BOLTZIEMANMACHINE YO!

Johan Nereng

Department of Physics, University of Oslo, Norway

Spring, 2020

Abstract

her kjem det greier

Author's comments: *Lalalla*

1 Introduction

method and the "automated blocking" algorithm from [**?**].

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 [**?**], 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;

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

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

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

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, $m = 1$;

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

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

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

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

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

Thus

$$E_L(\mathbf{r}) = \frac{1}{2}\sum_{k=1}^N \left(\frac{\nabla_k^2 \phi_k}{\phi_k} + 2\frac{\nabla_k \phi_k}{\phi_k} \sum_{l\neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_l}{r_{kl}} u'_{kl} + \left(\sum_{j\neq k} \frac{(\boldsymbol{r}_k - \boldsymbol{r}_j)}{r_{kj}} u'_{kj}\right)^2 + \sum_{l\neq k}\left(u''_{kl} + \frac{2}{r_{kl}} u'_{kl}\right) + V_{ext}(\boldsymbol{r}_i)\right) \tag{15}$$

Where

$$\begin{aligned} \frac{\nabla_k \phi_k}{\phi_k} &= -2\alpha(x_k \vec{e}_1 + y_k \vec{e}_2 + \beta z_k \vec{e}_3), \\ \frac{\nabla_k^2 \phi_k}{\phi_k} &= -2\alpha(d - 1 + \beta) + 4\alpha^2(x_k + y_k^2 + \beta^2 z_k^2), \\ \frac{\partial u_{kl}}{\partial r_{kl}} &= \frac{a}{r_{kl}(r_{kl} - a)}, \\ \frac{\partial^2 u_{kl}}{\partial r_{kl}^2} &= \frac{a^2 - 2a r_{kl}}{r_{kl}^2 (r_{kl} - a)^2} \end{aligned}$$

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

$$\begin{aligned} H &= \sum_i^N \left(\frac{-\hbar^2}{2m}\frac{1}{a_{ho}^2}\nabla_i^2 + \frac{1}{2}m\omega_{ho}^2 a_{ho}^2 (x^2 + y^2 + \frac{\omega_z^2}{\omega_{ho}^2}z^2] \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \\ &= \sum_i^N \left(\frac{-\hbar^2}{2m}\frac{m\omega_{ho}}{\hbar}\nabla_i^2 + \frac{1}{2}m\omega_{ho}^2 \frac{\hbar}{m\omega_{ho}} (x^2 + y^2 + \frac{\omega_z^2}{\omega_{ho}^2}z^2) \right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \qquad (16) \\ &= \sum_i^N \frac{1}{2}\left(-\nabla_i^2 + x^2 + y^2 + \lambda^2 z^2\right) + \sum_{i<j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j) \end{aligned}$$

2.2 Variational Monte Carlo

There are various methods available to circumnavigate having to directly solve the Schrõdinger 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 [**?**][p.256.];

$$E_0 \leq E[H] = \langle H \rangle = \langle \Psi | H | \Psi \rangle \qquad (17)$$

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];

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

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 [**?**][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;

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

, 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.

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

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

2.2.1 Metropolis algorithm

The Metropolis algorithm [**?**][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 [**?**][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;

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) [**?**][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

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

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 [**?**], 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 "blocking" method (explained in detail in [**?**]) 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

[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.

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;

Algorithm: The automated "blocking" method

```
1. Initialize algorithm
        - Set $\alpha$
        - 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 $\vec{E}_i \rightarrow \vec{E}_{i+1}$, 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

4 Conclusions

Appendices

A Appendix 1.

A.1 Analytical expression for E_L

$$\begin{aligned}
\Psi(\mathbf{X}) &= F_{rbm}(\mathbf{X}) \\
&= \frac{1}{Z}\sum_{\{h_j\}} e^{-E(\mathbf{X},\mathbf{h})} \\
&= \frac{1}{Z}\sum_{\{h_j\}} e^{-\sum_i^M \frac{(X_i-a_i)^2}{2\sigma^2}+\sum_j^N b_j h_j+\sum_{i,j}^{M,N} \frac{X_i w_{ij} h_j}{\sigma^2}} \\
&= \frac{1}{Z} e^{-\sum_i^M \frac{(X_i-a_i)^2}{2\sigma^2}} \prod_j^N (1+e^{b_j+\sum_i^M \frac{X_i w_{ij}}{\sigma^2}})
\end{aligned}$$

$$\frac{1}{\Psi_T}\nabla_k \Psi = \nabla_k \ln \Psi_t$$

$$v(\boldsymbol{X}) = b_j + \sum_i^M \frac{X_i w_{ij}}{\sigma^2}$$

$$\begin{aligned}
\nabla_k \ln \Psi_T &= \nabla_k \big(\ln \frac{1}{Z} \ln - \sum_i^M \frac{(X_i-a_i)^2}{2\sigma^2} + \ln \prod_j^N (1+e^{v(\boldsymbol{X})}) \big) \\
&= -\frac{(X_k-a_k)}{\sigma^2} + \sum_i^N v'(\boldsymbol{X}) u(v(\boldsymbol{x})) \\
&= -\frac{(X_k-a_k)}{\sigma^2} + \sum_i^N \frac{W_{kj}}{\sigma^2} u(v(\boldsymbol{x}))
\end{aligned}$$

Where $u(x) = \frac{1}{1+e^{x}}$
And

$$\begin{aligned}
\nabla_k^2 \ln \Psi_T &= \nabla_k\big(-\frac{(X_k - a_k)}{\sigma^2} + \sum_i^N \frac{W_{kj}}{\sigma^2} u(v(\boldsymbol{x}))\big) \\
&= -\frac{1}{\sigma^2} + \sum_i^N \frac{W_{kj}}{\sigma^2} u(v(\boldsymbol{x}))u'(v(\boldsymbol{x}))v'(\boldsymbol{x}) \\
&= -\frac{1}{\sigma^2} + \sum_i^N \frac{W_{kj}}{\sigma^2} u(v(\boldsymbol{x}))u'(v(\boldsymbol{x}))v'(\boldsymbol{x})
\end{aligned}$$