

FYS4411
SPRING 2016

Variational Monte Carlo studies of electronic systems

John-Anders Stende, Håkon Kristiansen and
Stian Goplen

Date: June 17, 2016

Contents

1	Introduction	2
2	Methods	2
2.1	Monte Carlo integration	2
2.2	Metropolis algorithm	3
2.3	Importance sampling	4
2.4	Steepest descent	5
2.5	Blocking	5
3	Systems	6
3.1	Unperturbed system	6
3.2	Perturbed system	7
3.2.1	Perturbed system for $N = 2$	7
3.2.2	Perturbed system for $N > 2$	8
4	Slater machinery	8
4.1	Optimization of determinant-determinant ratio	8
4.2	Optimization of inverse Slater matrix	9
4.3	Slater determinant and spins	9
4.4	Optimization of the $\nabla D / D $ ratio	10
4.5	Optimization of the $\nabla^2 D / D $ ratio	11
4.6	Optimization of the correlation-to-correlation ratio	11
4.7	Optimization of the $\nabla\Psi_C/\Psi_C$ ratio	12
4.8	Optimization of the $\nabla^2\Psi_C/\Psi_C$ ratio	12
4.9	Steepest descent with Slater wave function	12
5	Implementation and validation	13
5.1	Code structure	13
5.2	Validation	14
5.3	Parallelization	14
6	Results and discussion	15
6.1	Optimal parameters	15
6.2	Ground state energies	15
6.3	One-body densities	16
6.4	Performance analysis	20
7	Conclusions	20
A	Appendix	21
A.1	Analytic local energy for two-body quantum dot	21
A.2	Single-particle wave functions and their derivatives	22
A.3	Hermite polynomials and their derivatives	23

The source code associated with this project is found at the following github addresses:
<https://github.com/hakii27/Project2> and https://github.com/hakii27/Project2_parallel

Abstract

The aim of this project is to use the Variational Monte Carlo (VMC) method to evaluate the ground state energy, onebody densities, expectation values of the kinetic and potential energies and single-particle energies of quantum dots with $N = 2$, $N = 6$, $N = 12$ and $N = 20$ electrons, i.e. closed-shell systems. A performance analysis of the code is also made.

We found that the computed ground state energies and one-body densities were in good agreement with theoretical results and earlier studies. Parallelizing the code resulted in a significant speed-up, enabling us to get better results and error estimates.

1 Introduction

Quantum dots are nano-scale semiconductor devices that contain strongly confined electrons. They exhibit discrete quantum levels due to their small size, including shell structures and magic numbers for the ground states, as in atoms and nuclei. The electronic properties of these materials can be tuned by applying external fields, and are thus of interest in many research applications such as transistors, solar cells, LEDs etc. Studies of quantum dots containing several electrons require reliable many-body methods that also incorporate uncertainty quantifications.

In this project we compute the ground state energy and kinetic and potential energies for $N = 2$, $N = 6$, $N = 12$ and $N = 20$ electrons confined in a two-dimensional harmonic oscillator trap with different oscillator frequencies ω . These values of N are magic numbers for the system. For the two-body case we find the energy both with and without the use of Slater determinants for benchmarking purposes. The one-body density is computed for all N and ω , both with and without correlations in the trial wave function. For $N = 2$ we compute the mean distance between the electrons. We also perform a timing analysis by comparing a serial and parallel version of the code.

The report begins with a presentation of the methods and algorithms we have used in this project. We then present the different systems we want to study, including a detailed description on how to evaluate the trial wave function and its derivatives. Next, we give an overview of the code structure and different ways to validate the implementation. Finally, we present the results of the simulation, and our conclusions with prospects for future work.

2 Methods

We use the *Variational Monte Carlo* (VMC) method in this project to obtain the ground state energy for our fermionic system. VMC applies the *variational principle* from quantum mechanics

$$E_0 \leq \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \quad (1)$$

which states that the ground state energy is always less or equal than the expectation value of the Hamiltonian H for any trial wavefunction Ψ_T . VMC consists in choosing a trial wavefunction depending on one or more variational parameters, and finding the values of these parameters for which the expectation value of the energy is the lowest possible. The main challenge is to compute the multidimensional integral

$$\frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} = \frac{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \boldsymbol{\alpha}) H(\mathbf{R}) \Psi_T(\mathbf{R}, \boldsymbol{\alpha})}{\int d\mathbf{R} \Psi_T^*(\mathbf{R}, \boldsymbol{\alpha}) \Psi_T(\mathbf{R}, \boldsymbol{\alpha})} \quad (2)$$

where \mathbf{R} is the positions of all the particles and $\boldsymbol{\alpha}$ is the set of variational parameters. Traditional integration methods like Gauss-Legendre methods are too computationally expensive, therefore other methods are needed.

2.1 Monte Carlo integration

Monte Carlo integration employs a non-deterministic approach to evaluate multidimensional integrals like (2), or in general

$$I = \int_{\Omega} f(\mathbf{x}) d\mathbf{x} \quad (3)$$

Instead of using an explicit integration scheme, we sample points

$$\mathbf{x}_1 \dots \mathbf{x}_N \in \Omega \quad (4)$$

according to some rule. The most naive approach is to use N uniform samples. The integral can then be approximated as the average of the function values at these points

$$I \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad (5)$$

This simple approach is however not very efficient, as it samples an equal amount of points in all regions of Ω , including those where f is zero.

2.2 Metropolis algorithm

A more clever approach is to sample points according to the probability distribution (PDF) defined by f . Such a PDF is in general difficult to obtain, thus we can't sample directly from it. Instead we use the Metropolis algorithm, which is a method to obtain random samples from a PDF for which direct sampling is difficult. These sample values are produced iteratively, with the distribution of the next sample being dependent only on the current sample value, thus making the sequence of samples into a Markov chain. We define $\mathbf{P}_i^{(n)}$ to be the probability for finding the system in state i at step n . The Metropolis algorithm is as follows:

- Sample a possible new state j with some probability $T_{i \rightarrow j}$
- Accept the new state with probability $A_{i \rightarrow j}$ and use it as the next sample, or reject the new state with probability $1 - A_{i \rightarrow j}$ and use state i as sample again

The transition probability T and the acceptance probability A must fulfill the principle of detailed balance

$$\frac{A_{i \rightarrow j}}{A_{j \rightarrow i}} = \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}} \quad (6)$$

which ensures that $\mathbf{P}_i^{(n \rightarrow \infty)} \rightarrow p_i$, i.e. we end up at the correct distribution regardless of what we begin with.

The particles undergo a random walk under the guidance of the Metropolis algorithm. Defining the PDF

$$P(\mathbf{R}, \alpha) = \frac{|\Psi_T(\mathbf{R}, \alpha)|^2}{\int |\Psi_T(\mathbf{R}, \alpha)|^2 d\mathbf{R}} \quad (7)$$

and the local energy,

$$E_L(\mathbf{R}, \alpha) = \frac{1}{\Psi_T(\mathbf{R}, \alpha)} H \Psi_T(\mathbf{R}, \alpha), \quad (8)$$

the integral (2) can be rewritten as

$$\langle E_L \rangle = \int P(\mathbf{R}, \alpha) E_L(\mathbf{R}, \alpha) d\mathbf{R} \quad (9)$$

and we see that our problem amounts to finding the expectation value of the local energy E_L on the PDF P . Using Monte Carlo integration, we approximate this integral as

$$\langle E_L \rangle \approx \frac{1}{N} \sum_{i=1}^N P(\mathbf{R}_i, \alpha) E_L(\mathbf{R}_i, \alpha) \quad (10)$$

where N is the number of Monte Carlo cycles and \mathbf{R}_i is the position of the particles at step i . The integral $\int |\Psi_T(\mathbf{R}, \alpha)|^2 d\mathbf{R}$ is in general very difficult to compute, but the Metropolis algorithm only needs a *ratio* of probabilities to decide if a move is accepted or not. This can be seen if we rewrite (6) as

$$\frac{p_j}{p_i} = \frac{T_{i \rightarrow j} A_{i \rightarrow j}}{T_{j \rightarrow i} A_{j \rightarrow i}} \quad (11)$$

In our case $p_j = P(\mathbf{R}_j)$ and $p_i = P(\mathbf{R}_i)$. The simplest form of the Metropolis algorithm, called brute force Metropolis, is to assume that the transition probability $T_{i \rightarrow j}$ is symmetric, implying that $T_{i \rightarrow j} = T_{j \rightarrow i}$; the ratio of probabilities (11) thus equals the ratio of acceptance probabilities. This leads to a description of the Metropolis algorithm where we accept or reject a new move by calculating the ratio

$$w = \frac{|\Psi_T(\mathbf{R}_j)|^2}{|\Psi_T(\mathbf{R}_i)|^2} \quad (12)$$

If $w \geq s$, where s is a random number $s \in [0, 1]$, the new position is accepted, else we stay at the same place. We now have the full machinery of the Monte Carlo approach to obtain the ground state energy of our bosonic system:

- Fix the number of Monte Carlo steps and choose the initial positions \mathbf{R} and variational parameters α . Also set the step size $\Delta\mathbf{R}$ to be used when moving from \mathbf{R}_i to \mathbf{R}_j .
- Initialize the local energy
- Choose a random particle
- Calculate a trial position $\mathbf{R}_j = \mathbf{R}_i + r\Delta\mathbf{R}$ where r is a random variable $r \in [0, 1]$
- Use the Metropolis algorithm to accept or reject this move by calculating the ratio (12). If $w \geq s$, where s is a random number $s \in [0, 1]$, the new position is accepted, else we stay at the same place.
- If the step is accepted, set $\mathbf{R} = \mathbf{R}_j$ for the chosen particle
- Sample the local energy

When the Monte Carlo sampling is finished, we calculate the mean local energy, which is our approximation of the ground state energy of the system.

2.3 Importance sampling

A more efficient way to do Monte Carlo sampling is to replace the brute force Metropolis algorithm with a walk in coordinate space biased by the trial wavefunction. This approach is based on the Fokker-Planck equation and the Langevin equation for generating a trajectory in coordinate space.

The Langevin equation is a stochastic differential equation

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \quad (13)$$

where D is the diffusion constant and η a random variable. The new positions y in coordinate space are the solutions of (13) using Euler's method:

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \quad (14)$$

where ξ is a gaussian random variable and Δt is a chosen time step. D is equal to $1/2$ which comes from the factor $1/2$ in the kinetic energy operator. Δt is to be viewed as a parameter which yields stable values of the ground state energy for values $\Delta t \in [0.001, 0.01]$. (14) is similar to the brute force Metropolis equation for updating positions except for the term containing $F(x)$. This is the function that pushes the particles towards regions of configuration space where the wavefunction is large, in contrast to the brute force method where all regions are equally probable. In three dimension $F(x)$ is called the *quantum force* $\mathbf{F}(\mathbf{x})$. The quantum force can be found from the Fokker-Planck equation

$$\frac{\partial P}{\partial t} = \sum_i D \frac{\partial}{\partial \mathbf{x}_i} (\mathbf{x}_i - \mathbf{F}_i) P(\mathbf{x}, t) \quad (15)$$

The convergence to a stationary probability density can be obtained by setting the left hand side to zero. The resulting equation is only satisfied if all terms of the sum are equal to zero,

$$\frac{\partial^2 P}{\partial \mathbf{x}_i^2} = P \frac{\partial}{\partial \mathbf{x}_i} \mathbf{F}_i + \mathbf{F}_i \frac{\partial}{\partial \mathbf{x}_i} P \quad (16)$$

The quantum force should have the form $\mathbf{F} = g(\mathbf{x}) \frac{\partial P}{\partial \mathbf{x}}$. Inserting this in (16) yields

$$\mathbf{F} = 2 \frac{1}{\Psi_T} \nabla \Psi_T \quad (17)$$

The Monte Carlo method with the Metropolis algorithm can be seen as isotropic diffusion process by a time-dependent probability density, with or without a drift, corresponding to brute force and importance sampling respectively. The Fokker-Planck equation (15) describes such a diffusion process, our new transition probability is thus the solution to this equation, given by the Green's function

$$G(y, x, \Delta t) = \frac{1}{(4\pi D \Delta t)^{3N/2}} \exp(-(y - x - D \Delta t F(x))^2 / 4D \Delta t) \quad (18)$$

which in turn means that our brute force Metropolis accept/reject ratio (12) is replaced by the so-called Metropolis-Hastings article

$$q(y, x) = \frac{G(x, y, \Delta t) |\Psi_T(y)|^2}{G(y, x, \Delta t) |\Psi_T(x)|^2} \quad (19)$$

The Metropolis Hastings algorithm is the same as the brute force method, now with (12) replaced by (19) and trial positions calculated according to (14).

2.4 Steepest descent

We seek to minimize expectation value of the local energy with respect to the variational parameters $\alpha = \{\alpha_1, \dots, \alpha_n\}$ numerically. That is to say we want to find α such that,

$$\nabla_{\alpha} \langle E_L(\mathbf{R}; \alpha) \rangle = 0. \quad (20)$$

One way to this is by the method of steepest descent. The rationale behind this approach is the well known fact that if a function $F(\alpha)$ is differentiable in a neighborhood of α , then $F(\alpha)$ decreases fastest if one goes in the direction of $-\nabla_{\alpha} F(\alpha)$. Hence we can generate a new set of parameters α_{k+1} by the iterative scheme,

$$\alpha_{k+1} = \alpha_k - \gamma_i \nabla_{\alpha_k} F(\alpha_k). \quad (21)$$

If γ_i is sufficiently small ¹ we have $F(\alpha_k) \geq F(\alpha_{k+1})$ for all k and hopefully the sequence $\{\alpha_k\}_{k=0}^N$ converges to the desired minimum. The subtraction of $-\gamma_i \nabla_{\alpha_k} F(\alpha_k)$ is due to the fact that we want to move against the gradient towards a local minimum.

In its crudest form γ_i is a constant step size, while it is also possible to adapt γ_i for each iteration to improve upon convergence. We use a very simple adaption scheme for γ_i where we halve the stepsize if $|\nabla_{\alpha_k} F(\alpha_{k+1})| \geq |\nabla_{\alpha_k} F(\alpha_k)|$ and keeps the previous suggestion of α . Note also that we have to supply an initial guess α_0 and naturally the better the initial guess the more we can hope for convergence.

We will deal with two variational parameters α and β in this project. The details of the optimization for our electronic system will be given in section 4.9.

2.5 Blocking

Monte Carlo simulations can be treated as computer experiments. The results can be analyzed with the same statistical tools as we would use analyzing experimental data. We are looking for expectation values of these data, and how accurate they are. A stochastic process like a Monte Carlo experiment produces sequentially a chain of values

$$\{x_1, x_2 \dots x_k \dots x_N\} \quad (22)$$

called a sample. Each value x_k is called a measurement. The sample variance

$$\text{var}(x) = \frac{1}{N} \sum_{k=1}^n (x_k - \bar{x}_N) \quad (23)$$

where \bar{x}_N is the sample mean, is a measure of the statistical error of a *uncorrelated* sample. However, a Monte Carlo simulation with interacting particles produces a correlated sample, thus we need another measure of the sample error. It can be shown that an estimate of the error err_X of a correlated sample is

$$\text{err}_X = \frac{1}{N} \text{cov}(x) \quad (24)$$

where $\text{cov}(x)$ is the sample covariance

$$\text{cov}(x) \equiv \frac{1}{N} \sum_{kl} (x_k - \bar{x}_N)(x_l - \bar{x}_N) \quad (25)$$

which is a measure of the sequential correlation between succeeding measurements of a sample. (Note that (23) and (25) are experimental values for the sample, not the *true* properties of the stochastic variables, which we need an infinite number of measurements to calculate). With the help of the *autocorrelation function* from statistical theory we can rewrite this error as

$$\text{err}_X = \frac{\tau}{N} \text{var}(x) \quad (26)$$

where τ is the *autocorrelation time* which accounts for the correlation between measurements. In the presence of correlation the effective number of measurements becomes

$$n_{\text{eff}} = \frac{N}{\tau} \quad (27)$$

Neglecting τ thus gives an error estimate that is less than the true sample error. The autocorrelation time is however expensive to compute. We can avoid the computation of this quantity by using the technique of blocking. The idea

¹https://en.wikipedia.org/wiki/Gradient_descent

behind this method is to split the sample into blocks, find the mean of each block and then calculate the total mean and variance of all the block means. This is done for increasing block sizes n_b until the measurements of two sequential blocks are uncorrelated, enabling us to extract the value of $\tau = n_b \Delta t$. The true sample error,

$$\sigma = \left(\frac{1 + 2\tau/\Delta t}{N} (\langle E_L^2 \rangle - \langle E_L \rangle^2) \right)^{1/2} \quad (28)$$

can then be calculated.

The blocking algorithm is as follows:

- Do a Monte Carlo simulation, store the local energy for each step to file
- Read the file into an array
- Loop over increasing block sizes:
 - For each block size n_b , loop over array in steps of n_b taking the mean of elements $[in_b, (i+1)n_b], \dots$
 - Calculate total mean and variance of all block means and store
- Plot total variance for all block sizes.
- Extract τ and compute (28)

3 Systems

As mentioned in the introduction we want to study electrons confined in a Harmonic Oscillator (HO) trap in two dimensions. The trap we will use is an isotropic HO potential, with an idealized Hamiltonian given by

$$H = \sum_{i=1}^N \left(-\frac{1}{2} \nabla_i^2 + \frac{1}{2} \omega^2 r_i^2 \right) + \sum_{i < j} \frac{1}{r_{ij}} \quad (29)$$

where natural units ($\hbar = c = e = m_e = 1$) are used and all energies are in atomic units a.u. The Hamiltonian includes a standard HO part

$$H_0 = \sum_{i=1}^N \left(-\frac{1}{2} \nabla_i^2 + \frac{1}{2} \omega^2 r_i^2 \right) \quad (30)$$

and a repulsive interaction between two electrons, given by

$$H_1 = \sum_{i < j} \frac{1}{r_{ij}} \quad (31)$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between the electrons. The modulus of the positions of the electrons is defined as $r_i = \sqrt{r_{ix}^2 + r_{iy}^2}$.

3.1 Unperturbed system

If we only include the HO part of the Hamiltonian we say that the system is unperturbed since there are no interaction between the electrons. The wave function for one electron in an oscillator potential in two dimensions is

$$\phi_{n_x, n_y}(x, y) = A H_{n_x}(\sqrt{\alpha\omega}x) H_{n_y}(\sqrt{\alpha\omega}y) \exp\left(-\frac{\alpha\omega(x^2 + y^2)}{2}\right). \quad (32)$$

The functions $H_{n_x}(\sqrt{\alpha\omega}x)$ are Hermite polynomials (given in appendix A.3) while A is a normalization constant. For the lowest lying state, $n_x = n_y = 0$, the energy is given by $\epsilon_{n_x, n_y} = \omega(n_x + n_y + 1) = \omega$. Now, due the Pauli exclusion principle we can not have two electrons in the same state, however each energy level (distinct (n_x, n_y) pair) can contain two electrons since they can have either spin up or spin down. We can visualize this as follows: If we "fill" each energy level with electrons we say that we have a closed shell system, where the number of particles needed to fill each energy level constitutes a magic number. The first magic numbers are $N = 2, 6, 12, 20$. In the unperturbed case the ground state energy will just be the sum of each individual particle's energy.

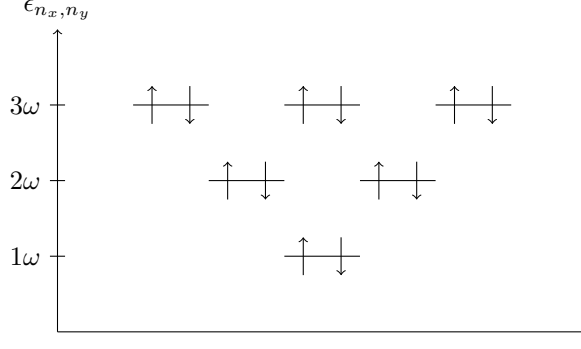


Figure 1: This is a visualization of the shell structure of the many-electron quantum dot. We can see the number of degeneracies of each energy level and that each level can contain one electron with spin up and one with spin down.

Table 1: Exact ground state energies E in atomic units for N electrons in a pure two-dimensional harmonic oscillator with oscillator frequency ω . The values of N are magic numbers for the system.

N	E
2	2ω
6	10ω
12	28ω
20	60ω

This provides an excellent way to check that our program is working properly when we exclude interaction between the electrons.

3.2 Perturbed system

Next we want to describe the perturbed system. We study electrons, which are fermions, thus they obey the Pauli exclusion principle. This means that we have to approximate the exact wave function for the system with a trial wave function that is antisymmetric. Our antisymmetric ansatz Ψ_T is a product of a Slater determinant $\Psi_{SD} = |D|$ and a linear Padé-Jastrow correlation term Ψ_C

$$\Psi_C = \exp\left(\frac{ar_{ij}}{1 + \beta r_{ij}}\right) \quad (33)$$

which accounts for the interaction between the electrons. The full wave function for N electrons can be written

$$\psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) = \Psi_{SD}\Psi_C = |D(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha)| \prod_{i < j}^N \exp\left(\frac{ar_{ij}}{1 + \beta r_{ij}}\right), \quad (34)$$

where a is equal to one when the two electrons i and j have anti-parallel spins and $1/3$ when the spins are parallel. \mathbf{r}_i is the Euclidian position of particle i , while α and β are the variational parameters. The Slater matrix D is defined as

$$D_{ij} = \phi_j(\mathbf{r}_i) \quad (35)$$

where $\phi_j(\mathbf{r}_i)$ are the single-particle wave functions (32), i.e. eigenfunctions of the unperturbed Hamiltonian (30). The rows correspond to the position of a given particle, while the columns stand for the various quantum numbers. The local energy (8) for the perturbed Hamiltonian (29) is

$$E_L = \frac{1}{\Psi_T} H \Psi_T = \sum_{i=1}^N \left(-\frac{1}{2} \frac{\nabla^2 \Psi_T}{\Psi_T} + \frac{1}{2} \omega^2 r_i^2 \right) + \sum_{i < j} \frac{1}{r_{ij}} \quad (36)$$

3.2.1 Perturbed system for $N = 2$

For $N = 2$ the trial wave function reduces to

$$\Psi_T(\mathbf{r}_1, \mathbf{r}_2) = C \exp(-\alpha\omega(r_1^2 + r_2^2)/2) \exp\left(\frac{ar_{12}}{1 + \beta r_{12}}\right). \quad (37)$$

In this case we have analytical expressions for the ground state energy for selected oscillator frequencies which we will use as a way to validate the program. In appendix A.1 we show that the local energy in this case is given by

$$E_L = \frac{1}{\Psi_T} H \Psi_T = 2\alpha^2 \omega^2 (r_1^2 + r_2^2) - 4\alpha\omega - \frac{2a\alpha\omega r_{12}}{(1 + \beta r_{12})^2} + \frac{2a}{(1 + \beta r_{12})^2} \left[\frac{a}{(1 + \beta r_{12})^2} + \frac{1}{r_{12}} - \frac{2\beta}{1 + \beta r_{12}} \right] + \frac{1}{2} \omega^2 (r_1^2 + r_2^2) + \frac{1}{r_{12}} \quad (38)$$

We also need the quantum force when we use importance sampling, derived in appendix A.1:

$$F = -2\alpha\omega \mathbf{r}_1 + \frac{2a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{12} - 2\alpha\omega \mathbf{r}_2 + \frac{2a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{21} \quad (39)$$

3.2.2 Perturbed system for $N > 2$

For $N > 2$ we need to use the full Slater determinant wave function (34). This wave function and its derivatives are challenging to compute. In the following section 4 we will outline the strategies for evaluating these quantities.

4 Slater machinery

The trial wave function (34) plays a central role in our VMC simulation. It is needed in the Metropolis algorithm and in the evaluation of the quantum force (17). Moreover, all observables like the local energy (8) is computed w.r.t. it. The most time-consuming part of the evaluation of the wave function is the computation of the Slater determinant. Computing a determinant of an $N \times N$ matrix by standard Gaussian elimination is of the order of $\mathcal{O}(N^3)$ calculations. As there are $N \cdot d$ independent coordinates we need to evaluate Nd Slater determinants for the gradient (quantum force and kinetic energy) and Nd for the Laplacian (kinetic energy). Therefore, it is imperative to find alternative ways of computing the quantities related to the trial wave function to improve performance.

4.1 Optimization of determinant-determinant ratio

The ratio (12) used in the Metropolis algorithm is the following for our trial wave function (not squared):

$$R = \frac{\Psi_T^{\text{new}}}{\Psi_T^{\text{old}}} = R_{SD} R_C \quad (40)$$

where the Slater determinant-to-determinant ratio is

$$R_{SD} = \frac{|D|^{\text{new}}}{|D|^{\text{old}}} \quad (41)$$

and the correlation-to-correlation is

$$R_C = \frac{\Psi_C^{\text{new}}}{\Psi_C^{\text{old}}} \quad (42)$$

It turns out that we can compute R_{SD} using an algorithm that requires to keep track of the *inverse* of the Slater matrix (35). The inverse of D can be expressed in terms of its cofactors C_{ij} and its determinant $|D|$,

$$D_{ij}^{-1} = \frac{C_{ji}}{|D|} \quad (43)$$

where C_{ji} is the transposed cofactor matrix. The Slater part R_{SD} of the ratio (40) can thus be written,

$$R_{SD} = \frac{|D(\mathbf{R}^{\text{new}})|}{|D(\mathbf{R}^{\text{old}})|} = \frac{\sum_{j=1}^N D_{ij}(\mathbf{R}^{\text{new}}) C_{ij}(\mathbf{R}^{\text{new}})}{\sum_{j=1}^N D_{ij}(\mathbf{R}^{\text{old}}) C_{ij}(\mathbf{R}^{\text{old}})} \quad (44)$$

When moving *one* particle for each Monte Carlo cycle, \mathbf{R}^{new} differs from \mathbf{R}^{old} by the position of only one, say the i -th particle. This means that only the i -th row of $D(\mathbf{R}^{\text{new}})$ and $D(\mathbf{R}^{\text{old}})$ will be different. Taking into account that the i -th row of a cofactor matrix C is independent of the entries of the i -th row of its corresponding matrix D , we have that

$$C_{ij}(\mathbf{R}^{\text{new}}) = C_{ij}(\mathbf{R}^{\text{old}}) \quad j \in \{1, \dots, N\} \quad (45)$$

and

$$R_{SD} = \frac{\sum_{j=1}^N D_{ij}(\mathbf{R}^{\text{new}}) D_{ji}^{-1}(\mathbf{R}^{\text{old}})}{\sum_{j=1}^N D_{ij}(\mathbf{R}^{\text{old}}) D_{ji}^{-1}(\mathbf{R}^{\text{old}})} \quad (46)$$

By definition, the denominator of this expression is unity, thus we obtain for the ratio,

$$R_{SD} = \sum_{j=1}^N D_{ij}(\mathbf{R}^{\text{new}}) D_{ji}^{-1}(\mathbf{R}^{\text{old}}) = \sum_{j=1}^N \phi_j(\mathbf{R}_i^{\text{new}}) D_{ji}^{-1}(\mathbf{R}^{\text{old}}) \quad (47)$$

where the last equality follows from the definition of the Slater matrix (35). This operation is simply a dot product of a vector of single-particle wave functions evaluated at the new position with the i -th column of the inverse matrix D^{-1} evaluated at the original position, and has a time scaling of $\mathcal{O}(N)$. The corresponding expression for R_C is derived in section 4.6

4.2 Optimization of inverse Slater matrix

The operation (47) demands that we maintain the inverse matrix D^{-1} for each MC cycle. Getting the inverse of an $N \times N$ -matrix på Gaussian elimination has a complexicty of order $\mathcal{O}(N^3)$ operations, which we cannot afford. An alternative way of updating the inverse of a matrix when only a row/column is changed was suggested by Sherman and Morris. This algorithm has a time scaling of $\mathcal{O}(N^2)$ and is as follows:

- Update all but the i -th column of D^{-1} . For each column $j \neq i$, calculate the quantity

$$S_j = \sum_{l=1}^N D_{il}(\mathbf{R}^{\text{new}}) D_{lj}^{-1}(\mathbf{R}^{\text{old}})$$

- The new elements of the j -th column of D^{-1} is then given by:

$$D_{kj}^{-1}(\mathbf{R}^{\text{new}}) = D_{kj}^{-1}(\mathbf{R}^{\text{old}}) - \frac{S_j}{R_{SD}} D_{ki}^{-1}(\mathbf{R}^{\text{old}}) \quad k = \{1, \dots, N\}, \quad j \neq i$$

- Finally the i -th column of D^{-1} is updated simply as follows:

$$D_{ki}^{-1}(\mathbf{R}^{\text{new}}) = \frac{1}{R_{SD}} D_{ki}^{-1}(\mathbf{R}^{\text{old}}) \quad k = \{1, \dots, N\}$$

This means that we only need to invert the Slater matrix with conventional methods like Gaussian elimination or LU-decomposition once, after we have initialized them. For the subesequent steps, we use the above algorithm to obtain the inverse matrix.

4.3 Slater determinant and spins

It can be shown, see for example Moskowitz and Kalos [3] that we can approximate the Slater determinant $|D|$ as a product of two smaller ones, where each can be identified with spin-up \uparrow and spin-down \downarrow respectively,

$$|D| = |D|_{\uparrow} \cdot |D|_{\downarrow} \quad (48)$$

To illustrate, we write out the Slater determinant (35) for $N = 4$:

$$|D| = \frac{1}{\sqrt{4!}} \begin{vmatrix} \phi_{00\uparrow}(\mathbf{r}_1) & \phi_{00\downarrow}(\mathbf{r}_1) & \phi_{10\uparrow}(\mathbf{r}_1) & \phi_{10\downarrow}(\mathbf{r}_1) \\ \phi_{00\uparrow}(\mathbf{r}_2) & \phi_{00\downarrow}(\mathbf{r}_2) & \phi_{10\uparrow}(\mathbf{r}_2) & \phi_{10\downarrow}(\mathbf{r}_2) \\ \phi_{00\uparrow}(\mathbf{r}_3) & \phi_{00\downarrow}(\mathbf{r}_3) & \phi_{10\uparrow}(\mathbf{r}_3) & \phi_{10\downarrow}(\mathbf{r}_3) \\ \phi_{00\uparrow}(\mathbf{r}_4) & \phi_{00\downarrow}(\mathbf{r}_4) & \phi_{10\uparrow}(\mathbf{r}_4) & \phi_{10\downarrow}(\mathbf{r}_4) \end{vmatrix} \quad (49)$$

The Slater determinant as written is zero since the spatial wave functions for the spin-up and spin-down states are equal. However, we can now factorize this determinant in the following way,

$$|D| = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_{00\uparrow}(\mathbf{r}_1) & \phi_{10\uparrow}(\mathbf{r}_1) \\ \phi_{00\uparrow}(\mathbf{r}_2) & \phi_{10\uparrow}(\mathbf{r}_2) \end{vmatrix} \cdot \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_{00\downarrow}(\mathbf{r}_3) & \phi_{10\downarrow}(\mathbf{r}_3) \\ \phi_{00\downarrow}(\mathbf{r}_4) & \phi_{10\downarrow}(\mathbf{r}_4) \end{vmatrix} \quad (50)$$

This ansatz is not antisymmetric under exchange of two electrons with opposite spins, but it can be shown that it gives the same expectation value for the energy as the full Slater determinant. The above is correct only for spin-independent Hamiltonians. Our trial wave function can now be written as

$$\Psi_T = |D|_{\uparrow} |D|_{\downarrow} \Psi_C \quad (51)$$

The factorization above makes it possible to perform the calculation of the Slater ratio (47) and the updating of the inverse Slater matrix separately for $|D|_{\uparrow}$ and $|D|_{\downarrow}$:

$$\frac{|D|_{\text{new}}}{|D|_{\text{old}}} = \frac{|D|_{\uparrow}^{\text{new}}}{|D|_{\uparrow}^{\text{old}}} \cdot \frac{|D|_{\downarrow}^{\text{new}}}{|D|_{\downarrow}^{\text{old}}} \quad (52)$$

We see from (50) that one of the two determinants is unaffected when we move only one particle at a time, it will therefore cancel from the ratio. This means that we only need to update one determinant of size $N/2$ when we compute R_{SD} and update $|D|^{-1}$. In our program we therefore operate with two separate matrices D_{\uparrow}^{-1} and D_{\downarrow}^{-1} . There is no need to store D_{\uparrow} and D_{\downarrow} after they have been inverted once because they are explicitly given by the single-particle wave functions ϕ . The efficiency enhancements due to optimization are summed up in Table 2.

Table 2: Comparison of the computational cost involved in the computation of the Slater determinant with and without optimization

Operation	No optimization	With optimization
Evaluation of R_{SD}	$\mathcal{O}(N)$	$\mathcal{O}(N/2)$
Updating inverse	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2/4)$
Transition of one particle	$\mathcal{O}(N) + \mathcal{O}(N^2)$	$\mathcal{O}(N/2) + \mathcal{O}(N^2/4)$

4.4 Optimization of the $\nabla|D|/|D|$ ratio

We need the ratio $\nabla\Psi_T/\Psi_T$ to compute the quantum force (17) used in importance sampling. This ratio can be written as

$$\frac{\nabla\Psi_T}{\Psi_T} = \frac{\nabla(\Psi_{SD}\Psi_C)}{\Psi_{SD}\Psi_C} = \frac{\Psi_C\nabla\Psi_{SD} + \Psi_{SD}\nabla\Psi_C}{\Psi_{SD}\Psi_C} = \frac{\nabla\Psi_{SD}}{\Psi_{SD}} + \frac{\nabla\Psi_C}{\Psi_C} \quad (53)$$

Inserting $\Psi_{SD} = |D|_{\uparrow}|D|_{\downarrow}$ yields

$$\frac{\nabla\Psi_T}{\Psi_T} = \frac{\nabla|D|_{\uparrow}}{|D|_{\uparrow}} + \frac{\nabla|D|_{\downarrow}}{|D|_{\downarrow}} + \frac{\nabla\Psi_C}{\Psi_C} \quad (54)$$

The process to obtain the gradient-determinant ratio needed in this expression is analogous to the one used in deriving (47). The result is

$$\frac{\nabla_i|D|_{\uparrow}}{|D|_{\uparrow}} = \sum_{j=1}^{N/2} \nabla_i D_{\uparrow ij}(\mathbf{R}^{\text{old}}) D_{\uparrow ji}^{-1}(\mathbf{R}^{\text{old}}) = \sum_{j=1}^{N/2} \nabla_i \phi_j(\mathbf{R}_i^{\text{old}}) D_{\uparrow ji}^{-1}(\mathbf{R}^{\text{old}}) \quad (55)$$

which is also valid for $|D|_{\downarrow}$. This expression is used to calculate the quantum force before the ratio (12) is computed. After the position of one particle is altered, the expression is modified to

$$\frac{\nabla_i|D|_{\uparrow}}{|D|_{\uparrow}} = \frac{1}{R_{SD}} \sum_{j=1}^{N/2} \nabla_i \phi_j(\mathbf{R}_i^{\text{new}}) D_{\uparrow ji}^{-1}(\mathbf{R}^{\text{old}}) \quad (56)$$

Note that the old inverse Slater matrix is used in both cases because this is not updated until after a step has been accepted. We see from (54) that we can compute the gradients of the spin-up and spin-down determinants separately. Furthermore, according to the factorization (50), one of the Slater gradients is always zero for a given particle i :

- $\nabla_i|D|_{\downarrow} = 0$ for $i \leq N/2$
- $\nabla_i|D|_{\uparrow} = 0$ for $i > N/2$

thus we only need to compute one of them for each particle. This is implemented as follows:

Listing 1: Computation of the Slater gradient ratio for particle i . The spin-up matrix only contains the single-particle wave functions as functions of the first half of the particles. The spin-down matrix is a function of the second half. One of them is thus zero for a given particle i .

```
// get position of particle i
double x = particles[i]->getPosition()[0];
double y = particles[i]->getPosition()[1];

// spin-up slater
```

```

if (i < m_numberOfParticlesHalf) {
    for (int j=0; j < m_numberOfParticlesHalf; j++) {
        int nx = m_quantumNumbers(j, 0);
        int ny = m_quantumNumbers(j, 1);
        std::vector<double> grad = singleParticleWFGradient(nx, ny, x, y);
        m_gradientUp[0] += grad[0] * m_slaterSpinUpInverse(j, i);
        m_gradientUp[1] += grad[1] * m_slaterSpinUpInverse(j, i);
    }
}
// spin-down slater
else {
    for (int j=0; j < m_numberOfParticlesHalf; j++) {
        int nx = m_quantumNumbers(j, 0);
        int ny = m_quantumNumbers(j, 1);
        std::vector<double> grad = singleParticleWFGradient(nx, ny, x, y);
        m_gradientDown[0] += grad[0] *
        m_slaterSpinDownInverse(j, i-m_numberOfParticlesHalf);
        m_gradientDown[1] += grad[1] *
        m_slaterSpinDownInverse(j, i-m_numberOfParticlesHalf);
    }
}
}

```

The ratio $\nabla\Psi_C/\Psi_C$ is derived in section 4.7.

4.5 Optimization of the $\nabla^2|D|/|D|$ ratio

We need the Laplacian of the trial wave function to compute the expectation value of the kinetic energy K . For electron i this is

$$\langle K_i \rangle = -\frac{1}{2} \frac{\langle \Psi_T | \nabla_i^2 | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \quad (57)$$

It can be shown that the Laplacian of the trial wave function can be written

$$\frac{\nabla^2 \Psi_T}{\Psi_T} = \frac{\nabla^2 |D|_{\uparrow}}{|D|_{\uparrow}} + \frac{\nabla^2 |D|_{\downarrow}}{|D|_{\downarrow}} + \frac{\nabla^2 \Psi_C}{\Psi_C} + 2 \left[\frac{\nabla |D|_{\uparrow}}{|D|_{\uparrow}} + \frac{\nabla |D|_{\downarrow}}{|D|_{\downarrow}} \right] \cdot \frac{\nabla \Psi_C}{\Psi_C} \quad (58)$$

where the Laplace-determinant-to-determinant ratio is given by

$$\frac{\nabla_i^2 |D|_{\uparrow}}{|D|_{\uparrow}} = \sum_{j=1}^{N/2} \nabla_i^2 D_{\uparrow ij}(\mathbf{R}^{\text{new}}) D_{\uparrow ji}^{-1}(\mathbf{R}^{\text{new}}) = \sum_{j=1}^{N/2} \nabla_i^2 \phi_j(\mathbf{R}_i^{\text{new}}) D_{\uparrow ji}^{-1}(\mathbf{R}^{\text{new}}) \quad (59)$$

Again, this also applies to $|D|_{\downarrow}$. This expression must be computed for all electrons to obtain the total Laplacian for the whole system, which is needed to obtain local energy (36). To save computation time, we only recalculate E_L when a step is accepted. The ratio $\nabla^2 \Psi_C / \Psi_C$ is derived in section 4.8.

4.6 Optimization of the correlation-to-correlation ratio

The correlation function Ψ_C (33) can be written

$$\Psi_C = \prod_{i < j}^N \exp(f_{ij}) = \exp(U) \quad (60)$$

where

$$f_{ij} = \frac{a_{ij} r_{ij}}{1 + \beta r_{ij}} \quad (61)$$

and

$$U = \sum_{i < j}^N f_{ij} \quad (62)$$

We see that Ψ_C depends on the relative distances r_{ij} between the electrons. The total number of different relative distances is $N(N-1)/2$, computing the correlation-to-correlation ratio R_C therefore scales as $\mathcal{O}(N^2)$. However, when

moving only one electron at a time, say the k -th electron, only the $N - 1$ distances having k as one of their indicies are changed. The rest of the factors thus cancel, and we have

$$R_C = \frac{\Psi_C^{\text{new}}}{\Psi_C^{\text{old}}} = \frac{\exp(U^{\text{new}})}{\exp(U^{\text{old}})} = \exp(\Delta U) \quad (63)$$

where

$$\Delta U = \sum_{i \neq k}^N (f_{ki}^{\text{new}} - f_{ki}^{\text{old}}) \quad (64)$$

4.7 Optimization of the $\nabla \Psi_C / \Psi_C$ ratio

The expression to be derived in the following is of interest when computing the quantum force (17) and the kinetic energy (57). From the discussion in the last section, only $N - 1$ terms survive when we differentiate w.r.t. particle k . The ratio $\nabla \Psi_C / \Psi_C$ for particle i in one dimension is thus

$$\frac{1}{\Psi_C} \frac{\partial \Psi_C}{\partial x_k} = \frac{1}{\exp U} \frac{\partial \exp U}{\partial x_k} = \frac{\partial U}{\partial x_k} = \sum_{i \neq k}^N \frac{\partial f_{ki}}{\partial x_k} \quad (65)$$

where

$$\frac{\partial f_{ki}}{\partial x_k} = \frac{\partial f_{ki}}{\partial r_{ki}} \frac{\partial r_{ki}}{\partial x_k} = \frac{a_{ki}}{(1 + \beta r_{ki})^2} \frac{x_k - x_i}{r_{ki}} \quad (66)$$

with similar expressions for y_k .

4.8 Optimization of the $\nabla^2 \Psi_C / \Psi_C$ ratio

The following ratio is needed to compute the Laplacian $\nabla^2 \Psi_T / \Psi_T$. For particle k in one dimension this is

$$\frac{1}{\Psi_C} \frac{\partial^2 \Psi_C}{\partial x_k^2} = \frac{1}{\exp U} \frac{\partial}{\partial x_k} \left(\frac{\partial \exp U}{\partial x_k} \right) = \frac{1}{\exp U} \frac{\partial}{\partial x_k} \left(\exp U \frac{\partial \exp U}{\partial x_k} \right) \quad (67)$$

$$= \left(\frac{\partial U}{\partial x_k} \right)^2 + \frac{\partial^2 U}{\partial x_k^2} = \left(\frac{1}{\Psi_C} \frac{\partial \Psi_C}{\partial x_k} \right)^2 + \sum_{i \neq k}^N \frac{\partial^2 f_{ki}}{\partial x_k^2} \quad (68)$$

The total Laplacian for particle k is thus

$$\frac{\nabla_k^2 \Psi_C}{\Psi_C} = \left(\frac{\nabla_k \Psi_C}{\Psi_C} \right)^2 + \sum_{i \neq k}^N \nabla_i^2 f_{ki} \quad (69)$$

where

$$\nabla_i^2 f_{ki} = \frac{1}{r_{ki}} \frac{\partial f_{ki}}{\partial r_{ki}} + \frac{\partial^2 f_{ki}}{\partial r_{ki}^2} \quad (70)$$

and

$$\frac{\partial^2 f_{ki}}{\partial r_{ki}^2} = -\frac{2a_{ki}\beta}{(1 + \beta r_{ki})^3} \quad (71)$$

We must sum over all particles k and both dimensions to obtain the full Laplacian for the system.

4.9 Steepest descent with Slater wave function

To find the optimal variational parameters with the steepest descent method, we need to calculate the quantity $\nabla_{\alpha} \langle E_L(\mathbf{R}, \alpha) \rangle$. We know that $\langle E_L(\mathbf{R}, \alpha) \rangle$ is a multidimensional integral (9), and the gradient w.r.t. α is not easily computed. Let us define

$$\bar{E}_{c_n} = \frac{\partial \langle E_L(\alpha) \rangle}{\partial c_n} \quad (72)$$

where c_n is a variational parameter. Using the chain rule and the hermicity of the Hamiltonian, it can be shown that

$$\bar{E}_{c_n} = 2 \left(\left\langle \frac{\partial \ln \Psi_T}{\partial c_n} E_L \right\rangle - \left\langle \frac{\partial \ln \Psi_T}{\partial c_n} \right\rangle \langle E_L \rangle \right) \quad (73)$$

thus we need the expectation values of

$$\frac{\partial \ln \Psi_T}{\partial c_n} E_L \quad (74)$$

and

$$\frac{\partial \ln \Psi_T}{\partial c_n} \quad (75)$$

For our trial wave function (34), we have

$$\frac{\partial \ln \Psi_T}{\partial c_n} = \frac{\partial \ln |D|_{\uparrow}}{\partial c_n} + \frac{\partial \ln |D|_{\downarrow}}{\partial c_n} + \frac{\partial \ln \Psi_C}{\partial c_n} \quad (76)$$

Inserting our variational parameters α and β , we observe that

$$\frac{\partial \ln \Psi_C}{\partial \alpha} = 0 \quad (77)$$

and

$$\frac{\partial \ln |D|_{\uparrow}}{\partial \beta} = \frac{\partial \ln |D|_{\downarrow}}{\partial \beta} = 0 \quad (78)$$

so we end up with

$$\frac{\partial \ln \Psi_T}{\partial \alpha} = \frac{\partial \ln |D|_{\uparrow}}{\partial \alpha} + \frac{\partial \ln |D|_{\downarrow}}{\partial \alpha} \quad (79)$$

and

$$\frac{\partial \ln \Psi_T}{\partial \beta} = \frac{\partial \ln \Psi_C}{\partial \beta} \quad (80)$$

We can calculate the above derivatives with the help of the following linear algebra identity: If A is an invertible matrix which depends on a real parameter t , and if dA/dt exists, then

$$\frac{d}{dt} |A| = |A| \text{tr} \left(A^{-1} \frac{dA}{dt} \right) \quad (81)$$

thus we have

$$\frac{d}{dt} \ln |A|(t) = \text{tr} \left(A^{-1} \frac{dA}{dt} \right) = \sum_{i=1}^N \sum_{j=1}^N \frac{dA_{ij}}{dt} A_{ji}^{-1} \quad (82)$$

For the Slater determinants, this expression is analogous to the gradient-ratio (55), only now we differentiate the single-particle wave functions w.r.t. α :

$$\frac{\partial \ln |D|}{\partial \alpha} = \sum_{i=1}^{N/2} \sum_{j=1}^{N/2} \frac{\partial \phi_j(\mathbf{R}_i)}{\partial \alpha} D_{ji}^{-1}(\mathbf{R}_i) \quad (83)$$

The corresponding expression for the correlation function is

$$\frac{\partial \ln \Psi_C}{\partial \beta} = \sum_{i < j}^N \frac{\partial}{\partial \beta} \left(\frac{a_{ij} r_{ij}}{1 + \beta r_{ij}} \right) = - \sum_{i < j}^N \frac{a_{ij} r_{ij}^2}{(1 + \beta r_{ij})^2} \quad (84)$$

5 Implementation and validation

5.1 Code structure

In Project1 (insert ref??) we made an object-oriented code in C++ that could compute the ground-state energies for bosonic systems. We give here an overview of the class structure and what the different classes do,

- *Main program*: Sets all the parameters needed to a simulation.
- *System*: Runs the Monte Carlo cycles with/without importance sampling
- *Sampler*: Samples quantities we want to measure for each cycle and computes expectation values
- *Particle*: Sets and adjusts particle positions
- *SteepestDescent*: Runs the steepest descent method
- *InitialState*: Super-class for setting up different initial states
 - *RandomUniform*: Assigns initial positions according to a uniform distribution

- *WaveFunction*: Super-class for different wave functions. Sets the variational parameters. All subclasses must implement functions to evaluate the analytical expressions for Ψ_T , $\nabla\Psi_T$, $\nabla^2\Psi_T$ and $d\Psi_T/d\alpha$.
- *Hamiltonian*: Super-class for different Hamiltonians. Calculates E_L by computing potential and kinetic energy, either numerically or analytically for the latter. The analytical Laplacian is obtained from *WaveFunction*.

In addition to these we use class *Random* to generate pseudo-random numbers.

All the information are stored in *System* in the form of class objects of the other classes, which in turn receives the *System* object so that they can access this information via setters and getters in the *System* class. This way of communicating between classes limits the user's capability to alter vital functionality and also makes the program more user-friendly. Object-oriented code is also easy to expand on. We don't need to add new functionality to e.g. implement a new wave function; only the specifics of this new wave function needs to be implemented.

In this project we mainly expand on the previous program. For the two-electron quantum dot we only need to add a new wavefunction sub-class with needed functionality, such as evaluation of the wavefunction its derivatives with respect to the variational parameters and its gradient. Also we need to add a two-electron quantum dot hamiltonian which computes the kinetic, potential and local energy. For the local energy we have derived a analytic expression (see Appendix).

For the quantum dot with more than two electrons we have to create a new wavefunction we have called *SlaterWavefunction* which implements the expressions for the different ratios described in the above section in order to run the metropolis algorithm. We have also added a new hamiltonian class for this case. Note also that we have modify/create a new method for running the metropolis algorithm in *System*. The most important alteration to this method is that we have to call on *SlaterWavefunction's updateInverse* method if a new position is accepted.

5.2 Validation

With large programs it is important to be able to validate the code as we add new layers to the program. The first step is to check that if we have no interaction, no Jastrow-factor and $\alpha = 1$ we get the energies in Table 1. This test is passed for all N and ω with variance exactly zero using the *SlaterWavefunction*. For the two-electron case without a Slater-Wavefunction this test passes all ω with zero variance.

Next we want to benchmark the program with previous studies. For the two-electron case we compare the energies with analytic expressions found by Taut(Insert ref). In this case we do not use the Slater-Machinery. Another test we make is that if we use the Slater-wavefunction with $N = 2$ we get approximately the same energies as if we do not use the Slater-wavefunction. At last we study the $N = \{6, 12, 20\}$ cases and compare the estimates of the ground-state energies with results from previous studies.

5.3 Parallelization

We have parallelized the program using MPI. The way we have done this is that we initialize MPI and let each process compute all quantities such as the expectation value of the ground-state energy, kinetic and potential energy and variance of a system. At last we use MPI_REDUCE to collect all data from each process by letting Process1 be the "master" and then compute the average of the collected data. Note that when the problem is this easy to parallelize there is close to no communication between the processes, since the only communication needed is when we collect the data from each process so we expect to close 100% speed-up. Under results and discussion we present a performance analysis where we compare the time of a serial version of the program with the time of the parallel version.

6 Results and discussion

6.1 Optimal parameters

Table 3: Optimal α and β values for given ω 's with jastrow factor and Slater determinant. In the last column we have included the optimal α 's we found when we had switched off the jastrow factor.

N	ω	α	β	$\alpha_{\text{wo-jastrow}}$
2	1	0.992067	0.400016	0.686717
	0.5	0.952981	0.354743	0.669455
	0.1	0.952833	0.354292	0.655921
	0.05	0.913976	0.235196	0.526488
	0.01	0.911692	0.203919	0.477602
6	1	1.03741	0.472513	0.599828
	0.5	0.931202	0.395044	0.599619
	0.1	0.831104	0.211443	0.599619
	0.05	0.849742	0.155566	0.599616
	0.01	0.842683	0.10789	0.599616
12	1	1.10364	0.468861	0.569619
	0.5	0.936306	0.414534	0.488599
	0.1	0.84105	0.208143	0.354431
	0.05	0.842746	0.153921	0.277028
	0.01	0.835467	0.0923274	0.220951
20	1	1.06019	0.474467	0.5137
	0.5	0.944121	0.419367	0.4359
	0.1	0.856981	0.200372	0.2854
	0.05	0.842341	0.153017	0.2260
	0.01	0.835301	0.0922157	0.1316

6.2 Ground state energies

Table 4: In the following table we have computed the ground-state energy of all the described systems. Compared with previous studies we have excellent agreement. Especially it is important to note that for $N = 2$ and $\omega = 1$ we have found an energy which is close to the analytic energy found by Taut[1]. The given variance is obtained after blocking has been used. All computations have been done with importance sampling and $N_{\text{cycles}} = 4 \cdot 10^8$ on four processors. We have used the optimal parameters listed above. We can see that if we do not include the Jastrow factor we get systematically higher energies than if the Jastrow factor is included. Note also that the variance increases with number of particles, but decreases with respect to decreasing oscillator frequency. (a)[4], (b)[2].

N	ω	$\langle E \rangle$	$\langle E_{\text{wo-jastrow}} \rangle$	$\langle E_{\text{ref}}^{(a)} \rangle$	$\langle E_{\text{ref}}^{(b)} \rangle$	σ^2	$\langle K \rangle$	$\langle V \rangle$	Acceptance	$\langle r_{12} \rangle$
2	1	3.00055	3.11315	3.00030	3.00000	3.0478e-08	0.879709	2.12084	0.993751	1.63513
	0.5	1.66096	2.0897	1.66021	1.65975	2.22567e-08	0.433121	1.22784	0.995564	2.45743
	0.1	0.459013	0.531735	0.44130	-	7.33253e-08	0.0975641	0.361449	0.997945	5.43922
	0.05	0.261726	0.318454	-	-	1.62292e-08	0.0478937	0.213833	0.998530	8.68116
	0.01	0.0881618	0.115187	0.07406	-	8.97366e-09	0.010319	0.0778428	0.999322	18.5670
6	1	20.3143	20.835	20.1902	20.1597	3.82438e-07	5.53228	14.782	0.973272	
	0.5	11.7369	12.828	11.8103	11.7888	5.3671e-08	2.58952	9.14733	0.942306	
	0.1	3.53863	4.60462	3.5690	-	1.0771e-08	0.512675	3.02596	0.974318	
	0.05	2.16528	3.08048	-	-	2.79175e-09	0.274734	1.89055	0.940855	
	0.01	0.730792	1.25365	-	-	2.38439e-09	0.0593103	0.671482	0.972479	
12	1	66.3061	67.2075	65.7905	65.700	1.67436e-06	13.2386	53.0675	0.970871	
	0.5	39.0934	40.4228	39.2343	39.159	6.35675e-07	6.00468	33.0887	0.980363	
	0.1	12.3499	13.2956	12.3162	-	1.61496e-07	1.27686	11.073	0.990964	
	0.05	7.64362	8.21335	-	-	8.23388e-08	0.669975	6.97364	0.993451	
	0.01	2.58158	2.97721	-	-	2.11037e-08	0.140185	2.4414	0.997012	
20	1	156.139	157.178	156.062	155.868	8.43247e-06	23.5598	132.579	0.990506	
	0.5	93.812	95.2564	94.0236	93.867	2.12929e-06	11.2513	82.5607	0.993433	
	0.1	30.5009	31.0954	30.0729	-	4.91884e-07	2.61678	27.8842	0.996848	
	0.05	18.8873	19.3607	-	-	2.38327e-07	1.30327	17.5841	0.997767	
	0.01	6.40633	6.52191	-	-	1.79865e-07	0.264965	6.14137	0.998983	

6.3 One-body densities

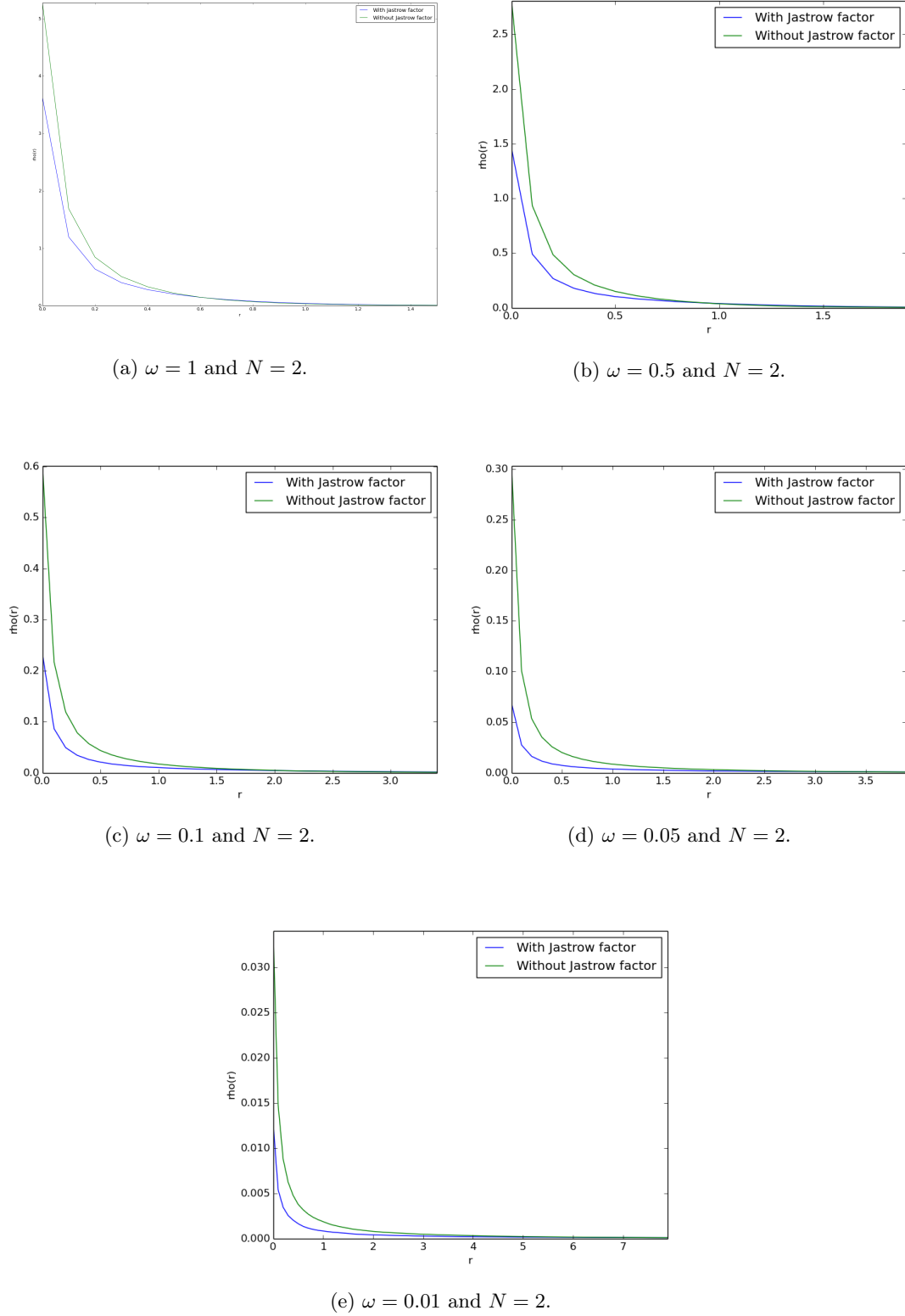
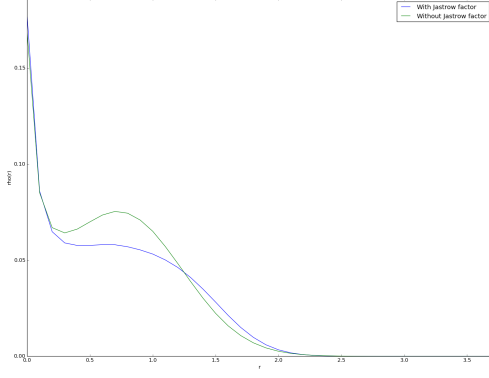
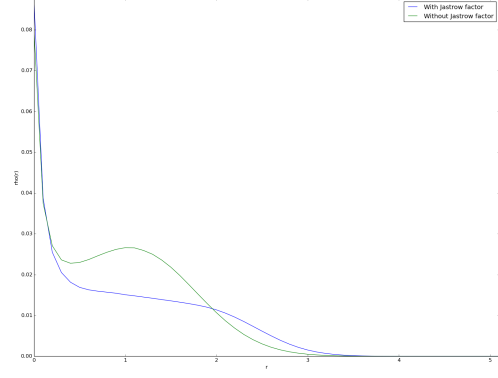


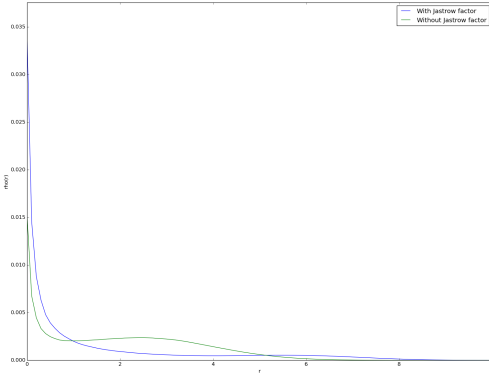
Figure 2: We have not used the Slater-wavefunction when we have computed the densities for $N = 2$. Importance sampling have been used for all frequencies and the number of cycles $N_{\text{cycles}} = 4 \cdot 10^8$ with $P = 4$ and timestep, $\Delta t = 0.0001$. We have used the optimal parameters we found in Table3 with and without Jastrow factor. For the first closed shell, $N = 2$, the OBD takes on a well behaved slope with higher values closer to the center. For a strong oscillator potential, ω , the interaction between the particles itself will be of less importance and they will be found closer to each other, and closer to the center. As the ω decreases the interaction between the particles will have a bigger impact and push the particles further out. This is expected due to the fact that the electrons are less confined by the potential well.



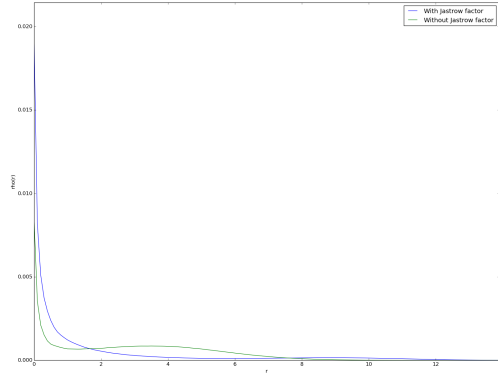
(a) $\omega = 1$ and $N = 6$ with Slater determinant.



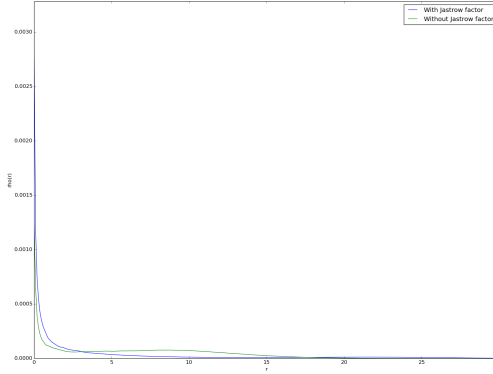
(b) $\omega = 0.5$ and $N = 6$ with Slater determinant.



(c) $\omega = 0.1$ and $N = 6$ with Slater determinant.

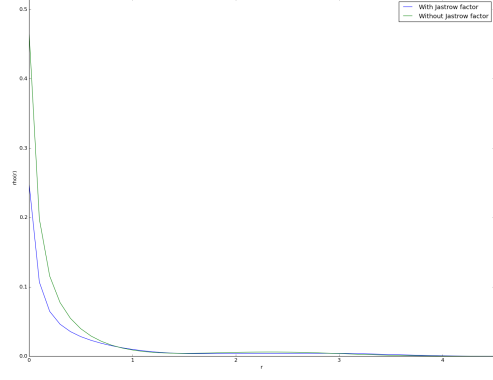
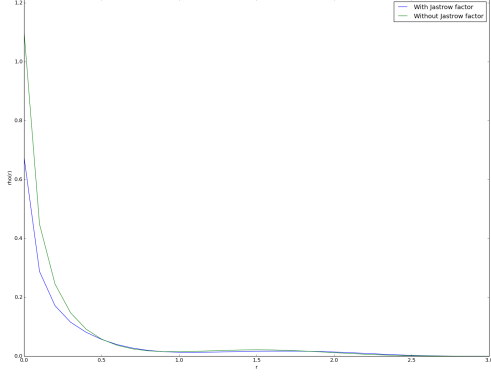


(d) $\omega = 0.05$ and $N = 6$ with Slater determinant.

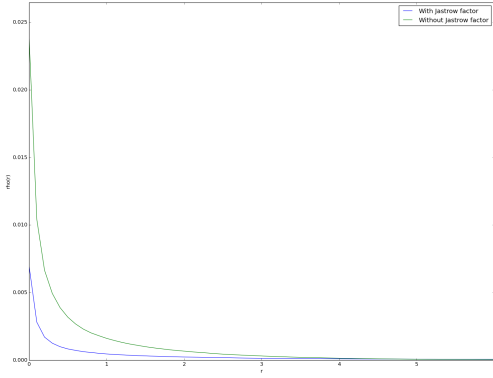
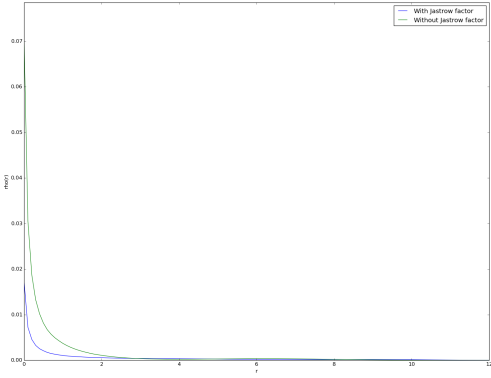


(e) $\omega = 0.01$ and $N = 6$ with Slater determinant.

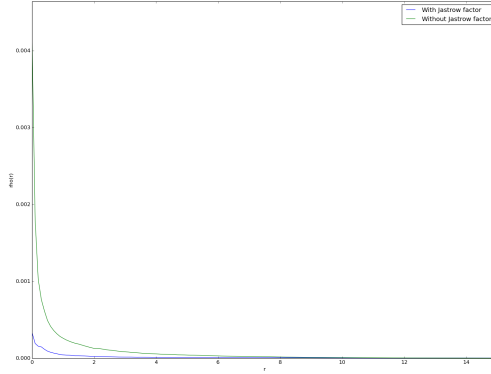
Figure 3: Importance sampling have been used for all frequencies and the number of cycles $N_{\text{cycles}} = 4 \cdot 10^8$ with $P = 4$ and timestep, $\Delta t = 0.0001$. We have used the optimal parameters we found in Table3 with and without Jastrow factor. For the second closed shell, $N = 6$, the slope takes on a more unorthodox shape. For a higher ω this is more apparent because the density is more compressed in the x-direction. We can see that this effect is greater with the Jastrow factor. In other words, the Jastrow factor makes the interaction less important and the shape is more compressed than without the Jastrow factor. If we compare with the case with $N = 2$ we can see that the density is higher further from the center. This is due to the fact that with more particles in the system there one particle interacts with more particles and it will be pushed further out. Note also that density falls off and then reaches a new peak before it finally goes to zero.



(a) $\omega = 1$ and $N = 12$ with Slater determinant. (b) $\omega = 0.5$ and $N = 12$ with Slater determinant.

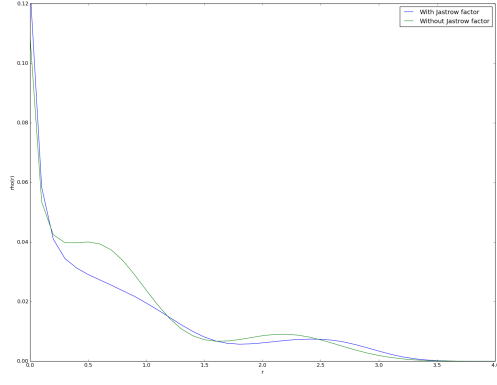


(c) $\omega = 0.1$ and $N = 12$ with Slater determinant. (d) $\omega = 0.05$ and $N = 12$ with Slater determinant.

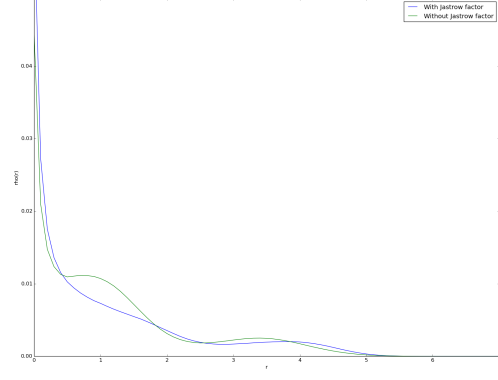


(e) $\omega = 0.01$ and $N = 12$ with Slater determinant.

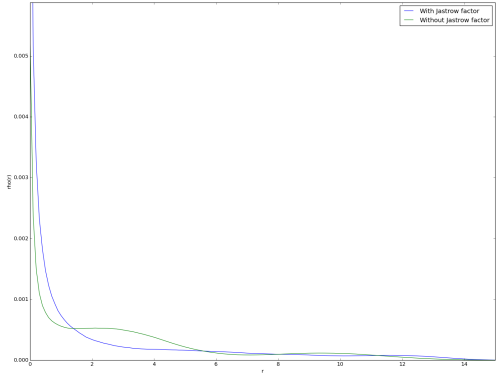
Figure 4: Importance sampling have been used for all frequencies and the number of cycles $N_{\text{cycles}} = 4 \cdot 10^8$ with $P = 4$ and timestep, $\Delta t = 0.0001$. We have used the optimal parameters we found in Table3 with and without Jastrow factor. For the third closed shell, $N = 12$ we can see the same features as for the previous shells as ω decreases. However, compared with $N = 6$ we do not see the same distinct peak in the density. Nevertheless, we can see that there is a slight increase in density but its further from the center than for $N = 6$. This is due to the fact that the energy is higher in this system.



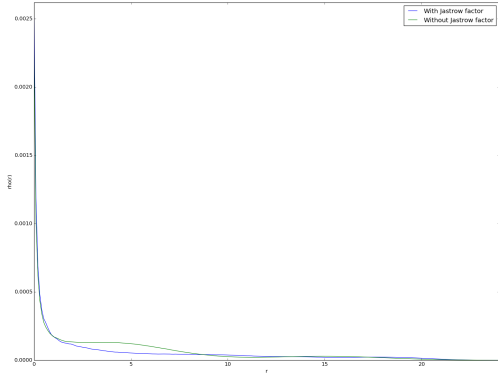
(a) $\omega = 1$ and $N = 20$ with Slater determinant.



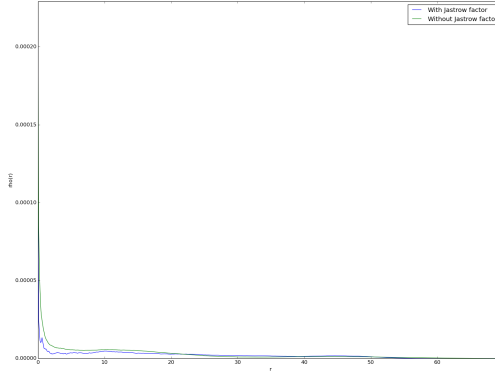
(b) $\omega = 0.5$ and $N = 20$ with Slater determinant.



(c) $\omega = 0.1$ and $N = 20$ with Slater determinant.



(d) $\omega = 0.05$ and $N = 20$ with Slater determinant.



(e) $\omega = 0.01$ and $N = 20$ with Slater determinant.

Figure 5: Importance sampling have been used for all frequencies and the number of cycles $N_{\text{cycles}} = 4 \cdot 10^8$ with $P = 4$ and timestep, $\Delta t = 0.0001$. We have used the optimal parameters we found in Table3 with and without Jastrow factor. Now we can see that we have two distinct peaks in the densities. It is interesting to note that we get a new peak in this case. It seems like that for every even number of energy levels we get a new peak, while for every other odd number of energy levels we have the same number of peaks as the previous shell, but its less distinct and further from the center. This in accordance with the shape of the probability distribution of one electron in a harmonic oscillator potential.

6.4 Performance analysis

In this section we present a performance analysis of the program. We look at the time usage of the serial version for $N = \{6, 12, 20\}$ with $N_{steps} = \{10^6, 10^7\}$ and then repeat these timings for the parallelized version of the program in order to check if we have a close to 100% speed-up. All timings are done with all filewriting turned off and with optimal parameters, oscillator frequency $\omega = 1$ and with Jastrow factor. Filewriting and the samples needed for steepest descent are switched off. The quantities which we compute while doing the analysis is the expectation value of the ground-state, kinetic and potential energy and the variance. We also assume that the initialization time of MPI is negligible compared to the time it takes to run all Monte Carlo cycles. Hence, we have clocked the time the MPI program takes to complete the Metropolis algorithm. Both the serial and parallel version have been run on the machines in the computational physics computer lab.

Since these machines have four cores we have used four processors when running the parallel program. This means that when the parallel program is run we simulate four systems of the same size as one in the serial version. Speed-up, S is then measured as

$$S = \frac{t_s}{t_p}$$

where t_s is the time of the serial program and t_p is the time of the parallel program.

Table 5: In the following we present the average run time of the serial program and the parallelized version using MPI with $N = \{6, 12, 20\}$ and $N_{steps} = \{10^6, 10^7\}$. The cpu-time estimates is an average over ten measurements. The timings have been repeated in order to eliminate cold start effects. We also did not have any other programs running in the background to avoid interference. When these measurements were made only small deviations in the time measurements was observed indicating that we managed to avoid these problems. As such, ten measurements seems reasonable in order to give a proper average. Note that when we increase the number of cycles with a factor of ten, the cpu-time also increases with a factor ten. This feature is consistent for all number of particles and if we use the serial or parallel version. When we increase the number of particles from six to twelve there is approximately a factor of three increase in run-time for both the serial and parallel program. An increase from twelve to twenty particles gives an increase in run-time of approximately a factor of two and a half in both cases. In the case of $N = 6$ we have 75% speed-up for 10^6 MC-cycles and 80% speed-up for 10^7 MC-cycles. For $N = 12$ there is 80% speed-up for both 10^6 and 10^7 MC-cycles while for $N = 20$ there is 84% speed-up in both cases. This indicates that there is a cost of running with MPI in itself. We note that the speed-up is larger as we have more MC-cycles and more particles which indicates that the overhead induced from having MPI running is negligible if we run for many MC-cycles and larger number of particles provided that there is close to no communication between the processes which is the case in this particular problem.

N	$\langle \text{Cpu} \rangle_{10^6} [s]$	$\langle \text{Cpu} \rangle_{10^7} [s]$	$\langle \text{Cpu} \rangle_{10^6}^{\text{MPI}} [s]$	$\langle \text{Cpu} \rangle_{10^7}^{\text{MPI}} [s]$
6	4.21	42.19	5.54	52.19
12	13.06	130.86	16.83	160.76
20	33.19	334.04	39.84	399.87

7 Conclusions

In this project we have written a VMC code that computes the ground-state energy for fermionic systems. In particular we have studied electrons confined in an isotropic harmonic oscillator potential. We have computed the ground state energy for $N = 2, 6, 12, 20$ electrons and found them to be in excellent agreement with referenced results. The program has been validated against a simplified case with exact accordance with analytic results. We have computed the onebody density for all systems that we have studied. Finally we have given a short performance analysis where we compare the efficiency of a serial version of the program with a parallel version that used MPI and found that we get a significant speed-up with the parallel program. Further we want to remark that the steepest descent method have been challenging to work with and it would have been useful to study other optimization algorithms such as the conjugate gradient method. Nevertheless, it seems that we have succeeded in finding quite good optimal parameters due the fine agreement with previous studies for the ground state energy.

We have found it rewarding to work with this project. It has been especially challenging to manage all the data since we have studied so many systems with different oscillator frequencies. In the end we feel that we managed to present a vast amount of results. However we feel that we could have discussed the results more thoroughly if we had been able to manage the data more efficiently. Even though, we feel content and pleased with the final result of this project.

A Appendix

A.1 Analytic local energy for two-body quantum dot

$$\Psi_T(\mathbf{r}_1, \mathbf{r}_2) = \exp(-\alpha\omega(r_1^2 + r_2^2)/2) \exp\left(\frac{ar_{12}}{1 + \beta r_{12}}\right) \quad (85)$$

$$= K_1 K_2 \quad (86)$$

Laplacian of this wave function:

$$\nabla^2 \Psi_T(\mathbf{r}_1, \mathbf{r}_2) = \frac{\partial \Psi_T}{\partial x_1^2} + \frac{\partial \Psi_T}{\partial y_1^2} + \frac{\partial \Psi_T}{\partial x_2^2} + \frac{\partial \Psi_T}{\partial y_2^2} \quad (87)$$

Using the following

$$\frac{\partial r_1}{\partial x_1} = x_1/r_1 \quad (88)$$

and

$$\frac{\partial r_{12}}{\partial x_1} = (x_1 - x_2)/r_{12} \quad (89)$$

Gradient of first term:

$$\frac{\partial K_1}{\partial x_1} = -\alpha\omega x_1 K_1 \quad (90)$$

so that

$$\nabla_1 K_1 = -\alpha\omega K_1 \mathbf{r}_1 \quad (91)$$

and

$$\nabla_2 K_1 = -\alpha\omega K_1 \mathbf{r}_2 \quad (92)$$

where

$$\mathbf{r}_i = (x_i, y_i) \quad (93)$$

Gradient of second term:

$$\frac{\partial K_2}{\partial x_1} = K_2 \frac{a(x_1 - x_2)}{r_{12}(1 + \beta r_{12})^2} \quad (94)$$

so that

$$\nabla_1 K_2 = K_2 \frac{a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{12} \quad (95)$$

and

$$\nabla_2 K_2 = K_2 \frac{a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{21} \quad (96)$$

where

$$\mathbf{r}_{ij} = (x_i - x_j, y_i - y_j) \quad (97)$$

The quantum force used in importance sampling is thus

$$\begin{aligned} F = 2 \frac{\nabla \Psi_T}{\Psi_T} &= 2 \frac{\nabla(K_1 K_2)}{K_1 K_2} = 2 \frac{1}{K_1 K_2} (K_2 \nabla K_1 + K_1 \nabla K_2) = \\ &= -2\alpha\omega \mathbf{r}_1 + \frac{2a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{12} - 2\alpha\omega \mathbf{r}_2 + \frac{2a}{r_{12}(1 + \beta r_{12})^2} \mathbf{r}_{21} \end{aligned} \quad (98)$$

Laplacian of first term:

$$\frac{\partial^2 K_1}{\partial x_1^2} = K_1 (\alpha^2 \omega^2 x_1^2 - \alpha\omega) \quad (99)$$

so that

$$\nabla^2 K_1 = K_1 (\alpha^2 \omega^2 (r_1^2 + r_2^2) - 4\alpha\omega) \quad (100)$$

Laplacian of second term:

$$\begin{aligned} \frac{\partial^2 K_2}{\partial x_1^2} &= K_2 \left[\frac{a^2(x_1 - x_2)^2}{r_{12}^2(1 + \beta r_{12})^4} + \frac{ar_{12}(1 + \beta r_{12})^2}{r_{12}^2(1 + \beta r_{12})^4} \right. \\ &\quad \left. - \frac{a(x_1 - x_2)[(x_1 - x_2)/r_{12}(1 + \beta r_{12})^2 + 2r_{12}(1 + \beta r_{12})\beta(x_1 - x_2)/r_{12}]}{r_{12}^2(1 + \beta r_{12})^4} \right] \end{aligned} \quad (101)$$

thus

$$\frac{\partial^2 K_2}{\partial x_1^2} = K_2 \left[\frac{a^2(x_1 - x_2)^2}{r_{12}^2(1 + \beta r_{12})^4} + \frac{a}{r_{12}(1 + \beta r_{12})^2} - \frac{a(x_1 - x_2)^2}{r_{12}^3(1 + \beta r_{12})^2} - \frac{2a\beta(x_1 - x_2)^2}{r_{12}^2(1 + \beta r_{12})^3} \right] \quad (102)$$

so that

$$\begin{aligned} \nabla^2 K_2 &= K_2 \left[\frac{2a^2}{(1 + \beta r_{12})^4} + \frac{4a}{r_{12}(1 + \beta r_{12})^2} - \frac{2a}{r_{12}(1 + \beta r_{12})^2} - \frac{2a\beta}{(1 + \beta r_{12})^3} \right] \\ &= K_2 \frac{2a}{(1 + \beta r_{12})^2} \left[\frac{a}{(1 + \beta r_{12})^2} + \frac{1}{r_{12}} - \frac{2\beta}{1 + \beta r_{12}} \right] \end{aligned} \quad (103)$$

Have that

$$\nabla^2 \Psi_T = \nabla^2 K_1 K_2 + 2\nabla K_1 \nabla K_2 + K_1 \nabla^2 K_2 \quad (104)$$

and

$$\begin{aligned} \nabla K_1 \nabla K_2 &= -K_1 K_2 \frac{a\alpha\omega}{r_{12}(1 + \beta r_{12})^2} \left[x_1(x_1 - x_2) + y_1(y_1 - y_2) - x_2(x_1 - x_2) - y_2(y_1 - y_2) \right] \\ &= -K_1 K_2 \frac{a\alpha\omega}{r_{12}(1 + \beta r_{12})^2} \left[(x_1 - x_2)(x_1 - x_2) + (y_1 - y_2)(y_1 - y_2) \right] \\ &= -K_1 K_2 \frac{a\alpha\omega r_{12}}{(1 + \beta r_{12})^2} \end{aligned} \quad (105)$$

The analytic expression for the Laplacian ratio is thus

$$\begin{aligned} \frac{\nabla^2 \Psi_T}{\Psi_T} &= 2\alpha^2 \omega^2 (r_1^2 + r_2^2) - 4\alpha\omega - \frac{2a\alpha\omega r_{12}}{(1 + \beta r_{12})^2} + \\ &\quad \frac{2a}{(1 + \beta r_{12})^2} \left[\frac{a}{(1 + \beta r_{12})^2} + \frac{1}{r_{12}} - \frac{2\beta}{1 + \beta r_{12}} \right] \end{aligned} \quad (106)$$

and we get for the local energy in the Hamiltonian (29)

$$\begin{aligned} E_L = \frac{1}{\Psi_T} H \Psi_T &= 2\alpha^2 \omega^2 (r_1^2 + r_2^2) - 4\alpha\omega - \frac{2a\alpha\omega r_{12}}{(1 + \beta r_{12})^2} + \\ &\quad \frac{2a}{(1 + \beta r_{12})^2} \left[\frac{a}{(1 + \beta r_{12})^2} + \frac{1}{r_{12}} - \frac{2\beta}{1 + \beta r_{12}} \right] + \\ &\quad \frac{1}{2} \omega^2 (r_1^2 + r_2^2) + \frac{1}{r_{12}} \end{aligned} \quad (107)$$

A.2 Single-particle wave functions and their derivatives

The Slater matrix is a function of the single-particle wave functions

$$\phi_{n_x, n_y}(x, y) = A H_{n_x}(\sqrt{\alpha\omega}x) H_{n_y}(\sqrt{\alpha\omega}y) \exp(-\alpha\omega(x^2 + y^2)/2). \quad (108)$$

We need the gradient and Laplacian of these functions to compute $\nabla|D|/|D|$ and $\nabla^2|D|/|D|$. The two components of the gradient w.r.t to particle i is

$$\frac{\phi_{n_x, n_y}(x_i, y_i)}{\partial x_i} = \exp[-\alpha\omega(x_i^2 + y_i^2)/2] H_{n_y}(\sqrt{\alpha\omega}y_i) \left(\frac{\partial H_{n_x}(\sqrt{\alpha\omega}x_i)}{\partial x_i} - H_{n_x}(\sqrt{\alpha\omega}x_i) \alpha\omega x_i \right) \quad (109)$$

and

$$\frac{\phi_{n_x, n_y}(x_i, y_i)}{\partial y_i} = \exp[-\alpha\omega(x_i^2 + y_i^2)/2] H_{n_x}(\sqrt{\alpha\omega}x_i) \left(\frac{\partial H_{n_y}(\sqrt{\alpha\omega}y_i)}{\partial y_i} - H_{n_y}(\sqrt{\alpha\omega}y_i) \alpha\omega y_i \right) \quad (110)$$

and the Laplacian is given as

$$\begin{aligned} \nabla^2 \phi_{n_x, n_y}(x_i, y_i) &= \exp[-\alpha\omega(x_i^2 + y_i^2)/2] \\ &\quad \left(H_{n_y}(\sqrt{\alpha\omega}y_i) \frac{\partial^2 H_{n_x}(\sqrt{\alpha\omega}x_i)}{\partial x_i^2} + H_{n_x}(\sqrt{\alpha\omega}x_i) \frac{\partial^2 H_{n_y}(\sqrt{\alpha\omega}y_i)}{\partial y_i^2} + \right. \\ &\quad \left. \alpha\omega H_{n_x} H_{n_y} (\alpha\omega(x_i^2 + y_i^2) - 2) - \right. \\ &\quad \left. 2\alpha\omega x_i H_{n_y} \frac{\partial H_{n_x}(\sqrt{\alpha\omega}x_i)}{\partial x_i} - 2\alpha\omega y_i H_{n_x} \frac{\partial H_{n_y}(\sqrt{\alpha\omega}y_i)}{\partial y_i} \right) \end{aligned} \quad (111)$$

We also need the derivatives w.r.t. the variational parameters for optimizing:

$$\begin{aligned} \frac{\partial \phi_{n_x, n_y}(x_i, y_i)}{\partial \alpha} = & \exp[-\alpha\omega(x_i^2 + y_i^2)/2] \\ & \left(H_{n_x}(\sqrt{\alpha\omega}x_i) \frac{\partial H_{n_y}(\sqrt{\alpha\omega}y_i)}{\partial \alpha} + H_{n_y}(\sqrt{\alpha\omega}y_i) \frac{\partial H_{n_x}(\sqrt{\alpha\omega}x_i)}{\partial \alpha} - \right. \\ & \left. \frac{1}{2}\omega H_{n_x}(\sqrt{\alpha\omega}x_i) H_{n_y}(\sqrt{\alpha\omega}y_i)(x_i^2 + y_i^2) \right) \end{aligned} \quad (112)$$

and

$$\frac{\partial \phi_{n_x, n_y}(x_i, y_i)}{\partial \beta} = 0 \quad (113)$$

A.3 Hermite polynomials and their derivatives

The Hermite polynomials are the solutions of the following differential equation

$$\frac{d^2 H(x)}{dx^2} - 2x \frac{dH(x)}{dx} + (\lambda - 1)H(x) = 0. \quad (114)$$

We need the first four polynomials in this project,

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_2(x) &= 4x^2 - 2 \\ H_3(x) &= 8x^3 - 12x \end{aligned} \quad (115)$$

We list here the different derivatives of $H_{n_x}(\sqrt{\alpha\omega}x_i)$. The corresponding relations for $H_{n_y}(\sqrt{\alpha\omega}y_i)$ can be obtained by substituting $x_i \rightarrow y_i$.

$$\begin{aligned} \frac{\partial H_0(\sqrt{\alpha\omega}x_i)}{\partial x_i} &= 0 \\ \frac{\partial H_1(\sqrt{\alpha\omega}x_i)}{\partial x_i} &= 2\sqrt{\alpha\omega} \\ \frac{\partial H_2(\sqrt{\alpha\omega}x_i)}{\partial x_i} &= 8\alpha\omega x_i \\ \frac{\partial H_3(\sqrt{\alpha\omega}x_i)}{\partial x_i} &= 24(\alpha\omega)^{3/2}x_i^2 - 12\sqrt{\alpha\omega} \end{aligned} \quad (116)$$

and

$$\begin{aligned} \frac{\partial^2 H_0(\sqrt{\alpha\omega}x_i)}{\partial x_i^2} &= 0 \\ \frac{\partial^2 H_1(\sqrt{\alpha\omega}x_i)}{\partial x_i^2} &= 0 \\ \frac{\partial^2 H_2(\sqrt{\alpha\omega}x_i)}{\partial x_i^2} &= 8\alpha\omega \\ \frac{\partial^2 H_3(\sqrt{\alpha\omega}x_i)}{\partial x_i^2} &= 48(\alpha\omega)^{3/2}x_i \end{aligned} \quad (117)$$

and

$$\begin{aligned} \frac{\partial H_0(\sqrt{\alpha\omega}x_i)}{\partial \alpha} &= 0 \\ \frac{\partial H_1(\sqrt{\alpha\omega}x_i)}{\partial \alpha} &= \sqrt{\omega/\alpha}x_i \\ \frac{\partial H_2(\sqrt{\alpha\omega}x_i)}{\partial \alpha} &= 4\omega x_i^2 \\ \frac{\partial H_3(\sqrt{\alpha\omega}x_i)}{\partial \alpha} &= 12\sqrt{\alpha\omega}^{3/2}x_i^3 - 6\sqrt{\omega/\alpha}x_i \end{aligned} \quad (118)$$

References

- [1] M. Taut, *Two electrons in an external oscillator potential: Particular analytic solutions of a Coulomb interaction problem* Phys. Rev. A **48**, 3561 - 3566 (1993).
- [2] M. L. Pedersen, G. Hagen, M. Hjorth-Jensen, S. Kvaal, and F. Pederiva, *Ab initio computation of the energies of circular quantum dots* Phys. Rev. B **84**, 115302 (2011)
- [3] Jules W. Moskowitz, M. H. Kalos, *A new look at correlations in atomic and molecular systems. I. Application of fermion monte carlo variational method.* Int. J. Quantum Chem. **20** 1107 (1981)
- [4] Høgberget, Jørgen, *Quantum Monte-Carlo Studies of Generalized Many-Body Systems* Thesis for the degree of Master of Science, University of Oslo (2013)