Ground state energy of hard sphere Bose gas in Elliptical HO potential by VMC methods

Adrian Martinsen Kleven

Autumn 2020

Contents

List of Figures

List of Tables

1   Abstract

In this paper, we are studying systems of interacting bosons in elliptical harmonic oscillator potentials using variational Monte- Carlo methods approximate the ground state energy. In the non- interacting case, the model gives the exact ground state energy as the analytic solution, with importance sampling reducing the number of Monte- Carlo steps needed to reach equilibration. For the interacting case we show that the particle number- normalized energy diverges for systems with different number of particles as expected from the interaction potential. We also look at how the Jastrow factor effects the distribution of bosons in the trap.

2   Introduction

When studying Bose- Einstein condensates in gases of alkali atoms confined in magnetic traps, many studies have used the Gross- Pitaevskii equation. These models are valid depending among other factors, on the mean atomic distance far exceeding the range for inter- atomic interactions. Barring this, we may see a tuning of the scattering length due to the so- called Feshbach resonance. To remedy this, we can use Variational Monte- Carlo methods to evaluate the ground state energy of a trapped, hard sphere Bose gas, using an ansatz for the wave function with a hard- shell interaction.

First we'll treat some of the theory behind the Metropolis algorithm and importance sampling, then how we're modelling the Hamiltonian and the choice of trial wave function, calculating the useful properties on the way.

We will then treat the non- interacting, spherical case as a proof of concept, as these have analytic solutions. Then we will look at replacing the analytical expression for the double derivative with numerical differentiation. We can then move on to the interacting case with an elliptical harmonic oscillator potential. The results will follow this same procedure. Then we will look at how the normalized energy levels diverge when we introduce the repulsive interaction and present the results for how the Jastrow factor impacts the distribution of bosons.

Finally we will discuss the success of the implementation based on the accuracy to the analytic case, and whether the results for the interacting case matched expectations. We will also discuss some of the many ways this program should have been improved.

3   Theory

3.1   The Variational method

One method for approximating the ground state of a quantum mechanical system, is the variational method. By choosing a wave function ansatz with some variational parameter, and then calculating the expectation value for the energy of that wave function, we can get an upper bound for the ground state by varying the variational parameter until the expectation value is the lowest possible.

The expectation value of the energy is given by

$$\langle E \rangle = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \tag{1}$$

this calculation however, becomes infeasible for systems with large numbers of particles. One solution is to use Variational Monte Carlo methods.

3.2   The Metropolis Algorithm

The Metropolis Algorithm we are using consists of just a few steps. We start by initializing the system with randomly distributed particles. We then iterate the following. We choose one

particle at random and adjust its position by some amount. We then calculate the acceptance probability according to equation (2) below

$$A\left(x \to x'\right) = \min\left(1, \frac{P\left(x'\right)}{P\left(x\right)} \frac{T\left(x \to x'\right)}{T\left(x' \to x\right)}\right) \tag{2}$$

where $P(x)$ is the probability distribution of the system with that particle in the position x and $T\left(x \to x'\right)$ is the transition- probability from a system with the particle in position x to a system with position x'. We then choose whether to accept or reject this move based on if this acceptance probability exceeds a uniformly distributed random number between 0 and 1. If $T\left(x \to x'\right)$ is symmetric under the interchange of x and x', we can cancel the two terms in equation (2). Additionally, if the wave function is separable (as it is in our non- interacting case), we can get away with simply calculating the wave function of the one particle before and after the move. Because we're basing the moves on a randomly distributed number, this algorithm has a low acceptance rate. There is a better way of deciding these moves.

3.2.1   Metropolis- Hastings

In place of the transition probability, the Metropolis- Hastings algorithm uses the Greens function solution to the Fokker- Planck equation (4) giving the new expression for the acceptance probability in equation (3)

$$A\left(x \to x'\right) = \min\left(1, \frac{P\left(x'\right)}{P\left(x\right)} \frac{G\left(x, x', \delta t\right)}{G\left(x', x, \delta t\right)}\right) \tag{3}$$

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} - F\right)P(x,t) \tag{4}$$

The Greens function in this case is

$$G\left(x, x', \delta t\right) = \frac{1}{(4\pi D\delta t)^{\frac{3}{2}N}} exp\left(-\frac{(x' - x - D\delta t F(x))^2}{4D\delta t}\right). \tag{5}$$

Luckily the prefactor, cancels in the expression for the acceptance probability. To propose a new move for a particle, we use the Langevin equation (6)

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

To get the new position we use a discrete time step δt which gives us equation (7) below.

$$x' = x + DF(x)\delta t + \eta\sqrt{\delta t} \tag{7}$$

the D in these equations is the diffusion coefficient, while η is a randomly distributed number between 0 and 1. $F(x)$ is the drift force, or quantum force which is calculated from the gradient of the trial wave function, see equations (21) and (22).

3.3   Hamiltonian

The bosons will be confined by a spherical (S) and later, an elliptical (E) harmonic oscillator, with the external potential being given by equation (8) below.

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \tag{8}$$

with ω_{ho}^2 defining the strength of the HO potential in the spherical case and the oscillating frequency in the xy- plane in the elliptic case, wherein ω_z^2 is the frequency perpendicular to the

xy- plane.

The Hamiltonian then, is given by

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{9}$$

where the interaction potential is given by

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{10}$$

where a is the hard- core diameter of the bosons.
Note that we're using the the following notation for the double sum over all pairwise interactions:

$$\sum_{i<j}^N V_{ij} \equiv \sum_{i=1}^N \sum_{j=i+1}^N V_{ij}. \tag{11}$$

3.4 Trial Wave Function

The trial wave function for the ground state is given by equation (12) below

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right], \tag{12}$$

with

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right]. \tag{13}$$

and

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{14}$$

where α and β are variational parameters. We choose $\beta = 1$ for the spherical harmonic oscillator. For non-interacting bosons, we have $a = 0$ (no Hard- core diameter) and $\alpha = 1/2a_{ho}^2$.

3.5 Non- interacting bosons in spherical HO

To benchmark our algorithm, we work with a simplified model with no interactions, and a symmetric potential. Here we can find analytic expressions for the local energy using the expression in equation (15) and our simplified trial wave function (16).

$$E_L(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} H \Psi_T(\mathbf{r}), \tag{15}$$

$$\Psi_T(\mathbf{r}) = \exp\left[-\alpha \sum_i^N \mathbf{r}_i^2\right]. \tag{16}$$

Using natural units of $\hbar = m = 1$ we find that our simplified Hamiltonian is

$$H = \sum_i^N \left(-\frac{1}{2}\nabla_i^2 + V_{ext}(\mathbf{r}_i) \right). \tag{17}$$

5

Thankfully, this simplified trial wave function is separable. We find that

$$H\Psi_T(\mathbf{r}) = \left(\left(\frac{1}{2}\omega_{ho}^2 - 2\alpha^2\right)\sum_i^N \mathbf{r}_i^2 + ND\alpha\right)\Psi_T(\mathbf{r}) \tag{18}$$

where D is the number of dimensions. This gives us a local energy of

$$\left(\frac{1}{2}\omega_{ho}^2 - 2\alpha^2\right)\sum_i^N \mathbf{r}_i^2 + ND\alpha \tag{19}$$

which, when inserting the correct expression for $\alpha = \frac{1}{2a_{ho}^2} = \frac{\omega}{2}$, gives

$$E_L(\mathbf{r}) = \frac{DN\omega_{ho}}{2} \tag{20}$$

3.5.1 Drift Force

For importance sampling, we're also going to need an expression for the Drift Force on each particle. This is given by the expression in equation (21) below.

$$\mathbf{F}_i = \frac{2\nabla\Psi_T}{\Psi_T} = -4\alpha\mathbf{r}_i \tag{21}$$

3.6 Interacting bosons in elliptical HO

We have found the necessary results for benchmarking our program. It now remains to extend these results to the complete problem. Towards this effort, we need to obtain the gradient and Laplacian for the trial wave function. The derivation can be found in Appendix A [7]. We find that the expression for the gradient is

$$\nabla_k\Psi_T(\mathbf{r}) = \nabla_k\phi(\mathbf{r}_k)\left[\prod_{i\neq k}\phi(\mathbf{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right) \tag{22}$$

$$+ \left[\prod_i \phi(\mathbf{r}_i)\right]\exp\left(\sum_{j<m}u(r_{jm})\right)\sum_{l\neq k}\nabla_k u(r_{kl}), \tag{23}$$

and using the expression for the normalized Laplacian:

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

we find that

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

$$+ \sum_{i\neq k}\sum_{j\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj})$$

$$+ \sum_{j\neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right)$$

with

$$u'(r_{kj}) = \frac{r_{kj}}{r_{kj} - a} \tag{24}$$

and

$$u''(r_{kj}) = \frac{a^2 - 2r_{kj}}{r_{kj}^2(r_{kj} - a)^2} \tag{25}$$

for $r_{kj} > a$.

Moving on, we now scale the Hamiltonian to better suit numerical implementation. We will fix, as in Refs. [1] and [3], $a/a_{ho} = 0.0043$ and introduce lengths in units of a_{ho}, $r^2 \leftarrow r^2/a_{ho}^2 = \frac{m\omega_{ho}}{\hbar}r^2$ and energy in units of $\hbar\omega_{ho}$. The full Hamiltonian for an elliptic HO is

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{26}$$

from which we can factor out the unit energy to get

$$\hbar\omega_{ho} \sum_i^N \left(\frac{-\hbar}{2m\omega_{ho}}\nabla_i^2 + \frac{m\omega_{ho}}{2\hbar}[x^2 + y^2 + \frac{\omega_z^2}{\omega_{ho}^2}z^2] \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{27}$$

now, introducing the natural length scale, we get

$$\hbar\omega_{ho} \sum_i^N \left(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}[x^2 + y^2 + \gamma^2 z^2] \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \tag{28}$$

with $\gamma = \frac{\omega_z}{\omega_{ho}}$ and the interaction potential, now reading

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a/a_{ho} = 0.0043 \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a/a_{ho} = 0.0043 \end{cases} \tag{29}$$

we have

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

We choose the initial value for $\beta = \gamma = 2.82843$ and compute ground state energy using α as the only variational parameter.

3.7 Finding the optimal variational parameter

We don't want to run many Monte- Carlo steps for values of α far away from the optimal value. To remedy this, we implement steepest descent. To do this, we need an expression for the gradient of the expectation value of the local energy with respect to α. We have

$$\frac{\partial \langle E \rangle}{\partial \alpha} = 2\left(\langle E \frac{1}{\Psi_T(\mathbf{r})}\frac{\partial \Psi_T(\mathbf{r})}{\partial \alpha} \rangle - \langle E \rangle \langle \frac{1}{\Psi_T(\mathbf{r})}\frac{\partial \Psi_T(\mathbf{r})}{\partial \alpha} \rangle \right) \tag{30}$$

where the expectation values are the sample mean of values sampled after the algorithm has reached equilibration. We can the update α according to $\alpha' = \alpha - \gamma\frac{\partial \langle E \rangle}{\partial \alpha}$, where γ is the learning rate.

Rather than implementing a learning schedule, we can simply start with a larger learning rate, then run gradient descent until the system starts oscillating between two gradients of roughly equal magnitude, then restart the gradient descent at that α with a smaller learning rate.

4 Implementation

4.1 Sampling

There is no implementation for determining when the equilibration stage of the algorithm should terminate. Instead, the algorithm spends some fraction of the number of sampling steps to equilibrate before sampling begins. This is obviously not ideal, as the sampling will eventually begin whether or not the system has equilibrated.

4.2 Numerical Derivative

The numerical double derivative was calculated using Newton's forward difference formula with a step size of 0.0001. The numerical double derivative was only implemented for the uncorrelated system in order to determine whether or not it would yield usable results.

4.3 Post Analysis and Error Estimation

The post analysis involved printing the energy at each sampling stage to file, then analysing it using Python. The error estimates for the expectation value of the local energy were evaluated using Marius Jonsson's Blocking code [2].

4.4 Parallelization

Building a solution for c++ in Visual Studio in x- 64 release mode, automatically vectorizes and parallelizes many loops that don't have dependencies that can't be parallelized. The only loop to be explicitly parallelized in the program was a grid search in the α parameter space, with each thread handling a unique value of α. This loop was parallelized using OpenMP [4].

5 Results

The actual energy of the system is not the interesting part, if we instead look at the energy normalized by the number of particles in the system the differences between the correlated and uncorrelated system become much more apparent.

5.1 Non- interacting particles in SHO

5.1.1 Standard Metropolis Sampling

In figure 1 below, we see an example of what happens when the algorithm starts sampling before the system has equilibrated.
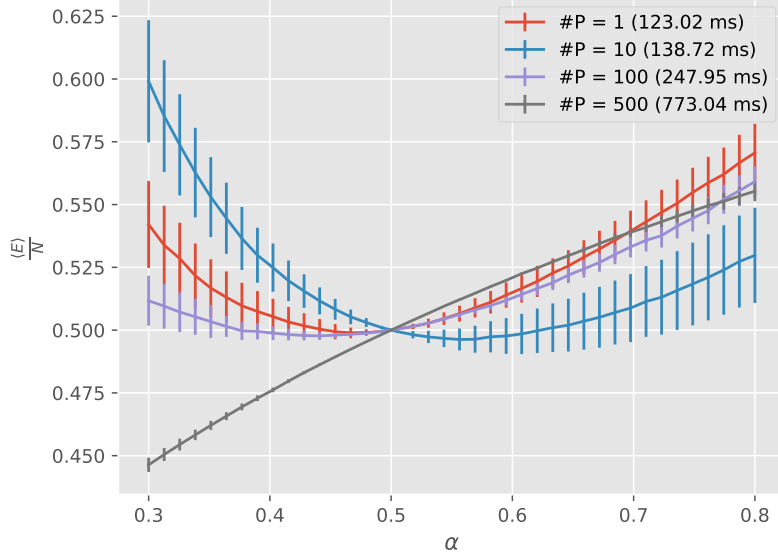
Figure 1: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 1 dimension, using 2^{18} MC- steps

We see that the systems with many particles no longer have a clear inflexion point at $\alpha = 0.5$. This is fixed by simply increasing the number of steps the algorithm runs before beginning sampling. As we see in figure 2 below
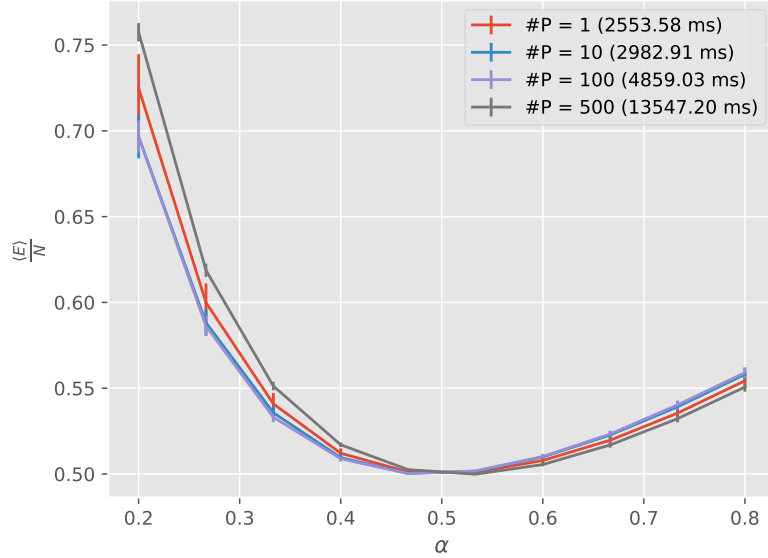

Figure 2: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 1 dimension, using 2^{22} MC- steps

we now have a clear inflexion point at $\alpha = 0.5$ with much smaller error bars. Additionally, the lines are much closer together, as they should be for an uncorrelated system. We also see that the lowest normalized energy is 0.5. This is correct according to the analytical result for a 1- dimensional system with $\omega_{ho} = 1$ (see equation (20)).

We have the same findings for the 2-, and 3- dimensional systems in figures 3 and 4 below.
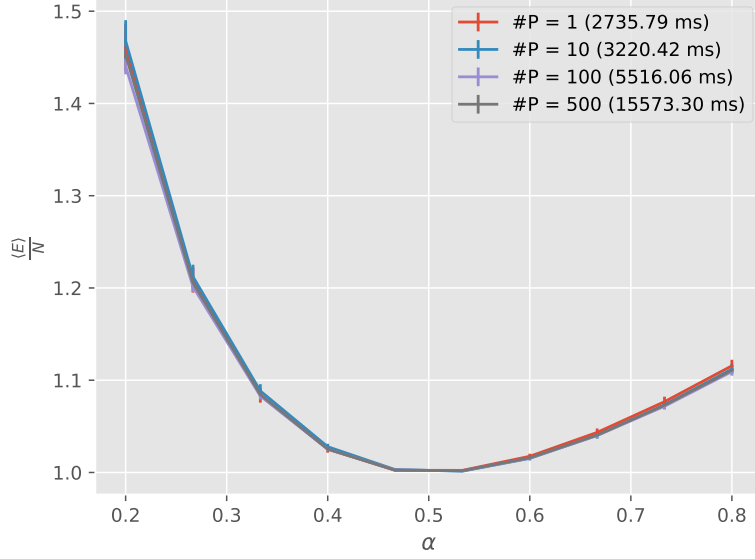
Figure 3: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 2 dimensions, using 2^{22} MC- steps
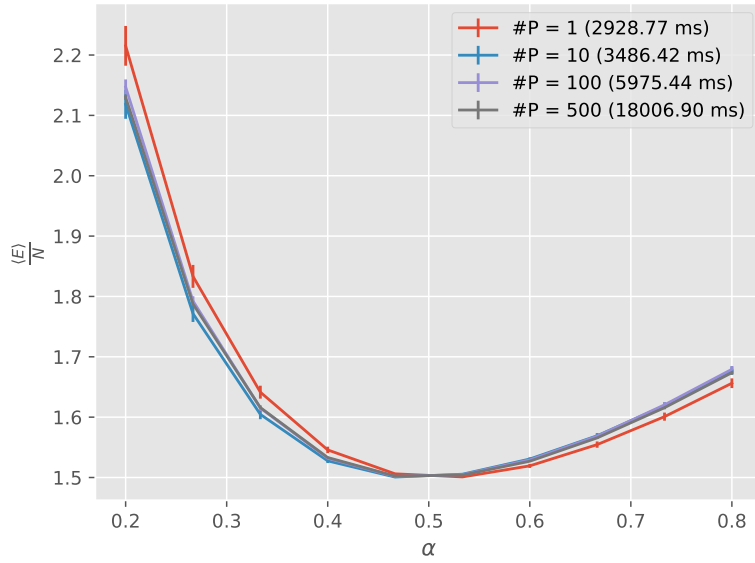

Figure 4: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 3 dimensions, using 2^{22} MC- steps

5.1.2 Analytic v. numerical derivative

For the uncorrelated system, using the numerical double derivative for calculating the kinetic energy, we find that performing the same computation takes up to 115 times as long for 500 particles in 3 dimensions using 2^{22} steps.

In figure 5 below, we see that the lowest normalized energy of 0.025 happens at $\alpha = 0.6$ rather than 0.5. This result is off by an order of magnitude.
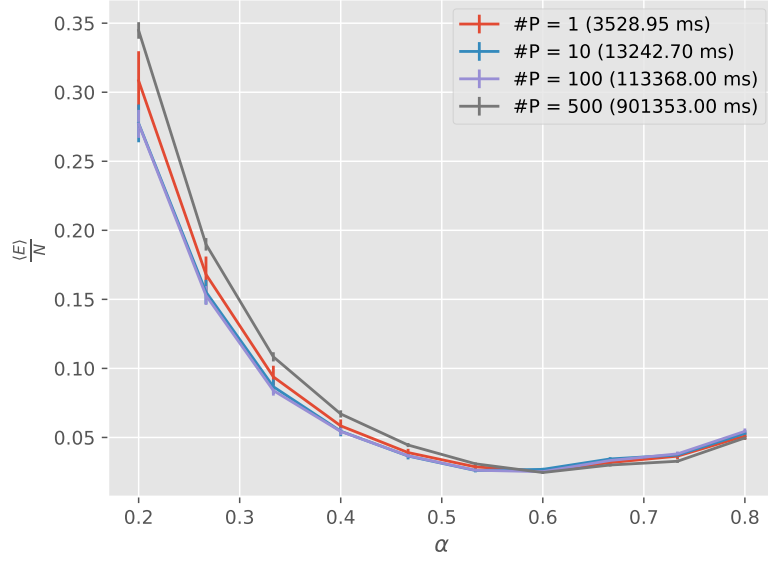
Figure 5: The particle number- normalized expectation value for the energy as a function of the variational parameter α.For 1 dimension, using 2^{22} MC- steps and numerical differentiation

In figure 6 below, we again see that the energy is off by and order of magnitude while the energy minimum has shifted closer to $\alpha = 0.5$
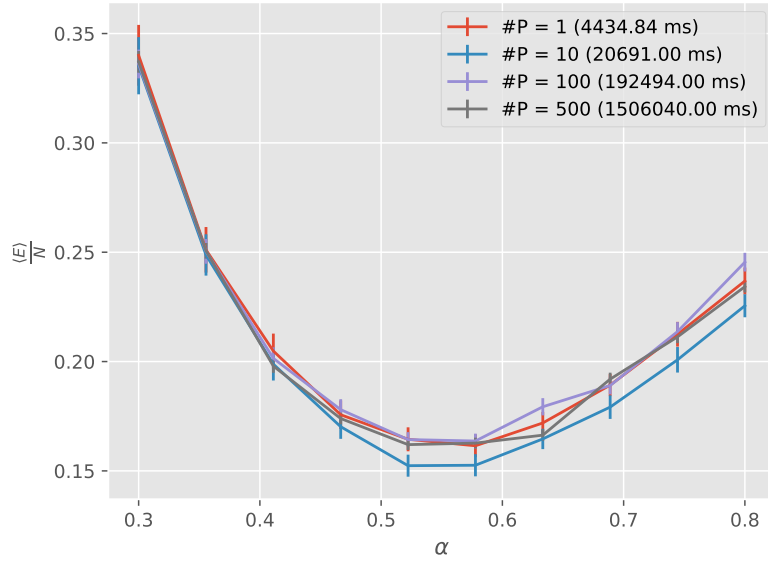

Figure 6: The particle number- normalized expectation value for the energy as a function of the variational parameter α.For 2 dimensions, using 2^{22} MC- steps and numerical differentiation

In figure 7 below, we see that the energy is now only slightly off while the energy minimum is nearly happening at $\alpha = 0.5$
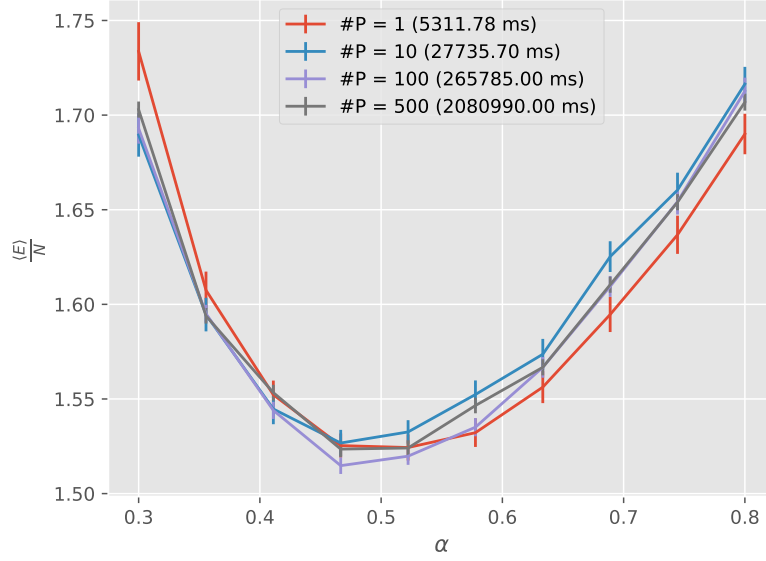
Figure 7: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 3 dimensions, using 2^{22} MC- steps and numerical differentiation

5.1.3 Importance Sampling

As we can see in figures 8,9 and 10 below the addition of importance sampling on the uncorrelated systems gives the same minimums as brute force sampling, but the lines are now much closer together, indicating that the system has reached equilibration faster as they use the same number of Monte Carlo steps.
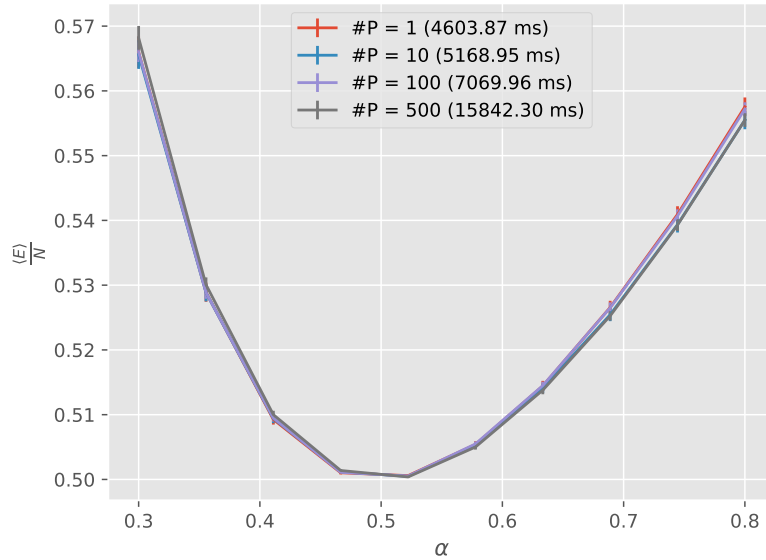

Figure 8: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 1 dimension, using 2^{22} MC- steps with importance sampling

12

Figure 9: The particle number- normalized expectation value for the energy as a function of the variational paramter α.For 2 dimensions, using 2^{22} MC- steps with importance sampling
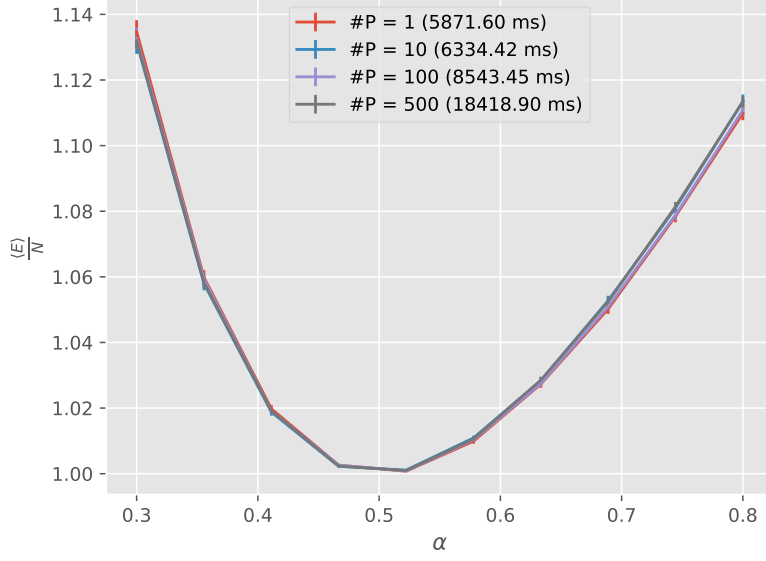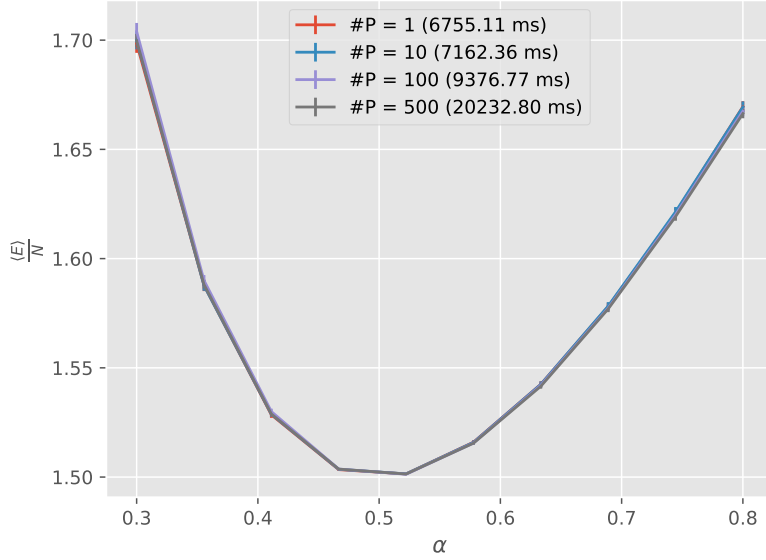

Figure 10: The particle number- normalized expectation value for the energy as a function of the variational parameter α.For 3 dimensions, using 2^{22} MC- steps with importance sampling

5.1.4 Importance Sampling Time Step Dependence

To study the effect of the time step δt in equation (7), we plot the proportion of the proposed steps that are accepted according to the Metropolis- Hastings algorithm as a function of the time step size. As we see in figure 11 below, the acceptance rate is highest for values closest to zero, with the smallest time step being 10^{-4}. These small time steps come at the cost of slower equilibration as the particles take longer to travel. Clearly then, there is an optimal value for this time step as we want long enough to be efficient but also small enough to have a high acceptance rate.
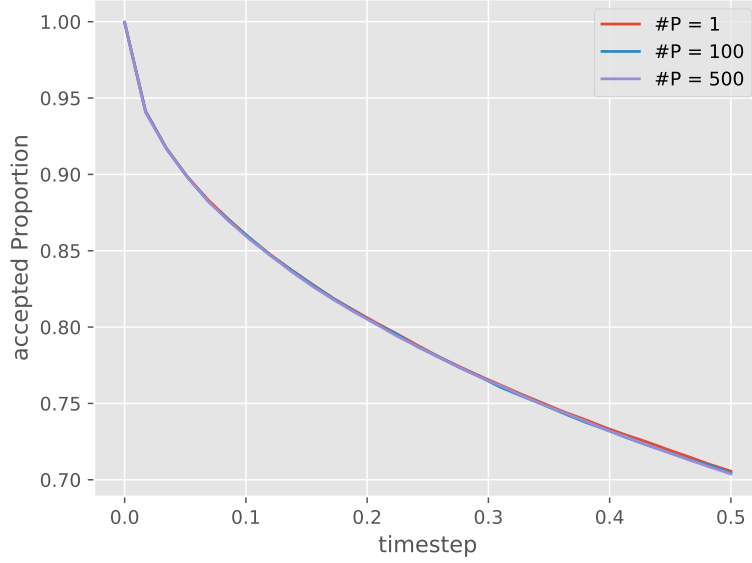
Figure 11: The proportion of accepted moves as a function of the size of the time step. For 1 dimension, using 2^{18} MC- steps with importance sampling

5.2 Interacting particles in EHO

For the interacting particles in the elliptical harmonic oscillator, we finally see that the normalized energies are diverging. As we can see, in figure 12 the normalized energy levels increase with increasing particle number. This is expected, as more particles interacting, will increase the energy.
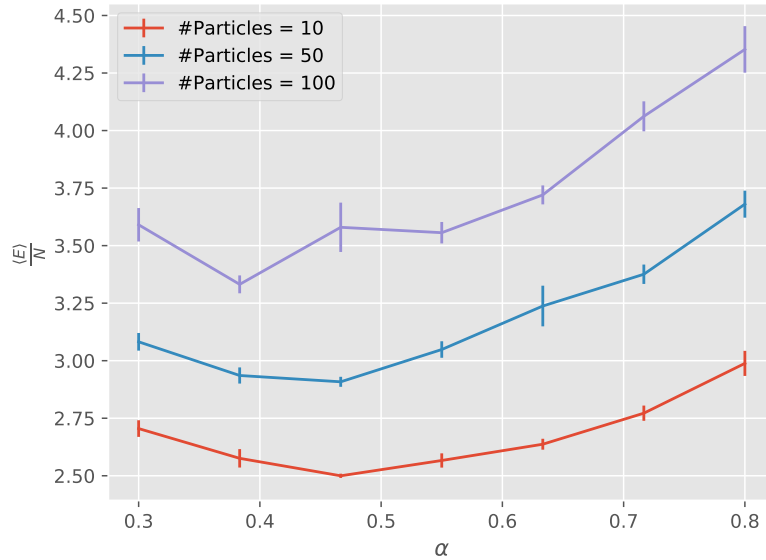

Figure 12: The particle number- normalized expectation value for the energy as a function of the variational parameter α. For 3 dimensions, using 2^{18} MC- steps for the corrolated system in an elliptical HO- potential with $\gamma = \beta = 2.82843$.

5.2.1 Steepest Descent Result

Because large systems take so long to simulate, there hasn't been time to complete a gradient descent analysis for systems with 50 and 100 particles. Instead we have analyses for systems

with 10, 20 and 30 particles to analyse how the values for the optimal α changes. Because we're searching the parameter space with a fixed step- size, we can provide an upper limit for the error in the optimal value for α.

Nr. Particles	$\langle E \rangle$	α
10	25.3222	0.42 ± 0.000909
20	53.8389	0.362182 ± 0.00109
30	81.0876	0.402727 ± 0.001363

Table 1: Correlated EHO potential, upper bound for the error in provided by doing a grid search around α and seing that the energy increases in both directions.

Here we see the same trend that appeared in figure 12, that the optimal value for α looks to shift towards a lower value, for systems with more particles.

5.2.2 Time expenditure

While the computation time in the non- interacting case has a nearly linear relationship with the number of particles, this is not the case in the case of interacting particles. We can see in figure 13 a line of best fit through the time spent calculating a single data point for different numbers of particles.
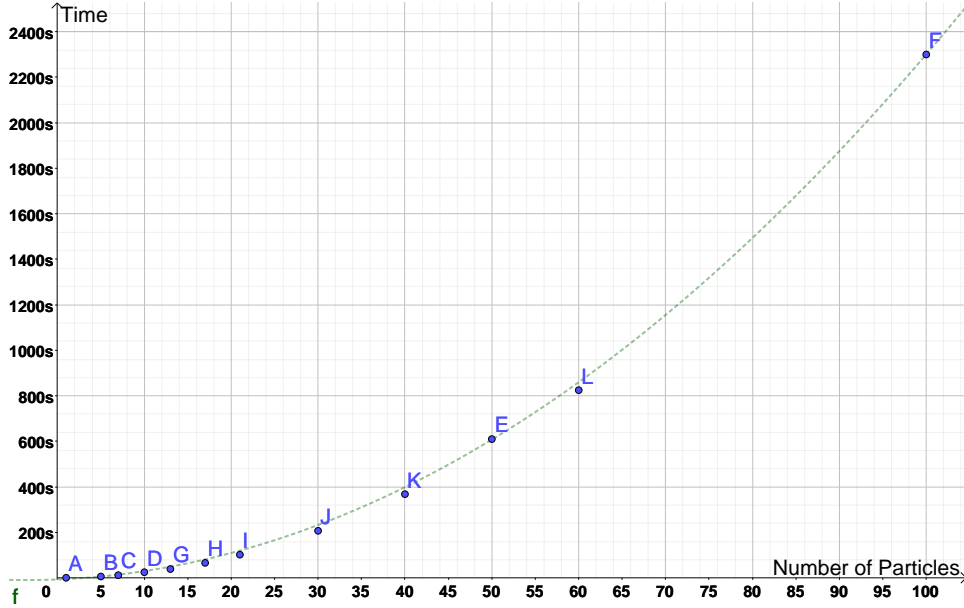

Figure 13: Time expenditure per datapoint in seconds v. the number of particles for a correlated system with an elliptical HO- potential using 2^{18} MC- steps.
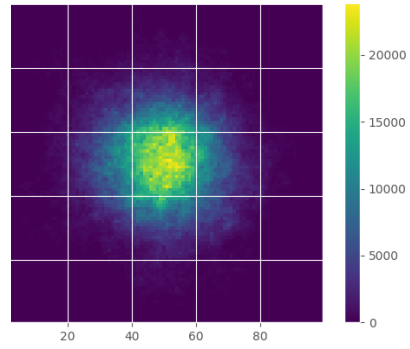
the relationship turns out to be a 2'nd degree polynomial.

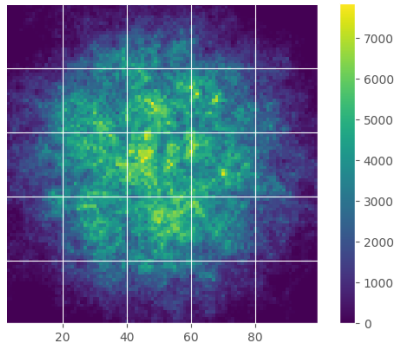## 5.3 The Jastrow factor and The Distribution of Bosons

By changing the so- called Hard- Core Diameter of the bosons, we can influence the Jastrow factor and study its effects. We see the results of this in figure 14 below.
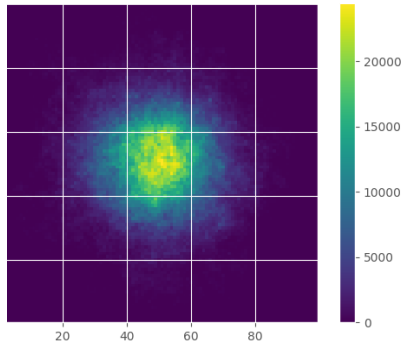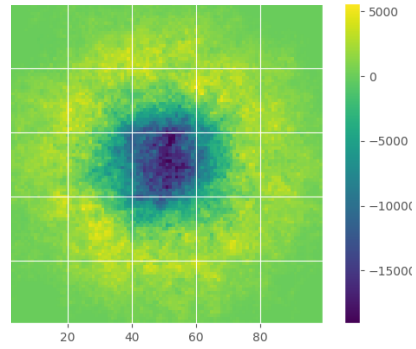
(a) Hard- Core diameter = 0

(b) Hard- Core diameter = 0.00433

(c) Hard- Core diameter = 0.433

(d) Figure 14c minus Figure 14b

Figure 14: The distribution of bosons projected onto the xy- plane using 20 particles in three dimensions, $\alpha = 0.5$ and 2^{20} Monte- Carlo steps. The Hard- Core Diameter is 0.00433 as per [1] and [3]. Each grid- line corresponds to 1.2 units of distance.

.

As we can see, between figures 14a and 14b the difference is nearly imperceptible, with only a slight dimming of the central spot. The difference becomes much more noticeable between figures 14b and 14c with their difference shown in figure 14d. The bosons become much more spread out, as clustering together takes more energy when the Jastrow factor is higher.

6 Conclusion

Based on the success of the program in the non- interacting case we can conclude that both the Metropolis Algorithm and The Metropolis- Hastings algorithm were correctly implemented. The implementation of numerical differentiation had unstable results when varying the step length and there wasn't enough time to find an optimal step length for the different systems. This implementation was largely a failure, as it yielded results that were wrong by an order of magnitude. The difference in time- expenditure was substantial, proving that analytical solutions should be applied where possible.

We saw that importance sampling greatly contributed in reducing the error in the energy calculation, as the system equilibrated earlier.
The implementation of the full correlated system in the elliptical harmonic oscillator showed that there is an additional gain in energy with more particles beyond the the simple multiplicative factor coming from multiplying the one- body problem. The optimal value for α seemed to

shift to lower values in the systems with the most particles.

There are several things that should be implemented and improved upon. The number of equilibration steps should not be decided by some arbitrary number. We could instead sample a running mean and start sampling only when the variance falls below a desired value, then sample for some number of steps that are a multiple of two, so as to work with Blocking. The Energies obtained through gradient descent, need error estimates but again, there wasn't time. With a way to classify the end of equilibration, we could study the time to reach equilibration v. the length of the time step in order to obtain an optimal value for the time step, as it's a trade-off between the acceptance rate and the time taken to reach equilibration.

More of the code should have been optimized for parallelization, as currently there are a lot of dependencies between classes that prohibit the compiler from vectorizing and parallelizing the code.

7   Appendix A

We rewrite

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

as

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

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

$$f(r_{ij}) = \exp\left(u(r_{ij}) \right),$$

with $u(r_{ij}) = \ln f(r_{ij})$. We also write

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

Starting with the gradient, we get

$$\nabla_k \Psi_T(\mathbf{r}) = \exp\left(\sum_{j<m} u(r_{jm}) \right) \nabla_k \left[\prod_i \phi(\mathbf{r}_i) \right] + \left[\prod_i \phi(\mathbf{r}_i) \right] \nabla_k \exp\left(\sum_{j<m} u(r_{jm}) \right) \tag{31}$$

solving each gradient separately, we get

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

and

$$\nabla_k \exp\left(\sum_{i<k} u(r_{ik}) \right) = \exp\left(\sum_{j<m} u(r_{jm}) \right) \sum_{l\neq k} \nabla_k u(r_{kl})$$

17

giving us

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

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

From here, we need to find the Laplacian. We accomplish this by taking the scalar product of the gradient we just got, with the gradient operator. Solving each term in turn, we get

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

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

and for the second term,

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

$$
\left[\prod_i \phi(\mathbf{r}_i) \right] \exp \left(\sum_{j<m} u(r_{jm}) \right) \left(\sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \right) \sum_{l \neq k} \nabla_k u(r_{kl}) +
$$

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

We can now resolve the last of the gradients while dividing every term by the trial wave function, this in turn, gets us the final expression for the normalized Laplacian of the trial wave function

$$
\frac{1}{\Psi_T(\mathbf{r})} \nabla_k^2 \Psi_T(\mathbf{r}) = \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left(\sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}} u'(r_{kj}) \right) \tag{32}
$$

$$
+ \sum_{i \neq k} \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \tag{33}
$$

$$
+ \sum_{j \neq k} \left(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right). \tag{34}
$$

References

[1] J. L. DuBois and H. R. Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Physical Review A*, 63(2), Jan 2001.

[2] Marius Jonsson. Standard error estimation by an automated blocking method. *Phys. Rev. E*, 98:043304, Oct 2018.

[3] 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. *Physical Review A*, 71(5), May 2005.

[4] OpenMP Architecture Review Board. OpenMP application program interface version 3.0, May 2008.