More Accurate QMC Calculations by Using Machine Learning

by

Even Marius Nordhagen

THESIS

for the degree of

MASTER OF SCIENCE

Faculty of Mathematics and Natural Sciences
University of Oslo

February 3, 2019

Abstract

Abstract should be written here

Acknowledgements

First of all I would like to thank Christine for cooking delicious food and feeding me when I need it the most. She and her food have maintained my motivation in the best way, and they are the main reason why I've survived those two years of masters.

Secondly, my annoying cohabitants have forced me to spend more time at university. I'm so glad you behave the way you do, if you were more cozy I would certainly spend more time at home and this thesis would not be the same.

Acknowledgements should be written here

-One of my mottos used to be that everything has a reason, which I used to point out whenever people were talking about things that apparently could not be described immediately. It was first when I learned about quantum mechanics that I understood I was wrong, and that is one of the reasons why quantum mechanics catched me so hardly.

Contents

1 Introduction **9**
- 1.1 Many-Body problem . . . 10
- 1.2 Machine learning . . . 10
- 1.3 Goals and milestones . . . 10

2 Quantum Many-Body Physics **11**
- 2.1 Introductory Quantum Physics . . . 11
  - 2.1.1 The Schrödinger Equation . . . 11
  - 2.1.2 The Variational Principle . . . 12
  - 2.1.3 Assumptions raised . . . 12
- 2.2 The Trial Wave Function . . . 12
  - 2.2.1 Bosons and fermions . . . 12
  - 2.2.2 Slater determinant . . . 13
  - 2.2.3 Electron structure calculations . . . 14
  - 2.2.4 Cusp condition . . . 14
  - 2.2.5 Basis set . . . 14
  - 2.2.6 Jastrow factor . . . 14

3 Systems and basis sets **15**
- 3.1 Quantum dots . . . 15
- 3.2 Atomic systems . . . 15
- 3.3 Molecular systems . . . 16
- 3.4 Electron gas . . . 16
- 3.5 Helium gas . . . 16

4 Quantum Monte-Carlo Methods **17**
  - 4.0.1 Isotropic processes . . . 17
  - 4.0.2 Anisotropic processes . . . 17
  - 4.0.3 Variational Monte Carlo . . . 17

5 Ab Initio Quantum Methods **21**
- 5.1 Hartree-Fock . . . 21
- 5.2 Configuration Interaction . . . 21
- 5.3 Coupled Cluster . . . 21

6 Machine Learning **23**
- 6.1 Supervised Learning . . . 23
- 6.2 Unsupervised Learning . . . 23
  - 6.2.1 Statistical foundation . . . 23
  - 6.2.2 Boltzmann Machines . . . 23

6.2.3 Restricted Boltzmann Machines 23

7 Optimization **25**
7.1 Optimization of Trial Wave Function 25
7.1.1 Energy Minimization 25
7.1.2 Variance Minimization 25
7.2 Minimization Algorithms 25
7.2.1 Gradient Descent 25
7.2.2 Conjugate Gradient 25

8 Scientific Programming **27**
8.1 Object Orientated Programming 27
8.1.1 Inheritance 28
8.1.2 Virtual Functions 28

9 Implementation **29**
9.1 Foundation 29
9.2 Structure 29
9.3 Optimization algorithms 30
9.3.1 Stochastic Gradient Descent 30
9.3.2 ADAM 30

10 Results **31**
10.1 Comparison of Wave Functions 31
10.2 Ground state energies 31

11 Conclusion and future work **33**

A Dirac notation **35**

B Scaling **37**
B.0.1 Harmonic Oscillator - Natural units 37
B.0.2 Atomic systems - Atomic units 37

C Machine Learning Wave Function **39**

D Wave Function Elements **41**
D.1 Kinetic Energy Calculations 41
D.2 Parameter Update 42
D.3 Derivatives 42
D.3.1 Simple Gaussian 42
D.3.2 Hydrogen-Like Orbitals 43
D.3.3 Padé-Jastrow Factor 43
D.3.4 Slater Determinant 43
D.3.5 NQS-Gaussian 43
D.3.6 NQS-Jastrow Factor 43

List of abbreviations

Letters		**Meaning**
RBM	-	Restricted Boltzmann Machine
MC	-	Monte Carlo
VMC	-	Variational Monte Carlo
DMC	-	Diffusion Monte Carlo
ML	-	Machine Learning
WF	-	Wave Function
SPF	-	Single Particle Function

Table 1: List of symbols used with explaination.

Source Code

The source code is given in `https://github.com/evenmn`

Chapter 1

Introduction

Properties and behavior of quantum many-body systems are determined by the laws of quantum physics which have been known since the 1930s. The time-dependent Schrödinger equation describes the bounding energy of atoms and molecules, as well as the interaction between particles in a gas. In addition, it has been used to determine the energy of artificial structures like quantum dots, nanowires and ultracold condensates. [Electronic Structure Quantum Monte Carlo Michal Bajdich, Lubos Mitas]

Even though we know the laws of quantum mechanics, many challenges are encountered when calculating real-world problems. First, interesting systems often involve large number of particles, which causes expensive calculations. Second, we do not have a good model for the three-body interaction, which is vital when it comes to strong correlations. Paul Dirac recognized those problems already in 1929,

”*The general theory of quantum mechanics is now almost complete... ...The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.*”

-Paul Dirac, Quantum Mechanics of Many-Electron Systems, 1929

”At the same time, advent of computer technology has offered us a new window of opportunity for studies of quantum (and many other) problems. It spawned a “third way” of doing science which is based on simulations, in contrast to analytical approaches and experiments. In a broad sense, by simulations we mean computational models of reality based on fundamental physical laws. Such models have value when they enable to make predictions or to provide new information which is otherwise impossible or too costly to obtain otherwise. In this respect, QMC methods represent an illustration and an example of what is the potential of such methodologies.” [Electronic Structure Quantum Monte Carlo Michal Bajdich, Lubos Mitas]

History of QMC before and after the invention of electronic computers. Enrico Fermi 1930s similarities between imaginary time Schrödinger equation and stochastic processes in statistical mechanics. Metropolis VMC early 1950s. Kalos Greens's function Monte Carlo late 1950s. Ceperly and Alder 1980 homogeneous electron gas.

Multi scale calculations are... A field of interest is how a systems behave when the interaction gets weaker. One way to model this, is to have a harmonic oscillator with a decreasing system frequency.

With machine learning, a function can be fitted to everything as long as we can define a cost function to minimize. The purpose of this thesis is to construct optimal wave functions for different systems by using machine learning.

- Low frequency (weakly interacting electrons) field of interest - Multi scale calculations - Cartesian - Introduce the wavefunction - Mention the uncertainity principle and also quantum entanglement to catch the readers interest

1.1 Many-Body problem

Not possible to solve analytically

1.2 Machine learning

Branch of artificial intelligence

1.3 Goals and milestones

- Investigate a new method to solve the Many-Body problem

Chapter 2

Quantum Many-Body Physics

Here I might present basic quantum mechanics briefly.

2.1 Introductory Quantum Physics

Even though it is called elementary, quantum mechanics is not basic at all.

2.1.1 The Schrödinger Equation

The Schrödinger equation is a natural starting point, which gives the energy eigenstates of a system defined by a Hamiltonian $\hat{\mathrm{H}}$ and its eigenfunctions, Ψ, which are the wave functions. The time-independent Schrödinger equation reads

$$\hat{\mathcal{H}}\Psi_n(\boldsymbol{r}) = \epsilon_n\Psi_n(\boldsymbol{r}) \tag{2.1}$$

where the electronic Hamiltonian takes the form

$$\hat{\mathrm{H}} = -\sum_i^N \frac{\hbar^2}{2m}\nabla_i^2 + \sum_i^N u_i + \sum_i^N \sum_{j>i}^M k\frac{e^2}{r_{ij}} \tag{2.2}$$

with u_i as an arbitrary external potential and the last term as the Coulomb potential between electrons. r_{ij} is the relative distance between electron i and j, defined by $r_{ij} \equiv |\boldsymbol{r}_i - \boldsymbol{r}_j|$.

Setting up equation (2.1) with respect to the energies, we obtain some integrals,

$$\epsilon_n = \frac{\int d\boldsymbol{r}\Psi_n^*(\boldsymbol{r})\hat{\mathrm{H}}\Psi_n(\boldsymbol{r})}{\int d\boldsymbol{r}\Psi_n^*(\boldsymbol{r})\Psi_n(\boldsymbol{r})}, \tag{2.3}$$

which not necessarily are trivial to solve. If we take the wave function squared we get the probability of finding a particle at a certain position,

$$P(\boldsymbol{r}) = \Psi_n^*(\boldsymbol{r})\Psi_n(\boldsymbol{r}) = |\Psi_n(\boldsymbol{r})|^2 \tag{2.4}$$

so the nominator is simply the integral over all probabilities. If the wave function is normalized correctly, this should always give 1. Assuming that is the case, the expectation value can be expressed more elegantly by using Dirac notation,

$$E[\Psi] = \langle\Psi|\hat{\mathrm{H}}|\Psi\rangle, \tag{2.5}$$

where the first part, $\langle\Psi|$ is called a bra and the last part, $|\Psi\rangle$ is called a ket. At first this might look artificial and less informative, but it simplifies the notation significantly. More information about the notation is found in Appendix B.

We often do not know the exact wave function, and need to guess a trial wave function. Henceforth, we will use Ψ as the exact total wave function, ψ as the exact single particle function (SPF), Φ as the total trial wave function and ϕ as the trial SPF. [Gri05]

2.1.2 The Variational Principle

In the equations above, the presented wave functions are assumed to be the exact eigen functions of the Hamiltonian. But often we do not know the exact wave functions, and we need to guess what the wave functions might be. In those cases we make use of the variational principle, which states that only the groundstate wave function is able to give the groundstate energy. All other wave functions with the required properties (see section 2.2) give higher energies, and mathematically we can express the statement with

$$\epsilon_0 \leq \langle\Psi_T|\hat{\mathrm{H}}|\Psi_T\rangle \,. \tag{2.6}$$

We have here introduced a trial wave function Ψ_T which not necessary is the true ground state wave function. This means that we can adjust our trail wave function with respect to the energy, and the lower energy we get the better is our wave function.

2.1.3 Assumptions raised

- Non-relativistic
- Point-like particles
- Born-Oppenheimer

2.2 The Trial Wave Function

By the first postulate of quantum mechanics, the wave function contains all the information specifying the state of the system. This means that all observable in classical mechanics can also be measured from the wave function, resulting finding the wave function as our main goal.

The true wave function is in most cases unknown, and we need an initial wave function based on educated guesses, called a trial wave function. This function has to meet some requirements in order to be used in the variational principle, including the CUSP condition, normalizable and ... In addition, the wave function should be symmetric or anti-symmetric under exchange of two coordinates.

2.2.1 Bosons and fermions

An introductary sentence is needed to increase the FLOW. Assume we have a permutation operator $\hat{P}$ which exchanges two coordinates in the wave function,

$$\hat{P}(i\to j)\Psi_n(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i,\ldots,\boldsymbol{x}_j,\ldots,\boldsymbol{x}_M) = p\Psi_n(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_j,\ldots,\boldsymbol{x}_i,\ldots,\boldsymbol{x}_M), \tag{2.7}$$

where p is just a factor which comes from the transformation. If we again apply the $\hat{P}$ operator, we should switch the same coordinates back, and we expect to end up with the initial wave function. For that reason, $p = \pm 1$. [1]

The particles that have an antisymmetric (AS) wavefunction under exchange of two coordinates are called fermions, named after Enrico Fermi, and have half integer spin. On the other hand, the particles that have a symmetric (S) wavefunction under exchange of two coordinates are called bosons, named after Satyendra Nath Bose, and have integer spin.

It turns out that because of their antisymmetric wavefunction, two identical fermions cannot be found at the same position at the same time, known as the Pauli principle. This causes some difficulties when dealing with multiple fermions, because we always need to ensure that the total wavefunction becomes zero if two identical particles happen to be at the same position. To do this, we introduce a Slater determinant as described in the next chapter. In this particular project, we are going to focus on electrons and therefore fermions. Anyway, much of the theory applies for bosons as well.

Read https://manybodyphysics.github.io/FYS4480/doc/pub/secondquant/html/secondquant-bs.html

2.2.2 Slater determinant

For a system of more particles we can define a total wavefunction, which is a composition of all the single particle wavefunctions (SPF) and contains all the information about the system. For fermions we need to compile the SPFs such that the Pauli principle is fulfilled at all times. One way to do this is by setting up the SPFs in a determinant, known as a Slater determinant.

Consider a system of two identical fermions with SPFs ϕ_1 and ϕ_2 at positions $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ respectively. The way we define the wavefunction of the system is then

$$\Psi_T(\boldsymbol{r}_1, \boldsymbol{r}_2) = \begin{vmatrix} \phi_1(\boldsymbol{r}_1) & \phi_2(\boldsymbol{r}_1) \\ \phi_1(\boldsymbol{r}_2) & \phi_2(\boldsymbol{r}_2) \end{vmatrix} = \phi_1(\boldsymbol{r}_1)\phi_2(\boldsymbol{r}_2) - \phi_2(\boldsymbol{r}_1)\phi_1(\boldsymbol{r}_2), \tag{2.8}$$

which is set to zero if the particles are at the same position. This is called a Slater determinant, and yields the same no matter how big the system is.

Notice that we denote the wave function with the $'T'$, which indicates that it is a trial wave function. We do this because the spin part is avoided with ψ as the radial parts only, thus this wavefunction is not the true wavefunction. We will look closer at how we can factorize out the spin part later. The spin part is assumed to not affect the energies.

A general Slater determinant for a system of N particles takes the form

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_N) = \begin{vmatrix} \psi_1(\boldsymbol{r}_1) & \psi_2(\boldsymbol{r}_1) & \dots & \psi_N(\boldsymbol{r}_1) \\ \psi_1(\boldsymbol{r}_2) & \psi_2(\boldsymbol{r}_2) & \dots & \psi_N(\boldsymbol{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\boldsymbol{r}_N) & \psi_2(\boldsymbol{r}_N) & \dots & \psi_N(\boldsymbol{r}_N) \end{vmatrix} \tag{2.9}$$

where the ψ's are the true single particle wave functions, which are the tensor products

$$\psi = \phi \otimes \xi \tag{2.10}$$

with ξ as the spin part.

---

[1] This was true until 1976, when J.M. Leinaas and J. Myrheim discovered the anyon, https://www.uio.no/studier/emner/matnat/fys/FYS4130/v14/documents/kompendium.pdf.

There is a similar "Slater permanent" which applies for bosons. It is similar to a determinant, but all negative signs are replaced by positive signs.

2.2.3 Electron structure calculations

For our purpose we will study fermions with spin $\sigma = \pm 1/2$ only, which can be seen as electrons. In this particular case, the SPFs can be arranged in spin-up and spin-down parts, such that the Slater determinant can be simplied to

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) = \begin{vmatrix} \phi_1(\boldsymbol{r}_1)\xi_\uparrow & \phi_1(\boldsymbol{r}_1)\xi_\downarrow & \ldots & \phi_{N/2}(\boldsymbol{r}_1)\xi_\downarrow \\ \phi_1(\boldsymbol{r}_2)\xi_\uparrow & \phi_1(\boldsymbol{r}_2)\xi_\downarrow & \ldots & \phi_{N/2}(\boldsymbol{r}_2)\xi_\downarrow \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{r}_N)\xi_\uparrow & \phi_1(\boldsymbol{r}_N)\xi_\downarrow & \ldots & \phi_{N/2}(\boldsymbol{r}_N)\xi_\downarrow \end{vmatrix}. \tag{2.11}$$

This is called the wavefunction ansatz, because assumptions are raised, like two particles with opposite spins are found to be at the same position all the time, i.e, an equal number of fermions have spin up as spin down, are applied. Further the...

SHOULD END UP WITH SPLITTED DETERMINANTS HERE

For a detailed walkthrough, see appendix I in REF(The Stochastic Gradient Approximation: an application to Li nanoclusters, Daniel Nissenbaum).

2.2.4 Cusp condition

The Coulomb potential gives a cusp when two charged particles approach each other. This means that the wave function should also have a cusp, known as the Coulomb cusp condition, Kato theorem or just the cusp condition.

2.2.5 Basis set

To go further, we need to define a basis set, $\phi_n(\boldsymbol{r})$ which should be chosen carefully based on the system.

2.2.6 Jastrow factor

The Jastrow factor is introduced in order to capture interaction properties, ...

We will use the Padé-Jastrow factor, which can be expressed as

$$J(\boldsymbol{r}; \beta, \gamma) = \exp\left(\sum_{i=1}^{N}\sum_{j=1}^{i}\frac{\beta(i,j)r_{ij}}{1+\gamma r_{ij}}\right) \tag{2.12}$$

which gets small when r_{ij} gets small. If we recall that the probability distribution is given by the wave function squared, we see that the Padé-Jastrow factor gives a lower probability of finding particles close to each other, which is what we want.

Chapter 3

Systems and basis sets

Derive the Hamiltonian given in equation (2.2), both the kinetic term and the interaction term (Coulomb). Crash course in electrostatics

We often know the exact wave functions of the systems when interaction is dropped, but when we include interaction the wave functions ae unknown.

Specific systems need specific basis sets

3.1 Quantum dots

Quantum dots are very small particles, and contain fermions or bosons hold together by an external potential. Since these particles have discrete electronic states like an atom, they are often called artificial atoms.

In this thesis we will study electrons trapped in harmonic oscillators, which gives an external potential affecting particle i is given by

$$u_i = \frac{1}{2}\omega^2 r_i^2. \tag{3.1}$$

Thus, the Hamiltonian reads

$$\hat{\mathrm{H}} = \sum_{i=1}^{P}(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}\omega^2 r_i^2) + \sum_{i<j}\frac{1}{r_{ij}} \tag{3.2}$$

and we will use the Hermite functions as our basis set. The one dimensional Hermite functions read

$$f_n(x) = H_n(x)\exp\bigl(-x^2/2\bigr) \tag{3.3}$$

and are known to be the exact SPFs in a harmonic oscillator.

3.2 Atomic systems

We will also investigate real atoms, which have Hamiltonians defined by the Born-Oppenheimer approximation.

The External potential affecting particle i is therefore

$$u_i = -\frac{1}{2}\frac{Z}{r_i} \tag{3.4}$$

$$\hat{\mathrm{H}} = \sum_{i=1}^{P}(-\frac{1}{2}\nabla_i^2 - \frac{1}{2}\frac{Z}{r_i}) + \sum_{i<j}\frac{1}{r_{ij}}, \tag{3.5}$$

which applies for an atom with stationary nucleus at origin. The first part is related to the kinetic energy, while the second is

"Colloquially, we call such solutions and derived properties as electronic structure."

3.3 Molecular systems

3.4 Electron gas

3.5 Helium gas

He^3 and He^4

Chapter 4

Quantum Monte-Carlo Methods

Some great methods are developed over the past decade. We will focus on the Hartree-Fock and variational Monte Carlo methods and give a detailed explanation of those methods. Additionally, the well-known methods configuration interaction and coupled cluster will be described briefly for some kind of completeness.

Monte Carlo methods in quantum mechanics are a bunch of methods that are built on diffusion processes, and includes Variational Monte Carlo (VMC), Diffusion Monte Carlo (DMC) and others. The common denominator is that we move particles in order to find the optimal configuration, usually where the energy is minimized. The particles can be moved isotropic, i.e., uniformly in all directions, or they can be affected by a drift force which makes the process anisotropic.

4.0.1 Isotropic processes

In isotropic processes, we have random walks where the particles move randomly in space which falls under the category Markov chains. If we assume constant timestep, the step index can be considered the time, thus we have a time-dependent probability density $P(x,t)$. This probability density needs to satisfy the isotropic diffusion equation,

$$\frac{\partial P(\boldsymbol{x},t)}{\partial t} = D\frac{\partial^2}{\partial \boldsymbol{x}^2}P(\boldsymbol{x},t). \tag{4.1}$$

4.0.2 Anisotropic processes

For anisotropic processes, we have a drift and the moves are no longer considered random but falls still under the category Markov chains. Because of the drift, we need to rewrite the diffusion equation and we end up with the Fokker-Planck equation,

$$\frac{\partial P(\boldsymbol{x},t)}{\partial t} = D\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial}{\partial \boldsymbol{x}} - F\right)P(\boldsymbol{x},t) \tag{4.2}$$

which needs to be satisfied. The new positions in coordinate space are given as solution of the Langevin equation

$$\frac{\partial \boldsymbol{x}(t)}{\partial t} = D\boldsymbol{F}(\boldsymbol{x}(t)) + \eta \tag{4.3}$$

4.0.3 Variational Monte Carlo

The Variational Monte Carlo (hereafter, VMC) method is today widely used when it comes to the study of ground state properties of quantum mechanical systems. It is a Monte Carlo method which makes use of Metropolis sampling, and has been used in studies of fermionic systems since the 1970's. [Deb14] If we go back to the variational principle in equation (2.6), we see that by choosing a wave function which satisfies the criteria, we will get an energy larger or equal to the ground state energy.

There are two main problems we need to solve

1. We seldomly know the correct wave function
2. The integral we need to find the energy is hard or impossible to solve

Let us take the last problem first. The solution is to approximate the integral with a sum,

$$\begin{aligned} E &\leq \frac{\int \Psi_T(\boldsymbol{r})^* \hat{H} \Psi_T(\boldsymbol{r}) d\boldsymbol{r}}{\int \Psi_T(\boldsymbol{r})^* \Psi_T(\boldsymbol{r}) d\boldsymbol{r}} \\ &= \int P(\boldsymbol{r}) E_L(\boldsymbol{r}) d\boldsymbol{r} \\ &\approx \frac{1}{N} \sum_{i=1}^{N} E_L(\boldsymbol{r}_i) \end{aligned} \tag{4.4}$$

where the local energy is defined as

$$E_L(\boldsymbol{r}) \equiv \frac{1}{\Psi_T(\boldsymbol{r})} \hat{H} \Psi_T(\boldsymbol{r}) \tag{4.5}$$

and the $\boldsymbol{r}_i$ is withdrawn from the probability distribution $P(\boldsymbol{r})$, which is given by

$$P(\boldsymbol{r}) = \frac{|\Psi_T(\boldsymbol{r})|^2}{\int |\Psi_T(\boldsymbol{r})|^2 d\boldsymbol{r}}. \tag{4.6}$$

The energy found from equation (4.4) is an expectation value, and we therefore know that the true energy lies within the standard error. When increasing the number of Monte-Carlo cycles, the standard error decreases and we get a more accurate energy. In the limit N goes to infinity, the variance should approach zero,

$$\langle E \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} E_L(\boldsymbol{r}_i). \tag{4.7}$$

For more statistical details, see [Deb14].

Given a wave function we are now able to estimate the corresponding energy, but how do we find a good wave function? In VMC, we define a wave function with variational parameters, which are adjusted in order to minimize the energy for every iteration. For every iteration, we run N Monte-Carlo cycles where we withdraw a new position $\boldsymbol{r}_i$. Whether or not the proposed move should be accepted is determined by the Metropolis algorithm.

4.0.3.1 The Metropolis Algorithm

The genius of the metropolis algorithm, is that the acceptance of a move is not based on the probabilities themselves, but the ratio between the new and the old probability
. In that way, we avoid calculating the sum over all probabilities, which often is expensive or even impossible to calculate.

In its simplest form, the move is proposed randomly, and it is accepted if the ratio is larger than a random number between 0 and 1. However, with this approach a lot of moves will be rejected, which wastes CPU time. A better method is **importance sampling**, which makes a educated guess of the best way to move based on diffusion processes, and move the particle in that direction.

A time-dependent probability density needs to satisfy the Fokker-Planck equation

Chapter 5

Ab Initio Quantum Methods

The term *ab initio* means from first principles, implying that only physical constants are put into the methods. The Monte-Carlo methods are not considered *ab initio* as nonphysical hyper parameters are required. Some example methods are Hartree-Fock methods and post-Hartree-Fock methods like Configuration Interaction and Coupled Cluster, which will be discussed in the following.

5.1 Hartree-Fock

Describe this detailed

5.2 Configuration Interaction

The completeness relation

5.3 Coupled Cluster

The coupled cluster method is the *de facto* standard wave function-based method for electronic structure calculations. [Pal05] The method approximates the wave function with an exponential expansion,

$$|\Psi_{\mathrm{CC}}\rangle = e^{\hat{T}} |\Phi_0\rangle \tag{5.1}$$

where $\hat{T}$ is the cluster operator, entirely given by $\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \dots$ with

$$\hat{T}_n = \left(\frac{1}{n!}\right)^2 \sum_{abc\dots} \sum_{ijk\dots} t_{ijk\dots}^{abc\dots} a_a^\dagger a_b^\dagger a_c^\dagger \cdots a_k a_j a_i. \tag{5.2}$$

We again want to solve the Schrödinger equation,

$$\hat{H} |\Psi\rangle = \hat{H} e^{\hat{T}} |\Phi_0\rangle = \epsilon e^{\hat{T}} |\Phi_0\rangle, \tag{5.3}$$

which can be simplified by multiplying with $e^{-\hat{T}}$ from the left. This introduces us to the **similarity transformed Hamiltonian**

$$\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}. \tag{5.4}$$

If we on one hand now multiply with the reference bra on the left hand side, we easily observe that

$$\langle \Phi_0 | \bar{H} | \Phi_0 \rangle = \epsilon \tag{5.5}$$

which is the coupled cluster energy equation. On the other hand, we can multiply with an excited bra on left hand side, and find that

$$\left\langle \Phi_{ijk\ldots}^{abc\ldots} \middle| \bar{H} \middle| \Phi_0 \right\rangle = 0 \tag{5.6}$$

which are the coupled cluster amplitude equations. The similarity transformed Hamiltonian can be rewritten using the Baker-Campbell-Hausdorff expansion

$$\begin{aligned} \bar{H} = \hat{H} &+ [\hat{H}, \hat{T}] \\ &+ \frac{1}{2}[[\hat{H}, \hat{T}], \hat{T}] \\ &+ \frac{1}{6}[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}] \\ &+ \frac{1}{24}[[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}], \hat{T}] \\ &+ \cdots \end{aligned} \tag{5.7}$$

and we are in principle set to solve the amplitude equations with respect to the amplitudes $t_{ijk\ldots}^{abc\ldots}$ and then find the energy. The expansion is able to reproduce the true wave function exactly using a satisfying number of terms and an infinite basis. This is, of course, not possible, but even by limiting us to the first few coupled cluster operators, the results are often good compared to other methods. [DCFSI07]

Chapter 6

Machine Learning

Parameter estimation is a large part of science, because we are often not able to measure things directly. For instance, when Millikan in ... performed his famous electron charge experiment or when we estimate Sun's mass.

Sometimes when people talk about machine learning, it sounds like its a totally new field, but the truth is that many techniques that used to have other names, are now classified as machine learning. As the UiO professor Anne Solberg said during one of her lectures

"In the early 1990's I was working a lot with machine learning, but at that time we called it pattern recognition and regression"

Machine learning includes all methods where we want to fit a model to a data set, and pattern recognition and regression are clearly some of those methods. In both cases we have concrete targets for the training, and the training is therefore called supervised training. In other cases we do not have targets, but want to train our model based on a probability distribution or so, which is called unsupervised learning.

6.1 Supervised Learning

Regression, pattern recognition, Feed-forward neural networks,

6.2 Unsupervised Learning

How can we train using unsupervised learning?

6.2.1 Statistical foundation

Bayesian statistics

6.2.2 Boltzmann Machines

Based on a Hopfield network How the things are derived, including Boltzmann distribution and stuff from statistical mechanics

6.2.3 Restricted Boltzmann Machines

Chapter 7

Optimization

7.1 Optimization of Trial Wave Function

7.1.1 Energy Minimization

7.1.2 Variance Minimization

7.2 Minimization Algorithms

Gradient methods

7.2.1 Gradient Descent

7.2.2 Conjugate Gradient

Chapter 8

Scientific Programming

Since this thesis is much about writing code, it is natural including a few words about it.

The computer's language itself is binary, and is the lowest level. To translate commands to this language, we need a ”translator”, which is a language that fills the gap between the binary language and human commands. This language is categorized in levels based on how similar they are to the binary language. Low-level languages are similar to the binary language, which means fast but complicated. High-level languages are easy to work with, but are not as fast as low-level languages. Might mention grammar etc

One can either do *procedural programming* or *object oriented programming*. The former means that the code is written in the same order as the program flow goes, while one in the latter defines objects.

8.1 Object Orientated Programming

In the everyday life, we are surrounded by objects all the time which we can place in different categories. For instance, a *cat* is an object with a name, race, age and so on, and can be placed in the class *animals*. In object oriented programming, the class could be implemented as

```
class Animal:
  def __init__(self, animal, name, race, age):
    self.animal = animal
    self.name = name
    self.race = race
    self.age = age

  def __call__(self):
    return "%s is a %s that's %d years old and of the race %s"%(self.name, self.animal,
        ↪ self.age, self.race)
```

where...

The next step is to define an object, which in our case is the cat with name ”Schrodinger”:

```
Alma = Animal("cat", "Schroedeger", "Ragdoll", 4)
print(Alma())
```

This implies that ”Schrodinger” is a cat of race ”Ragdoll” and of age 4. When calling this class from ”Schrodinger”, the class returns

```
$python3 simple_class.py
>>> Schrodinger is a cat that's 4 years old and of the race Ragdoll
```

You might wonder how this is related to scientific programming. The answer is that it is often convenient to define various parts of the code as objects to increase the liability and maximize reuse of code. For example, we use various Hamiltonians, where each can be defined as a subclass of the Hamiltonian superclass.

The code above is written in Python, but the exact task could be performed in C++.

INCLUDE C++ IMPLEMENTATION

As one can see,

In C++ one needs to define constructors and destructors. All variables used inside the functions are defines in the constructor, while they are removed in the destructor to free up memory. In Python, the constructors are called `__init__` by default, and memory is handled automatically.

8.1.1 Inheritance

This is also called parent and child, respectively.

Parent and child Polymorphy: Child inherit from the parents. Virtual functions to achieve runtime polymorphism Should define virtual destructor as well

1. Single inheritance
2. Multiple inheritances

Python and C++ support multiple inheritances. Multilevel inheritance: Got child and grand child. Hierarchical inheritance: Parent got several children. https://www.geeksforgeeks.org/inherita
in-python/

8.1.2 Virtual Functions

Often one wants to define a template of objects... where the super class defines which functions its objects should have. In C++, this can be achieved by virtual functions, functions with arguments specified but task undefined. Those functions are overwritten by the corresponding functions in the object (hence virtual),

Chapter 9

Implementation

In many cases, planning is half the job, and so is true for a good VMC implementation. In fact, the program was restructured three times before we landed on this final version. It is based on Morten Ledum's VMC framework [`https://github.com/mortele/variational-monte-carlo-fys4411`], which was meant as an example implementation in the course *FYS4411 - Computational Physics II: Quantum Mechanical Systems*.

During the implementations we had two main areas of focus

- Efficient code
- High legibility, tidy code

Unlike many other VMC codes, our code was developed flexible with respect to the wave functions. This means that one can combine various wave function elements, where each element is implemented separately. For instance, the simple Gaussian function and the Padé-Jastrow factor were implemented separately, but they can easily be combined. The way one does this in practice, is to append multiple wave function elements to the vector `WaveFunctionElements` in main. If one for instance want to combine the simple Gaussian with the Padé-Jastrow factor, this can be done by

```
System* system = new System();
std::vector<class WaveFunction*> WaveFunctionElements;
WaveFunctionElements.reserve(2);
WaveFunctionElements.push_back(new class SimpleGaussian(system, alpha));
WaveFunctionElements.push_back(new class PadeJastrow(system, beta, Gamma));
system->setWaveFunction                    (WaveFunctionElements);
```

The big advantage of this implementation is that we do not need to hard code every possible combination of wave function elements, which reduces the number of code lines significantly. This also eases adding new elements, since we only need to calculate the derivatives of the particular element (do not need to worry about cross terms). Exactly how this is done can be read in Appendix D.

The con is that the program will be slightly slower.

9.1 Foundation

9.2 Structure

Add structure chart

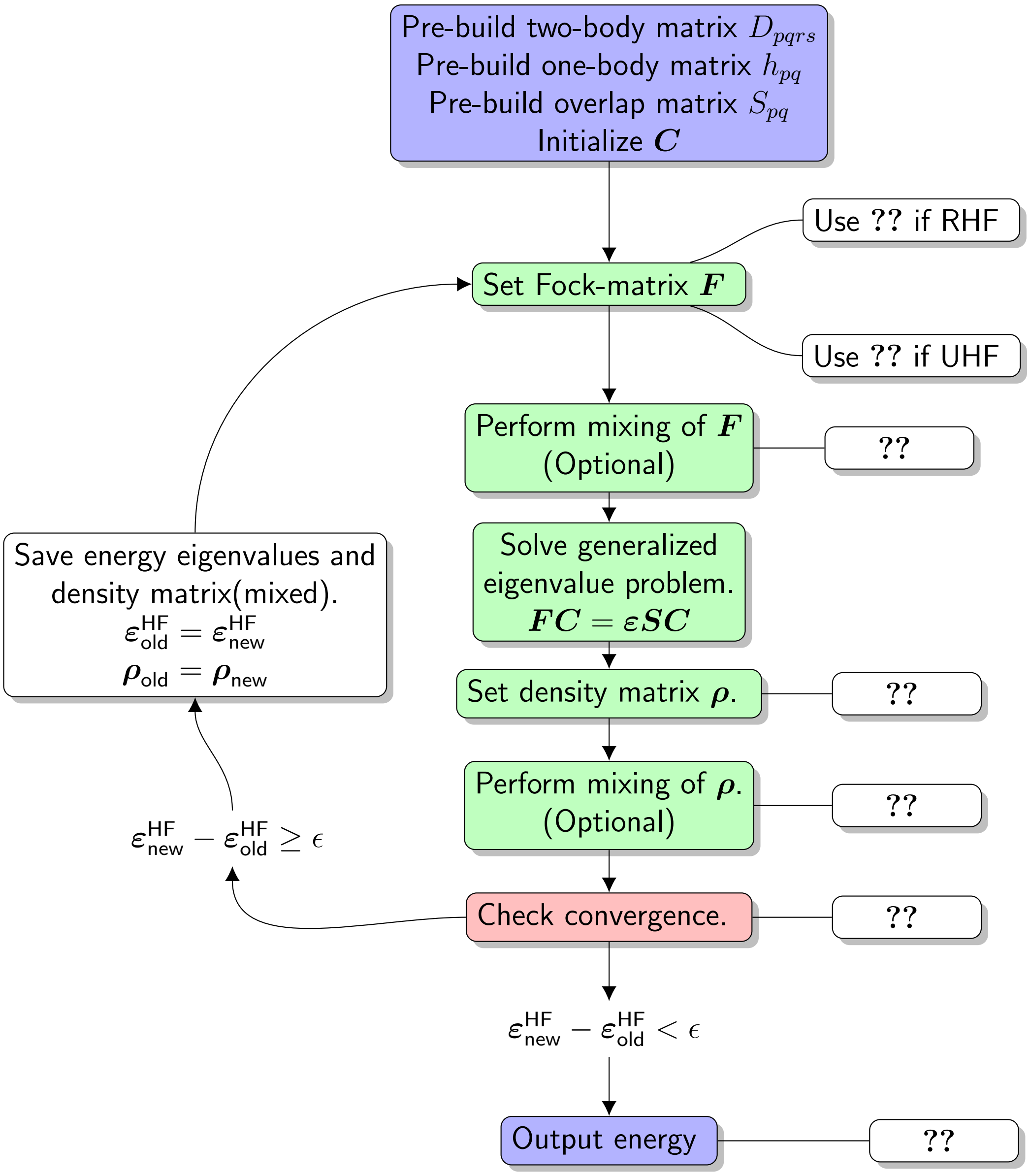
Pre-build two-body matrix D_{pqrs}
Pre-build one-body matrix h_{pq}
Pre-build overlap matrix S_{pq}
Initialize $\boldsymbol{C}$
Use ?? if RHF
Set Fock-matrix $\boldsymbol{F}$
Use ?? if UHF
Perform mixing of $\boldsymbol{F}$ (Optional)
??
Solve generalized eigenvalue problem. $\boldsymbol{FC} = \boldsymbol{\varepsilon SC}$
Save energy eigenvalues and density matrix(mixed). $\varepsilon^{\mathsf{HF}}_{\mathsf{old}} = \varepsilon^{\mathsf{HF}}_{\mathsf{new}}$ $\boldsymbol{\rho}_{\mathsf{old}} = \boldsymbol{\rho}_{\mathsf{new}}$
Set density matrix $\boldsymbol{\rho}$.
??
Perform mixing of $\boldsymbol{\rho}$. (Optional)
??
$\varepsilon^{\mathsf{HF}}_{\mathsf{new}} - \varepsilon^{\mathsf{HF}}_{\mathsf{old}} \geq \epsilon$
Check convergence.
??
$\varepsilon^{\mathsf{HF}}_{\mathsf{new}} - \varepsilon^{\mathsf{HF}}_{\mathsf{old}} < \epsilon$
Output energy
??

Figure 9.1: Hartree-Fock Algorithm.

9.3 Optimization algorithms

9.3.1 Stochastic Gradient Descent

9.3.2 ADAM

Chapter 10

Results

10.1 Comparison of Wave Functions

Compare machine learning wave functions to standard VMC wave functions

10.2 Ground state energies

Chapter 11

Conclusion and future work

See if machine learning is able to describe the three-body interaction, with nuclear physics applications.

Appendix A

Dirac notation

The Dirac notation, also called bracket notation, was suggested by Paul Dirac in a 1939 paper with the purpose of improving the reading ease. [Dir39]

Appendix B

Scaling

B.0.1 Harmonic Oscillator - Natural units

Hamiltonian is in one dimension given by

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2 \tag{B.1}$$

which has corresponding wavefunctions

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} \cdot \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} \exp\Big(-\frac{m\omega}{2\hbar}x^2\Big) H_n\Big(\sqrt{\frac{m\omega}{\hbar}}x\Big). \tag{B.2}$$

We want to get rid of $\hbar$ and m in equation (B.1), and we start with scaling $H' \equiv H/\hbar$, such that the Hamiltonian reduces to

$$H' = -\frac{\hbar}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}\frac{m\omega^2}{\hbar}x^2 \tag{B.3}$$

One can now observe that the fraction $\hbar/m$ comes in both terms, which we can take out by scaling the length $x' \equiv x \cdot \sqrt{m/\hbar}$. The final Hamiltonian is

$$H = \frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}\omega^2 x^2 \tag{B.4}$$

which corresponds to setting $\hbar = m = 1$. In natural units, one often sets $\omega = 1$ as well by scaling $H' = H/\hbar\omega$, but since we want to keep the ω-dependency, we do it slightly different. This means that the exact wavefunctions for the one-particle-one-dimension case is

$$\psi_n(x) = \exp\Big(-\frac{\omega}{2}x^2\Big) H_n(\sqrt{\omega}x) \tag{B.5}$$

where we take advantage of the Metropolis algorithm and ignore the normalization constant.

B.0.2 Atomic systems - Atomic units

Appendix C

Machine Learning Wave Function

We have seen that the probability of having a set of positions $\boldsymbol{X}$ with a set of hidden nodes $\boldsymbol{h}$ is given by

$$F(\boldsymbol{X},\boldsymbol{H}) = \frac{1}{Z}e^{-\beta E(\boldsymbol{X},\boldsymbol{H})} \tag{C.1}$$

where we set $\beta = 1/kT = 1$, Z is the partition function and $E(\boldsymbol{X},\boldsymbol{h})$ is the system energy

$$E(\boldsymbol{X},\boldsymbol{H}) = \sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{N} b_j H_j - \sum_{i,j=1}^{M,N}\frac{X_i W_{ij} H_j}{\sigma_i^2} \tag{C.2}$$

such that

$$F_{\text{RBM}}(\boldsymbol{X},\boldsymbol{h}) = e^{\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}} e^{\sum_{j=1}^{N}\left(b_j h_j + \sum_{i=1}^{M}\frac{X_i W_{ij}}{\sigma^2}\right)}. \tag{C.3}$$

We skip the partition function because it will not affect the results (it is just a normalization constant). The probability of a set of positions only is therefore the sum over all sets of $\boldsymbol{h}$, $\{\boldsymbol{h}\}$:

$$\begin{aligned}
F_{\text{RBM}}(\boldsymbol{X}) &= \sum_{\{\boldsymbol{h}\}} e^{\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}} \prod_{j=1}^{N} e^{b_j h_j + \sum_{i=1}^{M}\frac{X_i W_{ij} h_j}{\sigma^2}} \\
&= \sum_{h_1}\sum_{h_2}\dots\sum_{h_N} e^{\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}} \times \\
&\quad e^{b_1 h_1 + \sum_{i=1}^{M}\frac{X_i W_{i1} h_1}{\sigma^2}} e^{b_2 h_2 + \sum_{i=1}^{M}\frac{X_i W_{i2} h_2}{\sigma^2}} \dots e^{b_N h_N + \sum_{i=1}^{M}\frac{X_i W_{iN} h_N}{\sigma^2}} \\
&= e^{\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}} \prod_{j=1}^{N}\sum_{h_j=0}^{1} e^{b_j h_j + \sum_{i=1}^{M}\frac{X_i W_{ij} h_j}{\sigma^2}} \\
&= e^{\sum_{i=1}^{M}\frac{(X_i-a_i)^2}{2\sigma^2}} \prod_{j}^{N}\left(1 + e^{b_j + \frac{\boldsymbol{x}^T \boldsymbol{w}_{*j}}{\sigma^2}}\right)
\end{aligned} \tag{C.4}$$

Appendix D

Wave Function Elements

D.1 Kinetic Energy Calculations

The local energy, defined in equation (4.5), is

$$E_L = \frac{1}{\Psi_T}\hat{\mathrm{H}}\Psi_T \tag{D.1}$$

$$= \sum_{k=1}^{M}\Big[-\frac{1}{2\Psi_T}\nabla_k^2\Psi_T + U_k + V_k\Big]. \tag{D.2}$$

The first term, which is the kinetic energy term, is the only wave function-dependent one. It will in this appendix be evaluated for various wave function elements. From the definition of differentiation of a logarithm, we have that

$$\frac{1}{\Psi_T}\nabla_k\Psi_T = \nabla_k \ln\Psi_T, \tag{D.3}$$

which provides the following useful relation

$$\frac{1}{\Psi_T}\nabla_k^2\Psi_T = \nabla_k^2\ln\Psi_T + (\nabla_k\ln\Psi_T)^2. \tag{D.4}$$

Consider a trial wave function, Ψ_T, consisting of a product of p wave function elements, $\{\phi_1, \phi_2 \ldots \phi_p\}$,

$$\Psi_T = \prod_{i=1}^{p}\phi_i. \tag{D.5}$$

The kinetic energy related to this trial wave function is then computed by

$$\frac{1}{\Psi_T}\nabla_k^2\Psi_T = \sum_{i=1}^{p}\nabla_k^2\ln\phi_i + \Big(\sum_{i=1}^{p}\nabla_k\ln\phi_i\Big)^2, \tag{D.6}$$

which can be computed given all local derivatives $\nabla_k^2\ln\phi_i$ and $\nabla_k\ln\phi_i$. For each wave function element given below, those local derivatives will be evaluated. In addition, we need to know the derivative of the local energy with respect to the variational parameters in order to update the parameters correctly.

D.2 Parameter Update

In gradient based optimization methods, as we use, one needs to know the gradient of the local energy with respect to all variational parameters α_i,

$$\partial_{\alpha_i} E_L \equiv \frac{\partial E_L(\alpha_i)}{\partial \alpha_i}. \tag{D.7}$$

If we assume that each parameter, α, only exists in a wave function element,

$$\Psi_T(\alpha) = \phi_1(\alpha) \prod_{i=2}^{p} \phi_i \tag{D.8}$$

the derivative of the entire local energy is reduced to the derivative of the kinetic energy term given the wave function element,

$$\begin{aligned} \partial_\alpha E_L &= \partial_\alpha \Bigg(\nabla_k^2 \ln \phi_1(\alpha) + \sum_{i=2}^{p} \nabla_k^2 \ln \phi_i + \Big(\nabla_k \ln \phi_1(\alpha) + \sum_{i=2}^{p} \nabla_k \ln \phi_i \Big)^2 \Bigg) && \text{(D.9)} \\ &= \partial_\alpha \nabla_k^2 \ln \phi_1(\alpha) + 2 \Big(\sum_{i=1}^{p} \nabla_k \ln \phi_i \Big) \cdot \partial_\alpha \nabla_k \ln \phi_1(\alpha). && \text{(D.10)} \end{aligned}$$

The sum is already evaluated in the kinetic energy calculations, which means that to calculate the gradients, it is sufficient to calculate $\partial_\alpha \nabla_k \ln \phi_1(\alpha)$ and $\partial_\alpha \nabla_k^2 \ln \phi_1(\alpha)$ for all wave function elements with respect to their variational parameters.

D.3 Derivatives

D.3.1 Simple Gaussian

A natural starting point is the Gaussian function, since it appears in all harmonic oscillator calculations. For N particles in D dimensions, the function is given by

$$\Psi(\alpha, \boldsymbol{r}) = \exp\Big[-\frac{1}{2} \alpha \sum_{i=1}^{N} r_i^2 \Big] \tag{D.11}$$

where the derivative with respect to coordinate r_k is

$$\nabla_k \ln \Psi(\alpha) = -\alpha r_k \tag{D.12}$$

and the second derivative is

$$\nabla_k^2 \ln \Psi(\alpha) = -\alpha N D. \tag{D.13}$$

The gradients for those derivatives are

$$\partial_\alpha \nabla_k \ln \Psi(\alpha) = -r_k \tag{D.14}$$

and

$$\partial_\alpha \nabla_k^2 \ln \Psi(\alpha) = -ND \tag{D.15}$$

respectively.

D.3.2 Hydrogen-Like Orbitals

$$\Psi(\alpha, \boldsymbol{r}) = \exp\Big[-\frac{1}{2}\alpha\sum_{i=1}^{N} r_i\Big] \tag{D.16}$$

where the derivative with respect to coordinate r_k is

$$\nabla_k \ln\Psi(\alpha) = -\alpha \tag{D.17}$$

and the second derivative is

$$\nabla_k^2 \ln\Psi(\alpha) = 0 \tag{D.18}$$

The gradients for those derivatives are

$$\partial_\alpha \nabla_k \ln\Psi(\alpha) = -1 \tag{D.19}$$

and

$$\partial_\alpha \nabla_k^2 \ln\Psi(\alpha) = 0 \tag{D.20}$$

respectively.

D.3.3 Padé-Jastrow Factor

The Padé-Jastrow factor is introduced in order to take care of the correlations.

D.3.4 Slater Determinant

We need to introduce the Slater Determinant to study more fermions.

D.3.5 NQS-Gaussian

Now over to the real deal; the machine learning inspired wave function elements.

D.3.6 NQS-Jastrow Factor

Bibliography

[Dir39] P. A. M. Dirac. “A new notation for quantum mechanics”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), 416–418. DOI: `10.1017/S0305004100021162`.

[Gri05] D.J. Griffiths. *Introduction to quantum mechanics*. 2nd Edition. Pearson PH, 2005. ISBN: 0131911759.

[Pal05] Josef Paldus. *The beginnings of coupled-cluster theory: An eyewitness account*. Amsterdam: Elsevier, 2005, pp. 115 –147. ISBN: 9780444517197. DOI: `10.1016/B978-044451719-7/50050-0`.

[DCFSI07] T Daniel Crawford and Henry F. Schaefer III. “An Introduction to Coupled Cluster Theory for Computational Chemists”. In: vol. 14. Jan. 2007, pp. 33 –136. ISBN: 9780470125915. DOI: `10.1002/9780470125915.ch2`.

[Deb14] Sukanta Deb. “Variational Monte Carlo technique”. In: *Resonance* 19.8 (2014), pp. 713–739. ISSN: 0973-712X. DOI: `10.1007/s12045-014-0079-x`. URL: `https://doi.org/10.1007/s12045-014-0079-x`.