Project 2

Amund Midtgård Raniseth Anna Stray Rongve
Knut Magnus Aasrud

June 6, 2021

Abstract

Rust is used to develop a ground state solver for fermionic systems in a harmonic oscillator potential, utilizing the variational principle, Monte Carlo integration and two different implementations of the Metropolis-Hastings algorithm. The results are compared with analytical solutions to the corresponding systems.

The effect of the so-called Jastrow factor shows a **increase/decrease** on energies, one-body densities and . . . at **these conditions - eg system N = 2** and **this result** on the higher number of particles.

Chapter 1

Introduction

Quantum dots are considered as point particles with confined electrons, usually existing in semiconductor materials, with different electrical and optical properties in relation with its host. The difference in properties is due to quantum mechanical effects, yielding quantification of the energy of the system. This technology can be utilized in photovoltaic devices to enhance performance, in LED's, photodetectors and more.

This paper presents a numerical study of such quantum dots, confined in a harmonic oscillator potential. The system in question consists of closed shell configurations, from $N = 2$ to $N = 12$ electrons. The energy of the ground state is calculated using the variational Monte Carlo method, where a set of variational parameters describing a trial wave function are optimized to obtain the lowest energy. For the simplest system the numerical results are compared to analytical calculations.

The report first gives an overview of the theoretical background of the system and the methods. The derivations of the analytical expressions for the simplest system are presented, to give a benchmark unto which we can compare the performance and accuracy of our numerical solver. Thereafter we present the methodology behind our numerical solver. Lastly, the results from our calculations are presented, followed by a discussion of them. Larger derivations are shown in the appendix.

Chapter 2

Theory

We consider a system of electrons situated in an isotropic harmonic oscillator potential. We will use Hartree's atomic units[1] to get the idealized Hamiltonian presented below:

$$H = \sum_{i=1}^{N} \left(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}\omega^2 |\mathbf{r}_i|^2 \right) + \sum_{i<j} \frac{1}{r_{ij}}. \tag{2.1}$$

Here $r_{ij} = |r_i - r_j|$ is the distance between two electrons. The first sum is the single particle harmonic oscillator potential. Because electrons repel each other, we also get the latter repulsive sum as part of the Hamiltonian - the perturbation of the system.

2.1 The unperturbed wave function

Disregarding interactions, there is a closed-form solution for the Hamiltonian shown in equation (2.1) for a single particle. The solutions follows [2]:

$$\phi_{n_x,n_y}(x,y,\alpha) = AH_{n_x}(\sqrt{\alpha\omega}x)H_{n_y}(\sqrt{\alpha\omega}y)\exp\left[-\frac{\alpha\omega}{2}(x^2+y^2)\right]. \tag{2.2}$$

Here, H_{n_i} are Hermite polynomials (see A.1), and A is the normalization constant. For the lowest lying state, we have $n_x = n_y = 0$ and hence the energy of a non-interacting fermion ϵ is:

$$\epsilon_{n_x,n_y} = \omega(n_x + n_y + 1) = \omega. \tag{2.3}$$

[1] $\hbar = c = e = m_e = 1$, see [1].

The Pauli exclusion principle states that two fermions can not occupy the same quantum state simultaneously. For each state (n_x, n_y) a fermion may have spin up or down, which means it can be occupied by at most two fermions. Using this principle, the ground state energies of the closed-shell configurations $N = 2, 6, 12$ and 20 can easily be calculated using equation (2.3). The energies are given in table **??**.

Table: The ground state energy of N non-interacting particles in an isotropic harmonic potential well. ω is the oscillator frequency. Energies are given in Hartree's atomic units. | **Number of particles** N | E **[a.u]** | | —-: | — | | 2 | 2ω | | 6 | 10ω | | 12 | 28ω | | 20 | 60ω | {#tbl:energies}

These energies serve as great values to benchmark our program against.

2.2 The complete wave function

Single harmonic oscillators are solvable analytically, but introducing the repulsive perturbation forces us to tackle the problem differently. We choose a variational Monte Carlo approach, and use the Slater-Jastrow type of trial wave function, namely

$$\Psi_T(\mathbf{R}, \alpha, \beta) = \Psi_D \Psi_J = \det(D(\mathbf{R}, \alpha)) \exp(J(\mathbf{R}, \beta)),$$

where $D(\mathbf{R})$ is a Slater matrix and $J(\mathbf{R})$ is a Padé-Jastrow correlation function. $\mathbf{R}$ here represents the set of all the individual particle's positions, and α and β are the variational parameters. Following [2], our ansatz for the factors of this trial wave function is:

$$\begin{aligned} \Psi_D &= \det(D(\mathbf{R}, \alpha)), \qquad D_{ij} = \phi_j(\mathbf{r}_i, \alpha), \\ \Psi_J &= \prod_{i<j}^{N} \exp\left(\frac{a r_{ij}}{1 + \beta r_{ij}}\right). \end{aligned} \tag{2.4}$$

$\phi_j(\mathbf{r}_i)$ is the single particle wave function for the i-th fermion, as shown in (2.2), with j being an index describing each unique quantum state[2]. The coefficient $a = 1$ when the electrons i and j have anti-parallel spins, and $a = \frac{1}{3}$ when their spins are parallel. The index notation on the product is as explained in A.2.

2.2.1 A system of $N = 2$ fermions

Expanding the ansatz (2.4) for a system of two fermions, the trial wave function is reduced to:

[2] E.g. $(0, 0, \uparrow)$, $(2, 1, \downarrow)$, etc.

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

The total spin in the ground state of this system is simply zero as the two fermions are paired with opposite spins.

2.3 Local energy

We define the *local energy* of our wave function as:

$$E_L \equiv \frac{1}{\Psi} H\Psi.$$

As shown in B.1, the local energy for a two-fermion system is:

$$\begin{aligned} E_L &= 2\alpha\omega + \frac{1}{2} + \omega^2(1-\alpha^2)(r_1^2 + r_2^2) \\ &- \frac{a}{(1+\beta r_{12})^2}\left(-\alpha\omega r_{12} + \frac{a}{(1+\beta r_{12})^2} + \frac{1-\beta r_{12}}{r_{12}(1+\beta r_{12})}\right) + \frac{1}{r_{12}}. \end{aligned}$$

2.4 Quantum Force

Importance sampling requires the quantum force, which for the two-fermion case is given by

$$F = -2\alpha\omega \mathrm{r}_1 + \frac{2a}{r_{12}\left(1+\beta r_{12}\right)^2}\mathrm{r}_{12} - 2\alpha\omega \mathrm{r}_2 + \frac{2a}{r_{12}\left(1+\beta r_{12}\right)^2}\mathrm{r}_{21},$$

or generally as

$$F = 2\frac{\nabla\Psi_T}{\Psi_T}$$

as shown in Appendix B.1.

2.5 One-body density

The one-body density can aid in visualizing and optimizing the wavefunction, as well as calculating exitation energies. It describes the probability of finding any of the N electrons in the volume $d\mathbf{r}_1$. The density is defined as [3]:

$$\rho(\mathbf{r_1}) = \int_{\mathbf{r}_2}\int_{\mathbf{r}_3}\cdots\int_{\mathbf{r}_N} |\Psi(\mathbf{r_1}, ..., \mathbf{r}_N)|^2 d\mathbf{r_2}...d\mathbf{r}_N. \tag{2.5}$$

Notice that we integrate $|\psi|^2$ over all particles but the one we are considering. The one-body density is normalized over the number of particles.

By the Pauli principle, no electrons can occupy the same state, making the integral a bit simpler. This is accounted for by the repulsion/pertubation term in the Hamiltonian.

2.6 The Virial Theorem

The Virial theorem relates the avarage time $\langle T \rangle$ for the total kinetic and potential energy in a system of N particles (or planets) by the following equation

$$\langle T \rangle = -\frac{1}{2}\sum_{k=1}^{N} \langle \mathbf{F_k} \cdot \mathbf{r_k} \rangle,$$

where $\mathbf{F}_k$ and $\mathbf{r}_k$ is the is the force on and position of particle k respectively. The theorem allows for calculations of the average total kinetic energy of complex systems, independent of the temperature.

A more relevant quentum mechanical version of the theorem arises when we consider the potential around the confined particles instead of the force. Hence,

$$2\langle T \rangle = -\frac{1}{2}\sum_{n} \langle X_n \frac{dV}{dX_n} \rangle.$$

For a simple Harmonic oscillator, the potential, V, is

$$V = \frac{1}{2}kx^2$$

.

Chapter 3

Method

3.1 Variational Monte Carlo

Our variational Monte Carlo approach is as explained in our previous work [4]. Roughly, it proceeds by proposing a change to the system $\mathbf{R} \mapsto \mathbf{R}'$ by changing the position of a single particle $\mathbf{r}_i$. The choice of this particle and how it moves is done both randomly and by way of the *quantum force*, both explained by [4]. From the states $\mathbf{R}$ and $\mathbf{R}'$, and the trial wave function Ψ_T, we evaluate an acceptance factor, that determines whether or not we accept the proposed changed system. The flowchart shown in figure 3.1 briefly describes this process.

Regardless of whether the new step is accepted or not, the desired quantities - in our case the energy, its gradient and their composites - are sampled in Monte Carlo integration. The integrated values are then used in steepest gradient descent [4] to find the optimal variational parameters.

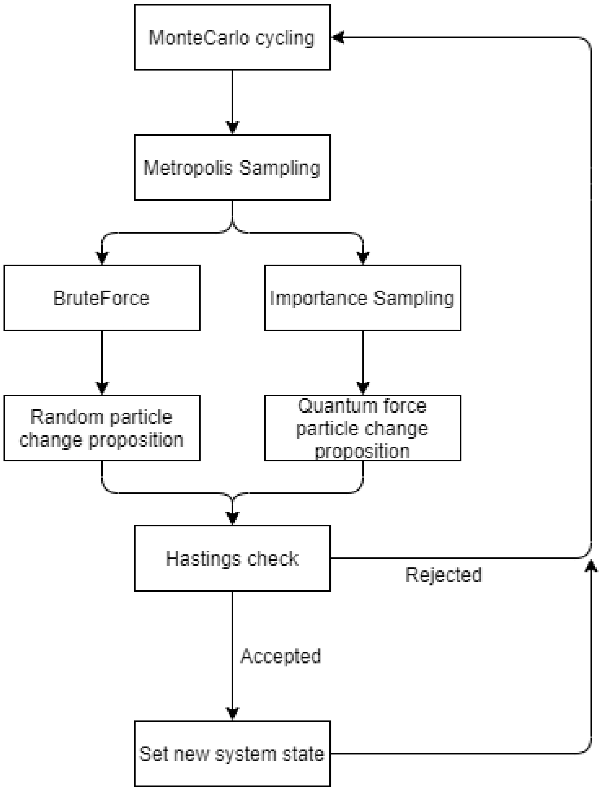

Figure 3.1: Flowchart showcasing our Monte Carlo sampling

3.2 Optimization of the wave function ratio

In our approach, the most time-consuming calculation is the evaluation of the wave function. For each proposed step in the Metropolis algorithm, we need to evaluate it to determine the acceptance factor, and if the step is accepted, yet another evaluation is needed (although this might be stored for reuse). This is expensive, so we need to optimize this process to scale well with the size of the system.

As previously stated, we find the acceptance factor in our Metropolis algorithm by introducing the proposed system change $\mathbf{R} \mapsto \mathbf{R}'$ in our Metropolis algorithm, with a single particle change $\mathbf{r}_p \mapsto \mathbf{r}'_p$. The acceptance factor depends on the wave function ratio $\mathcal{R}$, which we can split up like this:

$$\mathcal{R} = \frac{\Psi_T(\mathbf{R}')}{\Psi_T(\mathbf{R})} = \frac{\Psi_D(\mathbf{R}')}{\Psi_D(\mathbf{R})} \cdot \frac{\Psi_J(\mathbf{R}')}{\Psi_J(\mathbf{R})} = \mathcal{R}_D \mathcal{R}_J.$$

We will optimize each of these ratios separately.

3.2.1 Optimizing $\mathcal{R}_D$

Each new ratio $\mathcal{R}_D = \frac{\det(D(\mathbf{R}'))}{\det(D(\mathbf{R}))}$ would require $\mathcal{O}(N^3)$ operations if done with Gaussian elimination. However, as found by [5], this is decreased to $\mathcal{O}(N)$ by utilizing the inverse matrix D^{-1} as such:

$$\mathcal{R}_D = 1 + \mathbf{v}_p^T D^{-1} \mathbf{e}_p, \quad \text{where } \mathbf{v}_p = \begin{bmatrix} \phi_1(\mathbf{r}'_p) - \phi_1(\mathbf{r}_p) \\ \vdots \\ \phi_N(\mathbf{r}'_p) - \phi_N(\mathbf{r}_p) \end{bmatrix}.$$

$\mathbf{e}_p$ is simply the unit vector with 1 on the p-th entry and zero everywhere else. It serves the purpose of extracting the p-th column from D^{-1}.

Now, how do we calculate D^{-1} effectively? Once again, Gaussian elimination gives us an $\mathcal{O}(N^3)$ cost, which is no-go. However, if we do the matrix inversion with Gaussian elimination initially, to acquire D_0^{-1}, we can iteratively find the succeeding inversions by using a special case of the *Sherman-Morrison-Woodbury formula*[1] [6], which states:

$$\left(D + D^{-1}\mathbf{e}_p\mathbf{v}_p^T\right)^{-1} = D^{-1} - \frac{D^{-1}\mathbf{e}_p\mathbf{v}_p^T D^{-1}}{1 + \mathbf{v}_p^T D^{-1}\mathbf{e}_p}.$$

We introduce the index k, referring to an arbitrary Monte Carlo step. Recognizing $D_k + D_k^{-1}\mathbf{e}_p\mathbf{v}_p^T$ as D_{k+1} [5], we can simplify this to the iterative statement

[1] Which, confusingly, is just called the *Sherman-Morrison formula*.

$$D_{k+1}^{-1} = \left(\mathbf{I} - \frac{D_k^{-1}\mathbf{e}_p\mathbf{v}_p^T}{\mathcal{R}_{D,k}} \right) D_k^{-1}.$$

This has an operation complexity of $\mathcal{O}(N)$.

3.2.2 Optimizing $\mathcal{R}_J$

We consider the ratio

$$\mathcal{R}_J = \frac{\Psi_J'}{\Psi_J} = \prod_{i<j}^{N} \frac{\exp(J(r_{ij}', \beta))}{\exp(J(r_{ij}, \beta))}.$$

This naive operation scales in the order of $\mathcal{O}\left(\frac{N^2(N-1)}{2}\right)$. It is however easily optimized, by employing the fact that we only move one particle's position $\mathbf{r}_p$, which means only distances r_{pj} are changed. We thus get

$$\mathcal{R}_J = \prod_{i\neq p}^{N} \frac{\exp(J(r_{pi}', \beta))}{\exp(J(r_{pi}, \beta))} = \exp\left(\sum_{i\neq p}^{N} J(r_{pi}', \beta) - J(r_{pi}, \beta) \right),$$

which scales in the order of $\mathcal{O}(N-1)$ operations.

3.3 Numerical derivatives

For gradients, we utilize the two-point approximation of the first derivative:

$$\frac{dg(x)}{dx} \approx \frac{g(x+\Delta x) - g(x-\Delta x)}{2\Delta x}.$$

Similarly, we also calculate the Laplacians numerically by using the three-point approximation, as follows:

$$\frac{d^2g(x)}{dx^2} \approx \frac{g(x+\Delta x) - 2g(x) + g(x-\Delta x)}{\Delta x^2}$$

We let the step size Δx run towards 0 and have found 10^{-5} to be sufficiently small. The error here goes as $\mathcal{O}(\delta x^2)$.

3.4 Testing

Testing in Rust is normally divided in two categories: *unit tests* and *integration tests*. Unit tests are small codes to test specific functions inside the code. These tests are normally written in the same file as the functions themselves, but inside a module annotated with `#[cfg(test)]`.

On the other hand, integration tests are written externally to the library, and is made to test the integration of the functions in the program. These tests are often much larger than unit tests and are made to make sure that the internal functions work well with each other, from the standpoint of an external user. Therefore, integration tests are normally written in a separate `tests` directory at the same level as the `src` directory.

We will write mainly unit tests in our program, to ensure that our functions return the expected values, and to reduce the mental overhead of debugging when making larger changes to the codebase.

More on testing can be found in the official documentation of the Rust programming language [7].

3.5 Evaluation and performance of the VMC solver

The first performance evaluations are done for a case with two electrons in a quantum dot with frequency of $\hbar\omega = 1$.

3.5.1 Performance evaluation of different energy calculation methods

The performance of the analytical expression for the local energy is compared to the performance of the numerical derivation of the kinetic energy in results section **??**. This test is performed without importance sampling and the Jastrow factor. Following this, importance sampling is added and tested only with the analytical expression for the local energy. Lastly, a blocking analysis is performed in order to obtain the optimal standard deviation.

The energy should equal 2.0 atomic units with a variance exactly equal to zero.

3.5.2 Evaluating the variational parameters

By using the steepest descent method, the best variational parameters, α and β are found. The results for this is found in section 4.2.

3.5.3 Computation of the two-electron system

The minimum energy of the system is computed and compared to Taut's work . In addition, the mean distance between the two electrons and the one-body density is calculated for the best variational parameters. These results are also compared with the results form the same computations, where only the pure harmonic oscillator wavefunctions are used, and where pure HO wavefunctions are used but without the Jastrow factor.

Lastly the expectation value for the kinetic energy is calculated with $\omega \in 0.01, 0.05, 0.1, 0.5, 1.0$.

3.6 Increasing computational performance

Being able to simulate many-body system at large scale without running out of time is crucial. Hence taking advantage of available tools such as compilation flags (e.g., for vectorization) and parallelization is important. The Rust language provides a great set of such tools - like the ones used in C++, but safer.

Since we run many different simulations with unique parameters, we parallelize over these simulations, to keep the logic of our program simple. This allows us to utilize all our cores' computation power, while still not needing to program concurrently. For future work, parallelization should be done further into the "core" of the program, to increase performance in single runs. For details on how we do vectorization in Rust, see [4].

Chapter 4

Results

4.1 Two fermions

To validate our algorithm a simulation of the simplest case with two electrons without the Jastrow factor and perturbation was done, expecting an energy-output of exactly 2 au and a variance of 0. The results, together with a performance analysis (see below), is listed in Table 4.1.

A performance analysis, taking the average time over 10 runs, of the analytical expression for the local energy, numerical derivation of the kinetic energy and the analytical local energy with importance sampling is shown in table4.1 below.

Table 4.1: Results from computations of the expectation value of the energy using both Importance Sampling and the Metropolis algorithm for both the analytical expression for the local energy(see equation **??**) and numerical derivation of the kinetic energy. To compare the performance of the different configurations, the algorithms are timed over 10 runs and averaged.

Sampling method	Avarage time [s]	$\langle E \rangle$	$\langle E_{kinetic} \rangle$	σ^2	σ_{blocking}
Analytical w/ Metropolis	5.25	2.000	1.03	0.0000	0.0000
Analytical w/ Importance Sampling	6.13	2.000	1.04	0.0000	0.0000
Numerical w/ Metropolis	5.82	1.9999	0.99	0.0000	0.0000

Sampling method	Avarage time [s]	$\langle E\rangle$	$\langle E_{kinetic}\rangle$	σ^2	σ_{blocking}
Numerical w/ Importance Sampling	6.3	1.9999	1.02	0.0000	0.0000

The **blocking analysis** shows that the optimal standard deviation is $FILL$.

4.2 Evaluating the variational parameters

To obtain the optimal variational parameters for the ground state energy, the steepest decent method is implemented in the Variational Monte Carlo calculations. The result, with and without the Jastrow factor are shown in Table **??**. The table also

Table 4.2: Optimal variational parameters for the different systems without the Jastrow factor, for comparison.{#tbl:results-variational-parameters}

N	ω	α	$\alpha_{\text{wo J}}$	β	$\langle E\rangle$	σ_{blocking}	$\langle E_k\rangle$	$\langle E_p\rangle$	$\langle r_{12}\rangle$
2	1								
	0.5								
	0.1								
	0.05								
	0.01								
—	———	———	——-:						
6	1								
	0.5								
	0.1								
	0.05								
	0.01								
—	———	———	——-						
12	1								
	0.5								
	0.1								
	0.05								
	0.01								
—	———	———	——-						
20	1								
	0.5								
	0.1								
	0.05								
	0.01								

4.3 Ground state energies

4.4 One body densities

The one body density is computed for 2 and 6 particles with the optimal parameters obtained during the previous calculations and $\omega = 1$. The calculations are executed with and without the Jastrow factor **(and perturbation)???** for comparison and analysis of their influence.

The one-body densities are calculated with and without the Jastrow factor for two (2) and (6) fermions. The results are shown in figure **??** below

One Body densities for 2 and 6 fermions with and without the Jastrow factor. The computations are done with $\alpha = FILL$, $\beta = FILL$ and $\omega = 1$

4.5 Performance analysis

Lastly a analysis of the algorithms are given for $N = 6$ electrons, $FILLIN$ Monte Carlo cycles, optimal variational parameters $\omega = 1$ and without the Jastrow factor. The analysis is done by comparing the average time used for a calculation with and without vectorization. The procedure is repeated with parallelization, expecting approximately a 100% speedup. The quantities which are calculated are **FILL INN**. The most time-consuming part is the **FILL INN**, and hence the clock is started here. The results are presented in table **??**. **ADD SOME MORE DESCRIPTION OF HOW HERE**

Table 4.3: Results from performance analysis with and without vectorization and compile flags. The time is averaged over 10 runs with **FILL INN** MC cycles. The sampling method is the **Brute Force Metropolis OR Importance sampling**

Optimization/compile flags	$\bar{t}$ [s]
With vectorization	
Flag 1	
Flag 2	
Flag 2	
Without Vectorization	
Flag 1	
Flag 2	
Flag 3	
Parallelization	

Chapter 5

Discussion

5.1 Verification/ Two electrons

As a validation test for our algorithm, we calculated the energy without the Jastrow factor for $N = 2$ electrons, expecting a energy of $2.0 $ au and a variance of zero. Running this system with the Metropolis (and Importance sampling) method returned a energy of $FILL$ ($FILL$) and a variance of $FILL$ ($FILL$). Together with the energy and statistical calculations, a performance analysis showed a **FILL- large, intermediate...** difference in time consumption. The analytical algorithm used $FILL$ seconds, while the numerical approach used $FILL$ seconds. Comparing the two sampling methods in question, the **Metropolis/Importance** sampling was **faster/slower** than the **Metropolis/Importance**.

5.2 Variational parameters

The variational parameters were obtained using the Steepest Gradient Decent method both with and without the Jastrow factor. The results are to be found in Table **??**. Without the factor our calculations returned $\alpha = FILL$, with the expectation of $\alpha = 1$. With the Jastrow factor, there is an additional variational parameter, β.

5.3 Energy minima

The results from the energy minima calculations are also listed in Table **??**.

The obtained mean distance between two fermions shows to be strongly (inverse) **dependent/independent** on the frequency, ω, which is **expected/unexpected**. Decreasing the frequency with a factor of $FILL$

increases the distance with a factor of $FILL$. Looking at the energy, there is a **decrease/increase** in energy when **decreasing** the frequency.

5.4 One Body density

The results from the one body density calculations utilizing optimal parameters are shown in Figure **??**. With perturbation (interaction term), there is a (small?) **increase/decrease**

5.5 Performance analysis

Table **??** gives an overview of the performance analysis of running our algorithm with and without vectorization and parallelization for $N = 6$ electrons.

Chapter 6

Conclusion

The well-known Variational Monte Carlo algorithm has been implemented to perform calculations on confined electrons with closed shell configurations in a quantum dot. To find the energy closes to the ground state, two variational parameters are implemented. Using the steepest gradient decent these two are found for different vibrational frequencies. The results are compared with both results from calculations utilizing analytical expressions and values from literature.

The comparison shows a **significant**(?) speedup using the analytical expression compared to numerical calculations. The energies **are more/less off** for the numerical approach. Hence, if an analytical expression is obtainable, it is recommended implementing this, instead of the using the numerical approach. This applies especially to higher order (dimension and number of particles) systems.

For $N = 2$ electrons we compared the performance of two sampling methods, Brute Force Metropolis, and Importance sampling. The **first mentioned** showed a higher performance, but from earlier (**proj1 ref**) we know that the performance Importance Sampling will rapidly catch up and overtake for a higher number of particles.

Large parts of this project involve studying the influence the Jastrow factor has on various parameters. Looking at the energy, there is a $FILL$ **higher/lower** energy with the Jastrow factor, compared to without. It gives a **higher/lower** accuracy and the performance **increase/decrease**. The effect of the one-body densities is even **greater(/less)** , as seen in Figure **??**. For the distance between the particles, there is an (a) **increase/decrease** utilizing the Jastrow factor. Hence, if accuracy is important for calculations, the Jastrow factor should be **used/turned off** and

As expected, increasing the number of particles gives more time-consuming calculations, both du to the utilization of the slater determinant and the non-existing analytical expressions. (**Check (and prob rewrite this**) Looking at the one-body density, one sees a **increased/decreased** density with increasing number of particles. This is also **expected/unexpected**. Maybe more importantly, the energy calculation gets a **lower/higher/unaffected** accuracy (compared to Taut's work).

Looking at the two variational parameters, α and β, we find several sets of the two yielding lower energies. **Check this**

The frequency dependent calculations shows a great impact on the energy, as expected. Increased frequency gives a **increase/decrease** in energy, where the **potential/kinetic** is affected the most.

Running a performance analysis for $N = 6$ particles with and without different optimization flags we found that **FILL** worked the best, as seen in Table **??**.

To improve performance of this algorithm, we recommend looking further into the parallelization of our code. Other things are **FILL**

Appendix A

Definitions and notation

A.1 Hermite polynomials

The Hermite polynomials are the solutions to the following contour integral [8]:

$$H_n(z) = \frac{n!}{2\pi i} \oint e^{-r^2 - tz} t^{-n-1} dt.$$

In this report, we will consider the real Hermite polynomials, and only up to o the der of four. A computationally efficient way of finding these is given by the following sequence:

$$H_n(x) = c_{n+m} x^n + c_{n-1+m} x^{n-1} ... + c_m x^0, \qquad m = \sum_{i=1}^{n} i$$

where the coefficients c_n are given by the triangle sequence shown in (A.1).

$$c_i = 1, 0, 2, -2, 0, 4, 0, -12, 0, 8, 12, 0, -48, ... \tag{A.1}$$

This is just a a selection of the sequence, more can be found from the work by [9].

A.2 Index notation for sums and products

For products and sums, the following convention is used:

$$\sum_{i<j}^{N} = \sum_{i=1}^{N} \sum_{j=i+1}^{N}, \quad \text{or} \quad \prod_{i<j}^{N} = \prod_{i=1}^{N} \prod_{j=i+1}^{N}$$

Appendix B

Derivations

B.1 Analytical derivation of the Quantum Force, Laplacian and Local energy of two-fermion systems

The trial wavefunction of a two-particle system is

$$\Psi_T(\mathbf{r_1}, \mathbf{r_2}) = \Psi_1 * \Psi_2 = C \exp\left(-\alpha\omega(r_1^2 + r_2^2)/2\right) \exp\left(\frac{ar_{12}}{1 + \beta r_{12}}\right)$$

The Laplacian is the double derivative in all dimensions, defined as:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

The calculations: First we change the laplacian to work with a cartesian twodimensional system:

$$\Delta f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \frac{\partial^2 f}{\partial y_1^2} + \frac{\partial^2 f}{\partial y_2^2}$$

Then the wavefunction is inserted.

$$\Delta \Psi_T = \frac{\partial^2 \Psi_T}{\partial x_1^2} + \frac{\partial^2 \Psi_T}{\partial x_2^2} + \frac{\partial^2 \Psi_T}{\partial y_1^2} + \frac{\partial^2 \Psi_T}{\partial y_2^2}$$

We see that the trial wavefunction is composed of two exponential terms, and to do the derivative, we can use the derivative product rule twice.

$$(fg)'' = (f'g + fg')' = f''g + 2f'g' + fg'' \tag{B.1}$$

where

$$\begin{aligned} f =& C\exp\left(-\alpha\omega(r_1^2 + r_2^2)/2\right) \\ g =& \exp\left(\frac{ar_{12}}{1+\beta r_{12}}\right) \end{aligned}$$

The two following equalities are then used to find the first derivative of f and g

$$\frac{\partial r_1}{\partial x_1} = x_1/r_1, \quad \frac{\partial r_{12}}{\partial x_1} = \left(x_1 - x_2\right)/r_1$$

$$\frac{\partial f}{\partial x_1} = -\alpha\omega x_1 f, \quad \nabla_i f = -\alpha\omega f \mathbf{r}_i \tag{B.2}$$

Where i denotes the specific particle, and the particle position r_i equals (x_i, y_i). For the second term g we have

$$\frac{\partial g}{\partial x_1} = g\frac{a\left(x_1 - x_2\right)}{r_{12}\left(1+\beta r_{12}\right)^2}, \quad \nabla_i g = g\frac{a}{r_{12}\left(1+\beta r_{12}\right)^2}\mathbf{r}_{ij} \tag{B.3}$$

Where j is the opposite particle of i and the distance from j to i, $\mathbf{r}_{ij} = (x_i - x_j, y_i - y_j)$.

From this we can actually find an analytical solution to the *quantum force* used in importance sampling, defined as

$$F = 2\frac{\nabla\Psi_T}{\Psi_T} = 2\frac{f'g + fg'}{fg}$$

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

Next, we calculate the Laplacian, or double derivative, of the first term, f.

$$\frac{\partial^2 f}{\partial x_1^2} = f\left(\alpha^2\omega^2 x_1^2 - \alpha\omega\right), \quad \nabla^2 f = f\left(\alpha^2\omega^2\left(r_1^2 + r_2^2\right) - 4\alpha\omega\right) \tag{B.4}$$

And the second term, g.

$$\frac{\partial^2 g}{\partial x_1^2} = g[\frac{a^2 \left(x_1 - x_2\right)^2}{r_{12}^2 \left(1 + \beta r_{12}\right)^4} + \frac{a r_{12} \left(1 + \beta r_{12}\right)^2}{r_{12}^2 \left(1 + \beta r_{12}\right)^4} \\ - \frac{a \left(x_1 - x_2\right) \left[\left(x_1 - x_2\right)/r_{12} \left(1 + \beta r_{12}\right)^2 + 2 r_{12} \left(1 + \beta r_{12}\right) \beta \left(x_1 - x_2\right)/r_{12}\right]}{r_{12}^2 \left(1 + \beta r_{12}\right)^4}\Bigg]$$

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

With this, we get

$$\nabla^2 g = g \left[\frac{2a^2}{\left(1 + \beta r_{12}\right)^4} + \frac{4a}{r_{12} \left(1 + \beta r_{12}\right)^2} - \frac{2a}{r_{12} \left(1 + \beta r_{12}\right)^2} - \frac{2 a \beta}{\left(1 + \beta r_{12}\right)^3}\right]$$

Which can be further shortened by pulling $\frac{2a}{(1+\beta r_{12})^2}$ outside the brackets to:

$$\nabla^2 g = g \frac{2a}{\left(1 + \beta r_{12}\right)^2} \left[\frac{a}{\left(1 + \beta r_{12}\right)^2} + \frac{1}{r_{12}} - \frac{2\beta}{1 + \beta r_{12}}\right] \tag{B.5}$$

Now, by inserting f'', g'', f' and g' from equations B.4, B.5, B.2 and B.3 into equation B.1, we actually obtain the *Laplacian* of the trial wavefunction $\nabla^2 \Psi_T$. First we simplify the middle term:

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

And then, we insert the double derivatives.

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

References

[1] D. R. Hartree, “The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 1, pp. 89–110, Jan. 1928, doi: 10.1017/S0305004100011919.

[2] M. Hjorth-Jensen, “Project 2, Variational Monte Carlo studies of electronic systems.” Department of Physics, University of Oslo, Jul. 2021.

[3] J. Høgberget, “Quantum Monte-Carlo Studies of Generalized Many-body Systems,” pp. 66–67, 2013.

[4] K. M. Aasrud, A. S. Rongve, and A. M. Raniseth, “Variational Monte Carlo studies of bosonic systems,” University of Oslo, 1, Apr. 2021. [Online]. Available: https://github.com/kmaasrud/vmc-fys4411.

[5] P. K. V. V. Nukala and P. R. C. Kent, “A Fast and Efficient Algorithm for Slater Determinant Updates in Quantum Monte Carlo Simulations,” *The Journal of Chemical Physics*, vol. 130, no. 20, p. 204105, May 2009, doi: 10.1063/1.3142703.

[6] G. H. Golub and C. F. Van Loan, *Matrix computations*, Fourth edition. Baltimore: The Johns Hopkins University Press, 2013.

[7] Rust-Docs, “Test organization,” 2021. https://doc.rust-lang.org/book/ch11-03-test-organization.html (accessed Apr. 27, 2021).

[8] G. B. Arfken and H.-J. Weber, *Mathematical methods for physicists*, 6th ed. Boston: Elsevier, 2005.

[9] V. Jovovic, “A060821 - triangle t(n,k) read by rows giving coefficients of hermite polynomial of order n (n >= 0, 0 <= k <= n).” *The On-Line Encyclopedia of Integer Sequences*. Apr. 2001, Accessed: Apr. 27, 2021. [Online]. Available: https://oeis.org/A060821.