

Project 1: Simulation of hard-sphere Bose gas using variational Monte Carlo

Author:

Elisabeth CHRISTENSEN

Author:

Håkon D. FOSSHEIM

April 7, 2020

In this project we apply the variational Monte-Carlo method to find the ground state of a dilute hard-sphere bose gas trapped in a harmonic potential. The method is first tested against the closed-form expression for a non-interacting system with a spherically symmetric potential by use of the Metropolis algorithm. It is then expanded to include two-body interactions in an elliptical potential by use of the Metropolis-Hastings algorithm. This is done in one, two and three dimensions for up to 500 particles. The results are benchmarked against [cite article] (and found to be in agreement for...). Due to correlated measurements, the blocking method is applied in order to present a proper error analysis. The optimal variational parameter for the trial wavefunction is calculated analytically to be $\alpha = 0.5$ for the non-interacting spherically symmetric system. No closed-form solution is obtainable for the elliptical system and so we resort to the steepest descent method to find the value of α that minimizes local energy. Lastly, we plot the one-body density of the interacting and non-interacting system and find that it qualitatively fits our theoretical model.

I. Introduction

A Bose-Einstein condensate is a state of matter achieved by cooling spin-1 particles, viz. bosons, to very low temperatures, causing them to "condense" into the lowest accessible quantum state. Here we present an upper bound for this ground state for a bose gas consisting of alkali atoms confined in a magnetic trap. We do so using a method known as *Variational Monte Carlo*.

Variational Monte Carlo is a computational method utilizing the Monte Carlo algorithm combined with the variational principle from quantum mechanics to approximate an upper bound on the ground state energy of a system. We will compute this through mainly two algorithms: the Standard Metropolis and the Metropolis-Hastings algorithm. Furthermore, by building on the algorithms mentioned, we can develop a more complex quantum mechanical system consisting of either inter-

acting or non-interacting bosons in a coordinate system of our choosing.

The variational principle relies heavily on the wavefunction necessary to describe our system, i.e. the Hamiltonian. Now, if we're clever about it, we can make an educated guess which allows us to find a normalized trial wavefunction that describes our system fairly well. However, by still making some room for adjustments, we can reach an even lower upper bound on the ground state energy if need be. This can be done by introducing a variational parameter, thereby the name variational principle.

But what if we don't want to go through all possible values for our variational parameter by hand? Well, that's easy, we can just include values that will ensure to *minimize* our energy. Computationally, we can perform a gradient descent, converging into the most optimal parameter for the *tightest* bound on the expectation value of the energy.

We can see the effects of interacting bosons vs. non-interacting bosons by studying the one-body density in an ellipsoidal system. We will lastly compare these effects to that of an ideal, i.e. spherical, system.

II. Theory

A. Variational Monte Carlo

When dealing with complicated, intricate systems, such as many-body quantum mechanical problems, we often encounter large integrals concerning the behaviour of the objects within the system. An example of this, which will be the main focus of our report, is the expectation value of a given operator \hat{O} when considering a system consisting of N particles dependent on a wavefunction Ψ ,

$$\langle \hat{O} \rangle = \frac{\int d\mathbf{r}_1 d\mathbf{r}_2 \dots d\mathbf{r}_N \Psi^\dagger(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) \hat{O} \Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)}{\int d\mathbf{r}_1 d\mathbf{r}_2 \dots d\mathbf{r}_N \Psi^\dagger(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) \Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)}. \quad (1)$$

Replacing the arbitrary quantum mechanical operator \hat{O} with the more concrete operator \hat{H} , i.e. the Hamiltonian of the system, we can find the expectation value of the ground state energy. However, what if the exact form of the wavefunction is unknown? This is where the variational principle comes into play, as it allows us to find the *upper bound* for the ground state energy of a system. That is, we guess a form of the wavefunction which satisfies the criteria to the system in question, and calculate the ground state energy E_{gs} as a function of the Hamiltonian. This is what we will apply to our system of the hard-sphere Bose gas. The variational principle is based on the claim that

$$E_{gs} \leq \langle \Psi_T | \hat{H} | \Psi_T \rangle \equiv \langle \hat{H} \rangle, \quad (2)$$

where Ψ_T is the trial wavefunction and is considered to be normalised. The Hamiltonian can be expressed as

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

1. Local energy From the definitions of the Hamiltonian and the trial wavefunction we can declare a new operator \hat{E}_L , also known as the local energy,

$$\hat{E}_L(\mathbf{r}; \alpha) = \frac{1}{\Psi_T(\mathbf{r}; \alpha)} \hat{H} \Psi_T(\mathbf{r}; \alpha). \quad (4)$$

The expected value of the local energy can be found by multiplying the integrand in the numerator of eq. (1) by Ψ/Ψ :

$$\langle \hat{H} \rangle = \frac{\int d\mathbf{r} \Psi_T^\dagger(\mathbf{r}; \alpha) \Psi_T(\mathbf{r}; \alpha) \frac{1}{\Psi_T(\mathbf{r}; \alpha)} \hat{H} \Psi_T(\mathbf{r}; \alpha)}{\int d\mathbf{r} \Psi_T^\dagger(\mathbf{r}; \alpha) \Psi_T(\mathbf{r}; \alpha)} \quad (5)$$

$$= \frac{\int d\mathbf{r} E_L(\mathbf{r}; \alpha) |\Psi_T(\mathbf{r}; \alpha)|^2}{\int d\mathbf{r} |\Psi_T(\mathbf{r}; \alpha)|^2} = \langle \hat{E}_L \rangle \quad (6)$$

Which may be rewritten as

$$\langle \hat{E}_L \rangle = \int d\mathbf{r} E_L(\mathbf{r}; \alpha) \rho_T(\mathbf{r}; \alpha) \quad (7)$$

where

$$\rho_T(\mathbf{r}; \alpha) = \frac{|\Psi_T(\mathbf{r}; \alpha)|^2}{\int d\mathbf{r} |\Psi_T(\mathbf{r}; \alpha)|^2} \quad (8)$$

Finding a closed-form expression for $\langle \hat{E}_L \rangle$ is in general not possible when dealing with interacting many-body systems. One must therefore resort to numerical integration. For a fixed step length, the truncation error in integration methods based on Taylor expansion goes as $N^{-k/d}$ where N is the number of grid points, d is the dimensionality of the system and $k \geq 1$ is an integer that depends on the method of choice. If we want to go from a one dimensional integral to a d dimensional one whilst keeping the truncation error fixed, we must compensate by increasing the number of grid points by a factor of $N^{(d-1)}$. This quickly results in absurd run-times. In contrast, the truncation error in Monte-Carlo integration scales as $\sigma \sim 1/\sqrt{N}$ for all dimensions.

The idea behind Monte-Carlo integration is to estimate the integral by taking the sum of a large set of samples randomly distributed in phase-space with weights corresponding to the underlying PDF

$$\langle \hat{E}_L \rangle = \int d\mathbf{r} E_L(\mathbf{r}; \alpha) \rho(\mathbf{r}; \alpha) \approx \sum_{i=1}^N E_L(\mathbf{r}; \alpha) \rho(\mathbf{r}; \alpha) \quad (9)$$

Because we are unable to perform the integral in equation (8) analytically, we do not know the normalization

constant of the trial PDF. This is where the Metropolis algorithm comes to our rescue. We pick a random starting position and evaluate E_L . We then propose a new move in an arbitrary direction, and accept it if

$$1 < \frac{\rho_T^{new}}{\rho_T^{old}} \quad (10)$$

and reject it with a 50 % probability otherwise.¹ In this way, we can sample $E_L(\mathbf{r}; \alpha)$ directly from $\rho_T(\mathbf{r}; \alpha)$ without having to calculate the normalization constant. Now the likelihood of getting a specific $E_L(\mathbf{r}; \alpha)$ is given by $\rho_T(\mathbf{r}; \alpha)$ and so there is no need for the weights in equation (9).

$$\langle \hat{E}_L \rangle \approx \frac{1}{N} \sum_{i=1}^N E_L(\mathbf{r}; \alpha) \quad (11)$$

Here, the wavefunction is dependent on a variational parameter α , which allows us to experiment which wavefunction that will give us the best estimate on the minimal value of the local energy.

B. Systems

The system we will be focusing on throughout this report is that of a harmonic oscillator. From eq. (3) the external potential $V_{onebody}$ can then be expressed in the cases of a spherical (S) and an elliptical (E) harmonic trap as

$$V_{onebody}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & \text{(S)} \\ \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] & \text{(E)} \end{cases} \quad (12)$$

where ω_{ho}^2 is the trap potential strength. The repulsive potential V_{int} is expressed as

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (13)$$

simulating the interaction between the particles. Here, a is the hard-core diameter of the bosons, so that it is not possible for the bosons to penetrate each other.

1. Trial wavefunction The trial wavefunction must be able to satisfy the criteria following that of a harmonic oscillator. One criterion is that the wavefunction, including its derivative, must be well-defined at the origin. That is, at the boundary condition of when a

boson approaches zero ($r_i \rightarrow 0$), the wavefunction must converge. In such a case, we can require that

$$\Psi_T \propto e^{-Zr_i} \quad (14)$$

However, there is also the criterion of interaction between the particles, which can be incorporated in a so-called *cusp* condition. In other words, we must take into consideration the distance r_{ij} between the particles. The generalized trial wavefunction for N particles can then be expressed as

$$\begin{aligned} \Psi_T(\mathbf{r}) &= \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)\dots\phi(\mathbf{r}_N) \prod_{j < k} f(r_{jk}) \\ &= \left[\prod_i \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)] \right] \left[\prod_{j < k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right] \end{aligned} \quad (15)$$

Here, $\phi(\mathbf{r}_i)$ corresponds to the single-particle wavefunction of particle i with a Gaussian shape. The product over $f(r_{jk})$ is also known as the Jastrow factor of the wavefunction, and takes into account the hard-sphere diameter of the bosons. This factor is expressed as

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (16)$$

2. Local energy The analytic expression for the local energy for N non-interacting particles in d dimensions can be deduced from the trial wavefunction and the Hamiltonian as defined above², i.e.

$$\begin{aligned} E_L(\mathbf{r}) &= \frac{1}{\Psi_T(\mathbf{r})} \hat{H} \Psi_T \\ &= \frac{1}{\Psi_T} \sum_i^N \left[\frac{-\hbar^2}{2m} 2\alpha(2\alpha r_i^2 - d) \Psi_T + \frac{1}{2} m \omega_{ho}^2 r_i^2 \Psi_T \right] \end{aligned} \quad (17)$$

3. Drift force The drift force F as experienced by a particle can be expressed as the derivative of the trial wavefunction³, i.e.

$$F = \frac{2\nabla \Psi_T}{\Psi_T} = -4\alpha \mathbf{r}_i \quad (18)$$

While for the interacting case we have that the double derivative in (3), when acting on the trial wavefunction,

¹ See sections III A 1 and III A 2 under III. Methods for further explanation on the methods used to achieve this computationally.

² See section B under appendices for the derivation of this expression.

³ See section B under appendices for the derivation of eqs. (18) and (19).

is

$$\begin{aligned} \frac{1}{\Psi_T(\vec{r})} \nabla_k^2 \Psi_T(\mathbf{r}) &= \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} u'(r_{kj}) \\ &+ \sum_{i \neq k} \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \\ &+ \sum_{j \neq k} \left(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right). \end{aligned} \quad (19)$$

4. One-Body Density For a system of N particles, the one-body density of particle i is

$$\rho(\mathbf{r}_i) = \int |\psi(\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_N)|^2 d\mathbf{r}_1 \cdots d\mathbf{r}_{i-1} d\mathbf{r}_{i+1} \cdots d\mathbf{r}_N \quad (20)$$

And is the probability of finding particle i within the volume $d\mathbf{r}_i^3$. The likelihood of finding the particle anywhere is one, and so

$$\int \rho(\mathbf{r}_i) d\mathbf{r}_i^3 = 1 \quad (21)$$

The one-body density is the same for every particle, and thus

$$\sum_{i=1}^N \int \rho(\mathbf{r}_i) d\mathbf{r}_i^3 = \int \rho(\mathbf{r}) d\mathbf{r}^3 = N \quad (22)$$

Where we defined $\rho(\mathbf{r}) \equiv \sum_{i=1}^N \rho(\mathbf{r}_i)$.

In the case of an elliptical potential, the single particle wavefunction $\psi \propto e^{-\alpha(x^2+y^2+\beta z^2)}$ which gives $\rho(\mathbf{r}) \propto e^{-2\alpha(x^2+y^2+\beta z^2)}$. $\rho(\mathbf{r})$ is therefore constant on the surfaces of ellipsoids. For $\beta > 1$ the ellipsoids are oblate spheroids (disc-shaped) and for $\beta < 1$, they are prolate spheroids (cigar-shaped). In either case, the following holds

$$x^2 + y^2 + \beta z^2 = R^2 \quad (23)$$

Where R^2 is the radius of the projection of the ellipsoid onto the xy-plane, i.e. $R^2 = (x^2 + y^2)|_{z=0}$. We can then perform the integral in equation (22) by integrating $\rho(R)$ with respect to ellipsoidal shells of thickness dR , see figure 1.

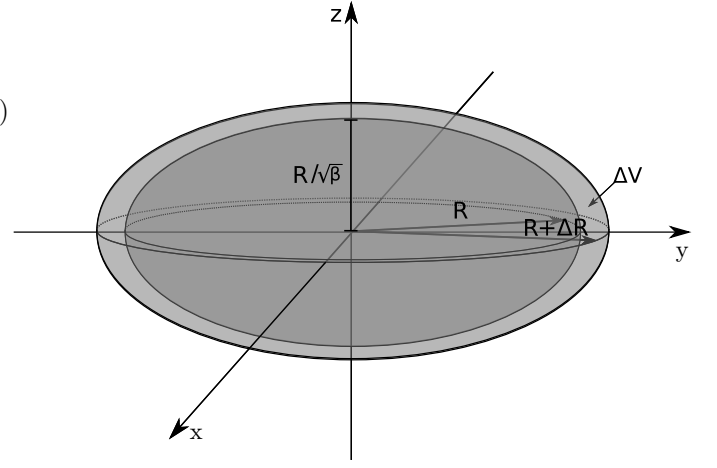


FIG. 1.

The volume of our ellipsoid is given by

$$V = \frac{4\pi}{3\sqrt{\beta}} R^3 \quad (24)$$

And so

$$dV = \frac{4\pi}{3\sqrt{\beta}} ((R + dR)^3 - (R)^3) \quad (25)$$

Equation (22) then becomes

$$N = \int \rho(\mathbf{r}) d\mathbf{r}^3 = \int_0^\infty \rho(R) dV \quad (26)$$

We can discretize this integral. Let R_{max} be large enough that $\rho(R_{max})$ is practically zero. For a chosen ΔR , the number of bins is given by $M = R_{max}/\Delta R$, and so the discretized version of the integral is

$$N \approx \sum_{i=0}^M \rho(R_i) \Delta V_i \quad (27)$$

Or

$$N_i \approx \rho(R_i) \Delta V_i \quad (28)$$

Where N_i is the number of particles in shell i . And

$$\Delta V_i = \frac{4\pi}{3\sqrt{\beta}} ((R_i + \Delta R)^3 - (R_i)^3) \quad (29)$$

We can rearrange equation (28) to get the one-body density expressed in terms of variables we can easily extract from our Monte Carlo steps:

$$\rho(R_i) \approx \frac{N_i}{\Delta V_i} = \frac{3\sqrt{\beta}}{4\pi ((R_i + \Delta R)^3 - (R_i)^3)} N_i \quad (30)$$

Sampling R_i for every particle in every Monte Carlo step and adding the expression in equation (30) to the bin corresponding to R_i yields a histogram. Normalizing this histogram by dividing each bin by the number of particles times the number of Monte-Carlo steps gives us an approximation of the one-body density $\rho(\mathbf{r})$. Note that in the spherically symmetric case we get the exact same equations, but with $\sqrt{\beta} = 1$.

C. Statistical physics

When performing a Monte Carlo sampling, one may be subject to both systematic and statistical errors. A possible source is that of the steplengths chosen for our system, which can lead to increasing nuisance parameters, such as the variance σ^2 . So, let's look at the nuisance parameters necessary to tell us about the precision of our Monte Carlo sampling.

1. Uncorrelated stochastic variables When dealing with Monte Carlo sampling we often draw random values, but by doing so at large scales the system can become predictable. In such a case, we can adjust the nuisance parameters to include stochastic variables within a certain sample size n . Thus, the mean can be expressed as

$$\bar{x}_n \equiv \frac{1}{n} \sum_{k=1}^n x_k. \quad (31)$$

The variance of the measurements is given by

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_n)^2. \quad (32)$$

Where we used \bar{x}_n as an estimate of the true sample mean, μ . The central limit theorem tells us that \bar{x}_n approaches μ as the number of measurements goes to infinity.

Furthermore, the covariance of the measurements can be expressed as

$$\text{cov}(x) \equiv \frac{1}{n} \sum_{kl} (x_k - \bar{x}_n)(x_l - \bar{x}_n) \quad (33)$$

Which for an uncorrelated set of measurements is just the variance given in eq.(32). To determine how well the sample mean approximates the true mean value, we can calculate the sample error, which is just the standard deviation of the sample mean itself. It turns out that the sample variance is related to the measurements in the following manner [1]

$$\text{err}_x^2 = \text{var}(\bar{x}) = \frac{1}{n} \text{cov}(x). \quad (34)$$

2. Serially correlated stochastic variables The Metropolis algorithm produces a time-series which is serially correlated, meaning that each measurement covaries with neighbouring measurements within a certain distance. We are interested in finding this "radius" of influence. From eq. (34) and (33), we have

$$\text{err}_x^2 = \frac{1}{n} \text{cov}(x) = \frac{1}{n} \frac{1}{n} \sum_{kl} (x_k - \bar{x}_n)(x_l - \bar{x}_n) \quad (35)$$

We may split the covariance into cross terms and terms where $x_k = x_l$. The latter terms just sum up to the variance, and so

$$\text{err}_x^2 = \frac{1}{n} \text{var}(x) + \frac{1}{n} (\text{cov}(x) - \text{var}(x)) \quad (36)$$

The cross terms sum over all pairwise combinations of x_k and x_l twice while excluding the terms where $x_k = x_l$. We therefore get

$$\text{err}_x^2 = \frac{1}{n^2} \sum_{k=1}^n (x_k - \bar{x}_n)^2 + \frac{2}{n^2} \sum_{k < l} (x_k - \bar{x}_n)(x_l - \bar{x}_n) \quad (37)$$

Denoting $(x_k - \bar{x}_n)(x_l - \bar{x}_n)$ by $\tilde{x}_k \tilde{x}_l$, the correlation term may be written as

$$\begin{aligned} \sum_{k < l} \tilde{x}_k \tilde{x}_l &= \sum_{k=1}^n \sum_{l=k+1}^n \tilde{x}_k \tilde{x}_l \quad (38) \\ &= \tilde{x}_1 \tilde{x}_2 + \tilde{x}_1 \tilde{x}_3 + \tilde{x}_1 \tilde{x}_4 + \tilde{x}_1 \tilde{x}_5 + \dots + \tilde{x}_1 \tilde{x}_n \\ &\quad + \tilde{x}_2 \tilde{x}_3 + \tilde{x}_2 \tilde{x}_4 + \tilde{x}_2 \tilde{x}_5 + \dots + \tilde{x}_2 \tilde{x}_n \\ &\quad + \tilde{x}_3 \tilde{x}_4 + \tilde{x}_3 \tilde{x}_5 + \dots + \tilde{x}_3 \tilde{x}_n \\ &\quad \vdots \\ &\quad + \tilde{x}_{n-1} \tilde{x}_n \end{aligned}$$

This sum can be split up into its diagonal parts. As can be seen above, the samples in a given diagonal are all separated by a fixed distance d , with $d = 1$ for the leftmost diagonal and $d = n - 1$ for the rightmost one. The sum over a given diagonal d is then

$$\sum_{k=1}^{n-d} \tilde{x}_k \tilde{x}_{k+d} \quad (39)$$

Defining

$$f_d = \frac{1}{n} \sum_{k=1}^{n-d} \tilde{x}_k \tilde{x}_{k+d} \quad (40)$$

The correlation part of equation (34) can then be written as

$$\frac{2}{n^2} \sum_{k < l} \tilde{x}_k \tilde{x}_l = \frac{2}{n} \sum_{d=1}^{n-1} f_d \quad (41)$$

When $d = 0$, f_d is simply the variance. It is therefore natural to introduce the auto correlation function

$$\kappa_d = \frac{f_d}{\text{var}(x)} \quad (42)$$

Which is always equal to 1 for $d = 0$. The sample error can now be written as

$$\text{err}^2 = \frac{1}{n} \text{var}(x) + \frac{2}{n} \text{var}(x) \sum_{d=1}^{n-1} \frac{f_d}{\text{var}(x)} \quad (43)$$

$$= \frac{1}{n} \text{var}(x) \left(1 + 2 \sum_{d=1}^{n-1} \kappa_d \right) \quad (44)$$

Defining the autocorrelation time as

$$\tau \equiv \left(1 + 2 \sum_{d=1}^{n-1} \kappa_d\right) \quad (45)$$

The final expression for our sample error becomes

$$\text{err}_x^2 = \frac{\tau}{n} \text{var}(x) \quad (46)$$

The sample can be thought of as being reduced to effectively $n_{\text{eff}} = \frac{n}{\tau}$ measurements. When all the measurements are uncorrelated, $\tau = 1$ and the sample error is given by $\text{var}(x)/n$. For correlated samples, $\tau > 1$ and so the error increases. This means that simply calculating the sample error with $\tau = 1$ will in our case give a too low estimate of the error, since our measurements are correlated. We therefore need to find the autocorrelation time in order to find the true sample error. Doing this every time we run a Monte-Carlo simulation by actually calculating f_d which would be very computationally expensive, as every measurement would have to be stored. In addition we would have to calculate the sum of $(n^2 - n)/2$ products in order to evaluate f_d , which is no small task for $n \sim 10^7$.

Instead we can determine the autocorrelation time and the correct sample error through blocking. We calculate err_X^2 for each blocking transformation. When the error stops increasing and reaches a plateau, the last measurements of a block are no longer correlated with the first, and we can set this block size as the correlation time. The blocking procedure is described in more detail in section III B 2.

III. Methods

A. Variational Monte Carlo

1. Standard Metropolis method When simulating the ground state energy of a trapped, hard-sphere Bose gas we can utilize a variational Monte Carlo method based on the Metropolis algorithm. Two requirements to a Markov process is that of detailed balance, i.e. a system which reaches equilibrium stays at equilibrium, and that of ergodicity in which the ensemble average equals the time average. Here, the Standard Metropolis algorithm satisfies both and can thus be used as a reliable option when modelling a system of both interacting and non-interacting bosons with a certain trial wavefunction [2].

The Standard Metropolis algorithm, also known as *brute force* Metropolis, is based on initializing the particles at a random location x^i in space, with a uniform distribution. From that point on, we can suggest a move for each individual particle at a state μ to a new state ν with a certain transition probability $T(\mu \rightarrow \nu)$. If the

ratio of transition probabilities ($T(\nu \rightarrow \mu)/T(\mu \rightarrow \nu)$) is higher than some tolerance s , of our choosing, then the move is accepted and the trial positions now become the current positions of our particles. If the ratio is lower than s , then the move is rejected, and the particles remain stationary at their original positions. In our case, the transition probabilities are based on the values generated by the wavefunction for each new state.

2. Metropolis-Hastings method When we are not dealing with a symmetric initial distribution of particles in the phase space, as is the assumption under the brute force Metropolis algorithm, we can make use of what is known as the Metropolis-Hastings algorithm, i.e. *importance sampling*. Like the brute-force method, it is a Markov chain Monte Carlo method (MCMC) and utilises the same approach to calculate the acceptance or rejection of a move from one state μ to a new state ν . But, there is a profound difference when considering the actual move to the new state. Here, one takes into account the time-evolution of the probability distribution function (PDF) of the velocities of the particles. The time-evolution is due to the particles experiencing a stochastic drift force F^4 and can be expressed by the Fokker-Planck equation in eq. 48.

The total probability distribution function at three different times $t_0 < t' < t$, can be expressed by the Einstein-Smoluchenski-Kolmogorov-Chapman (ESKC) equation

$$W(\mathbf{x}, t | \mathbf{x}_0, t_0) = \int_{-\infty}^{\infty} W(\mathbf{x}, t | \mathbf{x}', t') W(\mathbf{x}', t' | \mathbf{x}_0, t_0) d\mathbf{x}' \quad (47)$$

and is derived from the three different probability functions at separate times t_0, t', t , as defined below

$$\begin{aligned} w(\mathbf{x}, t) &= \int_{-\infty}^{\infty} W(\mathbf{x}, t | \mathbf{x}', t') w(\mathbf{x}', t') d\mathbf{x}', \quad (??) \\ w(\mathbf{x}, t) &= \int_{-\infty}^{\infty} W(\mathbf{x}, t | \mathbf{x}_0, t_0) w(\mathbf{x}_0, t_0) d\mathbf{x}_0, \\ w(\mathbf{x}', t') &= \int_{-\infty}^{\infty} W(\mathbf{x}', t' | \mathbf{x}_0, t_0) w(\mathbf{x}_0, t_0) d\mathbf{x}_0. \end{aligned}$$

The time-evolution of the probability distribution function in (47) can be expressed as

$$\frac{\partial W(\mathbf{x}, s | \mathbf{x}_0)}{\partial s} = -\frac{\partial M_1 W(\mathbf{x}, s | \mathbf{x}_0)}{\partial x} + \frac{1}{2} \frac{\partial^2 M_2 W(\mathbf{x}, s | \mathbf{x}_0)}{\partial x^2}, \quad (48)$$

⁴ See section II B 3 under II. Theory for a more thorough, theoretical explanation on the drift force, otherwise also known as the quantum force.

i.e. the Fokker-Planck equation. Here, \mathbf{x} represents the spatial dependence, $s = t' - t_0$, and x_0 is the position at time t_0 .

The Langevin equation describes the stochastic drift force a particle is subject to as it moves throughout the medium, and is expressed as

$$\frac{dv}{dt} = -\xi v + F. \quad (49)$$

Once these properties are taken into consideration, one can calculate the acceptance ratio by also taking into account the Green's function as a function of the quantum force experienced by a particle in the state μ , and the quantum force experienced after the move to a new state ν .

B. Resampling methods

1. Bootstrap The idea of the bootstrap method is as follows. We have a sample consisting of n independent, identically distributed (i.i.d.) variables represented by the vector \hat{x} . Our goal is to find the PDF of some function of \hat{x} , $\hat{\theta}(\hat{x})$. To find this PDF, we draw with replacement n numbers from \hat{x} . Denote this vector of numbers by \hat{x}^* . Calculate $\hat{\theta}^*(\hat{x}^*) = \hat{\theta}(\hat{x}^*)$ and repeat the process n or less times. From this we can create a histogram of the relative frequency of $\hat{\theta}^*$, which is our estimate of the PDF. This is rarely done in practice, as one usually just calculates the quantity in interest with a point estimator.

In our case, we are interested in knowing the variance of the sample means. For a sample of size n , the bootstrap algorithm then tells us to draw n measurements with replacement from the same Monte-Carlo sample and calculate the mean \bar{x}^* . Repeating this process $\sim n$ times, an estimate of the sample variance is found by use of eq.(34). This method works best for small samples. For larger sample sizes the complexity of the bootstrap method is deterring, not to mention that the measurements in our case aren't i.i.d.

2. Blocking In contrast to the bootstrap method, the blocking method works best for large sample sizes. Let the original sample contain $n = 2^d$ measurements, where d is an integer. In order to guarantee that the automation process introduced later in this section works, let's restrict ourselves to the case where these measurements form a stationary time series, which is the case when using the Metropolis algorithm. Denote this time series by the vector \hat{X} . The idea behind blocking is to perform so-called blocking transformations, which halves the length of the vector by taking the mean of two adjacent elements of the previous vector. Mathematically, a

transformation k , where $k \in [1, 2, \dots, d]$, is expressed as

$$\begin{aligned} (\hat{X}_0)_k &\equiv (\hat{X})_k \\ (\hat{X}_{i+1})_k &\equiv \frac{1}{2}((\hat{X}_i)_{2k-1} + (\hat{X}_i)_{2k}) \end{aligned} \quad (50)$$

It can be shown that if \hat{X}_0 is a stationary time series, then so is \hat{X}_k [3]. Now, we may treat each element/block of a given vector \hat{X}_i as an estimate of the mean value of a Monte-Carlo sample. For brevity, denote the elements of a given vector by X_k . We wish to find the variance of these elements/means for a given blocking iteration. Ignoring the blocking iteration index, the variance is given by [3]

$$\text{var}(\bar{X}) = \text{var}\left(\frac{1}{n_k} \sum_{k=1}^{n_k} X_k\right) = \frac{1}{n_k^2} \text{cov}\left(\sum_{i=1}^{n_k} X_i, \sum_{j=1}^{n_k} X_j\right) \quad (51)$$

The autocovariance is defined as

$$\gamma(h) = \text{cov}(X_i, X_{i+h}), \quad (52)$$

where $h = |i - j|$. From the distributive property of the covariance, we get

$$\begin{aligned} \text{var}(\bar{X}) &= \frac{1}{n_k^2} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \text{cov}(X_i, X_j) \\ &= \frac{1}{n_k^2} [n_k \gamma(0) + (n_k - 1) \gamma(1) + (n_k - 2) \gamma(2) + \dots + \gamma(n_k) \\ &\quad + (n_k - 1) \gamma(-1) + (n_k - 2) \gamma(-2) + \dots + \gamma(1 - n_k)] \end{aligned} \quad (53)$$

where $\gamma(0)$ is just the sample variance with no consideration to correlation. Let's denote this variance by σ_k^2 . This yields

$$\text{var}(\bar{X}) = \frac{\sigma_k^2}{n_k} + \frac{2}{n_k^2} \sum_{h=1}^{n_k-1} (n_k - h) \gamma(h) \quad (54)$$

$$= \frac{\sigma_k^2}{n_k} + \underbrace{\frac{2}{n_k} \sum_{h=1}^{n_k-1} \left(1 - \frac{h}{n_k}\right) \gamma(h)}_{e_k} \quad (55)$$

Where e_k is called the truncation error. It can be shown that [1]

$$\bar{X}_{j+1} = \bar{X}_j, \quad (56)$$

as one might expect. By induction one can then show that

$$\text{var}(\bar{X}_0) = \text{var}(\bar{X}_i) = \text{var}(\bar{X}). \quad (57)$$

The variance of the blocks is therefore constant as a function of blocking iteration k ;

$$\text{var}(\bar{X}) = \frac{\sigma_k^2}{n_k} + e_k \quad (58)$$

The sequence $\{e_k\}_{k=0}^{d-1}$ is decreasing [3], and thus e_k can be made as small as we like by increasing k through increasing d . For the $\text{var}(\bar{X})$ to remain constant for each blocking iteration, $\frac{\sigma_k^2}{n_k}$ must increase as a function of k and eventually reach a plateau when $e_k \approx 0$ and the blocks are no longer correlated. We can then extract the variance from one of these uncorrelated blocking iterations.

This can be done in a brute force manner by plotting σ_k^2/n_k vs k and estimating when the plateau is reached, or one may automate this process. The automation procedure is explained in great detail in [1]. The outline is as follows: For each blocking iteration, calculate the test statistic M_j given by

$$M_j = \sum_{k=j}^{d-1} \frac{n_k [(n_k - 1) \sigma_k^2 / (n_k^2) \gamma_k(1)]}{\sigma_k^4} \quad (59)$$

M_j is χ^2 -distributed with $d - k$ degrees of freedom. Next, pick a significance level, α , and find the smallest k such that $M_k \leq q_{d-k}(1 - \alpha)$. Using this k , the final estimate of the sample variance is $\sigma_b^2 = \sigma_k^2/n_k$.

C. Optimization methods

In order to achieve the minimum value of the local energy expressed in eq. (4), we can adjust the variational parameter α of the trial wavefunction in eq. (15). This can be done by multiple methods such as the conjugate gradient method, the steepest descent method or simply by Newton's method. In this project we will rely on the steepest descent method, computationally known as gradient descent (GD).

1. Gradient descent When dealing with a differentiable function F dependent on a variational parameter, we can find the local minimum by taking steps in the negative direction proportional to $-\nabla F$.⁵ Now, there is a certain learning curve as to how quickly we can reach a local minima. This learning curve is dependent on the learning rate γ , often also referred to as the step length. The smaller the learning rate, the longer it takes to converge to an optimal value. The greater the learning rate the more prone we are to experience fluctuant behaviour.

The gradient descent method is based on an iterative scheme in which the next value x_{k+1} is dependent on the previous value x_k . This can also be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla F(\mathbf{x}_k), \quad (60)$$

where we start off the sequence with an initial guess x_0 . In our case, the differentiable function F corresponds to

the derivative of the local energy E_L wrt. α , whereas x_k corresponds to the current value of α_k .

One must however be aware of the limits of conjugate gradient methods, such as the gradient descent. For one, the gradient descent will converge to a minimum, but not necessarily the global minimum. In other words, it may get stuck in a local minimum unless we run the optimization method for a very long time, or choose a near perfect initial value x_0 . And second, as mentioned earlier, it is dependent on the learning rate γ_k , and can thus become very slow if we want a higher accuracy in the optimal parameter.

IV. Results

A. Non-interacting spherical system

1. Standard Metropolis We start off by looking at the simplest case, i.e. examining the brute force Metropolis algorithm. By setting $\beta = 1$ and $a = 0$ we will look at a spherical harmonic oscillator with no interaction. Table I shows the measurements of the local energy as a function of α . Here, σ corresponds to the biased sample error of the Monte Carlo runs, whereas σ_b corresponds the error produced by the blocking method. As we can see, σ produces lower errors between the measurements than that of the blocking error, giving a misleading confidence in our results. The number of accepted steps is represented by the variable A_{ratio} , while t_{CPU} tells us the amount of computational time required (in seconds) to calculate the local energy.

This computation is furthermore done for $N = 1, 10$ and 100 particles in both $d = 1, 2$ and 3 dimensions, as is visualized in figure 2 showing the local energy as a function of α .

2. Adding importance sampling Next, we look at the case of importance sampling, i.e. implementing the Metropolis-Hastings algorithm. We continue to look at the case of a spherical harmonic oscillator with $\beta = 1$ and $a = 0$. As found from the brute-force algorithm, we can use the fact that we can obtain a local minima of the local energy when $\alpha = 0.5$. And as can be seen from table II, we compare the time required to compute the local energy in the cases of $N = 1, 10, 100$ and 500 particles in 3 dimensions. This was done using $1e5$ Metropolis steps, with a steplength of 0.5 and timestep 0.01. The CPU run times have been averaged over 10 samples. Since the run time will vary depending on what the machine is doing, a breakpoint was used to get samples that are more spaced out in time. Here, we see a clear difference between the two methods. Importance sampling uses a longer computational time, both for the analytic and the numeric solutions. But in return it gives a higher acceptance ratio which leads to a lower estimate on the biased error, as

⁵ If we want the local maximum we may take steps in the positive direction proportional to $+\nabla F$.

can be shown in table III.

TABLE I. Sample measurements of the local energy using Standard Metropolis algorithm, for 1 particle in 3 dimensions. This is run with 2^{20} Metropolis steps and a step length of 1.0.

α	E_L	σ	σ_b	A_{ratio}	$t_{CPU}[s]$
Analytic					
0.20	2.1353	0.0015	0.0017	0.6697	1.6700
0.25	1.8486	0.0011	0.0016	0.6344	1.6447
0.30	1.6882	0.0008	0.0009	0.6025	1.6083
0.35	1.5953	0.0006	0.0009	0.5749	1.5492
0.40	1.5366	0.0004	0.0008	0.5501	1.4919
0.45	1.5105	0.0002	0.0002	0.5254	1.4062
0.50	1.5000	0.0000	0.0000	0.5051	1.3278
0.55	1.5028	0.0002	0.0003	0.4852	1.2876
0.60	1.5132	0.0003	0.0005	0.4673	1.0491
0.65	1.5325	0.0005	0.0005	0.4490	0.9935
0.70	1.5559	0.0006	0.0007	0.4332	0.9996
Numeric					
0.20	2.1315	0.0015	0.0046	0.6698	6.3758
0.25	1.8493	0.0011	0.0042	0.6335	6.1484
0.30	1.6880	0.0008	0.0010	0.6023	5.9014
0.35	1.5951	0.0006	0.0017	0.5748	5.6388
0.40	1.5382	0.0004	0.0006	0.5502	5.3990
0.45	1.5103	0.0002	0.0003	0.5257	5.1965
0.50	1.5000	0.0000	0.0000	0.5045	5.0115
0.55	1.5026	0.0002	0.0004	0.4848	4.5887
0.60	1.5136	0.0003	0.0002	0.4669	4.4997
0.65	1.5325	0.0005	0.0011	0.4492	4.2859
0.70	1.8482	0.0003	0.0038	0.9173	3.6689

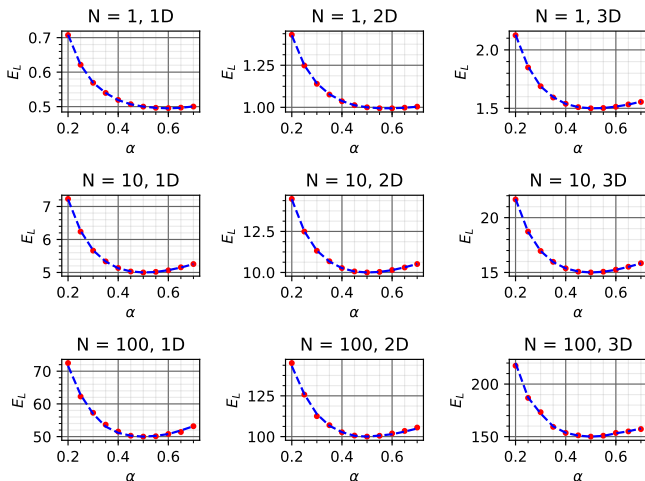


FIG. 2. Local energy as a function of α using the standard Metropolis algorithm. This is run for 2^{20} number of steps for 1 and 10 particles, whereas 100 particles was only run with 2^{16} steps due to the computational time required. The red dots correspond to the analytic calculations, while the blue dashed line corresponds to the numeric calculations.

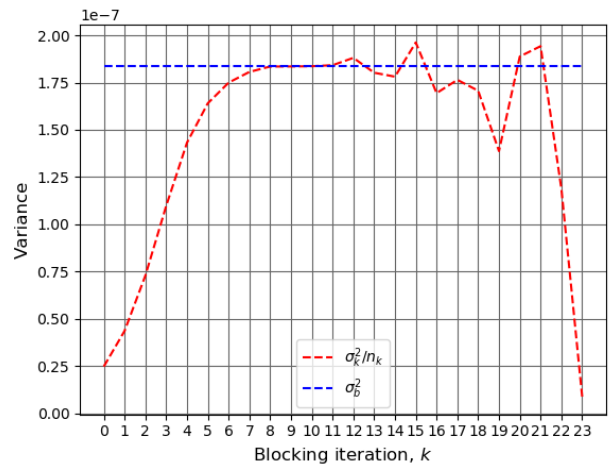


FIG. 3. Comparison of the biased estimator σ_k^2/n_k and σ_b^2 found through the automated blocking described in III B 2, using a significance level of $\alpha = 0.01$. As can be seen, the plateau is reached at about the sixth iteration, indicating a correlation time of $\tau \sim 2^6$ [You sure?] for a step length of 1. As expected, the behaviour of σ_k^2/n_k becomes increasingly erratic for high k . This is because the number of blocks are too low to give a good estimate of the sample variance.

Of course, the number of accepted steps is dependent on the timestep chosen for importance sampling. The relation between the number of accepted steps and the timesteps can be shown in figure 4, where $\delta t \in [0.1, 1.0]$. Here, we see that the ratio declines as the timesteps increases. Optimally, the timestep δt is to be within the range $[0.001, 0.01]$ for stable values of the local energy. We can choose a low timestep, such as $\delta t = 0.001$, but this would require even more computational power as more steps are accepted. Thus, there is a fine line between the desired accuracy of the local energy, and the computational power needed behind the scenes when it comes to importance sampling.

TABLE II. The computational time required to calculate the local energy. This was run with 3 dimensions, and $1e5$ Metropolis steps with a step length of 0.5.

N	E_L	Brute Force		Importance Sampling	
		$t_{anal}[s]$	$t_{num}[s]$	$t_{anal}[s]$	$t_{num}[s]$
1	0.5	0.1219	0.4981	0.1633	0.6423
10	5.0	0.6938	22.2768	0.9069	29.4808
100	50.0	6.4496	2051.1641	8.4655	2587.9464
500	250.0	32.6107	—	42.3359	—

B. Interacting elliptical system

Now, we turn to the system in which we include the repulsive potential through the Jastrow factor in eq. (15). Here, we set $\beta = 2.82843$ and $a = 0.0043$, thus allow-

TABLE III. Sample measurements of the local energy using the Metropolis-Hastings algorithm, for 1 particle in 3 dimensions. This is run with $1e5$ Metropolis steps and a step length of 0.5 and a time step $\delta t = 0.01$.

α	E_L	σ	σ_b	A_{ratio}	$t_{CPU}[s]$
Analytic					
0.20	1.3802	0.0009	0.0184	0.9520	1.0247
0.25	1.2976	0.0006	0.0141	0.9473	1.0047
0.30	1.2969	0.0005	0.0093	0.9431	1.0066
0.35	1.3205	0.0003	0.0062	0.9388	0.9951
0.40	1.3662	0.0002	0.0022	0.9355	0.9943
0.45	1.4284	0.0001	0.0017	0.9317	0.9918
0.50	1.5000	0.0000	0.0000	0.9288	0.9922
0.55	1.5796	0.0001	0.0013	0.9261	0.9905
0.60	1.6655	0.0002	0.0017	0.9230	0.9857
0.65	1.7519	0.0002	0.0017	0.9197	0.9779
0.70	1.8447	0.0003	0.0042	0.9171	0.9768
Numeric					
0.20	1.3594	0.0009	0.0174	0.9528	3.7641
0.25	1.3035	0.0007	0.0149	0.9478	3.7386
0.30	1.2929	0.0005	0.0072	0.9434	3.8651
0.35	1.3184	0.0003	0.0070	0.9393	3.7873
0.40	1.3662	0.0002	0.0028	0.9351	3.7665
0.45	1.4288	0.0001	0.0012	0.9318	3.7455
0.50	1.5000	0.0000	0.0000	0.9294	3.7112
0.55	1.5789	0.0001	0.0014	0.9255	3.7167
0.60	1.6666	0.0002	0.0014	0.9229	3.6921
0.65	1.7566	0.0002	0.0029	0.9201	3.6837
0.70	1.8482	0.0003	0.0038	0.9173	3.6689

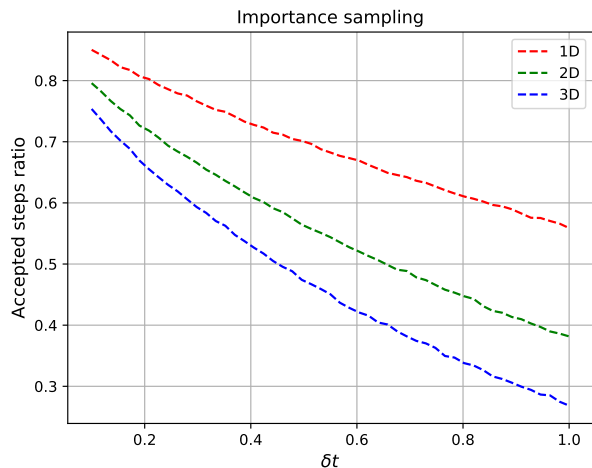


FIG. 4. The ratio of accepted steps compared to the timestep $\delta t \in [0.1, 1.0]$ as computed by the Metropolis-Hasting algorithm. This was run with 1 particle, in $d = 1, 2, 3$ dimensions, with 2^{19} Metropolis steps.

ing us to look at an elliptical, interacting boson system. From here on out, we will mainly use the Standard Metropolis algorithm.

1. Local energy The local energy for an interacting elliptical system in 3 dimensions for $N = 10, 50$ and 100 particles can be seen in table IV. Here, due to the computational power required to perform these calculations we had to scale down the number of Metropolis steps run for the N particles. For $N = 10$ and 50 we ran 2^{17} Metropolis steps, whereas for $N = 100$ we ran 2^{16} Metropolis steps. We only start sampling the energy after about 5% of the Metropolis steps have been run, to allow for the calibration of the energy. As one can see, due to the lower number of Metropolis steps, the error will increase as the total sample size decreases.

2. Optimal parameter Using the method of gradient descent we can try to find the optimal variational parameter α which will lead to a minimal value of the local energy. This can be seen in figure 5, where we find the local energy using 2^{17} steps over 100 iterations. This is done using a learning rate of $\gamma = 0.1$, and test values of $\alpha \in [0.1, 0.9]$. Taking the mean of α at the 100'th iteration for each test value of α , we get that the optimal parameter is

$$\alpha = 0.514232 \pm 0.013922.$$

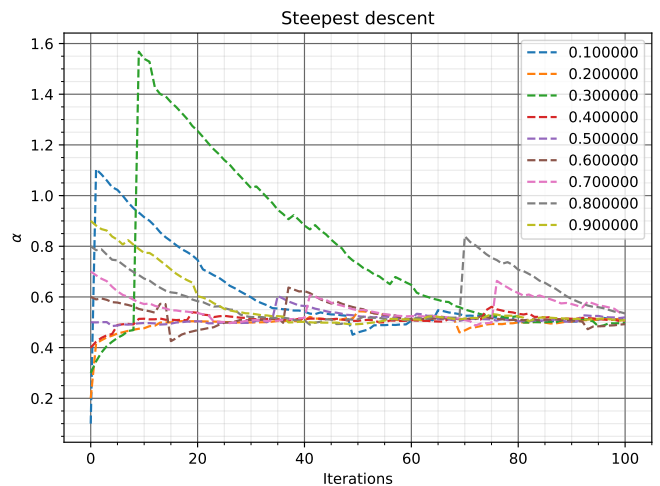


FIG. 5. Gradient descent using 2^{17} Metropolis steps and a learning rate $\gamma_k = 0.1$.

3. One-body density The one-body densities are shown in figure 6, where we have plotted the density for the ellipsoidal system with and without the Jastrow factor as well as the spherical non-interacting system. As seen, the effect of including the Jastrow factor, viz. interaction, is to spread out the particles. This is because the hard-core radius prevents the particles from stacking up on each other. This effect would be more pronounced for a larger number of particles.

Note that the one-body density for the spherical wavefunction is smaller than that of the elliptical wavefunction. This is to be expected, as ΔV increases by a factor of $\sqrt{\beta}$ when integrating over spherical shells compared to elliptical ones. From equation (30), the elliptical one-body density should therefore be a factor of $\sqrt{\beta}$ greater than the spherical. A rough estimate from figure 6 reveals this factor to be $1.666 \approx 8^{1/4} = \sqrt{\beta}$.

TABLE IV. Local energy as a function of α when running the brute force algorithm for $N = 10, 50$ and 100 particles in 3 dimensions. This is done with 2^{17} steps for $N = 10, 50$, whereas the computation for $N = 100$ is done with 2^{16} steps. All is run with a step length 1.0 .

α	E_L	σ	σ_b	A_{ratio}
10 particles				
0.20	36.3429	0.1635	1.3209	0.6489
0.25	31.0444	0.0860	0.2377	0.6150
0.30	28.5313	0.0374	0.2050	0.5851
0.35	27.0695	0.0991	0.1233	0.5588
0.40	30.9139	1.9405	3.0032	0.5342
0.45	26.4922	0.1349	0.8399	0.5114
0.50	30.0830	1.1763	1.4501	0.4914
0.55	28.8557	0.8074	1.3812	0.4732
0.60	27.5154	0.1505	1.0744	0.4554
0.65	27.2751	0.3708	0.8324	0.4395
0.70	28.5333	0.1853	0.3816	0.4241
50 particles				
0.20	190.3719	0.1971	1.9594	0.5855
0.25	167.2134	0.2417	2.0284	0.5454
0.30	166.5803	0.8381	6.5717	0.5114
0.35	155.6323	0.4971	4.2984	0.4831
0.40	182.5975	2.8327	18.5162	0.4569
0.45	158.9000	1.4974	5.1179	0.4335
0.50	163.9296	3.7315	9.7222	0.4106
0.55	166.6583	1.3770	22.6293	0.3915
0.60	167.3481	0.8089	6.3983	0.3744
0.65	234.4480	4.1277	27.1030	0.3597
0.70	237.2478	8.8231	21.1818	0.3440
100 particles				
0.20	444.8302	3.6103	32.8586	0.5713
0.25	353.9400	0.4392	6.1599	0.5377
0.30	361.6295	1.7793	20.8677	0.5084
0.35	350.0379	3.2279	20.7966	0.4792
0.40	484.9373	10.3091	110.6710	0.4542
0.45	370.9348	5.8136	32.6495	0.4317
0.50	417.5323	6.0516	35.9970	0.4138
0.55	367.7318	1.7960	31.0120	0.3913
0.60	413.8120	2.2802	30.6715	0.3754
0.65	527.4375	4.3581	51.8095	0.3587
0.70	411.8401	2.9253	30.0112	0.3450

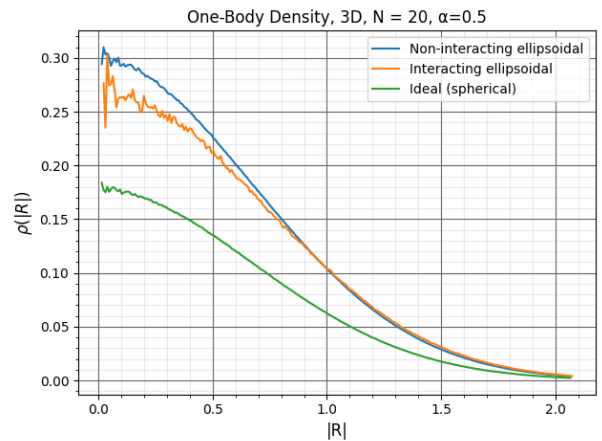


FIG. 6. One-body density vs. radial distance in units of a_{ho} . This was run with 20 particles in 3 dimensions, with a steplength of 0.1 and $1e7$ number of Metropolis steps.

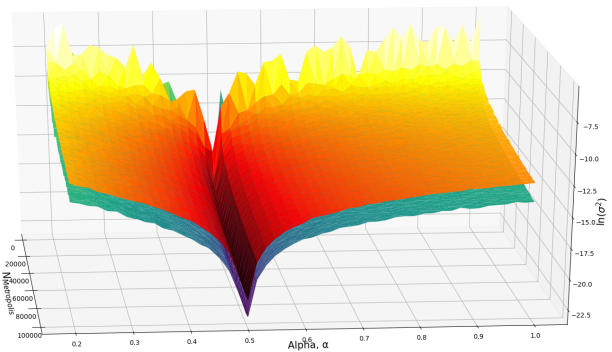


FIG. 7. Biased sample variance as a function of α and number of Metropolis steps. The top layer represents brute force whilst the bottom layer is for importance sampling. This is for the spherically symmetric non-interacting case.

V. Discussion

A. Non-interacting spherical system

1. Standard Metropolis From table I we see that the brute force method allows us to calculate the local energy with a small margin for error. The analytic method requires less computational power as it does not require multiple evaluations of the wavefunction at different time steps when computing the double derivative required for the kinetic energy. As for the numeric version, the double derivative does require this evaluation at different time steps.

In figure 2 we see that the analytic and numeric approximations to the local energy as a function of α agrees quite well, in all dimensions. The local energy reaches a minimum around $\alpha = 0.5$ as expected.

2. Adding importance sampling When adding importance sampling, as can be seen in table III, we do not assume a symmetric distribution. Here, the error σ appears smaller than that of the brute force method, which is due to the acceptance ratio being higher in this case. However, this is also reflected in the blocking error σ_b which is higher than that for the brute force method. This is due to the fact that the autocorrelation time τ increases along with the accepted number of steps.

The computational time required for importance sampling is longer than that of brute force, as can be seen in table II. This is due to the additional computational power required in calculating the quantum force, as expressed in eq. (18).

B. Interacting elliptical system

1. Local energy The local energy when including interaction between the bosons in an elliptical system can be seen in table IV. Here, E_L seems to fluctuate a bit, and reaches a minimum around $\alpha = 0.45$. From the blocking error we also see that there is a bit uncertainty around the numbers produced. This can also be due to the number of Metropolis steps chosen to produce the energies. We made an attempt at keeping the number of accepted steps $A_{ratio} \sim 0.5$ in order to minimize the autocorrelation time, giving us a more realistic blocking error. The reasoning is explained a bit further down in the section 'Extra Remarks'. As the number of particles also increases, the error increases as well.

2. Optimal parameter From the use of gradient descent we find that the most optimal value of α for an interacting, elliptical system is $\alpha = 0.514232 \pm 0.013922$. Figure 5 shows the learning curve of approaching α , when starting out with an initial 'guess' value. We could have chosen a smaller learning rate, which would hopefully converge to a more precise value of α , but this would then have required more iterations. Another thing to be aware of is the sudden spikes in α , especially for $\alpha_0 = 0.1, 0.3$ and 0.8 , although they converge back to $\alpha \sim 0.51$.

3. One-body density The one-body density is found by estimating the integral in eq. (26) by use of Monte-Carlo integration. As explained in section II B 4, we sum the one-body densities of all the particles rather than calculating the one-body density of an arbitrary particle in the ensemble. This is to save CPU cycles. To get a reasonably smooth plot, we needed $1e7$ Monte-Carlo steps for 20 particles. Sampling the position of every particle at each step is a lot cheaper than increasing the number of steps by a factor of 20 and only sampling the position of one particle each time.

Ideally we would like to plot the one-body density for 100 particles to see a more pronounced spread for the

interacting case compared to the non-interacting. The drastic increase in run-time is what prevented us from doing so. Future improvements would include optimizing our code so as to make this possible.

Our initial plots of the one-body density were suboptimal due to the way we estimated ΔV . For a discussion on this, see Appendix G.

VI. Extra remarks

As can be seen from table IV there is a noticeable difference between the error as calculated by the Monte Carlo runs and the blocking error. This is due to the autocorrelation time τ . If we had adjusted the step length in such a way that the number of accepted steps increased, then the time between a sample and the next uncorrelated sample would increase, thus resulting in a higher blocking error. This is illustrated in table V below, where each measurement of the local energy has been run using a step length of 0.1. Here, we see that the blocking error drastically increases, along with the number of particles, compared to the blocking error from table IV.

VII. Conclusion

Throughout this report we have attempted to find the local energy of both a non-interacting as well as an interacting boson system. This has been done in all 3 dimensions for $N = 1, 10, 50, 100$ and 500 particles. We started out with a non-interacting, spherical boson system using the Standard Metropolis algorithm, which allowed us to find an approximation to the minima of the local energy at $\alpha = 0.5$. Furthermore, we added importance sampling to the same system and observed the effects it had on the blocking error along with some additional computational time compared to that of the brute force method.

We have also examined how the Jastrow factor, i.e. an interacting system, affects the local energy when dealing with an elliptical potential. This was done using the brute force method. We also studied how to find the optimal variational parameter α leading to a minima of the local energy, although one must be aware that the method used, that is gradient descent, has a tendency to converge to a local minima of a system, and not necessarily the global minima.

Finally, we studied the one-body densities in the ellipsoidal system both with and without the Jastrow factor. Here, we found that the effects of including the Jastrow factor leads to a larger spread between the bosons and thus a lower density compared to that of a non-interacting system.

TABLE V. Local energy as a function of α when running the brute force algorithm for $N = 10, 50$ and 100 particles in 3 dimensions. This is done with 2^{17} steps for $N = 10, 50$, whereas the computation for $N = 100$ is done with 2^{16} steps. All is run with a step length 0.1.

α	E_L	σ	σ_b	A_{ratio}
10 particles				
0.20	34.0713	0.0611	1.6308	0.9281
0.25	31.5360	0.0342	0.7784	0.9212
0.30	28.2917	0.0539	0.8228	0.9189
0.35	25.8278	0.0823	0.7285	0.9159
0.40	25.9493	0.5688	1.0597	0.9104
0.45	26.9736	0.1466	1.2192	0.9060
0.50	27.1611	0.1133	2.1561	0.9056
0.55	24.7051	0.8227	4.2643	0.8992
0.60	30.1142	0.2282	10.5249	0.8982
0.65	27.3815	0.4034	1.8682	0.8938
0.70	28.3706	0.2960	2.6699	0.8908
50 particles				
0.20	190.1579	3.4246	15.9905	0.9306
0.25	194.2594	2.3330	4.0890	0.9236
0.30	128.8150	5.3416	5.0189	0.9196
0.35	180.0029	2.5183	11.7675	0.9178
0.40	132.1552	1.5432	9.5690	0.9112
0.45	200.1127	1.9996	34.1202	0.9068
0.50	182.7901	0.8094	19.4530	0.9035
0.55	154.4427	1.0658	15.5231	0.9028
0.60	161.1288	0.5748	24.0146	0.9015
0.65	253.8886	2.2427	29.1721	0.8947
0.70	207.7466	1.4643	113.9832	0.8939
100 particles				
0.20	624.8332	16.4875	504.4450	0.9051
0.25	631.0877	21.3615	9.6819	0.8955
0.30	346.0408	2.1510	11.2545	0.8898
0.35	466.0410	17.7181	46.5227	0.8852
0.40	390.7578	1.1315	25.2673	0.8816
0.45	326.9681	7.6810	60.7939	0.8774
0.50	802.4169	17.0915	93.5271	0.8782
0.55	495.5016	7.2860	9.1320	0.8753
0.60	410.5282	1.7056	52.0087	0.8719
0.65	969.0100	10.1001	41.0221	0.8674
0.70	787.3473	35.4289	152.2204	0.8661

References

- [1] M. Hjorth-Jensen. *Computational Physics 2: Statistical analysis*. Jan. 2020. URL: <http://compphysics.github.io/ComputationalPhysics2/doc/pub/statanalysis/pdf/statanalysis-print.pdf>.
- [2] M. Hjorth-Jensen. *Computational Physics Lecture Notes Fall 2015*. Aug. 2015. URL: <https://github.com/CompPhysics/ComputationalPhysics2/blob/gh-pages/doc/Literature/lectures2015.pdf>.
- [3] Marius Jonsson. “Standard error estimation by an automated blocking method”. In: *Phys. Rev. E* 98 (043304 Oct. 2018). DOI: 10.1103/PhysRevE.98.043304. URL: <https://journals.aps.org/pre/abstract/10.1103/PhysRevE.98.043304>.

Appendices

A. Trial wavefunction

The trial wavefunction is given by

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) \quad (61)$$

$$= \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j < k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right] \quad (62)$$

Where

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ \left(1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}\right) & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (63)$$

Is the correlation wavefunction and

$$g(\alpha, \beta, \mathbf{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)] \quad (64)$$

Where α and β are variational parameters. $\beta = 1$ for spherical traps and for non-interacting bosons ($a = 0$) we have $\alpha = 1/2a_{ho}^2$

B. Local Energy

We want to find the local energy

$$E_L(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} H \Psi_T(\mathbf{r}) \quad (65)$$

For the harmonic oscillator potential in one, two and three dimensions with

$$\begin{aligned} a = 0 & \quad \text{Non-interacting potential} \\ \beta = 1 & \quad \text{Spherically symmetric potential} \end{aligned}$$

For N particles. The external potential in this case is given by

$$V_{ext} = \begin{cases} \frac{1}{2}m\omega_{ho}^2 x^2 & 1D \\ \frac{1}{2}m\omega_{ho}^2 (x^2 + y^2) & 2D \\ \frac{1}{2}m\omega_{ho}^2 (x^2 + y^2 + z^2) & 3D \end{cases} \quad (66)$$

And the Hamiltonian is

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) \quad (67)$$

Inserting $a = 0$ into (84) gives $f(a, |\mathbf{r}_i - \mathbf{r}_j|) = 1$ and so

$$\Psi_T = \prod_i g(\alpha, \beta, \mathbf{r}_i) = \prod_j \exp[-\alpha(x_j^2 + y_j^2 + z_j^2)] \quad (68)$$

Next, we calculate the laplacian of the trial wavefunction. Differentiating only with respect to x gives

$$\begin{aligned}\frac{\partial^2}{\partial x_i^2} \Psi_T &= \frac{\partial}{\partial x_i} \left(-2\alpha x_i \prod_j \exp[-\alpha(x_j^2 + y_j^2 + z_j^2)] \right) \\ &= 2\alpha(2\alpha x_i^2 - 1) \prod_j \exp[-\alpha(x_j^2 + y_j^2 + z_j^2)]\end{aligned}$$

Repeating the same calculation for the y and z coordinates yields

$$\nabla_i^2 \Psi_T = 2\alpha(2\alpha \mathbf{r}_i^2 - 3) \Psi_T \quad (69)$$

This can be generalized for a system of dimension $d \in [1, 2, 3]$ to

$$\nabla_i^2 \Psi_T = 2\alpha(2\alpha \mathbf{r}_i^2 - d) \Psi_T \quad (70)$$

Inserting this into eq(...) gives

$$E_L = \frac{1}{\Psi_T} \sum_i^N \left[\frac{-\hbar^2}{2m} 2\alpha(2\alpha \mathbf{r}_i^2 - d) \Psi_T + \frac{1}{2} m \omega_{ho}^2 \mathbf{r}_i^2 \Psi_T \right] \quad (71)$$

Since we are dealing with non-interacting bosons, $\alpha = 1/2a_{ho}^2 = \frac{m\omega_{ho}}{2\hbar}$. Inserting this into equation (...) makes the first and last terms cancel, and we are left with

$$E_L = \frac{d}{2} \hbar \omega_{ho} N \quad (72)$$

Where $d \in [1, 2, 3]$ is the dimension of our system and N the number of particles.

The drift force is given by

$$F = \frac{2\nabla \Psi_T}{\Psi_T} \quad (73)$$

$$= \frac{2}{\Psi_T} \nabla_i \prod_j \exp[-\alpha(x_j^2 + y_j^2 + z_j^2)] \quad (74)$$

$$= \frac{2}{\Psi_T} \left(-2\alpha(x_i \hat{i} + y_i \hat{j} + z_i \hat{k}) \right) \Psi_T \quad (75)$$

$$(76)$$

Or in general

$$F = -4\alpha \mathbf{r}_i \quad (77)$$

Where $\mathbf{r}_i = x_i \hat{i} + y_i \hat{j}$ in 2D and $\mathbf{r}_i = x_i \hat{i}$ in 1D.

C. First derivative

The trial wavefunction (reference)

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) \quad (78)$$

$$= \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j < k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right] \quad (79)$$

Can be rewritten as

$$\Psi_T(\mathbf{r}) = \left[\prod_i \phi(\mathbf{r}_i) \right] \exp \left(\sum_{j < k} u(r_{jk}) \right) \quad (80)$$

By defining

$$\begin{aligned}\phi(\mathbf{r}_i) &\equiv g(\alpha, \beta, \mathbf{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + z_i^2)] \\ r_{ij} &\equiv |\mathbf{r}_i - \mathbf{r}_j|\end{aligned}$$

and

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

Where $u(r_{ij}) = \ln f(r_{ij})$.

The first derivative for particle k is then

$$\begin{aligned}\nabla_k \Psi_T(\mathbf{r}) &= \left[\nabla_k \prod_i \phi(\mathbf{r}_i) \right] \exp \left(\sum_{j < k} u(r_{jk}) \right) \\ &+ \left[\prod_i \phi(\mathbf{r}_i) \right] \nabla_k \exp \left(\sum_{j < k} u(r_{jk}) \right) \quad (81)\end{aligned}$$

Where the derivative of the exponential is

$$\nabla_k \exp \left(\sum_{j < m} u(r_{jm}) \right) = \exp \left(\sum_{j < m} u(r_{jm}) \right) \sum_{j < m} \nabla_k u(r_{jm})$$

And the sum

$$\begin{aligned}\sum_{j < m} u(r_{jm}) &= \sum_{j=1} \sum_{m=j+1} u(r_{jm}) \\ &= [u(r_{12}) + u(r_{13}) + \dots + u(r_{1N})] \\ &\quad + [u(r_{23}) + \dots + u(r_{2N})]\end{aligned}$$

$$\begin{aligned}\vdots & \\ & + [u(r_{N-1,N})]\end{aligned}$$

Ensures that every particle couples once to every other particle except itself. We see that the order in which they

couple is not consistent, but because r_{jm} is symmetric under interchange of indices, we get

$$\sum_{j < m} \nabla_k u(r_{jm}) = \sum_{l \neq k} \nabla_k u(r_{kl}) \quad (82)$$

The derivative of the product is simply

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

And so the first derivative of the trial wavefunction is

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

For our specific trial wavefunction...

D. Second derivative

$$\frac{1}{\Psi_T(\mathbf{r})} \nabla_k^2 \Psi_T(\mathbf{r}) = \frac{1}{\left[\prod_i \phi(\mathbf{r}_i) \right] \exp \left(\sum_{i < j} u(r_{ij}) \right)} \left\{ \quad (83) \right.$$

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

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

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

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

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

Using

$$\frac{\prod_{i \neq k} \phi(\mathbf{r}_i)}{\prod_i \phi(\mathbf{r}_i)} = \frac{1}{\phi(\mathbf{r}_k)}$$

And equation (...), the first three terms can be contracted to yield

$$(i) + (ii) + (iii) = \frac{\nabla_i^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \nabla_k u(r_{jm})$$

$\nabla_k u(r_{jm})$ can be calculated by use of the chain rule

$$\begin{aligned} \nabla_k u(r_{kj}) &= \left(\frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j} + \frac{\partial}{\partial z} \hat{k} \right) u(r_{kj}) \\ &= \left(\frac{\partial r_{kj}}{\partial x} \hat{i} + \frac{\partial r_{kj}}{\partial y} \hat{j} + \frac{\partial r_{kj}}{\partial z} \hat{k} \right) \frac{\partial u(r_{kj})}{\partial r_{kj}} \end{aligned}$$

Where

$$\begin{aligned} \frac{\partial r_{kj}}{\partial x} \hat{i} &= \frac{\partial}{\partial x} |\mathbf{r}_k - \mathbf{r}_j| \hat{i} \\ &= \frac{\partial}{\partial x} \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2 + (z_k - z_j)^2} \hat{i} \\ &= \frac{(x_k - x_j) \hat{i}}{r_{kj}} \end{aligned}$$

And similarly for the y and z derivatives. Equation ... therefore becomes

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

Where $u'(r_{kj}) \equiv \frac{\partial u(r_{kj})}{\partial r_{kj}}$

Inserting this back into equation (...) yields

$$(i) + (ii) + (iii) = \frac{\nabla_i^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} u'(r_{kj})$$

After differentiating the exponent, the fourth term reduces to

$$\begin{aligned} (iv) &= \sum_{i \neq k} \nabla_k u(r_{ki}) \sum_{j \neq k} \nabla_k u(r_{kj}) \\ &= \sum_{i \neq k} \sum_{j \neq k} (\nabla_k u(r_{ki})) (\nabla_k u(r_{kj})) \\ &= \sum_{i \neq k} \sum_{j \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \end{aligned}$$

After cancelling terms in the numerator and denominator of the fifth term, one is left with

$$\begin{aligned} (v) &= \nabla_k \sum_{j \neq k} \nabla_k u(r_{kj}) = \sum_{j \neq k} \nabla_k \left[\frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} u'(r_{kj}) \right] \\ &= \sum_{j \neq k} \left[\nabla_k \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \right] u'(r_{kj}) + \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \nabla_k u'(r_{kj}) \end{aligned}$$

Using the product rule gives

$$\nabla_k \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} = \frac{[\nabla_k(\mathbf{r}_k - \mathbf{r}_j)] r_{kj} - (\mathbf{r}_k - \mathbf{r}_j) \nabla_k r_{kj}}{r_{kj}^2}$$

Where $\nabla_k(\mathbf{r}_k - \mathbf{r}_j) = 3$ and $\nabla_k r_{kj} = (\mathbf{r}_k - \mathbf{r}_j)/r_{kj}$. Inserting this into eq (...) yields

$$\nabla_k \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} = \frac{3}{r_{kj}} - \frac{(\mathbf{r}_k - \mathbf{r}_j)^2}{r_{kj}^3} = \frac{3}{r_{kj}} - \frac{r_{kj}^2}{r_{kj}^3} = \frac{2}{r_{kj}}$$

To calculate the last term, we use the chain rule as in equation (...) to get

$$\begin{aligned} \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \nabla_k u'(r_{kj}) &= \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \nabla_k \frac{\partial u(r_{kj})}{\partial r_{kj}} \\ &= \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \underbrace{\left(\frac{\partial r_{kj}}{\partial x} \hat{i} + \frac{\partial r_{kj}}{\partial y} \hat{j} + \frac{\partial r_{kj}}{\partial z} \hat{k} \right)}_{(\mathbf{r}_k - \mathbf{r}_j)/r_{kj}} \frac{\partial^2 u(r_{kj})}{\partial r_{kj}^2} \\ &= \frac{(\mathbf{r}_k - \mathbf{r}_j)^2}{r_{kj}^2} u''(r_{kj}) \end{aligned}$$

And so

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

Inserting eq(..) and eq(..) into eq(...) gives

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

Finally adding all the terms together gives

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

Where $u'(r_{kj}) \equiv \frac{\partial u(r_{kj})}{\partial r_{kj}}$ and $u''(r_{kj}) \equiv \frac{\partial^2 u(r_{kj})}{\partial r_{kj}^2}$. Recall the definition of $u(r_{kj}) \equiv \ln f(r_{kj})$ with

$$f(a, r_{kj}) = \begin{cases} 0 & r_{kj} \leq a \\ \left(1 - \frac{a}{r_{kj}}\right) & r_{kj} > a \end{cases} \quad (84)$$

Where we defined $r_{kj} \equiv |\mathbf{r}_k - \mathbf{r}_j|$.

For $r_{kj} \leq a$ we get

$$u'(r_{kj}) = \frac{\partial}{\partial r_{kj}} \ln 0 = \frac{\partial}{\partial r_{kj}} (-\infty) = 0 \quad (85)$$

$$u''(r_{kj}) = 0 \quad (86)$$

Whilst for $r_{kj} > a$ we have

$$u'(r_{kj}) = \frac{\partial}{\partial r_{kj}} \ln \left(1 - \frac{a}{r_{kj}}\right) = \frac{1}{1 - \frac{a}{r_{kj}}} \frac{\partial}{\partial r_{kj}} \left(1 - \frac{a}{r_{kj}}\right) \quad (87)$$

$$= \frac{a}{r_{kj}(r_{kj} - a)} \quad (88)$$

Differentiating again yields

$$u''(r_{kj}) = \frac{a^2 - 2r_{kj}a}{r_{kj}^2(r_{kj} - a)^2} \quad (89)$$

E. Finding $\frac{\partial}{\partial \alpha} \langle E_L(\alpha) \rangle$

We want to find

$$\frac{\partial}{\partial \alpha} \langle E_L(\alpha) \rangle = \frac{\partial}{\partial \alpha} \frac{\int \psi_T^* E_L \psi_T d\tau}{\int \psi_T^* \psi_T d\tau} \quad (90)$$

In our case, the trial wavefunction is real. We therefore get $\psi_T^* = \psi_T$. Furthermore, the local energy is defined as $E_L = \frac{1}{\psi_T} \hat{H} \psi_T$. This gives

$$\frac{\partial}{\partial \alpha} \langle E_L(\alpha) \rangle = \frac{\partial}{\partial \alpha} \frac{\int \psi_T \hat{H} \psi_T d\tau}{\int \psi_T^2 d\tau} \quad (91)$$

$$= \underbrace{\frac{1}{C} \int \psi_\alpha \hat{H} \psi_T + \psi_T \frac{\partial}{\partial \alpha} \hat{H} \psi_T d\tau}_i - \underbrace{\frac{\int \psi_T \hat{H} \psi_T d\tau}{C^2} \int 2\psi_T \psi_\alpha d\tau}_{ii} \quad (92)$$

Where we defined $C \equiv \int \psi_T^2 d\tau$ and $\psi_\alpha \equiv \frac{\partial \psi_T}{\partial \alpha}$. From the expression of \hat{H} given in eq(...) it can be seen that $\frac{\partial}{\partial \alpha}$ and \hat{H} commute, and so $\frac{\partial}{\partial \alpha} \hat{H} = \hat{H} \frac{\partial}{\partial \alpha}$

$$(i) = \frac{1}{C} \int \psi_\alpha \hat{H} \psi_T + \psi_T \hat{H} \psi_\alpha d\tau \quad (93)$$

Our hamiltonian is hermitian, meaning that $\psi_T \hat{H} \psi_\alpha = \psi_\alpha \hat{H} \psi_T$. Using this gives

F. Scaling the Hamiltonian

$$\frac{\partial}{\partial \alpha} \langle E_L(\alpha) \rangle = \frac{2}{C} \int \psi_\alpha \hat{H} \psi_T d\tau - 2 \frac{\int \psi_T \hat{H} \psi_T d\tau}{C} \cdot \frac{\int \psi_T \psi_\alpha d\tau}{C} \quad (94)$$

$$= \frac{2}{C} \int \frac{\psi_\alpha}{\psi_T} \frac{1}{\psi_T} \hat{H} \psi_T \cdot \psi_T^2 d\tau - 2 \frac{\int \frac{1}{\psi_T} \hat{H} \psi_T \cdot \psi_T^2 d\tau}{C} \cdot \frac{\int \frac{\psi_\alpha}{\psi_T} \psi_T^2 d\tau}{C} \quad (95)$$

$$\Rightarrow \frac{\partial}{\partial \alpha} \langle E_L(\alpha) \rangle = 2 \left\langle \frac{\psi_\alpha}{\psi_T} E_L(\alpha) \right\rangle - 2 \langle E_L(\alpha) \rangle \left\langle \frac{\psi_\alpha}{\psi_T} \right\rangle \quad (96)$$

The Hamiltonian in the interacting case is given by

$$H = \sum_{i=1}^N -\frac{\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) + \sum_{i=1}^N V_{int}(\mathbf{r}_i \mathbf{r}_j) \quad (97)$$

The elliptical external potential is given by

$$V_{ext} = \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z^2 z^2] \quad (98)$$

Where $\omega_{ho} = \omega_\perp$ is the angular frequency perpendicular to the z-axis. It is related to the characteristic length of the trap a_{ho} in the following manner

$$a_{ho} = \sqrt{\frac{\hbar}{m\omega_{ho}}} \quad (99)$$

In our project, we use lengths in units of a_{ho} . This means that we need to multiply \mathbf{r} by a_{ho} to get back to SI units:

$$V_{ext} = \frac{1}{2} m [\omega_{ho}^2 a_{ho}^2 (x[a_{ho}]^2 + y[a_{ho}]^2) + \omega_z^2 a_{ho}^2 z[a_{ho}]^2] \quad (100)$$

Inserting the expression for a_{ho} in eq. (...) into eq. gives

$$V_{ext} = \frac{1}{2} [\hbar \omega_{ho} (x^2 + y^2) + \hbar \frac{\omega_z^2}{\omega_{ho}} z^2] \quad (101)$$

Which measured in units of $\hbar \omega_{ho}$ is

$$V_{ext} = \frac{1}{2} [(x^2 + y^2) + \gamma z^2] \quad (102)$$

Where $\gamma = \frac{\omega_z^2}{\omega_{ho}^2}$. The Laplacian transforms as follows in going from lengths in units of meters to a_{ho} :

$$\nabla^2[\text{m}] = \frac{1}{a_{ho}^2} \nabla^2[a_{ho}] = \frac{m\omega_{ho}}{\hbar} \nabla^2[a_{ho}] \quad (103)$$

Inserting this along with eq. (...) back into the Hamiltonian and converting to energy in units of $\hbar \omega_{ho}$ yields

$$H = \sum_{i=1}^N \frac{1}{2} (-\nabla_i^2 + (x^2 + y^2) + \gamma z^2) + \sum_{i=1}^N V_{int}(\mathbf{r}_i \mathbf{r}_j) \quad (104)$$

G. One-Body Density

An alternative method for estimating dV would be differentiating equation (...) to get

$$dV = \frac{4\pi}{\sqrt{\beta}} R^2 dR \quad (105)$$

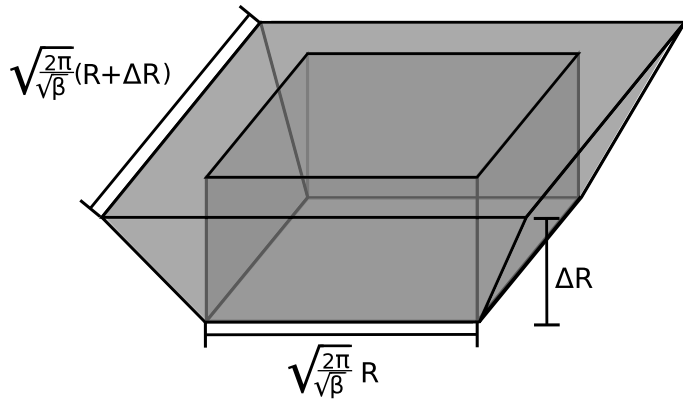


FIG. 8. $\Delta V'$ is given by the volume of the cuboid whilst ΔV is given by the volume of the inverted frustum. As $R \rightarrow 0$, ΔV approaches the volume of an inverted pyramid with a square base of length $\frac{\sqrt{2\pi}}{\beta^{1/4}}\Delta R$ while $\Delta V'$ goes to zero.

and using

$$\Delta V' = \frac{4\pi}{\sqrt{\beta}} R_i^2 \Delta R \quad (106)$$

To discretize the integral. However, this method underestimates ΔV , as can be seen in figure (...). This because we are simply multiplying the surface area of the ellipsoid by ΔR rather than integrating the area between R_i and $R_i + \Delta R$, which is effectively what we are doing by taking the volumetric difference of two ellipsoids. For $R \gg \Delta R$, the relative difference between the two methods is small. The problem arises when $R = \mathcal{O}(\Delta R)$. Now the underestimation becomes significant, and manifests itself as a spike in $\rho(\mathbf{r})$ as $R \rightarrow 0$.