University of Oslo
FYS4411
Computational physics II: Quantum mechanical systems

Project 1

Bendik Samseth

March 27, 2018

1 Introduction

The aim of this project is to use the Variational Monte Carlo (VMC) method and evaluate the ground state energy of a trapped, hard sphere Bose gas for different numbers of particles with a specific trial wave function. This trial wave function is used to study the sensitivity of condensate and non-condensate properties to the hard sphere radius and the number of particles.

2 Theory

2.1 Physical System

The trap we will use is a spherical (S) or an elliptical (E) harmonic trap in one, two and finally three dimensions, with the latter given by

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

The Hamiltonian of the system will be

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

Here ω_{ho}^2 defines the trap potential strength. In the case of the elliptical trap, $V_{ext}(x,y,z)$, $\omega_{ho} = \omega_\perp$ is the trap frequency in the perpendicular or xy plane and ω_z the frequency in the z direction. The mean square vibrational amplitude of a single boson at $T = 0K$ in the trap (1) is $\langle x^2\rangle = (\hbar/2m\omega_{ho})$ so that $a_{ho} \equiv (\hbar/m\omega_{ho})^{\frac{1}{2}}$ defines the characteristic length of the trap. The ratio of the frequencies is denoted $\lambda = \omega_z/\omega_\perp$ leading to a ratio of the trap lengths $(a_\perp/a_z) = (\omega_z/\omega_\perp)^{\frac{1}{2}} = \sqrt{\lambda}$.

Note: In the rest of this report, as well as in accompanying source code, we will use natural units with $\hbar = m = 1$.

We will represent **the inter-boson interaction** by a pairwise, repulsive potential:

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

where a is the so-called hard-core diameter of the bosons. Clearly, $V_{int}(|\mathbf{r}_i - \mathbf{r}_j|)$ is zero if the bosons are separated by a distance $|\mathbf{r}_i - \mathbf{r}_j|$ greater than a but infinite if they attempt to come within a distance $|\mathbf{r}_i - \mathbf{r}_j| \le a$.

The trial wave function for the ground state with N atoms will be given by

$$\begin{aligned}\Psi_T(\mathbf{r}) &= \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, \alpha, \beta) \\ &= \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i<j} f(a, |\mathbf{r}_i - \mathbf{r}_j|), \end{aligned} \quad (4)$$

where α and β are variational parameters. We choose the single-particle wave function to be proportional to the harmonic oscillator function for the ground state, i.e., we define $g(\alpha, \beta, \mathbf{r}_i)$ as:

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

For spherical traps we have $\beta = 1$ and for non-interacting bosons ($a = 0$) we have $\alpha = 1/2a_{ho}^2$ resulting in the exact wave function. The correlation wave function is

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

2.1.1 Scaling the System

We will use distances in units of a_{ho} in this report, $\boldsymbol{r}' = \boldsymbol{r}/a_{ho}$. Performing this substitution in the Hamiltonian (2), using $\nabla'^2 = a_{ho}^2 \nabla^2$, we get:

$$\begin{aligned} H &= \sum_i^N \left(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\boldsymbol{r}_i) \right) + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \\ &= \sum_i^N \Big(-\frac{\hbar^2}{2m}\frac{1}{a_{ho}^2}\nabla_i'^2 \\ &\qquad + \frac{m}{2}\left[\omega_{ho}^2 a_{ho}^2 (x_i'^2 + y_i'^2) + \omega_z^2 a_{ho}^2 z_i'^2\right] \Big) \\ &\qquad + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \\ &= \frac{\hbar\omega_{ho}}{2} \sum_i^N \left(-\nabla_i'^2 + x_i'^2 + y_i'^2 + \lambda^2 z_i'^2 \right) \\ &\qquad + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \end{aligned} \quad (7)$$

where once again $\lambda = \omega_z/\omega_{ho}$ describes the asymmetry in the trap. The interaction potential remains the same, as all lengths are scaled, and only the relative distances are used.

There will also be a scaling factor for the single-particle wave functions,

$$g(\alpha, \beta, \boldsymbol{r}_i) = \exp\left[-\alpha a_{ho}^2 (x_i'^2 + y_i'^2 + \beta z_i'^2)\right]. \quad (8)$$

The correlation wave functions remains unaffected.

All energies will due to this be given in units of $\hbar\omega_{ho}$, and lengths in units of α_{ho}. We fix the value of $\omega_{ho} = 1$ so that, along with the use of the natural units of $\hbar = m = 1$, we have

$$a_{ho} = \sqrt{\frac{\hbar}{m\omega_{ho}}} = 1. \quad (9)$$

This means the scaling has no effect on the numbers we would get, but now we have a well defined scale to relate the numbers to.

For notational sanity, the use of ticks to denote the scaled variables will be omitted, and can be assumed in the remainder of this report.

2.2 The Objective

Our objective is to evaluate the expectation value of the Hamiltonian. We cannot do this without the true wave function of the system, something we do not possess. We can, however, approximate the energy with the trial wave function.

$$E[H] = \langle H \rangle = \frac{\int \mathrm{d}\boldsymbol{R}\, \Psi_T^* H \Psi_T}{\int \Psi_T^* \Psi_T}. \quad (10)$$

where $\boldsymbol{R}$ is the matrix containing all the positions of the particles in the system, $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N]$. In order to numerically evaluate this integral we first manipulate it a bit. The probability density at position $\boldsymbol{R}$, under the trial wave function, is

$$P(\boldsymbol{R}, \boldsymbol{\alpha}) = \frac{|\Psi_T|^2}{\int \mathrm{d}\boldsymbol{R}\, |\Psi_T|^2}. \quad (11)$$

where $\boldsymbol{\alpha}$ is used for shorthand and represents the vector of all the variational parameters. We finally define a new quantity, called **the local energy**:

$$E_L(\boldsymbol{R}, \boldsymbol{\alpha}) = \frac{1}{\Psi_T} H \Psi_T \quad (12)$$

Combining these two definitions we can now rewrite $\langle H \rangle$ as follows:

$$\begin{aligned} \langle H \rangle &= \int \mathrm{d}\boldsymbol{R}\, P(\boldsymbol{R}, \boldsymbol{\alpha}) E_L(\boldsymbol{R}, \boldsymbol{\alpha}) \\ &\approx \frac{1}{n} \sum_{i=1}^n E_L(\boldsymbol{R}_i, \boldsymbol{\alpha}), \end{aligned} \quad (13)$$

where R_i are randomly drawn positions from the PDF $P(\boldsymbol{R}, \boldsymbol{\alpha})$. We have therefore that estimating the

average value of E_L yields an approximated value for $\langle H \rangle$. This value is in turn be an upper bound on the ground state energy, E_0. By the variational principle, if we minimize $\langle H \rangle$ under the variational parameters, we find an estimate for the true ground state energy of the system.

2.2.1 Exact Result for Simple System

It will be useful to be able to compare our results with exact analytical results where we have these. In the case of the symmetric harmonic oscillator trap, ignoring any iteractions between the bosons, we have an exact form for the ground state energy:

$$E_0 = \sum_{i=1}^{N} \sum_{d=1}^{D=\{1,2,3\}} \frac{\hbar\omega_{ho}}{2} = \frac{N \times D}{2}, \tag{14}$$

for N independent bosons in D dimensions (the two sums goes over all the degrees of freedom in the system). This follows from the setting $\alpha = 1/2$ (and $\beta = 1$) in Ψ_T, and using $a = 0$ (no interaction).

We also have an exact value for the variance of the energy in this case:

$$\begin{aligned} \sigma_E^2 &= \left\langle H^2 \right\rangle - \langle H \rangle^2 \\ &= \langle \Psi | H^2 | \Psi \rangle - \langle \Psi | H | \Psi \rangle^2 \\ &= \langle \Psi | E^2 | \Psi \rangle - \langle \Psi | E | \Psi \rangle^2 \\ &= E^2 \langle \Psi | \Psi \rangle - (E \langle \Psi | \Psi \rangle)^2 = 0. \end{aligned} \tag{15}$$

This follows when we have the exact wavefunction, which satisfies the time independent Schrödinger equation, $H|\Psi\rangle = E|\Psi\rangle$.

2.2.2 Gross-Pitaevskii Equation

The interactive system under investigation here can be approximated by the Gross-Pitaevskii equation, which is valid when the Bose-gas in question is sufficiently dilute. The energy under this model is given as[1]:

$$E_{\mathrm{GP}}[\Psi] = \int d\boldsymbol{R} \left[\frac{1}{2} \|\nabla \Psi\|^2 + V_{ext} |\Psi|^2 + 2\pi a |\Psi|^4 \right]. \tag{16}$$

If we insert the elliptical potential, and drop the Jastrow factor from the wavefunction (i.e. solve for the non-interacting case, setting $a = 0$), we solve this analytically and obtain an exact formula for the energy in the elliptical case as well. This will prove very useful as a benchmark for our implementation. Evaluating the integral we end up with (assuming $\omega_z = \beta$, which will also be used later)

$$E_{\mathrm{GP}} = N \cdot \left(\frac{1}{4\alpha} + \alpha \right) \cdot \left(1 + \frac{\beta}{2} \right), \tag{17}$$

which we can see corresponds to (14) when $\beta = 1$.

2.3 Calculating the Local Energy E_L

As the local energy is the quantity we are interested in computing for a large set of positions we would do well to consider how best to evaluate this expression effectively. For this we have two alternative approaches, 1) numerical differentiation and 2) finding an analytic, direct expression.

2.3.1 Numerical differentiation

We may set up an algorithm for the numerical approximation of the local energy as shown in Algorithm 1.

Algorithm 1 Calculate the local energy E_L using numerical differentiation.

Require: $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N]$, D = dimensions
Ensure: $y = E_L$

```
y = −2 × N × D × Ψ_T(R)
for i = 1 to N do
    for d = 1 to D do
        R_+ ← R + h e_{i,d}
        R_− ← R − h e_{i,d}
        y ← y + Ψ_T(R_+) + Ψ_T(R_−)
    end for
end for
y ← − y/2h²
y ← y/Ψ_T(R) + Σ_{i=1}^{N} V_ext + Σ_{i<j}^{N} V_int
```

An evaluation of E_L using this algorithm would be $\mathcal{O}(N^3 \times D) = \mathcal{O}(N^3)$ with interaction, and $\mathcal{O}(N^2)$ without interaction, from the complexity of Ψ_T.

2.3.2 Finding an Analytical Expression for E_L

Straight forward numerical differentiation is of course an option, but this is likely to be quite time-expensive to do. We will here try to speed up the calculation by producing a direct formula.

The hard part of the expression for E_L is

$$\frac{1}{\Psi_L}\sum_k^N \nabla_k^2 \Psi_L. \tag{18}$$

To get going, we rewrite the wave function as

$$\Psi_L(\boldsymbol{R}) = \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right), \tag{19}$$

where $r_{ij} = \|\boldsymbol{r}_{ij}\| = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|$, $u(r_{ij}) = \ln f(r_{ij})$, and $\phi(\boldsymbol{r}_i) = g(\alpha, \beta, \boldsymbol{r}_i)$.

Lets first evaluate the gradient with respect to particle k

$$\begin{aligned}
\boldsymbol{\nabla}_k \Psi_T(\boldsymbol{r}) &= \boldsymbol{\nabla}_k \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right) \\
&= \prod_{i\neq k} \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right)\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k) \\
&+ \prod_i \phi(\boldsymbol{r}_i)\boldsymbol{\nabla}_k \exp\left(\sum_{i<j} u(r_{ij})\right) \\
&= \Psi_T\left[\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right].
\end{aligned} \tag{20}$$

The first term is evaluated quite simply:

$$\begin{aligned}
\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} &= \frac{\boldsymbol{\nabla}_k}{\phi(\boldsymbol{r})}\exp\left[-\alpha\left(x_k^2 + y_k^2 + \beta z_k^2\right)\right] \\
&= -2\alpha\hat{\boldsymbol{r}}_k,
\end{aligned} \tag{21}$$

where the notation $\hat{\boldsymbol{r}}_k = (x, y, \beta z)$ is introduced for brevity. Note that in the 1D and 2D case we simply have $\hat{\boldsymbol{r}}_k = \boldsymbol{r}_k$.

The second term may be evaluated as follows:

$$\begin{aligned}
\boldsymbol{\nabla}_k u(r_{kj}) &= u'(r_{kj})\boldsymbol{\nabla}_k\sqrt{\|\boldsymbol{r}_k - \boldsymbol{r}_j\|^2} \\
&= \frac{u'(r_{kj})}{2r_{kj}}\boldsymbol{\nabla}_k\left(\|\boldsymbol{r}_k\|^2 - 2\boldsymbol{r}_k\cdot\boldsymbol{r}_j + \|r_j\|^2\right) \\
&= u'(r_{kj})\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\
&= \frac{\partial}{\partial r_{kj}}\left[\ln\left(1 - \frac{a}{r_{kj}}\right)\right]\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\
&= \frac{\boldsymbol{r}_{kj}}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)}.
\end{aligned} \tag{22}$$

Now we can find the Laplacian by taking the divergence of (20):

$$\begin{aligned}
\frac{1}{\Psi_L}\nabla_k^2\Psi_L &= \frac{1}{\Psi_L}\boldsymbol{\nabla}_k\cdot\Psi_T\left[\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\boldsymbol{\nabla}u(r_{kj})\right] \\
&= \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\nabla_k^2 u(r_{kj}) \\
&+ \frac{\boldsymbol{\nabla}_k(\phi(\boldsymbol{r}_k))\cdot\left(\sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)}{\phi(\boldsymbol{r}_k)} \\
&+ \left[\left(\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)\right. \\
&\left.\cdot\left(\sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)\right] \\
&= \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)}\cdot\sum_{j\neq k}\left(\frac{\boldsymbol{r}_{kj}}{r_{kj}}u'(r_{kj})\right) \\
&+ \sum_{i,j\neq k}\frac{\boldsymbol{r}_{ki}\cdot\boldsymbol{r}_{kj}}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \sum_{j\neq k}\nabla_k^2 u(r_{kj}).
\end{aligned} \tag{23}$$

There are two new quantities here which need to be

evaluated before we are done:

$$\frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} = 2\alpha\Big[2\alpha\|\hat{\boldsymbol{r}}_k\|^2 - d(\beta)\Big], \quad \text{with } d(\beta) = \begin{cases} 1 & \text{for} \quad \text{1D} \\ 2 & \text{for} \quad \text{2D} \\ 2+\beta & \text{for} \quad \text{3D} \end{cases}, \tag{24}$$

and

$$\begin{aligned} \nabla_k^2 u(r_{kj}) &= \boldsymbol{\nabla}_k \cdot u'(r_{kj})\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\ &= u'(r_{kj})\frac{2}{r_{kj}} + \frac{\boldsymbol{r}_{kj}}{r_{kj}} \cdot \boldsymbol{\nabla}_k u'(r_{kj}) \\ &= u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj}), \end{aligned} \tag{25}$$

where

$$\begin{aligned} u''(r_{ij}) &= \frac{\partial^2}{\partial r_{ij}^2}\ln\left(1 - \frac{a}{r_{ij}}\right) \\ &= \frac{a(a-2r_{ij})}{r_{ij}^2(r_{ij}-a)^2}. \end{aligned} \tag{26}$$

Inserting all of this back into (23) we get:

$$\begin{aligned} \frac{1}{\Psi_L}\nabla_k^2\Psi_L = 2\alpha\Big[2\alpha\|\hat{\boldsymbol{r}}_k\|^2 - d(\beta)\Big] \\ - 4\alpha\hat{\boldsymbol{r}}_k \cdot \left[\sum_{j\neq k}\frac{\boldsymbol{r}_{kj}}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)}\right] \\ + \sum_{i,j\neq k}\frac{\boldsymbol{r}_{ki}\cdot\boldsymbol{r}_{kj}}{r_{ki}r_{kj}}\frac{a}{r_{ki}(r_{ki}-a)}\frac{a}{r_{kj}(r_{kj}-a)} \\ + \sum_{j\neq k}\left(\frac{a(a-2r_{kj})}{r_{kj}^2(r_{kj}-a)^2} + \frac{2}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)}\right). \end{aligned} \tag{27}$$

We may note that without interactions ($a = 0$), this simplifies to only the first term, as all the other terms are proportional to a.

The complete expression for the local energy is then:

$$\begin{aligned} E_L &= \frac{1}{\Psi_T}H\Psi_T \\ &= \sum_i V_{ext}(\boldsymbol{r}_i) + \sum_{i<j}V_{int}(\boldsymbol{r}_i,\boldsymbol{r}_j) - \frac{1}{2}\sum_k \frac{1}{\Psi_T}\nabla_k^2\Psi_T \end{aligned} \tag{28}$$

where we substitute in (27) in the final sum. A single evaluation of the local energy is then $\mathcal{O}(N^3)$ with interaction, and $\mathcal{O}(N)$ without.

We can see that without interaction we obtain a linear-time expression, compared to quadratic-time using numerical differentiation. With interaction we have not been able to improve the complexity in terms of Big-O. This does not, however, mean that no improvement is obtained. A closer look shows that the numerical approach uses more evaluations by a constant factor of about three. This stems from the three wave function evaluations used in the central difference approximation of the second derivative. The analytic approach is closer to using a single evaluation, although the exact ratio is hard to define as the wave function is not directly used here.

In summary, we will expect a significant speedup using the analytic expression both with and without interaction enabled.

2.4 Calculating the Quantum Drift Force

Anticipating its later use, we will also find an expression for the so called quantum drift force, which we shall use when we consider importance sampling. For now, we just give its definition:

$$\boldsymbol{F}_k = \frac{2\boldsymbol{\nabla}_k\Psi_T}{\Psi_T} \tag{29}$$

This is interpreted as the force acting on particle k due to the trap and/or presence of other particles. As a physical justification for why $\boldsymbol{F}_k$ takes this form we can see that $\boldsymbol{F}_k$ is proportional with the gradient of Ψ_T, which we can intuitively understand as a force pushing the particle towards regions of space with higher probability.

Luckily this can now be quickly evaluated due to the results of the previous section,

$$\boldsymbol{F}_k = 2\left[\sum_{j\neq k}\frac{\boldsymbol{r}_{kj}}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)} - 2\alpha\hat{\boldsymbol{r}}_k\right]. \tag{30}$$

With interaction this is $\mathcal{O}(N)$, and without ($a = 0$) it simplifies to $\mathcal{O}(1)$.

For brevity, we may later use the notation $\boldsymbol{F}(\boldsymbol{R}) = [\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_N]$, denoting the matrix of all the individual forces on each particle.

2.5 One-Body Density

A quantity which is often of great interest is the one-body density. It is defined for particle 1 as

$$\rho(\boldsymbol{r}_1) = \int \mathrm{d}\boldsymbol{r}_2 \, \mathrm{d}\boldsymbol{r}_3 \ldots \mathrm{d}\boldsymbol{r}_N \, |\Psi(\boldsymbol{R})|^2, \qquad (31)$$

and similarly for particle $i = 2, 3 \ldots, N$. All particles are indistinguishable, so which i we consider is arbitrary.

The one-body density gives a measure of the distribution of particles in the system. It can be read as: if we marginalize out the positions of all other particles, what is the likelihood of finding particle i in a given position? The answer to this question is given by $\rho_(r_i)$.

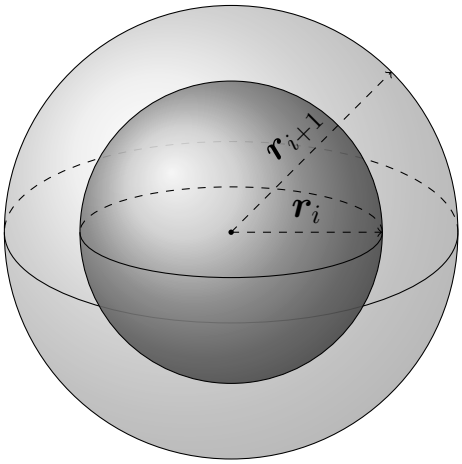
For the non-interacting case we can evaluate this expression exactly, as the wavefunction separates nicely.

$$\begin{aligned} \rho(\boldsymbol{r}_k) &= g(\boldsymbol{r}_k) \prod_{i \neq k} \iiint_{-\infty}^{\infty} \mathrm{d}x_i \, \mathrm{d}y_i \, \mathrm{d}z_i \, g(\boldsymbol{r}_i) \\ &= \left(\frac{\pi^{3/2}}{2\sqrt{2}\alpha^{3/2}\sqrt{\beta}} \right)^{N-1} g(\boldsymbol{r}_k) \end{aligned} \qquad (32)$$

The constant is of little importance to us, but what we want is the functional form of $\rho(\boldsymbol{r}_k$, which we see follows the Gaussian shape of the single-particle wavefunction.

In the non-interacting case this integral is far less friendly to us, and we have no exact formula or shape. We shall therefore compute the integral numerically. However, instead of going about this with straight forward Monte Carlo integration, we shall take a more intuitive approach.

We split the position space into M spheres[1] of increasing radii, $r_i = i * r_{\text{step}}$ for $i = 1, \ldots, M$, all centered at the origin. The volume of the sphere with radius r_{i+1}, minus the volume of the sphere with radius r_i, we call region $i + 1$. The illustration below shows this graphically, where region $i+1$ is the lightly shaded region.



For each sampled position $\boldsymbol{R}$, we can note down which region particle 1 (arbitrary choice) is within, and keep a running tally for each region. After sampling many positions, we normalize each region count by both the number of samples and the volume[2] of the region. Given a small enough r_{step}, and enough samples, we would then plot the normalized region counts against their radii, and this would be expected to approximate the one-body density well.

As a technical note, because the choice of which particle we observe is arbitrary, we could have done this for any other particle. We could therefore do it for all particles in parallel, by noting which region *every* particle lands in. Since they should all produce a similar result, we could average the recorded counts across all particles. This should be roughly the same as if we had N times as many samples and only counted for one particle.

3 Algorithms

In (13) we reformulated our objective of estimating the ground state energy of the system, $\langle H \rangle$, to minimizing the average local energy, $\langle E_L(\boldsymbol{\alpha}) \rangle$ w.r. to the variational parameters $\boldsymbol{\alpha} = (\alpha, \beta)$. Key to this reformulation was that the local energy had to be

[1] Or circles in 2D, and line segments in 1D.

[2] Or area in 2D, and length in 1D

sampled with positions $\boldsymbol{R}$ that where drawn from the PDF $P(\boldsymbol{R}, \boldsymbol{\alpha})$. As P is far from any standard PDF with a known inverse CDF, we cannot trivially generate such samples. In addition, the normalisation term in its denominator is computationally expensive to compute. These limitations make the *Metropolis algorithm* the obvious choice to solve this problem. This algorithm provides a way to generate random samples from a PDF where we only know the probabilities up to a proportionality constant.

3.1 Metropolis Algorithm

The algorithm in its simplest form is described in Algorithm 2.

Algorithm 2 The Metropolis algorithm in its simplest form, as it pertains to our specific application.

Require: M, generates $M \times N$ samples.
Ensure: samples $\overset{\text{d}}{\leftarrow} P(\boldsymbol{R}, \boldsymbol{\alpha})$

```
samples ← empty list
R ← randomly initialized matrix of positions
for M iterations do
    for every particle i ∈ [1, N] do
        Δr ← random perturbation vector
        R* ← R
        R*_i ← R*_i + Δr
        q ← |Ψ_T(R*)|² / |Ψ_T(R)|²
        r ←(d) Unif(0, 1)
        if r ≤ q then
            R ← R*
        end if
        Append R to samples
    end for
end for
```

In this algorithm we move around in position space randomly, accepting new positions biased towards areas of space where $P(\boldsymbol{R}, \boldsymbol{\alpha})$ is higher. We choose to move one particle at a time based on computational efficiency, as recalculating the local energy can be done more easily when we know only one particle has moved[3].

[3]Not actually implemented yet.

With the generated list of positions we may produce and average for the local energy, and therefore an estimate for an upper bound on the ground state energy, as well as the one-body density, or any other quantity of interest.

3.1.1 Limitations

This algorithm has two major drawbacks. Firstly, the samples generated are not independent. It is quite clear from the algorithm that the probability of drawing a certain position is highly dependent on what position we were at previously. The has implications on how we perform our statistical analysis, as we must attempt to correct for this limitation. More on this in section 4.

Secondly this algorithm will be quite ineffective in that a significant portion of the suggested moves (new positions, $\boldsymbol{R}^*$ in Algorithm 2) will be rejected. This is because the new positions are generated at random, which might cause us to wander around in regions of position space that are of very little significance, and it might take a while before we (by chance) stumble upon a more high-probability region. The effect is then a list of samples that may not be an accurate representation of the PDF we were trying to approximate to begin with. This will be especially true for smaller sample sizes, were these defects will account for a larger proportion of the samples.

3.2 Metropolis-Hastings Algorithm - Including Importance Sampling

The first limitation of the simple algorithm (not i.i.d.) is inherent to this kind of sampling, and is hard to avoid. We may, however, attempt to remedy the second limitation by guiding the random walker towards more promising regions of position space by proposing new positions in a smarter way than purely randomly.

3.2.1 Physical Motivation of Results

We will limit our selfs to a superficial derivation, focusing only on giving a physical motivation for the

results we end up using, as it is outside the scope of this project to derive this rigorously.

Better generation of new positions:
A reasonable assumption is to say that particles will tend towards regions of space where $|\Psi_T|^2$ is larger. We may say that this is the result of a force, namely the quantum drift force given in (29), as we know this force pushes particles towards regions where Ψ_T is large. In addition, as this a quantum system, we expect some degree of random motion as well. This intuitive view is exactly what is described by the Langevin equation,

$$\frac{\partial \boldsymbol{r}_k}{\partial t} = D\boldsymbol{F}_k(\boldsymbol{r}_k) + \boldsymbol{\eta}, \tag{33}$$

which describes how the position of a particle changes with time under the influence of a drift force and random impulses. Here, D is a constant scalar referred to as the drift coefficient. We set $D = 1/2$, originating from the same factor in the kinetic energy. The term $\boldsymbol{\eta}$ is a vector of uniformly distributed random values, giving the particle some random motion in each dimension.

Solving the Langevin equation we can obtain new positions at some time $t + \Delta t$. Using Euler's method we get:

$$\boldsymbol{r}^* = \boldsymbol{r} + \frac{1}{2}\boldsymbol{F}_k(\boldsymbol{r}_k)\Delta t + \boldsymbol{\xi}\sqrt{\Delta t}, \tag{34}$$

given a time step Δt, and where $\boldsymbol{\xi}$ is the normally distributed equivalent of $\boldsymbol{\eta}$.

Adjusting the acceptance probability:
In Algorithm 2 the acceptance probability for a new position $\boldsymbol{R}^*$ was

$$q(\boldsymbol{R}^*, \boldsymbol{R}) = \frac{|\Psi_T(\boldsymbol{R}^*)|^2}{|\Psi_T(\boldsymbol{R})|^2}. \tag{35}$$

This was based on the transition probability for going to $\boldsymbol{R}^*$ from $\boldsymbol{R}$ being uniform. When the transition probabilities are different depending on where we are and where we want to go, the acceptance probability has to be modified in order for the algorithm to work as advertised. In general, we have the following form for $q(\boldsymbol{R}^*, \boldsymbol{R})$ in our case:

$$q(\boldsymbol{R}^*, \boldsymbol{R}) = \frac{T_{j\to i}}{T_{i\to j}} \frac{|\Psi_T(\boldsymbol{R}^*)|^2}{|\Psi_T(\boldsymbol{R})|^2}, \tag{36}$$

where $T_{i\to j}$ is the transition probability from the original state i into the new state j. We see that a uniform transition probability gives us back (35). We need therefore an expression for the transition probabilities when we use the improved generation of new positions.

To this end, we consider the *Fokker-Planck* equation, which for one dimension and one particle can be written as

$$\frac{\partial \Psi_T}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} - F\right)\Psi_T, \tag{37}$$

which describes the time-evolution of a probability distribution under the influence of a drift force and random impulses. We let Ψ_T play the role of the probability distribution. The factor D is as before, and F is here the one-dimensional, one-particle analog to $\boldsymbol{F}$. In fact, this equation is the origin of the specific form of $\boldsymbol{F}$ presented in (29).

Equation (37) yields a solution given by the following Green's function (for one particle):

$$G(\boldsymbol{r}_k^*, \boldsymbol{r}_k, \Delta t) \propto \exp\left[-\frac{\|\boldsymbol{r}_k^* - \boldsymbol{r}_k - D\Delta t\boldsymbol{F}_k(\boldsymbol{r}_k)\|^2}{4D\Delta t}\right]. \tag{38}$$

This is interpreted as the probability of transitioning to position $\boldsymbol{r}_k^*$ from position $\boldsymbol{r}_k$ in a time interval Δt. Replacing the transition probabilities in Equation (36) we get the new acceptance probability:

$$q(\boldsymbol{R}^*, \boldsymbol{R}) = \frac{G(\boldsymbol{r}_k, \boldsymbol{r}_k^*, \Delta t)}{G(\boldsymbol{r}_k^*, \boldsymbol{r}_k, \Delta t)} \frac{|\Psi_T(\boldsymbol{R}^*)|^2}{|\Psi_T(\boldsymbol{R})|^2}, \tag{39}$$

3.2.2 Improved Algorithm

We are now ready to present the proper Metropolis-Hastings algorithm, with importance sampling used to increase the number of accepted transitions.

Algorithm 3 The Metropolis-Hastings algorithm, as it pertains to our specific application.

Require: M, generates $M \times N$ samples.
Ensure: samples $\overset{\text{d}}{\leftarrow} P(\boldsymbol{R}, \boldsymbol{\alpha})$

```
samples ← empty list
R ← randomly initialized matrix of positions
for M iterations do
    for every particle i ∈ [1, N] do
        Δr*_i ← ½ F_i(r_i) Δt + ξ √Δt
        R* ← R
        R*_i ← R*_i + Δr*_i
        q ← |Ψ_T(R*)|² / |Ψ_T(R)|²
        q ← q × G(r_k, r*_k, Δt) / G(r*_k, r_k, Δt)
        r ← Unif(0, 1)
        if r ≤ q then
            R ← R*
        end if
        Append R to samples
    end for
end for
```

3.3 Optimizing the Variational Parameters

The point of VMC is to vary the parameters of the wavefunction, here α and β, in such a way that the energy is minimized. This is in other words a classic optimization problem, to which we can apply one of the many available techniques. One of the simpler and more intuitive approaches is Gradient Decent, which is the one we will apply. We could have[4] opted for a better method, such as Conjugate Gradient, but standard Gradient Decent was chosen for simplicity.

3.3.1 Gradient Decent

The intuition for Gradient Decent (along with most other techniques) is based on the fact that extrema, $\boldsymbol{x}^*$, of a function $f(\boldsymbol{x}^*)$ have the property of zero gradient, $\boldsymbol{\nabla} f(\boldsymbol{x}^*) = 0$. Using this, and starting at any initial guess $\boldsymbol{x}_0$, we can update this guess to a point with *smaller* gradient by following the direction of $-\boldsymbol{\nabla} f(\boldsymbol{x}_0)$. In less words, the improved guess $\boldsymbol{x}_1$ is given by

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \gamma \boldsymbol{\nabla} f(\boldsymbol{x}_0), \tag{40}$$

where γ is a hyper-parameter referred to as the *learning rate*, and typically $\gamma < 1$. Following the same recipe, we can generate arbitrarily small gradients by iterating

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma \boldsymbol{\nabla} f(\boldsymbol{x}_i). \tag{41}$$

We are guaranteed, given a sufficiently small γ and enough iterations, to arrive at *a* minimum with this technique. As with all numerical optimization schemes, we are always having to consider the possibility of the minimum only being local, and not the global one.

Defining the Gradient:

In our case, the function we want to minimize is $E_0 = \langle H \rangle$. The only parameter we will treat as varying in this report is α, as we shall fix β later on. Therefore, the gradient is

$$\boldsymbol{\nabla}_\alpha \langle H \rangle = \frac{\partial \langle H \rangle}{\partial \alpha} \tag{42}$$

With some algebraic massage we can rewrite this as follows [2]:

$$\boldsymbol{\nabla}_\alpha \langle H \rangle = 2\left[\left\langle \frac{E_L}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} \right\rangle - \langle E_L \rangle \left\langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} \right\rangle \right] \tag{43}$$

The derivative of the wavefunction with respect to α is found easily,

$$\frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} = -\sum_i^N \left(x_i^2 + y_i^2 + \beta z_i^2 \right) \tag{44}$$

as dividing by Ψ_T cancels the interaction factor. For 1D and 2D we simply drop the other coordinate terms.

We can now compute the two expectation values in (43) from the sampled positions $\boldsymbol{R}_j$, and get an approximation for the gradient.

[4] Perhaps even should have, in hindsight.

Limitations:
Gradient Decent is wonderfully simple, but has its limitations. Firstly, the results are highly dependent on on the hyper-parameters, which are the initial guess $\boldsymbol{x}_0$ and the learning rate.

A bad guess can mean we spend a long time finding our way to the minimum. A bad guess also increases the likelihood of getting stuck in local optima on the way.

The learning rate must also be just right. To low and we take forever to get anywhere, and we very easily get stuck in small bumps in the function output space. Make γ to large and we have trouble actually converging to anything. We may also end up with exploding gradients if we are so unlucky.

All in all there are tons of pitfalls, but we hope our energy function is relatively well behaved, and that we can provide good values for the hyper-parameters.

4 Evaluating the Validity of Results

Once one of the above algorithms has been used to obtain numerical results, these are of little use to us as scientist if we know nothing about the certainty with which we can trust our the answers we get. To that end we would like to make an estimate of the magnitude of the error in our results.

There are two main sources of errors that enter our results, namely 1) systematic error and 2) statistical error. The systematic error comes from our assumptions about how we model the quantum mechanical system, i.e. the form of the Hamiltonian, the ansatz for our wavefunction, the hyper-parameters used in the computation etc. This error is intrinsic to our approach and is hard to both quantify and avoid.

More of interest to us is therefore the statistical error, as we *can* in fact make an estimate for this.

4.1 Standard Estimate of the Standard Error of the Mean

The quantity we are looking to estimate is the standard error of the mean local energy, typically defined as

$$\sigma_{\bar{E}_L} = \sqrt{\frac{\sigma^2_{E_L}}{n}} = \sqrt{\frac{1}{n}\Big[\langle E_L^2\rangle - \langle E_L\rangle^2\Big]}, \tag{45}$$

where n denotes the number of samples[5], and $\sigma^2_{E_L}$ is the true population variance for the local energy. We do not know the underlying population variance, and so we estimate this by the sample variance instead:

$$\hat{\sigma}_{\bar{E}_L} = \sqrt{\frac{\hat{\sigma}^2_{E_L}}{n}} = \sqrt{\frac{1}{n}\Big[\widehat{\langle E_L^2\rangle} - \bar{E}_L^2\Big]}. \tag{46}$$

There is, however, a complication that should be taken into account when we estimate this quantity, namely that our E_L samples will not be drawn independently. The above estimation for the standard error is based on the central limit theorem, and key to this is the assumption that the samples should be independently drawn.

Inherent to the Monte Carlo sampling techniques is that the value of a sample at one point in time is in fact correlated to the sample before it. Figure 1 shows an example of consecutive samples obtained using Algorithm 3. Although the values vary quite a bit, we still see that if the values are very low it takes some time before it can get large again, and vice versa.

The effect of this is that using (46) directly as the measure for the statistical error will tend to *underestimate* its true value.

4.2 Adjusting for Correlated Samples

We can make the error estimate in (46) less biased by replacing the sample variance by the sample *covariance*:

$$\begin{aligned}\hat{\sigma}_{\bar{E}_L} &= \sqrt{\frac{\mathrm{Cov}(E_L)}{n}} \\ &= \sqrt{\frac{1}{n^2}\sum_{i,j}\big(E_{L,i} - \bar{E}_L\big)\big(E_{L,j} - \bar{E}_L\big)},\end{aligned} \tag{47}$$

[5]Upper case N will be used for the number of particles in the system, lower case n for the number of samples.

which we can rewrite in terms of the uncorrelated and correlated contributions to the error as follows:

$$\hat{\sigma}^2_{\bar{E}_L} = \frac{\hat{\sigma}^2_{E_L}}{n} + \frac{2}{n^2}\sum_{i<j}\big(E_{L,i} - \bar{E}_L\big)\big(E_{L,j} - \bar{E}_L\big). \tag{48}$$

We observe that if the energies are uncorrelated in time, we would just get back (46).

To simplify the notation on the way to the end result, we define following two functions:

$$f_d = \frac{1}{n-d}\sum_{k=1}^{n-d}\big(E_{L,k} - \bar{E}_L\big)\big(E_{L,k+d} - \bar{E}_L\big) \tag{49}$$

$$\kappa_d = \frac{f_d}{\hat{\sigma}^2_{E_L}}. \tag{50}$$

The function f_d describes the correlation of samples spaced by d. We have $f_0 = \sigma^2_{E_L}$, and for an uncorrelated system we have $f_d = 0 \;\forall\; d > 0$.

Now (48) can be written as

$$\begin{aligned}\hat{\sigma}^2_{\bar{E}_L} &= \frac{\hat{\sigma}^2_{E_L}}{n} + \frac{2\hat{\sigma}^2_{E_L}}{n}\sum_{d=1}^{n-1}\kappa_d \\ &= \frac{\tau}{n}\,\hat{\sigma}^2_{\bar{E}_L},\end{aligned} \tag{51}$$

where we have defined the *autocorrelation time* τ,

$$\tau = 1 + 2\sum_{d=1}^{n-1}\kappa_d, \tag{52}$$

which can be interpreted as the spacing between samples for which we no longer observe any correlation. For a completely uncorrelated sample set we have $\tau = 1$, which again results in (46). Also, any $\tau > 1$ will lead to an *increase* in the estimated error.

This new estimate of the error will be a much more realistic estimate. However, computing the autocorrelation time τ has time complexity $\mathcal{O}(n^2)$, which will prove unfeasible when we plan on producing millions of samples.

4.3 Methods for Improving the Error Estimate

We look now at two methods of obtaining improved estimates for the standard error, besides computing the autocorrelation time τ directly.

We will look at two such tools here, 1) Bootstrap and 2) Blocking.

4.3.1 Bootstrap

The best way to improve statistical results is usually to simply generate more data, as statistical estimates tend towards their true value as the number of samples grow. So ideally, we would simply generate many more sample sets, $\{E_L\}_b \quad \text{for} \quad b = 1, 2, \ldots, B$, compute the mean for each and produce a histogram of the results. That would then, for a sufficiently large B, serve as a good estimate for the probability distribution for $\bar{E}_L$. We could then compute the standard deviation of said distribution and call that the standard error of the mean local energy. This would not strictly address the problem of correlation directly, but averaging results over more data would lessen the effect. Also, we would expect the individual sample sets to be uncorrelated with each other, as long as they had differing random seeds.

In practice this is not feasible though, as producing each set of energies will be a sizable computational task in it self, and not something we can repeat several times. This is were Bootstrap comes in. The idea of Bootstrap is to simulate the production of new sample sets by resampling, with replacement, from the original data set. From the original set $\{E_L\}_1$ we draw n samples with a uniform probability to get new sets $\{E_L\}_b \quad \text{for} \quad b = 2, 3, \ldots, B$.

As long as the original data set is a representative sample of the space of energy values (i.e. not biased towards large/small values), and we use a large enough value for B, this will produce a good approximation of the statistical error.

This approach has to main limitations. First, it is computationally expensive to do when the size of the data set is very large, and we need a large B value.

Second, Bootstrap is still meant to be used on iid. samples, and any estimates for the error we get from it will still be an underestimate. This comes from the fact that the original sample set won't be a fully representative sample of the space, due to the correlations within it. Using Bootstrap will improve the estimate, but it will still be biased.

4.4 Blocking

In some sense Blocking takes the complete opposite approach to that of Bootstrap. Blocking takes the original samples and combines them to make less data points.

The idea is to group consecutive samples into *blocks* of a size such that sample i in block j is uncorrelated with sample i in block $j+1$. Then, treating each block as a single sample (i.e. by computing the average energy in each block), we have independent samples of the mean.

The only issue with this approach is determining how large the blocks need to be in order for the blocks to become independent. A natural choice would be the autocorrelation time τ, but that just brings us back to the original problem of computational infeasibility.

A simple, empirical solution to finding a suitable block size is to make a plot of the standard deviation of the block means as a function of block size. Because correlated samples meant we *underestimated* the error (aka. standard deviation of the sample mean), increasing the block size (i.e. making the samples less correlated) should increase the observed error. This would keep going until the block size is so large that the samples have become independent, at which points it should plateau. Choosing the block size at the plateau point would then be optimal.

Even better would be an automated way to determine the ideal block size. Such methods do exists, and will for simplicity be used in this project, although the manual way would work. The derivation of how this works, however, will not be discussed any further in this report.

5 Results for Non-Interacting System

5.1 Standard Metropolis Sampling

5.1.1 Comparison of Configurations of Parameters

We now apply the standard Metropolis sampling algorithm described in Algorithm2 to a variety of different settings. Table 1 shows selected results obtained using $\alpha = 1/2$, which corresponds to running with the exact wavefunction.

We see that all the results have $\langle E_L \rangle$ equal (within numerical precision) to the exact result, following (14). In addition, the variance is also equal to or extremely close to zero, as we expect to observe when we use the correct form for the exact wavefunction. This is strong evidence that the codes developed are correct, at least for the non-interacting case.

We see a rather strong dependence on the step size used in the acceptance rates. It appears we need smaller step sizes when N increases. We would like to tune the step size such that we have a significant portion accepted, but still not run up into the$\sim 100\,\%$ region. This is because at this point we are just stuck in one place, with no real movement. This is not an issue in the ideal case, because the energy here is actually not dependent on the positions. But as soon as we move out of this ideal case, this will matter a great deal. We conclude that a step of 1 works well in most cases, but must be lowered a bit for systems with $N \geq 100$.

Perhaps most interesting is to look at the time spent on each configuration. If we compare the running times for $N = 100$ and $N = 500$ with analytical expressions turned off, we see the time increasing by a factor of ~ 25, which is the square of the factor we increased N by. Looking at the expressions involved in Algorithm 2, we see that this is exactly what we would expect, namely that the algorithm has time complexity $\mathcal{O}(N^2)$.

Similarly, comparing the run times when analytical expression is turned on, we observe that time scales as $\mathcal{O}(N)$, resulting in a significantly reduced run times. It is quite clear that the use of analytic expressions is far superior. We can, however, expect the difference between the to approaches to be reduced when we include interaction, as this reduced the analytical approach to the same complexity as the numerical one.

5.1.2 Varying the Variational Parameters

Figure 2 shows a plot of the expected energy, and corresponding variance, for a $1D$ system of one par-

ticle, using $5 \cdot 10^4$ Monte Carlo cycles, as a function of the variational parameter α.

We can see a minimum in both graphs, both centered around the ideal value $\alpha = 1/2$. The energy graph is, however, a bit jagged, and the actual minimum has been found to be a little left of what we would expect. The curve gets smoother as we increase the number of samples, and extending to 10^5 samples yield a 'perfect' graph, and this is easily within our computational reach. The relatively low value of $5 \cdot 10^4$ was used to give a visual difference to the result in the next section, when we include importance sampling.

5.2 Metropolis-Hastings Sampling

5.2.1 Comparison of Configurations of Parameters

We now apply the improved Metropolis-Hastings sampling algorithm described in Algorithm3 to a variety of different settings. Table 2 shows selected results obtained using $\alpha = 1/2$, which corresponds to running with the exact wavefunction.

We see once again that all the results have $\langle E_L \rangle$ equal (within numerical precision) to the exact result, and very low variance, indicating that the implementation works as it should.

Looking at the acceptance rates now we see $\sim 100\,\%$ acceptance of new moves, indicating that the proposed moves are quite probable and therefore more likely to be accepted by the sampling algorithm. When Δt is to large, however, the result become unstable, as before. At this point the randomness in (34) dominates, with steps larger than what allows for convergence. We may tune this parameter as we wish, so choosing a value that is small enough for the acceptance rate to be close to one, but also not so small that we would need an excessive amount of MC cycles in order to have any exploration in space. We will once again have to tune this parameter depending on what system we look at, as we need a smaller step for larger systems.

The run times reported have not changed in any significant way, as would be expected, as adding importance sampling did not change the complexity of the algorithm.

5.2.2 Varying the Variational Parameters

Figure 3 shows a plot of the expected energy, and corresponding variance, for the same system and number of MC cycles as that shown in Figure 2. In this version we have enabled importance sampling.

We can see a very similar result as before, with the notable exception that the line is less jagged. This implies that the sampling results are more stable. This is due to the improved exploration of the integration space which importance sampling gives us. We conclude therefore that the improved algorithm allows us better, more stable results, compared with the simple sampler. This allows us to compute less MC samples in order to obtain good enough results, something which is highly desired for an application where we are constantly fighting against the large time complexity of this kind of system.

To get a better view of the effect of Δt on the Metropolis-Hastings sampling algorithm, Figure 4 shows the same graph as before, but for a set of different values for Δt. From this plot it becomes very clear that choosing the correct value for this parameter will be very important when we try to find the optimal variational parameters. We see that the apparent minima appear on both the left and the right of the ideal value.

All values are to some extent usable given enough MC samples, but an unrealistic number might be needed for Δt values far away from the optimum. Figure 4 provides further evidence that Δt must be chosen with care.

5.3 Statistical Error

Figure 5 shows the results of a simulation on a system of one boson in 3D, using $n = 2^{14}$ Monte Carlo samples. On the graph, error bars are included to visually express the certainty with which we know the results. The magnitude of the error bars are given by the standard error, as determined by the Blocking method. We see that the error estimate goes promptly to zero as we approach the optimal $\alpha = 1/2$ value, and in fact we get exactly zero as the standard error at this

point. This is not that surprising, as using the optimal value for α here means we are solving the exact system, and therefore there should not be any doubt about the answer either.

The error estimates might be of higher importance as we traverse into the interacting case.

5.4 One-Body Density

It is also interesting to see how the calculated one-body density comes out for the ideal non-interacting case. Figure 6 shows the plot for a system of 10 bosons in the 3D symmetric and elliptical harmonic oscillator traps. Also drawn is the expected shape from (32), for the symmetric curve. All lines have been normalized so that they integrate to one, as the constants are irrelevant to us here. To our delight the approximation agrees very well with the expectation. The discrepancies around $r = 0$ are due the volumes of the smallest regions being very small, and hence their occupation counts become somewhat unstable. We can still make out the trend though, which again is in great agreement with the analytical result.

The elliptical curve deviates from that of the symmetrical (not surprising), and we see that the effect is that particles tend to be closer to the origin. This is understandable, as the parameters used here increase the z-component of the potential by a factor $\sqrt{8}$. With a stronger confining potential, we would expect exactly this behaviour

5.5 Optimization via Gradient Decent

Even though we know the optimal value for α in the non-interacting case, it will serve as a nice implementation test to verify that the algorithm is capable of finding the minimum. Looking at Figure 2 and 3 we see that the energy function has a very simple shape, with a single well defined minimum. Therefore we should expect to be able to find the optimal α, even with not so good guesses.

We will only state the observed results here. Many of them are reproduced in the unit tests of the Gradient Decent method, if a source is desired.

We find that Gradient Decent is in fact able to find the minimum, and that to arbitrary precision, given enough iterations. In the tests we used 10^4 Monte Carlo samples per iteration, which proved quite sufficient. This indicates that the implementation works.

However, we observe it to be quite unstable with respect to the initial guess. It works fine with guesses around $\alpha_0 \in (0.3, 1)$, but if we start any further away than this the results are unstable. This is likely due to the gradient and variance increasing rapidly when we stray far from the optimum (as seen in e.g. Figure 2).

6 Results for Interacting System

We now turn our attention to the elliptical trap with the repulsive interaction activated. We shall in this report fix $a/a_{ho} = 0.0043$, and $\beta = \gamma = 2.82843 \simeq \sqrt{8}$, as done in [3, 1]. We will consider systems with $N = 10, 50$ and 100 bosons.

6.1 Varying the Variational Parameter

Table 3 shows results for all systems, using α values in $[0.4, 0.6]$. Figure 7 Shows the corresponding data in graphical form. We observe, as expected that the energy per particle increases with increasing N. This makes sense, as there are more bosons interacting, and hence the energy should also be higher.

Comparing our results to the exact solution for the non-interacting case, we can see that this is consistently

The resolution of the data is too low to make any certain claims on where the optimal value for α might be (it might even be outside the range tried here). However with the data available we are suggested to believe that some minimum exists in the range $\alpha \in [0.5, 0.6]$.

6.2 Employing Gradient Decent

After the initial scan made in the previous section Gradient Decent was applied to the $N = 10$ system. The full result is available on Github. The search was started at $\alpha = 0.55$, based on suspicions from

the previous section. The algorithm was given 10^4 samples per iteration, and a learning rate of 0.001.

Interestingly the algorithm did not agree with the 'findings' of the last section. Quite quickly the algorithm reduced α, and landed finally on

$$\alpha_{N=10} \simeq 0.4978 \tag{53}$$

The corresponding energy, calculated with 2^{20} samples was then found to be:

$$E_{N=10} \simeq 24.3984 \pm 0.00080. \tag{54}$$

Ideally we would repeat the process for $N = 50$ and $N = 100$. However, due to the bad complexity in N, the time needed to do this is more than was available at the time. To achieve results in reasonable time, we would need to lower the amount of samples per iteration significantly. Trying to do this however has proved to yield unstable results for the gradients, and hence the algorithm is not able to converge.

6.3 One-Body Density

Figure 8 shows the one-body densities for each of the systems, with the ideal symmetrical curve for reference as before. We see a trend very similar to that of Figure 6, in that the elliptical curve lines indicates that the stronger potential squeezes the particle closer to the origin, compared with the interacting case.

We would maybe expect the presence of interactions to spread the one-body density out more, as the particles may not be allowed to be to close. There is however not any clear visible difference to make out from these plots. The jaggedness around the origin makes it a bit difficult to make out any potential differences. We could have improved the resolution of these plots once again by using more samples. However, there are limits to what we have CPU capacity to do, and so the present data must suffice.

7 Conclusion

We have seen how a Variational Monte Carlo approach was able to accurately reproduce the expected results for the simple, non-interacting system of bosons in a harmonic oscillator potential. We could see this in both the calculated energies, and in the one-body density.

We have looked at the differences in sampling technique, namely the difference between the standard Metropolis algorithm, and the physically motivated version of Metropolis-Hastings algorithm. We observed how using importance sampling was able to produce significantly better results in terms of stability and accuracy, for instance seen in the difference between Figure 2 and 3. We also saw evidence that choosing an appropriate step size was of great importance, and could greatly influence how the algorithms perform.

Finally for the non-interacting system we implemented Gradient Decent, which was able to easily reproduce the ideal α, although this was largely dependent (again) on appropriate hyper-parameters.

In the interacting regime we first and foremost observed the $\mathcal{O}(N^3)$ time complexity of our numerical approach. This made calculations computationally very expensive to do, and heavily influenced the resolution of the results we were able to produce.

The calculated energies per particle increased as N increased, in accordance with expectations. By comparing the energies we found to (17), we were able to see by how much the energies increased relative to the ideal case. The effect was small for $N \leq 10$ particles, but became quite apparent for larger N.

Results for the one-body density did not show any noticeable effects of the interaction. However, whether this is actually the case or not is hard to conclude from the precision we have obtained.

Gradient Decent was applied to $N = 10$, with satisfactory results, obtaining an energy which has not been subsided by any other parameter value. Due to time constraints, and heavy CPU demand, Gradient decent was not successfully implemented for the $N = 50, 100$ systems.

8 Further Work

It would be interesting to spend some more time optimizing the code, including expanding to make use of MPI for parallelization. It would be very interest-

ing to see if the implementation was able to find good numbers for the larger systems.

Due to a regrettable mistake in time estimation, rooted in unforeseen coding complications, the resolution of the results for the interacting case are not as good as desired. Regardless of the conclusions made here, I therefore intend to revisit this project, at least by letting some week-long jobs run once the code has gotten a little optimization facelift.

References

[1] J.K. Nilsen et al. "Vortices in atomic bose-einstein condensates in the large-gas-parameter region". In: *Physical Review A* 71.5 (2005), p. 053610.

[2] Morten Hjorth-Jensen. *Computational Physics 2: Variational Monte Carlo methods, Lecture Notes.* http://compphysics.github.io/ComputationalPhysics2/doc/web/course. Spring 2018.

[3] J.L. DuBois and H.R. Glyde. "Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis". In: *Physical Review A* 63.2 (2001), p. 023602.

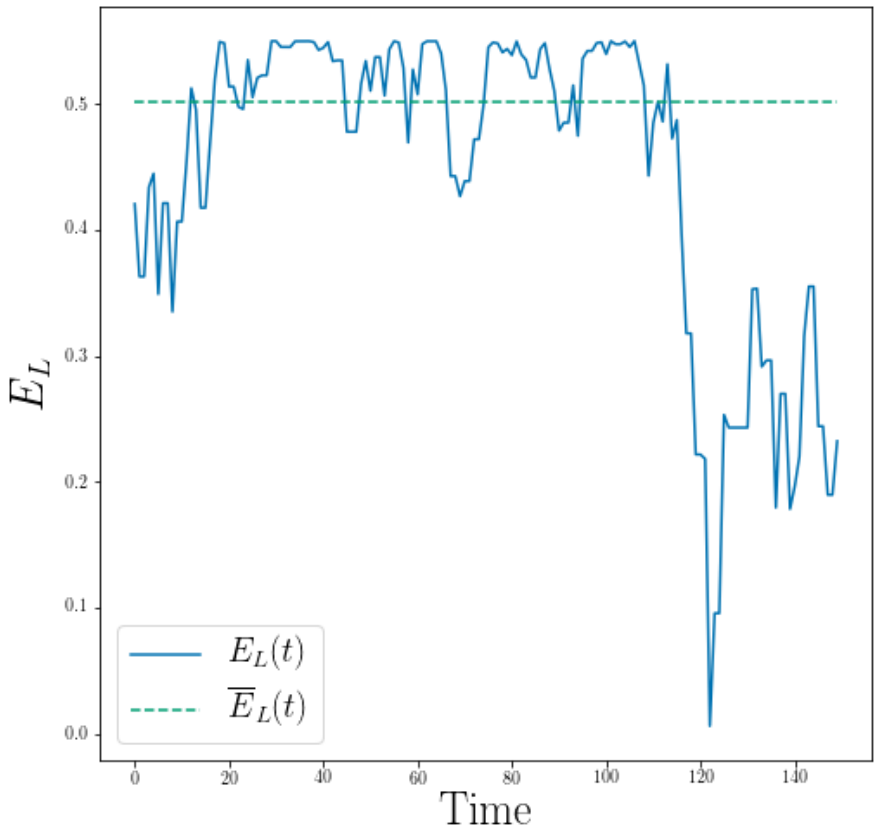


Figure 1: Example time series of local energy values obtained from a simulation using Algorithm 2. Energies shown are a small snippet of energies obtained for a system of one boson in one dimension, using $\alpha = 0.55$ and a step size of 1. The mean from the entire set of samples is shown for reference

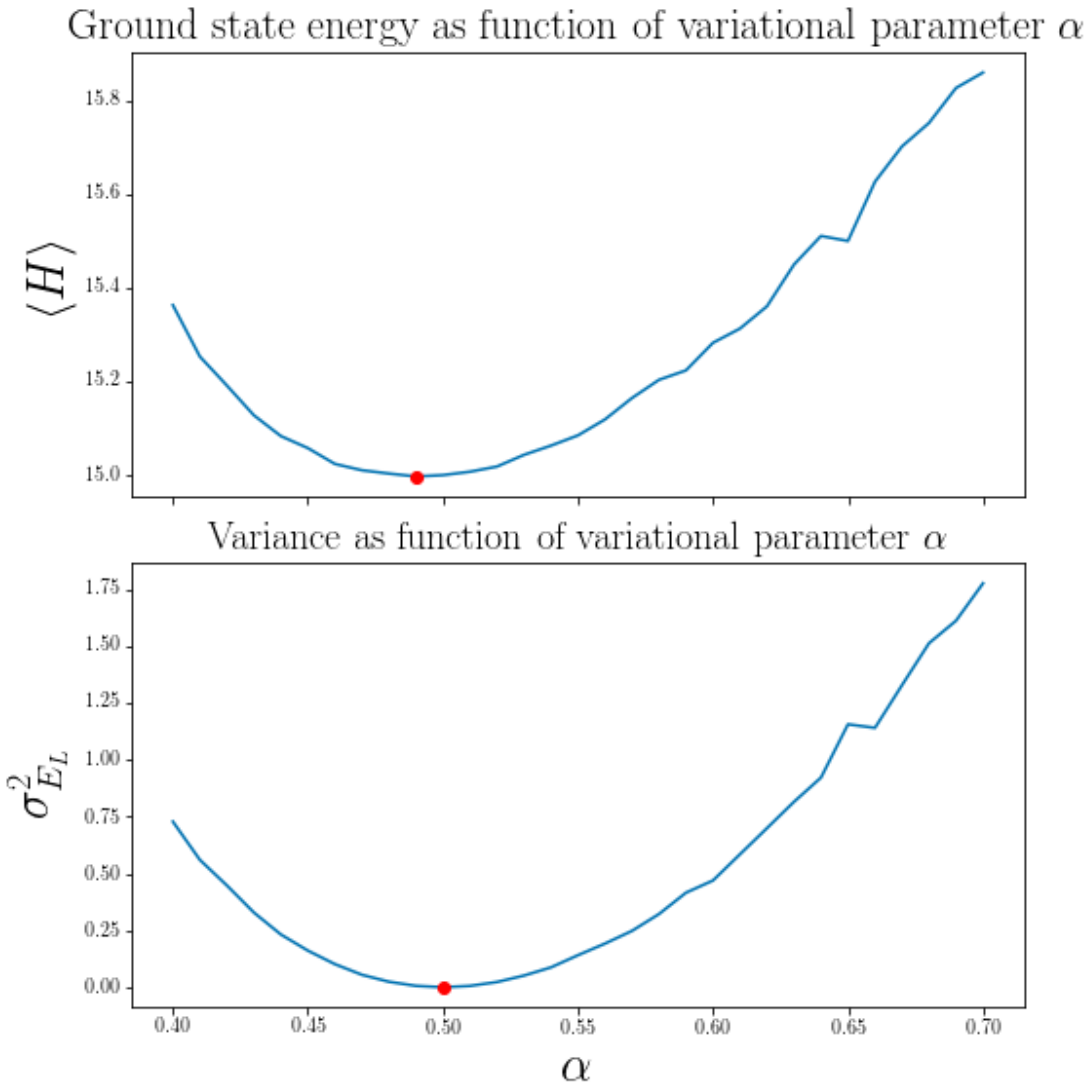


Figure 2: Plot of average local energy and accompanying population variance, as functions of the variational parameter α. The minimum of both graphs are marked, and is expected to be placed at $\alpha = 1/2$. Results shown are for $1D$, one particle and $5 \cdot 10^4$ Monte Carlo cycles, using Algorithm 2.

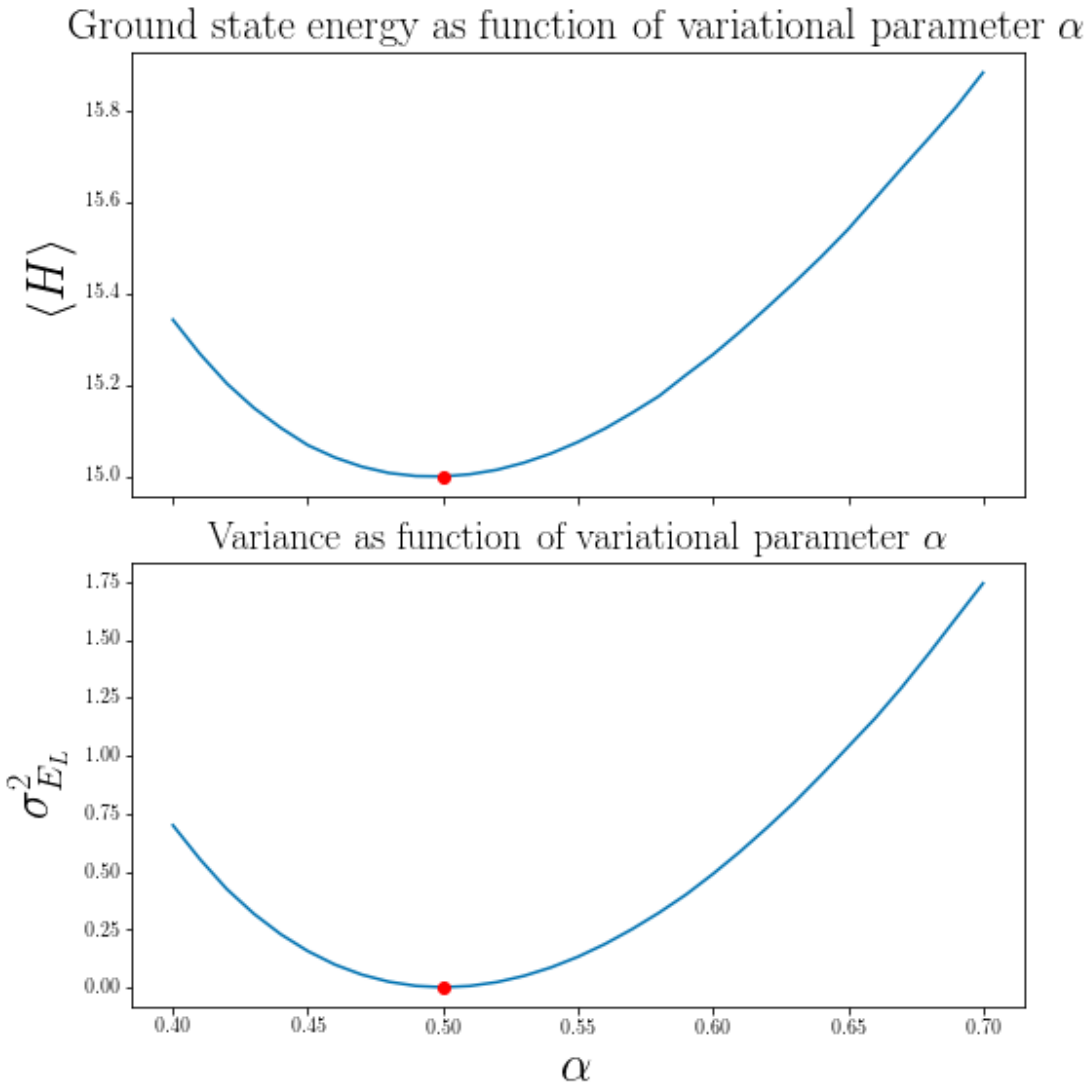


Figure 3: Plot of average local energy and accompanying population variance, as functions of the variational parameter α. The minimum of both graphs are marked, and is expected to be placed at $\alpha = 1/2$. Results shown are for $1D$, one particle and $5 \cdot 10^4$ Monte Carlo cycles, using Algorithm 3

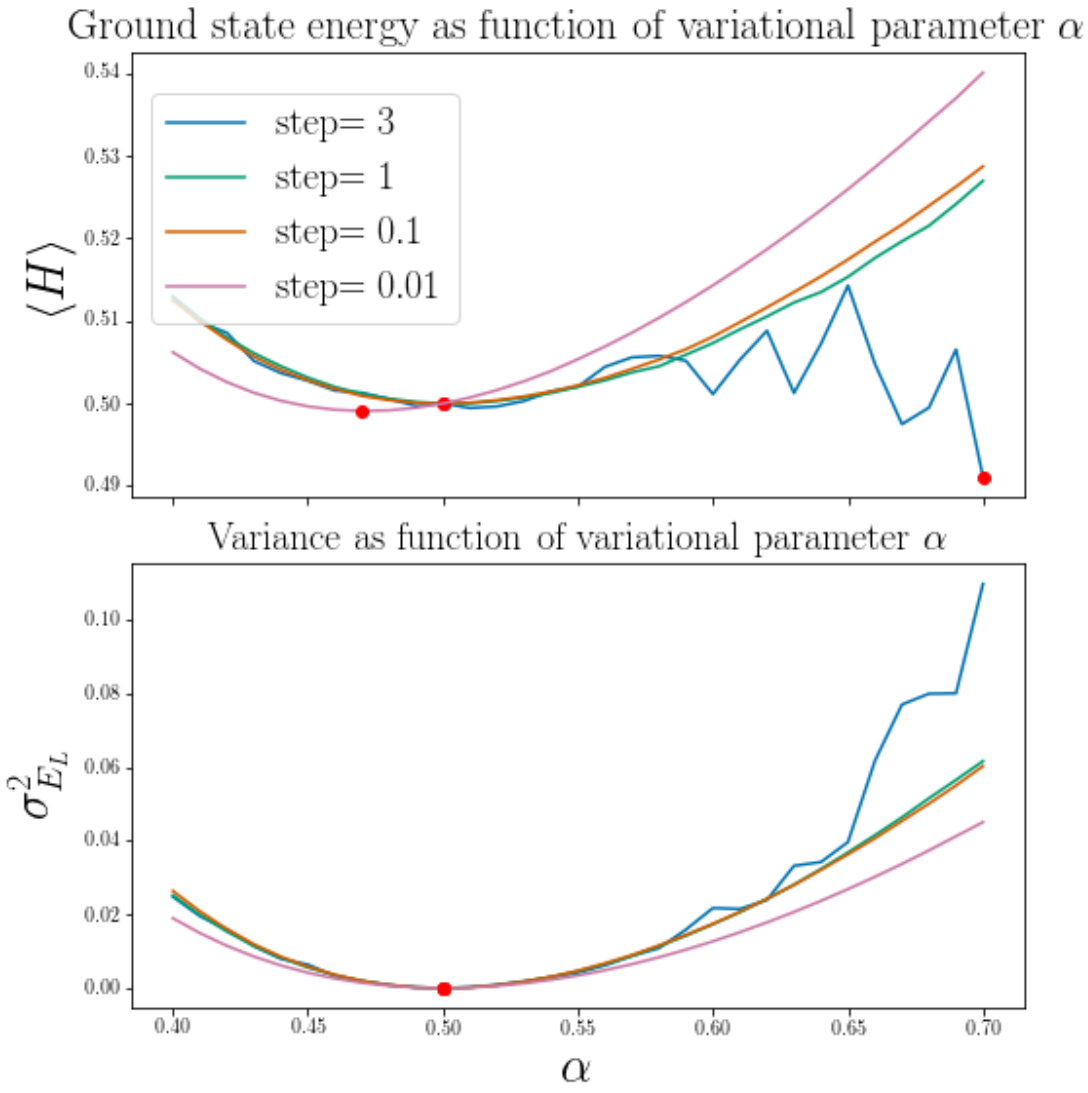


Figure 4: The same plot as shown in Figure 3, shown for multiple different values for Δt. We see that both too large, and to small values yield less than ideal results.

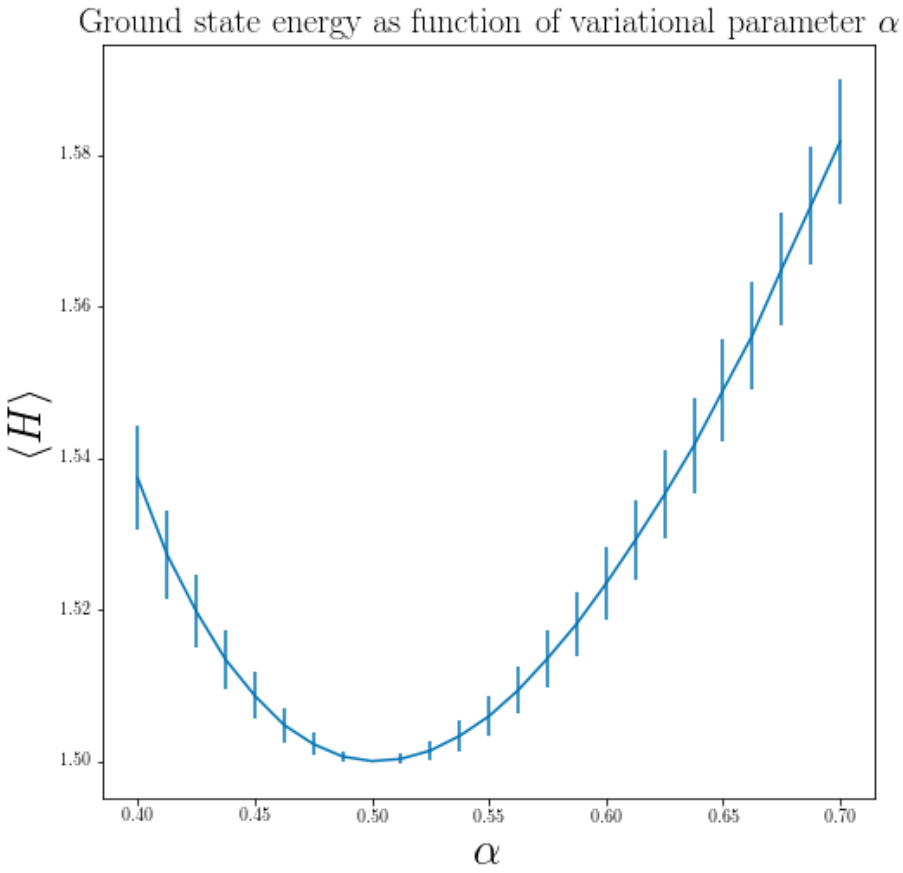


Figure 5: Simulation results using Algorithm 3, running on a system of one boson in 3D, with $\Delta t = 0.1$. Error bars have been included, whose magnitude are the standard error of the mean energy, as obtained by the blocking method. In this simulation, $n = 2^{14}$ Monte Carlo samples were generated.

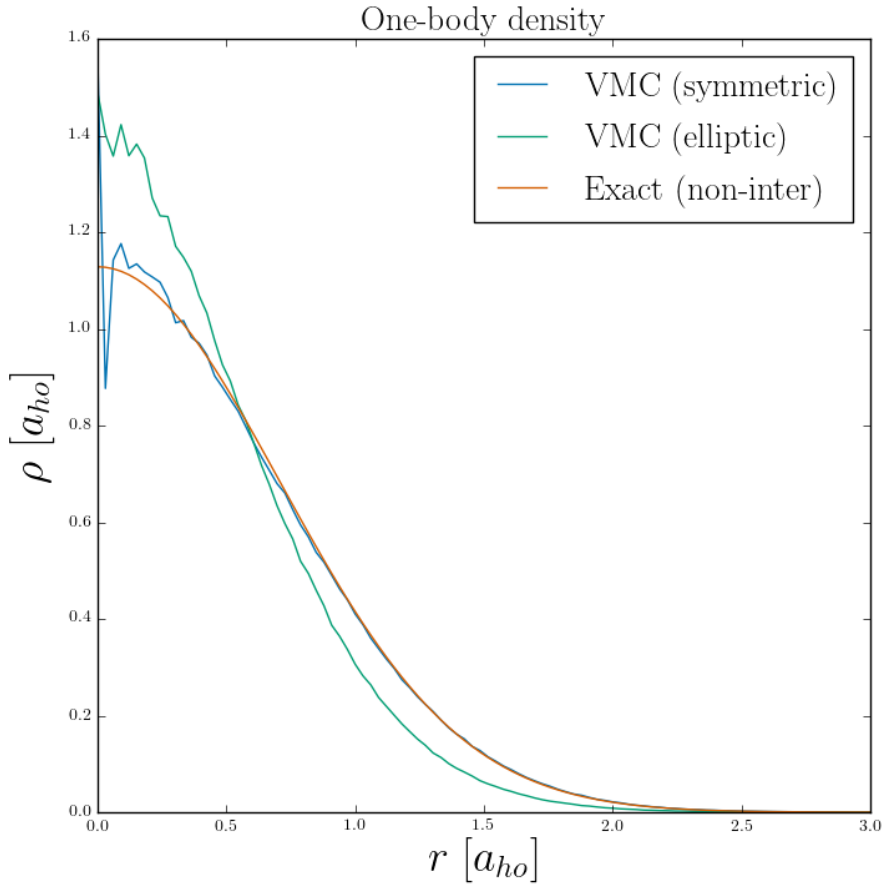


Figure 6: One-Body density for a particle in a non-interacting system of 10 bosons in the 3D symmetric and elliptical harmonic oscillator traps. The results are with $\alpha = 1/2$, using Algorithm 2 with a step length of 1 and 10^6 samples. For the elliptical trap, $\beta = \omega_z = 2.82843$ was used. For reference is the exact Gaussian form expected (for the symmetric curve). All graphs have been normalized so that they integrate to 1. The jagged shape around $r = 0$ is an artifact of the approximation used, where the volume of the regions become very small, and their counts become less stable as a result. The r-axis is given in units of a_{ho}, and ρ is similarly dimensionless.

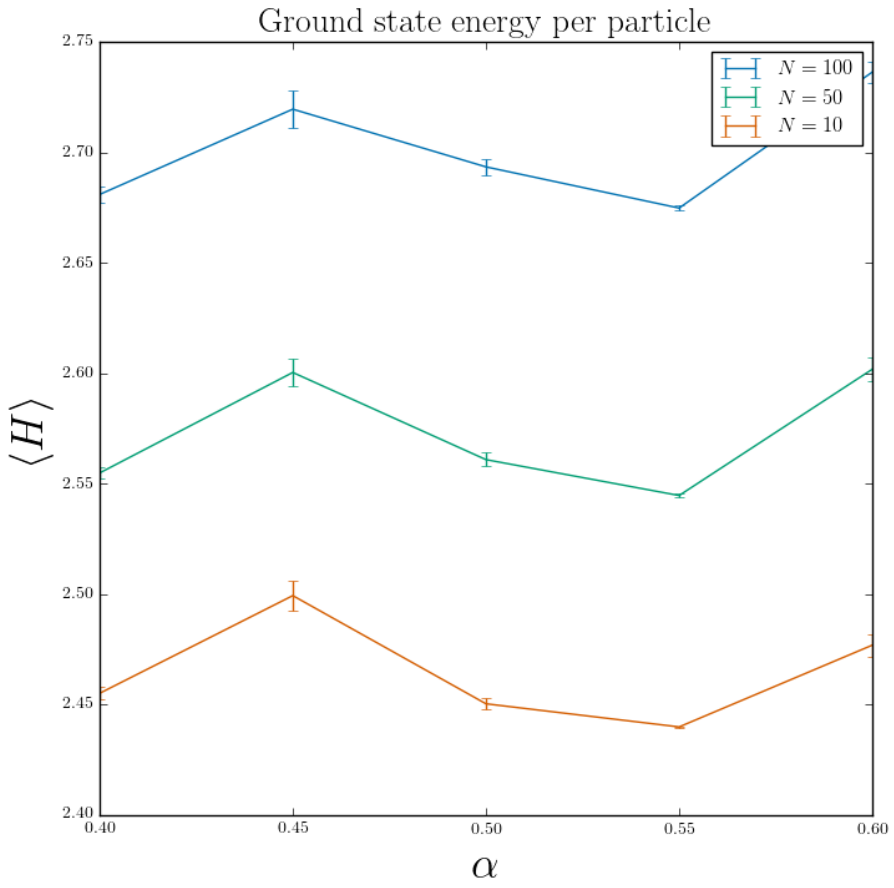


Figure 7: Energy per particle, as a function of α. We observe, as expected, a higher energy per particle when we have more bosons in the system. The plot has to low resolution to make any certain predictions about where the optimal value might be. This would have been improved given sufficient time.

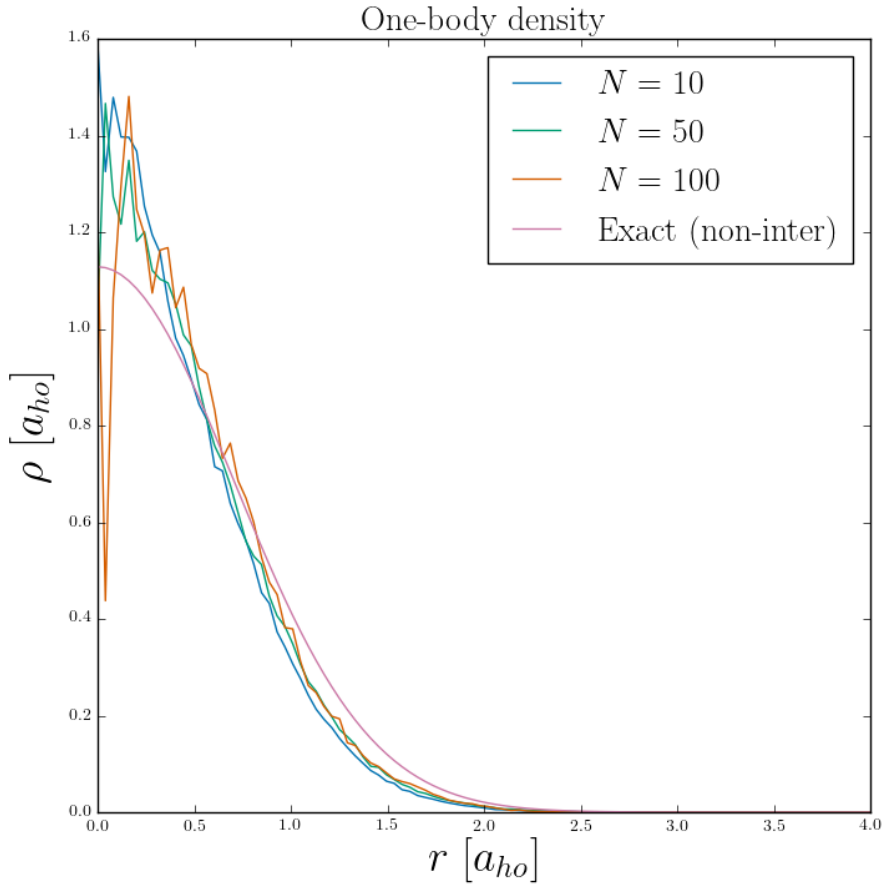


Figure 8: One-body densities for the interacting case, drawn for all systems. The *alpha* used here is $\alpha = 0.55$, same as in the apparent minimum seen in Figure 7. We see a trend very similar to that of Figure 6, where the elliptical potential forces the particles closer than in the symmetric case. There is no visual effect of adding the interaction in the distributions shown. Again, the discrepancies around $r = 0$ are artifacts of the approximation, and should not be considered in depth.

Table 1: Selected results using the standard Metropolis sampling algorithm. All runs have been made with $\alpha = 1/2$, $\beta = 1$, with a symmetric trap with strength $\omega_{ho} = 1$ and 1000 MC cycles.

Analytic	Dimensions	Particles	Step length	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	1	1	1.000	0.500000	0.000000	0.866	5.960e-04
OFF	1	1	0.100	0.500000	0.000000	0.995	5.400e-04
ON	1	1	1.000	0.500000	0.000000	0.866	3.890e-04
ON	1	1	0.100	0.500000	0.000000	0.995	2.920e-04
OFF	1	10	1.000	5.000000	0.000000	0.876	9.675e-03
OFF	1	10	0.100	4.999999	0.000000	0.997	9.434e-03
ON	1	10	1.000	5.000000	0.000000	0.876	1.353e-03
ON	1	10	0.100	5.000000	0.000000	0.997	1.352e-03
OFF	1	100	1.000	49.999990	0.000000	0.000	5.754e-01
OFF	1	100	0.100	49.999988	0.000000	0.999	5.495e-01
ON	1	100	1.000	50.000000	0.000000	0.000	9.845e-03
ON	1	100	0.100	50.000000	0.000000	0.999	9.606e-03
OFF	1	500	1.000	249.999947	0.000000	0.000	1.350e+01
OFF	1	500	0.100	249.999938	0.000000	0.999	1.336e+01
ON	1	500	1.000	250.000000	0.000000	0.000	4.214e-02
ON	1	500	0.100	250.000000	0.000000	0.999	4.125e-02
OFF	2	1	1.000	1.000000	0.000000	0.799	5.180e-04
OFF	2	1	0.100	1.000000	0.000000	0.984	5.190e-04
ON	2	1	1.000	1.000000	0.000000	0.799	2.420e-04
ON	2	1	0.100	1.000000	0.000000	0.984	2.470e-04
OFF	2	10	1.000	9.999999	0.000000	0.801	1.436e-02
OFF	2	10	0.100	9.999998	0.000000	0.994	1.360e-02
ON	2	10	1.000	10.000000	0.000000	0.801	1.091e-03
ON	2	10	0.100	10.000000	0.000000	0.994	1.147e-03
OFF	2	100	1.000	99.999979	0.000000	0.000	1.123e+00
OFF	2	100	0.100	99.999975	0.000000	0.994	1.110e+00
ON	2	100	1.000	100.000000	0.000000	0.000	8.919e-03
ON	2	100	0.100	100.000000	0.000000	0.994	8.794e-03
OFF	2	500	1.000	499.999897	0.000000	0.000	2.758e+01
OFF	2	500	0.100	499.999875	0.000000	0.993	2.730e+01
ON	2	500	1.000	500.000000	0.000000	0.000	4.287e-02
ON	2	500	0.100	500.000000	0.000000	0.993	4.292e-02
OFF	3	1	1.000	1.500000	0.000000	0.715	5.070e-04
OFF	3	1	0.100	1.500000	0.000000	0.983	5.090e-04
ON	3	1	1.000	1.500000	0.000000	0.715	2.480e-04
ON	3	1	0.100	1.500000	0.000000	0.983	2.460e-04
OFF	3	10	1.000	14.999999	0.000000	0.727	5.873e-03
OFF	3	10	0.100	14.999997	0.000000	0.994	5.730e-03
ON	3	10	1.000	15.000000	0.000000	0.727	9.090e-04

Continued on next page

Table 1 – continued from previous page

Analytic	Dimensions	Particles	Step length	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
ON	3	10	0.100	15.000000	0.000000	0.994	9.500e-04
OFF	3	100	1.000	149.999969	0.000000	0.000	3.194e-01
OFF	3	100	0.100	149.999963	0.000000	0.996	3.189e-01
ON	3	100	1.000	150.000000	0.000000	0.000	6.856e-03
ON	3	100	0.100	150.000000	0.000000	0.996	7.036e-03
OFF	3	500	1.000	749.999845	0.000000	0.000	7.310e+00
OFF	3	500	0.100	749.999813	0.000000	0.994	7.556e+00
ON	3	500	1.000	750.000000	0.000000	0.000	3.251e-02
ON	3	500	0.100	750.000000	0.000000	0.994	3.339e-02

Table 2: Selected results using improved Metropolis-Hastings sampling algorithm. All runs have been made with $\alpha = 1/2$, $\beta = 1$, with a symmetric trap with strength $\omega_{ho} = 1$ and 1000 MC cycles.

Analytic	Dimensions	Particles	Δt	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	1	1	0.100	0.500000	0.000000	0.990	4.930e-04
OFF	1	1	0.010	0.500000	0.000000	1.000	4.860e-04
OFF	1	1	0.001	0.500000	0.000000	1.000	4.870e-04
ON	1	1	0.100	0.500000	0.000000	0.990	3.450e-04
ON	1	1	0.010	0.500000	0.000000	1.000	3.490e-04
ON	1	1	0.001	0.500000	0.000000	1.000	3.610e-04
OFF	1	10	0.100	5.000000	0.000000	0.988	8.216e-03
OFF	1	10	0.010	4.999999	0.000000	1.000	7.941e-03
OFF	1	10	0.001	4.999999	0.000000	1.000	7.956e-03
ON	1	10	0.100	5.000000	0.000000	0.988	1.208e-03
ON	1	10	0.010	5.000000	0.000000	1.000	1.209e-03
ON	1	10	0.001	5.000000	0.000000	1.000	1.267e-03
OFF	1	100	0.100	49.999990	0.000000	0.000	5.483e-01
OFF	1	100	0.010	49.999989	0.000000	1.000	5.825e-01
OFF	1	100	0.001	49.999988	0.000000	1.000	5.948e-01
ON	1	100	0.100	50.000000	0.000000	0.000	9.673e-03
ON	1	100	0.010	50.000000	0.000000	1.000	9.363e-03
ON	1	100	0.001	50.000000	0.000000	1.000	9.317e-03
OFF	1	500	0.100	249.999948	0.000000	0.000	1.365e+01
OFF	1	500	0.010	249.999940	0.000000	0.931	1.342e+01
OFF	1	500	0.001	249.999938	0.000000	0.999	1.349e+01
ON	1	500	0.100	250.000000	0.000000	0.000	4.214e-02
ON	1	500	0.010	250.000000	0.000000	0.931	4.187e-02
ON	1	500	0.001	250.000000	0.000000	0.999	4.379e-02

Continued on next page

Table 2 – continued from previous page

Analytic	Dimensions	Particles	Δt	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	2	1	0.100	1.000000	0.000000	0.992	6.690e-04
OFF	2	1	0.010	1.000000	0.000000	1.000	6.750e-04
OFF	2	1	0.001	1.000000	0.000000	1.000	7.910e-04
ON	2	1	0.100	1.000000	0.000000	0.992	4.430e-04
ON	2	1	0.010	1.000000	0.000000	1.000	4.740e-04
ON	2	1	0.001	1.000000	0.000000	1.000	4.230e-04
OFF	2	10	0.100	9.999999	0.000000	0.983	1.492e-02
OFF	2	10	0.010	9.999999	0.000000	0.999	1.565e-02
OFF	2	10	0.001	9.999998	0.000000	1.000	1.498e-02
ON	2	10	0.100	10.000000	0.000000	0.983	1.307e-03
ON	2	10	0.010	10.000000	0.000000	0.999	1.305e-03
ON	2	10	0.001	10.000000	0.000000	1.000	1.320e-03
OFF	2	100	0.100	99.999980	0.000000	0.000	1.154e+00
OFF	2	100	0.010	99.999978	0.000000	0.992	1.150e+00
OFF	2	100	0.001	99.999975	0.000000	0.998	1.184e+00
ON	2	100	0.100	100.000000	0.000000	0.000	9.822e-03
ON	2	100	0.010	100.000000	0.000000	0.992	1.023e-02
ON	2	100	0.001	100.000000	0.000000	0.998	9.653e-03
OFF	2	500	0.100	499.999899	0.000000	0.000	2.817e+01
OFF	2	500	0.010	499.999876	0.000000	0.000	2.784e+01
OFF	2	500	0.001	499.999876	0.000000	1.000	2.808e+01
ON	2	500	0.100	500.000000	0.000000	0.000	4.673e-02
ON	2	500	0.010	500.000000	0.000000	0.000	4.731e-02
ON	2	500	0.001	500.000000	0.000000	1.000	4.587e-02
OFF	3	1	0.100	1.500000	0.000000	0.991	6.880e-04
OFF	3	1	0.010	1.500000	0.000000	0.999	6.960e-04
OFF	3	1	0.001	1.500000	0.000000	1.000	7.340e-04
ON	3	1	0.100	1.500000	0.000000	0.991	4.580e-04
ON	3	1	0.010	1.500000	0.000000	0.999	4.600e-04
ON	3	1	0.001	1.500000	0.000000	1.000	4.610e-04
OFF	3	10	0.100	14.999999	0.000000	0.974	6.382e-03
OFF	3	10	0.010	14.999998	0.000000	0.999	6.431e-03
OFF	3	10	0.001	14.999997	0.000000	1.000	7.076e-03
ON	3	10	0.100	15.000000	0.000000	0.974	1.260e-03
ON	3	10	0.010	15.000000	0.000000	0.999	1.635e-03
ON	3	10	0.001	15.000000	0.000000	1.000	1.619e-03
OFF	3	100	0.100	149.999969	0.000000	0.000	3.291e-01
OFF	3	100	0.010	149.999967	0.000000	0.983	3.232e-01
OFF	3	100	0.001	149.999963	0.000000	1.000	3.423e-01
ON	3	100	0.100	150.000000	0.000000	0.000	6.995e-03
ON	3	100	0.010	150.000000	0.000000	0.983	7.066e-03
ON	3	100	0.001	150.000000	0.000000	1.000	7.016e-03
OFF	3	500	0.100	749.999848	0.000000	0.000	7.526e+00

Continued on next page

Table 2 – continued from previous page

Analytic	Dimensions	Particles	Δt	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	3	500	0.010	749.999812	0.000000	0.000	7.549e+00
OFF	3	500	0.001	749.999813	0.000000	1.000	7.438e+00
ON	3	500	0.100	750.000000	0.000000	0.000	3.292e-02
ON	3	500	0.010	750.000000	0.000000	0.000	3.302e-02
ON	3	500	0.001	750.000000	0.000000	1.000	3.272e-02

Table 3: Results for the interacting system. The energies have been calculated for selected α's, and the standard error is as given by the Blocking method. For reference is the exact result for the non-interacting system.

	$N = 10$	
α	VMC	GP
0.40	24.552 ± 0.002865	24.746
0.45	24.994 ± 0.006891	24.276
0.50	24.504 ± 0.002704	24.142
0.55	24.400 ± 0.000232	24.252
0.60	24.770 ± 0.005075	24.545

	$N = 50$	
α	VMC	GP
0.40	127.744 ± 0.002530	123.729
0.45	130.019 ± 0.006282	121.381
0.50	128.046 ± 0.003077	120.711
0.55	127.234 ± 0.000727	121.259
0.60	130.091 ± 0.005271	122.723

	$N = 100$	
α	VMC	GP
0.40	268.091 ± 0.003828	247.457
0.45	271.952 ± 0.008271	242.763
0.50	269.340 ± 0.003764	241.422
0.55	267.476 ± 0.001183	242.519
0.60	273.630 ± 0.004904	245.445