University of Oslo
FYS4411
Computational physics II: Quantum mechanical systems

Project 1

Bendik Samseth

March 18, 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

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

The Hamiltonian of the system will be

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

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:

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

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

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

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:

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

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

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

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.

$$E[H] = \langle H \rangle = \frac{\int \mathrm{d}\boldsymbol{R}\, \Psi_T^* H \Psi_T}{\int \Psi_T^* \Psi_T}. \tag{7}$$

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

$$P(\boldsymbol{R}, \boldsymbol{\alpha}) = \frac{|\Psi_T|^2}{\int \mathrm{d}\boldsymbol{R}\, |\Psi_T|^2}. \tag{8}$$

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**:

$$E_L(\boldsymbol{R}, \boldsymbol{\alpha}) = \frac{1}{\Psi_T} H \Psi_T \tag{9}$$

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

$$\begin{aligned} \langle H \rangle &= \int \mathrm{d}\boldsymbol{R}\, P(\boldsymbol{R}, \boldsymbol{\alpha}) E_L(\boldsymbol{R}, \boldsymbol{\alpha}) \\ &\approx \frac{1}{N} \sum_{i=1}^{N} E_L(\boldsymbol{R}_i, \boldsymbol{\alpha}), \end{aligned} \tag{10}$$

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:

$$E_0 = \sum_{i=1}^{N} \sum_{d=1}^{D=\{1,2,3\}} \frac{\hbar \omega_{ho}}{2} = \frac{N \times D}{2}, \tag{11}$$

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:

$$\begin{aligned} \sigma_E^2 &= \langle H^2 \rangle - \langle H \rangle^2 \\ &= \langle \Psi | H^2 | \Psi \rangle - \langle \Psi | H | \Psi \rangle^2 \\ &= \langle \Psi | E^2 | \Psi \rangle - \langle \Psi | E | \Psi \rangle^2 \\ &= E^2 \langle \Psi | \Psi \rangle - (E \langle \Psi | \Psi \rangle)^2 = 0. \end{aligned} \tag{12}$$

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

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: $y = E_L$
$y = -2 \times N \times D \times \Psi_T(\boldsymbol{R})$
for $i = 1$ to $N$ do
    for $d = 1$ to $D$ do
        $\boldsymbol{R}_+ \leftarrow \boldsymbol{R} + h\boldsymbol{e}_{i,d}$
        $\boldsymbol{R}_- \leftarrow \boldsymbol{R} - h\boldsymbol{e}_{i,d}$
        $y \leftarrow y + \Psi_T(\boldsymbol{R}_+) + \Psi_T(\boldsymbol{R}_-)$
    end for
end for
$y \leftarrow -\,y/2h^2$
$y \leftarrow y/\Psi_T(\boldsymbol{R}) + \sum_{i=1}^N V_{ext} + \sum_{i<j}^N V_{int}$
```

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

$$\frac{1}{\Psi_L}\sum_k^N \nabla_k^2 \Psi_L. \tag{13}$$

To get going, we rewrite the wave function as

$$\Psi_L(\boldsymbol{R}) = \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right), \tag{14}$$

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

$$\begin{aligned}
\boldsymbol{\nabla}_k \Psi_T(\boldsymbol{r}) &= \boldsymbol{\nabla}_k \prod_i \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right) \\
&= \prod_{i\neq k} \phi(\boldsymbol{r}_i)\exp\left(\sum_{i<j} u(r_{ij})\right)\boldsymbol{\nabla}_k \phi(\boldsymbol{r}_k) \\
&+ \prod_i \phi(\boldsymbol{r}_i)\boldsymbol{\nabla}_k \exp\left(\sum_{i<j} u(r_{ij})\right) \\
&= \Psi_T\left[\frac{\boldsymbol{\nabla}_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k} \boldsymbol{\nabla}_k u(r_{kj})\right].
\end{aligned} \tag{15}$$

The first term is evaluated quite simply:

$$\begin{aligned}
\frac{\boldsymbol{\nabla}_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} &= \frac{\boldsymbol{\nabla}_k}{\phi(\boldsymbol{r})}\exp\left[-\alpha\left(x_k^2 + y_k^2 + \beta z_k^2\right)\right] \\
&= \frac{\boldsymbol{\nabla}_k}{\phi(\boldsymbol{r})}\exp\left[-\alpha \hat{\boldsymbol{r}}_k^2\right] \\
&= -2\alpha\hat{\boldsymbol{r}}_k,
\end{aligned} \tag{16}$$

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:

$$\begin{aligned}
\boldsymbol{\nabla}_k u(r_{kj}) &= u'(r_{kj})\boldsymbol{\nabla}_k\sqrt{\|\boldsymbol{r}_k - \boldsymbol{r}_j\|^2} \\
&= \frac{u'(r_{kj})}{2r_{kj}}\boldsymbol{\nabla}_k\left(\|\boldsymbol{r}_k\|^2 - 2\boldsymbol{r}_k \cdot \boldsymbol{r}_j + \|r_j\|^2\right) \\
&= u'(r_{kj})\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\
&= \frac{\partial}{\partial r_{kj}}\left[\ln\left(1 - \frac{a}{r_{kj}}\right)\right]\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\
&= \frac{\boldsymbol{r}_{kj}}{r_{kj}}\frac{a}{r_{kj}(r_{kj} - a)}.
\end{aligned} \tag{17}$$

Now we can find the Laplacian by taking the divergence of (15):

$$\begin{aligned}
\frac{1}{\Psi_L}\nabla_k^2\Psi_L &= \frac{1}{\Psi_L}\boldsymbol{\nabla}_k \cdot \Psi_T \left[\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\boldsymbol{\nabla}u(r_{kj})\right] \\
&= \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\nabla_k^2 u(r_{kj}) \\
&+ \frac{\boldsymbol{\nabla}_k(\phi(\boldsymbol{r}_k)) \cdot \left(\sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)}{\phi(\boldsymbol{r}_k)} \\
&+ \left[\left(\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + \sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)\right. \\
&\left.\cdot\left(\sum_{j\neq k}\boldsymbol{\nabla}_k u(r_{kj})\right)\right] \\
&= \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2\frac{\boldsymbol{\nabla}_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} \cdot \sum_{j\neq k}\left(\frac{\boldsymbol{r}_{kj}}{r_{kj}}u'(r_{kj})\right) \\
&+ \sum_{i,j\neq k}\frac{\boldsymbol{r}_{ki}\cdot\boldsymbol{r}_{kj}}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) + \sum_{j\neq k}\nabla_k^2 u(r_{kj}).
\end{aligned} \tag{18}$$

There are two new quantities here which need to be evaluated before we are done:

$$\frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} = 2\alpha\Big[2\alpha\|\hat{\boldsymbol{r}}_k\|^2 - d(\beta)\Big],$$

$$\text{with } d(\beta) = \begin{cases} 1 & \text{for} \quad 1\text{D} \\ 2 & \text{for} \quad 2\text{D} \\ 2+\beta & \text{for} \quad 3\text{D} \end{cases}, \tag{19}$$

and

$$\begin{aligned}
\nabla_k^2 u(r_{kj}) &= \boldsymbol{\nabla}_k \cdot u'(r_{kj})\frac{\boldsymbol{r}_{kj}}{r_{kj}} \\
&= u'(r_{kj})\frac{2}{r_{kj}} + \frac{\boldsymbol{r}_{kj}}{r_{kj}} \cdot \boldsymbol{\nabla}_k u'(r_{kj}) \\
&= u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj}),
\end{aligned} \tag{20}$$

where

$$\begin{aligned}
u''(r_{ij}) &= \frac{\partial^2}{\partial r_{ij}^2}\ln\left(1 - \frac{a}{r_{ij}}\right) \\
&= \frac{a(a-2r_{ij})}{r_{ij}^2(r_{ij}-a)^2}.
\end{aligned} \tag{21}$$

Inserting all of this back into (18) we get:

$$\begin{aligned}
\frac{1}{\Psi_L}\nabla_k^2\Psi_L &= 2\alpha\Big[2\alpha\|\hat{\boldsymbol{r}}_k\|^2 - d(\beta)\Big] \\
&- 4\alpha\hat{\boldsymbol{r}}_k \cdot \left[\sum_{j\neq k}\frac{\boldsymbol{r}_{kj}}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)}\right] \\
&+ \sum_{i,j\neq k}\frac{\boldsymbol{r}_{ki}\cdot\boldsymbol{r}_{kj}}{r_{ki}r_{kj}}\frac{a}{r_{ki}(r_{ki}-a)}\frac{a}{r_{kj}(r_{kj}-a)} \\
&+ \sum_{j\neq k}\left(\frac{a(a-2r_{kj})}{r_{kj}^2(r_{kj}-a)^2} + \frac{2}{r_{kj}}\frac{a}{r_{kj}(r_{kj}-a)}\right).
\end{aligned} \tag{22}$$

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:

$$\begin{aligned}
E_L &= \frac{1}{\Psi_T}H\Psi_T \\
&= \sum_i V_{ext}(\boldsymbol{r}_i) + \sum_{i<j}V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) - \frac{1}{2}\sum_k \frac{1}{\Psi_T}\nabla_k^2\Psi_T
\end{aligned} \tag{23}$$

where we substitute in (22) 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:

$$\boldsymbol{F}_k = \frac{2\boldsymbol{\nabla}_k \Psi_T}{\Psi_T} \tag{24}$$

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,

$$\boldsymbol{F}_k = 2\left[\sum_{j\neq k} \frac{\boldsymbol{r}_{kj}}{r_{kj}} \frac{a}{r_{kj}(r_{kj}-a)} - 2\alpha\hat{\boldsymbol{r}}_k\right]. \tag{25}$$

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.

3 Algorithms

In (10) 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 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-Hastings 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 Simple Metropolis-Hastings

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

Algorithm 2 The Metropolis-Hastings 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
$\boldsymbol{R}$ ← randomly initialized matrix of positions
for $M$ iterations do
    for every particle $i \in [1, N]$ do
        $\Delta\boldsymbol{r}$ ← random perturbation vector
        $\boldsymbol{R}^* \leftarrow \boldsymbol{R}$
        $\boldsymbol{R}^*_i \leftarrow \boldsymbol{R}^*_i + \Delta\boldsymbol{r}$
        $q \leftarrow |\Psi_T(\boldsymbol{R}^*)|^2 \big/ |\Psi_T(\boldsymbol{R})|^2$
        $r \overset{\text{d}}{\leftarrow} \text{Unif}(0,1)$
        if $r \leq q$ then
            $\boldsymbol{R} \leftarrow \boldsymbol{R}^*$
        end if
        Append $\boldsymbol{R}$ 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.

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.

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 **??**.

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 with 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 (24), 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,

$$\frac{\partial \boldsymbol{r}_k}{\partial t} = D\boldsymbol{F}_k(\boldsymbol{r}_k) + \boldsymbol{\eta}, \tag{26}$$

which describes how the position of a particle changes with time. 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:

$$\boldsymbol{r}^* = \boldsymbol{r} + \frac{1}{2}\boldsymbol{F}_k(\boldsymbol{r}_k)\Delta t + \boldsymbol{\xi}\sqrt{\Delta t}, \tag{27}$$

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

$$q(\boldsymbol{R}^*, \boldsymbol{R}) = \frac{|\Psi_T(\boldsymbol{R}^*)|^2}{|\Psi_T(\boldsymbol{R})|^2}. \tag{28}$$

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:

$$q(\boldsymbol{R}^*, \boldsymbol{R}) = \frac{T_{j\to i}}{T_{i\to j}}\frac{|\Psi_T(\boldsymbol{R}^*)|^2}{|\Psi_T(\boldsymbol{R})|^2}, \tag{29}$$

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 (28). 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

$$\frac{\partial \Psi_T}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} - F\right)\Psi_T, \tag{30}$$

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 (24).

Equation (30) yields a solution given by the following Green's function (for one particle):

$$G(\boldsymbol{r}_k^*, \boldsymbol{r}_k, \Delta t) \propto \exp\left[-\frac{\left\|\boldsymbol{r}_k^* - \boldsymbol{r}_k - D\Delta t \boldsymbol{F}_k(\boldsymbol{r}_k)\right\|^2}{4D\Delta t}\right]. \tag{31}$$

This is interpreted as the probability of transitioning to position $\boldsymbol{r}_k^*$ from position $\boldsymbol{r}$ in a time interval Δt. Equation (29) needs only the ratio of these probabilities, so we may simplify this as

$$\frac{G(\boldsymbol{r}_k, \boldsymbol{r}_k^*, \Delta t)}{G(\boldsymbol{r}_k^*, \boldsymbol{r}_k, \Delta t)} = \exp\left\{\begin{array}{l} [\boldsymbol{F}_k(\boldsymbol{r}_k) + \boldsymbol{F}_k(\boldsymbol{r}_k^*)] \\ \qquad\qquad \bullet \\ [D\Delta t(\boldsymbol{F}_k(\boldsymbol{r}_k) - \boldsymbol{F}_k(\boldsymbol{r}_k^*)) \\ +2\boldsymbol{r}_k - 2\boldsymbol{r}_k^*] \end{array}\right\} \tag{32}$$

We may then finally insert this ratio into (29).

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, with importance sampling, as it pertains to our specific application.

```
Require: $M$, generates $M \times N$ samples.
Ensure: samples $\overset{\mathrm{d}}{\leftarrow} P(\boldsymbol{R}, \boldsymbol{\alpha})$
samples ← empty list
$\boldsymbol{R} \leftarrow$ randomly initialized matrix of positions
for $M$ iterations do
    for every particle $i \in [1, N]$ do
        $\Delta\boldsymbol{r}_i^* \leftarrow \frac{1}{2}\boldsymbol{F}_i(\boldsymbol{r}_i)\Delta t + \boldsymbol{\xi}\sqrt{\Delta t}$
        $\boldsymbol{R}^* \leftarrow \boldsymbol{R}$
        $\boldsymbol{R}_i^* \leftarrow \boldsymbol{R}_i^* + \Delta\boldsymbol{r}_i^*$
        $q \leftarrow |\Psi_T(\boldsymbol{R}^*)|^2 \big/ |\Psi_T(\boldsymbol{R})|^2$
        $q \leftarrow q \times G(\boldsymbol{r}_k, \boldsymbol{r}_k^*, \Delta t)/G(\boldsymbol{r}_k^*, \boldsymbol{r}_k, \Delta t)$
        $r \overset{\mathrm{d}}{\leftarrow} \mathrm{Unif}(0, 1)$
        if $r \leq q$ then
            $\boldsymbol{R} \leftarrow \boldsymbol{R}^*$
        end if
        Append $\boldsymbol{R}$ to samples
    end for
end for
```

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

$$\sigma_{\bar{E}_L} = \sqrt{\frac{\sigma^2_{E_L}}{n}} = \sqrt{\frac{1}{n}\left[\langle E_L^2\rangle - \langle E_L\rangle^2\right]}, \qquad (33)$$

where n denotes the number of samples[1], 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:

$$\hat{\sigma}_{\bar{E}_L} = \sqrt{\frac{\hat{\sigma}^2_{E_L}}{n}} = \sqrt{\frac{1}{n}\left[\widehat{\langle E_L^2\rangle} - \bar{E}_L^2\right]}. \qquad (34)$$

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 (34) 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 (34) less biased by replacing the sample variance by the sample *covariance*:

$$\begin{aligned}\hat{\sigma}_{\bar{E}_L} &= \sqrt{\frac{\mathrm{Cov}(E_L)}{n}} \\ &= \sqrt{\frac{1}{n^2}\sum_{i,j}\big(E_{L,i} - \bar{E}_L\big)\big(E_{L,j} - \bar{E}_L\big)},\end{aligned} \qquad (35)$$

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

$$\hat{\sigma}^2_{\bar{E}_L} = \frac{\hat{\sigma}^2_{E_L}}{n} + \frac{2}{n^2}\sum_{i<j}\big(E_{L,i} - \bar{E}_L\big)\big(E_{L,j} - \bar{E}_L\big). \qquad (36)$$

We observe that if the energies are uncorrelated in time, we would just get back (34).

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

$$f_d = \frac{1}{n-d}\sum_{k=1}^{n-d}\big(E_{L,k} - \bar{E}_L\big)\big(E_{L,k+d} - \bar{E}_L\big) \qquad (37)$$

$$\kappa_d = \frac{f_d}{\hat{\sigma}^2_{E_L}}. \qquad (38)$$

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

Now (36) can be written as

$$\begin{aligned}\hat{\sigma}^2_{\bar{E}_L} &= \frac{\hat{\sigma}^2_{E_L}}{n} + \frac{2\hat{\sigma}^2_{E_L}}{n}\sum_{d=1}^{n-1}\kappa_d \\ &= \frac{\tau}{n}\,\hat{\sigma}^2_{\bar{E}_L},\end{aligned} \qquad (39)$$

where we have defined the *autocorrelation time* τ,

$$\tau = 1 + 2\sum_{d=1}^{n-1}\kappa_d, \qquad (40)$$

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 (34). 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 up to billions of samples.

[1] Upper case N will be used for the number of particles in the system, lower case n for the number 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, $\{E_L\}_b \quad \text{for} \quad b = 1, 2, \dots, B$, 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 $\{E_L\}_b \quad \text{for} \quad b = 2, 3, \dots, B$.

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

On 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 be used in this project. The derivation of how this works, however, will not be discussed any further in this report.

5 Results

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 (11). 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 can see the already motioned downside of using the simple Metropolis algorithm in the reported acceptance rate. We can see that about $15\,\% - 25\,\%$ of the suggested moves are dropped, which means a significant amount of work has been wasted.

Perhaps most interesting is to look at the time spent on each configuration. If we compare the durations for $N = 100$ and $N = 500$ with analytical expressions turned on, 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 for numerical differentiation, we observe that time scales as $\mathcal{O}(N^3)$, resulting in quite substantial running 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.

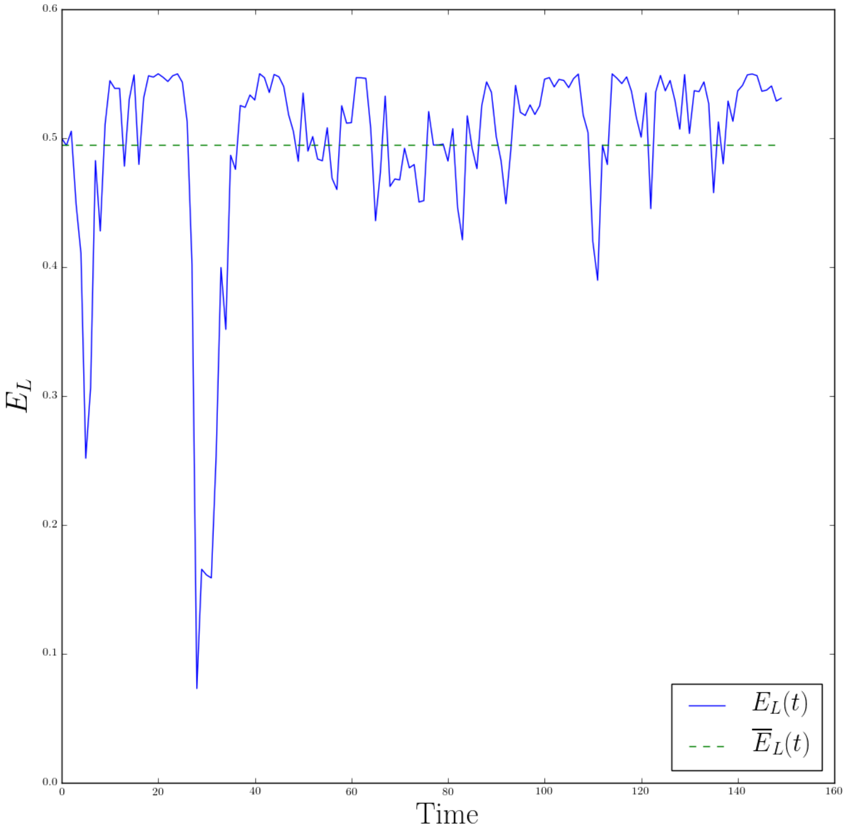

Figure 1: Example time series of local energy values obtained from a simulation using Algorithm 3. Energies shown are a small snippet of energies obtained for a system of one boson in one dimension.

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 100 MC cycles.

Analytic	Dimensions	Particles	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	1	1	4.999999e-01	2.468504e-15	0.82	6.00e-05
ON	1	1	5.000000e-01	0.000000e+00	0.82	4.80e-05
OFF	1	10	5.000000e+00	1.110006e-13	0.88	2.34e-03
ON	1	10	5.000000e+00	9.624103e-32	0.88	5.34e-04
OFF	1	100	5.000000e+01	2.618902e-12	0.87	7.66e-01
ON	1	100	5.000000e+01	4.143476e-29	0.87	3.15e-02
OFF	1	500	2.500000e+02	4.534363e-11	0.86	8.73e+01
ON	1	500	2.500000e+02	4.204242e-27	0.86	7.74e-01
OFF	2	1	9.999999e-01	9.514142e-15	0.81	5.30e-05
ON	2	1	1.000000e+00	0.000000e+00	0.81	3.50e-05
OFF	2	10	9.999999e+00	2.095685e-13	0.79	3.19e-03
ON	2	10	1.000000e+01	2.871454e-31	0.79	4.58e-04
OFF	2	100	9.999999e+01	9.343919e-12	0.80	1.77e+00
ON	2	100	1.000000e+02	2.646736e-28	0.80	3.58e-02
OFF	2	500	5.000000e+02	2.731972e-10	0.79	2.05e+02
ON	2	500	5.000000e+02	1.681820e-26	0.79	8.89e-01
OFF	3	1	1.500000e+00	1.925760e-14	0.68	6.10e-05
ON	3	1	1.500000e+00	0.000000e+00	0.68	3.60e-05
OFF	3	10	1.500000e+01	7.509506e-13	0.73	5.09e-03
ON	3	10	1.500000e+01	1.072851e-30	0.73	5.42e-04
OFF	3	100	1.500000e+02	1.686698e-11	0.74	3.02e+00
ON	3	100	1.500000e+02	5.333861e-28	0.74	4.19e-02
OFF	3	500	7.499999e+02	8.238718e-10	0.74	3.57e+02
ON	3	500	7.500000e+02	4.322122e-26	0.74	1.03e+00

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 particle, using 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 low value of 10^4 was used to give a visual difference to the result in the next section, when we include importance sampling.

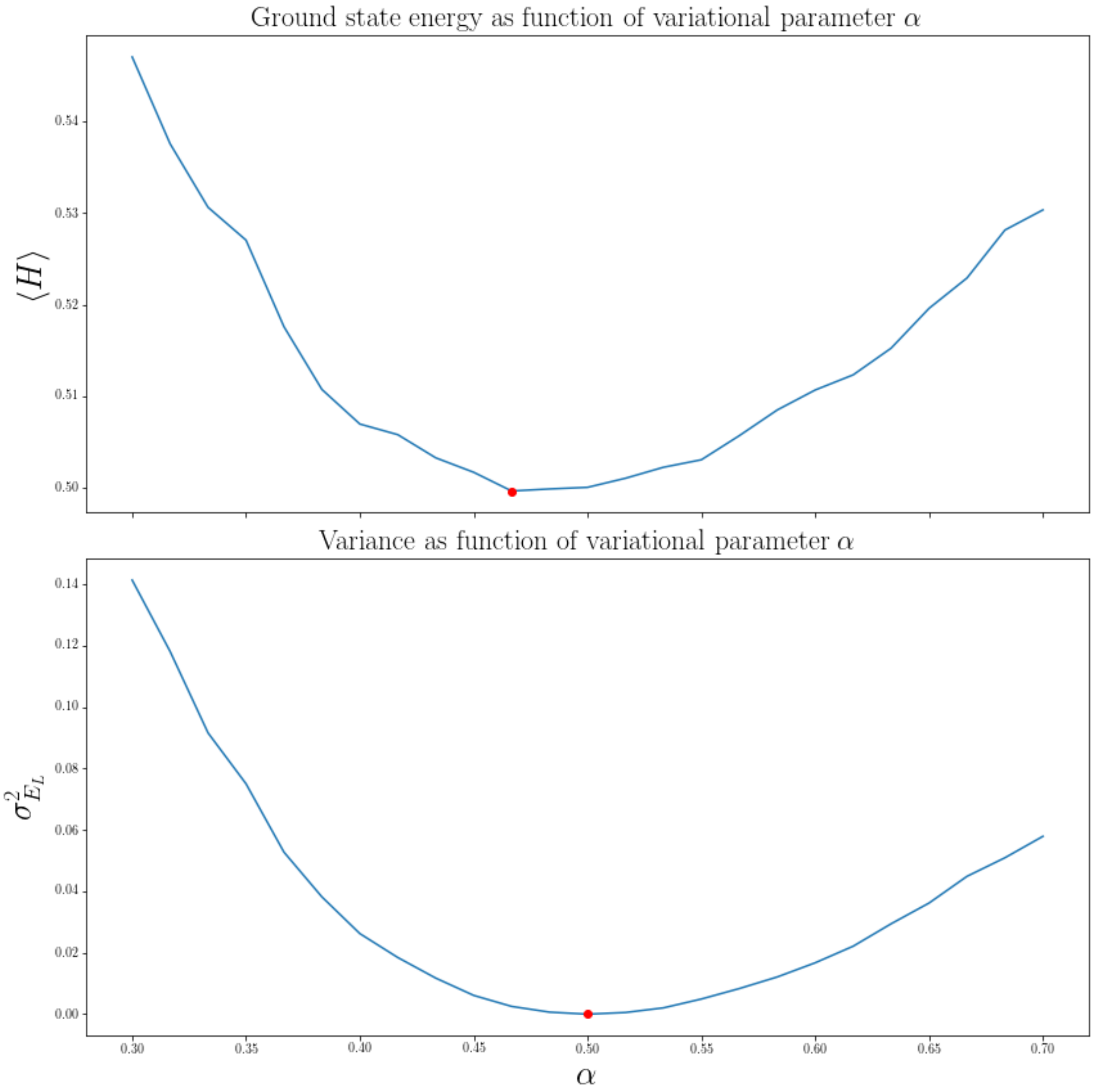

Figure 2: Plot of average local energy and accompanying population variance, as functions of the variational parameter α. The minimum of both graphs are marked, and is expected to be placed at $\alpha = 1/2$. Results shown are for $1D$, one particle and 10^4 Monte Carlo cycles.

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 100 MC cycles.

Analytic	Dimensions	Particles	Δt	$\langle H \rangle$	Var(E_L)	Acceptance Rate	Time (s)
OFF	1	1	1.00e+00	5.000000e-01	1.200124e-14	0.79	6.80e-05
OFF	1	1	1.00e-01	5.000000e-01	7.324577e-15	0.99	5.70e-05
OFF	1	1	1.00e-02	4.999999e-01	2.515150e-15	1.00	5.70e-05
ON	1	1	1.00e+00	5.000000e-01	0.000000e+00	0.79	4.60e-05
ON	1	1	1.00e-01	5.000000e-01	0.000000e+00	0.99	4.60e-05
ON	1	1	1.00e-02	5.000000e-01	0.000000e+00	1.00	4.80e-05
OFF	1	10	1.00e+00	5.000000e+00	8.567668e-14	0.79	1.87e-03
OFF	1	10	1.00e-01	5.000000e+00	1.116008e-13	0.99	1.83e-03
OFF	1	10	1.00e-02	4.999999e+00	5.788522e-14	1.00	2.03e-03
ON	1	10	1.00e+00	5.000000e+00	1.356841e-31	0.79	5.59e-04
ON	1	10	1.00e-01	5.000000e+00	8.046381e-32	0.99	5.34e-04
ON	1	10	1.00e-02	5.000000e+00	1.735494e-31	1.00	5.18e-04
OFF	1	100	1.00e+00	5.000000e+01	9.927316e-13	0.78	8.18e-01
OFF	1	100	1.00e-01	5.000000e+01	1.161532e-12	0.99	7.99e-01
OFF	1	100	1.00e-02	4.999999e+01	3.082697e-12	1.00	7.73e-01
ON	1	100	1.00e+00	5.000000e+01	3.903662e-29	0.78	3.92e-02
ON	1	100	1.00e-01	5.000000e+01	3.738065e-29	0.99	3.21e-02
ON	1	100	1.00e-02	5.000000e+01	1.461753e-28	1.00	3.11e-02
OFF	1	500	1.00e+00	2.500000e+02	1.294711e-11	0.78	8.68e+01
OFF	1	500	1.00e-01	2.500000e+02	3.165400e-11	0.99	8.60e+01
OFF	1	500	1.00e-02	2.500000e+02	1.075666e-10	1.00	8.64e+01
ON	1	500	1.00e+00	2.500000e+02	3.030825e-27	0.78	7.99e-01
ON	1	500	1.00e-01	2.500000e+02	2.970128e-27	0.99	7.55e-01
ON	1	500	1.00e-02	2.500000e+02	1.643839e-26	1.00	7.41e-01
OFF	2	1	1.00e+00	9.999999e-01	1.973416e-14	0.69	7.30e-05
OFF	2	1	1.00e-01	9.999999e-01	9.885133e-15	0.99	7.20e-05
OFF	2	1	1.00e-02	9.999998e-01	2.693806e-15	1.00	7.30e-05
ON	2	1	1.00e+00	1.000000e+00	0.000000e+00	0.69	5.30e-05
ON	2	1	1.00e-01	1.000000e+00	0.000000e+00	0.99	5.10e-05
ON	2	1	1.00e-02	1.000000e+00	0.000000e+00	1.00	1.10e-04
OFF	2	10	1.00e+00	9.999999e+00	1.870478e-13	0.67	3.62e-03
OFF	2	10	1.00e-01	1.000000e+01	3.084756e-13	0.99	3.57e-03
OFF	2	10	1.00e-02	9.999999e+00	5.122561e-13	1.00	3.56e-03
ON	2	10	1.00e+00	1.000000e+01	2.303474e-31	0.67	6.67e-04
ON	2	10	1.00e-01	1.000000e+01	3.186998e-31	0.99	6.30e-04
ON	2	10	1.00e-02	1.000000e+01	4.985601e-31	1.00	6.73e-04
OFF	2	100	1.00e+00	9.999999e+01	2.286146e-12	0.66	1.77e+00
OFF	2	100	1.00e-01	9.999999e+01	5.579622e-12	0.99	1.77e+00
OFF	2	100	1.00e-02	9.999999e+01	1.711425e-11	1.00	1.83e+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)
ON	2	100	1.00e+00	1.000000e+02	1.219566e-28	0.66	3.90e-02
ON	2	100	1.00e-01	1.000000e+02	1.699800e-28	0.99	3.55e-02
ON	2	100	1.00e-02	1.000000e+02	4.700753e-28	1.00	3.52e-02
OFF	2	500	1.00e+00	5.000000e+02	2.908057e-26	0.00	2.13e+02
OFF	2	500	1.00e-01	5.000000e+02	1.966011e-10	0.99	2.10e+02
OFF	2	500	1.00e-02	4.999999e+02	6.132290e-10	1.00	2.08e+02
ON	2	500	1.00e+00	5.000000e+02	2.908057e-26	0.00	1.10e+00
ON	2	500	1.00e-01	5.000000e+02	9.117211e-27	0.99	8.25e-01
ON	2	500	1.00e-02	5.000000e+02	4.929447e-26	1.00	8.23e-01
OFF	3	1	1.00e+00	1.500000e+00	1.637063e-14	0.39	8.70e-05
OFF	3	1	1.00e-01	1.500000e+00	3.177261e-14	0.98	8.50e-05
OFF	3	1	1.00e-02	1.500000e+00	1.128693e-14	1.00	1.14e-04
ON	3	1	1.00e+00	1.500000e+00	0.000000e+00	0.39	5.70e-05
ON	3	1	1.00e-01	1.500000e+00	0.000000e+00	0.98	5.80e-05
ON	3	1	1.00e-02	1.500000e+00	0.000000e+00	1.00	5.80e-05
OFF	3	10	1.00e+00	1.500000e+01	4.022822e-13	0.61	5.30e-03
OFF	3	10	1.00e-01	1.500000e+01	3.770274e-13	0.99	5.27e-03
OFF	3	10	1.00e-02	1.500000e+01	5.380093e-13	1.00	5.27e-03
ON	3	10	1.00e+00	1.500000e+01	1.025519e-30	0.61	7.37e-04
ON	3	10	1.00e-01	1.500000e+01	7.699282e-31	0.99	6.95e-04
ON	3	10	1.00e-02	1.500000e+01	1.514613e-30	1.00	7.69e-04
OFF	3	100	1.00e+00	1.500000e+02	6.783233e-12	0.59	3.03e+00
OFF	3	100	1.00e-01	1.500000e+02	1.261666e-11	0.99	3.36e+00
OFF	3	100	1.00e-02	1.500000e+02	3.451246e-11	1.00	3.16e+00
ON	3	100	1.00e+00	1.500000e+02	4.937234e-28	0.59	4.58e-02
ON	3	100	1.00e-01	1.500000e+02	4.433979e-28	0.99	4.07e-02
ON	3	100	1.00e-02	1.500000e+02	1.079131e-27	1.00	4.31e-02
OFF	3	500	1.00e+00	nan	nan	0.00	3.59e+02
OFF	3	500	1.00e-01	7.499999e+02	8.293598e-10	0.97	3.61e+02
OFF	3	500	1.00e-02	7.499999e+02	1.534635e-09	1.00	3.66e+02
ON	3	500	1.00e+00	7.500000e+02	1.163223e-25	0.00	1.33e+00
ON	3	500	1.00e-01	7.500000e+02	2.568525e-26	0.97	9.78e-01
ON	3	500	1.00e-02	7.500000e+02	9.327185e-26	1.00	9.69e-01

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.

The acceptance rates are now quite interesting. We see that for $\Delta t \leq 0.1$, we have $\sim 100\,\%$ 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. At this point the randomness in (27) 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. Setting $\Delta t = 0.1$ seems to work well in this case, and will be used unless otherwise stated.

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. Small values tend to not increase the energy when we increase alpha, and too large numbers makes the energy plot approach a straight line with minimum at the lowest alpha tried. Neither of these behaviours are desired, so some care has to be taken in ensuring that we apply a value that gives results we expect.

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 $\Delta t = 0.1$ is a sensible choice.

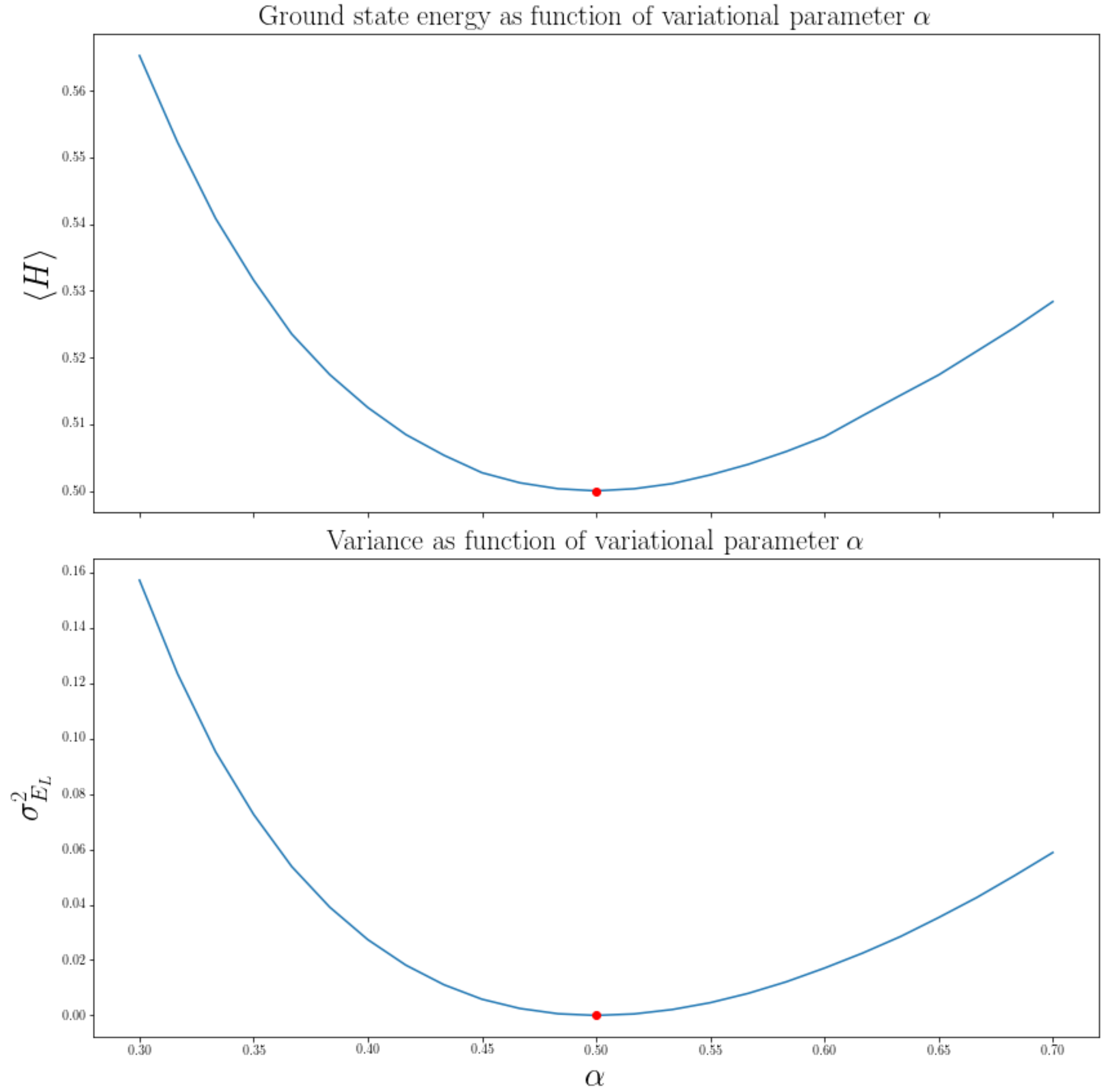

Figure 3: Plot of average local energy and accompanying population variance, as functions of the variational parameter α. The minimum of both graphs are marked, and is expected to be placed at $\alpha = 1/2$. Results shown are for $1D$, one particle and 10^4 Monte Carlo cycles, using the Metropolis-Hastings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

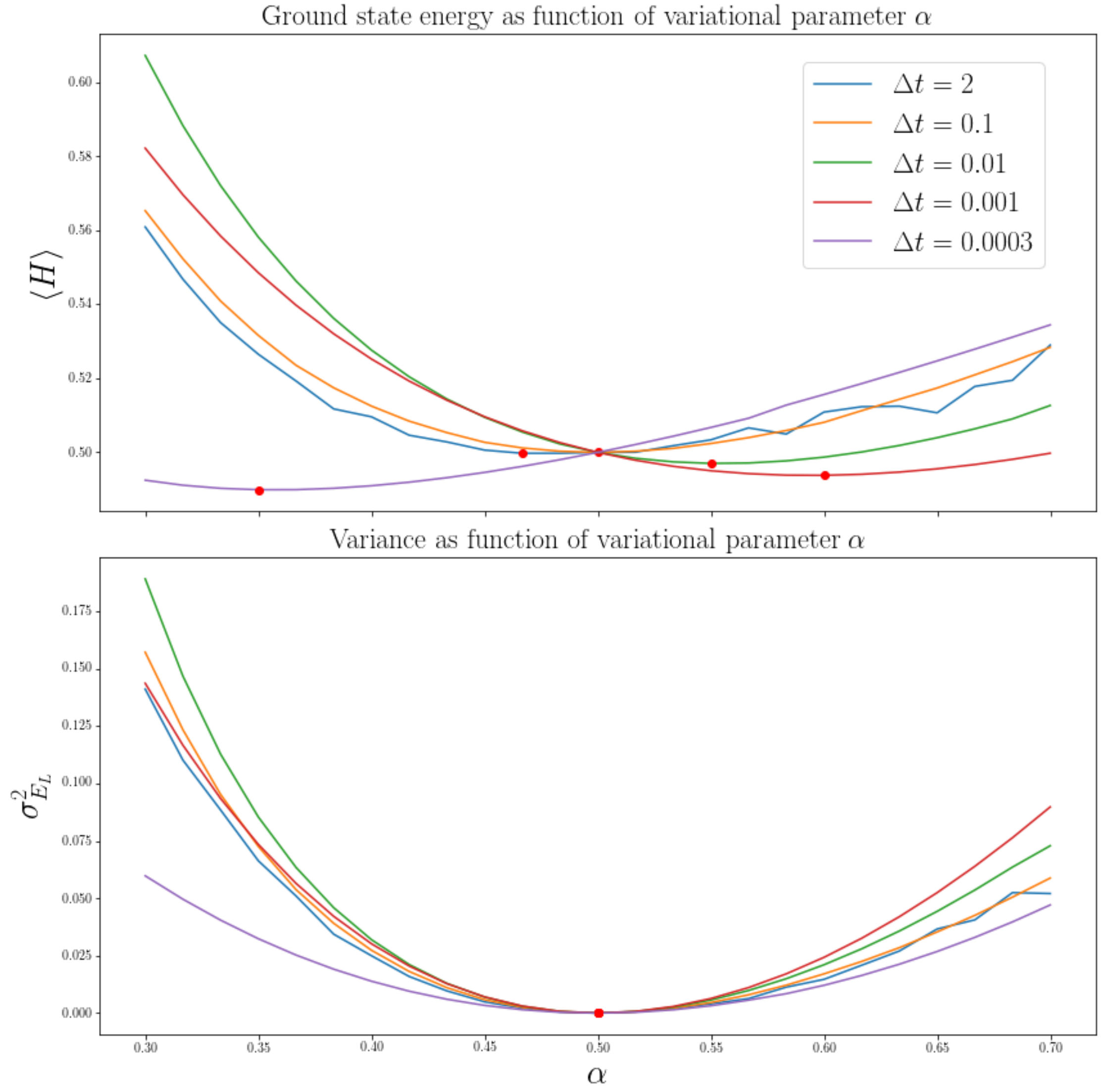

Figure 4: The same plot as shown in Figure 3, shown for multiple different values for Δt. We see that both too large, and to small values yield less than ideal results.