VMC: Effects of Importance Sampling and Jastrow factor on error estimates

Johan Nereng

Department of Physics, University of Oslo, Norway

Spring, 2020

Abstract

    This project aimed to examine using Vartiational Monte Carlo (VMC) in making an estimate, $\bar{E}$, of the ground state energy, E_0, of a trapped, hard sphere Bose gas. The error of the energy estimate was evaluated by using the sample error with "blocking", $\hat{\sigma}$. This method accounts for sample correlation, which was concluded to be necessary as the raw sample error was largely deceptive. Two approaches to the sampling scheme involved in the Metropolis algorithm were used, "Importance Sampling" and the "Brute Force" approach. "Importance sampling" was found to equilbrate the system faster than the "Brute Force" approach, albeit only for suitable values of Δt. Several other parameters, such as the number of particles, N and the value α was observed to affect the suitability of different Δt values. Increases in particle number, N, was generally observed to increase $\hat{\sigma}$; after 2^{18} Monte Carlo cycles, $\hat{\sigma} \sim 10^3$ ($N = 10$) increased to $\sim 10^2$ ($N = 50$) and $\sim 10^{-1}$ ($N = 100$), both with and without the Jastrow factor, which accounts for particle-particle correlation. Incorporating the Jastrow factor was concluded to have an increasing effect on the average estimated energy per particle $\bar{E}/N$ as N increased, from $\bar{E}/N \approx 2.45 \approx$ constant for all values of N, to 4.55 ($N = 50$) and 2.66 ($N = 100$). This impact dependence was further verified by qualitative comparison of one-body densities.

1 Introduction

Variational Monte Carlo (VMC) is widely used to approximate ground states in quantum systems, and may be applied where other methods, such as the mean field Gross-Pitaevskii equation, fails. For VMC methods, the accuracy of the estimated ground energy is largely dependent on factors such as the number of Monte Carlo cycles used, and the number of parameters involved in tuning the simulation. However, one of the most significant error reducing factors is the application of the Jastrow factor, which accounts for particle-particle correlation. In this project the impact of including the Jastrow factor in a hard sphere Bose gas is studied for various numbers of particles, using an improved error estimation method called "blocking". Without "blocking", which addresses the correlation between samples in the computer generated data, the error estimate tends to be overly optimistic, and may - when not careful - lead to results which implies entirely wrong conclusions. Two VMC methods are compared in this project, the "brute force" implementation of the Metropolis algorithm, and in improvement on it's sampling scheme called "Importance Sampling". Where the "brute force" suggests particle moves in entirely random directions, "Importance Sampling" uses the drift force acting on a particle to suggest more likely moves, thus suggesting moves that are more likely to be accepted by the Metropolis algorithm.

First, the system of Bose gas is introduced, including the expressions for the local energy, E_L and drift force F. Then, analytic expressions for E_L and F are derived, both with and without the Jastrow factor. The variational principle, which is the theoretical basis for approximating E_0 by an integral over E_L, is then introduced. Application of this principle relies on approximating the integral by a sum over E_L for a number of system states and their corresponding probability. In order to generate these local energies and probabilities, the Metropolis algorithm is used. The algorithm is based on initiating a random state of N particles, before moving one particle at a time, and either rejecting or accepting each proposed new state according to a sampling rule. Having outlined this algorithm, the improved sampling method called "Importance Sampling" is briefly explained. In order to optimize α when including the Jastrow factor, the Gradient Descent method for minimizing a cost function is outlined, using the local energy as a cost function dependent on α. Finally, the error analysis is presented, including the "blocking" method and the "automated blocking" algorithm from [6].

In order to write this project paper and the code required to produce the results, I used a variety of tools, including: C++, Python 3.7.7, NumPy [9], as well as a number of books, web-pages and articles - of which most are listed under references. All the code required to reproduce the results may be found on my github page .

2 Material and methods

2.1 System: Boson gas

The system consists of N Bosons in an harmonic spherical (S) ($\omega_{ho}^2 = \omega_z^2$) or elliptical (E) ($\omega_{ho}^2 \neq \omega_z^2$) trap (1);

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

, where ω_{ho} and ω_z are trap frequencies in the xy plane and z direction respectively. The characteristic length of the trap is defined $a_{ho} = (\hbar/m\omega_{ho})^{1/2}$, while the ratio of the trap lengths in the xy plane and the z direction is $a_{ho}/a_z = (\omega_z/\omega_{ho})^2 = \sqrt{\lambda}$, which will be used when scaling the system. The

Hamiltonian of the system is given by

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

where the Boson interaction is given by the repulsive force prohibiting particles to come within a distance a of other particles;

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \le a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \tag{3}$$

In order to evaluate the ground-state energy, a trial wave function (TWF) is used (4), where α and β are variational parameters used to tune the wave function towards ground-state (more under a the section on VMC).

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_N, \alpha, \beta) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \left[\prod_{j<k} f(a, |\mathbf{r}_j - \mathbf{r}_k|) \right], \tag{4}$$

Where

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2) \right]. \tag{5}$$

is the non-interactive part of the wave function, and

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

the interactive part, or the Jastrow correlation factor.

2.1.1 Local energy and drift force

When using the Monte Carlo methods described later, the particles will be moved one at a time. After each such move, the local energy (7) will be evaluated, and used to calculate the ground energy of the system.

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

As the algorithms will involve repeatedly calculating the local energy, every reduction of floating point operations (FLOPs) involved in doing so will lead to a significant speed up - which is why an analytical expression for the local energy is desirable. As will be discussed more under importance sampling, the drift force of the particles are also needed;

$$F_i = \frac{2\nabla\Psi_T}{\Psi_T} \tag{8}$$

2.1.1.1 Local energy and drift force: Non-interacting boson sphere

For a spherical harmonic oscillator with no interaction between particles, $a = 0$ and $\beta = 1$, the TWF (4) reduces to;

$$\Psi_T(\boldsymbol{r}) = \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right] \tag{9}$$

Thus simplifying the expression needed for the local energy (7) to;

$$E_L(\mathbf{r}) = \frac{1}{\prod_i g(\alpha, \beta, \mathbf{r}_i)} \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + \frac{1}{2} m\omega_{ho}^2 r^2 \right) \left[\prod_i g(\alpha, \beta, \mathbf{r}_i) \right]$$

which leads to (see Appendix 1: Analytic local energy, non-interacting spherical) the following expression when using natural units, $\hbar = c = 1$, and unity mass, $m = 1$;

$$E_L(\mathbf{r}) = \alpha dN + \left(-2\alpha^2 + \frac{1}{2}\omega_{ho}^2 \right) \sum_i^N r_i^2. \tag{10}$$

With drift force;

$$F_i = \frac{2\nabla\Psi_T}{\Psi_T} = -4\alpha\mathbf{r}_i \tag{11}$$

2.1.1.2 Local energy and drift force: interacting boson sphere with elliptic trap

Next, an analytic expression for the local energy of an elliptical trap with particle interaction, $\beta \neq 1$ and $a \neq 0$, is required. In order to derive it, the TWF (4) is re-written. First, the non-interactive part;

$$g(\alpha, \beta, \mathbf{r}_i) = \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2) \right] = \phi(\mathbf{r}_i) = \phi_i.$$

Secondly, the interactive part;

$$\prod_{i<j} f(r_{ij}) = \exp\left(\sum_{i<j} u_{jk} \right) \tag{12}$$

Where $u_{jk} = \ln f(r_{ij})$, using the shorthand notation $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$. Thus, the WTF (4) is expressed as;

$$\Psi_T(\mathbf{r}) = \left[\prod_i \phi_i \right] \exp\left(\sum_{j<k} u_{jk} \right)$$

Which means that the local energy, using natural units and unity mass as before, (7) can be written as

$$E_L(\mathbf{r}) = \frac{1}{[\prod_i \phi_i] \exp\left(\sum_{j<k} u_{jk} \right)} \sum_i^N \left(\frac{1}{2}\nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) \left[\prod_i \phi_i \right] \exp\left(\sum_{j<k} u_{jk} \right) \tag{13}$$

where (see Appendix 1: Analytic local energy, interacting elliptical)

$$\begin{aligned}
\frac{1}{\Psi_T(\mathbf{r})} \nabla_k^2 \Psi_T(\mathbf{r}) = {} & \frac{\nabla_k^2 \phi_k}{\phi_k} + 2\frac{\nabla_k \phi_k}{\phi_k} \sum_{l \neq k} \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'_{kl} \\
& + \sum_{j \neq k} \sum_{l \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_l)(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj} r_{kl}} u'_{kj} u'_{kl} \\
& + \sum_{l \neq k} \left(u''_{kl} + \frac{2}{r_{kl}} u'_{kl} \right)
\end{aligned} \tag{14}$$

Thus

$$E_L(\mathbf{r}) = \frac{1}{2} \sum_{k=1}^{N} \left(\frac{\nabla_k^2 \phi_k}{\phi_k} + 2 \frac{\nabla_k \phi_k}{\phi_k} \sum_{l \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'_{kl} + \left(\sum_{j \neq k} \frac{(\boldsymbol{r}_k - \boldsymbol{r}_j)}{r_{kj}} u'_{kj} \right)^2 + \sum_{l \neq k} \left(u''_{kl} + \frac{2}{r_{kl}} u'_{kl} \right) + V_{ext}(\boldsymbol{r}_i) \right)$$

(15)

Where

$$\frac{\nabla_k \phi_k}{\phi_k} = -2\alpha(x_k \vec{e}_1 + y_k \vec{e}_2 + \beta z_k \vec{e}_3),$$

$$\frac{\nabla_k^2 \phi_k}{\phi_k} = -2\alpha(d - 1 + \beta) + 4\alpha^2(x_k + y_k^2 + \beta^2 z_k^2),$$

$$\frac{\partial u_{kl}}{\partial r_{kl}} = \frac{a}{r_{kl}(r_{kl} - a)},$$

$$\frac{\partial^2 u_{kl}}{\partial r_{kl}^2} = \frac{a^2 - 2ar_{kl}}{r_{kl}^2 (r_{kl} - a)^2}$$

And drift force;

$$F_k = \frac{2\nabla \Psi_T}{\Psi_T} = 2 \left(\frac{\nabla_k \phi_k}{\phi_k} + \sum_{l \neq k} \nabla_k u_{kl} \right)$$

Where $\nabla_k u_{kl} = \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} \frac{\partial u_{kl}}{\partial r_{kl}}$

2.1.2   Scaling the system

As in [8] and [1], the system is scaled using $a/a_{ho} = 0.0045$, with units of length of a_{ho}, $r \rightarrow r/a_{ho}$ and energy units of $\hbar \omega_{ho}$. This means that $\nabla_i \rightarrow \frac{1}{a_{ho}} \nabla_i$, thus the Hamiltonian (2) can be written as;

$$H = \sum_{i}^{N} \left(\frac{-\hbar^2}{2m} \frac{1}{a_{ho}^2} \nabla_i^2 + \frac{1}{2} m \omega_{ho}^2 a_{ho}^2 (x^2 + y^2 + \frac{\omega_z^2}{\omega_{ho}^2} z^2] \right) + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j)$$

$$= \sum_{i}^{N} \left(\frac{-\hbar^2}{2m} \frac{m \omega_{ho}}{\hbar} \nabla_i^2 + \frac{1}{2} m \omega_{ho}^2 \frac{\hbar}{m \omega_{ho}} (x^2 + y^2 + \frac{\omega_z^2}{\omega_{ho}^2} z^2) \right) + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j)$$

(16)

$$= \sum_{i}^{N} \frac{1}{2} \left(-\nabla_i^2 + x^2 + y^2 + \lambda^2 z^2 \right) + \sum_{i<j}^{N} V_{int}(\mathbf{r}_i, \mathbf{r}_j)$$

2.2   Variational Monte Carlo

There are various methods available to circumnavigate having to directly solve the Schrödinger equation in order to find the eigenpair of the system we are looking for. An important underlying principle for the methods employed in this project is the variational principle [3][p.256.];

$$E_0 \leq E[H] = \langle H \rangle = \langle \Psi | H | \Psi \rangle$$

(17)

This principle states that given a Hamiltonian, H, the ground-state energy, E_0, is upper bound by the expectation value $\langle H \rangle$, and holds for any normalized function Ψ. Thus, by using the Hamiltonian

of the Bose gas, and trial wave function described earlier with variational parameters $\boldsymbol{\alpha}$, it possible to hone in on the minimal value of $\langle H \rangle$ and the desired ground-state energy, E_0, by solving the integral [1];

$$E[H, \boldsymbol{\alpha}] = \langle H \rangle = \frac{\int d\boldsymbol{r}\, \Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) H(\boldsymbol{r}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})}{\int d\boldsymbol{r}\, \Psi_T^*(\boldsymbol{r}, \boldsymbol{\alpha}) \Psi_T(\boldsymbol{r}, \boldsymbol{\alpha})}. \tag{18}$$

Direct dependence on $\boldsymbol{\alpha}$ is omitted from the rest of this project, but is included here to show that the energy also depends on the variational parameters.

Instead of solving this multi-dimentional integral analytically, which for more than a single particle becomes complicated, if not impossible, I've used Monte Carlo integration. Monte Carlo integration [4][Ch. 3] is a stochastic method aimed at evaluating a deterministic and high dimensional integral. This involves drawing a sequence of $i = 1, ..., M$ random evens with a probability P_i, multiplying each event by it's probability, and taking the average - which will converge to the desired integral. The following shows this method applied to evaluating the ground energy of the Boson gas, based on local energy samples;

$$E[H] = \int P(\boldsymbol{r}) E_L(\boldsymbol{r}) \approx \frac{1}{M} \sum_{i=1}^{M} P(\boldsymbol{r}_i) E_L(\boldsymbol{r}_i) \tag{19}$$

, where $E_L(\boldsymbol{r}_i)$ (7) is the local energy of the system in configuration $\boldsymbol{r}_i$, and $P(\boldsymbol{r}_i)$ the probability of that local energy.

$$P(\boldsymbol{r}) = \frac{|\Psi_T(\boldsymbol{r})|^2}{\int |\Psi_T(\boldsymbol{r})|^2 d\boldsymbol{r}} \tag{20}$$

2.2.1 Metropolis algorithm

The Metropolis algorithm [4][p.86-88] is a highly adaptable method which generates a sequence of random samples from a probability distribution. The method makes iterative changes to a single degree of freedom and either accepts or rejects each change. By using the relative probability between samples, the method does not rely on directly sampling from a probability distribution, making the methods well suited for problems which involve complex probability distributions. The probability of the system being in state i at time $t + \epsilon$ can be expressed through the probabilities of all states at time t; $P_i(t + \epsilon) = \sum_j w(j \to i) P_j(t)$, where $w(j \to i)$ is the transition probability from j to i. (20) can be used to find the probability density, but the transition probabilities are unknown. These probabilities can be expressed as $w(j \to i) = A(j \to i) T(j \to i)$, where T is the likelihood of making a transition, and A the likelihood of that transition being accepted. The Metropolis algorithm works by initializing the system is a certain state, then using a random walker to suggest new states. By assuming $T(j \to i) = T(i \to j)$, or what is known as detailed balance [4][p.86], one obtains;

$$\frac{P_i}{P_j} = \frac{W(j \to i)}{W(i \to j)} = \frac{A(j \to i) T(j \to i)}{A(i \to j) T(i \to j)} = \frac{A(j \to i)}{A(i \to j)}.$$

Using the probability ratio means that normalization is unnecessary, but more importantly this ratio also tells whether or not the random walker is moving into a region of high probability. Since the acceptance probability is unknown, brute force Metropolis uses $A(j \to i) = min(1, \frac{P_i}{P_j})$. The adaptation of the algorithm used in this project reads as;

---

[1]Since Ψ_T is not normalized, but normalizable, the favefunction $\dfrac{\Psi_T(\boldsymbol{r})}{\sqrt{\int d\boldsymbol{r}\, \Psi_T^*(\boldsymbol{r}) \Psi_T(\boldsymbol{r})}}$, is normalized

---

Algorithm: Brute Force Metropolis

1. Initialize algorithm

   - Set the number of Monte Carlo cycles, M, and step length, l
   - Set variational parameters $\boldsymbol{\alpha}$
   - Set particle positions, $\vec{r}$
   - Set energy $E = E_L(\boldsymbol{r})$

2. Propose new configuration

   - Select random particle
   - Randomly move particle up to a threshold l. Store proposed particle position $\vec{r}*$

3. Evaluate proposal

   - Calculate $w = \frac{P(\vec{r}*)}{P(\vec{r})} = \frac{|\Psi_T(\vec{r}*)|^2}{|\Psi_T(\vec{r})|^2}$
   - Generate a random number q between 0 and 1.
   - If $w \geq q$, accept proposal, calculate E_L and set $E = E + E_L$. Else, reject proposal.

4. If the number of cycles from step 2 to 4 is less than M, go to step 2. Else calculate $\langle H \rangle = \frac{E}{M}$ and end simulation.

2.2.2 Importance sampling

A problem with the brute force methodology described above is that one may end up with a lot of proposed configurations from regions of relatively lower probability, or in other words, wasted cycles. A method that can be used to reduce the wastage is Importance sampling, a variance reducing technique which instead of suggesting random moves, suggest moves based on the current configuration of the system.

The new particle position is based on the stochastic differential equation

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \tag{21}$$

where F is the drift force, D the diffusion constant (in atomic units, $D = 1/2$) , and η is random noise. Based on this, using Euler's method, the new position, $\boldsymbol{r}*$ of the randomly chosen particle is suggested by

$$\boldsymbol{r}* = \boldsymbol{r} + DF(\boldsymbol{r})\Delta t + \xi\sqrt{\Delta t} \tag{22}$$

where $\boldsymbol{\xi}$ is a vector of Gaussian random variables, and Δt the time step. To accommodate this, the ratio of probabilities used in the brute force method needs to be altered:

For a time-dependent probability density, $T(x, t)$, the one-dimensional time evolution may be modeled using the Fokker-Planck equation, which comes from fluid mechanics;

$$\frac{\partial T}{\partial t} = D(\frac{\partial^2}{\partial x^2} - F)T(x, t). \tag{23}$$

The most likely state corresponds to having a constant transition probability, $dT/dt = 0$, so;

$$\frac{\partial^2}{\partial x_i^2}T(x, t) = T\frac{\partial}{\partial x_i}F_i + F_i\frac{\partial}{\partial x_i}T \tag{24}$$

Inserting $F = g(x)\partial T/\partial x$;

$$\frac{\partial^2}{\partial x_i^2}T(x,t) = T\frac{\partial g}{\partial x_i}(\frac{\partial T}{\partial x_i})^2 + Tg\frac{\partial^2 T}{\partial x_i^2} + g(\frac{\partial T}{\partial x_i})^2 \tag{25}$$

means that $g = \frac{1}{T}$, and in the case of $P = \Psi$, the drift force is

$$F = 2\frac{1}{\Psi}\nabla\Psi. \tag{26}$$

The solution to the Fokker-Planck equation using this drift force can be approximated by the Green's function;

$$G(\boldsymbol{r}, \boldsymbol{r}*, \Delta t) = \frac{1}{(4\pi D\Delta t)^{dN/2}}exp\left(-\frac{(\boldsymbol{r} - \boldsymbol{r}* - D\Delta t F(\boldsymbol{r}*))^2}{4D\Delta t}\right) \tag{27}$$

,where Δt is time time step, d the number of dimensions. Having obtained an expression which approximates the transition probability, detailed balance is no longer required. This means that an improved approach to the acceptance probability can be written as

$$q = \frac{G(\boldsymbol{r}, \boldsymbol{r}*)|\Psi_T(y)|^2}{G(\boldsymbol{r}*, \boldsymbol{r})|\Psi_T(x)|^2} \tag{28}$$

Meaning that only the ratio of the exponentials in the Green's function needs to be calculated.

2.2.3 Gradient Descent

By including the Jastrow factor, or the correlation part of the wave equation, the ground energy of a system is expected to increase when compared to a corresponding single-particle evaluation. Because of the particle interaction, $\alpha = 0.5$ will necessarily no longer be the correct parameter value for all values of N. As such, I have opted to implement gradient descent on iterative runs in order to better approximate the value of α.

As the desired outcome is to find an α s.t the energy is minimized, the energy may be viewed as a cost function to be minimized. In general, the minimization of a cost function may in some cases be carried out analytically, in this case however this is not feasible. For problems such as this one, numerical methods for obtaining the desired parameters $\boldsymbol{\alpha}$ is instead applied. One such method is the gradient descent (GD) (29) [7][p.247], also known as steepest descent. When utilizing GD, one makes an initial guess for $\boldsymbol{\alpha}$, evaluates the gradient, $\boldsymbol{g}$ of the cost function in order to obtain an estimate closer to the desired minimizing value. This process is then iteratively repeated till the sufficient convergence of the coefficients is reached. In order to not overshoot the $k + 1$'th estimate, a *learning rate* of η is applied to the gradient. The value of the learning rate can either be fixed, usually by trial and error, such that it works well for the problem at hand, or it may be adapted to each step.

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - \eta_k\boldsymbol{g}(\boldsymbol{\beta}_k) \tag{29}$$

In order to implement this method for the energy of the system, $\boldsymbol{g}(\boldsymbol{\alpha}) = \nabla_\alpha\langle E_L\rangle$ is required, or the gradient of the local energy w.r.t α. By applying the chain rule to (18), the following expression is obtained;

$$\nabla_\alpha\langle E_L\rangle = 2\left(\langle\frac{\bar{\Psi}_\alpha}{\Psi[\alpha]}E_L[\alpha]\rangle - \langle\frac{\bar{\Psi}_\alpha}{\Psi[\alpha]}\rangle\langle E_L[\alpha]\rangle\right)$$

Where $\bar{\Psi}_\alpha = \frac{d\Psi[\alpha]}{d\alpha} = \left(-\sum_i(x_i^2 + y_i^2 + \beta z_i^2)\right)\Psi[\alpha]$. In a computational perspective, this means accumulating the values of $\sum -(x_i^2 + y_i^2 + \beta z_i^2)$, and it's product with E_L at each cycle, then calculating the averages after all cycles are completed. Thus, as higher numbers of cycles is expected to bring $\bar{E}$ closer to μ_E, higher numbers of cycles should also produce a more accurate α gradient.

2.3 Error analysis

The ground-state energy of the system is estimated by the sample mean of the local energies over M cycles (or the fraction left after equilibration);

$$\bar{E} = \frac{1}{M}\sum^M E_L(\vec{r}_i) \tag{30}$$

This means that the error of the sample mean, or sample error (squared) (31), coincides with the estimation error of E_0;

$$err_{\bar{E}}^2 = var(\bar{E}) = \frac{1}{M}\sum^M (E_L(\vec{r}_i) - \mu_{\bar{E}_L})^2, \tag{31}$$

where $\mu_{\bar{E}_L}$ is the true mean of E_L. Assuming that the samples are independent of one another, then by the law of large numbers[2], the difference between the sample mean and the true mean will decreases as the number of samples increase. For a sufficiently large number of Monte Carlo cycles, this would reduce the sample error arbitrarily close to zero, as long as $\mu_{\bar{E}_L} = E_0$, which depends on having chosen the correct variational parameters. Assuming that $\bar{E} = \mu_{\bar{E}_L} = E_0$, the error of the sample mean can be directly compared to the variance of the ground-energy; In general, $H\Psi = E_0\Psi$ when Ψ is the exact ground-state (the same holds for any eigenpair) of the system. This means that $\langle H^n \rangle = \langle \Psi | H^n | \Psi \rangle = E^n$, which also means that the exact wave function has variance of zero;

$$var(E) = \langle H^2 \rangle - \langle H \rangle^2 = E^2 - (E)^2 = 0 \tag{32}$$

In other words, the sample error should be (machine precision) equal to the energy variance, which is zero, if the variational parameters are correct and the number of cycles sufficiently large. If the paramters are not correct, then the closer the parameters are to the correct values, the smaller the sample error. The magnitude of this sample error however, will only be correct if the assumption of independent samples holds, which it in this case does not.

2.3.1 Error estimate of correlated energy samples; the "blocking" method

Any system state, other than the initial, is directly dependent on the previous configuration of the system through the use of a random walker. Because of this, the associated energy samples are correlated. The error estimate (31) above does not reflect this correlation, which means that the error estimate will be less accurate. In order to remedy this, I have chosen to use the "blocking" or "bunching" meothd [2], which benefits from being more accurate the larger the sample size, unlike for example bootstrapping.

As the series of energy samples from the VMC is ordered as a function of time, it is a time series of finite length. In general, a time series of n samples can be used to form an n-vector or n-tuple, $\vec{X}$, where (X_i) denotes the i'th sample in the series. In order to estimate the variance of the sample mean, $var(\bar{X})$, by use of the "blocking" method, the time series is divided into blocks of increasing size. By combining these transformations with theorems on strictly stationary time series an automated

---

[2] $lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n x_i p(x_i) = \mu_x$

"blocking" method (explained in detail in [6]) may be applied to the VMC sample series. Pivotal for this method is the distribution of the quantity M (36), which depends on a small number different statistical quantities from the transformed series. Using this, the automated "blocking" method may be applied to the VMC sample series. Below is a brief outline of the automated "blocking" method.

Starting with one block per sample, followed by two samples per block and so on, the sample variance, $\hat{\sigma}^2$ is calculated. As these blocks increase in size, $\hat{\sigma}^2$ of the time series gradually unravels from the sample covariance. From $\boldsymbol{X}_0 = \boldsymbol{X}$, the subsequent "blocking" transformation, $\boldsymbol{X}_0 \to \boldsymbol{X}_1$, produces a new time series $\boldsymbol{X}_1$ with length $n_1 = n/2$ [3]. The subsequent "blocking" number k, continues in the same fashion, where each transformation is achieved through

$$(X_i)_k = \frac{1}{2}(X_{2i-1})_{k-1} + (X_{2i})_{k-1} \tag{33}$$

for $i = 1, .., n_k$, where n_k is the length of the $\boldsymbol{X}_k$. These transformations are carried out until only one block remains.

The estimate for the variance of the sample mean can be expressed as $var(\bar{X}) = \frac{\sigma_k}{n_k} + \epsilon_k$, where ϵ_k is the truncation error, which after a certain number of transformations is expected to become constant. Finding the lowest k, such that $\epsilon_k = \epsilon_{k+1}$ is the goal of the last step. By calculating M (36), where d comes from $n = 2^d$, the smallest k such that $M_k \leq q_{d-k}(1-\alpha)$ gives which k to use when calculating $v\hat{a}r(\bar{x}) = \sigma_k^2/n_k$ to use as an estimate for the sample mean [4], with confidence interval $(1-\alpha)$. Because M requires both σ_k, and $\gamma_k(1)$, these quantities should be calculated after each transformation.

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{i=1}^{n-h} (X_i - \bar{X})(X_{i+h} - \bar{X}) \tag{34}$$

which is a measure of the h-order $(h = |i - j|)$ covariance of a time series series with length n.

$$\hat{\gamma}_k = \frac{1}{4} \tag{35}$$

$$M_j = \sum_{k=j}^{d-1} \frac{n_k \left[(n_k - 1)\hat{\sigma}_k^2/(n_k^2) + \hat{\gamma}_k(1)\right]^2}{\sigma_k^4} \tag{36}$$

When tailored to finding an estimate for the variance of $\bar{E}_L$, from the VMC calculations, the automatic "blocking" algorithm reads as;

---

[3] In the case where a time series is not divisible by two, a random sample is omitted.

[4] $q_{d-k}(1-\alpha)$ may be found in a table for the χ distribution.

Algorithm: The automated "blocking" method

1. Initialize algorithm
   - Set α
   - Estimate $var(\bar{E})$
   - Set $i = 0$

2. Iterative "blocking" while size of $E_{i+1} \geq 2$:
   - If $n_i \% 2 \neq 0$, remove a random energy sample from E_i
   - Compute $\hat{\sigma}_i^2, \hat{\gamma}_i(1)$
   - Transform data $\vec{E}_i \to \vec{E}_{i+1}$, and set $i = i + 1$

3. Calculate final values
   - Compute M_j from $\hat{\sigma}_j^2, \hat{\gamma}_j(i)$
   - Find smallest k such that $M_k \leq q_{d-k}(1-\alpha)$
   - Calculate $v\hat{a}r(\bar{E}) = \hat{\sigma}_k^2/n_k$

---

where E_i is the time series of energy samples after a number of i "blocking" transformations.

3 Results

3.1 Non-interacting Boson gas

The algorithms were implemented with C++, based on a class structure made by Morten Ledum[5]. Below, the Metropolis algorithm with both the "Brute Force" method and "Importance Sampling" is examined, using both a one-dimensional system and a three-dimensional system with varying numbers of particles from $N = 1, 10, 50$, to $N = 100$.

For one particle in one dimension the exact ground energy is obtained at $\alpha = \frac{m\omega}{2\hbar}$ or $\alpha = \omega/2 = 0.5$, when using natural units. Looking to (10), this means that the estimated ground energy, $\bar{E}$, for $\omega = 1$, $N = 1$, $d = 1$, should be exactly equal to $E_L(\alpha = \frac{m\omega}{2\hbar}) = \frac{\omega dN}{2} = 0.5$. Table 1 shows the simulation results from the brute force method, using $N = 1$, $d = 1$, $l = 0.5$ after $M = 2^{21}$ Monte Carlo cycles with an equilibrium fraction of 0.3. As α tends towards 0.5, the table clearly indicates that the estimated energy tends towards 0.5. Additionally the table shows that the estimation error of $\bar{E}$, σ, also decreases towards zero as α tends towards 0.5, as was described in the section on error analysis. Comparing the error estimates with and without blocking, σ and $\hat{\sigma}$, shows that the error without blocking is underestimated for all values of α (except for the exact value of 0.5), which is consistent with the predicted behavior of the sampling error of correlated data.

In the $N = 10, d = 1$ case, shown in appendix 1, $\bar{E}$ and $\hat{\sigma}$ are similar to those produced in the $N = 1$ case for all values of α, only scaled up ten times - which is not surprising as the particles are non-interacting. Looking at (10), it is also clear that $\alpha = 0.5$ should indeed give an energy of 5.0 in this case. In terms of CPU time, $N = 10$ used on average 2661 ms while $N = 1$ used 648 ms. The reason why the increased time is not also ten fold can be explained by the computational overhead.

---

[5]The class structure can be found on Morten Ledum's github page

Further increase in particle numbers for $d = 1$ does not show any trends of interest other than what is already mentioned, and are therefor not included.

α	$\bar{E}$	σ	$\hat{\sigma}$	$t(ms)$
0.30	0.568	3.12×10^{-4}	1.87×10^{-3}	-
0.35	0.533	2.13×10^{-4}	1.19×10^{-3}	-
0.40	0.513	1.33×10^{-4}	7.04×10^{-4}	-
0.45	0.503	6.17×10^{-5}	3.10×10^{-4}	-
0.50	0.500	0.00	0.00	-
0.55	0.502	5.63×10^{-5}	2.65×10^{-4}	-
0.60	0.508	1.08×10^{-4}	6.97×10^{-4}	-
0.65	0.517	1.56×10^{-4}	6.87×10^{-4}	-
0.70	0.523	2.01×10^{-4}	8.66×10^{-4}	-
Average	-	-	7.09×10^{-4}	648

Table 1: Non-interacting. One particle in one dimension after 2^{21} cycles, with a 0.3 equilibrium fraction, for various values of α.

3.1.1 Importance sampling

Having established a working initial implementation, simulations using three dimension, $d = 3$, and $N = 10$ are used to highlight some of the differences between using the brute force method and the importance sampling method, as well as effects of δt on estimation error.

Δt	$\hat{\sigma}$	A
0.005	3.68×10^{-2}	1.000
0.050	1.31×10^{-2}	0.995
0.070	1.08×10^{-2}	0.985
0.100	9.11×10^{-3}	0.985
0.200	5.80×10^{-3}	0.960
0.300	5.36×10^{-3}	0.925
0.400	4.34×10^{-3}	0.884
0.500	4.92×10^{-3}	0.839
0.600	4.83×10^{-3}	0.791
1.000	5.20×10^{-3}	0.586
1.500	5.09×10^{-3}	0.366

Table 2: Non-interacting. Importance sampling: Averages over $\alpha = 0.3, 0.35, ..., 0.7$ for various values of δt using $N = 10$, $d = 3$, and 2^{21} cycles with a 0.3 equilibrium fraction.

Table 2 shows different values of $\hat{\sigma}$ and acceptance rates, A for a range of α values using the importance sampling method. The table suggests that a sufficiently small Δt will give $A = 1.0$.

However, as Δt is increased, both A and $\hat{\sigma}$ decrease. In the case of $\hat{\sigma}$, the value starts increasing again after $\Delta t = 0.4$. Looking at the expression for q in the algorithm, $G(\boldsymbol{r}, \boldsymbol{r}*)$ will be evaluated to 1 regardless of the suggested position if Δt is small enough. Furthermore, sufficiently small values of Δt will also lead to minuscule particle movement. This likely means both that moves which actually increase energy are more likely to be accepted, and that convergence towards a minimal energy state takes a relatively long time in terms of cycles.

Δt	$\bar{E}$	σ	$\hat{\sigma}$	A
$0.01(\alpha = 0.3)$	57.3478	3.314×10^{-3}	4.36×10^{-2}	0.999
$0.05(\alpha = 0.3)$	56.7378	3.109×10^{-3}	1.92×10^{-1}	0.998
$0.10(\alpha = 0.3)$	56.6193	3.100×10^{-3}	1.29×10^{-1}	0.997
$0.30(\alpha = 0.3)$	56.5863	3.123×10^{-3}	8.21×10^{-2}	0.983
$0.50(\alpha = 0.3)$	56.6613	3.118×10^{-3}	6.97×10^{-2}	0.963
$0.01(\alpha = 0.45)$	50.3799	6.479×10^{-4}	7.14×10^{-3}	0.999
$0.05(\alpha = 0.45)$	50.2770	6.136×10^{-4}	2.85×10^{-2}	0.998
$0.10(\alpha = 0.45)$	50.2603	6.133×10^{-4}	2.04×10^{-2}	0.994
$0.30(\alpha = 0.45)$	50.2723	6.172×10^{-4}	1.44×10^{-2}	0.968
$0.50(\alpha = 0.45)$	50.2767	6.172×10^{-4}	1.18×10^{-2}	0.932

Table 3: Non-interacting. Importance sampling: Various values of δt using $N = 100$, $d = 3$, for $\alpha = 0.3$ and $\alpha = 0.34$, after 2^{21} cycles with a 0.3 equilibrium fraction. Average $\hat{\sigma}(\alpha = 0.3) = 1.03 \times 10^{-1}$, average $\hat{\sigma}(\alpha = 0.45) = 1.65 \times 10^{-2}$

In table 3 values of σ, $\hat{\sigma}$, and A is shown for a selection of Δt values for $N = 100$, $d = 3$, using importance sampling. Instead of averaging over a specter of α values, the values were produced with $\alpha = 0.3$ (an incorrect α-value) and $\alpha = 0.45$ (also incorrect, but closer to the correct value of 0.5). The results are from a single run, meaning that any conclusions based on these results are not robust - a more thorough study using multiple runs with different seeds are required to remove noise from the observations. However, as the transformation between σ and $\hat{\sigma}$ is predictable, and not stochastic, observations pertaining to it may be more robust. With that it mind, it is evident that the error estimate without "blocking" is large deceptive. The two values of $\sigma(\Delta t = 0.01)$ are marginally smaller, $\sim 10^{-6}$ for $\alpha = 0.3$ and $\sim 10^{-5}$ for $\alpha = 0.45$, than the other values of $\sigma(\Delta t)$. After applying "blocking" however, the corresponding differences in values have increased in magnitudes to $\sim 10^{-1}$ and $\sim 10^{-2}$. Based on this, I conclude that comparisons between error estimates without "blocking" or other similar methods, is irrelevant when studying the effect of Δt on estimation error. This causes some challenges for further in-depth study in this project, as it's VMC implementation is in C++, and the "blocking" technique is done post-simulation in Python. In order to more precisely study the importance sampling method, VMC algorithm implementation should be carried out with a in "blocking" method in the class structure.

For both values of α, $\Delta t = 0.01$ gives the lowest $\hat{\sigma}$, while the highest values are for $\Delta t = 0.05$ and 0.1. This is distinctly different from the trends seen in table 2, where low values of Δt give strictly higher $\hat{\sigma}$ - possibly suggesting that the effects of Δt are dependent on the number of particles, or the ratio between cycles and particles. Comparing $\bar{E}$ of the different Δt values, shows that $\Delta T = 0.01$, despite it's lower $\hat{\sigma}$, has produced a significantly different energy estimate than the other values. Especially in the $\alpha = 0.3$ case. This observation highlights the fact that the error estimates are based on sample variances; an estimated energy deviating far from the desired energy, is still an undesirable

estimate, even if it has a low sample variance. These apparent inconsistencies in the effects of Δt, on $\hat{\sigma}$ and $\bar{E}$ may imply that what is to be considered a "good" Δt value, is dependent on how close α is to yielding $\bar{E} = E_0$, which is consistent with the observations above on the change in magnitudes of differences between error estimates before and after "blocking". This also warrants further study and testing on the subject than what is done in this project.
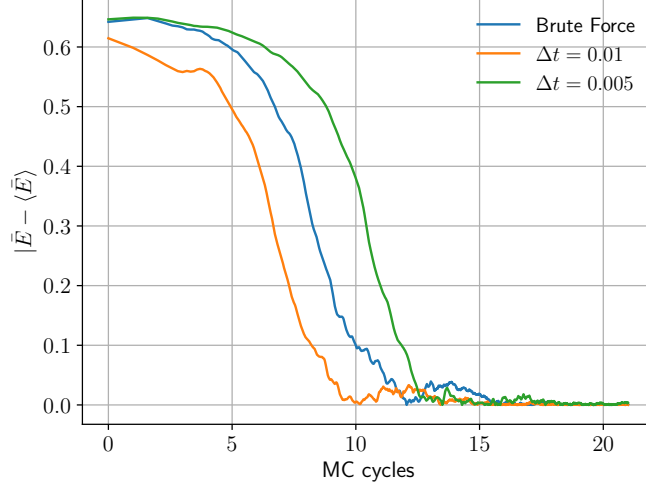

Figure 1: Non-interacting. Average energy deviation from mean over 10 runs of $N = 10$, $d = 3$, $\alpha = 0.45$, for brute force, and importance sampling method with $\Delta t = 0.01$ and $\Delta t = 0.005$, over as a function of Monte Carlo cycles

Figure 1 shows a comparison of the average energy deviation as a function of Monte Carlo cycles for both brute force and importance sampling. The figure indicates that importance sampling using $\Delta t = 0.01$ reaches a stable value of $\bar{E}$ after fewer cycles than the brute force method. However, $\Delta t = 0.005$ requires still more cycles than brute force force method. This suggests that importance sampling with the suitable value of Δt may be used to decrease the number of cycles before starting to sample the local energy. On the other hand, unsuitable values of Δt may lead to worse performance in terms of required equilibrium cycles, compared to the brute force approach. Table 4 shows the corresponding values for the estimation error, acceptance rate and time. Both the brute force method and the importance sampling method at $\Delta t = 0.1$ show $\hat{\sigma}$ of order 10^{-3}, while the $\Delta t = 0.005$ has an order of magnitude larger average $\hat{\sigma}$ value. In this last case this is likely due to the trend described earlier about lower values of Δt. Comparing $\Delta t = 0.1$ and the brute force method, the brute force method comes out on top, owing to the fact that the algorithm is faster. However, this does not imply that the brute force method is to be preferred over importance sampling in general, both because $\Delta t = 0.1$ might not be the best suited value for the tested configuration and because similar tests for more complex systems are also needed, as well as other values of α.

On the basis of the observations and discussions above, I am is unable to make any exact conclusions on the consequences of the choice of Δt, other than that the number of particles, N, and the value of α, appear to effect the dependence of $\hat{\sigma}$ on Δt. Further more, the value of Δt appears to effect the correlation time of the samples, to the extent where σ as a quality measurement may be misleading. The value of $\bar{E}$ as well as $\hat{\sigma}$ must be evaluated when determining if one has used the

Method	$\bar{E}$	$\hat{\sigma}$	$\bar{A}$	$\bar{t}$ [ms]
Brute force	15.1	5.49×10^{-3}	0.737	7633
Importance sampling ($\Delta t = 0.1$)	15.1	4.08×10^{-3}	0.8987	9503
Importance sampling ($\Delta t = 0.005$)	15.1	1.64×10^{-2}	0.999	9599

Table 4: Non-interacting. Comparison of brute force and importance sampling averaged over 10 runs of $N = 10$, $d = 3$, $\alpha = 0.45$, at 21^3 cycles with a 0.3 equilibrium fraction.

correct Δt, and simulations requiring high precision should include some variations on Δt. When compared to the brute force method, importance sampling appears to equilibrate after fewer MC cycles than what than the brute force method, but only if Δt is at a suitable value, which comes at the cost of increased CPU time.

3.2 Interacting Bose gas in elliptical trap

Having established a working implementation for multiple particles in three dimensions both using the Importance Sampling and Brute Force methods, I now move on to results from the interacting Bose gas implementation. Including particle interaction in the simulations will necessarily mean that $\alpha = 0.5$ no longer produces the lowest energy for all particle systems. As such, I have chosen to closer examine the effects of using Gradient Descent (GD) to approximate α (initial testing of the implementation of the GD method is shown to be performing as expected in figure 4 in appendix 1). After the discussion on GD, I move on to comparing interacting and non-interacting systems for various numbers of particles, before presenting a visualization of the of wave-function in terms of one-body densities. All the presented results below are produced using Importance Sampling, and the number of dimensions, d, is set to 3. Unless stated otherwise, $\lambda = \sqrt{8}$.

Figure 2 shows the iterative evolution of α when using gradient descent for different numbers of MC cycles. It is clear from the graph that the 2^{21} cycles case has less variance in the produced values of α than the two other values, and that 2^{15} has the highest variance in α values. This may indicate that higher numbers of MC cycles give more accurate α-gradients, as one should expect. In order to further test this, the by-eye mean of α for 2^{21} cycles and 2^{15} and 2^{18} cycles, $\alpha = 0.496$ and $\alpha = 0.4965$ respectively, were used in 10 runs with 2^{21} cycles (table 5). The table shows that $\hat{\sigma}$ for $\alpha = 0.4960$ is of order 10^{-5} smaller than for $\alpha = 0.4965$, simultaneously $\bar{E}$ is smaller in the $\alpha = 0.4965$ case, by a magnitude of 10^{-3}. This inconsistency is likely caused by having used too few experiments, that the parameter values are too similar to produce easily interpreted results, and/or that the accuracy limit of the current implementation is reached before discrepancies of order 10^{-4} in α becomes evident.

α	$\bar{E}$	$\hat{\sigma}$
0.4960	24.3993	9.99×10^{-4}
0.4965	24.3986	8.69×10^{-4}

Table 5: Interacting. Comparison between α values based on figure 2, of averages over 10 runs, using 2^{21} MC cycles and $\Delta t = 0.1$.
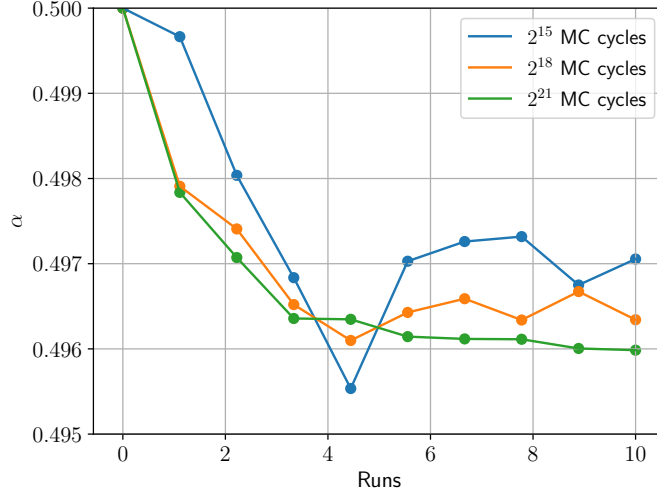
Figure 2: Interacting. Gradient Descent with adaptive learning rate $\eta_{k+1} = 0.9\eta_k$, on α for $N = 10$ using $\Delta t = 0.01$ for various numbers of MC cycles (with a 0.3 equilibration factor).

N	Interacting				Non-interacting			
	$\bar{E}$	$\bar{E}/N$	$\hat{\sigma}$	time[s]	$\bar{E}$	$\bar{E}/N$	$\hat{\sigma}$	time[s]
$10(\alpha = 0.496)$	24.399	2.440	2.87×10^{-3}	4	24.349	2.435	9.01×10^{-3}	1
$50(\alpha = 0.457)$	127.267	2.545	4.05×10^{-2}	50	123.312	2.466	7.18×10^{-2}	4
$100(\alpha = 0.494)$	266.150	2.662	1.33×10^{-1}	190	244.630	2.446	1.6×10^{-1}	9

Table 6: Comparison between interacting and non-interacting systems using α from GD and $\beta = \gamma = 2.82843$. The values are averages over 10 runs, using 2^{18} MC cycles and $\Delta t = 0.01$.

Table 6 shows the estimated energy $\bar{E}$, estimated energy per particle $\bar{E}/N$, error estimate using blocking, $\hat{\sigma}$, and CPU time in seconds for different numbers of particles, with and without particle interaction, using elliptical harmonic oscillator potential, $\beta = \gamma = 2.82843$. Firstly, it is clear that the ground energy of the elliptical trap is substantially higher than for the spherical trap, which is to be expected when for example looking at the Hamiltonian of the scaled system (16). Secondly, as N increases, the table shows that the error estimate increases from $\hat{\sigma} \sim 10^{-3}$ ($N = 50$) to $\sim 10^{-2}$ ($N = 50$) and $\sim 10^{-1}$ ($N = 100$) both with and without the Jastrow factor. However, this does not dilute the validity of the following observations on $\bar{E}/N$ as $\hat{\sigma}$ scales to N at a slower rate. Comparing the different values of N in the non-interacting case, it is clear that the $\bar{E}/N \approx 2.45 \approx$ constant due to the system being a superposition of the single-particle systems of each particle. When also encompassing particle interaction however, $\bar{E}/N$ increases as the number of interactions on each particle is strengthened, from $\bar{E}/N \approx 2.55$ in the case of $N = 50$ to $\bar{E}/N \approx 2.66$ for $N = 100$. As such, it is evident that the inclusion of the Jastrow factor is increasingly important for higher values of N. Even though the error estimate increases drastically from $\hat{\sigma} \sim 10^3$ ($N = 10$) to $\sim 10^2$ ($N = 50$) and $\sim 10^{-1}$ ($N = 100$), this has only minor impact on $\bar{E}/N$, as $\bar{E}$ also increases .

In the interacting case, the CPU time per run increases by a factor ~ 50 from $N = 10$ to $N = 100$, while in the non-interacting case, the time increase is ~ 10 for the same increase in particles. As the inclusion of the Jastrow factor becomes increasingly important for higher numbers of particles, this also means that optimization is especially important when simulating higher numbers of particles. The error estimates increase with N, both due to increased truncation as well as a decreased number of average MC cycle per particle. Although the error estimates are of the same magnitude in both the interacting and non-interacting case, the interacting system yields in general a higher error on the estimated ground energy, possibly due to more truncation following the increased complexity of the local energy expression. As such, achieving results with a specific desired accuracy will most likely require increasing the number of Monte Carlo cycles as N is increased.

The α-values used in producing table 6 were obtained by Gradient Descent over 10 runs prior to the runs used in making the table. As the α gradient from each of these preliminary parameter runs is based on one run, the $\hat{\sigma}$ value in each such run directly reflects upon the variance in GD over time, meaning that N large will results in less accurate estimates on α, which again means a larger $\hat{\sigma}$ when calculating $\bar{E}$. In other words, as N increases, it becomes increasingly difficult to make a decent estimation of α, leading to increased error. This can possibly remedied to some extent by using a high number of cycles in each run when using GD. However, as N and α both affect which value of Δ should be used in the Metropolis algorithm, this becomes a challenge indeed. Because there are many parameters involved in the VMC engine, and because, as has been shown above, several of them interact with regards to the error estimate, these types of simulations are well suited for machine learning methods.
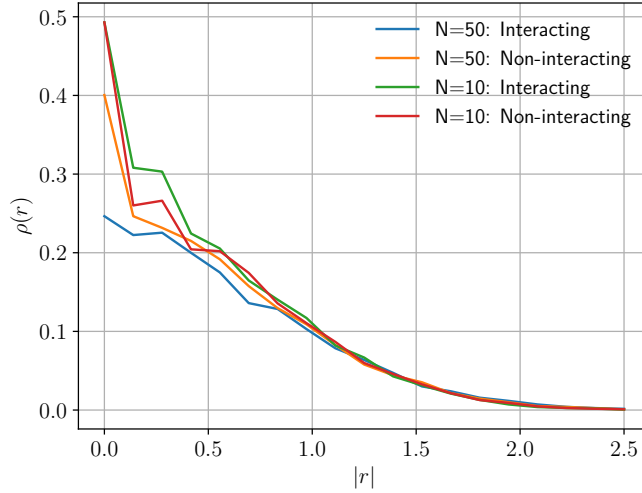

Figure 3: Bose gas in spherical trap, $\lambda = 1$; One-Body density, $\rho(r)$ for $N = 10$ and $N = 50$, both interacting and non-interacting. The data is averaged from 7500 particles total for each plot using importance sampling ($\Delta t = 0.1$) after 2^{15} runs.

Figure 3 shows the one-body densities [5], which serves as a visualization of wave-functions, for $N = 10$ and $N = 50$, with and without the Jastrow factor/particle interaction. The figure indicates that the $N = 10$ systems and the non-interacting $N = 50$ case have somewhat higher one-body densities in the region $r = [0, 1.0]$ than in the $N = 50$ interacting case. In the case of the $N = 50$

interacting system, the lower one-body density in the this region is most likely due to the combination of a relative high number of particles and the inclusion of the repulsive potential. After all, as the particles are in a three dimensional room, the available volume in $r = [0, 1.0]$ for example is significantly less that than in region $r = [1.0, 2.0]$. In the $N = 10$, $a > 0$ case, there are likely too few particles for the repulsive potential to make a significant change on the distribution of particles. Because $\rho(r) = H(r)/r^2$, differences in particle numbers for larger values of r will be less visible than difference for smaller values of r - which explains why the one-body density of the $N = 50$ interacting case is not visibly higher for $r > 1.0$, as the lower density in the $r < 1.0$ region most definitely implies.

4   Conclusions

Based on the results and discussion above, it is clear that the error estimate, σ, on correlated data without methods such as "blocking" is large deceptive. In most cases, the error estimate using "blocking", $\hat{\sigma}$, was entirely necessary, such as when studying the effect of Δt used in the Importance Sampling method on estimation quality of $\bar{E}$. In this project, I used "blocking" post-simulation in a separate Python implementation, which encumbered the analysis of the results severely. Similar research in the future will in my opinion benefit from implementing the required error analysis in such a way that it may be be executed effectively in sequence with generating the data samples.

The presented results indicated that Importance Sampling may accelerate the equilbration of the system due to a higher acceptance rate A, enabling simulations with fewer Monte Carlo cycles than when using the Brute Force Monte Carlo integration. This requires finding a suitable Δt value, as some of the tested values were shown to equilibrate the local energy E_L slower than the Brute Force method, and accepting nearly all proposed particle configurations, leading to a larger error estimate $\hat{\sigma}$. Without the Jastrow factor, for $N = 10$ $\alpha = 0.45$ after 2^{21} MC cycles, $\Delta t = 0.1$ produced an average $\hat{\sigma} \sim 10^{-3}$ ($A = 0.9$), which was the same order of magnitude as the Brute Force method, which had $A = 0.74$. Simultaneously, $\Delta t = 0.005$ resulted in $\hat{\sigma} \sim 10^{-2}$ with $A \approx 1$.

Increases in particle number N was generally observed to increase $\hat{\sigma}$. After 2^{18} cycles $\hat{\sigma} \sim 10^3$ ($N = 10$) increased to $\sim 10^2$ ($N = 50$) and $\sim 10^{-1}$ ($N = 100$), both with and without the Jastrow factor. This did not however impact on the validity of the results on average estimated energy per particle, $\bar{E}/N$ as $\hat{\sigma}$ scaled to N at a slower rate than $\bar{E}$. When including the Jastrow factor however, $\bar{E}/N$ increased from $\bar{E}/N \approx 2.45 \approx$ constant for all values of N, to 4.55 ($N = 50$) and 2.66($N = 100$). Based on this, I concluded that the Jastrow factor is increasingly important for higher values of N, which was further underlined by studying the one-body density of systems of $N = 50$ and $N = 100$ particles.

The simulations performed in this project have to some extent yielded results that were difficult if not impossible to draw any definite conclusions from. This has largely been due to the stochasticity introduced through the Monte Carlo algorithms, which for some purposes require averaging a high number of simulations in order to study the finer nuances. The large number of parameters involved (N, Δt, α, η, etc.) have further complicated interpreting the results. Due to this, further study focusing on certain aspects such as; the effect of the approximation of α on the optimal Δt, one-body densities in the $r \to 0$ limit, or the impact of N on the best suited Δt have been only been studied cursory. More elaborate and focused research than what is encompassed by the scope of this project is therefore required in order to make any conclusions on those subjects. Furthermore, the time required to run each simulation went from $\sim 10^0$ ($N = 10$) seconds to 10^2 ($N = 100$). As such, I find that VMC is well suited for application of Machine Learning methods, especially in tandem with well optimized and parallelized code.

References

[1] J. L. Dubois and H. R. Glyde. Bose-einstein condensation in trapped bosons: A variational monte carlo analysis. *Physical Review A*, 63(2), 2001.

[2] H. Flyvbjerg and H. G. Petersen. Error estimates on averages of correlated data, 1989.

[3] David J Griffiths. Introduction to quantum mechanics, 1995.

[4] A. Weisse H. Fehske, R. Schneider. Computational many-particle physics, 2008.

[5] Jørgen Høgberget. Quantum monte-carlo studies of generalized many-body systems, 2013.

[6] Marius Jonsson. Standard error estimation by an automated blocking method, 2018.

[7] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Adaptive computation and machine learning. MIT Press, Cambridge, 2012.

[8] J. K. Nilsen, Jordi Mur Petit, Montserrat Guilleumas, M. Hjorth Jensen, and Artur Polls Martí. Vortices in atomic bose-einstein condensates in the large-gas-parameter region. *Physical Review A*, 2005.

[9] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

Appendices

A   Appendix 1.

α	$\bar{E}$	$\hat{\sigma}$	$t(ms)$
0.30	5.69	1.94×10^{-2}	-
0.35	5.34	1.19×10^{-2}	-
0.40	5.14	6.94×10^{-3}	-
0.45	5.03	3.12×10^{-3}	-
0.50	5.00	0.00	-
0.55	5.02	2.65×10^{-3}	-
0.60	5.08	4.88×10^{-3}	-
0.65	5.17	6.19×10^{-3}	-
0.70	5.27	8.16×10^{-3}	-
Average	-	7.15×10^{-3}	2661

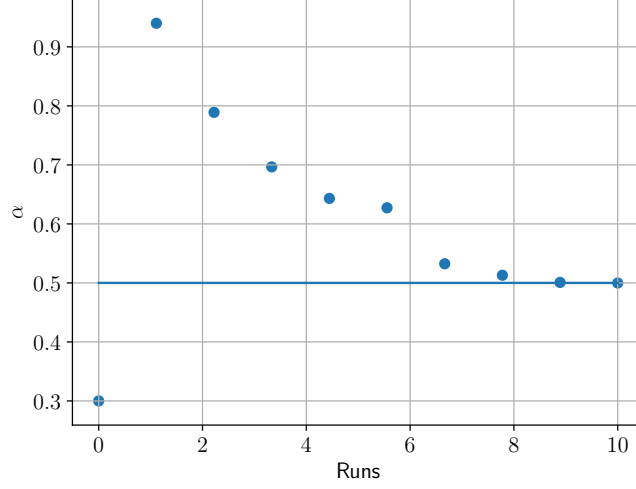Table 7: $N = 10$, $D = 1$ after 2^{21} cycles, with a 0.3 equilibrium fraction, for various values of α.

Figure 4: Non-interacting case. Gradient Descent on α for $N = 100$, $d = 3$, using importance sampling ($\Delta t = 0.1$)

Appendix 2.

Analytic local energy, non-interacting spherical

Gradient of g:

$$\nabla_i g(\alpha, \beta, \mathbf{r}_i) = -2\alpha(\boldsymbol{x} + \boldsymbol{y} + \beta\boldsymbol{z})g(\alpha, \beta, \mathbf{r}_i)$$

Laplacian of g: From the product rule, derivation of $\mathbf{r}_i$, gives a coefficient d representing the dimensionality of r.

$$\nabla_i^2 g(\alpha, \beta, \mathbf{r}_i) = (-2(d - 1 + \beta)\alpha + 4\alpha^2(x^2 + y^2 + \beta^2 z^2))g(\alpha, \beta, \mathbf{r}_i)$$

Meaning that in the $\beta = 1$ case,

$$E_L(\mathbf{r}) = \sum_i^N \left(\frac{-\hbar^2}{2m}(-2d\alpha + 4\alpha^2 \mathbf{r}_i^2) + \frac{1}{2}m\omega_{ho}^2 r_i^2 \right)$$

Using natural units, $\hbar = c = 1$, and unity mass $m = 1$, the equation becomes;

$$E_L(\mathbf{r}) = \alpha dN + \left(-2\alpha^2 + \frac{1}{2}\omega_{ho}^2 \right) \sum_i^N r_i^2$$

And using the gradient of g, the drift force is

$$F_i = \frac{2\nabla\Psi_T}{\Psi_T} = -4\alpha\mathbf{r}_i$$

Analytic local energy, interacting eliptical

Starting with

$$E_L(\mathbf{r}) = \frac{1}{[\prod_i \phi_i] \exp\left(\sum_{j<k} u_{jk}\right)} \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\omega_{ho}^2 r^2\right) \left[\prod_i \phi_i\right] \exp\left(\sum_{j<k} u_{jk}\right)$$

With the first goal being to calculate the term

$$\frac{1}{\Psi_T(\mathbf{r})} \sum_i^N \nabla_i^2 \Psi_T(\mathbf{r}),$$

starting by taking the gradient with respect to the k'th particle;

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \left(\left[\prod_i \phi_i\right] \exp\left(\sum_{j<m} u_{jm}\right)\right)$$

$$= \left(\nabla_k \left[\prod_i \phi_i\right]\right) \exp\left(\sum_{j<m} u_{jm}\right) + \left[\prod_i \phi_i\right] \left(\nabla_k \exp\left(\sum_{j<m} u_{jm}\right)\right)$$

The gradient of the non-interacting part of the TW:

$$\nabla_k \left[\prod_i \phi_i\right] = \nabla_k \phi_k \left[\prod_{i\neq k} \phi_i\right] = \nabla_k \phi_k \frac{\prod_i g(\alpha, \beta, \mathbf{r}_i)}{\phi_k} \tag{37}$$

And the gradient of the interacting part, remembering that $r_{kl} = r_{lk}$;

$$\nabla_k \exp\left(\sum_{j<m} u_{jm}\right) = \exp\left(\sum_{j<m} u_{jm}\right) \sum_{j\neq k} \nabla_k u_{kj} = \prod_{j<m} f(r_{jm}) \sum_{l\neq k} \nabla_k u_{kl} \tag{38}$$

Thus (37) is

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \phi_k \left[\prod_{i\neq k} \phi_i\right] \exp\left(\sum_{j<m} u_{jm}\right)$$

$$+ \left[\prod_i \phi_i\right] \exp\left(\sum_{j<m} u_{jm}\right) \sum_{l\neq k} \nabla_k u_{kl} \tag{39}$$

or

$$\nabla_k \Psi_T(\mathbf{r}) = \nabla_k \phi_k \frac{\prod_i g(\alpha, \beta, \mathbf{r}_i)}{\phi_k} \prod_{j<m} f(r_{jm})$$

$$+ \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{j<m} f(r_{jm}) \sum_{l\neq k} \nabla_k u_{kl} \tag{40}$$

$$= \left(\frac{\nabla_k \phi_k}{\phi_k} + \sum_{l\neq k} \nabla_k u_{kl}\right) \Psi_T(\mathbf{r})$$

Next, the above expressions are used to find the second derivative;

$$
\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{1}{\Psi_T(\mathbf{r})}\nabla_k\left(\left(\frac{\nabla_k\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k u_{kl}\right)\Psi_T(\mathbf{r})\right)
$$

$$
= \frac{1}{\Psi_T(\mathbf{r})}\left(\left(\phi_k\nabla_k\frac{1}{\phi_k}+\frac{\nabla_k^2\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k^2 u_{kl}\right)\Psi_T(\mathbf{r})+\left(\frac{\nabla_k\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k u_{kl}\right)^2\Psi_T(\mathbf{r})\right)
$$

$$
= \left(\frac{\nabla_k\phi_k}{\phi_k}\right)^2+\frac{\nabla_k^2\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k^2 u_{kl}+\left(\frac{\nabla_k\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k u_{kl}\right)^2
$$

$$
= -\left(\frac{\nabla_k\phi_k}{\phi_k}\right)^2+\frac{\nabla_k^2\phi_k}{\phi_k}+\sum_{l\neq k}\nabla_k^2 u_{kl}+\left(\frac{\nabla_k\phi_k}{\phi_k}\right)^2+2\left(\frac{\nabla_k\phi_k}{\phi_k}\sum_{l\neq k}\nabla_k u_{kl}\right)+\left(\sum_{l\neq k}\nabla_k u_{kl}\right)^2
$$

$$
= \frac{\nabla_k^2\phi_k}{\phi_k}+2\frac{\nabla_k\phi_k}{\phi_k}\sum_{l\neq k}\nabla_k u_{kl}+\sum_{l\neq k}\nabla_k^2 u_{kl}+\left(\sum_{l\neq k}\nabla_k u_{kl}\right)^2
$$

$$
\text{(41)}
$$

In order to simplify applying the ∇_k-operator to u_{kl}, the operator is re-written:

$$
\nabla_k = \nabla_k\frac{\partial r_{kl}}{\partial r_{kl}} = \nabla_k\sqrt{(\mathbf{r}_k-\mathbf{r}_l)^2}\frac{\partial}{\partial r_{kl}} = \frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}\frac{\partial}{\partial r_{kl}}
$$

This re-written operator is then applied to the $\nabla_k u_{kl}$ terms, such that

$$
\nabla_k u_{kl} = \frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}\frac{\partial u_{kl}}{\partial r_{kl}} = \frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}u'_{kl}
$$

And

$$
\nabla_k^2 u_{kl} = \left(\nabla_k\frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}\right)\partial u'_{kl}+\frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}(\nabla_k u'_{kl})
$$

$$
= \left(\frac{\nabla_k(\mathbf{r}_k-\mathbf{r}_l)}{r_{kl}}\right)u'_{kl}+(\mathbf{r}_k-\mathbf{r}_l)\left(\nabla_k\frac{1}{r_{kl}}\right)u'_{kl}+\frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}(\nabla_k u'_{kl})
$$

$$
= \frac{d}{r_{kl}}u'_{kl}-(\mathbf{r}_k-\mathbf{r}_l)\frac{(\mathbf{r}_k-\mathbf{r}_l)}{r_{kl}^3}u'_{kl}+\left(\frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}\right)^2 u''_{kl}
$$

$$
= \left(\frac{d}{r_{kl}}-\frac{(\mathbf{r}_k-\mathbf{r}_l)^2}{r_{kl}^3}\right)u'_{kl}+\left(\frac{\mathbf{r}_k-\mathbf{r}_l}{r_{kl}}\right)^2 u''_{kl}
$$

Where $(\mathbf{r}_k-\mathbf{r}_l)^2 = r_{kl}^2$, thus

$$
\nabla_k^2 u_{kl} = \left(\frac{d}{r_{kl}}-\frac{1}{r_{kl}}\right)u'_{kl}+u''_{kl} = \frac{d-1}{r_{kl}}u'_{kl}+u''_{kl}
$$

Applied to the Laplacian;

$$\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\phi_k}{\phi_k} + 2\frac{\nabla_k\phi_k}{\phi_k}\sum_{l\neq k}\frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}}u'_{kl}$$
$$+ \left(\sum_{l\neq k}\frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}}\partial u'_{kl}\right)^2 \qquad (42)$$
$$+ \sum_{l\neq k}\left(\frac{d-1}{r_{kl}}u'_{kl} + u''_{kl}\right)$$

Expanding the third term, re-arranging, and inserting $d = 3$;

$$\frac{1}{\Psi_T(\mathbf{r})}\nabla_k^2\Psi_T(\mathbf{r}) = \frac{\nabla_k^2\phi_k}{\phi_k} + 2\frac{\nabla_k\phi_k}{\phi_k}\sum_{l\neq k}\frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}}u'_{kl}$$
$$+ \sum_{j\neq k}\sum_{l\neq k}\frac{(\mathbf{r}_k - \mathbf{r}_l)(\mathbf{r}_k - \mathbf{r}_j)}{r_{kj}r_{kl}}u'_{kj}u'_{kl} \qquad (43)$$
$$+ \sum_{l\neq k}\left(u''_{kl} + \frac{2}{r_{kl}}u'_{kl}\right)$$

For the drift Force; Using (40), the drift force of for particle k in the interacting system

$$F_k = \frac{2\nabla\Psi_T}{\Psi_T} = 2\left(\frac{\nabla_k\phi_k}{\phi_k} + \sum_{l\neq k}\nabla_k u_{kl}\right)$$