_k - \mathbf{P}^{i-1} + \mathbf{B}^{i-1}\|_F^2$
2. $\mathbf{Q}^i = \arg\min_{\mathbf{Q}} \frac{1}{\mu}|\mathbf{Q}| + \frac{\eta}{2}\|\mathbf{\Psi}_k^i - \mathbf{Q} + \mathbf{b}^{i-1}\|_F^2$
3. $\mathbf{P}^i = \arg\min_{\mathbf{P}} \|\mathbf{L\Psi}_k^i - \mathbf{P} + \mathbf{B}^{i-1}\|_F^2$ s.t. $\mathbf{P}^T\mathbf{P} = \mathbf{I}$
4. $\mathbf{b}^i = \mathbf{b}^{i-1} + \mathbf{\Psi}_k^i - \mathbf{Q}^i$
5. $\mathbf{B}^i = \mathbf{B}^{i-1} + \mathbf{L\Psi}_k^i - \mathbf{P}^i$
6. $E^i = \frac{1}{\mu}\left|\mathbf{\Psi}_k^i\right| + \mathrm{Tr}(\mathbf{\Psi}_k^{i\,T}\mathbf{H}_k\mathbf{\Psi}_k^i)$
7. $e = \left|\frac{E^i - E^{i-1}}{E^i}\right|$
8. **if** $e <$ tol; **then** $\mathbf{\Psi}_k^{\mathrm{L}} = \mathbf{\Psi}_k^i$

end

In Algorithm 3, η and κ are the penalty factors for each constraint, tol is the stopping criteria for the error measure e. $\mathbf{\Psi}_k^{\mathrm{L}}$ is the computed localized one-dimensional functions. The solution for the three sub-problems 1-3 are

$$\begin{aligned} &2(\mathbf{H}_k + \eta + \kappa\mathbf{L})\mathbf{\Psi}_k^i = \kappa(\mathbf{P}^{i-1} - \mathbf{B}^{i-1}) + \eta(\mathbf{Q}^{i-1} - \mathbf{b}^{i-1}) \\ &\mathbf{Q}^i = \mathrm{sign}(\mathbf{\Psi}_k^i + \mathbf{b}^{i-1}) \max\left(0, \left|\mathbf{\Psi}_k^i + \mathbf{b}^{i-1}\right| - \frac{1}{\eta\mu}\right) \quad , \\ &\mathbf{P}^i = (\mathbf{L\Psi}_k^i + \mathbf{B}^{i-1})\mathbf{U}\mathbf{S}^{-\frac{1}{2}}\mathbf{V}^T \end{aligned} \tag{4.34}$$

where $\mathbf{U}$, $\mathbf{V}$, $\mathbf{S}$ are the left singular vectors, the right singular vectors, and the singular

values for $(\mathbf{L}\boldsymbol{\Psi}_k^i + \mathbf{B}^{i-1})^T(\mathbf{L}\boldsymbol{\Psi}_k^i + \mathbf{B}^{i-1})$, respectively.

Finally, upon solving the variational problem Eq. 4.33 for **M**-orthogonal variance using Algorithm 3, the minimizer of the variational problem, henceforth denoted as $\boldsymbol{\Psi}_k^{\mathrm{L}}$, provides localized functions whose span closely approximates the eigensubspace of the lowest N_k eigenfunctions of $\mathcal{H}_k$, as will be demonstrated subsequently in Sec. 4.3.

4.2.4 Computation of the three-dimensional localized Tucker tensor basis

The one-dimensional localized functions whose span is a close approximation to the subspace spanned by the one-dimensional eigenfunctions of $\mathcal{H}_k$ are subsequently used to construct the three-dimensional Tucker tensor basis. Denoting the one-dimensional localized functions as $\psi^{\mathrm{L}}_{1,i_1}(x_1)$, $\psi^{\mathrm{L}}_{2,i_2}(x_2)$, $\psi^{\mathrm{L}}_{3,i_3}(x_3)$, the three-dimensional localized tensor-structured basis functions T_I^{L} are given by

$$T_I^{\mathrm{L}} = \psi^{\mathrm{L}}_{1,i_1}(x_1)\psi^{\mathrm{L}}_{2,i_2}(x_2)\psi^{\mathrm{L}}_{3,i_3}(x_3), \tag{4.35}$$

where $1 \leq i_d \leq R_d$ and I is the composite index $I = (i_1, i_2, i_3)_{1\leq i_d \leq R_d}$. The rank of the Tucker tensor basis is given by (R_1, R_2, R_3) which denotes the number of localized one-dimensional functions in each direction. The space spanned by the three-dimensional localized tensor-structured basis functions is denoted as $\mathbb{T}^{\mathrm{L}}$.

4.2.5 Projection of the Kohn-Sham Hamiltonian onto localized Tucker tensor space

The Kohn-Sham Hamiltonian is projected onto $\mathbb{T}^{\mathrm{L}}$, the localized Tucker tensor space, spanned by the three-dimensional tensor-structured localized basis functions. We note that the Kohn-Sham effective potential v_{eff} is a functional of the electron-density ρ, and is comprised of a local-part $v_{\mathrm{eff}}^{\mathrm{loc}}$ (local in real-space) and a non-local part $v_{\mathrm{ext}}^{\mathrm{nl}}$. $v_{\mathrm{eff}}^{\mathrm{loc}}$ includes the Hartree potential (v_{H}), the exchange-correlation potential and

the local-part of the pseudopotential, whereas $v_{\text{ext}}^{\text{nl}}$ comprises of the non-local projectors of the pseudopotential as shown in Sec. 4.2.1. The convolution integral involved in the evaluation of v_{H} can be efficiently computed using a low-rank Tucker tensor decomposition of the electron-density (R_ρ denoting the rank of the decomposition) and approximating the Coulomb integral by a series of Gaussian functions [160, 161] (cf. Sec. 4.2.1.1). Subsequently, a low-rank Tucker tensor approximation of $v_{\text{eff}}^{\text{loc}}$ and $v_{\text{ext}}^{\text{nl}}$ is utilized, with R_{V} and R_{V}^{nl} denoting the corresponding ranks, respectively. Denoting the low-rank Tucker approximation of the effective potential v_{eff} by $\tilde{v}_{\text{eff}}$, whose approximation error decays exponentially with the Tucker rank [160, 42], the projection of the Kohn-Sham Hamiltonian onto $\mathbb{T}^{\text{L}}$ is given by

$$\tilde{H}_{I,J}^{\text{L}} = \left\langle T_I^{\text{L}} \right| -\frac{1}{2}\nabla^2 + \tilde{v}_{\text{eff}}(\rho;\mathbf{R}) \left| T_J^{\text{L}} \right\rangle . \tag{4.36}$$

We note that the low-rank representation $\tilde{v}_{\text{eff}}$ reduces the integrals involved in Eq. (4.36) to tensor products of one-dimensional integrals, thereby facilitating the efficient evaluation of Hamiltonian matrix elements. Using the effective potential decomposed into the Tucker tensor format, the computation of the projected Hamiltonian is elaborated here for clarity.

Substituting the decomposed quantities Eq. 4.18, Eq. 4.23 into Eq. 4.36, the entries of the Hamiltonian matrix in the localized Tucker tensor basis are given by

$$\begin{aligned}\tilde{H}_{I,J}^{\text{L}} =& \left\langle T_I^{\text{L}} \right| -\frac{1}{2}\nabla^2 + \tilde{v}_{\text{eff}}^{\text{loc}} + \tilde{v}_{\text{ext}}^{\text{nl}} \left| T_J^{\text{L}} \right\rangle \\ =& \frac{1}{2}\int \nabla T_I^{\text{L}} \cdot \nabla T_J^{\text{L}} d\mathbf{x} + \int T_I^{\text{L}} \tilde{v}_{\text{eff}}^{\text{loc}} T_J^{\text{L}} d\mathbf{x} + \int T_I^{\text{L}} \tilde{v}_{\text{ext}}^{\text{nl}} T_J^{\text{L}} d\mathbf{x} .\end{aligned} \tag{4.37}$$

Recall that $T_I^{\text{L}} = \psi_{1,i_1}^{\text{L}}(x_1)\psi_{2,i_2}^{\text{L}}(x_2)\psi_{3,i_3}^{\text{L}}(x_3)$ has a tensor structure, and thus each term in Eq. 4.37 can be computed using one-dimensional integrals as follows:

$$\begin{aligned}\int \nabla T_I^{\mathrm{L}} \cdot \nabla T_J^{\mathrm{L}} d\mathbf{x} &= \int \nabla(\psi_{1,i_1}^{\mathrm{L}} \psi_{2,i_2}^{\mathrm{L}} \psi_{3,i_3}^{\mathrm{L}}) \cdot \nabla(\psi_{1,j_1}^{\mathrm{L}} \psi_{2,j_2}^{\mathrm{L}} \psi_{3,j_3}^{\mathrm{L}})\, d\mathbf{x} \\ &= G_{i_1 j_1}^{dx_1} G_{i_2 j_2}^{x_2} G_{i_3 j_3}^{x_3} + G_{i_1 j_1}^{x_1} G_{i_2 j_2}^{dx_2} G_{i_3 j_3}^{x_3} + G_{i_1 j_1}^{x_1} G_{i_2 j_2}^{x_2} G_{i_3 j_3}^{dx_3}\,,\end{aligned} \tag{4.38}$$

where

$$G_{i_d j_d}^{dx_d} = \int \frac{d\psi_{d,i_d}^{\mathrm{L}}(x_d)}{dx_d} \frac{d\psi_{d,j_d}^{\mathrm{L}}(x_d)}{dx_d} dx_d\,,$$

and

$$G_{i_d j_d}^{x_d} = \int \psi_{d,i_d}^{\mathrm{L}}(x_d) \psi_{d,j_d}^{\mathrm{L}}(x_d) dx_d\,.$$

$$\begin{aligned}\int T_I^{\mathrm{L}} \tilde{v}_{\mathrm{eff}}^{\mathrm{loc}} T_J^{\mathrm{L}} d\mathbf{x} &= \int \Big\{ \psi_{1,i_1}^{\mathrm{L}}(x_1) \psi_{2,i_2}^{\mathrm{L}}(x_2) \psi_{3,i_3}^{\mathrm{L}}(x_3) \\ &\qquad \left(\sum_{r_1,r_2,r_3=1}^{R_V} \sigma_{r_1 r_2 r_3}^{V} u_1^{r_1}(x_1) u_2^{r_2}(x_2) u_3^{r_3}(x_3) \right) \\ &\qquad \psi_{1,j_1}^{\mathrm{L}}(x_1) \psi_{2,j_2}^{\mathrm{L}}(x_2) \psi_{3,j_3}^{\mathrm{L}}(x_3) \Big\} d\mathbf{x} \\ &= \sum_{r_1,r_2,r_3=1}^{R_V} \Big(\sigma_{r_1 r_2 r_3}^{V} \int \psi_{1,i_1}^{\mathrm{L}}(x_1) u_1^{r_1}(x_1) \psi_{1,i_1}^{\mathrm{L}}(x_1) dx_1 \\ &\qquad \int \psi_{2,i_2}^{\mathrm{L}}(x_2) u_2^{r_2}(x_2) \psi_{2,i_2}^{\mathrm{L}}(x_2) dx_2 \\ &\qquad \int \psi_{3,i_3}^{\mathrm{L}}(x_3) u_3^{r_3}(x_3) \psi_{3,i_3}^{\mathrm{L}}(x_3) dx_3 \Big)\,.\end{aligned} \tag{4.39}$$

For the non-local part of the effective potential, we first consider the expression

$$\begin{aligned}&\int T_I^{\mathrm{L}} v_{\mathrm{ext}}^{\mathrm{nl}}(\mathbf{x}) T_J^{\mathrm{L}} d\mathbf{x} \\ =&\sum_{J}^{N_a} \sum_{lm} \frac{1}{\nu_{lm}^J} \int \psi_{1,i_1}^{\mathrm{L}} \psi_{2,i_2}^{\mathrm{L}} \psi_{3,i_3}^{\mathrm{L}} \Lambda_{lm}^J(\mathbf{x}) d\mathbf{x} \int \Lambda_{lm}^J(\mathbf{x}) \psi_{1,j_1}^{\mathrm{L}} \psi_{2,j_2}^{\mathrm{L}} \psi_{3,j_3}^{\mathrm{L}} d\mathbf{x}\,.\end{aligned} \tag{4.40}$$

We note that the right-hand side of equation Eq. 4.40 is a matrix operation $Q^T Q$, where

$$Q_{Jlm,i_1 i_2 i_3} = \int \psi_{1,i_1}^{\mathrm{L}}(x_1) \psi_{2,i_2}^{\mathrm{L}}(x_2) \psi_{3,i_3}^{\mathrm{L}}(x_3) \Lambda_{lm}^J(\mathbf{x}) d\mathbf{x}\,. \tag{4.41}$$

Using the low-rank Tucker decomposition of $\Lambda^J_{lm}(\mathbf{x})$ in Eq. 4.22,

$$\tilde{\Lambda}^J_{lm}(\mathbf{x}) = \sum_{r_1,r_2,r_3=1}^{R_V^{\text{nl}}} \sigma_{r_1 r_2 r_3}^{\Lambda^J_{lm}} \phi_1^{\Lambda^J_{lm},r_1}(x_1)\phi_2^{\Lambda^J_{lm},r_2}(x_2)\phi_3^{\Lambda^J_{lm},r_3}(x_3)\,,$$

$Q_{Jlm,i_1i_2i_3}$ can be computed in a tensor decomposed form using $\tilde{\Lambda}^J_{lm}$ as

$$\tilde{Q}_{Jlm,i_1i_2i_3} = \sum_{r_1,r_2,r_3}^{R_V^{\text{nl}}} \Bigg(\sigma_{r_1 r_2 r_3}^{\Lambda^J_{lm}} \int \phi_1^{\Lambda^J_{lm},r_1}(x_1)\psi^{\text{L}}_{1,i_1}(x_1)dx_1 \int \phi_2^{\Lambda^J_{lm},r_2}(x_2)\psi^{\text{L}}_{2,i_2}(x_2)dx_2 \int \phi_3^{\Lambda^J_{lm},r_3}(x_3)\psi^{\text{L}}_{3,i_3}(x_3)dx_3\Bigg)\,.$$

Thus, the non-local part of the projected Hamiltonian in the three-dimensional localized tensor-structured format with $\tilde{v}^{\text{nl}}_{\text{ext}}$ is given by

$$\int T^{\text{L}}_I \tilde{v}^{\text{nl}}_{\text{ext}}(\mathbf{x}) T^{\text{L}}_J d\mathbf{x} = \frac{1}{\nu^J_{lm}} \tilde{Q}^T \tilde{Q}\,. \tag{4.42}$$

The projected Kohn-Sham Hamiltonian matrix elements have now been computed.

We also note that a truncation tolerance is introduced to zero out the Hamiltonian matrix elements below the tolerance. This truncation is performed in every SCF iteration, which improves the sparsity of the Hamiltonian matrix and thereby reducing the memory footprint of the calculation. Furthermore, the sparsity of the Hamiltonian matrix also reduces the computational complexity of the algorithm employed to solve the Kohn-Sham equations, which is discussed subsequently. We present data to quantify the error introduced by this truncation in the computed ground-state energy. To this end, we again consider the two benchmark problems: (a) Al_{147} and (b) $Si_{220}H_{144}$. Table 4.2 provides the ground-state energies for various values of the truncation tolerance, computed using the three-dimensional localized Tucker tensor basis with Tucker rank 70 for Al_{147} and rank 80 for $Si_{220}H_{144}$. Firstly, we note that there is a systematic decrease in the error with decreasing truncation tolerance. Notably, a truncation tolerance of 1e-4 results in a ~ 1 meV/atom error, which is significantly

Truncation tolerance	**Matrix Sparsity**	**E/atom (eV)**	**difference with no truncation (meV/atom)**
no truncation	N/A	-56.61893288	0
1.00E-16	0.294987723	-56.61886453	0.068350
1.00E-12	0.652828618	-56.61858753	0.345342
1.00E-09	0.875758917	-56.61833460	0.598280
1.00E-06	0.983255385	-56.61806465	0.868228
1.00E-04	0.996135765	-56.61791358	1.019299
1.00E-03	0.999547258	-56.61543244	3.500435

(a)

Truncation tolerance	**Matrix Sparsity**	**E/atom (eV)**	**difference with no truncation (meV/atom)**
no truncation	N/A	-71.38538743	0
1.00E-16	0.375215542	-71.38521376	0.17367
1.00E-12	0.743421981	-71.38493422	0.45321
1.00E-09	0.927431159	-71.38472245	0.66497
1.00E-06	0.992731562	-71.38462106	0.76637
1.00E-04	0.999817424	-71.38409125	1.29617
1.00E-03	0.999932175	-71.38269574	2.69168

(b)

Table 4.2: The effect of truncation tolerance on ground-state energy and sparsity of $\mathbf{H}^{\mathrm{L}}$ for: (a) Al_{147}; (b) $Si_{220}H_{144}$.

lower than the basis discretization error and the desired accuracy, but provides excellent sparsity in $\mathbf{H}^{\mathrm{L}}$. Thus, a truncation tolerance of 1e-4 has been adopted in benchmark studies on aluminum nano-particles and silicon quantum dots.

4.2.6 Computation of occupied eigenstates

The discretized Kohn-Sham problem, corresponding to Eq. 4.1, in the localized orthonormal tensor-structured basis is given by the standard eigenvalue problem

$$\mathbf{H}^{\mathrm{L}}\mathbf{\Psi}_i = \epsilon_i \mathbf{\Psi}_i\,, \quad i \in \{1, \ldots, N\} \tag{4.43}$$

where $\mathbf{H}^{\mathrm{L}}$ denotes the truncated sparse Kohn-Sham Hamiltonian matrix. We use the Chebyshev filtering based subspace iteration (ChFSI) [164] to efficiently solve the Kohn-Sham equations. The ChFSI method has been demonstrated to be efficient with good parallel scalability for real-space implementations of DFT [25, 26, 3]. In the ChFSI method, in each SCF iteration, a suitably constructed Chebyshev filter using $\mathbf{H}^{\mathrm{L}}$ is employed to construct a close approximation to the relevant eigensubspace of the occupied states. The action of the Chebyshev filter on a given subspace can be cast as a recursive iteration involving matrix-vector multiplications between $\mathbf{H}^{\mathrm{L}}$ and vectors obtained during the course of recursive iteration. Since $\mathbf{H}^{\mathrm{L}}$ is sparse, the computational complexity of the Chebyshev filtering operation scales as $\mathcal{O}(R^3N)$, where $R = \max\{R_1, R_2, R_3\}$. In ChFSI, the Chebyshev filtered vectors are orthogonalized using a Gram-Schmidt orthogonalization procedure, and subsequently the Kohn-Sham eigenstates are computed by projecting $\mathbf{H}^{\mathrm{L}}$ onto the Chebyshev filtered subspace and diagonalizing this projected Hamiltonian. The computational complexity of the orthogonalization procedure and the subspace projection scales as $\mathcal{O}(R^3N^2)$ while the diagonalization cost scales as $\mathcal{O}(N^3)$. As demonstrated in Table 4.4 and Table 4.6, Chebyshev filtering, which scales linearly with N, remains the dominant cost even at 25,000 electrons for the various benchmark examples considered in this work. We note that, at even larger system sizes, other costs that exhibit quadratic-scaling (orthogonalization and subspace projection) and cubic-scaling (diagonalization) with N can start to compete. However, at such a point, explicit diagonalization can be avoided, and already developed ideas [41] of localizing the Chebyshev filtered vectors in conjunction with Fermi-operator expansion can be adopted to retain the reduced-order scaling for systems with or without a gap. We will give a more complete discussion of Chebyshev filtered subspace iteration method in the next chapter that present the GPU acceleration aspect of the tensor-structured algorithm.

4.3 Eigenspace representability of the L_1 localized functions

We now demonstrate the ability of the L_1 localized functions to closely approximate the eigensubspace of $\mathcal{H}_k$ using Al_{147} nano-particle with icosahedral symmetry. We compute the additive separable approximation of the Kohn-Sham Hamiltonian for this nano-particle, and, then compute the lowest 70 eigenstates of $\mathcal{H}_k$. We subsequently use the L_1 localization approach to compute the localized functions that are a close approximation to the eigensubspace. For the ease of presentation, we show the lowest 5 eigenfunctions of $\mathcal{H}_1$ (one of the one-dimensional separable Hamiltonian) (top) and the corresponding one-dimensional localized functions (bottom) in Figure 4.1. An illustration of all 70 eigenstates and the corresponding one-dimensional localized functions is presented in Fig. 4.2 for readers' reference. It is evident that, while the eigenfunctions are global in nature, the functions obtained from the L_1 localization approach are localized in real-space. This locality is key to the sparsity of the Kohn-Sham Hamiltonian matrix in the Tucker tensor basis, and the resulting computational efficiency.

In order to demonstrate the accuracy of the L_1 localization approach in closely approximating the eigensubspace of the separable Hamiltonian, we consider the first 70 eigenstates of $\mathcal{H}_1$ and the eigenvalues of the matrix $K_{ij} = \left\langle \psi^{\mathrm{L}}_{1,i} \middle| \mathcal{H}_1 \middle| \psi^{\mathrm{L}}_{1,j} \right\rangle$, $1 \leq i, j, \leq 70$. Figure 4.3 shows the eigenvalues of $\mathcal{H}_1$ and the eigenvalues of K_{ij}. It is interesting to note that the eigenvalues of the first 65 states are almost identical, with only slight deviations for the higher states. This demonstrates that the space spanned by the localization functions obtained using the L_1 localization approach is a close approximation to the eigensubspace of $\mathcal{H}_k$. We also note here that better accuracy can be achieved, when necessary, by simply increasing the size of N_k to be solved for in Eq. 4.33. In order to assess the accuracy afforded by the localization procedure in the ground-state energy, we computed the ground-state energy of Al_{147} using the three-dimensional localized basis with rank 70, and compared that with the

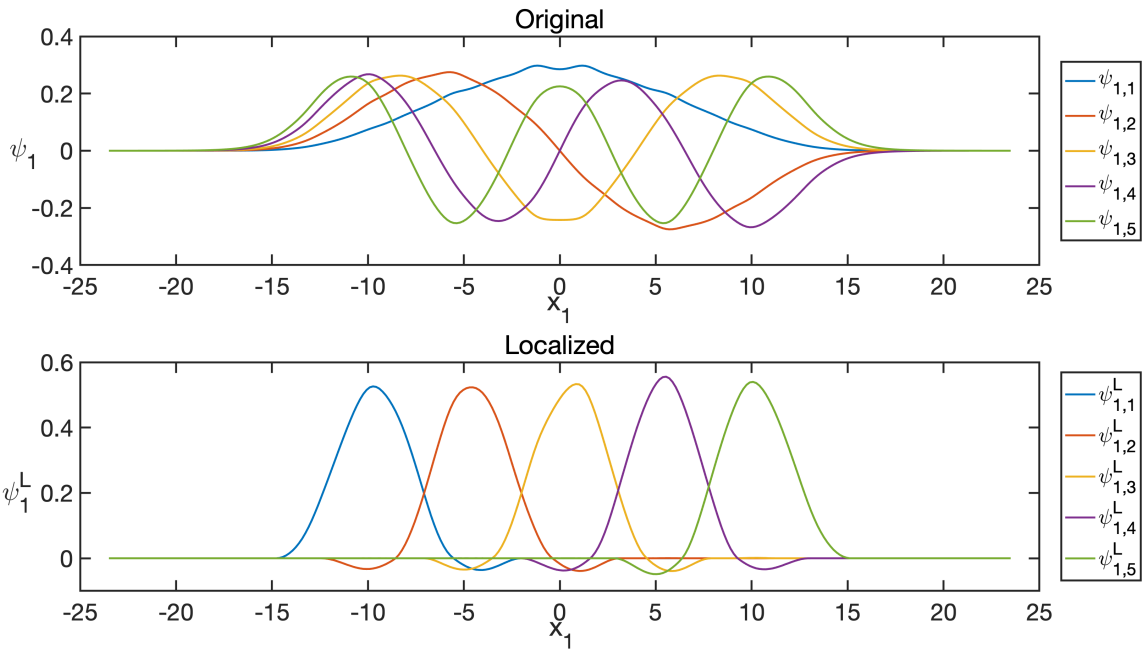


Figure 4.1: One-dimensional functions for separable Hamiltonian. one-dimensional functions in x_1 direction constructed from the additive separable approximation of the Kohn-Sham Hamiltonian for Al_{147} nano-particle. Top: Lowest five eigenfunctions of $\mathcal{H}_1$. Bottom: The corresponding L_1 localized one-dimensional functions.

energy obtained using the eigenbasis of $\mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$ (i.e., without localization) of the same rank. The energy obtained without localization is -56.61882 eV per atom in comparison to -56.61893 eV per atom obtained using localization. Thus, the error introduced due to localization is $\sim$0.1meV per atom, which is substantially smaller than the basis discretization error of $\sim$8.5 meV per atom corresponding to rank 70 (reference energy is -56.6274 eV per atom; cf. Table 4.3).

4.4 Convergence study

We next investigate the convergence properties of the three-dimensional Tucker tensor basis constructed from the one-dimensional localized functions. For the convergence study we consider two benchmark problems: (i) C_{60} (fullerene) molecule; (ii) tris (bipyridine) ruthenium, a transition metal complex. We note that these systems have no tensor structure symmetry and serve as stringent benchmarks to assess

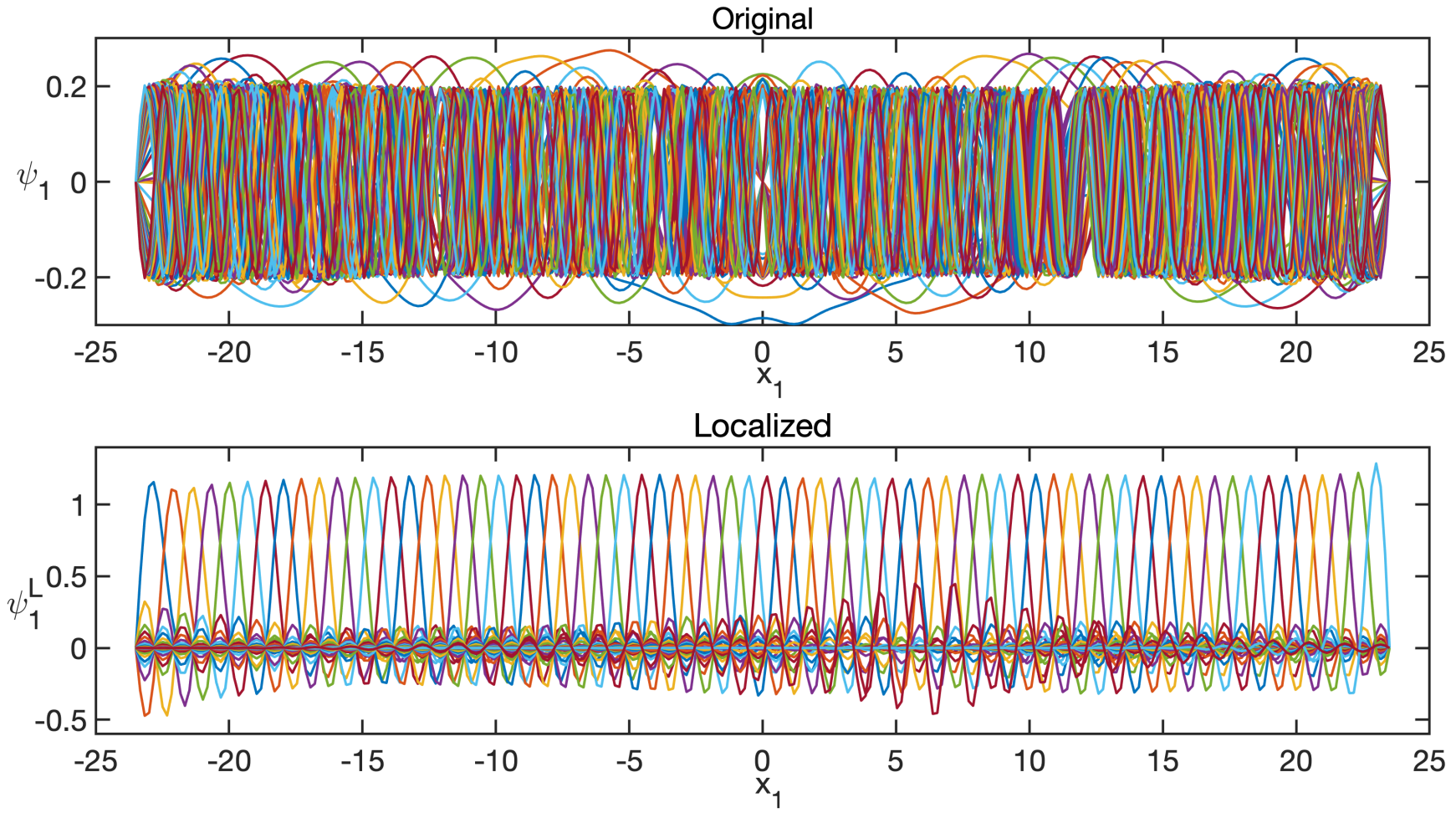


Figure 4.2: One-dimensional functions for 70 states of the separable Hamiltonian. one-dimensional functions in x_1 direction constructed from the additive separable approximation of the Kohn-Sham Hamiltonian for Al_{147} nano-partice. Top: Lowest 70 eigenfunctions of $\mathcal{H}_1$. Bottom: The corresponding L_1 localized one-dimensional functions.

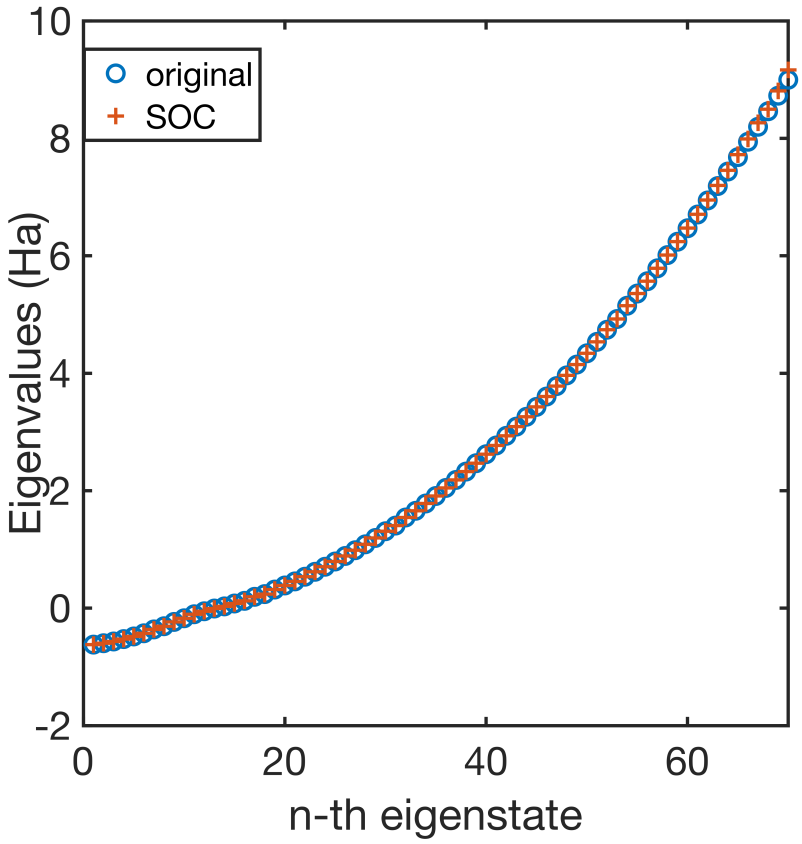


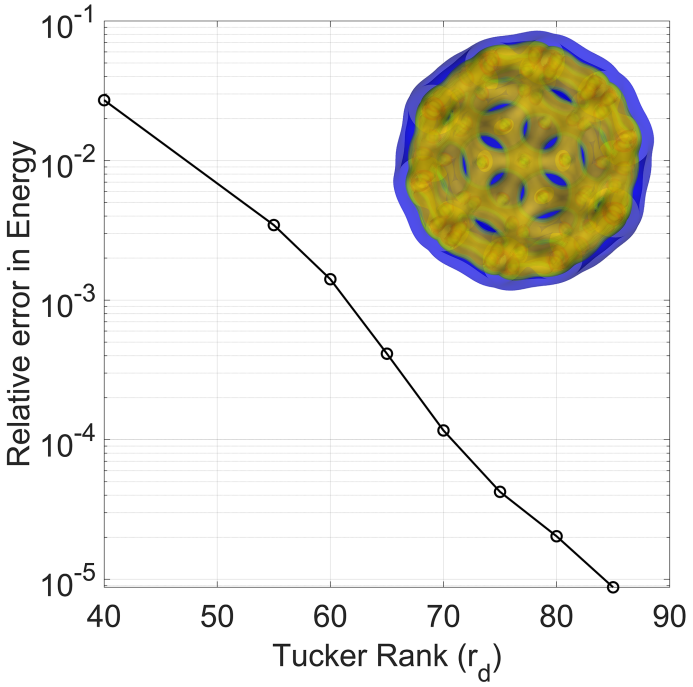
Figure 4.3: Representability of the one-dimensional localized functions. Comparison of the eigenvalues of the one-dimensional separable Hamiltonian in x_1 direction of Al_{147} nano-particle (marked with blue circle) with the eigenvalues of $K_{ij} = \left\langle \psi^{\mathrm{L}}_{1,i} \right| \mathcal{H}_k \left| \psi^{\mathrm{L}}_{1,j} \right\rangle$ (marked with red cross and labeld as SOC).

the convergence and accuracy afforded by the proposed Tucker tensor basis. The ground-state energy for these molecules is computed for various Tucker tensor ranks R ($R_1 = R_2 = R_3 = R$), with the three-dimensional Tucker tensor basis getting systematically refined with increasing R. In this study, R_ρ, R_{V} and $R_{\mathrm{V}}^{\mathrm{nl}}$ are chosen stringently such that the resulting errors are significantly smaller than the basis discretization errors, and they are held fixed for increasing R. In particular, we used $R_\rho = 45$, $R_{\mathrm{V}} = 65$ and $R_{\mathrm{V}}^{\mathrm{nl}} = 20$ for fullerene, and $R_\rho = 80$, $R_{\mathrm{V}} = 55$ and $R_{\mathrm{V}}^{\mathrm{nl}} = 20$ for tris (bipyridine) ruthenium. The basis discretization error (convergence with respect to R) is measured with respect to a well-converged Quantum Espresso result. The converged Quantum Espresso ground-state energies for fullerene molecule is taken to be -155.1248 eV per atom ($E_{\mathrm{cut}} = 60$ Ha) and that of tris (bipyridine) ruthenium is taken to be -118.2128 eV per atom ($E_{\mathrm{cut}} = 65$ Ha).

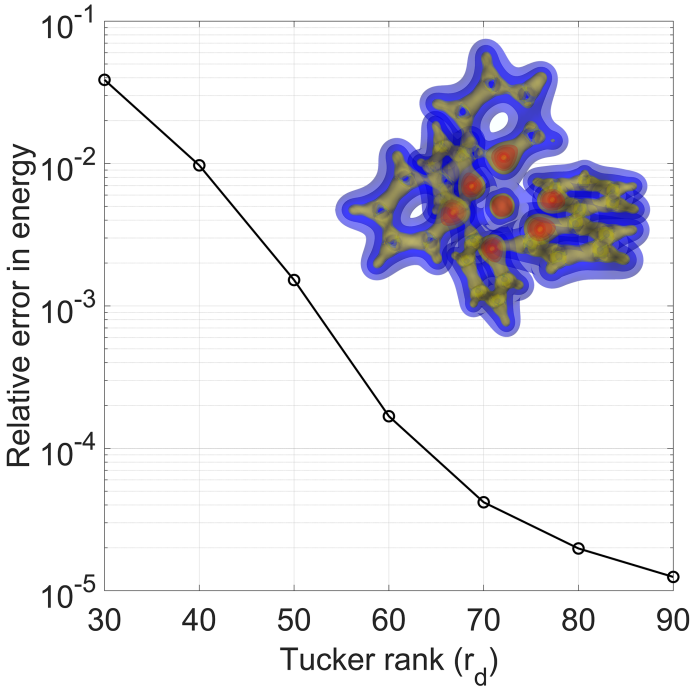
Figure 4.4 (a) and Fig. 4.4 (b) show the relative error in the ground-state energy for the various ranks of the Tucker tensor basis. It is evident from these results that the Tucker tensor basis constructed using our approach provides an exponential convergence in the ground-state energy with increasing Tucker rank. The convergence study of these molecules suggests that the proposed tensor-structured technique provides systematic convergence with high accuracy and is capable of handling generic materials systems, including those involving transition metals.

4.5 Performance benchmark

To study the performance and scaling with system-size of the proposed tensor-structured approach for DFT calculations, we consider two classes of benchmark systems: (i) Aluminum nano-particles of various sizes ranging from 13 atoms to 6,525 atoms; (ii) Silicon quantum dots with system-sizes ranging from 26 atoms to 7,355 atoms. These benchmark systems constitute materials systems with and without a gap, thus allowing us to assess the system-size scaling for both classes of materials.



(a) Fullerene



(b) Tris (bipyridine) ruthenium

Figure 4.4: Convergence with respect to the Tucker rank. The electron-density iso-contours are provided in the inset.

In order to compare the efficiency of the proposed tensor-structured approach with the widely used plane-wave DFT calculations, we also conducted the DFT calculations using Quantum Espresso wherever possible. For the sake of estimating the computational efficiency, the energy cut-off for Quantum Espresso and the Tucker rank are chosen such that the ground-state energy is converged to within 10 meV per atom measured with respect to a highly converged reference calculation. The reference ground-state energies are obtained from Quantum Espresso (using a high energy cut-off) for smaller systems and the DFT-FE code [26]—a massively parallel real-space code for large-scale DFT calculations—for larger system-sizes. The cell size for plane-wave calculations is chosen such that each atom is at least 10 Bohr away from the boundary, which was needed to obtain the desired accuracy.

In these benchmark calculations, the additive separable approximation to the Kohn-Sham Hamiltonian is computed only in the first SCF iteration and the resulting three-dimensional Tucker basis is held fixed for subsequent SCF iterations. We note that the approximation properties of an adaptive Tucker basis (where the basis is regenerated for every SCF iteration) and the fixed Tucker basis are similar, with the differences in the accuracy being substantially smaller than the basis discretization error for a given Tucker rank. We refer to Table 4.2 which provides data supporting this observation. In the tensor-structured calculations reported subsequently, all the numerical parameters—ranks for approximating electron-density and effective Kohn-Sham potential $(R_\rho, R_{\mathrm{V}}, R_{\mathrm{V}}^{\mathrm{nl}})$ in $\tilde{\mathbf{H}}^{\mathrm{L}}$, and the truncation tolerance adopted in computing $\mathbf{H}^{\mathrm{L}}$—are chosen such that the resulting errors are substantially smaller than the basis discretization error in the ground-state energy associated with the Tucker rank R of the localized three-dimensional Tucker tensor basis and the desired chemical accuracy. In particular, for the Aluminum nano-particles we used $R_\rho = 40$, $R_{\mathrm{V}} = 50$ and $R_{\mathrm{V}}^{\mathrm{nl}} = 25$. In the case of Silicon quantum dots, we used $R_\rho = 55$, $R_{\mathrm{V}} = 55$ and $R_{\mathrm{V}}^{\mathrm{nl}} = 25$. The truncation tolerance in computing $\mathbf{H}^{\mathrm{L}}$ was chosen to be

10^{-4} Ha for all the calculations, which provides excellent sparsity for $\mathbf{H}^{\mathrm{L}}$, and, importantly, the sparsity is either steady or improves with increasing system-size. The error in ground-state energy associated with this choice of truncation tolerance is ~ 1 meV per atom, as opposed to the targeted accuracy in this study of being within 10 meV per atom of reference energies.

4.5.1 Aluminum nano-particles

The computational efficiency afforded by the proposed tensor-structured approach in comparison to Quantum Espresso for the various aluminum nano-particles with icosahedral symmetry considered in this work is provided in Table. 4.3. We note that the Tucker rank required to achieve the desired accuracy only grows slowly with increasing system-size. Importantly, we note that the number of basis functions needed to achieve the desired accuracy using the localized Tucker basis is smaller than the plane-wave basis. As previously discussed, this is a consequence of the superior approximation properties of the Tucker tensor basis generated as the eigenbasis of an additive separable approximation of the Kohn-Sham Hamiltonian that in addition to the Laplace operator retains some characteristics of the Kohn-Sham potential, as opposed to the plane-wave basis which corresponds to the eigenbasis of the Laplace operator.

In terms of the computational time, while Quantum Espresso is more efficient for the smaller system-sizes, the tensor-structured approach starts to substantially outperform for larger system-sizes. Notably, for Al_{2057}, the tensor-structured approach is 8-fold more efficient.

Furthermore, using the computational times, the scaling of the proposed tensor-structured approach is estimated to be around $\mathcal{O}(N_{\mathrm{e}}^{1.78})$ with N_{e} denoting the number of electrons (cf. Fig. 4.5 a). Notably, the scaling with system-size is sub-quadratic for this metallic system over system-sizes spanning many thousands of atoms, as

Nano particle	Tucker				QE			Reference Energy/atom
	rank	# basis/ atom	E/atom	time	#basis/ atom	E/atom	time	
Al_{13}	40	4923	-55.9965	0.00067	12403	-55.9993	0.00022	-56.0034*
Al_{147}	70	2333	-56.6179	0.069	5078	-56.6198	0.028	-56.6274*
Al_{561}	85	1095	-56.8119	0.69	4490	-56.8122	1.24	-56.8191*
Al_{2057}	120	840	-56.9192	7.96	5015	N/A	66.68	-56.9284†
Al_{6525}	150	517	-57.0013	55.08	N/A	N/A	N/A	-57.0090†

Table 4.3: Comparison of the computational performance for Aluminum nano-particles. Comparison of the computational performance of the tensor-structured approach with Quantum Espresso (QE) for the benchmark aluminum nano-particles systems. All energies are reported in eV, and the computational times are reported in node-hrs per SCF iteration. The plane-wave cut-off employed for QE calculations to target the desired accuracy is 25 Ha. Full ground-state calculations were performed using the tensor-structured approach for all systems. In the case of QE, full-ground-state calculations were performed for the systems where the ground-state energies are provided, whereas for Al_{2057} only a few SCF iterations were performed to compute the stable SCF time due to significantly increased computational cost. Al_{6525} system was beyond reach using QE. The reference energies are computed using QE (*) with higher plane-wave cut-off 55 Ha for smaller systems. The reference energies for the larger systems are obtained using DFT-FE (†).

opposed to the cubic-scaling complexity for plane-wave DFT calculations. We note that this is a consequence of the slow growth of the Tucker rank with system-size that results in a sub-linear growth of the total number of basis functions with system-size. The breakdown of the computational costs for the various steps of the calculation is provided in Table 4.4.

4.5.2 Silicon quantum dots

Table 4.5 compares the computational performance of the proposed tensor-structured approach with Quantum Espresso for a wide range of silicon quantum dots passivated with hydrogen. As in the case of aluminum nano-particles, the Tucker tensor basis is more efficient than the plane-wave basis in terms of the number of basis functions to attain the desired accuracy. In terms of computational time, the proposed tensor-structured approach starts competing with Quantum Espresso beyond a few hundred

	# atoms	# e-	Density fraction	Time/scf (node-hrs)	ChF (node-hrs)	Orth (node-hrs)	Sub proj (node-hrs)	Others (node-hrs)
Al_{13}	13	39	2.54e-03	6.69e-04	2.65e-04	1.76e-04	4.41e-05	1.84e-04
Al_{147}	147	441	3.86e-03	6.86e-02	4.67e-02	3.89e-04	6.53e-03	1.50e-02
Al_{561}	561	1683	7.57e-04	0.693	0.443	0.012	0.052	0.186
Al_{2057}	2057	6171	8.16e-04	7.964	4.974	0.461	0.593	1.936
Al_{6525}	6525	19575	6.33e-04	55.077	32.606	12.681	4.108	5.682

Table 4.4: Breakdown of computational times for the various steps in the solution of the Kohn-Sham equations in the localized Tucker tensor basis using the Chebyshev filtering based subspace iteration. The benchmark system considered is aluminum nano-particles.

atoms, and significantly outperforms for larger systems. The breakdown of the computational costs for the various steps of the calculation is provided in Table 4.6.

Moreover, the scaling with system-size for the tensor-structured algorithm, for a range of system-sizes with the largest containing 7,355 atoms, is estimated to be $\mathcal{O}(N_e^{1.8})$ (cf. Fig. 4.5 b). Notably, this scaling is similar to that obtained for aluminum nano-particles as the algorithm treats systems with and without a gap on a similar footing.

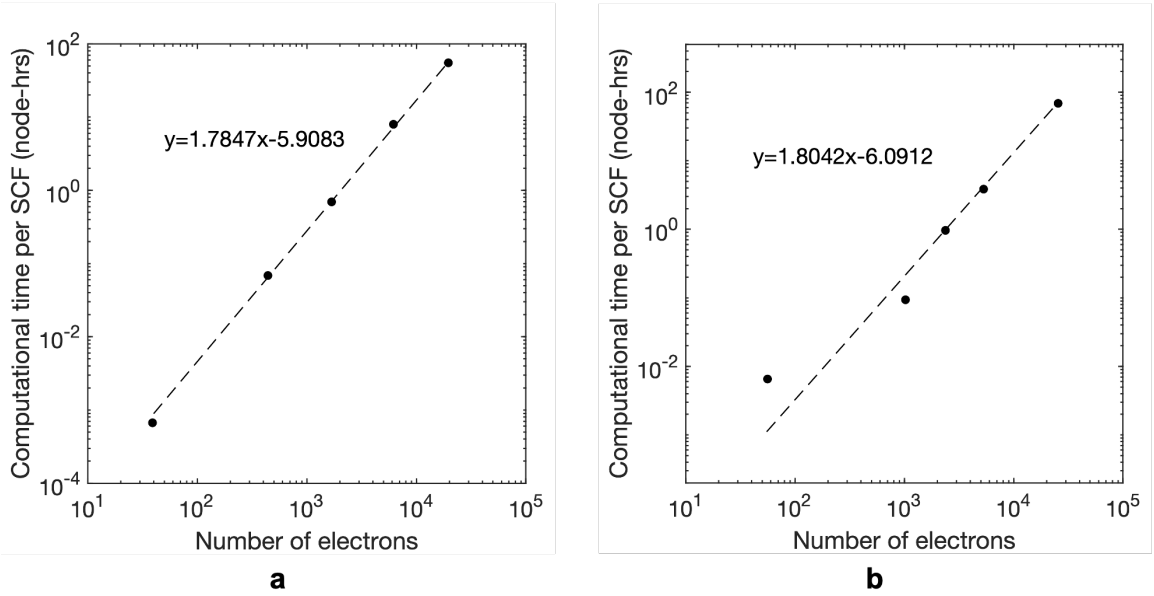


Figure 4.5: Computational complexity of the tensor-structured approach. (a) Aluminum nano-particles $\mathcal{O}(1.78)$; (b) Silicon quantum dots $\mathcal{O}(1.8)$.

Si quantum dots	Tucker				QE			Reference Energy/atom
	rank	# basis/ atom	E/atom	time	# basis/ atom	E/atom	time	
$Si_{10}H_{16}$	45	3505	-51.0271	0.0065	7048	-51.0279	0.00014	-51.0339*
$Si_{220}H_{144}$	80	1407	-71.3841	0.094	2534	-71.3839	0.096	-71.3930*
$Si_{525}H_{276}$	90	910	-76.1182	0.96	2251	-76.1194	1.12	-76.1279*
$Si_{1214}H_{504}$	100	582	-80.8627	3.85	2132	N/A	20.01	-80.8717†
$Si_{6047}H_{1308}$	140	373	-91.5659	67.49	N/A	N/A	N/A	-91.5741†

Table 4.5: Comparison of the computational performance for Silicon quantum dots. Comparison of the computational performance of the tensor-structured approach with Quantum Espresso (QE) for the benchmark Silicon quantum dots systems. All energies are reported in eV, and the computational times are reported in node-hrs per SCF iteration. The plane-wave cut-off employed for QE calculations to target the desired accuracy is 20 Ha. Full ground-state calculations were performed using the tensor-structured approach for all systems. In the case of QE, full-ground-state calculations were performed for the systems where the ground-state energies are provided, whereas for $Si_{1214}H_{504}$ only a few SCF iterations were performed to compute the stable SCF time due to significantly increased computational cost. $Si_{6047}H_{1308}$ system was beyond reach using QE. The reference energies are computed using QE (*) with higher plane-wave cut-off 50 Ha for smaller systems. The reference energies for the larger systems are obtained using DFT-FE (†).

	# atoms	# e-	Density fraction	Time/scf (node-hrs)	ChF (node-hrs)	Orth (node-hrs)	Sub proj (node-hrs)	Others (node-hrs)
$Si_{10}H_{16}$	26	56	7.38e-03	6.54e-03	1.67e-03	7.88e-04	2.04e-04	3.88e-03
$Si_{220}H_{144}$	364	1024	1.83e-04	9.37e-02	5.94e-02	2.26e-03	7.11e-03	2.49e-02
$Si_{525}H_{276}$	801	2376	6.58e-04	0.964	0.609	0.033	0.081	0.241
$Si_{1214}H_{504}$	1718	5360	4.16e-04	3.853	2.674	0.292	0.319	0.568
$Si_{6047}H_{1308}$	7355	25496	7.38e-04	67.492	43.211	13.800	5.579	4.902

Table 4.6: Breakdown of computational times for the various steps in the solution of the Kohn-Sham equations in the localized Tucker tensor basis using the Chebyshev filtering based subspace iteration. The benchmark system considered is silicon quantum dots.

4.6 Summary

In this chapter, we have presented a tensor-structured algorithm, where the Tucker tensor basis is constructed as a tensor product of localized one-dimensional functions whose span closely approximates the eigensubspace of a suitably constructed additive separable approximation to the Kohn-Sham Hamiltonian. The resulting localized Tucker tensor basis, that is adapted to the Kohn-Sham Hamiltonian, provides a systematically convergent basis as evidenced by the exponential convergence of the ground-state energy with increasing Tucker rank. Our numerical studies on the computational performance suggest that the proposed approach exhibits sub-quadratic scaling (with system-size) over a wide range of system-sizes with the largest involving many thousands of atoms. Importantly, sub-quadratic scaling is realized for both systems with and without a gap, as the algorithm treats both metallic and insulating systems on an equal footing. Further, comparing the computational efficiency of the proposed approach with Quantum Espresso, we observe significant outperformance for system-sizes beyond 5,000 electrons.

We note that the sub-quadratic scaling is a consequence of the slow growth of the Tucker rank with system-size, with the resulting number of basis functions growing sub-linearly with system-size even for systems containing many thousands of atoms. By combining the proposed approach with reduced-order scaling techniques that exploit the locality of the wavefunctions in real-space, there is further room to reduce the scaling with system-size and is a useful future direction to pursue. Further, the proposed tensor-structured approach is amenable to GPU acceleration that can further substantially enhance the computational efficiency of the approach, and will be presented in the next chapter. We note that the benchmark systems presented here were restricted to non-periodic calculations as the proposed tensor-structured approach was implemented in a non-periodic setting as a first step of an ongoing effort. However, we remark that the ideas presented here are generic and can be extended

to periodic calculations.

CHAPTER V

GPU Acceleration for Tensor-Structured Algorithm

In this chapter, we seek to improve the computational efficiency of the proposed tensor-structured algorithm for Kohn-Sham density functional calculation presented in the previous chapter. To this end, we utilize graphic processing unit (GPU) which provides extreme computing power, especially in linear algebra operation, to improve the efficiency of the matrix-matrix multiplication kernel that is the most computationally expensive part in the tensor-structured algorithm for Kohn-Sham DFT calculation. The structure of this chapter is as follows. An introduction to the motivation and implementation of this work is presented in Section 5.1. The idea and algorithm of the Chebyshev filtering subspace iteration method is presented in Section 5.2 to facilitate the later discussion. Various aspects regarding GPU acceleration including data layout and the algorithm are presented in Section 5.3. The benchmark comparison between pure CPU code and GPU accelerated code is provided in Section 5.4. Finally, we summarize this work in Section 5.5.

5.1 Introduction

In the solution of the Kohn-Sham equations using the localized Tucker tensor basis, the Chebyshev filtering step in each SCF iteration is the most computationally expensive step for even systems comprising of $\sim 10,000$ electrons. The main kernel in the Chebyshev filtering is the sparse-dense matrix-matrix multiplication, and a GPU acceleration of this kernel can result in substantial reductions in the wall-times of the DFT calculation.

To this end, the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ and the wavefunction matrix $\mathbf{X}$ are partitioned row-wise (in contrast to $\mathbf{\Psi}$, we use $\mathbf{X}$ to specify the (wavefunction) matrix against which the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ multiplies because essentially, $\mathbf{X}$ changes during the course of Chebyshev filtering process and need not be the true solution to the Kohn-Sham equation.) We note that this work takes advantage of band-parallelism to reduce the communication costs and improve parallel scalability, where a subset of wavefunctions are assigned to each group of MPI tasks via sub-communicators. Thus, each GPU owns multiple rows of the Hamiltonian matrix and the wavefunction matrix corresponding to the sub-group. The details of the data layout for the Hamiltonian matrix and the wavefunction matrix are elaborated in following sections. We also remark that the sparsity pattern of $\mathbf{H}^{\mathrm{L}}$ is such that the matrix has less sparsity around the diagonal, whereas the sparsity increases away from the diagonal. This structure is due to the spatial locality of the L_1 localized Tucker tensor basis. We take advantage of this structure to develop an efficient implementation of the matrix-matrix multiplication kernel in the Chebyshev filtering step. Further, we take advantage of the fact that $\mathbf{H}^{\mathrm{L}}$ is symmetric to reduce communication costs.

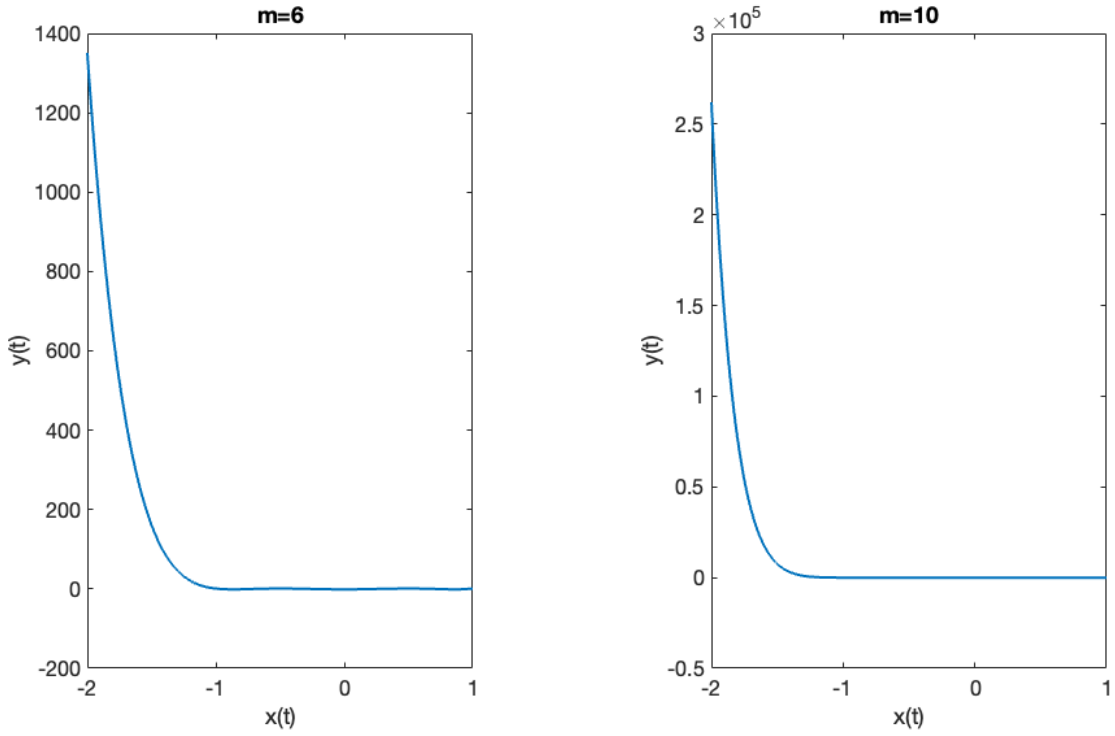
The remainder of this chapter will firstly introduce the Chebyshev filtering subspace iteration method and provide the skeleton to the algorithm for readers' reference. In the following section, we present our implementation of the various aspects

of the tensor-structured algorithm code for Kohn-Sham DFT calculation that have been GPU accelerated, which include: (ii) the details of the data layout for the Kohn-Sham Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ and the wavefunction matrix $\mathbf{X}$, (ii) the algorithm for matrix-matrix multiplication of $\mathbf{H}^{\mathrm{L}}\mathbf{X}$ based on GPU, and (iii) applying the matrix-matrix multiplication kernel for the subspace projection $\mathbf{H}_F^{\mathrm{L}} = \mathbf{X}_F^{\mathrm{T}}\mathbf{H}^{\mathrm{L}}\mathbf{X}_F$. Finally, we will present the computational benchmark and summarize this work.

5.2 Chebyshev filtered subspace iterative (ChFSI) method

The standard eigenvalue problem in Eq. 4.43 is solved using the Chebyshev filtering subspace iteration (ChFSI) method [165]. The ChFSI method has been demonstrated to be an effective method for large-scale real-space DFT calculations [26, 3]. In every SCF iteration, the ChFSI method seeks to compute a good approximation to the subspace spanned by the occupied states of the Kohn-Sham Hamiltonian. This is realized by taking advantage of the property of Chebyshev polynomials that are bounded in the interval $[-1, 1]$, but grow rapidly outside this interval. As shown in the plot of m degree Chebyshev polynomial in Fig. 5.1, the values out of the bounded interval ($[-1, 1]$) are amplified rapidly, while the values inside the bounded interval are nicely dampened. We also not that as the Chebyshev polynomial degree increases from $m = 6$ to $m = 10$, the values out of the bounded interval are much more amplified as shown in the y-axis of Fig. 5.1 (a) and Fig. 5.1 (b). For the sake of completeness, we remark that, while neither used nor necessary in this work, we direct readers to [166] for an algorithm to determine the Chebyshev polynomial degree m when the search space of the Chebyshev polynomial degree m is large and the cost for determining a good m is high.

To utilize this nice rapidly growing-dampening property of Chebyshev polynomials to compute the occupied states of the Kohn-Sham Hamiltonian, the discrete Kohn-Sham Hamiltonian is scaled and shifted such that the unwanted spectrum maps to



(a) Chebyshev polynomial degree m = 6 (b) Chebyshev polynomial degree m = 10

Figure 5.1: Chebyshev polynomial with the polynomial degree $m = 6$ and $m = 10$.

$[-1, 1]$ and the desired spectrum of the occupied and partially occupied states maps to $(-\infty, -1)$. Thus, the application of a Chebyshev polynomial filter, constructed from the scaled-and-shifted Hamiltonian, on a set of vectors provides a subspace that is a close approximation to the desired occupied eigenspace. The Chebyshev filtered vectors are orthogonalized using the Gram-Schmidt orthogonalization procedure, and the Kohn-Sham eigenvalue problem (Eq. 4.36) is solved by projecting the problem onto the Chebyshev filtered subspace.

Algorithm 4: ChFSI [164]

Input: $\mathbf{H}^{\mathrm{L}}$, $\mathbf{X}$, m, ϵ_0, ϵ_{ub}^{w}, ϵ_{ub}^{uw}
Output: $\mathbf{\Psi}$, diag $(\mathbf{\Lambda})$
1. Chebyshev filtering process
   Initialize:
   $e = \frac{1}{2}\left(\epsilon_{ub}^{uw} - \epsilon_{ub}^{w}\right)$; $c = \frac{1}{2}\left(\epsilon_{ub}^{uw} + \epsilon_{ub}^{w}\right)$; $\sigma = \frac{e}{\epsilon_0 - c}$
   $\sigma_1 = \sigma$; $\gamma = \frac{2}{\sigma_1}$; $\tilde{\mathbf{X}} = \frac{\sigma_1}{e}\left(\mathbf{H}^{\mathrm{L}}\mathbf{X} - c\mathbf{X}\right)$;
   for $i = 2 : m$
      $\sigma_2 = \frac{1}{\gamma - \sigma}$;
      $\tilde{\mathbf{X}}_{new} = \frac{2\sigma_2}{e}\left(\mathbf{H}^{\mathrm{L}}\tilde{\mathbf{X}} - c\tilde{\mathbf{X}}\right) - \sigma\sigma_2\mathbf{X}$;
      $\mathbf{X} = \tilde{\mathbf{X}}$;
      $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}_{new}$;
      $\sigma = \sigma_2$;
   end for
2. Orthonormalize the Chebyshev filtered basis functions, and denote by $\mathbf{X}_F = \mathrm{Orth}(\mathbf{X})$
3. Perform subspace projection: $\mathbf{H}_F^{\mathrm{L}} = \mathbf{X}_F^{\mathrm{T}}\mathbf{H}^{\mathrm{L}}\mathbf{X}_F$
4. Diagonalize $\mathbf{H}_F^{\mathrm{L}}$ with eigen-decomposition $\mathbf{H}_F^{\mathrm{L}}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}$
5. Rotate the basis $\mathbf{\Psi} = \mathbf{X}\mathbf{Q}$

The ChFSI method is outlined in Algorithm 4. In Algorithm 4, m denotes the Chebyshev polynomial degree; ϵ_0 and ϵ_{ub}^{w} are the lower and upper bound of the wanted spectrum, respectively; ϵ_{ub}^{uw} is the upper bound of the unwanted spectrum; $\mathbf{X}$ is the input wavefunction matrix; $\mathbf{\Psi}$ is the output wavefunction.

We note that Algorithm 4 is a scaled version of the Chebyshev filtering subspace iteration method. To elaborate, given the iterative sequence in a non-scaled Cheby-

shev polynomial subspace iteration procedure

$$\tilde{\mathbf{X}}_{new} = \frac{2}{e}\left(\mathbf{H}^{\mathrm{L}}\tilde{\mathbf{X}} - c\tilde{\mathbf{X}}\right) - \mathbf{X}\,,$$

we can recast the above expression as a matrix

$$\begin{pmatrix} \tilde{\mathbf{X}}_{new} \\ \\ \tilde{\mathbf{X}} \end{pmatrix} = \begin{pmatrix} \frac{2}{e}\left(\mathbf{H}^{\mathrm{L}} - c\mathbf{I}\right) & -\mathbf{I} \\ \\ \mathbf{I} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{X}} \\ \\ \mathbf{X} \end{pmatrix}.$$

It can be shown that the eigenvalues for the middle square matrix are complex with modulus one except those corresponding to the eigenvalues of $\mathbf{H}^{\mathrm{L}}$ that is smaller than the upper bound of the wanted spectrum [164]. During the course of the Chebyshev filtering process, the eigenvalues of the Hamiltonian $\mathbf{H}^{\mathrm{L}}$ within the wanted spectrum are mapped to greater than one eigenvalues (in magnitude) in $\frac{2}{e}\left(\mathbf{H}^{\mathrm{L}} - c\mathbf{I}\right)$. Hence, similar to a standard power iteration, it can readily be seen that the largest eigenvalue in $\frac{2}{e}\left(\mathbf{H}^{\mathrm{L}} - c\mathbf{I}\right)$ grows rapidly and could hence increase the condition number of the matrix and subsequently influence the stability of the algorithm. To this end, following the suggestion from the work by Zhou *et al.* [164], we use the lower bound of the wanted spectrum to introduce a further scaling to prevent $\mathbf{X}$ from overflowing during the Chebyshev filtering process. We also refer to [164, 165, 167] for more complete discussion.

Finally, we remark that, in the first SCF iteration, $\mathbf{X}$ is typically set to either a random full-rank matrix or represented by atomic orbitals, and the Chebyshev filtering is performed using higher polynomial degree m [168]. In the subsequent iterations, $\mathbf{X}$ is set to be the resultant $\boldsymbol{\Psi}$ from the previous SCF iteration, which provides a good guess and thus does not need a large m. For the various benchmark systems studied in this work, m is chosen to be $10 - 20$.

5.3 Data layout for GPU acceleration

In Algorithm 4, the most computationally expensive part lies in the matrix-matrix multiplication when repeatedly applying the Chebyshev polynomial to the wavefunction matrix. To this end, we port this calculation to GPU to improve the computational performance. In this section, we will show the data layout for the matrices and the algorithm for performing the matrix-matrix multiplication.

5.3.1 Data layout for $\mathbf{H}^{\mathrm{L}}$ and $\mathbf{X}$

Figure 5.2 provides a schematic of the data layout of the sparse Kohn-Sham Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ in the localized Tucker tensor basis. The ownership of the rows of the Hamiltonian matrix is distributed as evenly as possible so that each GPU shares similar working load. Particularly, given the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ of size $M \times M$, the matrix is distributed across N GPUs labeled from 0 to $N-1$ as shown in Fig. 5.2. Let τ be the quotient of M divided by N, then block of the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ residing on the k-th GPU owns the $k\tau$-th to the $((k+1)\tau-1)$-th rows of the Hamiltonian matrix, and is of size $\tau \times M$. In the case that M is not divisible by N, and ν be the remainder of M divided by N, the local block of the Hamiltonian matrix of the first ν GPUs are adjusted to be of size $(\tau+1) \times M$.

We remark that the sparsity pattern of $\mathbf{H}^{\mathrm{L}}$ is such that most of the non-zero entries of the matrix are concentrated on and around the diagonal of the matrix, owing to the spatial locality of the L_1 localized Tucker tensor basis. Thus, in a tiling of the matrix, the diagonal blocks are much denser compared to the off-diagonal blocks. Thus, the non-zero terms in the diagonal blocks could easily exceed 5%, which is the suggested minimal sparsity for sparse algorithm to be efficient [169], and deteriorate the overall performance. To this end, the diagonal blocks and the off-diagonal blocks of the Hamiltonian matrix are stored as dense and sparse matrices, respectively. The proposed data layout for the row-wise partitioned matrix on the

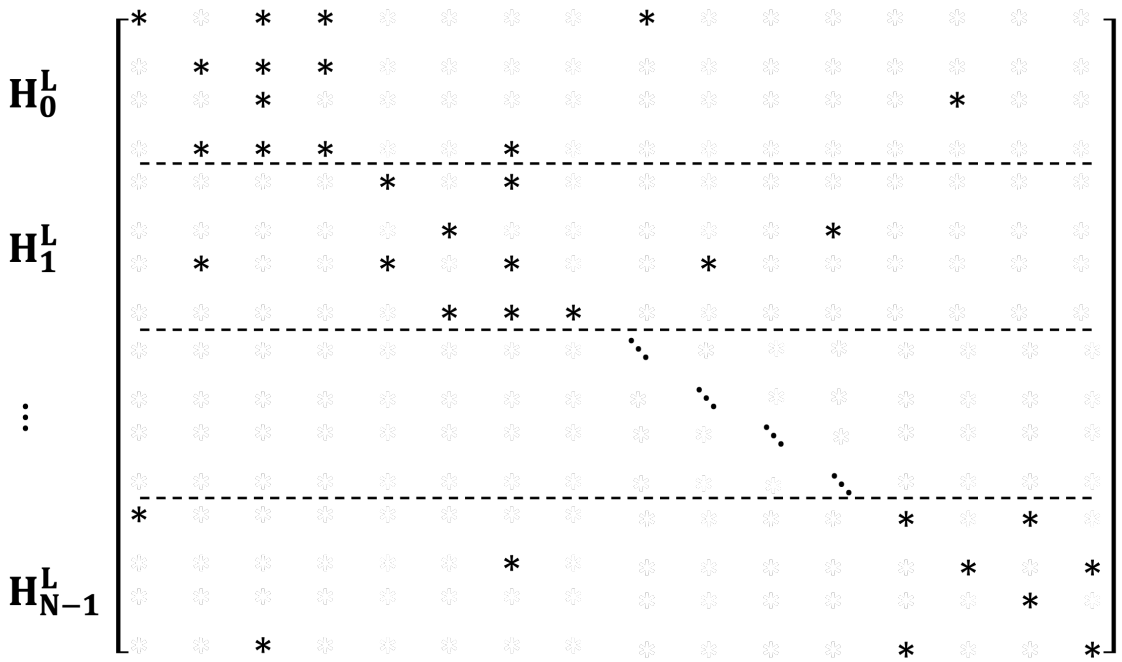


Figure 5.2: Schematic of the distribution of the projected Hamiltonian $\mathbf{H}^{\mathrm{L}}$ on each GPU. The $\mathbf{H}_k^{\mathrm{L}}$ $(k = 0, 1, \ldots N-1)$ partition of the projected Kohn-Sham Hamiltonian is assigned to the k-th GPU.

0-th GPU is illustrated in Fig. 5.3. The diagonal dense square matrix part of $\mathbf{H}_k^{\mathrm{L}}$ is denoted as $\mathbf{H}_k^{\mathrm{L(D)}}$ and the off-diagonal sparse matrix is denoted as $\mathbf{H}_k^{\mathrm{L(OD)}}$. The two parts of $\mathbf{H}_k^{\mathrm{L}}$ will then be treated using dense and sparse linear algebra library for the matrix-matrix multiplication kernel, respectively. We note that the number of rows owned by each GPU is chosen to be ∼30,000 in the current implementation so that the density of the diagonal block of the matrix exceeds 5%, yet fits in the GPU memory. Above the 5% threshold, the dense algorithm is generally considered to outperform the sparse algorithm.

The wavefunction matrix $\mathbf{X}$ is of size $M \times N_{\mathrm{orb}}$, where N_{orb} is the number of computed Kohn-Sham orbitals. Owing to double occupancy of the orbitals for spin-independent Hamiltonian, N_{orb} is usually chosen to be slightly larger than $N_e/2$, typically ∼10-15% larger. The rows of the wavefunctions are distributed consistently with the row-ownership of the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$. We note that during the computation of $\mathbf{X}' = \mathbf{H}^{\mathrm{L}}\mathbf{X}$, regardless of the implementation, collective communica-

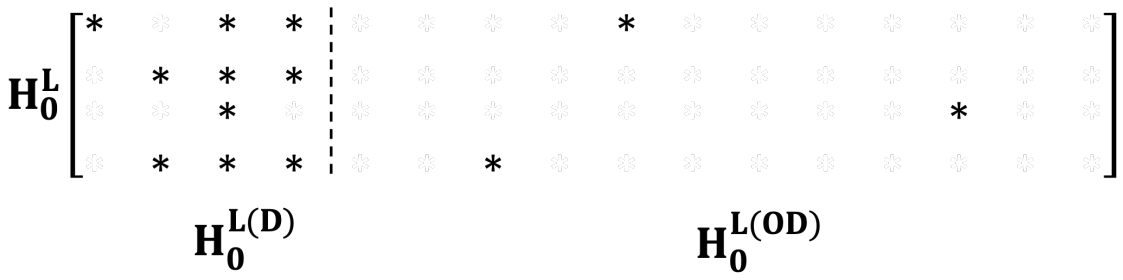


Figure 5.3: Schematic of the data layout of the row-wise partitioned Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ on the 0-th GPU.

tion over either $\mathbf{X}$ or $\mathbf{X}'$ will be needed. The cost for the collective communication is proportional to the number of processors and the data to be communicated within the (sub-)communicator [170, 171]. In a GPU calculation, this also requires data to be transferred from the device memory to the host memory and communication to other processors. Thus, this step will substantially increase the communication cost and deteriorate the overall performance. It is thus desirable to reduce this communication cost. To this end, in addition to the row-wise parallelization, columns of the wavefunction matrix $\mathbf{X}$ are further partitioned into groups (bands) labeled as $G_{p=0\ldots P-1}$, and this is referred to as band parallelization henceforth in keeping with the nomenclature of DFT literature. In the present implementation, the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ is stored on each GPU group. Hence, each group will perform the matrix-matrix multiplication corresponding to the band of wavefunctions, and the collective communication after the matrix-matrix multiplication is only within the processors in the group. The number of processors to be communicated will thus be reduced by a factor P by using band-parallelism. Thus, the communication burden is significantly alleviated, and the overall performance of the Chebyshev filtering step is improved. A schematic illustration of the data layout for the wavefunction matrix $\mathbf{X}$ is provided in Fig. 5.4, where $\mathbf{X}_k$ is the portion of the wavefunction matrix having the same row-ownership of the Hamiltonian matrix $\mathbf{H}_k^{\mathrm{L}}$ in Fig. 5.2. $\mathbf{X}_k^{G_i}$ is the portion of the wavefunction matrix $\mathbf{X}_k$ belonging to the G_i processor group. The data layout

for the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}$ and the wavefunction matrix $\mathbf{X}$ are then used to implement the sparse-dense matrix-matrix multiplication kernel, which is subsequently discussed.

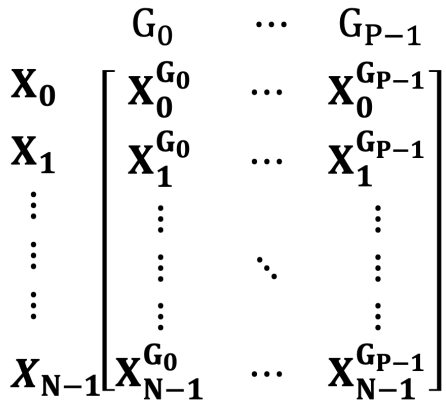


Figure 5.4: Schematic of the data layout of the wavefunction matrix $\mathbf{X}$.

5.3.2 $\mathbf{H}^{\mathrm{L}} \times \mathbf{X}$ implementation

As noted previously, the Hamiltonian matrix is distributed row-wisely across GPUs in each group G_i. In the matrix-matrix multiplication, each $\mathbf{H}^{\mathrm{L}}_k \times \mathbf{X}^{G_i}$, where $\mathbf{X}^{G_i}$ of size $M \times \frac{N_{\mathrm{orb}}}{P}$ is the collection of all $\mathbf{X}^{G_i}_k$ $(k = 0, 1, \ldots, N-1)$, yields $(\mathbf{H}^{\mathrm{L}}\mathbf{X})^{G_i}_k$ on the k-th GPU in the group G_i. The evaluation of each $(\mathbf{H}^{\mathrm{L}}\mathbf{X})^{G_i}_k$ requires communication of the off-diagonal block of the Hamiltonian matrix $\mathbf{H}^{\mathrm{L(OD)}}_k$ to all processors other than k, as well as collecting information back from those processors. This communication also involves data transfer between the host and the device memory of GPUs, and can severely diminish the performance of the sparse-dense matrix-matrix multiplication kernel.

In order to avoid the aforementioned communication of $\mathbf{H}^{\mathrm{L(OD)}}_k$, the matrix-matrix multiplication kernel is recast by taking advantage of the symmetric nature of $\mathbf{H}^{\mathrm{L}}$. We note that as $\mathbf{H}^{\mathrm{L}}$ is real and symmetric,

$$\mathbf{H}^{\mathrm{L}}_{(:,a:b)} = \left(\mathbf{H}^{\mathrm{L}}_{(a:b,:)}\right)^{\mathrm{T}}. \tag{5.1}$$

Eq. 5.1 states that a column block of the Hamiltonian matrix $\mathbf{H}^{\mathrm{L}}_{(:,a:b)}$, which is a matrix containing the a-th to b-th columns of $\mathbf{H}^{\mathrm{L}}$, is equivalent to the transpose of a row block $\mathbf{H}^{\mathrm{L}}_{(a:b,:)}$ comprising the a-th to b-th rows of $\mathbf{H}^{\mathrm{L}}$. Further, we note that the evaluation of a matrix-matrix product $\mathbf{C} = \mathbf{AB}$, where A and B are $m \times n$ and $n \times m$ matrices, is given by $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$ with a_{ij}, b_{ij} and c_{ij} denoting the matrix elements of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. The expression $\sum_{k=1}^{n} a_{ik} b_{kj}$ can also be viewed as a summation over k of the outer product of the k-th column vector of $\mathbf{A}$ with the k-th row vector of $\mathbf{B}$. Thus, using this interpretation of the matrix-matrix multiplication as a sum of the outer product of column and row vectors of the constituent matrices, we can write

$$
\begin{aligned}
\mathbf{H}^{\mathrm{L}}\mathbf{X} &= \sum_{k=0}^{N-1} \sum_{\xi=a:=k\tau}^{b:=(k+1)\tau-1} \mathbf{H}^{\mathrm{L}}_{(:,\xi)} \mathbf{X}_{(\xi,:)} \\
&:= \sum_{k=0}^{N-1} \mathbf{H}^{\mathrm{L}}_{(:,a:b)} \mathbf{X}_{(a:b,:)} \\
&= \sum_{k=0}^{N-1} \left(\mathbf{H}^{\mathrm{L}}_{(a:b,:)} \right)^{\mathrm{T}} \mathbf{X}_{(a:b,:)} \\
&= \sum_{k=0}^{N-1} \left(\mathbf{H}^{\mathrm{L}}_{k} \right)^{\mathrm{T}} \mathbf{X}_{k} \,.
\end{aligned}
\tag{5.2}
$$

In the above, $\tau = \frac{M}{N}$ follows the definition in Sec. 5.3.1, $\mathbf{H}^{\mathrm{L}}_{k}$ and $\mathbf{X}_{k}$ follow the notation in Fig. 5.2 and Fig. 5.4. A schematic for $\left(\mathbf{H}^{\mathrm{L}}_{k} \right)^{\mathrm{T}} \mathbf{X}_{k}$ on the 0-th GPU is illustrated in Fig. 5.5. As shown in the figure, the multiplication involves $(\mathbf{H}^{\mathrm{L}}_{0})^{\mathrm{T}}$ of size $M \times \tau$ and $\mathbf{X}^{G_i}_{0}$ of size $\tau \times \frac{N_{\mathrm{orb}}}{P}$, resulting in matrix $(\mathbf{H}^{\mathrm{L}}_{0})^{\mathrm{T}} \mathbf{X}^{G_i}_{0}$ of size $M \times \frac{N_{\mathrm{orb}}}{P}$. The final outcome $\mathbf{H}^{\mathrm{L}}\mathbf{X}^{G_i}$ can then be obtained by summing over k using `Allreduce` communication with MPI, as evident from the last equality of Eq. 5.2. We note that both $(\mathbf{H}^{\mathrm{L}}_{k})^{\mathrm{T}}$ and $\mathbf{X}_{k}$ are locally stored on the k-th GPU. Thus, for each matrix-matrix multiplication call during the Chebyshev filtering step, this approach avoids the MPI communications and overheads associated with transferring data between the device and the host memory for the off-diagonal block of the Hamiltonian matrix.

To understand the improvement in the efficiency by avoiding communicating the

off-diagonal block of the Hamiltonian matrix, we present a performance comparison by computing $\mathbf{H}^{\mathrm{L}}\mathbf{X}$ using the proposed algorithm and using the method with off-diagonal block communication (henceforth referred to as the general method). On the i-th processor, the general method is implemented by sending the off-diagonal blocks of the Hamiltonian matrix to all j-th ($j \neq i$) processors whose row-ownership coincide with the columns of the off-diagonal blocks. The off-diagonal block is then multiplied by the locally owned block of wavefunction matrix on the j-th processor and the result is reduced back to the i-th processor. To ensure the representability of this comparison, we choose the Hamiltonian matrix of Al_{147}, the benchmark system used for performance analysis in the later sections, to run this calculation. This benchmark calculation is run on the GreatLakes HPC cluster with each node comprising 2 Intel Xeon Gold 6148 CPUs with 40 physical cores per node. In this numerical experiment, the general method takes 481.92 cpu-secs and our proposed approach takes 232.18 cpu-secs. The $\sim 2\times$ improvement resulting from the communication efficiency, validates the use of the proposed approach.

Next, we turn our attention to leveraging the sparsity structure of $\mathbf{H}^{\mathrm{L}}$ to further optimize the matrix-matrix multiplication kernel. As we noted earlier, the density of the diagonal square block $\mathbf{H}_k^{\mathrm{L(D)}}$ can be large making sparse linear algebra operations inefficient [169]. To this end, we use different linear algebra libraries to treat the dense and the sparse blocks of the Hamiltonian matrix separately. As shown in Fig. 5.3, the diagonal blocks $\mathbf{H}_k^{\mathrm{L(D)}}$ are stored as a dense matrix and the off-diagonal blocks $\mathbf{H}^{\mathrm{L(OD)}}$ are stored as a sparse matrix. Further, the matrix-matrix multiplication operation depicted in Fig. 5.5 is further split into a dense-dense multiplication for the diagonal block $\left(\left(\mathbf{H}_k^{\mathrm{L(D)}}\right)^{\mathrm{T}}\mathbf{X}_k^{G_i}\right)$ and a sparse-dense multiplication for the off-diagonal block $\left(\left(\mathbf{H}_k^{\mathrm{L(OD)}}\right)^{\mathrm{T}}\mathbf{X}_k^{G_i}\right)$. The dense-dense multiplication kernel is implemented using `cublasDgemm` provided by `cuBLAS` [172], NVIDIA GPU-accelerated implementation for basic linear algebra subroutines (BLAS). On the other hand, the sparse-dense

multiplication kernel is computed with `cusparseDcsrmm` provided by `cuSPARSE` [169], NVIDIA GPU-accelerated implementation for sparse basic linear algebra subroutines. The two libraries are available in `CUDA Toolkit` [173] or NVIDIA High performance computing software development kit (`NVIDA HPC SDK` [174]). Once the computation is completed for the dense-dense and the sparse-dense matrix multiplication, the resultant matrices are assembled as $\left(\mathbf{H}_k^{\mathrm{L}}\right)^{\mathrm{T}} \mathbf{X}_k^{G_i}$ (see Fig. 5.5 for a schematic plot for the 0-th processor). The assembled matrix $\left(\mathbf{H}_k^{\mathrm{L}}\right)^{\mathrm{T}} \mathbf{X}_k^{G_i}$ is transferred back to the host memory. On the host memory, summation over k in Eq. 5.2 is completed using `MPI_Allreduce` within the wavefunction group.

5.3.3 Subspace projection: Evaluation of $\mathbf{H}_F^{\mathrm{L}}$

In the Chebyshev filtering subspace iteration algorithm, upon computing the Chebyshev filtered vectors that represent a close approximation to the eigen-subspace of interest, the Kohn-Sham eigenvalue problem is projected onto the Chebyshev filtered subspace to solve the eigenvalue problem in this subspace. This entails the evaluation of $\mathbf{H}_F^{\mathrm{L}} = \mathbf{X}_F^{\mathrm{T}} \mathbf{H}^{\mathrm{L}} \mathbf{X}_F$ (step 3 in Algorithm 4), where $\mathbf{X}_F$ is comprised of the orthonormalized Chebyshev filtered vectors. The evaluation of $\mathbf{H}_F^{\mathrm{L}}$ includes a matrix-matrix multiplication between $\mathbf{H}^{\mathrm{L}}$ and $\mathbf{X}_F$. Thus, it is natural to adopt the strategy discussed in Sec. 5.3.2 in evaluating $\mathbf{H}^{\mathrm{L}} \mathbf{X}_F$. Upon evaluating $\mathbf{H}^{\mathrm{L}} \mathbf{X}_F$, this matrix is transferred back to the host memory and left-multiplied with $\mathbf{X}^{\mathrm{T}}$ using MPI-based matrix-matrix multiplication kernel from `PETSc` library [175, 176, 177].

5.4 Results

The systematic convergence, accuracy, and efficacy of the Tucker tensor basis and the tensor-structured algorithm for DFT calculations have been established in prior works [45, 46]. In particular, it was demonstrated that the Tucker tensor basis was systematically improvable and the basis discretization error decreased exponentially

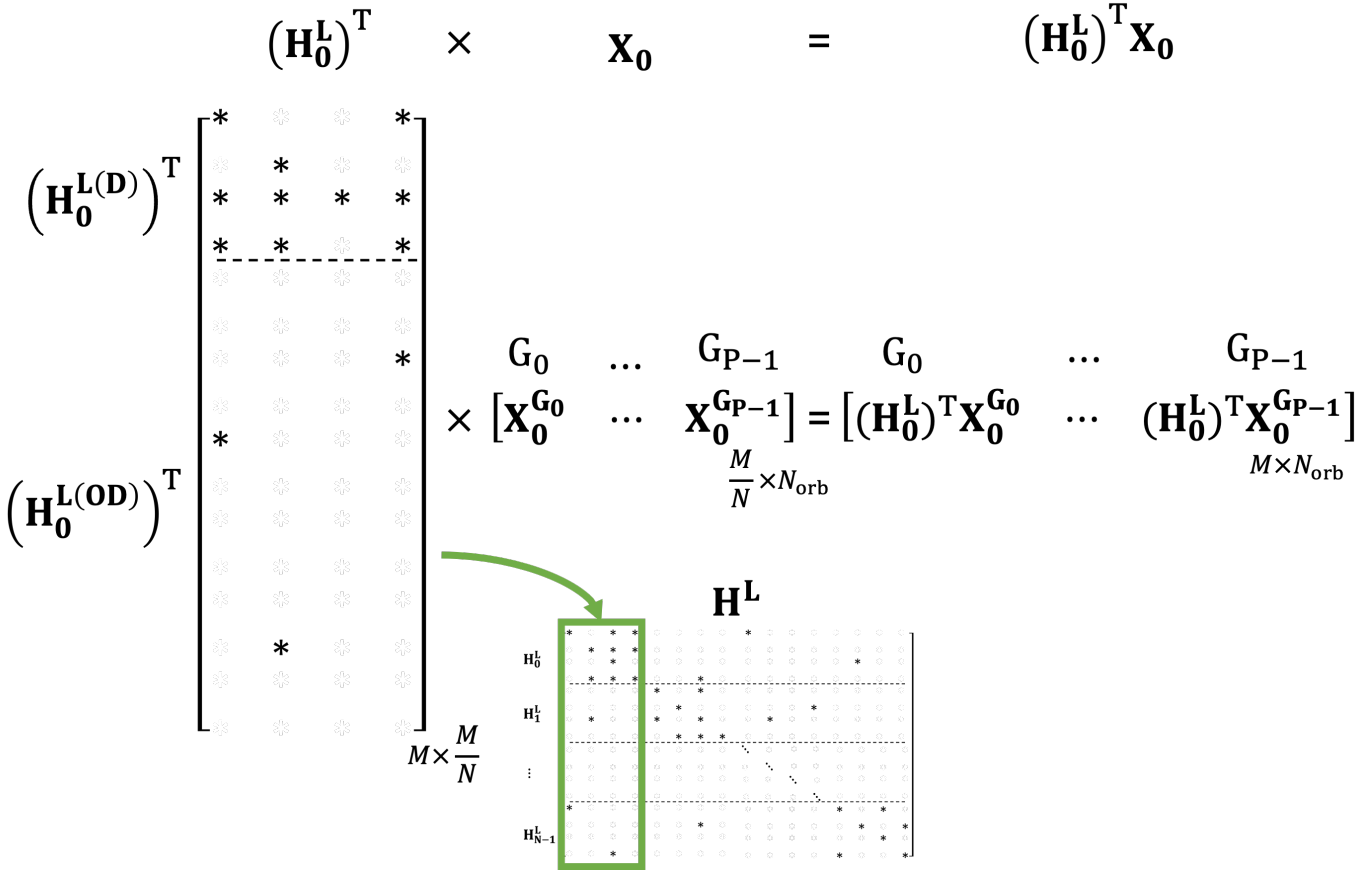


Figure 5.5: Schematic of the computation on 0-th processor in the evaluation of $\mathbf{H}^{\mathrm{L}}\mathbf{X}$. As $\mathbf{H}^{\mathrm{L}}$ is symmetric, the transpose of $\mathbf{H}_0^{\mathrm{L}}$ is equivalent to the columns of $\mathbf{H}^{\mathrm{L}}$ enclosed by the highligted box. $(\mathbf{H}_0^{\mathrm{L(D)}})^{\mathrm{T}}$ and $(\mathbf{H}_0^{\mathrm{L(OD)}})^{\mathrm{T}}$ denote the dense diagonal square block and the sparse off-diagonal part of the $(\mathbf{H}_0^{\mathrm{L}})^{\mathrm{T}}$ matrix, respectively.

with increasing Tucker rank [45, 46], thus providing spectral convergence similar to plane-wave discretization. We refer to [46] for a comprehensive numerical study of the approximation properties of the localized Tucker tensor basis in DFT calculations. Further, a comparative study of the computational efficiency of the localized Tucker tensor basis with a plane-wave basis has revealed that the Tucker tensor basis is not only more efficient in terms of the number of basis functions required to achieve chemical accuracy, but also provides significant computational savings owing to the reduced-order scaling with system size. Benchmark calculations on both systems with and without a gap have revealed that the solution of the DFT problem in the Tucker tensor basis is substantially more efficient than the plane-wave basis for systems beyond 2,000 electrons, with up to 8× improvement in computational efficiency (measured in node-hrs) over plane-wave calculations conducted using Quantum Espresso (cf. [46]).

In the present work, we focus on optimizing the most computationally expensive part of the calculation—the repetitive matrix-matrix multiplication kernel called during the Chebyshev filtering step—and further using GPU acceleration to improve the computational efficiency of the calculations. In order to assess the optimization realized, we use the benchmark systems from our previous work [46] comprising of aluminum nano-particles and silicon quantum dots of various sizes. The aluminum nano-particles ranging from Al_{13} to Al_{6525} are constructed using icosahedral symmetry. The silicon quantum dots are constructed by rounding the diamond-structured silicon crystal and passivating the surface with hydrogen atoms. The silicon quantum-dots considered here range from $Si_{10}H_{16}$ to $Si_{6047}H_{1308}$. Ball and stick models for the two smallest clusters of both systems are depicted in Fig. 5.6 and Fig. 5.7. In order to conduct a performance analysis, we ran the benchmark calculations by solely using CPUs and compared with the acceleration obtained by utilizing GPUs for the matrix-matrix multiplication kernel in the Chebyshev filtering and subspace projection steps.

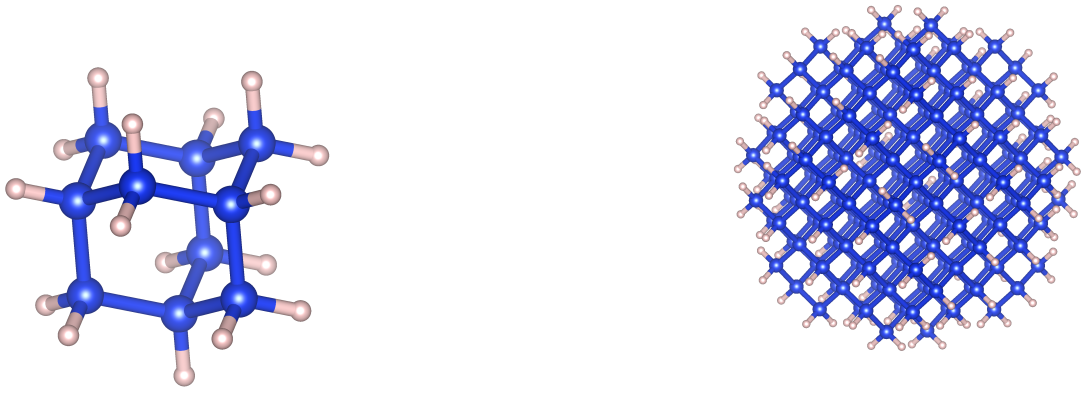
Further, to ensure the accuracy of the code with GPU acceleration, we compare the ground-state energies obtained via CPU-only and CPU-GPU calculations of the two smallest clusters for both systems.

The numerical parameters used in the present study follow the previously converged CPU-based calculations of the benchmark systems [46]. In this work, we use local density approximation (LDA) for exchange-correlation functional [76, 79, 78] and a norm-conserving Troullier-Martin pseudopotential in Kleinmann-Bylander form [157, 158]. The Fermi-Dirac smearing temperature is set to $T = 500K$ for computing the fractional occupancy of the orbitals. The Chebyshev polynomial degree is chosen to be $10 - 20$ for various materials systems. The Tucker decomposition ranks [46] in the evaluation of the Hartree potential (R_H), in the representation of local part of the effective Kohn-Sham potential (R_V) and the non-local part of the effective potential (R_V^{nl}) are chosen to be $R_H = 40$, $R_V = 50$, $R_V^{nl} = 25$ for all aluminum nano-particles system and $R_H = 55$, $R_V = 55$ and $R_V^{nl} = 25$ for all silicon quantum dots system. The prescribed truncation tolerance for the Kohn-Sham Hamiltonian is set to 10^{-4} Ha for both aluminum nano-particles and silicon quantum dots according to the previous error analysis in [46]. The numerical parameters used are consistent for both the CPU- and GPU-based calculations in the performance analysis.

The performance benchmarks have been conducted on the Summit supercomputer, with each node comprising of 2 IBM Power 9 CPUs (with 42 physical cores) and 6 NVIDIA Tesla V100 GPUs. We note that the number of nodes used to conduct the calculations is chosen such that the calculation is within the good parallel-scaling regime to obtain a representative measure of computational efficiency. In particular, for the larger systems considered in this work, the number of nodes are chosen such that the number of rows of $\mathbf{H}^{\mathrm{L}}$—the Kohn-Sham Hamiltonian matrix in the localized Tucker tensor basis—owned by each GPU is around 30,000, which maintains a good balance between memory limitation and parallel scaling efficiency.

(a) Al_{13} (b) Al_{147}

Figure 5.6: Schematics of the benchmark aluminum nano-particles.



(a) $Si_{10}H_{16}$ (b) $Si_{220}H_{144}$

Figure 5.7: Schematics of the benchmark silicon quantum dots.

5.4.1 Accuracy analysis of CPU and GPU implementation

In this section, we perform full ground-state calculations for the smaller benchmark systems of both aluminum nano-particles and silicon quantum dots to verify that the GPU implementation provides the same results as the CPU implementation. For the aluminum nano-particles, we choose Al_{13} and Al_{147}. The ball and stick model for the systems are provided in Fig. 5.6 (illustrated with VESTA [178]). For the silicon quantum dots, $Si_{10}H_{16}$ and $Si_{220}H_{144}$ are investigated. The ball and stick model for the silicon quantum dots are illustrated in Fig. 5.7. The converged results are tabulated in Table 5.1. We note that the results from the CPU-based and GPU-based calculations are identical up to the chemical accuracy of interest, with the differences at $\mathcal{O}(10^{-8})$ eV in the ground-state energy per atom. This small difference is possibly a result of the round-off error accumulations during the course of the ground-state calculation.

	Al_{13}	Al_{147}	$Si_{10}H_{16}$	$Si_{220}H_{144}$
CPU	-55.996571	-56.617932	-51.027192	-71.384192
GPU	-55.996571	-56.617932	-51.027192	-71.384192
|Error|	8.97×10^{-9}	4.35×10^{-8}	9.26×10^{-9}	1.21×10^{-8}

Table 5.1: Accuracy comparison of CPU and GPU implementation in ground state energy per atom (eV) for Al_{13}, Al_{147}, $Si_{10}H_{16}$, and $Si_{220}H_{144}$.

5.4.2 Performance analysis

In order to assess the computational efficiency derived from GPU acceleration, we compare the single SCF execution time in node-hours for the CPU-based implementation with that of the GPU acceleration. The computational times (in node-hours) for the various benchmark systems are provided in Table 5.2 and Table 5.3 for the aluminum nano-particles and silicon quantum dots, respectively. In particular, the breakdown of the single SCF computational time is provided for all the major steps

of Algorithm 4: (i) ChF: Chebyshev filtering step; (ii) Orth: Orthogonalization of the Chebyshev filtered vectors; (iii) Sub proj: Projection of the Kohn-Sham Hamiltonian matrix onto the Chebyshev filtered subspace; (iv) Others: all other costs in the SCF, including solution of the eigenvalue problem in the Chebyshev filtered subspace. The total computational cost for a single SCF iteration and the speedup from GPU acceleration is also provided.

5.4.2.1 Aluminum nano-particles

Table 5.2 shows the single SCF breakdown of the computational times of the CPU-based calculations and that of the GPU accelerated calculations for aluminum nano-particles of various sizes. We remark that the main focus of this work is to optimize the matrix-matrix multiplication kernel in the Chebyshev filtering step—the most expensive step in each SCF iteration—using GPU acceleration with the approach proposed in Sec. 5.3.2. Thus, the orthogonalization of the Chebyshev filtered vectors and other parts of in the calculations are essentially done on CPUs, which are identified using * in the table. For the smallest benchmark system considered (Al_{13}), we note that the calculation using GPU acceleration is slower than the CPU-based calculation, and the performance is comparable for Al_{147}. This is due to the overhead costs for transferring data between the host and the device memory that are competing in small system sizes with the arithmetic efficiency gained from GPU acceleration. However, for all the other systems, we obtain overall GPU acceleration in the Chebyshev filtering step and the subspace projection step—the two parts of the algorithm that are affected by the GPU acceleration of the matrix-matrix multiplication kernel. In particular, for the largest system size considered, Al_{6525} nano-particle, we obtain $\sim 7.8\times$ and $\sim 8.2\times$ computational efficiency in the Chebyshev filtering step and the subspace projection step, respectively, due to the GPU acceleration. This, in turn, provides a $\sim 3.1\times$ improvement in the computational efficiency for the SCF iteration

		ChF (node-hrs)	Orth* (node-hrs)	Sub proj (node-hrs)	Others* (node-hrs)	Time/SCF (node-hrs)	$\frac{\text{GPU}}{\text{CPU}}$
Al_{13}	CPU	1.87E-04	8.70E-05	3.57E-05	1.42E-04	4.52E-04	0.85
	GPU	2.34E-04	7.24E-05	3.12E-05	1.92E-04	5.30E-04	
Al_{147}	CPU	3.21E-02	2.14E-04	5.32E-03	1.12E-02	4.88E-02	1.39
	GPU	2.19E-02	2.02E-04	3.21E-03	9.80E-03	3.51E-02	
Al_{561}	CPU	0.331	0.008	0.047	0.156	0.542	1.75
	GPU	0.135	0.009	0.013	0.152	0.309	
Al_{2057}	CPU	3.724	0.228	0.482	1.327	5.761	2.97
	GPU	0.513	0.212	0.099	1.119	1.943	
Al_{6525}	CPU	30.119	6.192	3.422	4.132	43.865	3.12
	GPU	3.872	5.871	0.415	3.891	14.049	

Table 5.2: Breakdown of single-SCF computational times (in node-hours) for CPU-based and GPU-based calculations for the benchmark systems comprising of Al nano-particles. The columns marked with asterisk * are computed on the host (CPU) without GPU optimization.

step that is representative of the full ground-state calculation. We note that with possible GPU acceleration of the orthogonalization step, the potential improvement in the computational efficiency by using GPU acceleration can even be greater.

5.4.2.2 Silicon quantum dots

Table 5.3 shows the single SCF breakdown of computational times of CPU-based and GPU-accelerated calculations for various sizes of silicon quantum dots. Similar to the aluminum nano-particles, the benefits of GPU acceleration improve with system size. Notably, for the largest quantum dot system $Si_{6047}H_{1308}$ which contains 6355 atoms, the computational efficiency gain by using GPU acceleration is $\sim 7.2\times$ in Chebyshev filtering step and $\sim 6.8\times$ for the subspace projection step. The computational efficiency gain for the full SCF iteration is $\sim 3.4\times$.

		ChF (node-hrs)	Orth* (node-hrs)	Sub proj (node-hrs)	Others* (node-hrs)	Time/SCF (node-hrs)	$\frac{\text{GPU}}{\text{CPU}}$
$Si_{10}H_{16}$	CPU	1.37E-03	5.99E-04	1.83E-04	2.42E-03	4.57E-03	1.02
	GPU	1.52E-03	5.87E-04	1.79E-04	2.21E-03	4.50E-03	
$Si_{220}H_{144}$	CPU	6.12E-02	1.10E-03	6.99E-03	2.33E-02	9.26E-02	1.72
	GPU	2.79E-02	1.21E-03	4.62E-03	2.01E-02	5.38E-02	
$Si_{525}H_{276}$	CPU	0.515	0.018	0.067	0.214	0.814	1.9
	GPU	0.203	0.017	0.028	0.180	0.428	
$Si_{1214}H_{504}$	CPU	2.132	0.132	0.258	0.552	3.074	2.47
	GPU	0.611	0.127	0.087	0.422	1.247	
$Si_{6047}H_{1308}$	CPU	38.511	6.525	4.259	3.515	52.810	3.36
	GPU	5.385	6.473	0.629	3.223	15.710	

Table 5.3: Breakdown of single-SCF computational times (in node-hours) for CPU-based and GPU-based calculations for the benchmark systems comprising of silicon quantum dots. The columns marked with asterisk * are computed on the host (CPU) without GPU optimization.

5.5 Summary

We have presented the GPU accelerated algorithm used for accelerating the main compute intensive kernels of the calculation with GPU. In particular, the GPU accelerated tensor-structured algorithm for Kohn-Sham DFT calculation is based on using a systematically convergent localized basis that is generated from an additive separable approximation of the Kohn-Sham Hamiltonian [46]. The solution to the discrete Kohn-Sham problem is computed via Chebyshev filtering subspace iteration method [165]. The compute intensive kernels in this GPU accelerated code that involve matrix-matrix multiplication of a symmetric sparse matrix (Hamiltonian matrix) and a dense matrix (wavefunction matrix) have been GPU accelerated. The benchmark studies show a substantial improvement for the GPU-accelerated steps of the algorithm—$\sim 8\times$ for the largest system sizes—which improves the overall computational efficiency of the calculation. We note that, as shown in Chapter IV, the tensor-structured Kohn-Sham DFT algorithm can substantially outperform plane-wave implementations for large-scale systems [46], owing to the reduced-order scaling

with system size. The present GPU-based tensor-structured Kohn-Sham DFT code is a further step towards enabling systematically convergent and computationally efficient large-scale DFT calculations.

We note that upon GPU accelerating the compute intensive kernels, the main computational bottleneck in the tensor-structured algorithm for Kohn-Sham DFT calculation has now shifted to the step involving orthogonalization of the Chebyshev filtered vectors. Developing an efficient GPU implementation of the orthogonalization procedure and GPU porting other parts of the code can further improve the performance of the tensor-structured Kohn-Sham DFT code, and is a useful direction to pursue.

CHAPTER VI

Application of Tensor-Structured Techniques on Two-Electron Integrals

In the previous chapters, we have concluded our work on applying tensor-structured techniques on constructing the algorithm for generating an efficient basis for Kohn-Sham DFT calculation to achieve reduced-order scaling. In this chapter, we seek to further extend the use of tensor-structured techniques to other aspects of electronic structure calculation. We hence shift our focus to applying tensor-structured techniques to compute quantities appearing in wavefunction based methods. In particular, we set up a computation kernel for computing one-electron integrals and two-electron integrals appearing in a full configuration interaction (FCI) calculation using tensor-structured techniques. We note that the developed computation kernel could potentially enable us to develop a reduced-order basis for FCI computation using Kohn-Sham type orbitals in future work.

The remainder of this chapter is structured as follows. Section 6.1 gives an introduction to the basic nomenclature of the wavefunction based methods and previous attempts of using Kohn-Sham orbitals as an alternative molecular orbitals (MOs) to Hartree-Fock orbitals. Section 6.2 summarizes the derivation of the quantities appearing in the CI matrix. Section 6.3 and Section 6.4 discuss the implementation of the computation kernel for computing one-electron and two-electron quantities in a

CI matrix respectively. Section 6.5 and Section 6.6 present the computation results and summarize this work.

6.1 Introduction

As presented in the previous chapters, density functional theory provides an efficient way to compute the ground-state energy, hence other ground-state properties, by using electron density as the basic variable. However, density functional theory, to this date, lacks a systematically improvable functional for higher accuracy. On the other hand, wavefunction based methods use multi-determinant wavefunctions or perturbation theory to account for the electron correlation. The wavefunction methods generally can be categorized into three main types and their variances. The three types of electron correlation methods include configuration interaction methods (CI methods), many-body perturbation theory (MBPT, usually Møller-Plesset perturbation theory, or MP methods), and coupled-cluster methods (CC methods). We note that the details of these methods are out of the scope of this work and refer to Jensen [47] and Szabo & Ostlund [61] for a more complete introduction. Typically, in wavefunction based methods, the accuracy can be routinely improved at the expense of exponentially growing computational costs. To clarify, we summarize the putative scaling of different methods in Table 6.1. Therein, N denotes a system size measure, which is almost always proportional to the number of electrons in the system or the number of the basis functions used. The appended number in the column of MP methods specifies the order of correction used for MP methods. The S, D, T, Q letters in the columns of the CI Methods and the CC Methods indicate what excited Slater determinants are used in that method. The letter respectively stands for Singles (S-type, singly excited), Doubles (D-type, doubly excited), Triples (T-type, triply excited), and Quadruples (Q-type, quadruply excited). An example schematic of a system with three occupied orbitals and three virtual orbitals is shown in Fig. 6.1,

Scaling	CI Methods	MP Methods	CC Methods	Others
N^3				KS-DFT
N^4				HF†, DFT+Hybrid Functional
N^5	CIS	MP2	CC2	
N^6	CISD	MP3	CCSD	
N^7		MP4	CC3, CCSD(T)	
N^8	CISDT	MP5	CCSDT	
N^9	CISDT	MP6		
N^{10}	CISDTQ	MP7	CCSDTQ	
$N^{2m^{ex}+2}$	full CI			

Table 6.1: Putative scaling of different computational quantum chemistry methods. m^{ex} is the number of excited states used for the full CI calculation, † denotes that the practical scaling of Hartree-Fock method is N^2.

where the leftmost specifies the Hartree-Fock configuration that no electron is excited.

It is clear in Table 6.1 that as the level of accuracy increases, the computational complexity grows exponentially. Therefore, it is desirable to use an efficient basis set for the wavefunction based calculations to reduce the computational cost. To this end, we remark that, as all of the wavefunction based methods are based on Hartree-Fock approximation, it is intuitive and has been a common practice to construct the excited Slater determinants in the wavefunction based methods using molecular orbitals obtained from a Hartree-Fock calculation as shown in Fig. 6.1. On the other hand, recall that in Sec. 2.2 and Sec. 2.3.2, it has been shown that the Hartree-Fock method shares a similar formulation with the Kohn-Sham method. In both approaches, the original many-body Hamiltonian is re-formulated as a Hamiltonian representing a system with many non-interacting electrons moving in an effective mean-field generated from the contribution of all other electrons in the system. The only difference is being in that the Hartree-Fock method does not account for the correlation effect at all. This gives a hint that Kohn-Sham orbitals could carry more physical information and have better properties than Hartree-Fock orbitals. Some

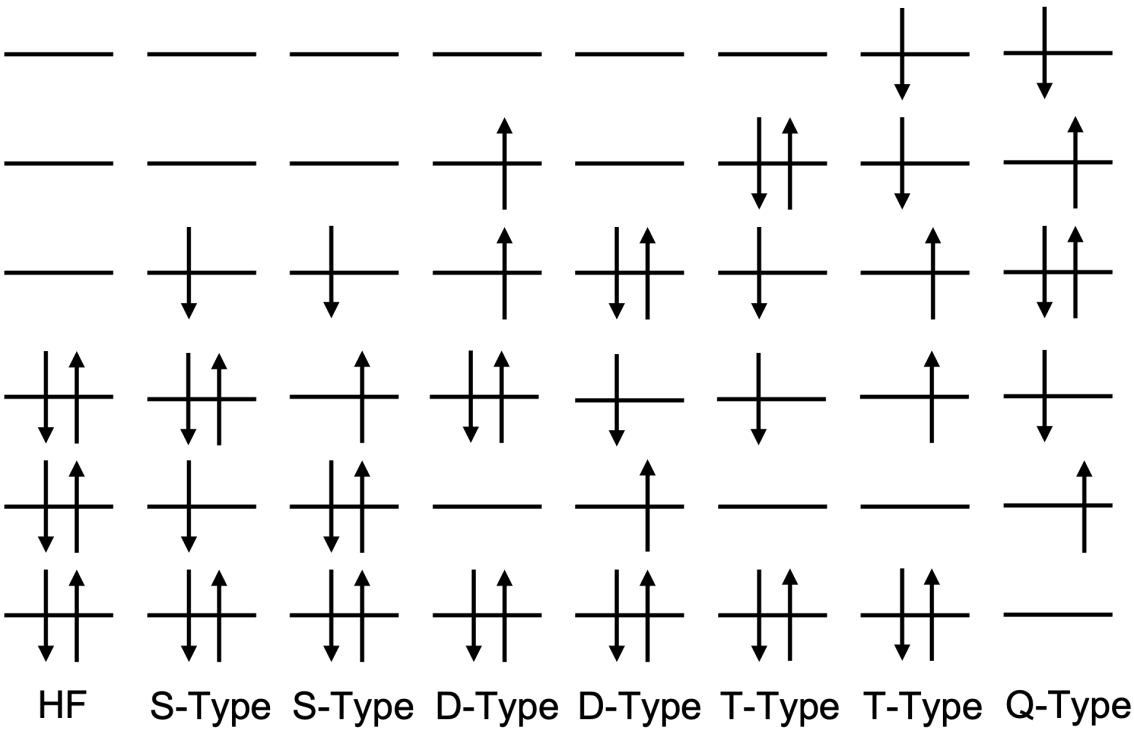


Figure 6.1: Different configurations of excited Slater determinant. Each arrow denotes an electron with spin and each bar denotes an orbital.

reports have even shown that the Kohn-Sham orbitals provide better shape for virtual orbitals and hence could be an ideal basis for molecular excitations [179, 180]. As a result, it is now reasonable to ask the question: *would Kohn-Sham orbitals be a better basis to account for electron correlation in a high-level wavefunction based method calculation than Hartree-Fock orbitals?*

This question is still an active area of research and, to some extent, has been partially answered. It has been shown that Kohn-Sham orbitals could be a better alternative to Hartree-Fock orbitals for some types of wavefunction based methods. For a CCSD(T) calculation, is has been shown that BLYP [82, 83, 84] Kohn-Sham orbitals can be a less expensive alternative to Hartree-Fock type orbitals [181, 48]. Some studies use exchange-only optimized effective potential [182, 183] under Krieger-Li-Iafrate approximation [184, 185] show that Kohn-Sham orbitals require smaller active space to achieve similar accuracy than Hartree-Fock orbitals in CIS and CISD

calculations [50, 186]. Even a recent study that doubts that Kohn-Sham orbitals would be more beneficial than Hartree-Fock orbitals in a CC type calculation in terms of energy convergence with respect to the full configuration interaction limit suggests that Kohn-Sham orbitals could still be used in the case that Hartree-Fock SCF calculations are hard to converge [49].

With all the evidence that Kohn-Sham orbitals could be a potential reduced-order alternative, or at least a useful alternative, to Hartree-Fock orbitals in many wavefunction based electron correlation methods, the applicability of the Kohn-Sham orbital to a full configuration interaction (FCI) calculation has not yet been discussed due to the high computational cost of FCI. It is thus desirable to set up a framework to investigate the applicability of Kohn-Sham orbitals on an FCI calculation. To this end, we propose to use DFT-FE [26, 3], a massively parallelized finite-element based DFT calculation, to obtain the Kohn-Sham orbitals. Subsequently, the Kohn-Sham orbitals computed on an adaptive finite element mesh should be interpolated to a tensor-structured mesh. On the tensor-structured mesh, we compute the one-electron and two-electron integrals, the central quantities for constructing the matrix element for an FCI calculation. The computationally expensive convolution integrals in the two-electron integral evaluation are computed using efficient tensor-structured techniques. Finally, the FCI calculation is done using the recently proposed incremental full configurational interaction (iFCI) method, which is shown to recover the correlation energy with relatively low cost by dividing the full configuration interaction into n-body interaction terms and approximately solves for each incremental energy [52, 51].

As a preliminary step, we seek to set up a workflow for the proposed framework. To this end, we need a universal computation kernel to generate the one-electron and two-electron integrals for CI matrix elements computation from any molecular orbitals projected on a tensor-structured finite-element mesh. In this work, we aim at setting

up this computation kernel for computing the one-electron and two-electron integrals from a mesh generated from DFT-FE and validating the consistency of the computed quantities with the FCI code. Therefore, an FCI computation is first performed for a hydrogen molecule using Hartree-Fock orbitals. We then construct a tensor-structured finite-element mesh from the DFT-FE code and interpolate the Hartree-Fock orbitals onto this mesh. Next, the one-electron and two-electron integrals are computed with these Hartree-Fock orbitals using the developed computation kernel. We then compare the residual norm of the two-electron integral tensor computed using the proposed computation kernel and the one generated intrinsically during the FCI calculation. The computed quantities are also passed to the FCI code to confirm the consistency of the computation kernel in terms of the CI energy with the normal FCI calculation.

6.2 CI matrix elements

Recall the electronic many-body Hamiltonian in Eq. 6.1 (the nuclei-nuclei repulsion is a constant under the Born-Oppenheimer approximation and omitted here for the sake of brevity)

$$\mathcal{H} = -\frac{1}{2}\sum_{i=1}^{N_e}\nabla_i^2 - \sum_{i=1}^{N_e}\sum_{I=1}^{N_a}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{i=1}^{N_e}\sum_{j<i}^{N_e}\frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|}\,. \tag{6.1}$$

The CI wavefunction constructed from the linear combination of excited Slater determinant has the form

$$\Psi^{\mathrm{CI}} = a_0\Psi_0^{\mathrm{S}} + \sum_i a_i\Psi_i^{\mathrm{S}}\,, \tag{6.2}$$

where Ψ_0^{S} is the 0-th Slater determinant of occupied orbitals, Ψ_i^{S} is a Slater determinant corresponding to different excitation, and a_i is the coefficient of that excited Slater determinant. Writing out the Lagrangian of the CI energy subject to the

normality constraint with the Lagrange multiplier ϵ as

$$\mathcal{L} = \left\langle \Psi^{\mathrm{CI}} \middle| \mathcal{H} \middle| \Psi^{\mathrm{CI}} \right\rangle - \epsilon \left(\left\langle \Psi^{\mathrm{CI}} \middle| \Psi^{\mathrm{CI}} \right\rangle - 1 \right) \tag{6.3}$$

and set the derivative with respect to the coefficient to zero, we arrive at the CI secular equation

$$\begin{aligned} &\frac{\partial \mathcal{L}}{\partial a_i} = 2 \sum_j a_j \left\langle \Psi_i^{\mathrm{S}} \middle| \mathcal{H} \middle| \Psi_j^{\mathrm{S}} \right\rangle - 2\epsilon a_i = 0 \\ &\mathbf{H}^{\mathrm{CI}} \mathbf{a} = \epsilon \mathbf{a} \,, \end{aligned} \tag{6.4}$$

where

$$\begin{aligned} &(\mathbf{H}^{\mathrm{CI}})_{ij} = H_{ij}^{\mathrm{CI}} = \left\langle \Psi_i^{\mathrm{S}} \middle| \mathcal{H} \middle| \Psi_j^{\mathrm{S}} \right\rangle \\ &(\mathbf{a})_i = a_i \end{aligned} \,.$$

Now, consider the molecular orbitals ψ^{MO} used to construct the Slater determinant are orthonormalized, it can be easily shown that only Slater determinants differing by less or equal to 2 molecular orbitals will give non-zero H_{ij}^{CI} entry. This result is called Slater-Condon rules [187, 188]. For example, using the shorthand notation introduced in Eq. 2.19, if we have two determinants differing by three MOs (can be Hartree-Fock orbitals or Kohn-Sham orbitals)

$$\begin{aligned} \left| \Psi_1^{\mathrm{CI}} \right\rangle &= \left| \psi_1^{\mathrm{MO}} \psi_2^{\mathrm{MO}} \psi_3^{\mathrm{MO}} \right\rangle \\ \left| \Psi_2^{\mathrm{CI}} \right\rangle &= \left| \psi_4^{\mathrm{MO}} \psi_5^{\mathrm{MO}} \psi_6^{\mathrm{MO}} \right\rangle \end{aligned} ,$$

the matrix element corresponding to these two wavefunctions is

$$\begin{aligned} \left\langle \Psi_1^{\mathrm{CI}} \middle| \mathcal{H} \middle| \Psi_2^{\mathrm{CI}} \right\rangle = \int\int\int \Bigg\{ & \psi_1^{\mathrm{MO}}(\mathbf{r}_1)^* \psi_2^{\mathrm{MO}}(\mathbf{r}_2)^* \psi_3^{\mathrm{MO}}(\mathbf{r}_3)^* \\ & \left(-\frac{1}{2} \sum_{i=1}^{3} \nabla_i^2 - \sum_{i=1}^{3} \sum_{I=1}^{N_a} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{i=1}^{3} \sum_{j<i}^{3} \frac{1}{\left|\mathbf{r}_i - \mathbf{r}_j\right|} \right) \cdot \\ & \psi_4^{\mathrm{MO}}(\mathbf{r}_1) \psi_5^{\mathrm{MO}}(\mathbf{r}_2) \psi_6^{\mathrm{MO}}(\mathbf{r}_3) \Bigg\} d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 + ... \end{aligned}$$

Therein, it is obvious that each term in this expression has an overlap integral of MOs from both Slater determinants. One of the intermediate term in the above derivation is

$$\int\int\Bigg\{\psi_1^{\mathrm{MO}}(\mathbf{r}_1)^*\psi_2^{\mathrm{MO}}(\mathbf{r}_2)^*\left(-\frac{1}{2}\nabla_1^2-\sum_{I=1}^{N_a}\frac{Z_I}{|\mathbf{r}_1-\mathbf{R}_I|}+\frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}\right)\psi_4^{\mathrm{MO}}(\mathbf{r}_1)\psi_5^{\mathrm{MO}}(\mathbf{r}_2)\Bigg\}d\mathbf{r}_1d\mathbf{r}_2\int\psi_3^{\mathrm{MO}}(\mathbf{r}_3)^*\psi_6^{\mathrm{MO}}(\mathbf{r}_3)d\mathbf{r}_3 .$$

Due to the orthogonality of MO orbitals, it is obvious that the overlapping term $\int\psi_3^{\mathrm{MO}}(\mathbf{r}_3)^*\psi_6^{\mathrm{MO}}(\mathbf{r}_3)d\mathbf{r}_3$ is zero, and subsequently, the whole term is zero. As a result, only Slater determinants differing by 0, 1, 2 MOs will contribute non-zero elements to the Hamiltonian matrix.

Finally, as shown in the above discussion, the non-zero terms in the CI Hamiltonian matrix involve only one-electron and two-electron integrals, where the one-electron integral is defined as

$$(i|h|j):=\int\psi_i^{\mathrm{MO}*}(\mathbf{r})\Bigg\{-\frac{1}{2}\nabla^2-\sum_I^{N_a}\frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}\Bigg\}\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r}\,, \tag{6.5}$$

and the two-electron integral is defined as

$$(ij|kl):=\int\int\psi_i^{\mathrm{MO}*}(\mathbf{r})\psi_j^{\mathrm{MO}*}(\mathbf{r})\frac{1}{|\mathbf{r}-\mathbf{r}'|}\psi_k^{\mathrm{MO}}(\mathbf{r}')\psi_l^{\mathrm{MO}}(\mathbf{r}')d\mathbf{r}d\mathbf{r}'\,. \tag{6.6}$$

We remark that there is 4-fold symmetry in Eq. 6.6, namely

$$(ij|kl)=(ji|lk)=(kl|ij)^*=(lk|ji)^*\,. \tag{6.7}$$

If the MOs are real-valued functions, which is the case in this work, the symmetry becomes 8-fold

$$\begin{aligned}(ij|kl)=(ij|lk)=(ji|lk)=(ji|kl)\\ =(kl|ij)=(lk|ij)=(lk|ji)=(kl|ji)\end{aligned}\,. \tag{6.8}$$

Hence, instead of computing all N^4 two-electron integrals, only $\sim \frac{N^4}{8}$ terms need to be computed. We also note that, as the orbitals with different spin will be integrated out due to orthogonality, the spin degrees of freedom are omitted here. For the next two sections, we will discuss the design of our computation kernel for computing the one-electron and two-electron integrals in this work. Besides, as a side note, the dimension of these two quantities is the same as the energy dimension, and the unit is Hartree if the atomic unit is used.

6.3 One-electron integrals computation

In this work, an MO orbital is computed on a tensor-structured finite-element mesh. On the mesh, the MO orbitals can be expressed as in the finite-element basis

$$\psi_i^{\mathrm{MO}}(\mathbf{r}) = \sum_j^M N^j(\mathbf{r}) \psi_i^{MO,j} \,, \tag{6.9}$$

where M is the number of finite-element basis used, $N^j(\mathbf{r})$ is the finite-element basis function, or called the shape function, and $\psi_i^{MO,j}$ denotes the coefficients of the finite-element basis, or the function value of the MO on the spatial nodes. We note that the mesh used in the current work is constructed to be tensor-structured. The three-dimensional shape function corresponding to the mesh hence has tensor structure and can be written as the tensor product of one-dimensional shape functions as

$$N^j(\mathbf{r}) = N_1^{j_1}(x_1) N_2^{j_2}(x_2) N_3^{j_3}(x_3) \,, \tag{6.10}$$

where $j = (j_1, j_2, j_3)$ is a composite index of three dimensions. In this work, the conventional Lagrange polynomial is used to construct the one-dimensional shape function as

$$N^i(x) = \prod_{j \neq i}^M \frac{x - x_j}{x_i - x_j} \,, \tag{6.11}$$

and its first order derivative is

$$\frac{dN^i(x)}{dx} := N^i(x)' = \sum_{k\neq i} \frac{1}{x_i - x_k} \prod_{j\neq i, j\neq k}^{M} \frac{x - x_j}{x_i - x_j}\,. \tag{6.12}$$

The MOs are then interpolated onto the Gauss-Lobatto-Legendre (GLL) quadrature points using the finite-element basis expansion in Eq. 6.9. The external potential $\sum_I^{N_a} \frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}$ is also computed analytically on the GLL quadrature points. Finally, a simple inner product of the three fields, $\psi_i^{\mathrm{MO}}(\mathbf{r})$, $\sum_I^{N_a} \frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}$, $\psi_j^{\mathrm{MO}}(\mathbf{r})$ on quadrature points gives the external part of the one-electron integral. We note that a full introduction to the finite-element methods and the Gauss-Lobatto-Legendre integration is beyond the scope of this discussion. We refer to [189] for a complete picture of finite-element methods.

Next, we turn our focus to the kinetic part of the one-electron integral. The expression of the kinetic part can be written as

$$(i|-\frac{1}{2}\nabla^2|j) = -\int \psi_i^{\mathrm{MO}*}(\mathbf{r})\frac{1}{2}\nabla^2\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r}\,. \tag{6.13}$$

To facilitate the evaluation of Eq. 6.13, using integration by part technique, the integral can be re-formulated as

$$\begin{aligned} -\frac{1}{2}\int \psi_i^{\mathrm{MO}*}(\mathbf{r})\nabla^2\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r} =& \frac{1}{2}\int \nabla\psi_i^{\mathrm{MO}*}(\mathbf{r})\cdot\nabla\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r} \\ & - \frac{1}{2}\int \psi_i^{\mathrm{MO}*}(\mathbf{r})\nabla\psi_j^{\mathrm{MO}}(\mathbf{r})\cdot\hat{\mathbf{n}}d\partial\mathbf{r} \end{aligned}\,, \tag{6.14}$$

where $\hat{\mathbf{n}}$ is the normal vector on the boundary $\partial\mathbf{r}$. We note that the second term on the right-hand side vanishes as MO vanishes on the boundary. Hence, we have

$$-\frac{1}{2}\int \psi_i^{\mathrm{MO}*}(\mathbf{r})\nabla^2\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r} = \frac{1}{2}\int \nabla\psi_i^{\mathrm{MO}*}(\mathbf{r})\cdot\nabla\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r}\,. \tag{6.15}$$

Now, substituting Eq. 6.9—Eq. 6.11 into Eq. 6.15, we arrive at the expression of the

kinetic part in finite-element basis

$$
\begin{aligned}
-\frac{1}{2}\int \psi_i^{\mathrm{MO}*}(\mathbf{r})\nabla^2\psi_j^{\mathrm{MO}}(\mathbf{r})d\mathbf{r} = \frac{1}{2}\sum_{a_1,a_2,a_3}\sum_{b_1,b_2,b_3}\Big\{\psi_i^{MO,a_1a_2a_3\,*}\psi_i^{MO,b_1b_2b_3} \\
\Big(\int N_1^{a_1}(x_1)'N_1^{b_1}(x_1)'dx_1\int N_2^{a_2}(x_2)N_2^{b_2}(x_2)dx_2\int N_3^{a_3}(x_3)N_3^{b_3}(x_3)dx_3 \\
+\int N_1^{a_1}(x_1)N_1^{b_1}(x_1)dx_1\int N_2^{a_2}(x_2)'N_2^{b_2}(x_2)'dx_2\int N_3^{a_3}(x_3)N_3^{b_3}(x_3)dx_3 \\
+\int N_1^{a_1}(x_1)N_1^{b_1}(x_1)dx_1\int N_2^{a_2}(x_2)N_2^{b_2}(x_2)dx_2\int N_3^{a_3}(x_3)'N_3^{b_3}(x_3)'dx_3\Big)\Big\}\,,
\end{aligned}
\tag{6.16}
$$

where $\psi_i^{MO,a_1a_2a_3}$ is the coefficient of the a-th three-dimensional finite-element basis, where $a=(a_1,a_2,a_3)$ is a composite index. Meanwhile, the $\psi_i^{MO,a_1a_2a_3}$ term can also be viewed as the MO function value at $\mathbf{r}=(x_1^{a_1},x_2^{a_2},x_3^{a_3})$. Furthermore, since the finite-element mesh employed here is tensor structured, the $\psi_i^{MO,a}$ is also a tensor indexed by (a_1,a_2,a_3). We also note that the one-dimensional integral terms

$$\int N_d^{a_d}(r_d)'N_d^{b_d}(r_d)'dr_d$$

and

$$\int N_d^{a_d}(r_d)N_d^{b_d}(r_d)dr_d\,,$$

where $d=1,2,3$ denotes the dimension, are precomputed. With the one-dimensional terms precomputed, the evaluation of Eq. 6.16 is simply a sparse matrix multiplication with a vector and a vector inner product. Hence, the computation can be very efficient using shell matrix approach or many highly optimized sparse numerical linear algebra computation kernels.

6.4 Two-electron integrals computation using tensor-structured techniques

To evaluate the two-electron integrals in Eq. 6.6, we utilize the approach employed in Sec. 4.2.1.1, where we used the tensor-structured approach to compute the convolution integrals for Hartree potential v_{H} evaluation. To elaborate a little bit more, consider the Gauss error function

$$\frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-t^2}dt = \lim_{z\to\infty}\mathrm{erf}(z) = \lim_{z\to\infty}\frac{2}{\sqrt{\pi}}\int_0^{z} e^{-t^2}dt = 1\,, \tag{6.17}$$

by change of variables $t = \alpha r$, we have

$$\frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-(\alpha r)^2}d(\alpha r) = \frac{2r}{\sqrt{\pi}}\int_0^{\infty} e^{-(\alpha r)^2}d\alpha = 1\,. \tag{6.18}$$

Simply by moving r from the middle expression to the rightmost one and apply the relation $r = \sqrt{x_1^2 + x_2^2 + x_3^3}$ to the r in the integrand, we have

$$\frac{1}{r} = \frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-(\alpha r)^2}d\alpha = \frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-\alpha^2(x_1^2+x_2^2+x_3^2)}d\alpha\,. \tag{6.19}$$

The integration can be written in the quadrature form as shown in Eq. 4.16

$$\frac{1}{r} \approx \sum_{k=1}^{K} w_k e^{-\alpha_k(x_1^2+x_2^2+x_3^2)}\,, \tag{6.20}$$

where α_k is the value of α^2 on the quadrature point k and w_k is the weight of the quadrature point k. While usual GLL quadrature rule can also be used to evaluate this integration, the infinite right integral boundary and varying r during convolution integrals evaluation make it hard to *a priori* determine the optimal quadrature points. To this end, it has been shown that $\forall r \in [a, b] \subseteq (0, \infty]$, $\exists$ unique best approximation

of the form Eq. 6.20 for $f(r) = \frac{1}{r}$ [190][1]. The coefficients w_k and α_k that give the best approximation can be obtained from the Remez algorithm [192]. In this work, we use the coefficients precomputed by Hackbusch that can be found on [162].

Now, we shift our focus back to the two-electron integrals. As the mesh on which the molecular orbitals are represented is tensor-structured, the molecular orbitals are also tensors. In the evaluation of the two-electron integral $(ij|kl)$, we first compute the Hadamard product (cf. Eq. 3.7) of the i, j pair and the k, l pair

$$\begin{aligned} \psi_{ij}{}^* &= \psi_i^{\mathrm{MO}*} * \psi_j^{\mathrm{MO}*} \\ \psi_{kl} &= \psi_k^{\mathrm{MO}} * \psi_j^{\mathrm{MO}} \end{aligned}, \tag{6.21}$$

and compute the Tucker tensor decomposition of ψ_{kl} as

$$\tilde{\psi}_{kl}(\mathbf{r}) = \sum_{r_1,r_2,r_3=1}^{R_\psi^{kl}} \sigma_{r_1r_2r_3}^{kl} \phi_{r_1}^{kl}(x_1)\phi_{r_2}^{kl}(x_2)\phi_{r_3}^{kl}(x_3)\,. \tag{6.22}$$

We note that Eq. 6.20 is a series of Gaussian functions, which is actually of the form of canonical representation (cf. Eq. 3.17). The $\frac{1}{r}$ kernel hence has the desirable separable structure for applying tensor-structured techniques to efficiently compute the convolution integrals. Now, combining Eq. 6.20 and Eq. 6.22, we then compute

$$\begin{aligned} \int \frac{\tilde{\psi}_{kl}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' = \sum_{k=1}^{K} w_k \sum_{r_1,r_2,r_3=1}^{R_\psi^{kl}} \Bigg(& \sigma_{r_1r_2r_3}^{kl} \int \phi_{r_1}^{kl}(x_1') e^{-\alpha_k(x_1-x_1')^2} dx_1' \\ & \int \phi_{r_2}^{kl}(x_2') e^{-\alpha_k(x_2-x_2')^2} dx_2' \int \phi_{r_3}^{kl}(x_3') e^{-\alpha_k(x_3-x_3')^2} dx_3' \Bigg) \end{aligned}, \tag{6.23}$$

on the quadrature points, namely, $\mathbf{r} = (x_1, x_1, x_3)$ are on the quadrature points.

Finally, we again interpolate $\psi_{ij}{}^*$ onto the quadrature points and compute the inner product of $\psi_{ij}{}^*$ (on quadrature points) and $\int \frac{\tilde{\psi}_{kl}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$, which is essentially a

[1]The cited chapter actually shows the proof for the more general γ-polynomial, where the kernel $e^{-\alpha x}$ used for approximating $\frac{1}{\sqrt{x}}$ is just a special case. Similar discussions specifically for $\frac{1}{x}$ and $\frac{1}{\sqrt{x}}$ can be found in earlier chapter in the same literature, [191, 161], or pp.273–274 in [89].

numerical integration using GLL quadratures, to obtain the two-electron integral $(ij|kl)$. As the MOs used in this work are real, we note again that, due to the symmetry shown in Eq. 6.8, only $\sim \frac{N^4}{8}$ terms need to be computed. Furthermore, we note that the convolution integrals are the most computationally expensive part for evaluating the two-electron integrals. While has not yet been implemented in the current scheme, we remark that this cost can be significantly reduced by carefully re-formulating the computation procedure so that the convolution term $\int \frac{\tilde{\psi}_{kl}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$ can be reused and only $\sim \frac{N^2}{2}$ convolution integrals need to be computed.

6.5 Results

Having the computation kernel set up, we now seek to validate the two-electron integrals computation. To this end, we use hydrogen molecule at its equilibrium bond length as the benchmark system. The coordinates of the hydrogen molecule is provided below in atomic unit

$$\begin{array}{ccc} \text{x} & \text{y} & \text{z} \\ (0.000000000000, & 0.000000000000, & 0.000000000000) \\ (0.000000000000, & 0.000000000000, & 1.398397231500) \end{array} .$$

To begin with, we present the two-electron integrals computed with Hartree-Fock molecular orbitals in a minimal Gaussian basis in Table 6.2 (unit: Hartree). The top part specifies the index of the computed two-electron integrals in each blank. The middle part is the values of the two-electron integral tensors T^{2e}_{RI} computed internally during the FCI calculation using the resolution of identity approximation (RI) [53, 54]. The bottom part is the values of two-electron integral tensor T^{2e}_{tensor} computed using the tensor-structured technique described in Sec. 6.4. It can readily be seen that all terms match to the third decimal place. To further compare the two calculations, we compute the norms of the two-electron integral tensor T^{2e}_{RI} and T^{2e}_{tensor} (cf. Sec. 3.1 for the definition of tensor norm). The norm is 1.3995426 for RI approximation and

1.3999828 for tensor techniques, respectively. Also, the relative residual norm defined by

$$\text{Tensor relative residual norm} := \frac{\|T^{2e}_{\text{RI}} - T^{2e}_{\text{tensor}}\|}{\|T^{2e}_{\text{RI}}\|} \tag{6.24}$$

is 3.1700×10^{-4}, which shows that the two-electron integral tensor computed using the two methods are satisfactorily close and our two-electron integrals computation kernel passed the sanity check. We note that the difference could be a combination of error from mesh discretization, $\frac{1}{r}$ kernel approximation, and the intrinsic error in the RI approximation itself.

Index							
(00|00)	(00|01)	(00|10)	(00|11)	(01|00)	(01|01)	(01|10)	(01|11)
(10|00)	(10|01)	(10|10)	(10|11)	(11|00)	(11|01)	(11|10)	(11|11)
RI							
0.67469887	0.00000000	0.00000000	0.66438368	0.00000000	0.18149731	0.18149731	0.00000000
0.00000000	0.18149731	0.18149731	0.00000000	0.66438368	0.00000000	0.00000000	0.69923143
Tensor techniques							
0.67494529	0.00000000	0.00000000	0.66459772	0.00000000	0.18153657	0.18153657	0.00000000
0.00000000	0.18153657	0.18153657	0.00000000	0.66459772	0.00000000	0.00000000	0.69942717

Table 6.2: The indices (top) and values for each two-electron integral tensor element computed using RI approximation (middle) and using tensor techniques (bottom).

Using the same hydrogen molecule system, we next examine the FCI computed energy using the internal RI approximation routine and the tensor techniques. The FCI energies are computed for the integrals using the first 5, 10, 15, and 20 Hartree-Fock orbitals spanned by a 92 Gaussian basis, respectively, from a Hartree-Fock calculation. As a sanity check, we again compute the tensor norms and the relative residual norms of the two-electron integral tensors using RI approximation and the proposed tensor techniques. The norms are summarized in Table 6.3. These integrals are provided as inputs to the FCI computation routine to obtain the energies tabulated in Table 6.4.

As shown in Table 6.3, the relative residual norms are at the order of 10^{-2}, which

	RI	**Tensor techniques**	**Residual**
5	1.13056	1.13361	1.69505E-02
10	2.05697	2.05897	1.61560E-02
15	3.32841	3.33877	1.53256E-02
20	4.69748	4.70746	1.59274E-02

Table 6.3: Tensor norms and residual norms of the two-electron integral tensors from RI approximation and tensor techniques using 5, 10, 15, 20 Hartree-Fock orbitals.

	$E_{\text{RI,FCI}}$ (Ha)	$E_{\text{tensor,FCI}}$ (Ha)	Error (mHa/atom)
5	-1.13460797	-1.13422915	0.189
10	-1.13602973	-1.13565113	0.189
15	-1.14093061	-1.14055086	0.190
20	-1.14378636	-1.14340782	0.189

Table 6.4: The FCI computed energy using 5, 10, 15, 20 Hartree-Fock orbitals. The CI matrix quantities are computed with RI approximation and tensor techniques respectively as denoted in the table.

is still quite close. The slightly larger relative residual norm, in comparison to the relative residual norm when using only two Hartree-Fock orbitals spanned by a smaller basis set presented in Table 6.2, could be because the higher states are more oscillatory and more diffused. As a result, the mesh might have to be more refined to achieve similar accuracy than the two Hartree-Fock orbitals case does. On the other hand, Table 6.4 shows that the energy per atom differences between FCI calculations using CI matrix components from RI approximation and tensor techniques are consistently below 0.2 mHa/atom for using 5, 10, 15, 20 Hartree-Fock orbitals calculations. This result suggests that the current mesh could be enough if the 0.2 mHa/atom accuracy in energy is satisfactory. Finally, we note that these investigations for the proposed computation kernel check that the proposed computation kernel in this work gives consistent CI matrix components. The proposed workflow should now be ready to be used to study the efficacy of Kohn-Sham orbitals for an FCI calculation.

6.6 Summary

In this chapter, we proposed a computation kernel based on tensor-structured techniques to generate the one-electron and two-electron integrals—central quantities for constructing the CI matrix—from any molecular orbitals projected on a tensor-structured finite-element mesh. We have shown that the two-electron integral tensors computed with the proposed computation kernel and the RI approximation used internally in the FCI code are consistent with the residual norm at the order of 10^{-2} for all 5–20 Hartree-Fock orbitals calculations. Furthermore, the difference in FCI computed energy using both methods for all 5–20 Hartree-Fock orbitals are within 0.2 mHa/atom.

The results confirm that the developed computation kernel correctly reproduces the one-electron and two-electron integrals computed internally inside of the FCI code. At the same time, the computation kernel can handle any real-valued molecular orbitals projected on the tensor-structured finite-element mesh. This provides more flexibility on the choice of molecular orbitals. Therefore, the Hartree-Fock orbitals used in this sanity check can be replaced by Kohn-Sham orbitals. The computation kernel is hence useful for studying the applicability of Kohn-Sham orbitals on an FCI type calculation and is currently being pursued.

CHAPTER VII

Conclusions

7.1 Summary

In this dissertation, our contribution is three-fold. The first part of this dissertation work developed a reduced-order tensor-structured algorithm for large-scale Kohn-Sham DFT calculations. The second part improved the computational efficiency of the tensor-structured algorithm for Kohn-Sham DFT using GPU acceleration. The third part of this dissertation work developed a computation kernel to generate one-electron and two-electron integrals with the flexibility to use any molecular orbitals projected on a tensor-structured finite-element mesh using tensor-structured techniques.

In the first part, we developed a tensor-structured algorithm for reduced-order large-scale Kohn-Sham DFT calculations. In the proposed algorithm, we first construct an additive separable approximation to the Kohn-Sham Hamiltonian by minimizing the energy functional of the Kohn-Sham equation with respect to the wavefunction in the rank-1 tensor space subjected to the orthonormality constraint. The minimizers of the energy functional ψ_1, ψ_2, ψ_3 are provided to the Kohn-Sham equation to obtain a set of one-dimensional Hamiltonians which are the additive separable approximation to the Kohn-Sham Hamiltonian, that is $\mathcal{H} \approx \mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$. Subsequently, the splitting orthogonal constraint algorithm is applied to the one-

dimensional Hamiltonians to obtain a set of one-dimensional localized functions whose span closely approximates the eigensubspace of the additive separable Hamiltonians. The outer product of the localized one-dimensional functions hence gives the desired localized Tucker tensor basis.

The convergence study for the fullerene and tris (bipyridine) ruthenium molecules, which have no tensor structure symmetry and serve as stringent benchmark systems, suggests that the proposed localized Tucker tensor basis converges exponentially with respect to the Tucker rank. The performance analysis for (1) aluminum nano-particles ranging from 13 to 6525 atoms and (2) silicon quantum dots ranging from 26 to 7715 atoms shows that the sub-quadratic scaling with respect to the system size is achieved by the localized Tucker tensor basis constructed from the proposed tensor-structured algorithm for both systems with and without a band gap. Furthermore, the efficiency comparison with Quantum Espresso, the state-of-the-art plane-wave based code, shows that the proposed localized Tucker tensor basis outperforms Quantum Espresso significantly for system-sizes beyond 5,000 electrons.

In the second part of this dissertation work, we improved the computational efficiency of the proposed Tensor-structured algorithm by GPU acceleration. During the course of solving the Kohn-Sham equation, the breakdown of computational times suggests that the Chebyshev filtering process, which involves heavy matrix-matrix multiplication between the sparse Hamiltonian and the dense wavefunction matrices, occupies the most computational time in the pure CPU code. This computationally expensive matrix-matrix multiplication is GPU accelerated using the CUDA library. Our performance analysis shows that the matrix-matrix multiplication is accelerated by $\sim$8x in comparison to the pure CPU code. We note that in combination with the previous results that the tensor-structured algorithm provides reduced-order scaling for large-scale Kohn-Sham DFT, this acceleration is a step further towards enabling systematically convergent and computationally efficient large-scale DFT calculations.

In the third part, we developed a computation kernel for one-electron and two-electron integrals using tensor-structured techniques. The computation kernel provides a flexible framework to compute the integrals for any molecular orbitals projected on a tensor-structured finite-element mesh. In this preliminary work, we used Hartree-Fock orbitals as molecular orbitals for a hydrogen molecule at its equilibrium bond length as a benchmark system to verify the computed two-electron integrals and the resulting full configuration interaction (FCI) energy of the proposed computation kernel. Our numerical results show that the residual norms of the two-electron integral tensors computed using the proposed computation kernel and the resolution of identity (RI) approximation, which is the internal subroutine for computing the two-electron integrals in FCI code, respectively are at the order of 10^{-2}. Furthermore, the difference in the resulting FCI energy is within 0.2 mHa/atom for various numbers of molecular orbitals.

7.2 Future directions

In this dissertation work, we have developed a tensor-structured algorithm for reduced-order scaling large-scale Kohn-Sham DFT calculations and accelerated the algorithm with GPU. Meanwhile, we developed a flexible computation kernel using tensor-structured techniques to provide the one-electron and two-electron integrals for FCI calculations. Based on these results, we summarize some useful directions being pursued or to pursue in the future as follows.

Further GPU acceleration for tensor-structured algorithm for Kohn-Sham DFT

In the present GPU acceleration for the tensor-structured algorithm for Kohn-Sham DFT, we focused on accelerating the previously computationally expensive spare-dense matrix-matrix multiplication kernel for the sparse Hamiltonian and the

dense wavefunction matrices during the Chebyshev filtering process. As shown in Table 5.2 and Table 5.3, the bottleneck has now shifted to the orthogonalization of the filtered Chebyshev vectors. It will hence be useful to develop an efficient GPU implementation for the orthogonalization procedure and GPU port other parts of the code, which will be a step forward to enabling DFT calculations for currently computationally prohibitive systems.

Computation of one-electron and two-electron integrals using Kohn-Sham orbitals

In Chapter VI, a computation kernel with the flexibility to generate one-electron and two-electron integrals for any molecular orbitals projected on a tensor-structured finite-element mesh is proposed. This computation kernel can be used to provide the one-electron and two-electron integrals using Kohn-Sham orbitals obtained from DFT-FE, a massively parallelized finite-element based DFT code. These integrals can subsequently be used to compute FCI energy and investigate the feasibility of Kohn-Sham orbitals as a reduced-order basis for FCI calculations. This framework can potentially reduce the computational cost of FCI calculation and serves as the key step towards enabling calculations for larger systems with an FCI level of accuracy.

Larger-scale FCI calculations for more accurate exchange-correlation functional

With a reduced-orbital basis for FCI calculations, it is possible to perform the ground-state calculation with FCI level of accuracy for previously computationally inaccessible systems. This ground state can be used to compute the electron density and serves as a training dataset for machine learning to generate a more accurate exchange-correlational functional, e.g. using the scheme proposed in [74]. This framework could be a step forward to a more affordable DFT calculation with a higher level

of accuracy.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Hai Xiao, Jamil Tahir-Kheli, and William A. Goddard. Accurate Band Gaps for Semiconductors from Density Functional Theory. *The Journal of Physical Chemistry Letters*, 2(3):212–217, feb 2011.

[2] Roman Zhuravel, Haichao Huang, Georgia Polycarpou, Savvas Polydorides, Phani Motamarri, Liat Katrivas, Dvir Rotem, Joseph Sperling, Linda A. Zotti, Alexander B. Kotlyar, Juan Carlos Cuevas, Vikram Gavini, Spiros S. Skourtis, and Danny Porath. Backbone charge transport in double-stranded DNA. *Nature Nanotechnology*, 15(10):836–840, oct 2020.

[3] Sambit Das, Phani Motamarri, Vikram Gavini, Bruno Turcksin, Ying Wai Li, and Brent Leback. Fast, scalable and accurate finite-element based ab initio calculations using mixed precision computing: 46 pflops simulation of a metallic dislocation system. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Jianwei Sun. Application of wavefunction methods to metals. *Nature Computational Science*, 1(12):780–781, dec 2021.

[5] Peng Xu, Emilie B. Guidez, Colleen Bertoni, and Mark S. Gordon. Perspective: Ab initio force field methods derived from quantum mechanics. *Journal of Chemical Physics*, 148(9), 2018.

[6] Alexander Urban, Dong Hwa Seo, and Gerbrand Ceder. Computational understanding of Li-ion batteries. *npj Computational Materials*, 2(January), 2016.

[7] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), 2013.

[8] Paul C. Jennings, Steen Lysgaard, Jens Strabo Hummelshøj, Tejs Vegge, and Thomas Bligaard. Genetic algorithms for computational materials discovery accelerated by machine learning. *npj Computational Materials*, 5(1), 2019.

[9] Nicola Marzari, Andrea Ferretti, and Chris Wolverton. Electronic-structure methods for materials design. *Nature Materials*, 20(6):736–749, 2021.

[10] Lidunka Vočadlo, Dario Alfè, M. J. Gillan, I. G. Wood, J. P. Brodholt, and G. David Price. Possible thermal and chemical stabilization of body-centered-cubic iron in the Earth's core. *Nature*, 424(6948):536–539, 2003.

[11] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[12] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[13] Aurora Pribram-Jones, David A. Gross, and Kieron Burke. DFT: A Theory Full of Holes? *Annual Review of Physical Chemistry*, 66(1):283–304, apr 2015.

[14] Axel D. Becke. Perspective: Fifty years of density-functional theory in chemical physics. *The Journal of Chemical Physics*, 140(18):18A301, May 2014.

[15] R. O. Jones. Density functional theory: Its origins, rise to prominence, and future. *Reviews of Modern Physics*, 87(3):897–923, August 2015.

[16] Narbe Mardirossian and Martin Head-Gordon. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Molecular Physics*, 115(19):2315–2372, June 2017.

[17] John P. Perdew, Adrienn Ruzsinszky, Jianmin Tao, Viktor N. Staroverov, Gustavo E. Scuseria, and Gábor I. Csonka. Prescription for the design and selection of density functional approximations: More constraint satisfaction with fewer fits. *The Journal of Chemical Physics*, 123(6):062201, August 2005.

[18] Michael G. Medvedev, Ivan S. Bushmarinov, Jianwei Sun, John P. Perdew, and Konstantin A. Lyssenko. Density functional theory is straying from the path toward the exact functional. *Science*, 355(6320):49–52, January 2017.

[19] Kasper P. Kepp. Comment on "density functional theory is straying from the path toward the exact functional". *Science*, 356(6337):496.2–496, May 2017.

[20] G. Kresse and J. Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct 1996.

[21] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, Ph. Ghosez, J.-Y. Raty, and D.C. Allan. First-principles computation of material properties: the abinit software project. *Computational Materials Science*, 25(3):478 – 492, 2002.

[22] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502 (19pp), 2009.

[23] P Giannozzi, O Andreussi, T Brumme, O Bunau, M Buongiorno Nardelli, M Calandra, R Car, C Cavazzoni, D Ceresoli, M Cococcioni, N Colonna, I Carnimeo, A Dal Corso, S de Gironcoli, P Delugas, R A DiStasio Jr, A Ferretti, A Floris, G Fratesi, G Fugallo, R Gebauer, U Gerstmann, F Giustino, T Gorni, J Jia, M Kawamura, H-Y Ko, A Kokalj, E Küçükbenli, M Lazzeri, M Marsili, N Marzari, F Mauri, N L Nguyen, H-V Nguyen, A Otero de-la Roza, L Paulatto, S Poncé, D Rocca, R Sabatini, B Santra, M Schlipf, A P Seitsonen, A Smogunov, I Timrov, T Thonhauser, P Umari, N Vast, X Wu, and S Baroni. Advanced capabilities for materials modelling with QUANTUM ESPRESSO. *Journal of Physics: Condensed Matter*, 29(46):465901, 2017.

[24] Stewart J. Clark, Matthew D. Segall, Chris J. Pickard, Phil J. Hasnip, Matt I. J. Probert, Keith Refson, and Mike C. Payne. First principles methods using CASTEP. *Zeitschrift für Kristallographie - Crystalline Materials*, 220(5-6):567–570, May 2005.

[25] P. Motamarri, M.R. Nowak, K. Leiter, J. Knap, and V. Gavini. Higher-order adaptive finite-element methods for Kohn–Sham density functional theory. *Journal of Computational Physics*, 253:308–343, 2013.

[26] Phani Motamarri, Sambit Das, Shiva Rudraraju, Krishnendu Ghosh, Denis Davydov, and Vikram Gavini. DFT-FE – A massively parallel adaptive finite-element code for large-scale density functional theory calculations. *Computer Physics Communications*, 246:106853, 2020.

[27] Evgeny Epifanovsky, Andrew T. B. Gilbert, Xintian Feng, Joonho Lee, Yuezhi Mao, Narbe Mardirossian, Pavel Pokhilko, Alec F. White, Marc P. Coons, Adrian L. Dempwolff, Zhengting Gan, Diptarka Hait, Paul R. Horn, Leif D. Jacobson, Ilya Kaliman, Jörg Kussmann, Adrian W. Lange, Ka Un Lao, Daniel S. Levine, Jie Liu, Simon C. McKenzie, Adrian F. Morrison, Kaushik D. Nanda, Felix Plasser, Dirk R. Rehn, Marta L. Vidal, Zhi-Qiang You, Ying Zhu, Bushra Alam, Benjamin J. Albrecht, Abdulrahman Aldossary, Ethan Alguire, Josefine H. Andersen, Vishikh Athavale, Dennis Barton, Khadiza Begam, Andrew Behn, Nicole Bellonzi, Yves A. Bernard, Eric J. Berquist, Hugh G. A. Burton, Abel Carreras, Kevin Carter-Fenk, Romit Chakraborty, Alan D. Chien,

Kristina D. Closser, Vale Cofer-Shabica, Saswata Dasgupta, Marc de Wergifosse, Jia Deng, Michael Diedenhofen, Hainam Do, Sebastian Ehlert, Po-Tung Fang, Shervin Fatehi, Qingguo Feng, Triet Friedhoff, James Gayvert, Qinghui Ge, Gergely Gidofalvi, Matthew Goldey, Joe Gomes, Cristina E. González-Espinoza, Sahil Gulania, Anastasia O. Gunina, Magnus W. D. Hanson-Heine, Phillip H. P. Harbach, Andreas Hauser, Michael F. Herbst, Mario Hernández Vera, Manuel Hodecker, Zachary C. Holden, Shannon Houck, Xunkun Huang, Kerwin Hui, Bang C. Huynh, Maxim Ivanov, Ádám Jász, Hyunjun Ji, Hanjie Jiang, Benjamin Kaduk, Sven Kähler, Kirill Khistyaev, Jaehoon Kim, Gergely Kis, Phil Klunzinger, Zsuzsanna Koczor-Benda, Joong Hoon Koh, Dimitri Kosenkov, Laura Koulias, Tim Kowalczyk, Caroline M. Krauter, Karl Kue, Alexander Kunitsa, Thomas Kus, István Ladjánszki, Arie Landau, Keith V. Lawler, Daniel Lefrancois, Susi Lehtola, Run R. Li, Yi-Pei Li, Jiashu Liang, Marcus Liebenthal, Hung-Hsuan Lin, You-Sheng Lin, Fenglai Liu, Kuan-Yu Liu, Matthias Loipersberger, Arne Luenser, Aaditya Manjanath, Prashant Manohar, Erum Mansoor, Sam F. Manzer, Shan-Ping Mao, Aleksandr V. Marenich, Thomas Markovich, Stephen Mason, Simon A. Maurer, Peter F. McLaughlin, Maximilian F. S. J. Menger, Jan-Michael Mewes, Stefanie A. Mewes, Pierpaolo Morgante, J. Wayne Mullinax, Katherine J. Oosterbaan, Garrette Paran, Alexander C. Paul, Suranjan K. Paul, Fabijan Pavošević, Zheng Pei, Stefan Prager, Emil I. Proynov, Ádám Rák, Eloy Ramos-Cordoba, Bhaskar Rana, Alan E. Rask, Adam Rettig, Ryan M. Richard, Fazle Rob, Elliot Rossomme, Tarek Scheele, Maximilian Scheurer, Matthias Schneider, Nickolai Sergueev, Shaama M. Sharada, Wojciech Skomorowski, David W. Small, Christopher J. Stein, Yu-Chuan Su, Eric J. Sundstrom, Zhen Tao, Jonathan Thirman, Gábor J. Tornai, Takashi Tsuchimochi, Norm M. Tubman, Srimukh Prasad Veccham, Oleg Vydrov, Jan Wenzel, Jon Witte, Atsushi Yamada, Kun Yao, Sina Yeganeh, Shane R. Yost, Alexander Zech, Igor Ying Zhang, Xing Zhang, Yu Zhang, Dmitry Zuev, Alán Aspuru-Guzik, Alexis T. Bell, Nicholas A. Besley, Ksenia B. Bravaya, Bernard R. Brooks, David Casanova, Jeng-Da Chai, Sonia Coriani, Christopher J. Cramer, György Cserey, A. Eugene DePrince, Robert A. DiStasio, Andreas Dreuw, Barry D. Dunietz, Thomas R. Furlani, William A. Goddard, Sharon Hammes-Schiffer, Teresa Head-Gordon, Warren J. Hehre, Chao-Ping Hsu, Thomas-C. Jagau, Yousung Jung, Andreas Klamt, Jing Kong, Daniel S. Lambrecht, WanZhen Liang, Nicholas J. Mayhall, C. William McCurdy, Jeffrey B. Neaton, Christian Ochsenfeld, John A. Parkhill, Roberto Peverati, Vitaly A. Rassolov, Yihan Shao, Lyudmila V. Slipchenko, Tim Stauch, Ryan P. Steele, Joseph E. Subotnik, Alex J. W. Thom, Alexandre Tkatchenko, Donald G. Truhlar, Troy Van Voorhis, Tomasz A. Wesolowski, K. Birgitta Whaley, H. Lee Woodcock, Paul M. Zimmerman, Shirin Faraji, Peter M. W. Gill, Martin Head-Gordon, John M. Herbert, and Anna I. Krylov. Software for the frontiers of quantum chemistry: An overview of developments in the q-chem 5 package. *The Journal of Chemical Physics*, 155(8):084801, 2021.

[28] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb,

J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian~16 Revision C.01, 2016. Gaussian Inc. Wallingford CT.

[29] E. Aprà, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. W. Aquino, R. Atta-Fynn, J. Autschbach, N. P. Bauman, J. C. Becca, D. E. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Boschen, J. Brabec, A. Bruner, E. Cauët, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. J. O. Deegan, T. H. Dunning, M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Früchtl, L. Gagliardi, J. Garza, N. Gawande, S. Ghosh, K. Glaesemann, A. W. Götz, J. Hammond, V. Helms, E. D. Hermes, K. Hirao, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jónsson, R. A. Kendall, M. Klemm, R. Kobayashi, V. Konkov, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, R. J. Littlefield, A. J. Logsdail, K. Lopata, W. Ma, A. V. Marenich, J. Martin del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, T. Nakajima, D. R. Nascimento, J. A. Nichols, P. J. Nichols, J. Nieplocha, A. Otero-de-la Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, R. M. Richard, P. Sadayappan, G. C. Schatz, W. A. Shelton, D. W. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. S. Thomas, V. Tipparaju, D. G. Truhlar, K. Tsemekhman, T. Van Voorhis, Á. Vázquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Woliński, A. T. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison. Nwchem: Past, present, and future. *The Journal of Chemical Physics*, 152(18):184102, 2020.

[30] Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Computer Physics Communications*, 180(11):2175–2196, 2009.

[31] Stefan Goedecker. Linear scaling electronic structure methods. *Rev. Mod. Phys.*, 71:1085–1123, Jul 1999.

[32] D R Bowler and T Miyazaki. $\mathcal{O}$(N) methods in electronic structure calculations. *Reports on Progress in Physics*, 75(3):036503, Feb 2012.

[33] X.-P. Li, R. W. Nunes, and David Vanderbilt. Density-matrix electronic-structure method with linear system-size scaling. *Phys. Rev. B*, 47:10891–10894, Apr 1993.

[34] Francesco Mauri, Giulia Galli, and Roberto Car. Orbital formulation for electronic-structure calculations with linear system-size scaling. *Phys. Rev. B*, 47:9973–9976, Apr 1993.

[35] S. Goedecker and L. Colombo. Efficient linear scaling algorithm for tight-binding molecular dynamics. *Phys. Rev. Lett.*, 73:122–125, Jul 1994.

[36] Jeongnim Kim, Francesco Mauri, and Giulia Galli. Total-energy global optimizations using nonorthogonal localized orbitals. *Phys. Rev. B*, 52:1640–1648, Jul 1995.

[37] Uwe Stephan and David A. Drabold. Order-N projection method for first-principles computations of electronic quantities and wannier functions. *Phys. Rev. B*, 57:6391–6407, Mar 1998.

[38] P. D. Haynes, C. K. Skylaris, A. A. Mostofi, and M. C. Payne. ONETEP: linear-scaling density-functional theory with local orbitals and plane waves. *Phys. Stat. Sol.(b)*, 243:2489–2499, 2006.

[39] C. J. García-Cervera, Jianfeng Lu, Yulin Xuan, and Weinan E. Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for kohn-sham density functional theory. *Phys. Rev. B*, 79:115110, Mar 2009.

[40] Lin Lin, Mohan Chen, Chao Yang, and Lixin He. Accelerating atomic orbital-based electronic structure calculation via pole expansion and selected inversion. *Journal of Physics: Condensed Matter*, 25(29):295501, Jun 2013.

[41] Phani Motamarri and Vikram Gavini. Subquadratic-scaling subspace projection method for large-scale Kohn-Sham density functional theory calculations using spectral finite-element discretization. *Physical Review B*, 90(11):115127, 2014.

[42] Wolfgang Hackbusch and Boris N. Khoromskij. Tensor-product approximation to operators and functions in high dimensions. *Journal of Complexity*, 23(4):697 – 714, 2007. Festschrift for the 60th Birthday of Henryk Woźniakowski.

[43] B. N. Khoromskij and V. Khoromskaia. Multigrid accelerated tensor approximation of function related multidimensional arrays. *SIAM Journal on Scientific Computing*, 31(4):3002–3026, 2009.

[44] T. Blesgen, V. Gavini, and V. Khoromskaia. Approximation of the electron density of aluminium clusters in tensor-product format. *Journal of Computational Physics*, 231(6):2551 – 2564, 2012.

[45] Phani Motamarri, Vikram Gavini, and Thomas Blesgen. Tucker-tensor algorithm for large-scale Kohn-Sham density functional theory calculations. *Physical Review B*, 93(12):125104, Mar 2016.

[46] Chih-Chuen Lin, Phani Motamarri, and Vikram Gavini. Tensor-structured algorithm for reduced-order scaling large-scale Kohn–Sham density functional theory calculations. *npj Computational Materials*, 7(1):50, Dec 2021.

[47] Frank Jensen. *Introduction to Computational Chemistry*. John Wiley & Sons, Ltd, Hoboken, NJ, USA, 3rd edition, 2016.

[48] Luke W. Bertels, Joonho Lee, and Martin Head-Gordon. Polishing the Gold Standard: The Role of Orbital Choice in CCSD(T) Vibrational Frequency Prediction. *Journal of Chemical Theory and Computation*, 17(2):742–755, 2021.

[49] Zsolt Benedek, Paula Tímár, Tibor Szilvási, and Gergely Barcza. Sensitivity of Coupled Cluster Electronic Properties on the Reference Determinant: Can Kohn-Sham Orbitals Be More Beneficial than Hartree-Fock Orbitals? sep 2021.

[50] Jaechang Lim, Sunghwan Choi, Jaewook Kim, and Woo Youn Kim. Outstanding performance of configuration interaction singles and doubles using exact exchange Kohn-Sham orbitals in real-space numerical grid method. *Journal of Chemical Physics*, 145(22), 2016.

[51] Paul M. Zimmerman. Strong correlation in incremental full configuration interaction. *The Journal of Chemical Physics*, 146(22):224104, jun 2017.

[52] Paul M. Zimmerman. Incremental full configuration interaction. *The Journal of Chemical Physics*, 146(10):104102, mar 2017.

[53] Martin Feyereisen, George Fitzgerald, and Andrew Komornicki. Use of approximate integrals in ab initio theory. An application in MP2 energy calculations. *Chemical Physics Letters*, 208(5-6):359–363, jun 1993.

[54] Florian Weigend, Marco Häser, Holger Patzelt, and Reinhart Ahlrichs. RI-MP2: optimized auxiliary basis sets and demonstration of efficiency. *Chemical Physics Letters*, 294(1-3):143–152, sep 1998.

[55] Paul Adrien Maurice Dirac. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society A*, 123(792):714–733, apr 1929.

[56] E. Schrödinger. An Undulatory Theory of the Mechanics of Atoms and Molecules. *Physical Review*, 28(6):1049–1070, dec 1926.

[57] Wikipedia. Hartree atomic units — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Hartree\%20atomic\%20units&oldid=1042443500, 2021. [Online; accessed 05-December-2021].

[58] David J. Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2nd edition, aug 2004.

[59] P A M Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(3):416–418, jul 1939.

[60] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. Oxford University Press, New York, NY, USA, 4th edition, 1958.

[61] Neil S. Ostlund Attila Szabo. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. McGraw-Hill, New York, NY, USA, 1st edition, 1989.

[62] Max C. Holthausen Wolfram Koch. *A Chemist's Guide to Density Functional Theory*. WILEY-VCH, 2nd edition, 2001.

[63] Dieter Cremer. Density functional theory: Coverage of dynamic and non-dynamic electron correlation effects. *Molecular Physics*, 99(23):1899–1940, 2001.

[64] Robert G. Parr and Weitao Yang. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, New York, NY, USA, 1989.

[65] T. L. Gilbert. Hohenberg-Kohn theorem for nonlocal external potentials. *Physical Review B*, 12(6):2111–2120, sep 1975.

[66] Mel Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem. *Proceedings of the National Academy of Sciences*, 76(12):6062 LP – 6065, dec 1979.

[67] Mel Levy. Electron densities in search of Hamiltonians. *Physical Review A*, 26(3):1200–1208, sep 1982.

[68] L. H. Thomas. The calculation of atomic fields. *Mathematical Proceedings of the Cambridge Philosophical Society*, 23(5):542–548, jan 1927.

[69] Enrico Fermi. Un metodo statistico per la determinazione di alcune priorieta dell'atome. *Rend. Accad. Naz. Lincei*, 6(602-607):32, 1927.

[70] F. Sarcinella, E. Fabiano, L. A. Constantin, and F. Della Sala. Nonlocal kinetic energy functionals in real space using a Yukawa-potential kernel: Properties, linear response, and model functionals. *Physical Review B*, 103(15):155127, apr 2021.

[71] Yan Alexander Wang and Emily A Carter. *Orbital-Free Kinetic-Energy Density Functional Theory*, pages 117–184. Springer Netherlands, Dordrecht, 2002.

[72] Vikram Gavini, Jaroslaw Knap, Kaushik Bhattacharya, and Michael Ortiz. Non-periodic finite-element formulation of orbital-free density functional theory. *Journal of the Mechanics and Physics of Solids*, 55(4):669–696, apr 2007.

[73] Phani Motamarri, Mrinal Iyer, Jaroslaw Knap, and Vikram Gavini. Higher-order adaptive finite-element methods for orbital-free density functional theory. *Journal of Computational Physics*, 231(20), 2012.

[74] Bikash Kanungo, Paul M. Zimmerman, and Vikram Gavini. Exact exchange-correlation potentials from ground-state electron densities. *Nature Communications*, 10(1), 2019.

[75] Johannes Gedeon, Jonathan Schmidt, Matthew J P Hodgson, Jack Wetherell, Carlos L. Benavides-Riveros, and Miguel A. L. Marques. Machine learning the derivative discontinuity of density-functional theory. *Machine Learning: Science and Technology*, 3(1):015011, mar 2022.

[76] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566–569, Aug 1980.

[77] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of Physics*, 58(8):1200–1211, aug 1980.

[78] J. P. Perdew and Alex Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, May 1981.

[79] John P. Perdew and Yue Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B*, 45:13244–13249, Jun 1992.

[80] John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized Gradient Approximation Made Simple. *Physical Review Letters*, 77(18):3865–3868, oct 1996.

[81] Yan Zhao and Donald G. Truhlar. A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *The Journal of Chemical Physics*, 125(19):194101, nov 2006.

[82] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Physical Review A*, 38(6):3098–3100, sep 1988.

[83] Chengteh Lee, Weitao Yang, and Robert G. Parr. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Physical Review B*, 37(2):785–789, jan 1988.

[84] Axel D. Becke. Density-functional thermochemistry. III. The role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, apr 1993.

[85] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *The Journal of Physical Chemistry*, 98(45):11623–11627, nov 1994.

[86] John P. Perdew. Jacob's ladder of density functional approximations for the exchange-correlation energy. In *AIP Conference Proceedings*, volume 577, pages 1–20. AIP, 2001.

[87] Richard M. Martin. *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press, 2004.

[88] Mike Finnis. *Interatomic Forces in Condensed Matter*. Oxford University Press, New York, NY, USA, 2010.

[89] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*, volume 42 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[90] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, Aug 2009.

[91] So Hirata. Tensor contraction engine: Abstraction and automated parallel implementation of configuration-interaction, coupled-cluster, and many-body perturbation theories. *Journal of Physical Chemistry A*, 107(46):9887–9897, 2003.

[92] Paul Springer and Paolo Bientinesi. Design of a High-Performance GEMM-like Tensor–Tensor Multiplication. *ACM Transactions on Mathematical Software*, 44(3):1–29, jan 2018.

[93] Rafael Ballester-Ripoll. tntorch - Tensor Network Learning with PyTorch.

[94] Paul Springer, Tong Su, and Paolo Bientinesi. HPTT: A High-Performance Tensor Transposition C++ Library. In *Proceedings of the 4th ACM SIGPLAN International Workshop on Libraries, Languages, and Compilers for Array Programming*, ARRAY 2017, pages 56–62, New York, NY, USA, 2017. ACM.

[95] NVIDIA. cutensor, v1.4.0, 2021.

[96] Frank L. Hitchcock. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics*, 6(1-4):164–189, apr 1927.

[97] Frank L Hitchcock. Multiple Invariants and Generalized Rank of a P-Way Matrix or Tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79, apr 1928.

[98] L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. In C. W. Harris, editor, *Problems in measuring change.*, pages 122–137. University of Wisconsin Press, Madison WI, 1963.

[99] Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[100] L. R. Tucker. The extension of factor analysis to three-dimensional matrices. In H. Gulliksen and N. Frederiksen, editors, *Contributions to mathematical psychology.*, pages 110–127. Holt, Rinehart and Winston, New York, 1964.

[101] C. J. Appellof and E. R. Davidson. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. *Analytical Chemistry*, 53(13):2053–2056, nov 1981.

[102] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, jul 2017.

[103] Eleftherios Kofidis and Phillip A. Regalia. Tensor approximation and signal processing applications. pages 103–133. 2001.

[104] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear Analysis of Image Ensembles: TensorFaces. pages 447–460. 2002.

[105] Hongyang Chen, Sergiy A. Vorobyov, Hing Cheung So, Fauzia Ahmad, and Fatih Porikli. Introduction to the Special Issue on Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):433–437, apr 2021.

[106] Hemanth Kolla, Konduri Aditya, and Jacqueline H. Chen. Higher Order Tensors for DNS Data Analysis and Compression. In *Data Analysis for Direct Numerical Simulations of Turbulent Combustion*, pages 109–134. Springer International Publishing, Cham, 2020.

[107] Thomas Schultz and H.-P. Seidel. Estimating Crossing Fibers: A Tensor Decomposition Approach. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1635–1642, nov 2008.

[108] Shahrooz Faghih Roohi, Dornoosh Zonoobi, Ashraf A Kassim, and Jacob L Jaremko. Dynamic MRI reconstruction using low rank plus sparse tensor decomposition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1769–1773. IEEE, sep 2016.

[109] Sibasankar Padhy, Griet Goovaerts, Martijn Boussé, Lieven De Lathauwer, and Sabine Van Huffel. The Power of Tensor-Based Approaches in Cardiac Applications. pages 291–323. 2020.

[110] C.F. Beckmann and S.M. Smith. Tensorial extensions of independent component analysis for multisubject FMRI analysis. *NeuroImage*, 25(1):294–311, mar 2005.

[111] Chenyu Zhang, Rungang Han, Anru R. Zhang, and Paul.M. Voyles. Denoising atomic resolution 4D scanning transmission electron microscopy data with tensor singular value decomposition. *Ultramicroscopy*, 219:113123, Dec 2020.

[112] F.I. Allen, M. Watanabe, Z. Lee, N.P. Balsara, and A.M. Minor. Chemical mapping of a block copolymer electrolyte by low-loss EFTEM spectrum-imaging and principal component analysis. *Ultramicroscopy*, 111(3):239–244, Feb 2011.

[113] Velimir V. Vesselinov, Boian S. Alexandrov, and Daniel O'Malley. Nonnegative tensor factorization for contaminant source identification. *Journal of Contaminant Hydrology*, 220:66–97, Jan 2019.

[114] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, Jan 2009.

[115] M.V. Rakhuba and I.V. Oseledets. Grid-based electronic structure calculations: The tensor decomposition approach. *Journal of Computational Physics*, 312:19–30, may 2016.

[116] Venera Khoromskaia and Boris N. Khoromskij. *Tensor Numerical Methods in Quantum Chemistry*. De Gruyter, Berlin, Boston, jun 2018.

[117] B. N. Khoromskij and V. Khoromskaia. Low rank tucker-type tensor approximation to classical potentials. *Central European Journal of Mathematics*, 5(3):523–550, Sep 2007.

[118] V. Khoromskaia, B. N. Khoromskij, and R. Schneider. Tensor-Structured Factorized Calculation of Two-Electron Integrals in a General Basis. *SIAM Journal on Scientific Computing*, 35(2):A987–A1010, Jan 2013.

[119] V. Khoromskaia. Computation of the Hartree-Fock Exchange by the Tensor-Structured Methods. *Computational Methods in Applied Mathematics*, 10(2):204–218, 2010.

[120] Mario Motta, James Shee, Shiwei Zhang, and Garnet Kin Lic Chan. Efficient Ab Initio Auxiliary-Field Quantum Monte Carlo Calculations in Gaussian Bases via Low-Rank Tensor Decomposition. *Journal of Chemical Theory and Computation*, 15(6):3510–3521, 2019.

[121] A. Tichai, R. Schutski, G. E. Scuseria, and T. Duguet. Tensor-decomposition techniques for ab initio nuclear structure calculations: From chiral nuclear potentials to ground-state energies. *Physical Review C*, 99(3):1–15, 2019.

[122] V. Khoromskaia and B. N. Khoromskij. Prospects of Tensor-Based Numerical Modeling of the Collective Electrostatics in Many-Particle Systems. *Computational Mathematics and Mathematical Physics*, 61(5):864–886, 2021.

[123] Bo Peng and Karol Kowalski. Highly Efficient and Scalable Compound Decomposition of Two-Electron Integral Tensor and Its Application in Coupled Cluster Calculations. *Journal of Chemical Theory and Computation*, 13(9):4179–4192, 2017.

[124] Karl Pierce, Varun Rishi, and Edward F. Valeev. Robust Approximation of Tensor Networks: Application to Grid-Free Tensor Factorization of the Coulomb Interaction. *Journal of Chemical Theory and Computation*, 17(4):2217–2230, 2021.

[125] I. V. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, jan 2011.

[126] Age Smilde, Rasmus Bro, and Paul Geladi. *Multi-Way Analysis with Applications in the Chemical Sciences*. John Wiley & Sons, Ltd, Chichester, UK, aug 2004.

[127] Boris N. Khoromskij. *Tensor Numerical Methods in Scientific Computing*. De Gruyter, Berlin, Boston, jun 2018.

[128] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, (16):1–84, 1970.

[129] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, sep 1970.

[130] Henk A. L. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122, may 2000.

[131] Mikael Sørensen, Lieven De Lathauwer, Pierre Comon, Sylvie Icart, and Luc Deneire. Canonical Polyadic Decomposition with a Columnwise Orthonormal Factor Matrix. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1190–1213, jan 2012.

[132] Johan Håstad. Tensor rank is NP-complete. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 372 LNCS, pages 451–460. 1989.

[133] Tamara G. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2002.

[134] Joseph Levin. Three-mode factor analysis. *Psychological Bulletin*, 64(6):442–452, 1965.

[135] Pieter M. Kroonenberg and Jan de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.

[136] Arie Kapteyn, Heinz Neudecker, and Tom Wansbeek. An approach to n-mode components analysis. *Psychometrika*, 51(2):269–275, jun 1986.

[137] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[138] J. B. Kruskal. Rank, decomposition, and uniqueness for 3-way and N-way arrays. pages 7–18. North-Holland, Amsterdam, 1989.

[139] Woody Austin, Grey Ballard, and Tamara G. Kolda. Parallel tensor compression for large-scale scientific data. In *IPDPS'16: Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium*, pages 912–922, May 2016.

[140] Grey Ballard, Alicia Klinvex, and Tamara G. Kolda. TuckerMPI: A parallel C++/MPI software package for large-scale data compression via the tucker tensor decomposition. *ACM Trans. Math. Softw.*, 46(2):13:1–13:31, 2020.

[141] Claus A. Andersson and Rasmus Bro. Improving the speed of multi-way algorithms: Part I. Tucker3. *Chemometrics and Intelligent Laboratory Systems*, 42(1-2):93–103, 1998.

[142] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the Best Rank-1 and Rank-(R 1 , R 2 ,. . ., R N) Approximation of Higher-Order Tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, jan 2000.

[143] Lars Eldén and Berkant Savas. A Newton–Grassmann Method for Computing the Best Multilinear Rank-(r_1, r_2, r_3) Approximation of a Tensor. *SIAM Journal on Matrix Analysis and Applications*, 31(2):248–271, jan 2009.

[144] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, August 2013.

[145] J. Douglas Carroll, Sandra Pruzansky, and Joseph B. Kruskal. Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45(1):3–24, mar 1980.

[146] David Hong, Tamara G. Kolda, and Jed A. Duersch. Generalized Canonical Polyadic Tensor Decomposition. *SIAM Review*, 62(1):133–163, jan 2020.

[147] Evrim Acar and B. Yener. Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1):6–20, jan 2009.

[148] R. A. Harshman. PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22:30–47, 1972.

[149] R. A. Harshman. Models for analysis of asymmetrical relationships among N objects or stimuli. In *First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology*, Hamilton, Ontario, 1978. McMaster University.

[150] Rasmus Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149–171, oct 1997.

[151] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Communications in Mathematical Physics*, 144(3):443–490, mar 1992.

[152] Stellan Östlund and Stefan Rommer. Thermodynamic Limit of Density Matrix Renormalization. *Physical Review Letters*, 75(19):3537–3540, nov 1995.

[153] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2):022320, aug 2006.

[154] Lars Grasedyck. Hierarchical Singular Value Decomposition of Tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, jan 2010.

[155] W. Hackbusch and S. Kühn. A New Scheme for the Tensor Representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, oct 2009.

[156] Wolfgang Hackbusch and Reinhold Schneider. Tensor spaces and hierarchical tensor representations. In *Extraction of Quantifiable Information from Complex Systems*, pages 237–261. Springer International Publishing, 2014.

[157] N. Troullier and José Luís Martins. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B*, 43:1993–2006, Jan 1991.

[158] Leonard Kleinman and D. M. Bylander. Efficacious form for model pseudopotentials. *Phys. Rev. Lett.*, 48:1425–1428, May 1982.

[159] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[160] B.N. Khoromskij, V. Khoromskaia, S.R. Chinnamsetty, and H.-J. Flad. Tensor decomposition in electronic structure calculations on 3d cartesian grids. *Journal of Computational Physics*, 228(16):5749 – 5762, 2009.

[161] Dietrich Braess and Wolfgang Hackbusch. On the efficient computation of high-dimensional integrals and the approximation by exponential sums. In *Multiscale, Nonlinear and Adaptive Approximation*, pages 39–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[162] Wolfgang Hackbusch. Approximation of 1/sqrt(x) by exponential sums, 2008.

[163] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.

[164] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration. *Phys. Rev. E*, 74:066704, Dec 2006.

[165] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Self-consistent-field calculations using chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219(1):172–184, November 2006.

[166] Alex Breuer and Xin-Cindy Wang. More robust Chebyshev filtering for SCF iteration, with applications in real-space DFT. *Journal of Computational Physics*, 374:27–46, dec 2018.

[167] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics, 2011.

[168] Yunkai Zhou, James R. Chelikowsky, and Yousef Saad. Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation. *Journal of Computational Physics*, 274:770–782, oct 2014.

[169] NVIDIA. cusparse, release: Aug 2, 2021, 2021.

[170] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G.E. Fagg, E. Gabriel, and J.J. Dongarra. Performance analysis of mpi collective operations. In *19th IEEE International Parallel and Distributed Processing Symposium*, pages 8 pp.–, 2005.

[171] William Gropp, Ewing Lusk, and Rajeev Thakur. *Using MPI-2*. 2019.

[172] NVIDIA. cublas, release: Nov 28, 2019, 2019.

[173] NVIDIA. Cuda toolkit, release: Nov 23, 2021, 2021.

[174] NVIDIA. Cuda hpc sdk, release: Dec 9, 2021, 2021.

[175] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[176] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dmitry Karpeyev, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.14, Argonne National Laboratory, 2020.

[177] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dmitry Karpeyev, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page, 2019.

[178] Koichi Momma and Fujio Izumi. VESTA 3for three-dimensional visualization of crystal, volumetric and morphology data. *Journal of Applied Crystallography*, 44(6):1272–1276, October 2011.

[179] E. J. Baerends, O. V. Gritsenko, and R. Van Meer. The Kohn-Sham gap, the fundamental gap and the optical gap: The physical meaning of occupied and virtual Kohn-Sham orbital energies. *Physical Chemistry Chemical Physics*, 15(39):16408–16425, 2013.

[180] R. van Meer, O. V. Gritsenko, and E. J. Baerends. Physical Meaning of Virtual Kohn–Sham Orbitals and Orbital Energies: An Ideal Basis for the Description of Molecular Excitations. *Journal of Chemical Theory and Computation*, 10(10):4432–4441, oct 2014.

[181] Gregory J.O. Beran, Steven R. Gwaltney, and Martin Head-Gordon. Approaching closed-shell accuracy for radicals using coupled cluster theory with perturbative triple substitutions. *Physical Chemistry Chemical Physics*, 5(12):2488–2493, 2003.

[182] James D. Talman and William F. Shadwick. Optimized effective atomic central potential. *Physical Review A*, 14(1):36–40, jul 1976.

[183] R. T. Sharp and G. K. Horton. A Variational Approach to the Unipotential Many-Electron Problem. *Physical Review*, 90(2):317–317, apr 1953.

[184] J. B. Krieger, Yan Li, and G. J. Iafrate. Exact relations in the optimized effective potential method employing an arbitrary Exc[{ψiσ}]. *Physics Letters A*, 148(8-9):470–474, 1990.

[185] J. B. Krieger, Yan Li, and G. J. Iafrate. Construction and application of an accurate local spin-polarized Kohn-Sham potential with integer discontinuity: Exchange-only theory. *Physical Review A*, 45(1):101–126, 1992.

[186] Jaewook Kim, Kwangwoo Hong, Sunghwan Choi, Sang Yeon Hwang, and Woo Youn Kim. Configuration interaction singles based on the real-space numerical grid method: Kohn-Sham versus Hartree-Fock orbitals. *Physical Chemistry Chemical Physics*, 17(47):31434–31443, 2015.

[187] J. C. Slater. The Theory of Complex Spectra. *Physical Review*, 34(10):1293–1322, nov 1929.

[188] E. U. Condon. The Theory of Complex Spectra. *Physical Review*, 36(7):1121–1133, oct 1930.

[189] Thomas J. R. Hughes. *Finite Element Method: Linear Static And Dynamic Finite Element Analysis*. Dover Publications, Inc, Mineola, New York, 1st edition, 2000.

[190] Dietrich Braess. *Chebyshev Approximation by γ-Polynomials*, pages 181–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

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

[192] Eugene Remes. Sur un procédé convergent d'approximations successives pour déterminer les polynômes d'approximation. *CR Acad. Sci. Paris*, 198:2063–2065, 1934.