University of Oslo
FYS4411
Computational physics II: Quantum mechanical systems

Project 1

Bendik Samseth

March 6, 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

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

2.2 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.

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

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

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

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**:

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

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

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

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:

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

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:

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

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: $y = E_L$
1: $y = -2 \times N \times D \times \Psi_T(\boldsymbol{R})$
2: **for** $i = 1$ **to** N **do**
3: **for** $d = 1$ **to** D **do**
4: $\boldsymbol{R}_+ \leftarrow \boldsymbol{R} + h\boldsymbol{e}_{i,d}$
5: $\boldsymbol{R}_- \leftarrow \boldsymbol{R} - h\boldsymbol{e}_{i,d}$
6: $y \leftarrow y + \Psi_T(\boldsymbol{R}_+) + \Psi_T(\boldsymbol{R}_-)$
7: **end for**
8: **end for**
9: $y \leftarrow -y/2h^2$
10: $y \leftarrow y/\Psi_T(\boldsymbol{R}) + \sum_{i=1}^{N} V_{ext} + \sum_{i<j}^{N} V_{int}$

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 di-

vergence of (15):

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

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

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

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

and

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

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
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.

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 impli-

cations 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):

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

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

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

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 $\overset{\text{d}}{\leftarrow} P(\boldsymbol{R}, \boldsymbol{\alpha})$

```
samples ← empty list
R ← randomly initialized matrix of positions
for M iterations do
    for every particle i ∈ [1, N] do
        Δr_i^* ← (1/2) 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
```

4 Results

4.1 Standard Metropolis VMC

We now apply the standard Metropolis sampling algorithm described in Algorithm2 to a variety of different settings. Table 1 shows selected results.

We see that all the results have $\langle E_L \rangle$ equal (within numerical precision) to the exact result. 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.

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.

Similarly, comparing the run times with 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, remember to expect the difference to be reduced when we include interaction, as this reduced the analytical approach to the same complexity as the numerical one.

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**	$\langle E_L \rangle$	Var(E_L)	Acceptance Rate	Time (ms)
ON	1	1	5.000000e-01	0.000000e+00	0.820	0
OFF	1	1	5.000000e-01	6.050715e-15	0.900	0
ON	1	10	5.000000e+00	0.000000e+00	0.869	0
OFF	1	10	5.000000e+00	1.811884e-13	0.885	1
ON	1	100	5.000000e+01	0.000000e+00	0.868	22
OFF	1	100	5.000000e+01	-1.818989e-12	0.865	748
ON	1	500	2.500000e+02	0.000000e+00	0.863	393
OFF	1	500	2.500000e+02	1.018634e-10	0.864	48496
ON	2	1	1.000000e+00	0.000000e+00	0.740	0
OFF	2	1	1.000000e+00	2.176037e-14	0.720	0
ON	2	10	1.000000e+01	0.000000e+00	0.777	0
OFF	2	10	1.000000e+01	4.973799e-13	0.781	2
ON	2	100	1.000000e+02	0.000000e+00	0.794	18
OFF	2	100	9.999999e+01	-2.182787e-11	0.794	936
ON	2	500	5.000000e+02	0.000000e+00	0.793	476
OFF	2	500	5.000000e+02	2.735760e-09	0.790	104129
ON	3	1	1.500000e+00	0.000000e+00	0.770	0
OFF	3	1	1.500000e+00	2.353673e-14	0.740	0
ON	3	10	1.500000e+01	0.000000e+00	0.740	0
OFF	3	10	1.500000e+01	3.694822e-13	0.750	3
ON	3	100	1.500000e+02	0.000000e+00	0.746	22
OFF	3	100	1.500000e+02	-1.964509e-10	0.738	1597
ON	3	500	7.500000e+02	0.000000e+00	0.743	528
OFF	3	500	7.499999e+02	7.334165e-09	0.737	176949