

FYS4411 - Computational Physics II

Project 1

Dorthea Gjestvang
Even Marius Nordhagen

March 23, 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 an upper bound limit of the ground state energy and onebody density for different number of particles. The results are benchmarked with analytical results obtained from the Gross-Pitaevskii (GP) equation and the articles in [2], [3] and [4].

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 when introducing the hard-sphere potential. The system behaves mostly as expected, and our results reproduce the analytical calculations from GP where they are valid. However, for a larger number of particles, we are unable to reproduce the mean energy per particle as seen in [2].

Contents

1	Introduction	4
2	Theory	4
2.1	Presentation of potential and trial wavefunction	4
2.2	Energy calculation	6
2.2.1	Numerical calculation of E_L	7
2.2.2	Gross-Pitaevskii equation	8
2.3	Onebody density	8
2.4	Scaling	9
2.5	Error estimation	10
3	Method	12
3.1	Variational Monte Carlo	12
3.2	Metropolis Algorithm	12
3.2.1	Brute force	13
3.2.2	Importance sampling	14
3.3	Minimization methods	15
3.3.1	Gradient descent	15
3.4	Blocking method	16
4	Code	17
4.1	Code structure	17
4.1.1	Test	18
4.2	Implementation	19
4.2.1	VMC	19
4.2.2	GD	19
5	Results	19
5.1	E as a function of the variational parameter α	20
5.2	E calculation and CPU-time	21
5.3	Acceptance ratios	22
5.3.1	Brute force dependence on step size	22
5.3.2	Importance sampling dependence on timestep	23
5.4	Variance calculation	24
5.5	VMC with repulsive interaction	25
5.5.1	Gradient Decent	25
5.5.2	Onebody density	26

6	Discussion	27
7	Conclusion	30
8	Appendix A: E_L calculation	32
8.1	Without repulsive interaction	32
8.1.1	One particle, one dimension	32
8.1.2	N particles, three dimensions	33
8.1.3	General	34
8.2	With repulsive interaction	34
9	Appendix B: Drift force F calculation	36
9.1	N particles, 1 dimension	36
9.2	N particles, 3 dimensions	37
10	Appendix C: Scaling and rewriting of \hat{H}	37
11	Appendix D: Solving Gross-Pitaevskii	38
12	References	40

1 Introduction

In this project, we study the Bose-Einstein condensation (BEC) of a dilute gas of ^{87}Rb atoms trapped in a harmonic oscillator. Bose-Einstein condensation occurs when bosonic 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 system is evaluated both for a spherical and elliptical harmonic oscillator trap. 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.

Using variational Monte Carlo (VMC) method with two different Metropolis algorithms, presented in section 3, and α as variational parameter, we find the onebody density and an limit on the ground state energy of the system. The optimal α is found using the gradient descent method. In section 5, the upper bound energy is presented along with the onebody density. In section 6 we discuss our results, and compare them to results from articles [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

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

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, where all dimensions have

the same scales and the harmonic oscillator frequency is ω_{ho} , or elliptical, where the z dimension has a different frequency ω_z .

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

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

The trial wavefunction is on the form

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

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

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

which is practical since

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

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 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. It is sufficient to study the ground state wave function only, since all particles occupy this state in a Bose-Einstein condensation.

The f presented above is the correlation wave function, which is

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

where a is a parameter describing the minimum distance allowed between particles in the trap. This is also denoted as the Jastrow factor, and it sets the wave function to zero when the distance between two particles is smaller than a , such that we get a hard-sphere repulsive potential.

2.2 Energy calculation

We want to find the ground state energy of our system for a given wave function Ψ_T . The energy is given as the expectation value of the Hamiltonian, given by equation 8.

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

By defining a quantity called the local energy E_L given by equation 9

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

the energy can be written as

$$E = \int |\Psi_T|^2 E_L d\vec{r} \quad (10)$$

This equation can be solved by calculating the local energy $E_{L,i}$ for each iteration in the Monte Carlo loop, summing them, and then divide by the number of Monte Carlo cycles:

$$E = \frac{\sum_i^M E_{L,i}}{M} \quad (11)$$

We want to calculate the energy E as a function of α for the trial wave function given in equation 4 by using Variational Monte Carlo (VMC) described in section 3.1.

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

$$E_L = dim \cdot N \cdot \alpha + \left(\frac{1}{2} - 2\alpha^2 \right) \sum_i \vec{r}_i^2, \quad (12)$$

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, which is also given in appendix A. 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 13.

$$E = \frac{1}{2}N \cdot \dim \cdot \hbar\omega_{ho}. \quad (13)$$

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

$$f(a, r_{ij}) = \exp \left(\sum_{i < j} u(r_{ij}) \right) \quad (14)$$

and doing a change of variables

$$\frac{\partial}{\partial \vec{r}_k} = \frac{\partial}{\partial \vec{r}_k} \frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \vec{r}_k} \frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}} \frac{\partial}{\partial r_{kj}} \quad (15)$$

one will end up with

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

which is also shown in appendix A. This is not a pretty expression, but will yield correct results for both the interacting and non-interacting 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 16, and calculate the local energy with a numerical approach where the second derivative needed to find the kinetic energy can be approximated by the three-point formula, see equation 17.

$$E_{L,i} = -\frac{\hbar^2}{2m} \frac{\nabla_i^2 \Psi_T}{\Psi_T} + V_{ext}(\vec{r}_i) = E_{k,i} + E_{p,i} \quad (16)$$

$$f''(x) \simeq \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (17)$$

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 energies are implemented, and in section 5.2, the CPU time for the analytical and numerical approach are compared for a various number of particles.

2.2.2 Gross-Pitaevskii equation

Since we study a dilute bose gas, we can use the Gross-Pitaevskii (GP) equation to estimate the energy density, also when including particle-particle interaction [9], [10]. The energy is then given by an integral over all the positions [3] as follows:

$$E_{\text{GP}}[\Psi] = \int d\vec{r} \left[\frac{\hbar^2}{2m} |\nabla \Psi(\vec{r})|^2 + V_{\text{ext}}(\vec{r}) |\Psi|^2 + \frac{2\pi\hbar^2 a}{m} |\Psi|^4 \right]. \quad (18)$$

This integral can easily be solved analytically when we ignore the interaction, and for three dimensions we get

$$E_{\text{GP}}(\alpha, \beta, N) = N \cdot \left(1 + \frac{\beta}{2} \right) \left(\frac{1}{4\alpha} + \alpha \right) \quad (19)$$

which is derived in appendix D.

This expression is valid for few particles and small hard-sphere diameter. As the average energy per particle increases as we increase N , the expression is no longer valid for a higher number of particles. For the interacting case, we could solve the integral using a numerical method, for instance Monte Carlo integration. We will not implement this, but rather benchmark our results against the solution of GP when ignoring interaction.

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. 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 :

$$\rho_i = \int_{-\infty}^{\infty} d\vec{r}_1 \dots d\vec{r}_{i-1} d\vec{r}_{i+1} \dots d\vec{r}_N |\Psi(\vec{r}_1, \dots, \vec{r}_N)|^2. \quad (20)$$

For the non-interacting case this integral can be solved analytically, as shown in equation 21. 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.

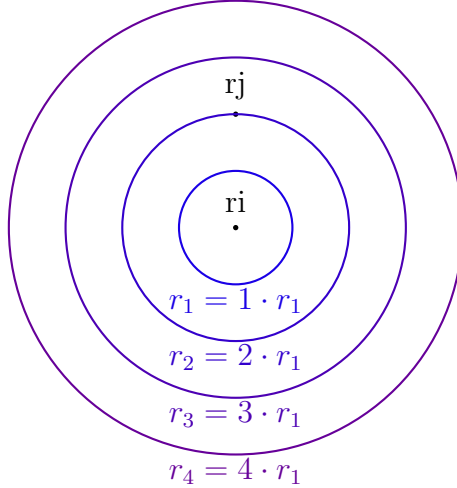


Figure 1: One can find the onebody density by dividing the volume around a particle with coordinates \vec{r}_i into bins and then count the number of particles, here illustrated in two dimensions.

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 21.

$$\rho_i(x_i, y_i, z_i) = \left(\frac{2\alpha}{\pi}\right)^{3/2} \beta^{1/2} e^{-2\alpha(x_i^2 + y_i^2 + \beta z_i^2)}. \quad (21)$$

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 13. The equivalent dimensionless

equation can then be written as

$$E' = \frac{N \cdot \dim}{2} \quad (22)$$

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

$$r'_i = \frac{r_i}{a_{ho}} = r_i \cdot \sqrt{\frac{m\omega_{ho}}{\hbar}}, \quad (23)$$

and the Hamiltonian turns into

$$H = \frac{1}{2} \sum_i \left(-\nabla^2 + \vec{r}_i^2 \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j) \quad (24)$$

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

$$H = \frac{1}{2} \sum_i \left(-\nabla^2 + x_i^2 + y_i^2 + \beta^2 z_i^2 \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j) \quad (25)$$

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 data point. 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 θ with n measured points, $\{x_1, \dots, x_n\}$, the sample mean $\langle x_\theta \rangle$ of the experiment is defined as shown in equation 26.

$$\langle x_\theta \rangle = \frac{1}{n} \sum_{k=1}^n x_{\theta,k} \quad (26)$$

The corresponding sample variance σ_θ is then defined as

$$\sigma_\theta^2 = \frac{1}{n} \sum_{k=1}^n (x_{\theta,k} - \langle x_\theta \rangle)^2 \quad (27)$$

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

$$\langle x_m \rangle = \frac{1}{n} \sum_{k=1}^n \langle x_\theta \rangle. \quad (28)$$

The total variance is then

$$\sigma_m^2 = \frac{1}{m} \sum_{\theta=1}^m (\langle x_\theta \rangle - \langle x_m \rangle)^2. \quad (29)$$

This can be reduced to

$$\sigma_m^2 = \frac{\sigma^2}{n} + \text{covariance term}, \quad (30)$$

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

$$\sigma^2 = \frac{1}{mn} \sum_{\theta=1}^m \sum_{k=1}^n (x_{\theta,k} - \langle x_m \rangle)^2 \quad (31)$$

and the covariance is the linear correlation between the measured points. The definition of the covariance is shown in equation 32.

$$\text{cov}(x, y) = \frac{1}{n^2} \sum_i \sum_{j>i} (x_i - x_j)(y_i - y_j). \quad (32)$$

A common simplification is to reduce equation 30 to the following:

$$\sigma^2 \approx \langle x^2 \rangle - \langle x \rangle^2 \quad (33)$$

This equation, however, does not take into account the covariance term from equation 30, and as the covariance term is added to the expression for

the variance, equation 33 will underestimate the uncertainty σ for positive covariances.

A direct implementation of equation 30 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.

$$\vec{R}_{new} = \vec{R} + r \cdot \text{step}. \quad (34)$$

Both the direction and particles moves are randomly chosen, so with a plain VMC implementation the particles will move at random, independent from 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 towards 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

$$W_{i \rightarrow j} = T_{i \rightarrow j} \cdot A_{i \rightarrow j} \quad (35)$$

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

$$P_i^{(n)} = \sum_j \left[P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} + P_i^{(n-1)} T_{i \rightarrow j} (1 - A_{i \rightarrow j}) \right] \quad (36)$$

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 $\sum_j T_{i \rightarrow j} = 1$, so we can rewrite this as

$$P_i^{(n)} = P_i^{(n-1)} + \sum_j \left[P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - P_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} \right]. \quad (37)$$

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

$$\sum_j \left[p_j T_{j \rightarrow i} A_{j \rightarrow i} - p_i T_{i \rightarrow j} A_{i \rightarrow j} \right] = 0. \quad (38)$$

Rearranging, we obtain the useful result

$$\frac{A_{j \rightarrow i}}{A_{i \rightarrow j}} = \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}} \quad (39)$$

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 $T_{i \rightarrow j} = T_{j \rightarrow i}$, and then end up with the plain ratio

$$w = \frac{P(\vec{R}_{new})}{P(\vec{R})} = \frac{|\Psi_T(\vec{R}_{new})|^2}{|\Psi_T(\vec{R})|^2}. \quad (40)$$

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:

$$\text{New position: } \begin{cases} \text{accept} & \text{if } w > r \\ \text{reject} & \text{if } w \leq r. \end{cases} \quad (41)$$

3.2.2 Importance sampling

The importance sampling technique is often referred to as Metropolis-Hastings algorithm. The approach is the same as for the brute force Metropolis algorithm, but we will end up with a slightly more complicated acceptance 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

$$\frac{\partial P(R, t)}{\partial t} = D \frac{\partial}{\partial R} \left(\frac{\partial}{\partial R} - F \right) P(R, t). \quad (42)$$

where F is the drift force given by equation 43 and D is the diffusion coefficient, in this case equal to 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.

$$F(R) = \frac{2\nabla\psi_T}{\psi_T} \quad (43)$$

Even though the probability density function can give a lot of useful information, an equation describing the motion of such a 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

$$\frac{\partial R(t)}{\partial t} = DF(R(t)) + \eta \quad (44)$$

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

$$R_{new} = R + DF(R)\Delta t + \xi\sqrt{\Delta t} \quad (45)$$

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

$$\begin{aligned} G(R_{new}, R, \Delta t) &= \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp[-(R_{new} - R - D\Delta t F(R))^2 / 4D\Delta t] \\ &= T_{R \rightarrow R_{new}} \end{aligned} \quad (46)$$

and the acceptance criteria becomes

$$r < \frac{G(R, R_{new}, \Delta t) |\Psi_T(R_{new})|^2}{G(R_{new}, R, \Delta t) |\Psi_T(R)|^2}. \quad (47)$$

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 (GD), which reads

$$\alpha^+ = \alpha - \eta \cdot \frac{d\langle E(\alpha) \rangle}{d\alpha}. \quad (48)$$

where α^+ is the updated α and η is a 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 $\frac{d\langle E(\alpha) \rangle}{d\alpha}$ is smaller than a given tolerance, or the energy fluctuates around a value are smaller than a tolerance, and thus changes minimally.

To implement equation 48, we need an expression for the derivative of E with respect to alpha:

$$\bar{E}_\alpha = \frac{d\langle E(\alpha) \rangle}{d\alpha}. \quad (49)$$

By using the expression for the expectation value for the energy $\langle E(\alpha) \rangle$ in equation 50

$$\langle E(\alpha) \rangle = \frac{\langle \psi_T(\alpha) | H | \psi_T(\alpha) \rangle}{\langle \psi_T(\alpha) | \psi_T(\alpha) \rangle} \quad (50)$$

and applying the chain rule of differentiation, it can be shown that equation 49 is equal to equation 51

$$\bar{E}_\alpha = 2 \left[\langle E_L(\alpha) \frac{\bar{\psi}_\alpha}{\psi_\alpha} \rangle - \langle E_L(\alpha) \rangle \langle \frac{\bar{\psi}_\alpha}{\psi_\alpha} \rangle \right] \quad (51)$$

where

$$\bar{\psi}_\alpha = \frac{d\psi(\alpha)}{d\alpha}. \quad (52)$$

The algorithm of this minimization method is thus as follows:

```

for (max number of iterations with minimizing)

    do M Monte Carlo cycles
    calculate E and dE/dalpha

    Check if dE/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 32.

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 $\{x_1, \dots, x_i, \dots, x_n\}$ 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

$$\sigma^2 = \frac{1}{n-1} \sum_i (x_i - m) \quad (53)$$

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

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

$$x'_i = \frac{1}{2}(x_{2i} + x_{2i+1}) \quad (54)$$

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

$$\hat{\sigma} = \frac{\sigma_1 + \dots + \sigma_{n'}}{n'} \quad (55)$$

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 30, 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

The focus of this project is developing a code to make VMC run correctly, and in this section we will explain briefly how the code works and which implementation techniques we have used. All quantities used in the implementation are dimensionless.

4.1 Code structure

To keep the program neat, specific parts were placed in specific files. The file `wavefunction.cpp` contains the class `Wavefunction`, which both sets up the wave function and calculate the local energy given a wave function. The file `metropolis.cpp` contains the Variational Monte Carlo loop, and the file `gd.cpp` handles the calculation of the optimal variational parameter α . 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. The user only has to deal with `main.cpp`, as all choices of parameters are to be specified here. In figure 2 we give an overview of the structure, where the lines indicate which scripts that communicate.

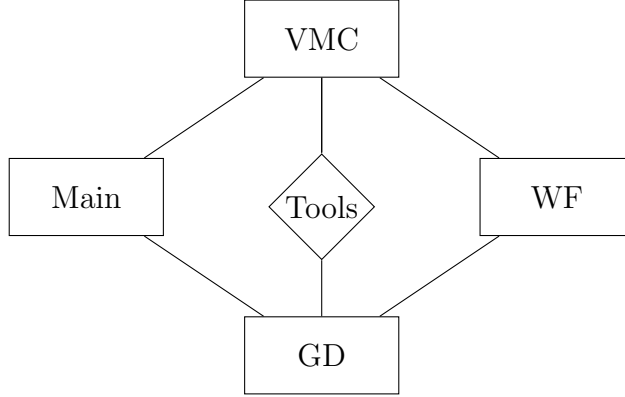


Figure 2: An overview of the code structure. `metropolis.cpp` is referred to as VMC, `gd.cpp` is GD, `wavefunction.cpp` is WF, `tools.cpp` is Tools and obviously `main.cpp` is Main. See text for further description.

A standard VMC run goes as follows: the desired simulation case is set up in `main.cpp` (Main) by specifying the given variables. `metropolis.cpp` (VMC) is called from Main, which calls `wavefunction.cpp` (WF) to set up the class `WaveFunction`. The Monte Carlo loop begins, where the move of one random particle is proposed for each iteration. Depending of whether brute force or importance sampling have been chosen, either WF or `tools.cpp` (Tools) is called to obtain the transition probability, and the move is either accepted or rejected. VMC calls WF to obtain the local energy of the new position. After M iterations, the estimation of the energy is printed.

When running the gradient decent method, then Main calls `gd.cpp` (GD). A start value for α is given in the program, and the Monte Carlo loop is conducted as described above. After the loop, the estimated E -value is printed. Based on the parameter $\frac{dE}{d\alpha}$, a new value for α is chosen. After several MC loops for different α , the optimal α is chosen.

4.1.1 Test

To make sure the program behaves as expected, we have implemented tests. All tests are functions, and can be found in the `test.cpp` file.

The test "test EL" checks the output estimated energy after the Monte Carlo loops for a given α in the no interaction case, and compares it to the analytical energy obtained from the Gross-Pitaevskii equation from equation 19. If the difference between the two are larger than a given tolerance, an error is raised.

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

```
Set up position matrix of N particles in dim dimensions
Initialize class WaveFunction

for M Monte Carlo iterations
    move random particle
    accept or reject move based on transition probability
    calculate energy of new position
    update sum of E.L

print estimated E
```

4.2.2 GD

```
first choice of alpha, number of gradientdecent-iterations T and tolerance
    ↪ is specified

for T iterations
    for M Monte Carlo cycles
        move random particle
        accept or reject move based on transition probability
        calculate energy of new position
        update sum of E.L and dE/dalpha

    print E and dE/dalpha

    test if optimal alpha -> if optimal, break loop and print optimal
        ↪ alpha

    based on dE/d alpha, suggest new alpha
```

5 Results

We will first present the results obtained without repulsive interaction, and throughout 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 energy. In this part we will also turn on the elliptical harmonic oscillator potential, using $\beta = \sqrt{8}$ is mentioned above.

5.1 E as a function of the variational parameter α

In search for 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 is presented in figure 4, and we see that the energy minimum and the variance minimum share the same α .

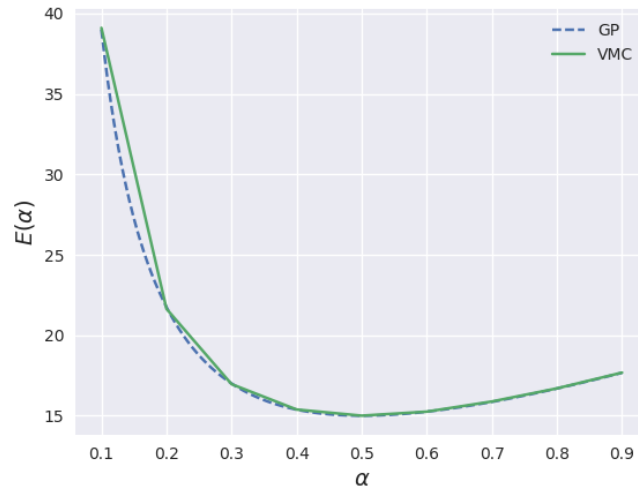


Figure 3: The energy E calculated with the brute force Metropolis algorithm, as a function of the variational parameter α in the interval $[0.1, 0.9]$ (continuous line). For each α we used $1e6$ Monte Carlo cycles in three spatial dimensions with a step length of 1.0. We have also added the analytical results found from the Gross-Pitaevskii equation, which is the dashed line.

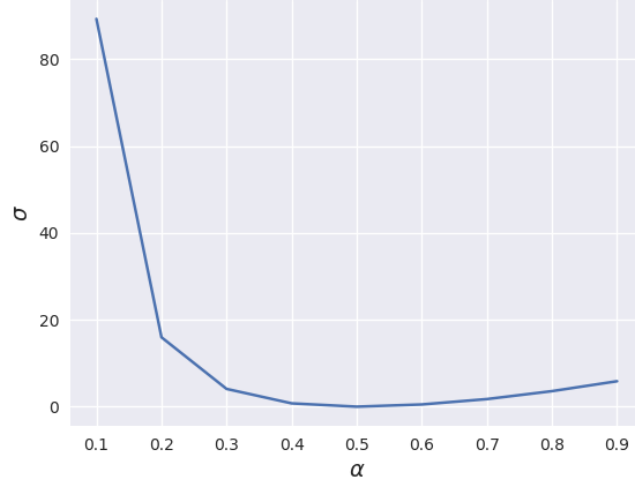


Figure 4: The variance of the energy E calculated with the brute force Metropolis algorithm, as a function of the variational parameter α associated with the energy plot in figure 3. The variance is calculated with equation 33.

5.2 E calculation and CPU-time

For the brute force Metropolis algorithm we developed both an analytical and a numerical method to calculate the 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 energy calculated without hard-sphere interaction with brute force Metropolis algorithm, for both the analytical and numerical energy calculation. The variational parameter α is 0.5, we study three dimensions, the step length is 1.0, the number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the calculated energies. The results are compared to the exact answer, obtained from equation 13.

N	Analytical		Numerical		Exact
	$\langle E \rangle [\hbar\omega_{ho}]$	CPU-time [s]	$\langle E \rangle [\hbar\omega_{ho}]$	CPU-time [s]	$E [\hbar\omega_{ho}]$
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 energy calculated without hard-sphere interaction with Metropolis-Hastings algorithm. The variational parameter α is 0.5, we study three dimensions, the time step is 1.0 and the number of Monte Carlo cycles is fixed to $M = 1e6$. The results are compared to the exact answer, obtained from equation 13.

N	Analytical		Exact
	$\langle E \rangle [\hbar\omega_{ho}]$	CPU-time [s]	$E [\hbar\omega_{ho}]$
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 step size

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

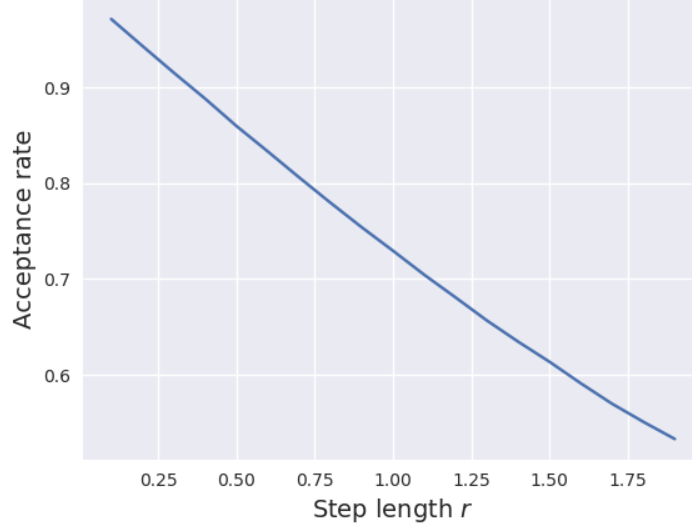


Figure 5: Acceptance ratio for different choices of the step length in the interval $[0.1, 1.9]$ when using in the brute force Metropolis algorithm. We use $1e6$ Monte Carlo cycles, the other parameters have no impact.

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.

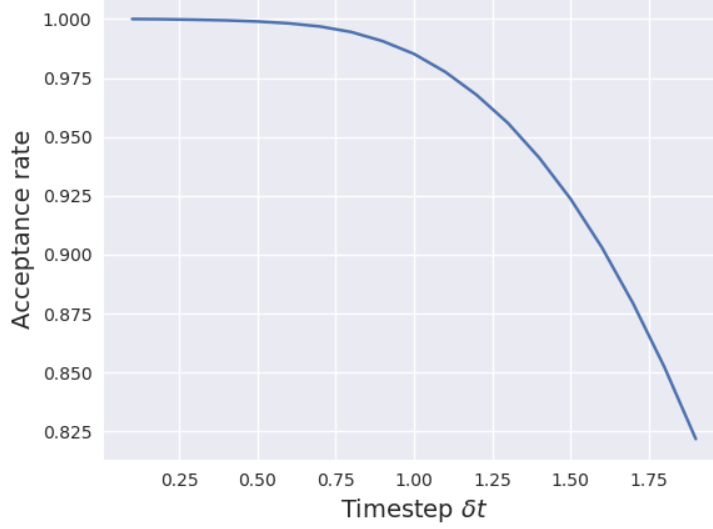


Figure 6: Acceptance ratio for different choices of the time step δt in the interval $[0.1, 1.9]$ when using in the Metropolis-Hastings algorithm. We use $1e6$ Monte Carlo cycles, the other parameters have no impact.

We observe that the acceptance rate for the brute force Metropolis algorithm decreases almost linearly as increasing the step length, while the Metropolis-Hastings algorithm has a drop that increases gradually. The acceptance rate is overall higher for Metropolis-Hastings.

5.4 Variance calculation

In table 3, we present the results of the 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 33.

Table 3: The energy calculated without hard-sphere interaction with the Metropolis-Hastings algorithm, this time with proper evaluation of the statistical error. The calculations are run in three dimensions with 1e6 Monte Carlo cycles, the variational parameter $\alpha = 0.5$ and a step length of 1.0.

N	$\langle E \rangle [\hbar\omega_{ho}]$	σ^2 Blocking	$\sigma^2 \approx \langle E^2 \rangle - \langle E \rangle^2$
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

We now turn on the interaction and the elliptical trap (setting $a = 0.0043$ and $\beta = \sqrt{8}$). 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 energy E .

Table 4: The energy E calculated for different α 's in the interval $[0.2, 0.8]$, with $a = 0.0043$ (interaction), $\beta = \sqrt{8}$ (elliptical trap) and with the brute force algorithm. The calculations are run in three dimensions with 1e6 Monte Carlo cycles and a step length of 1.0. We compare the VMC results to the results found by the Gross-Pitaevskii equation (GP) for 1, 10 and 100 particles. The variances are calculated using 33.

α	N=1		N=10		N=100	
	VMC	GP	VMC	GP	VMC	GP
0.3	2.7(1.2)	2.7361	27.6(3.8)	27.361	359(14)	273.61
0.4	2.47(50)	2.4746	24.9(1.5)	24.746	354.6(8.7)	247.46
0.5	2.4142(24)	2.4142	24.264(32)	24.142	371.0(7.4)	241.42
0.6	2.45(40)	2.4545	24.7(1.3)	24.545	395.6(7.9)	245.45
0.7	2.55(77)	2.5522	25.6(2.4)	25.522	425(10)	255.22

5.5.1 Gradient Decent

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

Table 5: The optimal variation parameter α for 1, 10 and 100 particles. The local energies and the "energy slope" is also presented in the table. We used $a = 0.0043$ with hard-sphere interaction, elliptical trap and $\beta = \sqrt{8}$, with the brute force algorithm. The calculations are run in for three dimensions with $1e6$ Monte Carlo cycles with a step length of 1.0. The variances are calculated using 33.

N	α	E	$ \partial E/\partial \alpha $
1	0.50000	2.4142(73)	7.2134e-06
10	0.50015	24.264(22)	4.7969e-05
100	0.53379	379.4(9.9)	4.1789e-03

5.5.2 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. With the optimal parameter α for the ground state wave function and for both the interacting and the non-interacting case, which was found to be about $\alpha = 0.5$ in both cases, the onebody density plot is found in figure 7.

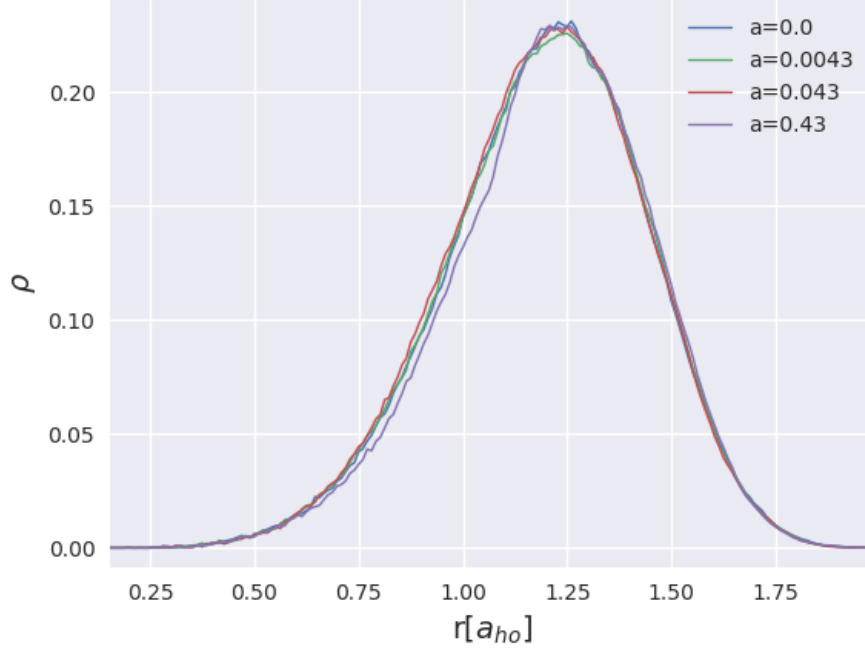


Figure 7: The onebody density of the bosonic system, which is run with $\alpha = 0.5$ for $a = 0$, $a = a_{Rb} = 0.0043$ and $a = 10 \cdot a_{Rb} = 0.043$. The radius is given in units of a_{ho} along the x-axis, and the density is in units of a_{ho}^{-3} . We conduct the investigations in elliptic traps, with 10 particles, 3 dimensions and 1e6 Monte Carlo cycles.

From the onebody density plot, the only difference between the graphs is the amplitude, which seems lower the larger scattering length we add. The graphs have the same shape as the wave function.

6 Discussion

For the case without interaction and spherical harmonic oscillator we see from figure 3 that the minimum of the energy is found when the variational parameter $\alpha = 0.5$. We also observe that $\alpha = 0.5$ gives the minimal variance, and seen from figure 4. The energy and the variance explode when α is approaching zero.

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 13. Also the numerically calculated E_L , as described in section 2.2.1, reproduces the exact answers with the same precision 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 17, 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 Hastings 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. The transition probability $T_{i \rightarrow j}$ and $T_{j \rightarrow i}$ are also calculated, as they are no longer assumed to be identical, as they are for brute force. While these calculations 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 linearly 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 33 to a proper variance calculation using the blocking method, by evaluation equation 30. As expected, the simplified variance calculation underestimates the statistical error, as it does not include the covariance term in equation 30. 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 we get some differences between the variational Monte Carlo and the Gross-Pitaevskii calculations. Gross-Pitaevskii (GP) only holds for dilute bose gases, and since the gas gets less dilute the more particles we add, we will expect some deviations when the number of particles increase. Nonetheless, for the interacting case we study a maximum of 100 particles, so this should be a small effect, and we expect the exact GP equation to be accurate for our case. Anyway, we have not solved GP for the interacting case, so the GP benchmark should underestimate the obtained energy, which is also the case.

We can also compare the results to [2], where they have plotted energy per particle as a function of number of particles. According to that plot, the energy per particle should increase rapidly when particles are added to a small number of particles, something that we also observe when going from 10 to 100 particles and partly when going from 1 to 10 particles. The exact energy difference is though hard to read from the graph, but our energies are possibly a bit larger than they should be. We also see that the lowest energy for 100 particles is when $\alpha = 0.4$. This differs from the case without interaction, where the optimal α is 0.5.

We find the optimal α using the gradient descent method, and for one and ten particles the results match the results obtained when running VMC for various α 's, see table 4 and 5. For 100 particles the α obtained is not the same as found in 4, and indicates that the gradient descent implementation gives wrong answers for many particles. According to the gradient decent method, the optimal α tends to increase when adding particles. The fact that α changes is not surprising, as the optimal wave function should change when

adding particles with interaction. What value one obtains as the optimal α with GD will be dependent on whether if one starts with an overestimated or an underestimated guess, and we have therefore tried both and decreased the step length η to achieve an accurate optimal α .

We observe from table 4 and 5 that VMC and GD do not yield the same optimal α , even though they yield about the same energy for the same α . This might be because we chose the wrong starting α -value for the gradient decent method. As described in section 3, GD is not an accurate method, and is heavily dependent on the initial α -guess. Another reason for this might be that the test in GD checking if we have reached the optimal α , is too innaccurate, and thus returned the wrong value of α . The error might also be in the VMC calculation, as we have no proper benchmark to check these, but results presenten in [2] suggest that our energy per particle is too large, as mentioned above.

In the onebody density plot we see that the densities has gaussian shapes, which come from the wave functions. The peak is moved to slightly higher energies when increasing a . This is as expected, as the bosons for higher a are forced further apart, though we would have expected a larger difference.

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 on 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 33. This is as expected.

With interaction we got reasonable results for one and ten particles compared to calculations based on the Gross-Pitaevskii equation. For 100 par-

ticles the energies were probably higher than they should be. The optimal variational parameter α obtained from VMC did not match the optimal α gotten from GD, and perhaps we have a bug in the code which appears for a large number of particles.

The one body density behaves as expected for different choices of a .

8 Appendix A: E_L calculation

In the theory section we presented the analytical expressions for the local energy, and in this appendix we go through 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:

$$\Psi_T(\vec{r}) = \prod_i^N e^{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)} \quad (56)$$

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 56, the trial wave function for one particle and one dimension is as follows:

$$\Psi_T(x) = e^{-\alpha x^2} \quad (57)$$

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

$$\frac{d\Psi_T}{dx} = -2\alpha x e^{-\alpha x^2} \quad (58)$$

$$\frac{d^2\Psi_T}{dx^2} = -2\alpha e^{-\alpha x^2} + 4\alpha^2 x^2 e^{-\alpha x^2}. \quad (59)$$

To get a neat expression, we use the dimensionless Hamiltonian with spherical harmonic oscillator potential, from equation 24, and obtain

$$\begin{aligned} E_L(\alpha) &= -\frac{1}{2}(-2\alpha + 4\alpha^2 x^2) + \frac{1}{2}x^2 \\ &= \alpha + \left(\frac{1}{2} - 2\alpha^2\right)x^2 \end{aligned} \quad (60)$$

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 56, and the Hamiltonian will be as listed in equation 24 with the dimensionless spherical harmonic oscillator potential $\frac{1}{2}r_i^2$.

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

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) \quad (61)$$

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

$$\frac{\partial \Psi_T}{\partial r_j} = -2\alpha r_j \exp \left[-\alpha \left(\sum_i r_i^2 \right) \right]. \quad (62)$$

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

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

and we obtain

$$\frac{\nabla_j^2 \Psi_T}{\Psi_T} = -6\alpha + 4\alpha^2 r_j^2. \quad (63)$$

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

$$\begin{aligned} E_L(\alpha) &= \sum_j -\frac{1}{2}(-6\alpha + 4\alpha^2 r_j^2) + \frac{1}{2}r_j^2 \\ &= 3N\alpha + \left(\frac{1}{2} - 2\alpha^2\right) \sum_{j=1}^N r_j^2 \end{aligned} \quad (64)$$

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

$$E_L(\alpha) = \dim \cdot N \cdot \alpha + \left(\frac{1}{2} - 2\alpha^2\right) \sum_{j=1}^N r_j^2. \quad (65)$$

A similar expression for the elliptical trap can be found to be

$$E_L(\alpha) = 2\alpha + \alpha\beta + \left(\frac{1}{2} - 2\alpha^2\right) \sum_{j=1}^N r_j^2. \quad (66)$$

in three dimensions.

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 $\exp(u(r_{ij}))$ such that the total trial wavefunction becomes

$$\Psi_T(\vec{r}_1, \dots, \vec{r}_N) = \left[\prod_i \Phi(\vec{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij})\right) \quad (67)$$

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

$$\frac{\nabla_k^2 \Psi_T(\vec{r})}{\Psi_T(\vec{r})}, \quad (68)$$

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

$$\begin{aligned} \nabla_k \Psi_T(\vec{r}) &= \nabla_k \Phi(\vec{r}_k) \left[\prod_{i \neq k} \Phi(\vec{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij})\right) \\ &+ \prod_i \Phi(\vec{r}_i) \exp\left(\sum_{i<j} u(r_{ij})\right) \sum_{j \neq k} \nabla_k u(r_{ij}) \end{aligned} \quad (69)$$

For the second derivative we get five terms in total where two of them are cross terms and therefore equal.

$$\begin{aligned}
\nabla_k^2 \Psi_T(\vec{r}) = & \nabla_k^2 \Phi(\vec{r}_k) \left[\prod_{i \neq k} \Phi(\vec{r}_i) \right] \exp \left(\sum_{i < j} u(r_{ij}) \right) \\
& + \nabla_k \Phi(\vec{r}_k) \left[\prod_{i \neq k} \Phi(\vec{r}_i) \right] \exp \left(\sum_{i < j} u(r_{ij}) \right) \sum_{j \neq k} \nabla_k u(r_{ij}) \\
& + \nabla_k \Phi(\vec{r}_k) \left[\prod_{i \neq k} \Phi(\vec{r}_i) \right] \exp \left(\sum_{i < j} u(r_{ij}) \right) \sum_{j \neq k} \nabla_k u(r_{ij}) \quad (70) \\
& + \prod_i \Phi(\vec{r}_i) \exp \left(\sum_{i < j} u(r_{ij}) \right) \sum_{i \neq k} \nabla_k u(r_{ij}) \sum_{j \neq k} \nabla_k u(r_{ij}) \\
& + \prod_i \Phi(\vec{r}_i) \exp \left(\sum_{i < j} u(r_{ij}) \right) \sum_{j \neq k} \nabla_k^2 u(r_{ij})
\end{aligned}$$

To simplify this, we do a change of variables

$$\frac{\partial}{\partial \vec{r}_k} = \frac{\partial}{\partial \vec{r}_k} \frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \vec{r}_k} \frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}} \frac{\partial}{\partial r_{kj}} \quad (71)$$

where we have used that

$$\frac{\partial r_{kj}}{\partial \vec{r}_k} = \frac{\vec{r}_k - \vec{r}_j}{|\vec{r}_k - \vec{r}_j|} = \frac{\vec{r}_k - \vec{r}_j}{r_{kj}}. \quad (72)$$

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

$$\nabla_k u(r_{ij}) = \frac{\partial r_{kj}}{\partial \vec{r}_k} \frac{\partial}{\partial r_{kj}} (u(r_{ij})) = \frac{\vec{r}_k - \vec{r}_j}{|\vec{r}_k - \vec{r}_j|} u'(r_{kj}) \quad (73)$$

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

where we can simplify the last term

$$\frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_j)}{|\vec{r}_k - \vec{r}_j|^3} = \frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_j)}{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_j)|\vec{r}_k - \vec{r}_j|} = \frac{1}{|\vec{r}_k - \vec{r}_j|}. \quad (74)$$

Finally we can write out the expression from equation (68)

$$\begin{aligned}
\frac{\nabla_k^2 \Psi_T}{\Psi_T} &= \frac{\nabla_k^2 \Phi(\vec{r}_k)}{\Phi(\vec{r}_k)} + 2 \frac{\nabla_k \Phi(\vec{r}_k)}{\Phi(\vec{r}_k)} \left(\sum_{j \neq k} \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}} u'(r_{kj}) \right) \\
&+ \sum_{ij \neq k} \frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_i)}{r_{kj} r_{ki}} u'(r_{kj}) u'(r_{ki}) \\
&+ \sum_{j \neq k} \left(\frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}} u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right)
\end{aligned} \tag{75}$$

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

$$\begin{aligned}
E_L &= \frac{1}{\Psi_T} \left(\sum_i \left(-\frac{1}{2} \nabla_i^2 + V_{ext}(\vec{r}_i) \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j) \right) \Psi_T \\
&= \sum_i \left(-\frac{\nabla_i^2 \Psi_T}{2 \Psi_T} + V_{ext}(\vec{r}_i) \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j)
\end{aligned} \tag{76}$$

9 Appendix B: Drift force F calculation

This section shows the analytical calculations for the drift force F , as given by equation 43 for the non-interaction case.

9.1 N particles, 1 dimension

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

$$\psi_T = \prod_i e^{-\alpha x_i^2} \tag{77}$$

The drift force F is then

$$F = \frac{2 \nabla (\prod_i e^{-\alpha x_i^2})}{\prod_i e^{-\alpha x_i^2}} \tag{78}$$

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

$$\frac{d}{dx} \left(\prod_i e^{-\alpha x_i^2} \right) = \prod_i e^{-\alpha x_i^2} \sum_i (-2\alpha x_i) \tag{79}$$

This yields the drift force F

$$F = 2 \cdot \prod_i e^{\alpha x_i^2} \prod_i e^{-\alpha x_i^2} \sum_i (-2\alpha x_i) = -4\alpha \sum_i x_i \quad (80)$$

9.2 N particles, 3 dimensions

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

$$\psi_T = \prod_i e^{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)} \quad (81)$$

As we operate in three dimensions, $\nabla = (\frac{d}{dx}, \frac{d}{dy}, \frac{d}{dz})$. The differentiation in one dimension of ψ is shown in equation 79, 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

$$F = -4\alpha \sum_i (x_i, y_i, \beta z_i) \quad (82)$$

10 Appendix C: Scaling and rewriting of \hat{H}

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

$$H = \sum_i \left(\frac{1}{2} \left(-\nabla^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2 \right) \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j) \quad (83)$$

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

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

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

$$H' = \sum_i \frac{1}{2} \left(-\nabla_i^2 + (x'_i)^2 + (y'_i)^2 + \frac{\omega_z^2}{\omega_{ho}^2} (z'_i)^2 \right) + \sum_{i < j} V_{int}(\vec{r}_i, \vec{r}_j) \quad (84)$$

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} .

11 Appendix D: Solving Gross-Pitaevskii

To check if the VMC results are correct, we need a benchmark, and for a dilute bose gas the Gross-Pitaevskii equation is a natural choice. We could have solved the equation for the interacting case using numerical methods, but it would require significantly more effort and we take care of the non-interacting case instead. We start with the dimensionless brother of equation 18,

$$E_{GP}[\Psi] = \int d\vec{r} \left[\frac{1}{2} |\nabla \Psi(\vec{r})|^2 + V_{ext}(\vec{r}) |\Psi|^2 + 2\pi a |\Psi|^4 \right]. \quad (85)$$

where

$$V_{ext}(\vec{r}) = \frac{1}{2} (x_1^2 + y_1^2 + \beta^2 z_1^2 + \dots + x_N^2 + y_N^2 + \beta^2 z_N^2) \quad (86)$$

for N particles in an 3D elliptical trap. The normalized wave function without the Jastow factor reads

$$\Psi(\vec{r}) = \left(\frac{2\alpha}{\pi} \right)^{3N/4} \cdot \beta^{N/4} \exp \left(-\alpha(x_1^2 + y_1^2 + \beta^2 z_1^2 + \dots + x_N^2 + y_N^2 + \beta^2 z_N^2) \right). \quad (87)$$

This gives us the following expression for the gradient of the wave function squared

$$|\nabla \Psi(\vec{r})|^2 = 4\alpha^2 (x_1^2 + y_1^2 + \beta^2 z_1^2 + \dots + x_N^2 + y_N^2 + \beta^2 z_N^2) |\Psi(\vec{r})|^2. \quad (88)$$

Since we ignore the interaction, $a = 0$, we do not need to take the last term into account. The intergral to solve is then

$$E_{GP} = \left(\frac{1}{2} + 2\alpha^2 \right) \int_{-\infty}^{\infty} (x_1^2 + y_1^2 + \beta^2 z_1^2 + \dots + x_N^2 + y_N^2 + \beta^2 z_N^2) |\Psi|^2 d\vec{r}$$

which is gaussian and can be found in a collection of integrals:

$$\int_{-\infty}^{\infty} e^{-(ax^2+b)} dx = \frac{1}{2a} \sqrt{\frac{\pi}{a}} e^{-b}. \quad (89)$$

With some straightforward computations one will find

$$\begin{aligned} E_{\text{GP}}(\alpha, \beta, N) &= \left(\frac{1}{2} + 2\alpha^2\right) \left(\frac{2\alpha}{\pi}\right)^{3N/2} \beta^{N/2} \frac{1}{4\alpha} \sqrt{\frac{\pi}{2\alpha}}^{3N} (2 + \beta) \cdot N \cdot \beta^{-N/2} \\ &= N \cdot \left(\frac{1}{4\alpha} + \alpha\right) \cdot \left(1 + \frac{\beta}{2}\right). \end{aligned} \quad (90)$$

This gives the correct analytical results for the non-interacting case both in the spherical trap ($E = 0.5N \cdot \dim$ for $\alpha = 0.5$) and the elliptical trap ($E = N(1 + \beta/2)^1$ for $\alpha = 0.5$ in 3D).

¹We start from the exact energy formula for a harmonic oscillator, $E_n = \hbar\omega(n + 1/2)$, and study how the ground state energy changes if the z-component has different frequency ω_z from the other components, $E_0 = \hbar(\omega_{ho} + \omega_z/2)$. If we then scale with respect to $\hbar\omega_{ho}$ as done before, we get $E' = 1 + \beta/2$ for one particle. For our particular project, we have $\beta = \sqrt{8}$, and get an ground state energy $1 + \sqrt{2}$ per particle.

12 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] J. Emspak, *States of Matter: Bose-Einstein Condensate*, LiveScience, (2016). <https://www.livescience.com/54667-bose-einstein-condensate.html> Downloaded March 15th 2018.
- [6] S. 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).
- [9] E. P. Gross, *Structure of a quantized vortex in boson systems*, Il Nuovo Cimento, **20** (3): 454–457, (1961).
- [10] L. P. Pitaevskii, *Vortex lines in an imperfect Bose gas*, Sov. Phys. JETP. **13** (2): 451–454, (1961).