University of Oslo

FYS4411

Computational Physics II: Quantum mechanical systems

Project 1 - Variational Monte Carlo of Bosonic systems

Author:
Line G. Pedersen

Author:
Trond-Wiggo Johansen

April, 2018

Contents

Abstract

A numerical study of a system of bosons is presented. We have developed a code that calculates the energy of such a system in one, two or three dimensions and a desired number of particles. There are implemented several algorithms for this purpose. We found that the analytic derivation was faster than the numerical derivation. Another improvement to our code was to use Importance sampling instead of a brute force Metropolis algorithm. With Importance sampling the energy of the system stabilizes much quicker than it's counterpart, so using this method one could in theory use fewer Monte Carlo cycles to achieve the same results.

For better handling of errors we implemented the blocking technique. This we found to decrease the calculation error drastically. For finding the optimal value for our variational parameter, α, we implemented the conjugate descent method. In the non-interacting case, this returned $\alpha = 0.5$ as expected and in the interacting case we found it to be slightly lower and dependent on the number of particles.

1 Introduction

When Bengali physicist Satyendra Nath Bose sent a paper [1] to Albert Einstein in 1924 showing how he had derived Planck's quantum radiation law using only the quantum statistics of light quanta, or photons, without any use of classical physics, Einstein was impressed. The German physicist later extended the work of Bose and the result was a concept of a Bose gas [2], which describes a statistical distribution of identical bosons. This is called the Bose-Einstein condensate, and describes a new phase of matter.

As well as photons, other composite atoms with integer spins are bosons. An example of this is helium-4, which is demonstrated to behave as a super-fluid when cooled to below 2.12 K and thus behave as the expected Bose-Einstein condensate.

Over the years Bose-Einstein condensation has been demonstrated for several different bosons, including photons and alkali atoms conffined in magnetic traps [3]. The latter has led to an explosion of interest in confined Bose systems.

The problems one face when trying to solve many-body problems analytically are great. We would therefore like to present our numerical studies to the Bose-Einstein condensation gas, based on analytic calculations. Our goal is to use Variational Monte Carlo Methods to find the ground state energy of a trapped, hard sphere Bose gas. The code developed for our studies can be found in it's entirety at Github repository: `https://github.com/wiggoen/FYS4411/tree/master/Project_1`

2 Theory

Much of the following theory is gotten from [4].

We want to study particles trapped in a harmonic potential which can be either spherical (S) or elliptical (E). In three dimensions, this potential is given as

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

where ω_{ho}^2 defines the trap potential strength. With the elliptic trap potential, $V_{ext}(\mathbf{r})$, the trap frequency in the perpendicular (xy-plane) is $\omega_{ho} = \omega_\perp$ and the frequency in the z-direction is ω_z. In the trap given by (1), the mean square vibrational amplitude of a single boson at $T = 0$ K is $\langle x^2 \rangle = (\hbar/2m\omega_{ho})$ so that $a_{ho} \equiv (\hbar/m\omega_{ho})^{1/2}$ defines the characteristic length of the trap. The ratio of the frequencies can be denoted by $\gamma = \omega_z/\omega_\perp$, which leads to a ratio of the trap lengths $(a_\perp/a_z) = (\omega_z/\omega_\perp)^{1/2} = \sqrt{\gamma}$.

When we have more than one boson, we must also introduce an internal potential acting on two bosons in a system. We assume that the only force acting on them is coming from the hard-core diameter. If this is a, then the internal potential is simply

$$V_{int}(\mathbf{r}_i, \mathbf{r}_j) = \begin{cases} \infty & \text{for } |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & \text{for } |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{2}$$

Clearly the Hamiltonian of the system must include both the internal and the external potential. We define the two-body Hamiltonian, namely the one acting between two particles, to be

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \tag{3}$$

We define a trial wave function for the ground state, which means that we will take a calculated guess on what the wave function might be and use it for calculations. For N atoms, our trial wave function is defined as

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, \alpha, \beta) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i<j} f(a, \mathbf{r}_i, \mathbf{r}_j) \tag{4}$$

where α and β are variational parameters which describe the spread of the bosons in the trap depending on the hard-core diameter a. The single particle function $g(\alpha, \beta, \mathbf{r}_i)$ is defined as

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

while the pair Jastrow function $f(a, \mathbf{r}_i, \mathbf{r}_j)$ is

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

Note that the Jastrow function don't depend on any variational parameters. This can lead to inaccuracies.

2.1 Local Energy

We can define a quantity known as the local energy by

$$E_L(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} H \Psi_T(\mathbf{r}), \tag{7}$$

which describes the energy of our system.

With a complicated wave function, the local energy can get quite tricky to calculate and in our case the calculation is not completely trivial. We will therefore begin with only the harmonic potential, setting $a = 0$ and $\beta = 1$.

First, we derive the local energy for a system with only one boson. In this case, $f(a, \mathbf{r}_i, \mathbf{r}_j)$ is simply just equal to 1 and we are only left with the wave function to be

$$\Psi_T(x_i) = \prod_i g(\alpha, 1, x_i) = \prod_i e^{-\alpha x_i^2}$$

as we in one dimension put $y = z = 0$. The one-dimensional Hamiltonian of our system is

$$H = \sum_k^N \left(\frac{-\hbar^2}{2m} \frac{\partial^2}{\partial x_k^2} + \frac{1}{2} m \omega_{ho}^2 x_k^2 \right)$$

where N is the number of particles. We recognize this as the normal harmonic oscillator potential and the energy of the system is thus

$$\frac{1}{\Psi_T(x_i)} H \Psi_T(x_i) = \frac{1}{2} N \hbar \omega_{ho}$$

More thorough calculations for the non-interacting case can be found in the Appendix, in Section 9.1. Section 9.1 shows the calculations of the local energy in 1, 2

and 3 dimensions and in addition the calculation of the quantum force (drift force). For the interacting case, the calculation of the energy can be found in Section 9.2.4.

The way a particle is most likely to move is defined by the wave function. We define a force term which describes which direction the particle is being "pushed", or most likely to move. The quantum force is defined as

$$F = \frac{2\nabla\Psi_T}{\Psi_T}. \tag{8}$$

The calculation of the quantum force for the non-interacting case can be found in Section 9.1.5, and the interacting case can be found in Section 9.2.5.

When studying the three-dimensional case, we arrive at the hardest analytical parts. The derivative of the trial wave function is given as

$$\frac{1}{\Psi_T(\mathbf{r})}\sum_i^N \nabla_i^2\Psi_T(\mathbf{r}),$$

where Ψ_T is the trial wavefunction as described above, which can be rewritten as

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

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

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

with $u(r_{ij}) = \ln f(r_{ij})$. We can also write the single-particle wave function as

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

Inserting this, the wave function becomes

$$\Psi_T(\mathbf{r}) = \prod_i \phi(\mathbf{r_i})\exp\left(\sum_{i<j} u(r_{ij})\right) \tag{9}$$

This is the expression for the wave function that we are going to use from now. We see that to calculate the Hamiltonian, needed to calculate the local energy, we need an expression for the double derivative of the trial wave function.

Inserting our expressions above we can find the first derivative of the wavefunction to be

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i\neq k} \phi(\mathbf{r}_i)\right] \exp\left(\sum_{i<j} u(r_{ij})\right) + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij})\right) \sum_{j\neq k} \nabla_k u(r_{kj}) \tag{10}$$

The calculation of the first derivative can be found in Section 9.2.2. We also need the second derivative, which can be found as

$$\begin{aligned}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2 \Psi_T(\mathbf{r}) = \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + \frac{2\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left(\sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj})\right) \\ + \sum_{ij\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) + \sum_{j\neq k} \left(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj})\right)\end{aligned} \tag{11}$$

The calculation of the second derivative can be found in Section 9.2.3. These calculations were done for the spherical harmonic oscillator potential. If, however, we want to study the elliptical potential instead, we use the slightly different Hamiltonian. If we introduce a scaling of the system by lengths in units of a_{ho}, such that $r \to r/a_{ho}$ and energy in units of $\hbar\omega_{ho}$, we can show that the Hamiltonian can be rewritten as

$$H = \sum_{i=1}^{N} \frac{1}{2}\left(-\nabla_i^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2\right) + \sum_{i<j} V_{int}(\mathbf{r}_i, \mathbf{r}_j). \tag{12}$$

We will be using $\beta = \gamma = 2.82843$ and $a/a_{ho} = 0.0043$ when using the elliptical potential. See section 9.2.1 for the rewrite of the Hamiltonian.

We want to compare the numerical differentiation of the wavefunction to the analytical implementation. For the numerical differentiation of the second derivative we use the central difference approximation given by

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

where the dominating errors goes as $\mathcal{O}(h^2)$ [5].

2.2 Blocking

In statistics we often talk about events and whether they are correlated or not. Throwing a die multiple times is an example of a set of event that are independent

of each other, as the probability of the next throw giving a certain number of eyes does not change regardless of previous results. In physics however, many statistical properties are indeed correlated. For example, the energy of a system at time $t + \Delta t$ will be affected by the energy at time t assuming that Δt is small. Mathematically, we can talk about correlation when defining the standard deviation of a system. If the samples are uncorrelated, the standard deviation can be calculated as

$$\sigma = \sqrt{\frac{1}{n}(\langle M^2 \rangle - \langle M \rangle^2)}$$

for n samples. If on the other hand our samples are correlated, this should be rewritten as

$$\sigma = \sqrt{\frac{1 + 2\tau/\Delta t}{n}(\langle M^2 \rangle - \langle M \rangle^2)}$$

where τ is the correlation time, or the time between a sample and the next uncorrelated sample, and Δt is the time between each sample.

In the case where Δt is much bigger than τ, we can use equation (2.2) to calculate the standard deviation. However, it is much more common that this is not the case, and that Δt is indeed smaller than the correlation time. When using data blocking, we can divide our sample into different blocks and calculate the mean of each block and use this to calculate the total mean and the variance of our samples [6]. For this to work, each block must be sufficiently large so that the mean of each block is uncorrelated with all the other blocks. A good choice for the block size would be to use the correlation time, τ.

The problem with blocking often comes from the simple fact that τ is unknown or difficult to compute. In theory it could be easy to compute, but it might take up a lot of CPU time unnecessarily when dealing with large data sets. To avoid this problem we might do an alternative solution and instead plot the standard deviation as a function of blocksize. When looking at this plot, we will see that the standard deviation will stabilize and reach a plateau when the block size is big enough.

2.3 Steepest Descent

In Monte Carlo simulations we usually have some variational parameters that might be unknown. The conjugate descent method is a way to determine the best value for one such parameter. In our case, we want to determine the best possible value for the variational parameter α, which is the α that gives the minimum energy. To do this there are different numerical methods to go about, like Steepest descent, conjugate gradient method or simply Newton's method. In our code we have chosen

to use the steepest descent method. The following theory is largely gotten from M. Hjorth-Jensen's lecture notes [7].

In order to find the minimum of a function relying on a parameter α, one needs to find the derivative of that function with respect to α. We define the derivative of the energy with respect to α as

$$\overline{E}_\alpha = \frac{d\langle E_L[\alpha]\rangle}{d\alpha} \tag{13}$$

and the derivative of the wave function as

$$\overline{\psi}_\alpha = \frac{d\psi[\alpha]}{d\alpha} \tag{14}$$

Using the chain rule and the hermiticity of the Hamiltonain we arrive at the expression for energy to be

$$\overline{E}_\alpha = 2\left(\left\langle \frac{\overline{\psi}_\alpha}{\psi[\alpha]} E_L[\alpha]\right\rangle - \left\langle \frac{\overline{\psi}_\alpha}{\psi[\alpha]}\right\rangle \langle E_L[\alpha]\rangle\right) \tag{15}$$

In other words, we need to numerically or analytically find the expectation values for the wavefunction derivated times local energy together as well as the expectation value for the wavefunction derivated and the expectation value for local energy respectively.

2.4 Onebody densities

An object fundamental to quantum mechanics is the one-body density. It provides us with a possibility to optimize our wavefunction, calculation of excitation energies and enables us to visualize the wave function [8]. The one-body density matrix for a normalized wave function Ψ is defined as

$$\rho(\mathbf{r}_1) = \int |\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)|^2 d\mathbf{r}_2 \cdots d\mathbf{r}_N$$

where the integration is over all coordinates except one. Physically this tells us the probability of finding a particle in the volume element $d\mathbf{r}_1$ in the position $\mathbf{r}_1$ [9]. Since the bosons we are looking at have a physical size, they can not be in the same position and can not overlap with each other. The physical minimum distance between two bosons is dependent on the hard-core diameter a, and thus gives a physical minimum distance of $a/2$ without interaction.

We can count the probability in different dimensions. In one dimension we can count how many particles is inside the increasing distances dx_i (where $i = i \cdot r_{\text{step}}$) from the

origin. In two dimensions we can count how many particles we find in an increasing square area $d\mathbf{A_i}$ from the origin. And in three dimensions we can count how many particles we find in increasing spheres with radii $d\mathbf{r}_i$ from the origin. The counts are done in the increasing intervals (spheres) where $(j-1)\cdot r_{step} \leq |\mathbf{r}_i| < j \cdot r_{step}$ and j is the bin number of the histogram. Figure 1 shows an illustration of the 3 dimensional spheres with increasing radii.

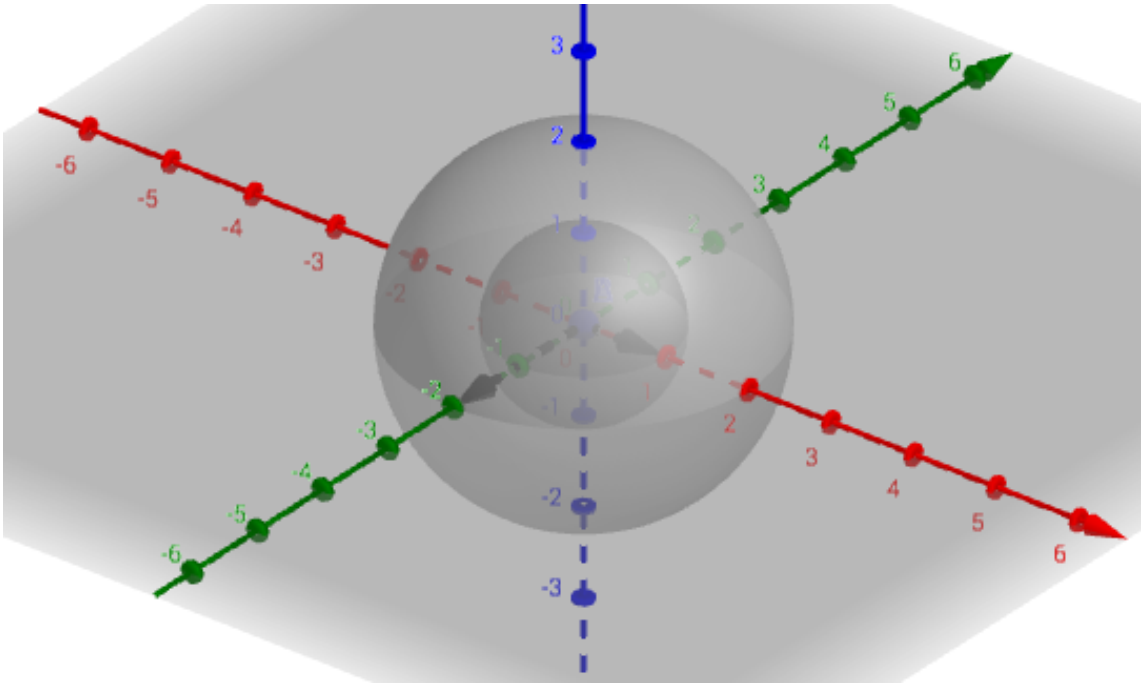

Figure 1: Illustration of spheres with increasing radii starting from the origin.

3 Methods and algorithms

3.1 Brute force Metropolis algorithm

When we have a physical system that is hard to solve analytically, for example systems with many particles, it might be more efficient to use Monte Carlo methods instead. In our code, we have used what is called brute force Metropolis, which simply consists of picking a particle and using a random number generator to give it a new position. We can then compare the energy of the system before and after moving the particle and use yet a new random number to decide whether the new position is kept or not.

Algorithm 1 The brute force Monte Carlo algorithm.

1: Create an $N \times$ dim matrix where N is the number of particles and dim is the number of dimensions.
2: Loop over all the particles
3: Draw a random new position for a particle
4: Calculate the new value of the wave function
5: Calculate the acceptance weight, $w = \Psi^2_{\text{new}}/\Psi^2_{\text{old}}$
6: Pick random uniform number r between 0 and 1
7: If $r \leq w$:
8: Keep new position, else:
9: Go back to old position
10: Repeat from step 3

The brute force Metropolis is the simplest algorithm we implemented in our code. Other algorithms were used for optimization.

3.2 Importance Sampling: The Fokker-Planck and Langevin Equations

One downside of the brute force Metropolis algorithm is that the random walk can happen in every direction. In some cases, the walk is more heavily drawn in a specific direction. If this is the case, we can replace the brute force Metropolis with some other algorithm that can make a trajectory in coordinate space and hence take into account that not all directions are equally likely to move in. In our case, we use an approach which is based on the Fokker-Planck equation and the Langevin equation.

We start by looking at the Fokker-Planck equation [6], which states that

$$\frac{\partial P}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} - F\right)P(x,t) \tag{16}$$

where F is a drift term, D is the diffusion coefficient and $P(x,t)$ is a time-dependent probability density. This equation describes the diffusion process characterized by $P(x,t)$ for one particle in one dimension. This particle will get a new position as defined by the solutions to the Langevin equation

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \tag{17}$$

where η is a random variable. The solution, or the new position, is then

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \tag{18}$$

where ξ is a gaussian random variable and Δt is a chosen time step in the interval from 0.001 to 0.01. We use atomic units and can thus set D to be 0.5, which comes from the 1/2 factor in the kinetic energy operator.

This can be expanded to work in multiple dimensions as well. Then the Fokker-Planck equation yields

$$\frac{\partial P}{\partial t} = \Sigma D \frac{\partial}{\partial x_i}\left(\frac{\partial}{\partial x_i} - F_i\right) P(x,t) \tag{19}$$

where F_i is the i-th component of the drift term caused by an external potential and D is the diffusion coefficient. It can be shown that the drift vector, or the quantum force, is

$$F = \frac{2}{\Psi_T}\nabla\Psi_T \tag{20}$$

It is the quantum force that push the particles towards the configuration space where the trial wave function is large, making the problem more realistic.

Adding the Importance sampling to our code, we use almost the same algorithm as the one explained for brute force Metropolis, in Algorithm 1. The only differences are in point 4 and 5. When we calculate the new value of the wave function, we must also calculate the new drift term, or the quantum force. The acceptance weight is also changed, now to be

$$w = \frac{G(x,y,\Delta t)|\Psi_T(y)|^2_{new}}{G(y,x,\Delta t)|\Psi_T(x)|^2_{old}}$$

where $\Delta t \in [0.001, 0.01]$, x is the old position, y is the new position and G is the Green's function given by

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

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

This leads to the Green's ratio

$$\begin{aligned}
G_{ratio} &= \frac{G(x,y,\Delta t)}{G(y,x,\Delta t)} \\
&= \frac{\frac{1}{(4\pi D\Delta t)^{3N/2}}\exp(-(x-y-D\Delta t F(y))^2/4D\Delta t)}{\frac{1}{(4\pi D\Delta t)^{3N/2}}\exp(-(y-x-D\Delta t F(x))^2/4D\Delta t)} \\
&= \exp(-(x-y-D\Delta t F(y))^2/4D\Delta t - (-(y-x-D\Delta t F(x))^2/4D\Delta t))) \\
&= \exp([-(x-y-D\Delta t F(y))^2 + (y-x-D\Delta t F(x))^2]/4D\Delta t)
\end{aligned}$$

The Green's ratio per particle can be expressed by

$$\begin{aligned} G_{ratio,i} &= \sum_{i}^{N} \frac{G(x_i, y_i, \Delta t)}{G(y_i, x_i, \Delta t)} \\ &= \sum_{i}^{N} \exp([-(x_i - y_i - D\Delta t F(y_i))^2 + (y_i - x_i - D\Delta t F(x_i))^2]/4D\Delta t) \end{aligned}$$

Since the only changes per particle is the new position and new quantum force, the rest of the position matrix and quantum force matrix is still the same. This leads to

$$G_{ratio,i} = (N-1) \exp([-(x_i - y_i - D\Delta t F(y_i))^2 + (y_i - x_i - D\Delta t F(x_i))^2]/4D\Delta t)$$

4 Implementation

All our code is written in a class-based C++ program, except the blocking part which was analyzed in Python. The program can be found in it's entirety at GitHub by user `wiggoen`[1].

The code is given default values for number of particles, dimensions, cycles and some default values for parameters like α and timestep. These can however be overwritten by sending in other values as command line arguments.

The steepest descent method was implemented in our `VariationalMonteCarlo` class. When this function is called, all expectation values are set to zero and calculated again when calling the function the runs the Monte Carlo integration chosen, for each test-value of α. The new value for α is calculated in each loop and the process is repeated.

For the blocking part of our project we used a blocking code especially developed for this purpose by M. Jonsson [10]. Used with permission. This code is designed in a specific way so that the ideal number of blocks per data set is calculated automatically. Thus there was no need to run the blocking multiple times merely to find the optimal number of blocks for each new data set.

[1] `https://github.com/wiggoen/FYS4411/tree/master/Project_1`

5 Code Validation

We used CATCH2[2], a specially designed test framework for unit-testing. With this, we could easily design tests that check if the analytical result is returned for specific initial parameters. This was done for all our expressions where we knew the analytic expression for the energy. For the other values of α, in the non-interacting case, we could make sure that the energy never got smaller than the energy given for $\alpha = 0.5$ for the specific number of particles and dimensions.

Another code validation we did was to test the steepest descent method in the non-interacting case. For this case we knew that the best parameter α was supposed to be 0.5. With the steepest descent method, we do indeed get 0.5 to be the best value.

6 Results

To test the timing of our code, we ran several calculations with the brute force Metropolis algorithm with both the analytic expressions and numeric derivation. With different values of α, we ran the calculations for $N = 1$, 10, 100 and 500 particles for 1, 2 and 3 dimensions respectively. The results are shown in Table 1. A plot of how the energy of a system changes with time (here measured in cycles), is shown in Figure 2.

[2] `https://github.com/catchorg/Catch2`

Table 1: Average run time after 10 tests with 1e6 cycles each and $\alpha = 0.5$. For these calculations we used the brute force Metropolis algorithm with analytic derivation (A) and numeric derivation (N). The numerical derivation turned out to take so much time that we stopped testing after 100 particles.

Dimensions	**Number of particles**	**Avg. time [s], A.**	**Avg. time [s], N.**
1	1	4.23	0.44
1	10	6.07	22.48
1	100	56.50	5061.44
1	500	988.47	-
2	1	3.03	0.63
2	10	5.13	47.40
2	100	72.19	14167.10
2	500	1405.52	-
3	1	3.11	0.81
3	10	5.63	73.29
3	100	89.85	26006.90
3	500	1870.39	-

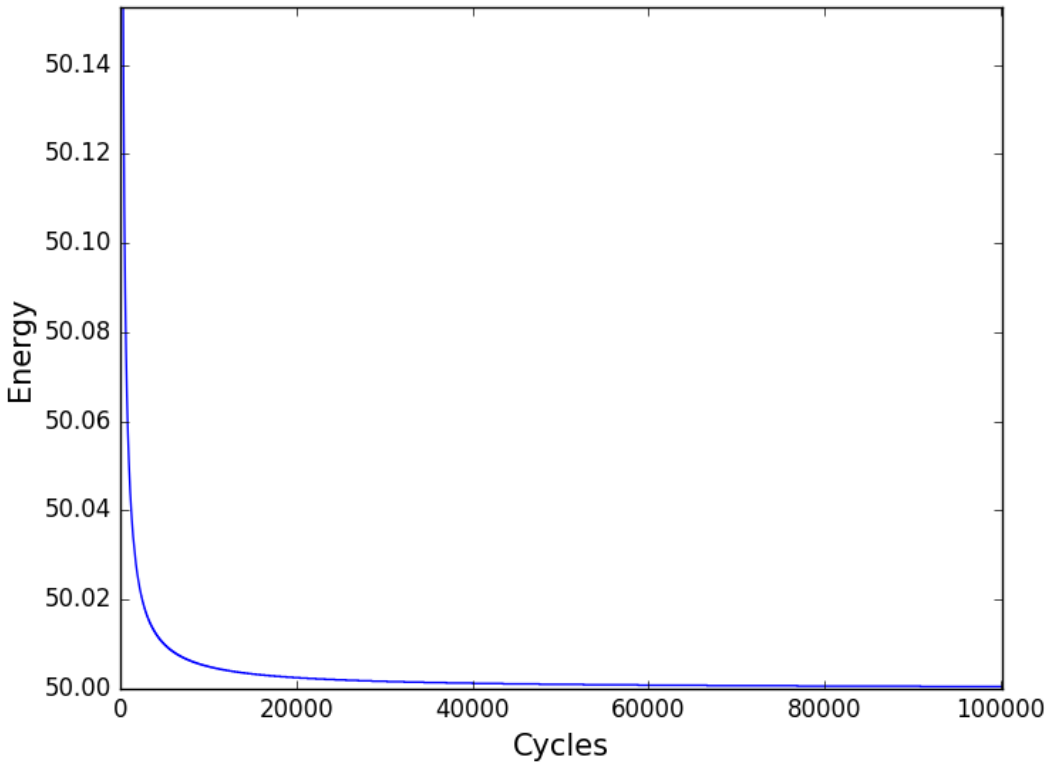

Figure 2: Energy as a function of Monte Carlo cycles. This was done with the brute force Metropolis algorithm and with $\alpha = 0.5$. Illustrated is how the energy slowly moves towards the analytically expected value for the energy.

After implementing the Fokker-Planck and Langevin equations and created the importance sampling part of the code, we wanted to compare the CPU time for the two algorithms. The timing results for importance sampling can be found in Table 2. Comparing this with the brute force Metropolis, we actually find that the code is slightly slower for big N using the Importance sampling. We also tested the algorithm for varying δt from 0.001 to 0.01, but found no dependence in the results from this. The only dependence on the time-step was in the acceptance ratio, which would vary from 0.9 to almost 1 as a function of δt.

Table 2: Average run time after 10 tests with 1e6 cycles each and $\alpha = 0.5$. Here the importance sampling method was applied instead of brute force Metropolis. A denotes analytic derivation and N denotes numeric derivation. Again, the numerical derivation took so much time that we stopped testing after 100 particles.

Dimensions	**Number of particles**	**Avg. time (s), A.**	**Avg. time (s), N.**
1	1	0.368	1.011
1	10	3.62	45.14
1	100	71.39	9866.35
1	500	1092.95	-
2	1	0.437	1.36
2	10	4.52	95.61
2	100	99.77	27011.00
2	500	1680.14	-
3	1	0.523	1.76
3	10	5.54	148.00
3	100	127.55	51999.30
3	500	2254.31	-

By varying our variational parameter α we can find the minimum energy of our system. This was done for 1, 2, and 3 dimensions and for $N = 1$, 10, 100 and 500 particles. The result using brute force Metropolis sampling is plotted in Figure 3.

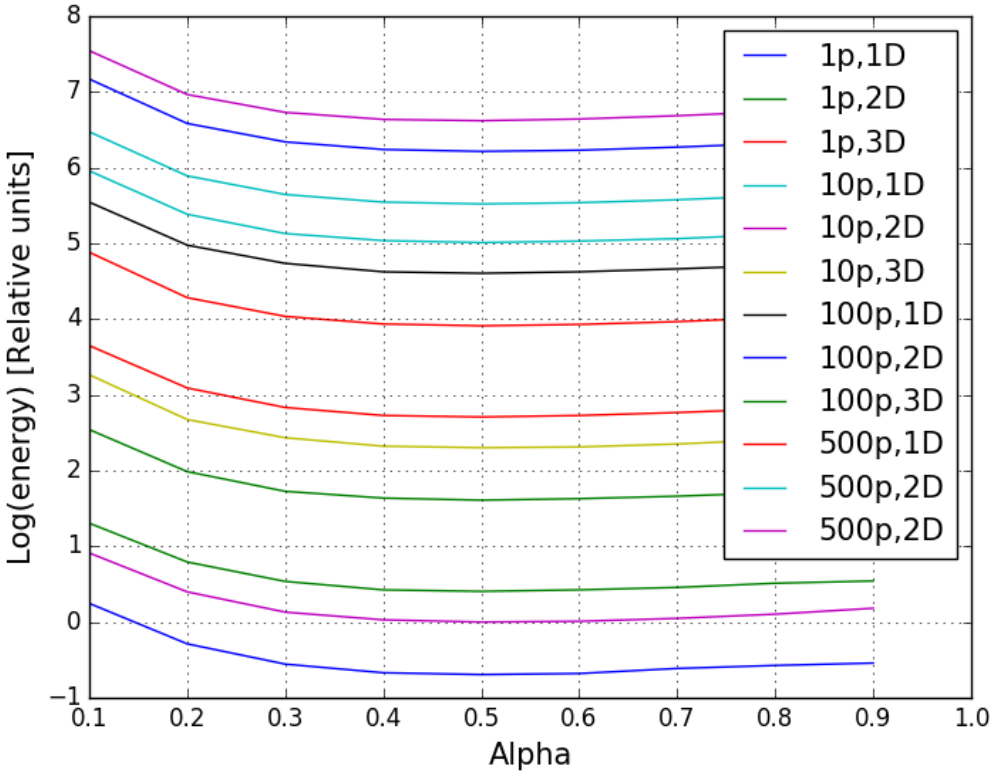

Figure 3: Energy of the system for varying α using brute force sampling. It clearly shows a minimum at $\alpha = 0.5$. (There is a little typo in the plot legend, the purple (last) legend is actually for 500p in 3D, not 500p in 2D.)

As we discovered that $\alpha = 0.5$ was a special value, we wanted to see how this affected the evolution of the energy. This is shown in Figure 4.

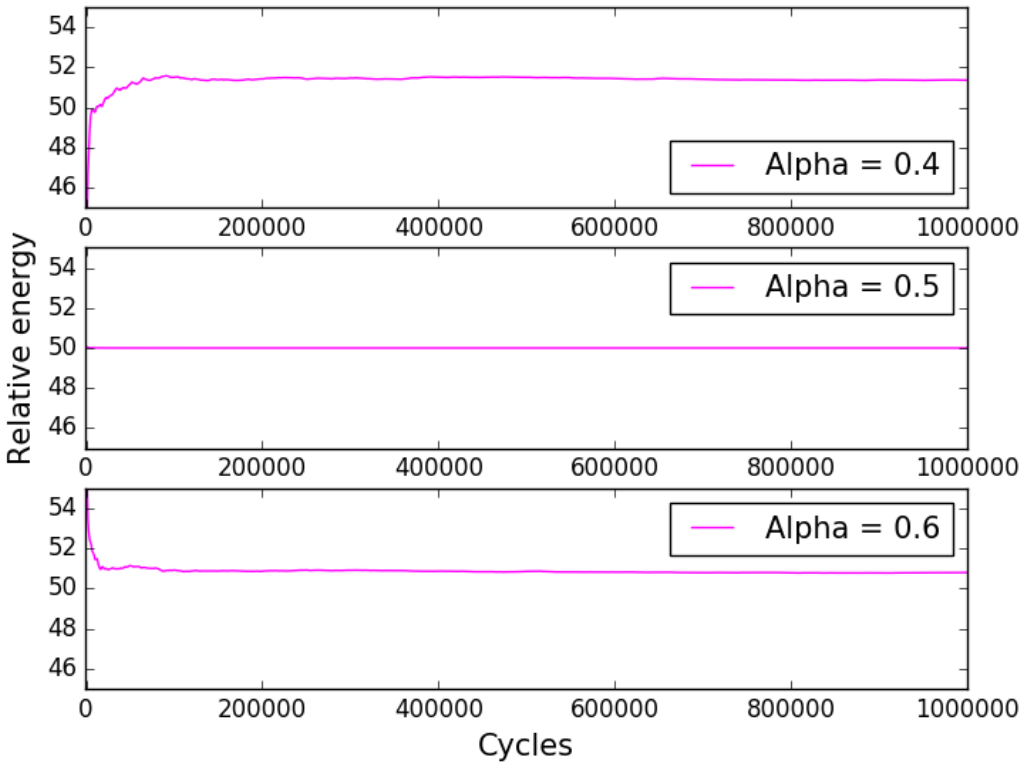

Figure 4: Plotting the energy as a function of time (or really cycles), we see that it is very stable for $\alpha = 0.5$ compared to other close-lying values of α. These calculations were done using the brute force Metropolis algorithm in 1 dimension for 100 particles in the non-interacting case.

Figure 5 shows how the energy varies depending on α using the importance sampling. This plot is close to identical with Figure 3, which is good. The two methods yield the same result, so it is only the timing and acceptance ratio that is different for the two methods.

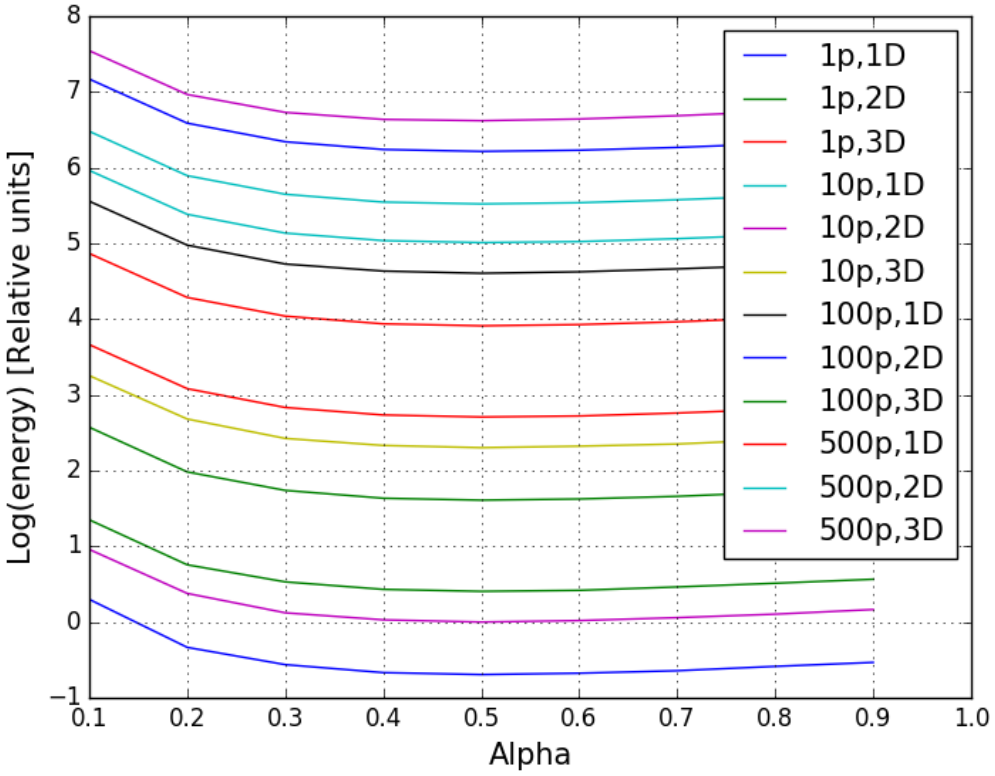

Figure 5: Energy of the system for varying α using importance sampling. It clearly shows a minimum at $\alpha = 0.5$, just as the brute force implementation.

A comparison between the brute force Metropolis (MC) and the importance sampling with Fokker-Planck (FP) is shown in Figure 6.

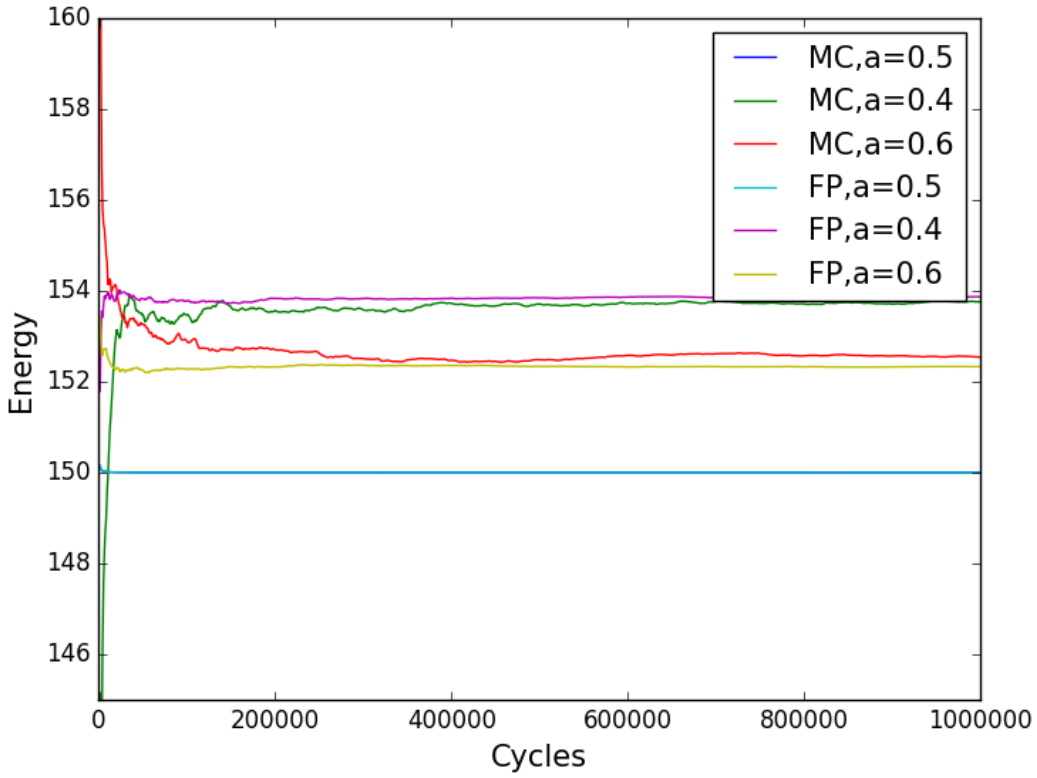

Figure 6: Comparison between the brute force Metropolis (MC) and the importance sampling with Fokker-Planck (FP). This example is for 100 particles in 3 dimensions. This figure illustrates how the two are identical for the $\alpha = 0.5$ case. The difference between the two only become apparent for different values of α, where we see that the Fokker-Planck simulations stabilize much quicker than the MC algorithm and thus needs fewer cycles than the MC algorithm.

To get better statistical error analysis we used the blocking method. Figure 7 shows the standard deviation as a function of α, and Figure 8 shows a comparison between the error analysis with and without the blocking method.

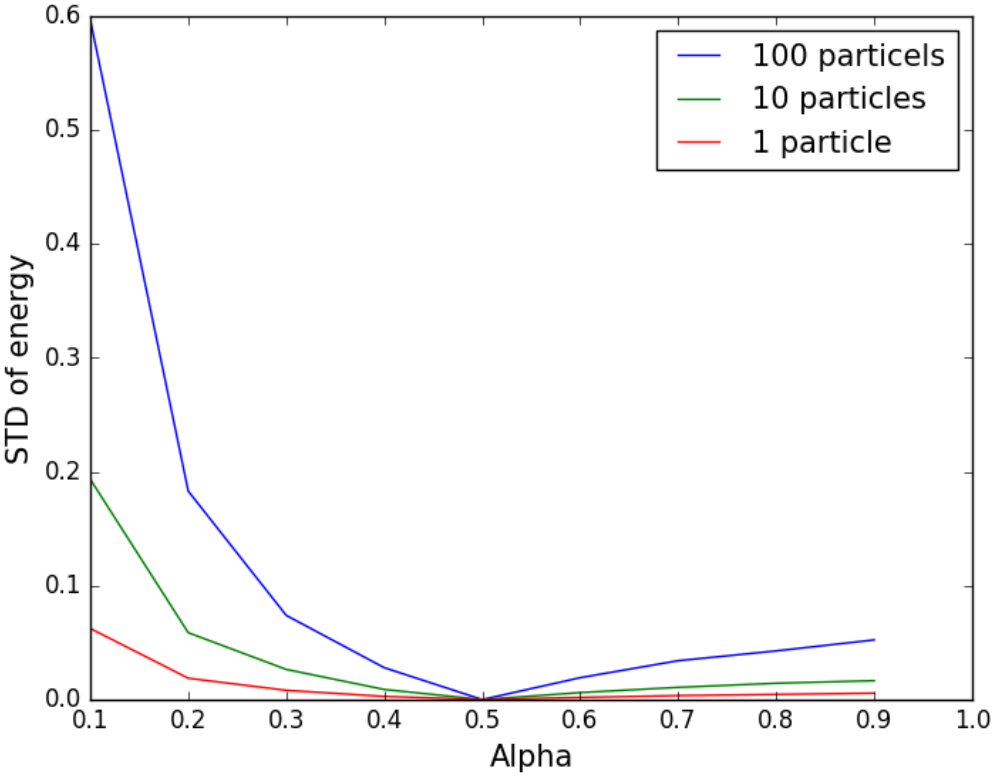

Figure 7: Standard deviation as a function of α, when using the blocking method. All data is gotten from looking at the 3D case using importance sampling. As expected we see that the standard deviation is 0 for $\alpha = 0.5$ and increases as we deviate from this value.

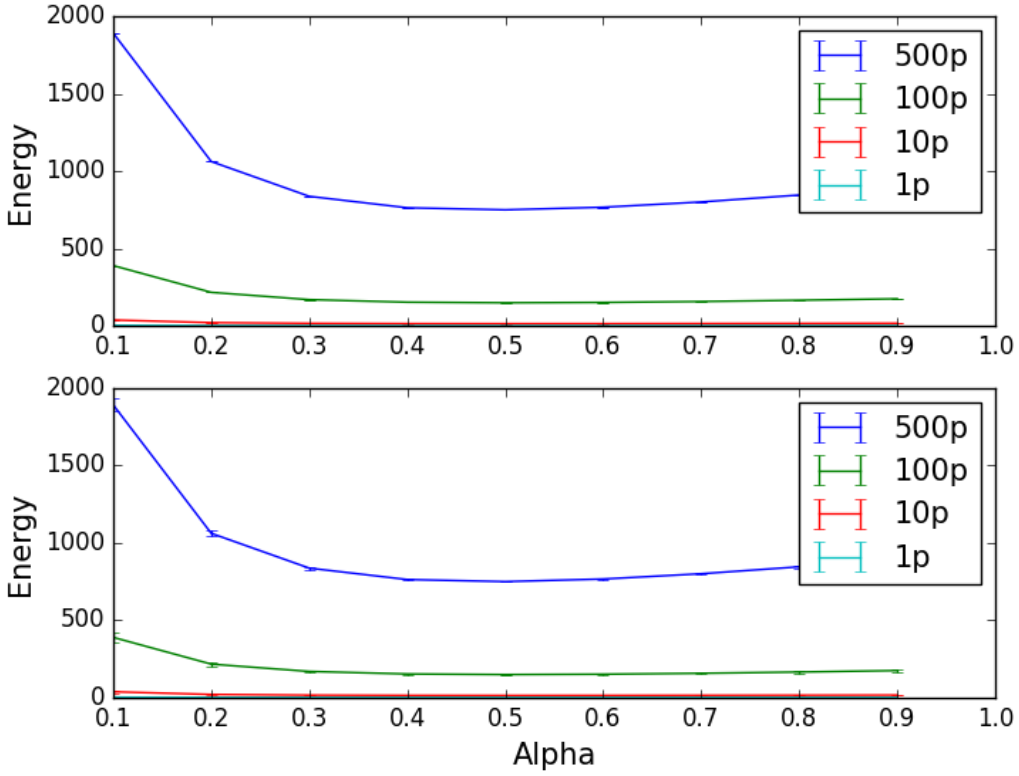

Figure 8: Comparison of standard deviation with (top) and without (bottom) blocking. For values of α far away from 0.5, we see that the standard deviation is of a noteworthy size. However, the deviations found when using the blocking method is orders of magnitude smaller than the ones done without blocking. This figure thus shows that the uncertainty in our calculations reduces significantly when using the blocking method.

Before we went ahead with the steepest descent method on the interacting particles, we started out with a simple calculation of the energies as a function of α. The result is plotted in Figure 9 and shows that, for ten interacting particles, the optimal value should be around 0.5. After this, we ran the steepest descent for ten particles with initial $\alpha = 0.5$ and ended up with the best α to be stable at 0.497. To see how α changes as we increase the number of interacting we also ran the steepest descent for 20 particles, which gave us $\alpha = 0.495$.

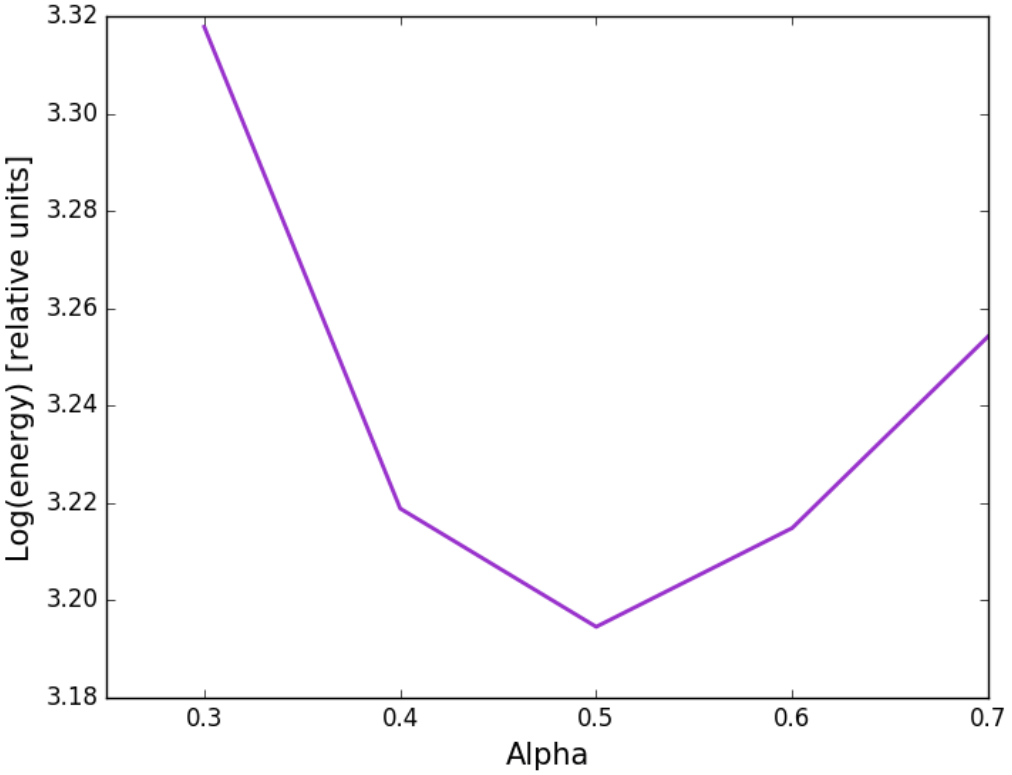

Figure 9: Energy in a system of ten interacting particles for varying α. This figure tells us that the optimal α is around 0.5, so we could start our initial $\alpha = 0.5$ when performing the steepest descent method.

When adding the interaction term, we wanted to see the shape of the energy of the system as a function of time. What we are interested in is how stable the energy is and how new states are accepted. To do this we plotted the relative error as shown in Figure 10.

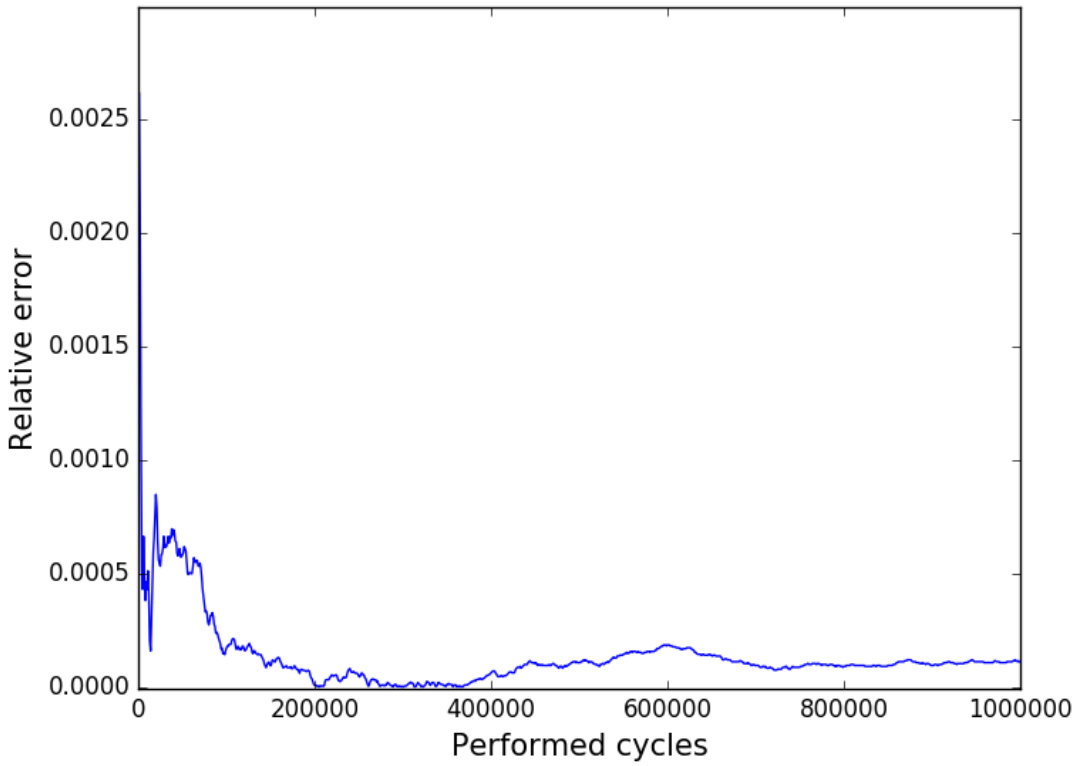

Figure 10: Relative error of the energy of a system of 10 particles as a function of cycles. This has the same shape as the energy as a function of time, but it is scaled down to be zero when the energy in a certain time step is exactly the mean of the energy.

For the interacting case, we only computed in 3 dimensions using the brute force sampling. The energies with their respective α-parameter are shown in Table 3, and the one-body density for 20 particles are shown in Figure 11. The acceptance ratio for the runs was in the interval $0.96 < \mathcal{A}_{bf} < 0.98$.

Table 3: Energy for the interacting bosons using brute force sampling with 1e6 cycles, for varying number of particles and different α. These calculations were done by another computer than the earlier results, and the run time can therefore not be directly compared to the run time without interaction. That said, the run times for the interacting case is higher than the run times for the non-interacting case.

Dimensions	**Number of particles**	$\boldsymbol{\alpha}$	**Energy** ($\hbar\omega_{ho}$)	**Variance**	**Approx. time (s), A.**
3	5	0.5	12.1	0.000453	69
3	10	0.497	24.4	0.00295	679
3	10	0.5	24.4	0.00289	719
3	20	0.495	49.3	0.0185	8997

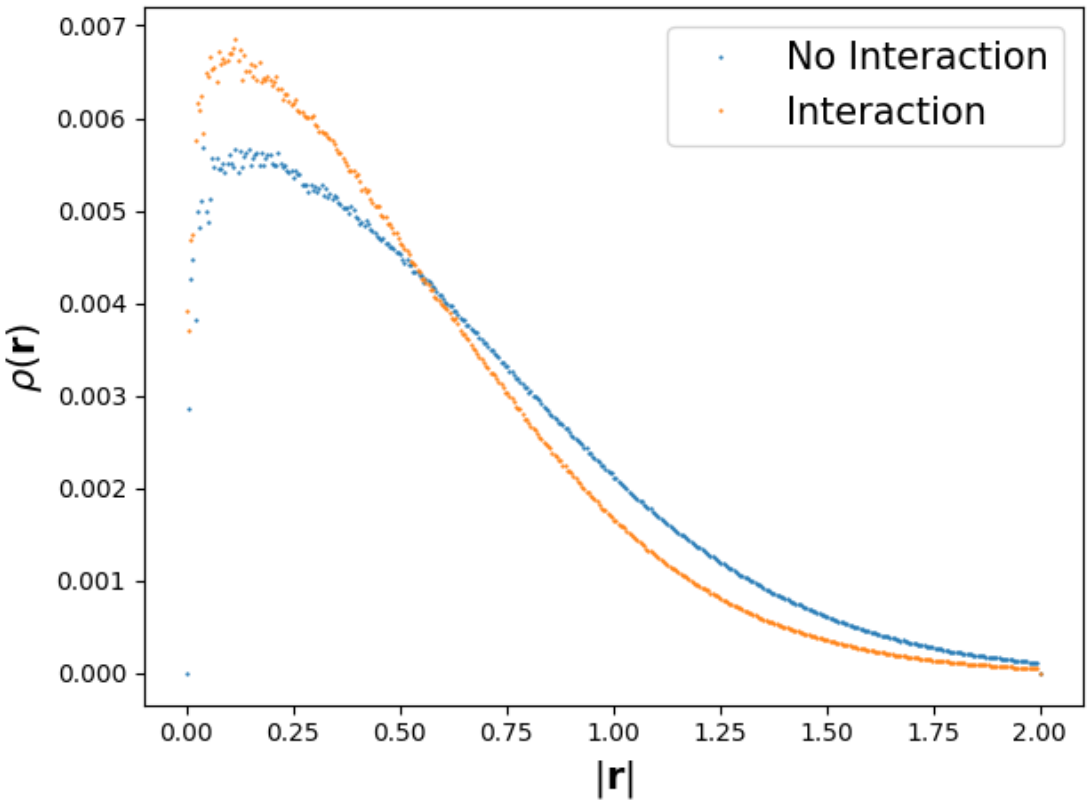

Figure 11: One-body density for both interaction and no interaction. This plot is based on 20 particles and $\alpha = 0.495$.

7 Discussion

Using brute force sampling, the difference in CPU time for numerical and analytic derivation appeared to be huge, as shown in Table 1. The same was apparent using importance sampling, as shown in Table 2. While the analytic work on this project was tedious, the difference between CPU time proved that the analytic work was worth the effort. Adding the analytic expressions made the code run smoothly and we could run for 500 particles in 3D in less than an hour, while the same case with numerical derivation run for days and we in the end decided that it was not worth the time.

A simple test of our code is to compare the results it yields with our analytic results. When using $\alpha = 0.5$ and running brute force sampling without interactions, we have found that the code gives exactly the results that we expect analytically for both the analytic and numeric implementation. This is the case for any number of particles in one, two or three dimensions. The only exceptions was for 1 particle in 1 dimension in the numeric implementation, which gave $E_L = 0.4997187881$ (≈ 0.5) using brute force sampling, and when using importance sampling it gave $E_L = 0.4996833194$ (≈ 0.5). The variance for the analytic runs was zero, and for the numerical derivation the highest variance was $1.06 \cdot 10^{-16}$, which is basically zero in numerical precision. The time it takes to run the different tests is given by Table 4. This was not timed by the program in any scientific way, this is just a stopwatch timing done manually to get an idea of how long the tests took.

Table 4: Manual timing of the code when running the various tests.

Testing method	**Time (approximately)**
Brute force sampling (analytic)	50 sec
Brute force sampling (numeric)	12 min and 50 sec
Importance sampling (analytic)	1 min and 40 sec
Importance sampling (numeric)	23 min and 10 sec

When comparing the importance sampling results with the brute force Metropolis results, we find that they both yield the same energies for any chosen alpha. There is thus not much difference, other than CPU time and acceptance ratio. As shown in Tables 1 and 2, our brute force Metropolis algorithm is slower for few particles, but faster for many particles compared to our algorithm for importance sampling. However, we see that they do not stabilize at the same rate. As illustrated in Figure 6 the code with importance sampling stabilized more rapid than the brute force Metropolis. Importance sampling gave a higher acceptance ratio, $\mathcal{A}_{im}$, in the

interval $0.99 < \mathcal{A}_{im} \leq 1$, and often it was at 1, accepting all steps. The brute force acceptance ratio, $\mathcal{A}_{bf}$, was in the interval $0.96 < \mathcal{A}_{bf} < 1$, which was actually pretty high to be brute force sampling.

We tested for time-steps δt ranging from 0.001 to 0.01 and found no dependence on this from our results. Why the time-step has no effect on our results might only prove that we have chosen δt in a good range. Ideally, we do not want our time-steps to have a big impact on our results, so this might be a good sign. The only difference we found when varying the δt was the varying acceptance ratio, where we found it to vary between 0.9 and 1.

As shown in Figure 8 we see that the uncertainty is reduced significantly when introducing the blocking method to our data. This is a very simple way of reducing the standard deviation without running the program for more cycles. Letting the program run longer, the STD would also have been reduced. However, due to an already high CPU time, using the Blocking method was much more efficient. While talking about the Blocking method, it is also worth to mention that the blocking code we used had implemented an algorithm to find the optimal number of blocks. This was in other words something we did not have to study here.

Figure 7 shows us how the standard derivation (STD) of the energy varies as a function of α. Not surprisingly, this shows us that the STD is zero for $\alpha = 0.5$ and increases when diverging away from this value. This is a result of how the system is more unstable for $\alpha \neq 0.5$ and thus more moves will be accepted, making the system more unstable. Even if the STD increases as we move away from 0.5, it is still a major difference than when we did the same without blocking.

We included a part of our code where we took the interaction between the particles into consideration. Our initial plan was to run the code for $N = 10$, 50 and 100 particles, but the code seemed to be much slower than expected. With 50 particles, one run took about 200000 seconds, or slightly more than two days. This was for one value of α and one million cycles. With this kind of time estimate, doing something like the deepest descent method was impossible for such a high number of particles. Instead, we ended up with having to do the steepest descent for fewer particles. However, we do know that the code works and it is perfectly possible to run the steepest descent for many more particles, as long as the user is willing to wait or have access to a super computer. There might be smarter ways to implement the equations to make it run faster. Another way is to parallelize the code.

To double-check that the steepest descent method worked as we wanted, we tested it for the non-interacting case. This gave us the expected result $\alpha = 0.5$, which corresponds precisely to our analytic calculations and our numerical results, such as those presented in Figures 4 and 5.

As we found the calculation of energy of many interacting particles time consuming, we decided to only do the steepest descend for a limited number of particles. We wanted to run for such a high number of interacting particles, that we could benchmark our results with those obtained in [11] which looked at the result for 500 particles. They found the variational parameter to be between 0.74 and 0.78, depending on what model they used for calculating. This is higher than the non-interacting case, unlike our results which yield a lower α for interacting particles. We believe that the difference comes from having different values for the hard-core diameter a.

From Table 3 we clearly see that the interacting case gives a higher energy than the non-interacting case. For 10 particles in the interacting case we have an energy of 24.4 $\hbar\omega_{ho}$ (with a variance of 0.00294), which for the non-interacting case would have been 15 $\hbar\omega_{ho}$ (with zero variance), and a slightly different α. The energy per particle here is in the interacting case $E/N = 2.44\ \hbar\omega_{ho}$ and the non-interacting case $E/N = 1.5\ \hbar\omega_{ho}$.

The one-body density shown in Figure 11 clearly shows two Gaussian curves. The non-interacting and interacting curves overlap at around $|\mathbf{r}| \approx 0.55$. The Jastrow function $f(r_{ij})$ in the interacting case leads to a narrower Gaussian curve with a little more one-body density closer to zero (the origin).

8 Conclusion

We have designed a code that calculates the energy of a system consisting of bosons. This is done by using different methods, namely numeric and analytic derivation and also looking at importance sampling as well as a brute force Metropolis algorithm.

We have found that the code gives our exact analytical values when studying a case of non-interacting bosons, as expected. In this case we have seen that the energy stabilizes quickly as a function of Monte Carlo cycles for variational parameter $\alpha = 0.5$, which is the α that yields the energy minimum. There is a difference in stabilization time when comparing the brute force Metropolis algorithm with the Importance sampling algorithm, as shown in Figure 6. Importance sampling makes the energy stable much quicker as the random numbers drawn in this algorithm is selected in a specific way and it is thus more likely that a new position is accepted. For the interacting case, stabilization is reached for other values of α, depending on the number of particles in the system.

We have implemented different ways to do the calculations, one where we use numerical derivation and one where we calculate analytically found expressions. Testing

the timing for the two methods, we found that the analytical expression proved to be much quicker, except for the cases with only one particle. It became apparent that the analytical derivation was so superior that it was no reason to use the numerical derivation for other parts of the project.

Instead of only studying our results with common error analysis, we have also used the blocking technique to get a better error analysis. By implementing this we found the errors to be lower than without, which gives us room for better analysis of the data. When using the blocking method we found that the error is reduced by orders of magnitude.

By implementing the steepest descent method we could find the optional value for our variational parameter, α. For the case with non-interacting bosons, we found this value to be 0.5, as expected. When including interactions we find that the value of α decreases as we add more particles. As examples, we get 0.497 and 0.495 for systems of 10 and 20 particles, respectively.

9 Appendix

9.1 Calculations for non-interacting bosons

9.1.1 Local energy in 1 dimension

In the one dimensional case for non-interacting bosons ($a = 0$), we use spherical harmonic traps ($\beta = 1$) and we have

$$\alpha = \frac{1}{2a_{ho}^2} = \frac{1}{2\frac{\hbar}{m\omega_{ho}}} = \frac{m\omega_{ho}}{2\hbar} \qquad \text{since} \quad a_{ho} \equiv \left(\frac{\hbar}{m\omega_{ho}}\right)^{1/2}$$

For non-interacting bosons the correlation wavefunction $f(a, \mathbf{r}_i, \mathbf{r}_j) = 1$ and the inter-boson interaction $V_{int}(\mathbf{r}_i, \mathbf{r}_j) = 0$. The trial wavefunction and the spherical harmonic trap potential can thus be expressed by

$$\Psi_T(x_i) = \prod_i g(0, 1, x_i) = \prod_i e^{-\alpha x_i^2} \qquad V_{ext}(x_k) = \frac{1}{2} m\omega_{ho}^2 x_k^2$$

giving the Hamiltonian

$$H = \sum_k^N \left(\frac{-\hbar^2}{2m}\frac{\partial^2}{\partial x_k^2} + \frac{1}{2}m\omega_{ho}^2 x_k^2\right)$$

When $i = k$, the derivative of the trial wavefunction is

$$\begin{aligned}
\frac{\partial^2}{\partial x_k^2}\Psi_T(x_i) &= \frac{\partial^2}{\partial x_k^2}\left(\prod_i e^{-\alpha x_i^2}\right) \\
&= \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_k}\left(\prod_i e^{-\alpha x_i^2}\right)\right) \\
&= \frac{\partial}{\partial x_k}\prod_i \left(-2\alpha x_k e^{-\alpha x_i^2}\right) \\
&= \prod_i \left(-2\alpha e^{-\alpha x_i^2} + \left(-2\alpha x_k(-2\alpha x_k)e^{-\alpha x_i^2}\right)\right) \\
&= \prod_i \left(4\alpha^2 x_k^2 e^{-\alpha x_i^2} - 2\alpha e^{-\alpha x_i^2}\right) \\
&= \prod_i (4\alpha^2 x_k^2 - 2\alpha)e^{-\alpha x_i^2} \\
&= (4\alpha^2 x_k^2 - 2\alpha)\prod_i e^{-\alpha x_i^2}
\end{aligned}$$

Using the Hamiltonian on the trial wavefunction gives us

$$
\begin{aligned}
H\Psi_T(x_i) &= \sum_k^N \left(\frac{-\hbar^2}{2m}\frac{\partial^2}{\partial x_k^2} + \frac{1}{2}m\omega_{ho}^2 x_k^2 \right) \prod_i e^{-\alpha x_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m}\frac{\partial^2}{\partial x_k^2} \left(\prod_i e^{-\alpha x_i^2} \right) + \frac{1}{2}m\omega_{ho}^2 x_k^2 \left(\prod_i e^{-\alpha x_i^2} \right) \right) \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m} \left((4\alpha^2 x_k^2 - 2\alpha) \prod_i e^{-\alpha x_i^2} \right) + \frac{1}{2}m\omega_{ho}^2 x_k^2 \left(\prod_i e^{-\alpha x_i^2} \right) \right) \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m}(4\alpha^2 x_k^2 - 2\alpha) + \frac{1}{2}m\omega_{ho}^2 x_k^2 \right) \prod_i e^{-\alpha x_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m} \left(4\left(\frac{m\omega_{ho}}{2\hbar}\right)^2 x_k^2 - 2\left(\frac{m\omega_{ho}}{2\hbar}\right) \right) + \frac{1}{2}m\omega_{ho}^2 x_k^2 \right) \prod_i e^{-\alpha x_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m} \left(\frac{m^2\omega_{ho}^2}{\hbar^2} x_k^2 - \frac{m\omega_{ho}}{\hbar} \right) + \frac{m\omega_{ho}^2}{2} x_k^2 \right) \prod_i e^{-\alpha x_i^2} \\
&= \sum_k^N \left(-\frac{m\omega_{ho}^2}{2} x_k^2 + \frac{\hbar\omega_{ho}}{2} + \frac{m\omega_{ho}^2}{2} x_k^2 \right) \prod_i e^{-\alpha x_i^2} \\
&= \sum_k^N \frac{\hbar\omega_{ho}}{2} \prod_i e^{-\alpha x_i^2} \\
&= N\frac{\hbar\omega_{ho}}{2} \prod_i e^{-\alpha x_i^2} \\
&= \tfrac{1}{2}N\hbar\omega_{ho} \prod_i e^{-\alpha x_i^2}
\end{aligned}
$$

and the local energy becomes

$$
\begin{aligned}
E_L(x_i) &= \frac{1}{\Psi_T(x_i)} H\Psi_T(x_i) \\
&= \frac{1}{\prod_i e^{-\alpha x_i^2}} \tfrac{1}{2}N\hbar\omega_{ho} \prod_i e^{-\alpha x_i^2} \\
&= \tfrac{1}{2}N\hbar\omega_{ho}
\end{aligned}
$$

9.1.2 Local energy in 2 dimensions

In the two dimensional case for non-interacting bosons we use the same arguments as for the one dimensional case, except that the two dimensions can be expressed by

$$\mathbf{q}_i^2 = x_i^2 + y_i^2$$

The trial wavefunction and the spherical harmonic trap potential can thus be expressed by

$$\Psi_T(\mathbf{q}_i) = \prod_i g(0, 1, \mathbf{q}_i) = \prod_i e^{-\alpha \mathbf{q}_i^2} = \prod_i e^{-\alpha(x_i^2+y_i^2)}$$
$$V_{ext}(\mathbf{q}_k) = \frac{1}{2} m\omega_{ho}^2 \mathbf{q}_k^2 = \frac{1}{2} m\omega_{ho}^2 (x_k^2 + y_k^2)$$

giving the Hamiltonian

$$H = \sum_k^N \left(\frac{-\hbar^2}{2m} \left(\frac{\partial^2}{\partial x_k^2} + \frac{\partial^2}{\partial y_k^2} \right) + \frac{1}{2} m\omega_{ho}^2 (x_k^2 + y_k^2) \right)$$

When $i = k$, the derivative of the trial wavefunction is

$$\begin{aligned}
\frac{\partial^2}{\partial x_k^2} \Psi_T(\mathbf{q}_i) &= \frac{\partial^2}{\partial x_k^2} \left(\prod_i e^{-\alpha(x_i^2+y_i^2)} \right) \\
&= \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_k} \left(\prod_i e^{-\alpha(x_i^2+y_i^2)} \right) \right) \\
&= \frac{\partial}{\partial x_k} \prod_i \left(-2\alpha x_k e^{-\alpha(x_i^2+y_i^2)} \right) \\
&= \prod_i \left(-2\alpha e^{-\alpha x_i^2} + \left(-2\alpha x_k (-2\alpha x_k) e^{-\alpha(x_i^2+y_i^2)} \right) \right) \\
&= \prod_i \left(4\alpha^2 x_k^2 e^{-\alpha(x_i^2+y_i^2)} - 2\alpha e^{-\alpha(x_i^2+y_i^2)} \right) \\
&= (4\alpha^2 x_k^2 - 2\alpha) \prod_i e^{-\alpha(x_i^2+y_i^2)}
\end{aligned}$$

and similarly

$$\begin{aligned}
\frac{\partial^2}{\partial y_k^2} \Psi_T(\mathbf{q}_i) &= \frac{\partial^2}{\partial y_k^2} \left(\prod_i e^{-\alpha(x_i^2+y_i^2)} \right) \\
&= (4\alpha^2 y_k^2 - 2\alpha) \prod_i e^{-\alpha(x_i^2+y_i^2)}
\end{aligned}$$

Together this becomes

$$\begin{aligned}\left(\frac{\partial^2}{\partial x_k^2}+\frac{\partial^2}{\partial y_k^2}\right)\Psi_T(\mathbf{q}_i) &= \frac{\partial^2}{\partial x_k^2}\Psi_T(\mathbf{q}_i)+\frac{\partial^2}{\partial y_k^2}\Psi_T(\mathbf{q}_i)\\ &= (4\alpha^2x_k^2-2\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2)}+(4\alpha^2y_k^2-2\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= (4\alpha^2x_k^2-2\alpha+4\alpha^2y_k^2-2\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= (4\alpha^2(x_k^2+y_k^2)-4\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2)}\end{aligned}$$

Using the Hamiltonian on the trial wavefunction gives us

$$\begin{aligned}H\Psi_T(\mathbf{q}_i) &= \sum_k^N\left(\frac{-\hbar^2}{2m}\left(\frac{\partial^2}{\partial x_k^2}+\frac{\partial^2}{\partial y_k^2}\right)+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= \sum_k^N\left(\frac{-\hbar^2}{2m}\left(\frac{\partial^2}{\partial x_k^2}+\frac{\partial^2}{\partial y_k^2}\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\prod_i e^{-\alpha(x_i^2+y_i^2)}\right)\\ &= \sum_k^N\left(\frac{-\hbar^2}{2m}(4\alpha^2(x_k^2+y_k^2)-4\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2)}+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\prod_i e^{-\alpha(x_i^2+y_i^2)}\right)\\ &= \sum_k^N\left(\frac{-\hbar^2}{2m}(4\alpha^2(x_k^2+y_k^2)-4\alpha)+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= \sum_k^N\left(\frac{-\hbar^2}{2m}\left(4\left(\frac{m\omega_{ho}}{2\hbar}\right)^2(x_k^2+y_k^2)-4\left(\frac{m\omega_{ho}}{2\hbar}\right)\right)+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= \sum_k^N\left(\frac{-\hbar^2}{2m}\left(\frac{m^2\omega_{ho}^2}{\hbar^2}(x_k^2+y_k^2)-2\frac{m\omega_{ho}}{\hbar}\right)+\frac{1}{2}m\omega_{ho}^2(x_k^2+y_k^2)\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= \sum_k^N\left(-\frac{m\omega_{ho}^2}{2}(x_k^2+y_k^2)+\hbar\omega_{ho}+\frac{m\omega_{ho}^2}{2}(x_k^2+y_k^2)\right)\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= \sum_k^N\hbar\omega_{ho}\prod_i e^{-\alpha(x_i^2+y_i^2)}\\ &= N\hbar\omega_{ho}\prod_i e^{-\alpha(x_i^2+y_i^2)}\end{aligned}$$

and the local energy becomes

$$\begin{aligned} E_L(\mathbf{q}_i) &= \frac{1}{\Psi_T(\mathbf{q}_i)} H\Psi_T(\mathbf{q}_i) \\ &= \frac{1}{\prod_i e^{-\alpha(x_i^2+y_i^2)}} N\hbar\omega_{ho} \prod_i e^{-\alpha(x_i^2+y_i^2)} \\ &= N\hbar\omega_{ho} \end{aligned}$$

9.1.3 Local energy in 3 dimensions

In the three dimensional case for non-interacting bosons we use the same arguments as for the one dimensional case, except that the three dimensions can be expressed by

$$\mathbf{r}_i^2 = x_i^2 + y_i^2 + \beta z_i^2 = x_i^2 + y_i^2 + z_i^2$$

since $\beta = 1$. The trial wavefunction and the spherical harmonic trap potential can thus be expressed by

$$\begin{aligned} \Psi_T(\mathbf{r}_i) &= \prod_i g(0, 1, \mathbf{r}_i) = \prod_i e^{-\alpha \mathbf{r}_i^2} = \prod_i e^{-\alpha(x_i^2+y_i^2+z_i^2)} \\ V_{ext}(\mathbf{r}_k) &= \frac{1}{2} m\omega_{ho}^2 \mathbf{r}_k^2 = \frac{1}{2} m\omega_{ho}^2 (x_k^2 + y_k^2 + z_k^2) \end{aligned}$$

giving the Hamiltonian

$$H = \sum_k^N \left(\frac{-\hbar^2}{2m} \nabla_k^2 + \frac{1}{2} m\omega_{ho}^2 \mathbf{r}_k^2 \right)$$

where

$$\nabla_k^2 = \frac{\partial^2}{\partial x_k^2} + \frac{\partial^2}{\partial y_k^2} + \frac{\partial^2}{\partial z_k^2}$$

When $i = k$, the derivative of the trial wavefunction is

$$\begin{aligned}
\frac{\partial^2}{\partial x_k^2}\Psi_T(\mathbf{r}_i) &= \frac{\partial^2}{\partial x_k^2}\left(\prod_i e^{-\alpha(x_i^2+y_i^2+z_i^2)}\right) \\
&= \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_k}\left(\prod_i e^{-\alpha(x_i^2+y_i^2+z_i^2)}\right)\right) \\
&= \frac{\partial}{\partial x_k}\prod_i\left(-2\alpha x_k e^{-\alpha(x_i^2+y_i^2+z_i^2)}\right) \\
&= \prod_i\left(-2\alpha e^{-\alpha x_k^2} + \left(-2\alpha x_k(-2\alpha x_k)e^{-\alpha(x_i^2+y_i^2+z_i^2)}\right)\right) \\
&= \prod_i\left(4\alpha^2 x_k^2 e^{-\alpha(x_i^2+y_i^2+z_i^2)} - 2\alpha e^{-\alpha(x_i^2+y_i^2+z_i^2)}\right) \\
&= (4\alpha^2 x_k^2 - 2\alpha)\prod_i e^{-\alpha(x_i^2+y_i^2+z_i^2)} \\
&= (4\alpha^2 x_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2}
\end{aligned}$$

and similarly

$$\begin{aligned}
\frac{\partial^2}{\partial y_k^2}\Psi_T(\mathbf{r}_i) &= (4\alpha^2 y_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2} \\
\frac{\partial^2}{\partial z_k^2}\Psi_T(\mathbf{r}_i) &= (4\alpha^2 z_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2}
\end{aligned}$$

Together this becomes

$$\begin{aligned}
\nabla_k^2\Psi_T(\mathbf{r}_i) &= \nabla_k^2\left(\prod_i e^{-\alpha\mathbf{r}_i^2}\right) \\
&= \left(\frac{\partial^2}{\partial x_k^2} + \frac{\partial^2}{\partial y_k^2} + \frac{\partial^2}{\partial z_k^2}\right)\prod_i e^{-\alpha\mathbf{r}_i^2} \\
&= \frac{\partial^2}{\partial x_k^2}\left(\prod_i e^{-\alpha\mathbf{r}_i^2}\right) + \frac{\partial^2}{\partial y_k^2}\left(\prod_i e^{-\alpha\mathbf{r}_i^2}\right) + \frac{\partial^2}{\partial z_k^2}\left(\prod_i e^{-\alpha\mathbf{r}_i^2}\right) \\
&= (4\alpha^2 x_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2} + (4\alpha^2 y_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2} + (4\alpha^2 z_k^2 - 2\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2} \\
&= (4\alpha^2(x_k^2 + y_k^2 + z_k^2) - 6\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2} \\
&= (4\alpha^2\mathbf{r}_k^2 - 6\alpha)\prod_i e^{-\alpha\mathbf{r}_i^2}
\end{aligned}$$

Using the Hamiltonian on the trial wavefunction gives us

$$
\begin{aligned}
H\Psi_T(\mathbf{r}_i) &= \sum_k^N \left(\frac{-\hbar^2}{2m}\nabla_k^2 + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m}\nabla_k^2 \left(\prod_i e^{-\alpha \mathbf{r}_i^2} \right) + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \left(\prod_i e^{-\alpha \mathbf{r}_i^2} \right) \right) \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m}(4\alpha^2\mathbf{r}_k^2 - 6\alpha) \prod_i e^{-\alpha \mathbf{r}_i^2} + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \left(\prod_i e^{-\alpha \mathbf{r}_i^2} \right) \right) \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m}(4\alpha^2\mathbf{r}_k^2 - 6\alpha) + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m} \left(4\left(\frac{m\omega_{ho}}{2\hbar}\right)^2 \mathbf{r}_k^2 - 6\left(\frac{m\omega_{ho}}{2\hbar}\right) \right) + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \sum_k^N \left(\frac{-\hbar^2}{2m} \left(\frac{m^2\omega_{ho}^2}{\hbar^2}\mathbf{r}_k^2 - 3\frac{m\omega_{ho}}{\hbar} \right) + \frac{1}{2}m\omega_{ho}^2\mathbf{r}_k^2 \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \sum_k^N \left(-\frac{m\omega_{ho}^2}{2}\mathbf{r}_k^2 + \frac{3\hbar\omega_{ho}}{2} + \frac{m\omega_{ho}^2}{2}\mathbf{r}_k^2 \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \sum_k^N \left(\frac{3\hbar\omega_{ho}}{2} \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= N \left(\frac{3\hbar\omega_{ho}}{2} \right) \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \tfrac{3}{2}N\hbar\omega_{ho} \prod_i e^{-\alpha \mathbf{r}_i^2}
\end{aligned}
$$

and the local energy becomes

$$
\begin{aligned}
E_L(\mathbf{r}_i) &= \frac{1}{\Psi_T(\mathbf{r}_i)} H\Psi_T(\mathbf{r}_i) \\
&= \frac{1}{\prod_i e^{-\alpha \mathbf{r}_i^2}} \tfrac{3}{2}N\hbar\omega_{ho} \prod_i e^{-\alpha \mathbf{r}_i^2} \\
&= \tfrac{3}{2}N\hbar\omega_{ho}
\end{aligned}
$$

9.1.4 Local energy in general

We see a trend in the local energy from sections 9.1.1 - 9.1.3.

$$\begin{aligned} 1D: &\quad E_L(x_i) = \tfrac{1}{2}N\hbar\omega_{ho} \\ 2D: &\quad E_L(\mathbf{q}_i) = N\hbar\omega_{ho} = \tfrac{2}{2}N\hbar\omega_{ho} \\ 3D: &\quad E_L(\mathbf{r}_i) = \tfrac{3}{2}N\hbar\omega_{ho} \end{aligned}$$

This can be written in the general form as

$$E_L(\mathbf{r}) = \tfrac{D}{2}N\hbar\omega_{ho}$$

where D is the dimension and

$$\mathbf{r}^2 = \begin{cases} x_i^2, & 1 \text{ dimension}, D = 1 \\ x_i^2 + y_i^2, & 2 \text{ dimensions}, D = 2 \\ x_i^2 + y_i^2 + z_i^2, & 3 \text{ dimensions}, D = 3 \end{cases}$$

9.1.5 Drift force (quantum force)

The drift force to be used with importance sampling is given by

$$F = \frac{2\nabla_k \Psi_T}{\Psi_T}$$

The trial wavefunction for non-interacting bosons is expressed by

$$\Psi_T(\mathbf{r}_i) = \prod_i g(0, 1, \mathbf{r}_i) = \prod_i e^{-\alpha \mathbf{r}_i^2} = \prod_i e^{-\alpha(x_i^2 + y_i^2 + z_i^2)}$$

which gives the drift force

$$\begin{aligned} F &= \frac{2\nabla_k \Psi_T}{\Psi_T} \\ &= \frac{2\nabla_k \prod_i e^{-\alpha \mathbf{r}_i^2}}{\Psi_T} \\ &= \frac{2\prod_i e^{-\alpha \mathbf{r}_i^2} \nabla_k(-\alpha \mathbf{r}_k^2)}{\Psi_T} \\ &= \frac{2\Psi_T(-2\alpha \mathbf{r}_k)}{\Psi_T} \\ &= -4\alpha \mathbf{r}_k \end{aligned}$$

9.2 Calculations for interacting bosons

9.2.1 Rewrite of the Hamiltonian in 3 dimensions

In the three dimensional case for interacting bosons, we use a pairwise repulsive potential given by

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

The three dimensions can be expressed by

$$\mathbf{r}_i^2 = x_i^2 + y_i^2 + \beta z_i^2$$

with a trial wavefunction and an elliptic trap potential

$$\Psi_T(\mathbf{r}_i) = \prod_i \phi(\mathbf{r}_i) f(r_{ij}) = \exp(-\alpha(x_i^2 + y_i^2 + \beta z_i^2)) \exp\left(\sum_{i<j} u(r_{ij})\right)$$
$$V_{ext}(\mathbf{r}_i) = \frac{1}{2} m[\omega_{ho}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2]$$

The original Hamiltonian can be expressed by

$$\begin{aligned} H &= \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + V_{ext}(\mathbf{r}_i)\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \\ &= \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2} m[\omega_{ho}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \\ &= \sum_i^N \frac{1}{2}\left(\frac{-\hbar^2}{m}\nabla_i^2 + m[\omega_{ho}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \end{aligned}$$

If we introduce lengths in units of a_{ho}, $r' = r/a_{ho}$ and energy of units $\hbar\omega_{ho}$ we can rewrite the Hamiltonian.

The scaled length

$$r' = r/a_{ho} \qquad \Longrightarrow \qquad r = a_{ho} r'$$

leads to the Laplacian

$$\nabla_i^{'2} = a_{ho}^2 \nabla_i^2 \qquad \Longrightarrow \qquad \nabla_i^2 = \frac{1}{a_{ho}^2}\nabla_i^{'2}$$

which, when put into the Hamiltonian gives

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

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

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

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

$$H = \sum_i^N \frac{1}{2}\left(-\hbar\omega_{ho}\nabla_i^{'2} + \hbar\omega_{ho}[x_i^{'2} + y_i^{'2} + \gamma^2 z_i^{'2}]\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j)$$

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

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

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

where $\gamma = \omega_z/\omega_\perp$ and $a' = a/a_{ho} = 0.0043$.

By renaming from the marked variables $'$ to normal notation, we get

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

with $a = 0.0043$.

9.2.2 The first derivative of the wavefunction

The substitution $r_{kj} = |\mathbf{r}_k - \mathbf{r}_j|$ makes us want to write the gradient ∇_k in another way. We can write the gradient as

$$\nabla_k = \nabla_k \frac{\partial r_{kj}}{\partial r_{kj}} = \nabla_k r_{kj} \frac{\partial}{\partial r_{kj}} = \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \frac{\partial}{\partial r_{kj}} = \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{\partial}{\partial r_{kj}} \tag{21}$$

where $\mathbf{r}_{kj} = \mathbf{r}_k - \mathbf{r}_j$.

By inserting the wavefunction given by (9), the first derivative of the wavefunction for particle k is

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

If we let the derivative in the last term affect the sum and write it out, we see that all terms without r_k becomes zero. An effect of this is that only terms where $i = k$ are left and as we started with only $i < j$, we find that $i = k$ can never be equal to j. We are thus left with the expression for the first derivative to be

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j \neq k} \nabla_k u(r_{kj})$$

9.2.3 The second derivative of the wavefunction

The second derivative of the wavefunction can be found by

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

Starting with the first term, we find

$$
\begin{aligned}
&\nabla_k \left[\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \right] \\
&= \left[\nabla_k^2 \phi(\mathbf{r}_k) \right] \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) + \nabla_k \phi(\mathbf{r}_k) \left[\nabla_k \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \\
&\quad + \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \left[\nabla_k \exp\left(\sum_{i<j} u(r_{ij}) \right) \right] \\
&= \nabla_k^2 \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \\
&\quad + \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \\
&= \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \left(\nabla_k^2 \phi(\mathbf{r}_k) + \nabla_k \phi(\mathbf{r}_k) \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \right)
\end{aligned}
$$

Similarly, we calculate the second term as

$$
\begin{aligned}
&\nabla_k \left[\prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k u(r_{kj}) \right] \\
&= \left[\nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k u(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \left[\nabla_k \exp\left(\sum_{i<j} u(r_{ij}) \right) \right] \sum_{j\neq k} \nabla_k u(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \left[\nabla_k \left[\sum_{j\neq k} \nabla_k u(r_{kj}) \right] \right] \\
&= \nabla_k \phi(\mathbf{r}_k) \prod_{i\neq k} \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k u(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{i\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)}{r_{ki}} u'(r_{ik}) \sum_{j\neq k} \nabla_k u(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k^2 u(r_{kj}) \\
&= \nabla_k \phi(\mathbf{r}_k) \prod_{i\neq k} \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{i\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)}{r_{ki}} u'(r_{ik}) \sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k^2 u(r_{kj})
\end{aligned}
$$

For $\nabla_k^2 u(r_{kj})$, we have

$$
\begin{aligned}
\nabla_k^2 u &= \nabla_k \cdot \nabla_k u && u = u(r_{kj}) \\
&= \nabla_k \cdot \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{\partial u}{\partial r_{kj}} && \nabla_k u = \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{\partial u}{\partial r_{kj}} \\
&= \sum_d \frac{\partial}{\partial x_k^{(d)}} \left(\frac{x_{kj}^{(d)}}{r_{kj}} \frac{\partial u}{\partial r_{kj}} \right) && x_{kj} = x_k - x_j \\
&= \sum_d \left[\left(\frac{\partial}{\partial x_k^{(d)}} \frac{x_{kj}^{(d)}}{r_{kj}} \right) \frac{\partial u}{\partial r_{kj}} + \frac{x_{kj}^{(d)}}{r_{kj}} \frac{\partial}{\partial x_k^{(d)}} \frac{\partial u(r_{kj})}{\partial r_{kj}} \right] \\
&= \sum_d \left[\left(\frac{r_{kj} - x_{kj}^{(d)2} r_{kj}^{-1}}{r_{kj^2}} \right) \frac{\partial u}{\partial r_{kj}} + \frac{x_{kj}^{(d)}}{r_{kj}} \frac{\partial r_{kj}}{\partial x_k^{(d)}} \frac{\partial^2 u}{\partial r_{kj}^2} \right] && \frac{\partial r_{kj}}{\partial x_k^{(d)}} = \frac{x_{kj}^{(d)}}{r_{kj}} \\
&= \sum_d \left[\left(\frac{1}{r_{kj}} - \frac{x_{kj}^{(d)2}}{r_{kj^3}} \right) \frac{\partial u}{\partial r_{kj}} + \frac{x_{kj}^{(d)2}}{r_{kj}^2} \frac{\partial^2 u}{\partial r_{kj}^2} \right] \\
&= \left(\frac{D}{r_{kj}} - \frac{1}{r_{kj}} \right) \frac{\partial u}{\partial r_{kj}} + \frac{\partial^2 u}{\partial r_{kj}^2} \\
&= \frac{D-1}{r_{kj}} \frac{\partial u}{\partial r_{kj}} + \frac{\partial^2 u}{\partial r_{kj}^2} \\
&= \frac{\partial^2 u}{\partial r_{kj}^2} + \frac{D-1}{r_{kj}} \frac{\partial u}{\partial r_{kj}}
\end{aligned}
$$

Since we operate in three dimensions ($D = 3$), we get

$$
\nabla_k^2 u = \frac{\partial^2 u}{\partial r_{kj}^2} + \frac{3-1}{r_{kj}} \frac{\partial u}{\partial r_{kj}} = \frac{\partial^2 u}{\partial r_{kj}^2} + \frac{2}{r_{kj}} \frac{\partial u}{\partial r_{kj}} = u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj})
$$

This leads to the calculation of the second term to be

$$\begin{aligned}
&\nabla_k \left[\prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \nabla_k u(r_{kj}) \right] \\
&= \nabla_k \phi(\mathbf{r}_k) \prod_{i\neq k} \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{ij\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \\
&\quad + \prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} \left(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right)
\end{aligned}$$

A common factor in all our terms is

$$\prod_i \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right)$$

which, if we take a second look at equation (9), we recognize as a part of our wave function! To simplify this enormous expression, we can thus divide our second derivative of Ψ_T by only Ψ_T and we arrive at the wonderful expression

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

with the derivative of $u(r_{kj})$ as

$$\begin{aligned} u'(r_{kj}) &= \frac{\partial}{\partial r_{kj}} u(r_{kj}) & u(r_{kj}) &= \ln f(r_{kj}) \\ &= \frac{1}{f(r_{kj})} f'(r_{kj}) & r_{kj} &= |\mathbf{r}_k - \mathbf{r}_j| \\ &= \frac{1}{\left(1 - \frac{a}{r_{kj}}\right)} \frac{a}{r_{kj}^2} \\ &= \frac{a}{r_{kj}^2 - a r_{kj}} \\ &= \frac{a}{r_{kj}(r_{kj} - a)} \end{aligned}$$

and similarly the derivative of $u(r_{ki})$ as

$$u'(r_{ki}) = \frac{a}{r_{ki}(r_{ki} - a)} \qquad r_{ki} = |\mathbf{r}_k - \mathbf{r}_i|$$

and the double derivative of $u(r_{kj})$ as

$$\begin{aligned} u''(r_{kj}) &= \frac{\partial^2}{\partial r_{kj}^2} u(r_{kj}) \\ &= \frac{\partial}{\partial r_{kj}} u'(r_{kj}) \\ &= \frac{\partial}{\partial r_{kj}} \left(\frac{a}{r_{kj}^2 - a r_{kj}} \right) \\ &= \left(-\frac{a}{(r_{kj}^2 - a r_{kj})^2} \right) (2 r_{kj} - a) \\ &= \frac{-a(2 r_{kj} - a)}{(r_{kj}^2 - a r_{kj})^2} \\ &= \frac{a(a - 2 r_{kj})}{(r_{kj}^4 - 2 r_{kj}^2 a r_{kj} + a^2 r_{kj}^2)} \\ &= \frac{a(a - 2 r_{kj})}{r_{kj}^2 (r_{kj}^2 - 2 a r_{kj} + a^2)} \\ &= \frac{a(a - 2 r_{kj})}{r_{kj}^2 (r_{kj} - a)^2} \end{aligned}$$

We can further calculate the terms given by the double derivative. We have that the gradient term is

$$\frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} = \frac{\nabla_k \exp(-\alpha \mathbf{r}_k^2)}{\phi(\mathbf{r}_k)} = \frac{\exp(-\alpha \mathbf{r}_k^2)\nabla_k(-\alpha \mathbf{r}_k^2)}{\phi(\mathbf{r}_k)} = \frac{\phi(\mathbf{r}_k)(-2\alpha \mathbf{r}_k)}{\phi(\mathbf{r}_k)} = -2\alpha \mathbf{r}_k$$

The laplacian term is

$$\begin{aligned}
\frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} &= \frac{\nabla_k \cdot \nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \\
&= \frac{\nabla_k \cdot [(-2\alpha \mathbf{r}_k)\phi(\mathbf{r}_k)]}{\phi(\mathbf{r}_k)} \\
&= \frac{(\nabla_k(-2\alpha \mathbf{r}_k))\phi(\mathbf{r}_k) + (-2\alpha \mathbf{r}_k)\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \\
&= \frac{-2\alpha(1+1+\beta)\phi(\mathbf{r}_k) + (-2\alpha \mathbf{r}_k)\left[(-2\alpha \mathbf{r}_k)\phi(\mathbf{r}_k)\right]}{\phi(\mathbf{r}_k)} \\
&= \frac{-2\alpha(2+\beta)\phi(\mathbf{r}_k) + (4\alpha^2 \mathbf{r}_k^2)\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \\
&= \frac{(-2\alpha(2+\beta) + (4\alpha^2 \mathbf{r}_k^2))\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \\
&= 4\alpha^2 \mathbf{r}_k^2 - 2\alpha(2+\beta)
\end{aligned}$$

We see that the first term of the laplacian is the squared of the gradient term. The first sum becomes

$$\sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) = \sum_{j \neq k} \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{a}{r_{kj}(r_{kj} - a)} = \sum_{j \neq k} \frac{a \mathbf{r}_{kj}}{r_{kj}^2 (r_{kj} - a)} = a \sum_{j \neq k} \frac{\mathbf{r}_{kj}}{r_{kj}^2 (r_{kj} - a)}$$

The second sum becomes

$$\begin{aligned}
\sum_{ij \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) &= \sum_{ij \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} \frac{a}{r_{ki}(r_{ki} - a)} \frac{a}{r_{kj}(r_{kj} - a)} \\
&= a^2 \sum_{ij \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki}^2 (r_{ki} - a) r_{kj}^2 (r_{kj} - a)}
\end{aligned}$$

The third sum becomes

$$\begin{aligned}
\sum_{j\neq k}\left(u''(r_{kj})+\frac{2}{r_{kj}}u'(r_{kj})\right) &= \sum_{j\neq k}\left(\frac{a(a-2r_{kj})}{r_{kj}^2(r_{kj}-a)^2}+\frac{2}{r_{kj}}\left(\frac{a}{r_{kj}(r_{kj}-a)}\right)\right)\\
&= \sum_{j\neq k}\left(\frac{a^2-2ar_{kj}}{r_{kj}^2(r_{kj}-a)^2}+\frac{2a(r_{kj}-a)}{r_{kj}^2(r_{kj}-a)^2}\right)\\
&= \sum_{j\neq k}\frac{a^2-2ar_{kj}+2ar_{kj}-2a^2}{r_{kj}^2(r_{kj}-a)^2}\\
&= \sum_{j\neq k}\frac{-a^2}{r_{kj}^2(r_{kj}-a)^2}\\
&= -a^2\sum_{j\neq k}\frac{1}{r_{kj}^2(r_{kj}-a)^2}
\end{aligned}$$

Putting the variables into the double derivative given by Equation (11), gives

$$\begin{aligned}
\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}+\frac{2\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right)\\
+\sum_{ij\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj})+\sum_{j\neq k}\left(u''(r_{kj})+\frac{2}{r_{kj}}u'(r_{kj})\right)\\
= 4\alpha^2\mathbf{r}_k^2-2\alpha(2+\beta)+2a(-2\alpha\mathbf{r}_k)\sum_{j\neq k}\frac{\mathbf{r}_{kj}}{r_{kj}^2(r_{kj}-a)}\\
+a^2\sum_{ij\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}^2(r_{ki}-a)r_{kj}^2(r_{kj}-a)}-a^2\sum_{j\neq k}\frac{1}{r_{kj}^2(r_{kj}-a)^2}
\end{aligned}$$

where we see that the first term is a part of the second term squared

$$4\alpha^2\mathbf{r}_k^2=(-2\alpha\mathbf{r}_k)^2$$

9.2.4 Local energy in 3 dimensions with interaction

Using the calculations from sections 9.2.1 - 9.2.3, the local energy becomes

$$\begin{aligned}
E_L(\mathbf{r}) &= \frac{1}{\Psi_T(\mathbf{r})} H\Psi_T(\mathbf{r}) \\
&= \sum_k^N \frac{1}{2}\left(-\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) + x_k^2 + y_k^2 + \gamma^2 z_k^2\right) + \sum_{k<j}^N V_{int}(\mathbf{r}_k, \mathbf{r}_j) \\
&= \sum_k^N \frac{1}{2}\Bigg[- \Bigg(4\alpha^2\mathbf{r}_k^2 - 2\alpha(2+\beta) + 2a(-2\alpha\mathbf{r}_k)\sum_{j\neq k}\frac{\mathbf{r}_{kj}}{r_{kj}^2(r_{kj}-a)} \\
&\quad + a^2\sum_{ij\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki}^2(r_{ki}-a)r_{kj}^2(r_{kj}-a)} - a^2\sum_{j\neq k}\frac{1}{r_{kj}^2(r_{kj}-a)^2}\Bigg) + x_k^2 + y_k^2 + \gamma^2 z_k^2\Bigg] \\
&\quad + \sum_{k<j}^N V_{int}(\mathbf{r}_k, \mathbf{r}_j)
\end{aligned}$$

9.2.5 Drift force (quantum force)

The drift force to be used with importance sampling is given by

$$F = \frac{2\nabla_k\Psi_T}{\Psi_T}$$

The trial wavefunction for interacting bosons is expressed by

$$\Psi_T(\mathbf{r}_i) = \prod_i \phi(\mathbf{r}_i) f(r_{ij}) = \exp(-\alpha(x_i^2 + y_i^2 + \beta z_i^2))\exp\left(\sum_{i<j} u(r_{ij})\right)$$

which gives the drift force (using the new gradient from Equation (21) on the second

term)

$$
\begin{aligned}
F &= \frac{2\nabla_k \Psi_T}{\Psi_T} \\
&= \frac{2\nabla_k \left[\prod_i \phi(\mathbf{r}_i) f(r_{ij})\right]}{\Psi_T} \\
&= 2\frac{\left[\nabla_k \prod_i \phi(\mathbf{r}_i)\right] f(r_{ij}) + \prod_i \phi(\mathbf{r}_i) \left[\nabla_k f(r_{ij})\right]}{\prod_i \phi(\mathbf{r}_i) f(r_{ij})} \\
&= 2\left[\frac{\left[-2\alpha \mathbf{r}_k \prod_i \phi(\mathbf{r}_i)\right] f(r_{ij})}{\prod_i \phi(\mathbf{r}_i) f(r_{ij})} + \frac{\prod_i \phi(\mathbf{r}_i) \left[f(r_{ij}) \nabla_k \sum_{i<j} u(r_{ij})\right]}{\prod_i \phi(\mathbf{r}_i) f(r_{ij})}\right] \\
&= 2\left[-2\alpha \mathbf{r}_k + \nabla_k \sum_{i<j} u(r_{ij})\right] \\
&= 2\left[-2\alpha \mathbf{r}_k + \sum_{j\neq k} \nabla_k u(r_{kj})\right] \\
&= 2\left[-2\alpha \mathbf{r}_k + \sum_{j\neq k} \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{\partial}{\partial r_{kj}} u(r_{kj})\right] \\
&= 2\left[-2\alpha \mathbf{r}_k + \sum_{j\neq k} \frac{\mathbf{r}_{kj}}{r_{kj}} u'(r_{kj})\right] \\
&= 2\left[-2\alpha \mathbf{r}_k + \sum_{j\neq k} \frac{\mathbf{r}_{kj}}{r_{kj}} \frac{a}{r_{kj}(r_{kj}-a)}\right] \\
&= -4\alpha \mathbf{r}_k + 2a \sum_{j\neq k} \frac{\mathbf{r}_{kj}}{r_{kj}^2 (r_{kj}-a)}
\end{aligned}
$$

References

[1] Bose. Plancks gesetz und lichtquantenhypothese. *Zeitschrift für Physik*, 26 (1):178–181, Dec 1924. ISSN 0044-3328. doi: 10.1007/BF01327326. URL https://doi.org/10.1007/BF01327326.

[2] A. Einstein. Quantentheorie des einatomigen idealen gases. *Sitzungsberichte der Preussischen Akademie der Wissenschaften. 1: 3*, 1925.

[3] Jan Klaers, Julian Schmitt, Frank Vewinger, and Martin Weitz. Bose–einstein condensation of photons in an optical microcavity. *Nature*, 468, Nov 2010. doi: 10.1038/nature09567.

[4] J. L. DuBois and H. R. Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Phys. Rev. A*, 63:023602, Jan 2001. doi: 10.1103/PhysRevA.63.023602. URL https://link.aps.org/doi/10.1103/PhysRevA.63.023602.

[5] Morten Hjorth-Jensen. Computational physics, lecture notes fall 2015. Department of Physics, University of Oslo, 2015. URL https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/Literature/lectures2015.pdf.

[6] M. Hjorth-Jensen. Lecture notes on Variational Monte Carlo, 2018. URL http://compphysics.github.io/ComputationalPhysics2/doc/pub/vmc/html/vmc.html.

[7] M. Hjorth-Jensen. Lecture notes on conjugate descent and similar methods, 2018. URL http://compphysics.github.io/ComputationalPhysics2/doc/pub/cg/html/cg.html.

[8] Paul Richard Charles Kent. Techniques and Applications of Quantum Monte Carlo. Robinson College, Cambridge, 1999. URL https://web.ornl.gov/~kentpr/thesis/Thesis.html.

[9] Håkon Emil Kristiansen. Time Evolution of Quantum Mechanical Many-Body Systems. University of Oslo, 2017. URL https://www.duo.uio.no/bitstream/handle/10852/60338/1/H-konEmilKristiansen_Thesis.pdf.

[10] Marius Jonsson. Blocking code. URL https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/Programs/BlockingCodes/blocking.py.

[11] J. K. Nilsen, J. Mur-Petit, M. Guilleumas, M. Hjorth-Jensen, and A. Polls. Vortices in atomic bose-einstein condensates in the large-gas-parameter region. *Phys. Rev. A*, 71:053610, May 2005. doi: 10.1103/PhysRevA.71.053610. URL `https://link.aps.org/doi/10.1103/PhysRevA.71.053610`.