FYS4411 - Project 1

Espen Kristian Wulff Wold

Abstract

We build a Markov Chain Monte Carlo numerical integration routine to solve the many-dimensional integrals needed to simulate interacting particles trapped in harmonic potentials.

Note: the code for this project is available at https://github.com/EspenWold/ComputationalQuantumPhysics, and is needed to fully understand this report and be able to recreate its results. The code needed to run the various tasks is grouped in blocks in the file main.py that will run the task in question and make it easy to tweak inputs or dive deeper into the code.

1. Introduction

In this project we will solve unwieldly integrals by use of a Markov Chain Monte Carlo (MCMC) numerical scheme. In short MCMC is a way for us to sample points from a Markov Chain such that they are guaranteed in the limit of infinite samples to be sampled from a certain distribution, π. Our points are generated as a Markov Chain by effecting some transition from the current stats, giving a probability for state x_j

$$P(X_{n+1} = x_j) = \sum_i P(X_n = x_i)t(x_i, x_j)$$

Where $t(x_i, x_j)$ denotes the probability that if we are in position x_i, we will generate a move to x_j and the sum runs over all possible states. Since we would like our Markov Chain to converge, we must demand that the final distribution is stationary, i.e. the 'flow' from x_i to x_j must equal the one from x_j to x_i. This requirement is called detailed balance:

$$\pi(x_i)t(x_i, x_j) = \pi(x_j)t(x_j, x_i)$$

If we split our transition probabilities $t(x_i, x_j)$ into the probability to suggest a move $g(x_i, x_j)$ and the probability to accept that same move $A(x_i, x_j)$, we can rearrange detailed balance into

$$\frac{A(x_i, x_j)}{A(x_j, x_i)} = \frac{\pi(x_j)g(x_j, x_i)}{\pi(x_i)g(x_i, x_j)}$$

If we make the choice that the greatest of these acceptance ratios should always be one, we end up with the acceptance ratio that defines the Metropolis-Hastings algorithm

$$A(x_i, x_j) = \min\left(1, \frac{\pi(x_j)g(x_j, x_i)}{\pi(x_i)g(x_i, x_j)}\right) \tag{1}$$

if we accept new points whenever a uniform random number in the range 0-1 is smaller than the acceptance

ratio, we will eventually be sampling from the desired distribution π. (There are some caveats to this, e.g. we have to generate moves that make it possible to reach the whole region of interest, but in practice these constraints are usually met.)

In the special case where the transition probabilities are symmetric (i.e. we are equally likely to suggest a move from x_i to x_j as from x_j to x_i) the acceptance ratio reduces to

$$A(x_i, x_j) = \min\left(1, \frac{\pi(x_j)}{\pi(x_i)}\right)$$

This acceptance ratio defines the Metropolis algorithm, sometimes called brute force Metropolis due to the simplicity of uniform (i.e. dumb) moves.

Lastly, we note that the only thing we need to know about a distribution in order to sample from it in this way is how to evaluate in each point some function that is proportional to the distribution. That we need only proportionallity saves us from bothering with difficult normalisation factors.

2. Theory

We will model our system as simple spherical (S) or elliptical (E) harmonic potentials given by

$$V_S(\boldsymbol{r}) = \frac{1}{2}m\omega^2 r^2 \tag{2}$$

$$V_E(\boldsymbol{r}) = \frac{1}{2}m\left[\omega^2(x^2 + y^2) + \omega_z^2 z^2\right] \tag{3}$$

where $\boldsymbol{r}$ is the position vector for a particle interacting with the potential, r is the length of the position vector, m is the particle mass, and the frequencies ω and ω_z defines the strengths of the potentials.

We have given the elliptical potential V_E for the three dimensional case while the spherical potential V_S retains

its form if we consider the vector $\boldsymbol{r}$ to be one, two or three dimensions.

For the non-interacting case, our Hamiltonian for an N-particle system then is simply the sum of single particle Hamiltonians

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\boldsymbol{r}_i) \right) \qquad (4)$$

where the external potential V_{ext} can be either V_S or V_E and $\boldsymbol{r}_i$ denotes the poristion of the ith particle. To model a simple pairwise interaction, we will consider a repulsive interaction on the form

$$V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) = \begin{cases} \infty & r_{ij} \leq a \\ 0 & r_{ij} > a \end{cases} \qquad (5)$$

where $r_{ij} = r_{ji} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$ is the distance between the two particles and $a \geq 0$ is a 'hard diameter' for the particles. Counting the interaction between each pair of particles once gives us the a Hamiltonian for this simply interacting system:

$$H = \sum_i \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\boldsymbol{r}_i) \right) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \qquad (6)$$

where $\sum_{i<j}$ is a shorthand for the summation over every unique pair of particles $\sum_{i=1}^N \sum_{j=i+1}^N$.

3. Method

We want to find the ground state wave function of a system described by (6) by use of a variational Monte Carlo scheme. As our trial wavefunction for this strategy, we will take one separable into single-particle wavefunctions g (proportional to the harmonic oscillator ground state) and particle corrolation wave functions f.

$$\Psi_T(\boldsymbol{r}) = \Psi_T(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta)$$
$$= \left[\prod_i g(\alpha, \beta, \boldsymbol{r}_i) \right] \left[\prod_{i<j} f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) \right] \qquad (7)$$

α and β are variational parameters and a is the parameter controlling the interaction term in the Hamiltonian. The functions g and f are given by

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

$$f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) = \begin{cases} 0 & r_{ij} \leq a \\ \left(1 - \frac{a}{r_{ij}}\right) & r_{ij} > a \end{cases} \qquad (9)$$

Since we would like to compute ground state energies by a numerical scheme rather than by analytical methods, we rewrite the integrals in the standard calculation of the expectation value of the energy as

$$\begin{aligned} \langle E[\boldsymbol{\alpha}] \rangle &= \frac{\int d\boldsymbol{r} \Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) H(\boldsymbol{r}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})}{\int d\boldsymbol{r} \Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})} \\ &= \int d\boldsymbol{r} \left(\frac{\Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})}{\int d\boldsymbol{r} \Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})} \right. \\ &\quad \left. \times \frac{1}{\Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})} H \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha}) \right) \\ &= \int d\boldsymbol{r} P(\boldsymbol{r}, \boldsymbol{\alpha}) E_L(\boldsymbol{r}, \boldsymbol{\alpha}) \\ &\approx \frac{1}{N} \sum_{1=1}^N E_L(\boldsymbol{r}_i, \boldsymbol{\alpha}) \qquad \boldsymbol{r}_i \sim P(\boldsymbol{r}, \boldsymbol{\alpha}) \end{aligned} \qquad (10)$$

where we denote the position in configuration space by $\boldsymbol{r}$ and all variational parameters by $\boldsymbol{\alpha}$. We have introduced the probability density of the trial wavefunction $P(\boldsymbol{r}, \boldsymbol{\alpha})$ and the local energy $E_L(\boldsymbol{r}, \boldsymbol{\alpha})$. As the last line indicates, we can estimate this integral by evaluating the local energy at N positions $\boldsymbol{r}_i$ generated from the distribution $P(\boldsymbol{r}, \boldsymbol{\alpha})$. We will employ a Marcov Chain Monte Carlo sampling routine to generate samples from a distribution proportional to $P(\boldsymbol{r}, \boldsymbol{\alpha})$.

Since any trial wavefunction we might use can be expanded in a complete, orthonormal basis of energy eigenstates, we will have $\langle E[\boldsymbol{\alpha}] \rangle \geq E_{\text{Ground state}}$ with equality if and only if the trial wavefunction in question is exactly the ground state. Hence, if the trial wavefunction can be equal to the ground state wavefunction if the parameters are set correctly, the problem of finding the ground state energy is simply a minimisation of the expected energy as a function of the variational parameters:

$$E_{\text{Ground state}} = \arg\min_{\boldsymbol{\alpha}} \left[\langle E[\boldsymbol{\alpha}] \rangle \right] \qquad (11)$$

Ideally, we would like to have an analytical expression for the local energy, but for a given problem, it might be difficult or impossible to find such an expression. However, as is often the case, in our Hamiltonian the only troublesome term is the kinetic energy term due to the second derivative. If we can't or don't want to calculate the exact derivative, we could make a numerical estimation of it. Taking two Taylor expansions of our function around a point x_0

$$f(x_0+h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)h^2}{2} + \frac{f'''(x_0)h^3}{6} + \mathcal{O}(h^4) \qquad (12)$$

$$f(x_0-h) = f(x_0) - f'(x_0)h + \frac{f''(x_0)h^2}{2} - \frac{f'''(x_0)h^3}{6} + \mathcal{O}(h^4) \qquad (13)$$

and adding them together gets us

$$f''(x_0) = \frac{f(x_0 + h) + f(x_0 - h) - 2f(x_0)}{h^2} + \mathcal{O}(h^2) \qquad (14)$$

which - disregarding the $\mathcal{O}(h^2)$ term - will be very close to the exact value for small h. Using this approximation, we could estimate $\nabla^2 \Psi_T$ by evaluating Ψ_T at three points which we should always be able to do.

Summing up these various components, our strategy for evaluating the energy given a certain α takes the following form:

Algorithm 1 Basic MCMC algorithm

1: Define the number of MCMC cycles, M
2: Define the number of particles, N
3: Define the number of dimensions, D
4: Define the step size, S
5: Initiate zero valued array $positions$[N][D]
6: $energy \leftarrow 0$
7: **for** particleIndex = 1:N **do**
8: $\quad positions[particleIndex] \leftarrow S \times \text{randNormal}(0, 1)$
9: **for** m = 1:M **do**
10: $\quad n \leftarrow \text{randInt}(0, N)$
11: $\quad newPos \leftarrow positions[n] + S \times \text{randNormal}(0, 1)$
12: $\quad A \leftarrow \text{WF}(newPos)^2/\text{WF}(positions[n])^2$
13: $\quad$ **if** A > random() **then**
14: $\quad\quad positions[n] \leftarrow newPos$
15: $\quad energy \leftarrow energy + E_L(positions, \alpha)$
16: **return** $energy/M$

Where randNormal(0,1) is a function giving us a random number from a normal distriution with mean zero and variance of one for each dimension. The other random functions have unitform distributions and give an integer in the supplied range or a number between zero and one respectively. WF is our single particle contribution to the wavefunction, and E_L is our local energy function.

4. Task a) Local energy

We want to find the local energy

$$E_L(\boldsymbol{r}) = \frac{1}{\Psi_T(\boldsymbol{r})} H \Psi_T(\boldsymbol{r})$$
$$= \sum_i \left(\frac{-\hbar}{2m} \frac{1}{\Psi_T(\boldsymbol{r}_i)} \nabla_i^2 \Psi_T(\boldsymbol{r}_i) + V_{ext}(\boldsymbol{r}_i) \right) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{15}$$

for our system described by the Hamiltonian (6) and trial wavefunction (17). We also want to find the so-called 'quantum force' F which will play the part of a drift force in our diffusion Monte Carlo scheme.

$$F(\boldsymbol{r}) = \frac{2\nabla\Psi_T(\boldsymbol{r})}{\Psi_T(\boldsymbol{r})} \tag{16}$$

Using the shorthand $g(\alpha, \beta, \boldsymbol{r}_i) = \phi(\boldsymbol{r}_i)$ and rewriting $f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) = \exp u(r_{ij})$ where $u(r_{ij}) = \ln \big[f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) \big]$, we get

$$\Psi_T(\boldsymbol{r}) = \Psi_T(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta)$$
$$= \left[\prod_i \phi(\boldsymbol{r}_i) \right] \exp \left[\sum_{i<j} u(r_{ij}) \right] \tag{17}$$

Let's start by applying a specific ∇_k to the trial wavefunction. Since ∇_k commutes with any $\phi(\boldsymbol{r}_{i \neq k})$ it will simply pick out $\phi(\boldsymbol{r}_k)$ for the repeated product.

$$\nabla_k \Psi_T(\boldsymbol{r}) = \left(\nabla_k \left[\prod_i \phi(\boldsymbol{r}_i) \right] \right) \exp \left[\sum_{i<j} u(r_{ij}) \right]$$
$$+ \left[\prod_i \phi(\boldsymbol{r}_i) \right] \left(\nabla_k \exp \left[\sum_{i<j} u(r_{ij}) \right] \right)$$
$$= \nabla_k \phi(\boldsymbol{r}_k) \left[\prod_{i \neq k} \phi(\boldsymbol{r}_i) \right] \exp \left[\sum_{i<j} u(r_{ij}) \right]$$
$$+ \left[\prod_i \phi(\boldsymbol{r}_i) \right] \exp \left[\sum_{i<j} u(r_{ij}) \right] \nabla_k \left(\sum_{i<j} u(r_{ij}) \right) \tag{18}$$

For the sum over $u(r_{ij})$, ∇_k will zero out any pair not including the particle k, giving (since u is symmetric in the particle indices)

$$\nabla_k \left(\sum_{i<j} u(r_{ij}) \right) = \sum_{l \neq k} \nabla_k u(r_{kl})$$
$$= \sum_{l \neq k} (\nabla_k r_{kl}) u'(r_{kl})$$
$$= \sum_{l \neq k} (\nabla_k |\boldsymbol{r}_k - \boldsymbol{r}_l|) u'(r_{kl}) \tag{19}$$
$$= \sum_{l \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl})$$

Considering that

$$\nabla_k \phi(\boldsymbol{r}_k) = \nabla_k \exp \big[-\alpha(x_k^2 + y_k^2 + \beta z_k^2) \big]$$
$$= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \exp \big[-\alpha(x_k^2 + y_k^2 + \beta z_k^2) \big]$$
$$= \left(-2\alpha x_k, -2\alpha y_k, -2\alpha\beta z_k \right) \exp \big[-\alpha(x_k^2 + y_k^2 + \beta z_k^2) \big]$$
$$= -2\alpha \big(x_k, y_k, \beta z_k \big) \phi(\boldsymbol{r}_k) = -2\alpha \boldsymbol{\beta}_k \phi(\boldsymbol{r}_k) \tag{20}$$

(where we have denoted the vector $\big(x_k, y_k, \beta z_k \big)$ by $\boldsymbol{\beta}_k$), we get

$$\nabla_k \Psi_T(\boldsymbol{r}) = -2\alpha \boldsymbol{\beta}_k \left[\prod_i \phi(\boldsymbol{r}_i) \right] \exp \left[\sum_{i<j} u(r_{ij}) \right]$$
$$+ \left(\sum_{l \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \right) \left[\prod_i \phi(\boldsymbol{r}_i) \right] \exp \left[\sum_{i<j} u(r_{ij}) \right]$$
$$= \left[-2\alpha \boldsymbol{\beta}_k + \sum_{l \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \right] \Psi_T(\boldsymbol{r}) \tag{21}$$

Inserting this into the expression for the quantum force, we get the quantum force on the particle k (or at least the part of the whole system's quantum force vector that applies to the part of configuration space pertaining to the position of particle k):

$$
\begin{aligned}
F_k(\boldsymbol{r}) &= \frac{2\nabla_k \Psi_T(\boldsymbol{r})}{\Psi_T(\boldsymbol{r})} \\
&= 2\sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) - 4\alpha\boldsymbol{\beta}_k
\end{aligned}
\tag{22}
$$

Moving on to the second derivative, we get

$$
\begin{aligned}
\nabla_k^2 \Psi_T(\boldsymbol{r}) &= \nabla_k \cdot \nabla_k \Psi_T(\boldsymbol{r}) \\
&= \nabla_k \cdot \left[-2\alpha\boldsymbol{\beta}_k + \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \right] \Psi_T(\boldsymbol{r}) \\
&+ \left[-2\alpha\boldsymbol{\beta}_k + \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \right] \cdot \nabla_k \Psi_T(\boldsymbol{r}) \\
&= -2\alpha\nabla_k \cdot \boldsymbol{\beta}_k + \sum_{l\neq k} \left[\left(\nabla_k \cdot \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} \right) u'(r_{kl}) \right. \\
&+ \left. \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} \cdot \nabla_k u'(r_{kl}) \right] \Psi_T(\boldsymbol{r}) \\
&+ \left[4\alpha^2(x_k^2 + y_k^2 + \beta^2 z_k^2) - 4\alpha\boldsymbol{\beta}_k \cdot \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \right. \\
&+ \left. \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \cdot \sum_{h\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_h}{r_{kh}} u'(r_{kh}) \right] \Psi_T(\boldsymbol{r})
\end{aligned}
\tag{23}
$$

Factoring out the trial wavefunction for readability, and using

$$
\begin{aligned}
\nabla_k \cdot \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} &= \frac{\nabla_k \cdot (\boldsymbol{r}_k - \boldsymbol{r}_l)}{r_{kl}} + (\boldsymbol{r}_k - \boldsymbol{r}_l) \cdot \nabla_k \frac{1}{r_{kl}} \\
&= \frac{1}{r_{kl}} + (\boldsymbol{r}_k - \boldsymbol{r}_l) \cdot \frac{\nabla_k r_{kl}}{r_{kl}^2} \\
&= \frac{1}{r_{kl}} + \frac{(\boldsymbol{r}_k - \boldsymbol{r}_l) \cdot (\boldsymbol{r}_k - \boldsymbol{r}_l)}{r_{kl}^3} = \frac{2}{r_{kl}}
\end{aligned}
$$

and

$$
\frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} \cdot \nabla_k u'(r_{kl}) = \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} \cdot \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u''(r_{kl}) = u''(r_{kl})
$$

we arrive at

$$
\begin{aligned}
\frac{1}{\Psi_T(\boldsymbol{r})} \nabla_k^2 \Psi_T(\boldsymbol{r}) &= \sum_{l\neq k} \left[\frac{2}{r_{kl}} u'(r_{kl}) + u''(r_{kl}) \right] \\
&- 2\alpha\nabla_k \cdot \boldsymbol{\beta}_k + 4\alpha^2(x_k^2 + y_k^2 + \beta^2 z_k^2) \\
&- 4\alpha\boldsymbol{\beta}_k \cdot \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'(r_{kl}) \\
&+ \sum_{l\neq k}\sum_{h\neq k} \frac{(\boldsymbol{r}_k - \boldsymbol{r}_l) \cdot (\boldsymbol{r}_k - \boldsymbol{r}_h)}{r_{kl}r_{kh}} u'(r_{kl})u'(r_{kh})
\end{aligned}
\tag{24}
$$

Inserting this into (15) and choosing natural units with lengths in units of $a_{\text{ho}} = \sqrt{\frac{\hbar}{m\omega}}$ so that $\boldsymbol{r}_i \to \frac{\boldsymbol{r}_i}{a_{\text{ho}}}$ our Hamiltonian becomes

$$
\begin{aligned}
H &= \sum_i \left(\frac{-\hbar^2}{2ma_{\text{ho}}^2} \nabla_i^2 + a_{\text{ho}}^2 V_{ext}(\boldsymbol{r}_i) \right) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \\
&= \sum_i \left(\frac{-\hbar\omega}{2} \nabla_i^2 + \hbar\omega v_{ext}(\boldsymbol{r}_i) \right) + \hbar\omega \sum_{i<j} v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \\
&= \frac{1}{2} \sum_i \left(-\nabla_i^2 + \boldsymbol{\gamma}_i^2 \right) + \sum_{i<j} v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)
\end{aligned}
\tag{25}
$$

where

$$
v_{ext}(\boldsymbol{r}_i) = \frac{a_{\text{ho}}^2}{\hbar\omega} V_{ext}(\boldsymbol{r}_i) = \frac{a_{\text{ho}}^2}{\hbar\omega} \frac{m\omega^2}{2}[x_i^2 + y_i^2 + \gamma^2 z_i^2] = \frac{\gamma_i^2}{2}
$$

is now dimensionless, as is $v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) = V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)$ which in unaffected by the scaling due to being either zero or infinity. We have introduces the vector $\boldsymbol{\gamma}_i = (x_i, y_i, \gamma z_i)$ for convenience and $\gamma = \frac{\omega_z}{\omega}$ denotes the potential strength in the z-direction relative to the other axes. In the last line we have introduced an energy scale in units of $\hbar\omega$.

Combining our now dimensionless Hamiltonian with our reslults in (24) we get an expression for the local energy

$$
\begin{aligned}
E_L(\boldsymbol{r}) &= \sum_i \left(-\sum_{j\neq i} \left[\frac{u'(r_{ij})}{r_{ij}} + \frac{u''(r_{ij})}{2} \right] \right. \\
&+ \alpha\nabla_i \cdot \boldsymbol{\beta}_i - 2\alpha^2(x_i^2 + y_i^2 + \beta^2 z_i^2) \\
&+ 2\alpha\boldsymbol{\beta}_i \cdot \sum_{j\neq i} \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{r_{ij}} u'(r_{ij}) \\
&- \sum_{j\neq i}\sum_{k\neq i} \frac{(\boldsymbol{r}_i - \boldsymbol{r}_j) \cdot (\boldsymbol{r}_i - \boldsymbol{r}_k)}{2r_{ij}r_{ik}} u'(r_{ij})u'(r_{ik}) \\
&+ \left. \frac{\gamma_i^2}{2} \right) + \sum_{i<j} v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)
\end{aligned}
\tag{26}
$$

Now if we are interested in the simpler case of non-interacting particles (i.e. $a = 0$) in a spherical harmonic oscillator potential (i.e. $\beta = 1$, $\boldsymbol{\beta}_i = (x_i, y_i, z_i) = \boldsymbol{r}_i$, $\omega_z = \omega$ and $\gamma_i^2 = r_i^2$) then both (22) and (26) simplifies a bit. With $a = 0$, we get $v_{int} = 0$ and $f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) = 1$, and $u(r_{ij})$ and its derivatives goes to zero. The quantum force and local energy becomes

$$
F_k(\boldsymbol{r}) = \frac{2\nabla_k \Psi_T(\boldsymbol{r})}{\Psi_T(\boldsymbol{r})} = -4\alpha\boldsymbol{r}_k
\tag{27}
$$

$$E_L^d(\boldsymbol{r}) = \sum_i \left(\alpha \nabla_i \cdot \boldsymbol{\beta}_i - 2\alpha^2 r_i^2 + \frac{r_i^2}{2} \right)$$

$$= \sum_i \left(\alpha d - 2\alpha^2 r_i^2 + \frac{r_i^2}{2} \right) \quad (28)$$

$$= Nd\alpha + \frac{1 - 4\alpha^2}{2} \sum_i r_i^2$$

Where d denotes the number of dimensions and N the number of particles.

5. Task b) Developing the code

We now want to simulate the behaviour of this simplified system. We will want to know the exact energy for a given trial wavefunction as a basis for comparison. The probability density for our trial wavefunction for particle i alone $\Psi_T(\boldsymbol{r}_i, \alpha) = e^{-\alpha r_i^2}$ (which is of use to us since the wavefunction is separable) is

$$P(\boldsymbol{r}_i, \alpha) = \frac{\Psi_T^* \Psi_T}{\int d\boldsymbol{r}_i \Psi_T^* \Psi_T} = \frac{e^{-2\alpha r_i^2}}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \quad (29)$$

So that we get an expectation value for the energy of

$$\langle E_L^d[\alpha] \rangle = \int d\boldsymbol{r} P(\boldsymbol{r}, \alpha) E_L^d(\boldsymbol{r}, \alpha)$$

$$= Nd\alpha + \frac{1 - 4\alpha^2}{2} \int d\boldsymbol{r} \prod_i^N [P(\boldsymbol{r}_i, \alpha)] \sum_i^N r_i^2$$

$$= Nd\alpha + \frac{1 - 4\alpha^2}{2} \sum_i^N \left[\frac{\int r_i^2 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right]$$

$$= N \left[d\alpha + \frac{1 - 4\alpha^2}{2} \frac{\int r_i^2 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right]$$

$$(30)$$

As we are also interested in the variance, we need the expectation value of the energy squared as well

$$\langle [E_L^d[\alpha]]^2 \rangle = \int d\boldsymbol{r} P(\boldsymbol{r}, \alpha) [E_L^d(\boldsymbol{r}, \alpha)]^2$$

$$= (Nd\alpha)^2 + Nd\alpha(1 - 4\alpha^2) \int d\boldsymbol{r} P(\boldsymbol{r}, \alpha) \sum_i^N r_i^2$$

$$+ \left(\frac{1 - 4\alpha^2}{2} \right)^2 \int d\boldsymbol{r} P(\boldsymbol{r}, \alpha) \left(\sum_i^N r_i^2 \right)^2$$

$$= (Nd\alpha)^2 + N^2 d\alpha(1 - 4\alpha^2) \int d\boldsymbol{r} P(\boldsymbol{r}_i, \alpha) r_i^2$$

$$+ \frac{(1 - 4\alpha^2)^2}{4} \int d\boldsymbol{r} P(\boldsymbol{r}, \alpha) \sum_i^N \sum_j^N r_i^2 r_j^2$$

$$= (Nd\alpha)^2 + N^2 d\alpha(1 - 4\alpha^2) \frac{\int r_i^2 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i}$$

$$+ \frac{(1 - 4\alpha^2)^2}{4} \left[N \frac{\int r_i^4 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right.$$

$$\left. + N(N - 1) \left(\frac{\int r_i^2 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right)^2 \right]$$

$$(31)$$

Combining these expressions, we get a varaiance

$$\sigma^2[\alpha] = \langle [E_L^d[\alpha]]^2 \rangle - \langle E_L^d[\alpha] \rangle^2$$

$$= N \frac{(1 - 4\alpha^2)^2}{4} \left[\frac{\int r_i^4 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right. \quad (32)$$

$$\left. - \left(\frac{\int r_i^2 e^{-2\alpha r_i^2} d\boldsymbol{r}_i}{\int e^{-2\alpha r_i^2} d\boldsymbol{r}_i} \right)^2 \right]$$

In one dimension, the integrals become definite $\int d\boldsymbol{r}_i \rightarrow \int_{-\infty}^{\infty} dr_i$, and have solutions

$$\int_{-\infty}^{\infty} e^{-2\alpha r_i^2} r_i^4 dr_i = \frac{3\sqrt{\pi}}{4(2\alpha)^{\frac{5}{2}}}$$

$$\int_{-\infty}^{\infty} e^{-2\alpha r_i^2} r_i^2 dr_i = \frac{\sqrt{\pi}}{2(2\alpha)^{\frac{3}{2}}}$$

$$\int_{-\infty}^{\infty} e^{-2\alpha r_i^2} dr_i = \frac{\sqrt{\pi}}{\sqrt{2\alpha}}$$

Hence, for one dimension we get

$$\langle E_L^{1\text{D}}[\alpha] \rangle = N \left[\alpha + \frac{1 - 4\alpha^2}{2} \frac{1}{4\alpha} \right] = N \left[\alpha + \frac{1}{8\alpha} - \frac{\alpha}{2} \right]$$

$$= \frac{N}{2} \left(\frac{1}{4\alpha} + \alpha \right)$$

$$(33)$$

$$\sigma^2[\alpha] = N \frac{(1 - 4\alpha^2)^2}{4} \left[\frac{3}{16\alpha^2} - \left(\frac{1}{4\alpha} \right)^2 \right]$$

$$= N \frac{(1 - 4\alpha^2)^2}{2(4\alpha)^2} = \frac{N}{2} \left(\frac{1}{4\alpha} - \alpha \right)^2$$

$$(34)$$

In two dimensions, we get $\int d\boldsymbol{r}_i \to \int_0^\infty \int_0^{2\pi} r_i dr_i d\theta_i$ and the integrals give us soultions

$$\int_0^\infty \int_0^{2\pi} r_i^5 e^{-2\alpha r_i^2} dr_i d\theta_i = \frac{\pi}{4\alpha^3}$$

$$\int_0^\infty \int_0^{2\pi} r_i^3 e^{-2\alpha r_i^2} dr_i d\theta_i = \frac{\pi}{4\alpha^2}$$

$$\int_0^\infty \int_0^{2\pi} r_i e^{-2\alpha r_i^2} dr_i d\theta_i = \frac{\pi}{2\alpha}$$

Giving us for the two dimensional case

$$\langle E_L^{2\mathrm{D}}[\alpha]\rangle = N\left[2\alpha + \frac{1-4\alpha^2}{2}\frac{1}{2\alpha}\right] = N\left[2\alpha + \frac{1}{4\alpha} - \alpha\right]$$
$$= N\left[\frac{1}{4\alpha} + \alpha\right]$$

$$(35)$$

$$\sigma^2[\alpha] = N\frac{(1-4\alpha^2)^2}{4}\left[\frac{1}{2\alpha^2} - \left(\frac{1}{2\alpha}\right)^2\right]$$
$$= N\frac{(1-4\alpha^2)^2}{(4\alpha)^2} = N\left(\frac{1}{4\alpha} - \alpha\right)^2$$

$$(36)$$

And finally, in three dimensions we get $\int d\boldsymbol{r}_i \to \int_0^\infty \int_0^{2\pi} \int_0^\pi r_i^2 \sin\phi_i dr_i d\theta_i d\phi_i$ giving solutions

$$\int_0^\infty \int_0^{2\pi} \int_0^\pi r_i^6 e^{-2\alpha r_i^2} \sin\phi_i dr_i d\theta_i d\phi_i = \frac{15\pi^{\frac{3}{2}}}{4(2\alpha)^{\frac{7}{2}}}$$

$$\int_0^\infty \int_0^{2\pi} \int_0^\pi r_i^4 e^{-2\alpha r_i^2} \sin\phi_i dr_i d\theta_i d\phi_i = \frac{3\pi^{\frac{3}{2}}}{2(2\alpha)^{\frac{5}{2}}}$$

$$\int_0^\infty \int_0^{2\pi} \int_0^\pi r_i^2 e^{-2\alpha r_i^2} \sin\phi_i dr_i d\theta_i d\phi_i = \frac{\pi^{\frac{3}{2}}}{(2\alpha)^{\frac{3}{2}}}$$

and expected energy and variance

$$\langle E_L^{3\mathrm{D}}[\alpha]\rangle = N\left[3\alpha + \frac{1-4\alpha^2}{2}\frac{3}{4\alpha}\right] = N\left[3\alpha + \frac{3}{8\alpha} - \frac{3\alpha}{2}\right]$$
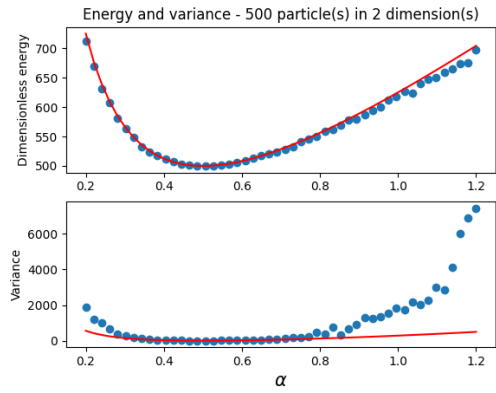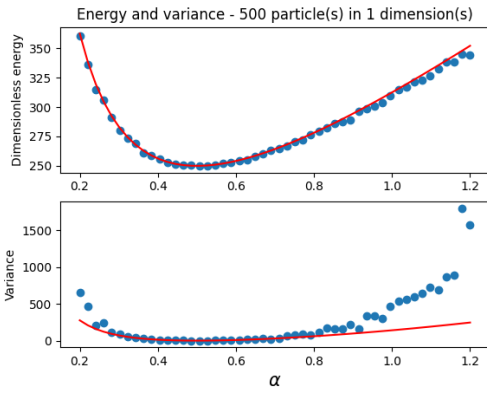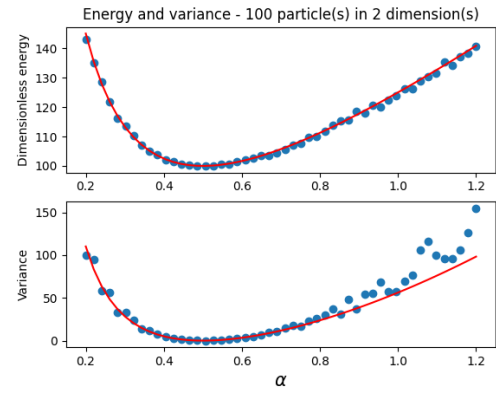$$= \frac{3N}{2}\left(\frac{1}{4\alpha} + \alpha\right)$$

$$(37)$$
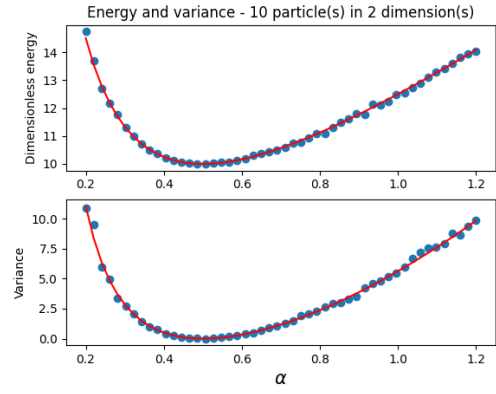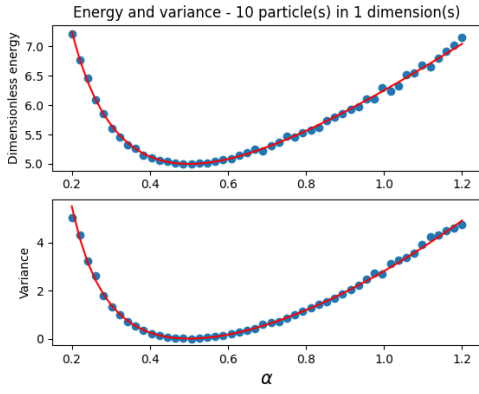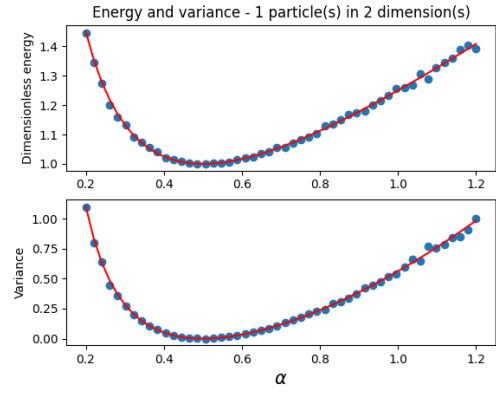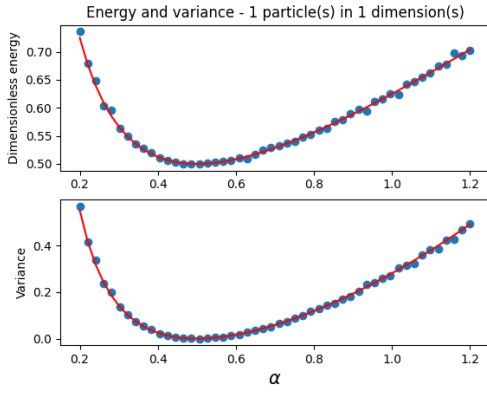
$$\sigma^2[\alpha] = N\frac{(1-4\alpha^2)^2}{4}\left[\frac{15}{(4\alpha)^2} - \left(\frac{3}{4\alpha}\right)^2\right]$$
$$= \frac{3N}{2}\frac{(1-4\alpha^2)^2}{(4\alpha)^2} = \frac{3N}{2}\left(\frac{1}{4\alpha} - \alpha\right)^2$$

$$(38)$$
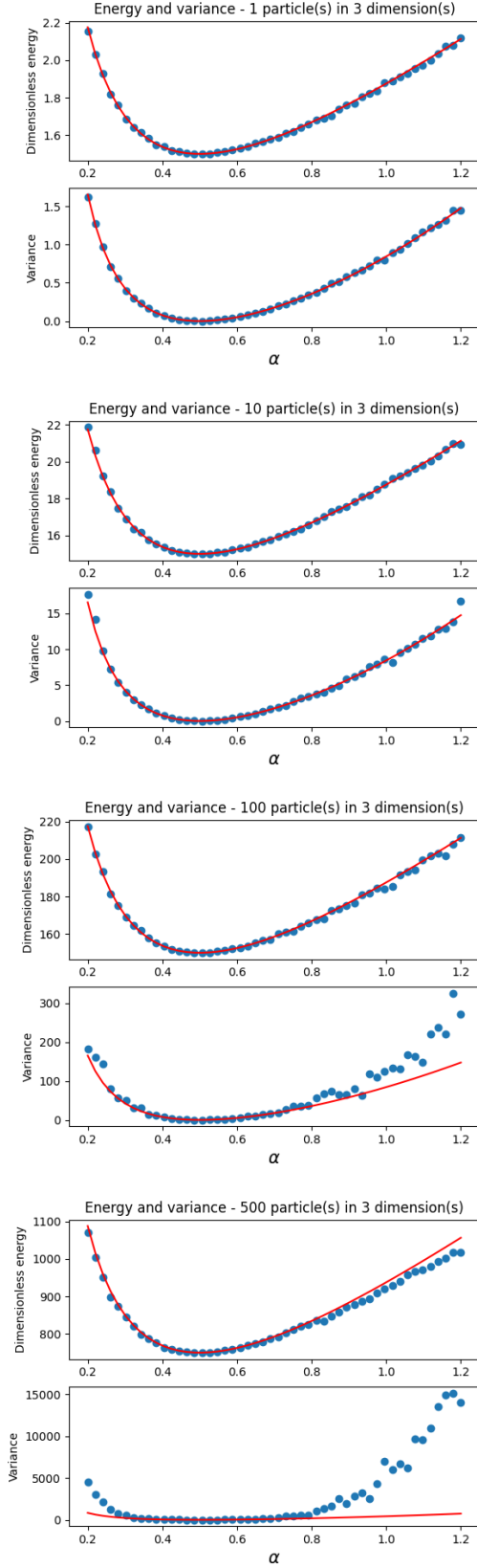
If we did not have the analytical expression (28) for the local energy our recourse would be a numerical estimate based on (14), yielding

$$E_L^d(\boldsymbol{r}) = \sum_i \left(\frac{-1}{2\Psi(\boldsymbol{r}_i)}\nabla_i^2\Psi(\boldsymbol{r}_i) + v_S(\boldsymbol{r}_i)\right)$$
$$\approx \frac{1}{2}\sum_i \left(\frac{-1}{\Psi(\boldsymbol{r}_i)}\sum_{n=1}^d \left[\tilde{\nabla}_i^2\Psi_T(\boldsymbol{r}_i)\right]_n + r_i^2\right)$$

$$(39)$$

Where $\left[\tilde{\nabla}_i^2\Psi_T(\boldsymbol{r}_i)\right]_n = \frac{\Psi_T(x_{ni}+h)+\Psi_T(x_{ni}-h)-2\Psi_T(x_{ni})}{h^2}$ is the contribution to the numerical estimate from dimension n and x_{ni} denotes the coordinate for the position in dimension n for particle i.

Finally we are ready to run our simulations of these systems for 1, 10, 100 and 500 atoms in one, two and three dimensions. We run all simulations with 100.000 MCMC cycles and a uniform step size, first with the analytical calculation of the local energy. Results are plotted in the following figures.

Energy and variance - 1 particle(s) in 1 dimension(s)

Energy and variance - 1 particle(s) in 2 dimension(s)

Energy and variance - 10 particle(s) in 1 dimension(s)

Energy and variance - 10 particle(s) in 2 dimension(s)

Energy and variance - 100 particle(s) in 1 dimension(s)

Energy and variance - 100 particle(s) in 2 dimension(s)

Energy and variance - 500 particle(s) in 1 dimension(s)

Energy and variance - 500 particle(s) in 2 dimension(s)

Energy and variance - 1 particle(s) in 3 dimension(s)

Energy and variance - 10 particle(s) in 3 dimension(s)

Energy and variance - 100 particle(s) in 3 dimension(s)

Energy and variance - 500 particle(s) in 3 dimension(s)

Overall, we see that the simulation reproduces our analytical results quite faithfully, with the exception that the variance diverges from the analytical curve at high and low values for α. The effect is markedly worse both for higher

particle numbers and higher dimensions. This might be expected from our implementation, which moves one random particle each MCMC cycle. This means each simulation run will be comparable in terms of computational resources, but it also means we are making comparatively smaller changes to the systems with many particles. We should thus expect the simulation to need more cycles to be as accurate for systems with more particles.

Another systematic effect observed was that acceptance rate steadily increased for higher α in each run, while also being lower in higher dimensions. For one dimension it ran from 80% to 58% while for three dimensions it declined from 62% to 27%. Particle numbers had no visible effect. This is probably entirely an effect of using a uniform step size along each dimension across all runs, which gives larger steps for higher dimensions. The correlation with α-values is to be expected once we recall that the parameter controls the width of the gaussians used in our trial wavefunction: higher α-values gives more narrow wavefunctions which are easier to 'step outside of'.

Repeating these calculations using the numerical differentiation given by (39), we get results that look identical in terms of simulation results and acceptance rates, but that take about twice the time as measured by a simple timing scheme using some inbuilt Python clock function. Timing just the part of the routine that updates the local energy estimate each cycle reveals that the numerical version takes about three times the time of the analytical expression.

6. Task c) Adding importance sampling

We now want to pick candidate positions in a more efficient way than by simple random moves. Taking inspiration from diffusion processes, we will model the moves as a random walk under the influence of a drift force given in out case by (27).

$$\boldsymbol{r}_i^{n+1} = \boldsymbol{r}_i^n + \eta\sqrt{\tau} + DF(\boldsymbol{r}_i^n)\tau \qquad (40)$$

where upper indices indicate the number in the Markov Chain, lower indices denote particle number as usual, η is a normally distributed random variable with mean zero and standard deviation set to one, τ is a time increment playing the role of tuning the step size, and D is a diffusion constant set to one half. Intuitively, we can expect that this action of pushing our walker in the direction of the gradient of the wavefunction will increase the likelyhood of picking 'good' positions and hence increase our acceptance rate. We should thus be able to get by with fewer cycles, at the cost of spending more computation on calculating the quantum force.

This new way of generating moves is not symmetric, and so we can no longer rely on the brute force Metropolis algorithm where we need only compare the propabilities of the current position and the candidate position. Instead we need to use the Metropolis-Hastings acceptance ratio (1)

which accounts for the different transition probabilities. So what will the transition probabilities be for our moves generated by (40)? We will skip a full derivation, and just note that the only source of randomness is in the Gaussian variable η. Hence, we should expect the probability distribution for $\boldsymbol{r}_i^{n+1}$ to follow a Gaussian distribution centred on $\boldsymbol{r}_i^n + DF(\boldsymbol{r}_i^n)\tau$. This is indeed what we get:

$$g(x_i, x_j) = \frac{1}{\kappa} \exp\left(\frac{-(x_j - x_i - DF(x_i)\tau)^2}{4D\tau}\right) \quad (41)$$

Here κ is a normalisation factor we can safely ignore since it will cancel when we divide one transition probability with another.

Running our simulations with the exact same setup as in b), we get significantly higher accptance ratios (98% to 72% in one dimension and 95% to 48% in three) and we clearly get less deviations from exact values. Compare for instance these plots with the corresponding ones from b):
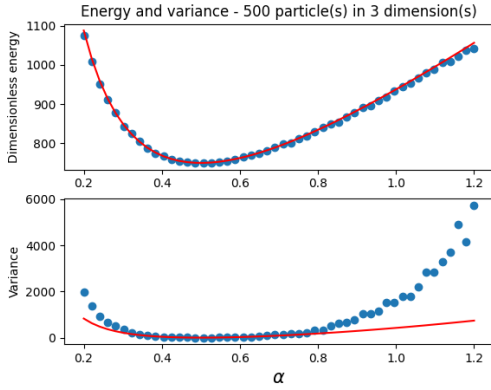
However, our simulation took twice the time of the brute force version. The interesting question is of course how many cycles we need under this scheme of importance sampling to get results comparable with what the brute force version achieved in 100.000 cycles. A bit of trial and error showed that we could get results very similar to the original simulation using 35.000 cycles with importance sampling, with most of the plots difficult to distinguish between the two versions and the importance sampling variant slightly better for many paticles in three dimesions. However, with the added computation of calculating the quantum force we still spent about $\frac{2}{3}$ of the time despite doing only about $\frac{1}{3}$ of the cycles. So in sum we made some efficiency improvement, but nothing monumental.

7. Task d) Finding the best parameters

In the above, we have used the fact that we know the analytical solutions to simply calculate the expectation value for the energy at an interval of variational parameters that we knew would include the ground state. If we instead want our program to find the optimal parameters without wasting Monte Carlo cycles, we must build a solver for the minimisation problem posed by (11).

In this effort, it will be very useful to be able to calculate the gradient of the expected energy with regards to the variational parameters. Suppressing the explicit notation for the wave function's dependence on $\boldsymbol{r}$ and $\boldsymbol{\alpha}$ and the Hamiltonian's dependence on $\boldsymbol{r}$, we have

$$
\begin{aligned}
\nabla_\alpha \langle E[\boldsymbol{\alpha}]\rangle &= \nabla_\alpha \left(\frac{\int d\boldsymbol{r}\,\Psi_T^* H \Psi_T}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T}\right) \\
&= \frac{\int d\boldsymbol{r}\,\nabla_\alpha(\Psi_T^* H \Psi_T)}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T} - \frac{\int d\boldsymbol{r}\,\Psi_T^* H \Psi_T}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T}\frac{\int d\boldsymbol{r}\,\nabla_\alpha(\Psi_T^* \Psi_T)}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T} \\
&= \frac{\int d\boldsymbol{r}\,[(\nabla_\alpha\Psi_T^*)H\Psi_T + \Psi_T^* H(\nabla_\alpha\Psi_T)]}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T} \\
&\quad - \int d\boldsymbol{r}\,[PE_L]\frac{\int d\boldsymbol{r}\,[(\nabla_\alpha\Psi_T^*)\Psi_T + \Psi_T^*(\nabla_\alpha\Psi_T)]}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T} \\
&= 2\left[\frac{\int d\boldsymbol{r}\,\Psi_T^* H(\nabla_\alpha\Psi_T)}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T} - \langle E_L\rangle\frac{\int d\boldsymbol{r}\,\Psi_T^*(\nabla_\alpha\Psi_T)}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T}\right] \\
&= 2\left[\frac{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T \frac{1}{\Psi_T} H \Psi_T \frac{\nabla_\alpha\Psi_T}{\Psi_T}}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T}\right. \\
&\quad \left. - \langle E_L\rangle\frac{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T \frac{\nabla_\alpha\Psi_T}{\Psi_T}}{\int d\boldsymbol{r}\,\Psi_T^* \Psi_T}\right] \\
&= 2\left[\int d\boldsymbol{r}\,[PE_L\frac{\nabla_\alpha\Psi_T}{\Psi_T}]\right. \\
&\quad \left. - \langle E_L\rangle\int d\boldsymbol{r}\,[P\frac{\nabla_\alpha\Psi_T}{\Psi_T}]\right] \\
&= 2\left[\langle E_L \frac{\nabla_\alpha\Psi_T}{\Psi_T}\rangle - \langle E_L\rangle\langle \frac{\nabla_\alpha\Psi_T}{\Psi_T}\rangle\right]
\end{aligned}
$$
$$(42)$$

where we have used that $(\nabla_\alpha\Psi_T^*) = (\nabla_\alpha\Psi_T)^*$ and that $\langle\Psi_A|\hat{F}\Psi_B\rangle = \langle\hat{F}\Psi_A|\Psi_B\rangle$ for hermetian operators $\hat{F}$. It should be obvious from the derivation that this gradient can be estimated by the same Markov Chain Monte Carlo scheme we have been using to estimate the energy, provided we evaluate not only the local energy at each sampled point but also $\frac{\nabla_\alpha\Psi_T}{\Psi_T}$. For our trial wavefunction in the case of non-interacting particles and spherical potential, we get

$$\frac{\nabla_\alpha\Psi_T}{\Psi_T} = \frac{\frac{\partial}{\partial\alpha}\Psi_T}{\Psi_T} = \frac{\frac{\partial}{\partial\alpha}\exp\left[-\alpha\sum_i r_i^2\right]}{\exp\left[-\alpha\sum_i r_i^2\right]} = -\sum_i r_i^2 \quad (43)$$

Now that we have a workable expression for the gradient, we can make an intial guess for our variational parameters $\boldsymbol{\alpha}_0$ and simply produce our next parameter values by

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n - \gamma\nabla_\alpha\langle E[\boldsymbol{\alpha}]\rangle \quad (44)$$

where γ is a parameter controlling the learning rate which we will have to tune to produce appropriately sized steps for our simple optimisation routine.

We now run this optimisation with just 10.000 MCMC cycles for 500 particles in three dimensions and an initial guess of $\alpha = 1.5$. A litte trial and error showed that 10^{-5}

was an appropriate learning rate, and the algorithm converged on $\alpha = 0.5$ (giving an energy of 750) in about ten iterations. Taking the derivative of the expected energy with regards to α shows that this is indeed a value for alpha that represents the minima of the energy.

$$0 = \frac{\partial}{\partial \alpha}\langle E_L^{\text{3D}}[\alpha]\rangle = \frac{\partial}{\partial \alpha}\frac{3N}{2}\left(\frac{1}{4\alpha} + \alpha\right) = 750\left(-\frac{1}{4\alpha^2} + 1\right)$$
$$\alpha = \sqrt{\frac{1}{4}} = \frac{1}{2}$$
$$(45)$$

8. Task e) A better statistical analysis

To present our results in a professional way, we would like to have a proper estimate for the statistical errors. The vaules we report for the energy after each run of our Monte Carlo routine (with n cycles) which we will call experiment α is the sample mean, given by

$$\bar{\mu}_\alpha = \frac{1}{n}\sum_{i=1}^n x_{\alpha i} \qquad (46)$$

where x_i denotes the value attained from the i-th cycle. Similarly, the variance we report is the sample variance for the run:

$$\bar{\sigma}_\alpha^2 = \frac{1}{n}\sum_{i=1}^n (x_{\alpha i} - \bar{\mu}_\alpha)^2 \qquad (47)$$

Generally, when performing experiments what we are really interested in is not the mean and variance we see in our particular run of experiments, but rather the true underlying value we are trying to ascertain and the variance inherent to our experiment as a way of producing estimates of this true value. Hence for a single experiment the sample values will generally not equal the true values, and we have

$$\bar{\mu}_\alpha \neq \mu = \int dx P(x) x \qquad (48)$$

$$\bar{\sigma}_\alpha^2 \neq \sigma^2 = \int dx P(x)(x - \mu)^2 \qquad (49)$$

If we now preform m such experiments and assume them to be independent and identically distributed, then by the central limit theorem the sample mean of our experimental values $z = \frac{1}{m}\sum_{\alpha=1}^m$ will follow a Gaussian distribution with mean μ and variance $\frac{\sigma^2}{m}$ in the limit of infinite experiments.

Thisgs do get a bit more complicated if our experimental values are not independent and identically distributed. We then have to take into account the covariance

$$COV(x_i, x_j) = \langle (x_i - \langle x_i\rangle)(x_j - \langle x_j\rangle)\rangle$$
$$= \int (x_i - \langle x_i\rangle)(x_j - \langle x_j\rangle)P(x_1, ..., x_n)dx_1...dx_n \qquad (50)$$

where the integral is over all possibile combinations of values for all the n variables in our experiment, and $P(x_1, ..., x_n)$ is the joint probability distribution over said variables. Obviously then $\langle x_i\rangle = \int x_i P(x_1, ..., x_n)dx_1...dx_n$

For $i = j$ we easily see that we get back the expression for the variance in variable x_i, while if $i \neq j$, we get

$$COV(x_i, x_j) = \langle (x_i - \langle x_i\rangle)(x_j - \langle x_j\rangle)\rangle$$
$$= \langle x_i x_j - \langle x_i\rangle x_j - x_i\langle x_j\rangle + \langle x_i\rangle\langle x_j\rangle\rangle \quad (51)$$
$$= \langle x_i x_j\rangle - \langle x_i\rangle\langle x_j\rangle$$

If x_i and x_j were independent variables, these terms would be equal and the covariance becomes zero, as it should.

If we now imagine we perform m experiments (labeled by α) ant take as our experimental result $\mu_m = \frac{1}{m}\sum_{\alpha=1}^m \bar{\mu}_\alpha$, what we will be interested in is the variance

$$\sigma_m^2 = \frac{1}{m}\sum_{\alpha=1}^m (\bar{\mu}_\alpha - \mu_m)^2$$
$$= \frac{1}{m}\sum_{\alpha=1}^m \left[\bar{\mu}_\alpha^2 - 2\bar{\mu}_\alpha\mu_m + \mu_m^2\right] = \frac{1}{m}\sum_{\alpha=1}^m \left[\bar{\mu}_\alpha^2 - \mu_m^2\right]$$
$$= \frac{1}{m}\sum_{\alpha=1}^m \left[\left(\frac{1}{n^2}\sum_{k=1}^n x_{\alpha k}\sum_{l=1}^n x_{\alpha l}\right) - \mu_m^2\right]$$
$$= \frac{1}{mn^2}\sum_{\alpha=1}^m\sum_{k=1}^n\sum_{l=1}^n (x_{\alpha k} - \mu_m)(x_{\alpha l} - \mu_m)$$
$$= \frac{1}{mn^2}\sum_{\alpha=1}^m \left[\sum_{k=1}^n (x_{\alpha k} - \mu_m)^2 + 2\sum_{k<l}^n (x_{\alpha k} - \mu_m)(x_{\alpha l} - \mu_m)\right]$$
$$= \frac{\sigma^2}{n} + \frac{2}{mn^2}\sum_{\alpha=1}^m\sum_{k<l}^n (x_{\alpha k} - \mu_m)(x_{\alpha l} - \mu_m)$$
$$(52)$$

where we have introduced the sample variance for all the single observations $x_{\alpha k}$ across our experiments

$$\sigma^2 = \frac{1}{mn}\sum_{\alpha=1}^m\sum_{k=1}^n (x_{\alpha k} - \mu_m)^2$$

and we can easily see that the second term in our expression for σ_m^2 is an expression of the correlation between our samples. We want to revrite this correlation term for conveniance by introducing the distance between two samples $d = |l - k|$, giving

$$\sum_{k<l}^n (x_{\alpha k} - \mu_m)(x_{\alpha l} - \mu_m)$$
$$= \sum_{d=1}^{n-1}\sum_{k=1}^{n-d} (x_{\alpha k} - \mu_m)(x_{\alpha(k+d)} - \mu_m) \qquad (53)$$

Then defining the function

$$f_d = \frac{1}{mn} \sum_{\alpha=1}^{m} \sum_{k=1}^{n-d} (x_{\alpha k} - \mu_m)(x_{\alpha(k+d)} - \mu_m) \qquad (54)$$

and seeing that $f_0 = \sigma^2$ allows us to rewrite the correlation term to get

$$\sigma_m^2 = \frac{\sigma^2}{n} + \frac{2}{n} \sum_{d=1}^{n-1} f_d = \frac{\sigma^2}{n} \left[1 + \frac{2}{\sigma^2} \sum_{d=1}^{n-1} f_d \right] = \frac{1}{n} \left[f_0 + 2 \sum_{d=1}^{n-1} f_d \right] \qquad (55)$$

Computing this expression quickly becomes costly due to the double sum involved in the correlation term. To avoid the $\mathcal{O}(n^2)$ scaling that double sum implies, we turn to the Blocking method by Flyvbjerg and Petersen[1] as an efficient way to estimate this quantity. The strategy is to halve the number of points in our data set $\{x_1, x_2, ..., x_n\}$ by introducing $\{x'_1, x'_2, ..., x'_{\frac{n}{2}}\}$ where $x'_i = \frac{1}{2}(x_{2i-1} + x_{2i})$.

This transformation obviously preserves the mean, i.e. $\mu'_m = \mu_m$ and Flyvbjerg and Petersen shows[1] that it preserves the variance as well $\sigma_m^2(\mu') = \sigma_m^2(\mu)$. Let's look at what happens to the f_ds at each blocking transformation, and consider that we are only performing this blocking operation on the samples from a single experiment. Specifically we want to look at $f_{\alpha 0} = \bar{\sigma}_\alpha^2$

$$\begin{aligned}
f'_{\alpha 0} &= \frac{1}{n'} \sum_{k=1}^{n'} (x'_k - \mu_m)^2 = \frac{2}{n} \sum_{k=1}^{n'} (x'^2_k - \mu_m^2) \\
&= \frac{2}{n} \sum_{k=1}^{n'} \left[\frac{1}{4}(x_{2k-1}^2 + 2x_{2k-1}x_{2k} + x_{2k}^2) \right] - \mu_m^2 \\
&= \frac{1}{2n} \sum_{k=1}^{n'} \left[x_{2k-1}^2 + x_{2k}^2 + 2x_{2k-1}x_{2k} \right] - \mu_m^2 \\
&= \frac{1}{2n} \sum_{k=1}^{n} (x_n^2 - \mu_m^2) + \frac{1}{n} \sum_{k=1}^{n'} (x_{2k-1}x_{2k}) - \frac{\mu_m^2}{2} \\
&= \frac{1}{2} f_{\alpha 0} + \frac{1}{n} \sum_{k=1}^{n'} \Big[(x_{2k-1} - \mu_m)(x_{2k} - \mu_m) \\
&\quad + x_{2k-1}\mu_m + x_{2k}\mu_m - \mu^2 \Big] - \frac{\mu_m^2}{2} \\
&= \frac{1}{2} f_{\alpha 0} + \frac{1}{n} \sum_{k=1}^{n'} (x_{2k-1} - \mu_m)(x_{2k} - \mu_m) \\
&\quad + \frac{1}{n} \sum_{k=1}^{n'} (x_{2k-1}\mu_m + x_{2k}\mu_m) - \mu_m^2 \\
&= \frac{1}{2} f_{\alpha 0} + \frac{1}{n} \sum_{k=1}^{n-1} (x_k - \mu_m)(x_{k+1} - \mu_m) \\
&\quad + \frac{1}{n} \sum_{k=1}^{n} (x_k \mu_m) - \mu_m^2 \\
&= \frac{1}{2} f_{\alpha 0} + \frac{1}{2} f_{\alpha 1} + 0
\end{aligned} \qquad (56)$$

If we imagine repeating this process recursively, we see that in each step, the $\frac{f_0^{(k)}}{n^{(k)}}$ term absorbs part of the correlation term. Flyvbjerg and Petersen showed that the correlation term is indeed decreasing under this process[1], and considering that $\sigma_m^2(\mu') = \sigma_m^2(\mu)$ we see that the first term will become a better and better approximation with each blocking transformation.

We now pick up where we left off in task d), and initiate a run of 2^{20} MCMC cycles for the optimal alpha value found by our gradient descent routine. Taking advantage of the code provided in the course material for performing a statistical analysis with the Blocking method, we analyse the list of over a million samples for the energy in 1.5 seconds. The resulting estimate for the ground state energy was

$$E_{\text{Ground state}} = 750.00026 \pm 0.000133$$

9. Task f) Parallelizing the code

I did not find the time to complete this task, and hence have not taken any structural measures to enhance performance beyond having the core functions called by the MCMC routine compiled with numba.

10. Task g) The repulsive interaction

After testing our code thoroughly on the analytically solvable problem posed by the non-interacting case, we now want to leverage it for the interacting case with possibly elliptical potentials. The dimensionless Hamiltonian (25) is then

$$
H = \frac{1}{2} \sum_i \left(-\nabla_i^2 + \boldsymbol{\gamma}_i^2 \right) + \sum_{i<j} v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)
$$
$$
= \sum_i \frac{1}{2} \left(-\nabla_i^2 + [x_i^2 + y_i^2 + \gamma^2 z_i^2] \right) + \sum_{i<j} v_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)
$$
(57)

and we are not able to simplify the local energy (26) as well as we did for the shperical potential non-interacting case.

We are going to have to calculate the derivatives of $u(r_{ij}) = \ln f(a, \boldsymbol{r}_i, \boldsymbol{r}_j) = \ln f(a, r_{ij})$ where

$$
f(a, r_{ij}) = \begin{cases} 0 & r_{ij} \leq a \\ \left(1 - \frac{a}{r_{ij}}\right) & r_{ij} > a \end{cases}
$$
(58)

and hence the derivatives with regards to r_{ij} are

$$
u'(r_{ij}) = \frac{f'(a, r_{ij})}{f(a, r_{ij})}
$$
(59)

$$
u''(r_{ij}) = \frac{f''(a, r_{ij})}{f(a, r_{ij})} - \frac{f'(a, r_{ij})^2}{f(a, r_{ij})^2}
$$
(60)

Inserting the expressions for the derivatives of $f(a, r_{ij})$

$$
f'(a, r_{ij}) = \begin{cases} 0 & r_{ij} \leq a \\ \frac{a}{r_{ij}^2} & r_{ij} > a \end{cases}
$$
(61)

$$
f''(a, r_{ij}) = \begin{cases} 0 & r_{ij} \leq a \\ \frac{-2a}{r_{ij}^3} & r_{ij} > a \end{cases} = \frac{-2}{r_{ij}} f'(a, r_{ij})
$$
(62)

we get

$$
u'(r_{ij}) = \frac{f'(a, r_{ij})}{f(a, r_{ij})} = \begin{cases} 0 & r_{ij} \leq a \\ \frac{1}{r_{ij}} \frac{a}{r_{ij}-a} & r_{ij} > a \end{cases}
$$
(63)

and

$$
u''(r_{ij}) = u'(r_{ij}) \left(\frac{-2}{r_{ij}} - u'(r_{ij}) \right)
$$
$$
= \begin{cases} 0 & r_{ij} \leq a \\ \frac{-2}{r_{ij}^2} \frac{a}{r_{ij}-a} - \frac{1}{r_{ij}^2} \frac{a^2}{(r_{ij}-a)^2} & r_{ij} > a \end{cases}
$$
(64)

These results allow us some simplifications of the individual terms in the local energy:

$$
\left[\frac{u'(r_{ij})}{r_{ij}} + \frac{u''(r_{ij})}{2} \right] = \left[\frac{u'(r_{ij})}{r_{ij}} - \frac{u'(r_{ij})}{r_{ij}} - \frac{u'(r_{ij})^2}{2} \right]
$$
$$
= \frac{-u'(r_{ij})^2}{2} = \begin{cases} 0 & r_{ij} \leq a \\ \frac{-1}{2r_{ij}^2} \frac{a^2}{(r_{ij}-a)^2} & r_{ij} > a \end{cases}
$$
(65)

Noting that the interaction term will make the energy infinte if any particle is closer than a to any other particle, and considering the three dimensional case, we can write for the contribution from one particle in the non-infinite case

$$
E_{L\,i}^{3D}(\boldsymbol{r}) = \alpha(2+\beta) - 2\alpha^2(x_i^2 + y_i^2 + \beta^2 z_i^2) + \frac{x_i^2 + y_i^2 + \gamma^2 z_i^2}{2}
$$
$$
+ \sum_{j\neq i} \left[\frac{u'(r_{ij})^2}{2} + 2\alpha\boldsymbol{\beta}_i \cdot \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{r_{ij}} u'(r_{ij}) \right]
$$
$$
- \sum_{j\neq i} \sum_{j\neq k} \frac{1}{2} \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{r_{ij}} \cdot \frac{\boldsymbol{r}_i - \boldsymbol{r}_k}{r_{ik}} u'(r_{ij}) u'(r_{ik})
$$
(66)

$$
F_i(\boldsymbol{r}) = 2 \sum_{j\neq i} \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{r_{ij}} u'(r_{ij}) - 4\alpha\boldsymbol{\beta}_i
$$

To be able to run our optimisation to find good variational parameters, we also need to calculate the gradient term

$$
\frac{\nabla_\alpha \Psi_T}{\Psi_T} = \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha}
$$
$$
= \frac{1}{\Psi_T} \frac{\partial}{\partial \alpha} \exp\left(-\alpha \sum_i [x_i^2 + y_i^2 + \beta z_i^2] + \sum_{i<j} u(r_{ij}) \right)
$$
$$
= -\sum_i [x_i^2 + y_i^2 + \beta z_i^2] + \sum_{i<j} \frac{\partial}{\partial \alpha} u(r_{ij})
$$
$$
= -\sum_i [x_i^2 + y_i^2 + \beta z_i^2]
$$
(67)

We now have all the ingredients to run our full simulation, an would like to test it by trying to reproduce the results from the non-interacting case. Setting $\beta = \gamma = 1$ and turning interactions off by disregarding the interaction terms of our various calculated quantities, we run a gradient descent with 10.000 MCMC cycles each step for 10, 50 and 100 particles. We see that the optimisation converges on $\alpha = 0.5$ as expected, and an energy of roughly 15, 75 and 150.

Now we first introduce the elliptical potential by setting $\beta = \gamma = 2.8284$ and repeat the process. We get the same value for α, but the energies change to 24.14, 120.71 and 241.42 giving very nearly an exact increase of 60.9% for all three or an energy of $2.414\hbar\omega$ per particle. Intuitively, it makes sense that increasing the strength of the trapping potential would raise the energy of the ground state.

As we are still in the non-interacting case, it is ressuring to see the same increase across different particle number (since the system is essentially many one-particle systems). This also corresponds very well to the results in [2] for the interacting case in the limit of decreasing particle number to one and hence reproducing a non-interacting case (at least the correspondence is as good as the precistion with which we are able to read out the results from Fig. 2 in [2]).

Finally we want to turn on interactions an repeat the process for the full system. Here I ran into some issues with my solution. For some reason, my performance optimisation of calculating the local energy for the initial position and then just adjusting it with the effect of the single particle move made each cycle failed, with the adjustments resulting in a steady net increase of the local energy value over many samples. I was unable to figure out the root cause of this issue, but reverting to calculating the local energy by a loop over all single particle contributions each cycle produced converging energy estimates and hence convergence in the variational optimisation (although there still seemed to be some instability that made the optimisation flutter around a final value rather than just converging cleanly as it did in the non-interacting case). In combination with the fact that I was not able to find the time to complete the parallellisation task, this resulted in brutally slow calculations.

With the 100 particle calculation having to run overnight, I finally got some values for the α and energy:

$\alpha_{10} = 0.500716$ and $E_{\text{Ground state}} = 24.14098 \pm 0.001001$

$\alpha_{50} = 0.50116$ and $E_{\text{Ground state}} = 120.621827 \pm 0.015730$

$\alpha_{100} = 0.48973$ and $E_{\text{Ground state}} = 241.372572 \pm 0.041187$

for 10, 50 and 100 particles respectively. We should be very skeptical that these results are practically the same as for the non-interacting case. From [2] we should expect an increase in energy per particle, which we don't see at all. It also seems dubious that we added a repulsive interaction and got practically the same value for α which encodes the spatial spread of the single particle components of the wavefunction. We would expect the energy to increase and α to decrease under the effect of turning on the interaction and with increasing particle number. From an investigation of the same system by [3], we can see that the obtained values for $N = 500$ particles was $\alpha = 0.475$ and $\frac{E}{N} = 3.3242\hbar\omega$.

These considerations make me suspect that there is some mistake in the interaction part of the local energy function, though I am not sure whether it is a coding mistake or a faulty analytical derivation. From investigating which term caused the steady increase of the local enegy value described above, my main suspicion is that the culprit is the second single sum term (i.e. the one involving β_i). I don't have time to investigate further before the project

deadline, and will probably be pretty annoyed with myself when I figure it out at some later point.

11. Task h) Onebody densities

Since we obviously have no way of visualising the whole wavefunction of such a multi-dimensional system, we would like to come up with something low dimensional that can be visualised and does tell us something interesting about the system.

Our idea will be to look to the wavefunction of a single particle, keeping in mind that the particles are indistinguishable and hence what we learn from one particle is very relevant for the system as a whole. An obvious choice is to extract the probability density function for a single particle. We can do this by integrating the squared wavefunction over all positions except the position of the particle in question.

Disregarding the normalisation for the same reasons as above, we have

$$
\begin{aligned}
\rho(\boldsymbol{r}_1) &= \int d\boldsymbol{r}_2...d\boldsymbol{r}_N \Psi_T^*(\boldsymbol{r},\boldsymbol{\alpha})\Psi_T(\boldsymbol{r},\boldsymbol{\alpha}) \\
&= \int d\boldsymbol{r}_2...d\boldsymbol{r}_N |\Psi_T(\boldsymbol{r},\boldsymbol{\alpha})|^2 \\
&= \int d\boldsymbol{r}_1' d\boldsymbol{r}_2...d\boldsymbol{r}_N P(\boldsymbol{r}_1'\boldsymbol{r}_2...d\boldsymbol{r}_N,\boldsymbol{\alpha})\delta(\boldsymbol{r}_1 - \boldsymbol{r}_1') \\
&\approx \frac{1}{N}\sum_{k=1}^{N}\delta(\boldsymbol{r}_1 - \boldsymbol{r}_1'^{\,k}) \qquad \boldsymbol{r}^k \sim P(\boldsymbol{r},\boldsymbol{\alpha})
\end{aligned}
\tag{67}
$$

Where we have used an upper script k to denote the position sampled in the kth run of an MCMC sampler. In general, we would not expect to be able to solve this integral analytically, though in the non-interacting case a look at the wavefunction should allow us to reason that the result would be a Gaussian. As the last line indicates we can simply use our MCMC routine to perform this integral: simply record how many steps we find the particle at r_1 and divide it by the total number of steps to get the probability density. Obviously as we only have finite sampling steps, we need to discretisise our space and count each time particle 1 is in each spatial bin.

However, since we might want to experiment with spatial resolutions for the bins we just run the MCMC sampling and record the position of particle 1 in each cycle. Then we could plot a simple normalised histogram of positions in whatever discretisized spatial bins we want without rerunning the calculations. Actually, since the particles are indistinguishable, we might as well record the position of all the particles each cycle, to get more data and hence smaller statistical deviations in the aggregate. Below is one such plot for 10 particles in the non-interacting case, with spherical potential in blue and elliptical potential in red.

Simulated one-body density

Simulated one-body density for 100 particles
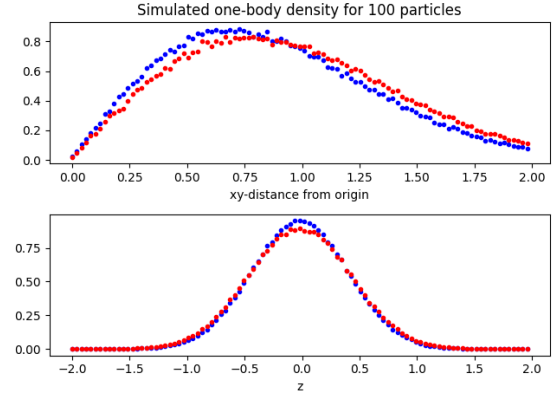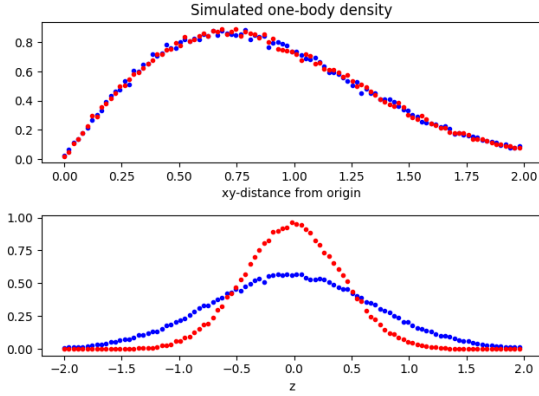
We see that the distribution in the xy-plane is identical for the two potentials, while increasing the strength of the z-component squeezes the distribution of particles closer to the z-axis. This is pretty much what we would expect, and the simulation traces out rather nice Gaussians.

We plot the same figures for our interacting (blue) and non-interacting cases below, for 10, 50 and 100 particles respectively:

Simulated one-body density for 10 particles

Simulated one-body density for 50 particles
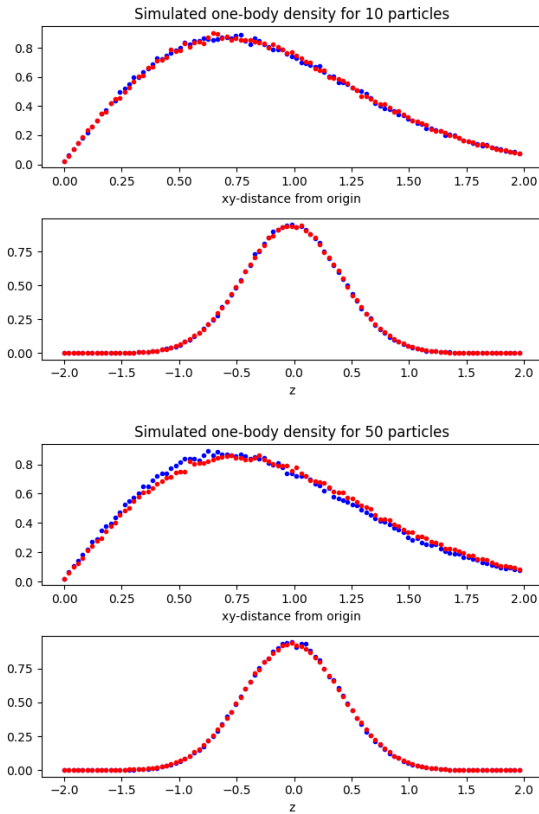
we see that as the values suggested there is almost no difference introduced by the interaction. The exception is the 100 particle case where we see a slight move of the density away from the origin, as we would expect with the slightly lower value for α. However, as we see a hint of the same trend for the 50 particle plot (for which $\alpha > 0.5$ in the interacting case) this could also - at least in part - be an effect of the interaction term in the wavefunction itself.

It would be interesting to see if this trend would be more pronounced at higher particle numbers, but as our code in its currents state is not practically capable of simulating with more particles this has not been tested.

12. Closing remarks

Although I fatally misjudged the ammount of work in the last tasks and was left with an incomplete simulation despite a week of working every available hour, I just have to say that I loved this project! This really has been an excellent way to learn, especially considering that I have not been able to follow the lectutres or cumputer labs live. The only point of improvement that I can imagine is that some of the project description could have been a bit more clearly explained. Especially the descriptions of each task sometimes left we wondering if I understood exactly what was meant, and had me bother Morten with a few questions over mail. This could very possibly have been an issue only because I have not been able to attend the computer labs where I imagine the tasks are discussed in more detail.

All said, I am looking forward to the next project!

Referanser

[1] Flyvbjerg and Petersen, *Error estimates on averages of correlated data*, The Journal of Chemical Physics, (1989).

[2] J. L. DuBois and H. R. Glyde, H. R., *Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis*, Phys. Rev. A 63, 023602 (2001)

[3] J. K. Nilsen, J. Mur-Petit, M. Guilleumas, M. Hjorth-Jensen, and A. Polls, *Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region*, Phys. Rev. A 71, 053610 (2005)