

Computational physics II: Project 1

Jon Aleksander Prøitz
(Dated: April 10, 2024)

Code used in the project can be found here: https://github.com/Jonaproitz/FYS4411-Project_1/. The code is logged in different branches coenciding with the problems in the project.

I. INTRODUCTION

Theoretical studies of Bose-Einstein condensates in gases consisting of alkali atoms confined in magnetic or optical traps are often based off of the Gross-Pitaevskii equation. Note that validity of these studies are dependent on the condition that the system is dilute, or in other words that the average distance between atoms is much larger than the range of the inter-atomic interactions. Thus interactions of the system is largely dominated by two-body collisions which are well described by the s-wave scattering length a . When defining the diluteness of a gas an important aspect is the gas parameter $x(r) = n(r)a^3$, where $n(r)$ is the local density of the system, and for low values of the average gas parameter $x_{av} = 10^{-3}$ the mean field Gross-Pitaevskii equation works excidingly well. Yet it has been shown experimentally that the local gas parameter can exceed this value by tuning the scattering length in the presence of a Feshbach resonance. Thus improved many-body methods like Monte-Carlo calculations may be required.

The goal of this project is to evaluate the ground state energy of a trapped, hard sphere Bose gas for different number of particles, using a specific trial wavefunction Ψ_T . We have used the Variational Monte Carlo method, starting with non-interacting particles in a spherical trap to check numerical data against analytical results. The trial wavefunction is used to study the sensitivity of condensate and non-condensate properties to the hard sphere radius and the number of particles.

Our numerical studies start out with a simple Variational Monte Carlo method using the Metropolis algorithm to test the validity of our numerical data. We will then go on to add importance sampling based on the Fokker-Planck and Langevin equations, before adding a stochastic gradient descent algorithm to find optimal values of the trial wavefunction parameter α . The result are then analyzed with the blocking technique. Finally we can then parallelize our program and start tackling the repulsive two-body problem.

II. QUANTUM MONTE CARLO THEORY

Quantum Monte Carlo

We approach our system with a numerical Monte Carlo calculation, using the Metropolis algorithm. In general for systems consisting of several particles, the integrals involved in the calculations of various expectation values are of a multi-dimensional characteristic and traditional methods such as Gauss-Legendre quadrature will not be suitable for our manybody system. The Monte Carlo calculation tests a variety of different states averaging the expectation value of the system, relying on the variational principle to yield the lowest states with a given symmetry. However in most cases, a wave function yields small values in large parts of configuration space, and a straightforward procedure which uses homogenously distributed random points in configuration space will most likely lead to poor results. Therefore we will apply the Metropolis algorithm combined with an importance sampling to more efficiently obtain the ground state energies. This is done with the goal that the regions in configuration space where the wavefunction has higher values will be sampled more efficiently.

With this in mind the basic procedure of a Variational Monte Carlo calculations consists of:

- Constructing a trial wavefunction, for a manybody system consisting of n particles located in positions $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_n)$. The trial wavefunction depends on m variational parameters $\alpha = (\alpha_1, \dots, \alpha_m)$.
- Evaluating the expectation value of the Hamiltonian H .

- Vary α according to some minimization algorithm and return to the first step if we are not satisfied with the results.

In order to bring in the Monte Carlo machinery, we must define a probability density distribution (PDF). Using our ansatz for the trial wave function $\Psi_T(\mathbf{r}; \alpha)$ we define a trial PDF

$$P(\mathbf{r}) = \frac{|\Psi_T(\mathbf{r}; \alpha)|^2}{\int |\Psi_T(\mathbf{r}; \alpha)|^2 d\mathbf{r}} \quad (1)$$

which is our model for the probability distribution function. The approximation to the expectation value of the Hamiltonian is now

$$\bar{E}[\alpha] = \frac{\int d\mathbf{r} \Psi_T^*(\mathbf{r}; \alpha) H(\mathbf{r}) \Psi_T(\mathbf{r}; \alpha)}{\int d\mathbf{r} \Psi_T^*(\mathbf{r}; \alpha) \Psi_T(\mathbf{r}; \alpha)} \quad (2)$$

We can then define a new quantity called the local energy

$$E_L(\mathbf{r}; \alpha) = \frac{1}{\Psi_T(\mathbf{r}; \alpha)} H \Psi_T(\mathbf{r}; \alpha) \quad (3)$$

which, together with our trial PDF gives

$$\bar{E}[\alpha] = \int d\mathbf{r} P(\mathbf{r}) E_L(\mathbf{r}; \alpha) \approx \frac{1}{N} \sum_{i=1}^N E_L(\mathbf{r}; \alpha) \quad (4)$$

where N is the number of samples in the Monte Carlo calculation. The expression on the right hand side stems from Bernoulli's law of large numbers, which states that in the limit $N \rightarrow \infty$, the sample mean approaches the true mean.

With the above in hand we can now define the main lines of our Monte Carlo algorithm, which goes as:

- Fix the number of Monte Carlo steps with an initial set of positions \mathbf{r} and variational parameters α , then calculate $|\Psi_T^\alpha(\mathbf{r})|^2$
- Initialise the energy and the variance and start the Monte Carlo calculation.
 - Calculate a trial position $\mathbf{r}_p = \mathbf{r} + a \cdot \text{step}$ where a is a random variable.
 - Use the Metropolis algorithm to accept or reject this move, with a probability of acceptance $w = P(\mathbf{r}_p)/P(\mathbf{r})$
 - If the step is accepted, then we set $\mathbf{r} = \mathbf{r}_p$
 - Update averages
- Finish and compute final averages

Note that the jumping in space is governed by the variable step. This is often referred to as the brute-force sampling and is normally replaced by what is called importance sampling.

Importance sampling

For a diffusion process characterized by a time-dependent probability density $P(x, t)$ in one dimension the Fokker-Planck equation is

$$\frac{\partial P}{\partial t} = D \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} - F \right) P(x, t) \quad (5)$$

where F is a drift term and D is the diffusion coefficient. The new position in coordinate space is given as the solutions of the Langevin equation using Euler's method, namely, we go from the Langevin equation

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \quad (6)$$

with η is a random variable, giving a new position

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \quad (7)$$

where ξ is gaussian random variable, Δt is the chosen time step and the quantity D is equal to $1/2$ in atomic units. Note that Δt is viewed as a parameter. In general values of $\Delta t \in [0.001, 0.01]$ yield rather stable values of the ground state energy.

The process of isotropic diffusion characterized by a time-dependent probability density $P(x, t)$ approximately follows the Fokker-Planck equation given by

$$\frac{\partial P}{\partial t} = \sum_i D \frac{\partial}{\partial \mathbf{x}_i} \left(\frac{\partial}{\partial \mathbf{x}_i} - \mathbf{F}_i \right) P(x, t) \quad (8)$$

where \mathbf{F}_i is the i th component of the drift term caused by an external potential, and D is the diffusion coefficient. The convergence to a stationary probability density can be obtained by setting the left hand side to zero such that the resulting equation will be satisfied if and only if all the terms of the sum are equal zero

$$\frac{\partial^2 P}{\partial \mathbf{x}_i^2} = P \frac{\partial}{\partial \mathbf{x}_i} \mathbf{F}_i + \mathbf{F}_i \frac{\partial}{\partial \mathbf{x}_i} P \quad (9)$$

The drift vector should be of the form $\mathbf{F} = g(\mathbf{x}) \frac{\partial P}{\partial \mathbf{x}}$. wich then gives

$$\frac{\partial^2 P}{\partial \mathbf{x}_i^2} = P \frac{\partial g}{\partial P} \left(\frac{\partial P}{\partial \mathbf{x}_i} \right)^2 + P g \frac{\partial^2 P}{\partial \mathbf{x}_i^2} + g \left(\frac{\partial P}{\partial \mathbf{x}_i} \right)^2 \quad (10)$$

The condition of stationary density means that the left hand side equals zero. Thus the terms containing first and second derivatives have to cancel each other. This is possible only when $g = 1/P$, wich yields

$$\mathbf{F} = 2 \frac{1}{\Psi_T} \nabla \Psi_T \quad (11)$$

which is known as the quantum force. This term is responsible for "pushing" the random walker towards regions of the configuration space where the trial wave function has larger values. This increases the efficiency of the simulation in contrast to the brute force Metropolis algorithm where the walker has the same probability of moving in every direction.

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

$$G(y, x, \Delta t) = \frac{1}{(4\pi D \Delta t)^{3N/2}} \exp(-(y - x - D \Delta t F(x))^2 / 4 D \Delta t) \quad (12)$$

which in turn means that our brute force Metropolis algorithm

$$A(y, x) = \min(1, q(y, x)) \quad (13)$$

with $q(y, x) = |\Psi_T(y)|^2 / |\Psi_T(x)|^2$ is now replaced by the Metropolis-Hastings algorithm

$$q(y, x) = \frac{G(x, y, \Delta t) |\Psi_T(y)|^2}{G(x, y, \Delta t) |\Psi_T(x)|^2} \quad (14)$$

Gradient descent

In general when doing Monte Carlo calculations, we end up computing the expectation value of the energy in terms of some parameters $\alpha_0, \alpha_1, \dots, \alpha_n$ and search for a minimum in this multi-variable parameter space. This leads to an energy minimization problem where we need the derivative of the energy as a function of the variational parameters. To find the derivatives of the local energy expectation value as function of the variational parameters, we can use the chain rule and the hermiticity of the Hamiltonian. Limiting ourself to one variable we then define

$$\bar{E}_\alpha = \frac{dE_L[\alpha]}{d\alpha} \quad (15)$$

as the derivative of the energy with respect to the variational parameter α . We then also define the derivative of the trial function with respect to the paramter α

$$\bar{\Psi}_\alpha = \frac{d\Psi[\alpha]}{d\alpha} \quad (16)$$

The elements of the gradient of the local energy are then

$$\bar{E}_\alpha = 2 \left(\left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} E_L[\alpha] \right\rangle - \left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} \right\rangle \langle E_L[\alpha] \rangle \right) \quad (17)$$

From a computational point of view this means that we need to compute the expectation values of

$$\left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} E_L \right\rangle \quad (18)$$

and

$$\left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} \right\rangle \langle E_L[\alpha] \rangle \quad (19)$$

The basic idea of the gradient descent method is that a function $F(\mathbf{x})$, $\mathbf{x} \equiv (x_1, x_2, \dots, x_n)$, decreases fastest if one goes from \mathbf{x} in the direction of the negative gradient $-\nabla F(\mathbf{x})$. It can be shown that if

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla F(\mathbf{x}_k) \quad (20)$$

with $\gamma_k > 0$. If γ_k is sufficiently small, then $F(\mathbf{x}_{k+1}) \leq F(\mathbf{x}_k)$. This means that for a sufficiently small γ_k we are always moving towards smaller function values, i.e a minimum.

The previous observation is the basis of the steepest descent method, which is also referred to as just the Gradient Descent method. Here we start with an initial guess x_0 for a minimum of F and compute new approximations according to

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla F(\mathbf{x}_k), \quad k \geq 0 \quad (21)$$

The parameter γ_k is also often referred to as the step length or the learning rate within the context of Machine Learning.

Ideally the sequence $\{\mathbf{x}_k\}_{k=0}$ converges to a global minimum of the function F . However in general we do not know if we are in a global or local minimum of the function. In the special case when F is a convex function, all local minima are also global minima, so in this case gradient descent can converge to the global solution. The advantage of this scheme is that it is conceptually simple and straightforward to implement. However the method in this form has some severe limitations.

In machine learning we are often faced with non-convex high dimensional cost functions with many local minima. Since the Gradient Descent method is deterministic we will get stuck in a local minimum, if the method converges, unless we have a very good initial guess. This also implies that the scheme is sensitive to the chosen initial condition. Note that the gradient is a function of $\mathbf{x} = (x_1, \dots, x_n)$ which makes it expensive to compute numerically. The gradient descent method is also sensitive to the choice of learning rate γ_k . This stems from the fact that we are only guaranteed that $F(\mathbf{x}_{k+1}) \leq F(\mathbf{x}_k)$ for sufficiently small values of γ_k . The problem is to determine an optimal learning rate. If the learning rate is chosen too small the method will take a long time to converge and if it is too large we can experience erratic behavior. Many of these shortcomings can be alleviated by introducing randomness. One such method is that of Stochastic Gradient Descent.

Resampling methods

Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample such that we obtain additional information about the fitted model. For example, in order to estimate the variability of a linear regression fit, we can repeatedly draw different samples from the training data, fit a linear regression to each new sample, and then examine the extent to which the resulting fits differ. Such an approach may allow us to obtain information that would not be available from fitting the model only once using the original training sample.

Resampling methods are an important tool in modern statistics. However these approaches can be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data. Yet, due to recent advances in computing power, the computational requirements of resampling methods generally are not prohibitive.

The reasons that we choose to use these resampling methods are that we treat the computer simulations as experiments run numerically on a computer. This is especially true for the Monte Carlo methods we have used in this

project. We therefore analyse our results using the same statistics tools as we would use analysing experimental data. And as in all experiments, we are looking for expectation values and an estimate of how accurate they are, i.e. possible sources for errors. As in other experiments, these sources of errors come in two classes, statistical and systematical errors. Statistical errors can be estimated using standard tools from statistics, while systematical errors are method specific and must be treated differently from case to case.

In this project we will use the Blocking method, made popular by Flyvbjerg and Pedersen (1989) and it has since become one of the main ways to estimate $V(\hat{\theta})$ for exactly one $\hat{\theta}$, namely $\hat{\theta} = \bar{X}$. The strength of the blocking method is when the number of observations, n is large. For instance for large n , the complexity of dependent bootstrapping scales poorly, but the blocking method does not, moreover, it becomes more accurate the larger n is.

In the Blocking method we assume $n = 2^d$ where d is an integer greater than 1 and X_1, X_2, \dots, X_n is a stationary asymptotically uncorrelated time series. Then by switching to vector notation by arranging X_1, X_2, \dots, X_n in an n -tuple, we can define

$$\hat{\mathbf{X}} = (X_1, X_2, \dots, X_n) \quad (22)$$

We can now define blocking transformations. The idea is to take the mean of subsequent pairs of elements from $\hat{\mathbf{X}}$ and form a new vector $\hat{\mathbf{X}}_1$. Continuing in the same way by taking the mean of subsequent pairs of elements of $\hat{\mathbf{X}}_1$ we obtain $\hat{\mathbf{X}}_2$, and so on. We then define $\hat{\mathbf{X}}_i$ by

$$(\hat{\mathbf{X}}_0)_k \equiv (\hat{\mathbf{X}})_k \quad (23)$$

and

$$(\hat{\mathbf{X}}_{i+1})_k \equiv \frac{1}{2} \left((\hat{\mathbf{X}}_i)_{2k-1} + (\hat{\mathbf{X}}_i)_{2k} \right) \text{ for all } 1 \leq i \leq d-1 \quad (24)$$

Where the quantity $\hat{\mathbf{X}}_k$ is subject to k blocking transformations. We now have d vectors $\hat{\mathbf{X}}_0, \hat{\mathbf{X}}_1, \dots, \hat{\mathbf{X}}_{d-1}$ containing the subsequent averages of observations. It can then be shown that if the components of $\hat{\mathbf{X}}$ is a stationary time series, then the components of $\hat{\mathbf{X}}_i$ is also a stationary time series for all i where $0 \leq i \leq d-1$. Thus we can compute the autocovariance, the variance, sample mean, and number of observations for each i . Letting γ_i , σ_i and $\bar{\mathbf{X}}_i$ denote the autocovariance, variance and average of the elements of $\hat{\mathbf{X}}_i$ and let n_i be the number of elements of $\hat{\mathbf{X}}$. It then follows by induction that $n_i = n/2^i$.

Using the definition of the blocking transformation and the distributive property of the covariance, it is clear that since $h = |i - j|$ we can define

$$\begin{aligned} \gamma_{k+1}(h) &= \text{cov}((X_{k+1})_i, (X_{k+1})_i) \\ &= \frac{1}{4} \text{cov}((X_k)_{2i-1} + (X_k)_{2i}, (X_k)_{2j-1} + (X_k)_{2j}) \\ &= \frac{1}{2} \gamma_k(2h) + \frac{1}{2} \gamma_k(2h+1)h = 0 \\ &= \frac{1}{4} \gamma_k(2h-1) + \frac{1}{2} \gamma_k(2h) + \frac{1}{4} \gamma_k(2h+1) \text{ else} \end{aligned}$$

The quantity $\hat{\mathbf{X}}$ is asymptotically uncorrelated by assumption, and $\hat{\mathbf{X}}_k$ is also asymptotically uncorrelated.

Let's turn our attention to the variance of the sample mean $V(\bar{X})$. We have

$$V(\bar{X}_k) = \frac{\sigma_k^2}{n_k} + \frac{2}{n_k} \sum_{h=1}^{n_k-1} \left(1 - \frac{h}{n_k} \right) \gamma_k(h) = \frac{\sigma_k^2}{n_k} + e_k \text{ if } \gamma_k(0) = \sigma_k^2 \quad (25)$$

Where e_k is the truncation error:

$$e_k = \frac{2}{n_k} \sum_{h=1}^{n_k-1} \left(1 - \frac{h}{n_k} \right) \gamma_k(h) \quad (26)$$

We can now show that $V(\bar{X}_i) = V(\bar{X}_j)$ for all $0 \leq i \leq d-1$ and $0 \leq j \leq d-1$. Thus

$$n_{j+1}\bar{X}_{j+1} = \sum_{i=1}^{n_j+1} (\hat{\mathbf{X}}_{j+1})_i = \frac{1}{2} \sum_{i=1}^{n_j/2} (\hat{\mathbf{X}}_j)_{2i-1} + (\hat{\mathbf{X}}_j)_{2i} = \frac{n_j}{2} \bar{X}_j = n_{j+1} \bar{X}_j \quad (27)$$

By repeated use of this equation we get $V(\bar{X}_i) = V(\bar{X}_0) = V(\bar{X})$ for all $0 \leq i \leq d-1$. This has the consequence that

$$V(\bar{X}) = \frac{\sigma_k^2}{n_k} + e_k \text{ for all } 0 \leq k \leq d-1 \quad (28)$$

Flyvbjerg and Petersen demonstrated that the sequence $\{e_k\}_{k=0}^{d-1}$ is decreasing, and conjecture that the term e_k can be made as small as we would like by making k sufficiently large. Thus we can apply blocking transformations until e_k is sufficiently small, and then estimate $V(\bar{X})$ by $\hat{\sigma}_k^2/n_k$.

III. ANALYTIC EXPRESSIONS

We represent our system with either a spherical (S), or elliptical (E) trap, given by

$$V_{\text{ext}}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \quad (29)$$

in the spherical case ω_{ho} defines the trap potential, while for the elliptical case ω_{ho} defines the trap frequency in the xy -plane and ω_z is the trap frequency in the z -direction. The mean square vibrational amplitude of a single boson at $T = 0K$ is $\langle x^2 \rangle = \sqrt{\hbar/2m\omega_{ho}}$ such that the characteristic length is $a_{ho} = \sqrt{\hbar/m\omega_{ho}}$. The two-body Hamiltonian of the system is

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{\text{ext}} \right) + \sum_{i < j}^N V_{\text{int}} \quad (30)$$

with N being the number of particles and the inter-boson interaction is represented by a pairwise repulsive potential

$$V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (31)$$

where a is the hard-core diameter of the bosons. The trial wavefunction for the ground state with N atoms is given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j < k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right], \quad (32)$$

with

$$g(\alpha, \beta, \mathbf{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)] \quad (33)$$

and

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \quad (34)$$

Local energy

For non-interacting bosons we set $a = 0$ in a spherical potential, $\beta = 1$, such that the trial wavefunction can be written

$$\Psi_T(\mathbf{r}) = e^{-\alpha r^2} = e^{-\alpha(x^2 + y^2 + z^2)} \quad (35)$$

The local energy in one dimension is then given by the double derivative in one dimension

$$E_L = \frac{1}{\Psi_T} H \Psi_T = \frac{1}{\Psi_T} \frac{\partial^2 \Psi_T}{\partial x^2} + V_{ext} = -2\alpha \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial x} (x \Psi_T) + V_{ext} = 2\alpha(2\alpha x^2 - 1) + V_{ext} \quad (36)$$

Denoting the vector $\mathbf{r} \equiv \mathbf{x} = (x_1, x_2, x_3)$, with $\nabla^2 = \sum_{i=1}^3 \frac{\partial^2}{\partial x_i^2}$, we obtain the double derivative of a single boson in d dimensions

$$\frac{1}{\Psi_T(\mathbf{x})} \sum_{i=1}^d \frac{\partial^2 \Psi_T(\mathbf{x})}{\partial x_i^2} = 2\alpha \left(2\alpha \sum_{i=1}^d x_i^2 - d \right) = 2\alpha (2\alpha r^2 - d) \quad (37)$$

Hence the local energy of the total system in d dimensions with N particles is given by

$$E_L = \sum_i^N \left(\frac{-\hbar^2}{2m} 2\alpha (2\alpha r_i^2 - d) + \frac{1}{2} m \omega_{ho}^2 r_i^2 \right) = \sum_i^N \frac{d}{2} \hbar \omega_{ho} = \frac{Nd}{2} \hbar \omega_{ho} \quad (38)$$

here we have used $\alpha = 1/2a_{ho}^2$ and $a_{ho} = \sqrt{\hbar/m\omega_{ho}}$ in the second calculation. We note that setting $\hbar = \omega_{ho} = m = 1$ gives $\alpha = 1/2$.

Drift force

Again denoting $\mathbf{r} \equiv \mathbf{x} = (x_1, x_2, x_3)$, the quantum force

$$\mathbf{F} = \frac{2\nabla \Psi_T(\mathbf{x})}{\Psi_T(\mathbf{x})} = \frac{2}{\Psi_T} \sum_{i=1}^d \hat{\mathbf{x}}_i \frac{\partial^2}{\partial x_i^2} e^{-\alpha x^2} = -4\alpha \sum_{i=1}^d x_i \hat{\mathbf{x}}_i \quad (39)$$

Two-body interaction

Define $g(\alpha, \beta, \mathbf{r}_i) = \phi(\mathbf{r}_i)$ and $r_{jm} = |\mathbf{r}_j - \mathbf{r}_m|$, such that

$$\Psi_T = \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j<m} \ln f(r_{jm})} = \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j<m} u(r_{jm})} \quad (40)$$

with $u(r_{jm}) = \ln f(r_{jm})$. The derivative of particle k is then

$$\nabla_k \Psi_T = \nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j<m} u(r_{jm})} + \left[\prod_i \phi(\mathbf{r}_i) \right] \nabla_k \left(e^{\sum_{j<m} u(r_{jm})} \right) \quad (41)$$

we note that in the first term only particle k is affected by the derivation, such that

$$\nabla_k \prod_i \phi(\mathbf{r}_i) = \nabla_k \phi(\mathbf{r}_i) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \quad (42)$$

while the second term

$$\nabla_k \left(e^{\sum_{j<m} u(r_{jm})} \right) = e^{\sum_{j<m} u(r_{jm})} \nabla_k \sum_{j<m} u(r_{jm}) = e^{\sum_{j<m} u(r_{jm})} \sum_{l \neq k} \nabla_k u(r_{kl}) \quad (43)$$

such that

$$\nabla_k \Psi_T = \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j<m} u(r_{jm})} + \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j<m} u(r_{jm})} \sum_{l \neq k} \nabla_k u(r_{kl}) \quad (44)$$

To find the double derivative we start by derivating the first term of eq 44

$$\nabla_k \left(\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})} \right) = \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})} \nabla_k^2 \phi(\mathbf{r}_k) + \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \nabla_k e^{\sum_{j < m} u(r_{jm})} \quad (45)$$

the first term of eq 45 is then

$$\frac{\left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})}}{\Psi_T} \nabla_k^2 \phi(\mathbf{r}_k) = \frac{\left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})}}{\phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})}} \nabla_k^2 \phi(\mathbf{r}_k) = \frac{1}{\phi(\mathbf{r}_k)} \nabla_k^2 \phi(\mathbf{r}_k) \quad (46)$$

Using eq 43 we find that the second term of eq 45

$$\frac{\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right]}{\Psi_T} \nabla_k e^{\sum_{j < m} u(r_{jm})} = \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{l \neq k} \nabla_k u(r_{kl}) \quad (47)$$

moving on to the second term of eq 44

$$\begin{aligned} & \nabla_k \left(\left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})} \sum_{l \neq k} \nabla_k u(r_{kl}) \right) \\ &= e^{\sum_{j < m} u(r_{jm})} \sum_{l \neq k} \nabla_k u(r_{kl}) \nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] + \left[\prod_i \phi(\mathbf{r}_i) \right] \sum_{l \neq k} \nabla_k u(r_{kl}) \nabla_k e^{\sum_{j < m} u(r_{jm})} \\ &+ \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})} \nabla_k \sum_{l \neq k} \nabla_k u(r_{kl}) \end{aligned} \quad (48)$$

we find the first term of eq 48

$$\frac{1}{\Psi_T} e^{\sum_{j < m} u(r_{jm})} \sum_{l \neq k} \nabla_k u(r_{kl}) \nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] = \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{l \neq k} \nabla_k u(r_{kl}) \quad (49)$$

then the second term

$$\frac{1}{\Psi_T} \left[\prod_i \phi(\mathbf{r}_i) \right] \sum_{l \neq k} \nabla_k u(r_{kl}) \nabla_k e^{\sum_{j < m} u(r_{jm})} = \left(\sum_{l \neq k} \nabla_k u(r_{kl}) \right) \left(\sum_{i \neq k} \nabla_k u(r_{ki}) \right) \quad (50)$$

and lastly the third term

$$\frac{1}{\Psi_T} \left[\prod_i \phi(\mathbf{r}_i) \right] e^{\sum_{j < m} u(r_{jm})} \nabla_k \sum_{l \neq k} \nabla_k u(r_{kl}) = \sum_{l \neq k} \nabla_k^2 u(r_{kl}) \quad (51)$$

Now with the realtions

$$\nabla_k u(r_{ki}) = \frac{\mathbf{r}_k - \mathbf{r}_i}{r_{ki}} u'(r_{ki}) \quad (52)$$

and

$$\nabla_k^2 u(r_{ki}) = u''(r_{ki}) + \frac{2}{r_{ki}} u'(r_{ki}) \quad (53)$$

we can find the second derivative

$$\begin{aligned} \frac{1}{\Psi_T} \nabla_k^2 \Psi_T &= \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kl}} u'(r_{kj}) \\ &+ \sum_{j \neq k} \sum_{i \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)(\mathbf{r}_k - \mathbf{r}_i)}{r_{kj} r_{ki}} u'(r_{kj}) u'(r_{ki}) + \sum_{j \neq k} u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \end{aligned} \quad (54)$$

Then using

$$u'(r_{kj}) = \frac{d \ln(f(a, r_{kj}))}{dr_{kj}} = \frac{1}{f(a, r_{kj})} f'(a, r_{kj}) = \frac{a}{r_{kj}^2} \frac{1}{f(a, r_{kj})} \quad (55)$$

and

$$u''(r_{kj}) = - \left(2 \frac{a}{r_{kj}^3} \frac{1}{f(a, r_{kj})} + \frac{a^2}{r_{kj}^4} \frac{1}{f^2(a, r_{kj})} \right) \quad (56)$$

we can rewrite equaton 54

$$\begin{aligned} \frac{1}{\Psi_T} \nabla_k^2 \Psi_T &= \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} - 4 (x_k \hat{\mathbf{x}} + y_k \hat{\mathbf{y}} + \beta z_k \hat{\mathbf{z}}) \sum_{k \neq j} (\mathbf{r}_k - \mathbf{r}_j) \frac{a}{r_{kj}^3} \frac{1}{f(a, r_{kj})} \\ &+ \sum_{k \neq j} \sum_{k \neq i} (\mathbf{r}_k - \mathbf{r}_j) (\mathbf{r}_k - \mathbf{r}_i) \frac{a}{r_{kj}^3} \frac{a}{r_{ki}^3} \frac{1}{f(a, r_{kj}) f(a, r_{ki})} - \sum_{k \neq j} \frac{a^2}{r_{kj}^4} \frac{1}{f^2(a, r_{kj})} \end{aligned} \quad (57)$$

Lastly going from $r \rightarrow r/a_{ho}$ we can rewrite the elliptical Hamiltonian

$$\begin{aligned} H &= \sum_{i=1}^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] \right) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \\ \rightarrow H &= \sum_{i=1}^N \left(\frac{-\hbar^2}{2m} \frac{\nabla_i^2}{a_{ho}^2} + \frac{a_{ho}^2}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] \right) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \\ &= \sum_{i=1}^N \left(\frac{-\hbar \omega_{ho}}{2} \nabla_i^2 + \frac{\hbar \omega_{ho}}{2} [x^2 + y^2 + \omega_{ho} \gamma^2 z^2] \right) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \end{aligned}$$

where we have used $a_{ho} = \sqrt{\hbar/m\omega_{ho}}$ and $\gamma = \omega_z/\omega_{ho}$. In units $\hbar\omega_{ho}$ the Hamiltonian can therefore be written

$$\begin{aligned} H &= \sum_{i=1}^N \left(\frac{-1}{2} \nabla_i^2 + \frac{1}{2} [x^2 + y^2 + \omega_{ho} \gamma^2 z^2] \right) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \\ &= \frac{1}{2} \sum_{i=1}^N (-\nabla_i^2 + x^2 + y^2 + \omega_{ho} \gamma^2 z^2) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \end{aligned} \quad (58)$$

Onebody densities

The onebody density for N particles is given by

$$\rho(\mathbf{r}_1) = N \int \prod_{i=2}^N d^3 \mathbf{r}_i |\Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)|^2 \quad (59)$$

Inserting our trial wavefunction from the harmonic oscillator case

$$\rho(\mathbf{r}_1) = N \phi^2(\mathbf{r}_1) \int \prod_{i=2}^N d^3 \mathbf{r}_i \phi^2(\mathbf{r}_i) \prod_{j < i} f^2(a, |\mathbf{r}_i - \mathbf{r}_j|) \quad (60)$$

Limiting ourselves to two particles, we can see that in the non-interacting case the onebody density reduces to

$$\rho(\mathbf{r}_1) = 2\phi^2(\mathbf{r}_1) \int d^3 \mathbf{r}_2 \phi^2(\mathbf{r}_2) \quad (61)$$

while in the interacting case

$$\rho(\mathbf{r}_1) = 2\phi^2(\mathbf{r}_1) \int d^3 \mathbf{r}_2 \phi^2(\mathbf{r}_2) f^2(a, |\mathbf{r}_1 - \mathbf{r}_2|) \quad (62)$$

IV. NUMERICAL APPROACH

The starting point of our numerical calculations is the Monte Carlo calculation described in section II. To build the numerical structure of our program, we use the spherical trap described eq: 29. With non-interacting bosons as our particle of choice. The Metropolis algorithm is used on a single particle in a single dimension at a time and sampled once every particle in every dimension has undergone a single Monte Carlo step. When all Monte Carlo steps are completed we calculate the mean energy and variance of the system for a given parameter α and redo the calculation with different values of α to get an understanding of how the system changes with this parameter.

The calculations are run for $n = 1$, $n = 10$, $n = 100$ and $n = 500$, where n denotes the number of particles, with the parameter $\alpha \in [0.2, 0.9]$. The results of these calculations are shown in figures 1, 2, 3 and 4. For non-interacting bosons, using natural units, we expect to find the minimum energies for $\alpha = 0.5$. This matches all three of our calculations, and we note that the variance for $\alpha = 0.5$ is 0 which is also to be expected from eq 38.

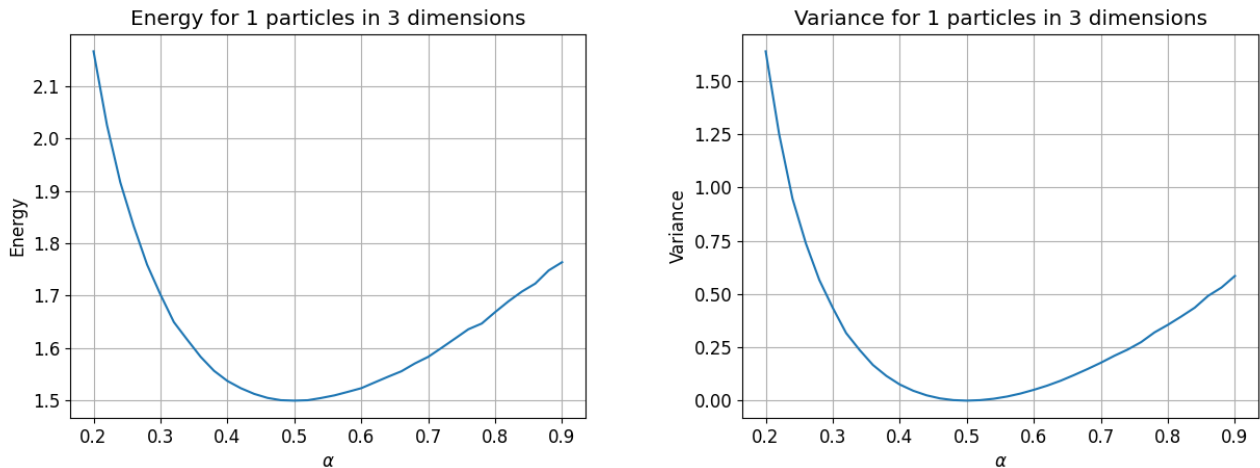


Figure 1. Calculated energy and variance using the Monte Carlo calculations for different values of the parameter α for 1 non-interacting boson in a spherical trap.

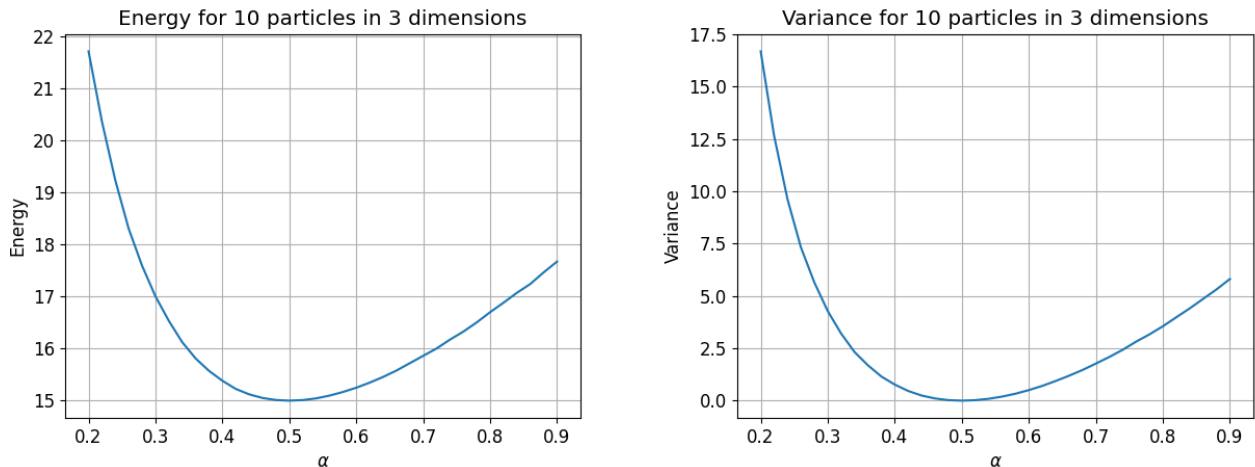


Figure 2. Calculated energy and variance using the Monte Carlo calculations for different values of the parameter α for 10 non-interacting bosons in a spherical trap.

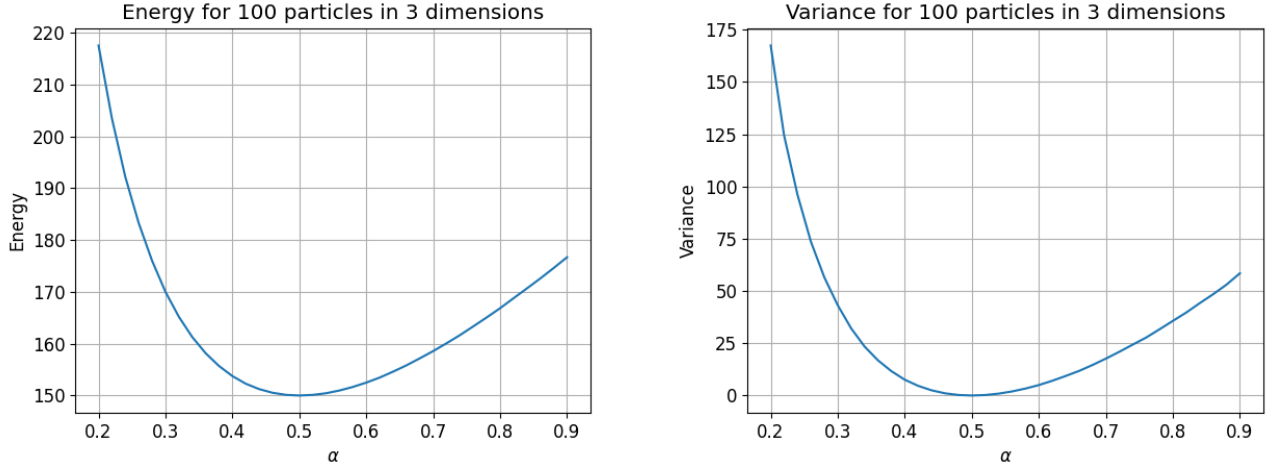


Figure 3. Calculated energy and variance using the Monte Carlo calculations for different values of the parameter α for 100 non-interacting bosons in a spherical trap.

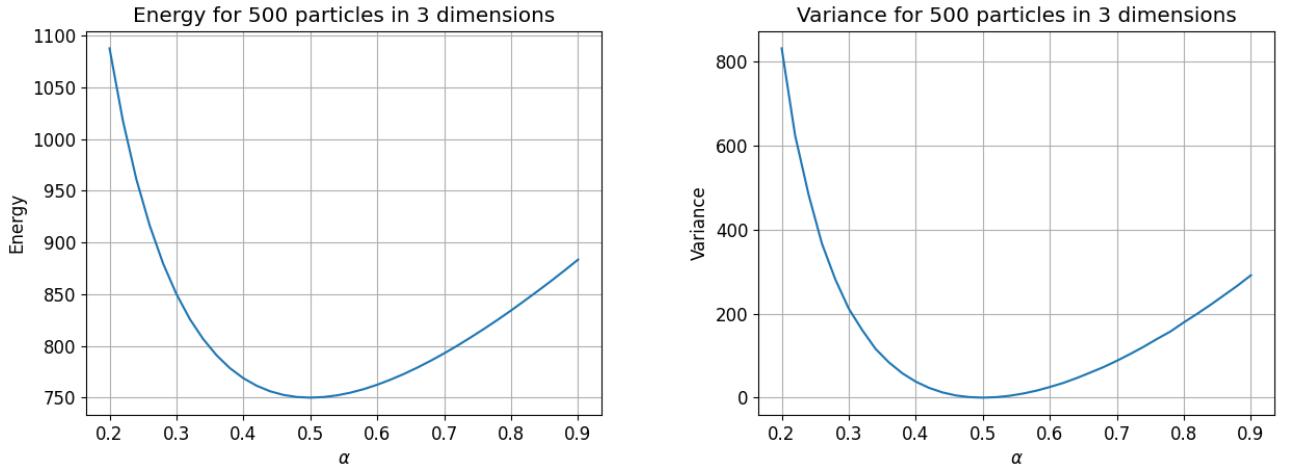


Figure 4. Calculated energy and variance using the Monte Carlo calculations for different values of the parameter α for 500 non-interacting bosons in a spherical trap.

The next goal is to implement importance sampling. This is done using a quantum force based on the Fokker-Planck and Langevin equations. In contrast to the brute force Monte Carlo calculation used previously, the importance sampling adds an "incentive" for the particles to move towards the areas where the trial wave function has greater values, as described in section II. The implementation of importance sampling to our calculations allows us to more efficiently sample the areas in coordinate space where the trial wavefunction is greater. Thus we aim to find the same ground state energies as with the brute force calculations but with fewer steps needed.

The calculations are again run for $n = 1$, $n = 10$, $n = 100$ and $n = 500$ with the parameter $\alpha \in [0.2, 0.9]$. With the new results shown in figures 5, 6, 7 and 8. Comparing these to figures 1, 2, 3 and 4 we note the great similarities and same overall results from the brute force Monte Carlo calculations. The mayor difference between the importance sampling and the brute force way is the number of Monte Carlo cycles used. In the brute force scenario we have used 10^6 cycles while when using importance sampling we have used 10^5 cycles. This naturally leads to lower times spent on these calculations, this is shown in table I.

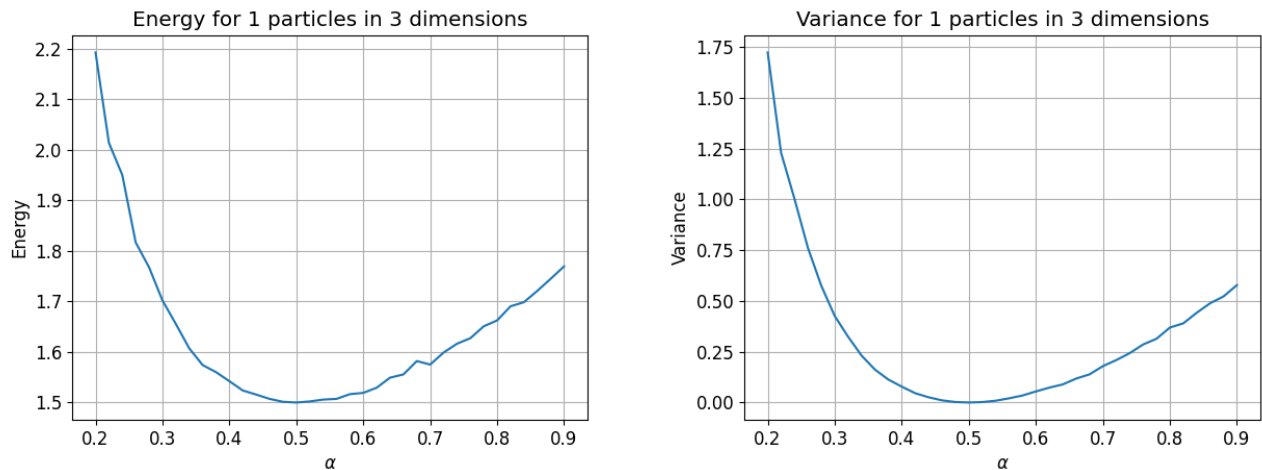


Figure 5. Calculated energy and variance using the Monte Carlo calculations with importance sampling for different values of the parameter α for 1 non-interacting boson in a spherical trap.

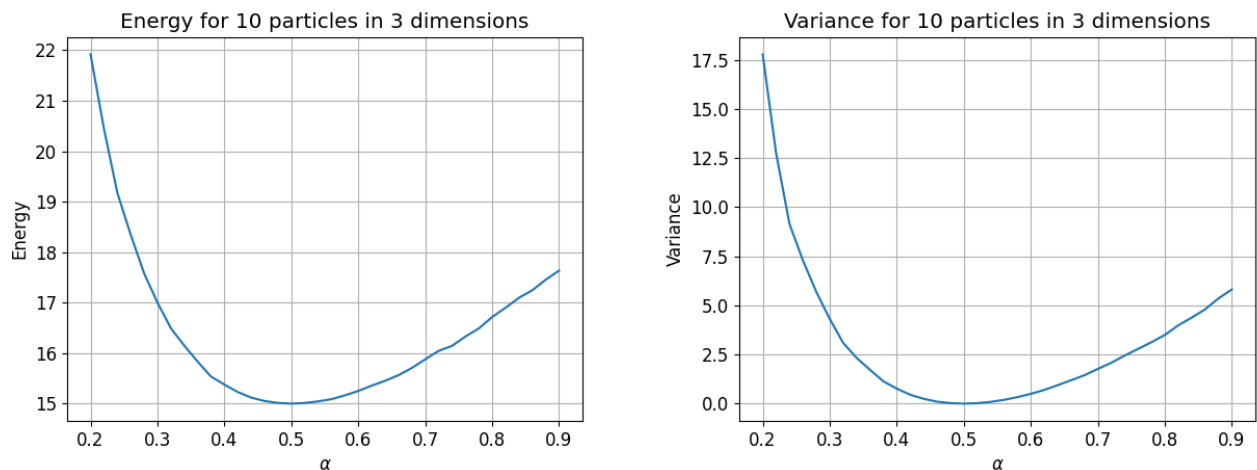


Figure 6. Calculated energy and variance using the Monte Carlo calculations with importance sampling for different values of the parameter α for 10 non-interacting bosons in a spherical trap.

	$n = 1$	$n = 10$	$n = 100$	$n = 500$
Brute Force	29.43 seconds	278.72 seconds	2710.04 seconds	13177.01 seconds
Importance sampling	4.36 seconds	42.01 seconds	409.61 seconds	2021.77 seconds

Table I. CPU time for a Monte Carlo calculation, with both equilibration steps and the main algorithm for $n = 1$, $n = 10$, $n = 100$ and $n = 500$.

In the previous calculations we simply plotted the expectation values of the energies as a function of the parameter α . For large systems this means that we spend equally many Monte Carlo cycles for values of α that are far away from the energy minimum as those that are close to it. To improve upon this we implement a steepest descent algorithm to minimize the energy and obtain the best possible value of α . This algorithm is tested with a weight of $\eta = 0.01$, although this was later changed to $\eta = 0.1/n$, for $n = 1$ in one dimension, $n = 2$ in two dimensions and $n = 3$ in three dimensions. The algorithm is stopped if the approximation for the energy derivative is lower than 10^{-4} . For one particle in one dimension the algorithm reached $\alpha = 0.493966$ after 154 iterations. For two particles in two dimension the algorithm reached $\alpha = 0.4938934$ after 57 iterations. For three particles in three dimension the algorithm reached $\alpha = 0.499258$ after 23 iterations.

In table II we see the results from the previously done calculations, with importance sampling, analysed using the

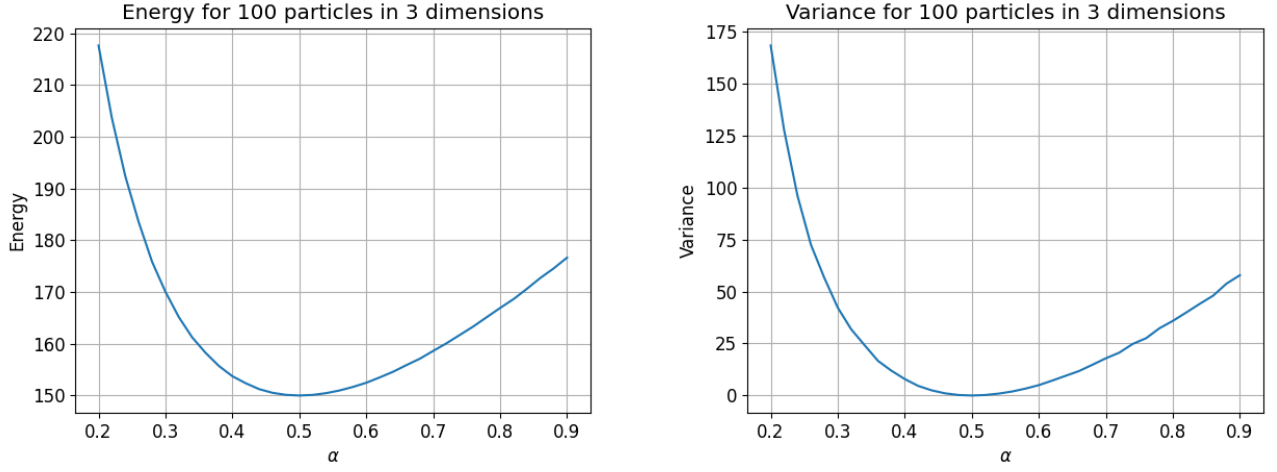


Figure 7. Calculated energy and variance using the Monte Carlo calculations with importance sampling for different values of the parameter α for 100 non-interacting bosons in a spherical trap.

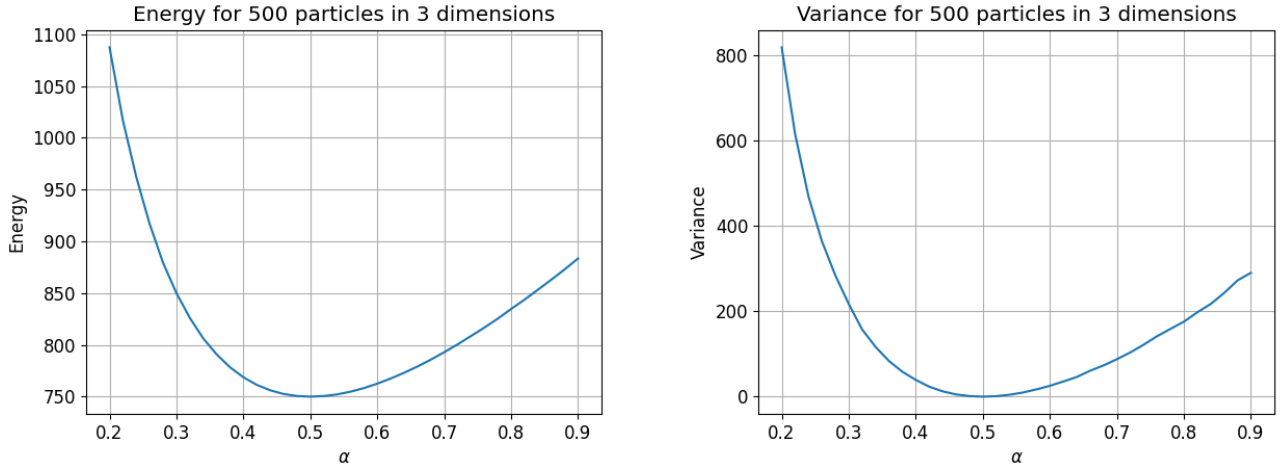


Figure 8. Calculated energy and variance using the Monte Carlo calculations with importance sampling for different values of the parameter α for 500 non-interacting bosons in a spherical trap.

blocking method. We note that the results largely match the analytic results and that the standard deviations are low in all cases. The calculations are done for $n = 1$, $n = 10$, $n = 50$ and $n = 100$ with the values of α found with the gradient descent method. We can see that even with small variations in α the resulting energy is very close to the expected values at the optimal $\alpha = 0.5$. All computations that are shown in table II have started with an initial guess for the variational parameter $\alpha = 0.4$.

Number of particles	Variational parameter α	Ground state energy	Standard deviation
1	0.498959	1.500023	0.000037
10	0.499256	14.999892	0.000052
50	0.495986	75.001869	0.001104
100	0.492543	150.026922	0.002663

Table II. Results from Monte Carlo calculation for non-interacting particles in a spherical trap.

Now that it is clear the calculations give the expected results for the non-interacting case we can move on to parallelize the code using OpenMP. The main idea behind the parallelization is to run the calculations several times averaging over the results.

Then everything is ready for us to implement the repulsive interaction between particles. The trap used so far is converted to a elliptical one, described in eq 29. For convenience we slightly restructure our Metropolis function to move a single particle at a time, rather than a single particle in a single dimension. This is done such that $f(a, |r_i - r_j|)$ can be calculated once rather than three times for each particle, in three dimensions.

V. THE REPUSLIVE INTERACTION

Table III shows the results of our Monte Carlo calculation for an elliptical trap with $\beta = \gamma = 2.82843$ and $a/a_{ho} = 0.0043$ with $n = 10$, $n = 50$ and $n = 100$ interacting particles. Table IV shows the results of our Monte Carlo

Number of particles	Variational parameter α	Ground state energy	standard deviation
10	0.498085	24.450653	0.027901
50	0.493351	129.103313	0.155081
100	0.491609	275.444469	0.01354

Table III. Results from Monte Carlo calculation for interacting particles in a elliptical trap with $\beta = \gamma = 2.82843$ and $a/a_{ho} = 0.0043$.

calculation for an elliptical trap with $\beta = \gamma = 2.82843$ with $n = 10$, $n = 50$ and $n = 100$ non-interacting particles. The

Number of particles	Variational parameter α	Ground state energy	standard deviation
10	0.5	24.14405	0.000605
50	0.5	120.678625	0.027237
100	0.5	241.34062	0.065614

Table IV. Results from Monte Carlo calculation for non-interacting particles in a elliptical trap with $\beta = \gamma = 2.82843$ and $a/a_{ho} = 0.0043$.

calculations are run using 2^{15} Monte Carlo cycles four times and then averaged. The optimization is run with 10^4 Monte Carlo cycles for each value of α . The number of equilibration steps is 10^4 . The number of Monte Carlo cycles is chosen as the systems seem to equilibrate at a rapid pace such that running the calculations for longer wont change the results much. This can also be seen from the low standard deviations when calculating the energies.

Comparing table III and IV we note the increase in energy and decrease in the variational parameter α when we increase the number of particles. The increase in energy seems to match the results found in [1]. The decrease in α matches the results from [2], although in our case the change is rather small.

	Onebody density
Without repulsive interaction, $a = 0$	$0.702154 \phi^2(\mathbf{r}_1)$
With repulsive interaction, $a/a_{ho} = 0.0043$	$0.669945 \phi^2(\mathbf{r}_1)$

Table V. Calculated values for the onebody density for two particles, with and without the repulsive interaction.

In table V we see the results of the onebody calculations run for two particles with and without the repulsive interaction. We see that there is small decrease in the onebody density as a result of introducing the repulsive interaction. Yet it worth mentioning that running this calculation repeatedly gives slightly different results, however the overall result that the onebody density is higher without the repulse interaction stays prominent, although how much it changes from run to run varies. A statistical analysis of the onebody density should therefore be done, although I do not have time for it at the current moment.

VI. REVIEW OF THE PROJECT

Let us start with the positives. I feel like I have learned quite a few things working my way through this project. Programming in C++ is not something I have alot of experience in, and the class structures in this project was a good way of getting some hands on experience in how to write in this language. Furthermore it was a good introduction on how to think about coding projects, and how to plan ahead in your coding structure. I have learned alot through the mistakes that I made along the way and gotten a viewpoint of what aspects of the program should be prioritized when starting to write.

Then to the negatives, of which I only really have one. The pacing of this project seemed very off. It was in all of ways very slow to start with until it all suddenly exploded towards the end. Some clearer goals of where you should be in the project at certain times would definitely be appreciated, and I believe it would be a good tool for knowing which aspects that need to be shown in the lectures.

-
- [1] JL DuBois and HR Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Physical Review A*, 63(2):023602, 2001.
 - [2] JK Nilsen, Jordi Mur-Petit, Montserrat Guilleumas, M Hjorth-Jensen, and A Polls. Vortices in atomic bose-einstein condensates in the large-gas-parameter region. *Physical Review A*, 71(5):053610, 2005.