FYS4411 - Computational Physics II

Project 1

Dorthea Gjestvang
Even Marius Nordhagen

March 14, 2018

Abstract

Write the abstract here

- Github repository containing programs and results are in: `https://github.com/evenmn/FYS4411/tree/master/Project%201`

1 Introduction

Introduction

2 Theory

We study a system of N bosons trapped in a harmonic oscillator with the Hamiltonian given by

$$\hat{H} = \sum_i^N \bigg(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\vec{r}_i)\bigg) + \sum_{i<j}^N V_{int}(\vec{r}_i, \vec{r}_j) \tag{1}$$

with V_{ext} as the external potential, which is the harmonic oscillator potential, and V_{int} as the interaction term. The interaction will in the first place be ignored, and is specified later. We will consider a harmonic oscillator which can either be spherical (all dimensions have the same scales) or elliptical (the vertical dimension has a different frequency from the horizontals),

$$V_{ext}(\vec{r}) = \begin{cases} \frac{1}{2}m\omega_{HO}^2\vec{r}^2 & \text{(Spherical)} \\ \frac{1}{2}m[\omega_{HO}^2(x^2+y^2) + \omega_z^2 z^2] & \text{(Elliptical).} \end{cases} \tag{2}$$

The trial wavefunction is on the form

$$\Psi_T(\vec{r}_1, \vec{r}_2, ..., \vec{r}_N, \alpha, \beta) = \prod_i^N g(\alpha, \beta, \vec{r}_i) \prod_{i<j} f(a, r_{ij}) \tag{3}$$

where $r_{ij} = |\vec{r}_i - \vec{r}_j|$ and g is assumed to be an exponential function,

$$g(\alpha, \beta, \vec{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)], \tag{4}$$

which is practical since

$$\prod_i^N g(\alpha, \beta, \vec{r}_i) = \exp\Big[-\alpha \sum_{i=1}^N (x_i^2 + y_i^2 + \beta z_i^2)\Big]. \tag{5}$$

α is a variational parameter that we later use to find the energy minimum, and β is a constant. This is also the form of the exact ground state wave function for a harmonic oscillator, so by choosing the correct α, we will find the exact ground state energy (when V_{int} is ignored). The f presented above is the correlation wave function, which is

$$f(a, r_{ij}) = \begin{cases} 0 & r_{ij} \leq a \\ \left(1 - \frac{a}{r_{ij}}\right) & r_{ij} > a. \end{cases} \tag{6}$$

The first case we will take into account is when $a = 0$, such that the corretation term is 1.

We want to calculate the local energy as a function of α, and then use Variational Monte Carlo (VMC) described in section 3.1. For the non-interacting case, the analytical expression for one particle in a harmonic oscillator is well-known and reads $E = \hbar\omega(n + dim/2)$ where n is the energy level and dim is number of dimensions. In this project we will study the ground state only, such that $n = 0$, and for N particles and dim free dimensions we therefore obtain

$$E = \frac{1}{2} N \cdot dim \cdot \hbar\omega_{HO}. \tag{7}$$

We will focus on the the local energy, which follows from a remodeling of Schrödinger's equation

$$E_L(\vec{r}) = \frac{1}{\Psi_T(\vec{r})} \hat{H} \Psi_T(\vec{r}). \tag{8}$$

When the repulsive interaction is ignored ($a = 0$), it can be shown that the local energy for a system of N particles and dim free dimensions is given by

$$E_L = dim \cdot N \cdot \alpha + \left(\frac{1}{2} - 2\alpha^2\right) \sum_i \vec{r}_i^2, \tag{9}$$

which is actually proven in Appendix A.

For $a \neq 0$ it gets rather more complicated, because we need to deal with the correlation term as well. By defining

$$f(a, r_{ij}) = \exp\left(\sum_{i<j} u(r_{ij})\right) \tag{10}$$

and doing a change of variables

$$\frac{\partial}{\partial \vec{r}_k} = \frac{\partial}{\partial \vec{r}_k}\frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \vec{r}_k}\frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}}\frac{\partial}{\partial r_{kj}} \tag{11}$$

one will end up with

$$\begin{aligned} E_L = \sum_k \Bigg(-\frac{1}{2}\Bigg(4\alpha^2\Big(x_k^2 + y_k^2 + \beta^2 z_k^2 - \frac{1}{\alpha} - \frac{\beta}{2\alpha}\Big) \\ - 4\alpha \sum_{j\neq k} (x_k, y_k, \beta z_k)\frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}} u'(r_{kj}) \\ + \sum_{ij\neq k} \frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_i)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \\ + \sum_{j\neq k}\Big(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj})\Big)\Bigg) + V_{ext}(\vec{r}_k)\Bigg). \end{aligned} \tag{12}$$

This is not a pretty expression, but hopefully it will give us the correct answers. We could also split up the local energy expression

$$E_{L,i} = -\frac{\hbar^2}{2m}\frac{\nabla_i^2 \Psi_T}{\Psi_T} + V_{ext}(\vec{r}_i) = E_{k,i} + E_{p,i} \tag{13}$$

and calculate the local energy with a numerical approach where the second derivative can be approximated by the three-point formula:

$$f''(x) \simeq \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \tag{14}$$

In our case the position is a three dimensional vector, so we need to handle each dimension separately. Both the analytical and the numerical local energy are implemented, and in section 4.1, the CPU time for the analytical and numerical approach are compared for a various number of particles.

The most interesting and realistic case is when the interaction is included, so let us now set it to a so-called hard-sphere potential

$$V_{int}(\vec{r}_i, \vec{r}_j) = \begin{cases} 0 & \text{if} \quad |\vec{r}_i - \vec{r}_j| \geq a \\ \infty & \text{if} \quad |\vec{r}_i - \vec{r}_j| < a. \end{cases} \tag{15}$$

which ensures that the particles are separated by a distance a. The local energy is slightly changing, because we need to add the interaction potential to it as well.

2.1 One-body density

In many cases it is convenient to know the positions of the particles, but when the number of particles increases, the set of positions turns into a messy collection of numbers which is not really informative. Instead of presenting the positions, the density of particles can give us a good overview of where the particles can be found. With N particles, the one-body density with respect to a particle i is an integral over all particles but particle i

$$\rho_i = \int_{-\infty}^{\infty} d\vec{r}_1 \dots d\vec{r}_{i-1} d\vec{r}_{i+1} \dots d\vec{r}_N |\Psi(\vec{r}_1, \dots \vec{r}_N)|^2. \tag{16}$$

For the non-interacting case this integral can be solved analytically, or we can use Monte Carlo integration to solve it for any case. Anyhow, the interesting part is the radial density, so we either have to solve the integral in spherical coordinates or convert to spherical coordinates afterwards.

Alternatively, the one-body radial density can be found in a more intuitive way. Imagine we divide the volume around particle i into bins, where bin j is located at a distance $j \cdot r_1$ (the radii are quantized). By counting the number of particles in a bin and dividing on the surface area, we find the average density of the bin. If we further decrease the initial radius r_1 (radius of the innermost bin) such that we have a large number of bins, this method can be used to find the one-body density.

2.2 Scaling

For big numerical projects, working with dimensionless quantities is a big advantage. Not only does it improve the code structure and performance, but it also avoids truncation errors due to small constants. For this project a natural scaling parameter for the energy is $\hbar\omega_{HO}$, which appears in the analytical energy expression in equation (7). The equivalent dimensionless equation can then be written as

$$E' = \frac{N \cdot dim}{2} \tag{17}$$

where $E' = E/\hbar\omega_{HO}$. Additionally we can scale the position with respect to the length of the spherical trap, a_{HO}, such that

$$r_i' = \frac{r_i}{a_{HO}} = r_i \cdot \sqrt{\frac{m\omega_{HO}}{\hbar}}, \tag{18}$$

and the Hamiltonian turns into

$$H = \frac{1}{2}\sum_i \Big(-\nabla^2 + \vec{r}_i^2 \Big) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{19}$$

A watchful eye will see that this corresponds to setting $\hbar = \omega_{HO} = m = 1$, which is the natural units.

For the spherical trap situation we are left with the variational parameters α and β only, but when we study an elliptical trap we still want to get rid of ω_z. Since β^2 should be the factor in front of the z-coordinate when the Hamiltonian is dimensionless, it can be proven that $\beta = \omega_z/\omega_{HO}$, see Appendix B. We end up with the Hamiltonian

$$H = \sum_i \left(\frac{1}{2}\Big(-\nabla^2 + x_i^2 + y_i^2 + \beta^2 z_i^2 \Big) \right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{20}$$

where β is chosen to be 2.82843 due to experimental results.

2.3 Error estimation

When presenting data from an experiment, one should always know the errors in the answer. Experimental data, including data from numerical experiments, are never determined beyond any doubt, and an estimate of this error should therefore be presented alongside the data. There are two kinds of errors. Statistical errors originate from how much statistics one has; when 10^6 measured points give approximately the same answer, one can be more sure that the actual value is close to those points, more than if one only has 1 point of statistical data. Estimating the statistical error is easily done. The systematic error, however, is harder to handle. It arises for example from calculations being based on faulty theory, or defect measurement devices. Here, we will present how to get an estimate of the statistical error in a numerical experiment.

When conducting an experiment α with n measured points,x_n, the sample mean of the experiment $\langle x \rangle$ is defined as shown in equation 21.

$$\langle x_\alpha \rangle = \frac{1}{n}\sum_{k=1}^{n} x_{\alpha,k} \tag{21}$$

The corresonding sample variance σ_α is then defined as

$$\sigma_\alpha^2 = \frac{1}{n}\sum_{k=1}^{n} (x_{\alpha,k} - \langle x_\alpha \rangle)^2 \tag{22}$$

This gives us the error in the given experiment α. If we repeat this experiment m times, the mean after all the experiments are

$$\langle x_m \rangle = \frac{1}{n} \sum_{k=1}^{n} \langle x_\alpha \rangle \tag{23}$$

The total variance is then

$$\sigma_m^2 = \frac{1}{m} \sum_{\alpha=1}^{m} (\langle x_\alpha \rangle - \langle x_m \rangle)^2 \tag{24}$$

This can be reduced to

$$\sigma_m^2 = \frac{\sigma^2}{n} + covarianceterm \tag{25}$$

where σ is the sample variance over all the experiments, defined as

$$\sigma^2 = \frac{1}{mn} \sum_{\alpha=1}^{m} \sum_{k=1}^{n} (x_{\alpha,k} - \langle x_m \rangle)^2 \tag{26}$$

and the covariance is the linear correlation between the measured points. The definition of the covariance is shown in equation **??**.

$$cov(x, y) = \frac{1}{n^2} \sum_{i} \sum_{j>i} (x_i - x_j)(y_i - y_j) \tag{27}$$

A common simplification is to reduce equation 25 to the following:

$$\sigma^2 \approx \langle x^2 \rangle - \langle x \rangle^2 \tag{28}$$

This equation, however, does not take into account the covariance term from equation 25, and as the covariance term is added to the expression for the variance, 28 will underestimate the uncertainty σ for positive covariances.

A direct implementation of equation 25 including the covariance term is not suitable, as the expresion for the covariance includes a double sum, and for a large number of iterations, this will turn into an extremely time-consuming process for a large number of Monte Carlo iterations. Luckily, there are methods for calculating an accurate estimation of the variance without including a double loop in the Monte Carlo program. One of these methods is the blocking method, which is presented in section 3.4.

3 Methods

3.1 Variational Monte Carlo

Variational Monte Carlo (VMC) is a widely used method for approximating the ground state of a quantum system. The method is based on Markov chains, and move a particle (or a set of particles) one step for each cycle, i.e.

$$\vec{R}_{new} = \vec{R} + r \cdot \text{step}. \tag{29}$$

Both the direction and the change in position are randomly chosen, so with a plain VMC implementation the particles will move randomly and independently of each other. We are going to use the Metropolis algorithm in addition to the VMC, which accepts or rejects moves based on the probability ratio between the old and the new position. This makes the system approach the most likely state, and the idea is that after a certain number of cycles the system will be in the most likely state.

3.2 Metropolis Algorithm

As mentioned above the task of the Metropolis algorithm is to move the system against the most likely state. The standard algorithm, here named brute force, is the simplest one, and does not deal with the transition probabilities. The modified Metropolis-Hastings algorithm includes, on the other hand, the transition probabilities and will be slightly more time consuming per cycle. We expect the latter to converge faster to the most likely state.

The foundation of the Metropolis algorithm is that the probability for a system to undergo a transition from state i to state j is given by the transition probability multiplied by the acceptance probability

$$W_{i\to j} = T_{i\to j} \cdot A_{i\to j} \tag{30}$$

where $T_{i\to j}$ is the transition probability and $A_{i\to j}$ is the acceptance probability. Built on this, the probability for being in a state i at time (step) n is

$$P_i^{(n)} = \sum_j \left[P_j^{(n-1)} T_{j\to i} A_{j\to i} + P_i^{(n-1)} T_{i\to j}(1 - A_{i\to j}) \right] \tag{31}$$

since this can happen in two different ways. One can start in this state i at time $n-1$ and be rejected or one can start in another state j at time $n-1$ and complete an accepted move to state i. In fact $\sum_j T_{i\to j} = 1$, so we can rewrite this as

$$P_i^{(n)} = P_i^{(n-1)} + \sum_j \left[P_j^{(n-1)} T_{j\to i} A_{j\to i} - P_i^{(n-1)} T_{i\to j} A_{i\to j} \right]. \tag{32}$$

When the times goes to infinity, the system will approach the most likely state and we will have $P_i^{(n)} = p_i$, which requires

$$\sum_j \left[p_j T_{j\to i} A_{j\to i} - p_i T_{i\to j} A_{i\to j} \right] = 0. \tag{33}$$

Rearranging, we obtain a quite useful result

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i T_{i\to j}}{p_j T_{j\to i}} \tag{34}$$

3.2.1 Brute force

In the brute force Metropolis algorithm we want to check if the new position is more likely than the current position, and for that we calculate the probabilities $P(\vec{R}) = |\Psi_T(\vec{R})|^2$ for both positions. We get rid off the transition probabilities setting $T_{i\to j} = T_{j\to i}$, and then end up with the plain ratio

$$w = \frac{P(\vec{R}_{new})}{P(\vec{R})} = \frac{|\Psi_T(\vec{R}_{new})|^2}{|\Psi_T(\vec{R})|^2}. \tag{35}$$

w will be larger than one if the new position is more likely than the current, and smaller than one if the current position is more likely than the new one. Metropolis handle this by accepting if the ratio w is larger than a random number r in the interval $[0, 1]$, and rejecting if not:

$$\text{New position:} \begin{cases} \text{accept} & \text{if} \quad w > r \\ \text{reject} & \text{if} \quad w \leq r. \end{cases} \tag{36}$$

3.2.2 Importance sampling

The importance sampling technique is often refered to as Metropolis-Hastings algorithm. The approach is the same as for the brute force Metropolis algorithm, but we will end ut with a slightly more complicated acceptation criteria. To understand the details, we need to begin with the Fokker-Planck equation, which describes the time-evolution of the probability density function $P(R, t)$. In one dimension it reads

$$\frac{\partial P(R,t)}{\partial t} = D \frac{\partial}{\partial R} \left(\frac{\partial}{\partial R} - F \right) P(R,t). \tag{37}$$

where F is the drift force and D is the diffusion coefficient. Even though the probability density function can give a lot of useful information, an equation describing the motion of a such particle would be more appropriate for our purposes. Fortunately this equation exists, and satisfies the Fokker-Planck equation. The Langevin equation can be written as

$$\frac{\partial R(t)}{\partial t} = DF(R(t)) + \eta \tag{38}$$

where η can be considered as a random variable. This differential equation can be solved by applying the forward Euler method and introducing gaussian variables ξ

$$R_{new} = R + DF(R)\Delta t + \xi\sqrt{\Delta t} \tag{39}$$

which will be used to update the position. Moreover we also need to update the acceptance criteria since we no longer ignore the transition probabilities. With the Fokker-Planck equation as base, the transition probabilities are given by Green's function

$$\begin{aligned} T_{R\to R_{new}} &= G(R_{new}, R, \Delta t) \\ &= \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp[-(R_{new} - R - D\Delta t F(R))^2/4D\Delta t] \end{aligned} \tag{40}$$

and the acceptance criteria becomes

$$r < \frac{G(R, R_{new}, \Delta t)|\Psi_T(R_{new})|^2}{G(R_{new}, R, \Delta t)|\Psi_T(R)|^2}. \tag{41}$$

3.3 Minimization methods

When the interaction term is excluded, we know which α that corresponds to the energy minima, and it is in principle no need to try different α's. However, sometimes we have no idea where to search for the minimum point, and we need to try various α values to determine the lowest energy. If we do not know where to start searching, this can be a time consuming activity. Would it not be nice if the program could do this for us?

In fact there are multiple techniques for doing this, where the most complicated ones obviously also are the best. Anyway, in this project we will have good initial guesses, and are therefore not in need for the most fancy algorithms.

3.3.1 Gradient descent

Perhaps the simplest and most intuitive method for finding the minima is the gradient descent method, which reads

$$\alpha^{+} = \alpha - \eta \cdot \frac{d\langle E_L(\alpha)\rangle}{d\alpha}. \tag{42}$$

where α^{+} is the updated α and η is the step size. The idea is that one finds the gradient of the energy with respect to a certain α, and moves in the direction which minimizes the energy. This is repeated until one has found an energy minimum.

3.4 Blocking method

4 Results

4.1 E_L calculation and CPU-time

For the brute force Metropolis algorithm we developed both an analytical and a numerical method to calculate the local energy. In table (1) we present the results from these calculations and the performance. The results from the calculations with the Metropolis-Hastings algorithm are presentet in table 2. All the measurements are done in three dimensions with $1e6$ Monte Carlo cycles. a is fixed to zero.

Table 1: The local energy calculated for $a = 0$ without hard-sphere interaction with brute force Metropolis algorithm, for both the analytical and numerical local energy calculation. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the local energies.

	Analytical		**Numerical**	
N	$\langle E_L\rangle$ [$\hbar\omega_{HO}$]	CPU-time [s]	$\langle E_L\rangle$ [$\hbar\omega_{HO}$]	CPU-time [s]
1	1.50000	0.15420	1.49999	0.65714
10	15.000	0.54785	14.9999	9.9844
100	150.00	13.573	149.999	2743.4
500	750.00	282.79	749.996	2.8744e5

Table 2: The local energy calculated for $a = 0$ without hard-sphere interaction with the Metropolis-Hastings algorithm with analytical local energy calculation. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the local energies.

	Analytical	
N	$\langle E_L \rangle$ [$\hbar\omega_{HO}$]	CPU-time [s]
1	1.5000	0.26693
10	15.000	0.54930
100	150.00	16.117
500	750.00	292.10

4.2 E_L as function of the variational parameter α

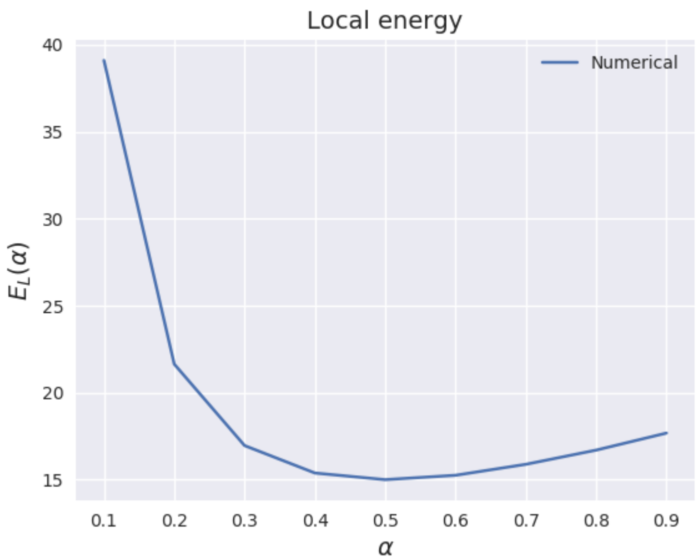

Figure 1: The local energy E_L calculated with the brute force Metropolis algorithm, as a function of the variational parameter α

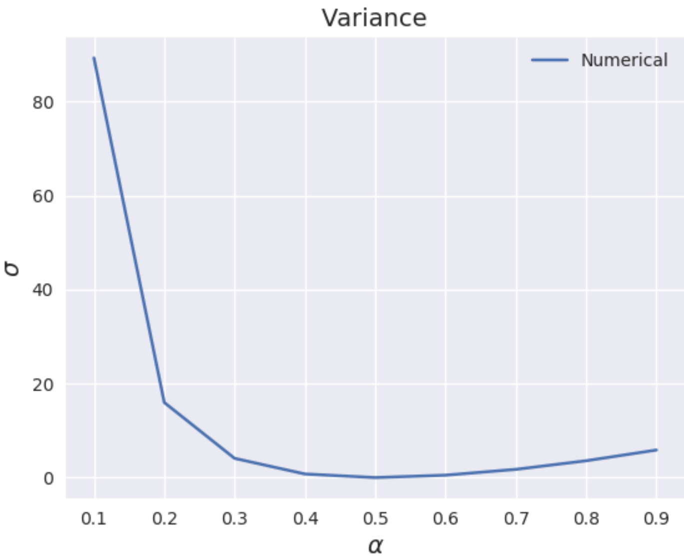

Figure 2: The variance of the local energy E_L calculated with the brute force Metropolis algorithm, as a function of the variational parameter α. CALCULATED with imprecise var

4.3 Acceptance ratios

We study the acceptance ratio for the brute force and importance sampling algorithms, with no interaction. These calculations are done with ten particles in three dimentions. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the variational parameter α is equal to 0.5.

4.3.1 Importance sampling dependence on timestep

The acceptance ratio for the importance sampling algorithm as a function of the timestep δt is shown in figure 3.

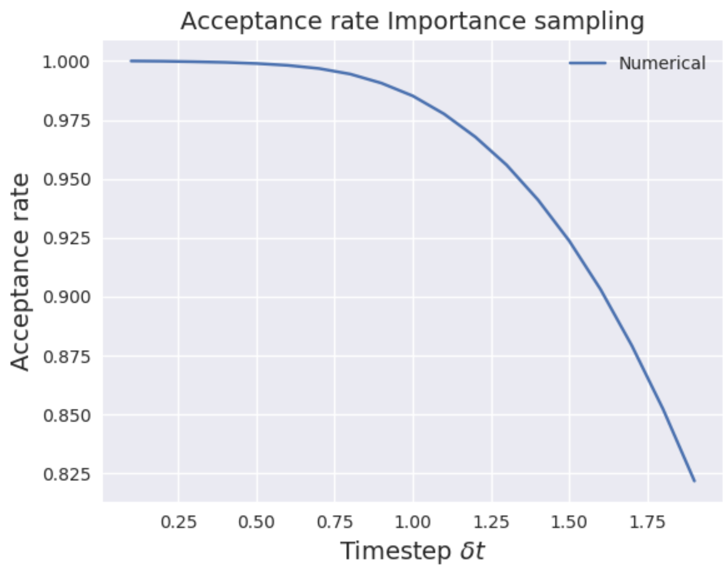

Figure 3: Acceptance ratio for different choices of the timestep δt with the Metropolis-Hastings algorithm.

4.3.2 Brute force dependence on stepsize

The acceptance ratio for the brute force algorithm as a function of the stepsize is shown in figure 4.

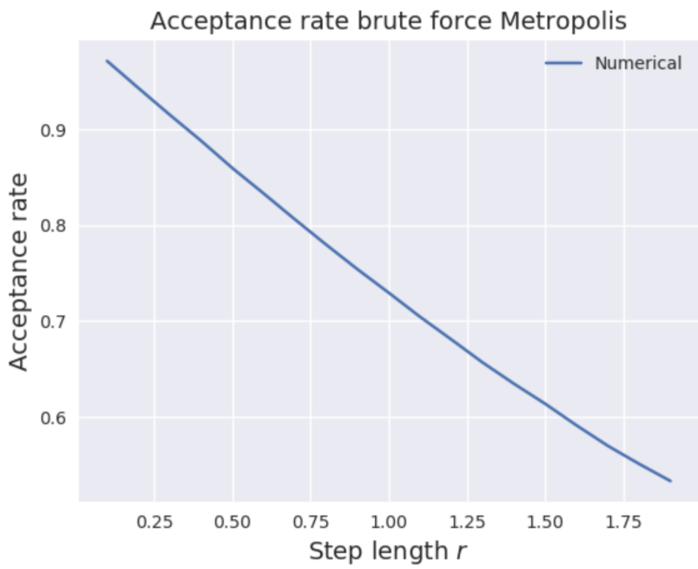

Figure 4: Acceptance ratio for different choices of the stepsize used in the brute force algorithm

4.4 Variance calculation

In table 3, we present the results of the local energy calculation with proper error evaluation of statistical error. The statistical error as found with the Blocking method is also compared to the simple way of approximating the variance by $\sigma^2 \approx \langle E_L^2 \rangle - \langle E_L \rangle^2$.

Table 3: The local energy calculated for $a = 0$ without hard-sphere interaction with the Metropolis-Hastings algorithm with analytical local energy calculation, this time with proper evaluation of the statistical error. The calculations are run in three dimentions with $1e6$ Monte Carlo cycles, and the variational paramameter $\alpha = 0.6$

N	$\langle E_L \rangle$ $[\hbar\omega_{HO}]$	σ^2 Blocking	$\sigma^2 \approx \langle E_L^2 \rangle - \langle E_L \rangle^2$
1	1.5000	5.0102e-2	5.0100e-2
10	15.000	4.9701e-1	4.9678e-1
100	150.00	5.2417	5.2184
500	750.00	9.0710	8.4831

4.5 VMC with repulsive interaction

In table 4, the results from the calculation of the elliptical trap are presented. The variational parameter α was varied manually to find the minimum of the local energy E_L.

Table 4: The local energy E_L calculated for different α, with $a = 0.0043$ with hard-sphere interaction, elliptical trap and $\beta = 2.82843$, with the brute force algorithm with analytical local energy calculation. The calculations are run in three dimentions with $1e6$ Monte Carlo cycles

N	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.\ 35$	$\alpha = 0.\ 4$	$\alpha = 0.\ 5$
1	1.96397	1.70441	1.68752	1.70148	1.79225
10	19.6679	17.1322	16.9584	17.1359	18.0973
100	255.34	253.248	262.777	275.639	307.241

4.5.1 Gradient Decent

The gradient decent method was then used to find the minimum in the local energy in the interacting case. The results are presented in table 5.

Table 5: The local energy E_L calculated with the gradient decenet method, with $a = 0.0043$ with hard-sphere interaction, elliptical trap and $\beta = 2.82843$, with the brute force algorithm with analytical local energy calculation. The table shows the optimal α found by the method, and the resulting local energy. The calculations are run in for three dimentions with $1e6$ Monte Carlo cycles.

N	α	E_L	$\frac{\partial E_L}{\partial \alpha}$
1	0.349923	1.68819	0.000853655
10	0.345019	16.9141	-0.365277

4.6 Onebody densities

In section 2.1 we presented a couple of ways computing the onebody density. Because both should give the same result, we selected the simplest one, which is the method with the bins. We conduct the investigations in elliptic traps, with 100 particles, 3 dimensions and 1e6 Monte Carlo cycles. For three different choices of a, the onebody density plots are found in figure (5-7).

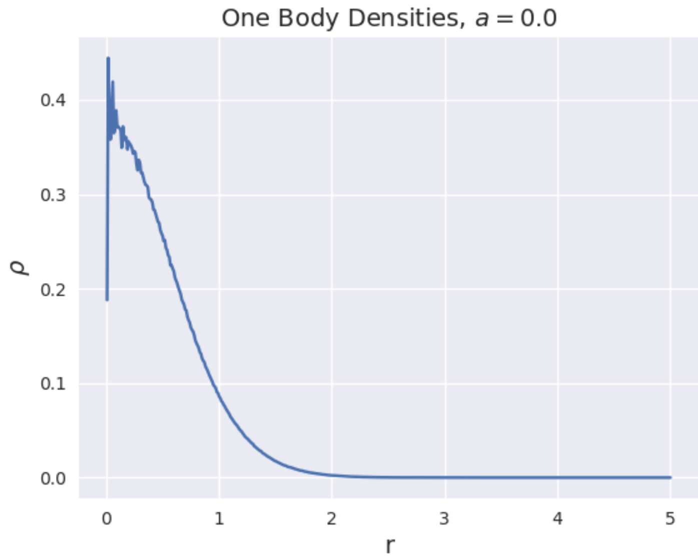

Figure 5: Add caption

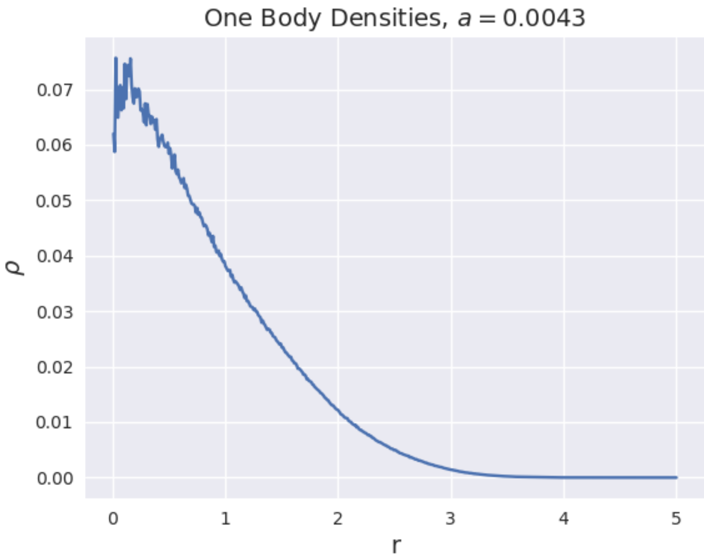

Figure 6: Add caption

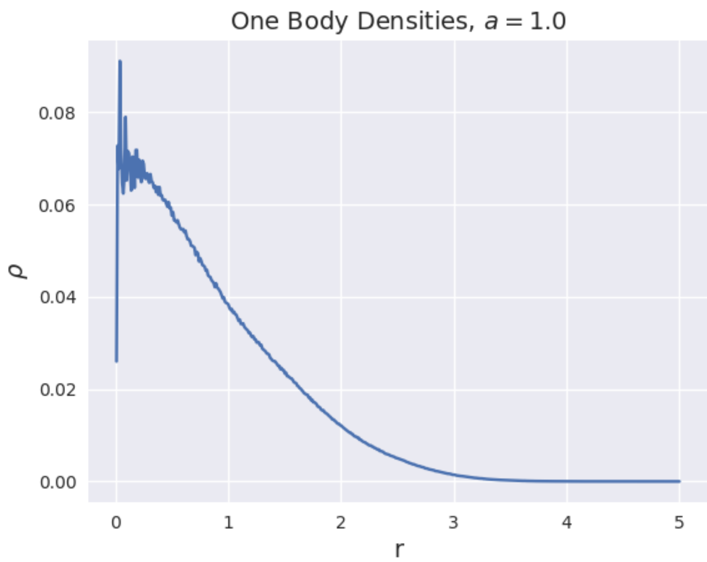

Figure 7: Add caption

5 Discussion

6 Conclusion

7 Appendix

7.1 Appendix A

We calculated the analytical expression for the local energy E_L, as given by 8, for the no-interaction ($a = 0$) case for the spherical harmonic oscillator. In this case, the wave trial function for only consists of the one body part, and is thus for N particles given by:

$$\Psi_T(\vec{r}) = \prod_i^N e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)} \tag{43}$$

We now want to calculate the analytical expressions for E_L for one particle and one dimention, and N particles and three dimentions.

7.1.1 One particle, one dimention

From 43, the trial wave function for one particle and one dimention is as follows:

$$\Psi_T(x) = e^{-\alpha x^2} \tag{44}$$

In the Hamiltonian, the Laplace-operator reduces to the partial derivative in the x-direction, $\frac{\partial^2}{\partial x^2}$, and E_L is

$$\begin{aligned} E_L(x) &= \frac{1}{\Psi_T(x)}\hat{H}\Psi_T(x) \\ &= e^{\alpha x^2}(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega_{HO}^2 x^2)e^{-\alpha x^2} \\ &= e^{\alpha x^2}[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}(e^{-\alpha x^2}) + \frac{1}{2}m\omega_{HO}^2 x^2(^{-\alpha x^2})] \end{aligned} \tag{45}$$

Double differentiation of $e^{-\alpha x^2}$ yields

$$\frac{\partial^2}{\partial x^2}(e^{-\alpha x^2}) = e^{-\alpha x^2} 2\alpha(2x^2 - 1) \tag{46}$$

Inserting this into the expression for E_L gives

$$E_L(\alpha) = -\frac{\hbar^2}{m}\alpha(2x\alpha^2 - 1) + \frac{1}{2}m\omega_{HO}^2 x^2 \tag{47}$$

7.1.2 N particles, three dimentions

When extending the non-interaction case to N particles and three dimentions, the trial wave function will take the form listed in 43, and the Hamiltonian will be as listed in 1, with the spherical harmonic oscillator potential $\frac{1}{2}m\omega_{HO}^2 r_i^2$, as listed in 2.

The local energy is then

$$E_L(\alpha) = \prod_i^N e^{\alpha(x_i^2+y_i^2+\beta z_i^2)} \sum_i [-\frac{\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\omega_{HO}^2(x_i^2+y_i^2+z_i^2)] \prod_i^N e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)} \tag{48}$$

Double differentiation of Ψ_T with respect to x gives

$$\frac{\partial^2}{\partial x^2}(\prod_i e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)}) = \prod_i e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)}[\sum_{i,j} 4\alpha^2 x_i x_j - 2\alpha N] \tag{49}$$

Double differentiation with respect to y will give similar answer, while for z there will be a factor β difference

$$\frac{\partial^2}{\partial z^2}(\prod_i e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)}) = \prod_i e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)}[\sum_{i,j} 4\alpha^2\beta^2 y_i y_j - 2\alpha\beta N] \tag{50}$$

By including the answers from the differentiations in the x,y and z directions and simplifying the expression results in the following expression for the local energy

$$\begin{aligned} E_L(\alpha) = -\frac{\hbar^2}{2m}N[\sum_{i,j} 4\alpha^2 x_i x_j + \sum_{i,j} 4\alpha^2 y_i y_j + \sum_{i,j} 4\alpha^2\beta^2 z_i z_j] + \\ \frac{2\hbar^2}{m}\alpha N^2 + \sum_i \frac{1}{2} m\omega_{HO^2}(x_i^2 + y_i^2 + z_i^2) + \frac{\hbar^2}{m}\beta\alpha N^2 \end{aligned} \tag{51}$$

7.2 Appendix B

In the theory part we claimed that the Hamiltonian could be written as

$$H = \sum_i \left(\frac{1}{2}\Big(-\nabla^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2\Big)\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{52}$$

with

$$\gamma = \beta = \frac{\omega_z}{\omega_{HO}}$$

for an elliptical harmonic oscillator potential with repulsive interaction. Let us start from scratch, where the unscalled Hamiltonian for an elliptical harmonic oscillator reads

$$H = \sum_i \left(-\frac{\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\Big(\omega_{HO}^2(x_i^2 + y_i^2) + \omega_z^2 z_i^2\Big)\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j).$$

We then scale the entire equation with respect to $\hbar\omega_{HO}$

$$\frac{H}{\hbar\omega_{HO}} = \sum_i \left(-\frac{\hbar}{2m\omega_{HO}}\nabla_i^2 + \frac{1}{2}\frac{m}{\hbar}\omega_{HO}(x_i^2+y_i^2) + \frac{1}{2}\frac{m}{\hbar}\frac{\omega_z^2}{\omega_{HO}}z_i^2\right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j)$$

where we can take $H' = H/\hbar\omega_{HO}$ as the dimensionless energy. Further we scale all the lengths in the same way

$$x_i^2 = (x_i')^2 \cdot a_{HO}^2 = (x_i')^2 \cdot \frac{\hbar}{m\omega_{HO}},$$

and we finally obtain

$$H' = \sum_i \frac{1}{2}\bigg(-\nabla_i^2 + (x_i')^2 + (y_i')^2 + \frac{\omega_z^2}{\omega_{HO}^2}(z_i')^2 \bigg) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \qquad (53)$$

which is the Hamiltonian that we were hunting. We know how the z-component is affected by β, so β has to be equal to ω_z/ω_{HO}.