# Variational Monte Carlo for Bosonic Systems

Philip K. Sørli Niane
Department of Physics, University of Oslo, Norway
Github address: Link here

March 26, 2021

**Abstract**

In this article, a quantum mechanical system consisting of bosons trapped in a spherical and elliptical harmonic oscillator were simulated using the variational Monte Carlo method. The optimal variational parameters were found using the gradient decent method. The ground state energies were calculated with interacting and non interacting bosons. For the non interacting case, the energy found using both the Metropolis algorithm and the Metropolis-Hastings algorithm as sampling methods equaled the exact energy. The found energies where found to be 0.5 $\hbar\omega$ per boson in one dimension, 1.0 $\hbar\omega$ per boson in two dimensions and 1.5 $\hbar\omega$ per boson in three dimensions. This was calculated for 1, 10, 100 and 500 bosons, with the optimal parameter $\alpha = 0.5$. The interacting case on the other hand resulted in the following ground state energies: 24.2 $\hbar\omega$ for 10 bosons with $\alpha = 0.4951$, 127.25 $\hbar\omega$ for 50 bosons with $\alpha = 0.4847$ and 265.5$\hbar\omega$ for 100 bosons with $\alpha = 0.4696$ in three dimensions with an elliptical trap. Compared to [1] and [2] the dissimilarity was 0.8%, 4.3% and 7.0% for 10, 50 and 100 bosons in the trap respectively, which is a satisfactory result. Computing the one body density concluded that the forces emerging from the Jastrow factor were quiet weak.

# Contents

# 1 Introduction

Simulating quantum mechanical systems and calculating ground state energies are quiet popular quests within numerical computing and many body quantum mechanics. In this article there will be explored how one can simulate a variety of particle-amounts in one, two and three dimensions, while having the particles constricted by a harmonic oscillator. The particles will be bosons that in first place will not have the possibility to interact with each other, but will gain the skill to interact after the non interacting systems have been tested.

The ground state energies will also be explored. The exploration of the energies will be investigated by using the variational Monte Carlo method, using the most advantageous variational parameters. The variational parameters will be found by using the gradient decent method.

Both the Metropolis algorithm and the Metropolis-Hastings algorithm will be used and compared in addition to comparing the same methods with different step sizes, to see how different sampling methods can be used in variational Monte Carlo methods. The one body density will also be explored and analyzed, with and without implementation of the Jastrow factor, to see if the electron correlation between particles are repulsive, attractive or quiet weak and neutral.

# 2 Theory

## 2.1 The Variational Method

Most many-body systems in quantum mechanics have the property of being highly advanced and complex. Because of this complexity an analytical solution might not always be available. That is why the variational method often is used to find a highly accurate estimate.

The variational method is based on the fact that the expectation value of the Hamiltonian H is an upper bound for the ground state energy $E_{gs}$, for any wave function chosen. The proof can be seen in [3]

$$E_{gs} \leq \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \tag{1}$$

Where $\Psi_T$ is the trial wave function chosen. The trial wave function can be any wave function but it is usually chosen according to the quantum mechanical system, due to similar systems usually having quiet similar wave functions. To ensure lowest possible energy, the trial wave function contains some parameter(s) that is determined by deciding which value of the parameter(s) is giving the lowest energy, knowing that this energy is just an upper bound for the ground state energy [4].

## 2.2 The Quantum Mechanical System

### 2.2.1 The Hamiltonian

The Hamiltonian operator of the system is given by the following:

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

Where N is the number of particles in the system. $V_{ext}$ is the trap potential described in (3) suppressing the particles and $V_{int}$ is the potential created by interacting particles described in (4).

### 2.2.2 The Potential

The quantum mechanical system consists of bosons trapped in a speherical or elliptical trap in one, two or three dimensions. The traps can be expressed like the following equation:

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2} m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2} m[\omega_{ho}^2(x^2+y^2) + \omega_z^2 z^2] & (E) \end{cases} \tag{3}$$

Where (S) is the spherical trap, and (E) is the elliptical trap. $w_{ho}$ is the trap frequency in the xy-direction, while $w_z$ is the trap frequency in the z-direction perpendicular to $w_{ho}$.

There is also internal potential between the bosons. The boson-boson interaction potential can be expressed as the following:

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{4}$$

Where $a$ is the diameter of the bosons. Which makes sense due to bosons not being able to merge together giving $V_{int} = \infty$ when the distance is less than $a$, and $V_{int} = 0$ else.

### 2.2.3 The Trial Wave Function

The trial wave function is based on the fact that similar systems tend to have similar wave functions. The quantum mechanical system contributes with potential energy through the harmonic oscillator traps and boson-boson interaction. Therefore the trial wave function will consist of two parts. The first part arise from the trap potential, giving a trial function proportional with the harmonic oscillator function:

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

Where $\alpha$ and $\beta$ are variational parameters. When dealing with spherical traps $\beta = 1$ making the elliptical trap symmetric in all directions.

The second part of the trial wave function arise from the interaction potential, giving a trial function that depends on the distance between the particles.

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

When dealing with non-interacting cases, $a = 0$ and $\alpha = 1/2a_{ho}^2$.

Putting the two parts together gives the final trial wave function for the quanum mechanical system:

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i)\right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|)\right], \tag{7}$$

where $\alpha$ and $\beta$ are the variational parameters.

### 2.2.4 The Systems Local Energy

The systems local energy can be described by combining equation (1), the Hamiltonian in equation (2) and the trial wave function from equation (7), which gives the following expression:

$$E_L = \frac{1}{\Psi_T} H \Psi_T = \sum_{i=1}^{N} \left( -\frac{\hbar^2}{2m\Psi_T} \nabla_i^2 \Psi_T + V_{\text{ext}}(\boldsymbol{r}_i) \right) + \sum_{i<j}^{N} V_{\text{int}} \tag{8}$$

After inserting the trial wave function the final analytical expression for the local energy can the be written as:

$$\begin{aligned} E_L = -\frac{\hbar^2}{2m} \sum_{i=1}^{N} & \left( \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left( \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \right) \right. \\ & + \sum_{i \neq k} \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \\ & \left. + \sum_{j \neq k} \left( u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right) \right) + \sum_{i=1}^{N} V_{\text{ext}}(\boldsymbol{r}_i) + \sum_{i<j}^{N} V_{\text{int}} \end{aligned} \tag{9}$$

Where the derivation can be seen in the appendix.

## 2.3 Variational Monte Carlo

The variational Monte Carlo method is a method that practice the variational principle from section 2.1 to find an estimation of the ground state energy.

After finding a trial function $\Psi_T$ containing some variational parameter, the probability distribution function (PDF) will be the following [5]:

$$P(\mathbf{r},\alpha,\beta) = \frac{|\Psi_T(\mathbf{r},\alpha,\beta)|^2}{\int |\Psi_T(\mathbf{r},\alpha,\beta)|^2 d\mathbf{r}} \tag{10}$$

Combining the PDF with the local energy expressed as:

$$E_L(\mathbf{r},\alpha,\beta) = \frac{1}{\Psi_T(\mathbf{r},\alpha,\beta)} H\Psi_T(\mathbf{r},\alpha,\beta), \tag{11}$$

Gives the following expectation value of the Hamiltonian [6]:

$$E[H(\alpha)] = \int P(\mathbf{r},\alpha,\beta)E_L(\mathbf{r},\alpha,\beta)d\mathbf{r} \approx \frac{1}{K}\sum_{i=1}^{K} E_L(\mathbf{r_i},\alpha) \tag{12}$$

Where K is the number of Metropolis steps.

### 2.3.1 Metropolis Algorithm

The Metropolis algorithm is used when proposing and carrying on a step in a Monte Carlo simulation. In obediance with the metropolis algorithm, steps are chosen randomly by the following equation:

$$r^{new} = r^{old} + \in [0,1] \cdot dr \tag{13}$$

Where dr is a set step length. The step length is multiplied by a random number from the interval [0,1].

After the new step is proposed the decision of acceptance or rejection remains. Calculating the ratio $\phi$ and demanding that the ratio is larger than a random variable from the interval [0,1] accepts the step. The ratio is computed as the following:

$$\phi = \frac{|\Psi_T(\mathbf{r}^{new})|^2}{|\Psi_T(\mathbf{r}^{old})|^2} \tag{14}$$

Having the steps accepted by the Metropolis algorithm is securing that the simultation and sampling is mainly focused around the largest part of the probability density function, because the ratio can be written as:

$$\frac{|\Psi_T(\mathbf{r}^{new})|^2}{|\Psi_T(\mathbf{r}^{old})|^2} = \frac{P(\mathbf{r}^{new})}{P(\mathbf{r}^{old})} \tag{15}$$

### 2.3.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm, also known as importance sampling is an algorithm that can be used instead of the Metropolis algorithm for sampling and moving particles. The movements is affected by the trial wave function, while the Fokker-Planck equation and the Langevin equation is used to produce the particle route.

The main difference between a usual Metropolis algorithm and the Metropolis-Hasting algorithm is that the later implements a so called drift force which is helping the particle move toward the largest part of the probability density function. The drift force $F$ is expressed as the following:

$$F = \frac{2\nabla\Psi_T}{\Psi_T}. \tag{16}$$

Finding the drift force by using the trial wave function found in equation (7) gives the following expression:

$$\boldsymbol{F}_k = -4\alpha\left(x_k\hat{\boldsymbol{\imath}} + y_k\hat{\boldsymbol{\jmath}} + \beta z_k\hat{\boldsymbol{k}}\right) + 2\sum_{j\neq k}\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}u'(r_{kj}) \tag{17}$$

Where the derivation of the expression can be seen in the appendix.

Further into the Metropolis-Hastings algorithm there is a need for Green's function. The reason for this is that Green's function works as the transition probability contributing to the ratio $\phi$ seen in equation (14), this way:

$$\phi = \frac{G(\boldsymbol{r}^{old}, \boldsymbol{r}^{new})|\Psi_T(\boldsymbol{r}^{old})|^2}{G(\boldsymbol{r}^{new}, \boldsymbol{r}^{old})|\Psi_T(\boldsymbol{r}^{new})|^2} \tag{18}$$

Where Green's function is expressed as followed:

$$G(\boldsymbol{r}^{old}, \boldsymbol{r}^{new}) = \frac{1}{(4\pi D\Delta t)^{3/2}}\sum_{i=1}^{N}\exp\left(-\frac{(\boldsymbol{r}_i^{old} - \boldsymbol{r}_i^{new} - D\Delta t\boldsymbol{F}_i(\boldsymbol{r}^{new}))^2}{4D\Delta t}\right) \tag{19}$$

Where D is the diffusion constant and $\Delta t$ is the stepstep.

After the new step is proposed the decision of acceptance or rejection remains. Calculating the ratio and demanding that the ratio $\phi$ is larger than a random variable from the interval [0,1] accepts the step.

The reason for using the Metropolis-Hastings algorithm instead of the Metropolis algorithm is that more steps would be accepted due to the drift force leading forward in the PDF. More on Metropolis-Hasting algorithm in [7].

## 2.4 Gradient Decent

A clever way of finding the best variational parameter $\alpha$ that gives the lowest energy for the quantum mechanical system, is using the gradient decent method to search for the best value of $\alpha$.

The gradient decent method is quiet straight forward. An initial alpha, also known as a guess, is needed to start of the method and compute the next $\alpha$. The gradient decent formula is the following [8]:

$$\alpha_{n+1} = \alpha_n - \gamma \frac{dE_L}{d\alpha} \tag{20}$$

Where $\gamma$ is the step size.

To compute the derivative of $E_L$, the following expression can be used:

$$\frac{\partial E_L}{\partial \alpha}(\alpha_n) = 2\left( \Big\langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} E_L[\alpha_n] \Big\rangle - \Big\langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} \Big\rangle \Big\langle E_L[\alpha_n] \Big\rangle \right) \tag{21}$$

Where the derivation can be seen in the appendix.

## 2.5 The Blocking Method

Using numerical computing techniques, combined with randomness always introduces some kind of error. To deal with these statistical errors the so called blocking technique is an excellent choice when the variables are correlated.

The blocking method is used by starting of with multiple correlated samples N. The samples are then divided into a number of blocks. The amount of samples need to be of the form $2^K$ to ensure that the blocks are possible to create. Then by calculating the average within each block, the correlation will fade away, and the standard deviation can be calculated as uncorrelated samples. More on the blocking method can be seen here [9]

## 2.6 One body density

When wanting to compare theoretical results with experimental results, the one body density is a popular way to do it. The one body density is a way of distinguishing between different wave functions in addition to having a look on how particle correlations is affecting the quantum mechanical system.

The one body density can be expressed as follows:

$$\rho(\boldsymbol{r_1}) = \int d\boldsymbol{r_2} d\boldsymbol{r_3} ... d\boldsymbol{r_N} |\Psi(d\boldsymbol{r_1} d\boldsymbol{r_2} ... d\boldsymbol{r_N})|^2 \tag{22}$$

Where $\rho(\boldsymbol{r_1})$ is the possibility of finding particle $\boldsymbol{r_1}$ within a distance r from the origin.

There are multiple ways of normalizing the one body density, and one of them is normalizing $\rho$ by having $\int \rho(\boldsymbol{r_1}) d\boldsymbol{r_1}$ return the amount of particles in the system.

# 3 Method

## 3.1 The Program Flow

The flow of the variational Monte Carlo implementation starts of by determining the desired parameters, and the preferred methods e.g. gradient decent, Metropolis-Hastings algorithm, numerical differentiation etc. The variational Monte Carlo simulation then starts of by spreading the particles into random positions within a grid. After everything is initialized, one particle at the time is picked at random and moved according to a proposed step which is either accepted or rejected. If the step is accepted the move is done, else the particle stands still. The acceptance and rejection is implemented as explained in section 2.3.1 and 2.3.2.

## 3.2 Implementation of the Gradient Decent

Implementation of the gradient decent is quiet straight forward. Using equation (20) combined with equation (21) gives a new $\alpha$. Therefore the Variational Monte Carlo is simulated multiple times, where each iteration gives a more optimized $\alpha$. The step size used in the implementation is 0.005. The length of the simulation can be long and short depending on how hard it is for the gradient decent to converge towards an optimal $\alpha$. The optimization method runs the simulation until one of two options are ticked. Either an amount of 35 iterations is reached, or $\frac{dE_L}{d\alpha}$ is less than a tol of $10^{-5}$.

## 3.3 The Hamiltonian as a dimensionless expression

When implementing the expression for the Hamiltonian it is natural to implement the Hamiltonian in the form a dimensionless expression. By using a dimensionless expression, the HAmiltonian can be expressed as follows:

$$H = \sum_{i=1}^{N} \left( -\nabla_i^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2 \right) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{23}$$

Where the differentiation can be seen in the appendix.

## 3.4 Numeric Evaluation

The analytical expression for the double derivative of the trial wave function is a quiet complex expression, especially when particles are interacting with eachother. Because of this complexity, numerical differentiation will also be implemented to find the expression.

The numerical method used for finding the derivative is a simple finite difference formula for approximating second derivatives. The formula used is the following:

$$f''(X) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{24}$$

## 3.5 The Blocking Method

To find the standard deviation of the local energies found, the code on blocking written by Marius Jonsson [9] will be used.

## 3.6 Parallelization of the System

Since the code only will be ran on one computer, it is sufficient to parallelize the code by using fork(). Fork() duplicates the current state of the processor which then can be changed and ran parallel to the original, also called a parent process. By duplicating the state of the code right after kicking of the simulation, and assigning different parameters to each process, the simulation will run multiple times in parallel instead of just one. This way it is possible to collect N times more results from each run, where N is how many times the process is duplicated. onto another core.

# 4 Results

## 4.1 Non Interacting Particles

In this subsection the results for non interacting particle systems will be unveiled. All the simulations done in this section were performed with a=0, $\beta = 1$ and $w_{ho} = 1$. The computations were also done using a spherical trap.

### 4.1.1 Choosing the Step Length and Time Step

Before setting of the more heavy calculations, the step length used in the Metropolis algorithm and the time step used in the Metropolis-Hastings algorithm was examined. Therefore the mean energies for different step lengths and time steps were plotted as a functions of the amount of Monte Carlo cycles/Metropolis steps. A random variational parameter of $\alpha = 0.4$ were used. The results for the variation of step length for the Metropolis algorithm can be seen in figure 1. This was also done for the Metropolis-Hastings method by varying the time step. The results can be seen in figure 2.

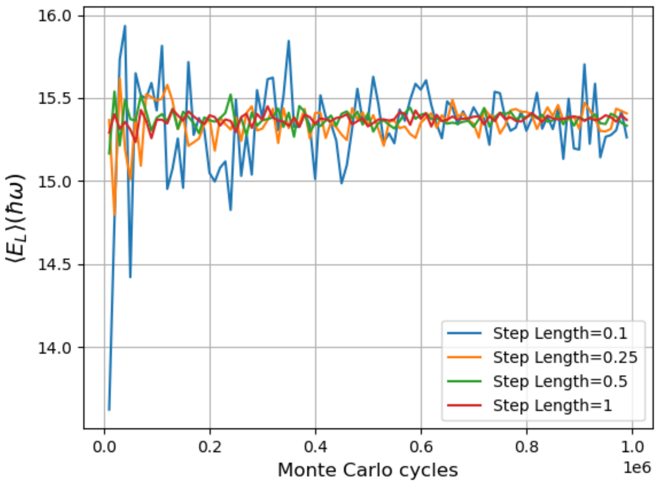

Figure 1: The mean energy plotted as a function of Monte Carlo cycles/Metropolis steps, using the Metropolis algorithm with different step lengths with non interacting particles. N=10, dimensions=3, $\alpha = 0.4$ and $\beta = 1$

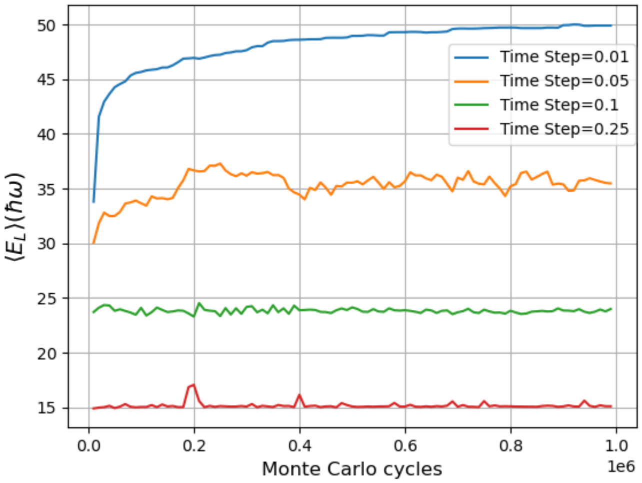

Figure 2: The mean energy plotted as a function of Monte Carlo cycles/Metropolis steps, using the Metropolis-Hastings algorithm with different time steps with non interacting particles. N=10, dimensions=3, $\alpha = 0.4$ and $\beta = 1$

The chosen step length for the Metropolis algorithm was decided to be set as 0.5 for the upcoming calculations, while the time step for the Metropolis-Hastings algorithm was decided to be set as 0.25 for the upcoming calculations. The reason for this can be seen in section 5.1.

### 4.1.2 Finding the Lowest Energy: Varying the Variational Parameter by a Set Amount

By changing the variational parameter $\alpha$ by a set amount, it is easy to get an estimate of where the minimum is. Therefore the mean energy was calculated by varying $\alpha$ by 0.05 in the interval of $[0.3, 0.7]$. The results can be seen in figure 3 and figure 4.

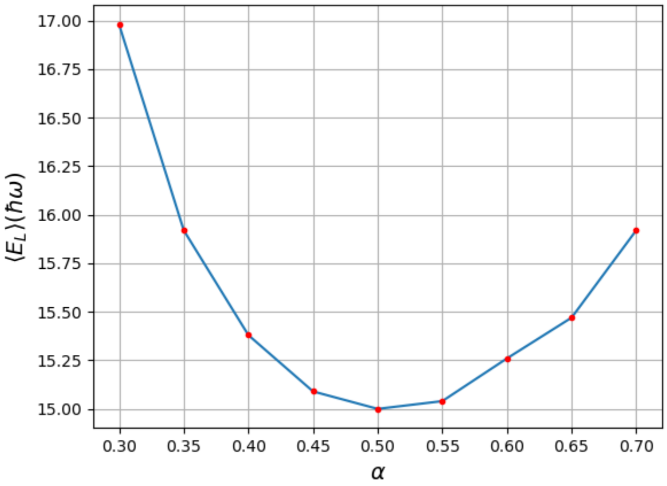

Figure 3: The mean energy plotted as a function of the variational parameter $\alpha$, using the Metropolis algorithm with non interacting particles. Metropolis steps=$2^{19}$, N=10, dimensions=3, and $\beta = 1$

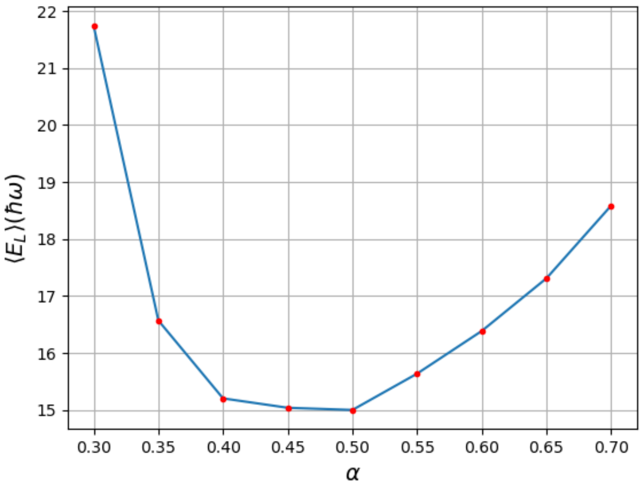

Figure 4: The mean energy plotted as a function of the variational parameter $\alpha$, using the Metropolis-Hastings algorithm with non interacting particles. Metropolis steps=$2^{19}$, N=10, dimensions=3, and $\beta = 1$

Both the Metropolis and the Metropolis-Hastings methods had an energy minimum at $\alpha = 0.5$.

#### 4.1.3 Finding the Lowest Energy: Finding the Variational Parameter with Gradient Decent

Differing from varying the variational parameter $\alpha$ by a set amount, it is also possible to find the best variational parameter by using an optimization method. Therefore a simple gradient decent method was used and tested in search of the best variational parameter. The initial $\alpha$ was set to 0.3 in figure 5 while the initial $\alpha$ was set to 0.7 in figure 6.

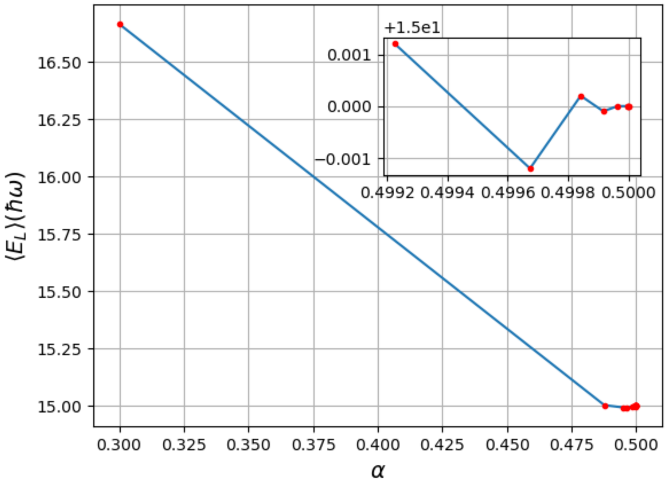

Figure 5: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the standard Metropolis algorithm with non interacting particles. Initial $\alpha = 0.3$, Metropolis steps per iteration $=10^5$, GD step size $\gamma = 0.005$, N=10, dimensions=3, and $\beta = 1$. Zoomed in plot of the last convergence part can be seen in the corner of the main plot, showing the method settling down at $\alpha = 0.5$.

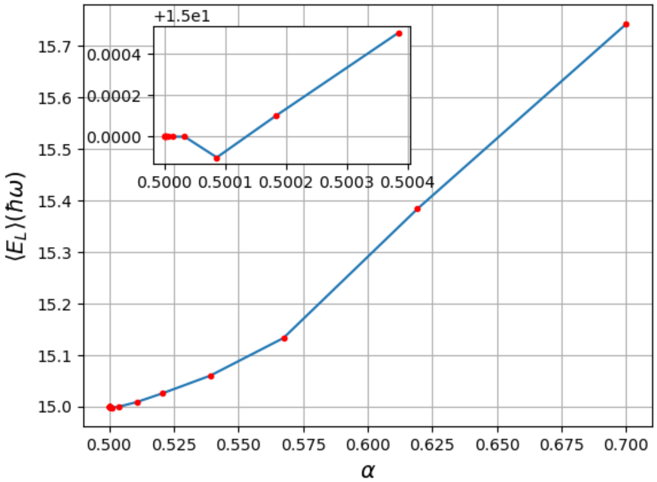

Figure 6: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the standard Metropolis algorithm with non interacting particles. Initial $\alpha = 0.7$, Metropolis steps per iteration=$10^5$, GD step size $\gamma = 0.005$, N=10, dimensions=3, and $\beta = 1$. Zoomed in plot of the last convergence part can be seen in the corner of the main plot, showing the method settling down at $\alpha = 0.5$.

In both cases, starting from each side of the optimal $\alpha$, the values of alpha both converged to 0.5. The gradient decent was also tested for a larger amount of particles with N=100 to see how it performed. The results can be seen in figure 7.

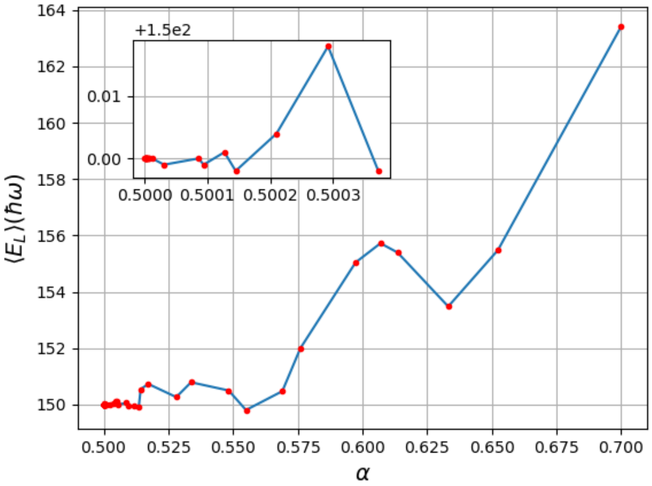

Figure 7: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the Metropolis algorithm with non interacting particles. Initial $\alpha = 0.7$, Metropolis steps per iteration=$10^5$, GD step size $\gamma = 0.005$, N=100, dimensions=3, and $\beta = 1$. Zoomed in plot of the last convergence part can be seen in the corner of the main plot, showing the method settling down at $\alpha = 0.5$.

It was a bit of a 'bumpy ride' on the road towards 0.5, due too some instability in the algorithm, but it ended up at 0.5 in all three cases so the results are favoured.

### 4.1.4 The Local Energies

The methods were tested for a variety of different amounts of particles and dimensions. The derivatives of the trial wavefunctions were also calculated by using analytical and numerical differentiation. The results by using the standard Metropolis algorithm can be seen in table 1, 2 and 3.

| Particles | $E_E(\hbar\omega_{ho})$ | $E_A(\hbar\omega_{ho})$ | $E_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | $E_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 0.5 | 0.5 | 0 | $1.3{\cdot}10^{-9}$ | 1.91 | 2.35 | 0.5 |
| 10 | 5 | 5 | 5 | 0 | $1.3{\cdot}10^{-8}$ | 6.67 | 35.2 | 0.5 |
| 100 | 50 | 50 | 50 | 0 | $1.7{\cdot}10^{-7}$ | 53.9 | 3397 | 0.5 |
| 500 | 250 | 250 | 250 | 0 | $6.0{\cdot}10^{-6}$ | 257 | 60038 | 0.5 |

Table 1: Local energies in 1 dimension with N=1, 10, 100 and 500 particles, using the Metropolis algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

| Particles | $E_E(\hbar\omega_{ho})$ | $E_A(\hbar\omega_{ho})$ | $E_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | $E_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | $1.8{\cdot}10^{-9}$ | 1.95 | 3.05 | 1 |
| 10 | 10 | 10 | 10 | 0 | $1.7{\cdot}10^{-8}$ | 6.83 | 66.0 | 1 |
| 100 | 100 | 100 | 100 | 0 | $1.2{\cdot}10^{-7}$ | 55.0 | 6525 | 1 |
| 500 | 500 | 500 | 500 | 0 | $6.0{\cdot}10^{-6}$ | 263 | 257434 | 1 |

Table 2: Local energies in 2 dimension with N=1, 10, 100 and 500 particles, using the Metropolis algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

| Particles | $E_E(\hbar\omega_{ho})$ | $E_A(\hbar\omega_{ho})$ | $E_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | $E_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.5 | 1.5 | 1.5 | 0 | $1.9{\cdot}10^{-9}$ | 1.99 | 3.78 | 1.5 |
| 10 | 15 | 15 | 15 | 0 | $1.5{\cdot}10^{-8}$ | 6.98 | 97.1 | 1.5 |
| 100 | 150 | 150 | 150 | 0 | $1.3{\cdot}10^{-7}$ | 55.9 | 9607 | 1.5 |
| 500 | 750 | 750 | 750 | 0 | $9.2{\cdot}10^{-7}$ | 269 | 357039 | 1.5 |

Table 3: Local energies in 3 dimension with N=1, 10, 100 and 500 particles, using the Metropolis algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

The methods were also tested and ran for different number of particles, and dimensions by using the Metropolis-Hastings algorithm. The results from the computations can be seen in table 4, 5 and 6.

| Particles | E$_E(\hbar\omega_{ho})$ | E$_A(\hbar\omega_{ho})$ | E$_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | E$_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 0.5 | 0.5 | 0 | $4.3{\cdot}10^{-10}$ | 1.90 | 2.75 | 0.5 |
| 10 | 5 | 5 | 5 | 0 | $4.2{\cdot}10^{-9}$ | 6.59 | 40.0 | 0.5 |
| 100 | 50 | 50 | 50 | 0 | $6.7{\cdot}10^{-8}$ | 54.8 | 3006 | 0.5 |
| 500 | 250 | 250 | 250 | 0 | $9.2{\cdot}10^{-7}$ | 135 | 133992 | 0.5 |

Table 4: Local energies in 1 dimension with N=1, 10, 100 and 500 particles, using the Metropolis-Hasting algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

| Particles | E$_E(\hbar\omega_{ho})$ | E$_A(\hbar\omega_{ho})$ | E$_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | E$_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | $1.7{\cdot}10^{-9}$ | 1.96 | 3.64 | 1 |
| 10 | 10 | 10 | 10 | 0 | $1.2{\cdot}10^{-8}$ | 6.77 | 72.1 | 1 |
| 100 | 100 | 100 | 100 | 0 | $9.4{\cdot}10^{-8}$ | 55.0 | 5093 | 1 |
| 500 | 500 | 500 | 500 | 0 | $4.4{\cdot}10^{-7}$ | 137 | 257844 | 1 |

Table 5: Local energies in 2 dimension with N=1, 10, 100 and 500 particles, using the Metropolis-Hasting algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

| Particles | E$_E(\hbar\omega_{ho})$ | E$_A(\hbar\omega_{ho})$ | E$_N(\hbar\omega_{ho})$ | $\mu_A$ | $\mu_N$ | CPU time$_A(s)$ | CPU time$_N(s)$ | E$_E/N(\hbar\omega_{ho})$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.5 | 1.5 | 1.5 | 0 | $1.9{\cdot}10^{-9}$ | 2.00 | 4.56 | 1.5 |
| 10 | 15 | 15 | 15 | 0 | $1.6{\cdot}10^{-8}$ | 6.97 | 104.4 | 1.5 |
| 100 | 150 | 150 | 150 | 0 | $1.3{\cdot}10^{-7}$ | 28.8 | 6896 | 1.5 |
| 500 | 750 | 750 | 750 | 0 | $5.6{\cdot}10^{-7}$ | 137 | 356536 | 1.5 |

Table 6: Local energies in 3 dimension with N=1, 10, 100 and 500 particles, using the Metropolis-Hasting algorithm, with non interacting particles. $\beta = 1$, Metropolis steps=$2^{19}$. The standard deviation $\mu$ was computed by using the blocking method. The undercored E, A and N stands for exact, analytical and numerical.

## 4.2 Interacting Particles

Then it's time for the interacting cases. In this subsection the results for interacting particle systems will be unveiled. All the simulations done in this section were performed with a=0.0043, $\beta = 2.82843 (w_z = 2.82843)$ and $w_{ho} = 1$. The computations were also done using an elliptical trap.

### 4.2.1 Finding the Lowest Energy: Varying the Variational Parameter by a Set Amount

To get an estimation of where the optimal alpha is laying, the mean energy was calculated by varying $\alpha$ by a set amount of 0.05 in the interval of $[0.3, 0.7]$. The results can be seen in figure 8.

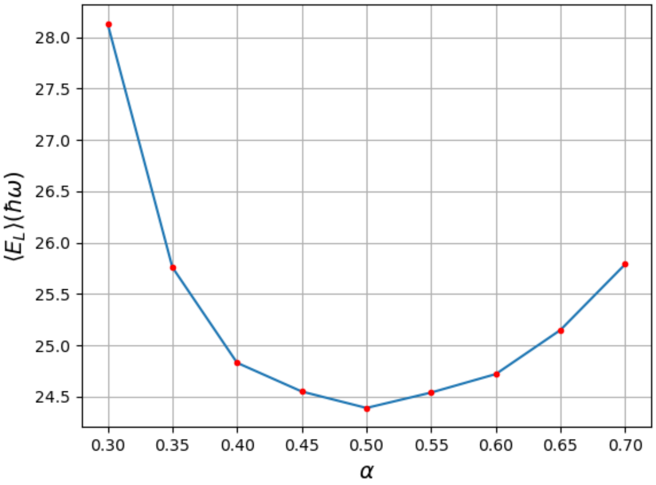

Figure 8: The mean energy plotted as a function of the variational parameter $\alpha$, using the Metropolis algorithm with interacting particles and numerical differentiation. Metropolis steps=$2^{18}$, N=10, dimensions=3, and $\beta = 2.82843$

### 4.2.2 Finding the Lowest Energy: Finding the Variational Parameter with Gradient Decent

To find the best value of $\alpha$ in the interacting case, the gradient decent was performed with three different amounts of particles in three dimensions. From 8 it is quiet intuitive that the optimal $\alpha$ is laying between 0.45 and 0.55, which is why the initial $\alpha$ was set as 0.45 in all three cases. The results can be seen in figure 9, 10 and 11.

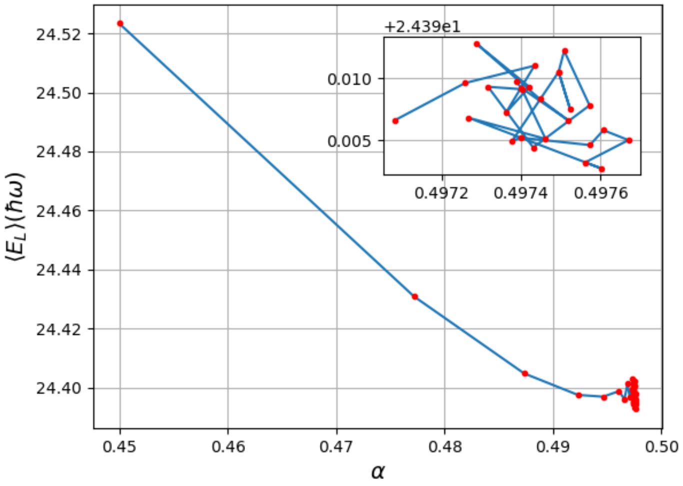

Figure 9: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the Metropolis algorithm with numerical differentiation and interacting particles. Initial $\alpha = 0.45$, Metropolis steps per GD cycle $=10^3$, GD step size $\gamma = 0.005$, N=10, dimensions=3, and $\beta = 2.82843$. Zoomed in plot of the last convergence part can be seen in the corner of the main plot. The method converges towards $\alpha$=0.4951.

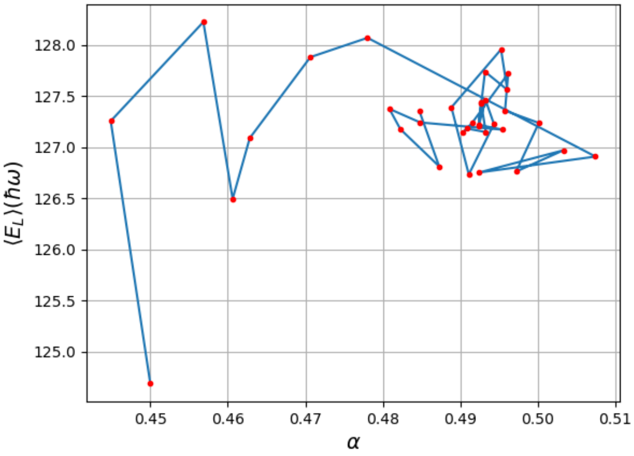

Figure 10: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the Metropolis algorithm with numerical differentiation with interacting particles. Initial $\alpha = 0.45$, Metropolis steps per GD cycle $=10^3$, GD step size $\gamma = 0.005$, N=10, dimensions=3, and $\beta = 2.82843$. The method settled at $\alpha$=0.4847 after some unstable walking around.

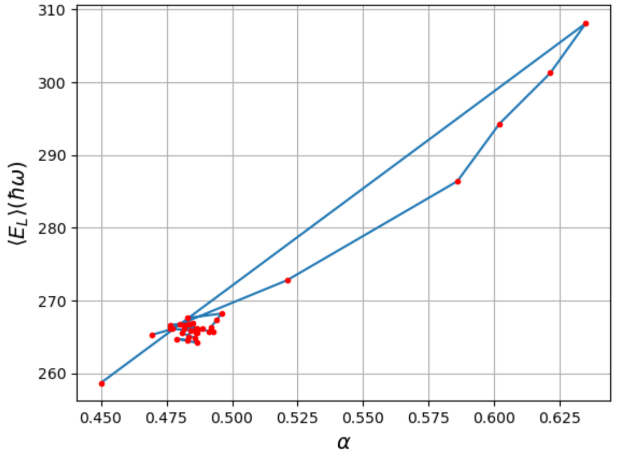

Figure 11: The mean energy plotted as a function of the variational parameter $\alpha$, using the gradient decent method and the Metropolis algorithm with numerical differentiation with interacting particles. Initial $\alpha = 0.45$, Metropolis steps per GD cycle $=10^3$, GD step size $\gamma = 0.005$, N=10, dimensions=3, and $\beta = 2.82843$. The method settled at $\alpha$=0.4696 after some unstable walking

### 4.2.3 The Local Energies

After running gradient decent for the three cases with interacting particles, the resulting $\alpha$ for each case was used to compute the local energy, and then compared to the values found in [1] and [2]. The results can be seen in table 7.

| Particles | $E_{from[1][2]}(\hbar\omega)$ | $E_C(\hbar\omega)$ | $\mu_C(\hbar\omega)$ | $\alpha$ | M. steps | Dissimilarity (%) | CPU time(s) | $E_C/N(\hbar\omega)$ |
|---|---|---|---|---|---|---|---|---|
| 10 | 24.2 | 24.396 | 0.004 | 0.4951 | $2^{16}$ | 0.8 | 155.838 | 2.44 |
| 50 | 122 | 127.25 | 0.07 | 0.4847 | $2^{16}$ | 4.3 | 17725.5 | 2.55 |
| 100 | 247 | 265.5 | 0.4 | 0.4696 | $2^{16}$ | 7.0 | 259034 | 2.66 |
| 10 | 24.2 | 24.400 | 0.005 | 0.4951 | $2^{20}$ | 0.8 | 4609.3 | 2.44 |

Table 7: Results from N=10, N=50 and N=100 particles with interacting particles, using the Metropolis algorithm with numerical differentiation and values of $\alpha$ found with gradient decent optimization. $\beta = 2.82843$, and $2^{16}$ metropolis steps were used in addition to one run with $2^{20}$ steps for N=10. The deviation was computed by using the blocking method. The undercored C stands for computed.

### 4.2.4 One Body Density

The one body density was plotted for two cases. One with 10 particles and one with 50 particles. Both the plots can be seen in figure 12.

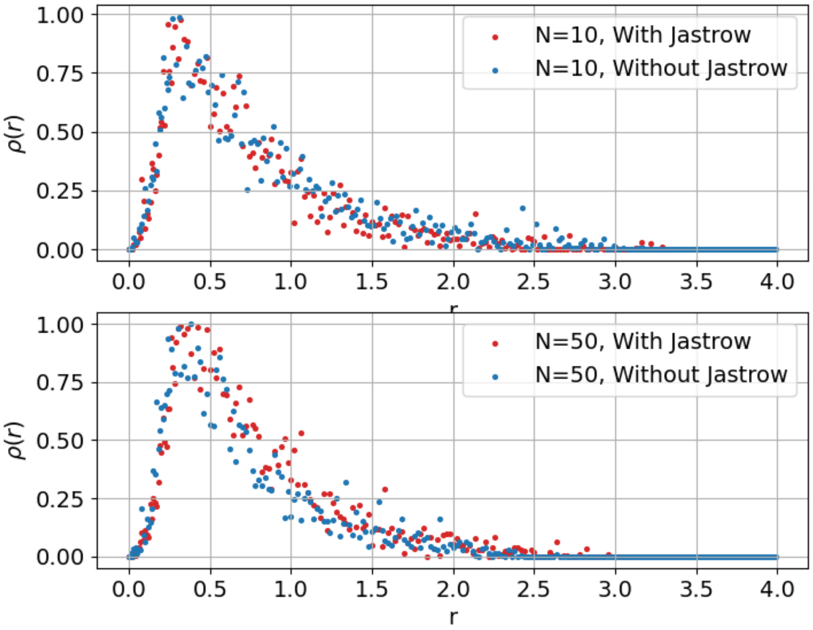

Figure 12: The one body density for N=50 in the upper subplot and N=10 in the lower subplot, with and without the Jastrow factor. The probability is plotted as a function of length r from the origin.

# 5 Discussion

## 5.1 Choosing the Step Length and Time Step

It is easy to see in figure 1, that the Metropolis algorithm was the most stable with a step length of 1 or 0.5, compared to the functions with step length 0.1 and 0.25. The decision of choosing 0.5 over 1 was because of the amount of accepted steps. With a step length of 1, an acceptance rate of 74% followed, while with a step length of 0.5, an acceptance rate of 86% followed, making it the most preferable step length.

For the Metropolis-Hastings algorithm the most stable mean energy values was obtained with a time step of 0.1 and 0.25. The reason 0.25 came out as the preferred time step, was because it followed with an acceptance rate of 65% while a time step of 0.1 only gave an acceptance rate of 38% which is quiet low using the Metropolis-Hastings algorithm.

## 5.2 Non Interacting Particles

### 5.2.1 Finding the Variational Parameter $\alpha$

Finding the best value of $\alpha$ to be used in the non interacting case was a combination of plotting $\alpha$ as a function of the energy while changing $\alpha$ by a set amount. This gave an estimate of where the minimum was laying so it would be easier to choose an initial $\alpha$ when using gradient decent.

In figure 3 and 4 the only difference was that in the former figure the Metropolis algorithm was used, while in the later figure the Metropolis-Hastings algorithm was used. Both resulted in the optimal $\alpha$ of 0.5 which naturally shows that both methods uses the same optimal parameters.

The Metropolis algorithm continued to be used while finding $\alpha$ using gradient decent. To test the gradient decent method, figure 5 and 6 were created to see if choosing initial values on both sides of the minimum would result in the same $\alpha$. It did, both initial values ended up with $\alpha = 0.5$, which shows that the gradient decent method works on both sides of the minimum.

Then the gradient decent method was tested with a higher amount of particles, N=100 to be exactly. With a higher amount of particles, $\alpha$ resulted in being 0.5 which was the expected. The unexpected was that the road towards $\alpha = 0.5$ was a bit more bumpy than expected. In figure 7 the gradient decent method had some jumps on the way towards 0.5, which could be due to some stability issues in the implementation of the algorithm, in addition to the amount of Metropolis steps used per iteration might have been a bit low for the energy to settle down. Due too all the gradient decent tests finishing with the favoured results of $\alpha = 0.5$, it was concluded that the gradient decent managed to give preferred results, and the affection of the instability not being on a large scale. The gradient decent method continued being used forward in the project.

### 5.2.2 Calculation of the Local Energies

Having a look at table 1, 2 and 3 showing the results from the Metropolis algorithm, shows that the calculated energies are really close to the exact energies. The analytically calculated energies equals the correct energies while the numerical calculated energies are almost exactly the same, but with a really small deviation on the order between $10^{-9}$ to $10^{-7}$. Which shows that both analytically and numerical differentiation works just fine. The same conclusion stands for table 4, 5 and 6 showing results form the Metropolis-Hastings algorithm.

Timewise, running the simulations seem to increase the time needed to run quiet steady with increasing particles for both Metropolis algorithm and Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm seems to use a bit less time than the Metropolis Algorithm, but this might be the cause of a bit ineffective implementation for the Metropolis-Hastings case. The time needed to simulate seems to stay quiet constant when adding and subtracting dimensions, except for when the particle amounts with numerical differentiation reaches 100, then the time needed to simulate increases almost 100x more than the time needed for simulating 10x less particles. This might be due to some 'traffic jam' when the code parallelizes such a high amount of particles. Even though the results are still correct.

## 5.3 Interacting Particles

### 5.3.1 Finding the Variational Parameter $\alpha$

Finding the best value of $\alpha$ to be used in the interacting case was a combination of plotting $\alpha$ as a function of the energy in figure 8 This gave an estimate of where the minimum was laying, which opened up for choosing $\alpha = 0.45$ as the initial value for the gradient decent simulations in the interacting case.

The gradient decent method was used three times for the interacting case for N=10,50 and 100 particles which gave $\alpha$= 0.4951, 0.4847 and 0.4696 respectively. Which seems like values within reason of the quantum mechanical system. In figure 9, 10 and 11, which shows the route from 0.45 to the mentioned values of $\alpha$, exposes the instability of the gradient decent method with some of the energy values in the plots being lower than the minimum found. The reason causing the instability is most probably due to the low amount of Metropolis steps used when calculating the derivative in the gradient decent. Because of time consuming simulations the amount of steps were lowered from $10^5$ to $10^3$ when moving from the non interacting case, to the interacting case. The amount of steps being the cause is supported by the plots from gradient decent applied to the non interacting cases compared to the same plots for the interacting cases, showing that the non interacting cases is much more stable.

### 5.3.2 Calculation of the Local Energies

Having a look at table 7 showing the results from the fully interacting cases, shows that the calculated energies using the values of $\alpha$ are quiet close to the referenced energies in [1] and [2]. With N=10 the difference was 0.8%, for N=50, the difference was 4.3% and when N=100, the difference is 7.0%. Which is results that is within approval. Especially the case with 10 particles which had under 1% difference. The energy per particle seems to increase a small bit when increasing the particle amounts, but this might be due to the increase in numerical instability when increasing the amount of particles.

The three interacting simulations from table 7 were done with $2^{16}$ Metropolis steps. The simulation were done again for N=10 particles but this time with $2^{20}$ Metropolis steps. The second simulation resulted in approximately the same result which assists that the simulations in the same table were run with a high enough number of Metropolis steps.

### 5.3.3 The One Body Density

From figure 12 it is not quiet easy to see a difference between the plots with the Jastrow factor included and without the Jastrow factor which indicates that the interacting forces between particles are quiet weak in this case. In the lower plot with 50 particles in the system, especially in the range of [0.5,1] the plot without the Jastrow factor is pushing itself a bit in front of the plot with the Jastrow factor. This indicates that the Jastrow factor contributes with a repulsive force in the system. That supports the upper subplot in the same figure where there is only 10 particles present, because less particles leads to less contribution from the Jastrow factor since there is less particles to interact with each other.

# 6 Conclusion

After exploring the quantum mechanical system using the variational Monte Carlo method, it has been shown how it is possible to simulate trapped bosons in different types of harmonic oscillator traps, in one, two and three dimensions. Both the Metropolis algorithm and the Metropolis-Hastings algorithm as sample methods worked well and gave approximately similar results. The simulation in both cases where the bosons were interacting and not interacting, expressed satisfactory results.

In the non interacting case it was also shown that the ground state energies were calculated with such a small error that claiming the results to be exact is within reach. When the bosons were interacting, the results were closer to the referenced values of [1] and [2] when the amount of bosons in the trap were smaller. Even though the results were shown to be acceptable with a dissimilarity of 7% in the case of having 100 interacting bosons, while having 0.8% dissimilarity when the amount of bosons summed up to 10. The 50 boson case had 4.3% dissimilarity.

The one body density also shows that in this quantum mechanical system of interacting bosons, the forces contributed by interaction is quiet weak, and needs more particles embedded in the system for a higher contribution of in this case repulsion.

## 6.1 Prospects for the future

The main prospects for the future would quiet naturally be to optimize the gradient decent method or switch out gradient decent with another optimisation algorithm, for example a stochastic gradient decent method. As most codes, there is always room for optimization. Optimizing the code where it is possible, would be satisfying, making the code run more effectively saving computer resources. There would also be naturally to scout for more ways to calculate the double derivative of the trial function. Another prospect for the future that sounds quiet fun, would be to inspect similarities and dissimilarities in the same quantum mechanical system if it were to contain fermions instead of bosons.

# References

[1] J. L. DuBois and H. R. Glyde, “Bose-einstein condensation in trapped bosons: A variational monte carlo analysis,” *Phys. Rev. A*, vol. 63, p. 023 602, 2 Jan. 2001. DOI: 10.1103/PhysRevA.63.023602. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.63.023602.

[2] 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*, vol. 71, p. 053 610, 5 May 2005. DOI: 10.1103/PhysRevA.71.053610. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.71.053610.

[3] D. J. Griffiths, *Introduction to Quantum Mechanics (3rd Edition)*, 3rd. 2018, pp. 327–331.

[4] Wikipedia contributors, *Variational method (quantum mechanics) — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Variational_method_(quantum_mechanics)&oldid=1004825982, [Online; accessed 27-February-2021], 2021.

[5] ——, *Variational monte carlo — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Variational_Monte_Carlo&oldid=1000813186, [Online; accessed 27-February-2021], 2021.

[6] Morten Hjorth-Jensen, *Lecture notes-introduction to the course and start variational monte carlo*, http://compphysics.github.io/ComputationalPhysics2/doc/pub/week1/pdf/week1.pdf, [Online; accessed 27-February-2021], 2021.

[7] W. K. HASTINGS, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970, ISSN: 0006-3444. eprint: https://academic.oup.com/biomet/article-pdf/57/1/97/8545904/57-1-97.pdf.

[8] Wikipedia contributors, *Gradient descent — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Gradient_descent&oldid=1008553674, [Online; accessed 28-February-2021], 2021.

[9] M. Jonsson, “Standard error estimation by an automated blocking method,” *Phys. Rev. E*, vol. 98, p. 043 304, 4 Oct. 2018. DOI: 10.1103/PhysRevE.98.043304. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.98.043304.

[10] Morten Hjorth-Jensen, *Finalization of importance sampling, start gradient methods, steepest descent and conjugate gradient methods*, http://compphysics.github.io/ComputationalPhysics2/doc/pub/week5/pdf/week5.pdf, [Online; accessed 1-March-2021], 2021.

# A Appendix

## A.1 Source code

All the source code is located in this GitHub repository.

## A.2 Derivation of an Analytical Expression of the Local Energy

To derive the analytical expression for the local energy, an intuitive place to start is by the ecpression for the local energy and inserting the Hamiltonian expression:

$$
\begin{aligned}
E_L &= \frac{1}{\Psi_T} H \Psi_T && (25)\\
&= \sum_{i=1}^{N} \left( -\frac{\hbar^2}{2m\Psi_T} \nabla_i^2 \Psi_T + V_{\text{ext}}(\boldsymbol{r}_i) \right) + \sum_{i<j}^{N} V_{\text{int}} && (26)
\end{aligned}
$$

The main element of the derivation is the differentiation of the trial wave function.

### A.2.1 The First Derivative

Starting thereby off by finding the first derivative. The differentiation of the trial wave function by particle k, can be written as the following:

$$\nabla_k \Psi_T = \nabla_k \left[ \prod_i \phi(\mathbf{r}_i) \right] \exp\left( \sum_{i<j} u(r_{ij}) \right) \tag{27}$$

By exchanging $f(r_{ij}) = \exp\left(u(r_{ij})\right)$ and $g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right] = \phi(\mathbf{r}_i)$. Where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$.

The product rule seems like a nice way to differentiate the expression. The first part before the exponential function can be differentiated by extracting particle $k$ from the product, this part is left as:

$$\nabla_k \phi(\mathbf{r}_k) \left[ \prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left( \sum_{i<j} u(r_{ij}) \right) \tag{28}$$

By using the following notation:

$$\sum_{i<j}^{N} V_{ij} \equiv \sum_{i=1}^{N} \sum_{j=i+1}^{N} V_{ij}, \tag{29}$$

It is possible to write the second part as follows:

$$\prod_i \phi(\boldsymbol{r}_i)\nabla_k \left[\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{j-1} u(r_{ij})\right)\right] \tag{30}$$

To differentiate the exponential part it is necessary to inspect the sums. Since the expression is differentiated by the $k$'th particle, all the terms without this particle fades away due to the differentiation. The second part in the product rule can then be written as the following, consisting of two sums where particle $k$ is included in all the terms:

$$\prod_i \phi(\boldsymbol{r}_i) \exp\left(\sum_{i<j} u(r_{ij})\right)\left[\sum_{j=k+1}^{n} \nabla_k u(r_{kj}) + \left[\sum_{i=1}^{k-1} \nabla_k u(r_{ik})\right]\right] \tag{31}$$

Combining equation (28) and equation (31) the first derivative stands as the following:

$$\begin{aligned}\nabla_k \Psi_T = \nabla_k \phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\left(\sum_{i<j} u(r_{ij})\right)\\ + \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right)\sum_{j\neq k} u(r_{kj})\end{aligned} \tag{32}$$

### A.2.2 The Second Derivative

To calculate the second derivative the same procedure as the first differentiation is used:

$$\begin{aligned}\nabla_k^2 \Psi_T = \nabla_k \Bigg( \nabla_k \phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\left(\sum_{i<j} u(r_{ij})\right)\\ + \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right)\sum_{j\neq k} u(r_{kj})\end{aligned} \tag{33}$$

Which can be written as the following using the product rule the same way as in section A.2.1. Which gives the following expression:

$$\begin{aligned} \nabla_k^2 \Psi_T = \nabla_k^2 \phi(\mathbf{r}_k) \left[ \prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp \left( \sum_{i<j} u(r_{ij}) \right) \\ +2 \left( \nabla_k \phi(\mathbf{r}_k) \left[ \prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp \left( \sum_{i<j} u(r_{ij}) \right) \left[ \sum_{j \neq k} \nabla_k u(r_{kj}) \right] \right) \\ + \prod_i \phi(\boldsymbol{r}_i) \exp \left( \sum_{i<j} u(r_{ij}) \right) \left[ \sum_{j \neq k} \nabla_k u(r_{kj}) \right]^2 \\ + \prod_i \phi(\boldsymbol{r}_i) \exp \left( \sum_{i<j} u(r_{ij}) \right) \left[ \sum_{j \neq k} \nabla_k^2 u(r_{kj}) \right] \end{aligned} \tag{34}$$

Which is the second derivative of the trial function.

### A.2.3 The Final Local Energy Expression

Now that almost all the elements are ready, the main thing missing is to express:

$$\frac{1}{\Psi_T(\mathbf{r})} \nabla_k^2 \Psi_T(\mathbf{r}) \tag{35}$$

And the toss everything inside equation (26) and compute the local energy.

Firstly, equation (34) is divided by $\Psi_T$ which gives the following:

$$\begin{aligned} \frac{1}{\Psi_T} \nabla_k^2 \Psi_T = \frac{\nabla_k^2 \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2 \frac{\nabla_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} \sum_{j \neq k} \nabla_k u(r_{kj}) \\ + \left( \sum_{j \neq k} \nabla_k u(r_{kj}) \right)^2 + \sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) \end{aligned} \tag{36}$$

This can be written as

$$\begin{aligned} = \frac{\nabla_k^2 \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2 \frac{\nabla_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} \sum_{j \neq k} \nabla_k u(r_{kj}) \\ + \sum_{j \neq k} \nabla_k u(r_{kj}) \sum_{j \neq k} \nabla_k u(r_{kj}) + \sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) \end{aligned} \tag{37}$$

It is easy to see that the summations in three of the four terms is the same. By writing the summations on a different form, it is possible to derive a better expression for the local energy. Therefore the summation part will be written the following way by using the chain rule and the derivative of $r_{kj}$ as $\frac{\mathbf{r_k}-\mathbf{r_j}}{r_{kj}}$.

$$\sum_{j\neq k}\nabla_k u(r_{kj}) = \sum_{j\neq k}\frac{du(r_{kj})}{dr_{kj}}\frac{dr_{kj}}{d\mathbf{r_k}} = \sum_{j\neq k}\frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{r_{kj}}u'(r_{kj}) \tag{38}$$

Equation (38) is then inserted into equation (37) which gives the following:

$$\frac{1}{\Psi_T}\nabla_k^2\Psi_T = \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2\frac{\nabla_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)}\left(\sum_{j\neq k}\frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{r_{kj}}u'(r_{kj})\right) + \\ \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \nabla_k\left(\sum_{j\neq k}\frac{\boldsymbol{r}_k-\boldsymbol{r}_j}{r_{kj}}u'(r_{kj})\right) \tag{39}$$

The last part of the expression can be solved with the same procedure as used when finding the double derivatives earlier. Which then leaves us with an expression ready to be slapped into the equation for the local energy:

$$\begin{aligned}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = {} & \frac{\nabla_k^2\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2\frac{\nabla_k\phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)}\left(\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_j)}{r_{kj}}u'(r_{kj})\right) \\ & + \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \\ & + \sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right)\end{aligned} \tag{40}$$

Inserting this into equation (26) gives the local energy:

$$
\begin{aligned}
E_L =& \sum_{i=1}^{N} \left( -\frac{\hbar^2}{2m\Psi_T} \nabla_i^2 \Psi_T + V_{\text{ext}}(\boldsymbol{r}_i) \right) + \sum_{i<j}^{N} V_{\text{int}} \\
=& -\frac{\hbar^2}{2m} \sum_{i=1}^{N} \left( \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left( \sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \right) \right. \\
& + \sum_{i\neq k} \sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \\
& \left. + \sum_{j\neq k} \left( u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right) \right) + \sum_{i=1}^{N} V_{\text{ext}}(\boldsymbol{r}_i) + \sum_{i<j}^{N} V_{\text{int}}
\end{aligned}
\tag{41}
$$

Which is the expression for the local energy of the quantum mechanical system.

## A.3 Deriving the Drift Force

The drift force is expressed as the following:

$$
F = \frac{2\nabla \Psi_T}{\Psi_T}. \tag{42}
$$

Luckily the derivative of the trial wave function derived earlier helps alot for this task. Inserting the first derivative found in section A.2.1 gives the following:

$$
\begin{aligned}
F = \frac{2}{\Psi_T} \nabla_k \phi(\mathbf{r}_k) & \left[ \prod_{i\neq k} \phi(\mathbf{r}_i) \right] \exp\left( \sum_{i<j} u(r_{ij}) \right) \\
& + \prod_{i} \phi(\boldsymbol{r}_i) \exp\left( \sum_{i<j} u(r_{ij}) \right) \sum_{j\neq k} u(r_{kj})
\end{aligned}
\tag{43}
$$

Dividing equation (43) by the trial function the same way done for the double derivative in equation (36) and rewriting the expression by inserting (38) gives the following expression for the drift force:

$$
\boldsymbol{F} = 2 \frac{\nabla_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2 \sum_{j\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}} u'(r_{kj}) \tag{44}
$$

Which is the final expression.

## A.4 Differentiation of $\langle E_L \rangle$ with Respect to $\alpha$

To find an expression for the derivative of the local energy with respect to $\alpha$, the footing is taken from the following lecture notes [10], where the derivation almost is completed with the following expression:

$$\bar{E}_\alpha = 2\left(\langle \frac{\bar{\psi}_\alpha}{\psi[\alpha]} E_L[\alpha] \rangle - \langle \frac{\bar{\psi}_\alpha}{\psi[\alpha]} \rangle \langle E_L[\alpha] \rangle\right) \tag{45}$$

Where we simply trade the following expressions to adapt to our system:

$$\psi[\alpha] \to \Psi_T \tag{46}$$

$$\bar{\psi}_\alpha \to \bar{\Psi}_{T\alpha} \tag{47}$$

Which gives the following expression:

$$\bar{E}_\alpha = 2\left(\langle \frac{\bar{\Psi}_{T\alpha}}{\Psi_T} E_L[\alpha] \rangle - \langle \frac{\bar{\Psi}_{T\alpha}}{\Psi_T} \rangle \langle E_L[\alpha] \rangle\right) \tag{48}$$

This can be rewritten as:

$$\frac{d\langle E_L \rangle}{d\alpha} = 2\left(\langle \frac{E_L}{\Psi_T} \frac{d\Psi_T}{d\alpha} \rangle - \langle E_L \rangle \langle \frac{1}{\Psi_T} \frac{d\Psi}{d\alpha} \rangle\right) \tag{49}$$

Which is the final expression for the derivative of the expectation value to the local energy with respect to $\alpha$.

## A.5 Rewriting the Hamiltonian as a Dimensionless Expression

By starting of with the original expression for the Hamiltonian from equation (2).

$$H = \sum_i^N \left( \frac{-\hbar^2}{2m} \bigtriangledown_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \tag{50}$$

Then inserting the expression for $V_{ext}$:

$$H = \sum_{i=1}^N \left( -\frac{\hbar^2}{2m} \nabla_i^2 + \frac{1}{2} m (x_i^2 \omega_{ho}^2 + y_i^2 \omega_{ho}^2 + \omega_z^2 z_i^2) \right) + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{51}$$

Then it is time to write length in units of $a_0$, $a_0 = (\frac{\hbar}{m\omega_{ho}})^{\frac{1}{2}}$. In other words, the units representing lengths in $V_{ext}$ needs to be multiplied by $a_{ho}$, while the derivative expression needs to be divided by $a_{ho}$, which gives the following expression:

$$H = \sum_{i=1}^{N} \left( -\frac{\hbar^2}{2m} \frac{\omega_{ho} m}{\hbar} \nabla_i^2 + \frac{1}{2} m \frac{\hbar}{m\omega_{ho}} (x_i^2 \omega_{ho}^2 + y_i^2 \omega_{ho}^2 + \omega_z^2 z_i^2) \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{52}$$

Which can be written as the following after getting rid of some of the extra values:

$$H = \sum_{i=1}^{N} \left( -\frac{\hbar\omega_{ho}}{2} \nabla_i^2 + \frac{\hbar\omega_{ho}}{2} (x_i^2 + y_i^2 + \frac{\omega_z^2 z_i^2}{\omega_{ho}^2}) \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{53}$$

Then dividing the expression by $\hbar\omega_{ho}$ to get energy results in terms of $\hbar\omega_{ho}$:

$$H = \frac{1}{2} \sum_{i=1}^{N} \left( -\nabla_i^2 + x_i^2 + y_i^2 + \gamma z_i^2 \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{54}$$

Where $\gamma = \frac{\omega_z}{\omega_{ho}}$