Finding the ground state energy state for trapped, hard sphere Bose gas using Variational Monte Carlo

Lars Opgård & Ali Reza Asghari Zadeh

(Dated: June 2, 2024)

In this project we have created a Variational Monte Carlo program to find the best wavefunction parameters for the lowest state energy for a given trial wavefunction. We start with a Harmonic Oscillator with a simple program that varies the parameters with a set change and a brute-force Metropolis sampling method so that we can compare with the analytical solution. Then we add in a better sampling method with the Metropolis-Hastings method and using the Stochastic gradient descent method too much better optimize for the parameters. We then found the results to fit well with the analytical result and started with introducing the two-body interaction with the Jastrow function. Then we would optimize for the parameters in the interactive case and found that the optimal $\alpha = 4.9$. With the optimal parameter we found the lowest state energy $E = 2.32 \pm 2.95 \cdot 10^{-4}$ for a system with 100 particles. We found that the energy per particles increased with the number if particles in the system. This was agreed with in the article from DuBois and Glyde [2].

CONTENTS

https://github.com/larsfop/fys4411-project1

I. INTRODUCTION

Given a trial wavefunction, we want to find optimal parameters for the lowest energy state. This will be achieved numerically using Variational Monte Carlo method for both a simple non interacting Gaussian and a more complex interacting wavefunction. We will then compare different algorithms like the brute-force standard Metropolis against the more popular and improved Metropolis-Hastings. There exists many different optimizations methods for finding the best wavefunction parameters, like Broyden's algorithm and Steepest descent. We will use stochastic gradient descent too better optimize the wavefunction parameters.

After we have made the VMC program and checked it to the analytical solution we can start with the interactive wavefunction by first optimizing the parameter α. After we have found the best parameter we can run one final Monte Carlo iteration and find the lowest ground state energy for 10, 50 and 100 particles. We then use the blocking method to better do the error analysis. Finally we compute the one-body density and compare with the non interactive case how the density shifts.

II. METHOD

The main goal for this project is finding the lowest energy state for a given trial wavefunction by varying parameters using Variational Monte Carlo. In this project we will use two different versions of the trial wavefunction, the first one being a simple Harmonic oscillator using a simple Gaussian function without any particle interaction. Then we will add interaction to the wavefunction via the Jastrow factor 7 and an inter-boson interaction potential. We will also compare different algorithms like sampling algorithms and parameter optimization algorithms. The system and numerical methods used are described later in this section.

A. System

We will trap the bosons in a hard sphere which can be either spherical (S) or elliptical (E) with N bosons of mass m. The traps are defined as an external potential $V_{\text{ext}}(\vec{r})$

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

and an internal two-body interaction $V_{\text{int}}(\vec{r}_1, \vec{r}_2)$. The Hamiltonian of the system is given as

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

Where ω_{ho}^2 is the trap potential strength. In the case of the elliptical trap, ω_{ho} is the trap frequency in the xy plane and ω_z in the z direction.

The interboson interaction is given by a pairwise hard-core potential

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

where a is the boson diameter, thus this will ensure that our bosons does not clip into eachother.

We will change the units for length in units of a_{ho}, $r \to r/a_{ho}$, and units of energy as $\hbar\omega_{ho}$. Thus the Hamiltonian can be written as

$$H = \sum_{i=1}^{N}\frac{1}{2}(-\nabla_i^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2) + \sum_{i<j}^{N} V_{\text{int}}(|\vec{r}_i - \vec{r}_j|) \quad (4)$$

B. Wavefunction

We make up the trial wavefunction for N particles with two parts. The harmonic oscillator $g(\vec{r})$ and the two-particle interaction Jastrow function $f(|\vec{r}_i - \vec{r}_j|)$. Thus the trial wavefunction come in the form

$$\Psi_T(\vec{r}, \alpha, \beta) = \prod_{i}^{N} g(\vec{r}_i, \alpha, \beta) \prod_{i<j}^{N} f(|\vec{r}_i - \vec{r}_j|, a) \quad (5)$$

where α and β are variational parameters. With the single particle harmonic oscillator

$$g(\vec{r}_i, \alpha, \beta) = \exp\{-\alpha(x^2 + y^2 + \beta z^2)\} \quad (6)$$

For spherical traps $\beta = 1$ and in the non interacting case where $a = 0$ we have $\alpha = \frac{1}{2}a_{ho}^2$. Then the correlation function is given as such

$$f(|\vec{r}_i - \vec{r}_j|, a) = \begin{cases} 0 & |\vec{r}_i - \vec{r}_j| \le a \\ 1 - \frac{a}{|\vec{r}_i - \vec{r}_j|} & |\vec{r}_i - \vec{r}_j| > a \end{cases} \quad (7)$$

We will then approximate the expectation value for the energy, as that would be computationally difficult with all the integrals needed. Instead we will use the local energy E_L, where the expectation value is given by

$$\bar{E} = \int P(\vec{r}) E_L(\vec{r}, \vec{\alpha}) d\vec{r} \approx \frac{1}{N}\sum_{i=1}^{N} E_L(\vec{r}_i) \quad (8)$$

Where the local energy is given by

$$E_L = \frac{1}{\Psi(\vec{r})} H \Psi(\vec{r}) \quad (9)$$

Considering the local energy is computed by the Hamiltonian 2. The double derivative is given in appendix A for the simple gaussian, and appendix B for the interacting wavefunction. We can then find the analytical solution for the local energy for both cases. The simple gaussian can be found from the double derivative A4 and the external potential energy 1

$$E_L = \sum_{i=1}^{N} \alpha(2+\beta) + \frac{1}{2}(x_i^2+y_i^2+z_i^2) - 2\alpha^2(x_i^2+y_i^2+\beta^2 z_i^2) \tag{10}$$

Later for the Metropolis-Hastings algorithm we will also need to compute the quantum force. The quantum force is given by

$$F_Q = \frac{2\nabla\Psi(\vec{r})}{\Psi(\vec{r})} = -4\alpha(x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + \beta z\hat{\mathbf{k}}) \tag{11}$$

Where the single derivative is given here A1.

For the interacting wavefunction we will need to compute the same expressions. First we will define some new variables. First we will use a new Hamiltonian 4. Then define the two new variables

$$\phi(\vec{r}) = g(\vec{r}) \tag{12}$$
$$u(r_{kj}) = \ln f(r_{kj}) \tag{13}$$

The interacting case is more complex and the calculations are given in appendix B. First the local energy E_L is given by

$$\begin{aligned} E_L = \sum_{k}^{N} \Bigg\{ &\frac{1}{2}\frac{\nabla_k^2 \phi(\vec{r}_k)}{\phi(\vec{r}_k)} + \frac{\nabla_k \phi(\vec{r}_k)}{\phi(\vec{r}_k)} \left(\sum_{j\neq k}^{N} \frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj}) \right) \\ &+ \frac{1}{2}\sum_{i\neq k}^{N}\sum_{j\neq k}^{N} \frac{(\vec{r}_k-\vec{r}_i)(\vec{r}_k-\vec{r}_j)}{r_{ki}r_{kj}} u'(r_{ki})u'(r_{kj}) \\ &+ \frac{1}{2}\sum_{j\neq k}^{N}\left[u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right] \\ &+ \frac{1}{2}(x_k^2+y_k^2+\gamma^2 z_k^2) \Bigg\} + \sum_{k<j}^{N} V_{\text{int}}(r_{kj}) \end{aligned} \tag{14}$$

The derivatives are given in appendix B. The quantum force is also given as

$$F_Q = 2\left[\frac{\nabla_k \phi(\vec{r})}{\phi(\vec{r})} + \sum_{j\neq k}^{N} \frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj}) \right] \tag{15}$$

We will also need to calculate the Green's function for importance sampling later. The Green's function is given by

$$\begin{aligned} G(y,x,\Delta t) = &\frac{1}{(4\pi D\Delta t)^{3N/2}} \\ &\times \exp\{-(y-x-D\Delta t F(x))^2/4D\Delta t\} \end{aligned} \tag{16}$$

where x and y are respectably the old and new positions and D the diffusion coefficient. As we need to compute $\frac{G(x,y,\Delta t)}{G(y,x,\Delta t)}$ we can simplify the expression

$$\begin{aligned} \frac{G(x,y,\Delta t)}{G(y,x,\Delta t)} = \exp\Bigg\{ &\frac{1}{2}(F(x)+F(y) \\ &\times \left[\frac{1}{2}D\Delta t(F(x)-F(y)) - y + x\right]\Bigg\} \end{aligned} \tag{17}$$

The calculations for the simplifications are done in appendix C.

C. Numerical approaches

As mentioned we will use Variational Monte Carlo to simulate the particles. MC is a method of using random steps to eventually converge to a state that is close to the analytical solution by going through a large number of cycles. Then varying the wavefunction parameters for each MC cycle. While our trial wavefunction have a few variational parameters, we will only vary the α parameter, as the others are known to us. This will reduce the workload and make it a lot faster to find the best parameters. We will for the simple Gaussian start simple and look through a set interval where we guess the first value and increase it with a set step. This is quite inefficient and we will later introduce a better algorithm that will not waste time far away from the true best value and require a good guess to even arrive to a good result. Our code will roughly follow a simple algorithm (**alg:** 1) with a few tweaks here and there as we develop the program.

Algorithm 1 Variational Monte Carlo

```
• Initialization :
    → Initialize the initial positions $\vec{r}$ and parameters $\vec{\Theta}$ for each particle
One Monte Carlo cycle :        ▷ To be repeated $N$ times
• For each particle
    → Calculate a new position $\vec{r}+\Delta\vec{r}*r$ where r is a random value between 0 and 1
    → Use the Metropolis algorithm to decide if the new position is accepted
    → Sample the necessary information for the current iteration
• Compute the averages
```

We will be using two methods for accepting new states during a MC cycle. Starting with the simple brute-force method Metropolis. The Metropolis algorithm follows the *detailed balance requirement*

$$\frac{A_{y\to x}}{A_{x\to y}} = \frac{w_x T_{x\to y}}{w_y T_{y\to x}} \tag{18}$$

where x and y are the old and new states respectably, $T_{x\to y}$ is the probability for sampling a new state, $A_{x\to y}$ is the probability for accepting a new state y. Then w_x

and w_y will be the probability distribution. We want to maximize the A values such that the probability to change states is

$$A_{x\to y} = \min\left(1, \frac{w_x T_{x\to y}}{w_y T_{y\to x}}\right) \tag{19}$$

For the brute-force method we will only compare the probability distributions w given by

$$w_i = \frac{|\Psi_T(\vec{r}_i|^2}{\int d\vec{r}|\Psi_T(\vec{r})|^2} \tag{20}$$

Thus the brute-force Metropolis algorithm is given by

$$A_{x\to y} = \min\left(1, \frac{|\Psi_T(y)|^2}{|\Psi_T(x)|^2}\right) \tag{21}$$

The probability ratio can be calculated analytically There is one issue with the Metropolis algorithm and that is that there is an equal probability for the state to move in either direction. This means we will be spending equal amounts of time far away from the lowest energy state as close to it, this can however be improved with a more complicated version, Metropolis-Hastings algorithm.

The Metropolis-Hastings is built upon the Metropolis algorithm. It uses two equations, Fokker-Planck and the Langevin equation. The new position in space is given as a solution of the Langevin equation using the Euler's method yielding

$$y = x + DF(x)\Delta t + \epsilon\sqrt{\Delta t} \tag{22}$$

where y and x is the new and old position respectably, D is the diffusion coefficient and is given in atomic units equal to $\frac{1}{2}$ coming from the $\frac{1}{2}$ factor from the kinetic equation. ϵ is a random gaussian variable and Δt is the time step, values varying between $\Delta t \in [0.001, 0.01]$ gives the most stable values for the ground state energy. The Fokker-Planck equation gives us the transition probability given by the Green's function (16). Which means that our probability ratio now is extended with the Green's function as a product with the former Metropolis.

$$q(y,x) = \frac{G(x,y,\Delta t)|\Psi_T(y)|^2}{G(y,x,\Delta t)|\Psi_T(x)|^2} \tag{23}$$

For now we have just varied our parameters from our initial guess with a constant change, this have worked okay for the simple gaussian, but that is only because we can easily find the analytical value for the ground state energy and optimal values. This is much harder to do with a more complicated wavefunction and with the inefficiency we get as we spend a lot of time with values far from the optimal and result is dependent on a good guess and change. Thus we want to introduce a better optimization algorithm. The algorithm we will use is the stochastic gradient descent (SGD) method in this project, but there are many other methods. With this method we will be able to approach the optimal value much faster with much greater precision that is not dependent on our initial guess or how we change the parameters. SGD uses batches of data when computing the gradient. The idea is to minimize the cost function which can be written as such for normal gradient descent (GD) and computing the gradient

$$\nabla_\beta C(\beta) = \sum_{i=1}^{n} \nabla_\beta c_i(\vec{x}_i, \beta) \tag{24}$$

where β is a variational parameter. We sum over all the gradients in the dataset for each change of parameter. With SGD we only take the gradient as a sum over a few batches of data called *mini-batches*. With n data and M size of each mini-batch we have n/M mini-batches, where each mini-batch is denoted by B_k where $k = 1, \ldots, n/M$. Then the gradient is then calculated by

$$\nabla_\beta C(\beta) = \sum_{i=1}^{n} \nabla_\beta c_i(\vec{x}_i, \beta) \to \sum_{i\in B_k}^{n} \nabla_\beta c_i(\vec{x}_i, \beta) \tag{25}$$

We will use the extreme case of SGD where we only have one batch, that means we will sum over the gradients for the whole dataset as we do not have too much data. Thus we will compute the next parameter as

$$\beta_{j+1} = \beta_j - \eta \sum_{i=1}^{n} \nabla_\beta c_i(\vec{x}_i, \beta) \tag{26}$$

where η is a constant we call the learning rate which will make sure that our parameter does not vary to much and have a hard time of converging. The gradient we will compute will be the energy derivative with regards to the variational parameter α $\bar{E}_\alpha$. This value is given as

$$\bar{E}_\alpha = 2\left(\left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} E_L[\alpha] \right\rangle - \left\langle \frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} \right\rangle \langle E_L[\alpha]\rangle\right) \tag{27}$$

Where $\bar{E}_\alpha = \frac{d\Psi[\alpha]}{d\alpha}$ the derivative of the trial wavefunction with regards to α can be solved easily as

$$\frac{\bar{\Psi}_\alpha}{\Psi[\alpha]} = -\sum_{i=1}^{N}(x_i^2 + y_i^2 + \beta z_i^2) \tag{28}$$

With this we can optimize the variational parameters with SGD as

$$\alpha_{j+1} = \alpha_j - \eta \sum_{i=1}^{n} \bar{E}_\alpha \tag{29}$$

Now if our MC cycles where completely independent from eachother we could quite easily find the error on our results by taking the standard deviation

$$s^2 = \frac{1}{N+1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \tag{30}$$

but our cycles are not completely independent from eachother. For each iteration we have random walkers and random number generators (rng's) are pseudo random. That means we have can't just take the variance as our standard deviation. Though there are method to get around this, one of those methods is the blocking method. The blocking method is good for large amount of data, while also becoming more accurate the larger the data is. The blocking method works by taking pairs of the data and taking the average of each pair. Continuing till we have taken the average of the final pair after k iterations given as such

$$(\vec{X}_{i+1})_k = \frac{1}{2}\left((\vec{X}_i)_{2k+1} + (\vec{X}_i)_{2k}\right) \tag{31}$$

We can then sample the covariance, mean and number of observations $(\gamma_i, \sigma_i^2, \vec{X}_i)$ for each i. [Jonsson)Jonsson]

Finally we will need to compute the one-body density. This will give us a plot that show us how the particles spread out from one single particle. The one-body density can be computed from

$$\rho(\vec{r}_1) = \int d\vec{r}_2 d\vec{r}_3 \dots d\vec{r}_N \times |\Psi(\vec{r}_1, \vec{r}_2 \dots \vec{r}_N; \hat{\Theta}|^2 \tag{32}$$

by adding up all the particles that are a distance away from a single particle and grouping them up we end up with density graph that describes what distance is the most likelihood for a particle to be in. Then by comparing the interactive and non-interactive case we can see from the graph whether the interaction is repulsive or attractive from how the graph is shifted.

III. RESULTS AND DISCUSSION

A. Harmonic Oscillator

We start with experimenting with a simple gaussian, this helps us get good results while writing our code. First with the brute-force Metropolis method we found results that follows the analytical result 4 quite well for the given iterations, with the equation for the analytical solution is given in A11. Varying it over α we find that the curve deviates more for value of α further from the minimum, while as expected we get the exact solution when $\alpha = 0.5$ corresponds to the lowest energy state. Where we also get no variance.

Building onto the program with the Metropolis-Hastings method we were able to achieve results that fit much better with the analytical result with the same amount of iterations 5. Though while the results are much closer to the analytical solution, the variance is still more on the same level as with the brute-force method. As shown by plot 7 while the variance is roughly on the same value, the brute-force method is much more unstable. Further we can that as α increases the two sampling methods deviates more and more from eachother after we have passed the lowest energy state as seen by the plot 6.

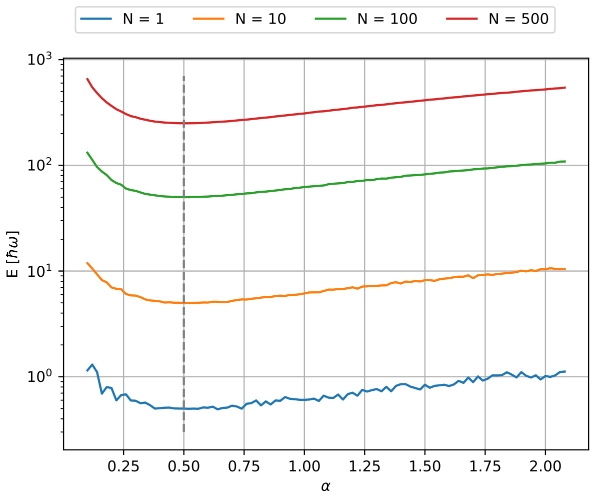

FIG. 1. Plot showing the energy as a function of the wavefunction parameter α for the 1 dimensional case. The graph is done using the brute-force Metropolis method for the simple Gaussian.

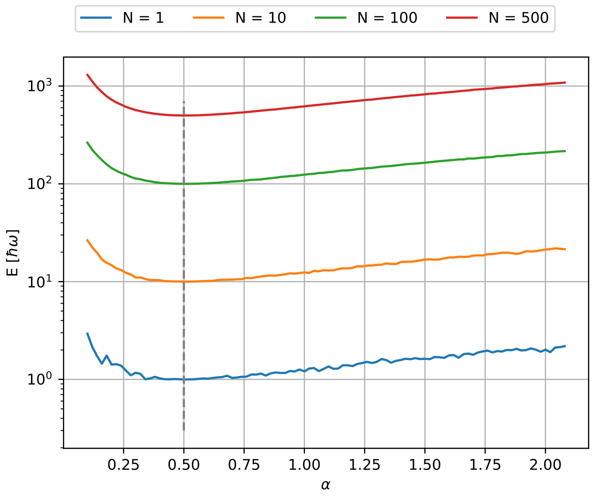

FIG. 2. Plot showing the energy as a function of the wavefunction parameter α for the 2 dimensional case. The graph is done using the brute-force Metropolis method for the simple Gaussian.

B. Two-body Interaction

Next up we have introduced the two-body interaction through the Jastrow function. We will now repeat what we did now setting $\beta = 2.82843$. The two-body interaction is quite a lot more computationally expensive and thus we added parallelization to the program. One thing that is quickly noticed is that our optimization algorithm

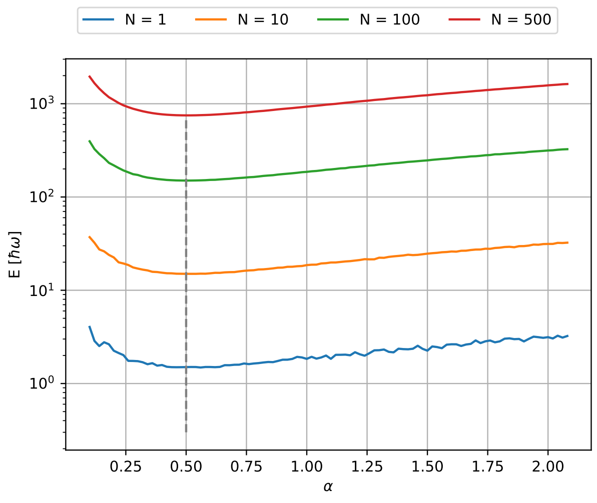

FIG. 3. Plot showing the energy as a function of the wavefunction parameter α for the 3 dimensional case. The graph is done using the brute-force Metropolis method for the simple Gaussian.

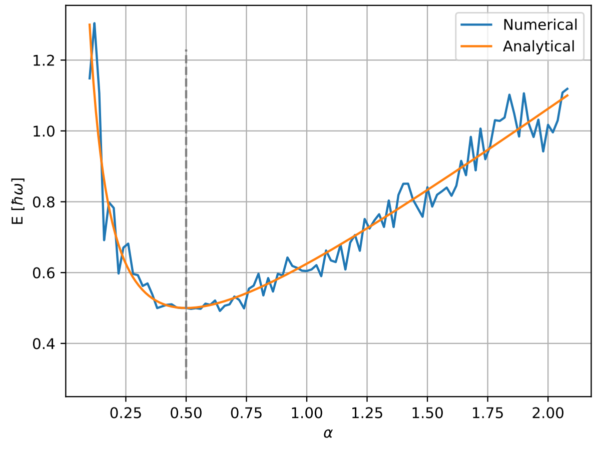

FIG. 4. Analytical and numeric approximation to the simple Gaussian wave function as function of the varying parameter using brute-force Metropolis method.

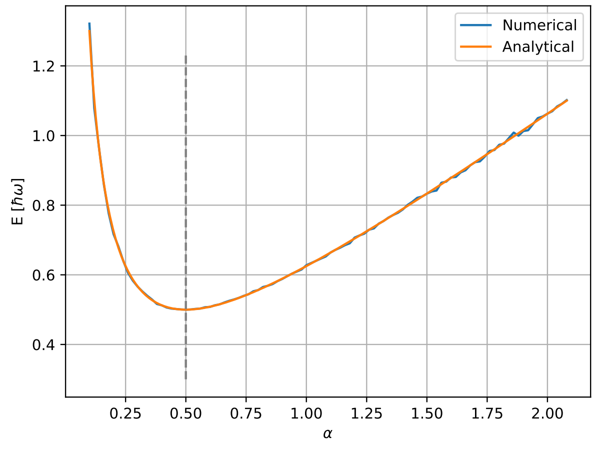

FIG. 5. Analytical and numeric approximation to the simple Gaussian wave function as function of the varying parameter using Metropolis-Hastings method.

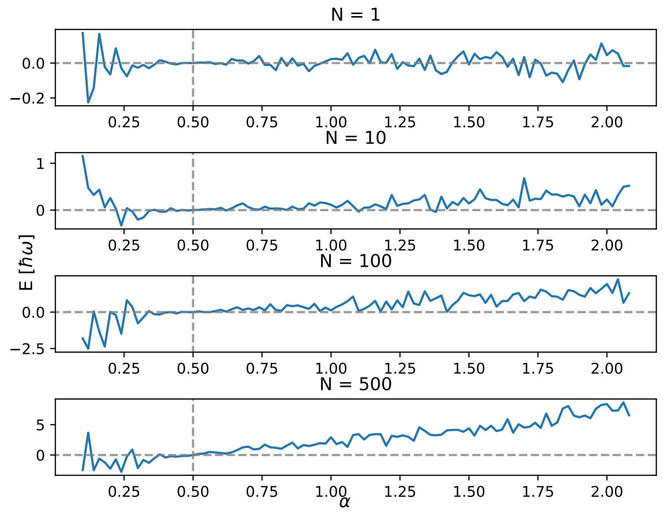

FIG. 6. How the energy changes for brute-force Metropolis versus metropolis-Hastings, as a function of varying parameter α. At a value of 1, they give the same energy.

does not converge to a α value with low enough gradient. This means our results can change a lot depending on what value we choose and that as we will see soon that our error will be a bit larger. Performing the calculations for 10, 50 and 100 sets of particles, we were even though not able to get a perfect value that α would converge to, would still quickly reach a value around 0.49. The plot 8 shows that though the variance quickly reach a rough stability. After running the optimizations we decided that we would use $\alpha = 0.490144$ which was what we got after running the optimization for 100 particles through a hundred optimization iterations. Using the blocking we get the energy with the error and running the interactive wavefunction for the best parameters that the average energy is

$$E = 2.32 \pm 2.95 \cdot 10^{-4} / \hbar\omega \tag{33}$$

Then from plot 9 show that the energy per particle calculated with the two-body interaction increases with number of particles. We did not plot the energy per particle as a function of number of particles as shown in the article [2], but we can still see from both our results and the article that the energy per particle will increase as we add more particles to the calculation. This compared to the harmonic oscillator where the energy per particles will be constant even after increasing the number of particles with a factor of 10 or 100.

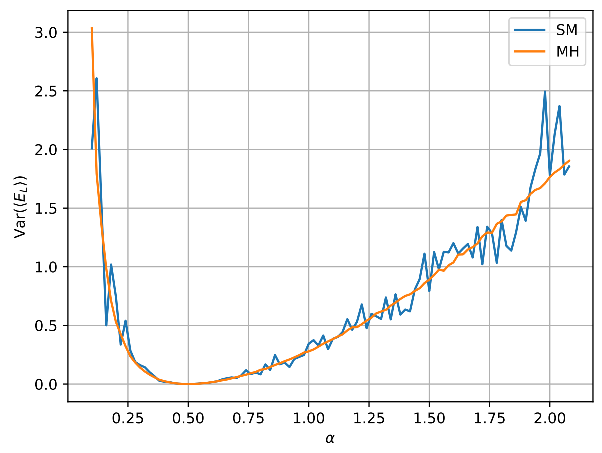

FIG. 7. Variance of the calculated energy from the MC samples for the standard Metropolis and Metropolis-Hastings algorithm. The MH shows much smaller variance.

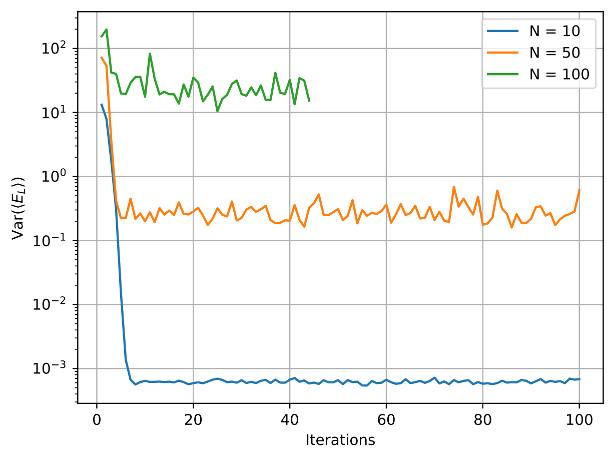

FIG. 8. The variance for the interactive case for 10, 50 and 100 particles as a function of optimization iterations. The variance drops quickly to a stable value as the optimization algorithm finds the optimal parameter. The variance is more chaotic for systems with more particles, this would be do to the number of MC cycles done per optimization iteration, which is then enhanced after taking the average over the 8 threads used for the calculation.

With the optimal parameters we found earlier we can again compute one iteration of VMC with the interactive wavefunction and compute the one-body density. Thus the plot 10 show how the particles will spread away from eachother. As expected we get a slight shift when comparing the two-body interaction with the no interaction wavefunction. As the interaction slightly shifts the graph to the right, that is further away from the particle. This is because of that the Jastrow two-body interaction function is repulsive and will further spread the particles out.

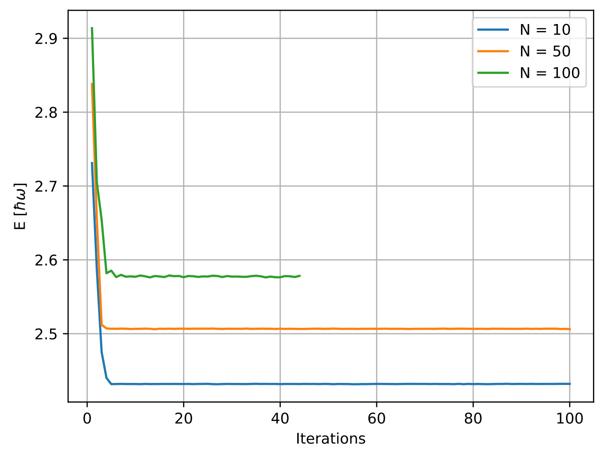

FIG. 9. This graph shows the energy per particle as a function of optimization iterations. As it is shown in [2] the energy per particle increases as the number of particles in the system also increases. The sudden drop in energy comes from the optimization algorithm quickly converging to the optimal parameter from the starting point $\alpha_0 = 0.3$. Afterwards the energy barely changes, which if we compare to the variance 8 you would expect the energy to also behave similarly. Though this could be due to the multi threading giving weird results skewing the average.

The egdges are not supposed to be like that, just an artifact from how I computed the one-body density.

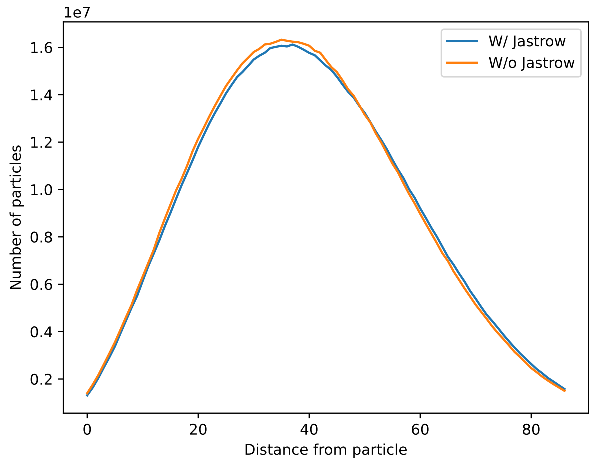

FIG. 10. The one-body density for the interacting case with a system consisting of 100 particles. The two graphs are done with and without the two-body interaction and comparing the two we see as expected that the two-body interaction (Jastrow function) shifts the graph slightly to the right, resulting in a repulsive two-body interaction.

IV. CONCLUSION

In this article we started with a simple gaussian trial wavefunction that we used with the analytical solution to create a Variational Monte Carlo algorithm that will find the best values for the wavefuncion parameters. We decided that we would optimize for a single parameter α. We then made our program more robust by adding more complex algorithm like Metropolis-Hastings for sampling and Stochastic gradient descent for optimizing. After doing this and preparing for the statistical analysis we introduced the two-body interaction function, the Jastrow function 7. With the new and more complicated wavefunction we found the optimal parameter $\alpha = 0.49$ and computed the energy $E = 2.32 \pm 2.95 \cdot 10^{-4}/\hbar\omega$.

[Jonsson)Jonsson] Jonsson, Marius*Standard error estimation by an automated blocking method*, 4: 043304.

[2] DuBois, J. L. / Glyde, H. R.(2001): *Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis*, 2: 023602.

Appendix A: Simple Gaussian

For the simple gaussian doing the derivatives is quite straight forward. Knowing that $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Then the first derivative would be a vector and the result is by using the chain rule

$$\nabla\Psi(\vec{r}) = -2\alpha(x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + \beta z\hat{\mathbf{k}}) \exp\{-\alpha(x^2 + y^2 + \beta z^2)\} \tag{A1}$$

We can then continue with the second derivative following the same process

$$\nabla(\nabla\Psi(\vec{r})) = (-2\alpha(2+\beta) + 4\alpha^2(x^2 + y^2 + \beta^2 z^2)) \exp\{-\alpha(x^2 + y^2 + \beta z^2\} \tag{A2}$$

It is then easy to see how the equation would look in one and two dimensions

$$\nabla^2\Psi(x) = (-2\alpha + 4\alpha^2 x^2) \exp\{-\alpha x^2\}$$
$$\nabla^2\Psi(x,y) = (-4\alpha + 4\alpha^2(x^2 + y^2)) \exp\{-\alpha(x^2 + y^2)\}$$

And from the Hamiltonian 2 we can also find the expression for N particles and the expression for the harmonic oscillator part of the trial wavefunction 6, that we should have to calculate the following expression

$$\sum_{i=1}^{N} \nabla^2\Psi(\vec{r}) = \sum_{j=1}^{N} \nabla_i^2 \prod_{j=1}^{N} \exp\{-\alpha(x_j^2 + y_j^2 + \beta z_j^2)\} \tag{A3}$$

and that each term is only non-zero for when $i = j$. Thus we can omit the product operator and sum over for each particle, which will give us a similar solution to the single particle

$$\sum_{i=1}^{N} (-2\alpha(2+\beta) + 4\alpha^2(x_i^2 + y_i^2 + \beta^2 z_i^2)) \exp\{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\} \tag{A4}$$

and in one and two dimensions

$$\sum_{i=1}^{N} \nabla^2\Psi(x) = \sum_{i=1}^{N} (-2\alpha + 4\alpha^2 x_i^2) \exp\{-\alpha x_i^2\}$$
$$\sum_{i=1}^{N} \nabla^2\Psi(x,y) = \sum_{i=1}^{N} (-4\alpha + 4\alpha^2(x_i^2 + y_i^2)) \exp\{-\alpha(x_i^2 + y_i^2)\}$$

This gives us the local energy for 3 dimensions and N particles

$$E_L = 2N(2+\beta) - \sum_{i=1}^{N} \left[4\alpha^2(x_i^2 + y_i^2 + \beta z_i^2) + V_{ext}(\vec{r}_i)\right] \tag{A5}$$

where the potential energy is given 1.

Finally for the simple gaussian we can simplify our probability ratio

$$w = \frac{|\Psi(\vec{r}_{\text{new}})|^2}{|\Psi(\vec{r}_{\text{old}})|^2} = \prod_{i=1}^{N} \frac{|\exp\{-\alpha((x_i + \Delta x_i)^2 + (y_i + \Delta y_i)^2 + \beta(z_i + \Delta z_i)^2)\}|^2}{|\exp\{(-\alpha(x_i^2 + y_i^2 + \beta z_i^2))\}|^2} \tag{A6}$$

and using that $e^a e^b = e^{a+b}$ and $\frac{e^2}{e^b} = e^{a-b}$, and that

$$(x + \Delta x)^2 - x^2 = x^2 - x^2 + 2x\Delta x + \Delta x^2 = \Delta x(2x + \Delta x) \tag{A7}$$

we get the final expression

$$w = \exp\left\{-2\alpha \sum_{i=1}^{N} (\Delta x_i(2x_i + \Delta x_i) + \Delta y_i(2y_i + \Delta y_i) + \beta\Delta z_i(2z_i + \Delta z_i))\right\} \tag{A8}$$

We can compute the analytical expectation value for the energy $\bar{E}$ with the formula

$$\bar{E}[\alpha] = \int P(\vec{r}) E_L(\vec{r}, \alpha) d\vec{r} \tag{A9}$$

where the probability density function (pdf) $P(\vec{r}) = |\Psi(\vec{r})|^2$ and the local energy we already have computed A5. Thus we can using WolframAlpha to compute the intergral find the analytical solution in one dimension and one particle, and with $\beta = 1$

$$\left(\frac{2a}{\pi}\right)^{1/2} \int_{-\infty}^{\infty} \exp\{-2\alpha x^2\} \frac{1}{2}(2\alpha + x^2(1-\alpha^2)) \tag{A10}$$

which gives us after solving the integral

$$\bar{E} = \frac{4\alpha^2 + 1}{8\alpha} \tag{A11}$$

Appendix B: Interacting Wavefunction

For the first derivative we must use the chain rule on the wavefunction

$$\nabla_k \Psi(\vec{r}) = \left[\nabla_k \prod_i \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} + \left[\prod_i \phi(\vec{r}_i)\right] \left[\nabla_k \exp\left\{\sum_{j<l} u(r_{jl})\right\}\right] \tag{B1}$$

It is then easy to see that ∇_k will only interact with the k'kt element in $\phi(\vec{r}_i)$ and thus that term can be written as

$$\nabla_k \phi(\vec{r}_k) \left[\prod_{i\neq k} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \tag{B2}$$

For the second term we use the chain rule and notice that ∇_k will give non-zero for all terms in the sum except for when $l = k$, this then gives us

$$\left[\prod_i \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m\neq k} \nabla_k u(r_{km}) \tag{B3}$$

which gives us the single k'th derivative of the interacting wavefunction

$$\nabla_k \Psi(\vec{r}) = \nabla_k \phi(\vec{r}_k) \left[\prod_{i\neq k} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} + \left[\prod_i \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m\neq k} \nabla_k u(r_k m) \tag{B4}$$

First we must define $\nabla_k u(r_{kj})$ and $\nabla_k^2 u(r_{kj})$

$$\nabla_k u(r_{kj}) = (\nabla_k r_{kj}) u'(r_{kj}) = \frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj}) \tag{B5}$$

Where we have used that $\frac{d|u|}{du} = \frac{u}{|u|}$ and the substitution that $u = |\vec{r}_k - \vec{r}_j|$. The second derivative is then

$$\nabla_k^2 u(r_{kj}) = \nabla_k\left(\frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj})\right) = \frac{(\vec{r}_k - \vec{r}_j)^2}{r_{kj}^2} u''(r_{kj}) + \left(\nabla_k \frac{\vec{r}_k - \vec{r}_j}{r_{kj}}\right) u'(r_{kj}) \tag{B6}$$

Then we can rewrite the expression for the single derivative by dividing by the trial wavefunction which will be used for the quantum force

$$\frac{\nabla_k \Psi(\vec{r})}{\Psi(\vec{r})} = \frac{\nabla_k \phi(\vec{r})}{\phi(\vec{r})} + \sum_{j\neq k} \frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj}) \tag{B7}$$

The first term with the double derivative of u is simple as it follows the same process as for the first derivative B5 squared and that $\frac{(\vec{r}_k-\vec{r}_j)^2}{r_{kj}^2} = 1$. The second term require the usage of the quotient rule

$$\begin{aligned}
&\frac{(\nabla_k(\vec{r}_k - \vec{r}_j)|\vec{r}_k - \vec{r}_j| - (\nabla_k|\vec{r}_k - \vec{r}_j|)(\vec{r}_k - \vec{r}_j)}{r_{kj}^2} \\
&= \frac{|\vec{r}_k - \vec{r}_j| - (\vec{r}_k - \vec{r}_j)\frac{(\vec{r}_k-\vec{r}_j)}{|\vec{r}_k-\vec{r}_j|}}{|\vec{r}_k - \vec{r}_j|^2} \\
&= \frac{|\vec{r}_k - \vec{r}_j|^2 - (\vec{r}_k - \vec{r}_j)^2}{|\vec{r}_k - \vec{r}_j|^3}
\end{aligned}$$

Which gives us

$$\nabla_k^2 u(r_{kj}) = u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \tag{B8}$$

Taking the derivative of the previous result B4 we get five terms, let us start with the first one, which will just be the second derivative of $\phi(\vec{r}_k)$

$$\nabla_k^2 \phi(\vec{r}_k) \left[\prod_{i \neq k} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \tag{B9}$$

Then when we do the derivative of the $\exp\{u\}$ in the first term and ϕ in the second term we get the same expression

$$2\nabla_k \phi(\vec{r}_k) \left[\prod_{i \neq k} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m \neq k} \nabla_k u(r_k m) \tag{B10}$$

Using B5 we get

$$2\nabla_k \phi(\vec{r}_k) \left[\prod_{i \neq k} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \left(\sum_{m \neq k} \frac{\vec{r}_k - \vec{r}_j}{r_{kj}} u'(r_{kj})\right) \tag{B11}$$

The next step is the derivative of $\exp\{u(r_{kj})\}$ in the second term. This term will be similar to the second term already with an additional u'.

$$\left[\prod_{i} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m \neq k} \nabla_k u(r_k m) \sum_{n \neq k} \nabla_k u(r_k n) \tag{B12}$$

which gives us

$$\left[\prod_{i} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m \neq k} \sum_{n \neq k} \frac{(\vec{r}_k - \vec{r}_m)(\vec{r}_k - \vec{r}_n)}{r_{km} r_{kn}} u'(r_{km}) u'(r_{kn}) \tag{B13}$$

And for the final term we take the second derivative of $u(r_{km})$

$$\left[\prod_{i} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m \neq k} \nabla_k^2 u(r_k m) \tag{B14}$$

which gives us with B5 and B8

$$\left[\prod_{i} \phi(\vec{r}_i)\right] \exp\left\{\sum_{j<l} u(r_{jl})\right\} \sum_{m \neq k} \left[u''(r_{km}) + \frac{2}{r_{km}} u'(r_{km})\right] \tag{B15}$$

Which gives us the final expression after dividing by the trial wavefunction

$$\begin{aligned}\frac{1}{\Psi(\vec{r})}\nabla_k^2\Psi(\vec{r}) = \frac{\nabla_k^2\phi(\vec{r}_k)}{\phi(\vec{r}_k)} + 2\frac{\nabla_k\phi(\vec{r}_k)}{\phi(\vec{r}_k)}\left(\sum_{j\neq k}^{N}\frac{\vec{r}_k-\vec{r}_j}{r_{kj}}u'(r_{kj})\right) \\ + \sum_{i\neq k}^{N}\sum_{j\neq k}^{N}\frac{(\vec{r}_k-\vec{r}_i)(\vec{r}_k-\vec{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \\ + \sum_{j\neq k}^{N}\left[u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right]\end{aligned} \tag{B16}$$

Then we will also find an expression for u' and u'', where $u(r_{kj}) = \ln f(r_{kj})$. First u'

$$u'(r_{kj}) = [\ln f(r_{kj})]' = \frac{1}{f(r_{kj})}f'(r_{kj}) = \frac{1}{1-\frac{a}{r_{kj}}} = \frac{1}{1-\frac{a}{r_{kj}}}\frac{a}{r_{kj}^2} = \frac{a}{r_{kj}(r_{kj}-a)} \tag{B17}$$

And the second derivative

$$u''(r_{kj}) = \frac{-a(r_{kj}(r_{kj}-a))'}{(r_{kj}(r_{kj}-a))^2} = \frac{a(a-2r_{kj})}{r_{kj}^2(r_{kj}-a)^2} \tag{B18}$$

Appendix C: Greens Function

To simplify the Green's function ratio 16 $\frac{G(x,y,\Delta t)}{G(y,x,\Delta t)}$, we can neglect the constant in front of the exponential and use $e^a/e^b = e^{a-b}$

$$\frac{G(x,y,\Delta t)}{G(y,x,\Delta t)} = \exp\left\{\frac{1}{4D\Delta t}(y-x-D\Delta tF(x))^2 - (x-y-D\Delta tF(y))^2\right\} \tag{C1}$$

Computing the squared terms we can see that a lot of the terms will disappear and we are left with

$$\frac{G(x,y,\Delta t)}{G(y,x,\Delta t)} = \exp\left\{\frac{1}{4D\Delta t}((D\Delta tF(x))^2 - (D\Delta tF(y))^2 + 2D\Delta t(F(x)+F(y))(x-y))\right\} \tag{C2}$$

Which we can finally simplify to

$$\exp\left\{\frac{1}{2}(F(x)+F(y))\left[\frac{1}{2}D\Delta t(F(x)-F(y)) - y + x\right]\right\} \tag{C3}$$