Leveraging symmetries and low-rank structure of matrices and tensors in high-dimensional quantum chemistry problems

Siwar Badreddine

Sorbonne université

Inria

Symétries et structures de rang faible des matrices et tenseurs pour des problèmes en chimie quantique.

Par: Siwar Badreddine

Thèse de doctorat en mathématiques appliquées
dirigée par Laura Grigori

École doctorale Sciences Mathématiques de Paris Centre
Laboratoire Jacques-Louis Lions

Composition du jury

Bart Vandereycken	University of Geneva	Rapporteur
Edmond Chow	Georgia Institute of Technology	Rapporteur
Laura Grigori	EPFL Lausanne	Directrice de thèse
Alex Townsend	Cornell University	Examinateur
Anthony Nouy	Ecole Centrale Nantes	Examinateur
Eric Cancès	Ecole des Ponts ParisTech	Examinateur
Yvon Maday	University of Sorbonne	Examinateur

Sorbonne université
Inria

Leveraging symmetries and low-rank structure of matrices and tensors in high-dimensional quantum chemistry problems

By: Siwar Badreddine

Ph.D thesis in applied mathematics
directed by Laura Grigori

Doctoral School Sciences Mathématiques de Paris Centre
Laboratoire Jacques-Louis Lions

Committee members

Bart Vandereycken	University of Geneva	Reviewer
Edmond Chow	Georgia Institute of Technology	Reviewer
Laura Grigori	EPFL Lausanne	Ph.D Advisor
Alex Townsend	Cornell University	Examiner
Anthony Nouy	Ecole Centrale Nantes	Examiner
Eric Cancès	Ecole des Ponts ParisTech	Examiner
Yvon Maday	University of Sorbonne	Examiner

Acknowledgements

The most gratifying aspect of conducting this research work lies in the moments of understanding something. While the journey is undoubtedly challenging and, at times, frustrating combined with existential despair (at some point), the experience becomes exceptionally motivating and rewarding when this is achieved. This pursuit of understanding has been the essence and ultimate satisfaction of my thesis.

Nevertheless, this dissertation would not be possible without the efforts of several individuals, either in terms of scientific input or moral support. It is imperative for me to acknowledge their roles. So, dear reader, I must forewarn you that this section might extend a bit longer than expected, and for that, I extend my apologies in advance. Please bear with me until the end.

I am immensely grateful to my supervisor, Laura Grigori, for her trust and guidance throughout this journey. She has provided me with numerous opportunities to showcase my work, attend conferences abroad, and engage with some of the brightest minds in this field, who have significantly enriched my understanding. I also want to express my appreciation for her patience and support. During moments of panic or setbacks or when things appeared to be not working out, her encouragement has been a constant source of motivation. Additionally, I wish to thank Edmond Chow and Bart Vandereycken for kindly agreeing to review my dissertation.

Over the last years, I have had the privilege of collaborating closely with Igor Chollet, Mi-Song Dupuy, and Eric Cancès, whose profound influence has greatly contributed to my development as a young scientist. Our discussions and collaborative efforts were instrumental in shaping the trajectory of my PhD thesis. I express my heartfelt gratitude to Igor Chollet for his indispensable assistance and the countless enlightening discussions we shared. His generosity in sharing his expertise in FMM and particularly in using his library *defmm*, has been invaluable to me. Additionally, I am grateful for Mi-Song Dupuy's patience and kindness in answering my questions, no matter how seemingly trivial, which I truly admire. Our discussions on board have significantly enhanced my understanding of complex topics. Finally, I extend my appreciation to Eric Cancès for his guidance and pedagogical explanations on intricate subjects. I have learned a lot from you.

Special thanks to Igor Chollet, Mi-Song Dupuy, and Muhammad Hassan for helping me proof-reading sections of this thesis and providing invaluable feedback. Furthermore, I am thankful for the insightful discussions I had with Julien Toulouse and Emmanuel Giner during my first year as a PhD student, who provided essential support in quantum chemistry fundamentals. Additionally, their generosity in sharing expertise through their quantum package proved invaluable.

Over the past four years, I have had the pleasure of encountering remarkable individuals in various settings. I extend my heartfelt gratitude to those who warmly welcomed me to the Alpines team at Inria Suraj, Van Than, and Matthias. Thank you for making a delightful work environment and for helping me out in every way and supporting me, especially during these final months. To my longest academic brother and conference travel partner, Edouard Timsit, your humor made this journey more enjoyable and I am going to miss your jokes by the way. To those who joined our circle later on Daniel Torres, Jean-Guillaume de Damas, Emile Parolin, Mathieu Verite, Yanfei Xiang, Nicola Galante, and Yuejia Zhang thank you for all the good moments that we had. I am also grateful to all the researchers and staff at Inria, particularly Sever Hirstoaga, Frédéric Nataf, Frédéric Hecht, Laurence Bourcier, and Christine Tanguy. I am immensely grateful to other members of the Inria teams. Foremost, I extend my appreciation to Chourouk, Juliette, and Haibo for their profound kindness throughout these years. They have consistently been available for discussions and guidance, particularly in navigating French

bureaucracy. Lastly, I am thankful to my dear friends whom I met during the Cemracs summer school, Roberta, Giulia, and Augustine. Our time together was filled with laughter and memorable experiences, making it truly enjoyable. I would also like to extend my heartfelt gratitude to the wonderful friends I made at the Roscoff meetings, Mohamed, Diata, Olivier, and my seafood partner, Ionna. Special thanks are extended to the PIs of the EMC2 project Eric Cancès, Laura Grigori, Yvon Maday, and Jean-Philip Piquemal, for their instrumental role in organizing theses invaluable Roscoff meetings.

On a personal note, I am deeply grateful to all my friends who have been an important source of support throughout the years. Whether during moments of joy or difficult times, they have stood by me, offering their understanding and companionship. Their presence has never failed to bring a smile to my face. My thoughts go to Lynda, Yosra, Manel, Ines, Salma, Oumaima, Maha, Mouna, Maryem, Mahdi, Niez, Youssef, Khaled, Iheb, Hachem, Oussema. I am deeply thankful to my parents, Mounir and Hayet, who have put in me a love for science. I thank them for never ceasing to support me, even in my lowest moments. From the very beginning, they encouraged me to strive for excellence and supported my decisions to pursue this path. A special dedication to my father, who taught me the very basics of mathematics. This work is dedicated to you. I must express my heartfelt gratitude to Syrine, my older sister, who has been by my side through thick and thin. My warmest thoughts extend to her husband, Nedhir, and my adorable niece, Aicha, who truly is the cherry on top of the cake. I am deeply grateful to all my family members who supported me throughout my academic journey. Their encouragement, understanding, and belief in me have been a constant source of motivation. Their willingness to offer advice, and provide a sense of stability during challenging times have been invaluable. Special thanks to my aunt Monia and my uncle Moez. Lastly, thank you my dear fiance, Mohamed, who has been by my side over the years, witnessing both the joys and challenges I have faced. He always found a way to bring a smile to my face and cheer me up, even in the toughest of times. You followed me in this adventure and I am forever grateful for you. Thank you to the reader who has made it this far and still has more to read. I hope you enjoy this dissertation.

Abstract

This thesis presents novel numerical algorithms and conducts a comprehensive study of some existing numerical methods to address high-dimensional challenges arising from the electronic Schrödinger equation in quantum chemistry. Focusing on two specific problems, our approach involves the identification and exploitation of symmetries and low-rank structures within matrices and tensors, aiming to mitigate the curse of dimensionality.

The first problem considered in this thesis is the efficient numerical evaluation of the long-range component of the range-separated Coulomb potential and the long-range two-electron integrals 4th-order tensor which occurs in many quantum chemistry methods. We present two novel approximation methods. This is achieved by relying on tensorized Chebyshev interpolation, Gaussian quadrature rules combined with low-rank approximations as well as Fast Multipole Methods (FMM). This work offers a detailed explanation of these introduced approaches and algorithms, accompanied by a thorough comparison between the newly proposed methods.

The second problem of interest is the exploitation of symmetries and low-rank structures to derive efficient tensor train representations of operators involved in the Density Matrix Renormalization Group (DMRG) algorithm. This algorithm, referred to as the Quantum Chemical DMRG (QC-DMRG) when applied in the field of quantum chemistry, is an accurate iterative optimization method employed to numerically solve the time-independent Schrödinger equation. The aim of this work is to understand and interpret the results obtained from the physics and chemistry communities and seek to offer novel theoretical insights that, to the best of our knowledge, have not received significant attention before. We conduct a comprehensive study and provide demonstrations, when necessary, to explore the existence of a particular block-sparse tensor train representation of the Hamiltonian operator and its associated eigenfunction. This is achieved while maintaining physical conservation laws, manifested as group symmetries in tensors, such as the conservation of the particle number.

The third part of this work is dedicated to the realization of a proof-of-concept Quantum Chemical DMRG (QC-DMRG) Julia library, designed for the quantum chemical Hamiltonian operator model. We exploit here the block-sparse tensor train representation of both the operator and the eigenfunction. With these structures, our goal is to speed-up the most time consuming steps in QC-DMRG, including tensor contractions, matrix-vector operations, and matrix compression through truncated Singular Value Decompositions (SVD). Furthermore, we provide empirical results from various molecular simulations, while comparing the performance of our library with the state-of-the-art ITensors library where we show that we attain a similar performance.

Keywords—Numerical linear algebra, Multilinear algebra, High-dimensional problems, Quantum chemistry, Schrödinger equation, Two-electron integrals, Symmetries, low-rank approximation, SVD, Tensor train representations, Hamiltonian operator, QC-DMRG.

Contents

Résumé en français

La représentation de l'information repose souvent sur des matrices et des tenseurs. Ces outils mathématiques ont trouvé leur utilité dans divers domaines, allant des mathématiques pures aux disciplines appliquées telles que la physique, la chimie et l'économie. Ils ont été utiles pour résoudre des défis du monde réel depuis des décennies. Cependant, le développement technologique récent et rapide apporte de nouveaux défis mathématiques et informatiques, ouvrant la voie à de nouvelles orientations de recherche. Pour relever ces nouveaux défis, il est nécessaire de continuer à progresser dans le domaine de l'algèbre linéaire et multilinéaire.

L'un des défis majeurs d'aujourd'hui concerne le traitement d'ensembles de données volumineux résultant de divers problèmes de la vie réelle, tels que les simulations numériques de modèles physiques pour simuler tout système composé de composants physiques réels. On peut rencontrer cela dans de nombreux domaines, notamment l'astrophysique, la biologie, la climatologie, la chimie quantique, etc. Ce problème est couramment connu sous le nom de *curse of dimensionality*, un terme inventé par Bellman dans le contexte de la théorie de l'approximation [8]. Ce terme fait référence à la nécessité d'utiliser un nombre de degrés de liberté en croissance exponentielle à mesure que la dimension du problème augmente. Par conséquent, il existe deux principales voies : l'une consiste en le développement continu d'outils matériels puissants et coûteux pour résoudre ces problèmes de haute dimension, et l'autre consiste en le développement d'algorithmes innovants qui rendent la haute dimensionnalité gérable. Pour le deuxième point, on peut étudier et exploiter les structures sous-jacentes des matrices ou des tenseurs impliqués dans la représentation de ces problèmes.

Notamment, une technique d'algèbre linéaire établie pour traiter cette question implique la méthode d'approximation de rang faible, obtenue par la décomposition en valeurs singulières (SVD) pour les matrices, par exemple. La SVD a notamment trouvé une large utilité dans de nombreuses disciplines, dont la science des données, l'astronomie, la chimie quantique, le traitement des signaux et la science du climat. Comme les dimensions continuent de s'étendre, il devient de plus en plus pertinent d'explorer d'autres formats de représentation des données, tels que les tenseurs.

Il est à noter que les tenseurs sont connus et utilisés depuis des décennies, sous différentes notations et concepts au sein de différentes communautés, notamment la physique, la chimie et les mathématiques. Le terme *tenseurs* a initialement émergé dans le contexte de la mécanique, introduit par Hamilton en 1854 [41]. Par la suite, entre 1880 et 1916, les tenseurs ont été utilisés comme généralisation des scalaires, des vecteurs et des matrices dans les domaines de la géométrie différentielle et de la physique, notamment dans la théorie de la relativité générale d'Einstein [97]. Entre 1927 et 2011, le concept de décomposition tensorielle est apparu, tel qu'introduit par Hitchcock en 1927 [46], permettant la représentation de tenseurs de haute dimension sous la forme d'une série de composants de dimension inférieure. En conséquence, entre 1960 et 1970, de nombreux algorithmes de factorisation ont été proposés, tels que la décomposition canonique (CP) et la décomposition de Tucker [68]. En outre, dans les années 2010 et 2011, Grasedyck a introduit la décomposition hiérarchique de Tucker, et Osledets a décrit le format du train de tenseurs (TT) [92], qui correspond au concept *Matrix Product States* (MPS) introduit précédemment dans la communauté physique dès 1989. Ce concept de réseau de tenseurs est ensuite devenu intimement lié au *Density Matrix Renormalization Group* (DMRG), une technique d'optimisation populaire introduite par White en 1992 pour approximer l'état fondamental, c'est-à-dire l'état d'énergie la plus basse, des systèmes quantiques fortement corrélés.

Dans ce travail, notre objectif est de développer de nouveaux algorithmes numériques,

d'étudier des méthodes numériques existantes pour traiter les défis de grande dimension en chimie quantique, et de fournir une analyse numérique approfondie des méthodes proposées. La résolution des équations mécaniques quantiques à l'aide de méthodes de discrétisation de base peut rapidement conduire à des problèmes de grande dimension et demandant beaucoup de calculs. Simultanément, les solutions doivent maintenir un haut niveau de précision pour être pratiquement utiles. Même aujourd'hui, ces limites computationnelles sont déjà atteintes lorsqu'il s'agit de problèmes de chimie quantique de petite ou moyenne taille. Dans ce manuscrit, on se concentre sur deux problèmes contribuant tous deux à un objectif commun : résoudre l'équation de Schrödinger dans un contexte de grande dimension.

Cette thèse comporte quatre chapitres qui peuvent être décrits dans l'ordre suivant.

Le chapitre 1 introduit les notations matricielles et tensorielles, les techniques de décomposition à rang faible qui conduisent à des factorisations exactes ou approximatives de matrices ou de tenseurs à rang faible. En particulier, on revisite la décomposition TT, mettant en évidence ses avantages pour effectuer efficacement des opérations arithmétiques au sein du format TT. Par ailleurs, on introduit deux algorithmes connus pour représenter des tenseurs en format TT ou les compresser en conséquence, à savoir la TT-SVD et le TT-rounding, respectivement. En outre, notre intérêt se porte sur des applications pratiques dans le domaine de la chimie quantique, en mettant particulièrement l'accent sur la résolution de l'équation de Schrödinger indépendante du temps. Au fil des années, les méthodes et algorithmes élaborés pour rendre ce problème complexe gérable ont révélé plusieurs étapes coûteuses nécessitant une attention particulière. Cela inclut notamment le traitement efficace du tenseur d'ordre 4 des intégrales biélectroniques, un composant essentiel présent dans de nombreuses méthodes de chimie quantique. La discussion est ensuite étendue à l'une des méthodes d'optimisation les plus renommées dans ce contexte, à savoir le DMRG, et son utilisation pour ajuster variationnellement la représentation de la fonction propre du Hamiltonien en TT. Le formalisme de la seconde quantification est également revu, ainsi que l'expression de l'opérateur Hamiltonien moléculaire. On fournit également des informations sur les complexités des étapes de calcul les plus coûteuses dans le DMRG.

Le chapitre 2 se concentre sur l'évaluation numérique de la fonction multivariée (tenseur) qui représente la composante à longue portée du potentiel de Coulomb à séparation de portée. Cela conduit à l'introduction de deux nouvelles méthodes d'approximation pour évaluer numériquement la partie à longue portée du potentiel de Coulomb à séparation de portée et les intégrales bi-électroniques à longue portée. Cela est réalisé en s'appuyant sur l'interpolation Chebyshev tensorielle, la règle de quadrature gaussienne combinée avec des approximations à rang faible ainsi que des méthodes déjà établies pour résoudre les problèmes à N corps, telles que la FMM. De plus, on étend ces approches pour approximer le tenseur d'ordre 4 d'intégrales bi-électroniques à longue portée. Nous présentons en détail les algorithmes qui sous-tendent ces méthodes, et on présente également une comparaison détaillée entre les approches introduites. Il est à noter que ce chapitre intègre le contenu de notre article publié disponible dans [4].

Le chapitre 3 se concentre principalement sur la représentation TT de l'opérateur Hamiltonien, qui est au cœur de l'algorithme DMRG. On commence par fournir un aperçu des méthodes existantes pour construire une représentation TT exacte ou approximative de l'opérateur Hamiltonien. On revisite l'approche naïve impliquant l'algorithme de TT-rounding et montrons comment il peut introduire des instabilités numériques : perte de symétrie, rupture des relations de commutation impliquant l'opérateur Hamiltonien, et apparition d'interactions virtuelles, à plus de deux corps, pour un opérateur à deux corps. On explique du point de vue théorique l'origine de certaines de ces instabilités numériques. On démontre ensuite la présence d'une structure creuse par blocs dans la représentation TT d'un opérateur Hamiltonien à p corps ($p \in \mathbb{N}$), c'est-à-dire un opérateur qui implique au plus des interactions à p corps, lorsqu'il commute avec l'opérateur de nombre de particules. Cette partie de notre travail s'inspire du travail de [2]. Bien que l'existence de cette structure inhérente soit reconnue au sein de la communauté de la physique et de la chimie, nous offrons ici une dérivation constructive directe. À notre connaissance, la représentation de la structure creuse par blocs dans la représentation TT de l'opérateur Hamiltonien n'a pas été présentée précédemment de cette manière. Poursuivant dans cette

voie, on propose une analyse approfondie de la structure de la représentation TT de la fonction propre de l'opérateur, notamment lorsque des conditions supplémentaires telles que des symétries non-Abéliennes, comme SU(2) correspondant à la conservation du spin total, sont considérées. Notre objectif est d'offrir des perspectives théoriques nouvelles, lorsque cela est nécessaire, qui n'ont pas reçu une attention significative auparavant.

Dans le chapitre 4, on présente une partie importante de notre travail consacrée à la réalisation d'une bibliothèque prototype en Julia, conçue pour le modèle de l'opérateur Hamiltonien de chimie quantique qui intègre la conservation du nombre de particules. On exploite ici la représentation TT creuse par blocs de l'opérateur ainsi que celle de sa fonction propre associée. Tout d'abord, on donne un aperçu des logiciels basés sur des tenseurs couramment utilisés pour les calculs de structure électronique. Ensuite, on explique la motivation derrière la conception de notre propre bibliothèque et on fournit des algorithmes utilisés pour exploiter la structure creuse par blocs dans les représentations TT. Avec ces structures, notre objectif est d'accélérer les étapes les plus coûteuses de QC-DMRG, notamment les contractions de tenseurs, les opérations matrice-vecteur et les compressions de matrices par SVD. De plus, on présente des résultats empiriques de diverses simulations moléculaires, tout en comparant les performances de notre bibliothèque avec la bibliothèque ITensors de pointe, où on montre qu'on atteint des performances similaires. Il est à noter que cette bibliothèque est destinée à être accessible. Elle vise à permettre aux chercheurs, au sein de notre groupe et au-delà, de réaliser leurs propres simulations DMRG ou d'explorer des approches créatives pour améliorer ses performances. Cette bibliothèque sera rendue publiquement accessible et sa parallélisation est en cours.

Dans la dernière partie de cette thèse, on présente nos conclusions et perspectives futures. En ce qui concerne les annexes, dans l'annexe .1, on fournit un pseudo-algorithme implémenté en C++ qui utilise la bibliothèque *defmm*, comme introduite dans le chapitre 2. Cette bibliothèque est centrale pour la deuxième méthode numérique qu'on introduit pour l'évaluation du tenseur d'intégrales bi-électroniques à longue portée. De plus, dans l'annexe .2, on présente une autre application de la première méthode numérique dans le chapitre 2, spécifiquement dans l'évaluation des matrices d'échange et Hartree-Fock utilisées dans les calculs Hartree-Fock [44]. Dans les annexes .3 et .4, on offre des explications concises des méthodes analytiques existantes utilisées pour construire une représentation TT exacte de l'opérateur Hamiltonien de la chimie quantique, comme introduit dans le chapitre 1. Dans l'annexe .5, on examine une interprétation graphique de la structure de la représentation TT de la fonction propre lorsque la conservation du nombre de particules est maintenue. On montre qu'une fonction propre, étant un vecteur propre d'un opérateur du nombre de particules avec une valeur propre N (où N représente le nombre de particules fixé dans le système) peut être construite à travers une chaîne récursive de transformations orthogonales en utilisant les TT-cores de la représentation TT des fonctions propres. De plus, les rangs TT théoriques peuvent être dérivés à l'aide d'une illustration graphique.

Cette thèse a conduit aux publications et pré-publications suivantes :

- **Journal paper published** S.Badreddine, I.Chollet, and L.Grigori. *Factorized structure of the long-range two-electron integrals tensor and its application in quantum chemistry.* In: *Journal of Computational Physics.*
- **Journal paper in preparation** S.Badreddine, M.Dupuy, E.Cancès, and L.Grigori. *Sparse and symmetry preserving compression of tensor trains arising in QC-DMRG.*
- **Journal paper in preparation** S.Badreddine, M.Dupuy, E.Cancès, L.Grigori, D.Torres. *Algorithmic design and numerical experiments with QC-DMRG.*

Ce projet a reçu un financement du Conseil européen de la recherche (ERC) dans le cadre du programme de recherche et d'innovation Horizon 2020 de l'Union européenne (accord de subvention n° 810367).

Introduction

In today's data-driven world, information representation often relies on matrices and tensors, i.e, multidimensional generalizations of matrices. These mathematical tools have found use in various fields, ranging from pure mathematics to applied disciplines like physics, chemistry, and economics. They have been useful in solving real-world challenges for decades. However, the recent and rapidly advancing technological developments bring new mathematical and computational challenges, paving the way to new research directions. To tackle these new challenges, continued progress in the realm of linear and multilinear algebra is required.

One of the foremost challenges today revolves around dealing with large set of data arising from various real-life problems, such as the numerical simulations of physical models for simulating any system that consists of real physical components. This can be encountered in many fields, including astrophysics, biology, climatology, quantum chemistry etc.This problem is commonly known as the *curse of dimensionality*, a term coined by Bellmann in the context of approximation theory [8]. The curse of dimensionality refers to the necessity of employing an exponentially increasing number of degrees of freedom as the dimensionality of the problem grows. Therefore, to tackle these problems, two main avenues are available to us: one is the continuous development of powerful and expensive hardware to tackle these high-dimensional problems, and the other is the development of innovative algorithms that render the high-dimensionality tractable. Concerning the latter, one can study and exploit the underlying structures of the matrices or tensors involved in representing the high-dimensional problems. One established linear algebra technique for addressing this issue involves the low-rank approximation method, achieved through truncated Singular Value Decomposition (SVD) for matrices, for example. Notably, the SVD has found a wide-ranging utility across many disciplines, including data science, astronomy, quantum chemistry, signal processing, and climate science. As the amount of data needed continue to expand, it becomes increasingly pertinent to explore other formats of data representations such as tensors.

It is worth noting that tensors have been known and employed for decades, under varying notations and concepts within different communities, notably physics, chemistry, and mathematics. The term *tensors* initially emerged in the context of mechanics, introduced by Hamilton in 1854 [41]. Subsequently, between 1880 and 1916, tensors found application as a generalization of scalars, vectors, and matrices within the realms of differential geometry and physics, notably in Einstein's theory of general relativity [97]. Between 1927 and 2011, the concept of tensor decomposition emerged as elucidated by Hitchcock in 1927 [46], allowing the representation of high-dimensional tensors as a series of lower-dimensional components. Consequently, between 1960 and 1970 many factorization algorithms were proposed such as the Canonical Decomposition (CP), and Tucker decomposition [68]. Furthermore, in the years 2010 and 2011, Grasedyck introduced the hierarchical Tucker decomposition [37], and Osledets described the tensor train format (TT-format) [92] , which corresponds to the concept of Matrix Product State (MPS) [17], previously explored within the physics community as early as 1989. This concept of tensor network, later became intertwined with the Density Matrix Renormalization Group (DMRG) algorithm, also known as the Quantum Chemical DMRG (QC-DMRG) in the realm of quantum chemistry, originated as a successful approach pioneered by White in 1992 [131]. This algorithm serves as a popular optimization technique to approximate the ground state, i.e lowest-energy state, of strongly correlated quantum systems.

In this work, our goal is to develop new numerical algorithms and study existing numerical methods to address high-dimensional challenges in quantum chemistry. We will narrow our focus to two problems in quantum chemistry, both contributing to a

common goal: solving the fundamental equation of quantum chemistry, the electronic Schrödinger equation.

To begin, we consider the approximation of a multivariate function that defines the long-range component of the range-separated Coulomb potential. Here, the Coulomb potential is defined by $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$. Then, we focus on the numerical approximation of the long-range two-electron integrals tensor, an important component in various quantum chemistry methods. Our attention shifts then to the application of tensor networks, particularly the TT decomposition, for solving the Schrödinger equation. Indeed, this equation defines a multidimensional problem whose dimension increases linearly with the number of the electrons in the system and the task of solving it using basic discretization methods can quickly lead to high-dimensional, curse of dimensionality, and computationally demanding problems. Simultaneously, the solutions must maintain a high level of accuracy to be practically useful. At this stage, the use of tensor networks in particular tensor trains and low-rank decomposition methods turns out to be useful to numerically solve this equation and break the curse of dimensionality. However, when diving into the literature on tensor networks, one encounters a notable barrier: results obtained from different communities, such as mathematics, physics, or chemistry, are not always easily communicated between communities. Consequently, one can find several papers addressing the same problem but using different terminologies, and some already established results from one community could be rediscovered by another. Therefore, in the second part of the work, the aim is to understand and interpret the results obtained from the physics and chemistry communities and seek to offer novel theoretical insights that, to the best of our knowledge, have not received significant attention before. Within this context, we focus on the DMRG algorithm combined with tensor networks, in particular the TT decomposition. We mainly study the tensor train representation of the operator, referred to as TTO, achieved through low-rank approximation method and particularly when physical conservation laws are preserved, which manifest as group symmetries in the tensors. This thesis has four chapters that can be described in the following order.

Chapter 1 gives an overview on high-dimensional problems arising in quantum chemistry and associated tensor representations. It starts by introducing notations and basic operations for tensors. It then discusses low-rank decomposition techniques that lead to either exact or approximate decompositions for both matrices and tensors. In particular, for tensors, we revisit the TT decomposition, highlighting its benefits in performing efficiently arithmetic operations within the TT format. Furthermore, we introduce two known algorithms for representing tensors in TT format or compressing them accordingly. Additionally, we go towards practical applications within the realm of quantum chemistry, such that the concern centers around solving the time-independent Schrödinger equation. Over the years, problem-adapted methods and algorithms designed to make this complex problem tractable showed to have several time-consuming steps which need careful consideration. One such critical step involves the efficient treatment of the two-electron integrals 4-th order tensor, an important component that appears in many quantum chemistry methods. The discussion is then further extended to one of the most renowned optimization methods in this context, namely, DMRG, and its use to variationally tune tensor train representation of the eigenfunction of the Hamiltonian operator. The second quantization formalism is reviewed as well as the expression of the quantum chemical Hamiltonian operator. We also provide some insights into the time complexities of the most time-consuming computational steps in DMRG.

Chapter 2 focuses on the numerical evaluation of the multivariate function (tensor) that represents the long-range component of the long-range Coulomb potential. This yields to introducing two novel approximation methods for numerically evaluating the long-range part of the range-separated Coulomb potential and the long-range two-electron integrals. This is achieved by relying on the tensorized Chebyshev interpolation, Gaussian quadrature rule combined with low-rank approximations as well as already established methods for solving N-body problems, FMM. Furthermore, we extend these approaches to approximate the high dimensional long-range two-electron integrals tensor. We provide comprehensive insights into the algorithms behind these methods, and we also provide a detailed comparison between the introduced approaches. It is worth noting that this chapter incorporates content from our published paper available in [4].

Chapter 3 focuses primarily on the TTO representation of the Hamiltonian operator which is at the core of the DMRG algorithm. We start by providing an overview of existing methods to construct an exact or approximated TTO representation of the Hamiltonian operator. We revisit the naive approach involving the TT-rounding algorithm and show how it can introduce numerical instabilities: loss of symmetry, breakdown of the commutation relations involving the Hamiltonian operator, and the occurrence of more than 2-body interactions for a strictly 2-body operator, an operator that involves at most 2-body interactions. We explain from a theoretical point of view the origin of some of these numerical issues. We subsequently demonstrate the existence of a block-sparse structure within the TT-cores of the TTO representation for a p-body Hamiltonian operator ($p \in \mathbb{N}$), particularly when it commutes with the particle number operator. This part of our work is inspired by the contributions of [2]. While the existence of this inherent structure is acknowledged within the physics and chemistry community, we offer a straightforward constructive derivation. To the best of our knowledge, the specific representation of the block-sparse structure within the TT-cores of the TTO representation of the Hamiltonian operator has not been presented previously in this manner. Continuing in this fashion, we provide a comprehensive study of the structure of the TT-cores in the TT representation of the eigenfunction. We are interested in the underlying structure of the TT-cores when additional conditions are applied, such as non-Abelian symmetries like SU(2) symmetry corresponding to the total spin conservation. Our aim is to offer novel theoretical insights, when necessary, that have not received significant attention before.

Chapter 4, is dedicated to the realization of a proof-of-concept QC-DMRG Julia library, designed for the quantum chemical Hamiltonian operator model which incorporates particle number conservation. We exploit here the block-sparse TTO representation of the operator as well as it's associated eigenfunction. First, we give an overview of existing tensor-based software packages commonly employed for electronic structure calculations, i.e the calculation of electronic states and energies. Then, we explain the motivation behind designing our own library and provide algorithms used to exploit the block sparsity within the TT representations. With these structures, our goal is to speed up the most time consuming steps in QC-DMRG, including tensor contractions, matrix-vector operations, and matrix compression through SVD. Furthermore, we provide empirical results from various molecular simulations, while comparing the performance of our library with the state-of-the-art ITensors library where we show that we attain a similar performance. It is worth noting that our library is intended to be accessible and user-friendly. It is intended to allow researchers, within our group and beyond, to carry out their own DMRG simulations or explore creative approaches to enhance its performance. Our library will be made publicly available and its parallelization is ongoing effort.

In the final part of this thesis, we present our conclusions and future perspectives as well as the appendices, that contain supplementary material. In particular, in Appendix .1, we provide a pseudo-algorithm implemented in C++ that uses the *defmm* library, as introduced in Chapter 2. This library is central to the second numerical method we introduce for the evaluation of long-range two-electron integrals tensor. Additionally, in Appendix .2, we provide another application of the first numerical method in Chapter 2, specifically for the evaluation of Hartree-Fock exchange matrices employed in Hartree-Fock calculations [44]. In Appendices .3 and .4, we offer concise explanations of the existing analytical methods used to construct an exact TTO representation, i.e without low-rank approximation, of the quantum chemical Hamiltonian operator, as introduced in Chapter 1. In Appendix .5, we review an interesting graphical interpretation of the TT representation structure of the eigenfunction when particle number conservation is maintained. We show that an eigenfunction, serving as an eigenvector of a particle number operator with eigenvalue N (representing the fixed number of particles in the system), can be constructed recursively. This construction involves a recursive sequence of orthogonal transformations using the TT-cores from the TT-representation of eigenfunctions. Theoretical minimal TT-ranks are then derived through a graphical illustration. In Appendix .6, a diagram is presented, offering an overview of key structures within our proof-of-concept library.

This thesis has led to the following publications and upcoming pre-prints

- **Journal paper published** S.Badreddine, I.Chollet, and L.Grigori. *Factorized*

structure of the long-range two-electron integrals tensor and its application in quantum chemistry. In: *Journal of Computational Physics.*

- **Journal paper in preparation** S.Badreddine, M.Dupuy, E.Cancès, and L.Grigori. *Sparse and symmetry preserving compression of tensor trains arising in QC-DMRG.*.

- **Journal paper in preparation** S.Badreddine, M.Dupuy, E.Cancès, L.Grigori, D.Torres. *Algortihmic design and numerical experiments with QC-DMRG.*

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 810367).

Chapter 1

An overview of high-dimensional problems in quantum chemistry and associated tensor representations

Contents

In this section, we start by introducing the most frequently used mathematical objects and notations in this work. We provide as well some definitions of basic operations, linear and multilinear operations, commonly performed on matrices and tensors. We begin by introducing four essential operations in the matrix and tensor framework: the tensor product, the Kronecker product, the Hadamard product, and the Khatri-Rao product. Then, we give background on low-rank approximation methods for matrices and tensors particularly relevant for addressing high-dimensional problems like those encountered when solving the Schrödinger equation in a high-dimensional context. This guides the reader into a brief introduction of essential concepts in quantum chemistry necessary to understand the topics covered in our manuscript. The background information provided contains the many-body Schrödinger equation, definitions of one and two-electron integrals tensors, the second quantization formalism, and an explanation of how tensor networks are applied in quantum chemistry, combined with the well-known DMRG algorithm.

1.1 Notations and definitions

- Scalars are either lowercase letters x, y, z, α, β, γ or uppercase latin letters N, M, T. Vectors are denoted by lowercase boldface letters such as $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, matrices are denoted by uppercase boldface letters such as $\mathbf{A}, \mathbf{B}, \mathbf{C}$, tensors are denoted by calligraphic symbols such as $\mathcal{A}, \mathcal{B}, \mathcal{C}$.
- Generic algebraic field is denoted by $\mathbb{A}$, vector spaces are denoted by $\boldsymbol{\mathcal{A}}$ and linear or multilinear operators on finite-dimensional vector spaces are denoted by $\hat{A}$.
- $[N]$defines the set $\{i \in \mathbb{N} \mid 1 \leq i \leq N\}$.

- The object $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ is a matrix with n_1 rows and n_2 columns and $\mathbf{A}^* \in \mathbb{R}^{n_2 \times n_1}$ is its transpose. $\mathbf{0}_{n_1 \times n_2} \in \mathbb{R}^{n_1 \times n_2}$ denotes a matrix with zero entries. Occasionally, when notations become complex, we may use only the notation $\mathbf{0}$ for simplicity.
- The object $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a tensor. Here, $d \in \mathbb{N}$ is called the order of the tensor, and n_k defines the k-th mode size for $k \in [d]$, i.e., the size of the k-th dimension in the tensor.
- $\boldsymbol{a}(i)$ is the i-th entry of the vector $\boldsymbol{a}$, $\mathbf{A}(i;j)$ is the (i,j)-th entry of the matrix $\mathbf{A}$, $\mathcal{A}(i_1;\ldots;i_d)$ is the $(i_1,\ldots,i_d)$-th entry of the tensor $\mathcal{A}$ of order $d \in \mathbb{N}$.
- $\mathbf{A}[:,j]$ (Julia/Matlab notations) denotes the subvector containing the column of $\mathbf{A}$ indexed by j, $\mathbf{A}[j,:]$ denotes the subvector containing the row of $\mathbf{A}$ indexed by j, $\mathcal{A}[:,:,j]$ denotes the submatrix extracted from $\mathcal{A}$ at index j, $\mathcal{A}[:,:,:,j]$ denotes the subtensor extracted from $\mathcal{A}$ at index j associated with the fourth mode.
- $\mathbf{A}|_{S_i \times S_j}$ denotes the restriction of the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ to a submatrix indexed by a couple (i,j) of size $S_i \times S_j$, i.e, we consider only the rows and columns specified by the dimensions i and j, respectively, from the original matrix $\mathbf{A}$.
- $\|\boldsymbol{x}-\boldsymbol{y}\| = \sqrt{(\boldsymbol{x}_1-\boldsymbol{y}_1)^2+(\boldsymbol{x}_2-\boldsymbol{y}_2)^2+(\boldsymbol{x}_3-\boldsymbol{y}_3)^2}$ is the euclidean distance between two points $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3$ with coordinates $(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3),(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}_3)$ respectively.
- The Kronecker delta is
$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$
- $\|.\|_F$ is the Frobenius norm.
- $\mathbf{I}_n$ is the identity matrix of size $n \times n$ and $\mathbf{I}_{n_1 \times n_2}$ is the identity matrix of size $n_1 \times n_2$.
- $|x|$ is the absolute value of x.
- $\#$ is the cardinality (or size) of a given set, while dim() denotes the dimension of a finite-dimensional vector space.
- $\otimes_K$ is the Kronecker product.
- $\otimes$ is the tensor product.
- $\diamond$ is the row-wise Khatri-Rao product and $\bullet$ is the column-wise Khatri-Rao product.
- $\odot$ is the Hadamard product.
- $\oplus$ is the Direct sum.
- $\langle \cdot,\cdot \rangle$ is the inner product.
- $\cong$ is the symbol that represents isomorphism between vector spaces.
- $\equiv$ is the equivalent symbol.
- $\approx$ is the symbol denoting approximate equality.

Definition 1. (Kronecker product) The Kronecker product between matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times m_2}$ which we denote $\mathbf{A} \otimes_K \mathbf{B} \in \mathbb{R}^{n_1 n_2 \times m_1 m_2}$, is defined as:

$$\mathbf{A} \otimes_K \mathbf{B} = \begin{bmatrix} \mathbf{A}(1;1)\mathbf{B} & \mathbf{A}(1;2)\mathbf{B} & \cdots & \mathbf{A}(1;m_1)\mathbf{B} \\ \mathbf{A}(2;1)\mathbf{B} & \mathbf{A}(2;2)\mathbf{B} & \cdots & \mathbf{A}(2;m_1)\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}(n_1;1)\mathbf{B} & \mathbf{A}(n_1;2)\mathbf{B} & \cdots & \mathbf{A}(n_1;m_1)\mathbf{B} \end{bmatrix} \tag{1.1}$$

.

Remark 1.1.1. For ease of notation, we can also use the compact product notation $\bigotimes_{k=1}^{d}{}_K$. Given d matrices $\mathbf{A}_k \in \mathbb{R}^{n_k \times m_k}, k \in [d]$, the Kronecker product of d matrices denoted as $\bigotimes_{k=1}^{d}{}_K \mathbf{A}_k \in \mathbb{R}^{\left(\prod_{k=1}^{d} n_k\right) \times \left(\prod_{k=1}^{d} m_k\right)}$, is given by:

$$\bigotimes_{k=1}^{d}{}_K \mathbf{A}_k = \mathbf{A}_1 \otimes_K \mathbf{A}_2 \otimes_K \ldots \otimes_K \mathbf{A}_d. \tag{1.2}$$

Definition 2. (Tensor product) Let $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ be vector spaces over the same field $\mathbb{F}$. If $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ are finite-dimensional spaces with respective basis $B_{\boldsymbol{\mathcal{U}}} = \{\boldsymbol{u}_i, i \in [I]\}$ and $B_{\boldsymbol{\mathcal{V}}} = \{\boldsymbol{v}_j, [J]\}$, then, the tensor product $\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{V}}$ has basis $B_{\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{V}}} = \{\boldsymbol{u}_i \otimes \boldsymbol{v}_j, i \in [I], j \in [J]\}$.

Remark 1.1.2. When discussing the tensor product between vectors, it is important to note that we are referring to the outer product, denoted by the symbol $\otimes$ as well. For example, the tensor product between two vectors $\boldsymbol{v} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$, denoted as $\boldsymbol{v} \otimes \boldsymbol{u}$, is defined as an $n \times m$ matrix. This tensor product is a special case of the Kronecker product, and it can be expressed as:

$$\boldsymbol{v} \otimes \boldsymbol{u} = \boldsymbol{v} \otimes_K \boldsymbol{u}^*. \tag{1.3}$$

Definition 3. (Hadamard product) The Hadamard product between matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{B} \in \mathbb{R}^{n_1 \times n_2}$ which we denote $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{n_1 \times n_2}$, is defined as:

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{A}(1;1)\mathbf{B}(1;1) & \mathbf{A}(1;2)\mathbf{B}(1;2) & \cdots & \mathbf{A}(1;n_2)\mathbf{B}(1;n_2) \\ \mathbf{A}(2;1)\mathbf{B}(2;1) & \mathbf{A}(2;2)\mathbf{B}(2;2) & \cdots & \mathbf{A}(2;n_2)\mathbf{B}(2;n_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}(n_1;1)\mathbf{B}(n_1;1) & \mathbf{A}(n_1;2)\mathbf{B}(n_1;2) & \cdots & \mathbf{A}(n_1;n_2)\mathbf{B}(n_1;n_2). \end{bmatrix}. \tag{1.4}$$

Definition 4. (Column-wise Khatri-Rao product) Consider two matrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}[:,1] & \mathbf{A}[:,2] & \ldots & \mathbf{A}[:,m_1] \end{bmatrix} \in \mathbb{R}^{n_1 \times m_1},$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}[:,1] & \mathbf{B}[:,2] & \ldots & \mathbf{B}[:,m_1] \end{bmatrix} \in \mathbb{R}^{n_2 \times m_1},$$

where $\mathbf{A}[:,k] \in \mathbb{R}^{n_1 \times 1}$ and $\mathbf{B}[:,k] \in \mathbb{R}^{n_2 \times 1}$ for each $k \in [m_1]$. The column-wise Khatri-Rao product between $\mathbf{A}$ and $\mathbf{B}$ which we denote $\mathbf{A} \bullet \mathbf{B} \in \mathbb{R}^{(n_1 n_2) \times m_1}$, is defined as:

$$\mathbf{A} \bullet \mathbf{B} = \begin{bmatrix} \mathbf{A}[:,1] \otimes_K \mathbf{B}[:,1] & \mathbf{A}[:,2] \otimes_K \mathbf{B}[:,2] & \ldots & \mathbf{A}[:,m_1] \otimes_K \mathbf{B}[:,m_1] \end{bmatrix}, \tag{1.5}$$

The row-wise, that we denote by the symbol $\diamond$ is related to the column-wise Khatri-Rao product as follows:

$$(\mathbf{A} \bullet \mathbf{B})^* = \mathbf{A}^* \diamond \mathbf{B}^*. \tag{1.6}$$

Remark 1.1.3. For ease of notation, we use the following compact notation. Given d matrices $\mathbf{A}_k \in \mathbb{R}^{n_1 \times m_k}, k \in [d]$, the row-wise Khatri-Rao product of d matrices, denoted as $\diamond_{k=1}^{d} \mathbf{A}_k \in \mathbb{R}^{n_1 \times \left(\prod_{k=1}^{d} m_k\right)}$, is given by:

$$\diamond_{k=1}^{d} \mathbf{A}_k = \mathbf{A}_1 \diamond \mathbf{A}_2 \diamond \ldots \diamond \mathbf{A}_d, \tag{1.7}$$

and given d matrices $\mathbf{A}_k \in \mathbb{R}^{n_k \times m_1}, k \in [d]$, the column-wise Khatri-Rao product of d matrices, denoted as $\bullet_{k=1}^{d} \mathbf{A}_k \in \mathbb{R}^{\left(\prod_{k=1}^{d} n_k\right) \times m_1}$, is given by:

$$\bullet_{k=1}^{d} \mathbf{A}_k = \mathbf{A}_1 \bullet \mathbf{A}_2 \bullet \ldots \bullet \mathbf{A}_d. \tag{1.8}$$

Definition 5. (Direct sum) The direct sum between matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$ and $\mathbf{B} \in \mathbb{R}^{n_2 \times m_2}$ is defined as the block diagonal matrix $\mathbf{A} \oplus \mathbf{B} \in \mathbb{R}^{(n_1+n_2) \times (m_1+m_2)}$ such that

$$\mathbf{A} \oplus \mathbf{B} = \operatorname{diag}(\mathbf{A}, \mathbf{B}) = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{n_1 \times m_2} \\ \mathbf{0}_{n_2 \times m_1} & \mathbf{B} \end{bmatrix}. \tag{1.9}$$

Remark 1.1.4. We will also encounter the direct sum of finite-dimensional vector spaces, we denote as well by $\oplus$, which is defined as follows: let $\boldsymbol{\mathcal{V}}$ be a vector space, let $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{W}}$ be two subspaces such that for every $\boldsymbol{v} \in \boldsymbol{\mathcal{V}}$,

- there exist vectors $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}, \boldsymbol{w} \in \boldsymbol{\mathcal{W}}$ such that $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{w}$,
- If $\boldsymbol{v} = \boldsymbol{u}_1 + \boldsymbol{w}_1$ and $\boldsymbol{v} = \boldsymbol{u}_2 + \boldsymbol{w}_2$ where $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \boldsymbol{\mathcal{U}}$, $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \boldsymbol{\mathcal{W}}$ then $\boldsymbol{u}_1 = \boldsymbol{u}_2$ and $\boldsymbol{w}_1 = \boldsymbol{w}_2$.

Then $\boldsymbol{\mathcal{V}}$ is the direct sum of $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{W}}$. We write $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{U}} \oplus \boldsymbol{\mathcal{W}}$.

In the following proposition, we give useful relations among these matrix products that we will use in our derivations.

Proposition 1.1.1 ([81])**.** *Consider matrices* $\mathbf{A} \in \mathbb{R}^{I_1 \times J_1}$, $\mathbf{B} \in \mathbb{R}^{I_1 \times J_2}$, $\mathbf{C} \in \mathbb{R}^{J_1 \times J_3}$, *and* $\mathbf{D} \in \mathbb{R}^{J_2 \times J_4}$, *then*

$$(\mathbf{A} \otimes_K \mathbf{B})(\mathbf{C} \otimes_K \mathbf{D}) = (\mathbf{AC}) \otimes_K (\mathbf{BD}). \tag{1.10}$$

Additionally,

$$(\mathbf{A} \diamond \mathbf{B})(\mathbf{C} \otimes_K \mathbf{D}) = (\mathbf{AC}) \diamond (\mathbf{BD}). \tag{1.11}$$

Additionally, if $J_4 = J_3$, *it holds that*

$$(\mathbf{A} \diamond \mathbf{B})(\mathbf{C} \bullet \mathbf{D}) = (\mathbf{AC}) \odot (\mathbf{BD}). \tag{1.12}$$

In what follows, we provide the definition of tensors and basic multilinear operations. We start by viewing tensors from a practical point of view as commonly used in computer science, where they are regarded as multidimensional arrays (data structures). Just as a matrix represents a two-dimensional array, a tensor extends this concept to higher dimensions. A visual representation of tensors with different orders is given in Figure 1.3,

$$\begin{array}{|cc|} \hline \mathbf{A}(1;1) & \mathbf{A}(1;2) \\ \mathbf{A}(2;1) & \mathbf{A}(2;2) \\ \hline \end{array}$$

Figure 1.1: 2-order tensor/matrix of dimension 2×2.

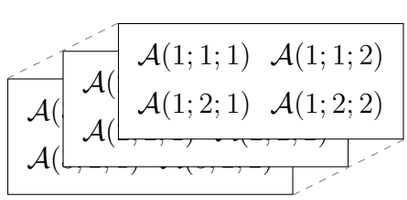


Figure 1.2: 3-order tensor of dimension $2 \times 2 \times 3$.

Figure 1.3: Tensors of order $d \in \{2, 3\}$.

It should be noted that, from a multilinear algebra point of view, tensors can be seen as elements of the tensor product of linear spaces denoted by $\boldsymbol{\mathcal{V}}$. Given a vector space $\boldsymbol{\mathcal{V}}$ of dimension $n \in \mathbb{N}$, let $B_{\boldsymbol{\mathcal{V}}} = \{\boldsymbol{v}_i, i \in [n]\}$ be a basis of $\boldsymbol{\mathcal{V}}$ so any element $\boldsymbol{v}$ in $\boldsymbol{\mathcal{V}}$ can be written as follows:

$$\boldsymbol{v} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i, \tag{1.13}$$

with $\lambda_i \in \mathbb{R}, i \in [n]$.

Definition 6. Let $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ be d vector spaces with respective dimension $n_k \in \mathbb{N}$, and respective basis $B_{\boldsymbol{\mathcal{V}}_k} = \{\boldsymbol{v}_{i_k}, i_k \in [n_k]\}$ for $k \in [d]$. According to Definition 2, $\boldsymbol{\mathcal{V}}_1 \otimes \cdots \otimes \boldsymbol{\mathcal{V}}_d$ is a vector space with the following basis:

$$B_{\boldsymbol{\mathcal{V}}_1 \otimes \ldots \otimes \boldsymbol{\mathcal{V}}_d} = \{\boldsymbol{v}_{i_1} \otimes \cdots \otimes \boldsymbol{v}_{i_d}, 1 \leq i_1 \leq n_1, \cdots, 1 \leq i_d \leq n_d\}.$$

By linearity, any tensor $\boldsymbol{\mathcal{A}} \in \boldsymbol{\mathcal{V}}_1 \otimes \cdots \otimes \boldsymbol{\mathcal{V}}_d$ can be written as follows:

$$\boldsymbol{\mathcal{A}} = \sum_{i_1=1}^{n_1} \ldots \sum_{i_d=1}^{n_d} \mathcal{A}(i_1; \cdots; i_d) \boldsymbol{v}_{i_1} \otimes \cdots \otimes \boldsymbol{v}_{i_d}. \tag{1.14}$$

Here, $\boldsymbol{\mathcal{A}}$ is called a d-order tensor with entries $\mathcal{A}(i_1; \cdots; i_d)$. Therefore, a tensor, which is an element of a tensor product space, can be identified as a multidimensional array once a basis for the tensor product is established.

Another important interpretation of tensors is that they are regarded as multilinear operators analogous to the matrix case. For example, consider a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$, which describes the action of a linear operator from $\mathbb{R}^{m_1}$ to $\mathbb{R}^{n_1}$ with respect to a given basis on both spaces. In the case of tensors, we can define a multilinear operator as $\hat{A} : \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_{d_1}} \to \mathbb{R}^{m_1} \otimes \cdots \otimes \mathbb{R}^{m_{d_2}}$, where $\mathcal{A} \in \mathbb{R}^{(n_1 \times \cdots \times n_{d_1}) \times (m_1 \times \cdots \times m_{d_2})}$ is the multidimensional array representing the multilinear operator with fixed basis for tensor product spaces $\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_{d_1}}$ and $\mathbb{R}^{m_1} \otimes \cdots \otimes \mathbb{R}^{m_{d_2}}$. The latter can be referred to as the tensor operator of order $d_1 + d_2$ that acts on tensors of order d_2.

In the following, we present a concise overview of various multilinear operations on tensors, which will be essential for understanding the content of this manuscript.

When working with tensors, it is possible to modify their shape or dimensions while keeping the total number of elements unchanged. This operation is commonly referred to as *reshaping* and involves rearranging the elements of a tensor to match a desired new shape. Reshaping is often used to convert tensors into a matrix or a vector, respectively, which are known as *matricization* or *unfolding* and *vectorization* operations, respectively. These operations allow for different ways of organizing and accessing the tensor elements to suit specific computational requirements or algorithms.

Definition 7. (Multi-index) For each $k \in [d]$, consider a subset $S_k = [n_k]$. Define the function $\xi : S_1 \times \ldots \times S_d \to \mathbb{N}$ such that

$$\xi(i_1; \ldots; i_d) = 1 + \sum_{l=1}^{d} \left((i_l - 1) \prod_{m=1}^{l-1} n_m \right), \tag{1.15}$$

Moving forward, the expression of a multi-index $\overline{i_1, \ldots, i_d}$ denotes the image of ξ, i.e $\xi(i_1; \ldots; i_d) := \overline{i_1, \ldots, i_d}$.

Definition 8. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}, n_k \in \mathbb{N}, k \in [d]$, be a d-order tensor. The mode-k matricization or unfolding of a tensor $\mathcal{A}$, denoted as $\mathbf{A}^{(k)} \in \mathbb{R}^{n_k \times (n_1 n_2 \cdots n_{k-1} n_{k+1} \cdots n_d)}$, $k \in [d]$, is defined element-wise as follows, for $i_k \in [n_k]$ and $j \in [n_1 \ldots n_{k-1} n_{k+1} \ldots n_d]$:

$$\mathbf{A}^{(k)}(i_k; j) = \mathbf{A}^{(k)}(i_k; \overline{i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d}), \tag{1.16}$$

where j is related to the multi-index $j := \overline{i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d}$ and has value

$$j = 1 + \sum_{\substack{l=1 \\ l \neq k}}^{d} \left((i_l - 1) \prod_{\substack{m=1 \\ m \neq k}}^{l-1} n_m \right).$$

Definition 9. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-order tensor. The mode-$(1:k)$ matricization of a tensor $\mathcal{A}$ denoted as $\mathbf{A}^{<k>} \in \mathbb{R}^{(n_1 \cdots n_k) \times (n_{k+1} \cdots n_d)}$, $k \in [d]$, is defined element-wise as follows, for $j_1 \in \left[\prod_{i=1}^{k} n_i\right]$ and $j_2 \in \left[\prod_{i=k+1}^{d} n_i\right]$:

$$\mathbf{A}^{<k>}(j_1; j_2) = \mathbf{A}^{<k>}(\overline{i_1, \cdots, i_k}; \overline{i_{k+1}, \ldots, i_d}), \tag{1.17}$$

with $j_1 = 1 + \sum_{l=1}^{k} \left((i_l - 1) \prod_{m=1}^{l-1} n_m \right)$ and $j_2 = 1 + \sum_{l=k+1}^{d} \left((i_l - 1) \prod_{m=k+1}^{l-1} n_m \right)$.

Definition 10. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a $d-th$ order tensor. The vectorization of $\mathcal{A}$, denoted as $\boldsymbol{a} = \mathrm{vec}(\mathcal{A}) \in \mathbb{R}^{\mathrm{n_1 \cdots n_d}}$ can be defined element-wise as follows, for $j \in [n_1 \ldots n_d]$:

$$\boldsymbol{a}(j) := \boldsymbol{a}(\overline{i_1, i_2, \ldots, i_d}) = \mathrm{vec}(\mathcal{A})(j) = \mathcal{A}(i_1; \ldots; i_d), \tag{1.18}$$

with $j = 1 + \sum_{l=1}^{d} \left((i_l - 1) \prod_{m=1}^{l-1} n_m \right)$.

Example 1.1.1. Let $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$ be a 3-order tensor given in the following figure

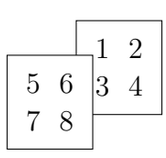


Figure 1.4: 3-order tensor of dimension $2 \times 2 \times 2$.

One way to vectorize the tensor $\mathcal{A}$ with entries $\mathcal{A}(i_1, i_2, i_3)$ involves horizontally concatenating elements $\mathcal{A}(i_1; i_2; i_3)$ while keeping indices $i_2, i_3 \in \{1, 2\}$.

$$\begin{aligned}\boldsymbol{a} = \text{vec}(\mathcal{A}) &= \begin{bmatrix}\mathcal{A}(1;1;1) & \mathcal{A}(2;1;1) & \mathcal{A}(1;2;1) & \mathcal{A}(2;2;1) & \mathcal{A}(1;1;2) & \mathcal{A}(2;1;2) & \mathcal{A}(1;2;2) & \mathcal{A}(2;2;2)\end{bmatrix}^* \\ &= \begin{bmatrix}5 & 7 & 6 & 8 & 1 & 3 & 2 & 4\end{bmatrix}^*.\end{aligned} \tag{1.19}$$

One way to obtain the mode-1 matricization of $\mathcal{A}$ is:

$$\mathbf{A}^{(1)} = \begin{bmatrix}5 & 6 & 1 & 2 \\ 7 & 8 & 3 & 4\end{bmatrix} \tag{1.20}$$

Remark 1.1.5. The matricization of a tensor can be related to the Kronecker product as follows: let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-order tensor. Suppose that $\mathcal{A}$ is expressed in the basis of $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ as follows:

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \mathcal{A}(i_1; \cdots; i_d) \boldsymbol{v}_{i_1} \otimes \cdots \otimes \boldsymbol{v}_{i_d}. \tag{1.21}$$

The mode-k matricization of $\mathcal{A}$, $k \in [d]$, using Kronecker products, can be expressed as:

$$\mathbf{A}^{(k)} = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \mathcal{A}(i_k; \overline{i_1, \cdots, i_{k-1}, i_{k+1}, \cdots, , i_d}) \boldsymbol{v}_{i_k} \otimes_K (\bigotimes_{\substack{l \neq k \\ l=1}}^{d} {}_K \boldsymbol{v}_{i_l})^*. \tag{1.22}$$

It is noted that the reshaping operations described above will be of particular interest for tensor trains that will be introduced in Section 1.2.

Remark 1.1.6. The vectorization as well as matricization (tensor unfolding) operations can be interpreted by the existence of vector space isomorphism such that [1]

$$\mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_k} \otimes \dots \otimes \mathbb{R}^{n_d} \cong \mathbb{R}^{(n_1 \dots n_k) \times (n_{k+1} \dots n_d)} \cong \mathbb{R}^{n_1 \dots n_k \dots n_d}. \tag{1.23}$$

We will now proceed to discuss one of the most commonly used operations between tensors, known as tensor contraction. This operation combines two tensors by summing over a pair of indices, where one index of the first tensor is contracted with the corresponding index of the second tensor. This is also a generalization of the matrix-vector or matrix-matrix products as described in the following definition

Definition 11. (Tensor contractions) Consider two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_{d_1}}$ and $\mathcal{B} \in \mathbb{R}^{m_1 \times \cdots \times m_{d_2}}$, of orders d_1 and d_2, respectively. For $k \in [d_1], h \in [d_2]$, if $n_k = n_h$, the contraction product between these two tensors is denoted by $\mathcal{A} \times_{kh} \mathcal{B}$ and the resulting tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times n_{k+1} \times \cdots \times n_d \times m_1 \times \cdots \times m_{h-1} \times m_{h+1} \times \cdots m_d}$ is defined element-wise as follows:

$$\mathcal{C}(i_1; \dots; i_{k-1}; i_{k+1}; \dots; i_{d_1}; j_1 \dots; j_{h-1}; j_{h+1}; \dots; j_{d_2}) = \sum_{\ell=1}^{n_k} \mathcal{A}(i_1; \dots; \ell; \dots; i_{d_1}) \mathcal{B}(j_1; \dots; \ell; \dots; j_{d_2}). \tag{1.24}$$

and we have:

$$\mathcal{C} = \mathcal{A} \times_{kh} \mathcal{B}. \tag{1.25}$$

For the sake of simplicity and consistency throughout the manuscript, we use the notation $\mathcal{A} \times_k \mathcal{B}$ to represent the contraction product between two tensors when $n_k = n_h$.

So far, we have discussed the contraction between tensors with a single common index or mode. However, this concept can be extended to involve multiple modes such that instead of only summing over the index ℓ in Equation (1.24), we can generalize the contraction to include all common indices. Therefore, in the contraction notation, we will replace the single index k with a set of common indices, denoting the contracted modes. For example, given two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_{d_1} \times m_1 \times \dots \times m_{d_2}}$ and $\mathcal{B} \in \mathbb{R}^{m_1 \times \dots \times m_{d_2} \times n'_1 \times \dots \times n'_{d_3}}$,

[1] A vector space isomorphism is a bijective linear mapping between two vector spaces that preserves the vector space structure.

the contraction product between tensors $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} \times_{\{d_1+1 \ldots d_1+d_2\}} \mathcal{B}$ and the resulting tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \ldots \times n_{d_1} \times n'_1 \times \ldots \times n'_{d_3}}$ is defined element-wise as follows:

$$\mathcal{C}(i_1; \ldots; i_{d_1}; j_1; \ldots; j_{d_3}) = \sum_{\ell_1=1}^{m_1} \ldots \sum_{\ell_{d_2}=1}^{m_{d_2}} \mathcal{A}(i_1; \ldots; i_{d_1}; \ell_1; \ldots; \ell_{d_2}) \mathcal{B}(\ell_1; \ldots; \ell_{d_2}; j_1; \ldots; j_{d_3}). \tag{1.26}$$

Furthermore, it is important to note that throughout this work, we will encounter the contraction product between a tensor and a matrix. This operation is defined as follows: let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-order tensor and $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$, if for a fixed $k \in [d]$, $n_k = m_1$, then the contraction product between $\mathcal{A}$ and $\mathbf{B}$ is denoted by $\mathcal{A} \times_k \mathbf{B}$. We keep the first common index, i.e k in the contraction symbol.

Further details about basic tensor operations as tensor unfolding, contractions and several tensor products as inner product, outer product, are provided in [3, 68, 70, 73].

1.2 Low-rank approximation for matrices and tensors

1.2.1 Introduction

In what follows, we give a brief introduction on low-rank approximation techniques for matrices and tensors that aim to represent data more efficiently by approximating high-dimensional matrices and tensors with low dimensional counterparts. We begin by introducing the Singular Value Decomposition (SVD) and QR decomposition [49] which are well-known tools used for exploiting low-rank structures in matrices. Additionally, we introduce the Tensor Train (TT) decomposition as a generalization of SVD for tensors, showcasing its potential for efficient data representation in high-dimensional settings.

1.2.2 Low-rank approximation for matrices

Singular value decomposition (SVD)

Definition 12. (Singular value decomposition (SVD)) The Singular Value Decomposition of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\mathbf{A}) = \min\{m, n\}$ is defined as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{1.27}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices, $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal matrix composed of non-negative real numbers on the diagonal called singular values. The factorization in (1.27) is equivalent to the following expression

$$\mathbf{A} = \sum_{k=1}^{\min\{m,n\}} \sigma_k \mathbf{U}[:, k] \otimes \mathbf{V}[:, k], \tag{1.28}$$

where σ_k is the k-th singular value for $k \in [\min\{m, n\}]$, $\mathbf{U}[:, k]$, respectively $\mathbf{V}[:, k]$ is the k-th column of the matrix $\mathbf{U}$, respectively $\mathbf{V}$. Additionally, the number of nonzero singular values is equal to the rank of $\mathbf{A}$.

It is known that the SVD of $\mathbf{A}$ is related to the eigenvalue decomposition of the matrices $\mathbf{A}\mathbf{A}^*$ and $\mathbf{A}^*\mathbf{A}$, with the singular values corresponding to the positive square roots of their respective eigenvalues. Additionally, for non-degenerate or distinct singular values, the SVD decomposition is said to be unique up to permutations of the columns of the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$.

Definition 13. (Low-rank matrix) $\mathbf{A} \in \mathbb{R}^{m \times n}$ is said to have low-rank if its rank, denoted by $\text{rank}(\mathbf{A}) = r$, satisfies $r \ll \min\{m, n\}$.

The aim here is to approximate a given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ by a low-rank matrix. The key idea of low-rank approximation is that a given matrix may not be necessarily of exact low-rank, meaning it does not have a decomposition that exactly reproduces the original matrix. Nevertheless, it can be well approximated by a low-rank matrix. In what follows, we describe the truncated SVD (tSVD) method to obtain a low-rank approximation of a given matrix.

Truncated singular value decomposition

Definition 14. (truncated SVD (tSVD)) Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a matrix of full rank, $\text{rank}(\mathbf{A}) = \min\{m, n\}$. For a chosen $r \in \mathbb{N}$ such that $r < \text{rank(A)}$, one can construct the so-called truncated SVD of $\mathbf{A}$ denoted by $\tilde{\mathbf{A}}$ as follows:

$$\mathbf{A} \approx \tilde{\mathbf{A}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^* = \sum_{k=1}^{r} \sigma_k \tilde{\mathbf{U}}[:, k]\tilde{\mathbf{V}}[:, k]^*, \tag{1.29}$$

where $\tilde{\mathbf{U}} \in \mathbb{R}^{m\times r}$, (resp. $\tilde{\mathbf{V}}$), is the matrix containing the first r columns of $\mathbf{U}$ (resp. $\mathbf{V}$), and $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$ is the top-left submatrix of $\mathbf{\Sigma}$, with $\sigma_k, k \in [r]$ being the truncated singular values of $\mathbf{A}$.

It is obvious that the rank of $\tilde{\mathbf{A}}$ is r. $\tilde{\mathbf{A}}$ is the best rank-r approximation of the matrix $\mathbf{A}$ when the approximation error is measured in an unitarily invariant norms: the 2-norm or the Frobenius norm. The error of this approximation is given by (showing only the Frobenius norm):

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F = \left(\sum_{k\geq r+1} \sigma_k^2\right)^{1/2}. \tag{1.30}$$

This rank-r approximation method, known as the tSVD, is a popular technique for dimensionality reduction. Indeed, if $r \ll \min\{n, m\}$, then the storage cost of the matrix $\tilde{\mathbf{A}}$ reduces significantly from $\mathcal{O}(mn)$ to $\mathcal{O}(r(m + n + 1))$. Additionally, when multiplying $\tilde{\mathbf{A}}$ with another matrix $\mathbf{B} \in \mathbb{R}^{n\times q}$, the computational cost of the product $\tilde{\mathbf{A}}\mathbf{B}$ reduces from $\mathcal{O}(mnq)$ to $\mathcal{O}(rq(m+n+1))$. However, it is worth noting that for large dimensions, computing the tSVD can become expensive as it requires first computing the full rank SVD, which costs $\mathcal{O}(mn \ \min\{m, n\})$, and then extracting the leading singular values based on a specified threshold.

From a geometric point of view, the existence of a best approximation of any matrix $\mathbf{A}$ by another matrix of rank at most r implies the following proposition (see [126])

Proposition 1.2.1. *The set of matrices of rank at most r is defined as:*

$$\mathcal{M}_{\leq r} = \left\{\mathbf{A} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{A}) \leq r\right\} \tag{1.31}$$

is a closed subset of $\mathbb{R}^{m\times n}$.

This proposition suggests that any continuous and bounded function on this set reaches a minimum within this set. The distance from this set, given by Equation (1.30), implies that the matrix $\mathbf{A}$ has a good low-rank approximation in Frobenius norm if the singular values decay sufficiently fast. This set is an algebraic variety, not smooth on the elements $\mathbf{A}$ with rank strictly less than r. Therefore, when dealing with geometrical optimization methods, these are often performed on smooth manifolds. In this case the smooth variety of $\mathcal{M}_{\leq r}$ is considered which is the set of matrices with fixed ranks $\mathcal{M}_{=r} = \{\mathbf{A} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{A}) = r\}$. While we refrain from delving into details regarding the geometrical properties of this manifold, interested readers can find comprehensive insights in [126].

QR decomposition

Throughout this work, we will encounter a popular factorization method, which is known as the QR decomposition. This decomposition is described in the following definition

Definition 15. (QR Decomposition) For a given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the QR decomposition expresses $\mathbf{A}$ as the product of an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{R}^{m\times n}$

$$\mathbf{A} = \mathbf{QR}, \tag{1.32}$$

where $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}_m$.

One common approach to compute these matrices is by employing the Gram-Schmidt procedure on the columns of $\mathbf{A}$ or alternatively the Householder transformations [50].

1.2.3 Low-rank approximation for tensors

Consider the canonical basis of $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \cdots \otimes \mathbb{R}^{n_d}$ denoted as:

$$B_{\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \cdots \otimes \mathbb{R}^{n_d}} = \{\boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_d}, 1 \leq i_1 \leq n_1, \cdots, 1 \leq i_d \leq n_d\}. \tag{1.33}$$

Then, any tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \cdots \otimes \mathbb{R}^{n_d}$ can be expressed as:

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}(i_1; \cdots; i_d) \boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_d}, \tag{1.34}$$

where $\mathcal{A}(i_1; \cdots; i_d)$ is the $(i_1, \cdots, i_d)$-th entry of $\mathcal{A}$. The number of coefficients in the example (1.34) clearly grows as $\mathcal{O}(n^d)$, with $n = \max_{1 \leq k \leq d} \{n_k\}$. This exponential dependency, known as the curse of dimensionality, poses a significant computational challenge. Various decomposition techniques have been proposed over the years to alleviate this challenge [68, 92]. These techniques aim to efficiently represent tensors through exact or approximate decompositions involving tensors with lower dimensions than the original tensor.

In the following, we first introduce a graphical representation of tensors which aims to simplify tensor notations. We describe three main tensor decompositions: the canonical decomposition (CP) decomposition, the Tucker decomposition, and the tensor train (TT) decomposition. We highlight their distinctive computational aspects and discuss why the TT decomposition is the most suitable when dealing with problems in high-dimensions, thus the focal point of Chapters 3 and 4.

Graphical representation of tensors

It is useful to visualize tensors and tensor operations by employing tensor network diagrams using nodes and edges. Figure 1.5 illustrates tensor diagrams representing a vector $\boldsymbol{v} \in \mathbb{R}^n$, a matrix $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ and a 3-order tensor $\mathcal{A} \in \mathbb{R}^{m \times n \times l}$. Each of these objects is represented by a node and one edge for each dimension along a specific mode. For instance, as illustrated in Figure 1.5(a), the vector $\boldsymbol{v} \in \mathbb{R}^n$ is represented by a node with one edge labeled with n to denote the dimension along the first mode, which is n. Edges connecting nodes represent the indices shared between tensors in a tensor contraction, as illustrated in Figure 1.5(d). This figure illustrates the contraction product denoted as $\mathcal{A} \times_2 \mathcal{B}$ of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a tensor $\mathcal{B} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$, with $n_2 = m_2$. Figure 1.5(e) represents the SVD of a matrix of size $m \times n$ of rank r using the diagrams where orthogonal matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{m \times r}$ are represented by half filled circles and the diagonal matrix $\mathbf{\Sigma} \in \mathbf{U} \in \mathbb{R}^{r \times r}$ by a white circle.

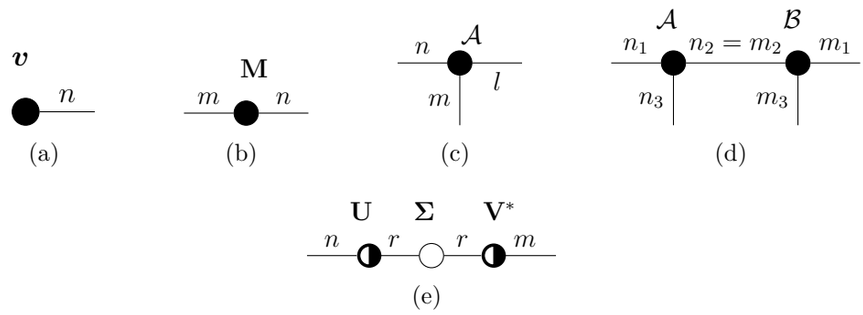


Figure 1.5: Graphical representation of (a) a vector $\boldsymbol{v} \in \mathbb{R}^n$, (b) a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, (c) 3-order tensor$\mathcal{A} \in \mathbb{R}^{m \times n \times l}$, (d) the contraction product between two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ with $n_2 = m_2$, and (e) the SVD of a matrix of size $n \times m$ and of rank r.

CP and Tucker decomposition

The (CP) [46] and *Tucker decomposition* [125] can be seen as high-order extensions of the SVD of matrices. Before delving into these decompositions, it is essential to establish the following definition regarding tensor rank.

Definition 16. (Simple tensor and tensor rank [68]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-order tensor. $\mathcal{A}$ is a simple/rank-1 tensor if it is expressed as follows:

$$\mathcal{A} = \boldsymbol{e}_1 \otimes \boldsymbol{e}_2 \cdots \otimes \boldsymbol{e}_d. \tag{1.35}$$

It follows that for a general d-order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with rank r, the rank r is defined as the minimum number of simple tensors needed to sum to generate $\mathcal{T}$.

Definition 17. (CP decomposition) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-order tensor. The CP decomposition of $\mathcal{A}$ can be expressed as the linear decomposition involving r rank-1 (simple) tensors:

$$\mathcal{A} = \sum_{i_1 = \ldots = i_d = 1}^{r} \lambda_r \boldsymbol{u}_{i_1} \otimes \cdots \otimes \boldsymbol{u}_{i_d}, \boldsymbol{u}_{i_k} \in \mathbb{R}^{n_k}, \tag{1.36}$$

where $\lambda_r \in \mathbb{R}$ and $B_{\mathbb{R}^r} = \{\boldsymbol{u}_{i_k}, i_k \in [r]\}$ is the orthonormal basis of a subset of $\mathbb{R}^{n_k}$ of size r for each $k \in [d]$. The rank of the tensor $\mathcal{A}$ is denoted r.

The CP decomposition can also be expressed in terms of the contraction product as follows:

$$\mathcal{A} = ((\mathcal{C} \times_1 \mathbf{U}_2) \times_2 \mathbf{U}_1) \ldots \times_d \mathbf{U}_d). \tag{1.37}$$

Here, $\mathcal{C} \in \mathbb{R}^{r \times \cdots \times r}$ is a d-order tensor with nonzero elements solely on its super-diagonal, i.e., $\mathcal{C}(i_1; \ldots; i_d) = \lambda_r$ if $i_1 = \ldots = i_d$, and $\mathcal{C}(i_1; \ldots; i_d) = 0$, otherwise. Additionally, the matrices $\mathbf{U}_k \in \mathbb{R}^{n_k \times r}$ are defined such that $\mathbf{U}_k[:, i_k] = \boldsymbol{u}_{i_k}$ for $i_k \in [r]$ and $k \in [d]$. The graphical notation for the CP format is given in Figure 1.6.

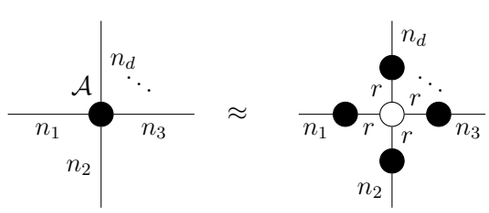


Figure 1.6: Graphical representation of CP decomposition: the white circle represents the super-diagonal tensor $\mathcal{C} \in \mathbb{R}^{r \times \cdots \times r}$ with super-diagonal elements λ_r.

Clearly the storage cost is reduced from $\mathcal{O}(n^d)$ to $\mathcal{O}(ndr)$ with $n = \max_{1 \leq k \leq n_k} \{n_k\}$, r is the tensor rank and d is its order. For matrices, the best rank-r approximation is often determined by analyzing the decay of singular values and selecting the leading ones according to a threshold, by employing tSVD. However, this approach does not directly extend to high-order tensors. Unlike matrices, high-order tensors exhibit a more complex structure, and their best rank-r approximation is not always straightforward to find. For example, in the sense of the CP decomposition, the best rank-r approximation is ill-posed. This was exemplified in [68], where it was illustrated that a rank-3 tensor, i.e expressed as a linear combination of 3 rank-1 tensors can be effectively approximated by a rank-2 tensor, i.e expressed as a linear combination of 2 rank-1 tensors. Further details can be found in [68]. This difficulty can also be understood from a geometric viewpoint, where the set of tensors with CP decomposition with tensor rank less than or equal to r is not a closed set, in contrast to what is known for matrices, as indicated in Proposition 1.2.1 [68].

Definition 18. (Tucker decomposition) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-order tensor. The Tucker decomposition of $\mathcal{A}$ can be expressed as:

$$\mathcal{A} = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \mathcal{C}(i_1; \cdots; i_d) \boldsymbol{u}_{i_1} \otimes \cdots \boldsymbol{u}_{i_d}, \tag{1.38}$$

where $\mathcal{C} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is a d-order tensor, also-called the core tensor, r_k are the ranks of the mode-k matricization of $\mathcal{A}$ denoted by $\mathbf{A}^{(k)}$ and $B_{\mathbb{R}^{r_k}} = \{\boldsymbol{u}_{i_k}, i_k \in [r_k]\}$ is the orthonormal basis of a subspace of $\mathbb{R}^{n_k}$ of dimension r_k, for $k \in [d]$.

The Tucker decomposition can also be expressed in terms of the contraction product as follows:

$$\mathcal{A} = (\mathcal{C} \times_1 \mathbf{U}_1) \times_2 \mathbf{U}_2) \cdots \times_d \mathbf{U}_d), \tag{1.39}$$

where $\mathbf{U}_k \in \mathbb{R}^{n_k \times r_k}$ are orthogonal matrices, referred to as the factor matrices, such that $\mathbf{U}_k[:, i_k] = \boldsymbol{u}_{i_k}$, $i_k \in [r_k], k \in [d]$. The graphical notation of the *Tucker decomposition* is given in Figure 1.7.

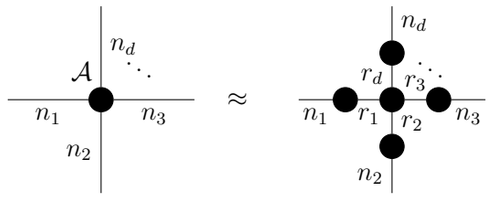


Figure 1.7: Graphical representation of *Tucker decomposition*.

We observe that the storage cost of the Tucker decomposition can be smaller than that of the original tensor. However, it still grows exponentially with the order of the tensor d, resulting in a storage complexity of $\mathcal{O}(r^d + dnr)$, with $r = \max_{1 \le k \le d} \{r_k\}$ and $n = \max_{1 \le k \le d} \{n_k\}$. The exponential growth primarily arises from the storage of the core tensor in the Tucker decomposition, which makes this decomposition more efficient for tensors with small orders d. Therefore, alternative decompositions have been proposed to overcome the exponential scaling. One promising technique is the Tensor Train (TT) decomposition, which will be the subject of Chapter 3 and Chapter 4. The TT decomposition has emerged as an ideal choice for several applications, particularly in quantum chemistry, where the Density Matrix Renormalization Group (DMRG) method, as will be introduced in Section 1.3, can involve the manipulation of tensors with large orders d to compute the ground-state, lowest energy, of a quantum many-body system.

Tensor train decomposition

The TT decomposition described in detail by Osledets [91, 92], is a powerful representation that addresses the exponential scaling associated with high-order tensors. Originating as a mathematical formulation of the Matrix Product States (MPS) in the physics community, the TT decomposition gained widespread recognition following the introduction of the DMRG algorithm pioneered by White [131]. The TT decomposition can be viewed as a specific case of tensor networks and offers an elegant solution to efficiently handle high-dimensional tensors.

At its core, the TT decomposition expresses a d-order tensor as a contraction product of 2-order and 3-order tensors with reduced dimensions. This decomposition will be the focal point of Chapters 3 and 4, due to its ability to tackle high-dimensional problems effectively.

A TT decomposition of a tensor can be defined as follows:

Definition 19. (Tensor train decomposition [92]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d- order tensor. The set $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ is said to be a tensor train decomposition of $\mathcal{A}$, if $\mathcal{A}$ decomposes as:

$$\mathcal{A} = \mathcal{A}_1 \times_3 \mathcal{A}_2 \times_3 \cdots \times_3 \mathcal{A}_{d-1} \times_3 \mathcal{A}_d, \tag{1.40}$$

where $\mathcal{A}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ are the so-called TT-cores and r_k are the TT-ranks with $k \in \{2, \ldots, d-1\}$, $r_0 = r_d = 1$ where $\mathcal{A}_1 \in \mathbb{R}^{1 \times n_1 \times r_1}$ and $\mathcal{A}_d \in \mathbb{R}^{r_{d-1} \times n_d \times 1}$.

Alternatively, let $\mathbf{A}_k[i_k] := \mathcal{A}_k[:, i_k, :] \in \mathbb{R}^{r_{k-1} \times r_k}$, for fixed $i_k \in [n_k]$, for each $k \in [d]$. The $(i_1, \cdots, i_d)$-th element of $\mathcal{A}$ is given as:

$$\mathcal{A}(i_1; \ldots; i_d) = \mathbf{A}_1[i_1] \cdots \mathbf{A}_d[i_d]. \tag{1.41}$$

Remark 1.2.1. (Minimal TT-rank [47]) The TT decomposition $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ of a d-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is said to be of minimal ranks , if all the TT-cores have full left (resp. right) ranks, i.e for each $k \in [d]$, the mode-(1:2) matricization of $\mathcal{A}_k$ denoted by $\mathbf{A}_k^{<2>} \in \mathbb{R}^{r_{k-1} n_k \times r_k}$ is of rank r_k. (resp. the mode-(1) matricization of $\mathcal{A}_k$ denoted by $\mathbf{A}_k^{(1)} \in \mathbb{R}^{r_{k-1} \times n_k r_k}$ is of rank r_{k-1}).

The graphical representation of the TT decomposition is given as follows: each node corresponds to a TT-core as denoted by $\mathbf{A}_k[i_k] \in \mathbb{R}^{r_{k-1}\times r_k}, i_k \in [n_k], k \in [d]$, edges connecting nodes are labeled by the TT-ranks as denoted by $r_k, k \in [d]$, and edges without connections are labeled by the dimension of the tensor along each mode as denoted by $n_k, k \in [d]$.

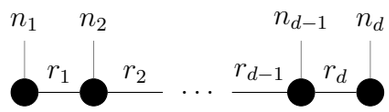


Figure 1.8: Graphical representation of a TT decomposition.

We next recall one of the main theorems of the TT decomposition which gives a way to construct the TT decomposition of a given tensor of order d.

Theorem 1. ([91] Theorem 2.1) Let $\mathcal{A} \in \mathbb{R}^{n_1\times\ldots\times n_d}$ be a d-order tensor and let $\mathbf{A}^{<k>} \in \mathbb{R}^{(n_1\ldots n_k)\times(n_{k+1}\ldots n_d)}$ be the mode-(1:k) matricization of $\mathcal{A}$ for $k \in [d]$. If for each k:

$$\operatorname{rank}(\mathbf{A}^{<k>}) = r_k, \tag{1.42}$$

then there exists a TT decomposition of $\mathcal{A}$ of the form (1.40) with TT-ranks less or equal to $r_k, k \in [d]$.

In order to obtain a TT decomposition of a given d-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$, the TT-SVD method is employed. The latter can be viewed as a sequential series of SVDs on auxiliary matrices: Given the input tensor $\mathcal{A}$, the tensor is first reshaped along the first mode n_1, resulting in an auxiliary matrix which is the mode-(1) matricization of $\mathcal{A}$ denoted by $\mathbf{B}_1 = \mathbf{A}^{(1)} \in \mathbb{R}^{n_1\times(\prod_{i=2}^{d} n_i)}$ with rank r_1, then an SVD is performed yielding the decomposition $\mathbf{B}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$. The matrix $\mathbf{U}$ is then reshaped into the first TT-core $\mathcal{A}_1 \in \mathbb{R}^{1\times n_1\times r_1}$, where r_1 is the TT-rank of the first TT-core and suppose that $r_0 = 1$. The product of matrices $\mathbf{\Sigma}\mathbf{V}^* \in \mathbb{R}^{r_1\times(\prod_{i=2}^{d} n_i)}$ is then reshaped into a new matrix $\mathbf{B}_2 \in \mathbb{R}^{r_1 n_2\times(\prod_{i=3}^{d} n_i)}$ assumed to be with rank r_2. Another SVD is performed on that matrix to extract the second TT-core, denoted by $\mathcal{A}_2 \in \mathbb{R}^{r_1\times n_2\times r_2}$. This iterative process continues, resulting in d TT-cores. **Algorithm** 1 provides a detailed description of the TT-SVD method, for more details see [92].

In **Algorithm** 1, the Reshape function, a default function implemented in Julia or MATLAB. It constructs an array with same entries but with different dimensions. The syntax for the Reshape function is as follows:

$$\operatorname{Reshape}(A, dims), \tag{1.43}$$

where A is the data-structure (it can be a vector, matrix or a tensor) that we want to reshape and *dims* is a tuple specifying the desired dimensions of the reshaped array. For instance, in line 7 of **Algorithm** 1, for $k \in [d]$, the operation $\operatorname{Reshape}(\mathbf{U}, (r_{k-1}, n_k, r_k))$ represents the tensor folding of $\mathbf{U}$ into a tensor of size $r_{k-1}\times n_k\times r_k$. Similar operation is performed in line 9. In line 8, the operation $\mathbf{B}_{k+1} = \operatorname{Reshape}(\mathbf{\Sigma}\mathbf{V}^*, (r_k n_{k+1}, \frac{n}{\prod_{i=1}^{k+1} n_i}))$ corresponds to representing the matrix $\mathbf{\Sigma}\mathbf{V}^* \in \mathbb{R}^{r_k\times(\prod_{i=k+1}^{d} n_i)}$ as a matrix $\mathbf{B}_{k+1} \in \mathbb{R}^{\left(r_k n_{k+1}\right)\times\left(\frac{n}{\prod_{i=1}^{k+1} n_i}\right)}$.

Algorithm 1 TT-SVD algorithm proposed by Osledets (see [92])

procedure TT-SVD($\mathcal{A}$) ▷ Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.
  $r_0 = 1, r_d = 1$, $n = \prod_{i=1}^{d} n_i$.
  Mode-1 matricization of $\mathcal{A}$ into the matrix $\mathbf{A}^{(1)} \in \mathbb{R}^{n_1 \times (\prod_{i=2}^{d} n_i)}$.
  $\mathbf{B}_1 = \mathbf{A}^{(1)}$ and $r_1 = \text{rank}(\mathbf{B}_1) \leq \min\left\{n_1, \prod_{i=2}^{d} n_i\right\}$.
  for $k = 1$ to $d-1$ **do**
    $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \text{SVD}(\mathbf{B}_k)$ ▷ $\mathbf{U} \in \mathbb{R}^{r_{k-1}n_k \times r_k}$, $\mathbf{\Sigma} \in \mathbb{R}^{r_k \times r_k}$, $\mathbf{V} \in \mathbb{R}^{\left(\prod_{i=k+1}^{d} n_i\right) \times r_k}$.
    $\mathcal{A}_k = \text{Reshape}(\mathbf{U}, (r_{k-1}, n_k, r_k))$.
    $\mathbf{B}_{k+1} = \text{Reshape}(\mathbf{\Sigma}\mathbf{V}^*, (r_k n_{k+1}, \frac{n}{\prod_{i=1}^{k+1} n_i}))$.
  end for
  $\mathcal{A}_d = \text{Reshape}(\mathbf{B}_d, (r_{d-1}, n_d, r_d))$.
  return TT decomposition with TT-cores $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$.
end procedure

Remark 1.2.2. It is worth mentioning that the TT decomposition is a special case of the Hierarchical Tucker decomposition, which is a hierarchical tree-like decomposition, we refer the reader to [37] for more details on this decomposition.

Remark 1.2.3. (Unicity of the TT decomposition [47]) The TT decomposition is not unique. For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in TT format, with TT-cores $(\mathcal{A}_1, \ldots, \mathcal{A}_d)$, there exist square invertible matrices $\mathbf{U}_k \in \mathbb{R}^{r_k \times r_k}$ for $k \in [d]$ such that

$$\mathcal{A}(i_1; \cdots; i_d) = \mathbf{A}_1[i_1] \cdots \mathbf{A}_k[i_k] \cdots \mathbf{A}_d[i_d] = \mathbf{B}_1[i_1] \ldots \mathbf{B}_k[i_k] \cdots \mathbf{B}_d[i_d], \tag{1.44}$$

where

$$\begin{aligned} \mathbf{B}_1[i_1] = \mathbf{A}_1[i_1]\mathbf{U}_1, \mathbf{B}_d[i_d] = \mathbf{U}_{d-1}^{-1}\mathbf{A}_d[i_d], \\ \mathbf{B}_k[i_k] = \mathbf{U}_{k-1}^{-1}\mathbf{A}_k[i_k]\mathbf{U}_k. \end{aligned} \tag{1.45}$$

This results into another equivalent TT representation with new TT-cores.

In what follows, we present a definition of the concepts of left and right orthogonalization in the context of the TT decomposition.

Definition 20. ([92]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-order tensor and let $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ be its TT decomposition according to Definition 19. The TT-cores are said to be *left* orthogonal if the following is satisfied:

$$(\mathbf{A}_k^{<2>})^* \mathbf{A}_k^{<2>} = \sum_{i_k=1}^{n_k} (\mathbf{A}_k[i_k])^* \mathbf{A}_k[i_k] = \mathbf{I}_{r_k}, \tag{1.46}$$

where $\mathbf{A}_k[i_k] := \mathcal{A}_k[:, i_k, :]$, and $\mathbf{A}_k^{<2>} \in \mathbb{R}^{r_{k-1}n_k \times r_k}$ is the mode-(1:2) matricization of the TT-core $\mathcal{A}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, for each $k \in [d]$. The TT-cores are said to be *right* orthogonal if the following is satisfied:

$$\mathbf{A}_k^{(1)} (\mathbf{A}_k^{(1)})^* = \sum_{i_k=1}^{n_k} \mathbf{A}_k[i_k](\mathbf{A}_k[i_k])^* = \mathbf{I}_{r_{k-1}}, \tag{1.47}$$

where $\mathbf{A}_k^{(1)} \in \mathbb{R}^{r_{k-1} \times n_k r_k}$ is the mode-(1) matricization of the TT-core $\mathcal{A}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, for each $k \in [d]$.

Algorithm 1 yields a left-orthogonal TT decomposition. To obtain a right-orthogonal TT decomposition, we can simply apply the same iterative procedure backward. One of the benefits of achieving this orthogonalization is the simplified calculation of the tensor's norm. For a left (resp. right) orthogonal TT decomposition, the tensor's norm can be obtained simply from the last (resp. first) TT-core of the decomposition. To illustrate,

let us consider a left-orthogonal TT decomposition $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ of the d-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. The Frobenius norm of $\mathcal{A}$ can be given as follows:

$$\begin{aligned}
\|\mathcal{A}\|_F^2 &= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} (\mathcal{A}(i_1; \cdots; i_d))^2 \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} (\mathbf{A}_1[i_1]\mathbf{A}_2[i_2] \cdots \mathbf{A}_d[i_d])^2 \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} (\mathbf{A}_d[i_d]^* \mathbf{A}_{d-1}[i_{d-1}]^* \cdots \mathbf{A}_1[i_1]^*)(\mathbf{A}_1[i_1]\mathbf{A}_2[i_2] \cdots \mathbf{A}_d[i_d]) \\
&= \sum_{i_d=1}^{n_d} \mathbf{A}_d[i_d]^* \mathbf{A}_d[i_d] = \|\mathcal{A}_d\|_F^2 .
\end{aligned} \tag{1.48}$$

Remark 1.2.4. ([47] Theorem 1) As highlighted in [47], a TT-SVD of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ gives a TT-decomposition of minimal ranks, the orthogonalization helps with making the TT decomposition unique up to orthogonal transformations, i.e let $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ be a TT decomposition of the tensor $\mathcal{A}$, then there exist orthogonal matrices $\mathbf{Q}_k \in \mathbb{R}^{r_k \times r_k}$ such that

$$\mathcal{A}(i_1; \cdots; i_d) = \mathbf{A}_1[i_1] \cdots \mathbf{A}_k[i_k] \cdots \mathbf{A}_d[i_d] = \mathbf{B}_1[i_1] \cdots \mathbf{B}_k[i_k] \cdots \mathbf{B}_d[i_d], \tag{1.49}$$

where

$$\mathbf{B}_1[i_1] = \mathbf{A}_1[i_1]\mathbf{Q}_1,\ \mathbf{B}_k[i_k] = \mathbf{Q}_{k-1}^* \mathbf{A}_k[i_k]\mathbf{Q}_k,\ \text{and } \mathbf{B}_d[i_d] = \mathbf{Q}_{d-1}^* \mathbf{A}_d[i_d]. \tag{1.50}$$

In the literature [26], this is commonly referred to as the reduction of the *gauge freedom* in the TT representation.

In what follows, we review some of the important arithmetic operations that can be performed among tensors in the TT-format.

Proposition 1.2.2. *Let $\mathcal{A}$ and $\mathcal{B} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ be two d-order tensors, with TT decompositions given by:*

$$\mathcal{A}(i_1; \ldots; i_d) = \mathbf{A}_1[i_1] \ldots \mathbf{A}_d[i_d], \tag{1.51}$$

$$\mathcal{B}(i_1; \ldots; i_d) = \mathbf{B}_1[i_1] \ldots \mathbf{B}_d[i_d]. \tag{1.52}$$

with $\mathcal{A}_k[i_k] \in \mathbb{R}^{r_{k-1}^A \times r_k^A}, \mathcal{B}_k[i_k] \in \mathbb{R}^{r_{k-1}^B \times r_k^B}, k \in [d]$ and $r_0^A = r_d^A = r_0^B = r_d^B = 1$. The following statements hold:

1. ***Addition of two tensor trains :*** *the tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ defined as the addition of two tensors, $\mathcal{A}$ and $\mathcal{B}$ in TT-format with TT-ranks $(r_1^A, \ldots, r_d^A)$, $(r_1^B, \ldots, r_d^B)$, has a TT decomposition with TT-ranks $(r_1^A + r_1^B, \ldots, r_d^A + r_d^B)$.*
2. ***Multiplication by a scalar:*** *the multiplication of a TT by a scalar keeps the TT-ranks invariant.*

Proof. 1. Let $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be the addition of two tensors $\mathcal{A}$ and $\mathcal{B}$. Then we can write $\mathcal{C}$ entry-wise as:

$$\begin{aligned}
&\mathcal{C}(i_1; \cdots; i_d) = \mathbf{A}_1[i_1] \ldots \mathbf{A}_d[i_d] + \mathbf{B}_1[i_1] \ldots \mathbf{B}_d[i_d] \\
&= \underbrace{\begin{bmatrix} \mathbf{A}_1[i_1] & \mathbf{B}_1[i_1] \end{bmatrix}}_{\in \mathbb{R}^{\left(r_0^A + r_0^B\right) \times \left(r_1^A + r_1^B\right)}} \underbrace{\begin{bmatrix} \mathbf{A}_2[i_2] & \mathbf{0}_{r_1^A \times r_2^B} \\ \mathbf{0}_{r_1^B \times r_2^A} & \mathbf{B}_2[i_2] \end{bmatrix}}_{\in \mathbb{R}^{\left(r_1^A + r_1^B\right) \times \left(r_2^A + r_2^B\right)}} \cdots \underbrace{\begin{bmatrix} \mathbf{A}_d[i_d] \\ \mathbf{B}_d[i_d] \end{bmatrix}}_{\in \mathbb{R}^{\left(r_{d-1}^A + r_{d-1}^B\right) \times \left(r_d^A + r_d^B\right)}}.
\end{aligned} \tag{1.53}$$

This results into a TT with TT-ranks $(r_1^A + r_1^B, \ldots, r_d^A + r_d^B)$ as required.

2. Let $\alpha \in \mathbb{R}$, the product $\mathcal{B} = \alpha\mathcal{A}$ is equivalent to scaling only one of the tensor cores of the TT.

$$\mathcal{A}_k = \begin{cases} \alpha\tilde{\mathcal{A}}_k, \text{ if } k = 1 \\ \tilde{\mathcal{A}}_k, \text{ if } k > 1. \end{cases} \tag{1.54}$$

This results into a TT with the same TT-ranks.

□

Computational aspects and memory requirements

Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ be a d-order tensor and let $(\mathcal{A}_1, \ldots, \mathcal{A}_d)$ be its TT decomposition, with $\mathcal{A}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, $k \in [d]$, $r_0 = r_d = 1$. It can readily be checked that the storage cost of the TT decomposition is only $\mathcal{O}(dnr^2)$ where $r = \max_{1 \leq k \leq d} \{r_k\}$ and $n = \max_{1 \leq k \leq d} \{n_k\}$. This implies that memory requirements increase linearly with the tensor order d [91] and dimension n and quadratically with the TT-ranks, which is contrasted with the Tucker decomposition described in Definition 18. Therefore, the TT decomposition becomes advantageous for making computational costs affordable, especially when the TT-ranks are bounded or small.

One significant limitation of the TT-format does arise during arithmetic operations, such as the addition operation as described in Proposition 1.2.2, that tends to increase the TT-ranks. To mitigate this issue, compression methods become crucial. As mentioned in [91], it is possible to achieve an approximate TT decomposition with reduced TT-ranks in comparison to the original TT decomposition. This can be achieved by performing truncated SVD, at a given accuracy δ during each iteration, instead of the full SVD as outlined in **Algorithm** 1. In this case, an error bound can be estimated. Given **Algorithm** 1, suppose that instead of SVD, tSVD is performed at each iteration such that the singular values of the auxiliary matrices are truncated at accuracy δ, the relative error between the original tensor $\mathcal{A}$ and its approximation denoted by $\mathcal{A}_\epsilon$ is given by (see [91]):

$$\|\mathcal{A} - \mathcal{A}_\epsilon\|_F \leq \sqrt{d-1}\delta. \tag{1.55}$$

Therefore, to obtain the relative accuracy ϵ for the approximated tensor $\mathcal{A}_\epsilon$, one should choose $\delta = \frac{\epsilon \|\mathcal{A}\|_F}{\sqrt{d-1}}$.

Similarly to matrices, a geometric viewpoint on the set of TT with TT-ranks that are bounded element-wise by $\boldsymbol{r} = (r_1, \ldots, r_d)$, can also be found in the literature [47], where the TT-SVD with tSVD plays the same role as SVD with truncation for matrices

Proposition 1.2.3 ([47])**.** *The set of TT of TT-ranks at most* $\boldsymbol{r} = (r_1, \ldots, r_d)$ *defined as:*

$$M_{\mathrm{TT} \leq \boldsymbol{r}} = \left\{ \mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d} \text{ is a tensor of TT ranks} \leq \boldsymbol{r} \right\}, \tag{1.56}$$

is a closed set.

The proof can be established by expressing the set as follows:

$$M_{\mathrm{TT} \leq \boldsymbol{r}} = \bigcap_{k=1}^{d-1} \left\{ \mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}, \mathrm{rank}(\mathbf{A}^{(1:k)}) \leq r_k \right\}. \tag{1.57}$$

Since each set in the intersection is a closed set (as per Definition 1.2.1), it follows that the set $M_{\mathrm{TT} \leq \boldsymbol{r}}$ is also a closed set. This property is advantageous when compared to the CP decomposition, as it ensures that every d-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ admits a best approximation by a TT in $M_{\mathrm{TT} \leq \boldsymbol{r}}$.

Remark 1.2.5. [126] Similarly to the matrix case, the set of TT of TT-ranks at most $\boldsymbol{r} = (r_1, \ldots, r_d)$ as defined in Proposition 1.2.3, is an algebraic variety but not a smooth manifold. Instead, the study of geometrical optimization methods is carried out on the set of TT with fixed TT-ranks which is known to be a smooth manifold (see [126]).

In what follows, we will introduce another used algorithm to reduce the TT-ranks of a tensor already given in TT-format. Such process is known as TT-rounding [92] and is given in **Algorithm** 2. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-order tensor with a TT decomposition $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ according to Definition 19. **Algorithm** 2 can be broken down into two main steps: right to left orthogonalization and then left to right truncation. The right to left orthogonalization step involves a sequence of LQ-decompositions[2] of the mode-1 matricization of each TT-core denoted by $\mathbf{A}_k^{(1)} \in \mathbb{R}^{r_{k-1} \times n_k r_k}$ moving backward starting from $k = d$ until $k = 2$. In the LQ-decomposition, the matrix is factored into the product of a lower-triangular matrix $\mathbf{L}$ and an orthonormal matrix $\mathbf{Q}$ with orthonormal rows. The triangular factor $\mathbf{L}$ is then applied to the preceding TT-core $\mathcal{A}_{k-1}$. The left

[2]The LQ decomposition of a matrix is the QR decomposition of its transpose.

to right truncation step employs a sequence of truncated SVDs applied to the mode-(1:2) matricization of each TT-core denoted by $\mathbf{A}_k^{<2>} \in \mathbb{R}^{r_{k-1}n_k \times r_k}$, $k \in [d-1]$. The next TT-core $\mathcal{A}_{k+1}$ is then updated by multiplying it with the truncated singular values and right singular vectors. The direction of these two steps can be reversed.

Algorithm 2 TT-rounding algorithm [92]

1: **procedure** TT-ROUNDING($\mathcal{A}, \epsilon$) ▷ Input: TT decomposition $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ with TT-ranks $r_k, k \in [d]$, accuracy $\epsilon \in \mathbb{R}_+$.
2: {right to left orthogonalization}.
3: **for** $k = d$ to 2 **do**
4: $\mathbf{L}, \mathbf{Q} = \text{LQ}(\mathbf{A}_k^{(1)})$, ▷ $\mathbf{L} \in \mathbb{R}^{r_{k-1} \times r_{k-1}}, \mathbf{Q} \in \mathbb{R}^{r_{k-1} \times n_k r_k}, \mathbf{Q}^*\mathbf{Q} = \mathbf{I}$.
5: $\mathcal{A}_k = \text{Reshape}(\mathbf{Q}, (r_{k-1}, n_k, r_k))$.
6: $\mathcal{A}_{k-1} = \mathcal{A}_{k-1} \times_3 \mathbf{L}$.
7: **end for**
8: Compute $\|\mathcal{A}\|_F = \|\mathcal{A}_1\|_F$ and $\delta = \frac{\epsilon \|\mathcal{A}_1\|_F}{\sqrt{d-1}}$
9: {left to right truncation}
10: **for** $k = 1$ to $d-1$ **do**
11: $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \text{tSVD}(\mathbf{A}_k^{<2>}, \delta)$ ▷ $\mathbf{U} \in \mathbb{R}^{r_{k-1}n_k \times r_k}$, $\mathbf{\Sigma} \in \mathbb{R}^{r_k \times r_k}$, $\mathbf{V} \in \mathbb{R}^{r_k \times r_k}$.
12: $\mathcal{A}_k = \text{Reshape}(\mathbf{U}, (r_{k-1}, n_k, r_k))$.
13: $\mathcal{A}_{k+1} = (\mathbf{\Sigma}\mathbf{V}^*) \times_2 \mathcal{A}_{k+1}$.
14: **end for**
15: **return** TT-cores $(\mathcal{A}_1, \cdots, \mathcal{A}_d)$ with reduced TT-ranks.
16: **end procedure**

Here, the term tSVD represents a function that takes a matrix and a specified threshold δ as input. It uses this threshold to truncate singular values and provides the resulting truncated decomposition, which serves as an approximation to the input matrix in line 10 of **Algorithm** 2.

In practice, the drawback of using TT-SVD is that we need to store the full tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in advance, which scales exponentially with its order d. Therefore, if the tensor is already in TT-format, it is possible to reduce the TT-ranks using the TT-rounding algorithm yielding an approximate TT decomposition with reduced ranks. The latter is beneficial in reducing the computational cost when performing basic operations involving tensors in TT-format. The computational complexity of the TT-rounding algorithm is $\mathcal{O}(dnr^3)$ [92], with $r = \max_{1 \le k \le d} \{r_k\}$ and $n = \max_{1 \le k \le d} \{n_k\}$. This complexity is obtained from the cost of applying the LQ and SVD operations on the matricization of each TT-core $\mathcal{A}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, $k \in [n]$ which requires a computational complexity of $\mathcal{O}(nr^3)$.

1.2.4 High-dimensional eigenvalue problems

In this thesis, we address the problem of finding the smallest eigenvalue of a matrix, which is a fundamental task intensively used in diverse scientific domains. The goal is to identify the minimal eigenpair consisting of the smallest eigenvalue and its corresponding eigenfunction. This particular eigenpair holds significant importance in various applications, including quantum chemistry (for more details, refer to Section 1.3).

Mathematically, we seek to reformulate the problem as follows: given a symmetric square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the eigenvalue problem can be given as follows:

$$\mathbf{A}\boldsymbol{x} = \lambda \boldsymbol{x}, \boldsymbol{x} \neq \mathbf{0}, \lambda \in \mathbb{R}. \tag{1.58}$$

Equation (1.58) can equivalently correspond to finding the stationary points of the functional $R_{\mathbf{A}} : \mathbb{R}^n \backslash \mathbf{0} \to \mathbb{R}$ such that

$$R_{\mathbf{A}}(\boldsymbol{x}) = \frac{1}{2} \frac{\langle \mathbf{A}\boldsymbol{x}, \boldsymbol{x} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}, \tag{1.59}$$

where $R_{\mathbf{A}}(\boldsymbol{x})$ is the Rayleigh quotient. Finding the smallest eigenvalue from the Equation

(1.58) corresponds to minimizing the Rayleigh quotient over $\boldsymbol{x} \in \mathbb{R}^n$.

$$\lambda_{\min} = \min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{x}\neq 0} \underbrace{\frac{1}{2}\frac{\langle \mathbf{A}\boldsymbol{x}, \boldsymbol{x}\rangle}{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}}_{R_{\mathbf{A}}(\boldsymbol{x})}. \tag{1.60}$$

Over the years, substantial efforts have been dedicated to solving this problem, employing diverse approaches: classical direct methods such as calculating the roots of the characteristic polynomial or iterative methods such as the inverse power method, Krylov subspace methods. [101]. However, despite their effectiveness, these methods are inherently limited by the dimensionality of the problem.

To render the problem tractable, instead of minimizing over the entire space, the minimization is performed over the manifold of tensors represented in the TT-format with bounded TT-ranks as defined in Definition 1.2.3. This idea originates from DMRG method in the context of quantum chemistry, which uses Matrix Product States (MPS) representations, (see Section 1.3). This means that one can interpret the problem as the following modified minimization problem: assuming that $\boldsymbol{x}$ can be viewed as a tensor denoted by $\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_d}$, with $n = \prod_{i=1}^d n_i$ and with a TT decomposition with TT-rank at most equal to $\boldsymbol{r} = (r_1, \ldots, r_d)$, then:

$$\lambda_{\min} = \min_{\mathcal{X}\in M_{\mathrm{TT}\leq \boldsymbol{r}}, \mathcal{X}\neq 0} \frac{1}{2}\frac{\langle \mathbf{A}\,\mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{X})\rangle}{\langle \mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{X})\rangle}, \tag{1.61}$$

with $\mathrm{vec}(\mathcal{X}) \in \mathbb{R}^{n_1\cdots n_d}$ being the vectorization of the tensor $\mathcal{X}$. While solving (1.61) has existed in the physics community since the work of White [17], its formulation in the mathematical literature in the context of tensor trains is done by researchers from the numerical linear algebra, in particular Holtz, Rohwedder, Schneider and Uschmajew [47, 48]. They proposed the terminologies of ALS (Alternating Least Squares) and MALS (Modified Alternating Least Squares) for solving the eigenvalue problem using tensors in the TT-format. This modified minimization problem is the one used in the following section which focuses on the eigenvalues of the Hamiltonian operator in quantum chemistry.

1.3 High-dimensional problems arising in quantum chemistry

1.3.1 Introduction

As mentioned previously, our primary focus in this thesis is tackling high-dimensional problems that arise in quantum chemistry. The upcoming section provides the motivation behind introducing the concept of matrices and tensors, along with their low-rank approximation techniques, particularly in the context of high-dimensional problems. As part of the Extreme-Scale Mathematically-based Computational Chemistry (EMC2) project, we engaged in insightful discussions with chemists to explore how the ideas introduced in the previous section could be beneficial to speed up some applications in molecular simulations.

Our work focuses on two key applications. The first application revolves around approximating the long-range two-electron integrals 4-th order tensor, discussed in Chapter 2. The second application, extensively discussed in Chapter 3 and Chapter 4, centers around the efficient TT representation of the Hamiltonian operator and how it is employed in DMRG method to evaluate the ground-state energy of a given molecular system. Both applications involve the use of tensors, low-rank approximation techniques and symmetries.

1.3.2 Many-body electronic Schrödinger equation

The Schrödinger equation is a fundamental equation in quantum mechanics that describes the quantum state of a physical system. Consider a molecule with N electrons at position $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{3N}$, $\boldsymbol{x}_i = (x_i, y_i, z_i)$ and with discrete spin-variable[3]

[3]The spin-variable is one of the four non-relativistic coordinates of electrons (the other three are the spatial position). The latter can be spin up, $\frac{1}{2}$, or spin down, $\frac{-1}{2}$.

$s_i \in \left\{\pm\frac{1}{2}\right\}, i \in [N]$ and M nuclei at position $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_M) \in \mathbb{R}^{3M}$. Under the Born–Oppenheimer [1] approximation, since the nuclei are much heavier than the electrons, these behave like classical particles whereas electrons are treated quantumly. The electronic properties of the molecule can be inferred from the spectrum of the electronic Hamiltonian operator $\hat{H}$ given by:

$$\hat{H} = \sum_{i=1}^{N} \left(-\frac{1}{2}\Delta_i - \sum_{k=1, k\neq i}^{M} \frac{Z_k}{\|\boldsymbol{x}_i - \boldsymbol{r}_k\|} + \frac{1}{2} \sum_{j=1, j\neq i}^{N} \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} \right), \tag{1.62}$$

where Δ_i denotes the Laplacian operator with respect to the i-th spatial coordinate $\boldsymbol{x}_i$ and $Z_k > 0$ denotes the nuclear charges. The first term in (1.62) is the kinetic energy, the second term corresponds to the Coulomb interaction with the nuclei and the last term is the electron-electron Coulomb repulsion.

The ground-state denoted by Ψ is then the eigenfunction associated to the lowest eigenvalue of $\hat{H}$ denoted by E_0 and referred to as *the ground-state energy*. The eigenvalue equation is given by [39]:

$$\hat{H}\Psi = E_0\Psi. \tag{1.63}$$

Remark 1.3.1. In the literature, the eigenfunctions of the Hamiltonian operator are commonly referred to as the wavefunctions.

The function Ψ is a multivariable function and is defined by the following mapping [105]:

$$\Psi : \left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)^N \to \mathbb{C}, \quad \Psi\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_N, s_N\right), \tag{1.64}$$

where Ψ belongs to the Hilbert space $L^2(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\})$, which is the function space of square integrable functions, with L^2-inner product:

$$\langle \Psi_1, \Psi_2 \rangle_{L^2} = \sum_{s_k = \pm\frac{1}{2}} \int_{\mathbb{R}^{3N}} \overline{\Psi_1\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_N, s_N\right)} \Psi_2\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_N, s_N\right) \mathrm{d}\boldsymbol{x}_1 \ldots \mathrm{d}\boldsymbol{x}_N, \tag{1.65}$$

with $\Psi_1, \Psi_2 \in L^2(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\})$ and $\overline{\Psi_1\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_N, s_N\right)}$ denoting the conjugate of the complex value $\Psi_1\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_N, s_N\right)$. The wavefunction must obey the *Pauli exclusion principle* [39] such that

$$\Psi\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_i, s_i, \ldots, \boldsymbol{x}_j, s_j, \ldots, \boldsymbol{x}_N, s_N\right) = -\Psi\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_j, s_j, \ldots, \boldsymbol{x}_i, s_i, \ldots, \boldsymbol{x}_N, s_N\right). \tag{1.66}$$

It follows that Ψ belongs to the antisymmetric tensor subspace denoted by $\bigwedge_{i=1}^{N} L^2(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\})$ which can be defined as follows:

$$\begin{aligned} &\bigwedge_{i=1}^{N} L^2(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}) \\ &= \Big\{\Psi \in L^2(\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)^N) : \Psi\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_i, s_i, \ldots, \boldsymbol{x}_j, s_j, \ldots, \boldsymbol{x}_N, s_N\right) = \\ &- \Psi\left(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_j, s_j, \ldots, \boldsymbol{x}_i, s_i, \ldots, \boldsymbol{x}_N, s_N\right)\Big\}. \end{aligned} \tag{1.67}$$

The Hamiltonian operator $\hat{H}$ is a self-adjoint operator acting on the Hilbert space $\bigwedge_{i=1}^{N} L^2(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\})$. Without going into details we refer readers to [98] for a comprehensive analysis and insights into the properties of this operator.

The Equation (1.63) can be seen as a linear eigenvalue equation, which becomes numerically intractable with standard discretization schemes as soon as we deal with more than a few electrons. This is again the so-called *curse of dimensionality*. By Rayleigh-Ritz principle, the search of the lowest eigenvalue can be recasted into the following minimization problem:

$$E_0 = \min\left\{\left\langle \Psi, \hat{H}\Psi \right\rangle, \|\Psi\|_{L^2} = 1, \Psi \in \boldsymbol{\mathcal{V}}_N\right\}, \tag{1.68}$$

where the solution Ψ must be restricted to the so-called variational space $\boldsymbol{\mathcal{V}}_N$, on which Ψ is minimized, defined by (see [98, 105] for more details):

$$\boldsymbol{\mathcal{V}}_N = H^1\left(\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)^N\right) \cap \bigwedge_{i=1}^{N} L^2\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right), \tag{1.69}$$

where $H^1\left(\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)^N\right)$ refers to the Sobolev space defined as the function space of all first derivatives belonging to $L^2\left(\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)^N\right)$.

A standard way to solve the problem (1.68), is by approximating the wavefunction Ψ as the product of separable functions, i.e tensor product wise, and by taking into account the antisymmetry constraint, as described in Equation (1.66). An ideal candidate is the Slater determinant. Consider a finite-dimensional subspace of $\boldsymbol{\mathcal{V}}_N$, with dimension d with $d \geq 2N$, with the following basis:

$$B_d := \left\{\varphi_{i_l}(\boldsymbol{x}_l, s_l) \in H^1\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right), i_l \in [d], l \in [N]\right\}, \tag{1.70}$$

where the functions φ_{i_l} are L^2-orthonormal.

Remark 1.3.2. In quantum chemistry, φ_{i_l} might be referred to as the *single-site basis functions*. They are commonly referred to as the *spin-orbital basis* since they depend on spatial coordinates $\boldsymbol{x}_{i_l} \in \mathbb{R}^3$ and the spin-variable $s = \pm\frac{1}{2}$.

We define the Slater determinant of N-particles by the following basis functions [105]:

$$\begin{aligned}
&\Phi_{[i_1,\ldots,i_N]}(\boldsymbol{x}_1, s_1, \cdots, \boldsymbol{x}_i, s_i, \cdots, \boldsymbol{x}_j, s_j, \cdots, \boldsymbol{x}_N, s_N) \\
&= \frac{1}{\sqrt{N!}} det \begin{pmatrix} \varphi_{i_1}(\boldsymbol{x}_1, s_1) & \varphi_{i_2}(\boldsymbol{x}_1, s_1) & \cdots & \varphi_{i_N}(\boldsymbol{x}_1, s_1) \\ \varphi_{i_1}(\boldsymbol{x}_2, s_2) & \varphi_{i_2}(\boldsymbol{x}_2, s_2) & \cdots & \varphi_{i_N}(\boldsymbol{x}_2, s_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{i_1}(\boldsymbol{x}_N, s_N) & \varphi_{i_2}(\boldsymbol{x}_N s_N) & \cdots & \varphi_{i_N}(\boldsymbol{x}_N, s_N) \end{pmatrix} \\
&= \frac{1}{\sqrt{N!}} \det\left(\varphi_{i_l}(\boldsymbol{x}_j, s_j)\right)_{l,j=1}^N,
\end{aligned} \tag{1.71}$$

Example 1.3.1. For $N = 2$, with N being the number of electrons. The Slater determinant of 2-particles in terms of two orthonormal functions $\varphi_1, \varphi_2 \in H^1\left(\mathbb{R}^3 \otimes \left\{\pm\frac{1}{2}\right\}\right)$ is:

$$\begin{aligned}
\Phi_{[1,2]}(\boldsymbol{x}_1, s_1, \boldsymbol{x}_2, s_2) &= \frac{1}{\sqrt{2}} det \begin{pmatrix} \varphi_1(\boldsymbol{x}_1, s_1) & \varphi_2(\boldsymbol{x}_1, s_1) \\ \varphi_1(\boldsymbol{x}_2, s_2) & \varphi_2(\boldsymbol{x}_2, s_2) \end{pmatrix} \\
&= \varphi_1(\boldsymbol{x}_1, s_1)\varphi_2(\boldsymbol{x}_2, s_2) - \varphi_1(\boldsymbol{x}_2, s_2)\varphi_2(\boldsymbol{x}_1, s_1).
\end{aligned} \tag{1.72}$$

It can be verified that $\Phi_{[1,2]}(\boldsymbol{x}_1, s_1, \boldsymbol{x}_2, s_2)$ satisfies the antisymmetry constraint defined in (1.66).

Now, we regard these Slater determinants as the basis functions spanning a finite-dimensional space known as the *Full Configuration Interaction (FCI) space*, denoted by $\boldsymbol{\mathcal{V}}_N^d$, and defined as: let d be the number of single-site basis functions $\varphi_{i_l},\ i_l \in [d], l \in [N]$ where N is the number of electrons. Let $\boldsymbol{\mathcal{V}}_N^d$ be defined as:

$$\boldsymbol{\mathcal{V}}_N^d := \mathrm{Span}\left\{\Phi_{[i_1,\ldots,i_N]}(\boldsymbol{x}_1, s_1, \ldots, \boldsymbol{x}_i, s_i, \ldots, \boldsymbol{x}_N, s_N) \middle| 1 \leq i_1 < \cdots < i_N \leq d, \boldsymbol{x}_k \in \mathbb{R}^3, s_k \in \left\{\pm\frac{1}{2}\right\} k \in [d]\right\}. \tag{1.73}$$

It can be seen that

$$\boldsymbol{\mathcal{V}}_N^d \subset \boldsymbol{\mathcal{V}}_N. \tag{1.74}$$

Typically, the solution to Equation (1.68), denoted as Ψ_0, is approximated by confining the space $\boldsymbol{\mathcal{V}}_N$ to a subspace $\boldsymbol{\mathcal{V}}_N^d$. This specific solution, denoted as $\Psi_0 \in \boldsymbol{\mathcal{V}}_N^d$, is commonly known as the FCI solution. It follows that Ψ_0 can be expressed as a linear combination of Slater determinants as follows:

$$\Psi_0 = \sum_{1 \leq i_1 < \cdots < i_N \leq d} C_{i_1 \ldots i_N} \Phi_{[i_1,\ldots,i_N]}, \tag{1.75}$$

with $C_{i_1 \ldots i_N} \in \mathbb{R}$ being the coefficients of this linear combination.

Remark 1.3.3. The representation of the wavefunction through a single Slater determinant, i.e., a function of the form described in Equation (1.71), is the so-called *discrete Hartree Fock (HF) approximation*, and the HF energy is the energy obtained by solving the minimization problem (1.68) over this single Slater determinant term.

For an N-electron system, the dimension of the space $\boldsymbol{\mathcal{V}}_N^d$ can be understood as the number of possible configurations the system can exhibit in a d-dimensional space. This dimension is given by the binomial coefficient:

$$\dim(\boldsymbol{\mathcal{V}}_N^d) = \begin{pmatrix} d \\ N \end{pmatrix} = \frac{d!}{N!(d-N)!}, \tag{1.76}$$

which signifies the various ways N electrons can be arranged among d single-site basis functions. As the number of electrons N increases, this dimension grows rapidly, approximately as $\mathcal{O}\left(d^N\right)$. Hence, effective approximation techniques become essential to render the problem tractable. A clever approach relies on parameterizing the coefficients in the linear combination presented in Equation (1.75) within second-quantization formalism. The second-quantization and this parameterization are detailed in Section 1.3.4. Then, by employing the Density Matrix Renormalization Group (DMRG) method, based on the tensor factorization of the newly parameterized coefficients, an approximate solution is obtained. The latter will also be described in Section 1.3.4.

1.3.3 Two-electron integrals tensor

Let N be the number of electrons, one can write the Hamiltonian operator as the sum of one-electron operator and two-electron operator as:

$$\hat{H} = -\sum_{i=1}^{N} h(\boldsymbol{x}_i) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{1.77}$$

where $h(\boldsymbol{x}_i), \boldsymbol{x}_i \in \mathbb{R}^3$,$i \in [N]$ is the so-called one-electron operator defined as:

$$h(\boldsymbol{x}_i) = \frac{1}{2}\Delta_i + \sum_{k=1}^{M} \frac{Z_k}{\|\boldsymbol{x}_i - \boldsymbol{r}_k\|}, \tag{1.78}$$

and $K(\boldsymbol{x}_i, \boldsymbol{x}_j), \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^3$, $i, j \in [N]$ is the so-called two-electron operator defined as:

$$K(\boldsymbol{x}_i, \boldsymbol{y}_j) = \frac{1}{\left\|\boldsymbol{x}_i - \boldsymbol{y}_j\right\|}. \tag{1.79}$$

Let d be the number of Slater determinant basis functions as defined in Equation (1.71) that span the finite-dimensional variational space $\boldsymbol{\mathcal{V}}_N^d$ as defined in Equation (1.73). Let $\mathbf{H}$ be the matrix representation of the Hamiltonian operator $\hat{H}$ within the basis of $\boldsymbol{\mathcal{V}}_N^d$ spanned by the Slater determinants defined in (1.71). Then the entries of the matrix representation of the Hamiltonian operator denoted by $\mathbf{H}(\mu, \nu)$ is defined as:

$$\begin{aligned} \mathbf{H}(\mu,\nu) &= \left\langle \Phi_\mu, \hat{H}\Phi_\nu \right\rangle_{L^2} \\ &= -\left\langle \Phi_\mu, \sum_{i=1}^{N} h(\boldsymbol{x}_i)\Phi_\nu \right\rangle_{L^2} + \frac{1}{2}\left\langle \Phi_\mu, \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j)\Phi_\nu \right\rangle_{L^2}, \end{aligned} \tag{1.80}$$

where $\mu := [i_1, \ldots, i_N]$ and $\nu := [j_1, \ldots, j_N], i_l \in [d], l \in [N]$. The evaluation of the L^2-inner products in (1.80) can be obtained by the *Slater-Condon Rules* [117]. These rules show that the first term in (1.80), $\left\langle \Phi_\mu, \sum_{i=1}^{N} h(\boldsymbol{x}_i)\Phi_\nu \right\rangle_{L^2}$, is written in terms of the so-called one-electron integrals and the second term $\left\langle \Phi_\mu, \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j)\Phi_\nu \right\rangle_{L^2}$ is written in terms of the so-called two-electron integrals. These integrals are defined in Equations (1.81) and (1.82) where $\{\varphi_{i_l}\}_{i_l\in[d], l\in[N]}$ are the single-site basis functions defined in Equation (1.70). We refer the reader to [117] for more details about the calculations of these inner products.

The one-electron integrals obtained from the Slater-Condon rules are defined as:

$$h_{i_1 j_1} = \sum_{s_1=\pm\frac{1}{2}} \int_{\mathbb{R}^3} \varphi_{i_1}(\boldsymbol{x}_1, s_1)\, h(\boldsymbol{x}_1)\varphi_{j_1}(\boldsymbol{x}_1, s_1)\, \mathrm{d}\boldsymbol{x}_1. \tag{1.81}$$

The two-electron integrals obtained from the Slater Condon rules are defined as:

$$v_{i_1 j_1 i_2 j_2} = \sum_{s_1, s_2 = \pm\frac{1}{2}} \int_{\mathbb{R}^3} \varphi_{i_1}(\boldsymbol{x}_1, s_1)\, \varphi_{i_2}(\boldsymbol{x}_2, s_2)\, K(\boldsymbol{x}_1, \boldsymbol{x}_2) \varphi_{j_1}(\boldsymbol{x}_1, s_1)\, \varphi_{j_2}(\boldsymbol{x}_2, s_2)\, \mathrm{d}\boldsymbol{x}_1\, \mathrm{d}\boldsymbol{x}_2, \tag{1.82}$$

where $i_1, i_2, j_1, j_2 \in [d]$. The order of the indices i_1, j_1, i_2, j_2 in (1.82) follows Mulliken's convention [44], i.e the indices i_1, j_1 label the single-site basis functions with variables $(\boldsymbol{x}_1, s_1)$ and the indices i_2, j_2 label the single-site basis functions with variables $(\boldsymbol{x}_2, s_2)$.

Remark 1.3.4. (Spatial-orbital basis functions [118]) The single-site basis functions $\varphi_{i_l}(\boldsymbol{x}_l, s_l)$, $i_l \in [d]$, $s_l = \pm\frac{1}{2}$, also referred to as the spin-orbital basis, are commonly written as the product of separable functions. For $l \in [N]$, one function in $H^1(\mathbb{R}^3)$ depends on the spatial coordinate $\boldsymbol{x}_l \in \mathbb{R}^3$ while the other function depends on the spin $s_l = \pm\frac{1}{2}$. Suppose that d is even, let us define the orthonormal basis set $\left\{\phi_{i_l}(\boldsymbol{x_l}) \in H^1(\mathbb{R}^3), i_l \in \left[\frac{d}{2}\right], l \in [N]\right\}$ such that [118], for each spatial-orbital of the form $\phi_{i_l}(\boldsymbol{x}_l)$, there are 2 spin-orbitals of the form $\varphi_{i_l}(\boldsymbol{x}_l, \frac{1}{2})$ and $\varphi_{i_l}(\boldsymbol{x}_l, \frac{-1}{2})$ and the following holds:

$$\begin{aligned} \varphi_{2i_l}(\boldsymbol{x}_l, s_l) &= \phi_{i_l}(\boldsymbol{x}_l)\eta_+(s_l),\ \eta_+(s_l) = \begin{cases} 1, & \text{if } s_l = \frac{1}{2}, \\ 0, & \text{otherwise,} \end{cases} \\ \varphi_{2i_l-1}(\boldsymbol{x}_l, s_l) &= \phi_{i_l}(\boldsymbol{x}_l)\eta_-(s_l),\ \eta_-(s_l) = \begin{cases} 1, & \text{if } s_l = \frac{-1}{2}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{1.83}$$

Hence, from $\frac{d}{2}$ spatial-orbital basis functions $\phi_{i_l},\ i_l \in \left[\frac{d}{2}\right]$, a set of d spin-orbital basis functions $\varphi_{i_l},\ i_l \in [d]$ can be formed. To ensure clarity and prevent confusion, moving forward, when referring to the number of orbitals, we will be using the number of spin-orbital basis functions denoted as d which is supposed to be even. Consequently, the number of spatial orbitals, denoted as d_{spatial}, is defined as half the number of spin-orbitals, i.e., $d_{\text{spatial}} = \frac{d}{2}$.

According to Remark 1.3.4 and in the interest of practicality and simplification, it is common to overlook the spin indices denoted by s_1, s_2 in the Equations (1.81) and (1.82) when dealing with integrals. This can be justified as follows: we define $h_{\tilde{i}_1, \tilde{j}_1}$ for $\tilde{i}_1, \tilde{j}_1 \in [d_{\text{spatial}}]$ as follows:

$$h_{\tilde{i}_1, \tilde{j}_1} = h_{2i_1, 2j_1} + h_{2i_1-1, 2j_1} + h_{2i_1, 2j_1-1} + h_{2i_1-1, 2j_1-1},\ i_1,\ j_1 \in [d]\,. \tag{1.84}$$

According to Equations (1.83), we have $h_{2i_1-1, 2j_1} = h_{2i_1, 2j_1-1} = 0$. This yields:

$$\begin{aligned} h_{\tilde{i}_1, \tilde{j}_1} &= h_{2i_1, 2j_1} + h_{2i_1-1, 2j_1-1} \\ &= 2 \int_{\mathbb{R}^3} \phi_{\tilde{i}_1}(\boldsymbol{x}_1)\, h(\boldsymbol{x}_1) \phi_{\tilde{j}_1}(\boldsymbol{x}_1)\, \mathrm{d}\boldsymbol{x}_1, \tilde{i}_1, \tilde{j}_1 \in [d_{\text{spatial}}]\,. \end{aligned} \tag{1.85}$$

Note here that the one-electron integrals $h_{\tilde{i}_1, \tilde{j}_1}$ do not depend on the spin-variable $s_1 = \pm\frac{1}{2}$, i.e the indices s_1 are dropped. However, for each couple of indices $\tilde{i}_1, \tilde{j}_1 \in [d_{\text{spatial}}]$, we need two one-electron integrals of the form $\int_{\mathbb{R}^3} \phi_{\tilde{i}_1}(\boldsymbol{x}_1)\, h(\boldsymbol{x}_1) \phi_{\tilde{j}_1}(\boldsymbol{x}_1)\, \mathrm{d}\boldsymbol{x}_1$. In a similar way, we employ a comparable approach to handle the two-electron integrals outlined in Equation (1.82). This adaptation allows us to work within the space of d spatial basis functions, specifically within $H^1(\mathbb{R}^3)$. Consequently, the modified form of the integrals can be expressed as follows, for $i_1, j_1, i_2, j_2 \in [d]$:

$$h_{i_1 j_1} = \int_{\mathbb{R}^3} \phi_{i_1}(\boldsymbol{x}) h(\boldsymbol{x}) \phi_{j_1}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1.86}$$

and

$$v_{i_1 j_1 i_2 j_2} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \phi_{i_1}(\boldsymbol{x}_1)\, \phi_{i_2}(\boldsymbol{x}_2)\, K(\boldsymbol{x}_1, \boldsymbol{x}_2) \phi_{j_1}(\boldsymbol{x}_1)\, \phi_{j_2}(\boldsymbol{x}_2)\, \mathrm{d}\boldsymbol{x}_1\, \mathrm{d}\boldsymbol{x}_2, \tag{1.87}$$

with $\phi_{2l-1}(\boldsymbol{x}) = \phi_{2l}(\boldsymbol{x})$, $l \in \left[\frac{d}{2}\right]$. Given integration rules, these integrals exhibit symmetry properties given as follows:

$$h_{i_1 j_1} = h_{j_1 i_1}, \tag{1.88}$$

and

$$v_{i_1j_1i_2j_2} = v_{i_2j_2i_1j_1} = v_{i_2j_2j_1i_1} = v_{j_2i_2j_1i_1} = v_{j_2i_2i_1j_1} = v_{i_1j_1j_2i_2} = v_{j_1i_1j_2i_2} = v_{j_1i_1i_2j_2}. \quad (1.89)$$

To simplify the mathematical treatment of spatial-orbital basis functions, commonly known as molecular orbitals, in many quantum chemistry methods, the representation of these functions are expanded over alternative basis functions. Let $B_g = \{g_\mu(\boldsymbol{x}) \in H^1(\mathbb{R}^3), \mu \in [N_b]\}$ be a finite-dimensional basis of dimension N_b. The basis functions $\phi_i, i \in [d]$, are represented as follows:

$$\phi_i(\boldsymbol{x}) = \sum_{\mu=1}^{N_b} C_{\mu i} g_\mu(\boldsymbol{x}),\ i \in [d], \boldsymbol{x} \in \mathbb{R}^3, \quad (1.90)$$

where $C_{\mu i} \in \mathbb{R}$ represent the expansion coefficients of the spatial basis functions.

Remark 1.3.5. In the literature [117], Equation (1.90) corresponds to the so-called Linear Combination of Atomic Orbitals (LCAO) theory, where $g_\mu, \mu \in [N_b]$ function are referred to as the atomic orbitals.

Plugging the expansion (1.90) in the Equation (1.87) yields the so-called two-electron integrals corresponding to the finite-basis $\{g_\mu\}_{1 \le \mu \le N_b}, g_\mu \in H^1(\mathbb{R}^3)$:

$$\mathcal{B}_{\mu\nu\kappa\lambda} = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{g_\mu(\boldsymbol{x}) g_\nu(\boldsymbol{x}) g_\kappa(\boldsymbol{y}) g_\lambda(\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|} d\boldsymbol{x}d\boldsymbol{y},\ with\ \mu,\nu,\kappa,\lambda \in [N_b]. \quad (1.91)$$

The choice of the basis is restricted by the analytic integrability and required accuracy for an efficient computation of these integrals. In literature [84], the basis set commonly used are Slater type functions or Cartesian Gaussian type functions. The Slater type functions are characterised by quantum numbers[1], i.e set of parameters that describe various properties of electrons, denoted by n, l, m and exponents ζ. Slater type functions are of the form:

$$f_{nlm}(\boldsymbol{x}, \theta, \phi) = (\boldsymbol{x}-\boldsymbol{r})^{n-1} e^{-\zeta\|\boldsymbol{x}-\boldsymbol{r}\|} Y_l^m(\theta,\phi), \boldsymbol{x}, \boldsymbol{r} \in \mathbb{R}^3, \quad (1.92)$$

where every μ number corresponds to specific quantum numbers (n, l, m), $\boldsymbol{x}$ is the position of the electron, $Y_l^m(\theta,\phi)$ are the spherical harmonic functions defined on the spherical angular coordinates θ and ϕ, and $\boldsymbol{r}$ refers to the coordinates of the atom nucleus that is fixed and known in practice. On the other hand, cartesian Gaussian type functions, also-called *primitive Gaussian type functions* are of the form:

$$f_\mu(\boldsymbol{x}) = \prod_{l=1}^{3} (\boldsymbol{x}_l - \boldsymbol{r}_l)^{\boldsymbol{p}_\mu(l)} e^{-\alpha_\mu \|(\boldsymbol{x}_l - \boldsymbol{r}_l)\|^2},\ \boldsymbol{x}_l,\ \boldsymbol{r}_l \in \mathbb{R}, \mu \in [N_b], \quad (1.93)$$

where the exponent α_μ is a parameter whose reference value is found, for instance, in [94] and $\boldsymbol{p}_\mu$ is a vector with three coordinates which are exponents depending on the chosen basis function, i.e value of μ, see example 1.3.2.

Example 1.3.2. Let us consider the electronic configurations of two example molecules: water (H_2O) and Di-Hydrogen (H_2) as depicted in Figure 1.9.

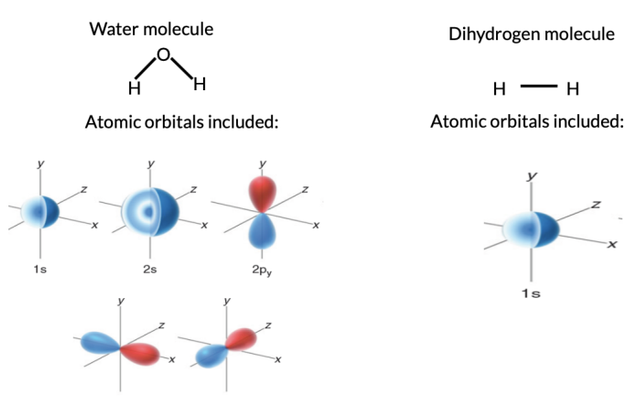


Figure 1.9: Electronic configurations of Water molecule as well as Di-Hydrogen molecule [58].

For the water molecule, each Hydrogen atom has one s atomic orbital denoted by $1s$, while the Oxygen atom has 5 atomic orbitals: two s atomic orbitals denoted by $1s, 2s$ and three atomic orbitals p_x, p_y, p_z denoted by $2p_x, 2p_y, 2p_z$. For the Di-Hydrogen molecule, each Hydrogen atom has one s atomic orbital denoted by $1s$. Relating this example to the index μ and $\boldsymbol{p}_\mu$: each μ corresponds to an atomic orbital. If μ corresponds to s, then $\boldsymbol{p}_\mu = (0, 0, 0)$; if μ corresponds to p_x, then $\boldsymbol{p}_\mu$=(1,0,0); if μ corresponds to p_y, then $\boldsymbol{p}_\mu$=(0,1,0); and if μ corresponds to p_z, then $\boldsymbol{p}_\mu$=(0,0,1). Overall, the tuple values $\boldsymbol{p}_\mu$ for $\mu \in [N_b]$ depend on the electronic configuration of the selected molecule. Each μ can be associated to a specific atomic orbital within the constituent atoms of the molecule.

The analytical properties of Gaussian-type basis functions make them the dominant choice in many molecular quantum chemistry methods. Indeed, the computation of integrals (1.87) is simple when employing Gaussian-type functions, in contrast to Slater-type functions. The difficulties in employing Slater-type functions arise when attempting to efficiently compute the product of two such functions situated on distinct nucleus centers. Furthermore, a common practice involves the linear combination of multiple Gaussian-type functions to fulfill the cusp condition (discontinuous derivative) at $\boldsymbol{x} = \boldsymbol{r}$, with $\boldsymbol{x}$ being the position of electrons and $\boldsymbol{r}$ being the coordinates of the atom nucleus, we refer the interested reader to [34] for a more comprehensive understanding of the choice of the basis functions. It follows that each g_μ function is expressed as a linear combination of primitive Gaussian functions, defined in (1.93). This results into the following expansion of the basis functions $\{g_\mu\}_{1\leq\mu\leq N_b}, g_\mu \in H^1(\mathbb{R}^3)$:

$$g_\mu(\boldsymbol{x}) = \sum_{j=1}^{I_\mu} c_j f_\mu^{(j)}(\boldsymbol{x}), I_\mu \in \mathbb{N}, \boldsymbol{x} \in \mathbb{R}^3, \tag{1.94}$$

where $f_\mu^{(j)}$ corresponds to the j-th primitive Gaussian function, as defined in Equation (1.93), $c_j \in \mathbb{R}$ are the expansion coefficients and I_μ is the number of primitive Gaussians used in the linear combination.

Once the type of the basis functions is decided, an additional consideration is the number of basis functions $\{g_\mu\}_{1\leq\mu\leq N_b}$, N_b. This typically depends on the choice of the basis set as exemplified in the Table 1.1 for the molecule H_2O. Indeed, the accurate description of the basis functions $g_\mu(\boldsymbol{x})$ depends on the choice of expansion coefficients and exponents which defines the so-called basis set. We refer the reader to [45] for more details and explanations of Gaussian basis sets for molecular simulations.

Basis set	STO-3G	6-311G	CC-PVDZ
N_b	7	19	25

Table 1.1: Example: total number of basis functions of molecules vs choice of basis [94].

Now, back to Equation (1.91), the latter represents six-dimensional integrals which constitute the entries of a 4-th order tensor, referred to as $\mathcal{B}$, such as $\mathcal{B}(\mu;\nu;\kappa;\lambda) := \mathcal{B}_{\mu\nu\kappa\lambda}$, with $\mu,\ \nu,\ \kappa,\ \lambda \in [N_b]$. This tensor has $O(N_b^4)$ entries with N_b being the number of basis functions $\{g_\mu\}_{1\leq\mu\leq N_b}$. Considerable efforts have been devoted to minimize the cost of the integrals evaluation which is a challenging computational problem as they are at the core of many quantum chemistry calculations. Note that computing this tensor requires the evaluation of $\mathcal{O}(N_b^4)$ six-dimensional integrals that are singular due to the presence of the Coulomb potential, or the two-electron operator $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}$ and where N_b increases drastically with the molecular system size and the choice of the basis set, $N_b = \mathcal{O}(d_{\text{spatial}})$, with d_{spatial} being the number of spatial-orbital basis functions.

Numerous studies can be found in existing literature, covering both analytical and numerical methods to calculate these integrals, as outlined in Chapter 2. One of these methods involves employing a regularization technique through the range separation of the two-electron operator described in Equation (1.79). This operator, which is also-called the Coulomb potential, is split into a smooth long range-part and a singular short-range part. The range separation technique and the efficient numerical evaluation of the resulting long-range two-electron integrals will be the focal point and one of the main contributions of the Chapter 2.

Finally, let us point out that computing the integrals consists in one of the first computational steps within numerous quantum chemistry approaches for the search of the ground-state and the corresponding ground-state energy. Among these methods, in what follows, we review the DMRG approach, a well known approach to numerically approximate the eigenfunction of the Hamiltonian operator associated to the lowest eigenvalue.

1.3.4 Density Matrix Renormalization Group (DMRG)

As discussed in Section 1.3, the pioneering DMRG approach introduced by White [131] in 1992 and later rediscussed in the mathematical community under the name of MALS method [48], a modification of ALS, is a powerful optimization technique for tackling high-dimensional linear systems or eigenvalue problems. Further details about this method and its various applications can be found in [48, 106, 118, 127, 131].

In the physics and chemistry community, the DMRG approach finds significant application in finding the ground-states of Hamiltonians of quantum many-body systems. As already discussed in Section 1.2.4, this is achieved through the reformulation of the linear eigenvalue problem as the minimization of a Rayleigh quotient, given as:

$$R_{\mathbf{H}}(\mathbf{\Psi}) = \frac{\langle \mathbf{\Psi}, \mathbf{H}\mathbf{\Psi} \rangle}{\langle \mathbf{\Psi}, \mathbf{\Psi} \rangle}. \tag{1.95}$$

Here, $\mathbf{\Psi} \in \mathbb{R}^{n_1 \dots n_d}$ represents the eigenfunction, which is a vector of dimension $n = \prod_{i=1}^{d} n_i$, $d \in \mathbb{N}$, and $\mathbf{H} \in \mathbb{R}^{(n_1 \dots n_d) \times (n_1 \dots n_d)}$ denotes the symmetric matrix that represents the Hamiltonian operator, in the context of the second quantization formalism. We now explain the rudiments of the second quantization approach.

Discrete Fock space and second quantization

Let N be the number of electrons, and let d be the number of Slater determinants spanning the finite-dimensional variational space $\boldsymbol{\mathcal{V}}_N^d$ (also-called the FCI space). This space is defined as follows (see Equation (1.73)):

$$\boldsymbol{\mathcal{V}}_N^d := \text{Span}\left\{ \Phi_{[i_1,\dots,i_N]}(\boldsymbol{x}_1, s_1, \dots, \boldsymbol{x}_i, s_i, \dots, \boldsymbol{x}_N, s_N) \Big| 1 \leq i_1 < \cdots < i_N \leq d, s_k \in \left\{ \pm \frac{1}{2} \right\}, k \in [N] \right\}, \tag{1.96}$$

with $\Phi_{[i_1,\dots,i_N]}$ being the Slater determinant basis functions defined in Equation (1.71) for $i_k \in [d], k \in [N]$. Then, we define the discrete Fock space $\boldsymbol{\mathcal{F}}_d$ as:

$$\boldsymbol{\mathcal{F}}_d := \bigoplus_{N=0}^{d} \boldsymbol{\mathcal{V}}_N^d, \tag{1.97}$$

The discrete Fock space is defined as the direct sum of subspaces $\boldsymbol{\mathcal{V}}_N^d$ (larger than the FCI space). The dimension of the Fock space is:

$$\dim(\boldsymbol{\mathcal{F}}_d) = \sum_{N=0}^{d} \dim(\boldsymbol{\mathcal{V}}_N^d) = \sum_{N=0}^{d} \binom{d}{N} = 2^d. \tag{1.98}$$

For all $\Psi, \Phi \in \boldsymbol{\mathcal{F}}_d$, the discrete Fock space is a Hilbert space equipped with the following inner product:

$$\langle \Psi, \Phi \rangle_{\boldsymbol{\mathcal{F}}_d} = \sum_{k=0}^{d} \langle \Psi^k, \Phi^k \rangle_{L^2}, \quad \Psi = \bigoplus_{N=0}^{d} \Psi^N, \Phi = \bigoplus_{N=0}^{d} \Phi^N, \Psi^N, \Phi^N \in \boldsymbol{\mathcal{V}}_N^d. \tag{1.99}$$

It is important to discuss the alternative representation of the elements within the Fock space, i.e Slater determinants. Note that a clever way to obtain a simpler representation of Slater determinants $\Phi_{[i_1,\dots,i_N]}$, $i_l \in [d]$, $l \in [N]$, is to use binary labeling $(\xi_1, \dots, \xi_d) \in \{0,1\}^d$ as explained in the following definition.

Definition 21. (Occupation number representation [118]) Consider a binary tuple $(\xi_1, \dots, \xi_d)$ where $\xi_i \in \{0,1\}$ for $i \in [d]$, representing the presence or absence of spin-orbital basis functions φ_i in the Slater determinant function $\Phi_{[i_1,\dots,i_N]}$. If $\xi_i = 1$, we designate the orbital φ_i, as defined in (1.70), as occupied in the Slater determinant. Conversely, if $\xi_i = 0$,

we indicate that the orbital φ_i is unoccupied in the Slater determinant. The notation used is:

$$\Phi_{\xi_1 \ldots \xi_d} := \Phi_{[i_1,\ldots,i_N]}, \xi_i \in \{0,1\}, i \in [d]. \tag{1.100}$$

Furthermore, for any eigenfunction $\Psi \in \boldsymbol{\mathcal{F}}_d$, Ψ can be expressed as:

$$\Psi = \sum_{\xi_1 \ldots \xi_d \in \{0,1\}^d} \Psi_{\xi_1 \ldots \xi_d} \Phi_{\xi_1 \ldots \xi_d}, \tag{1.101}$$

with $\Psi_{\xi_1 \ldots \xi_d} \in \mathbb{R}$ being the coefficients of the expansion over the new representation of the Slater determinants basis functions.

Let $\Psi^N \in \boldsymbol{\mathcal{V}}_N^d$ be an N-electron eigenfunction, following the previously defined expression of Ψ^N in Equation (1.75), as:

$$\Psi^N = \sum_{1 \leq i_1 < \cdots < i_N \leq d} C_{i_1 \ldots i_N} \Phi_{[i_1,\ldots,i_N]}, \tag{1.102}$$

where $C_{i_1 \ldots i_N} \in \mathbb{R}, i_l \in [d], l \in [N]$. Following Definition 21, Ψ^N can be associated to the new representation of the eigenfunction as follows:

$$\Psi^N := \sum_{\xi_1 \ldots \xi_d \in \{0,1\}^d} \Psi_{\xi_1 \ldots \xi_d} \Phi_{\xi_1 \ldots \xi_d}, \tag{1.103}$$

with

$$\Psi_{\xi_1 \ldots \xi_d} = \begin{cases} 0 & \text{if } \sum_{i=1}^d \xi_i \neq N, \\ C_{i_1 \ldots i_N} & \text{if } \xi_i = 1 \text{ when } i \in \{i_1, \ldots, i_N\}, i_1 < \ldots < i_N. \end{cases} \tag{1.104}$$

We note that by the normalization of Ψ^N, we have:

$$\sum_{\xi_1 \ldots \xi_d \in \{0,1\}^d} |\Psi_{\xi_1 \ldots \xi_d}|^2 = 1. \tag{1.105}$$

To make this more concrete, we consider the following example.

Example 1.3.3. Let us consider a 2-electron eigenfunction, $N = 2$, expanded on 6 Slater determinant basis functions, $d = 4$, such that

$$\begin{aligned} \Psi^2 &= \sum_{1 \leq i_1 < i_2 \leq 4} C_{i_1 i_2} \Phi_{[i_1,i_2]} \\ &= C_{12}\Phi_{[12]} + C_{13}\Phi_{[13]} + C_{23}\Phi_{[23]} + C_{14}\Phi_{[14]} + C_{24}\Phi_{[24]} + C_{34}\Phi_{[34]}. \end{aligned} \tag{1.106}$$

The first term in Equation (1.106), represented as $\Phi_{[12]}$, signifies the occupation of only the spin-orbitals φ_1 and φ_2, with φ_3 and φ_4 remaining unoccupied, given that $d = 4$. Following the Definition 21, $\Phi_{[12]}$ is associated to Φ_{1100}. The same reasoning applies for the remaining terms, leading to the subsequent equivalent expression:

$$\Psi^2 := \Psi_{1100}\Phi_{1100} + \Psi_{1010}\Phi_{1010} + \Psi_{0110}\Phi_{0110} + \Psi_{1001}\Phi_{1001} + \Psi_{0101}\Phi_{0101} + \Psi_{0011}\Phi_{0011}. \tag{1.107}$$

With this new parameterization of Slater determinant basis functions, as defined in Definition 21, one can perform transformations on these basis functions by adjusting their occupation numbers. Through these transformations, it becomes possible to transition from the space $\boldsymbol{\mathcal{V}}_N^d$ to $\boldsymbol{\mathcal{V}}_{N+1}^d$ using the creation operator and from $\boldsymbol{\mathcal{V}}_N^d$ to $\boldsymbol{\mathcal{V}}_{N-1}^d$ using the annihilation operator. We define now these operators as follows:

Definition 22. (Creation and annihilation operators [118]) Let N be the number of electrons, let $\boldsymbol{\mathcal{V}}_N^d$ be the finite-dimensional variational space as defined in (1.96), spanned by d Slater determinant basis function. Using the occupation number representation, see Definition 21, for a single Slater determinant basis functions, $\Phi_{\xi_1 \ldots \xi_d} \in \boldsymbol{\mathcal{V}}_N^d$, $\xi_l \in \{0,1\}$, $l \in [d]$, we define the annihilation operator, denoted by $a_i : \boldsymbol{\mathcal{V}}_N^d \to \boldsymbol{\mathcal{V}}_{N-1}^d$, $i \in [d]$, as the operator acting on $\Phi_{\xi_1 \ldots \xi_d} \in \boldsymbol{\mathcal{V}}_N^d$ such that the following holds:

$$a_i \Phi_{\xi_1,\ldots,\xi_d} := \begin{cases} 0 & if\ \xi_i = 0 \\ (-1)^{\ell_i - (i-1)} \Phi_{\xi_1,\ldots,\xi_i - 1,\ldots,\xi_d} & otherwise. \end{cases} \tag{1.108}$$

We define the creation operator, denoted by $a_i^* : \boldsymbol{\mathcal{V}}_N^d \to \boldsymbol{\mathcal{V}}_{N+1}^d, i \in [d]$, as the operator acting on $\Phi_{\xi_1 \ldots \xi_d} \in \boldsymbol{\mathcal{V}}_N^d$ such that the following holds:

$$a_i^* \Phi_{\xi_1,\ldots,\xi_d} := \begin{cases} 0 & if\ \xi_i = 1 \\ (-1)^{\ell_i - (i-1)} \Phi_{\xi_1,\ldots,\xi_i+1,\ldots,\xi_d} & otherwise, \end{cases} \tag{1.109}$$

with

$$\ell_i = \sum_{l=1}^{i-1} \xi_l, \xi_l \in \{0,1\}, i \in [d]. \tag{1.110}$$

The annihilation operator, represented here as a_i, operates on Slater determinants, transforming them into other Slater determinants with a reduced number of particles, effectively destroying one particle at the spin-orbital indexed by $i, i \in [d]$. Whereas, the creation operator, represented here as a_i^*, operates on Slater determinants, transforming them into other Slater determinants with an increased number of particles, effectively creating one particle at the spin-orbital indexed by $i, i \in [d]$.

To make again this more concrete, we consider the following example.

Example 1.3.4. Let us consider a 2-electron eigenfunction, $N = 2$, expanded on 6 Slater determinant basis functions, $d = 4$, such that according to Example 1.3.3:

$$\Psi^2 := \Psi_{1100}\Phi_{1100} + \Psi_{1010}\Phi_{1010} + \Psi_{0110}\Phi_{0110} + \Psi_{1001}\Phi_{1001} + \Psi_{0101}\Phi_{0101} + \Psi_{0011}\Phi_{0011}, \tag{1.111}$$

Restricting our focus to the initial Slater determinant basis, denoted as Φ_{1100} in Equation (1.111), we aim to either introduce a particle into the first orbital or eliminate one, and this can be reformulated, using Definition 22, as:

$$a_1\Phi_{1100} = \Phi_{0100},\ a_1^*\Phi_{1100} = 0. \tag{1.112}$$

The result of $a_1\Phi_{1100}$ is interpreted as the annihilation of a particle at the first occupied orbital, and the outcome of $a_1^*\Phi_{1100}$ is understood as the creation of a particle at the first occupied orbital. However, in accordance with Pauli's exclusion principle[117], the latter operation results in 0.

For a more abstract representation of these operators, in the literature, they serve as generators for the algebra of canonical anti-commutation relations (CAR) [95]. In this specific case, they operate over a finite-dimensional Hilbert space and adhere to the so-called *anti-commutation relations*, expressed as follows for $i, j \in [d]$:

$$\begin{aligned} a_i a_j + a_j a_i &= 0, \\ a_i^* a_j^* + a_j^* a_i^* &= 0, \\ a_i a_j^* + a_j^* a_i &= \delta_{ij}. \end{aligned} \tag{1.113}$$

By the Slater-Condon rules, employed to derive the expression of the one and two-electron integrals as defined in Equations (1.86), (1.87), the electronic Hamiltonian operator $\hat{H}$ acting on the Fock space $\boldsymbol{\mathcal{F}}_d$ in terms of the creation and annihilation operators, can be represented as follows (for more details on its derivation, refer to [117]):

$$\hat{H} = \sum_{i,j=1}^{d} h_{ij} a_i^* a_j + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d} v_{ijk\ell} a_i^* a_k^* a_\ell a_j, \tag{1.114}$$

where h_{ij} and $v_{ijk\ell}$ are the one and two-electron integrals defined in (1.86) and (1.87). Equation (1.114) is a formulation of the Schrödinger equation, defined in (1.62), in the so-called second quantization formalism. It is common to view $\hat{H}$ as a sum of 1-body terms (the initial terms in the sum (1.114), of the form $a_i^* a_j$) and 2-body terms (the subsequent terms in the sum (1.114), of the form $a_i^* a_k^* a_\ell a_j$).

Remark 1.3.6. In the physics and chemistry communities, this Hamiltonian defined in Equation 1.114 is also referred to as a 2-body Hamiltonian operator due to the presence of up to 2-body interactions (second term in Equation (1.114)). It is also commonly known as the molecular Hamiltonian, the quantum chemical Hamiltonian operator, the second-quantized fermionic Hamiltonian, or the ab initio electronic Hamiltonian.

Note that by introducing the Fock space, we have lost track of the total number of particles N, i.e the Fock space can be seen as a space with all possible combinations of states with different number of particles. To reinstate this constraint, we define the particle number operator, denoted as $\hat{N} : \boldsymbol{\mathcal{V}}_N^d \to \boldsymbol{\mathcal{V}}_N^d$, as follows:

$$\hat{N} = \sum_{i=1}^{d} a_i^* a_i. \tag{1.115}$$

Let $\Psi \in \boldsymbol{\mathcal{V}}_N^d \subset \boldsymbol{\mathcal{F}}_d$ be a wavefunction with N electrons. Using Equations (1.103),(1.104), (1.108), (1.109), and (1.115) it can be showed that

$$\hat{N}\Psi = N\Psi. \tag{1.116}$$

By considering the constraint over the particle number N, one needs to solve the constrained minimization problem (1.68) that writes:

$$\min \left\{ \langle \Psi, \hat{H}\Psi \rangle_{\boldsymbol{\mathcal{F}}_d}, \Psi \in \boldsymbol{\mathcal{F}}_d, \|\Psi\|_{\boldsymbol{\mathcal{F}}_d} = 1, \hat{N}\Psi = N\Psi \right\}, \tag{1.117}$$

where $\langle ., . \rangle_{\boldsymbol{\mathcal{F}}_d}$ is defined in Equation (1.99) and $\|\Psi\|^2_{\boldsymbol{\mathcal{F}}_d} = \langle \Psi, \Psi \rangle_{\boldsymbol{\mathcal{F}}_d}$.

Now, by expressing Slater determinants through binary labeling, see Definition 21, it is common to associate each Slater determinant $\Phi_{\xi_1 \dots \xi_d} \in \boldsymbol{\mathcal{F}}_d$ with exactly one unit vector $\boldsymbol{e}_{\mu_1 \dots \mu_d} \in \mathbb{R}^{2^d}$ [118]. We define these unit vectors as follows: let $(\xi_1, \dots, \xi_d)$ be a binary tuple which labels the Slater determinant $\Phi_{\xi_1 \dots \xi_d}$, such that for each index $\xi_k \in \{0, 1\}, k \in [d]$, we associate a unit vector $\boldsymbol{e}_{\mu_k} \in \mathbb{R}^2$ with $\mu_k \in \{1, 2\}$. Let us define the following mapping function:

$$q_2 : \{1, 2\} \to \{0, 1\}. \tag{1.118}$$

This function serves as a one-to-one mapping between indices and occupation numbers. This corresponds to labeling two possible occupation states where 1 or $|-\rangle$ refers to absent (the occupation number is 0) , 2 or $|1\rangle$ refers to occupation with a particle (the occupation number is 1). Now, each unit vector $\boldsymbol{e}_{\mu_k},\ k \in [d]$, is defined as:

$$\boldsymbol{e}_{\mu_k} = \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{if } \xi_k = q_2(\mu_k) = 0, \\ \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{if } \xi_k = q_2(\mu_k) = 1. \end{cases} \tag{1.119}$$

We define $\boldsymbol{e}_{\mu_1 \mu_2 \dots \mu_d} \in \mathbb{R}^{2^d}$ as follows:

$$\boldsymbol{e}_{\mu_1 \mu_2 \dots \mu_d} = \boldsymbol{e}_{\mu_1} \otimes_K \boldsymbol{e}_{\mu_2} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_d}. \tag{1.120}$$

By associating each Slater determinant $\Phi_{\xi_1 \dots \xi_d} \in \boldsymbol{\mathcal{F}}_d$ with exactly one unit vector $\boldsymbol{e}_{\mu_1 \dots \mu_d} \in \mathbb{R}^{2^d}$, we have an isomorphism between the Fock space and $\mathbb{R}^{2^d}$. Let us denote this isomorphism by $g : \boldsymbol{\mathcal{F}}_d \to \mathbb{R}^{2^d}$, such that

$$g(\Phi_{\xi_1 \dots \xi_d}) := g(\Phi_{q_2(\mu_1) \dots q_2(\mu_d)}) := \boldsymbol{e}_{\mu_1 \dots \mu_d},\ g(\Psi) := \boldsymbol{\Psi} \in \mathbb{R}^{2^d}, \Psi \in \boldsymbol{\mathcal{F}}_d. \tag{1.121}$$

For example, consider the Slater determinant Φ_{1100}, with $d = 4$, following Equation (1.121), we have:

$$\begin{aligned} g(\Phi_{1100}) &:= g(\Phi_{q_2(2)q_2(2)q_2(1)q_2(1)}) := \boldsymbol{e}_{2211} \\ &= \boldsymbol{e}_2 \otimes_K \boldsymbol{e}_2 \otimes_K \boldsymbol{e}_1 \otimes_K \boldsymbol{e}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes_K \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes_K \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes_K \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{1.122}$$

By embracing this viewpoint, the creation and annihilation operators can be seen from a linear algebra perspective as high-dimensional square matrices defined as follows (see [118]): the matrix representation of the annihilation operator, denoted as $\mathbf{A}_i \in \mathbb{R}^{2^d \times 2^d}$ for $i \in [d]$, is given by:

$$\begin{aligned} g \circ a_i \circ g^{-1} := \mathbf{A}_i &= \left(\bigotimes_{k=1}^{i-1}{}_K \mathbf{S} \right) \otimes_K \mathbf{A} \otimes_K \left(\bigotimes_{k=i+1}^{d}{}_K \mathbf{I}_2 \right) \\ &= \underbrace{\mathbf{S} \otimes_K \dots \otimes_K \mathbf{S}}_{i-1 \text{ terms}} \otimes_K \mathbf{A} \otimes_K \underbrace{\mathbf{I}_2 \otimes_K \dots \otimes_K \mathbf{I}_2}_{d-i-1 \text{ terms}}, \end{aligned} \tag{1.123}$$

where $\circ$ refers to the operation of function composition. The matrix representation of the creation operator, which is defined as the transpose of $\mathbf{A}_i \in \mathbb{R}^{2^d \times 2^d}$, see Equation (1.123), is given by:

$$\begin{aligned} g \circ a_i^* \circ g^{-1} := \mathbf{A}_i^* &= \left(\bigotimes_{k=1}^{i-1}{}_K \mathbf{S}\right) \otimes_K \mathbf{A}^* \otimes_K \left(\bigotimes_{k=i+1}^{d}{}_K \mathbf{I}_2\right) \\ &= \mathbf{S} \otimes_K \cdots \otimes_K \mathbf{S} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I}_2 \otimes_K \cdots \otimes_K \mathbf{I}_2, \end{aligned} \tag{1.124}$$

where

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{1.125}$$

Remark 1.3.7. One can represent $\mathbf{A}_i$ concisely using as follows:

$$\mathbf{A}_i = \bigotimes_{l=1}^{d}{}_K \mathbf{X}_l, \quad \mathbf{X}_l = \begin{cases} \mathbf{S} & \text{if } l < i, \\ \mathbf{A} & \text{if } l = i, \\ \mathbf{I} & \text{if } l > i. \end{cases} \quad i \in [d]\,. \tag{1.126}$$

In this work, we will frequently encounter partial representations of Kronecker product structures as in (1.126), which can be defined as:

$$(\mathbf{A}_i)^{<s} = \bigotimes_{l=1}^{s-1}{}_K \mathbf{X}_l, \quad (\mathbf{A}_i)^{\leq s} = \bigotimes_{l=1}^{s}{}_K \mathbf{X}_l, \tag{1.127}$$

and similarly

$$(\mathbf{A}_i)^{>s} = \bigotimes_{l=s+1}^{d}{}_K \mathbf{X}_l, \quad (\mathbf{A}_i)^{\geq s} = \bigotimes_{l=s}^{d}{}_K \mathbf{X}_l, \tag{1.128}$$

where $s \in [d]$. As for the Hamiltonian operator, we define it as:

$$g \circ \hat{H} \circ g^{-1} : \mathbb{R}^{2^d} \to \mathbb{R}^{2^d}, \tag{1.129}$$

where $\hat{H}$ is defined in Equation (1.114). The matrix representation of the Hamiltonian operator, denoted as $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$, is expressed as follows:

$$g \circ \hat{H} \circ g^{-1} = \mathbf{H} = \sum_{i,j=1}^{d} h_{ij} \mathbf{A}_i^* \mathbf{A}_j + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d} v_{ijk\ell} \mathbf{A}_i^* \mathbf{A}_k^* \mathbf{A}_\ell \mathbf{A}_j, \tag{1.130}$$

and the matrix representation of the particle number operator is given by:

$$g \circ \hat{N} \circ g^{-1} = \mathbf{N} = \sum_{i=1}^{d} \mathbf{A}_i^* \mathbf{A}_i. \tag{1.131}$$

One can easily verify that the Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$ is symmetric.

Remark 1.3.8. Considering the matrix representation structure of the creation and annihilation operators, it is crucial to emphasize that the Hamiltonian in (1.130) is formulated as a sum involving a series of Kronecker products of matrices. Each term within this sum, constituting a Kronecker product of multiple matrices, is a rank-1 tensor train. In this context, every TT-core is effectively represented by an individual matrix derived from that particular Kronecker product.

Remark 1.3.9. For simplicity in representation, we have employed the binary occupation number representation, i.e dealing only with spin-orbital basis function $\varphi_i, i \in [d]$. However, in Quantum Chemical DMRG (QC-DMRG), as it will be described in the following subsections, it is more common to deal with spatial-basis functions denoted by $\phi_i, i \in [d_{\text{spatial}}]$, with $d = 2d_{\text{spatial}}$ see Equation (1.83). In this framework, each spatial-orbital basis function ϕ_i is associated to two spin-orbital basis functions φ_i. Therefore, there is a modification of the occupation number for the Slater determinant. We consider a tuple $(\xi_1, \ldots, \xi_{d_{\text{spatial}}}) \in \{0, 1, 2\}^{d_{\text{spatial}}}$. Depending on the presence or absence of ϕ_i, where $i \in \{i_1, \ldots, i_N\}$, in the Slater determinant function $\Phi_{[i_1,\ldots,i_N]}$. The meaning of ξ_i

is defined as follows: if $\xi_i = 0$, the spatial-orbital ϕ_i is unoccupied in the Slater determinant; if $\xi_i = 1$, the spatial-orbital ϕ_i can be occupied with a spin-up, i.e $\uparrow$, in the Slater determinant or occupied with a spin-down, i.e $\downarrow$, in the Slater determinant; if $\xi_i = 2$, the spatial-orbital ϕ_i is doubly occupied with both spin-up and spin-down.

We can define new annihilation and creation operators, which act on Slater determinants spanned over spatial-orbital basis functions $\phi_i, i \in [d_{\text{spatial}}]$. These operators create or remove a particle with a spin value $s \in \left\{\pm\frac{1}{2}\right\}$ at a specific orbital i. We represent the annihilation operator as $a_{i,s}$ and the creation operator as $a^*_{i,s}$. These operators satisfy anti-commutation relations, expressed as follows, for $s, s' \in \left\{\pm\frac{1}{2}\right\}, i, j \in [d_{\text{spatial}}]$:

$$\begin{aligned} a_{i,s}a_{j,s'} + a_{j,s'}a_{i,s} &= 0, \\ a^*_{i,s}a^*_{j,s'} + a^*_{j,s'}a^*_{i,s} &= 0, \\ a_{i,s}a^*_{j,s'} + a^*_{j,s'}a_{i,s} &= \delta_{ij}\delta_{ss'}. \end{aligned} \tag{1.132}$$

In alignment with Remark 1.3.9, we recognize that each spatial-orbital basis $\phi_i,\ i \in [d_{\text{spatial}}]$ can be linked to two spin-orbital basis functions $\varphi_i, i \in [d]$. Within this framework, the Fock space $\boldsymbol{\mathcal{F}}_{d_{\text{spatial}}}$ is isomorphic to $\mathbb{R}^{4^{d_{\text{spatial}}}}$. In this context, for each $\xi_k \in \{0, 1, 2\}$, where $k \in [d_{\text{spatial}}]$, we identify a unit vector $\boldsymbol{e}_{\mu_k} \in \mathbb{R}^4$, $\mu_k \in \{1, 2, 3, 4\}$. Let us define the following mapping function:

$$q_4 : \{1, 2, 3, 4\} \to \{0, 1, 1, 2\}. \tag{1.133}$$

This function serves as a mapping between indices and occupation numbers. This corresponds to labeling four possible occupation states such that in physics or chemistry language 1 or $|-\rangle$ refers to absent (the occupation number is 0) , 2 or $|\uparrow\rangle$ refers to present with spin-up only (the occupation number is 1), 3 or $|\downarrow\rangle$ refers to present with spin-down only (the occupation number is 1), and 4 or $|\uparrow\downarrow\rangle$ refers to present with both spin-up and spin-down (the occupation number is 2). Now, each unit vector $\boldsymbol{e}_{\mu_k},\ k \in [d]$, is defined as:

$$\begin{aligned} \boldsymbol{e}_{\mu_k} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ if } \xi_k = q_4(\mu_k) = 0,\ \ \boldsymbol{e}_{\mu_k} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ if } \xi_k = q_4(\mu_k) = 1, \mu_k = 2, \\ \boldsymbol{e}_{\mu_k} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ if } \xi_k = q_4(\mu_k) = 1, \mu_k = 3, \text{and } \ \boldsymbol{e}_{\mu_k} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \text{ if } \xi_k = q_4(\mu_k) = 2. \end{aligned} \tag{1.134}$$

It follows that each Slater determinant $\Phi_{\xi_1 \dots \xi_{d_{\text{spatial}}}}$ is identified by a unit vector $\boldsymbol{e}_{\mu_1 \dots \mu_{d_{\text{spatial}}}} = \boldsymbol{e}_{\mu_1} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_{d_{\text{spatial}}}} \in \mathbb{R}^{4^{d_{\text{spatial}}}}$. Within this context, the matrix representations of the creation and annihilation operators are given by:

The matrix representation of the annihilation operator, denoted by $\mathbf{A}_{i,s} \in \mathbb{R}^{4^{d\text{spatial}} \times 4^{d_{\text{spatial}}}}$, where $i \in [d_{\text{spatial}}]$ and $s = \pm\frac{1}{2}$, can be expressed as:

$$\begin{aligned} \mathbf{A}_{i,s} &= \left(\bigotimes_{k=1}^{i-1} {}_K \mathbf{Z}\right) \otimes_K \mathbf{A}_s \otimes_K \left(\bigotimes_{k=i+1}^{d_{\text{spatial}}} {}_K \mathbf{I}_4\right) \\ &= \mathbf{Z} \otimes_K \cdots \otimes_K \mathbf{Z} \otimes_K \mathbf{A}_s \otimes_K \mathbf{I}_4 \otimes_K \cdots \otimes_K \mathbf{I}_4. \end{aligned} \tag{1.135}$$

and the matrix representation of the creation operator, which is the transpose of $\mathbf{A}_{i,s}$ where $i \in [d_{\text{spatial}}]$ and $s = \pm\frac{1}{2}$, can be expressed as:

$$\begin{aligned} \mathbf{A}^*_{i,s} &= \left(\bigotimes_{k=1}^{i-1} {}_K \mathbf{Z}\right) \otimes_K \mathbf{A}^*_s \otimes_K \left(\bigotimes_{k=i+1}^{d_{\text{spatial}}} {}_K \mathbf{I}_4\right) \\ &= \mathbf{Z} \otimes_K \cdots \otimes_K \mathbf{Z} \otimes_K \mathbf{A}^*_s \otimes_K \mathbf{I}_4 \otimes_K \cdots \otimes_K \mathbf{I}_4. \end{aligned} \tag{1.136}$$

Here

$$\mathbf{A}_s = \begin{cases} \mathbf{A} \otimes_K \mathbf{I}_2, & \text{if } s = \frac{1}{2}, \\ \mathbf{S} \otimes_K \mathbf{A}, & \text{if } s = -\frac{1}{2}. \end{cases}, \text{ and } \mathbf{Z} = \mathbf{S} \otimes \mathbf{S} \in \mathbb{R}^{4^{d_{\text{spatial}}} \times 4^{d_{\text{spatial}}}}, \tag{1.137}$$

with $\mathbf{A}$ and $\mathbf{S}$ being defined in Equation (1.125). While this formalism, the Hamiltonian matrix $\mathbf{H}_{\text{spatial}} \in \mathbb{R}^{4^{d_{\text{spatial}}} \times 4^{d_{\text{spatial}}}}$ and the matrix representation of the particle number denoted by $\mathbf{N} \in \mathbb{R}^{4^{d_{\text{spatial}}} \times 4^{d_{\text{spatial}}}}$ are:

$$\mathbf{H}_{\text{spatial}} = \sum_{i,j=1}^{d_{\text{spatial}}} \sum_{s=\pm\frac{1}{2}} h_{ij} \mathbf{A}_{i,s}^* \mathbf{A}_{j,s} + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d_{\text{spatial}}} \sum_{s,s'=\pm\frac{1}{2}} v_{ijk\ell} \mathbf{A}_{i,s}^* \mathbf{A}_{k,s'}^* \mathbf{A}_{j,s'} \mathbf{A}_{\ell,s}, \tag{1.138}$$

and

$$\mathbf{N} = \sum_{i=1}^{d_{\text{spatial}}} \sum_{s=\pm\frac{1}{2}} \mathbf{A}_{i,s}^* \mathbf{A}_{i,s}. \tag{1.139}$$

Henceforth, for the sake of simplicity, we continue working in the spin-orbital formulation, with d being the number of spin-orbital basis functions. We designate $n = 2$ and continue to denote the Hamiltonian matrix by $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$.

With increasing number of d, the dimensions of both the Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ and the eigenfunction $\mathbf{\Psi} \in \mathbb{R}^{n^d}$ grow exponentially, as $\mathcal{O}(n^{2d})$ for $\mathbf{H}$ and $\mathcal{O}(n^d)$ for $\mathbf{\Psi}$. To alleviate this growth efficiently, we can employ the TT-format for both $\mathbf{H}$ and $\mathbf{\Psi}$. The TT representation of $\mathbf{\Psi}$ can be defined according to Definition 19 in the following:
let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$, $\mathcal{U} \in \mathbb{R}^{r_{k-1} \times n \times r_k}$, $r_d = r_0 = 1$ be the TT decomposition of the tensor folding of $\mathbf{\Psi} \in \mathbb{R}^{n^d}$ denoted by $\psi \in \mathbb{R}^{n \times \cdots \times n}$and let $\mathbf{U}_k[\mu_k] := \mathcal{U}_k[:, \mu_k, :] \in \mathbb{R}^{r_{k-1} \times r_k}$, for a fixed value of $\mu_k \in [n]$, for each $k \in [d]$. The $(\mu_1, \cdots, \mu_d)$-th element of the tensor ψ is given as:

$$\psi(\mu_1; \ldots; \mu_d) = \mathbf{U}_1[\mu_1] \cdots \mathbf{U}_d[\mu_d]. \tag{1.140}$$

In the following, we refer to the TT representation of the Hamiltonian operator as TTO, also known as Matrix Product Operator (MPO) in the physics community [2, 48, 106, 131]. Let $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$, its TTO representation can be described by the contraction product of 3-order and 4-order TT-cores, denoted by $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$, where $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times n \times n \times R_k}$ $(R_0 = R_d = 1)$, where R_k are referred to as the TTO-ranks. The TTO decomposition is:

$$\mathcal{H} = \mathcal{H}_1 \times_4 \mathcal{H}_2 \times_4 \ldots \times_4 \mathcal{H}_{d-1} \times_4 \mathcal{H}_d. \tag{1.141}$$

By defining $\mathbf{H}_k[\mu_k, \nu_k] := \mathcal{H}_k[:, \mu_k, \nu_k, :]$, $\mu_k, \nu_k \in [n]$, $k \in [d]$, the $(\mu_1, \ldots, \mu_d, \nu_1, \ldots, \nu_d) - th$ element of $\mathcal{H}$ satisfies:

$$\mathcal{H}(\mu_1; \ldots; \mu_d; \nu_1; \ldots; \nu_d) = \mathbf{H}_1[\mu_1, \nu_1] \ldots \mathbf{H}_d[\mu_d, \nu_d]. \tag{1.142}$$

A graphical description of the TTO is given in the following Figure 1.10.

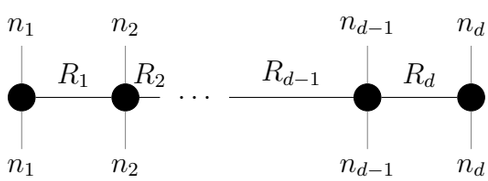


Figure 1.10: Graphical representation of TTO with $n_1 = \ldots = n_d = n$.

Remark 1.3.10. (TTO representation in physics literature) In the physics literature, we often encounter an alternative decomposition of $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$, expressed as:

$$\mathbf{H} = \sum_{\beta_0=1}^{R_0} \sum_{\beta_1=1}^{R_1} \ldots \sum_{\beta_{d-1}}^{R_{d-1}} \sum_{\beta_d=1}^{R_d} \mathbf{H}'_1(\beta_0; \beta_1) \otimes_K \ldots \otimes_K \mathbf{H}'_d(\beta_{d-1}; \beta_d), \tag{1.143}$$

where we define $\mathbf{H}'_k(\beta_{k-1}; \beta_k) := \mathcal{H}_k[\beta_{k-1}, :, :, \beta_k] \in \mathbb{R}^{n \times n}$, $\beta_k \in [R_k]$, $k \in [n]$.

Remark 1.3.11. (Representation with strong Kronecker product) We also encounter the following representation of the TTO:

$$\mathbf{H} = \mathbf{H}_1^{<2>} \bowtie \mathbf{H}_2^{<2>} \bowtie \ldots \bowtie \mathbf{H}_d^{<2>}, \tag{1.144}$$

where $\mathbf{H}_k^{<2>} \in \mathbb{R}^{R_{k-1}n \times nR_k}, k \in [d]$ is the mode-$(1:2)$ matricization of $\mathcal{H}_k$ and $\bowtie$ refers to the strong Kronecker product, (see [2, 73]). For example, for two block matrices $\mathbf{X}, \mathbf{Y}$ we have

$$\begin{aligned}&\begin{bmatrix} \mathbf{X}_{1,1} & \mathbf{X}_{1,2} \\ \mathbf{X}_{2,1} & \mathbf{X}_{2,2} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{Y}_{1,1} & \mathbf{Y}_{1,2} \\ \mathbf{Y}_{2,1} & \mathbf{Y}_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}_{1,1} \otimes_K \mathbf{Y}_{1,1} + \mathbf{X}_{1,2} \otimes_K \mathbf{Y}_{2,1} & \mathbf{X}_{1,1} \otimes_K \mathbf{Y}_{1,2} + \mathbf{X}_{1,2} \otimes_K \mathbf{Y}_{2,2} \\ \mathbf{X}_{2,1} \otimes_K \mathbf{Y}_{1,1} + \mathbf{X}_{2,2} \otimes_K \mathbf{Y}_{2,1} & \mathbf{X}_{2,1} \otimes_K \mathbf{Y}_{1,2} + \mathbf{X}_{2,2} \otimes_K \mathbf{Y}_{2,2} \end{bmatrix},\end{aligned}$$

where $\mathbf{X}_{i,j}, \mathbf{Y}_{i,j}$ represent dense blocks for $i, j \in \{1, 2\}$.

Expressing the operator in TTO format enables to compute smartly basic algebraic operations, such as the matrix-vector product and the matrix-matrix product as explained in the following propositions (for a general case of operators).

Proposition 1.3.1. *(Matrix-vector product with TT decompositions) Let a vector $\boldsymbol{u} \in \mathbb{R}^{n_1 \dots n_d}$ and a matrix $\mathbf{H} \in \mathbb{R}^{(m_1 \dots m_d) \times (n_1 \dots n_d)}$ be represented by TT decompositions. Let $\mathcal{U} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ be the tensor folding of $\boldsymbol{u}$ with TT decomposition $(\mathcal{U}_1, \dots, \mathcal{U}_d)$, with $\mathcal{U}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, $r_0 = r_d = 1$ and let $\mathcal{H} \in \mathbb{R}^{m_1 \times \dots \times m_d \times n_1 \times \dots \times n_d}$ be the tensor folding of $\mathbf{H}$ with TTO decomposition $(\mathcal{H}_1, \dots, \mathcal{H}_d)$, with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times m_k \times n_k \times R_k}$, $R_0 = R_d = 1$. Let $(\mathcal{T}_1, \dots, \mathcal{T}_d)$ be the TT decomposition of the resulting matrix-vector product $\mathbf{H}\boldsymbol{u}$. The latter can be represented as:*

$$\begin{aligned}(\mathbf{H}\boldsymbol{u})(\overline{\mu_1, \dots, \mu_d}) &= \sum_{\nu_1=1}^{n_1} \dots \sum_{\nu_d=1}^{n_d} \mathcal{H}(\mu_1; \dots; \mu_d; \nu_1; \dots; \nu_d) \mathcal{U}(\nu_1; \dots; \nu_d) \\ &= \sum_{\nu_1=1}^{n_1} \dots \sum_{\nu_d=1}^{n_d} \left(\mathbf{H}_1[\mu_1, \nu_1] \dots \mathbf{H}_d[\mu_d, \nu_d]\right) \left(\mathbf{U}_1[\nu_1]\right) \dots \mathbf{U}_d[\nu_d])\end{aligned} \tag{1.145}$$

where $\mathbf{H}_k[\mu_d, \nu_d] := \mathcal{H}_k[:, \mu_k, \nu_k, :]$ and $\mathbf{U}_k[\nu_k] := \mathcal{U}_k[:, \nu_k, :], \mu_k \in [m_k], \nu_k \in [n_k], k \in [d]$. By using the Kronecker product property defined in (1.10)*, we have:*

$$(\mathbf{H}\boldsymbol{u})(\overline{\mu_1, \dots, \mu_d}) = \underbrace{\left(\sum_{\nu_1=1}^{n_1} (\mathbf{H}_1[\mu_1, \nu_1] \otimes_K \mathbf{U}_1[\nu_1])\right)}_{\mathbf{T}_1[\mu_1] \in \mathbb{R}^{r_0 R_0 \times r_1 R_1}} \dots \underbrace{\left(\sum_{\nu_d=1}^{n_d} (\mathbf{H}_d[\mu_d, \nu_d] \otimes_K \mathbf{U}_d[\nu_d])\right)}_{\mathbf{T}_d[\mu_d] \in \mathbb{R}^{r_{d-1} R_{d-1} \times r_d R_d}}, \tag{1.146}$$

where $\mathbf{T}_k[\mu_k] := \mathcal{T}_k[:, \mu_k, :], \mu_k \in [m_k], k \in [d]$.

Proposition 1.3.2. *(Matrix-Matrix product with TT decomposition) Let a matrix $\mathbf{H} \in \mathbb{R}^{(m_1 \dots m_d) \times (n_1 \dots n_d)}$ and a matrix $\widetilde{\mathbf{H}} \in \mathbb{R}^{(n_1 \dots n_d) \times (m'_1 \dots m'_d)}$ be represented by TTO decompositions. Let $\mathcal{H} \in \mathbb{R}^{m_1 \times \dots \times m_d \times n_1 \times \dots \times n_d}$ be the tensor folding of $\mathbf{H}$ with TTO decomposition $(\mathcal{H}_1, \dots, \mathcal{H}_d)$, with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times m_k \times n_k \times R_k}$, $R_0 = R_d = 1$ and let $\widetilde{\mathbf{H}} \in \mathbb{R}^{n_1 \times \dots \times n_d \times m'_1 \times \dots \times m'_d}$ be the tensor folding of $\widetilde{\mathbf{H}}$ with TTO decomposition $(\widetilde{\mathcal{H}}_1, \dots, \widetilde{\mathcal{H}}_d)$, with $\widetilde{\mathcal{H}}_k \in \mathbb{R}^{R'_{k-1} \times n_k \times m'_k \times R'_k}$, $R'_0 = R'_d = 1$. Let $(\mathcal{T}_1, \dots, \mathcal{T}_d)$ be the TT decomposition of the resulting matrix-matrix product $\mathbf{H}\widetilde{\mathbf{H}}$. The latter can be represented as:*

$$\begin{aligned}(\mathbf{H}\widetilde{\mathbf{H}})(\overline{\mu_1, \dots, \mu_d}; \overline{\nu_1, \dots, \nu_d}) &= \sum_{z_1=1}^{n_1} \dots \sum_{z_d=1}^{n_d} \mathcal{H}(\mu_1; \dots; \mu_d; z_1; \dots; z_d) \mathcal{H}(z_1; \dots; z_d; \nu_1; \dots; \nu_d) \\ &= \sum_{z_1=1}^{n_1} \dots \sum_{z_d=1}^{n_d} \left(\mathbf{H}_1[\mu_1, z_1] \dots \mathbf{H}_d[\mu_d, z_d]\right) \left(\widetilde{\mathbf{H}}_1[z_1, \nu_1] \dots \widetilde{\mathbf{H}}_d[z_d, \nu_d]\right),\end{aligned} \tag{1.147}$$

where $\mathbf{H}_k[\mu_d, \nu_d] := \mathcal{H}_k[:, \mu_k, \nu_k, :]$ and $\widetilde{\mathbf{H}}_k[\mu_d, \nu_d] := \widetilde{\mathcal{H}}_k[:, \mu_k, \nu_k, :]$, $\mu_k \in [m_k]$, $\nu_k \in [m'_k]$, $k \in [d]$. By using the Kronecker product property defined in (1.10)*, we have:*

$$(\mathbf{H}\widetilde{\mathbf{H}})(\overline{\mu_1, \dots, \mu_d}; \overline{\nu_1, \dots, \nu_d}) = \underbrace{\left(\sum_{z_1=1}^{n_1} (\mathbf{H}_1[\mu_1, z_1] \otimes_K \widetilde{\mathbf{H}}_1[z_1, \nu_1])\right)}_{\mathbf{T}_1[\mu_1, \nu_1] \in \mathbb{R}^{R'_0 R_0 \times R'_1 R_1}} \dots \underbrace{\left(\sum_{z_d=1}^{n_d} (\mathbf{H}_d[\mu_d, z_d] \otimes_K \widetilde{\mathbf{H}}_d[z_d, \nu_d])\right)}_{\mathbf{T}_d[\mu_d, \nu_d] \in \mathbb{R}^{R'_{d-1} R_{d-1} \times R'_d R_d}}, \tag{1.148}$$

where $\mathbf{T}_k[\mu_k, \nu_k] := \mathcal{T}_k[:, \mu_k, \nu_k, :], \mu_k \in [m_k], \nu_k \in [m'_k], k \in [d]$.

Constructing the TTO is at the core of the QC-DMRG algorithm. Several different approaches have been proposed for the construction of an efficient TTO representation of the quantum chemical Hamiltonian operator, see Equation (1.138). These approaches yield a compact TTO representation with TTO-ranks that grow quadratically with the system size as $\mathcal{O}(d^2)$, see [2, 17]. This will be reviewed and explained in Chapter 3.

DMRG algorithm

Back to Equation (1.61), let d be the size of the system, i.e number of spin-orbitals basis functions, let $n = 2$, let $\psi \in \mathbb{R}^{n\times\ldots\times n}$ be the d-order tensor defined as the tensor folding of the eigenfunction $\mathbf{\Psi}$. Assume that ψ has a TT decomposition with at most TT-rank equal to $\boldsymbol{r} = (r_1, \ldots, r_d)$, let $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$ be the Hamiltonian symmetric matrix. Then the minimization problem is:

$$\begin{aligned}\lambda_{\min} &= \min_{\psi\in M_{\mathrm{TT}\leq\boldsymbol{r}},\psi\neq 0} R_{\mathbf{H}}(\mathrm{vec}(\psi)),\\ &= \min_{\psi\in M_{\mathrm{TT}\leq\boldsymbol{r}},\psi\neq 0} \frac{1}{2}\frac{\langle \mathbf{H}\,\mathrm{vec}(\psi), \mathrm{vec}(\psi)\rangle}{\langle \mathrm{vec}(\psi), \mathrm{vec}(\psi)\rangle},\end{aligned} \tag{1.149}$$

with $\mathrm{vec}(\psi)$ being the vectorization of the tensor ψ, see Definition 10, and $M_{\mathrm{TT}\leq\boldsymbol{r}}$ being the set of TTs with TT-ranks less or equal to $\boldsymbol{r} = (r_1, \ldots, r_d)$, as introduced in Proposition 1.2.3. The key idea behind DMRG is the alternating optimization of Rayleigh quotient over the TT manifold defined in Proposition 1.2.3.

Let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ be the TT decomposition of ψ, where $\mathcal{U}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$, $r_0 = r_d = 1$. Let $\hat{P}_k : \mathbb{R}^{r_{k-1}nr_k} \to \mathbb{R}^{n^d}$, $k \in [d]$ be the linear operator with the property that for all $\mathcal{X}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$, the element-wise expression of $\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))$ is given by:

$$\left(\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))\right)(\overline{\mu_1, \ldots, \mu_d}) = \mathbf{U}_1[\mu_1]\ldots\mathbf{U}_{k-1}[\mu_{k-1}]\mathbf{X}_k[\mu_k]\mathbf{U}_{k+1}[\mu_{k+1}]\ldots\mathbf{U}_d[\mu_d], \tag{1.150}$$

where $\mathbf{X}_k[\mu_k] = \mathcal{X}_k[:, \mu_k, :]$ for $k \in [d]$.

In each step of the DMRG algorithm, as elaborated in [48, 106], the reduced minimization problem that is tackled involves minimizing the following Rayleigh quotient:

$$R_{\mathbf{H}}\left(\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))\right) = \frac{1}{2}\frac{\left\langle \hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k)), \mathbf{H}\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))\right\rangle}{\left\langle \hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k)), \hat{P}_k\,\mathrm{vec}\,(\mathcal{X}_k))\right\rangle}. \tag{1.151}$$

Definition 23. Let $\hat{P}_k : \mathbb{R}^{r_{k-1}nr_k} \to \mathbb{R}^{n^d}$, $k \in [d]$ be the linear operator with the property defined in Equation (1.150), the matrix representation of $\hat{P}_k$ denoted as $\mathbf{P}_k \in \mathbb{R}^{(n_1\ldots n_d)\times(r_{k-1}nr_k)}$ is given by:

$$\mathbf{P}_k = \mathbf{L}_{k-1}^{<k-1>} \otimes_K \mathbf{I}_n \otimes_K \left(\mathbf{R}_{k+1}^{(1)}\right)^* \in \mathbb{R}^{n^d\times(r_{k-1}nr_k)}, \tag{1.152}$$

where

$$\mathcal{L}_{k-1} = \mathcal{U}_1 \times_3 \ldots \times_3 \mathcal{U}_{k-1} \in \mathbb{R}^{\underbrace{n\times\ldots\times n}_{k-1\,\text{terms}}\times r_{k-1}}, \tag{1.153}$$

and

$$\mathcal{R}_{k+1} = \mathcal{U}_{k+1} \times_3 \ldots \times_3 \mathcal{U}_d \in \mathbb{R}^{r_k\times\underbrace{n\times\ldots\times n}_{d-k\,\text{terms}}}. \tag{1.154}$$

where $\mathbf{L}^{<k-1>}$ is the mode-$(1 : k-1)$ matricization of $\mathcal{L}_{k-1}$ and $\mathbf{R}^{(1)}$ is the mode-1 matricization of $\mathcal{R}_{k+1}$. According to the Definition 23, it can readily checked that for $\mathcal{X}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$:

$$\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k)) = \mathbf{P}_k\,\mathrm{vec}\,(\mathcal{X}_k)\,. \tag{1.155}$$

Assuming that the TT-cores $(\mathcal{U}_1, \ldots, \mathcal{U}_{k-1})$ and $(\mathcal{U}_{k+1}, \ldots, \mathcal{U}_d)$ are left-orthogonal and right-orthogonal, respectively, then $\mathbf{P}_k$ is an orthogonal matrix. Therefore, the solution of the minimization problem (1.149) is the solution of the first order condition given by (see [48]):

$$\begin{aligned}\nabla R_{\mathbf{H}}\left(\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))\right) &= \frac{1}{\|\,\mathrm{vec}\,(\mathcal{X}_k)\,\|}\left(\mathbf{P}_k^*\mathbf{H}\mathbf{P}_k\,\mathrm{vec}\,(\mathcal{X}_k) - \frac{\langle \mathbf{P}_k^*\mathbf{H}\mathbf{P}_k\,\mathrm{vec}\,(\mathcal{X}_k)\,, \mathrm{vec}\,(\mathcal{X}_k)\rangle}{\langle \mathrm{vec}\,(\mathcal{X}_k)\,, \mathrm{vec}\,(\mathcal{X}_k)\rangle}\mathbf{P}_k^*\mathbf{P}_k\,\mathrm{vec}\,(\mathcal{X}_k)\right)\\ &= \frac{1}{\|\,\mathrm{vec}\,(\mathcal{X}_k)\,\|}\left(\mathbf{M}_k\,\mathrm{vec}\,(\mathcal{X}_k) - \lambda_k\,\mathrm{vec}\,(\mathcal{X}_k)\right)\\ &= \mathbf{0}, \quad \mathrm{vec}\,(\mathcal{X}_k) \neq \mathbf{0},\end{aligned} \tag{1.156}$$

with $\nabla R_{\mathbf{H}}\left(\hat{P}_k(\mathrm{vec}\,(\mathcal{X}_k))\right)$ being the gradient of $R_{\mathbf{H}}\left(\hat{P}_k(.)\right)$ with respect to $\mathrm{vec}\,(\mathcal{X}_k)$, and $\mathbf{M}_k = \mathbf{P}_k^*\mathbf{H}\mathbf{P}_k$. The matrix $\mathbf{M}_k$, also called the reduced matrix, is commonly known as the *effective Hamiltonian*. It is symmetric, as outlined in the following proposition:

Proposition 1.3.3. *The reduced matrices* $\mathbf{M}_k = \mathbf{P}_k^*\mathbf{H}\mathbf{P}_k \in \mathbb{R}^{(r_{k-1}nr_k)\times(r_{k-1}nr_k)}$ *are square and symmetric matrices.*

Proof. The proof is based on the symmetry of the matrix $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$. Indeed, we have:

$$\mathbf{M}_k^* = \mathbf{P}_k^*\mathbf{H}^*\mathbf{P}_k = \mathbf{P}_k^*\mathbf{H}\mathbf{P}_k = \mathbf{M}_k. \tag{1.157}$$

□

Remark 1.3.12. It is worth noting that if the TT-cores are not left-orthogonal or right-orthogonal, the expression in (1.156) leads to a generalized eigenvalue problem.

By defining $\boldsymbol{u} = \mathrm{vec}(\mathcal{X}_k)$, the reduced eigenvalue problem, for each $k \in [d]$, is given by:

$$\mathbf{M}_k\boldsymbol{u} = \lambda_k\boldsymbol{u}, \tag{1.158}$$

We now have all the necessary ingredients to state the DMRG algorithm in **Algorithm** 3. In **Algorithm** 3, each iteration within the main loop, corresponds to a half-sweep, and the eigenvalue problem solved during each half-sweep is termed as a *microstep*. Completing two consecutive loops is what we refer to as a *sweep*.

Algorithm 3 1-site DMRG (ALS)

Input:

- Initial guess of TT-cores $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$.
- Right-orthogonalized TT-cores $(\mathcal{U}_2, \ldots, \mathcal{U}_d)$ (see Equation (1.47)).
- TTO decomposition $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ of the Hamiltonian operator $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$, see Equation (1.141).

Output: λ and a TT decomposition $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$, such that the resulting vector $\boldsymbol{\Psi} \in \mathbb{R}^{n^d}$ from this TT, see Equation (1.140), is the eigenfunction of the lowest eigenvalue λ.

procedure DMRG
  while not converged **do**
    for $k = 1$ to $d-1$ **do** ▷ Forward half-sweep
      $\boldsymbol{u} = \mathrm{vec}\,(\mathcal{U}_k)$, $\mathbf{M}_k = \mathbf{P}_k^*\mathbf{H}\mathbf{P}_k$
      Solve $\mathbf{M}_k\boldsymbol{u} = \lambda_k\boldsymbol{u}$, ▷ see Equation (1.156)
      $\lambda = \lambda_k$
      $\mathbf{U} = \mathrm{Reshape}(\boldsymbol{u}, (r_{k-1}n, r_k))$,
      $\mathbf{Q}, \mathbf{R} = \mathrm{QR}(\mathbf{U})$, ▷ QR decomposition.
      $\mathcal{U}_k = \mathrm{Reshape}(\mathbf{Q}, (r_{k-1}, n, r_k))$, ▷ $\mathbf{Q} \in \mathbb{R}^{r_{k-1}n\times r_k}$, $\mathbf{R} \in \mathbb{R}^{r_k\times r_k}$.
      $\mathcal{U}_{k+1} = \mathbf{R} \times_2 \mathcal{U}_{k+1}$.
    end for
    for $k = d$ to 2 **do** ▷ Backward half-sweep
      $\boldsymbol{u} = \mathrm{vec}\,(\mathcal{U}_k)$,
      Solve $\mathbf{M}_k\boldsymbol{u} = \lambda_k\boldsymbol{u}$,
      $\lambda = \lambda_k$,
      $\mathbf{U} = \mathrm{Reshape}(\boldsymbol{u}, (r_{k-1}, nr_k))$,
      $\mathbf{L}, \mathbf{Q} = \mathrm{LQ}(\mathbf{U})$, ▷ LQ decomposition.
      $\mathcal{U}_k = \mathrm{Reshape}(\mathbf{Q}, (r_{k-1}, n, r_k))$, ▷ $\mathbf{Q} \in \mathbb{R}^{r_{k-1}\times nr_k}$, $\mathbf{L} \in \mathbb{R}^{r_{k-1}\times r_{k-1}}$.
      $\mathcal{U}_{k-1} = \mathcal{U}_{k-1} \times_3 \mathbf{L}$.
    end for
  end while
end procedure

Since the 1-site DMRG involves optimizing over a single fixed-rank TT-core during each microstep, its main limitation is the lack of rank adaptivity: 1-site DMRG typically

employs fixed ranks during the optimization process. Therefore the initial guess of TT-ranks must be well chosen to have a satisfactory convergence. It is however true that one can increase the ranks by introducing random orthogonal vectors to $\mathcal{U}_k$. This can impact the algorithm convergence. An alternative approach is the 2-site DMRG, also known as Modified ALS (MALS), where the optimization in each microstep occurs over two consecutive TT-cores instead of one. This enables a certain rank adaptivity between these two components.

In the framework of 2-site DMRG, an alternative linear operator is given in the following definition.

Definition 24. Let $\mathcal{X}_{k,k+1} \in \mathbb{R}^{r_{k-1}\times n\times n\times r_{k+1}}$ be a 4-order tensor defined as:

$$\mathcal{X}_{k,k+1} = \mathcal{X}_k \times_3 \mathcal{X}_{k+1},\ k \in [d-1]\,, \tag{1.159}$$

where $\mathcal{X}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$ and $\mathcal{X}_{k+1} \in \mathbb{R}^{r_k\times n\times r_{k+1}}$. Let $\hat{P}_{k,k+1}$ be a linear operator, operating within vector spaces according to:

$$\hat{P}_{k,k+1} : \mathbb{R}^{r_{k-1}n^2r_{k+1}} \to \mathbb{R}^{n^d},\quad k \in [d-1]\,, \tag{1.160}$$

with the property that for all $\mathcal{X}_{k,k+1} \in \mathbb{R}^{r_{k-1}\times n\times n\times r_{k+1}}$, the element-wise expression of $\hat{P}_{k,k+1}(\mathrm{vec}\,(\mathcal{X}_{k,k+1}))$ is given by:

$$\left(\hat{P}_{k,k+1}(\mathrm{vec}\,(\mathcal{X}_{k,k+1}))\right)(\overline{\mu_1,\ldots,\mu_d}) = \mathbf{U}_1[\mu_1]\ldots\mathbf{U}_{k-1}[\mu_{k-1}]\mathbf{X}_k[\mu_k]\mathbf{X}_{k+1}[\mu_{k+1}]\mathbf{U}_{k+1}[\mu_{k+1}]\ldots\mathbf{U}_d[\mu_d], \tag{1.161}$$

where $\mathbf{X}_k[\mu_k] = \mathcal{X}_k[:,\mu_k,:]$, and $\mathbf{X}_{k+1}[\mu_{k+1}] = \mathcal{X}_{k+1}[:,\mu_{k+1},:]$, for $k \in [d]$. The matrix representation of $\hat{P}_{k,k+1}$ denoted as $\mathbf{P}_{k,k+1}$ is given by:

$$\mathbf{P}_{k,k+1} = \mathbf{L}_{k-1}^{<k-1>} \otimes_K \mathbf{I}_n \otimes \mathbf{I}_n \otimes_K (\mathbf{R}_{k+2}^{<1>})^*, \tag{1.162}$$

where $\mathbf{L}_{k-1}^{<k-1>}$ is the mode-$(1:k-1)$ matricization of $\mathcal{L}_{k-1}$ defined in Equation (1.153) and $\mathbf{R}_{k+2}^{<1>}$ is the mode-1 matricization of $\mathcal{R}_{k+2}$ defined as:

$$\mathcal{R}_{k+2} = \mathcal{U}_{k+2} \times_3 \ldots \times_3 \mathcal{U}_d \in \mathbb{R}^{r_{k+1}\times\underbrace{n\times\ldots\times n}_{d-k-1\,\text{terms}}}. \tag{1.163}$$

In 2-site DMRG, each microstep involves solving, during the first half-sweep, the following:

$$\mathbf{M}_{k,k+1}\,\mathrm{vec}\,(\mathcal{X}_{k,k+1}) = \mathbf{P}_{k,k+1}^*\mathbf{H}\mathbf{P}_{k,k+1}\,\mathrm{vec}\,(\mathcal{X}_{k,k+1}) = \lambda_{k,k+1}\,\mathrm{vec}\,(\mathcal{X}_{k,k+1})\,,\quad k \in [d-2]\,, \tag{1.164}$$

and likewise during the second half-sweep:

$$\mathbf{M}_{k-1,k}\,\mathrm{vec}\,(\mathcal{X}_{k,k+1}) = \mathbf{P}_{k-1,k}^*\mathbf{H}\mathbf{P}_{k-1,k}\,\mathrm{vec}\,(\mathcal{X}_{k,k+1})_{k,k+1} = \lambda_{k,k+1}vec\,(\mathcal{X}_{k,k+1})_{k,k+1}\,,\quad k \in \{3,\ldots,d\}\,. \tag{1.165}$$

At each microstep, after solving the eigenvalue problem, an SVD is performed on the mode-$(1:2)$ matricization of $\mathcal{X}_{k,k+1}$ denoted as $\mathbf{X}_{k,k+1}^{<2>}$ to update the components $\mathcal{U}_k = \mathcal{X}_k$ and $\mathcal{U}_{k+1} = \mathcal{X}_{k+1}$ and to recover the left (resp. right) orthogonality.

Note that employing a TTO decomposition of the Hamiltonian operator during the initialization step in **Algorithm** 3 is essential for efficiently computing $\mathbf{H}_k\boldsymbol{u}$, as elucidated in the subsequent explanation.

In what follows, we give some details about DMRG implementation using tensor diagrams. Additionally, we provide a table summarizing the different complexities involved in various computational steps in 1-site (resp. 2-site) DMRG.

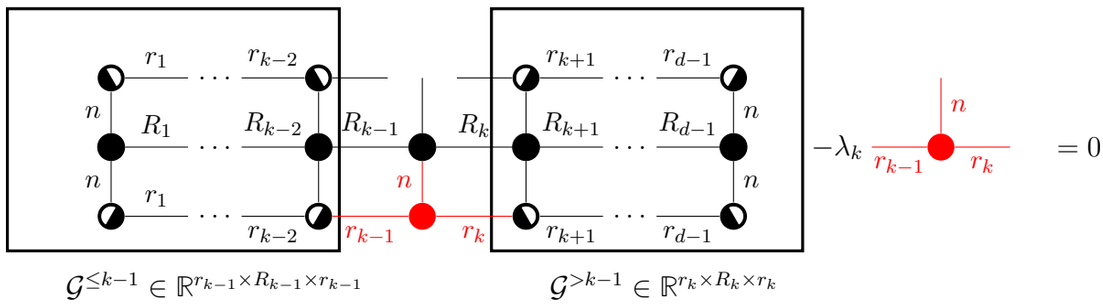


Figure 1.11: Graphical representation of the minimization problem at each microstep: the left most part corresponds to $\mathbf{M}_k$ $\text{vec}(\mathcal{X}_k)$=$\mathbf{P}_k^* \mathbf{H} \mathbf{P}_k \text{vec}(\mathcal{X}_k)$, for $k \in [d]$, r_k are the TT-ranks of the TT representation of the eigenfunction $\mathbf{\Psi}$, R_k are the TTO-ranks of the TTO representation of the Hamiltonian operator and $n = 2$.

.

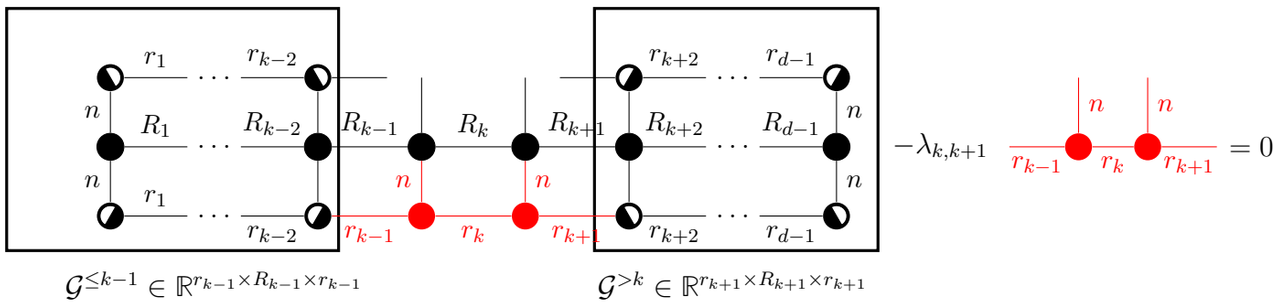


Figure 1.12: Graphical representation of the minimization problem at each microstep: the left most part corresponds to $\mathbf{M}_{k,k+1}$ $\text{vec}(\mathcal{X}_{k,k+1})$=$\mathbf{P}_{k,k+1}^* \mathbf{H} \mathbf{P}_{k,k+1} \text{vec}(\mathcal{X}_{k,k+1})$, for $k \in [d]$, r_k are the TT-ranks of the TT representation of the eigenfunction $\mathbf{\Psi}$, R_k are the TTO-ranks of the TTO representation of the Hamiltonian operator and $n = 2$.

The expressions of tensors of the left and right components denoted respectively as $\mathcal{G}^{\leq k-1}$, $\mathcal{G}^{>k-1}$ (resp. $\mathcal{G}^{>k}$) for the 1-site (resp. 2-site) DMRG, in terms of the TT-cores of the TT decomposition of the eigenfunction $\mathbf{\Psi}$ as well as the Hamiltonian operator, are given in the following definition

Definition 25. Let $(\mathcal{U}_1, \ldots \mathcal{U}_d)$ be the TT decomposition of the eigenfunction $\mathbf{\Psi}$ and let $(\mathcal{H}_1, \ldots \mathcal{H}_d)$ be the TTO decomposition of the Hamiltonian operator. Let $\mathbf{Z}_k \in \mathbb{R}^{r_{k-1}^2 R_{k-1} \times r_k^2 R_k}$, $k \in [d]$ be matrices defined as follows:

$$\mathbf{Z}_k = \sum_{\mu_k=1}^{n} \sum_{\nu_k=1}^{n} \mathbf{U}_k[\mu_k] \otimes_K \mathbf{H}_k[\mu_k, \nu_k] \otimes_K \mathbf{U}_k[\nu_k]. \tag{1.166}$$

The definition of tensors $\mathcal{G}^{\leq \ell}$, for $\ell \in [d-1]$ (resp. $\ell \in [d-2]$) for the 1-site (resp. 2-site) DMRG, is given as follows:

$$\text{vec}(\mathcal{G}^{\leq \ell}) = \mathbf{Z}_1 \mathbf{Z}_2 \ldots \mathbf{Z}_\ell, \tag{1.167}$$

where $\text{vec}(\mathcal{G}^{\leq \ell}) \in \mathbb{R}^{r_\ell^2 R_\ell}$ is the vectorization of $\mathcal{G}^{\leq \ell}$. The definition of tensors $\mathcal{G}^{>\ell}$, for $\ell \in \{2, \ldots, d\}$ (resp. $\ell \in \{3, \ldots, d\}$) for the 1-site (resp. 2-site) DMRG, is given as follows:

$$\text{vec}(\mathcal{G}^{>\ell}) = \mathbf{Z}_{\ell+1} \ldots \mathbf{Z}_{d-1} \mathbf{Z}_d, \tag{1.168}$$

where $\text{vec}(\mathcal{G}^{>\ell}) \in \mathbb{R}^{r_{\ell+1}^2 R_{\ell+1}}$ is the vectorization of $\mathcal{G}^{>\ell}$.

In terms of the left and right components, one can derive the expression of the reduced matrices, $\mathbf{M}_k = \mathbf{P}_k^* \mathbf{H} \mathbf{P}_k \in \mathbb{R}^{r_{k-1}^2 n \times n r_k^2}$ for the 1-site DMRG, for $k \in [d]$. The tensor folding of $\mathbf{M}_k$, denoted as $\mathcal{M}_k \in \mathbb{R}^{r_{k-1} \times n \times r_{k-1} \times r_k \times n \times r_k}$, is given by:

$$\mathcal{M}_k = \mathcal{G}^{\leq k-1} \times_2 \mathcal{H}_k \times_3 \mathcal{G}^{>k-1}. \tag{1.169}$$

It is possible to derive the expression of the reduced matrix, $\mathbf{M}_{k,k+1} = \mathbf{P}^*_{k,k+1}\mathbf{H}\mathbf{P}_{k,k+1} \in \mathbb{R}^{r^2_{k-1}n^2 \times r^2_k n^2}$ for the 2-site DMRG, for $k \in [d-1]$. The tensor folding of $\mathbf{M}_{k,k+1}$, denoted by $\mathcal{M}_{k,k+1} \in \mathbb{R}^{r_{k-1}\times n\times n\times r_{k-1}\times r_{k+1}\times n\times n\times r_{k+1}}$, is given by:

$$\mathcal{M}_{k,k+1} = \mathcal{G}^{\leq k-1} \times_2 (\mathcal{H}_k \times_4 \mathcal{H}_{k+1}) \times_3 \mathcal{G}^{>k}. \tag{1.170}$$

As can be seen from the tensor diagrams and Equations (1.169), (1.170), the DMRG algorithm entails many tensor contraction products between the TT representation of $\mathbf{\Psi}$ and the TTO decomposition of the Hamiltonian operator. In what follows, in Table 1.2, we provide insights into the computational complexities of the steps in the DMRG algorithm.

Computational steps	1-site/ALS	2-site/MALS
Building $\mathcal{G}^{\leq k-1}$ and $\mathcal{G}^{>k-1}$ (resp. $\mathcal{G}^{>k}$)	$\mathcal{O}(nr^3R + n^2r^2R^2)$	$\mathcal{O}(nr^3R + n^2r^2R^2)$
Building $\mathbf{M}_k \operatorname{vec}(\mathcal{X}_k)$ resp. $\mathbf{M}_{k,k+1} \operatorname{vec}(\mathcal{X}_{k,k+1})$	$\mathcal{O}(nr^3R + n^2r^2R^2)$	$\mathcal{O}(nr^3R + n^2r^2R^2 + n^4R^3)$
QR/SVD	$\mathcal{O}(nr^3)$	$\mathcal{O}(n^3r^3)$

Table 1.2: Time-complexities of some computational steps of each iteration k of DMRG.

Here, the ranks r and R are defined as follows: $r = \max\limits_{1\leq k\leq d}\{r_k\}, R = \max\limits_{1\leq k\leq d}\{R_k\}$. Note that in the site DMRG, additional terms are included in the complexity $\mathcal{O}(n^4R^3 + n^2r^3)$, which arises from the contraction product performed between two consecutive TT-cores when constructing $\mathbf{M}_{k,k+1}\mathcal{X}^{<4>}$ for the 2-site DMRG. In Table 1.2, when the TT-ranks r are large, the second step involving several matrix-vector multiplications becomes the most dominant step.

Remark 1.3.13. Several key factors impact the efficiency and performance of the DMRG algorithm. In order to solve the reduced eigenvalue problem, as defined in Equation (1.158), one can use an iterative eigensolver like for example the Lanczos algorithm. This iterative algorithm includes matrix-vector multiplications of the form $\mathbf{M}_k \operatorname{vec}(\mathcal{X}_k)$ or $\mathbf{M}_{k,k+1} \operatorname{vec}(\mathcal{X}_{k,k+1})$. These operations stand out as the most time-intensive component, as highlighted in Table 1.2. To improve computational speed and mitigate the time complexities, reducing the TT-ranks without compromising the algorithm's convergence requires optimizing both the efficient TT representation of $\mathbf{\Psi}$ and the operator. This point is addressed in Chapter 3, where we focus on a popular physics and chemistry approach, which revolves around the use of low-rank approximation and various conservation laws.

Regarding the algorithm's convergence, a well-chosen initial $\mathbf{\Psi}$ input greatly accelerates convergence and prevents convergence issues linked to local minima. Lastly, the sequence in which spin-orbital basis $\varphi_i, i \in [d]$, as defined in (1.70) are ordered can have an influence on the algorithm's convergence behavior, underlining the importance of this factor in achieving optimal results. Yet, it is noteworthy that, as of today, there is no well-established theory for the global convergence of DMRG method. Despite this, numerical experiments consistently show convergence within a finite number of iterations or sweeps, which are not necessarily dependent on the system size. While a comprehensive theory for global convergence is lacking, there are some theoretical insights into the local convergence of the algorithm where this analysis assumes the invertibility of the Hessian of the functional $R_{\mathbf{H}}$ at a critical point, i.e the solution of (1.156) when the TT-ranks are estimated correctly (see [99] for further details).

1.4 Concluding remarks

In conclusion, this introductory section consisted of two primary sections, each contributing to a comprehensive understanding of tensor algebra, underlying arithmetic, and exact or approximate decompositions of matrices/tensors. Furthermore, it gives insights on their applications in the field of quantum chemistry.

The first section provides an overview of matrix and tensor notations, various linear operations and decomposition methods yielding either to exact or approximate low-rank matrix or tensor decompositions. Particularly, detailed explanations are given on the TT decomposition, highlighting the advantages of performing arithmetic operations within

the TT format. Additionally, we present two commonly used algorithms for the TT representation or the compression of tensors in TT format. These algorithms are commonly known as TT-SVD and TT-rounding, respectively. The second section shifts focus to tensor-based high-dimensional problems arising from the electronic Schrödinger equation in quantum chemistry. Notably, we start by the efficient treatment of the 4-th order tensor of two-electron integrals, as it appears in many quantum chemistry methods for the search of the ground-state energy. The discussion is further extended to one of the well-known approximation methods, DMRG and its integration with TT representations. We initiated with an overview of the second quantization formalism and the representation of the Hamiltonian operator within this framework. We further elaborated its representation in a TTO-format. Then, we gave a detailed description of the algorithm alongside time complexities of key computational steps, outlining the most time-consuming one. As we transition from this introductory section, our focus now shifts to the first tensor-based high-dimensional problem, namely, the treatment of the 4-th order two-electron integrals tensor in the upcoming chapter.

Chapter 2

Low-rank approximation of long-range two-electron integrals tensors

Contents

2.1 Introduction

In the initial introductory chapter, we highlighted the importance of the two-electron integrals 4-th order tensor as a key element in various quantum chemistry methods. These integrals are essential but challenging to compute due to the presence of singularity as well as the exponential scaling in terms of the number of basis functions. To address this challenge, our attention was drawn to an efficient approach for the treatment of singularities, that is based on the range-separation technique. Therefore, in this chapter, we introduce two new approximation methods for the numerical evaluation of the long-range part

of the range-separated Coulomb potential and the approximation of the resulting high dimensional Two-Electron Integrals tensor (TEI) with long-range interactions. The first method exploits the tensorized structure of the compressed two-electron integrals obtained through two-dimensional Chebyshev interpolation combined with Gaussian quadrature. The second method exploits the Fast Multipole Method (FMM) which we briefly review in this chapter. Numerical experiments for different medium size molecules on high quality basis sets outline the efficiency of the two methods. Detailed algorithmic is provided in this chapter as well as numerical comparison of the introduced approaches. This chapter corresponds to our publication [4]. Igor Chollet (assistant professor in LAGA, Sorbonne university) also contributed to this work by bringing his expertise in FMM method and actively participating in the testing of his library, *defmm*[1]. The initial inspiration for this work came from Julien Toulouse (assistant professor in Sorbonne university), who introduced us to the chemistry background and the crucial role of these integrals in the quantum package, [33], they were developing for running molecular simulations. These integrals posed a significant time and memory challenge in their simulations.

2.2 Context and related work

As already introduced in Section 1.2, given a finite basis set $\{g_\mu\}_{1\leq\mu\leq N_b}$, $g_\mu \in H^1(\mathbb{R}^3)$, these integrals are defined as:

$$\mathcal{B}_{\mu\nu\kappa\lambda} = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{g_\mu(\boldsymbol{x})g_\nu(\boldsymbol{x})g_\kappa(\boldsymbol{y})g_\lambda(\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|} d\boldsymbol{x}d\boldsymbol{y},\ with\ \mu,\nu,\kappa,\lambda \in \{1,..,N_b\}\,. \tag{2.1}$$

These six-dimensional integrals are the entries of a fourth-order tensor, referred to as $\mathcal{B}$, with $O(N_b^4)$ entries with N_b being the number of basis functions $\{g_\mu\}_{1\leq\mu\leq N_b}$. We denote each entry of the tensor $\mathcal{B}$ by $\mathcal{B}(\mu;\nu;\kappa;\lambda) := \mathcal{B}_{\mu\nu\kappa\lambda}$. These integrals arise in various quantum chemistry methods: ab initio Hartree-Fock (HF) calculations, post-HF models, Density Functional Theory (DFT) and they are involved in the generation of the Hamiltonian appearing in QC-DMRG calculations [13, 14, 44, 105, 117]. Considerable efforts have been devoted to minimize the cost of the integrals evaluation which is a challenging computational problem since it requires the evaluation N_b^4 six-dimensional integrals that are singular due to the presence of the two-electron operator, also known as the Coulomb potential $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}$ and where N_b increases drastically with the molecular system size.

Indeed, many works exist in the literature for the evaluation of these integrals. Initially, they were calculated analytically, in the particular case of Gaussian type basis functions as defined in (1.93), which made it easy to derive analytical formulas. But since the analytical evaluation is specific to each integral, the last does not allow for a systematic evaluation. To address this issue, exact evaluation of these integrals appeared, using various techniques such as recurrence relations and Rys quadrature, McMurchie and Davidson method[25, 44, 65, 87]. These techniques are restricted to certain types of basis functions, in particular Gaussian-type functions. Efficient screening techniques can be used, as discussed in [100], with aim to minimize the number of computed integrals. So far, these evaluations could potentially become a computational bottleneck due to their dependence on the chosen basis set and the lack of systematic evaluations. To address this issue, more recently, many works has focused on developing numerical approximations to speed-up the evaluation of these integrals. Two commonly used methods for approximating the TEI tensor are Density Fitting (DF) [109, 132] and Cholesky decomposition [67]. Both methods construct a low-rank approximation of the TEI tensor. On one hand, the DF method employs an auxiliary fitting basis substituting the calculation of four-index tensor integrals with the computation of three-index tensor integrals. The accuracy of the approximation depends on the choice of this basis. On the other hand, the Cholesky decomposition approximates the matricized TEI tensor, which is a symmetric positive semi-definite matrix, with a low-rank approximate matrix whose rank depends on the desired accuracy of the approximation. Better compression can be achieved with Continuous Fast Multipole Methods (CFMM)[130] where linear scaling can be obtained in terms

[1] https://github.com/IChollet/defmm

of the number of basis functions N_b. By using multipole expansion and exploiting the fact that well-separated charge distributions (the product of the two basis functions g_μ and g_ν) can be well approximated, the two-electron integrals can be represented in a compressed form. This approach is particularly useful for the construction of the Coulomb matrix, which appears in HF and DFT calculations [75]. Additionally, there exist more efficient algebraic generalizations of CFMM, as discussed in [136]. Another method proposed in [62] uses tensor-product numerical integration. It involves performing operations along one dimension, such as the 1D Hadamard product, 1D convolution product, and 1D scalar product, to evaluate the TEI tensor, instead of performing 3D algebraic operations. It can be generalized to any low-rank basis sets and can achieve high accuracy with fine 3D Cartesian grids. In [62], the representation of the TEI tensor is further compressed using Quantized Tensor Train (QTT)[63] and low-rank Cholesky decomposition.

An alternative approach to tackle the evaluation of the two-electron integrals is to develop methods dealing with smooth potential. We consider in this chapter an approach that relies on the range-separation of the Coulomb potential where the last is split into a smooth *long-range part* and a complementary diverging *short-range part*. This partition strategy, known as Ewald method introduced in 1921 by Paul P. Ewald as well, has been employed in many quantum chemistry methods [35, 71, 72, 79, 96, 102, 110, 120, 121, 122, 123]. Specifically, it is prominent in hybrid methodologies [96, 121, 123], where both wave-function-based methods such as HF and density-based approach such as DFT are combined to exploit their respective strengths. One of the primary goals of using range separation technique in electronic structure calculations are efficiency and accuracy [30, 96]. It is particularly relevant for improving the accuracy of DFT, which depends on an approximate exchange-correlation functional that treats the electron correlation effects [14]. However, it has been shown that these approximate functionals may be less reliable and accurate when dealing with systems involving long-range or static electron correlation effects [30, 96, 121]. In order to cope with this, two different strategies have been considered. Correcting the available functional or introduce post-HF methods by means of range separation. The range separation technique allows to apply two different methods: when wave-function-based methods are used for the long-range part, accurate and faster convergence is achieved with respect to the basis set [30], and when DFT is used for the short-range part, low computational costs are achieved. As a result, the total energy of a given molecular system is computed via wave-function-based methods, such as self-consistent field method (SCF) [117], for long-range contributions and DFT for short-range contributions. These hybrid methods involve the evaluation of long-range two-electron integrals, long-range Coulomb and exchange matrices [79] which is the focus of this work.

The splitting of the Coulomb potential is done through the error function $erf(\omega \|\boldsymbol{x}-\boldsymbol{y}\|)$ as follows:

$$\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} = \frac{erfc\left(\omega \|\boldsymbol{x}-\boldsymbol{y}\|\right)}{\|\boldsymbol{x}-\boldsymbol{y}\|} + \frac{erf\left(\omega \|\boldsymbol{x}-\boldsymbol{y}\|\right)}{\|\boldsymbol{x}-\boldsymbol{y}\|}, 0 \leq \omega < \infty, \tag{2.2}$$

with

$$\frac{erf\left(\omega \|\boldsymbol{x}-\boldsymbol{y}\|\right)}{\|\boldsymbol{x}-\boldsymbol{y}\|} = \frac{2}{\|\boldsymbol{x}-\boldsymbol{y}\|\sqrt{\pi}} \int_{\left[0,\omega\|\boldsymbol{x}-\boldsymbol{y}\|\right]} e^{-t^2} dt, \tag{2.3}$$

and

$$erfc\left(\omega \|\boldsymbol{x}-\boldsymbol{y}\|\right) = 1 - erf\left(\omega \|\boldsymbol{x}-\boldsymbol{y}\|\right), \tag{2.4}$$

where $\boldsymbol{x} = (x_1, x_2, x_3), \boldsymbol{y} = (y_1, y_2, y_3) \in \mathbb{R}^3$, ω is a positive parameter that controls the separation range. The long-range contribution in equation (2.3) is a smooth function such that, for small ω, the singularity is eliminated at $\|\boldsymbol{x}-\boldsymbol{y}\| = 0$. When $\omega = 0$, the long-range part vanishes and when $\omega \to \infty$, it approaches the Coulomb potential $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}$. The short-range contribution (the complementary function in Equation (2.2)) has singularity at $\|\boldsymbol{x}-\boldsymbol{y}\| = 0$.

Some numerical methods exist already for handling the long-range kernel (2.3). One approach, as shown in [22, 71], employs spherical numerical integration of the kernel's Fourier transform in spherical coordinates in reciprocal space kernel. In [78, 79], the numerical integration uses spherical harmonics and spherical Bessel functions. In both cases, a truncated sum of the product of separable functions expressed in terms of the

3D coordinates of electrons is obtained. Notably, The range separation has found application in other contexts, such as solving the linear and nonlinear Poisson-Boltzmann equations, in [9, 10, 69]. In these studies, the authors adopt a more generalized interpretation of range separation, one that does not necessarily rely on the error function. By combining the Laplacian Gaussian transform with sinc-quadratures and by considering a fine 3D Cartesian grid, a canonical representation is obtained for the Coulomb potential. Subsequently, this representation is partitioned into distinct long-range and short-range components based on the quadrature points and the underlying functions' support. By noting that the long-range component inherent in the canonical tensor representation exhibits low canonical rank, it is further compressed using the canonical to Tucker decomposition technique [64].

Now, following Equation (2.2), the two-electron integrals tensor can be expressed as the sum of two terms:

$$\begin{aligned}\mathcal{B}(\mu;\nu;\kappa;\lambda) = &\underbrace{\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{erf\left(\omega\left\|\boldsymbol{x}-\boldsymbol{y}\right\|\right) g_\mu(\boldsymbol{x})g_\nu(\boldsymbol{x})g_\kappa(\boldsymbol{y})g_\lambda(\boldsymbol{y})}{\left\|\boldsymbol{x}-\boldsymbol{y}\right\|} d\boldsymbol{x}d\boldsymbol{y}}_{\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)} \\ &+ \underbrace{\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{erfc\left(\omega\left\|\boldsymbol{x}-\boldsymbol{y}\right\|\right) g_\mu(\boldsymbol{x})g_\nu(\boldsymbol{x})g_\kappa(\boldsymbol{y})g_\lambda(\boldsymbol{y})}{\left\|\boldsymbol{x}-\boldsymbol{y}\right\|} d\boldsymbol{x}d\boldsymbol{y}}_{\mathcal{B}^{sr}(\mu;\nu;\kappa;\lambda)},\end{aligned} \tag{2.5}$$

with $\mu,\nu,\kappa,\lambda \in [N_b]$, $\mathcal{B}^{lr}$ refers to the long-range TEI tensor and $\mathcal{B}^{sr}$ refers to the short-range TEI tensor. In this work, we focus on the numerical treatment of the long-range kernel, that we denote $K(\boldsymbol{x},\boldsymbol{y}) = \frac{erf(\omega\|\boldsymbol{x}-\boldsymbol{y}\|))}{\|\boldsymbol{x}-\boldsymbol{y}\|}, \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3$, and on the approximation of the long-range two-electron integrals given by:

$$\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(\boldsymbol{x})g_\nu(\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{y})g_\kappa(\boldsymbol{y})g_\lambda(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}, \mu,\nu,\kappa,\lambda \in [N_b]\,. \tag{2.6}$$

We consider restrictions of the finite basis functions $\{g_\mu\}_{1\le\mu\le N_b}$ to sufficiently large compact support $[-b,b]^3 \subset \mathbb{R}^3$ such that we have:

$$\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) = \int_{[-b,b]^3}\int_{[-b,b]^3} g_\mu(\boldsymbol{x})g_\nu(\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{y})g_\kappa(\boldsymbol{y})g_\lambda(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}, \mu,\nu,\kappa,\lambda \in [N_b]\,. \tag{2.7}$$

In what follows, we introduce two numerical approaches for the numerical evaluation of the smooth long-range interaction and the approximation of the long-range TEI tensor. First, instead of performing a naive numerical computation of $K(\boldsymbol{x},\boldsymbol{y})$ over $N\times N\times N$ 3D Cartesian grids, we consider two-dimensional Chebyshev interpolation method using only $N^{\frac{1}{3}}\times N^{\frac{1}{3}}$ isotropic Chebyshev grids combined with Gaussian-quadrature rule in order to approximate $K(\boldsymbol{x},\boldsymbol{y})$. We refer to this approach as TA for Tensorized Approximation and we denote the approximation method for the evaluation of the long-range two-electron integrals by LTEI-TA. This numerical approximation yields to a tensorized expression of the six-dimensional integrals with erf-interaction leading to substantial time complexity reduction to evaluate one integral of the form $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$. In practice, especially in HF calculations, multiple matrix-vector or matrix-matrix multiplication using this fourth-order tensor are performed, which is one of the most-time consuming step. Therefore, we introduce, using LTEI-TA approach, a new alternative way to approximate these integrals by means of a factorized representation of the fourth-order tensor $\mathcal{B}^{lr} \in \mathbb{R}^{N_b\times N_b\times N_b\times N_b}$, leading to an efficient application of the matricization of $\mathcal{B}^{lr}$ to a vector with a significant reduction in time complexity to $\mathcal{O}(\epsilon N^{4/3}), \epsilon \ll N$ instead of $\mathcal{O}\left(N^2\right)$ given a naive computation. These complexities may be further reduced, if a Gaussian basis is used, due to their interesting properties. Additionally, we propose to express the high dimensional fourth-order tensor $\mathcal{B}^{lr}$ in a more compressed format by using screening techniques and low-rank approximation methods.

On a second note, we consider Chebyshev interpolation combined with FMM [29, 38] leading to linear time complexity when computing the FMM-accelerated matrix vector product involving the two-electron integrals tensor. This method is referred to as *LTEI-FMM*. We provide detailed comparison between the two approaches and discuss to what extent the relative performances of these methods make them attractive for different application cases. In order to test the performance of our algorithm, we use the data

sets of molecular properties calculated from quantum chemistry for some moderate size molecules. These data sets are extracted from *quantum package* [33].

This chapter is organized as follows: in Section 2.3, we describe our new tensorized method to approximate $K(\boldsymbol{x}, \boldsymbol{y})$ and we present our LTEI-TA scheme for the element-wise evaluation of the two-electron integrals based on the underlying tensorized structure. We describe also using LTEI-TA a factorized expression of the two-electron integrals tensor and we derive error bounds and theoretical complexities for the approximation process we use. In Section 2.4 we demonstrate that our kernel $K(\boldsymbol{x}, \boldsymbol{y})$ is asymptotically smooth, so that we can benefit from fast hierarchical methods (especially FMM) in order to efficiently evaluate the two-electron integrals decompositions. Hence, we reformulate these decompositions as N-body problems on non-uniform particle distributions. In Section 2.5, we showcase a practical application within electronic calculations, which is the evaluation of the Coulomb and Exchange matrices arising in SCF calculations, by using the decompositions of the two-electron integrals tensor obtained through the new introduced approaches. In Section 2.6.1, further compression techniques are also presented, extending screening approaches and low rank approximation methods to our new decompositions. Finally, results of numerical tests of both methods are presented as well as a summary of our findings. We use Julia open-source language to test the new approximation method TA and the evaluation scheme LTEI-TA[2] and the C++ library $defmm$[3]for LTEI-FMM.

We review in the following several definitions and properties that we use in the subsequent sections.

In the different approximations derived in this work, the product of two Gaussian type functions is often used. Therefore, we recall the general product rule between two Gaussian functions.

Proposition 2.2.1 ([66])**.** *Let $g_1(\boldsymbol{x}) = e^{-c_1\|\boldsymbol{x}-\boldsymbol{r}_1\|^2}, g_2(\boldsymbol{x}) = e^{-c_2\|\boldsymbol{x}-\boldsymbol{r}_2\|^2}$ be Gaussian functions with $\boldsymbol{x}, \boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{R}^3, c_1, c_2 \in \mathbb{R}$. The product of these functions is:*

$$g_{12}(\boldsymbol{x}) = g_1(\boldsymbol{x})g_2(\boldsymbol{x}) = e^{\frac{-c_1c_2}{c_1+c_2}\|\boldsymbol{x}-\boldsymbol{r_{12}}\|^2} e^{-(c_1+c_2)\|\boldsymbol{x}-\boldsymbol{r_{12}}\|^2}, \tag{2.8}$$

where $\boldsymbol{r_{12}} = \frac{c_1}{c_1+c_2}\boldsymbol{r}_1 + \frac{c_2}{c_1+c_2}\boldsymbol{r}_2$.

We also have recourse to two-dimensional Chebyshev interpolation. Therefore, we give the expressions of the Chebyechev polynomials as well as the Chebyshev coefficients.

Definition 26 (Two dimensional Chebyshev interpolation [103, 124])**.** For a given continuous function f(x,y) on $[a, b]^2, a, b \in \mathbb{R}$, the two-dimensional Chebyshev interpolation of this function is given by its interpolating polynomial that we denote:

$$\tilde{f}(x, y) = \sum_{n,m=0}^{N} \alpha_{nm} T_n(x) T_m(y), \tag{2.9}$$

where N is the number of interpolation nodes, $T_n(x) = cos(n\ acos(x)), x \in [a, b]$, $n \in [N]$ are the Chebyshev polynomials:

$$\alpha_{nm} = \frac{c_{nm}}{N^2} \sum_{k,k'=1}^{N} f(x_k, y'_k) T_n(x_k) T_m(y'_k), \quad c_{n,m} = \begin{cases} 1 & \text{if } m = n = 0 \\ 2 & \text{if } m \neq n = 0 \text{ or } n \neq m = 0 \\ 4 & \text{if } m \neq 0, n \neq 0 \end{cases} \tag{2.10}$$

are Chebyshev interpolation coefficients. The nodes x_k, y'_k form the Chebyshev two-dimensional grids such as Chebyshev-Gauss points (first kind):

$$x_k = \cos\theta_k, \quad \theta_k = \frac{(2k-1)\pi}{2N}, \quad k = 1, \ldots, N, \tag{2.11}$$

or Chebyshev-Lobatto points (second kind):

$$x_k = \cos\phi_k, \quad \phi_k = \frac{(k-1)\pi}{N-1}, \quad k = 1, \ldots, N. \tag{2.12}$$

[2]https://github.com/sbadred/LTEI_TA.jl.git

[3]https://github.com/IChollet/defmm

The following proposition gives the interpolation error of the two-dimensional Chebyshev approximation.

Proposition 2.2.2 (Interpolation error [54]). *Let $\tilde{f}(x,y)$ be an interpolating polynomial of $f(x,y)$ on $[a,b]^2$ at Chebyshev N interpolation nodes and suppose that the partial derivatives $\frac{\partial^{N+1} f(x,y)}{\partial x^{N+1}}$ and $\frac{\partial^{N+1} f(x,y)}{\partial y^{N+1}}$ exist and are continuous for all $(\mathrm{x},\mathrm{y}) \in [a,b]^2$. We have:*

$$|f(x,y) - \tilde{f}(x,y)| \leq \frac{\left(\frac{b-a}{2}\right)^{N+1}}{2^N\,(N+1)!} c_1 + \frac{\delta\left(\frac{b-a}{2}\right)^{N+1}}{2^N\,(\mathrm{N}+1)!} c_2, \tag{2.13}$$

where

$$c_1 = \max_{\xi\in[a,b]} \left| \frac{\partial^{N+1} f(\xi,y)}{\partial \xi^{N+1}} \right|, c_2 = \max_{(\xi,\eta)\in[a,b]^2} \left| \frac{\partial^{\mathrm{N}+1} f(\xi,\eta)}{\partial \eta^{\mathrm{N}+1}} \right|, \tag{2.14}$$

$$\delta = \max_{s\in[a,b]} \sum_{i=0}^{N} |L_{i,N}(s)|\,. \tag{2.15}$$

The so-called Lebesgue constant δ grows only logarithmically if Chebyshev interpolation nodes are used, $L_{i,N}(s)$ are Lagrange polynomials of degree N.

The following proposition recalls the upper bound of Gaussian-quadrature rule error.

Proposition 2.2.3 (Quadrature error, Section 5.2 [57]). *Let $[a,b]$ be a real closed interval of length $|b-a| > 0$ and let $f \in \mathbb{C}^{2N_q}([a,b]), N_q \geq 1$, the integration of f over $[a,b]$ can be given as follows, using Gaussian quadrature rule:*

$$\int_{[a,b]} f(x)dx = \int_{[-1,1]} f\left(\frac{b-a}{2}z + \frac{a+b}{2}\right)\frac{dx}{dz}dz = \frac{b-a}{2}\sum_{i=1}^{N_q} w_i f\left(\frac{b-a}{2}z_i + \frac{a+b}{2}\right) + R_{N_q}, \tag{2.16}$$

where w_i and x_i are the weights and nodes of the quadrature rule, N_q is the number of quadrature points and R_{N_q} refers to the Gaussian quadrature error. This last quantity verifies:

$$|R_{N_q}| \leq \frac{|b-a|^{2N_q+1}\,(N_q!)^4}{(2N_q+1)[(2N_q)!]^3} \left\| \frac{d^{2N_q}}{ds^{2N_q}} f(s) \right\|_{\infty,[a,b]}. \tag{2.17}$$

2.3 Long-range two-electron integrals tensor factorization through tensorized approximation

2.3.1 The element-wise evaluation of the two-electron integrals tensor

In this section we introduce a new numerical method that allows to evaluate efficiently the two-electron integrals through the factorization of the long-range Coulomb potential. This method, that we refer to as TA, factorizes the fourth order long-range two-electron integrals tensor $\mathcal{B}^{lr}$ through the approximation of the long-range kernel $K(\boldsymbol{x},\boldsymbol{y})$ with two-dimensional Chebyshev interpolation and Gaussian quadrature. Error bounds for the numerical approximation of the long-range two-electron integrals are also provided.

2.3.2 The element-wise evaluation of the long-range TEI tensor

We first describe the efficient evaluation of the six-dimensional integrals $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$ defined in (2.6). We start by presenting our approach for computing the long-range function $K(\boldsymbol{x},\boldsymbol{y})$ defined as:

$$K(\boldsymbol{x},\boldsymbol{y}) = \frac{erf(\omega\,\|\boldsymbol{x}-\boldsymbol{y}\|)}{\|\boldsymbol{x}-\boldsymbol{y}\|} = \frac{2}{\sqrt{\pi}} \frac{\int_{[0,\omega\|\boldsymbol{x}-\boldsymbol{y}\|]} e^{-t^2} dt}{\|\boldsymbol{x}-\boldsymbol{y}\|}, \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3. \tag{2.18}$$

Let $t = s\,\|\boldsymbol{x}-\boldsymbol{y}\|$. With this change of variable, we obtain:

$$K(\boldsymbol{x},\boldsymbol{y}) = \frac{2}{\sqrt{\pi}} \int_{[0,\omega]} e^{-s^2\|\boldsymbol{x}-\boldsymbol{y}\|^2} ds, \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3. \tag{2.19}$$

Using the Gaussian quadrature rule (see Proposition 2.2.3), we can evaluate numerically the integral in (2.19) as:

$$\int_{[0,\omega]} e^{-s^2\|\boldsymbol{x}-\boldsymbol{y}\|^2} ds = \frac{\omega}{2}\int_{[-1,1]} e^{-(\frac{\omega}{2}+\frac{\omega}{2}z)^2\|\boldsymbol{x}-\boldsymbol{y}\|^2} dz \approx \frac{\omega}{2}\sum_{i=1}^{N_{q_1}} w_i e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2\|\boldsymbol{x}-\boldsymbol{y}\|^2}, \tag{2.20}$$

where w_i are the Gaussian quadrature weights, z_i are the Gaussian quadrature nodes, and N_{q_1} is the number of quadrature points. The coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$ are denoted by $(x_1, x_2, x_3), (y_1, y_2, y_3)$ respectively. The exponential term in (2.20) can be written as:

$$e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2\|\boldsymbol{x}-\boldsymbol{y}\|^2} = \prod_{l=1}^{3} e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2(x_l-y_l)^2}, l \in \{1,2,3\}. \tag{2.21}$$

Given the truncated computational box $[-b,b]^3, b \in \mathbb{R}$, each function of the form $e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2(x_l-y_l)^2}$, $i \in [N_{q_1}], l \in \{1,2,3\}$ is smooth, differentiable (hence continuous) on $[-b,b]^2$, so that it is an excellent candidate for two-dimensional Chebyshev interpolation. According to Definition 26, the interpolated function can be written as:

$$e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2(x_l-y_l)^2} \approx \sum_{n_l,m_l=1}^{N_i} \alpha^{(i)}_{n_l m_l} T^{(i)}_{n_l}(x_l) T^{(i)}_{m_l}(y_l), \tag{2.22}$$

where N_i is the number of interpolation nodes for $i \in [N_{q_1}]$, $x_l,\ y_l \in [-b,b]$, and $l \in \{1,2,3\}$.

Remark 1. As opposed to polynomial interpolation for fast evaluation methods for singular kernels (often also based on Chebyshev grids [29]) as appearing in integral equations, we here deal with non-singular asymptotically smooth kernels (see Proposition (2.4.1)) (3D long-range kernel with erf interaction as well as exponential 1D kernels in equation (2.22)), allowing to perform interpolations in the whole domain at once, without caring about admissibility conditions. This is the reason why error estimates, as well as required number of quadrature points and interpolation nodes depend on ωb instead of ratio between well-separated cell centers and distance (see Section 2.4).

We recall that among the advantages of using two-dimensional Chebyshev interpolation method is that forming two-dimensional Chebyshev grids $N_i \times N_i$ for each function (2.22) takes $\mathcal{O}(N_i^2)$ storage complexity, where N_i is the number of interpolation points needed. Furthermore, Chebyshev-Lobatto nodes can be obtained in linearithmic time using Fast Fourier Transform (FFT) [93]. This is one of the reasons for which we use Chebyshev basis. Our implementation that we discuss in more details in Section 2.7 uses FFTW [31] routine in Julia and the *chebfun2* library [124] to find the number of interpolation points N_i of the functions in (2.22). By replacing (2.22) and (2.20) in (2.19), the numerical approximation of the kernel $K(\boldsymbol{x},\boldsymbol{y})$ becomes:

$$K(\boldsymbol{x},\boldsymbol{y}) \approx \frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} w_i \left(\sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \alpha^{(i)}_{n_l m_l} T^{(i)}_{n_l}(x_l) T^{(i)}_{m_l}(y_l) \right), \tag{2.23}$$

where $\omega \geqslant 0$ is the parameter that regulates the separation range of the long-range/short-range interactions, $\alpha^{(i)}_{n_l m_l}$ are the N_i Chebyshev nodes, $T^{(i)}_{n_l}(x_l), T^{(i)}_{m_l}(y_l)$ are the Chebyshev polynomials (see Definition 26) and w_i are the Gaussian quadrature weights with $i \in [N_{q_1}]$. All along this work, we denote N the maximum number of interpolation points in the tensorized Chebyshev grid in all directions such that $N = (\max_{1\leq i\leq N_{q_1}} \{N_i\})^3$. The pre-computation cost here to approximate the kernel (2.23) is $\mathcal{O}(N_{q_1} N^{\frac{1}{3}}(log(N^{\frac{1}{3}}) + N^{\frac{1}{3}}))$: $\mathcal{O}(N_{q_1} N^{\frac{1}{3}} log(N^{\frac{1}{3}}))$ FLOPS[4] for the evaluation of the Chebyshev coefficient matrices using FFT algorithm, linearithmic in the number of interpolation points *in a single direction* $N^{\frac{1}{3}}$ and linear in the number of quadrature points, and $\mathcal{O}(N_{q_1} N^{\frac{2}{3}})$ FLOPS for forming the Chebyshev two-dimensional grids. Figure 2.1, displays the accuracy of the approximation of the long-range kernel for $\omega = 0.1$ on the computational box $[-10,10]^3$ by using

[4]Floating Point Operations Per Second

a Chebyshev grid with maximum number of interpolation points $N = 8000$ and $N_{q_1} = 11$ by varying only $x_1, y_1 \in [-10, 10]$.

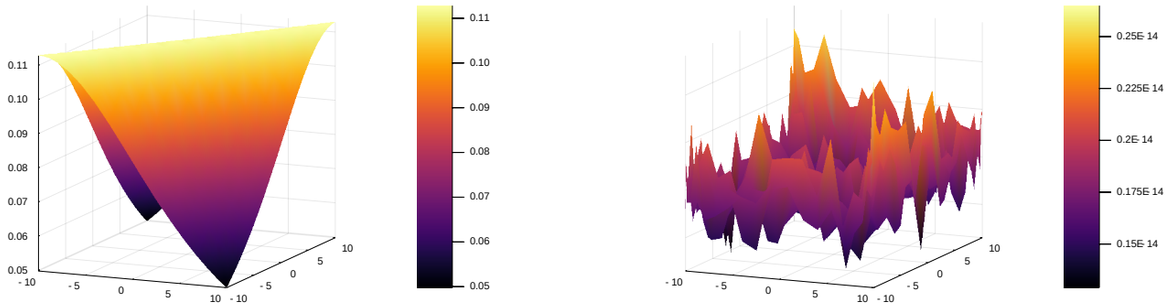


Figure 2.1: Approximation of the long-range Coulomb potential and associated relative error in Frobenius norm, for $\omega = 0.1$.

We consider now the finite six-dimensional integral $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$ defined in (2.7) on the same truncated computational box $[-b,b]^3 \times [-b,b]^3, b \in \mathbb{R}$ with $\mu,\nu,\kappa,\lambda \in [N_b]$, where N_b is the number of basis functions that we defined in (1.94) and b is the size of the computational box that is chosen according to the most slowly decaying basis functions. We discuss this aspect in more details in Section 2.6.2. By replacing $K(\boldsymbol{x},\boldsymbol{y})$ with its approximation from (2.23), the numerical approximation of $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$, denoted by $\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)$, writes:

$$\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda) = \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \left(\int_{[-b,b]^3} \int_{[-b,b]^3} g_{\mu\nu}(\mathbf{x}) g_{\kappa\lambda}(\mathbf{y}) \left(\sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \alpha^{(i)}_{n_l m_l} T^{(i)}_{n_l}(\mathbf{x}_l) T^{(i)}_{m_l}(\mathbf{y}_l) \right) d\mathbf{x} d\mathbf{y}, \right. \tag{2.24}$$

where $g_{\mu\nu}(\boldsymbol{x}) = g_\mu(\boldsymbol{x}) g_\nu(\boldsymbol{x})$, $g_{\kappa\lambda}(\boldsymbol{y}) = g_\kappa(\boldsymbol{y}) g_\lambda(\boldsymbol{y})$ such that according to (1.94) we have (showing only $g_{\mu\nu}$ expression)

$$g_{\mu\nu}(\boldsymbol{x}) = g_\mu(\boldsymbol{x}) g_\nu(\boldsymbol{x}) \quad = \quad \sum_{j_1=1}^{I_\mu} \sum_{j_2=1}^{I_\nu} c_{j_1} c_{j_2} \prod_{l=1}^{3} f^{(j_1)}_\mu(\boldsymbol{x}_l) f^{(j_2)}_\nu(\boldsymbol{x}_l) = \sum_{j=1}^{I_{\mu\nu}} c_j \prod_{l=1}^{3} f^{(j)}_{\mu\nu}(\boldsymbol{x}_l) \tag{2.25}$$

where $I_{\mu\nu} = I_\mu I_\nu, c_j = c_{j_1} c_{j_2}, f^{(j)}_{\mu\nu}(\boldsymbol{x}_l) = f^{(j_1)}_\mu(\boldsymbol{x}_l) f^{(j_2)}_\nu(\boldsymbol{x}_l)$. Expressing the three dimensional function $g_{\mu\nu}(\boldsymbol{x})$ as a sum of separable functions is important to reduce the evaluation cost of $\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)$ such that after replacing the Gaussian basis functions in (2.24) by their separable expression (2.25) we obtain:

$$\begin{aligned}
\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda) &= \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \left(\int_{[-b,b]^3} \int_{[-b,b]^3} g_{\mu\nu}(\boldsymbol{x}) g_{\kappa\lambda}(\boldsymbol{y}) \left(\sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \alpha^{(i)}_{n_l m_l} T^{(i)}_{n_l}(\boldsymbol{x}_l) T^{(i)}_{m_l}(\boldsymbol{y}_l) \right) d\boldsymbol{x} d\boldsymbol{y} \right) \\
&= \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \left(\int_{[-b,b]^3} \int_{[-b,b]^3} \left(\sum_{j=1}^{I_{\mu\nu}} \sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \prod_{l=1}^{3} f^{(j)}_{\mu\nu}(\boldsymbol{x}_l) f^{(j')}_{\kappa\lambda}(\boldsymbol{y}_l) \right) \left(\sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \alpha^{(i)}_{n_l m_l} T^{(i)}_{n_l}(\boldsymbol{x}_l) T^{(i)}_{m_l}(\boldsymbol{y}_l) \right) d\boldsymbol{x} d\boldsymbol{y} \right) \\
&= \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \sum_{j=1}^{I_{\mu\nu}} \sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \underbrace{\sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \left(\alpha^{(i)}_{n_l m_l} \int_{[-b,b]} f^{(j)}_{\mu\nu}(\boldsymbol{x}_l) T^{(i)}_{n_l}(\boldsymbol{x}_l) d\boldsymbol{x}_l \int_{[-b,b]} f^{(j')}_{\kappa\lambda}(\boldsymbol{y}_l) T^{(i)}_{m_l}(\boldsymbol{y}_l) d\boldsymbol{y}_l \right)}_{\approx \mathbf{F}_i(j;j')}.
\end{aligned} \tag{2.26}$$

We note that the expression of $\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)$ in (2.26) involves the numerical evaluation of one dimensional integrals. We associate each such integral with the element of a matrix and obtain two matrices $\mathbf{W}^{(i,l)}_{\mu\nu} \in \mathbb{R}^{I_{\mu\nu} \times N_i}$ and $\mathbf{W}^{(i,l)}_{\kappa\lambda} \in \mathbb{R}^{I_{\kappa\lambda} \times N_i}$ defined entry-wise as:

$$\mathbf{W}^{(i,l)}_{\mu\nu}(j;n_l) = \int_{[-b,b]} f^{(j)}_{\mu\nu}(\boldsymbol{x}_l) T^{(i)}_{n_l}(\boldsymbol{x}_l) d\boldsymbol{x}_l \text{ and } \mathbf{W}^{(i,l)}_{\kappa\lambda}(j';m_l) = \int_{[-b,b]} f^{(j')}_{\kappa\lambda}(\boldsymbol{y}_l) T^{(i)}_{m_l}(\boldsymbol{y}_l) d\boldsymbol{y}_l. \tag{2.27}$$

We use one-dimensional Gaussian quadrature rule for the evaluation of (2.27). Their approximation is denoted by $\mathbf{W}_{\mu\nu}^{(i,l)}(j;n_l)$(resp. $\mathbf{W}_{\kappa\lambda}^{(i,l)}(j';m_l)$. We further define matrices $\mathbf{F}_i, i \in [N_{q_1}]$, as displayed in (2.26). By replacing the expressions of $\tilde{\mathbf{W}}_{\mu\nu}^{(i,l)}$ and $\tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)}$, we obtain

$$\mathbf{F}_i(j;j') = \sum_{\substack{n_1,m_1,\\ \cdots,n_3,m_3=1}}^{N_i} \prod_{l=1}^{3} \left(\alpha_{n_l m_l}^{(i)} \tilde{\mathbf{W}}_{\mu\nu}^{(i,l)}(j;n_l) \tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)}(j';m_l)\right). \tag{2.28}$$

By changing the order of summation in (2.28) and exploiting Khatri-Rao as well as Kronecker structures (see their definitions in Section 1.1), we obtain the factorized representation of $\mathcal{B}_{LTEI-TA}^{lr}$ as given in the following theorem.

Theorem 2. The long-range two-electrons integrals has a factorized representation that writes

$$\mathcal{B}_{LTEI-TA}^{lr}(\mu;\nu;\kappa;\lambda) = \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \sum_{j=1}^{I_{\mu\nu}} \sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j'), \tag{2.29}$$

where $\mathbf{F}_i \in \mathbb{R}^{I_{\mu\nu} \times I_{\kappa\lambda}}$

$$\mathbf{F}_i = (\diamond_{l=1}^{3} \tilde{\mathbf{W}}_{\mu\nu}^{(i,l)})(\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i)(\diamond_{l=1}^{3} \tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)})^* = \odot_{l=1}^{3} \tilde{\mathbf{W}}_{\mu\nu}^{(i,l)} \mathbf{A}_i \tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)*}, \tag{2.30}$$

where $\mathbf{A}_i \in \mathbb{R}^{N_i \times N_i}$ are the Chebyshev coefficients matrices such that $\mathbf{A}_i(n_l;m_l) = \alpha_{n_l m_l}^{(i)}$ for $n_l, m_l \in [N_i], l \in \{1,2,3\}$ with $\alpha_{n_l m_l}^{(i)}$ defined in (2.22),$\tilde{\mathbf{W}}_{\mu\nu}^{(i,l)} \in \mathbb{R}^{I_{\mu\nu} \times N_i}$ and $\tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)} \in \mathbb{R}^{I_{\kappa\lambda} \times N_i}$ are the numerical approximation of the one-dimensional integrals defined in (2.27).

It is noted that the work of [62] also yields a factorization similar to ours for the two-electron integrals without the range separation, but with a different discretization scheme.

In what follows, algorithm 4 computes the approximated entries $(\mathcal{B}_{LTEI-TA}^{lr})_{\mu\nu\kappa\lambda}$ (2.29) given the coefficient matrix obtained from the two-dimensional Chebyshev interpolation $\mathbf{A}_i \in \mathbb{R}^{N_i \times N_i}$, for $i \in [N_{q_1}]$ and for any given pairs of $\mu, \nu, \kappa, \lambda$. This approach allows to reduce the storage complexity (resp. arithmetic complexity) to $\mathcal{O}\left(\sum_{i=1}^{N_{q_1}} N_i(N_i + I_{\mu\nu} + I_{\kappa\lambda})\right) \sim \mathcal{O}\left(N_{q_1} N^{\frac{1}{3}}(N^{\frac{1}{3}} + I_{\mu\nu} + I_{\kappa\lambda})\right)$ (resp. $\mathcal{O}\left(\sum_{i=1}^{N_{q_1}} N_i I_{\kappa\lambda}(N_i + I_{\mu\nu})\right) \sim \mathcal{O}\left(N_{q_1} N^{\frac{1}{3}} I_{\kappa\lambda}(N^{\frac{1}{3}} + I_{\mu\nu})\right)$), with $N^{\frac{1}{3}} = \max_{1 \le i \le N_{q_1}} \{N_i\}$, instead of $\mathcal{O}(N(N + I_{\mu\nu} + I_{\kappa\lambda}))$ (resp. $\mathcal{O}(N I_{\kappa\lambda}(N + I_{\mu\nu} + I_{\kappa\lambda}))$), using naïve tensorized three dimensional quadrature on the computational box $[-b,b]^3$. Numerical results for this element-wise factorization are summarized in Section 2.7.

Algorithm 4 Compute $\mathcal{B}_{LTEI-TA}^{lr}(\mu;\nu;\kappa;\lambda)$

Input: Chebyshev coefficient matrices $\mathbf{A}_i$, $\mu, \nu, \kappa, \lambda$, w_i for $i \in [N_{q_1}]$. **Output:** $\mathcal{B}_{LTEI-TA}^{lr}(\mu;\nu;\kappa;\lambda)$

procedure ELEMENT-WISE LTEI-TA
  Set $s = 0$.
  for i=1 to N_{q_1} **do**
    Compute $\tilde{\mathbf{W}}_{\mu\nu}^{(i,l)}, l \in \{1,2,3\}$, ▷ $I_{\mu\nu} \times N_i$ matrices (according to (2.27)).
    Compute $\tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)}, l \in \{1,2,3\}$, ▷ $I_{\kappa\lambda} \times N_i$ matrices (according to (2.27)).
    $\mathbf{F}_i = \odot_{l=1}^{3} \tilde{\mathbf{W}}_{\mu\nu}^{(i,l)} \mathbf{A}_i \tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)*}$.
    $s = s + w_i \sum_{j=1}^{I_{\mu\nu}} \sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j')$.
  end for
  $\mathcal{B}_{LTEI-TA}^{lr}(\mu;\nu;\kappa;\lambda) = \frac{\omega}{\sqrt{\pi}} s$.
end procedure

2.3.3 Error bound of the two-electron integrals numerical approximation

In what follows, we give a theoretical error bound associated with the element-wise numerical approximation of $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$ introduced in (2.29).

Proposition 2.3.1. *The element-wise error ϵ between the long-range two-electron integrals $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$, given a finite box $[-b,b]^3$, and it's approximation $\mathcal{B}^{lr}_{LTEI-TA}$ can be bounded as follows:*

$$|\epsilon| := \left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) - \mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)\right| \leq c_1(\omega, N_{q_1}, b) \sup_{\boldsymbol{x},\boldsymbol{y}\in[-b,b]^3} \left(\left\| \frac{d^{2N_{q_1}}}{ds^{2N_{q_1}}} f(s,\boldsymbol{x},\boldsymbol{y})\right\|_{\infty,[0,\omega]}\right) + \frac{\omega}{\sqrt{\pi}} c_2(N_{q_1}, b, N^{\frac{1}{3}}),$$

where we define the multivariate function:

$$f(s,\boldsymbol{x},\boldsymbol{y}) = exp(-s^2 \left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2), s \in [0,\omega], \boldsymbol{x},\boldsymbol{y} \in [-b,b]^3. \tag{2.31}$$

ϵ is the approximation error, N_{q_1} is the number of quadrature points, c_1 and c_2 are defined in the following proof.

Proof. We start by introducing the following function

$$h(z_i,\omega) = \int_{[-b,b]^3}\int_{[-b,b]^3} g_{\mu\nu}(\boldsymbol{x}) g_{\kappa\lambda}(\boldsymbol{y}) exp\left(-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2 \left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2\right) d\boldsymbol{x}d\boldsymbol{y} \tag{2.32}$$

$$= \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \left(\prod_{l=1}^{3}\int_{[-b,b]^2} f^{(j)}_{\mu\nu}(x_l) f^{(j')}_{\kappa\lambda}(y_l) exp\left(-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2 (x_l-y_l)^2\right) dx_l dy_l\right), \tag{2.33}$$

with $z_i, i \in [N_{q_1}]$ being the Gaussian quadrature nodes. The upper bound of ϵ can be found as follows:

$$|\epsilon| = \left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) - \mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)\right| \leq \underbrace{\left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) - \frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} w_i h(z_i,\omega)\right|}_{\epsilon_1} + \underbrace{\left|\frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} w_i h(z_i,\omega) - \mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)\right|}_{\epsilon_2}, \tag{2.34}$$

Using Proposition 2.2.3, triangle inequality, and Stirling formula given by $n! \approx \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n$, ϵ_1 is bounded as follows:

$$\epsilon_1 \leqslant c_1(\omega, N_{q_1}, b) \sup_{\boldsymbol{x},\boldsymbol{y}\in[-b,b]^3} \left(\left\| \frac{d^{2N_{q_1}}}{ds^{2N_{q_1}}} f(s,\boldsymbol{x},\boldsymbol{y})\right\|_{\infty,[0,\omega]}\right), \tag{2.35}$$

where

$$c_1(\omega, N_{q_1}, b) = \frac{2e_{N_{q_1}}}{\sqrt{\pi}} b^6 \left\|g_{\mu\nu}\right\|_{\infty,[-b,b]^3} \left\|g_{\kappa\lambda}\right\|_{\infty,[-b,b]^3}, \tag{2.36}$$

with $e_{N_{q_1}} = \frac{\omega^{2N_{q_1}+1} e^{2N_{q_1}} (N_{q_1}\pi)^{\frac{1}{2}}}{2^{6N_{q_1}+1} N_{q_1}^{2N_{q_1}} (2N_{q_1}+1)}$. The error bound of ϵ_2 needs a more detailed explanation. We replace $\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)$ by its expression defined in (2.29) such that

$$|\epsilon_2| = \left|\frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} w_i \left(h(z_i,\omega) - \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j')\right)\right| \leq \frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} |w_i| \left|h(z_i,\omega) - \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j')\right|, \tag{2.37}$$

with $\mathbf{F}_i(j;j')$ being defined in (2.28). Using the triangle inequality, the expression of $\left|h(z_i,\omega) - \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j')\right|$, for $i \in [N_{q_1}]$, can be bounded as follows:

$$\left|h(z_i) - \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j')\right| \leq \sum_{j=1}^{I_{\mu\nu}}\sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \left|\prod_{l=1}^{3}\int_{[-b,b]^2} f^{(j)}_{\mu\nu}(x_l) f^{(j')}_{\kappa\lambda}(y_l) e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2 (x_l-y_l)^2} dx_l dy_l - \mathbf{F}_i(j;j')\right|. \tag{2.38}$$

In order to evaluate the bound of (2.38), one needs to evaluate the error bound of the following expression using Proposition 2.2.2 and Proposition 2.2.3

$$\left| \int_{[-b,b]^2} f_{\mu\nu}^{(j)}(x_l) f_{\kappa\lambda}^{(j')}(y_l) e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2(x_l-y_l)^2} - \sum_{n_l,m_l}^{N_i} \alpha_{n_l m_l}^{(i)} \tilde{\mathbf{W}}_{\mu\nu}^{(i,l)}(j;n_l) \tilde{\mathbf{W}}_{\kappa\lambda}^{(i,l)}(j';m_l) \right| \leq \beta_i, l \in \{1,2,3\}\,, \tag{2.39}$$

where for $i \in [N_{q_1}]\,, j \in [I_{\mu\nu}]\,,$ and $j' \in [I_{\kappa\lambda}]$, β_i is defined as follows:

$$\begin{aligned} \beta_i =& (2b)^2 \|f_{\mu\nu}^{(j)}\|_{\infty,[-b,b]} \|f_{\kappa\lambda}^{(j')}\|_{\infty,[-b,b]} e_{N_i} \\ &+ e_{N_{q_2}} \left(\|f_{\kappa\lambda}^{(j')}\|_{\infty,[-b,b]} \left\| \frac{d^{2N_{q_2}}(f_{\mu\nu}^{(j)} T_{n_1}^{(i)})(x)}{dx^{2N_{q_2}}} \right\|_{\infty,[-b,b]} \right. \\ &+ N_{q_2} \max_{1\leq i\leq N_{q_2}}(w_i) \|f_{\mu\nu}^{(j)}\|_{\infty,[-b,b]} \\ &\left. \left\| \frac{d^{2N_{q_2}}(f_{\kappa\lambda}^{(j')} T_{m_1}^{(i)})(y)}{dy^{2N_{q_2}}} \right\|_{\infty,[-b,b]} \right), \end{aligned} \tag{2.40}$$

with $e_{N_{q_2}} = \frac{(2b)^{2N_{q_2}+1} e^{2N_{q_2}} (N_{q_2}\pi)^{\frac{1}{2}}}{2^{6N_{q_2}+1} N_{q_2}^{2N_{q_2}} (2N_{q_2}+1)}$. The term e_{N_i} is defined as follows:

$$e_{N_i} = \frac{b^{N_i+1}}{2^{1+N_i}} \frac{e^{-N_i+1}}{\sqrt{2\pi\,(N_i+1)}\left(1+N_i^{(N_i+1)}\right)} \left[\max_{-b\leq\xi\leq b} \left| \frac{\partial^{N_i+1}\mathscr{F}(z_i,\xi,y)}{\partial\xi^{N_i+1}} \right| + \delta \max_{-b\leq\xi,\eta\leq b} \left| \frac{\partial^{N_i+1}\mathscr{F}(z_i,\xi,\eta)}{\partial\eta^{N_i+1}} \right| \right], \tag{2.41}$$

where $\mathscr{F}(z_i,x,y) = e^{-(\frac{\omega}{2}+\frac{\omega}{2}z_i)^2(x-y)^2}$ with z_i being the Gaussian quadrature points and δ is defined in (2.2.2). Now, by factorizing (2.38) and using (2.39), one arrives at the desired error bound of ϵ_2:

$$\epsilon_2 \leq \frac{\omega}{\sqrt{\pi}} c_2(N_{q_1}, b, N^{\frac{1}{3}}), \tag{2.42}$$

with

$$\begin{aligned} c_2(N_{q_1}, b, N^{\frac{1}{3}}) = N_{q_1} \sup_{1\leq i\leq N_{q_1}} \Bigg(& |w_i| \sum_{j}^{I_{\mu\nu}} \sum_{j'}^{I_{\kappa\lambda}} c_j c_{j'} \\ & (2b)^4 \|f_{\mu\nu}^{(j)}\|_{\infty,[-b,b]}^2 \|f_{\kappa\lambda}^{(j')}\|_{\infty,[-b,b]}^2 \\ & \sup_{1\leq n_1,m_1\leq N_i} \left((1 + N_i \alpha_{n_1,m_1}^{(i)} + (N_i \alpha_{n_1,m_1}^{(i)})^2) \beta_i \right) \Bigg). \end{aligned} \tag{2.43}$$

□

Throughout this study, we maintain a constant number of quadrature points N_{q_2}, for evaluating (2.27). It is important to note that our analysis does not focus on varying the parameter N_{q_2} , as it remains fixed.

As we notice here, the approximation error bound depends on the value of ω, the number of quadrature points N_{q_1}, the number of interpolation points N, the regularity of the function f , the Gaussian-type functions and the dimension of the hypercube b. It is worth noting that for fixed values of ω and b, $c_1(\omega, N_{q_1}, b) \to 0$ when $N_{q_1} \to \infty$ and $c_2(N_{q_1}, b, N^{\frac{1}{3}}) \to 0$ when $N^{\frac{1}{3}} \to \infty$.

2.3.4 A new decomposition of the two-electron integrals tensor through a tensorized approximation approach

As already discussed in the introduction, one of the main steps in many methods in quantum chemistry involves the application of the two-electron integrals tensor $\mathcal{B}^{lr} \in \mathbb{R}^{N_b\times N_b\times N_b\times N_b}$ to a vector with N_b^2 elements or a set of such vectors. To perform efficiently this contraction operation, we introduce in this section a factorized representation of the fourth-order two-electron integrals tensor $\mathcal{B}^{lr}$ that expands the factorized representation of its elements summarized in Theorem 2. We show also that the obtained tensorized structure is beneficial to accelerate contraction operations involved in HF calculations.

Factorized expression of $\mathcal{B}^{lr}$

In what follows we derive the factorized representation of $\left(\mathbf{B}^{lr}\right)^{<2>}$(mode-(1:2) matricization of $\mathcal{B}^{lr} \in \mathbb{R}^{N_b \times N_b \times N_b \times N_b}$). We slightly modify the expression of the approximation of the two-electron integrals (see Theorem 2) by changing the order of summation to obtain:

$$\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda) = \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \left[\sum_{\substack{n_1,m_1,\\ \cdots n_3,m_3=1}}^{N_i} \left(\sum_{j=1}^{I_{\mu\nu}} c_j \prod_{l=1}^{3} \tilde{\mathbf{W}}^{(i,l)}_{\mu\nu}(j;n_l) \right) \prod_{l=1}^{3} \alpha^{(i)}_{n_l m_l} \left(\sum_{j'=1}^{I_{\kappa\lambda}} c_{j'} \prod_{l=1}^{3} \tilde{\mathbf{W}}^{(i,l)}_{\kappa\lambda}(j';m_l) \right) \right]. \tag{2.44}$$

We introduce the matrices $\mathbf{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$ with single entries $\sum_{j=1}^{I_{\mu\nu}} c_j \prod_{l=1}^{3} \tilde{\mathbf{W}}^{(i,l)}_{\mu\nu}(j;n_l)$, $\mu,\nu \in [N_b]$, $n_l \in [N_i]$, $l \in \{1,2,3\}$ such that $\left(\mathbf{B}^{lr}_{LTEI-TA}\right)^{<2>}$ writes

$$\left(\mathbf{B}^{lr}_{LTEI-TA}\right)^{<2>} = \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \mathbf{M}_i^{TA} (\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i) \left(\mathbf{M}_i^{TA}\right)^* \in \mathbb{R}^{N_b^2 \times N_b^2}, \tag{2.45}$$

and

$$\mathbf{M}_i^{TA}(\overline{\mu,\nu};\overline{n_1,n_2,n_3}) = \sum_{j=1}^{I_{\mu\nu}} c_j \prod_{l=1}^{3} \tilde{\mathbf{W}}^{(i,l)}_{\mu\nu}(j;n_l), i \in [N_{q_1}]. \tag{2.46}$$

Fast evaluation of tensor products

In practice, it is sufficient to compute the matrix $\mathbf{M}_i^{TA}$ with the largest number of interpolation points, denoted as N. This matrix is represented by $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$. Notably, the remaining matrices $\mathbf{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$, where $N_i^3 \leq N$ and $i \in [N_{q_1}]$, share common elements with $\mathbf{M}_{TA,max}$. For example, given the two following matrices $\mathbf{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$ and $\mathbf{M}_j^{TA} \in \mathbb{R}^{N_b^2 \times N_j^3}$ with $N_i < N_j$ and $i,j \in [N_{q_1}]$, we have:

$$\mathbf{M}_i^{TA}(\overline{\mu,\nu};\overline{n_1,n_2,n_3}) = \mathbf{M}_j^{TA}(\overline{\mu,\nu};\overline{n_1,n_2,n_3}), n_1,n_2,n_3 \in [N_i]. \tag{2.47}$$

This can also be illustrated in Figure 2.2. Therefore, the storage complexity for storing $\mathbf{M}_{TA,max}$ is $\mathcal{O}(NN_b^2)$.

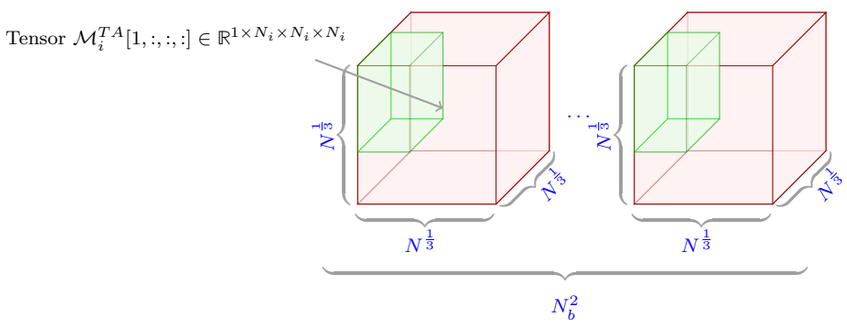


Figure 2.2: The tensor $\mathcal{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N^{\frac{1}{3}} \times N^{\frac{1}{3}} \times N^{\frac{1}{3}}}$ is obtained through the tensor folding of the matrix $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$.

By doing so, we can extract $\mathcal{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i \times N_i \times N_i}$ tensors that we unfold back to matrices $\mathbf{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$ through mode-1 matricization. We can exploit the tensorized structure of the factorized long-range TEI tensor in Equation (2.45) to reduce the application cost of the product between the tensorized form $\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i \in \mathbb{R}^{N_i^3 \times N_i^3}$ and $\mathbf{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$ from $\mathcal{O}\left(N_i^6 N_b^2\right)$ to $\mathcal{O}\left(N_i^4 N_b^2\right)$. This complexity reduction is achieved as follows: given the Definition 1.16, the product $\left(\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i\right)\left(\mathbf{M}_i^{TA}\right)^*$ can be defined entry-wise by

$$\begin{aligned}\left(\left(\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i\right)\left(\mathbf{M}_i^{TA}\right)^*\right)(n;j) &= \sum_{m_1,m_2,m_3=1}^{N_i} \left(\prod_{l=1}^{3} \mathbf{A}_i(n_l;m_l)\right)\left(\mathbf{M}_i^{TA}\right)^*(m_1,m_2,m_3;j) \\ &= \sum_{m_1,m_2,m_3=1}^{N_i} \left(\prod_{l=1}^{2} \mathbf{A}_i(n_l;m_l)\right)\left(\mathbf{A}_i\left(\mathbf{M}_i^{TA}\right)^{(4)}\right)(n_3;m_1,m_2,j),\end{aligned} \tag{2.48}$$

where $\left(\mathbf{M}_i^{TA}\right)^{(4)}$ is the mode-4 matricization of the fourth order tensor $\mathcal{M}_i^{TA} \in \mathbb{R}^{N_b^2 \times N_i \times N_i \times N_i}$. From (2.48), we notice that we need to perform three times the matrix-matrix products of sizes $N_i \times N_i$ and $N_i \times N_i^2 N_b^2$, leading to an overall time complexity of $\mathcal{O}\left(3N_i^4 N_b^2\right) \sim \mathcal{O}\left(N_i^4 N_b^2\right)$. If we want to compute the whole tensor, we need to sum over $i \in [N_{q_1}]$ which yields to a complexity of $\mathcal{O}\left(N^{\frac{4}{3}} N_b^2\right)$.

In practical applications, it is unnecessary to compute the complete two-electron integrals tensor; instead, maintaining its tensorized structure allows for efficient matrix operations when applying it to vectors or matrices. Further insight into a specific application case will be provided in Section 2.5.

An important point when implementing these tensor product evaluations is that the presented method can benefit from BLAS operations [24]. Indeed, (2.48) can be interpreted as the application of a sequence of products of permutation matrices and block-diagonal matrices (with the same blocks $\mathbf{A}_i$ along the diagonal) to $\left(\mathbf{M}_i^{TA}\right)^{(k)}, k \in \{2,3,4\}$. Matrix-vector products with block-diagonal matrices of this form can be numerically reformulated as matrix-matrix products between one of these diagonal blocks and a matrix composed of the concatenation of subvectors of the original one [18]. Since matrix-matrix products can be performed more efficiently than matrix-vector products using BLAS routines (namely BLAS-3 instead of BLAS-2), this optimization results in efficient implementations. In our case, we have even larger concatenation of subvectors because we apply these tensor products to matrices (not simply vectors), resulting in even better exploitation of BLAS-3 routines.

2.4 Long-range two-electron integrals tensor factorization through Fast Multipole Methods

Moving forward, we will provide a brief overview of Fast Multipole Methods (FMM) and its relevance to our problem. This will be preceded by a demonstration of the *asymptotic smoothness* of our kernel. Furthermore, we will give a comparison between the LTEI-TA and LTEI-FMM methodologies for approximating $\mathcal{B}^{lr}$, highlighting their resemblances and distinctions. First, we start by providing the definition of asymptotically smooth kernel.

Definition 27 (Definition 5.1 in [15])**.** A kernel $K(.,.) \quad : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is said to be asymptotically smooth if there exist two constants c_1, c_2 and a singularity degree $\sigma \in \mathbb{N}_0$ such that $\forall z \in \{\boldsymbol{x}_l, \boldsymbol{y}_l\} \in \mathbb{R}, \forall n \in \mathbb{N}_0, \forall \boldsymbol{x} \neq \boldsymbol{y}$,

$$\left| \frac{\partial^n}{\partial z^n} K(\boldsymbol{x}, \boldsymbol{y}) \right| \leq n! c_1 \left(c_2 \left\| \boldsymbol{x} - \boldsymbol{y} \right\|\right)^{-n-\sigma}.$$

Based on this property, efficient hierarchical schemes can be derived for the evaluation of N-body problems involving asymptotically smooth kernels.

2.4.1 Fast Multipole Methods

Considering two point clouds with $N_{\mathbf{X}}, N_{\mathbf{Y}}$ points, where we denote these clouds by $\{\boldsymbol{x}_n\}_{n=1}^{N_{\mathbf{X}}}, \{\boldsymbol{y}_n\}_{n=1}^{N_{\mathbf{Y}}} \subset \mathbb{R}^3$ (whose elements are referred to as 3D points or *particles*), $q : \{\boldsymbol{y}_n\}_{n=1}^{N_{\mathbf{Y}}} \to \mathbb{R}$ and an asymptotically smooth function $K : (\mathbb{R}^3 \times \mathbb{R}^3) \backslash \{\mathbf{0}\} \to \mathbb{R}$, one may express the associated N-body problem as the computation of $p : \{\boldsymbol{x}_n\}_{n=1}^{N_{\mathbf{X}}} \to \mathbb{R}$ such that

$$p(\boldsymbol{x}) := \sum_{\boldsymbol{y} \in \{\boldsymbol{y}_n\}_{n=1}^{N_{\mathbf{Y}}}} K(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{y}). \tag{2.49}$$

Computing p naively requires $\mathcal{O}(N^2)$ floating point operations, with $N = \max\{N_{\mathbf{X}}, N_{\mathbf{Y}}\}$. Thanks to hierarchical methods, such as *hierarchical matrices* or *Fast Multipole Methods* (FMM), which is a fast and a popular, initially introduced by Greengard and Rokhlin in 1985 [38], this complexity can be reduced to $\mathcal{O}(N\, logN)$ or even $\mathcal{O}(N)$ (but at the cost of an error we can control). These methods rely on decompositions of $\{\boldsymbol{x}_n\}_{n=1}^{N_{\mathbf{X}}}$ and $\{\boldsymbol{y}_n\}_{n=1}^{N_{\mathbf{Y}}}$ into groups of particles whose interaction can be efficiently performed through low-rank matrix approximations if their distance is sufficiently large compared to their radius. For

non-oscillatory kernels K, FMMs are able to reach the $\mathcal{O}(N)$ complexity, so that they are attractive algorithm for efficiently solving N-body problems.

Among the different formulation of FMMs, we seek for particular features needed for our application case. Indeed, the method has to:

- perform efficiently (actually in a linear time with respect to the number of points) on highly non-uniform point distributions, such as the three-dimensional Chebyshev grids,
- handle the kernel K (which is non-standard kernel in the FMM community),
- be able to reach the precision required in realistic chemistry applications.

2.4.2 Application of Fast Multipole Methods to two-electron integrals

First, in order to exploit FMM on the two-electron integrals, one has to check that the underlying kernel is asymptotically smooth. In our case, we want the FMM to act on the long-range kernel $K(\boldsymbol{x},\boldsymbol{y}), \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3$ (see (2.19)), which leads us to demonstrate the result of Proposition 2.4.1.

Proposition 2.4.1. *$K(\boldsymbol{x},\boldsymbol{y}) = \frac{erf(\omega\|\boldsymbol{x}-\boldsymbol{y}\|))}{\|\boldsymbol{x}-\boldsymbol{y}\|}, \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3, 0 \leq \omega < \infty$ is asymptotically smooth.*

Proof. Given the function $K(\boldsymbol{x},\boldsymbol{y}) = \frac{erf(\omega\|\boldsymbol{x}-\boldsymbol{y}\|))}{\|\boldsymbol{x}-\boldsymbol{y}\|}, \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^3, 0 \leq \omega < \infty$, we want to evaluate the function's partial derivative upper bound with respect to $x_1 \in \mathbb{R}$ such that $\forall n \in \mathbb{N}_0, \forall \boldsymbol{x} \neq \boldsymbol{y}$, the nth derivative of $K(\boldsymbol{x},\boldsymbol{y})$ with respect to x_1 writes

$$\frac{\partial^n}{\partial x_1^n} K(\boldsymbol{x},\boldsymbol{y}) = \frac{\partial^n}{\partial x_1^n}\left(\frac{2}{\sqrt{\pi}} \int_0^\omega exp\left(-s^2 \left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2\right) ds\right) \tag{2.50}$$

$$= \frac{2^{n+1}}{\sqrt{\pi}} n! \int_0^\omega \sum_{k=0}^{\left[\frac{n}{2}\right]} \frac{(-1)^{n-2k} 2^{-2k} (x_1-y_1)^{n-2k}}{k!(n-2k)!} s^{2n-2k} exp\left(-s^2 \left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2\right) ds. \tag{2.51}$$

If n is even, the term under the integral in (2.50) is positive. Otherwise, it is either negative or positive. Therefore, (2.50) can be bounded by the absolute value of the nth derivative of the Coulomb potential that writes:

$$\frac{\partial^n}{\partial x_1^n} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} = \frac{2^{n+1}}{\sqrt{\pi}} n! \int_0^\infty \sum_{k=0}^{\left[\frac{n}{2}\right]} \frac{(-1)^{n-2k} 2^{-2k} \left(x_1-y_1\right)^{n-2k}}{k!(n-2k)!} s^{2n-2k} exp\left(-s^2 \left\|\boldsymbol{x}-\boldsymbol{y}^2\right\|\right) ds, \tag{2.52}$$

and

$$\frac{\partial^n}{\partial x_1^n} K(\boldsymbol{x},\boldsymbol{y}) \leq \left| \frac{\partial^n}{\partial x_1^n} \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|} \right|. \tag{2.53}$$

Since $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}$ is asymptotically smooth [7, 40], this shows that $K(\boldsymbol{x},\boldsymbol{y})$ is also asymptotically smooth. This proof applies for all the other directions. □

Hence, thanks to the asymptotically smooth behavior of K, FMM can be applied to this kernel and the far field contribution of the N-body problem can be efficiently approximated, especially by exploiting polynomial interpolation. Similar to the previous sections, we consider the finite six-dimensional integral $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$ defined in (2.7) on a truncated computational box $[-b,b]^3 \times [-b,b]^3, b \in \mathbb{R}$ as follows:

$$\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda) \;=\; \int_{[-b,b]^3} \int_{[-b,b]^3} g_{\mu\nu}(\boldsymbol{x}) K(\boldsymbol{x},\boldsymbol{y}) g_{\kappa\lambda}(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}. \tag{2.54}$$

Instead of applying Gaussian quadrature rule on the kernel $K(\boldsymbol{x},\boldsymbol{y})$ as we did in the previous Section 2.3, we use Chebyshev polynomials evaluated in a six-dimensional Chebyshev

grid, the low-rank approximation of $K(\boldsymbol{x},\boldsymbol{y})$ can be written, as explained in [29], as follows:

$$K(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{N} L(\boldsymbol{x}_i,\boldsymbol{x}) \underbrace{\sum_{j=1}^{N} K(\boldsymbol{x}_i,\boldsymbol{y}_j) L(\boldsymbol{y}_j,\boldsymbol{y})}_{N\text{-body problem as in Eq. 2.49}}, \tag{2.55}$$

where N is the total number of Chebyshev interpolation points (we use the same N as the one introduced in Section 2.3), $\boldsymbol{x}_i = (\boldsymbol{x}_{i_1},\boldsymbol{x}_{i_2},\boldsymbol{x}_{i_3})$ and $\boldsymbol{y}_i = (\boldsymbol{y}_{i_1},\boldsymbol{y}_{i_2},\boldsymbol{y}_{i_3}) \in \mathbb{R}^{3N}$, for $i \in \{1,2,\ldots,N\}$, $i_l, j_l \in [N]\,, l \in \{1,2,3\}$. We also have:

$$L(\boldsymbol{x}_i,\boldsymbol{x}) = L^{(1)}\left(\boldsymbol{x}_{i_1},\boldsymbol{x}_1\right) L^{(2)}\left(\boldsymbol{x}_{i_2},\boldsymbol{x}_2\right) L^{(3)}\left(\boldsymbol{x}_{i_3},\boldsymbol{x}_3\right). \tag{2.56}$$

$$L^{(l)}\left(\boldsymbol{x}_{i_l},\boldsymbol{x}_l\right) = \frac{1}{N^{\frac{1}{3}}} + \frac{2}{N^{\frac{1}{3}}} \sum_{k=2}^{N^{\frac{1}{3}}} T_k(\boldsymbol{x}_{i_l}) T_k(\boldsymbol{x}_l),\ l \in \{1,2,3\}\,. \tag{2.57}$$

One may notice that the Equation (2.57) appears as a simple reformulation of the interpolation presented in Definition 26, combining the equation (2.9) and the equation (2.10). The important point here is that we want the kernel to explicitly appear (evaluated on Chebyshev interpolation nodes) in the expression, so that a FMM algorithm can be derived, following [18, 29]. Chebyshev polynomials are used here as interpolation basis and were already defined in Definition 26. The long-range two-electron integrals in (2.7) can be written as follows:

$$\begin{aligned}\mathcal{B}^{lr}_{LTEI-FMM}(\mu;\nu;\kappa;\lambda) &= \sum_{i=1}^{N} \underbrace{\int_{[-b,b]^3} g_{\mu\nu}(\boldsymbol{x}) L(\boldsymbol{x}_i,\boldsymbol{x}) d\boldsymbol{x}}_{\boldsymbol{z}_{\mu\nu}(i)} \left(\sum_{j=1}^{N} K(\boldsymbol{x}_i,\boldsymbol{y}_j) \underbrace{\int_{[-b,b]^3} g_{\kappa\lambda}(\boldsymbol{y}) L(\boldsymbol{y}_j,\boldsymbol{y}) d\boldsymbol{y}}_{\boldsymbol{z}_{\kappa\lambda}(j)} \right) \\ &= \sum_{i=1}^{N} \boldsymbol{z}_{\mu\nu}(i) \left(\sum_{j=1}^{N} K(\boldsymbol{x}_i,\boldsymbol{y}_j) \boldsymbol{z}_{\kappa\lambda}(j) \right).\end{aligned} \tag{2.58}$$

Equation (2.58) can be written in matrix formulation as follows for fixed $\mu,\nu,\kappa,\lambda \in [N_b]$:

$$\mathcal{B}^{lr}_{LTEI-FMM}(\mu;\nu;\kappa;\lambda) = \boldsymbol{z}_{\mu\nu}\mathbf{K}\boldsymbol{z}^*_{\kappa\lambda}, \boldsymbol{z}_{\mu\nu}, \boldsymbol{z}_{\kappa\lambda} \in \mathbb{R}^N, \mathbf{K} \in \mathbb{R}^{N\times N}, \tag{2.59}$$

with $\mathbf{K}(\boldsymbol{x}_i;\boldsymbol{y}_j) = \frac{erf(\omega\left\|\boldsymbol{x}_i-\boldsymbol{y}_j\right\|))}{\left\|\boldsymbol{x}_i-\boldsymbol{y}_j\right\|}, i,j \in [N]\,.$

The last term into parenthesis in (2.58) corresponds to an N-body problem as in Equation (2.49), whose evaluation can be performed in $\mathcal{O}(N)$ FLOPS using FMM. One may notice that the FMM accuracy can be chosen accordingly to the interpolation error in equation (2.58). For all $\mu,\nu,\kappa,\lambda \in [N_b]$, the factorized representation of the mode-(1,2) matricization of the fourth-order tensor $\mathcal{B}^{lr}_{LTEI-FMM}$ (2.58) is then given by:

$$\left(\mathbf{B}^{lr}_{LTEI-FMM}\right)^{<2>} = \mathbf{M}^{FMM}\mathbf{K}\left(\mathbf{M}^{FMM}\right)^* \in \mathbb{R}^{N_b^2\times N_b^2},\ \mathbf{M}^{FMM} \in \mathbb{R}^{N_b^2\times N}, \tag{2.60}$$

with $\mathbf{M}^{FMM}[\mu\nu,:] = \boldsymbol{z}_{\mu\nu} \in \mathbb{R}^N$, for $\mu,\nu \in [N_b]$. Hence, the entire computation of (2.60) requires the application of the FMM method to each column of $\mathbf{M}^{FMM}$, the overall evaluation complexity of FMM becomes $\mathcal{O}(N \times N_b^2)$ to compute $\mathbf{K}\left(\mathbf{M}^{FMM}\right)^*$.

Remark 2. The FMM formulation we opted for relies on precomputations (at a linear cost with respect to the number of particles) for the construction of low-rank approximations (see Section 2.4.1) that depends only on the particle distribution. Because the interpolation points are the same for each $\boldsymbol{z}_{\kappa\lambda}(j)$, our particle distributions do not change, so these precomputations can be performed only once and reused for each FMM application.

To summarize, LTEI-FMM is the combination of two steps:

- a **global** interpolation of the kernel, mainly introduced to switch integrals to basis functions: Chebyshev polynomials basis,
- the application of a fast linear-complexity summation method for N-body problems to the induced highly non-uniform distribution of tensorized Chebyshev nodes (possibly relying itself on **local** polynomial interpolations, such as in [29]).

2.4.3 Similarities and differences

In table 2.1 we summarize the approximated expressions of (2.7) obtained through LTEI-TA and LTEI-FMM approaches.

Table 2.1: Factorization of TEI

Approaches	LTEI-TA	LTEI-FMM
Distribution	N Chebyshev points	N Chebyshev points
Entry-wise evaluation	$\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu;\kappa;\lambda)$ $:= \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q1}} (w_i \sum_{j=1}^{I_{\mu\nu}} \sum_{j'=1}^{I_{\kappa\lambda}} c_j c_{j'} \mathbf{F}_i(j;j'))$	$\mathcal{B}^{lr}_{LTEI-FMM}(\mu;\nu;\kappa;\lambda)$ $:= \boldsymbol{z}_{\mu\nu} \mathbf{K} \boldsymbol{z}^*_{\kappa\lambda}$.
Factorized representation	$\left(\mathbf{B}^{lr}_{LTEI-TA}\right)^{<2>}$ $:= \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q1}} w_i \mathbf{M}_i^{TA} (\bigotimes_{l=1}^{3}{}_K \mathbf{A}_i) \left(\mathbf{M}_i^{TA}\right)^* \in \mathbb{R}^{N_b^2 \times N_b^2}$.	$\left(\mathbf{B}^{lr}_{LTEI-FMM}\right)^{<2>}$ $:= \mathbf{M}^{FMM} \mathbf{K} \left(\mathbf{M}^{FMM}\right)^* \in \mathbb{R}^{N_b^2 \times N_b^2}$.

We discuss here the differences and similarities between both approaches. On one hand, for TA approach, we start by applying a change of variable to the long-range kernel $K(\boldsymbol{x},\boldsymbol{y})$ (2.19) in order to remove the term $\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}$, then we apply one-dimensional Gaussian quadrature (see (2.20)) with N_{q_1} quadrature points. In addition to that, we apply two-dimensional Chebyshev interpolation on 2D smooth exponential functions which yields a tensorized form obtained in (2.29), (2.45). Thus, we need to evaluate $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$ which involves the evaluation of one-dimensional integrals over $[-b,b]$. On the other hand, when applying interpolation directly on the original kernel K, one ends up with a three dimensional N-body problem that can be efficiently handled using FMM approach given that our kernel is asymptotically smooth. Thus, we need to compute $\mathbf{M}^{FMM} \in \mathbb{R}^{N_b^2 \times N}$ which involves also the evaluation of one-dimensional integrals over $[-b,b]$. The similarities between both approaches consist in employing Chebyshev interpolation with the same total number of interpolation points N.

Remark 3. One may mention that for low level optimisations (such as explicit formula for the polynomials or fast FFT-based assembling of the interpolation coefficients), we opted for slightly different interpolation nodes in the two methods. Indeed, *Gauss-Chebyshev-Lobatto* nodes are used for LTEI-TA method while *Chebyshev* nodes are used for LTEI-FMM. These last points are defined as (showing only $\boldsymbol{x}_{i_l}$ expression):

$$\boldsymbol{x}_{i_l} = \cos\left(\frac{2k-1}{2N^{\frac{1}{3}}}\pi\right), k \in \left[N^{\frac{1}{3}}\right], l \in \{1,2,3\}. \tag{2.61}$$

However, for both cases, the same number of interpolation nodes is considered for a given targeted precision, $N^{\frac{1}{3}}$ per direction, so that this detail does not impact the complexity estimates and the comparison between them.

2.5 Application to electronic structure calculations

We describe in what follows an application case for the long-range two-electron integrals tensor using LTEI-TA as well as LTEI-FMM. The application case consists in the construction of the long-range Coulomb matrix arising in the Hartree-Fock calculations for solving the Hartree-Fock equations using the iterative Self Consistent Field (SCF) method as detailed in [61, 79, 82, 117, 136], the evaluation of the long-range exchange matrix is outlined in Appendix .2 . This approach is often used in the context of range-separated hybrid approximation approaches [32, 55, 96]. We define in the following the long-range Coulomb matrix in the molecular spatial-orbital basis set $\{\phi_i\}_{1\leq i\leq d}$ defined in (1.90). In this basis, the Coulomb long-range integral reads

$$\mathbf{J}^{lr}(i;j) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} K(\boldsymbol{x},\boldsymbol{y}) \sum_{i=1}^{d} |\phi_i(\boldsymbol{x})|^2 \sum_{j=1}^{d} |\phi_j(\boldsymbol{y})|^2 \, d\boldsymbol{x} d\boldsymbol{y} \tag{2.62}$$

$$= \sum_{\mu,\nu,\kappa,\lambda=1}^{N_b} \sum_{i,j=1}^{d} C_{i\mu} C_{i\nu} \left(\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} K(\boldsymbol{x},\boldsymbol{y}) g_{\mu\nu}(\boldsymbol{x}) g_{\kappa\lambda}(\boldsymbol{y}) C_{j\kappa} C_{j\lambda} d\boldsymbol{x} d\boldsymbol{y}\right), \tag{2.63}$$

with $C_{i\mu}$ being the coefficients of the expansion of $\{\phi_i\}_{1\leq i\leq d}$ with $d \sim N_b$.

Let us define the rectangular matrix $\mathbf{Q} \in \mathbb{R}^{d\times N_b^2}$ with entries $\mathbf{Q}(i;\mu,\nu) = q_{i\mu}q_{i\nu}$ such that $\mathbf{J}^{lr}$ writes in matrix notation as

$$\mathbf{J}^{lr} = \mathbf{Q}\left(\mathbf{B}^{lr}\right)^{<2>}\mathbf{Q}^* \in \mathbb{R}^{d\times d}, \tag{2.64}$$

where $\left(\mathbf{B}^{lr}\right)^{<2>} \in \mathbb{R}^{N_b^2\times N_b^2}$ is the mode-(1,2) matricization of $\mathcal{B}^{lr}$. A naive approach to evaluate (2.64), given ω, the matrix $\mathbf{Q} \in \mathbb{R}^{d\times N_b^2}$, and the long-range two-electron integrals $\left(\mathbf{B}^{lr}\right)^{<2>}$, is to first compute the matrix product $\mathbf{B}^{lr}\mathbf{Q}^*$ and then perform $\mathbf{Q}\left(\mathbf{B}^{lr}\mathbf{Q}^*\right)$. The last has an arithmetic cost of $\mathcal{O}(N_b^4 d)$. Given a truncated computational box $[-b,b]^3$, one can use the factorized structure $\left(\mathbf{B}^{lr}_{LTEI-TA}\right)^{<2>}$ defined in (2.45) or $\left(\mathbf{B}^{lr}_{LTEI-FMM}\right)^{<2>}$ defined in (2.60) to evaluate (2.64) efficiently. Given the two approximation approaches (LTEI-TA and LTEI-FMM), we arrive at the following matrix representations:

$$\mathbf{J}^{lr}_{LTEI-TA} = \frac{\omega}{\sqrt{\pi}}\sum_{i=1}^{N_{q_1}} w_i \left(\mathbf{QM}_i^{TA}\right)\left(\bigotimes_{l=1}^{3}{}_K\mathbf{A}_i\right)\left(\mathbf{QM}_i^{TA}\right)^* \tag{2.65}$$

and

$$\mathbf{J}^{lr}_{LTEI-FMM} = \left(\mathbf{QM}^{FMM}\right)\mathbf{K}\left(\mathbf{QM}^{FMM}\right)^*. \tag{2.66}$$

We provide an overview of the storage complexities achieved by the LTEI-TA and LTEI-FMMmethods for evaluating the long-range two-electron integrals tensor element-wise as well as the storage complexities when the latter is applied to compute the long-range Coulomb matrix defined in Equation (2.64).

Table 2.2: Storage complexity comparison

	LTEI-TA	LTEI-FMM
Element-wise TEI	$\mathcal{O}(N^{\frac{1}{3}}N_{q_1}(N^{\frac{1}{3}}+I_{\mu\nu}+I_{\kappa\lambda}))$	$\mathcal{O}(N)$
Application (2.64)	$\mathcal{O}(N^{\frac{2}{3}}(dN^{\frac{1}{3}}+N_{q_1}))$	$\mathcal{O}(N(1+d))$

The storage complexity of the element-wise evaluation for LTEI-FMM is a consequence of (2.59), i.e. linear with regard to the number of interpolation points N. The storage complexities for the evaluation of (2.64) are obtained as follows: for LTEI-TA approach

1. Instead of forming all matrices $\mathbf{QM}_i^{TA}$ for $i \in [N_{q_1}]$, we form only (as explained in Section 2.3.4) $\mathbf{QM_{TA,max}}$ that requires $\mathcal{O}(dN)$ storage.

2. As discussed before, we keep $\left(\bigotimes_{l=1}^{3}{}_K\mathbf{A}_i\right)$ in tensorized form. Hence, forming all coefficient matrices $\mathbf{A}_i$ of size $N_i \times N_i$, for $i \in [N_{q_1}]$ requires $\mathcal{O}(\sum_{i=1}^{N_{q_1}} N_i^2) \sim \mathcal{O}(N_{q_1}N^{\frac{2}{3}})$ storage.

So in total, the storage complexity is $\mathcal{O}(N^{\frac{2}{3}}(dN^{\frac{1}{3}}+N_{q_1}))$. For LTEI-FMM approach

1. Forming $\mathbf{QM}^{FMM}$ requires $\mathcal{O}(dN)$ of storage.

2. Forming $\mathbf{K}$ requires $\mathcal{O}(N)$ of storage.

So in total, the storage complexity is $\mathcal{O}(N(1+d))$. According to Table.2.2, the storage demand for this evaluation seems lower (in order) for LTEI-TA compared to LTEI-FMM. However, we cannot conclude on the best method in terms of storage complexity since N_{q_1} and $N^{\frac{1}{3}}$ depend on the value of ω and the chosen computational box $[-b,b]^3$. This motivates numerical comparisons between the two approaches for different parameters (see Section 2.7).

2.6 Compression techniques for the factorized long-range two-electron integrals tensor

2.6.1 Compression by using low-rank methods

One of the main precomputation steps required to obtain the factorized representation of $\left(\mathbf{B}^{lr}\right)^{<2>}$ is based on the evaluation of $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$ (resp. $\mathbf{M}^{FMM} \in \mathbb{R}^{N_b^2 \times N}$) matrix. This step tends to be expensive in terms of both computational and memory requirements for molecules of moderate size, as we consider in our experiments. In this section we address this problem by discussing different approaches to compress $\mathbf{M}_{TA,max}$, with some of these techniques potentially being applicable to $\mathbf{M}^{FMM}$. In many cases, the matrix $\mathbf{M}_{TA,max}$ is numerically low-rank as we will discuss in the numerical experiments section (see Figure 2.12). It is possible to reduce its dimensions by exploiting its low rank structure. Additionally, when employing Gaussian-type basis functions, the compression can be extended even further through the *screening* technique [100]. This approach consists in simply discarding "negligible" pairs of Gaussian type basis functions as explained in 2.6.3. Methods for low-rank approximation, such as truncated SVD, can also be directly employed on $\mathbf{M}_{TA,max}$. However, it's worth noting that this approach can be computationally demanding, with a complexity that scales as $\mathcal{O}(N_b^2 N \min\{N_b^2, N\})$. We introduce in this section a different compression method that exploits the Khatri-rao products and associated properties.

Let $\tilde{\mathbf{W}}^{(i,l)} \in \mathbb{R}^{N_b^2 I_{\mu\nu,max} \times N_i}$ be defined by for $i \in [N_{q_1}]$:

$$\tilde{\mathbf{W}}^{(i,l)} = \begin{bmatrix} \tilde{\mathbf{W}}^{(i,l)}_{\mu\nu} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \} I_{\mu\nu} \times N_i \\ \} (I_{\mu\nu,max} - I_{\mu\nu}) \times N_i, \; I_{\mu\nu,max} = \max\limits_{1 \leq \mu,\nu \leq N_b} \{I_{\mu\nu}\}. \end{matrix} \tag{2.67}$$

Its low rank $R_{i,l}$ approximation can be written as:

$$\tilde{\mathbf{W}}^{(i,l)} \approx \mathbf{U}^{(i,l)} \left(\mathbf{V}^{(i,l)}\right)^*, \tag{2.68}$$

where $\mathbf{U}^{(i,l)} \in \mathbb{R}^{N_b^2 I_{\mu\nu,max} \times R_{i,l}}$ and $\left(\mathbf{V}^{(i,l)}\right)^* \in \mathbb{R}^{R_{i,l} \times N_i}$. Given the decomposition (2.68), Proposition 1.1.1 is used to obtain the following expression

$$\begin{aligned}
&\left(\diamond_{l=1}^3 (\tilde{\mathbf{W}}^{(i,l)})\right) \left(\bigotimes_{l=1}^3 {}_K \mathbf{A}_i\right) \left(\bullet_{l=1}^3 (\tilde{\mathbf{W}}^{(i,l)})^*\right) \\
&= \left(\diamond_{l=1}^3 (\mathbf{U}^{(i,l)} \mathbf{V}^{(i,l)})\right) (\mathbf{A}_i \otimes \mathbf{A}_i \otimes \mathbf{A}_i) \left(\bullet_{l=1}^3 (\mathbf{U}^{(i,l)} \mathbf{V}^{(i,l)})^*\right) \\
&= \left(\diamond_{l=1}^3 \mathbf{U}^{(i,l)}\right) (\bigotimes_{l=1}^3 {}_K \mathbf{V}^{(i,l)}) (\mathbf{A}_i \otimes \mathbf{A}_i \otimes \mathbf{A}_i) (\bigotimes_{l=1}^3 {}_K \left(\mathbf{V}^{(i,l)}\right)^*) \left(\bullet_{l=1}^3 (\mathbf{U}^{(i,l)})^*\right) \\
&= \left(\diamond_{l=1}^3 \mathbf{U}^{(i,l)}\right) \bigotimes_{l=1}^3 {}_K (\mathbf{V}^{(i,l)} \mathbf{A}_i \left(\mathbf{V}^{(i,l)}\right)^*) \left(\bullet_{l=1}^3 (\mathbf{U}^{(i,l)})^*\right).
\end{aligned} \tag{2.69}$$

By replacing the low rank approximation of the matrix $\tilde{\mathbf{W}}^{(i,l)}$ in the expression of $\left(\mathbf{B}^{lr}\right)^{<2>}_{LTEI-TA}$ in equation (2.45), we obtain:

$$\left(\mathbf{B}^{lr}_{LTEI-TA}\right)^{<2>} \approx \frac{\omega}{\sqrt{\pi}} \sum_{i=1}^{N_{q_1}} w_i \tilde{\mathbf{U}}_i \bigotimes_{l=1}^3 {}_K (\mathbf{V}^{(i,l)} \mathbf{A}_i \left(\mathbf{V}^{(i,l)}\right)^*) \tilde{\mathbf{U}}_i^*, \tag{2.70}$$

where $\tilde{\mathbf{U}}_i = \sum_{j=1}^{I_{\mu\nu,max}} c_j \mathcal{U}_i[j,:,:] \in \mathbb{R}^{N_b^2 \times \prod_{l=1}^3 R_{i,l}}$ with $\mathcal{U}_i \in \mathbb{R}^{I_{\mu\nu,max} \times N_b^2 \times \prod_{l=1}^3 R_{i,l}}$ the tensor folding of $\left(\diamond_{l=1}^3 \mathbf{U}^{(i,l)}\right) \in \mathbb{R}^{I_{\mu\nu,max} N_b^2 \times \prod_{l=1}^3 R_{i,l}}$. In practice, we compute only the matrix $\tilde{\mathbf{U}}_i \in \mathbb{R}^{N_b^2 \times (\max_{1 \leq i \leq N_{q_1}} \{\prod_{l=1}^3 R_{i,l}\})}$ with the maximum rank $\max\limits_{1 \leq i \leq N_{q_1}} \left\{\prod_{l=1}^3 R_{i,l}\right\}$ as discussed in Section 2.3.4. Through this compression, the storage complexity for the evaluation of the long-range Coulomb matrix in (2.64) can be reduced to $\mathcal{O}(R^{\frac{2}{3}}(N_{q_1} + dR^{\frac{1}{3}})$ where $R = \left(\max\limits_{1 \leq i \leq N_{q_1}} \left\{\prod_{l=1}^3 R_{i,l}\right\}\right)^3$.

2.6.2 Adaptive approach for the choice of the integration domain for Gaussian type basis functions

We discuss now an adaptive approach for the choice of the integration domain $[-b, b]$. For each pair of Gaussian functions, we identify its numerical support $[-b, b]$. We cluster together these numerical supports to obtain overall $N_{partitions}$ supports, $\{[-b_s, b_s]\}_{s\in[N_{partitions}]}$. For each pair of Gaussian functions $(\mu, \nu), \mu, \nu \in [N_b]$, we proceed as follows: Given the general Gaussian product rule (Definition 2.2.1), the product of two primitive Gaussian type functions $f^{(j)}_{\mu\nu}(x_l) = f^{(j_1)}_{\mu}(x_l) f^{(j_2)}_{\nu}(x_l)$ is:

$$f^{(j)}_{\mu\nu}(x_l) = f^{(j_1)}_{\mu}(x_l) f^{(j_2)}_{\nu}(x_l) = (x_l - r_l)^{p_{\mu_l}} (x_l - r'_l)^{p_{\nu_l}} e^{-\frac{\mu_{j_1}\nu_{j_2}}{\mu_{j_1}+\nu_{j_2}}(r_l - r'_l)^2} \sigma_{\mu_{j_1}\nu_{j_2}}(x_l), \tag{2.71}$$

where

$$\sigma_{\mu_{j_1}\nu_{j_2}}(x_l) = e^{-(\mu_{j_1}+\nu_{j_2})(x_l - \frac{\mu_{j_1} r_l + \nu_{j_2} r'_l}{\mu_{j_1}+\nu_{j_2}})^2}, j_1 \in [I_\mu], j_2 \in [I_\nu], \tag{2.72}$$

and (see (1.93))

$$f^{(j_1)}_{\mu}(x_l) = (x_l - r_l)^{p_{\mu_l}} e^{-\mu_{j_1}(x_l - r_l)^2} \text{ and } f^{(j_2)}_{\nu}(x_l) = (x_l - r'_l)^{p_{\nu_l}} e^{-\nu_{j_2}(x_l - r'_l)^2}, \tag{2.73}$$

with $j = (j_1, j_2) \in [I_{\mu\nu}]$, $I_{\mu\nu} = I_\mu I_\nu$, $l \in \{1, 2, 3\}$, $\mu, \nu \in [N_b]$. The numerical support $[-b, b]$ is chosen according to a cutoff threshold $\tau_{adaptive} > 0$ such that

$$\sigma_{\mu_{j_1}\nu_{j_2}}(x_l) \leqslant \tau_{adaptive}, l \in \{1, 2, 3\}. \tag{2.74}$$

To illustrate this adaptive approach, for a given pair (μ, ν), we represent in Figure 2.3 the exponential terms $\sigma_{\mu_{j_1}\nu_{j_2}}(x_1)$ in the expression (2.71) for $j \in [I_{\mu\nu}]$ with respect to the first direction (l=1). The exponential decay of these functions enables us to limit the range of the numerical grid according to a chosen threshold $\tau_{adaptive}$. Through this adaptive technique, Figure 2.4 illustrates the distribution of the numerical support (dimension b). Each bar represents the percentage of Gaussian function pairs (μ, ν) associated to the exponential terms $\sigma_{\mu_{j_1}\nu_{j_2}}(x_l)$ lying in the range $[-b, b]$. It is showed that the distribution depends on the molecule choice as well as the number of basis functions N_b.

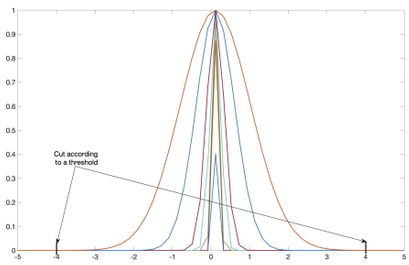


Figure 2.3: Identifying numerical supports of different pairs of Gaussian functions. Each color in the plot represents the exponential term $\sigma_{\mu_{j_1}\nu_{j_2}}(x_1)$. Here the selected numerical support is $[-4, 4]$.

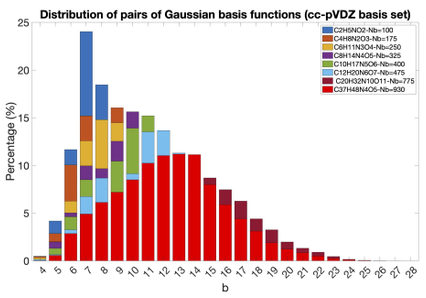


Figure 2.4: Distribution of numerical supports $[-b, b]$ for a given threshold $\tau_{adaptive} = 10^{-20}$. The x-axis shows the dimension b of the box, and the y-axis shows the percentage of the Gaussian function pairs.

The advantages of using this approach is that there is no need to fix in advance the size of the numerical box b since it is determined by the support of these Gaussian functions. Moreover, it is possible to reduce the storage demand since instead of storing the matrix $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$, smaller matrices of sizes $N^2_{b,s} \times N_s$ are stored, where $N^2_{b,s}$ are the pairs of Gaussian functions associated to the integration domain $[-b_s, b_s]$ and N_s is the maximum number of Chebyshev interpolation points in the interval $[-b_s, b_s]$. We must point out that by using this adaptive method, multiple tensor contraction calculations need to be performed to compute (2.65) which will depend on the number of partitions $\mathcal{P}_s$. This can be costly if we consider a sequential algorithm. However, this adaptive approach offers a possibility to parallelize the evaluation of (2.65).

2.6.3 Compression through screening

It is possible to further reduce the dimensions of $\mathbf{M}_{TA,max}$ by exploiting the properties of the Gaussian type basis functions. In fact, given the product of two-primitive Gaussians introduced in (2.71), we notice that $f^{(j)}_{\mu\nu}(x_l) = f^{(j)}_{\nu\mu}(x_l)$, for $j \in [I_{\mu\nu}], \mu, \nu \in [N_b]$ and $l \in \{1, 2, 3\}$. Therefore, there are only $\frac{N_b}{2}(N_b + 1)$ choices for N_b^2 combinations of μ and ν. We also apply the screening technique that is often used by chemists to reduce the computational cost of the evaluation of integrals [100]. From the Gaussian product rule (2.71), the higher the exponent of a primitive Gaussian, the faster the products with primitives from other centers decay with distance and the sooner they become negligible. Therefore, for large enough molecules, it is possible to discard a consistent number of pairs of primitive Gaussians which is illustrated in the numerical experiment section in Figure 2.12. In practice, we discard the primitive pair that satisfies the following condition for a given threshold $\tau_{screening}$:

$$e^{-\frac{\mu_{j_1}\nu_{j_2}}{\mu_{j_1}+\nu_{j_2}}\sum_{l=1}^{3}(r_l - r'_l)^2} \leq \tau_{screening}. \tag{2.75}$$

2.7 Numerical results

In this section, we evaluate numerically our novel method LTEI-TA[5] by using a prototype implementation in Julia language version 1.5.3. We also compare it with LTEI-FMM method using *defmm* library [18]. The *defmm* library is a C++ code[6] that is particularly well-suited for the two-electronic integrals context since it implements various important features with $\mathcal{O}(N)$ complexity on non-oscillatory kernels in both precomputation and application cost. More precisely, *defmm* is

- *kernel-independent*, meaning that the user has to provide only a routine evaluating $K(\boldsymbol{x}, \boldsymbol{y})$ to use the code and the handling of erf function can be added at minimal implementation effort,
- *adaptive*, meaning that the algorithm automatically adapts to the potential non-uniformity of the particle distribution. Similar performance was observed for *defmm* using non-oscillatory kernels applied on uniform and highly non-uniform distributions [18] (such as our tensorized Chebyshev grids),
- *convergent* for any asymptotically smooth kernel, including our kernel $K(\boldsymbol{x}, \boldsymbol{y})$ (see Proposition 2.4.1), as proven in [19].

An example of a call to *defmm* library is provided in Appendix .1. *defmm* is compiled using the intel C++ compiler (version 19.1.2.254) and FFTW3 (since *defmm* relies on FFTs for the far field compression/evaluation). We remind that the evaluation algorithm in LTEI-TA, which is written in Julia, is based on matrix-matrix products, performed with optimized BLAS operations (see Section 2.3.4) for the dense linear algebra computations. Hence, the effect of the programming language choice has a negligible impact for LTEI-TA. This justifies the comparison between c++ calls (*defmm*) and our implementation of LTEI-TA in Julia. We are also aware that results presented in the following correspond to prototypes in which we simply link *defmm* with outputs from our Julia code, regardless of further possible optimizations. All the calculations are carried out using Cleps cluster from Inria, Paris, France. This machine has 4 partitions. We use cpu-homogen partition which contains 20 nodes with hyper-threading such that we can allocate a maximum of 64 logical cores per node (Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz) with a memory of 6GB per core. We start always by the data initialization step which consists in reading input files generated from *quantum package*. These files contain molecular properties: number of atoms, number of basis functions, coordinates of the nuclei, basis set parameters. For all molecules we use the "cc-pVDZ" Gaussian basis set [94].

[5] https://github.com/sbadred/LTEI_TA.jl.git
[6] https://github.com/IChollet/defmm

2.7.1 Approximation error and computational cost

The following numerical results present the approximation errors with respect to different parameters ω, N, and N_{q_1}. We start by providing the approximation error for the element-wise evaluation of the long-range two-electron integrals tensor and then we provide the numerical error convergence obtained for the evaluation of the long-range Coulomb matrix as defined in (2.64) using both methods: LTEI-TA and LTEI-FMM.

Approximation error

First we provide convergence results of LTEI-TA method for the evaluation of the long-range two-electron integrals given in equation (2.29). For the following numerical tests, we consider small sized molecules : NH_3 and CO_2, where we represent the mean relative error of 10^3 randomly chosen elements from the tensor $\mathcal{B}^{lr}$. In Figure 2.5, on the left-hand side, we maintain a constant number of Chebyshev interpolation points N, while varying the value of N_{q_1}. On the right-hand side of Figure 2.5, we fix the number of quadrature points N_{q_1}, while varying N.

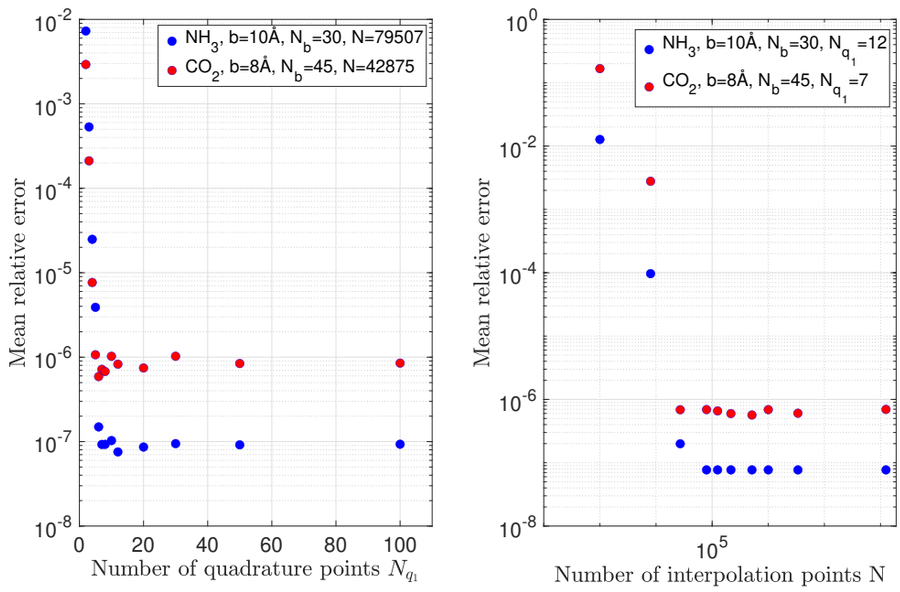
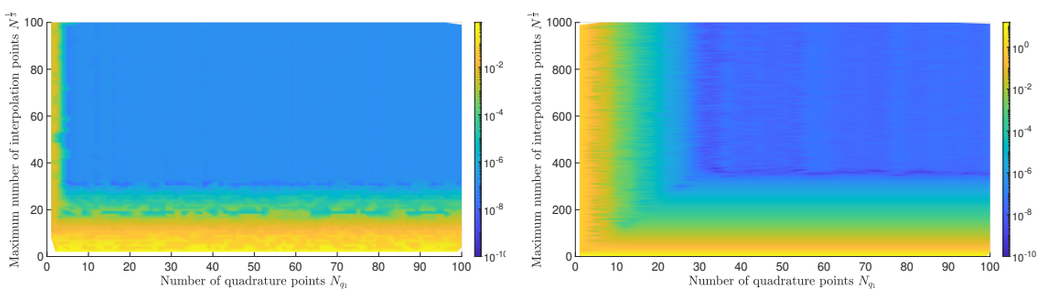


Figure 2.5: Approximation error of the long-range two-electron integrals using LTEI-TA,



(a) $\omega = 0.5$ (b) $\omega = 5$

Figure 2.6: Approximation error of the element-wise evaluation of the two-electron integrals (2.29) with respect to (#interpolation points per direction, #quadrature points) $\equiv (N^{\frac{1}{3}}, N_{q_1})$ for the optimal accuracy using NH_3 molecule in the cc-pVDZ basis set for different values of ω. The colorbar shows the mean relative approximation error.

In Figure 2.5, with fixed $\omega = 0.5$, we notice the fast convergence of the relative error towards the value of 10^{-7} for both subfigures such that the analytical results, generated from *quantum package*, and numerical results are in reasonably good agreement for both molecules. We note that the stagnation of the error is a consequence of the approximations

used (Chebyshev interpolation and Gaussian quadrature rule), hence in order to optimize our method for a desired accuracy, we need to find a good compromise between the parameters $N^{\frac{1}{3}}$ and N_{q_1}, as shown in Figure 2.6(b). Indeed, we note that in Figure 2.6(b), for each number of interpolation points, there is a number of quadrature points that allows to reach a small relative error (up to 10^{-10}). Another noteworthy aspect is that due to the truncation of the support of the primitive Gaussians, the error is constrained by b , which corresponds to the bounds of the integration box. This constraint is emphasized in the error bound provided in Proposition 2.3.1.

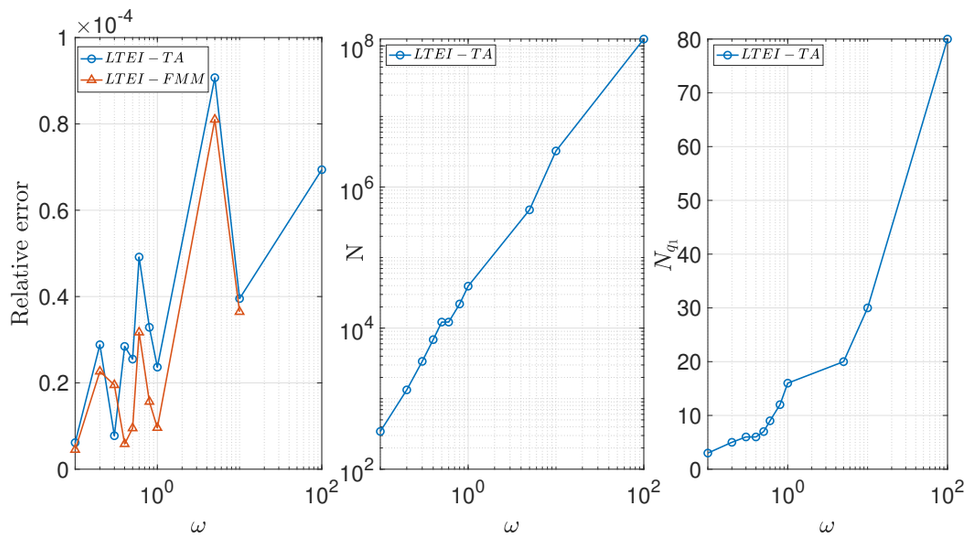


Figure 2.7: (Leftmost figure) The approximation error of the element-wise evaluation of the two-electron integrals with respect to ω for both approaches: LTEI-TA and LTEI-FMM. (Middle figure) The number of interpolation points N needed to reach the imposed accuracy (relative error smaller than 10^{-4}) with respect to ω. (Rightmost figure) The number of quadrature points N_{q_1} needed to reach the imposed accuracy (relative error smaller than 10^{-4}) with respect to ω.

Figure 2.7 displays the number of interpolation points N (middle figure) and the number of quadrature points N_{q_1} (rightmost figure) with respect to ω for computing a single entry of the long-range two-electron integrals tensor $\mathcal{B}^{lr}$ through LTEI-TA and LTEI-FMM approaches. The entry $\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)$ is chosen randomly and we impose that the relative error is smaller than 10^{-4}, where the relative error is defined as $\frac{\left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)-\mathcal{B}^{lr}_{LTEI-TA}(\mu;\nu\kappa;\lambda)\right|}{\left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)\right|}$ for LTEI-TA, and as $\frac{\left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)-\mathcal{B}^{lr}_{LTEI-FMM}(\mu;\nu;\kappa;\lambda)\right|}{\left|\mathcal{B}^{lr}(\mu;\nu;\kappa;\lambda)\right|}$ for LTEI-FMM, respectively. We observe that the number of interpolation points N and the number of quadrature points N_{q_1} needed to reach the desired accuracy grow with ω, as it can be seen in the middle and rightmost figures. This is explained by the fact that when $\omega \to \infty$, LTEI-TA needs to approximate a nearly singular kernel, which increases its cost. The leftmost figure also shows that the accuracy of LTEI-TA and LTEI-FMM for the evaluation of an element of $\mathcal{B}^{lr}$ is comparable for the same number of interpolation points N. This is because both approaches are based on Chebyshev interpolation. We note that the quadrature in LTEI-TA is chosen to be at least as precise as the interpolation and the FMM error is controlled by a parameter [18] whose value is practically calibrated so that this error equals the numerical interpolation. Both methods thus lead to the expected accuracy.

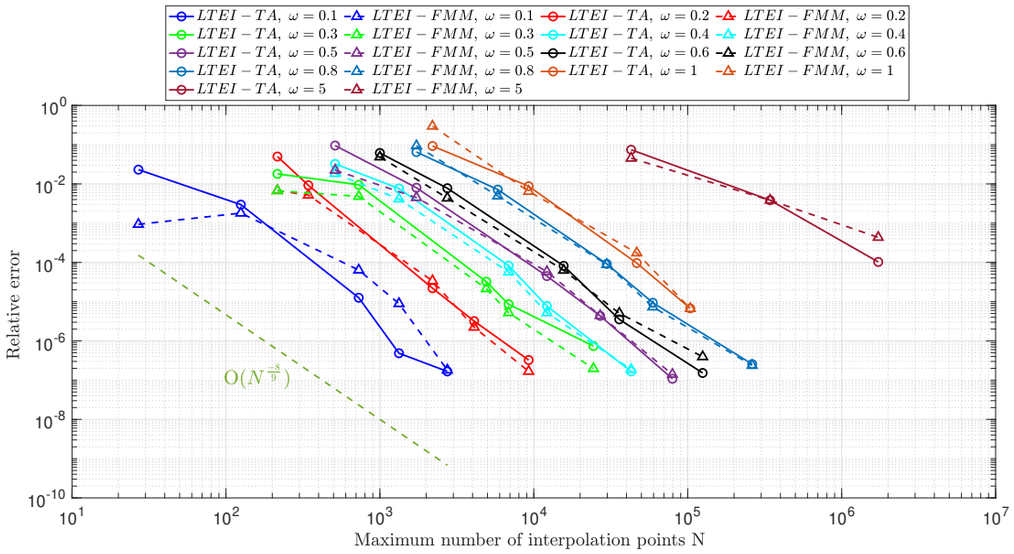


Figure 2.8: Approximation error of the evaluation of the long-range Coulomb matrix using LTEI-TA and LTEI-FMM with respect to the number of interpolation points N for various values of ω: convergence rate estimation. These calculations were carried for the Glycine molecule with $N_b = 100$ in the cc-pVDZ basis set.

Figure 2.8 considers the evaluation of the long-range Coulomb matrix using LTEI-TA and LTEI-FMM approaches as described in (2.65). It displays the relative error with respect to the number of interpolation points N for different values of ω, where the relative error of LTEI-TA (resp. LTEI-FMM) is $\frac{\left\|\mathbf{J}^{lr}-\mathbf{J}^{lr}_{LTEI-TA}\right\|_F}{\left\|\mathbf{J}^{lr}\right\|_F}$ (resp. $\frac{\left\|\mathbf{J}^{lr}-\mathbf{J}^{lr}_{LTEI-FMM}\right\|_F}{\left\|\mathbf{J}^{lr}\right\|_F}$). We note that we were not able to evaluate theoretically the convergence rate of this evaluation with respect to the number of interpolation points N. We observe, however, that the numerical error seems to have an almost linear-scaling in the 3D tensorized interpolations grid size N for small values of ω, i.e $\omega \leq 1$. However, this scaling is lost for larger ω. Indeed, we expect our method to be far less efficient for very large ω since the underlying kernel tends to the (singular) Coulomb one when $\omega \to +\infty$.

Computational cost

We first discuss the execution time required for the evaluation of an element of the long-range tensor $\mathcal{B}^{lr}$, as displayed in Figure 2.9 . The computational complexity of this evaluation is of order $\mathcal{O}(N_{q_1} N^{\frac{1}{3}} I_{\kappa\lambda}(N^{\frac{1}{3}} + I_{\mu\nu}))$ as discussed in Section 2.3.2. For small values of N_{q_1} and a few number of interpolations points $N^{\frac{1}{3}}$, we obtain linear scaling with respect to $N^{\frac{1}{3}}$ as shown in Figure 2.9. This is explained by the fact that the term $I_{\kappa\lambda} I_{\mu\nu}$ dominates the overall complexity for small ω. However, when ω increases, a quadratic complexity is observed with respect to $N^{\frac{1}{3}}$, which correponds to $\mathcal{O}(N_{q_1} N^{\frac{1}{3}} I_{\kappa\lambda}(N^{\frac{1}{3}} + I_{\mu\nu}))$. We also compare LTEI-TA with LTEI-FMM and with a naive numerical computation such that the two-electron integrals are computed with an integration over $N \times N \times N$ tensorized three dimensional Cartesian grids. We notice here that the LTEI-FMM approach has a linear scaling with regards to the number of interpolation points N as expected. We conclude that for the element-wise evaluation, LTEI-TA is the most efficient method.

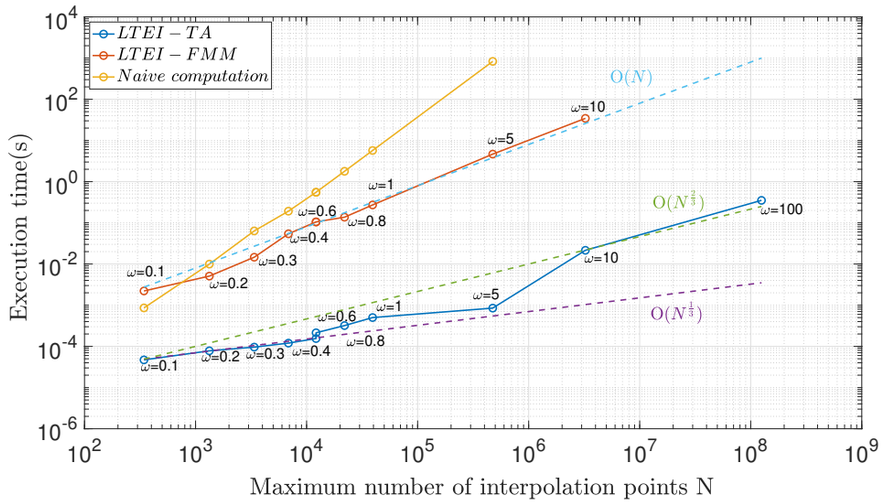


Figure 2.9: Computational time versus the maximum number of interpolation points N for different values of ω for the evaluation of the long-range two-electron integrals with relative error smaller than $\leqslant 10^{-4}$.

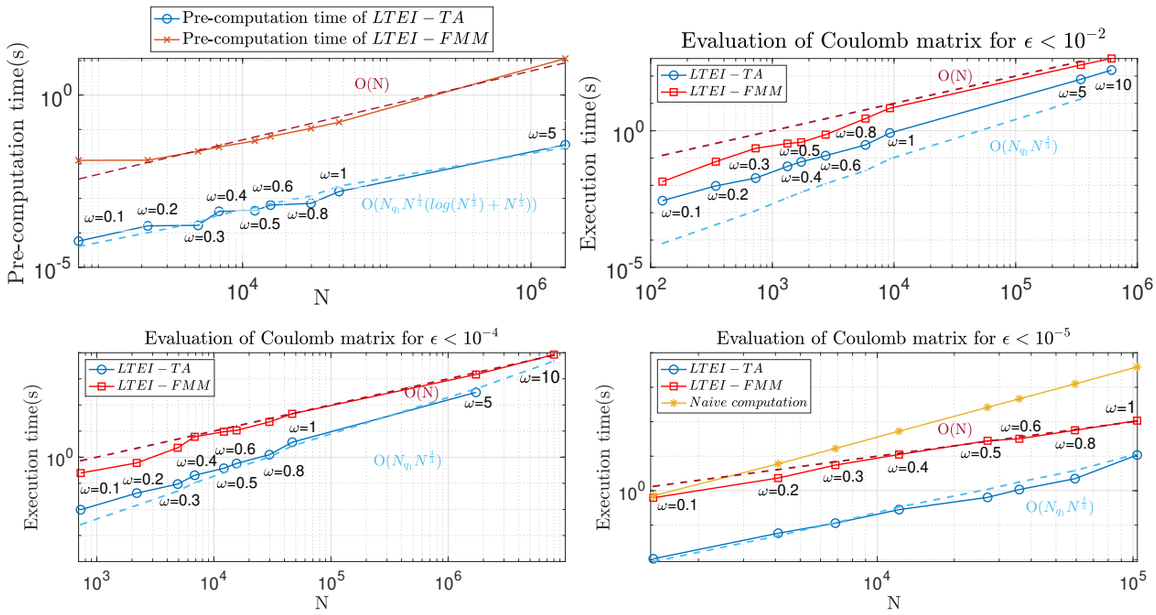


Figure 2.10: The leftmost plot represents the precomputation time for each approximation approach (LTEI-TA and LTEI-FMM) with respect to the maximum number of Chebyshev interpolation points N. We impose here that the relative error denoted by ϵ is smaller than $\leqslant 10^{-4}$. We provide in the other plots a comparison in terms of the computational time required for the evaluation of (2.64) between both approaches by varying the error bound ϵ and ω . We use the Glycine molecule $C_2H_5NO_2$ with fixed $N_b = 100$ and $N_{orb} = 95$ in the cc-pVDZ basis set.

Second, we compare the precomputation cost required to approximate the long-range kernel $K(\boldsymbol{x}, \boldsymbol{y})$, as given in (2.19), by using both approaches LTEI-TA and LTEI-FMM and by varying ω from 0.1 to 5. The results are displayed in the leftmost part of Figure 2.10. We observe that the runtime of LTEI-FMM depends linearly on the total number of interpolation points $\mathcal{O}(N)$, independently of the value of ω. LTEI-TA has also a precomputation time in accordance with the theory $\mathcal{O}(N_{q_1} N^{\frac{1}{3}}(log(N^{\frac{1}{3}}) + N^{\frac{1}{3}}))$ as explained in Section 2.3 . We observe that LTEI-TA is two orders of magnitude faster than LTEI-FMM for all the considered values of ω (which is a consequence of its small precomputation complexity). Hence, according to these numerical evidences, small test

cases (i.e. small ωb values) should better be handled using LTEI-TA while larger ones should better be considered through the LTEI-FMM.

Third, we discuss the time required to evaluate the long-range Coulomb matrix, as given in equation (2.64), which involves the multiplication of the matricization of $\mathcal{B}^{lr}$ with a matrix. Figure 2.10 illustrates the execution time with respect to the number of interpolation points N needed to achieve different relative errors for various values of ω for the evaluation of the Coulomb matrix. The relative error of LTEI-TA (resp. LTEI-FMM) is $\frac{\left\|\mathbf{J}^{lr}-\mathbf{J}^{lr}_{LTEI-TA}\right\|_F}{\left\|\mathbf{J}^{lr}\right\|_F}$ (resp. $\frac{\left\|\mathbf{J}^{lr}-\mathbf{J}^{lr}_{LTEI-FMM}\right\|_F}{\left\|\mathbf{J}^{lr}\right\|_F}$). We observe in Figure 2.10 that the evaluation of the long-range Coulomb matrix using LTEI-FMM approach scales linearly with the number of interpolation points $\mathcal{O}(N)$, but more than linearithmically for LTEI-TA. This reflects the complexity analysis of LTEI-TA method, $\mathcal{O}(N_{q_1}N^{\frac{4}{3}})$, provided in Section 2.3.4. However, LTEI-TA is still faster than LTEI-FMM for relatively small values of ω and for different relative errors. This numerical gain can be explained by the important prefactor of the LTEI-FMM approach: even if the complexity is linear, there is an important constant hidden in the big $\mathcal{O}$ notations [83]. While for small values of ω, N_{q_1} is small and hence LTEI-TA is more efficient. However, LTEI-TA is not asymptotically competitive with respect to LTEI-FMM approach. Indeed, as ω controls the regularity of the *erf*-interaction function, when ω increases, LTEI-TA needs a larger number of interpolation points N as well as quadrature points N_{q_1} to achieve a given accuracy. As a consequence, LTEI-TA becomes more costly and less efficient than LTEI-FMM.

To summarize, these results demonstrate two major things: first, LTEI-TA is a numerically highly efficient method, able to outperfom LTEI-FMM on tested cases. Second, we are able to reach the linear complexity (with regard to the total number of interpolation points) by exploiting LTEI-FMM, which allows to deal with more singular cases (with large values of ω). In the following, we want to study the efficiency of our numerical approaches for variable N_b.

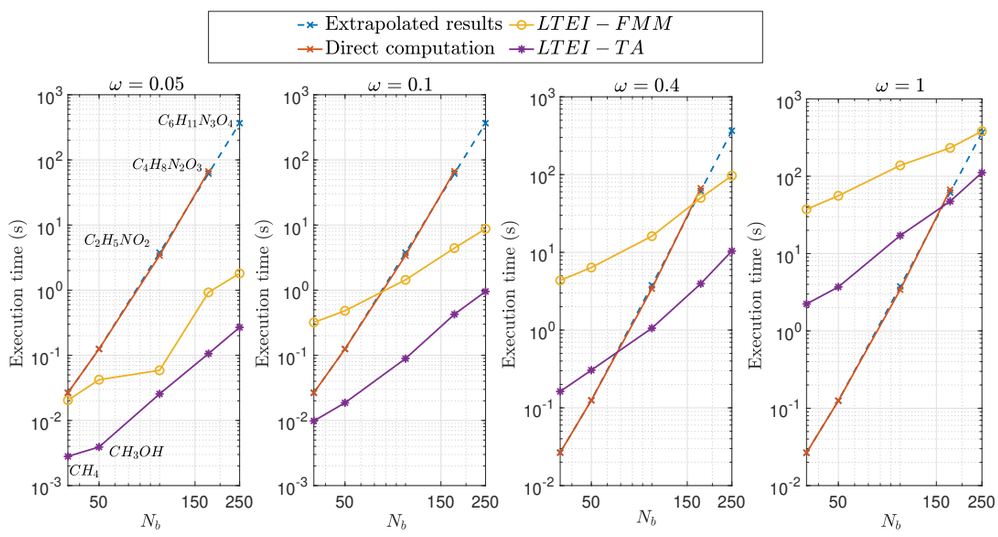


Figure 2.11: Execution time(s) required for the evaluation of (2.64) using the TEI tensor $\mathbf{B}^{lr}$ for different values of N_b, for $\omega = 0.05, \omega = 0.1, \omega = 0.4$, and $\omega = 1$ with imposed relative error smaller than 10^{-5}.

Figure 2.11 displays the execution times required to evaluate (2.64) with respect to the number of basis functions N_b for different values of ω and different molecules. We impose here that the relative errors of LTEI-TA and LTEI-FMM approaches for the evaluation of the long-range Coulomb matrix are smaller than 10^{-5}. We compare the running times between three approaches: the first approach involves the computation of the long-range Coulomb matrix via tensor contraction using the full two-electron integrals tensor, which is denoted by $\left(\mathbf{B}^{lr}\right)^{<2>} \in \mathbb{R}^{N_b^2 \times N_b^2}$ (times for $N_b > 175$ are obtained by extrapolation), such that its entries are pre-computed through quantum package [33]. The second (resp.

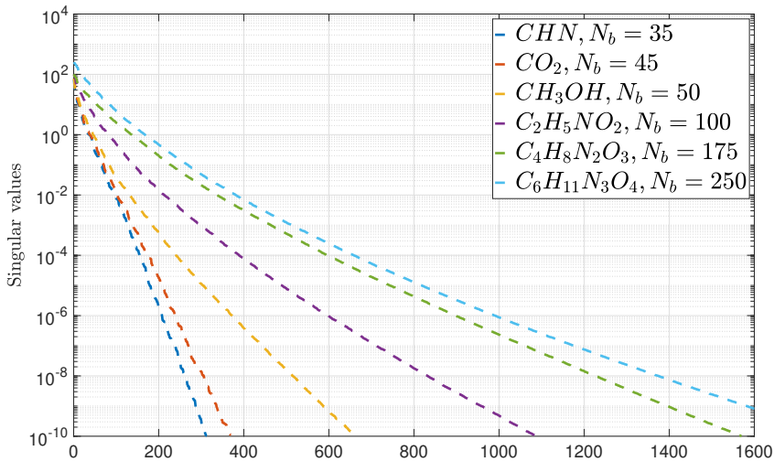
third) approach exploits the factorized structure of $\left(\mathbf{B}^{lr}\right)^{<2>}$ obtained through LTEI-TA (resp. LTEI-FMM) to compute (2.64). For small ω, we notice that a faster computation of (2.64) is obtained through LTEI-TA and LTEI-FMM methods: LTEI-TA is about one order of magnitude faster than LTEI-FMM. For important values of ω (ω=1), the new introduced approaches, LTEI-TA and LTEI-FMM, are less efficient given the high number of interpolation points N needed as well as the number of quadrature points N_{q_1} for LTEI-TA method as we notice in Figure 2.10. However, as N_b increases, the tensor contractions using the direct method become more computationally intensive and memory-demanding. In certain cases, the size of $\mathbf{B}^{lr}$ can even exceed available memory capacity . Therefore in some cases, it would be beneficial to use one of the new factorization methods to reduce the computational and storage cost. The numerical results are obtained for different molecules with different topologies. Therefore, in order to preserve the accuracy, in practice, we choose the size of the computational box $[-b, b]$ depending on the size of the molecule as well as the Gaussian functions decay as explained previously in Section 2.3, or one can opt for the adaptive approach explained in Section 2.6.2.

2.7.2 Tensor compression techniques

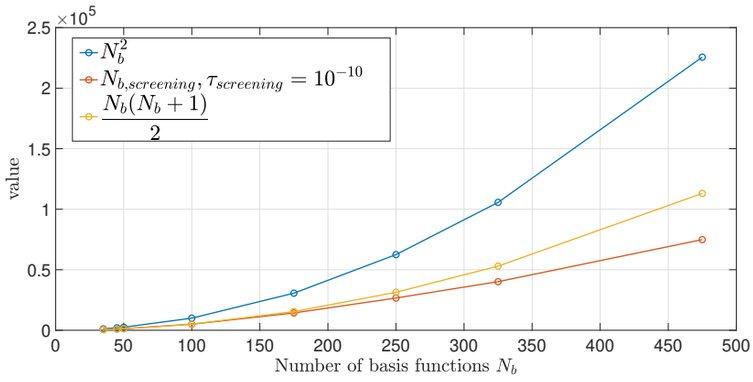
In this section we study numerically compression techniques to reduce the computation and storage cost of $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$ or $\mathbf{M}^{FMM} \in \mathbb{R}^{N_b^2 \times N}$ in order to speed up the evaluation of the Coulomb matrix (2.64). These techniques were discussed in Section 2.6.1. First, the number of basis functions N_b can be reduced by using screening techniques that exploit the symmetries of the pairs of basis functions as well as the properties of Gaussian type-functions. Indeed, Figure 2.12(b) shows that the number of pairs of Gaussian type basis functions N_b^2 can be reduced by using screening. Second, for small values of ω and different numbers of basis functions N_b, Figure 2.12(a) shows that the singular values of $\mathbf{M}_{TA,max}$ decay quickly , so $\mathbf{M}_{TA,max}$ can be approximated by a low-rank matrix. Therefore, we had recourse to three different approaches for the compression of $\mathbf{M}_{TA,max}$: the first approach, denoted by SVD, consists in approximating $\mathbf{M}_{TA,max}$ using ϵ-truncated SVD; the second approach, denoted by KR, exploits the Khatri-Rao product properties as discussed in Section 2.6.1; and the third approach, denoted by ADAP+KR, includes the partitioning of pairs of basis functions in terms of their numerical supports combined with KR approach as explained in Section 2.6.2.

Figure 2.13 (resp. Figure 2.14) displays the compression rate obtained between uncompressed $\mathbf{M}_{TA,max}$ matrix (resp. screened $\mathbf{M}_{TA,max}$ matrix) and its compressed representation, for different molecules with different number of basis functions N_b in the basis set cc-pVDZ. We notice that the best compression rate, i.e $(1-\frac{size\ of\ compressed\ version}{size\ of\ original})*100$, is obtained through the ADAP+KR approach as observed in Figure 2.14 (86% for $N_b = 175$) compared to the other approaches SVD (75% for $N_b = 175$) and KR (83% for $N_b = 175$). We observe that for SVD, the larger N_b ($N_b \geq 50$), the better the compression. While screening techniques reduce the storage requirements of the matrix $\mathbf{M}_{TA,max}$ [100], better compression results are obtained when they are combined with additional techniques introduced here. Figure 2.15, shows the computational time required for the compression of $\mathbf{M}_{TA,max}$. The worst execution time is obtained for SVD method as expected, in particular for large values of N_b ($N_b \geq 100$).

In summary, the adaptive approach leads to the best reduction in terms of storage while being the fastest among the tested methods. Moreover, the choice of the dimension of the computational box b does not have to be fixed in advance, since it depends on the pairs of Gaussian type-functions (1.94). We further investigate the accuracy of this method in Table 2.3. We display in this table the relative error obtained when approximating the Coulomb matrix (2.64) by using either $\mathbf{M}_{TA,max}$ compressed by the adaptive approach or a fixed computational box $[-b, b]$. The results show that the adaptive approach is more accurate than the ones obtained by fixing the computational box in advance. However, by using the adaptive method, the computation of the Coulomb matrix requires multiple matrix-matrix multiplications, and this can be more costly than fixing the computational box $[-b, b]$ in advance. However, since these multiplication can be performed in parallel, parallelization might be a key component to speed up the computation of the long-range Coulomb matrix (2.64).



(a)



(b)

Figure 2.12: (a) Singular values of $\mathbf{M}_{TA,max} \in \mathbb{R}^{N_b^2 \times N}$ for different molecules with $\omega = 0.1$ and $N_{q_1} = 3$. (b) Number of reduced pairs of basis functions obtained by exploiting symmetry (yellow curve), as well as symmetry+properties of Gaussian type functions with $\tau_{screening} = 10^{-10}$ (red curve).

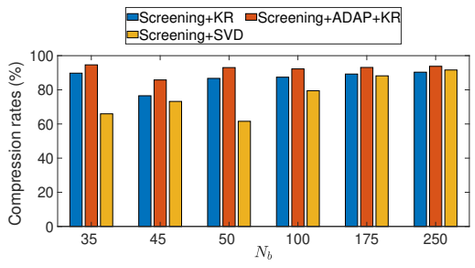


Figure 2.13: Compression rate between the original computed $\mathbf{M}_{TA,max}$ matrix and its compressed representation for $\omega = 0.3$ for different values of N_b, for the different molecules displayed in Figure 2.12.

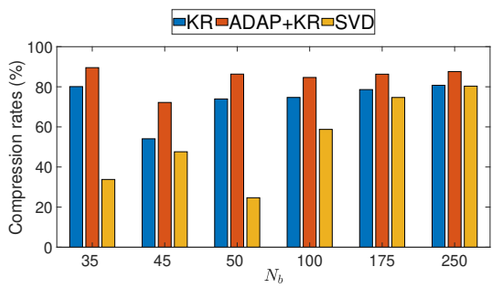


Figure 2.14: Compression rate between $\mathbf{M}_{TA,max}$ matrix (after screening) and its compressed representation for $\omega = 0.3$ for different values of N_b, for the different molecules displayed in Figure 2.12.

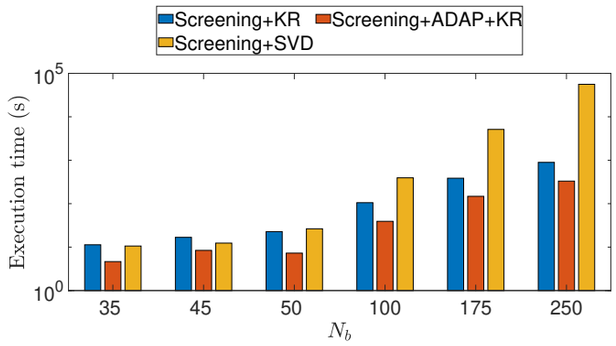


Figure 2.15: Execution time(s) of different compression methods defined in Section 2.6.1 for $\omega = 0.3$, for different values of N_b, for the different molecules displayed in Figure 2.12.

Table 2.3: Adaptive method, $\omega = 0.5$

Molecule	$C_2H_5NO_2$	$C_4H_8N_2O_3$	$C_6H_{11}N_3O_4$
N_b	100	175	250
Adaptive approach	1.0354e-7	2.4882e-8	4.587e-7
$b = 15$	1.6058e-7	1.8332e-7	8.2245e-7
$b = 10$	3.7359e-07	0.001	0.02068

2.8 Concluding remarks and perspectives

This work introduces two new compression methods for the long-range kernel K and the approximation of the long-range six-dimensional two-electron integrals tensor. The first approach, referred to as LTEI-TA, relies on two-dimensional Chebyshev interpolation, Gaussian quadrature for numerical integration, and FFT for computing Chebyshev coefficients. The approximation of the long-range two-electron integrals tensor $\mathcal{B}^{lr}$ by using this method allows to exploit a tensorized structure that leads to an efficient application of the matricization of $\mathcal{B}^{lr}$ to evaluate the long-range Coulomb matrix for fixed N_b and N_{orb}, with $\mathcal{O}(N_{q_1} N^{\frac{4}{3}})$ complexity, where N is the number of Chebyshev interpolation points and N_{q_1} is the number of quadrature points. The second approach, referred to as LTEI-FMM, relies on kernel-independent Fast Multipole Methods, with $\mathcal{O}(N)$ complexity. It exploits the asymptotically smooth behaviour of the long-range kernel K. The storage and time complexity of the presented methods were analysed and compared numerically, exhibiting both the high efficiency of LTEI-TA and the linear complexity of LTEI-FMM. We further investigated the compression of $\mathcal{B}^{lr}$ by using screening techniques, low-rank methods, and an adaptive approach. LTEI-TA approach is particularly efficient for small values of ω, where ω is the separation parameter that controls the regularity of K. However, for large values of ω, in order to preserve accuracy, the number of interpolation points as well as the number of quadrature points becomes important for LTEI-TA and thus LTEI-FMM becomes more efficient. It's worth noting that if we use a low-rank basis $\{g_\mu\}_{1 \leq \mu \leq N_b}$ where basis functions are expressed as tensor products of one-dimensional functions both novel approximation methods introduced here can be adapted to accommodate various low-rank basis functions, not just Gaussian-type ones. However, for the adaptive method discussed in Section 2.6.2, the use of Gaussian-type functions is still necessary.

In conclusion, our study has introduced promising methodologies to effectively address the long-range component of the range-separated Coulomb potential as well as the efficient evaluation of the long-range TEI tensor within our computational framework. Nevertheless, it's crucial to acknowledge the existing established approaches aimed at handling similar kernels, as initially highlighted in the introduction. One common technique involves factorizing the long-range kernel through Fourier transforms in spherical coordinates, combined with spherical numerical integration, as it was considered in the

works of [71, 79, 110]. This approach expresses the kernel as a sum of product of separable functions, similar to what was obtained with our factorization methods, but using distinct discretization strategies. While it was not covered in this manuscript, we anticipate that a comparative analysis against our proposed methods could yield valuable insights. Such a comparison would involve considering various parameters, including the number of required discretization nodes, achieved accuracy, and suitability for high-performance computing (HPC) implementations. Additionally, as future work, we are planning to explore the potential of LTEI-TA for small values of ω in a boarder range of quantum chemical contexts as post-HF models with range separation or hybrid approaches such as (long-range) DMRG–short-range DFT [43]. Such work might be also beneficial for Particle Mesh Ewald methods [21]. We also acknowledge the importance of investigating the treatment of the short-range TEI tensor, as its computational cost and accuracy depend on the value of ω, which is the trade-off between the long-range and short-range parts, and the numerical method used. Therefore, it would be interesting to link LTEI-FMM with singular quadrature based evaluation for short-range. Additionally, efficient parallelization of FMM could also benefit to LTEI-FMM on distributed memory architecture. Now, we shift our focus to the second tensor-based high-dimensional problem: the efficient representation of the tensor train decomposition involved in the DMRG method, which will be the subject of the upcoming chapter.

Chapter 3

Symmetry-preserving tensor train representations

Contents

3.1 Introduction

This chapter presents the outcomes of a collaborative project with Eric Cancès (Professor at the Ecole des Ponts - ParisTech), Mi-Song Dupuy (Associate professor at Sorbonne university), and my supervisor, Laura Grigori. The foundation for this project was established during the Cemracs summer school in 2021 and several workshops held in Roscoff part of the EMC2 project. Our primary goal is to gain a deeper understanding of the efficient TTO representation of the Hamiltonian operator, through low-rank approximation methods such as truncated SVD. We aim also to investigate how the incorporation of physical symmetries as the conservation of particle number yields to practical advantages from a numerical point of view. Indeed, imposing physical symmetries yields a block structure in the TT-cores of the TT representaion of both the operator and its eigenfunction/wavefunction. Thereby, we show that the structure obtained allows to speed-up the most time-consuming steps in QC-DMRG computations, including contractions, compression, and matrix-vector multiplications encountered in the eigensolver. After revisiting the fundamental properties of the quantum chemical Hamiltonian operator and surveying various existing approaches for constructing a TTO representation in Section 3.2.2, the primary contributions of this chapter unfold with

- In Section 3.2.3, we demonstrate that the low-rank approximation during a TT-SVD process, using tSVD with respect to the degenerate singular values (i.e, for a specific numerical threshold, if the truncation lies within a set of degenerate singular values, this set is either truncated or kept entirely) preserves important properties of the Hamiltonian operator as outlined in Theorem 3. Specifically, it retains the property

of the Hamiltonian being represented by a symmetric matrix and preserves Abelian symmetries associated, for example, with the conservation of particle number. We also show that applying tSVD, based on some numerical thresholds, affects additional properties preserved by the Hamiltonian operator. These include non-Abelian symmetries associated to the conservation of total spin, as well as the emergence of non-existing interaction terms for an operator limited to at most 2-body interactions.

- In Section 3.3.1, inspired by the work of [2], we offer a constructive demonstration showcasing the block-sparse structure inherent in the TT-cores of the TTO decomposition of a general p-body particle-number preserving Hamiltonian operator, as highlighted in Theorem 4. Further investigation revealed that the block sparsity in tensor trains stemming from the preservation of the particle number, is known in the chemistry and physics communities but has not been introduced in this particular representation before.

- In Section 3.3.2, with invaluable assistance from Eric Cancès, we invest significant effort in understanding the structure of the TT decomposition of eigenfunctions when non-Abelian symmetries such as SU(2), associated with the conservation of the total spin, are applied. This is provided in Theorem 5. We provide a comprehensive study of the TT representation of these eigenfunctions, presenting preliminary theoretical insights and providing results in a simplified manner with examples to ensure accessibility for readers who may not possess a background in physics. It is worth noting that some of the findings we present have already been established within the physics and chemistry communities.

3.2 Structure preserving TTO representation

The numerical study of large and strongly correlated quantum systems remains a challenging problem due to the exponential growth of the Hilbert space with the system size d. One promising approach to tackle this problem is the use of tensor networks methods as the TT representation [27, 74, 90], introduced in Section 1.2.3. As such, the construction of the TTO representation, known as the matrix product operator (MPO) in physics language, of an Hamiltonian operator is at the core of the QC-DMRG algorithm, as introduced in Section 1.3.4. In what follows, we review the properties of the electronic Hamiltonian operator and we recall known approaches that have been proposed for the construction of a TTO representation of the quantum chemical operator with at-most 2-body interactions.

Remark 4. Let d denote the size of the system, i.e., the number of considered orbitals, and n represent the number of possible states occupying each orbital. As outlined in Remarks 1.3.4 and 1.3.9, when working in the spatial-orbital basis, we set $d := d_{\text{spatial}}$ and $n = 4$. Conversely, in the spin-orbital basis, we have $d := 2d_{\text{spatial}}$ and $n = 2$. Moving forward, we maintain the notation for the system size and number of possible states as d and n, respectively, and specify the basis whenever necessary.

3.2.1 Properties of the electronic Hamiltonian operator

The 2-body Hamiltonian operator under consideration can be represented by the matrix $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ and is subject to specific properties, as outlined in [36, 106, 133], that must be fulfilled. Here, we consider two distinct types of symmetries: permutation symmetries of tensors (invariance under the permutation of specific modes) and physical/group symmetries arising from conservation laws, as elaborated below

1. Symmetric Hamiltonian matrix: $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ is symmetric, let $\mathcal{H} \in \mathbb{R}^{n \times \cdots \times n}$ be the $2d$-th order tensor folding of $\mathbf{H}$ and let $\tilde{\mathcal{H}} \in \mathbb{R}^{n \times \cdots \times n}$ be its permutation according to specific modes such that, for $\mu_k, \nu_k \in [n], k \in [d]$, the following holds:

$$\begin{aligned} &\mathbf{H}(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) = \mathbf{H}(\overline{\nu_1, \ldots, \nu_d}; \overline{\mu_1, \ldots, \mu_d}) \\ &\Leftrightarrow \tilde{\mathcal{H}}(\mu_1; \nu_1; \ldots; \mu_d; \nu_d) = \tilde{\mathcal{H}}(\nu_1; \mu_1; \cdots; \nu_d; \mu_d), \end{aligned} \tag{S}$$

2. Particle number conservation: The Hamiltonian is an operator that preserves the number of particles, i.e it commutes with the particle number operator represented by the matrix $\mathbf{N} \in \mathbb{R}^{n^d \times n^d}$ as defined in (1.139) (if $n = 4$), such that

$$\begin{aligned} &\mathbf{NH} - \mathbf{HN} = 0 \\ &\Leftrightarrow \mathbf{H}(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) \neq 0 \Rightarrow \sum_{k=1}^{d} q_n(\mu_k) = \sum_{k=1}^{d} q_n(\nu_k), \end{aligned} \tag{PN}$$

where depending on the value of n, $q_2 : \{1, 2\} \mapsto \{0, 1\}$ and $q_4 : \{1, 2, 3, 4\} \mapsto \{0, 1, 1, 2\}$ are mapping functions, see Equations (1.118) and (1.133).

3. Conservation of the z-component of the total spin: the Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{4^d \times 4^d}$, expressed in the spatial-orbital basis, commutes with the z-component of the total spin operator represented by the matrix $\mathbf{S}^z \in \mathbb{R}^{4^d \times 4^d}$. The matrix $\mathbf{S}^z$ is given by:

$$\mathbf{S}^z = \frac{1}{2} \sum_{i=1}^{d} \left(\mathbf{A}^*_{i,\frac{1}{2}} \mathbf{A}_{i,\frac{1}{2}} - \mathbf{A}^*_{i,\frac{-1}{2}} \mathbf{A}_{i,\frac{-1}{2}} \right). \tag{3.1}$$

Let us define the following mapping function:

$$q_z : \{1, 2, 3, 4\} \mapsto \left\{ 0, \frac{1}{2}, \frac{-1}{2}, 0 \right\}. \tag{3.2}$$

This function serves as a mapping between tensor indices and the spin of the occupation state of the orbital. A spatial orbital, when occupied, can accommodate a spin-up electron with a spin value of $\frac{1}{2}$, a spin-down electron with a spin value of $-\frac{1}{2}$, or both a spin-up and spin-down electron with a spin value of 0. Overall, each occupation is linked to a spin value, which can be $\frac{1}{2}$, $\frac{-1}{2}$, or 0. It is important to note that the spin value is 0 when the orbital is unoccupied. Now, the following conditions hold:

$$\begin{aligned} &\mathbf{S}^z\mathbf{H} - \mathbf{HS}^z = 0 \\ &\Leftrightarrow \mathbf{H}(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) \neq 0 \Rightarrow \sum_{k=1}^{d} q_z(\mu_k) = \sum_{k=1}^{d} q_z(\nu_k). \end{aligned} \tag{SZ}$$

Later in our discussion, we show that the image of the function q_z corresponds to the eigenvalues of $(\mathbf{S}^z)^{\leq 1} = \frac{1}{2} \left(\mathbf{A}^*_{1,\frac{1}{2}} \mathbf{A}_{1,\frac{1}{2}} - \mathbf{A}^*_{1,\frac{-1}{2}} \mathbf{A}_{1,\frac{-1}{2}} \right)$.

4. Conservation of the total spin: the Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{4^d \times 4^d}$ commutes with the total spin operator, denoted in the literature as $\hat{S}^2$. We maintain this notation when representing the operator by a matrix, such that it is defined as:

$$\begin{aligned} \hat{S}^2 &= (\mathbf{S}^x)^2 + (\mathbf{S}^y)^2 + (\mathbf{S}^z)^2 \\ &= \frac{1}{2} \left(\mathbf{S}^+\mathbf{S}^- + \mathbf{S}^-\mathbf{S}^+ \right) + \mathbf{S}^z\mathbf{S}^z \in \mathbb{R}^{4^d \times 4^d}, \end{aligned} \tag{3.3}$$

where

$$\mathbf{S}^x = \frac{1}{2} \sum_{k=1}^{d} \left(\mathbf{A}^*_{k,\frac{1}{2}} \mathbf{A}_{k,\frac{-1}{2}} + \mathbf{A}^*_{k,\frac{-1}{2}} \mathbf{A}_{k,\frac{1}{2}} \right), \quad \mathbf{S}^y = \frac{1}{2i} \sum_{k=1}^{d} \left(\mathbf{A}^*_{k,\frac{1}{2}} \mathbf{A}_{k,\frac{-1}{2}} - \mathbf{A}^*_{k,\frac{-1}{2}} \mathbf{A}_{k,\frac{1}{2}} \right), \tag{3.4}$$

and

$$\mathbf{S}^+ = \mathbf{S}^x + i\mathbf{S}^y, \ \mathbf{S}^- = \mathbf{S}^x - i\mathbf{S}^y, i^2 = -1. \tag{3.5}$$

Remark 5. In quantum physics, operators that commute with the Hamiltonian operator, as the particle number operator $\mathbf{N}$, represent *observables* or physical quantities. These observables are self-adjoint operators, implying that their spectrum consist of real numbers. The eigenvalues of these observables correspond to physical quantities known as *quantum numbers*, examples of which include the number of particles denoted by N.

5. p-body interactions: the Hamiltonian matrix $\mathbf{H}$ is derived from an operator with at most p-body interactions, $p \in \mathbb{N}$, such that

$$\mathbf{H}(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) \neq 0 \Rightarrow \sum_{k=1}^{d} |q_n(\mu_k) - q_n(\nu_k)| \leq 2p. \tag{pB}$$

For the 2-body Hamiltonian operator as defined in (1.138) or in (1.130), $p = 2$, (due to the presence of 2-body terms along with 1-body terms, see Remark 1.3.6).

Remark 6. We observe that the electronic Hamiltonian operator, under consideration, commutes with both the particle number operator, denoted as $\mathbf{N}$, and the azimutal spin operator, denoted as $\mathbf{S}^z$. In the context of physics, this implies that the Hamiltonian $\mathbf{H}$ is a particle-preserving operator and also preserves the z-component of the total spin. This conservation property arises from the fact that these operators commute with an operator that remains invariant under symmetry transformations associated with the Abelian unitary group U(1) [111]. Moreover, the electronic Hamiltonian operator commutes with the total spin operator $\hat{S}^2$, signifying the conservation of the total spin. This invariance under symmetry transformations is related to the non-Abelian group SU(2) [112].

In the existing literature, there are two approaches for constructing the TTO representation of the 2-body electronic Hamiltonian operator. The first method is the exact construction, optimally hand-crafted according to the expression of the operator. This involves a great effort to capture all the specific rules required for constructing the TTO decomposition due to its unique structure. In Section 3.2.2, we provide an overview of the various approaches used to construct an exact TTO representation of the Hamiltonian operator. The second approach is the generic/naïve construction of an approximate TTO representation, which relies on the analytical description of the Hamiltonian as a sum of rank-1 tensor trains and uses the defined arithmetic operations (as outlined in 1.2.3) with additional compression to systematically and automatically generate the TTO decomposition with optimal TTO-ranks. While this approach provides high adaptability and can be applied to a wide range of operator models due to their analytical description as a sum of rank-1 tensor trains, as discussed in Section 3.2.3, it does have numerical limitations that we will address later.

3.2.2 Related work: Exact construction of TTO representation of Hamiltonian operators with at most 2-body interactions

The most commonly used method involves manual design of the TTO decomposition, achieved by deriving recursive relationships between operators at consecutive iterations, either between k and $k+1$ or between $k-1$ and k, see [17], with $k \in [d]$. For instance, it starts by partitioning the number of orbitals into left and right components at a specific splitting index, denoted as k, and subsequently identifies a recurrence relation that incorporates the TT-cores within the TTO representation. This process can be explained as follows: let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be a TTO representation of the Hamiltonian operator $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$, with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times n \times R_k}$, $R_0 = R_d = 1$, for $k \in [d]$. $\mathbf{H}$ can be expressed as, see (1.143):

$$\mathbf{H} = \sum_{\beta_1=1}^{R_1} \ldots \sum_{\beta_{k-1}=1}^{R_{k-1}} \sum_{\beta_k=1}^{R_k} \ldots \sum_{\beta_{d-1}=1}^{R_{d-1}} \mathbf{H}_1(1; \beta_1) \otimes_K \ldots \otimes_K \mathbf{H}_k(\beta_{k-1}; \beta_k) \otimes_K \ldots \mathbf{H}_d(\beta_{d-1}; 1), \tag{3.6}$$

where we denote the matrices $\mathbf{H}_k(\beta_{k-1}; \beta_k) := \mathcal{H}_k[\beta_{k-1}, :, :, \beta_k] \in \mathbb{R}^{n \times n}$. Following a partition into left blocks $L_{\leq k}$ and right blocks $R_{>k}$, the operator can be expressed as:

$$\mathbf{H} = \sum_{\beta_k=1}^{R_k} \mathbf{H}_{L_{\leq k}}[:, \beta_k] \otimes_K \mathbf{H}_{R_{>k}}[\beta_k, :], \tag{3.7}$$

where

$$\mathbf{H}_{L_{\leq k}}[:, \beta_k] = \sum_{\beta_1=1}^{R_1} \ldots \sum_{\beta_{k-1}=1}^{R_{k-1}} \mathbf{H}_1(1; \beta_1) \otimes_K \ldots \otimes_K \mathbf{H}_k(\beta_{k-1}; \beta_k) \in \mathbb{R}^{n^k \times n^k},$$

and

$$\mathbf{H}_{R_{>k}}[\beta_k, :] = \sum_{\beta_k=1}^{R_k} \dots \sum_{\beta_{d-1}=1}^{R_{d-1}} \mathbf{H}_{k+1}(\beta_k; \beta_{k+1}) \otimes_K \dots \otimes_K \mathbf{H}_d(\beta_{d-1}; 1) \in \mathbb{R}^{n^{d-k} \times n^{d-k}}.$$

Now, it is notable that the left $\mathbf{H}_{L_{\leq k}}$ and right $\mathbf{H}_{R_{>k}}$ operators are explicitly linked to their counterparts at partitions $k+1$ and $k-1$ as follows:

$$\mathbf{H}_{L_{\leq k+1}}[:, \beta_{k+1}] = \sum_{\beta_k=1}^{R_k} \mathbf{H}_{L_{\leq k}}[:, \beta_k] \otimes_K \mathbf{H}_{k+1}(\beta_k; \beta_{k+1}), \tag{3.8}$$

and

$$\mathbf{H}_{R_{>k-1}}[\beta_{k-1}, :] = \sum_{\beta_k=1}^{R_k} \mathbf{H}_k(\beta_{k-1}; \beta_k) \otimes_K \mathbf{H}_{R_{>k}}[\beta_k, :]. \tag{3.9}$$

Therefore, in the systematic derivation of a TTO representation, the authors in [17] start by determining the partitioning expression of the quantum chemical Hamiltonian operator. Subsequently, they manually design the TT-cores following specific recursion rules between neighboring orbitals indexed at $k, k+1$, or $k-1, k$, as depicted in (3.8) and (3.9). For a comprehensive understanding of this process, please refer to Appendix .3. As explained in Appendix .3, manually deriving the TTO representation may introduce redundant terms in the TT-cores, resulting in a sub-optimal representation. This issue can be mitigated by introducing complementary operators, as exemplified in Appendix .3 and discussed in [17]. In [59], the authors employ a similar approach, often referred to as *fork-merge* or *fork-merge-merge* operations, to efficiently reuse common intermediate terms that appear in the TT-cores. In [113], the authors use an alternative approach, involving the search for an efficient TTO representation of the various sums of creation and annihilation operators in the expression of the Hamiltonian operator, which involves also the factorization of the one-electron and two-electron integrals. We review this process next. We begin by defining the following simple operator in a spin-orbital basis:

$$\mathbf{O} = \sum_{i=1}^{d} t_i \mathbf{A}_i^* \mathbf{A}_i, t_i \in \mathbb{R}. \tag{3.10}$$

Let $(\mathcal{H}_1, \dots, \mathcal{H}_d)$ be the TTO decomposition of $\mathbf{O}$ and let $\mathbf{H}_k^{<2>} \in \mathbb{R}^{2r_{k-1} \times 2r_k}$ be the mode-(1:2) matricization of $\mathcal{H}_k, k \in [d]$. Given the definitions of operators $\mathbf{A}_i$ and $\mathbf{A}_i^*$, see (1.124) and (1.123) respectively, the TTO representation of (3.10), employing the strong Kronecker product, as defined in (1.144), is expressed as follows:

$$\mathbf{O} = \underbrace{\begin{bmatrix} \mathbf{I}_2 & t_1 \mathbf{A}^* \mathbf{A} \end{bmatrix}}_{\mathbf{H}_1^{<2>} \in \mathbb{R}^{(1*2) \times (2*2)}} \bowtie \underbrace{\begin{bmatrix} \mathbf{I}_2 & t_2 \mathbf{A}^* \mathbf{A} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{H}_2^{<2>} \in \mathbb{R}^{(2*2) \times (2*2)}} \bowtie .... \bowtie \underbrace{\begin{bmatrix} t_d \mathbf{A}^* \mathbf{A} \\ \mathbf{I}_2 \end{bmatrix}}_{\mathbf{H}_d^{<2>} \in \mathbb{R}^{(2*2) \times (2*1)}}, \tag{3.11}$$

where $\mathbf{A}$ is defined in (1.125). $\mathbf{O}$ has a TTO representation with a maximum TTO-rank of 2, as indicated by the highlighted red values in Equation (3.11). The construction process of (3.11) is explained in Appendix .4. Now, it is more complicated to express the Hamiltonian in Equation (1.130) in a TTO form. In order to achieve this, we begin by examining the 1-body operator, which corresponds to the first sum in (1.130). To simplify, we consider the spin-orbital basis. The one-body term, denoted as $\mathbf{H}_{(1)}$, can be divided into two separate sums.

$$\begin{aligned} \mathbf{H}_{(1)} &= \sum_{i,j=1}^{d} h_{ij} \mathbf{A}_i^* \mathbf{A}_j \\ &= \underbrace{\sum_{i=1}^{d} h_{ii} \mathbf{A}_i^* \mathbf{A}_i}_{\mathbf{T}_1} + \underbrace{\sum_{1 \leq i < j \leq d} \Big(h_{ij} \mathbf{A}_i^* \mathbf{A}_j + h_{ij} \mathbf{A}_j^* \mathbf{A}_i \Big)}_{\mathbf{T}_2}, \end{aligned} \tag{3.12}$$

with h_{ij} being the one-electron integrals defined in Equation (1.81). Similarly to (3.11), the first sum $\mathbf{T}_1$ leads to a TTO of the form:

$$\mathbf{T}_1 = \begin{bmatrix} \mathbf{I}_2 & h_{11} \mathbf{A}^* \mathbf{A} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{I}_2 & h_{22} \mathbf{A}^* \mathbf{A} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \bowtie .... \bowtie \begin{bmatrix} h_{dd} \mathbf{A}^* \mathbf{A} \\ \mathbf{I}_2 \end{bmatrix}. \tag{3.13}$$

For the second term $\mathbf{T}_2$, the author in [113] employs a factorization of the one-electron integrals matrix, articulated as follows: Let $\mathbf{M} \in \mathbb{R}^{d\times d}$ be the matrix with entries $\mathbf{M}(i;j) := h_{ij}$ of rank $r \leq d$, for $i, j \in [d]\,, 1 \leq i < j \leq d$. Suppose that there exist matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d\times r}$, such that $\mathbf{M}$ decomposes as:

$$\mathbf{M}(i;j) = h_{ij} = \sum_{\ell=1}^{r} \mathbf{U}(i;\ell)\mathbf{V}^*(\ell;j), \tag{3.14}$$

then, it follows:

$$\sum_{1\leq i<j\leq d} h_{ij}\mathbf{A}_i^*\mathbf{A}_j = \sum_{1\leq i<j\leq d}\sum_{\ell=1}^{r} \mathbf{U}(i;\ell)\mathbf{V}^*(\ell;j)\mathbf{A}_i^*\mathbf{A}_j \;\; = \sum_{1\leq i<j\leq d}\sum_{\ell=1}^{r} \left(\mathbf{U}(i;\ell)\mathbf{A}_i^*\right)\left(\mathbf{V}^*(\ell;j)\mathbf{A}_j\right). \tag{3.15}$$

Now, by construction, one can write the TTO of $\mathbf{H}_{(1)}$ as follows:

$$\begin{aligned}
\mathbf{H}_{(1)} &= \mathbf{T}_1 + \mathbf{T}_2 \\
&= \begin{bmatrix} \mathbf{I}_2 & h_{11}\mathbf{A}^*\mathbf{A} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{I}_2 & h_{22}\mathbf{A}^*\mathbf{A} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \bowtie .... \bowtie \begin{bmatrix} h_{dd}\mathbf{A}^*\mathbf{A} \\ \mathbf{I}_2 \end{bmatrix} \\
&+ \sum_{\ell=1}^{r} \begin{bmatrix} \mathbf{I}_2 & \mathbf{U}(1;\ell)\mathbf{A}^* & \mathbf{0} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{I}_2 & \mathbf{U}(2;\ell)\mathbf{A}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{S} & \mathbf{V}^*(\ell;2)\mathbf{A} \\ \mathbf{0} & & \mathbf{I}_2 \end{bmatrix} \bowtie \cdots \bowtie \begin{bmatrix} \mathbf{0} \\ \mathbf{V}^*(\ell;d)\mathbf{A} \\ \mathbf{I}_2 \end{bmatrix} \\
&+ \sum_{\ell=1}^{r} \begin{pmatrix} \mathbf{I}_2 & \mathbf{U}(1;\ell)\mathbf{A} & \mathbf{0} \end{pmatrix} \bowtie \begin{bmatrix} \mathbf{I}_2 & \mathbf{U}(2;\ell)\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} & \mathbf{V}^*(\ell;2)\mathbf{A}^* \\ \mathbf{0} & & \mathbf{I}_2 \end{bmatrix} \bowtie \cdots \bowtie \begin{bmatrix} \mathbf{0} \\ \mathbf{V}^*(\ell;d)\mathbf{A}^* \\ \mathbf{I}_2 \end{bmatrix} \\
&= \mathbf{H}_1 \bowtie \mathbf{H}_2 \bowtie \cdots \bowtie \mathbf{H}_d.
\end{aligned} \tag{3.16}$$

$\mathbf{H}_{(1)}$ can be expressed as a sum of TTO decompositions with TTO-ranks of 2 and 3. By leveraging the properties of the sum of TTO decompositions, as elucidated in Proposition 1.2.2, $\mathbf{H}_{(1)}$ has a TTO decomposition with a TTO-rank that scales linearly with d where $r \leq d$.

Representing the 2-body operator, which is the second sum in (1.130), in TTO form is a more challenging task. It involves using anti-commutation relations and expressing the two-electron integrals in a factorized format, similar to the approach employed for the one-body operator. For more details on this, we refer the reader to [113]. In Bachmayr's work [2], alternative exact TTO representations with optimal ranks for the one- and 2-body operators are explored.

While these representations provide optimal TTO representations for the quantum chemical Hamiltonian operator with TTO-ranks that scale quadratically with the number of orbitals, i.e $\mathcal{O}(d^2)$ see [2], and demonstrate hand-crafted intelligent design, it is important to note that more complicated Hamiltonian operators may require individual redesign and re-implementation. It gets complicated when dealing with models beyond the 2-body electronic Hamiltonian operator that involve more than 2-body interactions. Within this category, there are Hamiltonian operators introducing 3-body interactions, such as the nuclear Hamiltonian operator usually encountered in nuclear physics [119], or the transcorrelated Hamiltonian [5, 6], a non-Hermitian 3-body operator. This renders the previously introduced approaches for constructing a compact and exact TTO representation less generic and automatic when dealing with these operators. A thoughtful redesign is necessary. In the following, we revisit the second well-known naive approach for constructing the TTO decomposition of Hamiltonian operators expressed as the sum of Kronecker products of matrices, as the case of the quantum chemical Hamiltonian operator in (1.130).

3.2.3 Generic and compression-based construction of TTO

An approximate TTO representation can be obtained through a naive construction which relies on applying a sequence of tSVD factorizations and arithmetic operations involving tensor trains as addition, or by eliminating linearly dependent terms, see [52]. While this approach offers generality and automation across different operators, it undoubtedly

introduces a numerical error whose impact on subsequent calculations may be challenging to quantify in advance. Additionally, the time required for numerical compression can become significant when dealing with large TTO-ranks.

We start by briefly reviewing the naive construction method. The annihilation and creation operators, denoted by $\mathbf{A}_i$ and $\mathbf{A}_i^*$ in (1.123) and (1.124), can be regarded as rank-1 TTO. Consequently, we can represent these operators as a set of rank-1 TT-cores. With the creation and annihilation operators expressed as tensor trains, we employ arithmetic operations such as addition to construct the TTO representation of $\mathbf{H}$. The product between rank-1 TT-cores followed by multiplication to a scalar such as $(h_{ij}\mathbf{A}_i^*\mathbf{A}_j, i, j \in [d])$ will keep the rank equal to 1 (see Proposition 1.2.2). However, adding these terms together $(\sum_{i,j=1}^{d} h_{ij}\mathbf{A}_i^*\mathbf{A}_j)$, will increase linearly the ranks (i.e $\mathcal{O}(d^2)$) for the 1-body operator and $\mathcal{O}(d^4)$ for the 2-body operator). In order to avoid this scaling, a compression process is necessary, which can be accomplished through truncated SVD. Such process is known as the TT-rounding as described in Algorithm 2 such that the accuracy $\delta = \frac{\epsilon}{\sqrt{d-1}}$ is chosen a priori and the compressed TTO decomposition is ϵ-close to the original TTO decomposition. The construction of the compressed TTO representation given the Hamiltonian operator is described in **Algorithm** 5. We specify a priori a parameter τ that controls the number of tensor trains added before truncation.

Algorithm 5 TTO representation of $\mathbf{H}$

Input: One-electron integrals h_{ij}, two electron integrals v_{ijkl}, τ, creation and annihilation operators in the TTO-format (rank-1 tensor trains).

Initialization: $count = 0$, empty TT-cores $\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_d$.

Output: Compressed TTO representation $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$.

procedure TTO REPRESENTATION OF $\mathbf{H}$
  for $i, j = 1$ to d **do**
    Add rank-1 TTs obtained from $(h_{ij}\mathbf{A}_i^*\mathbf{A}_j)$ to the TT-cores using Proposition 1.2.2.
    $count = count + 1$.
    if $mod(count, \tau) = 0$ **then**
      Reduce the TTO-ranks of the TT-cores with Algorithm 2.
    end if
  end for
  for all pairs $(i, j, k, \ell) = 1$ to d **do**
    Add rank-1 TTs obtained from $(v_{ijkl}\mathbf{A}_i^*\mathbf{A}_k^*\mathbf{A}_\ell\mathbf{A}_j)$ to the TT-cores using Proposition 1.2.2.
    $count = count + 1$.
    if $mod(count, \tau) = 0$ **then**
      Reduce the TTO-ranks of the TT-cores with Algorithm 2.
    end if
  end for
end procedure

Remark 7. It is possible to improve Algorithm 5 by neglecting small values of one-electron integrals and two-electron integrals, and taking into account all of the symmetries that appear in these integrals. This includes the two-fold symmetry of one-electron integrals and the eight-fold symmetry of two-electron integrals, see Equations (1.88), (1.89).

Most generic methods rely on arithmetic operations between different TTO representations, but the compression techniques may vary [52]. It has been shown that SVD often provides quasi-optimal approximation, i.e best-approximation, of a TTO with TTO-ranks in alignment with the theoretical ranks. However, it should be applied with caution, as it can lead to the destruction of the key properties of the Hamiltonian operator that we will explain in the following section.

3.2.4 Numerical instabilities of rounding process and theoretical conditions

Constructing the TTO representation of the quantum chemical Hamiltonian operator through low-rank approximation methods, such as truncated SVD, can result in the non-conservation of the operator's properties, as defined in Section 3.2.1. These properties include the symmetry of the Hamiltonian operator matrix, the conservation of particle number, the conservation of the z-component of the total spin, the conservation of the total spin, and the operator's structure, which involves, here, at most 2-body interactions. A visual representation of these compromised properties is presented in Figures 3.2(a), 3.2(c), 3.2(b), and 3.1.

To address the preservation of symmetry, particle number conservation, and the z-component of the total spin, the observed phenomenon in Figures 3.2(a), 3.2(b), and 3.1 can be linked to the careless compression that occurs during the truncation process, especially when dealing with degenerate singular values, i.e the presence of equal singular values when applying SVD. When using a fixed threshold value, denoted as δ, for the truncation of singular values, it becomes possible to unintentionally discard other degenerate singular values. This is illustrated by the *Careless truncation* shown in the Figures 3.2(a), 3.2(b), and 3.1. This shows to significantly impact the properties of the original operator. Moreover, numerical observations indicate that this issue has significant implications for the eigenvalue problem, as well, at each microstep within the DMRG algorithm, as described in Chapter 1. During this step, it is essential that the effective Hamiltonian, as denoted by $\mathbf{M}_k, k \in [d]$ in (1.156), remains symmetric, as elaborated in Proposition 1.3.3. To address this challenge, two potential solutions can be proposed in practice. First, one may introduce a symmetrization step for each microstep in the DMRG iterations for $\mathbf{M}_k$. Alternatively, if the truncation lies within a set of degenerate singular values based on the numerical threshold δ, the degenerate subspace, defined by the set of singular vectors associated to the same singular value will either be entirely truncated or entirely kept. To further emphasize the importance of considering degeneracy in the truncation process, we offer the following theoretical insights, particularly focusing on the preservation of symmetry within the Hamiltonian operator matrix.

Theorem 3. Consider a symmetric matrix $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ representing the Hamiltonian operator, and satisfying Equation (S). Suppose $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ represents its compressed TTO decomposition with accuracy ϵ, obtained through truncated SVD while preserving the degenerate subspaces. Then, the approximated matrix, denoted as $\mathbf{H}_\epsilon \in \mathbb{R}^{n^d \times n^d}$, resulting from the TTO decomposition is also symmetric and satisfies Equation (S).

To prove Theorem 3, we start by introducing the Proposition 3.2.1. Let us first establish the mathematical concept of a tensor with invariant entries under a given permutation of its indices.

Consider a d-th order tensor $\mathcal{H} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$, $n_k \in \mathbb{N}, k \in [d]$ with entries that remain invariant with respect to a permutation of its indices, denoted by π. This permutation can be represented by the permutation operator $\hat{P}_\pi : \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d} \to \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$, and we have for $\mu_k \in [n_k], \mu_{\pi(k)} \in \left[n_{\pi(k)}\right], k \in [d]$:

$$\begin{aligned}\mathcal{H} = \hat{P}_\pi(\mathcal{H}) &\Leftrightarrow \mathcal{H}(\mu_1; \ldots; \mu_d) = \mathcal{H}(\mu_{\pi(1)}; \ldots; \mu_{\pi(d)}) \\ &\Leftrightarrow \mathbf{H}^{<k>}(\overline{\mu_1, \ldots, \mu_k}; \overline{\mu_{k+1}, \ldots, \mu_d}) = (\hat{P}_\pi(\mathcal{H}))^{<k>}(\overline{\mu_{\pi(1)}, \ldots, \mu_{\pi(k)}}; \overline{\mu_{\pi(k+1)}, \ldots, \mu_{\pi(d)}}).\end{aligned} \tag{3.17}$$

We introduce permutation matrices $\mathbf{P}_\pi^{\leq k} \in \mathbb{R}^{(n_1 \ldots n_k) \times (n_1 \ldots n_k)}$, and $\mathbf{P}_\pi^{>k} \in \mathbb{R}^{(n_{k+1} \ldots n_d) \times (n_{k+1} \ldots n_d)}$, $k \in [d]$. These matrices can be defined entry-wise as follows:

$$\begin{aligned}&\mathbf{P}_\pi^{\leq k}(j; \mu) = 1 \text{ if } j = \pi(\mu) = \mu_\pi \text{ otherwise } \mathbf{P}_\pi^{\leq k}(j; \mu) = 0, \\ &\text{with } \mu = \overline{\mu_1, \ldots, \mu_k}, \text{ and } \mu_\pi = \overline{\mu_{\pi(1)}, \ldots, \mu_{\pi(k)}},\end{aligned} \tag{3.18}$$

and

$$\begin{aligned}&\mathbf{P}_\pi^{>k}(\mu; j) = 1 \text{ if } j = \pi(\mu) = \mu_\pi, \text{ otherwise } \mathbf{P}_\pi^{>k}(\mu; j) = 0 \\ &\text{with } \mu = \overline{\mu_{k+1}, \ldots, \mu_d}, \text{ and } \mu_\pi = \overline{\mu_{\pi(k+1)}, \ldots, \mu_{\pi(d)}}.\end{aligned} \tag{3.19}$$

Thus we have:

$$\mathbf{H}^{<k>} = \mathbf{P}_\pi^{\leq k} \mathbf{H}^{<k>} \mathbf{P}_\pi^{>k}. \tag{3.20}$$

Here, $\mathbf{H}^{<k>} \in \mathbb{R}^{(n_1 \ldots n_k) \times (n_{k+1} \ldots n_d)}$ is the mode-$(1:k)$ matricization of $\mathcal{H}$. Moving forward, let $\{\sigma_\ell\}_{\ell \in [p]}$ be distinct singular values of the SVD of $\mathbf{H}^{<k>}$, each with respective multiplicities $\{m_l\}_{l \in [p]}$. The SVD of $\mathbf{H}^{<k>}$ can be represented as:

$$\mathbf{H}^{<k>} = \sum_{i=1}^{p} \sigma_i \mathbf{U}_i \mathbf{V}_i^*, \quad \mathbf{U}_i \in \mathbb{R}^{n_1 \ldots n_k \times m_i}, \mathbf{V}_i \in \mathbb{R}^{n_{k+1} \ldots n_d \times m_i}. \tag{3.21}$$

The expression (3.21) can also be defined entry-wise as follows:

$$\mathbf{H}^{<k>}(\overline{\mu_1, \ldots, \mu_k}; \overline{\mu_{k+1}, \ldots, \mu_d}) = \sum_{i=1}^{p} \sigma_i \left(\sum_{j=1}^{m_i} \mathbf{U}_i(\overline{\mu_1, \ldots, \mu_k}; j) \mathbf{V}_i^*(j; \overline{\mu_{k+1}, \ldots, \mu_d}) \right), \tag{3.22}$$

and the expression of the SVD of the permuted matrix can be defined entry-wise as follows:

$$\begin{aligned} &(\mathbf{P}_\pi^{\leq k} \mathbf{H}^{<k>} \mathbf{P}_\pi^{>k})(\overline{\mu_{\pi(1)}, \ldots, \mu_{\pi(k)}}; \overline{\mu_{\pi(k+1)}, \ldots, \mu_{\pi(d)}}) \\ &= \sum_{i=1}^{p} \sigma_i \left(\sum_{j=1}^{m_i} \mathbf{U}_i(\overline{\mu_{\pi(1)}, \ldots, \mu_{\pi(k)}}; j) \mathbf{V}_i^*(j; \overline{\mu_{\pi(k+1)}, \ldots, \mu_{\pi(d)}}) \right), \end{aligned} \tag{3.23}$$

with $\mu_k \in [n_k]$, $\mu_{\pi(k)} \in \left[n_{\pi(k)}\right]$ and $k \in [d]$. With these concepts in place, we can now introduce the following proposition.

Proposition 3.2.1. *If the d-order tensor $\mathcal{H} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ exhibits invariant entries under a specified index permutation denoted as π, and $\{\sigma_l\}_{l \in [p]}$ represent distinct singular values of its mode-(1:k) matricization, denoted as $\mathbf{H}^{<k>}$, with corresponding multiplicities $\{m_l\}_{l \in [p]}$, then the orthogonal matrices resulting from the SVD of $\mathbf{H}^{<k>}$, as outlined in Equation* (3.22)*, satisfy the following equality: for all $k \in [d]$, $\mu_k \in [n_k]$, $\mu_{\pi(k)} \in \left[n_{\pi(k)}\right]$ and $i \in [p]$:*

$$\sum_{j=1}^{m_i} \mathbf{U}_i(\overline{\mu_1, \ldots, \mu_k}; j) \mathbf{V}_i^*(j; \overline{\mu_{k+1}, \ldots, \mu_d}) = \sum_{j=1}^{m_i} \mathbf{U}_i(\overline{\mu_{\pi(1)}, \ldots, \mu_{\pi(k)}}; j) \mathbf{V}_i^*(j; \overline{\mu_{\pi(k+1)}, \ldots, \mu_{\pi(d)}}). \tag{3.24}$$

Proof of Proposition 3.2.1. Suppose that $\{\sigma_l\}_{l \in [p]}$ are distinct singular values of the matrix $\mathbf{H}^{<k>}, k \in [d]$, with multiplicities $\{m_l\}_{l \in [p]}$. Suppose that $r \leq \min\left\{\prod_{l=1}^{k} n_l, \prod_{l=k+1}^{d} n_l\right\}$ is the rank of $\mathbf{H}^{<k>}$. We define the matrix $\mathbf{S}_r = \operatorname{diag}(\sigma_1 \mathbf{I}_{m_1}, \sigma_2 \mathbf{I}_{m_2}, \ldots, \sigma_p \mathbf{I}_{m_p}) \in \mathbb{R}^{r \times r}$ such that $\mathbf{S} = \begin{bmatrix} \mathbf{S}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$. The SVD of $\mathbf{H}^{<k>}$ writes:

$$\mathbf{H}^{<k>} = \mathbf{U}\mathbf{S}\mathbf{V}^*. \tag{3.25}$$

with $\mathbf{U} \in \mathbb{R}^{(n_1 \ldots n_k) \times (n_1 \ldots n_k)}$ and $\mathbf{V} \in \mathbb{R}^{(n_{k+1} \ldots n_d) \times (n_{k+1} \ldots n_d)}$ being orthogonal matrices. Let $\hat{\mathbf{U}} \in \mathbb{R}^{(n_1 \ldots n_k) \times (n_1 \ldots n_k)}$ and $\hat{\mathbf{V}} \in \mathbb{R}^{(n_{k+1} \ldots n_d) \times (n_{k+1} \ldots n_d)}$ be orthogonal matrices as well. Now, according to Autonne's uniqueness theorem, see [49], if $\mathbf{H}^{<k>} = \mathbf{U}\mathbf{S}\mathbf{V}^* = \hat{\mathbf{U}}\mathbf{S}\hat{\mathbf{V}}^*$, then there exist orthogonal matrices $\mathbf{G}_1 \in \mathbb{R}^{m_1 \times m_1}, \ldots, \mathbf{G}_p \in \mathbb{R}^{m_p \times m_p}$, $\tilde{\mathbf{U}} \in \mathbb{R}^{\left(\prod_{l=1}^{k} n_l - r\right) \times \left(\prod_{l=1}^{k} n_l - r\right)}$, and $\tilde{\mathbf{V}} \in \mathbb{R}^{\left(\prod_{l=k+1}^{d} n_l - r\right) \times \left(\prod_{l=k+1}^{d} n_l - r\right)}$ such that we can define block-diagonal matrices $\mathbf{Q}_1 \in \mathbb{R}^{\left(\prod_{l=1}^{k} n_l\right) \times \left(\prod_{l=1}^{k} n_l\right)}$ and $\mathbf{Q}_2 \in \mathbb{R}^{\left(\prod_{l=k+1}^{d} n_l\right) \times \left(\prod_{l=k+1}^{d} n_l\right)}$ as:

$$\mathbf{Q}_1 = \operatorname{diag}\left(\mathbf{G}_1, \cdots, \mathbf{G}_p, \tilde{\mathbf{U}}\right), \ \mathbf{Q}_2 = \operatorname{diag}\left(\mathbf{G}_1, \cdots, \mathbf{G}_p, \tilde{\mathbf{V}}\right). \tag{3.26}$$

It follows:

$$\hat{\mathbf{U}} = \mathbf{U}\mathbf{Q}_1, \text{and}\, \hat{\mathbf{V}} = \mathbf{V}\mathbf{Q}_2. \tag{3.27}$$

Given (3.20), we have:

$$\mathbf{H}^{<k>} = \mathbf{U}\mathbf{S}\mathbf{V}^* = \mathbf{P}_\pi^{\leq k} \mathbf{U}\mathbf{S}\mathbf{V}^* \mathbf{P}_\pi^{>k}. \tag{3.28}$$

According to Equation (3.27), we can assert the following, for $i \in [p]$:

$$(\mathbf{U}_i \mathbf{G}_i)(\mathbf{V}_i \mathbf{G}_i)^* = \mathbf{P}_\pi^{\leq k} (\mathbf{U}_i \mathbf{G}_i)(\mathbf{V}_i \mathbf{G}_i)^* \mathbf{P}_\pi^{>k}, \tag{3.29}$$

where the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ are defined in Equation (3.21). Thus, we obtain:

$$\mathbf{U}_i \mathbf{V}_i^* = \mathbf{P}_\pi^{\leq k} \mathbf{U}_i \mathbf{V}_i^* \mathbf{P}_\pi^{>k}, \tag{3.30}$$

□

As indicated in Proposition 3.2.1, it becomes evident that when certain sets of degenerate singular values are disregarded, the Equation (3.24) may not remain true. This makes (3.24) an essential condition for preserving the matrix invariance under specific index permutations. With all the necessary components now in place, we can proceed to prove Theorem 3. This theorem ensures that preserving the degenerate singular values during the truncation process is key to achieving an approximate TT representation with preserved desired symmetry, as the one considered in Equation (S).

Proof of Theorem 3. Consider the matrix representation of a given Hamiltonian operator $\mathbf{H} \in \mathbb{R}^{(n_1,\dots,n_d)\times(n_1,\dots,n_d)}$. Suppose that $\mathbf{H}$ is symmetric, see (S). Let $\mathcal{H} \in \mathbb{R}^{n_1\times n_2\times\dots\times n_d\times n_1\times n_2\times\dots\times n_d}$ be the tensor folding of $\mathbf{H}$ and $\tilde{\mathcal{H}} \in \mathbb{R}^{n_1\times n_1\times n_2\times n_2\dots\times n_d\times n_d}$ the permutation of $\mathcal{H}$ over specific modes. Since $\mathbf{H}$ is symmetric, according to (3.24), we have $\hat{P}_\pi(\tilde{\mathcal{H}}) = \tilde{\mathcal{H}}$ and for $k \in [d]$, we have:

$$\tilde{\mathbf{H}}^{<2k>} = \mathbf{P}_\pi^{\leq 2k}\tilde{\mathbf{H}}^{<2k>}\mathbf{P}_\pi^{>2k}, \tag{3.31}$$

where $\tilde{\mathbf{H}}^{<2k>}$ is the mode-$(1:2k)$ matricization of $\tilde{\mathcal{H}}$ and where the permutation over indices denoted by π can be evaluated as follows:

$$\begin{cases}\pi(\ell) = \ell - 1, & \text{if } \ell \text{ is even}\\ \pi(\ell) = \ell + 1, & \text{if } \ell \text{ is odd,}\end{cases}, \forall \ell \in [2k]\,. \tag{3.32}$$

In this context, ℓ denotes the position of a specific index within the multi-index used to label an entry in the tensor, as seen in $\tilde{\mathcal{H}}(\mu_1;\nu_1;\dots;\mu_d;\nu_d)$. For instance, the position of ν_1 is indicated by $\ell = 2$. Now for $k = 1$, let $\{\sigma_{1,l}\}_{l\in[p_1]}$ be the distinct singular values of $\tilde{\mathbf{H}}^{<2>} \in \mathbb{R}^{(n_1n_1)\times(n_2n_2\dots n_dn_d)}$ with respective multiplicities $\{m_{1,l}\}_{l\in[p_1]}$ and let $\mathbf{S}_1 = \operatorname{diag}(\sigma_{1,1}\mathbf{I}_{m_{1,1}}, \sigma_{1,2}\mathbf{I}_{m_{1,2}}, \dots, \sigma_{1,p_1}\mathbf{I}_{m_{1,p_1}})$. Let R_1 be the rank of $\tilde{\mathbf{H}}^{<2>}$. The SVD of $\tilde{\mathbf{H}}^{<2>}$ writes:

$$\begin{aligned}\tilde{\mathbf{H}}^{<2>} &= \underbrace{\mathbf{U}_1}_{\in\mathbb{R}^{(n_1n_1)\times R_1}}\ \underbrace{\mathbf{S}_1\mathbf{V}_1^*}_{\in\mathbb{R}^{R_1\times(n_2n_2n_3n_3\dots n_dn_d)}}\\ &\Leftrightarrow \tilde{\mathbf{H}}^{<2>}(\overline{\mu_1,\nu_1};\overline{\mu_2,\nu_2,\dots,\mu_d,\nu_d}) = \sum_{\alpha_1=1}^{R_1}\mathbf{U}_1(\overline{\mu_1,\nu_1};\alpha_1)\,(\mathbf{S}_1\mathbf{V}_1^*)\,(\alpha_1;\overline{\mu_2,\nu_2,\dots,\mu_d,\nu_d}).\end{aligned} \tag{3.33}$$

Using Proposition 3.2.1, we have for $i_1 \in [p_1]$:

$$\sum_{j_1=1}^{m_{1,i_1}}\mathbf{U}_{1,i_1}(\overline{\mu_1,\nu_1};j_1)\mathbf{V}_{1,i_1}^*(j_1;\overline{\mu_2,\nu_2,\dots,\mu_d,\nu_d}) = \sum_{j_1=1}^{m_{1,i_1}}\mathbf{U}_{1,i_1}(\overline{\nu_1,\mu_1};j_1)\mathbf{V}_{1,i_1}^*(j_1;\overline{\nu_2,\mu_2,\dots,\nu_d,\mu_d}), \tag{3.34}$$

where $\mathbf{U}_{1,i_1} \in \mathbb{R}^{(n_1n_1)\times m_{1,i_1}}$ and $\mathbf{V}_{1,i_1} \in \mathbb{R}^{(n_2n_2\dots n_dn_d)\times m_{1,i_1}}$.

We denote $\mathbf{W} = \mathbf{S}_1\mathbf{V}_1^* \in \mathbb{R}^{R_1\times(n_2n_2n_3n_3\dots n_dn_d)}$. Let $\mathcal{W} \in \mathbb{R}^{R_1\times n_2\times n_2\times n_3\times n_3\dots\times n_d\times n_d}$ be the tensor folding of $\mathbf{W}$. The next step in TT-SVD algorithm consists in applying an SVD over $\mathbf{W}^{<3>}$ of rank R_2 such that

$$\mathbf{W}^{<3>} = \underbrace{\mathbf{U}_2}_{\in\mathbb{R}^{(R_1n_2n_2)\times R_2}}\ \underbrace{\mathbf{S}_2\mathbf{V}_2^*}_{\in\mathbb{R}^{R_2\times(n_3n_3\dots n_dn_d)}}\quad. \tag{3.35}$$

Given (3.35), we can restore the elements of $\tilde{\mathcal{H}}$ as follows:

$$\begin{aligned}\tilde{\mathcal{H}}(\mu_1;\nu_1;\dots;\mu_d;\nu_d) &= \sum_{\alpha_1=1}^{R_1}\sum_{\alpha_2=1}^{R_2}\mathbf{U}_1(\overline{\mu_1,\nu_1};\alpha_1)\mathbf{U}_2(\overline{\alpha_1,\mu_2,\nu_2};\alpha_2)\,(\mathbf{S}_2\mathbf{V}_2^*)\,(\alpha_2;\overline{\mu_3,\nu_3,\dots,\mu_d,\nu_d})\\ &\Leftrightarrow \tilde{\mathbf{H}}^{<4>} = \tilde{\mathbf{U}}_2\,(\mathbf{S}_2\mathbf{V}_2^*)\,,\end{aligned} \tag{3.36}$$

where $\tilde{\mathbf{U}}_2 \in \mathbb{R}^{(n_1n_1n_2n_2)\times R_2}$ is the mode-$(1:4)$ matricization of $\mathcal{U}_1 \times_3 \mathcal{U}_2$, with $\mathcal{U}_1$ and $\mathcal{U}_2$ being the tensor folding of the matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ respectively. Using Proposition 3.2.1, we have:

$$\sum_{j_2=1}^{m_{2,i_2}}\tilde{\mathbf{U}}_{2,i_2}(\overline{\mu_1,\nu_1,\mu_2,\nu_2};j_2)\mathbf{V}_{2,i_2}^*(j_2;\overline{\mu_3,\nu_3,\dots,\mu_d,\nu_d}) = \sum_{j_2=1}^{m_{2,i_2}}\tilde{\mathbf{U}}_{2,i_2}(\overline{\nu_1,\mu_1,\nu_2,\mu_2};j_2)\mathbf{V}_{2,i_2}^*(j_2;\overline{\nu_3,\mu_3,\dots,\nu_d,\mu_d}), \tag{3.37}$$

where $\tilde{\mathbf{U}}_{2,i_2} \in \mathbb{R}^{(n_1 n_1 n_2 n_2)\times m_{2,i_2}}$ and $V_{2,i_2} \in \mathbb{R}^{(n_3 n_3 \ldots n_d n_d)\times m_{2,i_2}}, i_2 \in [p_2]$. Then, by construction, at iteration $k \in [d]$, it follows:

$$\begin{aligned}\tilde{\mathcal{H}}(\mu_1;\nu_1;\ldots;\mu_d;\nu_d) = &\sum_{\alpha_1=1}^{R_1}\sum_{\alpha_2=1}^{R_2}\ldots\sum_{\alpha_k=1}^{R_k}\mathbf{U}_1(\overline{\mu_1,\nu_1};\alpha_1)\mathbf{U}_2(\overline{\alpha_1,\mu_2,\nu_2};\alpha_2)\\ &\ldots\mathbf{U}_k(\overline{\alpha_{k-1},\mu_k,\nu_k};\alpha_k)\,(\mathbf{S}_k\mathbf{V}_k^*)\,(\alpha_k;\overline{\mu_{k+1},\nu_{k+1},\ldots,\mu_d,\nu_d})\\ &\Leftrightarrow \tilde{\mathbf{H}}^{<2k>} = \tilde{\mathbf{U}}_k\,(\mathbf{S}_k\mathbf{V}_k^*)\,,\end{aligned} \tag{3.38}$$

where $\tilde{\mathbf{U}}_k \in \mathbb{R}^{(n_1 n_1 n_2 n_2 \ldots n_k n_k)\times R_k}$ represents the mode-$(1:2k)$ matricization of $\mathcal{U}_1 \times_3 \mathcal{U}_2 \times_4 \ldots \times_4 \mathcal{U}_k$. Using Proposition 3.2.1, we have:

$$\begin{aligned}&\sum_{j_k=1}^{m_{k,i_k}}\tilde{\mathbf{U}}_{k,i_k}(\overline{\mu_1,\nu_1,\mu_2,\nu_2,\ldots,\mu_d,\nu_d};j_k)\mathbf{V}_{k,i_k}^*(j_k;\overline{\mu_{k+1},\nu_{k+1},\ldots,\mu_d,\nu_d})\\ &=\sum_{j_k=1}^{m_{k,i_k}}\tilde{\mathbf{U}}_{k,i_k}(\overline{\nu_1,\mu_1,\nu_2,\mu_2,\ldots,\nu_d,\mu_d};j_k)\mathbf{V}_{k,i_k}^*(j_k;\overline{\nu_{k+1},\mu_{k+1},\ldots,\nu_d,\mu_d}),\end{aligned} \tag{3.39}$$

with $\tilde{\mathbf{U}}_{k,i_k} \in \mathbb{R}^{(n_1 n_1 \ldots n_k n_k)\times m_{k,i_k}}$ and $\mathbf{V}_{k,i_k} \in \mathbb{R}^{(n_{k+1} n_{k+1} \ldots n_d n_d)\times m_{k,i_k}}, i_k \in [p_k]$. Hence, the condition in (3.39) must be verified at each iteration $k \in [d]$ in order to obtain a symmetric Hamiltonian operator matrix. Therefore, if the degenerate subspaces are not respected during the compression process, the condition in (3.39) is not verified which yields to loosing the symmetry property. □

As per Theorem 3, it is observed that a careless truncation during the SVD process can lead to the loss of symmetry property of the Hamiltonian operator matrix. Furthermore, we numerically observe that this also affects its commutation relations with the particle number operator represented by $\mathbf{N}$ as well as the z-component of the spin operator represented by $\mathbf{S}^z$, see Figure 3.2. We can illustrate this with a straightforward example in the spin-orbital basis for simplification, which demonstrates that failing to preserve the degenerate singular values disrupts the inherent block-sparse structure that arises when the Hamiltonian operator commutes with the particle number operator. We consider the following trivial example for $d = 2$, $n_1 = n_2 = 2$.

Example 3.2.1. Let $\mathbf{H} \in \mathbb{R}^{2^2\times 2^2}$. Suppose that $\mathbf{H}$ and the particle number operator $\mathbf{N}$ commute, as shown by (PN). This implies that for $\mu_k, \nu_k, k \in \{1, 2\}$:

$$\mathbf{NH} - \mathbf{HN} = 0 \Leftrightarrow \mathbf{H}(\overline{\mu_1,\mu_2};\overline{\nu_1,\nu_2}) \neq 0 \Rightarrow \sum_{k=1}^{2} q_2(\mu_k) = \sum_{k=1}^{2} q_2(\nu_k), \tag{3.40}$$

where $q_2 : \{1, 2\} \mapsto \{0, 1\}$ is a mapping function between tensor indices and occupation numbers in the spin-orbital basis. Here, we can observe that the Hamiltonian operator $\mathbf{H}$, with entries $\mathbf{H}(\overline{\mu_1,\mu_2};\overline{\nu_1,\nu_2})$, has a block-sparse structure represented as follows:

$$\mathbf{H} = \begin{pmatrix}\mathbf{H}(\overline{1,1};\overline{1,1}) & 0 & 0 & 0\\ 0 & \mathbf{H}(\overline{1,2};\overline{1,2}) & \mathbf{H}(\overline{1,2};\overline{2,1}) & 0\\ 0 & \mathbf{H}(\overline{2,1};\overline{1,2}) & \mathbf{H}(\overline{2,1};\overline{2,1}) & 0\\ 0 & 0 & 0 & \mathbf{H}(\overline{2,2};\overline{2,2})\end{pmatrix}. \tag{3.41}$$

Now, let $\mathcal{H} \in \mathbb{R}^{2\times 2\times 2\times 2}$ be the tensor folding of $\mathbf{H}$, and $\tilde{\mathcal{H}}$ be the permuted tensor defined by its entries $\tilde{\mathcal{H}}(\mu_1;\nu_1;\mu_2;\nu_2), \mu_k, \nu_k, k \in \{1, 2\}$. The mode-(1:2) of $\tilde{\mathcal{H}}$, denoted as $\tilde{\mathbf{H}}^{<2>}$ with entries $\tilde{\mathbf{H}}^{<2>}(\overline{\mu_1,\nu_1};\overline{\mu_2,\nu_2})$, also exhibits a block-sparse structure represented as follows:

$$\tilde{\mathbf{H}}^{<2>} = \begin{pmatrix}\tilde{\mathbf{H}}(\overline{1,1};\overline{1,1}) & \tilde{\mathbf{H}}(\overline{1,1};\overline{2,2}) & 0 & 0\\ \tilde{\mathbf{H}}(\overline{2,2};\overline{1,1}) & \tilde{\mathbf{H}}(\overline{1,2};\overline{1,2}) & 0 & 0\\ 0 & 0 & 0 & \tilde{\mathbf{H}}(\overline{1,2};\overline{2,1})\\ 0 & 0 & \tilde{\mathbf{H}}(\overline{2,1};\overline{1,2}) & 0\end{pmatrix}. \tag{3.42}$$

To compress $\tilde{\mathbf{H}}^{<2>}$, one can apply a truncated SVD to each block separately. Let us focus on the second block, that we denote by $\mathbf{B}$, which can be expressed as:

$$\mathbf{B} = \begin{pmatrix}0 & \tilde{\mathbf{H}}(\overline{1,2};\overline{2,1})\\ \tilde{\mathbf{H}}(\overline{2,1};\overline{1,2}) & 0\end{pmatrix}. \tag{3.43}$$

Due to symmetry, we have $\tilde{\mathbf{H}}(\overline{1,2};\overline{2,1}) = \tilde{\mathbf{H}}(\overline{2,1};\overline{1,2})$, which allows us to represent the SVD as follows:

$$\mathbf{B} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{v}_2^* + \lambda_1 \boldsymbol{u}_2 \boldsymbol{v}_1^*. \tag{3.44}$$

Here, λ_1 refers to the degenerate singular value, and $\boldsymbol{u}_1$, $\boldsymbol{v}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{v}_2$ are the singular vectors associated with the singular value λ_1. Since λ_1 is degenerate, there is more freedom in choosing the singular vectors. In other words, any vector that is a linear combination of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_2$ is a singular vector associated with λ_1, satisfying

$$\mathbf{B} = \lambda_1(\alpha \boldsymbol{u}_1 + \beta \boldsymbol{u}_2)(\alpha \boldsymbol{v}_2^* + \beta \boldsymbol{v}_1^*) + \lambda_1(\alpha \boldsymbol{u}_2 - \beta \boldsymbol{u}_1)(\alpha \boldsymbol{v}_1^* - \beta \boldsymbol{v}_2^*), \tag{3.45}$$

where $\alpha^2 + \beta^2 = 1$. When retaining only the first singular value during truncation, it leads to $\mathbf{B}(1;1) \neq 0$ and $\mathbf{B}(2;2) \neq 0$ resulting in the failure to preserve the particle number as well as the block-sparse structure.

In what follows, we give error portraits using two truncation methods: the *Careful truncation* (which maintains degenerate singular values and preserves the desired properties of the Hamiltonian operator) and *Careless truncation* . In Figures 3.2(b),3.2(a),and 3.2(c), we illustrate how the careless truncation impacts the commutation relations involving $\mathbf{H}$ with $\mathbf{N}$, $\mathbf{S}^z$, and $\mathbf{S}^2$. Additionally, we evaluate the relative compression error between the original Hamiltonian operator $\mathbf{H}$ and its compressed format, denoted as $\mathbf{H}_\epsilon$, with ϵ representing the desired accuracy. In the figures presented below, we focus on a H_4 molecule with a dimension of $d = 8$. We have intentionally chosen a small system size to facilitate the evaluation of numerical compression errors. Forming the complete $\mathbf{H}$ for larger systems can be computationally expensive. Nonetheless, this choice provides insights into potential numerical issues that can be encountered when dealing with larger systems.

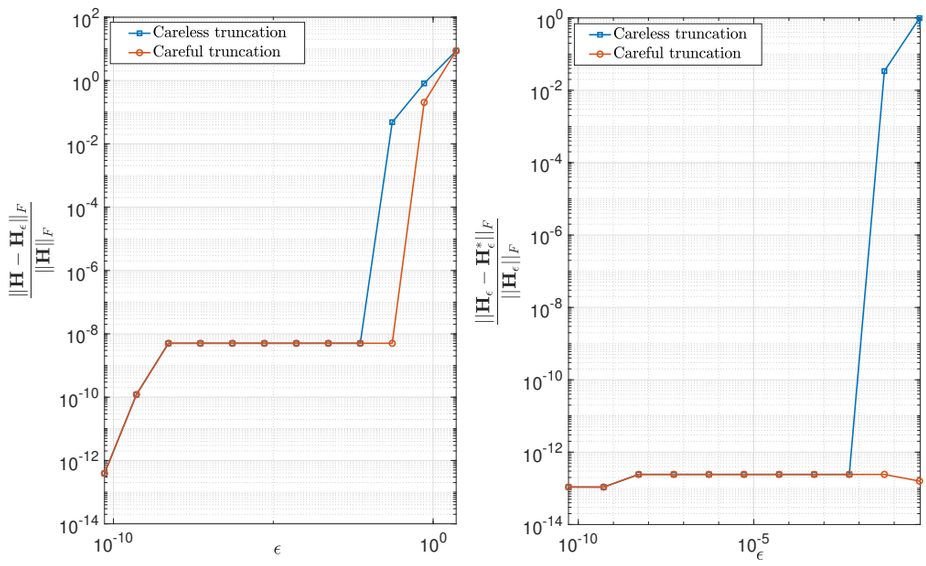
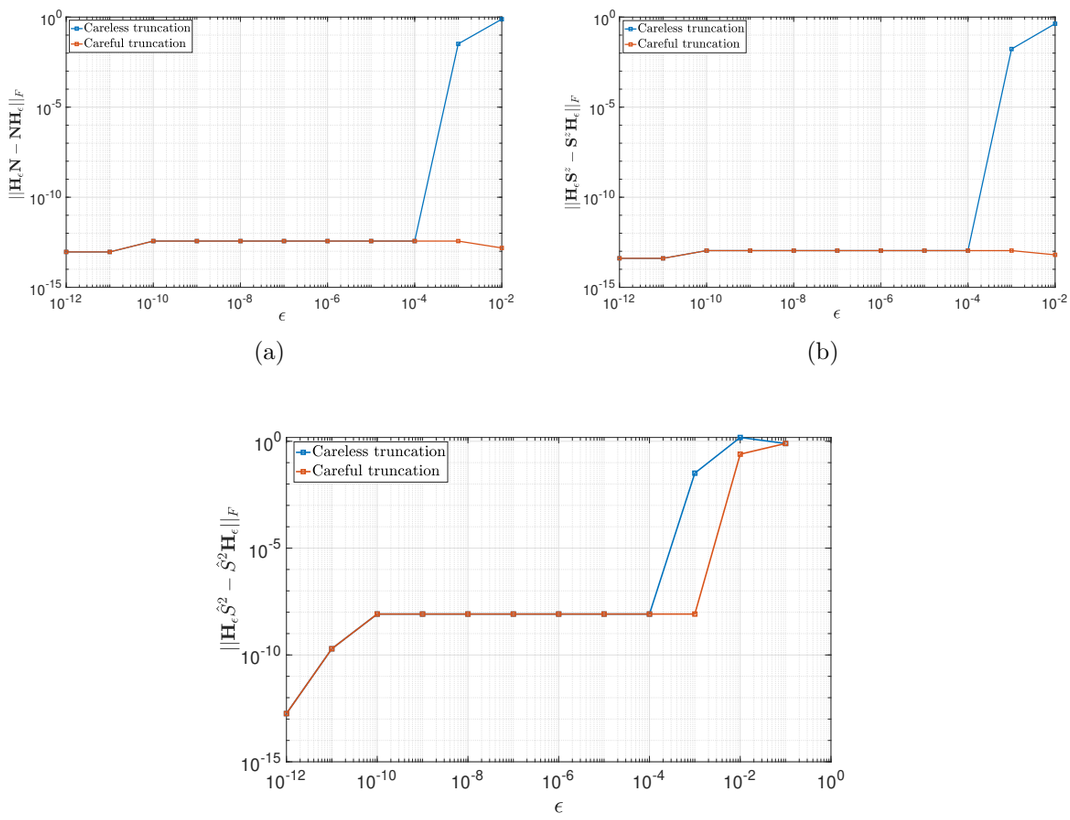


Figure 3.1: The left figure displays how the relative error in Frobenius norm between the original operator matrix $\mathbf{H}$ and the compressed matrix $\mathbf{H}_\epsilon$ changes with varying accuracy ϵ. The right figure illustrates the variation in the relative error between $\mathbf{H}_\epsilon$ and its transpose $\mathbf{H}_\epsilon^*$ as accuracy ϵ varies.



(a) (b)

(c)

Figure 3.2: The figures above illustrate the impact of truncation on the commutation relations between $\mathbf{H}_\epsilon$ and various operators: particle number operator $\mathbf{N}$ (a), the z-component of spin operator $\mathbf{S}^z$ (b), and the total spin operator $\hat{S}^2$(c).

In Figure 3.2, we observe that careful implementation of truncation during compression results in the preservation of the Hamiltonian operator's essential properties, including symmetry and the conservation of particle number, as well as the z-component of the total spin. However, for the conservation of the total spin, we currently lack a theoretical explanation for the observed behaviour in Figure 3.2(c). It is evident however that the error varies with the chosen accuracy ϵ.

We also observe, as illustrated in Figure 3.3, that compression at certain threshold, $\epsilon > 10^{-5}$, can lead to the emergence of virtual interactions in a strictly one-body operator Hamiltonian (as shown in the left side of the Figure 3.3) or a two-body operator (as shown in the right side of the Figure 3.3). We briefly explain the evaluation of these interactions. For simplification, we consider the spin-orbital basis. A general p-body Hamiltonian operator, with at most p-body interactions can be expressed in terms of creation and annihilation operators as follows, see [95]:

$$\mathbf{H}_{\leq p} = \sum_{J,J' \subseteq X} \mathbf{C}(J;J') \prod_{\boldsymbol{x}\in J} \mathbf{A}^*_{\boldsymbol{x}} \prod_{\boldsymbol{x}'\in J'} \mathbf{A}_{\boldsymbol{x}'}, \tag{3.46}$$

where X represents a set of multi-indices defined by $X = \{(\xi_1, 0, \ldots, 0), (\xi_1, \xi_2, 0, \ldots, 0), \ldots, (\xi_1, \xi_2, \ldots, \xi_p)$ $\xi_k \in [d]\,, k \in [p]\,$, and $\#X = p$. $\boldsymbol{x}$ and $\boldsymbol{x}'$ represent a vector of indices such that

$$\mathbf{A}^*_{\boldsymbol{x}} = \mathbf{A}^*_{x_1} \ldots \mathbf{A}^*_{x_p}, \tag{3.47}$$

$$\mathbf{A}_{\boldsymbol{x}'} = \mathbf{A}_{x'_1} \ldots \mathbf{A}_{x'_p}, \tag{3.48}$$

with the convention $\mathbf{A}_{x_k} = \mathbf{I}_2$ if $x_k = 0$, for $k \in [p]$. The entries of $\mathbf{H}_{\leq p}$ expressed in (3.46) can be obtained by evaluating the following so called *vacuum expectation value*:

$$\begin{aligned}\mathbf{H}_{\leq p}(\boldsymbol{\alpha};\boldsymbol{\alpha}') &= \sum_{J,J'\subseteq X} \mathbf{C}(J;J') \underbrace{\boldsymbol{\Phi}^*_{0\ldots0} \left(\prod_{\boldsymbol{\alpha}\in J} \mathbf{A}_{\boldsymbol{\alpha}} \prod_{\boldsymbol{x}\in J} \mathbf{A}^*_{\boldsymbol{x}} \prod_{\boldsymbol{x}'\in J'} \mathbf{A}_{\boldsymbol{x}'} \prod_{\boldsymbol{\alpha}'\in J'} \mathbf{A}^*_{\boldsymbol{\alpha}'} \right) \boldsymbol{\Phi}_{0\ldots0}}_{\mathbf{M}(\overline{\boldsymbol{\alpha},\boldsymbol{\alpha}'};\overline{\boldsymbol{x},\boldsymbol{x}'})} \\ &= \sum_{\boldsymbol{x}\in J,\boldsymbol{x}'\in J'} \mathbf{C}(\boldsymbol{x};\boldsymbol{x}') \mathbf{M}(\overline{\boldsymbol{\alpha},\boldsymbol{\alpha}'};\overline{\boldsymbol{x},\boldsymbol{x}'}).\end{aligned} \tag{3.49}$$

with $\mathbf{\Phi}_{0\ldots0} = \boldsymbol{e}_1 \otimes_K \ldots \otimes_K \boldsymbol{e}_1 \in \mathbb{R}^{2^d}$, where $\boldsymbol{e}_1$ are unit vectors introduced in (1.119). $\mathbf{H}_{\leq p}$ can be obtained through matrix-vector multiplication as follows:

$$\mathrm{vec}(\mathbf{H}_{\leq p}) = \mathbf{M}\,\mathrm{vec}(\mathbf{C}). \tag{3.50}$$

For instance, in the case of a 1-body operator with $p = 1$, see Equation (1.130):

$$\mathbf{H}_{\leq 1} = \sum_{i,j=1}^{d} h_{ij}\mathbf{A}_i^*\mathbf{A}_j. \tag{3.51}$$

To align with the general expression of a p-body operator, we have here $X = \{(\xi_1)\}$, $J = (\xi_1)$, $J' = (\xi_1')$, and $\mathbf{C}(J;J') = \mathbf{C}(\xi_1;\xi_1') = h_{\xi_1\xi_1'}$. One way to identify undesired interactions is by following these steps

- Compute the resulting compressed Hamiltonian operator from a compressed TTO decomposition with truncated SVD at accuracy ϵ. The compressed matrix is denoted as $(\mathbf{H}_{\leq p})_\epsilon$.
- Evaluate the matrix $\mathbf{M}$ defined in the Equation (3.49). The entries of this matrix can be evaluated using Wick's theorem. More details about this theorem can be found in [108].
- Evaluate the matrix containing all the new coefficients, denoted as $\tilde{\mathbf{C}}$, using Equation (3.50).
- Recompute the matrix $(\tilde{\mathbf{H}}_{\leq p})_\epsilon$ using the Equation (3.46), and the matrix $\tilde{\mathbf{C}}$.
- Evaluate the relative error between $(\mathbf{H}_{\leq p})_\epsilon$ and $(\tilde{\mathbf{H}}_{\leq p})_\epsilon$.

We applied these steps to both 1-body and 2-body operator Hamiltonians, as illustrated in Figure 3.3.

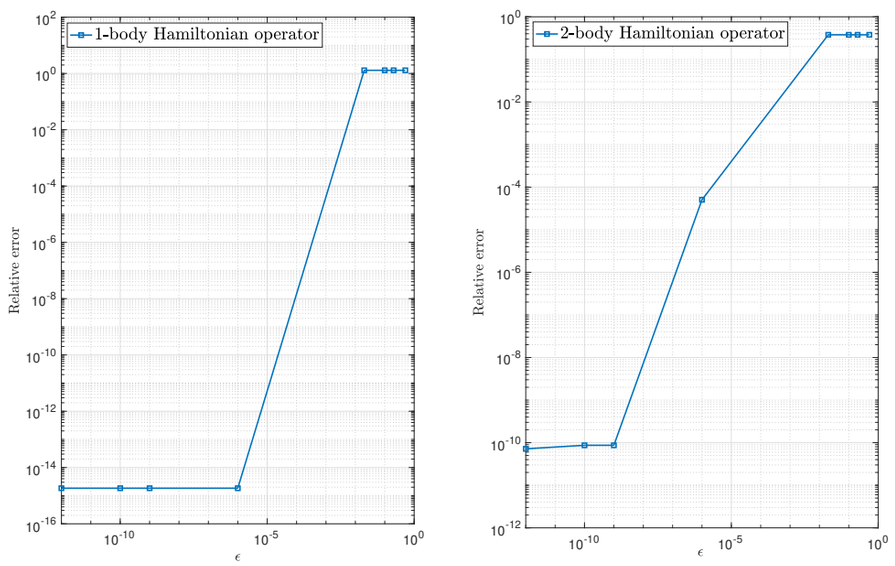


Figure 3.3: Presence of spurious interactions in a 1-body Hamiltonian operator (left figure) (resp. 2-body Hamiltonian operator (right figure)) at a fixed accuracy ϵ.

3.3 Symmetry-preserving TT representations

As discussed in the previous section, several different approaches exist for the efficient TTO construction of the Hamiltonian operator. Shifting our focus to compression-based methods for constructing a compressed TTO representation, the straightforward approach is the TT-SVD, elaborated upon in [92]. This algorithm involves applying multiple truncated SVDs on tensor unfoldings. Obviously, in practice TT-SVD is not suitable for our

application case since it requires the access to all the entries of the operator, yielding to an exponential scaling with the system size d. As previously discussed in the preceding section, we have also described an alternative approach that scales linearly with the system size d. This method leverages the structure of the Hamiltonian operator within the second quantization formalism, enabling it to be represented as a sum of rank-1 TTs. This approach begins with a tensor already in TT-format, with the goal of achieving a compressed representation that closely approximates the original one. To accomplish this, the TT-rounding algorithm, as described in [92], is employed. This approach does not require forming the entire tensor explicitly, however the time cost spent for numerical compression is not negligible when the number of terms in the sum is large (large TT-ranks). As demonstrated in previous numerical examples, a brute-force compression method, if done naively, may fail to preserve essential properties of the Hamiltonian operator, including the symmetry of the Hamiltonian operator matrix, the emergence of virtual interactions, and the preservation of the natural block structure that arises from particle number conservation for instance. Moreover, performing an SVD on a matrix with a high degree of sparsity will typically yield to a dense matrix.

An essential feature when handling operators arising in quantum chemistry is that their matrix representation exhibits a block-sparse structure when physical/group symmetries are exploited. This includes the conservation of the particle number conservation, the z-component of the total spin and the total spin. Incorporating these symmetries can lead to more effective TT structure, in particular a block-sparse TT representation. In this section, we provide an extensive review and a comprehensive study about the different structures that emerge in the TT representations of both the eigenfunction and the Hamiltonian operator through the exploitation of Abelian symmetries (conservation of the particle number and the z-component of total spin). Notably, the latter has a long history, particularly in the context of the representation of the eigenfunctions (MPS in physics language). This can lead to a block structure in the TT-cores of the TT decomposition of the eigenfunction, as well as in the TT-cores of the TTO representation of the Hamiltonian operator.

Indeed, such a structure has been previously exploited in the literature, see [2, 51, 60, 85, 86, 106, 107, 111, 112, 129, 134]. In particular, the authors of [2] have shown that within the context of Abelian symmetries, a block-sparse structure can emerge within the TT-cores of the TT decomposition of the eigenfunction. This occurs when the latter is an eigenvector of the particle number operator, i.e the particle number is conserved.

We will provide a concise overview of this specific structure. Additionally, inspired by the work of [2], we can extend it by showing that enforcing Abelian symmetries, such as particle number conservation or the conservation of the z-component of the total spin, results in an interesting block-sparse structure in the TT-cores of the TTO representation of a general p-body Hamiltonian operator. To the best of our knowledge, this structure has not been presented in this way before. Nevertheless, we recognize a comparable structure was shown in the study mentioned in [60]. The block-sparse TT representations of both the eigenfunction and the Hamiltonian operator offer significant memory and computational advantages, as will be detailed in Chapter 4:

- it minimizes memory footprint: dense blocks in the TT representation are stored into a list format,
- it allows to efficiently perform block-wise compression and orthogonalization, which can be performed in parallel,
- it is particularly useful when carrying out block-wise contractions with block-structured tensor trains which is a commonly used operation in QC-DMRG,
- it is beneficial to speed-up matrix-vector operations encountered when solving the reduced eigenvalue problem, see Equation (1.156).

Extending these structures to non-Abelian symmetries, such as SU(2) corresponding to the conservation of the total spin, is less straightforward since more complex algebra is involved. We show that the incorporation of non-Abelian symmetries results as well in a block structured TT representations with fewer blocks. However, it requires the

introduction of additional coupling coefficients, which we will elaborate on in the Section 3.3.2, specifically for the TT representation of the eigenfunction which is considered as an eigenvector of the total spin operator represented by $\hat{S}^2$.

3.3.1 Block-sparse TT representations

Block-sparse structure in the TT decomposition of the eigenfunction

In this section, we present an interesting Corollary that will be employed throughout this thesis. This Corollary is derived from a theorem established in [2], with slight modifications made to align its notations with those used in this manuscript. We introduce this Corollary specifically for the case involving the spin-orbital basis, assuming $n = 2$, yet it can similarly be extended to the spatial-orbital basis.

Corollary 3.3.1. *[2] Let $\mathbf{\Psi} \in \boldsymbol{\mathcal{F}}_d$, $\mathbf{\Psi} \neq \mathbf{0}$ be an eigenvector of the particle number operator represented by $\mathbf{N}$, see Equation (1.131), such that*

$$\mathbf{N}\mathbf{\Psi} = N\mathbf{\Psi}, N \in [d]\,. \tag{3.52}$$

Let $\psi \in \mathbb{R}^{n\times n\times\ldots\times n}$, with $n = 2$, be the d-order tensor folding of $\mathbf{\Psi}$. Let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ be the TT decomposition of ψ with TT-ranks $(r_1, \ldots, r_d)$.

Then the TT-cores have a block-sparse structure with at most $N+1$ nonzero blocks such that the following holds: for $k \in [d]$, and for $\mu_k \in [n]$, the matrices $\mathbf{U}_k[\mu_k] := \mathcal{U}_k[:, \mu_k, :] \in \mathbb{R}^{r_{k-1}\times r_k}$ have nonzero entries only in the following blocks of sizes, denoted as $\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{\Psi}}_{k,i_k}$:

$$\mathbf{U}_k[\mu_k]|_{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}}\times\rho^{\mathbf{\Psi}}_{k,i_k}} := \mathbf{U}_k^{(i_{k-1},i_k)}[\mu_k] \neq \mathbf{0} \text{ if } i_{k-1} + q_n(\mu_k) = i_k, \tag{3.53}$$

where q_n is defined in (PN)*. Additionally, $i_0 = 0$, $i_d = N + 1$, and $\rho^{\mathbf{\Psi}}_{0,0} = \rho^{\mathbf{\Psi}}_{d,N+1} = 1$ and for $k \in \{2, \ldots, d-1\}$, we have:*

$$r_{k-1} = \sum_{i_{k-1}\in S^{\mathbf{\Psi}}_{k-1}} \rho^{\mathbf{\Psi}}_{k-1,i_{k-1}},\ r_k = \sum_{i_k\in S^{\mathbf{\Psi}}_k} \rho^{\mathbf{\Psi}}_{k,i_k}, \tag{3.54}$$

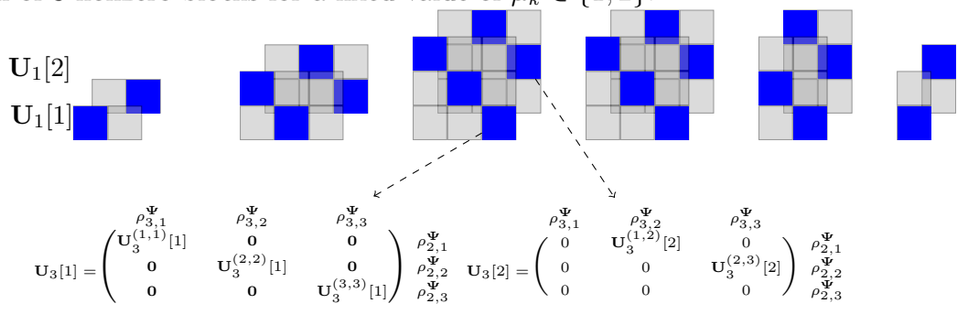
where

$$S^{\mathbf{\Psi}}_{k-1} = \{\max\{1, N-d+k\}, \ldots, \min\{N+1, k\}\}, \tag{3.55}$$

$$S^{\mathbf{\Psi}}_{k} = \{\max\{1, N-d+k+1\}, \ldots, \min\{N+1, k+1\}\}, \tag{3.56}$$

For clarification, we give in what follows an example to illustrate the block-sparse representation of the TT representation of an eigenvector of the particle number operator.

Example 3.3.1. Consider the case $d = 6$, $n = 2$, $k \in [d]$ and $N = 2$. Suppose that $\mathbf{\Psi}$ has a TT representation, such that $\forall\,(\mu_1, \ldots, \mu_d) \in \{1, 2\}^d$, $\psi(\mu_1; \cdots; \mu_d) = \mathbf{U}_1[\mu_1]\mathbf{U}_2[\mu_2]\cdots\mathbf{U}_d[\mu_d]$. The TT representation takes the following form, where each TT-core $\mathbf{U}_k[\mu_k]$ has a maximum of 3 nonzero blocks for a fixed value of $\mu_k \in \{1, 2\}$.



Remark 3.3.1. In the case of spatial-orbital basis, i.e $n = 4$ and $d_{\text{spatial}} = \frac{d}{2}$, a similar block-sparse TT representation can be derived, such that the TT-cores exhibit a block-sparse structure with at most $N + 1$ nonzero blocks.

Remark 3.3.2. $\mathbf{\Psi}$ acting as an eigenvector of the particle number operator with the eigenvalue N, is the wavefunction. An interesting interpretation of its TT representation as discussed in [77], is that this TT representation can be seen as a recursive orthogonal transformation of elements in the basis of the discrete Fock space $\boldsymbol{\mathcal{F}}_k$, as defined in (1.97), at each iteration k, with the TT-cores acting as successive linear maps, leading to the target wavefunction $\mathbf{\Psi}$ with a fixed particle number of N. For more details, we direct the reader to Appendix .5 for further elaboration.

Block-sparse structure of the TTO representation

In Section 3.2.1, we discussed the properties of the Hamiltonian operator that need to be preserved during compression. In what follows, we show that enforcing physical symmetries such as the conservation of the particle number or the z-component of the total spin, i.e the Hamiltonian operator satisfies (PN) and (SZ) results into a similar block-sparse structure in the TT-cores of the TTO representation of $\mathbf{H}$ as the one described in the Corollary 3.3.1. This motivates the introduction of the following theorem. It considers a general Hamiltonian operator with p-body interactions where p is a non-negative integer. In particular, it can be applied to the 2-body operator defined in the spin-orbital basis or the spatial-orbital basis. The following result is inspired by the work in [2].

Theorem 4. Let $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$, $n \in \{2, 4\}$, be the symmetric matrix representation of the particle-number preserving Hamiltonian operator with at most p-body interactions, i.e satisfying (PN), (pB), (S). Let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be its TTO representation with TTO-ranks $(R_1, \ldots, R_d)$. Then,

(a) there is a block-sparse representation of the TT-cores $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times n \times n \times R_k}, k \in [d]$. This can be expressed formally as follows:

For fixed $\mu_k, \nu_k \in [n]$, $\mathbf{H}_k[\mu_k, \nu_k] := \mathcal{H}_k[:, \mu_k, \nu_k, :] \in \mathbb{R}^{R_{k-1} \times R_k}$ are block matrices. These blocks are denoted by:

$$\mathbf{H}_k[\mu_k, \nu_k]|_{\rho^{\mathbf{H}}_{k-1,j_{k-1}} \times \rho^{\mathbf{H}}_{k,j_k}} := \mathbf{H}_k^{(j_{k-1}, j_k)}[\mu_k, \nu_k] \in \mathbb{R}^{\rho^{\mathbf{H}}_{k-1,j_{k-1}} \times \rho^{\mathbf{H}}_{k,j_k}}, \tag{3.57}$$

where $j_0 = 0$, $j_d = 0$, $\rho^{\mathbf{H}}_{0,0} = \rho^{\mathbf{H}}_{d,0} = 1$ and for $k \in \{2, \ldots, d-1\}$, we have:

$$R_{k-1} = \sum_{j_{k-1} \in S^{\mathbf{H}}_{k-1}} \rho^{\mathbf{H}}_{k-1,j_{k-1}}, \; R_k = \sum_{j_k \in S^{\mathbf{H}}_k} \rho^{\mathbf{H}}_{k,j_k}, \tag{3.58}$$

where

$$S^{\mathbf{H}}_k = \{-\beta_k, \ldots, \beta_k\}, \text{with } \beta_k = \min\left\{\frac{n}{2}k, (d-k)\frac{n}{2}, 2p\right\}, \tag{3.59}$$

p denotes the number of p-body interactions considered and we have:

$$j_{k-1} + t_k = j_k, \tag{3.60}$$

where $t_k = q_n(\mu_k) - q_n(\nu_k)$, and q_n is defined in (PN). One has the following representation

- If $t_k = 0$,

$$\mathbf{H}_k[\mu_k, \nu_k] = \begin{pmatrix} \mathbf{H}_k^{(-\beta_{k-1}, -\beta_{k-1})}[\mu_k, \nu_k] & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \mathbf{H}_k^{(\beta_{k-1}, \beta_{k-1})}[\mu_k, \nu_k] \end{pmatrix}, \text{(block-diagonal matrix)}$$

- If $t_k \in \{-1, -2\}$

$$\mathbf{H}_k[\mu_k, \nu_k] = \begin{pmatrix} \mathbf{0} & & & & \\ \mathbf{H}_k^{(-\beta_{k-1}, -\beta_{k-1}+t_k)}[\mu_k, \nu_k] & \mathbf{0} & & & \\ & \ddots & \ddots & & \\ & & & \mathbf{H}_k^{(\beta_{k-1}, \beta_{k-1}+t_k)}[\mu_k, \nu_k] & \mathbf{0} \end{pmatrix},$$

- If $t_k \in \{1, 2\}$

$$\mathbf{H}_k[\mu_k, \nu_k] = \begin{pmatrix} \mathbf{0} & \mathbf{H}_k^{(-\beta_{k-1}, -\beta_{k-1}+t_k)}[\mu_k, \nu_k] & & \\ & \ddots & \ddots & \\ & & \mathbf{0} & \mathbf{H}_k^{(\beta_{k-1}, \beta_{k-1}+t_k)}[\mu_k, \nu_k] \\ & & & \mathbf{0} \end{pmatrix},$$

It follows that for $k \in [d]$, the TT-cores $\mathbf{H}_k[\mu_k, \nu_k]$ have at most $2\beta_k + 1$ nonzero blocks. Each element of $\mathbf{H}$ can be expressed in terms of these blocks as:

$$\mathbf{H}(\overline{\mu_1, \dots, \mu_d}; \overline{\nu_1, \dots, \nu_d}) = \sum_{t_1, \cdots, t_d = -\frac{n}{2}}^{\frac{n}{2}} \mathbf{H}_1^{(0,t_1)}[\mu_1, \nu_1] \cdots \mathbf{H}_k^{\left(\sum_{l=1}^{k-1} t_l, \sum_{l=1}^{k} t_l\right)}[\mu_k, y_k] \dots \mathbf{H}_d^{\left(\sum_{l=1}^{d-1} t_l, 0\right)}[\mu_d, \nu_d]. \tag{3.61}$$

(b) The nonzero blocks in the TT-cores can be related to one another through the existence of orthogonal matrices $\mathbf{Q}$ for each $j_k \in \{-\beta_k, \cdots, \beta_k\}$, such that

$$\mathbf{H}_k^{(j_k, j_k + t_k)}[\mu_k, \nu_k] = \mathbf{H}_k^{(-j_k, -j_k - t_k)}[\nu_k, \mu_k]\mathbf{Q}, \tag{3.62}$$

Remark 3.3.3. The results in Theorem 4 are also valid for complex hermitian matrices. As a consequence, if $\mathbf{H}$ is a particle-number preserving Hamiltonian with p-body interactions, all the matrices $f(\mathbf{H})$ in the C*-algebra generated by $\mathbf{H}$ via functional calculus for hermitian matrices share the same block representation.

In the following, we split the proof of Theorem 4 into three lemmas.

Outline The first Lemma establishes the block-sparse structure in the TT-cores of an Hamiltonian operator that satisfies (PN). The second lemma assesses the maximum number of nonzero blocks present in the TT-cores derived from an operator satisfying both (PN) and (pB). The third lemma demonstrates that the nonzero blocks within the TT-cores are interrelated through the incorporation of orthogonal matrices.

Lemma 3.3.2. *Consider the Hamiltonian operator matrix $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ with a TTO representation $(\mathcal{H}_1, \dots, \mathcal{H}_d)$ and TTO-ranks $(R_1, \dots, R_d)$. We have that the Hamiltonian and the particle number operator commute, i.e., $\mathbf{HN} = \mathbf{NH}$, then the TTO representation of $\mathbf{H}$ exhibits a block-sparse structure, as depicted in statement (a) of Theorem 4.*

Proof. We start by recalling the orthonormal basis $\{\boldsymbol{e}_{\mu_k}\}_{k \in [d], \mu_k \in [n]}$, as introduced in (1.119), such that vectors $\boldsymbol{e}_{\mu_1 \mu_2 \dots \mu_d} := \boldsymbol{e}_{\mu_1} \otimes_K \boldsymbol{e}_{\mu_2} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_d}$ is a basis of the Fock space $\boldsymbol{\mathcal{F}}_d$. $\boldsymbol{e}_{\mu_1 \mu_2 \dots \mu_d}$ are eigenvectors of $\mathbf{N}$. We have that $\mathbf{H}$ and $\mathbf{N}$ commute, $\mathbf{H}$ can be expanded in the set of the eigenvectors of $\mathbf{N}$. This allows us to write $\mathbf{H}$ as follows:

$$\mathbf{H} = \sum_{\substack{\mu_1, \cdots, \mu_d, \\ \nu_1, \cdots, \nu_d = 1}}^{n} \mathbf{H}(\overline{\mu_1, \dots, \mu_d}; \overline{\nu_1, \dots, \nu_d}) \boldsymbol{e}_{\mu_1 \mu_2 \dots \mu_d} \boldsymbol{e}^*_{\nu_1 \nu_2 \dots \nu_d}, \tag{3.63}$$

where $\mathbf{H}(\overline{\mu_1, \dots, \mu_d}; \overline{\nu_1, \dots, \nu_d})$ are the entries of $\mathbf{H}$. Let $\mathcal{H} \in \mathbb{R}^{n \times n \times \cdots \times n}$ be the $2d$-th tensor folding of $\mathbf{H}$. Let $\tilde{\mathcal{H}} \in \mathbb{R}^{n \times n \times \cdots \times n}$ be the permuted tensor that can be defined as follows: for $k \in [d]$, the mode-$(1 : 2k)$ matricization of $\tilde{\mathcal{H}}$ denoted by $\tilde{\mathbf{H}}^{<2k>} \in \mathbb{R}^{n^{2k} \times n^{2d-2k}}$ is given by:

$$\tilde{\mathbf{H}}^{<2k>} = \sum_{\substack{\mu_1, \cdots, \mu_d, \\ \nu_1, \cdots, \nu_d = 1}}^{d} \mathbf{H}(\overline{\mu_1, \nu_1, \dots, \mu_k, \nu_k}; \overline{\mu_{k+1}, \nu_{k+1}, \cdots, \mu_d, \nu_d}) \boldsymbol{e}_{\mu_1 \nu_1 \dots \mu_k \nu_k} \otimes_K \boldsymbol{e}^*_{\mu_{k+1} \nu_{k+1} \dots \mu_d \nu_d}.$$

We define the following functions:

$$\begin{aligned} \mathcal{Q}_{\leq k} : \{1, \dots, n\}^k \to \mathbb{N}_0 \quad \mathcal{Q}_{>k} : \{1, \dots, n\}^{d-k} \to \mathbb{N}_0 \\ \mu_{\leq k} \to \sum_{l=1}^{k} q_n(\mu_l), \qquad \mu_{>k} \to \sum_{l=k+1}^{d} q_n(\mu_l), \end{aligned} \tag{3.64}$$

with $\mu_{\leq k} = (\mu_1, \dots, \mu_k)$ and $\mu_{>k} = (\mu_{k+1}, \dots, \mu_d)$.

For $k \in [d]$, by enforcing the particle number conservation, the following condition must be satisfied (see (PN)), we have:

$$\sum_{k=1}^{d} t_k = 0, \tag{3.65}$$

where $t_k = q_n(\mu_k) - q_n(\nu_k),\ k \in [d]$. The range of values for t_k depends on the basis. In the case where $n = 2$, $t_k \in \{-1, 0, 1\}$, otherwise, for $n = 4$, $t_k \in \{-2, -1, 0, 1, 2\}$, thus, for $n \in \{2, 4\}$, $t_k \in \left\{-\frac{n}{2}, \dots, \frac{n}{2}\right\}$.

Hence, $\tilde{\mathbf{H}}^{<2k>}$ has the following expression:

$$\tilde{\mathbf{H}}^{<2k>} = \sum_{\substack{\mu_{\leq k},\mu_{>k} \\ \nu_{\leq k},\nu_{>k}}} \sum_{t_1,\cdots,t_k} \mathbf{H}(\overline{\mu_{\leq k},\nu_{\leq k}};\overline{\mu_{>k},\nu_{>k}}) \left[\delta^{\mathcal{Q}_{\leq k}(\mu_{\leq k})-\sum_{l=1}^{k} t_l}_{\mathcal{Q}_{\leq k}(\nu_{\leq k})} \boldsymbol{e}_{\mu_{\leq k}\nu_{\leq k}}\right] \left[\delta^{\mathcal{Q}_{>k}(\mu_{>k})}_{\mathcal{Q}_{>k}(\nu_{>k})-\sum_{l=k+1}^{d} t_l} \boldsymbol{e}_{\mu_{>k}\nu_{>k}}\right]^*, \tag{3.66}$$

where for simplicity in notation, we modified the Kronecker delta notation $\delta_{i,j} := \delta_i^j$. $\mathbf{H}(\overline{\mu_{\leq k},\nu_{\leq k}};\overline{\mu_{>k},\nu_{>k}}) := \mathbf{H}(\overline{\mu_1,\nu_1,\ldots,\mu_k,\nu_k};\overline{\mu_{k+1},\nu_{k+1},\cdots,\mu_d,\nu_d})$, $\boldsymbol{e}_{\mu_{\leq k}\nu_{\leq k}} := \boldsymbol{e}_{\mu_1\nu_1\ldots\mu_k\nu_k}$, and $\boldsymbol{e}_{\mu_{>k}\nu_{>k}} := \boldsymbol{e}_{\mu_{k+1}\nu_{k+1}\ldots\mu_d\nu_d}$.

To obtain the TTO decomposition of this operator, we use the classical TT-SVD procedure. For k=1, $\tilde{\mathbf{H}}^{<2>} \in \mathbb{R}^{n^2\times n^{2d-2}}$ can be defined entry-wise as follows:

$$\tilde{\mathbf{H}}^{<2>} = \sum_{t_1=-\frac{n}{2}}^{\frac{n}{2}} (\mathbf{H})^{\mu_{>1},\nu_{>1}}_{\mu_1,\nu_1} \left[\delta^{\mathcal{Q}_1(\mu_1)-t_1}_{\mathcal{Q}_1(\nu_1)} \boldsymbol{e}_{\mu_1\nu_1}\right] \left[\delta^{\mathcal{Q}_{>1}(\mu_{>1})}_{\mathcal{Q}_{>1}(\nu_{>1})-t_1} \boldsymbol{e}_{\mu_{>1}\nu_{>1}}\right]. \tag{3.67}$$

It is noted that $\tilde{\mathbf{H}}^{<2>}$ has a block form. Let us begin by defining the following sets for $k \in [d]$, and $t_k \in \left\{-\frac{n}{2},\ldots,\frac{n}{2}\right\}$:

$$S_{k,t_k} = \left\{\mu_k,\nu_k \in [n], q_n(\mu_k) - q_n(\nu_k) = t_k\right\}, \tag{3.68}$$

$$S_{>k,t} = \left\{\mu_l,\nu_l \in [n], \sum_{l=k+1}^{d} q_n(\mu_l) - q_n(\nu_l) = t\right\}, \tag{3.69}$$

$$S_{\leq k,t} = \left\{\mu_l,\nu_l \in [n], \sum_{l=1}^{k} q_n(\mu_l) - q_n(\nu_l) = t\right\}, \tag{3.70}$$

with cardinality $\#S_{k,t_k} \in \left[n^{2k}\right]$, and $\#S_{>k,t} \in \left[n^{2d-2k}\right]$, such that the matrix representation of $\tilde{\mathbf{H}}^{<2>}$ can be represented by the following anti-diagonal block matrix:

$$\tilde{\mathbf{H}}^{<2>} = \begin{array}{c} \\ \#S_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{1,0} \\ \vdots \\ \#S_{1,\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{ccccc} \#S_{>1,-\frac{n}{2}} & \cdots & \#S_{>1,0} & \cdots & \#S_{>1,\frac{n}{2}} \end{array} \\ \begin{pmatrix} & & & & \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,-\frac{n}{2}}\times\#S_{>1,\frac{n}{2}}} \\ & & & \iddots & \\ & & \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,0}\times\#S_{>1,0}} & & \\ & \iddots & & & \\ \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,\frac{n}{2}}\times\#S_{>1,-\frac{n}{2}}} & & & & \end{pmatrix} \end{array}. \tag{3.71}$$

The matrix $\tilde{\mathbf{H}}^{<2>}$ can be also expressed as the product between a diagonal matrix and an anti-diagonal matrix as follows:

$$\tilde{\mathbf{H}}^{<2>} = \mathbf{D}_1\mathbf{P}_1, \tag{3.72}$$

where

$$\mathbf{D}_1 = \begin{array}{c} \\ \#S_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{1,0} \\ \vdots \\ \#S_{1,\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{ccccc} \#S_{>1,\frac{n}{2}} & \cdots & \#S_{>1,0} & \cdots & \#S_{>1,-\frac{n}{2}} \end{array} \\ \begin{pmatrix} \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,-\frac{n}{2}}\times\#S_{>1,\frac{n}{2}}} & & & & \\ & \ddots & & & \\ & & \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,0}\times\#S_{>1,0}} & & \\ & & & \ddots & \\ & & & & \tilde{\mathbf{H}}^{<2>}|_{\#S_{1,\frac{n}{2}}\times\#S_{>1,-\frac{n}{2}}} \end{pmatrix} \end{array}, \tag{3.73}$$

and

$$\mathbf{P}_1 = \begin{array}{c} \\ \#S_{>1,\frac{n}{2}} \\ \vdots \\ \#S_{>1,0} \\ \vdots \\ \#S_{>1,-\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{cccccc} \#S_{>1,-\frac{n}{2}} & \cdots & \#S_{>1,0} & \cdots & |S_{>1,\frac{n}{2}} \end{array} \\ \begin{pmatrix} & & & \mathbf{I}_{\#S_{>1,\frac{n}{2}} \times \#S_{>1,\frac{n}{2}}} \\ & & \iddots & \\ & \mathbf{I}_{\#S_{>1,0} \times \#S_{>1,0}} & & \\ \iddots & & & \\ \mathbf{I}_{\#S_{>1,-\frac{n}{2}} \times \#S_{>1,-\frac{n}{2}}} & & & \end{pmatrix} \end{array}, \tag{3.73}$$

Suppose that $\mathbf{D}_1$ is of rank R_1. We apply a block-wise SVD on $\mathbf{D}_1$, such that

$$\mathbf{D}_1 = \mathbf{U}_1(\mathbf{S}_1\mathbf{V}_1^*), \tag{3.74}$$

where $\mathbf{U}_1 \in \mathbb{R}^{n^2 \times R_1}$ and $\mathbf{S}_1\mathbf{V}_1^* \in \mathbb{R}^{R_1 \times n^{2d-2}}$ such that

$$\mathbf{U}_1 = \begin{array}{c} \\ \#S_{1,-\frac{n}{2}}\rho^{\mathbf{H}}_{0,0} \\ \vdots \\ \#S_{1,0}\rho^{\mathbf{H}}_{0,0} \\ \vdots \\ \#S_{1,\frac{n}{2}}\rho^{\mathbf{H}}_{0,0} \end{array} \begin{array}{c} \begin{array}{cccccc} \rho^{\mathbf{H}}_{1,-\frac{n}{2}} & \cdots & \rho^{\mathbf{H}}_{1,0} & \cdots & \rho^{\mathbf{H}}_{1,\frac{n}{2}} \end{array} \\ \begin{pmatrix} \mathbf{U}_1|_{\#S_{1,-\frac{n}{2}}\rho^{\mathbf{H}}_{0,0} \times \rho^{\mathbf{H}}_{1,-\frac{n}{2}}} & & & & \\ & \ddots & & & \\ & & \mathbf{U}_1|_{\#S_{1,0}\rho^{\mathbf{H}}_{0,0} \times \rho^{\mathbf{H}}_{1,0}} & & \\ & & & \ddots & \\ & & & & \mathbf{U}_1|_{\#S_{1,\frac{n}{2}}\rho^{\mathbf{H}}_{0,0} \times \rho^{\mathbf{H}}_{1,\frac{n}{2}}} \end{pmatrix} \end{array},$$

and

$$(\mathbf{S}_1\mathbf{V}_1^*) = \begin{array}{c} \\ \rho^{\mathbf{H}}_{1,-\frac{n}{2}} \\ \vdots \\ \rho^{\mathbf{H}}_{1,0} \\ \vdots \\ \rho^{\mathbf{H}}_{1,\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{cccccc} \#S_{>1,\frac{n}{2}}\rho^{\mathbf{H}}_{d,0} & \cdots & \#S_{>1,0}\rho^{\mathbf{H}}_{d,0} & \cdots & \#S_{>1,-\frac{n}{2}}\rho^{\mathbf{H}}_{d,0} \end{array} \\ \begin{pmatrix} (\mathbf{S}_1\mathbf{V}_1^*)|_{\rho^{\mathbf{H}}_{1,-\frac{n}{2}} \times \#S_{>1,\frac{n}{2}}\rho^{\mathbf{H}}_{d,0}} & & & & \\ & \ddots & & & \\ & & (\mathbf{S}_1\mathbf{V}_1^*)|_{\rho^{\mathbf{H}}_{1,0} \times \#S_{>1,0}\rho^{\mathbf{H}}_{d,0}} & & \\ & & & \ddots & \\ & & & & (\mathbf{S}_1\mathbf{V}_1^*)|_{\rho^{\mathbf{H}}_{1,\frac{n}{2}} \times \#S_{>1,-\frac{n}{2}}\rho^{\mathbf{H}}_{d,0}} \end{pmatrix} \end{array},$$

where $\rho^{\mathbf{H}}_{1,t_1}$ are the sizes of the ranks of the nonzero blocks for $t_1 \in \left\{-\frac{n}{2}, \cdots, \frac{n}{2}\right\}$ such that $\sum_{t_1} \rho^{\mathbf{H}}_{1,t_1} = R_1, \rho^{\mathbf{H}}_{0,0} = \rho^{\mathbf{H}}_{d,0} = 1$. For $t_1 \in \left\{-\frac{n}{2}, \cdots, \frac{n}{2}\right\}$, $\mu_1, \nu_1 \in S_{1,t_1}$, and $\mu_{>1}, \nu_{>1} \in S_{>1,t_1}$. We note that

$$\tilde{\mathbf{H}}^{<2>}(\overline{\mu_1, \nu_1}; \overline{\mu_{>1}, \nu_{>1}}) = \sum_{t_1=-\frac{n}{2}}^{\frac{n}{2}} \sum_{\gamma_1=1}^{R_1} \mathbf{U}_1(\overline{\mu_1, \nu_1}; \gamma_1) \left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)(\gamma_1; \overline{\mu_{>1}, \nu_{>1}}) \delta^{\mathcal{Q}_1(\mu_1)-t_1}_{\mathcal{Q}_1(\nu_1)} \delta^{\mathcal{Q}_{>1}(\mu_{>1})}_{\mathcal{Q}_{>1}(\nu_{>1})-t_1}, \tag{3.75}$$

it follows that:

$$\left(\mathbf{U}_1|_{\rho^{\mathbf{H}}_{0,0}\#S_{1,t_1} \times \rho^{\mathbf{H}}_{1,t_1}}\right)(\overline{\alpha_0, \sigma_1}; \alpha_1) = \mathbf{U}_1(\overline{\mu_1, \nu_1}; \gamma_1) \delta^{\mathcal{Q}_1(\mu_1)-t_1}_{\mathcal{Q}_1(\nu_1)}, \tag{3.76}$$

and

$$\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)|_{\rho^{\mathbf{H}}_{1,t_1} \times \#S_{>1,t_1}\rho^{\mathbf{H}}_{d,0}}\Big)(\alpha_1; \overline{\sigma_{d-1}, \alpha_d}) = \left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)(\gamma_1; \overline{\mu_{>1}, \nu_{>1}}) \delta^{\mathcal{Q}_{>1}(\mu_{>1})-t_1}_{\mathcal{Q}_{>1}(\nu_{>1})}, \tag{3.77}$$

where we select only the indices $\gamma_1 \in \left\{\sum_{\ell=-\frac{n}{2}}^{t_1-1} \rho_{1,\ell}^{\mathbf{H}} + 1, \cdots, \sum_{\ell=-\frac{n}{2}}^{t_1} \rho_{1,t_1}^{\mathbf{H}}\right\}$, $\alpha_0 = \alpha_d = 1$, $\alpha_1 \in \left[\rho_{1,t_1}^{\mathbf{H}}\right]$, $\sigma_1 \in [\#S_{1,t_1}]$, $\sigma_{d-1} \in [\#S_{>1,t_1}]$.

For simplicity, we introduce the following notation:

$$\mathbf{U}_1^{(0,t_1)}[\mu_1,\nu_1](\alpha_0;\alpha_1) = \mathbf{U}_1(\overline{\mu_1,\nu_1};\gamma_1)\delta_{\mathcal{Q}_1(\nu_1)}^{\mathcal{Q}_1(\mu_1)-t_1}, t_1 \in \left\{-\frac{n}{2},\cdots,\frac{n}{2}\right\}, \tag{3.78}$$

where $\mathbf{U}_1^{(0,t_1)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,t_1}^{\mathbf{H}}}$. Similar notations are applied for the blocks of $(\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1$ such that

$$\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_{>1},\nu_{>1}](\alpha_1;\alpha_d) = \left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)(\gamma_1;\overline{\mu_{>1},\nu_{>1}})\delta_{\mathcal{Q}_{>1}(\nu_{>1})}^{\mathcal{Q}_{>1}(\mu_{>1})-t_1}, \tag{3.79}$$

where $\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_{>1},\nu_{>1}] = \left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_2,\nu_2,\cdots,\mu_d,\nu_d] \in \mathbb{R}^{\rho_{1,t_1}^{\mathbf{H}} \times \rho_{d,0}^{\mathbf{H}}}$. Since $\tilde{\mathbf{H}}^{<2>} = \mathbf{U}_1(\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1$, it can be defined entry-wise as follows, using the new notations in (3.78) and (3.79):

$$\begin{aligned}\tilde{\mathbf{H}}^{<2>}(\overline{\mu_1,\nu_1};\overline{\mu_{>1},\nu_{>1}}) &= \sum_{t_1=-\frac{n}{2}}^{\frac{n}{2}} \mathbf{U}_1^{(0,t_1)}[\mu_1,\nu_1]\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_{>1},\nu_{>1}] \\ &= \sum_{t_1=-\frac{n}{2}}^{\frac{n}{2}} \mathbf{H}_1^{(0,t_1)}[\mu_1,\nu_1]\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_{>1},\nu_{>1}].\end{aligned} \tag{3.80}$$

Here, we define:

$$\mathbf{H}_1^{(0,t_1)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,t_1)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,t_1}^{\mathbf{H}}}, t_1 \in \left\{-\frac{n}{2},\cdots,\frac{n}{2}\right\}. \tag{3.81}$$

It follows that for $k=1, n=4$, $\mu_1,\nu_1 \in \{1,2,3,4\}$, the first TT-core denoted by $\mathbf{H}_1$ has the following block-sparse structure:

$$\mathbf{H}_1[\mu_1,\nu_1] = \begin{cases} \mathbf{H}_1^{(0,-2)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,-2)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,-2}^{\mathbf{H}}}, \text{ if } \mu_1,\nu_1 \in S_{1,-2}, \\ \mathbf{H}_1^{(0,-1)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,-1)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,-1}^{\mathbf{H}}}, \text{ if } \mu_1,\nu_1 \in S_{1,-1}, \\ \mathbf{H}_1^{(0,0)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,0)}[\mu_1,\nu_1]\mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,0}^{\mathbf{H}}}, \text{ if } \mu_1,\nu_1 \in S_{1,0}, \\ \mathbf{H}_1^{(0,1)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,1)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,1}^{\mathbf{H}}}, \text{ if } \mu_1,\nu_1 \in S_{1,1}, \\ \mathbf{H}_1^{(0,2)}[\mu_1,\nu_1] = \mathbf{U}_1^{(0,2)}[\mu_1,\nu_1] \in \mathbb{R}^{\rho_{0,0}^{\mathbf{H}} \times \rho_{1,2}^{\mathbf{H}}}, \text{ if } \mu_1,\nu_1 \in S_{1,2}. \end{cases} \tag{3.82}$$

In what follows, we aim at proving that this property is preserved for the remaining TT-cores by construction. It can be verified that

$$\delta_{\mathcal{Q}_{>1}(\nu_{>1})-t_1}^{\mathcal{Q}_{>1}(\mu_{>1})} = \sum_{t_2=-\frac{n}{2}}^{\frac{n}{2}} \delta_{q_n(\nu_2)}^{q_n(\mu_2)-t_2}\delta_{\mathcal{Q}_{>2}(\nu_{>2})-(t_1+t_2)}^{\mathcal{Q}_{>2}(\mu_{>2})}, \tag{3.83}$$

then by replacing (3.83) in (3.77), one obtains:

$$\begin{aligned}\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)^{(t_1,0)}[\mu_{>1},\nu_{>1}](\alpha_1;\alpha_d) &= \sum_{t_2=-\frac{n}{2}}^{\frac{n}{2}} \delta_{q_n(\nu_2)}^{q_n(\mu_2)-t_2}\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)(\gamma_1;\overline{\mu_{>1},\nu_{>1},\alpha_d})\delta_{\mathcal{Q}_{>2}(\nu_{>2})-(t_1+t_2)}^{\mathcal{Q}_{>2}(\mu_{>2})} \\ &= \sum_{t_2=-\frac{n}{2}}^{\frac{n}{2}} \delta_{q_n(\nu_2)}^{q_n(\mu_2)-t_2}\left((\mathbf{S}_1\mathbf{V}_1^*)\mathbf{P}_1\right)(\overline{\gamma_1,\mu_2,\nu_2};\overline{\mu_{>2},\nu_{>2},\alpha_d})\delta_{\mathcal{Q}_{>2}(\nu_{>2})-(t_1+t_2)}^{\mathcal{Q}_{>2}(\mu_{>2})}.\end{aligned} \tag{3.84}$$

Now, suppose that (3.84) are the entries of the following tensor $\mathcal{W} \in \mathbb{R}^{\left(\sum_{t_1=\frac{-n}{2}}^{\frac{n}{2}} \rho_{1,t_1}^{\mathbf{H}}\right) \times \underbrace{n \times \ldots \times n}_{2d-2} \times \rho_{d,0}^{\mathbf{H}}}$.

It is noted that the mode-(1:3) matricization of $\mathcal{W}$ denoted by $\mathbf{W}^{<3>} \in \mathbb{R}^{(n^2\sum_{t_1=\frac{-n}{2}}^{\frac{n}{2}} \rho_{1,t_1}^{\mathbf{H}}) \times (n^{2d-4}\rho_{d,0}^{\mathbf{H}})}$ is a block-sparse matrix and can be represented as follows:

$$\mathbf{W}^{<3>} = \begin{array}{c} \\ \#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}} \\ \vdots \\ \#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{ccccc} \#S_{>2,-n} & \cdots & \#S_{>2,0} & \cdots & \#S_{>2,n} \end{array} \\ \begin{pmatrix} & & & & \mathbf{W}^{<3>}|_{\#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}}\times\#S_{>2,n}} \\ & & & \iddots & \\ & & \mathbf{W}^{<3>}|_{\#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}}\times\#S_{>2,0}} & & \\ & & \vdots & & \\ & & \mathbf{W}^{<3>}|_{\#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}}\times\#S_{>2,0}} & & \\ & \iddots & & & \\ \mathbf{W}^{<3>}|_{\#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}}\times\#S_{>2,-n}} & & & & \end{pmatrix} \end{array}.$$

In a similar way, this matrix can be expressed as the product of a diagonal matrix and an anti-diagonal matrix such that

$$\mathbf{W}^{<3>} = \mathbf{D}_2\mathbf{P}_2 \in \mathbb{R}^{(n^2\sum_{t_1=\frac{-n}{2}}^{\frac{n}{2}}\rho^{\mathbf{H}}_{1,t_1})\times(n^{2d-4}\rho^{\mathbf{H}}_{d,0})}. \tag{3.85}$$

Here, $\mathbf{D}_2$ (resp. $\mathbf{P}_2$) has a similar structure as $\mathbf{D}_1$ (resp. $\mathbf{P}_1$). As previously demonstrated, a block-wise SVD is applied on $\mathbf{D}_2$, supposed to be of rank R_2, such decomposition writes:

$$\mathbf{D}_2 = \mathbf{U}_2(\mathbf{S}_2\mathbf{V}_2^*). \tag{3.86}$$

This decomposition has the following matrix representation:

$$\mathbf{U}_2 = \begin{array}{c} \\ \#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}}\cdots\#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}} \\ \vdots \\ \#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}} \end{array} \begin{array}{c} \begin{array}{ccccc} \rho^{\mathbf{H}}_{2,-n} & \cdots & \rho^{\mathbf{H}}_{2,0} & \cdots & \rho^{\mathbf{H}}_{2,n} \end{array} \\ \begin{pmatrix} \mathbf{U}_2|_{\#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}}\times\rho^{\mathbf{H}}_{2,-n}} & & & & \\ & \ddots & & & \\ & & \mathbf{U}_2|_{(\#S_{2,-\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}}+\cdots+\#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,-\frac{n}{2}})\times\rho^{\mathbf{H}}_{2,0}} & & \\ & & & \ddots & \\ & & & & \mathbf{U}_2|_{\#S_{2,\frac{n}{2}}\rho^{\mathbf{H}}_{1,\frac{n}{2}}\times\rho_{2,n}} \end{pmatrix} \end{array},$$

and

$$(\mathbf{S}_2\mathbf{V}_2^*) = \begin{array}{c} \\ \rho^{\mathbf{H}}_{2,-n} \\ \vdots \\ \rho^{\mathbf{H}}_{2,0} \\ \vdots \\ \rho^{\mathbf{H}}_{2,n} \end{array} \begin{array}{c} \begin{array}{ccccc} \#S_{>2,n}\rho^{\mathbf{H}}_{d,0} & \cdots & \#S_{>2,0}\rho^{\mathbf{H}}_{d,0} & \cdots & \#S_{>2,-n}\rho^{\mathbf{H}}_{d,0} \end{array} \\ \begin{pmatrix} (\mathbf{S}_2\mathbf{V}_2^*)|_{\rho_{2,-n}}\times\#S_{>2,n}\rho^{\mathbf{H}}_{d,0} & & & & \\ & \ddots & & & \\ & & (\mathbf{S}_2\mathbf{V}_2^*)|_{\rho^{\mathbf{H}}_{2,0}\times\#S_{>2,0}\rho^{\mathbf{H}}_{d,0}} & & \\ & & & \ddots & \\ & & & & (\mathbf{S}_2\mathbf{V}_2^*)|_{\rho^{\mathbf{H}}_{2,n}\times S_{d-2,-n}\rho^{\mathbf{H}}_{d,0}} \end{pmatrix} \end{array},$$

where $\sum_{t_1,t_2=-\frac{n}{2}}^{\frac{n}{2}}\rho^{\mathbf{H}}_{2,t_1+t_2} = R_2$. As a result, $\mathbf{W}^{<3>}$ can be defined entry-wise as follows:

$$\mathbf{W}^{<3>}(\overline{\alpha_1,\mu_2,\nu_2};\overline{\mu_{>2},\nu_{>2},\alpha_d}) = \mathbf{H}_2^{(t_1,t_1+t_2)}[\mu_2,\nu_2]\left((\mathbf{S}_2\mathbf{V}_2^*)\mathbf{P}_2\right)^{(t_1+t_2,0)}[\mu_{>2},\nu_{>2}], \tag{3.87}$$

where

$$\mathbf{H}_2^{(t_1,t_1+t_2)}[\mu_2,\nu_2](\alpha_1;\alpha_2) = \mathbf{U}_2(\overline{\alpha_1,\mu_2,\nu_2};\gamma_2)\delta^{\mathcal{Q}_2(\mu_2)-t_2}_{\mathcal{Q}_2(\nu_2)}, \tag{3.88}$$

and

$$\left((\mathbf{S}_2\mathbf{V}_2^*)\mathbf{P}_2\right)^{(t_1+t_2,0)}[\mu_{>2},\nu_{>2}](\alpha_2;\alpha_d) = \left((\mathbf{S}_2\mathbf{V}_2^*)\mathbf{P}_2\right)(\gamma_2;\overline{\mu_{>2},\nu_{>2},\alpha_d})\delta^{\mathcal{Q}_{>2}(\mu_{>2})}_{\mathcal{Q}_{>2}(\nu_{>2})-(t_1+t_2)}, \tag{3.89}$$

with $\alpha_1 \in \left[\rho^{\mathbf{H}}_{1,t_1}\right]$, $\alpha_2 \in \left[\rho^{\mathbf{H}}_{2,t_1+t_2}\right]$, $\alpha_d = 1$ and $\gamma_2 \in \left\{\sum_{\ell=-\frac{n}{2}}^{t_2-1} \rho^{\mathbf{H}}_{2,t_1+\ell} + 1, \cdots, \sum_{\ell=-\frac{n}{2}}^{t_2} \rho^{\mathbf{H}}_{2,t_1+\ell}\right\}$. For $k = 2$, the entries of the mode-$(1:4)$ matricization of $\tilde{\mathbf{H}}$, denoted by $\tilde{\mathbf{H}}^{<4>}$, are given by:

$$\tilde{\mathbf{H}}^{<4>}(\overline{\mu_1,\nu_1,\mu_2,\nu_2};\overline{\mu_{>2},\nu_{>2}}) = \sum_{t_1,t_2=-\frac{n}{2}}^{\frac{n}{2}} \mathbf{H}_1^{(0,t_1)}[\mu_1,\nu_1]\mathbf{H}_2^{(t_1,t_1+t_2)}[\mu_2,\nu_2]\left((\mathbf{S}_2\mathbf{V}_2^*)\mathbf{P}_2\right)^{(t_1+t_2,0)}[\mu_{>2},\nu_{>2}]. \tag{3.90}$$

Here, the second TT-core, denoted by $\mathcal{H}_2 \in \mathbb{R}^{R_1\times n\times n\times R_2}$, can be derived such that $\mathbf{H}_2[\mu_2,\nu_2] = \mathcal{H}_2[:,\mu_2,\nu_2,:] \in \mathbb{R}^{R_1\times R_2}$ are block-sparse matrices with at most $kn+1 = 2n+1$ nonzero blocks $\mathbf{H}_2^{(t_1,t_1+t_2)}[\mu_2,\nu_2] \in \mathbb{R}^{\rho^{\mathbf{H}}_{1,t_1}\times\rho^{\mathbf{H}}_{2,t_1+t_2}}$. By iteration, until $k = d$, the elements of the mode-$(1:2d)$ matricization of $\tilde{\mathcal{H}}$, denoted by $\tilde{\mathbf{H}}^{<2d>}$, are given by:

$$\tilde{\mathbf{H}}^{<2d>}(\overline{\mu_1,\nu_1,\cdots,\mu_d,\nu_d}) = \sum_{t_1,\cdots,t_d=-\frac{n}{2}}^{\frac{n}{2}} \mathbf{H}_1^{(0,t_1)}[\mu_1,\nu_1]\cdots\mathbf{H}_k^{\left(\sum_{l=1}^{k-1}t_l,\sum_{l=1}^{k}t_l\right)}[\mu_k,y_k]\ldots\mathbf{H}_d^{\left(\sum_{l=1}^{d-1}t_l,\sum_{l=1}^{d}t_l\right)}[\mu_d,\nu_d], \tag{3.91}$$

where $\sum_{l=1}^{d} t_l = 0$ and for $k \in [d]$, the matrices $\mathbf{H}_k^{\left(\sum_{l=1}^{k-1}t_l,\sum_{l=1}^{k}t_l\right)}[\mu_k,\nu_k] \in \mathbb{R}^{\rho^{\mathbf{H}}_{k-1,\sum_{l=1}^{k-1}t_l},\rho^{\mathbf{H}}_{k,\sum_{l=1}^{k}t_l}}$ are the blocks of the TT-cores $\mathbf{H}_k[\mu_k,\nu_k]$.

$\mathbf{H}_k[\mu_k,\nu_k]$ are block-sparse matrices with at most $kn+1$ nonzero blocks on sets $\left\{\rho^{\mathbf{H}}_{k-1,\sum_{l=1}^{k-1}t_l} \times \rho^{\mathbf{H}}_{k,\sum_{l=1}^{k}t_l}\right\}_{k\in[d]}$. A similar reasoning can be employed when using TT-SVD, starting from right to left. This yields block-sparse TT-cores $\mathbf{H}_k[\mu_k,\nu_k]$ with a maximum of $(d-k)n+1$ nonzero blocks on sets $\left\{\rho^{\mathbf{H}}_{d-k,\sum_{l=1}^{d-k}t_l} \times \rho^{\mathbf{H}}_{d-k,\sum_{l=1}^{d-k}t_l}\right\}_{k\in[d]}$. Therefore, we say that $\mathbf{H}_k[\mu_k,\nu_k]$ are block-sparse matrices with at most $\min\{kn+1,(d-k)n+1\}$ blocks. □

We now introduce the following lemma, which complements the previous lemma

Lemma 3.3.3. *Let $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$ be a Hamiltonian operator in TTO-format, $n \in \{2,4\}$ and let p be a non-negative integer. Assume that the operator has at most p-body interactions and that it commutes withe the particle number operator. Then, for each $k \in [d]$ and for $\mu_k,\nu_k \in [n]$, the TT-cores $\mathbf{H}_k[\mu_k,\nu_k]$ have at most $\min\{kn+1,(d-k)n+1,2p+1\}$ nonzero blocks.*

Proof. Let $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$ be a Hamiltonian operator with TTO decomposition, and let p be a non-negative integer. Assume that $\mathbf{H}$ is derived from an operator that exhibits at most p-body interactions and that preserves the particle number, it satisfies (PN) and (pB). Then for $\mu_k,\nu_k \in [n]$ and $k \in [d]$:

$$\mathbf{H}(\overline{\mu_1,\mu_2,\cdots,\mu_d};\overline{\nu_1,\nu_2,\cdots,\nu_d}) \neq 0 \Leftrightarrow \begin{cases} \sum_{k=1}^{d}|t_k| \leq 2p, \\ \sum_{k=1}^{d} t_k = 0, \end{cases} \tag{3.92}$$

with $t_k = q_n(\mu_k) - q_n(\nu_k), k \in [d]$. According to Lemma 3.3.2, the matrices $\mathbf{H}_k[\mu_k,\nu_k]$ have nonzero blocks on sets $\left\{\rho^{\mathbf{H}}_{k-1,\sum_{l=1}^{k-1}t_l} \times \rho^{\mathbf{H}}_{k,\sum_{l=1}^{k}t_l}\right\}_{k\in[d]}$. Given (3.92), and given the following set $(t_1,t_2,\ldots,t_{d-1},t_d)$, there are at most $2p$ terms with $t_k = \pm 1$ for $k \in [d]$. Therefore, $\max\limits_{1\leq k\leq d}\left\{\sum\limits_{l=1}^{k} t_l\right\} = \min\left\{\frac{n}{2}(d-k),\frac{n}{2}k,p\right\}$ and $\min\limits_{1\leq k\leq d}\left\{\sum\limits_{l=1}^{k} t_l\right\} = \max\left\{-\frac{n}{2}(d-k),-\frac{n}{2}k,-p\right\}$. Overall, the maximum number of nonzero blocks in the TT-cores of the TTO representation of $\mathbf{H}$ is reduced from $dn+1$ to $\min\{kn+1,(d-k)n+1,2p+1\}$ for $k \in [d]$. □

The nonzero blocks within the TT-cores can be interrelated by introducing orthogonal matrices, denoted as $\mathbf{Q}$. This is outlined in the following lemma:

Lemma 3.3.4. *Let $\mathbf{H} \in \mathbb{R}^{n^d\times n^d}$ be the symmetric matrix representation of the Hamiltonian operator with TTO decomposition, satisfying equations* (PN), (S), *and* (pB). *For*

$\beta_k = \min\{kn+1, (d-k)n+1, 2p+1\}$ *and for each* $j_k = \sum_{l=1}^{k} t_l \in \{-\beta_k, \ldots, \beta_k\}$*, where* $k \in [d]$*, there exist an orthogonal matrix* $\mathbf{Q}$ *such that*

$$\mathbf{H}_k^{(j_k, j_k+t_k)}[\mu_k, \nu_k] = \mathbf{H}_k^{(-j_k, -j_k-t_k)}[\nu_k, \mu_k]\mathbf{Q}, \tag{3.93}$$

with $t_k = q_n(\mu_k) - q_n(\nu_k)$.

Proof. Let $\mathbf{H} \in \mathbb{R}^{n^d \times n^d}$ be a symmetric Hamiltonian operator matrix satisfying the equations (PN), (S), and (pB). Let $\tilde{\mathcal{H}} \in \mathbb{R}^{n \times \cdots \times n}$ be the $2d$-th folding tensor of the permuted matrix $\mathbf{H}$ according its indices with entries $\tilde{\mathcal{H}}(\mu_1; \nu_1; \ldots; \mu_d; \nu_d)$. As described in Lemma 3.3.2, the TT-cores of the TTO decomposition denoted by $\mathbf{H}_k[\mu_k, \nu_k] \in \mathbb{R}^{R_{k-1} \times R_k}$, for fixed values of $\mu_k, \nu_k \in [n]$ and $k \in [d]$ are block-sparse. Additionally, let $\tilde{\mathbf{H}}^{<2k>}$ be the mode-$(1:2k)$ matricization of $\tilde{\mathcal{H}}$. It can be observed that for a fixed value of k, the matrix $\tilde{\mathbf{H}}^{<2k>}$ can be represented in the form of a block-sparse matrix. The matrix can be written as follows:

$$\tilde{\mathbf{H}}^{<2k>} = \begin{array}{c} \\ \#S_{\leq k,-\beta_k} \\ \vdots \\ \#S_{\leq k,0} \\ \vdots \\ \#S_{\leq k,\beta_k} \end{array} \begin{array}{c} \begin{array}{cccccc} \#S_{>k,-\beta_k} & \cdots & \#S_{>k,0} & \cdots & \#S_{>k,\beta_k} \end{array} \\ \begin{pmatrix} & & & & \mathbf{B}^{(-\beta_k,\beta_k)} \\ & & & \iddots & \\ & & \mathbf{B}^{(0,0)} & & \\ & \iddots & & & \\ \mathbf{B}^{(\beta_k,-\beta_k)} & & & & \end{pmatrix} \end{array}, \tag{3.94}$$

where $S_{\leq k,j_k}$ and $S_{>k,j_k}$ are defined in (3.69) and $\mathbf{B}^{(j_k,-j_k)}$ are nonzero blocks, for $j_k \in \{-\beta_k, \cdots, \beta_k\}$. Since $\mathbf{H}$ is symmetric, the following relation holds:

$$\tilde{\mathbf{H}}^{<2k>}(\overline{\mu_1, \nu_1, \cdots, \mu_k, \nu_k}; \overline{\mu_{k+1}, \nu_{k+1}, \cdots, \mu_d, \nu_d}) = \tilde{\mathbf{H}}^{<2k>}(\overline{\nu_1, \mu_1, \cdots, \nu_k, \mu_k}; \overline{\nu_{k+1}, \mu_{k+1}, \cdots, \nu_d, \mu_d}). \tag{3.95}$$

From this equality, we can deduce the following relation for the block matrices $\mathbf{B}^{(j_k,-j_k)}$ entry-wise:

$$\mathbf{B}^{(-j_k,j_k)}(\overline{\mu_1, \nu_1, \cdots, \mu_k, \nu_k}; \overline{\mu_{k+1}, \nu_{k+1}, \cdots, \mu_d, \nu_d}) = \mathbf{B}^{(j_k,-j_k)}(\overline{\nu_1, \mu_1, \cdots, \nu_k, \mu_k}; \overline{\nu_{k+1}, \mu_{k+1}, \cdots, \nu_d, \mu_d}). \tag{3.96}$$

It can be verified that the matrices $\mathbf{B}^{(-j_k,j_k)}$ and $\mathbf{B}^{(j_k,-j_k)}$ can be expressed in terms of the nonzero blocks of the TT-cores as follows:

$$\begin{aligned}\mathbf{B}^{(j_k,-j_k)}(\overline{\mu_1, \nu_1, \cdots, \mu_k, \nu_k}; \overline{\mu_{k+1}, \nu_{k+1}, \cdots, \mu_d, \nu_d}) = \mathbf{H}_1^{(0,t_1)}[\mu_1, \nu_1]\mathbf{H}_2^{(t_1,t_2+t_1)}[\mu_2, \nu_2] \cdots \\ \mathbf{H}_k^{(j_k, j_k+t_k)}[\mu_k, \nu_k] \cdots \mathbf{H}_d^{\left(j_k + \sum_{l=k+1}^{d-1} t_l, 0\right)}[\mu_d, \nu_d],\end{aligned} \tag{3.97}$$

where $t_k = q_n(\mu_k) - q_n(\nu_k)$ and $\sum_{l=1}^{d} t_l = 0$. And

$$\begin{aligned}\mathbf{B}^{(-j_k,j_k)}(\overline{\nu_1, \mu_1, \cdots, \nu_k, \mu_k}; \overline{\nu_{k+1}, \mu_{k+1}, \cdots, \nu_d, \mu_d}) = \mathbf{H}_1^{(0,-t_1)}[\nu_1, \mu_1]\mathbf{H}_2^{(-t_1,-t_2-t_1)}[\nu_2, \mu_2] \cdots \\ \mathbf{H}_k^{(-j_k, -j_k-t_k)}[\nu_k, \mu_k] \cdots \mathbf{H}_d^{\left(-j_k - \sum_{l=k+1}^{d-1} t_l, 0\right)}[\nu_d, \mu_d].\end{aligned} \tag{3.98}$$

The nonzero blocks in the TT-cores are the result of applying SVD to equal matrices due to the property of symmetry. By the properties of SVD, it is possible to find an orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{H}_k^{(j_k, j_k+t_k)}[\mu_k, \nu_k] = \mathbf{H}_k^{(-j_k, -j_k-t_k)}[\nu_k, \mu_k]\mathbf{Q}. \tag{3.99}$$

□

Given a p-body particle-number preserving Hamiltonian operator, it is noteworthy that Theorem 4 yields an intriguing result. The block sparsity in the TT-cores allows one to reduce the memory usage when manipulating tensor cores. If we suppose that all the blocks in the TT-cores are of equal sizes, the required memory can be reduced from R^2 to $\frac{R^2}{(np+1)}$ with $R = \max_{1 \leq k \leq d}\{R_k\}$. Moreover, by combining this block-sparse TTO representation of $\mathbf{H}$ with the block-sparse structure in the TT representation of the

eigenfunction , as described in [2] and in the Corollary 3.3.1, basic operations on tensor trains are less expensive as they are performed over independent smaller blocks instead of the whole tensor which makes it possible to use parallelization techniques. Overall, using physical symmetry as the particle number conservation permits the tensor trains to be in a block form which allows one to reduce the computational costs while explicitly preserving some of desired properties of the Hamiltonian operator.

Remark 3.3.4. In quantum chemistry, for the 2-body quantum chemical Hamiltonian operator $\mathbf{H} \in \mathbb{R}^{4^d \times 4^d}$ with $p = 2$, as defined in (1.138), not only is the particle number conserved, but also the z-component of the total spin. It follows that if the Hamiltonian operator, expressed in the spatial-orbital basis, commutes with both $\mathbf{N}$ and $\mathbf{S}^z$:

$$\begin{cases} \mathbf{HN} = \mathbf{NH} \\ \mathbf{HS}^z = \mathbf{S}^z\mathbf{H}. \end{cases} \tag{3.100}$$

then the blocks in the TT-cores of the TTO representation of $\mathbf{H}$ are themselves block-sparse. Specifically, if $\mathbf{H}$ satisfies (PN), (SZ), and (pB), ensuring the following conditions on the entries:

$$\mathbf{H}(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) \neq 0 \text{ if } \begin{cases} \sum_{k=1}^d q_4(\mu_k) = \sum_{k=1}^d q_4(\nu_k), \\ \sum_{k=1}^d q_z(\mu_k) = \sum_{k=1}^d q_z(\nu_k), \\ \sum_{k=1}^d |q_4(\mu_k) - q_4(\nu_k)| \leq 2p, \end{cases} \tag{3.101}$$

where $q_z : \{1,2,3,4\} \mapsto \left\{0, \frac{1}{2}, \frac{-1}{2}, 0\right\}$ and $q_4 : \{1,2,3,4\} \mapsto \{0,1,1,2\}$, the TT-cores of the TTO representation of $\mathbf{H}$ denoted by $\mathbf{H}^{(j_{k-1},j_k)}[\mu_k, \nu_k] \in \mathbb{R}^{\rho_{k-1},j_{k-1} \times \rho_{k,j_k}}$ with $j_k \in \{-\beta_k, \ldots, \beta_k\}$, $\mu_k, \nu_k \in \{1,2,3,4\}$, have themselves a block representation with nonzero blocks on the sets $\rho^{\mathbf{H}}_{k-1,j_{k-1},s_{k-1}} \times \rho^{\mathbf{H}}_{k,j_k,s_k}$ where $s_0 = s_d = 1$ and for $k \in \{2, \ldots, d-1\}$, we have

$$s_k \in \left\{ -\min\left\{\frac{p}{2}, k, d-k\right\}, -\min\left\{\frac{p}{2}, k, d-k\right\} + \frac{1}{2}, \ldots, 0, \ldots, \min\left\{\frac{p}{2}, k, d-k\right\}\right\},$$

and $\sum_{s_k} \rho^{\mathbf{H}}_{k,j_k,s_k} = \rho^{\mathbf{H}}_{k,j_k}$.To establish this block structure, we employ a procedure analogous to the proof outlined in Lemma 3.3.2. We start by introducing the following functions:

$$\begin{aligned} \tilde{\mathcal{Q}}_{\leq k} : \{1,2,3,4\}^k \to \mathbb{N}_0 \quad \tilde{\mathcal{Q}}_{>k} : \{1,2,3,4\}^{d-k} \to \mathbb{N}_0 \\ \mu_{\leq k} \to \sum_{l=1}^k q_z(\mu_l), \qquad \mu_{>k} \to \sum_{l=k+1}^d q_z(\mu_l), \end{aligned} \tag{3.102}$$

where q_z is defined in (SZ). By enforcing the particle number conservation as well as $\mathbf{S}^z$ symmetry, for a fixed value of k, $\tilde{\mathbf{H}}^{<2k>}$ has the following expression:

$$\tilde{\mathbf{H}}^{<2k>} = \sum_{\substack{\mu_1,\cdots,\mu_d \\ \nu_1,\cdots,\nu_d=1}}^{d} \mathbf{H}(\overline{\mu_{\leq k}, \nu_{\leq k}}; \overline{\mu_{>k}, \nu_{>k}}) g(\mu_{\leq k}, \nu_{\leq k}, \mu_{>k}, \nu_{>k}), \tag{3.103}$$

where

$$g(\mu_{\leq k}, \nu_{\leq k}, \mu_{>k}, \nu_{>k}) = \sum_{s_1,\cdots,s_k \in \left\{0, \frac{-1}{2}, \frac{1}{2}, 0\right\}} \sum_{t_1,\cdots,t_k=-2}^{2} \left[\delta^{\mathcal{Q}_{\leq k}(\mu_{\leq k}) - \sum_{l=1}^k t_l}_{\mathcal{Q}_{\leq k}(\nu_{\leq k})} \delta^{\tilde{\mathcal{Q}}_{\leq k}(\mu_{\leq k}) - \sum_{l=1}^k s_l}_{\tilde{\mathcal{Q}}_{\leq k}(\nu_{\leq k})} \boldsymbol{e}_{\mu_{\leq k}} \otimes_K \boldsymbol{e}_{\nu_{\leq k}}\right]$$
$$\left[\delta^{\mathcal{Q}_{>k}(\mu_{>k})}_{\mathcal{Q}_{>k}(\nu_{>k}) - \sum_{l=1}^k t_l} \delta^{\tilde{\mathcal{Q}}_{>k}(\mu_{>k})}_{\tilde{\mathcal{Q}}_{>k}(\nu_{>k}) - \sum_{l=1}^k s_l} \boldsymbol{e}_{\mu_{>k}} \otimes_K \boldsymbol{e}_{\nu_{>k}}\right]^*,$$

where $t_k = q_4(\mu_k) - q_4(\nu_k)$ and $s_k = q_z(\mu_k) - q_z(\nu_k)$. The newly derived expression for the mode-$(1:2k)$ matricization of the Hamiltonian operator, as presented in Equation (3.103), reveals that for each value of k, the corresponding mode-$(1:2k)$ matricization is a block-sparse matrix. Based on the preceding analysis, we can proceed in a manner analogous to that used in the proof of Lemma (3.3.2) in order to arrive at the desired conclusion.

Up to this point, our considerations have focused on symmetries as the conservation of the particle number and z-component of total spin, associated with the Abelian unitary group U(1). However, when dealing with other symmetries, such as the conservation of total spin linked to the non-Abelian group SU(2). The analysis becomes more intricate. The reasoning employed earlier to derive the block-structure in the TT-cores does not directly apply. We delve into this complexity in the following section, where our aim is to derive the structure of the TT representation of the wavefunction acting as an eigenfunction of the total spin operator.

3.3.2 A comprehensive study of the TT representation of the wavefunction as the eigenfunction of the total spin operator: preliminary theoretical insights

In Section 3.3.1, we reviewed the TT representation of the wavefunction, which serves as an eigenfunction of the particle number operator (i.e invariant under U(1) symmetry), resulting in a block-sparse structure for the TT-cores. In this section, we consider a wavefunction that acts as an eigenfunction of the total spin operator represented by $\hat{S}^2$, see Equation (3.3). Here, we aim to formulate the TT-cores expression within the TT-representation of the eigenfunction invariant under these symmetries. This TT representation stands apart from the one obtained earlier in Section 3.3.1. Here, we will consider the wavefunction as an eigenfunction of three operators: the particle number operator represented by $\mathbf{N}$, the z-component of the total spin operator represented by $\mathbf{S}^z$, and the total spin operator represented by $\hat{S}^2$, defined in (1.115), (3.1), and (3.3) respectively. First, we justify the choice of transitioning from the canonical basis, as introduced in (1.120), to a more suitable *symmetry-adapted basis* as will be introduced in Definition 30, in which the wavefunction can be expanded effectively. Using this basis, we give the new decomposition of the Fock spaces. Subsequently, we present the expression for reshaping the wavefunction, for a singlet state, in the context of a bipartite system which yields to a block-diagonal matrix. Moving forward, we use the properties of SVD and revisit the TT-decomposition of the wavefunction in the canonical basis. By doing so, we derive the formulation for the TT-cores within the TT-decomposition of the wavefunction in the symmetry-adapted basis as will be outlined in Theorem 5.

Remark 3.3.5. We consider the spatial-orbital basis, where each orbital has four possible occupation states: absent (0), occupied with spin up (↑), occupied with spin down (↓), and present with both spin up and down (2). Moving forward, we represent the number of spatial-orbitals as d. Previously, within the spatial-orbital basis, we labeled tensor indices as shown in (PN) for instance, with $\mu_k \in \{1,2,3,4\}$, $k \in [d]$. Each value of μ_k corresponds to one of the possible orbital occupation states. For clarity and ease of explanation going forward, we will directly describe the values of μ_k by their corresponding occupation states, i.e:

$$\mu_k \in \{1,2,3,4\} := \{0,\uparrow,\downarrow,2\}\,. \tag{3.104}$$

Fock space and SU(2) symmetry

We begin by recalling the definitions of the discrete Fock space associated with d spatial-orbitals, denoted as $\boldsymbol{\mathcal{F}}_d$. Similarly, we consider the discrete Fock space for the initial k spatial-orbitals denoted as $\boldsymbol{\mathcal{F}}_k$, as well as the discrete Fock space for the remaining $d-k$ orbitals, referred to as $\boldsymbol{\mathcal{F}}_{d-k}$.

Definition 28. Let $\boldsymbol{\mathcal{F}}_d$, as already defined in (1.97), be the fermionic Fock space defined by:

$$\boldsymbol{\mathcal{F}}_d := \mathrm{Span}\left\{\Phi_{\mu_1\dots\mu_d}\middle|\mu_k \in \{0,\uparrow,\downarrow,2\}, k \in [d]\right\}, \tag{3.105}$$

with dimension $\dim(\boldsymbol{\mathcal{F}}_d) = 4^d$ and $\Phi_{\mu_1\dots\mu_d}$ being the Slater determinants as introduced in (1.71) and in Definition 21.

Definition 29. Let $\boldsymbol{\mathcal{F}}_k$ be the fermionic Fock space of the k first orbitals defined by:

$$\boldsymbol{\mathcal{F}}_k := \mathrm{Span}\left\{\Phi_{\mu_1\dots\mu_k}\middle|\mu_i \in \{0,\uparrow,\downarrow,2\}, i \in [k]\right\}, \tag{3.106}$$

with dimension $\dim(\boldsymbol{\mathcal{F}}_k) = 4^k$, and $\boldsymbol{\mathcal{F}}_{d-k}$ be the fermionic Fock space of the remaining $d-k$ orbitals defined by:

$$\boldsymbol{\mathcal{F}}_{d-k} := \mathrm{Span}\left\{\Phi_{\mu_{k+1}\dots\mu_d}\middle|\mu_i \in \{0,\uparrow,\downarrow,2\}, i \in \{k+1,\dots,d\}\right\}, \tag{3.107}$$

with dimension $\dim(\boldsymbol{\mathcal{F}}_{d-k}) = 4^{d-k}$.

By showing that $\boldsymbol{\mathcal{F}}_d$ is isomorphic to $\mathbb{R}^{4^d}$, see Remark 1.3.9, each wavefunction represented by $\boldsymbol{\Psi} \in \mathbb{R}^{4^d}$, writes as:

$$\boldsymbol{\Psi} = \sum_{\mu_1,\cdots,\mu_d \in \{0,\uparrow,\downarrow,2\}} \Psi_{\mu_1\cdots\mu_d} \boldsymbol{e}_{\mu_1\dots\mu_d}, \in \mathbb{R}^{4^d}, \tag{3.108}$$

with $\boldsymbol{e}_{\mu_1\dots\mu_d} = \boldsymbol{e}_{\mu_1} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_d}$. Suppose that $\boldsymbol{\Psi}$ satisfies the following eigenvalue relations, see [2, 111, 133]:

$$\mathbf{N}\boldsymbol{\Psi} = N\boldsymbol{\Psi},\ \hat{S}^2\boldsymbol{\Psi} = J(J+1)\boldsymbol{\Psi},\ \mathbf{S}^z\boldsymbol{\Psi} = M\boldsymbol{\Psi}. \tag{3.109}$$

The eigenvalues depending on N,M and J are the so-called *quantum numbers*. N accounts for the number of particles, J is the total spin with $J \in \left\{0, \frac{1}{2}, 1, \frac{3}{2}, \cdots, \frac{d}{2}\right\}$ and $M \in \{-J, \dots, J\}$ is the angular momentum projection onto the z-axis. The vector $\boldsymbol{e}_{\mu_k}$ for $k \in [d]$ is an eigenvector of the particle number operator, the z-component of the total spin operator and the total spin operator, such that

$$\mathbf{N}^{\leq 1}\boldsymbol{e}_{\mu_k} = \begin{cases} 0 & \text{if } \mu_k = 0, \\ 1 & \text{if } \mu_k = \uparrow \text{ or } \mu_k = \downarrow, \\ 2 & \text{if } \mu_k = 2, \end{cases} \ (\mathbf{S}^z)^{\leq 1}\boldsymbol{e}_{\mu_k} = \begin{cases} 0 & \text{if } \mu_k = 0, \\ \frac{1}{2} & \text{if } \mu_k = \uparrow, \\ \frac{-1}{2} & \text{if } \mu_k = \downarrow, \\ 0 & \text{if } \mu_k = 2, \end{cases}, \tag{3.110}$$

and

$$(\hat{S}^2)^{\leq 1}\boldsymbol{e}_{\mu_k} = \begin{cases} 0 & \text{if } \mu_k = 0, (J=0), \\ \frac{3}{4} & \text{if } \mu_k = \uparrow \text{ or } \mu_k = \downarrow, (J=\frac{1}{2}), \\ 0 & \text{if } \mu_k = 2, (J=0), \end{cases} \tag{3.111}$$

with $\mathbf{N}^{\leq 1}, (\mathbf{S}^z)^{\leq 1}, (\hat{S}^2)^{\leq 1}$ are defined in Equations (1.115), (3.1), and (3.3) by taking $d = 1$. We already observe that J depends only on singly occupied orbitals. Additionally, we observe that a single vector, $\boldsymbol{e}_{\mu_k}$, serves as an eigenvector for all three operators, however, this does not hold for certain elements within the chosen canonical basis of the Fock space $\boldsymbol{\mathcal{F}}_d = \text{Span}\left\{\boldsymbol{e}_{\mu_1\dots\mu_d} \middle| \mu_k \in \{0, \uparrow, \downarrow, 2\}, k \in [d]\right\}$, as illustrated in the Example 3.3.2. It is more convenient to write the wavefunction in the common basis of the eigenvectors of $\mathbf{N}, \hat{S}^2$ and $\mathbf{S}^z$. We illustrate this explicitly in the following example.

Example 3.3.2. Suppose $d = 2$, $\boldsymbol{\Psi}$ writes as, in terms of the canonical basis $\left\{\boldsymbol{e}_{\mu_1\mu_2}\right\}_{\mu_1,\mu_2\in\{0,\uparrow,\downarrow,2\}}$, as follows:

$$\begin{aligned} \boldsymbol{\Psi} = {} & \Psi_{00}\boldsymbol{e}_{00} + \Psi_{\uparrow 0}\boldsymbol{e}_{\uparrow 0} + \Psi_{\downarrow 0}\boldsymbol{e}_{\downarrow 0} + \Psi_{0\uparrow}\boldsymbol{e}_{0\uparrow} + \Psi_{0\downarrow}\boldsymbol{e}_{0\downarrow} + \Psi_{\uparrow\uparrow}\boldsymbol{e}_{\uparrow\uparrow} + \Psi_{\downarrow\downarrow}\boldsymbol{e}_{\downarrow\downarrow} \\ & + \Psi_{\uparrow\downarrow}\boldsymbol{e}_{\uparrow\downarrow} + \Psi_{\downarrow\uparrow}\boldsymbol{e}_{\downarrow\uparrow} + \Psi_{20}\boldsymbol{e}_{20} + \Psi_{02}\boldsymbol{e}_{02} + \dots, \end{aligned} \tag{3.112}$$

with $\sum_{\mu_1,\mu_2\in\{0,\uparrow,\downarrow,2\}} |\Psi_{\mu_1\mu_2}|^2 = 1$. If $\boldsymbol{\Psi}$ is an eigenvector of both $\mathbf{S}^z$ and $\mathbf{N}$ with respective eigenvalues M and N, certain conditions are imposed on the entries of $\boldsymbol{\Psi}$ as follows:

$$\Psi_{\mu_1\cdots\mu_d} \neq 0 \text{ if } \sum_{i=1}^{d} q_4(\mu_i) = N \text{ and } \sum_{i=1}^{d} q_z(\mu_i) = M, \tag{3.113}$$

where $q_4 : \{0, \uparrow, \downarrow, 2\} \mapsto \{0, 1, 1, 2\}$ and $q_z : \{0, \uparrow, \downarrow, 2\} \mapsto \left\{0, \frac{1}{2}, \frac{-1}{2}, 0\right\}$. Back to (3.112), by supposing that $N = 2$ and $M = 0$, we observe a reduction in the number of terms in Equation (3.114). Here, we denote $\boldsymbol{\Psi}^{N,M}$ as the target wavefunction with a total particle number of $N = 2$ and spin of $M = 0$, such that

$$\boldsymbol{\Psi}^{N,M} := \boldsymbol{\Psi}^{2,0} = \Psi_{\uparrow\downarrow}\boldsymbol{e}_{\uparrow\downarrow} + \Psi_{\downarrow\uparrow}\boldsymbol{e}_{\downarrow\uparrow} + \Psi_{02}\boldsymbol{e}_{02} + \Psi_{20}\boldsymbol{e}_{20}. \tag{3.114}$$

We can see that $\boldsymbol{\Psi}^{N,M} \in \text{Span}\{\boldsymbol{e}_{\uparrow\downarrow}, \boldsymbol{e}_{\downarrow\uparrow}, \boldsymbol{e}_{02}, \boldsymbol{e}_{20}\}$. Now, a simple calculation shows that $\boldsymbol{e}_{\uparrow\downarrow}$ and $\boldsymbol{e}_{\downarrow\uparrow}$ are not eigenvectors of $\hat{S}^2$.

$$\begin{aligned} \hat{S}^2\boldsymbol{e}_{\downarrow\uparrow} &= (\frac{1}{2}(\mathbf{S}^+\mathbf{S}^- + \mathbf{S}^-\mathbf{S}^+) + \mathbf{S}^z\mathbf{S}^z)\boldsymbol{e}_{\downarrow\uparrow} \\ &= (\frac{1}{2}(\mathbf{S}^+\mathbf{S}^- + \mathbf{S}^-\mathbf{S}^+)(\boldsymbol{e}_{\downarrow\uparrow}) + \underbrace{\mathbf{S}^z\mathbf{S}^z(\boldsymbol{e}_{\downarrow\uparrow})}_{=\mathbf{0}} = \alpha^-_{\frac{1}{2},\frac{1}{2}}\alpha^+_{\frac{1}{2},\frac{-1}{2}}(\boldsymbol{e}_{\downarrow\uparrow} + \boldsymbol{e}_{\uparrow\downarrow}), \end{aligned} \tag{3.115}$$

with $\alpha^-_{\frac{1}{2},\frac{1}{2}}\alpha^+_{\frac{1}{2},\frac{-1}{2}}$ being the coefficients obtained by applying $\mathbf{S}^+$ and $\mathbf{S}^-$ operators, defined in (3.5). The new basis would naturally be the span of the eigenvectors of $\hat{S}^2, \mathbf{S}^z$ and $\mathbf{N}$ referred to as the symmetry-adapted basis.

In what follows, we establish a well-defined expression for the total particle number N: let N_p be the number of orbitals that are doubly occupied, i.e occupied with both

spin up and spin down, and let N_s be the number of orbitals that are singly occupied, i.e., occupied by a spin up or a spin down, such that N can be expressed as:

$$N = 2N_p + N_s. \tag{3.116}$$

For instance, the first term in the example (3.114), $N_p = 0$ and $N_s = 2$ which corresponds to having 2 singly occupied orbitals: a spin up in the first orbital and spin down in the second orbital. Now, we have all the ingredients to write down $\Psi^{N,M}$ in terms of the symmetry-adapted basis that we denote by $\boldsymbol{\xi}^{N_p,N_s,t}_{d,N,J,M}$, where t is the degeneracy of the corresponding subspace. Let $\boldsymbol{\Psi}^{N,M}$ be the wavefunction with N being the particle number and M being the z-component of the total spin, such that

$$\boldsymbol{\Psi}^{2,0} = \Psi^{1,0,1}_{2,2,0,0}\boldsymbol{\xi}^{1,0,1}_{2,2,0,0} + \Psi^{1,0,2}_{2,2,0,0}\boldsymbol{\xi}^{1,0,2}_{2,2,0,0} + \Psi^{0,2,3}_{2,2,0,0}\boldsymbol{\xi}^{0,2,3}_{2,2,0,0} + \Psi^{0,2,1}_{2,2,1,0}\boldsymbol{\xi}^{0,2,1}_{2,2,1,0}, \tag{3.117}$$

where we can readily identify that

$$\boldsymbol{\xi}^{1,0,1}_{2,2,0,0} = \boldsymbol{e}_{02},\ \boldsymbol{\xi}^{1,0,2}_{2,2,0,0} = \boldsymbol{e}_{20}, \tag{3.118}$$

$$\boldsymbol{\xi}^{0,2,3}_{2,2,0,0} = \frac{\boldsymbol{e}_{\uparrow\downarrow} - \boldsymbol{e}_{\downarrow\uparrow}}{\sqrt{2}}, \tag{3.119}$$

$$\boldsymbol{\xi}^{0,2,1}_{2,2,1,0} = \frac{\boldsymbol{e}_{\uparrow\downarrow} + \boldsymbol{e}_{\downarrow\uparrow}}{\sqrt{2}}, \tag{3.120}$$

$$\Psi^{0,2,3}_{2,2,0,0} = \frac{\Psi_{\uparrow\downarrow} - \Psi_{\downarrow\uparrow}}{\sqrt{2}}, \Psi^{0,2,1}_{2,2,1,0} = \frac{\Psi_{\uparrow\downarrow} + \Psi_{\downarrow\uparrow}}{\sqrt{2}}, \tag{3.121}$$

$$\Psi^{1,0,1}_{2,2,0,0} = \Psi_{02}, \Psi^{1,0,2}_{2,2,0,0} = \Psi_{20}. \tag{3.122}$$

The symmetry-adapted basis is assigned labels based on the quantum numbers N, J, M, and the value N_p and N_s such that $N = 2N_p + N_s$. We note that the use of the symmetry-adapted basis provides a clearer physical insight which allows to target specific states with targeted quantum numbers. Additionally, it is noted that in this example $t \in \{1, \ldots, 3\}$ refers to the degeneracy of J. Now, suppose that the total spin is $J = 0$, then $\Psi^{0,2,1}_{2,2,1,0} = 0$. Let $\boldsymbol{\Psi}^{N,J,M}$ be the wavefunction with N particle number, M z-component of the total spin , and J the total spin , such that

$$\boldsymbol{\Psi}^{N,J,M} := \boldsymbol{\Psi}^{2,0,0} = \Psi^{1,0,1}_{2,2,0,0}\boldsymbol{\xi}^{1,0,1}_{2,2,0,0} + \Psi^{1,0,2}_{2,2,0,0}\boldsymbol{\xi}^{1,0,2}_{2,2,0,0} + \Psi^{0,2,3}_{2,2,0,0}\boldsymbol{\xi}^{0,2,3}_{2,2,0,0}. \tag{3.123}$$

Back to Equation (3.110), we see that the total spin J only depends on the number of singly occupied orbitals. We derive in what follows the theoretical bounds on the total spin J, in terms of only N_s. By employing a recurrence approach, we can establish the boundaries for J, as outlined in the following proposition.

Proposition 3.3.5. *Let N_s be the number of singly occupied orbitals, then the following holds true:*

$$\frac{mod(N_s, 2)}{2} \leq J \leq \frac{N_s}{2}, \tag{3.124}$$

with J referring to the total spin of the wavefunction.

Proof. We recall that if we have two singly occupied spatial-orbitals, associated with J_1 and J_2, respectively, when combined together, the resulting total spin represented as J satisfies the following condition, see [128]:

$$|J_1 - J_2| \leq J \leq |J_1 + J_2|\,. \tag{3.125}$$

Suppose that the number of singly occupied orbitals is denoted by N_s. Proceeding with the recurrence, as we transition from one orbital to another, N_s increases by one. We initiate the proof by confirming its validity for $N_s = 2$. Let us combine the two singly occupied orbitals, with $J^{\leq 1} := J_1 = \frac{1}{2}$ and $J_2 = \frac{1}{2}$ being their respective total spins. Let $J^{\leq 2}$ be the resulting total spin. According to (3.125), one obtains:

$$0 \leq J^{\leq 2} \leq 1. \tag{3.126}$$

Now, assuming that for all $N_s \in [d]$, we have the following bounds:

$$\frac{mod(N_s,2)}{2} \leq J^{\leq N_s} \leq \frac{N_s}{2}. \tag{3.127}$$

We aim to demonstrate that for N_s+1, the following also holds true:

$$\frac{mod(N_s+1,2)}{2} \leq J^{\leq N_s+1} \leq \frac{N_s+1}{2}. \tag{3.128}$$

By combining N_s singly occupied orbitals with an additional singly occupied orbital featuring $J_{N_s+1} = \frac{1}{2}$, we establish an upper bound:

$$J^{\leq N_s+1} \leq J^{\leq N_s} + J_{N_s+1} = J^{\leq N_s} + \frac{1}{2} \leq \frac{N_s+1}{2}. \tag{3.129}$$

For the lower bound, if N_s is even:

$$\left|\frac{mod(N_s,2)}{2} - \frac{1}{2}\right| = \frac{1}{2} = \left|\frac{mod(N_s+1,2)}{2}\right|. \tag{3.130}$$

If N_s is odd:

$$\left|\frac{mod(N_s,2)}{2} - \frac{1}{2}\right| = 0 = \left|\frac{mod(N_s+1,2)}{2}\right|. \tag{3.131}$$

Thus, we conclude:

$$\frac{mod(N_s+1,2)}{2} \leq J^{\leq N_s+1} \leq \frac{N_s+1}{2}. \tag{3.132}$$

This concludes the proof. □

In the following, we give the expression of the decomposition of the fermionic Fock space $\boldsymbol{\mathcal{F}}_d$ when employing the symmetry-adapted basis.

Proposition 3.3.6. *The fermionic Fock space $\boldsymbol{\mathcal{F}}_d$ has the following decomposition:*

$$\boldsymbol{\mathcal{F}}_d = \bigoplus_{N=0}^{2d} \bigoplus_{\substack{N_s,N_p=0 \\ 2N_p+N_s=N}}^{d} \bigoplus_{J=\frac{\text{mod }(N_s,2)}{2}}^{J=\frac{N_s}{2}} \bigoplus_{M=-J}^{J} \boldsymbol{\mathcal{F}}_{d,N,J,M}^{N_p,N_s}. \tag{3.133}$$

For all values of N, N_p and N_s such that $N = 2N_p + N_s$, for all J, M, there are vector spaces $\mathscr{P}_{d,N,J}^{N_p,N_s}$ and $\boldsymbol{\mathcal{S}}_{d,J,M}$ such that

$$\boldsymbol{\mathcal{F}}_{d,N,J,M}^{N_p,N_s} = \mathscr{P}_{d,N,J}^{N_p,N_s} \otimes \boldsymbol{\mathcal{S}}_{d,J,M}, \tag{3.134}$$

where $\mathcal{S}_{d,J,M}$ is a one-dimensional space. Moreover $\mathscr{P}_{d,N,J}^{N_p,N_s}$ has dimension:

$$\dim(\mathscr{P}_{d,N,J}^{N_p,N_s}) = \binom{d}{N_p}\binom{d-N_p}{N_s}\tau^{N_s,J}, \tag{3.135}$$

with

$$\tau^{N_s,J} = \binom{N_s}{\frac{N_s}{2}-J} - \binom{N_s}{\frac{N_s}{2}-J-1}, \tag{3.136}$$

and $\bigoplus_{M=-J}^{J} \boldsymbol{\mathcal{S}}_{d,J,M}$ is a $(2J+1)$-dimensional space.

Proof. The decomposition in Equation (3.133) arises from the fact that the Fock space is spanned by the eigenvectors of the operators $\mathbf{N}$, $\mathbf{S}^z$, and $\hat{S}^2$, all of which share a common basis. As a result, $\boldsymbol{\mathcal{F}}_d$ can be expressed as the direct sum of distinct eigenspaces associated with these operators. Equation (3.134) is derived from a well-established result involving the action of SU(2) symmetry on the finite-dimensional vector subspaces $\boldsymbol{\mathcal{F}}_{d,N,J,M}^{N_p,N_s}$ spanned by the orthonormal symmetry-adapted basis sets denoted as $\left\{\boldsymbol{\xi}_{d,N,J,M}^{N_p,N_s,t}\right\}_{t\in[\dim(\mathscr{P}_{d,N,J}^{N_p,N_s})]}$ [104, 112]. This action allows to decompose the Fock space into the direct sum of vector spaces denoted by $\mathscr{P}_{d,N,J}^{N_p,N_s}$ spanned by the orthonormal basis, denoted as $\left\{\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t}\right\}_{t\in[\dim(\mathscr{P}_{d,N,J}^{N_p,N_s})]}$, with t referring to the degeneracy of the total spin

J, along with a $(2J+1)$-dimensional vector space spanned by the orthonormal basis, denoted as $\{\boldsymbol{s}_{d,J,M}\}_{M\in\{-J,\ldots,J\}}$. The dimension of $\mathscr{P}^{N_p,N_s}_{d,N,J}$ can be determined through further manipulations, which we will discuss with an illustration for a more comprehensive explanation.

The dimension $\dim(\mathscr{P}^{N_p,N_s}_{d,N,J})$ refers to the degeneracy associated with the total spin J which also means the number of times $\bigoplus_{M=-J}^{J}\boldsymbol{\mathcal{S}}_{d,J,M}$ appears in the decomposition, and $\tau^{N_s,J}$ can be interpreted as the number of potential paths required, while keeping N_s fixed, to reach the value of J. J can be reached by progressively coupling total spins starting from the first orbital and proceeding to the last one. To explain this process, we start by combining the first two orbitals, characterized by different total spins J_1 and J_2. This results in a new state with total spin $J^{\leq 2}$, which satisfies the condition $|J_1 - J_2| \leq J^{\leq 2} \leq |J_1 + J_2|$. We then proceed by coupling this newly obtained state with the next orbital associated with a total spin J_3, yielding a further new state with $J^{\leq 3}$ satisfying $\left|J^{\leq 2} - J_3\right| \leq J^{\leq 3} \leq \left|J^{\leq 2} + J_3\right|$. This process continues until we reach a state with the target total spin value of J, which satisfies the condition $\left|J^{\leq d-2} - J_{d-1}\right| \leq J \leq \left|J^{\leq d-2} + J_{d-1}\right|$. To summarize, according to (3.110), each orbital offers two options for the total spin: $J = 0$ when the orbital is either doubly occupied or empty, and $J = \frac{1}{2}$ when the orbital is singly occupied. Notably, the value of J relies on the presence of singly occupied orbitals. Therefore, our focus narrows to the singly occupied orbitals, wherein each coupling step introduces a choice: either increasing or decreasing the value of J. Therefore, we can interpret this by having upward or downward paths. Figure 3.4 illustrates this concept for $d = 4$, assuming all orbitals are singly occupied, i.e $N_s = 4$, resulting in a structured pattern resembling a half binary tree. The goal here is to determine $\tau^{N_s,J}$ which is the total number of paths required to attain the desired total spin value J.

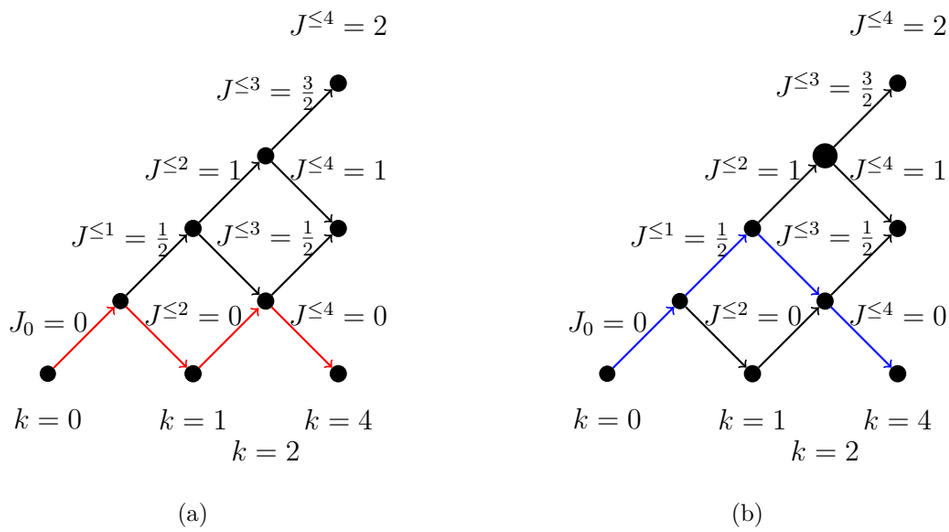


Figure 3.4: The total number of schemes needed to reach $J = J^{\leq 4} = 0$ is 2, with a fixed $N_s = 4$, corresponding to the configurations highlighted in blue and red.

Now, to establish a general formulation for the number of paths required to reach the total spin J, we start by considering a binary tree structure. In this binary tree, each level consists of nodes, with each node on a given level having at most two children, labeled as $J^{\leq k}$, where $k \in [N_s]$. For simplicity, we assign an identifier to each node on every level, ranging from 0 to the number of nodes within that level, as depicted in Figure 3.5.

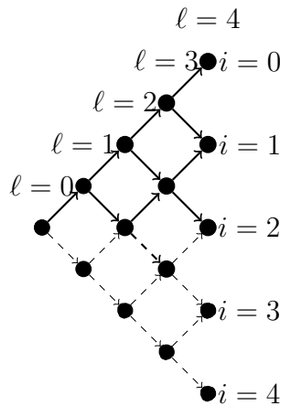


Figure 3.5: Binary tree structure: levels are denoted by ℓ and target node is denoted by i.

To reach a node i within level ℓ, one typically requires $\binom{\ell}{i}$ paths. For instance, in Figure 3.5, to reach node $i = 0$ at level $\ell = 4$, there is only, one possible scheme ($\binom{4}{0} = 1$). However, since we consider a half binary tree structure, we need to account for the exclusion of certain paths, referred to as "bad paths". These "bad paths" are the ones that involve dashed arrows (as illustrated in Figure 3.5). To address this, we introduce the following mapping:

$$\phi_i^\ell : \bar{\mathcal{P}}_i^\ell \mapsto \mathcal{P}_{i-1}^\ell, \tag{3.137}$$

where

$$\mathcal{P}_{i-1}^\ell = \{\text{Set of all paths leading to node } i-1 \text{ within level } \ell\}\,, \quad \#\mathcal{P}_{i-1}^\ell = \binom{\ell}{i-1}, \tag{3.138}$$

and

$$\bar{\mathcal{P}}_i^\ell = \{\text{Set of all bad paths leading to node } i \text{ within level } \ell\}\,. \tag{3.139}$$

Each element within the set $\bar{\mathcal{P}}_i^\ell$ can be represented as a sequence $(m_1, m_2, \ldots, m_\ell)$, where $m_k = \pm 1$ for $k \in [\ell]$. We can define ϕ_i^ℓ as follows:

$$\phi_i^\ell(m_1, m_2, \ldots, m_s, \ldots, m_\ell) = (-m_1, -m_2, \ldots, -m_s, m_{s+1}, \ldots, m_\ell), \tag{3.140}$$

where s is defined as $\sum_{k=s}^{\ell} m_k = (\ell - 2i + 1)$. On can verify that ϕ_i^ℓ is a bijective function by showing that is both injective and surjective. Therefore, we establish that $\#\mathcal{P}_{i-1}^\ell = \#\bar{\mathcal{P}}_i^\ell = \binom{\ell}{i-1}$. Consequently, the number of desired paths, can be calculated as $\binom{\ell}{i} - \binom{\ell}{i-1}$, in our case $\ell = N_s$ and $i = \frac{N_s}{2} - J$. Therefore, we conclude that, the number of good paths is $\binom{N_s}{\frac{N_s}{2}-J} - \binom{N_s}{\frac{N_s}{2}-J-1}$. In Figures 3.6 and 3.7, we illustrate the concept of "bad paths" and how the mapping ϕ_i^ℓ transforms these bad paths in Figure 3.8.

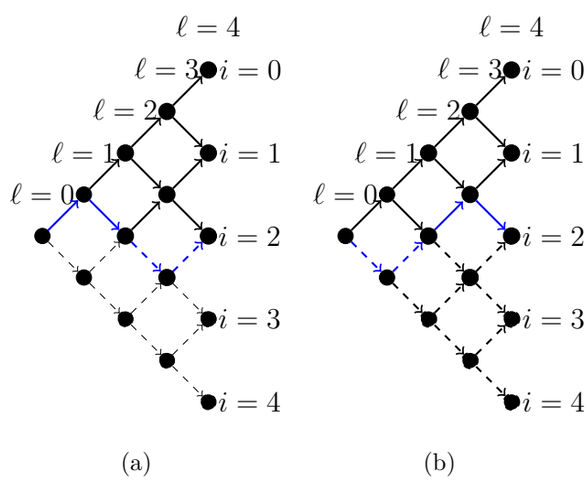


Figure 3.6: Bad paths (blue schemes) to reach node 2, i.e $i = 2$, at level $\ell = 4$.

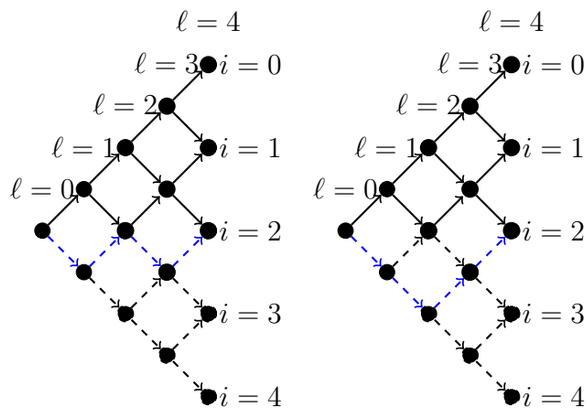


Figure 3.7: Bad paths (blue schemes) to reach node 2, i.e $i = 2$, at level $\ell = 4$.

We give the following illustration to show how the mapping function is applied

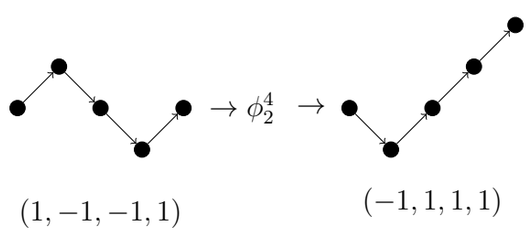


Figure 3.8: $\phi_2^4(1, -1, -1, 1) = (-1, 1, 1, 1)$ and $s = 3$.

□

Moving forward, we introduce the following definition

Definition 30 (Symmetry-adapted basis)**.** Let $\left\{\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t}\right\}_{t\in[\dim(\mathscr{P}_{d,N,J}^{N_p,N_s})]}$ be a basis of $\mathscr{P}_{d,N,J}^{N_p,N_s}$ and $\{\boldsymbol{s}_{d,J,M}\}_{M\in\{-J,\dots,J\}}$ be a basis of $\boldsymbol{\mathcal{S}}_{d,J,M}$. We say that $\boldsymbol{\xi}_{d,N,J,M}^{N_p,N_s,t} = \boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}$ is a symmetry-adapted basis of $\boldsymbol{\mathcal{F}}_d$. According to Proposition 3.3.6, vector spaces $\boldsymbol{\mathcal{F}}_{d,N,J,M}^{N_p,N_s}$ can be defined as:

$$\begin{aligned}\boldsymbol{\mathcal{F}}_{d,N,J,M}^{N_p,N_s} &:= \operatorname{Span}\left\{\boldsymbol{\xi}_{d,N,J,M}^{N_p,N_s,t}\middle| t \in \left[\dim(\mathscr{P}_{d,N,J}^{N_p,N_s})\right]\right\} \\ &= \operatorname{Span}\left\{\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}\middle| t \in \left[\dim(\mathscr{P}_{d,N,J}^{N_p,N_s})\right]\right\}.\end{aligned} \tag{3.141}$$

Remark 3.3.6. We observe that $\boldsymbol{e}_{\mu_k}$ serves as an eigenvector for the operators $\hat{S}^2$, $\mathbf{N}$, and $\mathbf{S}^z$, as outlined in (3.110), with corresponding eigenvalues $J(\mu_k)$, $N(\mu_k)$, and $M(\mu_k)$. The latter can be defined as follows:

$$N(\mu_k) = \begin{cases} 0 & \text{if } \mu_k = 0, \\ 1 & \text{if } \mu_k = \uparrow \text{ or } \mu_k = \downarrow, \\ 2 & \text{if } \mu_k = 2, \end{cases} \tag{3.142}$$

and

$$J(\mu_k) = \begin{cases} 0 & \text{if } \mu_k = 0 \text{ or } \mu_k = 2, \\ \frac{1}{2} & \text{if } \mu_k = \uparrow \text{ or } \mu_k = \downarrow, \end{cases} \qquad M(\mu_k) = \begin{cases} 0 & \text{if } \mu_k = 0 \text{ or } \mu_k = 2, \\ \frac{1}{2} & \text{if } \mu_k = \uparrow, \\ \frac{-1}{2} & \text{if } \mu_k = \downarrow. \end{cases} \tag{3.143}$$

Thus, we can define $\boldsymbol{e}_{\mu_k}$ in terms of the symmetry-adapted basis as follows:

$$\boldsymbol{e}_{\mu_k} := \boldsymbol{\xi}_{N(\mu_k),J(\mu_k),M(\mu_k)}^{N_p(\mu_k),N_s(\mu_k),1} \equiv \boldsymbol{\varrho}_{N(\mu_k),J(\mu_k)}^{N_p(\mu_k),N_s(\mu_k),1} \otimes_K \boldsymbol{s}_{J(\mu_k),M(\mu_k)}. \tag{3.144}$$

To present a comprehensive representation of $\mathbf{\Psi}$ within the Fock space, considering its decomposition as indicated in (3.133), we introduce the following Corollary.

Corollary 3.3.7. *Let $\mathbf{\Psi} \in \boldsymbol{\mathcal{F}}_d$, such that, according to Proposition 3.133*

$$\mathbf{N}\mathbf{\Psi} = N\mathbf{\Psi}, \quad \hat{S}^2\mathbf{\Psi} = J(J+1)\mathbf{\Psi}, \quad \textit{and} \quad \mathbf{S}^z\mathbf{\Psi} = M\mathbf{\Psi}.$$

$$\begin{aligned}\mathbf{\Psi} &= \sum_{N=0}^{2d} \sum_{\substack{N_p,N_s=0 \\ N=2N_p+N_s}}^{d} \sum_{J=\frac{mod(N_s,2)}{2}}^{\frac{N_s}{2}} \sum_{M=-J}^{J} \sum_{t=1}^{\dim(\mathscr{P}^{N_p,N_s}_{d,N,J})} \Psi^{N_p,N_s,t}_{d,N,J,M} \boldsymbol{\xi}^{N_p,N_s,t}_{d,N,J,M} \\ &= \sum_{N=0}^{2d} \sum_{\substack{N_p,N_s=0 \\ N=2N_p+N_s}}^{d} \sum_{J=\frac{mod(N_s,2)}{2}}^{\frac{N_s}{2}} \sum_{M=-J}^{J} \sum_{t=1}^{\dim(\mathscr{P}^{N_p,N_s}_{d,N,J})} \Psi^{N_p,N_s,t}_{d,N,J,M} \boldsymbol{\varrho}^{N_p,N_s,t}_{d,N,J} \otimes_K \boldsymbol{s}_{d,J,M},\end{aligned} \tag{3.145}$$

with $\Psi^{N_p,N_s,t}_{d,N,J,M} \in \mathbb{R}$.

Now, in order to express the TT-cores in the TT representation of $\mathbf{\Psi}$, we first require the decomposition of the Fock space $\boldsymbol{\mathcal{F}}_d$ for a bipartite system with a fixed value of $k \in [d]$. As it turns out, for the total spin, a tensorization of a symmetry-adapted basis of $\boldsymbol{\mathcal{F}}_k$ and $\boldsymbol{\mathcal{F}}_{d-k}$ does not necessarily yield an eigenvector of $\boldsymbol{\mathcal{F}}_d$ that satisfy Equation (3.109). One needs to linearly combine the tensorization of the symmetry-adapted basis to retrieve a symmetry-adapted basis of the full Fock space $\boldsymbol{\mathcal{F}}_d$. This requires the introduction of coupling coefficients, as detailed in the following proposition.

Proposition 3.3.8. *Let $\boldsymbol{\varrho}^{(N_p,N_s,t)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}}$ (resp. $\boldsymbol{\varrho}^{(N_p,N_s,t)^{>k}}_{d-k,(N,J)^{>k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}$) be an orthonormal symmetry-adapted basis of $\boldsymbol{\mathcal{F}}_k$ (resp. $\boldsymbol{\mathcal{F}}_{d-k}$).*

Then

$$\begin{aligned}\boldsymbol{\varrho}^{N_p,N_s,t}_{d,N,J} \otimes_K \boldsymbol{s}_{d,J,M} \equiv \sum_{\substack{M^{\leq k},M^{>k} \\ N^{\leq k},N^{>k},N_p^{\leq k},N_p^{>k},N_s^{\leq k},N_s^{>k} \\ J^{\leq k},J^{>k},t^{\leq k},t^{>k}}} & \chi^{N,N_p,N_s,J,t}_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}} C^{JM}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}} \\ & \boldsymbol{\varrho}^{(N_p,N_s,t)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes_K \boldsymbol{\varrho}^{(N_p,N_s,t)^{>k}}_{d-k,(N,J)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}},\end{aligned} \tag{3.146}$$

is a symmetry-adapted basis of $\boldsymbol{\mathcal{F}}_d$, where $(N,J)^{\leq k}, (N_p,N_s,t)^{\leq k}$, and $(J,M)^{\leq k}$ refer to labels $N^{\leq k}, N_p^{\leq k}, N_s^{\leq k}, t^{\leq k}$ resp. $J^{\leq k}, M^{\leq k}$.

$C^{JM}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}}$ are the so-called Clebsch-Gordan coefficients [53, 117] which are nonzero if:

$$\begin{aligned}\left|J^{\leq k} - J^{>k}\right| \leq J \leq J^{\leq k} + J^{>k}, \\ M^{\leq k} + M^{>k} = M,\end{aligned} \tag{3.147}$$

and $\chi^{N,N_p,N_s,J,t}_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}}$ are chosen to be nonzero if:

$$N_p^{\leq k} + N_p^{>k} = N_p, \;\; N_s^{\leq k} + N_s^{>k} \;\; = N_s, \;\; N^{\leq k} + N^{>k} = 2N_p + N_s = N. \tag{3.148}$$

Proof. According to Proposition 3.3.6, $\boldsymbol{\mathcal{F}}_k$ and $\boldsymbol{\mathcal{F}}_{d-k}$ decompose as follows:

$$\boldsymbol{\mathcal{F}}_k = \bigoplus_{\substack{N^{\leq k},N_s^{\leq k},N_p^{\leq k} \\ 2N_p^{\leq k}+N_s^{\leq k}=N^{\leq k} \\ J^{\leq k}}} \left(\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes \left(\bigoplus_{M^{\leq k}} \boldsymbol{\mathcal{S}}_{k,(J,M)^{\leq k}} \right) \right), \tag{3.149}$$

where $(N,J)^{\leq k}, (N_p,N_s)^{\leq k}$, and $(J,M)^{\leq k}$ refer to labels $N^{\leq k}, N_p^{\leq k}, N_s^{\leq k}$ resp. $J^{\leq k}, M^{\leq k}$ and

$$\boldsymbol{\mathcal{F}}_{d-k} = \bigoplus_{\substack{N^{>k},N_s^{>k},N_p^{>k} \\ 2N_p^{>k}+N_s^{>k}=N^{>k} \\ J^{>k}}} \left(\mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}} \otimes \left(\bigoplus_{M^{>k}} \boldsymbol{\mathcal{S}}_{d-k,(J,M)^{>k}} \right) \right). \tag{3.150}$$

Now, let $\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}$ be an orthonormal symmetry-adapted basis element of the full Fock space $\boldsymbol{\mathcal{F}}_d$. By noting that

$$\begin{aligned}\mathbf{I} = \sum_{\substack{N^{\leq k},N_s^{\leq k},N_p^{\leq k},J^{\leq k},M^{\leq k},t^{\leq k}\\ N^{>k},N_s^{>k},N_p^{>k},J^{>k},M^{>k},t^{>k}}} & (\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}) \\ & (\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}})^*. \end{aligned} \tag{3.151}$$

The latter is the so-called *completeness relation*. It follows:

$$\begin{aligned}&\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M} = \mathbf{I}\left(\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}\right) \\ &= \sum_{\substack{N^{\leq k},N_s^{\leq k},N_p^{\leq k},J^{\leq k},M^{\leq k},t^{\leq k}\\ N^{>k},N_s^{>k},N_p^{>k},J^{>k},M^{>k},t^{>k}}} (\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}) \\ &(\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}})^* \left(\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}\right) \\ &= \sum_{\substack{N^{\leq k},N_s^{\leq k},N_p^{\leq k},J^{\leq k},M^{\leq k},t^{\leq k}\\ N^{>k},N_s^{>k},N_p^{>k},J^{>k},M^{>k},t^{>k}}} \left(\chi_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}}^{N,N_p,N_s,J,t} C_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}}^{JM}\right) \boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \\ &\boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}, \end{aligned} \tag{3.152}$$

where $\chi_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}}^{N,N_p,N_s,J,t}$ and $C_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}}^{JM}$ are scalar values defined as:

$$\chi_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}}^{N,N_p,N_s,J,t} = (\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}})^* \boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t}, \tag{3.153}$$

and

$$C_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}}^{JM} = (\boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}})^* \boldsymbol{s}_{d,J,M}, \tag{3.154}$$

Notably, (3.152) can be interpreted as the basis change between $\boldsymbol{\varrho}_{d,N,J}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,J,M}$ and $\boldsymbol{\varrho}_{k,(N,J)^{\leq k}}^{(N_p,N_s,t)^{\leq k}} \otimes_K \boldsymbol{\varrho}_{d-k,(N,J)^{>k}}^{(N_p,N_s,t)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}$.

Using that for finite-dimensional spaces $\mathcal{V}_i, \tilde{\mathcal{V}}_i$, $\mathcal{U}_i$, and $\tilde{\mathcal{U}}_i$, there exists a canonical isomorphism between the following spaces:

$$\left(\bigoplus_i (\mathcal{V}_i \otimes \tilde{\mathcal{V}}_i)\right) \otimes \left(\bigoplus_j (\mathcal{U}_j \otimes \tilde{\mathcal{U}}_j)\right) \cong \bigoplus_{ij} \left(\mathcal{V}_i \otimes \tilde{\mathcal{V}}_i \otimes \mathcal{U}_j \otimes \tilde{\mathcal{U}}_j\right) \cong \bigoplus_{ij} \left(\mathcal{V}_i \otimes \mathcal{U}_j \otimes \tilde{\mathcal{V}}_i \otimes \tilde{\mathcal{U}}_j\right), \tag{3.155}$$

we see that $\boldsymbol{\mathcal{F}}_d$ is isomorphic to the decomposition:

$$\bigoplus_{\substack{N^{\leq k},N_p^{\leq k},N_s^{\leq k},J^{\leq k},M^{\leq k}\\ N^{>k},N_p^{>k},N_s^{>k},J^{>k},M^{>k}\\ 2N_p^{\leq k}+N_s^{\leq k}=N^{\leq k}\\ 2N_p^{>k}+N_s^{>k}=N^{>k}}} \left(\mathscr{P}_{k,(N,J)^{\leq k}}^{(N_p,N_s)^{\leq k}} \otimes \mathscr{P}_{k,(N,J)^{>k}}^{(N_p,N_s)^{>k}}\right) \otimes \left(\boldsymbol{\mathcal{S}}_{k,(J,M)^{\leq k}} \otimes \boldsymbol{\mathcal{S}}_{k,(J,M)^{>k}}\right). \tag{3.156}$$

□

Remark 3.3.7. One can verify from the definitions of $\dim(\mathscr{P}_{k,(N,J)^{\leq k}}^{(N_p,N_s)^{\leq k}})$ and $\dim(\mathscr{P}_{d-k,(N,J)^{>k}}^{(N_p,N_s)^{>k}})$ that, at fixed k, we have:

$$\dim(\mathscr{P}_{d,N,J}^{N_p,N_s}) = \sum_{\substack{N^{\leq k},N_s^{\leq k},N_p^{\leq k},J^{\leq k}\\ N^{>k},N_s^{>k},N_p^{>k},J^{>k}\\ \left|J^{\leq k}-J^{>k}\right|\leq J\leq J^{\leq k}+J^{>k}\\ 2(N_p^{\leq k}+N_p^{>k})+N_s^{\leq k}+N_s^{>k}=N}} \dim(\mathscr{P}_{k,(N,J)^{\leq k}}^{(N_p,N_s)^{\leq k}}) \dim(\mathscr{P}_{d-k,(N,J)^{>k}}^{(N_p,N_s)^{>k}}). \tag{3.157}$$

To provide context for the concepts introduced, we give the following simplified example.

Example 3.3.3. Let $d = 4$, $N = 4$, $N_s = 4$, $N_p = 0$, $J = 0$ and $M = 0$, in the canonical basis, the wavefunction can be expanded as follows:

$$\boldsymbol{\Psi} = \Psi_{\uparrow\uparrow\downarrow\downarrow}\boldsymbol{e}_{\uparrow\uparrow\downarrow\downarrow} + \Psi_{\downarrow\downarrow\uparrow\uparrow}\boldsymbol{e}_{\downarrow\downarrow\uparrow\uparrow} + \Psi_{\uparrow\downarrow\uparrow\downarrow}\boldsymbol{e}_{\uparrow\downarrow\uparrow\downarrow} + \Psi_{\downarrow\uparrow\downarrow\uparrow}\boldsymbol{e}_{\downarrow\uparrow\downarrow\uparrow} + \Psi_{\uparrow\downarrow\downarrow\uparrow}\boldsymbol{e}_{\uparrow\downarrow\downarrow\uparrow} + \Psi_{\downarrow\uparrow\uparrow\downarrow}\boldsymbol{e}_{\downarrow\uparrow\uparrow\downarrow}. \tag{3.158}$$

By the normalization of $\boldsymbol{\Psi}$, we have:

$$\Psi^2_{\uparrow\uparrow\downarrow\downarrow} + \Psi^2_{\downarrow\downarrow\uparrow\uparrow} + \Psi^2_{\uparrow\downarrow\uparrow\downarrow} + \Psi^2_{\downarrow\uparrow\downarrow\uparrow} + \Psi^2_{\uparrow\downarrow\downarrow\uparrow} + \Psi^2_{\downarrow\uparrow\uparrow\downarrow} = 1. \tag{3.159}$$

The question that arises is how to choose the coefficients in order for $\boldsymbol{\Psi}$ to be an eigenvector of $\mathbf{N}$, $\mathbf{S}^z$, and $\hat{S}^2$ with $J = 0$, $M = 0$ and $N = 4$. This involves finding the correct linear combination within the canonical basis. To achieve this, one must work within the symmetry-adapted basis first, as discussed previously. This results in the subsequent expansion of $\boldsymbol{\Psi}$. According to Corollary 3.3.7, when $N = 4$, $N_p = 0$, $N_s = 4$, $J = 0$, and $M = 0$ are fixed, we obtain:

$$\begin{aligned}\boldsymbol{\Psi} &= \sum_{t=1}^{\dim(\mathscr{P}^{0,4}_{4,4,0})} \Psi^{0,4,t}_{4,4,0,0}\boldsymbol{\xi}^{0,4,t}_{4,4,0,0} = \sum_{t=1}^{2} \Psi^{0,4,t}_{4,4,0,0}\boldsymbol{\xi}^{0,4,t}_{4,4,0,0}, \\ &= \Psi^{0,4,1}_{4,4,0,0}\boldsymbol{\xi}^{0,4,1}_{4,4,0,0} + \Psi^{0,4,2}_{4,4,0,0}\boldsymbol{\xi}^{0,4,2}_{4,4,0,0},\end{aligned} \tag{3.160}$$

where $\dim(\mathscr{P}^{0,4}_{4,4,0}) = 2$ according to (3.135). Now, for $k = 2$, according to Proposition 3.3.8, we have

Parameter	Value
$N^{\leq 2}$	2
$N_s^{\leq 2}$	2
$N_p^{\leq 2}$	0
$N^{>2}$	2
$N_s^{>2}$	2
$N_p^{>2}$	0
$J^{\leq 2} = J^{>2}$	$\{0, 1\}$
$M^{\leq 2}$	$\{-J^{\leq 2}, \ldots, J^{\leq 2}\}$
$M^{>2}$	$\{-J^{>2}, \ldots, J^{>2}\}$
$t^{\leq 2}, t^{>2}$	1

where $J^{\leq 2} = J^{>2}$, given $J = 0$. It follows:

$$\begin{aligned}\boldsymbol{\xi}^{0,4,1}_{4,4,0,0} &\equiv \underbrace{C^{0,0}_{0,0,0,0}}_{=1}\boldsymbol{\varrho}^{0,2,1}_{2,2,0} \otimes_K \boldsymbol{\varrho}^{0,2,1}_{2,2,0} \otimes_K \boldsymbol{s}_{2,0,0} \otimes_K \boldsymbol{s}_{2,0,0}, \\ \boldsymbol{\xi}^{0,4,2}_{4,4,0,0} &\equiv \underbrace{C^{0,0}_{1,1,-1,1}}_{=\frac{1}{\sqrt{3}}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,-1} \otimes_K \boldsymbol{s}_{2,1,1} + \underbrace{C^{0,0}_{1,1,1,-1}}_{=\frac{1}{\sqrt{3}}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,1} \otimes_K \boldsymbol{s}_{2,1,-1} \\ &+ \underbrace{C^{0,0}_{1,1,0,0}}_{=-\frac{1}{\sqrt{3}}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,0} \otimes_K \boldsymbol{s}_{2,1,0}.\end{aligned} \tag{3.161}$$

The Clebsch-Gordan coefficients can be obtained from [53]. By replacing (3.161) in (3.160), one writes:

$$\begin{aligned}\boldsymbol{\Psi} \equiv \Psi^{0,4,1}_{4,4,0,0}(&\boldsymbol{\varrho}^{0,2,1}_{2,2,0} \otimes_K \boldsymbol{\varrho}^{0,2,1}_{2,2,0} \otimes_K \boldsymbol{s}_{2,0,0} \otimes_K \boldsymbol{s}_{2,0,0}) + \Psi^{0,4,2}_{4,4,0,0}(\frac{1}{\sqrt{3}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,-1} \otimes_K \boldsymbol{s}_{2,1,1} \\ &+ \frac{1}{\sqrt{3}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,1} \otimes_K \boldsymbol{s}_{2,1,-1} - \frac{1}{\sqrt{3}}\boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{\varrho}^{0,2,2}_{2,2,1} \otimes_K \boldsymbol{s}_{2,1,0} \otimes_K \boldsymbol{s}_{2,1,0}).\end{aligned} \tag{3.162}$$

According to Definition 30, we obtain:

$$\begin{aligned}\boldsymbol{\Psi} \equiv \Psi^{0,4,1}_{4,4,0,0}(&\boldsymbol{\xi}^{0,2,1}_{2,2,0,0} \otimes_K \boldsymbol{\xi}^{0,2,1}_{2,2,0,0}) + \Psi^{0,4,2}_{4,4,0,0}(\frac{1}{\sqrt{3}}\boldsymbol{\xi}^{0,2,2}_{2,2,1,-1} \otimes_K \boldsymbol{\xi}^{0,2,2}_{2,2,1,1} \\ &+ \frac{1}{\sqrt{3}}\boldsymbol{\xi}^{0,2,2}_{2,2,1,1} \otimes_K \boldsymbol{\xi}^{0,2,2}_{2,2,1,-1} - \frac{1}{\sqrt{3}}\boldsymbol{\xi}^{0,2,2}_{2,2,1,0} \otimes_K \boldsymbol{\xi}^{0,2,2}_{2,2,1,0}),\end{aligned} \tag{3.163}$$

we also have:

$$\boldsymbol{\xi}^{0,2,1}_{2,2,0,0} = \frac{(\boldsymbol{e}_{\uparrow\downarrow} - \boldsymbol{e}_{\downarrow\uparrow})}{\sqrt{2}},\ \boldsymbol{\xi}^{0,2,2}_{2,2,1,0} = \frac{(\boldsymbol{e}_{\uparrow\downarrow} + \boldsymbol{e}_{\downarrow\uparrow})}{\sqrt{2}},\ \boldsymbol{\xi}^{0,2,2}_{2,2,1,1} = \boldsymbol{e}_{\uparrow\uparrow},\ \boldsymbol{\xi}^{0,2,2}_{2,2,1,-1} = \boldsymbol{e}_{\downarrow\downarrow}. \tag{3.164}$$

Therefore, by using the expression of (3.164) in (3.162), one ends up with:

$$\begin{aligned}\boldsymbol{\Psi} = &\frac{\Psi^{0,4,1}_{4,4,0,0}}{2}\left[\boldsymbol{e}_{\uparrow\downarrow\uparrow\downarrow} - \boldsymbol{e}_{\uparrow\downarrow\downarrow\uparrow} - \boldsymbol{e}_{\downarrow\uparrow\uparrow\downarrow} + \boldsymbol{e}_{\downarrow\uparrow\downarrow\uparrow}\right] \\ &+ \frac{\Psi^{0,4,2}_{4,4,0,0}}{\sqrt{3}}\left[\boldsymbol{e}_{\uparrow\uparrow\downarrow\downarrow} + \boldsymbol{e}_{\downarrow\downarrow\uparrow\uparrow} - \frac{1}{2}\left(\boldsymbol{e}_{\uparrow\downarrow\uparrow\downarrow} + \boldsymbol{e}_{\uparrow\downarrow\downarrow\uparrow} + \boldsymbol{e}_{\downarrow\uparrow\uparrow\downarrow} + \boldsymbol{e}_{\downarrow\uparrow\downarrow\uparrow}\right)\right].\end{aligned} \tag{3.165}$$

We deduce that the coefficients in the linear combination of (3.158) must be chosen such that

$$\begin{aligned}\Psi_{\uparrow\uparrow\downarrow\downarrow} = \Psi_{\downarrow\downarrow\uparrow\uparrow} &= \frac{\Psi^{0,4,2}_{4,4,0,0}}{\sqrt{3}}, \\ \Psi_{\uparrow\downarrow\uparrow\downarrow} = \Psi_{\downarrow\uparrow\downarrow\uparrow} &= \left(\frac{\Psi^{0,4,1}_{4,4,0,0}}{2} - \frac{\Psi^{0,4,2}_{4,4,0,0}}{2\sqrt{3}}\right), \\ \Psi_{\uparrow\downarrow\downarrow\uparrow} = \Psi_{\downarrow\uparrow\uparrow\downarrow} &= -\left(\frac{\Psi^{0,4,1}_{4,4,0,0}}{2} + \frac{\Psi^{0,4,2}_{4,4,0,0}}{2\sqrt{3}}\right).\end{aligned} \tag{3.166}$$

with $(\Psi^{0,4,2}_{4,4,0,0})^2 + (\Psi^{0,4,1}_{4,4,0,0})^2 = 1$.

TT representation of a singlet state

As defined in Equation (1.140), let $(\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_d)$ be the TT representation of the d-order tensor $\psi \in \mathbb{R}^{4\times\cdots\times 4}$, which represents the tensor folding of $\boldsymbol{\Psi} \in \mathbb{R}^{4^d}$, with $\mathcal{U}_k \in \mathbb{R}^{r_{k-1}\times 4\times k}$, $k \in [d]$. We have for $\mu_k \in \{0, \uparrow, \downarrow, 2\}$:

$$\begin{aligned}\psi(\mu_1; \mu_2; \ldots; \mu_d) &= \mathbf{U}_1[\mu_1]\mathbf{U}_2[\mu_2]\cdots\mathbf{U}_d[\mu_d], \\ &= \sum_{\alpha_1=1}^{r_1}\sum_{\alpha_2=1}^{r_2}\cdots\sum_{\alpha_{d-1}=1}^{r_{d-1}} \mathcal{U}_1(1;\mu_1;\alpha_1)\mathcal{U}_2(\alpha_1;\mu_2;\alpha_2)\cdots\mathcal{U}_d(\alpha_{d-1};\mu_d;1),\end{aligned} \tag{3.167}$$

where $\mathbf{U}_k[\mu_k] := \mathcal{U}_k[:, \mu_k, :] \in \mathbb{R}^{r_{k-1}\times r_k}$ and $r_0 = r_d = 1$. As outlined in Definition 20, we recall that a TT-core $\mathcal{U}_k \in \mathbb{R}^{r_{k-1}\times 4\times r_k}$ is said to be left-orthogonal if

$$\sum_{\mu_k} \mathbf{U}_k[\mu_k]^* \mathbf{U}_k[\mu_k] = \mathbf{I}_{r_k},$$

and right-orthogonal if

$$\sum_{\mu_k} \mathbf{U}_k[\mu_k] \mathbf{U}_k[\mu_k]^* = \mathbf{I}_{r_{k-1}}.$$

Moving forward, we assume that for all $k \in [d]$, there is a TT representation $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ such that all TT-cores $(\mathcal{U}_1, \ldots, \mathcal{U}_{k-1})$ are left-orthogonal and $(\mathcal{U}_{k+1}, \ldots, \mathcal{U}_d)$ are right-orthogonal.

Remark 3.3.8. We introduce the following notation regarding a matrix block, which will be encountered in subsequent discussions. Let $\boldsymbol{\mathcal{V}}$ be a finite-dimensional vector space that writes $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{U}}_1 \bigoplus \boldsymbol{\mathcal{U}}_2$, where $\boldsymbol{\mathcal{U}}_1$ and $\boldsymbol{\mathcal{U}}_2$ are subspaces with fixed basis. Let $\mathbf{A}$ be the matrix representation of the linear operator, denoted by $\hat{A} : \boldsymbol{\mathcal{V}} \to \boldsymbol{\mathcal{V}}$, with respect to this basis. The matrix has a block format, and each block corresponds to the restriction of $\mathbf{A}$ to the subspace $\boldsymbol{\mathcal{U}}_i \times \boldsymbol{\mathcal{U}}_j$, where $i, j \in \{1, 2\}$ and is denoted by:

$$\mathbf{A}\Big|_{\boldsymbol{\mathcal{U}}_i\times\boldsymbol{\mathcal{U}}_j}. \tag{3.168}$$

Henceforth, for simplification, our focus shifts to the singlet state. The latter is characterized by $J = 0$ and $M = 0$ with fixed particle number N, denoted by $\boldsymbol{\Psi}^{N,0,0}$. Our goal lies in deriving the underlying structure of the TT representation for a singlet state.

Theorem 5. Let $\mathbf{\Psi}^{N,0,0} \in \boldsymbol{\mathcal{F}}_d$ such that

$$\mathbf{N}\mathbf{\Psi}^{N,0,0} = N\mathbf{\Psi}^{N,0,0}, \quad \hat{S}^2\mathbf{\Psi}^{N,0,0} = 0, \quad \text{and} \quad \mathbf{S}^z\mathbf{\Psi}^{N,0,0} = 0. \tag{3.169}$$

Let $(\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_d)$ be the TT representation of $\mathbf{\Psi}^{N,0,0}$. For all $k \in [d-1]$, there are subspaces $\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}$ and $\tilde{\mathcal{S}}_{k,(J,M)^{\leq k}}$ such that

$$\mathbb{R}^{r_k} \cong \bigoplus_{N^{\leq k}, J^{\leq k}, M^{\leq k}} \tilde{\mathscr{P}}_{k,(N,J)^{\leq k}} \otimes \tilde{\mathcal{S}}_{k,(J,M)^{\leq k}}, \tag{3.170}$$

with $N^{\leq k} \in [2k]$, $J^{\leq k} \in \left\{0, \frac{1}{2}, 1, \ldots, \frac{k}{2}\right\}$, and $M^{\leq k} \in \left\{-J^{\leq k}, \ldots, J^{\leq k}\right\}$. Additionally, we have $\dim(\tilde{\mathcal{S}}_{k,(J,M)^{\leq k}}) = 1$ and $\sum_{N^{\leq k}, J^{\leq k}} (2J^{\leq k}+1)\dim(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}) = r_k$.

Moreover, if the TT-cores are such that $(\mathcal{U}_1, \ldots, \mathcal{U}_{k-1})$ are left-orthogonal and $(\mathcal{U}_{k+1}, \ldots, \mathcal{U}_d)$ are right-orthogonal, see Definition 20, then there are orthogonal matrices $\mathbf{G}_{k-1} \in \mathbb{R}^{r_{k-1}\times r_{k-1}}$, $\mathbf{G}_k \in \mathbb{R}^{r_k \times r_k}$ and matrices $\tilde{\mathcal{U}}_k \in \mathbb{R}^{\tilde{r}_{k-1}\times 3 \times \tilde{r}_k}$, with $\tilde{r}_k = \sum_{N^{\leq k}, J^{\leq k}} \dim(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}})$, such that by denoting $\tilde{\mathbf{U}}_k[N(\mu_k)] := \tilde{\mathcal{U}}_k[:, N(\mu_k), :]$, we have (see Figure 3.9):

$$\begin{aligned}(\mathbf{G}_{k-1}^* \mathbf{U}_k[\mu_k]\mathbf{G}_k)\Big|_{\left(\tilde{\mathscr{P}}_{k-1,(N,J)^{\leq k-1}} \otimes \tilde{\mathcal{S}}_{k-1,(J,M)^{\leq k-1}}\right) \times \left(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}} \otimes \tilde{\mathcal{S}}_{k,(J,M)^{\leq k}}\right)} \\ = (\tilde{\mathbf{U}}_k[N(\mu_k)])\Big|_{\tilde{\mathscr{P}}_{k-1,(N,J)^{\leq k-1}} \times \tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}} \begin{pmatrix} J^{\leq k-1} & J(\mu_k) & J^{>k} \\ M^{\leq k-1} & M(\mu_k) & M^{>k} \end{pmatrix},\end{aligned} \tag{3.171}$$

where given that $\mathbf{\Psi}^{N,0,0}$ is a singlet state, i.e it satisfies Equation (3.169), we have:

$$J^{\leq k} = J^{>k}, \quad M^{\leq k-1} + M(\mu_k) = M^{\leq k} = -M^{>k} \text{ and } N^{\leq k-1} + N(\mu_k) = N^{\leq k} = N - N^{>k}. \tag{3.172}$$

$\begin{pmatrix} J^{\leq k-1} & J(\mu_k) & J^{>k} \\ M^{\leq k-1} & M(\mu_k) & M^{>k} \end{pmatrix}$ are the 3-j symbols [11] which can be given in terms of the Clebsch–Gordan coefficients as follows:

$$\begin{pmatrix} J^{\leq k-1} & J(\mu_k) & J^{>k} \\ M^{\leq k-1} & M(\mu_k) & M^{>k} \end{pmatrix} := \frac{(-1)^{J^{\leq k-1} - J(\mu_k) - M^{>k}}}{\sqrt{2J^{>k}+1}} C^{J^{>k}, -M^{>k}}_{J^{\leq k-1}, J(\mu_k), M^{\leq k-1}, M(\mu_k)}. \tag{3.173}$$

The diagrammatic representation of the TT-cores according to the Theorem 3.154 can be given as:

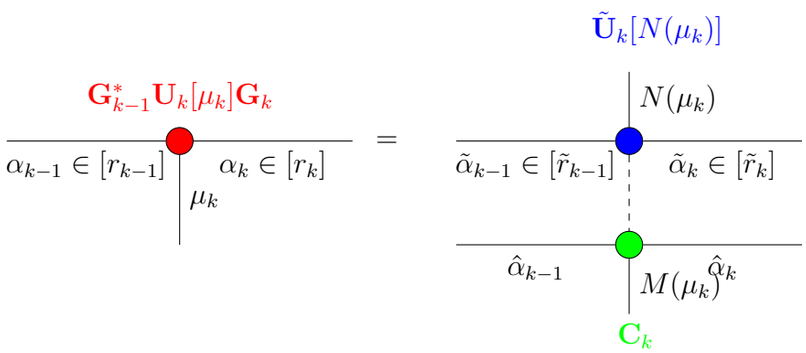


Figure 3.9: Tensor diagram depicting the TT-cores of the TT representations of eigenfunctions, satisfying eigenvalue relations (3.169), for each $\mu_k \in \{0, \uparrow, \downarrow, 2\}$. Here, $\hat{\alpha}_{k-1} \in \left[\sum_{J^{\leq k-1}} (2J^{\leq k-1}+1)\right]$ and $\hat{\alpha}_k \in \left[\sum_{J^{\leq k}} (2J^{\leq k}+1)\right]$. The matrix $\mathbf{C}_k \in \mathbb{R}^{\left(\sum_{J^{\leq k-1}} (2J^{\leq k-1}+1)\right) \times \left(\sum_{J^{\leq k}} (2J^{\leq k}+1)\right)}$ comprises 3-j symbols from (3.173). The dashed line denotes the block-wise Kronecker product between the block matrices in $\tilde{\mathbf{U}}_k[N(\mu_k)]$ and $\mathbf{C}_k$, sharing the same indices $J^{\leq k}$ and $J^{\leq k-1}$.

Now, to prove this theorem, we first look at the structure of the mode-$(1:k)$ matricization of $\psi^{N,0,0}$, the tensor folding of $\mathbf{\Psi}^{N,0,0}$, denoted as $(\mathbf{\Psi}^{N,0,0})^{<k>}$.

Proposition 3.3.9. *Let* $\mathbf{\Psi}^{N,0,0} \in \boldsymbol{\mathcal{F}}_d$ *such that*

$$\mathbf{N}\mathbf{\Psi}^{N,0,0} = N\mathbf{\Psi}, \quad \hat{S}^2\mathbf{\Psi}^{N,0,0} = 0, \quad \textit{and} \quad \mathbf{S}^z\mathbf{\Psi}^{N,0,0} = 0.$$

For all $k \in [d-1]$, *there exist orthogonal matrices* $\mathbf{Q}_L$ *and* $\mathbf{Q}_R$ *and matrices* $\mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}$ *of size* $\dim(\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}}) \times \dim(\mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}})$ *such that*

$$\Big(\mathbf{Q}_L\Big((\mathbf{\Psi}^{N,0,0})^{<k>}\Big)\mathbf{Q}_R\Big)\Big|_{\left(\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes \mathcal{S}_{k,(J,M)^{\leq k}}\right)\times\left(\mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}} \otimes \mathcal{S}_{d-k,(J,M)^{>k}}\right)} = \mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}. \tag{3.174}$$

The matrix $\mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}$ *is nonzero only if* $2(N_p^{>k} + N_p^{\leq k}) + N_s^{\leq k} + N_s^{>k} = N$ *and* $J^{\leq k} = J^{>k}$, *with* $\frac{mod(N_s^{\leq k},2)}{2} \leq J^{\leq k} \leq \frac{N_s^{\leq k}}{2}$, $\frac{mod(N_s^{>k},2)}{2} \leq J^{>k} \leq \frac{N_s^{>k}}{2}$.

Proof. According to Corollary (3.3.7) and Proposition 3.3.8, $\mathbf{\Psi}^{N,0,0}$ can be expanded as follows, for a fixed $k \in [d]$:

$$\mathbf{\Psi}^{N,0,0} \equiv \sum_{t,N_p,N_s} \Psi^{N_p,N_s,t}_{N,0,0} \left(\sum_{\substack{N^{\leq k},N_p^{\leq k},N_s^{\leq k},J^{\leq k} \\ N^{>k},N_p^{>k},N_s^{>k},J^{>k} \\ t^{\leq k},t^{>k}}} \chi^{N,N_p,N_s,0,t}_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}} \right.$$
$$\left. \left(\sum_{M^{\leq k}M^{>k}} C^{0,0}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}} (\boldsymbol{\varrho}^{(N_p,N_s,t)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes_K \boldsymbol{\varrho}^{(N_p,N_s,t)^{>k}}_{d-k,(N,J)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}) \right) \right), \tag{3.175}$$

$$\mathbf{\Psi}^{N,0,0} \equiv \left(\sum_{\substack{N^{\leq k},N_p^{\leq k},N_s^{\leq k},J^{\leq k} \\ N^{>k},N_p^{>k},N_s^{>k},J^{>k} \\ t^{\leq k},t^{>k}}} \Psi^{N_p,N_s,t}_{N,0,0} \chi^{N,N_p,N_s,0,t}_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}} \delta_{N^{\leq k}+N^{>k},N} \delta_{N_p^{\leq k}+N_p^{>k},N_p} \delta_{N_s^{\leq k}+N_s^{>k},N_s} \right.$$
$$\left. \left(\sum_{M^{\leq k}M^{>k}} C^{0,0}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}} (\boldsymbol{\varrho}^{(N_p,N_s,t)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes_K \boldsymbol{\varrho}^{(N_p,N_s,t)^{>k}}_{d-k,(N,J)^{>k}} \otimes_K \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}}) \right) \right), \tag{3.176}$$

where

$$N = N^{\leq k} + N^{>k}, N_p = N_p^{\leq k} + N_p^{>k}, N_s = N_s^{\leq k} + N_s^{>k}, \tag{3.177}$$

and $t \in \left[\dim(\mathscr{P}^{N_p,N_s}_{d,N,J})\right]$ refers to each couple $(t^{\leq k}, t^{>k})$ with ranges satisfying the condition elaborated in (3.157). The coefficients $C^{00}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}}$ are given by , see [53]:

$$C^{00}_{J^{\leq k}J^{>k}M^{\leq k}M^{>k}} = \delta_{J^{\leq k},J^{>k}} \delta_{M^{\leq k},-M^{>k}} \frac{(-1)^{J^{\leq k}-M^{\leq k}}}{\sqrt{2J^{\leq k}+1}}. \tag{3.178}$$

We denote the entries of the vector $\mathbf{\Psi}^{N,0,0}$ as $\left(\mathbf{\Psi}^{N,0,0}\right)^{(N,N_p,N_s,J,M,t)^{>k}}_{(N,N_p,N_s,J,M,t)^{\leq k}}$. By combining the Equations (3.178) and (3.176), these entries can be defined as:

$$\left(\mathbf{\Psi}^{N,0,0}\right)^{(N,N_p,N_s,J,M,t)^{>k}}_{(N,N_p,N_s,J,M,t)^{\leq k}} = \underbrace{\Psi^{N_p,N_s,t}_{N,0,0} \chi^{N,N_p,N_s,0,t}_{(N,N_p,N_s,J,t)^{\leq k},(N,N_p,N_s,J,t)^{>k}} \delta_{J^{\leq k},J^{>k}} \frac{(-1)^{J^{\leq k}}}{\sqrt{2J^{\leq k}+1}}}_{\theta^{(N,N_p,N_s,J,t)^{>k}}_{(N,N_p,N_s,J,t)^{\leq k}}} \delta_{M^{\leq k},-M^{>k}} (-1)^{-M^{\leq k}}. \tag{3.179}$$

By introducing left and right orthogonal matrices that serve as the change of basis from the canonical basis to the symmetry-adapted basis, the mode-$(1:k)$ matricization of

$\psi^{N,0,0}$, the tensor folding of $\mathbf{\Psi}^{N,0,0}$, writes:

$$\mathbf{Q}_L((\mathbf{\Psi}^{N,0,0})^{<k>})\mathbf{Q}_R = \left(\sum_{\substack{N^{\leq k},N_p^{\leq k},N_s^{\leq k},J^{\leq k}\\ N^{>k},N_p^{>k},N_s^{>k},J^{>k}\ t^{\leq k},t^{>k}}} \theta^{(N,N_p,N_s,J,t)^{>k}}_{(N,N_p,N_s,J,t)^{\leq k}} \boldsymbol{\varrho}^{(N_p,N_s,t)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes_K \right. \\ \left. \boldsymbol{\varrho}^{(N_p,N_s,t)^{>k}*}_{d-k,(N,J)^{>k}}\right)\left(\sum_{M^{\leq k}=-J^{\leq k}}^{J^{\leq k}} \boldsymbol{s}_{k,(J,M)^{\leq k}} \otimes_K \boldsymbol{s}^*_{k,(J,M)^{\leq k}}\right). \tag{3.180}$$

Let $\mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}$ of size $\dim(\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}}) \times \dim(\mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}})$ denote matrices with entries $\theta^{(N,N_p,N_s,J,t)^{>k}}_{(N,N_p,N_s,J,t)^{\leq k}}$, where

$$\Big(\mathbf{Q}_L\Big((\mathbf{\Psi}^{N,0,0})^{<k>}\Big)\mathbf{Q}_R\Big)\Big|_{\left(\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}} \otimes \mathcal{S}_{k,(J,M)^{\leq k}}\right)\times\left(\mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}} \otimes \mathcal{S}_{d-k,(J,M)^{>k}}\right)} = \mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}. \tag{3.181}$$

where

$$\mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}} \neq \mathbf{0}, \text{ if } 2(N_p^{\leq k}+N_p^{>k})+N_s^{\leq k}+N_s^{>k} = N^{\leq k}+N^{>k} = N, \text{and } J^{\leq k} = J^{>k}. \tag{3.182}$$

□

Alternatively, one can write:

$$\Big(\mathbf{Q}_L\Big((\mathbf{\Psi}^{N,0,0})^{<k>}\Big)\mathbf{Q}_R\Big) = \mathrm{diag}(\mathbf{\Theta}^{(N,J)^{>k}}_{(N,J)^{\leq k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{>k}+1)}), \tag{3.183}$$

with

$$\left(\mathbf{\Theta}^{(N,J)^{>k}}_{(N,J)^{\leq k}}\right)\Big|_{\mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}} \times \mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}}} = \mathbf{\Theta}^{(N,N_p,N_s,J)^{>k}}_{(N,N_p,N_s,J)^{\leq k}}. \tag{3.184}$$

Moving forward, we introduce the spaces $\mathscr{P}_{k,(N,J)^{\leq k}}$, $\mathscr{P}_{d-k,(N,J)^{>k}}$, defined as:

$$\mathscr{P}_{k,(N,J)^{\leq k}} = \bigoplus_{\substack{N_p^{\leq k},N_s^{\leq k},2N_p^{\leq k}+N_s^{\leq k}=N^{\leq k},\\ N_s^{\leq k}\in\left\{2J^{\leq k},2J^{\leq k}+2,\ldots,k-\mathrm{mod}(2J^{\leq k},2)\right\}}} \mathscr{P}^{(N_p,N_s)^{\leq k}}_{k,(N,J)^{\leq k}}, \tag{3.185}$$

and

$$\mathscr{P}_{d-k,(N,J)^{>k}} = \bigoplus_{\substack{N_p^{>k},N_s^{>k},2N_p^{>k}+N_s^{\leq k}=N^{>k},\\ N_s^{>k}\in\left\{2J^{>k},2J^{>k}+2,\ldots,(d-k)-\mathrm{mod}(2J^{>k},2)\right\}}} \mathscr{P}^{(N_p,N_s)^{>k}}_{d-k,(N,J)^{>k}}, \tag{3.186}$$

such that $\mathbf{\Theta}^{(N,J)^{>k}}_{(N,J)^{\leq k}}$ is of size $\dim(\mathscr{P}_{k,(N,J)^{\leq k}}) \times \dim(\mathscr{P}_{d-k,(N,J)^{>k}})$.

Remark 3.3.9. The preceding proposition can be established by using Schur's lemma from the group representation theory [20]. One needs to establish commutation relations between between $(\mathbf{\Psi}^{N,0,0})^{<k>}$ at fixed k and the generators of the SU(2) symmetry group representation [112], notably operators defining $\hat{S}^2$, see (3.3), described on the basis of Fock spaces $\boldsymbol{\mathcal{F}}_k$ resp. $\boldsymbol{\mathcal{F}}_{d-k}$ such as $\left(\mathbf{S}^x\right)^{\leq k}$, $\left(\mathbf{S}^x\right)^{>k}$, $\left(\mathbf{S}^y\right)^{\leq k}$, $\left(\mathbf{S}^y\right)^{>k}$, and $\left(\mathbf{S}^z\right)^{\leq k}$, $\left(\mathbf{S}^z\right)^{>k}$, where (showing only the expression of $\mathbf{S}^z$, with a similar reasoning for $\mathbf{S}_x$ or $\mathbf{S}_y$), given the definition of $\mathbf{S}^z$

$$\mathbf{S}^z = \frac{1}{2}\sum_{i=1}^{d}\left(\mathbf{A}^*_{i,\frac{1}{2}}\mathbf{A}_{i,\frac{1}{2}} - \mathbf{A}^*_{i,\frac{-1}{2}}\mathbf{A}_{i,\frac{-1}{2}}\right), \tag{3.187}$$

where

$$\mathbf{A}^*_{is}\mathbf{A}_{i,s} = \left(\otimes_{l=1}^{i-1}\mathbf{I}\right)\otimes_K(\mathbf{A}^*_s\mathbf{A}_s)\otimes_K\left(\otimes_{l=i+1}^{d}\mathbf{I}\right), s\in\left\{\pm\frac{1}{2}\right\}. \tag{3.188}$$

It follows:

$$\mathbf{S}^z = \sum_{i=1}^{k}\left(\otimes_{l=1}^{i-1}\mathbf{I}\right)\otimes_K\left(\mathbf{A}^*_{\frac{1}{2}}\mathbf{A}_{\frac{1}{2}} - \mathbf{A}^*_{\frac{-1}{2}}\mathbf{A}_{\frac{-1}{2}}\right)\otimes_K\left(\otimes_{l=i+1}^{k}\mathbf{I}\right)\otimes_K\left(\otimes_{l=k+1}^{d}\mathbf{I}\right) \tag{3.189}$$

$$+\sum_{i=k+1}^{d}\left(\otimes_{l=1}^{k}\mathbf{I}\right)\left(\otimes_{l=k+1}^{i-1}\mathbf{I}\right)\otimes_K\left(\mathbf{A}^*_{\frac{1}{2}}\mathbf{A}_{\frac{1}{2}} - \mathbf{A}^*_{\frac{-1}{2}}\mathbf{A}_{\frac{-1}{2}}\right)\otimes_K\left(\otimes_{l=i+1}^{d}\mathbf{I}\right) \tag{3.190}$$

$$= \left(\mathbf{S}^z\right)^{\leq k}\otimes_K\left(\otimes_{l=k+1}^{d}\mathbf{I}\right) + \left(\otimes_{l=1}^{k}\mathbf{I}\right)\otimes_K\left(\mathbf{S}^z\right)^{>k}, \tag{3.191}$$

where

$$\left(\mathbf{S}^z\right)^{\leq k} = \sum_{i=1}^{k} \left(\otimes_{l=1}^{i-1}\mathbf{I}\right) \otimes_K \left(\mathbf{A}^*_{\frac{1}{2}}\mathbf{A}_{\frac{1}{2}} - \mathbf{A}^*_{\frac{-1}{2}}\mathbf{A}_{\frac{-1}{2}}\right) \otimes_K \left(\otimes_{l=i+1}^{k}\mathbf{I}\right), \tag{3.192}$$

$$\left(\mathbf{S}^z\right)^{>k} = \sum_{i=k+1}^{d} \left(\otimes_{l=k+1}^{i-1}\mathbf{I}\right) \otimes_K \left(\mathbf{A}^*_{\frac{1}{2}}\mathbf{A}_{\frac{1}{2}} - \mathbf{A}^*_{\frac{-1}{2}}\mathbf{A}_{\frac{-1}{2}}\right) \otimes_K \left(\otimes_{l=i+1}^{k}\mathbf{I}\right). \tag{3.193}$$

Thus, $\mathbf{S_z}\mathbf{\Psi}^{N,0,0} = \mathbf{0}$ can be written as $\mathbf{S_z}^{\leq k}\left(\mathbf{\Psi}^{N,0,0}\right)^{<k>} + \left(\mathbf{\Psi}^{N,0,0}\right)^{<k>}\mathbf{S_z}^{>k} = \mathbf{0}$. Using the Schur's lemma, we deduce that $\mathbf{\Psi}^{<k>}$ is diagonal in the basis diagonalizing $\mathbf{S_z}^{\leq k}$ and $\mathbf{S_z}^{>k}$.

Now, from the TT representation of $\mathbf{\Psi}$ (3.167), one writes:

$$\mathbf{\Psi} = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \sum_{\alpha_k=1}^{r_k} \mathbf{L}_{\leq k-1}[:,\alpha_{k-1}] \otimes_K \left(\sum_{\mu_k \in \{0,\uparrow,\downarrow,2\}} \mathcal{U}_k(\alpha_{k-1};\mu_k;\alpha_k)\boldsymbol{e}_{\mu_k} \right) \otimes_K \mathbf{R}_{>k}[\alpha_k,:], \tag{3.194}$$

where

$$\mathbf{L}_{\leq k-1}[:,\alpha_{k-1}] = \sum_{\mu_1} \dots \sum_{\mu_{k-1}} \left(\mathbf{U}_1[\mu_1]\dots\mathbf{U}_{k-1}[\mu_{k-1}]\right)(\overline{\mu_1,\dots,\mu_{k-1}};\alpha_{k-1})\boldsymbol{e}_{\mu_1\dots\mu_{k-1}}, \tag{3.195}$$

$$\mathbf{R}_{>k}[\alpha_k,:] = \sum_{\mu_{k+1}} \dots \sum_{\mu_d} \left(\mathbf{U}_{k+1}[\mu_{k+1}]\dots\mathbf{U}_d[\mu_d]\right)(\alpha_k;\overline{\mu_{k+1},\dots,\mu_d})\boldsymbol{e}_{\mu_{k+1}\dots\mu_d}. \tag{3.196}$$

We suppose here that the TT-cores in $(\mathcal{U}_1,\dots,\mathcal{U}_{k-1})$ are left-orthogonal and the TT-cores in $(\mathcal{U}_{k+1},\dots,\mathcal{U}_d)$ are right-orthogonal, for $k \in \{2,\dots,d-1\}$, if $k=1$, $(\mathcal{U}_2,\dots,\mathcal{U}_d)$ are right-orthogonal and if $k=d$, $(\mathcal{U}_1,\dots,\mathcal{U}_{d-1})$ are left-orthogonal. The matrix $\mathbf{L}_{\leq k-1}$ (resp. $\mathbf{R}_{>k}$) is the matrix representations of the operator $\hat{L}_{\leq k-1} : \mathbb{R}^{r_{k-1}} \to \boldsymbol{\mathcal{F}}_{k-1}$ (resp. $\hat{R}_{>k} : \boldsymbol{\mathcal{F}}_{d-k} \to \mathbb{R}^{r_k}$).

Proposition 3.3.10. *For all $k \in [d-1]$, there exist vector spaces $\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}$ and $\tilde{\boldsymbol{\mathcal{S}}}_{k,(J,M)^{\leq k}}$ such that $\mathbb{R}^{r_k}$ decomposes as:*

$$\mathbb{R}^{r_k} \cong \bigoplus_{N^{\leq k},J^{\leq k},M^{\leq k}} \left(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}} \otimes \tilde{\boldsymbol{\mathcal{S}}}_{k,(J,M)^{\leq k}}\right), \tag{3.197}$$

where $N^{\leq k} \in \{0,\dots,2k\}$, $J^{\leq k} \in \left\{0,\frac{1}{2},1,\dots,\frac{k}{2}\right\}$, and $M^{\leq k} \in \left\{-J^{\leq k},\dots,J^{\leq k}\right\}$. There exist linear operators that act on these spaces denoted by $\hat{\tilde{L}}_{\leq k-1} : \mathbb{R}^{r_{k-1}} \to \boldsymbol{\mathcal{F}}_{k-1}$, $\hat{\tilde{R}}_{>k} : \boldsymbol{\mathcal{F}}_{d-k} \to \mathbb{R}^{r_k}$, and orthogonal matrices $\mathbf{G}_{k-1} \in \mathbb{R}^{r_{k-1}\times r_{k-1}}$, and $\mathbf{G}_k \in \mathbb{R}^{r_k \times r_k}$, such that

$$\tilde{\mathbf{L}}_{\leq k-1} = \mathbf{Q}_L\mathbf{L}_{\leq k-1}\mathbf{G}_{k-1}, \quad \tilde{\mathbf{R}}_{>k} = \mathbf{G}_k^*\mathbf{R}_{>k}\mathbf{Q}_R. \tag{3.198}$$

The matrices $\tilde{\mathbf{L}}_{\leq k-1}$ and $\tilde{\mathbf{R}}_{>k}$ are block-diagonal.

Proof. For clarity, we introduce shorthand notations as follows: Let Ξ be the index set defined as:

$$\Xi^{\leq k} = \Bigg\{ (N,J)^{\leq k} \,|\, 0 \leq N^{\leq k} \leq 2k, \frac{\mathrm{mod}(N_s^{\leq k},2)}{2} \leq J^{\leq k} \leq \frac{N_s^{\leq k}}{2}, \text{ for all } N_s^{\leq k} \text{ and } N_p^{\leq k} \text{ with}$$
$$0 \leq N_p^{\leq k} \leq k, 0 \leq N_s^{\leq k} \leq k, 2N_p^{\leq k} + N_s^{\leq k} = N^{\leq k} \Bigg\}. \tag{3.199}$$

and let $\Xi^{>k}$ be the index set defined as:

$$\Xi^{>k} = \Bigg\{ (N,J)^{>k} \,|\, 0 \leq N^{>k} \leq 2(d-k), \frac{\mathrm{mod}(N_s^{>k},2)}{2} \leq J^{>k} \leq \frac{N_s^{>k}}{2}, \text{ for all } N_s^{>k} \text{ and } N_p^{>k} \text{ with}$$
$$0 \leq N_p^{>k} \leq d-k, 0 \leq N_s^{>k} \leq d-k, 2N_p^{>k} + N_s^{>k} = N^{>k} \Bigg\}. \tag{3.200}$$

with $N^{\leq k} + N^{>k} = N$ and $J^{\leq k} = J^{>k}$.

Using the Proposition 3.3.9, we have:

$$\mathbf{Q}_L(\left(\mathbf{\Psi}^{N,0,0}\right)^{<k-1>})\mathbf{Q}_R = \text{diag}(\mathbf{\Theta}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{>k-1}+1)}). \tag{3.201}$$

Since $J = 0$, we have $J^{\leq k-1} = J^{>k-1}$. The SVD of matrices $\mathbf{\Theta}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}}$ is given by:

$$\mathbf{\Theta}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} = \mathbf{U}_{k-1}\mathbf{S}_{k-1}\mathbf{V}^*_{k-1}, \tag{3.202}$$

with $\mathbf{U}_{k-1}, \mathbf{V}_{k-1}$ being orthogonal matrices and $\mathbf{S}_{k-1}$ being a diagonal matrix. By using the following notations:

$$\mathbf{U}^{\Xi^{\leq k-1}}_{\Xi^{\leq k-1}} := \mathbf{U}_{k-1},\ \mathbf{V}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} := \mathbf{S}_{k-1}\mathbf{V}^*_{k-1}, \tag{3.203}$$

it follows that the block-wise SVD of $\mathbf{Q}_L(\left(\mathbf{\Psi}^{N,0,0}\right)^{<k-1>})\mathbf{Q}_R$ yields to

$$\mathbf{Q}_L(\left(\mathbf{\Psi}^{N,0,0}\right)^{<k-1>})\mathbf{Q}_R = \text{diag}\left(\mathbf{U}^{\Xi^{\leq k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)}\right) \tag{3.204}$$

$$\text{diag}\left(\mathbf{V}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)}\right), \tag{3.205}$$

or we have according to Equation (3.194):

$$\left(\mathbf{\Psi}^{N,0,0}\right)^{<k-1>} = \mathbf{L}_{\leq k-1}\left(\mathbf{R}_{>k-1}\right), \tag{3.206}$$

therefore, we obtain

$$\mathbf{Q}_L\mathbf{L}_{\leq k-1}\left(\mathbf{R}_{>k-1}\right)\mathbf{Q}_R = \text{diag}\left(\mathbf{U}^{\Xi^{\leq k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)}\right)\text{diag}\left(\mathbf{V}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)}\right), \tag{3.207}$$

with $J^{\leq k-1} = J^{>k-1}$. Since the matrices $\mathbf{Q}_L$ and $\mathbf{L}_{\leq k-1}$ are orthogonal, we can introduce an orthogonal matrix denoted as $\mathbf{G}_{k-1}$, associated with the operator $\hat{G}_{k-1} : \mathbb{R}^{r_{k-1}} \to \mathbb{R}^{r_{k-1}}$ such that by defining:

$$\tilde{\mathbf{L}}_{k-1} = \text{diag}\left(\mathbf{U}^{\Xi^{\leq k-1}}_{\Xi^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)}\right), \tag{3.208}$$

we have:

$$\mathbf{Q}_L\mathbf{L}_{\leq k-1} = \tilde{\mathbf{L}}_{k-1}\mathbf{G}^*_{k-1}. \tag{3.209}$$

The proof is straightforward and is connected to the gauge freedom inherent in left-orthogonal TT decomposition, see [47]. Similarly, the SVD of matrices $\mathbf{\Theta}^{\Xi^{>k}}_{\Xi^{\leq k}}$ is given by:

$$\mathbf{\Theta}^{\Xi^{>k}}_{\Xi^{\leq k}} = \mathbf{U}_k\mathbf{S}_k\mathbf{V}^*_k, \tag{3.210}$$

with $\mathbf{U}_k, \mathbf{V}_k$ being orthogonal matrices and $\mathbf{S}_k$ being a diagonal matrix. By using the following notations:

$$\mathbf{U}^{\Xi^{\leq k}}_{\Xi^{\leq k}} := \mathbf{U}_k\mathbf{S}_k,\ \mathbf{V}^{\Xi^{>k-1}}_{\Xi^{\leq k-1}} := \mathbf{V}^*_k. \tag{3.211}$$

The block-wise SVD of $\mathbf{Q}_L\left(\mathbf{\Psi}^{N,0,0}\right)^{<k>}\mathbf{Q}_R$ yields to

$$\mathbf{Q}_L\left(\mathbf{\Psi}^{N,0,0}\right)^{<k>}\mathbf{Q}_R = \text{diag}(\mathbf{\Theta}^{\Xi^{>k}}_{\Xi^{\leq k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{\leq k}+1)}) \tag{3.212}$$

$$= \text{diag}\left(\mathbf{U}^{\Xi^{\leq k}}_{\Xi^{\leq k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{\leq k}+1)}\right) \tag{3.213}$$

$$\text{diag}\left(\mathbf{V}^{\Xi^{>k}}_{\Xi^{\leq k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{\leq k}+1)}\right) \tag{3.214}$$

$$= \mathbf{Q}_L\mathbf{L}_{\leq k}\left(\mathbf{R}_{>k}\right)\mathbf{Q}_R. \tag{3.215}$$

By defining:

$$\tilde{\mathbf{R}}_{k+1} = \text{diag}\left(\mathbf{V}^{\Xi^{>k}}_{\Xi^{\leq k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{\leq k}+1)}\right). \tag{3.216}$$

and given that the matrices $\mathbf{Q}_R$ and $\mathbf{R}_{>k}$ are orthogonal, similar to what we did previously, we can introduce an orthogonal matrix $\mathbf{G}_k$ associated with the operator $\hat{G}_k : \mathbb{R}^{r_k} \to \mathbb{R}^{r_k}$, such that

$$\left(\mathbf{R}_{>k}\right)\mathbf{Q}_R = \mathbf{G}_k\tilde{\mathbf{R}}_{k+1}. \tag{3.217}$$

$\tilde{\mathbf{L}}_{k-1}$ and $\tilde{\mathbf{R}}_{k+1}$ are block-diagonal matrix representations of the following linear operators:

$$\hat{\tilde{L}}_{\leq k-1} : \mathbb{R}^{r_{k-1}} \to \boldsymbol{\mathcal{F}_{k-1}}, \hat{\tilde{R}}_{>k} : \boldsymbol{\mathcal{F}_{d-k}} \to \mathbb{R}^{r_k}, \tag{3.218}$$

where for $\mathbb{R}^{r_k}$, it decomposes as:

$$\mathbb{R}^{r_k} \cong \bigoplus_{N^{\leq k}, J^{\leq k}, M^{\leq k}} \left(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}} \otimes \tilde{\mathcal{S}}_{k,(J,M)^{\leq k}} \right), \tag{3.219}$$

and where for $k \in [d]$:

$$r_k = \dim(\mathbb{R}^{r_k}) = \sum_{N^{\leq k}, J^{\leq k}} 2(J^{\leq k} + 1) \dim(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}), \tag{3.220}$$

with $\dim(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}}) \leq \min \left\{ \sum_{N_p^{\leq k}, N_s^{\leq k}} \dim(\mathscr{P}_{k,(N,J)^{\leq k}}^{(N_p,N_s)^{\leq k}}), \sum_{N_p^{>k}, N_s^{>k}} \dim(\mathscr{P}_{d-k,(N,J)^{>k}}^{(N_p,N_s)^{>k}}) \right\}$.

Let $\mathbf{L}_{\leq k-1,(N,J)^{>k}}$ (resp . $\mathbf{R}_{>k,(N,J)^{>k}}$) denote matrices defined as:

$$\mathbf{L}_{\leq k-1,(N,J)^{\leq k-1}} = \mathbf{U}_{\Xi^{\leq k-1}}^{\Xi^{>k-1}}, \tag{3.221}$$

respectively

$$\mathbf{R}_{>k,(N,J)^{>k}} = \mathbf{V}_{\Xi^{\leq k}}^{\Xi^{>k}}. \tag{3.222}$$

□

Now that we have all the necessary elements in place, we can express the TT-cores of the TT representation of a singlet state.

Proof of Theorem 5. As $\boldsymbol{\Psi}^{N,0,0}$ represents a singlet state ($J = 0$) with fixed particle number N, it must adhere to the following condition:

$$\boldsymbol{\Psi}^{N,0,0} \in \mathrm{Span} \left\{ \boldsymbol{\varrho}_{d,N,0}^{N_p,N_s,t} \otimes_K \boldsymbol{s}_{d,0,0} | 0 \leq N_p \leq d, 0 \leq N_s \leq d, 2N_p + N_s = N, t \in \left[\dim(\mathscr{P}_{d,N,0}^{N_p,N_s}) \right] \right\}. \tag{3.223}$$

To satisfy this condition, the 3j-Wigner condition, see [11], must be met, for fixed k:

$$\sum_{M^{\leq k-1}=-J^{\leq k-1}}^{J^{\leq k-1}} \sum_{M(\mu_k)=-J(\mu_k)}^{J(\mu_k)} \sum_{M^{>k}=-J^{>k}}^{J^{>k}} \boldsymbol{s}_{k,(J,M)^{\leq k-1}} \otimes_K \boldsymbol{s}_{J(\mu_k),M(\mu_k)} \otimes_K \boldsymbol{s}_{d-k,(J,M)^{>k}} \begin{pmatrix} J^{\leq k-1} & J(\mu_k) & J^{>k} \\ M^{\leq k-1} & M(\mu_k) & M^{>k} \end{pmatrix} = \boldsymbol{s}_{d,0,0}, \tag{3.224}$$

with $J(\mu_k)$ and $M(\mu_k)$ being defined in Equation (3.143). By Proposition 3.3.10, we have:

$$\tilde{\mathbf{L}}_{\leq k-1} = \mathrm{diag} \left(\mathbf{L}_{\leq k-1,(N,J)^{\leq k-1}} \otimes_K \mathbf{I}_{(2J^{\leq k-1}+1)\times(2J^{\leq k-1}+1)} \right), \tag{3.225}$$

and

$$\tilde{\mathbf{R}}_{>k} = \mathrm{diag} \left(\mathbf{R}_{>k,(N,J)^{>k}} \otimes_K \mathbf{I}_{(2J^{\leq k}+1)\times(2J^{\leq k}+1)} \right). \tag{3.226}$$

For ease of notation, we introduce the following block-matrix, denoted as $\mathbf{B}_{\mu_k}$, for each $\mu_k \in \{0, \uparrow, \downarrow, 2\}$, such that it can be defined as:

$$\mathbf{B}_{\mu_k} = (\mathbf{G}_{k-1}^* \mathbf{U}_k[\mu_k] \mathbf{G}_k) \Big|_{\left(\tilde{\mathscr{P}}_{k-1,(N,J)^{\leq k-1}} \otimes \tilde{\mathcal{S}}_{k-1,(J,M)^{\leq k-1}} \right) \times \left(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}} \otimes \tilde{\mathcal{S}}_{k,(J,M)^{\leq k}} \right)}. \tag{3.227}$$

It follows, from Equation (3.194), we have:

$$\boldsymbol{\Psi}^{N,0,0} = \sum_{\substack{N^{\leq k-1}, N^{>k} = N - N^{\leq k}, \\ J^{\leq k-1}, J^{>k} = J^{\leq k}}} \sum_{M^{\leq k-1}, M^{>k}} \Big[\sum_{i=1}^{\dim(\tilde{\mathscr{P}}_{k-1,(N,J)^{\leq k-1}})} \sum_{j=1}^{\dim(\tilde{\mathscr{P}}_{k,(N,J)^{\leq k}})} \mathbf{L}_{\leq k-1,(N,J)^{\leq k-1}}[:, i] \otimes_K \left(\sum_{\mu_k} \mathbf{B}_{\mu_k}(i;j) \boldsymbol{e}_{\mu_k} \right) \otimes_K \mathbf{R}_{>k,(N,J)^{>k}}[j, :] \Big]. \tag{3.228}$$

Thus, necessarily using the 3j-Wigner condition (3.224), it follows that there exist block matrices, denoted as $\tilde{\mathcal{U}}_k \in \mathbb{R}^{\tilde{r}_{k-1}\times 3\times \tilde{r}_k}$, such that

$$\begin{aligned}(\mathbf{G}_{k-1}^{*}\mathbf{U}_k[\mu_k]\mathbf{G}_k)\Big|_{\left(\tilde{\mathscr{P}}_{k-1,(N,J)\leq k-1}\otimes\tilde{\mathcal{S}}_{k-1,(J,M)\leq k-1}\right)\times\left(\tilde{\mathscr{P}}_{k,(N,J)\leq k}\otimes\tilde{\mathcal{S}}_{k,(J,M)\leq k}\right)}& \\ = (\tilde{\mathbf{U}}_k[N(\mu_k)])\Big|_{\tilde{\mathscr{P}}_{k-1,(N,J)\leq k-1}\times\tilde{\mathscr{P}}_{k,(N,J)\leq k}}&\begin{pmatrix} J^{\leq k-1} & J(\mu_k) & J^{>k} \\ M^{\leq k-1} & M(\mu_k) & M^{>k}\end{pmatrix},\end{aligned} \tag{3.229}$$

where $\tilde{\mathbf{U}}_k[N(\mu_k)] := \mathcal{U}_k[:, N(\mu_k), :]$, $N(\mu_k)$, $J(\mu_k)$ and $M(\mu_k)$ are defined in Equations (3.142) and (3.143). Additionally, (showing only the expression of $\tilde{r}_k$)

$$\tilde{r}_k = \sum_{N^{\leq k}, J^{\leq k}} \dim(\tilde{\mathscr{P}}_{k,(N,J)\leq k}). \tag{3.230}$$

□

Remark 3.3.10. In Equation (3.220), we observe that r_k is related to $\dim(\tilde{\mathscr{P}}_{k,(N,J)\leq k})$ in the following manner:

$$r_k = \dim(\mathbb{R}^{r_k}) = \sum_{N^{\leq k}, J^{\leq k}} 2(J^{\leq k}+1)\dim(\tilde{\mathscr{P}}_{k,(N,J)\leq k}). \tag{3.231}$$

Consequently, when expressing the TT-core $\mathbf{U}_k[\mu_k]$ in a compact form, we only need to store the new TT-cores, denoted as $\tilde{\mathbf{U}}_k$, with nonzero blocks $(\tilde{\mathbf{U}}_k[N(\mu_k)])\Big|_{\tilde{\mathscr{P}}_{k-1,(N,J)\leq k-1}\times\tilde{\mathscr{P}}_{k,(N,J)\leq k}}$. It is worth noting that there is an advantage in using SU(2) symmetry, which leads to a reduction in the TT-rank to approximately $\frac{r_k}{\sum_{J\leq k}(2J^{\leq k}+1)}$. This reduction becomes particularly evident when $J^{\leq k}$ for $k \in [d]$ are sufficiently large. Moreover, it is always possible to further reduce these TT-ranks by a truncated SVD.

3.4 Concluding remarks and perspectives

In this chapter, we have focused on the TTO decomposition of the quantum chemical Hamiltonian operator which is at the core of the QC-DMRG algorithm. We started by giving a comprehensive review of both the exact and approximate approaches for constructing the TTO decomposition of the operator. One of the key takeaways has been the necessity to carefully employ the low-rank approximation method through tSVD. We showed that it is imperative to respect degenerate singular values, as this ensures the preservation of essential operator properties, including symmetry, particle number conservation, as well as the conservation of the z-component of the total spin. Furthermore, we have presented numerical results showing that employing tSVD with some numerical thresholds can lead to the non-conservation of the total spin and can impact the operator's structure. This impact manifests in the emergence of non-existent virtual interactions, even when dealing with an operator with at most 2-body interactions. Inspired by the work of [2], we have provided a constructive demonstration of the structure of the TT-cores within the TTO decomposition of a general particle-preserving p-body Hamiltonian operator. Our findings illustrate a particular block-sparse structure in this context with at-most $2p+1$ blocks per TT-core. Additionally, we have offered preliminary theoretical insights into the structure of the TT-cores within the TT decomposition of the corresponding eigenfunction, the wavefunction, that is assumed to be invariant under non-Abelian symmetries, such as SU(2), which leads as well to block structured TT-cores with fewer blocks. Our analysis were only limited for the case of the singlet state ($J = 0$). While these insights might not be entirely novel within the physics and chemistry communities, our goal has been to present this content in a more accessible and reader-friendly manner. Looking ahead, our ongoing work focuses on exploring the structure of the Hamiltonian operator under SU(2) symmetries and implementing these findings in our forthcoming library, which we will introduce in the following chapter.

Chapter 4

Algorithmic design and numerical experiments with QC-DMRG

Contents

4.1 Introduction

In this chapter, we present the algorithmic design for the major operations in QC-DMRG, which have been implemented in our proof-of-concept QC-DMRG library. This library is specifically tailored for the quantum chemical Hamiltonian operator model defined in Equation (1.138). Notably, it incorporates particle number conservation as specified in Equation (PN). While I am the main developer of the library, this collaborative effort includes Mi-Song Dupuy and my supervisor, Laura Grigori. Additionally, we have the valuable contribution of Daniel Torres (Research Engineer at Inria) who has contributed to both code development and the ongoing parallelization effort of specific parts of the algorithms using MPI. Our goal is to create a user-friendly QC-DMRG library that is based on the block-sparse structures of the TT decomposition of the operator and eigenfunction, see Corollary 3.3.1 and Theorem 4. This library will be accessible to researchers both within our group and beyond, making it easier for them to perform their own numerical simulations. This chapter includes the findings intended for submission of a paper with Laura Grigori, Daniel Torres, Mi-Song Dupuy and Eric Cancès.

The core of this implementation is built upon exploiting the block-sparse TTO decomposition of the Hamiltonian operator, as discussed in Section 3.3.1, along with its corresponding eigenfunction, as already derived in [2] and revisited in Section 3.3.1. This particular representation leads to performing matrix-free operations on the nonzero blocks within QC-DMRG calculations. These specific techniques are detailed in Section 4.3. While existing DMRG libraries certainly offer robust tools for practical applications, as reviewed in Section 4.2, developing our own library allows us to engage directly with the algorithm on a deeper level and be able to design algorithms for key steps of QC-DMRG.

This library was driven by specific research goals. On a technical level, we aim to design a generic method for the construction of a block-sparse TTO decompositions of quantum chemical particle-preserving Hamiltonian operators, based only on the knowledge of the particle number N, number of orbitals d, and the one and two-electron integrals tensors. This approach can also be extended to Hamiltonian operators involving p-body interactions, as commonly encountered in nuclear physics [119]. Additionally, we aim to exploit as well the block-sparse structure present in the TT decomposition of the associated eigenfunction, assumed to be an eigenvector of the particle number operator, to develop an

heuristic scheme for block-sparse tensor contractions, which are present in QC-DMRG calculations. Furthermore, we seek to leverage this block sparsity to enhance the speed of matrix operations in the iterative eigenvalue solver during each sweep. Further details can be found in Section 4.3. At a broader level, our proof-of-concept library is designed to be accessible, adaptable, and user-friendly, enabling researchers within our group or beyond who wish to conduct their own simulations using DMRG or explore innovative ideas to improve its performance, while being able to study relevant realistic systems.

Hence, it is important to emphasize that our implementation is currently in the prototyping phase, and our goal is not necessarily to directly compete with existing libraries which are under development for more than a decade. However, we do aspire to achieve a level of performance comparable with the cutting-edge libraries used for molecular simulations with tensor networks as ITensor library [28]. To ensure the utmost accessibility and flexibility of this implementation, we chose to use the high-level Julia programming language. To validate our code functionality, we can perform QC-DMRG calculations on diverse molecules. For this purpose, we can extract essential molecular properties, such as the number of orbitals considered, i.e size of the system, number of particles, and the one- and two-electron integrals (defined in Equations (1.81) and (1.82)), from files named FCIDUMP, generated using the Pyscf library [116] which can be called through our Julia-based code. In terms of computational efficiency, the execution times within our code are primarily dominated by tensor contractions on blocks, which ultimately involve multiple matrix multiplications. These multiplications are efficiently carried out using the well-known Julia library, *TensorOperations.jl* [23], which in turn calls BLAS routines. In the context of constructing TTO decomposition of the Hamiltonian operator for large molecules, we use *Distributed.jl* [89] library within Julia for distributed and parallel computing. All the calculations are carried out using Cleps cluster from Inria, Paris, France. This machine has 4 partitions. We use cpu homogen partition which contains 20 nodes with hyper-threading such that we can allocate a maximum of 64 logical cores per node (Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz) with a memory of 6GB per core.

In our implementation, we use several composite types also known as *struct* in Julia which mimic the behaviour of traditional classes. Each composite type has fields (attributes) to store data, similar to instance variables in other programming languages. We refer interested readers to Appendix .6, which features a comprehensive diagram providing an overview of the fundamental structures within our proof-of-concept library.

The reminder of Chapter 4 is organized as follows: we start by describing the construction approach for a block-sparse TTO decomposition of the quantum chemical Hamiltonian operator. Initially designed for 2-body interactions in the spatial-orbital basis, the approach is extended to accommodate a general p-body particle-preserving Hamiltonian operator. Then, we cover the algorithms for basic arithmetic operations on block-sparse TT representations such as compression and orthogonalization. Leveraging the block-sparse structure in the TT decomposition of the eigenfunction and the TTO decomposition of the operator, efficient block-sparse tensor contractions are discussed. Numerical experiments are conducted on various molecules, including a Hydrogen chain, LiH, and Nitrogen. Comparative evaluations of computational time for QC-DMRG calculations, with fixed sweeps, are performed between the developed code and the ITensor library and each presented subsection concludes with numerical results.

4.2 Related work

In this section, we provide an overview of existing tensor-based software packages commonly employed for electronic structure calculations. These software packages can be categorized into two main groups that we detail here.

1. The first group offer a versatile framework for a wide range of tensor-based algorithms including DMRG, suitable for applications in quantum many-body physics, condensed matter physics, and quantum chemistry. Among these packages, the most popular library is *ITensor* [28] primarily written in C++ which also provides a native Julia interface. It is renowned for its efficiency in handling tensor operations and memory management. It offers a 2-site DMRG implementation with the flexibility of constructing a TTO decomposition for a wide range of Hamiltonian models.

It employs SVD-based compression techniques to reduce the TTO-ranks. Additionally, ITensor supports Abelian symmetries such as the particle number conservation and plans to include SU(2) spin symmetry in a future version. *TenPy* [42] is also a well-known software for large-scale numerical simulations involving tensor networks. TenPy's implementation primarily relies on Python, with some components in C++. It has some similar features to ITensor: generation of TTO decomposition of Hamiltonian models, support of Abelian symmetries, implementation of both 1-site and 2-site DMRG.

2. The second group of software packages is specifically designed for ab initio[1] quantum chemistry calculations, with a strong focus on efficiently implementing QC-DMRG method tailored for quantum chemistry applications. Within this category, there are four notable codes developed by different research groups: *Block/Block2* [137, 138], *CheMPS2* [134], *QCMaquis* [59] and the recently developed package *Kylin 1.0* [135]. Most of these packages are primarily written in the C++ programming language and are compatible with various interfaces like PySCF (the first three codes), allowing for the generation of quantum chemistry integrals: the one- and two-electron integrals. They offer support for various symmetries, including Abelian symmetries (ensuring the conservation of particle number) and SU(2) spin symmetries. They also provide both 1-site and 2-site QC-DMRG implementations.

4.3 Algorithmic aspects of QC-DMRG

Prior to delving into the detailed aspects of algorithms within our library, we revisit the block-sparse structure inherent to the TTO decomposition of the Hamiltonian operator, as demonstrated in Theorem 4. Additionally, we revisit the block-sparse structure of the TT decomposition of its eigenfunction, as described in Corollary 3.3.1, and provide illustrations for more clarity. Here, we employ the spin-orbital basis, as introduced in (1.70). Let $n = 2$ and d be the number of spin-orbital basis functions considered. All along this chapter, we suppose that the particle number is conserved.

- **Block-sparse TT decomposition of the eigenfunction $\psi \in \mathbb{R}^{2\times\cdots\times 2}$.** Let $\psi \in \mathbb{R}^{2\times\cdots\times 2}$ be a d-order tensor, let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ be the TT decomposition of ψ with TT-ranks $(r_1, \ldots, r_d)$. We recall that $\mathcal{U}_k \in \mathbb{R}^{r_{k-1}\times 2\times r_k}$, for $k \in \{2, \ldots, d-1\}$, and $\mathcal{U}_1 \in \mathbb{R}^{1\times 2\times r_1}$, $\mathcal{U}_d \in \mathbb{R}^{r_{d-1}\times 2\times 1}$ such that $r_0 = r_d = 1$. According to Corollary 3.3.1, the TT-cores have a block-sparse structure. For a fixed value of $\mu_k \in \{1, 2\}$, the matrices $\mathbf{U}_k[\mu_k] := \mathcal{U}_k[:, \mu_k, :] \in \mathbb{R}^{r_{k-1}\times r_k}$ are block matrices. These blocks are denoted by $\mathbf{U}_k^{(i_{k-1},i_k)}[\mu_k] \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}}\times\rho^{\mathbf{\Psi}}_{k,i_k}}$, $\mu_k \in \{1, 2\}$ for $k \in [d]$, $i_0 = 0$, $i_d = N + 1$, and $\rho^{\mathbf{\Psi}}_{0,0} = \rho^{\mathbf{\Psi}}_{d,N+1} = 1$. For $k \in \{2, \ldots, d-1\}$, we have:

$$r_{k-1} = \sum_{i_{k-1}\in S^{\mathbf{\Psi}}_{k-1}} \rho^{\mathbf{\Psi}}_{k-1,i_{k-1}}, \; r_k = \sum_{i_k\in S^{\mathbf{\Psi}}_k} \rho^{\mathbf{\Psi}}_{k,i_k}, \tag{4.1}$$

where

$$S^{\mathbf{\Psi}}_{k-1} = \{\max\{1, N-d+k\}, \ldots, \min\{N+1, k\}\}, \tag{4.2}$$

$$S^{\mathbf{\Psi}}_{k} = \{\max\{1, N-d+k+1\}, \ldots, \min\{N+1, k+1\}\}, \tag{4.3}$$

with N being the number of particles and

$$i_{k-1} + q_2(\mu_k) = i_k, \tag{4.4}$$

with $q_2 : \{1, 2\} \to \{0, 1\}$. We provide in Figure 4.1, an illustration presenting the corresponding structure of the TT-cores within the block-sparse TT representation of the eigenfunction, according to the description above.

[1]Here, equations are derived directly from the Schrödinger equation, based only on first principles, without any reliance on experimental input or additional assumptions.

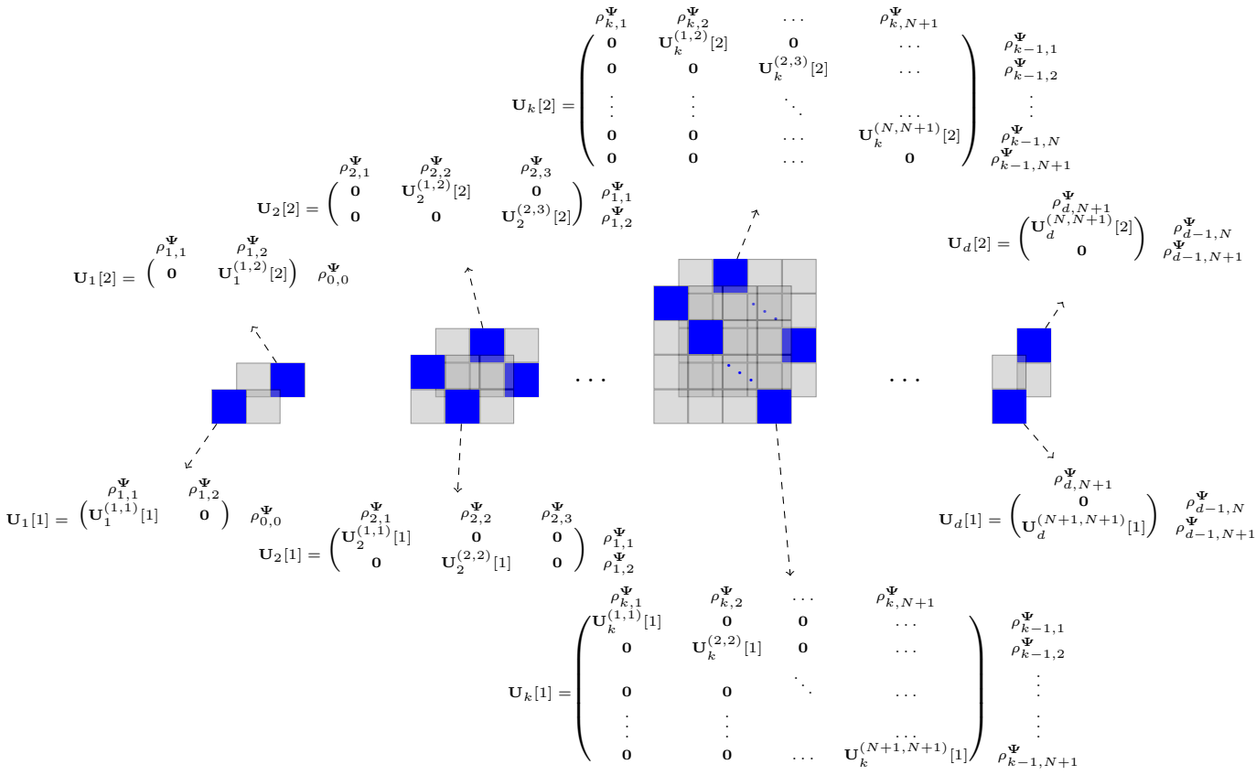


Figure 4.1: Block-sparse TT decomposition of $\psi \in \mathbb{R}^{n\times n\times\ldots\times n}$. For each TT-core, $\mathbf{U}_k[\mu_k]$ with fixed $\mu_k \in \{1,2\}$ and $k \in [d]$, a distinct block structure is present: a block-diagonal matrix when $\mu_k = 1$ and a block matrix with nonzero blocks only on the upper diagonal when $\mu_k = 2$.

- **Block-sparse TTO decomposition of the Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{2^d\times 2^d}$.** Let $(\mathcal{H}_1,\ldots,\mathcal{H}_d)$ be the TTO decomposition of $\mathbf{H}$ with TTO-ranks $(R_1,\ldots,R_d)$. We recall that $\mathcal{H}_k \in \mathbb{R}^{R_{k-1}\times 2\times 2\times R_k}$, for $k \in \{2,\ldots,d-1\}$, $\mathcal{H}_1 \in \mathbb{R}^{1\times 2\times 2\times R_1}$ and $\mathcal{H}_d \in \mathbb{R}^{R_{d-1}\times 2\times 2\times 1}$ such that $R_0 = R_d = 1$. According to Theorem 4, the TT-cores have a block-sparse structure. For fixed values of $\mu_k,\nu_k \in \{1,2\}$, $\mathbf{H}_k[\mu_k,\nu_k] := \mathcal{H}_k[:,\mu_k,\nu_k,:] \in \mathbb{R}^{R_{k-1}\times R_k}$ are block matrices. These blocks are denoted by $\mathbf{H}_k^{(j_{k-1},j_k)}[\mu_k,\nu_k] \in \mathbb{R}^{\rho^{\mathbf{H}}_{k-1,j_{k-1}}\times\rho^{\mathbf{H}}_{k,j_k}}$, $\mu_k,\nu_k \in \{1,2\}$ for $k \in [d]$, $j_0 = 0$, $j_d = 0$, $\rho^{\mathbf{H}}_{0,0} = \rho^{\mathbf{H}}_{d,0} = 1$, and for $k \in \{2,\ldots,d-1\}$, we have:

$$R_{k-1} = \sum_{j_{k-1}\in S^{\mathbf{H}}_{k-1}} \rho^{\mathbf{H}}_{k-1,j_{k-1}}, \quad R_k = \sum_{j_k\in S^{\mathbf{H}}_k} \rho^{\mathbf{H}}_{k,j_k}, \tag{4.5}$$

where

$$S^{\mathbf{H}}_k = \{-\beta_k,\ldots,\beta_k\}. \tag{4.6}$$

Here $\beta_k = \min\{k, d-k, p\}$, for $k \in [d]$, where p denotes the number of p-body interactions considered. Additionally, we have:

$$j_{k-1} + t_k = j_k, \tag{4.7}$$

where $t_k = q_2(\mu_k) - q_2(\nu_k)$. We provide in Figure 4.2, an illustration presenting the corresponding structure of the TT-cores within the block-sparse TTO representation of the Hamiltonian operator, according to the description above.

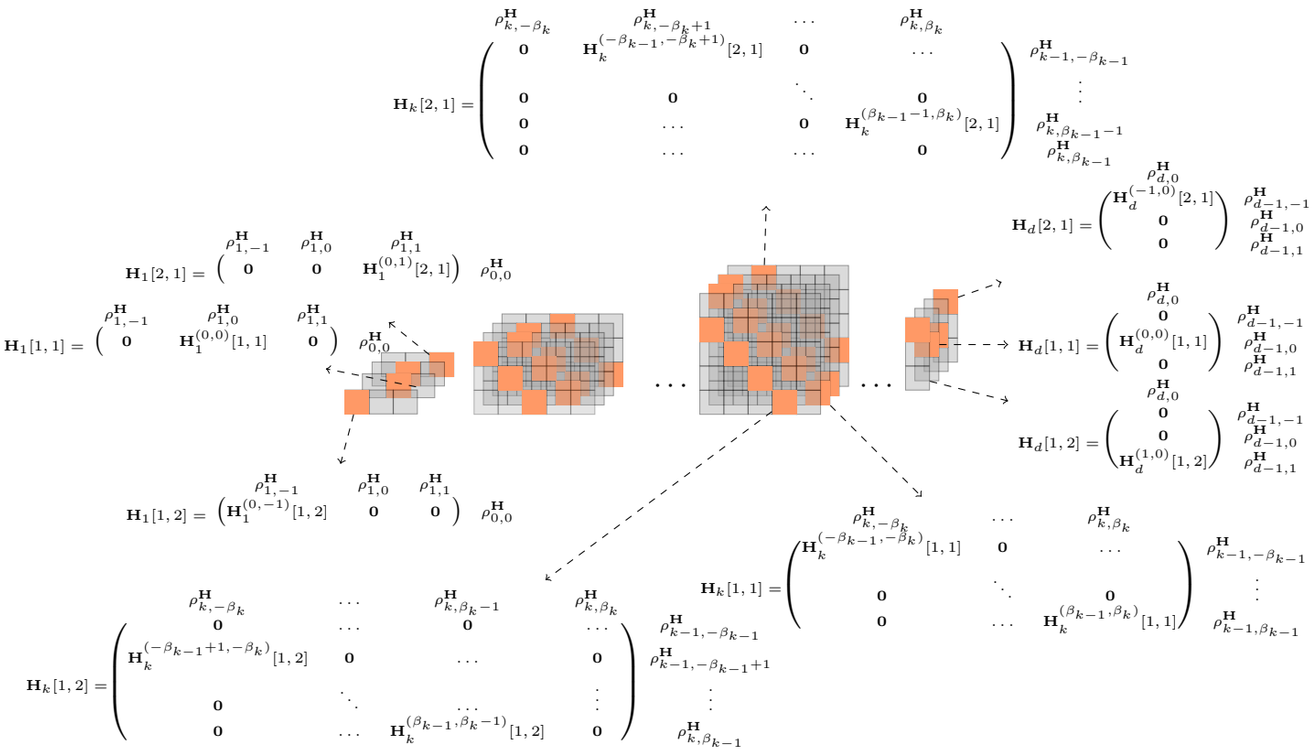


Figure 4.2: TTO decomposition of the Hamiltonian operator: for $k \in [d]$, $\mathbf{H}_k[1,1]$ and $\mathbf{H}_k[2,2]$ share the same block-sparse structure. For each TT-core, $\mathbf{H}_k[\mu_k, \nu_k]$ with fixed $\mu_k, \nu_k \in \{1,2\}$ and $k \in [d]$, a distinct block structure is present: a block-diagonal matrix when $(\mu_k, \nu_k) = (1,1)$ or $(\mu_k, \nu_k) = (2,2)$, a block matrix with nonzero blocks only on the upper diagonal when $(\mu_k, \nu_k) = (2,1)$, and a block matrix with nonzero blocks only on the lower diagonal when $(\mu_k, \nu_k) = (1,2)$.

In the following sections, we start by describing the construction approach employed to generate a block-sparse TTO decomposition of the quantum chemical Hamiltonian operator, with at most 2-body interactions in the spatial-orbital basis (refer to Remark 1.3.9), as defined in (1.138), such that we explain how its expression can be reformulated within the spin-orbital basis. Subsequently, we extend this construction approach to accommodate a general p-body particle-preserving Hamiltonian operator, as defined in (3.46). The algorithms presented cover basic arithmetic operations, including the multiplication and addition of two block-sparse TT representations. We also introduce algorithms for the compression and orthogonalization of TT representations in a block-sparse structure.

Taking advantage of the block-sparse structure in both the TT decomposition of the eigenfunction (see Figure (4.1)) and the TTO decomposition of the operator (see Figure 4.2), we offer insights into efficient block-sparse tensor contractions. Additionally, we elaborate on how these inherent structures can be leveraged to accelerate matrix-vector multiplications involved in the eigensolver at each micro-step of the QC-DMRG algorithm.

Notes on numerical experiments: The experiments are conducted using various molecules. We will look at a Hydrogen chain, going from the Di-Hydrogen molecule H_2 to the molecule H_{16}, where $d = 4$ denotes the minimum number of spin-orbitals, and $d = 32$ represents the maximum number of spin-orbitals in the chain, in the STO-3g basis set. Furthermore, we will look at the LiH molecule in the cc-pVDZ basis set, considering $d = 38$. Additionally, we will consider the Nitrogen molecule N_2 in the cc-pVDZ basis set, as well, with $d = 52$. Throughout our analysis, we present a comparative evaluation of the computational time required for QC-DMRG calculations, with fixed sweeps, between our code and the state-of-the-art library ITensor. In this chapter, following each subsection, numerical results are presented.

4.3.1 Block-sparse TTO construction

In this subsection, we provide a comprehensive explanation of the construction approach we have chosen for generating the TTO decomposition of the quantum chemical 2-body Hamiltonian operator defined in (1.138), while preserving the particle number and thereby ensuring a block-sparse structure in its TT-cores as explained in Section 3.3.1. We will also introduce the related key algorithms. Importantly, the approach we present can be extended to a p-body Hamiltonian operator, as defined in (3.46).

This section is structured into four main parts. Firstly, we recall the expression of the 2-body Hamiltonian operator in the spatial-orbital basis, by showcasing how it can be further simplified through the application of anti-commutation relations between the creation and annihilation operators as defined in (1.132). Secondly, we explain how the Hamiltonian operator can be expressed in the spin-orbital basis, elucidating the various essential transformation operations involved. Thirdly, we explain the primary construction steps involved in creating the TTO decomposition of the 2-body Hamiltonian with block-sparse structure. Lastly, we conclude by providing an extension of this construction to a p-body Hamiltonian operator.

Quantum chemical Hamiltonian operator

We recall the 2-body Hamiltonian operator expressed in the spatial-orbital basis, as defined in (1.138). Let d_{spatial} be the number of spatial-orbital basis functions, see Remark 1.3.9, the Hamiltonian matrix $\mathbf{H}_{\text{spatial}} \in \mathbb{R}^{4^{d_{\text{spatial}}} \times 4^{d_{\text{spatial}}}}$ is:

$$\mathbf{H}_{\text{spatial}} = \sum_{i,j=1}^{d_{\text{spatial}}} \sum_{s=\pm\frac{1}{2}} h_{ij} \mathbf{A}_{i,s}^* \mathbf{A}_{j,s} + \frac{1}{2} \sum_{i,j,k,\ell=1}^{d_{\text{spatial}}} \sum_{s,s'=\pm\frac{1}{2}} v_{ijk\ell} \mathbf{A}_{i,s}^* \mathbf{A}_{k,s'}^* \mathbf{A}_{\ell,s'} \mathbf{A}_{j,s}, \tag{4.8}$$

where h_{ij} (resp. $v_{ijk\ell}$) are the one-electron integrals (resp. two-electron integrals) defined in Equation (1.86) (resp. (1.87)). The matrices $\mathbf{A}_{i,s}, i \in [d_{\text{spatial}}]$, $s \in \left\{\pm\frac{1}{2}\right\}$ are the matrix representation of the creation and annihilation operators, defined in Equations (1.135), (1.136).

Given the anti-commutation relations between the creation and annihilation operators, as specified in (1.132), we revisit these relations in the following by employing the matrix representations of these operators. For $s, s' \in \left\{\pm\frac{1}{2}\right\}, i, j \in [d_{\text{spatial}}]$, we have:

$$\begin{aligned} &\mathbf{A}_{i,s}\mathbf{A}_{j,s'} + \mathbf{A}_{j,s'}\mathbf{A}_{i,s} = \mathbf{0}, \\ &\mathbf{A}_{i,s}^*\mathbf{A}_{j,s'}^* + \mathbf{A}_{j,s'}^*\mathbf{A}_{i,s}^* = \mathbf{0}, \\ &\mathbf{A}_{i,s}\mathbf{A}_{j,s'}^* + \mathbf{A}_{j,s'}^*\mathbf{A}_{i,s} = \delta_{ij}\delta_{ss'}\mathbf{I}_{4^{d_{\text{spatial}}}}. \end{aligned} \tag{4.9}$$

By using the relations in (4.9), the second terms in Equation (4.8), denoted by $\mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s}$ for $i,k,\ell,j \in [d_{\text{spatial}}], s, s' \in \left\{\pm\frac{1}{2}\right\}$, can be expressed as:

$$\begin{aligned} \mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} &= \mathbf{A}_{k,s'}^*\mathbf{A}_{i,s}^*\mathbf{A}_{j,s}\mathbf{A}_{\ell,s'} \\ \mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} &= -\mathbf{A}_{k,s'}^*\mathbf{A}_{i,s}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} \\ \mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} &= -\mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{j,s}\mathbf{A}_{\ell,s'}, \end{aligned} \tag{4.10}$$

and

$$\begin{aligned} \mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} &= \mathbf{A}_{i,s}^*(\delta_{\ell,k}\mathbf{I}_{4^{d_{\text{spatial}}}} - \mathbf{A}_{\ell,s'}\mathbf{A}_{k,s'}^*)\mathbf{A}_{j,s} \\ &= \delta_{\ell,k}\mathbf{A}_{i,s}^*\mathbf{A}_{j,s} - \mathbf{A}_{i,s}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{k,s'}^*\mathbf{A}_{j,s}. \end{aligned} \tag{4.11}$$

Additionally, we define, see [2, 17]:

$$w_{ijk\ell} = \begin{cases} \frac{1}{2}\left(v_{ijk\ell} + v_{k\ell ij} - v_{kji\ell} - v_{i\ell kj}\right), & i < k, \ell < j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

By using Equations (4.12),(4.11), and (4.10), one obtains a simplified expression of $\mathbf{H}_{\text{spatial}}$ as follows:

$$\mathbf{H}_{\text{spatial}} = \sum_{i,j=1}^{d_{\text{spatial}}} \sum_{s=\pm\frac{1}{2}} h_{ij}\mathbf{A}_{i,s}^*\mathbf{A}_{j,s} + \sum_{k,j=1}^{d_{\text{spatial}}} \sum_{i=1}^{k-1} \sum_{\ell=1}^{j-1} \sum_{s,s'=\pm\frac{1}{2}} w_{ijk\ell}\left(\delta_{\ell,k}\mathbf{A}_{i,s}^*\mathbf{A}_{j,s} - \mathbf{A}_{i,s}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{k,s'}^*\mathbf{A}_{j,s}\right). \tag{4.13}$$

By noting that $\mathbf{A}_{i,s}^*\mathbf{A}_{k,s'}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{j,s} = 0$ if $i = k$, $\ell = j$, and $s = s'$, (4.13) can be further simplified as follows:

$$\begin{aligned}\mathbf{H}_{\text{spatial}} = &\sum_{i=1}^{d_{\text{spatial}}}\sum_{s=\pm\frac{1}{2}}\left(h_{ii}\mathbf{A}_{i,s}^*\mathbf{A}_{i,s} + w_{iiii}\left(\mathbf{A}_{i,s}^*\mathbf{A}_{i,s} - \mathbf{A}_{i,s}^*\mathbf{A}_{i,-s}\mathbf{A}_{i,-s}^*\mathbf{A}_{i,s}\right)\right)\\ &+\sum_{\substack{i,j=1\\ i\neq j}}^{d_{\text{spatial}}}\sum_{s=\pm\frac{1}{2}} h_{ij}\mathbf{A}_{i,s}^*\mathbf{A}_{j,s} + \sum_{j=1}^{d_{\text{spatial}}}\sum_{\ell=1}^{j-1}\sum_{i=1}^{\ell-1}\sum_{s,s'=\pm\frac{1}{2}} w_{ij\ell\ell}\left(\mathbf{A}_{i,s}^*\mathbf{A}_{j,s} - \mathbf{A}_{i,s}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{\ell,s'}^*\mathbf{A}_{j,s}\right)\\ &-\sum_{k,j=1}^{d_{\text{spatial}}}\sum_{i=1}^{k-1}\sum_{\substack{\ell=1\\ k\neq\ell}}^{j-1}\sum_{s,s'=\pm\frac{1}{2}} w_{ijk\ell}\mathbf{A}_{i,s}^*\mathbf{A}_{\ell,s'}\mathbf{A}_{k,s'}^*\mathbf{A}_{j,s}.\end{aligned} \tag{4.14}$$

To construct the Hamiltonian matrix $\mathbf{H}_{\text{spatial}}$, we need to carry out numerous operations involving $\mathbf{A}_{i,s}^*\mathbf{A}_{j,s'}$ for $i,j \in [d_{\text{spatial}}]$ and $s,s' \in \left\{\pm\frac{1}{2}\right\}$, which we will refer to as the one-body terms. For the sake of simplicity, we can work within the spin-orbital basis. Consequently, we need to represent the Hamiltonian in this basis as described in the following.

Spin-orbital basis transformation

As defined in (1.136),(1.135), the creation and annihilation operators can be expressed as follows:

$$\mathbf{A}_{i,s} = \left(\bigotimes_{k=1}^{i-1}{}_K\,\mathbf{Z}\right)\otimes_K \mathbf{A}_s \otimes_K \left(\bigotimes_{k=i+1}^{d_{\text{spatial}}}{}_K\,\mathbf{I}_4\right) \tag{4.15}$$

and

$$\mathbf{A}_{i,s}^* = \left(\bigotimes_{k=1}^{i-1}{}_K\,\mathbf{Z}\right)\otimes_K \mathbf{A}_s^* \otimes_K \left(\bigotimes_{k=i+1}^{d_{\text{spatial}}}{}_K\,\mathbf{I}_4\right), \tag{4.16}$$

where $i \in \left[d_{\text{d}_{\text{spatial}}}\right]$ and $s \in \left\{\pm\frac{1}{2}\right\}$. Given that

$$\mathbf{A}_s = \begin{cases}\mathbf{A}\otimes_K \mathbf{I}_2, & \text{if } s = \frac{1}{2},\\ \mathbf{S}\otimes_K \mathbf{A}, & \text{if } s = -\frac{1}{2}.\end{cases}, \text{ and } \mathbf{Z} = \mathbf{S}\otimes\mathbf{S} \in \mathbb{R}^{4^{d_{\text{spatial}}}\times 4^{d_{\text{spatial}}}}, \tag{4.17}$$

with

$$\mathbf{S} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \mathbf{A} = \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}. \tag{4.18}$$

$\mathbf{A}_{i,s}$ and $\mathbf{A}_{i,s}^*$ can be expressed differently as follows:

$$\mathbf{A}_{i,s} = \left(\bigotimes_{k=1}^{2(i-1)}{}_K\,\mathbf{S}\right)\otimes_K \mathbf{A} \otimes_K \left(\bigotimes_{k=2i+1}^{2d_{\text{spatial}}}{}_K\,\mathbf{I}_2\right), \tag{4.19}$$

and

$$\mathbf{A}_{i,s}^* = \left(\bigotimes_{k=1}^{2(i-1)}{}_K\,\mathbf{S}\right)\otimes_K \mathbf{A}^* \otimes_K \left(\bigotimes_{k=2i+1}^{2d_{\text{spatial}}}{}_K\,\mathbf{I}_2\right). \tag{4.20}$$

Following Remark 1.3.9, let d be the number of spin-orbital basis functions such that $d = 2d_{\text{spatial}}$. One can verify that $\mathbf{A}_{i,s}$ can be expressed as follows: for $i \in [d_{\text{spatial}}], \eta \in [d]$

$$\mathbf{A}_{i,s} = \mathbf{A}_\eta, \tag{4.21}$$

with $\mathbf{A}_\eta$ being defined in Equation (1.123) such that

$$\eta = 2(i-1)+1 \text{ if } s = \frac{1}{2},\ \eta = 2(i-1)+2 \text{ if } s = \frac{-1}{2}. \tag{4.22}$$

As an illustration, consider the case where $i = 2$ and $s = \frac{1}{2}$. According to (4.19), we obtain:

$$\mathbf{A}_{2,\frac{1}{2}} = \mathbf{S}\otimes_K \mathbf{S}\otimes_K \mathbf{A} = \mathbf{A}_3. \tag{4.23}$$

By combining (4.21) with (4.14), the formulation of the Hamiltonian in the spin-orbital basis, denoted by $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$, is as follows:

$$\mathbf{H} = \mathbf{H}_{(1)} + \mathbf{H}_{(2)}, \tag{4.24}$$

where

$$\begin{aligned}\mathbf{H}_{(1)} = &\sum_{i=1}^{d}\sum_{l=1}^{2}(h_{ii} + w_{iiii})\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(i-1)+l}\right) + \sum_{\substack{i,j=1\\ i\neq j}}^{d}\sum_{l=1}^{2} h_{ij}\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(j-1)+l}\right)\\ &+ \sum_{j=1}^{d}\sum_{\ell=1}^{j-1}\sum_{i=1}^{\ell-1}\sum_{l,l'=1}^{2} w_{ij\ell\ell}\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(j-1)+l'}\right),\end{aligned} \tag{4.25}$$

and

$$\begin{aligned}\mathbf{H}_{(2)} = &-\sum_{i=1}^{d}\sum_{l=1}^{2} w_{iiii}\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(i-1)+l+(-1)^{l+1}}\mathbf{A}^*_{2(i-1)+l+(-1)^{l+1}}\mathbf{A}_{2(i-1)+l}\right)\\ &- \sum_{k,j=1}^{d}\sum_{i=1}^{k-1}\sum_{\substack{\ell=1\\ k\neq\ell}}^{j-1}\sum_{l,l'=1}^{2} w_{ijk\ell}\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(\ell-1)+l'}\mathbf{A}^*_{2(k-1)+l'}\mathbf{A}_{2(j-1)+l}\right)\\ &- \sum_{j=1}^{d}\sum_{\ell=1}^{j-1}\sum_{i=1}^{\ell-1}\sum_{l,l'=1}^{2} w_{ij\ell\ell}\left(\mathbf{A}^*_{2(i-1)+l}\mathbf{A}_{2(\ell-1)+l'}\mathbf{A}^*_{2(\ell-1)+l'}\mathbf{A}_{2(j-1)+l}\right).\end{aligned} \tag{4.26}$$

Here, $\mathbf{H}_{(1)}$ denotes the 1-body operator, which is written as a sum over products of matrices in the form $\mathbf{A}^*_\eta\mathbf{A}_\kappa \in \mathbb{R}^{2^d\times 2^d}$, where $\eta, \kappa \in [d]$. The product $\mathbf{A}^*_\eta\mathbf{A}_\kappa$ is referred to as the 1-body term. On the other hand, $\mathbf{H}_{(2)}$ represents the 2-body operator, written as a sum over products of two 1-body terms, i.e $(\mathbf{A}^*_\eta\mathbf{A}_\kappa)(\mathbf{A}^*_\gamma\mathbf{A}_\lambda) \in \mathbb{R}^{2^d\times 2^d}$, where $\eta, \kappa, \lambda, \gamma \in [d]$. The product$(\mathbf{A}^*_\eta\mathbf{A}_\kappa)(\mathbf{A}^*_\gamma\mathbf{A}_\lambda)$ is referred to as the 2-body term.

In the upcoming discussion, we explain the process of deriving the block-sparse TTO decomposition for both the 1-body and 2-body operators, using Theorem 4. This involves employing arithmetic operations on TTs, as outlined in Proposition 1.2.2.

Block-sparse TTO decomposition of the 1-body operator

As elaborated in Section 3.2.3, the sum of 1-body term, leads to the construction of a TTO decomposition for the 1-body operator. This representation can be viewed as a sum of rank-1 TTO decompositions, i.e TTO decompositions where all the TTO-ranks are equal to 1. In what follows, we explain the process of building a block-sparse TTO decomposition for the 1-body operator, as described in Section 3.3. First, let us examine the matrix elements of the 1-body terms. Let $\mathbf{A}^*_\eta\mathbf{A}_\kappa \in \mathbb{R}^{2^d\times 2^d}, \eta, \kappa \in [d]$ be the matrix with entries $\left(\mathbf{A}^*_\eta\mathbf{A}_\kappa\right)(\overline{\mu_1,\ldots,\mu_d};\overline{\nu_1,\ldots,\nu_d})$, for $\mu_k, \nu_k \in \{1,2\}$, $k \in [d]$. There are three distinct cases to consider, as outlined in [113]: $\eta = \kappa$, $\eta > \kappa$, and $\eta < \kappa$.

Assume $\eta = \kappa$, then

$$\left(\mathbf{A}^*_\eta\mathbf{A}_\eta\right)(\overline{\mu_1,\ldots,\mu_d};\overline{\nu_1,\ldots,\nu_d}) = \delta_{\mu_1,\nu_1}\ldots\delta_{\mu_\eta,\nu_\eta}\delta_{\nu_\eta,1}\ldots\delta_{\mu_d,\nu_d}. \tag{4.27}$$

Assume $\eta < \kappa$, then

$$\left(\mathbf{A}^*_\eta\mathbf{A}_\kappa\right)(\overline{\mu_1,\ldots,\mu_d};\overline{\nu_1,\ldots,\nu_d}) = (-1)^{\nu_\eta+\ldots+\nu_{\kappa-1}}\delta_{\mu_1,\nu_1}\ldots\delta_{\mu_\eta,\nu_\eta+1}\ldots\delta_{\mu_\kappa,\nu_\kappa-1}\ldots\delta_{\mu_d,\nu_d}. \tag{4.28}$$

Assume $\eta > \kappa$, then

$$\left(\mathbf{A}^*_\eta\mathbf{A}_\kappa\right)(\overline{\mu_1,\ldots,\mu_d};\overline{\nu_1,\ldots,\nu_d}) = (-1)^{1+\nu_\kappa+\ldots+\nu_{\eta-1}}\delta_{\mu_1,\nu_1}\ldots\delta_{\mu_\kappa,\nu_\kappa-1}\ldots\delta_{\mu_\eta,\nu_\eta+1}\ldots\delta_{\mu_d,\nu_d}. \tag{4.29}$$

Let $(\mathcal{T}_1,\ldots,\mathcal{T}_d)$ be the block-sparse TTO decomposition of a 1-body term (see Theorem 4), with $\mathcal{T}_k \in \mathbb{R}^{1\times 2\times 2\times 1}, k \in [d]$ such that all TTO-ranks are equal to 1. The matrix elements of the 1-body terms are given by (see (1.142)):

$$\left(\mathbf{A}^*_\eta\mathbf{A}_\kappa\right)(\overline{\mu_1,\ldots,\mu_d};\overline{\nu_1,\ldots,\nu_d}) = \mathbf{T}_1[\mu_1,\nu_1]\mathbf{T}_2[\mu_2,\nu_2]\ldots\mathbf{T}_d[\mu_d,\nu_d], \tag{4.30}$$

with $\eta, \kappa \in [d]$ and $\mathbf{T}_k[\mu_k, \nu_k] := \mathcal{T}_k[:, \mu_k, \nu_k, :], \mu_k, \nu_k \in \{1, 2\}, k \in [d]$. According to Theorem 4, Equation (4.30) can be written as follows:

$$\left(\mathbf{A}_\eta^* \mathbf{A}_\kappa\right)(\overline{\mu_1, \ldots, \mu_d}; \overline{\nu_1, \ldots, \nu_d}) = \mathbf{T}_1^{(0,t_1)}[\mu_1, \nu_1]\mathbf{T}_2^{(t_1,t_1+t_2)}[\mu_2, \nu_2] \ldots \mathbf{T}_d^{(\sum_{l=1}^{d-1} t_l, \sum_{l=1}^{d} t_l)}[\mu_d, \nu_d], \tag{4.31}$$

with $\sum_{l=1}^{d} t_l = 0$ and where $t_k = q_2(\mu_k) - q_2(\nu_k)$ with q_2 being defined in Equation (1.118) and $\mathbf{T}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k]$ are the block matrices within the TT-cores $\mathbf{T}_k[\mu_k, \nu_k]$ with $j_{k-1} + t_k = j_k$, $k \in [d]$, as illustrated in Figure 4.2. In this case, for a single 1-body term we have $\mathbf{T}_k[\mu_k, \nu_k] = \mathbf{T}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k]$. There are scalar values defined as follows: for $j_{k-1} = \sum_{l=1}^{k-1} t_l$, $j_k = j_{k-1} + t_k$, we have:

$$\mathbf{T}_k[\mu_k, \nu_k] = \mathbf{T}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k] = \begin{cases} \{\delta_{\mu_k,\nu_k}, \delta_{\mu_k,\nu_k}\delta_{\nu_k,1}\} & \text{if } \eta = \kappa \\ \{\delta_{\mu_k,\nu_k}, \delta_{\mu_k,\nu_k+1}, \delta_{\mu_k,\nu_k-1}\} & \text{otherwise} \end{cases} \tag{4.32}$$

with $\eta, \kappa \in [d]$. Here, we must point out that for $k \in [d-1]$, $j_{k-1}, j_k \in \{-1, 0, 1\}$ as per Equation (4.38). This corresponds to $p = 1$, indicating the number of p-body interactions, in this case being equal to 1 due to the presence of only 1-body terms.

The following example provides a better clarification on the construction of $\mathbf{T}_k[\mu_k, \nu_k] = \mathbf{T}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k]$:

Example 4.3.1. Let $d = 2$ be the number of spin-orbital basis functions. Initially, we consider the case $\eta = \kappa = 1$. Let $(\mathcal{T}_1, \mathcal{T}_2)$ be the block-sparse TTO decomposition of the 1-body term $\mathbf{A}_1^*\mathbf{A}_1$. According to (4.27), for $\mu_1, \nu_1, \mu_2, \nu_2 \in \{1, 2\}$ we have:

$$(\mathbf{A}_1^*\mathbf{A}_1)(\overline{\mu_1, \mu_2}; \overline{\nu_1, \nu_2}) = \underbrace{\delta_{\mu_1,\nu_1}\delta_{\nu_1,1}}_{\mathbf{T}_1[\mu_1,\nu_1]} \underbrace{\delta_{\mu_2,\nu_2}}_{\mathbf{T}_2[\mu_2,\nu_2]}, \tag{4.33}$$

For $\mu_1 = \nu_1 = \mu_2 = \nu_2 = 1$, we have

$$(\mathbf{A}_1^*\mathbf{A}_1)(\overline{1, 1}; \overline{1, 1}) = \underbrace{1}_{\mathbf{T}_1[1,1]} \underbrace{1}_{\mathbf{T}_2[1,1]}, \tag{4.34}$$

Here we set $\mathbf{T}_1[1, 1] = \mathbf{T}_1^{(0,0)}[1, 1]$ and $\mathbf{T}_2[1, 1] = \mathbf{T}_2^{(0,0)}[1, 1]$ with $t_1 = t_2 = 0$.

Now, let us consider the case $\eta = 1 < \kappa = 2$ and let $(\mathcal{T}_1, \mathcal{T}_2)$ be the TTO decomposition of the 1-body term $\mathbf{A}_1^*\mathbf{A}_2$. According to (4.28), we have:

$$(\mathbf{A}_1^*\mathbf{A}_2)(\overline{\mu_1, \mu_2}; \overline{\nu_1, \nu_2}) = \underbrace{(-1)^{\nu_1}\delta_{\mu_1,\nu_1+1}}_{\mathbf{T}_1[\mu_1,\nu_1]} \underbrace{\delta_{\mu_2,\nu_2-1}}_{\mathbf{T}_2[\mu_2,\nu_2]}. \tag{4.35}$$

For $\mu_1 = 2, \nu_1 = 1, \mu_2 = 1, \nu_2 = 2$,

$$(\mathbf{A}_1^*\mathbf{A}_2)(\overline{2, 1}; \overline{1, 2}) = \underbrace{-1}_{\mathbf{T}_1[2,1]} \underbrace{1}_{\mathbf{T}_2[1,2]}. \tag{4.36}$$

with $\mathbf{T}_1[2, 1] = \mathbf{T}_1^{(0,1)}[2, 1]$ and $\mathbf{T}_2[2, 1] = \mathbf{T}_2^{(1,0)}[1, 2]$ such that $t_1 = 1$, $t_2 = -1$.

We note that in some cases (see Example 4.3.1, Equations (4.35),(4.36)) we need to multiply the first TT-core by a factor of -1.

In the following, we present **Algorithm** 6 for constructing the TTO decomposition of a single 1-body term of the form $\mathbf{A}_\eta^*\mathbf{A}_\kappa, \eta, \kappa \in [d]$, where *BSTT* stands for *Block-Sparse Tensor Train*.

Algorithm 6 Create 1-body TT-cores

Input: pairs $(\eta, \kappa), \eta, \kappa \in [d]$.

Output: Block-sparse TTO decomposition $(\mathcal{T}_1, \ldots, \mathcal{T}_d)$ with block matrices $\mathbf{T}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k], \mu_k, \nu_k \in \{1,2\}$, $j_{k-1}, j_k \in \{-1,0,1\}$, $k \in [d]$.

procedure BSTT_ONEBODY
  if $\eta = \kappa$ **then**
    find pairs $(\mu_k, \nu_k), \mu_k, \nu_k \in \{1,2\}, k \in [d]$ such that

$$\delta_{\mu_1,\nu_1} \ldots \delta_{\mu_\eta,\nu_\eta} \delta_{\nu_\eta,1} \ldots \delta_{\mu_d,\nu_d} \neq 0$$

  end if
  if $\eta < \kappa$ **then**
    find pairs $(\mu_k, \nu_k), \mu_k, \nu_k \in \{1,2\}, k \in [d]$ such that

$$(-1)^{\nu_\eta + \ldots + \nu_{\kappa-1}} \delta_{\mu_1,\nu_1} \ldots \delta_{\mu_\eta,\nu_\eta+1} \ldots \delta_{\mu_\kappa,\nu_\kappa-1} \ldots \delta_{\mu_d,\nu_d} \neq 0$$

  end if
  if $\kappa > \eta$ **then**
    find pairs $(\mu_k, \nu_k), \mu_k, \nu_k \in \{1,2\}, k \in [d]$ such that

$$(-1)^{1+\nu_\eta + \ldots + \nu_{\kappa-1}} \delta_{\mu_1,\nu_1} \ldots \delta_{\mu_\kappa,\nu_\kappa-1} \ldots \delta_{\mu_\eta,\nu_\eta+1} \ldots \delta_{\mu_d,\nu_d} \neq 0$$

  end if
  Construct $\mathbf{T}_1^{(0,t_1)}[\mu_1, \nu_1], \ldots, \mathbf{T}_k^{(\sum_{l=1}^{k-1} t_l, \sum_{l=1}^{k} t_l)}[\mu_k, \nu_k], \ldots, \mathbf{T}_d^{(\sum_{l=1}^{d-1} t_l, \sum_{l=1}^{d} t_l)}[\mu_d, \nu_d]$, ▷ $t_l = q_2(\mu_l) - q_2(\nu_l)$, $j_{k-1} = \sum_{l=1}^{k-1} t_l$, $j_k = \sum_{l=1}^{k} t_l$, and $j_k, j_{k-1} \in \{-1,0,1\}$.
end procedure

Now, let $(\widetilde{T}_1, \ldots, \widetilde{T}_d)$ be the block-sparse TTO decomposition of the 1-body operator $\mathbf{H}_{(1)}$, that is derived by summing individual TTO decompositions $(\mathcal{T}_1, \ldots, \mathcal{T}_d)$ from 1-body terms, refer to Equation (4.30), and by employing the additive property among TTs, as explained in Proposition 1.2.2. Suppose that $\widetilde{\mathbf{T}}_k \in \mathbb{R}^{R_{k-1}^{\mathbf{H}_{(1)}} \times 2 \times 2 \times R_k^{\mathbf{H}_{(1)}}}$ for $k \in \{2, \ldots, d-1\}$, and $\widetilde{T}_1 \in \mathbb{R}^{1 \times 2 \times 2 \times R_1^{\mathbf{H}_{(1)}}}$ and $\widetilde{T}_d \in \mathbb{R}^{R_{d-1}^{\mathbf{H}_{(1)}} \times 2 \times 2 \times 1}$ such that $R_0^{\mathbf{H}_{(1)}} = R_d^{\mathbf{H}_{(1)}} = 1$. According to Theorem 4 and as illustrated in Figure 4.2, the TT-cores can be arranged in a block-sparse structure. For fixed $\mu_k, \nu_k \in \{1,2\}$, $\widetilde{\mathbf{T}}_k[\mu_k, \nu_k] := \widetilde{\mathbf{T}}_k[:, \mu_k, \nu_k, :] \in \mathbb{R}^{R_{k-1}^{\mathbf{H}_{(1)}} \times R_k^{\mathbf{H}_{(1)}}}$ has block structure. These blocks are denoted by $\widetilde{\mathbf{T}}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k] \in \mathbb{R}^{\rho_{k-1,j_{k-1}}^{\mathbf{H}_{(1)}} \times \rho_{k,j_k}^{\mathbf{H}_{(1)}}}$, $\mu_k, \nu_k \in \{1,2\}$ for $k \in [d]$, $\rho_{0,0}^{\mathbf{H}_{(1)}} = \rho_{d,0}^{\mathbf{H}_{(1)}} = 1$ and for $k \in \{2, \ldots, d-1\}$, we have:

$$R_{k-1} = \sum_{j_{k-1} \in S_{k-1}^{\mathbf{H}_{(1)}}} \rho_{k-1,j_{k-1}}^{\mathbf{H}_{(1)}}, \; R_k = \sum_{j_k \in S_k^{\mathbf{H}_{(1)}}} \rho_{k,j_k}^{\mathbf{H}_{(1)}}, \tag{4.37}$$

where

$$S_k^{\mathbf{H}_{(1)}} = \min\{-\min\{k, d-k, 1\}, \ldots, \min\{k, d-k, 1\}\} = \{-1, 0, 1\}. \tag{4.38}$$

The set $S_k^{\mathbf{H}_{(1)}}$ is obtained according to Equation (4.38), by considering the case where $p = 1$, with p representing the number of p-body interactions; here it is equal to 1, due to the presence of only 1-body terms. This TTO has the following structure: for $k \in \{2, \ldots, d-1\}$

$$\widetilde{\mathbf{T}}_k[1,1] = \begin{pmatrix} \rho_{k,-1}^{\mathbf{H}_{(1)}} & \rho_{k,0}^{\mathbf{H}_{(1)}} & \rho_{k,1}^{\mathbf{H}_{(1)}} \\ \widetilde{\mathbf{T}}_k^{(-1,-1)}[1,1] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{T}}_k^{(0,0)}[1,1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{T}}_k^{(1,1)}[1,1] \end{pmatrix} \begin{matrix} \\ \rho_{k-1,-1}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,0}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,1}^{\mathbf{H}_{(1)}} \end{matrix}, \; \widetilde{\mathbf{T}}_k[2,2] = \begin{pmatrix} \rho_{k,-1}^{\mathbf{H}_{(1)}} & \rho_{k,0}^{\mathbf{H}_{(1)}} & \rho_{k,1}^{\mathbf{H}_{(1)}} \\ \widetilde{\mathbf{T}}_k^{(-1,-1)}[2,2] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{T}}_k^{(0,0)}[2,2] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{T}}_k^{(1,1)}[2,2] \end{pmatrix} \begin{matrix} \\ \rho_{k-1,-1}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,0}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,1}^{\mathbf{H}_{(1)}} \end{matrix} \tag{4.39}$$

$$\widetilde{\mathbf{T}}_k[1,2] = \begin{pmatrix} \rho_{k,-1}^{\mathbf{H}_{(1)}} & \rho_{k,0}^{\mathbf{H}_{(1)}} & \rho_{k,1}^{\mathbf{H}_{(1)}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \widetilde{\mathbf{T}}_k^{(0,-1)}[1,2] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{T}}_k^{(1,0)}[1,2] & \mathbf{0} \end{pmatrix} \begin{matrix} \\ \rho_{k-1,-1}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,0}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,1}^{\mathbf{H}_{(1)}} \end{matrix}, \; \widetilde{\mathbf{T}}_k[2,1] = \begin{pmatrix} \rho_{k,-1}^{\mathbf{H}_{(1)}} & \rho_{k,0}^{\mathbf{H}_{(1)}} & \rho_{k,1}^{\mathbf{H}_{(1)}} \\ \mathbf{0} & \widetilde{\mathbf{T}}_k^{(-1,0)}[2,1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{T}}_k^{(0,1)}[2,1] \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{matrix} \\ \rho_{k-1,-1}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,0}^{\mathbf{H}_{(1)}} \\ \rho_{k-1,1}^{\mathbf{H}_{(1)}} \end{matrix}. \tag{4.40}$$

For $k = 1$ respectively $k = d$,

$$\widetilde{T}_1[1,1] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{1,-1}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,0}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,1}} \\ \mathbf{0} & \widetilde{\mathbf{T}}_k^{(0,0)}[1,1] & \mathbf{0} \end{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{0,0}} \quad , \widetilde{T}_1[2,2] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{1,-1}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,0}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,1}} \\ \mathbf{0} & \widetilde{T}_1^{(0,0)}[2,2] & \mathbf{0} \end{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{0,0}} \quad , \tag{4.41}$$

$$\widetilde{T}_1[1,2] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{1,-1}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,0}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,1}} \\ \widetilde{T}_1^{(0,-1)}[1,2] & \mathbf{0} & \mathbf{0} \end{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{0,0}} \quad , \widetilde{T}_1[2,1] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{1,-1}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,0}} & \overset{\mathbf{H}_{(1)}}{\rho_{1,1}} \\ \mathbf{0} & \mathbf{0} & \widetilde{T}_1^{(0,1)}[2,1] \end{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{0,0}} \tag{4.42}$$

$$\widetilde{T}_d[1,1] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{d,0}} \\ \mathbf{0} \\ \widetilde{T}_d^{(0,0)}[1,1] \\ 0 \end{pmatrix} \begin{matrix} \overset{\mathbf{H}_{(1)}}{\rho_{d-1,-1}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,0}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,1}} \end{matrix} \quad , \widetilde{T}_d[2,2] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{d,0}} \\ \mathbf{0} \\ \widetilde{T}_d^{(0,0)}[2,2] \\ \mathbf{0} \end{pmatrix} \begin{matrix} \overset{\mathbf{H}_{(1)}}{\rho_{d-1,-1}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,0}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,1}} \end{matrix} \tag{4.43}$$

$$\widetilde{T}_d[1,2] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{d,0}} \\ \mathbf{0} \\ \mathbf{0} \\ \widetilde{T}_d^{(1,0)}[1,2] \end{pmatrix} \begin{matrix} \overset{\mathbf{H}_{(1)}}{\rho_{d-1,-1}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,0}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,1}} \end{matrix} \quad , \widetilde{T}_d[2,1] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{d,0}} \\ \widetilde{T}_d^{(-1,0)}[2,1] \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{matrix} \overset{\mathbf{H}_{(1)}}{\rho_{d-1,-1}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,0}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{d-1,1}} \end{matrix} \tag{4.44}$$

where $\overset{\mathbf{H}_{(1)}}{\rho_{0,0}} = \overset{\mathbf{H}_{(1)}}{\rho_{d,0}} = 1$. Let us consider the following example for further clarification.

Example 4.3.2. Let $d = 4$ be the number of spin-orbital basis functions. Consider the following sum over 1-body terms $\mathbf{S} = h_{11}\mathbf{A}_1^*\mathbf{A}_1 + h_{12}\mathbf{A}_1^*\mathbf{A}_2 + h_{43}\mathbf{A}_4^*\mathbf{A}_3$. Let $(\widetilde{\mathcal{T}}_1, \widetilde{\mathcal{T}}_2, \widetilde{\mathcal{T}}_3, \widetilde{\mathcal{T}}_4)$ be the TTO decomposition of $\mathbf{S} \in \mathbb{R}^{2^4 \times 2^4}$. Initially, we look at the matrix elements of $\mathbf{S}$:

$$\begin{aligned} \mathbf{S}(\overline{\mu_1,\mu_2,\mu_3,\mu_4};\overline{\nu_1,\nu_1,\nu_2,\nu_3,\nu_4}) &= h_{11}(\mathbf{A}_1^*\mathbf{A}_1)(\overline{\mu_1,\mu_2,\mu_3,\mu_4};\overline{\nu_1,\nu_1,\nu_2,\nu_3,\nu_4}) \\ &+ h_{12}(\mathbf{A}_1^*\mathbf{A}_2)(\overline{\mu_1,\mu_2,\mu_3,\mu_4};\overline{\nu_1,\nu_1,\nu_2,\nu_3,\nu_4}) \\ &+ h_{43}(\mathbf{A}_3^*\mathbf{A}_2)(\overline{\mu_1,\mu_2,\mu_3,\mu_4};\overline{\nu_1,\nu_1,\nu_2,\nu_3,\nu_4}) \\ &= h_{11}\delta_{\mu_1,\nu_1}\delta_{\nu_1,1}\delta_{\mu_2,\nu_2}\delta_{\mu_3,\nu_3}\delta_{\mu_4,\nu_4} + (-1)^{\nu_1}h_{12}\delta_{\mu_1,\nu_1+1}\delta_{\mu_2,\nu_2-1}\delta_{\mu_3,\nu_3}\delta_{\mu_4,\nu_4} \\ &+ (-1)^{1+\nu_3}h_{43}\delta_{\mu_1,\nu_1}\delta_{\mu_2,\nu_2}\delta_{\mu_3,\nu_3-1}\delta_{\mu_4,\nu_4+1}. \end{aligned} \tag{4.45}$$

According to the Proposition 1.2.2, each element $\mathbf{S}(\overline{\mu_1,\mu_2,\mu_3,\mu_4};\overline{\nu_1,\nu_1,\nu_2,\nu_3,\nu_4})$ is equal to:

$$\underbrace{\begin{bmatrix} h_{11}\delta_{\mu_1,\nu_1}\delta_{\nu_1,1} & (-1)^{\nu_1}h_{12}\delta_{\mu_1,\nu_1+1} & (-1)h_{43}\delta_{\mu_1,\nu_1} \end{bmatrix}}_{\widetilde{\mathbf{T}}_1[\mu_1,\nu_1]} \underbrace{\begin{bmatrix} \delta_{\mu_2,\nu_2} & & \\ & \delta_{\mu_2,\nu_2-1} & \\ & & \delta_{\mu_2,\nu_2} \end{bmatrix}}_{\widetilde{\mathbf{T}}_2[\mu_2,\nu_2]} \underbrace{\begin{bmatrix} \delta_{\mu_3,\nu_3} & & \\ & \delta_{\mu_3,\nu_3} & \\ & & \delta_{\mu_3,\nu_3-1} \end{bmatrix}}_{\widetilde{\mathbf{T}}_3[\mu_3,\nu_3]} \underbrace{\begin{bmatrix} \delta_{\mu_4,\nu_4} \\ \delta_{\mu_4,\nu_4} \\ \delta_{\mu_4,\nu_4+1} \end{bmatrix}}_{\widetilde{\mathbf{T}}_4[\mu_4,\nu_4]}. \tag{4.46}$$

Here, we rearrange the nonzero values within the TT-core $\widetilde{\mathbf{T}}_3[1,1]$, aligning them to adhere to the block-sparse structure. The same idea extends to the other TT-cores.

$$\widetilde{\mathbf{T}}_3[1,1] = \begin{pmatrix} \overset{\mathbf{H}_{(1)}}{\rho_{3,-1}} & \overset{\mathbf{H}_{(1)}}{\rho_{3,0}} & \overset{\mathbf{H}_{(1)}}{\rho_{3,1}} \\ \underbrace{0}_{\widetilde{\mathbf{T}}_3^{(-1,-1)}[1,1]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\widetilde{\mathbf{T}}_3^{(0,0)}[1,1]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \underbrace{0}_{\widetilde{\mathbf{T}}_3^{(2,1)}[1,1]} \end{pmatrix} \begin{matrix} \overset{\mathbf{H}_{(1)}}{\rho_{2,-1}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{2,0}} \\ \overset{\mathbf{H}_{(1)}}{\rho_{1,1}} \end{matrix}. \tag{4.47}$$

To summarize, constructing the block-sparse TTO decomposition of the 1-body operator $\mathbf{H}_{(1)}$ involves an initial step of building the TTO decomposition for individual 1-body terms, as detailed in **Algorithm** 6. Subsequently, the block-sparse TTO representations of these single 1-body terms are combined by concatenating their corresponding nonzero

blocks. This yields the final block-sparse TTO representation. However, the TTO-ranks will inevitably increase, enlarging the size of the blocks within the TT-cores. To manage this expansion, one can employ the TT-rounding approach outlined in **Algorithm**4.3.2. This method involves applying tSVD per block. Further elaboration will be provided in Section 4.3.2.

Block-sparse TTO decomposition of the 2-body operator

In what follows, we detail the process of obtaining a block-sparse TTO decomposition of the 2-body operator $\mathbf{H}_{(2)} \in \mathbb{R}^{2^d \times 2^d}$, see Equation (4.26). This is achieved through the sum of the TTO decompositions of 2-body terms of the form $\left(\mathbf{A}_\eta^* \mathbf{A}_\kappa\right)\left(\mathbf{A}_\gamma^* \mathbf{A}_\lambda\right)$ for $\eta, \kappa, \gamma, \lambda \in [d]$. The TTO decomposition of each 2-body term can be obtained through the multiplication of the TTO decomposition of two 1-body terms.

For fixed values of $\eta, \kappa, \gamma, \lambda \in [d]$, let $(\mathcal{T}_1, \ldots, \mathcal{T}_d)$ (resp. $(\mathcal{T}'_1, \ldots, \mathcal{T}'_d)$) be the TTO decomposition of $\mathbf{A}_\eta^* \mathbf{A}_\kappa$ (resp. $\mathbf{A}_\gamma^* \mathbf{A}_\lambda$), with $\mathcal{T}_k, \mathcal{T}'_k \in \mathbb{R}^{1\times 2\times 2\times 1}, k \in [d]$. Let us consider $(\mathcal{W}_1, \ldots, \mathcal{W}_d)$ the TTO decomposition of $\left(\mathbf{A}_\eta^* \mathbf{A}_\kappa\right)\left(\mathbf{A}_\gamma^* \mathbf{A}_\lambda\right)$. According to Proposition 1.3.2, each TT-core $\mathcal{W}_k, k \in [d]$ is expressed as, for $\mu_k, \nu_k \in \{1, 2\}$:

$$\mathbf{W}_k[\mu_k, \nu_k] = \sum_{z=1}^{2} (\mathbf{T}_k[\mu_k, z] \otimes_K \mathbf{T}'_k[z, \nu_k]), \tag{4.48}$$

where $\mathbf{T}_k[\mu_k, z] := \mathcal{T}_k[:, \mu_k, z, :], \mathbf{T}'_k[z, \nu_k] := \mathcal{T}'_k[:, z, \nu_k, :]$, and $\mathbf{W}_k[\mu_k, \nu_k] := \mathcal{W}[:, \mu_k, \mu_k, :]$, for $\nu_k, \mu_k, z \in \{1, 2\}$.

As previously explained, for a single 1-body term, the TT-cores $\mathbf{T}_k[\mu_k, z]$ and $\mathbf{T}'_k[z, \nu_k]$ are nonzero if they satisfy the following conditions (see Equation (4.32)):

$$\begin{aligned} \mathbf{T}_k[\mu_k, z] &= \mathbf{T}_k^{(j_{k-1}, j_k)}[\mu_k, z] \neq 0 \text{ if } j_k + t_k = j_{k-1}, \\ \mathbf{T'}_k[z, \nu_k] &= \mathbf{T'}_k^{(j'_{k-1}, j'_k)}[z, \nu_k] \neq 0 \text{ if } j'_k + t'_k = j'_{k-1}, \end{aligned} \tag{4.49}$$

where $t_k = q_2(\mu_k) - q_2(z)$, $t'_k = q_2(z) - q_2(\nu_k)$, and $j_{k-1}, j'_{k-1}, j_k, j'_k \in \{-1, 0, 1\}$, for $k \in \{2, \ldots, d-1\}$ and $j_0 = j'_0 = j_d = j'_d = 0$ otherwise.

It is clear that the matrices $\mathbf{W}_k[\mu_k, \nu_k]$ in Equation (4.48) can be arranged into a block-sparse. We denote the nonzero blocks of $\mathbf{W}_k[\mu_k, \nu_k]$ by $\mathbf{W}_k^{(\gamma_{k-1}, \gamma_k)}[\mu_k, \nu_k]$ where $\gamma_{k-1} = j_{k-1} + j'_{k-1}$ and $\gamma_k = j_k + j'_k$. In the case of the TTO decomposition of a single 2-body term, defined by the product $\left(\mathbf{A}_\eta^* \mathbf{A}_\kappa\right)\left(\mathbf{A}_\gamma^* \mathbf{A}_\lambda\right)$, we have $\mathbf{W}_k[\mu_k, \nu_k] := \mathbf{W}_k^{(\gamma_{k-1}, \gamma_k)}[\mu_k, \nu_k]$ such that

$$\mathbf{W}_k^{(\gamma_{k-1}, \gamma_k)}[\mu_k, \nu_k] = \sum_{z=1}^{2} \mathbf{T}_k^{(j_{k-1}, j_k)}[\mu_k, z] \otimes_K \mathbf{T'}_k^{(j'_{k-1}, j'_k)}[z, \nu_k] \neq 0 \text{ if } \gamma_{k-1} + t_k + t'_k = \gamma_k. \tag{4.50}$$

It can be showed that $\gamma_k \in \{-2, -1, 0, 1, 2\}$ for $k \in \{2, \ldots, d-2\}$, $\gamma_1, \gamma_{d-1} \in \{-1, 0, 1\}$ and $\gamma_0 = \gamma_d = 0$.

Now, let $(\widetilde{W}_1, \ldots, \widetilde{W}_d)$ be the block-sparse TTO decomposition of the 2-body operator $\mathbf{H}_{(2)}$, that is derived by summing individual TTO decompositions $(\mathcal{W}_1, \ldots, \mathcal{W}_d)$ from 2-body terms (refer to Equation (4.30)) and by employing the additive property among TTs, as explained in Proposition 1.2.2. We suppose that $\widetilde{W}_k \in \mathbb{R}^{R_{k-1}^{\mathbf{H}_{(2)}} \times 2\times 2\times R_k^{\mathbf{H}_{(2)}}}$ for $k \in \{2, \ldots, d-1\}$, and $\widetilde{W}_1 \in \mathbb{R}^{1\times 2\times 2\times R^{\mathbf{H}_{(2)}}}$ and $\widetilde{W}_d \in \mathbb{R}^{R_{d-1}^{\mathbf{H}_{(2)}} \times 2\times 2\times 1}$ such that $R_0^{\mathbf{H}_{(2)}} = R_d^{\mathbf{H}_{(2)}} = 1$. The TT- cores can be arranged in a block-sparse structure, such that for fixed $\mu_k, \nu_k \in \{1, 2\}$, $\widetilde{W}_k[\mu_k, \nu_k] := \widetilde{W}_k[:, \mu_k, \nu_k, :] \in \mathbb{R}^{R_{k-1}^{\mathbf{H}_{(2)}} \times R_k^{\mathbf{H}_{(2)}}}$ has a block structure. These blocks are denoted by $\widetilde{W}_k^{(\gamma_{k-1}, \gamma_k)}[\mu_k, \nu_k] \in \mathbb{R}^{\rho_{k-1, j_{k-1}}^{\mathbf{H}_{(2)}} \times \rho_{k, j_k}^{\mathbf{H}_{(2)}}}$, $\mu_k, \nu_k \in \{1, 2\}$ for $k \in [d]$, $\rho_{0,0}^{\mathbf{H}_{(2)}} = \rho_{d,0}^{\mathbf{H}_{(2)}} = 1$ and for $k \in \{2, \ldots, d-1\}$, we have:

$$R_{k-1}^{\mathbf{H}_{(2)}} = \sum_{\gamma_{k-1} \in S_{k-1}^{\mathbf{H}_{(2)}}} \rho_{k-1, \gamma_{k-1}}^{\mathbf{H}_{(2)}}, \; R_k^{\mathbf{H}_{(2)}} = \sum_{\gamma_k \in S_k^{\mathbf{H}_{(2)}}} \rho_{k, \gamma_k}^{\mathbf{H}_{(2)}}, \tag{4.51}$$

where

$$S_k^{\mathbf{H}_{(2)}} = \min\left\{-\min\left\{k, d-k, 2\right\}, \ldots, \min\left\{k, d-k, 2\right\}\right\}, \tag{4.52}$$

such that

$$S_k^{\mathbf{H}_{(2)}} = \{-2,-1,0,1,2\}\,, k \in \{3,\dots,d-2\} \text{ and } S_2^{\mathbf{H}_{(2)}} = S_{d-1}^{\mathbf{H}_{(2)}} = \{-1,0,1\}\,. \tag{4.53}$$

More generally, the block-sparse representation of the TT-cores $\widetilde{W}_k$ can be obtained as follows, for $k \in \{3,\dots,d-2\}$:

$$\widetilde{W}_k[1,1] = \begin{pmatrix} \overset{\mathbf{H}_{(2)}}{\rho_{k,-2}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,-1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,0}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,2}} & \\ \widetilde{W}_k^{(-2,-2)}[1,1] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-2}} \\ \mathbf{0} & \widetilde{W}_k^{(-1,-1)}[1,1] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-1}} \\ \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(0,0)}[1,1] & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,0}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(1,1)}[1,1] & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(2,2)}[1,1] & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,2}} \end{pmatrix}, \tag{4.54}$$

$$\widetilde{W}_k[2,2] = \begin{pmatrix} \overset{\mathbf{H}_{(2)}}{\rho_{k,-2}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,-1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,0}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,2}} & \\ \widetilde{W}_k^{(-2,-2)}[2,2] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-2}} \\ \mathbf{0} & \widetilde{W}_k^{(-1,-1)}[2,2] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-1}} \\ \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(0,0)}[2,2] & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,0}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(1,1)}[2,2] & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(2,2)}[2,2] & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,2}} \end{pmatrix}, \tag{4.55}$$

$$\widetilde{W}_k[1,2] = \begin{pmatrix} \overset{\mathbf{H}_{(2)}}{\rho_{k,-2}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,-1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,0}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,2}} & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-2}} \\ \widetilde{W}_k^{(-1,-2)}[1,2] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-1}} \\ \mathbf{0} & \widetilde{W}_k^{(0,-1)}[1,2] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,0}} \\ \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(1,0)}[1,2] & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(2,1)}[1,2] & 0 & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,2}} \end{pmatrix}, \tag{4.56}$$

$$\widetilde{W}_k[2,1] = \begin{pmatrix} \overset{\mathbf{H}_{(2)}}{\rho_{k,-2}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,-1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,0}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,1}} & \overset{\mathbf{H}_{(2)}}{\rho_{k,2}} & \\ \mathbf{0} & \widetilde{W}_k^{(-2,-1)}[2,1] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-2}} \\ \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(-1,0)}[2,1] & \mathbf{0} & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,-1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{W}_k^{(0,1)}[2,1] & \mathbf{0} & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,0}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \widetilde{W}_k^{(1,2)}[2,1] & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \overset{\mathbf{H}_{(2)}}{\rho_{k-1,2}} \end{pmatrix}. \tag{4.57}$$

This is the same structure as described in Section 3.3, where $p = 2$.

For additional clarity, we give the following example.

Example 4.3.3. Consider the matrix $\mathbf{S} \in \mathbb{R}^{2^d \times 2^d}$ along with its TTO decomposition $(\widetilde{\mathcal{T}}_1,\dots,\widetilde{\mathcal{T}}_4)$ as defined in Example 4.3.2. We want to derive the expression of the TTO decomposition of $\mathbf{S}^2$ which describes the sum over 2-body terms. Let $(\widetilde{\mathcal{W}}_1,\dots,\widetilde{\mathcal{W}}_4)$ be the TTO decomposition of $\mathbf{S}^2$. According to Proposition 1.3.2, each TT-core $\widetilde{\mathcal{W}}_k, k \in \{1,\dots,4\}$ is given, for $\mu_k, \nu_k \in \{1,2\}$ as:

$$\widetilde{\mathbf{W}}_k[\mu_k,\nu_k] = \sum_{z=1}^{2} \widetilde{\mathbf{T}}_k[\mu_k,z] \otimes_K \widetilde{\mathbf{T}}_k[z,\nu_k]. \tag{4.58}$$

We provide in (4.59) a partial matrix representation of $\widetilde{\mathbf{W}}_3[1,1]$. This matrix is block-

diagonal, with grouped blocks sharing the same labels $\gamma_1, \gamma_2 \in \{-2, -1, 0, 1, 2\}$.

$$\widetilde{\mathbf{W}}_3[1,1] = \begin{pmatrix} \underbrace{\widetilde{\mathbf{T}}_3^{(-1,-1)}[1,1] \otimes_K \widetilde{\mathbf{T}}_3^{(-1,-1)}[1,1]}_{=\widetilde{\mathbf{W}}_3^{(-2,-2)}[1,1]} & & & \\ & \underbrace{\begin{pmatrix} \widetilde{\mathbf{T}}_3^{(0,0)}[1,1] \otimes_K \widetilde{\mathbf{T}}_3^{(-1,-1)}[1,1] & \widetilde{\mathbf{T}}_3^{(0,-1)}[1,2] \otimes_K \widetilde{\mathbf{T}}_3^{(-1,0)}[2,1] \\ 0 & \widetilde{\mathbf{T}}_3^{(-1,-1)}[1,1] \otimes_K \widetilde{\mathbf{T}}_3^{(0,0)}[1,1] \end{pmatrix}}_{=\widetilde{\mathbf{W}}_3^{(-1,-1)}[1,1]} & & \\ & & \ddots & \\ & & & \ddots \\ & & & & \ddots \end{pmatrix}. \tag{4.59}$$

Block-sparse TTO decomposition of the Hamiltonian operator

Now, to construct the block-sparse TTO decomposition, denoted by $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ (see Figure 4.2), of the Hamiltonian operator $\mathbf{H} = \mathbf{H}_{(1)} + \mathbf{H}_{(2)}$ one needs to combine the TTO representations of 1-body and 2-body operators. This can be achieved using the **Algorithm** 7.

Algorithm 7 Addition of two block-sparse TTO decompositions

Input: Block-sparse TTO representations of the 1-body operator and the 2-body operator, denoted as $(\widetilde{\mathcal{T}}_1, \cdots, \widetilde{\mathcal{T}}_d)$, and $(\widetilde{\mathcal{W}}_1, \cdots, \widetilde{\mathcal{W}}_d)$ respectively.

Output: Compute the block-sparse TTO representations $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$.

1: **procedure** BSTT_SUM
2: **for** $k = 1$ to d **do**
3: **for** $\mu_k, \nu_k = 1$ to 2 **do** ▷ $t_k = q_2(\mu_k) - q_2(\nu_k)$
4: **for** $\gamma_{k-1} \in S_{k-1}^{\mathbf{H}_{(2)}}, j_{k-1} \in S_{k-1}^{\mathbf{H}_{(1)}}$ **do**
5: **if** $\gamma_{k-1} = j_{k-1}$ and $\gamma_k = \gamma_{k-1} + t_k = j_{k-1} + t_k$ **then**
6:
$$\mathbf{H}_k^{(\gamma_{k-1},\gamma_k)}[\mu_k, \nu_k] = \mathrm{diag}(\widetilde{\mathbf{T}}_k^{(j_{k-1},j_k)}[\mu_k, \nu_k], \widetilde{\mathbf{W}}_k^{(\gamma_{k-1},\gamma_k)}[\mu_k, \nu_k]), \tag{4.60}$$
7: **else**
8:
$$\mathbf{H}_k^{(\gamma_{k-1},\gamma_k)}[\mu_k, \nu_k] = \widetilde{\mathbf{W}}_k^{(\gamma_{k-1},\gamma_k)}[\mu_k, \nu_k]. \tag{4.61}$$
9: **end if**
10: **end for**Return $\mathbf{H}_k[\mu_k, \nu_k]$ as a list of nonzero blocks.
11: **end for**
12: **end for**
13: **end procedure**

Remark 4.3.1. In **Algorithm** 7, when $k = 1$ or $k = d$, instead of performing the operation described in line 6, one should carry out horizontal concatenation/stacking or vertical concatenation, respectively between blocks with matching row index $j_{k-1} = \gamma_{k-1}$ or column index $j_k = \gamma_k$ respectively, as depicted in Figure 4.3.

In Figure 4.3, we provide a comprehensive example of combining block-sparse TTO representations of 1-body operator (red illustration) with 2-body operator (blue illustration)

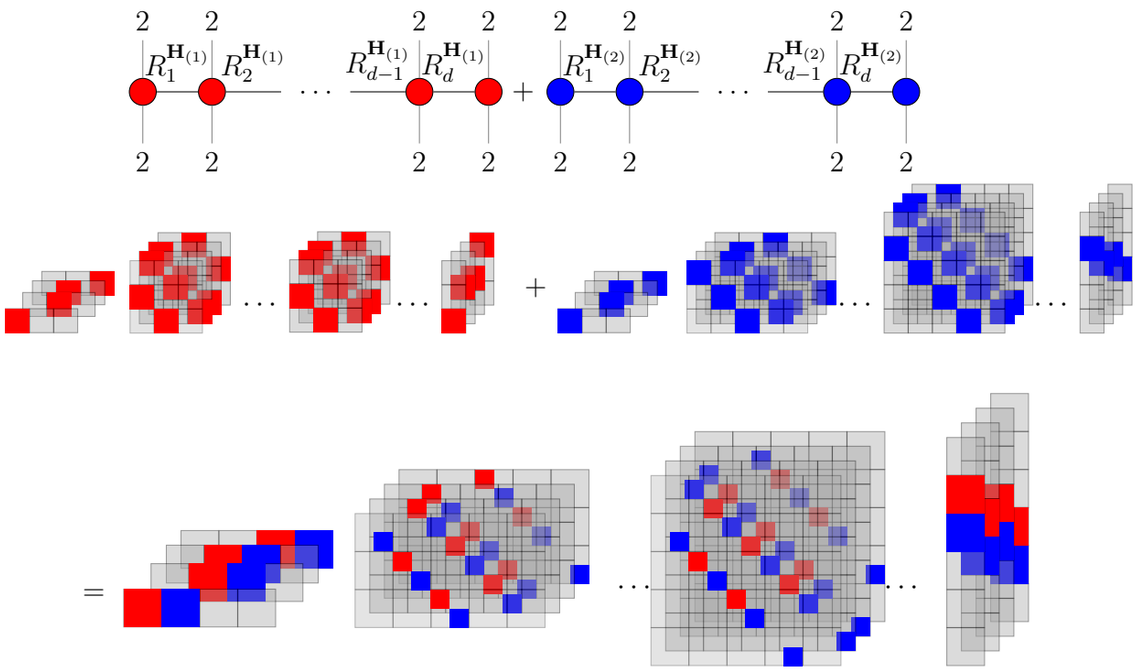


Figure 4.3: Addition of two block-sparse TT representations. The symbol + refers to summing two tensors in TTO formats, see Proposition 1.2.2.

While the TT-cores of the TTO decomposition of the operator initially exhibit a memory-efficient block structure that preserves the particle number, performing addition and product operations inevitably increases the ranks of these blocks. To ensure that the TTO-ranks per block remain reasonable, it becomes essential to apply compression during each summation of terms. This compression can be independently applied to blocks, see **Algorithm** 8, using tSVD with a predefined threshold.

Remark 4.3.2. Extension to a p-body Hamiltonian Operator A general p-body Hamiltonian operator as described in (3.46), with at most p-body terms, can be expressed in terms of the creation and annihilation operators in the spin-orbital basis as follows:

$$\mathbf{H}_{\leq p} = \sum_{J,J' \subseteq X} \mathbf{C}(J;J') \sum_{\boldsymbol{l}=1}^{2} \prod_{\boldsymbol{x} \in J} \mathbf{A}^*_{2(\boldsymbol{x}-1)+\boldsymbol{l}} \prod_{\boldsymbol{x'} \in J'} \mathbf{A}_{2(\boldsymbol{x'}-1)+\boldsymbol{l}}, \tag{4.62}$$

where $X = \{(\xi_1, 0, \ldots, 0), (\xi_1, \xi_2, 0, \ldots, 0), \ldots, (\xi_1, \xi_2, \ldots, \xi_p)\}$, $\boldsymbol{l} = (l_1, \ldots, l_p)$, $\boldsymbol{x} = (x_1, \ldots, x_p)$, $\xi_k \in [d]\,,\ k \in [p]$, and we define $\mathbf{A}_{2(\boldsymbol{x}-1)+\boldsymbol{l}}$ and $\mathbf{A}_{2(\boldsymbol{x'}-1)+\boldsymbol{l}}$ as follows:

$$\mathbf{A}_{2(\boldsymbol{x}-1)+\boldsymbol{l}} = \mathbf{A}_{2(x_1-1)+l_1} \ldots \mathbf{A}_{2(x_p-1)+l_p}, \tag{4.63}$$

$$\mathbf{A}_{2(\boldsymbol{x'}-1)+\boldsymbol{l}} = \mathbf{A}_{2(x'_p-1)+l_p} \ldots \mathbf{A}_{2(x'_1-1)+l_1}, \tag{4.64}$$

where $\mathbf{A}_{2(x_k-1)+l_k} = \mathbf{I}$, if $x_k = 0$ for $k \in [p]$ (same applies to $\mathbf{A}_{2(\boldsymbol{x'}-1)+\boldsymbol{l}}$). It is important to observe that in Equation (4.62), if the number of creation operators exceeds the number of annihilation operators (or vice versa), the resulting matrix product is zero. Now, our goal is to represent each term within the sum in (4.62) that characterizes p-body interactions as a sum of products of 1-body terms, similar to what we did for the 2-body term. For example, in (4.62), the p-body term can be obtained by taking $J = \overline{\xi_1, \xi_2, \ldots, \xi_p}$, $J' = \overline{\xi'_1, \xi'_2, \ldots, \xi'_p}$ as follows:

$$\mathbf{H}_{(p)} = \sum_{\substack{\xi_1,\ldots,\xi_p \\ \xi'_1,\ldots,\xi'_p=1}}^{d} \mathbf{C}(\overline{\xi_1, \ldots, \xi_p}; \overline{\xi'_1, \ldots, \xi'_p}) \sum_{l_1 \ldots l_p=1}^{2} \left(\mathbf{A}^*_{2(\xi_1-1)+l_1} \ldots \mathbf{A}^*_{2(\xi_p-1)+l_p} \mathbf{A}_{2(\xi'_p-1)+l_p} \ldots \mathbf{A}_{2(\xi'_1-1)+l_1} \right). \tag{4.65}$$

To relate this general expression to the 2-body Hamiltonian operator $\mathbf{H}$ defined in (1.138), we can write:

$$\mathbf{H} = \mathbf{H}_{\leq 2} = \mathbf{H}_{(1)} + \mathbf{H}_{(2)}, \tag{4.66}$$

where the matrix entries are related to one-electron and two-electron integrals as: $\mathbf{C}(\xi_1;\xi_1') = h_{\xi_1\xi_1'}$ and $\mathbf{C}(\overline{\xi_1,\xi_2};\overline{\xi_1',\xi_2'}) = \frac{1}{2}v_{\xi_1\xi_1'\xi_2\xi_2'}$. As explained in Section 3.3, the TTO representation of $\mathbf{H}_{(p)}$, denoted as $(\mathcal{H}_1,\ldots,\mathcal{H}_d)$, exhibits a block-sparse structure with a maximum of $2p+1$ nonzero blocks per TT-core. To construct a block-sparse representation for the p-body Hamiltonian, we can employ the same approach used for the 2-body operator $\mathbf{H}_{(2)}$. By exploiting the anti-commutation relations of the creation and annihilation operators, one obtains:

$$\begin{aligned}\mathbf{H}_{(p)} = &\sum_{\xi_1,\xi_1'}\sum_{\substack{\xi_k,\xi_k'\neq\xi_1,\xi_1'\\ k\in\{2,\ldots,p-1\}}}\mathbf{C}(\overline{\xi_1,\ldots,\xi_p};\overline{\xi_1',\ldots,\xi_p'})c_{\xi_k,\xi_k'}\mathbf{A}^*_{2(\xi_1-1)+l_1}\mathbf{A}_{2(\xi_1'-1)+l_1}\\
&+\sum_{\xi_1,\xi_1',\xi_2,\xi_2'}\sum_{\substack{\xi_k,\xi_k'\neq\xi_1,\xi_1',\xi_2,\xi_2'\\ k\in\{3,\ldots,p-2\}}}\mathbf{C}(\overline{\xi_1,\ldots,\xi_p};\overline{\xi_1',\ldots,\xi_p'})c_{\xi_k,\xi_k'}\mathbf{A}^*_{2(\xi_1-1)+l_1}\mathbf{A}_{2(\xi_2'-1)+l_2}\mathbf{A}^*_{2(\xi_1'-1)+l_2}\mathbf{A}_{2(\xi_2-1)+l_1}\\
&+\ldots+\sum_{\substack{\xi_1,\ldots,\xi_p\\ \xi_1',\ldots,\xi_p'=1}}^{d}\mathbf{C}(\overline{\xi_1,\ldots,\xi_p};\overline{\xi_1',\ldots,\xi_p'})\mathbf{A}^*_{2(\xi_1-1)+l_1}\mathbf{A}_{2(\xi_p-1)+l_p}\ldots\mathbf{A}^*_{2(\xi_p-1)+l_p}\mathbf{A}_{2(\xi_1'-1)+l_1}.\end{aligned} \tag{4.67}$$

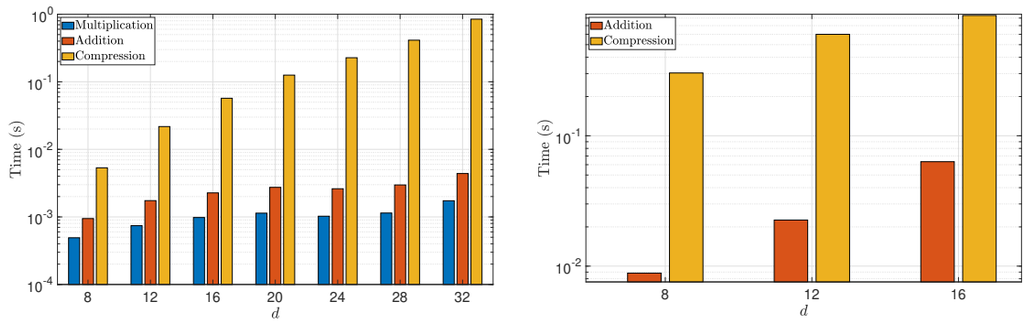
Here, $c_{\xi_k,\xi_k'}$ represents additional terms arising from the deltas in the anti-commutation relations. After expressing (4.65) in terms of 1-body terms, we employ a similar approach as described earlier to derive a TTO representation of $\mathbf{H}_{(p)}$ with a block-sparse structure. However, in this case, there are additional terms to incorporate into the summation.

Numerical results

Now, we provide numerical results concerning the construction of the block-sparse TTO representation of the Hamiltonian operator for various molecules with different number of spin-orbitals d.

We present in Figure 4.4 the mean time required to execute a single operation among the most dominant steps within a generic algorithm for constructing the TTO representation of the quantum chemical Hamiltonian operator, as defined in Equation (1.138). The x-axis indicates the number of spin-orbitals d, while the y-axis indicates the average time required for each operation. Figure 4.4(a) displays the construction of the block-sparse TTO decomposition, using **Algorithms** 6 and 7. The steps highlighted include the multiplication of TTOs in a block-sparse format for the construction of the 2-body operator, denoted as *Multiplication*, the addition of TTOs in a block-sparse format, referred to as *Addition*, and the TTO-rank reduction achieved through tSVD after each rounding step to maintain reasonable TTO-ranks, identified as *Compression*. Similarly, Figure 4.4(b) displays the time consumption associated with the naive construction of the TTO decomposition using Algorithm 2. It is noted that all calculations are carried out using single thread for a Hydrogen chain in the STO-3g basis set.

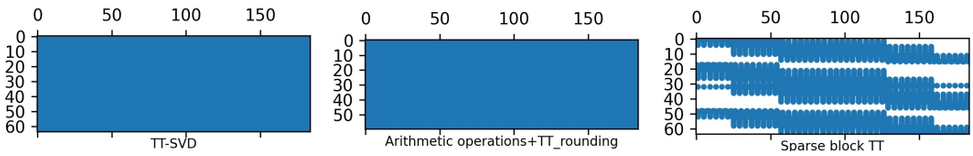
From Figure 4.4, we can notice that the computational time of the compression increases with the number of spin-orbitals d in both figures, which aligns with our expectations. As the number of orbitals grows, more terms need to be considered within the sums, resulting in the generation of larger blocks within the TT-cores that subsequently require compression. However, it is important to highlight a significant advantage provided by the block-sparse structure in this context. Due to this structure and the ability to perform operations such as SVD, addition, and multiplication on individual blocks, we observe a noteworthy acceleration in the execution of operations like addition and compression when compared to their counterparts in the naive construction method. Upon closer examination of the first three number of orbitals, for $d \in \{8, 12, 16\}$, it is noteworthy that the addition of two TTs exhibits significantly faster computation, nearly one order of magnitude, when using a block-structured TT, see Figure 4.4(a). This observation holds true for the compression process using the tSVD algorithm as well.



(a) (b)

Figure 4.4: Average time in seconds spent by the most time-consuming steps in the numerical construction of the TTO decomposition of the quantum chemical Hamiltonian operator as defined in (1.138). The left figure (Figure 4.4(a)), displays the execution time associated with the construction of the block-sparse TTO decomposition, whereas the right figure (Figure 4.4(b)), displays the execution time associated with the naive TTO decomposition. The chosen accuracy is $\epsilon = 1e - 12$.

Figures 4.5 and 4.6 provide a visual representation of the sparsity within the TT-cores when the particle number condition is satisfied. This sparsity results from enforcing the block-sparse structure in the TTO decomposition, as explained in previous sections. In contrast, when employing the naive methods, the TT-cores tend to become dense. In Figure 4.6, we present the sparsity within the TT-cores, illustrating the block-sparse TT decomposition of the eigenfunction of $\mathbf{H}$, as described in Section 3.3.1.



(a) (b) (c)

Figure 4.5: The TTO decomposition of $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$ for $d = 8$ is obtained through three methods: (1) using the TT-SVD algorithm (refer to **Algorithm** 1), (2) employing the sum of rank-1 TTOs+TT-rounding algorithm (see Section3.2.3), and (3) enforcing the block-sparse structure in the TT-cores. These approaches are represented in the left, middle, and right figures, respectively.

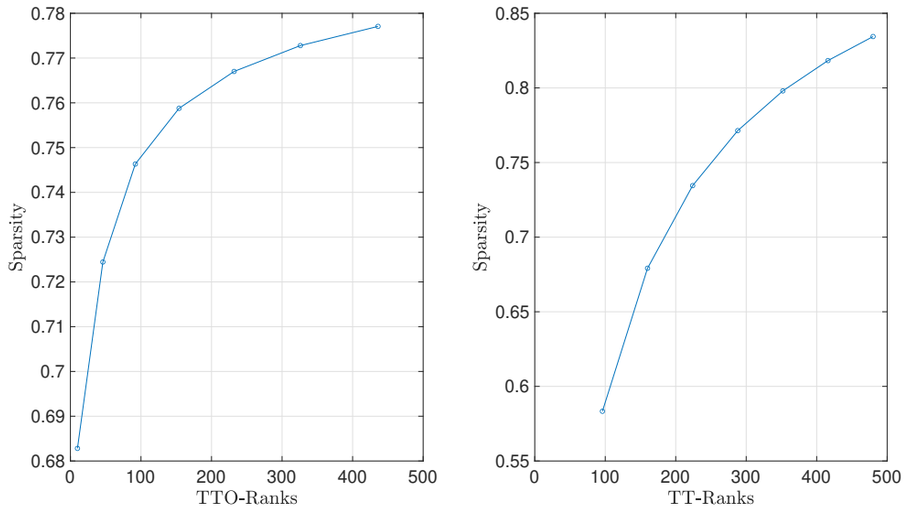


Figure 4.6: Average sparsity within TT-cores and TT-ranks. For each block-sparse TT-core, sparsity is measured as the ratio of zero elements to the total number of elements. The left figure depicts the sparsity within the TT-cores of the TTO decomposition while varying the TTO-ranks. The right figure showcases the sparsity within the TT-cores of the TT decomposition of the eigenfunction at different TT-ranks. Both illustrations are conducted for a Hydrogen chain, taking into account the block-sparse structure.

To properly put this construction of the block-sparse TTO decomposition into context, we conduct in Figure 4.7 a comparison with the state-of-the art tensor network software ITensor, which implements QC-DMRG. We focus here on a Hydrogen chain where $d \in \{4, \ldots, 20\}$. We begin by measuring the time needed to construct the TTO decomposition of the Hamiltonian operator while ensuring the preservation of particle number for both our method and ITensor. It is worth noting that while ITensor also yields block-sparse tensors, they are not stored in the same manner or follow the same structure as our method. We then compare these times with a straightforward naive construction approach that does not consider the block-sparse structure in the TT decompositions. Additionally, we verify whether the TTO-ranks obtained from both libraries align with the expected theoretical TTO-rank, which scales quadratically with the system's size d, i.e $\mathcal{O}(d^2)$, (for further insights on the derivation of this scaling, we direct the reader to [2, 17]). All benchmark tests are conducted using a single thread to maintain consistency and fairness in our evaluations.

It is noteworthy that when $d > 4$, our method exhibits a notable advantage, being approximately one order of magnitude faster than the naive construction method. However, it is also observed to be about one order of magnitude slower than ITensor's approach. ITensor's out-performance can be attributed to the use of a highly efficient functionality known as AutoMPO/OpSum, which allows for the efficient addition of sums of rank-1 TTs, see [28] for more details on this functionality. Nevertheless, it is essential to highlight that the construction of the TTO can be readily parallelized, as elucidated in Figure 4.13(a). This parallelization offers a promising avenue to speedup the TTO construction, which is a crucial preliminary step before beginning the QC-DMRG procedure. Additionally, from the Figure 4.7, we note that the obtained numerical TT-ranks are in accordance with the expected theoretical TTO-ranks.

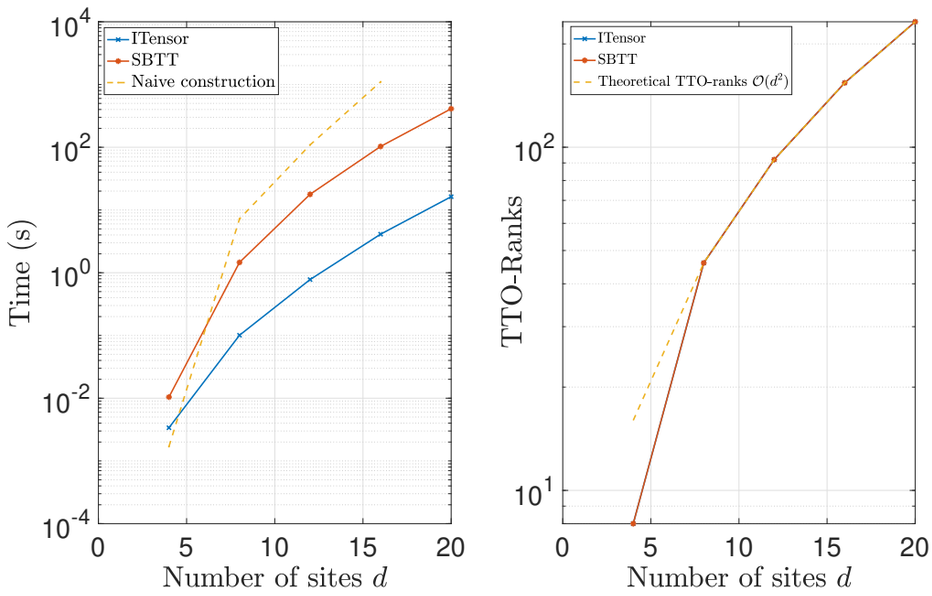


Figure 4.7: Comparison between three methods, ITensor, Sparse Block Tensor Train (SBTT), and the naive TTO construction, for computing the TTO decomposition of Hamiltonian operators within a Hydrogen chain in the STO-3g basis set.

In Figure 4.8, the left figure illustrates the variation of TTO-ranks R_k for each $k \in \{1, \ldots, 38\}$ while varying the compression threshold δ. These results are for the LiH molecule in the cc-pVDZ basis set, with $d = 38$ and $N = 4$ electrons. In the right figure, we present the corresponding relative error of the QC-DMRG energies to evaluate the impact of compression.

Notably, for this molecule, at around $\epsilon = 10^{-5}$ with a maximum rank of 534 (which is less than the theoretical rank of 781), we already achieve a relative error of 10^{-5}. This shows that while the exact representation (without compression, as introduced in Section 3.2.2) is advantageous in avoiding numerical errors that can arise from numerical compression, compression is also valuable for reducing TTO-ranks, which may not be straightforward to achieve with exact representations.

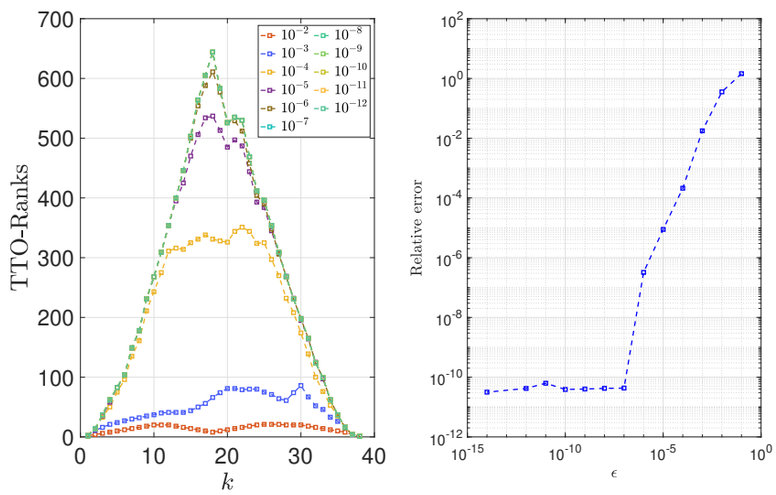


Figure 4.8: Left figure: TTO-ranks at each fixed iteration k for different accuracies ϵ. Right figure: relative error obtained for numerically finding the ground-state energy with QC-DMRG vs accuracy ϵ.

4.3.2 Basic operations on block-sparse structured TT representations:

Compression and orthogonalization with block-sparse structured TT representations

Given the block-sparse structure in the TT-cores of both the TTO decomposition of the Hamiltonian operator (refer to Figure 4.1) and the eigenfunction denoted as $\mathbf{\Psi}$ (refer to Figure 4.2), substantial reductions in computational costs can be realized. This proves particularly valuable during compression, orthogonalization, and contraction which are common operations in QC-DMRG calculations.

As outlined in the introductory section, reducing the TT-ranks of a tensor in TT-format can be achieved using TT-rounding, see Algorithm 2. This algorithm involves applying a tSVD to the matricization of the TT-cores. Orthogonalization is an integral part of the TT-rounding algorithm, and it serves also as a prerequisite for obtaining left-orthogonal or right-orthogonal TT-cores within the TT representation of $\mathbf{\Psi}$.

Let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ be the block-sparse TT decomposition of $\mathbf{\Psi} \in \mathbb{R}^{2^d}$, with $\mathcal{U}_k \in \mathbb{R}^{r_{k-1} \times 2 \times r_k}$, $k \in [d]$, $r_0 = r_d = 1$. Let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be the block-sparse TTO decomposition of $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$, with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times 2 \times 2 \times R_k}$, $k \in [d]$, $R_0 = R_d = 1$.

Here, we need to perform QR and SVD operations on the mode-(1:2) matricization or the mode-(1) matricization, of the TT-cores $\mathcal{U}_k, k \in [d]$, as well as the mode-(1:3) matricization or the mode-(1) matricization of the TT-cores $\mathcal{H}_k, k \in [d]$. Importantly, these operations can be optimized by exploiting the block structure present in the TT-cores.

Notably, the matricization of the block-sparse TT-cores results in a block-diagonal matrix form. The latter can be represented as follows, for $k \in [d]$:

$$\mathbf{U}_k^{<2>} = \operatorname{diag}(\mathbf{L}_k^1, \mathbf{L}_k^2, \ldots, \mathbf{L}_k^{n_k}), \tag{4.68}$$

and

$$\mathbf{U}_k^{(1)} = \operatorname{diag}(\mathbf{R}_k^1, \mathbf{R}_k^2, \ldots, \mathbf{R}_k^{n_{k-1}}), \tag{4.69}$$

where $\mathbf{U}_k^{<2>} \in \mathbb{R}^{2r_{k-1} \times r_k}$ is the mode-$(1:2)$ matricization of $\mathcal{U}_k \in \mathbb{R}^{r_{k-1} \times 2 \times r_k}$, $\mathbf{U}_k^{(1)} \in \mathbb{R}^{r_{k-1} \times 2r_k}$ is the mode-(1) matricization of $\mathcal{U}_k$, and

$$n_k = \min\{N+1, k+1\} - \max\{1, N-d+k+1\} + 1. \tag{4.70}$$

For each $\ell \in [n_k]$, there exist corresponding pairs (i_{k-1}, i_k); $i_k \in S_k^{\mathbf{\Psi}}$, $i_{k-1} \in S_{k-1}^{\mathbf{\Psi}}$, with $S_k^{\mathbf{\Psi}}, S_{k-1}^{\mathbf{\Psi}}$ being defined in (4.2), such that the block matrices $\mathbf{L}_k^\ell$ and $\mathbf{R}_k^\ell$ are given by:

$$\mathbf{L}_k^1 = \mathbf{U}_k^{(\max\{1,N-d+k\},\max\{1,N-d+k+1\})}[1], \quad \mathbf{L}_k^\ell = \begin{pmatrix} \mathbf{U}_k^{(i_{k-1},i_k)}[2] \\ \mathbf{U}_k^{(i_{k-1},i_k)}[1] \end{pmatrix} = \begin{pmatrix} \mathbf{U}_k^{(i_k-1,i_k)}[2] \\ \mathbf{U}_k^{(i_k,i_k)}[1] \end{pmatrix}, \tag{4.71}$$

and

$$\mathbf{R}_k^{n_k} = \mathbf{U}_k^{(\min\{N+1,k\},\min\{N+1,k+1\})}[1], \ \mathbf{R}_k^\ell = \begin{pmatrix} \mathbf{U}_k^{(i_{k-1},i_k)}[1] & \mathbf{U}_k^{(i_{k-1},i_k)}[2] \end{pmatrix} = \begin{pmatrix} \mathbf{U}_k^{(i_{k-1},i_{k-1})}[1] & \mathbf{U}_k^{(i_{k-1},i_{k-1}+1)}[2] \end{pmatrix} \tag{4.72}$$

Here $\mathbf{U}_k^{(i_{k-1},i_k)}[\mu_k] \in \mathbb{R}^{\rho_{k-1,i_{k-1}}^{\mathbf{\Psi}} \times \rho_{k-1,i_{k-1}}^{\mathbf{\Psi}}}$ are the blocks of the TT-cores $\mathbf{U}_k[\mu_k]$ with $\mu_k \in \{1,2\}$, and $i_{k-1} + q_2(\mu_k) = i_k$ for $k \in [d]$.

Let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be the block-sparse TTO decomposition of $\mathbf{H}$. The expression of the matricization of the TT-cores is given as follows, for $k \in [d]$:

$$\mathbf{H}_k^{<3>} = \operatorname{diag}(\tilde{\mathbf{L}}_k^1, \tilde{\mathbf{L}}_k^2, \ldots, \tilde{\mathbf{L}}_k^{m_k}). \tag{4.73}$$

and

$$\mathbf{H}_k^{(1)} = \operatorname{diag}(\tilde{\mathbf{R}}_k^1, \tilde{\mathbf{R}}_k^2, \ldots, \tilde{\mathbf{R}}_k^{m_k}). \tag{4.74}$$

where $m_k = 2\beta_k + 1$, with $\beta_k = \min\{k, d-k, 2\}$. Here, for $\ell \in [m_k]$, $\tilde{\mathbf{L}}_k^\ell$ represents the concatenation of blocks within the TT-cores sharing the same column-index $j_k' \in S_k^{\mathbf{H}}$, $S_k^{\mathbf{H}}$ is defined in (4.6) for the mode-$(1:3)$ matricization of $\mathcal{H}_k$. $\tilde{\mathbf{R}}_k^\ell$ represents the concatenation

of blocks within the TT-cores, sharing the same row-index $j'_{k-1} \in S^{\mathbf{H}}_{k-1}$ for the mode-(1) matricization of $\mathcal{H}_k$.

Algorithm 8, outlines the main steps to achieve a left-orthogonal block-sparse TT decomposition of $\mathbf{\Psi}$.

Algorithm 8 Left-orthogonalization of block-sparse TT representation

Input: block-sparse TT representations $(\mathcal{U}_1, \cdots, \mathcal{U}_d)$

Output: Compute the block-sparse TT representation $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ with left-orthogonal TT-cores.

1: **procedure** BSTT_QR
2: **for** $k = 1$ to $d-1$ **do**
3:

$$\mathrm{QR}(\underbrace{\mathbf{U}_k^{<2>}}_{\in \mathbb{R}^{2r_{k-1} \times r_k}}) = \mathrm{QR}(\begin{bmatrix} \mathbf{L}_k^1 & & & \\ & \mathbf{L}_k^2 & & \\ & & \ddots & \\ & & & \mathbf{L}_k^{n_k} \end{bmatrix}) = \begin{bmatrix} \mathbf{Q}^1 & & & \\ & \mathbf{Q}^2 & & \\ & & \ddots & \\ & & & \mathbf{Q}^{n_k} \end{bmatrix} \begin{bmatrix} \mathbf{R}^1 & & & \\ & \mathbf{R}^2 & & \\ & & \ddots & \\ & & & \mathbf{R}^{n_k} \end{bmatrix}.$$

4: $\mathbf{U}_k^{<2>} := \mathrm{diag}(\mathbf{Q}^1, \mathbf{Q}^2, \cdots, \mathbf{Q}^{n_k})$.
5: **for** $j = 1$ to n_k **do**
6: $\mathbf{R}^j_{k+1} := \mathbf{R}^j \mathbf{R}^j_{k+1}$. ▷ Update the blocks $\mathbf{R}^j_{k+1}$, see (4.69).
7: **end for**
8: $\mathbf{U}^{(1)}_{k+1} := \mathrm{diag}(\mathbf{R}^1_{k+1}, \ldots, \mathbf{R}^{n_k}_{k+1})$.
9: **end for**
10: **end procedure**

This algorithm can also be extended to the TT-cores of the TTO decomposition of the Hamiltonian operator, assuming prior knowledge of the matricization of the block-sparse structure, see Equations (4.73) and (4.74).

Remark 4.3.3. (Time complexity) One can estimate the time complexity of the step 3 in **Algorithm** 8, by making certain assumptions. For simplification, assume that all the blocks in the TT-core at iteration k are of equal sizes with $n \leq (N+1)$ being the number of blocks, i.e $r = n\tilde{\rho}^{\mathbf{\Psi}}$ with $\tilde{\rho}^{\mathbf{\Psi}} = \max_{i_{k-1} \in S^{\mathbf{\Psi}}_{k-1}, i_k \in S^{\mathbf{\Psi}}_k} \left\{ \rho^{\mathbf{\Psi}}_{k-1,i_{k-1}}, \rho^{\mathbf{\Psi}}_{k,i_k} \right\}$, where $S^{\mathbf{\Psi}}_{k-1}, S^{\mathbf{\Psi}}_k$ are defined in Equations (4.2). Under these assumptions, the computational time needed for executing a block-sparse QR at step 3 scales as $\mathcal{O}\left(\frac{r^3}{n^2}\right)$. This implies performing QR decomposition on blocks n times. Additionally, when it comes to compression, a similar approach can be adopted. Here, one simply applies a tSVD with a defined threshold δ, instead of a QR factorization which gives as well an estimation of the computational time of $\mathcal{O}(\frac{r^3}{n^2})$ under the same assumptions.

Remark 4.3.4. In order to perform a right-to-left orthogonalization, an analogous approach is employed by iterating backward from $k = d$ to $k = 2$ and using LQ factorization instead of QR.

Remark 4.3.5. While the operations we have discussed so far are encountered in 1-site QC-DMRG calculations, as explained in the introductory section, it is important to note that to take advantage of rank adaptivity, one can perform 2-site QC-DMRG. In 2-site QC-DMRG, optimization within each micro-step occurs between consecutive TT-cores of the TT representation of $\mathbf{\Psi}$. This involves an additional operation in each micro-step, which is the contraction product between consecutive TT-cores of the TT representation of $\mathbf{\Psi}$ and the contraction product between consecutive TT-cores of the TTO decomposition.

Exploiting the block-sparse structure within the TT-cores results in a similar block-sparse structure within the newly contracted TT-cores. This new representation allows all the operations described earlier to be seamlessly performed on the newly obtained block-sparse contracted TT-cores. To illustrate this, consider the contraction product between two block-sparse TT-cores from the TT representation of $\mathbf{\Psi}$ at fixed indices k and $k+1$, $k \in [d-1]$, which produces a new TT-core, denoted as $\mathcal{U}_{k,k+1} = \mathcal{U}_k \times_3 \mathcal{U}_{k+1} \in \mathbb{R}^{r_{k-1} \times 2 \times 2 \times r_{k+1}}$.

Let $\mathbf{U}_{k,k+1}[w_k] := \mathcal{U}_{k,k+1}[:, \mu_k, \nu_k, :] \in \mathbb{R}^{r_{k-1} \times r_{k+1}}$, with $w_k = \mu_k + 2(\nu_k - 1) \in \{1, 2, 3, 4\}$, such that

$$\mathbf{U}_{k,k+1}[w_k] = \mathbf{U}_k[\mu_k]\mathbf{U}_{k+1}[\nu_k]. \tag{4.75}$$

The matrix $\mathbf{U}_{k,k+1}[w_k] \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,j_{k-1}} \times \rho^{\mathbf{\Psi}}_{k+1,j_{k+1}}}$ has a block-sparse structure, such that each block is expressed as:

$$\mathbf{U}^{(j_{k-1},j_{k+1})}_{k,k+1}[w_k] = \sum_{j_k \in S^{\mathbf{\Psi}}_k} \mathbf{U}^{(j_{k-1},j_k)}_k[\mu_k]\mathbf{U}^{(j_k,j_{k+1})}_{k+1}[\nu_k], \tag{4.76}$$

with $j_{k-1} \in S^{\mathbf{\Psi}}_{k-1}, j_{k+1} \in S^{\mathbf{\Psi}}_{k+1}$ and $S^{\mathbf{\Psi}}_k$ is defined in (4.2). Here, we provide the matrix representations of $\mathbf{U}^{(j_{k-1},j_{k+1})}_{k,k+1}[w_k]$ for $w_k \in \{1, 2, 3, 4\}$, assuming that the matrices $\mathbf{U}_k[\mu_k]$ and $\mathbf{U}_{k+1}[\nu_k]$ each have $N + 1$ blocks, where N is the number of particles:

$$\mathbf{U}_{k,k+1}[1] = \begin{pmatrix} \mathbf{U}^{(1,1)}_{k,k+1}[1] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{(2,2)}_{k,k+1}[1] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{U}^{(N+1,N+1)}_{k,k+1}[1] \end{pmatrix}, \mathbf{U}_{k,k+1}[2] = \begin{pmatrix} \mathbf{0} & \mathbf{U}^{(1,2)}_{k,k+1}[2] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{U}^{(N,N+1)}_{k,k+1}[2] \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}, \tag{4.77}$$

$$\mathbf{U}_{k,k+1}[3] = \begin{pmatrix} \mathbf{0} & \mathbf{U}^{(1,2)}_{k,k+1}[3] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{U}^{(N,N+1)}_{k,k+1}[3] \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}, \mathbf{U}_{k,k+1}[4] = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{U}^{(1,3)}_{k,k+1}[4] & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & & \mathbf{U}^{(N-1,N+1)}_{k,k+1}[4] \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{4.78}$$

Likewise, a block-sparse structure can be obtained by contracting consecutive TT-cores of the TTO decomposition of the Hamiltonian operator, i.e performing $\mathcal{H}_{k,k+1} = \mathcal{H}_{k-1} \times_4 \mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times 2 \times 2 \times 2 \times 2 \times R_{k+1}}$.

Tensor contractions with block-sparse structured TT representations

One of the fundamental operations during QC-DMRG sweeps, using the TT representation of both the operator and the eigenfunction, involves constructing the left and right components, referred to as $\mathcal{G}^{\leq k-1}$ and $\mathcal{G}^{>k-1}$, as depicted in Figure 1.11 in the context of 1-site QC-DMRG. In what follows, we aim to derive a block-sparse representation of these contraction products by exploiting the structures of the TT-cores. This allows us to design a scheme to perform these contractions effectively, as they are frequently encountered in QC-DMRG calculations.

We recall the expressions of $\mathcal{G}^{\leq k-1}$ and $\mathcal{G}^{>k-1}$ as introduced in Definition 25. Let $(\mathcal{H}_1, \cdots, \mathcal{H}_d)$ be the TTO representation of the Hamiltonian operator with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times 2 \times 2 \times R_k}$, $R_0 = R_d = 1$. Let $(\mathcal{U}_1, \cdots, \mathcal{U}_d)$ be the TT representation of the eigenfunction, with $\mathcal{U}_k \in \mathbb{R}^{r_{k-1} \times 2 \times r_k}, r_0 = r_d = 1$. Let $\text{vec}\left(\mathcal{G}^{\leq k-1}\right) \in \mathbb{R}^{r^2_{k-1} R_{k-1}}$ (resp. $\text{vec}\left(\mathcal{G}^{>k-1}\right) \in \mathbb{R}^{r^2_k R_k}$) be the vectorization of $\mathcal{G}^{\leq k-1}$ (resp. $\mathcal{G}^{>k-1}$) such that

$$\text{vec}\left(\mathcal{G}^{\leq k-1}\right) = \mathbf{Z}_1 \cdots \mathbf{Z}_{k-1} \in \mathbb{R}^{r^2_{k-1} R_{k-1}}, \tag{4.79}$$

$$\text{vec}\left(\mathcal{G}^{>k-1}\right) = \mathbf{Z}_{k+1} \cdots \mathbf{Z}_d \in \mathbb{R}^{r^2_k R_k}, \tag{4.80}$$

where $\mathbf{Z}_k \in \mathbb{R}^{r^2_{k-1} R_{k-1} \times r^2_k R_k}$ are matrices defined as follows, see Equation (1.166):

$$\mathbf{Z}_k = \sum_{\mu_k=1}^{2} \sum_{\nu_k=1}^{2} \mathbf{U}_k[\mu_k] \otimes_K \mathbf{H}_k[\mu_k, \nu_k] \otimes_K \mathbf{U}_k[\nu_k]. \tag{4.81}$$

Each TT-core $\mathcal{U}_k$ and $\mathcal{H}_k$ exhibits a block-sparse structure, with each block satisfying the following conditions:

$$\begin{aligned} \mathbf{U}^{(i_{k-1},i_k)}_k[\mu_k] &\neq \mathbf{0} \text{ if } i_{k-1} + q_2(\mu_k) = i_k, \\ \mathbf{U}^{(i'_{k-1},i'_k)}_k[\mu_k] &\neq \mathbf{0} \text{ if } i'_{k-1} + q_2(\nu_k) = i'_k, \\ \mathbf{H}^{(j_{k-1},j_k)}_k[\mu_k, \nu_k] &\neq \mathbf{0} \text{ if } j_{k-1} + (q_2(\mu_k) - q_2(\nu_k)) = j_k, \end{aligned} \tag{4.82}$$

where $i_k, i'_k \in S_k^{\mathbf{\Psi}}$, and $j_k \in S_k^{\mathbf{H}}$. We have also $\sum_{i_k \in S_k^{\mathbf{\Psi}}} \rho_{k,i_k}^{\mathbf{\Psi}} = \sum_{i'_k \in S_k^{\mathbf{\Psi}}} \rho_{k,i'_k}^{\mathbf{\Psi}} = r_k$, and $\sum_{j_k \in S_k^{\mathbf{H}}} \rho_{k,j_k}^{\mathbf{H}} = R_k$.

Equation (4.82) indicates that $\mathbf{Z}_k$ represents a block-sparse matrix subject to specific constraints on its block entries. Here, we define ℓ_{k-1} and ℓ_k as follows, for $k \in [d-1]$:

$$\ell_{k-1} = j_{k-1} + i_{k-1} + i'_{k-1}, \tag{4.83}$$
$$\ell_k = j_k + i_k + i'_k. \tag{4.84}$$

such that $\ell_{k-1} \in S_{k-1}^{\mathbf{Z}_{k-1}}$ and $\ell_{k-1} \in S_{k-1}^{\mathbf{Z}_k}$, where for $k \in \{2, \ldots, d-1\}$, $S_{k-1}^{\mathbf{Z}_{k-1}}$ and $S_{k-1}^{\mathbf{Z}_k}$ can be defined as:

$$\begin{aligned} S_{k-1}^{\mathbf{Z}_{k-1}} &:= \{-\beta_{k-1} + 2\max\{1, N-d+k\}, \ldots, \beta_{k-1} + 2\min\{N+1, k\}\}, \\ S_k^{\mathbf{Z}_k} &:= \{-\beta_k + 2\max\{1, N-d+k+1\}, \ldots, \beta_k + 2\min\{N+1, k+1\}\}, \end{aligned} \tag{4.85}$$

with β_k being defined in (4.6), $\ell_0 = 2$ and $\ell_d = 2(N+1)$.

We denote the nonzero blocks of $\mathbf{Z}_k$ as $\mathbf{Z}_k^{(\ell_{k-1},\ell_k)}$. The blocks are determined by grouping together, via concatenation, the matrices sharing identical indices $\ell_{k-1} = i_{k-1} + i'_{k-1} + j_{k-1}$ and $\ell_k = i_k + i'_k + j_k$. These matrices have the following expression:

$$\sum_{\mu_k=1}^{2}\sum_{\nu_k=1}^{2}\left(\mathbf{U}_k^{(i_{k-1},i_k)}[\mu_k] \otimes_K \mathbf{H}_k^{(j_{k-1},j_k)}[\mu_k,\nu_k] \otimes_K \mathbf{U}_k^{(i'_{k-1},i'_k)}[\nu_k]\right) \in \mathbb{R}^{(\rho_{k-1,i_{k-1}}^{\mathbf{\Psi}}\rho_{k-1,i'_{k-1}}^{\mathbf{\Psi}}\rho_{k-1,j_{k-1}}^{\mathbf{H}})\times(\rho_{k,i_k}^{\mathbf{\Psi}}\rho_{k,i'_k}^{\mathbf{\Psi}}\rho_{k,j_k}^{\mathbf{H}})}. \tag{4.86}$$

We refer the reader to Example 4.3.4, Equation (4.89) for more clarification on the generation of the blocks $\mathbf{Z}_k^{(\ell_{k-1},\ell_k)}$.

Now, according to the conditions outlined in Equation (4.82), we have:

$$\mathbf{Z}_k^{(\ell_{k-1},\ell_k)} \neq \mathbf{0} \text{ if } \ell_k - \ell_{k-1} = 2\eta, \eta \in \{1,2\}. \tag{4.87}$$

We provide in (4.88) the block-sparse matrix illustration of $\mathbf{Z}_k \in \mathbb{R}^{r_{k-1}^2 R_{k-1} \times r_k^2 R_k}$:

$$\mathbf{Z}_k = \begin{pmatrix} \blacksquare & \blacksquare & & & & & & & \\ & \blacksquare & \blacksquare & & & & & & \\ & & \blacksquare & \blacksquare & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & & \blacksquare & & & & \\ & & & & & \ddots & & & \\ & & & & & & \blacksquare & & \\ & & & & & & & \ddots & \\ & & & & & & & & \ddots \end{pmatrix}. \tag{4.88}$$

Remark 4.3.6. The variable *diagonal_number* in the *DiagonalQuadraticForm* struct, as illustrated in Appendix .6, refers to either the first diagonal in (4.88) (*diagonal_number* = 1) or the second diagonal (*diagonal_number* = 2).

Example 4.3.4. Consider the contraction product between the TT-cores $\mathcal{U}_k$ and $\mathcal{H}_k$ at iteration $k = 2$, obtained from a TT representation for a system with $d = 4$ spin-orbitals, $N = 2$ particles, and $p = 2$. According to Equation (4.86), the matrix $\mathbf{Z}_2$ takes the following form:

$$\mathbf{Z}_2 = \begin{pmatrix} \mathbf{Z}_2^{(1,1)} & & \mathbf{Z}_2^{(1,3)} & & \\ & \ddots & & \ddots & \\ & & \ddots & & \\ & & & \mathbf{Z}_2^{(5,5)} & \mathbf{Z}_2^{(5,7)} \end{pmatrix},$$

and $\mathbf{Z}_2^{(1,1)}$, for example, can be constructed by concatenating matrices sharing the same indices $\ell_1 = 1$ and $\ell_2 = 1$, as follows:

$$\begin{aligned}\mathbf{Z}_2^{(1,1)} = \Big[&\sum_{\mu_k=1}^{2}\sum_{\nu_k=1}^{2} \left(\mathbf{U}_2^{(1,1)}[\mu_k] \otimes_K \mathbf{H}_2^{(-1,-1)}[\mu_k,\nu_k] \otimes_K \mathbf{U}_2^{(1,1)}[\nu_k]\right) \\ &\sum_{\mu_k=1}^{2}\sum_{\nu_k=1}^{2} \left(\mathbf{U}_2^{(1,1)}[\mu_k] \otimes_K \mathbf{H}_2^{(-1,-2)}[\mu_k,\nu_k] \otimes_K \mathbf{U}_2^{(1,2)}[\nu_k]\right) \\ &\sum_{\mu_k=1}^{2}\sum_{\nu_k=1}^{2} \left(\mathbf{U}_2^{(1,2)}[\mu_k] \otimes_K \mathbf{H}_2^{(-1,-2)}[\mu_k,\nu_k] \otimes_K \mathbf{U}_2^{(1,1)}[\nu_k]\right) \Big].\end{aligned} \tag{4.89}$$

It can be showed that the maximum number of nonzero blocks that we can obtain within matrices $\mathbf{Z}_k, k \in [d]$ is $4(N+1)+2p-2$.

Now, given this block-sparse representation of matrices $\mathbf{Z}_k$, we want to design a contraction scheme to compute left (resp. right) parts of $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ as defined in (4.79) (resp. $\text{vec}\left(\mathcal{G}^{>k-1}\right)$ as defined in (4.81)). For the left component, defined as $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$, we can express it differently, using the nonzero blocks of $\mathbf{Z}_k$ as follows: we denote the nonzero blocks of $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ by $\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})}$, such that

$$\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})} = \sum_{\ell_1 \in S_1^{\mathbf{Z}_1}} \sum_{\ell_2 \in S_2^{\mathbf{Z}_2}} \cdots \sum_{\ell_{k-2} \in S_{k-2}^{\mathbf{Z}_{k-2}}} (\mathbf{Z}_1^{(\ell_0,\ell_1)} \mathbf{Z}_2^{(\ell_1,\ell_2)} \ldots \mathbf{Z}_{k-1}^{(\ell_{k-2},\ell_{k-1})}). \tag{4.90}$$

By construction and according to Equation (4.87), we can deduce that for $k \in \{2,\ldots,d\}$:

$$\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})} \neq \mathbf{0}, \text{ if } \ell_{k-1} \in S_{k-1}^{\mathbf{Z}_{k-1}} \cap \{\ell_0, \ell_0+2, \ell_0+4, \ldots, \ell_0+2(k-1)\}. \tag{4.91}$$

Thus, $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ can be represented as a block-sparse vector with a maximum of k blocks. According to the expression of ℓ_0, see Equation (4.83), we have $\ell_0 = 2$. In a similar way, we can write down the right part $\text{vec}\left(\mathcal{G}^{>k-1}\right)$ as follows:

$$\left(\text{vec}\left(\mathcal{G}^{>k-1}\right)\right)^{(\ell_k,\ell_d)} \neq \mathbf{0}, \text{ if } \ell_k \in S_k^{\mathbf{Z}_k} \cap \{\ell_d, \ell_d-2, \ell_d-4, \ldots, \ell_d-2(d-k)\}. \tag{4.92}$$

To provide a clearer understanding, we present an example illustrating the construction of the left part $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ at a fixed iteration k. The same example also extends to the right part.

Example 4.3.5. Let $d = 4$, $N = 2$, and $p = 2$. The representation of $\text{vec}\left(\mathcal{G}^{\leq d}\right)$, according to Equations (4.79) and (4.86), can be expressed as:

$$\text{vec}\left(\mathcal{G}^{\leq d}\right) = \begin{pmatrix}\mathbf{Z}_1^{(2,2)} & \mathbf{Z}_1^{(2,4)}\end{pmatrix} \begin{pmatrix} \mathbf{Z}_2^{(1,1)} & & \mathbf{Z}_2^{(1,3)} & & \\ & \ddots & & \ddots & \\ & & \ddots & & \\ & & & \mathbf{Z}_2^{(5,5)} & \mathbf{Z}_2^{(5,7)} \end{pmatrix} \begin{pmatrix} \mathbf{Z}_3^{(2,2)} & & \mathbf{Z}_3^{(2,4)} & & \\ & \ddots & & \ddots & \\ & & & & \mathbf{Z}_3^{(6,8)} \\ & & & \ddots & \\ & & & & \mathbf{Z}_3^{(8,8)} \end{pmatrix} \begin{pmatrix} & \\ \mathbf{Z}_4^{(4,6)} & \\ & \\ \mathbf{Z}_4^{(6,6)} & \end{pmatrix}. \tag{4.93}$$

The construction of $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ can be performed using block matrix multiplications. For example, at $k = 3$:

$$\begin{aligned}\text{vec}\left(\mathcal{G}^{\leq 2}\right) &= \begin{pmatrix}\mathbf{Z}_1^{(2,2)} & \mathbf{Z}_1^{(2,4)}\end{pmatrix} \begin{pmatrix} \mathbf{Z}_2^{(1,1)} & & \mathbf{Z}_2^{(1,3)} & & \\ & \mathbf{Z}_2^{(2,2)} & & \mathbf{Z}_2^{(2,4)} & \\ & & \ddots & & \ddots \\ & & & \mathbf{Z}_2^{(4,4)} & & \mathbf{Z}_2^{(4,6)} \\ & & & & \mathbf{Z}_2^{(5,5)} & & \mathbf{Z}_2^{(5,7)} \end{pmatrix} \\ &= \begin{bmatrix} \underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 2}\right)\right)^{(2,2)}}_{\mathbf{Z}_1^{(2,2)}\mathbf{Z}_2^{(2,2)}} & \underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 2}\right)\right)^{(2,4)}}_{\mathbf{Z}_1^{(2,2)}\mathbf{Z}_2^{(2,4)}+\mathbf{Z}_1^{(2,4)}\mathbf{Z}_2^{(4,4)}} & \underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 2}\right)\right)^{(2,6)}}_{\mathbf{Z}_1^{(2,4)}\mathbf{Z}_2^{(4,6)}} \end{bmatrix}.\end{aligned} \tag{4.94}$$

As you may have noticed, in Equation (4.94), we selectively used only the red nonzero blocks for the multiplication to form vec $\left(\mathcal{G}^{\leq 2}\right)$. When constructing vec $\left(\mathcal{G}^{\leq 3}\right)$, we use the result obtained for vec $\left(\mathcal{G}^{\leq 2}\right)$ and apply it to $\mathbf{Z}_3$. This approach extends to all iterations where $k \in \{4, \cdots, d\}$. For each iteration, specific blocks from the matrices $\mathbf{Z}_{k-1}$ are required to build the left part vec $\left(\mathcal{G}^{\leq k-1}\right)$. The same methodology applies when computing the right part. It is important to note that, in practice, we do not store these block matrices formed through Kronecker products, except when required.

Algorithm 9 gives the formal procedure to perform a left to right contraction product with block-sparse TT representations. The final result vec $\left(\mathcal{G}^{\leq k}\right)$, $k \in [d]$, is stored as a list of nonzero blocks.

Algorithm 9 Left to right contraction product

Input: block-sparse TT decomposition of both the Hamiltonian operator $(\mathcal{H}_1, \cdots, \mathcal{H}_d)$ and eigenfunction $(\mathcal{U}_1, \cdots, \mathcal{U}_d)$, iteration number k

Output: vec $\left(\mathcal{G}^{\leq k-1}\right)$.

Initialization: vec $\left(\mathcal{G}^{\leq 1}\right) = \left(\mathbf{Z}_1^{(2,2)} \quad \mathbf{Z}_1^{(2,4)}\right)$, $list_1 = S_1^{\mathbf{Z}_1} \cap \{\ell_0, \ell_0 + 2\}$.

procedure ContractionProductLR
  for m=2 to k-1 **do**
    $list_m = S_m^{\mathbf{Z}_m} \cap \{\ell_0, \ell_0 + 2, \ell_0 + 4, \ldots, \ell_0 + 2m\}$.
    for n in $list_m$ **do**
      $\left(\text{vec}\left(\mathcal{G}^{\leq m}\right)\right)^{(\ell_0, n)} = \sum_{j \in list_{m-1}} \left(\text{vec}\left(\mathcal{G}^{\leq m-1}\right)\right)^{(\ell_0, j)} \mathbf{Z}_m^{(j,n)}$.
    end for
    vec $\left(\mathcal{G}^{\leq m}\right) = \left[\left(\text{vec}\left(\mathcal{G}^{\leq m}\right)\right)^{(\ell_0, \ell_0)}, \left(\text{vec}\left(\mathcal{G}^{\leq m}\right)\right)^{(\ell_0, \ell_0+2)}, \ldots, \left(\text{vec}\left(\mathcal{G}^{\leq m}\right)\right)^{(\ell_0, \ell_0+2m)} \right]$.
  end for
end procedure

Remark 4.3.7. (Time complexity) For a fixed value of $k, k \in [d]$, the time complexity required to compute the left part denoted by vec $\left(\mathcal{G}^{\leq k-1}\right)$ scales as:

$$\mathcal{O}\left(\frac{d}{\bar{n}^2 \bar{m}}\left(\frac{r^3 R}{\bar{n}} + \frac{r^2 R^2}{\bar{m}}\right)\right). \tag{4.95}$$

This can be derived by making some assumptions. Let $\bar{m}$ be the minimum number of blocks in the TT-cores of the TTO decomposition of the Hamiltonian operator, such that $\bar{m} \leq 2p + 1$. Assume that all the blocks per TT-core are of the same size, i.e $R = \bar{m} \max_{k \in [d]} \max_{j_k \in S_k^{\mathbf{H}}} \left\{\rho_{k,j_k}^{\mathbf{H}}\right\}$. Additionally, let $\bar{n}$ be the minimum number of blocks in the TT-cores of the TT decomposition of the eigenfunction, such that $\bar{n} \leq N + 1$ and assume that all blocks per TT-core are of the same size, i.e $r = \bar{n} \max_{k \in [d]} \max_{i_k \in S_k^{\mathbf{\Psi}}} \left\{\rho_{k,i_k}^{\mathbf{H}}\right\}$. At iteration k, each block in the block-sparse vector vec $\left(\mathcal{G}^{\leq k-1}\right)$ can be evaluated recursively as follows:

$$\begin{aligned}
\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0, \ell_{k-1})} &= \sum_{\ell_{k-2} \in S_{k-1}^{\mathbf{Z}_{k-2}} \cap \{\ell_0, \ldots, \ell_0 + 2(k-2)\}} \left(\text{vec}\left(\mathcal{G}^{\leq k-2}\right)\right)^{(\ell_0, \ell_{k-2})} \mathbf{Z}_{k-1}^{(\ell_{k-2}, \ell_{k-1})} \\
&= \sum_{\mu_{k-1}=1}^{2} \sum_{\nu_{k-1}=1}^{2} \sum_{\substack{j_{k-2} \in S_{k-2}^{\mathbf{H}}, i_{k-2}, i'_{k-2} \in S_{k-2}^{\mathbf{\Psi}} \\ j_{k-2} + i_{k-2} + i'_{k-2} \in S_{k-1}^{\mathbf{Z}_{k-2}} \cap \{\ell_0, \ldots, \ell_0 + 2(k-2)\}}} \left(\text{vec}\left(\mathcal{G}^{\leq k-2}\right)\right)^{(\ell_0, j_{k-2} + i_{k-2} + i'_{k-2})} \\
&\left(\mathbf{U}_{k-1}^{(i_{k-2}, i_{k-1})}[\mu_{k-1}] \otimes_K \mathbf{H}_{k-1}^{(j_{k-2}, j_{k-1})}[\mu_{k-1}, \nu_{k-1}] \otimes_K \mathbf{U}_{k-1}^{(i'_{k-2}, i'_{k-1})}[\nu_{k-1}]\right),
\end{aligned} \tag{4.96}$$

where we replaced ℓ_{k-1} by $\ell_{k-1} = i_{k-1} + i'_{k-1} + j_{k-1}$ for a given set of indices $(i_{k-1}, i'_{k-1}, j_{k-1})$.

Equation (4.96) comprises multiple terms involving the product of a vector with a series of Kronecker products of matrices. By assuming that $\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0, \ell_{k-1})} \in \mathbb{R}^{\frac{r^2 R}{\bar{m}\bar{n}^2}}$, the time complexity for evaluating each term in the sum of Equation (4.96) scales as $\mathcal{O}\left(\frac{1}{\bar{n}^2 \bar{m}}\left(\frac{r^3 R}{\bar{n}} + \frac{r^2 R^2}{\bar{m}}\right)\right)$. This complexity is achieved through the efficient application of multiplication techniques involving vectors and a series of Kronecker products of matrices, as elaborated upon in Section 2.3.4.

Considering that we have k nonzero blocks $\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})}$ to construct the entire left part $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$, the total time complexity can be in order of $\mathcal{O}\left(\frac{d}{\bar{n}^2\bar{m}}\left(\frac{r^3R}{\bar{n}}+\frac{r^2R^2}{\bar{m}}\right)\right)$.

Remark 4.3.8. Note that the variables *left_tensors* respectively *right_tensors*, as depicted in Appendix .6, correspond to a list of $\text{vec}\left(\mathcal{G}^{\leq k-1}\right), k \in [d]$, respectively list of $\text{vec}\left(\mathcal{G}^{>k}\right), k \in \{2,\ldots,d\}$. Each block within $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$, denoted as $\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})}$ corresponds to a variable of type **BlockContractions**. This variable is characterized by its *diagonal_number*, representing its position in the list, and an associated array of vectors. Further details about this array of vectors will be explained in Remark 4.3.9.

Numerical results

Figure 4.9 displays the efficiency of our block-sparse contraction scheme. We employ **Algorithm** 9 to perform a left-to-right contraction product, to evaluate $\text{vec}\left(\mathcal{G}^{\leq d}\right)$, which results in a scalar. This computation is then compared against a naive contraction, which does not take into account the block sparsity, using the optimized Julia library for tensor contractions, known as *TensorOperations* [23].

In this context, it is important to note that we keep the TTO-rank constant $R = 326$, with $R = \max_{1\leq k\leq d}\{R_k\}$, while allowing the TT-rank of the TT representation of the eigenfunction to vary, indicated as $r = \max_{1\leq k\leq d}\{r_k\}$. Notably, as the TT-Rank r exceeds 130, we observe that the block-sparse tensor contractions outperforms the direct contraction approach by approximately one order of magnitude. Additionally, it demonstrates alignment with the expected theoretical computational complexity. This numerical result is conducted for the H_{12} molecule with $d = 24$, $N = 12$ in the STO-3g basis. All numerical results are conducted using a single thread.

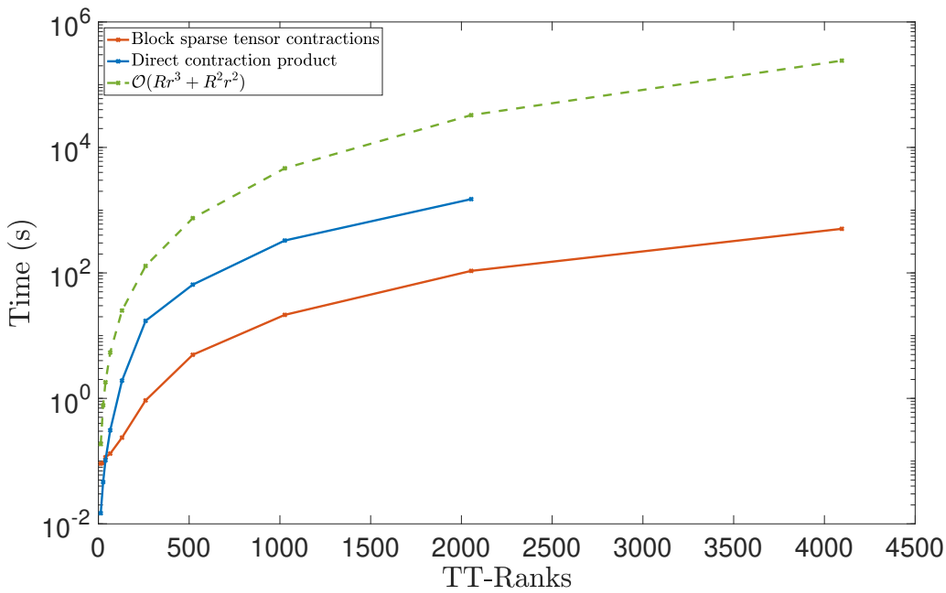


Figure 4.9: Comparison between execution times for block-sparse tensor contractions and direct contraction products across different TT-ranks of the TT representation of the eigenfunction.

Remark 4.3.9. In practice, the blocks in the block-sparse vector $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ are constructed by concatenating multiple vectors resulting from Kronecker products between different blocks of the TT representation of the operator and the eigenfunction, sharing the same index ℓ_{k-1}. Indeed, each block in $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ has the following form:

$$\left(\text{vec}\left(\mathcal{G}^{\leq k-1}\right)\right)^{(\ell_0,\ell_{k-1})} := \left[\boldsymbol{v}_1^L \ \boldsymbol{v}_2^L \ \ldots \ \boldsymbol{v}_{n_{\ell_{k-1}}}^L\right], \tag{4.97}$$

where $\ell_{k-1} \in \{\ell_0, \ell_0+2, \ldots, \ell_0+2(k-1)\}$. Here, $n_{\ell_{k-1}}$ refers to the number of nonzero vectors $\boldsymbol{v}_i^L \in \mathbb{R}^{\rho^{\boldsymbol{\Psi}}_{k-1,i_{k-1}}\rho^{\mathbf{H}}_{k-1,j_{k-1}}\rho^{\boldsymbol{\Psi}}_{k-1,i'_{k-1}}}$ for $i \in \left[n_{\ell_{k-1}}\right]$ satisfying $j_{k-1}+i_{k-1}+i'_{k-1} = \ell_{k-1}$.

For more clarification, we give the following example.

Example 4.3.6. Let $d = 4$, $N = 2$ and $k = 2$. According to Equation (4.91), $\text{vec}\left(\mathcal{G}^{\leq 1}\right)$ is a block-sparse vector that can be viewed as a list of blocks as the following:

$$\text{vec}\left(\mathcal{G}^{\leq 1}\right) := \left[\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,2)} \ \left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,4)}\right], \tag{4.98}$$

where $\ell_0 = 2, \ell_1 = 2, \ell_2 = 4$. Now, if we focus on the block $\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,2)}$, we need to find all indices (i_1, j_1, i_1') satisfying $i_1 + j_1 + i_1' = 2$, for $i_1, i_1' \in S_1^{\mathbf{\Psi}}$ and $j_1 \in S_1^{\mathbf{H}}$. It can be verified that $(i_1, j_1, i_1') \in \{(2, -1, 1), (1, -1, 2), (1, 0, 1)\}$, such that $\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,2)}$ is stored as a list of blocks:

$$\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,2)} := \left[\underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,2-1+1)}}_{\boldsymbol{v}_1^L} \ \underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,1-1+2)}}_{\boldsymbol{v}_2^L} \ \underbrace{\left(\text{vec}\left(\mathcal{G}^{\leq 1}\right)\right)^{(2,1+0+1)}}_{\boldsymbol{v}_3^L}\right], \tag{4.99}$$

such that according to Equations (4.86) and (4.90), (showing only the expression of $\boldsymbol{v}_1^L \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{1,i_1} \times \rho^{\mathbf{H}}_{1,j_1} \times \rho^{\mathbf{\Psi}}_{1,i_1'}}$), we have:

$$\begin{aligned} \boldsymbol{v}_1^L &:= \sum_{\mu_1=1}^{2} \sum_{\nu_1=1}^{2} \left(\mathbf{U}_1^{(1,2)}[\mu_1] \otimes_K \mathbf{H}_1^{(0,-1)}[\mu_1, \nu_1] \otimes_K \mathbf{U}_1^{(1,1)}[\nu_1]\right) \\ &= \mathbf{U}_1^{(1,2)}[2] \otimes_K \mathbf{H}_1^{(0,-1)}[1, 2] \otimes_K \mathbf{U}_1^{(1,1)}[1]. \end{aligned} \tag{4.100}$$

Similarly, for the right part, each block in $\text{vec}\left(\mathcal{G}^{>k-1}\right)$ has the following form:

$$\left(\text{vec}\left(\mathcal{G}^{>k-1}\right)\right)^{(\ell_k, \ell_d)} := \left[\boldsymbol{v}_1^R \ \boldsymbol{v}_2^R \ \ldots \ \boldsymbol{v}_{n_{\ell_k}}^R\right]^*, \tag{4.101}$$

where $\ell_k \in \{\ell_d, \ell_d - 2, \ldots, \ell_d - 2(d-k)\}$. Here, n_{ℓ_k} refers to the number of nonzero vectors $\boldsymbol{v}_i^R \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k,i_k} \rho^{\mathbf{H}}_{k,j_k} \rho^{\mathbf{\Psi}}_{k,i_k'}}$ for $i \in [n_{\ell_k}]$, satisfying $\ell_k = j_k + i_k + i_k'$.

Within QC-DMRG calculations, as we will elaborate on in the subsequent section, especially for the matrix-vector operations within the eigensolver, our attention is directed toward the tensorized representation of these vectors, designated as $\mathcal{V}_i^L \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{H}}_{k-1,j_{k-1}} \times \rho^{\mathbf{\Psi}}_{k-1,i_{k-1}'}}$ and $\mathcal{V}_i^R \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k,i_k} \times \rho^{\mathbf{H}}_{k,j_k} \times \rho^{\mathbf{\Psi}}_{k,i_k'}}$.

Matrix-vector multiplication within the eigensolver with block-sparse structured TT representations

In this section, we will provide an explanation on how the block-sparse structure in the TT representation can be exploited to speed-up iterative methods for solving the reduced eigenvalue problem, see Equation (1.151), during QC-DMRG micro-steps. Performing the matrix-vector multiplication in iterative methods for solving the eigenvalue problem stands out as the most time and memory consuming operation, see Figures 4.11(a), 4.11(b) and Table 1.2.

Let $(\mathcal{U}_1, \ldots, \mathcal{U}_d)$ be the block-sparse TT decomposition of the eigenfunction $\mathbf{\Psi} \in \mathbb{R}^{2^d}$, with $\mathcal{U}_k \in \mathbb{R}^{r_{k-1} \times 2 \times r_k}$, $r_0 = r_d = 1$, and $k \in [d]$, see Corollary 3.3.1. Let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be the block-sparse TTO decomposition of the Hamiltonian operator $\mathbf{H} \in \mathbb{R}^{2^d \times 2^d}$, with $\mathcal{H}_k \in \mathbb{R}^{R_{k-1} \times 2 \times 2 \times R_k}$, $R_0 = R_d = 1$, and $k \in [d]$, see Theorem 4. As already discussed in the introductory section, each micro-step in QC-DMRG calculations requires solving (for the 1-site QC-DMRG, see line 13 in **Algorithm** 3 for $k \in [d]$, the following eigenvalue equation:

$$\underbrace{\mathbf{P}_k^* \mathbf{H} \mathbf{P}_k}_{\mathbf{M}_k \in \mathbb{R}^{2r_{k-1}^2 \times 2r_k^2}} \text{vec}\left(\mathcal{U}_k\right) = \lambda_k \, \text{vec}\left(\mathcal{U}_k\right), \tag{4.102}$$

where $\mathbf{P}_k$ is given in Definition (23). Let $\boldsymbol{w}_k := \mathbf{M}_k \, \text{vec}\left(\mathcal{U}_k\right) \in \mathbb{R}^{2r_{k-1}r_k}$. The time complexity for computing $\boldsymbol{w}_k$ scales as $\mathcal{O}(2r^3R^2)$, where $r = \max_{1 \leq k \leq d} \{r_k\}$ and $R = \max_{1 \leq k \leq d} \{R_k\}$.

Now, let $\mathcal{W}_k \in \mathbb{R}^{r_{k-1} \times 2 \times r_k}$ be the tensor folding of $\boldsymbol{w}_k$. $\mathcal{W}_k$ is given by:

$$\mathcal{W}_k = (\underbrace{\mathcal{G}^{\leq k-1} \times_2 \mathcal{H}_k \times_3 \mathcal{G}^{>k-1}}_{\in \mathbb{R}^{r_{k-1} \times 2 \times r_{k-1} \times r_k \times 2 \times r_k}}) \times_{\{3,4,5\}} \mathcal{U}_k, \tag{4.103}$$

where $\times_{\{3,4,5\}}$ refers to the contraction product over the three common modes, see the example in (1.26). $\mathcal{G}^{\leq k-1} \in \mathbb{R}^{r_{k-1} \times R_{k-1} \times r_{k-1}}$ and $\mathcal{G}^{>k-1} \in \mathbb{R}^{r_k \times R_k \times r_k}$ are defined in Equations (4.79) and (4.79). As explained in Remark 4.3.9, in practice, each block in the block-sparse vectors $\text{vec}\left(\mathcal{G}^{\leq k-1}\right)$ and $\text{vec}\left(\mathcal{G}^{>k-1}\right)$ is stored as a list of tensors. An example is given in Figure 4.10 for $k = 2$.

In the following we provide a step-by-step illustration in Figure 4.10 of the operations involved in constructing $\mathcal{W}_k$. It is worth noting that the TT-cores $\mathcal{W}_k$ maintain the same block-sparse structure as $\mathcal{U}_k$, as demonstrated in [2]. For a fixed set of indices (i_{k-1}, μ_k, i_k), we denote the blocks in the matrices, defined as $\mathbf{W}_k[\mu_k] := \mathcal{W}_k[:, \mu_k, :]$, for $\mu_k \in \{1, 2\}$, by:

$$\mathbf{W}_k^{(i_{k-1}, i_k)}[\mu_k] := \mathbf{W}_k[\mu_k]|_{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{\Psi}}_{k,i_k}} \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{\Psi}}_{k,i_k}}, \tag{4.104}$$

where $i_k \in S_k^{\mathbf{\Psi}}$, with $S_k^{\mathbf{\Psi}}$ being defined in (4.2). The block $\mathbf{W}_k^{(i_{k-1}, i_k)}[\mu_k]$ can be generated in three steps; for a fixed set of indices (i_{k-1}, μ_k, i_k):

- Step 1 (depicted in Figure 4.10, step 1): involves contracting the nonzero blocks in the right part $\mathcal{G}^{>k-1}$ (stored as a list of tensors, as explained in Remark 4.3.9) with the nonzero blocks in $\mathbf{U}_k[\nu_k]$, leading to the generation of temporary objects. These temporary objects denoted by $\mathcal{T}^1[\nu_k]$ can be defined as follows, for $\nu_k \in \{1, 2\}$:

$$\mathcal{T}^1[\nu_k] = \mathcal{V}_i^R \times_3 \left(\mathbf{U}_k^{(i'_{k-1}, i'_k)}[\nu_k]\right)^* \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k,i_k} \times \rho^{\mathbf{H}}_{k,j_k} \times \rho^{\mathbf{\Psi}}_{k,i'_{k-1}}}, \tag{4.105}$$

  where tensors $\mathcal{V}_i^R \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k,i_k} \times \rho^{\mathbf{H}}_{k,j_k} \times \rho^{\mathbf{\Psi}}_{k,i'_k}}$ are introduced in Remark 4.3.9, for $i \in [n_{\ell_k}]$.

  Consider that all the blocks per TT-core are of the same size, i.e $R = \bar{m} \max\limits_{k \in [d]} \max\limits_{j_k \in S_k^{\mathbf{H}}} \left\{\rho^{\mathbf{H}}_{k,j_k}\right\}$, with $\bar{m} \leq 2p + 1$. Additionally, let $\bar{n}$ be the minimum number of blocks in the TT-cores of the TT decomposition of the eigenfunction, such that $\bar{n} \leq N + 1$ and assume that all nonzero blocks per TT-core are of the same size, i.e $r = \bar{n} \max\limits_{k \in [d]} \max\limits_{i_k \in S_k^{\mathbf{\Psi}}} \left\{\rho^{\mathbf{H}}_{k,i_k}\right\}$.

  It follows that, this first step scales as $\mathcal{O}(\frac{R}{\bar{m}} \frac{r^3}{\bar{n}^3})$.

- Step 2 (depicted in Figure 4.10, step 2): involves contracting the nonzero blocks in the left part $\mathcal{G}^{\leq k-1}$ (stored as a list of tensors, as well, as explained in Remark 4.3.9) with the temporary objects $\mathcal{T}^1[\nu_k]$. This results into the generation of second temporary objects denoted by $\mathcal{T}_l^2[\nu_k]$, for $\nu_k \in \{1, 2\}$:

$$\mathcal{T}_l^2[\nu_k] = \mathcal{V}_i^L \times_3 \mathcal{T}^1[\nu_k] \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{H}}_{k-1,j_{k-1}} \times \rho^{\mathbf{\Psi}}_{k,i_k} \times \rho^{\mathbf{H}}_{k,j_k}}. \tag{4.106}$$

  The tensors $\mathcal{V}_i^L \in \mathbb{R}^{\rho^{\mathbf{\Psi}}_{k-1,i_{k-1}} \times \rho^{\mathbf{H}}_{k-1,j_{k-1}} \times \rho^{\mathbf{\Psi}}_{k-1,i'_{k-1}}}$ are introduced in Remark 4.3.9, where $i \in \left[n_{\ell_{k-1}}\right]$. Here, each temporary object, $\mathcal{T}_l^2[\nu_k]$, is uniquely identified by an index l, where l refers to a set of indices $(i_{k-1}, j_{k-1}, j_k, i_k)$ satisfying the condition $\ell_k - \ell_{k-1} = 2q_2(\mu_k)$, with $\ell_k = i'_k + j_k + i_k$, $\ell_{k-1} = i'_{k-1} + j_{k-1} + i_{k-1}$, for all $i'_k \in S_k^{\mathbf{\Psi}}$, $i'_{k-1} \in S_{k-1}^{\mathbf{\Psi}}$, with $i'_{k-1} = i'_k - q_2(\nu_k)$ and $j_{k-1} = j_k - (q_2(\mu_k) - q_2(\nu_k))$.

- Step 3 (depicted in Figure 4.10, step 3): consists in contracting the temporary object $\mathcal{T}_l^2[\nu_k]$ with the corresponding nonzero blocks in the TT-cores $\mathbf{H}_k[\mu_k, \nu_k]$. We denote the number of the corresponding nonzero blocks by τ_{μ_k} such that $\tau_{\mu_k} < \sum_{i=1}^{k} \sum_{j=1}^{d-k+1} n_{\ell_{i-1}} n_{\ell_j}$, where $n_{\ell_{i-1}}$, and n_{ℓ_j} are introduced in Remark 4.3.9.

  The blocks $\mathbf{W}_k^{(i_{k-1}, i_k)}[\mu_k]$ can be obtained as follows:

$$\mathbf{W}_k^{(i_{k-1}, i_k)}[\mu_k] = \sum_{l=1}^{\tau_{\mu_k}} \left(\sum_{\nu_k=1}^{2} \mathbf{H}_k^{(j_{k-1}, j_k)}[\mu_k, \nu_k] \times_{\{1,2\}} \mathcal{T}_l^2[\nu_k] \right). \tag{4.107}$$

  The optimal scaling of the second and third steps combined scale as $\mathcal{O}(\tau(\frac{R^2}{\bar{m}^2} \frac{r^2}{\bar{n}^2} + \frac{Rr^3}{\bar{m}\bar{n}^3}))$, with $\tau = \max\limits_{k \in [d]} \max\limits_{\mu_k \in \{1,2\}} \tau_{\mu_k}$.

Note that a similar approach is conducted for the 2-site QC-DMRG.

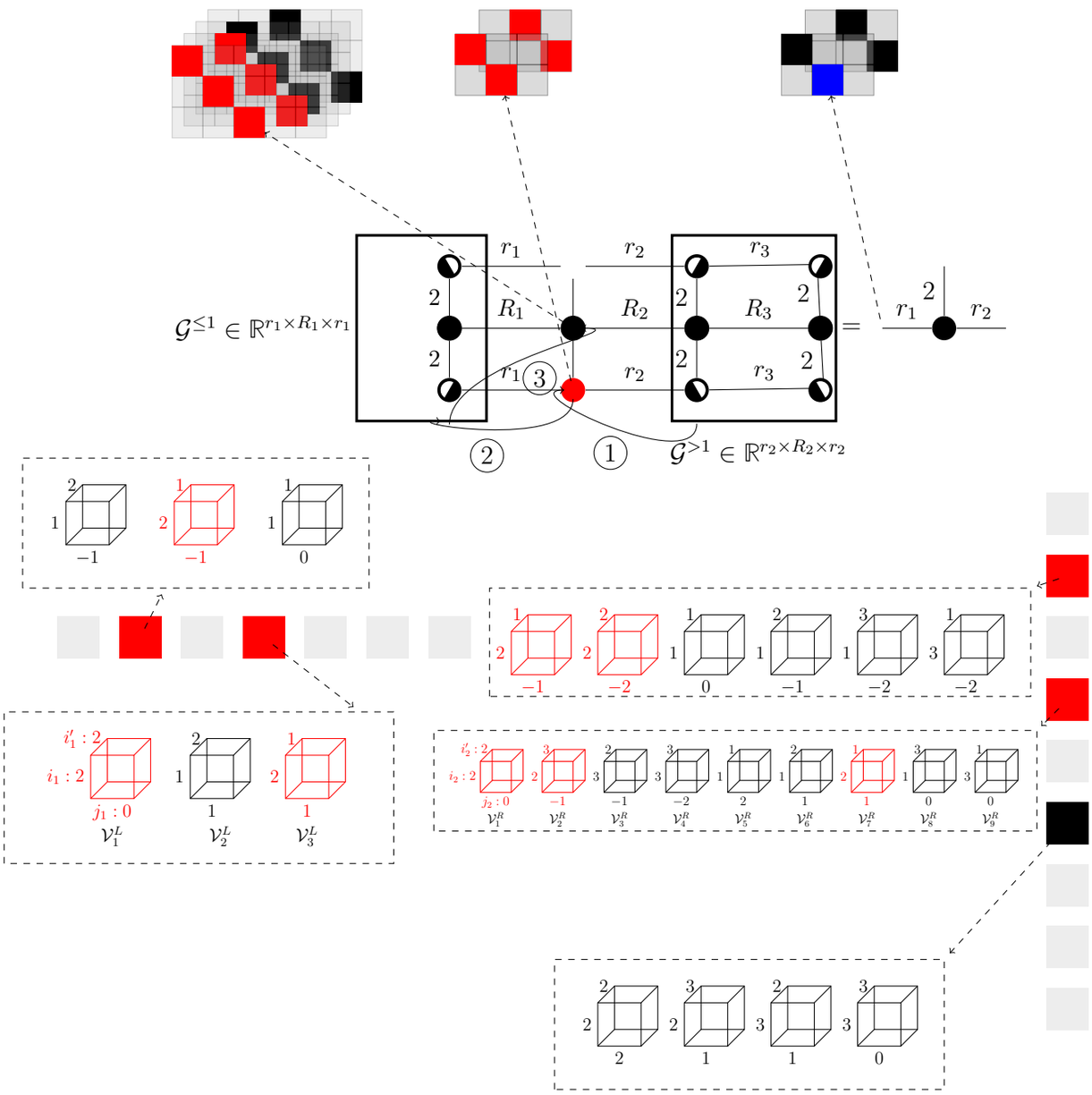


Figure 4.10: Let $d = 4, N = 2, k = 2$. Illustration of the blocks involved in computing $\mathbf{W}_1^{(i_1,i_2)}[\mu_1]$ for a specific set of indices that we denote by $(i_1, \mu_1, i_2) = (2, 1, 2)$, corresponding to the blue square. Only the red squares contribute to the construction of $\mathbf{W}_1^{(i_1,i_2)}[\mu_1]$. The labels on the leftmost cubes, elements of the nonzero blocks of $\mathrm{vec}\left(\mathcal{G}^{\leq 1}\right)$, denote indices (i_1, j_1, i_1') representing the sizes $(\rho^{\mathbf{\Psi}}_{1,i_1}, \rho^{\mathbf{H}}_{1,j_1}, \rho^{\mathbf{\Psi}}_{1,i_1'})$, while the labels on the rightmost cubes, elements of the nonzero blocks of $\mathrm{vec}\left(\mathcal{G}^{>1}\right)$, correspond to indices (i_2, j_2, i_2') representing the sizes $(\rho^{\mathbf{\Psi}}_{2,i_2}, \rho^{\mathbf{H}}_{2,j_2}, \rho^{\mathbf{\Psi}}_{2,i_2'})$. The three steps are described in equations (4.105), (4.106), (4.107).

Numerical results

In line with the numerical results previously presented for the TTO construction, we now provide in Figures 4.11(a) and 4.11(b) the average computational time required for performing the key steps in a 2-site QC-DMRG calculations. These steps include:

- **Eigsolve**: The search for the smallest eigenvalue of the reduced eigenvalue problem, described in Equations (1.164) and (1.165), and referred to as *Eigsolve*, where the Lanczos iterative method is used, (we typically used here few Lanczos iterations, say 2 iterations per reduced eigenvalue problem).
- **Compression**: This step involves the compression of the TT-cores resulting from the eigensolver through tSVD.
- **Contraction**: The process of updating either the left $\mathcal{G}^{\leq k-1}$ or the right parts $\mathcal{G}^{>k}$ through contractions, for $k \in [d]$.

As expected, we can observe that the most computationally demanding step is the eigensolver, which requires matrix multiplications by $\mathbf{H}_k$. However, it is worth highlighting that the block-sparse structure offers a noteworthy advantage in accelerating this operation when compared to a conventional, naive matrix-vector calculation using tensor contractions (without the block-structure representation) in Figure 4.11(b). We note that a gain of almost one order of magnitude is obtained when using a block structured TT representation for all the steps outlined above.

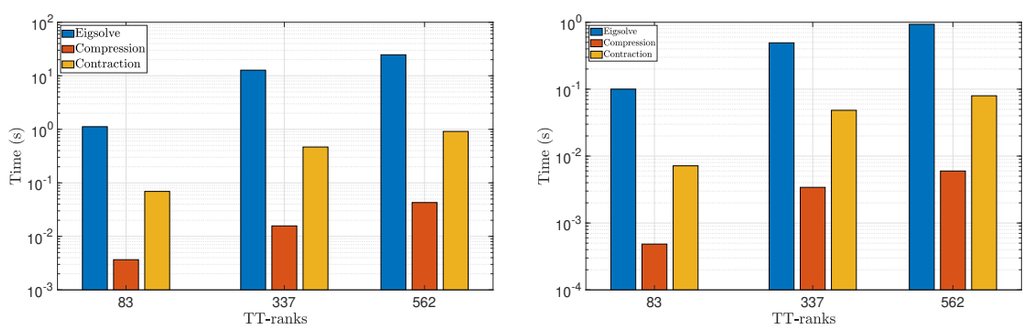


Figure 4.11: Comparing average computational times for performing the most dominant steps in a 2-site QC-DMRG calculations. The numerical illustrations were carried out for a H_{10} molecule, with $d = 20$, a STO-3g basis, and with a maximum TTO-Rank equal to 232.

Once again, to accurately assess the performance of our QC-DMRG algorithm designed for the inherent block-sparse structure of the TT representations, we compare it to ITensor, as illustrated in Figure 4.12. To ensure meaningful comparisons, both libraries employ BLAS2 for matrix multiplication, run on the same Julia version, and share similar parameters. These parameters include the number of Lanczos steps performed at the core of each QC-DMRG half-sweep, truncation parameters to achieve similar TT-ranks in the TT representation of the eigenfunction, and the number of sweeps. It is essential to emphasize that in ITensor, a sparse representation of the Hamiltonian is achieved by invoking the *splitblocks* command, see [28]. It is noted also that these benchmark tests are conducted using a single thread. Additionally, we provide a comparison with a naive QC-DMRG calculation.

As it can be seen from the figures, as the value of d increases, the performance of both libraries appears remarkably similar. Furthermore, Itensor exhibits slightly superior performance for d values within the set $\{4, 8, 12, 16\}$. However, our library demonstrates slightly better performance for the largest TT-ranks, with value 3740, and TTO-ranks, with a maximum value of 436, studied, $d = 28$. It is worth noting that both libraries consistently outperform the naive QC-DMRG calculation.

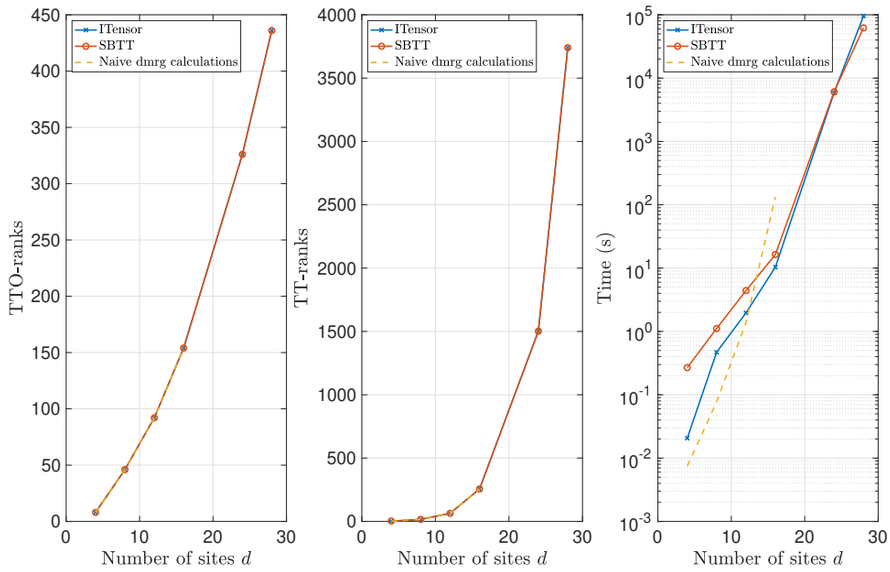
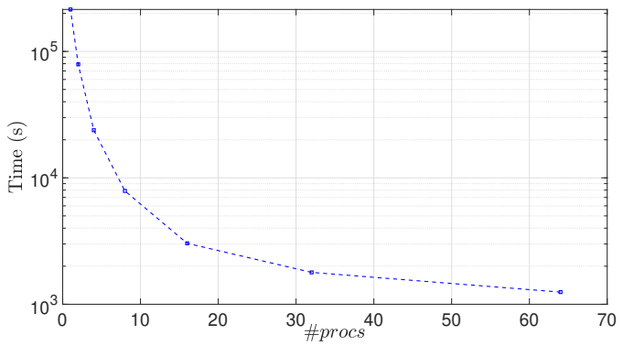


Figure 4.12: Comparison between ITensor, SBTT and naive QC-DMRG calculations. The leftmost figure presents the TTO-ranks vs the number of spin-orbitals d, the middle figure presents the TT-ranks vs d, and the rightmost figure presents the computational time required to perform QC-DMRG calculations after 5 sweeps. These numerical results are conducted for a Hydrogen chain $d \in \{4, \ldots, 28\}$ in the STO-3g basis.

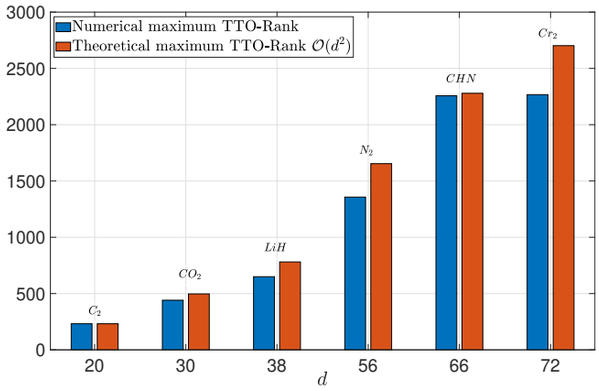
Note on parallelization strategies

As previously mentioned, constructing the TTO representation of the Hamiltonian operator can be efficiently parallelized. The coarse-grained approach involves decomposing the Hamiltonian, in Equation (1.138), into individual sub-Hamiltonian terms, each associated with specific orbital indices. This construction process is then parallelized across $\#procs$ processors, where $\#procs$ represents the number of processors. Each processor is responsible for independently constructing a TTO representation by following the steps detailed in Section 4.3.1. Subsequently, the results from all processors are gathered and further compressed to yield the TTO representation of the 2-body Hamiltonian operator. Figure 4.13(a) provides insights into the execution time for constructing the TTO representation versus the number of processors employed. We note that by using 64 processors, we are able to achieve a 99% reduction in execution time. Additionally, we present the TTO-ranks obtained from constructing the TTO representations of Hamiltonians associated with larger number of orbitals using 64 processors. Work is currently underway to address the challenge of parallelizing the QC-DMRG algorithm by exploiting the block-sparse structure. QC-DMRG is inherently sequential and iterative, with each step often relying on the outcomes of previous steps. This inherent sequential nature can limit the potential for significant parallel speedup. However, when simulating large molecules with more than 100 orbitals, parallelization becomes necessary. Efforts to introduce parallelism to specific aspects of QC-DMRG have been proposed in various works. The most straightforward approach involves parallelizing all linear algebra operations within QC-DMRG computations involving dense matrix-matrix multiplications. Considering the block-sparse structure, parallelization over dense blocks, often referred to as *parallelization over symmetry sectors* [12, 76, 88, 135], becomes particularly relevant when implementing QC-DMRG with symmetry restrictions, as is the case in our work. The block sparsity within the TT-cores provides the advantage of independently handling various operations such as contractions [76, 114], compression, and matrix-vector operations within the eigensolver [88]. These operations can be executed simultaneously since each operation is conducted between matching nonzero blocks, as illustrated in Figure 4.10. An alternative known approach, as described in [115], involves partitioning the system into a finite number of partitions, effectively dividing the set $[d]$ into several subsets. In this approach, the full sweep performed by the sequential algorithm is replaced with concurrent partial sweeps. Each process conducts its sweep along the TT-cores within its

designated partition. When it reaches the boundary of its partition, it communicates the results with its neighboring process to update the local TT-core. This update is achieved through optimization over the contraction of two neighboring orbitals, each belonging to a distinct partition and the update of TT-cores is achieved through SVD.



(a)



(b)

Figure 4.13: The left figure illustrates the parallel TTO construction for the molecule N_2 in the cc-pVDZ basis, with $d = 56$. This construction was carried out while varying the number of processors. The right figure showcases the parallel TTO construction of molecules with large number of spin-orbitals d, employing 64 processors. It provides a comparison between the obtained numerical ranks and theoretical ranks for a given compression threshold of $\epsilon = 10^{-14}$.

The tables 4.2 and 4.1, represent the relative energy errors obtained from our implemented QC-DMRG algorithm and ITensor's for various TT-ranks. We consider the N_2 molecule in the cc-pVDZ basis, typically consisting of $N = 14$ particles and $d = 56$ spin-orbitals. The TTO decomposition of the corresponding Hamiltonian matrix $\mathbf{H} \in \mathbb{R}^{2^{56} \times 2^{56}}$ is built in parallel using 64 processors during the pre-computational steps. In this example, we have a reference solution that approximates the full configuration interaction solution, i.e the energy of the exact eigenvalue on $\mathbb{R}^{2^{56}}$, obtained from [56], which is:

$$\lambda_{ref} = -109.280319 E_h, \tag{4.108}$$

where E_h is the unit of the energy, *Hartree*. Tables 4.2 and 4.1 provide a comprehensive view of how the TT-ranks evolve as the number of sweeps increases and how the relative error decreases with each successive sweep. Notably, the second table for ITensor also highlights a significant challenge – running out of memory during the third sweep, indicating a limitation of handling this specific computational task, compared to our code.

	Sweeps	TT-ranks	Relative error
SBTT	1	445	10^{-2}
	2	1589	10^{-3}
	3	1884	10^{-4}

Table 4.1: N_2 molecule in cc-pVDZ basis set with $N = 14$, $d = 56$, and maximum TTO-rank is 1280.

	Sweeps	TT-ranks	Relative error
ITensor	1	445	10^{-2}
	2	1589	10^{-3}
	3	1884	Out-of-memory

Table 4.2: N_2 molecule in cc-pVDZ basis set with $N = 14$, $d = 56$, and maximum TTO-rank is 1280.

4.4 Concluding remarks and perspectives

In this chapter, we introduced our proof-of-concept QC-DMRG library, which is specifically designed for the particle-preserving quantum chemical 2-body Hamiltonian operator. In this library, both the TT decomposition of the operator and the eigenfunction are represented using a block-sparse structure. Subsequently, we centered our attention on building the block-sparse TTO representation of a 2-body Hamiltonian operator, while having both the one and two-electron integrals. We illustrated how this approach can be extended to accommodate any p-body Hamiltonian operator, which can prove to be advantageous for the TT decomposition of operators employed in nuclear physics or for representing complex Hamiltonians, such as the transcorrelated Hamiltonian [5], a non-Hermitian 3-body operator. In addition to the construction process, we provided insights into the main algorithms involved, including main algebraic operations such as multiplication and addition between block-sparse TT-cores.

Our attention then turned to the integration of the block-sparse structure within the TT decompositions of both the operator and the eigenfunction, optimizing the block-sparse tensor contractions, as they are essential operations frequently employed during QC-DMRG sweeps. Additionally, we elaborated on how these structures can be used to accelerate matrix-vector multiplications within the eigensolver. To contextualize our proof-of-concept library, we conducted a comprehensive comparison of the fundamental operations employed, with both naive computations and the state-of-the-art ITensor library. This yielded to notable conclusions: the block-sparse structure provides a substantial advantage in speeding up the construction of the Hamiltonian operator, tensor contractions, and the eigensolver. While ITensor demonstrates superior performance in the construction of the TTO, it is worth noting that this is an embarrassingly parallel problem, and parallelization can be employed to accelerate the construction process. Additionally, we are actively exploring other avenues for improvement, such as optimizing the most time-consuming operations, like compression. This includes investigating alternatives to the costly truncated SVD approach. Ultimately, both ITensor and our proof-of-concept library appear to exhibit similar performances when executing QC-DMRG for a Hydrogen chain as illustrated in this chapter. In future work, we plan to investigate parallelization strategies to optimize all operations within our library, by exploiting the block sparsity within the TT decompositions. Additionally, we aim to generalize the construction of the operator to accommodate a broader range of p-body operators characterized by block-sparse TT decomposition. While the investigation into the algorithm's convergence falls beyond the scope of this thesis, in future work we are inclined towards delving into this aspect. It is noteworthy that, from a theoretical standpoint, there is still a deficiency in a comprehensive theory addressing global convergence. However, there are numerous numerical attempts to speed-up convergence, as referenced in Remark 1.3.13. On our part, we look into the exploration of innovative approaches, such as the potential incorporation of randomization methods.

Conclusion and perspectives

In this thesis, we have showed that it is possible to efficiently solve two high-dimensional problems arising in quantum chemistry by thoroughly investigating and exploiting the inherent structures of the matrices and tensors involved to overcome the curse of dimensionality. The main novel contributions of this thesis are summarized in the following list.

- **Efficient numerical evaluation of the long-range component of the Coulomb potential and the long-range 4-th order two-electron integrals tensor** In Chapter 2, our initial focus was on obtaining a low-rank approximation of the multivariate function (tensor) representing the long-range component of the range-separated Coulomb potential. In this context, we introduced two novel compression methods for the long-range component and for the long-range 4-th order six-dimensional two-electron integrals tensor.

  The first approach, referred to as LTEI-TA, relied on two-dimensional Chebyshev interpolation and Gaussian quadrature for numerical integration. This resulted in an expansion in terms of tensorized polynomials, as given in Theorem 2, for the evaluation of a single integral. An error bound for this approximation was provided in Proposition 2.3.1, showcasing that the approximation error depends on the value of ω, the separation parameter that controls the regularity of the long-range kernel, the number of quadrature points N_{q_1}, the number of interpolation points N, and the dimension of the hypercube b. Furthermore, we extended this approximation approach to the 4-th order two-electron integral tensor employed in quantum chemistry calculations. This extension resulted in a novel factorized structure, elaborated in Equation (2.45). We showed that this tensorized structure provides significant advantages in speeding up basic algebraic operations, notably matrix-matrix multiplications. After confirming that the underlying long-range kernel is asymptotically smooth, as demonstrated in Proposition 2.4.1, the second approach, termed as LTEI-FMM, employs kernel-independent Fast Multipole Methods to derive a factorized expression of the kernel and the associated 4-th order two-electron integral tensor, outlined in Equations (2.55) and (2.60). Following a comparative study between both methods across various molecules in order to evaluate the long-range Coulomb matrix, our findings revealed that the LTEI-TA approach exhibits particular efficiency and outperforms LTEI-FMM for small values of ω. However, for large values of ω, in order to preserve accuracy, the number of interpolation points as well as the number of quadrature points becomes important for LTEI-TA and thus LTEI-FMM becomes more efficient. We further investigated the compression of the long-range 4-th order tensor by using screening techniques, low-rank methods, and an adaptive approach, as described in Section 2.6.2. Most content of this chapter is based on our publication [4].

- **Symmetry preserving tensor train representations arising in QC-DMRG calculations** In Chapter 3, our focus centered on investigating the efficient TTO representation of the Hamiltonian operator, moving from the classical 2-body quantum chemical Hamiltonian operator to a more general p-body Hamiltonian operator. This representation is at the core of the QC-DMRG algorithm and can be achieved through low-rank approximation methods and the exploitation of physical/group symmetries. By focusing on the derivation of approximate TTO representations, one of the key takeaways of this work has been the necessity to carefully employ the low-rank approximation method through truncated SVD, as given in Theorem 3.

We showed that it is imperative to respect degenerate singular values, as this ensures the preservation of essential operator properties. In particular, it maintains its representation as a symmetric matrix and upholds Abelian symmetries, as the conservation of particle number and the conservation of the z-component of the total spin. Our numerical findings further showcased that the application of truncated SVD with specific thresholds affects additional properties retained by the Hamiltonian operator, such as non-Abelian symmetries and the occurrence of non-existing interaction terms, especially for operators constrained to at most 2-body interactions. Moreover, considering the specified conditions on the entries of the particle-preserving Hamiltonian operator matrix, we derived a constructive demonstration showcasing the block-sparse structure inherent within the TT-cores of the TTO decomposition of a general p-body particle-number preserving Hamiltonian operator, as highlighted in Theorem 4, such as the TT-cores have a block-sparse representation with at most $2p + 1$ nonzero blocks. The TTO representation derived has not been introduced in this particular representation before for a general p-body particle-preserving Hamiltonian operator. Furthermore, we laid the groundwork for theoretical insights into the block structure of the TT-cores within the TT decomposition of the eigenfunction, the wavefunction. This wavefunction is presumed to be invariant under non-Abelian symmetries, such as SU(2). In this study, we have derived a novel theoretical formulation for the required number of configurations to achieve a wavefunction with specific target quantum numbers, as described in Equation (3.135). This formulation, outlined in Proposition 3.3.6, to the best of our knowledge, has not been previously proposed. Furthermore, we have showed that the number of blocks within the TT-cores is further reduced when incorporating SU(2) symmetry, as illustrated in Theorem 5.

- **Efficient algorithmic design of basic operations in QC-DMRG and numerical experiments** In chapter 4, we have presented our proof-of-concept QC-DMRG library, designed for the quantum chemical Hamiltonian operator model that incorporates particle number conservation (PN). It can be generalized to accommodate a more general p-body particle-preserving operator, although this extension is not currently supported in the existing version. The library exploits the block-sparse structure of both the TT decomposition of the operator and the eigenfunction. We outlined key algorithms exploiting the block-sparse representation of the TT-cores, including algebraic operations encountered in QC-DMRG algorithm. Numerical results across diverse molecules, thus different system sizes d, highlight the benefits of employing block-sparse TT decompositions, including generic construction of the TTO representation of the Hamiltonian operator, block-wise compression and orthogonalization, tensor contractions and matrix-vector multiplication within the eigensolver. Comparisons were made with the state-of-the art ITensor library. Ultimately, both ITensor and our proof-of-concept library appear to exhibit similar performances when executing QC-DMRG for a Hydrogen chain as illustrated in this chapter. Still, ITensor outperforms our library in constructing the TTO decomposition.

In the following, we outline potential avenues for future research directions arising from the findings presented in this thesis.

- In the first part of this work, as highlighted in Chapter 2, it is crucial to acknowledge existing established approaches aiming at handling similar long-range kernels. Therefore, we anticipate that a comparative analysis against our proposed methods with the existing methods could yield valuable insights. In our forthcoming research, we aim also to delve deeper into the potential applications of LTEI-TA for small values of ω across a wider spectrum of quantum chemistry contexts. This exploration will encompass post-Hartree-Fock models featuring range separation and hybrid approaches like (long-range) DMRG–short-range DFT [43]. Additionally, we anticipate potential benefits for Particle Mesh Ewald methods [21] through these investigations. Furthermore, we acknowledge the significance of examining the treatment of the short-range TEI tensor, as its computational cost and accuracy depend on the value of ω, which is the trade-off between the long-range and short-range

parts, and the numerical method used. A natural extension to this work would be the treatment of the short-range part. It would be interesting to link LTEI-FMM with singular quadrature-based evaluation methods for short-range. Additionally, efficient parallelization of FMM could also benefit to LTEI-FMM on distributed memory architecture.

- In the second part of this work, as elaborated in Chapter 3, many open questions remain to be answered. Our current research is dedicated to further exploring the invariance under SU(2) symmetry and its implications on TT representations. In this dissertation, in Section 3.3.2, our analysis was limited to the singlet state ($J = 0$). An important next step, is to extend this work to encompass any value of the target total spin J and to examine the resulting TT representation not only for the eigenfunction but also for the Hamiltonian operator. To do so, we are currently exploring an alternative parametrization of the wavefunction in order to define a different basis compatible with the particle-number and spin-symmetry, instead of the symmetry-adapted basis introduced in Section 3.3.2. This work is still in its early stages and requires further investigation into the significant algebra involved.

- In the third part of this work, as elaborated in Chapter 4, we will focus on further optimizing all the library's operations, expanding the operator's construction to a broader range of p-body symmetry-preserving operators including SU(2) symmetry. We aim to investigate innovative numerical techniques for accelerating the eigensolver convergence, such as randomization. Furthermore, the ongoing work involves the parallelization of the basic operations within the QC-DMRG algorithm using Message Passing Interface (MPI) on large scale architectures by exploiting the inherent block-structure. However, beyond the parallelization of the basic algebraic operations, we are interested in exploring alternative approaches to parallelize the serial iterative process within DMRG. We are looking into a new approach inspired by Parareal method [80], originally developed for parallelizing time-dependent problems, in order to adapt its concepts to this context.

Bibliography

[1] Atkins, P. and Friedman, R. (2005). *Molecular Quantum Mechanics*. Oxford University Press.

[2] Bachmayr, M., Götte, M., and Pfeffer, M. (2022). Particle number conservation and block structures in matrix product states. *Calcolo*, 59(2).

[3] Bader, B. W. and Kolda, T. G. (2006). Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653.

[4] Badreddine, S., Chollet, I., and Grigori, L. (2023). Factorized structure of the long-range two-electron integrals tensor and its application in quantum chemistry. *Journal of Computational Physics*, 493:112460.

[5] Baiardi, A., Lesiuk, M., and Reiher, M. (2022). Explicitly correlated electronic structure calculations with transcorrelated matrix product operators. *Journal of Chemical Theory and Computation*, 18(7):4203–4217.

[6] Baiardi, A. and Reiher, M. (2020). Transcorrelated density matrix renormalization group. *The Journal of Chemical Physics*, 153(16).

[7] Bebendorf, M. (2008). Hierarchical matrices. *Lecture notes in computational science and engineering, v.63 (2008)*, 63.

[8] Bellman, R. E. (2010). *Dynamic Programming*. Princeton University Press.

[9] Benner, P., Khoromskaia, V., Khoromskij, B., Kweyu, C., and Stein, M. (2021). Regularization of Poisson–Boltzmann type equations with singular source terms using the range-separated tensor format. *SIAM Journal on Scientific Computing*, 43(1):A415–A445.

[10] Benner, P., Khoromskaia, V., and Khoromskij, B. N. (2018). Range-separated tensor format for many-particle modeling. *SIAM Journal on Scientific Computing*, 40(2):A1034–A1062.

[11] Biedenharn, L. C. and Louck, J. D. (1981). *Angular momentum in quantum physics. Theory and application*, volume 8.

[12] Brabec, J., Brandejs, J., Kowalski, K., Xantheas, S., Legeza, O., and Veis, L. (2020). Massively parallel quantum chemical density matrix renormalization group method. *Journal of Computational Chemistry*, 42(8):534–544.

[13] Cancès, E., Defranceschi, M., Kutzelnigg, W., Bris, C. L., and Maday, Y. (2003). Computational quantum chemistry: A primer. In *Handbook of Numerical Analysis*, pages 3–270. Elsevier.

[14] Cancès, E. and Friesecke, G., editors (2023). *Density Functional Theory*. Springer International Publishing.

[15] Chaillat, S., Desiderio, L., and Ciarlet, P. (2017). Theory and implementation of $\mathcal{H}$-matrix based iterative and direct solvers for Helmholtz and elastodynamic oscillatory kernels. *Journal of Computational Physics*.

[16] Chan, G. K.-L. (2004). An algorithm for large scale density matrix renormalization group calculations. *The Journal of Chemical Physics*, 120(7):3172–3178.

[17] Chan, G. K.-L., Keselman, A., Nakatani, N., Li, Z., and White, S. R. (2016). Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of Chemical Physics*, 145(1).

[18] Chollet, I. (2021). *Symmetries and Fast Multipole Methods for Oscillatory Kernels*. Theses, Sorbonne Université.

[19] Chollet, I., Claeys, X., Fortin, P., and Grigori, L. (2023). A directional equispaced interpolation-based fast multipole method for oscillatory kernels. *SIAM Journal on Scientific Computing*, 45(1):C20–C48.

[20] Cornwell, J. (1997). *Group Theory in Physics: An Introduction*. Techniques of Physics Series. Academic Press.

[21] Darden, T., York, D., and Pedersen, L. (1993). Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092.

[22] Demel, O., Lecours, M. J., Habrovský, R., and Nooijen, M. (2021). Toward laplace MP2 method using range separated Coulomb potential and orbital selective virtuals. *The Journal of chemical physics*, 155(15):154104.

[23] Devos, L., Van Damme, M., and Haegeman, J. (2023). Tensoroperations.jl.

[24] Dongarra, J., Croz, J., Hammarling, S., and Duff, I. (1990). A set of level 3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 16(1):1–17.

[25] Dupuis, M., Rys, J., and King, H. F. (1976). Evaluation of molecular integrals over gaussian basis functions. *The Journal of Chemical Physics*, 65(1):111–116.

[26] Evenbly, G. (2022). A practical guide to the numerical implementation of tensor networks : Contractions, decompositions, and Gauge freedom. *Frontiers in Applied Mathematics and Statistics*, 8.

[27] Evenbly, G. and Vidal, G. (2011). Tensor network states and geometry. *Journal of Statistical Physics*, 145(4):891–918.

[28] Fishman, M., White, S. R., and Stoudenmire, E. M. (2022). The ITensor Software Library for Tensor Network Calculations. *SciPost Phys. Codebases*, page 4.

[29] Fong, W. and Darve, E. (2009). The black-box fast multipole method. *Journal of Computational Physics*, 228(23):8712–8725.

[30] Franck, O., Mussard, B., Luppi, E., and Toulouse, J. (2015). Basis convergence of range-separated density-functional theory. *The Journal of Chemical Physics*, 142(7). 074107.

[31] Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[32] Fromager, E., Réal, F., Wåhlin, P., Wahlgren, U., and Jensen, H. J. A. (2009). On the universality of the long-/short-range separation in multiconfigurational density-functional theory. II. investigating f actinide species. *The Journal of Chemical Physics*, 131(5).

[33] Garniron, Y., Applencourt, T., Gasperich, K., Benali, A., Ferté, A., Paquier, J., Pradines, B., Assaraf, R., Reinhardt, P., Toulouse, J., Barbaresco, P., Renon, N., David, G., Malrieu, J.-P., Véril, M., Caffarel, M., Loos, P.-F., Giner, E., and Scemama, A. (2019). Quantum package 2.0: An open-source determinant-driven suite of programs. *Journal of Chemical Theory and Computation*, 15(6):3591–3609. PMID: 31082265.

[34] Gill, P. M. (1994). *Molecular integrals Over Gaussian Basis Functions*, page 141–205. Elsevier.

[35] Giner, E. (2021). A new form of transcorrelated Hamiltonian inspired by range-separated DFT. *The Journal of Chemical Physics*, 154(8):084119.

[36] Götte, M. (2022). Applications of tensor networks in quantum chemistry and polynomial regression.

[37] Grasedyck, L. (2010). Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054.

[38] Greengard, L. and Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348.

[39] Griffiths, D. J. and Schroeter, D. F. (2018). *Introduction to Quantum Mechanics*. Cambridge University Press.

[40] Hackbusch, W. (2015). *Hierarchical Matrices: Algorithms and Analysis*. Springer Berlin Heidelberg.

[41] Hamilton, W. R. (1855). VIII. on some extensions of quaternions. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9(56):46–51.

[42] Hauschild, J. and Pollmann, F. (2018). Efficient numerical simulations with tensor networks: Tensor network python (TeNPy). *SciPost Physics Lecture Notes*.

[43] Hedegård, E. D., Knecht, S., Kielberg, J. S., Jensen, H. J. A., and Reiher, M. (2015). Density matrix renormalization group with efficient dynamical electron correlation through range separation. *The Journal of Chemical Physics*, 142(22):224108.

[44] Helgaker, T., Jørgensen, P., and Olsen, J. (2000). *Molecular Electronic-Structure Theory*. Wiley.

[45] Hill, J. G. (2012). Gaussian basis sets for molecular applications. *International Journal of Quantum Chemistry*, 113(1):21–34.

[46] Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

[47] Holtz, S., Rohwedder, T., and Schneider, R. (2011). On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731.

[48] Holtz, S., Rohwedder, T., and Schneider, R. (2012). The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713.

[49] Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

[50] Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM*, 5(4):339–342.

[51] Hubig, C. (2017). *Symmetry-protected tensor networks*. PhD thesis.

[52] Hubig, C., McCulloch, I. P., and Schollwöck, U. (2017). Generic construction of efficient matrix product operators. *Physical Review B*, 95(3).

[53] J. Mathar, R. (2006). SO(3) Clebsch Gordan coefficients.

[54] JafariBehbahani, Z. and Roodaki, M. (2015). Two-dimensional chebyshev hybrid functions and their applications to integral equations. *Beni-Suef University Journal of Basic and Applied Sciences*, 4(2):134–141.

[55] János G., A., Iann C., G., Savin, A., and Toulouse, J. (2005). van der Waals forces in density functional theory: Perturbational long-range electron-interaction corrections. *Physical Review A*, 72(1).

[56] Johnson, R. (2002). Computational chemistry comparison and benchmark database, NIST standard reference database 101.

[57] Kahaner, D., Moler, C., and Nash, S. (1989). *Numerical Methods and Software*. Prentice-Hall series in computational mathematics. Prentice-Hall.

[58] Kapalı, Y. (2014). Atomic orbitals. `https://chemicalalgos.wordpress.com/2014/04/29/atomic-orbitals/`.

[59] Keller, S., Dolfi, M., Troyer, M., and Reiher, M. (2015). An efficient matrix product operator representation of the quantum chemical hamiltonian. *The Journal of Chemical Physics*, 143(24).

[60] Keller, S. and Reiher, M. (2016). Spin-adapted matrix product states and operators. *The Journal of Chemical Physics*, 144(13):134101.

[61] Khoromskaia, V. (2010). Computation of the Hartree-Fock exchange by the tensor-structured methods. *Computational Methods in Applied Mathematics*, 10(2):204–218.

[62] Khoromskaia, V., Khoromskij, B. N., and Schneider, R. (2013). Tensor-structured factorized calculation of two-electron integrals in a general basis. *SIAM Journal on Scientific Computing*, 35(2):A987–A1010.

[63] Khoromskij, B. N. (2011). O(dlogN)-quantics approximation of N-d tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34(2):257–280.

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

[65] King, H. F. and Dupuis, M. (1976). Numerical integration using rys polynomials. *Journal of Computational Physics*, 21(2):144–165.

[66] Knowino (2010). Gaussian type orbitals — knowino, an encyclopedia.

[67] Koch, H., de Merás, A. S., and Pedersen, T. B. (2003). Reduced scaling in electronic structure calculations using Cholesky decompositions. *The Journal of Chemical Physics*, 118(21):9481–9484.

[68] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

[69] Kweyu, C., Khoromskaia, V., Khoromskij, B., Stein, M., and Benner, P. (2021). Solution decomposition for the nonlinear Poisson-Boltzmann equation using the range-separated tensor format.

[70] Landsberg, J. (2011). *Tensors: Geometry and Applications*. American Mathematical Society.

[71] Lecours, Michael (2021). *Compact Sparse Coulomb Integrals using a Range-Separated Potential*. PhD thesis, University of Waterloo.

[72] Lee, A., Taylor, S., Dombroski, J., and Gill, P. (1997). Optimal partition of the coulomb operator. *Physical Review A - PHYS REV A*, 55:3233–3235.

[73] Lee, N. and Cichocki, A. (2017). Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, 29(3):921–960.

[74] Legeza, O., Rohwedder, T., Schneider, R., and Szalay, S. (2014). Tensor product approximation (DMRG) and coupled cluster method in quantum chemistry. In *Many-Electron Approaches in Physics, Chemistry and Mathematics*, pages 53–76. Springer International Publishing.

[75] Lehtola, S., Blockhuys, F., and Alsenoy, C. V. (2020). An overview of self-consistent field calculations within finite basis sets. *Molecules*, 25(5):1218.

[76] Levy, R., Solomonik, E., and Clark, B. K. (2020). Distributed-memory DMRG via sparse and dense parallel tensor contractions. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE.

[77] Li, Z. and Chan, G. K.-L. (2016). Hilbert space renormalization for the many-electron problem. *The Journal of Chemical Physics*, 144(8).

[78] Limpanuparb, T., Hollett, J. W., and Gill, P. M. W. (2012). Resolutions of the Coulomb operator. VI. computation of auxiliary integrals. *The Journal of Chemical Physics*, 136(10):104102.

[79] Limpanuparb, T., Milthorpe, J., Rendell, A., and Gill, P. (2013). Resolutions of the Coulomb operator: Vii. evaluation of long-range Coulomb and exchange matrices. *Journal of Chemical Theory and Computation*, 9:863–867.

[80] Lions, J.-L., Maday, Y., and Turinici, G. (2001). Résolution d'edp par un schéma en temps «pararéel ». *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332(7):661–668.

[81] Liu, S. and TRENKLER, O. (2008). Hadamard, Khatri-rao, Kronecker and other matrix products. *International Journal of Information , Systems Sciences*, 4.

[82] Losilla, S. A., Watson, M. A., Aspuru-Guzik, A., and Sundholm, D. (2015). Construction of the Fock matrix on a grid-based molecular orbital basis using GPGPUs. *Journal of Chemical Theory and Computation*, 11(5):2053–2062. PMID: 26574409.

[83] Lu, B., Cheng, X., and McCammon, Andrew, J. (2007). "New-version-fast-multipole-method" accelerated electrostatic calculations in biomolecular systems. *Journal of Computational Physics*, 226(2):1348–1366.

[84] Magalhães, A. L. (2014). Gaussian-type orbitals versus slater-type orbitals: A comparison. *Journal of Chemical Education*, 91(12):2124–2127.

[85] Mcculloch, I. and Gulácsi, M. (2001). Total spin in the density matrix renormalization group algorithm. *Philosophical Magazine Letters*, 81:447–453.

[86] McCulloch, I. P. and Gulá csi, M. (2002). The non-abelian density matrix renormalization group algorithm. *Europhysics Letters (EPL)*, 57(6):852–858.

[87] McMurchie, L. E. and Davidson, E. R. (1978). One- and two-electron integrals over cartesian Gaussian functions. *Journal of Computational Physics*, 26(2):218–231.

[88] Menczer, A. and Legeza, O. (2023). Massively parallel tensor network state algorithms on hybrid CPU-GPU based architectures.

[89] Nash, J. (2022). Distributed.jl.

[90] Orús, R. (2014). A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158.

[91] Oseledets, I. (2011a). DMRG approach to fast linear algebra in the TT-format. *Computational Methods in Applied Mathematics*, 11(3):382–393.

[92] Oseledets, I. (2011b). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317.

[93] Platte, R. B. and Trefethen, L. N. (2010). *Chebfun: A New Kind of Numerical Computing*, pages 69–87. Springer Berlin Heidelberg, Berlin, Heidelberg.

[94] Pritchard, B. P., Altarawy, D., Didier, B., Gibson, T. D., and Windus, T. L. (2019). New basis set exchange: An open, up-to-date resource for the molecular sciences community. *Journal of Chemical Information and Modeling*, 59(11):4814–4820. PMID: 31600445.

[95] Prodan, E. (2019). Computational Many-Body Physics via $\mathcal{M}_{2^q}$ Algebra.

[96] Rebolini, E. (2014). *Range-separated density-functional theory for molecular excitation energies*. Theses, Université Pierre et Marie Curie - Paris VI.

[97] Ricci, M. and Levi-Civita, T. (1901). Méthodes de calcul différentiel absolu et leurs applications. *Mathematische Annalen*, 54:125–201.

[98] Rohwedder, T. (2010). An analysis for some methods and algorithms of quantum chemistry.

[99] Rohwedder, T. and Uschmajew, A. (2013). On local convergence of alternating schemes for optimization of convex problems in the tensor train format. *SIAM Journal on Numerical Analysis*, 51(2):1134–1162.

[100] Rosal Sandberg, J. A. (2014). *New efficient integral algorithms for quantum chemistry*. PhD thesis, KTH, Theoretical Chemistry and Biology. QC 20140826.

[101] Saad, Y. (2011). *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics.

[102] Savin, A. (2020). Models and corrections: Range separation for electronic interaction—lessons from density functional theory. *The Journal of Chemical Physics*, 153(16):160901.

[103] Scheiber, E. (2015). On the Chebyshev approximation of a function with two variables.

[104] Schmoll, P., Singh, S., Rizzi, M., and Orús, R. (2020). A programming guide for tensor networks with global SU(2) symmetry. *Annals of Physics*, 419:168232.

[105] Schneider, R. (2009). Analysis of the projected coupled cluster method in electronic structure calculation. *Numerische Mathematik*, 113(3):433–471.

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

[107] Sharma, S. and Chan, G. K.-L. (2012). Spin-adapted density matrix renormalization group algorithms for quantum chemistry. *The Journal of Chemical Physics*, 136(12):124121.

[108] Shavitt, I. and Bartlett, R. J. (2009). *Many-Body Methods in Chemistry and Physics*. Cambridge University Press.

[109] Sherrill, C. (2010). Density-fitting approximations to the electron repulsion integrals.

[110] Simmonett, A., Brooks, B., and Darden, T. (2022). Efficient and scalable electrostatics via spherical grids and treecode summation. unpublished.

[111] Singh, S., Pfeifer, R. N. C., and Vidal, G. (2011). Tensor network states and algorithms in the presence of a global U(1) symmetry. *Phys. Rev. B*, 83:115125.

[112] Singh, S., Zhou, H.-Q., and Vidal, G. (2010). Simulation of one-dimensional quantum systems with a global SU(2) symmetry. *New Journal of Physics*, 12(3):033029.

[113] Snajberk, P. and Ochsenfeld, C. (2017). *Direct Density Matrix Renormalization Group Approaches for Strong Correlation Effects in Quantum Chemistry*. Universitätsbibliothek der Ludwig-Maximilians-Universität.

[114] Solomonik, E., Matthews, D., Hammond, J. R., Stanton, J. F., and Demmel, J. (2014). A massively parallel tensor contraction framework for coupled-cluster computations. *Journal of Parallel and Distributed Computing*, 74(12):3176–3190.

[115] Stoudenmire, E. M. and White, S. R. (2013). Real-space parallel density matrix renormalization group. *Physical Review B*, 87(15).

[116] Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., Wouters, S., and Chan, G. K.-L. (2017). Pyscf: the python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1).

[117] Szabo, A. and S.Ostlund, N. (1996). Modern quantum chemistry : introduction to advanced electronic structure theory. In *Special Volume, Computational Chemistry*, page 481 / 481. Mineola (N.Y.) : Dover publications.

[118] Szalay, S., Pfeffer, M., Murg, V., Barcza, G., Verstraete, F., Schneider, R., and Legeza, O. (2015). Tensor product methods and entanglement optimization for ab initio quantum chemistry. *International Journal of Quantum Chemistry*, 115(19):1342–1391.

[119] Tichai, A., Roth, R., and Duguet, T. (2020). Many-body perturbation theories for finite nuclei. *Frontiers in Physics*, 8.

[120] Toulouse, J. (2005). *Extension multidéterminantale de la méthode de Kohn-Sham en théorie de la fonctionnelle de la densité par décomposition de l'interaction électronique en contributions de longue portée et de courte portée*. Theses, Université Pierre et Marie Curie - Paris VI.

[121] Toulouse, J., Colonna, F., and Savin, A. (2004). Long-range–short-range separation of the electron-electron interaction in density-functional theory. *Phys. Rev. A*, 70:062505.

[122] Toulouse, J., Gori-Giorgi, P., and Savin, A. (2005). A short-range correlation energy density functional with multi-determinantal reference. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling*, 114:305.

[123] Toulouse, J., Zhu, W., Ángyán, J. G., and Savin, A. (2010). Range-separated density-functional theory with the random-phase approximation: Detailed formalism and illustrative applications. *Phys. Rev. A*, 82:032502.

[124] Townsend, A. and Trefethen, L. N. (2013). An extension of Chebfun to two dimensions. *SIAM Journal on Scientific Computing*, 35(6):C495–C518.

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

[126] Uschmajew, A. and Vandereycken, B. (2020). Geometric methods on low-rank matrix and tensor manifolds. In Grohs, P., Holler, M., and Weinmann, A., editors, *Variational methods for nonlinear geometric data and applications*. Springer.

[127] Verstraete, F., Nishino, T., Schollwöck, U., Bañuls, M. C., Chan, G. K., and Stoudenmire, M. E. (2023). Density matrix renormalization group, 30 years on. *Nature Reviews Physics*, 5(5):273–276.

[128] Vries, O. (2021). *Clebsch-Gordan Coefficients: A Quantum Mechanical and Mathematical Perspective*. PhD thesis, Delft University of Technology.

[129] Weichselbaum, A. (2012). Non-abelian symmetries in tensor networks: A quantum symmetry space approach. *Annals of Physics*, 327(12):2972–3047.

[130] White, C. A., Johnson, B. G., Gill, P. M., and Head-Gordon, M. (1994). The continuous fast multipole method. *Chemical Physics Letters*, 230(1–2):8–16.

[131] White, S. R. (1992). Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866.

[132] Whitten, J. L. (2003). Coulombic potential energy integrals and approximations. *The Journal of Chemical Physics*, 58(10):4496–4501.

[133] Wouters, S. and Neck, D. V. (2014). The density matrix renormalization group for ab initio quantum chemistry. *The European Physical Journal D*, 68(9).

[134] Wouters, S., Poelmans, W., Ayers, P. W., and Neck, D. V. (2014). CheMPS2: A free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry. *Computer Physics Communications*, 185(6):1501–1514.

[135] Xie, Z., Song, Y., Peng, F., Li, J., Cheng, Y., Zhang, L., Ma, Y., Tian, Y., Luo, Z., and Ma, H. (2023). Kylin 1.0: An ab-initio density matrix renormalization group quantum chemistry program. *Journal of Computational Chemistry*, 44(13):1316–1328.

[136] Xing, X. and Chow, E. (2020). Fast Coulomb matrix construction via compressing the interactions between continuous charge distributions. *SIAM Journal on Scientific Computing*, 42(1):A162–A186.

[137] Zhai, H. and Chan, G. K.-L. (2021). Low communication high performance ab initio density matrix renormalization group algorithms. *J. Chem. Phys.*, 154(22):224116.

[138] Zhai, H., Larsson, H. R., Lee, S., Cui, Z.-H., Zhu, T., Sun, C., Peng, L., Peng, R., Liao, K., Tölle, J., Yang, J., Li, S., and Chan, G. K.-L. (2023). <scp>block2</scp>: A comprehensive open source framework to develop and apply state-of-the-art dmrg algorithms in electronic structure and beyond. *The Journal of Chemical Physics*, 159(23).

List of Figures

Appendices

.1 The *defmm* library

The *defmm* library[2] is a easy to use C++ implementation of the directional interpolation-based Fast Multipole Method exploiting equispaced interpolation combined with Fast Fourier Transforms. Mainly, *defmm* ensures a $\mathcal{O}(N)$ complexity independently of the particle distribution. Here, we provide an example of a short program calling *defmm*: only five lines are needed to construct and apply the FMM matrix to a vector.

```
#include "path to defmm/include/interface.hpp"
using namespace defmm;
int main(){

  const int  DIM   = 3     ; // Dimension
  const int  ORDER = 4     ; // Interpolation order
  const int  NCRIT = 32    ; // Number of particle per leaf cell
  const flt  KAPPA = 0.    ; // Wavenumber (for oscillatory kernels)
  const int  N     = 41334; // Number of points

  // Get random charge vector
  Vecc Q(N), P(N);
  for(int n = 0; n < N; n++){
    Q[n] = cplx(urand);}

  IBFMM_Mat<DIM> A;                          // FMM matrix
  A.addSourceParticlesINP("Y.inp",N); // Read source particles in Y.inp
  A.addTargetParticlesINP("X.inp",N); // Read target particles in X.inp
  A.prcmpt(ORDER,NCRIT,KAPPA);               // Precompute
  gemv(A,Q,P);                            // Compute P = A Q

  return 0;
}
```

As a header-only library, *defmm* does not need to be compiled before calling. However, the library calls both BLAS and the FFTW3 library [31]. Input files for the listing of source and target particles (that can be the same) are given as a sequence of particle coordinates (one particle per line, coordinates separated by blanks).

.2 The Hartree-Fock exchange

The efficient construction of the long-range exchange matrix in the Fock matrix is also interesting [62, 79, 79] and it is considered more expensive than the Coulomb matrix. This matrix is calculated by using the long-range two-electron integrals tensor $\mathcal{B}^{lr}$. The long-range exchange matrix is given by:

$$\mathbf{K}^{lr}(\mu;\nu) = 2\sum_{j=1}^{d}\sum_{\lambda,\kappa=1}^{N_b^2} q_{j\lambda}q_{j\kappa}\mathcal{B}^{lr}(\mu;\lambda;\kappa;\nu), \mu,\nu \in [N_b]\,, \tag{109}$$

with $q_{j\lambda}$, $q_{j\kappa}$, and d being defined in Section 2.5. Using the long-range two-electron integrals tensor $\mathcal{B}^{lr}$, The evaluation of $\mathbf{K}^{lr}(\mu;\nu)$ costs $\mathcal{O}(N_b^2 d)$. One can use the factorized structure $\mathbf{B}^{lr}_{LTEI-TA}$ defined in (2.45) to reduce the computational cost to $\mathcal{O}(Nd(N_b + N_{q_1}N^{\frac{1}{3}})$ for LTEI-TA approach with N being the number of Chebyshev interpolation points and

[2] https://github.com/IChollet/defmm

N_{q_1} being the number of quadrature points. We obtain the following representation

$$\mathbf{K}^{lr}_{LTEI-TA}(\mu;\nu) = 2\sum_{j=1}^{d}\left(\sum_{i=1}^{N_{q_1}} w_i \left(\mathbf{Q}_j \mathbf{M}^{(i)}_{TA,\mu}\right) \bigotimes_{l=1}^{3}{}_K \mathbf{A}^{(i)} \left(\mathbf{Q}_j \mathbf{M}^{(i)}_{TA,\nu}\right)^*\right), \tag{110}$$

where for a fixed $j \in [d]$ and $\lambda \in [N_b]$, we have $\mathbf{Q}_j \in \mathbb{R}^{N_b}$ and $(\mathbf{Q}_j)_\lambda = q_{j\lambda}$. The matrices $\mathbf{M}^{(i)}_{TA,\mu} \in \mathbb{R}^{N_b \times N_i^3}, i \in [N_{q_1}]$ are obtained by fixing the index $\mu \in [N_b]$ in the tensorized representation of $\mathbf{M}^{(i)}_{TA} \in \mathbb{R}^{N_b^2 \times N_i^3}$. These tensor representations are denoted by $\mathcal{M}^{(i)}_{TA} \in \mathbb{R}^{N_b \times N_b \times N_i^3}$ such that

$$\mathcal{M}^{(i)}_{TA}[\mu, :, :] = \mathbf{M}^{(i)}_{TA,\mu}. \tag{111}$$

Figure 14 displays the execution times required to evaluate the long-range exchange matrix (109) with respect to the number of basis functions N_b, for small values of $\omega \in \{0.05, 0.1\}$. We impose that the relative error of LTEI-TA approach for this evaluation is smaller than 10^{-5} and we compare the running times between a direct computation of (109) given $\mathbf{B}^{lr} \in \mathbb{R}^{N_b^2 \times N_b^2}$ and the factorized structure of $\mathbf{B}^{lr}$ using $\mathbf{B}^{lr}_{LTEI-TA}$. It can be seen that in the case of small values of ω, we notice that a faster construction of (109) is obtained through LTEI-TA. Compression techniques introduced in Section 2.6.1, can be used here to get better running times.

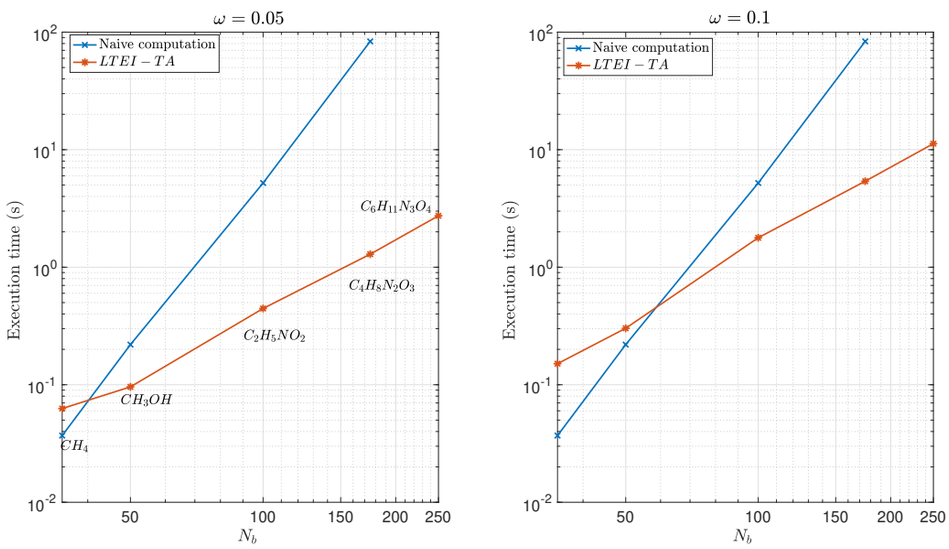


Figure 14: Execution time(s) required for the evaluation of (2.64) using the TEI tensor $\mathbf{B}^{lr}$ for different values of N_b, for $\omega = 0.05$ and $\omega = 0.1$ with imposed relative error smaller than 10^{-5}.

.3 Method 1: Exact construction of the TTO representation of the quantum chemical Hamiltonian

The quantum chemical Hamiltonian is given as follows: for simplicity we consider the Hamiltonian operator expressed in the spin-orbital basis, as defined in (1.130).

$$\mathbf{H} = \sum_{ij=1}^{d} h_{ij}\mathbf{A}_i^*\mathbf{A}_j + \sum_{ijkl=1}^{d} v_{ijkl}\mathbf{A}_i^*\mathbf{A}_k^*\mathbf{A}_l\mathbf{A}_j \tag{112}$$

We start by partitioning the system into left $L_{\leq s} = \{1, \ldots, s\}$ and right parts $R_{>s} = \{s+1, \ldots, d\}$. From this partitioning, we have four different partitions for the first term in the sum in (112): $(L_{\leq s}, L_{\leq s}), (R_{>s}, R_{>s}), (L_{\leq s}, R_{>s}), (R_{>s}, R_{>s})$, as well as 16 different partitions for the second term in the sum: $(L_{\leq s}, L_{\leq s}, L_{\leq s}, L_{\leq s})$, $(R_{>s}, R_{>s}, R_{>s}, R_{>s})$, $(R_{>s}, L_{\leq s}, L_{\leq s}, L_{\leq s})$, $(L_{\leq s}, R_{>s}, L_{\leq s}, L_{\leq s})$, $(L_{\leq s}, L_{\leq s}, R_{>s}, L_{\leq s})$, $(L_{\leq s}, L_{\leq s}, L_{\leq s}, R_{>s})$, $(L_{\leq s}, R_{>s}, R_{>s}, R_{>s})$, $(R_{>s}, L_{\leq s}, R_{>s}, R_{>s})$, $(R_{>s}, R_{>s}, L_{\leq s}, R_{>s})$, $(R_{>s}, R_{>s}, R_{>s}, L_{\leq s})$, $(L_{\leq s}, L_{\leq s}, R_{>s}, R_{>s})$, $(L_{\leq s}, R_{>s}, L_{\leq s}, R_{>s})$, $(L_{\leq s}, R_{>s}, R_{>s}, L_{\leq s})$,

$(R_{>s}, L_{\leq s}, L_{\leq s}, R_{>s})$, $(R_{>s}, R_{>s}, L_{\leq s}, L_{\leq s})$, $(R_{>s}, L_{\leq s}, R_{>s}, L_{\leq s})$. By exploiting the anti-commutation relations between the creation and annihilation operators, as well as the symmetries in the one-electron and two-electron integrals, we arrive at the following expression for the Hamiltonian operator

$$\begin{aligned}\mathbf{H} = &\sum_{ij\in L_{\leq s}} h_{ij}\mathbf{A}_i^*\mathbf{A}_j + \frac{1}{2}\sum_{ijkl\in L_{\leq s}} v_{ijkl}\mathbf{A}_i^*\mathbf{A}_j^*\mathbf{A}_l\mathbf{A}_k \\ &+ \sum_{ij\in R_{>s}} h_{ij}\mathbf{A}_i^*\mathbf{A}_j + \frac{1}{2}\sum_{ijkl\in R_{>s}} v_{ijkl}\mathbf{A}_i^*\mathbf{A}_j^*\mathbf{A}_l\mathbf{A}_k + \text{Interactions}^{(L_{\leq s}, R_{>s})},\end{aligned} \tag{113}$$

where

$$\begin{aligned}\text{Interactions}^{(L_{\leq s}, R_{>s})} = \frac{1}{2}\Bigg[&\sum_{i\in L_{\leq s}, j\in R_{>s},} h_{ij}\mathbf{A}_i^*\mathbf{A}_j + \sum_{i\in L_{\leq s}, j\in R_{>s}} h_{ji}\mathbf{A}_j^*\mathbf{A}_i \\ &+ \sum_{(i,k)\in L_{\leq s},(l,j)\in R_{>s},} v_{ijkl}\mathbf{A}_i^*\mathbf{A}_k^*\mathbf{A}_l\mathbf{A}_j + \sum_{(i,k)\in L_{\leq s},(l,j)\in R_{>s}} v_{ijkl}\mathbf{A}_l^*\mathbf{A}_j^*\mathbf{A}_i\mathbf{A}_k \\ &+ \sum_{(i,k)\in L_{\leq s},(l,j)\in R_{>s}} (v_{iklj} - v_{kijl})\mathbf{A}_i^*\mathbf{A}_k\mathbf{A}_l^*\mathbf{A}_j + \sum_{(i,k)\in L_{\leq s},(l,j)\in R_{>s}} \Big(v_{iklj} - v_{kijl})\mathbf{A}_j^*\mathbf{A}_l\mathbf{A}_k^*\mathbf{A}_i \\ &+ \sum_{i\in L_{\leq s},(j,k,l)\in R_{>s}} (v_{iklj} - v_{kilj})\,\mathbf{A}_i^*\mathbf{A}_k^*\mathbf{A}_l\mathbf{A}_j + \sum_{i\in L_{\leq s},(j,k,l)\in R_{>s}} (v_{iklj} - v_{kilj})\,\mathbf{A}_j^*\mathbf{A}_l^*\mathbf{A}_k\mathbf{A}_i \\ &+ \sum_{i\in R_{>s},(j,k,l)\in L_{\leq s}} (v_{iklj} - v_{kilj})\,\mathbf{A}_i^*\mathbf{A}_k^*\mathbf{A}_l\mathbf{A}_j + \sum_{i\in R_{>s},(j,k,l)\in L_{\leq s}} (v_{iklj} - v_{kilj})\,\mathbf{A}_j^*\mathbf{A}_l^*\mathbf{A}_k\mathbf{A}_i.\end{aligned} \tag{114}$$

Going from one partition $R_{>s}$ to the previous one $R_{>s-1}$ (for example from right to left) gives rise to the following recursive relation

$$\mathbf{H}^{R_{>s-1}} = \mathbf{I}_2 \otimes_K \hat{\mathbf{H}}^{R_{>s}} + \hat{\mathbf{H}}^s \otimes_K \mathbf{I}_{2^{d-s}} + \text{Interactions}^{(s,R_{>s})}, \tag{115}$$

where

$$\hat{\mathbf{H}}^{R_{>s}} = \sum_{ij\in R_{>s}} h_{ij}(\mathbf{A}_i^*)^{>s}(\mathbf{A}_j)^{>s} + \frac{1}{2}\sum_{ijkl\in R_{>s}} v_{ijkl}(\mathbf{A}_i^*)^{>s}(\mathbf{A}_k^*)^{>s}(\mathbf{A}_l)^{>s}(\mathbf{A}_j)^{>s}, \tag{116}$$

with $(\mathbf{A}_i^*)^{>s}$ being defined in (1.128). and

$$\hat{\mathbf{H}}^s = h_{ss}(\mathbf{A}^*\mathbf{A}) + \frac{1}{2}v_{ssss}(\mathbf{A}^*\mathbf{A}^*\mathbf{A}\mathbf{A}), \tag{117}$$

with $\mathbf{A}$ being defined in (1.125).

Matrices $\mathbf{A}_i^*$ and $\mathbf{A}_i$, as defined in (1.124), can be viewed as a TTO representation of rank one. Each term in the sums presented in (115) corresponds to a TTO decomposition with TTO-ranks equal to one. When applying arithmetic operations between sums of TTO representations, one ultimately obtains a TTO representation. However, it is not an optimal representation, as some of the terms in the TT-cores may be redundant.

Let $(\mathcal{H}_1, \ldots, \mathcal{H}_d)$ be the TTO decomposition of $\mathbf{H}$. As a reminder, the recursive relation in (115) using the TT-cores of a TTO representation have the following expression

$$\mathbf{H}^{R_{>s-1}}[\beta_s, :] = \sum_{\beta_s=1}^{R_s} \mathbf{H}_s(\beta_{s-1}; \beta_s) \otimes_K \mathbf{H}^{R_{>s}}[\beta_s, :], \tag{118}$$

with

$$\mathbf{H}^{R_{>s}}[\beta_s, :] = \sum_{\beta_s=1}^{R_s} \cdots \sum_{\beta_{d-1}=1}^{R_{d-1}} \mathbf{H}_{s+1}(\beta_k; \beta_{s+1}) \otimes_K \ldots \otimes_K \mathbf{H}_d(\beta_{d-1}; 1) \in \mathbb{R}^{n^{d-s}\times n^{d-s}}.$$

where $\mathbf{H}_s(\beta_s; \beta_{s+1}) := \mathcal{H}_s[\beta_s, :, :, \beta_{s+1}]$. Let $\mathbf{H}_s^{<2>} \in \mathbb{R}^{R_{s-1}n\times nR_s}$ be the mode-(1:2) matricization of $\mathcal{H}_s$. We have the following alternative expression of $\mathbf{H}^{R_{>s-1}}$

$$\mathbf{H}^{R_{>s-1}} = \mathbf{H}_s^{<2>} \bowtie \mathbf{H}^{R_{>s}}, \tag{119}$$

with $\mathbf{H}^{R_{>s}} \in \mathbb{R}^{R_{s+1}n^{d-s-1}\times n^{d-s-1}}$.

Now, with these recursive relations in mind, the goal is to derive the expressions for both the entries of $\mathbf{H}^{R_{>s}}$ and the TT-core $\mathbf{H}_s^{<2>}$. To illustrate this, let us consider an example in the case of $d = 4$ and $s = 1$, focusing only on the one-body term that we denote by $\mathbf{H}_{(1)}$, i.e the first sum in (112).

$$\begin{aligned}\mathbf{H}_{(1)} = \sum_{ij=1}^{4} h_{ij}\mathbf{A}_i^*\mathbf{A}_j =& h_{11}\mathbf{A}^*\mathbf{A} \otimes_K (\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{I}) + \mathbf{I} \otimes_K (h_{22}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I}) + \mathbf{I} \otimes_K (\mathbf{I} \otimes_K h_{33}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I}) \\ &+ h_{12}\mathbf{A}^* \otimes_K (\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I}) + h_{21}\mathbf{A} \otimes_K (\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{I}) + h_{13}\mathbf{A}^* \otimes_K (\mathbf{I} \otimes_K \mathbf{A} \otimes_K \mathbf{I}) \\ &+ h_{31}\mathbf{A} \otimes_K (\mathbf{I} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I}) + \mathbf{I} \otimes_K (h_{32}\mathbf{A} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I}) + \mathbf{I} \otimes_K (h_{23}\mathbf{A}^* \otimes_K \mathbf{A} \otimes_K \mathbf{I}) \\ &+ h_{14}\mathbf{A}^* \otimes_K (\mathbf{I} \otimes_K \mathbf{I} \otimes \mathbf{A}) + \mathbf{I} \otimes_K (h_{24}\mathbf{A}^* \otimes_K \mathbf{I} \otimes \mathbf{A}) + \mathbf{I} \otimes_K (\mathbf{I} \otimes_K h_{34}\mathbf{A}^* \otimes_K \mathbf{A}) \\ &+ h_{41}\mathbf{A} \otimes_K (\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^*) + \mathbf{I} \otimes_K (h_{42}\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^*) + \mathbf{I} \otimes_K (\mathbf{I} \otimes_K h_{43}\mathbf{A} \otimes_K \mathbf{A}^*) \\ &+ \mathbf{I} \otimes_K (\mathbf{I} \otimes_K \mathbf{I} \otimes_K h_{44}\mathbf{A}^*\mathbf{A}),\end{aligned} \tag{120}$$

For $s = 1$, we have

$$\mathbf{H}^{R_{>0}} = \mathbf{H}_{(1)} = \mathbf{H}_1^{<2>} \bowtie \mathbf{H}^{R_{>1}}. \tag{121}$$

Here it is clear that

$$\begin{aligned}\mathbf{H}^{R_{>1}} = \Big[&(\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{I}), (h_{22}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I}), (\mathbf{I} \otimes_K h_{33}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I}), (\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I}), (\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{I}), \\ &(\mathbf{I} \otimes_K \mathbf{A} \otimes_K \mathbf{I}), (\mathbf{I} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I}), (h_{32}\mathbf{A} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I}), (h_{23}\mathbf{A}^* \otimes_K \mathbf{A} \otimes_K \mathbf{I}), (\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{A}), \\ &(h_{24}\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{A}), \mathbf{I} \otimes_K h_{34}\mathbf{A}^* \otimes_K \mathbf{A}), (\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^*), (h_{42}\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^*), (\mathbf{I} \otimes_K h_{43}\mathbf{A} \otimes_K \mathbf{A}^*),\end{aligned} \tag{122}$$

$$(\mathbf{I} \otimes_K \mathbf{I} \otimes_K h_{44}\mathbf{A}^*\mathbf{A})\Big]^*. \tag{123}$$

and the first TT-core denoted by $\mathbf{H}_1^{<2>}$ is given by:

$$\mathbf{H}_1^{<2>} = \Big[h_{11}\mathbf{A}^*\mathbf{A}, \mathbf{I}, \mathbf{I}, h_{12}\mathbf{A}^*, h_{21}\mathbf{A}, h_{13}\mathbf{A}^*, h_{31}\mathbf{A}, \mathbf{I}, \mathbf{I}, h_{14}\mathbf{A}^*, \mathbf{I}, \mathbf{I}, h_{41}\mathbf{A}, \mathbf{I}, \mathbf{I}, \mathbf{I}\Big], \tag{124}$$

$\mathbf{H}_1^{<2>}$ has TTO-rank equals to 16, however, it is notable that there are many redundant terms in $\mathbf{H}_1^{<2>}$ that can be avoided by employing the following changes

$$\begin{aligned}\mathbf{H}^{R_{>1}} = \Big[&\underbrace{(\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{I})}_{\mathbf{I}_{22}}, (h_{22}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I} + \mathbf{I} \otimes_K h_{33}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} + h_{32}\mathbf{A} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I} \\ &+ h_{23}\mathbf{A}^* \otimes_K \mathbf{A} \otimes_K \mathbf{I} + h_{24}\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{A} + \mathbf{I} \otimes_K h_{34}\mathbf{A}^* \otimes_K \mathbf{A} + h_{42}\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^* \\ &+ \mathbf{I} \otimes_K h_{43}\mathbf{A} \otimes_K \mathbf{A}^* + \mathbf{I} \otimes_K \mathbf{I} \otimes_K h_{44}\mathbf{A}^*\mathbf{A}), (h_{12}\mathbf{A} \otimes_K \mathbf{I} + h_{13}\mathbf{I} \otimes_K \mathbf{A}), \\ &(h_{13}\mathbf{I} \otimes_K \mathbf{A} \otimes_K \mathbf{I} + h_{12}\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I} + h_{14}\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{A}), \\ &(h_{21}\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{I} + h_{31}\mathbf{I} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I} + h_{41}\mathbf{I} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^*)\Big]^*,\end{aligned} \tag{125}$$

and

$$\mathbf{H}_1^{<2>} = \Big[\underbrace{h_{11}\mathbf{A}^*\mathbf{A}}_{\hat{\mathbf{H}}^1}, \underbrace{\mathbf{I}}_{\mathbf{I}_2}, \mathbf{A}^*, \mathbf{A}\Big]. \tag{126}$$

According to Equation (115), we note that

$$\begin{aligned}\hat{\mathbf{H}}^{R_{>1}} = &h_{22}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{I} + \mathbf{I} \otimes_K h_{33}\mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I} + h_{32}\mathbf{A} \otimes_K \mathbf{A}^* \otimes_K \mathbf{I} \\ &+ h_{23}\mathbf{A}^* \otimes_K \mathbf{A} \otimes_K \mathbf{I} + h_{24}\mathbf{A}^* \otimes_K \mathbf{I} \otimes_K \mathbf{A} + \mathbf{I} \otimes_K h_{34}\mathbf{A}^* \otimes_K \mathbf{A} + h_{42}\mathbf{A} \otimes_K \mathbf{I} \otimes_K \mathbf{A}^* \\ &+ \mathbf{I} \otimes_K h_{43}\mathbf{A} \otimes_K \mathbf{A}^* + \mathbf{I} \otimes_K \mathbf{I} \otimes_K h_{44}\mathbf{A}^*\mathbf{A},\end{aligned} \tag{127}$$

This leads to a TTO-rank equal to 4, effectively reducing the rank of the TT-representation. Therefore, the approach is to combine terms that share common right/left operators, resulting in what is referred to as a complementary operator, see [17]. While constructing these complementary operators, particularly for the interaction term in the quantum chemical Hamiltonian operator, can be considerably more complicated, the authors in [17] have developed a concise formalism for (114).

$$
\begin{aligned}
Interaction^{(L_{\leq s},R_{>s})} = & \frac{1}{2}\left(\sum_{i\in L_{\leq s}} \mathbf{A}_i^* \mathbf{S}_i^{R_{>s}} + \text{ h.c. } + \sum_{i\in R_{>s}} \mathbf{A}_i^* \mathbf{S}_i^{L_{\leq s}} + \text{ h.c. }\right) \\
& + \frac{1}{2}\left(\sum_{ij\in L_{\leq s}} \mathbf{O}_{ij}^{L_{\leq s}} \mathbf{P}_{ij}^{R_{>s}} + \text{ h.c. }\right) \\
& - \frac{1}{2}\left(\sum_{ij\in L_{\leq s}} \mathbf{B}_{ij}^{L_{\leq s}} \mathbf{Q}_{ij}^{R_{>s}} + h.c.\right),
\end{aligned} \tag{128}
$$

where the complementary operators are defined as:

- $\mathbf{S}_i^{L_{\leq s}/R_{>s}} = \sum_{j\in L_{\leq s}/R_{>s}} h_{ij}\mathbf{A}_j + \sum_{jkl\in L_{\leq s}/R_{>s}} (v_{ijkl} - v_{jikl})\mathbf{A}_j^*\mathbf{A}_k\mathbf{A}_l$
- $\mathbf{O}_{ij} = \mathbf{A}_i^*\mathbf{A}_j^*$.
- $\mathbf{B}_{ij} = \mathbf{A}_i^*\mathbf{A}_j$
- $\mathbf{P}_{ij}^{R_{>s}} = \sum_{kl\in R_{>s}} v_{ijkl}\mathbf{A}_k\mathbf{A}_l$
- $\mathbf{Q}_{ij}^{R_{>s}} = \sum_{kl\in R_{>s}} \left(v_{iklj} - v_{kilj}\right)\mathbf{A}_k^*\mathbf{A}_l$.

with $h.c$ referring to hermitian conjugate. By noting that

$$
\mathbf{P}_{ij}^{R_{>s-1}} = \mathbf{I}_2 \otimes_K \hat{\mathbf{P}}_{ij}^{R_{>s}} + \hat{\mathbf{P}}_{ij}^{s} \otimes_K \mathbf{I}_{2^{d-s}} + \sum_{k\in R_{>s}} v_{ijkl}\mathbf{A} \otimes_K (\mathbf{A}_k)^{>s}, \tag{129}
$$

$\mathbf{H}^{R_{>s-1}}$ can be represented using the following strong Kronecker product

$$
\begin{pmatrix}
\mathbf{H}^{R_{>s-1}} \\
\left[\mathbf{P}_{ij}^{R_{>s-1}}\right]_{i,j=s-1} \\
\left[\mathbf{S}_i^{R_{>s-1}}\right]_{i=s-1} \\
\vdots \\
\left[\mathbf{Q}_{ij}^{R_{>s-1}}\right]_{i,j=s-1} \\
\mathbf{I}_{2^{d-s+1}}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{I}_2 & \hat{\mathbf{C}}^s & \hat{\mathbf{H}}^s \\
\mathbf{0} & \hat{\mathbf{P}}_1{}^s & \hat{\mathbf{P}}_2{}^s \\
\mathbf{0} & \hat{\mathbf{S}}_1{}^s & \hat{\mathbf{S}}_2{}^s \\
\vdots & & \\
\mathbf{0} & \hat{\mathbf{Q}}_1{}^s & \hat{\mathbf{Q}}_2{}^s \\
\mathbf{0} & \mathbf{0} & \mathbf{I}_2
\end{pmatrix}
\bowtie
\begin{pmatrix}
\hat{\mathbf{H}}^{R_{>s}} \\
\left[\mathbf{P}_{ij}^{R_{>s}}\right]_{i,j=s} \\
\left[\mathbf{S}_i^{R_{>s}}\right]_{i=s} \\
\vdots \\
\left[\mathbf{Q}_{ij}^{R_{>s}}\right]_{i,j=s} \\
\hat{\mathbf{I}}_{2^{d-s}}
\end{pmatrix}. \tag{130}
$$

with $\bowtie$ being defined in (1.144) and the terms in (130) can be found by using the recursion rules as explained in [16, 17].

.4 Method 2: Exact construction of the TTO representation of a 1-body operator

Consider the following simple operator

$$
\mathbf{O} = \sum_{i=1}^{d} t_i \mathbf{A}_i^* \mathbf{A}_i, t_i \in \mathbb{R}. \tag{131}
$$

Let us consider the case of $d = 2$ such that

$$
\mathbf{O} = t_1\mathbf{A}_1^*\mathbf{A}_1 + t_2\mathbf{A}_2^*\mathbf{A}_2, \tag{132}
$$

with

$$
\mathbf{A}_1^*\mathbf{A}_1 = \mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I}_2,\ \mathbf{A}_2^*\mathbf{A}_2 = \mathbf{I}_2 \otimes_K \mathbf{A}^*\mathbf{A} \tag{133}
$$

We can write then

$$
\mathbf{O} = \begin{bmatrix} \mathbf{I}_2 & t_1\mathbf{A}^*\mathbf{A} \end{bmatrix} \bowtie \begin{bmatrix} t_2\mathbf{A}^*\mathbf{A} \\ \mathbf{I}_2 \end{bmatrix}. \tag{134}
$$

Now, for $d = 3$, we have

$$
\mathbf{A}_1^*\mathbf{A}_1 = \mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I}_2 \otimes_K \mathbf{I}_2,\ \mathbf{A}_2^*\mathbf{A}_2 = \mathbf{I}_2 \otimes_K \mathbf{A}^*\mathbf{A} \otimes_K \mathbf{I}_2,\ \mathbf{A}_3 = \mathbf{I}_2 \otimes_K \mathbf{I}_2 \otimes_K \mathbf{A}^*\mathbf{A}, \tag{135}
$$

which yields to

$$\mathbf{O} = \begin{bmatrix} \mathbf{I}_2 & t_1\mathbf{A}^*\mathbf{A} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{I}_2 & t_2\mathbf{A}^*\mathbf{A} \\ 0 & \mathbf{I}_2 \end{bmatrix} \bowtie \begin{bmatrix} t_3\mathbf{A}^*\mathbf{A} \\ \mathbf{I}_2. \end{bmatrix} \tag{136}$$

The general case for $d \in \mathbb{N}$ can be expressed as follows:

$$\mathbf{O} = \underbrace{\begin{bmatrix} \mathbf{I}_2 & t_1\mathbf{A}^*\mathbf{A} \end{bmatrix}}_{\mathbf{H}_1} \bowtie \underbrace{\begin{bmatrix} \mathbf{I}_2 & t_2\mathbf{A}^*\mathbf{A} \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{H}_2} \bowtie .... \bowtie \underbrace{\begin{bmatrix} t_d\mathbf{A}^*\mathbf{A} \\ \mathbf{I}_2 \end{bmatrix}}_{\mathbf{H}_d}. \tag{137}$$

$\mathbf{O}$ has a TTO representation of TTO-ranks 2.

.5 Particle number preserving MPS

The goal of this appendix is to provide a graphical perspective on the block-sparse structure of the TT decomposition of a particle-preserving wavefunction, as described in [77]. We find this interpretation to be particularly intriguing. We show that the *target* wavefunction, representing a system with fixed N particles, can be constructed through a recursive chain of orthogonal transformations using the TT-cores. The latter can be viewed as linear maps. Then we delve into the characterization of these linear maps and aim to derive the minimal TT-ranks based on the graphical representation.

A general N-electron wavefunction, denoted as $\mathbf{\Psi}^N \in \boldsymbol{\mathcal{V}}_N^d \subset \boldsymbol{\mathcal{F}}_d$, with $\boldsymbol{\mathcal{F}}_d$ being defined in (1.97), satisfies

$$\mathbf{N}\mathbf{\Psi}^N = N\mathbf{\Psi}^N, \tag{138}$$

where $n = 2$. $\mathbf{\Psi}^N$ can be expressed in the orthonormal basis as follows:

$$\mathbf{\Psi}^N = \sum_{\mu_1}^{n} \dots \sum_{\mu_d}^{n} \mathbf{\Psi}^N(\overline{\mu_1, \dots, \mu_d}) \boldsymbol{e}_{\mu_1} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_d} \in \mathbb{R}^{n^d}, \tag{139}$$

and

$$\mathbf{\Psi}^N(\overline{\mu_1, \dots, \mu_d}) = \begin{cases} \mathbf{\Psi}^N(\overline{\mu_1, \dots, \mu_d}), & \sum_{k=1}^d q_n(\mu_k) = N, \\ 0, & otherwise. \end{cases} \tag{140}$$

with the mapping function q_n being defined in (PN). Let $(\mathcal{U}_1, \dots, \mathcal{U}_d)$ be a left-orthogonal TT decomposition of $\mathbf{\Psi}^N$, see Definition 20, with TT-ranks $(r_1, \dots, r_d)$. Each coefficient from (139) can be written in terms of the TT-cores as follows:

$$\mathbf{\Psi}^N(\overline{\mu_1, \dots, \mu_d}) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} \mathbf{U}_1[\mu_1](1; \alpha_1)\mathbf{U}_2[\mu_2](\alpha_1; \alpha_2) \cdots \mathbf{U}_d[\mu_d](\alpha_{d-1}; 1), \tag{141}$$

where $\mathbf{U}_k[\mu_k] \in \mathbb{R}^{r_{k-1} \times r_k}$, $k \in [d]$, and $r_0 = r_d = 1$. It follows that

$$\mathbf{\Psi}^N = \sum_{\mu_1=1}^{n} \sum_{\mu_2=1}^{n} \cdots \sum_{\mu_d}^{n} \left[\sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} \mathbf{U}_1[\mu_1](1; \alpha_1)\mathbf{U}_2[\mu_2](\alpha_1; \alpha_2) \cdots \mathbf{U}_d[\mu_d](\alpha_{d-1}; 1) \right] \boldsymbol{e}_{\mu_1} \otimes_K \dots \otimes_K \boldsymbol{e}_{\mu_d}. \tag{142}$$

By rewriting (142) in the following way,

$$\mathbf{\Psi}^N = \left[\sum_{\alpha_{d-1}, \mu_d} \mathbf{U}_d[\mu_d](\alpha_{d-1}; 1) \cdots \left[\sum_{\alpha_1, \mu_2} \mathbf{U}_2[\mu_2](\alpha_1; \alpha_2) \left[\sum_{\mu_1} \mathbf{U}_1[\mu_1](1; \alpha_1) \boldsymbol{e}_{\mu_1} \right] \otimes_K \boldsymbol{e}_{\mu_2} \right] \otimes_K \cdots \otimes_K \boldsymbol{e}_{\mu_d} \right]. \tag{143}$$

One observes that, for $k \in [d]$, the TT-cores $\mathbf{U}_k^{<2>} \in \mathbb{R}^{nr_{k-1} \times r_k}$ can be regarded as linear maps defined as follows:

$$\mathbf{U}_k^{<2>} : \boldsymbol{\mathcal{W}}_k \to \boldsymbol{\mathcal{W}}_{k-1} \otimes_K \boldsymbol{e}_{\mu_k}. \tag{144}$$

We define the spaces $\boldsymbol{\mathcal{W}}_k$ as follows: the recursive chain towards reaching the target state $\mathbf{\Psi}^N$ can be elaborated as follows:

$$\mathbf{\Psi}_{1,\alpha_1}^{N_1} = \sum_{\mu_1} \mathbf{U}_1[\mu_1](1; \alpha_1)\boldsymbol{e}_{\mu_1} \in \mathbb{R}^n, \tag{145}$$

$$\mathbf{\Psi}_{2,\alpha_2}^{N_2} = \sum_{\mu_2, \alpha_1} \mathbf{U}_2[\mu_2](\alpha_1; \alpha_2)\mathbf{\Psi}_{1,\alpha_1}^{N_2} \otimes_K \boldsymbol{e}_{\mu_2} \in \mathbb{R}^{n^2}, \tag{146}$$

$$\vdots \tag{147}$$

$$\mathbf{\Psi}^N := \mathbf{\Psi}_{d,1}^{N_d} = \sum_{\mu_d, \alpha_{d-1}} \mathbf{U}_d[\mu_d](\alpha_{d-1}; 1)\mathbf{\Psi}_{d-1,\alpha_{d-1}}^{N_{d-1}} \otimes_K \boldsymbol{e}_{\mu_d} \in \mathbb{R}^{n^d}, \tag{148}$$

where $\sum_{k=1}^{d} N_k = N$, $\boldsymbol{\mathcal{W}}_k := \text{Span}\left\{\boldsymbol{\Psi}_{k,\alpha_k}^{N_k} \middle| \alpha_k \in [r_k]\right\} \subseteq \boldsymbol{\mathcal{F}}_k = \oplus_{N_k=0}^{k} \boldsymbol{\mathcal{V}}_{N_k}^{k}$ and $\dim(\boldsymbol{\mathcal{W}}_k) = r_k \leq \dim(\boldsymbol{\mathcal{F}}_k) = n^k$. For N-electron wavefunction, intermediate vectors, denoted here by $\boldsymbol{\Psi}_{k,\alpha_k}^{N_k}$ are required to be eigenvectors of the partial particle number operator as discussed in [2, 77]. The latter can be expressed as $\mathbf{N}_{\leq k} = \sum_{i,j=1}^{k} \mathbf{A}_i^* \mathbf{A}_j$, such that for $k \in [d]$

$$\mathbf{N}_{\leq k} \boldsymbol{\Psi}_{k,\alpha_k}^{N_k} = N_k \boldsymbol{\Psi}_{k,\alpha_k}^{N_k}, \tag{149}$$

This restriction over the particle number yields a block-sparse structure in the TT-cores of the TT representation as established in [2] and as described in [77], a graphical representation of the recursive transformation can be provided to illustrate the block-sparse structure and to derive the minimal TT-ranks. Suppose $d = 3$, we represent the following illustration of linear mapping between different $\boldsymbol{\mathcal{W}}_k$ spaces through matrices $\mathbf{U}_k^{<2>} \in \mathbb{R}^{nr_{k-1} \times r_k}$.

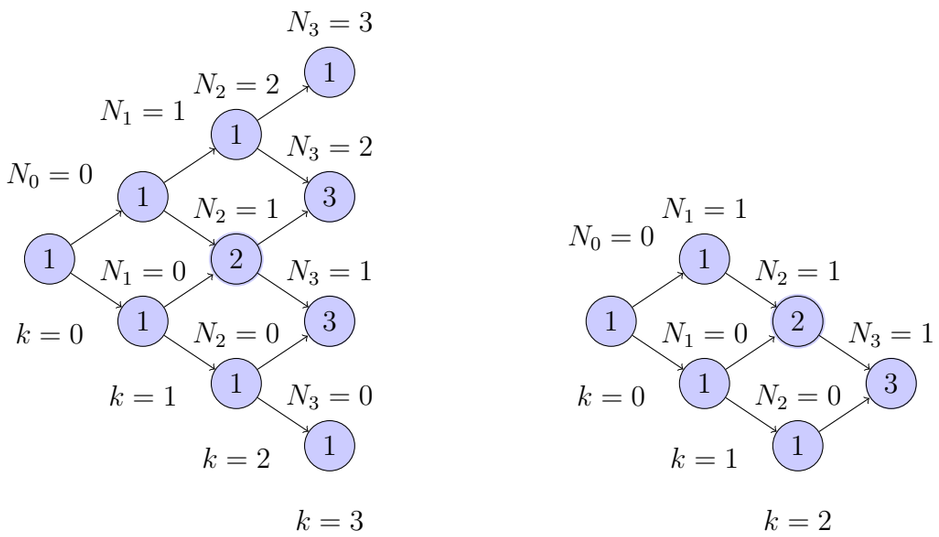


Figure 15: Graphical representation for $d = 3$

Figure 15(a) has a tree structure where each level corresponds to a spin-orbital k and each node represents the subspace $\boldsymbol{\mathcal{V}}_{N_k}^{k}$, labeled by its dimension, such that $\dim(\boldsymbol{\mathcal{V}}_{N_k}^{k}) = \begin{pmatrix} k \\ N_k \end{pmatrix}$. One can recognize that the arrows between different nodes in the graph represent the mapping from one state at k to a new state at $k+1$ through the TT-cores $\mathbf{U}_k[\mu_k] \in \mathbb{R}^{r_{k-1} \times r_k}$, i.e depending on whether the state k is occupied $\mu_k = 2$, $q_2(\mu_k) = 1$, or unoccupied $\mu_k = 1$, $q_2(\mu_k) = 0$, a new state in $k+1$, can be created or it remains the same. It is noted that without any restriction on the target particle number of the system, we have $\boldsymbol{\mathcal{W}}_k = \bigoplus_{N_k=1}^{k} \boldsymbol{\mathcal{V}}_k^{N_k}$ and therefore, the TT-ranks increase drastically with the number of orbitals d, here the TT-ranks, defined as the sum of labels of the nodes at level k, are $(r_1, r_2, r_3) = (2, 4, 9)$, if no truncation is made. Now suppose that the target particle number is $N := N_3 = 1$ while keeping $d = 3$. As shown in Figure 15(b), the TT-ranks r_k are reduced to $(2, 3, 3)$. As described in [77], it is interesting to investigate the minimal TT-rank r_k for a particle preserving state. To do so, one needs to consider a bipartite system: we divide our system into two distinct parts, namely left and right parts, such that at each iteration k

$$\boldsymbol{\Psi}^N = \sum_{\mu_1,\ldots\mu_k=1}^{n} \sum_{\mu_{k+1}\ldots\mu_d=1}^{n} \boldsymbol{\Psi}^N(\overline{\mu_1, \ldots \mu_k}; \overline{\mu_{k+1}, \ldots \mu_d}) \boldsymbol{e}_{\leq k} \otimes_K \boldsymbol{e}_{>k}, \tag{150}$$

where $\boldsymbol{e}_{\leq k} = \boldsymbol{e}_{\mu_1} \otimes_K \ldots \otimes_K \boldsymbol{e}_{\mu_k}$, $\boldsymbol{e}_{>k} = \boldsymbol{e}_{\mu_{k+1}} \otimes_K \ldots \otimes_K \boldsymbol{e}_{\mu_d}$. One can identify the minimal TT-rank r by the rank of the matrix $(\boldsymbol{\Psi}^N)^{<k>}$ of entries $\boldsymbol{\Psi}^N(\overline{\mu_1, \ldots \mu_k}; \overline{\mu_{k+1}, \ldots \mu_d})$. To determine the appropriate minimal TT-rank, we can employ graphical notations. Specifically, we set r as the minimum of r_k^L and r_k^R, where r_k^L represents the sum of the dimensions of subspaces $\boldsymbol{\mathcal{W}}_k$ obtained at level k while recursively constructing the target state from

left to right, as depicted in Figure 15(b). Similarly, r_k^R denotes the sum of dimensions for the subspaces $\boldsymbol{\mathcal{W}}_k$ at level k when constructing the target state from right to left, as shown in Figure 16(a). In Figure 16(b), we provide a graphical representation of the minimal TT-ranks for various subspaces by simply taking the minimum dimensions of left and right subspaces displayed in Figure 15(b) and Figure 16(b), respectively. In this context, the minimal TT-ranks are given as $(2, 2, 1)$.

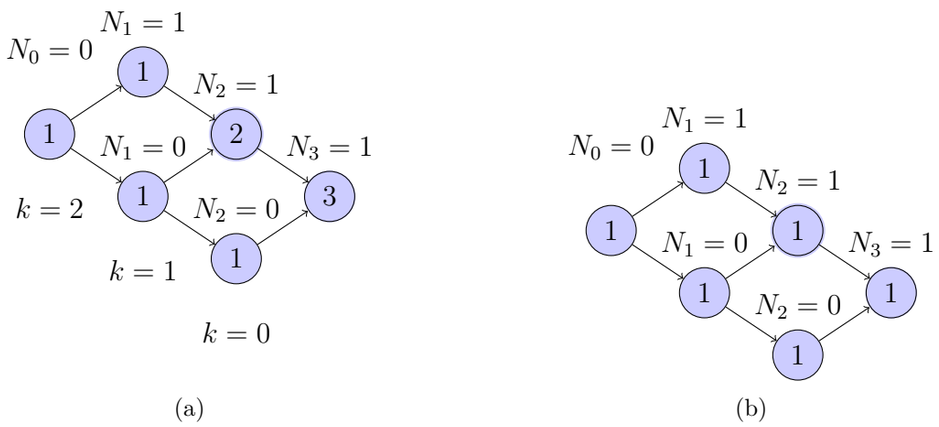


Figure 16: Graphical representation for $d = 3$

.6 Proof-of-concept of a QC-DMRG library

We present below a diagram that provides a fundamental overview of key structures within our proof-of-concept library, showcasing their respective fields and illustrating the different relationships between them. Each struct is described in the following

- **Dmrg:** includes *ksites* which denotes the number of sites or the number of spin orbitals, denoted previously by d_{spin} , *mpo* of type **Mpo** which corresponds to the TT representation of the Hamiltonian operator used in the calculation, while *mps* of type **Mps** stands for the TT representation of the eigenfunction. *left_tensors* and *right_tensors* are components obtained through contraction operations performed within QC-DMRG sweeps. They are illustrated in Figure 1.11 as $\mathcal{G}^{\leq k-1}$ and $\mathcal{G}^{>k-1}$ for a 1-site QC-DMRG at fixed $k \in [d]$ and are represented as lists of objects of type **BlockContractions**. Additionally, the *qf* of type **QuadraticForm** and *indices*, list of objects of type **IndexDmrg**, attributes contain matching block indices used for calculations within QC-DMRG sweeps, such as updating left and right components and solving local eigenvalue problems that involve multiple dense block matrix multiplications.

- **Mpo:** includes the number of sites *ksites*, the number of interactions p (where p equals to 2 for a 2-body Hamiltonian model) and w represents the list of TT-cores. Each TT-core follows a block sparse structure as elaborated in Section 3.3.1 and is represented as a collection of objects of type **BlockDiagonalMpo**.

- **Mps:** includes the number of sites *ksites*, the number of particles *nparticles* and x represents a list of TT-cores. Each TT-core follows a block sparse structure as elaborated in Section 3.3.1 and is represented as a collection of objects of type **BlockDiagonalMps**.

- **BlockDiagonalMpo/BlockDiagonalMps** includes the variable *diagonal_number*, where *diagonal_number* $\in \{1, 2\}$ for *mps* and $\in \{1, 2, 3, 4\}$ for *mpo*. It also depends on the parameter *nblocks*, which represents the number of blocks per *diagonal_number*. For *mps*, the maximum number of blocks is *nparticles* $+ 1$, and for *mpo*, it is $2p + 1$. Within this struct, we have a list of dense matrices *mat*, which serves as the representation of the blocks in the sparse block TT-cores. Additionally, *block_index* indicates the position of these blocks, and *tt_coresizes* is of type **Dict** (dictionary) and specifies the size of each block.

- **DiagonalQuadraticForm:** includes the position of blocks *block_indices*, and their respective sizes, referred to as *block_sizes*. The latter are computed using block Kronecker products between the TT-cores of *mps* and *mpo* at each site. The variable *diagonal_number* $\in \{1, 2\}$, will be elaborated upon further in Section 4.3. It is important to note that we do not store the resulting matrices from these Kronecker products, except when required as explained in Section 4.3. Instead, we retain only the corresponding block indices in memory. Further explanation of this can be found in Section 4.3.

- **QuadraticForm:** includes a list of objects of length *ksites* denoted by *block_info* and of type **DiagonalQuadraticForm**.

- **BlockContractions:** is caracterized by a *diagonal_number* and a vector of dense matrices denoted by *contraction*. These dense matrices play a crucial role in the construction of $\mathcal{G}^{\leq k-1}$ or $\mathcal{G}^{>k-1}$ (for the 1-site QC-DMRG) or $\mathcal{G}^{>k}$ (for the 2-site QC-DMRG) at a selected site $k \in [d]$.

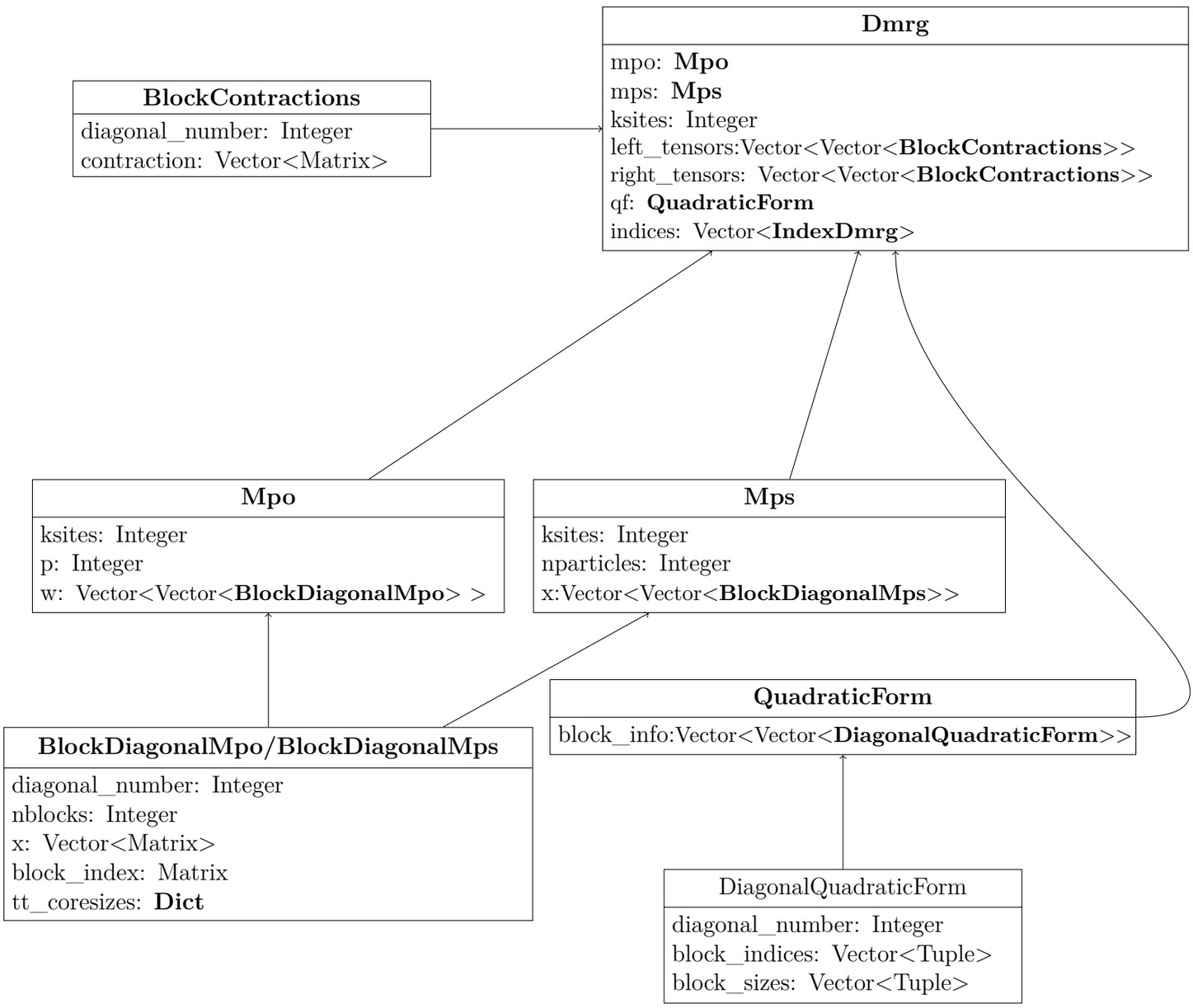


Figure 17: Visualization of key structs and their interactions in our QC-DMRG proof-of-concept library.