University of Oslo

Master Thesis

Coupled cluster studies of infinite systems

A thesis submitted in fulfilment of the requirements for the degree of Master

in the

Computational Physics Group
Department or Physics

August 2015

“What I cannot create, I do not understand.”

Richard Feynman

UNIVERSITY OF OSLO

Abstract

Department or Physics

Master

Coupled cluster studies of infinite systems

by Audun Skau Hansen

We investigate how the coupled cluster method at the level of doubles and triples amplitudes contributes to the ground state energy of the homogeneous electron gas. We present and derive the formalism and equations needed, and describe in detail how two independent and conceptually differing computational schemes may be implemented efficiently for the system uner study. We finally perform numerous calculations for the infinite electron gas, investigate how the gradual inclusion of more diagrams leading up to the full coupled cluster doubles triples (CCDT) method affects the energy, and we estimate the energy in the thermodynamical limit by extrapolating results from large scale computations. In order to check all equations, we have also developed a software which produces all equations needed at a given level of truncation of coupled cluster theory. This allows for efficient benchmarking of equations as well as codes for implementing various conributions to the theory.

Acknowledgements

The content of this thesis relies heavily on the work of others. While the many physicist who paved way for this work is far too many to each be mentioned by name, I would like to extend my appreciation of them all. Without their valuable contributions, we may still have been standing paralyzed when faced with the discrete nature of the universe. After years as a student of physics, I'm more impressed and perplexed than ever by the fact that someone could come up with the concepts and formalism of quantum mechanics.

Of the more direct contributions to this thesis, I wish to acknowledge the many made by my advisor, prof. Morten Hjorth-Jensen. His valuable and deep insight into many-body theory as well as the broader context of physics has been extremely helpful. His liberal approach to guidance has allowed me to also explore some of my own ideas and paths - even some detours - which I suspect may have resulted in some more confidence from my part when dealing with such matters as science. The time spent under his supervision at Michigan State University was very rewarding and productive.

Another direct contributor to this thesis is Gustav Baardsen, ph.d. In large parts, this thesis is founded on his doctoral thesis *Coupled-cluster theory for infinite matter* [1]. He has also been very helpful in supplying me with his code, as well as very extensive datasets for comparison with my own.

It has also been of great help to be part of the Computational Physics group at the University of Oslo. Everything from minor discussions on which libraries to utilize or how to install drivers for the printer, to more central issues as how to compile Armadillo on the Abel cluster (Thank you Jørgen Høgberget and Svenn-Arne Dragly) have been extremely valuable in my work. I am proud to be a member of this highly competent group.

Finally I wish to thank my family: my dear Kine for her care, understanding and dedication, as well as our two sons Erik and Alfred for allowing me to sleep after late nights working on this project.

As mentioned, some of the calculations in this thesis was performed on the Abel Cluster, owned by the University of Oslo and the Norwegian metacenter for High Performance Computing (NOTUR), and operated by the Department for Research Computing at USIT, the University of Oslo IT-department. [2] I am grateful for being trusted with access to these overwhelming resources.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis we will use the coupled cluster method, also called coupled cluster *theory*, to investigate the homogeneous electron gas. More specifically, we will investigate how the inclusion of triple excitations in the coupled cluster equations will affect calculations of the ground state energy. Numerous (see for example Refs. [1, 4, 6]) calculations of the system in question have been made using the coupled cluster doubles (CCD) method, doubles with perturbative triples (CCD(T)), and various other many-body methods, but to the author's best knowledge this thesis presents for the first time results containing triple excitations for the electron gas.

We shall systematically investigate how each diagram present in the equation contributes to the ground state energy for smaller basis sets, incrementally leading up the the full inclusion of triple excitations (CCDT), and we will estimate the energy in the thermodynamical limit using a CCDT-1 [7] approach and compare these results with existing results from full configuration interaction quantum monte carlo (FCIQMC) present in the literature. Furthermore, we have also derived a software which allows for the automatical derivation of various coupled cluster truncations, with the derivation of the corresponding analytical expressions, their pertinent diagrams and code.

In order to convey these results, this thesis is structured as follows: In the second chapter of this thesis 2, we review the context from which the many-body problem arises, and we motivate the need for the formalism developed in Chapters 3 and 4. In Chapter 5, we give a broad overview of the methodology involved, and we give some special attention to the methods that closely relate to the coupled cluster method, such as Hartree-Fock theory, configuration interaction theory and many-body perturbation theory.

In Chapter 6, we derive the coupled cluster method, and we introduce both diagrammatic techniques and computational implementations of these that we use to derive the actual equations for various truncations of the coupled cluster method. Furthermore, we discuss also our software for deriving automatically various coupled cluster truncations.

The homogeneous electron gas (HEG) is reviewed in Chapter 7, where we derive and present the expressions needed to evaluate the coupled cluster equations, such as the the plane wave basis, Fock matrix elements [8], the two-body interaction [7] and the general structure of the model.

In Chapter 8 we present implementational details, and we describe two conceptually different schemes for solving the coupled cluster equations for our system. We also discuss topics such as performance and memory usage.

In the final Chapter 9, we validate the implementations by comparing with results published in other studies and by comparing results from the two independent solvers. We then perform a series of calculations for smaller basis sets, ending up with the full inclusion of triple amplitudes. Estimates in the thermodynamical limit is performed by extrapolation from a data set obtained by running our CCDT-1 code on the Abel cluster (see Ref. [2]).

We discuss these findings in light of calculations performed by others, and finally, in the last chapter, we present our concluding remarks, recommendations and perspectives for future work.

Chapter 2

Many Body Quantum Theory

A brief review The aim of this chapter is to introduce some of the fundamental concepts relevant to this thesis, with an emphasis on many-body theory and the formalism of second quantization.

2.1 Many-body Theory

Many-body theory is the framework which to date best describes and predicts phenomena relating to interacting quantum systems. The main body of the theory was developed by physicists such as Fermi, Pauli and Dirac in the late 1920s. As the Davisson-Germer electron diffraction experiment confirmed the particle wave duality of matter in 1927 [9] and the discovery of half integer spin was made by Goudsmit and Uhlenbeck in 1925 [9], these results found their rationale in the theoretical work of Pauli, Fermi and Dirac [9]. Dirac introduced the *Second Quantization Formalism* in 1927 [7].

Important contributions have been made continuously over the years. Feynman introduced the diagrammatic formalism in 1949 [7], prior to the advent of many-body perturbation theory, introduced by Brueckner and Levinson in 1955 [7]. The subsequent decades have seen an explosion in the development of various so-called first principle methods[1], including several Monte Carlo methods, Green's function methods, full configuration interaction theory and many other many-body approaches. In this thesis, we will in particular pay attention to coupled cluster theory. This method was originally introduced in order to solve the nuclear many-body problem by Coester and Kümmel [10]. Coupled cluster theory, with its various approximations, has during the last five

[1] With first principle or *ab initio* methods, we mean methods that allow, with a given Hamiltonian, for either exact solutions of the many-body problem or approximative solutions which can be improved upon systematically

decades provided highly accurate predictions for a wide range of interacting quantum systems and has become one of the standard many-body methods in quantum chemistry and nuclear physics, providing precise benchmarks for systems up to hundreds of interacting electrons or nucleons.

Coupled cluster theory offers a range of methods for approximating energies and properties of systems. When choosing which method to utilize, one has to consider the trade off between performance and accuracy. While some approximations may in principle provide us with very precise results, they may have computational requirements beyond what is currently achievable. With these considerations, the so-called CCSD(T) (Coupled Cluster Singles and Doubles with Perturbative Triples) is considered to be the "gold standard" of coupled cluster theory, as it is both efficient and highly accurate.[2]

In the following sections we will briefly review some of the essential elements of quantum many-body theory and introduce notations used in the rest of this thesis. Since a great number of in-depth and excellent modern textbooks are written on these topics, the aim here is merely to introduce concepts and theory that will be utilized in later parts of this thesis. For an extensive introduction to the basic elements of many-body theory, the reader is referred to some of the many books on the subject. [7, 8, 11–13].

2.2 The Postulates of Quantum Mechanics

The mathematical framework of quantum mechanics is rooted in several fundamental postulates. We will here briefly state these.

(1) The Wave Function The state of a quantum mechanical system is fully specified in time and space by a wave function $|\Psi(x,t)\rangle$. Born's Statistical Interpretation [14] suggests that the probability of finding the system in the volume element dx at time t is defined by $\Psi(x,t)^*\Psi(x,t)dx$. Another important property is that the wave function should be normalized to one in the full occupied space Ω [14], that is

$$\int_{\Omega} \Psi(\mathbf{x},t)^*\Psi(\mathbf{x},t)d\Omega = 1.$$

Here the variable $\mathbf{x}$ can represent one or more-dimensional systems and include for example spin degrees of freedom.

[2] The author was not able to pinpoint exactly where this concept of "gold-standard" originated, but a search on Google Scholar with the keywords "coupled cluster gold standard" clearly shows that this terminology is commonly used within the quantum chemistry community.

(2) Observables For any measurable quantity, such as energy, momentum, or spin, there exists a corresponding linear, Hermitean operator. Such operators are commonly denoted with a hat; $\hat{A}$

(3) Measurement A measurement of any observables linked with the operator $\hat{A}$ acting on a given system, will result in a value a, corresponding to the eigenvalues of the equation

$$\hat{A}|\Psi\rangle = a|\Psi\rangle.$$

(4) Average measurement For a system in the state $|\Psi\rangle$, we define the average measurement of $\hat{A}$ by

$$\int_{\Omega} \Psi(\mathbf{x},t)^* \hat{A}\Psi(\mathbf{x},t) d\Omega \equiv \langle\Psi|\hat{A}|\Psi\rangle = \langle A\rangle.$$

Here we have assumed that the wave function is normalized. The average measurement is *not the most likely result*, merely the average of a multitude of measurements on identical systems.

(5) Time evolution The system will evolve in time in accordance with the time independent Schrödinger equation

$$\hat{H}|\Psi(x,t)\rangle = i\hbar\frac{\partial}{\partial t}|\Psi(x,t)\rangle. \tag{2.1}$$

While the more general requirement of a wave function is that it fulfills the time dependent Schrödinger equation, we may also seek stationary solutions to the *time independent* Schrödinger equation

$$\hat{H}|\Psi(x)\rangle = \epsilon|\Psi(x)\rangle, \tag{2.2}$$

where Ψ no longer has any dependence on t, and ϵ is considered the eigevalues of $\hat{H}$. Such a state may describe for example the ground state, in which case ϵ represents the ground state energy.

(6) The Pauli Exclusion Principle For systems composed of half-integer particles (fermions), the total wave function has to be antisymmetric. As a consequence of the this principle, when using a single-particle basis to build a many-body state, no two indistinguishable *fermions* can occupy the same quantum state.

While not all textbooks list the Pauli principle as a separate postulate, many experiments have been conducted in order to test the validity of the postulate, with at present no deviations, within experimental uncertainties, from the postulate[3]. Most of this thesis does rely on the Pauli Principle being true, so for all intents and purposes we may as well take it to be a fundamental postulate.

2.3 The Many Body Wave Function

A single particle may in isolation be completely described by a wave function in Hilbert space. We will refer to this single particle state by

$$\phi_i(\mathbf{x}),$$

where $\mathbf{x}$ now contains all the relevant spacial quantum numbers as well as spin degrees of freedom. Additional quantum numbers needed to specify a given state are included in the subscript i.

In the presence of other particles it will make sense to define a wave function that describes the system as a whole Φ, and it is reasonable to assume that this function relies on each constituent single particle state. For a system of N particles, we then have an N-body wave function of the form

$$\Phi \equiv \Phi(\psi_0(\mathbf{x}_0), \phi_1(\mathbf{x}_1), \phi_2(\mathbf{x}_2)...\phi_N(\mathbf{x}_N)). \tag{2.3}$$

Since every single particle state has an associated Hilbert space, the system's state space will be a tensor product of each single particle state space

$$\mathcal{H}_0 \otimes \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes ... \otimes \mathcal{H}_N. \tag{2.4}$$

It is however possible for a subspace of the above to be sufficient. We may also refer to the totality of these spaces as the *Fock space* [7].

[3] In "Foundations of Physics" (1957) by Lindsay and Margenau [15] it is even claimed that *"There is no way of deducing Pauli's principle; its validity has to be inferred from its results [...]"*

This may lead us to guess that the system's wave function is a product of single particle states

$$\Phi_h = \phi_0 \otimes \phi_1 \otimes ... \otimes \phi_N = \phi_0(\mathbf{x}_0)\phi_1(\mathbf{x}_1)...\phi_N(\mathbf{x}_N). \tag{2.5}$$

The subscript h denotes that the above product is the so called *Hartree product* or *Hartree function*. It may also be written as

$$\prod_i \phi_j(\mathbf{x_i}.) \tag{2.6}$$

2.4 Antisymmetry

The Hartree product lacks one important feature that is needed to properly describe fermionic systems, namely the antisymmetrization described in postulate six in the previous subsection. The Hartree product is *completely uncorrelated*, meaning that the probability of finding fermions simultaneously at locations $x_0, x_1, ...$ is given by

$$|\phi_i(x_0)\phi_j(x_1)...|^2 dx_0 dx_1... = |\phi_i(x_0)|^2 dx_0 |\phi_j(x_1)|^2 dx_2...$$

.

This is just the product of each constituent particle wave function squared. The motion of these particles is in effect independent of each other.

In this thesis we will focus on electronic many-body systems, with an emphasis on the infinite electron gas in three dimensions. Electrons are thus our constituent particles. Electrons are identical and indistinguishable spin 1/2 fermions [14].

Although it does not immediately solve the antisymmetrization issue, we may assume that each way of permuting the Hartree function is an equally valid representation of the system, implying that also a linear combination of such permuted Hartree products is a valid representation of the system, that is

$$\Phi_p = \frac{1}{\sqrt{N!}} \sum_{\pi}^{N!} \hat{P}_\pi \Psi_h. \tag{2.7}$$

The subscript p refers to a given set of permutations and $N!$ serves as a normalizing constant. The operator $\hat{P}_\pi$ is the permutation operator, performing all $N!$ possible permutations of the Hartree product.

The Pauli Exclusion Principle is an interpretation of experimental facts, such as the pairing tendency of electrons, and the relation between stability and particle count in a variety of systems. It is commonly stated as *no two indistinguishable fermions may occupy the same quantum state.* When applied on the permuted Hartree function, we see that this principle does not apply in its current form.

To mend this shortcoming of the permuted Hartree function we require that interchanging two particles should also change the sign of the resulting function. Thus, an odd number of permutations should result in a sign change, while an even number of permutations should not. We may express this by

$$\Phi_{SD} = \frac{1}{\sqrt{N!}} \sum_{\pi}^{N!} \hat{P}_{\pi} (-)^{n(\pi)} \Psi_h \equiv \sqrt{N!} \mathcal{A} \Psi_h. \tag{2.8}$$

The subscript SD now denotes the so-called Slater determinant. The antisymmetrizer $\mathcal{A}$ is introduced to ease upcoming manipulations. One important property of the antisymmetrizer is that it commutes with the Hamiltonian [16]

$$\left[\mathcal{A}, \hat{H}\right] = \mathcal{A}\hat{H} - \hat{H}\mathcal{A} = 0. \tag{2.9}$$

Another one is that its square is simply itself [16], that is

$$\mathcal{A}^2 = \mathcal{A}. \tag{2.10}$$

Furthermore, conjugation results in

$$\mathcal{A}^{\dagger} = \mathcal{A}. \tag{2.11}$$

Another common representation of the Slater determinant is [7]

$$\Phi_{SD}(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\mathbf{x_1}) & \psi_2(\mathbf{x_1}) & \cdots & \psi_N(\mathbf{x_1}) \\ \psi_1(\mathbf{x_2}) & \psi_2(\mathbf{x_2}) & \cdots & \psi_N(\mathbf{x_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x_N}) & \psi_2(\mathbf{x_N}) & \cdots & \psi_N(\mathbf{x_N}) \end{vmatrix}. \tag{2.12}$$

2.5 The Hamiltonian

In classical mechanics, the total energy of a particle is called the *Hamiltonian*, and is written as [14]

$$H(x,p) = \frac{p^2}{2m} + V(x), \tag{2.13}$$

where p is the momentum, x is the position, m is the mass and $V(x)$ is the potential acting on a given particle. By substituting $p \rightarrow \frac{\hbar}{i}\frac{\partial}{\partial x}$, we find the corresponding single-particle quantum mechanical Hamiltonian to be [14]

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x). \tag{2.14}$$

2.6 Operators and matrix elements

The form of the potential $V(x)$ in (2.14) will be of special interest to us when working with many-body systems. For interacting systems, it is not sufficient for this operator to have a dependency on the coordinates of one particle at the time, since some parts of the potential energy are attributed to forces between the particles. Such forces normally depend on the distance between the particles, in other words two sets of coordinates at a time. In this context, it makes sense to separate terms that relate to a common potential from the terms that relate to multiple particles that interact. For a particle present in the system we may therefore write

$$\hat{V}(x_i) = \hat{v}(x_i) + \sum_j \hat{v}(x_i, x_j) + \sum_{j<k, jk\neq i} \hat{v}(x_i, x_j, x_k) + ... \equiv \hat{v}_i + \hat{v}_{ij} + \hat{v}_{ijk} + ... \tag{2.15}$$

The first term now relates to the common potential or external potential felt by all particles, the second term relates to forces that act on two particles at a time, and the third relates to forces that involves three particles at a time. We could extend this to include four-body forces or more complicated ones, but in this thesis we will limit ourselves to at most two-body interactions. We will define these interactions in more detail in 3.

It is convenient to include the kinetic energy in the one-body force, allowing us to thereby define a one-body part of the full many-body Hamiltonian as

$$\hat{h}_0(x_i) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x_i^2} + v(x_i). \tag{2.16}$$

We will assume that our system consists of identical particles such as electrons. There is then no need to assign any index to the mass, since it will be the same for all particles. The reason why we define such a one-body operator is that it is common to define single-particle eigenbases which are eigenstates (and thereby eigenvalues) of $\hat{h}_0$. With such a basis, we can in turn construct a (in principle infinite) set of orthogonal and normalized many-body Slater determinants. This basis of Slater determinants will in turn allows to define the exact many-body state function.

We may now write our general many-body Hamiltonian

$$\hat{H} = \sum_i \hat{h}_0(x_i) + \sum_{i<j} v(x_i, x_j). \tag{2.17}$$

For our Slater determinants to be a reasonable representation of our system, each Slater determinant must have an associated eigenenergy ϵ_{SD}, so that the Schrödinger equation (see Eq. 2.2)is fulfilled. The general expressions for these eigenvalues may be found by multiplying both sides of the Schrödinger equation by $\langle \Phi_{SD}|$, to find

$$\langle \Phi_{SD}|\hat{H}|\Phi_{SD}\rangle = \langle \Phi_{SD}|\epsilon_{SD}|\Phi_{SD}\rangle = \epsilon_{SD}\langle \Phi_{SD}|\Phi_{SD}\rangle = \epsilon, \tag{2.18}$$

since we have assumed that the Slater determinant is normalized so that $\langle \Phi_{SD}|\Phi_{SD}\rangle = 1$. This procedure allows us to find an expression for the eigenenergy associated with the Slater determinant, by evaluating the expectation value

$$\langle \Phi_{SD}|[\sum_i \hat{h}_0(x_i) + \sum_{i<j} v(x_i, x_j)]|\Phi_{SD}\rangle = \\ \langle \Phi_{SD}|\sum_i \hat{h}_0(x_i)|\Phi_{SD}\rangle + \langle \Phi_{SD}|\sum_{i<j} v(x_i, x_j)|\Phi_{SD}\rangle. \tag{2.19}$$

If the above Slater determinant is an ansatz for the ground state, the last equation defines what is normally called the reference energy. We will discuss this quantity in greater detail in later chapters. If we consider the form of the SD defined in (2.8) and the properties of the antisymmetrizer, we find that

$$\epsilon_{SD} = \langle \sqrt{N!}\mathcal{A}\phi_h|\hat{H}|\sqrt{N!}\mathcal{A}\phi_h\rangle = \\ N!\langle \phi_h|\mathcal{A}^\dagger \hat{H}\mathcal{A}\phi_h\rangle = N!\langle \phi_h|\hat{H}\mathcal{A}\phi_h\rangle = \\ N!\langle \phi_h|\hat{H}\Psi_0\rangle = N!\langle \phi_h|\hat{H}|\Phi_{SD}\rangle, \tag{2.20}$$

where ϵ_{SD} will later define our so-called *reference energy*, and be assigned the label ϵ_{ref}. Inserting our Hamiltonian we find that

$$\epsilon_{SD} = \langle \phi_h|\sum_i^N \hat{h}_0(\mathbf{x_i})|\Phi_0\rangle + \frac{N!}{2}\langle \phi_h|\sum_{i,j\neq i}^N \hat{v}(\mathbf{x_i}, \mathbf{x_j})|\Phi_{SD}\rangle, \tag{2.21}$$

and the problem is naturally separated in terms relating to the one-body part and the two-body part.

2.6.1 The one body problem

Since the one body operator only acts on one particle at a time, we find that

$$\hat{h}_0(\mathbf{x_i}) \prod_{j=1}^{N} \phi_j(\mathbf{x_j}) = \left(\prod_{j=1}^{N-1} \phi_j(\mathbf{x_j}) \right) \hat{h}_0(\mathbf{x_i}) \phi_i(\mathbf{x_i}). \tag{2.22}$$

We may write out the inner product as an integral over all quantum numbers for every particle $d\tau = \prod_i d\mathbf{x}_i$

$$N! \int d\tau \left(\prod_{j=1}^{N} \phi_i^*(\mathbf{x_i}) \right) \hat{h}_0(\mathbf{x_j}) \left(\prod_{k=1}^{N} \phi_k(\mathbf{x_k}) \right) = \\ \prod_{i \neq j}^{N-1} \Big(\int d\mathbf{x_i} |\phi_i(\mathbf{x_i})|^2 \Big) \int d\mathbf{x_j} \big(\phi_j^*(\mathbf{x_j}) \hat{h}_0(\mathbf{x_j}) \phi_j(\mathbf{x_j}) \big). \tag{2.23}$$

In the case of an orthonormal basis, it is apparent that the outcome of this integral is depends only on how the one-body operator acts on the targeted state since

$$\prod_{i \neq j}^{N-1} \Big(\int d\mathbf{x_i} |\phi_i(\mathbf{x_i})|^2 \Big) = 1. \tag{2.24}$$

For the terms beyond the unpermuted Hartree product we will either find that

$$\int d\mathbf{x_j} \big(\phi_j^*(\mathbf{x_j}) \hat{h}_0(\mathbf{x_j}) \phi_i(\mathbf{x_j}) \big) = 0, \tag{2.25}$$

or that

$$\int d\mathbf{x_j} \big(\phi_j^*(\mathbf{x_j}) \phi_i(\mathbf{x_j}) \big) = 0. \tag{2.26}$$

This means that the one-body contribution to the energy ϵ_h becomes

$$\epsilon_h = \sum_i \langle \phi_i | \hat{h}_0 | \phi_i \rangle. \tag{2.27}$$

2.6.2 The two body problem

For the two body problem, we now seek a solution to

$$\epsilon_v = \frac{N!}{2} \sum_{i,j \neq i}^{N} \langle \phi_h | v_{ij} | \mathcal{A} \phi_h \rangle. \tag{2.28}$$

If we first consider only the unpermuted hartree product to the right we will find that

$$\langle \phi_h | v_{ij} | \phi_h \rangle = \prod_{k\neq(i,j)}^{N} \Big(\int d\mathbf{x_k} |\phi_k(\mathbf{x_k})|^2 \Big) \int d\mathbf{x_i} d\mathbf{x_j} \Big(\phi_i^*(\mathbf{x_i})\phi_j^*(\mathbf{x_j}) \hat{v}(\mathbf{x_i},\mathbf{x_j}) \phi_i(\mathbf{x_i})\phi_j(\mathbf{x_j}) \Big). \tag{2.29}$$

The factor in front will vanish if our basis is properly normalized. For the singly permuted Hartree products we find instead

$$\langle \phi_h | v_{ij} | \hat{P}_{ij} \phi_h \rangle = \int d\mathbf{x_i} d\mathbf{x_j} \Big(\phi_i^*(\mathbf{x_i})\phi_j^*(\mathbf{x_j}) \hat{v}(\mathbf{x_i},\mathbf{x_j}) \phi_i(\mathbf{x_j})\phi_j(\mathbf{x_i}) \Big). \tag{2.30}$$

The two-body operator's ability to bring the permuted states into alignment with the unpermuted states results in the above not necessarily being zero, so we will need to include it in the final energy evaluation. We may rewrite it as

$$\epsilon_v = \frac{1}{2} \sum_{i,j\neq i} \langle \phi_h | \hat{v}_{ij} | (1 - \hat{P}_{ij}) \phi_h \rangle = \frac{1}{2} \sum_{i,j\neq i}^{N} \Big(\langle ij|\hat{v}|ij\rangle - \langle ij|\hat{v}|ji\rangle \Big). \tag{2.31}$$

Summarizing, the expectation value of our single Slater determinant is then

$$\epsilon_{SD} = \epsilon_h + \epsilon_v = \sum_i \langle i|\hat{h}_0(x_i)|i\rangle + \frac{1}{2} \sum_{i,j\neq i}^{N} \Big(\langle ij|\hat{v}|ij\rangle - \langle ij|\hat{v}|ji\rangle \Big). \tag{2.32}$$

2.7 The Aim of Many Body Quantum Theory

At this point, we should note that while the Slater determinant fulfills the criterions laid out so far, we have still not defined the single-particle states.

Depending on the form of the Hamiltonian in the Schrödinger equation, we may or may not have some idea of the form of the single-particle states. In many systems, it is possible to separate the Hamiltonian into terms describing the interaction between the particles and terms associated with the constituent particles.

$$\hat{H} = \hat{H}_{onebody} + \hat{H}_{interaction} \tag{2.33}$$

By ignoring the interaction terms, we may then try to solve the Schrödinger equation for the one-body problem.

Consider for example a number of interacting fermions in a common potential. When solving the corresponding one body problem, one typically obtains a set of wave functions that fulfill the Schrödinger equation, where each constituent function corresponds to a given energy state with an associated eigenenergy. The number of states may be infinite. By letting this set of states populate the Slater determinant in different ways we may construct an infinite number of Slater determinants, in effect spanning the Fock space defined in 2.4.

While the Fock space completely spans the space for the system's wave function, it is also possible for a subset of Slater determinants to do the same. Another possibility is that most of the system's wave function is contained in such a subset, so that a truncation of the Fock space may be made while retaining a decent approximation to the systems wave function.

For example, in cases where

$$\hat{H}_{onebody} \gg \hat{H}_{interaction}, \tag{2.34}$$

we may expect to be able to represent most of the system's ground state wave function with a small subset of all Slater determinants.

This is in essence the aim of many-body theory: we seek the set of Slater determinants that gives the most accurate representation of the system's wave function.

To obtain such a set we may choose a variety of paths, but common to all is the fact that the mathematical framework used so far would prove very tedious in deriving the upcoming expressions. We will therefore need to utilize the formalism commonly called *second quantization* or *the occupation number representation*, and for even more simplicity we will extend this to a diagrammatic formalism.

In the next chapters we present some of the basic elements of second quantization.

Chapter 3

Second Quantization

A brief review In this chapter we introduce the Second Quantization (or Number Representation) formalism, and we use it to derive the Hamiltonian for many-body problems with at most two-body interactions.

3.1 Second Quantization

When choosing the single-particle states to populate the Slater determinants, as discussed in the previous chapter, we will normally choose from a set of states that solves the corresponding one-body problem. Each of these states will have an energy eigenvalue in the one-body problem. Because of the Pauli Exclusions Principle, any state may only occur once in the Hartree product, since multiple identical functions will cause the Slater determinant to be zero.

When setting up the Slater determinant in this way, we create a model for our system that is commonly referred to as an *independent particle model*, since it is constructed from what we are referring to as uncorrelated states.

For a collection of N completely non-interacting identical fermions in a common potential, it then makes sense to define an energy level corresponding to the distribution of states where the N lowest energy states are occupied. The energy of the last occupied single-particle state (normally defined in terms of the energy of the single-particle states) is called the Fermi energy. The Slater determinant which is defined by filling all states below the Fermi energy is called the Fermi vacuum. This particular many-body state defines normally the ansatz for the ground state of the system and is often dubbed 'the reference state'. The expectation value of this state defines the so-called *reference energy* mentioned in the previous chapter, leaving it to various many-body methods like coupled cluster theory to produce reliable estimates for the so-called correlation energy. The latter is defined as the difference between the exact energy and the reference energy. It is normal to use single-particle states obtained from a Hartree-Fock calculation in order to construct the ansatz for the ground state and all other possible Slater determinants.

In a Dirac notation, we write the ansatz for the ground state (or the new vacuum reference state) as [7]

$$|\Phi_0\rangle. \tag{3.1}$$

In the same way, we may consider the "true" vacuum, where no states are occupied as

$$|0\rangle. \tag{3.2}$$

It is in this context that second quantization will come in handy. We will first define the framework for the true vacuum, and thereafter for the Fermi vacuum. This is similar to the order in which these topics are introduced in most text on many-body methods, see for example Shavitt and Bartlett's *Many-body methods in chemistry and physics* [7]. In the following sections we will basically just restate the formalism as it appears in the latter reference.

3.2 Creation and annihilation operators

In the following, we will for the sake of simplicity assume all Slater determinants to form an orthogonal and normalized many-body basis. This follows from the fact that the constituent single-particle states are orthogonal and normalized.

To be *occupied*, means that the single-particle state occurs in the Slater determinant, and to be *unoccupied* then naturally means the opposite. Beginning with the true vacuum state, we may then define operators that create or annihilate (remove) occupied states in the Slater determinant. We call these operators *creation*- and annihilation operators, respectively.

In the second quantization formalism, we denote creation operators as $\hat{a}_p^\dagger$, where the subscript denotes the a particular state and its relevant quantum numbers. When acting on the true vacuum state, we have

$$\hat{a}_p^\dagger|0\rangle = |p\rangle, \tag{3.3}$$

indicating that we have a single-particle state labeled by the index p. The annihilation operator is defined as

$$\hat{a}_p|p\rangle = |0\rangle, \tag{3.4}$$

meaning that a particle is removed from the state identified by the label p, bringing us back to the true vacuum state. When a creation operator acts on a state which is already occupied, the Pauli exclusion principle requires that

$$\hat{a}_p^\dagger|pqr\rangle = 0, \tag{3.5}$$

where we here have defined a three-body state identified by labels pqr. Attempting to add a new particle with the quantum numbers given by the label p, must give zero in order to obey the Pauli principle. Annihilating states that are not present in the SD should also yield zero, that is

$$\hat{a}_p|qrs\rangle = 0. \tag{3.6}$$

Again, we have here defined a three-body state identified by the labels qrs. We may combine these operators to define the so called *number* operator $\hat{n}$ that lets us evaluate if a given state is occupied or not, as well as count all occupied states with the operator $\hat{N}$:

$$\hat{N} = \sum_p \hat{a}_p^\dagger \hat{a}_p \equiv \sum_p \hat{n}_p. \tag{3.7}$$

Its action on a given Slater determinant results in the number of particles present as eigenvalue.

3.3 Strings of operators

Constructing a slater determinant from a set of orbitals may now be performed by

$$\hat{a}_i^\dagger \hat{a}_j^\dagger ... \hat{a}_q^\dagger \hat{a}_r^\dagger |0\rangle = |ij...qr\rangle. \tag{3.8}$$

While the subcripts refer to the states present in the Slater determinant, the order in which these appear tells us which particles are occupying each state. If we need to explicitly express which particle occupies which state, we may give each particle a number and write

$$\hat{a}^{\dagger}_{i,1}\hat{a}^{\dagger}_{j,2}...\hat{a}^{\dagger}_{q,N-1}\hat{a}^{\dagger}_{r,N}|0\rangle = |i(1)j(2)...q(N-1)r(N)\rangle. \tag{3.9}$$

Since the permutation of occupied states changes the sign of the Slater determinant (2.8), so should the permutation of operators. Permuting operators acting on a Slater determinant is basically the same operation as permuting particles in the Slater determinant. We have

$$\hat{P}|\Phi\rangle = (-1)^{\sigma(P)}|\Phi\rangle, \tag{3.10}$$

which is equivalent to

$$\hat{a}^{\dagger}_{i}\hat{a}^{\dagger}_{j}...\hat{a}^{\dagger}_{m}\hat{a}^{\dagger}_{n}...\hat{a}^{\dagger}_{q}\hat{a}^{\dagger}_{r}|0\rangle = -\hat{a}^{\dagger}_{i}\hat{a}^{\dagger}_{j}...\hat{a}^{\dagger}_{n}\hat{a}^{\dagger}_{m}...\hat{a}^{\dagger}_{q}\hat{a}^{\dagger}_{r}|0\rangle. \tag{3.11}$$

Because of orthonormality, we will find that the the expectation value of an annihilation operator is

$$\langle\Phi'|\hat{a}_p|\Phi\rangle = \begin{cases} \pm 1, & \hat{n}_p(\Phi) = 1, \hat{n}_p(\Phi') = 0, \hat{n}_i(\phi) = \hat{n}_i\Phi'(i \neq p) \\ 0, & \text{else.} \end{cases} \tag{3.12}$$

This means that the expectation value will be nonzero if Φ' and Φ have the same occupied states, except when the state p occurs in Φ but not in Φ'. From this, we may deduce that

$$\langle\Phi'|\hat{a}_p|\Phi\rangle = \langle\hat{a}^{\dagger}_p\Phi'|\Phi\rangle, \tag{3.13}$$

which shows that $\hat{a}_p$ is the adjoint to $\hat{a}^{\dagger}_p$.

3.4 Anticommutation relations

To enable us to evaluate expectation values for strings of operators, we will need to be able to manipulate these strings. Such manipulations will at some abstract level be involved at a later stage in this thesis, for example when deriving the coupled cluster equations.

While we have already defined the interchange or permutations of two or more operators in strings of exclusively creation or annihilation operators (3.10), we will run into complications when the strings contain a mix of these operators. To this end, we will utilize the anticommutator, defined as

$$[\hat{A}, \hat{B}]_{+} \equiv \hat{A}\hat{B} + \hat{B}\hat{A}. \tag{3.14}$$

From what we already have discussed (see Eq. (3.11)), it is follows that

$$[\hat{a}_p, \hat{a}_q]_{+} = \hat{a}_p\hat{a}_q + \hat{a}_q\hat{a}_p = \hat{a}_p\hat{a}_q - \hat{a}_p\hat{a}_q = 0, \tag{3.15}$$

and

$$[\hat{a}_p^{\dagger}, \hat{a}_q^{\dagger}]_{+} = \hat{a}_p^{\dagger}\hat{a}_q^{\dagger} + \hat{a}_q^{\dagger}\hat{a}_p^{\dagger} = \hat{a}_p^{\dagger}\hat{a}_q^{\dagger} - \hat{a}_p^{\dagger}\hat{a}_q^{\dagger} = 0. \tag{3.16}$$

By evaluating how the anticommutator for mixed operators acts on certain Slater determinants, we find that

$$[\hat{a}_p^{\dagger}, \hat{a}_q]_{+} = [\hat{a}_p, \hat{a}_q^{\dagger}]_{+} = \delta_{p,q}. \tag{3.17}$$

This results allows us to rewrite certain strings of operators, since it means that for example

$$\hat{a}_p^{\dagger}\hat{a}_q = \delta_{p,q} - \hat{a}_q\hat{a}_p^{\dagger}. \tag{3.18}$$

3.5 Inner products and operators

The vacuum state is assumed to be normalized, that is

$$\langle 0|0\rangle = 1 \tag{3.19}$$

With the framework laid out so far, we are now able to evaluate inner products of multiple Slater determinants. Consider the following two Slater determinants

$$\hat{a}_p^{\dagger}\hat{a}_q^{\dagger}...\hat{a}_N^{\dagger}|0\rangle \tag{3.20}$$

and

$$\hat{a}_r^{\dagger}\hat{a}_s^{\dagger}...\hat{a}_M^{\dagger}|0\rangle. \tag{3.21}$$

Using the fact that the creation operator is the adjoint of the annihilation operator, we have

$$\langle 0|\hat{a}_M ... \hat{a}_s \hat{a}_r \hat{a}_p^\dagger \hat{a}_q^\dagger ... \hat{a}_N^\dagger |0\rangle. \tag{3.22}$$

For Eq. (3.22), we may find that the operators line up perfectly, meaning that

$$(r = p), (s = q), ..., (M = N) \rightarrow \langle 0|\hat{a}_M ... \hat{a}_s \hat{a}_r \hat{a}_p^\dagger \hat{a}_q^\dagger ... \hat{a}_N^\dagger |0\rangle = 1. \tag{3.23}$$

In other cases, the calculation of this inner product (also commonly called *matrix product* in the literature [7]), becomes a matter of reorganizing the creation and annihilation operators in such a manner that we end up with something we are able to evaluate. As will become apparent, a good strategy is to move all creation operators to the left of the annihilation operators, since this inner product will be zero.

One very important aim is to be able to evaluate the expectation value of the Hamiltonian, and as such we will also need to extend the formalism to include operators. We define the relevant operators in the following, focusing only on one-body and two-body operators since our Hamiltonian will contain at most two-body interactions.

The One-body Operator A one-body operator is defined as [7]

$$\hat{F} \equiv \sum_{k,l} f_{k,l} \hat{a}_k^\dagger \hat{a}_l = \sum_{k,l} \langle k|\hat{f} l\rangle \hat{a}_k^\dagger, \hat{a}_l. \tag{3.24}$$

where $\hat{f}$ may for example be identical to $\hat{h}_0$ which we used in Chapter 2. s It is useful to calculate the expectation value of this operator. We consider here the inner product

$$\sum_{k,l} f_{k,l} \langle 0|\hat{a}_M ... \hat{a}_s \hat{a}_r (\hat{a}_k^\dagger \hat{a}_l) \hat{a}_p^\dagger \hat{a}_q^\dagger ... \hat{a}_N^\dagger |0\rangle. \tag{3.25}$$

Depending on the Slater determinants present, we have four different outcomes of the above:

1. If we have perfect line up as in Eq. (3.23), we find that $\langle \hat{F} \rangle = \sum_i^N f_{i,i}$
2. If all single-particle states involved occur in both Slater determinants but in no particular order, we find that $\langle \hat{F} \rangle = (-1)^{\sigma(\hat{P})} \sum_i^N f_{i,i}$
3. If all except one single-particle state (one *non-coincidence* [7]) occurs in both Slater determinants, the one body operator may act on the Slater determinants in such

a way that when encountering the non-coincidence it brings it to coincide

$$(r=p),(s=q),...,(m\neq n),...,(M=N)\rightarrow$$
$$\sum_{k,l} f_{k,l}\langle 0|\hat{a}_M...\hat{a}_m...\hat{a}_s\hat{a}_r(\hat{a}_k^\dagger\hat{a}_l)\hat{a}_p^\dagger\hat{a}_q^\dagger...\hat{a}_n...\hat{a}_N^\dagger|0\rangle = (-1)^{\sigma(\hat{P})}f_{m,n} \quad (3.26)$$

4. If there is more than one non-coincidence, the resulting Slater determinants will be orthogonal so that $\langle\hat{F}\rangle = 0$.

The Two-body Operator We define the two-body operator within the second quantization formalism [7] as

$$\hat{G} = \frac{1}{2}\sum_{i,j,k,l}\langle i(1)j(2)|g_{12}|k(1)l(2)\rangle\hat{a}_i^\dagger\hat{a}_j^\dagger,\hat{a}_l\hat{a}_k. \quad (3.27)$$

Where we used the definition in Eq. 3.9 so that in the bra side of the inner product, particle one is in state i and particle two is in state j, while in the ket side we have particle one in state k and particle two in state l.

Similar to the case of the one-body operator, we need to know the expectation value of this operator in a general Fock space. We consider thus the inner product

$$\frac{1}{2}\sum_{i,j,k,l}\langle i(1)j(2)|g_{12}|k(1)l(2)\rangle\langle 0|\hat{a}_M...\hat{a}_s\hat{a}_r(\hat{a}_i^\dagger\hat{a}_j^\dagger\hat{a}_l\hat{a}_k)\hat{a}_p^\dagger\hat{a}_q^\dagger...\hat{a}_N^\dagger|0\rangle. \quad (3.28)$$

If we have perfectly aligned states in the two Slater determinants, we obtain

$$\langle\Phi|\hat{G}|\Phi\rangle = \sum_{i<j,ij\in\Phi}(\langle ij|\hat{g}|ij\rangle - \langle ij|\hat{g}|ij\rangle) \equiv \sum_{i<j,ij\in\Phi}\langle ij||ij\rangle. \quad (3.29)$$

This is the so-called *antisymmetric matrix element*, and the two terms appearing in this element are by convention named the *direct-* and *exhange* term, respectively.

For the case where the two Slater determinants differ by one single-particle state only, we find that

$$\langle\Phi'|\hat{G}|\Phi\rangle = \sum_{j\in\Phi}\langle i'(p)j||i(p)j\rangle, \quad (3.30)$$

meaning that particle p in this case does not occupy identical single particle states in the bra- and ket side of the inner product since $i'\neq i$.

If instead two single-particle states differ, we obtain

$$\langle \Phi' | \hat{G} | \Phi \rangle = \langle i'(p) j'(q) || i(p) j(q) \rangle . \tag{3.31}$$

With more than two differing single-particle states, the above matrix elements are zero since our interaction acts at most on two particles at the time.

The Hamiltonian We have now the means to write down a Hamiltonian in a second quantized form containing both one-body and two-body operators. This is the form of the Hamiltonian that will be the studied in this thesis. We will define later the explicit form of the Hamiltonian. For now, we will utilize the fact that

$$\langle ij || kl \rangle = -\langle ij || lk \rangle = \langle ji || lk \rangle = \langle ji || kl \rangle$$

to rewrite the Hamiltonian into the following form

$$\hat{H} = \sum_{ij} h_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{4} \sum_{i,j,k,l} \langle ij || kl \rangle \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k . \tag{3.32}$$

3.6 Normal ordering

As previously mentioned, reorganizing strings of operators so that all annihilation operators are to the right of the creation operators will be a good strategy when evaluating inner products. The reason for this is that this sequence of operators must yield zero when evaluated as the expectation value in the vacuum state, and in the process of this reorganization we will produce all nonzero contributions as Kronecker deltas in accordance with Eq. (3.17).

For this reason, the process of reorganizing strings of operators into this so called *normal ordered* sequence is of special interest when doing calculations on many-body wave functions. While the diagrammatic approach introduced by Feynman [7] will be our main weapon of choice when dealing with such problems at a later stage in this thesis, we will first treat this using *Wick's Theorem* [17]. This straightforward approach has the advantage of being easily translated into computer algebra, as utilized in for example the *Secondquant* package included in the SymPy package of Python [18]. Although we have already introduced the basic operations needed in this process, we will in this section see that it may be greatly simplified.

The normal ordered product (or simply *normal product*) is commonly denoted by either an "n" followed by square brackets, or curly brackets [7]

$$n[\hat{A}\hat{B}...] = \{\hat{A}\hat{B}...\}. \tag{3.33}$$

3.7 Contractions

We define the process of *contracting* two (creation or annihilation) operators by

$$\contraction{}{A}{}{B}AB \equiv AB - n[AB]. \tag{3.34}$$

For the operators discussed so far, we will only encounter four different situations when performing such contractions. Either we have the three cases where the contracted operators are already basically normal ordered

$$\contraction{}{\hat{a}^\dagger_p}{}{\hat{a}^\dagger_q}\hat{a}^\dagger_p\hat{a}^\dagger_q = \contraction{}{\hat{a}_p}{}{\hat{a}_q}\hat{a}_p\hat{a}_q = \contraction{}{\hat{a}^\dagger_p}{}{\hat{a}_q}\hat{a}^\dagger_p\hat{a}_q = 0, \tag{3.35}$$

or we have the singular nonzero case, where

$$\contraction{}{\hat{a}_p}{}{\hat{a}^\dagger_q}\hat{a}_p\hat{a}^\dagger_q = [\hat{a}_p, \hat{a}^\dagger_q]_+ = \delta_{p,q}. \tag{3.36}$$

3.8 Wick's theorem

Wick's theorem was introduced by Gian-Carlo Wick in 1950 [17], and states that

Theorem 3.1 (The time independent Wick's theorem)**.** *A product of a string of creation and annihilation operators is equal to their normal product plus the sum of all possible normal products with contractions.*

We have already seen that the expectation value of any normal product on the vacuum state will be zero, so this basically means that only the possible fully contracted normal products will contribute to the expectation value. As stated in Eq. (3.35), many of these contractions will also be zero, so we need only to consider the possible non-zero contractions.

This is a great simplification of the tedious reorganization of operators we have previously encountered, and it is an important tool when working with many-body wave states.

3.9 Particles and holes

We have previously briefly mentioned the *Fermi vacuum* (also commonly called the *reference state*) (3.1), although we up to this point have mainly considered operators acting on the true vacuum state (3.2). When we at a later stage will perform actual calculations, we will typically have a fixed number (N) of particles that may be represented by a number of fixed sized Slater determinants. In this context, it will make much more sense to define the second quantization formalism with respect to the Fermi vacuum, namely the Slater determinant where the N particles occupy the N lowest energy eigenstates of the single-particle basis.

We will then need to redefine normal ordering, creation and annihilation operators, and Wick's theorem with respect to this new reference state.

We will define the Fermi level to be the N'th lowest energy level where N is the number f particles in our system.

By convention, we will now refer to unoccupied states up to the Fermi level as holes (or hole states), and label them with the letters $i, j, k, ...$. Occupied states above the Fermi level will be referred to as *particles* (or particle states), and will be labeled by $a, b, c, ...$. Creation and annihilation operators will now behave differently depending on whether they target states above or below the Fermi level.

A *pseudo particle* creation operator may act on the reference state to either remove an occupied state below the Fermi level, thus *creating a hole*, or it may create a particle in an unoccupied particle state above the Fermi level. Operators acting on particle states will then behave the same as before, while operators acting on holes will have the opposite effect as previously discussed.

For example, we may excite the reference state by annihilating a state i below the Fermi level (thus creating a hole), and thereafter create a particle state above the Fermi level. By convention, such a process may be written

$$\hat{a}_a^\dagger \hat{a}_i^\dagger |\Phi_0\rangle = |\Phi_i^a\rangle, \tag{3.37}$$

Note that both operators above are given in terms of creation operators, denoting that they are pseudo-particle creation operators [7]. This inversion in notation for operators acting on states below the Fermi level is motivated by the fact that the way we previously defined the normal product no longer will yield zero when evaluated on the reference state. If we instead define our normal product to have all pseudo-creation operators to the left of all pseudo-annihilation operators, we achieve the same behavior as for the true vacuum.

In this new context, Wick's theorem will remain basically unchanged, apart from that the only possible nonzero contractions will be

$$\overline{\hat{a}_a\hat{a}_b^\dagger} = \delta_{a,b}, \tag{3.38}$$

and

$$\overline{\hat{a}_i^\dagger\hat{a}_j} = \delta_{i,j}. \tag{3.39}$$

3.10 The normal-ordered Hamiltionian and Wick's generalized theorem

By normal ordering the Hamiltonian we will achieve two things; we will derive expressions that are especially suitable for dealing with states in Fock space (post Hartree Fock calculations), and we will be able to utilize the so-called generalized Wick's theorem, that is[1]

Theorem 3.2 (The generalized time independent Wick's theorem)**.** *When considering products of strings of normal ordered creation and/or annihilation operators, we need only consider fully contracted contributions between the normal ordered strings. Internal contractions in the products will yield zero.*

To this end, we will benefit from rewriting the Hamiltonian into normal ordered form.

3.10.1 Normal-ordered one-body operator

Acting on all states (particles and holes), the one-body operator is rewritten as

$$\hat{F} = \sum_{p,q} f_{p,q}\hat{a}_p^\dagger\hat{a}_q. \tag{3.40}$$

Using Wick's theorem, we find that

$$\hat{F} = \sum_{p,q} f_{p,q}(\{\hat{a}_p^\dagger\hat{a}_q\} + \delta_{p,q\leq Fermilevel}) \equiv \hat{F}_N + \sum_i f_{i,i}. \tag{3.41}$$

[1] Note that this is not the actual formulation, see [7, p.86] for the full theorem. For our purpose this formulation is sufficient, since it means that when we encounter expectation values involving normal ordered strings, it will efficiently reduce the number of evaluations needed.

If we treat particle and hole states separately in the normal-ordered one-body operator, we find

$$\hat{F}_N = \sum_{i,j} f_{i,j}\hat{a}_j\hat{a}_i^\dagger + \sum_{a,b} f_{a,b}\hat{a}_a^\dagger\hat{a}_b + \sum_{a,i} f_{i,a}\hat{a}_i^\dagger\hat{a}_a + \sum_{a,i} f_{a,i}\hat{a}_i\hat{a}_a^\dagger. \tag{3.42}$$

The remaining term is now simply the expectation value associated with each single-particle state below the Fermi level.

3.10.2 Two-body operator

For the two-body operator we find that

$$\hat{G} = \frac{1}{2}\sum_{p,q,r,s}\langle pq|g|rs\rangle\hat{a}_p^\dagger\hat{a}_q^\dagger\hat{a}_s\hat{a}_r. \tag{3.43}$$

Again, by Wick's theorem, we find [7, p.82]

$$\hat{G} = \hat{G}_N + \frac{1}{2}\sum_{p,q}(\sum_i\langle pi|\hat{g}|qi\rangle_{AS}\{\hat{a}_p^\dagger\hat{a}_q\}) + \frac{1}{2}\sum_{ij}\rangle ij|\hat{g}|ij\rangle_{AS}. \tag{3.44}$$

The middle term is a one-body term associated with a two-body operator [7].

The normal-ordered term above may also be written in a way that specifies particles and holes, that is

$$\begin{aligned}\hat{G}_N = &\sum_{abcd}\langle ab|\hat{g}|cd\rangle\{\hat{a}_a^\dagger\hat{a}_b^\dagger\hat{a}_d\hat{a}_c\} + \sum_{ijkl}\langle ij|\hat{g}|kl\rangle\{\hat{a}_i^\dagger\hat{a}_j^\dagger\hat{a}_l\hat{a}_k\} + \sum_{aibj}\langle ij|\hat{g}|bj\rangle\{\hat{a}_a^\dagger\hat{a}_i^\dagger\hat{a}_j\hat{a}_b\} \\ &+ \sum_{abci}\langle ab|\hat{g}|ci\rangle\{\hat{a}_a^\dagger\hat{a}_b^\dagger\hat{a}_i\hat{a}_c\} + \sum_{iajk}\langle ia|\hat{g}|jk\rangle\{\hat{a}_i^\dagger\hat{a}_a^\dagger\hat{a}_k\hat{a}_j\} + \sum_{aibc}\langle ai|\hat{g}|bc\rangle\{\hat{a}_a^\dagger\hat{a}_i^\dagger\hat{a}_b\hat{a}_c\} \\ &+ \sum_{ijka}\langle ij|\hat{g}|ka\rangle\{\hat{a}_i^\dagger\hat{a}_j^\dagger\hat{a}_a\hat{a}_k\} + \sum_{abij}\langle ab|\hat{g}|ij\rangle\{\hat{a}_a^\dagger\hat{a}_b^\dagger\hat{a}_j\hat{a}_i\} + \sum_{ijab}\langle ij|\hat{g}|ab\rangle\{\hat{a}_i^\dagger\hat{a}_j^\dagger\hat{a}_b\hat{a}_a\}.\end{aligned} \tag{3.45}$$

3.10.3 The normal-ordered Hamiltonian

From the above derivations, we may now express the full Hamiltonian as

$$\hat{H} \;=\; \hat{F}_N + \frac{1}{2}\sum_{p,q}\sum_i\langle pi|\hat{v}|qi\rangle_{AS}\{\hat{a}_p^\dagger\hat{a}_q\} + \hat{V}_N + \frac{1}{2}\sum_{ij}\rangle ij|\hat{v}|ij\rangle_{AS} + \sum_i f_{i,i}, \tag{3.46}$$

where we conventionally renamed $\hat{G} \to \hat{V}$ to identify the interaction. We have previously derived the expectation value of one- and two-body operators for the reference state in

Eqns. (3.29) and (3.25), where we found that

$$\langle \Phi_0 | \hat{H} | \Phi_0 \rangle = \sum_{i \in \Phi} f_{ii} + \sum_{i<j, ij \in \Phi} \langle ij || ij \rangle. \tag{3.47}$$

Comparing Eq. (3.46) to Eq. (3.47), we see that the full Hamiltonian may be rewritten

$$\hat{H} = (\hat{F}_N + \hat{V}_N) + \langle \Phi_0 | \hat{H} | \Phi_0 \rangle \equiv \hat{H}_N + \langle \Phi_0 | \hat{H} | \Phi_0 \rangle. \tag{3.48}$$

The normal ordered Hamiltonian is defined as

$$\hat{H}_N = \hat{H} - \langle \Phi_0 | \hat{H} | \Phi_0 \rangle. \tag{3.49}$$

3.11 The correlation energy

We may interpret the result in Eq. (3.49) as follows: if we seek the expectation value of the normal ordered Hamiltonian on the true ground state wave function, we find that

$$\langle \Psi | \hat{H}_N | \Psi \rangle = \langle \Psi | \hat{H} | \Psi \rangle - \langle \Phi_0 | \hat{H} | \Phi_0 \rangle \equiv \langle \Psi | \hat{H} | \Psi \rangle - E_{ref}, \tag{3.50}$$

meaning that the energy found from calculating the above equals the true ground state energy of the system minus the part of the energy associated with the ground state ansatz Slater determinant Φ_0 (the reference state or the Fermi vacuum). We may express this as

$$\langle \Psi | \hat{H}_N | \Psi \rangle \equiv \Delta E = E - E_{ref}. \tag{3.51}$$

In cases where f_{pq} is diagonal, we will refer to the energy associated with the normal ordered Hamiltonian as the *correlation energy*[2], while the energy associated with the reference state will be referred to as the *reference energy.* The rationale of this ordering is, as well as the way they naturally occur in the equations, that the reference energy is normally found by seeking a decent reference state for the system, while the correlation energy may be sought in a variety of ways building upon this first approximation. Our aim is to obtain a as precise possible expression for the correlation energy using advanced first principle methods like coupled cluster theory.

In cases where we may justly treat the interaction as a small perturbation in our system (Eq. (2.34)), we may expect the reference energy to good approximation to the true ground state energy.

[2] See also Figure 5.1.

Chapter 4

Diagrammatic notation

4.1 Diagrammatic notation - background

While second quantization provides us with the foundation for working out expressions within the various many-body methods, it may prove cumbersome to derive comprehensive equations such as the coupled cluster equations using only second quantized operators and Wick's generalized theorem. To this end, the *diagrammatic notation*, originally introduced in the context of quantum electrodynamics by Richard Feynman in 1959 [7, p.1], will simplify calculations considerably. In addition, diagrams allow us to easily identify features of terms in the equations such as which terms vanish or should be naturally grouped together [7].

4.2 The Slater Determinant

Due to their origin in quantum field theory, diagrams normally express a time-ordered sequence of operators where time evolves in the upwards direction. This might seem strange when applied to the time-independent Schrödinger equation, but we shall only interpret the time axis as the sequence in which the different operators are applied. A starting point for the calculations will be the Fermi vacuum, expressed simply by a blank space. As we have already shown (see for example Eqn. 3.49, normal ordering of the Hamiltonian with respect to the Fermi vacuum will allow us mainly focus on excited states (represented by Slater determinants in our representation). These states, or Slater determinants, may be expressed simply by vertical lines, assigned either upwards- or downwards arrows to indicate particle or holes, respectively, as in Figs. 4.1 or 4.2.

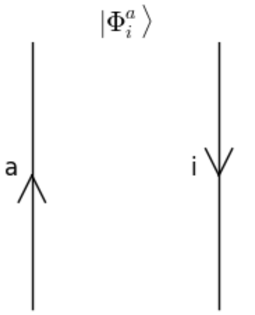


FIGURE 4.1: A Slater determinant with one hole state and one particle state.

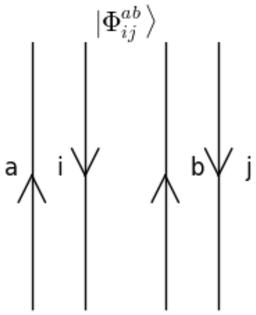


FIGURE 4.2: A Slater determinant with two hole states and two particle states.

4.3 Operators

We will use horizontal dashed lines to represent operators such as terms in the normal-ordered hamiltonian. While the one-body operator will have two lines entering and/or leaving, the two-body operator will have four lines entering and/or leaving. The lines entering from below represent the annihilation of pseudo particles, while lines exiting above represent creation of pseudo particles. Following this logic, we list the normal ordered one-body operator of Eq. (3.42) and the normal ordered two-body operator of Eq. (4.3) diagrammatically, as shown in Figs. 4.3, 4.4 and 4.5.

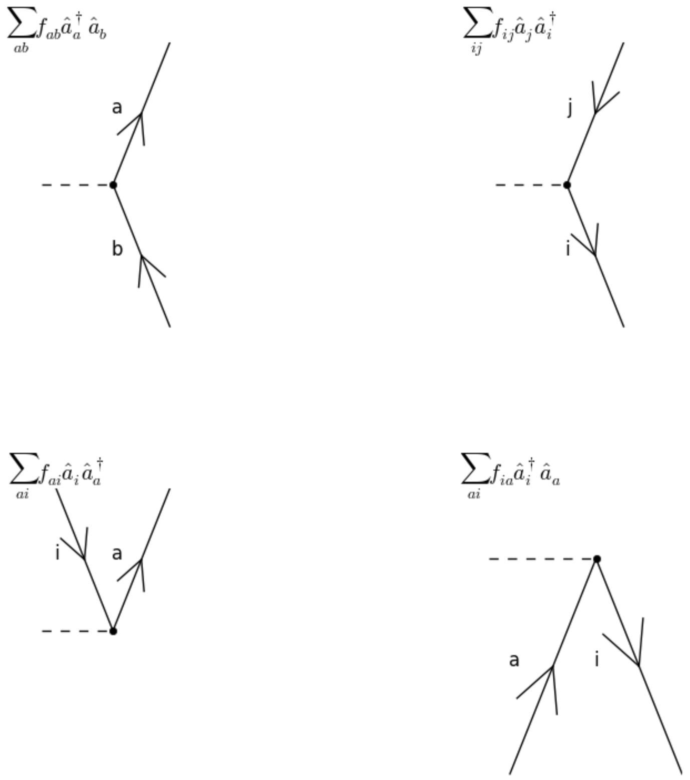
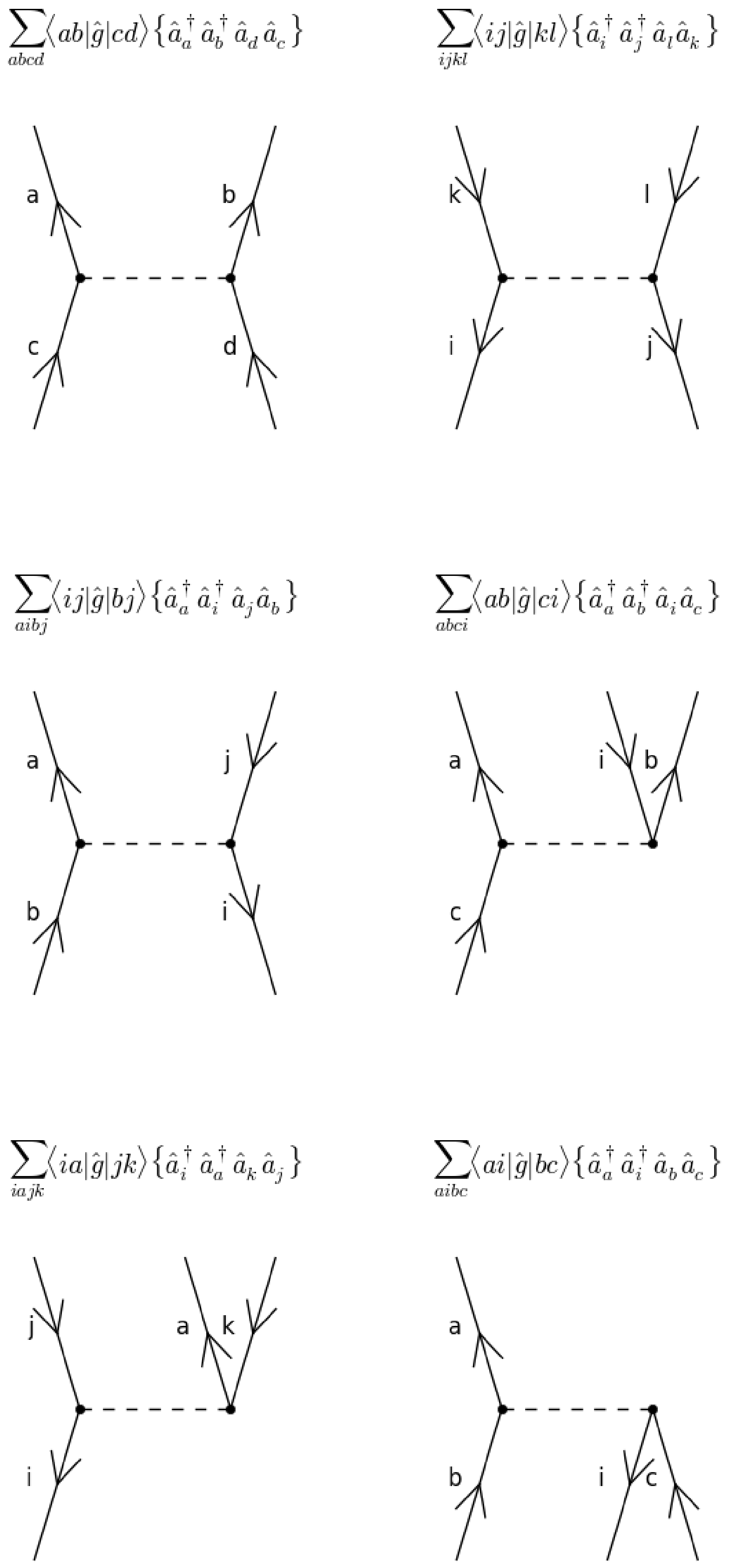

$\sum_{ab} f_{ab}\hat{a}_a^\dagger \hat{a}_b$
a
b
$\sum_{ij} f_{ij}\hat{a}_j\hat{a}_i^\dagger$
j
i
$\sum_{ai} f_{ai}\hat{a}_i\hat{a}_a^\dagger$
i
a
$\sum_{ai} f_{ia}\hat{a}_i^\dagger \hat{a}_a$
a
i

FIGURE 4.3: A normal ordered one-body operator with its mathematical expressions.



$$\sum_{abcd}\langle ab|\hat{g}|cd\rangle\left\{\hat{a}_a^\dagger\,\hat{a}_b^\dagger\,\hat{a}_d\,\hat{a}_c\right\}$$

$$\sum_{ijkl}\langle ij|\hat{g}|kl\rangle\left\{\hat{a}_i^\dagger\,\hat{a}_j^\dagger\,\hat{a}_l\hat{a}_k\right\}$$

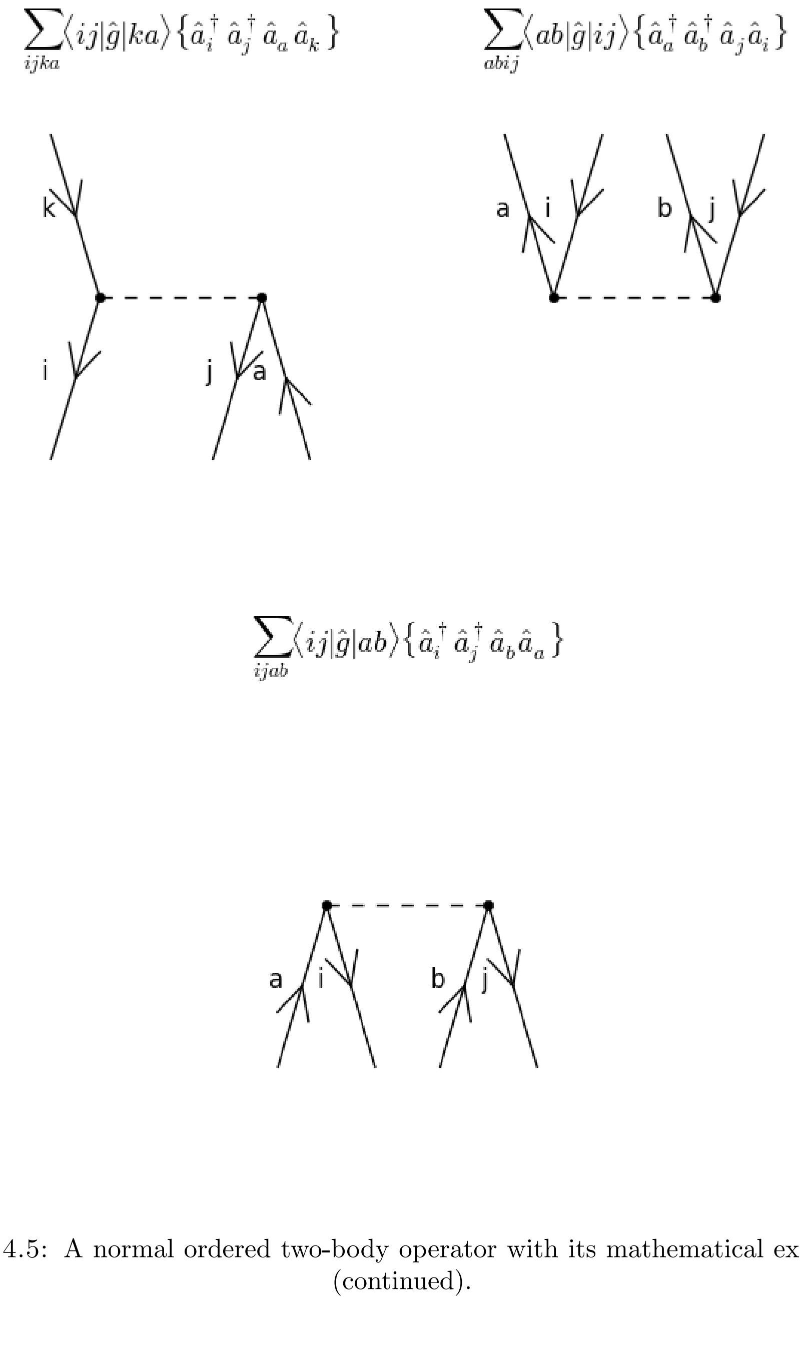
$$\sum_{aibj}\langle ij|\hat{g}|bj\rangle\left\{\hat{a}_a^\dagger\,\hat{a}_i^\dagger\,\hat{a}_j\hat{a}_b\right\}$$

$$\sum_{abci}\langle ab|\hat{g}|ci\rangle\left\{\hat{a}_a^\dagger\,\hat{a}_b^\dagger\,\hat{a}_i\hat{a}_c\right\}$$

$$\sum_{iajk}\langle ia|\hat{g}|jk\rangle\left\{\hat{a}_i^\dagger\,\hat{a}_a^\dagger\,\hat{a}_k\,\hat{a}_j\right\}$$

$$\sum_{aibc}\langle ai|\hat{g}|bc\rangle\left\{\hat{a}_a^\dagger\,\hat{a}_i^\dagger\,\hat{a}_b\hat{a}_c\right\}$$

Figure 4.4: A normal ordered two-body operator with its mathematical expressions.



$$\sum_{ijka}\langle ij|\hat{g}|ka\rangle\{\hat{a}_i^\dagger\,\hat{a}_j^\dagger\,\hat{a}_a\,\hat{a}_k\}$$

$$\sum_{abij}\langle ab|\hat{g}|ij\rangle\{\hat{a}_a^\dagger\,\hat{a}_b^\dagger\,\hat{a}_j\hat{a}_i\}$$

$$\sum_{ijab}\langle ij|\hat{g}|ab\rangle\{\hat{a}_i^\dagger\,\hat{a}_j^\dagger\,\hat{a}_b\hat{a}_a\}$$

Figure 4.5: A normal ordered two-body operator with its mathematical expressions (continued).

4.4 Contractions

Diagrams allow us to represent contractions in a straight forward manner, by connecting lines from Slater determinants to operators or between operators. For example, we may consider the contraction of a singly excited Slater determinants with a term in the normal ordered one-body operator, given by

$$(\sum_{bc} f_{bc}\{\hat{a}_b^\dagger \hat{a}_c\})|\Phi_i^a\rangle = (\sum_{bc} f_{bc}\{\hat{a}_b^\dagger \hat{a}_c\})\{\hat{a}_a^\dagger \hat{a}_i\}|\Phi_0\rangle = \sum_{bc} f_{bc}\delta_{ac}|\Phi_i^b\rangle, \tag{4.1}$$

where the contraction occurs between the indices in the Kronecker delta, in terms of the diagrammatic contraction performed in Fig. 4.6.

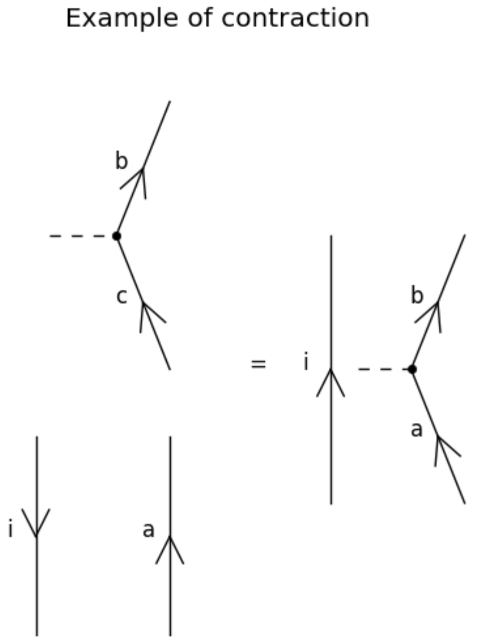


FIGURE 4.6: A contraction of a one-body operator and a singly excited Slater determinant.

The diagrams resulting from this process may in turn be interpreted back into mathematical expressions that we may use in implementations of the various many-body methods.

4.5 Interpreting diagrams

The various many body methods which will be introduced in the upcoming chapters rely on operator expressions and contractions. The basic idea of using diagrams to derive such expressions may be outlined as follows. For a given method we shall see that the contracted expressions that are included in the evaluation of say the correlation energy, may be visually represented by a set of diagrams. The situation before these contractions may also be visually represented by combinations of such uncontracted vertices as shown in Figs. 4.4, 4.5, 4.3 (and so on).

By defining consistent rules for how the contracted diagrams are obtained from the uncontracted vertices, we may skip the mathematical contraction altogether, derive the various resulting terms, and interpret them back into mathematical expressions by rules that ensure that all factors and distinct features of the expressions are preserved.

Luckily, such consistent rules are present in the literature [7] for many of the methods we discuss in the upcoming chapter.

Chapter 5

Methodology

5.1 Many-body Methods

As previously discussed in Chapter 2, the aim of many-body quantum mechanics is to find either the exact, or a reliable approximation to the many-body systems wave function Ψ. In methods like coupled cluster theory, this wave function resides in Fock space, a space spanned by all possible Slater determinants constructed by the single-particle states from the corresponding one-body problem. This wave function may be written (for N particles)

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ai} c_{ai}|\Phi_i^a\rangle + \sum_{abij} c_{abij}|\Phi_{ij}^{ab}\rangle + ... + \sum_{ab...N_a ij...N_i} c_{ab...N_a ij...N_i}|\Phi_{ij...N_i}^{ab...N_a}\rangle. \tag{5.1}$$

To determine the full wave function we therefore need a complete single-particle basis as well as the means to project it onto Fock space, determining its coefficients. In general, we may consider the process of generating the full wave function from the reference state to be an operator $\hat{\Omega}$ acting on the reference state

$$|\Psi\rangle = \hat{\Omega}|\Phi_0\rangle. \tag{5.2}$$

The explicit form of $\hat{\Omega}$ is to be determined by the various many-body methods.

5.2 The Hartree-Fock Method

A complete basis often means an infinite number of single-particle states, which in turn means an infinite number of Slater determinants. Such a system is not possible to implement computationally, so we may instead seek ways of truncating our basis. As previously discussed in chapters 1 and 2, the precision of our solution will now depend on how much of the exact wave function is present in the retained basis. To this end, we will seek a single-particle basis that ensures that most, or as much as possible, of the systems wave function is present in one single Slater determinant.

The *Hartree-Fock method* is a way of optimizing the single-particle states so that a single Slater determinant gives a good representation of the system. The resulting Slater determinant may serve as a starting point for several so called *post Hartree-Fock methods*, such as MBPT (many body perturbation theory), CI (Configuration Interaction) or CC (Coupled Cluster) method [7].

The first development of the method is attributed to Douglas Rayner Hartree who in 1927 introduced a procedure he called *the self consistent field method* [8] but which was later named the *Hartree-Fock-Rothaan* method. Some of the original differences between various Hartree-Fock approaches are nowadays seldomly mentioned or have simply been forgotten [7], and the more generic name Hartree-Fock normally refers to the method that will be discussed in the following sections.

5.2.1 The variational principle

The Hartree-Fock method is based on the variational principle, which states that when we evaluate the inner product of the Hamiltonian on *any* normalized wave function, the resulting energy will be an upper bound to the ground state energy [14], that is

$$E_0 \leq \langle \Psi_{trial} | \hat{H} | \Psi_{trial} \rangle . \tag{5.3}$$

Such wave functions are commonly called *trial* wave functions.

The variational principle provides us with an approximate scheme, and by parameterizing the trial wave function we may improve upon this solution.

5.2.2 Expanding the single-particle states

The parametrization of the Slater determinant may be performed by letting each single-particle state be represented by a linear combination of the eigenstates to the corresponding one-body problem. If we refer to the linear combinations by ψ with Latin indices and the basis states as ϕ with Greek indices, we may write this as

$$|\psi_i\rangle = \sum_{\alpha}^{A} c_{\alpha,i}|\phi_\alpha\rangle. \tag{5.4}$$

Note that we truncate the basis at the A'th function to make computations possible. More functions generally means a better representation. In principle, our chosen basis set spans an infinity of states, but due to obvious computational limits we have to truncate the above sum.

In accordance with the variational principle, we now want to determine the coefficients $c_{\alpha,i}$ in such a way that we minimize the energy. We should therefore insert the expanded single-particle states into the energy equation (2.32) that we derived in chapter 1. We will assume that the system's Hamiltonian has the form

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{i=1}^{N} \hat{h}_0(\mathbf{x}_i) + \sum_{i<j=1}^{N} \hat{v}(\mathbf{x}_i, \mathbf{x}_j), \tag{5.5}$$

where N is the number of particles in our system, $\hat{h}_0(\mathbf{x}_i)$ is the one-body operator acting on particle i, and $\hat{H}_I$ is the interaction. For simplicity, we will avoid defining these in more detail, and we will use the following shorthand notation for the two-body integrals

$$\langle\alpha\beta|V|\alpha\beta\rangle \equiv \int \psi_\alpha^*(\mathbf{r}_i)\psi_\beta^*(\mathbf{r}_j)V(r_{ij})\psi_\alpha(\mathbf{r}_i)\psi_\beta(\mathbf{r}_j)d\mathbf{r}_i\mathbf{r}_j, \tag{5.6}$$

and

$$\langle\alpha\beta|V\beta\alpha\rangle \equiv \int \psi_\alpha^*(\mathbf{r}_i)\psi_\beta^*(\mathbf{r}_j)V(r_{ij})\psi_\alpha(\mathbf{r}_j)\psi_\beta(\mathbf{r}_i)d\mathbf{r}_i\mathbf{r}_j, \tag{5.7}$$

where the quantity $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ indicates that the interaction depends upon the distance between the particles. We may even express this as a so-called antisymmetric matrix element:

$$\langle\alpha\beta|V|\alpha\beta\rangle_{AS} \equiv \langle\alpha\beta|V|\alpha\beta\rangle - \langle\alpha\beta|V|\alpha\beta\rangle. \tag{5.8}$$

By inserting the expanded states in Eq. (2.32), we will find the expecation value of the Hamiltonian in terms of one single Slater determinant expressed as a function of the set of coefficients $\{c\}$

$$\epsilon(\{c\}) = \sum_{i}^{N}\sum_{\alpha,\beta}^{A} c_{\alpha,i}^* c_{\beta,i} \sum_{i}^{N}\langle\phi_\alpha|\hat{h}_0|\phi_\beta\rangle + \frac{1}{2}\sum_{ij}^{N}\sum_{\alpha,\beta,\gamma,\delta}^{A} c_{\alpha,i}^* c_{\beta,j}^* c_{\gamma,i} c_{\delta,j} \langle\phi_\alpha\phi_\beta|\hat{v}|\phi_\gamma\phi_\delta\rangle_{AS}. \tag{5.9}$$

The way in which we now have set up the system, we may consider the coefficients to be a matrix with A rows (Greek letters) and N columns, where N is the number of particles. In the special case where diagonal elements are one and all others zero, we find the same system as we have discussed up to this point.

5.2.3 Ensuring orthonormality

The trial wave function is now parameterized by the introduction of coefficients. When varying or optimizing these coefficients to minimize the energy, we must be careful to constrain the orthonormality of the states (remember, failing to do so will cause a violation of the Pauli principle). Mathematically this constraint may be written

$$\langle \psi_i | \psi_j \rangle = \sum_{\alpha}^{A} c_{\alpha,i}^{*} c_{\alpha,j} = \delta_{ij}. \tag{5.10}$$

To this end, we may use *Lagrange's method of undetermined multipliers* [11, p.116] (or simply *Lagrange multipliers*), and set up a functional to be minimized with the constraint described above, namely

$$F\big(\epsilon\big(\{c\}\big), \lambda\big) = \epsilon\big(\{c\}\big) - \sum_{i}^{N} \lambda_i \sum_{\alpha}^{A} c_{\alpha,i}^{*} c_{\alpha,i}. \tag{5.11}$$

To find a minimum for this functional we must solve

$$\frac{\partial}{\partial c_{\alpha,i}^{*}} F\big(\epsilon\big(\{c\}\big), \lambda\big) = 0. \tag{5.12}$$

For a step by step solution, the reader is referred to Thijssen's *Computational Physics* [8] or Szabo's *"Modern Quantum Chemistry"* [11]. In the following we will just state the solution consistent with how it is given in these sources.

5.2.4 The Hartree-Fock Equations

By solving Eq. (5.12), we obtain a set of coupled one particle eigenvalue problems given by

$$\lambda_k c_{\alpha,k} \quad = \quad \sum_{\gamma}^{A} \sum_{i}^{N} c_{\gamma,k} \langle \alpha | \hat{h}_0 | \gamma \rangle \quad + \quad \sum_{\beta,\gamma,\delta}^{A} \sum_{i}^{N} c_{\beta,i}^{*} c_{\gamma,i} c_{\delta,k} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle_{AS}. \tag{5.13}$$

These are known as the Hartree-Fock equations, and by identifying λ as energy associated with single-particle state k [16] we may write them as

$$\epsilon_k^{HF} c_{\alpha,k} = \sum_{\gamma} \Big(\langle \alpha | \hat{h}_0 | \gamma \rangle + \sum_{j}^{N} \sum_{\beta\delta} c_{j\beta} c_{j\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle_{AS} \Big) c_{\gamma,k}, \tag{5.14}$$

or simply

$$\sum_{\gamma} f_{\alpha\gamma}^{HF} c_{\gamma,k} = \epsilon_k^{HF} c_{k,\alpha}, \tag{5.15}$$

where we have defined

$$f_{\alpha\gamma}^{HF} \equiv \langle \alpha | \hat{h}_0 | \gamma \rangle + \sum_{j}^{N} \sum_{\beta\delta} c_{j\beta} c_{j\delta} \langle \alpha\beta | V | \gamma\delta \rangle_{AS} \tag{5.16}$$

The matrix $\hat{f}^{HF}$ is commonly known as the *Fock* matrix, and has dependence upon all the spin orbitals [7]. As a consequence, a first guess for the coefficients is likely to be inconsistent when evaluating the left hand side and right hand side of Eq. (5.15), so they are normally solved in an iterative manner. This is why the method was initially named the self-consistent field method; the field produced by the particles should be consistent with the field "felt" by each particle.

A set of single particle states that fulfills Eq. (5.15) is called the *canonical Hartree-Fock wave function*, while the constituent states are called *the canonical spin orbitals*.

5.2.5 Koopman's theorem

The Hartree Fock energy for any single-particle state p may be compactly written

$$\epsilon_p^{HF} = \langle p | \hat{h}_0 | p \rangle + \frac{1}{2} \sum_j \langle pj || pj \rangle. \tag{5.17}$$

This is the energy associated with each orbital expanded as a linear combination of single-particle states. Each fermion in the system will then have an associated Hartree-Fock energy.

Koopman's theorem states that for closed-shell Hartree-Fock calculations, the ionization energy of the system is equal to the negative of the outermost hole state in the reference state. [8] The ionization energy may then be computed by performing a Hartree-Fock procedure, whereby we calculate only the outermost occupied orbital in equation Eq. (5.17).

5.2.6 Restricted and unrestricted Hartree-Fock

There are multiple ways of implementing the Hartree-Fock equations. For closed shell systems, such as He, H_2 and Be, the electrons (we consider only electronic degrees of freedom in this thesis), may be assumed to be paired with opposite spin electrons. In the restricted Hartree-Fock method (RHF), we consider only systems where all fermions (electrons in our case) are paired with opposite spin particles. This lets us scale down the computation, but it will naturally give poor results for systems where no such pairing occurs, for example in the case where two H atoms are interacting over large distances.

For systems with singly occupied states (unpaired fermions, typically odd numbered systems) the restriction of spin-pairing is no longer valid.

To take this into account, we set up two Fock matrices; one for each spin orientation. The resulting system is

$$\hat{F}_\alpha(C_\alpha, C_\beta)C_\alpha = \hat{S}C_\alpha\epsilon_\alpha, \tag{5.18}$$

and

$$\hat{F}_\beta(C_\alpha, C_\beta)C_\beta = \hat{S}C_\beta\epsilon_\beta. \tag{5.19}$$

The subindices α and β represent the distinction between the two possible spin-orientations in our system, for example α for spin up and β for spin down. For each orientation we have a separate coefficient matrix C.

The ground state energy will be a function of the eigenvectors of these two matrices. The dependency of opposite spin coefficients in the Fock matrices is due to the direct- or coulomb term from the two particle integrals [11, p.241]. This latter approach is called the *unrestricted Hartree-Fock* (UHF) method.

5.3 Post Hartree-Fock Methods

While the Hartree-Fock reference state may account for important correlations such as the Pauli exclusion principle and interactions with the mean field, it will lack the more complicated correlations. For this reason, certain correlations may never be fully accounted for with the Hartree-Fock method alone. Figure 5.1 illustrates what is often defined as the correlation energy; the energy attributed to correlations beyond the mean field.

A Hartree-Fock calculation may also provide us with energy minimized orbitals above the Fermi level that may be used to construct excited Slater determinants in so-called *post Hatree-Fock* methods.

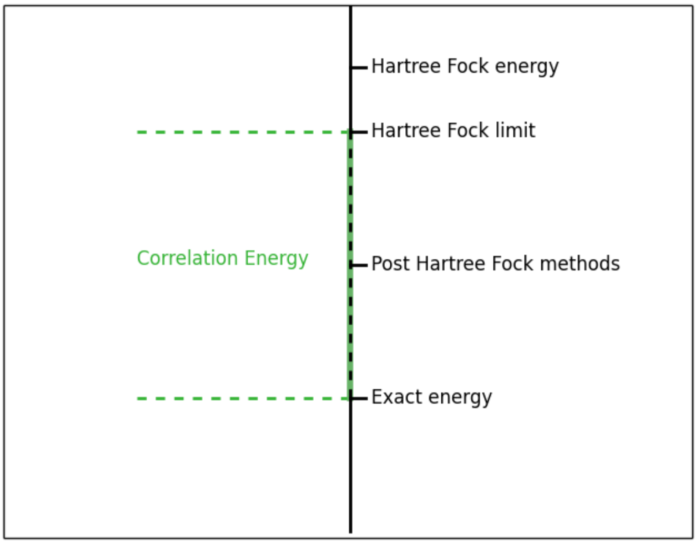


FIGURE 5.1: Correlations beyond the Hartree-Fock energy. At the so called Hartree-Fock limit, we have achieved the best possible description of our system using the mean field approach. The energy unaccounted for by the reference state is commonly referred to as the *correlation energy.* The so called *post Hartree-Fock* methods, such as the Coupled Cluster method, will help us include even more correlations and bring us closer to the exact energy.

By including excited Slater determinants, these methods may account for the correlations beyond the mean field. One such approach is the Coupled Cluster (CC) method, which is a central subject in this thesis. Alongside CC, we have several other methods such as Configuration Interaction (CI) or Many Body Perturbation Theory (MBPT).

Important insight into various physical systems may be gained by comparing results amongst these methods, as they all have their strengths and weaknesses. Prior to the treatment of the CC method in the next two chapters, we shall therefore just briefly discuss some of these alternative post Hartree-Fock methods, allowing us to make a meaningful comparison of results in the final chapter of this thesis.

5.3.1 Configuration Interaction theory

The Configuration Interaction (CI) method, sometimes referred to as the *method of superposition of configurations* [12] is based on the expansion of the system's wave function into a linear combination consisting of the reference Slater determinant and a (possibly infinite) set of excited versions of this Slater determinant, as briefly noted in the introduction to this chapter,

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ai} c_i^a \ vert\Phi_i^a\rangle + \sum_{abij} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + ... + \sum_{ab...N_a ij...N_i} c_{ij...N_i}^{ab...N_a}|\Phi_{ij...N_i}^{ab...N_a}\rangle. \quad (5.20)$$

The CI method is considered to be the mathematically simplest technique for inclusion of correlations beyond the mean field [12]. The operator that brings us from the reference Slater determinant to the true system wave function may be written

$$|\Psi\rangle = \Omega|\Phi_0\rangle = (c_0 + \hat{C})|\Phi_0\rangle, \quad (5.21)$$

where

$$\hat{C} = \sum_{ai} c_i^a \hat{a}_a^\dagger \hat{a}_i + \sum_{abij} c_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i + ... + \sum_{ab...N_a ij...N_i} c_{ij...N_i}^{ab...N_a} \hat{a}_a^\dagger \hat{a}_b^\dagger ... \hat{a}_{N_a}^\dagger \hat{a}_{N_i} ... \hat{a}_j \hat{a}_i. \quad (5.22)$$

The coefficients c are then obtained by diagonalization, which corresponds to minimizing

$$E = \langle\Psi|\hat{H}|\Psi\rangle. \quad (5.23)$$

We shall not go into more details concerning this. The reader is referred to for example [12, p.177] for a more complete treatment.

We will however make some general observations concerning the CI method.

5.3.2 Full Configuration Interaction theory

Equation (5.23) is solvable, but at great computational cost. The number of excited Slater determinants, and thereby the number of coefficients, scales *factorially* with the number of particle states and hole states.

The practical consequence is that the inclusion of all Slater determinants is only possible for smaller systems unless some truncation in the basis set is introduced. From a physical point of view, we may have systems where no single excitation is allowed, and for this reason the singly excited coefficients may be excluded with no loss of precision.

The various truncations are commonly referred to as CIS for inclusion of single excitations, CISD for single and double excitations, CISDT for single, double and triple excitations and so forth.

5.3.3 Configuration Interaction Quantum Monte Carlo

We will later in this thesis compare our results with those from a so-called FCIQMC (Full Configuration Interaction Quantum Monte Carlo) calculation. This approach determines the FCI coefficients by stochastic processes [19, 20]. Such results are very valuable for our purpose since they basically provide us with the exact ground state energy for smaller systems, which will allow us to evaluate to which degree the method accounts for correlations in the system.

5.3.4 Many-body Perturbation Theory

As opposed to the Hartree-Fock, Configuration Interaction and Coupled Cluster methods, *Many-body Perturbation Theory* (MBPT) offers a non-iterative approach to approximating the systems wave function. The following derivation is based on the one in Refs. [7] and [16].

The basic idea is to arrange the Hamiltonian into two parts

$$\hat{H} = \hat{H}_0 + \hat{V}. \tag{5.24}$$

This is very similar to what we have done so far, as we seek an operator $\hat{H}_0$ which is exactly solvable (which yields the single-particle basis) and an operator $\hat{V}$ that will be treated as a perturbation. The solution to the unperturbed problem is given

$$\hat{H}_0|\Phi_0\rangle = W_0|\Phi_0\rangle. \tag{5.25}$$

As for Configuration Interaction theory, we have the exact ground state wave function for our system represented by a linear combination of Slater determinants, where we assume the first term (reference state) to be the dominating term

$$|\Psi_0\rangle = |\Phi_0\rangle + \sum_{m}^{\infty} c_m|\Phi_m\rangle. \tag{5.26}$$

We then assume intermediate normalization $\langle\Phi_0|\Psi_0\rangle = 1$ and project our Schödinger equation onto $\langle\Phi_0|$:

$$\langle\Phi_0|\hat{H}|\Psi_0\rangle = \langle\Phi_0|\hat{H}_0 + \hat{V}|\Psi_0\rangle = E. \tag{5.27}$$

The true ground state energy is still unknown, but we may subtract the *unperturbed* energy to find an expression for the correlation energy

$$\langle\Phi_0|\hat{V}|\Psi_0\rangle = E - W_0 = \Delta E. \tag{5.28}$$

We may add and subtract a term $\omega|\Psi_0\rangle$ to the expression above and regroup it to find

$$|\Psi_0\rangle = \frac{1}{(\omega - \hat{H}_0)}(\hat{V} + \omega - E)|\Psi_0\rangle. \tag{5.29}$$

By interpreting the term ω in different ways, we will arrive at the various many-body perturbation schemes.

The system's true wave function is still unknown, but it is fully possible to expand it in a known basis $\{\phi_n\}$, so that

$$|\Psi\rangle = (\hat{P} + \hat{Q})|\phi_0\rangle, \tag{5.30}$$

where we have the projection operator $\hat{P} = |\phi_0\rangle\langle\phi_0|$, $\hat{P} = \hat{P}^\dagger = \hat{P}\hat{P}$ and $\hat{Q} = \sum_m |\phi_m\rangle\langle\phi_m|$ [7]. We insert this into Eq. (5.29) to find

$$|\Psi_0\rangle = \phi_0\rangle + \sum_i^\infty \Big(\frac{1}{(\omega - \hat{H}_0)}(\hat{V} + \omega - E)\Big)^i |\phi_0\rangle, \tag{5.31}$$

and the correlation energy takes the form

$$\Delta e = \sum_i^\infty \hat{V}\Big(\frac{1}{(\omega - \hat{H}_0)}(\hat{V} + \omega - E)\Big)^i |\phi_0\rangle. \tag{5.32}$$

By letting $\omega = E$, we obtain the so called Brillouin-Wigner Perturbation Theory, or we may let $\omega = W_0$ to obtain Rayleigh-Schrödinger Perturbation Theory (RSPT). In the Brillouin-Wigner case, a possible solution would involve iterations and self consistence as in the Hartree-Fock-, Configuration Interaction- and Coupled Cluster cases, while in the RSPT case we end up with terms that correspond to different orders of the correlation energy. For RSPT, we obtain

$$\Delta e = \Delta E^{(1)} + \Delta E^{(2)} + \Delta E^{(3)} + ..., \tag{5.33}$$

where

$$\Delta E^1 = \langle\phi_0|\hat{V}|\phi_0\rangle, \tag{5.34}$$

$$\Delta E^2 = \langle\phi_0|\hat{V}\frac{\hat{Q}}{(W_0 - \hat{H}_0)}(\hat{V} - \Delta E)|\phi_0\rangle, \tag{5.35}$$

$$\Delta E^3 = \langle\phi_0|\hat{V}\frac{\hat{Q}}{(W_0 - \hat{H}_0)}(\hat{V} - \Delta E)\frac{\hat{Q}}{(W_0 - \hat{H}_0)}(\hat{V} - \Delta E)|\phi_0\rangle, \tag{5.36}$$

represent the correlation energy to first, second and third order, respectively. Higher orders are obtained along similar lines. Every order in RSPT introduces different types of correlations. With a two-body force only, to second order we can obtain contributions from at most two-particle-two-hole excitations. At for example fourth order, we can also obtain contributions which represent four-particle-four-hole excitations. Whereas methods like Configuration Interaction or Coupled Cluster include such correlations to infinite order in the interaction, RSPT contains such correlations only up to the given order in the expansion.

5.3.5 The linked diagram theorem

We will not go into any further detail on the many-body perturbation methods, but we should note an important theorem introduced by Goldstone [21], see also Ref. [7, p.152].

Computations of the different orders in Rayleigh-Schrödinger Perturbation Theory may be performed diagrammatically. Based on a set of rules (see for example [7]), we may generate diagrams corresponding to the possible contractions operators present in each order of the perturbation. Actually, the diagrammatic rules will produce a lot more diagrams then what is actually needed in order to calculate the correlation energy.

The *linked diagram theorem* is a simple way of getting rid of a lot of these excess diagrams.

A diagram may be called *linked* if all parts of the diagram is linked with each other by contractions. Unlinked diagrams will be easily identified as it is possible to split them into smaller parts by drawing lines through them without crossing any lines in the diagram.

The linked diagram theorem states that these diagrams do not contribute to the correlation energy.

In Eq. (5.32) for Rayleigh-Schrödinger Perturbation Theory, we will therefore only have to consider terms where contractions occur for diagrams where every vertex is linked to another vertex by at least one particle or hole line.

5.4 Other many-body methods

The methods we have discussed so far are relevant to this thesis, but the full range of methods for dealing with many body problems extends even further. Some notable methods apart from the ones discussed so far is the wide range of *Quantum Monte Carlo* (QMC) methods, that build upon the same formalism as us, but use instead stochastic methods to approximate the systems wavefunction (see for example [8, Chapter 12]). Other methods such as *Density Functional Theory* (DFT) may even depart from the Hartree-Fock formalism to offer more suitable equations for solids, while at the same time represent alternatives to Hartree-Fock calculations for atoms and molecules [8, Chapter 5].

Chapter 6

The Coupled Cluster Method

6.1 Historical Account

As is the case with many contributions to science, it is hard to pinpoint the origin of the Coupled Cluster methods to one single scientific work. It is however commonly accepted that the groundbreaking work was made by the nuclear physicists Fritz Coester and Hermann Kümmel in the mid and late 50s. At this time the computational aspects of many body theory was still in its early infancy, although some variational Hartree-Fock calculations had been performed in the quantum chemistry community. [10]

One important discovery that may have motivated work on the Coupled Cluster method was made by J. Hubbard, who laboriously inspected the time independent perturbation series to all orders and found that only linked terms contribute to the energy, as well as that the energy associated with these terms was *extensive* [22].

Coester was then shortly after the work of Hubbard able to derive basically the same results using the *exponential ansatz* and *the Hausdorff expansion* [10]. In the following years, Kümmel and Coester published a number of papers describing the method [23, 24], and by the late 1950s the method was well understood [10].

Together with Haag, Coester was also able to show that the exponential ansatz was a natural choice for the system's wave function [25], meaning that the choice was not as arbitrary as one may get the impression of when reading modern books on the subject [7].

Despite this, it was not until 1966 that the first real applications of the method were made, and these were made by the quantum chemist Jiri Cizek [26]. Coupled Cluster calculations for nuclear matter were performed in the late 1970s and early 1980s [27, 28].

Today, the so called Hartree-Fock + CCSD(T) (Coupled Cluster Singles Doubles and Perturbative Triples) calculation is considered the "gold standard" of quantum chemistry, due to its relatively low computational cost compared to its ability to account for important correlations in many systems.

6.2 The exponential ansatz

The derivation of the coupled cluster method begins with assuming that the operator that brings the reference state into the true system state can be represented by the exponetial operator

$$e^{\hat{T}}, \tag{6.1}$$

where the *cluster operator* is defined as

$$\hat{T} \equiv \hat{T}_1 + \hat{T}_2 + ... = 1 + \sum_{ai} t_i^a a_a^\dagger a_i + \sum_{ai} t_{ij}^{ab} a_a^\dagger a_b^\dagger a_i a_j + ... \tag{6.2}$$

As may be seen from the second quantized form of the cluster operator, its terms will cause an excitation of the reference state. For example, the excitation operator $\hat{T}_1$ generates a singly excited reference state

$$\hat{T}_1|\Phi_0\rangle = \sum_{ai} t_i^a a_a^\dagger a_i |\Phi_0\rangle = \sum_{ai} t_i^a |\Phi_i^a\rangle, \tag{6.3}$$

while the $\hat{T}_2$ operator creates a doubly excited reference state

$$\hat{T}_2|\Phi_0\rangle = \sum_{abij} t_{ij}^{ab} a_a^\dagger a_b^\dagger a_i a_j |\Phi_0\rangle = \sum_{abij} t_{ij}^{ab} |\Phi_{ij}^{ab}\rangle, \tag{6.4}$$

and so on.

For reasons that will become clear shortly, there is no need to include excitation operators beyond $\hat{T}_4$ for systems containing at most two-body interactions. Truncations in the cluster operator are commonly made to give rise to the different types of coupled cluster methods.

The coefficients $t_{ij...}^{ab..}$ are commonly referred to as *amplitudes*, and they are the quantities whose solution we will seek in an iterative way.

As for any exponential function, we may expand it as

$$e^{\hat{T}} = 1 + \hat{T} + \frac{1}{2!}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + ..., \tag{6.5}$$

meaning that the exponential ansatz may be written

$$e^{\hat{T}}|\Psi_0\rangle = (1 + \hat{T} + \frac{1}{2!}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + ...)|\Psi_0\rangle. \tag{6.6}$$

6.3 Size consistency

The concept of size consistency was introduced by Pople *et al.* in 1978 [29] to describe methods that properly represent systems in the noninteracting limit. A system of electrons interacting through the Coulomb force has a noninteracting limit when the distance between the electrons is so large that the energy of this configuration should correspond to the energy of an equal number of noninteracting electrons.

In other words, a size consistent method should for the interacting systems A and B produce energy calculations in the noninteracting limit where [7, p.12]

$$E(AB) = E(A) + E(B) \tag{6.7}$$

Because of the fundamental assumption of pairing of electrons in the restricted Hartree-Fock (RHF) method, this method is not size consistent for systems such as the H_2 molecule. This shortcoming may however be mended by performing a coupled cluster calculation on top of the RHF, producing size consistent energies when gradually separating the hydrogen nucleis (and their electrons). This is illustrated by a comparison of RHF, unrestricted Hartree-Fock (UHF) and RHF+CCSD (coupled cluster theory at the level of singles and doubles excitations only) for the H_2 molecule in Fig. 6.1. It is clear that since the RHF method forces the electrons of each hydrogen atom to pair, the energy is wrongly estimated compared to the exact energy as we separate the hydrogen nuclei.

By performing a CCSD calculation employing the RHF basis, this restriction is lifted, allowing the electrons to fall into their natural orbits. An unrestricted Hartree-Fock calculation on the other hand does not assume the electrons to be paired, allowing thereby for a natural behavior as the separation in distance between the two hydrogen nuclei increases.

When comparing CC with CI, we will find that the $\hat{C}_2$ operator has a corresponding combination of cluster operators

$$\hat{C}_2 = \hat{T}_2 + \frac{1}{2}(\hat{T}_1)^2. \tag{6.8}$$

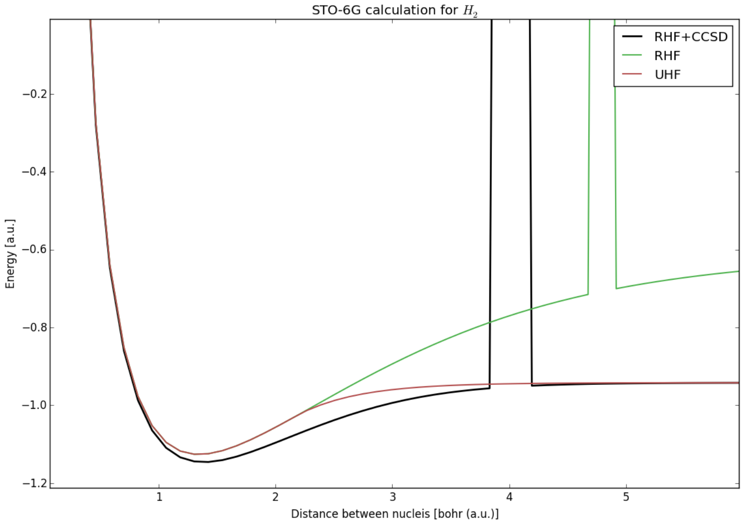


FIGURE 6.1: Comparison of UHF, RHF and RHF+CCSD for a H_2 molecule. These results were produced by the author using a self-developed solver [3] that utilizes gaussian basis sets to enable RHF, UHF, CCD and CCSD calculation on atoms and molecules. These results illustrate the size consistency of the CCSD equations, and the lack of size consistency in the RHF case. The vertical jumps in the curves are regions where the solver failed to converge due to improperly chosen relaxation parameters. Central parts of the code [3] that was used to produce these calculations was developed by the author as part of another course.

The last quadratic term above is commonly referred to as a *disconnected cluster*, and such terms are responsible for ensuring the consistency of the CC method.

6.4 Extensivity

A related concept to consistency is *size extensivity*, as introduced by Bartlett [30]. A quantum mechanical model may be called *extensive* if the energy of this model scales correctly with the size of the system [7, p.11]. This is analogous to extensive properties in statistical mechancial systems, that scales linearly with the size of the system, see for example Ref. [31, p.9].

When performing calculations on periodic systems such as gases or solids, extensivity is a feature that allows us to extrapolate results beyond the limits of the simulation cell.

For the coupled cluster method, extensivity is an inherent property of the exponential ansatz (see Eq. (6.1)). This may be shown by considering that the reference state is separable, implying that

$$\phi_0(A,B) = \phi_0(A)\phi_0(B). \tag{6.9}$$

Furthermore, the cluster operators are additive and we have

$$\hat{T}(A,B) = \hat{T}(A) + \hat{T}(B). \tag{6.10}$$

It then follows that the total wave function is multiplicatively separable

$$\Psi(A,B) = e^{T(A)+T(B)}\phi_0(A,B) = e^{T(A)}\phi_0(A)e^{T(B)}\phi_0(B). \tag{6.11}$$

Thus, the energy is additive:

$$\hat{H}(A,B)\Psi(A,B) = [E(A) + E(B)]\Psi(A,B). \tag{6.12}$$

6.5 Deriving the coupled cluster equations

We will derive the coupled cluster equations in four steps. First we shall do some more work on the exponential ansatz to derive a form that is more easily translated to actual calculations, secondly we will introduce diagrammatic rules that will simplify derivations considerably, and then we shall briefly comment upon how to translate these rules into a code that makes the process of deriving coupled cluster equations (and code) of any order remarkably simple.

Since the actual equations that are worked into a high-performance code will depend upon how we truncate the ansatz, we will then finally derive the full equations using the above mentioned code in increasing order of complexity.

6.6 Unwrapping the exponential ansatz

6.6.1 The CC effective Hamiltonian

The actual quantity we seek is the correlation energy Δe as found by solving the Schrödinger equation for the exponential ansatz and the normal ordered Hamiltonian Eq. (3.49)

$$\hat{H}_N e^{\hat{T}}|\Phi_0\rangle = \Delta e e^{\hat{T}}|\Phi_0\rangle. \tag{6.13}$$

We may reorganize this into

$$(\hat{H}_N - \Delta e)e^{\hat{T}}|\Phi_0\rangle = 0. \tag{6.14}$$

By multiplying with the inverse exponential, we find

$$(e^{-\hat{T}}\hat{H}_N e^{\hat{T}} - \Delta e)|\Psi_0\rangle = 0, \tag{6.15}$$

where the non-Hermitean operator

$$e^{-\hat{T}}\hat{H}_N e^{\hat{T}} \equiv \mathcal{H}, \tag{6.16}$$

is a similarity transformed Hamiltonian, often called the *CC effective Hamiltonian.*

By projecting Eq. (6.15) onto the reference state, we may then solve for the correlation energy

$$\langle\Phi_0|e^{-\hat{T}}\hat{H}_N e^{\hat{T}}|\Psi_0\rangle = \Delta e. \tag{6.17}$$

It also follows that

$$\langle\Phi^*|e^{-\hat{T}}\hat{H}_N e^{\hat{T}}|\Psi_0\rangle = 0, \tag{6.18}$$

where Φ^* represents any excited state. This last equation will aid us in solving the amplitudes.

6.6.2 Non-variational coupled cluster theory

At this point we should note that the CC effective Hamiltonian is no longer Hermitean, since

$$\mathcal{H}^\dagger = (e^{-\hat{T}}\hat{H}_N e^{\hat{T}})^\dagger = (e^{-\hat{T}})^\dagger\hat{H}_N(e^{\hat{T}})^\dagger = (e^{-\hat{T}^\dagger})\hat{H}_N(e^{\hat{T}^\dagger}) \neq \mathcal{H} \tag{6.19}$$

A consequence is that truncations in the cluster operator will cause the energy expression in Eq. (6.18) to no longer be variational, and thus the energy does no longer represent an upper bound to the ground state energy [32]. In practice, however, solving the resulting equations obtained from the projection technique in Eq. (6.18) will still provide us with an energy close to the real expectation value even for truncated cluster operators [32].

6.6.3 The Hausdorff Expansion

The Hausforff expansion (or Baker-Campbell-Hausdorff formula) provides us with the means of rewriting the CC effective Hamiltonian in terms of nested commutators [7, p.293]:

$$\mathcal{H} = \hat{H}_N + [\hat{H}_N, \hat{T}] + \frac{1}{2}[[\hat{H}_N, \hat{T}], \hat{T}] + \frac{1}{3!}[[[\hat{H}_N, \hat{T}], \hat{T}], \hat{T}] + \frac{1}{4!}[[[[\hat{H}_N, \hat{T}], \hat{T}], \hat{T}], \hat{T}]. \tag{6.20}$$

The reason that this sum of nested operators truncates at the four-fold commutator, will become apparent if we use Wicks generalized theorem to evaluate the expansion. For two strings of evenly numbered creation- and annihilation operators, we will find the commutator

$$[A, B] = AB - BA = np[A, B] + \overbracket{AB} - np[B, A] - \overbracket{BA}, \tag{6.21}$$

where the contraction represents the sum of all normal ordered products with one or more contractions present.

This is simplified further since both A and B (in our case H and T) contain an even number of creation and annihilation operators

$$np[A, B] = np[B, A], \tag{6.22}$$

meaning that the uncontracted normal ordered products cancel

$$[A, B] = \overbracket{AB} - \overbracket{BA}. \tag{6.23}$$

Since the cluster operators T_m commute, the nested operators will only have surviving terms where $\hat{H}_N$ contracts with one or more cluster operators.

The reason for the truncation of the Hausdorff expansion is then apparent; $\hat{H}_N$ is a sum of operators which at most contains four creation or annihilation operators. It is therefore not possible to contract any term in $\hat{H}_N$ to more than 4 cluster operators as long as the Hamiltonian contains at most two-body interactions.

6.6.4 Rewriting the Hausdorff expansion

We also have to take into account that particle creation operators can only produce a nonzero contraction with a particle annihilation operator to its left, and a hole annihilation operator may only produce a nonzero contraction with a hole creation operator to its left, meaning that the only contributing terms when computing the nested operator are terms that begin with the normal ordered Hamiltonian

$$\mathcal{H} = e^{-\hat{T}} \hat{H}_N e^{\hat{T}} = \hat{H}_N + \overbracket{\hat{H}_N \hat{T}} + \frac{1}{2} \overbracket{\hat{H}_N \hat{T} \hat{T}} + \frac{1}{3!} \overbracket{\hat{H}_N \hat{T} \hat{T} \hat{T}} + \frac{1}{4!} \overbracket{\hat{H}_N \hat{T} \hat{T} \hat{T} \hat{T}} \equiv (\hat{H}_N e^{\hat{T}})_C \tag{6.24}$$

The contractions indicate a sum over all terms in which the Hamiltonian connects by at least one contraction to each of the cluster operators to its right. The subscript C indicates connected terms only. For further details, see Ref. [7, p.294].

6.6.5 The CC equations

We are finally in a position to derive explicit expressions for the energy and amplitudes. We have the general equation

$$(\hat{H}_N e^{\hat{T}})_C|\Phi_0\rangle = \Delta e|\Phi_0\rangle. \tag{6.25}$$

Finding the energy is now simple, we have

$$\langle\Phi_0|(\hat{H}_N e^{\hat{T}})_C|\Phi_0\rangle = \Delta e. \tag{6.26}$$

The amplitudes may be found from the equations

$$\langle\Phi_i^a|(\hat{H}_N e^{\hat{T}})_C|\Phi_0\rangle = 0, \tag{6.27}$$

and

$$\langle\Phi_{ij}^{ab}|(\hat{H}_N e^{\hat{T}})_C|\Phi_0\rangle = 0, \tag{6.28}$$

and so on, depending on how we truncate the cluster operator. We will refer to the first equation (6.26) as the *energy equation*, while the following ones will be referred to as the *amplitude equations*.

6.6.6 Truncating the ansatz

The connection requirements of the CC effective Hamiltonian in Eq. (6.24) will also impact on truncations in the cluster operator. Since our Hamiltonian at most contains four pairs of creation- and annihilation operators (that is we at most a two-body interaction), it will not be able to fully contract with cluster operators beyond the four-fold excitation.

The explicit form of the equations will depend on how we truncate the cluster operator, and this is what defines the different types of coupled cluster methods.

While it may seem reasonable to have the first truncation only including $\hat{T}_1$, we know from Thouless' theorem [7, p.257]that this approach would only transform a single determinant into another single determinant, so this truncation does not occur. This is in analogy to Brillouin's theorem in configuration interaction theory [7].

The simplest coupled cluster truncation is therefore to include only double excitations in the $\hat{T}_2$ excitation operator, so that $\hat{T} = \hat{T}_2$. Solving the CC equations 6.26 and 6.28 for this truncation is called the Coupled Cluster Doubles (CCD) method.

Table 6.1: Common CC truncations

Truncation	Name
$\hat{T} = \hat{T}_2$	CCD
$\hat{T} = \hat{T}_1 + \hat{T}_2$	CCSD
$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3$	CCSDT
$\hat{T} = \hat{T}_2 + \hat{T}_3$	CCDT
$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \hat{T}_4$	CCSDTQ

To include more correlations, we may also include $\hat{T}_1$, implying that $\hat{T} = \hat{T}_1 + \hat{T}_2$. This truncation defines the Coupled Cluster Singles Doubles (CCSD) method.

Some common truncations are stated in table 6.1.

Even more variations in the CC methods may be achieved by making subselections of contributing terms (or diagrams, or channels) within the various truncations [33].

6.7 Diagrammatic rules

It is of course possible to derive the explicit coupled cluster equations for any truncation from Eqs. (6.26) - (6.28) using Wick's theorem. Such derivations may be found for example in Shavitt and Bartlett's *"Many Body Methods in Chemistry and Physics"* [7, Chapter 9].

This process is very cumbersome and prone to human errors, especially when higher excitations are included. Consider for example the multitude of terms arising from the CCSDT method with the terms from $\hat{T}^4 = (\hat{T}_1 + \hat{T}_2 + \hat{T}_3)^4$, and their contractions with the normal ordered Hamiltonian.

Instead of using Wick's theorem to derive the explicit expressions for this term, we may manipulate diagrammatic equivalents to the operators, subject to rules that ensure consistency with the algebraic approach. This way, the derivations of the terms in question become more easy to perform by hand.

The basic framework for this was introduced in Chapter 4, and we shall now build upon this to present rules that allow us derive the coupled cluster equations. The rules we follow for diagram generation are identical to those found in the work of Shavitt and Bartlett [7, p.297].

6.7.1 The cluster operators

We have previously introduced the various terms of the normal ordered Hamiltonian in Figs. 4.3, 4.4 and 4.5, in chapter 3. The excitation operators are represented by contiguous horizontal lines with an even number of particle- and hole lines above, as shown in Figs. 6.2, 6.3, 6.4 and 6.5.

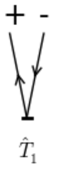


FIGURE 6.2

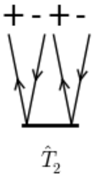


FIGURE 6.3

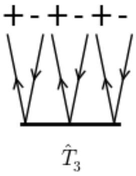


FIGURE 6.4

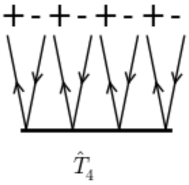


FIGURE 6.5

6.7.2 Contractions of operators

Analogous to Wicks's theorem, we need to find all possible ways to contract the normal ordered Hamiltonian with strings of excitation operators. Our objective is thus to combine excitation operators, possibly strings of excitation operators, with the various terms in the normal ordered Hamiltonian, so they form topologically distinct diagrams.

In Figs. 6.2, 6.3, 6.4 and 6.5, lines exiting the excitation operators are assigned a number of plus and minus signs depending on the number of particle- and hole lines in the operator. In the same way, we may assign plus or minus signs to lines *below* the interaction in the terms in the normal ordered Hamiltonian. For strings of excitation operators, we shall denote separations between the operators by a vertical line, so that for example the product

$$\hat{T}_1\hat{T}_2 \rightarrow + - \,|\, + - + - \tag{6.29}$$

Contractions will be represented diagrammatically as connecting corresponding lines exiting the excitation operator and entering the interaction vertex in the normal ordered Hamiltonian. The purpose of the signs are to derive all topologically distinct connections between these operators.

The plus and minus signs representing lines in the Hamiltonian term must then *all* connect to a corresponding sign in the string of excitation operators, so that no lines entering or exiting below the interaction is unconnected. At the same time, we seek only combinations where at least one connection is made between the Hamiltonian term and each of the excitation operators.

The practicalities of this procedure may be outlined in three steps. First we must create any sequence containing the plus and minus signs that occur in the Hamiltonian and the vertical bars from the string of excitation operators. No vertical bars in this string of excitation operators means that we have only one operator, and that no such vertical bars should occur in the created sequence.

The next step is to consider all possible permutations of this sequence. For each permutation we should first ensure that the diagram is *linked* (see 5).

Contractions between the interaction and each of the excitation operators will be separated by the vertical bars, so that unconnected operators may be identified by either repeated vertical bars in the sequence as in

$$+ + || - | - \tag{6.30}$$

or by vertical bars at the start or end of the sequence as in

$$+ + | - | - | \tag{6.31}$$

These connection patterns will correspond to unconnected diagrams and thus not contribute to the equations. Next, we need to ensure that the connection pattern is compatible to the operators, in the sense that the number of connections of particle and holes ($+$ and $-$) to each excitation operator must not exceed the corresponding plus and minus signs present in the excitation operator.

As an example, we may consider the Hamiltonian term in Fig. 6.6 connecting to the string of operators in Eq. (6.29). These will together produce the sequence

$$\hat{H}_{N,9}\hat{T}_1\hat{T}_2 \rightarrow + - + - | \tag{6.32}$$

Now, in the order above this is obviously not a contributing term or even a possible connection pattern, since there are no connections between $\hat{H}_N$ and $\hat{T}_2$, and the number of connections to $\hat{T}_1$ exceeds the possible number of connections to this operator.

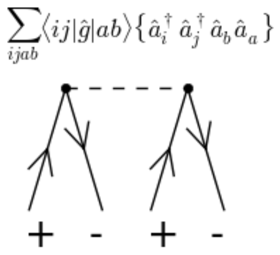

$\sum_{ijab}\langle ij|\hat{g}|ab\rangle\{\hat{a}_i^\dagger\,\hat{a}_j^\dagger\,\hat{a}_b\hat{a}_a\}$
+
-
+
-

FIGURE 6.6: Term in the normal ordered hamiltonian

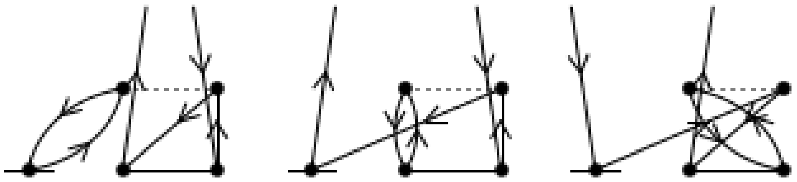


FIGURE 6.7: Diagrams produced by $\hat{H}_{N,9}\hat{T}_1\hat{T}_2$. The diagrams are generated by our code and because of this the lines connect in a somewhat arbitrary fashion. Still, the diagrams are correct and non-ambiguous.

However, considering the possible permutations of this sequence will enable us to identify three distinct connection patterns which are

$$+-|+- \qquad -|++- \qquad +|+-- \tag{6.33}$$

From these three connection patterns we may draw three different diagrams, shown in Fig. 6.7. These diagrams will in turn be translated back into algebraic expressions that we work into our code.

A special case is when multiple occurrences of the same excitation operators occur in the sequence. In this case the operators should be treated as indistinguishable, so that only distinct connections occur.

For example we may consider the case of

$$\hat{H}_{N,9}\hat{T}_2\hat{T}_2 \rightarrow +-+-| \tag{6.34}$$

The possible connection patterns from these operators are

$$+-|+- \qquad -|++- \qquad +|+-- \qquad ++|-- \tag{6.35}$$

The last three terms could just as well be written

$$++-|- \qquad +--|+ \qquad --|++ \tag{6.36}$$

but would not produce a diagram topologically distinct from the connections in Eq. (6.35), and is thereby excluded.

6.7.3 Excitation level

As may be seen from the diagrams in Fig. 6.7, it is fully possible for particles or holes in the excitation operator to remain unconnected. In this case, we have three diagrams where all have unconnected particle and hole lines. Such a diagram will produce an excited Slater determinant, in this case creating a 1p1h (one particle, one hole) state corresponding to the excitation level +1.

In other cases, we may have diagrams with no unconnected lines, corresponding to excitation level 0.

We will also have cases were particle and/or hole lines above the interaction cause excitations.

In general, we will find either zero or an even number of both particle- and hole lines unconnected in the diagram, and dividing the number of these by two will yield the excitation level.

The significance of the excitation level is that it quickly allows us to evaluate in which of the coupled cluster equations a diagram will contribute. Because of the orthonormality of the Slater determinants, only diagrams of excitation level 0 will contribute to the energy. Only diagrams of excitation level 2 will contribute to the t2-amplitude equation, and so on.

6.7.4 Interpretation rules for diagrams

Once all possible diagrams for a given sequence of operators are drawn, we will interpret them back into algebraic expressions that can be worked into code. The rules for interpreting diagrams are here just stated, pretty much as they appear in Shavitt and Bartlett's *"Many Body Methods in Chemistry and Physics"* [7, Chapter 9].

6.7.5 Label all lines

Internal and external particle and hole lines are assigned labels. For consistency and readability one should use the conventional naming with letters *"abcd..."* reserved for particles, and *"ijkl..."* for holes. Since the diagrams represents sum over (internal) lines, it is preferable to reserve certain labels for summation indices and other for "static" indices. This will be helpful when writing the actual code.

6.7.6 Identify the one-body operator

Every one-body vertex should be interpreted as the one-body operator of the states exciting and entering the vertex, so that it produces an expression of the form $\hat{f}(p,q)$.

6.7.7 Identify the two-body operator

The two-body operator is identified as the two-body vertex with a dotted horizontal line, and the labels entering and/or leaving the vertex produce the following interaction:

$$\langle leftout, rightout || leftin, rightin \rangle \tag{6.37}$$

6.7.8 Identify the amplitudes

Each amplitude will occur as solid horizontal lines with particles and holes above it. Depending on the labeling, they are denoted algebraically as $\hat{t}_i^a$, $\hat{t}_{ij}^{ab}$, $\hat{t}_{ijk}^{abc}$ and $\hat{t}_{ijkl}^{abcd}$, for $\hat{T}_1$,$\hat{T}_2$,$\hat{T}_3$ and $\hat{T}_4$ respectively.

6.7.9 Summation indices

We then sum over all so-called internal indices that connect the amplitudes to the interaction. These are easily identified as the only connected lines in the diagram.

6.7.10 Identify equivalent internal lines

Equivalent lines are pairs of lines that connect at the same amplitude and interaction, in the same direction (particle-particle or hole-hole). For each such pair, we multiply the expression by a factor of $\frac{1}{2}$.

6.7.11 Identify equivalent $\boldsymbol{T}$-vertices

T-vertices are considered equivalent if they connect to the same interaction vertex in the same configurational pattern. For each such pair, multiply by a factor of $\frac{1}{2}$.

6.7.12 The phase factor

Count the number of hole lines and loops, and multiply the expression with a phase factor given by $(-)^{n_{holes}-n_{loops}}$. The number of holes is the number of lines pointing downwards, and a loop is easily identified as a pair of particle and hole lines connecting to the same two vertices[1].

6.7.13 External permutations

A pair of external lines (unconnected lines) are considered equivalent if they connect to the same vertex. We must sum over all inequivalent external lines, and include a parity factor of $(-)^{\sigma(P)}$ where $\sigma(P)$ is the number of permutations.

6.7.14 Cancel factors caused by external permutations

For each pair of external lines connected to equivalent vertices, cancel one factor of $\frac{1}{2}$ caused by equivalent T-vertices.

6.7.15 The correlation energy

As may be seen from the energy equation (see Eq. (6.26)), the diagrams that occur in this equation should have excitation level 0. When considering possible diagrams, this obviously means that the Hamiltonian term should not have any unconnected lines

[1] Actually it is possible to draw diagrams in a manner that makes these loops hard to spot, but we shall not concern ourselves about this here.

above the interaction, while all lines from the amplitudes should be connected to the interaction. In other words, we seek diagrams composed of only internal lines.

The only terms in the Hamiltonian that fulfill these criterions are the ones where all lines occur below the interaction, shown in Figs. 6.8 and 6.9. Considering the excitation level of these interactions, we find that they cause an excitation level of -1 for Fig. 6.8 and -2 for Fig. 6.9.

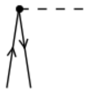


Figure 6.8

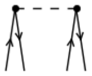


Figure 6.9

It is then quite clear that only very few combinations are possible. For example, no excitation beyond doubles will enter the energy expression, since they will excite the reference state beyond the +2 level. Only the single excitation operator $\hat{T}_1$ (see Fig. 6.2) will be able to produce an excitation level of +1, and thus connect with the one-body operator in Fig. 6.8.

We will find a total of three different diagrams that contribute to the energy, shown in Figs. 6.10, 6.11 and 6.12.

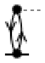


Figure 6.10

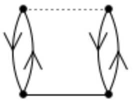


Figure 6.11

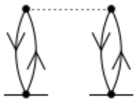


Figure 6.12

To translate these diagrams into algebraic expressions, we apply the rules for diagram interpretation. We find

$$\Delta e = \sum_{ck} \langle k||c\rangle t_k^c + \frac{1}{4}\sum_{cdkl} \langle kl||cd\rangle t_{kl}^{cd} + \frac{1}{2}\sum_{ckdl} \langle kl||cd\rangle t_k^c t_l^d. \tag{6.38}$$

6.8 Diagrams as code

Although the diagrammatic rules greatly simplify the process of deriving the equations, there is still a high probability for errors due to human inaccuracy. The CCSD $\hat{T}_1$ amplitude equations have for example 14 terms, making even the diagrammatic process quite tedious. This is reflected even in authoritative works on the subjects such as Shavitt and Bartlett's *"Many Body Methods in Chemistry and Physics"* [7], where errors in the equations have been found[2].

Hermann Kümmel noted in his "Biography of the Coupled Cluster Method" [10], that

"Because of the often large number of terms in all versions of the CCM it is rather hard work to obtain the explicit equations by hand. And after this is done one has to write a program to put them into the computer."

Is it really necessary to derive equations by hand and thereafter write code manually? Probably not, and for this reason it makes sense to ensure consistency in the derivation process by use of computers. If equations are derived by the computer, we may also task the computer to actually generate the code needed for numerical computation.

A number of symbolic frameworks that can handle such operations exists such as *Second Quantization* for SymPy (Python) [18], but these are based on the algebraic approach with Wick's theorem. As a part of the work on this thesis, a python script was developed aimed at deriving the CC equations at any level of truncation using the diagrammatic approach.

In this section, we shall briefly discuss how to translate the diagrammatic rules into code, and then proceed to derive the explicit equations for various truncations by use of this code.

6.8.1 Implementation

We will need to define operators in terms of unconnected lines above and below the interaction. It is natural to define a class for these operators, and define functions such as contractions that takes operator classes as parameters. A function for contractions of operators should return all distinct diagrams for the operators.

An operator is sufficiently described by its number of hole- and particle lines above and below the vertex. This is easily translated into two arrays for each operator, with a binary representation of particle or holes (for example 0 and 1).

To perform contractions of such operators, we will first need to set up the sequence of vertical bars and plus- and minus signs as previously described. This is basically just a new array, created by joining the array of particles and holes below the interaction vertex with a number of vertical bars equal to the number of excitation operators minus one.

[2] See as well Chapter 8 concerning the CCDT-1 validation.

The code then needs to seek through all possible permutations of this sequence and identify the valid sequences. Such valid sequences may be identified by connections between the Hamiltonian and all excitation operators to its right, and by the number of connections occurring in each excitation operator not exceeding the possible number of connections.

Next we need to avoid over counting connection patterns due to equivalent excitation operators, so we identify identical excitation operators and keep only one of each such equivalent connection pattern.

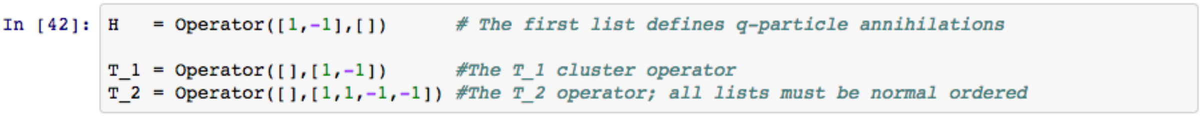
Finally we end up with a number of unique connection patterns that gives us a non-ambiguous recipe for constructing the diagrams.

On top of this functionality we may want a framework for plotting the diagrams or interpret them as algebraic expressions or code. IPython Notebook [34] provides an excellent environment for these kinds of operations. In the following sections we will explore such functionalities by discussing the software *CCAlgebra* developed in connection with the work with this thesis. The software *CCAlgebra* allows one to derive automatically all possible approximations in coupled cluster theory, providing thereby a benchmark to equations derived by paper and pencil. This provides an invaluable to benchmark the equations that enter our codes. Furthermore, the software *CCAlgebra* allows also for automatic generation of code, as well as mathematical expressions and figures.

6.8.2 CCAlgebra

CCAlgebra is a python software developed by the present author that may be imported into any IPython notebook. It has no dependencies outside those libraries that are normally included in extended python distributions such as Entougth Canopy [**?**] or Continuum Analytics Anaconda [35].

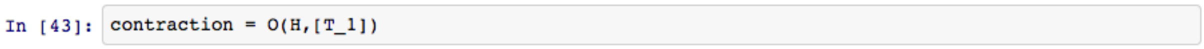
To start a session one just starts IPython Notebook and imports the CCAlgebra.py file into the notebook. It is then possible to define operators in the following manner



```
In [42]: H   = Operator([1,-1],[])       # The first list defines q-particle annihilations

         T_1 = Operator([],[1,-1])       #The T_1 cluster operator
         T_2 = Operator([],[1,1,-1,-1])  #The T_2 operator; all lists must be normal ordered
```

Above we defined three operators; the Hamiltonian term from Fig. 6.8, and the single and double excitation operators $\hat{T}_1$ and $\hat{T}_2$. Each operator is fully described by two lists; the first representing the pseudo particle annihilation operators, while the second represents pseudo particle creation operators.

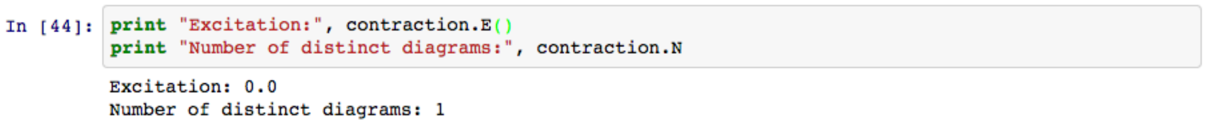
To contract these operators we use a function that takes the Hamiltonian operator and a list of excitation operators as parameters.



```
In [43]: contraction = O(H,[T_1])
```

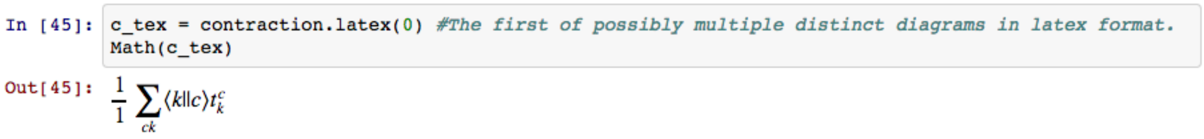
This will return an object that contains all possible diagrams generated from the contraction of these operators. The reason for that excitation operators should be brace-enclosed in a list is that this list contains all excitation operators to the right of the Hamiltonian, consistent with the prior derivation from the Hausdorff expansion.

Some simple information may be obtained from the resulting contraction, such as the excitation level and the number of resulting distinct diagrams



```
In [44]: print "Excitation:", contraction.E()
         print "Number of distinct diagrams:", contraction.N

         Excitation: 0.0
         Number of distinct diagrams: 1
```

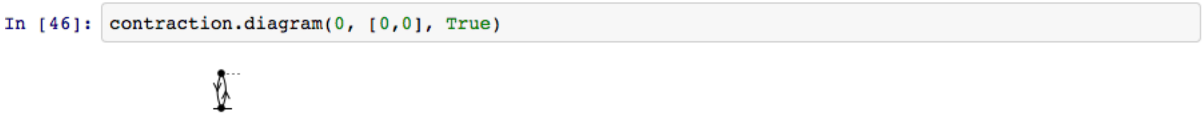
The resulting diagrams are numbered from 0 and up, so as we only have one diagram resulting from this contraction we may display it as latex formatted text by the *.latex()* method



```
In [45]: c_tex = contraction.latex(0) #The first of possibly multiple distinct diagrams in latex format.
         Math(c_tex)
```

Out[45]: $\frac{1}{1}\sum_{ck}\langle k||c\rangle t_k^c$

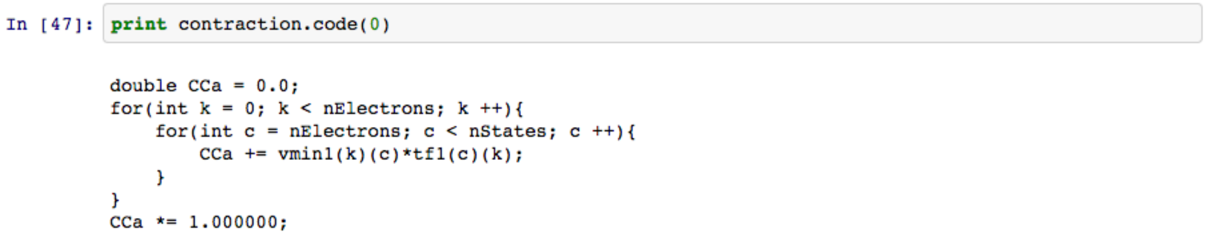
We may also display it as a diagram by using the *.diagram()* method.



```
In [46]: contraction.diagram(0, [0,0], True)
```

The first parameter here specifies the number of the diagram (beginning with 0), coordinates on the screen if we would like to display multiple diagrams in the same frame, and finally an on/off option for using built in formatting (turning of axis and scaling) or letting the outside script adjust these parameters (”false”).

With the computer’s understanding of the diagram it is now very easy to translate it directly into code:

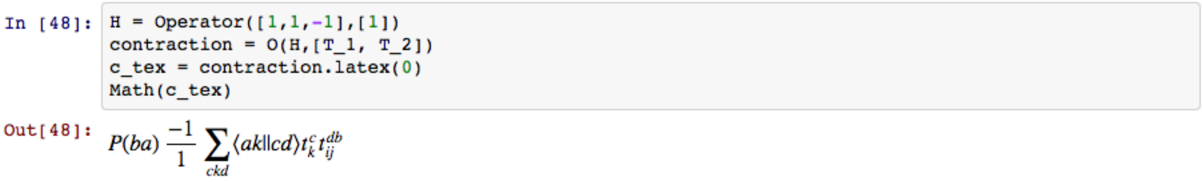


```
In [47]: print contraction.code(0)

         double CCa = 0.0;
         for(int k = 0; k < nElectrons; k ++){
             for(int c = nElectrons; c < nStates; c ++){
                 CCa += vmin1(k)(c)*tf1(c)(k);
             }
         }
         CCa *= 1.000000;
```

In this case, the code is a naive C++ implementation (*naive* in the sense that it is the most direct translation of the diagram, using for-loops where sums occur), but in principle it is possible to generate any kind of code. Quite possibly also highly optimized code if we supply the CCAlgebra script with some more information about the system we want to calculate.

Terms in the CC equations may now be represented as code, diagrams or mathematical expressions depending on how we prefer to view them. It does make sense to refer to these terms simply as ”diagrams”.

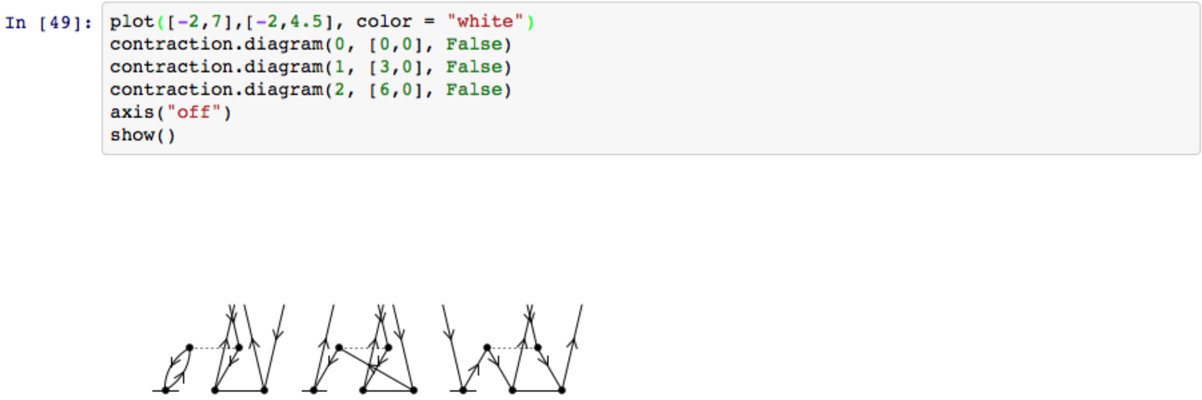
We may of course now easily derive more complex diagrams, simply by defining the operators we seek:



```
In [48]: H = Operator([1,1,-1],[1])
         contraction = O(H,[T_1, T_2])
         c_tex = contraction.latex(0)
         Math(c_tex)
```

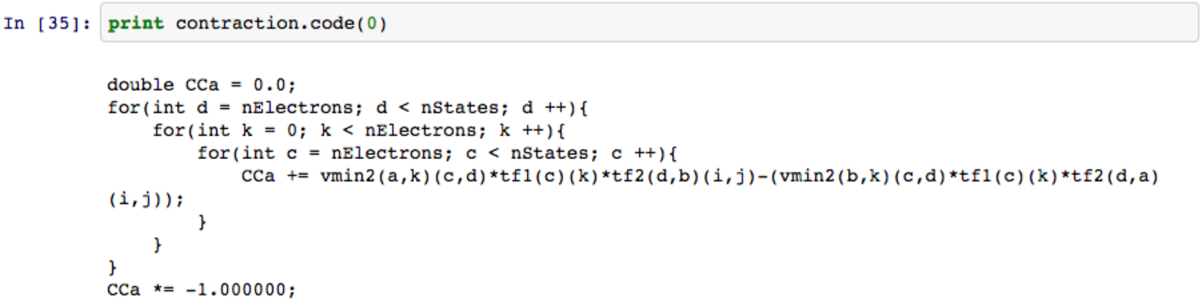
Out[48]: $P(ba)\frac{-1}{1}\sum_{ckd}\langle ak||cd\rangle t_{k}^{c}t_{ij}^{db}$

Many physicists find diagrams helpful to quickly gain insight into the nature of a given contribution. We have thus added to our software the functionality to quickly render diagrams on screen.



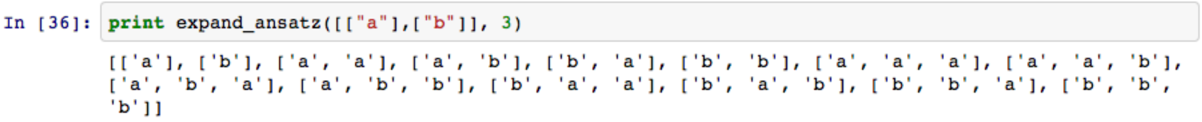
```
In [49]: plot([-2,7],[-2,4.5], color = "white")
         contraction.diagram(0, [0,0], False)
         contraction.diagram(1, [3,0], False)
         contraction.diagram(2, [6,0], False)
         axis("off")
         show()
```

In this case we used a two-body term from the Hamiltonian, and we sum over three internal lines. This results in a more computationally intensive code with three nested for-loops:



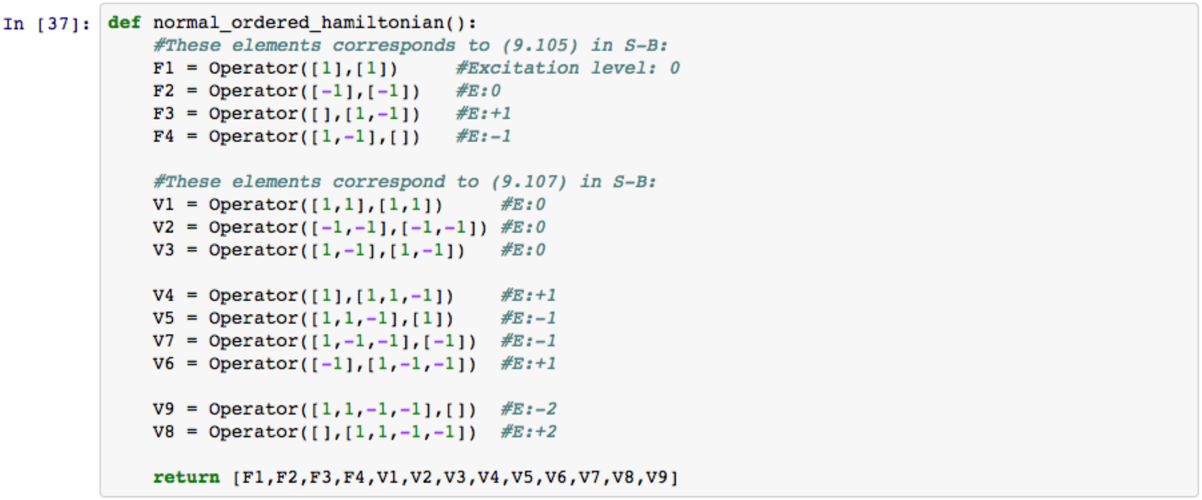
```
In [35]: print contraction.code(0)

         double CCa = 0.0;
         for(int d = nElectrons; d < nStates; d ++){
             for(int k = 0; k < nElectrons; k ++){
                 for(int c = nElectrons; c < nStates; c ++){
                     CCa += vmin2(a,k)(c,d)*tf1(c)(k)*tf2(d,b)(i,j)-(vmin2(b,k)(c,d)*tf1(c)(k)*tf2(d,a)
         (i,j));
                 }
             }
         }
         CCa *= -1.000000;
```

In our equations the cluster operator occurs in a power series, resulting in a multitude of products of excitation operators. To handle this more easily and avoid the need to precalculate these, some complementary functions to quickly evaluate such expanded power series of the cluster operator is included:



```
In [36]: print expand_ansatz([["a"],["b"]], 3)

         [['a'], ['b'], ['a', 'a'], ['a', 'b'], ['b', 'a'], ['b', 'b'], ['a', 'a', 'a'], ['a', 'a', 'b'],
         ['a', 'b', 'a'], ['a', 'b', 'b'], ['b', 'a', 'a'], ['b', 'a', 'b'], ['b', 'b', 'a'], ['b', 'b',
         'b']]
```

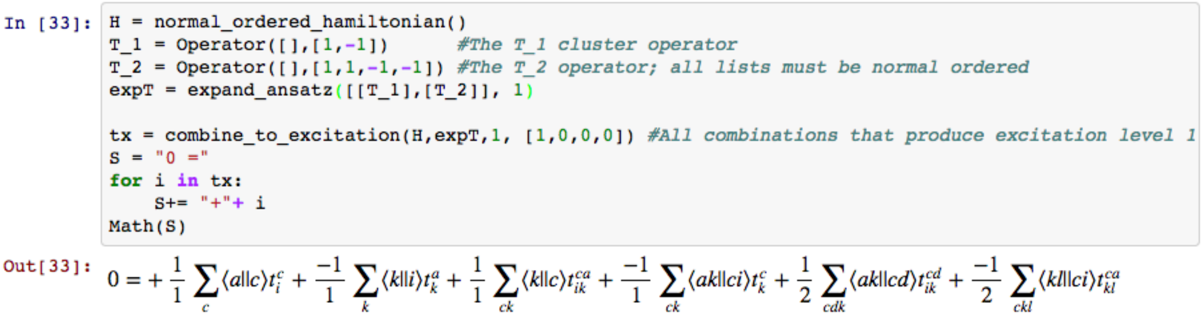
It is also convenient to predefine some operators that occur often. A normal ordered Hamiltonian will come in handy:



```
In [37]: def normal_ordered_hamiltonian():
             #These elements corresponds to (9.105) in S-B:
             F1 = Operator([1],[1])      #Excitation level: 0
             F2 = Operator([-1],[-1])    #E:0
             F3 = Operator([],[1,-1])    #E:+1
             F4 = Operator([1,-1],[])    #E:-1

             #These elements correspond to (9.107) in S-B:
             V1 = Operator([1,1],[1,1])        #E:0
             V2 = Operator([-1,-1],[-1,-1]) #E:0
             V3 = Operator([1,-1],[1,-1])      #E:0

             V4 = Operator([1],[1,1,-1])       #E:+1
             V5 = Operator([1,1,-1],[1])       #E:-1
             V7 = Operator([1,-1,-1],[-1])  #E:-1
             V6 = Operator([-1],[1,-1,-1])  #E:+1

             V9 = Operator([1,1,-1,-1],[])  #E:-2
             V8 = Operator([],[1,1,-1,-1])  #E:+2

             return [F1,F2,F3,F4,V1,V2,V3,V4,V5,V6,V7,V8,V9]
```

6.8.3 Deriving amplitude equations using CCAlgebra

Another convenient tool is a function that produces all possible diagrams with a given excitation level from a normal ordered Hamiltonian and the expanded exponential ansatz, in effect generating the full energy- and amplitude equations. In the following example we find all contributions to the $\hat{T}_1$ amplitude equation in the CCSD truncation ($\hat{T} = \hat{T}_1 + \hat{T}_2$)

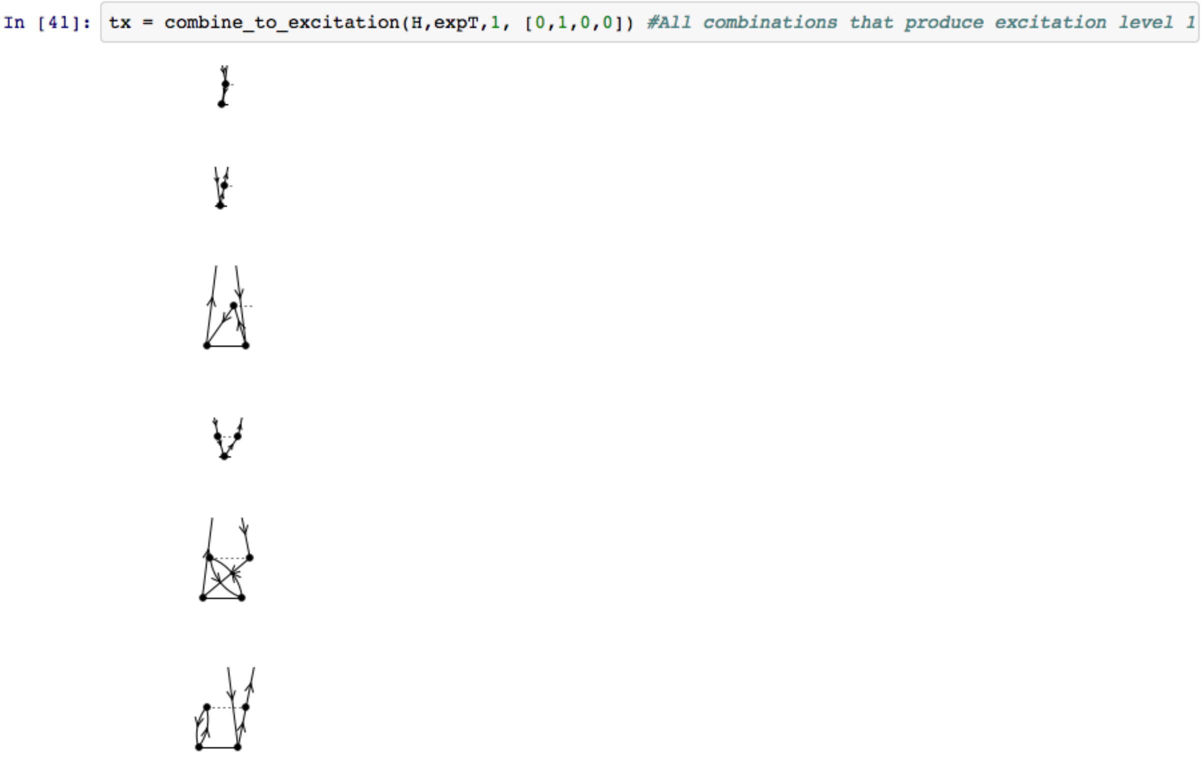


```
In [33]: H = normal_ordered_hamiltonian()
         T_1 = Operator([],[1,-1])       #The T_1 cluster operator
         T_2 = Operator([],[1,1,-1,-1]) #The T_2 operator; all lists must be normal ordered
         expT = expand_ansatz([[T_1],[T_2]], 1)

         tx = combine_to_excitation(H,expT,1, [1,0,0,0]) #All combinations that produce excitation level 1
         S = "0 ="
         for i in tx:
             S+= "+"+ i
         Math(S)
```

Out[33]:

$$0 = +\frac{1}{1}\sum_{c}\langle a||c\rangle t_{i}^{c} + \frac{-1}{1}\sum_{k}\langle k||i\rangle t_{k}^{a} + \frac{1}{1}\sum_{ck}\langle k||c\rangle t_{ik}^{ca} + \frac{-1}{1}\sum_{ck}\langle ak||ci\rangle t_{k}^{c} + \frac{1}{2}\sum_{cdk}\langle ak||cd\rangle t_{ik}^{cd} + \frac{-1}{2}\sum_{ckl}\langle kl||ci\rangle t_{kl}^{ca}$$

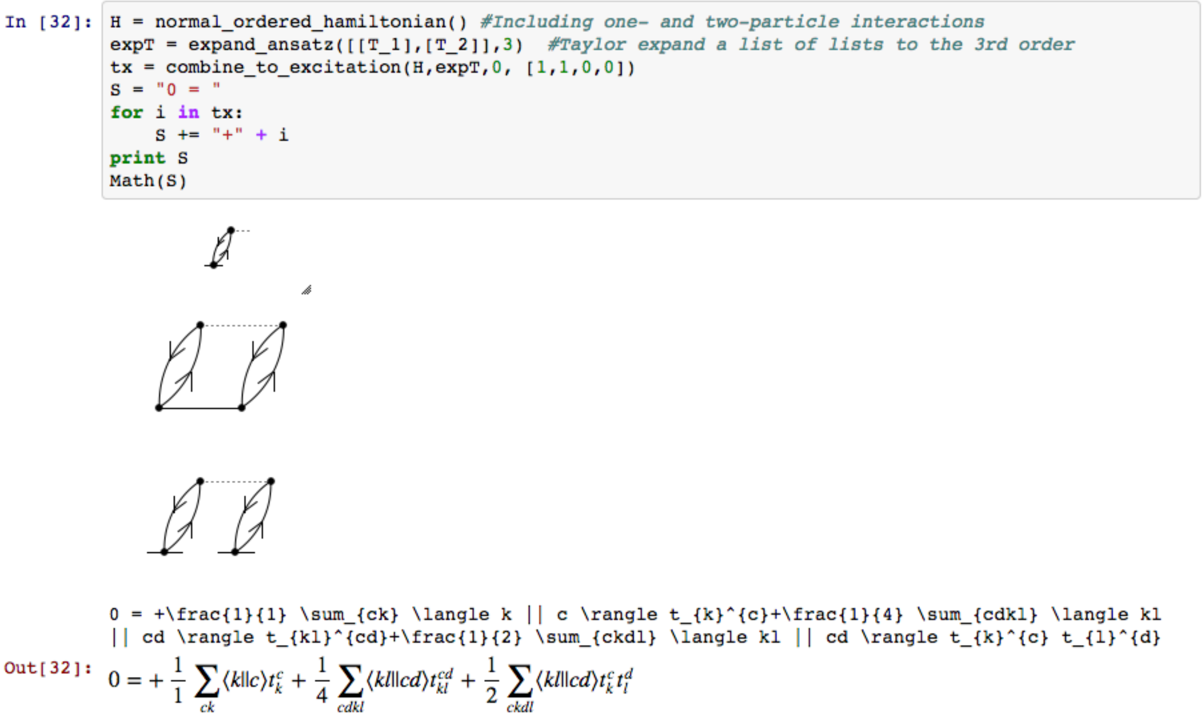
The diagram are still a bit awkwardly formatted, but they are correct:



```
In [41]: tx = combine_to_excitation(H,expT,1, [0,1,0,0]) #All combinations that produce excitation level 1
```

6.8.4 Deriving the energy using CCalgebra

We have already noted how only a very few terms enter the energy equations. Using CCAlgebra we may confirm that this is the indeed the case



```
In [32]: H = normal_ordered_hamiltonian() #Including one- and two-particle interactions
         expT = expand_ansatz([[T_1],[T_2]],3)  #Taylor expand a list of lists to the 3rd order
         tx = combine_to_excitation(H,expT,0, [1,1,0,0])
         S = "0 = "
         for i in tx:
             S += "+" + i
         print S
         Math(S)
```

```
0 = +\frac{1}{1} \sum_{ck} \langle k || c \rangle t_{k}^{c}+\frac{1}{4} \sum_{cdkl} \langle kl
|| cd \rangle t_{kl}^{cd}+\frac{1}{2} \sum_{ckdl} \langle kl || cd \rangle t_{k}^{c} t_{l}^{d}
```

Out[32]:

$$0 = +\frac{1}{1}\sum_{ck}\langle k||c\rangle t_{k}^{c} + \frac{1}{4}\sum_{cdkl}\langle kl||cd\rangle t_{kl}^{cd} + \frac{1}{2}\sum_{ckdl}\langle kl||cd\rangle t_{k}^{c}t_{l}^{d}$$

By using CCAlgebra we may now go on to derive the explicit equations for the various CC methods.

6.8.5 Validation of CCAlgebra

The developement of the CCAlgebra framework was mainly intended as an educational tool for better understanding of the CC equations and their derivation. As such, the code is written mainly as a proof of concept.

Although the generated code from CCAlgebra is not optimized to such an extent that it will be usable for the system we explore in this thesis, it is fully capable to generate code that reproduces accurate results for well-known systems such as CCSD dissociation energies for the hydrogen molecule using a Gaussian basis set, as briefly described in the section on size consistency. The code generated by CC algebra was in exact agreement with the code written by the present author for the CCSD case with only minor manual edits in the C++ code.

To a certain extent this validates the results obtained by the use of CCAlgebra, but it is still highly recommended to compare the equations and diagrams with the ones present in the literature before implementation.

6.9 The CCD equations

If we include only the double excitation operator in our cluster operator we get the CCD (Coupled Cluster Doubles) equations. This means that we let

$$\hat{T} = \hat{T}_2. \tag{6.39}$$

We then perform the contractions with CCAlgebra. Since no single excitations is included, we get an even simpler expression for the energy than what we had earlier

$$\Delta e = \frac{1}{4}\sum_{cdkl}\langle kl||cd\rangle t_{kl}^{cd}. \tag{6.40}$$

The only amplitude equation in this truncation is for $\hat{T}_2$:

$$\begin{aligned}0 = \langle ab||ij\rangle &+ P(ij)\sum_k f_{kj}t_{ki}^{ab} + P(ab)\sum_c f_{bc}t_{ij}^{ac} + \frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} \\ &+ \frac{1}{2}\sum_{kl}\langle kl||ij\rangle t_{kl}^{ab} - P(ba)P(ij)\sum_{ck}\langle ak||cj\rangle t_{ik}^{cb} - P(ij)\frac{1}{2}\sum_{cdkl}\langle kl||cd\rangle t_{ik}^{cd}t_{jl}^{ab} \\ &+ \frac{1}{4}\sum_{cdkl}\langle kl||cd\rangle t_{ij}^{cd}t_{kl}^{ab} - P(ab)\frac{1}{2}\sum_{ckld}\langle kl||cd\rangle t_{kl}^{ca}t_{ij}^{db} + P(ab)P(ij)\frac{1}{2}\sum_{ckdl}\langle kl||cd\rangle t_{ik}^{ca}t_{jl}^{db}.\end{aligned} \tag{6.41}$$

There are a lot of different diagrams occurring in the CC equations and it will make sense to classify them in ways that make them more easily distinguishable. What is common to all diagrams is that they contain only one Hamiltonian term contracted to possibly more than one excitation operator. For this reason we may classify them into orders of the various excitation operators occurring in them, and in this simple case with only one excitation operator we may assign a letter L to diagrams *linear* in $\hat{T}_2$ and the letter Q to diagrams *quadratic* in $\hat{T}_2$. Since there are multiple diagrams corresponding to each letter, we also assign a subscript letter in their order of appearance.

The naming of each diagram is listed in table 6.2, but for now we just group together the terms accordingly and define

$$L(t_{ij}^{ab}) \equiv \frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} + \frac{1}{2}\sum_{kl}\langle kl||ij\rangle t_{kl}^{ab} - P(ba)P(ij)\sum_{ck}\langle ak||cj\rangle t_{ik}^{cb}. \tag{6.42}$$

$$\begin{aligned}Q(t_{ij}^{ab}t_{ij}^{ab}) \equiv P(ij)\frac{1}{2}\sum_{cdkl}\langle kl||cd\rangle t_{ik}^{cd}t_{jl}^{ab} + \frac{1}{4}\sum_{cdkl}\langle kl||cd\rangle t_{ij}^{cd}t_{kl}^{ab} \\ - P(ab)\frac{1}{2}\sum_{ckld}\langle kl||cd\rangle t_{kl}^{ca}t_{ij}^{db} + P(ab)P(ij)\frac{1}{2}\sum_{ckdl}\langle kl||cd\rangle t_{ik}^{ca}t_{jl}^{db},\end{aligned} \tag{6.43}$$

and write the CCD equation

$$0 = \langle ab||ij\rangle + P(ij)\sum_k f_{kj}t_{ki}^{ab} + P(ab)\sum_c f_{bc}t_{ij}^{ac} + L(t_{ij}^{ab}) + Q(t_{ij}^{ab}t_{ij}^{ab}). \tag{6.44}$$

It is not immediately clear how we are supposed to solve this equation, but it is possible to rearrange the terms so that a self-consistency criteria is found. This may be done by factoring out the diagonal elements in the diagrams containing the one-body operator, so that

$$P(ij)\sum_{k} f_{kj}t_{ki}^{ab} = \sum_{k\neq j} f_{kj}t_{ki}^{ab} + f_{jj}t_{ji}^{ab} - \sum_{k\neq i} f_{ki}t_{kj}^{ab} - f_{ii}t_{ij}^{ab} = -(f_{jj}+f_{ii})t_{ij}^{ab} + \sum_{k\neq j} f_{kj}t_{ki}^{ab} - \sum_{k\neq i} f_{ki}t_{kj}^{ab}, \tag{6.45}$$

and

$$P(ab)\sum_{c} f_{ac}t_{ij}^{cb} = \sum_{c\neq a} f_{ac}t_{ij}^{cb} + f_{aa}t_{ij}^{ab} - \sum_{c\neq b} f_{bc}t_{ij}^{ca} - f_{bb}t_{ij}^{ba} = (f_{aa}+f_{bb})t_{ij}^{ab} + \sum_{c\neq a} f_{ac}t_{ij}^{cb} - \sum_{c\neq b} f_{bc}t_{ij}^{ca}. \tag{6.46}$$

In a canonical Hartree-Fock basis the one-body operator will only contain diagonal elements, and since we will do all upcoming calculations in such a basis we might as well ignore these non-diagonal sums. The full CCD amplitude equation then becomes

$$0 = \langle ab||ij\rangle - (f_{jj} + f_{ii})t_{ij}^{ab} + (f_{aa} + f_{bb})t_{ij}^{ab} + L(t_{ij}^{ab}) + Q(t_{ij}^{ab}t_{ij}^{ab}). \tag{6.47}$$

We may rewrite this expression to

$$t_{ij}^{ab} = \frac{\langle ab||ij\rangle + L(t_{ij}^{ab}) + Q(t_{ij}^{ab}t_{ij}^{ab})}{f_{jj} + f_{ii} - f_{aa} - f_{bb}}. \tag{6.48}$$

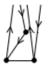
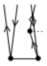
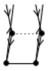
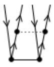
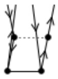
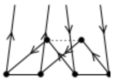
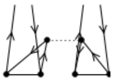
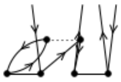
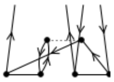
With this expression we will be able to iteratively obtain self consistence of the amplitudes. A reasonable guess for initial value for the amplitudes is found by letting the L and Q term be zero, so we have

$$t_{ij}^{ab} = \frac{\langle ab||ij\rangle}{f_{jj} + f_{ii} - f_{aa} - f_{bb}}. \tag{6.49}$$

These amplitudes actually correspond to second order in many-body perturbation energy, since the evaluation of the energy expression for the CCD upon initialization yields

$$\Delta e^{(2)} = \frac{\langle ij||ab\rangle\langle ab||ij\rangle}{f_{jj} + f_{ii} - f_{aa} - f_{bb}}. \tag{6.50}$$

Table 6.2: Diagrams CCD amplitude equation

Name	Factor	Permutation	Interpretation	Diagram
	-1	$P(ba)$	$\sum_c f_{a,c} t_{ij}^{cb}$	
		$P(ij)$	$\sum_k f_{k,j} t_{ik}^{ab}$	
L_a	$\frac{1}{2}$		$\sum_{cd} \langle ab\|\|cd\rangle t_{ij}^{cd}$	
L_b	$\frac{1}{2}$		$\sum_{kl} \langle kl\|\|ij\rangle t_{kl}^{ab}$	
L_c	-1	$P(ba)P(ij)$	$\sum_{ck} \langle ak\|\|cj\rangle t_{ik}^{cb}$	
$Q_a(D_{3a})$	$\frac{1}{4}$		$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ij}^{cd} t_{kl}^{ab}$	
$Q_b(D_{3b})$	$\frac{1}{2}$	$P(ab)P(ij)$	$\sum_{ckdl} \langle kl\|\|cd\rangle t_{ik}^{ca} t_{jl}^{db}$	
$Q_c(D_{3c})$	$-\frac{1}{2}$	$P(ab)$	$\sum_{ckld} \langle kl\|\|cd\rangle t_{kl}^{ca} t_{ij}^{db}$	
$Q_d(D_{3d})$	$-\frac{1}{2}$	$P(ij)$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ik}^{cd} t_{jl}^{ab}$	

6.10 The CCSD equations

To improve upon the results from the CCD truncation the next step is to include single excitations from the $\hat{T}_1$ excitation operator. Although no such excitations occur on the system subject to this thesis, we review this contributions here for completeness.

By inclusion of the single excitations, we have the cluster operator

$$\hat{T} = \hat{T}_1 + \hat{T}_2. \tag{6.51}$$

The diagrams contributing to the amplitude equations for $\hat{T}_1$ and $\hat{T}_2$ are given in tables 6.3 and 6.4 6.4.

In the same way as for the $\hat{T}_2$ amplitude, we may factor out terms from diagrams S_{3a} and S_{3b} to initialize the $\hat{T}_1$ amplitude, resulting in

$$t_i^a = \frac{f_{a,i}}{f(i,i) - f(a,a)}. \tag{6.52}$$

The next amplitude is found in the same manner as for the $\hat{T}_2$ amplitude for the CCD approximation, and we iterate until some convergence criteria is fulfilled.

Table 6.3: Contributions to the CCSD $\hat{T}_1$ amplitude equation

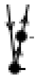
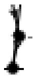
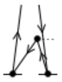
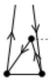
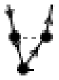
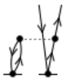
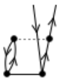
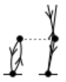
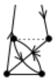
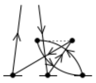
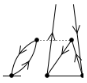
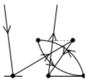
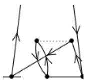
Name	Factor	Permutation	Interpretation	Diagram
S_{3a}			$\sum_c f_{a,c} t_i^c$	
S_{3b}	-1		$\sum_k f_{k,i} t_k^a$	
S_{5a}			$\sum_{ck} f_{k,c} t_i^c t_k^a$	
S_{2a}			$\sum_{ck} f_{k,c} t_{ik}^{ca}$	
S_{3c}	-1		$\sum_{ck} \langle ak\|\|ci\rangle t_k^c$	
S_{5b}			$\sum_{ckd} \langle ak\|\|cd\rangle t_k^c t_i^d$	
S_{2b}	$\frac{1}{2}$		$\sum_{cdk} \langle ak\|\|cd\rangle t_{ik}^{cd}$	
S_{5c}	-1		$\sum_{ckl} \langle kl\|\|ci\rangle t_k^c t_l^a$	
S_{2c}	$-\frac{1}{2}$		$\sum_{ckl} \langle kl\|\|ci\rangle t_{kl}^{ca}$	
S_6			$\sum_{ckdl} \langle kl\|\|cd\rangle t_i^c t_k^a t_l^d$	
S_{4c}			$\sum_{ckdl} \langle kl\|\|cd\rangle t_k^c t_{il}^{da}$	
S_{4a}	$\frac{1}{2}$		$\sum_{cdkl} \langle kl\|\|cd\rangle t_i^c t_{kl}^{da}$	
S_{4b}	$\frac{1}{2}$		$\sum_{kcdl} \langle kl\|\|cd\rangle t_k^a t_{il}^{cd}$	

TABLE 6.4: Contributions to the CCSD $\hat{T}_2$ amplitude equation (1)

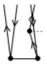
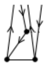
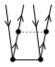
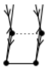
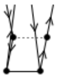
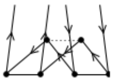
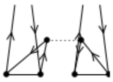
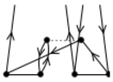
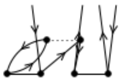
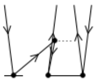
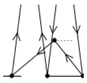
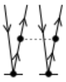
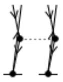
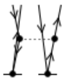
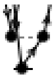
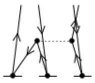
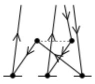
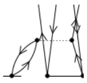
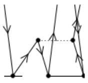
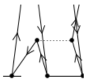
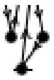
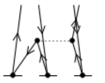
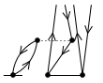
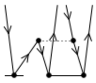
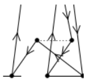
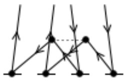
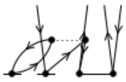
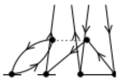
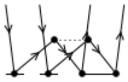
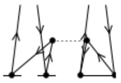
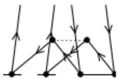
Name	Factor	Permutation	Interpretation	Diagram
	-1	$P(ba)$	$\sum_c f_{a,c} t_{ij}^{cb}$	
		$P(ij)$	$\sum_k f_{k,j} t_{ik}^{ab}$	
L_a	$\frac{1}{2}$		$\sum_{cd} \langle ab\|\|cd\rangle t_{ij}^{cd}$	
L_b	$\frac{1}{2}$		$\sum_{kl} \langle kl\|\|ij\rangle t_{kl}^{ab}$	
L_c	-1	$P(ba)P(ij)$	$\sum_{ck} \langle ak\|\|cj\rangle t_{ik}^{cb}$	
D_{3a}	$\frac{1}{4}$		$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ij}^{cd} t_{kl}^{ab}$	
D_{3b}	$\frac{1}{2}$	$P(ab)P(ij)$	$\sum_{ckdl} \langle kl\|\|cd\rangle t_{ik}^{ca} t_{jl}^{db}$	
D_{3c}	$-\frac{1}{2}$	$P(ab)$	$\sum_{ckld} \langle kl\|\|cd\rangle t_{kl}^{ca} t_{ij}^{db}$	
D_{3d}	$-\frac{1}{2}$	$P(ij)$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ik}^{cd} t_{jl}^{ab}$	
D_{5a}	-1	$P(ij)$	$\sum_{ck} f_{k,c} t_i^c t_{jk}^{ab}$	
D_{5b}	-1	$P(ab)$	$\sum_{kc} f_{k,c} t_k^a t_{ij}^{cb}$	
D_{6a}	$-\frac{1}{2}$	$P(ij)$	$\sum_{cd} \langle ab\|\|cd\rangle t_i^c t_j^d$	
D_{6b}	$-\frac{1}{2}$	$P(ab)$	$\sum_{kl} \langle kl\|\|ij\rangle t_k^a t_l^b$	
D_{6c}	-1	$P(ij)P(ba)$	$\sum_{ck} \langle ak\|\|cj\rangle t_i^c t_k^b$	
D_{4a}		$P(ij)$	$\sum_c \langle ab\|\|cj\rangle t_i^c$	
D_{8a}	$\frac{1}{2}$	$P(ij)P(ba)$	$\sum_{cdk} \langle ak\|\|cd\rangle t_i^c t_j^d t_k^b$	
D_{8b}	$\frac{1}{2}$	$P(ij)P(ba)$	$\sum_{cdk} \langle ak\|\|cd\rangle t_i^c t_j^d t_k^b$	
D_{5g}	-1	$P(ba)$	$\sum_{ckd} \langle ak\|\|cd\rangle t_k^c t_{ij}^{db}$	
D_{5c}		$P(ij)P(ba)$	$\sum_{cdk} \langle ak\|\|cd\rangle t_i^c t_{jk}^{db}$	
D_{5e}	$-\frac{1}{2}$	$P(ab)$	$\sum_{kcd} \langle kb\|\|cd\rangle t_k^a t_{ij}^{cd}$	
D_{4b}	-1	$P(ab)$	$\sum_k \langle kb\|\|ij\rangle t_k^a$	

Table 6.5: Contributions to the CCSD $\hat{T}_2$ amplitude equation (2)

Name	Factor	Permutation	Interpretation	Diagram
D_{8b}	$-\frac{1}{2}$	$P(ij)P(ab)$	$\sum_{ckl}\langle kl\|\|cj\rangle t_i^c t_k^a t_l^b$	
D_{5h}		$P(ji)$	$\sum_{ckl}\langle kl\|\|ci\rangle t_k^c t_{jl}^{ab}$	
D_{5f}	$\frac{1}{2}$	$P(ij)$	$\sum_{ckl}\langle kl\|\|cj\rangle t_i^c t_{kl}^{ab}$	
D_{5d}	-1	$P(ab)P(ji)$	$\sum_{kcl}\langle kl\|\|ic\rangle t_k^a t_{jl}^{cb}$	
D_9	$\frac{1}{4}$	$P(ij)P(ab)$	$\sum_{cdkl}\langle kl\|\|cd\rangle t_i^c t_j^d t_k^a t_l^b$	
D_{7d}	-1	$P(ij)$	$\sum_{ckdl}\langle kl\|\|cd\rangle t_k^c t_i^d t_{jl}^{ab}$	
D_{7e}	-1	$P(ab)$	$\sum_{ckld}\langle kl\|\|cd\rangle t_k^c t_l^a t_{ij}^{db}$	
D_{7a}	$-\frac{1}{4}$	$P(ij)$	$\sum_{cdkl}\langle kl\|\|cd\rangle t_i^c t_j^d t_{kl}^{ab}$	
D_{7c}		$P(ij)P(ab)$	$\sum_{ckdl}\langle kl\|\|cd\rangle t_i^c t_k^a t_{jl}^{db}$	
D_{7b}	$-\frac{1}{4}$	$P(ab)$	$\sum_{klcd}\langle kl\|\|cd\rangle t_k^a t_l^b t_{ij}^{cd}$	

6.11 The CCSDT equations

In this section we present the full CCSDT equations as they are generated by our software CCAlgebra for completeness. A thorough (and authorative) derivation and treatment of these equations may be found in [7]. It should be noted that inconsistencies were found between these equations and the ones derived in [7] concerning the signs in diagrams T_{1a}, T_{1b}, and that the signs as they appear in table 6.10 produced results more in agreement with expectations. While the signs in [7] produced CCD(T) energies above the CCD energy, the ones from CCAlgebra produced energies lower than the CCD results (for the three-dimensional homogeneous electron gas), as one would expect for our system.

For the CCSDT equations, we have the cluster operator

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3. \tag{6.53}$$

In table 6.6, all contributions to the CCSDT $\hat{T}_1$ equation are listed, and we find only one extra diagram compared to the CCSD truncation (S_7). The solution procedure for this equation will be the same as for the CCSD and $\hat{T}_1$ amplitudes.

In the CCSDT $\hat{T}_2$ amplitude, we find six additional diagrams when comparing to the CCSD $\hat{T}_2$ amplitudes. The additional diagrams are listed in table 6.7. The solution procedure for this equation shall also be the same as for the CCSD and CCD $\hat{T}_2$ amplitudes, by factoring out terms and using a self consistency procedure.

All contributions to the $\hat{T}_3$ amplitudes are listed in tables 6.8 - 6.9. Following a similar factorization procedure as for the previous equations, we may factor out terms from diagrams T_{2a} and T_{2b} so we find

$$\epsilon^{abc}_{ijk} t^{abc}_{ijk} = T_{1a} + T_{1b} + T_{2c} + ... + T_{10a}, \tag{6.54}$$

where we have defined

$$\epsilon^{abc}_{ijk} = f(i,i) + f(j,j) + f(k,k) - f(a,a) - f(b,b) - f(c,c). \tag{6.55}$$

6.11.1 Computational cost

The computational cost of the full CCSDT equation scales as $n_h^3 n_p^5$ per iteration, where n_p is the number of particle states and n_h is the number of hole states [7, p.316]. This, as well as the fact that such amplitudes actually have to be stored using $n_h^3 n_p^3$ preferably double precision floats, makes implementation of the full CCSDT a formidable challenge.

Table 6.6: Contributions to the CCSDT $\hat{T}_1$ amplitude equation

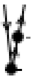
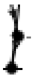
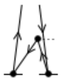
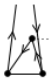
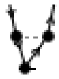
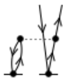
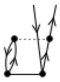
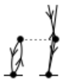
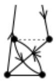
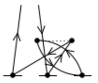
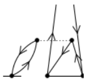
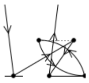
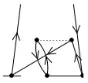
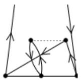
S_{3a}		$\sum_c f_{a,c} t_i^c$	
S_{3b}	-1	$\sum_k f_{k,i} t_k^a$	
S_{5a}		$\sum_{ck} f_{k,c} t_i^c t_k^a$	
S_{2a}		$\sum_{ck} f_{k,c} t_{ik}^{ca}$	
S_{3c}	-1	$\sum_{ck} \langle ak\|\|ci\rangle t_k^c$	
S_{5b}		$\sum_{ckd} \langle ak\|\|cd\rangle t_k^c t_i^d$	
S_{2b}	$\frac{1}{2}$	$\sum_{cdk} \langle ak\|\|cd\rangle t_{ik}^{cd}$	
S_{5c}	-1	$\sum_{ckl} \langle kl\|\|ci\rangle t_k^c t_l^a$	
S_{2c}	$-\frac{1}{2}$	$\sum_{ckl} \langle kl\|\|ci\rangle t_{kl}^{ca}$	
S_6		$\sum_{ckdl} \langle kl\|\|cd\rangle t_i^c t_k^a t_l^d$	
S_{4c}		$\sum_{ckdl} \langle kl\|\|cd\rangle t_k^c t_{il}^{da}$	
S_{4a}	$\frac{1}{2}$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_i^c t_{kl}^{da}$	
S_{4b}	$\frac{1}{2}$	$\sum_{kcdl} \langle kl\|\|cd\rangle t_k^a t_{il}^{cd}$	
S_7	$\frac{1}{4}$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ikl}^{cda}$	

Table 6.7: Additional diagrams in the CCSDT $\hat{T}_2$ amplitude equation (1)

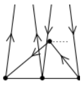
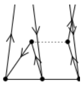
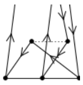
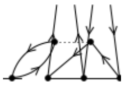
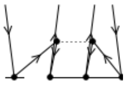
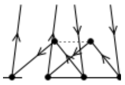
Name	Factor	Permutation	Interpretation	Diagram
D_{10a}			$\sum_{ck} f_{k,c} t_{ijk}^{cab}$	
D_{10b}	$-\frac{1}{2}$	$P(ba)$	$\sum_{cdk} \langle ak\vert\vert cd\rangle t_{ijk}^{cdb}$	
D_{10c}	$\frac{1}{2}$	$P(ij)$	$\sum_{ckl} \langle kl\vert\vert cj\rangle t_{ikl}^{cab}$	
D_{11a}			$\sum_{ckdl} \langle kl\vert\vert cd\rangle t_k^c t_{ijl}^{dab}$	
D_{11c}	$-\frac{1}{2}$	$P(ij)$	$\sum_{cdkl} \langle kl\vert\vert cd\rangle t_i^c t_{jkl}^{dab}$	
D_{11b}	$-\frac{1}{2}$	$P(ab)$	$\sum_{kcdl} \langle kl\vert\vert cd\rangle t_k^a t_{ijl}^{cdb}$	

TABLE 6.8: Contributions to the CCSDT $\hat{T}_3$ amplitude equation (1)

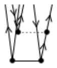
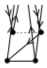
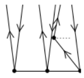
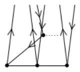
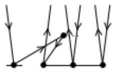
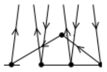
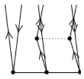
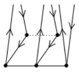
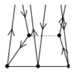
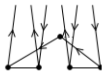
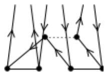
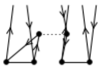
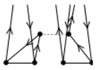
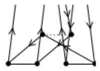
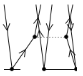
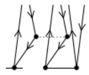
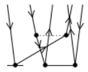
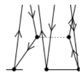
Name	Factor	Permutation	Interpretation	Diagram
T_{1a}	-1	$P(ba)P(bz)P(iw)P(jw)$	$\sum_c \langle ab\|\|ck\rangle t_{ij}^{cb}$	
T_{1b}		$P(az)P(bz)P(ij)P(iw)$	$\sum_k \langle kz\|\|j\rangle t_{ik}^{ab}$	
T_{2a}		$P(ba)P(za)$	$\sum_c f_{a,c} t_{ijw}^{cbz}$	
T_{2b}	-1	$P(iw)P(jw)$	$\sum_k f_{k,w} t_{ijk}^{abz}$	
T_{6a}		$P(ij)P(iw)$	$\sum_{ck} f_{k,c} t_i^c t_{jwk}^{abz}$	
T_{6b}		$P(ab)P(az)$	$\sum_{kc} f_{k,c} t_k^a t_{ijw}^{cbz}$	
T_{2c}	$\frac{1}{2}$	$P(za)P(zb)$	$\sum_{cd} \langle ab\|\|cd\rangle t_{ijw}^{cdz}$	
T_{2d}	$\frac{1}{2}$	$P(iw)P(ij)$	$\sum_{kl} \langle kl\|\|jw\rangle t_{ikl}^{abz}$	
T_{2e}	-1	$P(ba)P(za)P(iw)P(jw)$	$\sum_{ck} \langle ak\|\|cw\rangle t_{ijk}^{cbz}$	
T_{3a}		$P(ab)P(az)P(iw)P(jw)$	$\sum_{ck} f_{k,c} t_{ij}^{ca} t_{wk}^{bz}$	
T_{3d}	$\frac{1}{2}$	$P(iw)P(jw)P(ba)P(za)$	$\sum_{cdk} \langle ak\|\|cd\rangle t_{ij}^{cd} t_{wk}^{bz}$	
T_{3b}	-1	$P(bz)P(ba)P(ij)P(iw)P(za)$	$\sum_{ckd} \langle ak\|\|cd\rangle t_{ik}^{cb} t_{jw}^{dz}$	
T_{3c}		$P(ab)P(az)P(iw)P(ij)P(wj)$	$\sum_{ckl} \langle kl\|\|cj\rangle t_{ik}^{ca} t_{wl}^{bz}$	
T_{3e}	$-\frac{1}{2}$	$P(ab)P(az)P(iw)P(jw)$	$\sum_{ckl} \langle kl\|\|cw\rangle t_{ij}^{ca} t_{kl}^{bz}$	
T_{4a}		$P(ij)P(iw)P(za)P(zb)$	$\sum_{cd} \langle ab\|\|cd\rangle t_i^c t_{jw}^{dz}$	
T_{4b}		$P(ab)P(az)P(ji)P(jw)$	$\sum_{kl} \langle kl\|\|iw\rangle t_k^a t_{jl}^{bz}$	
T_{4c}		$P(ij)P(iw)P(ba)P(za)P(jw)$	$\sum_{ck} \langle ak\|\|cw\rangle t_i^c t_{jk}^{bz}$	
T_{4d}		$P(az)P(ab)P(zb)P(ji)P(wi)$	$\sum_{kc} \langle kb\|\|ic\rangle t_k^a t_{jw}^{cz}$	

TABLE 6.9: Contributions to the CCSDT $\hat{T}_3$ amplitude equation (2)

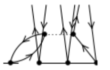
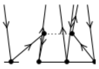
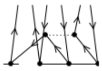
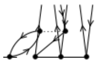
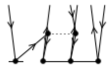
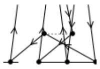
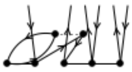
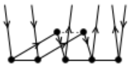
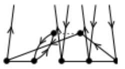
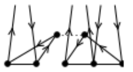
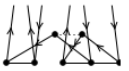
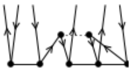
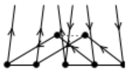
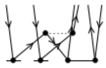
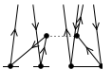
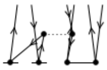
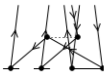
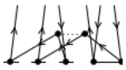
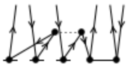
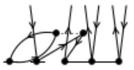
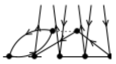
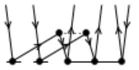
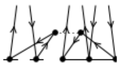
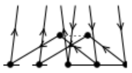
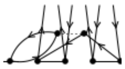
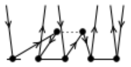
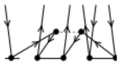
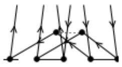
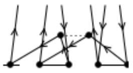
T_{6e}		$P(ba)P(za)$	$\sum_{ckd}\langle ak\vert\vert cd\rangle t_k^c t_{ijw}^{dbz}$	
T_{6c}		$P(ij)P(iw)P(ba)P(za)$	$\sum_{cdk}\langle ak\vert\vert cd\rangle t_i^c t_{jwk}^{dbz}$	
T_{6g}	$-\frac{1}{2}$	$P(az)P(ab)P(zb)$	$\sum_{kcd}\langle kb\vert\vert cd\rangle t_k^a t_{ijw}^{cdz}$	
T_{6f}	-1	$P(ji)P(wi)$	$\sum_{ckl}\langle kl\vert\vert ci\rangle t_k^c t_{jwl}^{abz}$	
T_{6h}	$\frac{1}{2}$	$P(ij)P(iw)P(jw)$	$\sum_{ckl}\langle kl\vert\vert cw\rangle t_i^c t_{jkl}^{abz}$	
T_{6d}	-1	$P(ab)P(az)P(ji)P(wi)$	$\sum_{kcl}\langle kl\vert\vert ic\rangle t_k^a t_{jwl}^{cbz}$	
T_{5b}	$\frac{1}{2}$	$P(ij)P(iw)$	$\sum_{cdkl}\langle kl\vert\vert cd\rangle t_{ik}^{cd} t_{jwl}^{abz}$	
T_{5f}	$\frac{1}{4}$	$P(iw)P(jw)$	$\sum_{cdkl}\langle kl\vert\vert cd\rangle t_{ij}^{cd} t_{wkl}^{abz}$	
T_{5c}	$\frac{1}{2}$	$P(ab)P(az)$	$\sum_{ckld}\langle kl\vert\vert cd\rangle t_{kl}^{ca} t_{ijw}^{dbz}$	
T_{5a}		$P(ab)P(az)P(ij)P(iw)$	$\sum_{ckdl}\langle kl\vert\vert cd\rangle t_{ik}^{ca} t_{jwl}^{dbz}$	
T_{5d}	$\frac{1}{2}$	$P(ab)P(az)P(iw)P(jw)$	$\sum_{cdkl}\langle kl\vert\vert cd\rangle t_{ij}^{ca} t_{wkl}^{dbz}$	
T_{5g}	$\frac{1}{4}$	$P(az)P(bz)$	$\sum_{klcd}\langle kl\vert\vert cd\rangle t_{kl}^{ab} t_{ijw}^{cdz}$	
T_{5e}	$\frac{1}{2}$	$P(az)P(bz)P(ij)P(iw)$	$\sum_{kcdl}\langle kl\vert\vert cd\rangle t_{ik}^{ab} t_{jwl}^{cdz}$	
T_{7c}	$-\frac{1}{2}$	$P(ij)P(iw)P(jw)P(ba)P(za)$	$\sum_{cdk}\langle ak\vert\vert cd\rangle t_i^c t_j^d t_{wk}^{bz}$	
T_{7a}	-1	$P(ij)P(iw)P(bz)P(ba)P(za)$	$\sum_{ckd}\langle ak\vert\vert cd\rangle t_i^c t_k^b t_{jw}^{dz}$	
T_{7b}		$P(iw)P(ij)P(ab)P(az)P(wj)$	$\sum_{ckl}\langle kl\vert\vert cj\rangle t_i^c t_k^a t_{wl}^{bz}$	
T_{7d}	$\frac{1}{2}$	$P(ab)P(az)P(bz)P(ji)P(wi)$	$\sum_{klc}\langle kl\vert\vert ic\rangle t_k^a t_l^b t_{jw}^{cz}$	
T_{10b}	$-\frac{1}{2}$	$P(ij)P(iw)P(jw)P(ab)P(az)$	$\sum_{cdkl}\langle kl\vert\vert cd\rangle t_i^c t_j^d t_k^a t_{wl}^{bz}$	
T_{10a}	$-\frac{1}{2}$	$P(ij)P(iw)P(ab)P(az)P(bz)$	$\sum_{ckld}\langle kl\vert\vert cd\rangle t_i^c t_k^a t_l^b t_{jw}^{dz}$	

Table 6.10: Contributions to the CCSDT $\hat{T}_3$ amplitude equation (3)

T_{9a}		$P(ij)P(iw)$	$\sum_{ckdl}\langle kl\|\|cd\rangle t_k^c t_i^d t_{jwl}^{abz}$	
T_{9b}		$P(ab)P(az)$	$\sum_{ckld}\langle kl\|\|cd\rangle t_k^c t_l^a t_{ijw}^{dbz}$	
T_{9c}	$-\frac{1}{4}$	$P(ij)P(iw)P(jw)$	$\sum_{cdkl}\langle kl\|\|cd\rangle t_i^c t_j^d t_{wkl}^{abz}$	
T_{9e}		$P(ij)P(iw)P(ab)P(az)$	$\sum_{ckdl}\langle kl\|\|cd\rangle t_i^c t_k^a t_{jwl}^{dbz}$	
T_{9d}	$-\frac{1}{4}$	$P(ab)P(az)P(bz)$	$\sum_{klcd}\langle kl\|\|cd\rangle t_k^a t_l^b t_{ijw}^{cdz}$	
T_{8a}		$P(ab)P(az)P(iw)P(jw)$	$\sum_{ckdl}\langle kl\|\|cd\rangle t_k^c t_{ij}^{da} t_{wl}^{bz}$	
T_{8b}	-1	$P(ij)P(iw)P(ab)P(az)P(jw)$	$\sum_{cdkl}\langle kl\|\|cd\rangle t_i^c t_{jk}^{da} t_{wl}^{bz}$	
T_{8d}	$\frac{1}{2}$	$P(ij)P(iw)P(ab)P(az)$	$\sum_{cdkl}\langle kl\|\|cd\rangle t_i^c t_{jw}^{da} t_{kl}^{bz}$	
T_{8e}	$\frac{1}{2}$	$P(ab)P(az)P(iw)P(jw)$	$\sum_{kcdl}\langle kl\|\|cd\rangle t_k^a t_{ij}^{cd} t_{wl}^{bz}$	
T_{8c}	-1	$P(ab)P(az)P(bz)P(ij)P(iw)$	$\sum_{kcld}\langle kl\|\|cd\rangle t_k^a t_{il}^{cb} t_{jw}^{dz}$	

6.12 The CCDT equations

The main aim of this thesis is to explore triple contributions in coupled cluster theory for the homogeneous electron gas. For homogeneous matter such as the electron gas, the singles contribution vanish due to symmetry and momentum conservation [1, 36]. For this reason we will only need to include diagrams from the CCDT (doubles - triples) truncation. Furthermore, esome of the diagrams in the CCDT equations will vanish due to the fact that we employ a canonical Hartree-Fock basis [7, p.314]. We may therefore omit the term D_{10a} from the $\hat{T}_2$ amplitude equation, and all diagrams containing single excitations are omitted from the $\hat{T}_3$ amplitude equation (as compared to the full CCSDT $\hat{T}_3$ equation.)

TABLE 6.11: Contributions to the CCDT $\hat{T}_2$ amplitude equation.

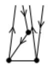
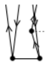
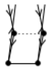
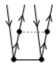
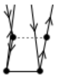
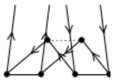
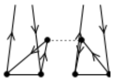
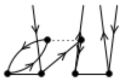
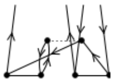
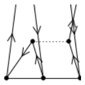
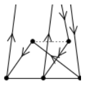
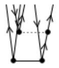
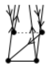
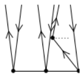
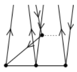
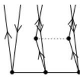
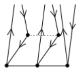
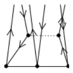
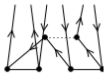
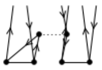
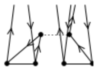
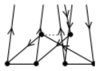
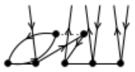
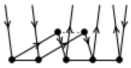
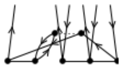
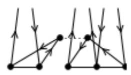
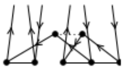
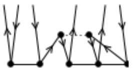
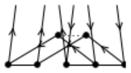
Name	Factor	Permutation	Interpretation	Diagram
	-1	$P(ba)$	$\sum_c f_{a,c} t_{ij}^{cb}$	
		$P(ij)$	$\sum_k f_{k,j} t_{ik}^{ab}$	
L_a	$\frac{1}{2}$		$\sum_{cd} \langle ab \vert\vert cd \rangle t_{ij}^{cd}$	
L_b	$\frac{1}{2}$		$\sum_{kl} \langle kl \vert\vert ij \rangle t_{kl}^{ab}$	
L_c	-1	$P(ba)P(ij)$	$\sum_{ck} \langle ak \vert\vert cj \rangle t_{ik}^{cb}$	
D_{3a}	$\frac{1}{4}$		$\sum_{cdkl} \langle kl \vert\vert cd \rangle t_{ij}^{cd} t_{kl}^{ab}$	
D_{3b}	$\frac{1}{2}$	$P(ab)P(ij)$	$\sum_{ckdl} \langle kl \vert\vert cd \rangle t_{ik}^{ca} t_{jl}^{db}$	
D_{3c}	$-\frac{1}{2}$	$P(ab)$	$\sum_{ckld} \langle kl \vert\vert cd \rangle t_{kl}^{ca} t_{ij}^{db}$	
D_{3d}	$-\frac{1}{2}$	$P(ij)$	$\sum_{cdkl} \langle kl \vert\vert cd \rangle t_{ik}^{cd} t_{jl}^{ab}$	
D_{10b}	$-\frac{1}{2}$	$P(ba)$	$\sum_{cdk} \langle ak \vert\vert cd \rangle t_{ijk}^{cdb}$	
D_{10c}	$\frac{1}{2}$	$P(ij)$	$\sum_{ckl} \langle kl \vert\vert cj \rangle t_{ikl}^{cab}$	

Table 6.12: Contributions to the CCDT $\hat{T}_3$ amplitude equation.

Name	Factor	Permutation	Interpretation	Diagram
T_{1a}	-1	$P(ba)P(bz)P(iw)P(jw)$	$\sum_c \langle ab\|\|ck\rangle t_{ij}^{cb}$	
T_{1b}		$P(az)P(bz)P(ij)P(iw)$	$\sum_k \langle kz\|\|j\rangle t_{ik}^{ab}$	
T_{2a}		$P(ba)P(za)$	$\sum_c f_{a,c} t_{ijw}^{cbz}$	
T_{2b}	-1	$P(iw)P(jw)$	$\sum_k f_{k,w} t_{ijk}^{abz}$	
T_{2c}	$\frac{1}{2}$	$P(za)P(zb)$	$\sum_{cd} \langle ab\|\|cd\rangle t_{ijw}^{cdz}$	
T_{2d}	$\frac{1}{2}$	$P(iw)P(ij)$	$\sum_{kl} \langle kl\|\|jw\rangle t_{ikl}^{abz}$	
T_{2e}	-1	$P(ba)P(za)P(iw)P(jw)$	$\sum_{ck} \langle ak\|\|cw\rangle t_{ijk}^{cbz}$	
T_{3d}	$\frac{1}{2}$	$P(iw)P(jw)P(ba)P(za)$	$\sum_{cdk} \langle ak\|\|cd\rangle t_{ij}^{cd} t_{wk}^{bz}$	
T_{3b}	-1	$P(bz)P(ba)P(ij)P(iw)P(za)$	$\sum_{ckd} \langle ak\|\|cd\rangle t_{ik}^{cb} t_{jw}^{dz}$	
T_{3c}		$P(ab)P(az)P(iw)P(ij)P(wj)$	$\sum_{ckl} \langle kl\|\|cj\rangle t_{ik}^{ca} t_{wl}^{bz}$	
T_{3e}	$-\frac{1}{2}$	$P(ab)P(az)P(iw)P(jw)$	$\sum_{ckl} \langle kl\|\|cw\rangle t_{ij}^{ca} t_{kl}^{bz}$	
T_{5b}	$\frac{1}{2}$	$P(ij)P(iw)$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ik}^{cd} t_{jwl}^{abz}$	
T_{5f}	$\frac{1}{4}$	$P(iw)P(jw)$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ij}^{cd} t_{wkl}^{abz}$	
T_{5c}	$\frac{1}{2}$	$P(ab)P(az)$	$\sum_{ckld} \langle kl\|\|cd\rangle t_{kl}^{ca} t_{ijw}^{dbz}$	
T_{5a}		$P(ab)P(az)P(ij)P(iw)$	$\sum_{ckdl} \langle kl\|\|cd\rangle t_{ik}^{ca} t_{jwl}^{dbz}$	
T_{5d}	$\frac{1}{2}$	$P(ab)P(az)P(iw)P(jw)$	$\sum_{cdkl} \langle kl\|\|cd\rangle t_{ij}^{ca} t_{wkl}^{dbz}$	
T_{5g}	$\frac{1}{4}$	$P(az)P(bz)$	$\sum_{klcd} \langle kl\|\|cd\rangle t_{kl}^{ab} t_{ijw}^{cdz}$	
T_{5e}	$\frac{1}{2}$	$P(az)P(bz)P(ij)P(iw)$	$\sum_{kcdl} \langle kl\|\|cd\rangle t_{ik}^{ab} t_{jwl}^{cdz}$	

6.13 CCDT-1

A common approach to include triple excitations is to include only the leading order terms T_{1a} and T_{1b} in the $\hat{T}_3$ amplitude equation, which then takes the form

$$\epsilon^{abc}_{ijk} t^{abc}_{ijk} = \hat{P}(a/bc|k/ij) \sum_d \langle bc||dk\rangle t^{ad}_{ij} - \hat{P}(c/ab|i/jk) \sum_l \langle lc||jk\rangle t^{ab}_{il} \tag{6.56}$$

We then proceed by using these amplitudes to calculate diagrams D_{10b} and D_{10c} in the $\hat{T}_2$ amplitudes.

This method is called the CCSDT-1 (in our case with no single excitations: CCDT-1) method [7], and scales as $\mathcal{O}(n_h^3 n_p^4)$ per iteration, where n_h is the number of hole states and n_p the number of particle states..

This is the actual approach we shall use in subsequent calculations. An alternative to this implementation is to first let the CCD amplitudes converge, whereby a perturbative third-order energy calculation is performed using the diagrams T_{1a} and T_{1b} above. This non-iterative approach is commonly referred to as the CCD(T) or *perturbative triples* approach. While CCSDT-1 normally yields better energy approximations [7, p.342], it is also known to overestimate the energy due to the fact that it only includes linear terms for the $\hat{T}_3$ amplitude. This problem is not present in the CCSD(T) since the method doesn't allow $\hat{T}_3$ amplitudes to affect the $\hat{T}_2$ amplitudes.

6.14 Intermediates

Because of the relatively great computational cost of the CC method, scientists have been searching for ways of reducing the number of floating point operations (flops) needed. On the one hand, one might attempt to rewrite the equations so that overlapping quantities are only computed once, and exploit symmetries in ways that reduce the computational cost, while on the other hand one might try out different schemes for the implementation that improves the efficiency of the code.

A popular approach to avoid computing overlapping quantities is the use of so-called *intermediates* [32], where we refactor the equations so that certain terms appear in multiple diagrams. The implications are that an intermediate calculation may be performed first, and thereafter multiple diagrams may be calculated by this intermediate. The use of intermediates may reduce the computational cost significantly [1].

As will become apparent in the upcoming chapters, the most computationally intense parts in the implementation of CCSD(T) for our purpose is not part of the CCD equation, but rather related to computation and storage of the $\hat{T}_3$ amplitudes. For this reason, intermediates were successfully implemented for educational purposes in one of the secondary solvers, rather than to reduce the computational cost in the main solver.

We shall just briefly state these intermediates for the CCD method in this section.

6.14.1 CCD intermediates

It is possible to rewrite the CCD equation as [1]

$$\begin{split}(f_{i,i} + f_{j,j} - f_{a,a} - f_{b,b})t_{ij}^{ab} = \langle ab||ij\rangle + \frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} + \\ \frac{1}{2}\sum_{kl} t_{kl}^{ab}\big(\langle kl||ij\rangle + \frac{1}{2}\sum_{cd}\langle kl||dc\rangle t_{ij}^{dc}\big) + P(ba)P(ij)\sum_{ck} t_{ik}^{ac}\big(\langle kb||cj\rangle + \frac{1}{2}\sum_{ld}\langle kl||cd\rangle t_{lj}^{db}\big) - \\ \frac{1}{2}P(ij)\sum_{k} t_{ik}^{ab}\big(\sum_{kcd}\langle kl||cd\rangle t_{jl}^{cd}\big) - \frac{1}{2}P(ab)\sum_{c} t_{ij}^{ac}\big(\sum_{kld}\langle kl||cd\rangle t_{kl}^{bd}\big). \quad (6.57)\end{split}$$

By then defining (and precomputing)

$$I_1 \equiv \langle kl||ij\rangle + \frac{1}{2}\sum_{cd}\langle kl||dc\rangle t_{ij}^{dc}, \quad (6.58)$$

$$I_2 \equiv \langle kb||cj\rangle + \frac{1}{2}\sum_{ld}\langle kl||cd\rangle t_{lj}^{db}, \quad (6.59)$$

$$I_3 \equiv \sum_{kcd}\langle kl||cd\rangle t_{jl}^{cd}, \quad (6.60)$$

$$I_4 \equiv \sum_{kld}\langle kl||cd\rangle t_{kl}^{bd}, \quad (6.61)$$

We may solve the CCD equation simply as

$$\begin{split}(f_{i,i}+f_{j,j}-f_{a,a}-f_{b,b})t_{ij}^{ab} = \langle ab||ij\rangle+\frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd}+\frac{1}{2}\sum_{kl} t_{kl}^{ab}I_1+P(ba)P(ij)\sum_{ck} t_{ik}^{ac}I_2- \\ \frac{1}{2}P(ij)\sum_{k} t_{ik}^{ab}I_3 - \frac{1}{2}P(ab)\sum_{c} t_{ij}^{ac}I_4. \quad (6.62)\end{split}$$

This provides a significant reduction of the computational cost, since some terms may even be reduced from an order of $\mathcal{O}(n_h^4 n_p^4)$ to $\mathcal{O}(n_h^4 n_p^2)$ [1]. These intermediates were successfully implemented in the upcoming sparse approach.

6.14.2 Intermediates for CCSD

Numerous schemes for factorization of diagrams into intermediates exists for the CCSD equation. Commonly used such factorizations are for example those of J. F. Stanton and J. Gauss [37], possibly referred to as simply the *Stanton-Gauss* intermediates, or the ones proposed by Scuseria, Scheiner, Lee, Rice and Schaefer [38].

In working on this thesis, a simple implementation of the Stanton-Gauss intermediates was written and tested on the H_2 molecule as well as the electron gas, and it was confirmed to exactly reproduce the results using no such intermediates. This implementation is found on the author's GitHub page [39].

Chapter 7

The Homogenous Electron Gas

7.1 The Homogenous Electron Gas

The homogenous electron gas (HEG) is a system where free electrons interact with each other and a uniformly distributed background charge [13]. The model is also known as the *Jellium Model* or the *Free Electron Gas* [13], and is currently a frequently studied system within many-body physics [1, 4, 6, 40].

Since the background charge is uniformly distributed, the model mainly focuses on effects due to interactions between the electrons. The model will in some sense be valid for systems where the electrons are weakly bound to the nuclei, such as periodic lattices with closed shells and weakly bound valence electrons [13].

A very similar treatment as we apply to the HEG system by expanding it in a plane wave basis will also be applicable to infinite, homogenous nuclear matter with minor alterations [1], and may provide insights into properties of supernova explosions [41] and neutron stars [42, 43].

Some of the earliest treatments of the HEG using CC were performed in the 1970's by Singal and Das [44], Freeman [45] and Bishop together with Luhrmann [46, 47].

7.2 The Hamiltonian

The Hamiltonian for the HEG is, see for example Ref. [13],

$$\hat{H} = \hat{H}_e + \hat{H}_{eb} + \hat{H}_{bb}, \tag{7.1}$$

where $\hat{H}_e$ relates to electron part only, $\hat{H}_{eb}$ is the interaction with the positive background and $\hat{H}_{bb}$ is the interaction between the background charges. This last term is analogous to the term that would be "frozen out" by the Born-Oppenheimer approximation if we were working with molecules or atoms, but in our case it will simply turn out to be a constant due to the uniformity of the density ρ [13], that is

$$\hat{H}_{bb} = \frac{e^2}{2}\int_\Omega d^3R \int_\Omega d^3R' \frac{\rho_i(\mathbf{R})\rho_i(\mathbf{R}')}{|\mathbf{R}-\mathbf{R}'|} = \hat{H}_{bb} = \frac{e^2}{2}\int_\Omega d^3R \int_\Omega d^3R' \frac{(\frac{\Omega}{N})^2}{|\mathbf{R}-\mathbf{R}'|}, \tag{7.2}$$

where e is the charge of the electron, ω is the volume and N is the number of charged ions in the background.

The Hamiltonian associated with the electrons consists of the kinetic energy of the electrons and their interactions is given as

$$\hat{H}_e = \sum_i \frac{\hat{p}_i}{2n} + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}. \tag{7.3}$$

The interaction between the electrons and the background charge may be written as

$$\hat{H}_{eb} = -\sum_i \int_\Omega d^3R \frac{N}{\Omega}\frac{e^2}{|\mathbf{r}_i - \mathbf{R}|}. \tag{7.4}$$

7.3 Ewald's summation technique

The electrons repel each other through the coulomb force, which has the general form

$$\frac{1}{2}\sum_{i\neq j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \tag{7.5}$$

where we have employed atomic units, that is $\hbar = c = e = 1$.

This expression is not convergent for an infinite number of particles, but in cases where the net charge of the system is neutral we may use *Ewald's summation technique* to make this energy convergent.

The error function is defined [48]

$$erf(x) \equiv \frac{2}{\sqrt{\pi}}\int_0^x dte^{-t^2}, \tag{7.6}$$

while the complementary error function is defined [48] as

$$erfc(x) \equiv 1 - erf(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty dte^{-t^2}. \tag{7.7}$$

Ewald [49] found that

$$\frac{1}{r} = \frac{erf(\frac{1}{2}\sqrt{\eta}r)}{r} + \frac{erfc(\frac{1}{2}\sqrt{\eta}r)}{r}, \tag{7.8}$$

which allows us to rewrite the electronic repulsion as

$$\frac{1}{2}\sum_{i\neq j}(\frac{erf(\frac{1}{2}\sqrt{\eta}r_{ij})}{r_{ij}} + \frac{erfc(\frac{1}{2}\sqrt{\eta}r_{ij})}{r_{ij}}), \tag{7.9}$$

where we have defined

$$r_{ij} \equiv |\mathbf{r}_i - \mathbf{r}_j|. \tag{7.10}$$

7.4 The Ewald interaction

It may be shown that the interaction between the electrons and the background, as well as interactions among the background charges vanish when using Ewald's summation technique [50], and we will end up with the interaction energy for the three dimensional HEG [51] as

$$v_E(\mathbf{r}) = \sum_{k\neq 0}\frac{4\pi}{L^3k^2}e^{i\mathbf{k}\bullet\mathbf{r}}e^{\frac{-\eta^2k^2}{4}} + \sum_{\mathbf{R}}\frac{1}{\mathbf{r}-\mathbf{R}}erfc(\frac{|\mathbf{r}-\mathbf{R}|}{\eta}) - \frac{\pi\eta^2}{L^3}, \tag{7.11}$$

where L is the length of one side in the simulation cell and the vector

$$\mathbf{R} = L(n_z\mathbf{u}_z + n_y\mathbf{u}_y + n_z\mathbf{u}_z), \tag{7.12}$$

is used to refer to all simulation cells in real space. The quantity $\mathbf{k}$ represents the momentum vector, while $\mathbf{r}$ is the position vector for each electron [1].

The parameter η allows us to gradually vary the amount of the different terms. We shall take it to be infinitesimally small and positive, as in [1, p.97], computing thereby the interaction in momentum space. This results in the following expression for the interaction

$$v_E(\mathbf{r}) = \sum_{k\neq 0}\frac{4\pi}{L^3k^2}e^{i\mathbf{k}\bullet\mathbf{r}}. \tag{7.13}$$

7.5 The antisymmetric matrix elements

The antisymmetric matrix elements for the three dimensional HEG are [1]

$$\langle pq||rs\rangle = \frac{4\pi}{L^3}\delta_{\mathbf{k}_p+\mathbf{k}_q,\mathbf{k}_r+\mathbf{k}_s}\big(\delta_{m_{s_p},m_{s_r}}\delta_{m_{s_q},m_{s_s}}(1-\delta_{k_p,k_r})\frac{1}{|\mathbf{k}_r-\mathbf{k}_p|^2} \\ - \delta_{m_{s_p},m_{s_s}}\delta_{m_{s_q},m_{s_r}}(1-\delta_{k_p,k_s})\frac{1}{|\mathbf{k}_s-\mathbf{k}_p|^2}\big). \quad (7.14)$$

The antisymmetric matrix elements may also be defined for the two dimensional case [1], but the focus of this thesis is on the three-dimensional electron gas.

7.6 The Hartree-Fock energy

For the electron gas the reference energy is [1]

$$E_{ref} = \sum_i \langle i|\hat{h}_0|i\rangle + \frac{1}{2}\sum_{ij}\langle ij||ij\rangle + \frac{1}{2}Av_0. \quad (7.15)$$

The number A is the number of particles, and the quantity v_0 is the so called Madelung constant. This term describes so-called *finite size* effects [1] that are stronger for small systems. As we increase the number of particle states towards the *thermodynamical limit*, it will vanish.

7.7 The Fock Matrix

The Fock matrix elements are [1]

$$\langle p|f|q\rangle = \frac{k_p^2}{2m}\delta_{\mathbf{k}_p\mathbf{k}_q}\delta_{m_{s_p}m_{s_q}} + \sum_i \langle pi||qi\rangle \quad (7.16)$$

7.8 The Wigner Seitz radius

We will not directly use the volume $\Omega = L^3$ in the implementation. We will instead follow the same procedure as [1, p.105], and calculate it using the dimensionless quantity r_s, implying that

$$\Omega(r_s) = \frac{4\pi}{3}r_B^3 r_s^3, \quad (7.17)$$

where

$$r_s = \frac{r_1}{r_B}, \tag{7.18}$$

and

$$\frac{\Omega}{N} = \frac{4\pi}{3} r_1^3, \tag{7.19}$$

where N is the number of electrons. The quantity r_s is called the *Wigner Seitz* radius, and is interpreted as an effective or mean distance between the electrons. The density increases as r_s becomes smaller, and $r_s \leq 1.5$ is sometimes referred to as *the high-density regime* (see for example Ref. [33]).

7.9 The plane wave basis

We will expand the system in a plane wave basis[1] for the finite volume $\Omega = L^3$, with single-particle basis functions defined as

$$\psi_{\mathbf{k}m_s}(\mathbf{r}) = \frac{1}{\sqrt{\Omega}} \exp{(i\mathbf{kr})}\xi_{m_s}. \tag{7.20}$$

The spin orientation is either up or down, and represented by ξ. Each such basis function has an associated single particle energy

$$\epsilon(x, y, z) = \frac{1}{2m}(\frac{2\pi}{L})^2(n_x^2 + n_y^2 + n_z^2). \tag{7.21}$$

The quantum numbers n_x, n_y and n_z allow us to define so-called magic numbers, as evidenced in table 7.1. The first three shells (resulting in 38 states) of the three dimensional plane wave basis are shown in table 7.1. We see that we have magic numbers corresponding to 2, 14, 38, 54, 66, 114, 162, 186 and so on.

[1] We assume the reader to be familiar with the notion of a plane wave basis as this forms the basis for the simplest possible quantum mechanical system taught in introductory quantum physics courses.

Table 7.1: The first three shells in the plane wave basis for the three-dimensional homogeneous electron gas

Shell	n_x	n_y	n_z	m_s
0	0	0	0	1
	0	0	0	-1
1	-1	0	0	1
	-1	0	0	-1
	0	-1	0	1
	0	-1	0	-1
	0	0	-1	1
	0	0	-1	-1
	0	0	1	1
	0	0	1	-1
	0	1	0	1
	0	1	0	-1
	1	0	0	1
	1	0	0	-1
2	-1	-1	0	1
	-1	-1	0	-1
	-1	0	-1	1
	-1	0	-1	-1
	-1	0	1	1
	-1	0	1	-1
	-1	1	0	1
	-1	1	0	-1
	0	-1	-1	1
	0	-1	-1	-1
	0	-1	1	1
	0	-1	1	-1
	0	1	-1	1
	0	1	-1	-1
	0	1	1	1
	0	1	1	-1
	1	-1	0	1
	1	-1	0	-1
	1	0	-1	1
	1	0	-1	-1
	1	0	1	1
	1	0	1	-1
	1	1	0	1
	1	1	0	-1

7.10 Recent progress on the Electron gas

A number of recent publications has sparked an interest in the homogeneous electron gas. Calculations using Coupled Cluster theory at the level of doubles excitations (CCD) have been performed by Roggero *et al.* [6] and Shepherd *et al.* [4], and to very high precision by Gustav Baardsen in his PhD thesis [1]. Shepherd *et al.* have frequently published on this topic since 2012, see for example Refs. [4, 33, 40, 52, 53], and in Ref. [40], the authors suggest a formal connection between the random phase approximation and the CCD approach.

Highly accurate calculations using Variational Monte Carlo (VMC) have also been performed as early as 1980 by Ceperley and Alder [54], but also in the recent years by Drummond *et al.* [55], Shepherd *et al.* [4] and Gurtubay *et al.* [56].

The system has also been a ”hot topic” in the Computational Physics group at the University of Oslo, and several master theses have touched upon this subject. From this group, Sarah Reimann has produced IM-SRG(2) results [57], while Karl Leikanger has produced FCIQMC [20] results for the tree-dimensional HEG. Similar results as those from Leikanger are available also from Shepherd *et al.* [4].

Comparisons between CCD and FCIQMC have shown that CCD fails to account for important correlations in the system [1].

Shepherd and Grüneis have also preformed CCD(T) calculations on the system [33], but they found that the perturbative treatment of the triples (using Möller-Plesset perturbation theory) resulted in divergent energies for the HEG. They propose in the same article a modification that lifts the divergent behavior.

To the present author’s knowledge, we have not been able to find any results beyond the CCD(T) results of Ref. [33]. This defines the rationale for this thesis, since a proper treatment of triples correlations are expected to have a non-negligible effect on the results for the ground state energy and the equation of state for the electron gas. The results to be presented here are thus the first ever studies of triples correlations for the homogeneous electron gas. Although we will focus on the three-dimensional electron gas, our codes can easily be applied to the two-dimensional electron gas. Furthermore, a proper assessment of such correlations has important consequences for studies of dense nuclear matter, expected to form the bulk matter of compact objects like neutron stars and proto-neutron stars. Since the theoretical description of infinite nuclear matter and/or neutron matter is similar to the electron gas, the developed formalism can be extended to such studies as well. Our codes are fully object oriented, allowing thereby for an easy extension to systems like dense nuclear matter.

Chapter 8

Implementation

8.1 Overview

This chapter deals with practical programming issues and conceptual perspectives on the implementation of the CCDT-1 equation for the system described in chapter 7.

As was clear from the derivation of the equations in chapter 6, the CCDT-1 consists of a CCD with an addition just four diagrams from the CCDT $\hat{T}_2$ and $\hat{T}_3$ equation. A natural stepping stone in the implementation is therefore the CCD equations, as they will also serve as means of benchmarking and validating the solver against existing results.

No CCM results beyond the CCD (with CCD(T) performed by Shepherd and Gruneis [53]) are known to the author at the time of writing this thesis, meaning that no direct benchmarking and validation of the CCDT-1 equations and the corresponding results is possible. To mend this shortcoming, we will develop two independent solvers for the CCDT-1 set of equations, allowing to corroborate our results. This approach will also be of great aid in the debugging process, as it will allow for extensive comparison and testing of subprocesses.

We will refer to the two solvers as the *block* implementation and the *sparse* implementation, and the reason for the naming will become clear in the upcoming discussion. While both solvers have similar aims, they are based on two conceptually different perspectives on the equations involved.

The sparse implementation approaches the problem in such a way the code very closely resembles the actual equations. One advantage of this is that it to an extent ensures the validity of the results, although bugs and misinterpretations are bound to occur. While memory usage of this approach is optimal, it has turned out that the actual iterative steps performs sub optimal.

The block implementation is built for the purpose of highly efficient computations and minimal memory usage. It is much more abstract in its comparison with the actual equations than the sparse solver, and it is to a large extent based on physical symmetries and restrictions present in the system.

Our strategy in this thesis is therefore to implement both solvers, whereby we validate both against the CCD results present in the literature (see Refs. [1, 4, 6, 33, 52, 53]). We then corroborate the CCDT-1 results by ensuring consistency between the solvers. We are then finally able to perform the calculations presented in the final chapter of this thesis.

8.1.1 The problem

While a lot of computational *results* are present in the literature, the author has not been able to find any descriptions of the actual algorithms used. In the literature, we find that corresponding CCD calculations for the HEG normally extrapolates results from basis sets of up to around 2000 particle states [6, 40], while some may even go as high as 20000 particle states [1].

Most descriptions of the actual algorithms for the coupled cluster method very closely resemble the code generated from CCAlgebra (see Ref. [58]) as presented in chapter 6. Such implementations may be called *naive*, as they have a one to one correspondence with the equations; each sum is translated into a nested for-loop, each diagram is added term by term, and at the bottom of each cluster of nested for-loops lies a couple of function calls and a multiplication. In the same way, we may refer to the storage of amplitudes and interactions as *naive*, if it has a one to one correspondence with the tensor representation in the equations. This would mean that we also store all elements equal to zero, neglecting any sparsity in the system.

Even with the inclusion of intermediates, such a naive approach to our system would be futile. The $\hat{T}_3$ amplitude is a tensor of rank 6, and if we were to naively store it for 14 particles using arrays it would scale roughly as shown in Fig. 8.1. This results in some serious limitations to the size of the basis, so we clearly need to move beyond any naive storage implementation.

Since the interaction that occurs in this system is defined by a product of Kronecker deltas over quantum numbers, we may expect most of the terms that occur in the diagrams to be zero. This fact actually reflects the conservation of momentum and spin in the system. The excitations due to the cluster operator should conserve spin and total momentum for the system. This is a strong indication that our system will be very sparse, and that alternative data structures to the naive one should be explored.

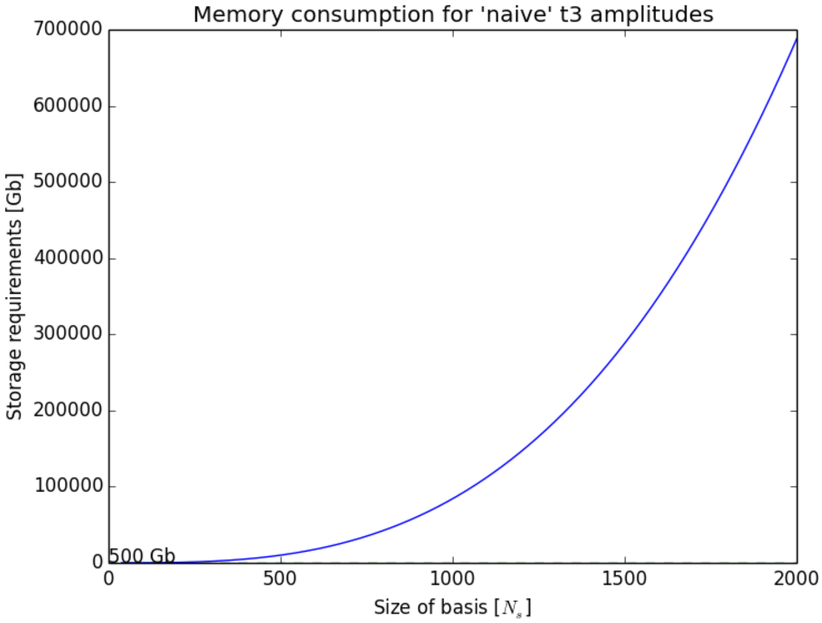


FIGURE 8.1: Size of the $\hat{T}_3$ amplitudes as a function of the number of particle states for a naive implementation, where the sparsity of the system is ignored.

To investigate the physical system properly with this method we will therefore need to write our code in a way that limits the use of memory and at the same time achieves optimal performance.

8.2 Contractions as matrix multiplications

We will need to generate diagrams, which are contractions between tensors of varying rank (at most 6). Common to both implementational approaches is that we shall conceptually consider diagrams as matrix multiplications. Bluntly speaking, and probably not in full accordance with linear algebra formalism, matrix multiplication *is already* a special case of contraction between two or more tensors of rank 2. Consider for example the matrix-matrix product

$$(MN)^{\alpha}_{\gamma} = \sum_{\beta} M^{\alpha}_{\beta} N^{\beta}_{\gamma}. \tag{8.1}$$

If we allow ourselves a more liberal approach, we may as well define

$$(MN)^{\alpha}_{\gamma} = \sum_{\beta} M^{\beta}_{\alpha} N^{\beta}_{\gamma} \rightarrow M^{T} N. \tag{8.2}$$

In 6, we interpreted contractions as sums over internal lines in the diagrams. The same may be done for the matrix multiplication above. We still need to define tensors of rank ≥ 2 as matrices for this to make full sense.

The motivation for introducing this concept, is twofold. Firstly it will - perhaps surprisingly - give a significant speedup of the solvers, while secondly it provides a suitable simplification of the actual coding process. A lot of nested for loops over varying indices may be subject to a number of mis-spellings/labelings or other kind of errors.

For optimal performance, we need to utilize BLAS (Basic Linear Algebra Subprograms) (see Ref. [59]) where possible. BLAS has three levels of operation depending on the order of the computational complexity. The optimization of operations is achieved by carefully utilizing the Level 1 and Level 2 cache, to reduce the cost assosciated with memory access [60]. The most efficient level of operation is BLAS Level 3, or operations of $\mathcal{O}(n^3)$, such as matrix-matrix multiplication [60]

To most efficiently perform all the floating point operations involved in this implementation, we therefore want to perform contractions as matrix-matrix multiplications. This basically means that we need to set up matrices for the interaction and the amplitudes, and align the elements in these matrices so that each resulting element contains the same sums as defined in the diagram.

8.2.1 Mapping diagrams onto matrices

In the same way that a Picasso painting preserves details beyond the line of sight, we will need to unambiguously map tensors of rank ≥ 2 onto matrices in such a ways that all elements are present and consistent amongst the matrices. One possibility for the interaction and the $\hat{T}_2$ amplitude, is

$$\langle pq|\hat{v}|rs\rangle = V_{\alpha(p,q),\beta(r,s)}. \tag{8.3}$$

Where

$$\alpha(p,q) = p + qN_p, \qquad \beta(r,s) = r + sN_r. \tag{8.4}$$

Note that the "steplength" of the second index equals the number of unique states in front of it. The above is easily extended to the $\hat{T}_3$ amplitude:

$$t_{ijk}^{abc} = t_{\alpha(a,b,c),\beta(i,j,k)}, \tag{8.5}$$

where

$$\alpha(a,b,c) = a + bN_a + cN_aN_b, \qquad \beta(i,j,k) = i + j * N_i + k * N_i * N_j. \tag{8.6}$$

The corresponding indices α, β will now correspond to the row and column in the matrices, respectively.

8.2.2 Subdividing the interaction matrix

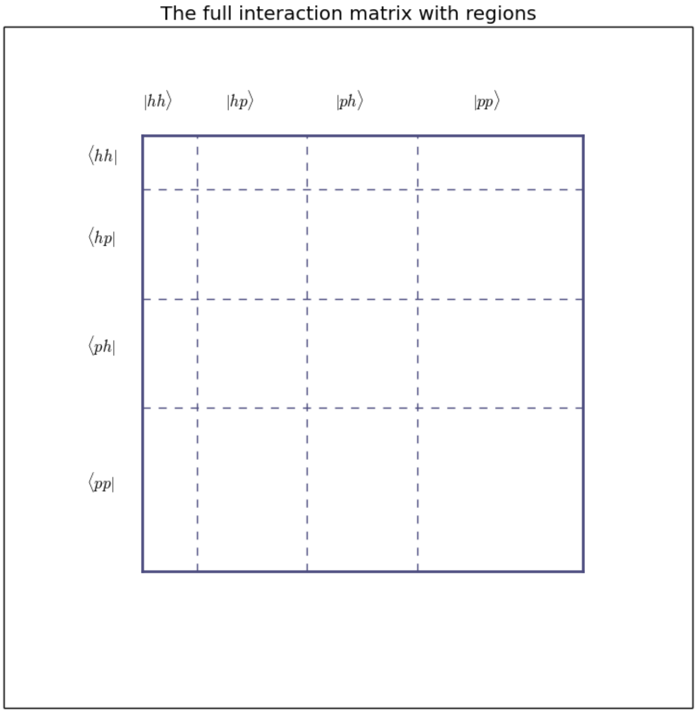


FIGURE 8.2: The interaction matrix with regions. The subdivision into regions is made so that each submatrix may be loaded when calculating the different diagrams. The difference in size is meant to illustrate that we will normally consider more particle states than hole states.

Diagrams naturally include terms summing over particle- or hole states. In no case do we sum over *all* the states (particles and holes) meanng that we generally find matrix elements of the form $\langle pp||pp\rangle$, $\langle pp||hh\rangle$, $\langle hp||ph\rangle$ and so on. There is a lot of symmetries that occur in the interaction matrix, so it will make sense to divide it into smaller regions. We will refer to these regions by their configuration in terms of particle- and hole states, as this is the way they occur in the diagrams. This subdivison is illustrated in 8.2. For example, when we consider diagrams where the interaction has the form $\langle ij||ab\rangle$, we will refer to the submatrix as the hh-pp matrix (hole-hole, particle-particle).

The off-diagonal submatrices in the interaction matrix have their transposed counterparts mirrored across the diagonal, and of course the usual symmetries due to the nature of spin in the two body interaction in Eq. (3.27) apply shown in table 8.1.

Table 8.1: Spin symmetries in the two body interaction matrix

$\langle p_\uparrow q_\uparrow \vert\vert r_\uparrow s_\uparrow\rangle$	$=$	$\langle pq\vert\hat{v}\vert rs\rangle - \langle pq\vert\hat{v}\vert sr\rangle$
$\langle p_\downarrow q_\downarrow \vert\vert r_\downarrow s_\downarrow\rangle$	$=$	$\langle pq\vert\hat{v}\vert rs\rangle - \langle pq\vert\hat{v}\vert sr\rangle$
$\langle p_\uparrow q_\downarrow \vert\vert r_\uparrow s_\downarrow\rangle$	$=$	$\langle pq\vert\hat{v}\vert rs\rangle$
$\langle p_\downarrow q_\uparrow \vert\vert r_\downarrow s_\uparrow\rangle$	$=$	$\langle pq\vert\hat{v}\vert rs\rangle$
$\langle p_\downarrow q_\uparrow \vert\vert r_\uparrow s_\downarrow\rangle$	$=$	$-\langle pq\vert\hat{v}\vert rs\rangle$
$\langle p_\uparrow q_\downarrow \vert\vert r_\downarrow s_\uparrow\rangle$	$=$	$-\langle pq\vert\hat{v}\vert rs\rangle$
$\langle p_\uparrow q_\uparrow \vert\vert r_\downarrow s_\downarrow\rangle$	$=$	0
$\langle p_\uparrow q_\uparrow \vert\vert r_\downarrow s_\uparrow\rangle$	$=$	0
$\langle p_\uparrow q_\uparrow \vert\vert r_\uparrow s_\downarrow\rangle$	$=$	0
$\langle p_\downarrow q_\downarrow \vert\vert r_\uparrow s_\uparrow\rangle$	$=$	0
$\langle p_\downarrow q_\downarrow \vert\vert r_\downarrow s_\uparrow\rangle$	$=$	0
$\langle p_\downarrow q_\downarrow \vert\vert r_\uparrow s_\downarrow\rangle$	$=$	0
$\langle p_\downarrow q_\uparrow \vert\vert r_\uparrow s_\uparrow\rangle$	$=$	0
$\langle p_\uparrow q_\downarrow \vert\vert r_\uparrow s_\uparrow\rangle$	$=$	0
$\langle p_\uparrow q_\downarrow \vert\vert r_\downarrow s_\downarrow\rangle$	$=$	0
$\langle p_\downarrow q_\uparrow \vert\vert r_\downarrow s_\downarrow\rangle$	$=$	0

8.2.3 Aligning matrices

In lack of a better word, we shall refer to the different ways in which higher dimensional tensors are mapped onto matrices as their *alignment*. Each element in the tensor will need to be assigned an index in the matrix, and these should be chosen such that the resulting matrix-matrix multiplications corresponds to the contractions specified in the diagrams.

Some diagrams align perfectly in the row and columns "out of the box". Consider for example the L_a "ladder" term [7] (see table 6.2), where

$$L_a = \sum_{cd} \langle ab|\hat{v}|cd\rangle t_{ij}^{cd}. \tag{8.7}$$

Here, we see that the columns in the interaction matrix aligns, since the mapping in Eq. (8.3) will result in

$$\langle ab|\hat{v}|cd\rangle \equiv v_{cd}^{ab} \to v_{\beta(c,d)}^{\alpha(a,b)}, \tag{8.8}$$

and

$$t_{ij}^{cd} \to t_{\gamma(i,j)}^{\beta(c,d)}. \tag{8.9}$$

By definition we have

$$\beta(c,d) = \beta(c,d), \tag{8.10}$$

implying that

$$(L_a)^{\alpha}_{\gamma} = \sum_{\beta} v^{\alpha}_{\beta} t^{\beta}_{\gamma} \to L_a = vt. \tag{8.11}$$

In other cases we are not that lucky. We may for example consider the L_c term, where

$$L_c = -P(ba)P(ij)\sum_{ck} \langle ak||cj\rangle t^{cb}_{ik}. \tag{8.12}$$

Just naively mapping onto the matrices will result in

$$\langle ak|\hat{v}|cj\rangle \equiv v^{ak}_{cj} \to v^{\alpha(a,k)}_{\beta(c,j)}, \tag{8.13}$$

$$t^{ik}_{cb} \to t^{\beta(i,k)}_{\gamma(c,b)}. \tag{8.14}$$

The matrix multiplication is "misaligned":

$$L_c \neq -P(ba)P(ij)v^{\alpha(a,k)}_{\beta(c,j)} t^{\beta(i,k)}_{\gamma(c,b)}. \tag{8.15}$$

This will not work, and doesn't even provide us with compatible matrix sizes that allow for such a multiplication to be performed. What we instead want, is to find a mapping so that

$$v^{ak}_{cj} \to \tilde{v}^{\alpha(a,j)}_{\beta(c,k)}, \tag{8.16}$$

$$t^{ik}_{cb} \to \tilde{t}^{\alpha(c,k)}_{\beta(b,i)}. \tag{8.17}$$

This will allow us to calculate

$$\tilde{L_c} = -P(ba)P(ij)\tilde{v}^{\alpha(a,j)}_{\beta(c,k)} \tilde{t}^{\alpha(c,k)}_{\beta(b,i)}. \tag{8.18}$$

This will provide us with an "unaligned" L_c denoted $\tilde{L_c}$ that may be "realigned" so it fits into the diagram summation in the $\hat{T}_2$ amplitude:

$$(\tilde{L_c})^{\alpha(a,j)}_{\beta(b,i)} \rightarrow (L_c)^{ab}_{ij}. \tag{8.19}$$

We will also have situations where contractions occur in more or less than two indices, so that our mapping cannot be directly applied. In these situations we introduce a general mapping rule, which states that a mapped index of elements $n_a..n_N$ in the order given by their numeric subindices and with the number of elements in each place $N_a, N_b...N_N$ may be expressed as

$$\alpha(a,b,c,...,N) \equiv abc..N = n_a + n_b N_a + n_c N_a N_b + n_d N_a N_b N_c + ... + n_N \prod_i^{N-1} N_i. \tag{8.20}$$

As long as this rule is consistently applied to each mapping of multiple particle or hole states onto rows or columns of the matrix, it will allow us to perform contractions over any number of states as matrix-matrix multiplications.

To demonstrate this, we consider finally the D_{3d} diagram 6.2, where

$$D_{3d} = -\frac{1}{2} P(ij) \sum_{cdkl} \langle kl||cd\rangle t^{cd}_{ik} t^{ab}_{jl}. \tag{8.21}$$

Here we have both a quadratic contribution from the $\hat{T}_2$ amplitudes and contractions of three lines between the interaction and the first, and only one contraction between the interaction and the last. We need to align the following matrices

$$v^{kl}_{cd} \rightarrow \tilde{v}^{l}_{kcd}, \tag{8.22}$$

$$t^{cd}_{ik} \rightarrow \tilde{t}^{kcd}_{i}, \tag{8.23}$$

$$t^{ab}_{jl} \rightarrow \tilde{t}^{jab}_{l}, \tag{8.24}$$

so we may compute the unaligned diagram

$$(\tilde{D}_{3d})^{jab}_{i} = \tilde{t}^{jab}_{l} \tilde{v}^{l}_{kcd} \tilde{t}^{kcd}_{i}, \tag{8.25}$$

and then realign the diagram so it will be possible to sum it into the equation

$$(\tilde{D}_{3d})_{i}^{jab} \rightarrow (D_{3d})_{ij}^{ab}. \tag{8.26}$$

8.2.4 Use of non standard terminology

By referring to this processes of *alignment* we have probably introduced some non standard terminology. There may exist corresponding and better expressions in use, but the author has not been able to find them in the literature due to the lack of implementational descriptions. From a theoretical point of view, it may be argued that the process of alignment is just a generalization of matrix transposition for tensors of any rank.We shall however use this terminology throughout this thesis with the reservation that better expressions may exist.

8.3 The sparse matrix approach

We have already mentioned the sparsity of our system, and we have explored how interactions and amplitudes may be represented by matrices. It is then only natural to consider whether a sparse matrix-matrix multiplicaton scheme may be beneficial and efficient for the system. At least, as we shall see, it is in principle simple to implement such a scheme and extend it to any order of the CC equations.

A matrix is generally considered sparse if most of its elements are zero [61, p.122]. As a rough estimate of the sparsity of our system, we may consider all interactions between particle states for a given number of basis states. These interactions is involved in the calculation of the L_a ladder term in the CCD approximation [7], and it is the most time consuming diagram in the calculation due to the number of states involved. By defining the density as the fraction of terms in the L_a sum that is nonzero, we see from Fig. 8.3 that the density seems to converge to somewhere below 0.01 percent as the number of particle states increases.

In such systems as ours, we will benefit from the implementation of a sparse matrix-matrix multiplication solver given that the arrangement of the elements in the matrix is completely random. As we shall see when considering the block implementation, this last part is not the case in our system.

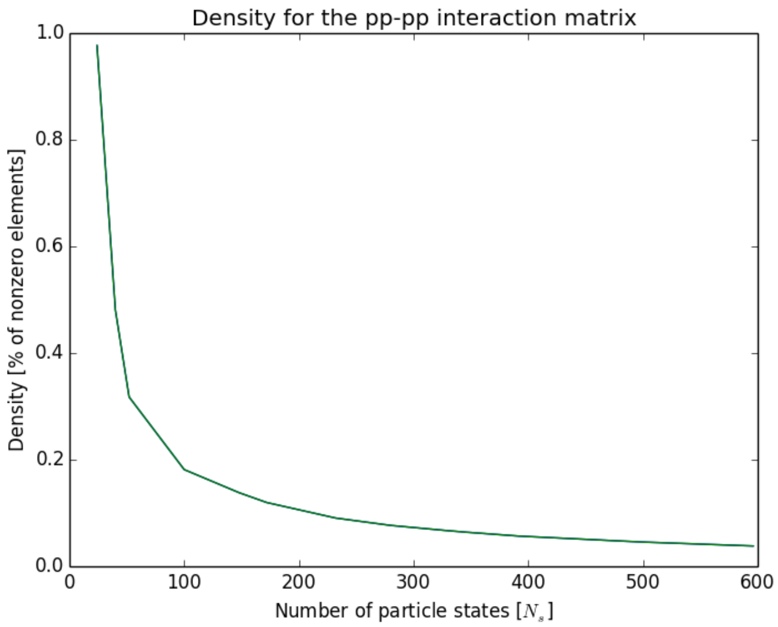


FIGURE 8.3: Density of the $\langle pp||pp\rangle$ interaction matrix. The density seems to converge towards somewhere below 0.1 percent, providing a reasonable argument for the use of sparse matrices.

8.4 Sparse matrix storage

Sparse matrices represent a commonly used data structure in computations where matrix elements are mostly zero. There are a number of different formats, such as the COOrdinate format, CRS and CCS [62], with well defined and efficient algorithms for matrix-vector or matrix-matrix multiplication.

The general idea is to store only three arrays; one containing the actual elements, and two others containing their corresponding indices in the full matrix. This is basically the COO format shown in table 8.2 with its dense interpretation in table 8.3. By sorting and compressing either the row or column arrays, one get the CRS or CCS formats respectively. These both reduce the spaces needed for storage somewhat, but also sorts the elements in such a way that matrix-matrix multiplications may be performed more efficiently. The backside is the fact that the sorting procedure may be relatively costly for large arrays.

TABLE 8.2: Sparse matrix storage (COOrdinate format)

Elements	a	b	c	d	e	f
row	1	1	2	3	5	5
column	0	3	1	2	1	3

Table 8.3: Sparse matrix storage (COOrdinate format) as dense matrix

0	0	0	0
a	0	0	0
0	c	0	0
b	0	d	0
0	0	0	0
0	e	0	f

8.5 Sparse tensor storage

Matrices are tensors of rank = 2. Generalizing the ideas used in the COO format, we may define a sparse tensor storage that works for tensors of any rank. This does really not involve anything else than having an optional number of element index arrays (rows and columns) available for the sparse tensor. For example, we may store the t_{ij}^{ab} tensor as in table 8.4.

8.6 Sparse matrix alignment

When considering how these sparse tensor elements are stored in conjunction with the alignment principles discussed previously, it is clear that the process of aligning sparse tensors will become remarkably simple. For example, we may perform the process in equation 8.23 simply by calculating

$$rows = k + cN_k + dN_kN_c = k + cN_h + dN_hN_p, \tag{8.27}$$

and

$$cols = i. \tag{8.28}$$

And then initialize a sparse matrix with the elements in the same order as they appear in the sparse tensor data structure, but with columns and rows as produced by the alignment process.

Table 8.4: Sparse tensor storage (COOrdinate format)

Elements	e_0	e_1	e_2	e_3	e_4	e_5
c	1	1	2	3	5	5
d	0	3	1	2	1	3
i	6	7	7	3	13	13
k	11	3	1	1	4	6

8.7 The sparse matrix implementation

The sparse matrix approach was implemented both in python and C++, but we will here only focus on the C++ implementation. To handle linear algebra operations, for vectors, matrices and sparse matrices, the linear algebra library Armadillo was extensively used (see Ref. [63]). Armadlillo also utilize all levels of BLAS, so it ensures optimal performance given a well written implementation. The implementation has no other dependencies aside from OpenMp and Armadillo.

The implementation is made in an object oriented manner, utilizing multiple classes for solvers, basis initialization, and tensor initialization. We will briefly discuss each class and how they fit in with the full picture.

8.7.1 Amplitude storage

Two different classes were implemented to store and align the $\hat{T}_2$ and $\hat{T}_3$ amplitudes, namely the *flexmat* and *flexmat6* classes respectively. The number 6 refers to the rank of the tensor.

These classes utilize the COO storage as specified for tensors of rank ≥ 2 (see table 8.4), and allow for a relatively efficient alignment of elements in the different diagrams. The inverse process of realigning incoming unaligned diagrams is also possible.

Although it is good coding practice to avoid repeating oneself (see Ref. [64]), a great deal of the functionality of this code was autogenerated with our python software, resulting in many similar functions. The motivation for this approach is twofold; it allows for a syntax that is very easily understood in terms of the different diagrams that occur in the CC equations, allowing for the actual implementation of the equations to be written very compactly and easily interpreted by humans. Secondly, this is an easy way to write highly specialized code for generating all possible alignments of elements needed for an implementation where this is not known in advance.

An alternative to this could be to write one single function that aligned the elements according to the parameters specified in the function call, but this implementation is

Table 8.5: Sparse tensor storage (COOrdinate format)

Expression	Flexmat function call
t_{ijk}^{abc}	t.pqr_stu()
$\tilde{t}_{abc}^{ijk}$	t.stu_pqr()
$t_{ij}^{ab}v_{kl}^{ij}$	t.pq_rs()*v.pq_rs()
$\tilde{t}_{bij}^{a}$	t.p_qrs()
$\tilde{t}_{bjc}^{aik}$	t.psu_qtr()

not as straightforward as the first one. Actually, large parts of the block implementation utilize such functionalities.

The flexmat syntax is aimed at high level usage for matrix contraction. Aside from initialization and various internal functions, the flexmat objects communicate with the solver class through two different operations. If we define any rank four tensor in its aligned form to be

$$t_{rs}^{pq}, \tag{8.29}$$

we will be able to call its equivalent flexmat sparse representation in C++ by

```
t.pq_rs();
```

The above function will return an Armadillo sparse matrix with the alignment specified by the order of letters and underscore. Some different alignments and their representation in previously used formalism in shown in table 8.5

We will also need to be able to realign elements by taking unaligned sparse matrices as parameters in the function calls. For each possible alignment, there is a function

```
update_as_pq_rs( ... );
```

This procedure unpacks the elements and indices from the incoming sparse matrix and makes them its own. For example we may again consider the misaligned diagram resulting from the D_{3d} matrix-matrix multiplication

$$(\tilde{D}_{3d})_{i}^{jab} \rightarrow (D_{3d})_{ij}^{ab}. \tag{8.30}$$

This would be properly aligned by calling

```
D3d.update_as_p_qrs(T.rpq_s()*vhhpp.q_prs()*T.spq_r(), Np, Nq, Nr, Ns);
```

The function call

```
D3d.pq_rs();
```

will now return the aligned tensor that may be added in with the other contributions to the $\hat{T}_2$ amplitude.

8.8 The sparse solver

The flexmat classes provide a very intuitive and simplistic environment in which to implement the actual equations. What remains is to translate each diagram into flexmat multiplications and realignments, so the full equation may be solved. In this section we tabulate each diagram, its flexmat representation and its realignment procedure if needed. Tables 8.6 and 8.8 list contributions to the $\hat{T}_2$ amplitudes, while tables 8.9, 8.10, 8.7 and 8.11 list diagrams occuring in the CCDT $\hat{T}_3$ amplitude equation.[1]

Table 8.6: Sparse alignment schemes for CCD equations

	Expression	Alignment	Interaction	t2(1)	t2(2)	Realignment
L_1	$\sum_{cd}\langle ab\|\|cd\rangle t_{ij}^{cd}$	$\langle ab\|\|cd\rangle t_{ij}^{cd}$				
L_2	$\sum_{kl}\langle kl\|\|ij\rangle t_{kl}^{ab}$	$t_{kl}^{ab}\langle kl\|\|ij\rangle$				
L_3	$\sum_{kc}\langle kb\|\|cj\rangle t_{ik}^{ac}$	$\langle jb\|\|ck\rangle t_{ai}^{ck}$	$v_{rs}^{pq} \rightarrow \tilde{v}_{rp}^{sq}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{pr}^{qs}$		$(L_3)_{rs}^{pq} \rightarrow (\tilde{L_3})_{pr}^{sq}$
Q_a	$\sum_{klcd}\langle kl\|\|cd\rangle t_{ij}^{cd}t_{kl}^{ab}$	$t_{kl}^{ab}\langle kl\|\|cd\rangle t_{ij}^{cd}$				
Q_b	$\sum_{klcd}\langle kl\|\|cd\rangle t_{ik}^{ac}t_{jl}^{bd}$	$t_{ck}^{ai}\langle kc\|\|ld\rangle t_{bj}^{ld}$	$v_{rs}^{pq} \rightarrow \tilde{v}_{qs}^{pr}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{sq}^{pr}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{ps}^{rq}$	$(Q_d)_{rs}^{pq} \leftarrow (\tilde{Q}_b)_{qs}^{pr}$
Q_c	$\sum_{klcd}\langle kl\|\|cd\rangle t_{ik}^{dc}t_{lj}^{ab}$	$t_{l}^{abj}\langle l\|\|kcd\rangle t_{i}^{kcd}$	$v_{rs}^{pq} \rightarrow \tilde{v}_{prs}^{q}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{r}^{pqs}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{r}^{sqp}$	$(Q_c)_{rs}^{pq} \leftarrow (\tilde{Q}_c)_{r}^{pqs}$
Q_d	$\sum_{klcd}\langle kl\|\|cd\rangle t_{lk}^{ac}t_{ij}^{db}$	$t_{klc}^{a}\langle klc\|\|d\rangle t_{bij}^{d}$	$v_{rs}^{pq} \rightarrow \tilde{v}_{s}^{pqr}$	$t_{rs}^{pq} \rightarrow \tilde{t}_{srq}^{p}$	$t_{rs}^{pq} \rightarrow \tilde{T}_{qrs}^{p}$	$(Q_d)_{rs}^{pq} \leftarrow (\tilde{Q}_d)_{qrs}^{p}$

[1] It should be noted that we have made a switch from the naming convention concerning the diagrams used in chapter 6. The reason for this inconvenience is that large parts of code were written before we adopted the naming convention used in Ref. [7] and chapter 6.

The flexmat syntax is shown in table 8.12, while a printout of the code for the actual CCD $\hat{T}_2$ amplitude equation is provided below.

```
void ccd::advance(){
    //advance the solution one step
    int Np = iSetup.iNp;
    int Nq = iSetup.iNp;
    int Nr = iSetup.iNh;
    int Ns = iSetup.iNh;

    L1_dense_multiplication();
    L2 = T.pq_rs()*vhhhh.pq_rs();

    fmL3.update(vhpph.sq_rp()*T.qs_pr(), Ns, Nq, Np, Nr);
    L3 = fmL3.rq_sp() - fmL3.qr_sp() -fmL3.rq_ps() +fmL3.qr_ps();

    fmQ1.update(T.rs_pq()*vhhpp.rs_pq()*T.rs_pq(), Nr, Ns, Np,Nq);
    Q1 = fmQ1.rs_pq();

    fmQ2.update(T.pr_qs()*vhhpp.rp_qs()*T.sq_pr(), Np, Nr, Nq, Ns);
    Q2 = fmQ2.pr_qs()-fmQ2.pr_sq(); //permuting elements

    fmQ3.update_as_r_pqs((T.r_sqp()*vhhpp.prs_q())*T.r_pqs(), Np, Nq, Nr, Ns);
    Q3 = fmQ3.pq_rs() - fmQ3.pq_sr(); //permuting elements

    fmQ4.update_as_p_qrs(T.p_srq()*vhhpp.pqr_s()*T.p_qrs(), Np, Nq, Nr, Ns);
    Q4 = fmQ4.pq_rs() - fmQ4.qp_rs(); //permuting elements



    Tprev.update(T.pq_rs(), Np,Nq,Nr,Ns);

    T.update(vpphh.pq_rs()+.5*(L1+L2)+L3+.25*Q1+Q2-.5*Q3-.5*Q4,Np,Nq,Nr,Ns);
    T.set_amplitudes(ebs.vHFEnergy); //energy denominator
    T.update(alpha*Tprev.pq_rs() + (1.0-alpha)*T.pq_rs(), Np, Nq,Nr,Ns);

    energy();
    T.shed_zeros();
    T.map_indices();
}
```

Table 8.7: Sparse alignment scheme for t_2 diagrams

Diagram	Factor	Permutation	Alignment
$(t_2)_a$	$+1$	$\hat{P}(k/ij\vert a/bc)$	$\sum_d \langle bc\vert\vert dk\rangle t_{ij}^{ad} \rightarrow \sum_d \langle bck\vert\vert d\rangle t_{aij}^{d}$
$(t_2)_b$	-1	$\hat{P}(i/jk\vert c/ab)$	$\sum_l \langle lc\vert\vert jk\rangle t_{il}^{ab} \rightarrow \sum_l t_l^{abi} \langle l\vert\vert cjk\rangle$

Table 8.8: Sparse alignment scheme for t_3 contribution to t2

Diagram	Factor	Permutation	Alignment
D_{10b}	$+\frac{1}{2}$	$\hat{P}(ab)$	$\sum_{mef} \langle bm\vert\vert ef\rangle t_{ijm}^{aef} \rightarrow (\sum_{mef} \langle b\vert\vert mef\rangle t_{ija}^{mef})_{ij}^{ab}$
D_{10c}	$-\frac{1}{2}$	$\hat{P}(ij)$	$\sum_{mne} \langle mn\vert\vert je\rangle t_{imn}^{abe} \rightarrow (\sum_{mne} t_{mne}^{abi} \langle mne\vert\vert j\rangle)_{ij}^{ab}$

Table 8.9: Sparse alignment scheme for $t_2 t_3$ diagrams

Diagram	Factor	Permutation	Alignment
$(t_2 t_3)_a$	$+1$	$\hat{P}(i/jk\vert a/bc)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{il}^{ad} t_{mjk}^{ebc} \rightarrow \sum_{me} \sum_{ld} t_{me}^{bjck} \langle me\vert\vert ld\rangle t_{ai}^{ld}$
$(t_2 t_3)_b$	$-\frac{1}{2}$	$\hat{P}(i/jk)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{li}^{de} t_{mjk}^{abc} \rightarrow \sum_{m} \sum_{lde} t_{m}^{abjck} \langle m\vert\vert lde\rangle t_{i}^{lde}$
$(t_2 t_3)_c$	$-\frac{1}{2}$	$\hat{P}(a/bc)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{lm}^{da} t_{ijk}^{ebc} \rightarrow \sum_{e} \sum_{lmd} t_{e}^{ibjck} \langle e\vert\vert lmd\rangle t_{a}^{lmd}$
$(t_2 t_3)_d$	$-\frac{1}{2}$	$\hat{P}(k/ij\vert a/bc)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{ij}^{ad} t_{lmk}^{bec} \rightarrow \sum_{lme} \sum_{d} t_{lme}^{bck} \langle lme\vert\vert d\rangle t_{aij}^{d}$
$(t_2 t_3)_e$	$-\frac{1}{2}$	$\hat{P}(i/jk\vert c/ab)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{il}^{ab} t_{jmk}^{dec} \rightarrow \sum_{mde} \sum_{l} t_{mde}^{jck} \langle mde\vert\vert l\rangle t_{abi}^{l}$
$(t_2 t_3)_f$	$+\frac{1}{4}$	$\hat{P}(k/ij)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{ij}^{de} t_{lmk}^{abc} \rightarrow \sum_{lm} \sum_{de} t_{lm}^{abck} \langle lm\vert\vert de\rangle t_{ij}^{de}$
$(t_2 t_3)_q$	$+\frac{1}{4}$	$\hat{P}(c/ab)$	$\sum_{ldme} \langle lm\vert\vert de\rangle t_{lm}^{ab} t_{ijk}^{dec} \rightarrow \sum_{de} \sum_{lm} t_{de}^{ijck} \langle de\vert\vert lm\rangle t_{ab}^{lm}$

Table 8.10: Sparse alignment scheme for $t_2 t_2$ diagrams

Diagram	Factor	Permutation	Alignment
$(t_2 t_2)_a$	-1	$\hat{P}(k/ij\vert a/bc)$	$\sum_{ld} \langle l\vert f\vert d\rangle t_{ij}^{ad} t_{lk}^{bc} \rightarrow \sum_{l} \sum_{d} t_{l}^{bck} \langle l\vert f\vert d\rangle t_{aij}^{d} = 0$
$(t_2 t_2)_b$	$+1$	$\hat{P}(i/jk\vert abc)$	$\sum_{lde} \langle lb\vert\vert de\rangle t_{il}^{ad} t_{jk}^{ec} \rightarrow \sum_{ld} \sum_{e} (t_{ld}^{ai} \langle ld\vert\vert be\rangle)_{e}^{aib} t_{cjk}^{e}$
$(t_2 t_2)_c$	$-\frac{1}{2}$	$\hat{P}(i/jk\vert c/ab)$	$\sum_{ldce} \langle lc\vert\vert de\rangle t_{il}^{ab} t_{jk}^{de} \rightarrow \sum_{de} \sum_{l} (t_{de}^{jk} \langle de\vert\vert lc\rangle)_{l}^{jkc} t_{abi}^{l}$
$(t_2 t_2)_d$	$+\frac{1}{2}$	$\hat{P}(k/ij\vert a/bc)$	$\sum_{ldmk} \langle lm\vert\vert dk\rangle t_{ij}^{ad} t_{lm}^{bc} \rightarrow \sum_{lm} \sum_{d} (t_{lm}^{bc} \langle lm\vert\vert dk\rangle)_{d}^{bck} t_{aij}^{d}$

Table 8.11: Sparse alignment scheme for t_3 diagrams

Diagram	Factor	Permutation	Alignment
$(t_3)_a$	$+\frac{1}{2}$	$\hat{P}(c/ab)$	$\sum_{de} \langle ab\vert\vert de\rangle t_{ijk}^{dec} \rightarrow \sum_{de} \langle ab\vert\vert de\rangle t_{cijk}^{de}$
$(t_3)_b$	$+\frac{1}{2}$	$\hat{P}(k/ij)$	$\sum_{lm} \langle lm\vert\vert ij\rangle t_{lmk}^{abc} \rightarrow \sum_{lm} t_{lm}^{abck} \langle lm\vert\vert ij\rangle$
$(t_3)_c$	$+1$	$\hat{P}(i/jk\vert a/bc)$	$\sum_{ld} \langle al\vert\vert id\rangle t_{ljk}^{dbc} \rightarrow \sum_{ld} \langle ai\vert\vert ld\rangle t_{bcjk}^{ld}$

Table 8.12: Sparse alignment schemes for CCDT equations

Diagram name	Code interpretation
L_a	update_as_pq_rs(vpppp.pq_rs()*t2.pq_rs())
L_b	update_as_pq_rs(t2.pq_rs()*vhhhh.pq_rs())
L_c	update_as_pq_rs(vhpph.sq_rp()*t2.qs_pr())
Q_a	update_as_pq_rs(T.rs_pq()*vhhpp.rs_pq()*T.rs_pq())
Q_b	update_as_pr_qs(T.pr_qs()*vhhpp.rp_qs()*T.sq_pr())
Q_c	update_as_r_pqs((T.r_sqp()*vhhpp.prs_q())*T.r_pqs())
Q_d	update_as_p_qrs(T.p_srq()*vhhpp.pqr_s()*T.p_qrs())
$(t_2t_3)_a$	update_as_qtru_ps(t3.qtru_sp() ∗ vhhpp.qs_pr() ∗ t2.sq_pr())
$(t_2t_3)_b$	update_as_pqtru_s(t3.pqtru_s()∗vhhpp.q_prs()∗t2.rpq_s())
$(t_2t_3)_c$	update_as_sqtru_p(t3.sqtru_p()∗vhhpp.s_pqr()∗t2.rsp_q())
$(t_2t_3)_d$	update_as_qru_pst(t3.pru_stq()∗vhhpp.pqs_r()∗t2.q_prs())
$(t_2t_3)_e$	update_as_tru_pqs(t3.sru_tpq()∗vhhpp.qrs_p()∗t2.s_pqr())
$(t_2t_3)_f$	update_as_pqru_st(t3.pqru_st()∗vhhpp.pq_rs()∗t2.pq_rs())
$(t_2t_3)_q$	update_as_stru_pq(t3.stru_pq()∗vhhpp.rs_pq()∗t2.rs_pq())
$(t_2t_2)_a$	(canonical HF basis) → no contribution
$(t_2t_2)_b$	update_as_psq_rtu(t2.pr_sq()∗vhppp.pr_qs()∗t2.p_qrs())
$(t_2t_2)_c$	update_as_tur_pqs(t2.rs_pq()∗vhppp.rs_pq()∗t2.s_pqr())
$(t_2t_2)_d$	update_as_qru_pst(t2.pq_rs()∗vhhph.pq_rs()∗t2.q_prs())
$(t_3)_a$	update_as_pq_rstu(vpppp.pq_rs() ∗ t3.pq_rstu())
$(t_3)_b$	update_as_pqru_st(t3.pqrs_tu()∗ vphhp.pq_rs())
$(t_3)_c$	update_as_ps_qrtu(vphhp.pr_qs() ∗ t3.sp_qrtu())
$(t_2)_a$	update_as_qru_pst(vppph.pqs_r() ∗ t2.q_prs())
$(t_2)_b$	update_as_pqs_rtu(t2.pqr_s() ∗ vhphh.p_qrs())
D_{10b}	update_as_q_rsp(vphpp.p_qrs() ∗ t3.uqr_stp())
D_{10c}	update_as_pqr_s(t3.pqs_tur() ∗ vhhhp.pqs_r())

8.9 A crossover scheme

While the sparse implementation is able to both reproduce CCD results from the literature and produce new CCDT-1 and full CCDT results not seen before, it has some strong limitations that restricts calculations to no more than approximately 186 basis states for the CCDT-1 and CCDT.

The sparse implementation is naive in the sense that no symmetries or physical restrictions is used to improve performance during the iterations. While it is fully possible to calculate the interaction matrix ”on the fly”, it would slow down performance significantly unless we were to somehow store blocks of the interaction matrix where nonzero elements occur. This naive approach actually makes the implementation use a lot more memory than what is strictly needed, and whats more each call to a specific alignment of the flexmat objects actually creates a new sparse matrix which needs both sorting and memory space.

The sparse matrix-matrix multiplication performance relies strongly on the elements being sorted, and such sorting schemes also require computational resources. And even the multiplication itself will not benefit from BLAS3, since it actually no longer is a matrix-matrix multiplication [60].

To reduce memory usage of this algorithm a crossover scheme into the upcoming block implementation was introduced. The main bottle-neck in at least the CCD case is the calculation of the L_a ladder diagrams. The reason for this is that it involves the pp-pp interaction matrix, which scales with as $\mathcal{O}(N_p^4)$ elements in the naive implementation. It is therefore a reasonable strategy to find a more efficient calculation of the ladder term.

Sparse matrices may easily be split into terms that when added up yield the original matrix. If we consider the COO format, we see that any subdivision of the element, column and row arrays will give us new and smaller arrays that may be cast onto sparse matrices with the same size as the original. Mathematically, we may consider the process

$$\hat{S} = \hat{s}_1 + \hat{s}_2 + \hat{s}_3 + ... \tag{8.31}$$

All these matrices have the same size, but the number of elements of the $\hat{s}_n$ added up is the same as the number of elements in $\hat{S}$. In the case of the ladder term, we may write it as

$$L_a = \hat{v}_{pppp}(\hat{t})_2 = (\hat{v}_1 + \hat{v}_2 + \hat{v}_3 + ...)((\hat{t_2})_1 + (\hat{t_2})_2 + (\hat{t_2})_3 + ...) = \sum_{\alpha\beta} \hat{v}_\alpha (\hat{t_2})_\beta . \tag{8.32}$$

If the elements in $\hat{v}_{pppp}$ appeared at completely random locations in the matrix, we would have to calculate all products and add them together. As previously mentioned, this is not the case in our system. We actually *do* have some valuable insight into the structure of the pp-pp interaction matrix, namely the conservation of spin and momentum. We will shortly discuss this in detail, but just to conclude our discussion of the sparse scheme we note that this insight will allow us to subdivide the matrices in such a way we end up with

$$L_a = \hat{v}_{pppp}(\hat{t})_2 = \sum_{\alpha\beta} \hat{v}_\alpha (\hat{t_2})_\beta \delta_{\alpha\beta} = \sum_{\alpha} = \hat{v}_\alpha (\hat{t_2})_\alpha . \tag{8.33}$$

This is implemented and working in the current version of the sparse implementation. It still does not improve performance or memory usage to such an extent that the scheme may produce results for larger basis sets, but it may function as a great motivation for the upcoming block implementation for the following reason: Equation (8.33) no longer describes sparse matrices.

By subdividing our matrices into regions with nonzero elements, we have reduced the full matrix-matrix multiplication of the large sparse matrices into a number of smaller dense matrix-matrix multiplications. Sparse matrix-matrix multiplication will outperform dense matrix-matrix multiplication only in the case where the density is below a certain threshold. In the case described in Eq. (8.33) this is no longer the case. Actually, by performing this multiplications with dense matrices, we will be able to benefit from BLAS3 and get a significant speedup (see Ref. [60]). It is this approach we will use in our block implementation.

8.10 The block implementation

While both implementations of the CCM discussed in this thesis is based on the concept of contractions as matrix multiplications, only the block implementation is able to fully benefit from this approach. Where the sparse implementation was written mainly to gain knowledge and function as verification, the block implementation is intended as a high performance solver, especially optimized for the HEG system. In addition to the aligned matrix approach, we shall take into account more of the symmetries and restrictions in our system. This will allow us to reduce basically all computations of diagrams into sums over multiplications of blocks within the full matrices, as described in Eq. (8.33).

It will also reduce the need for memory storage, as we for all diagrams simply will store the rows and columns present in each block, and calculate the actual elements when needed.

Regrettably there exists no implementational guide to these kinds of algorithms, at least not to the authors knowledge. For this reason, we will run into even more non-standard terminology, and a lot of the algorithmic approaches used in this implementation could most probably be greatly improved over time and in collaboration with others. However, as we shall see, the block implementation performs relatively well and utilizes memory in a reasonable way, making computations of significantly larger system than what we had for the sparse implementation possible.

8.11 Channels in the diagrams

As discussed above, most of the matrix elements will be zero due to the structure of the interaction. Most notably, we find that the first Kronecker delta appearing in the interaction enforces a conservation of total momentum, since

$$\delta_{\mathbf{k}_p+\mathbf{k}_q,\mathbf{k}_r+\mathbf{k}_s} \rightarrow \mathbf{k}_p + \mathbf{k}_q = \mathbf{k}_r + \mathbf{k}_s. \tag{8.34}$$

The interaction will also conserve the total spin. We need thus to include the conservation of spin when evaluating interaction elements, namely

$$m_{s_p} + m_{s_q} = m_{s_r} + m_{s_s}. \tag{8.35}$$

The amplitude will have the same requirements as the interaction, which may be seen upon initialization of the amplitude as corresponding to the second order many-body perturbation energy

$$(t_{ij}^{ab})_{t=0} = \frac{\langle ab||ij\rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \tag{8.36}$$

These conservation requirements have a direct impact on which amplitudes and interactions may occur. Let us again consider the L_a ladder diagram given by

$$L_a = \sum_{cd} \langle ab|\hat{v}|cd\rangle t_{ij}^{cd}. \tag{8.37}$$

We then impose the conservation requirements on L_a

$$\mathbf{k}_a + \mathbf{k}_b = \mathbf{k}_c + \mathbf{k}_d = \mathbf{k}_i + \mathbf{k}_j, \tag{8.38}$$

and

$$m_{s_a} + m_{s_b} = m_{s_c} + m_{s_d} = m_{s_i} + m_{s_j}. \tag{8.39}$$

The contracted indices occur in the middle of the expressions above since they are common to both tensors, showing that in effect each diagram should conserve both momentum and total spin for the system. We will for example have no contributions to the diagram where

$$\mathbf{k}_a + \mathbf{k}_b = \mathbf{k}_c + \mathbf{k}_d \neq \mathbf{k}_i + \mathbf{k}_j. \tag{8.40}$$

This means that only a subset of the internal line elements in each tensor needs to be contracted (summed over) when evaluating the diagrams. In principle, this is what gives us the simplification described in Eq. (8.33).

When we also take into account the concept of tensors as matrices, it is straightforward to see that these subsets of elements correspond to regions or *blocks* in the matrix representation of the tensor. z

We will find that for each diagram, the full matrix-matrix multiplication may now be reduced to a series of multiplications of the corresponding blocks in each tensor present in the diagram. We will refer to such corresponding blocks as a *channels*.

8.12 Mapping channels in the diagrams

Upon initialization, we will need to identify channels in each diagram and sort each tensor into corresponding blocks. To this end, it makes sense to define a unique quantity for each possible combination of quantum numbers in the states. One possible such unique identifier is

$$k_{unique}(\mathbf{k}, m_s) = k_x + k_y \Delta k + k_z \Delta k^2 + m_s \Delta k^3, \tag{8.41}$$

where we have defined a steplength

$$\Delta k \equiv 2max(\mathbf{k}) + 1, \tag{8.42}$$

to ensure that each combination of quantum numbers is assigned a unique value.

It should be clear that

$$k_{unique}(\mathbf{k}_p + \mathbf{k}_q, m_{s_p} + m_{s_q}) = k_{unique}(\mathbf{k}_p, m_{s_p}) + k_{unique}(\mathbf{k}_q, m_{s_q}). \tag{8.43}$$

Each index in the row or column of the matrix may now be assigned a unique identifier. Indices which share a common identifier belong to the same block, and we may therefore also associate this unique identifier with each block. All tensors in the diagram may now be reduced to such blocks, and the channels are defined by the intersection of unique identifiers from each of the tensors.

8.13 Channels and alignment of tensors

While the reasoning so far works fine for the "out of the box" aligned L_a ladder diagram, we run into some minor complications when we encounter differently aligned diagrams. It turns out, however, that the solution is simple. For example, we need to perform index transformations to the tensors in the D_{3d} diagram before it is possible to perform the matrix-matrix multiplication.

$$D_{3d} = -\frac{1}{2}P(ij)\sum_{cdkl}\langle kl||cd\rangle t_{ik}^{cd}t_{jl}^{ab} \rightarrow (\tilde{D}_{3d})_i^{jab} = -\frac{1}{2}P(ij)\tilde{t}_l^{jab}\tilde{v}_{kcd}^{l}\tilde{t}_i^{kcd}. \tag{8.44}$$

Our conservation requirements still apply, but need to be expressed differently to make sense in this case. We may just reorganize the terms to obtain

$$-\mathbf{k}_j + \mathbf{k}_a + \mathbf{k}_b = \mathbf{k}_l = -\mathbf{k}_k + \mathbf{k}_c + \mathbf{k}_d = -\mathbf{k}_i, \tag{8.45}$$

and the same for the spin:

$$-m_{s_j} + m_{s_a} + m_{s_b} = m_{s_l} = -m_{s_k} + m_{s_c} + m_{s_d} = -m_{s_i}. \tag{8.46}$$

This will then define another set of channels, corresponding to the aligned diagram D_{3d}. We may use the same procedure for any diagram present in our equations.

8.14 Implementing the channels

The number of channels in each diagram mainly depends on the number of orbitals and the alignment of the involved tensors. Mapping them by hand would be very tedious and inefficient, so we should let the program itself perform this mapping upon initialization of the system. With the introduction of the $\hat{T}_3$ amplitudes in our equations we have created yet another bottleneck in our code, since the number of rows in this matrix scales as $\mathcal{O}(Np^3)$.

Luckily for us, we will not need to evaluate every single row (and column) index in the tensors. On the downside, to avoid doing this we have to write a more complicated code than what we would in a "naive" approach.

To illustrate this approach, we may consider an unrealistic low number of particle states $N_p = 3$ (This number does not correspond to any of the "magic" numbers found in the 3D HEG system). The full row indices are shown (truncated) in table 8.13. In this case, we search for blocks defined by the unique quantity

$$k_{unique}(\mathbf{k}_a + \mathbf{k}_b + \mathbf{k}_c, m_{s_a} + m_{s_b} + m_{s_c}). \tag{8.47}$$

TABLE 8.13: The full row for $N_p = 3$

a	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	...	2
b	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	...	2
c	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	...	2

It is easy to see that we have six occurences per state of the unique quantity for any index where three different states $\mathbf{k}_a \neq \mathbf{k}_b \neq \mathbf{k}_c \neq \mathbf{k}_a$ (3 unequal indices) occur but in varying order:

$$\mathbf{k}_a+\mathbf{k}_b+\mathbf{k}_c = \mathbf{k}_b+\mathbf{k}_a+\mathbf{k}_c = \mathbf{k}_c+\mathbf{k}_b+\mathbf{k}_a = \mathbf{k}_a+\mathbf{k}_c+\mathbf{k}_b = \mathbf{k}_b+\mathbf{k}_c+\mathbf{k}_a = \mathbf{k}_c+\mathbf{k}_a+\mathbf{k}_b \tag{8.48}$$

In the case where we have only one inequality in the indices, for example $\mathbf{k}_a \neq \mathbf{k}_b = \mathbf{k}_c$, we find three rows per state corresponding to this configuration

$$\mathbf{k}_a + \mathbf{k}_b + \mathbf{k}_c = \mathbf{k}_b + \mathbf{k}_a + \mathbf{k}_c = \mathbf{k}_c + \mathbf{k}_b + \mathbf{k}_a = \tag{8.49}$$

And finally, the case where all indices are the same $\mathbf{k}_a = \mathbf{k}_b = \mathbf{k}_c$ correspond to only one row per particle state in our system. This is a very nice feature, since we may both use it to initialize the blocks and at the same time map the permutations that are applied in many of the diagrams.

It is important to note that although these states are constructed from the single "reduced" row (or column) in the tensor we are mapping, they still have their unambiguously defined placement in the matrix. This index is given as

$$row = a + bN_p + cN_p^2. \tag{8.50}$$

For consistency, we must ensure that the elements generated from the reduced row appear in the correct and increasing order. Failing to do so will make the element in the blocks misaligned, while the blocks remain correctly sized. It may therefore result in incorrect or even unstable results in the equations.

The practical implications for the initialization, is that we now need only initialize the single occurence of each combination of states whereby we simply express the associated states by permuting the indices in this state. The reduced row will scale as $\mathcal{O}(Np(Np+1)(Np+2)/6)$, and by comparison with table 8.13 it will look as in table 8.14, and we may derive all other quantities from this one.

TABLE 8.14: The reduced row for $N_p = 3$

a	0	1	1	1	2	2	2	2	2	2
b	0	0	1	1	0	1	1	2	2	2
c	0	0	0	1	0	0	1	0	1	2

8.15 Permutations

The discussion in the previous section is very closely related to how permutations may be efficiently performed on our diagrams. The reason for this is that any permutation of two or more elements will only occur within the blocks, and the result of the permutation will rely on the relationship between the indices at each row- or column index. In Fig. 8.4 an illustration of what these permutations may look like when applied to matrices.

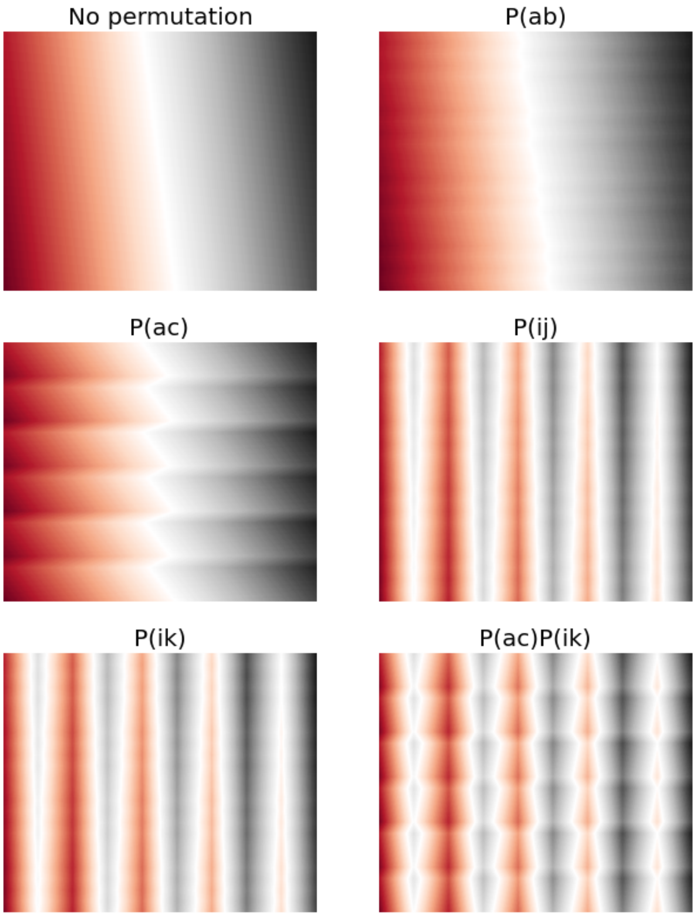


FIGURE 8.4: Permuting a continuous image as a $\hat{T}_3$ amplitude stored as block matrices.

TABLE 8.15: Permutation of lines

a	0	1	2	0	1	2		b	0	0	0	1	1	1
b	0	0	0	1	1	1	→	a	0	1	2	0	1	2
c	0	0	0	0	0	0		c	0	0	0	0	0	0

What the permutations actually do may be described in terms of the following pseudo algorithm for permuting elements a and b in a matrix representation of the $\hat{T}_3$ amplitude:

```
for a in 0 ... Np:
    for b in 0 ... Np:
        for c in 0 ... Np:
            for i in 0 ... Nh:
                for j in 0 ... Nh:
                    for k in 0 ... Nh:
                        P(ab)t3(a,b,c,i,j,k) = t3(b,a,c,i,j,k)
```

By considering table 8.13, we may view such a permutation as just the interchanging of line indices as in table 8.15, but we should then also be very careful about the ordering of the elements. By our definition for rows and columns, the table 8.15 is *unsorted*.

When comparing with our previous consideration on element initialization, we now see that these two processes are interlinked. It is therefore reasonable to map out permutations at the same time as we map the blocks in the "natural" alignment of the $\hat{T}_3$ tensor.

The process may be described as follows: we begin by setting up the reduced rows and columns for the tensor. We then scan through all unique combinations of total quantum numbers. We then check each row- and column element for inequalities. This testing may have basically three outcomes as shown in Fig. 8.5. For each possible outcome, we map the permutations accordingly.

One way of storing the permutations is to simply store an array with the permuted order of rows for each block. For example, we could store the permutation

$$a, b, c, d \to c, d, b, a, \tag{8.51}$$

simply as

$$2, 3, 1, 0. \tag{8.52}$$

Using Armadillo, we may then simply rearrange any rows in the matrix by calling

```
t2.rows(Pab(n));
```

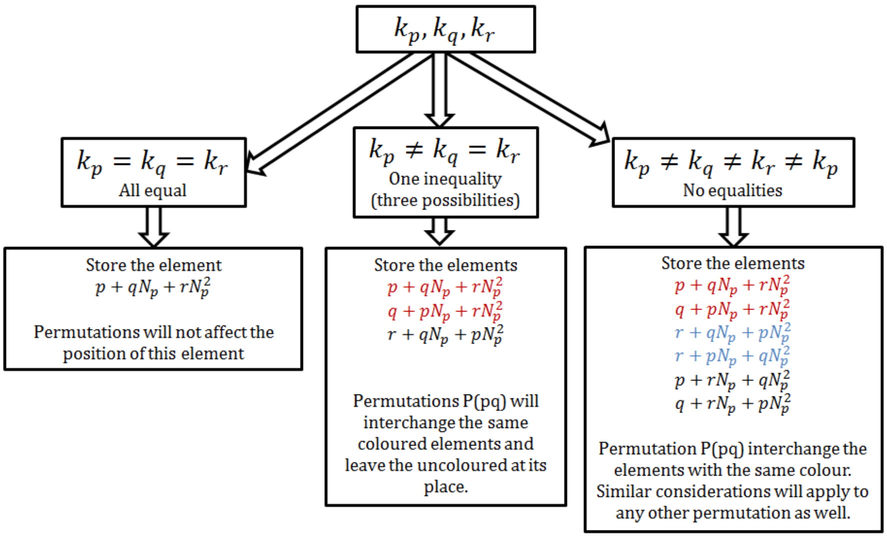


FIGURE 8.5: Flowchart for the permutation and block identification algorithm.

The equivalent command exists also for column ordering.

8.16 Element storage

As opposed to the sparse implementation, we now only need to store rows and columns for the blocks in the interaction. All matrix elements are then calculated on the fly, since our interaction is not that computationally intensive.

In the case of the amplitudes, we also need to store the actual elements. In this case we must also store the full blocks. While the $\hat{T}_3$ elements are stored in a large, one-dimensional contiguous array of doubles, the different alignments are stored as blocks where each element corresponds to a position in the $\hat{T}_3$ element array. This setup is illustrated in Fig. 8.6. A closeup of the blocks is shown in Fig. 8.7.

The procedure of generating these blocks may be outlined as follows: To make sure each alignment refers to the same elements in the $\hat{T}_3$ element array, we begin by storing a unique unsigned integer (unsigned long long) for each nonzero index from the full tensor matrix in a array associated with the $\hat{T}_3$ elements. These elements may be be assigned values as

$$t_{ijk}^{abc} = a + bN_p + cN_p^2 + iN_p^3 + jN_p^3N_h + kN_p^3N_h^2. \tag{8.53}$$

This number may unambiguously point us back to all the indices of the element in the full tensor matrix when needed.

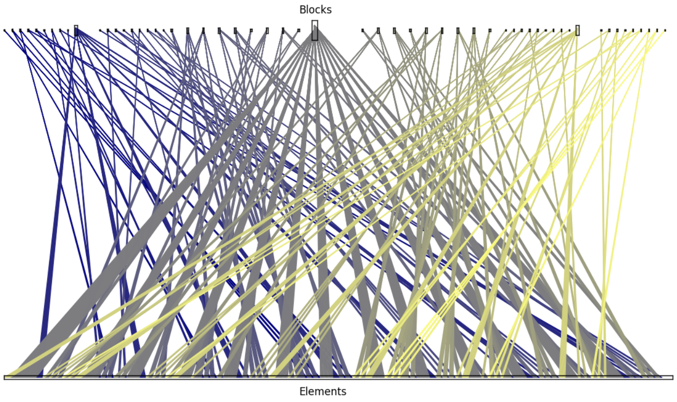


Figure 8.6: The relation between the contiguous array containing amplitudes and the partitioned blocks.

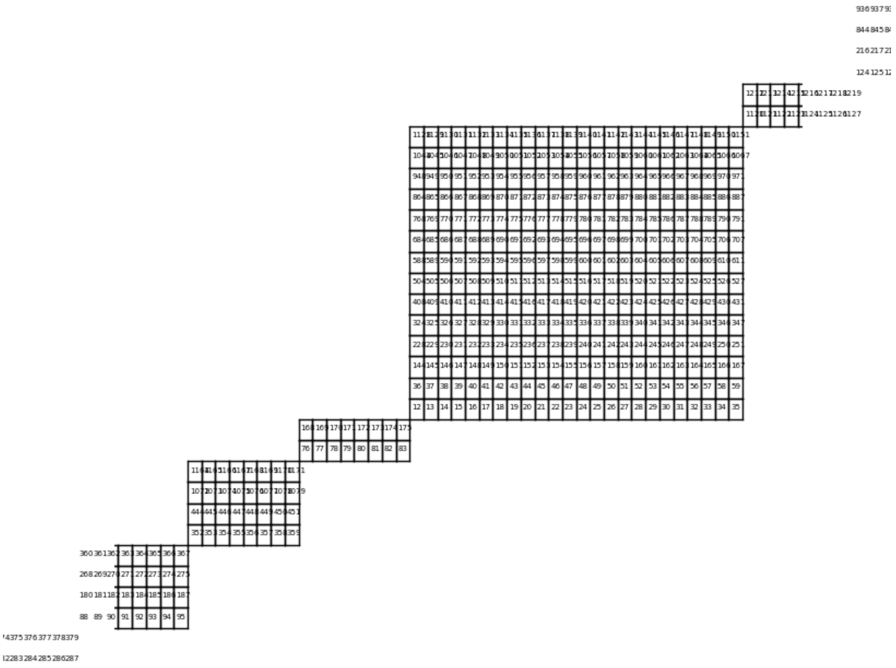


Figure 8.7: Closeup of some block matrices in the amplitude.

When all nonzero elements are found in the channels needed for the diagram, we consolidate the elements stored in this array with the existing elements present in the tensor. If none are previously initialized, we may simply start assigning the indices from the $\hat{T}_3$ element array to the elements in the blocks starting with element 0. If elements are already present, we must update our blocks to point to the correct address in the element array. New elements may be added to the end of the element array.

Elements already present are found simply by identifying equal elements in the two unique unsigned integer arrays.

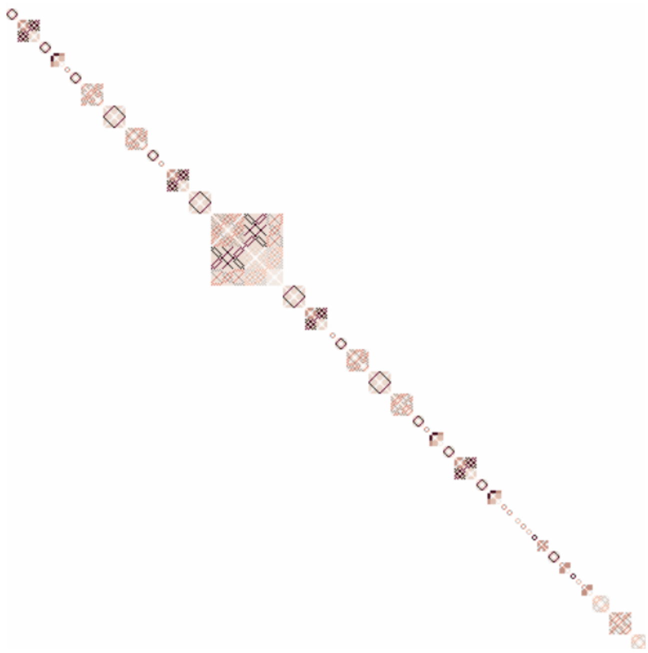


FIGURE 8.8: Closeup of some block matrices in the pppp interaction.

8.17 A sparse crossower scheme

Upon mapping the aligned blocks needed for the various CCDT-1 diagrams, we find that the number of channels appearing in each diagram is lower than the number of unique configurations in a diagram's constituent tensors. This makes it possible to reduce memory usage, since we for any alignment need only store those blocks that appears in this intersection.

All contributions to the $\hat{T}_3$ amplitude in the CCDT-1 method comes from contractions between the interaction and a linear $\hat{T}_2$ amplitude in diagrams T_{1a} and T_{1b}. When adding

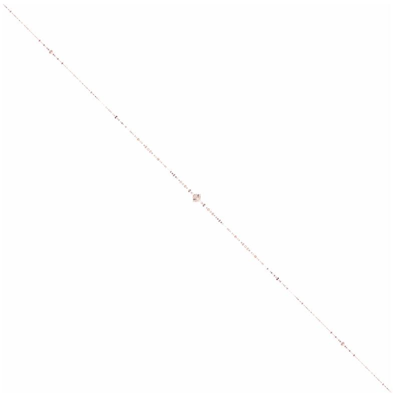


Figure 8.9: All blocks for the pppp interaction for 66 basis states.

in this contributions in the $\hat{T}_2$ equations through diagram D_{10b} and D_{10c}, we will find that the alignment used for the $\hat{T}_3$ amplitude in D_{10c} is actually the same as the one used to realign diagram T_{1b}.

The only time we actually need the "naturally" aligned blocks from t_{ijk}^{abc} is when we perform the permutations.

Although the conservation arguments that define our blocks ensure that a given alignment corresponds to certain channels in the diagrams, they do not guarantee that the "naturally" aligned elements will belong to the same block. In fact, we will find that when we first set up the alignments needed before mapping the "natural" alignment, all the blocks in the "natural" alignment will have nonzero elements.

This means that we must map and perform permutations of *all* the channels in the "natural" alignment. Such a process will consume a lot of memory and computing power to store and process a great deal of elements containing zeros. In the actual implementation, these difficulties are avoided by using a scheme inspired by parts of the sparse algorithm.

We first map the three alignments needed for the four different diagrams. When all these elements are consolidated with the element array in the tensor, we map the "natural" alignment needed for permutations in a similar way as before. The only two differences will be that when we encounter elements not already present in the $\hat{T}_3$ amplitude, we let them vanish, and when we encounter values present in the $\hat{T}_3$ tensor we store them in an unsorted COO format together with the number of rows and columns of the matrix. It is then very easy to reconstruct the dense matrix when needed, and it reduces the memory consumption significantly.

Permutations are stored and performed exactly as described in the previous section.

Chapter 9

Results

9.1 Validation of the CCD

The first step in the validation of the solvers is to ensure that results are consistent with those present in the literature. While most published results are plotted, we have been given extensive tabulated results from Baardsen [1, 5]. Roggero et al. also tabulates some CCD results in their article [6] that we may compare with our own. Comparisons between these results and the ones produced by our implementations is shown in table 9.1.

First we may note that the results are consistent within numerical precision for the sparse and block implementation, as one should expect. Minor deviations in the 15th digit may possibly be attributed to round-off errors or minor differences in the implementation (such as the sparse implementation actually not storing any zeros, where ”zeros” are defined to be values below a certain threshold).[1]

Next, we find that the results agree up to the 8th digit for $r_s = 1$ with the results from Baardsen [5].

The results for the various Wigner-Seitz radii are in perfect agreement with Ref. [1], while there is still deviations in the 3rd digit compared to Ref. [6].

When plotted, the author was not able to visually distinguish between the results from Refs. [4] and [5], and the ones produced by the sparse- and block implementations. The deviation between [5] and our results doesn't seem to have any systematic sign or pattern, so it could possibly be attributed to precision issues rather than accuracy issues. The latter would of course be a more serious issue.

[1] The author suspects that the cause for this is the ”.shed_zeros()” method in the flexmat objects, responsible for ensuring that round-off errors does not cause growth in the size of sparse storage vectors.

While the reason for this deviation is unclear, it should be noted that there are implementational differences between the two computations. The results from [5] were produced using Fortran and Intels MKL compiler, while our own is produced with C++. The author has been given the Fortran source code, but not been able to use it due to priorities concerning the C++ code and lack of experience using Fortran and Intel MKL.

Still, since there are very good agreement between [4], [5] and ourselves, we may conclude that our CCD results are valid within a tolerance needed to perform accurate calculations. They are in agreement with the literature, except when compared to those published by [6].

Table 9.1: CCD Validation

r_s	N_s	E_{CCD} [6]	E_{CCD} [1, 5]	E_{CCD} (block)	E_{CCD} (sparse)
1.0	54			-0.317822843688933	-0.317822843688934
1.0	66		-0.392696592145699	-0.3926965898061966	-0.3926965898061968
1.0	114		-0.447910594079769	-0.4479105961757175	-0.4479105961757175
1.0	162		-0.480557256148887	-0.4805572589306416	-0.4805572589306415
1.0	186		-0.485522928800019	-0.4855229317521318	-0.4855229317521321
1.0	246		-0.492924570774712	-0.4929245740023975	-0.4929245740023973
1.0	294		-0.498490911103782	-0.4984909094066818	-0.4984909094066817
1.0	342	-0.506701	-0.501952677927664	-0.5019526761547779	-0.5019526761547777
1.0	358		-0.502519675395361	-0.502519673607641	-0.50251967360764
0.5	114		-0.512015	-0.5120153541478306	-0.5120153541478306
0.5	342	-0.572682		-0.572964549890367	-0.5729645498903665
2.0	114		-0.357798	-0.3577968843144996	-0.3577968843144999
2.0	342	-0.417946		-0.4014136184665555	-0.4014136184665558

Validation of CCD implementations by comparing the correlation energy with results from G. Baardsen [1, table 5.2] and [5]. All results are given in Hartrees (atomic units). Variations in the Wigner Seitz radius is included for a more extensive validation.

9.1.1 The reference energy

Although the basis may already be assumed to function as it should since we have valid results for the CCD, it is possible to validate the reference energy as a stepping stone in the implementation.[2] From our implementation of the basis, we find for 14 electrons a reference energy of 1.94336533365203 Ry. per particle, whereas Baardsen [5]reports finding 1.9434 Ry. This is exactly the same results as expected.

[2] For anyone trying to reproduce the results from this thesis, this validation may serve as a first validation when implementing the code.

9.2 Validation of the CCDT-1

Since the author has not been able to find any results beyond the modified CCD(T) calculations from Shepherd et al. [33], the validation of the CCDT-1 will be somewhat circular in nature. Our two solvers will provide independent confirmation of each other given that they produce the same numbers.

A comparison of the CCDT-1 results are shown in table 9.2. As for the CCD results, we have alternating signs in the error and the deviation occur in the 16th digit indicating a round-off error. This is near perfect agreement. A comparison of the convergence in the iteration-wise energy also show highly corresponding behavior for the algorithms 9.3.

Table 9.2: Validation of the CCDT-1 results.

r_s	N_s	E_{CCDT-1} (block)	E_{CCDT-1} (sparse)	$\Delta\epsilon$
1.0	54	-0.3247616709272834	-0.3247616709272829	-4.99600361081e-16
1.0	66	-0.4014439489508850	-0.4014439489508858	8.32667268469e-16
1.0	114	-0.4642919485466862	-0.4642919485466857	-4.99600361081e-16
2.0	114	-0.3985520447482135	-0.3985520447482140	4.99600361081e-16
0.5	114	-0.5175412726087226	-0.5175412726087228	1.11022302463e-16

A comparison of the CCDT-1 energy results. All values given in Hartrees. The last column shows the difference.

It would be a stretch to say that our results is validated, but we may at least conclude that this corroborates our approach. Erroneous results could possibly arise simultaneously in both implementations due to theoretical misconceptions, or - highly unlikely - similar bugs in the two independent implementations. The author is therefore relatively confident that the CCDT-1 results are valid.[3]

[3] *Relatively* with respect to the CCD results, that is. While the discrepancy with Baardsens results are within our tolerance, one would expect such implementations to be more in agreement.

Table 9.3: Convergence comparison for the CCDT-1 implementations.

n	$E_{n,CCDT-1}$ (block)	$E_{n,CCDT-1}$ (sparse)	Ratio (block/sparse)
0	-0.5965687989792419	-0.5965687989792402	1.0000000000000029
1	-0.5297752822607126	-0.5297752822607129	0.9999999999999993
2	-0.5194857887853007	-0.5194857824022109	1.0000000122873234
3	-0.5178605713063522	-0.5178605700120056	1.0000000024994113
4	-0.5175960498600901	-0.5175960496868687	1.0000000003346654
5	-0.5175511889076089	-0.5175511888974993	1.0000000000195335
6	-0.5175431736069834	-0.5175431736109477	0.9999999999923402
7	-0.5175416559828336	-0.5175416559851100	0.9999999999956015
8	-0.5175413525580813	-0.5175413525589245	0.9999999999983707
9	-0.5175412894018753	-0.5175412894021506	0.9999999999994680
10	-0.5175412760304082	-0.5175412760304935	0.9999999999998352
11	-0.5175412732416370	-0.5175412732416628	0.9999999999999503
12	-0.5175412726966006	-0.5175412726966085	0.9999999999999846
13	-0.5175412726074502	-0.5175412726074521	0.9999999999999963
14	-0.5175412726006164	-0.5175412726006166	0.9999999999999998
15	-0.5175412726040832	-0.5175412726040834	0.9999999999999996
16	-0.5175412726066893	-0.5175412726066895	0.9999999999999996
17	-0.5175412726079237	-0.5175412726079236	1.0000000000000002
18	-0.5175412726084266	-0.5175412726084268	0.9999999999999998
19	-0.5175412726086172	-0.5175412726086169	1.0000000000000004
20	-0.5175412726086855	-0.5175412726086858	0.9999999999999996
21	-0.5175412726087104	-0.5175412726087101	1.0000000000000007
22	-0.5175412726087186	-0.5175412726087184	1.0000000000000004
23	-0.5175412726087215	-0.5175412726087211	1.0000000000000009
24	-0.5175412726087222	-0.5175412726087221	1.0000000000000002
25	-0.5175412726087227	-0.5175412726087223	1.0000000000000009
26	-0.5175412726087227	-0.5175412726087226	1.0000000000000002
27	-0.5175412726087227	-0.5175412726087227	1.0000000000000000
28	-0.5175412726087227	-0.5175412726087228	0.9999999999999998
29	-0.5175412726087227	-0.5175412726087228	0.9999999999999998
30	-0.5175412726087226	-0.5175412726087227	0.9999999999999998
31	-0.5175412726087226	-0.5175412726087228	0.9999999999999996
32	-0.5175412726087226	-0.5175412726087228	0.9999999999999996
33	-0.5175412726087226	-0.5175412726087228	0.9999999999999996
34	-0.5175412726087226	-0.5175412726087228	0.9999999999999996

An iteration-wise comparison of the block- and sparse methods show very similar behavior. The calculation is done for $r_s = 0.5$, $N_s = 114$ and relaxation parameter $\alpha = 0.3$. These results may also provide valuable debugging measures for anyone trying to reproduce the CCDT-1 results for the HEG presented in this thesis.

9.3 No validations beyond the CCDT-1

Since the CCDT-1 treats the $\hat{T}_3$ amplitudes as a linear function of the $\hat{T}_2$ amplitudes, it has a tendency of overestimating the energies. [7, p.320] We may try to compensate for this by including the quadratic $\hat{T}_2$ terms, often referred to as the CCSDT-2 approach [7, p.320], but with the optimized block implementation it is a very tedious process introducing new diagrams into the equations. In the syntax developed for the sparse approach this is a reasonably simple process, but at the same time the sparse implementation limits us to rather small basis sets.

The use of what Shepherd refers to as *judicious subsets of diagrams* [40] is commonly utilized in Many Body Perturbation Theory. Although the author may lack the*"good judgement"* when it comes to deciding which diagrams to include, we shall try the same approach. We will therefore systematically try to map how the different diagrams, or families of diagrams, contributes to the energy for a range of radii.

Because of the lack of validation of such calculations beyond the CCDT-1, these results should be used carefully. While the author has gone to great lengths to ensure the validity of these, there is no guarantee that all included diagrams behave as they should.

Still, this approach will allow us to arrive at the inclusion of all diagrams in the CCDT for the HEG system.

9.4 A comparison of methods

In table 9.4, we compare the CCDT-1 results with Baardsen's [1] CCD-results, Reimann's IM-SRG(2) results [57] and Leikanger's [20] and Shepherd's [4] FCIQMC results. This table is basically the same as in Ref. [1, p.104], but with our results for the CCDT-1 inserted in the middle.

For simplicity, and motivated by the fact that the FCIQMC results in principle should give the best possible fit for our basis, we may compare the results from the various methods by their ratio to the FCIQMC results from Ref. [4]. This is done in table 9.5.

We see that the CCDT-1 slightly overestimates the energy (as expected) by a factor of roughly $0.1 - 0.2\%$ for $r_s = 0.5$, roughly 0.7% for $r_s = 1.0$ and approx 3.7% in the $r_s = 2.0$ case . The CCD generally underestimates the energy by a factor of roughly 1% for $r_s = 0.5$, roughly 3% for $r_s = 1.0$ and approx. 7% for $r_s = 2.0$.

It is clear that the CCDT-1 accounts for more correlations than what we had in the CCD case.

Table 9.4: Method comparison.

r_s	n_s	E_{CCD}	E_{CCDT-1}	$E_{IM-SRG(2)}$	E^L_{FCIQMC}	E^S_{FCIQMC}
0.5	114	-0.073145	-0.0739344675155	-0.073638	-0.073842	-0.07384
0.5	186	-0.079047	-0.0799666850332	-0.079582	-0.07986	-0.07984
0.5	358	-0.081954	-0.0829461773725	-0.082504	-0.08284	-0.08281
1.0	114	-0.063987	-0.066327421221	-0.065290	-0.06583	-0.06587
1.0	186	-0.069360	-0.0720817173139	-0.070779	-0.07152	-0.07156
1.0	358	-0.071789	-0.0747195934729	-0.073235	-0.07413	-0.07412
2.0	114	-0.051114	-0.0569360063926	-0.053746	-0.05487	-0.05489

A comparison of the CCDT-1 results with various similar calculations for the HEG. We compare with Baardsen's [1] CCD-results, Reimann's IM-SRG(2) results [57] and Leikanger's [20] and Shepherd's [4] FCIQMC results respectively in the two last columns. All energies are given in Rydbergs.

Table 9.5: Ratio comparison of methods.

r_s	n_s	E_{CCD}	E_{CCDT-1}	$E_{IM-SRG(2)}$
0.5	114	0.990587757313	1.00127935422	0.997264355363
0.5	186	0.990067635271	1.00158673639	0.996768537074
0.5	358	0.989663084169	1.00164445565	0.996304794107
1.0	114	0.971413390011	1.00694430273	0.991194777592
1.0	186	0.969256567915	1.00729062764	0.98908608161
1.0	358	0.968550998381	1.0080894964	0.98805990286
2.0	114	0.931207870286	1.03727466556	0.979158316633

Results are given as ratios to the FCIQMC results from Ref. [4].

9.5 Comparisons beyond the CCDT-1

The results for the CCDT-1 calls for inclusion of more diagrams. For a fixed number of basis states $N_s = 114$, we therefore calculate the CCDT-2 by including the quadratic $\hat{T}_2$ terms, and finally the CCDT with all diagrams present. These results are shown in table 9.6, together with results presented so far in table 9.4.

Again, we are mainly interested in to which extent the methods accounts for correlations in the system, so we also calculate the ratios to the FCIQMC in Ref.[4]. These results are shown in table 9.7.

From these results it is obvious that for all values of r_s, the gradual inclusion of diagrams improves upon the approximation. For example in the case $r_s = 1$, the CCD is off by just below 3%, while the CCDT deviates from the FCIQMC by only about 0.3%. For all methods the ratios shows a tendency to increase with r_s, indicating a divergence with increasing r_s.

Plotting these values, as in Fig.9.1 shows that even the full CCDT tends to diverge from the FCIQMC results as we increase r_s.

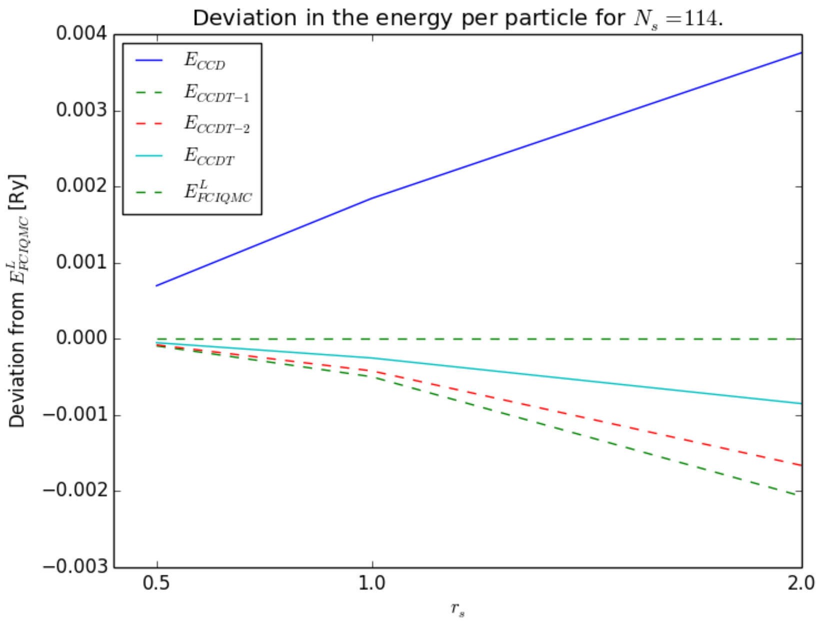


FIGURE 9.1: A comparison of the correlation energies per particle in Rydbergs from CCDT-1, CCDT-2 and CCDT calculations with the FCIQMC from Ref. [4].

TABLE 9.6: Beyond the CCDT-1 for $N_s = 114$.

r_s	.5	1.0	2.0
E_{CCD}	-0.073145	-0.063987	-0.051114
E_{CCDT-1}	-0.0739344675155	-0.066327421221	-0.0569360063926
E_{CCDT-2}	-0.07392170125916613	-0.06625038155163217	-0.05653233124188183
E_{CCDT}	-0.07389270791372174	-0.06608125094071378	-0.055720738752582745
$E_{IM-SRG(2)}$	-0.073638	-0.065290	-0.053746
E^L_{FCIQMC}	-0.073842	-0.06583	-0.05487
E^S_{FCIQMC}	-0.07384	-0.06587	-0.05489

Comparison of results beyond the CCDT-1. Still, we use Baardsen's [1] CCD-results, Reimann's IM-SRG(2) results [57] and Leikanger's [20] and Shepherd's [4] FCIQMC results respectively in the bottom two rows.

TABLE 9.7: Ratios beyond the CCDT-1 for $N_s = 114$.

r_s	.5	1.0	2.0
E_{CCD}	0.990587757313	0.971413390011	0.931207870286
E_{CCDT-1}	1.00127935422	1.00694430273	1.03727466556
E_{CCDT-2}	1.00110646342	1.00577473131	1.02992040885
E_{CCDT}	1.00071381248	1.00320708882	1.01513461018

The table shows ratios of the various CC approximations in discussion to the FCIQMC value from Ref. [4].

9.6 Thermodynamical limit estimates

In the thermodynamical limit the space spanned by the plane wave basis is almost complete, and our model space gives the best possible approximation of the system's true wave function. Comparing the estimates of our ground state energy at this limit with corresponding calculations from methods such as FCIVMC will therefore enable us to evaluate how well our method approximates the system's wave function.

In the literature (see for example Refs. [1, 4, 6, 33, 40]), such estimates are usually performed by first considering the ground state energy as a function of N_s^{-1}, and extrapolating the resulting function to the limit $N_s^{-1} = 0$.

This setup is shown in Fig. 9.3, where we have compared our results for the CCDT-1 with similar results from the literature.

9.6.1 Finite-size effects

As may be seen from the plot in Fig. 9.3, the line showing the CCDT-1 energy becomes increasingly smooth as the number of basis functions is increased. Certain irregularities are clearly visible for small basis sets. Such irregularities arise mainly since we have used an incomplete basis, and because we have a limited number of particles. These effects are commonly called *finite-size* effects and *incomplete basis error*, and they will introduce errors in the estimates. The finite size error falls off as $\mathcal{O}(N^{-1})$ for the error associated to the kinetic energy in the 3D electron gas. [65]

While Shepherd use a technique to very carefully treat the finite size effects (see Ref. [33]), we shall ignore these in our extrapolation. We will simply use a polynomial fitting technique, and estimate the ground state energy in the thermodynamical limit by gradually including more of the data points, starting with the smallest basis. This will allow us to perform an extrapolation, but the results should be treated cautiously since they obviously will contain errors that may be accounted for by some of the methods discussed in Ref. [65]. Because of the visible sloping of the line it seems reasonable to fit a second degree polynomial to the energy.

For each inclusion of new data points in our interpolation, the thermodynamic limit estimate will change, hopefully closer to the value we seek. In this context, it will make sense to plot the projected thermodynamical limit estimate as a function of the included data points so far. This setup is shown in Fig. 9.2. For each new point of data added in our set, starting with the rightmost calculation, we plot the estimated value - that is, the value of our interpolated polynomial in $N_s^{-1} = 0$.

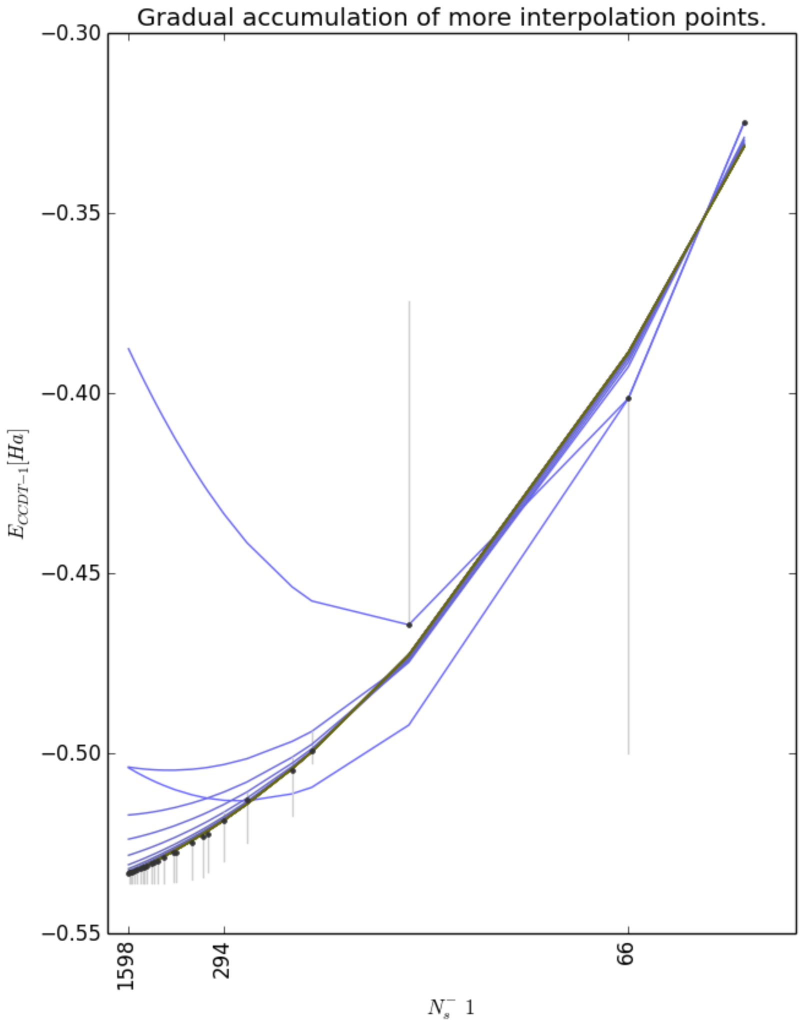


FIGURE 9.2: Gradual inclusion of more interpolation points (black dots) is assumed to bring us closer to the true thermodynamical limit estimate. The vertical lines indicates the estimated value for each interpolation point.

The extrapolate energy is also plotted in the same diagram, beginning with only the two leftmost data points resulting in the leftmost thermodynamical limit estimate. As we move to the right, we also accumulate more of the smaller basis sets, thereby also introducing more finite-size effects.

9.6.2 A note on obtaining the data

While our CCSDT-1 implementation easily handles smaller basis sets on a typical home computer, it quickly runs out of resources when we scale up towards the basis set sizes usually presented in the literature.

To obtain the data presented in Fig. 9.3, we therefore had to utilize the Abel Cluster (see Ref. [2]), which on some nodes offers up to one terabyte memory. These are the so-called *hugemem* nodes.

To perform these calculations as efficiently as possible, we have parallelized crucial parts of the code, such as the initialization of the $\hat{T}_3$ amplitudes, as well as the iterative procedure. This was achieved by the use of open MP (see Ref. [66]). We also eased the convergence criteria for these calculations, since the iterations for systems of this size is time consuming. Given time, it is fully possible to obtain results with the same precision as we had for the smaller basis sets.

9.6.3 Results from the large basis extrapolation

The results from these calculations is visualized in Fig. 9.3, while the actual energies are given in table 9.8. The number of basis functions in our data set allows for a meaningful comparison with CCD and FCIQMC results in the literature. The reason we have stopped at $N_s = 1598$ is not exhausted system resources on Abel, rather some of the vectors in our code use static allocation so at this point we actually had a segmentation fault.

In table II of Ref. [4], we find a comparable estimates from FCIQMC caclulations using a *single-point extrapolation* technique described in the same text. For $N_s = 14$ and up to 1850 orbitals, Shepherd *et al.* found a thermodynamical limit estimate of $-0.5316(4)$ Hartrees, where the number in the parenthesis represents the stochastic error. From table 9.9, we see that the CCDT-1 gives an estimate approximately at just above -0.536 Hartrees. By comparison, using the same technique on the CCD results from Ref. [5], we obtain an estimate just above -0.515 Hartrees. (By correcting for finite-size effects and incomplete basis error, Shepherd *et al.* obtained approximately -0.546 Hartrees from corresponding CCSD calculations in Ref. [33].)

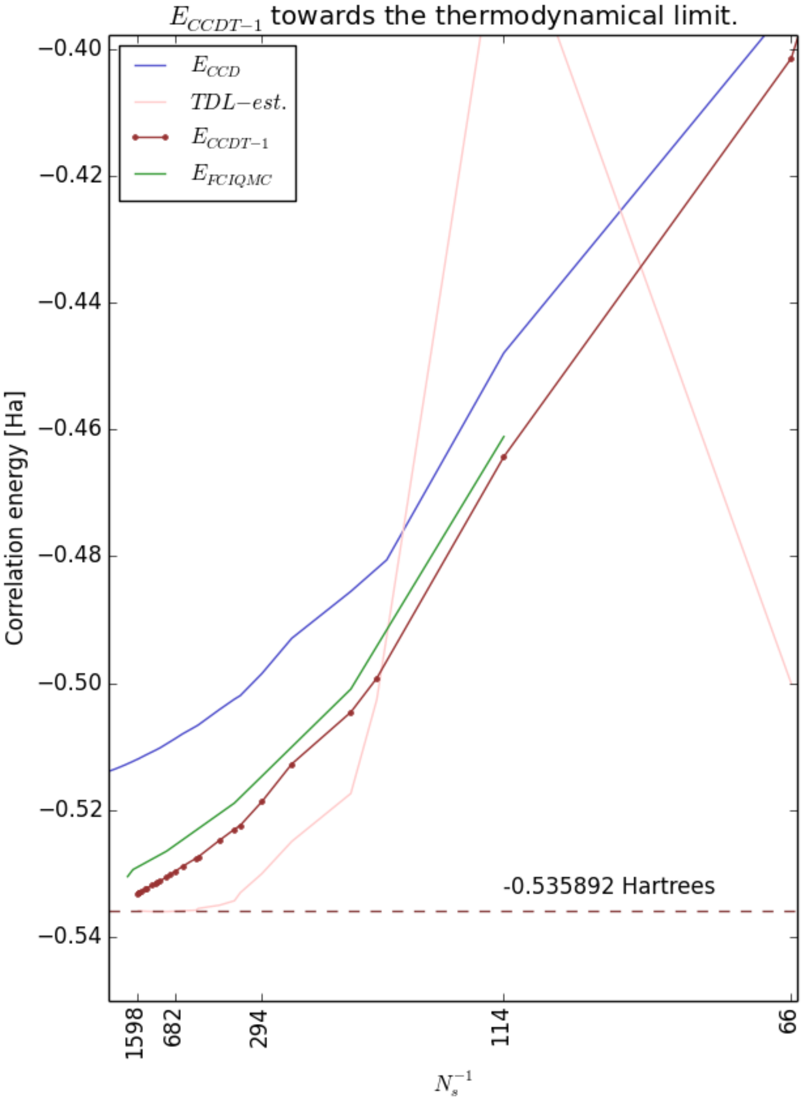


Figure 9.3: Our CCDT-1 results compared to FCIQMC from Ref. [4] and CCD from Ref. [5]. The thermodynamical limit results are extrapolated using a polynomial fitting in $(N_s^{-1})^2$, and the inclusion of all data points results in the dotted horizontal line at $E_{CCDT-1}(N_s \to \infty) \approx -0.53589205$ Hartrees.

In Ref. [33], they were able to perform calculations using CCD(T) for HEG systems of up to 200 particles. To overcome the computational costs, they approximated these contributions in a way that did not alter its qualitative behavior. This approximation is referred to as CCSD(scT). By using 700 to 1600 orbitals, they then estimated the ground state energy to be approximately −0.56 Hartrees[4] for 14 electrons at $r_s = 1.0$

[4] Number obtained by visual inspection of Fig. 4 in Ref. [33]

TABLE 9.8: Ground state energies from the CCDT-1 for $r_s = 1.0$

N_s	E_{CCDT-1}	$\Delta e_{f.i.}$
54	-0.3247616709272834	
66	-0.4014439489508850	
114	-0.4642919485466862	
168	-0.4992170228805105	
186	-0.5045720211973277	
246	-0.5127494951081482	
294	-0.5186916762333957	
342	-0.5224209034109932	
358	-0.5230371543102510	
406	-0.52477086	4.83611717e-09
502	-0.52738302	5.08323494e-09
514	-0.52759733	5.0938509e-09
610	-0.52890112	5.18983734e-09
682	-0.52973516	-9.0516229e-09
730	-0.53016462	5.27640609e-09
778	-0.53055221	5.29870681e-09
874	-0.53121663	5.33532052e-09
922	-0.53145774	5.34856293e-09
970	-0.53165927	5.35778177e-09
1030	-0.53186170	5.36616518e-09
1174	-0.53231755	5.38305334e-09
1238	-0.53249388	5.3889998e-09
1382	-0.53280698	5.3976934e-09
1478	-0.53300407	5.40233591e-09
1502	-0.533038	5.40305467e-09
1598	-0.53317370	5.40562695e-09

Results from computations on the so-called "hugemem" nodes on the Abel cluster in the lower part of the table is separated by a horizontal line from those obtained at an ordinary office computer. The precision is decreased for these calculations to reduce the computation time, but it given enough time (at most a couple of days) the precision may be easily improved. The rightmost column shows deviation between the two final iterations in the CCDT-1 and reflect our convergence criteria.

Table 9.9: Convergence in the thermodynamical limit estimate.

N_s	Estimated energy
114	-0.500074170875
168	-0.37448232
186	-0.50261060
246	-0.51739203
394	-0.52495653
342	-0.53004505
358	-0.53303659
406	-0.53430978
502	-0.53501392
514	-0.53551920
610	-0.53579986
682	-0.53590259
730	-0.53596398
778	-0.53599564
874	-0.53601389
922	-0.53602668
970	-0.53602951
1030	-0.53602389
1174	-0.53600828
1238	-0.53598359
1382	-0.53596086
1478	-0.53593480
1502	-0.53591225
1598	-0.53589205

Estimates of the thermodynamical limit ground state energy from CCDT-1 results on the homogeneous electron gas in 3D . We ignore finite-size effects and use the method visualized in Fig.9.2.

9.7 Contributions from diagrams

Introducing more diagrams in our calculations may have an effect on the energy. Some diagrams may add to the energy, others may subtract, and yet some may have no visible effect at all. Complicated couplings of the diagrams may make it hard to unambiguously decide how they play in,

A qualitative study of how each diagram contributes to the calculation may yield some insights into which kind of excitations are important in the system and which are not.

We begin by investigating how the gradual inclusion of the linear $\hat{T}_3$ diagrams as well as the two extra diagrams in the $\hat{T}_2$ amplitude due to the triple excitations affects the energy up to the CCDT-1 equations. The results of this gradual inclusion is shown for a range of r_s values and $N_s = 54$ in Fig. 9.4, while the deviation of these results with the corresponding CCDT calculation is shown in Fig. 9.5. Obviously, Fig.9.4 is not that informative, since contributions are tightly packed and hard to distinguish. Since we are only after the contributions from each diagrams, we may therefore omit such plots for the rest of the diagrams, as these will be even more tightly packed. The full numerical results are provided in tables 9.10 to 9.18.[5]

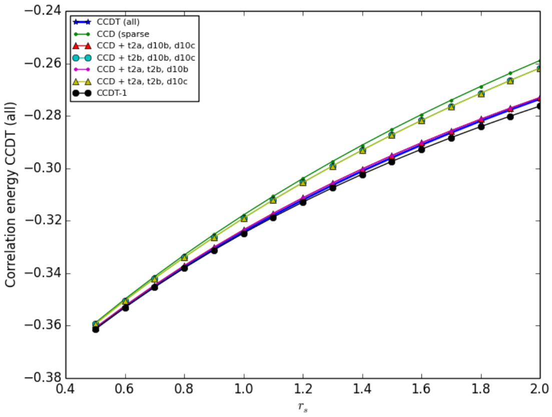


FIGURE 9.4: Correlation energies in Hartrees for various inclusions of diagrams leading up to the CCDT-1 equation.

In Fig. 9.7 we proceed by including even more diagrams in the CCDT1 equation, until we have the CCDT2 equations with contributions from all diagrams quadratic in $\hat{T}_2$. We then finally inspect individually contributions from each of the "mixed" $t2t3$ terms in Fig. **??**.

It is clear from all these results that most of the differences in the various approximations diverge from the CCDT correlation energy as we leave the high density regime where $r_s \leq 1.5$. One exception is the combinations containing both T_{2a} and D_{10b} in Fig. 9.5. The calculations with these contributions overlaps almost perfectly slightly above the CCDT calculation in a quadratic shape, and indicates that it will cross the CCDT correlation energy at some point. For $r_s = 2.0$, these calculations are even closer to the CCDT energy than the complete CCDT-1.

Generally, the deviations from the CCDT calculation decrease with roughly a factor of 10^{-1} between CCD, CCDT1. CCDT2 and CCDT, as may be seen by reading off the y-axes of the figures in question.

[5] These results may come in handy for those who wish to reproduce our results or in debug processes for similar codes. Please remember that these results have not been verified, so they may contain errors.

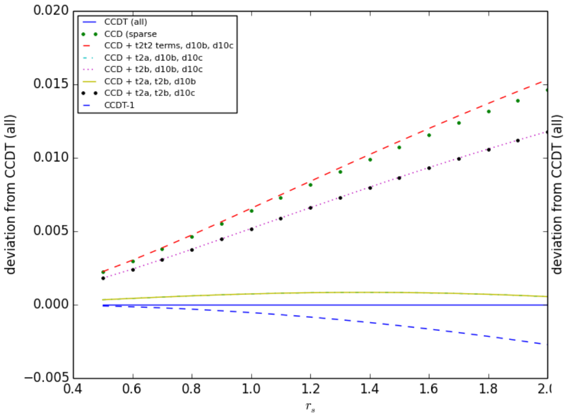


FIGURE 9.5: Deviations in Hartrees for the correlation energies from various truncations leading up to the CCDT-1 equation.

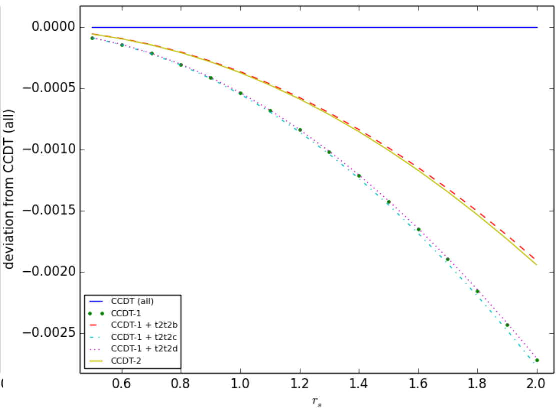


FIGURE 9.6: Deviations in Hartrees for the correlation energies from various truncations leading up to the CCDT-2 equation.

In the stage beginning with CCDT-1 and ending with CCDT-2 in Fig. 9.6, we may clearly see that the $(t2t2)_b$ term that adds in most of the CCDT-2 correlations.

From CCDT-2 to the inclusion of terms linear in $\hat{T}_3$, we see that all diagrams contributes significantly, but that $(t3)_a$ is responsible for most correlations, while $(t3)_b$ has the least impact.

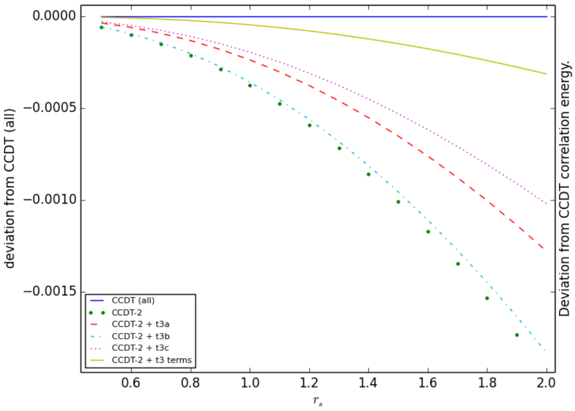


FIGURE 9.7: Deviations in Hartrees for the correlation energies from various truncations beyond the CCDT-2 equation.

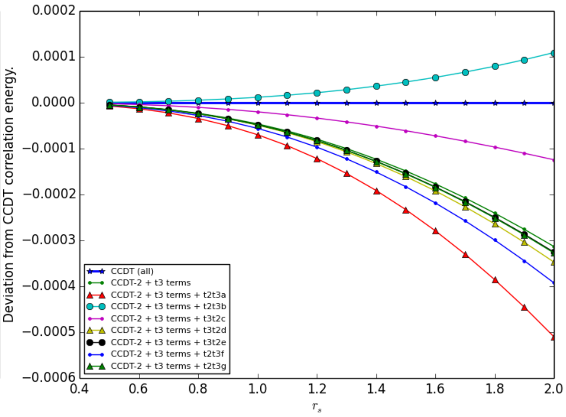


FIGURE 9.8: Deviations in Hartrees for the correlation energies from including "mixed" terms in the $\hat{T}_3$ amplitude leading up to the CCDT equation.

In the final stage, from CCDT-2 with linear $\hat{T}_3$ terms, ending with CCDT, it is clear that the diagrams with the largest contributions is $(t2t3)_b$ and $(t2t3)_c$, while these two are also the only ones increasing the energy towards the CCDT. Is is also notable that the diagram $(t2t3)_b$ also brings us above the CCSDT energy, and thus closer to the true correlation energy given that these results are comparable to the ones for $N_s = 114$.

The significance - if any - of these results are unclear to the author, but the clear improvement from including only the diagrams T_{2a} and d_{10b} does suggest more experiments with this truncation. Including only these diagrams could provide the means of reducing computational cost, as well as an improvement upon the CCDT-1 truncation outside the high density domain.

Table 9.10: From CCD to CCDT1 (a)

r_s	CCDT (all)	CCD (sparse)	CCD+all t2t2 , d_{10b}, d_{10c}	CCD+t_{2a}, d_{10b}, d_{10c}
0.5	-0.361182136242	-0.358965573995	-0.358937799509	-0.360848662897
0.6	-0.35289477153	-0.349913298941	-0.349869014363	-0.352467423986
0.7	-0.345095172951	-0.34129880764	-0.341233775603	-0.344578165852
0.8	-0.337741036665	-0.33309549971	-0.333005531117	-0.337141804947
0.9	-0.330794429481	-0.325278123609	-0.325159145375	-0.330122838461
1.0	-0.324221302444	-0.317822843689	-0.317670940833	-0.323489024517
1.1	-0.317991054686	-0.310707258113	-0.310518697541	-0.317211078584
1.2	-0.312076145029	-0.30391038201	-0.303681623395	-0.311262390387
1.3	-0.306451747765	-0.297412606757	-0.297140304484	-0.305618763253
1.4	-0.301095448541	-0.291195643538	-0.290876642595	-0.300258176233
1.5	-0.295986976242	-0.285242457229	-0.28487378513	-0.295160568459
1.6	-0.291107966848	-0.279537195014	-0.279116051241	-0.290307644542
1.7	-0.286441755584	-0.27406511291	-0.273588856927	-0.285682699556
1.8	-0.281973193938	-0.268812502441	-0.268278641007	-0.281270462005
1.9	-0.277688488513	-0.263766619014	-0.263172793299	-0.277056953128
2.0	-0.273575059003	-0.258915613005	-0.258259585841	-0.273029360975

Table 9.11: From CCD to CCDT1 (b)

r_s	CCD+t_{2b}, $d10b$, d_{10c}	CCD+t_{2a}, t_{2b}, d_{10b}	CCD+t_{2a}, t_{2b}, d_{10c}	CCDT-1
0.5	-0.359382812741	-0.360848610871	-0.35938286414	-0.361271280331
0.6	-0.350475868693	-0.352467326244	-0.350475964803	-0.353040097402
0.7	-0.342016839697	-0.344578001572	-0.342017000386	-0.345313192843
0.8	-0.333976230676	-0.337141550367	-0.333976478267	-0.338048918278
0.9	-0.326326455022	-0.330122467573	-0.326326813525	-0.331209710045
1.0	-0.319041801529	-0.323488509766	-0.319042295892	-0.324761670927
1.1	-0.312098369047	-0.317210391538	-0.312099024469	-0.318674187521
1.2	-0.30547398076	-0.31126150236	-0.305474822082	-0.312919583715
1.3	-0.299148086706	-0.30561764586	-0.2991491379	-0.307472809107
1.4	-0.29310166071	-0.30025680189	-0.293102944449	-0.302311160242
1.5	-0.287317096026	-0.295158910812	-0.287318633349	-0.29741403209
1.6	-0.281778102658	-0.290305678835	-0.281779912717	-0.292762696991
1.7	-0.276469608293	-0.285680402945	-0.276471708165	-0.288340108337
1.8	-0.27137766407	-0.281267813814	-0.27138006864	-0.284130726343
1.9	-0.266489355882	-0.277053935064	-0.266492077776	-0.280120363495
2.0	-0.261792721537	-0.273025957299	-0.261795771106	-0.276296047416

Table 9.12: From CCDT1 to CCDT2 (a)

r_s	CCDT(all)	CCDT-1	CCDT-1+$(t2t2)_b$
0.5	-0.361182136242	-0.361271280331	-0.361241081392
0.6	-0.35289477153	-0.353040097402	-0.352991511388
0.7	-0.345095172951	-0.345313192843	-0.345241188004
0.8	-0.337741036665	-0.338048918278	-0.33794837969
0.9	-0.330794429481	-0.331209710045	-0.331075519112
1.0	-0.324221302444	-0.324761670927	-0.324588760945
1.1	-0.317991054686	-0.318674187521	-0.318457581714
1.2	-0.312076145029	-0.312919583715	-0.312654420478
1.3	-0.306451747765	-0.307472809107	-0.307154357932
1.4	-0.301095448541	-0.302311160242	-0.301934830914
1.5	-0.295986976242	-0.29741403209	-0.296975379025
1.6	-0.291107966848	-0.292762696991	-0.292257420127
1.7	-0.286441755584	-0.288340108337	-0.287764051589
1.8	-0.281973193938	-0.284130726343	-0.283479874385
1.9	-0.277688488513	-0.280120363495	-0.279390837409
2.0	-0.273575059003	-0.276296047416	-0.275484099609

Table 9.13: From CCDT1 to CCDT2 (b)

r_s	CCDT-1+$(t2t2)_c$	CCDT-1 + $(t2t2)_d$	CCDT-2
0.5	-0.361272932544	-0.361270990289	-0.361242443
0.6	-0.353042798434	-0.353039597895	-0.352993711412
0.7	-0.345317255471	-0.345312404144	-0.345244458542
0.8	-0.338054668921	-0.338047749984	-0.337952955242
0.9	-0.33121748294	-0.331208062146	-0.331081631683
1.0	-0.324771803463	-0.324759434901	-0.324596636327
1.1	-0.318687016504	-0.318671247419	-0.318467436695
1.2	-0.312935442374	-0.31291581723	-0.312666460793
1.3	-0.307492024708	-0.30746808862	-0.307168776809
1.4	-0.302334052183	-0.302305353836	-0.301951808081
1.5	-0.297440910392	-0.29740700453	-0.296995080102
1.6	-0.292793861102	-0.29275431068	-0.292279996315
1.7	-0.2883758462	-0.288330224228	-0.2877896396
1.8	-0.28417131367	-0.284119204837	-0.283508596572
1.9	-0.280166063215	-0.2801070653	-0.279422802045
2.0	-0.276347109259	-0.276280834387	-0.275519401298

Table 9.14: From CCDT2 to CCDT2 + linear t3 (a)

r_s	CCDT	CCDT-2	CCDT-2 $+(t3)_a$
0.5	-0.361182136242	-0.361242443	-0.361218738719
0.6	-0.35289477153	-0.352993711412	-0.352955389336
0.7	-0.345095172951	-0.345244458542	-0.345187410081
0.8	-0.337741036665	-0.337952955242	-0.337872969247
0.9	-0.330794429481	-0.331081631683	-0.330974462634
1.0	-0.324221302444	-0.324596636327	-0.324458053866
1.1	-0.317991054686	-0.318467436695	-0.318293261114
1.2	-0.312076145029	-0.312666460793	-0.312452587945
1.3	-0.306451747765	-0.307168776809	-0.306911195085
1.4	-0.301095448541	-0.301951808081	-0.301646609441
1.5	-0.295986976242	-0.296995080102	-0.296638466677
1.6	-0.291107966848	-0.292279996315	-0.291868283709
1.7	-0.286441755584	-0.2877896396	-0.287319257769
1.8	-0.281973193938	-0.283508596572	-0.282976088915
1.9	-0.277688488513	-0.279422802045	-0.278824823213
2.0	-0.273575059003	-0.275519401298	-0.274852714074

Table 9.15: From CCDT2 to CCDT2 + linear t3 (b)

r_s	CCDT-2 + t3b	CCDT-2 + $(t3)_c$	CCDT-2 + t3(all)
0.5	-0.361240074436	-0.361212496588	-0.361186968791
0.6	-0.352989680206	-0.352945032659	-0.352903754956
0.7	-0.345238159145	-0.345171587896	-0.345110139593
0.8	-0.337943708706	-0.337850204441	-0.337764065954
0.9	-0.331068693447	-0.33094316437	-0.330827795998
1.0	-0.324579203596	-0.324416526415	-0.324267427027
1.1	-0.318444655364	-0.318239709672	-0.318052460081
1.2	-0.312637431899	-0.312385121921	-0.312155416584
1.3	-0.307132562511	-0.306827829204	-0.306551499031
1.4	-0.301907437187	-0.30154526299	-0.301218291375
1.5	-0.296941553166	-0.296516961116	-0.296135494817
1.6	-0.292216290327	-0.291724338872	-0.291284694974
1.7	-0.287714712303	-0.2871504868	-0.286649156719
1.8	-0.283421390437	-0.282779992098	-0.282213643357
1.9	-0.279322247926	-0.278598780801	-0.277964257128
2.0	-0.27540442181	-0.274593978214	-0.27388829842

Table 9.16: From CCDT2 + linear t3 to CCDT (a)

r_s	CCDT	CCDT-2 + t3(all)	CCDT-2 + t3(all) + $(t2t3)_a$
0.5	-0.361182136242	-0.361186968791	-0.361189161494
0.6	-0.35289477153	-0.352903754956	-0.352907932372
0.7	-0.345095172951	-0.345110139593	-0.345117267621
0.8	-0.337741036665	-0.337764065954	-0.337775292461
0.9	-0.330794429481	-0.330827795998	-0.33084443613
1.0	-0.324221302444	-0.324267427027	-0.324290947458
1.1	-0.317991054686	-0.318052460081	-0.318084463335
1.2	-0.312076145029	-0.312155416584	-0.31219762608
1.3	-0.306451747765	-0.306551499031	-0.306605745257
1.4	-0.301095448541	-0.301218291375	-0.301286499351
1.5	-0.295986976242	-0.296135494817	-0.296219672924
1.6	-0.291107966848	-0.291284694974	-0.291386925122
1.7	-0.286441755584	-0.286649156719	-0.286771585795
1.8	-0.281973193938	-0.282213643357	-0.282358475852
1.9	-0.277688488513	-0.277964257128	-0.278133748843
2.0	-0.273575059003	-0.27388829842	-0.27408475114

Table 9.17: From CCDT2 + linear t3 to CCDT (b)

r_s	CCDT-2 + t3(all)+$(t2t3)_b$	CCDT-2 + t3(all)+$(t3t2)_c$	CCDT-2 +t3(all)+$(t3t2)_d$
0.5	-0.361181173957	-0.361184340036	-0.361187127038
0.6	-0.352892878689	-0.352898818243	-0.352904091207
0.7	-0.345091851048	-0.345101835074	-0.345110770555
0.8	-0.337735673415	-0.337751170412	-0.337765146693
0.9	-0.330786303973	-0.330808950068	-0.330829522494
1.0	-0.324209592489	-0.324241162451	-0.324270037743
1.1	-0.317974845376	-0.318017225821	-0.318056236736
1.2	-0.312054437757	-0.312109603165	-0.312160684269
1.3	-0.306423469119	-0.306493460869	-0.306558625822
1.4	-0.30105945867	-0.301146367031	-0.301227687558
1.5	-0.295942076381	-0.296048025283	-0.296147611826
1.6	-0.291052905834	-0.291180040147	-0.291300024139
1.7	-0.286375235231	-0.286525710259	-0.286668227882
1.8	-0.281893873224	-0.282069846146	-0.282237023418
1.9	-0.277594986573	-0.27779860954	-0.277992548569
2.0	-0.273465956943	-0.273699371593	-0.273922137838

Table 9.18: From CCDT2 + linear t3 to CCDT (c)

r_s	CCDT-2 + t3(all)+$(t3t2)_e$	CCDT-2 + t3(all)+$(t2t3)_f$	CCDT-2 + t3(all)+$(t2t3)_g$
0.5	-0.361187132882	-0.361187901826	-0.361187148223
0.6	-0.352904063306	-0.352905522504	-0.352904092948
0.7	-0.345110658791	-0.345113140129	-0.345110710161
0.8	-0.337764873187	-0.337768769556	-0.337764955452
0.9	-0.330828977512	-0.330834737646	-0.330829101587
1.0	-0.324269076573	-0.324277199626	-0.324269255115
1.1	-0.318054677444	-0.318065707615	-0.318054924807
1.2	-0.312158306205	-0.312172827611	-0.312158638377
1.3	-0.306555168756	-0.306573800645	-0.306555603257
1.4	-0.30122285132	-0.301246243665	-0.301223407072
1.5	-0.296141056346	-0.296169885828	-0.296141753505
1.6	-0.291291369789	-0.291326336145	-0.291292229537
1.7	-0.286657056042	-0.286698878762	-0.286658100338
1.8	-0.282222877189	-0.282272292533	-0.282224128479
1.9	-0.277974933588	-0.278032691874	-0.277976414461
2.0	-0.273900523131	-0.273967386287	-0.273902255934

9.8 Benchmarking

Performance benchmarks are provided for both our implementations in Figs. 9.9, 9.11, 9.10, 9.12 and 9.13. Note that not all algorithm have implemented OpenMP [66], thus the overlap in CPU time and elapsed real time. We have obtained some performance benchmarks from Baardsen (in Ref. [5]) that clearly indicates that his CCD implementation outperforms our own. For the largest basis set benchmarked in Figs. 9.9 and 9.10, Baardsen measured only 5 seconds CPU time in total, whereas the initialization time in our implementation is approximately 6 second and each iteration is performed in approximately 0.4 seconds.

Some of this discrepancy may be explained by the fact that the compiler provided by Intel used by Baardsen is known to give better performance than the GNU compiler used by ourselves (see Ref. [67]), but since our main focus has been to efficiently optimize the bottlenecks in our CCDT-1 implementation, we have neglected such bottlenecks in the plain CCD implementation. The performance issues that arise from inclusion of the triple amplitudes overshadows the CCD challenges by far, so any improvement in the CCD implementation would probably not give any significant boost of the CCDT-1. Other explanations could be related to differences between Fortran and C++, such as the many compiler challenges posed by the object orientation in C++.

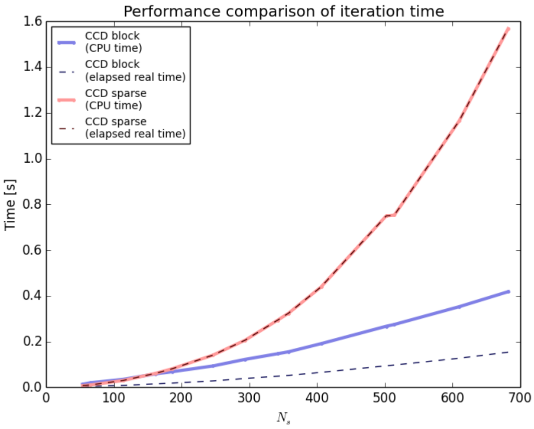


FIGURE 9.9: Elapsed real time and CPU time for the CCD iterations.

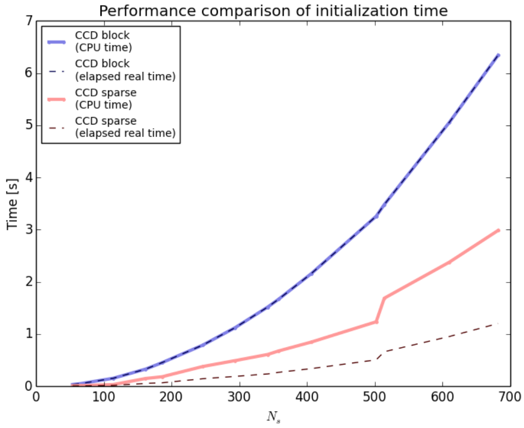


FIGURE 9.10: Elapsed real time and CPU time for the CCD iterations.

When comparing the CCDT-1 imlementations in Figs.9.11, 9.12 and 9.13, it is interesting to note the very large deviation in the initialization time between the two. The reason for this is that the sparse solver does not require any mapping of channels in the diagrams. As we discussed in detail in chapter 8, we may then avoid any sorting of large arrays in the initialization, although initializing the sparse matrices will require some sorting at a later stage. Still, the block implementation outperforms the sparse by far in the CCD iterations, since these actually requires both sorting and traversing of lists, while the block implementation benefits greatly from the BLAS3 optimization of the dense-dense multiplication.

Table 9.19 indicates the relation between the basis set size and the memory requirements for the t3 amplitude. The right column shows the number of nonzero elements in the $\hat{T}_3$ amplitude. For each such nonzero element, several associated values are stored, so as a rough estimate of the memory usage of our CCDT-1 implementation, it is possible to multiply this number with 8 for each such associated data container (storage of current

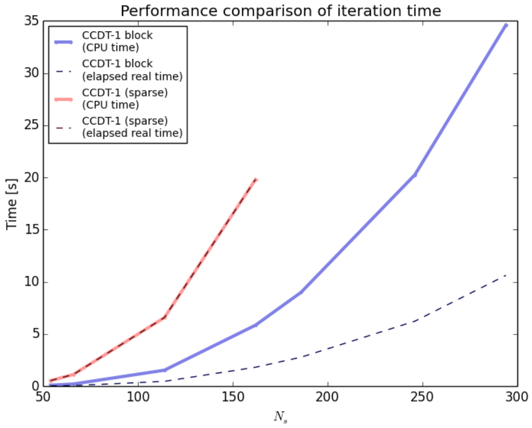


FIGURE 9.11: Elapsed real time and CPU time for the CCD-1 iterations.

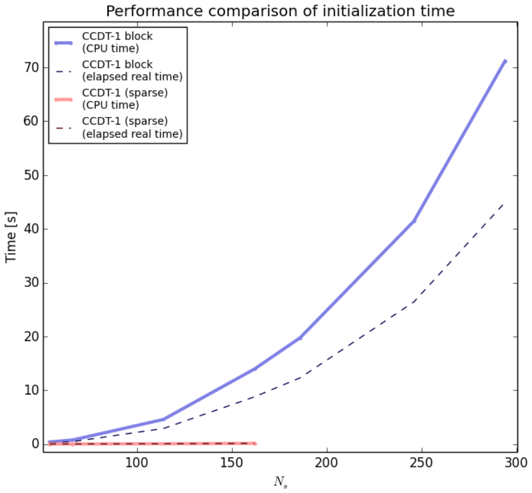


FIGURE 9.12: Elapsed real time and CPU time for the CCD-1 initializations.

and previous amplitudes, energy denominators, the three different block alignments, and finally the sparse blocks for permutations (see chapter 8), and again multiply this number with the number of bytes used for each stored value. All these data types in our implementation used 64 bits, meaning we had 8 byte per data container. This would indicate a memory usage of approximately 2 GB for the largest basis set in table 9.19, which agrees with observations made during runtime.[6]

[6] The author did not perform extensive studies of the memory usage, although we used Valgrind extensively in the debugging process. The observations mentioned was made using the ”top” command during runtime in Ubuntu.

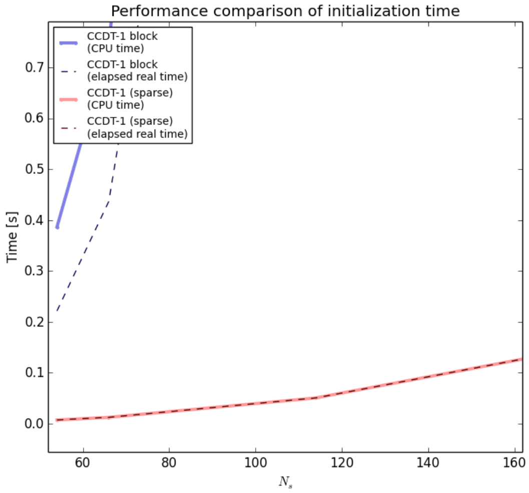


FIGURE 9.13: Elapsed real time and CPU time for the CCD-1 initializations. (Zoomed)

TABLE 9.19: Number of nonzero entries in the $\hat{T}_3$ amplitude

N_s	$N_{nonzeros}$
54	247777
66	503209
114	2745529
162	7813177
186	11025937
246	21651901
294	34037605

The table shows the length of the element vector in the $\hat{T}_3$ amplitude, indicating memory requirements as a function of basis size.

Chapter 10

Conclusions and future prospects

10.1 Conclusions

The electron gas has been studied with various ab-initio techniques, and in recent years these techniques also include coupled cluster doubles (CCD) calculations (see Refs. [1, 4]), many-body perturbation theory (see Ref. [33]), stochastic methods (see Refs.[4, 6, 20]), and even CCD with perturbative treatment of triples (see Ref. [33]). From these studies, it has been clear that the CCD fails to account for important correlations in the homogeneous electron gas in two and three dimensions. In the concluding remarks of his doctoral thesis (see Ref. [1]), Baardsen suggests that inclusion of triple correlations may be necessary to obtain accurate results for the two-dimensional electron gas, and he also propose doing CCDT calculations on the three dimensional case for small basis sets to explore how these correlations play in.

This thesis is likely to be the first time such results are presented for the three-dimensional homogeneous electron gas. With our results,we have demonstrated that it is computationally feasible to do large scale coupled cluster calculations on this system even with the inclusion of triples. The formalism and techniques discussed are intended to be general, so that the same procedure may easily be extended to similar systems for infinite matter.

With our results, we have been able to demonstrate that the triple amplitudes indeed do account for significant correlations in the homogeneous electron gas in three dimensions. While the correlation energy is generally overestimated by the various CCDT truncations, it tends to reduce the deviation to the FCIQMC results (see Refs. [4, 20]) by a factor of roughly 10 compared to the CCD energies, although this ratio has dependency on r_s.

In the same way as for the CCD results, the correlation energy associated with most diagrams in the $\hat{T}_3$ amplitude equation tend to diverge, at least from the CCDT, but probably also from the FCIQMC, as we increase the Wigner-Zeiss radius r_s. This suggests that more complicated correlations from the CCSDTQ become important in this region.

We have performed some light qualitative studies of how the various diagrams contributes to the correlation energy, and we have identified some few diagrams that contribute more than others. These results could possibly be utilized to do truncations in the CCDT for the sake of performance and memory usage, and still retain accurate results.

Some reservation in these conclusions may arise due to the fact that we have neglected both finite size effects and incomplete basis errors. From the comparison with Shepherd *et al.* in Ref. [33], it is clear that the single point extrapolation technique will impact the thermodynamic limit estimates considerably. It would also be interesting to obtain third-party CCDT-1 results, to achieve better validity of our findings.

10.2 Future prospects and recommendations

With the code we have developed, we expect to be able to produce a lot more results than what is presented in this thesis.

First, the main results produced in this thesis is in the authors opinion the data set used in the estimations for the thermodynamical limit. This set could be extended by correcting a minor coding issue in the amplitude initialization function. We suspect that basis sets above 1598 states resulted in a segmentation fault due to a static allocation of memory. At this point, we set up Abel's "hugemem" nodes to prepare 320GB of memory to our calculation, while the maximum limit on these nodes is 1TB. It should be possible to do CCDT-1 calculations beyond 2000 basis states.

Another way of improving this data set is make it more accurate by lowering the convergence threshold. This will lead to prolonged iteration time, but since we have not implemented any optimized conrgence methods, such as *DIIS* (see Ref. [68]), we may be able to compensate for this easily.

Another issue with these results is the insufficient treatment of finite-size effects and incomplete basis error. The single point extrapolation technique described in Refs. [4] and methods discussed in [69] should be used on our results to obtain a more comparable estimate to those of Ref. [33].

While the sparse implementation is mainly used for smaller basis sets, it could be optimized considerably by choosing a more suitable algorithm and container for the sparse tensors. Sparse matrix-matrix multiplication is basically performed by sorting and traversing two lists only once. While our algorithm relies on the Armadillo library for this operation, it does not utilize any parallelization. Functionality such as that provided by CUDA [70] or OpenMP [66] could possibly improve performance to such a degree that also larger basis calculations could be feasible. The syntax used in the sparse implementation allows for fast prototyping, so this could possibly provide us with means to include even more correlations from the CCDTQ equations for smaller basis sets.

Further, it seems to be great potential for optimizations of both implementations. Comparison of the CCD performance benchmarks form Baardsen (see Ref. [5] suggests that the CCD could be improved considerably, and in doing this we could as well get some ideas as to how the CCDT-1 implementation may be optimized using similar techniques. It would be wise to write such an implementation using the Intel MKL library and either Fortran or plain C to achieve best performance possible. Avoiding libraries such as Armadillo will also simplify compilation on computing clusters as they are not always supported, while the Intel compiler generally is.

The simplicity of the sparse implementation syntax is also easy to interface this with the CCAlgebra (see [58]) script discussed in Chapter 6. This process would allow the user to perform complicated calculations of any coupled cluster truncation, even with n-body forces, directly from IPython notebook. IPython supports compilation of C++ code directly from the cells, and it would even be possible to set up the basis class externally, so changing the system would only be a matter of rewriting a few lines in python. This could be a powerful tool, especially in obtaining preliminary results or prototyping code.

It is at the present unknown to the author exactly how the inclusion of more diagrams (or even the full CCDT) in the block implementation would affect the memory usage. Since each diagrams may have values that add into the full element vector, we have also observed quite a intersection between these elements. It could very well be that a full CCDT implementation is still manageable with the basis sizes we have used for the CCDT-1. This should be explored further.

With the formalism, implementations and results presented in this thesis, it should be possible to carry out extensive experiments on the system in question[1]. It is the authors hope that the work presented in this thesis may enable future experiments to focus more on the physical aspects of the system in question, since a great deal of our work has

[1] Other similar systems could also be studied with these means.

been to develop a working and efficient implementation of the triple amplitudes for the homogeneous electron gas.

Appendix A

Diagram names

Because of the way we generated the diagrams in the CCDT equations, there is some inconsistencies concerning how the diagrams are named between chapters 6 and 8. While we use the same naming convention as Shavitt and Bartlett in Ref. [7] for chapter 6, we move to our own naming in chapter 8.

All code written in conjunction with this thesis use the latter naming convention, as these codes at an early stage inherited such names from the calculation with CCAlgebra. It should also be noted that even in Ref. [7], breach of naming convention occurs when the diagrammatic approach is introduced for the CCD equation.[1]

For this reason we provide this table A.1, that will make it easier to translate between the two naming conventions.

[1] They change from $L_{1a} \rightarrow D_{2a}$ for all linear terms, as well as $Q_n \rightarrow D_{xn}$ for the quadratic contributions to the CCD equation.

Table A.1: Translation of diagram names

Name in Ref.[7]	Names used in chapter 8
T_{1a}	$(t_2)_a$
T_{1b}	$(t_2)_b$
T_{2c}	$(t_3)_a$
T_{2d}	$(t_3)_b$
T_{2e}	$(t_3)_c$
T_{3b}	$(t_2t_2)_b$
T_{3c}	$(t_2t_2)_c$
T_{3d}	$(t_2t_2)_d$
T_{5a}	$(t_2t_3)_a$
T_{5b}	$(t_2t_3)_b$
T_{5c}	$(t_2t_3)_c$
T_{5d}	$(t_2t_3)_d$
T_{5e}	$(t_2t_3)_e$
T_{5f}	$(t_2t_3)_f$
T_{5g}	$(t_2t_3)_g$

Bibliography

[1] Gustav Baardsen. Coupled-cluster theory for infinite matter. (June), 2014.

[2] URL http://www.hpc.uio.no/.

[3] Audun Skau Hansen, 2014. URL http://www.github.com/audunsh/fermionmingle.

[4] James J Shepherd, George H Booth, and Ali Alavi. Investigation of the full configuration interaction quantum Monte Carlo method using homogeneous electron gas models. *The Journal of chemical physics*, 136(24), 2012. doi: 10.1063/1.4720076.

[5] Gustav Baardsen. Results from ccd calulations obtained through private communications. 2013.

[6] Alessandro Roggero, Abhishek Mukherjee, and Francesco Pederiva. Quantum Monte Carlo with Coupled-Cluster wave functions. *arXiv preprint arXiv:1304.1549*, 2013.

[7] Isaiah Shavitt and Rodney J. Bartlett. *Many-body methods in physics and chemistry*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK, 2009.

[8] J.M. Thijssen. *Computational Physics*. Cambridge University Press, 1999. ISBN 9780521575881.

[9] Gabriele Giuliani and Giovanni Vignale. *Quantum theory of the electornic liquid*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK, 2005.

[10] Hermann G. Kümmel. A biography of the coupled cluster method. *International Journal of Modern Physics B*, 17(28), 2003. doi: 10.1142/S0217979203020442.

[11] A. Szabo and N.S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, 1989. ISBN 9780486691862.

[12] F.E. Harris, H.J. Monkhorst, and D.L. Freeman. *Algebraic and diagrammatic methods in many-fermion theory.* International series of monographs on chemistry. Oxford University Press, 1992.

[13] E.K.U. Gross, E. Runge, and O. Heinonen. *Many-Particle Theory,.* Taylor & Francis, 1991. ISBN 9780750300728.

[14] D.J. Griffiths. *Introduction to Quantum Mechanics.* Pearson international edition. Pearson Prentice Hall, 2005. ISBN 9780131118928.

[15] R.B. Lindsay and H. Margenau. *Foundations of physics.* Dover Publications, 1957.

[16] Morten Hjorth-Jensen. Lecture notes from fys-kjm4480. 2013. URL http://www.uio.no/studier/emner/matnat/fys/FYS-KJM4480/h13.

[17] G. C. Wick. The evaluation of the collision matrix. *Phys. Rev.*, 80, 1950. doi: 10.1103/PhysRev.80.268.

[18] SymPy Development Team, 2014. URL http://docs.sympy.org/dev/modules/physics/secondquant.html.

[19] George H. Booth, Andreas Gruneis, Georg Kresse, and Ali Alavi. Towards an exact description of electronic wavefunctions in real solids. *Nature*, 493(7432), 2013. ISSN 0028-0836. doi: 10.1038/nature11770.

[20] Karl Leikanger. Full configuration interaction monte carlo studies og quantum dots. Master's thesis, University of Oslo, 2013.

[21] J. Goldstone. Derivation of the brueckner many-body theory. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 239 (1217), 1957. ISSN 0080-4630.

[22] J. Hubbard. The description of collective motions in terms of many-body perturbation theory. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 240(1223), 1957. doi: 10.1098/rspa.1957.0106.

[23] F. Coester. Bound states of a many-particle system. *Nuclear Physics*, 7, 1958. doi: http://dx.doi.org/10.1016/0029-5582(58)90280-3.

[24] F. Coester and R. Haag. Representation of states in a field theory with canonical variables. *Phys. Rev.*, 117, 1960. doi: 10.1103/PhysRev.117.1137.

[25] F. Coester and H. Kümmel. Short-range correlations in nuclear wave functions. *Nuclear Physics*, 17, 1960. doi: http://dx.doi.org/10.1016/0029-5582(60)90140-1.

[26] J. Cizek. On the correlation problem in atomic and molecular systems. calculation of wavefunction components in ursell type expansion using quantum field theoretical methods. *The Journal of Chemical Physics*, 45(11), 1966. doi: http://dx.doi.org/10.1063/1.1727484.

[27] Lührmann Kümmel and Zabolitzky. *Phys. Rep.*, 36:1, 1978.

[28] B. Day and J. G. Zabolitzky. *Nucl. Phys. A*, 366:221, 1981.

[29] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley. Electron correlation theories and their application to the study of simple reaction potential surfaces. *International Journal of Quantum Chemistry*, 14(5), 1978. doi: 10.1002/qua.560140503.

[30] Bartlett R.J. Many-body perturbation-theory and coupled cluster theory for electron correlation in molecules. *Annual review of physical chemistry*, 32, 1981. doi: {10.1146/annurev.pc.32.100181.002043}.

[31] Bruno Linder. *Thermodynamics and introductory statistical mechanics*. John Wiley & Sons, 2004. ISBN 0471474592.

[32] T. Daniel Crawford and Henry F. Schaefer. *An Introduction to Coupled Cluster Theory for Computational Chemists*. John Wiley & Sons, Inc., 2007. ISBN 9780470125915. doi: 10.1002/9780470125915.ch2.

[33] James J. Shepherd and Andreas Grüneis. Many-body quantum chemistry for the electron gas: Convergent perturbative theories. *Physical Review Letters*, 110, 2013. ISSN 00319007. doi: 10.1103/PhysRevLett.110.226401.

[34] The IPython development team, 2015. URL http://ipython.org/.

[35] URL http://continuum.io/.

[36] B. D. Day. Elements of the brueckner-goldstone theory of nuclear matter. *Rev. Mod. Phys.*, 39, 1967. doi: 10.1103/RevModPhys.39.719.

[37] Edgar Solomonik, Darin Matthews, Jeff R Hammond, and James Demmel. Cyclops tensor framework: reducing communication and eliminating load imbalance in massively parallel contractions. In *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, pages 813–824, 2013.

[38] Gustavo E Scuseria, Andrew C Scheiner, Timothy J Lee, Julia E Rice, and Henry F Schaefer III. The closed-shell coupled cluster single and double excitation (ccsd) model for the description of electron correlation. a comparison with configuration interaction (cisd) results. *The Journal of chemical physics*, 86(5), 1987.

[39] Audun Skau Hansen, 2014. URL http://www.github.com/audunsh/fermicc.

[40] James J Shepherd, Thomas M Henderson, and Gustavo E Scuseria. Coupled cluster channels in the homogeneous electron gas Coupled cluster channels in the homogeneous electron gas. 124102, 2014. doi: 10.1063/1.4867783.

[41] A. Burrows. *Rev. Mod. Phys.*, 85:245, 2013.

[42] F. Weber. Pulsars as astrophysical laboratories for nuclear and particle physics. *Institute of Physics Publishing, London*, 1999.

[43] H. Heiselberg and M. Hjorth-Jensen. *Phys. Rep.*, 328:237, 2000.

[44] C. M. Singal and T. P. Das. Self-consistent pair-correlation approach to many-body effects in metals. *Phys. Rev. B*, 8, 1973. doi: 10.1103/PhysRevB.8.3675.

[45] David L. Freeman. Coupled-cluster expansion applied to the electron gas: Inclusion of ring and exchange effects. *Phys. Rev. B*, 15, 1977. doi: 10.1103/PhysRevB.15.5512.

[46] R. F. Bishop and K. H. Lührmann. Electron correlations: I. ground-state results in the high-density regime. *Phys. Rev. B*, 17, 1978. doi: 10.1103/PhysRevB.17.3757.

[47] R. F. Bishop and K. H. Lührmann. Electron correlations. ii. ground-state results at low and metallic densities. *Phys. Rev. B*, 26, 1982. doi: 10.1103/PhysRevB.26.5523.

[48] Karl Rottmann. *Matematisk formelsamling.* Bracan Forlag, 1995. ISBN 8278220050.

[49] P. P. Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der Physik*, 369(3), 1921. ISSN 1521-3889. doi: 10.1002/andp.19213690304.

[50] Louisa M Fraser, W M C Foulkes, G Rajagopal, R J Needs, S D Kenny, and a J Williamson. Finite-size effects and Coulomb interactions in quantum Monte Carlo calculations for homogeneous systems with periodic boundary conditions. *Phys. Rev. B*, 53(4):1814–1832, 1996. doi: 10.1103/PhysRevB.53.1814.

[51] N. D. Drummond, R. J. Needs, A. Sorouri, and W. M. C. Foulkes. Finite-size errors in continuum quantum monte carlo calculations. *Phys. Rev. B*, 78, 2008. doi: 10.1103/PhysRevB.78.125106.

[52] James J. Shepherd, Andreas Grüneis, George H. Booth, Georg Kresse, and Ali Alavi. Convergence of many-body wave-function expansions using a plane-wave basis: From homogeneous electron gas to solid state systems. *Physical Review B - Condensed Matter and Materials Physics*, 86(3), 2012. ISSN 10980121. doi: 10.1103/PhysRevB.86.035111.

[53] James J. Shepherd and Andreas Grüneis. Many-body quantum chemistry for the electron gas: Convergent perturbative theories. *Phys. Rev. Lett.*, 110, 2013. doi: 10.1103/PhysRevLett.110.226401.

[54] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45, 1980. doi: 10.1103/PhysRevLett.45.566.

[55] P. López Ríos, A. Ma, N. D. Drummond, M. D. Towler, and R. J. Needs. Inhomogeneous backflow transformations in quantum monte carlo calculations. *Phys. Rev. E*, 74, 2006. doi: 10.1103/PhysRevE.74.066701.

[56] I G Gurtubay, R Gaudoin, and J M Pitarke. Benchmark quantum monte carlo calculations of the ground-state kinetic, interaction and total energy of the three-dimensional electron gas. *Journal of Physics: Condensed Matter*, 22(6), 2010.

[57] Sarah Reimann. Quantum-mechanical systems in traps and similarity renormalization group theory. Master's thesis, University of Oslo, 2013.

[58] Audun Skau Hansen, 2015. URL http://audunsh.github.io/ccalg_web.html.

[59] URL http://www.netlib.org/blas/.

[60] George M. Karniadakis and Robert M. Kirby. *Parallel Scientific Computing in C++ and MPI*. Cambridge University Press, New York, NY, USA, 2003. ISBN 0521817544.

[61] D.C. Lay. *Linear Algebra and Its Applications*. Addison-Wesley, 2006. ISBN 0321287134. URL https://books.google.no/books?id=_4bjtgAACAAJ.

[62] Jack Dongarra. URL http://netlib.org/linalg/html_templates/node90.html.

[63] Conrad Sanderson. Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments, 2010. URL http://arma.sourceforge.net/.

[64] Steve Smith. URL http://programmer.97things.oreilly.com/wiki/index.php/Don't_Repeat_Yourself.

[65] Neil Drummond. Ewald interactions and finite-size errors, 2007 summer school on computational materials science. 2007. URL http://www.mcc.uiuc.edu/summerschool/2007/qmc/tutorials/EwaldInteractions_Drummond.pdf.

[66] URL http://openmp.org/wp/.

[67] hpc drift@usit.uio.no. Private communication with department for research computing, usit. 2015.

[68] Gustavo E. Scuseria, Timothy J. Lee, and Henry F. Schaefer III. Accelerating the convergence of the coupled-cluster approachThe use of the DIIS method. *Chemical Physics Letters*, 130, 1986. doi: 10.1016/0009-2614(86)80461-4.

[69] Neil Drummond. Ewald Interactions and Finite-Size Errors. *Computational Materials Science*, 2007.

[70] URL http://www.nvidia.com/object/cuda_home_new.html.