UiO : Department of Physics
University of Oslo

# Quantum Dynamics

Many-body methods and basis sets

**Halvard Sutterud**

Master's Thesis, Autumn 2020

# Abstract

Accurate descriptions of many-particle quantum systems subject to laser interactions can be found using real-time ab initio methods. Of these the arguably most popular and exact is the multi-configuration time-dependent Hartree-Fock (MCTDHF) method. However, MCTDHF suffers from computational limitations in that it quickly becomes too time consuming. The orbital-adaptive time-dependent coupled-cluster (OATDCC) method represents a hierarchy of approximations to MCTDHF that are less computationally expensive while retaining as much accuracy as possible. Building on an existing codebase we have in this thesis generalized the OATDCC method to include Q-space orbital equations. A novel ground state solver is implemented, employing adiabatic switching, since imaginary time propagation is not feasible "out of the box". Furthermore, we implement a sinc-discrete variable representation basis for one-dimensional model systems. We demonstrate that ionization and high-harmonic processes can be described using the OATDCCD method. Comparison with the more accurate, yet more expensive, multiconfigurational time-dependent Hartree-Fock method indicates that the OATDCCD method is an excellent approximation.

# Acknowledgements

First and foremost, I would like to thank my crew of supervisors for excellent supervision. Thank you Morten, for introducing me to the field of Computational physics. Passion for teaching, work capasity and hospitality are just a few of the qualities that I find so inspiring about you. Thank you to Øyvind, Sebastian and Håkon for fruitful discussions, and for your help and support with forming this thesis. Especially Sebastian, because he watched me write this.

To my family, who have always been extremely supportive, understanding and kind. To my dad, Torfinn, I want to add that this thesis is a confirmation of your theory that productivity is inverse proportional with remaining time. And my brother Lars, you deserve a special thanks for your incredible flexibilty and ability to help me solve unforeseen problems on short notice here in Oslo. I am, however, sure that any of my family members would do the same if in Oslo, as I would likewise do the same to help any of you.

I would like to thank the different communities that I have been part of during my time at the University of Oslo. Realistforeningen is perfect for people like me, a.k.a. nerds who like beer. I really want to thank the people at the study hall Lillefy during my time there, which includes most of the time of the three years of my bachelor. I am amazed how advanced a game Uno ended up being in the end. And last but not least, the community of master students searching for problems to throw computers at in the Computational Physics group. You helped me hone important skills in Super Smash Bros, Mario Kart and later, in the more cultured premises of the Centre for Computing in Science Education, table tennis.

Thank you to my choir as of last year, Den Norske Studentersangforening, for providing me with the perfect combination of excellent musical quality, camaraderie and beer.

Thank you to my flat mate, Ivar, for introducing me to great text editors, music and games, and for our good times together these last years. You seem to always know about the most obscure things.

Finally, I want to give an extra special thank you to Emilie, for providing my life with structure and purpose, and for handling my impulsiveness and inability to plan for the unforeseen (and the could-have-been-foreseen...). I am incredibly thankful for the time I have had with you so far, and I look forward to a long life together.

# Contents

# List of Figures

# List of Tables

# CHAPTER 1

# Introduction

Since the discovery of the Schrödinger equation in 1926, the theory of quantum mechanics has been extremely effective at describing the smallest constituents of our universe. Quantum mechanics is the fundamental theory of nature in the realm of small time and length scales, where the behaviour of objects takes unintuitive forms. The central equation in quantum mechanics is the Schrödinger equation[42],

$$i\hbar\frac{\partial}{\partial t}\left|\Psi(t)\right\rangle = \hat{H}(t)\left|\Psi(t)\right\rangle . \tag{1.1}$$

It governs the wave-like behaviour of matter at the quantum scales in terms of the time-dependent wavefunction $|\Psi(t)\rangle$. Depending on the contents of the Hamiltonian operator $\hat{H}(t)$, the wave function can be made to describe any quantum system, from a single particle in one dimension to many-particle systems with complex external interactions. Solving the Schrödinger equation then involves either finding the stationary properties of quantum systems such as atoms and molecules, or finding their time dependence given a pertubation or interaction.

For systems larger than the hydrogen atom, there are very few closed form solutions for the Schrödinger equation. However, the massive increase in computational power over the last half century has allowed for the study of more and more complicated systems using numerical methods.

Even then, the exact solution of the Schrödinger equation is not possible for even moderately sized systems. For any numerical method, an approximation has to be made to be able to represent the solution on a computer, which has finite memory and storage. The complexity of a quantum system also increases exponentially with the number of particles, as the wavefunction couples the description of every particle at every point in space. With great complexity comes great computational cost, in terms of both CPU hours and memory storage, and finding the correct approximation to reduce the cost while simultaneously retaining an accurate description of the important many-body effects can be challenging.

Such approximative methods must also be able to model phenomena that are relevant and interesting. One particularly challenging field of numerical study is strong-field processes. The interaction between electronic systems and high-intensity laser pulses has received great interest lately due to the recent

advances in the field of experimental attosecond physics. This is exemplified by the 2018 nobel prize in physics[48], which partly was attributed the development of ultra-short, high intensity pulses for probing electronic systems. These pulses almost rip apart electronic systems.

One important consequence of strong-field process is high harmonic generation (HHG)[29]. It can be thought of as a process where electrons are ejected from the atom and subsequently pushed back to collide with the remaining electrons, generating high frequency oscillations of high harmonic order of the laser frequency. It gives a very specific signature of the processes involved. Another important side-effect of strong-field laser pulses is the ionization of systems, which is an important topic to study and quantify.

To study these effects in many-body systems, we need accurate real-time ab initio time-dependent methods that recover as much of the correlation effects between electrons as possible. A common, highly accurate method is multi-configurational time-dependent Hartree-Fock method (MCTDHF)[23], which can be considered exact in a given single-particle basis. However, MCTDHF suffers from exponential scaling with system size, and truncated approximations of MCTDHF such as restricted active space self-consistent field (RASSCF)[34] do not provide a size extensive treatment of the problem. Therefore, we consider the coupled-cluster (CC) hierarchy of methods[7]. The time-dependent variants of time-dependent coupled-cluster (TDCC) provide a hiearchy of size-extensive and increasingly accurate methods, at polynomial cost. Our work concerns a special form of TDCC called orbital-adaptive time-dependent coupled-cluster theory (OATDCC), which was developed recently and is an ongoing point of research.

The choice of basis set is imperative in any many-body calculation. A ground state calculation does not get better than the basis describing the single-particle orbitals. Even more importantly, in time-dependent methods the basis set needs to provide the electrons with the flexibility needed by the powerful pertubations of the laser. Our choice of basis is the discrete variable representation (DVR)[30], which provides localized basis functions located at grid points.

## 1.1 Goals

The main goal of this thesis is to study quantum mechanical systems in strong electromagnetic fields, with the doubles approximation of OATDCC (OATDCCD) theory on a DVR basis set. The OATDCC method is in general more stable than regular TDCC for strong electromagnetic fields[38], and allows for orbital rotations of a small computational basis set within a larger underlying basis set.

To use the OATDCCD method for this purpose, some changes must be made to the existing implementation by Schøyen and Winther-Larsen. The rotations of the elements of the small basis set are governed by both the P- and Q-space equations, where the P-space govern internal changes in the basis and the Q-space equations govern rotations into the underlying basis. The Q-space equations is not implemented in the given OATDCCD method. Finally, the DVR

basis provides a symmetry in the matrix elements. To exploit this symmetry it is necessary to adapt the equations of the OATDCCD-method explicitly.

Several subgoals are formulated in order to achieve the main goal;

- Implement the DVR basis set as part of the QuantumSystems library.
- To validate the DVR basis implementation, solve the single-particle problem numerically using the finite-difference method (FDM), for arbitrary potential surfaces in one dimension.
- Implement a Hartree-Fock (HF) solver in order provide a many-body reference state for the later coupled-cluster calculations with static orbitals. The Hartree-Fock method is a standard first step of any many-body calculations, and the methods of coupled-cluster and FCI are called post-Hartree-Fock methods due to this. Furthermore, the HF solver will provide a cheap method for experimenting with techniques, compared with the more expensive method of OATDCC.
- Implement a time-dependent solver for independent particle states, like single-particle and Hartree-Fock states.
- Get familiar with the coupled-cluster theory by implementing simple CCD solver.
- Adapt the CCD solver to allow for spin restriction.
- Implement spin restricted OATDCCD, which will be faster than OATDCCD without spin restriction for systems of interest.
- Implement the Q-space equations for OATDCCD. and find its ground staate Possible methods for ground state computations include the imaginary time relaxation method and adiabatic switching.
- Implement a DVR-adapted solver for OATDCCD with Q-space equations.
- Apply OATDCCD with the DVR basis to the study of strong-field phenomena, such as HHG and ionization.

One-dimensional systems provide a simple model of otherwise computationally expensive problems. Note that one-dimensional systems do not necessarily represent physical systems directly. Even then, in strong-field simulations, one-dimensional systems exhibit the same phenomena of HHG and ionization as higher dimensional systems. In one dimension, we can use a very fine resolution for the DVR grid, providing a cheap way of both qualitatively studying the phenomena of strong fields with great accuracy, and validating the chosen many-body method for this study. Thus, one-dimensional problems are useful, and have been studied extensively using methods such as MCTDHF and RASSCF.

The existing framework allows for higher-dimensional computations. Any further contributions or extensions as a result of this work should adhere to this standard. For example, after implementation, the DVR adapted OATDCCD method should function in higher dimensions as well as one dimension, like the pre-existing OATDCCD method.

## 1.2 Code standards

The programming language chosen for our implementation in this project has been `Python`. It gives an ease of code development compared to other languages, mainly because it is dynamically typed which lets us define abstractions and generalize the code easily. Though it is not the fastest of programming languages, it is popular in many scientific communities due to its well developed libraries for fast numerical calculation. These include `numpy` and `scipy`, which under the hood utilizes the routines of Linear Algebra PACKage (LAPACK)[2] and Basic Linear Algebra Subprograms (BLAS)[1], written in the fast language of Fortran.

The code of developed in this project can be found in four separate locations. Our work builds on the work of the previous master students at the computational science at UiO, Schøyen, Winther-Larsen and Kristiansen [26, 41, 51]. Their code is located in two github repositories, specifically `quantum-systems`[1] and `coupled-cluster`[2]. Of these, the second is a private repository, but access can be granted upon request. Our implementation of the DVR basis set has been implemented in the former, while the latter contains our implementation of the OATDCCD $Q$-space orbital equations in both regular and DVR-adapted versions.

The remaining developed code, which is typically not meant for inclusion in any of the two preexisting libraries, is found in our two repositories `py-master`[3] and `master`[4]. These contain a plethora of smaller scripts, functions and classes. Example simulations and analysis code to reproduce the main figures can be found in jupyter notebooks in the `master`-repository. It also contains scripts to reproduce the main data, which is not included in the repositories due to size constraints but can easily be granted upon request. Access to the code base of this thesis is also granted upon request, but it will be made publicly available for the sake of transparency and reproducibility.

## 1.3 Outline

The rest of the text is organized as follows:

**chapter 2** is a brief introduction of selected topics in quantum mechanics relevant in this work.

**chapter 3** entails the solution of the single-particle problem, and introduces the discrete variable representation.

**chapter 4** asserts the main theories we will use to solve the many body schroedinger equation.

**chapter 5** demonstrates the methods used and their implementations

**chapter 6** features results.

[1] https://www.github.com/schoyen/quantum-systems
[2] https://www.github.com/schoyen/coupled-cluster
[3] https://www.github.com/halvarsu/py-master
[4] https://www.github.com/halvarsu/master

**chapter 7** consists of discussion and future work.

# CHAPTER 2

# Quantum mechanics

## 2.1 Schrödinger equation

The most important equation in quantum mechanics is the Schrödinger equation,

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}\left|\Psi(t)\right\rangle = \hat{H}(t)\left|\Psi(t)\right\rangle, \tag{2.1}$$

where $\hat{H}(t)$ is the Hermitian Hamiltonian operator describing the system and its interactions with the environment, $|\phi\rangle$ a vector in Hilbert space $\mathcal{H}$ representing the state of the system. The Schrödinger equation is a first order differential equation in time, and governs the time evolution of quantum systems.

For a time-independent Hamiltonian $\hat{H}(t) = H$, we can find stationary solutions which do not depend on time. The Schrödinger equation then reduces to an eigenvalue equation to be solved for the energy eigenpair $(E, |\phi\rangle)$,

$$\hat{H}\left|\phi\right\rangle = E\left|\phi\right\rangle. \tag{2.2}$$

The solutions of the Schrödinger equation form an orthonormal basis $\{\phi_n\}_{n=1}^{\infty}$ which we can use to expand any other function,

$$\left|\psi\right\rangle = \sum c_n\left|\phi_n\right\rangle, \qquad \langle\phi_n|\phi_m\rangle = \delta_{nm}. \tag{2.3}$$

We name eqs. (3.1) and (3.2) the time-dependent Schrödinger equation and the time-independent Schrödinger equation, respectively.

Here both the operator $\hat{H}$ and the state $|\Psi(t)\rangle$ are time-dependent, where the former represent changes in the environment (changes in the system are reflected in $\Psi(t)$). In the case of no external influence, the eigensolutions of $\hat{H}(t) = \hat{H}$ are time independent up to a complex phase,

$$i\frac{\mathrm{d}}{\mathrm{d}t}\left|\phi_n(t)\right\rangle = \hat{H}\left|\phi_n(t)\right\rangle = E_n\left|\phi_n(t)\right\rangle \quad \Longrightarrow \quad \left|\phi_n(t)\right\rangle = e^{iE_nt}\left|\phi_n(0)\right\rangle. \tag{2.4}$$

We see that the time-evolution of any other state becomes

$$\left|\Psi(t)\right\rangle = \sum_n c_n(t)\phi_n, \quad \text{where} \quad c_n(t) = e^{iE_nt}c_n(0). \tag{2.5}$$

### 2.1.1 Atomic units

To simplify notation, we use atomic units, defined to make the Schrödinger equation for the Hydrogen atom dimensionless. The original form of said equation is the following,

$$\left(-\frac{\hbar^2}{2m_e}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 r}\right)\phi(\mathbf{r}) = E\phi(\mathbf{r}), \tag{2.6}$$

where $\hbar$ is the reduced Planck constant, $e$ and $m_e$ are the charge and mass of the electron, respectively, and $\epsilon_0$ is the vacuum permittivity, and $r = |\mathbf{r}|$ is the distance to the nucleus. We define the dimensionless position vector $\mathbf{r}'$ such that $\mathbf{r}' = \mathbf{r}/a_0$, where $a_0$ is the Bohr radius,

$$a_0 = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = 5.291772 \times 10^{-12}\,\text{m}. \tag{2.7}$$

A unit of length in atomic units is consequently equal to the Bohr radius. The Laplacian in atomic units becomes $\nabla^{2'} = a_0^2\nabla^2$, and inserting into eq. (2.6), we get

$$\frac{m_e e^4}{(4\pi)^2\epsilon_0^2\hbar^2}\left(-\frac{1}{2}\nabla^{2'} - \frac{1}{r'}\right)\phi'(\mathbf{r}') = E\phi'(\mathbf{r}'). \tag{2.8}$$

Note that the wavefunction is rescaled such that $\phi'(\mathbf{r}') = \phi(\mathbf{r})$. Next we rescale the energy $E' = E/E_a$, such that a unit of atomic energy is given by the Hartree $E_a$. It is defined as

$$E_a = \frac{m_e e^4}{(4\pi)^2\epsilon_0^2\hbar^2} = 4.35974 \times 10^{-18}\,\text{J}. \tag{2.9}$$

The Schrödinger equation in atomic units is finally

$$\left(-\frac{1}{2}\nabla^{2'} - \frac{1}{r'}\right)\phi(\mathbf{r}') = E'\phi(\mathbf{r}'). \tag{2.10}$$

## 2.2 Lasers

We consider interactions in the dipole approximation of the electromagnetic interaction, which is valid when the size of the system is much smaller than the wavelength of the interaction. We write the time-dependent part of the Hamiltonian as[1]

$$\hat{H}_I(t) = -\hat{\mathbf{d}} \cdot \boldsymbol{E}(t). \tag{2.11}$$

Here $\hat{\mathbf{d}}$ is the dipole moment of the system and $E(t)$ is the time-dependent, polarized electric field. We will use

$$\boldsymbol{E}(t) = \boldsymbol{\epsilon} E_0(t)\cos\omega t \tag{2.12}$$

$\boldsymbol{\epsilon}$ is the polarization vector, $E_0(t)$ includes the field strength and a time-dependent envelope, and the cosine terms describes a monochromatic light of frequency $\omega$. We write the dipole moment in the length gauge,

$$\hat{\mathbf{d}} = e\hat{\mathbf{r}}, \tag{2.13}$$

[1] Note that this is the one-body part. We will also consider time dependence in the form of adiabatic switching for the two-body operator elements.

where $\hat{\mathbf{r}}$ is the position operator, and $e$ is the electron charge, equal to unity in atomic units. The systems of study in this thesis project are one dimensional, and as such we use linearly polarized interactions polarized in the same direction as the system. Equation (2.14) then becomes

$$H_I(t) = -\hat{x}E(t)\cos\omega t. \tag{2.14}$$

## 2.3 The Variational Principle

In quantum mechanics, physical observables are represented mathematically as operators on Hilbert space. To calculate the expectation value of an observable for a given state $|\Phi\rangle$, we need to find the expectation value of the corresponding operator. Let the operator in question be $\hat{A}$. The expectation value can be written in terms of the expectation value functional,

$$\mathcal{E}_A\left[|\Psi\rangle\right] = \frac{\langle\Psi|\hat{A}|\Psi\rangle}{\langle\Psi|\Psi\rangle}. \tag{2.15}$$

The brackets imply a functional dependence on the whole form of $|\Psi\rangle$. By performing a variation of the wavefunction $|\Psi\rangle \to |\Psi\rangle + |\delta\rangle$, it can be shown[21] that the first order variations in the functional disappear if and only if the state $|\Psi\rangle$ is an eigenvalue of the operator $A$. If we let the operator be the Hamiltonian of the system, $\hat{A} = \hat{H}$, then finding the extremal points of $\mathcal{E}_H$ thus corresponds to solving the time-independent Schrödinger equation. This is called the variational principle.

### 2.3.1 The Variational Method

The variational method is a practical tool derived from the variational principle. When working with methods for solving the Schrödinger equation for complex systems, such as many-body systems described later in this thesis, it becomes necessary to introduce some approximation of the wavefunction, $|\Psi(T)\rangle$, in terms of a set of parameters $T$. The ground state in terms of this *ansatz* for the true wavefunction will then be an approximation to the true ground state. However, we know from the variational principle that the true ground state is the state that minimizes the energy functional; the energy of the state represented by $|\Psi(T)\rangle$ will as such be greater or equal to the exact ground state energy $E_0$. The expectation value of the energy for the parametrized function is then

$$E_H(T) = \frac{\langle\Psi(T)|\hat{H}|\Psi(T)\rangle}{\langle\Psi(T)|\Psi(T)\rangle} \geq E_0. \tag{2.16}$$

Notice that we here use regular parantheses for the expectation value. This is because the functional dependence in eq. (2.15) has been replaced by a dependence on $T$, which does not have a functional form but is instead a discrete set of values.

The variational method is then the minimization of this energy in terms of the parameters $T$, which gives the ground state of the ansatz. The strength of the variational method is that it ensures that the energy is bounded from below, and as such if changing the ansatz produces a lower energy then it is by definition a better approximation of the ground state.

The proof of eq. (2.16) is the following. Consider that $\hat{H}$ is Hermitian. Assuming $\hat{H}$ is diagonalizable, then we can find an orthonormal basis of eigenfunctions $\{\Phi_k\}$ with real eigenvalues $E_k$ for $k = 0, 1, 2, \ldots$. Any normalizable wavefunction, including $|\Psi(T)\rangle$, can be expanded in this basis,

$$|\Psi\rangle = \sum_k c_k \, |\Phi_k\rangle \, . \tag{2.17}$$

Inserting this into eq. (2.16) we get

$$E_H(T) = \frac{\sum_{kl} E_k c_l^* c_k \, \langle \Phi_l | \Phi_k \rangle}{\sum_{kl} c_l^* c_k \, \langle \Phi_l | \Phi_k \rangle} = \frac{\sum_k E_k |c_k|^2}{\sum_k |c_k|^2} \geq E_0 \frac{\sum_k |c_k|^2}{\sum_k |c_k|^2} = E_0.$$

Here we used the orthonormality of the basis $\langle \Phi_k | \Phi_l \rangle = \delta_{kl}$, and the inequality follows from the fact that the ground state is in fact the minimum of the variational functional in eq. (2.15).

### 2.3.2 The Time-Dependent Variational Principle

Extending the variational principle to the time-domain, we get the time-dependent variational principle. It is given by the principle of least action, which states that the physical time-dependent state is the one that minimizes the action functional,

$$S[\Psi(\cdot)] = \int_0^T \mathrm{d} \, \frac{\langle \Psi(t) | \Big( i\hbar \partial_t - \hat{H}(t) \Big) | \Psi(t) \rangle}{\langle \Psi(t) | \Psi(t) \rangle}. \tag{2.18}$$

The dot-notation indicates that the functional is dependent on the entire shape of $|\Psi(t)\rangle$. The equation of motion for $|\Psi(t)\rangle$ is found by requiring that the functional is stationary, such that $\delta S = 0$. In the general case, this recovers the Schrödinger equation, but for a specific ansatz it can be used to derive the equations of motion for parameters of the ansatz.

### 2.3.3 The Bivariational Principle

In the following, we will be following the theory section of [27]. The variational principle can be generalized to operators that are not necessarily Hermitian, where the right and left eigenvalues are different. Let $A$ be an operator which can work two on separate, independent Hilbert spaces $\mathcal{H}$ and $\mathcal{H}'$. We define the bivariational functional related to $A$ as a functional $\mathcal{E}_A : \mathcal{H} \times \mathcal{H}' \to \mathbb{C}$ which takes as argument two wavefunctions $\langle \Psi' | \in \mathcal{H}'$ and $|\Psi\rangle \in \mathcal{H}$, and produces a complex number,

$$\mathcal{E}_A \big[ \, \langle \Psi' | \, , | \Psi \rangle \, \big] \equiv \frac{\langle \Psi' | A | \Psi \rangle}{\langle \Psi' | \Psi \rangle}. \tag{2.19}$$

The critical points of this functional are given by the condition that the functional is stationary under variations in the two wavefunctions, or $\delta \mathcal{E}_A = 0$. This corresponds to the wavefunctions $|\Psi\rangle$ and $\langle \Psi' |$ being simultaneous right and left eigenvectors, respectively, of $A$ with the same eigenvalue,

$$(A - a) \, |\Psi\rangle = 0, \qquad\qquad \langle \Psi' | \, (A - a) = 0, \tag{2.20}$$

where $a$ is the corresponding eigenvalue and the value of the functional at this point,

$$a = \mathcal{E}_A\big[\,\langle\Psi'|\,,|\Psi\rangle\,\big]\big|_{\delta\mathcal{E}_A=0}\,. \tag{2.21}$$

To demonstrate this we perform the variations in $|\Psi\rangle$ and $\langle\Psi'|$ individually. We will only do the variation in $|\Psi\rangle$, as the one in $\langle\Psi'|$ is completely analogous:

$$\begin{aligned} 0 = \frac{\delta\mathcal{E}_A}{\delta\,|\Psi\rangle}\big[\,\langle\Psi'|\,,|\Psi\rangle\,\big] &= \frac{\delta}{\delta\,|\Psi\rangle}\frac{\langle\Psi'|A|\Psi\rangle}{\langle\Psi'|\Psi\rangle} \\ &= \frac{A\,|\Psi\rangle\,\langle\Psi'|\Psi\rangle - \langle\Psi'|A|\Psi\rangle\,|\Psi\rangle}{\langle\Psi'|\Psi\rangle^2} \\ &= \frac{A\,|\Psi\rangle}{\langle\Psi'|\Psi\rangle} - \frac{\mathcal{E}_A\,|\Psi\rangle}{\langle\Psi'|\Psi\rangle} \\ \implies A\,|\Psi\rangle &= \mathcal{E}_A\,|\Psi\rangle = a\,|\Psi\rangle\,, \end{aligned} \tag{2.22}$$

where $a \in \mathbb{C}$.

Computing the eigenpair of $A$ by solving $\delta\mathcal{E}_A = 0$ is called the bivariational principle, and is in our cases most relevant when the operator is the Hamiltonian, giving the bivariational energy functional

$$\mathcal{E}_H\big[\,\langle\Psi'|\,,|\Psi\rangle\,\big] \equiv \frac{\langle\Psi'|\hat{H}|\Psi\rangle}{\langle\Psi'|\Psi\rangle}. \tag{2.23}$$

Proceeding to the time-dependent case, we express the bivariational generalization of the action functional as

$$\mathcal{S}\big[\,\langle\Psi'|\,,|\Psi\rangle\,\big] \equiv \int_0^T \frac{\langle\Psi'(t)|i\frac{\partial}{\partial t} - \hat{H}|\Psi(t)\rangle}{\langle\Psi'(t)|\Psi(t)\rangle}. \tag{2.24}$$

Demanding stationarity for arbitrary variations in the wave functions with the endpoints fixed, we can extract the time-dependent Schröedinger equation and its complex conjugate,

$$i\frac{\partial}{\partial t}\,|\Psi(t)\rangle = \hat{H}\,|\Psi(t)\rangle\,, \quad \text{and} \quad -i\frac{\partial}{\partial t}\,\langle\Psi'(t)| = \langle\Psi'(t)|\,\hat{H}. \tag{2.25}$$

### 2.3.4 The Magnus Expansion

Another description of the time-evolution of a system is using the time-propagator $U(t,t_0)$, a unitary operator which propagates a state from time $t_0$ to a time $t$,

$$|\Psi(t)\rangle = U(t,t_0)\,|\Psi(t_0)\rangle\,. \tag{2.26}$$

Inserting into eq. (3.2), we see that $U(t,t_0)$ obeys the Schrödinger equation,

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}U(t,t_0) = \hat{H}(t)U(t,t_0). \tag{2.27}$$

Now we assume that $U(t,t_0)$ can be represented as an exponential,

$$U(t,t_0) = \exp\{\Omega(t,t_0)\}. \tag{2.28}$$

If the commutator of the Hamiltonian satisfies $\left[\hat{H}(t_1), \hat{H}(t_2)\right] = 0$ for all times $t_1$ and $t_2$, which is the case for the time-independent hamiltonian $\hat{H} = \hat{H}(t)$, then eq. (2.27) can be solved like a normal ordinary differential equation, with

$$U(t, t_0) = e^{\int_{t_0}^{t} \hat{H}(t)\mathrm{d}t/i\hbar} U(t_0, t_0) = e^{\int_{t_0}^{t} \hat{H}(t)\mathrm{d}t/i\hbar}, \tag{2.29}$$

where we used that $U(t_0, t_0) = \mathcal{I}$. However, this is normally not the case for Hamiltonians of interest. For the general case, Magnus [32] proved that $\Omega(t, t_0)$ satisfies

$$\frac{\partial}{\partial t}\Omega(t, t_0) = \frac{1}{i\hbar}\hat{H}(t) + \frac{1}{i\hbar}\sum_{k=1}^{\infty}(-1)^k \frac{B_k}{k!}\Big[\Omega(t, t_0), \Big[\cdots, \Big[\Omega(t, t_0), \hat{H}(t)\Big]\Big]\Big], \tag{2.30}$$

where $B_k(t)$ are the Bernoulli numbers

$$B_k(t) = 1, \frac{1}{2}, \frac{1}{6}, 0, -\frac{1}{30}, \ldots, \quad \text{for} \quad k = 0, 1, 2, 3, 4, \ldots. \tag{2.31}$$

Integrating eq. (2.30) we get

$$\Omega(t, t_0) = \frac{1}{i\hbar}\int_{t_0}^{t} \Big\{\hat{H}(t_1) + \Big[\Omega(t, t_0), \Big[\cdots, \Big[\Omega(t, t_0), \hat{H}(t)\Big]\Big]\Big]\Big\}\,\mathrm{d}t_1, \tag{2.32}$$

which can be iteratively inserted into itself to yield a pertubation series called the Magnus expansion. The first order approximation includes only one power of $\Omega(t, t_0)$,

$$\Omega(t_0, t) = \frac{1}{i\hbar}\int_{t_0}^{t} \mathrm{d}t_1 \hat{H}(t_1). \tag{2.33}$$

Including more orders of $\Omega(t, t_0)$ (or rather, more iterations of inserting $\Omega(t, t_0)$ on the right hand side) and ordering in terms of the number of Hamiltonian operators, we get better approximations. Up to third order, the expansion is

$$\begin{aligned}\Omega(t_0, t) =& \frac{1}{i\hbar}\int_{t_0}^{t} \mathrm{d}t_1 \hat{H}_1 \\ &- \frac{1}{2\hbar^2}\int_{t_0}^{t} \mathrm{d}t_1 \int_{t_0}^{t_1} \mathrm{d}t_2 \Big[\hat{H}_1, \hat{H}_2\Big] \\ &- \frac{1}{i6\hbar^3}\int_{t_0}^{t} \mathrm{d}t_1 \int_{t_0}^{t_1} \mathrm{d}t_2 \int_{t_0}^{t_2} \mathrm{d}t_3 \Big(\Big[\hat{H}_1, \Big[\hat{H}_2, \hat{H}_3\Big]\Big] + \Big[\hat{H}_3, \Big[\hat{H}_2, \hat{H}_1\Big]\Big]\Big) + \ldots,\end{aligned} \tag{2.34}$$

where we used the abbreviated $\hat{H}_i \equiv \hat{H}(t_i)$.

The key point for all the integrators resulting from the Magnus expansion is that they are *symplectic*, meaning they conserve the phase space relation of the wavefunction, where the real and imaginary parts of the quantum wavefunction serve as generalized coordinates and conjugate momenta[38].

## 2.4 Ground state calculations

Finding the ground state of many-body quantum systems is an important but challenging task. We use it as the initial state of our dynamics simulations, because chemical systems tend to be in the ground state before an interaction takes place. The ground state of a system is stationary, and as such we don't need to worry about any dependence on the initial time of a simulation.

Methods such as Hartree-Fock and coupled-cluster have a set of equations that can be solved to give the parameters of the ground state. However, the orbital-adapative coupled-cluster method, which one of the main topics of this thesis, does not have a such a ground state solver. Under the condition of a zero Q-space, a concept to be defined later, the ground state of the OATDCC method equals the one of the non-orthogonal orbital optimized coupled-cluster method (NOCC), and the NOCCD root solver can be used to find the ground state[37].

In this section we present two different methods for the ground state of many-body theories. These are imaginary time evolution and adiabatic switching.

### 2.4.1 Imaginary time

We now show that evolving a quantum mechanical system in imaginary time is one way of finding the ground state of a given Hamiltonian. To show this, we expand an the initial wave function $|\Psi(0)\rangle$ in terms of solutions to the stationary Hamiltonian, $|\Phi_i\rangle$,

$$|\Psi(0)\rangle = \sum_i c_i |\Phi_i\rangle . \tag{2.35}$$

In the case of a time-independent Hamiltonian, the time-dependent Schrödinger equation then has solutions

$$|\Psi(t)\rangle = \exp\Big\{-i\hat{H}t\Big\} \sum_i c_i |\Phi_i\rangle = \sum_i c_i \exp\{-iE_i t\} |\Phi_i\rangle . \tag{2.36}$$

Letting $\tau = it$, we we see that the components of the eigenfunctions decrease in norm according to the exponential of the energy

$$|\Psi(\tau)\rangle = \sum_i c_i \exp\{-E_i \tau\} |\Phi_i\rangle . \tag{2.37}$$

However, states with a higher energy dissipate more quickly, and as such the ground state will quickly dominate,

$$|\Psi(\tau)\rangle =_{\tau\to\infty} c_0 \exp\{-E_0 \tau\} |\Phi_0\rangle . \tag{2.38}$$

This is based on the assumptions that the difference in energy between the lowest lying state is not too small, and that the initial state $|\Psi(0)\rangle$ has a non-zero overlap with the ground state.

### 2.4.2 Adiabatic theorem

Suppose that $\hat{H}(t)$ describes a time-dependent Hamiltonian, with eigenpairs $\{E(t), \psi(t)\}$ at each time $t$. The time-independent Schrödinger equation is at each time-step then fulfilled as

$$\hat{H}(t)\psi(t) = E(t)\psi(t), \tag{2.39}$$

The adiabatic theorem tells us that as long as the change in $\hat{H}(t)$ is slow enough, then a system prepared in an eigenstate of $\hat{H}(t_1)$ at one time $t_1$ will be in the eigenstate of $\hat{H}(t)$ at all times $t$[5]. In general, a state initially prepared as

$$|\Psi(0)\rangle = \sum_n c_n |\psi_n(0)\rangle \tag{2.40}$$

will at a any later time be written as

$$|\Psi(t)\rangle = \sum_n c_n e^{i\theta_n(t)} e^{i\gamma_n(t)} |\psi_n(t)\rangle , \tag{2.41}$$

where

$$\theta_n(t) \equiv -1/\hbar \int_0^t \mathrm{d}t' E_n(t'), \tag{2.42}$$

is the dynamic phase, and

$$\gamma_n(t) \equiv i \int_0^t \mathrm{d}t' \, \langle \psi_n(t) | \partial_t \psi_n(t) \rangle , \tag{2.43}$$

is the geometric phase.

What does slow enough mean? We can recognize two characteristic times in the system, the internal time $T_i$ which characterizes internal oscillations and changes in the wavefunction, and the external time $T_e$ characterizing the changes of the external system or Hamiltonian itself. Slow enough then translates into $T_e \gg T_i$.

### 2.4.3 Adiabatic switching

Adiabatic switching is, like imaginary time propagation, a time evolution process. It is based on the adiabatic theorem, where initially the system is described by the non-interacting parts of the Hamiltonian, and the initial state is the ground state of the non-interacting Hamiltonian. The system is then evolved in time while the many-body interactions are slowly turned on. We can rewrite the many-body hamiltonian as

$$\hat{H}(t) = \hat{H}_0 + F(t)\hat{V}, \tag{2.44}$$

where $\hat{H}_0$ is the non-interacting hamiltonian , $F(t)$ is an adiabatic switching function which determines the speed and form of the process, and $\hat{V}$ is the two-body operator we wish to introduce over time.

If change in the Hamiltonian over time is slow enough, then the adiabatic theorem tells us that the system will stay in the ground state of $H(t)$ at each

time-step througout the process. If $F(t)$ then also converges towards 1, then the many-body interactions will be at full strength we will have recovered the many-body Hamiltonian as well as the many-body ground state. In theory, slow enough means infinitely slow. However, we show that the error of the process decreases exponentially with the length of the process, as long as we also introduce some favourable switching function.

We will construct the adiabatic switching function $F(t)$ such that it gives a smooth and tunable transition for a time evolution $t \in [0, T]$. We also want $F(0) \approx 0$ and $F(T) \approx 1$, such that the system starts and ends as close as possible to the non-interacting and the full many-body system, repsectively. In addition to this, we believe it to be desirable for the derivatives to start and end as close to 0 as possible, as we show that the linear switching function performs poorly no matter how slow the process is. However, as it is impossible to construct a function from 0 to 1 where all the derivatives start and end at zero, we have to make do with an exponential convergence towards the given start and end values. One example of a such a function is the Fermi function[22], given by

$$F(t) = 1 - \frac{1}{1 + \exp\big((t - T_{1/2})/\tau\big)}, \tag{2.45}$$

with a half-time of $T_{1/2}$ and a decay time of $\tau$. If $\tau$ is small enough, then the Fermi function gives the desired properties stated above.

# CHAPTER 3

# The Discrete Variable Representation

All methods for solving the many-body Schrödinger equation described later in this thesis are formulated with reference to some single particle basis $\{\phi_p\}_{p=1}^{L}$, and is based on the matrix elements of operators for orbitals in this basis. Solutions for the eigenvalue problem of the single-particle Hamiltonian $\hat{h}$ are used as the constituents of the reference Slater determinant of the many-body problem, either directly or by first rotating into a better suited basis, for example by performing a HF computation first. The quality of the many-body wavefunction is thus limited by the quality of the single-particle orbitals.

Many basis sets have been designed for use in molecules and atoms. The key quality of a basis set is to provide a good and efficient representation of the matrix elements of the Hamiltonian. For ground state calculations, one usually seeks orbitals that provide a good description of the domain of the ground state wavefunction, while calculations of excited states usually require some more diffuse orbitals to allow electrons to move out of the core region. Our goal is to be able to simulate strong-field processes where ionization needs to be handled, which means that we need to cover an area large enough for electrons to be in some sense disconnected from the remaining wavefunction.

For this purpose, we study the discrete variable representation (DVR). The DVR is a grid representation based on Gaussian Quadrature. The main advantage of the DVR in many-body calculations is that it provides sparse two-body matrix elements. This lets us utilize many DVR functions in a given calculation, and thus represent a large spatial extent.

In this chapter, we start with a presentation of regular grid methods, specifically the finite difference method (FDM). We then move on to the presentation of DVR, starting with the quadrature methods which gives the DVR its accuracy, and continuing with the details of DVR, with Gaussian DVR as an example. Finally, we briefly present the matrix elements of sinc-DVR, which is the basis we use for our many-body calculations.

## 3.1 One-body ground state

We consider the single-particle, time-independent Schrödinger equation in one dimension,

$$E\left|\psi\right\rangle = \hat{h}\left|\psi\right\rangle = (\hat{T}+\hat{V})\left|\psi\right\rangle, \tag{3.1}$$

where $\hat{T} = -\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}x^2}$ is the kinetic energy operator and $\hat{V}$ is the potential energy operator. The more general time-dependent version is

$$i\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = \hat{h}(t)\left|\psi(t)\right\rangle, \tag{3.2}$$

where both the Hamiltonian $h(t)$ and the state $|\psi(t)\rangle$ can be time-dependent.

The standard approach in numerical quantum mechanics is to expand the true single particle wave function $|\Psi\rangle$ in terms of a truncated basis $\{|\phi_n\rangle\}_{n=0}^{N-1}$ of size $N$. In this approximative representation, the one-body Hamiltonian $\hat{h}$ is projected from the original infinite-dimensional Hilbert space into the $N$-dimensional subspace spanned by the basis. We can write this in terms of a projection operator $P = \sum_{n=0}^{N-1}|\phi_n\rangle\langle\phi_n|$,

$$\hat{h} = \hat{T}+\hat{V} \to \hat{h}' = P\hat{h}P = \sum_{m,n=0}^{N-1}(T_{mn}+V_{mn})\left|\phi_m\right\rangle\left\langle\phi_n\right|. \tag{3.3}$$

From now on, we will use $\hat{h}$ for the projected one-body Hamilton operator. Such a representation is called a finite basis representation (FBR), which is a spectral basis. Our goal is to find ways to easily represent the kinetic and potential matrix elements, respectively $T_{mn} = \langle\phi_m|\hat{T}|\phi_n\rangle$ and $V_{mn} = \langle\phi_m|\hat{V}|\phi_n\rangle$. By easily, we mean either in terms of computational complexity or in terms of storage requirements. For single particle basis sets with applications in many-body problems, we are also interested in the form of the Coulomb operator, $u_{rs}^{pq} = \langle\phi_p\phi_q|\hat{u}|\phi_r\phi_s\rangle$.

We now present the methods to solve eq. (3.1), and later describe a prescription for eq. (3.2).

### 3.1.1 Particle on a grid

Equation (3.1) is an eigenvalue equation, which can be written

$$\hat{h}\left|\psi_\lambda\right\rangle = E_\lambda\left|\psi_\lambda\right\rangle. \tag{3.4}$$

Here $(|\psi_\lambda\rangle, E_\lambda)$ are eigenpairs for a given confining potential $V(x)$. We can convert this to a position basis representation by inserting a complete position basis $\int \mathrm{d}x\,|x\rangle\langle x|$ and left multiplying by a basis element $\langle x'|$,

$$\int \mathrm{d}x h_{x',x}\psi_\lambda(x) = E_\lambda\int \mathrm{d}x\psi_\lambda(x)\delta(x'-x) = E_\lambda\psi_\lambda(x'), \tag{3.5}$$

where $h_{x',x} = \langle x'|h|x\rangle$, $\psi_\lambda(x) = \langle x|\psi_\lambda\rangle$, and the Dirac delta function $\delta(x'-x)$ comes from the normalization of the basis functions $\langle x'|x\rangle = \delta(x'-x)$.

The simplest way to solve this is to use a discrete representation on a uniformly spaced grid of $N$ points, over a certain domain $x \in [a,b]$. The discretized

grid points are $x_n = a + n\Delta x$ for $n = 0, \ldots, N-1$, with a grid spacing of $\Delta x = (b-a)/(N-1)$. The spatial wave function $\psi(x)$ is also represented in the discretized grid basis,

$$\psi(x_n) \equiv \langle x_n|\psi\rangle\,. \tag{3.6}$$

When discretizing, normalization of the grid basis becomes $\langle x_n|x_m\rangle = \delta_{nm}$, and the inner product is changed from an integral to a sum over the given basis, $\int \mathrm{d}x \to \Delta x \sum_n$.

The action of the Hamiltonian on the wavefunction is also discretized,

$$\hat{h}\psi(x) = \left(-\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \hat{V}(x)\right)\psi(x) \tag{3.7}$$

$$\to \hat{h}\psi(x_n) = -\frac{1}{2}\frac{\psi(x_{n-1}) - 2\psi(x_n) + \psi(x_{n+1}))}{\Delta x^2} + V(x_n)\psi(x_n), \tag{3.8}$$

where we used a finite difference scheme for the second derivative operator. This leads gives us the finite difference method. The matrix elements of the Hamilton operator are

$$h_{nm} \equiv \langle x_n|h|x_m\rangle = -\frac{1}{2}\frac{\delta_{n,m-1} - 2\delta_{n,m} + \delta_{n,m+1}}{\Delta x^2} + V(x_n)\delta_{nm}. \tag{3.9}$$

Inserting these definitions into eq. (3.5), we get

$$\sum_n h_{nm}\psi_\lambda(x_m) = E_\lambda\psi_\lambda(x_n), \tag{3.10}$$

which is a simple matrix eigenvalue equation to be solved for the vectors $\psi_\lambda(x_n)$.

### Note on grid spacing

The quality of the representation of the wavefunction with respect to the spacing of the grid points $\Delta x$ is given by the momentum of the wave function. Consider that momentum is the Fourier transform of the position basis, i.e. the spatial frequencies. The Nyquist-Shannon sampling theorem[43] tells us that the maximal frequency of a discretized Fourier transform is given by half the sampling frequency. Translated into our problem, the maximal momentum we can represent is then half that of the reciprocal grid spacing (in atomic units).

### Matrix elements

To use the single-particle basis in further calculations, we are interested in computing matrix elements. The first is the overlap matrix, which for a general basis is

$$S_{pq} = \langle\phi_p|\phi_q\rangle = \int \mathrm{d}x\,\phi_p^*(x_1)\phi_q(x_1). \tag{3.11}$$

For time-dependent problems we will be using the length gauge for electromagnetic interactions. This introduces a one-body operator given by

$$\hat{h}_I(t) = F(t)\hat{x}. \tag{3.12}$$

Here $F(t)$ is a regular function which can be applied at each time-step, and as such it is sufficient to store the matrix elements of the position operator,

$$X_{pq} = \langle \phi_p | \hat{x} | \phi_q \rangle = \int \mathrm{d}x \, \psi_p^*(x_1) x \phi_q(x_1). \tag{3.13}$$

Next, for a many-body calculation, we are interested in the two-body matrix elements

$$\langle \psi_p \psi_q | \hat{u}(x_1, x_2) | \psi_r \psi_s \rangle = \int \int \mathrm{d}x_1 \mathrm{d}x_2 \, \psi_p^*(x_1) \psi_q^*(x_2) u(x_1, x_2) \phi_r(x_1) \phi_s(x_2), \tag{3.14}$$

where $\hat{u}(x_1, x_2)$ is some two-body operator. For electronic systems in sufficiently many dimensions, this is the Coulomb operator, which in atomic units is

$$\hat{u}(x_1, x_2) = \frac{1}{|x_1 - x_2|}. \tag{3.15}$$

Using a grid representation for the Coulomb potential can be challenging due to the singularity when $x_1 = x_2$. In three dimensions, this problem can be avoided by considering that the Coulomb operator is the Greens function of the Laplace operator[24]. This results in the expression

$$u_{\gamma\delta}^{\alpha\beta} = 2\pi \delta_{\alpha\gamma} \delta_{\beta\delta} \frac{T_{\alpha\beta}^{-1}}{(\Delta x)^3}, \tag{3.16}$$

where $T_{\alpha\gamma}^{-1}$ is the matrix elements of the inverse of the kinetic energy operator in the infinite size grid limit. However, in one dimension the integrals over the Coulomb potential diverge. Consequently, in the one-dimensional case we circumvent both of these problems by using the shielded Coulomb potential,

$$\hat{u}'(x_1, x_2) = \frac{1}{\sqrt{(x_1 - x_2)^2 + a^2}}, \tag{3.17}$$

where $a$ is a shielding parameter that removes the singularity. The amtrix elements can now be safely evaluated at the grid points.

## 3.2 DVR

A useful way of representing the basis of single particle Hamiltonian is the discrete variable representation (DVR). A definition of DVR is given by Groenenboom and Colbert [17]. They define a DVR as consisting of three ingredients, first an orthonormal single particle basis set,

$$\{\theta_\alpha(x), \quad \alpha = 1, \ldots, N_b\}, \qquad \langle \theta_\alpha | \theta_\beta \rangle = \delta_{\alpha\beta}. \tag{3.18}$$

This is connected in a one-to-one correspondence to a $N_b$-point quadrature with a set of grid points and weights,

$$\{(x_\beta, \omega_\beta), \quad \beta = 1, \ldots, N_b\}, \tag{3.19}$$

Finally, the basis functions needs to have the property that when evaluated at the grid points of the quadrature, only the corresponding grid point gives a non-zero value,

$$\theta_\alpha(x_\beta) = \omega_\beta^{-1/2}\delta_{\alpha\beta}. \tag{3.20}$$

The DVR is always connected to a FBR by a unitary transformation, meaning that the quality of the DVR is given by the quality of the corresponding FBR. There are several ways of constructing a DVR, but it is always necessary to know the value of the kinetic energy operator for the system in question, either for the FBR or the DVR.

The diagonality of the position operator in the DVR basis allows us to easily evaluate and store the matrix elements of both the potential energy operator $\hat{V}(x)$ and, more importantly, the Coulomb operator $\hat{u}(x_1, x_2)$. In a DVR basis, operators given as functions of the coordinates are evaluated at the grid points of the quadrature. The matrix elements of the potential matrix operator in the DVR basis are

$$V_{\alpha\beta} = \langle\theta_\alpha|\hat{V}(x)|\theta_\beta\rangle \approx V(x_\alpha)\delta_{\alpha\beta}, \tag{3.21}$$

and similarly the Coulomb matrix elements are

$$u_{\gamma\delta}^{\alpha\beta} = \langle\theta_\alpha\theta_\beta|\hat{u}(x_1, x_2)|\theta_\gamma\theta_\delta\rangle \approx \delta_{\alpha\gamma}\delta_{\beta\delta}u(x_\alpha, x_\beta). \tag{3.22}$$

This diagonality greatly reduces the memory cost of storage, but also the cost of any calculation where $u_{\gamma\delta}^{\alpha\beta}$ is involved. As an example, consider the transformation $u_{\gamma\delta}^{\alpha\beta} \to u_{rs}^{pq}$ given by

$$|\phi_p\rangle = C_p^\alpha\,|\theta_\alpha\rangle\,, \quad \left\langle\tilde{\phi}_p\right| = \tilde{C}_\alpha^p\left\langle\tilde{\theta}^\alpha\right|. \tag{3.23}$$

The transformation is explicitly

$$u_{rs}^{pq} = \tilde{C}_\alpha^p\tilde{C}_\beta^q C_r^\gamma C_s^\delta u_{\gamma\delta}^{\alpha\beta}. \tag{3.24}$$

If we let the size of the two basis sets be $L$ and $K$ for $|\phi_p\rangle$ and $|\theta_\alpha\rangle$, respectively, and assume $L > K$, then evaluating the expression using intermediates normally requires $\mathcal{O}(KL^4)$ operations. However, with diagonal Coulomb matrix elements the expression becomes

$$u_{rs}^{pq} = \sum_{\alpha\beta}\tilde{C}_\alpha^p\tilde{C}_\beta^q C_r^\alpha C_s^\beta u_{\alpha\beta}^{\alpha\beta}. \tag{3.25}$$

The present expression requires just $\mathcal{O}(LK^4 + K^2L^2)$ operations to evaluate, which is a lot smaller than $\mathcal{O}(KL^4)$, especially when $L/K$ becomes big. Consequently, we can expand a smaller computational basis of size $K$ (such as the Hartree Fock basis or the OA basis associated with OATDCC) in a much larger DVR grid of size $L$, and get a highly flexible basis at a low cost. This is a large advantage when doing time-evolution in strong electromagnetic fields, as the time-dependent wavefunction might be far removed from any initial ground state basis.

The following section contains a presentation of the underlying theory of DVR-basis sets, starting with the quadrature rule which follows a DVR, and continuing with the natural extension into Gauss-DVR based on orthogonal polynomials before venturing into the theory of sinc-DVR basis functions, which are the basis functions we have utilized.

### 3.2.1 Quadrature

According to the second ingredient of a DVR method in eq. (3.19), there is a quadrature underlying any DVR basis, and in the following we briefly clarify what this entails. The name "quadrature" stems from the antique method of calculating the area of a shape by the geometrical construction of a square with equal area, but has become more or less synonymous with numerical integration in the last century. An $N$-point quadrature rule approximates a definite integral of a function $f(x)$, over a domain $\Omega$, as a weighted sum,

$$\int_\Omega \mathrm{d}x\, \omega(x) f(x) \approx \sum_{\alpha=1}^{N} \omega_\alpha f(x_\alpha). \tag{3.26}$$

Here $\omega(x)$ is the weight function specific to the quadrature used, and $x_\alpha$ and $\omega_\alpha$ are the grid points and their associated weights.

For a domain $\Omega = [a, b]$ and weight function $\omega(x) = 1$, a typical example is the midpoint rule, where the single grid point is just the middle of the interval $x_0 = (a + b)/2$, and the weight is the width of the interval $\omega_0 = b - a$,

$$\int_a^b \mathrm{d}x\, f(x) \approx (b - a) f\left(\frac{a + b}{2}\right). \tag{3.27}$$

This is perhaps the simplest quadrature possible; the area of the function is approximated as the rectangle with width equal to the integral and height equal to the function value in the middle of the interval. A better approximation can be achieved by taking the composite midpoint rule, where one splits the integral into $N$ pieces of width $h = (b - a)/N$ and applies the midpoint rule to each of them. The grid points are then $x_\alpha = a + \alpha h + 1/2$ and the grid weights are $\omega_\alpha = h$ for $\alpha = 0, \ldots, N - 1$.

### 3.2.2 Gaussian Quadrature

Another important approach is the $N$-point Gaussian quadrature. In Gaussian quadrature, the grid points $x_\alpha$ and grid weights $\omega_\alpha$ of the quadrature are defined by a set of orthogonal polynomials $C_n(x)$,

$$\int_\Omega \mathrm{d}x\, \omega(x) C_n(x) C_l(x) = \delta_{nl}, \tag{3.28}$$

where $\omega(x)$ is a weight function. The polynomials satisfy a recursion relationship,

$$x C_n(x) = a_n C_{n+1}(x) + b_n C_n(x) + c_n p_{n-1}(x). \tag{3.29}$$

We use the normalized polynomials above, but an equivalent definition is the monic polynomials, where $a_n = 1$ for all $n \geq 0$. Examples of common orthogonal polynomials for different domains can be found in table 3.1, and the grid points and weights can be found in tabulated form. In **??** we show that a Gauss quadrature of degree $N$ approximates a function with a polynomial of degree $2N - 1$,

$$\int_\Omega \mathrm{d}x\, \omega(x) f(x) \approx \int_\Omega \mathrm{d}x\, P_{2N-1}(x) = \sum_\alpha \omega_\alpha f(x_\alpha). \tag{3.30}$$

The grid points are here given by the roots $x_\alpha$ of $C_N(x)$, while the weights are given by

$$\omega_\alpha = \frac{(C^{-1})_{\alpha 0}}{k_0}, \tag{3.31}$$

where $(C^{-1})_{\alpha n}$ is the inverse of the matrix $C_{n\alpha} \equiv C_n(x_\alpha)$, and $k_0 \equiv C_0(x)$, as $C_0(x)$ is a constant. The weights and grid points can be constructed efficiently using the Golub-Welsh algorithm[15].

Table 3.1: Examples of orthogonal polynomials. Definitions of the different polynomials are not subject to this thesis.

| Polynomial class of $C$ | Domain $\Omega$ | Weight $\omega(x)$ |
|---|---|---|
| Legendre | $[-1, 1]$ | $1$ |
| Jacobi | $(-1, 1)$ | $(1-x)^\alpha(1+x)^\beta, \quad \alpha, \beta > -1$ |
| Chebyshev | $(-1, 1)$ | $1/\sqrt{1-x^2}$ |
| Laguerre | $[0, \infty)$ | $e^{-x}$ |
| Hermite | $(-\infty, \infty)$ | $e^{-x^2}$ |

### 3.2.3 Gauss Quadrature DVR

Now that we have defined a quadrature, and shown that the usage of orthogonal polynomials can provide an accurate value for the integral of a function, we are ready to bridge the gap to DVR. The DVR is the representation where the position-operator is diagonal, with diagonal elements corresponding to grid points of the quadrature.

We start by defining the so-called finite basis representation (FBR) [30], by using weighted versions of the polynomials $C_n(x)$ as basis functions for a Hilbert space on the corresponding domain $\Omega$. The FBR basis functions are expressed in terms of the $N$ first polynomials of the given class as

$$\phi_n(x) = \sqrt{\omega(x)} C_n(x), \quad n = 0, \ldots, N-1, \tag{3.32}$$

where the weight ensures orthonormality,

$$\langle \phi_n | \phi_l \rangle = \int \mathrm{d}x\, \phi_n(x)\phi_l(x) = \int \mathrm{d}x\, \omega(x) C_n(x) C_l(x) = \delta_{nl}. \tag{3.33}$$

We now consider the matrix elements of the position operator $\hat{x}$. In the DVR basis, they are given as

$$X_{nl} = \langle \phi_n | x | \phi_l \rangle = \int_a^b \mathrm{d}x\, \phi_n(x)^* x \phi_l(x). \tag{3.34}$$

We can rewrite the integrand as

$$x\, \phi_n^*(x)\phi_l(x) = x\, \omega(x) C_n(x) C_l(x) = \omega(x) P^{n+l+1}. \tag{3.35}$$

Here $P^{n+1+l}$ is a polynomial of degree $n + l + 1 \leq 2N - 1$, and as such the Gaussian quadrature relation for the coordinate matrix in the FBR basis holds

exactly. Using the quadrature rule, we get a closed exact expression for the matrix elements of the Gauss quadrature FBR basis,

$$X_{nl} = \sum_{\alpha=1}^{N} \frac{\omega_\alpha}{\omega(x_\alpha)} \phi_n^*(x_\alpha) x_\alpha \phi_l(x_\alpha). \tag{3.36}$$

The DVR basis is defined as the one where $X_{nl}$ is diagonal, found by diagonalizing with a unitary transformation $T$,

$$X = T^\dagger X^{\mathrm{DVR}} T. \tag{3.37}$$

The exact analytical expression for $T$ is found by rewriting eq. (3.36),

$$\begin{aligned} X_{nl} &= \sum_{\alpha,\beta=1}^{N} \sqrt{\frac{\omega_\alpha}{\omega(x_\alpha)}} \phi_n^*(x_\alpha) x_\alpha \delta_{\alpha\beta} \sqrt{\frac{\omega_\beta}{\omega(x_\beta)}} \phi_l(x_\beta) \\ &\equiv \sum_{\alpha,\beta=1}^{N} (T^\dagger)_{n\alpha} X_{\alpha\beta}^{\mathrm{DVR}} T_{\beta l}, \end{aligned} \tag{3.38}$$

where we defined the diagonal position operator in the DVR basis $X_{\alpha\beta}^{DVR} = x_\alpha \delta_{\alpha\beta}$, and we identified the square transformation matrix,

$$T_{\alpha l} = \sqrt{\frac{\omega_\alpha}{\omega(x_\alpha)}} \phi_l(x_\alpha), \quad T \in \mathbb{R}^{N\times N}. \tag{3.39}$$

Note that $T$ is unitary due to the orthonormality of the basis functions,

$$\delta_{nl} = \langle \phi_n | \phi_l \rangle = \int_a^b \mathrm{d}x\, \phi_n(x)^* \phi_l(x) = \sum_{\alpha=1}^{N} \frac{\omega_\alpha}{\omega(x_\alpha)} \phi_n^*(x_\alpha) \phi_l(x_\alpha) = \sum_{\alpha=1}^{N} (T^\dagger)_{n\alpha} T_{\alpha l}. \tag{3.40}$$

### DVR Basis functions

We can define a DVR wave function localized around the grid point $x_\alpha$ as

$$\theta_\alpha(x) = \sum_n T_{\alpha n}^* \phi_n(x), \tag{3.41}$$

with the inverse transformation

$$\phi_m(x) = \sum_\alpha T_{\alpha m} \theta_\alpha(x). \tag{3.42}$$

Matrix elements belonging to the DVR basis will be exclusively represented by greek letters. The DVR wavefunctions $\theta_\alpha(x)$ are nonzero at their corresponding grid point $x_\alpha$ and zero at all other grid points $x_\beta\, (\beta \neq \alpha)$, as we see by evaluating,

$$\theta_\alpha(x_\beta) = \sum_n T_{\alpha n}^* \phi_n(x_\beta) = \sum_n T_{\alpha n}^* T_{\beta n} \sqrt{\frac{\omega(x_\beta)}{\omega_\beta}} = \delta_{\alpha\beta} \sqrt{\frac{\omega(x_\beta)}{\omega_\beta}}, \tag{3.43}$$

Where we inserted for $\phi_n(x_\beta) = T_{\beta n} \sqrt{\omega(x_\beta)/\omega_\beta}$ from eq. (3.39).

### Potential matrix elements

A strength of the DVR basis is that the matrix elements of the potential energy operator, or any local operator, can be approximated as diagonal. For a potential energy operator $\hat{V}(x)$, we get

$$\langle\theta_\alpha|\hat{V}(x)|\theta_\beta\rangle = \int \mathrm{d}x\, \theta_\alpha(x)^* V(x)\theta_\beta(x) \tag{3.44}$$

$$= \sum_{nm} T^*_{\alpha n} T_{\beta m} \int \mathrm{d}x\, \phi^*_n(x) V(x) \phi_m(x) \tag{3.45}$$

$$\approx \sum_{nm} T^*_{\alpha n} T_{\beta m} \sum_\gamma \frac{\omega_\gamma}{\omega(x_\gamma)} \phi_n(x_\gamma) V(x_\gamma) \phi_m(x_\gamma) \tag{3.46}$$

$$= \sum_\gamma \frac{\omega_\gamma}{\omega(x_\gamma)} \theta_\alpha(x_\gamma) V(x_\gamma) \theta_\beta(x_\gamma) \tag{3.47}$$

$$= \sum_\gamma \frac{\omega_\gamma}{\omega(x_\gamma)} \sqrt{\frac{\omega(x_\alpha)}{\omega_\alpha}} \delta_{\alpha\gamma} V(x_\gamma) \sqrt{\frac{\omega(x_\beta)}{\omega_\beta}} \delta_{\beta\gamma} \tag{3.48}$$

$$= \delta_{\alpha\beta} V(x_\beta). \tag{3.49}$$

Here we explicitly performed the transition to the FBR basis and back, to show that the DVR basis has the same level of accuracy as the FBR basis. This is expected, as the transformation $T_{\alpha n}$ between the two basis sets is unitary. Any error comes from the quadrature approximation, because we are approximating the integrals as sums. However, in the DVR basis the sums only contain a single non-zero element, at the grid-point of the matching bra and ket.

The other way to consider the error of the potential energy matrix is by recalling that the position operator is diagonal in the DVR basis. By Taylor expanding the potential energy operator and invoking the resolution of the identity, we can achieve the same result. We write the Taylor expansion of a function $f$ as

$$f(x) = \sum_n \frac{f^{(n)}(0)(x)^n}{n!}. \tag{3.50}$$

We define the operator evaluated function $f(\hat{A})$ by using the Taylor expansion of $f(x)$ and inserting the operator $\hat{A}$ instead of $x$,

$$\hat{f}(\hat{A}) = \sum_n \frac{f^{(n)}(0)\hat{A}^n}{n!}. \tag{3.51}$$

The spectral decomposition of $\hat{A}$ is, in a complete basis,

$$\hat{A} = \sum_{\alpha=0}^{\infty} \lambda_\alpha \left|\lambda_\alpha\right\rangle\!\left\langle\lambda_\alpha\right|, \tag{3.52}$$

where $(\lambda_\alpha, |\lambda_\alpha\rangle)$ for $\alpha = 0, \ldots, \infty$ is the eigenpairs of $\hat{A}$. Inserting into the

Taylor expansion we find the spectral decomposition of the operator function,

$$\hat{f}(\hat{A}) = \sum_n \frac{f^{(n)}(0)\left(\sum_\alpha \lambda_\alpha |\lambda_\alpha\rangle\langle\lambda_\alpha|\right)^n}{n!} \tag{3.53}$$

$$= \sum_\alpha |\lambda_\alpha\rangle\langle\lambda_\alpha| \sum_n \frac{f^{(n)}(0)\lambda_\alpha^n}{n!} \tag{3.54}$$

$$= \sum_\alpha f(\lambda_\alpha) |\lambda_\alpha\rangle\langle\lambda_\alpha| . \tag{3.55}$$

In the second step we used the orthonormality of the eigenvectors, $\langle\lambda_\alpha|\lambda_\beta\rangle = \delta_{\alpha\beta}$ to convert the product over sums $\left(\sum_\alpha \lambda_\alpha |\lambda_\alpha\rangle\langle\lambda_\alpha|\right)^n$, into a sum over products $\sum_\alpha |\lambda_\alpha\rangle\langle\lambda_\alpha| \lambda_\alpha^n$. We are left with an operator with the same eigenvectors $|\lambda_\alpha\rangle$, but with different eigenvalues $f(\lambda_\alpha)$. With a complete DVR basis, we could have used this exact relationship with $\hat{A} = \hat{X}$, and $\lambda_\alpha = \theta_\alpha$. However, the second step above is in effect a resolution the identity, which is not exact for a finite basis. Consequently, some approximation is made when evaluating the matrix elements in this way. We can write this approximation as

$$\langle\theta_\alpha|\hat{V}(x)|\theta_\beta\rangle = \hat{V}(x)_{\alpha\beta} \approx V(X^{DVR}_{\alpha\beta}) = V(x_\alpha)\delta_{\alpha\beta}. \tag{3.56}$$

Light and Carrington [30] name this error the *product approximation*, but for quadrature DVR the error is exactly the same as the quadrature error. This view of the error is, however, independent of the quadrature, and lets us generate a "product" DVR from any FBR basis. The only requirement is that we know both the kinetic energy matrix of the underlying FBR, and the transformation to the DVR basis with diagonal position operator $\hat{X}$. Importantly, one such basis is the sinc-DVR basis, which we will cover in the next section.

#### Additional note on the error

A downside of DVR is that the eigenvalue results of the hamiltonian is not variational. The potential energy operator is approximated as diagonal, and in the case of Gaussian-DVR, this error is equal to the quadrature error. Unfortunately, this error is not guaranteed to increase the energy, and might as well decrease the total energy below the true groundstate. However, this error quickly disappears with basis size, at least for lower lying states, due to the accuracy of Gauss quadrature.

## 3.3 Sinc-DVR

Many other DVRs than Gauss-DVR have been developed, with examples such as the In this section we will cover the sinc-DVR basis, which is not based on orthogonal polynomials. Instead we start with a continuous Fourier basis for

$$|\phi_k(x)\rangle = \frac{e^{ikx}}{\sqrt{2\pi}}, \quad -\infty < x < \infty, \tag{3.57}$$

where $k$ is the continuous momentum component of the wavefunction. We restict ourselves to a band-limited part of the Hilbert space, which means that the momentum component is limited to $-K < k < K$. In appendix A we show

that for an infinite sized grid, the Fourier basis functions turn into sinc-DVR functions,

$$s_n(x) = \frac{\sin(\pi(x - x_n)/\Delta x)}{\pi(x - x_n)}, \tag{3.58}$$

The quadrature grid for sinc-DVR spans the entire real line, and is defined by the points $x_n = n\Delta x$ for $n = 0, \pm 1, \pm 2, \ldots$, with spacing $\Delta x$.

The elements of the kinetic energy operator are also derived in the appendix,

$$T_{mn} = \frac{(-1)^{m-n}}{2(\Delta x)^2} \begin{cases} \frac{\pi^2}{3}, & m = n, \\ \frac{2}{(m-n)^2}, & m \neq n. \end{cases} \tag{3.59}$$

The infinite order evaluation of the kinetic operator causes the kinetic energy operator to be evaluated exactly for functions which are also band-limited[]. This is only true when the size of the grid is infinite, and in practice we are forced to truncate the grid at some distance. However, the error that this introduces reduces quickly with the size of the grid representation, as we will see from the quality of the DVR basis.

# CHAPTER 4

# Many-body theory

## 4.1 Basic many body theory

Now that we are fairly certain of how to solve the single-particle Schroedinger equation, we look towards systems with more electrons. Due to the fermionic nature of electrons, we need a wavefunction that is antisymmetric to interchange of particles. A mathematical object implementing this property is the Slater determinant. Given $N$ identical particles at coordinates $\{\mathbf{x}_j\}_{j=1}^{N}$ (including spin) and a basis of $L \geq N$ single particle orbitals $\{\phi_i(\mathbf{x}_j)\}_{i=1}^{L}$, we represent a manybody state up to a sign using the Slater determinant as

$$|\Phi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)\rangle = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_{p_1}(\mathbf{x}_1) & \phi_{p_2}(\mathbf{x}_1) & \cdots & \phi_{p_n}(\mathbf{x}_1) \\ \phi_{p_1}(\mathbf{x}_2) & \phi_{p_2}(\mathbf{x}_2) & \cdots & \phi_{p_n}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p_1}(\mathbf{x}_N) & \phi_{p_2}(\mathbf{x}_N) & \cdots & \phi_{p_n}(\mathbf{x}_N) \end{vmatrix} \tag{4.1}$$

where $p_i \in 1, \ldots, L$. As an example, consider the two-particle case. A wavefunction consisting of two orbitals $\phi_a(\mathbf{x})$ and $\phi_b(\mathbf{x})$ for two particles of coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ becomes

$$|\Phi(\mathbf{x}_1, \mathbf{x}_2)\rangle = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_a(\mathbf{x}_1) & \phi_b(\mathbf{x}_1) \\ \phi_a(\mathbf{x}_2) & \phi_b(\mathbf{x}_2) \end{vmatrix} = \frac{1}{\sqrt{2}} (\phi_a(\mathbf{x}_1)\phi_b(\mathbf{x}_2) - \phi_a(\mathbf{x}_2)\phi_b(\mathbf{x}_1)), \tag{4.2}$$

As expected, the explicit expression is antisymmetric under switching the coordinates of the two particles,

$$|\Phi(\mathbf{x}_2, \mathbf{x}_1)\rangle = \frac{1}{\sqrt{2}} (\phi_a(\mathbf{x}_2)\phi_b(\mathbf{x}_1) - \phi_a(\mathbf{x}_1)\phi_b(\mathbf{x}_2)) = - |\Phi(\mathbf{x}_1, \mathbf{x}_2)\rangle . \tag{4.3}$$

We will build approximate solutions to the many-body wave functions using the Slater determinants in two ways. The first involves rotating the basis set to optimize a single Slater determinant, such as for Hartree-Fock. The other is to add together different Slater determinants built from the same basis. The latter is necessary to account for many-body correlations in the system, and is the principle of all post Hartree-Fock methods. Orbital adaptive coupled-cluster theory, which is an important part of this thesis project, rotates the orbitals *and* combines the Slater determinants built from these using the coupled cluster expansion of the wave function. As such it combines both of these approaches simultaneously.

## 4.2 Second quantization

Using the Slater determinants directly to describe many-body fermionic systems can be cumbersome, and we want a unified description of systems of arbitrary basis sets and particle numbers. Introducing a Fock space we are able to consider all possible determinants resulting from a set of basis functions. A Fock space is simply put a direct product of Hilbert spaces for different number of particles.

### 4.2.1 Occupation Number representation

We will in the coming sections be representing the basis elements of the Fock space with the Occupation-Number (ON) representation. Througout we follow the presentation given by Helgaker et al. [21] closely.

In the ON representation, an occupation number is associated with each single particle basis function, giving the number of particles that inhabit each orbital. The Pauli exclusion principle states that the only possible occupation number for a fermionic single particle orbital is 0 or 1, a property related to the antisymmetric nature of the wavefunction. We can therefore represent an orthonormal many body basis vector in the ON representation in terms of a binary string $\mathbf{k}$ consisting of zeros and ones,

$$|\mathbf{k}\rangle = |k_1, k_2, \ldots, k_L\rangle\,, \quad k_p = \begin{cases} 0 & \text{if } \phi_p \text{ occupied,} \\ 1 & \text{if } \phi_p \text{ unoccupied.} \end{cases} \tag{4.4}$$

A general nonorthonormal vector is defined later. For any Fock space we can produce one-to-one mappings between the ON basis vectors and the Slater determinants with spin orbitals in a given ordering, the latter forming a basis of the Fock space. Consequently, the ON representation naturally also forms a basis for the Fock space. For a Fock space of $L$ single particle orbitals with an arbitrary number of particles $N = 0, \ldots, L$, we now see that the total number of Slater determinants is $N_s = 2^L$. If instead the number of particles is fixed, then the number of possible Slater determinants is given by the binomial coefficient,

$$N_s = \binom{L}{N} = \frac{L!}{(L-N)!N!}. \tag{4.5}$$

The inner product of two ON-vectors is given by

$$\langle \mathbf{k}|\mathbf{m}\rangle = \delta_{\mathbf{k},\mathbf{m}} = \prod_{p=1}^{L} \delta_{k_p m_p}. \tag{4.6}$$

With the inner product we now have the necessary tools to do mathematical operations on wavefunctions. States are vectors written in terms of the basis elements $|\mathbf{k}\rangle$ with coefficients $c_{\mathbf{k}}$ for each basis element,

$$|\mathbf{c}\rangle = \sum_{\mathbf{k}} c_{\mathbf{k}} \,|\mathbf{k}\rangle\,. \tag{4.7}$$

The inner product between two states $|\mathbf{c}\rangle$ and $|\mathbf{d}\rangle$ is

$$\langle \mathbf{c}|\mathbf{d}\rangle = \sum_{\mathbf{km}} c^*_{\mathbf{k}} d_{\mathbf{m}} \,\langle \mathbf{k}|\mathbf{m}\rangle = \sum_{\mathbf{k}} c^*_{\mathbf{k}} d_{\mathbf{k}}, \tag{4.8}$$

and the normalization condition for a state is given by

$$\langle \mathbf{c}|\mathbf{c}\rangle = \sum_{\mathbf{k}} |c_{\mathbf{k}}|^2 = 1. \tag{4.9}$$

The vacuum state is defined as the state with $k_p = 0\,\forall p$,

$$|-\rangle \equiv |0, 0, \ldots, 0\rangle\,. \tag{4.10}$$

### 4.2.2 Creation operators

The next important building blocks of second quantization are the creation and annihilation operators, $a_p^\dagger$ and $a_p$ respectively. These are operators that change the number of particles in the state they act on. Starting with the creation operator, its action on an arbitrary ON-vector is defined depending on the occupancy of spin-orbital $p$,

$$a_p^\dagger\,|k_1, \ldots, k_p, \ldots, k_L\rangle \equiv \begin{cases} \Gamma_p^{\mathbf{k}}\,|k_1, \ldots, 1_p, \ldots, k_L\rangle\,, & \text{if} \quad k_p = 0, \\ 0, & \text{if} \quad k_p = 1, \end{cases} \tag{4.11}$$

where $\Gamma_p^{\mathbf{k}}$ is a phase factor and $1_p$ is shorthand for a 1 in the $p$'th position. The first case is just the addition of a fermion to an unoccupied position, and the second is a result of the Fermi exclusion principle. The sign-factor $\Gamma^{\mathbf{k}}$ comes from having to rearrange the vector; returning to the Slater determinant picture we can picture this as adding a column with $\phi_p$ all the way to the left (and an extra row for the extra coordinate), and subsequently switching pairs of columns until the basis function is in the correct position according to the chosen mapping of ON-vectors to Slater determinants. This gives us a real phase factor as $\Gamma_p^{\mathbf{k}} = \prod_{q=1}^{p-1}(-1)^{k_q}$, since all the occupied states to the left of $p$ contribute one column switch, i.e. a minus sign. We will not give $\Gamma_p^{\mathbf{k}}$ much attention, as it drops out in the coming expressions.

We can rewrite eq. (4.11) as

$$a_p^\dagger\,|\mathbf{k}\rangle = \delta_{k_p 0}\Gamma_p^{\mathbf{k}}\,|k_1, \ldots, 1_p, \ldots, k_L\rangle\,. \tag{4.12}$$

The creation operator is nilpotent, as its action on any vector twice is zero,

$$(a_p^\dagger)^2\,|\mathbf{k}\rangle = a_p^\dagger a_p^\dagger\,|\mathbf{k}\rangle = \Gamma_p^{\mathbf{k}}\delta_{k_p 0}a_p^\dagger\,|k_1, \ldots, 1_p, \ldots, k_L\rangle = 0 \tag{4.13}$$

For two different creation operators, $a_p^\dagger$ and $a_q^\dagger$, the effect on a state is dependent on the ordering of $p$ and $q$. Letting $p < q$ gives

$$\begin{aligned} a_p^\dagger a_q^\dagger\,|\mathbf{k}\rangle &= a_p^\dagger \delta_{k_q 0}\Gamma_q^{\mathbf{k}}\,|\ldots, k_p, \ldots, 1_q, \ldots\rangle \\ &= \delta_{k_p 0}\Gamma_p^{\mathbf{k}}\delta_{k_q 0}\Gamma_q^{\mathbf{k}}\,|\ldots, 1_p, \ldots, 1_q, \ldots\rangle \end{aligned} \tag{4.14}$$

while the commuted case gives

$$\begin{aligned} a_q^\dagger a_p^\dagger\,|\mathbf{k}\rangle &= a_q^\dagger \delta_{k_p 0}\Gamma_p^{\mathbf{k}}\,|\ldots, 1_p, \ldots, k_q, \ldots\rangle \\ &= \delta_{k_q 0}(-\Gamma_q^{\mathbf{k}})\delta_{k_p 0}\Gamma_p^{\mathbf{k}}\,|\ldots, 1_p, \ldots, 1_q, \ldots\rangle\,. \end{aligned} \tag{4.15}$$

We can see that these two expressions are equal up to a sign difference due to the extra occupancy of $p$ when applying $a_q^\dagger$. When $q = p$, both expressions are zero due to the nilpotency of the creation operator, and $q < p$ follows from simply renaming $p$ and $q$. Adding eq. (4.14) and eq. (4.15), we see that the two operators fulfill the anticommutation relations when acting on a test vector $|\mathbf{k}\rangle$,

$$\{a_p^\dagger, a_q^\dagger\} \, |\mathbf{k}\rangle = \left(a_p^\dagger a_q^\dagger + a_q^\dagger a_p^\dagger\right) |\mathbf{k}\rangle = \left(a_p^\dagger a_q^\dagger - a_p^\dagger a_q^\dagger\right) |\mathbf{k}\rangle = 0. \tag{4.16}$$

This holds for any vector $|\mathbf{k}\rangle$, so we might as well drop it and write the anticommutation relations on the form

$$\{a_p^\dagger, a_q^\dagger\} \, = 0. \tag{4.17}$$

### 4.2.3 Annihilation operators

Annihilation operators are the hermitian adjoint of the creation operators, defined as $a_p \equiv (a_p^\dagger)^\dagger$. They remove particles from a given state, and all their properties can be found from the creation operators. Its anticommutation relation is found by hermitian conjugation of eq. (4.17),

$$\{a_p, a_q\} \, = \{a_p^\dagger, a_q^\dagger\}^\dagger = 0. \tag{4.18}$$

To find the action of an annihilation operator on a state $|\mathbf{k}\rangle$ we insert an identity $1 = \sum_{\mathbf{m}} |\mathbf{m}\rangle\langle\mathbf{m}|$ and take the complex conjugate of the resulting inner product,

$$\begin{aligned} a_q \, |\mathbf{k}\rangle &= \sum_{\mathbf{m}} |\mathbf{m}\rangle \, \langle\mathbf{m}|a_q|\mathbf{k}\rangle = \sum_{\mathbf{m}} |\mathbf{m}\rangle \, \langle\mathbf{k}|a_q^\dagger|\mathbf{m}\rangle^* \\ &= \sum_{\substack{\mathbf{m} \\ m_q = 0}} |\mathbf{m}\rangle \, \langle\mathbf{k}|\Gamma_q^{\mathbf{m}}|m_1, \ldots 1_q, \ldots m_L\rangle^* \, , \end{aligned} \tag{4.19}$$

Notice that we already have the restriction $m_q = 0$ from the action of the creation operator, and the fact that the phase factor is real. We find restrictions on the remaining indices in $\mathbf{m}$ by using the definition of the inner product of eq. (4.6),

$$\begin{aligned} a_q \, |\mathbf{k}\rangle &= \sum_{\substack{\mathbf{m} \\ m_q = 0}} |\mathbf{m}\rangle \, \Gamma_q^{\mathbf{m}} \left( \prod_{\substack{p=1 \\ p \neq q}}^{L} \delta_{k_p m_p} \right) \delta_{k_q 1} \\ &= \Gamma_q^{\mathbf{m}} \delta_{k_q 1} \, |k_1, \ldots, 0_q, \ldots, k_L\rangle \, . \end{aligned} \tag{4.20}$$

The annihilation operator $a_p$ has the opposite effect of the creation operator and removes a state from position $p$. It has the same phase factor, as the same number of permutations on antisymmetric states have to be performed to move state $p$ to the front of the operator.

### 4.2.4 Anticommutation relations

The final result we need for the annihilation and creation operators is the anticommutation relations between operators of different kind, $\{a_q^\dagger, a_p\}$. There are three cases: $p = q$, $p < q$ and $p > q$. Again we only consider the first two,

as the last is just a renaming of the second when it comes to anticommutation relations.

For the latter of the two, $p < q$, we have

$$\begin{aligned} a_p^\dagger a_q \left|\mathbf{k}\right\rangle &= a_p^\dagger \Gamma_q^{\mathbf{k}} \delta_{k_q 1} \left|k_1, \ldots k_p, \ldots, 0_q, \ldots, k_L\right\rangle \\ &= \Gamma_p^{\mathbf{k}} \Gamma_q^{\mathbf{k}} \delta_{k_p 1} \delta_{k_q 0} \left|k_1, \ldots, 1_p, \ldots, 0_q, \ldots, k_L\right\rangle . \end{aligned} \tag{4.21}$$

Similarly to eq. (4.15), the commuted version is equal up to an extra minus sign due to the extra occupancy when applying the second operator,

$$\begin{aligned} a_q a_p^\dagger \left|\mathbf{k}\right\rangle &= a_q \Gamma_p^{\mathbf{k}} \delta_{k_p 0} \left|k_1, \ldots 1_p, \ldots, k_q, \ldots, k_L\right\rangle \\ &= \Gamma_p^{\mathbf{k}} \left(-\Gamma_q^{\mathbf{k}}\right) \delta_{k_p 1} \delta_{k_q 0} \left|k_1, \ldots, 1_p, \ldots, 0_q, \ldots, k_L\right\rangle \\ &= -a_p^\dagger a_q \left|\mathbf{k}\right\rangle . \end{aligned} \tag{4.22}$$

This gives $\{a_p^\dagger, a_q\} = 0$. For $p = q$ on the other hand, we have

$$\begin{aligned} a_p^\dagger a_p \left|\mathbf{k}\right\rangle &= a_p^\dagger \Gamma_p^{\mathbf{k}} \delta_{k_p 1} \left|k_1, \ldots, 0_p, \ldots, k_L\right\rangle \\ &= \Gamma_p^{\mathbf{k}} \Gamma_p^{\mathbf{k}} \delta_{k_p 1} \left|k_1, \ldots, 1_p, \ldots, k_L\right\rangle = \delta_{k_p 1} \left|\mathbf{k}\right\rangle , \end{aligned} \tag{4.23}$$

and

$$\begin{aligned} a_p a_p^\dagger \left|\mathbf{k}\right\rangle &= a_p^\dagger \Gamma_p^{\mathbf{k}} \delta_{k_p 0} \left|k_1, \ldots, 1_p, \ldots, k_L\right\rangle \\ &= \Gamma_p^{\mathbf{k}} \Gamma_p^{\mathbf{k}} \delta_{k_p 0} \left|k_1, \ldots, 0_p, \ldots, k_L\right\rangle = \delta_{k_p 0} \left|\mathbf{k}\right\rangle . \end{aligned} \tag{4.24}$$

This is the only nonzero anticommutation relation, $\{a_p^\dagger, a_p\} = 1$. By putting these results together, we find the anticommutator of a creation and an annihilation operator as

$$\{a_p^\dagger, a_q\} = \delta_{pq}. \tag{4.25}$$

To summarize, the anticommutation relations of the fundamental operators $a_p$ and $a_p^\dagger$ are

$$\text{??}\{a_p, a_q\} = 0, \tag{4.26}$$

$$\{a_p^\dagger, a_q^\dagger\} = 0, \tag{4.27}$$

$$\{a_p^\dagger, a_q\} = \delta_{pq}. \tag{4.28}$$

### 4.2.5 Second quantized Hamiltonian

To complete the transition to second quantization, we need to make sure that our new mathematics give the same expectation values and matrix elements. These represent the only observables in quantum mechanics, and as such after a careful translation we will be ready to use our new theory.

In second-quantization, any one-particle operator can be written as[44]

$$\hat{h} = \sum_{pq} \langle \phi_p | \hat{h}^f | \phi_q \rangle \, a_p^\dagger a_q \equiv \sum_{pq} h_q^p a_p^\dagger a_q, \tag{4.29}$$

where the superscript $f$ of $\hat{h}^f$ denotes that it is the corresponding first quantization operator. The matrix elements $h_q^p$ are given by

$$h_q^p = \langle \phi_p | \hat{h} | \phi_q \rangle = \int \mathrm{d}\mathbf{x} \, \phi_p^*(\mathbf{x}) \hat{h}(\mathbf{x}) \phi_q(\mathbf{x}). \tag{4.30}$$

Likewise, two-body operators can be written on second quantized form as

$$\begin{aligned}\hat{u} &= \frac{1}{2}\sum_{pqrs}\langle\phi_p\phi_q|u^f|\phi_r\phi_s\rangle\, a_p^\dagger a_q^\dagger a_s a_r \\ &= \frac{1}{2}\sum_{pqrs} u_{rs}^{pq} a_p^\dagger a_q^\dagger a_s a_r. \end{aligned} \tag{4.31}$$

Again we drop the superscript of $\hat{u}^f$. This operator has the following expectation value in the coordinate representation,

$$u_{rs}^{pq} = \langle\phi_p\phi_q|\hat{u}|\phi_r\phi_s\rangle = \int \mathrm{d}\mathbf{x}_1\, d\mathbf{x}_2\, \phi_p^*(\mathbf{x}_1)\phi_q^*(\mathbf{x}_2)\hat{u}(\mathbf{x}_1,\mathbf{x}_2)\phi_r(\mathbf{x}_1)\phi_s(\mathbf{x}_2). \tag{4.32}$$

All particles involved are identical particles, and as such two-body operators have the permutation symmetry $u_{rs}^{pq}$.

The two-body operator can also be rewritten in an alternate form that will prove useful,

$$\hat{u} = \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq} a_p^\dagger a_q^\dagger a_s a_r. \tag{4.33}$$

Here we used the anticommutation rules of $a_s$ and $a_r$ given in **??**, and defined the antisymmetric matrix elements

$$u_{rs,AS}^{pq} \equiv u_{rs}^{pq} - u_{sr}^{pq}. \tag{4.34}$$

Using Wick's theorem (which we will define later) result in expressions which are antisymmetric with respect to interchange of two particles, and using the already anti-symmetric matrix elements $u_{rs,AS}^{pq}$ will remove half of the terms.

The full form of the first-quantized hamiltonian for electronic problems, where only one- and two-body operators are involved, is then written as

$$\hat{H} = \sum_{pq} h_q^p \hat{a}_p^\dagger a_q + \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq} a_p^\dagger a_q^\dagger a_s a_r, \tag{4.35}$$

where the matrix elements $h_q^p$ and $u_{rs,AS}^{pq}$ are defined by the problem at hand.

### Note on notation

In many problems in quantum mechanics, there is a transformation from one single-particle basis to another,

$$|\phi_p\rangle = \sum_\alpha C_{\alpha p}\,|\chi_\alpha\rangle\,. \tag{4.36}$$

In such cases, we will consistently use two different sets of indices to separate the two basis sets; One will have exclusively greek indices, while the other will have exclusively latin ones. This lets us use a short-hand notation for matrix elements, where the index shows which basis function is involved. For an $n$-body operator $\hat{O}$, we write the matrix elements as

$$O_{\beta_1,\beta_2,\ldots,\beta_n}^{\alpha_1,\alpha_2,\ldots,\alpha_n} \equiv \langle\chi_{\alpha_1}\chi_{\alpha_2}\cdots\chi_{\alpha_n}|\hat{O}|\chi_{\beta_1}\chi_{\beta_2}\cdots\chi_{\beta_n}\rangle\,, \tag{4.37}$$

for the greek basis, and

$$O^{p_1,p_2,\dots,p_n}_{q_1,q_2,\dots,q_n} \equiv \langle \phi_{p_1}\phi_{p_2}\cdots\phi_{p_n}|\hat{O}|\phi_{q_1}\phi_{q_2}\cdots\phi_{q_n}\rangle\,, \tag{4.38}$$

for the latin basis.

### 4.2.6 Wick's theorem

A very important result for many body theory is Wick's theorem[11, 44]. The main purpose of Wick's theorem is to alleviate the calculation of vacuum expectation values of operators, and their action on the vacuuum state. We will now present two building blocks required for Wick's theorem, and then state the theorem itself.

#### Normal ordering

First, the normal ordering of a string (i.e. product) of creation and annihilation operators $\bar{A} = A_1\cdots A_n$, where $A_i \in \{a^\dagger, a\}$, is defined as any ordering of the string where all annihilation operators are to the right of all creation operators. Mathematically, we can write this as

$$\{\bar{A}\} \equiv (-1)^{|\sigma|}A_{\sigma(1)}\cdots A_{\sigma(n)} = (-1)^{|\sigma|}[\text{creation ops.}]\times[\text{annihilation ops.}]. \tag{4.39}$$

Here, $\sigma \in S_n$ is a permutation of the indices, and $|\sigma|$ is the sign of the permutation. The normal ordering is not unique, as any permutation of two operators in either the creation operator block or the annihilation operator block will lead to a change in the sign of the permutation, due to the anticommutation rules.

Notably, a normal ordered operator has zero vacuum expectation value,

$$\langle -|\{\bar{A}\}|-\rangle = 0. \tag{4.40}$$

#### Contractions

The other ingredient is the contraction. The contraction between two annihilation or creation operators $\hat{A}_1$ and $\hat{A}_2$ is defined as

$$\overbrace{A_1 A_2} \equiv \hat{A}_1\hat{A}_2 - \{\hat{A}_1 A_2\} = \langle -|A_1 A_2|-\rangle\,. \tag{4.41}$$

The four possible values of a contraction for creation and annihilation operators $a^\dagger_p$ and $a_p$ (relative to the vacuum state $|-\rangle$) is

$$\overbrace{a_p a_q} = 0, \qquad \overbrace{a^\dagger_p a^\dagger_q} = 0, \tag{4.42}$$

$$\overbrace{a^\dagger_p a_q} = 0, \qquad \overbrace{a_p a^\dagger_q} = \delta_{pq}. \tag{4.43}$$

The contraction of two operators $A_j$ and $A_k$ in a normal ordered string $\bar{A} = A_1\cdots A_j\cdots A_k\cdots A_n$, possibly separated by other operators, is found by permuting the string such that the operators are side by side outside the string. We regard this as a definition,

$$\{A_1\cdots\overbrace{A_j\cdots A_k}\cdots A_n\} \equiv (-1)^{|\sigma|}\overbrace{A_j A_k}\{A_{\sigma(3)}\cdots A_{\sigma(n)}\}, \tag{4.44}$$

where the permutation $\sigma$ is now such that $\sigma(1) = j$ and $\sigma(2) = k$. For normal ordered strings containing contractions of multiple pairs of operators, this operation is done for each pair subsequently.

### Wick's theorem

Finally, Wick's theorem states that any string of operators $\bar{A} = A_1 \cdots A_n$ can be written as a sum of all normal ordered terms, where the sum is over all possible combinations of contractions performed between operators. The formal way of writing Wick's theorem is

$$\begin{aligned}\bar{A} =& \{\bar{A}\} \\ &+ \sum_{\text{one contraction}} \{A_1 \cdots \overbrace{A_k \cdots A_j} \cdots A_n\} \\ &+ \dots \\ &+ \sum_{\lfloor n/2 \rfloor \text{ contraction}} \{\overbrace{A_1 \cdots \overbrace{A_k \cdots \overbrace{A_j} \cdots} A_n}\}. \end{aligned} \tag{4.45}$$

Here $\lfloor n/2 \rfloor$ is the floor function, equal to $n/2$ if $n$ is even, and $(n-1)/2$ if $n$ is odd.

Note that if $n$ is even, then the terms in the final sum have no uncontracted operators, while in the case that $n$ is odd there is always one remaining uncontracted operator. This has consequences when considering the vacuum expectation value of second quantized operators, which is the main purpose of Wick's theorem. In fact, we see that the vacuum expectation value of any operator is given by the final sum in eq. (4.45),

$$\begin{aligned}\langle -|\bar{A}|-\rangle &= \sum_{\lfloor n/2 \rfloor \text{ contraction}} \langle -|\{\overbrace{A_1 \cdots \overbrace{A_k \cdots \overbrace{A_j} \cdots} A_n}\}|-\rangle \\ &= \begin{cases} \sum_{\lfloor n/2 \rfloor \text{ contraction}} \{\overbrace{A_1 \cdots \overbrace{A_k \cdots A_j} \cdots A_n}\}, & \text{if } n \text{ even}, \\ 0, & \text{if } n \text{ odd}. \end{cases} \end{aligned} \tag{4.46}$$

### Generalization

Consider that applying Wick's theorem to a string of operators that is already normal ordered does nothing. This is because the only non-zero contraction possible is $\overbrace{a_p a_q^\dagger}$, which can't be part of a normal ordered string. The same reasoning holds for a substring of operators within the original string; all contractions within a normal ordered substring give zero. This brings us to the generalized Wick's theorem, which states that for a string of operators $\bar{A}\bar{B}\cdots\bar{Z}$, where $\bar{A}$, $\bar{B}$ and $\bar{Z}$ are normal ordered substrings, we get the same result as in eq. (4.45), but with no contractions within a single substring.

## 4.3 Fermi Vacuum

With Wick's theorem in place, we can calculate the vacuum expectation value of any second quantized operator $\bar{A}$. Furthermore, consider that any two basis

Slater determinants $|\Phi_P\rangle$ and $|\Phi_Q\rangle$ can be written

$$|\Phi_P\rangle = \prod_{p=1}^{N} a_{P_p}^\dagger \, |-\rangle \,, \tag{4.47}$$

$$|\Phi_Q\rangle = \prod_{q=1}^{N} a_{Q_q}^\dagger \, |-\rangle \,, \tag{4.48}$$

where $P$ and $Q$ are sets of indices defining which orbitals are occupied in the two Slater determinants. Consequently, any expectation value or matrix element of Slater determinants is on the following form,

$$\langle \Phi_P | \bar{A} | \Phi_Q \rangle = \langle - | \left( \prod_{p=1}^{N} a_{P_{N-p}} \right) \bar{A} \left( \prod_{q=1}^{N} a_{Q_q}^\dagger \right) | - \rangle \,. \tag{4.49}$$

This is the form of eq. (4.46), which mean that we can use Wick's generalized theorem to calculate the expressions. Even though this is alot simpler than performing all anticommutations manually, the number of possible contractions increase very quickly with the number of particles. In addition, any theory derived from such a vacuum state would become particle dependent, as adding a single particle adds a pair of contractions and changes expression for expectation values of operators.

Instead, we define the Fermi vacuum as part of the solution to the stated problem. The Fermi vacuum is also called the reference state, and is the Slater determinant built from the $N$ first orbitals in the given single-particle basis $\{\phi_p\}_{p=1}^{L}$,

$$|\Phi\rangle \equiv \prod_{i=1}^{N} a_i^\dagger \, |-\rangle \,. \tag{4.50}$$

We must now take care when chosing the functions in $\{\phi_p\}_{p=1}^{L}$ and their ordering, as changing the basis functions will change the reference determinant. Formally, the single particle basis is now divided into two distinct sets,

$$\{\phi_p\}_{p=1}^{L} = \{\phi_i\}_{i=1}^{N} \cup \{\phi_a\}_{a=N+1}^{L}. \tag{4.51}$$

The orbitals in the former are called *occupied* states, while the remaining are called *unoccupied* or *virtual* states. To make the distinction, we use the eight first letters for virtual states,

$$a, b, \ldots, h > N, \tag{4.52}$$

the next eight for occupied states,

$$i, j, \ldots, o \leq N, \tag{4.53}$$

and the remaining letters of the alphabet designate an arbitrary state from both sets,

$$p, q, \cdots = 1, 2, \ldots. \tag{4.54}$$

This is especially important when using the Einstein summation convention, as the type of the index will determine the range of the summation.

### 4.3.1 Excitation and de-excitation operators

We define excitation operators as operators that replace occupied states with virtual states,

$$\hat{X}_{ij\ldots}^{ab\ldots} \equiv \cdots a_b^\dagger a_j a_a^\dagger a_i. \tag{4.55}$$

*Excited states* are created by acting on the reference state with an excitation operator,

$$\left|\Phi_{ij\ldots}^{ab\ldots}\right\rangle \equiv X_{ij\ldots}^{ab\ldots} \left|\Phi_0\right\rangle. \tag{4.56}$$

A general excitation index is written $\mu = \{ij\ldots, ab\ldots\}$, and can represent any excitation level. This lets us write a general excited state as

$$\left|\Phi_\mu\right\rangle = \hat{X}_\mu \left|\Phi_0\right\rangle. \tag{4.57}$$

The adjoint of the excitation operator is the deexcitation operator, or relaxation operator, which replaces virtual states with occupied ones,

$$[\hat{X}_{ij\ldots}^{ab\ldots}]^\dagger \equiv \hat{X}_{\ldots ba}^{\ldots ji} = a_i^\dagger a_a a_j^\dagger a_b \cdots. \tag{4.58}$$

Note that the deexcitation operator annihilates the reference state, as the final operator is an annihilation operator.

### 4.3.2 Quasiparticles

Continuing with the treatment of the Fermi vacuum, we recognize two sets of creation and annihilation operators that annihilate the Fermi vacuum,

$$a_i^\dagger \left|\Phi_0\right\rangle = 0, \qquad\qquad a_a \left|\Phi_0\right\rangle = 0. \tag{4.59}$$

The former is due to the Pauli exclusion principle, and the latter is due to the orbital $\phi_a$ being unoccupied in $\left|\Phi_0\right\rangle$. Accordingly we define the so called quasiparticle annihilation operators as

$$b_p = \begin{cases} a_p & \text{for} \quad p > N, \\ a_p^\dagger & \text{for} \quad p \leq N. \end{cases} \tag{4.60}$$

Its adjoint is the quasiparticle creation operator,

$$b_p^\dagger = \begin{cases} a_p^\dagger & \text{for} \quad p > N, \\ a_p & \text{for} \quad p \leq N. \end{cases} \quad . \tag{4.61}$$

We see that these operators have the same behaviour as regular creation and annihilation operators when acting on virtual states $p > N$, where $b_a^\dagger$ creates particles in unoccupied states and $b_a$ removes them. For occupied states, however, they serve the opposite role; a creation operator $b_p^\dagger$ with $p \leq N$ instead creates a hole in the Fermi vacuum, and the annihilation operator $b_p$ with $p \leq N$ annihilates the hole.

With respect to the creation of the quasiparticles (that are particles for virtual states and holes for occupied states), the Fermi vacuum is indeed the vacuum,

which is where the name comes from. The anticommutation relations are preserved,

$$\left\{b_p, b_q^\dagger\right\} = \delta_{pq}, \qquad \{b_p, b_q\} = \left\{b_p^\dagger, b_q^\dagger\right\} = 0. \tag{4.62}$$

Normal ordering with respect to the Fermi vacuum gives us the contraction of the quasiparticle operators. Contractions between quasiparticle operators are defined relative to the fermi vacuum, where they have the same contraction results as their regular conterparts, and using eq. (4.41) we find them to be

$$\begin{aligned}
\overbrace{b_p b_q} &\equiv \langle\Phi_0|b_p b_q|\Phi_0\rangle = 0,\\
\overbrace{b_p^\dagger b_q} &\equiv \langle\Phi_0|b_p^\dagger b_q|\Phi_0\rangle = 0,\\
\overbrace{b_p^\dagger b_q^\dagger} &\equiv \langle\Phi_0|b_p^\dagger b_q^\dagger|\Phi_0\rangle = 0,\\
\overbrace{b_p b_q^\dagger} &\equiv \langle\Phi_0|b_p b_q^\dagger|\Phi_0\rangle = \delta_{pq}.
\end{aligned}$$

The two ingredients for Wick's theorem are as such in place, i.e. anticommutation rules and contractions with respect to a vacuum. This means that we can use Wick's theorem with respect to the Fermi vacuum. This proves to be a very large simplification, because all the creation operators needed to create the reference state are absorbed into the Fermi vacuum. The remaining operators come from either excitation operators mentioned above, or from Hamiltonian, which has a maximum of four operators for a two-body Hamiltonian. This is used in many-body theories such as Configuration Interaction and coupled-cluster, where the missing correlation is found by exciting the reference state.

### 4.3.3 Normal ordered Hamiltonian

The Hamiltonian expectation value of the reference state is called the reference energy,

$$E_0 \equiv \langle\Phi_0|\hat{H}|\Phi_0\rangle\,. \tag{4.63}$$

It is also common to introduce the normal ordered Hamiltonian $H_N$ as the Hamiltonian minus its expectation value in the Fermi vacuum,

$$H_N \equiv \hat{H} - \langle\Phi_0|\hat{H}|\Phi_0\rangle = \hat{H} - E_0. \tag{4.64}$$

To find the explicit expressions for these two values, we apply Wick's theorem to the two-body Hamiltonian $\hat{H}$ relative to the Fermi vacuum. Recall that $\hat{H}$ is given in terms of the one-body operator $\hat{h}$ and the two-body operator $\hat{u}$,

$$\hat{H} = \hat{h} + \hat{u} = \sum_{pq} h_q^p \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r. \tag{4.65}$$

By converting the $\hat{a}_p^\dagger$ and $\hat{a}_q$ to their quasi-particle relatives, applying Wick's theorem, and converting back again, we get the following result for the one-body operator,

$$\hat{h} = \sum_i h_i^i + \sum_{pq} h_q^p \{\hat{a}_p^\dagger \hat{a}_q\}, \tag{4.66}$$

where $\{\dots\}$ determines normal ordering relative to the Fermi vacuum. Correspondingly for the two-body operator, we can refactor it into the following terms

$$\hat{u} = \frac{1}{2}u_{ij}^{ij} + \sum_{pq} u_{qi}^{pi}\{\hat{a}_p^\dagger \hat{a}_q\} + \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq}\{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r\}. \tag{4.67}$$

The normal ordered strings annihilates the vacuum, and as such the vacuum expectation value is given by the fully contracted terms of the two above operators,

$$E_0 = h_i^i + \frac{1}{2}u_{ij}^{ij}. \tag{4.68}$$

The normal ordered Hamiltonian $H_N$ is determined by the remaining terms,

$$\hat{H}_N = \sum_{pq}\left(h_q^p\{\hat{a}_p^\dagger \hat{a}_q\} + \sum_i u_{qi}^{pi}\{\hat{a}_p^\dagger \hat{a}_q\}\right) + \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq}\{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r\} \tag{4.69}$$

$$= \hat{F}_N + \hat{V}_N. \tag{4.70}$$

We defined the Fock operator $\hat{F}_N$ and the normal ordered two-body operator $\hat{V}_N$ to be the normal-ordered one-body and two-body terms of $\hat{H}$, respectively,

$$\hat{F}_N \equiv \sum_{pq}\left(h_q^p\{\hat{a}_p^\dagger \hat{a}_q\} + \sum_i u_{qi}^{pi}\{\hat{a}_p^\dagger \hat{a}_q\}\right), \tag{4.71}$$

$$\hat{V}_N \equiv \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq}\{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r\}. \tag{4.72}$$

### 4.3.4 Density matrices

An alternative representation of the expectation values of operators in a state is given in terms of density matrices[31]. For a general one- and two-electron hermitian operator, we have the form

$$\hat{\Omega} = \sum_{pq}\Omega_q^p a_p^\dagger a_q + \frac{1}{2}\sum_{pqrs}\Omega_{rs}^{pq} a_p^\dagger a_r^\dagger a_s a_q + \Omega_0. \tag{4.73}$$

The expectation value of this in a normalized state $|\Psi\rangle$, given by

$$|\Psi\rangle = \sum_{\mathbf{k}} c_{\mathbf{k}} |\mathbf{k}\rangle, \quad \langle\Psi|\Psi\rangle = 1, \tag{4.74}$$

is then

$$\langle\Psi|\hat{\Omega}|\Psi\rangle = \sum_{pq}\rho_p^q\Omega_q^p + \frac{1}{2}\sum_{pqrs}\rho_{pq}^{rs}\Omega_{rs}^{pq} + \Omega_0. \tag{4.75}$$

We here introduced the one- and two-body density matrices

$$\rho_p^q \equiv \langle\Psi|a_p^\dagger a_q|\Psi\rangle, \tag{4.76}$$

$$\rho_{pq}^{rs} \equiv \langle\Psi|a_p^\dagger a_r^\dagger a_s a_q|\Psi\rangle. \tag{4.77}$$

They hold all the information of the quantum state in question, and will be used in the calculation of various observables. Anticommutation of the annihilation and creation operators show us that $\rho_{pq}^{rs}$ has the same symmetries as the anti-symmetrised two-body operator $u_{AS}$,

$$\rho_{pq}^{rs} = -\rho_{qp}^{rs} = -\rho_{pq}^{sr} = \rho_{qp}^{sr}. \tag{4.78}$$

## 4.4 Hartree-Fock

Hartree Fock (HF) is a that seeks to find the single Slater determinant with the lowest energy, i.e. we minimize the energy function

$$E_{HF}(\Phi) = \langle \Phi | \hat{H} | \Phi \rangle \,. \tag{4.79}$$

The Slater determinant that minimizes $E_{HF}(\Phi)$ is then denoted $|\Phi_{HF}\rangle$. The HF Slater determinant is given in terms the HF orbitals $\phi_i$,

$$|\Phi_{HF}\rangle = |\phi_1 \phi_2 \cdots \phi_N\rangle \,. \tag{4.80}$$

The orbitals are required to be orthonormal,

$$\langle \phi_i | \phi_j \rangle = \delta_{ij}, \quad i,j = 1, \ldots, N, \tag{4.81}$$

such that the HF Slater determinant is normalized,

$$\langle \Phi_{HF} | \Phi_{HF} \rangle = 1. \tag{4.82}$$

With $H$ on the second quantized form of eq. (4.35), the expectation value functional under the Hartree Fock ansatz becomes

$$\mathcal{E}_{HF}[\Phi_{HF}] = \langle \Phi_{HF} | H | \Phi_{HF} \rangle = \sum_i \langle \phi_i | \hat{h} | \phi_i \rangle + \frac{1}{2} \sum_{ij} \langle \phi_i \phi_j | \hat{u} | \phi_i \phi_j \rangle \,. \tag{4.83}$$

Using the variational method, Szabo and Ostlund [47] shows that extremizing eq. (4.83) corresponds to solving the Hartree-Fock equations for the orbitals,

$$\left[ \hat{h} + \sum_{j \neq i} \hat{\mathcal{L}}_j - \sum_{j \neq i} \hat{\mathcal{K}}_j \right] |\phi_i\rangle = \varepsilon_i \, |\phi_i\rangle \,, \tag{4.84}$$

where $\hat{\mathcal{J}}_j$ and $\hat{\mathcal{K}}_j$ are the direct and exchange terms. They are first-quantized operators dependent on all the other particle states,

$$\hat{\mathcal{J}}_j \, |\phi_i\rangle \equiv \left[ \int dx_2 \phi_j(x_2) \hat{u}(x_1, x_2) \phi_j(x_2) \right] |\phi_i\rangle \,, \tag{4.85}$$

$$\hat{\mathcal{K}}_j \, |\phi_i\rangle \equiv \left[ \int dx_2 \phi_j(x_2) \hat{u}(x_1, x_2) \phi_i(x_2) \right] |\phi_j\rangle \,. \tag{4.86}$$

By recognizing that $[\hat{\mathcal{K}}_i - \hat{\mathcal{J}}_i]\phi_i = 0$, we can remove the summation restrictions in eq. (4.84) and introduce the *fock operator*,

$$\hat{f} \equiv \hat{h} + \sum_j \left[ \hat{\mathcal{J}}_j - \hat{\mathcal{K}}_j \right] . \tag{4.87}$$

The form of the hartree fock equations is now that of an eigenvalue problem,

$$\hat{f} \, |\phi_p\rangle = \varepsilon_p \, |\phi_p\rangle \,. \tag{4.88}$$

However, the non-locality of the $\hat{\mathcal{K}}$-operator makes solving this equations a non-trivial problem. In addition to being an eigenvalue equation, it is also an integro-differential equation, with integrals over all other solution present in the equation for each solution. Integro-differential equations are in general hard to solve directly, and we need some way of approximating solutions to the problem.

### 4.4.1 The Roothan Hall Equations

A common way to solve the Hartree Fock equations is to use a finite sized basis set $\{|\chi_\alpha\rangle\}_{\alpha=1}^{L}$ to approximate the one-dimensional Hilbert space. Note that we do not require the $|\chi_\alpha\rangle$'s to be orthogonal. This means that we define the overlap matrix $S_\beta^\alpha \equiv \langle\chi_\beta|\chi_\alpha\rangle$. The HF orbitals are expanded in this basis as

$$|\phi_p\rangle = \sum_{\alpha=1}^{L} C_{\alpha p}\,|\chi_\alpha\rangle\,, \quad \text{for} \quad 1 \leq p \leq L. \tag{4.89}$$

To find the transformation $C_{\alpha p}$, we insert eq. (4.89) into eq. (4.88) and left-project with $\langle\chi_\beta|$ to get

$$\sum_\alpha C_{\alpha p}\,\langle\chi_\beta|\hat{f}|\chi_\alpha\rangle = \epsilon_p \sum_\alpha C_{\alpha p}\,\langle\chi_\beta|\chi_\alpha\rangle\,. \tag{4.90}$$

This can be rewritten as a set of matrix eigenvalue equations called the Roothan-Hall equations[47],

$$F(C)C = \varepsilon SC. \tag{4.91}$$

Here we defined the Fock matrix $F$, overlap matrix $S$, eigenvalue vector $\epsilon$ and coefficient matrix $C$, with elements $F_\beta^\alpha = \langle\chi_\beta|\hat{f}|\chi_\alpha\rangle$, $S_\beta^\alpha = \langle\chi_\beta|\chi_\alpha\rangle$, $\epsilon_p$ and $C_p^\alpha$, respectively.

The Roothan-Hall equations are non-linear and need to be solved iteratively, because the elements of the Fock matrix are dependent on the coefficients,

$$\begin{aligned} F_\alpha^\beta = \langle\chi_\beta|\hat{f}|\chi_\alpha\rangle &= \langle\chi_\beta|\hat{h}|\chi_\alpha\rangle + \sum_{i=1}^{N} \langle\chi_\beta\phi_i|\hat{u}|\chi_\alpha\phi_i\rangle_{AS} \\ &= \langle\chi_\beta|\hat{h}|\chi_\alpha\rangle + \sum_{i=1}^{N}\sum_{\gamma,\delta=1}^{L} (C^\dagger)_\gamma^i C_i^\delta\,\langle\chi_\beta\chi_\gamma|\hat{u}|\chi_\alpha\chi_\delta\rangle_{AS}\,. \end{aligned} \tag{4.92}$$

### 4.4.2 Restricted Hartree Fock

For a more explicit treatment of spin, we split the Hilbert space into a spatial and a spin part, $\mathcal{H} = \mathcal{H}^{\text{spatial}} \otimes \mathcal{H}^{\text{spin}}$. Each basis function is split accordingly,

$$|\phi_p\rangle = |\varphi_p\rangle \otimes |\sigma_p\rangle\,, \tag{4.93}$$

where $|\varphi_p\rangle$ is a spatial orbital $\varphi(\vec{x})$ and $|\sigma_p\rangle$ is a general spinor, which is a vector in the two-dimensional Hilbert space $\mathcal{H}^{\text{spin}}$ (in the case of spin half particles).

If we restrict ourself to a spin-independent Hamiltonian and closed shell system, then we can usually (with some exceptions) make the assumption that the basis set reduces to a set of doubly occupied spatial orbitals,

$$|\phi_p\rangle = \left|\phi_{P\sigma_p}\right\rangle = |\varphi_P\rangle \otimes |\sigma_p\rangle\,, \tag{4.94}$$

where $P$ is an index over the set of spatial basis functions $\{|\varphi_P\rangle\}_{P=1}^{L/2}$ and $\sigma_p = 1, 2$ is an index of the spin basis functions $\{|\sigma_p\rangle\}_{\sigma_p=1,2}$. The spin basis functions are orthonormalized,

$$\langle\sigma_p|\sigma_q\rangle = \delta_{\sigma_p\sigma_q}. \tag{4.95}$$

Writing the original indices as composite indices, $p = (P, \sigma_p)$, etc, then matrix elements of the operators in Hartree-Fock theory become block-diagonal with two equal blocks,

$$h_q^p = \delta_{\sigma_p \sigma_q} h_Q^P, \tag{4.96}$$

and

$$u_{ri,AS}^{pi} = \langle \sigma_p | \sigma_q \rangle \langle \sigma_i | \sigma_i \rangle u_{QI}^{PI} - \langle \sigma_p | \sigma_i \rangle \langle \sigma_i | \sigma_q \rangle u_{IQ}^{PI} = 2 \langle \sigma_p | \sigma_q \rangle u_{QI}^{PI} - \langle \sigma_p | \sigma_q \rangle u_{IQ}^{PI}. \tag{4.97}$$

In Restricted Hartree Fock, we then represent only spatial part of the Fock matrices, and store only a single block given by

$$F_Q^P = h_Q^P + \sum_{I=1}^{N/2} \left( 2u_{QI}^{PI} - u_{IQ}^{PI} \right). \tag{4.98}$$

The coefficient matrices are also assumed to be block-diagonal, and the RHF Roothan-Hall equations are solved by only considering a single of the two blocks.

There is also a method called unrestricted Hartree-Fock (UHF), where the orbitals are allowed to be independent for different spin directions[47]. We will not consider the UHF method in this work, as we we do not utilize UHF for any results in this thesis, and as such we will not be presenting it here.

### 4.4.3 Time-dependent Hartree-Fock

As seen in eq. (4.88), the ansatz of HF to approximate the solution of the TISE with a single reference turns the problem into a single-particle problem, where solutions are independent of each other. The many-body parts of the Hamiltonian are projected out, and we end up with the Fock operator acting on the reference state. The same is the case in Time-Dependent Hartree-Fock (TDHF), where the TDSE for the time-dependent HF ansatz $|\Phi_{HF}(t)\rangle$ can be written as a single-particle problem[23],

$$-i \frac{\partial}{\partial t} |\phi_p(t)\rangle = \hat{f}(t) |\phi_p(t)\rangle, \tag{4.99}$$

where $\hat{f}(t)$ is the time-dependent Fock operator found by letting the one- and two-body operators of eq. (4.88) become time-dependent,

$$\hat{h}(x_1) \to \hat{h}(x_1, t), \qquad \hat{u}(x_1, x_2) \to \hat{u}(x_1, x_2, t). \tag{4.100}$$

As in section 4.4.1, we restrict the single-particle Hilbert space to a subset spanned by a finite basis $\{|\chi_\alpha\rangle\}_{\alpha=1}^{L}$. Note that the basis is independent of time, and the time-evolution of the Hartree Fock orbitals is now given in terms of the time-dependent expansion coefficients $C_p^\alpha(t)$. The time-dependent orbitals are explicitly given by

$$|\phi_p(t)\rangle = \sum_\alpha C_p^\alpha(t) |\chi_\alpha\rangle. \tag{4.101}$$

Inserting this into eq. (4.99), we get the following equation for the time-evolution of $C(t)$,

$$iS\dot{C}(t) = F(t)C(t). \tag{4.102}$$

Here $F(t)$ is the time-dependent Fock matrix, defined as the matrix elements of the time-dependent fock operator,

$$F_\alpha^\beta(t) = \langle\chi_\beta|\hat{h}(t)|\chi_\alpha\rangle + \sum_{i=1}^{N}\sum_{\gamma,\delta=1}^{L}(C(t)^\dagger)_\gamma^i C_i^\delta(t)\left\langle\chi_\beta\chi_\gamma|\hat{u}(t)|\chi_\alpha\chi_\delta\right\rangle_{AS}. \quad (4.103)$$

Note that the time-dependence of $F$ is both due to the time-dependence of the operators $\hat{h}(t)$ and $\hat{u}(t)$ and due to the coefficients $C(t)$. The overlap matrix $S_\beta^\alpha = \langle\chi_\alpha|\chi_\beta\rangle$ present in eq. (4.102), and can be handled by either using a diagonalization algorithm[47] or by using orthonormal basis functions $S_\beta^\alpha = \delta_\beta^\alpha$.

## 4.5 Configuration Interaction

Having introduced the single-reference theory of Hartree-Fock, we now turn to the logical next step, which is to represent the wavefunction in terms of a linear combination of Slater determinants. This theory is called configuration interaction[47]. In Full Configuration Interaction theory (FCI), the wavefunction is given by a linear of all determinants that can be constructed from the given set of orbitals,

$$|\Psi_J\rangle = \sum_I C_{IJ}|\Phi_I\rangle. \quad (4.104)$$

Here $I$ is a general excitation index, such that $|\Phi_I\rangle$ represents both the reference determinant for $I = 0$ as well as all possible excitations of the reference determinants, and $C_{IJ}$ is the matrix of coefficients determining the weight for each Slater determinant. The CI energy function is

$$\mathcal{E}(C) = \frac{\langle\Psi_J|H|\Psi_J\rangle}{\langle\Psi_J|\Psi_J\rangle}, \quad (4.105)$$

which has a dependency on the coefficient matrix $C = [C_{IJ}]$. To find the ground state, we employ the variational method,

$$0 = \frac{\partial}{\partial C_{IJ}}\mathcal{E}(C) = \frac{\partial}{\partial C_{IJ}}\frac{\langle\Psi_J|H|\Psi_J\rangle}{\langle\Psi_J|\Psi_J\rangle}. \quad (4.106)$$

Due to the simple form of the derivatives of the CI wave function,

$$\frac{\partial}{\partial C_{IJ}}|\Psi_J\rangle = |\Phi_I\rangle, \quad (4.107)$$

it is straight forward to work out the derivatives in eq. (4.106) and find the Schroedinger equation for the CI-wavefunction,

$$\langle\Phi_I|H|\Psi_J\rangle = E_J\langle\Phi_I|\Psi_J\rangle. \quad (4.108)$$

Inserting eq. (4.104), we see that this is an eigenvalue equation for the coefficients,

$$\sum_K H_{IK}C_{KJ} = E_J C_{IJ}, \quad (4.109)$$

where we assumed that $\langle\Phi_I|\Phi_J\rangle = \delta_{IJ}$, and we wrote the matrix elements of $H$ as $H_{IK} \equiv \langle\Phi_I|H|\Phi_K\rangle$. The coefficients $C_{IJ}$ are then given by the (unitary) matrix

that diagonalizes $H$, where the columns of $C$ give the different eigenfunctions of $H$.

The sum in eq. (4.104) is naturally truncated at $N$-fold excitations for an $N$-particle system. However, even then the number of Slater determinants $N_s$ for a given basis of size $L$ with $N$ particles scales as

$$N_s = \binom{L}{N}, \tag{4.110}$$

which is a factorial dependency of the basis size $L$. Thus, solving eq. (4.109) quickly becomes unfeasible for even moderate systems. However, FCI is still a valuable tool for benchmarking since it can provide exact solutions within the given basis, provided the system is small enough.

### 4.5.1 Truncated CI

A common technique to avoid the exponential scaling of FCI is to truncate the linear CI-expansion and include only a subset of excited Slater determinants. This reduces the cost to polynomial scaling with the number of basis functions, with the exponent dependent on the excitation level. However, this both reduces the accuracy of the method in a haphazard manner, and also destroys size extensivity and size consistency[11].

Shortly put, a size-consistent treatment of a system composite of two non-interacting subsystems will equal the same treatment applied to the two subsystems independently. When applying truncated CI to the composite system, the total excitation level of the states will be limited by the maximal excitation level of the CI-operator. On the other hand, applying CI truncated at the same level to each of the subsystems individually, we get twice a total excitation level which is twice that of the composite system. This causes a failure of Truncated CI to produce consistent results when noninteracting subsystems are put together. All in all, size consistency and size extensivity are important concepts for the study of quantum systems, and are discussed in more detail by Helgaker et al. [21].

## 4.6 Coupled-Cluster theory

Coupled-cluster was introduced in the late 60's by Čižek and Paldus [7–9]. It is often dubbed the gold standard of quantum chemistry, and has been used extensively to obtain ab initio calculations in both chemistry and nuclear physics.

Coupled-cluster is a post-Hartree-Fock method, which means that it provides an improvement of the description of correlation effects in a HF wavefunction, which is often used as the reference state. Further, it does not have the exponential scaling that FCI has, but is simultaneously a reliable and accurate step toward the full wavefunction. It is more affordable, as the cost of solving the CC equations scales polynomially with system size. Even though the exponent of the polynomial can be relatively large, $N^6$ for CCSD, the cost is low compared to the exponential scaling of Full Configuration Interaction theory.

Compared to truncated Configuration Interaction, it can be shown that if the corrections to the reference state can be considered a pertubation, then CC includes more orders of the correct wavefunction and energy than CI at a given truncation level[**paldus**]. This is because while the CI wavefunction only contains excitation up to the truncation level, the coupled-cluster wavefunction is represented by the exponential of the cluster operator, which allows for higher level excitations. In addition to this, it gives size-extensive results for the energy[11].

This section will cover the essentials of the coupled-cluster theory, starting with the exponential ansatz, via the amplitude equations to the bivariational formulation of coupled-cluster. We then introduce time-dependent coupled-cluster for static orbitals (TDCC), and the generalization to orbital-adaptive time-dependent coupled-cluster, where the orbitals also get a time-dependence similar to Hartree-Fock.

A thourough treatment of coupled-cluster theory can be found in the paper by Crawford and Schaefer [11] or textbook by Shavitt and Bartlett [44].

### 4.6.1 The exponential ansatz

We introduce the coupled-cluster method by rewriting the configuration interaction wavefunction into an exponential form,

$$|\Psi\rangle = (C_0 + \sum_{\mu} \hat{C}_\mu) |\Phi_0\rangle \rightarrow |\Psi\rangle = e^{\hat{T}} |\Phi_0\rangle . \tag{4.111}$$

This is the so called exponential ansatz of coupled-cluster, and it connects the wavefunction $|\Psi\rangle$ to the reference determinant $|\Phi_0\rangle$ through the cluster operator $\hat{T}$. The cluster operator is an excitation operator, and in the untruncated limit coupled-cluster converges to FCI, as then $\hat{T}$ includes all possible excitation and all slater determinants are included in the coupled-cluster wavefunction. The full cluster operator is split into excitation operators of different excitation level,

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \dots \tag{4.112}$$

where the $n'th$ level cluster operator $\hat{T}_n$ is given by a linear combination of all possible $n$-tuple excitations,

$$\hat{T}_n = \frac{1}{(n!)^2} \sum_{\substack{i_1,\dots,i_n \\ a_1,\dots,a_n}} \tau^{a_1,\dots,a_n}_{i_1,\dots,i_n} \{a_{i_1} a^\dagger_{a_1} \cdots a_{i_n} a^\dagger_{a_n}\} \tag{4.113}$$

Here $\tau_\mu$ are the cluster amplitudes, which are antisymmetric under change of any two indices of the same type (occupied or virtual). The prefactor $1/(n!)^2$ accounts for the unrestricted summations, which give multiple contributions from the same excitations, as switching two indices gives a minus from the amplitudes due to antisymmetry as well as from the operators due to the anticommutation rules.

As with truncated CI, we truncate the cluster operator to only include certain levels of excitation. However, higher order excitations are included in the coupled-cluster wavefunction due to the exponential ansatz. For a given

truncation level, CC includes more excitations than truncated CI. The different CC methods are named by which excitation levels are included in the cluster operator, such as coupled-cluster coubles (CCD), coupled-cluster singles doubles (CCSD), etc. We will mostly be working with CCD in our calculations, but we derive the theory independent of truncation level.

### 4.6.2 The Coupled-Cluster equations

Inserting the CC ansatz into the Schroedinger equation gives,

$$He^{\hat{T}} |\Phi_0\rangle = Ee^{\hat{T}} |\Phi_0\rangle . \tag{4.114}$$

Here $E$ is now the coupled-cluster energy. It is common to project out the reference energy and instead write the coupled-cluster equations in terms of the normal-ordered Hamiltonian $H_N$ as

$$H_N e^{\hat{T}} |\Phi_0\rangle = \Delta E_0 e^{\hat{T}} |\Phi_0\rangle . \tag{4.115}$$

The unknowns of this equation is the coupled-cluster correlation energy $\Delta E_0$, defined as the couped cluster energy minus the reference energy $\Delta E_0 = E - E_0$, and the amplitudes $t_\mu$. We see that applying the reference state $\langle\Phi_0|$ and excited states $\langle\Phi_\mu|$ from the left would yield one equation for each unknown. However, the resulting equations would be the unlinked form of the coupled-cluster equations. Even though solving them gives the correct coupled-cluster amplitudes, they are impractical for numerical implementation[21].

To obtain the linked form of the CC equations, we instead apply $e^{-\hat{T}}$ to eq. (4.115),

$$e^{-\hat{T}} H_N e^{\hat{T}} |\Phi_0\rangle = E_0 |\Phi_0\rangle . \tag{4.116}$$

Here we recognize a similarity transform of the normal ordered Hamiltonian $\bar{H}_N = e^{-\hat{T}} H_N e^{\hat{T}}$. Using the Baker-Campbell-Hausdorff-expansion, we can rewrite $\bar{H}$ as

$$\bar{H}_N \equiv e^{-\hat{T}} H_N e^{\hat{T}} = H_N + \Big[H_N, \hat{T}\Big] + \frac{1}{2!}\Big[\Big[H_N, \hat{T}\Big], \hat{T}\Big] \tag{4.117}$$

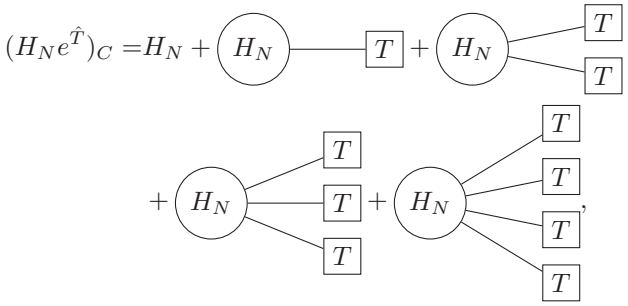
$$+ \frac{1}{3!}\Big[\Big[\Big[H_N, \hat{T}\Big], \hat{T}\Big], \hat{T}\Big] + \frac{1}{4!}\Big[\Big[\Big[\Big[H_N, \hat{T}\Big], \hat{T}\Big], \hat{T}\Big], \hat{T}\Big] \tag{4.118}$$

Here, the expansion is naturally truncated after four nested commutators for a Hamiltonian containing at most two-body operators, due to the connected cluster theorem[11, 44], which greatly simplifies any calculations. Another way to view this is by the so called connected-cluster form of the Schrödinger equation [8]. It was introduced by Čížek in 1966[8], where he showed that the Schroedinger equation could be written in a connected form,

$$(H_N e^{\hat{T}})_C |\Phi_0\rangle = \Delta E_0 |\Phi_0\rangle . \tag{4.119}$$

The subscript $C$ indicates that only terms where all operators are connected are included. This means that in each term, one should be able to find a "path" of contracted indices from one operator to any other, possibly via other operators. The cluster operators are excitation operators, but in terms of quasi-particle

annihilation operators they contain no annihilation operators. Consequently, contractions between them with respect to the Fermi vacuum give zero. For a term to be connected, every cluster operator thus has to be contracted with the Hamiltonian. We can represent this diagrammatically as




(4.120)

where each connecting line indicates that at least one contraction is made between the two operators.

Any of the forms of the CC equations above can be used to express and derive the coupled cluster amplitude equations. We will be using the final one due to its connection to diagrammatic coupled-cluster theory, to which [11] is an excellent introduction (even though this thesis does not contain diagrammatic coupled cluster theory explicitly, it has been our way to derive expressions for the coupled cluster amplitude equations). When using diagrams, the term connected has an intuitive meaning, as it means that parts of the diagram separate from the other parts.

### 4.6.3 Coupled-cluster energy

The coupled-cluster energy expression is found by projecting eq. (4.120) with the reference determinant,

$$\Delta E_0 = \langle \Phi_0 | (H_N e^{\hat{T}})_C | \Phi_0 \rangle \,. \tag{4.121}$$

By considering that all terms have to be fully connected and that the normal ordered Hamilton operator contains maximally two excitation operators, only the three following terms survive,

$$\Delta E_0 = \langle \Phi_0 | (F_N \hat{T}_1)_C + (V_N \hat{T}_2)_C + \frac{1}{2}(V_N \hat{T}_1^2)_C | \Phi_0 \rangle \,. \tag{4.122}$$

This can be expressed in terms of the singles-amplitudes $\tau_i^a$, doubles-amplitudes $\tau_{ij}^{ab}$, and the matrix elements of the one- and two-body parts of the normal orderd Hamiltonian, $f_a^i$ and $u_{ab}^{ij}$, respectively, as

$$\Delta E_0 = f_a^i \tau_i^a + \frac{1}{2} u_{ab}^{ij} \left[ \frac{1}{2} \tau_{ij}^{ab} + \tau_i^a \tau_j^b \right] \tag{4.123}$$

Se appendix A for a derivation. Note that for excitations above singles- and doubles, this expression is independent of truncation level. Higher level excitations only affect the energy indirectly through their effect on the other amplitudes though the non-linear amplitude equations.

### 4.6.4 The Coupled-Cluster amplitude equations

The amplitude equations are found by projecting with excited determinants, $\langle\Phi_\mu| = \langle\Phi_0| X_\mu^\dagger$,

$$\langle\Phi_\mu|(H_N e^{\hat{T}})_C|\Phi_0\rangle = 0. \tag{4.124}$$

This gives exactly one equation for each unknown, considering that there is a one to one correspondance between unknowns in the cluster operator and excited determinants. As an example, for coupled-cluster singles doubles (CCSD), the amplitude equations for the singles amplitudes $\tau_i^a$ are given by

$$\langle\Phi_i^a|(H_N e^{\hat{T}})_C|\Phi_0\rangle = 0, \tag{4.125}$$

and for the doubles amplitudes $\tau_{ij}^{ab}$,

$$\left\langle\Phi_{ij}^{ab}\right|(H_N e^{\hat{T}})_C\left|\Phi_0\right\rangle = 0. \tag{4.126}$$

### 4.6.5 Variational Coupled-Cluster

For variational problems, the properties of a wavefunction are calculated using Hellmann-Feynman theorem, which states that observables of variational wavefunctions can be calculated from derivatives of the energy with respect to given parameters. However, the above formulation of coupled-cluster is not variational, and calculation of observables and properties of the coupled cluster wave function formulated as such is a challenge.

There have been made efforts to reformulate coupled-cluster by applying the variational principle to an energy functional in terms of the coupled cluster wavefunction[8, 14],

$$\Delta E_0 = \frac{\langle\Phi_0|e^{\hat{T}^\dagger} H_N e^{\hat{T}}|\Phi_0\rangle}{\langle\Phi_0|e^{\hat{T}^\dagger} e^{\hat{T}}|\Phi_0\rangle}. \tag{4.127}$$

However, expressions like this involve the adjoint of the Cluster operator $\hat{T}$, and neither the Baker-Campbell-Hausdorff-expansion of the numerator nor the series expansion of the denominator terminate, which leaves us with a number of terms equal to the FCI expansion.

Two solutions to this problem was developed separately by Helgaker and Jørgensen [20] and Arponen [4]. They are quite different approaches to the same problem, but end up with the same solution. The approach of Helgaker and Jørgensen is to formulate coupled-cluster in terms of a constrained optimization problem, where the method of Lagrange undetermined multipliers is used. The Lagrange multipliers are a set of parameters with corresponding equations ensure that the coupled cluster wavefunction is a critical point of the so-called Lagrangian. As we now have a critical point for the energy, we can apply Hellman-Feynman to calculate other observables.

However, as noted in [28], there is also another less known approach. It was developed by Arponen [4] and is based on the bivariational principle, which is a generalization of the variational principle where the left and right eigenfunctions are independent. It gives the same parameters and equations for the coupled cluster wavefunction, and is equivalent to the approach of [20]. In the next

sections, we first demonstrate the use of the bivariational principle to get the equations of coupled-cluster theory and its time-dependent counterpart, before applying it to the OATDCC energy functional and deriving the OATDCCD equations of motion.

### 4.6.6 Bivariational CC

We now consider a bivariational treatment of coupled-cluster, as first developed by Arponen[4]. As usual in coupled-cluster, we parametrize the many-body wavefunction in terms of excitations of the reference state $|\Phi\rangle$ given by the cluster operator. However, we use a separate parametrization for the bra-state, in terms of a separate reference state $\langle\tilde{\Phi}|$. In total we get

$$|\Psi\rangle = e^{\hat{T}} |\Phi\rangle \tag{4.128}$$

$$\langle\Psi'| = \langle\tilde{\Phi}| e^{\hat{T}'} \tag{4.129}$$

where the two reference states are required to be biortonormalized,

$$\langle\tilde{\Phi}|\Phi\rangle = 1 \tag{4.130}$$

Note that $\hat{T}'$ is a *de-excitation* operator, as it excites the bra-state. Inserting the CC parametrization into eq. (2.23) we obtain the bivariational energy functional of coupled-cluster,

$$\mathcal{E}_H\left[\hat{T}', \hat{T}\right] = \frac{\langle\Psi'|H|\Psi\rangle}{\langle\Psi'|\Psi\rangle} = \frac{\langle\tilde{\Phi}|e^{\hat{T}'} H e^{\hat{T}}|\Phi\rangle}{\langle\tilde{\Phi}|e^{\hat{T}'} e^{\hat{T}}|\Phi\rangle}. \tag{4.131}$$

However, to find a more tractable form, of these equations, We immediately perform a change of variables $(\hat{T}', \hat{T}) \to (\hat{\Lambda}, \hat{T})$, where $\hat{\Lambda}$ is another deexcitation operator $\hat{\Lambda} = \sum_\mu \lambda_\mu (X^\dagger)^\mu$ ($\mu$ represents an excitation of arbitrary level) such that we get a new bra-state

$$\langle\tilde{\Psi}| = \frac{\langle\Psi'|}{\langle\Psi'|\Psi\rangle} = \langle\tilde{\Phi}| (1 + \hat{\Lambda}) e^{-\hat{T}}, \tag{4.132}$$

or, solving for $\langle\Psi'|$,

$$\langle\Psi'| = \langle\Psi'|\Psi\rangle \langle\tilde{\Phi}| (1 + \hat{\Lambda}) e^{-\hat{T}}. \tag{4.133}$$

Note that in Arponens paper, the change of variables give the expression $e^S e^{-\hat{T}}$ instead of $(1 + \hat{\Lambda})e^{-\hat{T}}$. However, similarly to how the coupled-cluster corresponds FCI in the untruncated limit (with $e^{\hat{T}} = 1 + C$ including all possible excitations), there is the equality $e^S = 1 + \hat{\Lambda}$ in the untruncated case here. When truncating however, we get different theories depending on the parametrization used. Accordingly, a truncated $e^S$ corresponds to the Extended coupled-cluster-theory of Arponen, while $1 + \hat{\Lambda}$ corresponds to the variational coupled-cluster of Helgaker et al. [21].

Inserting eq. (4.133) into eq. (4.131), we get the coupled-cluster expectation functional,

$$\mathcal{E}_H\left[\hat{\Lambda}, \hat{T}\right] = \langle\tilde{\Phi}|(1 + \hat{\Lambda}) e^{-\hat{T}} H e^{\hat{T}}|\Phi\rangle. \tag{4.134}$$

This expression is very similar to the coupled-cluster Lagrangian developed by [21], and indeed it gives the exact same equations for $\tau$ and $\lambda$. However, in this case the $\lambda$-amplitudes are not considered Lagrangian multipliers, but they are instead part of the description of the left wavefunction $\langle\tilde{\Psi}| = \langle\tilde{\Phi}|\,(1+\hat{\Lambda})e^{-\hat{T}}$, constructed to make the right and left eigenvalues of the Hamiltonian equal, i.e.,

$$\langle\tilde{\Psi}|\,\hat{H} = E, \qquad\qquad \hat{H}\,|\Psi\rangle = E. \tag{4.135}$$

That is, $\langle\tilde{\Psi}|$ and $|\Psi\rangle$ are left and right eigenvectors with the same eigenvalue $E$.

### Bivariational Coupled-Cluster Equations

Performing the variations of the expectation functional eq. (4.134) with respect to the $\lambda_\mu$-amplitudes, we get the coupled-cluster $\tau$-equations. These correspond, and are equal, to the regular coupled-cluster equations,

$$\frac{\partial\mathcal{E}_H}{\partial\lambda_\mu} = \frac{\partial}{\partial\lambda_\mu}\langle\tilde{\Phi}|(1+\hat{\Lambda})e^{-\hat{T}}He^{\hat{T}}|\Phi\rangle = \langle\tilde{\Phi}_\mu|e^{-\hat{T}}He^{\hat{T}}|\Phi\rangle = 0. \tag{4.136}$$

This is expected, as the right hand eigenvalue problem is still the same. However, the derivative with respect to the $\tau$ amplitudes give us a new set of equations,

$$\begin{aligned}\frac{\partial\mathcal{E}_H}{\partial\tau_\mu} &= \frac{\partial}{\partial\tau_\mu}\langle\tilde{\Phi}|(1+\hat{\Lambda})e^{-\hat{T}}\hat{H}e^{\hat{T}}|\Phi\rangle \\ &= \langle\tilde{\Phi}|\Big(-\hat{X}_\mu e^{-\hat{T}}\hat{H}e^{\hat{T}} + e^{-\hat{T}}\hat{H}\hat{X}_\mu e^{\hat{T}}\Big)|\Phi\rangle \\ &= \langle\tilde{\Phi}|e^{-\hat{T}}\Big[H,\hat{X}_\mu\Big]e^{\hat{T}}|\Phi\rangle = 0.\end{aligned} \tag{4.137}$$

These are called the $\hat{\Lambda}$-equations, because solving them give us expressions for the $\lambda$-amplitudes. As mentioned above, the $\lambda$-amplitudes are needed in order to compute expectation values of other operators than the energy, such as the dipole moment[28], consistent with the Hellmann-Feynman theorem.

## 4.6.7 Time-dependent Coupled-Cluster

We can now use the bivariational treatment to find the equations of motion for the amplitudes of coupled-cluster. We start by inserting eqs. (4.128) and (4.133) into the action-like functional in eq. (2.24)[27],

$$\mathcal{S}\big[\hat{T}(\cdot),\hat{\Lambda}(\cdot)\big] = \int_0^T i\,\langle\Phi|(1+\hat{\Lambda}(t))e^{-\hat{T}(t)}\frac{\partial}{\partial t}e^{\hat{T}(t)}|\Phi\rangle - \mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t)\big]\,\mathrm{d}t, \tag{4.138}$$

where $\hat{T}(\cdot)$ and $\hat{\Lambda}(\cdot)$ indicate dependency on the whole time domain $t\in[0,T]$. The first term gives

$$\begin{aligned}\langle\Phi|(1+\hat{\Lambda}(t))e^{-\hat{T}(t)}\frac{\partial}{\partial t}e^{\hat{T}(t)}|\Phi\rangle &= \langle\Phi|(1+\hat{\Lambda}(t))e^{-\hat{T}(t)}\sum_\mu \hat{X}_\mu\dot{\tau}_\mu e^{\hat{T}(t)}|\Phi\rangle \\ &= \sum_\mu \lambda_\mu\dot{\tau}_\mu.\end{aligned} \tag{4.139}$$

The CC action then becomes

$$\mathcal{S}\big[\hat{T}(\cdot),\hat{\Lambda}(\cdot)\big] = \int_0^T i\sum_\mu \lambda_\mu \dot{\tau}_\mu - \mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t)\big]\,\mathrm{d}t. \tag{4.140}$$

Alternatively we can subtract a total derivative $\frac{\mathrm{d}}{\mathrm{d}t}(i\sum_\mu \lambda_\mu\tau_\mu)$, up to which the action is invariant, and get

$$\mathcal{S}\big[\hat{T}(\cdot),\hat{\Lambda}(\cdot)\big] = \int_0^T i\sum_\mu -\dot{\lambda}_\mu\tau_\mu - \mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t)\big]\,\mathrm{d}t. \tag{4.141}$$

From these two expressions we get the equation of motion for the $\tau$- and $\lambda$-amplitudes, by the vanishing first order variations with respect to $\lambda_\mu$ and $\tau_\mu$, respectively,

$$i\dot{\tau}_\mu = \frac{\partial}{\partial\lambda_\mu}\mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t)\big], \tag{4.142}$$

$$-i\dot{\lambda}_\mu = \frac{\partial}{\partial\tau_\mu}\mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t)\big]. \tag{4.143}$$

This is the schrödinger equation for the bivariational time-dependent coupled-cluster theory, identical to what is found by [38].

## 4.7 OATDCC

In 2012, Kvaal introduced the orbital adaptive time-dependent coupled-cluster theory[27], or OATDCC for short. It is based on the bivariational formulation of coupled-cluster, where there are separate parametrizations of $\hat{T}$ and $\hat{T}'$ associated with the ket- and bra-wavefunctions, respectively. However, as can be inferred from the name of the method, there is an additional flexibility in that the orbitals are allowed to adapt with time.

Starting from bivariational representation of the wavefunctions found in section 4.6.6, the OATDCC bivariational expecation value is

$$\mathcal{E}_H\left[\hat{\Lambda},\hat{T},\tilde{\Phi},\Phi\right] = \langle\tilde{\phi}|(1+\hat{\Lambda})e^{-\hat{T}}He^{\hat{T}}|\phi\rangle \tag{4.144}$$

Note that in addition to the dependency of the amplitudes through $\hat{T}$ and $\hat{\Lambda}$, there is also an implicit dependence on the set of orbitals $\Phi = (\varphi_1,\varphi_2,\ldots,\varphi_L)$ and their duals $\tilde{\Phi} = (\tilde{\varphi}_1;\tilde{\varphi}_2;\ldots;\tilde{\varphi}_L)$. We will refer to these orbitals as the OA orbitals. These two sets of orbitals are as independent of each other as possible, with the only restriction being the biortogonality of the dual vectors,

$$\langle\tilde{\varphi}_q|\varphi_p\rangle = \delta_{qp}. \tag{4.145}$$

Defining the basis of OA Slater determinants from the OA orbitals

$$\langle\tilde{\phi}_{p_1,\ldots,p_N}| = \langle\tilde{\varphi}_{p_1}\cdots\tilde{\varphi}_{p_N}|, \tag{4.146}$$

and

$$|\phi_{q_1,\ldots,q_N}\rangle = |\varphi_{p_1}\cdots\varphi_{p_N}\rangle, \tag{4.147}$$

with overlap

$$\left\langle \tilde{\phi}_{p_1,\ldots,p_N} \middle| \phi_{q_1,\ldots,q_N} \right\rangle = \delta_{p1,q1} \cdots \delta_{p1,q1}, \tag{4.148}$$

the constraint of biortogonality is equivalent to the statement that for every $|\Psi\rangle \in \mathcal{V}[\Phi]$ there exists a $\left\langle \tilde{\Psi} \right| \in \mathcal{V}'[\Phi']$ such that $\left\langle \tilde{\Psi} \middle| \Psi \right\rangle \neq 0$ (or equivalently with the roles of $|\Psi\rangle$ and $\left\langle \tilde{\Psi} \right|$ reversed)[27]. In other words,

The creation and annihilation operators of the orbitals are defined as

$$c_p^\dagger \equiv \int \mathrm{d}x\, \varphi_p(x)\psi(x)^\dagger, \tag{4.149}$$

$$\tilde{c}_p \equiv \int \mathrm{d}x\, \tilde{\varphi}_p(x)\psi(x), \tag{4.150}$$

where the field operators adhere to the normal anticommutation rules,

$$\left\{\psi(x)^\dagger, \psi(x')\right\} = \delta(x - x'), \tag{4.151}$$

and $\delta(x - x')$ is the Dirac delta. It is readily shown that these creation and annihilation operators fulfill anticommutation rules,

$$\begin{aligned}\left\{c_p^\dagger, \tilde{c}_q\right\} &= \iint \varphi_p(x)\tilde{\varphi}_q(x')\left\{\psi(x)^\dagger, \psi(x')\right\}\mathrm{d}x\,\mathrm{d}x' \\ &= \int \varphi_p(x)\tilde{\varphi}_q(x)\mathrm{d}x \equiv \langle\tilde{\varphi}_p|\varphi_q\rangle = \delta_{pq},\end{aligned}$$

by biortogonality. As such, Wicks theorem holds, which lets us consider the biortogonal orbitals just as we would normal orbitals with regards to derivations of coupled-cluster equations. The only difference is that we let $\langle\varphi_p| = (|\varphi_p\rangle)^\dagger \to \langle\tilde{\varphi}_p|$.

Seeing as Wicks theorem holds, we can define the one- and two-body density matrices as

$$\rho_p^q \equiv \left\langle \tilde{\phi} \right|(1 + \hat{\Lambda})e^{-\hat{T}}c_p^\dagger \tilde{c}_q \left|\phi\right\rangle e^{\hat{T}}, \tag{4.152}$$

$$\rho_{pr}^{ps} \equiv \left\langle \tilde{\phi} \right|(1 + \hat{\Lambda})e^{-\hat{T}}c_p^\dagger c_r^\dagger \tilde{c}_s \tilde{c}_q e^{\hat{T}} \left|\phi\right\rangle. \tag{4.153}$$

Consider an Hamiltonian (or any operator) containing only one- and two-body interactions,

$$\hat{H} = \sum_{pq} h_q^p c_q^\dagger \tilde{c}_q + \frac{1}{4}\sum_{pqrs} u_{rs,AS}^{pq} c_p^\dagger c_r^\dagger \tilde{c}_s \tilde{c}_q. \tag{4.154}$$

Inserting into the bivariational expectation value, we then get only terms coupled to the one- and two-body operators stated above,

$$\mathcal{E}_H\left[\hat{\Lambda}, \hat{T}, \tilde{\Phi}, \Phi\right] = \rho_p^q h_q^p + \frac{1}{4}\rho_{pr}^{qs} u_{qs,AS}^{pr}. \tag{4.155}$$

This is a very practical description, as the dependency of the amplitudes is completely captured in the density operators, while the orbital dependency is captured in the matrix elements. We can write this as

$$\rho_p^q = \rho_p^q(\lambda, \tau), \tag{4.156}$$

$$\rho_{pr}^{qs} = \rho_{pr}^{qs}(\lambda, \tau), \tag{4.157}$$

$$h_p^q = \rho_p^q(\tilde{\Phi}, \Phi), \tag{4.158}$$

$$u_{qs,AS}^{pr} = u_{qs,AS}^{pr}(\tilde{\Phi}, \Phi). \tag{4.159}$$

The corresponding action of OATDCC is also deceptively similar to the TDCC-action,

$$\mathcal{S}\big[\hat{T}(\cdot),\hat{\Lambda}(\cdot),\tilde{\Phi}(\cdot),\Phi(\cdot)\big] = \int_0^T i\hbar\,\big\langle\tilde{\phi}(t)\big|(1+\hat{\Lambda}(t))e^{-\hat{T}(t)}\frac{\partial}{\partial t}e^{\hat{T}(t)}\big|\phi(t)\big\rangle - \mathcal{E}_H\big[\hat{\Lambda}(t),\hat{T}(t),\tilde{\Phi}(t),\Phi(t)\big]\,\mathrm{d}t. \tag{4.160}$$

However, some of the expressions are more involved. We start with the time-derivative $\frac{\partial}{\partial t}e^{\hat{T}(t)}\,|\phi(t)\rangle = \frac{\partial}{\partial t}\,|\Psi(t)\rangle$, and introduce the projection operator

$$\Pi = \big|\phi\big\rangle\big\langle\tilde{\phi}\big| + \sum_\mu \big|\phi_\mu\big\rangle\big\langle\tilde{\phi}_\mu\big|\,, \tag{4.161}$$

which projects a dual state onto $\mathcal{V}'[\Phi']$ and a state onto $\mathcal{V}[\Phi]$. Any dual state $|\Psi\rangle$ or state $\big\langle\tilde{\Psi}\big|$ already in the respective subspace will be unaffected by this operator, and as such

$$\begin{aligned}|\Psi\rangle = \Pi\,|\Psi\rangle &= |\phi\rangle + \sum_\mu \big\langle\tilde{\phi}^\mu\big|e^{\hat{T}}\big|\phi\big\rangle\,|\phi_\mu\rangle \\ &\equiv |\phi\rangle + \sum_\mu A^\mu\,|\phi_\mu\rangle\,.\end{aligned} \tag{4.162}$$

Computing the derivative then gives

$$\frac{\partial}{\partial t}\,|\Psi(t)\rangle = \frac{\partial}{\partial t}\Pi\,|\Psi(t)\rangle = \frac{\partial}{\partial t}\,|\phi\rangle + \sum_\mu \dot{A}^\mu\,|\phi_\mu\rangle + \sum_\mu A^\mu\frac{\partial}{\partial t}\,|\phi_\mu\rangle\,. \tag{4.163}$$

For the first and third terms, we need the derivative of a Slater determinant,

$$\begin{aligned}\frac{\partial}{\partial t}\,|\phi\rangle &= \frac{\partial}{\partial t}c^\dagger_{p_1}c^\dagger_{p_2}\cdots c^\dagger_{p_N}\,|0\rangle \\ &= \left(\sum_q \dot{c}^\dagger_q\tilde{c}_q\right)c^\dagger_{p_1}c^\dagger_{p_2}\cdots c^\dagger_{p_N}\,|0\rangle \qquad (4.164)\\ &= Dc^\dagger_{p_1}c^\dagger_{p_2}\cdots c^\dagger_{p_N}\,|0\rangle \qquad (4.165)\end{aligned}$$

where the sum in the operator $D \equiv \sum_q \dot{c}^\dagger_q\tilde{c}_q$ is unconstrained. The second term can be calculated by the consideration that creation and annihilation operators in the definition of $A^\mu$ can be dealt with using Wicks theorem, leaving only a dependence on the amplitudes in $\hat{T}$, $A^\mu = A^\mu(\tau)$. Using the chain rule, we get

$$\frac{\partial}{\partial t}A^\mu(\tau) = \sum_\nu \dot{\tau}^\nu\frac{\partial}{\partial\tau^\nu}A^\mu(\tau). \tag{4.166}$$

Combining the above with a zero-term $0 = \sum_\mu \dot{\tau}_\mu\frac{\partial 1}{\partial\tau_\mu}\,|\phi\rangle$, we get

$$\frac{\partial}{\partial t}\,|\Psi(t)\rangle = D\,|\Psi\rangle + \sum_\mu \dot{\tau}_\mu\frac{\partial}{\partial\tau_\mu}\,|\Psi\rangle = \left(D + \sum_\mu \dot{\tau}_\mu X_\mu\right)|\Psi\rangle\,. \tag{4.167}$$

We recognize the second term from eq. (4.139). It is only dependent on the amplitudes, and results the same term as in time-dependent coupled-cluster

$\sum_\mu \lambda_\mu \dot{\tau}_\mu$. Superficially, the operator $D$ then represents the difference between the current theory and TDCC. In total, the first term of the OATDCC action becomes

$$\begin{aligned} i\hbar \left\langle \tilde{\phi} \right| (1+\hat{\Lambda}) e^{-\hat{T}} \frac{\partial}{\partial t} e^{\hat{T}} \left| \phi \right\rangle &= \sum_\mu \lambda_\mu \dot{\tau}_\mu + i\hbar \left\langle \tilde{\phi} \right| (1+\hat{\Lambda}) e^{-\hat{T}} D e^{\hat{T}} \left| \phi \right\rangle \\ &= \sum_\mu \lambda_\mu \dot{\tau}_\mu + i\hbar \left\langle \tilde{\phi} \right| (1+\hat{\Lambda}) e^{-\hat{T}} \Pi D \Pi e^{\hat{T}} \left| \phi \right\rangle \quad (4.168) \\ &= \sum_\mu \lambda_\mu \dot{\tau}_\mu - \mathcal{E}_{-i\hbar D_0} \left[ \hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t) \right], \end{aligned} \quad (4.169)$$

where we introduced the projected operator $D_0 = \Pi D \Pi$. The explicit form of this operator is

$$D_0 \equiv \sum_{p,q=1}^{L} \langle \tilde{\varphi}_p | \varphi_q \rangle \, c_p^\dagger \tilde{c}_q \equiv \sum_{pq} \eta_q^p c_p^\dagger \tilde{c}_q. \quad (4.170)$$

Note that the one-body matrix element $\eta_q^p$ is dependent on the orbitals and the time-derivative of the orbitals as $\eta_q^p = \eta_q^p(\tilde{\Phi}, \dot{\Phi})$. Exploiting the linearity of the bivaritational expectation functional $\mathcal{E}_A$ and inserting into eq. (4.160) gives

$$\mathcal{S}\left[T(\cdot), \hat{\Lambda}(\cdot), \tilde{\Phi}(\cdot), \Phi(\cdot)\right] = \int_0^T i\hbar \sum_\mu \lambda_\mu \dot{\tau}^\mu - \mathcal{E}_{H-i\hbar D_0} \left[ \hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t) \right] \mathrm{d}t \quad (4.171)$$

A more explicit expression of the expectation value functional is in terms of metrix elements according to section 4.3.4, giving

$$\mathcal{S}\left[T(\cdot), \hat{\Lambda}(\cdot), \tilde{\Phi}(\cdot), \Phi(\cdot)\right] = \int_0^T i\hbar \sum_\mu \lambda_\mu \dot{\tau}^\mu \rho_p^q (h_q^p - i\eta_q^p) + \frac{1}{4} \rho_{pr}^{qs} u_q^{pr} s \, \mathrm{d}t. \quad (4.172)$$

Here we have defined the one-body and two-body density matrices,

$$\rho_p^q \equiv \left\langle \tilde{\Phi}_0 \right| (1+\Lambda) e^{-\hat{T}} c_p^\dagger \tilde{c}_q e^T \left| \Phi_0 \right\rangle, \quad (4.173)$$

$$\rho_{pq}^{rs} \equiv \left\langle \tilde{\Phi}_0 \right| (1+\Lambda) e^{-\hat{T}} c_p^\dagger c_q^\dagger \tilde{c}_s \tilde{c}_r e^T \left| \Phi_0 \right\rangle. \quad (4.174)$$

These depend only on the amplitudes, as the elements are evaluated using Wick's theorem. Further we have the one-body and anti-symmetrized two-body matrix elements of the Hamiltonian, and the one-body matrix elements of the the time-dependent,

$$h_q^p \equiv \left\langle \tilde{\phi}_p \right| \hat{h} \left| \phi_q \right\rangle, \quad (4.175)$$

$$u_{rs}^{pq} \equiv \left\langle \tilde{\phi}_p \tilde{\phi}_q \right| \hat{u} \left| \phi_r \phi_s \right\rangle_{AS}, \quad (4.176)$$

$$\eta_q^p \equiv \left\langle \tilde{\phi}_p \right| \frac{\partial}{\partial t} \left| \phi_q \right\rangle. \quad (4.177)$$

These do not depend on the amplitudes, but instead vary over time due to the dependence of the orbitals, like in TDHF.

### 4.7.1 Gauge freedom

As noted by Kvaal [27], there is a gauge freedom in the representation of the wavefunction. All of the $\tau_\mu$, $\lambda_\mu$ and the orbitals of $\Phi$ and $\tilde{\Phi}$ are free parameters, which makes it overdetermined. We can therefore find useful gauge-conditions to simplify calculations. Similarly to NOCC and MCTDHF, we perform orbital rotations on the wavefunction, such that singles excitations are included in the wavefunction. The singles cluster operator is not necessary and we set $\tau_\mu = \lambda_\mu = 0$, giving us OATDCCD as the simplest method of OATDCC. Due to the orbital rotations, OATDCCD includes singles accuracy as well as doubles accuracy.

In addition to this, we have the freedom to let the rotation of the orbitals over time be such that time-dependent rotations are zero between occupied-occupied and virtual-virtual orbitals,

$$\eta_j^i = \eta_b^a = 0, \tag{4.178}$$

This decouples the Q-space from the P-space equations, and allow us to write the OATDCC orbital equations in a simpler form.

### 4.7.2 OATDCCD

We will be working with the simplest scheme of OATDCC, in which the amplitudes are truncated at the doubles level,

$$\tau_\mu = \tau_{ij}^{ab} \qquad\qquad \lambda^\mu = \lambda_{ab}^{ij}.$$

This is the doubles scheme, giving OATDCCD. Varying the action with respect to the amplitudes gives an expression very similar to that of eqs. (4.142) and (4.143) for the doubles amplitudes,

$$i\hbar\dot{\tau}_{ij}^{ab} = \frac{\partial}{\partial\lambda_{ab}^{ij}}\mathcal{E}_{H-i\hbar D_0}\big[\hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t)\big], \tag{4.179}$$

$$-i\hbar\dot{\lambda}_{ab}^{ij} = \frac{\partial}{\partial\tau_{ij}^{ab}}\mathcal{E}_{H-i\hbar D_0}\big[\hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t)\big], \tag{4.180}$$

The difference is a one-body term $-i\hbar D_0 = -i\hbar\sum_{pq}\eta_q^p c_p^\dagger \tilde{c}_q$ added to the Hamiltonian. This term couples to the one-body operator according to eq. (4.155). In CCD, the matrix elements of the one-body density operator are block diagonal, with $\rho_a^i = \rho_i^a = 0$. Given that $\eta_b^a = \eta_j^i = 0$, the terms $\rho_p^q\eta_q^p$ disappears, and there is no contribution from $D_0$ to the action. Consequently, the equations of motion of the amplitudes and the orbitals separate. The amplitude equations become equal to the TDCCD amplitude equations,

$$i\hbar\dot{\tau}_{ij}^{ab} = \frac{\partial}{\partial\lambda_{ab}^{ij}}\mathcal{E}_H\big[\hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t)\big], \tag{4.181}$$

$$-i\hbar\dot{\lambda}_{ab}^{ij} = \frac{\partial}{\partial\tau_{ij}^{ab}}\mathcal{E}_H\big[\hat{\Lambda}(t), \hat{T}(t), \tilde{\Phi}(t), \Phi(t)\big]. \tag{4.182}$$

The orbital equations are those that govern the time-evolution of the orbitals. We define the projection operator onto the subspace defined by the OA orbitals,

$$P = \sum_p \big|\phi_p\big\rangle\big\langle\tilde{\phi}_p\big|. \tag{4.183}$$

The projection operator onto the remaining parts of the single-particle Hilbert space is denoted $Q = 1 - P$. With the chosen gauge conditions, it is only necessary to consider variations of the action which are governed by orbitals in the two orbital subsets separately. The orbital equations thus split into P-space equations and Q-space equations.

The P-space equations can be said to govern the matrix elements $\eta_q^p$, and are a set of equations given in [27] as

$$i\hbar A_{aj}^{ib}\eta_b^j = h_p^a\rho_p^i - h_q^i\rho_a^q + \frac{1}{2}u_{os}^{pq}\rho_{pq}^{as} - \frac{1}{2}u_{rs}^{aq}\rho_{iq}^{sr}, \tag{4.184}$$

for the occupied-virtual block of the matrix elements of $\hat{\eta}$ in our basis, and

$$-i\hbar\eta_j^b A_{bi}^{ja} = h_i^p\rho_p^a - h_q^a\rho_i^q + \frac{1}{2}u_{is}^{pq}\rho_{pq}^{sa} - \frac{1}{2}u_{rs}^{aq}\rho_{iq}^{sr} + i\hbar\partial_t\rho_i^a, \tag{4.185}$$

for the virtual-occupied block. These equations are then solved for $\eta_a^i$ and $\eta_i^a$, where all other elements are defined to be zero in the chosen gauge.

The Q-space equations are given by

$$i\hbar\rho_p^q\hat{Q}\partial_t\left|\phi_q\right\rangle = \rho_p^q\hat{Q}\hat{h}\left|\phi_q\right\rangle + \rho_{pq}^{rs}\hat{Q}\hat{W}_s^q\left|\phi_r\right\rangle, \tag{4.186}$$

for the ket-part, and

$$-i\hbar\rho_p^q\left(\partial_t\left\langle\tilde{\phi}_p\right|\right) = \rho_p^q\left\langle\tilde{\phi}_p\right|\hat{h}\hat{Q} + \rho_{pq}^{rs}\left\langle\tilde{\phi}_p\right|\hat{W}_s^q\hat{Q}, \tag{4.187}$$

for the bra-part. Here we have defined the mean-field operator

$$\hat{W}_s^q = \int \mathrm{d}x_1\tilde{\phi}_q(x_1)\hat{u}(x_1,x_2)\phi_s(x_1) > \tag{4.188}$$

We will expand the OA orbitals $\{\langle\phi_p|, |\phi_p\rangle\}_{p=1}^K$ in terms of a possibly larger computational basis $\{\langle\tilde{\chi}_\alpha|, |\chi_\alpha\rangle\}_{\alpha=1}^L$, given by $\tilde{C}_\alpha^p$ and $C_p^\alpha$ for the bra and ket transformations, respectively. In appendix A we derive expressions for the time-evolution of these coefficients. The ket-parts are given by

$$i\hbar\dot{C}_q^\alpha = i\hbar\eta_q^{p'}C_{p'}^\alpha + C_q^\beta h_\beta^\alpha - h_q^{p'}C_{p'}^\alpha + (\rho^{-1})_q^p\rho_{pr}^{q's}\left(C_{q'}^\beta W_{\beta s}^{\alpha r} - u_{sq'}^{p'r}C_{p'}^\alpha\right), \tag{4.189}$$

The corresponding bra-equation is

$$-i\hbar\dot{\tilde{C}}_\alpha^p = i\hbar\eta_{p'}^p\tilde{C}_\alpha^{p'} + \tilde{C}_\beta^p h_\alpha^\beta - h_{p'}^p\tilde{C}_\alpha^{p'} + (\rho^{-1})_q^p\rho_{p'r}^{qs}(\tilde{C}_\beta^{p'}W_{\alpha s}^{\beta r} - u_{q's}^{p'r}\tilde{C}_\alpha^{q'}). \tag{4.190}$$

If $P$ spans the entire Hilbert space, which when we use a basis expansion means that the $C$-matrices are square and unitary, then we say that the Q space is zero, and the orbital equations reduce to just the first term given by the P-space equations,

$$-i\hbar\dot{\tilde{C}}_\alpha^p = i\hbar\eta_{p'}^p\tilde{C}_\alpha^{p'}, \tag{4.191}$$

$$i\hbar\dot{C}_q^\alpha = i\hbar\eta_q^{p'}C_{p'}^\alpha. \tag{4.192}$$

# CHAPTER 5

# Implementation

The process of this master thesis in Computational Science has been threefold, and is reflected in the structure of the thesis; in general terms it involved delving into a theoretical framework, translating the theory into computer code and finally using the computer code to study physical systems. We have now laid the theoretical foundation in the previous chapters. The present chapter is focused on the details of implementing solvers for the aforementioned equations on a computer. The ultimate goal is of ionization using our own implementation of a novel combination of a DVR basis set and the OATDCC method, and this chapter introduces the buiding blocks needed to get to this point.

This chapter is divided into several sections; the first is an overview of the two main libraries involved in the thesis, including the `QuantumSystems` and `CoupledCluster` libraries developed by Schøyen, Winther-Larsen and Kristiansen [26, 41, 51]. A large part of the work on this thesis has been contributing to these two libraries, and we will here highlight the changes we have made.

The next section concerns the specifics of solving the time-independent Schrödinger equation (TISE) numerically. The hierarchy of methods from finding a single particle basis set, via Hartree-Fock (HF) to coupled-cluster (CC) ground state solvers are presented. The different methods are illustrated with `Python` implementations, which are benchmarked against known results.

The third section presents numerical solutions of the time-dependent Schrödinger equation (TDSE), with implementation details for the problems relevant to this thesis. We have used two approaches to solve the TDSE. The first approach is to approximate the unitary time propagation operator $U(t_1, t_0)$ described in section 2.3.4. This lets us evolve a state from time $t_0$ to time $t_1$,

$$|\Phi(t_1)\rangle = U(t_1, t_0)\,|\Phi(t_0)\rangle\,. \tag{5.1}$$

We use this approach to propagate independent particle methods, both for single particles on a grid and int time-dependent Hartree-Fock theory. The second approach described in this section is to solve the TDSE as an ordinary differential equation (ODE) on the form

$$\dot{y} = f(y, t), \tag{5.2}$$

where $y(t)$ is the time-dependent parameters of the ansatz used. This is used for the many-body action-based methods of time-Dependent coupled-cluster (TDCC) and orbital adaptive time-dependent coupled-cluster (OATDCC).

Finally there are some topics that does not naturally fit under the main sections, but are still very important. These topics are discussed in the final section called "special considerations". This includes topics such as imaginary time propagation, adiabatic switching, regularization, etc.

## 5.1 Overview

The main results of this thesis builds on the work done by previous students at the department. The first master thesis on the topic of Time-dependent coupled-cluster was the one of Kristiansen [26]. He was succeded by Winther-Larsen [51] and Schøyen [41], who together with Kristiansen developed a substantial code base in the form of two repositories, `coupled-cluster` and `quantum_systems`. This section present a brief overview of these two libraries, enough to understand the basic flow of a program that use them. Following this is a summary of the changes introduced as of late. These changes are improvements to the structure and new features, and the focus will be on the changes that facilitate a sparse DVR basis to be utilized in time-dependent coupled-cluster calculations.

Installation procedures for the repositories can be found in the documentation of the packages. The use of a virtual environment such as conda or virtualenv is highly recommended, as the libraries have some specific requirements when it comes to dependency versions.

### 5.1.1 The Quantum Systems Library

The `Python` library `quantum-systems`[1] contains the machinery to calculate and represent the basic elements of a quantum system. Currently, its main components are the class `BasisSet` (from now on the basis-set) with derivatives and the abstract class `QuantumSystem` (from now on the system) with derivatives. They have an aggregate relationship, where the constructor of a system object requires the input of a basis-set such that the methods and attributes of the former use the methods and attributes of the latter, but not vice versa.

While the basis-set contains the matrix elements of the single-particle hamiltonian, the basis functions and their domain and so on, it has no notion of occupancy. The basis-set can be viewed as a single-particle instance. The system, on the other hand, stores both the number of particles present and any external interactions. It also has a more explicit notion of anti-symmetrization and inclusion of spin than the basis-set. We now go into more detail on the two.

#### BasisSet

As stated above, an instance of `BasisSet` defines the single-particle problem underlying the system at hand. It stores the matrix elements of the hamiltonian operator $\hat{h}$ and the Coulomb operator $\hat{u}$ upon initialization, in addition to the

[1] https://github.com/schoyen/quantum-systems

overlap matrix $s_{ij} = \langle \phi_i | \phi_j \rangle$ and other quantities relevant to the single-particle basis. It contains methods to operate on the matrix elements with respect to the symmetries of the system, and to change the basis through a discrete transformation,

$$|\phi_p\rangle = C_p^\alpha \, |\chi_\alpha\rangle \tag{5.3}$$

Figure 5.1 contains an overview of the `BasisSet` class and its subclasses. A `BasisSet` object can be created directly and then supplied with matrix elements manually, but the preferred method is to use an existing or implement a subclass. In this case the matrix elements are produced on instantation.

There are currently four types of subclasses of `BasisSet` implemented:

1. One-dimensional quantum dots, which uses numerical integration (or analytical eigenfunctions in the case of ODHO) on a 1d grid to find eigenfunctions for arbitrary potentials in one dimension.

2. Two-dimensional quantum dots, which implements the numerical expression of [3] to find Coulomb integrals for the eigenfunctions of the two-dimensional harmonic oscillator potential. These basis sets have yet to be updated to the new formalism of Quantum Systems, and won't work at the moment.

3. Sinc-DVR, our implementation, which calculates and represents sinc-DVR matrix elements, with options for a sparse (2d) representation of the Coulomb matrix elements.

4. Random basis set, which provides a random matrix elements with correct symmetries, for testing purposes.

As an example, we can construct a one-dimensional quantum dot with a 1d harmonic oscillator potential and access its matrix elements in the following way,

```
from quantum_systems import ODQD

basis_set = ODQD(
    l=10,
    num_grid_points=201,
    grid_length=10,
    potential = ODQD.HOPotential(omega=0.25)
)
print(basis_set.h)
print(basis_set.u)
```

The arguments here are the number of basis functions $l$, the number of grid points and grid length for numerical evaluation of the Hamiltonian as described in section 3.1, and the potential energy operator, which has to be represented as a callable `Python` object. Since $l = 10$ in this example, the dimensions of the matrix elements are $\mathbb{R}^{10\times10}$ and $\mathbb{R}^{10\times10\times10\times10}$ for $h$ and $u$ respectively.

For a comprehensive list of the methods and attributes of a `BasisSet`, one can check the documentation.

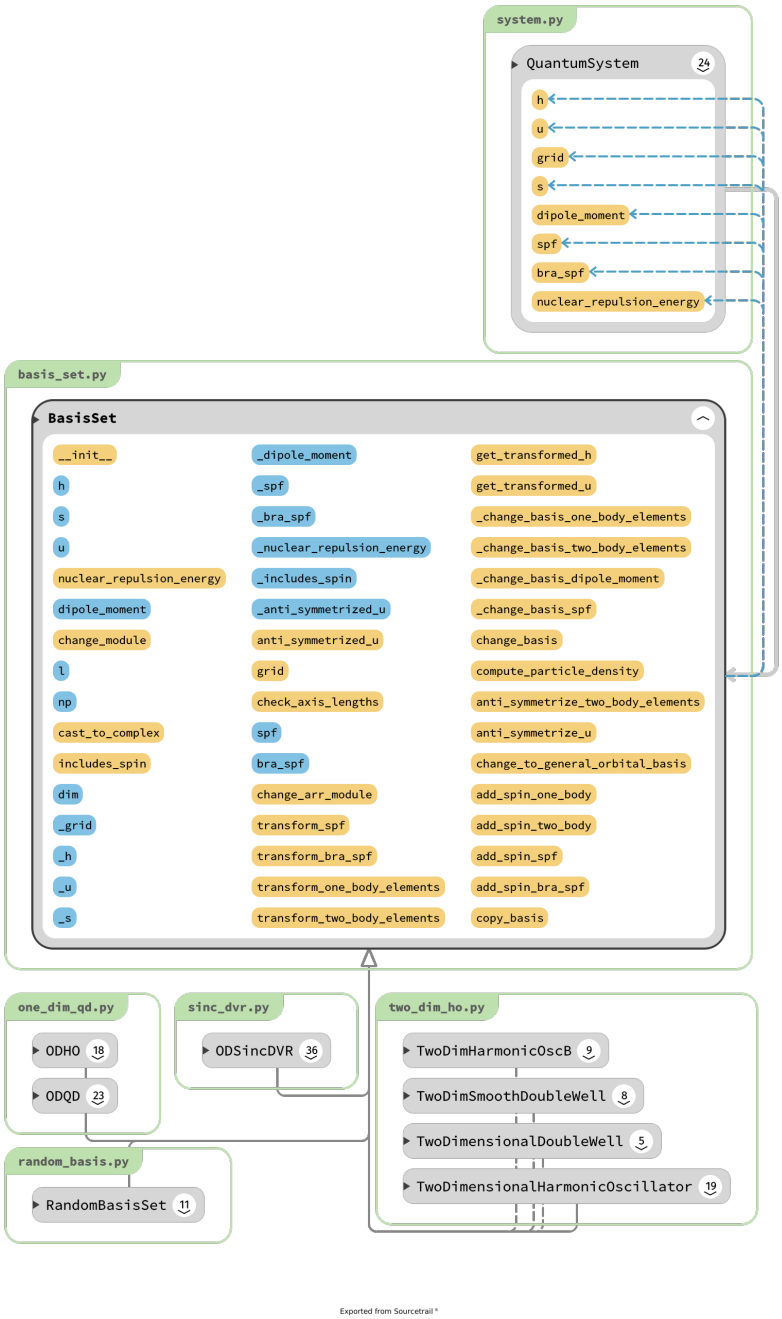



Figure 5.1: Snapshot of the abstract class basis set, its subclasses and its relation to the quantum system class at the moment of writing. Thin gray lines represent inheritance, with dashed representing multi-level inheritance. Thicker gray lines represent aggregation, such as between `QuantumSystem` and `BasisSet`. Blue dashed lines are labeled with ambiguous use by the SourceTrail software, and is due to `QuantumSystem` having properties referencing the corresponding properties of `BasisSet`.

### QuantumSystem

Whereas a `BasisSet` represents a single-particle system, an instance of a derived class of `QuantumSystem` (which we will call a *system* from now on) represents the entire many-body system. It includes the number of particles, and the external influences such as the electromagnetic interactions.

There are currently two subclasses of quantum-system implemented, the `GeneralOrbitalSystem` and the `SpatialOrbitalSystem`. The distinction between the two is that while the former has spin included for an explicit treatment of spin, the latter has the resctriction that all orbitals are double occupied, one for each spin. We demonstrate the generation of a `SpatialOrbitalSystem` (SOS) from a spin-independent basis set,

```
from quantum_systems import SpatialOrbitalSystem

n_electrons = 2
sos = SpatialOrbitalSystem(n_electrons, basis_set)
print(sos.l)
```

The print statement here should produce the same $l$ as the basis set was generated with. To create a `GeneralOrbitalSystem` (GOS), we can either use the method `construct_general_orbital_system` of the SOS (this creates a copy of the `BasisSet` and its matrix elements before spin doubling), or directly generate a GOS (this changes the `BasisSet` when spin doubling),

```
from quantum_systems import GeneralOrbitalSystem

# either this:
gos = sos.construct_general_orbital_system()
# or this:
gos = GeneralOrbitalSystem(n_electrons, basis_set)
print(gos.l)
```

In any case, $l$ will now be the double of what it was in the case of the SOS. This is because when a GOS is initialized with a spin-independent basis-set, each basis-function is assumed to be a spatial orbital that can be doubly occupied. As such, it doubles the number of orbitals, and effectively quadruples the size of the matrix elements (or times 16 in the case of the Coulomb matrix elements $u^{\alpha\beta}_{\gamma\delta}$).

In the code snippet above the `gos` object now holds a reference to the basis set, which is mutable. As such, the basis set should not be used outside the `system`, as that might lead to unintended consequences. An example is the following method

```
gos.change_basis(C)
```

This changes the matrix elements of the basis-set in place, such that for example the one-body operator becomes

$$h^{\alpha}_{\beta} \rightarrow h^{p}_{q} = (C^{\dagger})^{p}_{\alpha} C^{\beta}_{q} h^{\alpha}_{\beta}. \tag{5.4}$$

To prepare a time-dependent calculation, we can set up the quantum system with an external electromagnetic interaction represented as a callable `Python`

object, which is then equipped to the `QuantumSystem` using the time-evolution operator `LaserField`,

```
from quantum_systems.time_evolution_operators import LaserField
import numpy as np

F0 = 0.04
laser = lambda t: F0*np.sin(omega*t)
operator = LaserField(laser)
gos.set_time_evolution_operator(operator)
```

### 5.1.2 The Coupled-Cluster Library

We have covered how we can use the Quantum System library to set up a system of several particles, and the main usage of such a system is in a many-body solver. The solvers we will cover are part of the `coupled-cluster` repository[2], where both ground state solvers and time-dependent solvers are implemented. The solvers are implemented in two hierarchies, the first which are the ground state solvers implemented by abstract base class `CoupledCluster` with subclasses, and the time-dependent solvers of the abstract base class `TimeDependentCoupledCluster` with subclasses. An overview of the currently implemented subclasses can be seen fig. 5.2 and fig. 5.3, respectively, and include different symmetries and truncation levels of CC as well as orbital adaptive versions.

#### Coupled-Cluster Ground State Solvers

The ground state solvers are dependent on the anti-symmetrized matrix elements, and a more indepth explanation of the details of a ground state coupled-cluster calculation is given later in section 5.2.3. To use the coupled-cluster library, a fully set up system with matrix elements is required. We can use the ground state solver to find the coupled-cluster ground state of the harmonic oscillator potential using the `GeneralOrbitalSystem` that we set up in the previous section,

```
from coupled_cluster import CCD

ccd_solver = CCD(gos, verbose=True)
ccd_solver.compute_ground_state()
print(ccd_solver.compute_energy().real)
```

Here we have used the `CCD` class, which is the ground state solver implementation of the coupled-cluster doubles truncation level, to calculate the value of the coupled-cluster doubles energy for the given system. Using the HO basis and `GeneralOrbitalSystem` as shown above, the CCD energy is calculated to be 1.052 (alternatively, if the `change_to_HF_basis` method was called, the value is 0.838).

The amplitudes of the ground state can be retrieved with the `get_amplitudes` method. This constructs and returns an `AmplitudeContainer`, which are

[2] https://github.com/schoyen/coupled-cluster. Note that this repository is hidden. A copy of the code can be sent on request.

objects that store the amplitudes for easy unpacking in different methods. An example on how to extract the amplitudes from such a container is the following,

```
amplitudes = ccd_solver.get_amplitudes(get_t_0=False)
t, l = amplitudes
```

Here `amplitudes` is an instance of the `AmplitudeContainer` class, and $t$ and $l$ are lists of matrices for the different truncation levels. One can also include the parameter $t_0$, which is a global phase.

For descriptions of the methods and attributes of the `AmplitudeContainer` class, see the master theses of Schøyen and Winther-Larsen [41, 51]. In addition to the `AmplitudeContainer` class for the coupled-cluster methods, there is also a subclass called `OACCVector` for the OATDCC-methods. Whereas the former only stores the $\lambda$ and $\tau$ amplitudes for coupled-cluster at the given levels of excitation, the latter also stores the coefficients for the basis change to the OA-basis.

The `coupled-cluster` library also contains a ground state solver for orbital adaptive coupled-cluster doubles in general and spin-restricted versions (OACCD and ROACCD). Note, though, that the OACCD solver is in reality a Non-Orthogonal CCD (NOCCD) solver [37], and does not have support for a non-zero Q-space. As such it is not useful in the context of using a DVR basis, but it will prove an important benchmark for our ground state methods.

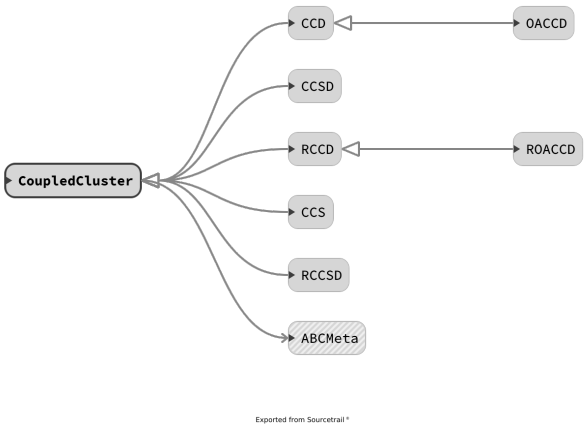



Figure 5.2: Inheritance of coupled-cluster classes. The abstract class `CoupledCluster` can not be instantiated. Instead, the subclasses are instantiated. To solve the corresponding amplitude equations (and, in the case of OA, orbital equations), and find the ground state of the included `QuantumSystem`, the method `compute_ground_state` is called.

### Time-dependent Coupled-Cluster

We have adapted the time-dependent methods of the `coupled-cluster` library to be on the same form, where the time-derivative of the parameters $y(t)$ are in

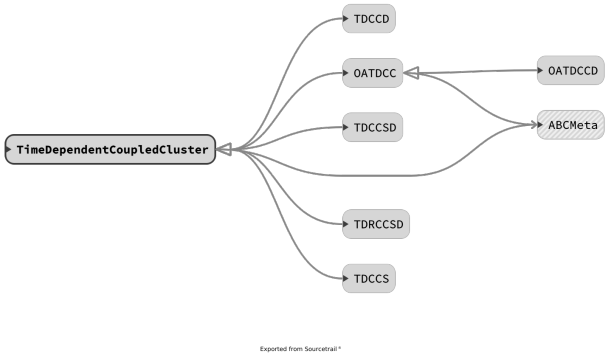



Figure 5.3: Inheritance of time-dependent coupled-cluster classes. The abstract class `TimeDependentCoupledCluster` can not be instantiated. Instead, the subclasses are instantiated. To solve the corresponding time-dependent equations, the class is used as a callable for an ode-solver such as those found in `scipy.integrate`.

terms of some vector evaluated function $f(t, y)$,

$$\dot{y} = f(t, y). \tag{5.5}$$

There are sofisticated numerical libaries for solving the above equation, which is the standard form for an ODE. As such, with our changes we can now use the ODE solvers of the `Python` library `scipy` to integrate the equations of motion of coupled-cluster. The different implementation of TDCC subclasses in the current version of the coupled-cluster library therefore all have a `__call__` method which implements the corresponding $f(t), y)$. In `Python`, any class with a `__call__`-method makes the instances of this class callable, which allows us to pass an instance of the class on to ode-wrappers such as `scipy.integrate.complex_ode` for solving.

The code snippet for time-dependent calculation of TDCC are a bit more involved than for the ground state calculations. This is natural as the set of possible numerical time-dependent experiments is much larger than the different ways of using a specialized ground state solver. Like the CC objects, the TDCC objects also require a reference to a `QuantumSystem` object, and this should be the same system if the TDCC simulation is based on a CC ground state. In the following snippets, we assume that the all the snippets presented previously in this chapter has been run, such that we now have a system `gos` for a harmonic oscillator hamiltonian with two particles in a Hartree-Fock basis, that is also equipped with an sinusoidal electromagnetic interaction at the oscillator frequency.

We also need the initial amplitudes in a raveled array, which is one where all the values are stored in a single one-dimensional array $y$. We here use the precalculated amplitudes from the CCD ground state solver to get the initial amplitudes on this form

```
    y0 = ccd_solver.get_amplitudes(get_t_0=True).asarray()
```

A corollary of the harmonic potential theorem[12] states that for a many-body harmonic oscillator system with an interaction at the oscillator frequency, the state should exhibit rigid translations. We test this for TDCCD with the following small script

```
from coupled_cluster import TDCCD
from gauss_integrator import GaussIntegrator
from scipy.integrate import complex_ode

tdcc = TDCCD(gos)

r = complex_ode(tdcc).set_integrator("GaussIntegrator")
r.set_initial_value(y0)

t_final = 2*np.pi/omega
dt = 1e-1
num_steps = int(t_final / dt) + 1
time_points = np.linspace(0, t_final, num_steps)

energy = np.zeros(num_steps, dtype=np.complex128)
energy[0] = tdcc.compute_energy(r.t, r.y)

particle_density = np.zeros((num_steps, num_grid_points))
particle_density[0] = tdcc.compute_particle_density(r.t, r.y)

for i, t in enumerate(time_points[1:]):
    particle_density[i+1] = tdcc.compute_particle_density(r.t, r.y)
    energy[i+1] = tdcc.compute_energy(r.t, r.y)
    r.integrate(t)
```

Note that we save the particle-density of the state at each time-step. In fig. 5.4 we plot the time evolution of the particle density, and see that there is almost no dynamics for this system using TDCCD. On the other hand, performing the same calculations with TDCCSD instead (just setting CCD to CCSD in all the above snippets), we see that qualitatively the expected rigid translations are present. This difference between the two methods is attributed to the fact that the electromagnetic field we used is a one-body operator, which will not affect the CCD state directly due to the lack of one-body excitation operators in the wavefunction.

### Changes to TDCC

Previous versions of the coupled-cluster library did not use `scipy` for numerical calculation, but instead manual implementations of ODE solvers such as the Runge Kutta and Gauss integrators. They where designed to accept instances of the `AmplitudeContainer` and `OACCVector` classes, as this was what the classes in coupled-cluster used to represent the amplitudes and coefficients.

However, `scipy` has a suite of fully implemented and tested ODE solvers for problems on the form of eq. (5.5), such as high orders of RungeKutta with adaptive step size[19], or the Variable-Coefficient ODE (VODE) methods[6]. An overview of the available integrators can be found at the scipy website[3].

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html

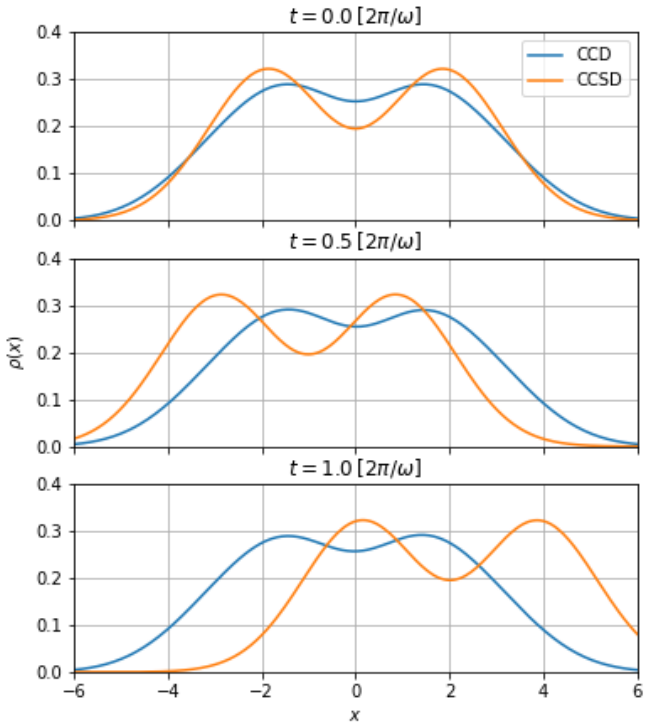



Figure 5.4: Simple numerical experiment of rigid translation of a harmonic oscillator of frequency $\omega = 0.25$ a.u. with two particles, using TDCCD and TDCCSD. The former fails to properly include the effects of the electromagnetic interaction, which is a sinusoidal laser of frequency $\omega$ and strength $F_0 = 0.04$.

Integrators not available in scipy, can be readily adapted into the same framework and used with the ode wrappers of scipy such as `scipy.integrate.ode` or . This has been done for the `GaussIntegrator`, which is described later in the section on integration **??**. .

To utilize the power of the integrators of `scipy`, we changed the time-dependent methods of the `coupled_cluster` library such that we could use the integrators of `scipy`.

The coupled-cluster libraries used the `AmplitudeContainer` and `OACCVector` objects to store and propagate the amplitudes and coefficients. This was changed, as the solvers of `scipy` expects and returns the a single raveled vector $y(t)$, where all the values are stored in a single one-dimensional array. This is according to eq. (5.5). In the case of coupled-cluster, the vector $y(t)$ eq. (5.5) contains all the different sized amplitudes $\lambda_\mu(t)$ and $\tau_\mu(t)$, with $\mu$ representing an arbitrary excitation of any truncation level. In the case of orbital adaptive coupled cluster, the coefficients of $C$ and $\tilde{C}$ are included as well.

We also removed the stateful `amplitude` attributes of the coupled-cluster classes, which were instances of `AmplitudeContainer`. Instead we use a template object called the `amp_template` to convert between the amplitudes and Numpy arrays. It is autogenerated based on the sizes of the system and the truncation level. Conversion back and forth between the two views of the amplitudes (and possibly coefficients) is done in the following way for `TDCC`,

```
amplitudes = tdcc.amp_template.fromarray(y) # returns an AmplitudeContainer
t, l = amplitudes
y = amplitudes.asarray()
```

and correspondingly for `OATDCC`,

```
amplitudes = oatdcc.amp_template.fromarray(y) # returns an OACCVector
t, l, C, C_tilde = amplitudes
y = amplitudes.asarray()
```

The ordering and size of the input array `y` is given uniquely by the AmplitudeContainer (or OACCVector) constructor. As an example, the size of the amplitudes vector for CCD is

$$\text{size}(y) = 1 + 2m^2n^2,$$

as it contains both the phase $t_0$, and the two sets of doubles amplitudes of $\tau_{ij}^{ab}$ and $\lambda_{ab}^{ij}$.

In the new `scipy`-friendly convention, the signature of all methods of the time-dependent solvers are also unified to accept the raveled array of amplitudes instead of using the internal state of the ground state solver. As an example, consider a method which all TDCC (and OATDCC) subclasses implements, `compute_energy`, with the following signature

```
tdcc.compute_energy(t, y)
```

### OATDCC

A strength of the OATDCC method is the fact that there are two separate bases involved in the parametrization of the wavefunction. One is static, and is typically chosen such that matrix elements of the one- and two-body hamiltonians can be readily pre-calculated or evaluated on the fly. The other basis covers a subset of the first and is used to construct the Slater determinants of coupled-cluster. This subspace is governed by the P-space equations, while the other is correspondingly governed by the Q-space equations.

The Q-space equations have been implemented as part of this thesis. Without them, OATDCC can still work as long as the Q-space is zero, i.e. the size of the two basis sets are equal. However, this removes some of the computational advantage of OATDCC; using different sized basis sets lets us cover a large spatial domain with a large number of basis functions, while simultaneously keeping the number of coupled-cluster determinants low.

Without Q-space equations, the OATDCC method is very similar in usage to the TDCC methods, except that there is the implicit change of

basis through the coefficient matrices. Example code can be found in `notebooks/TDCC_Harmonic_Oscillator.ipynb`.

On our branch in the coupled-cluster repository[4], we have implemented the necessary changes to OATDCC, enabling the use of the Q-space equations. See  for the details of our implemention of the OATDCC equation of motion.

To implement the Q-space equations, we had to make some changes to the base class of OATDCC in our branch given above. Most important is the fact that the equations now expect the non-anti-symmetrized matrix elements for the coulomb operator. The consequence of this is that the GOS that is given to the `OATDCC`-subclasses upon instantiation has to be set up with the parameter `anti_symmetrize` set to false. The OATDCC-based classes also expect the initial coefficients $C$ and $\tilde{C}$ upon instantiation(the latter can be dropped, upon which $\tilde{C} = C^\dagger$ is used). The size of the second axis of $C$ decide the size of the OA basis.

[4] https://github.com/Schoyen/coupled-cluster/tree/fix_q_space

### 5.1.3 Sinc-DVR basis set

Implementation and utilization of the Sinc-DVR basis set is a central part of this thesis. It is implemented as a subclass of the `BasisSet` in `quantum_systems`[5], called `ODSincDVR`, as can be seen in fig. 5.1.

When an object `ODSincDVR` is created, the one- and two-body hamiltonians $h_{\alpha\beta}$ and $u^{\alpha\beta}_{\gamma\delta}$ are generated in the following way. Following section 3.3, the former is given as the Toeplitz and Hermitian matrix,

$$h_{\alpha\beta} = T_{\alpha\beta} + V_{\alpha\beta}. \tag{5.6}$$

Here, the kinetic energy elements are given by eq. (A.36) as

$$T_{\alpha\beta} = \frac{(-1)^{\alpha-\beta}}{2(\Delta x)^2} \begin{cases} \frac{2}{(\alpha-\beta)^2}, & \text{for } \alpha \neq \beta, \\ \frac{\pi^2}{3}, & \text{for } \alpha = \beta, \end{cases} \tag{5.7}$$

with $\Delta x$ being the uniform grid spacing, and the potential energy matrix elements are just evaluations of the potential energy on grid points along the diagonal,

$$V_{\alpha\beta} = V(x_\alpha)\delta_{\alpha\beta}. \tag{5.8}$$

Instead we use the shielded Coulomb operator of **??**. With the DVR property that local operators are approximated as evaluated at the grid points in a DVR basis, we get

$$\begin{aligned} u^{\alpha\beta}_{\gamma\delta} &= \langle\theta_\alpha\theta_\beta|u'(x_1,x_2)|\theta_\gamma\theta_\delta\rangle \\ &= \int\int \mathrm{d}x_1\mathrm{d}x_2\theta_\theta(x_1)\theta_\beta(x_2)u'(x_1,x_2)\theta_\gamma(x_2)\theta_\delta(x_1) \\ &\approx \delta_{\alpha\gamma}\delta_{\beta\delta}u(x_\alpha,x_\beta). \end{aligned} \tag{5.9}$$

Upon initialization of a basis set of the type `ODSincDVR`, the following is the signature of the method.

```
basis_set = ODSincDVR(l, grid_length, potential=potential, u_repr="2d")
```

Here, `l` gives the number of grid points, `grid_length` gives the grid extent in the positive and negative directions and `potential` is a callable defining the value of the potential at the grid $V(x)$. The final parameter decides the representation of the Coulomb matrix elements, and should be a string of either `"4d"` for a 4d matrix representation $u_{pqrs}$, or `"2d"` for a 2d matrix $u_{pq} = u_{pqpq}$. The former is primarily for testing, and removes the purpose of using a DVR.

The matrix elements are calculated from the potential on setup of the basis, but this is a very fast procedure, and little care has to be made with respect to the size of the basis. In addition, the method `construct_sinc_dvr_functions` is provided to give values of the basis functions at arbitrary points.

[5] https://github.com/Schoyen/quantum-systems/blob/master/quantum_systems/sinc_dvr/one_dim/sinc_dvr.py

#### Sparse implementation

In earlier implementations we also had support for the COO-format via the `Python` library `sparse`[6], which provides efficient storage of sparse matrices. With it, the data is stored in three vectors (row, column, value), where the first two vectors store the coordinate of the corresponding value. The COO-format is an efficient format for sparse matrices, where most of the elements are zeros. Due to diagonality in two of the indices of the Coulomb matrix, the ratio of zero to non-zero elements are $L^2$, where $L$ is the basis size.

However, this implemetation currently suffers from computational inefficiency, as the implementation of `numpy` functions such as `einsum` and `tensordot` are not optimized in the current version of `sparse`. The format ended up being impractical and we decided to remove the COO-format entirely. Sparsity of the Coulomb matrix elements should instead be exploited by using the two-dimensional matrix representation described above. However, with specialized code for the contraction of sparse matrices utilizing the COO-format, symmetries like the local symmetry of DVR may more easily be exploited in a generalized manner.

### 5.1.4 Other libraries

For reusability of code and ease of installation, we have placed our own code in a separate github repository, called `py-master`[7]. It contains four small python libraries:

- `adiabatic_switching`, for time-evolution of systems with or without an adiabatic switching function. It contains the different adiabatic switching functions in `switching_functions.py`, the main time-evolution function for `OATDCC` classes in `run.py`, and sampler classes for sampling the relevant Coupled-cluster parameters in `sample.py`. Both tools for the other modules and methods for preparing non-interacting ground states for adiabatic switching is found in `tools.py`.

- `coupledcluster` contains coupled cluster code based on equations generated with symbolic programming, described later in this chapter. Not to be confused with `coupled-cluster` described above. The working and tested implementations are the CCD and the restricted CCD methods,

- `quantum_grid` contains our one-dimensional grid solver in `onedim.py` with potentials in `potential.py`, our Hartree-Fock implementations in `hf.py`, single-particle time-evolution methods in `crank.py`, and various convenience tools and tools for the other modules in `tools.py` and `miyagimadsen_tools.py`. There is also a small class structure for the calculation of Gauss-DVR functions in `dvr.py`.

- `quantum_grid.io` contains two input/output classes for storing, loading and organizing simulation data.

[6] https://pypi.org/project/sparse/
[7] https://www.github.com/halvarsu/py-master

In the parts of the code relating to simulation of OATDCC a variable called `par` can be seen. This is a dictionary containing all the relevant parameters of the simulation, which is generated from a `par.yaml` to serve as input to scripts that combine code to a full simulation with output. We have two such scripts, found in the folder `scripts/OACCD_gs` in the main thesis repository[8].

The first is `adiabatic_groundstate.py`, and covers the ground state calculations of OATDCC using adiabatic switching. An input yaml-file specifiec a range of parameters from an arbitrary choice of potential, via the basis set type (ODQD or ODSincDVR) to the choice of integrator and arguments to it. After a calculation, the script saves the output in the subfolder `output`. Example input files can be found in the subfolder `input/atomic`.

The second script is `mm_oa_laser.py`, which covers OATDCC simulations with a laser field interaction. The input `par.yaml` should now contain parameters such as the length of the time-simulation and other time-evolution parameters, whether the grid should be expanded in the case of a DVR basis, and the laser parameters. Example input files can be found in the subfolder `input/mm_oa_laser`.

Note that these repositories also contain both deprecated and unfinished code. That being said, we are fairly confident that the code is written in such a manner that it should be understandable for the somewhat experienced python user.

The libraries are all installed by running the single `setup.py` in the main folder of `py-master`, as the different libraries are packages in a larger python library.

## 5.2 Ground state solvers

The previous section contained an overview of both the `python` libraries that we have used and developed, and the contributions we have made to already existing libraries. In the two coming sections we'll go into more detail on how the solvers can be implemented. The present section is on solving the time-independent Schrödinger equation (TISE) numerically,)

$$H\left|\Psi\right\rangle = E\left|\Psi\right\rangle. \tag{5.10}$$

The TISE is an eigenvalue equation, which means that solving it involves finding the eigenpairs of the Hamiltonian, or, in our cases, just the lowest lying eigenpairs. The different many-body methods of quantum mechanics such as Hartree-Fock, Configuration Interaction and coupled-cluster are given by their respective ansatzes for the wave function. In mathematical terms this corresponds to taking different subspaces of the total Fock space which $H$ and $\left|\Psi\right\rangle$ act on and live in, respectively. We will now present the hierarchy of methods from single-particle methods used to find a single-particle basis used in a many-body calculation, via Hartree-Fock, arguably one of the simplest, but also most essential, many-body methods, before we present a simple ground-state solver for the Coupled-Cluster ansatz. The goal of this section is to present some straightforward ways of implementating these methods in `Python`.

[8] https://www.github.com/halvarsu/master

### 5.2.1 QuantumGridSolver

The theory behind this subsection is found in section 3.1, regarding the grid representation of a spectral basis for the single-particle problem in one dimension. The systems we are working with are simple enough that we will be able to represent the matrix elements of operators directly on a computer as matrices. To find the spectral basis, we set up and diagonalize the tridiagonal Hamiltonian given in eq. (3.9) as

$$h_{nm} = -\frac{1}{2}\frac{\delta_{n,m-1} - 2\delta_{n,m} + \delta_{n,m+1}}{\Delta x^2} + V(x_n)\delta_{nm}. \tag{5.11}$$

This gives us the eigenpairs $(\phi_n(x), E_n)$, where $\phi_p(x)$ are the wavefunctions in the grid representation that diagonalize $h_{nm}$.

A simple implementation of the above perscription for solving the single-particle problem using PYTHON is given by the class `QuantumGridSolver` in the library `quantum_grid`. This is our own implementation, though it is similar in nature to the `ODHO` class of `QuantumSystems`. Upon creation of an instance of `quantum_grid` the tridiagonal Hamiltonian is set up. The next step is to call the `solve` method, which first diagonalizes and then uses the wave functions $\phi_p(x)$ to carry out the integrals of eqs. (3.11), (3.13) and (3.14) in the grid basis. The process is given by the following steps,

```
from quantum_grid.onedim import QuantumGridSolver, HOPotential

grid_solver = QuantumGridSolver(
    Ngrid, L, potential = HOPotential(omega=1)
    )
energies, vecs = grid_solver.solve(k, a, calc_mel=True, **solver_kwargs)

# access matrix elements as attributes
u = grid_solver.u
x = grid_solver.dipole
```

Here $N_{grid}$ and $L$ defines the number of grid points and the interval, respectively. The argument `potential` is any callable, in this example one of the implemented potential classes found in the same module as `QuantumGridSolver`. Here we use `HOPotential`, which uses $V(x) = \frac{1}{2}\omega^2 x^2$ as the potentialenergy .

In the above example, the `solve` method returns the eigenpairs of the $k$ lowest lying states. If the boolean `calc_mel` is true, then matrix elements of the Coulomb operator $u$, the dipole moment $x$ and the overlap matrix $S$ are stored as parameters as shown above. The matrix elements of $u$ are given by the shielded Coulomb operator of eq. (3.17), and the parameter $a$ is the corresponding shielding parameter. Consequently, the grid solver now has the form required for many-body calculations by `QuantumSystem`.

Under the hood, `grid_solver.solve` uses `scipy.linalg.eigh_tridiagonal` to diagonalize the matrix, which is based on the `S/DSTEMR` routines from LAPACK (the Linear Algebra PACKage)[9]. The matrix is a tridiagonal hermitian matrix, which means that a large number of grid points can be used to

[9] http://www.netlib.org/lapack/

solve for the $k$ first eigenpairs in a fast manner. Examples can be found in the notebook `Onebody Grid Solver.ipynb` [10].

To integrate the matrix elements we use the implementation of Simpson's rule of `scipy.integrate.simps`[11]. Simpson's rule is based on a uniformly spaced grid $x_\alpha = a + \alpha\Delta x$, for $\alpha = 0, \ldots, n$, where $n$ is odd. and where the weights are given as

$$\omega_\alpha = \begin{cases} \frac{\Delta x}{3} & \text{if} \quad \alpha = 0, n \\ \frac{4\Delta x}{3} & \text{if} \quad \alpha \neq 0, n \text{ and } \alpha \text{ odd} \\ \frac{2\Delta x}{3} & \text{if} \quad \alpha \neq 0, n \text{ and } \alpha \text{ even} \end{cases} \tag{5.12}$$

For even $n$, `scipy.integrate.simps` has a choice of different ways to handle the extra interval. We use the default version which is calculate the integral twice each time omitting either the first or last point, using the trapezoidal rule for the missing interval, and averaging over the two resulting values.

### 5.2.2 Hartree-Fock

Hartree-Fock is the natural first use case for our basis sets in a manybody context. To recap, finding the Hartree-Fock state in a given basis is synonymous with solving the Roothan-Hall (RH) equations, given by

$$f(C)C = SC\varepsilon \tag{5.13}$$

here $C$ is a matrix of coefficients giving the transformation from an initial basis $|\chi_\alpha\rangle$ to the Hartree-Fock basis $|\phi_p\rangle = C_p^\alpha |\chi_\alpha\rangle$, $f(C)$ is the fock matrix,

$$f_\beta^\alpha = h_\beta^\alpha + u_{\beta i,AS}^{\alpha i} \tag{5.14}$$

Further, $S$ is the overlap matrix of the computational basis

$$S_\beta^\alpha = \langle \chi_\alpha | \chi_\beta \rangle \,, \tag{5.15}$$

and $\varepsilon$ is a vector of eigenvalues for the Fock matrix.

We are using the notation described in section 4.2.5, which means that the latin letter $i$ in $u_{\beta i}^{\alpha i}$ indicates the use of an occupied HF-basis element, which involves the coefficients, hence the dependency of $C$. Writing out this dependency, we get

$$f_\beta^\alpha = h_\beta^\alpha + (C^\dagger)_\gamma^i C_i^\delta u_{\beta\delta,AS}^{\alpha\gamma} = h_\beta^\alpha + D_\gamma^\delta u_{\beta\delta,AS}^{\alpha\gamma} \tag{5.16}$$

Here we recognized that an intermediate matrix $D_\gamma^\delta = (C^\dagger)_\gamma^i C_i^\delta$ could be constructed, to reduce the computational cost of the contractions. $D_\gamma^\delta$ is also called the density matrix.

The Hartree-Fock energy can be calculated from the density matrix and the Fock matrix as

$$E_{HF} = D_\alpha^\beta h_\beta^\alpha + \frac{1}{2} D_\alpha^\beta D_\gamma^\delta u_{\beta\delta,AS}^{\alpha\gamma}. \tag{5.17}$$

[10] https://github.com/halvarsu/master/blob/master/notebooks/Onebody Grid Solver.ipynb
[11] https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.simps.html

We rewrite this in terms of the precalculated Fock matrix to reduce the number of contractions,

$$E_{HF} = \frac{1}{2} D_{\alpha}^{\beta} (f_{\beta}^{\alpha} + h_{\beta}^{\alpha}). \tag{5.18}$$

Due to the non-linearity of the RH equations, the way to solve them is iteratively through a self-consistent procedure. In a given spin-restriction scheme, we use the following procedure to solve the RH equations

- Calculate an initial guess for the coefficients $C_0$, the density matrix $D_0$ and the Fock matrix $f_0$,
- while not converged:
  - Iterate the RH equations by solving the eigenvalue problem for the current Fock matrix, for a new set of coefficients $C^{(n)}$,
  - calculate the density matrix $D^{(n)}$,
  - calculate the Fock matrix $f^{(n)}$,
  - calculate the energy for convergence calculation $E^{(n)}$,
  - check convergence by some measure.

The convergence criteria is typically a test if the difference in energy between two subsequent steps is less than some tolerance,

$$\left| E_{HF}^{(n)} - E_{HF}^{(n-1)} \right| < \epsilon. \tag{5.19}$$

Another commonly used measure is the Frobenius norm of the difference in density matrices of two subsequent steps,

$$\left| D^{(n)} - D^{(n-1)} \right|_F < \epsilon. \tag{5.20}$$

The self-consistent procedure above is not guaranteed to converge, and a remedy is to use so-called mixing. With mixing, the density matrix of step $n$ is a mix of the density matrix of the previous step $D^{(n-1)}$ and the density matrix calculated from the current coefficients. If the latter is written $\bar{D}^{(n)}$, then at each step the density matrix is given by

$$D^{(n)} = \alpha D^{(n-1)} + (1 - \alpha) \bar{D}^{(n)} \tag{5.21}$$

Another popular procedure is the Direct Inversion of the Iterative Subspace (DIIS)[**ref**]. We have not implemented DIIS for HF, but we have used it as part of the `coupled-cluster` library and will therefore explain it in the section on coupled-cluster.

### Implementation

Our `Python` implementation of Hartree-Fock is with an object-oriented approach, to help reusability and readability of our code. The core is the abstract base-class called `HFBase`. The self-consistent procedure for solving the Roothan Hall equations are implemented in the `solve` method, which can be seen in **??** 1.

The different subclasses are based on the different spin and spatial symmetries of the basis.

A typical usage of our HF solver is the following,

```
from quantum_grid.hf import RHF
hf = RHF(system, check_convergence=check_conv)
C_hf = hf.solve(C0=None, tol=1e-8, max_iter=100, verbose=True, alpha=0.5)
```

Here the `RestrictedHartreeFock` subclass (which we will describe in more detail later) was used as an example of a subclass, and `system` is either a fully set up `QuantumGridSolver` or a `QuantumSystem`. Note that the matrix elements have to be on a restricted form to use `RHF`, which means that the spin is projected out. The function `check_conv` is optional, though if it is supplied then it should be callable which returns a boolean calculated from the `local()`-parameters of the `solve` method.

The fact that `HFBase` is abstract means that only subclasses of it can be instantiated. It also has four abstract methods which need to be overwritten by said subclasses, which are

- `density_matrix`, for the creation of the density matrix from the coefficients $C_{\alpha p}$,
- `fock_matrix`, for the creation of the Fock matrix $f_{\alpha\beta}$ from the matrix elements of $h_{\alpha\beta}$ and $u^{\alpha\beta}_{\gamma\delta}$ and the density matrix $D_{\alpha\beta}$,
- `iterate`, to iterate the RH equations by diagonalizing the Fock matrix, and
- `hf_energy` to calculate the energy from the density matrix and matrix elements.

### Spin symmetries

The implementation of the methods is given by the symmetries of the system at hand. As such, there are in total six different Hartree-Fock solvers, one normal and one DVR-adapted version for all three types of spin-symmetries, general, unrestricted and restricted.

The definition of the three types of spin-symmetries can be summarized as follows. In general Hartree-Fock, no assumptions are made about the spin or spatial degrees of freedom, and in theory all kinds of systems can be studied. In restricted Hartree-Fock, the restriction is that the system should be unaffected by spin, with the consequence that all spatial orbitals can be real and that all spatial orbitals are doubly occupied. In unrestricted Hartree-Fock, this strict restriction is eased and the spatial orbitals are allowed to be singly occupied and have different energy based on spin. This allows studies of spin-degenerate systems, such as larger separations of $H_2$.

The entire code base can be displayed efficiently using an open source software called Sourcetrail[12]. In fig. 5.5, we have used Sourcetrail to index the entire

[12] https://www.sourcetrail.com/

**Listing 1** Solve method of HFBase, common to all our HF solvers.

```
    def solve(
        self,
        C0: Optional[np.array] = None,
        tol: float = 1e-8,
        max_iter: int = 100,
        verbose: bool = False,
        alpha: float = 0,
    ) -> Tuple[np.array, np.array]:
        """Iteratively solves the Hartree-Fock equations on basis form,
            $F(C)C = S C eps$
        for a given initial C0.
        Parameters
        ==========
        C0  : initial coefficients (optional), default is identity matrix
        tol : tolerance criteria for convergence (depends on self.check_convergence)
        max_iter : maximum number of iterations before convergence
        verbose : to print or not to print. If int, then prints every verbose number of step
        alpha : mixing parameter for improved convergence.

        Returns
        =======

        C : coefficients
        """

        assert type(verbose) in (bool, int), "verbose must be bool or int"
        if C0 is None:
            C0 = np.eye(self.system.l)

        # initial density matrix and fock operator
        D = self.density_matrix(C0)
        D_prev = D
        F = self.fock_matrix(D)
        C = self.iterate(F)

        # some high initial energy to initialize convergence
        E = 1e8

        converged = False
        for it in range(max_iter):
            prev_E = E
            prev_D = D
            D = alpha * D + (1 - alpha) * self.density_matrix(C)
            F = self.fock_matrix(D)
            E = self.hf_energy(D, F)

            if verbose and ((it % verbose) == 0):
                print(it, E)

            if callable(self.check_convergence):
                converged = self.check_convergence(locals())
            else:
                converged = np.abs((prev_E - E) / E) < tol
            if converged:
                break
            C = self.iterate(F)

        if not converged:
            import warnings

            warnings.warn("did not converge before max_iter was reached")
        return C
```

class hierarchy, with calls to methods and references to parameters shown in blue and yellow, respectively.

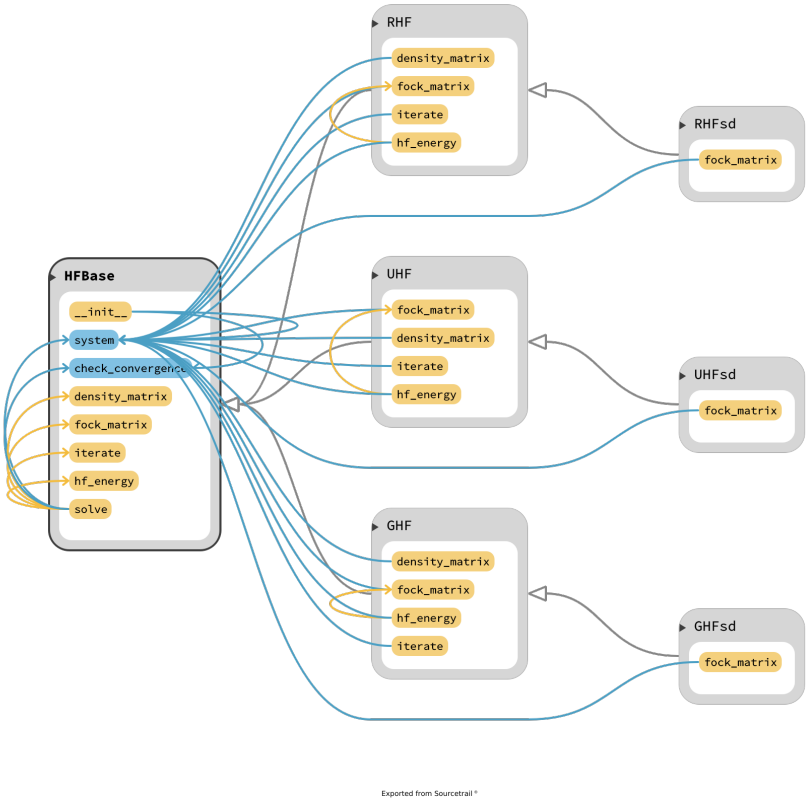



Figure 5.5: Code graph of our minimalistic HF implementation. The dependency of the system parameter is apparent. For boxes and arrows respectively then gray indicates classes and class inheritance, yellow indicates functions and function calls, blue indicates parameters and parameter references.

### 5.2.3 Coupled-Cluster

The next step in the hierarchy is coupled-cluster. The parameters of the coupled-cluster ansats are the amplitudes $\tau_\mu$ and $\lambda_\mu$, as formulated in section 4.6.6. We will here only focus on the $\tau_\mu$-equations (the $\lambda_\mu$ are found analogously), and we follow the presentation of Helgaker et al. [21]. The $\tau$-equations for the coupled-cluster ground state are given as

$$\langle \Phi_\mu | (H_N e^T)_C | \Phi_0 \rangle = 0. \tag{5.22}$$

where $\mu$ represents an arbitrary excitation such that

$$|\Phi_\mu\rangle \in \{ a_a^\dagger a_i \, |\Phi_0\rangle \, , a_a^\dagger a_i a_b^\dagger a_j \, |\Phi_0\rangle \, , \dots \}_{i,j,\dots=1,\dots,N}^{a,b,\dots=N,\dots,L}, \tag{5.23}$$

$H$ is the Hamiltonian, and $\hat{T}(\boldsymbol{\tau})$ is the cluster operator, which depends on the vector of amplitudes $\boldsymbol{\tau}$ with elements $\tau_\mu$.

The amplitude equations can be formulated as a single vector equation, where we wish to find the roots of the vector-evaluated function that evaluates the right hand side of the $\tau$-equations,

$$\Omega_\mu(\boldsymbol{\tau}) = \langle \Phi_\mu | e^{-\hat{T}(\boldsymbol{\tau})} \hat{H} e^{\hat{T}(\boldsymbol{\tau})} | \Phi_0 \rangle \tag{5.24}$$

Because $\Omega_\mu(\boldsymbol{\tau})$ is nonlinear in $\tau_\mu$, finding the roots must be done iteratively. One possibility would be to use Newton's method, which is derived by Taylor expanding and neglecting terms higher than first order in the update step $\Delta\tau_\mu$. This gives an equation to solve for $\Delta\tau_\mu$,

$$\begin{aligned} \Omega_\mu(\boldsymbol{\tau}^{(n)} + \Delta\boldsymbol{\tau}^{(n)}) \approx \Omega_\mu(\boldsymbol{\tau}^{(n)}) + \mathcal{J}_{\mu\nu}(\boldsymbol{\tau}^{(n)}) \Delta\tau_\nu^{(n)} = 0, \\ \mathcal{J}_{\mu\nu}(\boldsymbol{\tau}^{(n)}) \Delta\tau_\nu^{(n)} = -\Omega_\mu(\boldsymbol{\tau}^{(n)}) \end{aligned} \tag{5.25}$$

Here we defined the $\mathcal{J}_{\mu\nu}(\boldsymbol{\tau}^{(n)})$ as the Jacobian matrix

$$\mathcal{J}_{\mu\nu}(\boldsymbol{\tau}^{(n)}) = \frac{\partial}{\partial \tau_\nu} \Omega(\boldsymbol{\tau}) \bigg|_{\boldsymbol{\tau}=\boldsymbol{\tau}^{(n)}}, \tag{5.26}$$

However, solving eq. (5.25) directly is impractical due to the number of amplitudes in coupled-cluster, which makes a direct approach to solving for $\Delta\boldsymbol{\tau}$ unfeasible. Instead we write the Hamiltonian in terms of the one-body Fock operator $\hat{F}$ and a two-body pertubation operator $\hat{W}$,

$$\hat{H} = \hat{F} + \hat{W}. \tag{5.27}$$

Note that $\hat{F}$ is here the second quantized version of $\hat{F}$ in section 4.4. We can further split the fock operator into a diagonal and a non-diagonal part, $\hat{F} = \hat{F}_{diag} + \hat{F}_{non-diag}$, given as

$$\hat{F}_{diag} = \sum_p f_p^p a_p^\dagger a_p, \qquad \hat{F}_{non-diag} = \sum_{p \neq q} f_q^p a_p^\dagger a_q. \tag{5.28}$$

Then the part of the amplitude equations involving $\hat{F}_{diag}$ becomes

$$\langle \Phi_\mu | e^{-\hat{T}} \hat{F}_{diag} e^{\hat{T}} | \Phi_0 \rangle = D_\mu \tau_\mu, \tag{5.29}$$

where $D_\mu$ is the difference in single-particle energy between the unoccupied orbitals and the occupied orbitals of a given excitation,

$$D_\mu = \sum_{a \in \mu} h_a^a - \sum_{i \in \mu} h_i^i. \tag{5.30}$$

This lets us write the vector evaluated function and its Jacobian as

$$\Omega_\mu(\boldsymbol{\tau}^{(n)}) = D_\mu \tau_\mu^{(n)} + \langle \Phi_\mu | e^{-T(\boldsymbol{\tau}^{(n)})} (\hat{F}_{non-diag} + \hat{W}) e^{T(\boldsymbol{\tau}^{(n)})} | \Phi_0 \rangle, \tag{5.31}$$

$$\mathcal{J}_{\mu\nu}(\boldsymbol{\tau}^{(n)}) = D_\mu \delta_{\mu\nu} + \frac{\partial}{\partial \tau_\nu} \langle \Phi_\mu | e^{-T} (\hat{F}_{non-diag} + \hat{W}) e^{T} | \Phi_0 \rangle \bigg|_{\boldsymbol{\tau}=\boldsymbol{\tau}^{(n)}} \tag{5.32}$$

We get the so-called quasi-Newton approach by assuming the second term of the Jacobian can be neglected. This is a valid assumption if the reference state is such that $\hat{F}_{non-diag} + \hat{W}$ is small. If the reference state is the Hartree-Fock state then Brillouins theorem [46] states that the non-diagonal parts of the Fock operator are zero. Other reference states can be used, but the Fock operator should be diagonally dominant to ensure convergence.

Inserting into eq. (5.25), we find the quasi-Newton update to be

$$\Delta\tau_\mu^{(n)} = -\frac{\Omega_\mu(\boldsymbol{\tau}^{(n)})}{D_\mu}, \tag{5.33}$$

and the expression for the amplitudes in the next iteration is

$$\boldsymbol{\tau}^{(n+1)} = \boldsymbol{\tau}^{(n)} + \Delta\boldsymbol{\tau}^{(n)} = \boldsymbol{\tau}^{(n)} - \frac{\Omega_\mu(\boldsymbol{\tau}^{(n)})}{D_\mu}. \tag{5.34}$$

For the initial amplitudes, we use the zero amplitudes $\boldsymbol{t}^{(0)} = 0$. This is continued until convergence is met, which is defined to be when the difference in norm between two sets of amplitudes are less than a given tolerance $\epsilon$,

$$|\boldsymbol{\tau}^{(n+1)} - \boldsymbol{\tau}^{(n)}| < \epsilon. \tag{5.35}$$

.

The case for coupled-cluster with $\boldsymbol{\lambda}_\mu$ amplitudes is completely analogous, though the iteration procedure is carried out subsequently by first iterating the $\tau$-amplitudes and then the $\lambda$-amplitudes.

The quasi-Newton can in some cases be slow and is not guaranteed to converge. To increase the speed and likelyhood of convergence, it is common to use mixing as when solving the Roothan-Hall equations. While we explained Alpha mixing in the previous section, for coupled-cluster we use a more sophisticated procedure called Direct Inversion of The Active Subspace (DIIS) [21]. In this procedure, the amplitudes in the next iteration is instead obtained by interpolating between the previous estimates,

$$\boldsymbol{\tau}^{(n+1)} = \sum_{m=1}^{n} c_m \boldsymbol{\tau}^{(n)}. \tag{5.36}$$

The weights are constrained to sum to 1,

$$\sum_{m=1}^{n} c_m = 1 \tag{5.37}$$

Under this constraint we wish to set the weights as to minimize the error vector $\mathbf{e}_n$ given by the function that we wish to find the roots of

$$\mathbf{e}_n = \Omega(\boldsymbol{\tau}^{(n)}). \tag{5.38}$$

Minimization of an error vector under a constraint is given by Lagrange's multiplier method, see for example [21]. By minimizing the error vector given above, we are guaranteed that an optimal error vector corresponds to the solution of the coupled-cluster equations.

### Automatic Derivation of Coupled-Cluster Equations and Codes

We have implemented a simple coupled-cluster doubles solver (CCD) and a restricted coupled-cluster doubles sovler (RCCD) for demonstration purposes. The resulting `Python` modules was tested, and produces consistent results with existing coupled-cluster solvers. None of the results of this thesis are calculated using these implementations, and they only serve illustrative purposes with regards to the implementation of the code, and for the difference in time usage between general and spin restricted versions. This latter serves as a possible next step for the OATDCCD-method.

Note that for the actual calculations with OATDCCD in this thesis, we decided to use the coupled-cluster library of Schoyen and Winther-Larsen, as it is a well coded library which is easily extended to other systems and methods.

To generate the amplitude equations of CCD and RCCD we used the generative `Python` library `drudge`[13], based on the symbolic programming library `sympy`[14]. A typical use case can be seen in the notebook `RCCD derivation with DandG.ipynb`. [15]. We first define a particle-hole context, which contains definitions of dummy indices $p, \ldots, i, \ldots, a \ldots$ etc, excitation operators $E_i^a$ and the two-body Hamiltonian $H$. We then use the excitation operators to define the doubles cluster operator,

$$\hat{T}_2 = \frac{1}{2} t_{ij}^{ab} \hat{E}_i^a \hat{E}_j^b, \tag{5.39}$$

The similarity transformed hamiltonian is then calculated by iteratively producing and adding on nested commutators as

$$\bar{H} = \sum_{n=0}^{4} H^{(n)}, \tag{5.40}$$

where $H^{(n)} = \frac{1}{n}\big[H^{(n-1)}, T_2\big]$, $H^{(0)} = H$. The final step is to calculate the coupled-cluster equations

$$\langle \Phi_0 | \bar{H} | \Phi_0 \rangle = \Delta E, \tag{5.41}$$

$$\big\langle \Phi_{ij}^{ab} \big| \bar{H} \big| \Phi_0 \big\rangle = \langle \Phi_0 | E_a^i E_b^j \bar{H} | \Phi_0 \rangle = 0. \tag{5.42}$$

by taking the vacuum expectation value of the two expressions with drudge. The resulting equations can be seen in the appendix, along with the script to generate the equations.

We then use `gristmill`, another excellent `Python` library by the same author[**ref**], to produce code for languages such as `Python` or c++ directly.

### Code

With the amplitude equations gotten from the methods described above, we have implemented a small coupled-cluster library in `Python` called `coupledcluster`

[13] https://github.com/tschijnmo/drudge

[14] https://www.sympy.org/

[15] https://github.com/halvarsu/master/blob/master/notebooks/RCCD derivation with DandG.ipynb

(not to be confused with the library `coupled-cluster` described earlier). We use an object-oriented approach similar to our Hartree-Fock implementation. The core is the abstract class `CCBase`, and the quasi-Newton algorithm for solving the coupled-cluster equations are implemented in the `calc_amp` method, which can be seen in **??** 2. In addition to this, `CCBase` also covers setting up the pertubation-theory like numerator $D_\mu$ as described above.

**Listing 2** The solve method of CCBase, common to all our CC solvers.

```
    def calc_amp(self, max_iter=100, tol=1e-5, alpha=0.4, verbose=True):
        no = self._no
        nv = self._nv

        t = zeros((nv, nv, no, no))

        D = self.make_diagthing()
        v, o = self._v, self._o

        t1 = self._system.u[v, v, o, o] / D
        print(self.corr_energy(t))

        i = 0
        prev_energy = np.inf
        energy = self.corr_energy(t)
        residue = 0

        if verbose:
            print("{:4} {:10} {:10}".format("i", "residue", "energy"))
        while (np.abs(energy - prev_energy) > tol) and (i < max_iter):
            rhs = self.calc_rhs(t)
            residue = np.linalg.norm(rhs)
            t = rhs / D
            # t = (1-alpha)*t + alpha*rhs/D

            prev_energy = energy
            energy = self.corr_energy(t)

            if verbose:
                print("{:<4} {:<10.6f} {:<10.6f}".format(i + 1, residue, energy))
            i += 1
        self.t = t
        return t
```

The specific amplitude equation and energy expressions are implemented in the subclasses of `CCBase`. Currently only the `CCD` truncation level is implemented, though using `drudge` and `gristmill` it should be simple to define other subclasses as well, especially the spin restricted CCD method. Using `PYSCF` for calculation of matrix elements, we can find the CCD correlation energy of a Helium system,

```
from quantum_systems import construct_pyscf_system_rhf
from coupledcluster.ccd import CCD

system = construct_pyscf_system_rhf('HE')

ccd_solver = CCD(system)
ccd_solver.calc_amp(max_iter = 100, tol = 1e-5, verbose=False)
```

```
e_HF = calc_HF_energy(system)
e_CCD =  ccd_solver.corr_energy()

total_energy = e_HF + e_CCD
```

Here `construct_pyscf_system_rhf` is a method from the `quantum_systems` library[16], which, given an input describing the molecule according to the PySCF standard[17], constructs a system class with matrix elements from PySCF in the given basis. If omitted, the basis argument defaults to cc-pVDZ, which are Gaussian basis sets part of the correlation-consistent polarized valence basis sets class[13]. Similar calculations for all noble gases up to Krypton can be found in the notebook `CCD Calculations.ipynb`[18], both for CCD and RCCD. The total energy given by the HF energy plus the CCD correlation energy is shown in table 5.1. We included a graph of the time usage in fig. 5.6. For small systems the overhead of `Python` dominates, but for the larger ones the reduction of each dimension by a factor half in restricted CCD provides a great speedup.

Table 5.1: Total CCD energy (HF energy plus correlation energy) in atomic units for the noble gases up to Krypton. The basis is HF constructed from cc-PVDZ orbitals using `pySCF`, and the CCD energy is calculated using our `coupledcluster` Python library.

| Atom | CCD energy | rCCD energy |
|---|---|---|
| He | -2.88759 | -2.88759 |
| Be | -14.6169 | -14.6169 |
| Ne | -128.6795 | -128.6795 |
| Ar | -526.9561 | -526.9561 |
| Kr | -2752.121 | -2752.121 |

For future work with many-body theory it is clear that the drudge and gristmill libraries are powerful tools when it comes to deriving both equations in second quantization, and for producing effective implementations. Using gristmill, we could also have optimized the equations using intermediates.

## 5.3 Time-dependent problems

Having found the ground state properties of the system, we move on to the main task at hand, namely dynamics. To do this we solve the TDSE,

$$i\frac{\partial}{\partial t}\left|\Psi\right\rangle = H\left|\Psi\right\rangle. \tag{5.43}$$

The TDSE is a partial differential equation, where time and space is coupled. However, by expanding the wavefunction in the same basis sets used to solve

[16] https://github.com/Schoyen/quantum-systems/blob/master/quantum_systems/custom_system.py

[17] https://sunqm.github.io/pyscf/gto.html

[18] https://github.com/halvarsu/master/blob/master/notebooks/CCD Calculations.ipynb

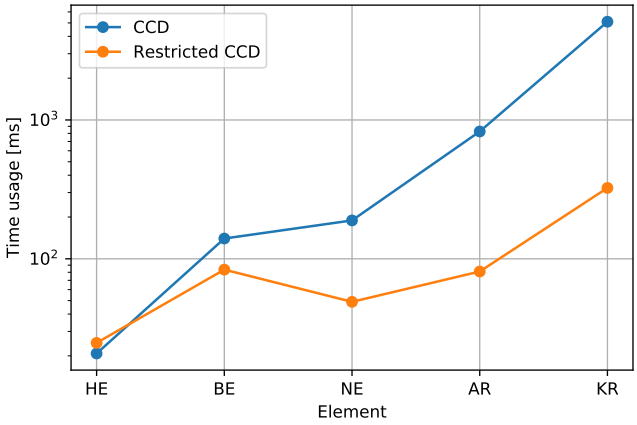



Figure 5.6: Time usage for the calculation of CCD ground state for the noble gases up to Krypton using both a regular CCD solver with explicit spin, and spin restriced CCD solver. The basis is HF constructed from cc-PVDZ orbitals using `pySCF`. For small systems, the overhead of `Python` dominates.

the TISE, we can get a time-dependent description by evolving the coefficients of this expansion in time.

We divide this into to separate cases. To propagate independent particle wavefunctions such as those of single-particle problems and time-dependent Hartree-Fock, we are varying the coefficients of a single-particle basis expansion. For many-body problems such as time-dependent coupled-cluster we are instead changing amplitudes of the many-body wavefunction in a basis of Slater determinants. In the case of orbital adaptive time-dependent coupled-cluster, the coefficients of the single-particle basis is also evolved simultaneously with the amplitudes describing the slater determinant.

We use two different formulations of the TDSE to solve the two separate cases. For the former, we use the time-propagator $\hat{U}(t, t_0)$ described in section 2.3.4 which propagates a wave-function in time from $t_0$ to $t$. This is possible because we have an explicit representations of the Hamiltonian, and can construct approximations for $\hat{U}(t_{i+1}, t_i)$ at each time-step $t_i$. This is formulated as a matrix equation at each step, where the coefficient matrix of the single-particle basis expansion in the next step, $C(t_{i+1})_p^\alpha$, is solved for.

For the latter case of TDCC and OATDCC we do not have explicit representation of the Hamiltonian. Instead we consider the equations of motion for the coefficients and amplitudes, derived from a principle of least action. These equations of motions are ordinary differential equations, which we solve numerically using the Gauss integrator[38].

### 5.3.1 Unitary time propagation operator

As stated above, to propagate independent particle methods in time we approximate the unitary time propagation operator $U(t_{i+1}, t_i)$ at each time-step

$t_i$, which lets us evolve a state along a discretized time step

$$|\Phi(t_{i+1})\rangle = U(t_{i+1}, t_i)\,|\Phi(t_i)\rangle\,, \quad i = 0, \ldots, N_t. \tag{5.44}$$

We use the approximations from the Magnus expansion descibed in section 2.3.4.

In other words, we discretize the time domain $[t_0, t]$ into a series of $N$ time steps $t_i = t_0 + i\Delta t$ for $i = 0, \ldots, N-1$, with $t_{N-1} = t$ and $\Delta t = (t - t_0)/(N-1)$. We then evolve a state from time $t_0$ to time $t$ by splitting the total unitary operator $U(t, t_0)$ into a series of subsequent unitary operators,

$$U(t, t_0) = \prod_{i=0}^{N} U(t_{i+1}, t_i) \tag{5.45}$$

In section 2.3.4 we wrote $U$ in terms of the exponential of an operator, and truncating the Magnus expansion after the first term gives us

$$U(t + \Delta t, t) \approx \exp\left\{ \frac{1}{i\hbar} \int_t^{t+\Delta t} \mathrm{d}t' H(t') \right\} \tag{5.46}$$

We get a forward Euler-like integrator of order $\mathcal{O}(\Delta t)$ if we approximate the Hamiltonian as constant over the time interval. This gives

$$U(t + \Delta t, t) \approx \exp\left\{ \frac{1}{i\hbar} \Delta t H(t) \right\}, \tag{5.47}$$

Higher order integrators are completely analogous to the solvers of regular ordinary differential equations, such as the midpoint method or leapfrog methods.

Using eq. (5.47) requires calculating the exponential of a matrix. The brute force way to do this numerically is to do a spectral decomposition like in section 3.2.3. We first diagonalize the operator $A$ into its eigenpairs, $(\lambda_i, \mathbf{v}_i)$,

$$A = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T \tag{5.48}$$

This can be done with the function `numpy.linalg.eigh` for a Hermitian matrix. The exponential of $A$ can then be found by applying the exponential function to the eigenvalues and then taking the outer product of the eigenvectors again

$$e^A = \sum_i e^{\lambda_i} \mathbf{v}_i \mathbf{v}_i^T \tag{5.49}$$

A simple implementation of a step with the Forward Euler like method using `Python` is the following

```
E, vecs = np.linalg.eigh(H)
U = vecs * (np.exp(-1j * dt * E)) @ vecs.T
```

Here `H` is the Hamiltonian evaluated at the current time and `dt` is the step length $\Delta t$. Alternatively, in `Python` we can use the function `scipy.linalg.expm`, which uses a more sofisticated approach[36] to get correct results for the

exponential of a matrix to machine precision in a faster way. Its usage is described in the documentation[19].

An alternative method is to instead approximate the Hamiltonian with its Taylor expansion,

$$e^x = 1 + x + \dots. \tag{5.50}$$

The simplest approach is to then truncate the series after the first terms,

$$U(t, t_0) \approx \exp\left\{\frac{\Delta t}{i\hbar}H(t)\right\} \approx 1 + \frac{\Delta t}{i\hbar}H(t) + \mathcal{O}(\Delta t^2). \tag{5.51}$$

However, the resulting operator is now not unitary. We can see this if we apply its conjugate to itself

$$\left(1 + \frac{\Delta t}{i\hbar}H(t)\right)\left(1 - \frac{\Delta t}{i\hbar}H(t)\right) = \left(1 + \frac{\Delta t^2}{\hbar^2}H(t)H(t)\right) \neq 1 \tag{5.52}$$

Non-unitarity leads to a serious energy drift over time, something we have confirmed with our implementation. Instead, a common method called the Crank-Nicholson algorithm[39] is found by using the approximation

$$e^x = \frac{1 + x/2}{1 - x/2} + \mathcal{O}(x^3), \tag{5.53}$$

Crank-Nicholson is correct to second order in the step length, which can be shown by taylor expanding both sides above and comparing order by order. This gives the following expression for $U(t, t_0)$,

$$U(t, t_0) \approx \left[1 - \frac{\Delta t}{2i\hbar}H(t)\right]^{-1}\left[1 + \frac{\Delta t}{2i\hbar}H(t)\right]. \tag{5.54}$$

The Crank-Nicholson method is unitary, as seen by writing

$$\begin{aligned} UU^\dagger &= (1 + i\tilde{H})^{-1}(1 - i\tilde{H})(1 + i\tilde{H})(1 - i\tilde{H})^{-1} \\ &= (1 + i\tilde{H})^{-1}(1 + i\tilde{H})(1 - i\tilde{H})(1 - i\tilde{H})^{-1} = 1, \end{aligned} \tag{5.55}$$

where we defined $\tilde{H} \equiv \Delta t H/2\hbar$ and recognized that the two middle terms commute. Rewriting eq. (5.54), we can formulate Crank-Nicholson as an implicit scheme on the form,

$$\left[1 - \frac{\Delta t}{2i\hbar}H(t)\right]|\psi(t_{i+1})\rangle = \left[1 + \frac{\Delta t}{2i\hbar}H(t)\right]|\psi(t_i)\rangle \tag{5.56}$$

Using the grid basis of the time-independent problem of the previous section, the Crank-Nicholson method can be written

$$\left[1 - \frac{\Delta t}{2i\hbar}H_{nm}(t)\right]\psi(t_{i+1}, x_m) = \left[1 + \frac{\Delta t}{2i\hbar}H_{nm}(t)\right]\psi(t_i, x_m), \tag{5.57}$$

with the unknown vector $\psi(t+\Delta t, x_m)$ to be solved for in a matrix equation of the type $Ax = b$. The explicit form of the time dependent part of the Hamiltonian will be adressed later, but in the dipole approximation it is diagonal the grid basis, $H^I_{nm}(t) = H^I_{nn}(t)\delta_{nm}$. As such, the position basis Hamiltonian is still tridiagonal just like in the time-independent case, and we can solve the linear equation in linear time with respect to the number of basis functions[49].

[19] https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.expm.html

### CrankEvolution

The class `CrankEvolution` uses the methods described above to evolve a position basis wave function in time. When creating an instance of `CrankEvolution`, the user can choose between one of the many `evolve_step_*` methods. However, in the grid basis the tridiagonal Crank-Nicholson method `evolve_step_crank_tridiag` is by far the optimal in terms of speed. It uses the tridiagonal form of the Hamiltonian to calculate the next time step in linear time.

Time evolution is generally a bit more involved than ground state calculations, but in the following code snippet we first find the ground state and first excited state of a Harmonic Oscillator potential using the `QuantumGridSolver`, and then use the tridiagonal Crank-Nicholson method to evolve it in time with frequency given by the resonance frequency.

```
import numpy as np
import matplotlib.pyplot as plt
from quantum_grid.crank import CrankEvolution, HOLaser
from quantum_grid.onedim import HOPotential, QuantumGridSolver

dx = 1e-1
L = 8
N = int(2*L/dx + 1)
print('N = {}'.format(N))
m = omega = 1
pot = HOPotential(omega=omega)

solver = QuantumGridSolver(N, L, potential=pot)
k = 2
energies, v0 = solver.solve(k=k)

# Time evolution
dt = 0.05

T = 500
dt = 0.01
Nt = int(T/dt)

amp = 0.1
laser = HOLaser(amp = amp, omega = omega)
HOevolver = CrankEvolution(dt, solver=solver, external=laser).setup(solver)

laser_interval=[0,5]
v_values, t_values = HOevolver.evolve(Nt, T, v0[:,0], ret_t=True, laser_interval=laser_interval)
```

## 5.3.2 Time dependent Hartree-Fock

To have a benchmark to compare our new methods, we first started with the implementation of a Time-dependent Hartree-Fock method. We made our own implementation of hartree fock in the `Python` module `quantum_grid.hf`, which includes RHF, UHF, GHF (general hartree fock for spin systems), and versions of these tailored to DVR-basis sets. The classes only require a system-class containing the $h$- and $u$-matrix elements to operate, and can be subsequently solved through the `solve` method. All contractions have been implemented with the einsum-method for its great tradeoff between simplicity and speed.

With the ground state hf-solver in place, the implementation of TDHF is very straightforward. According to **??**, the time evolution of the $C$-coefficients is governed directly by the fock matrix. Remembering to update the fock matrix each time step, we can then solve the time dependent Schrödinger equation for this problem using some standard method such as crank-nicholson to approximate the propagation operator $U$, or by calculating the exponential of the Hamiltonian in the current time step.

### 5.3.3 Equations of motion

The second approach we use to solve the TDSE in the many-body case is by considering the equations of motions of the parameters in the specific ansatz used. These equations of motion can be rewritten in the form of a set of coupled ordinary differential equation (ODE)

$$\dot{y} = f(y, t), \tag{5.58}$$

where $y(t)$ is the time-dependent parameters of the ansatz used. The simplest conceivable method to numerically solve an ODE is with the forward Euler method, where the derivative of a function is approximated using the forward finite difference formula

$$f(y(t_i), t_i) = \dot{y} = \frac{y(t_{i+1}) - y(t_i)}{\Delta t} \tag{5.59}$$

$$\Rightarrow y(t_{i+1}) = y(t_1) + \Delta t f(y(t_i), t_i). \tag{5.60}$$

This is an explicit method, where the solution in the next step is an explicit function of the previous step. However, forward euler is too simplistic for our needs, as it leads to severe energy deviations due to non-symplicity and also does not provide a good convergence with step size. We will now present other integrators with better accuracy and more favourable properties, specifically Runge-Kutta 4 (RK4) and the symplectic Gauss integrator[37], in a later section.

do this for the many-body action-based methods of time-dependent coupled-cluster (TDCC) and orbital adaptive time-dependent coupled-cluster (OAT-DCC).

### 5.3.4 Time-Dependent Coupled-Cluster

The equations of motion for TDCC is given in eqs. (4.142) and (4.143) in terms of the right-hand sides of the regular amplitude equations, which is more explicitly

$$\dot{\tau}_\mu = -i \left\langle \Phi_\mu | e^{-\hat{T}(t)} \hat{H}(t) e^{\hat{T}(t)} | \Phi \right\rangle ,$$
$$\dot{\lambda}_\mu = i \left\langle \Phi | \Big(1 + \hat{\Lambda}(t)\Big) e^{-\hat{T}(t)} \Big[\hat{H}(t), \hat{X}_\mu\Big] e^{\hat{T}(t)} | \Phi \right\rangle .$$

These are general equations which holds for any truncation level of coupled-cluster. The equations above are formulated as differential equations on the form that scipy requires, and as such it is simple to perform time-depdendent calculations on a coupled cluster state. For each step, one first updates the single-particle matrix elements of the Hamiltonian, $h^p_q(t)$ and $u^{pq}_{rs}(t)$. As written,

these updates might be in both the one-body parts of the two-body parts, such as for a dipole laser or the adiabatic switching function, respectively. In the normal-ordered formulation of the coupled cluster equations, the matrix elements of the fock-operator are then calculated $f_q^p(t) = h_q^p + \frac{1}{2}u_{qi}^{pi}$. We then use the updated matrix elements and the set of amplitudes to evaluate the right hand side of the amplitude equations, where they are the unmodified equations unlike the quasi-Newton approach. Note that the $\tau$-equations are still independent of the $\lambda$-amplitudes The number of evaluations of the right hand side above depends on the integrator used, with RK4 using four evaluations per time-step as an example.

We have not implemented a separate TDCC solver but use the one implemented in the library `coupled_cluster` described earlier.

### 5.3.5 Orbital Adaptive Time-dependent Coupled-Cluster

The equations of motion for the OATDCCD method quite a bit is more involved. Recall that the parametrization of the many-body wavefunction in OACC is

$$|\Psi(\mathbf{C}, \boldsymbol{\tau})\rangle = e^T |\Phi_0\rangle \tag{5.61}$$

$$\left\langle \tilde{\Psi}(\tilde{\mathbf{C}}, \boldsymbol{\tau}, \boldsymbol{\lambda})\right| = \left\langle \tilde{\Phi}_0\right| (1 + \Lambda)e^{-T(\tau_\mu)} \tag{5.62}$$

With this representation the wavefunction is parametrized by the two sets of amplitudes $\boldsymbol{\tau} = [\tau]_\mu$ and $\boldsymbol{\lambda} = [\lambda]_\mu$ indexed by an index $\mu$ which goes over all possible excitations at the given truncation level, and the coefficient matrices $\mathbf{C}$ and $\tilde{\mathbf{C}}$ giving the transformation between the original basis to the OA-basis[20],

$$|\varphi_p\rangle = C_p^\alpha |\chi_\alpha\rangle , \qquad \langle\tilde{\varphi}_p| = \tilde{C}_\alpha^p \langle\chi^\alpha| . \tag{5.63}$$

As long as the original basis is biortonormal $\langle\tilde{\chi}^\alpha|\chi_\beta\rangle = \delta_{\alpha\beta}$, demanding biortonormality in the OA-basis gives

$$\delta_q^p = \langle\tilde{\varphi}^p|\varphi_q\rangle = \tilde{C}_\alpha^p C_q^\alpha \tag{5.64}$$

We are able to use Wick's theorem to derive the regular coupled-cluster amplitudes due to the creation and annihilation operators of the OA-basis, obeying the anticommutation rules

$$\left\{d_p^\dagger, \tilde{d}_q\right\} = \delta_{pq}. \tag{5.65}$$

In the time-dependent case of OATDCC, both the amplitudes and the coefficients are allowed to vary in time,

$$\boldsymbol{\tau}, \boldsymbol{\lambda}, \mathbf{C}, \tilde{\mathbf{C}} \to \boldsymbol{\tau}(t), \boldsymbol{\lambda}(t), \mathbf{C}(t), \tilde{\mathbf{C}}(t) \tag{5.66}$$

With an integrator from `scipy`, our goal is to expose the right-hand side of the equations of motions via the `__call__`-method of the CC-classes as described in section 5.1.2. For OATDCC, a single time-step can be summarized as two computational steps,

[20] Note that in this section latin indices $p, q, r, \ldots$ (also with primes) exclusively label the OA-basis $\{\langle\tilde{\varphi}_p| , |\varphi_p\rangle\}_{p=1}^K$, and similarly the greek indices $\alpha, \beta$ exclusively label the original basis $\{\langle\tilde{\chi}_\alpha| , |\chi_\alpha\rangle\}_{\alpha=1}^L$.

1. Calculate the change in the amplitudes using the amplitude equations, which takes matrix elements from the OA-basis.

2. Calculate the change in the coefficients using the $P$ and $Q$ orbital equations, which takes a mix of matrix elements both the OA-basis and the original basis.

In the doubles approximation OATDCCD, the amplitude equations are the standard CCD equations with matrix elements from the OA-basis. As such, they require the transformed anti-symmetrized Coulomb matrix elements

$$u^{pq}_{rs,AS} = \langle\tilde{\varphi}_p\tilde{\varphi}_q|u|\varphi_r\varphi_s\rangle - \langle\tilde{\varphi}_p\tilde{\varphi}_q|u|\varphi_s\varphi_r\rangle\,. \tag{5.67}$$

The orbital equations are two-fold; the P-space equations only uses anti-symmetrized Coulomb matrix elements of the OA-basis shown above. The Coulomb matrix elements involved in the Q-space equations, on the other hand, are the un-anti-symmetrized variants,

$$u^{pq}_{rs} = \langle\tilde{\varphi}_p\tilde{\varphi}_q|u|\varphi_r\varphi_s\rangle\,. \tag{5.68}$$

The P-space equations involve solving for $\eta^i_a$ and $\eta^a_i$ in the P-space equations, given in eqs. (4.184) and (4.185) as

$$i\hbar A^{ib}_{aj}\eta^j_b = h^j_a\rho^j_i - h^i_b\rho^b_a + \frac{1}{2}u^{pr}_{as,AS}\rho^{is}_{pr} - \frac{1}{2}u^{ir}_{qs,AS}\rho^{qs}_{ar}, \tag{5.69}$$

for the occupied-virtual block of the matrix elements of $\hat{\eta}$ in our basis, and

$$-i\hbar\eta^b_j A^{ja}_{bi} = h^b_i\rho^a_b - h^a_j\rho^j_i + \frac{1}{2}u^{pr}_{is,AS}\rho^{as}_{pr} - \frac{1}{2}u^{ar}_{qs,AS}\rho^{qs}_{ir}, \tag{5.70}$$

for the virtual-occupied block, where

$$A^{ib}_{aj} = \delta^b_a\rho^i_j - \delta^i_j\rho^b_a. \tag{5.71}$$

We are only interested in finding $\eta^i_a$ and $\eta^a_i$, as all other elements are zero in the chosen gauge. To solve this, we first need to calculate the updated amplitudes to find the density matrices $\rho^q_p$ and $\rho^{sr}_{qp}$, where we have listed the non-zero elements in appendix A.1.2. We also need the transformed matrix elements with the current coefficients,

$$h^p_q(t) = \tilde{C}^p_\alpha(t)C^\beta_q(t)h^\alpha_\beta \tag{5.72}$$

$$u^{pq}_{rs,AS}(t) = = \tilde{C}^p_\alpha(t)\tilde{C}^q_\beta(t)C^\gamma_r(t)C^\delta_s(t)(u^{\alpha\beta}_{\gamma\delta} - u^{\alpha\beta}_{\delta\gamma}). \tag{5.73}$$

$$\tag{5.74}$$

The P-space ket equations are then solved by considering composite indices $A = (i, a)$ and $B = (b, j)$, which gives us the left hand side as $A^{ib}_{aj}\eta^i_b = A^A_B\eta^B$, and then solving as a matrix equation for $\eta^B$. The same strategy is used for the bra-equations.

Next we consider the Q-space equations, on the form of eqs. (A.8) and (A.9). The Coulomb contribution to the Q-space equations are summarized in eqs. (A.10) and (A.11) as

$$i\hbar\dot{C}_q^\alpha = i\hbar\eta_q^{p'}C_{p'}^\alpha + C_q^\beta h_\beta^\alpha - h_q^{p'}C_{p'}^\alpha + (\rho^{-1})_q^p\rho_{pr}^{q's}U_{q's}^{\alpha r}, \tag{5.75}$$

$$-i\hbar\dot{\tilde{C}}_\alpha^p = i\hbar\eta_{p'}^p\tilde{C}_\alpha^{p'} + \tilde{C}_\beta^p h_\alpha^\beta - h_{p'}^p\tilde{C}_\alpha^{p'} + (\rho^{-1})_q^p\rho_{p'r}^{qs}\tilde{U}_{\alpha s}^{p'r}. \tag{5.76}$$

All the contribution from the two-body Coulomb matrix elements is summarized in two tensors,

$$U_{q's}^{\alpha r} = (C_{q'}^\beta W_{\beta s}^{\alpha r} - u_{sq'}^{p'r}C_{p'}^\alpha), \tag{5.77}$$

$$\tilde{U}_{\alpha s}^{p'r} = (\tilde{C}_\beta^{p'}W_{\alpha s}^{\beta r} - u_{q's}^{p'r}\tilde{C}_\alpha^{q'}). \tag{5.78}$$

Here, $\hat{W}_s^r$ is the mean field operator,

$$\hat{W}_s^r(x_2) = \int \tilde{\phi}_r(x_1)\hat{u}(x_1,x_2)\phi_s(x_1)\,\mathrm{d}x_1, \tag{5.79}$$

with matrix elements

$$W_{\beta s}^{\alpha q} = \langle\tilde{\chi}_\alpha|\hat{W}_\sigma^\rho|\chi_\beta\rangle\,. \tag{5.80}$$

### Code

We here present the technicalities of our implementations of the Q-space equations for OATCC.

As the first step of the above procedure, we make sure to update the Hamiltonian of the original basis, with the same steps as in TDCC. This is done with the method `update_hamiltonian`, which updates the matrix elements of the instance of the class `QuantumSystem` stored in the `OATDCC`-object.

Next, the steps to calculate all the necessary Coulomb matrix elements with or without a non-zero Q-space have been implemented into a method for the `OATDCC` class called `update_oa_hamiltonian`, seen in **??** 3. The OATDCC method also has a new parameter `has_non_zero_Q_space`, which determines how the Coulomb contribution is calculated and whether the simlified form of the Q-space equations given in eq. (4.191) should be used.

The tensors $U_{qr}^{\alpha p}$ and $\tilde{U}_{\alpha s}^{pq}$ which contain the Coulomb contributions to the Q-space equations are named `u_quart_ket` and `u_quart_bra` in the code, respectively. To save some computational cost, we are careful about the order by which we update the matrix elements $u_{rs}^{pq}$. If the parameter `has_non_zero_Q_space` is false, then we just calculate the elements $u_{rs}^{pq}$ according to regular transformation,

$$u_{rs}^{pq} = \sum_{\alpha\beta\gamma\delta}\tilde{C}_\alpha^p\tilde{C}_\beta^q C_r^\gamma C_s^\delta u_{\gamma\delta}^{\alpha\beta}. \tag{5.81}$$

If it is true, on the other hand, then several intermediate steps are performed. First, the matrix elements of the mean field operator $W_{\beta s}^{\alpha q}$ is calculated, and its contributions are added to the $U_{q's}^{\alpha r}$, $\tilde{U}_{\alpha s}^{p'r}$ This is done using the methods `construct_mean_field_operator`, `contract_W_partially_ket`

**Listing 3** Method to update the matrix elements of OATDCCD.

```
    def update_oa_hamiltonian(self, C, C_tilde=None):
        np = self.np

        h = self.h
        u = self.u

        if C_tilde is None:
            C_tilde = C.T.conj()

        if self.has_non_zero_Q_space:
            self.construct_mean_field_operator(
                u, C, C_tilde, np=self.np, out=self.W
            )

            self.contract_W_partially_ket(
                self.W, C, C_tilde, np=self.np, out=self.u_quart_ket
            )
            self.contract_W_partially_bra(
                self.W, C, C_tilde, np=self.np, out=self.u_quart_bra
            )

            # non-antisymmetrized
            np.einsum(
                "bq,prbs->prqs",
                C,
                self.u_quart_bra,
                optimize=True,
                out=self.u_prime,
            )

            self.u_quart_ket -= np.tensordot(C, self.u_prime, axes=((1), (0)))
            self.u_quart_bra -= np.tensordot(
                self.u_prime, C_tilde, axes=((2), (0))
            ).transpose(0, 1, 3, 2)
        else:
            # non-antisymmetrized
            self.u_prime = self.system.transform_two_body_elements(
                self.u, C=C, C_tilde=C_tilde,
            )

        self.h_prime = self.system.transform_one_body_elements(h, C, C_tilde)
        self.u_prime = self.system._basis_set.anti_symmetrize_u(self.u_prime)
        self.f_prime = self.system.construct_fock_matrix(
            self.h_prime, self.u_prime
        )
```

and `contract_W_partially_ket`. Since the object `u_quart_ket` now contains the matrix elements $C^{\beta}_{q'} W^{\alpha r}_{\beta s}$, we use it to calculate the fully transformed elements $u^{pq}_{rs} = \tilde{C}^{p}_{\beta}(C^{\beta}_{r} W^{\alpha q}_{\beta s})$, before `u_quart_ket` and `u_quart_bra` are completed by subtracting the second terms of eq. (5.77).

We now have all the needed matrix elements. The next two steps are to calculate the updated amplitudes $\tau_\mu$ and $\lambda_\mu$ according to the amplitude equations, and the matrix elements of $\eta^{p}_{q}$ according to the P-space equations. This is done as described in [41, 51]. The update to the coefficients in terms of the P-space contributions is added,

```
    C_update = np.dot(C, eta)
    C_tilde_update = -np.dot(eta, C_tilde)
```

In the case of non-zero Q-space, we then regularize the one-body density matrix according to section 5.3.5. We calculate the exponential of the matrix using the `Python` function `scipy.expm`. Finally, remaining parts of the eqs. (A.10) and (A.11) in terms of the Q-space contribution to the coefficient matrices is added,

```
C_new += -1j * self.compute_q_space_ket_equations(
    C,
    self.h,
    self.h_prime,
    self.u_quart_ket,
    self.rho_qspr,
    rho_inv_pq,
)
C_tilde_new += 1j * self.compute_q_space_bra_equations(
    C_tilde,
    self.h,
    self.h_prime,
    self.u_quart_bra,
    self.rho_qspr,
    rho_inv_pq,
)
```

We have implemented `compute_q_space_ket_equations` and `compute_q_space_bra_equations` as

```
def compute_q_space_ket_equations(
    self, C, h, h_prime, u_quart_ket, rho_qspr, rho_inv_pq
):
    np = self.np

    rhs = np.dot(h, C)
    rhs -= np.dot(C, h_prime)

    temp_ap = np.tensordot(
        u_quart_ket, rho_qspr, axes=((1, 2, 3), (3, 0, 1))
    )
    rhs += np.dot(temp_ap, rho_inv_pq)

    return rhs

def compute_q_space_bra_equations(
    self, C_tilde, h, h_prime, u_quart_bra, rho_qspr, rho_inv_pq
):
```

```
    np = self.np

    rhs = np.dot(C_tilde, h)
    rhs -= np.dot(h_prime, C_tilde)

    temp_qb = np.tensordot(
        rho_qspr, u_quart_bra, axes=((1, 2, 3), (3, 0, 1))
    )
    rhs += np.dot(rho_inv_pq, temp_qb)

    return rhs
```

Note that the lines `np = self.np` is to allow using other numerical frameworks than `numpy`.

### Regularization

In MCTDHF and OATDCC with non-zero Q-space, the orbital equations involve the inverse of the one-body density matrix $\rho_{qp}$, as seen in eqs. (4.186) and (4.187). This object tends to become singular in certain systems, at which point the matrix inversion breaks down. This is a problem common to MCTDHF as well. To fix this, we regularize the one-body density matrix, adding small elements along the diagonal given. We use the exponential regularization given by

$$\rho_{qp} \to \rho_{qp} + \frac{1}{\epsilon} e^{-\epsilon \rho_{qp}}. \tag{5.82}$$

Here $\epsilon$ is a regularization parameter which we set to $\epsilon = 10^{-6}$ to ensure convergence of the equations.

### OATDCCD-DVR

The OATDCCD-DVR method we have implemented utilizes the "2d" representation of `ODSincDVR`. The two-body Coulomb operator is now represented as a 2-dimensional matrix. With our representation of the Q-space equations presented above, the only change needed is the computation of the tensors `u_quart_ket` and `u_quart_bra`, given in eq. (5.77).

### Computational cost of OATDCCD

This section covers the computational cost of one time step in the OATDCCD method. We use $N_b$ as the number of grid points and $L = 2N_b$ and $K$ as the number of basis functions in the grid and OA bases, respectively, $N$ as the number of particles (or occupied states) and $M = K - N$ as the number of virtual states in the OA-basis.

The amplitude equations are equal to a normal coupled-cluster calculations, where the worst scaling term is the non-linear contraction of the amplitudes with the matrix elements $u[k, l, d, c]$ (only dummy indices). There are several terms with equal cost. One of them is

$$\tau_{ik}^{ac} \tau_{kl}^{bd} u_{dc}^{kl} \tag{5.83}$$

which has a scaling of $\mathcal{O}(K^4 M^4)$. Using so-called intermediates, this can be reduced to $\mathcal{O}(K^6)$.

In the $P$-space orbital equations, we are solving for $\eta_i^a$ and $\eta_a^i$, which has a total size of $2KM$. The cost of this operation is maximally $\mathcal{O}((KM)^3)$. The right hand side of the equation is dominated by the four two-body terms, which has a cost of $\mathcal{O}(NMK^3)$.

#### Matrix element transformation

As shown in section 3.2, transformation of matrix elements from the DVR basis is given as

$$u_{rs}^{pq} = \sum_{\alpha\beta} \tilde{C}_\alpha^p \tilde{C}_\beta^q C_r^\alpha C_s^\beta u_{\alpha\beta}^{\alpha\beta}. \tag{5.84}$$

### 5.3.6 Antisymmetrization

Normally in a coupled-cluster calculation one can use the anti-symmetrized matrix elements directly in all calculations. The non-anti-symmetrized matrix elements never appear in any equation, and we are free to only store $u_{pqrs}^{AS} = u_{pqrs} - u_{pqsr}$, saving computational power by removing half of the amplitude terms that would be present in a non-anti-symmetrized treatment of the amplitude equations.

In OA with non-zero Q-space, on the other hand, we need access to the non-anti-symmetrized matrix elements for the Q-space equations. This is a shame, since we are then forced to store twice the amount of matrix elements.

Another problem appears when using a DVR-basis, where anti-symmetrization breaks the symmetry of sparseness. Consider the DVR Coulomb matrix elements $u_{rs}^{pq}$, which can be written in terms of a sparse matrix, $u_{pqrs} = u_{pq}\delta_{pr}\delta_{qs}$, with $u_{pq} = u_{pqrs}$. Antisymmetrization of this gives

$$u_{pqrs}^{AS} = u_{pq}\delta_{pr}\delta_{qs} - u_{pq}\delta_{ps}\delta_{qr}. \tag{5.85}$$

where $u_{pq}^{AS}$ would be the sparse anti-symmetrized Coulomb matrix elements corresponding to $u_{pq}$. This is not possible to write on the form $u_{pq}^{AS}\delta_{pr}\delta_{qs}$. However, we are saved by the fact that the anti-symmetrized matrix elements only appear in the transformed basis, and all expressions involving matrix elements of the DVR basis are un-anti-symmetrized. The solution to both of these problem was to force the system class to be non anti-symmetrized in the OATDCC method.

### 5.3.7 Orbital Adaptive Coupled-Cluster Ground State

The ground state of OACC with non-zero Q-space is notoriously hard to find. There are no existing solvers for it, and the biorthogonality. We have tried two methods to find the ground state of our systems. The first is imaginary time evolution, where we perform a Wick rotation on the governing equations of motion, which has the effect of propagating the system in imaginary time. Due to the structure of the eoms, imaginary time propagation can be shown to be equal to gradient descent towards the ground state. However, this also turned out to be challenging for the biorthogonal states of OATDCC. The main challenge is that the system of equations in OATDCC are poisson and not symplectic, and the structure matrix is unknown. In more physical terms,

the ket and the bra states need to be propagated in opposite time directions, and it is not clear how the Wick rotation should affeect different parts of the parametrization of the OATDCC ansatz.

The second way of finding the OATDCC ground state is using adiabatic switching. Adiabatic switching described in section 2.4.2. Using adiabatic switching for coupled-cluster, we start with a reference based on the single-particle functions and get the initial state by simply setting all amplitudes to zero. The correct amplitudes are then found through time-evolution with the time-dependent two-body operator $F(t)\hat{V}$. This is also the case with OATDCC, but here we also set the initial coefficient matrix to the identity matrix, which corresponds to using the spectral basis as the initial basis.

To be able to benchmark the quality of these solutions, we will compare our results to the root solver of NOCC, which provides a good ground state for OACC with nonzero Q-space.

### 5.3.8 Integration

When integrating in time, we will be solving ordinary differential equations (ODEs) on the form

$$\dot{q}(t) = f(q, t), \tag{5.86}$$

Here $q(t)$ is the function we are solving for.

Symplectic integrators can be derived from the classical Hamiltons principle, which states that the physical solution of the path $q(t)$ for some particle is the one that extremizes the action $\mathcal{S}$,

$$S = \int_{t_0}^{t_N} L(q, \dot{q})\,\mathrm{d}t. \tag{5.87}$$

Here $L(q, \dot{q})$ is the Lagrangian for the system. In the discretized case, this Hamilton's principle is also discretized, and we seek solutions that extremize the discretized action

$$\mathcal{S}_h(\{q_n\}_0^N) = \sum_{n=0}^{N-1} L_h(q_n, q_{n+1}). \tag{5.88}$$

The discretized Lagrangian $L_h(q_n, q_{n+1})$ is here an approximation of the integral of the Lagrangian over a small step $h$,

$$L_h(q_n, q_{n+1}) \approx \int_{t_n}^{t_{n+1}} L(q(t), \dot{q}(t))\,\mathrm{d}t. \tag{5.89}$$

In [18] they show that symplectic methods can be constructed from different ways of approximating this integral, and Gauss integration is the method we get if we use Gauss Legendre quadrature to approximate the integral.

### 5.3.9 Gauss Integrator

For the complicated equations of motion for the amplitudes of TDCC and amplitudes and coefficients of OATDCC, a symplectic integrator is favourable.

This is because the equations of motion of the $\tau_\mu$ and $\lambda_\mu$ amplitudes have the structure of a complex form of the classical Hamilton's equations, as can be seen from eqs. (4.142) and (4.143), with $\tau_\mu$ and $\lambda_\mu$ corresponding to the coordinates and their conjugate momenta, respectively[38].

The Gauss integrator is based on Gauss quadrature, which we covered in section 3.2.1. The idea is to implicitly find the next step by integrating and using a converging series, until the differential equation is satisfied at each quadrature point.

The Gauss integrator is symplectic, and for the integral over a step, each parameter is approximated as a polynomial according to Gauss-Legendre quadrature. It is an implicit approach, where the values at the quadrature is improved iteratively until the ODE is required to be satisfied at each of the quadrature points. This gives a symplectic and reversible integrator[18, 38]. The specific quadrature used is the Gauss-Legendre quadrature, as it has a weight function $\omega(x) = 1$ and is on a domain $\Omega = [-1, 1]$.

## 5.4 Special considerations

### 5.4.1 Absorbing boundary conditions

For a physical bound system, i.e. one that is bound in a well which is not infinitely deep, it is possible to ionize the system with a strong enough laser.

A measure we use on ionization in one dimension is the integral of the one-particle density,

$$I(t) = \int_a^b \mathrm{d}x\, \rho(x, t). \tag{5.90}$$

If the interval $[a, b]$ is such that the it contains the bound parts of the wavefunction, then we can calculate a measure of ionization as $N - I(t)$, where $N$ is the number of particles. This is the case of absorbing boundary conditions, where we are ensured that unbound states will eventually propagate out of the system. For reflecting boundary conditions, on the other hand, some parts of the wavefunction might reflect and reenter the interval above. It is therefore important that the grid is large enough when studying ionization, to allowintervals when studying ionization.

### 5.4.2 Adiabatic switching

Using the time-evolution operator functionality of `QuantumSystems`, it is simple to implement the required time-dependent operators for adiabatic switching. When chosing a switching function, a property we are looking for is that it should start and end as close as possible to 0 and 1, respectively. If the initial value is non-zero, then initially prepared non-interacting state won't be the ground state anymore, and equally if the final value is not unity then the final state won't have become the ground state of the system.

We have tested three different switching functions. The simplest and most naive switching function is just a linear switching,

$$F_0(t) = \begin{cases} 0, & \text{if} \quad t < 2T, \\ t/(2T), & \text{if} \quad 0 < t < 2T, \\ 1, & \text{if} \quad t > 2T. \end{cases} \tag{5.91}$$

where $T$ is the half-time of the switching function. However, this switching function has severe discontinuities in the derivaties, and leads to artifacts in the wavefunction which increase the total energy.

A more sophisticated switching function is the Fermi function[22]

$$F_1(t) = 1 - \frac{1}{1 + e^{(t-T)/\tau}}. \tag{5.92}$$

The deviation of $F_1(t)$ from 0 and 1 at the start and end of the interval, respectively, decays exponentially with the parameter $\tau$.

Another proposal is the error function given by

$$\operatorname{erf} z = \frac{2}{\sqrt{\pi}} \int_0^z \mathrm{d}t\, e^{-t^2}. \tag{5.93}$$

It converges even stronger towards the asymptotal values. We scale this to ensure that it converges towards 0 for $t \to -\infty$ and 1 for $t \to \infty$. The form of the final switching function is then

$$F_2(t) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{t-T}{\tau_2}\right)\right], \tag{5.94}$$

Setting the derivative of $F_1(t)$ and $F_2(t)$ equal in the midpoint gives us the relation $\tau_2 = 4\tau/\pi$, which gives us

$$F_2(t) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\sqrt{\pi}\frac{t-T}{4\tau}\right)\right], \tag{5.95}$$

The three functions can be seen in fig. 5.7.

To implement the switching functions, we use classes that implement the `__call__`-method in `Python`. The classes are set up with the parameters $T$ and $\tau$, such that they then behave like functions of a single parameter $t$. As an example, consider the following snippet

```
from adiabatic_switching.switching_functions import FermiFunction
f = FermiFunction(tau=6, half_time=50)
```

Here `adiabatic_switching` is a convenience `Python` library for the time-propagation of a system through an adiabatic switching process. It is part of the `py-master` repository, described above, and contains three modules

For the `FermiFunction` and `Erf` switching functions, we want to ensure that the slope is equal at the midway point, and as such also implemented the `ErfScaled`, which has equal slope the the `FermiFunction` at $t = 0$.

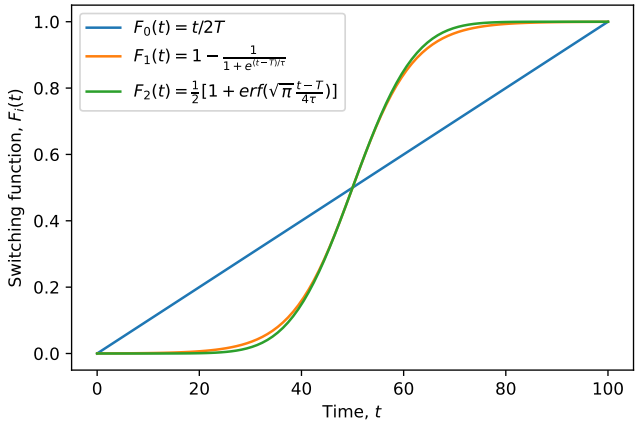



Figure 5.7: The three switching functions we have used, the linear $F_0(t)$, the fermi function $F_1(t)$ and the error function $F_2(t)$. The two latter are scaled such that their derivatives are equal in the midpoint. The midpoint is at $T = 50$, and the exponential parameter is $\tau = 6$. The values at $t = 0$ are $F_0(0) = 0$, $F_1(0) = 2.4 \times 10^{-4}$ and $F_2(0) = 8.8 \times 10^{-8}$.

### 5.4.3 Imaginary time propagation

We implemented a perscription for imaginary time propagation for coupled-cluster in the `coupled-cluster` library, on the branch `imag_time_method`. The time-dependent coupled-cluster base class `TDCC` is modified to accept an argument `time_direction`, which when set to `"imaginary"` modifies the equations to propagate the system in imaginary time. Imaginary time propagation used to find the CCSD ground state for helium in the cc-pVDZ basis is seen in fig. 5.8. Both the difference in energy between two subsequent time steps $|\Delta E_i| = |E_{i+1} - E_i|$ and the deviance from the CCSD ground state at each time step $|E_i - E_{CCSD}|$ is plotted.

#### Non-feasibility of imaginary time-propagation for OATDCC

Imaginary time propagation for the ground state of OATDCCD was also attempted. Kvaal states that imaginary time-progation is infeasible for OATDCC[27]. His main argument is that the energy should be complex analytic and as such the energy is also conserved under imaginary time-propagation. However, there are other challenges to the direct implementation of imaginary time-propagation. Consider that the parametrization of the two wavefunctions $|\Psi\rangle$ and $\langle\Psi|$ are independent. Under bivariational imaginary time-propagation, parameters related to the former should be propagated forward in imaginary time while parameters related to the latter should be propagated backwards. The OATDCCD equations are unlike the TDCCD equations coupled in a non-trivial way, which comes down to the Poisson nature of the equations of motion. Assigning the direction of the wick rotation is as such a non-trivial procedure.

However, for the time-dependent optimized coupled-cluster (TDOCC), which is

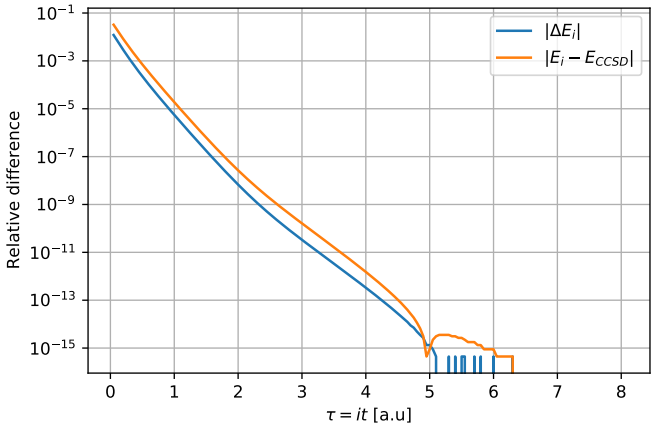



Figure 5.8: Imaginary time propagation for the CCSD ground state of helium.

similar to OATDCC but with orthogonal orbitals, imaginary time relaxation turns out to be feasible[40]. To chose the direction of the Wick rotation for different parameters in the equations of motion of OATDCC, we used a perscription obtained through communication with Sato et al. in the imaginary-time propagation branch of `coupled-cluster`[21]. As seen with a simple numerical experiment in fig. 5.9, the energy does not converge, and the method ends up breaking down. Clearly, OATDCCD requires a more careful treatment to allow imaginary time propagation, if it is indeed possible. If possible, then a more thorough theoretical analysis is needed to identify how the different parameters should be treated.

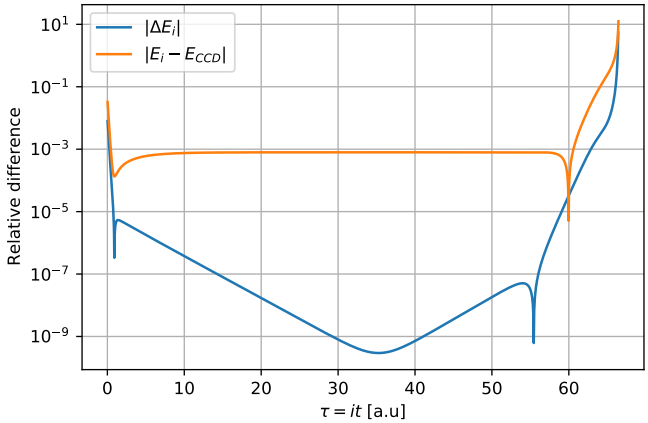



Figure 5.9: Imaginary time propagation for the OATDCCD ground state of helium.

[21] https://github.com/Schoyen/coupled-cluster/tree/imag_time_method

# CHAPTER 6

# Results

The purpose of this thesis project was to implement DVR for OATDCCD. In this chapter, we therefore present the results of this endeavour, split into three sections. The first concerns single-particle problems, where we compare our DVR basis to a basis found with the finite difference method (FDM) using the `QuantumGridSolver`. We compare the quality of the eigenvalues for a single particle in a harmonic oscillator, and then study the difference of the time-evolution of said system under the influence of a time-dependent electromagnetic interaction.

In the second section we present results regarding the Hartree-Fock and OACCD groundstates with a DVR basis. Adiabatic switching is verified on the HF state and is subsequently used to find the OACCD ground state with a non-zero Q-space. We reproduce ground state values calculated with MCTDHF for the one-dimensional atoms of helium, beryllium and carbon.

The third section covers the dynamics of TDHF and OATDCCD, where the one-dimensional atoms of the previous section are subjected to a strong electric field, and we study the both ionization of atoms as well as high harmonic generation.

Additional results for this thesis can be found in Jupyter notebooks in the main github repository[1]. All results in this chapter are presented in atomic units, abbreviated a.u.

## 6.1 Single-particle problem

To test our implementation of the sinc-DVR basis functions, we compare them against the finite difference method (FDM) described in section 3.1.1, which is a simple and commonly used method to solve the one-body Schrödinger equation. We show that DVR gives a good representation of the wavefunction with only a few grid points, making it applicable to even three-dimensional systems.

### 6.1.1 Onebody Ground State

As our single particle ground state problem, we consider the the ground state of a single particle in a harmonic oscillator potential in one dimension. The

[1] https://www.github.com/halvarsu/master

one-body Hamiltonian for a quantum harmonic oscillator system is

$$\hat{h} = -\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}x^2} + \frac{1}{2}\omega^2\hat{x}^2, \tag{6.1}$$

where $\omega$ is the oscillator frequency. This system has an analytic, closed form solution with eigenfunctions

$$\psi_n(x) = C_n H_n(x) e^{-\frac{\omega x^2}{2}}, \tag{6.2}$$

where $H_n(x)$ are the Hermite polynomials and $C_n$ is a normalization factor. The analytical expression for the $n$'th eigenenergy is given by $E_n = \omega(n + 1/2)$. A plot of the seven first wave functions can be found in fig. 6.1.

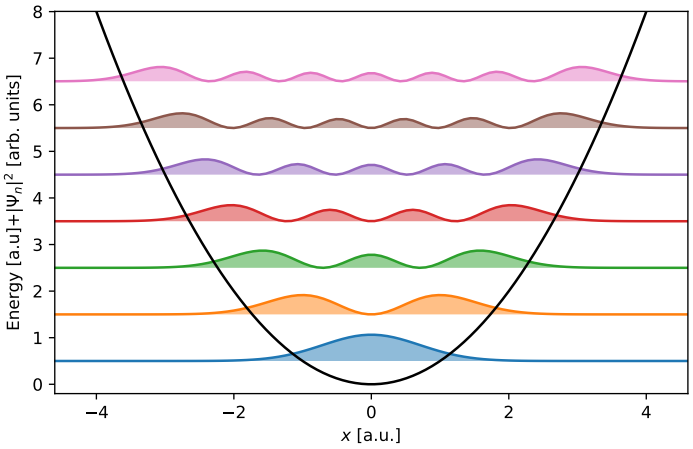



Figure 6.1: The first seven eigenfunction of a harmonic oscillator with $\omega = 1$, given in terms of the hermite polynomials $H_n(x)$. The eigenenergy is added to the corresponding eigenfunction.

We have used both the FDM and the DVR basis to find solutions of this potential. We set up and subsequently diagonalize the Hamiltonian for both grid methods, resulting in representations of the eigenfunctions in discretized position basis, $\psi_n(x_\alpha)$.

The maximal extent of a classical harmonic oscillator with energy $E_n$ is given by the distance to the origin $L$ such that $V(L) = E_n$. Solving for $L$ gives

$$\begin{aligned} \omega(n + \frac{1}{2}) &= \frac{1}{2}\omega^2 L^2 \\ \implies |L| &= \sqrt{\frac{2n+1}{\omega}} \end{aligned}$$

In the quantum case, the wavefunction has non-zero amplitude outside this area, but it is exponentially decaying. Consequently, $L(n)$ gives a lower bound on the grid size needed to represent the $n$'th wavefunction. We also know that the wave function of quantum number $n$ has $n + 1$ equally spaced nodes. We need

at least two grid point per node, and as such $N_{dvr} = 2(n+1)$ is an estimate of a lower bound for the sinc-DVR functions required to $\psi_n$.

For a harmonic oscillator frequency of $\omega = 1$, we test the representation of the $n = 100$ first wavefunctions on a grid of length $L = \sqrt{201} \approx 14.2$ a.u consisting of $N_{dvr} = 201$ grid points. Figure 6.2 shows the relative error of the $n$'th wavefunction calculated with FDM using both our implementation in `QuantumGridSolver` and `ODQD` of the `quantum_systems` library. This is compared to our implementation of sinc-DVR in `ODSincDVR`. We see that while FDM gives a poor representation of the system with such a low number of grid points, the sinc-DVR basis represents the system to machine preciscion for the 60 first wavefunctions.

In figs. 6.3 and 6.4 depict results from the same calculation as above for multiple times for grid lengths between 5 and 20. Again we measure the relative error in the energy of the both the FDM and the DVR basis functions as a function of quantum number $n$. In fig. 6.4 we also plot the classically allowed area $L(n)$, and see that the error in the DVR wavefunction is only due to the lack of grid representation.

While the grid solver needs many grid points per node of the wavefunction for a proper representation, the DVR performs very well for grids larger than the classically allowed area, which we have plotted in fig. 6.4. The difference between the two methods is due to the poor representation of the kinetic energy operator in the finite difference scheme.

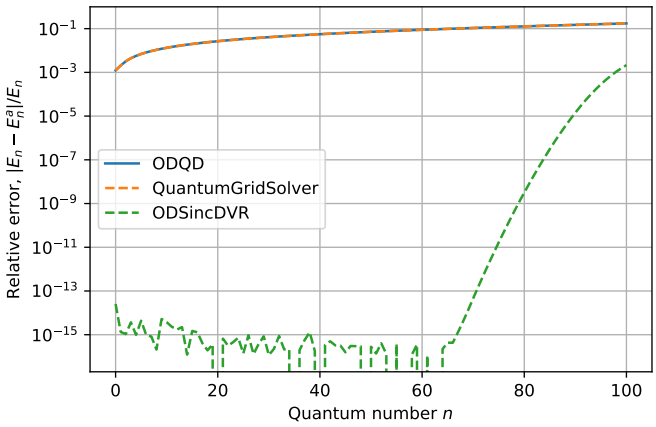



Figure 6.2: Relative error of the energy of the $n_{max} = 100$ first wave functions of a harmonic oscillator with $\omega = 1$ a.u. Calculated using a sinc-DVR representation with $N_{dvr} = 2(n_{max} + 1)$ and a grid length of $L = \sqrt{(2n_{max} + 1)/\omega}$.

### DVR quality with spacing

With a low number of quadrature points to represent a wavefunction, the quality of the fit of the sinc-DVR basis is sensitive to the specific placement of the grid

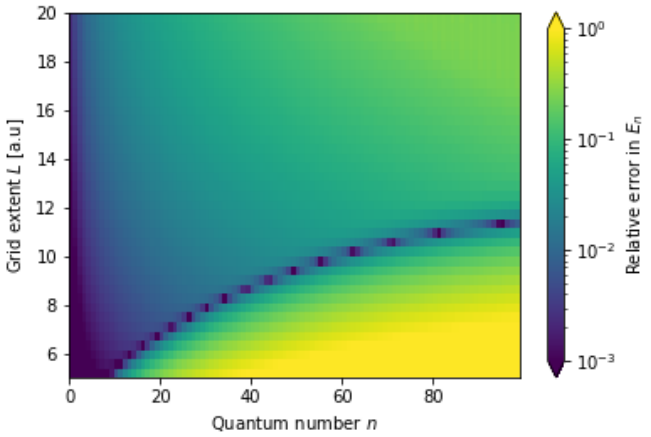



Figure 6.3: Relative error in the numerical energy of the finite difference method for the harmonic oscillator, compared with the analytical energy $E_n = 0.5 + n$ a.u. for $\omega = 1$ a.u. The logarithmic value is taken for the 100 first wavefunctions for a fixed number of grid points $N_{grid}$ with varying grid extent $L$. The number of grid points is fixed to 201.

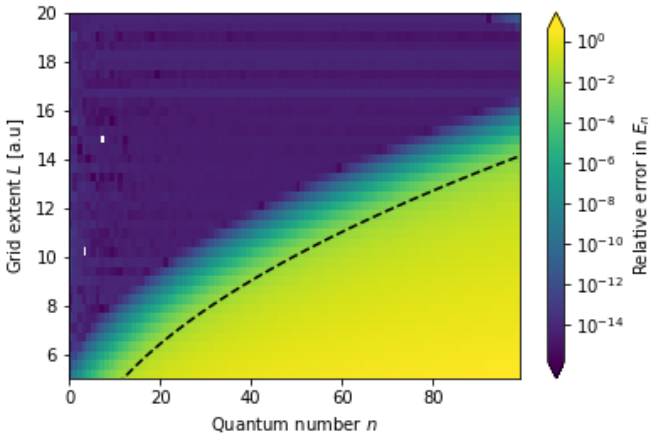



Figure 6.4: Relative error in the numerical energy of a DVR basis set for the harmonic oscillator, compared with the analytical energy $E_n = 0.5 + n$ a.u. for $\omega = 1$ a.u. The logarithmic value is taken for the 100 first wavefunctions for a fixed number of grid points $N_{grid}$ with varying grid extent $L$. The number of grid points is fixed to 201, and the classically allowed area for the harmonic oscillator is plotted as a dashed line.

points. Using a sinc-DVR basis for the one-dimensional atom model with $N_{dvr}$ basis points, we construct the wave function for the ground state from, using the spatial representation of the sinc-DVR basis functions. Figure 6.5 shows this for the 1d beryllium-atom, given by a shielded coulomb,

$$\hat{V}(x) = \frac{Z_a}{\sqrt{x^2 + c^2}}, \tag{6.3}$$

with $Z_a = 4$ corresponding to the Beryllium atom. We use $c = 1$ as shielding parameter. It is clear that the placement of the grid points is important for the representation of the wavefunction. It also shows the non-variational nature of this problem; more points does not necessarily give a better description. This is not relevant for our results, where the number of grid points completely eliminate this source of error, but it can be relevant when using DVR in higher dimensions, where the number of grid points is more limited. Consider 2000 grid points, which in one dimension this gives a very high quality description of the wavefunction, but in three dimensions correspond to a cube with 13 grid points per dimension. A possible way to overcome this error is to use extrapolation to infinitesimal grid spacings, such as the one used in [24].

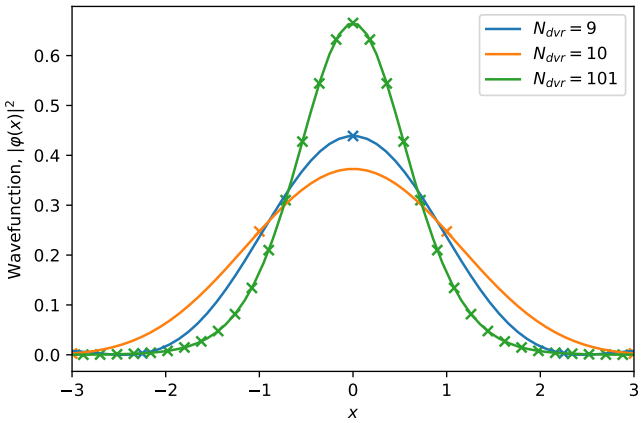



Figure 6.5: DVR basis functions for the one dimensional beryllium atom. The quality of the DVR-representation is highly dependent on placement of the grid points, and is not necessarily monotonic increasing with the number of grid points.

### 6.1.2 Driven harmonic oscillator

A simple dynamics problem for the one-dimensional harmonic oscillator is a sinusoidal dipole laser. The dipole laser interaction is given as a one-body operator,

$$h_I(t) = F_0 \sin(\Omega t)\hat{x}, \tag{6.4}$$

where $F_0$ is the laser amplitude and $\Omega$ is the laser frequency. We set the laser frequency and the harmonic oscillator frequencies equal to $\Omega = \omega = 0.5$ a.u. When they correspond, the system is resonant and quickly increases in energy.

We numerically set up the ground state of the system using a grid length of 20 a.u. and a grid spacing of $\Delta x = 0.1$ a.u. for both the sinc-DVR and the finite difference method. The Crank-Nicholson method is used to propagate both the DVR- and the grid systems. We apply the sinusoidal laser for a period of $T_{period} = 2\pi/E_0 \approx 25.13$ a.u, without an envelope, and with an amplitude of $F_0 = 1$ a.u.. This is not a realistic laser pulse [2], but its purpose is to excite the system so that we can test the one-body systems and the time-propagation algorithms. After the laser pulse, we propagate the system without an external interaction until $T = 1000$ a.u., corresponding to almost 80 periods. The time-step is set to $\Delta t = 0.01$ a.u.

A corollary of the harmonic potential theorem[12] states that a dipole interaction onto a harmonic oscillator induces rigid translations. As seen in fig. 6.6, both methods show this phenomena for short times, but the FD wavefunction slowly deteriorates with time. A possible explanation is that the DVR gives a better representation of the matrix elements of the harmonic oscillator, and therefore also conserves its properties better.

The energy is conserved for both basis sets, as can be seen in **??** in appendix A. A Fourier transform of the induced dipole moment, calculated as the expectation value of the position operator $d(t) = \langle\phi|x(t)|\phi\rangle$ can be seen in fig. 6.7. It shows the dipole allowed transition energies in the system, which for the HO is $E_{n+1} - E_n = \omega$. The FD representation has a slightly lower frequency than the DVR.

The notebook `Onebody Harmonic Oscillator Time Evolution.ipynb`[3] reproduces the results of this section.

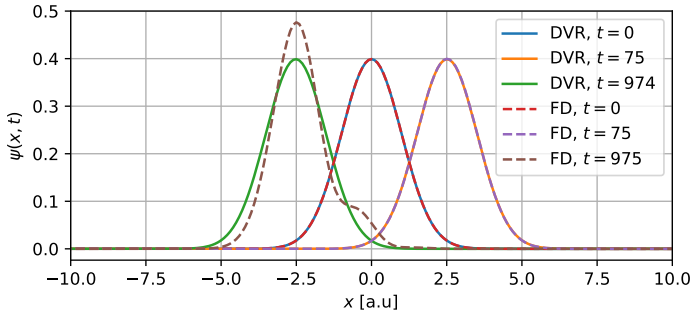



Figure 6.6: Snapshots of the wavefunction for a single particle harmonic oscillator. The DVR basis shows more stable rigid translation over time.

## 6.2 Many-body ground states

Having verified our DVR implementation for the harmonic oscillator, we now turn our attention to systems of interacting electrons. We compare the ground

[2] Realistic laser pulses have a ramp up and ramp down at both ends.

[3] https://github.com/halvarsu/master/blob/master/notebooks/Onebody Harmonic Oscillator Time Evolution.ipynb

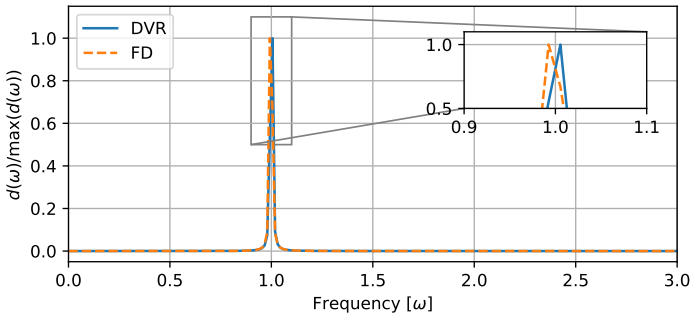



Figure 6.7: Normalized fast Fourier transform of the induced dipole moment $d(t)$ for the harmonic oscillator system described above, using the finite difference method and the DVR basis set. The frequency is in units of the system's oscillator frequency $\omega$.

state of many-body systems using our DVR basis for both the Hartree-Fock and the orbital adaptive coupled-cluster doubles (OACCD) groundstates.

### 6.2.1 Adiabatic switching

To test our adiabatic switching operators, we use the adiabatic switching process to find the RHF ground state for one-dimensional atoms. To avoid the divergence of the coulomb potential, the one-dimensional atomic potential is given by the shielded Coulomb potential

$$\hat{V}(x) = \frac{Z_a}{\sqrt{x^2 + c^2}}. \tag{6.5}$$

Here $c$ is the shielding parameter, and $Z_a$ the strength parameter or atomic number. In this section, we consider the one-dimensional beryllium atom with $Z_a = 4$, and use the shielding parameter $c = 1$.

To simulate the system, we first set up a `ODSincDVR` basis set using 64 DVR functions, which serves as input to a `SpatialOrbitalSystem` with $n = 4$ electrons. We then used our `RHF` implementation to solve the Roothan-Hall equations with a tolerance of $\epsilon = 10^{-12}$, giving an RHF energy of $E_{RH} = -6.73944772244$ a.u. to use as a reference[4].

We initially set the Hartree-Fock coefficients to coefficients of the single-particle solutions of the Hamiltonian. In other words, we set them to the unitary matrix $C_0$ that diagonalizes the one-body DVR Hamiltonian,

$$(C_0^\dagger)_\alpha^p h_\beta^\alpha (C_0)_q^\beta = \delta_{pq}\epsilon_p, \tag{6.6}$$

where $\epsilon_p$ are the single-particle energies of the system. These are then evolved in time using the Crank-Nicholson method. During the adiabatic switching process, we adjust the magnitude of the two-body operator according to the switching function $F(t)$ at each time step, $\hat{u}(t) = F(t)\hat{u}$. For $F(t)$ we use

[4]This is equal to the value stated in [34] up to the digits given there.

the three functions `LinearSwitching`, `ErfScaled` and `FermiFunction`, all described in section 5.4.2.

In fig. 6.8, we compare the results of the three switching functions for three different base values of the decay parameter $\tau$. Each data point constitutes a new simulation for a different final time, where the half-time and decay parameters of the switching functions are updated proportionally, such that the value of the switching function in the initial and final times remains the same.

Evident from the results, the linear switching functions performs poorly for all values of the final time, and converges very slowly towards the Roothan-Hall solution. The error function and the Fermi function perform similarly for steep switching processes given by a small $\tau$ or small $T$. For gentler processes, the value of the start and end-points become significantly different from 0 and 1, respectively. In this case the error function outperforms the Fermi function due to the lower deviance for the expected values.

Additionally, we see that the convergence of the final state of the adiabatic switching process towards the ground state is exponential with the time length of the switching when the values in the end points are negligible.

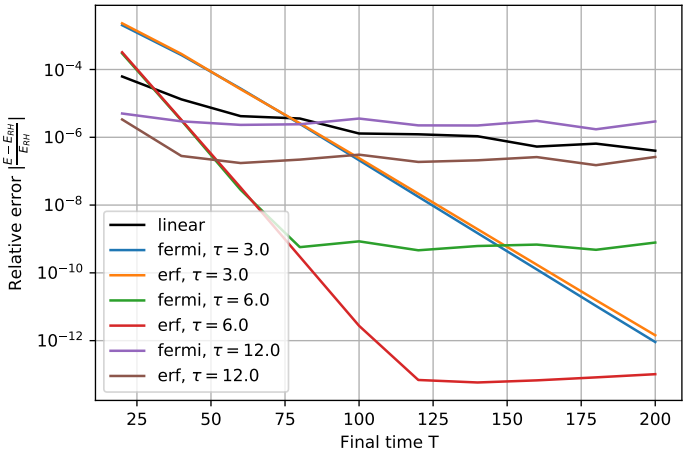



Figure 6.8: Erf vs fermi

### 6.2.2 OACCD Ground state energies

The one-dimensional atomic model has been studied extensively. In this section, we compare the adiabatic switching process as a means to finding the OACCD ground state with non-zero Q-space. We compare our OACCD energies for the one-dimensional atoms with those obtained with Multiconfigurational Time-Dependent Hartree-Fock (MCTDHF)[23, 34].

Using the one-dimensional atomic potential, we set up $N_{DVR} = 256$ sinc-DVR grid basis functions on a grid of extent 25 a.u., for an atomic potential with the default values $(c = 1, a = 1, Z = n)$. This is done for one-dimensional helium, beryllium and carbon, i.e. $n = 2, 4$ and 6, respectively. We find the

ground state with an adiabatic switching simulation. The adiabatic switching operator used is the Fermi function described in **??**, with parameters $\tau = 6$, $T_{half} = 50$. The system is integrated in time from $t = 0$ to $t = 100$. After such an adiabatic switching calculation, the interaction matrix elements $u$ are not restored fully to max since we stop the calculation after a finite time $T$, and we get $F(100) = 0.99976$ with the stated parameters. To get the correct energy for the final state, we are careful to reset $F(T)u \to u$ before calculating the OATDCCD energy.

The resulting energies can be seen in tables 6.1 to 6.3, for different values of the OA basis size $K$. For $n = 2$ electrons, OATDCCD is equal to MCTDHF, and we expect the ground state energy to be equal for the two methods[27]. Comparing table 6.1 with results for the MCTDHF ground state calculated with imaginary time propagation, we see that this is indeed the case[23].

For $N > 2$, OATDCCD is an approximation to MCTDHF. In tables 6.2 and 6.3, we compare our results for beryllium and carbon with those of Miyagi and Madsen [34]. For $K = N$, OATDCCD reduces to TDHF[27], and we use adiabatic switching for TDHF to calculate the value listed as HF. We see that our energies are close to the MCTDHF energies, especially for beryllium. However, note the discrepancy of the HF energies in both cases, which is even more significant in the case of carbon.

As the systems gets larger, the adiabatic switching process must be slower for the system to stay in the ground state. We consider the total correlation energy to be the difference in energy between MCTDHF and converged HF, which for carbon is $\Delta E = E_{\mathrm{MCTDHF}} - E_{\mathrm{HF}} = -0.10037$ a.u[34]. Given that our OATDCCD ground state converges toward a value of $E_{\mathrm{OATDCCD}} = -13.32024$, we recover 89% of the correlation energy. The discrepancy in the adiabatic switching energy for HF compared to the Roothan-Hall converged state is 5% of this energy. Assuming that this discrepancy is the same for the OATDCCD ground state, then OATDCCD recovers between 90% and 95% of the MCTDHF energy.

Table 6.1: OATDCCD ground state energy (in atomic units) of the one-dimensional helium atom ($Z = N = 2$), calculated with adiabatic switching. The case $K = 2$ is the RHF ground state energy calculated with adiabatic switching. We compare with the MCTDHF ground state energies of Hochstuhl et al. [23]. Note that the energies are converging toward the exact energy.

| K | $E_{\mathrm{OATDCCD}}$ | $E_{\mathrm{MCTDHF}}$ |
|---|---|---|
| HF | -2.2242 | -2.2242 |
| 4 | -2.2365 | -2.2365 |
| 6 | -2.2381 | -2.2381 |
| 8 | -2.2382 | -2.2382 |
| 10 | -2.23825 | -2.23825 |
| Exact | | -2.23826 |

Table 6.2: OATDCCD ground state energy (in atomic units) of the one-dimensional beryllium atom ($Z = N = 4$), calculated with adiabatic switching. The case $K = 4$ is the RHF ground state energy calculated with adiabatic switching. We compare with the MCTDHF ground state of Miyagi and Madsen [34].

| K | $E_{\text{OATDCCD}}$ | $E_{\text{MCTDHF}}$ |
|---|---|---|
| HF | -6.739442 | -6.739450 |
| 6 | -6.771284 | -6.771296 |
| 8 | -6.779852 | -6.780026 |
| 16 | -6.784570 | -6.785041 |
| 24 | -6.784598 | -6.785077 |
| 32 | -6.784599 | -6.785078 |
| 40 | -6.784600 | -6.785078 |

Table 6.3: OATDCCD ground state energy (in atomic units) of the one-dimensional carbon atom ($Z = N = 6$), calculated with adiabatic switching. The case $K = 6$ is the RHF ground state energy calculated with adiabatic switching. We compare with the MCTDHF ground state of Miyagi and Madsen [34].

| K | $E_{\text{OATDCCD}}$ | $E_{\text{MCTDHF}}$ |
|---|---|---|
| HF | -13.22568 | -13.23117 |
| 8 | -13.29216 | -13.29860 |
| 10 | -13.30422 | -13.31127 |
| 12 | -13.31236 | -13.32016 |
| 16 | -13.31922 | -13.33009 |
| 20 | -13.32012 | -13.33133 |
| 24 | -13.32023 | -13.33151 |
| 28 | -13.32024 | -13.33154 |

## 6.3 Many-body dynamics

Using TDHF we are able to simulate ionization directly using boundary conditions. We use the system given in [34], which is a one-dimensional beryllium atom interacting with a dipole laser. The laser is given by a vector potential on the form

$$A(t) = \frac{F_0}{\omega} \sin^2(\frac{\omega t}{T}) \sin \omega t, \tag{6.7}$$

with frequency $\omega = 0.0570$ a.u, corresponding to a wave length of 800 nm. The field is active for three cycles, $T = 3\frac{2\pi}{\omega} = 331$ a.u. This gives a one-body interaction term of $F(t)\hat{x}$, where $F(t)$ is the electric field defined as

$$\begin{aligned} F(t) &\equiv -\frac{\partial A(t)}{\partial t} \\ &= \Big[ \sin^2\left(\frac{\pi t}{T}\right) \cos(\omega t) + \frac{2\pi}{T\omega} \sin\left(\frac{\pi t}{T}\right) \cos\left(\frac{\pi t}{T}\right) \sin(\omega t) \Big] \hat{x} \Theta(T-t) \Theta(t), \end{aligned}$$

where $\Theta(t)$ is the Heaviside function. The strength is set to $F_0 = 0.0755$ a.u.

We implemented two types of boundary conditions, the first one using a complex absorbing potential (CAP) $W(x)$. It is given as a non-hermitian one-body contribution to the wavefunction,

$$\hat{H} = \hat{h} + \hat{v}(x) - iW(x), \tag{6.8}$$

where $W(x)$ is some function that is zero for most of the domain, and then increases smoothly towards the edges. For $W(x)$, we used the function defined in [35],

$$W(x) = 1 - \cos\left(\frac{\pi(|x| - x_{cap})}{2(L - x_{cap})}\right), \tag{6.9}$$

with $x_{cap} = 250$ a.u.

As an alternative to CAP, we also implemented a real masking function. The coefficients of the grid points are then multiplied by a mask $m(x)$ at each time step to remove the electron density at the edges[50]. With the DVR basis, this is equal to performing the following operation at each time step,

$$C_p^\alpha(t) \to C_p^\alpha(t) m(x_\alpha). \tag{6.10}$$

Contrary to the CAP $W(x)$, the mask $m(x)$ is equal to 1 inside of $x_{cap}$, and goes smoothly towards 0 outside. We use a mask given by

$$m(x) = \cos\left(\frac{\pi(|x| - x_{cap})}{2(L - x_{cap})}\right)^{1/4}. \tag{6.11}$$

To validate the use of these absorbing boundary conditions, we first set up a DVR with $N_{DVR} = 256$ grid points, for a grid size of $L = 60$ a.u.. We then solve for the RHF ground state of the beryllium atom with $N = Z_a = 4$ and shielding parameters $c = a = 1$ for the nuclear and two-body potentials. Using the above laser parameters, we simulate for 6 cycles of the laser frequency. This is done for the three cases of regular case of reflecting boundary, CAP and real masking function. In fig. 6.9 we plot the overlap of the RHF state $|\Phi(t)\rangle$ with the initial state $\langle\Phi(0)|$, calculated as

$$\langle\Phi(t)|\Phi(0)\rangle = \det\left[(C^\dagger)_\alpha^i(t) C_j^\alpha(0)\right]^4. \tag{6.12}$$

We see that the result closely match that of [35]. Also, both the absorbing boundary conditions give close results, which validates the implementation of both methods.

### 6.3.1 OATDCCD

Using the OATDCCD ground state of the one-dimensional atomic system as a starting point, we performed dynamics simulations, with the goal of studying the phenomena of ionization and HHG. During the implementation process, implementing boundary conditions for OATDCCD turned out to be a challenge. In the case of the CAP, the non-hermiticity of $-iW(x)$ causes different behaviour for the right and left states $|\Psi\rangle$ and $\langle\Psi|$, which are independent in the bivariational description. The non-hermitian term has opposite effects on time-evolution of the two wavefunctions, where it will either reduce or increase

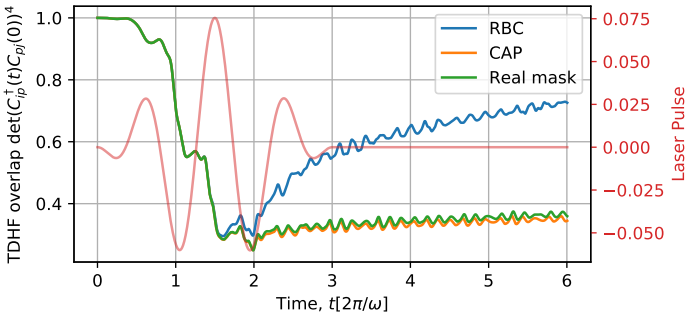



Figure 6.9: TDHF overlap with the initial state for a 1d beryllium atom subject to a laser with electric field strength $F_0 = 0.0755$ a.u. and frequency $\omega = 0.057$ a.u., for three types of boundary conditions: one reflecting and two absorbing described in the text.

the norm depending on the sign of the time . This is a similar challenge to the one of imaginary time propagation for OATDCC described in section 5.4.3. The masking function $m(x)$ is also unavailable for OATDCCD, because the loss of norm breaks biorthonormality of the OA orbitals, which in turn breaks the amplitude equations indirectly by rendering Wick's theorem invalid. To prevent reflection of the wavefunction affecting the observables of interest, we instead made sure to make the grid large enough. What large enough means depends of course on the strength of the laser field and the time of the simulation.

### One-dimensional Helium

For the one-dimensional helium atom we use the parameters of [34]. We set up a sinc-DVR basis of $N_{DVR} = 256$ sinc-DVR basis functions in the domain $[-25, 25]$, for an atomic potential with the default values $(c = 1, a = 1, Z = 2)$. This is a sufficiently large grid to represent the ground state, which we find with adiabatic switching. To get the full size of $L = 300$ used in the paper cited above, prior to simulating the dynamics of the system we add 896 grid points to each, for a total of 2048 grid points.

### One-dimensional Beryllium

Using values for the laser obtained from correspondance with Takeshi Sato, we simulate the one-dimensional beryllium system. He uses a strong electric field strength of $F_0 = 0.1194$ a.u., which is enough to excite the system strongly.

Snapshots of the particle density can be seen in fig. 6.10, with the laser waveform inset showing the corresponding snapshot times[5]. Evident from the figure is that each subsequent cycle increases the spatial frequency of the electrons.

To measure ionization, we integrate the particle density $\rho(x)$ over an area which is far enough away from the nucleus. However, as the integral of $\rho(x)$ over

[5]An animation of the wavefunction can be seen on the main github page of this thesishttps://github.com/halvarsu/master

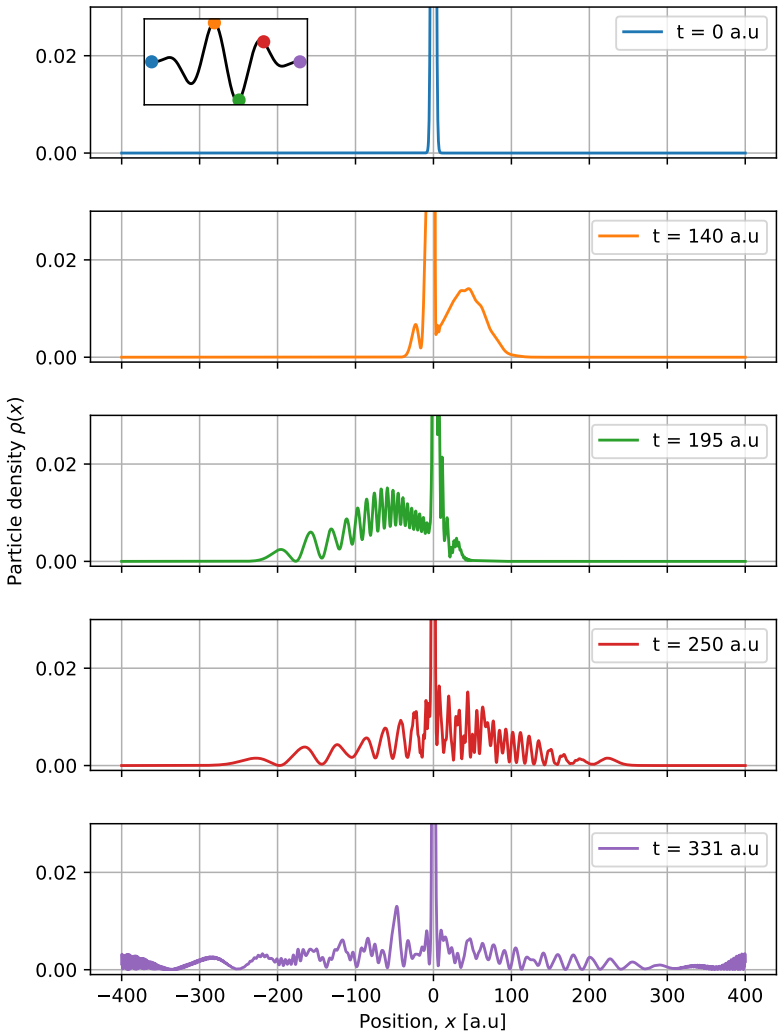



Figure 6.10: Snapshots of the OATDCCD particle density for a one-dimensional beryllium atom subjected to an intense laser field. The inset figure shows the waveform of the laser pulse with the times of the snapshots marked.

the whole grid equals the number of particles, in this case 4, we can instead integrate over just the atomic nucleus. We thus define ionization as

$$i(t) = 4 - \int_{x_{cap}}^{x_{cap}} \mathrm{d}x\, \rho(x). \tag{6.13}$$

Using $x_{cap} = 25$, we plot the ionization of the current system in fig. 6.11. We see that the present laser completely disrupts the atom, and there is a significant chance of more than a single electron being expelled.

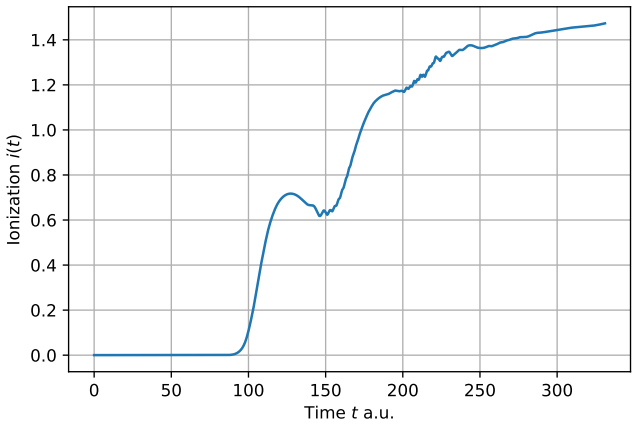



Figure 6.11: Ionization of the 1d beryllium atom subject to a strong electromagnetic field. Ionization is defined as the integral over the electron density that is away from the nucleus by a distance of 25 a.u..

### 6.3.2 High Harmonic Generation

An interesting effect in the study of quantum systems subject to electromagnetic fields is high harmonic generation (HHG). It was discovered in 1987[33], where gas atoms in intense laser fields emitted radiation of frequency of many times the laser frequency, specifically frequencies given by high harmonics of the laser frequency. It can also be observed from short-pulse lasers. A qualitative understanding of the phenomenon is that electrons expelled from the nucleus in the start of one laser cycle, are pushed back into the core when the electromagnetic field switches sign on the second half of the laser cycle. Collision with the remaining core electrons subsequently generate high frequency photons.

A common theoretical approach is to use the single active electron approximation, where the motion of a single electron is studied[16]. However, this fails to reproduce the many-electron contribution to the HHG spectrum[45]. Numerical ab initio methods have lately been used to describe more of the observed HHG spectrum than single active electron methods have been able to, as shown by Miyagi and Madsen [34]. In their paper, they use the time-dependent restriced-active-space self-consistent field method, which is an approximation to the

MCTDHF method. We here test the OATDCCD method for a similar purpose, and are able to qualitatively reproduce high-harmonic spectra.

To measure the HHG spectrum, we use the dipole acceleration of the electron. It can be given in two forms, either in velocity form,

$$P_v(t) = \frac{\mathrm{d}}{\mathrm{d}t} \langle \Psi(t) | \hat{\mathbf{p}} + \mathbf{A}(t) | \Psi(t) \rangle \,, \tag{6.14}$$

or length form,

$$P_r(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} \langle \Psi(t) | \hat{\mathbf{r}} | \Psi(t) \rangle \,. \tag{6.15}$$

Above, $\mathbf{p}$ is the momentum operator, and $\mathbf{A}$ is the vector potential of the electric field, which we will drop as it does not contribute to the HHG spectrum. For the exact wavefunction, the two forms are equal, as can be shown using commutator relations. However, these relations do not hold in the approximate case, and as such they give different results. Which form to use depends on what properties are of interest. Where the length form depends on the quality of the electron distribution in space, the velocity form depends on the momentum distribution only[25].

Applying Ehrenfests theorem, we can rewrite the velocity form as

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle \Psi(t) | \hat{\mathbf{p}} | \Psi(t) \rangle = \langle \Psi(t) | -\frac{\mathrm{d}\hat{V}(x)}{\mathrm{d}x} | \Psi(t) \rangle \,, \tag{6.16}$$

where we have written the derivative of the potential operator as an operator in itself. The one-body matrix elements of this operator in second quantized form are

$$a_q^p(t) \equiv \langle \phi_p(t) | -\frac{\mathrm{d}\hat{V}(x)}{\mathrm{d}x} | \phi_q(t) \rangle = \tilde{C}_\alpha^p(t) C_q^\beta(t) \langle \chi_\alpha | -\frac{\mathrm{d}\hat{V}(x)}{\mathrm{d}x} | \chi_\beta \rangle \tag{6.17}$$

$$= \tilde{C}_\alpha^p(t) C_q^\beta(t) a_\beta^\alpha, \tag{6.18}$$

where we rewrote in terms of the original time-independent basis $|\chi_\alpha\rangle$ and the OA coefficients $\tilde{C}_\alpha^p(t)$ and $C_p^\alpha(t)$. The dipole acceleration in the velocity form is then calculated using the one-body density matrix $\rho_p^q$ as

$$P_v(t) = \rho_p^q(t) a_q^p(t) = \rho_p^q(t) \tilde{C}_\alpha^p(t) C_q^\beta(t) a_\beta^\alpha. \tag{6.19}$$

To measure the dipole acceleration in the length form, we measure instead the dipole moment, given as the expectation value of the position operator,

$$d(t) = \langle \Psi(t) | \hat{x} | \Psi(t) \rangle = \rho_p^q(t) d_q^p = \rho_p^q(t) \tilde{C}_\alpha^p(t) C_q^\beta(t) d_\beta^\alpha. \tag{6.20}$$

We then calculate the numerical second derivative of this with respect to time to get the dipole acceleration in the length form,

$$P_x(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} d(t). \tag{6.21}$$

In the sinc-DVR basis for the one-dimensional atomic model, the matrix elements of the acceleration operator $a_\beta^\alpha$ are given as

$$a_\beta^\alpha = \left( -\frac{\mathrm{d}\hat{V}(x)}{\mathrm{d}x} \right)_\beta^\alpha = \delta_{\alpha\beta} \left( Z_a \frac{x}{(x^2 + c^2)^{3/2}} \right) \Bigg|_{x=x_\alpha} = \delta_{\alpha\beta} \frac{x_\alpha Z_a}{(x_\alpha^2 + c^2)^{3/2}}, \tag{6.22}$$

where we used the DVR basis to evaluate the local operator directly at the grid points.

The induced dipole moment $d(t)$ of the above dynamics simulation for one-dimensional beryllium can be seen in fig. 6.12. This is a severe dipole moment, due to the powerful electric field of $F_0 = 0.1194$ a.u., corresponding to $5 \times 10^{14}$ W/cm$^2$.

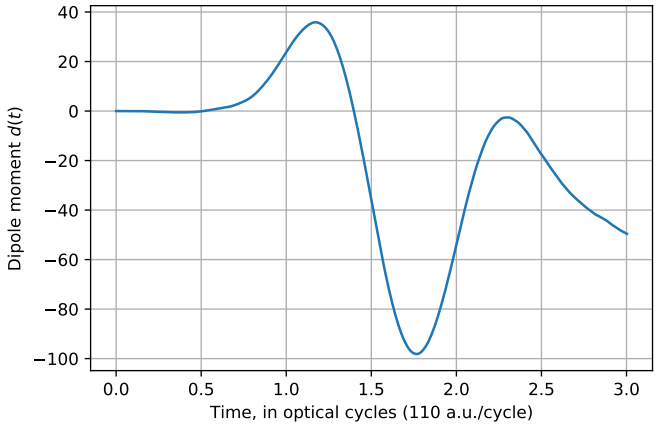



Figure 6.12: Strong dipole oscillations of the one-dimensional beryllium atom, for an electric field strength of $F_0 = 0.1194$ a.u.

To find the dipole acceleration, we use both the velocity form described above, and the length form. For the latter we compute the second derivative of the dipole moment numerically. As seen in fig. 6.10, there are some reflections at the boundaries towards the end of the simulation. To remove any artifacts resulting from this, we calculate the Fourier transform of particle density only for times in the interval $t \in [0, T_{stop}]$, and compare two values of $T_{stop} = 270, 331$ a.u..

We also calculate an expected value for the first cutoff in the harmonic spectrum. The first cutoff is given as $3U_p + I_p^{(1)} = 63\omega$ according to Lewenstein theory[29], where $U_p = F_0^2/(4\omega^2) = 1.097$ a.u is the ponderomotive energy, i.e. the mean energy of an electron oscillating in the given electric field, and $I_p^{(1)} = 0.3123$ a.u. is the first ionization potential for the atom, calculated in [35] using the energy of the highest occupied orbital in the HF approximation. It is, however, based on a single active electron-theory, and does not incorporate many-body effects.

The Fourier spectrum of the dipole acceleration is shown in length form in fig. 6.13, calculated for the two different values of $T_{stop}$. The plateau corresponding to the first cutoff can be seen clearly, though in a slightly higher energy than expected. One explanation is the low value of $K$, which brings it closer to the TDHF solution. The many-body interactions could serve to distribute the kinetic energy between the electrons such that the maximal energy per electron is lower, and a poor many-body description can then lead to higher cutoff energies.

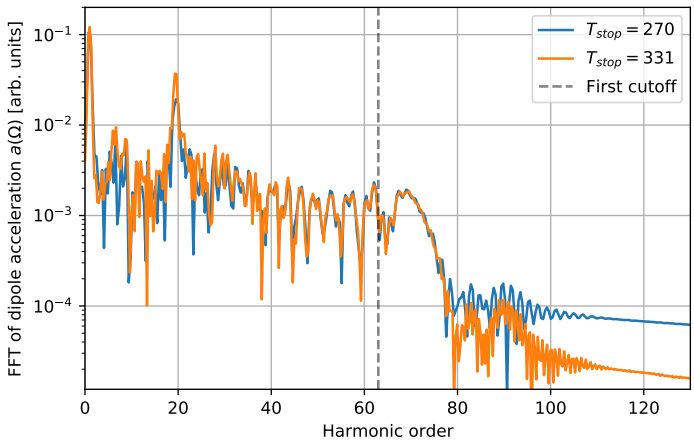



Figure 6.13: The HHG spectrum of one-dimensional beryllium, calculated in the length form as the numerical second derivative of the dipole moment. The first cutoff calculated using semi-empirical considerations is plotted, and roughly matches the expected plateau.

The reflections are seen to not affect the main outcome. This is expected since the dipole acceleration should be relatively small towards the end of the third laser cycle for the electron density far away from the nucleus.

In fig. 6.15, we compare the FFT of the velocity and length forms of the dipole acceleration, where the Fourier transform of the entire signal for $t \in [0, 331\text{a.u.}]$ is used. We find an exact correspondance between the two forms. We speculate that this is due to the gauge invariance of the OATDCC method, inherited from the closely related NOCC method[37]. We also did a wavelet analysis of the dipole acceleration signal, where the signal is convolved with wavelets that pick out different frequencies. Using $K = 32$, we plot the log of the absolute value of the wavelet transform in fig. 6.14. The HHG spectrum is seen to be emitted during the second and at the start of the third cycle, during snapshots two to four of fig. 6.10. The classical calculated cutoff was calculated to harmonic order $66\,\omega$, but we here observe even higher frequencies at the start of the second cycle, when the large lump of particle density of the third snapshot above hits the core again.

Unfortunately, we did not have the time to run more simulations for different $K$ values to study the convergence with $K$ for beryllium, and we did not have the time to study the carbon atom subject to an intense laser field either. The focus in this thesis has not been optimization of the implementation, and consequently the simulations take a long time to run, especially for extended grid sizes. Nonetheless, we have been able to produce the qualitative results, such as the HHG spectrum and the first cutoff.

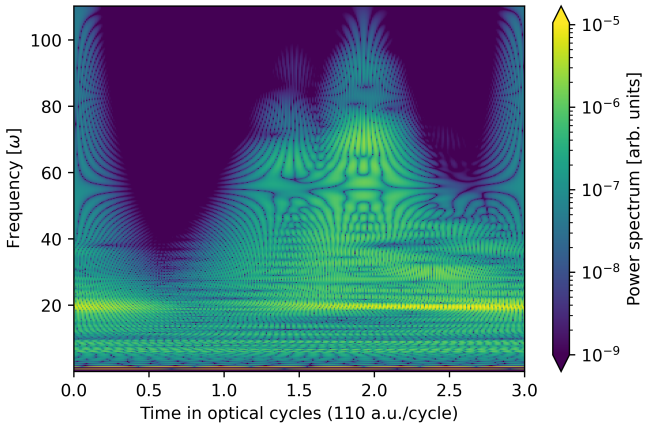



Figure 6.14: Wavelet transform of the dipole acceleration in length form, for a one-dimensional Beryllium atom subject to a laser pulse of three optical cycles. Note that the edge effects is an artifact of the wavelet transformation.

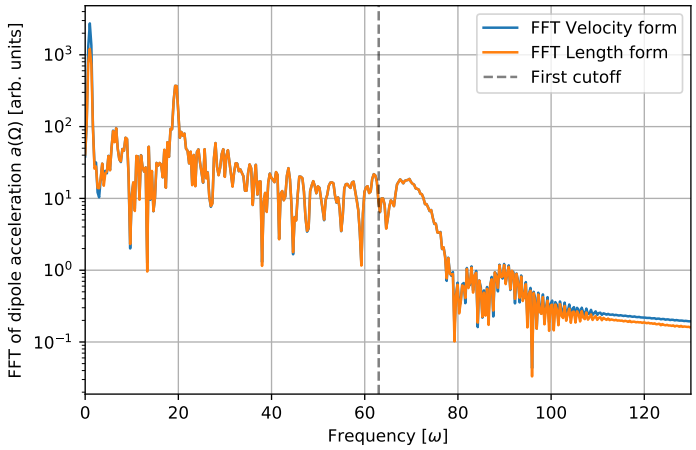



Figure 6.15: Comparison of the length and velocity forms of the dipole acceleration.

# CHAPTER 7

# Conclusion and Future Work

The main goal of this thesis was to implement the DVR basis set for the OATDCCD method. We successfully implemented the Q-space orbital equations for the OATDCCD method. The ground state of one-dimensional atoms represented with a DVR basis and the OATDCCD-dvr method was found using adiabatic switching, and proved to converge toward corresponding results from the highly accurate method of multiconfigurational time-dependent Hartree-Fock. This proved a validation of our implementation of both DVR method, the $Q$-space orbital equations and the adiabatic switching method. Further, we reproduced the qualitative time-dependent behaviour of one-dimensional systems under the influence of short-pulsed strong-field electromagnetic lasers. The computed spectrum corresponded to results expected from both more accurate methodsd such as MCTDHF, and gave the qualitative results expected from the theory of high harmonic generations. We provided the results in both the length form and the velocity form.

In addition to this, we solved the given subgoals described. The independent-particle time-propagation scheme for single-particle and Hartree-Fock solutions provided an important benchmarking tool. We used it both to validate the DVR implementation using the single-particle implementation of FDM, and to study the processes of imaginary time propagation and adiabatic switching in the case of HF.

## 7.1 Future work

A clear improvement of the implementation would be to rewrite the code in a low level language such as `C++`, which has far superior speed and better access to memory handling than `Python`. This would allow the code to utilize high performance computing facilities, allowing the study of much larger systems.

As seen from the time-use studies of the general CCD solver versus the restricted vesion, another significant speedup can be obtained by summing out the spin-degrees of freedom. The current use of the OATDCCD implementation is highly wasteful, as we study closed-shell systems with no spin-effects in the Hamiltonian. This is mainly because it provides a general treatment which is then easy to specialize later. However, implementing a restricted OATDCCD solver is an obivious next step for the `coupled-cluster` library.

A highly relevant avenue of research which would increase the cost, and as such require an implementation of effectivization is to implement the next step in the hierarchy of OATDCC theory by including triples. The resulting OATDCCDT method would provide an expensive but interesting point of study. This could also be done in a perturbative way, such as is commonly done in regular CC theory for with the CCSD(T)-method. However, it is unclear whether the perturbative triples provides the relevant contributions of the triples cluster in a strong-field regime, which is inherently non-perturbative. The perturbative triples would provide an interesting

Other points for future work include

- The theoretical study of the imaginary time OATDCC will determine its feasibility
- For studies of relativistic effects and magnetic fields, the inclusion of a B-field is highly interesting.
- The DVR basis set holds great interest for studies of infinite systems. This includes both the infinite electron gas, but even more important is the study of nuclear matter. In nuclear physics, operators of more bodies than two are relevant, which increases the importance of basis sets with symmetries in the matrix elements.

# Appendices

# APPENDIX A

# Mathematical derivations

## A.1 OATDCCD derivation

### A.1.1 P and Q-space equations

We will here derive the OATDCCD Q-space equations by performing the variations of the action with respect to the single-particle orbitals.

Solving eqs. (4.184) and (4.185), we have a way to construct the matrix elements of $\hat{\eta}$, we use the Q-space equations to find the time-evolution of the orbitals. Or rather, the P-space and Q-space equation together constitute the equations of motion of the orbital rotations. Defining first the mean-field potential as the one-particle operator with elements

$$\hat{W}_s^q = \int \mathrm{d}x_1 \tilde{\phi}_q(x_1)\hat{u}(x_1,x_2)\phi_s(x_1) \tag{A.1}$$

we write the ket-part of the Q-space equations as

$$i\hbar\rho_p^q\hat{Q}\partial_t\left|\phi_q\right\rangle = \rho_p^q\hat{Q}\hat{h}\left|\phi_q\right\rangle + \rho_{pq}^{rs}\hat{Q}\hat{W}_s^q\left|\phi_r\right\rangle. \tag{A.2}$$

The bra-part is given as

$$-i\hbar\rho_p^q\left(\partial_t\left\langle\tilde{\phi}_p\right|\right) = \rho_p^q\left\langle\tilde{\phi}_p\right|\hat{h}\hat{Q} + \rho_{pq}^{rs}\left\langle\tilde{\phi}_p\right|\hat{W}_s^q\hat{Q}. \tag{A.3}$$

We now wish to express these differential equations in terms of the coefficient matrices $C_{\alpha p}$ and $\tilde{C}_{p\alpha}$. We focus on the ket-equations (the bra-equations are completely analogous), and first insert for $Q$, giving

$$\begin{aligned} i\hbar\rho_p^q\left(1-\left|\phi_{p'}\right\rangle\left\langle\tilde{\phi}_{p'}\right|\right)\left|\dot{\phi}_q\right\rangle = {}& \rho_p^q\left(1-\left|\phi_{p'}\right\rangle\left\langle\tilde{\phi}_{p'}\right|\right)\hat{h}\left|\phi_q\right\rangle \\ & + \rho_{pq}^{rs}\left(1-\left|\phi_{p'}\right\rangle\left\langle\tilde{\phi}_{p'}\right|\right)\hat{W}_s^q\left|\phi_r\right\rangle \end{aligned}$$

which, with some rearrangement, becomes

$$\begin{aligned} i\hbar\rho_p^q\left|\dot{\phi}_q\right\rangle = {}& i\hbar\rho_p^q\eta_q^{p'}\left|\phi_{p'}\right\rangle + \rho_p^q\hat{h}\left|\phi_q\right\rangle - \rho_p^q h_q^{p'}\left|\phi_{p'}\right\rangle \\ & + \rho_{pr}^{qs}\hat{W}_s^r\left|\phi_q\right\rangle - \rho_{pr}^{qs}u_{sq}^{p'r}\left|\phi_{p'}\right\rangle \end{aligned}$$

Recalling that $\left|\phi_q\right\rangle = C_q^\beta\left|\chi_\beta\right\rangle$, we apply $\left\langle\tilde{\chi}^\alpha\right|$ from the left. This gives us the following matrix elements to consider,

$$\left\langle\tilde{\chi}^\alpha|\phi_p\right\rangle = C_p^\beta\left\langle\tilde{\chi}^\alpha|\chi_\beta\right\rangle = C_p^\beta\delta_\beta^\alpha = C_p^\alpha \tag{A.4}$$

$$\left\langle\tilde{\chi}^\alpha|\hat{h}|\phi_p\right\rangle = C_p^\beta h_\beta^\alpha \tag{A.5}$$

$$\left\langle\tilde{\chi}^\alpha|\hat{W}_s^r|\phi_p\right\rangle = C_p^\beta W_{\beta s}^{\alpha r} \tag{A.6}$$

where the mixed-representation mean-field operator matrix elements are explicitly

$$W_{\beta s}^{\alpha r} = \int\int \tilde{\chi}^{\alpha}(x_1)\tilde{\phi}^{r}(x_2)\hat{u}(x_1,x_2)\chi_{\beta}(x_1)\phi_s(x_2)\,\mathrm{d}x_1\mathrm{d}x_2. \tag{A.7}$$

Inserting gives

$$\begin{aligned} i\hbar\rho_p^q\dot{C}_q^{\alpha} = {} & i\hbar\rho_p^q\eta_q^{p'}C_{p'}^{\alpha} + \rho_p^q C_q^{\beta}h_{\beta}^{\alpha} - \rho_p^q h_q^{p'}C_{p'}^{\alpha} \\ & + \rho_{pr}^{qs}C_q^{\beta}W_{\beta s}^{\alpha r} - \rho_{pr}^{qs}u_{sq}^{p'r}C_{p'}^{\alpha} \end{aligned}$$

The final equation of motion for the ket-amplitudes is then found by inverting the one-body density matrix $\rho_p^q$. Contracting the free index $p$ with this and renaming some indices we get the final Q-space ket equation,

$$i\hbar\dot{C}_q^{\alpha} = i\hbar\eta_q^{p'}C_{p'}^{\alpha} + C_q^{\beta}h_{\beta}^{\alpha} - h_q^{p'}C_{p'}^{\alpha} + (\rho^{-1})_q^p\rho_{pr}^{q's}U_{q's}^{\alpha r}, \tag{A.8}$$

The corresponding bra-equation can be found similarly, and is

$$-i\hbar\dot{\tilde{C}}_{\alpha}^{p} = i\hbar\eta_{p'}^{p}\tilde{C}_{\alpha}^{p'} + \tilde{C}_{\beta}^{p}h_{\alpha}^{\beta} - h_{p'}^{p}\tilde{C}_{\alpha}^{p'} + (\rho^{-1})_q^p\rho_{p'r}^{qs}\tilde{U}_{\alpha s}^{p'r}. \tag{A.9}$$

Note that in these equations we have defined

$$U_{q's}^{\alpha r} = (C_{q'}^{\beta}W_{\beta s}^{\alpha r} - u_{sq'}^{p'r}C_{p'}^{\alpha}), \tag{A.10}$$

$$\tilde{U}_{\alpha s}^{p'r} = (\tilde{C}_{\beta}^{p'}W_{\alpha s}^{\beta r} - u_{q's}^{p'r}\tilde{C}_{\alpha}^{q'}). \tag{A.11}$$

These are the equations that we have implemented in our code, which we use to update the coefficients of the OA-basis.

### A.1.2 Density operators

We need the one- and two-body density operators for OATDDCCD, as given in [27]. The non-zero elements of the one-body density operator are

$$\rho_i^j = \delta_i^j - \frac{1}{2}\lambda_{ab}^{kj}\tau_{ki}^{ab}, \tag{A.12}$$

$$\rho_a^b = \delta_a^b - \frac{1}{2}\lambda_{ac}^{ij}\tau_{ij}^{bc}, \tag{A.13}$$

and the non-zero elements of the two-body density operator are

$$\rho_{ij}^{kl} = P(ij)\delta_i^k\delta_j^l - P(ij)P(kl)\frac{1}{2}\delta_i^k\lambda_{cd}^{lm}\tau_{jm}^{cd} + \frac{1}{2}\lambda_{cd}^{kl}\tau_{ij}^{cd}, \tag{A.14}$$

$$\rho_{ij}^{ab} = -P(ab)\frac{1}{2}\lambda_{cd}^{kl}\tau_{ij}^{ac}\tau_{kl}^{bd} + P(ij)\frac{1}{2}\lambda_{cd}^{kl}\tau_{ik}^{ac}\tau_{jl}^{bd} \tag{A.15}$$

$$+ P(ij)\frac{1}{2}\lambda_{cd}^{kl}\tau_{ij}^{ab}\tau_{kl}^{cd} + \frac{1}{4}\lambda_{cd}^{kl}\tau_{kl}^{ab}\tau_{ij}^{cd} + \tau_{ij}^{ab}, \tag{A.16}$$

$$\rho_{ia}^{jb} = -\rho_{ia}^{bj} = -\rho_{ai}^{jb} = \rho_{ai}^{bj} = \frac{1}{2}\delta_i^j\lambda_{ac}^{kl}\tau_{kl}^{bc} - \lambda_{ac}^{jk}\tau_{ij}^{bc}, \tag{A.17}$$

$$\rho_{ab}^{ij} = \lambda_{ab}^{ij}, \tag{A.18}$$

$$\rho_{ab}^{cd} = \frac{1}{2}\tau_{ij}^{cd}\lambda_{ab}^{ij}. \tag{A.19}$$

## A.2 Gauss Quadrature

To calculate the value of the weights $\omega_\alpha$ and the position of the integration points $x_\alpha$, we need some sort of connection between the orthogonal polynomials and the integral. To start with, we approximate the function $f(x)$ with a polynomial of degree $2N-1$,

$$\int_\Omega \mathrm{d}x\, \omega(x) f(x) \approx \int_\Omega \mathrm{d}x\, \omega(x) P_{2N-1}(x) \tag{A.20}$$

The specific degree of $P_{2N-1}(x)$ might appear to be arbitrary, but it is chosen for a specific reason. Using polynomial division, we can rewrite $P_{2N-1}(x)$ in terms of the known orthogonal polynomial $C_N(x)$ of degree $N$ and two unknown degree $N-1$ polynomials $P_{N-1}(x)$ and $Q_{N-1}(x)$,

$$P_{2N-1}(x) = C_N(x) P_{N-1}(x) + Q_{N-1}(x). \tag{A.21}$$

The first term here will cancel under the integral in eq. (A.20). To see this, consider that the set of polynomials $\{C_n\}_{n=0}^{N-1}$ forms a basis for $\mathbb{P}^{N-1}$, the space of polynomials of degree $N-1$. We can expand $P_{N-1}(x)$ in this basis,

$$P_{N-1}(x) = \sum_{n=0}^{N-1} p_n C_n(x). \tag{A.22}$$

However, from eq. (3.28) we know that $C_N(x)$ is orthogonal with all other $C_n(x)$ for $n \neq N$. As such, for the integration over the specific domain and corresponding weight function, it has to be orthogonal to all polynomials of degree lower than $N$, including $P_{N-1}(x)$. Using this and eq. (A.21) on the right hand side of eq. (A.20) we find the important relation

$$\int_\Omega \mathrm{d}x\, \omega(x) P_{2N-1}(x) = \int_\Omega \mathrm{d}x\, \omega(x) \left(C_N P_{N-1} + Q_{N-1}\right) = \int_\Omega \mathrm{d}x\, \omega(x) Q_{N-1}(x), \tag{A.23}$$

where we dropped the arguments in the middle for brevity. Additionally, let $x_\alpha$ be the roots of $C_N(x)$, such that $C_N(x_\alpha) = 0$. We find another relationship between $P_{2N-1}(x)$ and $Q_{N-1}(x)$ by evaluating,

$$P_{2N-1}(x_\alpha) = C_N(x_\alpha) P_{N-1}(x_\alpha) + Q_{N-1}(x_\alpha) = Q_{N-1}(x_\alpha) \tag{A.24}$$

Continuing, we now wish to evaluate the integral $\int_\Omega \mathrm{d}x\, \omega(x) Q_{N-1}$ in order to find the wanted approximation of the original integral. We first expand $Q_{N-1}(x)$ in terms of the $C_n(x)$ up to order $N-1$,

$$Q_{N-1}(x) = \sum_{n=0}^{N-1} \alpha_n C_n(x) \tag{A.25}$$

Inserting into the approximate integral gives

$$\begin{aligned}
\int_\Omega \mathrm{d}x\, \omega(x) Q_{N-1}(x) &= \sum_{n=0}^{N-1} \alpha_n \int_\Omega \mathrm{d}x\, \omega(x) C_n(x) \\
&= \sum_{n=0}^{N-1} \frac{\alpha_n}{k_0} \int_\Omega \mathrm{d}x\, \omega(x) C_n(x) C_0(x) \\
&= \frac{\alpha_0}{k_0}
\end{aligned}$$

where we multiplied and divided with the constant value of $C_0(x) = k_0$.

To find the value of $\alpha_0$, we need to find the coefficients of the expansion eq. (A.25). We know the values of $Q_{N-1}(x_\alpha)$ to be given by the approximation $Q_{N-1}(x_\alpha) = P_{2N-1}(x_\alpha) \approx f(x_\alpha)$ for $\alpha = 0, \ldots, N-1$. Evaluating eq. (A.25) at the points $x_\alpha$, it turns into a matrix equation,

$$f_\alpha = \sum_{n=0}^{N-1} \alpha_n C_{n\alpha}, \tag{A.26}$$

where $f_\alpha \equiv f(x_\alpha)$ and $C_{n\alpha} \equiv C_n(x_\alpha)$. Solving for $\alpha_n$ gives

$$\alpha_n = \sum_{\alpha=0}^{N-1} (C^{-1})_{n\alpha} f_\alpha, \tag{A.27}$$

which is an valid as long as $C_{n\alpha}$ has an inverse. In the case of Gauss quadrature, the inverse of $C_{n\alpha}$ is guaranteed by the fact that the orthogonal polynomials are linearly independent. Setting $n = 0$ and putting it all together, we can recognize the following quadrature rule

$$\int_\Omega \mathrm{d}x\, \omega(x) f(x) \approx \sum_{\alpha=0}^{N-1} \frac{(C^{-1})_{0\alpha}}{k_0} f(x_\alpha) \tag{A.28}$$

where the weights are given by $\omega_\alpha = (C^{-1})_{0\alpha}/k_0$ and the grid points by the roots of $C_N(x)$.

## A.3 Sinc-DVR

### A.3.1 Kinetic energy matrix elements

A fourier basis taken to the infinite grid gives us a sinc-dvr basis. We will show this a bit later, however we will use this fact to find analytical values for the kinetic energy matrix elements of the Sinc-DVR basis.

The fourier basis wave functions on a grid $x_i = a + i(b-a)/N$ $(i = 1, \ldots, N-1)$ are given by

$$\phi_n(x) = \left(\frac{2}{b-a}\right)^{1/2} \sin\left(\frac{n\pi(x-a)}{b-a}\right), \quad n = 1, \ldots, N-1, \tag{A.29}$$

where the endpoints are 0 for an infinite well potential. The grid point representation of the kinetic energy is then given by

$$\begin{aligned} T_{ii'} &= -\frac{1}{2}\Delta x \sum_{n=1}^{N-1} \phi_n(x_i) \left.\frac{\mathrm{d}^2\phi_n(x)}{\mathrm{d}x^2}\right|_{x=x_{i'}} \\ &= \frac{1}{2}\left(\frac{\pi}{b-a}\right)^2 \frac{2}{N} \sum_{n=1}^{N-1} n^2 \sin\left(\frac{n\pi i}{N}\right) \sin\left(\frac{n\pi i'}{N}\right) \end{aligned} \tag{A.30}$$

Focusing on the sum, we rewrite

$$\sum_{n=1}^{N-1} n^2 \sin\left(\frac{n\pi i}{N}\right) \sin\left(\frac{n\pi i'}{N}\right) = \frac{1}{2}\sum_{n=1}^{N-1} n^2 \left[\cos\left(\frac{n\pi(i-i')}{N}\right) - \cos\left(\frac{n\pi(i+i')}{N}\right)\right]$$
$$\equiv \frac{1}{2}\sum_{n=1}^{N-1} n^2 \left[\cos(nB) - \cos(nA)\right]$$

with $A \equiv \pi(i+i')/N$ and $B \equiv \pi(i-i')/N$. These two terms require the exact same treatment, so we focus on the first. Using a couple of tricks referenced in [10], we first rewrite

$$\frac{1}{2}\sum_{n=1}^{N-1} n^2 \left[\cos(nA)\right] = \frac{\partial}{\partial A}\frac{1}{2}\sum_{n=1}^{N-1} n \left[\sin(nA)\right] = -\frac{\partial^2}{\partial A^2}\frac{1}{2}\sum_{n=1}^{N-1} \left[\cos(nA)\right] \quad \text{(A.31)}$$

This sum can be rewritten as a geometric series as

$$\sum_{n=1}^{N-1} \cos(nA) = \mathrm{Re}\left\{\sum_{n=1}^{N-1} e^{inA}\right\} = \mathrm{Re}\left\{\frac{e^{iA} - e^{iAN}}{1 - e^{iA}}\right\}$$

where we rewrote the sum as a geometric series. We now pull out a factor of $e^{iA/2}$ from the denominator, find an explicitly real numerator and then use $\mathrm{Re}\{ie^{ix}\} = \mathrm{Re}\{i\cos x - \sin x\} = -\sin x$,

$$\mathrm{Re}\left\{\frac{e^{iA} - e^{iAN}}{1 - e^{iA}}\right\} = \mathrm{Re}\left\{\frac{e^{iA/2} - e^{iA(N-1/2)}}{e^{-iA/2} - e^{iA/2}}\right\} = \frac{\mathrm{Re}\{ie^{iA/2} - ie^{iA(N-1/2)}\}}{2\sin(A/2)}$$
$$= \frac{1}{2}\frac{\sin(A(N-1/2))}{\sin(A/2)} - \frac{1}{2}$$

Some labourious derivation is ahead when we consider again the original expression gained from eq. (A.31),

$$-\frac{\partial^2}{\partial A^2}\frac{1}{2}\mathrm{Re}\left\{\frac{e^{iA} - e^{iAN}}{1 - e^{iA}}\right\} = -\frac{1}{4}\frac{\partial^2}{\partial A^2}\left[\frac{\sin(A(N-1/2))}{\sin(A/2)} - 1\right]$$
$$= -\frac{1}{4}\frac{\partial^2}{\partial A^2}\left[\sin AN \cot A/2 - \cos AN\right]$$
$$= -\frac{1}{4}\frac{\partial}{\partial A}\left[N\cos AN \cot A/2 - \frac{1}{2}\sin AN \cot' A/2 + N\sin AN\right]$$
$$= -\frac{1}{4}\Big[-N^2 \sin AN \cot A/2 - N\cos AN \cot' A/2$$
$$+ \frac{1}{4}\sin AN \cot'' A/2 + N^2 \cos AN\Big]$$

Considering that $\sin(AN) = \sin(\pi(i-i')) = 0$ and similarly $\cos(AN) = (-1)^{i-i'}$, and that the derivative of $\cot A/2$ is $\cot' A/2 = -1/\sin^2(A/2) = -1/\sin^2(\pi(i-i')/2N)$, we get

$$-\frac{\partial^2}{\partial A^2}\frac{1}{2}\mathrm{Re}\left\{\frac{e^{iA} - e^{iAN}}{1 - e^{iA}}\right\} = N\frac{(-1)^{i-i'}}{4}\left[\frac{1}{\sin^2(\pi(i-i')/2N)} - N\right]$$

A calculation for $B$ is completely analogous, but with $i + i'$ instead of $i - i'$. We even have $(-1)^{i-i'} = (-1)^{i-i'+2i'} = (-1)^{i+i'}$. Inserting into eq. (A.30), the opposite sign makes the $N$-term disappear, and we get

$$T_{ii'} = \frac{1}{2}\left(\frac{\pi}{b-a}\right)^2 \frac{(-1)^{i-i'}}{2N}\left[\frac{1}{\sin^2(\pi(i-i')/2N)} - \frac{1}{\sin^2(\pi(i+i')/2N)}\right], \quad i \neq i' \tag{A.32}$$

Also, for $i = i'$ the calculations are very similar, although the initial sum now reads

$$\sum_{n=1}^{N-1} \frac{n^2}{2}(\cos(nA) - 1) \tag{A.33}$$

with $A = 2n\pi i/N$. We then get

$$T_{ii} = \frac{1}{2}\left(\frac{\pi}{b-a}\right)^2 \left[\frac{1}{\sin^2 \pi i/N} - \frac{N^3}{3} + \frac{N^2}{2} - \frac{N}{6}\right] \tag{A.34}$$

#### Infinite size grid limit

The infinite grid limit is found by simultaneously letting $a \to \infty$, $b \to \infty$, and $N \to \infty$ with $\Delta x = (b-a)/N$ kept fixed. Consider that with increasing $N$ we get

$$\frac{1}{\sin^2(\pi(i-i')/2N)} \approx \left(\frac{2N}{\pi(i-i')}\right)^2 \tag{A.35}$$

which gives us the infinite size limit as

$$T_{ij} = \frac{(-1)^{i-j}}{2(\Delta x)^2}\begin{cases} \frac{\pi^2}{3}, & i = j \\ \frac{2}{(i-j)^2}, & i \neq j \end{cases} \tag{A.36}$$

### A.3.2 Connection to Sinc-DVR

We now take a look at the wavefunction in the infinite grid limit. The wavefunction for the infinite sized grid is

$$\langle x|x_i\rangle = \frac{2}{n-a}\sum_{n=1}^{N-1} \sin\left(\frac{n\pi(x-a)}{b-a}\right)\sin\left(\frac{n\pi(x_i-a)}{b-a}\right) \tag{A.37}$$

Using the same methods as in the previous section, the analytical expression for the wave function is

$$\langle x|x_i\rangle = \frac{\sin(\pi(x-x_i)/\Delta x)}{\pi(x-x_i)} = s_i(x), \tag{A.38}$$

which we recognize as the sinc-dvr function associated with grid point i.

### A.3.3 Size extensivity of Coupled Cluster

In literature, the usage of the terms of size extensivity and size consistency is *extensively non-consistent*!

Assume that we have a system composite of two non-interacting systems. Consequently it can described by a hamiltonian $H^{(AB)}$ which is a direct sum of hamiltonians of the subsystems $H^{(A)}$ and $H^{(B)}$

$$H^{(AB)} = H^{(A)} + H^{(B)} \tag{A.39}$$

and solutions of the composite system are direct products of solutions of the subsystems

$$|\Psi^{(AB)}\rangle = |\Psi^{(A)}\rangle\, |\Psi^{(B)}\rangle \tag{A.40}$$

The cluster operator is a connected operator, which means that for two disconnected systems all parts will contain diagrams from either one or the other, but never both simultaneously. It can subsequently also be separated into cluster operators of the two subsystems,

$$T^{(AB)} = T^{(A)} + T^{(B)} \tag{A.41}$$

Cluster operators of different systems commute $\left[T^{(A)}, T^{(B)}\right] = 0$, which gives us

$$e^{T(A)+T(B)} = e^{T(A)} e^{T(B)}. \tag{A.42}$$

The wavefunction then becomes

$$|\Psi^{(AB)}\rangle = e^{(AB)}\, |\Phi_0^{(AB)}\rangle = e^{(A)} e^{(B)}\, |\Phi_0^{(A)}\rangle\, |\Phi_0^{(B)}\rangle = |\Psi^{(A)}\rangle \left|\Psi^{(B)}\right\rangle. \tag{A.43}$$

Comparing this to eq. (A.40) we see that the CC wavefunction has the proper behaviour and is size extensive, a result which holds for any truncation level of the cluster operator.

# Bibliography


[1] 'An updated set of basic linear algebra subprograms (BLAS)'. In: *ACM Transactions on Mathematical Software* vol. 28, no. 2 (1st June 2002), pp. 135–151.

[2] Anderson, E. et al. *LAPACK Users' Guide*. Software, Environments and Tools. Society for Industrial and Applied Mathematics, 1st Jan. 1999. 424 pp.

[3] Anisimovas, E. and Matulis, A. 'Energy spectra of few-electron quantum dots'. In: *Journal of Physics: Condensed Matter* vol. 10, no. 3 (26th Jan. 1998), pp. 601–615.

[4] Arponen, J. 'Variational principles and linked-cluster exp S expansions for static and dynamic many-body problems'. In: *Annals of Physics* vol. 151, no. 2 (1st Dec. 1983), pp. 311–382.

[5] Born, M. and Fock, V. 'Beweis des Adiabatensatzes'. In: *Zeitschrift für Physik* vol. 51, no. 3 (1st Mar. 1928), pp. 165–180.

[6] Brown, P. N., Byrne, G. D. and Hindmarsh, A. C. 'VODE: a variable-coefficient ODE solver'. In: *SIAM Journal on Scientific and Statistical Computing* vol. 10, no. 5 (1st Sept. 1989), pp. 1038–1051.

[7] Čižek, J. and Paldus, J. 'Correlation problems in atomic and molecular systems III. Rederivation of the coupled-pair many-electron theory using the traditional quantum chemical methods'. In: *International Journal of Quantum Chemistry* vol. 5, no. 4 (1971). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.560050402, pp. 359–379.

[8] Čížek, J. 'On the Correlation Problem in Atomic and Molecular Systems. Calculation of Wavefunction Components in Ursell-Type Expansion Using Quantum-Field Theoretical Methods'. In: *The Journal of Chemical Physics* vol. 45, no. 11 (1st Dec. 1966). Publisher: American Institute of Physics, pp. 4256–4266.

[9] Čížek, J. 'On the Use of the Cluster Expansion and the Technique of Diagrams in Calculations of Correlation Effects in Atoms and Molecules'. In: *Advances in Chemical Physics*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470143599.ch2. John Wiley & Sons, Ltd, 1969, pp. 35–89.

[10] Colbert, D. T. and Miller, W. H. 'A novel discrete variable representation for quantum mechanical reactive scattering via the $S$ -matrix Kohn method'. In: *The Journal of Chemical Physics* vol. 96, no. 3 (Feb. 1992), pp. 1982–1991.

[11] Crawford, T. D. and Schaefer, H. F. 'An Introduction to Coupled Cluster Theory for Computational Chemists'. In: *Reviews in Computational Chemistry*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470125915.ch2. John Wiley & Sons, Ltd, 2007, pp. 33–136.

[12] Dobson, J. F. 'Harmonic-Potential Theorem: Implications for Approximate Many-Body Theories'. In: *Physical Review Letters* vol. 73, no. 16 (17th Oct. 1994). Publisher: American Physical Society, pp. 2244–2247.

[13] Dunning, T. H. 'Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen'. In: *The Journal of Chemical Physics* vol. 90, no. 2 (15th Jan. 1989). Publisher: American Institute of Physics, pp. 1007–1023.

[14] Fink, M. 'A new method for evaluating the density matrix and its application to the ground state form factors of 4He and 16O'. In: *Nuclear Physics A* vol. 221, no. 1 (11th Mar. 1974). Publisher: North-Holland, pp. 163–172.

[15] Golub, G. H. and Welsch, J. H. 'Calculation of Gauss quadrature rules'. In: *Mathematics of Computation* vol. 23, no. 106 (1969), pp. 221–230.

[16] Gordon, A., Kärtner, F. X., Rohringer, N. and Santra, R. 'Role of Many-Electron Dynamics in High Harmonic Generation'. In: *Physical Review Letters* vol. 96, no. 22 (6th June 2006), p. 223902.

[17] Groenenboom, G. C. and Colbert, D. T. 'Combining the discrete variable representation with the S-matrix Kohn method for quantum reactive scattering'. In: *The Journal of Chemical Physics* vol. 99, no. 12 (15th Dec. 1993), pp. 9681–9696.

[18] Hairer, E., Lubich, C. and Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. 2nd ed. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag, 2006.

[19] Hairer, E., Nørsett, S. P. and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Google-Books-ID: F93u7VcSRyYC. Springer Science & Business Media, 16th Apr. 2008. 540 pp.

[20] Helgaker, T. and Jørgensen, P. 'Analytical Calculation of Geometrical Derivatives in Molecular Electronic Structure Theory'. In: *Advances in Quantum Chemistry* vol. 19 (1st Jan. 1988). Publisher: Academic Press, pp. 183–245.

[21] Helgaker, T., Jorgensen, P. and Olsen, J. *Molecular Electronic-Structure Theory*. Google-Books-ID: lNVLBAAAQBAJ. John Wiley & Sons, 11th Aug. 2014. 1125 pp.

[22] Hermanns, S., Balzer, K. and Bonitz, M. 'The non-equilibrium Green function approach to inhomogeneous quantum many-body systems using the generalized Kadanoff–Baym ansatz'. In: *Physica Scripta* vol. T151 (Nov. 2012). Publisher: IOP Publishing, p. 014036.

[23] Hochstuhl, D., Bauch, S. and Bonitz, M. 'Multiconfigurational time-dependent Hartree-Fock calculations for photoionization of one-dimensional Helium'. In: *Journal of Physics: Conference Series* vol. 220 (1st Apr. 2010), p. 012019.

[24] Jones, J. R. et al. 'An efficient basis set representation for calculating electrons in molecules'. In: *Molecular Physics* vol. 114, no. 13 (2nd July 2016), pp. 2014–2028. arXiv: 1507.03324.

[25] Jordan, G. and Scrinzi, A. 'Core-polarization effects in molecular high harmonic generation'. In: *New Journal of Physics* vol. 10, no. 2 (Feb. 2008). Publisher: IOP Publishing, p. 025035.

[26] Kristiansen, H. E. 'Time Evolution of Quantum Mechanical Many-Body Systems'. In: (2017). Accepted: 2018-02-22T22:28:14Z.

[27] Kvaal, S. 'Ab initio quantum dynamics using coupled-cluster'. In: *The Journal of Chemical Physics* vol. 136, no. 19 (18th May 2012), p. 194109.

[28] Kvaal, S. 'Variational formulations of the coupled-cluster method in quantum chemistry'. In: *Molecular Physics* vol. 111, no. 9 (1st July 2013). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00268976.2013.812254, pp. 1100–1108.

[29] Lewenstein, M., Balcou, P., Ivanov, M. Y., L'Huillier, A. and Corkum, P. B. 'Theory of high-harmonic generation by low-frequency laser fields'. In: *Physical Review A* vol. 49, no. 3 (1st Mar. 1994). Publisher: American Physical Society, pp. 2117–2132.

[30] Light, J. C. and Carrington, T. 'Discrete-Variable Representations and their Utilization'. In: *Advances in Chemical Physics*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470141731.ch4. John Wiley & Sons, Ltd, 2007, pp. 263–310.

[31] Löwdin, P.-O. 'Quantum Theory of Many-Particle Systems. I. Physical Interpretations by Means of Density Matrices, Natural Spin-Orbitals, and Convergence Problems in the Method of Configurational Interaction'. In: *Physical Review* vol. 97, no. 6 (15th Mar. 1955). Publisher: American Physical Society, pp. 1474–1489.

[32] Magnus, W. 'On the exponential solution of differential equations for a linear operator'. In: *Communications on Pure and Applied Mathematics* vol. 7, no. 4 (1954). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160070404, pp. 649–673.

[33] McPherson, A., Gibson, G., Jara, H., Johann, U., Luk, T. S., McIntyre, I. A., Boyer, K. and Rhodes, C. K. 'Studies of multiphoton production of vacuum-ultraviolet radiation in the rare gases'. In: *JOSA B* vol. 4, no. 4 (1st Apr. 1987). Publisher: Optical Society of America, pp. 595–601.

[34] Miyagi, H. and Madsen, L. B. 'Time-dependent restricted-active-space self-consistent-field singles method for many-electron dynamics'. In: *The Journal of Chemical Physics* vol. 140, no. 16 (25th Apr. 2014). Publisher: American Institute of Physics, p. 164309.

[35] Miyagi, H. and Madsen, L. B. 'Time-dependent restricted-active-space self-consistent-field theory for laser-driven many-electron dynamics'. In: *Physical Review A* vol. 87, no. 6 (21st June 2013). Publisher: American Physical Society, p. 062511.

[36] Al-Mohy, A. H. and Higham, N. J. 'A New Scaling and Squaring Algorithm for the Matrix Exponential'. In: *SIAM Journal on Matrix Analysis and Applications* vol. 31, no. 3 (1st Aug. 2009), pp. 970–989.

[37] Pedersen, T. B., Fernández, B. and Koch, H. 'Gauge invariant coupled cluster response theory using optimized nonorthogonal orbitals'. In: *The Journal of Chemical Physics* vol. 114, no. 16 (22nd Apr. 2001), pp. 6983–6993.

[38] Pedersen, T. B. and Kvaal, S. 'Symplectic integration and physical interpretation of time-dependent coupled-cluster theory'. In: *The Journal of Chemical Physics* vol. 150, no. 14 (14th Apr. 2019), p. 144106.

[39] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Google-Books-ID: 1aAOdzK3FegC. Cambridge University Press, 6th Sept. 2007. 1195 pp.

[40] Sato, T., Pathak, H., Orimo, Y. and Ishikawa, K. L. 'Communication: Time-dependent optimized coupled-cluster method for multielectron dynamics'. In: *The Journal of Chemical Physics* vol. 148, no. 5 (7th Feb. 2018), p. 051101.

[41] Schøyen, Ø. S. 'Real-time quantum many-body dynamics'. In: (2019). Accepted: 2020-02-07T23:45:50Z.

[42] Schrödinger, E. 'An Undulatory Theory of the Mechanics of Atoms and Molecules'. In: *Physical Review* vol. 28, no. 6 (1st Dec. 1926). Publisher: American Physical Society, pp. 1049–1070.

[43] Shannon, C. 'Communication in the Presence of Noise'. In: *Proceedings of the IRE* vol. 37, no. 1 (Jan. 1949). Conference Name: Proceedings of the IRE, pp. 10–21.

[44] Shavitt, I. and Bartlett, R. J. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. Cambridge: Cambridge University Press, 2009.

[45] Sukiasyan, S., McDonald, C., Destefani, C., Ivanov, M. Y. and Brabec, T. 'Multielectron Correlation in High-Harmonic Generation: A 2D Model Analysis'. In: *Physical Review Letters* vol. 102, no. 22 (5th June 2009), p. 223002.

[46] Surján, P. R. 'The Brillouin Theorem'. In: *Second Quantized Approach to Quantum Chemistry: An Elementary Introduction*. Ed. by Surján, P. R. Berlin, Heidelberg: Springer, 1989, pp. 87–92.

[47] Szabo, A. and Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Revised ed. edition. Mineola, N.Y: Dover Publications, 2nd July 1996. 480 pp.

[48] *The Nobel Prize in Physics 2018*. The Nobel Prize in Physics 2018. 31st Aug. 2020. URL: https://www.nobelprize.org/prizes/physics/2018/summary/ (visited on 01/09/2020).

[49] Thomas, L. H. 'Elliptic problems in linear difference equations over a network'. In: *Watson Sci. Comput. Lab. Rept., Columbia University, New York* vol. 1 (1949).

[50] Ullrich, C. *Time-Dependent Density-Functional Theory: Concepts and Applications*. Google-Books-ID: hCNNsC4sEtkC. OUP Oxford, 2012. 541 pp.

[51] Winther-Larsen, S. G. 'AQUADUCT Ab Initio Quantum Dynamics Using Coupled Cluster in Time'. In: (2019). Accepted: 2020-01-13T23:45:40Z.