Variational Monte Carlo

Written by:

Oliver Hebnes & Mohamed Ismail

Department of physics UiO

March 24, 2020

Abstract

In this piece of research we implement a variational Monte Carlo with both the brute force Metropolis and the Metropolis-Hastings algorithms for the purpose of simulating a Bose-Einstein condensation of a hard sphere Bose gas. We study both a non-interacting and interacting system with an external potential consisting of a spherical and elliptical harmonic oscillator, respectively. We find that the minimum energy for all tested number of particles 1, 10, 50, 100 and 500 are found with the variational parameter $\alpha = 0.5$, both in the non-interacting case and the interacting case. This does also include using the method gradient descent. Moreover, we also observe that having an analytical way of calculating the energy is a gift from the above, which makes the program run much faster, compared to using a numerical approach to calculating the energy. The brute force Metropolis and Metropolis-Hastings algorithms exhibit a difference in the behaviour of the acceptance rate for the proposed moves in the variational Monte Carlo simulation. The final part was dedicated to studying the one-body density of a 10-particle system, where the interacting case of the one-body density moved to a larger radii compared to the non-interacting one. However, the difference is not significantly between the cases. We receive this set-back with humility and grace, leaving the study and implementation of stronger programs up to future articles on this matter.

Contents

1 Introduction

In this project we will study the computational method *variational Monte Carlo* (VMC), in an attempt to simulate a *Bose-Einstein condensation* (BEC) of a trapped, hard sphere Bose gas. The objective is to simulate how particles will move relative to each other given a probability distribution and an acceptance algorithm that goes under the name *Metropolis*.

At first, we will utilise our pristine and unused machinery to try to minimise the energy of a system accommodating a single particle by using the variational principle, such that we only need to vary a peculiar, but noteworthy variational parameter. This will also serve as an excellent benchmark and will[1] prepare our machinery before we venture on to a numerical calculation of the energy on a *several-particle* system with a spherical potential.

Then we will turn our attention towards a neat and improved version of the Metropolis-Hastings algorithm, where we take advantage of the drifting of atoms and reduce the amount of useless iterations that our other algorithm, which is now known as primitive and will accordingly be named as brute force. And since the new algorithm is sampling everything important, it will go by the remarkable name of importance sampling.

Given that the newly equipped instruments are operational with our machinery, we will then evolve our system to utilise an elliptical trap with a pairwise, repulsive potential. This will introduce an elegant statistical method for calculating the correlation between every Monte Carlo cycle, namely the Blocking method. In addition, we will be using two different methods for finding the particular variational parameter for these systems, where one method is the classical 'manual vary parameter' method, while the other one is the esteemed gradient descent. For the final touch of the project, we will also take a quick glimpse at the distribution of the one-body density, with our newly developed VMC program.

2 Theory

2.1 System Properties

The system we are studying consists of N bosons trapped in a harmonic oscillator, where the Hamiltonian of the system is given as

$$H = \sum_{i}^{N} \left(\frac{-\hbar^2}{2m} \bigtriangledown_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{1}$$

with V_{ext} acts as the external potential of the system, which is the harmonic oscillator potential. The external potential we will use is a spherical (S) or an elliptical (E) harmonic trap in one, two and finally three dimensions, given as

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2} m \omega_{ho}^2 r^2 & \text{(Spherical)} \\ \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] & \text{(Elliptical)} \end{cases} \tag{2}$$

Here ω_{ho}^2 defines the trap potential strength. In the case of the elliptical trap, $V_{ext}(x, y, z)$, $\omega_{ho} = \omega_{\perp}$ is the trap frequency in the perpendicular or xy plane and ω_z the frequency in the z

[1] as a metaphor

direction.

V_{int} is defined as the inter-boson interaction by a pairwise, repulsive potential and is given as

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{3}$$

where a is the so-called hard-core diameter of the bosons. Clearly, $V_{int}(|\mathbf{r}_i - \mathbf{r}_j|)$ is zero if the bosons are separated by a distance $|\mathbf{r}_i - \mathbf{r}_j|$ greater than a but infinite if they attempt to come within a distance $|\mathbf{r}_i - \mathbf{r}_j| \leq a$.

The wavefunction we will use to study our system is the trial wave function for the ground state with N atoms given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|)\right], \tag{4}$$

where α and β are variational parameters. The single-particle wave function is proportional to the harmonic oscillator function for the ground state, i.e.,

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right]. \tag{5}$$

For spherical traps we have $\beta = 1$ and for non-interacting bosons ($a = 0$) we have $\alpha = 1/2a_{ho}^2$. The correlation part of the trial wave function is

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{6}$$

This trial wave function is used to study the sensitivity of condensate and non-condensate properties to the hard sphere radius and the number of particles.

2.2 Scaling

The practicalities of using scaled units are very much needed in a system where we use several different constants and measurements. Given our Hamiltonian

$$\begin{aligned} H &= \sum_i (-\frac{1}{2}\nabla_i^2 + V_{ext})) + \sum V_{int} \\ &= \frac{1}{2}\sum_i (-\frac{\hbar^2}{m}\nabla_i^2 + (m\omega_{ho}^2(x^2 + y^2) + w_z^2 z^2)) + \sum V_{int} \end{aligned}$$

we introduce energy in units of $\hbar\omega_{ho}$, such that we get a new Hamiltonian $H' = H/\hbar\omega_{ho}$. In addition, we scale every dimension such that we get for an arbitrary dimension

$$r' = \frac{r}{a_{ho}} = r \cdot \left(\frac{m\omega_{ho}}{\hbar}\right)^{1/2}$$

By applying the new units will result in the new Hamiltonian

$$\begin{aligned} H' &= \frac{1}{2}\sum_i \Big(\frac{\hbar}{m\omega_{ho}}\nabla_i^2 + \frac{m\omega_{ho}^2}{\hbar\omega_{ho}}(x'^2+y'^2)\frac{\hbar}{m\omega_{ho}} + \frac{m\omega_z^2}{\hbar\omega_{ho}}z'^2\cdot\frac{\hbar}{m\omega_{ho}}\Big) + \sum V_{int} \\ &= \frac{1}{2}\sum_i \Big(\nabla_i'^2 + x'^2 + y'^2 + \frac{\omega_z^2}{\omega_{ho}^2}z'^2\Big) + \sum V_{int} \\ &= \frac{1}{2}\sum_i \Big(\nabla_i'^2 + x'^2 + y'^2 + \gamma^2 z'^2\Big) + \sum V_{int} \end{aligned}$$

Thus, we introduce a new variable $\gamma = \omega_z/\omega_{ho}$. The same result can also be obtained by a slightly different method, which is setting $m = \hbar = \omega_{ho} = 1$. It is known as using dimensionless units and natural units, and is a well known in the computational science linguistic and culture.

2.3 Variational Principle

We want to find the ground state energy of our system for a given trial wavefunction Ψ_T.

The variational principle states that for a given trial wavefunction Ψ_T, the expectation value of the Hamiltonian, H will be an upper bound to the ground state energy, E_0 of the system

$$E_0 \leq E = \frac{\int \Psi_T^* H \Psi_T d\boldsymbol{r}}{\int \Psi_T^* \Psi_T d\boldsymbol{r}} \tag{7}$$

The complicated part of the variational principle is finding the trial wavefunction that minimizes the energy. However, by introducing variational parameters, we can minimize the energy with respect to these parameters.

2.4 Monte Carlo

By utilising the variational principle we can define a quantity called the local energy E_L, given as

$$E_L = \frac{1}{\Psi_T} H\Psi_T \tag{8}$$

such that we can use equation 7 and rewrite the energy to

$$E = \int |\Psi_T^2| E_L d\boldsymbol{r} \tag{9}$$

This integral can be solved by using Monte Carlo integration. In simple terms this means that we are approximating an integral to a sum.

$$\langle x \rangle = \int x p(x) dx \approx \frac{1}{N}\sum_{i=1}^{N} x_i p(x_i)$$

Where $\langle x \rangle$ is the sample mean, x is a random variable, $p(x)$ is a probability distribution function (PDF) and N is the number of Monte Carlo samples that is chosen to approximate the integral.

By taking advantage of the law of large numbers, it is possible to show that the approximated integral goes toward the true value, when N is very large.

$$\langle x \rangle = \int x p(x) dx = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} x_i p(x_i)$$

In our case this translates into $E_L = x$, and our probability distribution function is the wavefunction squared, $|\Psi_T|^2 = p(x)$.

$$\langle E \rangle = \int E_L |\Psi_T^2| d\boldsymbol{r} \approx \frac{1}{N} \sum_{i=1}^{N} E_{L,i} |\Psi_T(\boldsymbol{r}_i)^2|$$

2.5 Local Energy

Like it was mentioned earlier, one can calculate the energy of the system with a Monte Carlo simulation. To accomplish that, we need to calculate the local energy, E_L. Since the local energy is

$$\begin{aligned} E_L &= \frac{1}{\Psi_T} H \Psi_T \\ &= \frac{1}{\Psi_T} \sum_{i}^{N} \left(\frac{-\hbar^2}{2m} \bigtriangledown_i^2 + V_{ext}(\mathbf{r}_i) \right) \Psi_T + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j) \Psi_T, \end{aligned}$$

the most heaviest calculation will be of the second derivative of the trial wavefunction. There are two ways to approach a problem like this. One method is to find an analytical expression for the local energy and derive the second derivative of the trial wavefunction, which can be found for different cases in Appendix A.1 and B.1, while the other method is to find the second derivative by a numerical method.

2.5.1 Numerical approach

For some systems we do not know the analytical expression for the double derivative, and is thus forced to use supplementary aid. One approach to this is to use the finite difference scheme, which is to approximate the second derivative with previous values

$$\nabla^2 f(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

where h is the step size. We will compare this to the analytical approach to find out their time consumption behaviour and estimations.

2.6 Non-interacting system

We start by looking at a simple system, where there are no interaction between the particles, where $a = 0$ and $\beta = 1$. In this scenario we will use the spherical harmonic oscillator as the external potential.

As a consequence the trial wavefunction gets reduced to

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right]\right] \tag{10}$$

and for such a system, the local energy in natural units is

$$E_L = dim \times N \times \alpha + \left(\frac{1}{2} - 2\alpha^2\right) \sum_{i=1}^{N} r_i^2. \tag{11}$$

which is derived in Appendix A.1.4

Another important quantity is the drift force, which is defined as

$$F = \frac{2\nabla\Psi_T}{\Psi_T} \tag{12}$$

and it has been shown in Appendix A.2.4 that the drift force for the non-interacting system is

$$F = -4\alpha \sum_{i=1}^{N} r_i \tag{13}$$

2.7 Interacting system

In the interacting case we no longer use the spherical harmonic oscillator as the external potential, instead we use the elliptical harmonic oscillator, where $a \neq 0$ and $\beta \neq 1$. In addition, the trial wavefunction will also be rewritten and a correlation factor will be included.

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|)\right],$$

as

$$\Psi_T(\mathbf{r}) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right] \exp\left(\sum_{j<k} u(r_{jk})\right)$$

where we have defined $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ and

$$f(r_{ij}) = \exp\left(\sum_{i<j} u(r_{ij})\right),$$

with $u(r_{ij}) = \ln f(r_{ij})$. We have also

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right] = \phi(\mathbf{r}_i).$$

and for such a complex system, the local energy in natural units is

$$\begin{aligned} E_L &= \frac{1}{\Psi(\boldsymbol{r})} H\Psi(\boldsymbol{r}) \\ &= \frac{1}{\Psi(\boldsymbol{r})}\left[\sum_{i=1}^{N}\Big(-\frac{\nabla_i^2}{2} + V_{ext}(\boldsymbol{r}_i)\Big) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i,\boldsymbol{r}_j)\right]\Psi(\boldsymbol{r}) \\ &= \sum_{i=1}^{N}\Big(-\frac{\nabla_i^2\Psi(\boldsymbol{r})}{2\Psi(\boldsymbol{r})} + V_{ext}(\boldsymbol{r}_i)\Big) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i,\boldsymbol{r}_j). \end{aligned}$$

where the second derivative of the trial wave function is

$$\begin{aligned} \frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\Phi(\mathbf{r}_k)}{\Phi(\mathbf{r}_k)} &+ 2\frac{\nabla_k\Phi(\mathbf{r}_k)}{\Phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{ij})\right) \\ &+ \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \\ &+ \sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right). \end{aligned}$$

The long derivation of this can be found in Appendix B.1.

On the other hand, the drift force for the interacting case is not such a intimidating term, and is defined as

$$\begin{aligned} F &= \frac{2}{\Psi_T(\mathbf{r})}\nabla\Psi_T(\mathbf{r}) \\ &= 2\sum_{k=1}^{N}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k\Psi_T(\mathbf{r}) \\ &= 2\sum_{k=1}^{N}\left(\frac{\nabla_k\Phi(\boldsymbol{r}_k)}{\Phi(\boldsymbol{r}_k)} + \sum_{l\neq k}\nabla_k u(r_{kl})\right). \end{aligned}$$

where the derivation can be found in Appendix B.2.

2.8 One-body density

The one-body density is defined as

$$\rho(\mathbf{r}_1) = \int_{\mathbf{r}_2}\int_{\mathbf{r}_3}\cdots\int_{\mathbf{r}_N}|\Psi_T(\mathbf{r}_1,\ldots,\mathbf{r}_N)|^2 d\mathbf{r}_2\ldots\mathbf{r}_N.$$

This density describes the distribution all together of every particle in the system, which is different from $|\Psi_T|^2$ which describes the probability of finding any particle in the system. As an example, $\rho(\mathbf{r}_1 d\mathbf{r}_1)$ represents the probability of finding any of the system's N particles within

the volume element $d\mathbf{r}_1$ [1]. In this system we are dealing with bosons, which are known to be indistinguishable particles, thus it does not matter which particle's position we will use.

It is possible to calculate the one-body density analytically, however, it is also fully possible to use a Monte Carlo simulation to estimate the one-body density. The trick is to divide the volume of our system into 'bins', where for every Monte Carlo iteration we count the number of particles for every bin. Since we are estimating the integrals by a summation, it will be necessary to divide on the number of Monte Carlo iterations.

2.9 Statistical Analysis

Other quantities that are important for Monte Carlo calculations in addition to the calculation of the integral $\langle E \rangle$ are the variance σ_E^2 and the standard deviation σ_E.

The variance σ_E^2 is defined as

$$\sigma_E^2 = \frac{1}{N}\sum_{i=1}^{N}(E_{L,i} - \langle E \rangle)^2$$

and is a measure of the error in the calculation of the energy.

The standard deviation, on the other hand, is defined as the square root of the variance.

$$\sigma_E = \sqrt{\sigma_E^2}$$

Assuming that the above is a result acquired for a fixed value of N to be a measurement, that would make it possible to recalculate this for different measurements. Each such measurement can be denoted l and in every measurement we have $[E_{l,1}, E_{l,2}, \ldots, E_{l,N}]$ data points, that produces a set of averages $\langle E_l \rangle$ which is defined as

$$\langle E_l \rangle = \frac{1}{N}\sum_{i=1}^{N} E_{l,i}$$

If we repeat the measurements M times, the mean of all the measurements will be

$$\langle E_M \rangle = \frac{1}{M}\sum_{l=1}^{M} \langle E_l \rangle$$

we can then define the total variance of these series of measurements as

$$\sigma_M^2 = \frac{1}{M}\sum_{l=1}^{M}(\langle E_l \rangle - \langle E_M \rangle)^2$$

and this can be brought down into

$$\sigma_M^2 = \frac{\sigma_E^2}{N} + \text{covariance} \tag{14}$$

where σ_E^2 is the sample variance over all the different measurements, defined as

$$\sigma_E^2 = \frac{1}{MN^2}\sum_{l=1}^{M}\sum_{i=1}^{N}(E_{l,i} - \langle E_M\rangle)^2 \tag{15}$$

and the covariance is the correlation between the data points, and it is defined as

$$\text{covariance} = \frac{2}{MN^2}\sum_{l=1}^{M}\sum_{i<j}^{N}(E_{l,i} - \langle E_M\rangle)(E_{l,j} - \langle E_M\rangle) \tag{16}$$

However, when there are no correlation in the data set, that makes life easier, because then the variance can be roughly approximated to

$$\sigma_M^2 \approx \frac{1}{N}\Big(\langle E^2\rangle - \langle E\rangle^2\Big) = \frac{\sigma_E^2}{N} \tag{17}$$

where we don't have to evaluate the covariance which is a double sum. Then it is easy to see that the standard deviation is

$$\sigma_M = \sqrt{\sigma_M^2} = \sqrt{\frac{\sigma_E^2}{N}} = \frac{\sigma_E}{\sqrt{N}}.$$

From this it is possible to see that the standard deviation is proportional to the inverse square root of the amount of measurements

$$\sigma_M \sim \frac{1}{\sqrt{N}}.$$

In such a situation where we sample infinitely many times, $N \to \infty$, the standard deviation would go towards zero and the calculation of the energy would in theory converge to the exact answer.

In the case where there exists correlation in the data set, the covariance needs to be calculated, or else the estimation of the variance will be highly underestimated, which leads to an over-optimistic estimation of the uncertainty σ_E.

2.10 Markov Chains

When studying a physical system which evolves towards equilibrium and we want to understand how the system evolves with time, then Markov chains is the preferred way of accomplishing this.

Markov processes is a random walk with a selected probability for making a move. The new move is independent of the previous history of the system. The Markov process is used repeatedly in Monte Carlo simulations in order to generate new random states. The reason for choosing a Markov process is that when it is run for a long enough time, starting with a random state, we will eventually reach the most likely state of the system.

For the Markov processes, the time development of a systems PDF, is defined as

$$w_i(t+1) = \sum_j W_{i\to j} w_j(t)$$

or in matrix form

$$\hat{w}(t+1) = \hat{W}\hat{w}(t)$$

where $w_i(t)$ is the time dependent PDF of the state i, $W_{i\to j}$ is the transition probability of going from state j to i and both quantities are normalized. If we have that $\hat{w}(t=\infty) = \hat{W}\hat{w}(t=\infty)$, we say that we have reached the most likely state of the system, the so-called steady state also referred to the equilibrium state.

By rewriting the transition probability $W_{i\to j}$ into a product of two probabilities

$$W_{i\to j} = T_{i\to j} A_{i\to j}$$

where $T_{i\to j}$ is the probability for making the transition from state j to state i (suggesting moves), and $A_{i\to j}$ is the probability for accepting the proposed move from state j to the state i (accepting or rejecting moves).

We can then express the time development w_i as

$$w_i(t+1) = \sum_j \Big[w_j(t) T_{i\to j} A_{i\to j} + w_i(t) T_{j\to i} \big(1 - A_{j\to i}\big) \Big]$$

assuming that T and A are time-independent.

By utilizing that all the probabilities are normalised (as mentioned above), one can rewrite the equation above as

$$w_i(t+1) = w_i(t) + \sum_j \Big[w_j(t) T_{i\to j} A_{i\to j} - w_i(t) T_{j\to i} A_{j\to i} \Big]$$

$$w_i(t+1) - w_i(t) = \Big[w_j(t) T_{i\to j} A_{i\to j} - w_i(t) T_{j\to i} A_{j\to i} \Big]$$

In the limit $t \to \infty$, it can be shown that

$$w_i(t+1) = w_i \quad \text{and} \quad w_i(t) = w_i$$

which leads to

$$\sum_j w_j(t) T_{i\to j} A_{i\to j} = \sum_j w_i(t) T_{j\to i} A_{j\to i}$$

However, the condition that the rates should equal each other is in general not sufficient to guarantee that we, after many simulations, generate the correct distribution. We may risk to end

up with so-called cyclic solutions. To avoid this issue, one introduces an additional condition, namely that of detailed balance [2].

$$\begin{aligned} W_{i\to j}w_j &= W_{j\to i}w_i \\ w_j(t)T_{i\to j}A_{i\to j} &= w_i(t)T_{j\to i}A_{j\to i} \\ \frac{T_{i\to j}A_{i\to j}}{T_{j\to i}A_{j\to i}} &= \frac{w_i}{w_j} \end{aligned}$$

3 Methods

3.1 Variational Monte Carlo

In this project we will use the widely known Monte Carlo algorithm to simulate random walks in a given volume space. The procedure is as follows.

Variational Monte Carlo Algorithm

1. Initialise the system at N random position $\mathbf{r} = (\mathbf{r}_1, .., \mathbf{r}_N)$. These points will act as our particles and will serve as the basis of our system.

   We also have to fix the number of Monte Carlo steps and the variational parameter α.

2. Initialise the energy and variance and start the Monte Carlo calculation.

   I Choose a random particle and change its configuration.

   II Use the Metropolis-Hastings algorithm to accept or reject this new configuration.

   III If accepted, we update to the new configuration and update the averages.

3. Finish and compute the final averages.

The direction and amount of change of a particle's configuration is chosen by random, and thus every change will be independent from each other.

3.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm takes advantage of the Markov processes and its task is to sample a normalised probability distribution by a stochastic process. In this project, we do two different implementations where both are dependent on the acceptance criteria below.

$$\frac{A_{i\to j}}{A_{j\to i}} = \frac{w_i T_{j\to i}}{w_j T_{i\to j}} \tag{18}$$

3.2.1 Brute Force Metropolis

The Brute Force Metropolis algorithm is a straight forward implementation of the Metropolis algorithm. This algorithm makes the assumption that the probability of jumping from state i to j is the same as jumping from j to i, $T(ji) = T(ij)$, which leads to

$$\frac{A_{i\to j}}{A_{j\to i}} = \frac{w_i}{w_j}$$

Now we can calculate the probability $|\Psi_T(\mathbf{r})|^2$ given the new configuration, which is governed by a variable step length, and we find the ratio between the new and old position.

$$q(\mathbf{r}_i, \mathbf{r}_{i+1}) = \frac{w(\mathbf{r}_{i+1})}{w(\mathbf{r}_i)} = \frac{|\Psi_T(\mathbf{r}_{i+1})|^2}{|\Psi_T(\mathbf{r}_i)|^2}$$

If the ratio q is bigger than 1, the Metropolis will accept the move, while if its less than 1 it will accept the move if its bigger than a random generated number r from a uniform distribution in the interval $[0, 1]$. However, if it is lower, it will reject the move.

$$\text{New configuration} = \begin{cases} \text{Accept}, & \text{if } q > r \\ \text{Reject}, & \text{if } q \leq r \end{cases}$$

3.2.2 Importance sampling

For a diffusion process characterized by a time-dependent probability density for one particle, the Fokker-Planck equation reads

$$\frac{\partial \mathbf{P}}{\partial t} = D\nabla\Big(\nabla - \mathbf{F}\Big)\mathbf{P}$$

where $\mathbf{F}$ is the drift force given in equation 51 in the appendix, while D is the diffusion coefficient, which is 0.5 in natural units for our case. The new positions in space are found by the Langevin equation using Euler's method,

$$\frac{\partial \mathbf{r}}{\partial t} = D\mathbf{F} + \boldsymbol{\eta}$$

where $\boldsymbol{\eta}$ is a random variable for every dimension. This yields a new position

$$\mathbf{r}_{i+1} = \mathbf{r}_i + D\mathbf{F}\Delta t + \boldsymbol{\xi}\sqrt{\Delta t}$$

where $\boldsymbol{\xi}$ is a random variable from a normal distribution in the interval $[0, 1]$ for each dimension and Δt is an adjustable time step. The drift force (also known as the quantum force) gives which direction the particles are moving towards, which the new suggestion of position are dependent on. This can be considered an improvement compared to the brute force algorithm, where a new proposed configuration is not dependent on where the particles are and thus will have the same probability of moving in every direction.

The Fokker-Planck equation yields a transition probability given by Green's function

$$G(\mathbf{r}_{i+1}, \mathbf{r}_i, \Delta t) = \frac{1}{(4\pi D\Delta t)^{3/2}} \exp\Big(- (\mathbf{r}_{i+1} - \mathbf{r}_i - D\Delta t\mathbf{F})^2/4D\Delta t\Big) = T_{i\to j}$$

which in turn means that we get a new acceptance criteria.

$$q(\mathbf{r}_{i+1}, \mathbf{r}_i, \Delta t) = \frac{G(\mathbf{r}_i, \mathbf{r}_{i+1}, \Delta t)|\Psi_T(\mathbf{r}_{i+1})|^2}{G(\mathbf{r}_{i+1}, \mathbf{r}_i, \Delta t)|\Psi_T(\mathbf{r}_i)|^2}$$

3.3 Minimization methods

There exists a legion of challenges in the computational science world when it comes to reducing computational time and effort. How to efficiently find optimal parameters for different systems are undoubtedly one of the, if not the most, demanded solution to scientists in the field of this study. Gradient descent is one of the methods that can cause salvation for scientists for some systems, while just as easily exponentially increase the frustration for others since the range of parameters can range from a few to several thousands in complicated systems. In our system we are fortunate enough to have only one variational parameter α with a good initial understanding of a plausible interval.

3.3.1 Gradient Descent

The basic idea behind is that a function decreases fastest in the negative direction of the gradient of the same function. We want to minimise the ground state energy with respect to the variational parameter α by using its former value to suggest the next value α^+, as shown in the iterative formula

$$\alpha^+ = \alpha - \lambda \frac{d\langle E(\alpha)\rangle}{d\alpha}$$

where $\lambda > 0$ is the step length, which is also known as learning rate in the machine learning world. We already know the expression for the local energy as

$$\langle E(\alpha)\rangle = \frac{\langle \Psi_T(\alpha)|H|\Psi_T(\alpha)\rangle}{\langle \Psi_T(\alpha)|\Psi_T(\alpha)\rangle}$$

and then we differentiate with respect to α by using the chain rule and the hermiticity of the Hamiltonian and arrive at the expression

$$\frac{dE(\alpha)}{d\alpha} = 2\Big\langle E_L(\alpha)\frac{1}{\Psi_T(\alpha)}\frac{d\Psi_T}{d\alpha}\Big\rangle - 2\big\langle E_L(\alpha)\big\rangle\Big\langle \frac{1}{\Psi_T(\alpha)}\frac{d\Psi_T}{d\alpha}\Big\rangle$$

As this is an iterative parameter tuning, one must also take an iterative approach when to stop iterating. We stop iterating as the difference in the calculated energies are less than a given tolerance, or after a given number of iterations as the worlds most powerful machineries are far out of reach for this humble project.

Ideally, we would like the sequence to converge towards the global minimum, however, we do not know if we have encountered a global or local minimum. If the function that is being minimized is convex, we know that every local minimum must also be a global minimum. In addition, gradient descent is deterministic, which means that it will converge to a local minima, if we do not have a good initial guess. Another drawback is that the algorithm is very sensitive to the step length λ. If it is too low, the algorithm will converge very slowly and the computational effort will be too big to defend its use, while if it is too large it might not find the minima at all.

3.4 Statistical analysis methods

Monte Carlo is a method which makes a great foundation for sampling a vast amount of data with a strong correlation between the data points for the interacting system. Distressingly, this

will make the statistical analysis and the calculation of the covariance in equation 16 impractical considering the double summation. Consequently, the tools that will be equipped for post analysis must meet this requirement level. As luck would have it, there exists a method named blocking which can be of aid in these times.

3.4.1 Blocking

Blocking is a technique which gets increasingly accurate with a larger amount of observable data. With the stationary time series $\mathbf{X} = \{x_1, .., x_n\}$, which is our data set where it is assumed that $n = 2^d$ for some integer $d > 1$, the idea behind blocking is a blocking transformation, where we take the mean of subsequent pair of elements from the data set and form a new vector. This is done recursively as

$$\mathbf{X}'_i = \frac{1}{2}\Big(\mathbf{X}_{2i} + \mathbf{X}_{2i+1}\Big)$$

where i is the iteration of the elements in the previous vector. Each of the new vectors are what we would call a block, and the transformed data set will have $n' = n/2$ data points. Then we calculate the variance and covariance for every new time series until we have gone through all of the data. When this process is repeated, the total variance will stagnate until it is not increased any more. Then we have reached our destination, where the blocks are no longer correlated and we can use the standard deviation as a good estimate for the error. For more in-depth derivations we can recommend [3].

4 Implementation

Programs used in this project can be found on our github repository [4]. Here it is included a Jupyter Notebook for easy and reproducable results, and the technicalities included in visualizing data from the variational Monte Carlo simulations.

The code skeleton was initialized by Morten Ledum [5], and in addition the programs are inspired by the codes found in the course's github repository [2].

5 Results

5.1 Non-interacting spherical trap

5.1.1 Benchmark

First of all we are giving our attention to the situation with only one particle in three dimensions. In search for the energy minimum we tried various α values, and the optimal α value was 0.5, as illustrated in figure 1 below.

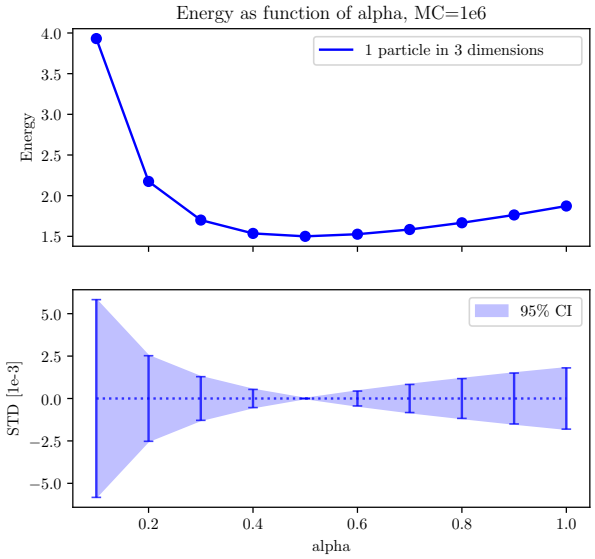


Figure 1: For the standard metropolis sampling we achieve an exact answer for $\alpha = 0.5$ as $E = 1.5$, as we would also expect from the analytical solution. However, we also see in this figure that the standard deviation will increase in a higher rate with decreasing alpha compared to an increasing alpha, showing the importance of not choosing a too small α.

5.1.2 Numerical and analytical

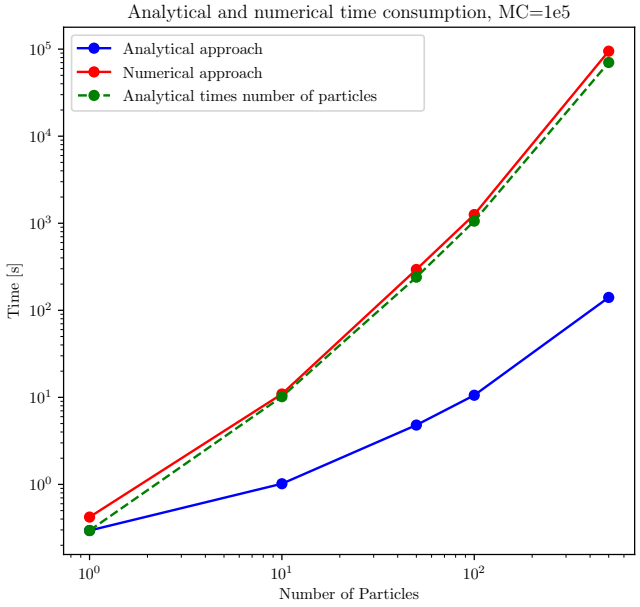
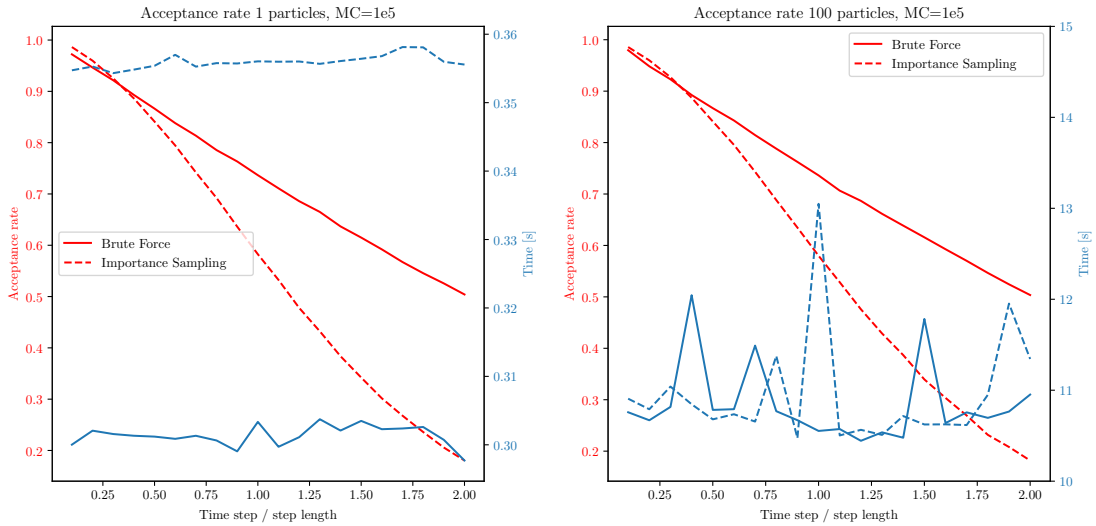


Figure 2: This figure illustrates the CPU consumption of the variational Monte Carlo given different number of particles, all calculated in three dimensions and with the optimisation flag 'O3' which introduces vectorization. As seen, the numerical calculations are highly dependent on the number of particles in the system.

	Numerical		**Analytical**		**Exact**
Particles	Energy	STD	Energy	STD	Energy
1	1.4999	$1.0 \cdot 10^{-4}$	1.50	$9.42 \cdot 10^{-11}$	1.50
10	14.9999	$1.0 \cdot 10^{-4}$	15.0	$5.33 \cdot 10^{-10}$	15.0
50	74.9995	$1.0 \cdot 10^{-4}$	75.0	$3.02 \cdot 10^{-9}$	75.0
100	149.9990	$1.01 \cdot 10^{-4}$	150.0	$6.03 \cdot 10^{-9}$	150.0
500	749.9946	$1.05 \cdot 10^{-4}$	750.0	$3.41 \cdot 10^{-8}$	750.0

Table 1: The numerical and analytical energies and their standard deviations compared to the exact energy given a non-interacting, spherical trap in three dimensions after computing 10^5 brute force Monte Carlo cycles. The main contribute to the numerical standard deviation is the step size from the finite finite difference scheme, as the error scales as h^2, where it has been used $h = 10^{-2}$, nonetheless it is possible to see that the standard deviation increases with the number of particles for both the numerical and the analytical method.

5.1.3 Importance sampling and brute force



(a) (b)

Figure 3: The acceptance rate and the CPU time given a non-interacting, spherical trap for brute force and importance sampling for (a) 1 particles and (b) 100 particles. The acceptance rate is only dependent on time steps, while the time is solely dependent on the number of particles.

5.2 Repulsive interaction in an elliptical trap

5.2.1 Finding variational parameter α

Particles	**10**		**50**		**100**	
α	Energy	STD	Energy	STD	Energy	STD
0.3	26.62	$4.90 \cdot 10^{-2}$	130.46	$7.55 \cdot 10^{-1}$	267.12	$1.40 \cdot 10^{-0}$
0.4	24.49	$1.10 \cdot 10^{-2}$	123.60	$1.35 \cdot 10^{-1}$	249.79	$3.63 \cdot 10^{-1}$
0.5	24.13	$1.04 \cdot 10^{-3}$	121.53	$4.36 \cdot 10^{-3}$	245.04	$3.96 \cdot 10^{-3}$
0.6	24.59	$2.20 \cdot 10^{-2}$	122.92	$7.51 \cdot 10^{-2}$	251.99	$1.88 \cdot 10^{-1}$
0.7	25.57	$2.87 \cdot 10^{-2}$	126.72	$6.91 \cdot 10^{-2}$	257.58	$3.30 \cdot 10^{-1}$

Table 2: This table represents 15 different variational Monte Carlo simulations with varying particles and the variational parameter α. There exists repulsive interaction between the bosons and the potential is elliptic. The standard deviation has been calculated using the blocking method, and the first 25% of 2^{14} Monte Carlo cycles has been used as an equilibrium phase.

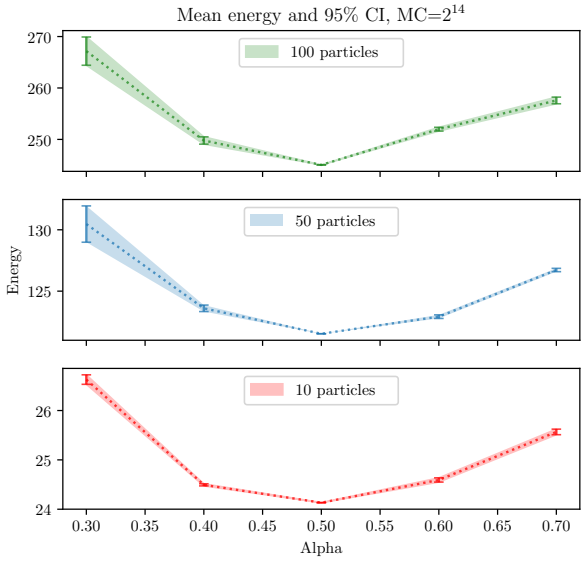


Figure 4: Visualisation of table 2, where we have used the 95% confidence interval as the standard. By increasing the number of particles, the confidence interval will also increase substantially, yet the shape stays the same. From all values of α it is clear that a minimum is met at $\alpha = 0.5$.

5.2.2 Gradient Descent

Particles	α	Mean energy	STD	Iterations	Learning rate
2	0.503587	4.817	$7.10 \cdot 10^{-4}$	7	$1 \cdot 10^{-2}$
10	0.501282	24.152	$1.42 \cdot 10^{-3}$	8	$1 \cdot 10^{-2}$
50	0.502029	121.678	$7.69 \cdot 10^{-3}$	6	$1 \cdot 10^{-3}$
100	0.502988	245.316	$3.80 \cdot 10^{-3}$	23	$1 \cdot 10^{-4}$

Table 3: Given the last iteration in figure 5, this table gives an estimate of the mean energy and standard deviation calculated with the blocking method from 2^{10} Monte Carlo cycles. The tolerance for acceptance of α was set as $\Delta E < 10^{-2}$, where the change is found from the last Monte Carlo cycle.

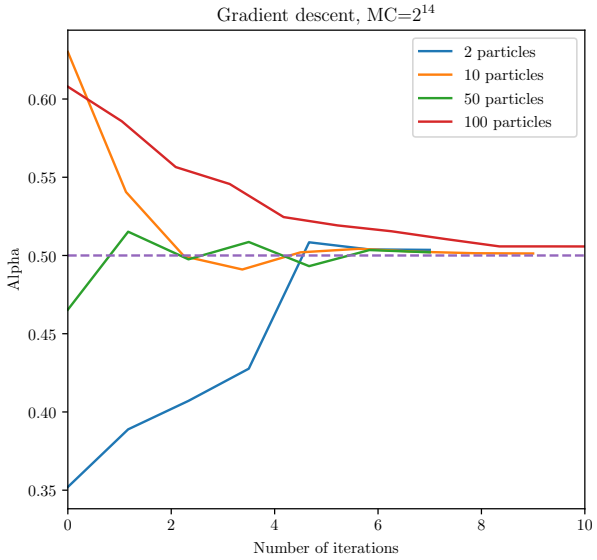


Figure 5: Since we already now know where the approximate interval α should be, we make a qualified guess and chose the starting value of α to be a random normal distributed number in the interval $[0.3, 0.7]$. This qualified guess made us reach convergence within 10 iterations for every set of particle except 100 particles.

5.3 One-body density

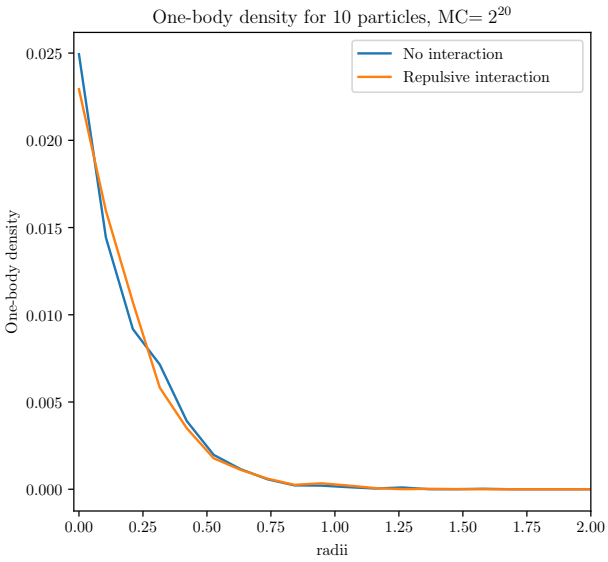


Figure 6: Radial one-body density computed for one system with and without the Jastrow factor for an elliptical trap. The two plots are found by dividing the volume of the system into bins, and thereby calculating the number of particles within each bin. There are in total 20 bins in equal size dividing up the volume interval $\mathbf{r} \in [0, 2]$.

6 Discussion

For the case without interaction and an external potential V_{ext} with a spherical harmonic oscillator, we observe from figure 1 that the minimum of the energy is found when the variational parameter $\alpha = 0.5$, which gives the true ground state of the system. We also observe from figure 1 that $\alpha = 0.5$ gives the minimal confidence interval of the energy.

Table 1 shows the results from the brute force Monte Carlo calculation with a spherical harmonic oscillator as an external potential and no interaction between the particles. We observe that the analytical expression for E gives the exact answer, as shown in equation 11. However, the numerically calculated E, gives an approximate that is very close to the exact answer. This is because of the truncation error in the approximation of the second derivative of the kinetic energy.

Furthermore, by comparing the CPU time difference between the analytical and numerical calculation of E, we clearly see that the numerical calculation is much more computational heavy compared to the analytical one. In figure 2 we observe the CPU time difference of the numerical calculation compared to the analytical one, and we see that the CPU time difference is approximately the number of particles in the system times the CPU time of the analytical calculation. The largest CPU time difference is observed for the system for 500 particles, in three dimensions, and we observe a factor of 10^3 for that instance, which makes it abundantly clear that the analytical calculation is the preferred one. The reason for this huge difference is must probably due to the numerical differentiation mentioned earlier, which is an added heavy numerical computation.

As we move on to figure 3, we can clearly see that the acceptance rate of Metropolis-Hastings is highly dependent on the time step, much more so then the brute force Metropolis is dependent on the step length. Thus, by choosing a small time step or step length will result in accepting roughly all new suggested states. This will, however, give a slow convergence so we choose to compromise with a step length equal to 1. Further on, we observe in figure 3a the difference in CPU consumption for brute force and importance sampling. We would expect that the CPU-difference would increase with particles in the gas. Interestingly, the time for both algorithms appears to use the same amount of time for 100 particles. This could indicate that it takes longer time for the importance sampled program to initialise and start up, while brute force on the other hand is quite efficient for this task. We can only speculate to give an explanation, but it is indeed known that initialising for-loops requires a small (but substantial) start-up time and may point towards that the importance sampled metropolis algorithm are using more for-loops than the brute force metropolis, but in return are more efficient. This could be a result of our optimization flag, which uses vectorization and can take a better advantage of the prior.

The results presented in table 2 and figure 4 shows when systems with different number of particles finds the optimal variational parameter α. We observe that the optimal variational parameter $\alpha = 0.5$ for all the systems. We also observe that the ground state energy for 10 and 100 particles differs from the case without interaction, and this is due to the implementation of the repulsive interaction and the external potential changing from a spherical harmonic oscillator to a elliptical harmonic oscillator.

Instead of manually testing different values of α to find the ground state, we used the gradient descent method to find the optimal α value. Figure 5 and table 3 shows the results of that

endeavour. We observe that the results from table 2 and figure 4, obtained when running VMC for various α's, match the results obtained when running VMC using gradient descent, where $\alpha \approx 0.5$.

An estimate of the radial one-body density is given in figure 6. We observe that for both including and excluding the Jastrow factor, which is bound to be correlated with a repulsive potential, the shape does not drastically change. Be that as it may, by looking closer at the graph one can see that the system with no interaction has a lower peak at the interval $\mathbf{r} \in [0, 0.05]$, which is the very first measurements. In addition, one can tell that the repulsive interaction distribution is in fact the same or a tiny bit wider compared to a system with no interaction at all. Nonetheless, we foresaw a bigger alteration of the curve compared to the result.

7 Conclusion

In this project, we have implemented the VMC method for solving the ground state of bosons in both a non-interacting system where we used a spherical harmonic oscillator as the external potential V_{ext} and an interacting system where V_{ext} was an elliptical harmonic oscillator.

The ground state energy for the non-interacting case was found to be $\alpha = 0.5$. To calculate the energy of the system we implemented both an analytical expression and a numerical one for the local energy E. It was discovered that the numerical calculation of E was much slower than the analytical calculation.

We also implemented two methods for proposing moves in the Monte Carlo algorithm. Those methods were the brute force Metropolis method and the importance sampling of Metropolis-Hastings. We observed a difference in the behaviour of the acceptance rate of the two methods, where the Metropolis-Hastings method was more dependent on the time step than the brute force Metropolis was on the step size. In addition, we also saw little to no difference in CPU time, between the two methods.

By including the repulsive interaction, the approximated way of calculating the statistical error shown in equation 17 would not suffice, because by having interaction we now had correlation in our data. Thus, we used the blocking method to estimate the statistical error, which included an estimation of the covariance of the data. After introducing interaction, a new search for the optimal variational parameter α had begun, and it was found to be $\alpha = 0.5$ for both when we manually tested different α-values and when we used gradient descent.

We also used our VMC program to study the one-body density of a 10-particle system. We observed a slight difference between the one-body density of the system with no interaction and the system with interaction, where the interacting case moves to a larger radii, as theory suggests.

8 Future aspects

As we look back at this project, we see a lot of dedication, hard work and intense linear algebra-scribbling. Since this project is part of a university subject, we have had teaching assistants

providing help in times of need, as well as having a deadline for delivery. In retrospect we do acknowledge that we should[2] swallow our pride and ask for help faster instead of being stuck at a tiny and unimportant barrier for a longer time. This could reduce the time spent on the project by an embarrassingly big deal.

And since time is the current subject, we are also experiencing some limiting factors such as an on-going pandemic, closed institutions and bound social distancing. To take necessary precautions for this has been experienced as difficult, since communication and efficiency has been halted. Either way, since we have now experienced it once, we are now very much ready if this situation will show itself again in the future.

Furthermore, for future improvements it would be interesting to use more advanced computational resources to investigate the system with a larger Monte Carlo iteration and study the results.

[2] some times

Appendices

A Non-Interacting System

A.1 Local Energy - $\mathbf{E}_L$

The local energy is defined as

$$E_L = \frac{1}{\Psi_T(\boldsymbol{r})} H\Psi_T(\boldsymbol{r}), \quad H = \sum_{i=1}^{N} \Big(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\boldsymbol{r}_i)\Big) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{19}$$

where H is the Hamiltonian for the system, V_{ext} is the external potential and V_{int} is the interactive potential between the particles.

In this case the Hamiltonian shortens down into

$$H = \sum_{i=1}^{N} \Big(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\boldsymbol{r}_i)\Big) \tag{20}$$

We will use the harmonical potential as V_{ext}, that means

$$V_{ext}(\boldsymbol{r}) = \frac{1}{2} m\omega^2 r, \quad r = \sqrt{x^2 + y^2 + z^2} \tag{21}$$

The trial wave function for the ground state with N atoms is given by

$$\Psi_T(\boldsymbol{r}) = \Psi_T(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) = \Big[\prod_i g(\alpha, \beta, \boldsymbol{r}_i)\Big]\Big[\prod_{j<k} f(a, |\boldsymbol{r}_j - \boldsymbol{r}_k|)\Big] \tag{22}$$

where α and β are variational parameters. The single-particle wave function is given as

$$g(\alpha, \beta, \boldsymbol{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)] \tag{23}$$

The correlation wave function is

$$f(a, |\boldsymbol{r}_j - \boldsymbol{r}_k|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{24}$$

A.1.1 1 particle in 1D

The trial wave function in 1D with the case of using only the harmonic oscillator potential $a = 0$ and for a spherical trap can be found from equation 22

$$\Psi_T(x, \alpha, \beta = 1) = \exp[-\alpha x^2]$$

Since there is only one particle in this case, we do (of course) not include any interactive potential, and we will use the Hamiltonian given in 20.

$$E_L = \frac{1}{\Psi_T(\boldsymbol{x})}\Big(-\frac{\hbar^2}{2m}\nabla^2\Psi_T(\boldsymbol{x}) + V_{ext}(\boldsymbol{x})\Psi_T(\boldsymbol{x})\Big)$$

where the double derivative of the trial function is found as

$$\begin{aligned}\nabla^2\Psi_T(\boldsymbol{x}) &= (-2\alpha + 4\alpha^2 x^2)\exp[-\alpha x^2] \\ &= (-2\alpha + 4\alpha^2 x^2)\Psi_T(\boldsymbol{x})\end{aligned}$$

and the spherical external potential is found from 21. The resulting local energy is

$$\begin{aligned}E_L &= \frac{1}{\Psi_T(\boldsymbol{x})}\Big(-\frac{\hbar^2}{2m}(-2\alpha + 4\alpha^2 x^2)\Psi_T(\boldsymbol{x}) + \frac{1}{2}m\omega_{ho}^2 r^2\Psi_T(\boldsymbol{x})\Big) \\ &= -\frac{\hbar^2}{2m}(-2\alpha + 4\alpha^2 x^2) + \frac{1}{2}m\omega_{ho}^2 x^2\end{aligned}$$

To make this expression tidy and smart we will make it dimensionless by saying that ω_{ho}, $\hbar$ and m equals to 1, as shown below.

$$E_L = \alpha - 2\alpha^2 x^2 + \frac{1}{2}x^2$$

This can be finalised into the nice expression

$$E_L = \alpha + (\frac{1}{2} - 2\alpha^2)x^2 \tag{25}$$

A.1.2 N-particles in 1D

This time we would like to find the analytical solution to N particles in 1 dimension. Luckily, this is quite analogous to the one particle case.

$$\begin{aligned}E_L &= \frac{1}{\Psi_T(\boldsymbol{x})}\sum_{i=1}^{N}\Big(-\frac{\hbar^2}{2m}\nabla_i^2\Psi_T(\boldsymbol{x}) + V_{ext}(\boldsymbol{x}_i)\Psi_T(\boldsymbol{x})\Big) \\ &= \sum_{i=1}^{N}\Big(-\frac{\hbar^2}{2m}(-2\alpha + 4\alpha^2 x_i^2) + \frac{1}{2}m\omega_{ho}^2 x_i^2\Big) \\ &= \frac{\hbar^2}{m}\alpha N + \Big(\frac{1}{2}m\omega_{ho}^2 - \frac{2\hbar\alpha^2}{m}\Big)\sum_{i=1}^{N}x_i^2\end{aligned}$$

Yet again, this expression can be made as neat as a new pin by making it dimensionless.

$$E_L = N\alpha + \Big(\frac{1}{2} - 2\alpha^2\Big)\sum_{i=1}^{N}x_i^2 \tag{26}$$

A.1.3 1 particle in 3D

The trial wavefunction for one particle in three dimensions is given by

$$\Psi_T(\boldsymbol{r}) = e^{-\alpha r^2} \tag{27}$$

We start from the local energy equation shown in Eq. 19, where we need to calculate the second derivative. For this case it might be better to use spherical coordinates, where the Laplace operator is given by

$$\nabla^2 = \frac{1}{r^2}\frac{\partial}{\partial r}\Big(r^2\frac{\partial}{\partial r}\Big) \tag{28}$$

where we only take the radial part into account, since we are working on a spherical symmetrical potential.

$$\frac{\partial}{\partial r}\Psi_T(\boldsymbol{r}) = \frac{\partial}{\partial r}e^{-\alpha r^2} = -2\alpha r e^{-\alpha r^2} = -2\alpha r\Psi_T(\boldsymbol{r})$$
$$\frac{\partial}{\partial r}\Big(-r^2\times 2\alpha r\Psi_T(\boldsymbol{r})\Big) = \frac{\partial}{\partial r}\Big(-2\alpha r^3 e^{-\alpha r^2}\Big)$$

To solve the second derivative, we need to use the product rule of derivation $(u\times v)' = u'v + uv'$.

$$\frac{1}{r^2}\frac{\partial}{\partial r}\Big(-2\alpha r^3 e^{-\alpha r^2}\Big) = \frac{1}{r^2}\Big(-6\alpha r^2 e^{-\alpha r^2} + 4\alpha^2 r^4 e^{-\alpha r^2}\Big) = -6\alpha e^{-\alpha r^2} + 4\alpha^2 r^2 e^{-\alpha r^2}$$
$$\nabla^2\Psi_T(\boldsymbol{r}) = \Big(-6\alpha + 4\alpha^2 r^2\Big)\Psi_T(\boldsymbol{r})$$

With the expression for the second derivative calculated we can now find the local energy.

$$\begin{aligned} E_L = \frac{1}{\Psi_T(\boldsymbol{r})}H\Psi_T(\boldsymbol{r}) &= \frac{1}{\Psi_T(\boldsymbol{r})}\Big(-\frac{\hbar^2}{2m}\nabla^2 + \frac{1}{2}m\omega^2 r^2\Big)\Psi_T(\boldsymbol{r}) \\ &= \frac{1}{\Psi_T(\boldsymbol{r})}\Big(-\frac{\hbar^2}{2m}\Big(-6\alpha + 4\alpha^2 r^2\Big)\Psi_T(\boldsymbol{r}) + \frac{1}{2}m\omega^2 r^2\Psi_T(\boldsymbol{r})\Big) \\ &= \frac{\hbar^2}{m}\Big(3\alpha - 2\alpha^2 r^2\Big) + \frac{1}{2}m\omega^2 r^2 \end{aligned}$$

$$E_L = \frac{\hbar^2}{m}\Big(3\alpha - 2\alpha^2 r^2\Big) + \frac{1}{2}m\omega^2 r^2 \tag{29}$$

And by making it dimensionless by utilizing natural units, the local energy is reduced to

$$E_L = 3\alpha - \Big(\frac{1}{2} - 2\alpha^2\Big)r^2 \tag{30}$$
$$= dim\times\alpha - \Big(\frac{1}{2} - 2\alpha^2\Big)r^2 \tag{31}$$

A.1.4 N-particles in 3D

To extend the non interaction case to N particles in three dimensions, is pretty straight forward, where you only need to sum up the expression for the one particle in three dimensions.

$$E_L = \sum_{i=1}^{N}\frac{\hbar^2}{m}\Big(3\alpha - 2\alpha^2 r_i^2\Big) + \frac{1}{2}m\omega^2 r_i^2 = 3N\alpha\frac{\hbar^2}{m} + \Big(\frac{1}{2}m\omega^2 - 2\alpha^2\frac{\hbar^2}{m}\Big)\sum_{i=1}^{N} r_i^2 \tag{32}$$

And in natural units the expression for the local energy becomes much more nicer

$$E_L = dim\times N\times\alpha + \Big(\frac{1}{2} - 2\alpha^2\Big)\sum_{i=1}^{N} r_i^2. \tag{33}$$

A.2 Drift force - F

In this section we will find the drift force and the expression for the drift force is

$$F = \frac{2\nabla\Psi_T}{\Psi_T}, \quad \Psi_T(\boldsymbol{r}) = e^{-\alpha r^2} \tag{34}$$

A.2.1 1-particle in 1D

The trial wavefunction for one particle in one dimension is given by

$$\Psi_T(x) = e^{-\alpha x^2}$$

The derivative is given as

$$\nabla\Psi_T(x) = -2\alpha x\Psi_T(x) \tag{35}$$

that means, the drift force is defined as

$$F = \frac{2\nabla\Psi_T}{\Psi_T} = -4\alpha x \tag{36}$$

A.2.2 N-particles in 1D

The trial wavefunction for N-particles in one dimension is given by

$$\Psi_T(x) = \prod_{i=1}^{N} e^{-\alpha x_i^2} = e^{-\alpha\sum_{i=1}^{N} x_i^2} \tag{37}$$

and the drift force for N-particle is just the summation of the expression for the drift force in the one particle case shown above.

$$F = \sum_{i=1}^{N} -4\alpha x_i = -4\alpha\sum_{i=1}^{N} x_i \tag{38}$$

A.2.3 1-particle in 3D

The trial wavefunction for one particle in three dimensions is given by

$$\Psi_T(\boldsymbol{r}) = e^{-\alpha r^2} \tag{39}$$

We start from the drift force equation shown in Eq. 34, where we need to calculate the first derivative.

$$\frac{\partial}{\partial r}\Psi_T(\boldsymbol{r}) = \frac{\partial}{\partial r}e^{-\alpha r^2} = -2\alpha\boldsymbol{r}e^{-\alpha r^2} = -2\alpha\boldsymbol{r}\Psi_T(\boldsymbol{r})$$

that means, the drift force is defined as

$$F = \frac{2\nabla\Psi_T(\boldsymbol{r})}{\Psi_T(\boldsymbol{r})} = -4\alpha\boldsymbol{r} \tag{40}$$

A.2.4 N-particles in 3D

The trial wavefunction for N-particles in three dimension is given by

$$\Psi_T(\boldsymbol{r}) = \prod_{i=1}^{N} e^{-\alpha r_i^2} = e^{-\alpha \sum_{i=1}^{N} r_i^2} \tag{41}$$

and the drift force for N-particle is just the summation of the expression for the drift force in the one particle case shown above.

$$F = \sum_{i=1}^{N} -4\alpha r_i = -4\alpha \sum_{i=1}^{N} r_i \tag{42}$$

B Interacting System

B.1 Local Energy - $\mathbf{E}_L$

The trial wave function for the ground state with N atoms is given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|)\right],$$

and can be rewritten as

$$\Psi_T(\mathbf{r}) = \left[\prod_i \Phi(\boldsymbol{r}_i)\right] \exp\left(\sum_{j<k} u(r_{jk})\right)$$

where we have defined $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$,

$$f(r_{ij}) = \exp\left(\sum_{i<j} u(r_{ij})\right),$$

with $u(r_{ij}) = \ln f(r_{ij})$ and

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right] = \Phi(\boldsymbol{r}_i).$$

To calculate the local energy we need the first and second derivative of the trial function. To derive the first derivative we utilise the product rule of derivation $(u \times v)' = u'v + uv'$.

$$\begin{aligned}\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \Phi(\boldsymbol{r}_k) \left[\prod_{i \neq k} \Phi(\boldsymbol{r}_i)\right] \exp\left(\sum_{j<m} u(r_{jm})\right) \\ + \left[\prod_i \Phi(\boldsymbol{r}_i)\right] \exp\left(\sum_{j<m} u(r_{jm})\right) \sum_{l \neq k} \nabla_k u(r_{kl}),\end{aligned}$$

For the second derivative we just use the product rule of derivation again, and then we end up with the expression

$$\begin{aligned}
\nabla_k^2\Psi_T(\mathbf{r}) = \nabla_k^2\Phi(\boldsymbol{r}_k)\left[\prod_{i\neq k}\Phi(\mathbf{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right) \\
+\nabla_k\Phi(\mathbf{r}_k)\left[\prod_{i\neq k}\Phi(\boldsymbol{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k u(r_{kl}) \\
+\nabla_k\Phi(\boldsymbol{r}_k)\left[\prod_{i\neq k}\Phi(\boldsymbol{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k u(r_{kl}) \\
+\left[\prod_{i}\Phi(\boldsymbol{r}_i)\right]\left\{\exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k u(r_{kl})\sum_{l\neq k}\nabla_k u(r_{kl}) + \exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k^2 u(r_{kl})\right\}
\end{aligned}$$

To simplifiy this expression, we need to do a change of variabels

$$\frac{\partial}{\partial \boldsymbol{r}_k} = \frac{\partial}{\partial \boldsymbol{r}_k}\frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \boldsymbol{r}_k}\frac{\partial}{\partial r_{kj}} = \frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{r_{kj}}\frac{\partial}{\partial r_{kj}} \tag{43}$$

where we have used that

$$\frac{\partial r_{kj}}{\partial \boldsymbol{r}_k} = \frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{|\boldsymbol{r}_k-\boldsymbol{r}_j|} = \frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{r_{kj}} \tag{44}$$

In addition we need to differentiate $u(r_{ij})$

$$\nabla_k u(r_{ij}) = \frac{\partial r_{kj}}{\partial \boldsymbol{r}_k}\frac{\partial}{\partial r_{kj}}(u(r)) = \frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{|\boldsymbol{r}_k-\boldsymbol{r}_j|}u'(r_{ij}) \tag{45}$$

$$\nabla_k^2 u(r_{ij}) = \nabla_k\Big(\frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{|\boldsymbol{r}_k-\boldsymbol{r}_j|}u'(r_{ij})\Big) \tag{46}$$

$$= \frac{1}{|\boldsymbol{r}_k-\boldsymbol{r}_j|}\frac{\partial}{\partial r_{kj}}\Big(u(r_{kj})\Big) + \frac{\partial^2}{\partial r_{kj}^2}\Big(u(r_{kj})\Big) + \frac{(\boldsymbol{r}_k-\boldsymbol{r}_j)(\boldsymbol{r}_k-\boldsymbol{r}_j)}{|\boldsymbol{r}_k-\boldsymbol{r}_j|^3}\frac{\partial}{\partial r_{kj}}\Big(u(r_{kj})\Big) \tag{47}$$

$$= \frac{1}{r_{kj}}u'(r_{kj}) + u''(r_{kj}) + \frac{|\boldsymbol{r}_k-\boldsymbol{r}_j|^2}{|\boldsymbol{r}_k-\boldsymbol{r}_j|^3}u'(r_{kj}) \tag{48}$$

$$= \frac{1}{r_{kj}}u'(r_{kj}) + u''(r_{kj}) + \frac{1}{r_{kj}}u'(r_{kj}) \tag{49}$$

$$= \frac{2}{r_{kj}}u'(r_{kj}) + u''(r_{kj}) \tag{50}$$

Finally we then have an expression needed in the calculation for the local energy.

$$\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\Phi(\mathbf{r}_k)}{\Phi(\mathbf{r}_k)} + 2\frac{\nabla_k\Phi(\mathbf{r}_k)}{\Phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{ij})\right) + \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \sum_{j\neq k}\left(u''(r_{kj})+\frac{2}{r_{kj}}u'(r_{kj})\right).$$

Now we can easily find the local energy expression by utilizing the general expression for it

$$\begin{aligned} E_L &= \frac{1}{\Psi(\boldsymbol{r})}H\Psi(\boldsymbol{r}) \\ &= \frac{1}{\Psi(\boldsymbol{r})}\left[\sum_{i=1}^{N}\Big(-\frac{\nabla_i^2}{2}+V_{ext}(\boldsymbol{r}_i)\Big)+\sum_{i<j}^{N}V_{int}(\boldsymbol{r}_i,\boldsymbol{r}_j)\right]\Psi(\boldsymbol{r}) \\ &= \sum_{i=1}^{N}\Big(-\frac{\nabla_i^2\Psi(\boldsymbol{r})}{2\Psi(\boldsymbol{r})}+V_{ext}(\boldsymbol{r}_i)\Big)+\sum_{i<j}^{N}V_{int}(\boldsymbol{r}_i,\boldsymbol{r}_j). \end{aligned}$$

B.2 Drift force - F

In this section we will find the drift force and the expression for the drift force is

$$F = \frac{2\nabla\Psi_T}{\Psi_T}. \tag{51}$$

The wavefunction is as defined earlier

$$\Psi_T(\mathbf{r}) = \left[\prod_i \Phi(\boldsymbol{r}_i)\right]\exp\left(\sum_{j<k}u(r_{jk})\right).$$

To calculate the drift force we need the derivative of the trial wavefunction. To derive the derivative we utilise the product rule of derivation $(u\times v)' = u'v + uv'$.

$$\nabla_k\Psi_T(\mathbf{r}) = \nabla_k\Phi(\boldsymbol{r}_k)\left[\prod_{i\neq k}\Phi(\boldsymbol{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right) + \left[\prod_i\Phi(\boldsymbol{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k u(r_{kl})$$

Then have the expression which is needed in the calculation for the drift force.

$$\frac{1}{\Psi_T(\mathbf{r})}\nabla_k\Psi_T(\mathbf{r}) = \frac{\nabla_k\Phi(\boldsymbol{r}_k)}{\Phi(\boldsymbol{r}_k)} + \sum_{l\neq k}\nabla_k u(r_{kl})$$

Then the drift force is

$$\begin{aligned} F &= \frac{2}{\Psi_T(\mathbf{r})}\nabla\Psi_T(\mathbf{r}) \\ &= 2\sum_{k=1}^{N}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k\Psi_T(\mathbf{r}) \\ &= 2\sum_{k=1}^{N}\left(\frac{\nabla_k\Phi(\boldsymbol{r}_k)}{\Phi(\boldsymbol{r}_k)} + \sum_{l\neq k}\nabla_k u(r_{kl})\right) \end{aligned}$$

References

[1] Jørgen Høgberget (2013), Quantum Monte-Carlo Studies of Generalized Many-body Systems, Master Thesis. Fysisk Institutt, Universitetet i Oslo.

[2] Morten Hjorth-Jensen, Github Repository,
https://github.com/CompPhysics/ComputationalPhysics2

[3] Marius Jonsson,
Standard error estimation by an automated blocking method,
Phys. Rev. E98, 15.10.2018
https://journals.aps.org/pre/abstract/10.1103/PhysRevE.98.043304

[4] Mohamed Ismail,
Contains all code and figures, as well as a Jupyter notebook for easy reproducibiliy.
https://github.com/Moejay10/FYS4411/tree/master/Project1

[5] Morten Ledum, Github Repository,
Contains the initialised code skeleton used in this project.
https://github.com/mortele/variational-monte-carlo-fys4411