Variational Monte Carlo
— Project 1 —
FYS4411

Heine Olsson Aabø

Abstract

In this project the variational Monte Carlo is studied to approximate the ground state energy of a trapped Bose gas. A model is developed by starting with a simple non-interacting case, and expanding the to interacting model limited to two-body interactions. For the simple non-interacting system, the exact ground state energy is found for a variational parameter $\alpha = 0$, while the interacting system proves to be more difficult to approximate, and gradient descent is used to find optimized parameters. As we are dealing with stochastic variables, the data produces must be assumed to be correlated and so a statistical analysis of the sample mean errors must is examined through the blocking method, which gives a better expression for the variance than ordinary underestimated variance. The implementation of the Monte Carlo is eventually found to be incorrect, leading to notable errors for the interacting case.

Contents

1 Introduction

When studying a physical microscopic system, the thermodynamical properties such as the ground state energy is (often) of particular interest. If the ground state energy is known, excited states are easier to study. However, in many cases the exact solution is very difficult to find, and approximate methods have to be used. One such method is the variational Monte Carlo (VMC). Compared to other many-body methods the VMC usually give good results with a much simpler implementation.

In this project the ground state energy of a trapped Bose gas will be evaluated, both with a spherical and elliptical harmonic oscillator trap. The ground state energy will be approximated for a varying number of particles in one to three dimensions, along with a variational parameter where an optimal value will be found using gradient descent. Since the generated data may be correlated, an analysis of the statistical errors will be made with the blocking method. All data represented can be reproduced with the supplementary material found at https://github.com/heineaabo/FYS4411/tree/master/Project1.

This project consist of 5 upcoming sections. First the theory behind the methods used and the system of interest will be presented. Next a derivation of the analytical expressions are found in the implementation. Then the results will briefly be stated, followed by a more thorough discussion in the end along with a short conclusion.

2 Theory

2.1 Monte Carlo Integration

Monte Carlo is well known method in physics that uses random sampling and statistical analysis to perform numerical calculations. Assume we want to evaluate the multidimensional integral over some domain Ω

$$I = \int_\Omega f(\mathbf{X}) d\mathbf{X} \tag{1}$$

this will become a very tedious process if using standard numerical algorithms. By using Monte Carlo integration we randomly sample N different values of f giving an approximation to I equal to the sample mean. From the law of large numbers we then have

$$I \approx \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{X}_i) \tag{2}$$

where $\boldsymbol{r}_i$ denotes the i-th sample. In this limit the samples are normal distributed according to the central limit theorem.

As f is randomly sampled, we can assume $\boldsymbol{X}$ to be a random variable, and thus f to be a random function with a probability distribution p. The expected value of f is then

$$\begin{aligned} E[f(\boldsymbol{X})] &= \int_\Omega f(\mathbf{X}) p(\mathbf{X}) d\mathbf{X} \\ &\approx \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{X}_i) p(\mathbf{X}_i) \end{aligned} \tag{3}$$

2.2 Variational Monte Carlo

The variational Monte Carlo is utilizes Monte Carlo integration to approximate the ground state energy of a system of particles. When the wavefunction of a system is unknown a trial wave function has to be guessed, from physically motivated grounds, along with some variational parameters. According to the variational principle, given a trial wave function Ψ_T and a Hamiltonian H, the expectation value of the $\langle H \rangle$ is an upper bound to the ground state energy E_0 of the system:

$$E_0 \leqslant \langle H \rangle \tag{4}$$

From quantum mechanics the expectation value an operator, in this case the Hamiltonian, is given as

$$\langle H \rangle = \frac{\int d\boldsymbol{r} \Psi_T^*(\mathbf{r}) H(\mathbf{r}) \Psi_T(\mathbf{r})}{\int d\mathbf{r} \Psi_T^*(\mathbf{r}) \Psi_T(\mathbf{r})} \tag{5}$$

$$= \frac{\int d\boldsymbol{r} |\Psi_T(\mathbf{r})|^2 \frac{1}{\Psi_T(\mathbf{r})} H(\boldsymbol{r}) \Psi_T(\mathbf{r})}{\int d\mathbf{r} |\Psi_T(\mathbf{r})|^2} \tag{6}$$

By defining a quantity called the local energy as

$$E_L(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} H \Psi_T(\mathbf{r}) \tag{7}$$

and a normalized probability distribution

$$p(\mathbf{r}) = \frac{|\Psi_T(\mathbf{r})|^2}{\int d\mathbf{r} |\Psi_T(\mathbf{r})|^2} \tag{8}$$

eq. (5) becomes equivalent to eq. (3), being the standard statistical expression for expectation values:

$$\langle E \rangle = \int d\mathbf{r} E_L(\mathbf{r}) p(\mathbf{r}) \tag{9}$$

with variance

$$\sigma_E^2 = \left\langle (E - \langle E \rangle)^2 \right\rangle = \left\langle E^2 \right\rangle - \langle E \rangle^2 \tag{10}$$

From equation eq. (2) we then have

$$\langle E \rangle \approx \frac{1}{N} \sum_{i=1}^{N} E_L(\mathbf{r}_i) p(\mathbf{r}_i) \tag{11}$$

for an arbitrary choice of N, being the number of Monte Carlo cycles.

2.3 Metrolpolis algorithm

The Monte Carlo will sample random positions in coordinate space, with $\boldsymbol{r}_i$ as a stochastic variable. The position of each particle is then modelled as random discrete jumps with time, called a Markov chain. This is ultilized in the brute force Metropolis algorithm and the Metropolis-Hastings algorithm, often just called Importance sampling.

2.3.1 Markov chains

A Markov chain is defined as a sequence of discrete stochastic events where the probability for transitioning from one state to another solely depends on the current state. When a Markov chain becomes long enough, that is when it runs for a long enough time, the most likely state will be reached if it obey the conditions of ergodicity and detailed balance. Simply defined, ergodicity means that all states are equally probable, so the Markov chain should be able to reach every state from any initial state if run long enough. Detailed balance can be expressed as

$$\frac{w_i}{w_j} = \frac{W(j \to i)}{W(i \to j)} \tag{12}$$

where W is the transition probability between two states, and w is a time-independent probability distribution for occupying a state. In our case w is given by eq. (8). When the system has reached its most likely state, an equilibrium distribution is reached. This is the case if there is an equal probability that the system makes a transition to or from any state i. Since the transition rate is unknown, it must be modelled as

$$W(i \to j) = T(i \to j)A(i \to j) \tag{13}$$

where T is the likelihood for making a transition and A is the acceptance likelihood. Equation (12) then becomes

$$\frac{|\Psi_T|_i^2}{|\Psi_T|_j^2} = \frac{T(j \to i)A(j \to i)}{T(i \to j)A(i \to j)} \tag{14}$$

if the probability $|\Psi_T|^2$ is assumed to be normalized.

2.3.2 Brute force

Equation (14) is the base for the so called Metropolis algorithm. When the Monte Carlo is sampling random position we want it to move towards the largest probabilities. This is done by accepting large probabilities, when $w_i/w_j > 1$. However smaller probabilities must be allowed so that the Monte Carlo does not always give the largest probability, $w_i/w_j < 1$. To begin with we assume that $T(i \to j) = T(j \to i)$. Since the ratio w_i/w_j will be calculated, and thereby known, the Metropolis algorithm can be represented as

$$A(j \to i) = \min\left(1, \frac{|\Psi_T|_i^2}{|\Psi_T|_j^2}\right) \tag{15}$$

accepting a sampled move if $\frac{|\Psi_T|_i^2}{|\Psi_T|_j^2} \geqslant X$, with a uniformly distributed random variable $X \in [0, 1]$.

2.3.3 Importance sampling

The exact ground state energy of a system will give zero variance. To reduce the variance we want to sample regions of coordinate space that give good values from the wave function, giving more relevant sampling. This can be done with the Metropolis-Hastings algorithm, represented as

$$A(j \to i) = \min\left(1, \frac{|\Psi_T|_i^2 T(j \to i)}{|\Psi_T|_j^2 T(i \to j)}\right) \tag{16}$$

Here $T(i \to j) \neq T(j \to i)$ is assumed, which actually is the case, both being unknown and so a model is needed.

In one dimension the Fokker-Planck equation is

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} - F(x)\right)P(x,t) \tag{17}$$

which describes the time evolution of a probability density function. Here D is the diffusion coefficient, which is set to $D = 0.5$ for this project, and F is the quantum force responsible for moving the particle towards region of coordinate space where the wave function gives large values. New positions will be the solution of the Langevin equation

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \tag{18}$$

where η is a random variable, giving the new position:

$$x' = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \tag{19}$$

where ξ is a gaussian random variable and Δt is a given time step. The quantum force is given as

$$F = 2\frac{1}{\Psi_T}\nabla\Psi_T \tag{20}$$

Then, choosing new positions with eq. (19) and accepting them according to eq. (16) will give more relevant sampling. The solution of the Fokker-Planck equation yields a transition probability given by Green's function.

$$T(j \to i) = G(y, x, \Delta t) = \frac{e^{-\frac{(y-x-D\Delta t F(x))^2}{4D\Delta t)}}}{(4\pi D\Delta t)^{3N/2}} \tag{21}$$

2.4 Blocking

The Monte Carlo simulation will generate a set $\{X_1, ..., X_N\}$ of stochastic variables, in this case being the ground state energy. As we are dealing with stochastic variables, we have to assume that there may be covariance between X_i and X_j.

$$\text{cov}(X_i, X_j) = \langle X_i X_j\rangle - \langle X_i\rangle\langle X_j\rangle \tag{22}$$

which is just the variance if $i = j$. A useful measurement of the pairwise correlations is the autocorrelation function

$$k_d = \frac{\gamma(d)}{var(X)} \tag{23}$$

Here $\gamma(d)$ is the correlation function

$$\gamma(h) = \frac{1}{n}\sum_{i=1}^{n-h}(X_i - \bar{X}_n)(X_{i+h} - \bar{X}_n) \tag{24}$$

where $\bar{X}_n$ is the sample mean. The statistical error, or the error of the sample mean, can be expressed with k_d

$$\mathrm{err}_X^2 = \left(1 + 2\sum_{d=1}^{n-1} k_d\right)\frac{1}{n}\mathrm{var}(X) = \frac{\tau}{n}\mathrm{var}(X) \tag{25}$$

τ is called the autocorrelation time and is $\neq 1$ for correlated data. For big data sets a good way to estimate the statistical error is with the blocking method. Here the algorithm given in [2] will be used.

The blocking methods transforms the data set into smaller and smaller sets, half as large per time, $\{X'_1, ..., X'_{n'}\}$ with $n' = \frac{1}{2}n$ and $X'_i = \frac{1}{2}(X_{2i-1} + X_{2i})$. Under this blocking-transformation the sample mean and sample variance is invariante; $\bar{X}' = \bar{X}$ and $\sigma^2(\bar{X}') = \sigma^2(X)$. This transformation continues as long as the new set conatins more than two elements. During every transformation the sample variance and sample covariance is computed. An appropriate estimator for the variance will be

$$\hat{\mathrm{var}}(\bar{X}) = \frac{\sigma_j^2}{n_j} \tag{26}$$

if $\gamma_k(1) = 0$ for all $k \geqslant j$, where $\gamma_k(1)$ is the correlation function given by eq. (24) for the k-th blocking transformation with $h = 1$.

2.5 Gradient descent

Gradient descent is a optimization method for finding minima of a given function. If the derivative of the function with respect to some parameter θ can be defined, then the direction of a minimum, local or global, will be where the derivative is negative. By iteratively taking steps in this direction a minimum is certain to be found if it exists. The process can be made faster by changing the step size, with several different approaches constituting different types of gradient descent. This can also make it easier to avoid local minima. The simpelest approach is to have a fixed step size, or learning rate η. More advanced approaches use adaptive learning rates depending on different properties of the gradient. For a fixed learning rate the iterative update of the parameters become:

$$\hat{\theta} \leftarrow \hat{\theta} - \eta\nabla\mathbf{f}(\hat{\theta}) \tag{27}$$

2.6 The system

The system of interest is a trapped Bose gas of alkali atoms ^{87}Rb. Such a system is said to be dilute, where the average spacing between the atoms is much larger than the range of the interaction between them. The atoms are treated as hard spheres with scattering length a. For this system it is chosen to be $a_{Rb} = 4.33 \times 10^{-3} a_{ho}$, where a_{ho} is a typical trap size. The trap potential V_{ext} is defined as a spherical (S) or elliptical (E) harmonic oscillator potential:

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2}m[\omega_{ho}^2(x^2+y^2) + \omega_z^2 z^2] & (E) \end{cases} \tag{28}$$

with potenital frequency ω_{ho} in the xy-plane, and ω_z in the z direction. For the elliptical trap $\omega_{ho} \neq \omega_z$.

2.6.1 Hamiltonian

The Hamiltonian of the system is given by

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + V_{\text{ext}}(\mathbf{r}_i)\right) + \sum_{i<j}^N V_{\text{int}}(\mathbf{r}_i, \mathbf{r}_j) \tag{29}$$

where V_{int} is the two-body interaction:

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leqslant a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{30}$$

2.6.2 Trial wavefunction

In order to obtain accurate results, we have to choose a a trial wave that can make a good approximation of the unknown eigenstates of the Hamiltonian. The trial wave function is constructed as two parts, a single-particle function and a correlation function limiting the interaction of the particles to two-body interactions. The single-particle function is chosen as proportional to the harmonic oscillator function for the ground state

$$g(\alpha, \beta, \mathbf{r}_i) = e^{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)} \tag{31}$$

The correlation function is chosen as the exact solution of the Schrödinger equation for low-energy atom pairs interacting via the hard-core potential V_{int}. Using a correlated trial wave function directly incorporate short range repulsion between particles into the wave function**??**, where the simplest correlation function is on the Jastrow form:

$$\Psi_C = \prod_{i<j} f(a, |\mathbf{r}_i - \mathbf{r}_j|) \tag{32}$$

with ansatz

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leqslant a \\ 1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|} & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{33}$$

This yields the trial wave function

$$\Psi_T(\mathbf{r}, \alpha, \beta) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i<j} f(a, |\mathbf{r}_i - \mathbf{r}_j|) \tag{34}$$

2.6.3 Local energy

As mentioned section 2.2 the local energy is given as

$$E_L(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})} H\Psi_T(\mathbf{r})$$

The analytical expression for this contains a comprehensive derivation of the laplacian of Ψ_T and will be elaborated in section 3 and section 7.

3 Implementation

The code is written in python optimized using the numba compiler. To avoid inefficient computations of the energy and quantum force, the analytic expressions need to be simplified. As we are dealing with small units, scaling these prevents the values from getting too small and thus loss of precision. Common in quantum mechanics is to express the energy in terms of $\hbar\omega$, and as the scattering length of the atoms is given in units of a_{ho} we can set $\hbar = \omega = m = a_{ho} = 1$. The Hamiltonian can now be written as

$$H = \sum_i^N \left(-\frac{\nabla_i^2}{2} + V_{\text{ext}}(\mathbf{r}_i) \right) + \sum_{i<j}^N V_{\text{int}}(\mathbf{r}_i, \mathbf{r}_j) \tag{35}$$

From eq. (35) it is easy to see that the local energy only is dependent of the laplacian of the wavefunction. In similar manor, from **??**, the quantum force is dependent of the gradient of the wavefunction. By writing the trial wave function as $\Psi_T = \Psi_{SP}\Psi_C$ where Ψ_{SP} is the single particle function and Ψ_C is the correlation function, the notation of these expressions becomes more organized. Further more $\Psi_C = \exp(\sum_{i<j} u(r_{kl}))$, with $u(r_{kl}) = \ln f$, where f is the ansatz from eq. (33). We then have for particle k

$$\nabla_k \Psi_T = [\nabla_k \Psi_{SP}]\Psi_C + [\nabla_k \Psi_C]\Psi_{SP} \tag{36}$$

$$\nabla_k^2 \Psi_T = [\nabla_k^2 \Psi_{SP}]\Psi_C + [\nabla_k^2 \Psi_C]\Psi_{SP} + 2[\nabla_k \Psi_{SP}][\nabla_k \Psi_C] \tag{37}$$

The extended derivation of the following equations are presented in section 7. Here it is found that

$$\nabla_k \Psi_{SP} = \frac{\nabla_k g(\mathbf{r}_k)}{g(\mathbf{r}_k)} \Psi_{SP} \tag{38}$$

$$\nabla_k^2 \Psi_{SP} = \frac{\nabla_k^2 g(\mathbf{r}_k)}{g(\mathbf{r}_k)} \Psi_{SP} \tag{39}$$

$$\nabla_k \Psi_C = \left[\sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] \Psi_C \tag{40}$$

$$\nabla_k \Psi_C = \left[\sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] \Psi_C \tag{41}$$

$$\nabla_k^2 \Psi_C = \left[\sum_{l \neq k}^N \nabla_k^2 u(r_{kl}) + 2 \sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] \Psi_C \tag{42}$$

with

$$\nabla_k u(r_{kl}) = \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \tag{43}$$

$$\nabla_k^2 u(r_{kl}) = \frac{D-1}{r_{kl}} u'(r_{kl}) + \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\mathbf{r}_k - \mathbf{r}_m}{r_{km}} u'(r_{kl}) u'(r_{km}) \tag{44}$$

with

$$u'(r_{kl}) = \frac{a}{r_{kl}(r_{kl} - a)} \tag{45}$$

$$u''(r_{kl}) = \frac{a^2 - 2ar_{kl}}{r_{kl}^2 (r_{kl} - a)^2} \tag{46}$$

Where D is the number of dimensions. Finally the gradient and laplacian can be written as

$$\frac{\nabla_k \Psi_T}{\Psi_T} = \frac{\nabla_k g(\mathbf{r}_k)}{g(\mathbf{r}_k)} + \sum_{l \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \tag{47}$$

$$\begin{aligned} \frac{\nabla_k^2 \Psi_T}{\Psi_T} =& \frac{\nabla_k^2 g(\mathbf{r}_k)}{g(\mathbf{r}_k)} + 2\frac{\nabla_k g(\mathbf{r}_k)}{g(\mathbf{r}_k)} \sum_{l \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \\ &+ \sum_{l,m \neq k} \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\mathbf{r}_k - \mathbf{r}_m}{r_{km}} u'(r_{kl}) u'(r_{km}) \\ &+ \sum_{l \neq k} \left(u''(r_{kl}) + \frac{D-1}{r_{kl}} u'(r_{kl}) \right) \end{aligned} \tag{48}$$

with

$$\frac{\nabla_k g(\mathbf{r}_k)}{g(\mathbf{r}_k)} = -2\alpha[x_i, y_i, \beta z_i] \tag{49}$$

$$\frac{\nabla_k^2 g(\mathbf{r}_k)}{g(\mathbf{r}_k)} = -2\alpha(D - 1 + \beta) + 4\alpha^2(x_i^2 + y_i^2 + \beta^2 z_i^2) \tag{50}$$

3.1 Spherical trap potential

For the spherical trap $\beta = 1$ and we assume that the bosons are non-interacting, giving scattering length $a = 0 \Rightarrow V_{int}(\boldsymbol{r}_i) = 0$. From eq. (34) the trial wave functions is expressed as

$$\Psi_T(\mathbf{r}, \alpha) = \prod_i^N \prod e^{-\alpha \mathbf{r}_i^2} \tag{51}$$

From eq. (28), the trap potential is given as

$$V_{ext}(\mathbf{r}_i) = \frac{1}{2}\mathbf{r}_i^2 \tag{52}$$

which gives the Hamiltonian

$$H = \frac{1}{2} \sum_i^N \left(-\nabla_i^2 + \mathbf{r}_i^2 \right) \tag{53}$$

The local energy can be expressed combining eq. (31), eq. (35), eq. (48) and eq. (50).

$$E_L(\mathbf{r}) = \sum_i^N (\alpha D + -2\alpha^2 \mathbf{r}_i^2 + \frac{1}{2}\mathbf{r}_i^2)$$
$$= N\alpha D + \sum_i^N (\frac{1}{2} - 2\alpha^2)\mathbf{r}_i^2 \quad (54)$$

The quantum force for particle i, combining eq. (20), eq. (47) and eq. (49), can be expressed as

$$F(\mathbf{r}_i) = -4\alpha[x_i, y_i, z_i] \quad (55)$$

3.2 Elliptical trap potential

For the elliptical trap $\beta = (\omega_z/\omega_{ho})^2 = \sqrt{8}$ with interacting bosons having scattering length $a = 0.0043$. The trial wave function is given by eq. (34). The Hamiltonian becomes

$$H = \frac{1}{2}\sum_i^N (-\nabla_i^2 + x_i^2 + y_i^2 + \beta^2 z_i^2) + \sum_{i<j}^N V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \quad (56)$$

The local energy becomes a messy expression best presented as

$$E_L(\mathbf{r}) = \sum_i^N \left(\frac{\nabla_k^2 \Psi_T}{\Psi_T} + x_i^2 + y_i^2 + \beta^2 z_i^2 \right)$$
$$+ \sum_{i<j}^N V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) \quad (57)$$

The quantum force for particle i is expressed combining **??** and eq. (47) as

$$F(\mathbf{r}_i) = -4\alpha[x_i, y_i, \beta z_i] + 2a \sum_{l\neq k}^N \frac{(\mathbf{r}_k - \mathbf{r}_l)}{r_{kl}^2(r_{kl} - a)} \quad (58)$$

4 Results

4.1 Spherical trap potential

The Monte Carlo simulation was run for 2^{18} cycles, with time step $\Delta t = 0.01$, for $N = 1, 10, 100, 500$ particles. The results for varying α values, for both brute force Metropolis and Importance sampling is listed below in table 1 and table 5, along with the standard error computed with the blocking method and the variance. Only the one dimensional case for one particle and three dimensional for 10 particles are shown, as the energy per particle and acceptance ratio was constant for larger number of particles, as shown in fig. 1. The standard error for the 10 particle case is shown in fig. 2.

	Metropolis			Importance sampling		
α	$\langle E \rangle$	σ^2	σ_{block}	$\langle E \rangle$	σ^2	σ_{block}
0.3	0.30625	0.00008	0.004155	0.30624	0.00008	0.00411
0.4	0.40347	0.00002	0.003089	0.40350	0.00003	0.00311
0.5	0.50000	0.00000	0.000000	0.50000	0.00000	0.00000
0.6	0.59581	0.00004	0.003393	0.59581	0.00004	0.00340
0.7	0.69092	0.00016	0.004998	0.69095	0.00016	0.00500

Table 1: Energy expectation value for one particles in one dimension, with brute force Metropolis and importance sampling, time step $\Delta t = 0.01$ and 2^{18} Monte Carlo cycles. Along with variance and the standard error.

	Metropolis			Importance sampling		
α	$\langle E \rangle$	σ^2	σ_{block}	$\langle E \rangle$	σ^2	σ_{block}
0.3	9.1861	0.0023	0.0098	9.1859	0.0023	0.0097
0.4	12.1034	0.0007	0.0072	12.1035	0.0007	0.0073
0.5	15.0000	0.0000	0.0000	15.0000	0.0000	0.0000
0.6	17.8760	0.0010	0.0079	17.8762	0.0010	0.0079
0.7	20.7324	0.0049	0.0117	20.7325	0.0049	0.0117

Table 2: Energy expectation value for 10 particles in three dimension, with brute force Metropolis and importance sampling, time step $\Delta t = 0.01$ and 2^{18} Monte Carlo cycles. Along with variance and the standard error.

The energy per particle is plotted in fig. 1 for $N = 1, 10, 100, 500$ in three dimensions.

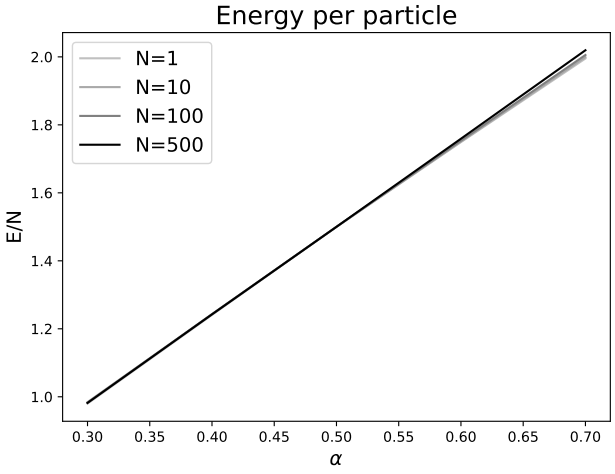

Figure 1: The energy per particle as a function of α, for $N = 1, 10, 100, 500$ in three dimensions.

The standard error for 10 particles in three dimensions is plotted in fig. 2.

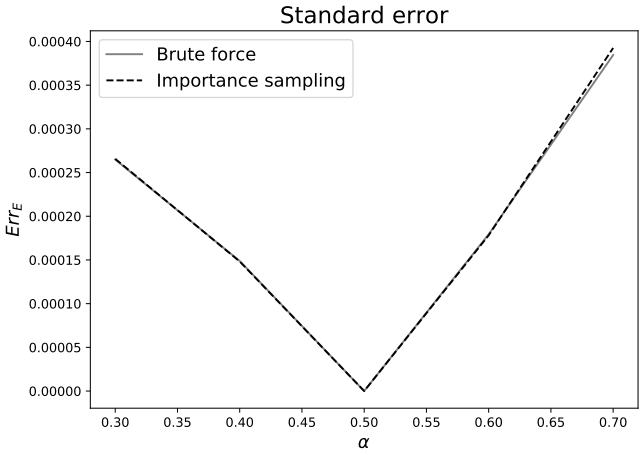

Figure 2: Standard error as a function of α, for the 10 particle case in three dimensions. For both brute force Metropolis and importance sampling.

The acceptance ratios as a function of Δt is plotted in fig. 3.

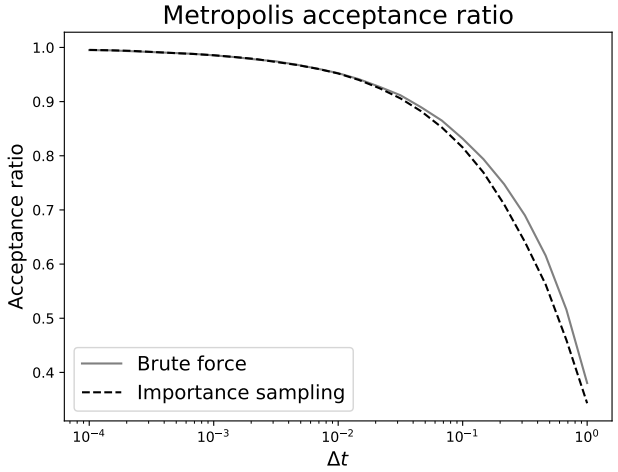

Figure 3: Acceptance ratio of the Metropolis algorithm as a function of Δt, for both brute force and importance sampling.

4.2 Elliptical trap potential

For the elliptical potential, the Monte Carlo simulation was run for 2^{15} cycles, with time step $\Delta t = 0.01$, for $N = 10, 50, 100$ particles.

	Metropolis			Importance sampling		
α	$\langle E \rangle$	σ^2	σ_{block}	$\langle E \rangle$	σ^2	σ_{block}
0.3	15.8737	0.0745	0.0389	15.8733	0.0553	0.0362
0.4	20.7016	0.5678	0.0645	20.6968	0.0708	0.0383
0.5	25.4636	0.1154	0.0434	25.4607	0.0764	0.0394
0.6	30.1732	0.1582	0.0463	30.1766	0.2533	0.0531
0.7	34.8315	0.3095	0.0554	34.8418	0.3306	0.0564

Table 3: Energy expectation value for 10 particles in three dimension in an elliptical trap, with brute force Metropolis and importance sampling, time step $\Delta t = 0.01$ and 2^{18} Monte Carlo cycles. Along with variance and the standard error.

	N = 50	
α	$\langle E \rangle$	σ_{block}
0.3	96.3026	0.1065
0.4	124.3491	0.5094
0.5	151.9292	0.0995
0.6	180.9621	0.8994
0.7	208.1334	0.1071

Table 4: Energy expectation value for 50 particles in three dimension, with importance sampling, time step $\Delta t = 0.01$ and 2^{15} Monte Carlo cycles. Along with standard error.

	N = 100	
α	$\langle E \rangle$	σ_{block}
0.3	227.3495	0.5917
0.4	291.0959	1.1581
0.5	357.3436	0.9340
0.6	423.9965	0.2930
0.7	491.5939	0.7752

Table 5: Energy expectation value for 100 particles in three dimension, with importance sampling, time step $\Delta t = 0.01$ and 2^{15} Monte Carlo cycles. Along with standard error.

The energy per particle is plotted in fig. 4 for $N = 10, 50, 500$ in three dimesions.

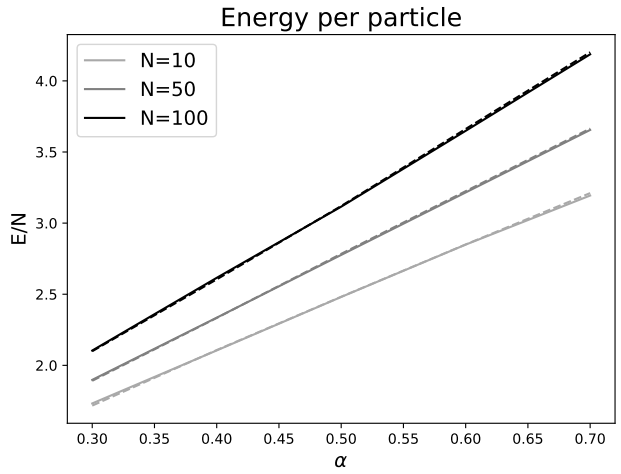

Figure 4: The energy per particle as a function of α, for $N = 10, 50, 100$ in three dimensions.

The standard error as a function of α is plotted in fig. 5 for 10 particles in three dimensions.

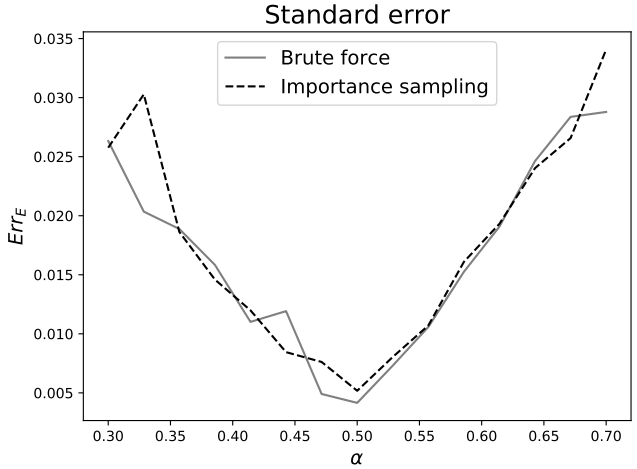

Figure 5: Standard error as a function of α, for the 10 particle case in three dimensions. For both brute force Metropolis and importance sampling.

The acceptance ratio as a function of Δt is plotted in fig. 6.

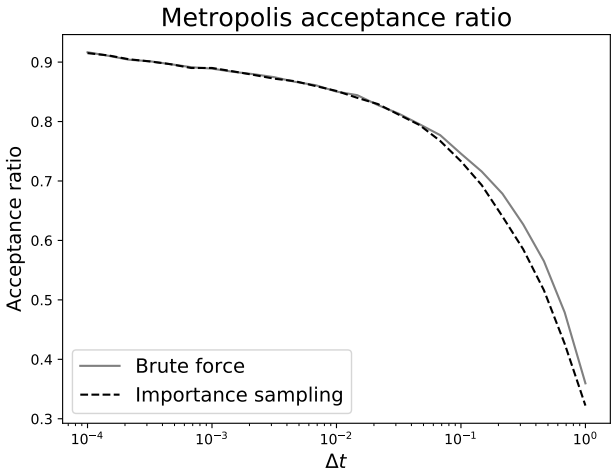

Figure 6: Acceptance ratio of the Metropolis algorithm as a function of Δt, for both brute force and importance sampling.

The optimal parameters found with gradient descent is given in table 6.

N	α	$\langle E \rangle$	σ^2
10	0.48711	24.32425	0.08222
50	0.41342	119.74848	30.39674
100	0.37088	245.73405	48.44908

Table 6

5 Discussion

5.1 Metropolis algorithm

From fig. 3 and fig. 6 it is clear that something went wrong with the implementation of either the importance sampling or the brute force Metropolis. The two methods tend to accept proposed steps at equal rate, also as Δt is changed. The difference between the two is here little, and so choosing either one should not make any major difference. However, as a correctly implemented importance sampling will push the particles towards more interesting regions, the results, and especially the variance, should be significantly better than the brute force method. From the tables in section 4 where the two is compared, the difference is small or non-existing. In order to better the results, a correct implementation of the Metropolis algorithm is needed. One possible source of error could be in the expression of the transition probabilities, the Green's function which was rewritten in order to avoid a division of the two transition probabilities. This was done as there initially were some problems that often led to zero-division, due to wrong analytic expressions for the trial wave function. Although this is not very likely as the mathematics should be correct. Looking back at the lecture notes, one property of the brute force algorithm seems to be overlooked, as the proposed step should be a random variable between 0

and 1, whereas the code samples random variables from the normal distribution, yielding possible negative values, an obvious source of error. Also the proposed move contains the Diffusion coefficient and seems to always assume importance sampling, this should probably not be the case.

5.2 Energy per particle

From fig. 1 and fig. 4 it seems like the difference between the interacting and non-interacting systems is the contribution of the interactions to the energy per particle, which increases with the number of interacting particles. This is due to the correlation function; as the number of particles in the system increases, the particles will get closer and by that increasing the energy of the system.

5.3 Ground state energy

The exact ground state energy was found for the spherical trap potential for $\alpha = \frac{1}{2}$, as expected from eq. (54) where it gives a stable value for the local energy. The variance further backs this claim, as it is equal to zero.

For the elliptical trap potential the optimal α values are listed in table 6. Here the variance is significantly larger than for the spherical trap, which suggest that the ground state energy was not approximated properly. To improve this, the gradient descent could be calculated several times with different initial guesses leading to better confidence that it did not get stuck in a local minima. Also the dependency on the learning rate should have been varied.

5.4 Statistical errors

It was expected that the variance expressed in eq. (10) would underestimate the statistical error. This was the case in section 4.1, where it underestimated the standard error given by the blocking method. For interacting system this was not the case, and a source of error could be that the amount of data was too small.

6 Conclusion

The variational Monte Carlo did in this project serve some of what it was supposed to for the spherical trap potential, finding the exact ground state energy a non-interacting system of bosons. But as a result of bad implementation of the Metropolis algorithm, and maybe some other parts of the code, it did not give results as expected in form of low variances. As most of the time was spent on trying to work out the code, the results has not been compared to [3] and [1]. The one-body densities was also not computed as there was not found any time for this either.

7 Appendix

7.1 Deriving the analytical expressions

We have the trial wave function:

$$\begin{aligned}\Psi_T(\mathbf{r}) &= \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n, \alpha, \beta) \\ &= \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i<j} f(a, |\mathbf{r}_i - \mathbf{r}_j|) \end{aligned} \tag{59}$$

which is rewritten to get simpler notation.

$$\begin{aligned}\Psi_T(\mathbf{r}) &= \prod_i g(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij})\right) \\ &= \Psi_{SP}(\mathbf{r})\Psi_C(\mathbf{r}) \end{aligned} \tag{60}$$

To find an analytic expression for the local energy and the quantum force it is necessary to calculate the gradient and laplacian Ψ_T:

$$\nabla\Psi_T = [\nabla\Psi_{SP}]\Psi_C + [\nabla\Psi_C]\Psi_{SP} \tag{61}$$

$$\nabla^2\Psi_T = [\nabla^2\Psi_{SP}]\Psi_C + [\nabla^2\Psi_C]\Psi_{SP} + 2[\nabla\Psi_{SP}][\nabla\Psi_C] \tag{62}$$

$$\begin{aligned}\nabla_k\Psi_C &= \nabla_k \prod_i g(\mathbf{r}_i) = \nabla_k(g_1 g_2 \dots g_k \dots g_n) \\ &= [\nabla_k g_k] \prod_{i\neq k} g_i = \frac{\nabla_k g_k}{g_k} g \end{aligned} \tag{63}$$

$$\begin{aligned}\nabla_k^2\Psi_{SP} &= \nabla_k[\nabla_k\Psi_{SP}] \\ &= \nabla_k[\nabla_k g_k] \prod_{i\neq k} g_i = \frac{\nabla_k^2 g_k}{g_k}\Psi_{SP} \end{aligned} \tag{64}$$

$$\begin{aligned}\nabla_k\Psi_C &= \nabla_k \exp\left(\sum_{i<j} u(r_{ij})\right) \\ &= \left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \exp\left(\sum_{i<j} u(r_{ij})\right) \\ &= \left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_C \end{aligned} \tag{65}$$

$$\begin{aligned}\nabla_k^2\Psi_C &= \nabla_k[\nabla_k\Psi_C] = \nabla_k \left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_C \\ &= \left[\sum_{l\neq k} \nabla_k^2 u(r_{kl})\right] \Psi_C + 2\left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_C \\ &= \left[\sum_{l\neq k} \nabla_k^2 u(r_{kl}) + 2\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_C \end{aligned} \tag{66}$$

$$\begin{aligned}\nabla\Psi_T &= \frac{\nabla_k g_k}{g_k}\Psi_T + \left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_T \\ &= \left[\frac{\nabla_k g_k}{g_k} + \sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_T \end{aligned} \tag{67}$$

$$\begin{aligned}\nabla^2\Psi_T =& \frac{\nabla_k^2 g_k}{g_k} g\Psi_C + \left[\sum_{l\neq k} \nabla_k^2 u(r_{kl}) + 2\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_T \\ &+ 2\frac{\nabla_k g_k}{g_k}\left[\sum_{l\neq k} \nabla_k u(r_{kl})\right] \Psi_T \\ =& \left[\frac{\nabla_k^2 g_k}{g_k} + \sum_{l\neq k} \nabla_k^2 u(r_{kl}) + 2\sum_{l\neq k} \nabla_k u(r_{kl})\right. \\ &\left. + 2\frac{\nabla_k g_k}{g_k}\left[\sum_{l\neq k} \nabla_k u(r_{kl})\right]\right] \Psi_T \end{aligned}$$

subsect(6

Rewriting ∇_k using $r_{kl} = |\mathbf{r}_k - \mathbf{r}_l|$

$$\begin{aligned}\nabla_k &= \nabla_k \frac{\partial r_{kl}}{\partial r_{kl}} = \nabla_k r_{kl} \frac{\partial}{\partial r_{kl}} \\ &= \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\partial}{\partial r_{kl}} \end{aligned} \tag{69}$$

gives

$$\nabla_k u(r_{kl}) = \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\partial}{\partial r_{kl}} u(r_{kl}) = \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \tag{70}$$

$$\begin{aligned}\nabla_k^2 u(r_{kl}) =& \nabla_k \left(\frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl})\right) \\ =& [\nabla_k \mathbf{r}_k] \frac{u'(r_{kl})}{r_{kl}} + (\mathbf{r}_k - \mathbf{r}_l)\left[\nabla_k \frac{1}{r_{kl}}\right] u'(r_{kl}) \\ &+ \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \nabla_k u'(r_{kl}) \\ =& \frac{D}{r_{kl}} u'(r_{kl}) + \frac{(\mathbf{r}_k - \mathbf{r}_l)^2}{r_{kl}^3} u'(r_{kl}) \\ &+ \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\mathbf{r}_k - \mathbf{r}_m}{r_{km}} u'(r_{kl}) u'(r_{km}) \\ =& \frac{D-1}{r_{kl}} u'(r_{kl}) + \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} \frac{\mathbf{r}_k - \mathbf{r}_m}{r_{km}} u'(r_{kl}) u'(r_{km}) \end{aligned} \tag{71}$$

Leading to an expression for the laplacian of Ψ_T

$$\begin{aligned}\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) =& \frac{\nabla_k^2 g_k}{g_k} + 2\frac{\nabla_k g_k}{g_k}\left(\sum_{j\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_l)}{r_{kl}} u'(r_{kl})\right) \\ &+ \sum_{l\neq k}\sum_{m\neq k} \frac{(\mathbf{r}_k - \mathbf{r}_l)(\mathbf{r}_k - \mathbf{r}_m)}{r_{kl} r_{km}} u'(r_{kl}) u'(r_{km}) \\ &+ \sum_{l\neq k}\left(u''(r_{kl}) + \frac{D-1}{r_{kl}} u'(r_{kl})\right). \end{aligned}$$

References

[1] J Dubois and H R. Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Physical Review A*, 63, 08 2000.

[2] Marius Jonsson. Standard error estimation by an automated blocking method. *Physical Review E*, 98, 10 2018.

[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, May 2005.