# FYS4411 - Computational Physics II

## Project 1

Dorthea Gjestvang
Even Marius Nordhagen

February 28, 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

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

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

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

$\alpha$ is a variational parameter that we later use to find the energy minimum, and $\beta$ is a constant. This is also the form of the exact ground state wave function for a harmonic oscillator, so by choosing the correct $\alpha$, 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 $\alpha$, 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

$$E = \hbar\omega(n + dim/2) \tag{7}$$

where $n$ is the energy level and $dim$ is number of dimensions. In this project we will study the ground state only, such that $n = 0$. The local energy is

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

which gives the following results considering $a = 0$:

INSERT ANALYTICAL EXPRESSIONS FROM A

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

$$f(a, r_{ij}) = \exp\left( \sum_{i<j} u(r_{ij}) \right) \tag{9}$$

and doing a change of variables

$$\frac{\partial}{\partial \vec{r}_k} = \frac{\partial}{\partial \vec{r}_k}\frac{\partial r_{kj}}{\partial r_{kj}} = \frac{\partial r_{kj}}{\partial \vec{r}_k}\frac{\partial}{\partial r_{kj}} = \frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}}\frac{\partial}{\partial r_{kj}} \tag{10}$$

one will end up with

$$\begin{aligned} E_L = \sum_k \Bigg( & -\frac{1}{2}\Bigg(4\alpha^2\Big(x_k^2 + y_k^2 + \beta^2 z_k^2 - \frac{1}{\alpha} - \frac{\beta}{2\alpha}\Big) \\ & - 4\alpha \sum_{j\neq k}(x_k, y_k, \beta z_k)\frac{(\vec{r}_k - \vec{r}_j)}{r_{kj}}u'(r_{kj}) \\ & + \sum_{ij\neq k}\frac{(\vec{r}_k - \vec{r}_j)(\vec{r}_k - \vec{r}_i)}{r_{ki}r_{kj}}u'(r_{ki})u'(r_{kj}) \\ & + \sum_{j\neq k}\Big(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\Big)\Bigg) + V_{ext}(\vec{r}_k)\Bigg). \end{aligned} \tag{11}$$

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

$$E_{L,i} = -\frac{\hbar^2}{2m}\frac{\nabla_i^2\Psi_T}{\Psi_T} + V_{ext}(\vec{r}_i) = E_{k,i} + E_{p,i} \tag{12}$$

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

$$f''(x) \simeq \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \tag{13}$$

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

$$V_{int}(\vec{r}_i, \vec{r}_j) = \begin{cases} 0 & \text{if} \quad |\vec{r}_i - \vec{r}_j| \geq a \\ \infty & \text{if} \quad |\vec{r}_i - \vec{r}_j| < a. \end{cases} \tag{14}$$

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.

For the case when we have elliptical harmonic oscillator potential, we can obtain a quite simple expression for the Hamiltonian when scaling with respect to $\hbar\omega_{HO}$ (something we actually already have done with the energy). The expression we get is

$$H = \sum_i \left( \frac{1}{2} \Big( -\nabla^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2 \Big) \right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{15}$$

where

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

This quantity is equivalent to $\beta$, see Appendix B to see how this expression is derived.

# 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.

$$\vec{R}_{new} = \vec{R} + r \cdot \text{step}. \tag{16}$$

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

$$W_{i\to j} = T_{i\to j} \cdot A_{i\to j} \tag{17}$$

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

$$P_i^{(n)} = \sum_j \left[ P_j^{(n-1)} T_{j\to i} A_{j\to i} + P_i^{(n-1)} T_{i\to j}(1 - A_{i\to j}) \right] \tag{18}$$

since this can happend in two 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

$$P_i^{(n)} = P_i^{(n-1)} + \sum_j \left[ P_j^{(n-1)} T_{j\to i} A_{j\to i} - P_i^{(n-1)} T_{i\to j} A_{i\to j} \right]. \tag{19}$$

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

$$\sum_j \left[ p_j T_{j\to i} A_{j\to i} - p_i T_{i\to j} A_{i\to j} \right] = 0. \tag{20}$$

Rearranging, we obtain a quite useful result

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i T_{i\to j}}{p_j T_{j\to i}} \tag{21}$$

### 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

$$w = \frac{P(\vec{R}_{new})}{P(\vec{R})} = \frac{|\Psi_T(\vec{R}_{new})|^2}{|\Psi_T(\vec{R})|^2}. \tag{22}$$

$w$ 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:

$$\text{New position: } \begin{cases} \text{accept} & \text{if} \quad w > r \\ \text{reject} & \text{if} \quad w \leq r. \end{cases} \tag{23}$$

### 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

$$\frac{\partial P(R,t)}{\partial t} = D\frac{\partial}{\partial R}\left(\frac{\partial}{\partial R} - F\right)P(R,t). \tag{24}$$

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

$$\frac{\partial R(t)}{\partial t} = DF(R(t)) + \eta \tag{25}$$

where $\eta$ can be considered as a random variable. This differential equation can be solved by applying the forward Euler method and introducing gaussian variables $\xi$

$$R_{new} = R + DF(R)\Delta t + \xi\sqrt{\Delta t} \tag{26}$$

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

$$\begin{aligned} T_{R\to R_{new}} &= G(R_{new}, R, \Delta t) \\ &= \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp[-(R_{new} - R - D\Delta t F(R))^2/4D\Delta t] \end{aligned} \tag{27}$$

and the acceptance criteria becomes

$$r < \frac{G(R, R_{new}, \Delta t)|\Psi_T(R_{new})|^2}{G(R_{new}, R, \Delta t)|\Psi_T(R)|^2}. \tag{28}$$

## 3.3 Minimization methods

When the interaction term is excluded, we know which $\alpha$ that corresponds to the energy minima, and it is in principle no need to try different $\alpha$'s. However, sometimes we have no idea where to search for the minimum point, and we need to try various $\alpha$ 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

$$\alpha^+ = \alpha - \eta \cdot \frac{d\langle E_L(\alpha)\rangle}{d\alpha}. \tag{29}$$

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

# 4 Results

## 4.1 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. All the measurements are done in three dimensions with $1e6$ Monte Carlo cycles. $a$ is fixed to zero.

Table 1: The local energy calculated numerically for $a = 0$ and without hard-sphere interaction. The number of Monte Carlo cycles is fixed to $M = 1e6$, and the performance is presented along with the local energies. The analytical local energy is given by $E_L = 0.5 \cdot N \cdot d$ where $N$ is the number of particles and $d$ is the number of dimensions, in our case set to 3.

| | **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 | 0.146392 | 1.49999 | 0.426018 |
| 10 | 15.000 | 11.8992 | 14.9999 | 38.0378 |
| 100 | 150.00 | 8326.16 | | |
| 500 | | | | |

We observe that the calculated local energy is close to the exact energy.

## 4.2 VMC without repulsive interaction

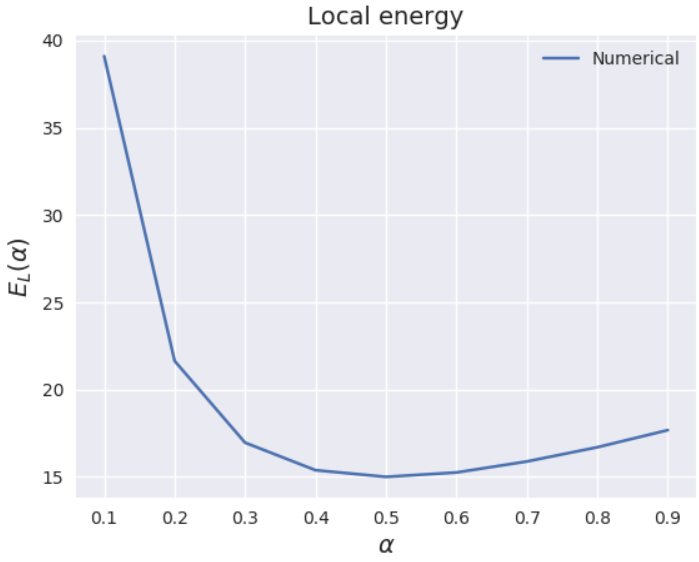


Figure 1: Add caption

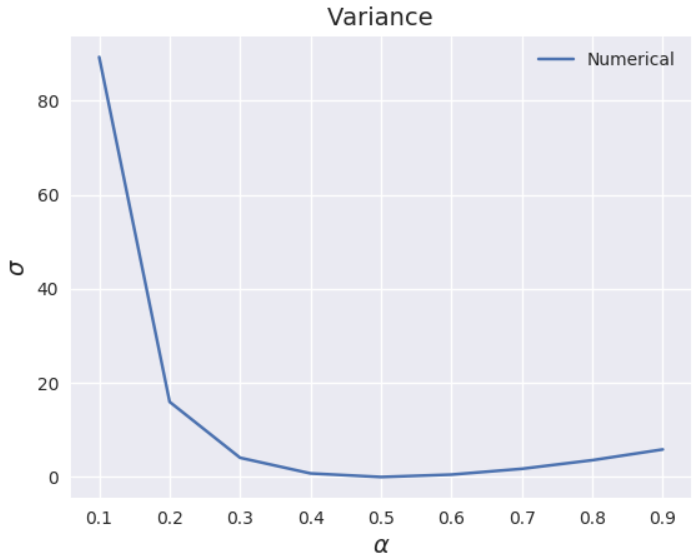


Figure 2: Add caption

### 4.3 Standard Metropolis vs. Importance sampling

We want to see if the importance sampling really converges faster, and compare the energy with respect to the number of cycles. The interaction is still turned off, and we study 10 particles in 3 dimensions in the minimum point. The only thing that is different is the acceptance ratio?

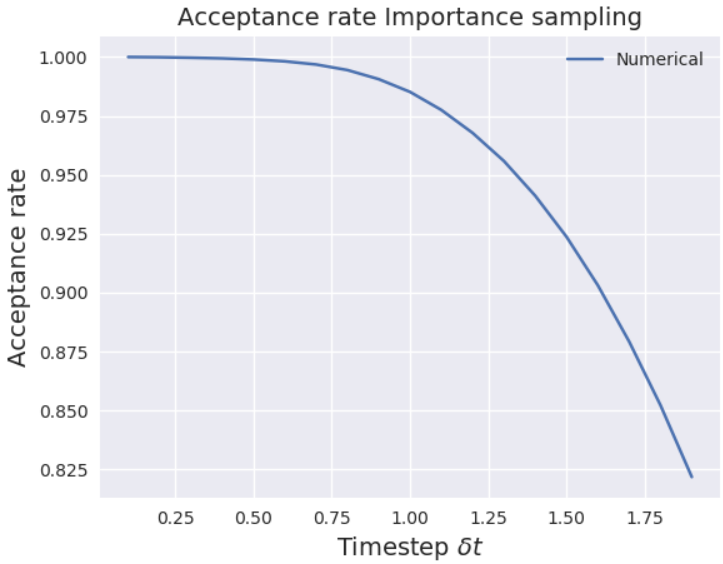


Figure 3: Add caption

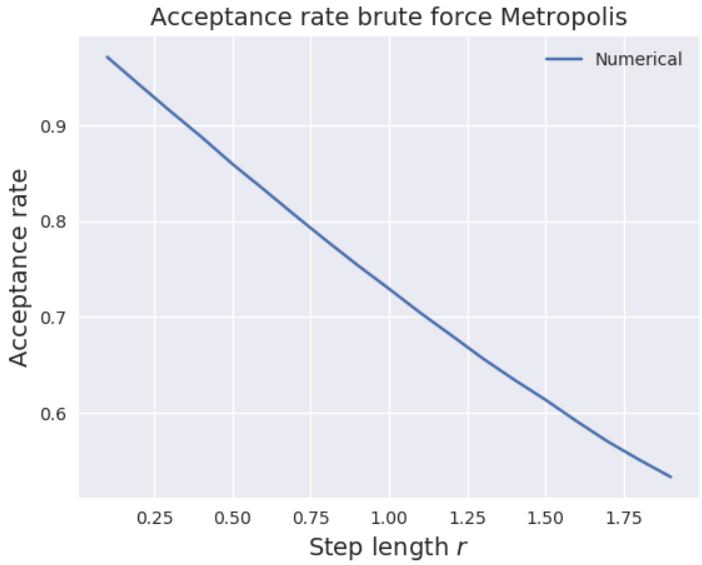


Figure 4: Add caption

### 4.4 VMC with repulsive interaction

We are now turning on the interaction and the elliptical harmonic oscillator at the same time.

# 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:

$$\Psi_T(\vec{r}) = \prod_i^N e^{-\alpha(x_i^2+y_i^2+\beta z_i^2)} \tag{30}$$

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

$$\Psi_T(x) = e^{-\alpha x^2} \tag{31}$$

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

$$\begin{aligned} E_L(x) &= \frac{1}{\Psi_T(x)}\hat{H}\Psi_T(x) \\ &= e^{\alpha x^2}(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega_{HO}^2 x^2)e^{-\alpha x^2} \\ &= e^{\alpha x^2}[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}(e^{-\alpha x^2}) + \frac{1}{2}m\omega_{HO}^2 x^2()e^{-\alpha x^2})] \end{aligned} \tag{32}$$

## 7.2 Appendix B

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

$$H = \sum_i \left( \frac{1}{2} \Big( -\nabla^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2 \Big) \right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j) \tag{33}$$

with

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

for an elliptical harmonic oscillator potential with repulsive interaction. All the quantities in the expression are scalled, such that it is dimensionless. The energy ($H$) is scalled with respect to $\hbar\omega_{HO}$, while the length is scalled with respect to the characteristic length of the trap, $a_{HO} = (\hbar/m\omega_{HO})^{1/2}$. 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 the 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

$$H' = \sum_i \frac{1}{2} \left( -\nabla_i^2 + (x_i')^2 + (y_i')^2 + \frac{\omega_z^2}{\omega_{HO}^2}(z_i')^2 \right) + \sum_{i<j} V_{int}(\vec{r}_i, \vec{r}_j). \tag{34}$$

which is the Hamiltonian that we were hunting.