UNIVERSITÀ DEGLI STUDI DI TRENTO

DIPARTIMENTO DI FISICA

LAUREA IN FISICA

Variational Monte Carlo Study of Circular Quantum Dots

Supervisore:
Francesco Pederiva

Laureando:
Matteo De Pellegrin

DATA ESAME FINALE: 19 Novembre 2019

Contents

1 Introduction **1**

2 Basics of Monte Carlo Methods **3**
2.1 Monte Carlo evaluation of integrals 3
2.2 Markov chains and Metropolis algorithm 4
2.2.1 The correlation problem 6
2.3 Quantum Variational Monte Carlo 7

3 Computational Approach **9**
3.1 The scaled Hamiltonian 9
3.2 The trial wave function 10
3.2.1 Single particle orbitals 10
3.2.2 Slater determinants 10
3.2.3 The Pade-Jastrow factor 11
3.2.4 The total trial wave function 12
3.3 Computational optimizations of the Metropolis algorithm 12
3.3.1 computation of the acceptance probability 12
3.3.2 Calculating the local energy 12
3.3.3 Updating inverse of matrices 13
3.4 Reweighing method 14
3.5 DFP method 14

4 Results **17**
4.1 Simple Minimization with graphics 17
4.1.1 $N = 4$ and Hund's rules 18
4.2 DFP minimization 19
4.3 Validation 21

5 Conclusions **23**

List of Figures

1.1 scheme of a semiconductor Q-dot. 1

2.1 Metropolis algorithm scheme. 5

4.1 Reweighing methods producing $E(\boldsymbol{\alpha})$ plots. 18
4.2 Energy plots for $N = 2$, 3, 5, 6. 19
4.3 Energy plot for N=2. Lateral views. 20
4.4 Energy plots for $N = 4$ in all the possible configurations of L and S. . . . 21
4.5 Path of the DFP minimization for $N = 2$. 22

List of Tables

3.1 Slater determinants building orbitals for N from 2 to 6 particles 11

4.1 Energy minima and correspondent parameters for $N = 2, 3, 5, 6$ calculated by manually varying the parameters. 18
4.2 Energy minima and correspondent parameters for $N = 4$ calculated by manually varying the parameters. 19
4.3 Energy minima and correspondent parameters calculated with DFP. . . . 20
4.4 Comparison of our results with those in Ref[12] and Ref[8] for closed shell Q-dots. 22

Chapter 1

Introduction

In the last decades technology has advanced creating mesoscopic devices that are approaching dimensions of the order of nanometers. If it is true that this techonological improvement is happening thanks to more insight on the physics of these mesoscopic structures, it is also true that better technology allows better experiments that in turn result in deepening our understanding. There are also some cases where the theoretical study of this devices represents a better way of understanding their own physics: this is the case with Quantum dots.

Quantum dots are fabricated nanoscale structures capable to capture electrons. In particular it is possibile to create two-dimensional Q-dots from 2 layers of semiconductors (e.g GaAs Q-dots). Typical dimensions are about 10 nm of thickness and few hundreds of nanometers of width. Between the interfaces takes place a really narrow potential well that allows the formation of a quasi-two-dimensional electron gas.

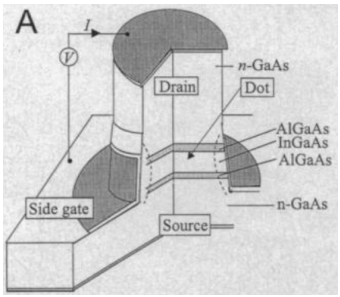

Figure 1.1: scheme of a semiconductor Q-dot. The drain and source contacts are used to probe the Q-dot properties while the side gate is used to tune the electron number[1].

Modern fabrication methods are able to make Q-dots so minute that the number of electrons they contain can be made arbitrary small. In addiction the properties of Quantum dots are easily tunable by using different designs and dimension or in real time with the use of external fields. For these reasons they are extremely interesting for practical applications, from light applications like lasers, photocathalists and Q-leds to quantum transistors or even the possibility of making quantum gates for quantum

computers[1], [2].

The theoretical study of quantum dots on the other hand is crucial to understand more on systems of fermions since they eliminate the nucleus-electron interaction. In particular Quantum dots have many atomic-like properties such a shell structure and they appear also to obey the Hund's first rule[1], [3]–[6]. For these resons they are often dubbed as *artificial atoms*.

In this thesis I will employ a simple Variational Monte Carlo computation of the ground states of Quantum dots containing from 2 to 6 electrons. The use of approximate statistical methods has been proven to give really accurate results[3], [4], [6]–[8] , even comparable with exact diagonalization methods, but their computation time is affected much less by the dimension of the system. The code is written in a object-oriented fashion using `C++`.

Chapter 2

Basics of Monte Carlo Methods

A good definition for Monte Carlo methods is given by Malvin H. Kalos and Paula A. Whitlock in Ref[9]:

> *A definition of a Monte Carlo method would be one that involves deliberate use of random numbers in a calculation that has the structure of a stochastic process.*

We should then clarify that a Monte Carlo method is not a simulation of a stochastic process but uses a stochastic process to solve non-probabilistic problems. Another important note we need to make, sice we are going to use computers to deploy these methods, is that machines cannot generate true-random numbers. What computers do with functions like `rand()` in `C` is to generate a sequence of numbers that maintain all the properties of random numbers, thus we may call them pseudo-random numbers.

2.1 Monte Carlo evaluation of integrals

Many computational problems involve the evaluation of integrals and in the majority of them the integration space is multidimensional. For example in classical mechanics we may want to simulate a gas in thermal equilibrium using the canonical ensemble. The problem reduces to the computation of the integral

$$\langle E \rangle = \frac{\int \mathcal{H}(\vec{r},\vec{p}) e^{-\beta \mathcal{H}(\vec{r},\vec{p})} d^{3N}\vec{r} d^{3N}\vec{p}}{\int e^{-\beta \mathcal{H}(\vec{r},\vec{p})} d^{3N}\vec{r} d^{3N}\vec{p}}. \tag{2.1}$$

To simulate a gas with a decent approximation to reality you cannot use less than 100 particles and even in this case you have to deal with two 600-dimensional integrals. Noticing that the kinetic part is given analytically we are still left with 300-dimensional integrals. Calculating this integrals with traditional methods (e.g. Simpson) requires evaluating the integrand at each point of a regular quadrature. To achieve for example 10% of relative error for the integral you would need to calculate the integrand on something around 10^D points, where D is the space dimension. Even if your computer could handle 10^{15} calculations of integrands per second and you only consider 20 particles

($D = 60$) you would need 10^{45} seconds, while the age of the universe is 10^{34} seconds. These are the kind of problems where Monte Carlo methods suits perfectly.

The Monte Carlo integration is based on the Central Limit theorem, which states that given $x_1, \ldots, x_N$ independent and identically distributed stochastic variables, if N is sufficiently large, the expectation value of $\langle f(x)\rangle = \int \mathcal{P}(x) f(x) dx$ can be approximated with its mean value $S_N = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$:

$$\lim_{N\to\infty} S_N = \langle f(x)\rangle. \tag{2.2}$$

Furthermore If N is finite, yet large, the theorem tells us that the probability distribution function of the mean value is a Gaussian with $FWHM = \sigma_N = \sqrt{\frac{1}{N}\langle (f - \langle f\rangle)^2\rangle}$:

$$\mathcal{P}(S_N) = \frac{1}{\sqrt{2\pi\sigma_N^2}} e^{-\frac{(S_N - \langle f\rangle)^2}{2\sigma_N^2}}. \tag{2.3}$$

σ_N represent the statistical error and it relates to the variance through the formula

$$\sigma_N = \sqrt{\frac{1}{N}(\langle f^2\rangle - \langle f\rangle^2)}. \tag{2.4}$$

This means that we can tackle integrals as the one in Equation 2.1 with a reasonable computation time by generating random variables distributed with the probability density function

$$\mathcal{P}(x) = \frac{e^{-\beta V(\vec{r})}}{\int e^{-\beta V(\vec{r})} d^{3N}\vec{r}}. \tag{2.5}$$

A question now arises spontaneously: How can I generate variables with a custom probability density function?

2.2 Markov chains and Metropolis algorithm

An effective method for sampling with custom probability is using the so-called Markov chains, that is essentially another way of referring to a random walk. Lets say that at time zero our system has coordinates $\boldsymbol{R}_0$ sampled from an arbitrary distribution $\mathcal{P}_0(\boldsymbol{R}_0)$. We then define a transition matrix $T_i(\boldsymbol{R}_{i+1} \leftarrow \boldsymbol{R}_i)$ that represent the probability density of finding the system in $\boldsymbol{R}_{i+1}$ after it was in $\boldsymbol{R}_i$. Therefore

$$\mathcal{P}_1 = \int d\boldsymbol{R}_0 \mathcal{P}_0(\boldsymbol{R}_0) T_0(\boldsymbol{R}_1 \leftarrow \boldsymbol{R}_0) = \hat{T}_0 \mathcal{P}_0(\boldsymbol{R}_0) \tag{2.6}$$

Where we introduced the integral operator $\hat{T}_i = \int d\boldsymbol{R}_i \mathcal{P}_i(\boldsymbol{R}_i) T_i(\boldsymbol{R}_{i+1} \leftarrow \boldsymbol{R}_i)$. Iterating the chain we find

$$\mathcal{P}_i(\boldsymbol{R}_i) = \hat{T}_{i-1}\mathcal{P}_{i-1}(\boldsymbol{R}_{i-1}) = \hat{T}_{i-1}\ldots\hat{T}_0\mathcal{P}_0 \tag{2.7}$$

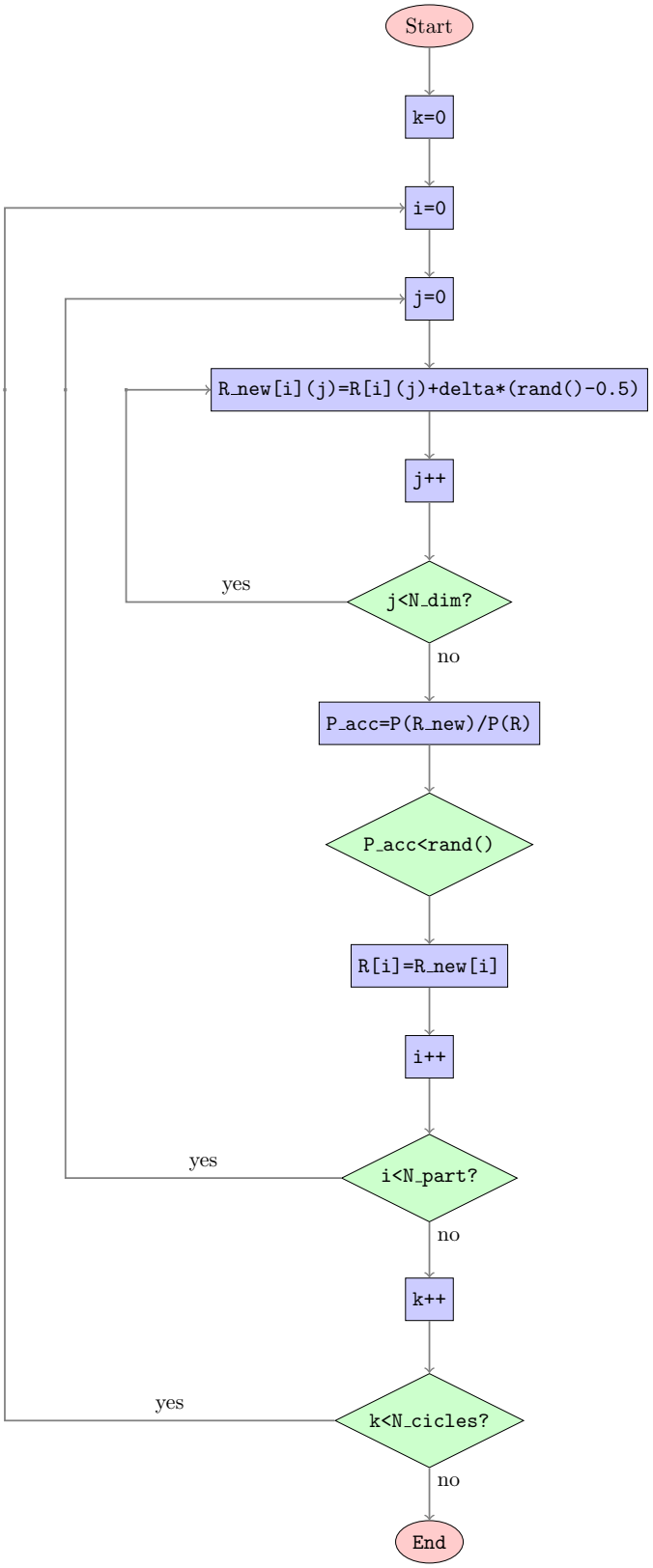

Figure 2.1: Metropolis algorithm scheme. Notice that `rand()` is supposed to generate uniform distributed random numbers in$(0, 1]$.

For our purposes we are only interested in stationary chains, namely $\hat{T}_j = \hat{T} \forall j$, so that the probability at step i depends only on the transition matrix and $\mathcal{P}_0$. Now lets suppose that our chain is convergent, and hence $\lim_{i\to\infty} = \mathcal{P}_\infty$, or in other words $\mathcal{P}_\infty$ is an eigenvector of $\hat{T}$ with eigenvalue 1 (otherwise we would loose the normalization property of $\mathcal{P}_\infty$). This means that if we iterate the chain from $\boldsymbol{R}_0$ with a transition matrix having such properties, after a sufficient amount of steps, say k, we will obtain a chain where every element is distributed with the probability density $\mathcal{P}_\infty$. Therefore calling m the total number of steps of the random walk, thanks to the Central Limit theorem, we can compute the expected value of our quantity f as

$$\frac{1}{m-k}\sum_{i=k+1}^{m} f(\boldsymbol{R}_i) \tag{2.8}$$

An easy way of defining a transition matrix with the properties described above is given by the Metropolis algorithm. The full derivation can be found in Ref[10] or more detailed in Ref[9]. Here I will only show a flow chart in Figure 2.1.

At each iteration the algorithm generates a new configuration moving the coordinates of the particles randomly within a box of sideΔ. Anyway it is convenient to consider new configurations also when only one particle coordinates change. Then it calculates the acceptance probability, defined as

$$\mathcal{P}_{acc} = \min\left(1, \frac{\mathcal{P}(\boldsymbol{R}_{new})}{\mathcal{P}(\boldsymbol{R})}\right) \tag{2.9}$$

where $\mathcal{P} = \mathcal{P}_\infty$ is the probability that we want to sample. This ratio represent the probability of accepting the transition. In order to follow this probability, when choosing if discard or accept the move, we can generate a uniform distributed random variable and accept when $\mathcal{P}_{acc}$ is smaller or discard otherwise.

2.2.1 The correlation problem

One of the hypothesis of the Central Limit theorem was that the variables need to be independent. Unfortunately variables generated with Markov chains are not independent, since every configuration depends on the previous one. If we use the Metropolis algorithm the key parameter that determines how much two neighbour configurations are correlated is the length Δ. If Δ is too large than new configurations are more likely to be discarded since they can easily move away from the equilibrium, while if it is too small configurations will be too similar. In both cases we obtain high correlation. One has to calibrate the parameter Δ in order to reduce correlation to a minimum. Even if we choose Δ to minimize correlation two neighbour samples will always be correlated, so we are left with 2 possibilities: taking "void" steps or blocking. The first means to compute the desired quantities every, say, τ time steps, so that for τ sufficiently large samples will be almost independent. The correlation of the samples can then be calculated as

$$\bar{C}_\tau(f) = \frac{\langle f(\boldsymbol{R}_i) f(\boldsymbol{R}_{i+\tau})\rangle - \langle f\rangle^2}{\langle f^2\rangle - \langle f\rangle^2}. \tag{2.10}$$

$C_\tau(f)$ usually has an exponential decay $C_\tau(f) \sim e^{-\frac{i}{\tau}}$,so that the true error can be approximated as $\sigma_N^2\tau$.

For blocking we mean that we calculate the average of the quantity for N_{block} time steps, and then we treat these mean values as our independent data samples. In this way the overall average will not change but the variance will be (about) the correct one.

Accurate calculation of Δ to minimize correlation is really expensive in terms of computation time and requires additional programming. A simple effective rule is to keep the acceptance ratio $\frac{accepted}{total}$ between 30% and 50% and use a conservative blocking[10]. I tried to keep acceptance around 40% and blocking every 1000 time steps.

2.3 Quantum Variational Monte Carlo

A recurrent problem in quantum mechanics is to calculate the eigenvalue of an observable:

$$\hat{O}|\Psi_k\rangle = O_k|\Psi_k\rangle \tag{2.11}$$

Unfortunately the analytical solution of this equations is rarely available and we have to use some numerical method instead. For some cases we can use exact diagonalization techniques, but usually the computational cost increases exponentially with the number of particles. On the other hand a large variety of Monte Carlo methods has been developed in order to solve this kind of problems and we usually refer to these methods as Quantum Monte Carlo methods (QMC).

In particular, if we want to find the ground state, the simplest method, but still largely used because of his effectiveness and interpretability, is the Variational Monte Carlo and as the name suggests it makes use of the variational principle:

$$E_0 \leq E[\Psi_T] = \frac{\langle\Psi_T|\mathcal{H}|\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle} = \frac{\int \Psi_T^*(\boldsymbol{R})\mathcal{H}(\boldsymbol{R})\Psi_T(\boldsymbol{R})d\boldsymbol{R}}{\int |\Psi_T(\boldsymbol{R})|^2 d\boldsymbol{R}}. \tag{2.12}$$

The principle means that given a trial wave function Ψ_T the energy associated will be always greater or at most equal than the actual ground state. The idea is to write a realistic trial wave function including some parameters $\boldsymbol{\alpha}$ so that we can minimize the energy as a function of those parameters. The variational principle in terms of the parameters would read

$$E_0 \leq E(\boldsymbol{\alpha}) = \frac{\int \Psi_T^*(\boldsymbol{R};\boldsymbol{\alpha})\mathcal{H}(\boldsymbol{R})\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})d\boldsymbol{R}}{\int |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2 d\boldsymbol{R}}. \tag{2.13}$$

At this point we might wonder if we can use some Monte Carlo integration method to solve the integral above, and indeed, just by a simple manipulation, we can rewrite it in the desired form:

$$\int \mathcal{P}(\boldsymbol{R};\boldsymbol{\alpha})E_{loc}(\boldsymbol{R};\boldsymbol{\alpha})d\boldsymbol{R} \tag{2.14}$$

Having defined the local energy as

$$E_{loc}(\boldsymbol{R};\boldsymbol{\alpha}) = \frac{\mathcal{H}\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})}{\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})} \tag{2.15}$$

And the probability is given by

$$\mathcal{P}(\boldsymbol{R};\boldsymbol{\alpha}) = \frac{|\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2}{\int |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2 d\boldsymbol{R}}. \tag{2.16}$$

The problem is now reduced to find the minimum in the parameters space of $E(\boldsymbol{\alpha})$. It is important to notice though that the variational principle does not ensure that the minum we find in this we will be equal to the actual ground state. If we want our minimum to be as close as possible to E_0 we need to use a realistic trial wave function.

Chapter 3

Computational Approach

In this work I deployed a Quantum Variational Monte Carlo method with a simple 2 parameters trial wave function, using the well-tested Metropolis algorithm. I improved the code in several ways to reduce the computation time, calculating the energy for more parameters configurations at the time using the reweighing method and using improved algorithms for updating matrices and calculating ratios between determinants. Furthermore I deployed the Davidon-Fletcher-Powell method for automatic minimization of the Energy.

3.1 The scaled Hamiltonian

The Hamiltonian of a Quantum dot in zero magnetic field is given by the three terms: kinetic, confinement and interaction. Written in CGS units reads

$$\mathcal{H}(\vec{r_1},\dots,\vec{r_N}) = \sum_{i=1}^{N} -\frac{\hbar^2\nabla_i^2}{2m^*} + \sum_{i=1}^{N} V_{conf}(\vec{r_i}) + \sum_{i<j} \frac{e^2}{\epsilon}\frac{1}{\|\vec{r_i}-\vec{r_j}\|}. \tag{3.1}$$

The most common confinement potential used in literature for circular Q-dots is the parabolic confinement, so that the single particle eigenfunctions are well-known and simple. Thus e can write the confinement as $V_{conf}(\vec{r_i}) = \frac{1}{2}m^*\omega r_i^2$.

In the previous formulas I used the electron effective mass m^* instead of the proper electron mass to correct the fact that I will consider a 2-dimensional system. For example if we consider a GaAs Q-dot the effective mass is $m^* = 0.067m_e$ and the dielectric constant $\epsilon = 12.4$[3].

The most convenient choice for scaling Equation 3.1 is to use the effective atomic units defined by setting $\hbar^2 = m^* = \frac{e^2}{\epsilon} = 1$, namely atomic units rescaled for m^* and ϵ. For the GaAs Q-dot we have the effective Bohr radius $a_0^* = a_0\frac{\epsilon m_e}{m^*} = 97.93\,\text{Å}$ and the effective Hartree $H^* = H\frac{m^*}{\epsilon^2 m_e} = 3.32\,\text{meV}$.

Finally I will only consider $\omega = 1$ for the parabolic potential. At the end the scaled Hamiltonian would read

$$\mathcal{H}(\vec{r_1},\ldots,\vec{r_N}) = \sum_{i=1}^{N} -\frac{\nabla_i^2}{2} + \sum_{i=1}^{N}\frac{1}{2}r_i^2 + \sum_{i<j}\frac{1}{\|\vec{r_i}-\vec{r_j}\|}. \tag{3.2}$$

3.2 The trial wave function

3.2.1 Single particle orbitals

The solutions of the 2-dimensional harmonic potential for one particle are well known, and can be written in terms of Hermite polynomials as

$$\Phi_{n_x,n_y}(x,y) = A(n_x,n_y)H_{n_x}H_{n_y}e^{-\frac{1}{2}(x^2+y^2)} \tag{3.3}$$

Where $A(n_x,n_y)$ is a normalization constant that will cancel out in the computation of the acceptance probability and local energy. The associated energies are $E_{n_x,n_y} = \hbar\omega(n_x+n_y+1)$.

We can also find a common basis for the Hamiltonian and the angular momentum $\vec{L} = L_z\hat{z}$. The derivation can be found in Ref[11] and yields the following wave functions written in polar coordinates:

$$\Phi_{n,m}(r,\phi) = A(n,m)e^{im\phi}r^{\|m\|}L_{n'}^{\|m\|}(r^2)e^{-\frac{1}{2}r^2} \tag{3.4}$$

Where $L_{n'}(r^2)$ are the Laguerre polynomials, $n = 0,1,...$ is the shell number, m is the angular momentum quantum number, with possible values $m = 0,\pm1,\pm2,...$ and $n' = \frac{n-m}{2}$. The corresponding energies are $E_n = \hbar\omega(n+1)$.

3.2.2 Slater determinants

To describe systems of fermions we usually write the total wave function as a sum of Slater determinants:

$$\Psi(\vec{r_1},\ldots,\vec{r_N}) = \sum_i c_i|D_i| \tag{3.5}$$

Where c_i are in general complex coefficients. In our case we will consider only one Slater determinants built with the single particle states of lower energies in order to find the ground state. In addition, as pointed out in Ref[12], [13], since our Hamiltonian is spin-independent we can split the Slater determinants in 2 smaller determinants for spin-up and for spin-down particles:

$$|D| = \frac{1}{(N/2)!}|D^{\uparrow}||D^{\downarrow}| \tag{3.6}$$

Where $D^{\uparrow}$ and $D^{\downarrow}$ are the matrices built respectively with the orbitals of the spin-up and spin-down particles. In Table 3.1 are presented all the studied configurations with the building orbitals chosen from the single particle states of lowest energy compatible

with the Pauli Exclusion Principle. Notice that for $N = 4$ there are 3 possible configurations for the total angular momentum L and the total spin S, so we have to consider them separately because we expect 3 different energies. In particular we can see which configuration correspond with the ground state and if the Hund's rules are satisfied.

N	L	S	Φ for $D^{\uparrow}$	Φ for $D^{\downarrow}$
2	0	0	$\Phi_{0,0}$	$\Phi_{0,0}$
3	1	1/2	$\Phi_{0,0}, \Phi_{1,1}$	$\Phi_{0,0}$
4	0	0	$\Phi_{0,0}, \Phi_{1,1}$	$\Phi_{0,0}, \Phi_{1,-1}$
4	2	0	$\Phi_{0,0}, \Phi_{1,1}$	$\Phi_{0,0}, \Phi_{1,1}$
4	0	1	$\Phi_{0,0}, \Phi_{1,1}, \Phi_{1,-1}$	$\Phi_{0,0}$
5	1	1/2	$\Phi_{0,0}, \Phi_{1,1}, \Phi_{1,-1}$	$\Phi_{0,0}, \Phi_{1,1}$
6	0	0	$\Phi_{0,0}, \Phi_{1,1}, \Phi_{1,-1}$	$\Phi_{0,0}, \Phi_{1,1}, \Phi_{1,-1}$

Table 3.1: Slater determinants building orbitals for N from 2 to 6 particles

3.2.3 The Pade-Jastrow factor

In the Hartree-Fock approach one would include some variational parameters in the component orbitals of the determinants and write the trial wave function as a single Slater determinant. Another largely used method to describe the interaction, and the choice of mine, is to multiply for an explicit correlation factor $J(\vec{r_1}, \ldots, \vec{r_N})$ usually called Jastrow function. This approach has already been deployed with success for Quantum dots (see for example Refs[3], [4], [6], [7]). The most commonly used is the Pade-Jastrow wave function[14], that consider only the 2-body interaction, and can in general be written as

$$J = \prod_{i<j} e^{g(r_{ij})} \tag{3.7}$$

With $r_{ij} = \|\vec{r_i} - \vec{r_j}\|$ and

$$g(r_{ij}) = \frac{\sum_{k=1}^{n} a_k r_{ij}^k}{1 + \sum_{k=1}^{n} b_k r_{ij}^k} \tag{3.8}$$

This function is symmetric in order to keep the antisymmetry of the total wave function. In this work I will consider only a Linear Pade-Jastrow function with 2 variational parameters a and b:

$$g(r_{ij}) = \frac{a r_{ij}}{1 + b r_{ij}} \tag{3.9}$$

3.2.4 The total trial wave function

Finally we can write the total trial wave function as the product of a single Slater determinant and a simple Pade-Jastrow factor. Without the normalization constants that cancel out in our computations it reads

$$\Psi_T(\vec{r_1}, \ldots, \vec{r_N}) = |D^{\uparrow}||D^{\downarrow}|J \tag{3.10}$$

3.3 Computational optimizations of the Metropolis algorithm

3.3.1 computation of the acceptance probability

Using the Metropolis algorithm we have to deal with the computation of the acceptance probability in Equation 2.9, that in Quantum Variational Monte Carlo, recalling Equation 2.16 and Equation 3.10, means the evaluation of the following ratio:

$$\frac{|D^{\uparrow}(\boldsymbol{R}_{new})|}{|D^{\uparrow}(\boldsymbol{R}_{old})|} \frac{|D^{\downarrow}(\boldsymbol{R}_{new})|}{|D^{\downarrow}(\boldsymbol{R}_{old})|} \frac{J(\boldsymbol{R}_{new})}{J(\boldsymbol{R}_{old})}. \tag{3.11}$$

First thing that we have to notice is that we move one particle at the time and so, depending on whether this particle has spin up or down, only one of the 2 determinant ratio has to be evaluated. Further more only one row of the matrix D^{α} has changed. Under these considerations an efficient algorithm is given below. The full derivation is available in Ref[12], [13].

$$\frac{|D^{\alpha}(\boldsymbol{R}_{new})|}{|D^{\alpha}(\boldsymbol{R}_{old})|} = \sum_{j=1}^{n_{\alpha}} \Phi_j(\vec{r}_i^{\,new}) D_{ji}^{\alpha^{-1}}(\boldsymbol{R}_{old}) \tag{3.12}$$

Where the particle that has been moved is the i-th, n_{α} is the number of electrons in the matrix D^{α} and $D^{\alpha^{-1}}$ is its inverse.

For the Jastrow function again we consider only the terms that has changed. With simple algebraic operations we can find

$$\frac{J(\boldsymbol{R}_{new})}{J(\boldsymbol{R}_{old})} = \exp\left(\sum_{j=1}^{i-1} \left(g(r_{ij}^{new}) - g(r_{ij}^{old}) \right) + \sum_{j=i+1}^{N} \left(g(r_{ji}^{new}) - g(r_{ji}^{old}) \right) \right) \tag{3.13}$$

3.3.2 Calculating the local energy

When calculating the local energy defined in Equation 2.15 we have to deal the following expression:

$$E_{loc}(\boldsymbol{R}) = \sum_{i=1}^{N} -\frac{1}{2} \frac{\nabla_i^2 \Psi_T}{\Psi_T} + \sum_{i=1}^{N} \frac{1}{2} r_i^2 + \sum_{i<j} \frac{1}{r_{ij}} \tag{3.14}$$

The kinetic part happens to be the only challenging part. Using the Leibniz rule we can split the problem in the evaluation of 4 ratios:

$$\frac{\nabla_i^2 \Psi_T}{\Psi_T} = \frac{\nabla_i^2 |D^\alpha(\boldsymbol{R})|}{|D^\alpha(\boldsymbol{R})|} + \frac{\nabla_i^2 J(\boldsymbol{R})}{J(\boldsymbol{R})} + 2\frac{\nabla_i |D^\alpha|}{|D^\alpha(\boldsymbol{R})|} \cdot \frac{\nabla_i J(\boldsymbol{R})}{J(\boldsymbol{R})} \tag{3.15}$$

Since ∇_i only affect the coordinates of the i-th particle and hence only one row of one matrix D^α and only few term of the Jastrow factor. The solution of the determinant part is given in Ref[13] and follows from a simple generalization of Equation 3.17.

$$\frac{\nabla_i |D^\alpha(\boldsymbol{R})|}{|D^\alpha(\boldsymbol{R})|} = \sum_{j=1}^{n_\alpha} \nabla_i \Phi_j(\vec{r_i}) D_{ji}^{\alpha^{-1}}(\boldsymbol{R}) \tag{3.16}$$

$$\frac{\nabla_i^2 |D^\alpha(\boldsymbol{R})|}{|D^\alpha(\boldsymbol{R})|} = \sum_{j=1}^{n_\alpha} \nabla_i^2 \Phi_j(\vec{r_i}) D_{ji}^{\alpha^{-1}}(\boldsymbol{R}) \tag{3.17}$$

The correlation part can also be simplified to the formulas below. Again the full derivation can be found in Ref[13].

$$\frac{\nabla_i J(\boldsymbol{R})}{J(\boldsymbol{R})} = \sum_{j=1}^{i-1} \frac{\vec{r_i} - \vec{r_j}}{r_{ij}} \frac{\partial g(r_{ij})}{\partial r_{ij}} + \sum_{j=i+1}^{N} \frac{\vec{r_i} - \vec{r_j}}{r_{ji}} \frac{\partial g(r_{ji})}{\partial r_{ji}} \tag{3.18}$$

$$\frac{\nabla_i^2 J(\boldsymbol{R})}{J(\boldsymbol{R})} = \left(\frac{\nabla_i^2 J(\boldsymbol{R})}{J(\boldsymbol{R})}\right)^2 + \sum_{j=1}^{i-1} \left[\frac{d-1}{r_{ij}} \frac{\partial g(r_{ij})}{\partial r_{ij}} + \frac{\partial^2 g(r_{ij})}{\partial r_{ij}^2}\right] + \sum_{j=i+1}^{N} \left[\frac{d-1}{r_{ji}} \frac{\partial g(r_{ji})}{\partial r_{ji}} + \frac{\partial^2 g(r_{ji})}{\partial r_{ji}^2}\right] \tag{3.19}$$

Where d is the dimension of the space, $d = 2$ in our case. For our $g(r_{ij})$ the derivatives are the following:

$$\frac{\partial g(r_{ij})}{\partial r_{ij}} = \frac{a}{(1 + b r_{ij})^2} \tag{3.20}$$

$$\frac{\partial^2 g(r_{ij})}{\partial r_{ij}^2} = \frac{-2ab}{(1 + b r_{ij})^3} \tag{3.21}$$

3.3.3 Updating inverse of matrices

Every time a move attempt is accepted, say the particle i, we need to update the inverse of the matrix D^α. Updating the matrix and recalculating the inverse would be too time expensive. A much more efficient algorithm for updating the inverse is given in Ref[12]–[14]:

$$D_{kj}^{new^{-1}} = \begin{cases} D_{kj}^{\alpha^{-1}}(\boldsymbol{R}_{old}) - \frac{D_{ki}^{\alpha^{-1}}(\boldsymbol{R}_{old})}{\boldsymbol{R}_D} \sum_{l=1}^{n_\alpha} \Phi_l(\vec{r_i^{new}}) D_{lj}^{\alpha^{-1}}(\boldsymbol{R}_{old}) & \text{if } j \neq i \\ \frac{D_{ki}^{\alpha^{-1}}(\boldsymbol{R}_{old})}{\boldsymbol{R}_D} & \text{if } j = i \end{cases} \tag{3.22}$$

Where R_D is the ratio defined in Equation 3.17.

3.4 Reweighing method

Lets call $\boldsymbol{\alpha}$ our set of variational parameters and $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \delta\boldsymbol{\alpha}$ with slightly changed values. The expectation value for the local energy for the set $\boldsymbol{\alpha}'$ is then given by

$$\langle E_{loc}(\boldsymbol{\alpha}')\rangle = \frac{\int |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha}')|^2 E_{loc}(\boldsymbol{R};\boldsymbol{\alpha}')d\boldsymbol{R}}{\int |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha}')|^2 d\boldsymbol{R}} \tag{3.23}$$

or we may write

$$\langle E_{loc}(\boldsymbol{\alpha}')\rangle = \frac{\int \frac{|\Psi_T(\boldsymbol{R};\boldsymbol{\alpha}')|^2}{|\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2} |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2 E_{loc}(\boldsymbol{R};\boldsymbol{\alpha}')d\boldsymbol{R}}{\int \frac{|\Psi_T(\boldsymbol{R};\boldsymbol{\alpha}')|^2}{|\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2} |\Psi_T(\boldsymbol{R};\boldsymbol{\alpha})|^2 d\boldsymbol{R}} \tag{3.24}$$

This last equation is telling us that we could use configurations generated from the probability $\mathcal{P}(\boldsymbol{R};\boldsymbol{\alpha})$ and still calculate the expected value for the parameters $\boldsymbol{\alpha}'$ by simply reweighing the samples. The mean we will compute then is simply a weighted average[10]:

$$\langle E_{loc}(\boldsymbol{\alpha}')\rangle \simeq \frac{\sum_k \frac{|\Psi_T(\boldsymbol{R}_k;\boldsymbol{\alpha}')|^2}{|\Psi_T(\boldsymbol{R}_k;\boldsymbol{\alpha})|^2} E_{loc}(\boldsymbol{R}_k;\boldsymbol{\alpha}')}{\sum_k \frac{|\Psi_T(\boldsymbol{R}_k;\boldsymbol{\alpha}')|^2}{|\Psi_T(\boldsymbol{R}_k;\boldsymbol{\alpha})|^2}} \tag{3.25}$$

Where the $\boldsymbol{R}_k$ are sampled using the set of parameters $\boldsymbol{\alpha}$. This is called reweighing methods, and it allows to reduce greatly the amount of time needed to find the minimum, allowing faster exploration of the parameter space and speeding up the computation of gradients in the parameters space. Furthermore gradients computed with this technique are much more accurate because the statisical fluctuations of $\langle E_{loc}(\boldsymbol{\alpha})\rangle$ are exactly the same of the one of $\langle E_{loc}(\boldsymbol{\alpha}')\rangle$. The access to gradients opens the possibility of using algorithms to find minima such as the Gradient Descent, the Conjugate Gradient and all theirs variations.

3.5 DFP method

For finding the parameters that minimize the energy one would like to automate the process rather than varying manually the parameters. In this thesis I will use the Davidon-Fletcher-Powell (DFP) algorithm[12], [15]. This algorithm is based on the conjugate gradient method and the steepest descent. Here I will only give a basic idea of the algorithm while linking the full derivation Ref[15]. Consider the Taylor expansion of the function $f(\boldsymbol{\alpha})$ around $\boldsymbol{\alpha}_i$:

$$f(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}_i) + \boldsymbol{b}\cdot\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}\cdot\boldsymbol{A}\boldsymbol{\delta} \tag{3.26}$$

Having defined

$$\boldsymbol{\delta} = \boldsymbol{\alpha} - \boldsymbol{\alpha}_i,\ \boldsymbol{b} = \nabla f(\boldsymbol{\alpha}_i),\ A_{kl} = \frac{\partial^2 f}{\partial \alpha_i^k \partial \alpha_i^l} \tag{3.27}$$

Doing this we are basically approximating our function with a quadratic one. In this second order approximation the gradient can be written as

$$\nabla f(\boldsymbol{\alpha}) = \boldsymbol{b} + \boldsymbol{A}(\boldsymbol{\alpha} - \boldsymbol{\alpha_i}) \tag{3.28}$$

And therefore minimizing f is equivalent to solve a linear system. In Newton's method one would set $\nabla f(\boldsymbol{\alpha}_{i+1}) = 0$ in order to find the next iteration point. In this case we choose the direction as in Newton's method:

$$\boldsymbol{s}_i = -\boldsymbol{A}^{-1} \nabla f(\boldsymbol{\alpha}_i) \tag{3.29}$$

And the next iteration point is given by

$$\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i + \gamma_i \boldsymbol{s}_i \tag{3.30}$$

Where the parameter γ is found through the 1-dimensional minimization of $f(\boldsymbol{\alpha}_i + \gamma_i \boldsymbol{s}_i)$. For this minimization I will use the Newton's method.

Furthermore instead of using the true Hessian of f the DFP method actually uses an approximation of its inverse usually named $\boldsymbol{H}_i$. The class of algorithms that uses approximations for the Hessian inverse are called *quasi-Newton* methods or also *variable metric* methods. The first advantage of using such methods is the fact that the approximate Hessian is built to be positive definite, while the true Hessian could not be so, and thus it prevents from reaching maxima instead of minima and obviously we do not need to compute the Hessian. For H_0 I will simply use I_2 that is the same as choosing the steepest descent direction for the first iteration. The algorithm for updating H_i is taken from Refs[12], [15], [16]:

$$\boldsymbol{H}_{i+1} = \boldsymbol{H}_i + \frac{\boldsymbol{\delta} \otimes \boldsymbol{\delta}}{\boldsymbol{\delta} \cdot \boldsymbol{\xi}} - \frac{(\boldsymbol{H}_i \boldsymbol{\xi}) \otimes (\boldsymbol{H}_i \boldsymbol{\xi})}{\boldsymbol{\xi} \cdot \boldsymbol{H}_i \boldsymbol{\xi}} \tag{3.31}$$

Where $\boldsymbol{\delta} = \boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i$, $\boldsymbol{\xi} = \nabla f(\boldsymbol{\alpha}_{i+1}) - \nabla f(\boldsymbol{\alpha}_i)$ and $\otimes$ is the outer product. Summarizing everything the algorithm looks like the following[15], [16]:

1. set $\boldsymbol{s}_i = -\boldsymbol{H}_i \nabla f(\boldsymbol{\alpha}_i)$,
2. line search along $\boldsymbol{s}_i$ giving $\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i + \gamma_i \boldsymbol{s}_i$,
3. update $\boldsymbol{H}_i$ giving $\boldsymbol{H}_{i+1}$.

The algorithm is proven to find the exact minimum in n iteration, like the conjugate gradient method, for quadratic n-dimensional functions and it is able to keep the number of iteration low for more complex functions.

Finally, taking advantage of the reweighing method (section 3.4), I use a 2-dimensional 5-points stencil formula to compute gradients:

$$\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha^k} = \frac{f(\boldsymbol{\alpha} + \boldsymbol{\delta}_k) - f(\boldsymbol{\alpha} - \boldsymbol{\delta}_k)}{2d\alpha^k}, \tag{3.32}$$

Where $\boldsymbol{\delta}_k = \begin{pmatrix} \vdots \\ \delta\alpha^k \\ \vdots \end{pmatrix}$, and 1-dimensional 5-points stencil formulas to compute linear derivatives:

$$\frac{d}{d\gamma}\Phi(\gamma) = \frac{\Phi(\gamma - 2\delta\gamma) - 8\Phi(\gamma - \delta\gamma) + 8\Phi(\gamma + \delta\gamma) - \Phi(\gamma + 2\delta\gamma)}{12\delta\gamma} \tag{3.33}$$

$$\frac{d^2}{d\gamma^2}\Phi(\gamma) = \frac{-\Phi(\gamma - 2\delta\gamma) + 16\Phi(\gamma - \delta\gamma) - 30\Phi(\gamma) + 16\Phi(\gamma + \delta\gamma) - \Phi(\gamma + 2\delta\gamma)}{12\delta\gamma^2} \tag{3.34}$$

With $\Phi(\gamma) = f(\boldsymbol{\alpha} + \gamma\boldsymbol{s})$.

Chapter 4

Results

In this chapter I will present the results obtained for Q-dots with 2 to 6 electrons. In order to find the optimal parameters I proceeded as follows:

1. Produce a plot of the function $E(\boldsymbol{\alpha})$ to see how it behaves,
2. starting from a resonable value looking at the plot, use DFP minimization to find the optimal parameters.

The uncertainties are calculated through Equation 2.4 assuming no statistical correlation (see subsection 2.2.1).

4.1 Simple Minimization with graphics

The first thing I did was to calculate the energy in a range of possible values for the variational parameters to make graphics and see how the energy behaves in the parameters space. The values are calculated 9 at the time, thanks to the reweighing technique (section 3.4), using the central values to sample the configurations as shown in Figure 4.1. Doing this we are able to reduce the computation time of about 70%. Since we are only interested in understanding the qualitative behaviour of the energy respect to the parameters, to make this plots we use just 10^6 Monte Carlo cycles for each configuration. I noticed that the computation time goes like $\sim \frac{N^2}{4}$, meaning that the algorithm is adequately optimized and it could handle bigger systems as well.

In Figure 4.2 are shown the plots of the energy for $N = 2$, 3, 5 and 6 electrons. For generating these graphics I varied the parameters through 21 values for a and 21 for b. As we can see the surface has a strong valley shape, meaning that there is a direction along which it is harder to find the exact minimum. Anyway this is not a big problem since in that direction, around the minimum the energy variations are comparable with our uncertainties. Orienting the a or b axis to be perpendicular to the sheet we can already try to extrapolate the optimal parameters and the minimized energy as presented in Figure 4.3 for the 2-electrons case. In Table 4.1 are presented the minimum registered energy values of the plots together with the corresponding parameters for $N = 2$, 3, 5 and 6.

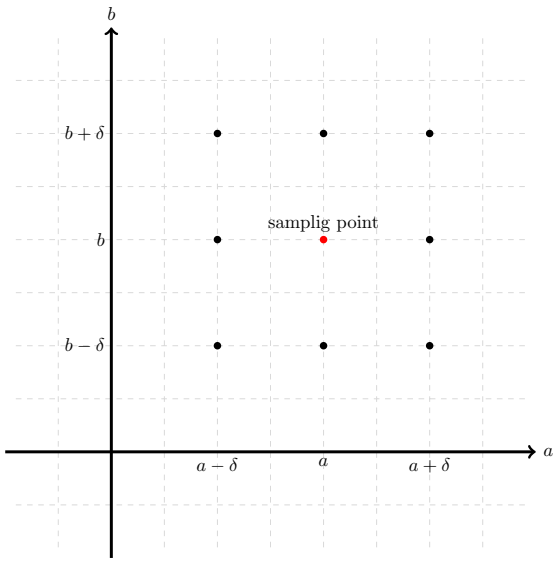

Figure 4.1: Reweighing methods producing $E(\boldsymbol{\alpha})$ plots.

N	E_{min} $[H*]$	a	b
2	3.0001(2)	0.820	0.275
3	6.3757(4)	0.98	0.50
5	14.9947(7)	0.94	0.56
6	20.2073(8)	0.98	0.60

Table 4.1: Energy minima and correspondent parameters for $N = 2$, 3, 5, 6 calculated by manually varying the parameters. In parentheses are the uncertainties in the last digit. All values are for $\omega = 1$

4.1.1 $N = 4$ and Hund's rules

The results for the three configurations for the 4-particle Quantum dot are presented in Table 4.2 and Figure 4.4. We notice that the ground state is given by $L = 0$ and $S = 1$ in agreement with the Hund's first rule, which states that for a given electron configuration the term with the maximum multiplicity has the lowest energy.

The Hund's first rule has been seen to be followed by Q-dots also in paper like Refs[3], [4] and experimentally in Refs[1], [5]. The second Hund's rules, for which at a given multiplicity the lowest energy is given by the highest angular momentum, is not seen to be valid for Q-dots.

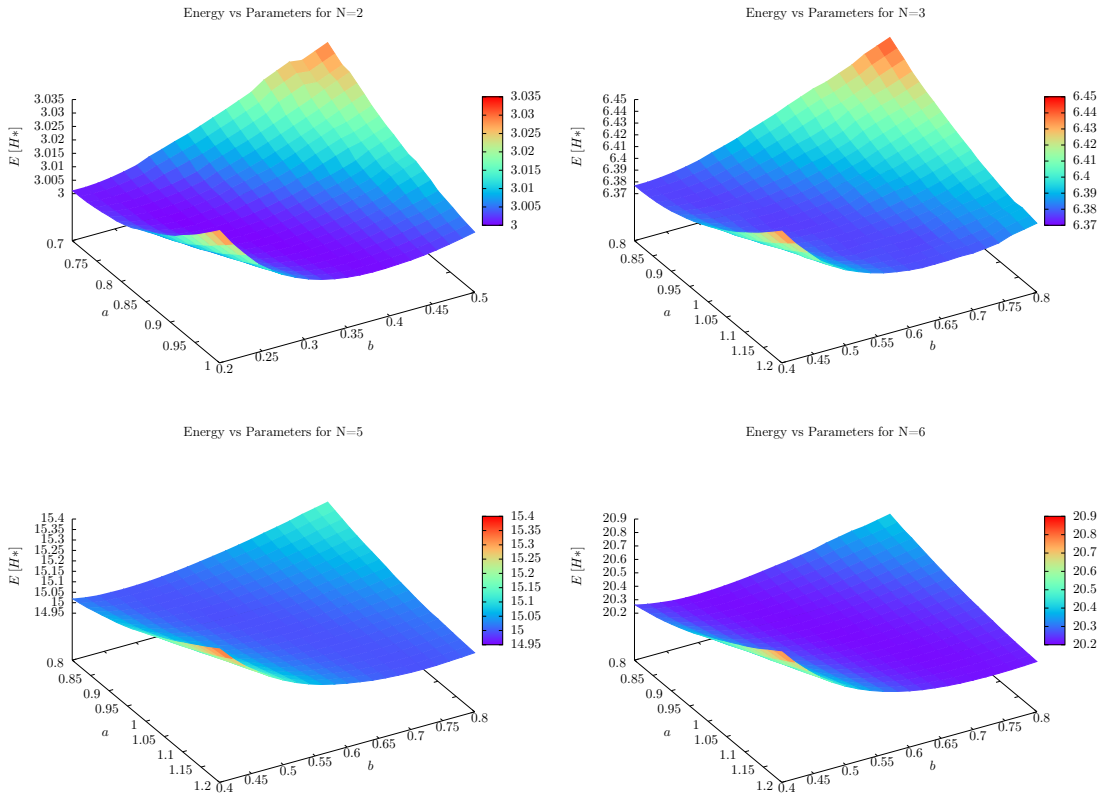

Figure 4.2: Energy plots for $N = 2,\ 3,\ 5,\ 6$.

L	S	$E_{min}\ [H*]$	a	b
0	1	10.2892(5)	0.96	0.58
0	0	10.351(2)	1.18	0.7
2	0	10.4878(6)	1.04	0.58

Table 4.2: Energy minima and correspondent parameters for $N = 4$ calculated by manually varying the parameters. In parentheses are the uncertainties in the last digit. All values are for $\omega = 1$.

4.2 DFP minimization

Calculations with the Davidon-Fletcher-Powell algorithm were challenging because of the flatness about the minimum of the Energy. I used a step of 0.0001 for calculating

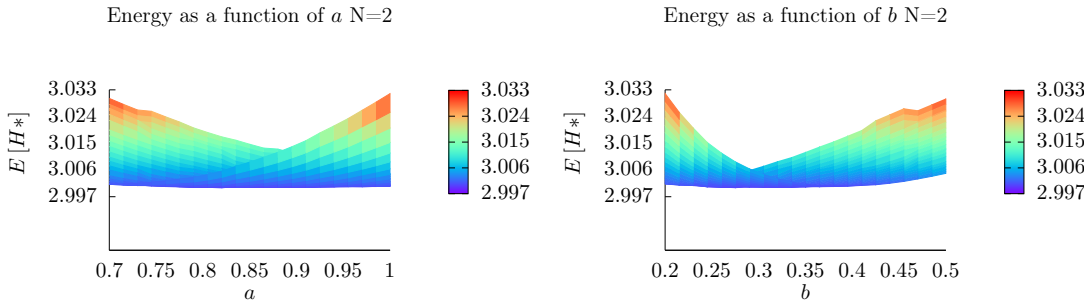

Figure 4.3: Energy plot for N=2. Lateral views.

derivates and 5×10^7 Monte Carlo cycles for each energy calculation. The computation of the minimum stops either when the gradient satisfies the threshold $\|\nabla E(\boldsymbol{\alpha})\| \leq 0.001$ or when the Newton's algorithm for the line search stops working because derivatives become too small compared with the statistical fluctuations of the energy. In order to achieve more accuracy I would needed to use more MC-cycles and calculation would have been unsustainable for the computational power and the time at my disposal.

N	L	S	$E_{min}\,[H*]$	a	b
2	0	0	3.00013(2)	0.87893	0.31450
3	1	1/2	6.37632(6)	1.03177	0.55051
4	0	1	10.28932(7)	0.95629	0.57807
4	0	0	10.35134(7)	1.00507	0.58889
4	2	0	10.48903(9)	1.02226	0.56605
5	1	1/2	14.99571(9)	0.97274	0.58759
6	0	0	20.2086(1)	0.992936	0.611292

Table 4.3: Energy minima and correspondent parameters calculated with DFP. In parentheses are the uncertainties in the last digit. All values are for $\omega = 1$.

The values are close to those estimated with manually plotting the energy but uncertainties are ~ 10 times less. The minimized parameters also changed and this is certainly due to the high uncertainties of the energies in the plots, but also to the flatness about the minima that makes difficult finding exact parameters. In Figure 4.5 are shown the iterations that took the algorithm to reach the minimum for the case $N = 2$. The first iteration went along the steepest descent direction, while the next iteration was pointing already right about the minima. The small steps after the change in direction are made in order to find a point with positive second derivative because using the Newton's method with a negative second derivative would lead to a maximum rather than a minimum.

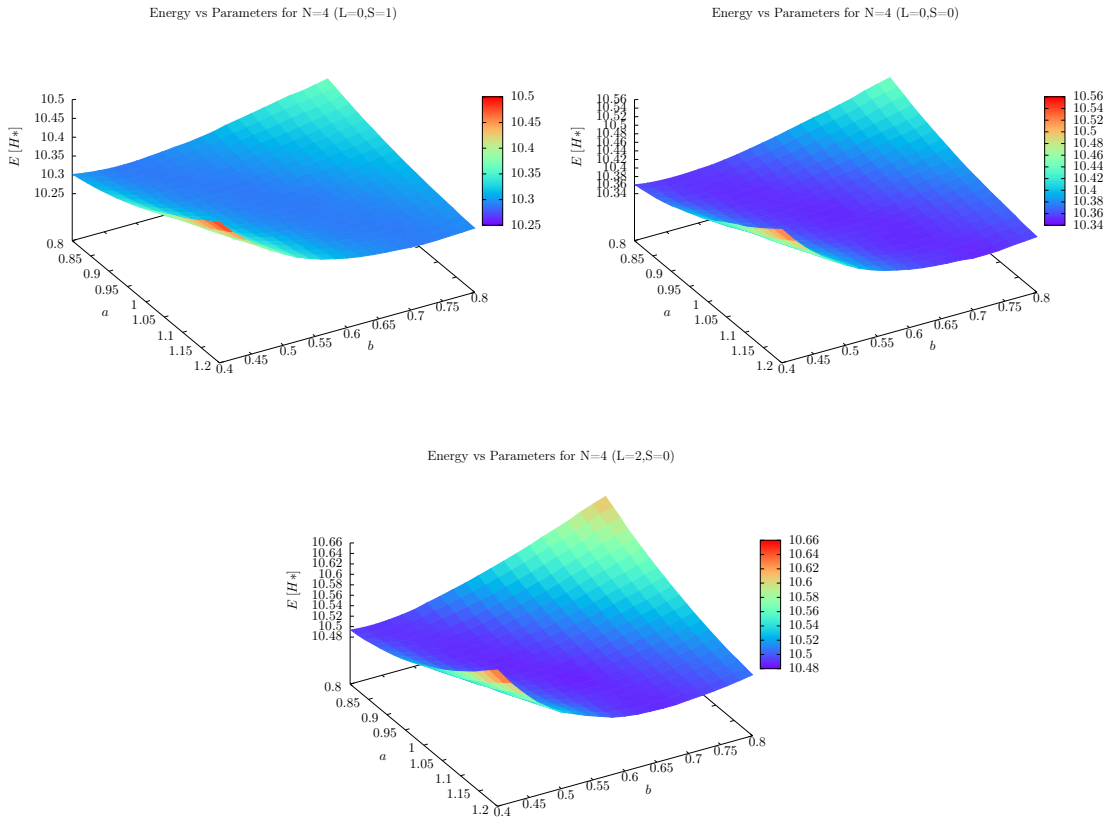

Figure 4.4: Energy plots for $N = 4$ in all the possible configurations of L and S.

4.3 Validation

Comparisons between results of different work are not easy because of the different values used for ϵ and ω. The only works I found to have those values as mine are Ref[12] with variational Monte Carlo and Ref[8] with diffusion Monte Carlo, but they made calculation only for closed shell Quantum dots, namely $N = 2$ and $N = 6$. Their results, together with mine, are shown in Table 4.4. The difference between the results is mainly due to the different trial wave functions used and the method. In particular we notice smaller values for the diffusion Monte Carlo technique. Anyway this comparison is good to validate the rightness of my results.

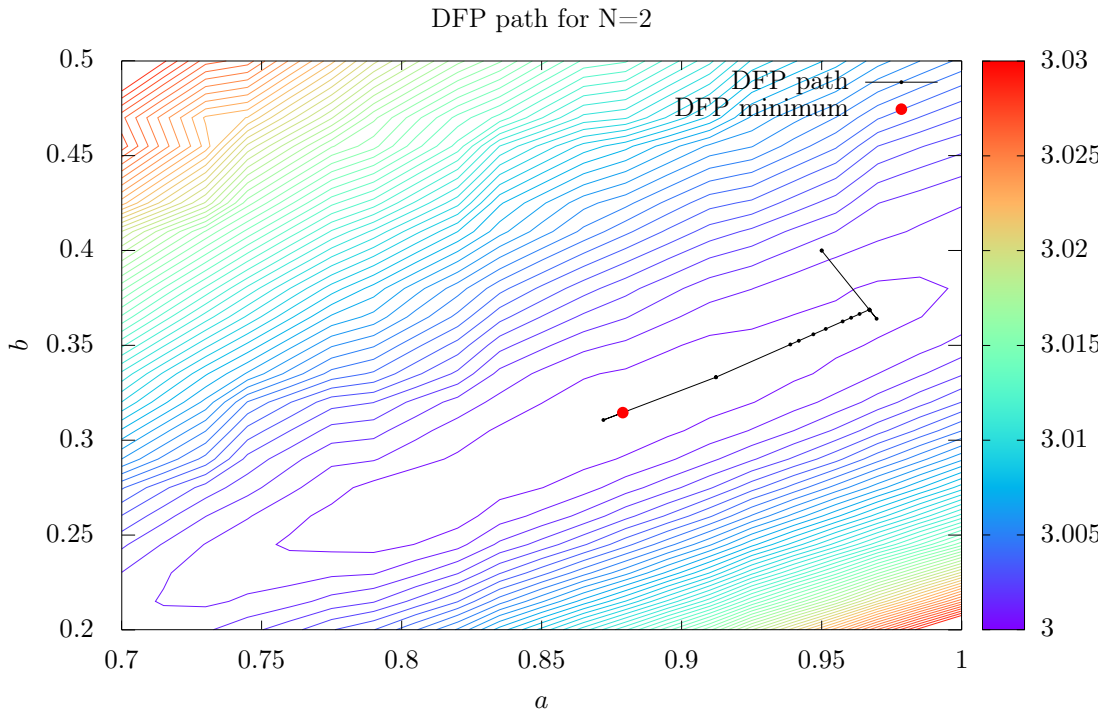

Figure 4.5: Path of the DFP minimization for $N = 2$.

E	E_{VMC}	E_{DMC}
3.00013(2)	3.00029(3)	3.00000(3)
20.2086(1)	20.1899(2)	20.1597(2)

Table 4.4: Comparison of our results (E) with those in Ref[12] (E_{VMC}) and Ref[8] (E_{DMC}) for closed shell Q-dots. In parentheses are the uncertainties in the last digit. All values are for $\omega = 1$.

Chapter 5

Conclusions

In this thesis I developed a `C++` program that deploy the quantum variational Monte Carlo method to calculate the ground states of 2-dimensional circular quantum dots. In particular I considered Q-dots with a number of electrons ranging from 2 to 6 with a parabolic confinement. The program has been improved with several optimized algorithms ot compute acceptance probability and methods like the reweighing method to decrease the overall computation time. These improvements all together allowed the program to have a time scaling with N of $\frac{N^2}{4}$, making it suitable for bigger systems as well. Furthermore I included the Davidon-Fletcher-Powell minimization algorithm that provides a much more intelligent way of finding minima than using variational plots. The use of this method, together with the reweighing methods to compute gradients and derivatives, allowed to find the minimum of the energy in just $5 - 10$ iterations. This was possible also because I already guessed the points using variational plots.

The results I found are in good agreement with those of Refs[8], [12], but I was not able to validate the results for open shell systems. In particular the main weakness of my model was to use only one Slater determinant in the trial wave function. This approximation is good for all the considered configurations, except for $N = 4$, $L = 0$, $S = 0$ because in this case the determinantial part is not an eigenvector of S^2[6].

Another possible cause of errors is to have not considered cusp conditions in our trial wave functions. These are conditions made to the trial wave function in order to prevent possible singularities on the energy when the relative distance becomes zero. For a 2-dimensional system, using the jastrow factor, they lead to set the a parameter to 1 for anti-parallel spins particles and $1/3$ for parallel spins (derivation in Ref[12]). Anyway there are other works where cusp-conditions are not used, for example in Ref[4].

Possible implementation of the code could then be the possibility of using more than one Slater determinant and the cusp conditions on the Jastrow factor. Furthermore one could use a much more general form for the Jastrow factor that might include implicity also multi-particles interaction as the one used in Ref[3]. It would also be interesting to study system in a non-zero magnetic field, with a slightly modified Hamiltonian than the one in Equation 3.2. Finally from the pure computational point of view an improvement could be the support for parallel computing.

Bibliography

[1] L. P. Kouwenhoven, T. H. Oosterkamp, M. W. S. Danoesastro, M. Eto, D. G. Austing, T. Honda, and S. Tarucha, "Excitation spectra of circular, few-electron quantum dots", *Science*, vol. 278, no. 5344, pp. 1788–1792, Dec. 1997. DOI: 10.1126/science.278.5344.1788. [Online]. Available: https://doi.org/10.1126/science.278.5344.1788.

[2] S. Siljamaki, "Wave function methods for quantum dots in magnetic fields", PhD thesis, University of Oslo, 2003. [Online]. Available: https://core.ac.uk/download/pdf/80701116.pdf.

[3] F. Pederiva, C. J. Umrigar, and E. Lipparini, "Diffusion monte carlo study of circular quantum dots", *Physical Review B*, vol. 62, no. 12, pp. 8120–8125, Sep. 2000. DOI: 10.1103/physrevb.62.8120. [Online]. Available: https://doi.org/10.1103/physrevb.62.8120.

[4] A. Harju, V. A. Sverdlov, R. M. Nieminen, and V. Halonen, "Many-body wave function for a quantum dot in a weak magnetic field", *Physical Review B*, vol. 59, no. 8, pp. 5622–5626, Feb. 1999. DOI: 10.1103/physrevb.59.5622. [Online]. Available: https://doi.org/10.1103/physrevb.59.5622.

[5] S. Tarucha, D. G. Austing, T. Honda, R. J. van der Hage, and L. P. Kouwenhoven, "Shell filling and spin effects in a few electron quantum dot", *Physical Review Letters*, vol. 77, no. 17, pp. 3613–3616, Oct. 1996. DOI: 10.1103/physrevlett.77.3613. [Online]. Available: https://doi.org/10.1103/physrevlett.77.3613.

[6] L. Colletti, F. Pederiva, E. Lipparini, and C. Umrigar, "Investigation of excitation energies and hund's rule in open shell quantum dots by diffusion monte carlo", *The European Physical Journal B - Condensed Matter*, vol. 27, no. 3, pp. 385–392, Jun. 2002. DOI: 10.1140/epjb/e2002-00169-x. [Online]. Available: https://doi.org/10.1140/epjb/e2002-00169-x.

[7] A. Harju, "Variational monte carlo for interacting electrons in quantum dots", *Journal of Low Temperature Physics*, vol. 140, no. 3-4, pp. 181–210, Jul. 2005. DOI: 10.1007/s10909-005-6308-7. [Online]. Available: https://doi.org/10.1007/s10909-005-6308-7.

[8] M. P. Lohne, G. Hagen, M. Hjorth-Jensen, S. Kvaal, and F. Pederiva, "Ab initio-computation of the energies of circular quantum dots", *Physical Review B*, vol. 84, no. 11, Sep. 2011. DOI: 10.1103/physrevb.84.115302. [Online]. Available: https://doi.org/10.1103/physrevb.84.115302.

[9] M. H. Kalos, *Monte Carlo Methods*. Wiley-VCH, Oct. 2008, ISBN: 9783527407606. [Online]. Available: https://www.xarg.org/ref/a/352740760X/.

[10] F. Pederiva, A. Roggero, and K. E. Schmidt, "Variational and diffusion monte carlo approaches to the nuclear few- and many-body problem", in *An Advanced Course in Computational Nuclear Physics*, Springer International Publishing, 2017, pp. 401–476. DOI: 10.1007/978-3-319-53336-0_9. [Online]. Available: https://doi.org/10.1007/978-3-319-53336-0_9.

[11] M. P. Lohne, "Coupled-cluster studies of quantum dots", Master's thesis, University of Oslo, 2010. [Online]. Available: https://www.duo.uio.no/bitstream/handle/10852/10966/mplohne.pdf?sequence=2.

[12] L. E. Lervåg, "VMC calculations of two-dimensional quantum dots", Master's thesis, University of Oslo, 2010. [Online]. Available: https://www.uio.no/studier/emner/matnat/fys/FYS4411/v11/undervisningsmateriale/Lecture_notes_and_literature/larseivind_thesis.pdf.

[13] M. Hjorth-Jensen, "Computational physics", 2010, [Online]. Available: http://www.uio.no/studier/emner/matnat/fys/FYS3150/h10/undervisningsmatriale/.

[14] B. L. Hammond, W. A. Lester, and P. J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry*. Nov. 1994, ISBN: 9789810203214.

[15] R. Fletcher, *Practical Methods of Optimization*. Wiley, May 2000, ISBN: 0471494631. [Online]. Available: https://www.xarg.org/ref/a/0471494631/.

[16] C. Griffin, "Numerical optimization", 2012, [Online]. Available: www.personal.psu.edu/cxg286/Math555.pdf.