FYS4411 - Computational Physics II

Project 1

Dorthea Gjestvang
Even Marius Nordhagen

March 20, 2018

- Github repository containing programs and results:
  `https://github.com/evenmn/FYS4411/tree/master/Project%201`

Abstract

This project aims to simulate a trapped Bose gas using Variational Monte Carlo (VMC) with both the brute force Metropolis and the Metropolis-Hastings algorithms. Using a trial wave function with either spherical or elliptical harmonic oscillator (HO) potential, with and without a hard sphere potential, we calculate the ground state energy and onebody density for different numbers of particles. The results are benchmarked with earlier publications.

Using the spherical HO with no interaction, both the brute force and the Metropolis-Hastings methods reproduces analytical calculations. Though the Hastings algorithm is slightly slower, it has a higher acceptance rate for the proposed moves in the VMC simulation. The onebody density distribution changes shape when introducing the hard sphere potential. The system behaves as expected, and our results reproduces the benchmarks.

1 Introduction

In this project, we study the Bose-Einstein condensation (BEC) of a dilute gas of ^{87}Rb atoms trapped in an elliptical harmonic oscillator. Bose-Einstein condensation occurs when atoms are cooled towards absolute zero, as they start occupying the same quantum state [5]. Even though the effect was predicted already in 1924, it was first observed by Anderson et.al. in 1995, which caused an immense interest in the subject [6], [7].

Our goal is to find an upper bound limit of the ground state energy for a system of a various number of particles, using an ansatz where the total wave function is a product of single particle wave functions. The Gross-Pitaevskii equation is applied to obtain an analytical energy approximation, which works as a benchmark for the energy. The theoretical framework is presented in section 2.

The main technique we will use is the variational Monte Carlo (VMC) method with two different Metropolis algorithms, presented in section 3. To achieve the optimal trial wave function we vary the variational parameter α using the gradient descent method. In section 5 the upper bound energy is presented with the optimal wave functions and the onebody density. In section 6 we discuss our results, and compare them to results obtained from the articles found at [2], [3] and [4].

2 Theory

2.1 Presentation of potential and trial wavefunction

We study a system of N bosons trapped in a harmonic oscillator with the Hamiltonian given by

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

with V_{ext} as the external potential, which is the harmonic oscillator potential defined in equation 2, and V_{int} as the interaction term, defined in equation 3, which ensures the particles to be separated by a distance a. $\hbar$ is the reduced Plancks constant and m is the mass of the particles in question. In this project we study the gas of ^{87}Rb alkali atoms, which have a scattering length $a_{\text{Rb}} = 0.0043$ given in units of a typical trap size a_{HO}. We will consider a harmonic oscillator which can either be spherical (all dimensions have the same scales) or elliptical (the vertical dimension has a different frequency

from the horizontals).

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

where ω_{HO} is the frequency of the spherical harmonic oscillator and ω_z is the frequency of the elliptical trap in z-direction.

$$V_{int}(\vec{r}_i,\vec{r}_j) = \begin{cases} 0 & \text{if} \quad |\vec{r}_i-\vec{r}_j| \geq a \\ \infty & \text{if} \quad |\vec{r}_i-\vec{r}_j| < a \end{cases} \tag{3}$$

The trial wavefunction is on the form

$$\Psi_T(\vec{r}_1,\vec{r}_2,...,\vec{r}_N,\alpha,\beta) = \prod_i^N g(\alpha,\beta,\vec{r}_i)\prod_{i<j} f(a,r_{ij}) \tag{4}$$

where $r_{ij} = |\vec{r}_i - \vec{r}_j|$ and g is assumed to be a gaussian function,

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

which is practical since

$$\prod_i^N g(\alpha,\beta,\vec{r}_i) = \exp\Big[-\alpha\sum_{i=1}^N(x_i^2+y_i^2+\beta z_i^2)\Big]. \tag{6}$$

α is a variational parameter that we later use to find the energy minimum, and β is a constant. The trial wave function used is the exact ground state wave function for a spherical harmonic oscillator with no interaction, so by choosing the correct α, we will find the exact ground state energy. The α for the ground state wave function in this case is known to be 0.5. The f presented above is the correlation wave function, which is

$$f(a,r_{ij}) = \begin{cases} 0 & r_{ij} \leq a \\ \left(1-\frac{a}{r_{ij}}\right) & r_{ij} > a. \end{cases} \tag{7}$$

where a is a parameter describing the minimum distance allowed between particles in the trap.

2.2 Local energy E_L calculation

In general we can find the energy of a system with a known wave function by solving the integral

$$E = \langle H\rangle = \frac{\int \psi^*(\vec{r})H\psi(\vec{r})d\vec{r}}{\int \psi^*(\vec{r})\psi(\vec{r})d\vec{r}} \tag{8}$$

which depends on $|\psi(\vec{r})|^2$ only. This implies that sampling points from the $|\psi|^2$ distribution corresponds to sampling from the integrand $\psi^* H\psi$. We are in a situation where we do not know the exact wavefunction, and need to sample from the trial wave function in equation 4, which will probably not give us the exact energy. However, as we move the wave function closer to the true ground state wave function using the Metropolis algorithm (see section 3.2), the computed energy (denoted local energy) should approach the ground state energy [8]. The algorithm used to conduct the Monte Carlo integration is presented in section 3.1.

For simplicity we use an expression for E_L obtained from Schrödinger's equation that let us avoid calculating the denominator in equation 8,

$$E_L(\vec{r}) = \frac{1}{\Psi_T(\vec{r})}\hat{H}\Psi_T(\vec{r}). \tag{9}$$

When the repulsive interaction is ignored ($a = 0$), it can be shown that the local energy for a system of N particles and dim spatial dimensions is given by

$$E_L = dim \cdot N \cdot \alpha + \left(\frac{1}{2} - 2\alpha^2\right)\sum_i \vec{r}_i^2, \tag{10}$$

which is proven in appendix A. This is only true for the a spherical harmonic oscillator trap, for the elliptical trap we need to add β in front of the z-component. You may also notice that this equation is scaled, more about that in section 2.4.

For the a spherical harmonic oscillator where particle interactions are ignored, the analytical expression for the energy is well-known and reads $E = \hbar\omega(n + dim/2)$ where n is the energy level and dim is number of spatial dimensions. In this project we will study the ground state only, such that $n = 0$, and for N particles and dim spatial dimensions we therefore obtain the expression for the ground state energy shown in equation 11.

$$E = \frac{1}{2}N \cdot dim \cdot \hbar\omega_{HO}. \tag{11}$$

For $a \neq 0$ it gets rather more complicated, because the Jastrow factor from equation 7 is now different from 1. We also need to add the interaction term V_{int}, the hard-sphere potential from equation 3. We are now ready to find an analytical expression for the local energy. By defining

$$f(a, r_{ij}) = \exp\left(\sum_{i<j} u(r_{ij})\right) \tag{12}$$

and doing a change of variables

$$\frac{\partial}{\partial \vec{r}_k} = \frac{\partial}{\partial \vec{r}_k}\frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \vec{r}_k}\frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}}\frac{\partial}{\partial r_{kj}} \tag{13}$$

one will end up with

$$E_L = \sum_k \Bigg(-\frac{1}{2}\Bigg(4\alpha^2\Big(x_k^2 + y_k^2 + \beta^2 z_k^2 - \frac{1}{\alpha} - \frac{\beta}{2\alpha}\Big) - 4\alpha \sum_{j\neq k}(x_k, y_k, \beta z_k)\frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}}u'(r_{kj})$$
$$+ \sum_{ij\neq k}\frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_i)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \sum_{j\neq k}\Big(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\Big)\Bigg) + V_{ext}(\vec{r}_k)\Bigg) + V_{int}.$$

which is also shown in appendix A. This is not a pretty expression, but will yield correct results for both the interacting and non-inteacting case.

2.2.1 Numerical calculation of E_L

Another approach when calculating E_L is to split up the local energy expression as shown in equation 14, and calculate the local energy with a numerical approach where the second derivative needed to kind the kinetic energy can be approximated by the three-point formula, see equation 15.

$$E_{L,i} = -\frac{\hbar^2}{2m}\frac{\nabla_i^2\Psi_T}{\Psi_T} + V_{ext}(\vec{r}_i) = E_{k,i} + E_{p,i} \tag{14}$$

$$f''(x) \simeq \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \tag{15}$$

In our case the position is a three dimensional vector, so we need to handle each dimension separately. Both the analytical and the numerical local energy are implemented, and in section 5.2, the CPU time for the analytical and numerical approach are compared for various number of particles.

2.3 Onebody density

In many cases it is convenient to know the positions of the particles, but when the number of particles increases, the set of positions turns into a messy collection of numbers which is not really informative (in fact the exact positions can hardly be revealed according to Heisenberg's uncertainity principle). Instead of presenting the positions, the density of particles can give us a good overview of where the particles are located. With N particles, the one-body

density with respect to a particle i is an integral over all particles but particle i:

$$\rho_i = \int_{-\infty}^{\infty} d\vec{r}_1 \dots d\vec{r}_{i-1} d\vec{r}_{i+1} \dots d\vec{r}_N |\Psi(\vec{r}_1, \dots \vec{r}_N)|^2. \tag{16}$$

For the non-interacting case this integral can be solved analytically, as shown in equation 17. Using Monte Carlo integration, the one-body density can be solved for any case. Anyhow, the interesting part is the radial density, so we either have to solve the integral in spherical coordinates or convert to spherical coordinates afterwards.

Alternatively, the onebody radial density can be found in a more intuitive way. Imagine we divide the volume around particle i into bins, where bin j is located at a distance $j \cdot r_1$, as shown in figure 1. The radii are thus quantized. By counting the number of particles in a bin and dividing by the surface area, we find the average density of particles in the bin. If we decrease the initial radius r_1 of the innermost bin such that we have a large number of bins, this method can be used to find the onebody density.

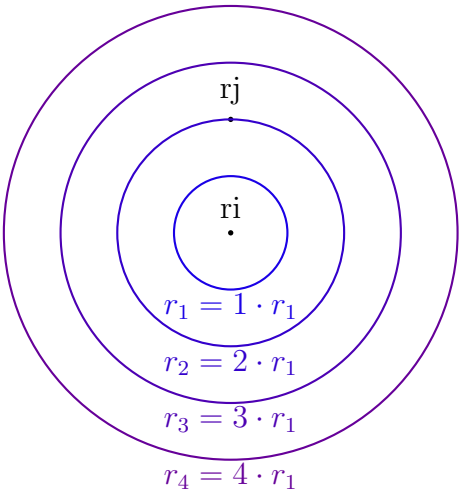

Figure 1: One can find the onebody density by dividing the volume around a particle with coordinates $\vec{r}_i$ into bins and then count the number of particles, here illustrated in two dimensions.

The analytical expression without the Jastrow factor is found from [2], and can be modified for our particular system, presented for the three dimensional case in equation 17.

$$\rho_i = \frac{N\alpha\beta^{1/2}}{8\pi^{3/2}} e^{-r_i^2}. \tag{17}$$

Since the particles are identical the choice of the particle i is of no consequence.

2.4 Scaling

For big numerical projects, working with dimensionless quantities is a great advantage. Not only does it improve the code structure and performance, but it also avoids truncation errors due to small constants. For this project a natural scaling parameter for the energy is $\hbar\omega_{HO}$, which appears in the analytical energy expression in equation 11. The equivalent dimensionless equation can then be written as

$$E' = \frac{N \cdot dim}{2} \tag{18}$$

where $E' = E/\hbar\omega_{HO}$. Additionally, we can scale the position with respect to the length of the spherical trap, a_{HO}, such that

$$r_i' = \frac{r_i}{a_{HO}} = r_i \cdot \sqrt{\frac{m\omega_{HO}}{\hbar}}, \tag{19}$$

and the Hamiltonian turns into

$$H = \frac{1}{2}\sum_i \Big(-\nabla^2 + \vec{r}_i^2 \Big) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{20}$$

A watchful eye will see that this corresponds to setting $\hbar = \omega_{HO} = m = 1$, which is the natural units.

For the spherical trap situation we are left with the variational parameters α and β only, but when we study an elliptical trap we still want to get rid of ω_z. Since β^2 should be the factor in front of the z-coordinate when the Hamiltonian is dimensionless, it can be proven that $\beta = \omega_z/\omega_{HO}$, see appendix C. We end up with the Hamiltonian

$$H = \sum_i \left(\frac{1}{2}\Big(-\nabla^2 + x_i^2 + y_i^2 + \beta^2 z_i^2 \Big) \right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{21}$$

where β is chosen to be $\sqrt{8} \cong 2.82843$, which was used in the experiment of Anderson et.al. [7].

2.5 Error estimation

When presenting data from an experiment, one should always know the errors in the answer. Experimental data, including data from numerical experiments, are never determined beyond any doubt, and an estimate of this error should therefore be presented alongside the data.

There are two kinds of errors. Statistical errors originate from how much statistics one has; when 10^6 measured points give approximately the same answer, one can be more sure that the actual value is close to those points, more so than if one only has 1 point of statistical data. Estimating the statistical error is easily done. The systematic error, however, is harder to handle. It arises for example from calculations being based on faulty theory, or defect measurement devices. Here, we will present how to get an estimate of the statistical error in a numerical experiment.

When conducting an experiment α with n measured points, $\{x_1, ..., x_n\}$, the sample mean $\langle x_\alpha \rangle$ of the experiment is defined as shown in equation 22.

$$\langle x_\alpha \rangle = \frac{1}{n} \sum_{k=1}^{n} x_{\alpha,k} \tag{22}$$

The corresonding sample variance σ_α is then defined as

$$\sigma_\alpha^2 = \frac{1}{n} \sum_{k=1}^{n} (x_{\alpha,k} - \langle x_\alpha \rangle)^2 \tag{23}$$

This gives us the error in the given experiment α. If we repeat this experiment m times, the mean after all the experiments are

$$\langle x_m \rangle = \frac{1}{n} \sum_{k=1}^{n} \langle x_\alpha \rangle. \tag{24}$$

The total variance is then

$$\sigma_m^2 = \frac{1}{m} \sum_{\alpha=1}^{m} (\langle x_\alpha \rangle - \langle x_m \rangle)^2. \tag{25}$$

This can be reduced to

$$\sigma_m^2 = \frac{\sigma^2}{n} + \text{covariance term}, \tag{26}$$

where σ is the sample variance over all the experiments, defined as

$$\sigma^2 = \frac{1}{mn} \sum_{\alpha=1}^{m} \sum_{k=1}^{n} (x_{\alpha,k} - \langle x_m \rangle)^2 \tag{27}$$

and the covariance is the linear correlation between the measured points. The definition of the covariance is shown in equation 28.

$$cov(x, y) = \frac{1}{n^2} \sum_i \sum_{j>i} (x_i - x_j)(y_i - y_j). \tag{28}$$

A common simplification is to reduce equation 26 to the following:

$$\sigma^2 \approx \langle x^2 \rangle - \langle x \rangle^2 \tag{29}$$

This equation, however, does not take into account the covariance term from equation 26, and as the covariance term is added to the expression for the variance, 29 will underestimate the uncertainty σ for positive covariances.

A direct implementation of equation 26 including the covariance term is not suitable, as the expression for the covariance includes a double sum, and for a large number of iterations, this will turn into an extremely time consuming process for a large number of Monte Carlo iterations. Luckily, there are methods for calculating an accurate estimation of the variance without including a double loop in the Monte Carlo program. One of these methods is the blocking method, which is presented in section 3.4.

3 Method

3.1 Variational Monte Carlo

Variational Monte Carlo (VMC) is a widely used method for approximating the ground state of a quantum system. The method is based on Markov chains, where one particle or a set of particles are moved one step for each cycle, i.e.

$$\vec{R}_{new} = \vec{R} + r \cdot \text{step}. \tag{30}$$

Both the direction and particles moves are randomly chosen, so with a plain VMC implementation the particles will move randomly and independently of each other. We are going to use the Metropolis algorithm in addition to the VMC, where the Metropolis algorithm accepts or rejects moves based on the probability ratio between the old and the new position. This makes the system approach the most likely state, and the idea is that after a certain number of cycles the system will be in the most likely state.

3.2 Metropolis Algorithm

As mentioned above the task of the Metropolis algorithm is to move the system against the most likely state. The standard algorithm, here named brute force, is the simplest one, and does not deal with the transition probabilities.

The modified Metropolis-Hastings algorithm includes, on the other hand, the transition probabilities and will be slightly more time consuming per cycle. We expect the latter to converge faster to the most likely state.

The foundation of the Metropolis algorithm is that the probability for a system to undergo a transition from state i to state j is given by the transition probability multiplied by the acceptance probability

$$W_{i\to j} = T_{i\to j} \cdot A_{i\to j} \tag{31}$$

where $T_{i\to j}$ is the transition probability and $A_{i\to j}$ is the acceptance probability. Built on this, the probability for being in a state i at time (step) n is

$$P_i^{(n)} = \sum_j \left[P_j^{(n-1)} T_{j\to i} A_{j\to i} + P_i^{(n-1)} T_{i\to j}(1 - A_{i\to j}) \right] \tag{32}$$

since this can happen in two different ways. One can start in this state i at time $n-1$ and be rejected or one can start in another state j at time $n-1$ and complete an accepted move to state i. In fact $\sum_j T_{i\to j} = 1$, so we can rewrite this as

$$P_i^{(n)} = P_i^{(n-1)} + \sum_j \left[P_j^{(n-1)} T_{j\to i} A_{j\to i} - P_i^{(n-1)} T_{i\to j} A_{i\to j} \right]. \tag{33}$$

When the times goes to infinity, the system approaches the most likely state and we will have $P_i^{(n)} = p_i$, which requires

$$\sum_j \left[p_j T_{j\to i} A_{j\to i} - p_i T_{i\to j} A_{i\to j} \right] = 0. \tag{34}$$

Rearranging, we obtain the useful result

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i T_{i\to j}}{p_j T_{j\to i}} \tag{35}$$

which will be used in the acceptance criteria.

3.2.1 Brute force

In the brute force Metropolis algorithm we want to check if the new position is more likely than the current position, and for that we calculate the probabilities $P(\vec{R}) = |\Psi_T(\vec{R})|^2$ for both positions. We get rid off the transition probabilities setting $T_{i\to j} = T_{j\to i}$, and then end up with the plain ratio

$$w = \frac{P(\vec{R}_{new})}{P(\vec{R})} = \frac{|\Psi_T(\vec{R}_{new})|^2}{|\Psi_T(\vec{R})|^2}. \tag{36}$$

w will be larger than one if the new position is more likely than the current, and smaller than one if the current position is more likely than the new one. Metropolis handle this by accepting the move if the ratio w is larger than a random number r in the interval $[0, 1]$, and rejecting if not:

$$\text{New position: } \begin{cases} \text{accept} & \text{if} \quad w > r \\ \text{reject} & \text{if} \quad w \leq r. \end{cases} \tag{37}$$

3.2.2 Importance sampling

The importance sampling technique is often refered to as Metropolis-Hastings algorithm. The approach is the same as for the brute force Metropolis algorithm, but we will end ut with a slightly more complicated acceptation criteria. To understand the details, we need to begin with the Fokker-Planck equation, which describes the time-evolution of the probability density function $P(R, t)$. In one dimension it reads

$$\frac{\partial P(R,t)}{\partial t} = D\frac{\partial}{\partial R}\left(\frac{\partial}{\partial R} - F\right)P(R,t). \tag{38}$$

where F is the drift force given by equation 39 and D is the diffusion coefficient, in this case equal 0.5. Calculations of the analytical expression of the drift force F for a spherical harmonic oscillator and $a = 0$ can be found in appendix B.

$$F(R) = \frac{2\nabla\psi_T}{\psi_T} \tag{39}$$

Even though the probability density function can give a lot of useful information, an equation describing the motion of a such particle would be more appropriate for our purposes. Fortunately this equation exists, and satisfies the Fokker-Planck equation. The Langevin equation can be written as

$$\frac{\partial R(t)}{\partial t} = DF(R(t)) + \eta \tag{40}$$

where η can be considered as a random variable. This differential equation can be solved by applying the forward Euler method and introducing gaussian variables ξ

$$R_{new} = R + DF(R)\Delta t + \xi\sqrt{\Delta t} \tag{41}$$

which will be used to update the position. This is an improved way of choosing the direction in which the particle is moved compared to the brute force algorithm, as the drift force $F(R)$ says something about which direction

the particle is pushed in, and the choice of the new proposed position is thus dependent on this.

Moreover we also need to update the acceptance criteria since we no longer ignore the transition probabilities. With the Fokker-Planck equation as base, the transition probabilities are given by Green's function

$$\begin{aligned} T_{R\to R_{new}} &= G(R_{new}, R, \Delta t) \\ &= \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp[-(R_{new} - R - D\Delta t F(R))^2/4D\Delta t] \end{aligned} \tag{42}$$

and the acceptance criteria becomes

$$r < \frac{G(R, R_{new}, \Delta t)|\Psi_T(R_{new})|^2}{G(R_{new}, R, \Delta t)|\Psi_T(R)|^2}. \tag{43}$$

3.3 Minimization methods

When the interaction term is excluded, we know which α that corresponds to the energy minimum, and it is in principle no need to try different α's. However, sometimes we have no idea where to search for the minimum point, and we need to try various α values to determine the lowest energy. If we do not know where to start searching, this can be a time consuming activity. Would it not be nice if the program could do this for us?

In fact there are multiple techniques for doing this, where the most complicated ones obviously also are the best. Anyway, in this project we will have good initial guesses, and are therefore not in need for the most fancy algorithms.

3.3.1 Gradient descent

Perhaps the simplest and most intuitive method for finding the minimum is the gradient descent method, which reads

$$\alpha^+ = \alpha - \eta \cdot \frac{d\langle E_L(\alpha)\rangle}{d\alpha}. \tag{44}$$

where α^+ is the updated α and η is the step size. The idea is that one finds the gradient of the energy with respect to a certain α, and moves in the direction which minimizes the energy. This is repeated until one has found an energy minimum, where the energy minimum is defined as either where $\frac{d\langle E_L(\alpha)\rangle}{d\alpha}$ is smaller than a given tolerance, or the α energy fluctuates around a value, and thus changes minimally.

To implement equation 44, we need an expression for the derivative of E_L with respect to alpha:

$$\bar{E_\alpha} = \frac{d\langle E_L(\alpha)\rangle}{d\alpha}. \tag{45}$$

By using the expression for the expectation value for the local energy $\langle E_L(\alpha)\rangle$ in equation 46

$$\langle E_L(\alpha)\rangle = \frac{\langle\psi_T(\alpha)|H|\psi_T(\alpha)\rangle}{\langle\psi_T(\alpha)|\psi_T(\alpha)\rangle} \tag{46}$$

and applying the chain rule of differentiation, it can be shown that equation 45 is equal to equation 47

$$\bar{E_\alpha} = 2[\langle E_L(\alpha)\frac{\bar{\psi_\alpha}}{\psi_\alpha}\rangle - \langle E_L(\alpha)\rangle\langle\frac{\bar{\psi_\alpha}}{\psi_\alpha}\rangle] \tag{47}$$

where

$$\bar{\psi_\alpha} = \frac{d\psi(\alpha)}{d\alpha}. \tag{48}$$

The algorithm of this minimization method is thus as follows:

```
for(max number of iterations with minimizing)

        do M Monte Carlo cycles
        calculate E_L and dE_L/dalpha

        Check if dE_L/dalpha < eps or alpha fluctuation over the last 5
            ↪ steps is < eps


                if yes, print optimal alpha and break loop
                if no, continue to next iteration
```

3.4 Blocking method

As described in section 2.5, we need a method to give a proper estimation of the variance σ^2 of the points in our experiment, preferably without calculating the double loop from the expression of the covariance in equation 28.

One method that can be used, is the blocking method, which is quite fast and can handle large data sets. Say that we have a data set $\{x_1, ..., x_i, ..., x_n\}$ from an experiment, which in our case will be the estimations of the local energies for each Monte Carlo cycle. The mean of this data set is m, and we

want to estimate the variance of this data set, $\sigma^2(m)$, including the covariance.

The variance is defined as

$$\sigma^2 = \frac{1}{n-1}\sum_i (x_i - m) \tag{49}$$

However, this does not include the covariance, as the points x_i are correlated.

If we transform the data set $\{x_1, ..., x_i, ..., x_n\}$ by taking the mean of two neighbouring points in the following way

$$x_i' = \frac{1}{2}(x_{2i} + x_{2i+1}) \tag{50}$$

the number of points in the transformed data set $n' = \frac{1}{2}n$. Each pair of points x_i' is referred to as a block, and using the variance formula above, the variance within each block is calculated. An estimate of the total variance $\hat{\sigma}$ is then

$$\hat{\sigma} = \frac{\sigma_1 + ... + \sigma_{n'}}{n'} \tag{51}$$

By repeating this procedure several times, and each time using more measurements for each block, the estimated standard deviation $\hat{\sigma}$ will grow, and eventually reach a plateau. When the estimated standard deviation $\hat{\sigma}$ flats out, it means that the blocks are no longer correlated, and thus the covariance which bugged us in equation 26, is now 0. The covariance is therefore accounted for, and the $\hat{\sigma}$ is now a good estimation of the error in the data set.

4 Code

This project is much about developing the code to make VMC run correctly, and in this section we will explain briefly how the code works and which implementation techniques we use. The implementation is dimensionless, such that the variational parameters are the only quantities.

4.1 Code structure

To keep the program neat, specific parts were placed in specific files. For instance, a file gd.cpp handle everything about the gradient descent method, and another file wavefunction.cpp (a little misleading) contains a class which both sets up the wave function and calculate the local energy given a wave

function. We also collected the tools that are used by multiple scripts in its own file to maximize the reuse of the code, in particular Green's function and the quantum force calculations. In figure 2 we give an overview of the implementation, where the lines indicate which scripts that communicate.

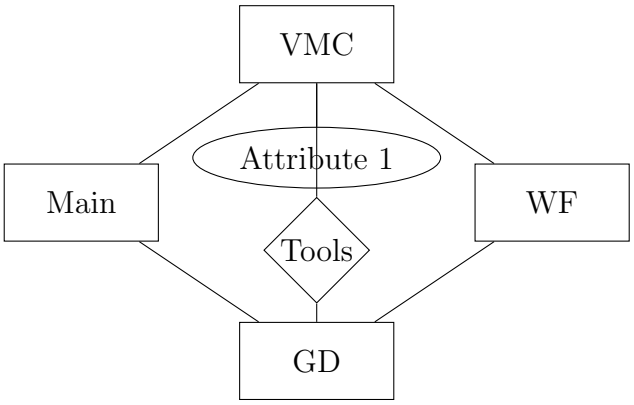

Figure 2: An overview of the code structure. metropolis.cpp is refered to as VMC, gd.cpp is GD, wavefunction.cpp is WF, tools.cpp is Tools and obviously main.cpp is Main. See text for furter description.

The acronyms are not really accurate, VMC is of course much about the Metropolis algorithm, and GD contains both the VMC and the Metropolis algorithm. Anyway, it is hard to find acronyms that tells everything about the script. A standard VMC run goes as following: metropolis.cpp (VMC) is called from main.cpp (Main), which agains calls wavefunction.cpp (WF) to set up the wave function, calls tools.cpp (Tools) to get the value of Green's function and the quantum force and then again calls WF to obtain the local energy. At the end the local energy is printed out.

4.2 Implementation

We use the vector package to create a 2D-array where the positions are stored. For the exact implementation, see the github repository linked on page one. We always struggle for better performance, and to achieve that we try to avoid heavy loops and repetition of calculations. A second priority is to make the code as general as possible, such that the code can be reused for other purposes. As an example we always loop over an arbitrary given number of dimensions, and similar operations are done throughout the programs.

4.2.1 VMC and Metropolis

Present pseudo code of how VMC and Metropolis are implemented

4.2.2 GD

4.2.3 Tests

5 Results

We will first present the results obtained without repulsive interaction, and thoughout this part we will consequently use $\alpha = 0.5$ and run for $1e6$ Monte Carlo cycles and for three spatial dimensions.

Thereafter we will turn to the more realistic and interesting situation where the particles interact with each other. We will still study the behavior in three dimensions and for $1e6$ cycles, but we need to set the scattering distance to $a = 0.0043$ and find the α which gives the minimum local energy. In this part we will also turn on the elliptical harmonic oscillator potential.

5.1 E_L as a function of the variational parameter α

In search of the energy minimum we tried various α values, and optimal α was 0.5, as one can see in figure 3. The variance associated with the energy measurement in figure 3, and we see that the energy minimum and the variance minimum share the same α.

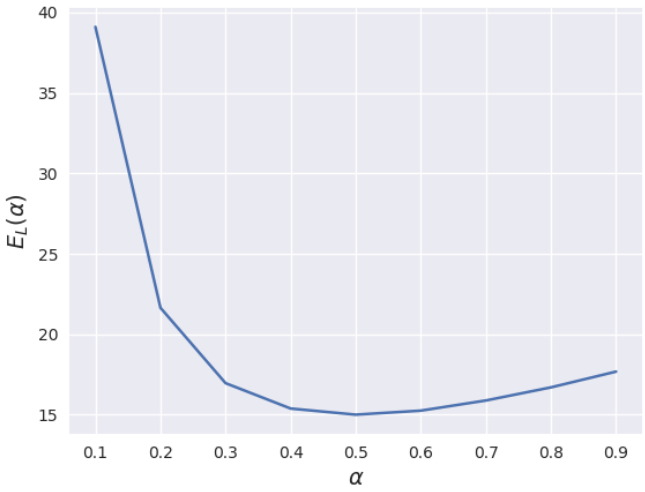

Figure 3: The local energy E_L calculated with the brute force Metropolis algorithm, as a function of the variational parameter α in the interval $[0.1, 0.9]$. for each α we used $1e6$ Monte Carlo cycles in three spatial dimensions.

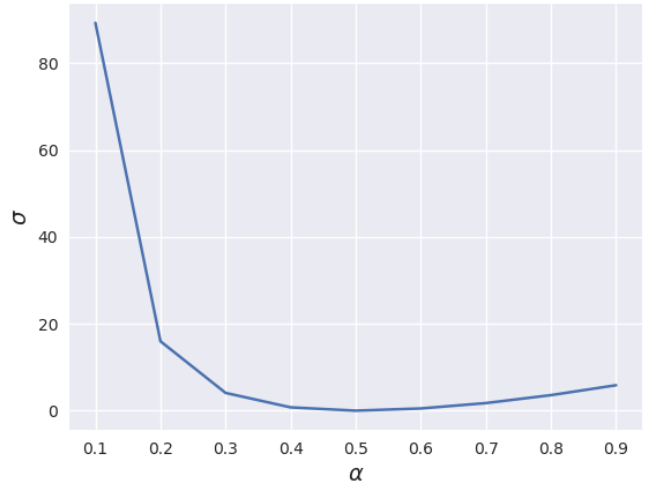

Figure 4: The variance of the local energy E_L calculated with the brute force Metropolis algorithm, as a function of the variational parameter α associated with the energy plot in figure 3. The variance is calculated with equation 29

.

5.2 E_L calculation and CPU-time

For the brute force Metropolis algorithm we developed both an analytical and a numerical method to calculate the local energy. In table 1 we present the results from these calculations and the performance. The results from the calculations with the Metropolis-Hastings algorithm are presented in table 2.

Table 1: The local energy calculated for $a = 0$ without hard-sphere interaction with brute force Metropolis algorithm, for both the analytical and numerical local energy calculation. The variational parameter α is 0.5. The results are compared to the exact answer, obtained from equation 11. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the calculated energies.

	Analytical		**Numerical**		Exact
N	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$	CPU-time [s]	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$	CPU-time [s]	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$
1	1.5000(15)	0.15420	1.5000(15)	0.65714	1.5000
10	15.000(15)	0.54785	15.000(15)	9.9844	15.000
100	150.00(15)	13.573	150.00(15)	2743.4	150.00
500	750.00(75)	282.79	750.00(75)	2.8744e5	750.00

Table 2: The local energy calculated for $a = 0$ without hard-sphere interaction with the Metropolis-Hastings algorithm with analytical local energy calculation. The variational parameter α is 0.5. The results are compared to the exact answer, obtained from equation 11. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the local energy calculations.

	Analytical		**Exact**
N	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$	CPU-time [s]	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$
1	1.5000(15)	0.26693	1.5000
10	15.000(15)	0.54930	15.000
100	150.00(15)	16.117	150.00
500	750.00(75)	292.10	750.00

5.3 Acceptance ratios

We study the acceptance ratio for the brute force and importance sampling algorithms, with no interaction. These calculations are done with ten particles in three dimensions.

5.3.1 Brute force dependence on stepsize

The acceptance ratio for the brute force algorithm as a function of the stepsize is shown in figure 5.

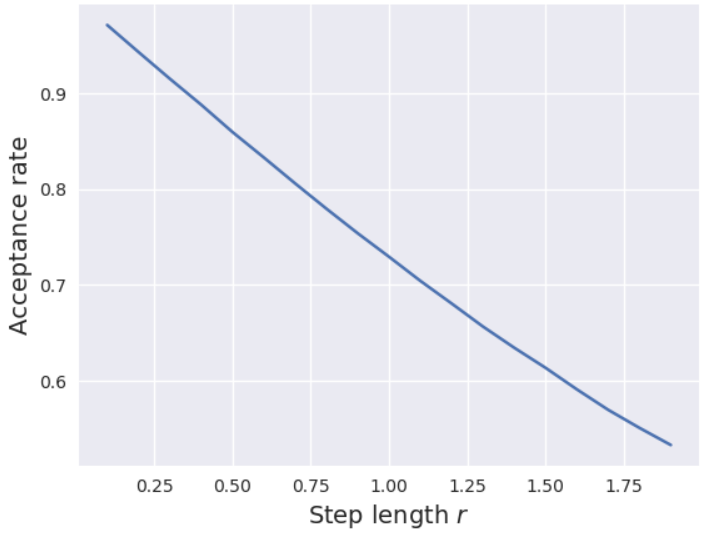

Figure 5: Acceptance ratio for different choices of the stepsize used in the brute force algorithm

5.3.2 Importance sampling dependence on timestep

The acceptance ratio for the importance sampling algorithm as a function of the timestep δt is shown in figure 6.

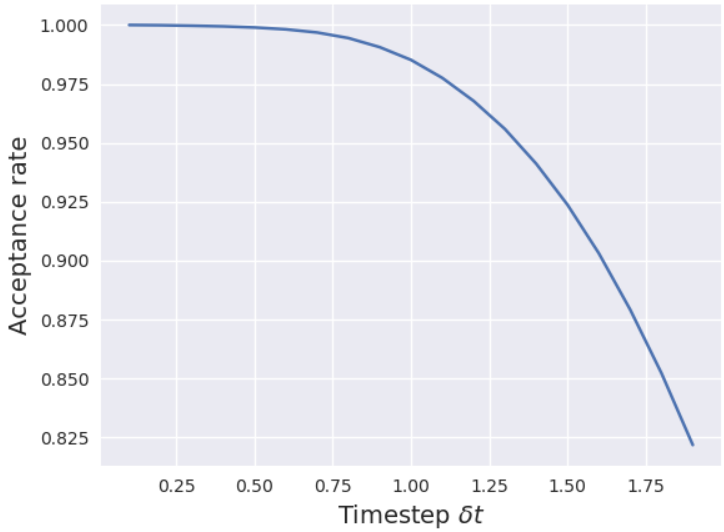

Figure 6: Acceptance ratio for different choices of the timestep δt with the Metropolis-Hastings algorithm.

5.4 Variance calculation

In table 3, we present the results of the local energy calculation with proper error evaluation of statistical error. The statistical error as found with the Blocking method is also compared to the simple way of approximating the variance with equation 29.

Table 3: The local energy calculated for $a = 0$ without hard-sphere interaction with the Metropolis-Hastings algorithm with analytical local energy calculation, this time with proper evaluation of the statistical error. The calculations are run in three dimensions with $1e6$ Monte Carlo cycles, and the variational paramameter $\alpha = 0.5$

N	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$	σ^2 Blocking	$\sigma^2 \approx \langle E_L^2 \rangle - \langle E_L \rangle^2$
1	1.5000	5.0102e-2	5.0100e-2
10	15.000	4.9701e-1	4.9678e-1
100	150.00	5.2417	5.2184
500	750.00	9.0710	8.4831

5.5 VMC with repulsive interaction

In table 4, the results from the calculation of the elliptical trap are presented. The variational parameter α was varied manually to find the minimum of the local energy E_L.

Table 4: The local energy E_L calculated for different α, with $a = 0.0043$ with hard-sphere interaction, elliptical trap and $\beta = 2.82843$, with the brute force algorithm with analytical local energy calculation. The calculations are run in three dimensions with $1e6$ Monte Carlo cycles

	N=1		**N=10**		**N=100**	
α	Comp.	GP	Comp.	GP	Comp.	GP
0.2	3.48489	3.44943	35.0915	34.9	413.853	
0.3	2.73719	2.69688	27.5577	27.4	359.313	
0.4	2.47403	2.43994	24.8541	24.6	354.602	
0.5	2.41422	2.38130	24.2644	24.2	371.025	
0.6	2.45456	2.42185	24.6661	24.6	395.560	
0.7	2.55036	2.51910	25.6275	25.6	425.227	
0.8	2.69082	2.65182	26.8884	26.8	457.480	

5.5.1 Gradient Decent

The gradient decent method was then used to find the minimum in the local energy in the interacting case. The results are presented in table 5.

Table 5: The local energy E_L calculated with the gradient decent method, with $a = 0.0043$ with hard-sphere interaction, elliptical trap and $\beta = 2.82843$, with the brute force algorithm with analytical local energy calculation. The table shows the optimal α found by the method, and the resulting local energy. The calculations are run in for three dimensions with $1e6$ Monte Carlo cycles.

N	α	E_L	$\frac{\partial E_L}{\partial \alpha}$
1	0.349923	1.68819	0.000853655
10	0.345019	16.9141	-0.365277

5.5.2 Wave function

Now as we have found the optimal α's for 1, 10 and 100 particles in an elliptical trap, we can plot the respective wave functions. In figure **??** the

wave functions are plotted as a function of the spread in x-direction (dashed lines) in addition to the ideal wave function in a spherical trap (solid line). The results are comparable to earlier publications (see for instance Ref.[4]), and show that the wave functions get a broader shape when particles are added.

5.5.3 Onebody density

In section 2.3 we presented a couple of ways computing the onebody density. Because both should give the same result, we selected the simplest one, which is the method with the bins. We conduct the investigations in elliptic traps, with 10 particles, 3 dimensions and 1e6 Monte Carlo cycles. With the optimal parameter α for the ground state wave function and for both the interacting and the non-interacting case, the onebody density plot is found in figure 7.

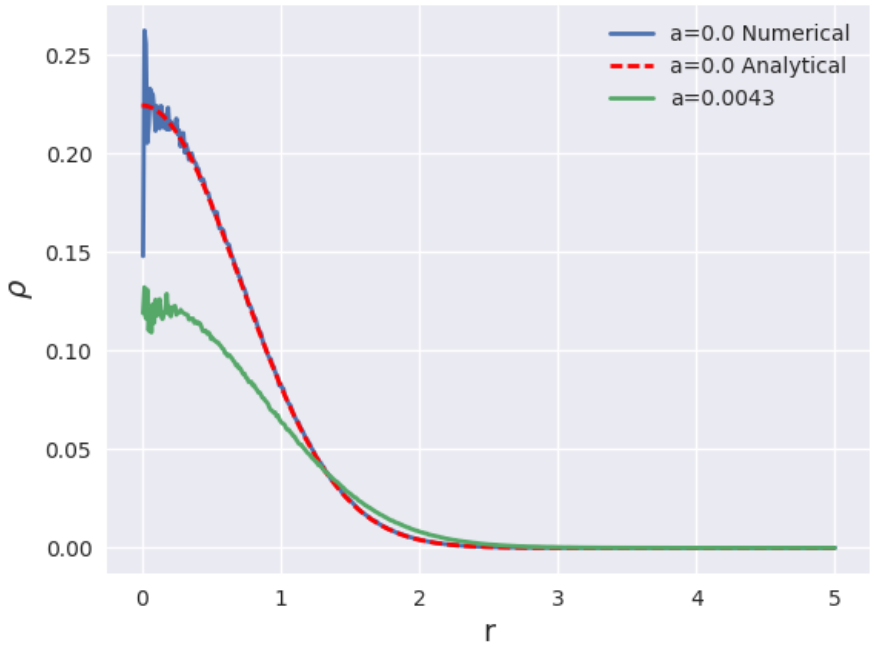

Figure 7: The onebody density of the bosonic system, which is run with $\alpha = 0.5$ for $a = 0$ and compared with the analytical expression. The optimal α for the interacting case with $a = 0.0043$ was found to be $\alpha = 0.3450$. The radius is given in units of a_{ho} along the x-axis, and the density is in units of a_{ho}^{-3}.

From the onebody density plot, we observe that the experimental onebody density matches the exact without interacting, and the half peak value is ~ 0.8. For the case with interaction, the peak is obviously lower, and we observe a half peak value of ~ 1.0, which makes it broader.

6 Discussion

For the case without interaction and spherical harmonic oscillator we see from figure 3 that the minimum of the local energy is found when the varitional parameter $\alpha = 0.5$. We also observe that $\alpha = 0.5$ gives the minimal variance, and seen from figure 4. For small α's the energies and variances explode.

When comparing the results from the brute force Monte Carlo calculation with spherical harmonic oscillator and no interaction, as shown in table 1, we observe that in the case when we use an analytical expression for E_L, the simulation results yields the exact energy, as obtained from equation 11. Also the numerically calculated E_L, as described in section 2.2.1, reproduces the exact answers with the same presicion as the analytical.

Moreover, when comparing the CPU time difference between these two calculations, the numerically calculated E_L spends much more time on the calculations compared to the analytically calculated E_L-case. The reason for this is probably the numerical differentiation mentioned in equation 15, which in addition will cause some minor errors. The time difference is as large as a factor of 10^3 for the 500 particles, three dimensions case. which makes it preferable to use the analytically calculated E_L in our calculations. However, not all trial wave functions Ψ_T or Hamiltonians H will yield an E_L which is analytically possible to calculate, and in this case, one should have a functioning algorithm for numerical calculation of E_L as well.

When comparing the results from the brute force and Metropolis-Hastings algorithms, shown in tables 1 and 2, the reader can observe that the Hastings algorithm reproduces the exact energies, as the brute force algorithm does, but that the Hastings algorithm is slightly slower. This slight time increase is due to the fact that while the brute force algorithm picks a proposed new step at random, the Hasting algorithm makes a more educated move by calculating the drift force F on the particle, and decide what direction the particle is most likely to move in, as described in section 3.2.2. While this calculation spends slightly more CPU time, it is rewarded by the Hastings algorithm having a higher acceptance rate of the proposed moves compared to the brute force Metropolis. This can be seen from figures 5 and 6. The acceptance ratio of the Hastings algorithm for a timestep $\Delta t = 1$ is much

higher than the acceptance ratio for brute force with a stepsize $r = 1$. This results in the Hastings algorithm moving faster towards the minima compared to the brute force method. This effect, however, is not that visible in our project, but can make a vital difference for larger Monte Carlo simulations.

Still studying figure 6 and 5, we observe that the importance sampling algorithm has a high acceptance rate for intermediate timesteps Δt, but the acceptance rate drops seemingly exponentially for higher timesteps. Comparing this to the brute force acceptance rate, the drop in the acceptance rate is more like a lineary decreasing curve. This is because the acceptance probability for brute force to do a step in the "wrong direction" is still present, while moving in the wrong direction with importance sampling is much less probable, and therefore the acceptance rate decreases rapidly with increased timestep.

In table 3, we repeated the calculation of the no interaction spherical potential with importance sampling, but this time we included a proper evaluation of the error. We compared the variances obtained from the simple, but incorrect way of calculating error, as shown in equation 29 to a proper variance calculation using the blocking method, by evaluation equation 26. As expected, the simplified variance calculation underestimates the statistical error, as it does not include the covariance term in equation 26. However, the deviation between the two is small for few number of particles. We would say that as blocking is a quick way of calculating the statistical error, it should be done when presenting results, but the simplified error calculation could still give a quick estimation of the magnitude of the variance in a simulation.

Further we have seen that the interaction causes some differences related to the.. WE REALLY NEED TO WRITE ABOUT THE INTERACTION CASE, INCLUDING GRADIENT DESCENT

For the onebody density plots we saw that the densities had gaussian shapes, which come from the wave functions (matematically the wave functions are gaussian functions). The peak decreased when adding interacting because the bosons then are more spread out, which also is the reason why it gets broader. We also observed noise for small radii, which is caused by the dividing on sphere surfaces, which are small numbers for small radii. All particles are located within a distance $3a_{HO}$ from particle i, which makes sence since we only study 10 particles.

Must benchmark results from e and f!

7 Conclusion

In this project, we have found that the Variational Monte Carlo method reproduces analytical calculations for the spherical harmonic oscillator without interparticle interaction. The minimum in the energy was found for this case to be $\alpha = 0.5$, which is as expected. Both the brute force method with an analytically calculated for E_L and the Hastings algorithm were quick and accurate, while the brute force method with a numerically calculated E_L was much slower.

The Hastings algorithm had a higher acceptance rate for the proposed moves in the Monte Carlo iterations compared to the brute force method. This should result in the Hastings method converging faster, although we did not observe that in this report. We also saw that the Hastings algorithm was more dependent on the timestep compared to the brute force method's dependence om stepsize. This is as we would expect from theory.

When including a proper evaluation of the error by using the blocking method, we saw that the blocking method gave a slightly higher estimated variance, compared to the estimation from equation 29. This is as expected.

ALSO HERE WE NEED TO MENTION THE INTERACTING CASE

Despite the noise seen for onebody density, the non-interacting calculation matched the exact expression pretty well. The particles also spread out when adding repulsive interaction, which is highly expected.

8 References

[1] Morten Hjorth-Jensen. Computational Physics 2: Variational Monte Carlo methods, Lecture Notes Spring 2018. Department of Physics, University of Oslo. (2018).

[2] J. L. DuBois and H. R. Glyde, H. R., *Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis*, Phys. Rev. A **63**, 023602 (2001).

[3] J. K. Nilsen, J. Mur-Petit, M. Guilleumas, M. Hjorth-Jensen, and A. Polls, *Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region*, Phys. Rev. A **71**, 053610 (2005).

[4] F. Dalfovo, S. Giorgini, L.P.Pitaevskii, S.Stringari, *Theory of Bose-Einstein condensation in trapped gases* Rev. Mod. Phys. **71**, 463 (1999).

[5] Jesse Emspak, *States of Matter: Bose-Einstein Condensate*, LiveScience (2016) `https://www.livescience.com/54667-bose-einstein-condensate.html` Downloaded March 15th 2018.

[6] Sidney Perkowitz *Bose-Einstein condensate* Encyclopaedia Britannica `https://www.britannica.com/science/Bose-Einstein-condensate` Downloaded March 15th 2018.

[7] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, E. A. Cornell, *Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor*, Science **269**, (1995).

[8] J. K. Nilsen *Bose-Einstein condensation in trapped bosons: A quantum Monte Carlo analysis*, Master thesis 2004, Department of Physics, University of Oslo, (2004).

Appendix A

In the theory section we presented the analytical expressions for the local energy, and in this appendix we go though the derivations carefully for those of you who are curious. Firstly we present the local energies where we ignore the two-particle interaction, and we also focus on the spherical harmonic oscillator trap since we do not look at the elliptical trap without interaction, even though the expressions are similar.

Secondly the general expression which can be used with and without interaction, with spherical or elliptical harmonic oscillator trap and with a various number of parameter values is derived. This can in principle be used for every case, but it is more complex than the special ones where we ignore interaction and we therefore expect it to be slower.

8.1 Without repulsive interaction

We calculated the analytical expression for the local energy E_L, as given by 9, for the non-interaction ($a = 0$) case for the spherical harmonic oscillator. In this case, the trial wave function only consists of the one body part, and is thus for N particles given by:

$$\Psi_T(\vec{r}) = \prod_i^N e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)} \tag{52}$$

We now want to calculate the analytical expressions for E_L for one particle and one dimension, and N particles and three dimensions.

8.1.1 One particle, one dimension

From equation 52, the trial wave function for one particle and one dimension is as follows:

$$\Psi_T(x) = e^{-\alpha x^2} \tag{53}$$

We start from the local energy equation in equation 9, where we need to take the second derivative

$$\frac{d\Psi_T}{dx} = -2\alpha x e^{-\alpha x^2} \tag{54}$$

$$\frac{d^2\Psi_T}{dx^2} = -2\alpha e^{-\alpha x^2} + 4\alpha^2 x^2 e^{-\alpha x^2}. \tag{55}$$

To get a neat expression, we use the dimensionless Hamiltonian with spherical harmonic oscillator potential, from equation 1, and obtain

$$\begin{aligned} E_L(\alpha) &= -\frac{1}{2}(-2\alpha + 4\alpha^2 x^2) + \frac{1}{2}x^2 \\ &= \alpha + \Big(\frac{1}{2} - 2\alpha^2\Big)x^2 \end{aligned} \tag{56}$$

8.1.2 N particles, three dimensions

When extending the non-interaction case to N particles and three dimensions, the trial wave function will take the form listed in equation 52, and the Hamiltonian will be as listed in equation 1 with the dimensionless spherical harmonic oscillator potential $\frac{1}{2}r_i^2$, as listed in 2.

For this case it might be better to use spherical coordinates, where the Laplace operator is given by

$$\nabla^2 = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) \tag{57}$$

when we only take the radial part into account. We transform the wavefunction from a product to a function with a sum in the exponent, and calculate the first derivative

$$\frac{\partial \Psi_T}{\partial r_j} = -2\alpha r_j \exp\Big[-\alpha\Big(\sum_i r_i^2\Big)\Big]. \tag{58}$$

After adding a r_j^2, we differentiate the expression again

$$\frac{\partial}{\partial r_j}\Big(-2\alpha r_j^3 \exp\Big[-\alpha\Big(\sum_i r_i^2\Big)\Big]\Big) = (-6\alpha r_j^2 + 4\alpha^2 r_j^4)\exp\Big[-\alpha\Big(\sum_i r_i^2\Big)\Big]$$

and we obtain

$$\frac{\nabla_j^2 \Psi_T}{\Psi_T} = -6\alpha + 4\alpha^2 r_j^2. \tag{59}$$

Again we get the local energy from the Hamiltonian with a spherical harmonic oscillator potential.

$$\begin{aligned} E_L(\alpha) &= \sum_j -\frac{1}{2}(-6\alpha + 4\alpha^2 r_j^2) + \frac{1}{2}r_j^2 \\ &= 3N\alpha + \Big(\frac{1}{2} - 2\alpha^2\Big)\sum_{j=1}^{N} r_j^2 \end{aligned} \tag{60}$$

8.1.3 General

If one now studies the local energy expressions for one particle in one dimension and N particles in three dimensions, one can see a pattern and it is easy to imagine that there exists a general expression for the local energy, when interaction is ignored and for the spherical harmonic oscillator case. At least for 1, 2, and 3 dimensions, it can be shown that

$$E_L(\alpha) = dim \cdot N \cdot \alpha + \Big(\frac{1}{2} - 2\alpha^2\Big) \sum_j r_j^2 \tag{61}$$

8.2 With repulsive interaction

As mentioned in the theory section, the calculations get more complicated when adding the interaction, and we are in fact not able to find the local energy analytically. However, we can simplify the local energy expression and hopefully gain some speed up compared to when we do all the calculations numerically.

We start with defining the onebody part of the wavefunction as Φ and the Jastrow factor as $\exp\big(u(r_{ij})\big)$ such that the total trial wavefunction becomes

$$\Psi_T(\vec{r}_1, \dots \vec{r}_N) = \Big[\prod_i \Phi(\vec{r}_i)\Big] \exp\Big(\sum_{i<j} u(r_{ij})\Big) \tag{62}$$

Thereafter we calculate the term connected to the kinetic part of the Hamiltonian,

$$\frac{\nabla_k^2 \Psi_T(\vec{r})}{\Psi_T(\vec{r})}, \tag{63}$$

which is the most difficult part. We apply the product rule and obtain the following expression for the first derivative

$$\begin{aligned} \nabla_k \Psi_T(\vec{r}) = \nabla_k \Phi(\vec{r}_k) \Big[\prod_{i \neq k} \Phi(\vec{r}_i)\Big] \exp\Big(\sum_{i<j} u(r_{ij})\Big) \\ + \prod_i \Phi(\vec{r}_i) \exp\Big(\sum_{i<j} r(r_{ij})\Big) \sum_{j \neq k} \nabla_k u(r_{ij}) \end{aligned} \tag{64}$$

For the second derivative we get five terms in total where two of them

are cross terms and therefore equal.

$$\begin{aligned}\nabla_k^2\Psi_T(\vec{r}) = \nabla_k^2\Phi(\vec{r}_k)\Big[\prod_{i\neq k}\Phi(\vec{r}_i)\Big]\exp\Big(\sum_{i<j}u(r_{ij})\Big) \\ + \nabla_k\Phi(\vec{r}_k)\Big[\prod_{i\neq k}\Phi(\vec{r}_i)\Big]\exp\Big(\sum_{i<j}u(r_{ij})\Big)\sum_{j\neq k}\nabla_k u(r_{ij}) \\ + \nabla_k\Phi(\vec{r}_k)\Big[\prod_{i\neq k}\Phi(\vec{r}_i)\Big]\exp\Big(\sum_{i<j}u(r_{ij})\Big)\sum_{j\neq k}\nabla_k u(r_{ij}) \\ + \prod_i\Phi(\vec{r}_i)\exp\Big(\sum_{i<j}u(r_{ij})\Big)\sum_{i\neq k}\nabla_k u(r_{ij})\sum_{j\neq k}\nabla_k u(r_{ij}) \\ + \prod_i\Phi(\vec{r}_i)\exp\Big(\sum_{i<j}u(r_{ij})\Big)\sum_{j\neq k}\nabla_k^2 u(r_{ij})\end{aligned} \tag{65}$$

To simplify this, we do a change of variables

$$\frac{\partial}{\partial\vec{r}_k} = \frac{\partial}{\partial\vec{r}_k}\frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial\vec{r}_k}\frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k-\vec{r}_j)}{r_{kj}}\frac{\partial}{\partial r_{kj}} \tag{66}$$

where we have used that

$$\frac{\partial r_{kj}}{\partial\vec{r}_k} = \frac{\vec{r}_k-\vec{r}_j}{|\vec{r}_k-\vec{r}_j|} = \frac{\vec{r}_k-\vec{r}_j}{r_{kj}}. \tag{67}$$

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

$$\nabla_k u(r_{ij}) = \frac{\partial r_{kj}}{\partial\vec{r}_k}\frac{\partial}{\partial r_{kj}}\big(u(r_{ij})\big) = \frac{\vec{r}_k-\vec{r}_j}{|\vec{r}_k-\vec{r}_j|}u'(r_{kj}) \tag{68}$$

$$\begin{aligned}\nabla_k^2 u(r_{ij}) &= \nabla_k\left(\frac{\vec{r}_k-\vec{r}_j}{|\vec{r}_k-\vec{r}_j|}u'(r_{kj})\right) \\ &= \frac{1}{|\vec{r}_k-\vec{r}_j|}\frac{\partial}{\partial r_{kj}}\big(u(r_{kj})\big) + \frac{\partial^2}{\partial r_{kj}^2}\big(u(r_{kj})\big) \\ &\quad + \frac{(\vec{r}_k-\vec{r}_j)(\vec{r}_k-\vec{r}_j)}{|\vec{r}_k-\vec{r}_j|^3}\frac{\partial}{\partial r_{kj}}\big(u(r_{kj})\big)\end{aligned}$$

where we can simplify the last term

$$\frac{(\vec{r}_k-\vec{r}_j)(\vec{r}_k-\vec{r}_j)}{|\vec{r}_k-\vec{r}_j|^3} = \frac{(\vec{r}_k-\vec{r}_j)(\vec{r}_k-\vec{r}_j)}{(\vec{r}_k-\vec{r}_j)(\vec{r}_k-\vec{r}_j)|\vec{r}_k-\vec{r}_j|} = \frac{1}{|\vec{r}_k-\vec{r}_j|}. \tag{69}$$

Finally we can write out the expression from equation (63)

$$\begin{aligned}\frac{\nabla_k^2\Psi_T}{\Psi_T} &= \frac{\nabla_k^2\Phi(\vec{r}_k)}{\Phi(\vec{r}_k)} + 2\frac{\nabla_k\Phi(\vec{r}_k)}{\Phi(\vec{r}_k)}\Big(\sum_{j\neq k}\frac{(\vec{r}_k-\vec{r}_j)}{r_{kj}}u'(r_{kj})\Big) \\ &+ \sum_{ij\neq k}\frac{(\vec{r}_k-\vec{r}_j)(\vec{r}_k-\vec{r}_j)}{r_{kj}r_{ki}}u'(r_{kj})u'(r_{ki}) \\ &+ \sum_{j\neq k}\Big(\frac{(\vec{r}_k-\vec{r}_j)}{r_{kj}}u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\Big)\end{aligned} \tag{70}$$

Now we can easily find a local energy expression using the general expression for the local energy:

$$\begin{aligned}E_L &= \frac{1}{\Psi_T}\Big(\sum_i\Big(-\frac{1}{2}\nabla_i^2 + V_{ext}(\vec{r}_i)\Big) + \sum_{i<j}V_{int}(\vec{r}_i,\vec{r}_j)\Big)\Psi_T \\ &= \sum_i\Big(-\frac{\nabla_i^2\Psi_T}{2\Psi_T} + V_{ext}(\vec{r}_i)\Big) + \sum_{i<j}V_{int}(\vec{r}_i,\vec{r}_j)\end{aligned} \tag{71}$$

Appendix B

This section shows the analytical calculations for the drift force F, as given by equation 39 for the non-interaction case.

8.2.1 N particles, 1 dimension

The no interaction, N particle trial wave function ψ_T is given by

$$\psi_T = \prod_i e^{-\alpha x_i^2} \tag{72}$$

The drift force F is then

$$F = \frac{2\nabla(\prod_i e^{-\alpha x_i^2})}{\prod_i e^{-\alpha x_i^2}} \tag{73}$$

The differentiation $\nabla \to \frac{d}{dx}$, as we operate in one dimension. The differentiation of ψ_T yields

$$\frac{d}{dx}\Big(\prod_i e^{-\alpha x_i^2}\Big) = \prod_i e^{-\alpha x_i^2}\sum_i(-2\alpha x_i) \tag{74}$$

This yields the drift force F

$$F = 2 \cdot \prod_i e^{\alpha x_i^2} \prod_i e^{-\alpha x_i^2} \sum_i (-2\alpha x_i) = -4\alpha \sum_i x_i \tag{75}$$

8.2.2 N particles, 3 dimensions

In the N particles, 3 dimensions case, the trial wave function ψ_T takes the following form

$$\psi_T = \prod_i e^{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)} \tag{76}$$

As we operate in three dimensions, $\nabla = (\frac{d}{dx}, \frac{d}{dy}, \frac{d}{dy})$. The differentiation in one dimension of ψ is shown in equation 74, and the results are similar for the differentiation with respect to y and z, the only difference being an extra factor β in the expression for $\frac{d\psi_T}{dz}$, due to the β in the expression for ψ_T. Thus the drift force F for N particles in three dimensions is

$$F = -4\alpha\Big(\sum_i xi, \sum_i y_i, \sum_i \beta z_i\Big) \tag{77}$$

Appendix C

In the theory part we claimed that the Hamiltonian could be written as

$$H = \sum_i \left(\frac{1}{2}\Big(-\nabla^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2\Big)\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{78}$$

for an elliptical harmonic oscillator potential with repulsive interaction. Let us start from scratch, where the unscalled Hamiltonian for an elliptical harmonic oscillator reads

$$H = \sum_i \left(-\frac{\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\Big(\omega_{HO}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2\Big)\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j).$$

We then scale the entire equation with respect to $\hbar\omega_{HO}$

$$\frac{H}{\hbar\omega_{HO}} = \sum_i \left(-\frac{\hbar}{2m\omega_{HO}}\nabla_i^2 + \frac{1}{2}\frac{m}{\hbar}\omega_{HO}(x_i^2 + y_i^2) + \frac{1}{2}\frac{m}{\hbar}\frac{\omega_z^2}{\omega_{HO}} z_i^2\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j)$$

where we can take $H' = H/\hbar\omega_{HO}$ as the dimensionless energy. Further we scale all the lengths in the same way

$$x_i^2 = (x_i')^2 \cdot a_{HO}^2 = (x_i')^2 \cdot \frac{\hbar}{m\omega_{HO}},$$

where we have used the relation $a_{HO} \equiv (\hbar/m\omega_{HO})^{1/2}$, and we finally obtain

$$H' = \sum_i \frac{1}{2}\bigg(-\nabla_i^2 + (x_i')^2 + (y_i')^2 + \frac{\omega_z^2}{\omega_{HO}^2}(z_i')^2 \bigg) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \qquad (79)$$

which is the Hamiltonian that we were hunting. We know how the factor in front of the z^2 factor is $\beta^2 = \gamma^2$, so $\beta = \gamma$ has to be equal to ω_z/ω_{HO}.