

# STOCHASTIC OPTIMIZATION FOR MADY-BODY PROBLEM

by

Andrei Kukharenka

THESIS

for the degree of

MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences  
University of Oslo

August 2018



# Acknowledgements

Now, when this Master thesis is finished and I am standing on the cusp of the turning point of my life, I would like to express my sincere gratitude to every person that has contributed to this work and supported me during such a challenging and important time period.

I am immensely grateful to my supervisor, Professor Morten Hjorth-Jensen – the one who has always lent both scientific and moral support. At the most hard moments Morten helped me to regain confidence and encouraged to assign difficult outstanding objectives. Working under the supervision of Professor Morten Hjorth-Jensen inspired me to strive for perfection regardless of how challenging my tasks were.

I would like to thank Professor Anders Malthe-Sørenssen and Grete Stavik-Døvle for their help and assistance with necessary arrangements at the start of my full-time study as a master student.

I am also very thankful to Håkon, Morten Ledum and Øyvind, Sebastian and Mathias for our conversations on physics, in particular, discussions about coupled cluster theory, which gave rise to many ideas, as well as for fun time that was of great importance during this hard work. As my teaching assistants on courses of computational physics, Håkon and Morten Ledum showed high professionalism and made great contribution to the achieved results.

I am deeply grateful to Alocias for his significant helping hand in coding process. I should also thank Håkon and Julia Vinnik for time and efforts they spent on proofread of my Slavic English. Thanks to my Italian friends Giovanni and Alessio for constant support and inspiration.

I'm grateful to Anna for endless discussion about physics and statistics. Your fantastic ability to spot errors in the code saved me a lot of time during last year.

Finally, thanks to every person accompanied me on this challenging path.



# Contents

0.1	Motivation . . . . .	1
0.2	Thesis structure . . . . .	1
0.3	Collaboration . . . . .	2
<b>1</b>	<b>Hartree-Fock Theory</b>	<b>3</b>
1.1	The many-body problem . . . . .	3
1.2	Hartree-Fock equations . . . . .	5
<b>2</b>	<b>Coupled Cluster Theory</b>	<b>9</b>
2.1	The exponential ansatz . . . . .	9
2.2	Coupled Cluster theory . . . . .	12
2.3	Variational Coupled Cluster . . . . .	14
2.4	Algebraic approach . . . . .	15
2.5	Diagrammatic approach . . . . .	18
<b>3</b>	<b>Quantum Monte Carlo Methods</b>	<b>25</b>
3.1	Variational Monte Carlo . . . . .	25
3.1.1	The expectation value of the Hamiltonian . . . . .	28
3.1.2	The Metropolis algorithm . . . . .	28
3.2	The Projector Monte Carlo . . . . .	30
3.3	The PMC general algorithm . . . . .	32
3.4	Diffusion Monte Carlo . . . . .	34
3.5	Full Configuration Interaction Quantum Monte Carlo . . . . .	36
3.6	The Population Monte Carlo Algorithm . . . . .	38
3.7	The sign problem . . . . .	39
<b>4</b>	<b>Coupled Cluster Monte Carlo</b>	<b>43</b>
4.1	the Coupled Cluster Monte Carlo theory . . . . .	43
4.2	The general CCQMC algorithm . . . . .	47
4.3	Normalization . . . . .	48
4.4	Population dynamics . . . . .	49
4.5	Population Control . . . . .	50
4.6	Initiator approximation . . . . .	50

<b>5</b>	<b>The Infinite Electron Gas</b>	<b>51</b>
5.1	The Hamiltonian . . . . .	52
5.2	Finite-box approximations of the electron gas . . . . .	52
5.3	Ewald's summation technique . . . . .	53
5.4	Analytical results for the electron gas . . . . .	55
<b>6</b>	<b>CCD solver implementation</b>	<b>57</b>
6.1	The Coupled Cluster Equations . . . . .	57
6.2	The Straight Forward Implementation . . . . .	60
6.3	The Sparse Implementation . . . . .	62
6.4	The Block Implementation . . . . .	64
6.5	The Symmetry Channels . . . . .	65
6.6	Diagrams calculation . . . . .	68
6.7	The block solver workflow . . . . .	69
6.8	Program structure and hierarchy . . . . .	71
6.8.1	Single-particle basis class . . . . .	71
6.8.2	The electron gas single-particle basis class . . . . .	71
6.8.3	Single-particle state class . . . . .	72
6.8.4	Symmetry channels class . . . . .	72
6.8.5	The electron gas symmetry channels class . . . . .	73
6.8.6	Channel class . . . . .	73
6.8.7	Symmetry block class . . . . .	74
6.8.8	CCD solver class . . . . .	74
6.8.9	MBPT2 classes . . . . .	75
6.9	Possible optimization . . . . .	75
6.10	Occupation Number Representation of SD . . . . .	77
6.11	CCQMC Cluster Size . . . . .	80
6.12	CCQMC Solver . . . . .	80
6.13	CCQMC Sampling . . . . .	82
6.14	Sampling of the action of the Hamiltonian . . . . .	86
6.15	The Hamiltonian matrix elements . . . . .	87
6.16	The sign of excip . . . . .	87
6.17	CCQMC code structure . . . . .	88
6.18	Possible optimization . . . . .	90
<b>7</b>	<b>Results and Discussion</b>	<b>93</b>
7.1	CCD benchmark . . . . .	93
7.2	Thermodynamic and complete basis set limit estimates . . . . .	96
7.3	CCQMC population dynamics . . . . .	96
7.4	Reference population . . . . .	101
7.5	Comments on the energy . . . . .	102

<b>8</b>	<b>Conclusions And Future Prospects</b>	<b>107</b>
8.1	Future outlook . . . . .	108
	<b>Appendices</b>	<b>109</b>
A.1	Second quantization . . . . .	111
B.2	Particle-hole formalism . . . . .	111
C.3	Normal ordering . . . . .	112
D.4	Wick's theorem . . . . .	112





## 0.1 Motivation

Coupled Cluster Quantum Monte Carlo (CCQM) is a novel *ab initio* quantum mechanical many-body methods method that holds great promise in tackling the large dimensionalities encountered in various many-body calculations. It combines features from two different approaches for the treatment of many-body problem. It includes a successful approximation of the true wavefunction with the exponential ansatz and it provides excellent results and solves a number of important problems, like size consistency and size extensivity as well as having the potential to handle the large dimensionality of basis functions, typically encountered in most wave function based methods like coupled cluster theory. It is possible to reduce extremely large spaces of basis functions thanks to the stochastic approach inherent in Monte Carlo methods.

For electronic systems, mean-field methods like density functional theory are among the most popular approach<sup>1</sup> among *ab initio* methods to study many-body systems. However, mean field methods in general tackle poorly strongly correlated systems. The hope is that this novel CCQM method, with the potential to model stochastically complicated states of many-body systems, can together with more traditional methods like full configuration interaction theory or coupled cluster theory, allow us to tackle the exponential growth in dimensionality in a computationally tractable way.

Our goal was to develop a software which implements the stochastic coupled cluster Monte Carlo method and benchmark it against other familiar methods and apply to selected many-body systems like the homogeneous electron gas. There is a limited amount of information available on the important technicalities of the algorithm and only recently scientific articles on topic have been published. Besides a group of Thom *et al.* at the university of Cambridge in the UK, our software is the only other one existing. We hope our study helps to create a foundation for the further development of this promising method.

## 0.2 Thesis structure

The structure of this thesis has been conceived as an introduction to readers with little- to no-experience in the field of many-body algorithms. Often materials on this topic contain a large amounts of information followed by the detailed mathematical derivations. We tried to make this thesis easy for reading, though still including all key points.

The aim of the second Chapter is to stand the many-body problem and to present independent particle model. One possibility to go beyond this model is the coupled cluster method, which includes correlation between particles. We

---

<sup>1</sup>6 out of 10 the most highly cited articles from the period 1904 - 2014 are about density functional theory[17].

take up coupled cluster theory in Chapter 3, discussing the algebraic approach and diagrammatic notation. Quantum Monte Carlo methods provide an alternative passage beyond the Hartree-Fock limit. We provide an overview of stochastic methods in many-body physics in Chapter 4, dwelling on Variational Monte Carlo and Projector Monte Carlo. Last sections of this chapter contain introduction into stochastic method in the space of Slater determinants (Full Configuration Quantum Monte Carlo) and population dynamics algorithm. We believe that this order of presenting material allows reader to acquire quickly general picture of stochastic many-body methods, and particularly the place of coupled Cluster Monte Carlo within it. Chapter 6 largely focuses on a brief description of the homogeneous electron gas covering the motivation to choose this system as the main object of research in this thesis. We then turn to the different aspects of the implementation starting in Chapter 7 with a thorough discussion of the standard coupled cluster algorithm. Optimization technicalities together with the class structure are discussed in the first sections. The reminder of this chapter is devoted to stochastic coupled cluster theory and presents the key points needed for producing a computer implementation of the method. The final chapters of this thesis are devoted to results of applying deterministic and stochastic coupled cluster approaches to the electron gas.

### 0.3 Collaboration

The computer code in this master thesis developed in collaboration with Anna Gribkovskaya. We worked closely together on the analysis of the results and understanding of the CCQMC method. Sections 7.4 - 7.5 of this thesis were written by two authors.

# Chapter 1

## Hartree-Fock Theory

The Hartree-Fock method is a popular method in quantum chemistry and material science even despite the fact that it is one of the oldest ones developed for computing an approximative solution to the many-body problem. Usually Hartree-Fock results are used as a starting point for other more sophisticated many-body methods. We start this section with a formulation of the many-body problem in studies of quantum mechanical systems followed by the discussion of the Hartree-Fock method. Finally, we focus our attention on the derivation of the Hartree-Fock equations. The emphasis is on systems with electronic degrees of freedom.

The main idea of the Hartree-Fock method is to describe the interaction between particles as an effective potential where they move independently. This leads to system of single particle Schrödinger equations for the individual one-electron wavefunctions with a potential determined by other electrons, possibly in addition to the external potential. According to this, the method is often called a mean-field approximation.

### 1.1 The many-body problem

Real systems unlike idealized cases such as the harmonic oscillator can rarely or almost never be solved analytically in quantum mechanics. Clearly we have to use approximations and simplifications and solve the task numerically. Let us consider a system of interacting electrons and a nucleus. The main quantity of interest is the energy of the system. If we treat the system from a quantum mechanical point of view the measurable quantities are expectation values of operators representing the observables of the system. To obtain the energy one needs to solve the eigenvalue problem:

$$\hat{H}\Psi = E\Psi,$$

where  $\hat{H}$  is the Hamiltonian of the system and  $\Psi$  is wavefunction. Let us write out the Hamiltonian assuming we have  $N_e$  electrons and  $N_n$  nuclei. The wavefunction of the system is now a function of all electronic and nucleic coordinates, namely  $\Psi = \Psi(\{\mathbf{r}, \mathbf{R}\}) = \Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e}, \mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{N_n})$ . The time-independent Schrödinger equation can be written as:

$$\hat{H}\Psi_n(\{\mathbf{r}, \mathbf{R}\}) = E_n\Psi_n(\{\mathbf{r}, \mathbf{R}\}), \quad (1.1)$$

with the Hamiltonian:

$$\hat{H} = -\sum_{j=1}^{N_e} \frac{\hbar^2 \nabla_j^2}{2m_e} - \sum_{\alpha=1}^{N_n} \frac{\hbar^2 \nabla_\alpha^2}{2m_\alpha} + \sum_{j < j'}^{N_e} \frac{e^2}{|\mathbf{r}_j - \mathbf{r}_{j'}|} + \sum_{\alpha \neq \alpha'}^{N_n} \frac{Z_\alpha Z_{\alpha'} e^2}{|\mathbf{R}_\alpha - \mathbf{R}_{\alpha'}|} - \sum_{j=1}^{N_e} \sum_{\alpha=1}^{N_n} \frac{Z_\alpha e^2}{|\mathbf{R}_\alpha - \mathbf{r}_j|},$$

where the first term represents the kinetic energy of the electrons, the second term is the kinetic energy of all nuclei, while the third represents the repulsive electron-electron interactions. The two last terms corresponds to repulsive nuclei-nuclei interactions and attractive electron-nuclei interactions respectively.

In equation (1.1) it is clear that the electronic motion is coupled to the ionic motion. Partial decoupling can be achieved by the so called Born-Oppenheimer approximation. We can then rewrite the wavefunction for the system as follows:

$$\Psi_n(\{\mathbf{r}, \mathbf{R}\}) = \psi_n(\{\mathbf{r}, \mathbf{R}\})\Theta_n(\{\mathbf{R}\}),$$

where  $\psi_n(\{\mathbf{r}, \mathbf{R}\})$  is the many-electron wavefunction which depends on the position of the nuclei and  $\Theta_n(\{\mathbf{R}\})$  is the wavefunction of the nuclei. Using separation of variables one obtains two differential equations which are coupled via the total electronic energy  $E_n(\mathbf{R})$ :

$$\left[ -\sum_{j=1}^{N_e} \frac{\hbar^2 \nabla_j^2}{2m_e} - \sum_{\alpha=1}^{N_n} \frac{\hbar^2 \nabla_\alpha^2}{2m_\alpha} + \sum_{j < j'}^{N_e} \frac{e^2}{|\mathbf{r}_j - \mathbf{r}_{j'}|} \right] \psi_n(\{\mathbf{r}, \mathbf{R}\}) = E_n(\mathbf{R})\psi_n(\{\mathbf{r}, \mathbf{R}\}), \quad (1.2)$$

and

$$\left[ -\sum_{I=1}^N \frac{\hbar^2 \nabla_I^2}{2m_\alpha} + \sum_{I \neq J}^N \frac{Z_I Z_J e^2}{|\mathbf{R}_I - \mathbf{R}_J|} + E_n(\mathbf{R}) \right] \Theta_n(\{\mathbf{R}\}) = E_n^{en} \Theta_n(\{\mathbf{R}\}). \quad (1.3)$$

Often the kinetic energy of the nuclei can be neglected and the energy of such a system consists of the electrostatic energy of the nuclei and the energy of the electrons. However solving equation (1.2) is still a difficult task since the number of degrees of freedom is large.

## 1.2 Hartree-Fock equations

Slater determinants constructed from  $N$  orthonormal single electron wavefunctions are normally used as a smart guess for the true many-body wavefunction, which is usually unknown:

$$\Phi(x_1, x_2, \dots, x_N, \alpha, \beta, \dots, \sigma) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_\alpha(x_1) & \psi_\alpha(x_2) & \dots & \psi_\alpha(x_N) \\ \psi_\beta(x_1) & \psi_\beta(x_2) & \dots & \psi_\beta(x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_\sigma(x_1) & \psi_\sigma(x_2) & \dots & \psi_\sigma(x_N) \end{vmatrix}, \quad (1.4)$$

where  $\alpha, \beta, \dots$  represent a set of quantum numbers describing one particular single-particle state. This design provides the wavefunction of the fermionic system with the antisymmetry property required by the Pauli principle:

$$\Phi(x_1, x_2, \dots, x_i, \dots, x_j, \beta, \dots, \sigma) = -\Phi(x_1, x_2, \dots, x_j, \dots, x_i, \beta, \dots, \sigma). \quad (1.5)$$

The Hamiltonian of the system can generally be written as:

$$\hat{H} = \hat{T} + \hat{U}_{ee} + \hat{V}_{ext}, \quad (1.6)$$

where  $\hat{T}$  is the electron kinetic energy,  $\hat{U}_{ee}$  is the coulomb interaction between the electrons and  $\hat{V}_{ext}$  is the external potential. The Hamiltonian then reduces to:

$$\hat{H} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 + \sum_{i<j}^N \frac{1}{r_{ij}} - \sum_{i=1}^N \frac{Z}{r_i} = \hat{H}_0 + \hat{H}_I,$$

where natural units are used for simplicity ( $\hbar = c = e = m_e = 1$ ). All one-body operators are united under  $\hat{H}_0$  while the two-body operator is denoted as  $\hat{H}_I$ . To find the ground state energy  $E_0$ , then we use the variational principle[36]:

$$E_0 \leq E[\Phi] = \langle \Phi | \hat{H}_0 | \Phi \rangle + \langle \Phi | \hat{H}_I | \Phi \rangle. \quad (1.7)$$

In equation (1.7) the analytical expression for the Hamiltonian can be written explicitly. A single Slater determinant is chosen to be a good approximation for the wavefunction, however the single-particle wavefunctions are generally not known for arbitrary systems. Hence, they are the primary unknowns to be optimized using the variational principle. One of the most popular strategies is to expand the single-particle functions in a known basis and vary the expansion coefficients.

$$\psi_a(\mathbf{r}) = \sum_{\alpha}^Z C_{a\alpha} \psi_{\alpha}(\mathbf{r}) \quad (1.8)$$

It's easy to show that<sup>1</sup> the first term in equation (1.7) can be simplified and expressed in terms of the matrix elements of one-body operator:

---

<sup>1</sup>See Ch. 15.3 in [15] for derivation in first-quantized form. Same result can be obtained by application of Wick's theorem and using second-quantized form[21].

$$\langle \Phi | \hat{H}_0 | \Phi \rangle = \sum_{\mu=1}^N \langle \mu | \hat{h}_0 | \mu \rangle. \quad (1.9)$$

Likewise it is not difficult to show that the second term can be simplified to:

$$\langle \Phi | \hat{H}_I | \Phi \rangle = \frac{1}{2} \sum_{\mu=1}^N \sum_{\nu=1}^N \langle \mu\nu | \hat{v} | \mu\nu \rangle_{AS}, \quad (1.10)$$

where

$$\begin{aligned} \langle \mu\nu | \hat{v} | \mu\nu \rangle_{AS} &= \langle \mu\nu | \hat{v} | \mu\nu \rangle - \langle \mu\nu | \hat{v} | \nu\mu \rangle \\ &= \langle \psi_\mu(x_i) \psi_\nu(x_j) | \hat{v}(r_{ij}) | \psi_\mu(x_i) \psi_\nu(x_j) \rangle \\ &\quad - \langle \psi_\mu(x_i) \psi_\nu(x_j) | \hat{v}(r_{ij}) | \psi_\nu(x_i) \psi_\mu(x_j) \rangle. \end{aligned}$$

For simplicity the single-particle states in the original basis(1.8) is denoted with Greek letters  $\psi_\mu(x_i) \rightarrow \mu$ . Summing (1.9) and (1.10) we obtain a functional which is minimized according to (1.7):

$$E[\Phi] = \sum_{\mu=1}^N \langle \mu | \hat{h}_0 | \mu \rangle + \frac{1}{2} \sum_{\mu=1}^N \sum_{\nu=1}^N \langle \mu\nu | \hat{v} | \mu\nu \rangle_{AS}, \quad (1.11)$$

In the new basis denoted by Latin letters (our Hartree-Fock orbitals), the energy functional is:

$$E[\Phi_{HF}] = \sum_{a=1}^N \langle a | \hat{h}_0 | a \rangle + \frac{1}{2} \sum_{a=1}^N \sum_{b=1}^N \langle ab | \hat{v} | ab \rangle_{AS}, \quad (1.12)$$

By substituting (1.8) into (1.12):

$$E[\Phi_{HF}] = \sum_{a=1}^N \sum_{\alpha,\beta=1}^Z C_{a\alpha} C_{a\beta} \langle \alpha | \hat{h}_0 | \beta \rangle + \frac{1}{2} \sum_{a,b=1}^N \sum_{\alpha,\beta,\gamma,\delta=1}^Z C_{a\alpha}^* C_{b\beta}^* C_{a\gamma} C_{b\delta} \langle \alpha\beta | \hat{v} | \gamma\delta \rangle_{AS}. \quad (1.13)$$

the coefficients  $C_{a\alpha}$  can be chosen to represent a unitary transformation preserving orthogonality. Knowing that  $\psi_\alpha$  is an orthogonal basis we find a constraint for the coefficients  $C_{a\lambda}$ :

$$\langle a | b \rangle = \delta_{a,b} = \sum_{\alpha\beta} C_{a\alpha}^* C_{b\beta} \langle \alpha | \beta \rangle = \sum_{\alpha} C_{a\alpha}^* C_{b\alpha}, \quad (1.14)$$

Recall that computing an extrema for the constrained problem is equivalent to an unconstrained extrema examination of the Lagrange functional with respect to

the expansion coefficients  $C$  and the Lagrange multipliers. Using the constraint (1.14) the Lagrange functional to be minimized is:

$$L[\Phi_{HF}] = E[\Phi_{HF}] - \sum_{a=1}^N \epsilon_a \sum_{\alpha=1}^Z C_{a\alpha}^* C_{a\alpha}. \quad (1.15)$$

Taking the derivative with respect to  $C_{i\alpha}^*$ :

$$\frac{\delta L[\Phi_{HF}]}{\delta C_{i\alpha}^*} = 0, \quad (1.16)$$

one obtains

$$\sum_{\beta} h_{\alpha\beta}^{HF} C_{i\beta} = \epsilon_i C_{i\alpha}. \quad (1.17)$$

Equation (1.17) is a standard eigenvalue problem, where  $h_{\alpha\beta}^{HF}$  is called the Hartree-Fock matrix and is expressed as follows:

$$h_{\alpha\beta}^{HF} = \langle \phi_{\alpha} | \hat{h}_0 | \phi_{\beta} \rangle + \sum_{j=1}^N \sum_{\gamma=1}^Z \sum_{\delta=1}^Z C_{j\gamma}^* C_{j\delta} \langle \phi_{\alpha} \phi_{\gamma} | \hat{v} | \phi_{\beta} \phi_{\delta} \rangle_{AS}. \quad (1.18)$$

The Hartree-Fock equations (1.17) are non-linear eigenvalue equations and must be solved iteratively. The Hartree-Fock basis is not known and can be found by solving (1.17) in order to find the expansion coefficients for (1.12). Equation (1.18) can be simplified for numerical calculations introducing the so called density matrix:

$$\rho_{\gamma\delta} = \sum_{i=1}^N C_{i\gamma} C_{i\delta}^*. \quad (1.19)$$





# Chapter 2

## Coupled Cluster Theory

In this chapter we review Coupled Cluster(CC) theory, which is referred to as the "golden standard" in the field of quantum chemistry. Born in the field of nuclear quantum mechanics around 1960, CC drew the attention of chemists soon after its introduction. Energies within "chemical accuracy" can be obtained with this method for many chemically interesting systems with a relatively low computational cost. By virtue of these aspects CC theory is reviewed in a systematic and detailed way in many sources, see for example the textbook by Shavitt and Bartlett [28] or Crawford and Schaefer [7]. We borrow notation and argumentation mostly from [7]. In the following sections we present the formal theory. Both an analytical and diagrammatic method for obtaining programmable CC equations are covered.

### 2.1 The exponential ansatz

The HF method is one of the more practical ways of treating the many-body problem. As mentioned in the previous chapter, Slater determinants are used as an assumption about the form of the wavefunction for fermionic system consisting of  $N$  particles:

$$|\Psi_{HF}\rangle \equiv |\Phi\rangle = |\psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\dots\psi_N(x_N)\rangle.$$

By design this wave function describes uncorrelated electrons except one property, the Pauli principle which is baked in into the properties of the determinant. Coupled cluster theory sets out to construct a better wavefunction in order to enhance our understanding of correlations influencing the properties of the system.

It stands to reason, one can take the best wavefunction approximation available(HF wavefunction) and improve it by introducing correlations:

$$|\Psi\rangle = \underbrace{|\Phi\rangle}_{\text{Hartree-Fock}} + \underbrace{\Delta|\Psi\rangle}_{\text{correlatons}}.$$

Intuitively, it is apparent to introduce some analogy of a correlation function for the particles. For a pair of fermions it can be defined as:

$$g_{1,2}(x_1, x_2) = \sum_{p>q} c_{pq} \psi_p(x_1) \psi_q(x_2),$$

and a somewhat improved wavefunction follows:

$$|\Psi_{\text{improved}}\rangle = |\Phi\rangle + \langle x_1, x_2, x_3 \dots x_N | g_{1,2} \psi_3 \psi_4 \dots \psi_N \rangle.$$

The function  $g$  is called a *cluster function*. The function  $|\Psi_{\text{improved}}\rangle$  describes the system where all electrons are independent of each other apart from two particular particles – a *cluster* formed by two electrons. However it is not strictly correct to consider correlation between two particular fermions as they are indistinguishable. Instead we construct a function correlating the motion of any two electrons within a couple(or more) of chosen orbitals. For the  $i$ - and  $j$ -orbitals it is written as:

$$g_{i,j}(x_m, x_n) = \sum_{a>b} t_{ij}^{ab} \psi_a(x_m) \psi_b(x_n), \quad (2.1)$$

where  $t_{ij}^{ab}$  are the cluster coefficients to be defined below. The improved wavefunction takes into consideration the correlated motion of any pair of fermions occupying the  $i$ - and  $j$ -orbitals and can be expressed as:

$$|\Psi'_{\text{improved}}\rangle = |\Phi\rangle + (-1)^{i-j+1} \langle x_1, x_2, x_3 \dots x_N | g_{i,j} \psi_1 \psi_2 \psi_3 \psi_4 \dots \cancel{\psi_i} \dots \cancel{\psi_j} \dots \psi_N \rangle. \quad (2.2)$$

We consider the case  $i < j$ , which means we to account for a sign in the expression via factor  $(-1)^{i-j+1}$ . Further improvements can be achieved by taking into account correlations between all possible pairwise combinations of orbitals. In addition by introduction of three-orbital cluster functions and continuing this process up to  $N$  orbitals we can obtain the exact wavefunction within the space spanned by the single-particle functions. One can even add single-orbital cluster functions, representing one-electron basis adjustment as other cluster functions are added. If one substitutes the expression for the cluster function (2.1) into equation (2.2) it is easy to notice that the right hand side is a linear combination of determinants each of which differs from the reference by a specific number of orbitals and can be rewritten as the result of the action of a substitution operator.

The formalism can be simplified if we use the formalism of second quantization. The doubles cluster operator is defined as:

$$\hat{t}_{ij} \equiv \sum_{a>b} t_{ij}^{ab} c_a^\dagger c_b^\dagger c_j c_i,$$

while the singles cluster operator is given by:

$$\hat{t}_i \equiv \sum_a t_i^a c_a^\dagger c_i.$$

Generalizing to  $N$ -orbitals we have that:

$$\hat{t}_{ij\dots} \equiv \sum_{ij\dots ab\dots} t_{ij\dots}^{ab\dots} c_a^\dagger c_b^\dagger \dots c_j c_i.$$

Equation (2.2) is then simplified to:

$$|\Psi'_{\text{improved}}\rangle = |\Phi\rangle + \hat{t}_{ij} |\Phi\rangle.$$

The creation operators in the cluster operator act on unoccupied orbitals only while the annihilation operators in turn can act on the occupied orbitals. Hence creation-annihilation pairs within the cluster operator anticommute:

$$c_p c_q^\dagger = \delta_{pq} - c_p^\dagger c_q,$$

$$c_i c_a^\dagger + c_a^\dagger c_i = \delta_{ai} = 0,$$

This leads to an important conclusion, all operators  $\hat{t}_{ij}$  commute among themselves, as they are linear combinations of products of excitation operators  $\hat{c}_a^\dagger \hat{c}_i$ .

We define the total one- and two-orbital cluster operators  $\hat{T}_1$  and  $\hat{T}_2$ . It is possible to define such operators up to  $N$  (that is  $N$ -particle- $N$ -hole excitations), as this represents the maximum level of excitations we can introduce by the scheme described above. The general cluster operator is then written as:

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \dots + \hat{T}_N,$$

where:

$$\hat{T}_1 = \sum_i \hat{t}_i = \sum_{ia} t_i^a c_a^\dagger c_i,$$

$$\hat{T}_2 = \sum_{i<j} \hat{t}_{ij}^{ab} = \frac{1}{2!^2} \sum_{ijab} t_{ij}^{ab} c_a^\dagger c_b^\dagger c_j c_i,$$

and

$$\hat{T}_3 = \sum_{i<j<k} \hat{t}_{ijk}^{abc} = \frac{1}{3!^2} \sum_{ijkabc} t_{ijk}^{abc} c_a^\dagger c_b^\dagger c_c^\dagger c_k c_j c_i$$

The fact that the total cluster operators ( $\hat{T}_1, \hat{T}_2, \dots$ ) commute allows us to write an expression for the wavefunction in the following form:

$$\Psi_{CC} = e^{\hat{T}} |\Phi\rangle. \quad (2.3)$$

Equation (2.3) is known as the *exponential ansatz*. The exponential general cluster operator acting on the reference wavefunction produces new function

containing cluster functions correlating the motion of electrons within certain orbitals. Truncation of the general cluster operator at certain levels leads to a hierarchy of coupled cluster approximations:

$$\Psi_{CC}^S = e^{\hat{T}_1} |\Phi\rangle \text{ CCS, couple-cluster singles,}$$

and

$$\Psi_{CC}^{SD} = e^{\hat{T}_1 + \hat{T}_2} |\Phi\rangle \text{ CCSD, couple-cluster singles, doubles,}$$

and

$$\Psi_{CC}^{SDT} = e^{\hat{T}_1 + \hat{T}_2 + \hat{T}_3} |\Phi\rangle \text{ CCSDT, couple-cluster singles, doubles, triples.}$$

The cluster operators  $\hat{T}_i$  are also called excitation operators, since acting on the reference determinant they produce determinants with excited states. The exponential parametrization represents a systematic way to improve upon the reference wavefunction which leads to a size-consistent solution, which will be described below. There is one more interesting property of this expansion. Recall the exponential series of the cluster operator and the energy expression:

$$E = \langle \Phi | \hat{H} (1 + \hat{T} + \frac{\hat{T}^2}{2!} + \frac{\hat{T}^3}{3!} + \dots) | \Phi \rangle. \quad (2.4)$$

Let us assume that the reference determinant  $\Phi$  is a single determinant constructed from one-electron functions. It is not difficult to note that the term  $\langle \Phi | \hat{H} \frac{\hat{T}^3}{3!} | \Phi \rangle$  is zero since determinants in bra- and ket-parts differ by more than two orbitals. The minimal excitation level that the  $\hat{T}^3$ -operator creates is three ( $\hat{T}^3 = \hat{T}_1 \hat{T}_1 \hat{T}_1 + \dots$ ).

## 2.2 Coupled Cluster theory

The wavefunction approximation reviewed so far, however, suffers from the inability to determine the amplitudes ( $t_i^a$ ,  $t_{ij}^{ab}$ , *etc.*) in the cluster expansion. The most obvious place to start is with the Schrödinger equation for the coupled cluster wavefunction:

$$\hat{H} e^{\hat{T}} |\Phi\rangle = E e^{\hat{T}} |\Phi\rangle. \quad (2.5)$$

Projecting onto the reference  $\langle \Phi |$  state and utilizing the orthogonality property one obtains:

$$\langle \Phi | \hat{H} e^{\hat{T}} |\Phi\rangle = E \langle \Phi | e^{\hat{T}} |\Phi\rangle = E \langle \Phi | \Psi_{CC} \rangle = E. \quad (2.6)$$

By projecting equation (2.5) onto excited determinants (produced by the action of the cluster operator  $\hat{T}$ ), we obtain the amplitude equations:

$$\langle \Phi_{ij\dots}^{ab\dots} | \hat{H} e^{\hat{T}} |\Phi\rangle = E \langle \Phi_{ij\dots}^{ab\dots} | e^{\hat{T}} |\Phi\rangle. \quad (2.7)$$

Projection on the determinant  $\Phi_{ij}^{ab}$  will produce an equation for the specific amplitude  $t_{ij}^{ab}$  coupled to the other amplitudes. The number of equations will match the number of possible determinants we can generate by excitation of the reference. Equation (2.6) is referred to as the *unlinked coupled-cluster amplitude equations*. Although, these equations are exact if there is no truncation in the operator  $\hat{T}$ , they are too complicated to work with since they are non-linear( $e^{\hat{T}}$ ) and energy dependent.

Large simplifications of the amplitude equations can be achieved by using the similarity-transformed Hamiltonian:

$$\hat{H} \rightarrow e^{-\hat{T}} \hat{H} e^{\hat{T}}.$$

Left-projection by the reference and excited determinants results in:

$$\langle \Phi | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi \rangle = E, \quad (2.8)$$

$$\langle \Phi_{ij...}^{ab...} | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi \rangle = 0 \quad (2.9)$$

Equation (2.9) is referred to as the *linked coupled-cluster amplitude equations*.

The first thing one notices is that the energy dependence disappeared from equation (2.9). We may therefore anticipate solving equations for the amplitudes and use equation (2.8) to obtain the energy afterwards. There is one more great simplification of the amplitude equation which not only saves labor but provides a better understanding of the equation structure. Before delving into the details here, we would like to discuss the legitimacy of the use of the similarity-transformed Hamiltonian. All physical observables are expectation values of the corresponding Hermitian operators, however, the operator  $e^{-\hat{T}} \hat{H} e^{\hat{T}}$  is not Hermitian. Even if we can formally use the similarity-transformed Hamiltonian due to the fact that transformations of this kind do not change the eigenvalues of the full Hamiltonian, it should be kept in mind that eigenvectors are not preserved. We refer to [34] for discussion about the usage of the similarity-transformed Hamiltonian.

Lets check if the linked form of the amplitude equations is equivalent to the unlinked one. Consider equation (2.9) by moving  $e^{-\hat{T}}$  into the bra-part:

$$\langle (e^{-\hat{T}})^* \Phi | (\hat{H} - E) e^{\hat{T}} | \Phi \rangle = 0, \quad (2.10)$$

$$\langle (e^{-\hat{T}})^* \Phi_{ij...}^{ab...} | (\hat{H} - E) e^{\hat{T}} | \Phi \rangle = 0. \quad (2.11)$$

Expanding the exponential operator into series we have:

$$\langle (1 - \hat{T}^* + \frac{1}{2}(\hat{T}^*)^2 - \dots) \Phi | (\hat{H} - E) e^{\hat{T}} | \Phi \rangle = 0, \quad (2.12)$$

$$\langle (1 - \hat{T}^* + \frac{1}{2}(\hat{T}^*)^2 - \dots) \Phi_{ij...}^{ab...} | (\hat{H} - E) e^{\hat{T}} | \Phi \rangle = 0. \quad (2.13)$$

Equation (2.8) follows from (2.12) immediately since  $\hat{T}^*|\Phi\rangle = 0$  ( $|\Phi\rangle$  has no particles/holes to annihilate). In Eq. (2.13) the bra-part will result in a linear combination of the reference  $\Phi$  and excited determinants  $\Phi_{ij\dots}^{ab\dots}$  with excitation level equal or less than the truncation level of the cluster operator which lead to a series of terms that vanishes. Now after proving the fact that the linked coupled-cluster equations are equivalent to the unlinked ones, the second benefit obtained by the similarity transformation of the Hamiltonian can be discussed. We start with a replacement of the exponential operators by a series of nested commutators using the Campbell-Baker-Hausdorff(CBH) formula:

$$e^{-\hat{T}}\hat{H}e^{\hat{T}} = \hat{H} + [\hat{H}, \hat{T}] + \frac{1}{2!}[[\hat{H}, \hat{T}], \hat{T}] + \frac{1}{3!}[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}] + \dots \quad (2.14)$$

Typically for fermionic systems, the Hamiltonian takes into account only two-body interactions,  $\hat{H}$  is at most a two-particle operator, corresponding to four creation/annihilation operators in second quantized form. The operator  $\hat{T}$  is at least a one-particle excitation operator. Applying Wick's theorem to Eq. (2.14), each commutator should lead to at least one contraction. Because  $\hat{T}$  is a particle-hole operator, the only surviving contractions are those with the Hamiltonian which in turn can be contracted at most four times. Hence, fourfold commutator term is the last non-zero term in the CBH-expansion, since all higher order terms necessarily vanish. Note that truncation follows from the form of the Hamiltonian only and does not depend upon the truncation level of  $\hat{T}$  or the number of particles in the system.

## 2.3 Variational Coupled Cluster

An alternative strategy can be to consider the cluster amplitudes as parameters and tune them in a such that:

$$E_{\text{exact}} \leq E_v = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\langle \Phi | (e^{\hat{T}})^* \hat{H} e^{\hat{T}} | \Phi \rangle}{\langle \Phi | (e^{\hat{T}})^* e^{\hat{T}} | \Phi \rangle},$$

is minimized according to the variational principle. Considerably more work is needed to determine the energy this way compared to the projective technique (2.8). It follows from the absence of a natural truncation of the power series expansion.

In contrast to the similarly transformed Hamiltonian  $e^{-\hat{T}}\hat{H}e^{\hat{T}}$ ,  $(e^{\hat{T}})^*\hat{H}e^{\hat{T}}$  does not truncate after a finite number of terms,

$$\langle \Phi | (e^{\hat{T}})^* \hat{H} e^{\hat{T}} | \Phi \rangle = \langle \Phi | (1 + \hat{T}^* + \frac{(\hat{T}^2)^*}{2!} + \frac{(\hat{T}^3)^*}{3!} + \dots) \hat{H} (1 + \hat{T} + \frac{\hat{T}^2}{2!} + \frac{\hat{T}^3}{3!} + \dots) | \Phi \rangle.$$

Indeed, the operator  $\hat{T}$ , acting to the right on  $|\Phi\rangle$  will produce an excited determinant and  $\hat{T}^*$  will similarly create an excitation of the  $\langle\Phi|$  and thus the Hamiltonian matrix elements will not vanish at higher excitations terms as we have seen for the BCH expansion.

Along with this complication for the construction of the variational coupled cluster theory, however, there are successful investigations of this approach by several researchers. We refer to the article by Szalay, Nooijen and Bartlett [37] for a review of variational coupled-cluster methods. Correlation in the results provided by variational coupled cluster methods for some realistic systems can be found in [8].

## 2.4 Algebraic approach

In this section we discuss the algebraic technique used to derive programmable coupled cluster amplitude equations. Widely used tools (second quantization, normal-ordering, contractions and Wick's theorem) are required to use successfully the algebraic approach. A brief introduction can be found in Appendix (A.1). However, a reader not familiar with many-body physics can find useful information from the following books [14, 7, 12, 36, 26].

Despite the substantial simplifications achieved through the use of the second quantization formalism in many body quantum physics and chemistry, certain equations require a great amount of efforts and labor to work with. Let's consider the typical quantity:

$$\langle p_1 p_2 | c_r^\dagger c_s | q_1 q_2 \rangle = \langle - | c_{p_1} c_{p_2} c_r^\dagger c_s c_{q_1}^\dagger c_{q_2}^\dagger | - \rangle. \quad (2.15)$$

To evaluate the value of (2.15) one can recall the anti-commutation relations. However the number of terms grows quickly, increasing the likelihood of making mistakes. Wick's theorem can drastically reduce calculations here if one uses a normal-ordered form of the operators. It is shown elsewhere, see for example [7], that the normal-ordered form of an operator is simply the operator itself minus its reference expectation value:

$$\hat{H}_N = \hat{H} - \langle \Phi_0 | \hat{H} | \Phi_0 \rangle = \hat{H} - E_{\text{Ref}}.$$

The normal-ordered Hamiltonian in a second-quantized form can be written as [14]:

$$\hat{H}_N = \hat{F}_N + \hat{V}_N = \sum_{pq} f_{pq} \{ c_p^\dagger c_q \} + \sum_{pqrs} w_{pq}^{rs} \{ c_p^\dagger c_q^\dagger c_s c_r \}. \quad (2.16)$$

The coupled cluster linked equations (2.9), (2.8) include products of Hamiltonian- and cluster operators. Let us for simplicity truncate the cluster operator at the level of two-particle-two-hole excitations:

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

Since the expectation value of the reference energy is just a number, the Hamiltonian of interest in equations (2.8) and (2.9) is on the similarity-transformed normal-ordered form. We denote it as  $\bar{H} = e^{-\hat{T}} \hat{H}_N e^{\hat{T}}$  and the BCH-expansion for it can be written as:

$$\bar{H} = \hat{H}_N + [\hat{H}_N, \hat{T}_1] + [\hat{H}_N, \hat{T}_2] + \frac{1}{2} [[\hat{H}_N, \hat{T}_1], \hat{T}_1] + \frac{1}{2} [[\hat{H}_N, \hat{T}_2], \hat{T}_2] + [[\hat{H}_N, \hat{T}_1], \hat{T}_2] + \dots, \quad (2.17)$$

where the cluster operators are already on the normal-ordered form:

$$\hat{T}_1 = \sum_{ia} t_i^a \{c_a^\dagger c_i\},$$

and,

$$\hat{T}_2 = \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \{c_a^\dagger c_b^\dagger c_j c_i\}.$$

Nothing new was introduced here, except for rewriting the Hamiltonian on a form that allows us to utilize the powerful machinery of Wick's theorem. One can think of it as just an optimization technicality to obtain programmable coupled-cluster equations.

Next we insert the BCH-expansion of the normal-ordered similarity transformed Hamiltonian (2.4) into the coupled cluster energy (2.8) and amplitudes (2.9) linked equations:

$$\langle \Phi | \bar{H} | \Phi \rangle = E, \quad (2.18)$$

and,

$$\langle \Phi_{ij\dots}^{ab\dots} | \bar{H} | \Phi \rangle = 0. \quad (2.19)$$

We consider the energy equation first.

$$\langle \Phi | (\hat{H}_N + [\hat{H}_N, \hat{T}_1] + [\hat{H}_N, \hat{T}_2] + \dots) | \Phi \rangle = E. \quad (2.20)$$

The leading term in equation (2.20) is the reference expectation value of the normal-ordered Hamiltonian, which is zero by definition of the normal ordering. The first commutator is expanded into the following expression according to (2.16):

$$[\hat{H}_N, \hat{T}_1] = [\hat{F}_N, \hat{T}_1] + [\hat{V}_N, \hat{T}_1] = \hat{F}_N \hat{T}_1 - \hat{T}_1 \hat{F}_N + \hat{V}_N \hat{T}_1 - \hat{V}_1 \hat{F}_N. \quad (2.21)$$

To evaluate  $\hat{F}_N \hat{T}_1$  between reference states we can apply the generalized Wick's theorem and rewrite it as a sum of fully contracted terms:

$$\langle \Phi | \hat{F}_N \hat{T}_1 | \Phi \rangle = \sum_{ia} t_i^a \sum_{pq} f_{pq} \langle \Phi | \{c_p^\dagger c_q\} \{c_a^\dagger c_i\} | \Phi \rangle. \quad (2.22)$$



All possible contractions between normal-ordered substrings  $\{c_p^\dagger c_q\}\{c_a^\dagger c_i\}$  are:

$$\begin{aligned} \{c_p^\dagger c_q\}\{c_a^\dagger c_i\} &= \{c_p^\dagger c_q c_a^\dagger c_i\} + \overbrace{\{c_p^\dagger c_q\}\{c_a^\dagger c_i\}} + \overbrace{\{c_p^\dagger c_q\}\{c_a^\dagger c_i\}} + \overbrace{\{c_p^\dagger c_q\}\{c_a^\dagger c_i\}} \\ &= \{c_p^\dagger c_q c_a^\dagger c_i\} + \delta_{pi}\{c_q c_a^\dagger\} + \delta_{qa}\{c_p^\dagger c_i\} + \delta_{pi}\delta_{qa}. \end{aligned} \quad (2.23)$$

For  $\hat{T}_1 \hat{F}_N$  there are no non-zero contractions:

$$\{c_a^\dagger c_i\}\{c_p^\dagger c_q\} = \{c_a^\dagger c_i c_p^\dagger c_q\}. \quad (2.24)$$

The commutator  $[\hat{F}_N, \hat{T}_1]$  from (2.21) simplifies to:

$$[\hat{F}_N, \hat{T}_1] = \sum_{aiq} f_{qi} t_i^a \{c_q^\dagger c_a\} + \sum_{iap} f_{pa} t_i^a \{c_p^\dagger c_i\} + \sum_{ia} f_{ia} t_i^a. \quad (2.25)$$

When evaluating this expression between reference states one immediately notes that due to the normal ordering ( $\langle \Phi | \{ \dots \} | \Phi \rangle = 0$ ) just one term will survive, namely:

$$\sum_{ia} f_{ia} t_i^a.$$

The second commutator  $[\hat{V}_N, \hat{T}_1]$  in equation (2.21) will not contribute to the energy equation since it is impossible to obtain fully-contracted terms simply because of the number of operators in substrings:

$$\hat{V}_N \hat{T}_1 \propto \{c_p^\dagger c_q^\dagger c_s c_r\} \{c_a^\dagger c_i\},$$

and,

$$\hat{T}_1 \hat{V}_N \propto \{c_a^\dagger c_i\} \{c_p^\dagger c_q^\dagger c_s c_r\}.$$

Using the approach described above all terms of equation (2.21) can be evaluated. There is one more simplification we need to mention – connectivity of operators. The only nonzero terms in the Hausdorff expansion are those in which the Hamiltonian has at least one contraction with every cluster operator on its right. In this case we say that the Hamiltonian and a sequence of cluster operators are connected. Using the procedure described above for evaluating the sequence of commutators in (2.20) between references one obtains the coupled cluster energy equation:

$$E_{CCSD} - E_0 = \sum_{ia} f_{ia} t_i^a + \frac{1}{4} \sum_{iajb} w_{ab}^{ij} t_{ij}^{ab} + \frac{1}{2} \sum_{iabj} w_{ij}^{ab} t_i^a t_j^b, \quad (2.26)$$

where we used the notation  $w_{ij}^{ab} = \langle ij || ab \rangle$  for anti-symmetrized two-body matrix elements. This equation is not restricted to the CCSD approximation, simply because cluster operators of level greater than two, do not create fully contracted

terms with the Hamiltonian. This follows from the form of Hamiltonian or number of creation/annihilation operator presented by inclusion of just two-body interactions. One uses the same energy equation for CCSDT and CCSDTQ approximations, where higher excitation cluster operators contribute through more precise parametrization of the wavefunction. In the amplitude equations (2.19) there are excited determinants in the bra-part:

$$\langle \Phi_i^a | \bar{H} | \Phi \rangle = 0,$$

$$\langle \Phi_{ij}^{ab} | \bar{H} | \Phi \rangle = 0.$$

And as a result, terms, vanishing in energy equation, begin to contribute. Excited determinants in the bra-part add a number of creation/annihilation operators. This results into the possibility to have fully-contracted strings of operators:

$$\langle \Phi_{ij}^{ab} | = \langle \Phi | \{c_i^\dagger c_j^\dagger c_b c_a\}. \quad (2.27)$$

The recipe to obtain linked coupled cluster amplitude equations is the same as described above: to insert the BCH-expansion of the normal-ordered similarity transformed Hamiltonian (2.4) into the linked amplitude (2.9) equations and take into account the excited bra-part (2.27). The only difference is that we have a greater number of contractions, and, as a result, a greater number of terms in the amplitude equation. For example the term  $\langle \Phi_{ij}^{ab} | \hat{H}_N | \Phi_0 \rangle$ :

$$\begin{aligned} \langle \Phi_{ij}^{ab} | (\hat{V}_N) | \Phi_0 \rangle &= \frac{1}{4} \sum_{pqrs} \langle pq || rs \rangle \langle \Phi_0 | \{c_i^\dagger c_j^\dagger c_b c_a\} \{c_p^\dagger c_q^\dagger c_s c_r\} | \Phi_0 \rangle = \\ &= \frac{1}{4} \sum_{pqrs} \langle pq || rs \rangle \left( \overbrace{\{c_i^\dagger c_j^\dagger c_b c_a c_p^\dagger c_q^\dagger c_s c_r\}}^{(ij)(ab)(pq)(rs)} + \overbrace{\{c_i^\dagger c_j^\dagger c_b c_a c_p^\dagger c_q^\dagger c_s c_r\}}^{(ij)(ab)(pq)(rs)} + \right. \\ &\quad \left. \overbrace{\{c_i^\dagger c_j^\dagger c_b c_a c_p^\dagger c_q^\dagger c_s c_r\}}^{(ij)(ab)(pq)(rs)} + \overbrace{\{c_i^\dagger c_j^\dagger c_b c_a c_p^\dagger c_q^\dagger c_s c_r\}}^{(ij)(ab)(pq)(rs)} \right) = \langle ab || ij \rangle. \end{aligned}$$

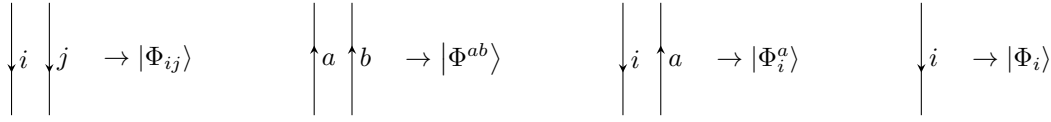
We will not present the derivation of the amplitude equations with the algebraic approach. Diagrammatic techniques offer a more practical approach to the construction of complicated coupled cluster equations allowing us to identify terms contributing to the amplitudes or the energy.

## 2.5 Diagrammatic approach

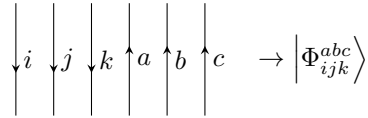
The described algebraic approach is simple in its application to some extent. The evaluation of operators between determinants becomes complicated for higher order excitations because the length of creation/annihilation operator strings becomes larger. This can be a possible source of errors. Below we present a more practical approach that provides us with a way to generate non-zero terms in a simple manner. The work-flow for such a procedure can be split up as follows:

1. We start with linked equations (2.8)-(2.9) as in the algebraic approach. And we try to draw separate operators, determinants as their diagrammatic analogies.
2. We try to merge together diagrammatic representations of the operators according to some simple rules.
3. We apply some simple algebraic rules, and the diagrams are read into expressions which are redundant to those obtained by Wick's theorem.

Similar to the described work-flow, we present material in the same order starting with the cornerstone of diagrammatic approach. That is the representation of determinant, vacuum, excitation operators, particle-hole formalism, components of the normal-ordered second quantized Hamiltonian. We use the diagrams populated by Kucharski and Bartlett, which are described in details in [7] and [14]. Hole- and particle-lines are shown in figure 2.1 together with singly-excited determinant. An example of a three-particle-three-hole determinant is presented in figure 2.2.



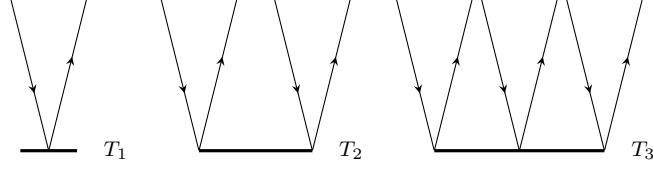
**Figure 2.1:** Diagrammatic representation of particle-hole formalism. Reference determinant is represented by an empty space. Orbitals differing from the reference are shown as upward- and downward-directed lines.



**Figure 2.2:** Diagrammatic representation of three-particle-three-hole determinant.

The following rules and symbols are used for dynamical operators:

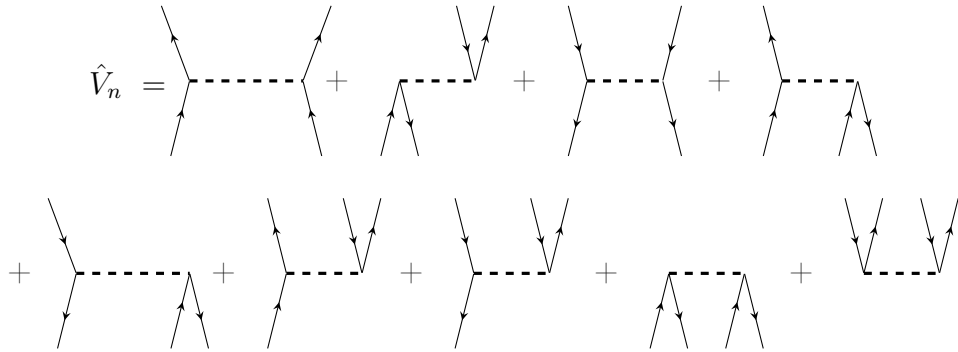
- Solid horizontal interaction line is used for cluster operators.
- Dashed horizontal interaction line is used for the Hamiltonian operators.
- Vertical lines represent creation/annihilation operator strings arising from vertices located on the horizontal interaction line.
- Creation operators lie above the interaction line, annihilation operators are below.



**Figure 2.3:** Diagrammatic representation of the cluster operators  $\hat{T}_1$ ,  $\hat{T}_2$  and  $\hat{T}_3$ .

According to the abovementioned rules, the cluster operators are easy to draw (figure 2.3). Since cluster operators excite determinants, or, in other words, contain only creation strings of operators, there are no lines below interaction line. The normal-ordered Hamiltonian in second-quantized form is written in (2.16). However we need to express it using the particle-hole formalism, where general indices  $p, q, \dots$  are replaced by particle-hole indices  $i, j, \dots a, b, \dots$ . The detailed derivation is provided elsewhere [21], we present just the final result. A two-body operator  $\hat{V}_N$  in the algebraic form is expressed in equation (2.28). The corresponding diagrammatic form is presented on figure 2.4.

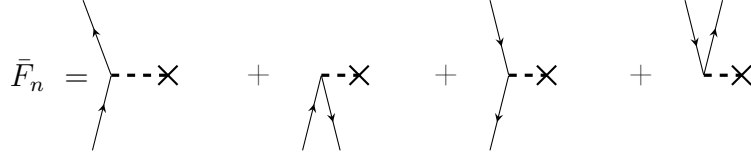
$$\begin{aligned}
 \hat{V}_N = & \frac{1}{4} \sum_{abcd} w_{cd}^{ab} \{\hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_d \hat{c}_c\} + \sum_{iabj} w_{bj}^{ia} \{\hat{c}_i^\dagger \hat{c}_a^\dagger \hat{c}_j \hat{c}_b\} + \frac{1}{4} \sum_{ijkl} w_{kl}^{ij} \{\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_l \hat{c}_k\} \\
 & + \frac{1}{2} \sum_{aibc} w_{bc}^{ai} \{\hat{c}_a^\dagger \hat{c}_i^\dagger \hat{c}_c \hat{c}_b\} + \frac{1}{2} \sum_{ijka} w_{ka}^{ij} \{\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_a \hat{c}_k\} + \frac{1}{2} \sum_{abci} w_{ci}^{ab} \{\hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_i \hat{c}_c\} \\
 & + \frac{1}{2} \sum_{iajk} w_{jk}^{ia} \{\hat{c}_i^\dagger \hat{c}_a^\dagger \hat{c}_k \hat{c}_j\} + \frac{1}{4} \sum_{abij} w_{ij}^{ab} \{\hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i\} + \frac{1}{4} \sum_{ijab} w_{ab}^{ij} \{\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_b \hat{c}_a\}
 \end{aligned} \quad (2.28)$$



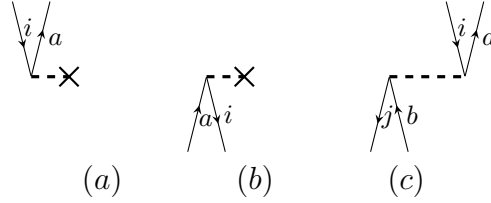
**Figure 2.4:** Diagrammatic representation of two-body operator  $\hat{V}_N$ . Each diagram term-by-term corresponds to the terms in equation (2.28).

The one-body part of the normal-ordered Hamiltonian is given by:

$$\hat{F}_N = \sum_{ab} f_{ab} \{\hat{c}_a^\dagger \hat{c}_b\} + \sum_{ia} f_{ia} \{\hat{c}_i^\dagger \hat{c}_a\} + \sum_{ij} f_{ij} \{\hat{c}_i^\dagger \hat{c}_j\} + \sum_{ia} f_{ai} \{\hat{c}_a^\dagger \hat{c}_i\} \quad (2.29)$$



**Figure 2.5:** Diagrammatic representation of one-body operator  $\hat{F}_N$ .



**Figure 2.6:** Diagrammatic representation of matrix elements of the operators between determinants. (a) –  $\langle \Phi_i^a | \hat{F}_N | \Phi \rangle$ , (b) –  $\langle \Phi | \hat{F}_N | \Phi_i^a \rangle$ , (c) –  $\langle \Phi_i^a | \hat{V}_N | \Phi_j^b \rangle$ .

The Matrix elements of the operators between different determinants are shown in figure 2.6.

The starting point for obtaining the coupled cluster energy expression is equation (2.20). However before combining different operators to evaluate terms like  $\hat{F}_N \hat{T}_1$  or  $\hat{V}_N \hat{T}_1$  between different determinants, it is practical to introduce *excitation level* concept which helps to construct only non-zero diagrams. The value of the excitation level can be obtained by subtraction of a number of the annihilation lines from a number of the creation lines and multiplying the result by the factor of  $\frac{1}{2}$ . For example, the diagrams on figure 2.5 have excitation levels 0 -1 0 +1. Let us consider equation (2.20). The leading term  $\langle \Phi | \hat{H}_N | \Phi \rangle$  is not possible to construct. The final diagram must contain no external lines. That is impossible according to figures 2.5- 2.4 and the fact that the reference vacuum is represented by an empty space. For the term  $\langle \Phi | \hat{F}_N \hat{T}_1 | \Phi \rangle$  we can easily merge together the second diagram on figure 2.5 with operator  $\hat{T}_1$  (figure 2.3) resulting in:

$$\langle \Phi | \hat{F}_N \hat{T}_1 | \Phi \rangle \rightarrow \text{diagram} \quad (2.30)$$

The excitation level of the resulting diagram must be zero because of the reference determinant in the bra-part. The operator  $\hat{T}_1$  gives an excitation level of +1, hence we need to choose diagrams with excitation level -1 for  $\hat{F}_N$ . Using either the excitation level rule or finding suited combination, the energy equation can be constructed more easily than in the algebraic approach:

$$E_{CC} = \text{diagram 1} + \text{diagram 2} + \text{diagram 3} \quad (2.31)$$

The diagram rules allow to read diagrams and to transform them into the algebraic expressions. These rules are listed as follows:

- All directed lines should be labeled with corresponding indices:  $i, j, k, \dots$  or  $a, b, c, \dots$
- Interaction lines contribute with an integral or an amplitude. The operators  $\hat{T}$  contribute amplitudes where indices are read in their left-to-right order in the diagram. One-body operator gives a factor  $f_{out,in}$ :

$$\text{diagram} \rightarrow t_{ij}^{ab} \quad (2.32)$$

$$\text{diagram} \rightarrow f_{ia}$$

Two-body operators provide a factor which should be read as  $\langle \text{left-out right-out} || \text{left-in right-in} \rangle$  for the outgoing lines from the  $\hat{V}_N$  diagram vertex:

$$\text{diagram} \rightarrow \langle ij || ab \rangle = w_{ij}^{ab} \quad (2.33)$$

- Summation goes over all internal lines. The internal lines are not extending to infinity below or above the diagram. (2.33) has four internal lines, and hence four summation indexes –  $ijkl$ .

- The sign of the diagram is determined according to:

$$(-1)^{h+l}, \quad (2.34)$$

where  $h$  is a number of hole-lines and  $l$  is a number of *loops*. A loop is defined as either a route along a series of directed lines that returns to its beginning, or a route that begins at one external line and ends at another.

- $n$  *equivalent vertices* provide factor  $\frac{1}{n!}$ . Such vertices should have the same number of line pairs and connect in the same way to the interaction vertex.

$$\begin{array}{c} \text{---} \text{---} \text{---} \\ \text{---} \text{---} \end{array} \begin{array}{c} i \\ a \end{array} \begin{array}{c} b \\ j \end{array} = \frac{1}{2} \sum_{ijab} w_{ij}^{ab} t_i^a t_j^b \quad (2.35)$$

- Each *equivalent* line-pair gives the factor of  $\frac{1}{2}$ . Two internal lines are defined to be equivalent if they point in the same direction and connect the same interaction lines.
- Each pair of *unique* external hole- or particle-lines introduces a permutation function  $P(pq)$ .

Basically, the same rules for the construction and reading of diagrams are applied to the amplitude equations. The amplitude equations have an excited determinant in the bra-part, and hence there are always a certain amount of creation lines above diagram. Moreover there are no annihilation lines below these diagrams.

The abovenentionedexcitation level rule can be used to verify the possibility of the construction of the diagrams. However one should be careful because of the growing number of different possibilities to connect operators. Diagrams must be topologically distinguishable which is not an easy task to determine by the eye. Normally, one needs to read both variants and check if the algebraic expressions cannot be obtained from each other by symmetric interchange of indices. Fortunately, Kucharsky and Barlett [7] developed a rule, helping to examine uniqueness of diagrams. This rule is also known as a *sign rule*.

We assign sign "+" to the particle lines, which lie below the operator line of the Hamiltonian or above the interaction line of the cluster operator. Oppositely, "-" sign is assigned to the hole lines satisfying mentioned condition. Then we connect lines of the same sign in unique ways to obtain diagram. As an example we consider operator string  $\hat{V}_N \hat{T}_1^2 \hat{T}_2$ . Each of the single excitation operators gives a pair of "+" "-" signs.  $T_2$  provides two such pairs, and the fragment of the Hamiltonian  $\hat{V}_N$  contributes two "-" and two "+" signs. By connecting operator lines preserving their sign we obtain the following sign structure for the diagrams on figure 2.7:

$$\underbrace{+-}_{\hat{T}_2} \underbrace{+}_{\hat{T}_1} \underbrace{-}_{\hat{T}_1}$$

By constructing such sign sequences one can test the equivalence of the diagrams.



**Figure 2.7:** Fragment of diagram corresponding to  $\hat{V}_N \hat{T}_1^2 \hat{T}_2$  operator string. Illustration of the sign rule.



# Chapter 3

## Quantum Monte Carlo Methods

Quantum Monte Carlo (QMC) methods are an accurate and powerful family of methods for treating quantum mechanical systems beyond the Hartree-Fock approach (normally called post Hartree-Fock methods). The essence of all Monte Carlo techniques is a reduction of computational efforts by converting sums over extremely large spaces into a series of discrete samples which approximate the full configuration space. Based on a desired accuracy, QMC methods can be computationally heavy, however at the same time these methods are easy to parallelize. QMC simulations often utilize results from traditional electronic structure methods like Hartree-Fock theory in order to increase the efficiency of the calculations. In this chapter we focus on the most used types of QMC methods, the so-called Variational Monte Carlo (VMC) and Projector Monte Carlo (PMC) approaches.

### 3.1 Variational Monte Carlo

In a quantum mechanical treatment of any system one is usually interested in expectation values of operators describing various observables. For some arbitrary operator  $\hat{O}$ , describing the observable it is possible to write the general expression:

$$\langle \Psi_0 | \hat{O} | \Psi_0 \rangle = \frac{\int d\mathbf{X} \Psi_0^*(\mathbf{X}) \hat{O}(\mathbf{X}) \Psi_0(\mathbf{X})}{\int d\mathbf{X} \Psi_0^*(\mathbf{X}) \Psi_0(\mathbf{X})}, \quad (3.1)$$

where  $\mathbf{X}$  – is a set of quantum numbers describing the chosen basis states or generally some basis set functions. The integrals in equation (3.1) are multi-dimensional ones and difficult to solve. This can however be simplified by introducing the so-called local operator:

$$\hat{O}_{loc}(\mathbf{X}) = \frac{1}{\Psi_0(\mathbf{X})} \hat{O} \Psi_0(\mathbf{X}). \quad (3.2)$$

Then equation (3.1) can be rewritten as:

$$\langle \Psi_0 | \hat{O} | \Psi_0 \rangle = \int d\mathbf{X} P(\mathbf{X}) \hat{O}_{loc}(\mathbf{X}), \quad (3.3)$$

where  $P(\mathbf{X})$  is the probability density to find the system in the state  $\mathbf{X}$ . It can be sampled stochastically using Monte Carlo method. This simple idea provides the basis for QMC studies of many-body systems.

We consider a system of particles. Usually the wavefunction of the system needed to solve the Schrödinger equation is unknown. So the first step is to choose an approximate function to be as realistic as possible. It is called the *trial wavefunction*. For a system of fermions a single Slater determinant is a good approximation. For the construction of the determinant one needs to choose single particle orbitals, which could be eigenstates of the harmonic oscillator or other well-known single-particle potentials or the orbitals coming from a Hartree-Fock calculation. More detailed analysis of the construction of the trial wavefunctions can be found elsewhere [16] [21]. Often the Hamiltonian does not contain terms acting on the spin and as a result particles preserve their initial states. In this case it is possible to express the Slater determinant as a product of the Slater determinants obtained from the single particle orbitals with different spins. Correlations are usually introduced by including a so-called Jastrow factor. The expectation value of the energy can be expressed as follows:

$$\langle E_T \rangle = \frac{\int d\mathbf{R} \Psi_T^*(\mathbf{R}) H(\mathbf{R}) \Psi_T(\mathbf{R})}{\int d\mathbf{R} \Psi_T^*(\mathbf{R}) \Psi_T(\mathbf{R})}, \quad (3.4)$$

where  $\Psi(\mathbf{R})$  is the trial wavefunction of the system and  $H(\mathbf{R})$  is the Hamiltonian, while  $E_T$  is the energy corresponding to the chosen wavefunction.  $\mathbf{R}$  stands for the spatial coordinates of all the particles  $(r_1, r_2, \dots, r_N)$ . However, the exact wavefunction is not known in most cases, and there is no clear rule on how to evaluate the quality of a trial wavefunction. The variational principle comes however to our rescue: principle:

$$E_0 \leq E_T = \frac{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \{\alpha, \beta, \dots\}) H(\mathbf{R}) \Psi_T(\mathbf{R}, \{\alpha, \beta, \dots\})}{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \{\alpha, \beta, \dots\}) \Psi_T(\mathbf{R}, \{\alpha, \beta, \dots\})}, \quad (3.5)$$

where the dependence of the trial wavefunction on some parameters  $\{\alpha, \beta, \dots\}$  is explicitly written out. As it has been already mentioned, for some operator  $\hat{O}$ , we define the local energy operator  $\hat{E}_L(\mathbf{R}, \alpha)$ :

$$\hat{E}_L(\mathbf{R}, \alpha) = \frac{1}{\Psi_T(\mathbf{R}) \Psi_T(\mathbf{R}, \alpha)} \hat{H} \Psi_T(\mathbf{R}) \Psi_T(\mathbf{R}, \alpha), \quad (3.6)$$

where  $\alpha$  denotes the whole set of the variational parameters  $\{\alpha, \beta, \dots\}$ , and equation (3.5) then becomes:

$$E_0 \leq E_T(\alpha) = \int d\mathbf{R} P(\alpha, \mathbf{R}) \hat{E}_L(\alpha, \mathbf{R}), \quad (3.7)$$

with the trial quantum-mechanical probability distribution function:

$$P(\boldsymbol{\alpha}, \mathbf{R}) = \frac{|\Psi_T(\boldsymbol{\alpha}, \mathbf{R})|^2}{\int d\mathbf{R} |\Psi_T(\boldsymbol{\alpha}, \mathbf{R})|^2}. \quad (3.8)$$

One can evaluate the value of the integral (3.7) with the Monte Carlo method:

$$E_T(\boldsymbol{\alpha}) = \int P(\mathbf{R}) E_L(\mathbf{R}) d\mathbf{R} \approx \frac{1}{N} \sum_{i=1}^N P(\mathbf{R}_i, \boldsymbol{\alpha}) E_L(\mathbf{R}_i, \boldsymbol{\alpha}), \quad (3.9)$$

where  $N$  is a number of Monte Carlo samples. The laws of statistics also provide us with algorithms to estimate statistical error, which is not dependent on the dimensionality of the integral and decreases with increasing the number of measurements. Then by applying a minimization method and varying parameters of the trial wavefunction one can iteratively calculate the integral (3.9). If the trial wavefunction is chosen to catch as much physics of the system as possible, the value of  $\langle E_T \rangle$  will approach the ground state energy  $\langle E_0 \rangle$ .

In functional minimization approaches such as the Hartree-Fock method, it is needed to minimize energy functional in respect to single particle functions constructing the total wavefunction of a system. In the VMC method one needs to minimize the energy functional with respect to a set of variational parameters to find the best trial wavefunction (combination of parameters) within some proper ansatz. The solution is therefore given by the solution of the Euler problem:

$$\frac{\delta E_T(\{\alpha, \beta, \dots\})}{\delta(\{\alpha, \beta, \dots\})} = 0. \quad (3.10)$$

It is required that we evaluate many times the integral (3.9) using different values of the variational parameters. A possible strategy here is to use a preliminary smaller number of Monte Carlo cycles to determine an optimal value for the variational parameters, and perform large-scale Monte Carlo calculation afterwards. The search for the variational minimum is a cumbersome part of the VMC method requiring a good knowledge of the system. The accuracy of the VMC results depends on the quality of the trial wave function.

Generally the VMC algorithm can be summarized as follows:

- Choose a trial wavefunction depending on variational parameters.
- Evaluate the expectation value of the Hamiltonian.
- Apply the minimization algorithm in the space of variational parameters.

We will not express the VMC method in fuller form in this thesis. But the topics needed to understand the difference between the QMC methods are covered. For detailed description of the VMC we refer to more specific texts on this topic [39] or [16].

### 3.1.1 The expectation value of the Hamiltonian

We present the simplest possible way to evaluate integral (3.9) with the Monte Carlo technique. In coordinate representation the states of the Hilbert space are sampled by sampling the particle positions from the squared modulus of the wavefunction. Thus stochastic integration can be split up into the following steps:

- Define the number of the Monte Carlo steps. Choose an initial ( $\mathbf{R}$ ) within fixed variational parameters  $\boldsymbol{\alpha}$  and calculate  $|\psi_T^\alpha(\mathbf{R})|^2$
- Start the Monte Carlo calculation by proposing a trial position  $\mathbf{R}_{new} = \mathbf{R}_{old} + r \cdot \mathbf{n}$ , where  $r$  is a uniformly distributed random variable ( $r \in [0, 1]$ ).  $\mathbf{n}$  is a step number. This is called brute-force sampling.
- Apply the Metropolis algorithm to accept or to reject proposed move simply by evaluating ratio between probabilities  $w = P(\mathbf{R}_{new})/P(\mathbf{R}_{old})$
- If the step is accepted then update  $\mathbf{R}_{old} = \mathbf{R}_{new}$ , otherwise the old position is used for the next move. Update average value of the local energy.
- Return to the first step.

When the Monte Carlo sampling is finished, the mean energy and the variance can be calculated.

### 3.1.2 The Metropolis algorithm

The basic idea of the Metropolis algorithm is to sample a normalized probability distribution by a stochastic process. One of the possible methods is to construct a Markov chain and to perform a random walk, where transition between states is determined by transition probability matrix  $W$ . The challenge here is that the transition probability is unknown in general. We approximate  $W_{i \rightarrow j}$  as the product of the probabilities  $T_{i \rightarrow j} A_{i \rightarrow j}$ .  $T_{i \rightarrow j}$  is the probability to make the move and  $A_{i \rightarrow j}$  is the probability that proposed move is successful. Let us denote via  $\mathbf{P}_i^{(n)}$  probability to find the system in the state  $i$  at step  $n$ . Then the Metropolis algorithm can be split up as follows:

1. Choose a new state  $j$  for the system with the probability  $T_{i \rightarrow j}$ .
2. Accept the new state  $j$  with the probability  $A_{i \rightarrow j}$  or reject this state with the probability  $1 - A_{i \rightarrow j}$ . The last case corresponds to the case when system remains in the initial state  $i$ .
3. Go back to the step (1) and propose a new transition.

The properties of  $T$  and  $A$  should lead to a process where  $\mathbf{P}_i^{(n \rightarrow \infty)}$  converges to the correct distribution  $\rightarrow p_i$ . The equation for  $P(n)_i$  can be written:

$$\mathbf{P}_i^{(n)} = \sum_j \left[ \mathbf{P}_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} + \mathbf{P}_i^{(n-1)} T_{i \rightarrow j} (1 - A_{i \rightarrow j}) \right] \quad (3.11)$$

The probability of being in the state  $i$  at step  $n$  is a sum of the probabilities. They are the probability of being in the state  $j$  at the  $n - 1$  step followed by accepted transition to  $i$  and the probability of being in the state  $i$  followed by the rejected move. The normalization requirement for probabilities allows us to rewrite this equation as follows:

$$\mathbf{P}_i^{(n)} = \mathbf{P}_i^{(n-1)} + \sum_j \left[ \mathbf{P}_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - \mathbf{P}_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} \right] \quad (3.12)$$

By moving  $\mathbf{P}_i^{(n-1)}$  term to the left, one gets time-evolution equation for  $\mathbf{P}_i$ :

$$\mathbf{P}_i^{(n)} - \mathbf{P}_i^{(n-1)} = \sum_j \left[ \mathbf{P}_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - \mathbf{P}_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} \right], \quad (3.13)$$

where  $\mathbf{P}_i^{(n)} - \mathbf{P}_i^{(n-1)}$  can be considered as analog to  $\frac{d\mathbf{P}_i}{dt}$  for an iterative discrete process. The last equation is similar to the master equation, and the left-hand side of it describes rate of systems movement from state  $i$  toward the final state  $j$ . In equilibrium this term should be zero  $\frac{d\mathbf{P}_i}{dt} = 0$ . And that leads to the so-called balance requirement:

$$\sum_j [p_j T_{j \rightarrow i} A_{j \rightarrow i} - p_i T_{i \rightarrow j} A_{i \rightarrow j}] = 0. \quad (3.14)$$

However, a more strict detailed balance requirement is enforced to guarantee that we finally generate the correct distribution. It results in the avoidance of so-called cyclic solutions[16]:

$$\frac{A_{j \rightarrow i}}{A_{i \rightarrow j}} = \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}}. \quad (3.15)$$

Then we can estimate the transition probability via the following expression:

$$A_{j \rightarrow i} = \min \left( 1, \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}} \right). \quad (3.16)$$

For the VMC method described above acceptance of such move, corresponding to the changing coordinate  $\mathbf{R}$  to  $\mathbf{R}'$  can be expressed as:

$$P(\mathbf{R} \rightarrow \mathbf{R}') = \min \left( 1, \frac{\Psi_T(\mathbf{R}') T(\mathbf{R} \rightarrow \mathbf{R}')}{\Psi_T(\mathbf{R}) T(\mathbf{R} \rightarrow \mathbf{R}')} \right), \quad (3.17)$$

where  $T(\mathbf{R} \rightarrow \mathbf{R}')$  is the transition probability to move from  $\mathbf{R}$  to  $\mathbf{R}'$  and originally is chosen to be uniformly random distributed variable over the interval  $\Delta\mathbf{R}$ . Generally, the described procedure, that uses homogeneously distributed random points will most likely lead to poor numerical efficiency. This allows us to suggest that regions where the wave function has greater values are sampled more frequently. To implement this we need to take into account our knowledge about the wavefunction when it is decided whether to accept or to reject the move to the new point in the configuration space. A common approach for improvement is to use the Fokker-Planck equation and the the Langevin equation for generating a trajectory [40]. This results in change of transition probability to:

$$T(\mathbf{R} \rightarrow \mathbf{R}') = \frac{1}{\sqrt{4\pi D\delta\tau}^{3N}} \exp\left[-\frac{(\mathbf{R}' - \mathbf{R} - \frac{1}{2}\mathbf{F}_q(\mathbf{R})\delta\tau)^2}{2\delta\tau}\right], \quad (3.18)$$

where  $N$  is the dimensionality of the system, the quantity  $D$  is, in atomic units,  $\frac{1}{2}$  and comes from similarities between diffusion equation and kinetic term in Schrödinger equation.  $\delta\tau$  is a small parameter and  $\mathbf{F}_q$  is so called *quantum force*, usually defined as:

$$\mathbf{F}_q = \nabla \ln|\Psi(\mathbf{R})|^2 \quad (3.19)$$

## 3.2 The Projector Monte Carlo

The accuracy of the VMC method depends on the quality of the trial wave function. This limitation can be overcome with the projector methods.

In linear algebra the power method helps to find a strictly dominant eigenvalue that is an eigenvalue which is larger in absolute value of magnitude than all the other eigenvalues. This is achieved by an iterative process of the matrix-vector multiplication[22]. The PMC methods enforces a stochastic implementation of the power method for determining the expectation value of an operator for the dominant eigenstate. Exact diagonalization methods like Lanczos or Davidson are the algorithms of choice in case if it is feasible to work with large matrices. However for systems with a high number of particles it is impossible to apply the abovementioned diagonalization methods due to memory limitations.

In 1930s Fermi noticed similarities between the imaginary time Schrödinger equation and the diffusion process. Fermi's ideas were acknowledged by Metropolis and Ulam in 1949, and they used a stochastic approach for the numerical solution of various problems. To our knowledge the first application to quantum mechanics was performed by Kalos and co-workers in the 1960s, see Anderson [1] in 1975 for the calculation of ground state of  $H_3^+$  and references therein.

The PMC methods use the advantage of that the exact energy can be written as a mixed expectation value:

$$E_0 = \frac{\langle \Psi_0 | \hat{H} | \Psi_T \rangle}{\langle \Psi_0 | \Psi_T \rangle}. \quad (3.20)$$

Contrary to the VMC approach where all quantities appearing in the calculations are explicitly known, one sees that in the PMC method, the expectation value of the energy contains the exact wavefunction (3.20) which is generally unknown. Even so, it still can be sampled with a repeated application of the so-called projector-function of the Hamiltonian with one eigenvalue equal to one and all others are less than one in absolute magnitude.

The imaginary-time Schrödinger equation is a starting point of all PMC methods. We introduce the Wick rotation of time to the standard time-dependent Schrödinger equation:

$$-i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle. \quad (3.21)$$

By defining imaginary time as  $\tau = \frac{it}{\hbar}$  one obtains the imaginary-time dependent Schrödinger equation:

$$-\frac{\partial}{\partial \tau} |\Psi(\tau)\rangle = \hat{H} |\Psi(\tau)\rangle, \quad (3.22)$$

Given a starting wavefunction  $|\Psi(\tau=0)\rangle$ , the wavefunction for arbitrary  $\tau$  can be expressed with a propagator:

$$|\Psi(\tau)\rangle = e^{-\tau \hat{H}} |\Psi(\tau=0)\rangle. \quad (3.23)$$

If we consider some arbitrary basis which can be discrete or continuous  $\{\phi_i\}$ , the wavefunction in this basis is written as follows:

$$|\Psi(\tau)\rangle = e^{-\tau \hat{H}} |\Psi(\tau=0)\rangle = e^{-\tau \hat{H}} \sum_i c_i |\phi_i\rangle = \sum_i c_i e^{-\tau E_i} |\phi_i\rangle. \quad (3.24)$$

In the limit with  $\tau \rightarrow \infty$ , coefficients  $e_i e^{-\tau E_i}$  will decrease for the positive values of  $E_i$  or increase for the negative values of  $E_i$ . It is easy to see that in long imaginary time limit coefficient corresponding to the ground state will dominate. Often a constant of proportionality  $e^{-\tau S}$  is introduced to keep the ground-state contribution from tending to zero:

$$|\Psi(\tau)\rangle = \lim_{\tau \rightarrow \infty} \sum_i c_i e^{-\tau(E_i - S)} |\phi_i\rangle.$$

In the example above the exponential projector, also known as imaginary-time propagator,  $\hat{P}(\tau)$  was used:

$$\hat{P}(\tau) = e^{-\tau(\hat{H} - \hat{1}S)}.$$

Thus the ground state wavefunction is obtained by an iterative application of a projection operator for a sufficient number of times:

$$|\Psi(\tau)\rangle = \lim_{\tau \rightarrow \infty} \hat{P}(\tau) |\Psi(\tau = 0)\rangle. \quad (3.25)$$

Depending on a particular PMC method different types of imaginary-time propagators can be used. A class of these are the so-called linear projectors:

$$\hat{P}(\tau) = \hat{1} - \tau(\hat{H} - \hat{1}S), \quad \hat{P}(\tau) = \frac{1}{\hat{1} - \tau(\hat{H} - \hat{1}S)}.$$

For high-dimensional vectors that are typically encountered in many-body quantum mechanics one needs to reformulate the power method in a stochastic form. That is possible to do by contracting the Markov Chain from states obtained by repeated application of the projector operator. In this case equation (3.25) is rewritten as:

$$|\Psi(\tau)\rangle = \lim_{n \rightarrow \infty} \hat{P}^n |\Psi(0)\rangle, \quad (3.26)$$

where  $|\Psi(0)\rangle$  is a starting ansatz for wavefunction,  $\hat{P}^n = e^{-\tau n(\hat{H} - S)}$  and  $n$  is a positive integer number.

The PMC methods can differ in many details and implementations, but they all have certain features in common and can be categorized accordingly. The first feature is the form of the projector. Earlier we gave an example of the linear and the exponential projector. The second attribute is the basis type. Coordinate- or orbital occupation spaces represent space where a random walk is performed. Lastly we need to consider the existence of the permutation symmetry. Second quantized states are used as (anti)symmetrized basis states. For example, the Diffusion Monte Carlo (DMC) method uses first quantized basis states in real space and utilizes exponential projector together with the Green Function Monte Carlo (GFMC), while the Full Configuration Monte Carlo (FCIQMC) [5] [6] and the Coupled Cluster Monte Carlo (CCMC) [33] [38] methods operates with linear projectors in an occupation number space of orthogonal orbitals, and the basis states are antisymmetrized.

### 3.3 The PMC general algorithm

We summarize the PMC algorithm in a few steps for coordinate space calculations in order to illustrate how it can be implemented.

By definition a Markov chain is a sequence of states where each state depends only on a previous one. And the transition from one state ( $S_i$ ) to another ( $S_j$ ) is defined by the probability from the stochastic transition matrix  $T_{ij}$  which should be normalized ( $\sum_j T_{ij} = 1$ ) and be non-negative. In the coordinate space representation a stochastic propagator should satisfy:



$$\int d\mathbf{R}' T(\mathbf{R}, \mathbf{R}') = 1 \quad (3.27)$$

The imaginary-time propagator  $\hat{P}$  is not generally normalized. To preserve the normalization it can be factored out as follows:

$$P_{ij} = T_{ij} w_j, \quad (3.28)$$

where  $w_j = \sum_i P_{ij}$  are elements of the weights vector  $W(\mathbf{R})$ . The Markov chain is formed by the states obtained from the transitions between states ( $i \rightarrow j$ ) with a probability defined by the stochastic matrix  $T$ . This process can be formulated as follows:

$$\begin{aligned} S(\mathbf{R}') &= \int d\mathbf{R}'' T(\mathbf{R}'', \mathbf{R}') S(\mathbf{R}'') \\ S(\mathbf{R}) &= \int d\mathbf{R}' T(\mathbf{R}', \mathbf{R}) S(\mathbf{R}') \\ &\dots \end{aligned}$$

The sequence of states  $\{\dots, S(\mathbf{R}'''), S(\mathbf{R}''), S(\mathbf{R}')\}$  forms the Markov chain. In the PMC methods we need to sample action of  $\hat{P}|\Psi_T\rangle$ . We perform a random walk and keep a vector  $W(\mathbf{R})$  to store all walkers:

$$\begin{aligned} \Psi(\mathbf{R}') &= S(\mathbf{R}') W(\mathbf{R}') = \int d\mathbf{R}'' P(\mathbf{R}'', \mathbf{R}') \Psi(\mathbf{R}'') = \int d\mathbf{R}'' T(\mathbf{R}'', \mathbf{R}') B(\mathbf{R}') \Psi(\mathbf{R}'') \\ \Psi(\mathbf{R}) &= S(\mathbf{R}) W(\mathbf{R}) = \int d\mathbf{R}' P(\mathbf{R}', \mathbf{R}) \Psi(\mathbf{R}') = \int d\mathbf{R}' T(\mathbf{R}', \mathbf{R}) B(\mathbf{R}) \Psi(\mathbf{R}') \\ &\dots, \end{aligned}$$

where  $B(\mathbf{R})$  determines the weight of the configuration at each step of the Markov chain. When it is not suitable to store the whole vector in memory (for example, due to memory limitations), one can store the most important components of it using some importance coefficients. These coefficients can have integer or real values and can be considered as a population of fictitious particles. In that case population dynamics represents tuning of the importance coefficients. This leads to the generalized stochastic PMC with not normalized transition probabilities, the so-called Population Monte Carlo (PopMC) algorithm. We will discuss it later in this thesis in connection to the coupled cluster Monte Carlo method. The PopMC PMC algorithm is presented here:

- Generate initial distribution of  $n$  walkers at coordinates  $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n$ . These points are intended for sampling the probability density function.

- Move each walker independently to a new position  $\mathbf{R}_i \rightarrow \mathbf{R}'_i$  according to probability  $T(\mathbf{R}, \mathbf{R}')$  and perform the Metropolis test. There is a possibility to use more numerically efficient proposal for moves and as a result Metropolis-Hastings test, which includes drifting Gaussian into the transition probability.
- Compute the weighting vector  $w_i = B(\mathbf{R}_i)$  and update weight for each walker multiplying by its value from the previous step.

$$w_i^{actual} = w_i w_i^{previous} \quad (3.29)$$

Ideally  $B(\mathbf{R}_i)$  should preserve the normalization of  $\hat{P}(\mathbf{R} \rightarrow \mathbf{R}')$  and hold  $w_i$  close to unity.

- Clone walkers with a large weight into multiple walkers. Delete from population walkers with small weights. This step is also referred in literature as branching process.

### 3.4 Diffusion Monte Carlo

We choose a coordinate representation of the wavefunction to look more closely into the DMC algorithm. We start with the solution of the imaginary-time Schrödinger equation:

$$|\Psi(\tau)\rangle = e^{-\tau(\hat{H}-E_0)} |\Psi(\tau=0)\rangle, \quad (3.30)$$

where  $E_0$  is the ground-state energy, or, more strictly, the best estimation of it at the start of the calculations. In coordinate representation the expression becomes:

$$\langle R|\Psi(\tau)\rangle = \int dR' \langle R|e^{-\tau(\hat{H}-E_0)}|R'\rangle \langle R'|\Psi(\tau=0)\rangle. \quad (3.31)$$

Let us consider the propagator  $\langle R|e^{-\tau(\hat{H}-E_0)}|R'\rangle$  and split the time interval into two equal ones ( $\tau/2$ ). Since the Hamiltonian commutes with itself:

$$\langle R|e^{-\tau(\hat{H}-E_0)}|R'\rangle = \langle R|e^{-\tau/2(\hat{H}-E_0)}e^{-\tau/2(\hat{H}-E_0)}|R'\rangle. \quad (3.32)$$

By continuing this process up to  $M$  times we obtain:

$$\langle R|e^{-\tau(\hat{H}-E_0)}|R'\rangle = \quad (3.33)$$

$$\int \dots \int dR' \dots dR^M \langle R|e^{-\tau/2(\hat{H}-E_0)}|R''\rangle \dots \langle R^M|e^{-\tau/2(\hat{H}-E_0)}|R'\rangle. \quad (3.34)$$

We see that inside the integrand factor represents a short imaginary-time propagation  $\Delta\tau = \tau/M$ . The propagator can be split using the Trotter-Suzuki formula:

$$e^{-\frac{\tau}{2M}(\hat{H}-E_0)} = e^{-\frac{\tau}{2M}(\hat{V}-E_0)} e^{-\frac{\tau}{M}\hat{T}} e^{-\frac{\tau}{2M}(\hat{V}-E_0)} + \mathcal{O}\left(\left(\frac{\tau}{2M}\right)^3\right), \quad (3.35)$$

where it is easy to recognize the solution of the diffusion equation in the kinetic term of the propagator. Indeed, there is an isomorphism between the diffusion equation and the kinetic energy term in the imaginary-time Schrödinger equation:

$$-\frac{\partial}{\partial \tau} |\Psi(\tau)\rangle = \left( \underbrace{\frac{1}{2} \nabla^2}_{\text{diffusion}} - \underbrace{(\hat{V} - E_0)}_{\text{branching}} \right) |\Psi(\tau)\rangle, \quad (3.36)$$

which is very similar to the diffusion equation with the first-order rate term used to describe chemically reactive systems:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - kC, \quad (3.37)$$

where  $D$  is a diffusion constant and  $C$  is the concentration of particles. The second term of this equation can be interpreted as a presence of a source or a sink of particles. The particle can disappear with probability proportional to  $k$  or oppositely be created with probability proportional to  $-k$ . Strictly speaking, the imaginary-time Schrödinger equation is not a diffusion equation since time is imaginary and the wavefunction is not a probability distribution function unless it is squared. However, the similarity between these equations can help us in understanding how to apply the MC method to the imaginary-time Schrödinger equation. Without the reactive term one gets a *free diffusion* equation:

$$-\frac{\partial}{\partial \tau} |\Psi(\tau)\rangle = \frac{1}{2} \nabla^2 |\Psi(\tau)\rangle, \quad (3.38)$$

where we omitted the  $R$ -dependence of the wavefunction and  $\frac{\hbar^2}{m} = 1$ .

For the wavefunction in the coordinate representation it's quite easy to implement sampling. We consider just a diffusion term of equation (3.36) and the whole ensemble of points  $R_{\{i\}}$  as the population of walkers. All walkers have some initial distribution, and after a short imaginary time interval each walker takes an independent random step sampled from the Gaussian distribution.

Now, if we consider just a so-called branching term, corresponding to a population of walkers in (3.36), the solution will be:

$$\Psi(X_{\{i\}}, \tau) = e^{-\tau(V(X_{\{i\}}) - E_0)} \Psi(X_{\{i\}}, \tau = 0), \quad (3.39)$$

which is nothing else but the potential term of the propagator. It is similar to the radioactive decay equation, representing the evolution in time of some particle population. The Monte Carlo algorithm for this part can be expressed as follows:

$$w_k^{(\tau+\delta\tau)} = w_k^{(\tau)} e^{-\delta\tau(V(X_{\{i\}}) - E_0)} \Psi(X_{\{i\}}, \tau = 0), \quad (3.40)$$

where the weight  $w_k$  is associated with each walker ( $w_k^{(0)} = 1$ ) and updated according to (3.40). However, the above mentioned way has a slight bias if the

population is not infinite [2]. Instead, one can introduce renormalized individual walker weights. These can be replaced by a stochastic birth/death process where a walker with weight  $> 1$  gives birth to the copy of the walker, otherwise the walker is suppressed. We discuss this topic in more details in the section dedicated to the population control stochastic algorithms.

The diffusion term and the branching term provide exact solutions to their differential equations. However we need to combine these solutions to solve the reactive diffusion equation, for example by applying sampling for two processes during one step. It has already been shown how to split diffusion and branching using the Trotter-Suzuki formula (3.35), but it was not mentioned that separation introduces time-step bias  $\mathcal{O}(\delta\tau^3)$ . To avoid time-step bias DMC calculations with smaller time steps can be used.

The straightforward procedure which uses the Coulomb potential as a population dynamics driven force most likely leads to the large fluctuations of the population. It has been suggested that the importance sampling may stabilize the population of walkers. The main idea here is to multiply the probability distribution by the trial wavefunction which in turn should carry information about the system. This leads to the simulation of the diffusion equation with a drift term, that moves walkers towards the regions of high value of the wavefunction. The rate process will be governed more intensively by the local energy in this case.

We will not dig into technicalities of DMC such as the Fixed-Node approximation, the MC sign problem, importance sampling etc. All these are described in the special texts devoted exclusively to the DMC. (See for example Refs. [19, 3, 16, 39, 2, 20])

### 3.5 Full Configuration Interaction Quantum Monte Carlo

In the Full Configuration Interaction (FCI) method wavefunction is approximated by a linear combination of the Slater determinants  $\{|D_i\rangle\}$ :

$$\Psi_0^{FCI} = \sum_i C_i |D_i\rangle, \quad (3.41)$$

where  $|D_i\rangle$  is defined as:

$$|D_i\rangle = |D_{1,2,\dots,N}\rangle = \hat{c}_1^\dagger \hat{c}_2^\dagger \dots \hat{c}_N^\dagger |-\rangle, \quad (3.42)$$

and coefficients  $C_i$  satisfy the following:

$$\sum_j \langle D_i | H | C_j \rangle C_j = E_0^{FCI} C_i. \quad (3.43)$$

As it was already mentioned one possible way to obtain ground state wavefunction is to solve the imaginary-time Schrödinger equation (3.22) by applying the projection operator (3.26) sufficient(large) number of times. We express initial state in the FCI basis:

$$\Psi(\tau) = \sum_i C_i(\tau) |D_i\rangle. \quad (3.44)$$

Let us define matrix  $K$ , where the matrix elements are the matrix elements of the Hamiltonian between some arbitrary slater determinants. In addition the Hartree-Fock energy is subtracted from the diagonal elements:

$$K_{ij} = \langle D_i | H | D_j \rangle - E_{HF} \delta_{ij} = H_{ij} - E_{HF} \delta_{ij}. \quad (3.45)$$

It is easy to see that the lowest eigenvalue of the matrix  $K$  is  $(E_0^{FCI} - E_{HF})$ , which is nothing else but the correlation energy. Taking into account the fact that the FCI wavefunction is a linear combination, one obtains the true wavefunction corresponding to the ground state by tuning the expansion coefficients in the long imaginary-time limit(3.44). We substitute expression (3.44) into the imaginary-time Schrödinger equation:

$$-\frac{dC_i}{d\tau} = \sum_j (K_{ij} - S\delta_{ij})C_j = (K_{ii} - S)C_i - \sum_{j \neq i} K_{ij}C_j, \quad (3.46)$$

where  $S$  is an arbitrary energy shift. If all  $C_i$  satisfy

$$\sum_j K_{ij}C_j = SC_i, \quad (3.47)$$

then  $C_i$  are not changing with  $\tau$ . In other words by finding the vector  $C$  with the components such as  $\frac{dC_i}{d\tau} = 0$  and taking into account that  $S$  and  $C$  correspond to the eigenvalue and the eigenvector of  $K$  (3.47), we find the correlation energy to be equal to  $S$ , and stationary  $C$  corresponding to the ground state of  $H$ . The solution of equation (3.46) with the deterministic methods requires storing in memory the whole vector  $C$ . However it can be difficult because of an extremely large number of excited determinants. Alavi et al. [6] proposed the approach similar to DMC: to replace the vector  $C$  with a population of walkers such that each walker is located at the determinant  $i$  and carry sign  $S_\alpha = \pm 1$ . Then the coefficient  $C_i$  at determinant  $|D_i\rangle$  is approximated as follows:

$$C_i \approx N_i = \sum_\alpha s_\alpha \delta_{i,i_\alpha}. \quad (3.48)$$

The population of each walker is changing during the imaginary time evolution and is guided by the equation:

$$N_i(\tau + \delta\tau) = N_i(\tau) - \delta\tau(K_{ii} - S)N_i(\tau) - \delta\tau \sum_{i \neq j} K_{ij}N_j(\tau), \quad (3.49)$$

which can be sampled stochastically. Typically one  $\delta\tau$  iteration of the FCIQMC algorithm has the following steps:

- **Spawn.** For each walker on the determinant  $|D_i\rangle$ (parent) we randomly choose a determinant  $\langle D_j|$ (child) and try to spawn walker of appropriate sign with the probability proportional to  $K_{ij}$ .
- **Death.** For each parent determinant the walker located on this particular determinant is suppressed with the probability proportional to  $(K_{ii} - S)$ . Negative values of  $K_{ii} - S$  lead to rare clone processes.
- **Annihilation.** The signed population of the walkers is merged with the population from the previous  $\tau$ -iteration, so that the walkers of different signs cancel each other.

### 3.6 The Population Monte Carlo Algorithm

Population Monte Carlo(PopMC) algorithms are a family of algorithms used in various scientific fields. The main idea is to represent a high-dimensional vector with an ensemble of virtual fictitious particles or walkers. The simulation is carried out by a random walk in the state space associated by cloning and death events. It is worth mentioning that the PopMC is not a canonical Monte Carlo Markov Chain because the transition probabilities are generally not normalized. As we will see below the propagator in the PMC method is not normalized, and hence we need some weighting function to factor the propagator into a normalized probability. We consider the following algorithm for the computation of the product of non-negative matrices  $G^1, G^2, \dots$  and a non-negative vector  $X^{(0)}$ . On the  $n$ -th step vector becomes:

$$X^{(n)}(i) = \sum_j G^n(i, j)X^{(n)}(j) \quad (3.50)$$

If the dimension of the vector  $X$  is extremely large, so we cannot even save it in computer memory. That is why we make an approximation, where we represent the vector as a weighted superposition of walkers(particles) with index  $k$ . Each of them is placed at  $j(k)$  with a weight coefficient  $w_k^n$ :

$$X^{(n)}(j) \approx \sum_k w_k^n \delta_{j^{(k)}, j}, \quad (3.51)$$

where  $\delta_{j^{(k)}, j}$  is the Kronecker delta. The algorithm can be expressed as follows:

1. Each walker  $k$  at the position  $j^{(n)}(k)$  is moved independently to a new position  $i$  with a transition probability:

$$P^{(n)}(j^n(k) \rightarrow i) = \frac{G^n(i, j^n(k))}{\sum_{i'} G^n(i', j^n(k))} \quad (3.52)$$

At this step we can ensure detailed balance by using the Metropolis algorithm.

2. Update weights for each walker, for example in the following manner:

$$\log w_k^{n+1} = \log W_k + \log w_k^n, \quad (3.53)$$

where

$$W_k = \sum_{i'} G^n(i', j^n(k)). \quad (3.54)$$

It can be useful to have a weight function. In this case  $w_k^n = B(j^{(n)}(k))$ , where  $B$  is a weight function which preserves normalization and maintains individual weights close to unity. So the weight of walker will be a product of its weights on the previous and the current iterations.

3. Split walker with greater weights and remove walkers with small weight.

## 3.7 The sign problem

The sign problem is particular for the case when the PMC methods deal with fermionic systems. It leads to an exponential growth of computational costs with increasing system size. The sign problem has a different nature for PMC methods. We consider briefly the DMC sign problem followed by discussion of the FCIQMC sign problem.

### The DMC sign problem

The wavefunction for the systems of fermions must have alternating sign to satisfy the antisymmetry property. Regions with different signs are separated by a so-called nodal surface<sup>1</sup>. Usually we do not know exactly the nodal surface of the ground state, and hence we cannot construct a wavefunction with the correct nodal structure to be used as some kind of guidance or importance sampling in the DMC calculations. The obvious idea is to split the wavefunction into positive and negative parts, which can be sampled stochastically. It does however not solve this problem either, since the solutions will cancel each other and lead to the dominance of stochastic noise.

---

<sup>1</sup>Regions where wavefunction has one definite sign are referred as nodal cells or nodal pockets

### The FCIQMC sign problem

It might appear that the calculation in an antisymmetrized basis can solve sign problem as there are no other states than fermionic. Of course it is not true. The FCIQMC method has a major difference compared with the DMC method. This lies in the fact that walkers have a sign. Booth et al. [6] showed that the annihilation process plays the key role in the FCIQMC. Without the annihilation process the energy does not converge to the ground state true energy. Later Spencer et al. [32] created a model to explore the sign problem. We will provide key steps of Spencer's derivation here followed by our discussion of the sign problem in the PMC methods. Assume first that annihilation is not important. Hence without it the algorithm will converge to the true ground state energy. Without the annihilation we allow walkers with different sign to populate the same determinant without any interactions. We denote  $n_i^-$  and  $n_i^+$  population of negative and positive walkers at determinant  $D_i$ . An imaginary-time dynamics of these populations is described by following equations:

$$\frac{dn_i^+}{d\tau} = \sum_j (T_{ij}^+ n_j^+ + T_{ij}^- n_j^-), \quad (3.55)$$

and,

$$\frac{dn_i^-}{d\tau} = \sum_j (T_{ij}^+ n_j^- + T_{ij}^- n_j^+). \quad (3.56)$$

Where  $T_{ij}$  are the elements of the transition matrix  $T$ , which are defined as:

$$T = T^+ - T^- = S - \hat{H}, \quad (3.57)$$

where the matrix elements  $T_{ij}^+ = \max(T_{ij}, 0)$  and  $T_{ij}^- = |\max(-T_{ij}, 0)|$ .

Taking the sum of equations (3.55) and (3.56) we obtain an equation that describes the population evolution in time without annihilation:

$$\frac{d(n_i^+ + n_i^-)}{d\tau} = \sum_j (T_{ij}^+ + T_{ij}^-)(n_j^+ + n_j^-), \quad (3.58)$$

Subtraction of (3.56) from (3.55) leads to the case with the annihilation:

$$\frac{d(n_i^+ - n_i^-)}{d\tau} = \sum_j (T_{ij}^+ - T_{ij}^-)(n_j^+ - n_j^-). \quad (3.59)$$

$(n_i^+ + n_i^-)$  will tend to eigenvector corresponding to the largest eigenvalue of  $(T_{ij}^+ + T_{ij}^-)$ , while  $(n_i^+ - n_i^-)$  – to the largest eigenvalue of  $(T_{ij}^+ - T_{ij}^-)$ . If the largest eigenvalues are the same there is no need in the annihilation procedure for the FCIQMC and similar algorithms. However it can be shown[32] that:

$$\lambda_{\max}^{T_{ij}^+ + T_{ij}^-} \geq \lambda_{\max}^{T_{ij}^+ - T_{ij}^-} \quad (3.60)$$



The largest eigenvalue of  $(T_{ij}^+ + T_{ij}^-)$  is greater than the largest eigenvalue of  $(T_{ij}^+ - T_{ij}^-)$  determining growth rate of the true ground state. Large population of walkers is required to achieve sufficient amount of annihilation events[6] [32].



# Chapter 4

## Coupled Cluster Monte Carlo

The Coupled Cluster Quantum Monte Carlo (CCQMC) is a novel Quantum Monte Carlo algorithm based on the PMC method. The ground state wavefunction is obtained by the population dynamics of walkers in Slater determinant space. This approach allows us to tackle problems which are impossible or difficult to solve with high-order truncation levels of the deterministic coupled cluster theory. The coupled cluster theory is formulated slightly differently from the deterministic approach described in chapter 2. In this chapter we present the CCQMC method recently created by Thom and collaborators [38].

### 4.1 the Coupled Cluster Monte Carlo theory

The main idea is to rewrite the coupled cluster equations in such form that it is possible to sample excitations (or various Slater determinants) stochastically. We start with a complete space of Slater determinants and second-quantized spin-orbitals. Let us have  $N$  occupied orbitals of  $2M$  orthogonal orbitals. We will denote them with  $i, j, k \dots$ , while the remaining  $2M - N$  unoccupied orbitals are  $a, b, c \dots$ . If we denote the reference determinant as  $D_0$ , all possible determinants can be written via excitation operators or so-called *excitors*:

$$|D_{ij}^{ab}\rangle = \hat{a}_{ij}^{ab} |D_0\rangle, \quad (4.1)$$

where  $i$  and  $j$  were replaced by  $a$  and  $b$  in the reference. By acting with every  $\binom{2M}{N}$  possible excitors on the reference determinant we obtain different excited determinants. There is a one-to-one correspondence between excitors and determinants, it is possible to denote  $D_i = \hat{a}_i D_0$  that simplifies notation later. Thus the coupled cluster wavefunction can be written as follows:

$$\Psi_{CC} = e^{\hat{T}} D_0, \quad (4.2)$$

where  $\hat{T} = \sum_i t_i \hat{a}_i$ .

At this point the coupled cluster equation can be obtained by the projection of the Schrödinger equation onto excited determinants and be solved by the iterative procedure as it was described in chapter 2:

$$\langle D_{\{i\}} | \hat{H} - E | \Psi_{CC} \rangle = 0, \quad (4.3)$$

where subscript  $\{i\}$  means that we need to solve the equation for all excited determinants. Instead we attack this problem starting from regular commencement for all PMC methods, the imaginary-time Schrödinger equation (3.22) and search a solution as the result of application of the imaginary-time propagator. We assume that the ground state wavefunction has a non-zero overlap with the initial guess for it ( $\langle \Psi_0 | \Psi(\tau = 0) \rangle \neq 0$ ). Usually a single determinant function (for example the Hartree-Fock determinant) is sufficient as a initial guess:

$$|\Psi_0\rangle \propto \lim_{\tau \rightarrow \infty} e^{-\tau \hat{H}} |\Psi(\tau = 0)\rangle. \quad (4.4)$$

We set the constant of proportionality to be equal to  $e^{\tau S}$  to maintain constant non-zero normalization:

$$|\Psi_0\rangle \propto \lim_{\tau \rightarrow \infty} e^{-\tau(\hat{H}-S)} |\Psi(\tau = 0)\rangle. \quad (4.5)$$

Later we will see that tuning of  $S$  will lead to convergence of it to the true ground state energy  $E_0$ . The series expansion of the exponential propagator leads to the discretization scheme where the linear projector is used ( $e^{-\tau(\hat{H}-E_0)} \approx (1 - \delta\tau(\hat{H} - S))$ ):

$$|\Psi_0\rangle \propto \lim_{N \rightarrow \infty} (1 - \delta\tau(\hat{H} - S))^N |\Psi(\tau = 0)\rangle \quad (4.6)$$

Here we must make a remark regarding the value of  $\delta\tau$ . The repeated application of the propagator  $(1 - \delta\tau(\hat{H} - S))$  converges to the exact ground state if the time step satisfies:

$$\delta\tau \leq \frac{2}{E_{max} - S}, \quad (4.7)$$

where  $E_{max}$  is the largest eigenvalue of  $\hat{H}$  and  $S \approx E_0$ . It is possible to estimate  $E_{max}$  as an energy of the most excited determinant available in a given one-electron basis. At the beginning of the calculations one can use the best estimated value available for  $E_0$ . This upper limit for the time step follows from the requirements that:

- The largest eigenvalue of propagator  $(1 - \delta\tau(\hat{H} - E_0))$  corresponding to the ground-state Hamiltonian must be equal to one.
- All other eigenvalues corresponding to the excited states must have absolute values less than one.

With large values of the time-step  $\delta\tau$  the excited-state contributions do not decay in time.

To obtain a ground state wavefunction we need to compute iteratively the following expression:

$$|\Psi(\tau + \delta\tau)\rangle = (1 - \delta\tau(\hat{H} - S))|\Psi(\tau)\rangle \quad (4.8)$$

Recalling the exponential ansatz for wavefunction and projecting equation on the excited determinant  $D_{\{i\}}$  one obtains:

$$\langle D_{\{i\}}|\Psi_{CC}(\tau + \delta\tau)\rangle = \langle D_{\{i\}}|\hat{1} - \delta\tau(\hat{H} - S)|\Psi_{CC}(\tau)\rangle, \quad (4.9)$$

$$\langle D_{\{i\}}|e^{\hat{T}}D_0\rangle = \langle D_{\{i\}}|e^{\hat{T}}D_0\rangle - \delta\tau\langle D_{\{i\}}|(\hat{H} - S)e^{\hat{T}}|D_0\rangle. \quad (4.10)$$

Now we consider one arbitrary determinant  $D_{\mathbf{i}}$  from the set  $D_{\{i\}}$ :

$$\langle D_{\mathbf{i}}|e^{\hat{T}}D_0\rangle = t_{\mathbf{i}} + \mathcal{O}(\hat{T}^2), \quad (4.11)$$

where the largest contribution we get from the linear amplitude and the small contribution from higher order terms, which can be canceled on both sides of equation (4.10). All other terms( $\neq i$ ) are zero:

$$t_{\{i\}} - \delta\tau\langle D_{\{i\}}|(\hat{H} - S)|\Psi_{CC}\rangle = t_{\{i\}}. \quad (4.12)$$

It is possible to look into this equation as an iterative procedure from time  $\tau$  to  $(\tau + \delta\tau)$ :

$$t_{\{i\}}(\tau + \delta\tau) = t_{\{i\}}(\tau) - \delta\tau\langle D_{\{i\}}|(\hat{H} - S)|\Psi_{CC}(\tau)\rangle, \quad (4.13)$$

Generally there is no guarantee that this iteration will converge to the coupled cluster solution. However it appears to be convergent with a correct choice of  $\delta\tau$  for single-reference systems[33]. The second term in equation(4.13) is not trivial and it might be easier to sample it stochastically. That can be performed by the following scheme:

### 1. Sampling of the wavefunction

Generally in the deterministic coupled cluster theory there is a truncation of the exponential ansatz of the wavefunction in energy expression. And there is a similar truncation for the amplitudes equations but at higher term since excited determinants are placed in the bra-part. The second term in equation (4.13) has excited determinants in the bra-part as well. It can be easily noted that this term is similar to the coupled cluster unlinked equations for amplitudes. It has already been mentioned that these equations are difficult to work with. Recall the exponential ansatz for the wavefunction as follows:

$$\Psi_{CC} = (1 + \hat{T} + \frac{\hat{T}^2}{2!} + \frac{\hat{T}^3}{3!} + \dots)|D_0\rangle, \quad (4.14)$$

where every  $\hat{T}$  operator is a sum of single, double etc. excitation operators, which in turn are expressed via creation/annihilation operators. We choose randomly only one term from this expansion, and (4.14) is written as:

$$\Psi_{CC} = t_{ij}^{ab} c_a^\dagger c_b^\dagger c_j c_i |D_0\rangle = t_{ij}^{ab} |D_{ij}^{ab}\rangle, \quad (4.15)$$

where double excitation was randomly chosen. The notations with bold subscripts introduced above become extremely handy now. Independently of the excitation level we can write:

$$\Psi_{CC} = t_{\mathbf{p}} \hat{c}_{\mathbf{p}} |D_0\rangle = t_{\mathbf{p}} |D_{\mathbf{p}}\rangle, \quad (4.16)$$

where the bold subscript at creation/annihilation operator  $\hat{c}$  denotes the fact that it can be possibly constructed by some complex string of creation/annihilation operators. It can in general represent any excitation, for example, one triple or two single excitations. In equation (4.15) we used the same notations as in chapter 2 describing the deterministic coupled cluster theory, while the notation of equation (4.16) is more suitable for our stochastic variant. The result of this first sampling is the collapse of the wavefunction to a determinant instance. Later in our text this step will be referred to the collapse of the wavefunction to the cluster. These clusters can be composite, consisting of two or more excitors, or non-composite, which will be described in more details in chapter 6.

## 2. Sampling of the action of the Hamiltonian

For sampled in the first stochastic process  $|D_{\mathbf{p}}\rangle$  we need to calculate  $\langle D_{\{\mathbf{q}\}} | \hat{H} - S | D_{\mathbf{p}} \rangle$  for all possible  $\langle D_{\{\mathbf{q}\}} |$  and store  $t_{\{\mathbf{q}\}}$ . As the two-body Hamiltonian connects only to single and double excitations from a determinant, we may instead pick  $\langle D_{\{\mathbf{q}\}} |$  as a single or double excitation of the collapsed wavefunction and store all  $t_{\{\mathbf{q}\}}$ . The sampling of the action of the Hamiltonian reviewed so far, however, suffers from the fact that many clusters have small contribution. The latest requires a lot of Monte Carlo samples to catch these weak contributions, as was pointed by Spenser[38]. Instead it is possible to represent amplitudes of excitors by a set of populations of fictitious particles – *excips*<sup>1</sup>. Each excip represents a signed unit weight and corresponds to excitation operator  $\hat{c}_{\mathbf{q}}$ . During simulation the population of excips is changed. Averaged over time number of excips corresponds to certain excitation operator and gives us excitor amplitude.

To summarize, the CCQMC algorithm it may be useful to compare (4.13) with the diffusion equation(3.37). The imaginary-time evolution dynamics of the

---

<sup>1</sup>Definition of *excitor particle* or *excip* was introduced by J. S. Spencer and A. J. Thom [33] and can have different meaning in early papers.

amplitudes is:

$$\frac{\delta t_{\{i\}}(\tau)}{\delta \tau} = -\langle D_{\{i\}} | (\hat{H} - S) | \Psi_{CC}(\tau) \rangle. \quad (4.17)$$

After choosing some arbitrary excitation  $\mathbf{j}$  such as  $\Psi_{CC} = t_j |D_j\rangle$ :

$$\frac{\delta t_{\{i\}}(\tau)}{\delta \tau} = -t_j \langle D_{\{i\}} | \hat{H} | D_j \rangle + t_j E \langle D_{\{i\}} | D_j \rangle. \quad (4.18)$$

Now we consider one arbitrary  $\mathbf{i}$  from a variety of all  $\{\mathbf{i}\}$ . According to Slater's rules the matrix elements of the Hamiltonian between determinants that differ by more than two orbitals are zero. This allows us to improve choice of  $\mathbf{i}$  by picking determinant  $\langle D_i |$  to be single or double excitation of the ket-part  $|D_j\rangle$ , which leads to  $\langle D_i | \hat{H} | D_j \rangle \neq 0$ . Considering two cases  $\mathbf{i} = \mathbf{j}$  and  $\mathbf{i} \neq \mathbf{j}$ , we can split equation (4.18) as follows:

$$\frac{\delta t_j(\tau)}{\delta \tau} = -t_j \langle D_j | \hat{H} | D_j \rangle + t_j E = -t_j H_{jj} + t_j E = -t_j (H_{jj} - S), \quad (4.19)$$

if  $\mathbf{i} \neq \mathbf{j}$ :

$$\frac{\delta t_i(\tau)}{\delta \tau} = -t_j \langle D_i | \hat{H} | D_j \rangle = -t_j H_{ij}. \quad (4.20)$$

Equations (4.19) and (4.20) have terms, corresponding to the sink/source term in the diffusion equation. We can consider the simulations to be similar to the DMC method but without diffusion moves. Indeed, a closer look at (4.19) and (4.20), recalling the amplitude representation as a population of virtual particles  $t_i \propto N_i$ , allows us to consider a space of Slater determinants as a discrete space of points where these virtual particles are being born or suppressed in imaginary time. This process is governed by the diffusion equations (4.19) and (4.20).

## 4.2 The general CCQMC algorithm

The starting point for all Monte Carlo algorithms is to initialize some initial walker distribution or, in our case, the distributions of walkers. The details of our initialization procedure will be described in chapter 6. The general CCQMC algorithm can be split up into the following steps:

1. We choose one Slater determinant from the whole set of  $\binom{2M}{N}$  determinants (collapse of wavefunction to a cluster). This step corresponds to the move from the general equation (4.17) to equation (4.18) where  $\Psi_{CC} \rightarrow t_j |D_j\rangle$ .
2. We choose random single or double excitations of the collapsed wavefunction  $t_j |D_j\rangle$ , denoting it  $|D_i\rangle$ . The determinant, corresponding to these random excitations is considered as the action of some other excitor on the

reference  $|D_i\rangle = \hat{c}_i D_0$ . This action may lead to a potential sign change of the expression. It is needed to be taken into account, since it influences the resulting excip sign.

3. We simulate the branching process with the probability proportional to  $(H_{jj} - S)\delta\tau$  and  $H_{ij}\delta\tau$ :
  - Spawn process. With probability proportional to  $H_{ij}\delta\tau$  we attempt to create a walker on excitor  $\hat{c}_i$ . Sign of excip should be treated with care.
  - Death process. With probability proportional to  $(H_{jj} - S)\delta\tau$  we kill excip on excitor  $\hat{c}_j$  by creating excip of an opposite sign.
4. We annihilate a pair of walkers with different signs populating the same determinant(excip).
5. We control the population adjusting parameter  $S$ . This step should be implied after *exponential* growth of the population is reached.
6. We sample the projected energy:

$$E_{proj} = \frac{\langle D_0 | \hat{H} | \Psi_{CC} \rangle}{\langle D_0 | \Psi_{CC} \rangle} \quad (4.21)$$

This step should be applied when the desired number of walkers is reached and the sign structure of the solution is correct. The sign problem affects the CCQMC in the same way as the FCIQMC and is covered in chapter 6.

7. We iterate to the next imaginary-time step by repeating the procedure from first step. The success of the CCQMC calculations depends on whether a critical population of walkers is reached.

### 4.3 Normalization

In equation (4.21) the intermediate normalization requires that  $\langle D_0 | \Psi_{CC} \rangle$  should be fixed to unity. We use the scheme of Spencer and Thom [33] to preserve the normalization by introduction of a variable  $N_0$ :

$$|\Psi_{CC}(\tau)\rangle = N_0(\tau) e^{\frac{\hat{T}(\tau)}{N_0(\tau)}} |D_0\rangle. \quad (4.22)$$

Here  $N_0$  is used to deal with the imbalance between populations on reference and in the excited space.  $N_0$  can be treated as population of excips on the reference. However there are no excitors on  $D_0$  and hence these *excips* on the reference are particles of the other kind and cannot be considered as excips of the cluster(excitor).



## 4.4 Population dynamics

The population of excips during the CCQMC calculations evolves through the following steps:

1. Initial growth.

We start with  $N_0$  excips living on  $D_0$  and no excips in the exited space. We allow them to spawn on the excited determinants, making sure the condition  $\langle \Phi_{i...}^a | \hat{H} | \Phi_0 \rangle \neq 0$  is fulfilled. More strictly,  $N_0$  population is a population of excips of a different kind or, in other words, a chosen way to preserve normalization. Additionally, excips cannot be moved among determinants. They die only with the spawn of the oppositely-signed excip of the same particular determinant(or excitor). We choose single or double excitation of the reference exactly  $N_0$  times and check spawn probability which, in case of success, lead to  $D_i$  is being populated with excip. Death never occurs on this step because  $\langle \Phi_0 | \hat{H} - S | \Phi_0 \rangle = 0$  due to  $S = E_{ref}$ . At this step the total rate of population growth is faster than  $N_0$  growth at the reference. Spenser and Thom[33] suggest that at this point one should not control population with energy shift  $S$  as it leads to the population of the reference drops to zero and unstable simulations. It is also expected the solution to have a wrong sign during this period [32] [33]. The total population grows with time at this step, however the annihilation event number gradually increases.

2. Annihilation plateau.

At some point of the simulation a number of the annihilation events is large enough to compensate spawning and death events. In other words spawning events result in an approximately equal to death occurrence number of positive and negative excips. The population of excips at different excitors changes sign to resemble sign corresponding to the correct parametrization of the ground state wavefunction. The height of the plateau is specific for different systems and depends on the initial population of walkers, the step value and the total number of single-particle states. We define the particle ratio as the ratio of the total population to the reference population. To determine whether a plateau is reached, a log-log plot of particles ratio to population of the excited subspace might be useful.

3. Exponential growth.

Spawning dominates over annihilation and the total population of excips grows exponentially with rate proportional to  $S - E_{cc}$ .

## 4.5 Population Control

Here we discuss briefly the last step of PopMc algorithm described in the previous chapter. The above mentioned step is not necessary for validity of the majority of PopMC algorithms. However it is needed to avoid statistical fluctuation in weights. In the CCQMC the population control provides additional way to evaluate the groundstate energy because of the form of the projector. One can think about the population control as a form of importance sampling used for stabilization of algorithm. Indeed uncontrolled population of walkers will either rapidly decay to walker graveyard or fill all the physical memory of a computer during simulation. Several examples are mentioned to stabilize a number of random walkers. If gradually more walker birth events take place, we need to lower  $S$  value. This can be performed as follows:

$$S = \langle E_0 \rangle + \alpha \frac{N_w^{(0)}}{N_w}, \quad (4.23)$$

where  $\langle E_0 \rangle$  is the best possible value at the current point of simulation,  $\alpha$  is a small parameter,  $N_w^{(0)}$  is desired number of random walkers,  $N_w$  is actual number of walkers. Main drawback of this simple procedure is a bias at small populations or large values of  $\alpha$ . Infinite population is required for the bias to disappear. Second idea is to introduce a global weight for the whole population of walkers:

$$\bar{w} = \frac{1}{N_w} \sum_{i=1}^{N_w} w_i. \quad (4.24)$$

In this case each walker has renormalized individual weight  $\omega_i$ :

$$\omega_i = \frac{w_i}{\bar{w}}. \quad (4.25)$$

Alternative idea for the population control is to make walker weight proportional to the number of these particular walkers.

## 4.6 Initiator approximation

Initiator approximation is an improvement of CCQMC algorithm, originally developed for the resemblant FCIQMC algorithm<sup>2</sup>. The basic idea is to introduce additional restrictions on spawning, specifically the prohibition on the spawning from unpopulated determinants. Given some predefined threshold  $n_{add}$ , determinants with the population lower than this value cannot spawn new excips. This changes the standard population dynamics by removing the annihilation plateau[33].

---

<sup>2</sup>Sometimes it is also called *i*-FCIQMC

## Chapter 5

# The Infinite Electron Gas

Our system of interest is the interacting *homogeneous electron gas* (HEG), sometimes called *free electron gas*.

In practice this model can be useful when we consider metals. If electrons bounded to the ions at the lattice sites form closed shells, the rest of the electrons can freely move through the volume. The model is expected to be a very good approximation to the valence electrons in alkali metals, where these electrons are bounded weakly to the ionic cores[12]. Furthermore, in condensed matter physics, the electron gas serves a similar function as infinite nuclear matter in nuclear physics[16]. It is worth mentioning that the three-dimensional electron gas is particularly important in density-functional theory as its basic functional, the local-density functional, is based on results from the uniform electron gas [31].

The electron gas is a typical example of a translationally invariant system. The fact that the self-consistent problem of the mean-field approximation is solved for translationally invariant systems makes the electron gas an obvious system of choice for study of new many-body methods. That is why the electron gas is also called a *test bed* for electron-electron correlations study.

The electron gas is perhaps the only realistic model of a system of many interacting particles that allows an analytical solution of the Hartree-Fock equations. Furthermore, one can also obtain an analytical expression for the total energy and the Hamiltonian matrix elements in the Hartree-Fock basis. Thus there is no necessity to perform self-consistent field iteration procedure for the Hartree-Fock equations, and that results in a great simplification for all post Hartree-Fock methods. We can start any advanced many-body algorithm without the preceding Hartree-Fock calculation, as the Hartree-Fock (HF) basis is known. Moreover, the electron gas has been studied using a large number of different many-body methods.

According to all mentioned above we use the electron gas as a benchmark system to compare stochastic coupled cluster approach with other known results including CCD.

## 5.1 The Hamiltonian

The free electron gas, also known as *jellium model*, is a physical abstraction for a system of many electrons which is not influenced by external forces except an attraction provided by a uniform background of ions. The charge density of the ions is uniformly smeared out through the volume of the system. It is assumed that ions are stationary and the system is neutral. The Hamiltonian for the described system can be written as follows:

$$\hat{H} = \hat{H}_{el} + \hat{H}_b + \hat{H}_{el-b}, \quad (5.1)$$

where  $\hat{H}_{el}$  is the energy of electronic subsystem.  $\hat{H}_b$  is the electrostatic energy of the positive uniformly distributed background charge.  $\hat{H}_{el-b}$  represents interaction of electrons with the positive background. The electronic term  $\hat{H}_{el}$  consists of the kinetic energy of the electrons and their interaction energy:

$$\hat{H}_{el} = \sum_{i=1}^N \frac{p_i^2}{2m} + \frac{e^2}{2} \sum_{i \neq j} \frac{e^{-\mu|\mathbf{r}_i - \mathbf{r}_j|}}{|\mathbf{r}_i - \mathbf{r}_j|}. \quad (5.2)$$

The energy contribution from the positive background charge is:

$$\hat{H}_b = \frac{e^2}{2} \int \int d\mathbf{r} d\mathbf{r}' \frac{n(\mathbf{r})n(\mathbf{r}')e^{-\mu|\mathbf{r} - \mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|}. \quad (5.3)$$

The interaction among electrons with the positive background charge is:

$$\hat{H}_{el-b} = -\frac{e^2}{2} \sum_{i=1}^N \int d\mathbf{r} \frac{n(\mathbf{r})e^{-\mu|\mathbf{r} - \mathbf{x}_i|}}{|\mathbf{r} - \mathbf{x}_i|}, \quad (5.4)$$

We assumed that the particle-particle interactions depend on the relative distance only, and the background charge has constant density  $n(\mathbf{r}) = N/\Omega$ , where  $N$  is the number of ions in volume  $\Omega$ . The considered system should be necessarily infinite. However such an idealization helps to avoid surface effects. We approximate the density of the electron gas via a cube with side  $L$  ( $\Omega = L^3$ ). In thermodynamic limit, namely  $N \rightarrow \infty$  and  $\Omega \rightarrow \infty$ , while the ratio  $N/\Omega$  is constant.

## 5.2 Finite-box approximations of the electron gas

We impose periodic boundary conditions on the edges of the cube mentioned above. The solution of the single-particle Schrödinger equation for a particle in the box problem is well known: the single-particle wave functions are given by plane wave functions normalized to a system volume. The single-particle functions can be written as follows:

$$\psi_{\mathbf{k}\sigma}(\mathbf{r}) = \frac{1}{\sqrt{\Omega}} \exp(i\mathbf{k}\mathbf{r})\xi_{\sigma}, \quad (5.5)$$

where  $\mathbf{k}$  is a wave-number, and  $\xi_{\sigma}$  represents spin. The spin part  $\xi_{\sigma}$  is given by:

$$\xi_{\sigma=+1/2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \xi_{\sigma=-1/2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (5.6)$$

The periodic boundary conditions impose restrictions on the wave-number:

$$k_i = \frac{2\pi n_i}{L} \quad i = x, y, z \quad n_i = 0, \pm 1, \pm 2, \dots \quad (5.7)$$

Each single particle function is associated with the energy in the following way:

$$\varepsilon_{n_x, n_y, n_z} = \frac{\hbar^2}{2m} \left( \frac{2\pi}{L} \right)^2 (n_x^2 + n_y^2 + n_z^2). \quad (5.8)$$

We choose the single-particle basis in a such way that both occupied and unoccupied single-particle spaces have a closed-shell structure. This means that all single-particle states corresponding to the energies below a chosen cutoff are included in the basis. We study only the unpolarized spin phase, in which all orbitals are occupied with one spin-up and one spin-down electron.

With the kinetic energy rewritten in terms of the discretized momenta we can set up a list of single-particle energies with momentum quantum numbers. The table (5.1) illustrates the relation between the single-particle energies and the energy shells. For the lowest six energy values the degeneracy in energy gives us 2, 14, 38, 54, 76 and 114 as magic numbers.

## 5.3 Ewald's summation technique

Let us consider the ion subsystem of an electron gas as a set of  $N$  point charges  $q_i$  at locations  $\mathbf{r}_1, \mathbf{r}_2 \dots \mathbf{r}_N$ . The total Coulomb interaction energy in this case is defined by the following expression:

$$E \propto \sum_{(i,j)}^N \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (5.9)$$

where the summation runs over all ionic pairs. The periodic boundary conditions mean that if there is an ion  $q_i$  at location  $\mathbf{r}_i$ , there is the same ion at location  $\mathbf{r}_i + n_1 \mathbf{c}_1 + n_2 \mathbf{c}_2 + n_3 \mathbf{c}_3$ , where  $n_1, n_2, n_3$  are arbitrary integers, and  $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$  are super-cell vectors. For cubic super-cell we can choose  $|\mathbf{c}_1| = |\mathbf{c}_2| = |\mathbf{c}_3| = L$ . The

$n_x^2 + n_y^2 + n_z^2$	$n_x$	$n_y$	$n_z$	$N_{\uparrow\downarrow}$
0	0	0	0	2
1	-1	0	0	14
1	1	0	0	
1	0	-1	0	
1	0	1	0	
1	0	0	-1	
1	0	0	1	
2	-1	-1	0	
2	-1	1	0	
2	1	-1	0	
2	1	1	0	
2	-1	0	-1	
2	-1	0	1	
2	1	0	-1	
2	1	0	1	
2	0	-1	-1	38
2	0	-1	1	
2	0	1	-1	
2	0	1	1	
3	-1	-1	-1	
3	-1	-1	1	
3	-1	1	-1	
3	-1	1	1	
3	1	-1	-1	
3	1	-1	1	
3	1	1	-1	54
3	1	1	1	

**Table 5.1:** Single-particle state energies(in atomic units)

total Coulomb interaction energy for these ions has to include the interactions between periodic images:

$$E \propto \frac{1}{2} \sum_n \sum_{i \neq j}^N \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}, \quad (5.10)$$

where  $\mathbf{n}$  is a vector with components  $n_1, n_2, n_3$ .

This sum is conditionally convergent with a slow convergence rate. Ewald proposed a method[9] which converges not only rapidly but also absolutely.

The charge density distribution for the system of point charges described by a collection of delta functions can be split up it into two terms by adding and

subtracting a Gaussian distribution. As a result, the Coulomb interaction is spit up into a short-range part, which is calculated in real space, and a long-range part, which is summed in Fourier space.

The Ewald electron-electron interaction operator can be written according to [9] as follows:

$$\hat{H}_{ee} = \sum_{i < j}^N v_E(\mathbf{r}_i - \mathbf{r}_j) + \frac{1}{2} N v_0, \quad (5.11)$$

where  $v_E(\mathbf{r})$  is the effective two-body interaction, defined as:

$$v_E(\mathbf{r}) = \sum_{\mathbf{k} \neq \mathbf{0}} \frac{4\pi}{L^3 k^2} e^{i\mathbf{k} \cdot \mathbf{r}} e^{-\eta^2 k^2 / 4} + \sum_{\mathbf{R}} \frac{1}{|\mathbf{r} - \mathbf{R}|} \text{erfc}\left(\frac{|\mathbf{r} - \mathbf{R}|}{\eta}\right) - \frac{\pi \eta^2}{L^3}. \quad (5.12)$$

$v_0$  is the self-interaction, defined by the following expression:

$$v_0 = \lim_{\mathbf{r} \rightarrow \mathbf{0}} \left\{ v_E(\mathbf{r}) - \frac{1}{r} \right\}. \quad (5.13)$$

$\text{erfc}(x)$  is the complementary error function, and  $\eta$  is a free parameter that can take any value in the interval  $(0, \infty)$ .

Fraser [10] explains how the electron-background and background-background terms  $\hat{H}_{eb}$  and  $\hat{H}_{bb}$ , cancel each other for the three-dimensional electron gas in the thermodynamical limit.

## 5.4 Analytical results for the electron gas

The electron gas is a well-studied system, illustrated by several excellent texts, see for example [12] or [21]. These texts contain detailed solution of the Hartree-Fock equations. We briefly present the results needed for the implementation of the electron gas basis in our code.

For the three-dimensional electron gas, the antisymmetrized matrix elements can be calculated by the following expression:

$$\langle \mathbf{k}_p m_{s_p} \mathbf{k}_q m_{s_q} | \hat{v} | \mathbf{k}_r m_{s_r} \mathbf{k}_s m_{s_s} \rangle_{AS} = \langle pq || rs \rangle \quad (5)$$

$$= \frac{4\pi}{L^3} \delta_{\mathbf{k}_p + \mathbf{k}_q, \mathbf{k}_r + \mathbf{k}_s} \left\{ \delta_{m_{s_p} m_{s_r}} \delta_{m_{s_q} m_{s_s}} (1 - \delta_{\mathbf{k}_p \mathbf{k}_r}) \frac{1}{|\mathbf{k}_r - \mathbf{k}_p|^2} - \delta_{m_{s_p} m_{s_s}} \delta_{m_{s_q} m_{s_r}} (1 - \delta_{\mathbf{k}_p \mathbf{k}_s}) \frac{1}{|\mathbf{k}_s - \mathbf{k}_p|^2} \right\}, \quad (5.14)$$

where the Kronecker delta functions  $\delta_{\mathbf{k}_p \mathbf{k}_s}$  and  $\delta_{\mathbf{k}_p \mathbf{k}_r}$  ensure that the momentum is not changed.  $\delta_{m_{s_p} m_{s_r}} \delta_{m_{s_q} m_{s_s}}$  reflects spin conservation.

The single-particle energies, defined by the Fock operator, are given by:

$$\langle p | \hat{f} | q \rangle = \frac{\hbar^2 k_p^2}{2m} \delta_{\mathbf{k}_p, \mathbf{k}_q} + \sum_i \langle pi || qi \rangle. \quad (7)$$

The reference energy can be evaluated by the following expression:

$$E_r = \sum_i \langle i | \hat{h}_0 | i \rangle + \frac{1}{2} \sum_{ij} \langle ij || ij \rangle + \frac{1}{2} N v_0. \quad (5.15)$$

The last term in this expression arises from the Ewalds summation technique, and  $v_0$  is defined by (5.12). In the thermodynamical limit this term vanishes.



# Chapter 6

## CCD solver implementation

In this chapter we present the implementation of the Coupled Cluster Doubles (CCD) and the Coupled Cluster Quantum Monte Carlo (CCQMC) methods.

C++ language was chosen for its efficiency and flexible object-orientation support. Additionally, the standard library (STL) provides big variety of data-structures and collection of algorithms which are easy to use. Moreover an extensive history of this language (together with a large inheritance from C and Fortran encompassing high performance libraries such as BLAS and LAPACK) is one of the benefits. In addition, great support of high performance and parallel computing libraries along with active development of language itself makes C++ an optimal choice for our algorithms. C++ was not designed for scientific computing[35] contrary to Fortran, however code re-usage is our higher priority. Python is mainly used for data-analysis and unit tests. Python libraries such as numpy allows one to work with different mathematical objects in a simple way by using algebraic expressions almost exactly. But for large-scale numerical calculations it was found to be slower than C++.

The focus of this chapter will be mainly on the details of the algorithms and C++ implementation. Source code with compiling and running instructions is freely available on github[18]. We begin with a discussion of the CCD solver. At the end of the chapter we present the CCQMC solver with detailed examination of sampling rules and sign problem.

### 6.1 The Coupled Cluster Equations

As explained earlier the most straight-forward way to solve the CC equation is to iterate the following non-linear algebraic equations:

$$\begin{aligned}
0 = & w_{ab}^{ij} + \hat{P}(ab) \sum_c f_{bc} t_{ij}^{ac} - \hat{P}(ij) \sum_k f_{kj} t_{kj}^{ab} + \frac{1}{2} \sum_{cd} w_{cd}^{ab} t_{ij}^{cd(n)} + \frac{1}{2} \sum_{kl} w_{ij}^{kl} t_{kl}^{ab(n)} \\
& + \hat{P}(ij|ab) \sum_{kc} w_{cj}^{kb} t_{ik}^{ac(n)} + \frac{1}{4} \sum_{klcd} w_{cd}^{kl} t_{ij}^{cd(n)} t_{kl}^{ab(n)} + \hat{P}(ij) \sum_{klcd} w_{cd}^{kl} t_{ik}^{ac(n)} t_{jl}^{bd(n)} \\
& - \frac{1}{2} \hat{P}(ij) \sum_{klcd} w_{cd}^{kl} t_{ik}^{dc(n)} t_{lj}^{ab(n)} - \frac{1}{2} \hat{P}(ab) \sum_{klcd} w_{cd}^{kl} t_{lk}^{ac(n)} t_{ij}^{db(n)}.
\end{aligned} \tag{6.1}$$

In the canonical Hartree-Fock basis the one-body operator contains diagonal elements only. Since all off-diagonal elements of the Fock matrix are zeros, we denote  $f_{ii} = \epsilon_i$ . In this case the iterative form of the full CCD amplitude equations can be written as follows:

$$\begin{aligned}
t_{ij}^{ab(n+1)} = & \frac{1}{\epsilon_{ij}^{ab}} (w_{ab}^{ij} + \frac{1}{2} \sum_{cd} w_{cd}^{ab} t_{ij}^{cd(n)} + \frac{1}{2} \sum_{kl} w_{ij}^{kl} t_{kl}^{ab(n)} + \hat{P}(ij|ab) \sum_{kc} w_{cj}^{kb} t_{ik}^{ac(n)} \\
& + \frac{1}{4} \sum_{klcd} w_{cd}^{kl} t_{ij}^{cd(n)} t_{kl}^{ab(n)} + \hat{P}(ij) \sum_{klcd} w_{cd}^{kl} t_{ik}^{ac(n)} t_{jl}^{bd(n)} - \frac{1}{2} \hat{P}(ij) \sum_{klcd} w_{cd}^{kl} t_{ik}^{dc(n)} t_{lj}^{ab(n)} \\
& - \frac{1}{2} \hat{P}(ab) \sum_{klcd} w_{cd}^{kl} t_{lk}^{ac(n)} t_{ij}^{db(n)}),
\end{aligned} \tag{6.2}$$

where  $\epsilon_{ij}^{ab} = \epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b$ . Let us denote all the diagram terms which are linear in  $\hat{T}_2$  as  $L$ , and denote all the quadratic in  $\hat{T}_2$  terms as  $Q$ . Additionally we assign index to each term in their order of appearance in equation (6.2). Equation (6.2) is then simplified to:

$$t_{ij}^{ab(n+1)} = \frac{w_{ab}^{ij} + \sum_{n=1}^3 L_n + \sum_{n=1}^4 Q_n}{\epsilon_{ij}^{ab}}. \tag{6.3}$$

With the amplitudes equation written in this form it's easy to see that by letting the  $L$ - and  $Q$ -terms be zero we get the second-order many-body perturbation theory equations. That in turn looks as a natural initial approximation for the amplitudes.

Having defined what is needed for the iterative solution, we are moving on to the discussion of the computer resources required to perform these computations. First of all we see that the set of the amplitudes  $t_{ij}^{ab}$  is a tensor of rank four, and the simplest possible implementation of this object on computer is a four-dimensional array. Let us estimate memory requirements. For electron gas to include 10 shells(that corresponds to 246 single-particle states) we need to

store the two-body matrix elements of the Hamiltonian  $\langle pq|\hat{v}|rs\rangle$ . If we store all two-body matrix elements in a four-dimensional array, we need  $264^4 \times 8$  bytes of memory which corresponds to approximately 300 GB. This is far beyond the limits of personal computers in our days. Some physical models, such as the electron gas, have analytical expressions for the calculation of the two-body matrix elements. This eliminates difficulties with the storage of the matrix elements since they can be easily calculated on the fly. However computer code for the CC solver should be written considering a possibility of a simple extension to other systems. Generally the two-body matrix elements are obtained numerically with algorithms often requiring a considerable time. Since the two-body matrix elements are not changed during CC iterative procedure, it is obvious to store them in memory. Hence some smart ways of saving them are required for the computer code to be efficient.

The second conspicuous problem in equation (6.2) is the CPU time. The implementation of all sums in equation (6.2) as nested for loops will result in a poor performance. As it is shown below these sums can be treated as matrix-matrix multiplications, and that provides us with a very efficient method to calculate them, using BLAS library[11].

---

**Algorithm 1:** CC iteration

---

**input** : MBPT2 Energy, Initial guess for Amplitudes

**output:** CC correlation energy

```

1 while EnergyDiff > Tolerance do
2   Calculate Amplitudes
3   Calculate Correlation Energy
4   Calculate Energy Difference With Previous Iteration
5   Update Correlation Energy
6   Update Amplitudes
7 end
```

---

In this thesis we have implemented the brute force solver and tested it against the available codes and the published results. The aim of the development of this solver was to write a code as close as possible to the CCD equations. With a code reproducing the correct results for the correlation energy, we believe that all the diagrams are calculated correctly. Later in this chapter, we show that numerically efficient block implementation is quite abstract in its comparison with the CC equations. And, hence, it is very useful to have a possibility to test the implementation of every single diagram against the brute force calculation. Our strategy was to use Python first, because of the fact that the library *numpy* provides an easy method (*einsum*) to calculate all terms in equation (6.2). After the CCD energy result of our Python solver coincided with the results of codes in [16] and, we started the implementation of the C++ brute force solver. The advantage of using the existing Python solver would be that each diagram from

equation (6.2) was already available. The implementation of equation (6.2) in C++ with nested for-loops requires a lot of labor with careful treatment of the indices increasing opportunity for mistakes. After the brute force solver in C++ was completed, we started implementation of the block solver. We found it useful to test the implementation of each diagram in the block solver against the brute-force one. The general iteration procedure is common for all solvers despite the fact that storing and calculation of the amplitudes may differ. This iterative scheme is presented in listing (1). Lastly it should be mentioned that sometimes we get divergent results. It is common to deal with this problem by the scaling the amplitudes with a parameter  $\alpha \in (0, 1]$  during the iterations in the following way:

$$t^{(n)} = \alpha t^{(n)} + (1 - \alpha)t^{(n-1)}. \quad (6.4)$$

## 6.2 The Straight Forward Implementation

The straight forward, or sometimes called *naive*, implementation stands for implementing CC amplitudes and energy equations as they appear in the mathematical form in equation (6.2). We have already discussed that the straight forward solver is neither memory optimized nor CPU efficient. However the incredible benefit of this code is the possibility to use it later for the benchmarking of more advanced implementations such as sparse or block. Inside the CCD loop we need to calculate the amplitudes and the correlation energy as it has been already discussed in listing (1). All sums are translated into a nested for-loops, and the diagrams are added term by term. The CCD amplitudes are calculated

in the following way:

---

**Algorithm 2:** Calculation of the CCD amplitudes. The straight forward implementation.

---

```

input : Initial guess for Amplitudes
output: Amplitudes 4D array

1 for  $i \leftarrow 1$  to  $N_{\text{Fermi}}$  do
2   for  $j \leftarrow 1$  to  $N_{\text{Fermi}}$  do
3     for  $a \leftarrow N_{\text{Fermi}}$  to  $N_{\text{States}}$  do
4       for  $b \leftarrow N_{\text{Fermi}}$  to  $N_{\text{states}}$  do
5          $V \leftarrow \text{TBME}(i, j, a, b)$ 
6          $\text{Denominator} \leftarrow \text{SPE}(i) + \text{SPE}(j) - \text{SPE}(a) - \text{SPE}(b)$ 
7         for  $c \leftarrow N_{\text{Fermi}}$  to  $N_{\text{States}}$  do
8           for  $d \leftarrow N_{\text{Fermi}}$  to  $N_{\text{states}}$  do
9              $\text{Amplitudes}(i, j, a, b) \leftarrow V +$ 
10               $\frac{1}{2} \text{TBME}(a, b, c, d) \text{AmplitudesOld}(c, d, i, j)$ 
11           end
12         end
13          $\text{Amplitudes}(i, j, a, b) \leftarrow + 6 \text{ last terms}$ 
14          $\text{Amplitudes}(i, j, a, b) /= \text{Denominator}$ 
15       end
16     end
17   end
18 end

```

---

In listing (2) on line 6 the TBME function returns the two-body matrix elements of the Hamiltonian. The function SPE on line 7 provides the single particle energy. The CCD amplitudes are stored as a four-dimensional array. There is a possibility to store the TBME and the amplitudes as one-dimensional array using simple index function. However it is just a reorganization of storage scheme, and we do not gain any benefits from this without additional restrictions for the two-body matrix elements. We discuss it in more detail in the next section. The simplest example of such index function can be written as the following expression:

$$I(p, q, r, s) = pN^3 + qN^2 + rN^2 + s, \quad (6.5)$$

where  $N$  is a maximal value of  $p, q, r$  or  $s$ .

To calculate the CCD energy function in a straightforward implementation one needs to implement the following equation:

$$\Delta E_{CCD} = \frac{1}{4} \sum_{ijab} w_{ab}^{ij} t_{ij}^{ab}. \quad (6.6)$$

Equation (6.6) is implemented in code in the simplest possible way as follows:

---

**Algorithm 3:** The CCD correlation energy. Straight forward implementation with nested for-loops.

---

```

input : Amplitudes 4D array and TBME 4D array
output: CCD Correlation energy

1 for  $i \leftarrow 1$  to  $N_{\text{Fermi}}$  do
2   for  $j \leftarrow 1$  to  $N_{\text{Fermi}}$  do
3     for  $a \leftarrow N_{\text{Fermi}}$  to  $N_{\text{States}}$  do
4       for  $b \leftarrow N_{\text{Fermi}}$  to  $N_{\text{states}}$  do
5         CCD Correlation Energy +=
          TBME( $i, j, a, b$ )*Amplitudes( $a, b, i, j$ )
6       end
7     end
8   end
9 end

```

---

### 6.3 The Sparse Implementation

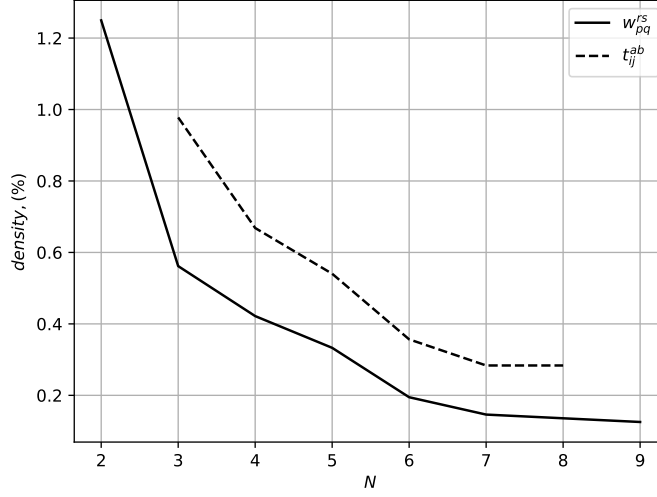
In equation (6.2) all the sums contain the product of the amplitudes and the two-body matrix elements of the Hamiltonian. As it has been already mentioned the two-body matrix elements of the Hamiltonian do not change during the iterative procedure. Hence we can precalculate these matrix elements and store them. One can easily notice that the majority of these elements are zero. In figure (6.1) we estimate the density of non-zero elements depending on the number of single-particle states.

The same pattern is observed for the CCD amplitudes. In figure (6.1) it is easy to see that the density of non-zero amplitudes is quite low (below 1%). The most logical course for improving the brute force storing scheme (four-dimensional array) is to use some kind of mapping function (for example (6.5)) to create one-to-one correspondence between indices:

$$\langle pq | rs \rangle \rightarrow \langle P || R \rangle. \quad (6.7)$$

Now we can store matrix elements in computer memory instead of four-dimensional array. However we have not gained any significant performance yet, as we still store a lot of zeros. But taking into account the density of the obtained matrix, we can treat it as so-called *sparse matrix*. There are several strict definitions of sparse matrices [25]. However, since our goal was not to write a sparse solver for CCD we just scratch the surface of this topic. We find a sparse solver to be worth mentioning as it provides overview of the whole CC code optimization strategy:

brute-force solver:	sparse solver:	block solver:
CPU is not optimized	→ CPU is not optimized	→ CPU is optimized
RAM is not optimized	RAM is optimized	RAM is optimized



**Figure 6.1:** Density of the  $T_2$  amplitudes and the two-body matrix elements for the 3D electron gas, given in percentages.  $N$  denotes maximum value of the shell number. The amplitude density was estimated for 14 electrons.

A sparse matrix can be written in a special form. As a simple example we can use coordinate list (COO) format. The name of format is self-explanatory: we store values and coordinates of the element, namely row and column indices in the following way:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 3 & 6 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 3 & 1 & 6 & 7 \\ 1 & 0 & 5 & 0 & 5 & 4 \\ 0 & 1 & 1 & 2 & 2 & 3 \end{pmatrix} \begin{matrix} \leftarrow \text{value} \\ \leftarrow \text{row} \\ \leftarrow \text{column} \end{matrix}$$

Summarizing, we can use the above mentioned sparse matrix storage format for the reduction of the memory usage. The drawback of this method compared to the naive implementation might be in growing time for accessing needed elements.

Let us discuss a possible way to optimize the CPU time. We consider equation (6.6) for the CCD correlation energy. Each pair of indices can be mapped into one:

$$\begin{aligned} ij &\rightarrow I \\ ab &\rightarrow A \end{aligned}$$

From now on, we denote the upper index, corresponding to the bra-part of the two-body matrix element, as a row index in the matrix representation. The lower index(ket-part) represents a column index. A tensor element  $w_{ab}^{ij}$  has become a

matrix element  $W_A^I$ , that can be written in a more common way as  $W_{AI}$ . Thus equation (6.6) looks like:

$$\Delta E_{CCD} = \frac{1}{4} \sum_{IA} W_A^I T_I^A = \frac{1}{4} \sum_{IA} W_{AI} T_{IA} = \frac{1}{4} \text{trace}(W \times T), \quad (6.8)$$

which is nothing else but a matrix-matrix multiplication. Thus the nested for-loops for evaluation of the sums can be replaced by fast linear algebra operations. Libraries like BLAS(Basic Linear Algebra Subprograms) are the obvious choice. However BLAS can multiply quickly only dense matrices, so we need either to create a fast library, which can multiply sparse matrices, or to use just a summation, instead of matrix-matrix multiplications. Additionally we must note that not all diagrams in equation (6.2) can be easily replaced by the matrix products. For example we consider the fourth term:

$$\hat{P}(ij|ab) \sum_{kc} w_{cj}^{kb} t_{ik}^{ac}.$$

Before the matrix multiplication elements should be rearranged so that their indices match definition of the matrix-matrix multiplication. The resulting matrix must be rearranged back. We can conclude that this scheme is suited for the memory optimization but not the CPU optimization. In the next section we present so-called *block* CCD solver which is both CPU and memory efficient.

## 6.4 The Block Implementation

The block solver is much more abstract in comparison to the naive CCD solver. However it is the best one in the sense of optimization of both RAM- and CPU-usage. This is achieved through the utilization of the physical symmetries in the interaction.

The idea underlying this method is to take into account symmetries in the two-body interaction, for example, the angular momentum conservation or the spin conservation. Indeed, if we consider two-body matrix element  $\langle pq|\hat{v}|rs\rangle$  it's easy to see, that it is non-zero only if two-body states  $|pq\rangle$  and  $|rs\rangle$  share the same quantum numbers.  $\hat{v}$  acting to the right do not change quantum numbers of  $|rs\rangle$ , and the result is left-projected onto  $\langle pq|$ . The similar property can be noted for the amplitudes, as they represent excitations due to the interaction, and, hence,  $t_{ij}^{ab} \neq 0$  if a set of the quantum numbers for state  $ij$  is the same as for  $ab$ . The last can be easily seen upon the initialization of the amplitudes. Equation (6.3) provides the first guess for the amplitudes:

$$t_{ij}^{ab(0)} = \frac{w_{ab}^{ij}}{\epsilon_{ij}}. \quad (6.9)$$



Let us consider  $\frac{1}{2} \sum_{ab} w_{cd}^{ab} t_{ij}^{cd}$  term from equation (6.2). According to the discussion above it is possible to evaluate it as a matrix-matrix multiplication. Taking into account the symmetries of the interaction, the only non-zero terms in the result matrix are obtained from the matrix elements which correspond to the single-particle states obeying the following requirements:

$$\mathbf{k}_a + \mathbf{k}_b = \mathbf{k}_c + \mathbf{k}_d = \mathbf{k}_i + \mathbf{k}_j, \quad (6.10)$$

and

$$s_a + s_b = s_c + s_d = s_i + s_j. \quad (6.11)$$

The obvious conclusion here is to construct the interaction matrix in a such way that its elements satisfy equations (6.10) and (6.11). In this case we avoid storing zero-elements and save the CPU time because there are no elements multiplied by zeros. The mentioned subsets of the TBME elements correspond to regions or *blocks* in the matrix representation of the tensor describing the interactions. The detailed analysis of the CCD amplitude equations shows that there are only terms where summations go over particle- or hole states. By definition the amplitudes are of the form particle-particle hole-hole. The two-body matrix elements are the elements of the form  $\langle pp||hh \rangle$   $\langle pp||pp \rangle$   $\langle ph||ph \rangle$  etc. So it's naturally to split whole interaction matrix into blocks. This allows to save memory and utilize symmetries to avoid multiplication by zeros for all diagrams. Additionally it is worth mentioning that all elements with repeated indices are zeros, for example  $t_{ii}^{ab}$  or  $\langle pp|\hat{v}|rs \rangle$ .

## 6.5 The Symmetry Channels

As it has been shown in the previous section, the construction of blocks obeying some particular symmetry laws is greatly improves numerical efficiency. In our system of interest (electron gas), these conservation laws are the conservation of the momentum and the conservation of the spin. For other systems these laws might be different. To generalize this discussion we define so-called *symmetry channels*. Let us denote  $(s_a + s_b)$  and  $(\mathbf{k}_a + \mathbf{k}_b)$  from equations (6.10) and (6.11) as  $S$  and  $\mathbf{K}$  correspondingly. Pair of two-body state quantum numbers  $(S, \mathbf{K})$  forms then so-called symmetry channel.

The general iterative procedure for solving the CCD equations, which is presented in listing (1) is not changed. We need just to add a function which sets up channels before the *while*-loop in line 2 and change methods for the calculation of the amplitudes and the correlation energy. To keep our CCD code more general, the amplitudes are set up in a different class. This is an obvious decision since channels reflect properties of the system.

The single-particle state quantum numbers for the electron gas are shown in table 5.1. The simplest possible way to set up symmetry channels is to loop

through all the combinations of the two-particle states and to check whether the quantum numbers are saved or not. We present this implementation in listing(4). In our C++ code method `setUpChannels` from `electrongaschannelset` class implements channel setup.

The result of this function is a vector of channels, where each channel contains two-body states obeying the symmetries of interactions. In our implementation one symmetry channel is represented by an object of class `channel`. Every instance of this class has a set of attributes which are vectors of index pairs of the states obeying symmetries in interaction. There are vectors of two-body states (represented by objects of class `channelindexpair`) like *particle minus hole*. Additionally we need *particle plus particle minus hole* and some other vectors. This is explained in more detail in the next section.

We should note that `setupchannels` method is pure virtual, and `electrongaschannelset` is a derived class. This allows to expand our code easily on other systems. To add a new system one needs to create a new derived class from the generalized class `channelset` and implement the `setupchannels` method utilizing the symmetries of this new system.

As it was already mentioned diagrams have no sum over all the states (particles and holes), thus the interaction matrix can be divided into blocks. All matrices are stored separately but are united by the channel. The interaction matrix can be split up into blocks as follows:

$$\begin{pmatrix} & |hh\rangle & |ph\rangle & |hp\rangle & |pp\rangle \\ \langle hh| & \langle hh|hh\rangle & \langle hh|ph\rangle & \langle hh|hp\rangle & \langle hh|pp\rangle \\ \langle ph| & \langle ph|hh\rangle & \langle ph|ph\rangle & \langle ph|hp\rangle & \langle ph|pp\rangle \\ \langle hp| & \langle hp|hh\rangle & \langle hp|ph\rangle & \langle hp|hp\rangle & \langle hp|pp\rangle \\ \langle pp| & \langle pp|hh\rangle & \langle pp|ph\rangle & \langle pp|hp\rangle & \langle pp|pp\rangle \end{pmatrix},$$

where the first column and the first row denotes kind of vector holding particular type of states. For example  $|pp\rangle$  is a vector that is made up from particle-particle states.

Summarizing, we store many small matrices or blocks for every symmetry channel instead of a four-dimensional array or a COO-formatted matrix. In the `ccd` class method `setUpInteractionMatrixBlocks` provides a creation of the interaction matrix blocks. For every channel it takes the vectors of index pairs as an input, created as described in listing 4, and returns the interaction matrix blocks with unique link to each channel. In order to create a block corresponding to  $\langle hp|pp\rangle$ -type of interaction, we need to create a matrix for every channel by constructing it from the corresponding vectors. Row index of this matrix is given by the bra-part  $\langle hp|$ , and column index is taken from the ket-part  $|pp\rangle$ . The value of the matrix element is calculated by TBME method of the generalized `SPBasis` class.

---

**Algorithm 4:** Setting up symmetry channels for 3D electron gas.
 

---

```

input : Single particle basis
output: Vector of matrices  $\langle pp||hh \rangle$ ,  $\langle pp||hh \rangle$  and  $\langle pp||hh \rangle$ 

1  $N_{\max}$  - maximum shell number;;
2  $S_{\max} = 2$ ; SP spin value is  $\pm 1$ ;
3 for  $N_x \leftarrow -N_{\max}$  to  $N_{\max}$  do
4   for  $N_y \leftarrow -N_{\max}$  to  $N_{\max}$  do
5     for  $N_z \leftarrow -N_{\max}$  to  $N_{\max}$  do
6       for  $S \leftarrow -S_{\max}$  to  $S_{\max}$  do
7         Create channel
8         for  $i \leftarrow 1$  to  $N_{\text{Fermi}}$  do
9           for  $j \leftarrow 1$  to  $N_{\text{Fermi}}$  do
10            if  $N_x, N_y, N_z, S$  of state  $(i + j) = N_x, N_y, N_z, S$ 
11              then
12                add i,j to HoleHole vector
13              end
14            end
15          add HoleHole vector to Channel
16          for  $i \leftarrow 1$  to  $N_{\text{Fermi}}$  do
17            for  $a \leftarrow N_{\text{Fermi}}$  to  $N_{\text{States}}$  do
18              if  $N_x, N_y, N_z, S$  of state  $(i + a) = N_x, N_y, N_z, S$ 
19                then
20                  add i,a to ParticleHole vector
21                end
22              end
23            add ParticleHole vector to Channel
24            ...
25            add ParticleParticle vector to Channel
26            add HoleParticle vector to Channel
27            add ParticleMinusHole vector to Channel
28            add HoleMinusParticle vector to Channel
29          end
30        end
31      end
32 end

```

---

The CCD correlation energy is calculated as sum of traces of all matrices obtained by the multiplication of the corresponding interaction matrices and the

amplitudes matrices within all channels.

---

**Algorithm 5:** CCD correlation energy calculation for the channel solver.

---

```

input : Interaction matrix blocks and amplitude blocks
output: correlation energy
1 double CorrelationEnergy = 0.0
2 for channelIndex  $\leftarrow$  0 to ChannelVariety.size() do
3   CorrelationEnergy +=
4   | trace(m_hhppVBlock.at(channelIndex).getMatrix()*
5   | m_pphhTBlock.at(channelIndex).getMatrix())
6 end
7 return 0.25*CorrelationEnergy

```

---

## 6.6 Diagrams calculation

Up until then, we considered the simplest diagram  $L_1$  and the CCD correlation energy expression in the context of the CPU optimization. The mentioned expressions have the ordering of the indices such as it is naturally and easy to notice matrix-matrix multiplication. Let us consider  $L_3$  term from equation (6.2):

$$L_3 = \sum_{kc} w_{cj}^{kb} t_{ik}^{ac}. \quad (6.12)$$

Now it is not trivial to write this sum as matrix product. So reordering of the indices must be made:

$$\sum_{kc} w_{cj}^{kb} t_{ik}^{ac} \rightarrow \sum_{kc} \tilde{w}_{jb}^{kc} \tilde{t}_{kc}^{ai}. \quad (6.13)$$

Now it is obvious how to calculate sum in equation(6.13):

$$\tilde{L}_3 = \sum_{kc} \tilde{w}_{jb}^{kc} \tilde{t}_{kc}^{ai} = \tilde{W} \times \tilde{T}. \quad (6.14)$$

However this reordering cannot be implemented easily. One needs to take into account symmetries in the interaction. In the original  $L_3$  term (6.12) the spin and the total momentum conservation must be satisfied for the following two-states sets of quantum numbers:

$$c + j = k + b = a + c = i + k. \quad (6.15)$$

The elements of matrices  $\tilde{W}$  and  $\tilde{T}$  must satisfy the same restriction (6.15). To do it we need to fill these matrices with the elements obeying the following conservation restriction:

$$j - b = k - c = a - i = k - c, \quad (6.16)$$

which is nothing else but relation (6.15), written out according to the reordering scheme introduced in (6.13). From the discussion above it follows that to calculate  $L_3$  diagram as matrix product we need to construct interaction matrix block following restriction (6.16). This means that this matrix must be constructed from the vector containing the following two-body states: hole sp-state minus particle sp-state. This explains now the existence of lines (27-28) in listing (4). We refer to the described procedure as *decoupling*. Some texts[13] [23] introduce term *alignment* for this. Finally we should mention that  $\tilde{L}_3$  must be recoupled again to match the indexing scheme of amplitudes.

I the considered example pairs of states represent rows and columns. However  $Q_3$  and  $Q_4$  diagrams is impossible to decouple according to the similar scheme. To perform the calculation of these diagrams as matrix product, decoupling must be applied according the following scheme: the columns represent three states, and the rows represent one state. In table 6.1 all decouplings are represented for the CCD block solver.

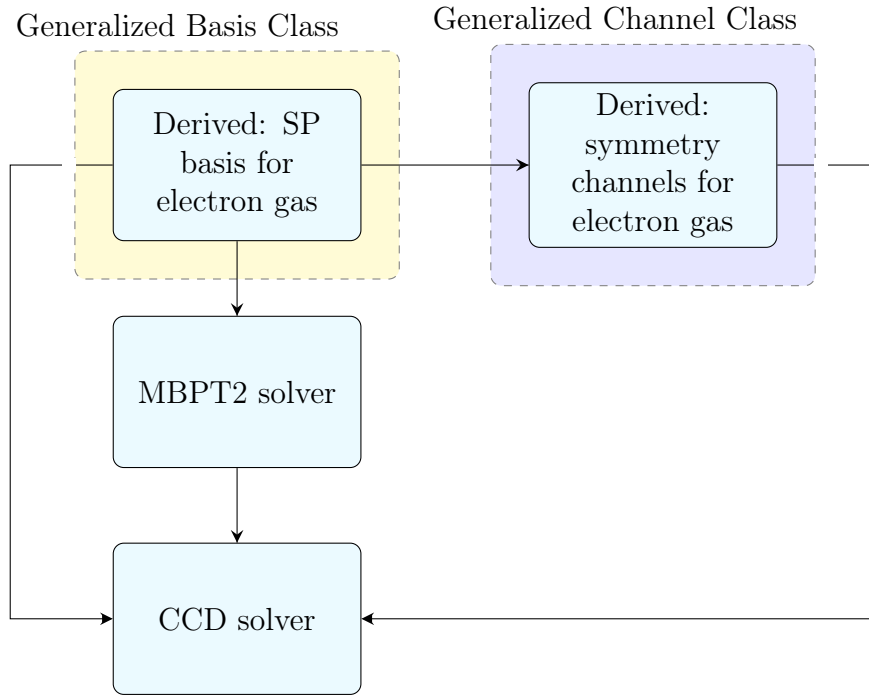
**Table 6.1:** Decoupling/Recoupling schemes for CCD diagrams.

Name	Expression	Factor	Permutation	Decoupling	Re-coupling
$L_1$	$\sum_{cd} w_{cd}^{ab} t_{ij}^{cd}$	$\frac{1}{2}$	—	$w_{cd}^{ab} t_{ij}^{cd}$	—
$L_2$	$\sum_{kl} w_{ij}^{kl} t_{kl}^{ab}$	$\frac{1}{2}$	—	$t_{kl}^{ab} w_{ij}^{kl}$	—
$L_3$	$\sum_{kc} w_{cj}^{kb} t_{ik}^{ac}$	1	$\hat{P}(ij ab)$	$w_{ck}^{jb} t_{ai}^{ck}$	$L_3'_{ai}{}^{jb} \rightarrow t_{ij}^{ab}$
$Q_1$	$\sum_{klcd} w_{cd}^{kl} t_{ij}^{cd} t_{kl}^{ab}$	$\frac{1}{4}$	—	$t_{ij}^{cd} w_{cd}^{kl} t_{kl}^{ab}$	—
$Q_2$	$\sum_{klcd} w_{cd}^{kl} t_{ik}^{ac} t_{jl}^{bd}$	1	$\hat{P}(ij)$	$t_{ai}^{ck} w_{ck}^{ld} t_{ld}^{dj}$	$Q_2'_{ai}{}^{dj} \rightarrow t_{ij}^{ab}$
$Q_3$	$\sum_{klcd} w_{cd}^{kl} t_{ik}^{dc} t_{lj}^{ab}$	$-\frac{1}{2}$	$\hat{P}(ij)$	$t_l^{abj} w_{kcd}^l t_i^{kcd}$	$Q_3'_{i}{}^{abj} \rightarrow t_{ij}^{ab}$
$Q_4$	$\sum_{klcd} w_{cd}^{kl} t_{lk}^{ac} t_{ij}^{db}$	$-\frac{1}{2}$	$\hat{P}(ab)$	$t_{klc}^a w_d^{klc} t_{bij}^d$	$Q_4'_{a}{}^{bij} \rightarrow t_{ij}^{ab}$

## 6.7 The block solver workflow

The pointer to the generalized basis class is passed to the solver, so that the methods inside the solver have access to the functions from the basis class. These

functions are, for example, the calculation of the one- and two-body matrix elements, etc. The benefit of this approach is the following: the solver has access to the special method, saying `basis->TBME(p, q, r, s)`. If one wants to perform the CCD calculation of any other system (water molecule, quantum dot etc.), the only requirement is to create a new basis derived class. Developer in this case is forced to create a new virtual function with prototype `double TBME(int, int, int, int)` in this new derived class. And no modifications are required for the solver code. The same arguments can be mentioned for the class describing symmetry channels of the system.



**Figure 6.2:** Classes needed for the CCD solver.

In our implementation, MBPT2 solver is used with symmetry channels. These channels are created inside the MBPT2 solver class. We need only particular state combinations, and the total number of channels is lower than one for the CCD solver. Thus standalone MBPT2 solver runs faster. However for the sake of generality, program architecture should be changed: MBPT2 solver should take channel setup as input. The CCD solver needs knowledge about system, particularly, the single particle basis and the symmetry channels. Since CCD solver is iterative, it needs initial guess for the amplitudes and the correlation energy. The last two are provided by the MBPT2 solver. On figure 6.2 the program work-flow is presented.

## 6.8 Program structure and hierarchy

We outline the structure of the program used for the calculations of the CCD equations for the electron gas in this thesis. We briefly describe functionality of the classes. For the detailed implementation of various methods we refer to the source code. All the names of functions, methods classes and variables were intended to be self-explanatory. We hope that comments in the code can provide suitable illumination. Following classes are needed to perform calculations:

- `generalSPclass`
  - `electronGasSPBasis`
- `CartesianState`
- `channelset`
  - `electrongaschannelset`
- `channel`
- `channelindexpair`
- `symblock`
- `ccd`
- `mbpt2`
- `mbpt2channels`

### 6.8.1 Single-particle basis class

This is the base class containing pure virtual functions. An Object of `generalSPclass` is given as input for all solver classes to provide them with information about the quantum system of interest. The two-body matrix elements of the Hamiltonian, single-particle energies, number of states and Fermi level are the typical quantities needed for almost every solver.

### 6.8.2 The electron gas single-particle basis class

This class is a derived class where the implementation of methods, described in `generalSPclass`, must be placed. As it follows from the name of this class it provides typical quantities of interest for the electron gas. The number of shells and electrons together with Wigner-Seitz radius are needed for the constructor, which calls two methods: `setUpUnits` and `setUpStatesCartesian`. That

creates a vector of objects `CartesianState`, where each object represents a single-particle state as a set of quantum numbers(see table 5.1). `setUpUnits` function is needed to create output of such key quantities as, for example, energy in atomic units. The other important methods which are heavily used in other classes are:

- `TBME` returns anty-symmetrized two-body matrix elements of the Hamiltonian. For the electron gas it is simply hard-coding of equation (5).
- `getQuantumStates`, `getQuantumStatesNumber` and `calculateReferenceEnergy` functions provide an useful information fo debug purposes.
- `computeSPenergies` returns vector with the single-particle energies in the Hartree-Fock basis.
- `sumState`, `subtractState` are used for channels setup in CCD solver and for the symmetry restriction check in CCQMC solver. These functions make a sum or a difference of two single-particle states.

### 6.8.3 Single-particle state class

Every instance of `CartesianState` class is a set of four integers, representing  $n_x$ ,  $n_y$ ,  $n_z$  and  $s$ . Generally, this is a small class which acts like a data structure in C++. Our motivation to have it as a separate class is based on a possibility to consider Cartesian state as a derived class of some generalized class. That improves generalization of the code by allowing to include easily other coordinate systems or dimensionalities.

### 6.8.4 Symmetry channels class

We present the base class for setting up the symmetry channels. This class contains virtual functions only. For the CCD solver there are three functions.

- `setUpChannels`
- `setUpChannelsQ3`
- `setUpChannelsQ4`

`setUpChannels` sets up channels for  $L_1 - Q2$  diagrams. Notation we use for different diagrams can be found in table 6.1.



### 6.8.5 The electron gas symmetry channels class

The implementation of the setup of the symmetry channels for the electron gas is located in this class. We have actually already covered this topic in section 6.5.

### 6.8.6 Channel class

Instances of this class act as an organized storage for single-particle states<sup>1</sup>. Objects of this class contain a set of vectors. Each vector corresponds to the particular interaction, which must obey symmetry constraints. These vectors are:

- `m_HoleHoleVec`
- `m_ParticleParticleVec`
- `m_ParticleHoleVec`
- `m_ParticleMinusHoleVec`
- `m_HoleParticleVec`
- `m_HoleMinusParticleVec`
- `m_ParticlePlusParticleMinusHoleVec`
- `m_HoleVec`
- `m_HolePlusHoleMinusParticleVec`
- `m_ParticleVec`

Each element in mentioned vectors is an object of class `channelindexpair`. For example, we consider amplitude  $t_{ij}^{ab}$ . As we know the value of this amplitude is non-zero if the following constrain is satisfied:

$$i + j = a + b, \quad (6.17)$$

for both the momentum and the spin. We store all hole-state indices  $i$  and  $j$  as a pair for a given channel in `m_HoleHoleVec`. Similarly we do with particle-state indices  $a$  and  $b$  and vector `m_ParticleParticleVec`. We must apologize here for a not a completely strict name of class `channelindexpair`, as it can hold one or even three indices, which are needed for the diagrams  $Q_3$  and  $Q_4$ .

---

<sup>1</sup>strictly speaking, we store indices of single-particle states in `m_shells` vector.

### 6.8.7 Symmetry block class

This class is used for the interaction matrices. The main purpose is to structure interactions for each channel into a matrix. It also serves the purpose of the creation of the index vectors, where we store one-to-one correspondence between the interaction matrix element and single-particle state number. Vectors with symmetry block objects are created inside `ccd solver` class. Since coupled cluster amplitudes are stored on the same form, this class is used to create amplitudes matrices with mapping as well.

### 6.8.8 CCD solver class

Here we discuss the main method of the `ccd` class, namely the `iterateCCD` class. Here one can find the main component of the CCD solver, which is an iterative procedure shown in listing 1. To start iterations we initialize interaction matrix blocks and amplitude matrices. It is done by the following functions:

- `setUpInteractionMatrixBlocks`
- `setUpInitialAmplitudes`
- `setUpInteractionMatrixBlocksQ3`
- `setUpInitialAmplitudesQ3`
- `setUpInteractionMatrixBlocksQ4`
- `setUpInitialAmplitudesQ4`

The result of these functions is a set of vectors with `symblock` objects. The complete list of these vectors is presented below:

<code>m_hhhhVBlock</code>	<code>m_pphhTBlock</code>
<code>m_ppppVBlock</code>	<code>m_pphhTBlockPrev</code>
<code>m_hMphMpVBlock</code>	<code>m_pMhhMpTBlock</code>
<code>m_hhppVBlock</code>	<code>m_pMhhMpTBlockPrev</code>
<code>m_hMppMhVBlock</code>	<code>m_ppMhTBlock</code>
<code>m_ppMhVBlock</code>	<code>m_ppMhTBlockPrev</code>
<code>m_hhMpVBlock</code>	<code>m_hhMpTBlock</code>
<code>m_pphhVBlock</code>	<code>m_hhMpTBlockPrev</code>

They are filled up in a such way that every `symblock` object in any of mentioned vector has the index which matches index of the channel. The most important data is stored in the interaction matrix of the `symblock` object. For certain channels some matrices, of course, are impossible to construct.

Inside the CCD iteration amplitudes are calculated and updated by the following functions: `calculateAmplitudes` and `updateAmplitudes`. Inside these functions amplitudes are calculated by the evaluation of the diagrams term by term, sometimes including decoupling and recoupling, that was described in section 6.6.

### 6.8.9 MBPT2 classes

`mbpt2` class is a brute-force implementation of the MBPT2 equations via sums and for-loops. This class can be useful as a unit test when adding new systems to the code. `mbpt2channels` class serves as input for the CCD solver. It is a small class with two major methods: `setUpChannels` and `calculateEnergyMatMult`. Reasons for having a separate function to setup symmetry channels in side this class were presented in section 6.7.

## 6.9 Possible optimization

### Intermediates

So-called *intermediates* provides us with a way of achieving reduction of execution time of our CCD code. According to [23] speed up gained by using them can reach values up to 25% which is a significant improvement. What is even more important is the usage of the intermediates for higher truncation levels. The basic idea is to find common quantities for different diagrams calculated repeatedly during one iteration. It seems to be logical to calculate these quantities only once in the beginning of each iteration and reuse them later for a diagram. As an example, we consider the last diagram of equation (6.2):

$$Q_4 = -\frac{1}{2}\hat{P}(ab) \sum_{klcd} w_{cd}^{kl} t_{lk}^{ac} t_{ij}^{db} = -\frac{1}{2}\hat{P}(ab) \sum_c t_{ij}^{ac} \sum_{kld} w_{cd}^{kl} t_{kl}^{bd} = -\frac{1}{2}\hat{P}(ab) \sum_c t_{ij}^{ac} I, \quad (6.18)$$

where intermediate  $I$  defined by equation (6.19) and it is calculated only once during iteration.

$$I = \sum_{kld} w_{cd}^{kl} t_{kl}^{bd} \quad (6.19)$$

Considered  $Q_4$  computed without intermediates must be calculated for all  $t_{ij}^{ab}$ , and it is easy to see that this operation scales as  $\mathcal{O}(N_{particles}^4 \cdot N_{holes}^4)$  number of flops. Implementation of intermediate 6.19 reduces power of  $N_{holes}^4$  by two.

### Mapping

To calculate permutations and to update amplitudes we need to know where elements should be placed in the original matrix. In the end of the CCD iteration

```

for(unsigned int chan = 0; chan < ChannelVariety.size();
    chan++){
    channel onechannel = ChannelVariety.at(chan);
    symblock amplitudes = m_pMhhMpTBlock.at(chan);
    for(unsigned int ai = 0; ai <
        onechannel.m_ParticleMinusHoleVec.size(); ai++){
        for(unsigned int jb = 0; jb <
            onechannel.m_HoleMinusParticleVec.size(); jb++){
            int a = amplitudes.getRowMap(ai).first();
            int i = amplitudes.getRowMap(ai).second();
            int j = amplitudes.getColMap(jb).first();
            int b = amplitudes.getColMap(jb).second();
            double value = updateAmplitudesL3(a,b,i,j);
            m_pMhhMpTBlockPrev.at(chan).setElement((int)ai,
                (int)jb, value);
        }
    }
}

```

**Figure 6.3:** Update of  $L_3$  diagram.

we obtain amplitudes in a form of  $t_{ij}^{ab}$  (particle + particle = hole + hole), so we need to know which elements to place into, for example, (particle - hole = hole - particle) matrix, which is needed for  $L_3$  diagram. In addition to that information about place in the new matrix is needed. `getRowMap` and `getColMap` functions have the simplest possible implementation for these mentioned one-to-one correspondence. Given `channelindexpair` object as an input, they return an integer index, so we know where this object is located in the mapping matrix. Code listing on figure 6.3 shows how these functions are called. We planned to use hash tables from the standard library and store `ChannelIndexPair` as value and index as a key. That could provide us with a faster search algorithm. However due to an unexpected bug, we have not managed to make it work.

Mapping of channels with unique value of  $\mathbf{k}$  should be considered as an alternative scheme for the organization of channels. As it was already described in sections 6.8.8, 6.8.6 and 6.8.7, we create a vector of channel objects, where each one contains information about unique value of interaction symmetry. When calculating one chosen diagram, say  $L_3$ , loop goes over all channels in `ChannelVariety` vector and calculates  $L_3$  with the code presented on figure 6.4. It can be seen from the listing that a simple storing scheme for channels by bonding them to the ascending index of the vector element comes with a drawback that is absence of flexibility. Each particular channel does not necessarily contains all possible block-matrices. In the listing there is check if they contains non-zero `m_ParticleMinusHoleVec` and `m_HoleMinusParticleVec` blocks. Instead it can be much more efficient to go through channels that already have

```

void ccd::L3(){
    for(unsigned int chan=0; chan < ChannelVariety.size();
        chan++){
        channel onechannel = ChannelVariety.at(chan);
        if (onechannel.m_ParticleMinusHoleVec.size()*
            onechannel.m_HoleMinusParticleVec.size() != 0){
            m_pMhhMpTBlock.at(chan).setMatrix(
                m_pMhhMpTBlockPrev.at(chan).getMatrix()*
                m_hMphMpVBlock.at(chan).getMatrix());
        }
    }
}

```

**Figure 6.4:**  $L_3$  diagram calculation.

these non-zero blocks.

Each channel can be distinguished from other by its unique combination of quantum numbers(lets denote them as  $U_i$ ). We can iterate before the channel setup and check how many  $U_i$  combinations exist. After that it can be created an one-dimensional array of symmetry block objects with a link between  $U_i$  and index of an object in this array. Afterward some mapping can be created, where we connect indices in the array with particular non-zero blocks.

### Parallelization

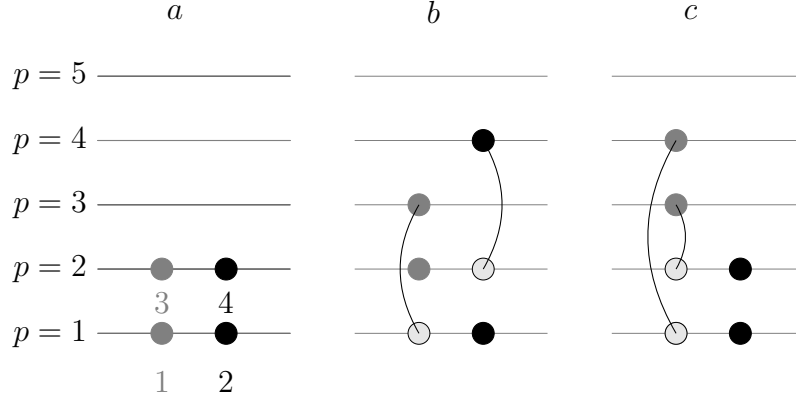
Message Passing Interface(MPI) and Multi Processing(MP) are tools which come to mind last, after all mathematical optimization is implemented. MPI is a standard developed for the distributed memory architectures, while MP is an application programming interface used for the shared memory architecture.

All mentioned optimization technicalities were skipped in our implementation as the main goal of the CCD solver was to produce the benchmark information for CCQMC solver. We are able to run simple calculations for the 3D electron gas with 14 electrons up to 13 shells on a typical personal computer.

## 6.10 Occupation Number Representation of SD

We consider a common energy diagram and show how it is possible to translate it into occupation number representation.

Let us consider a simple example of a system with four electrons and five energy levels. The reference determinant is shown on figure 6.5(a). It is possible to represent reference as a sorted array where index of each element refers to the state number, and its value reflects the information if this state is filled with the



**Figure 6.5:** The open-shell electron gas with four electrons and five energy levels. All states with spin up are gray, the opposite spin states are black. The labeling scheme of the single-particle states is shown on figure (a).

particle or not. For example, the determinant corresponding to figure 6.5(a) can be written as follows:

$$|D_0\rangle = |\phi_1\phi_2\phi_3\phi_4\rangle = |1_11_21_31_40_50_60_70_80_90_{10}\rangle.$$

Storing the array of integers, which can have just two values(0 and 1), is not an optimal way since 32 or 64 bits are required to store one integer number in memory. The best solution is to use bit-strings here, which are just a sequence of bits. Every bit can have just two values: 0 or 1. Since there is no smaller unit of information than bit, it is the most memory-efficient way to store Slater determinants. It can be crucial for us as the full space of the determinants is formed as  $\binom{2M}{N}$ , where  $N$  is number of the occupied orbitals out of  $2M$  orbitals.

In C++ we used BOOST dynamic\_bitset library as it allows to create bit-strings of dynamic size in contradiction to the standard library where the length of the bit-string must be defined during compilation time. The side effect of the occupation number representation of determinants as a bit-strings is inbuilt implementation of the second-quantized excitation operators through bitwise operation XOR(of course, up to sign). For example, we look at 6.5(b) where particle 1 is excited to state number 5 and particle from 4 is excited to state 8. To generate the determinant corresponding to 6.5(b) we act with two excitation operators  $\hat{a}_1^5\hat{a}_4^8$  on the reference. In the bit-string representation, excitations the operators look as follows:

$$1_10_20_31_41_50_60_71_80_90_{10}.$$

Applying this string through XOR operation to the reference we obtain excited

determinant up to sign:

$$\begin{array}{c}
 1001100100 \\
 XOR \\
 1111000000 \\
 \downarrow \\
 0110100100
 \end{array}$$

Additionally we need to stress that the occupation numbers represent only one of the  $N!$  determinants, which are possible to make from  $\phi_1$  to  $\phi_N$  single-particle functions. To illustrate this we show simple example for  $N = 3$ :

$$|11001\rangle = |1, 2, 5\rangle = -|2, 1, 5\rangle = -|5, 2, 1\rangle = -|1, 5, 2\rangle = |2, 5, 1\rangle = |5, 1, 2\rangle.$$

All these determinants are clearly linearly dependent.

To be consistent with the mathematical formalism describing action of creation/annihilation operators we need to determine sign. By definition the action of these operators on the determinant is written as follows:

$$\hat{a}_i |a, b, \dots, i, \dots\rangle = (-1)^{i-1} |a, b, \dots, i, \dots\rangle,$$

and

$$\hat{a}_i^\dagger |a, b, \dots, j, \dots\rangle = (-1)^{i-1} |a, b, \dots, i, j, \dots\rangle.$$

Later in this chapter it is shown that we operate with the following excitation operators:

$$\hat{t}_{ij\dots k}^{ab\dots c} = \hat{a}_a^\dagger \hat{a}_b^\dagger \dots \hat{a}_c^\dagger \hat{a}_k \dots \hat{a}_j \hat{a}_i,$$

where  $i < j < \dots < k$  and  $a < b < \dots < c$ . Hence the application of this excitor to the reference might result in a change of sign, i.e:

$$\hat{t}_{ij}^{ab} |D_0\rangle = -|D_{ij}^{ab}\rangle.$$

As a more concrete example, let us consider a set of states with the reference  $|D_0\rangle = |1, 2, 3\rangle$ . The action of two different excitors,  $\hat{t}_{13}^{58}$  and  $\hat{t}_{12}^{58}$ , is written out explicitly:

$$\begin{aligned}
 \hat{t}_{13}^{58} |1, 2, 3\rangle &= +\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_3 \hat{a}_1 \hat{a}_2^\dagger \hat{a}_3^\dagger |-\rangle \\
 &= +\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_3 \hat{a}_2^\dagger \hat{a}_3^\dagger |-\rangle \\
 &= -\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_3 \hat{a}_3^\dagger \hat{a}_2^\dagger |-\rangle \\
 &= -\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_2^\dagger |-\rangle \\
 &= +\hat{a}_5^\dagger \hat{a}_2^\dagger \hat{a}_8^\dagger |-\rangle \\
 &= -\hat{a}_2^\dagger \hat{a}_5^\dagger \hat{a}_8^\dagger |-\rangle \\
 &= -|2, 5, 8\rangle
 \end{aligned}$$

$$\begin{aligned}
\hat{t}_{12}^{58} |1, 2, 3\rangle &= +\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_2 \hat{a}_1 \hat{a}_1^\dagger \hat{a}_2^\dagger \hat{a}_3^\dagger |-\rangle \\
&= +\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_2 \hat{a}_2^\dagger \hat{a}_3^\dagger |-\rangle \\
&= +\hat{a}_5^\dagger \hat{a}_8^\dagger \hat{a}_3^\dagger |-\rangle \\
&= -\hat{a}_5^\dagger \hat{a}_3^\dagger \hat{a}_8^\dagger |-\rangle \\
&= +\hat{a}_3^\dagger \hat{a}_5^\dagger \hat{a}_8^\dagger |-\rangle \\
&= +|2, 5, 8\rangle
\end{aligned}$$

## 6.11 CCQMC Cluster Size

In chapter 4 we showed that instead of the direct integration of the imaginary-time Schrödinger equation we rewrite it in a such way, that it is possible to sample it. Using the coupled cluster expansion we can determine amplitudes as iteration procedure from time  $\tau$  to  $(\tau + \delta\tau)$ :

$$t_{\{i\}}(\tau + \delta\tau) = t_{\{i\}}(\tau) - \delta\tau \langle D_{\{i\}} | (\hat{H} - E) | \Psi_{CC}(\tau) \rangle. \quad (6.20)$$

To simulate this equation one needs to sample the coupled cluster wavefunction:

$$\Psi_{CC}(\tau) = [1 + \sum_{\mathbf{i}} \underbrace{t_{\mathbf{i}} \hat{a}_{\mathbf{i}}}_{\text{size 1}} + \sum_{\mathbf{ij}} \underbrace{t_{\mathbf{i}} t_{\mathbf{j}} \hat{a}_{\mathbf{i}} \hat{a}_{\mathbf{j}}}_{\text{size 2}} + \sum_{\mathbf{ijk}} \underbrace{t_{\mathbf{i}} t_{\mathbf{j}} t_{\mathbf{k}} \hat{a}_{\mathbf{i}} \hat{a}_{\mathbf{j}} \hat{a}_{\mathbf{k}}}_{\text{size 3}} + \dots] |D_0\rangle \quad (6.21)$$

We choose a single term from expansion (6.21) at the time. This process is known as the collapse of the wavefunction to the cluster. We define cluster size by the number of excitors it contains:  $t_{\mathbf{i}} \hat{a}_{\mathbf{i}}$  corresponds to the cluster of size one,  $t_{\mathbf{i}} t_{\mathbf{j}} \hat{a}_{\mathbf{i}} \hat{a}_{\mathbf{j}}$  represents the cluster of size two, etc.

We set maximal cluster size to be three. Additionally we consider just single, double and, possibly, triple excitations. That means that the cluster of size three is just three single excitations, while cluster of size one can be single, double or triple excitation. The detailed relations between the cluster size and the excitation level are shown in table 6.2.

## 6.12 CCQMC Solver

As it was already mentioned in chapter 4, we use a variable  $N_0$  (see equation (4.22)) to preserve the intermediate normalization. If we omit  $\tau$ -dependence (we consider the exponential ansatz of the wave-function during one  $\tau$ -iteration), the expansion of the coupled cluster wavefunction (6.21) becomes the following:

$$|\Psi_{CC}\rangle = [N_0 + \sum_{\mathbf{i}} t_{\mathbf{i}} \hat{a}_{\mathbf{i}} + \frac{1}{N_0 2!} \sum_{\mathbf{ij}} t_{\mathbf{i}} t_{\mathbf{j}} \hat{a}_{\mathbf{i}} \hat{a}_{\mathbf{j}} + \frac{1}{N_0^2 3!} \sum_{\mathbf{ijk}} t_{\mathbf{i}} t_{\mathbf{j}} t_{\mathbf{k}} \hat{a}_{\mathbf{i}} \hat{a}_{\mathbf{j}} \hat{a}_{\mathbf{k}} + \dots] |D_0\rangle. \quad (6.22)$$



**Table 6.2:** Cluster size and excitation level. Table explains how we can construct cluster of certain size using different excitation operators. For example, cluster of size three can be constructed just from single excitation to avoid reaching higher excitations than level three.

cluster size	Single	Double	Triple
0	—	—	—
1	$S$	$D$	$T$
2	$SS$	$DS/SD$	—
3	$SSS$	—	—

We can refer to  $N_0$  as the population of excips on the reference. However it is not true, because these excips are not the same kind of excip, populating the excitation space. In equation (4.22) the cluster operator  $\hat{T}$  is by definition:

$$\hat{T}(\tau) = \hat{T}_1(\tau) + \hat{T}_2(\tau) + \hat{T}_3(\tau) + \dots, \quad (6.23)$$

where, for example,  $\hat{T}_2(\tau)$  contains all the double-orbital replacements of the reference determinant through excitors  $\hat{a}_i$ :

$$\hat{T}_2 = \sum_{\mathbf{j} \in \text{double replacements}} t_{\mathbf{j}}(\tau) \hat{a}_{\mathbf{j}}, \quad (6.24)$$

where  $t(\tau)$  is a parametrization of the exponential expansion of the wavefunction. In the deterministic coupled cluster theory we used equations (2.9) to find their values. In the CCQMC we use similar to the DMC method by considering a population of walkers  $N_w$ . Each walker is located on determinant  $\mathbf{j}_\alpha$  and have sign  $s_\alpha = \pm$ . We define amplitude  $t_i$  to be proportional to the population of walkers. It is important that population sign of walkers is taken into account:

$$t_i \propto N_i = \sum_{\alpha} s_{\alpha} \delta_{i, i_{\alpha}}. \quad (6.25)$$

However total population is simply a sum of absolute values of the populations of all walkers:

$$N_w = \sum_i |N_i|. \quad (6.26)$$

We outline the general structure of the CCQMC solver. We simulate equation (6.20). This is performed by the `applyCCQMC` method of the `ccqmc` class. As it has been already mentioned, to evaluate a non-trivial term  $\langle D_{\{i\}} | (\hat{H} - E) | \Psi_{CC}(\tau) \rangle$  from equation (6.20) we perform two sampling processes. They are the sampling

of the wavefunction (6.22) and the sampling of the action of the Hamiltonian. The possible strategy here is to choose randomly clusters from the wavefunction (6.21) such, that the cluster size follows the exponential distribution:

$$p_{size}(s) = \frac{1}{2^{s+1}}, \quad (6.27)$$

where  $s$  is a cluster size. Instead we sample each cluster size separately. So during one  $\delta\tau$  iteration we call consistently functions `sampleReference`, `sampleClusterSizeOne` and `sampleClusterSizeTwo`. The sampling of the action of the Hamiltonian is included in these functions and will be covered below. The result of these mentioned function is a two vectors: `TemporaryStorage` consisting of excitors and `TemporaryStoragePopulation` which holds the population of excips on these excitors. During annihilation step in the end of the  $\tau$ -loop, these vectors are being merged with `StorageCumulativeDeterminants` and `StorageCumulativePopulation`. Excitors with no excips are deleted from `StorageCumulativeDeterminants` vector. The last is crucial since the length of the `StorageCumulativeDeterminants` vector is used in sampling. The abobe described procedure is presented in listing 6 as a pseudo-code.

---

**Algorithm 6:** The CCQMC iteration.

---

**input :**  $\delta\tau$ -timestep, max value of  $\tau_{max}$ , initial population  $N_0$   
**output:** Vector with population, vector with excitors

```

1 for  $\tau \leftarrow 0$  to  $\tau_{max}$  do
2   sampleReference  $N_0$  times
3   sampleClusterSizeOne  $\frac{2}{3}N_{ex}$  times
4   sampleClusterSizeTwo  $\frac{1}{3}N_{ex}$  times
5   merge current population with total population
6   annihilation step
7    $\tau = \tau + \delta\tau$ 
8 end
```

---

## 6.13 CCQMC Sampling

Because of the complicity of the sampling procedure, it might be best to present first a simplified block diagram. We present it in figure 6.6. Bearing in mind this diagram, one can delve into technicalities of the sampling scheme for the CCQMC algorithm. The sampling of the wavefunction involves two random events. First, we select the cluster size  $s$  randomly, according to the exponential distribution (6.27). In our implementation we sample different cluster sizes various number

of times. Second,  $s$  excitors are selected within the chosen cluster size. In this case the conditional probability to choose  $s$  excitors stands as:

$$p_{\text{clust}}(e|s) = s! \prod_{i=1}^s \frac{|N_i|}{N_{ex}}, \quad (6.28)$$

where the factor  $s!$  represents the number of permutations in the construction of a cluster. The quantity  $N_{ex}$  is the total number of the excitors which can be sampled in the system (doubles, if we choose the CCD truncation). According to the multiplication theorem on probability, one can sample all possible combinations of excitors with the total probability:

$$p_{\text{select}}^{\text{cluster}}(e) = p_{\text{size}}(s)p_{\text{clust}}(e|s) = \frac{s! \prod_{i=1}^s |N_i|}{2^{s+1} N_{ex}}. \quad (6.29)$$

One sample of the wavefunction can be described by equation (6.29), and we use this sample  $N_a$  times. So, naturally, one definite cluster can be sampled with probability  $\frac{1}{N_a}$ . However, it is easy to notice that the above mentioned two random processes will change  $N_a$  so, that it decreases. In this case the probability is  $\frac{1}{N'_a}$ , where  $N'_a = N_a p_{\text{select}}^{\text{cluster}}$ . After the wavefunction collapsed to a given excitation cluster, the result is  $\Psi_{CC}(\tau) \rightarrow A |D_m\rangle$ , where  $A$  is the amplitude of the constructed cluster and is defined as follows:

$$A = N_0 \prod_{i=1}^s \frac{N_i}{N_0}, \quad (6.30)$$

where  $N_i$  is the population of excips on the excitor  $i$ . The last term of equation (6.20) is now simplified into the following expression:

$$\delta\tau A \langle D_{\{i\}} | (\hat{H} - S) | D_m \rangle \quad (6.31)$$

According to the scheme in chapter 4, the sampling of the action of the Hamiltonian is performed by choosing random single or double excitations of  $|D_m\rangle$  for the bra-part  $D_{\{i\}}$ . We denote the obtained determinant as  $D_n$ . The Hamiltonian matrix element then became  $\langle D_n | \hat{H} | D_m \rangle = H_{nm}$ . The expression (6.31) is simplified as follows:

$$\delta\tau A H_{nm} \quad (6.32)$$

We spawn an excip on a  $D_n$  determinant with a probability proportional to (6.32), which includes the physics of the system. However we used quite complicated series of random events to obtain expression (6.32). So to unbiased the sampling we need to include combinatorial probabilities for the spawning of an excip. We multiply  $\delta\tau A H_{nm}$  by the probability discussed above:  $\frac{1}{N'_a}$ , and the probability for spawning is defined by the following expression:

$$\delta\tau A H_{nm} \frac{1}{N_a p_{\text{select}}^{\text{cluster}}} = \delta\tau A H_{nm} \frac{1}{N_a p_{\text{size}}(s) p_{\text{clust}}(e|s)}. \quad (6.33)$$

The only remaining factor is the probability for choosing the bra-part during the sampling of the action of the Hamiltonian. Thus the final probability for spawning can be written as follows:

$$P_{\text{spawn}} = \underbrace{\delta\tau|AH_{nm}|}_{\text{physics}} \underbrace{\frac{1}{N_a p_{\text{size}}(s) p_{\text{clust}}(e|s)}}_{\text{sampling ket-part}} \underbrace{\frac{1}{P_{\text{gen}}}}_{\text{sampling bra-part}}, \quad (6.34)$$

where  $P_{\text{gen}} = \frac{1}{N_{s+d}}$  is the probability to choose single or double excitations among all available.

The probability for an excip to be killed/cloned is proportional to the diagonal matrix elements of the Hamiltonian:

$$P_{\text{death}} = \delta\tau|A(H_{nn} - S)| \frac{1}{N_a p_{\text{size}}(s) p_{\text{clust}}(e|s)}, \quad (6.35)$$

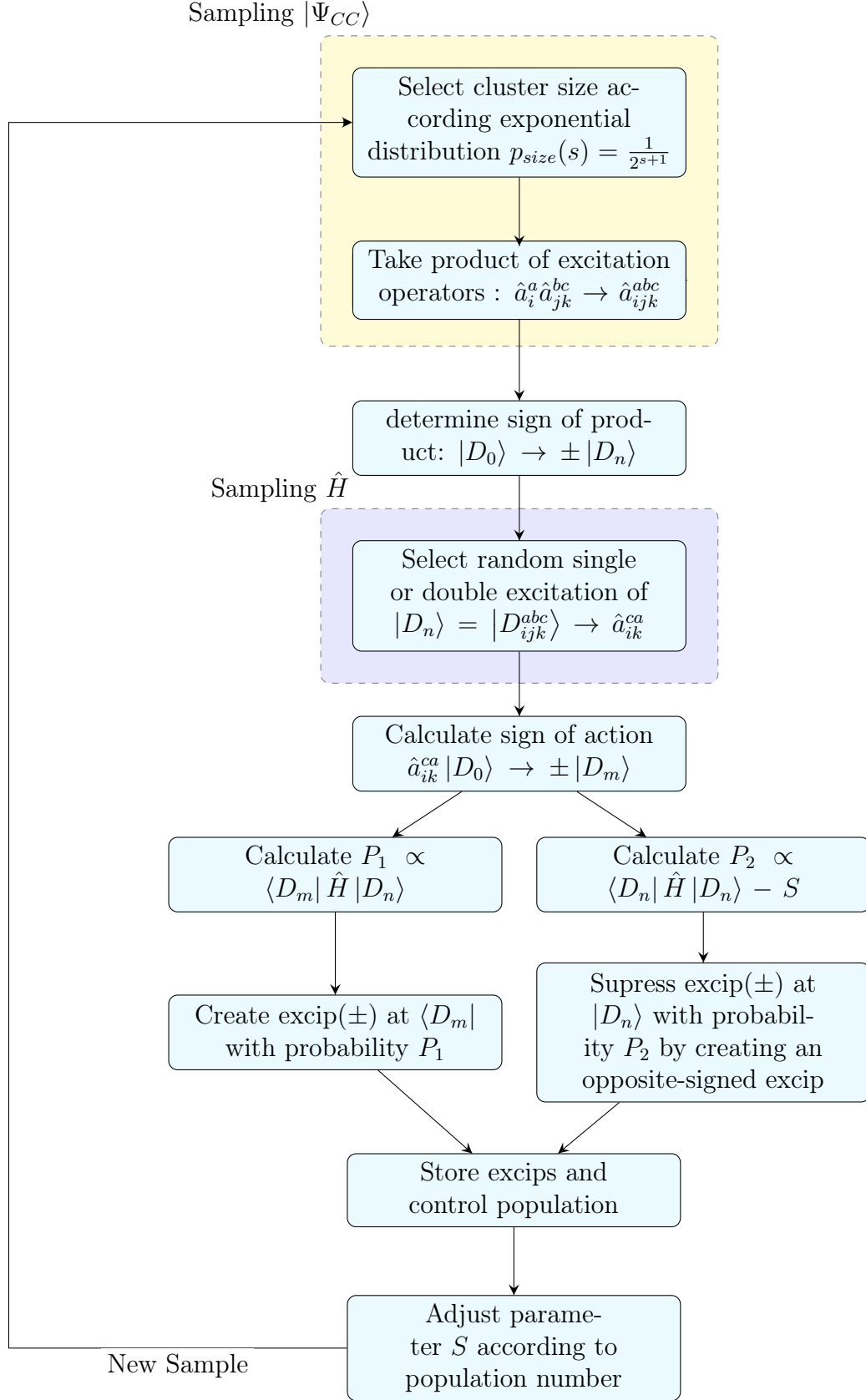


Figure 6.6: CCQMC sampling work-flow.

## 6.14 Sampling of the action of the Hamiltonian

The sampling of the action of the Hamiltonian is performed by choosing a random single or double excitation of the collapsed wavefunction  $|D_m\rangle$  for the bra-part  $D_{\{i\}}$ . We denote the obtained determinant as  $D_n$ .

For a double excitation one can write:

$$|D_m\rangle = a_r^\dagger a_s^\dagger a_q a_p |D_n\rangle, \quad (6.36)$$

where the orbitals  $p, q$  are chosen from the occupied set of orbitals, while  $r, s$  are selected from the unoccupied orbitals with respect to the  $D_n$ . The conditional probability is then can be written as follows:

$$p_{gen}(n|m) = p(r, s|p, q)p(p, q), \quad (6.37)$$

where  $p(p, q)$  is the probability to select a pair  $p, q$  in  $D_n$ , and  $p(r, s|p, q)$  is the probability to select pair  $r, s$  in condition that pair  $p, q$  is selected. According to the Slater-Condon rules we can select a single excitation of  $|D_m\rangle$  as well. In this case, denoting this excitation as:

$$|D_m\rangle = a_r^\dagger a_p |D_n\rangle,$$

the expression for the probability can be written:

$$p_{gen}(n|m) = p(r|p)p(p). \quad (6.38)$$

In general  $p_{gen}(n|m)$  is calculated using either double or single excitation:

$$p_{gen}(n|m) = \begin{cases} p(r, s|p, q)p(p, q) \cdot P_{\text{double}}, & \text{for double excitation} \\ p(r|p)p(p) \cdot (1 - P_{\text{double}}), & \text{for single excitation,} \end{cases}$$

where factors  $P_{\text{double}}$  and  $(1 - P_{\text{double}})$  were used for normalization.  $P_{\text{double}} \in (0, 1)$  represents probability of choosing double excitation. For the electron gas it is not possible to choose single excitations, and, hence,  $p_{gen}$  is defined by equation (6.37).

Let us consider the second factor of equation (6.37), representing the probability to select pair  $p, q$  in  $D_n$ . For the  $N$ -particle the probability to select *uniformly* a pair is the following:

$$p(p, q) = \binom{N}{2}^{-1} = \frac{2}{N(N-1)}. \quad (6.39)$$

Now we consider conditional probability  $p(r, s|p, q)$  from equation (6.37). Having selected  $p$  and  $q$ , we could either choose  $r$  and then  $s$  or vice versa.

$$p(r, s|p, q) = p(r|s, p, q)p(s|p, q) + p(s|r, p, q)p(r|p, q). \quad (6.40)$$

We select an  $s$  orbital with probability  $p(s|p, q)$  given that we have already selected a pair  $p, q$ . After this we choose the second orbital with the probability  $p(r|s, p, q)$ . A similar line of reasoning is applied to the second term in equation (6.40). The term  $p(s|p, q)$  is defined by the number of available orbitals, that fulfill certain constraints. For the system under consideration these are the conservation of momentum and spin. For a given  $p, q$  pair we need to check if orbital  $s$  satisfies:

$$|\mathbf{k}_p + \mathbf{k}_q - \mathbf{k}_s| < R_F, \quad (6.41)$$

which reflects that a combination of quantum numbers lies within the Fermi sphere. For the electron gas the fourth orbital is completely defined by the selection of the previous three, and hence, the remaining probability is:

$$p(r|s, p, q) = p(s|r, p, q) = 1. \quad (6.42)$$

## 6.15 The Hamiltonian matrix elements

For double excitation:

$$|D_m\rangle = a_r^\dagger a_s^\dagger a_q a_p |D_n\rangle. \quad (6.43)$$

In this case the corresponding matrix element then is:

$$\langle D_n | \hat{H} | D_m \rangle = \langle rs|pq \rangle_{AS}. \quad (6.44)$$

For single excitation:

$$|D_m\rangle = a_r^\dagger a_p |D_n\rangle. \quad (6.45)$$

And corresponding matrix element then is:

$$\langle D_n | \hat{H} | D_m \rangle = \sum_k^{N-1} \langle rk|pk \rangle_{AS}, \quad (6.46)$$

where summation goes over all common orbitals for  $D_n$  and  $D_m$ . The diagonal matrix elements are calculated with the following expression:

$$\langle D_n | \hat{H} | D_n \rangle = \sum_{p \in n} \langle p | \hat{h} | p \rangle + \frac{1}{2} \sum_{p, q \in n}^{N-1} \langle pq|pq \rangle_{AS}. \quad (6.47)$$

## 6.16 The sign of excip

The signs of the excips are crucial for the successful CCQMC simulations. In addition to the explicit minus sign in the propagation equation (4.17), one should consider the following sources of the sign change:

1. The sign of the Hamiltonian matrix element  $H_{nm}$ .
2. The sign of the parent excip. In the case of the composed cluster the sign of the amplitude of this cluster must be considered. The amplitude of the excips on a cluster of excitors is the product of the individual amplitudes, corresponding to the excitors, which construct the cluster.
3. Combining excitors to form a cluster, we perform reordering of the creation/annihilation operators. Odd number of such permutations causes a sign change in the excip population.
4. Applying the randomly chosen excitor to the reference can result in a sign change. This was described in section 6.10.
5. Sampling the action of the Hamiltonian, we choose the excited determinant in the bra-part. This determinant is a result of the action of some excitor on the reference. Thus if we consider spawning an excip on the excitor which create this determinant, we must take into account the sign of the excitor relative to the reference determinant.

In order to observe the correct dynamics, we must carefully accumulate all negative signs and take them into account before spawning an excip.

## 6.17 CCQMC code structure

We were not able to find any description of the CCQMC algorithm implementation in literature. To our knowledge there is the only one implementation that exists, namely the one developed by the authors of the algorithm. When we started writing our code we knew nothing about the technicalities of CCQMC implementation. Therefore, we decided to start with just one class and keep the code as simple as possible to illuminate potential sources of errors.

During the development of the code, we were guided by the paradigm formulated by Kent Beck, "Make it work, make it right, make it fast". When implementing the coupled cluster doubles truncation for the electron gas we found it difficult to split the solver into classes. It was found later that the class separation is necessary for including other systems and higher truncation levels.

The main difficulty in the CCQMC algorithm is to implement the sign structure of the wavefunction correctly. While we use a bit string representation of determinants, we found it useful to start with a vector representation. Below we present a code that can help to understand the basic concept, however it should be considered as a kind of unit test and not a part of the algorithm. We consider determinant  $|D_{ij}^{ab}\rangle$  as a small vector, where we write all creation/annihilation operators in a predetermined order. We look at the following example:



$$|D_{ij}^{ab}\rangle \rightarrow [i, j : a, b]$$

This is helpful as it allows to construct the exact creation/annihilation string of operators in correct order:

$$[i, j : a, b] \rightarrow \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_i \hat{c}_j.$$

If during sampling we choose excitor  $\hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i$ , which by action on the reference collapses the wavefunction into  $-|D_{ji}^{ab}\rangle$ . Vector representations of mentioned excitors will be different:

$$[i, j : a, b] \neq [j, i : a, b].$$

This slows down the algorithm significantly, however it eliminates one source of errors: we don't need to obtain the phase by acting with one annihilation/creation operator on the given Slater determinant.

```

auto it = find_if(TemporaryStorage.begin(),
    TemporaryStorage.end(), [&SampledCluster](const
    std::vector<int>& EachDet) {return
    std::search(EachDet.begin(), EachDet.end(),
    SampledCluster.begin(), SampledCluster.end()) !=
    EachDet.end();});
if (it != TemporaryStorage.end()){
    auto index = std::distance(TemporaryStorage.begin(), it);
    TemporaryStoragePopulation.at((index)) +=
        (-1)*SignOfCluster;
} else {
    TemporaryStorage.push_back(SampledCluster);
    TemporaryStoragePopulation.push_back((-1)*SignOfCluster);
}

```

**Figure 6.7:** Finding of Slater determinant in vector representation.

Bitstring is much more convenient data type for working with determinants, as it has been already discussed. However additional calculation are required when extracting a particular determinant from the cumulative storage vector, since the sign structure is not obtained immediately. The code on figure 6.8 shows how we extracted indices of the creation/annihilation operators.

To obtain so-called phase or sign of the determinant we use a very simple and straightforward algorithm, which can be easily understood from the code on figure 6.9.

```

boost::dynamic_bitset<> SpawnedOnThisDeterminant =
    StorageCumulativeDeterminants.at(ExcitorIndex);
boost::dynamic_bitset<> particles =
    SpawnedOnThisDeterminant&~SD_0;
int firstpartind = particles.find_first();
int secondpartind = particles.find_next(firstpartind);

boost::dynamic_bitset<> holes =
    SD_0&~SpawnedOnThisDeterminant;
int firstholeind = holes.find_first();
int secondholeind = holes.find_next(firstholeind);

```

**Figure 6.8:** Finding excitations in bitstring representation. SD\_0 is the bitstring representation of the reference.

```

boost::dynamic_bitset<> SpawnedOnThisDeterminant =
    StorageCumulativeDeterminants.at(ExcitorIndex);
boost::dynamic_bitset<>
    OperatorPair(qsys->getStatesStochastic());
int i = 0;
while (i != firstpartind-1){
    i++;
}
for (int ind = firstholeind+1; ind <= i; ind++){
    OperatorPair.set(ind);
}
int power = (OperatorPair&SpawnedOnThisDeterminant).count();
return std::pow(-1, power);

```

**Figure 6.9:** Finding the phase in bitstring representation. SD\_0 is the bitstring representation of the reference.

## 6.18 Possible optimization

Based on our observations the *better* way to write the CCQMC solver is to split different parts of the algorithm into the class structure described in this work. First of all the calculation of  $P_{gen}$  probabilities may differ for various systems, so the code for this task must be a part of the basis class. The purpose of the main solver class should be the implementation of the main imaginary-time loop, possibly, with a role of the MPI load balancer. Vectors with populations and determinants are better to keep as a separate class, which makes it more flexible to change between storage data types and search algorithms. It can definitely be needed during the debugging phase in the development of the code and the benchmarking of the results.

The advantages of the parallel computing is natural to use for Monte Carlo methods. Number of samples can be increased by the factor of number of CPU cores without considerable increase of the execution time. However parallelization of the algorithm should be implemented with care. The first problem might be increased covariance caused by the quality of random number generator. The second is a load balancing scheme, which we propose below. Master process performs initial sampling and creates an object with populations and determinants. Then each MPI-child process receives a copy of this object. Sampling of all cluster sizes is performed independently by each child process and sent back to the master without any additional data processing. After all the objects with populations and determinants are gathered, master process performs annihilation step by merging all objects into collective storage at this particular imaginary-time step. This require a significant amount of CPU time and can be speed up by OpenMP and a smart algorithm for sorting/searching elements in arrays. Also, it should be noted that during annihilation step child processes are in a "sleep" state. It happens because of the fact, they need an updated list with excips and their total number to start new sampling.

This study has found that development of the code for the CCQMC algorithm requires continuous monitoring of different parameters during simulations. This usually includes a number of spawning/death events, sign of spawned excips, total number of positive/negative death occurrences, event distribution over cluster size etc. We suggest that the output of the debugging information should be separated into a standalone class, and developers should pay particular attention to it. In our simulations we used a simple approach of appending a datafile in the end of every  $\tau$ -loop. Python environment was then employed to produce real-time plots. Plotting directly from C++ may be the best choice here due to speed of disk write operations.



# Chapter 7

## Results and Discussion

### 7.1 CCD benchmark

The brute force solvers written in Python and C++ are slow. They were used to compare contributions of different diagrams or to test permutations under development of the block solver. However for larger basis sizes and growing number of particles it was not possible to compare results with brute-force solvers.

The CCD approximation for the electron gas has been extensively studied by many researchers. An obvious way to validate our results is to compare the CCD correlation energy with energies of other codes. We compared our energies to the results of Hansen[13], Baardsen [4] and recent studies of Miller[23]. Our code produced results which corroborate the findings of a great deal of the previous work, as can be seen in table 7.1.

However comparison of the results with the precision beyond 6 digits might be useful only for specific debug purposes. We used  $10^{-6}$  tolerance for the iterative process. Besides, precision of different constants <sup>1</sup> used to set up units correctly is usually determined to about the 5-8th decimal. Additionally taking into account round-off errors and minor differences in the implementation, we find the agreement in results up to the  $10^{-6}$  to be adequate. It is apparent from this table that the results are consistent with above mentioned numerical precision.

---

<sup>1</sup>We refer here to the Planck constant, the speed of light, the fine-structure constant etc. which are used in `setUpUnits` function for the electron gas.

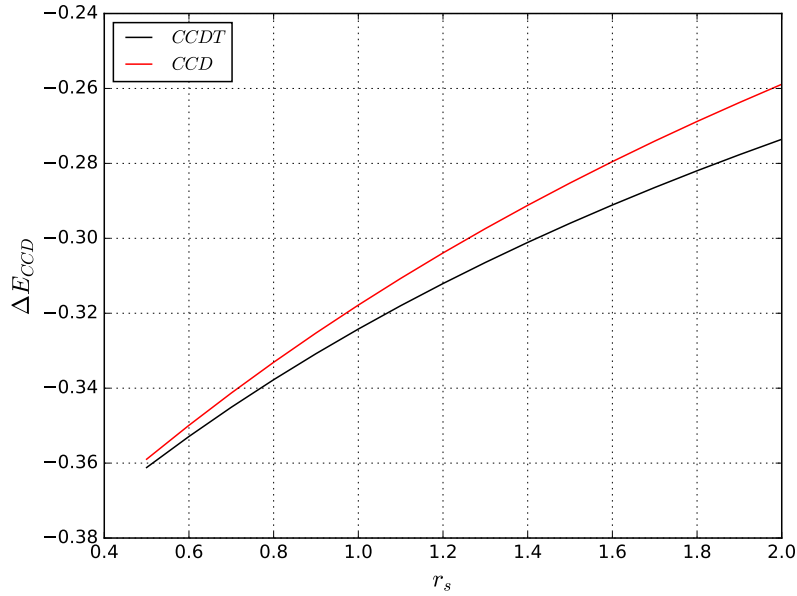
**Table 7.1:** CCD results comparison between our block CCD solver ( $\Delta E_{CCD}^M$ ), Miller's results, Hansen's CCD results ( $\Delta E_{CCD}^H$ ) for the HEG. Results of Baardsen ( $\Delta E_{CCD}^B$ ). Results are presented for 14 electrons. Mixing parameter  $\alpha = 0.8$  for Miller's results, and  $\alpha = 0.3$  for Hansen's and our results. All energies are presented in Hartree units.

$r_s$	States	$\Delta E_{CCD}$ [23]	$\Delta E_{CCD}$ [13]	$\Delta E_{CCD}$ [4]	$\Delta E_{CCD}$
1.0	54	-0.3178228436889338	-0.317822843688933		-0.3178230699319593
1.0	66	-0.3926965898061968	-0.3926965898061966		-0.3926968074770886
1.0	114	-0.4479105961757175	-0.4479105961757175	-0.447909	-0.4479109389185165
1.0	162	-0.4805572589306421	-0.4805572589306416		-0.4805570782443642
1.0	186	-0.4855229317521320	-0.4855229317521318	-0.485523	-0.4855227418241649
1.0	246	-0.4929245740023971	-0.4929245740023975		-0.4929243692209991
1.0	294	-0.4984909094066806	-0.4984909094066818		-0.4984906939593084
1.0	342	-0.5019526761547777	-0.5019526761547779		-0.5019524529049425
1.0	358	-0.5025196736076414	-0.502519673607641	-0.502523	-0.5025194488388953
0.5	114	-0.5120153541478306	-0.5120153541478306	-0.512015	-0.5120152296730573
0.5	186	-0.5114620616957333		-0.553329	-0.553329520936615
0.5	342	-0.5729645498903680	-0.572964549890367		-0.572964399507112
0.5	358	-0.5297417436322546		-0.573678	-0.5736804143578936
2.0	114	-0.3577968843144996	-0.3577968843144996	-0.357798	-0.3577955282575226
2.0	342	-0.4014136184665558	-0.4014136184665555		-0.4014117905655014

As a simple unit test we compare our results for a small number of electrons with the result of the CCD solver from [16]. The table 7.2 illustrates agreement in the correlation energy up to 15 significant digits of precision.

**Table 7.2:** CCD iteration comparison with solver from [16] for the HEG. Here,  $N_e = 2$ ,  $N_s = 66$ ,  $r_s = 1.0$ , and there is no mixing.

n	$\Delta E_{CCD}$	$\Delta E_{CCD}$ [16]
MBPT2	-0.02080623248720259	-0.0208062324872026
0	-0.01833394188507821	-0.0183339418850782
1	-0.01864313296809337	-0.01864313296809333
2	-0.01860449158525421	-0.01860449158525414
3	-0.01860932269795203	-0.01860932269795206
4	-0.01860871870720909	-0.01860871870720907



**Figure 7.1:** The correlation energy for 14 particles and 54 single-particle states for different densities of the electron gas. CCDT results were taken from [13].

## 7.2 Thermodynamic and complete basis set limit estimates

In the thermodynamic limit (TDL) the density of the electron gas remains constant while the particle number tends to infinity ( $N \rightarrow \infty$ ). However many-body methods scale relatively expensively with  $N$ . Calculations for large  $N$  were performed by several researchers and are available in a few codes with extensive parallelism. But even despite this only finite systems can be studied. Typical strategy is to begin with  $N$  electrons in a simulation-cell, and approach the thermodynamic limit using extrapolation techniques. Instead we can try to treat as many particles as possible with stochastic methods. The volume with a given number of particles can be fixed, then the number of particles can be increased together with the volume, while the density remains the same. Increasing the number of particles will lead to convergence of the energy to the true value.

The second difficulty arises from the fact that the many-body wavefunction is constructed from a finite one-particle basis set. By introducing a limited number of single-particle states ( $M$ ) we limit kinetic energy values for the electron gas. Strictly, both limits should be found: TDL ( $N \rightarrow \infty$ ) and the complete basis set (CBS) limit ( $M \rightarrow \infty$ )

Previous studies [30] of the plane wave basis expansion showed, that the correlation energy converges to the CBS limit as reciprocal number of the basis state functions, independently of chosen cut-off scheme. Shepherd et al. in their work introduced an exceptional extrapolation scheme <sup>2</sup>, which improves convergence for high-correlation regimes. However, since we study densities corresponding to the low correlations in the electron gas, we use a simple polynomial extrapolation. Our extracted value for the correlation energy is -0.5142, while FCIQMC study report [29] value of -0.5325(4). Spenser and Thom[33] found this value to be -0.5155(3) with CCQMC CCD. All presented values are obtained for the 3D electron gas with  $r_s = 1.0$ . Initiator approximation was used for the CCQMC algorithm. Recent paper by Neufeld and Thom[24] reported result for 11 150 basis functions without initiator approximation -0.51450(9), which is the closest result to the extrapolation of deterministic CCD. The difference in CCD CCQMC results might be explained by the bias introduced by the initiator approximation.

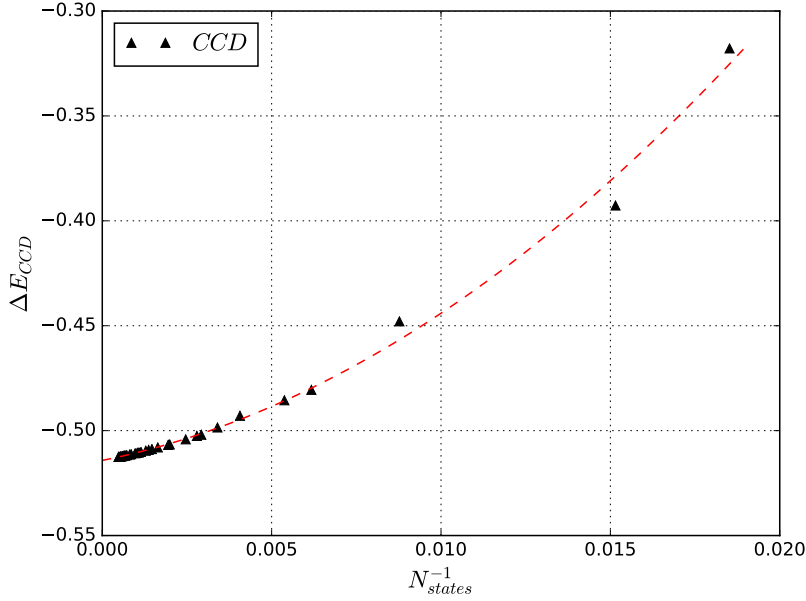
## 7.3 CCQMC population dynamics

While the electron gas is a common test system for deterministic many-body algorithms, its role as a proving ground for the CCQMC method should be considered with caution.

---

<sup>2</sup>It is called *single-point* extrapolation





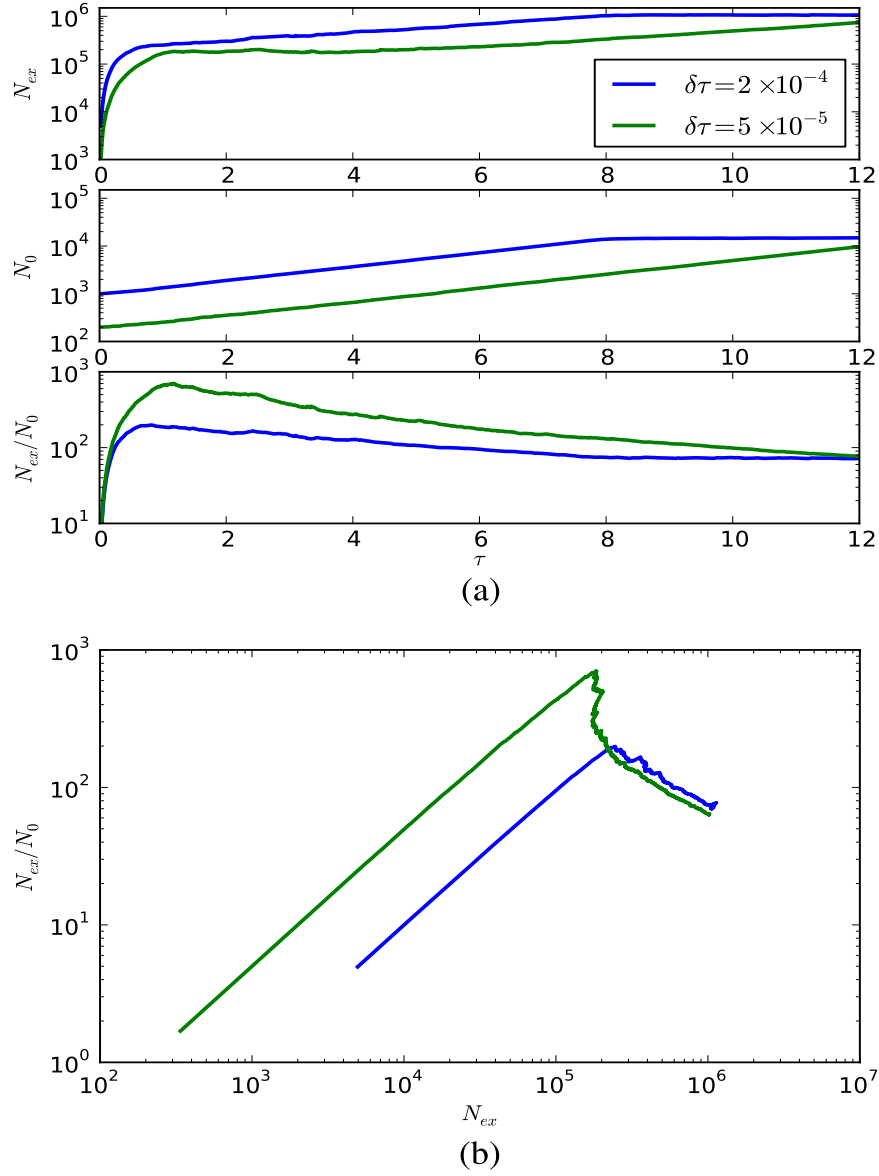
**Figure 7.2:** The CBS limit for the 3D electron gas is extrapolated by a polynomial of degree two in  $N_{states}^{-1}$ .  $N_{states}$  denotes maximum number of single-particle states included in basis. Calculations were performed for 14 electrons. We used Miller's[23] results for higher number of states([406 - 2090]). Extracted limit for  $N_{states} \rightarrow \infty$  was found to be -0.514204.

The absence of singles, which is an indisputable advantage for deterministic methods, presents some difficulties for the stochastic coupled cluster. We try to examine these difficulties by analyzing sampling of all cluster sizes.

$$\langle D_n | H - S | D_m \rangle \quad (7.1)$$

Foremost, we sample cluster of size zero, which corresponds the case when  $|D_m\rangle$  is equal to the reference in (7.1). We denote the level of excitation with red lines for the double and blue dashed line for the single excitation in (7.1). For the electron gas we can choose only double excitation and spawn an excip on the double-excited determinant. There is no death of excips at cluster of size zero before the tuning of parameter  $S$ . That is, before annihilation plateau this sampling step produces only spawning events. And since we choose cluster size according to the exponential distribution, the reference is sampled  $N_{ex}/2$  times, where  $N_{ex}$  is the total population of excips.

Furthermore clusters of size one are sampled, where we can choose as  $|D_m\rangle$  double- triple-excited determinant etc. Again, the bra-part should be single or



**Figure 7.3:** Ne cc-pVQZ CCSDTQ calculations starting with different initial particle numbers at the reference and different timesteps. (a): With a carefully chosen low timestep and initial population, a plateau is visible. An increased timestep and initial population overshoot the plateau but have a shoulder. The lower panel shows a maximum of the particle ratio at the position of the shoulder and plateau. (b): "Shoulder plots" allow shoulder height to be read off easily and calculations compared. Reproduced from [33], with the permission of AIP Publishing.

double excitation of  $|D_m\rangle$ . Given that we refuse to choose singles, there is no spawning events for this cluster size. Only death events can occur in this case. The number of samples is  $N_{ex}/4$ , which is two times less than the one for the reference. Hence possible outcome for death events is lower than for the spawn events from the cluster of size one.

Lastly, we consider the clusters of size two. If we now turn to an atomic or a molecular system, we can construct a cluster of size two by picking a single excitation and then adding another single or double to it. Cluster, constructed from two singles, provides a death opportunity within CCSD truncation. Moreover, it also can initiate a spawning event (blue dashed line in (7.1)). The electron gas differs from systems with single excitations in the way that described mechanism of spawning/death is impossible. Within CCSD approximation the only possible form of these composite clusters is double + double. This results into quadruply-excited determinant in the ket part and double-excited determinant in the bra part. Quadruples participate in death process can be ignored within CCSD. Summing up the last arguments it is obvious that clusters of size two are source of spawning events only.

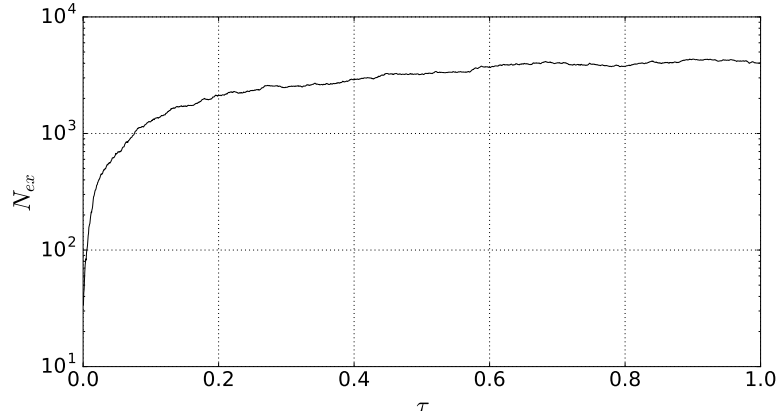
For the successful CCQMC calculations critical population of excips is needed. Search for this population is commonly referred to finding of the annihilation plateau. Its height may differ for various systems, basis sets and level of correlations. Below we present our findings about possible techniques helping in examining the critical population of excips and development of the code.

It has been demonstrated that careful selection of simulation parameters might help to find annihilation plateau by eye [33]. Spencer and Thom demonstrated that plateau is easy to see (figure 7.3(a)) for small values of  $\tau$ . However, these results were based upon data for the neon atom within CCSDTQ truncation. As it has been already discussed, every cluster size within such truncation results in death/spawn events. To verify the correctness of our implementation we need to obtain similar dynamics evolution in imaginary-time. The most difficult parts of the CCQMC algorithm are the variation of the population on reference  $N_0$  and the certainty of the sign structure. Since excips on reference are particles of another kind, it was decided that the best method to test sign structure was to fix population on reference and investigate population dynamics of the excited space only. We were unable to find in literature similar plots for the CCD truncation of the electron gas. We expected that behavior of the population of the excited space ( $N_{ex}$ ) should follow the pattern observed on figure 7.3(a). Figure 7.4 presents dependence of  $N_{ex}$  on  $\tau$  for 14 electrons and 54 single-particle states. Earlier studies of the electron gas with the CCQMC method reported that for  $r_s = 1$  and CCSDT the critical populations ranges from  $10^4$  to  $10^5$  <sup>3</sup> for basis sizes from 203 to 2109 single particle states. Besides it was outlined that for the smaller values of  $r_s = 0.5$  the sufficient number of excips decreases to the value

---

<sup>3</sup>Estimation was obtained by the visual inspection of figure 8 in [33]

of roughly  $10^3$ <sup>4</sup>. The current state of knowledge fails to establish correlations between the critical number of excips and the number of the basis state functions together with the level of correlation inside studied system. As it can be seen on figure 7.1, the difference in the correlation energies for CCD and CCDT truncations decreases on decreasing  $r_s$ . Based on this we expected that the plateau height for  $r_s = 0.5$  should be near  $10^3$  excips. Our estimations are presented on figure 7.4.

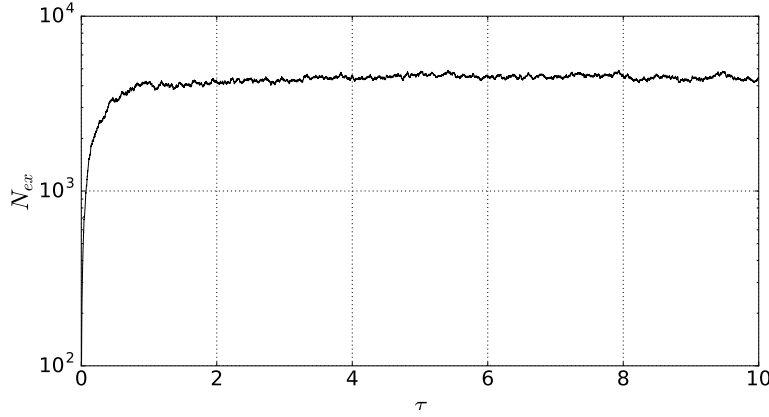


**Figure 7.4:** The population dynamics of the excited space for 14 electrons and 54 basis functions.  $r_s = 0.5$ ,  $\delta\tau = 0.0005$ .

We tested same code for 162 basis state functions and observed same pattern, but with number of excips grown closer to  $10^4$ . These results would seem to suggest that the sign structure of the wave-function is correct. However, these results should be treated cautiously since population on the reference was fixed in the beginning of the simulation. Additionally it can be seen from figure 7.9 that after 1600 iteration (that corresponds to the value of  $\tau = 0.6$  on figure 7.4) number of death and spawn events fluctuates around some value. That can mean that spawn and death probably give equal amount of positive and negative excips. Tuning of the energy shift parameter  $S$  might be considered as a qualitative measure of plateau. We found that attempts to vary  $S$  before plateau resulted into extinction of the population of the excited space. In contrast adjusting of the shift after plateau caused stable population (see figures 7.10 and 7.5). However these results must be interpreted with caution because of the imbalance between the population of the excited space and the population of the reference. The last might be the possible explanation for the absence of the expected after plateau exponential growth of the total population.

---

<sup>4</sup>within the CCSDT truncation for the 3D electron gas



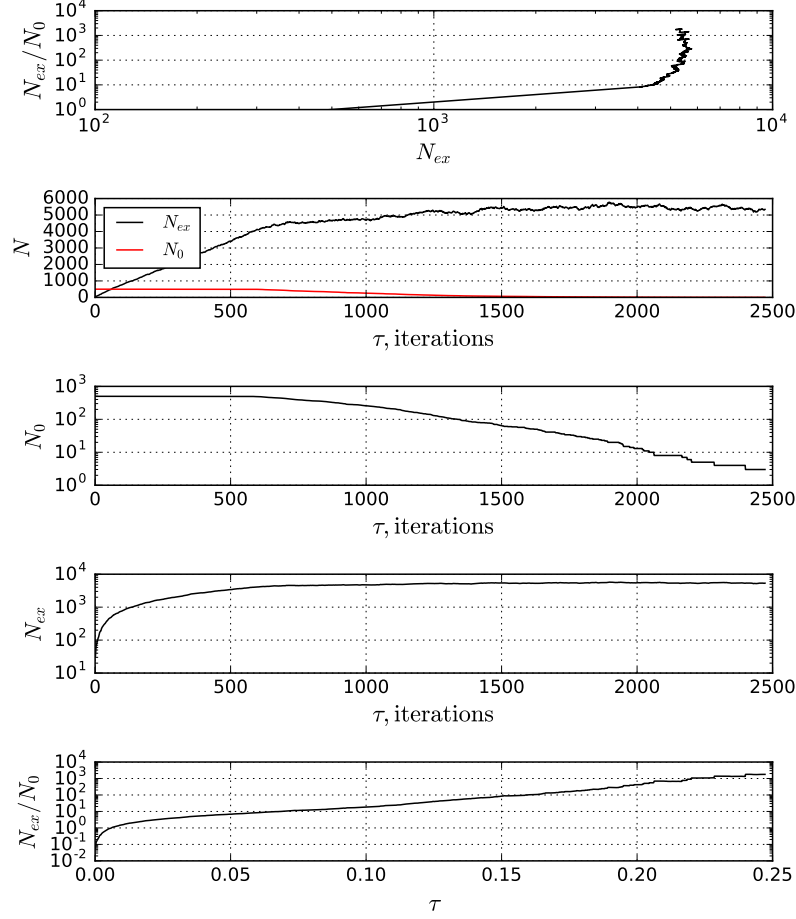
**Figure 7.5:** The population dynamics of the excited space for 14 electrons and 54 basis functions with  $r_s = 0.5$ ,  $\delta\tau = 0.0005$  and the population control enabled after  $5 \cdot 10^3$  iterations. Dampening parameter is  $\gamma = 0.05$ . The energy shift  $S$  is tuned every five iterations.

## 7.4 Reference population

The population of excips on the reference ( $N_0$ ) state plays an important role in the method as it has direct impact on the number of samples on the following iterations. Much to our surprise, we were unable to find any information about its adjustment in the literature. When we checked the sign structure of the population we fixed  $N_0$ . Figure 7.6 provides the result obtained from this study. As can be seen from the plot, the population on reference gradually dies out. this leads the simulation to divergence, as it can be seen from the upper plot on figure 7.6.

We tried several approaches to vary  $N_0$ . As soon as excips from the reference state are quasi-excips we cannot spawn them, however such an opportunity should be considered. The simplest possible *naïve* approach is to take the total number of attempts to choose a particular determinant divided by the total number of all available variants. However, with this scheme, caution must be applied, as the probability to choose the reference is equal to the probability for all other excips. This scheme destabilizes the population and leads to a gradual disappearance (death) of the reference state(see figure 7.7).

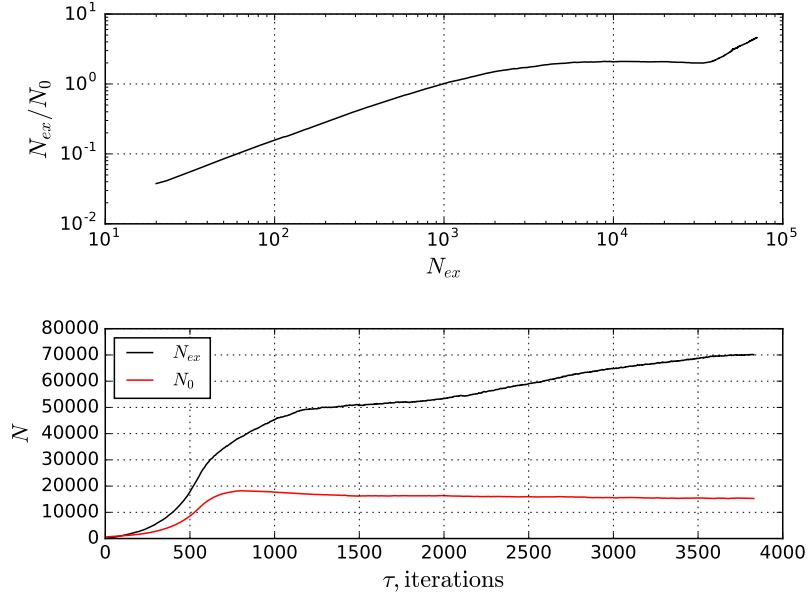
The second way of tuning  $N_0$  was chosen to add to the reference population a small quantity calculated by the following scheme. If we consider spawning events from the clusters of size zero, we can consider this process as a reversible one and add the number of successful spawn events to the reference population at the end of every iteration. The results of the application of this scheme show that population extincts(see lower plot on figure 7.8) which leads to the fail of the simulation(upper plot on figure 7.8).



**Figure 7.6:** The population dynamics for 14 electrons and 54 basis functions with  $r_s = 0.5$ ,  $\delta\tau = 0.0001$  and the population control enabled after  $5 \cdot 10^2$  iterations. Dampening parameter is  $\gamma = 0.05$ . The energy shift  $S$  is tuned every five iterations.

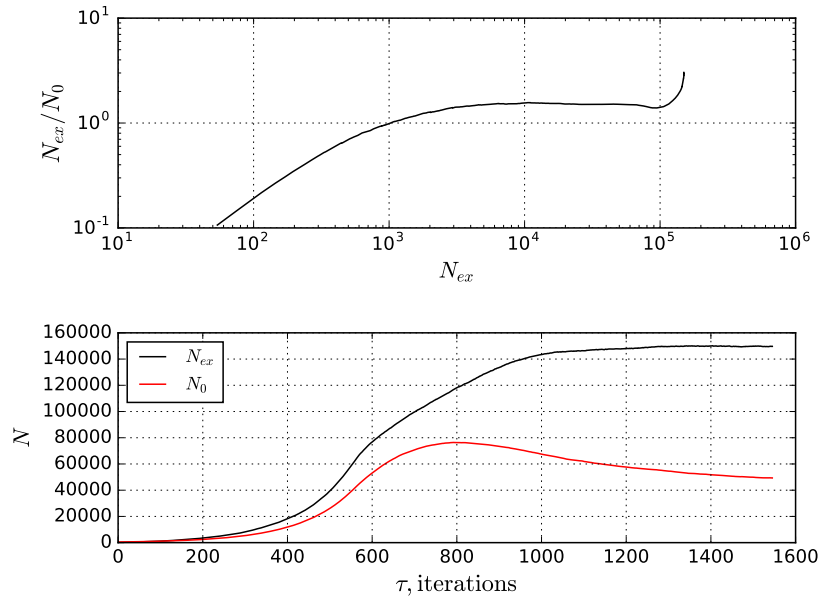
## 7.5 Comments on the energy

There are two possibilities to obtain the energy in the CCQMC method. The first one is to sample the projected energy, and the second is via the evaluation of the energy shift  $S$ . Both methods can be applied only after having stabilized of the population and correct structure of the wavefunction. This can be determined by the pattern similar to the one shown in figure 7.3(b). Since none of our proposed schemes to preserve normalization worked, we were unable to estimate energy values. The unnormalized stable simulation showed values of  $S$  fluctuating



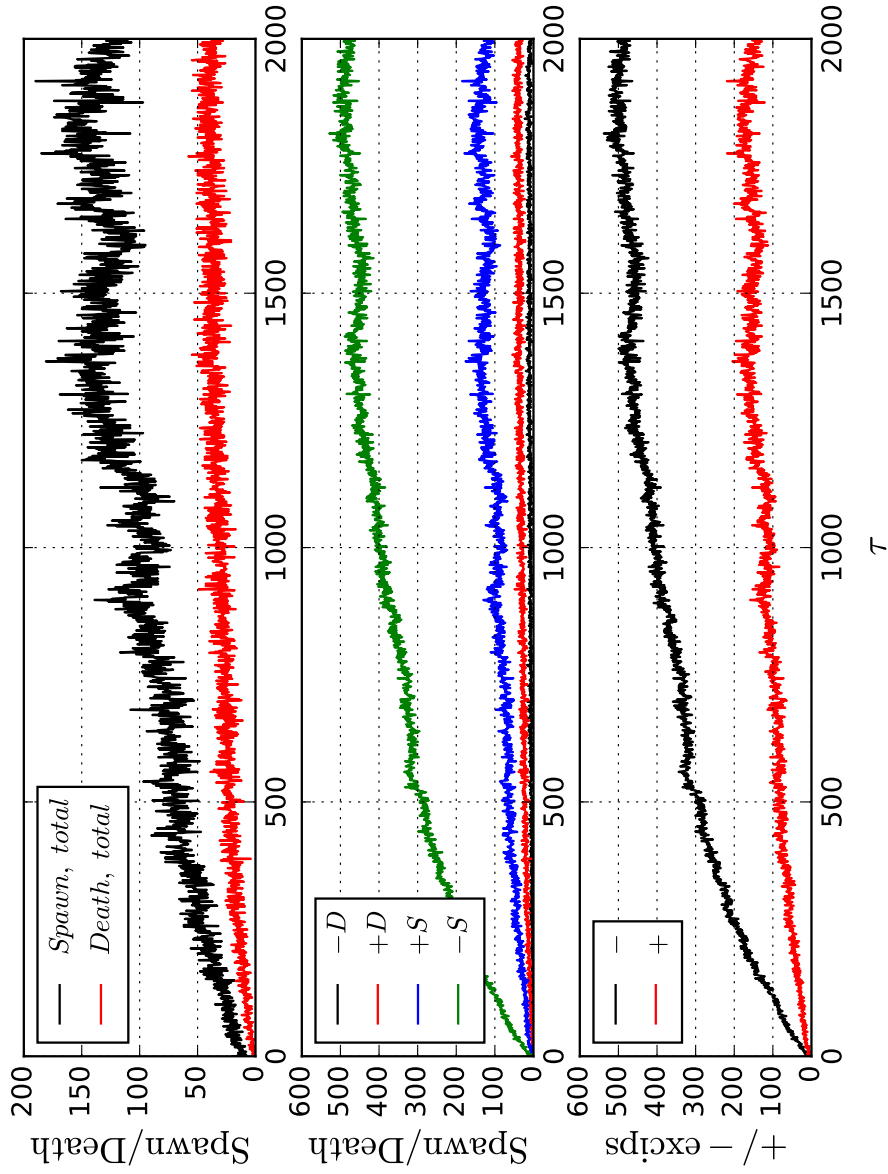
**Figure 7.7:** The population dynamics for 14 electrons and 54 basis functions with  $r_s = 0.5$ ,  $\delta\tau = 0.0001$  and the population control enabled after  $5 \cdot 10^2$  iterations. Dampening parameter is  $\gamma = 0.05$ . The energy shift  $S$  is tuned every five iterations. Naive scheme for tuning  $N_0$  was implemented.

around the value differing from the deterministic result by roughly 5%. However this must to be interpreted cautiously.

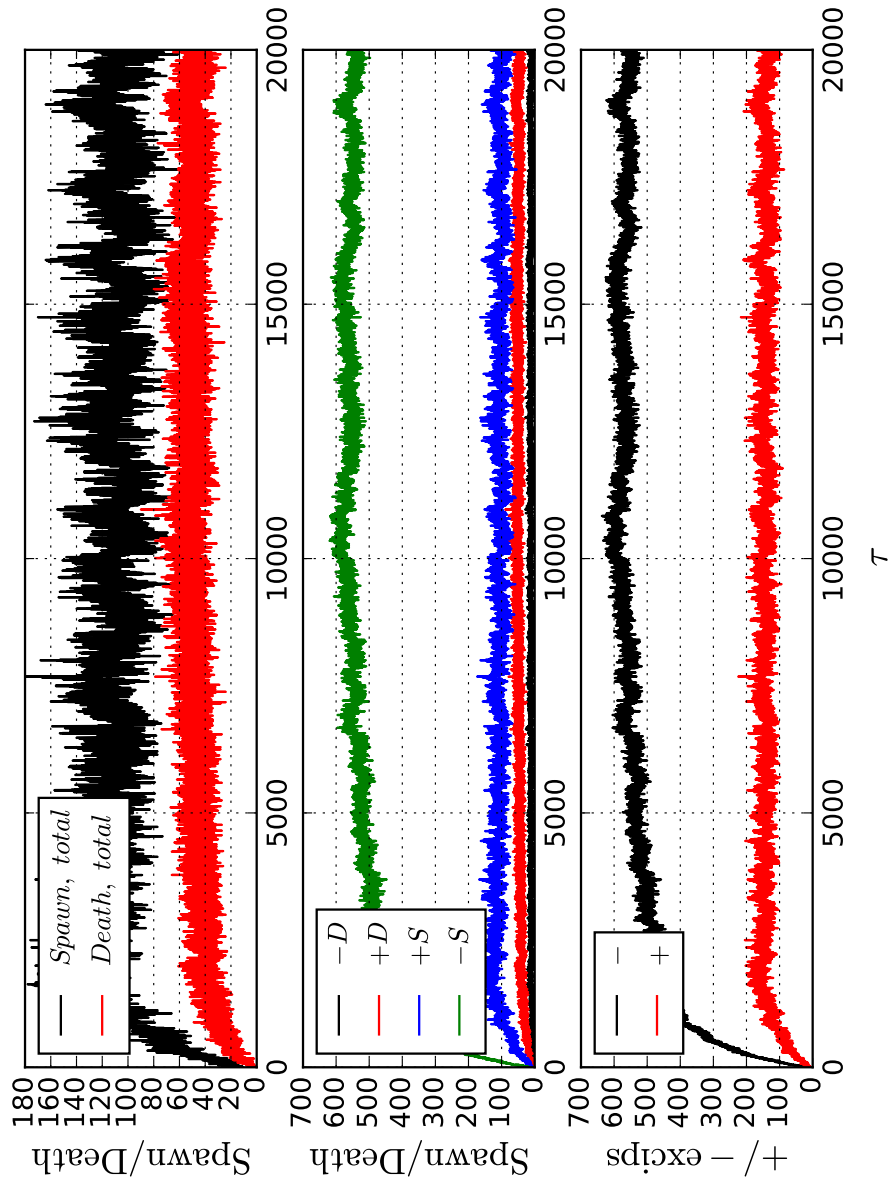


**Figure 7.8:** The population dynamics for 14 electrons and 54 basis functions with  $r_s = 0.5$ ,  $\delta\tau = 0.0001$  and the population control enabled after  $5 \cdot 10^2$  iterations. Dampening parameter is  $\gamma = 0.05$ . The energy shift  $S$  is tuned every five iterations. Inverse scheme for tuning  $N_0$  was implemented.





**Figure 7.9:** The population dynamics of the excited space for 14 electrons and 54 basis functions.  $\tau_s = 0.5$ ,  $\delta\tau = 0.0005$ .



**Figure 7.10:** The population dynamics of the excited space for 14 electrons and 54 basis functions with  $r_s = 0.5$ ,  $\delta\tau = 0.0005$  and the population control enabled after  $5 \cdot 10^3$  iterations. Dampening parameter is  $\gamma = 0.05$ . The energy shift  $S$  is tuned every five iterations.

## Chapter 8

# Conclusions And Future Prospects

The aim of this thesis was to study stochastic formulation of the coupled cluster theory. First we started with a review of the literature constituting the needed taxonomy of various many-body methods. Deterministic coupled cluster theory together with several common Monte Carlo methods for many-body problem were presented. Much effort was put into understanding and presenting the stochastic coupled cluster method in a simple way. The computer program using the deterministic CCD theory and CCQMC method was written in C++ with an object oriented approach. Our CCD solver reproduced published results for the chosen test system: the homogeneous electron gas. However, our codes are written in a very flexible way and can easily be extended to other correlated many-body systems.

It was found that the deterministic implementation provides a significant benchmark information for the further development of the code for the stochastic solver, especially, for the treatment of the sign problem. Our findings suggest that in general transitionally invariant systems are not an optimal choice as a test system for the CCQMC method, at least when only a specific set of excited states is studied (the so-called doubles excitations).

We explored the population dynamics and found that the population of the reference state plays one of the major roles for the CCQMC algorithm. The main focus of our study was on developing a code and exploring the stochastic method. Our observations on the choice of data structures, algorithms and optimization techniques are extensively covered. To our knowledge there exists only one implementation of this method besides ours. This work contributes by providing a strong foundation for further studies of the CCQMC algorithm and development of a program that can easily be extended to other systems. Finally, we discuss several areas for further research.

## 8.1 Future outlook

The CCQMC method is a relatively new. Even despite the fact that the main idea was presented in 2010, published research on this topic began to arrive only in 2017. Results published till the date of writing this thesis agree with other methods. There is no strict theory about convergence, and there are only numerical evidences that a successfully implemented algorithm works for particular studied cases for single-reference systems. The sign problem exists for CCQMC method, however it is much more easily treated compared to the standard diffusion and Green's function Monte Carlo (DMC) approach, where the so-called fixed node approximation is used. There is a limited theoretical framework for the CCQMC sign problem, and only simplified models are available. Easy parallelization provided by stochastic methods and simplicity of implementation in comparison to the deterministic coupled cluster method allows us to study relatively big systems including a greater numbers of basis state functions for the higher excitation levels. Further research might investigate different sampling schemes[27], which were recently introduced. The analysis of the behavior of the statistical errors for these schemes and optimization techniques such as initiator approximation might be a very important topic for future works. The optimization of the implementation both numerically and algorithmically is also an important question. The method seems to be a very promising one, even though there is a lack of extensive studies. Our plan is to extend and generalize the code to apply it to the study of quantum dots, which has not been done yet.

# Appendices



## A.1 Second quantization

Creation operator is defined as follows:

$$c_p^\dagger |\phi_q \dots \phi_s\rangle = |\phi_p \phi_q \dots \phi_s\rangle. \quad (1)$$

If orbital  $\phi_q$  already exists, the action of this operator results in zero:

$$c_p^\dagger |\phi_p \phi_q \dots \phi_s\rangle = 0. \quad (2)$$

Annihilation operator is defined as hermitian adjoint to the creation operator:

$$c_p |\phi_p \phi_q \dots \phi_s\rangle = |\phi_q \dots \phi_s\rangle, \quad (3)$$

where we have removed the first column and row from the Slater determinant. The action of the annihilation operator  $c_p$  on a determinant without  $\phi_p$  orbital will result into zero:

$$c_p |\phi_p \phi_q \dots \phi_s\rangle = 0. \quad (4)$$

The zero particle space is defined as a one-dimensional space spanned by the special state, called the *vacuum state*  $|-\rangle$ . It's worse mentioning that the vacuum state is not zero. Thus a string of the creation/annihilation operators, acting on the vacuum can create a Slater determinant:

$$c_1^\dagger \dots c_N^\dagger |-\rangle = |\phi_1 \dots \phi_N\rangle. \quad (5)$$

Pairwise permutations of the operators change the sign of the resulting determinant. It is easy to see on the example with two orbitals:

$$c_q^\dagger c_p^\dagger |-\rangle = |\phi_q \phi_p\rangle = -|\phi_p \phi_q\rangle = -c_p^\dagger c_q^\dagger |-\rangle. \quad (6)$$

The following properties can be derived for the relation of general creation/annihilation operators:

$$\begin{aligned} \{c_p, c_q\} &= 0, \\ \{c_p^\dagger, c_q^\dagger\} &= 0, \\ \{c_p, c_q^\dagger\} &= \delta_{p,q}. \end{aligned} \quad (7)$$

## B.2 Particle-hole formalism

It is more convenient to work with  $n$ -electron reference(vacuum) determinant than with true vacuum  $|-\rangle$ . Reference vacuum is often called *Fermi vacuum*.

We divide all single-particle functions into two subsets. The first subset corresponds to the occupied single-particle states and denoted by indices  $i, j, k \dots$ . The second contains all unoccupied (also called *virtual*) single-particle states.

We reserve indices  $a, b, c \dots$  for these states. Single-particle states which are occupied in the reference determinant( $i, j, k \dots$ ) are called *hole* states. Similarly, unoccupied states( $a, b, c \dots$ ) are referred to as *particle* states. Annihilation operator  $\hat{c}_i$  destroys particle located on the occupied orbital  $i$ , or with other words, creates a hole state. So  $\hat{c}_i$  can be considered as a quasi-particle(hole) creation operator. Similar trail of thoughts can be introduced for the  $\hat{c}^\dagger$  operator. Thus, we define quasi-particle annihilation operators as those that annihilate holes and particles:  $c_i^\dagger$  and  $c_a$ . Similarly, quasi-particle creation operators are introduced:  $c_i$  and  $c_a^\dagger$ . Anticommutator relations are changed for these quasi-particle operators, and the only non-zero ones are:

$$\begin{aligned} \{c_i^\dagger c_j\} &= \delta_{ij}, \\ \{c_a c_b^\dagger\} &= \delta_{ab}. \end{aligned} \tag{8}$$

### C.3 Normal ordering

If one considers a string of second-quantization operators(for example, creation/annihilation), this string is referred to as *normal – ordered* if all annihilation operators stand to the right of all creation operators. It is usually denoted by  $\{\}_N$ . Subscript  $N$  is often omitted.

Normal ordered string = (creation operators) · (annihilation operators).

More strictly we can write:

$$\{A_1 A_2 \dots A_n\} = (-1)^\sigma \text{normal ordered string}, \tag{9}$$

where  $\sigma$  is a number of permutations needed to rearrange the operators such, that their product is written on the normal ordered form. This form of writing of the operator strings is extremely useful, since the action of such string on the vacuum state is always zero by the definition of the annihilation operator. The latter is true both for the true vacuum and for the Fermi vacuum.

### D.4 Wick's theorem

To formulate Wick's theorem some definitions are needed.

Contraction between two arbitrary creation and annihilation operators is denoted by:

$$\overline{XY},$$

and represents the difference between current given order and a normal order:

$$\overline{XY} = XY - \{XY\}. \tag{10}$$



There is only one non-zero contraction between a pair of the creation and the annihilation operators:

$$\overline{c_p c_q^\dagger} = c_p c_q^\dagger - \{c_p c_q^\dagger\} = \delta_{pq}. \quad (11)$$

Contraction inside the operator string.

Let  $A_1 A_2 \dots A_n$  be an product of arbitrary second-quantized operators. We choose  $A_q$  and  $A_p$  to be a pair of the operators, and denote number of possible permutation which places  $A_q$  to the first place and  $A_p$  to the second place as  $\sigma$ .

$$\{A_1 \dots \overline{A_q \dots A_p} \dots A_n\} = (-1)^{|\sigma|} \{\overline{A_q A_p} A_{\sigma(3)} \dots A_{\sigma(n)}\}. \quad (12)$$

where  $\sigma$  is any number of permutations, satisfying  $\sigma(1) = q$  and  $\sigma(2) = p$ .

### Wick's theorem

Any operator string of creation and annihilation operators can be written as its normal ordered form plus a sum of all possible contractions inside normal ordered form.

$$\begin{aligned} A_1 A_2 \dots A_n = & \{A_1 A_2 \dots A_n\} + \sum_{\text{all single contractins}} \underbrace{\{A_1 A_2 \dots A_n\}}_{\text{all single-contracted terms}} + \\ & \sum_{\text{all double contractins}} \underbrace{\{A_1 A_2 \dots A_n\}}_{\text{all double-contracted terms}} + \dots + \sum_{\text{all } \frac{n}{2} \text{ contractins}} \underbrace{\{A_1 A_2 \dots A_n\}}_{\text{all } \frac{n}{2}\text{-contracted terms}}. \end{aligned} \quad (13)$$

If  $n$  is odd, all terms in the last sum of equation (13) have one uncontracted operator.

### Generalized Wick's theorem

Generalization of Wick's theorem extends it application to the case of multiple products of normal ordered substrings. Let us consider substring product:

$$\Lambda = \{A_1 A_2 \dots A_n\} \{B_1 B_2 \dots B_n\} \dots \{Z_1 Z_2 \dots Z_n\}.$$

Then  $\Lambda$  can be written as a sum over all fully-contracted sets of normal-ordered substrings:

$$\begin{aligned}
 \Lambda = & \{A_1 A_2 \dots A_n \dot{B}_1 B_2 \dots B_n \dot{\dots} \dot{Z}_1 Z_2 \dots Z_n\} + \\
 & \sum_{\text{all single contractins}} \underbrace{\{A_1 A_2 \dots A_n \dot{B}_1 B_2 \dots B_n \dot{\dots} \dot{Z}_1 Z_2 \dots Z_n\}}_{\text{all single-contracted substrings}} + \dots + \\
 & \sum_{\text{all } \frac{n}{2} \text{ contractins}} \underbrace{\{A_1 A_2 \dots A_n \dot{B}_1 B_2 \dots B_n \dot{\dots} \dot{Z}_1 Z_2 \dots Z_n\}}_{\text{all } \frac{n}{2} \text{-contracted substrings}}.
 \end{aligned} \tag{14}$$

# Bibliography

- [1] James B. Anderson. “A Random-walk Simulation of the Schrödinger Equation:  $H^+_3$ ”. en. In: *The Journal of Chemical Physics* 63.4 (Aug. 1975), pp. 1499–1503. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.431514](https://doi.org/10.1063/1.431514).
- [2] Alfin Aspuru-Guzik and William A Lester. “Quantum Monte Carlo Methods for the Solution of the Schrödinger Equation for Molecular Systems”. en. In: *Quantum Monte Carlo methods* (), p. 51.
- [3] Brian M. Austin, Dmitry Yu. Zubarev, and William A. Lester. “Quantum Monte Carlo and Related Approaches”. en. In: *Chemical Reviews* 112.1 (Jan. 2012), pp. 263–288. ISSN: 0009-2665, 1520-6890. DOI: [10.1021/cr2001564](https://doi.org/10.1021/cr2001564).
- [4] Gustav Baardsen. “Coupled-Cluster Theory for Infinite Matter”. PHD Thesis. University of Oslo, 2014.
- [5] George H. Booth and Ali Alavi. “Approaching Chemical Accuracy Using Full Configuration-Interaction Quantum Monte Carlo: A Study of Ionization Potentials”. en. In: *The Journal of Chemical Physics* 132.17 (May 2010), p. 174104. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.3407895](https://doi.org/10.1063/1.3407895).
- [6] George H. Booth, Alex J. W. Thom, and Ali Alavi. “Fermion Monte Carlo without Fixed Nodes: A Game of Life, Death, and Annihilation in Slater Determinant Space”. en. In: *The Journal of Chemical Physics* 131.5 (2009), p. 054106. ISSN: 00219606. DOI: [10.1063/1.3193710](https://doi.org/10.1063/1.3193710).
- [7] T. Daniel Crawford and Henry F. Schaefer. “An Introduction to Coupled Cluster Theory for Computational Chemists”. en. In: *Reviews in Computational Chemistry*. Ed. by Kenny B. Lipkowitz and Donald B. Boyd. Hoboken, NJ, USA: John Wiley & Sons, Inc., Jan. 2007, pp. 33–136. ISBN: 978-0-470-12591-5 978-0-471-35495-6. DOI: [10.1002/9780470125915.ch2](https://doi.org/10.1002/9780470125915.ch2).
- [8] Francesco A. Evangelista. “Alternative Single-Reference Coupled Cluster Approaches for Multireference Problems: The Simpler, the Better”. en. In: *The Journal of Chemical Physics* 134.22 (June 2011), p. 224102. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.3598471](https://doi.org/10.1063/1.3598471).
- [9] P. P Ewald. “Die Berechnung Optischer Und Elektrostatischer Gitterpotentiale”. English. In: *ANDP Annalen der Physik* 369.3 (1921). OCLC: 5155374561, pp. 253–287. ISSN: 0003-3804.

- [10] Louisa M. Fraser et al. “Finite-Size Effects and Coulomb Interactions in Quantum Monte Carlo Calculations for Homogeneous Systems with Periodic Boundary Conditions”. In: *Physical Review B* 53.4 (Jan. 1996), pp. 1814–1832. DOI: [10.1103/PhysRevB.53.1814](https://doi.org/10.1103/PhysRevB.53.1814).
- [11] Robert A. Van De Geijn and Enrique S. Quintana-ortí. *The Science of Programming Matrix Computations*. 2008.
- [12] E. K. U. Gross, E. Runge, and O. Heinonen. *Many-Particle Theory*. eng. Bristol ; Philadelphia: A. Hilger, 1991. ISBN: 978-0-7503-0072-8 978-0-7503-0155-8.
- [13] Audun Skau Hansen. “Coupled Cluster Studies of Infinite Systems”. Master Thesis. University of Oslo.
- [14] Frank E. Harris, Hendrik Jan Monkhorst, and David L. Freeman. *Algebraic and Diagrammatic Methods in Many-Fermion Theory*. en. Oxford University Press, Jan. 1992.
- [15] Morten Hjorth-Jensen. *Computational Physics. Lecture Notes Fall 2015*. 2015.
- [16] Morten Hjorth-Jensen, Maria Paola Lombardo, and Ubirajara Van Kolck, eds. *An Advanced Course in Computational Nuclear Physics: Bridging the Scales from Quarks to Neutron Stars*. eng. Lecture notes in physics volume 936. OCLC: 988901009. Cham: Springer, 2017. ISBN: 978-3-319-53336-0 978-3-319-53335-3.
- [17] <http://physicsdatabase.com/2014/05/03/top-10-most-cited-physics-papers>.
- [18] <https://github.com/andrei-fys/HEGMBPT2MC>.
- [19] Jindrich Kolorenc and Lubos Mitas. “Applications of Quantum Monte Carlo Methods in Condensed Systems”. en. In: *Reports on Progress in Physics* 74.2 (Feb. 2011), p. 026502. ISSN: 0034-4885, 1361-6633. DOI: [10.1088/0034-4885/74/2/026502](https://doi.org/10.1088/0034-4885/74/2/026502). arXiv: [1010.4992](https://arxiv.org/abs/1010.4992).
- [20] Ioan Kosztin, Byron Faber, and Klaus Schulten. “Introduction to the Diffusion Monte Carlo Method”. en. In: *American Journal of Physics* 64.5 (May 1996), pp. 633–644. ISSN: 0002-9505, 1943-2909. DOI: [10.1119/1.18168](https://doi.org/10.1119/1.18168).
- [21] S. Kvaal. *Lecture Notes for Fys-Kjm4480/9480; Quantum Mechanics For Many-Particle Systems*. 2015.
- [22] David C. Lay, Steven R. Lay, and Judi J. McDonald. *Linear Algebra and Its Applications*. English. 5 edition. Pearson, Jan. 2015. ISBN: 978-0-321-98238-4.
- [23] Sean Bruce Sangolt Miller. “Quantum Mechanical Studies of Infinite Matter by the Use of Coupled-Cluster Calculations, with an Emphasis on Nuclear Matter”. Master Thesis. University of Oslo, 2017.

- [24] Verena A. Neufeld and Alex J. W. Thom. “A Study of the Dense Uniform Electron Gas with High Orders of Coupled Cluster”. en. In: *The Journal of Chemical Physics* 147.19 (Nov. 2017), p. 194105. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.5003794](https://doi.org/10.1063/1.5003794).
- [25] Sergio Pissanetzky. *Sparse Matrix Technology*. en. Academic Press, June 2014. ISBN: 978-1-4832-7040-1.
- [26] Stanley Raimès. *Many-Electron Theory*. English. Amsterdam: North-Holland Pub. Co, 1972. ISBN: 978-0-444-10353-6.
- [27] Charles J. C. Scott and Alex J. W. Thom. “Stochastic Coupled Cluster Theory: Efficient Sampling of the Coupled Cluster Expansion”. en. In: *The Journal of Chemical Physics* 147.12 (Sept. 2017), p. 124105. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.4991795](https://doi.org/10.1063/1.4991795).
- [28] Isaiah Shavitt and Rodney J. Bartlett. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge molecular science. OCLC: ocn297147115. Cambridge ; New York: Cambridge University Press, 2009. ISBN: 978-0-521-81832-2.
- [29] James J. Shepherd, George H. Booth, and Ali Alavi. “Investigation of the Full Configuration Interaction Quantum Monte Carlo Method Using Homogeneous Electron Gas Models”. en. In: *The Journal of Chemical Physics* 136.24 (June 2012), p. 244101. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.4720076](https://doi.org/10.1063/1.4720076).
- [30] James J. Shepherd et al. “Convergence of Many-Body Wave-Function Expansions Using a Plane-Wave Basis: From Homogeneous Electron Gas to Solid State Systems”. en. In: *Physical Review B* 86.3 (July 2012). ISSN: 1098-0121, 1550-235X. DOI: [10.1103/PhysRevB.86.035111](https://doi.org/10.1103/PhysRevB.86.035111).
- [31] David S. Sholl and Janice A. Steckel. *Density Functional Theory*. en. Hoboken, NJ, USA: John Wiley & Sons, Inc., Mar. 2009. ISBN: 978-0-470-44771-0 978-0-470-37317-0. DOI: [10.1002/9780470447710](https://doi.org/10.1002/9780470447710).
- [32] J. S. Spencer, N. S. Blunt, and W. M.C. Foulkes. “The Sign Problem and Population Dynamics in the Full Configuration Interaction Quantum Monte Carlo Method”. en. In: *The Journal of Chemical Physics* 136.5 (Feb. 2012), p. 054110. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.3681396](https://doi.org/10.1063/1.3681396).
- [33] James S. Spencer and Alex J. W. Thom. “Developments in Stochastic Coupled Cluster Theory: The Initiator Approximation and Application to the Uniform Electron Gas”. en. In: *The Journal of Chemical Physics* 144.8 (Feb. 2016), p. 084108. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.4942173](https://doi.org/10.1063/1.4942173).

- [34] John F. Stanton and Rodney J. Bartlett. “The Equation of Motion Coupled-cluster Method. A Systematic Biorthogonal Approach to Molecular Excitation Energies, Transition Probabilities, and Excited State Properties”. en. In: *The Journal of Chemical Physics* 98.9 (May 1993), pp. 7029–7039. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.464746](https://doi.org/10.1063/1.464746).
- [35] Bjarne Stroustrup. *The C++ Programming Language*. Fourth edition. Upper Saddle River, NJ: Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.
- [36] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. en. Courier Corporation, July 1996. ISBN: 978-0-486-69186-2.
- [37] Péter G. Szalay, Marcel Nooijen, and Rodney J. Bartlett. “Alternative Ansätze in Single Reference Coupled-cluster Theory. III. A Critical Analysis of Different Methods”. en. In: *The Journal of Chemical Physics* 103.1 (July 1995), pp. 281–298. ISSN: 0021-9606, 1089-7690. DOI: [10.1063/1.469641](https://doi.org/10.1063/1.469641).
- [38] Alex J. W. Thom. “Stochastic Coupled Cluster Theory”. en. In: *Physical Review Letters* 105.26 (Dec. 2010). ISSN: 0031-9007, 1079-7114. DOI: [10.1103/PhysRevLett.105.263004](https://doi.org/10.1103/PhysRevLett.105.263004).
- [39] Julien Toulouse, Roland Assaraf, and Cyrus J. Umrigar. “Introduction to the Variational and Diffusion Monte Carlo Methods”. en. In: *Advances in Quantum Chemistry*. Vol. 73. Elsevier, 2016, pp. 285–314. ISBN: 978-0-12-803060-8. DOI: [10.1016/bs.aiq.2015.07.003](https://doi.org/10.1016/bs.aiq.2015.07.003).
- [40] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry - 3rd Edition*. English. North Holland: Elsevier B.V., 2007. ISBN: 978-0-444-52965-7.