Project 1, FYS4411

Erik Alexander Sandvik & Simon Schrader

March 26, 2021

Abstract

With the recent experimental advancements in Bose-Einstein condensation, the need to describe the collective behaviour of ground-state bosons arises. In this article, we used the Variational Principle to find an approximate wave function and its average energy $\langle H \rangle$ for a weakly interacting hard-sphere boson gas at $T = 0\text{K}$ trapped in an elliptic Harmonic Oscillator potential.

We implemented the Metropolis algorithm with and without importance sampling as well as Gradient Descent and verified their correct implementation for non-interacting particles. Using Gradient Descent, we managed to find the correct variational parameter α for the trial wave function $\psi(r) = e^{-\alpha r^2}$ in a spherical potential.

We used parameters $\omega_z = 2.82843\omega_{\text{ho}}$, and a boson radius of $a/a_{\text{ho}} = 0.0043$ where $\omega_{\text{ho}} = \omega_x = \omega_y$ and $a_{\text{ho}} \equiv (\hbar/m\omega_{\text{ho}})^{\frac{1}{2}}$ is the characteristic length of the trap. There, we found upper-bound mean energies in units of $\hbar\omega_{\text{ho}}$ for 10, 50 and 100 particles respectively, $\langle E \rangle_{10} = 24.39852(7)$, $\langle E \rangle_{50} = 127.262(1)$, $\langle E \rangle_{100} = 266.6(1)$. We made the observation that the average per-particle energy increases as the number of states is increased and a spread in the density distribution, both matching the results presented by DuBois and Glyde [1].

1 Introduction

The recent demonstration of Bose-Einstein condensation for particles such as ^{87}Rb,^{23}Na and ^{7}Li in magnetic traps has lead to an increased interest in theoretical studies of the behaviour of this peculiar phenomenon. However, these studies have been performed for dilute system, where the atom size is much smaller than the trap-size and the inter-atom spacing. In this cases, one can safely apply the Gross–Pitaevskii equation. In newer experiments, this diluteness condition has not been met, asking for more advanced approaches.

In this article, we use the variational principle to get an estimate of the upper bound energy of the ground state of a system of N bosons in an elliptic harmonic oscillator potential. We model each particle as a hard sphere with a radius $a = 0.0043a_0$; where $a_{\text{ho}} \equiv (\hbar/m\omega_{\text{ho}})^{\frac{1}{2}}$ is the characteristic dimension of the trap. We assume the wave function to be of the shape of a harmonic-oscillator ground state wave function, with an additional factor to account for the interatomic interactions. To find the expectation value of the energy, we use the Metropolis-Hastings algorithm, both with and without importance sampling. As this produces a correlated data set, we transform the data set using the Blocking method which allows us to get a proper estimate of the error.

We will first describe the general problem, with the forms of the trial wave function and the Hamiltonian being presented in section 2.1. Then, we will describe the variational principle and its application in section 2.2. Computational methods for sampling from a probability distribution and finding optimal parameters - the Metropolis-Hastings algorithm and gradient descent - are described in sections 2.3 and 2.4, respectively. Then, we describe the motivation and procedure of the Blocking method in section 2.5. We give an overarching view of our implementation of the variational Monte

Carlo in C++ in section 2.6 and discuss some computational aspects like optimization and parallelization using OpenMP. We present and discuss our results in section 3, first analyzing, comparing and benchmarking the different algorithms on the non-interacting case, then applying them to the interacting case. Finally, we make a summary in section 4.

2 Theory & Methods

The theoretical models & concepts are largely based on the lecture notes written by M. Hjorth-Jensen [2], unless otherwise stated.

2.1 The model

2.1.1 The Hamiltonian

A general Hamiltonian with a two-particle interaction potential is given by

$$H = \sum_{i}^{N} \left(-\frac{\hbar^2}{2m} \nabla_i^2 + V_{\text{ext}}(\mathbf{r}_i) \right) + \sum_{i<j}^{N} V_{\text{int}}(\mathbf{r}_i, \mathbf{r}_j) \tag{2.1}$$

We use for the external potential

$$V_{\text{ext}}(\mathbf{r}) = \begin{cases} \frac{1}{2} m \omega_{\text{ho}}^2 r^2 & (S) \\ \frac{1}{2} m [\omega_{\text{ho}}^2 (x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \tag{2.2}$$

where S stands for spherical and E for elliptic; and the two-particle interaction potential is given by

$$V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{2.3}$$

where a is a constant. The interaction potential can be considered to be that of hard spheres, that do not interact when they are separated by a distance a, but that cannot be at the same spot simultaneously.

The Hamiltonian can be rewritten in units of $a_{\text{ho}} \equiv (\hbar / m\omega_{\text{ho}})^{\frac{1}{2}}$, where we write r in units of $r' = r/a_{\text{ho}}$ and express the energy in units of $\hbar\omega_{\text{ho}}$, which we occasionally write as $\hbar\omega$. Doing this, we get the new Hamiltonian when using the elliptical potential (dropping the apostrophes)

$$H = \sum_{i=1}^{N} \frac{1}{2} \left(-\nabla_i^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2 \right) + \sum_{i<j} V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) \tag{2.4}$$

where $\gamma = \omega_z / \omega_{\text{ho}}$; and a is now expressed in terms of $a' = a/a_{\text{ho}}$. See the derivation in the appendix. In case $\omega_{\text{ho}} = \omega_z$, we see that $x_i^2 + y_i^2 + \gamma^2 z_i^2$ reduces to r_i^2, which is the rewritten spherical external potential.

In this article, we will use two models. In the first model, we fully ignore the two-particle interaction potential ($a = 0$) and set $\gamma = 1$, turning the Hamiltonian into that of a simple N-particle Harmonic Oscillator in 3D. This will be called *Simple Harmonic Oscillator*. In the other model, we will set $a = 0.0043$ and $\gamma = 2.82843$ to account for the interaction.

2.1.2 The trial wave function

In this article, we will use the following trial wave function

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right]\left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|)\right], \tag{2.5}$$

with

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\} = \phi(\mathbf{r}_i). \tag{2.6}$$

and

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = f(r_{ij}) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \le a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{2.7}$$

where α and β are variational parameters, and we employed the shorthand notation $|\mathbf{r}_i - \mathbf{r}_j| = r_{ij}$. We will however set $\beta = 2.82843$ when the interaction potential is turned on, so that we only have one variational parameter α. For the non-interacting case, we set $a = 0, \beta = 1$, thus the interaction part of the wave function (2.7) is equal to 1.

Defining $u(r_{ij}) = \ln f(r_{ij})$, the gradient of the trial wave function with respect particle k is given by

$$\begin{aligned}\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\left\{\left(\sum_{j<m} u(r_{jm})\right)\right\} \\ + \left[\prod_i \phi(\mathbf{r}_i)\right]\exp\left\{\left(\sum_{j<m} u(r_{jm})\right)\right\}\sum_{l\neq k}\nabla_k u(r_{kl}).\end{aligned} \tag{2.8}$$

Similarly, we fined the expression for the second derivative as

$$\begin{aligned}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2 \Psi_T(\mathbf{r}) = \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2\frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right) \\ + \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \\ + \sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right).\end{aligned} \tag{2.9}$$

The derivation of the first and the second derivative can be found in the appendix.

2.1.3 Expectation value of the energy

The expectation value of the energy is given by the Rayleigh-Ritz ratio,

$$\langle E_L \rangle = \langle H \rangle = \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle} \tag{2.10}$$

without the repulsion term in one dimension, $\psi(x) = \exp(-\alpha x^2)$ for one particle in one dimension, this can be shown to be

$$\langle H \rangle_1 = \alpha + \frac{1 - 4\alpha^2}{8\alpha} \tag{2.11}$$

For N particles in M dimensions , we clearly have that $\langle H \rangle_{NM} = NM\langle H \rangle_1$.

2.1.4 One-particle densities

The one-particle density for a wavefuntion $\psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$ is defined as

$$n(\mathbf{r}) = \int |\psi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_N)|^2 d\mathbf{r}_2 \ldots d\mathbf{r}_N \tag{2.12}$$

that is, we integrate out all particles but one. Due to the indistinguishability of the bosonic particles, it does not matter which of these we choose. We can further define the radial distribution function

$$\rho(r) = \iint n(\mathbf{r}) r^2 \sin(\theta) d\theta d\phi \tag{2.13}$$

and the probability distribution in, for example, the x-direction

$$f_x(x) = \iint n(\mathbf{r}) dz dy. \tag{2.14}$$

One very useful property is that these can be interpreted as the probability of finding a particle at a distance r from origo (2.13) or at a position x from origo (2.14). This has advantages for the computational implementation.
For a fully spherical potential, with the parameter $\alpha = 0.5$ and no interaction, which also is the true ground-state wave function (and real), the trial wave function is separable, thus

$$n(\mathbf{r}) = C\psi^2(\mathbf{r}) = C\exp(-r^2) \tag{2.15}$$

for some normalization constant C, and we get for

$$\rho(r) = \iint n(\mathbf{r}) r^2 \sin(\theta) d\theta d\phi \propto r^2 \exp(-r^2) \tag{2.16}$$

which, upon proper normalization, will be the radial distribution function.

2.2 The Variational Method

The Variational Method is useful for determining an upper bound for the ground state energy E_0 of a system whose Hamiltonian H is known. The key observation that motivates the Variational Method is the fact that for any state vector $|\Psi\rangle$ in the Hilbert space, the expectation value of the energy $\langle H\rangle$ is larger than or equal to the ground state energy of the system,

$$\langle H\rangle = \langle\Psi|H|\Psi\rangle \geq E_0, \tag{2.17}$$

with equality if and only if the state $|\Psi\rangle$ is the ground state $|\varphi_0\rangle$ of the system. This follows from one of the postulates of quantum mechanics which says that the eigenvectors of a linear Hermitian operator corresponding to a physical observable form an orthonormal basis of the Hilbert space. This allows us to expand the state $|\Psi\rangle$ in the energy eigenbasis $\{|\varphi_n\rangle : H|\varphi_n\rangle = E_n|\varphi_n\rangle\}$,

$$|\Psi\rangle = \sum_n c_n |\varphi_n\rangle, \tag{2.18}$$

so that we can write the expectation value of the energy as

$$\langle H\rangle = \sum_m \sum_n c_m^* c_n \langle\varphi_m|H|\varphi_n\rangle = \sum_m \sum_n c_m^* c_n E_n \delta_{mn} = \sum_n |c_n|^2 E_n. \tag{2.19}$$

Since $E_n \geq E_0$ we have

$$\langle H \rangle = \sum_n |c_n|^2 E_n \geq \sum_n |c_n|^2 E_0 = E_0, \tag{2.20}$$

where we've used that the moduli of the coefficients $\{c_n\}$ sum up to 1. Equality holds only if $c_0 = 1$, i.e when $|\Psi\rangle = |\varphi_0\rangle$.

To evaluate the inner product in Eq. (2.17) we have to choose a basis in which to represent the state vector $|\Psi\rangle$ and the Hamiltonian H. We choose the configuration basis $\{|\mathbf{R}\rangle\}$, where each $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$ is a point in configuration space, to write Eq. (2.17) as

$$E_0 \leq \langle H \rangle = \int d\mathbf{R} \Psi^*(\mathbf{R}) \mathcal{H} \Psi(\mathbf{R}), \tag{2.21}$$

where $\Psi(\mathbf{R})$ is the wavefunction of the system and

$$\mathcal{H} = -\frac{\hbar^2}{2m} \sum_i^N \nabla_i^2 + V(\mathbf{R}) \tag{2.22}$$

is the configuration representation of the Hamiltonian. The Laplacian ∇_i^2 is taken with respect to the position of the i-th particle of the system and $V(\mathbf{R})$ is the system's total potential energy.

Although we can pick any trial function $\Psi_T(\mathbf{R})$ whatsoever, the Variational Method approach involves making an educated guess at a trial function that reflects the properties of the system at hand. That is, in addition to being continuous and vanishing at infinity, it should reflect, say, the symmetries of the system (or lack thereof). Or if the system resembles another system whose solution is known, one can use a trial function which resembles the ground state wavefunction of the solved system, perhaps with a tweak that reflects whatever the difference between the two systems may be.

In addition, one or more variational parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p) \in \mathbb{R}^p$ are included in the trial function to account for any uncertainty in the information about the shape of the true ground state wavefunction. For example, if you know that the ground state wavefunction of a system must resemble a Gaussian function, but not what the characteristic width must be, a variational parameter can be included in the trial function to account for the lack of information about the width.

Denoting the (generally not normalized) trial function by $\Psi_T(\mathbf{R}; \boldsymbol{\alpha})$, the upper bound to the ground state energy is given by

$$E_0 \leq \min_{\boldsymbol{\alpha}} \frac{\int d\mathbf{R} \Psi_T^*(\mathbf{R}; \boldsymbol{\alpha}) \mathcal{H} \Psi_T(\mathbf{R}; \boldsymbol{\alpha})}{\int d\mathbf{R} |\Psi_T(\mathbf{R}; \boldsymbol{\alpha})|^2}. \tag{2.23}$$

2.2.1 The Local Energy

As Eq. (2.23) involves two possibly high-dimensional integrals for a large amount of particles in the system, Monte Carlo integration should be employed. By inserting $1 = \Psi_T/\Psi_T$ between Ψ_T^* and $\mathcal{H}$ we can write the fraction in Eq. (2.23) as

$$\frac{\int d\mathbf{R} |\Psi_T(\mathbf{R}; \boldsymbol{\alpha})|^2 E_L(\mathbf{R}; \boldsymbol{\alpha})}{\int d\mathbf{R} |\Psi_T(\mathbf{R}; \boldsymbol{\alpha})|^2} \tag{2.24}$$

which is the expectation value of the quantity

$$E_L(\mathbf{R};\boldsymbol{\alpha}) \equiv \frac{1}{\Psi_T(\mathbf{R};\boldsymbol{\alpha})}\mathcal{H}\Psi_T(\mathbf{R};\boldsymbol{\alpha}) \tag{2.25}$$

known as the local energy. We thus approximate

$$\frac{\int d\mathbf{R}|\Psi_T(\mathbf{R};\boldsymbol{\alpha})|^2 E_L(\mathbf{R};\boldsymbol{\alpha})}{\int d\mathbf{R}|\Psi_T(\mathbf{R};\boldsymbol{\alpha})|^2} \approx \frac{1}{M}\sum_j^M E_L(\mathbf{R}_j;\boldsymbol{\alpha}) \tag{2.26}$$

where the configurations $\mathbf{R}_j$ are randomly drawn from the probability distribution $|\Psi_T(\mathbf{R};\boldsymbol{\alpha})|^2$ for a total of M times.

2.3 The Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm is a Markov chain Monte Carlo method used to obtain random samples from a probability distribution where direct sampling is difficult, such as N-dimensional integrals with large N.

For a discrete distribution, we define $\mathbf{P}_i^{(n)}$ to be the probability of finding the system in the state i at step n. We now define the transition probability $T_{i\to j}$ and the acceptance probability $A_{i\to j}$ for going to the newly sampled state j; while there is a probability of $(1 - A_{i\to j})$ of remaining in the same state. The aim is now to find properties of $T_{i\to j}$ and $A_{i\to j}$ such that, for any starting distribution, $\lim_{n\to\infty}\mathbf{P}_i^n = p_i$.

From the above description, we see that the probability of being in the state $\mathbf{P}_i^{(n)}$ is given by the probability of going from any state j to the state i, plus the probability of not going to a state j from a state i:

$$\mathbf{P}_i^{(n)} = \sum_j \left[\mathbf{P}_j^{(n-1)} T_{j\to i} A_{j\to i} + \mathbf{P}_i^{(n-1)} T_{i\to j}\left(1 - A_{i\to j}\right)\right] \tag{2.27}$$

Knowing that a new state j will be sampled when we are in a state i, $\sum_j T_{i\to j} = 1$, and we can rewrite the equation

$$\mathbf{P}_i^{(n)} = \mathbf{P}_i^{(n-1)} + \sum_j \left[\mathbf{P}_j^{(n-1)} T_{j\to i} A_{j\to i} - \mathbf{P}_i^{(n-1)} T_{i\to j} A_{i\to j}\right] \tag{2.28}$$

In the limit, $\lim_{n\to\infty}\mathbf{P}_i^n = p_i$, hence $\lim_{n\to\infty}\mathbf{P}_i^n - \mathbf{P}_i^{(n-1)} = 0$, and we get

$$\sum_j \left[p_j T_{j\to i} A_{j\to i} - p_i T_{i\to j} A_{i\to j}\right] = 0 \tag{2.29}$$

One possible way to enforce this equation is to set each term individually to zero. This gives

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i T_{i\to j}}{p_j T_{j\to i}} \tag{2.30}$$

A common choice is to set either $A_{j\to i}$ or $A_{i\to j}$ equal to one, which then gives

$$A_{j\to i} = \min\left(1, \frac{p_i T_{i\to j}}{p_j T_{j\to i}}\right) \tag{2.31}$$

The assumption for convergence was that $\lim_{n\to\infty}\mathbf{P}_i^n = p_i$. This can be shown to hold. We can write

$$\mathbf{P}_i^{(n)} = \sum_j M_{ij}\mathbf{P}_j^{(n-1)} \tag{2.32}$$

with

$$M_{ij} = \delta_{ij}\left[1 - \sum_k T_{i\to k}A_{i\to k}\right] + T_{j\to i}A_{j\to i} \tag{2.33}$$

Summing over i, we get that $\sum_i M_{ij} = 1$, and because (by construction) $\sum_k T_{i\to k} = 1$ and $A_{i\to k} \le 1$, M is a stochastic matrix. The largest eigenvalue of such a stochastic matrix is 1, and there is only one eigenvector with this eigenvalue. Hence, the Metropolis-Hastings algorithm guarantees to converge to the correct result.

The simplest approach, which we will call *Brute Force Metropolis* or just *Metropolis*, is to assume that $T_{j\to i} = T_{i\to j}$. Then the acceptance rate is simplified to

$$A_{j\to i} = \min\left(1, \frac{p_i}{p_j}\right) \tag{2.34}$$

When suggesting a move, one normally chooses a state j such that around 50% of moves are accepted. Here, we do this by a threshold value Δx such that a randomly chosen particle is moved randomly by a value $\Delta x \cdot \omega$ in each dimension, where ω is a uniformly distributed random number between -1 and 1.

2.3.1 Importance Sampling

In order to refine this method, we need an explicit expression for the transition probabilities $T_{i\to j}$.

We define the *Fokker Planck-equation*: In a time-dependent diffusion process with Probability density $P(x,t)$, the Fokker-Planck-equation for one particle reads

$$\frac{\partial P}{\partial t} = \sum_i D\frac{\partial}{\partial \mathbf{x}_i}\left(\frac{\partial}{\partial \mathbf{x}_i} - \mathbf{F}_i\right)P(\mathbf{x},t), \tag{2.35}$$

where $\mathbf{F}_i$ is the i-th component of the so-called drift term and D is the diffusion coefficient.

When dealing with a time-independent (hence stationary) probability density, the left hand side can be set to zero, and because of the linear independence of the dimensions, we get for all i

$$\frac{\partial^2 P}{\partial \mathbf{x}_i^2} = P\frac{\partial}{\partial \mathbf{x}_i}\mathbf{F}_i + \mathbf{F}_i\frac{\partial}{\partial \mathbf{x}_i}P. \tag{2.36}$$

We see now that we can express $\mathbf{F}$ in the form $\mathbf{F} = g(\mathbf{x})\frac{\partial P}{\partial \mathbf{x}}$. This gives

$$\frac{\partial^2 P}{\partial \mathbf{x}_i^2} = P\frac{\partial g}{\partial P}\left(\frac{\partial P}{\partial \mathbf{x}_i}\right)^2 + Pg\frac{\partial^2 P}{\partial \mathbf{x}_i^2} + g\left(\frac{\partial P}{\partial \mathbf{x}_i}\right)^2. \tag{2.37}$$

But we know that the left hand side needs to be zero, because we require the density to be stationary. This gives that $g = \frac{1}{P}$.

The solution of the Fokker-Planck equation gives that the transition probability is now given by the function

$$G(\mathbf{y},\mathbf{x},t) = \frac{1}{(4\pi D\Delta t)^{3N/2}}\exp\left\{-\frac{(\mathbf{y}-\mathbf{x}-D\Delta t F(\mathbf{x}))^2}{4D\Delta t}\right\} \tag{2.38}$$

where $\mathbf{y}$ is the position of the new state i and $\mathbf{x}$ is the position of the previous state j. Putting this into the equation for the Metropolis-Hastings algorithm, we hence get that

$$A_{j\to i} = \min\left(1, \frac{G(\mathbf{x},\mathbf{y},\Delta t)p_i}{G(\mathbf{y},\mathbf{x},\Delta t)p_j}\right) \tag{2.39}$$

The new positions in coordinate space are given as the solutions of the *Langevin equation*

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \tag{2.40}$$

where η is a random variable. Using Euler's method, we get a new position

$$\mathbf{y} = \mathbf{x} + DF(\mathbf{x})\Delta t + \xi\sqrt{\Delta t} \tag{2.41}$$

where ξ follows a Gaussian distribution and Δt is a time step, which serves as computational parameter. We will call this approach either *Metropolis-Hastings* or *Importance Sampling.*

2.3.2 Application to wave functions

Applying these methods to the wave function is straightforward. In quantum mechanics, the absolute square of a wave function divided by a normalization constant defines the probability distribution $P(x) = \frac{|\psi(\mathbf{x})|^2}{\langle\psi|\psi\rangle}$. The drift term, given by $\mathbf{F}(\mathbf{x}) = \nabla P(\mathbf{x})/P(\mathbf{x})$, can now easily be rewritten (assuming the wave function is real, hence $|\psi(\mathbf{x})|^2 = \psi(\mathbf{x})^2$) as

$$\mathbf{F}(\mathbf{x}) = \frac{\nabla P(\mathbf{x})}{P(\mathbf{x})} = \frac{2}{\psi(\mathbf{x})^2}(\nabla\psi(\mathbf{x})\psi(\mathbf{x})) = \frac{2}{\psi(\mathbf{x})}\nabla\psi(\mathbf{x}). \tag{2.42}$$

This is called the *Quantum Force.*
The diffusion coefficient is set to $D = 1/2$ when the problem is dimensionless.

For the Brute Force Metropolis approach, we get hence the acceptance rate (observe how the normalization constant cancels)

$$A_{j\to i} = \min\left(1, \frac{|\psi(\mathbf{x}_i)|^2}{|\psi(\mathbf{x}_j)|^2}\right), \tag{2.43}$$

while the importance sampling approach gives

$$A_{j\to i} = \min\left(1, \frac{G(\mathbf{x}_j,\mathbf{x}_i,\Delta t)|\psi(\mathbf{x}_i)|^2}{G(\mathbf{x}_i,\mathbf{x}_j,\Delta t)|\psi(\mathbf{x}_j)|^2}\right) \tag{2.44}$$

These expressions can be simplified further for efficient computation, see section 2.6.5.

2.4 Gradient Descent

To find the optimal variational parameters $\boldsymbol{\alpha}_{\text{opt}}$ that minimizes the expectation value of the local energy, we update them iteratively using a plain gradient descent approach,

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \eta\nabla_{\boldsymbol{\alpha}}\langle E_L\rangle, \tag{2.45}$$

where $\eta > 0$, the "learning rate", is a small number, $\nabla_{\boldsymbol{\alpha}}$ is the gradient with respect to the variational parameters and $\boldsymbol{\alpha}_0$ is simply a guess. The philosophy behind this approach is that, at each step k, we "walk" in $\boldsymbol{\alpha}$-space in the direction where $\langle E_L\rangle$ decreases the most, which is the opposite direction of the gradient. Since $\langle E_L \rangle$ is a convex function of the variational parameters we'll eventually get sufficiently close to a global minimum given a sufficiently small η. We stop iterating when $|\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k| < \delta$ where $0 < \delta \ll 1$.

Now we'll derive a compact expression for $\nabla_{\boldsymbol{\alpha}} \langle E_L \rangle$ that we will use to calculate it numerically. As we showed in Eq. (2.24), the expectation value of the local energy equals the expectation value of the Hamiltonian, so

$$\nabla_{\boldsymbol{\alpha}} \langle E_L \rangle = \nabla_{\boldsymbol{\alpha}} \langle H \rangle = \nabla_{\boldsymbol{\alpha}} \left(\frac{\int d\mathbf{R} \Psi^* \mathcal{H} \Psi}{\int d\mathbf{R} |\Psi|^2} \right). \tag{2.46}$$

Using the product and quotient rules we can write this as

$$\nabla_{\boldsymbol{\alpha}} \langle E_L \rangle = \frac{\left(\int d\mathbf{R} |\Psi|^2 \right) \nabla_{\boldsymbol{\alpha}} \left(\int d\mathbf{R} \Psi^* \mathcal{H} \Psi \right) - \left(\int d\mathbf{R} \Psi^* \mathcal{H} \Psi \right) \nabla_{\boldsymbol{\alpha}} \left(\int d\mathbf{R} |\Psi|^2 \right)}{\left(\int d\mathbf{R} |\Psi|^2 \right)^2} \tag{2.47}$$

$$= \frac{\int d\mathbf{R} \left[\nabla_{\boldsymbol{\alpha}} \left(\Psi^* \right) \mathcal{H} \Psi + \Psi^* \nabla_{\boldsymbol{\alpha}} (\mathcal{H} \Psi) \right]}{\int d\mathbf{R} |\Psi|^2} - \frac{\left(\int d\mathbf{R} \Psi^* \mathcal{H} \Psi \right) \int d\mathbf{R} \nabla_{\boldsymbol{\alpha}} |\Psi|^2}{\left(\int d\mathbf{R} |\Psi|^2 \right)^2} \tag{2.48}$$

$$= \frac{\int d\mathbf{R} |\Psi|^2 \frac{\nabla_{\boldsymbol{\alpha}} (\Psi^*) \mathcal{H} \Psi + \Psi^* \nabla_{\boldsymbol{\alpha}} (\mathcal{H} \Psi)}{|\Psi|^2}}{\int d\mathbf{R} |\Psi|^2} - \frac{\int d\mathbf{R} \Psi^* \mathcal{H} \Psi}{\int d\mathbf{R} |\Psi|^2} \frac{\int d\mathbf{R} |\Psi|^2 \frac{\nabla_{\boldsymbol{\alpha}} |\Psi|^2}{|\Psi|^2}}{\int d\mathbf{R} |\Psi|^2} \tag{2.49}$$

$$= \left\langle \frac{\nabla_{\boldsymbol{\alpha}} \left(\Psi^* \right) \mathcal{H} \Psi + \Psi^* \nabla_{\boldsymbol{\alpha}} (\mathcal{H} \Psi)}{|\Psi|^2} \right\rangle - \langle H \rangle \left\langle \frac{\nabla_{\boldsymbol{\alpha}} |\Psi|^2}{|\Psi|^2} \right\rangle \tag{2.50}$$

$$= \left\langle \frac{\nabla_{\boldsymbol{\alpha}} \left(\Psi^* \right) \mathcal{H} \Psi + \nabla_{\boldsymbol{\alpha}} \left(\Psi^* \right) \mathcal{H} \Psi}{|\Psi|^2} \right\rangle - \langle H \rangle \left\langle \frac{2 |\Psi| \nabla_{\boldsymbol{\alpha}} |\Psi|}{|\Psi|^2} \right\rangle \tag{2.51}$$

$$= \left\langle \frac{2 \nabla_{\boldsymbol{\alpha}} \Psi^*}{\Psi^*} E_L \right\rangle - \langle E_L \rangle \left\langle \frac{2 \nabla_{\boldsymbol{\alpha}} |\Psi|}{|\Psi|} \right\rangle = 2 \left[\left\langle \frac{\nabla_{\boldsymbol{\alpha}} \Psi^*}{\Psi^*} E_L \right\rangle - \langle E_L \rangle \left\langle \frac{\nabla_{\boldsymbol{\alpha}} |\Psi|}{|\Psi|} \right\rangle \right] \tag{2.52}$$

Depending on the trial function, in particular if the trial function is an exponential or a product of exponentials, the following equivalent expression might be more practical.

$$\nabla_{\boldsymbol{\alpha}} \langle E_L \rangle = 2 \left[\langle E_L \nabla_{\boldsymbol{\alpha}} \ln \Psi^* \rangle - \langle E_L \rangle \langle \nabla_{\boldsymbol{\alpha}} \ln |\Psi| \rangle \right] \tag{2.53}$$

The expectation values are calculated the same way we calculate the expectation value of the local energy. That is, by taking a sample mean over configurations obtained by the Metropolis-Hastings algorithm.

2.5 Error Estimation and Blocking

In our calculation of the ground state energy there are two kinds of contributions to the error. Firstly, we might not have a trial function that equals the exact ground state wavefunction for any values of the variational parameters α. Even if we did, we can only hope to get close to the optimal values α_{opt} (if we disregard the possibility that they all happen to be rational numbers exactly representable by a computer and reaching them by blind luck). These are systematic sources of errors which are hard to quantify. Secondly, if we had a perfect trial function and were able to vary α such that the trial function approaches the wavefunction $\varphi_0(\mathbf{R})$ of the true ground state $|\varphi_0\rangle$, then the local energy approaches a constant,

$$E_L(\mathbf{R};\alpha) = \frac{1}{\Psi_T(\mathbf{R};\alpha)}\mathcal{H}\Psi_T(\mathbf{R};\alpha) \to \frac{1}{\varphi_0(\mathbf{R})}\mathcal{H}\varphi_0(\mathbf{R}) = \frac{1}{\varphi_0(\mathbf{R})}E_0\varphi_0(\mathbf{R}) = E_0, \tag{2.54}$$

which happens to be the ground state energy itself. This means that the variance of the local energy approaches zero, but in general it is non-zero. Since we're approximating the expectation value of the local energy by a sample mean over randomly drawn configurations, we get a statistical contribution to the error. It is this contribution that we attempt to quantify in this section. By "the error" we mean the standard deviation of the sample mean of the local energy, i.e

$$\sigma(\overline{E}_L) = \sqrt{\mathrm{Var}(\overline{E}_L)}, \tag{2.55}$$

where

$$\overline{E}_L = \frac{1}{M}\sum_i^M E_L(\mathbf{R}_i). \tag{2.56}$$

To make the notation more general, we consider m data sets $\{x_{\alpha,k}\}$ for $\alpha = 1,\ldots,m$ each containing n samples $x_{\alpha,k}$ for $k = 1,\ldots,n$. The sample mean and sample variance of one set is

$$\mu_\alpha = \frac{1}{n}\sum_k^n x_{\alpha,k}, \;\; \mathrm{Var}(x) = \frac{1}{n}\sum_k^n (x_{\alpha,k} - \mu_\alpha)^2 \tag{2.57}$$

respectively. We define the total mean

$$\mu_m = \frac{1}{m}\sum_\alpha^m \mu_\alpha = \frac{1}{mn}\sum_\alpha^m\sum_k^n x_{\alpha,k}, \tag{2.58}$$

and the total variance

$$\mathrm{Var}(\mu_\alpha) = \frac{1}{m}\sum_\alpha^m (\mu_\alpha - \mu_m)^2. \tag{2.59}$$

In our case, μ_α corresponds to $\overline{E}_L$ with $m = 1$ and $\mathrm{Var}(\mu_\alpha)$ is the square of the error which is what we're trying to estimate. In the appendix, we show that the total variance can be written as

$$\mathrm{Var}(\mu_\alpha) = \frac{\sigma^2}{n} + \frac{2}{mn^2}\sum_\alpha^m\sum_{k<l}^n (x_{\alpha,k} - \mu_m)(x_{\alpha,l} - \mu_m), \tag{2.60}$$

where

$$\sigma^2 \equiv \frac{1}{mn^2}\sum_\alpha^m\sum_k^n (x_{\alpha,k} - \mu_m)^2. \tag{2.61}$$

is the sample variance over all the data sets. The reason that we cannot use this quantity directly to estimate the error is that the total variance includes an extra term which accounts for correlations between the data points, of which we have plenty. So σ^2 is a too optimistic estimate in our case. The problem with Eq. (2.60) from a computational perspective is that, when the data set is correlated, it involves a double loop over l and $k < l$, which we'd rather avoid computing when the data set is large. One way to avoid this is to perform so-called Blocking transformations on the data set, which, when performed repeatedly, converges to a data set without correlations. This allows us to use the usual sample variance with no double loops.

Introducing the quantity

$$f_d \equiv \frac{1}{mn} \sum_{\alpha}^{m} \sum_{k}^{n-1} (x_{\alpha,k} - \mu_m)(x_{\alpha,k+d} - \mu_m) \tag{2.62}$$

we can write the total variance as

$$\text{Var}(\mu_\alpha) = \frac{\sigma^2}{n} + \frac{2}{n} \sum_{d=1}^{n-1} f_d. \tag{2.63}$$

In terms of the autocorrelation function $\kappa_d \equiv f_d/\sigma^2$ the total variance is

$$\text{Var}(\mu_\alpha) = \frac{\sigma^2}{n} \left(1 + 2 \sum_{d=1}^{n-1} \kappa_d \right) \tag{2.64}$$

and in terms of the so-called autocorrelation time $\tau \equiv 1 + 2 \sum_{d=1}^{n-1} \kappa_d$ we have

$$\text{Var}(\mu_\alpha) = \frac{\sigma^2}{n} \tau \tag{2.65}$$

which reduces to the well-known $\sigma(\mu_\alpha) = \sigma/\sqrt{n}$ for $\tau = 1$. The interpretation of τ is that it is the "distance" between uncorrelated data points. If $\tau = 1$ then subsequent data points are uncorrelated. Now, a blocking transformation is performed by simply replacing two subsequent data points in a data set by their average, i.e

$$(x_1, x_2, \ldots, x_{n-1}, x_n) \rightarrow \left(\frac{x_1 + x_2}{2}, \ldots, \frac{x_{n-1} + x_n}{2} \right), \tag{2.66}$$

where n is an integer power of 2. This is done repeatedly until the size of the data set is $n' \leq n/\tau$ so that subsequent data points are uncorrelated. It can be shown that the sample mean and the total variance is conserved under blocking transformations, but the correlation contribution to the total variance is reduced, so that the sample variance becomes a better and better estimate of the total variance [3].

2.6 Computational Implementation and Computational aspects

All code that was used in this article can be found here. The Monte Carlo sampling, the Metropolis- and the Metropolis-Hastings-algorithm, and gradient descent were implemented using C++. The code structure is based on a framework provided by Morten Ledum[1], whom we would like to thank for his work. We used openMP to parallelize our code. Plotting, Blocking and further data analysis is implemented in Python.

[1]it can be found here

2.6.1 Design

The code is fully object oriented, and rather modular. Both wave functions are represented by different classes, and there is an additional class for the simple harmonic oscillator wave function with numerical differentiation. Similarly, the two Hamiltonians are represented as different classes. There is several Sampler classes that do the sampling from the wave functions, and two coordinating system classes, which one to use depends on whether one wants to use Gradient Descent or Sampling; or whether Brute Force Monte Carlo or Importance Sampling should be used. In addition, there are some helper classes, used for setting up the system, creating random numbers etc.

2.6.2 Particle placing & equilibration

To begin with, the N particles are randomly placed in a box of length $1 \times 1 \times 1$ with the middle in origo. For the interacting case, it is important that the initialization state is legal, that is, particles are placed in such a way that they are at least a distance of a/a_{ho} away from all other particles. We do this by expanding the box if too many forbidden placements are suggested. In order to have the system equilbrate, there is a parameter upon initialization how many steps to discard.

2.6.3 One-particle density

We find the one-particle density directly from the Metropolis-Hastings algorithm, as the algorithms directly imitates the true distribution. Hence, to find for example $f_x(x)$ (2.14), we simply store, for each iteration, the position x of particle 1. The (normalized) histogram will then correspond to the probability distribution $f_x(x)$. Equally, by storing the distance from origo r, we find $\rho(x)$. Because of the indistinguishability of bosons, we can store the position of all particles, and upon normalization, acquire the same distribution. In order to make this less memory-demanding, we do not store the positions, but the number of times a particle is found in an interval $[r_i, r_{i+1}]$, where we sat $r_{i+1} - r_i = 0.05$.

2.6.4 Parallelization

Parallelization is implemented in such a way that we run n_p independent runs of the same problem (the same wave function with the same variational parameter α, the same number of particles etc.), but with different initial positions and a different seed for the random number generator. The final expectation values are then the average value of each individual run's expectation values. For Blocking, we simply write each local energy to file for each run, and then treat it as data from the same experiment.
For the gradient descent, we do the same, and the α we consider ideal, is then the average of the converged αs of each run (which should lie very close to each other).

2.6.5 Optimizations

We implemented several things to make the code run faster. Some examples are that we never evaluate Green's function explicitly, as only the quotient is needed (2.44), such that only one exponential needs to be calculated. That is, we can write

$$G(\mathbf{x}, \mathbf{y}, \Delta t)/G(\mathbf{y}, \mathbf{x}, \Delta t) = \frac{(4\pi D\Delta t)^{3N/2}}{(4\pi D\Delta t)^{3N/2}} \prod_i^N G(x_i, y_i, \Delta t)/G(y_i, x_i, \Delta t) \tag{2.67}$$

which is simply the exponent over a sum because $e^{a+b} = e^a e^b$, and the prefactor cancels.
We also implemented a trick for the quotient of the probability distributions, where parts of the wave function are not changed when only one particle is changed, hence these parts don't need to be evaluated, as demonstrated for the wave function (2.5) when moving particle 1 from $\mathbf{r}_1$ to $\mathbf{r}_1'$ (but

it applies to any index):

$$\begin{aligned}\frac{\Psi_T(\mathbf{r})}{\Psi_T(\mathbf{r}')} &= \frac{\Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta)}{\Psi_T(\mathbf{r}'_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta)} \\ &= \frac{\left[g(\alpha,\beta,\mathbf{r}_1)\prod_{i\neq 1} g(\alpha,\beta,\mathbf{r}_i)\right]\left[\prod_{j<k;j,k\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}_k|)\prod_{j\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}_1|)\right]}{\left[g(\alpha,\beta,\mathbf{r}'_1)\prod_{i\neq 1} g(\alpha,\beta,\mathbf{r}_i)\right]\left[\prod_{j<k;j,k\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}_k|)\prod_{j\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}'_1|)\right]} \\ &= \frac{g(\alpha,\beta,\mathbf{r}_1)\prod_{j\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}_1|)}{g(\alpha,\beta,\mathbf{r}'_1)\prod_{j\neq 1} f(a,|\mathbf{r}_j-\mathbf{r}'_1|)}\end{aligned} \tag{2.68}$$

Clearly, this also applies to the square, which is used in the Metropolis algorithm (2.43) & (2.44).

We avoid calculating the double sum in (2.9) by realizing that the double sum can be split into the product of two identical sums, as is done in the appendix. Finally, we apply the $-O3$ flag for compilation to speed up our program.

2.6.6 Testing

We implemented a few test functions to see that everything is implemented correctly. We tested that the expected energy $\langle E_L \rangle$ parameter $\alpha = 1.0$ with both the Brute Force Metropolis algorithm and Importance sampling is very close to the analytical expectation value for the simple Harmonic Oscillator. We also tested that for the numerical evaluation of the derivative. We also tested that the gradient descent algorithm goes to $\alpha = 0.5$. For the interacting wave function, we tested that, when setting the radius a to a value very close to zero, that the expectation value at $\alpha = 1.0$ becomes that of a simple Harmonic Oscillator. All tests were performed using 10 particles.

2.6.7 Data analysis and plotting

Most of data analysis and plotting is performed in Python. There are python scripts that run the C++ programs with the desired input parameters, the C++ files write to file (all data is either appended to the same file, or data is read from the terminal), and the Python scripts then read from that file to either create new files, write out to the terminal, or create plots. Especially the Blocking algorithm is implemented as a Python script.

3 Results & Discussion

3.1 The non-interacting case

3.1.1 Brute-force Monte Carlo and the Harmonic Oscillator

We applied a Simple Brute-Force Monte Carlo to the Harmonic Oscillator for 1, 10, 100 and 500 particles. From Quantum Mechanics, as stated previously, we know that the analytical energy is given as $E = \frac{1}{2}\hbar\omega NM$, where N is the number of particles and M the dimension. We know that the correct ground state is given by the first trial wave function when $\alpha = 0.5$, so we expect that this value gives correct energies. The general trend curve how $\langle E_L \rangle(\alpha)$ is shown in figure 1.

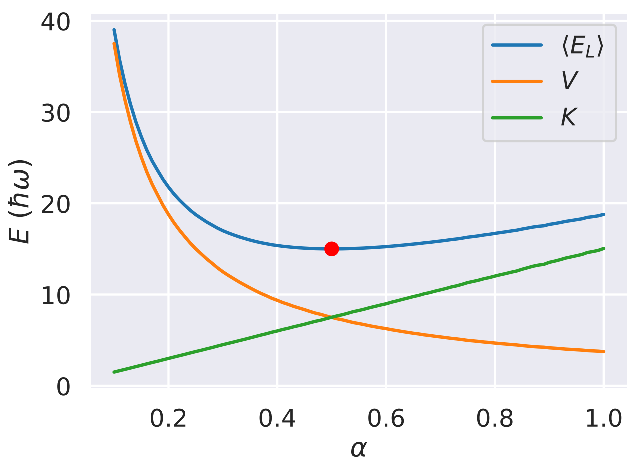

Figure 1: *Expected values $\langle E_L \rangle, \langle V \rangle, \langle K \rangle$ in units of $\hbar\omega$ as function of α for 10 particles in three dimensions with $N = 10^6$ Monte Carlo steps. The minimal energy is marked in red and lies at $\alpha = 0.5$.*

We see that, as expected, we get an energy of $\langle E_L \rangle = 15\hbar\omega$ at $\alpha = 0.5$. This is also the point where we have that $\langle V \rangle = \langle K \rangle$, as expected for an harmonic oscillator. Different energies as function of the number of Monte Carlo Cycles and the number of particles for 1, 2 and 3 dimensions are shown in table 5 in the appendix. It shows that the computational minimum is always attained at $\alpha = 0.5$ and that the function does not seem to have any local minimia around $\alpha = 0.5$, in the sense that the expectation value of the local energy keeps raising as we go further away from $\alpha = 0.5$, indicating qualitatively correct results, even though not enough MC cycles were performed to obtain quantitatively correct results.

We did the same analysis with importance sampling. This is shown in table 6, with the same implications.

3.1.2 Comparing numerical differentiation to analytic differentiation

We compared the time usage between numerical differentiation and analytical differentiation for the calculation of the local energy for the N-particle Harmonic Oscillator in three dimensions using Brute-Force Monte Carlo. We also compared the time consumption as function of the number of particles. Here, we used the -O3 optimization flag. This is shown in figure 2.

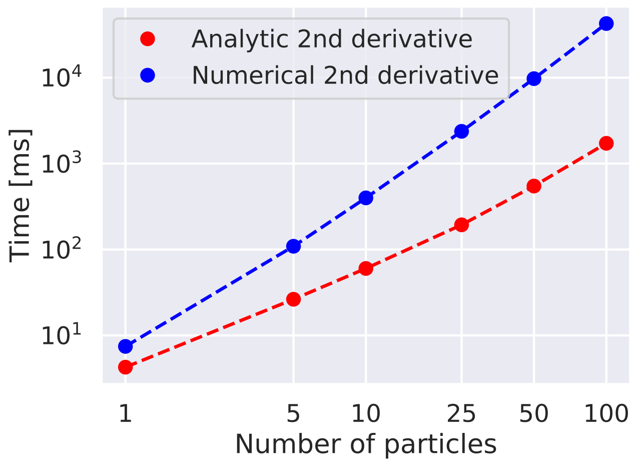

Figure 2: *Time consumption as function of number of particles for a 3D harmonic oscillator using Brute-Force Monte Carlo. The number of Monte Carlo cycles was chosen to be $MC = 10^4 \times (\text{number of particles})$. -O3 optimization flags where used. Time averaged over 11 runs with different values for the variational parameter $\alpha \in [0.3, 0.7]$.*

As we can see, using the analytical expression has two advantages, time-wise. First of all, it is faster in general, and beats the numerical expression clearly. We also observe that the difference grows as the number of particles increase going from a difference of ≈ 4 times faster for 5 particles to ≈ 25 times faster for 100 particles, showing that having an analytical expression is a game changer for many particles.

In terms of accuracy, we observe that numerical and analytical differentiation work in practice equally well, even though there are some numerical differences. For 100 particles and $h = 10^{-5}$, we see that both analytical and numerical derivation give the exact ground state energy at the correct value $\alpha = 0.5$, and both of them give the right trend (energy increases as one goes away from the ideal value), which is a minimum requirement for both of them to be usable. We see that the values are identical up to 7 decimals, which shows that the numerical derivation is not inferior in terms of accuracy with the chosen parameters, as seen in table 1, so the big caveat is time usage.

Table 1: *$\langle E_L \rangle$ with for the numerical and the analytical expression of the second derivative as function of the variational parameter α for N=100 particles in three dimensions. The energy is expressed as multiple of $\hbar\omega$. The number of Monte Carlo cycles was chosen to be 10^6 (10^4 per particle on average), disregarding 10^5 runs in addition, and the same seed was used for all runs to assure comparability. We have $h = 10^{-5}$ for numerical derivation.*

α	$\langle E_L \rangle_{numerical}$	$\langle E_L \rangle_{analytical}$
0.300	170.16320	170.16319
0.340	161.23484	161.23486
0.380	155.75057	155.75058
0.420	152.31722	152.31722
0.460	150.55610	150.55610
0.500	149.99999	150.00000
0.540	150.44809	150.44808
0.580	151.68609	151.68607
0.620	153.50851	153.50849
0.660	155.88232	155.88231
0.700	158.57620	158.57620

3.1.3 Comparing Brute Force Metropolis to Importance Sampling

We compared the convergence of the Brute Force-algorithm to the Importance Sampling algorithm for $\alpha = 1$, that is, a value where sampling actually plays a role.

The expectation value of the energy is given by equation (2.11) which is the result we expect to get from Brute Force and Importance sampling, too. For $N = 10$ particles in $M = 3$ dimensions and $\alpha = 1$, we get that $\langle H \rangle_{10,3} = 18.75$.

In figure 3, we plot the relative error compared to the analytical value 18.75 as function of the number of Metropolis steps, for different values of Δx for Brute-Force Monte Carlo, and Δt for Importance Sampling.

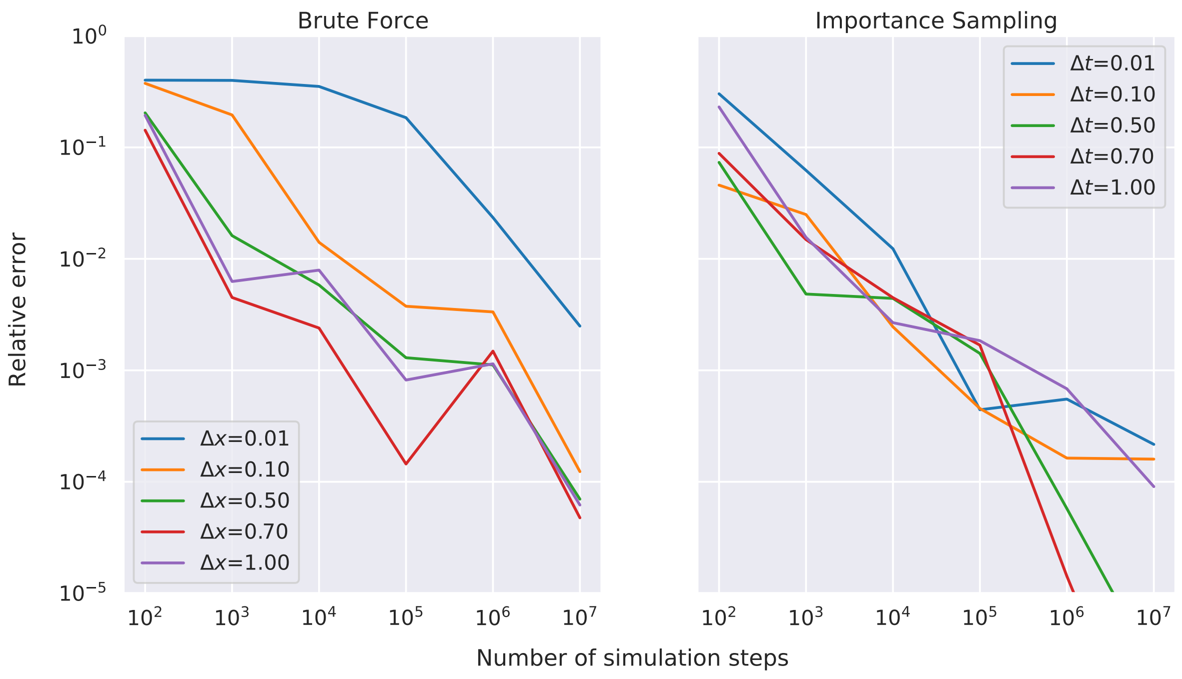

Figure 3: *Relative error of the expected value $\langle E_L \rangle$ in units of $\hbar\omega$ at $\alpha = 1$ as function of the number of Monte Carlo cycles for $N = 10$ particles in 3 dimensions. In addition, 10% of MC cycles were discarded. We took the average value over 10 different seeds in order to attempt to average out small fluctuations, which, however, cannot be eliminated fully.*

We see that Importance Sampling approaches give better values than Brute-Force Monte Carlo, which is expected, as it is a more sophisticated algorithm. The general trend curve seems to go down faster, showing that Importance Sampling may give faster convergence. We also see that the Importance sampling algorithm is much more robust when it comes to choosing Δt, the values lie relatively close (especially in absolute value), while there is more spread in the Brute Force approach - this justifies that many values for Δt can be used in Importance Sampling, while Brute Force Monte Carlo is "stricter" in the sense that it is customary to choose Δx such that the acceptance rate is around 50%. Indeed, the Brute Force plot shows that $\Delta x = 0.70$ gives the best results, where the acceptance rate was around 50%, together with $\Delta x = 1.0$ and $\Delta x = 0.5$, which has a somewhat lower/higher acceptance rate. The fluctuations that show up cannot be avoided, as there is an intrinsic insecurity in stochastic algorithms, but the general trend shows that Importance Sampling is overall better than Brute Force Monte Carlo and that the relative error goes down the more Monte Carlo steps are taken. However, the relative error is erratic, and that explains why it is necessary to have good error estimates.

3.1.4 Blocking

As a test, we wrote the local energy at each MC step to file using Importance sampling for a system consisting of $N = 10$ particles in three dimensions, setting again $\alpha = 1$. Figure 4 shows the expected value $\langle E_L \rangle$ as function of the Monte Carlo steps with the corresponding blocking error, as well as the naive error $\sigma_E = \frac{\sqrt{\langle E_L^2 \rangle - \langle E_L \rangle^2}}{\sqrt{MC}}$. Table 2 contains the same information.

Table 2: *Expectation value $\langle E_L \rangle$ and $\sigma_{Blocking}$ and the naive error estimate σ_E as function of the number of MC cycles for $N = 10$ particles in 3 dimensions with $\alpha = 1.0$ using importance sampling and $\Delta t = 0.1$. Observe that the analytical solution is $\langle E_L \rangle = 18.75$.*

MC cycles	$\langle E_L \rangle$	$\sigma_{Blocking}$	σ_E
2^2	21	1	1
2^3	19.8	0.8	0.8
2^4	19.4	0.4	0.4
2^5	19.4	0.3	0.3
2^6	20.3	0.4	0.2
2^7	20.1	0.3	0.1
2^8	19.6	0.6	0.2
2^9	19.9	0.4	0.1
2^{10}	19.87	0.4	0.08
2^{11}	19.12	0.4	0.06
2^{12}	18.87	0.2	0.04
2^{13}	18.70	0.2	0.03
2^{14}	18.75	0.1	0.02
2^{15}	18.65	0.1	0.02
2^{16}	18.71	0.07	0.01
2^{17}	18.733	0.05	0.008
2^{18}	18.700	0.04	0.006
2^{19}	18.731	0.03	0.004
2^{20}	18.723	0.02	0.003
2^{21}	18.727	0.01	0.002
2^{22}	18.737	0.01	0.001
2^{23}	18.742	0.007	0.001
2^{24}	18.7455	0.005	0.0007

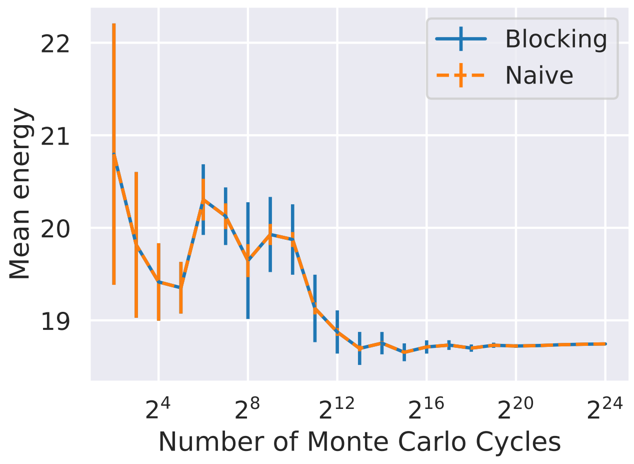

Figure 4: *Expectation value $\langle E_L \rangle$ and $\sigma_{Blocking}$ and the naive error estimate σ_E as function of the number of MC cycles for $N = 10$ particles in 3 dimensions with $\alpha = 1.0$ using importance sampling and $\Delta t = 0.1$. Observe that the analytical solution is $\langle E_L \rangle = 18.75$.*

We see that the blocking estimate is reasonable for many cycles, as can be seen from the fact that, for 2^{12} MC cycles and upover, the true mean error lies in the range $\langle E_L \rangle \pm 2\sigma_{Blocking}$. The naive estimate is however not reasonable, it lies 7 standard deviations σ_E away for 2^{24} MC cycles. Observe however that also Blocking seems to underestimate the standard error for very few MC cycles (for example with 2^7 MC cycles), showing that the method is not exact.

We also repeated the same experiment as in figure 4 / table 2 with a value for $\Delta t = 0.01$. This is shown in table 7 in the appendix. There, we got a much larger Blocking standard deviation. We do not know for sure why this happens, but we suppose that the very small step length leads to more correlation between runs, hence a longer decoherrence time. We also see that the average value for different seeds fluctuates more. The error estimate $\sigma_E = 0.0007$ using the naive approach is however the same. A similar experiment was performed using $\Delta t = 1$, which gives similar, but somewhat inferior results to $\Delta t = 0.01$.

We also did the same experiment for $N = 100$ particles. There, we found that $\Delta t = 1$ works better than $\Delta t = 0.1$, in the sense that the blocking estimate $\sigma_{Blocking}$ is lower. We suppose that this means that it is better to move particles farther due to the much larger parameter space, as the time for the next move is much higher.

Finally, we repeated the same experiment with brute force Monte Carlo and step length $\Delta x = 0.7$ where the acceptance rate is around 50%, and found again $\sigma_E = 0.0007$, with a blocking error of $\sigma_{Blocking} = 0.007$ for 2^{24} MC cycles, which is somewhat larger than the Importance Sampling value. This confirms that the naive estimate is not sufficient in a correlated system such as this one; where the next position is not independent of the previous, and only one particle is moved at a time. For the complex wave function, an additional complexity stems from the fact that the wave function itself is not separable and the particles are not independent.

3.1.5 One-particle densities

For $N = 100$ particles, we compare the analytical one-particle radial distribution function to the computational result with the ideal value $\alpha = 0.5$, the same is done for $f_x(x)$. This is shown in figure 5.

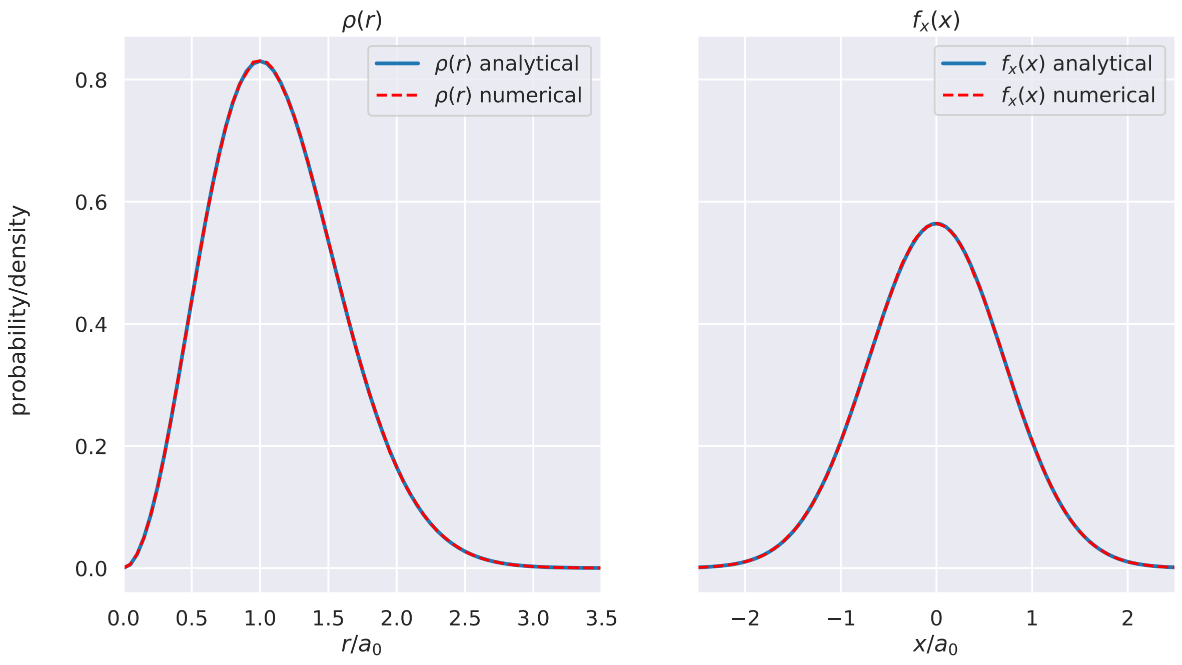

Figure 5: *Radial distribution function $\rho(r)$ and probability distribution over x $f_x(x)$ for 100 particles, both the analytical and the numerical result. The data was created using Importance sampling and 10^7 MC cycles; with the ideal parameter α.*

As expected, the two curves overlap almost perfectly, showing the correct implementation and interpretation of the functions $f_x(x)$ and $\rho(r)$.

3.2 The Wave function with the interaction term

We repeated some of the steps for the wave function with the interaction term. In figure 6, the energy as function of α is plotted for 10 particles.

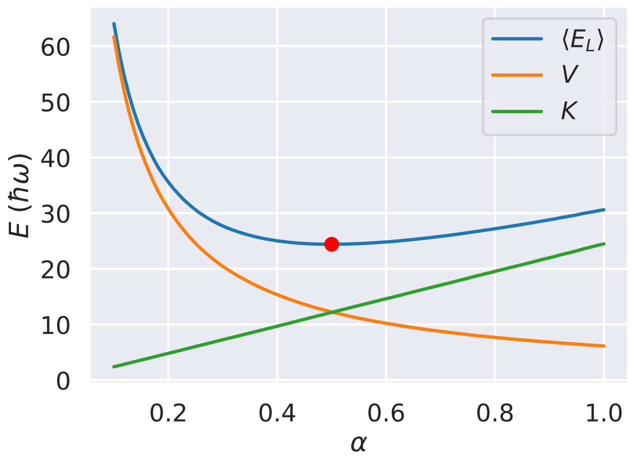

Figure 6: *Expected values $\langle E_L \rangle, \langle V \rangle, \langle K \rangle$ in units of $\hbar\omega$ as function of α for 10 particles in three dimensions with $N = 10^5$ Monte Carlo steps. The minimal energy is marked in red and lies at $\alpha = 0.5$, but this is only a rough estimate because of the coarse step length.*

We see that the energy $\langle E_L \rangle$ as function of α seems to have only one minimum close to $\alpha = 0.5$, which serves as starting guess for the gradient descent algorithm.

3.2.1 The parameter α as function of the number of particles

We used Gradient Descent to find the ideal variational value of α using importance sampling for different values of N. We observed that the choice of Δt has an impact on the value for α. We also observed that the final result is quite parameter-dependent, so we fixed the learning rate to be $0.1/N$ where N is the number of particles, and used as tolerance $tol = 10^{-5}$ for the gradient algorithm. It also depends on the number of warm up-steps and the number of MC-cycles. The results are shown in table 3. We also checked that very similar results where achieved when starting with a a different $\alpha = 0.49$ to check the general validity of our results.

Table 3: *Ideal value for the variational parameter* $\boldsymbol{\alpha}$ *as function of the number particles* N*, the step length* Δt *for Importance sampling, and the number of Monte Carlo samples. The initial estimate was sat to be* $\alpha = 0.5$*. As learning rate, we chose* $\eta = 0.1/N$ *and for the tolerance* $tol = 10^{-5}$*, with maximum 20 iterations. Each value is the average value of 4 runs that took place in parallel with the same parameters, but a different random number generator. The number of MC cycles is* 10^4 *and* 10^5 *for each Gradient-Descent iteration, respectively, and and we discarded* $10^4/3$ *and* $10^5/3$ *runs, respectively. Numbers are reported in such a way that only the last digit for different numbers of* Δt *differs.*

Number of particles	N=3	N=5	N=10	N=25	N=50	N=100
MC=10^4 $\Delta t = 0.01$	0.4993	0.4986	0.4979	0.493	0.491	0.483
MC=10^4 $\Delta t = 0.1$	0.4994	0.4988	0.4975	0.494	0.489	0.484
MC=10^4 $\Delta t = 1$	0.4994	0.4989	0.4977	0.494	0.490	0.475
MC=10^5 $\Delta t = 0.01$	0.49940	0.49881	0.49751	0.4940	0.488	0.486
MC=10^5 $\Delta t = 0.1$	0.49940	0.49882	0.49745	0.4939	0.489	0.483
MC=10^5 $\Delta t = 1$	0.49941	0.49885	0.49746	0.4938	0.489	0.482

We observe a couple of interesting trends. First of all, as the number of particles increases, the ideal value of α decreases, no matter the choice of Δt. This is not observed in the noninteracting case. This makes the wave function spread out more as N increases, which makes sense, as there is a repulsive force between particles.

Similarly to the noninteracting case, we observe differences based on the choice of Δt. For larger systems, the ideal values differ more and more from each other, but are still very close to one another. Finally, we see that the values achieved with either 10^4 MC cycles or 10^5 MC cycles don't differ by a lot for small systems, but they do differ a lot for larger systems, indicating that a large number of runs is necessary to estimate the gradient of α properly.

The numbers are not ideal, and we are only approximating the true ideal parameter. However, assuming that the shape of the energy resembles that of figure 6 for any N, we see that minimal changes in α don't make a large difference for $\langle E_L \rangle$.

3.2.2 The variational energy

Using the values of α found in the preceeding discussion, we performed a large-scale Monte Carlo simulation using Importance sampling to find the mean value $\langle E_L \rangle$. In table 4, we show $\langle E_L \rangle$ as function of the number of particles, as well as the Blocking standard error.

Table 4: *Expected energy $\langle E_L \rangle$ for different values of the number of particles N and the values of α used to obtain them, as well as the energy per particle $\langle E_L \rangle / N$ and the blocking standard deviation $\sigma_{Blocking}$. The energy of an Elliptic Harmonic Oscillator E_{EO} without repulsion with $\alpha = 0.5$, $\beta = 2.82843$, is presented too. We performed in total 2^{25} MC cycles in parallel on 4 threads (2^{23} per thread) using importance sampling with $\Delta t = 0.1$, additionally discarding $1000N$ samples for each run. Every sample was written to file for the Blocking analysis.*

Number of particles	$N = 3$	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
α	0.4994	0.4988	0.4975	0.4939	0.489	0.483
$\langle E_L \rangle$	7.259929	12.12857	24.39852	62.0215	127.262	266.6
$\sigma_{Blocking}$	0.000006	0.00002	0.00007	0.0004	0.001	0.1
$\langle E_L \rangle / N$	2.419976	2.425714	2.439852	2.48086	2.54524	2.660
E_{EO}	7.242645	12.071075	24.14215	60.355375	120.71075	241.4215

We observe a bunch of interesting trends. First of all, given that we rely on the Blocking result, we can obtain quite secure confidence intervals for $\langle E_L \rangle$, even though we lose accuracy as the number of particles go up (scaled by more than just the number of particles). This is partially because the same number of MC cycles was used in all runs, hence as N grows, each particle is moved less often on average, giving more deviations in the energy. We also think that we did not manage to get close enough to the ideal parameter α for N=100, which would explain the larger deviations. This is justified by the relatively large variations in α as function of Δt shown i table 3. It might however also be because the approximation of our wave function for the given potential becomes gradually worse for many particles.

We also observe how the energy per particle $\langle E_L \rangle / N$ grows as the number of particles is increased. This seems logical - from the single harmonic oscillator, we know that the particle is most likely to be found close to centre. However, due to the repulsive interaction, the particles are spread out more into an area with higher potential energy. This leads to a higher energy. This increase of per-particle energy was also observed in [1].

Finally, we observe that the numbers here are very similar to the ones in table 9, where experiments were run with slightly different α (based on table 8, which is actually based on a bug). This shows that a slight change in α only gives a slight change in the expectation value of the energy.

3.2.3 Change in the one-particle density

For $N = 100$ particles, we compare the analytical one-particle radial distribution function $\rho(r)$ for $a = 0$ and $a = 0.0043$ for the elliptic potential to the one for a spherical potential. The same is done with $f_x(x)$. This follows the procedure described in section 2.6.3.

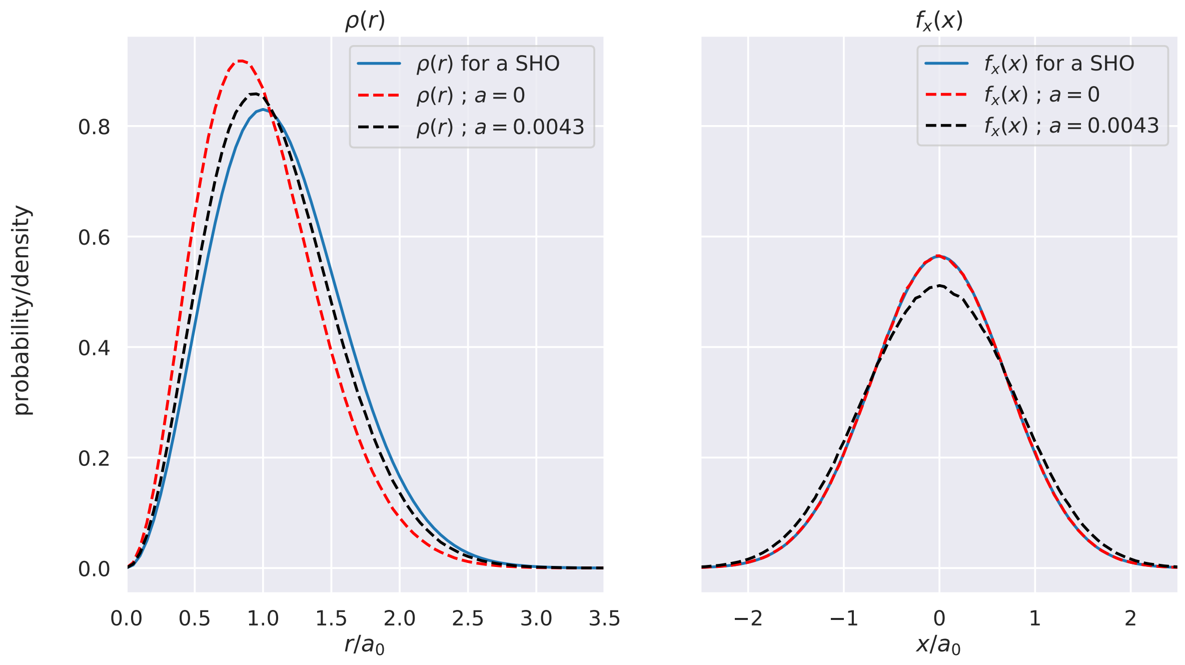

Figure 7: *Radial distribution function $\rho(r)$ and probability distribution over x $f_x(x)$ for 100 particles for a spherical harmonic oscillator (SHO), for an elliptical without interaction (a=0) and with interaction (a=0.0043). The data was created using Importance sampling and 10^7 MC cycles; with the ideal parameter α (0.5 for $a = 0$; 0.483 for $a = 0.0043$).*

The results are in agreement with the previous discussion. For $f_x(x)$, we see that, when $a = 0$, the distribution is identical to that of a spherical harmonic oscillator. This is not surprising, as the function is separable when $a = 0$, and we expect the distribution function to be $\propto \exp\{-x^2\}$. $\rho(r)$ for $a = 0$ is not identical to the spherical harmonic oscillator - this is because $\beta > 1$, making it less probable for the particle to be found far from origo in the z-direction, something that is also reflected in $\rho(r)$. With $a = 0.0043$, we observe that the interaction leads to a broadening of the distribution functions, which makes sense, as the particles being close is less likely in the wave function, and hence they spread out more. Similar observations on the orbitals of the condensate spreading were made in [1].

4 Conclusion

We've made an object-oriented and parallelized VMC implementation in C++ for estimating an upper bound of the ground state energy for systems of bosons in a harmonic oscillator trap. We tested the results of our implementation against the theoretical predictions for a non-interacting boson-gas, using both brute-force Metropolis and importance sampling. In both cases we found agreement with the theoretical prediction $E_0 = \hbar\omega NM/2$ at $\alpha = 0.5$. We also observed that importance had a tendency to be less dependent on the correct choice step length Δt and converged generally faster. Furthermore, as the Hamiltonian contains a Laplacian, we showed that using an analytical expression saves a lot of CPU cycles compared to calculating the second derivative numerically, allowing us to simulate larger systems. We used Gradient Descent to confirm that the minimal energy is attained at $\alpha \approx 0.5$. Extending the simulations to the interacting case, where the particles are modeled as hard spheres, we found that the repulsive force leads to an increased energy. The offset, compared to the non-interacting case, grows with the number of particles, that is, $\langle H \rangle / N$ gets larger as the number of particles N gets larger, with an increase of 0.2% for 3 particles compared to an increase of 10% for 100 particles. This is accompanied by an increased delocalization of the wave function. We are effectively reproducing the results in [1].

In this article, we have focused on the energy $\langle H \rangle$, however, there are other values of interest we could have looked at, such as the average inter-particle distance $\langle |\mathbf{r}_i - \mathbf{r}_j| \rangle$, $i \neq j$, the kinetic energy $\langle K \rangle$, or even the momentum $\langle p \rangle$. We could also have looked at different values for the particle radius a, or even more particles N. It would be interesting to implement different functions that are used to describe Bosonic behaviour, such as the Gross-Pitaevskii-equation, which was done in [1, 4]. More advanced methods to find the parameter α could have been used, such as Newton's method. We also had only one variational parameter; a more complex guess on the shape of the wave function could give more correct results.

5 Critique

Overall, this was a very interesting and fun project. The topic of Bose-Einstein-Condensation is interesting, and I would honestly have preferred a to learn a little bit more about that and the Gross-Pitaevskii equation, given that it is this we wrote the article about.

The work load is high, but not too high, though I would have preferred less tasks of the type "do this in 1D, 2D, 3D for 1,5,10,100,500 particles". This felt a little bit like busy work, but as it was part of the exercise, we felt like we had to include it in the report. I also think that the numerical laplace-operator felt a little bit like busy work.

The lectures are great, but I feel that we were a little bit slow. I would have preferred to have the final topic a week or two before the deadline (even though it was moved), not the day before.

Finally, the Conclusion and the Introduction are usually written in a couple of hours, while, by far, most work is put in the theory and results part. We experienced something similar with FYS3150 and FYS-STK3155. For that reason, we think that these are weighted too much in the final grade.

6 Appendix

6.1 Proofs

6.1.1 Hamiltonian in reduced units

Taking the Hamiltonian

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{6.1}$$

and writing $x = x' a_{ho}$ (and similarly for y and z), we get

$$\begin{aligned} \frac{-\hbar^2}{2m} \nabla_i^2 &= \frac{-\hbar^2}{2m} \left(\frac{\partial^2}{\partial (x_i' a_{ho})^2} + \frac{\partial^2}{\partial (y_i' a_{ho})^2} + \frac{\partial^2}{\partial (z_i' a_{ho})^2} \right) \\ &= \frac{-\hbar^2}{2m} \frac{1}{a_{ho}^2} \nabla_i'^2 = \frac{-\hbar^2}{2m} m\omega_{ho}/\hbar \nabla_i'^2 = -\frac{\hbar\omega_{ho}}{2} \nabla_i'^2 \end{aligned} \tag{6.2}$$

and

$$\begin{aligned} \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] &= \frac{1}{2} m a_{ho}^2 [\omega_{ho}^2 (x'^2 + y'^2) + \omega_z^2 z'^2] \\ &= \frac{\hbar}{2\omega_{ho}} [\omega_{ho}^2 (x'^2 + y'^2) + \omega_z^2 z'^2] \\ &= \frac{1}{2} \hbar\omega_{ho} [(x'^2 + y'^2) + \frac{\omega_z^2}{\omega_{ho}^2} z'^2] \end{aligned} \tag{6.3}$$

Realizing that $|\mathbf{r}_i - \mathbf{r}_j| \leq a \to a_{ho} |\mathbf{r}'_i - \mathbf{r}'_j| \leq a \to |\mathbf{r}'_i - \mathbf{r}'_j| \leq a/a_{ho}$ and inserting, we arrive at previously stated statement (putting $\hbar\omega_{ho}$ outside the summation sign).

6.1.2 Derivatives of the Trial wave function

The trial wave function for N particles is given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right], \tag{6.4}$$

where

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\{[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)]\} = \phi(\mathbf{r}_i) \tag{6.5}$$

and

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{6.6}$$

Defining $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ and $u(r_{ij}) = \ln f(r_{ij})$, this can be rewritten as

$$\Psi_T(\mathbf{r}) = \left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \tag{6.7}$$

The gradient with respect to particle k is then:

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \left(\left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \right) \tag{6.8}$$

$$= \left(\nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] \right) \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} + \left[\prod_i \phi(\mathbf{r}_i) \right] \left(\nabla_k \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \right) \tag{6.9}$$

$$= \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \tag{6.10}$$

$$+ \left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \sum_{l\neq k} \nabla_k u(r_{kl}) \tag{6.11}$$

where we used the product rule for gradients and the fact that only $\phi(\mathbf{r}_k)$ and $\exp\left(\sum_{l\neq k} u(r_{kl}) \right)$ are functions of $\mathbf{r}_k$.
The second derivative divided by the trial wave function is then given by

$$\frac{1}{\Psi_T(\mathbf{r})} \nabla_k^2 \Psi_T(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} \nabla_k \cdot (\nabla_k \Psi_T(\mathbf{r})) \tag{6.12}$$

$$= \frac{1}{\Psi_T(\mathbf{r})} \nabla_k \cdot \left(\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \right) \tag{6.13}$$

$$+ \frac{1}{\Psi_T(\mathbf{r})} \nabla_k \cdot \left(\left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \sum_{l\neq k} \nabla_k u(r_{kl}) \right) \tag{6.14}$$

$$= \frac{1}{\Psi_T(\mathbf{r})} \left(\nabla_k^2 \phi(\mathbf{r}_k) \right) \left[\prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \tag{6.15}$$

$$+ \frac{1}{\Psi_T(\mathbf{r})} \left(\left[\prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \nabla_k \phi(\mathbf{r}_k) \cdot \sum_{l\neq k} \nabla_k u(r_{kl}) \right) \tag{6.16}$$

$$+ \frac{1}{\Psi_T(\mathbf{r})} \left(\left[\prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \nabla_k \phi(\mathbf{r}_k) \cdot \sum_{l\neq k} \nabla_k u(r_{kl}) \right) \tag{6.17}$$

$$+ \frac{1}{\Psi_T(\mathbf{r})} \left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \left(\sum_{l\neq k} \nabla_k u(r_{kl}) \right)^2 \tag{6.18}$$

$$+ \frac{1}{\Psi_T(\mathbf{r})} \left[\prod_i \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{j<m} u(r_{jm}) \right) \right\} \sum_{l\neq k} \nabla_k^2 u(r_{kl}) \tag{6.19}$$

$$= \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \cdot \sum_{l\neq k} \nabla_k u(r_{kl}) \tag{6.20}$$

$$+ \sum_{l\neq k} \nabla_k u(r_{kl}) \cdot \sum_{i\neq k} \nabla_k u(r_{ki}) \tag{6.21}$$

$$+ \sum_{l\neq k} \nabla_k^2 u(r_{kl}) \tag{6.22}$$

where we used the product rule for divergences and gradients. The expressions for the gradient of $u(r_{ik})$ can be re-expressed using the chain rule. We have that, by the chain rule

$$\frac{\partial u(r_{ki})}{\partial x_k} \hat{\mathbf{x}}_\mathbf{k} = \frac{\partial r_{ki}}{\partial x_k} \frac{\partial u(r_{ki})}{\partial r_{ki}} \hat{\mathbf{x}}_\mathbf{k} = u'(r_{ki}) \frac{x_k - x_i}{r_{ki}} \hat{\mathbf{x}}_\mathbf{k} \tag{6.23}$$

and we get the equivalent expressions for the partial derivatives with respect to y_k and z_k due to symmetry. Combining these, we get

$$\nabla_k u(r_{ki}) = \frac{(\mathbf{r}_k - \mathbf{r}_i)}{r_{ki}} u'(r_{ki}) \tag{6.24}$$

Similarly, we get for the second partial derivative that

$$\frac{\partial^2 u(r_{ki})}{\partial x_k^2} = \frac{\partial}{\partial x_k}\left(u'(r_{ki})\frac{x_k - x_i}{r_{ki}}\right) = \frac{\partial u'(r_{ki})}{\partial x_k}\frac{x_k - x_i}{r_{ki}} + u'(r_{ki})\frac{\partial}{\partial x_k}\left(\frac{x_k - x_i}{r_{ki}}\right) \tag{6.25}$$

$$= u''(r_{ki})\frac{(x_k - x_i)^2}{r_{ki}^2} + u'(r_{ki})\left(\frac{1}{r_{ki}} - \frac{(x_k - x_i)^2}{r_{ki}^3}\right) \tag{6.26}$$

and again, we get equivalent expressions for the second partial derivatives with respect to y_k and z_k. Because $(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2 = r_{k_i}^2$, the Laplacian becomes

$$\nabla_k^2 u(r_{kl}) = u''(r_{ki})\frac{r_{ki}^2}{r_{ki}^2} + u'(r_{ki})\left(\frac{3}{r_{ki}} - \frac{r_{ki}^2}{r_{ki}^3}\right) = u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj}) \tag{6.27}$$

Inserting this in the expression for the second derivative, we get equation (2.9). Here are the analytic expressions for the quantum force and the second derivative: The quantum force $\frac{2\nabla\psi}{\psi}$ for particle k is given by:

$$\begin{aligned}
\frac{2\nabla_k\psi}{\psi} &= \frac{2\nabla_k\phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\Big\{\Big(\sum_{j<m}u(r_{jm})\Big)\Big\} + 2\left[\prod_i \phi(\mathbf{r}_i)\right]\exp\Big\{\Big(\sum_{j<m}u(r_{jm})\Big)\Big\}\sum_{l\neq k}\nabla_k u(r_{kl})}{\psi} \\
&= \frac{-4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k)\psi + 2\psi\sum_{l\neq k}\nabla_k u(r_{kl})}{\psi} \\
&= -4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) + 2\sum_{l\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_l)}{r_{kl}}u'(r_{kl}) \\
&= -4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) + 2\sum_{l\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_l)}{r_{kl}}\frac{-a}{ar_{kl} - r_{kl}^2} \\
&= -4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) - \sum_{l\neq k}\frac{2a}{ar_{kl}^2 - r_{kl}^3}(\mathbf{r}_k - \mathbf{r}_l)
\end{aligned} \tag{6.28}$$

Now looking at the second derivative, we find that We have the formula given by

$$\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \cdot \sum_{l\neq k}\nabla_k u(r_{kl}) \tag{6.29}$$

$$+ \sum_{l\neq k}\nabla_k u(r_{kl}) \cdot \sum_{i\neq k}\nabla_k u(r_{ki}) \tag{6.30}$$

$$+ \sum_{l\neq k}\nabla_k^2 u(r_{kl}) \tag{6.31}$$

$$= \left[4\alpha^2\left[x_k^2 + y_k^2 + \beta^2 z_k^2\right] - 4\alpha - 2\alpha\beta\right] - 4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) \cdot \sum_{k\neq j}\frac{-a}{ar_{kj}^2 - r_{kj}^3}(\mathbf{r}_k - \mathbf{r}_j) \tag{6.32}$$

$$+ \sum_{k\neq i}\sum_{k\neq j}(\mathbf{r}_k - \mathbf{r}_i)\cdot(\mathbf{r}_k - \mathbf{r}_j)\frac{a^2}{(ar_{kj}^2 - r_{kj}^3)(ar_{ki}^2 - r_{ki}^3)} \tag{6.33}$$

$$+ \sum_{k\neq j}\frac{a(a - 2r_{kj})}{r_{kj}^2(a - r_{kj})^2} - \frac{2a}{(ar_{kj}^2 - r_{kj}^3)} \tag{6.34}$$

$$= \left[4\alpha^2\left[x_k^2 + y_k^2 + \beta^2 z_k^2\right] - 4\alpha - 2\alpha\beta\right] - 4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) \cdot \sum_{k\neq j}\frac{-a}{ar_{kj}^2 - r_{kj}^3}(\mathbf{r}_k - \mathbf{r}_j) \tag{6.35}$$

$$+ \sum_{k\neq i}\frac{-a(\mathbf{r}_k - \mathbf{r}_i)}{(ar_{ki}^2 - r_{ki}^3)} \cdot \sum_{k\neq j}\frac{-a(\mathbf{r}_k - \mathbf{r}_j)}{(ar_{kj}^2 - r_{kj}^3)} \tag{6.36}$$

$$+ \sum_{k\neq j}\frac{-a^2}{r_{kj}^2(a - r_{kj})^2} \tag{6.37}$$

$$= \left[4\alpha^2\left[x_k^2 + y_k^2 + \beta^2 z_k^2\right] - 4\alpha - 2\alpha\beta\right] - 4\alpha(x_k\hat{x}_k + y_k\hat{y}_k + \beta z_k\hat{z}_k) \cdot \sum_{k\neq j}\frac{-a}{ar_{kj}^2 - r_{kj}^3}(\mathbf{r}_k - \mathbf{r}_j) \tag{6.38}$$

$$+ \left[\sum_{k\neq i}\frac{-a(\mathbf{r}_k - \mathbf{r}_i)}{(ar_{ki}^2 - r_{ki}^3)}\right]^2 \tag{6.39}$$

$$+ \sum_{k\neq j}\frac{-a^2}{r_{kj}^2(a - r_{kj})^2} \tag{6.40}$$

Where the fact that the vector product is linear was used in the last equality.

6.1.3 Numerical differentiation of the non-interacting trial function

Since the trial function is seperable in the particles $\Psi_T(\mathbf{R}) = \prod_j^N \psi_j(\mathbf{r}_j)$ we have that

$$\frac{\partial^2\Psi_T}{\partial x_i^2} = \frac{1}{\psi_i(\mathbf{r}_i)}\frac{\partial^2\psi_i(\mathbf{r}_i)}{\partial x_i^2}\prod_j^N\psi_j(\mathbf{r}_j) = \frac{1}{\psi_i(\mathbf{r}_i)}\frac{\partial^2\psi_i(\mathbf{r}_i)}{\partial x_i^2}\Psi_T(\mathbf{R}). \tag{6.41}$$

So the Laplacian with respect to all the particle coordinates is given by

$$\nabla^2\Psi_T(\mathbf{R}) \equiv \left(\sum_i^N\nabla_i^2\right)\Psi_T(\mathbf{R}) = \Psi_T(\mathbf{R})\sum_i^N\frac{1}{\psi_i(\mathbf{r}_i)}\left[\frac{\partial^2\psi_i(\mathbf{r}_i)}{\partial x_i^2} + \frac{\partial^2\psi_i(\mathbf{r}_i)}{\partial y_i^2} + \frac{\partial^2\psi_i(\mathbf{r}_i)}{\partial z_i^2}\right]. \tag{6.42}$$

We use the standard way to approximate a second derivative,

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \tag{6.43}$$

So in our case we have

$$\frac{\nabla^2 \Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})} \approx \tag{6.44}$$

$$\sum_i^N \frac{\psi_i(x_i+h) + \psi_i(y_i+h) + \psi_i(z_i+h) + \psi_i(x_i-h) + \psi_i(y_i-h) + \psi_i(z_i-h) - 2[\psi_i(x_i) + \psi_i(y_i) + \psi_i(z_i)]}{\psi_i(\mathbf{r}_i)h^2} \tag{6.45}$$

$$= \frac{1}{h^2} \sum_i^N \left[\frac{\psi_i(x_i+h) + \psi_i(y_i+h) + \psi_i(z_i+h) + \psi_i(x_i-h) + \psi_i(y_i-h) + \psi_i(z_i-h)}{\psi_i(\mathbf{r}_i)} - 6 \right] \tag{6.46}$$

6.1.4 Gradient of the local energy

$$\frac{\partial}{\partial \alpha} \Psi_T \propto \frac{\partial}{\partial \alpha} \prod_i^N g(\mathbf{r}_i, \alpha, \beta) \tag{6.47}$$

$$= \sum_l^N \left[\frac{\partial}{\partial \alpha} g(\mathbf{r}_l, \alpha, \beta) \right] \prod_{i \neq l}^N g(\mathbf{r}_i, \alpha, \beta) = -\sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) g(\mathbf{r}_l, \alpha, \beta) \prod_{i \neq l}^N g(\mathbf{r}_i, \alpha, \beta) \tag{6.48}$$

$$= -\sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \prod_i^N g(\mathbf{r}_i, \alpha, \beta) = -\prod_i^N g(\mathbf{r}_i, \alpha, \beta) \sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \tag{6.49}$$

$$\frac{\partial}{\partial \alpha} \Psi_T = -\prod_i^N g(\mathbf{r}_i, \alpha, \beta) \prod_{j<k}^N f(r_{jk}, a) \sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) = -\Psi_T \sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \tag{6.50}$$

$$\frac{\partial_\alpha \Psi_T}{\Psi_T} = -\sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \tag{6.51}$$

Inserting this into Eq. (2.52) we get

$$\frac{\partial}{\partial \alpha} \langle E_L \rangle = 2 \left[\left\langle \frac{\partial_\alpha \Psi_T}{\Psi_T} E_L \right\rangle - \langle E_L \rangle \left\langle \frac{\partial_\alpha \Psi_T}{\Psi_T} \right\rangle \right] \tag{6.52}$$

$$= 2 \left[\left\langle -\sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) E_L \right\rangle - \langle E_L \rangle \left\langle -\sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \right\rangle \right] \tag{6.53}$$

$$= 2 \left[\langle E_L \rangle \left\langle \sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \right\rangle - \left\langle E_L \sum_l^N \left(x_l^2 + y_l^2 + \beta z_l^2 \right) \right\rangle \right] \tag{6.54}$$

where the expression for the non-interacting case is obtained by setting $\beta = 1$.

6.1.5 Rewriting the total variance

$$\mathrm{Var}(\mu_\alpha) = \frac{1}{m}\sum_\alpha^m (\mu_\alpha - \mu_m)^2 \tag{6.55}$$

$$= \frac{1}{m}\sum_\alpha^m \left(\mu_\alpha^2 - 2\mu_m\mu_\alpha + \mu_m^2\right) = \frac{1}{m}\sum_\alpha^m \mu_\alpha^2 - 2\mu_m\frac{1}{m}\sum_\alpha^m \mu_\alpha + \mu_m^2\frac{1}{m}\sum_\alpha^m 1 \tag{6.56}$$

$$= \frac{1}{m}\sum_\alpha^m \mu_\alpha^2 - 2\mu_m^2 + \mu_m^2 = \frac{1}{m}\sum_\alpha^m \mu_\alpha^2 - \mu_m^2 \tag{6.57}$$

$$= \frac{1}{m}\sum_\alpha^m \frac{1}{n}\sum_k^n x_{\alpha,k}\frac{1}{n}\sum_l^n x_{\alpha,l} - \mu_m^2 = \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n x_{\alpha,k}x_{\alpha,l} - \mu_m^2 \tag{6.58}$$

$$= \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n x_{\alpha,k}x_{\alpha,l} - \mu_m^2 - \mu_m^2 + \mu_m^2 \tag{6.59}$$

$$= \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n x_{\alpha,k}x_{\alpha,l} - \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n \mu_m x_{\alpha,k} - \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n \mu_m x_{\alpha,l} + \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n \mu_m^2 \tag{6.60}$$

$$= \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n \left(x_{\alpha,k}x_{\alpha,l} - \mu_m x_{\alpha,k} - \mu_m x_{\alpha,l} + \mu_m^2\right) = \frac{1}{mn^2}\sum_\alpha^m\sum_k^n\sum_l^n (x_{\alpha,k} - \mu_m)(x_{\alpha,l} - \mu_m) \tag{6.61}$$

$$= \frac{1}{mn^2}\sum_\alpha^m\sum_k^n (x_{\alpha,k} - \mu_m)^2 + \frac{2}{mn^2}\sum_\alpha^m\sum_{k<l}^n (x_{\alpha,k} - \mu_m)(x_{\alpha,l} - \mu_m) \tag{6.62}$$

$$\equiv \frac{\sigma^2}{n} + \frac{2}{mn^2}\sum_\alpha^m\sum_{k<l}^n (x_{\alpha,k} - \mu_m)(x_{\alpha,l} - \mu_m) \tag{6.63}$$

6.2 Tables

Table 5: *$\langle E_L \rangle$ in units of $\hbar\omega$ for different values of the variational parameter α and the number of particles N for 1, 2 and 3 dimensions. Numbers were obtained using 10^5 brute Monte Carlo cycles and a step length of 0.7, discarding 10^4 samples. This is meant as an illustration of what values one can expect from a simulation, not as proper estimates of the true $\langle E_L \rangle$.*

	$\alpha = 0.40$	$\alpha = 0.45$	$\alpha = 0.50$	$\alpha = 0.55$	$\alpha = 0.60$
$\langle E_L \rangle$, N=1, 1D	0.514	0.503	0.500	0.502	0.508
$\langle E_L \rangle$, N=1, 2D	1.028	1.006	1.000	1.004	1.016
$\langle E_L \rangle$, N=1, 3D	1.538	1.508	1.500	1.507	1.526
$\langle E_L \rangle$, N=10, 1D	5.158	5.043	5.000	5.009	5.055
$\langle E_L \rangle$, N=10, 2D	10.263	10.068	10.000	10.041	10.136
$\langle E_L \rangle$, N=10, 3D	15.374	15.091	15.000	15.076	15.271
$\langle E_L \rangle$, N=100, 1D	51.401	50.286	50.000	50.238	50.895
$\langle E_L \rangle$, N=100, 2D	103.057	100.721	100.000	100.252	101.253
$\langle E_L \rangle$, N=100, 3D	153.863	150.869	150.000	150.676	152.489
$\langle E_L \rangle$, N=500, 1D	255.806	251.329	250.000	251.149	254.308
$\langle E_L \rangle$, N=500, 2D	510.351	501.903	500.000	503.282	510.142
$\langle E_L \rangle$, N=500, 3D	768.549	753.882	750.000	753.305	761.323

Table 6: *$\langle E_L \rangle$ in units of $\hbar\omega$ for different values of the variational parameter α and the number of particles N for 1, 2 and 3 dimensions. Numbers were obtained using 10^5 Monte Carlo steps using importance sampling and a step length of 0.7, discarding 10^4 samples. This is meant as an illustration of what values one can expect from a simulation, not as proper estimates of the true $\langle E_L \rangle$.*

	$\alpha = 0.40$	$\alpha = 0.45$	$\alpha = 0.50$	$\alpha = 0.55$	$\alpha = 0.60$
$\langle E_L \rangle$, N=1, 1D	0.513	0.503	0.500	0.502	0.508
$\langle E_L \rangle$, N=1, 2D	1.024	1.005	1.000	1.004	1.017
$\langle E_L \rangle$, N=1, 3D	1.539	1.509	1.500	1.506	1.525
$\langle E_L \rangle$, N=10, 1D	5.126	5.029	5.000	5.022	5.082
$\langle E_L \rangle$, N=10, 2D	10.235	10.051	10.000	10.042	10.158
$\langle E_L \rangle$, N=10, 3D	15.355	15.071	15.000	15.074	15.247
$\langle E_L \rangle$, N=100, 1D	51.259	50.267	50.000	50.267	50.926
$\langle E_L \rangle$, N=100, 2D	102.451	100.557	100.000	100.424	101.566
$\langle E_L \rangle$, N=100, 3D	153.846	150.836	150.000	150.658	152.361
$\langle E_L \rangle$, N=500, 1D	256.647	251.527	250.000	251.206	254.238
$\langle E_L \rangle$, N=500, 2D	511.935	502.623	500.000	502.372	508.172
$\langle E_L \rangle$, N=500, 3D	768.952	754.176	750.000	753.444	762.468

Table 7: *Expectation value $\langle E_L \rangle$ and $\sigma_{Blocking}$ and the naive error estimate σ_E as function of the number of MC cycles for $N = 10$ particles in 3 dimensions with $\alpha = 1.0$ using importance sampling and $\Delta t = 0.01$. Observe that the analytical solution is $\langle E_L \rangle = 18.75$.*

MC cycles	$\langle E_L \rangle$	$\sigma_{Blocking}$	σ_E
2^2	16.1	0.8	0.8
2^3	16.3	0.4	0.4
2^4	16.0	0.2	0.2
2^5	15.8	0.1	0.1
2^6	15.75	0.07	0.07
2^7	16.07	0.2	0.07
2^8	16.43	0.2	0.06
2^9	17.52	0.4	0.08
2^{10}	17.69	0.5	0.06
2^{11}	16.69	0.5	0.05
2^{12}	17.94	0.4	0.04
2^{13}	18.35	0.4	0.03
2^{14}	18.81	0.3	0.02
2^{15}	18.78	0.3	0.02
2^{16}	18.67	0.2	0.01
2^{17}	18.68	0.1	0.01
2^{18}	18.76	0.1	0.01
2^{19}	18.65	0.08	0.004
2^{20}	18.60	0.06	0.003
2^{21}	18.64	0.04	0.002
2^{22}	18.688	0.03	0.001
2^{23}	18.731	0.02	0.001
2^{24}	18.7233	0.02	0.0007

Table 8: *Ideal value for the variational parameter $\boldsymbol{\alpha}$ as function of the number particles N, the step length Δt for Importance sampling, and the number of Monte Carlo samples. The initial estimate was sat to be $\alpha = 0.5$. As learning rate, we chose $\eta = 0.8/N^2$ and for the tolerance $tol = 10^{-5}$, with maximum 50 iterations. Each value is the average value of 4 runs that took place in parallel with the same parameters, but a different random number generator. The number of MC cycles is 10^3 and 10^4 times the number of particles, respectively, and 10% of runs are discarded.* ***This data is not reproducible, as it was run while our code had a bug.*** *We left it in, however, as it is used (not for our final results).*

Number of particles	N=3	N=5	N=10	N=25	N=50	N=100
MC=$10^4 N$ $\Delta t = 0.01$	0.4994	0.4988	0.4975	0.4957	0.4946	0.4938
MC=$10^4 N$ $\Delta t = 0.1$	0.4994	0.4988	0.4975	0.4955	0.4940	0.4900
MC=$10^4 N$ $\Delta t = 1$	0.4994	0.4988	0.4975	0.4943	0.4943	0.4931
MC=$10^3 N$ $\Delta t = 0.01$	0.4994	0.4987	0.4975	0.4958	0.4940	0.4925
MC=$10^3 N$ $\Delta t = 0.1$	0.4995	0.4987	0.4976	0.4954	0.4947	0.4940
MC=$10^3 N$ $\Delta t = 1$	0.4995	0.4988	0.4974	0.4951	0.4932	0.4900

Table 9: *Expected energy $\langle E_L \rangle$ for different values of the number of particles N and the values of α used to obtain them, as well as the energy per particle $\langle E_L \rangle / N$ and the blocking standard deviation $\sigma_{Blocking}$. The value of a true Harmonic Oscillator E_{HO} without repulsion with $\alpha = 0.5$, $\beta = 2.82843$, is presented too. We performed in total 2^{25} MC cycles in parallel on 4 threads (2^{23} per thread) using importance sampling with $\Delta t = 0.1$, additionally discarding $1000N$ samples for each run. Every sample was written to file for the Blocking analysis.*

Number of particles	$N = 3$	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
α	0.4994	0.4988	0.4975	0.4950	0.4940	0.4900
$\langle E_L \rangle$	7.259929	12.12857	24.39852	62.0221	127.269	266.6
$\sigma_{Blocking}$	0.000006	0.00002	0.00007	0.0005	0.001	0.1
$\langle E_L \rangle / N$	2.419976	2.425714	2.439852	2.48088	2.54538	2.660
E_{HO}	7.242645	12.071075	24.14215	60.355375	120.71075	241.4215

References

[1] J. L. DuBois and H. R. Glyde. “Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis”. In: *Phys. Rev. A* 63 (2 Jan. 2001), p. 023602. DOI: 10.1103/PhysRevA.63.023602. URL: https://link.aps.org/doi/10.1103/PhysRevA.63.023602 (cit. on pp. 1, 21–23).

[2] M. Hjort-Jensen. *Advanced Topics in Computational Physics: Computational Quantum Mechanics*. 2021. URL: https://compphysics.github.io/ComputationalPhysics2/doc/LectureNotes/_build/html/intro.html (cit. on p. 2).

[3] H. Flyvbjerg and H. G. Petersen. “Error estimates on averages of correlated data”. In: *The Journal of Chemical Physics* 91.1 (1989), pp. 461–466. DOI: 10.1063/1.457480. eprint: https://doi.org/10.1063/1.457480. URL: https://doi.org/10.1063/1.457480 (cit. on p. 11).

[4] J. K. Nilsen et al. “Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region”. In: *Phys. Rev. A* 71 (5 May 2005), p. 053610. DOI: 10.1103/PhysRevA.71.053610. URL: https://link.aps.org/doi/10.1103/PhysRevA.71.053610 (cit. on p. 23).