


# COMPUTATIONAL PHYSICS II: QUANTUM MECHANICAL SYSTEMS

## A Restricted Boltzmann Machine coupled with Variational Monte Carlo

Mohamad Mahmoud

 [GitHub - click here -](#)

June 2021

### Abstract

A Gaussian-binary restricted Boltzmann Machine (RBM) coupled with variational Monte-Carlo (VMC) with the aim of developing better ways of defining a trial wave function for a many-body quantum system than that with standard VMC approaches. The study was conducted for a system of two ground state electrons in a two dimensional harmonic oscillator trap. We contrasted the results from the RBM to that of standard VMC and exact solutions. We looked at momentum based gradient descent and ADAM's reliability when interaction between particles is present and when disregarded. We also studied how the RBM's results are affected by the number of hidden nodes. We determined that the RBM's poor performance in the interactive case is due to the trial wave function lacking a Jastrow factor component to account for the repulsive effect between particles.

## Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>Theory</b>	<b>2</b>
i	The System . . . . .	2
ii	The Restricted Boltzmann Machine . . . . .	3
	Defining the Trial Wave Function . . . . .	4
<b>III</b>	<b>Method</b>	<b>5</b>
i	Variational Monte Carlo . . . . .	5
ii	Gradient Descent . . . . .	5
	Momentum based Gradient Descent . . . . .	6
	The ADAM Optimizer . . . . .	6
iii	Resampling: Blocking . . . . .	6
<b>IV</b>	<b>Results &amp; Discussions</b>	<b>6</b>
i	The General Solution . . . . .	7
	Gradient Descent . . . . .	7
	Hidden Nodes . . . . .	8
ii	The Squared Solution . . . . .	9
<b>V</b>	<b>Conclusion</b>	<b>10</b>
i	The General Solution . . . . .	10
ii	The Squared Solution . . . . .	10
iii	Summary . . . . .	10
	<b>References</b>	<b>12</b>
<b>VI</b>	<b>Appendix</b>	<b>13</b>
i	The General Wave Function . . . . .	13
ii	The Squared Wave Function . . . . .	15
iii	Figures & Tables . . . . .	16
	The General Wave Function . . . . .	16

## I. INTRODUCTION

The world of quantum mechanics is reliant on solving large-scale and complex many-body problems. Much of our insight is based on small systems with limited complexities or where simplifications are made, in order to obtain models that are analytically solvable. This is why for large multi-dimensional complex systems we tend to resort to numerical approaches. Which serve as the stepping-stone for understanding and arriving at solutions for many-body problems; that otherwise are impossible to solve analytically. The standard approach is encompassed within a family of computational methods denoted quantum Monte Carlo. One of them is the variational Monte Carlo (VMC) method which is what our work in [1] is based on. VMC is a standard approach to arrive at accurate approximations of the quantum many-body problem. Its shortcoming is that it's reliant on the ansatz' (trial wave function) generic form's ability to describe the quantum system. Which ultimately means that arriving at a general trial wave function that is suitable for different applications can prove itself to be difficult.

This is one of the reasons as to why advancements within quantum machine learning are taking place. With the goal of providing better and more suitable approaches to tackle the scalability problem quantum mechanics faces. In this project paper we aim to explore a particular development within this field dubbed Boltzmann machines. Much of what is done and implemented here is inspired by the work of Carleo and Troyer [2]. Our main focus is going to be the ground state energy of a system of two interacting electrons (or bosons) trapped in a 2-dimensional harmonic oscillator potential. By pinning a machine learning approach against VMC we can study the validity of the results produced, rectify the model and make adjustments if needed. Hopefully, arriving at a methodology that equates to the standard VMC approaches.

The version of the Boltzmann machine we are going to be working with is a restricted Boltzmann machine (RBM) with continuous Gaussian units  $\mathbf{x}$  in the visible layer and binary units  $\mathbf{h}$  in the hidden layer, referred to as the "Gaussian-binary" RBM. Generally speaking the trial wave function  $\Psi_T$  is a byproduct of how the generative RBM operates. Its weights and biases serve as the variational parameters and thus the trial wave function in theory is more suitable for generalizations. Which means that the need for the trial wave function to resemble the true solution  $\Psi$  from the very start is more or less an option now. The trial wave function is defined by the marginal distribution of  $\mathbf{x}$ ,  $P_{rbm}(\mathbf{x})$ , as either  $\Psi_T = P_{rbm}(\mathbf{x})$  (the general solution) or  $|\Psi_T|^2 = P_{rbm}(\mathbf{x})$  (the squared solution).

Taking the key findings from our work in [1], a VMC approach with Metropolis-Hastings sampling is used to estimate the local energy. The RBM's mechanism serves as a method at defining the so-called *Neural Quantum State* (NQS) trial wave function. We make use of two gradient descent schemes: *Momentum based Gradient*

*Descent* (MGD) and *ADaptive Moment estimation* (ADAM). Gradient descent's task is to tweak the variational parameters in order to arrive at a good estimate of the ground state energy. We test the validity of both schemes and look at the relation between the number of inputs in the visible layer and the number of nodes required in the hidden layer. Finally, we contrast the results from the general solution with the results from the squared solution and compare the overall results to the exact solutions provided by Taut [3], our VMC results from [1] and RBM result from [4].

## II. THEORY

Here we expand on our work from [1] and try and find an estimate  $\langle E \rangle$  for the ground state energy  $E_{gs}$  of a system of two electrons (or bosons) using a so-called restricted Boltzmann machine. The main foundation of our numerical model is still the variational principle and it's based on the fact that given a trial wave function  $\Psi_T$  the expectation value of the energy is an upper bound for the real ground state energy. Meaning

$$\langle E \rangle = \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \geq E_{gs} \quad (1)$$

which can be rewritten as

$$= \int P(\mathbf{R}) E_L(\mathbf{R}) d\mathbf{R} \quad (2)$$

where  $P(\mathbf{R}) = \frac{|\Psi|^2}{\int |\Psi|^2 d\mathbf{R}}$  is a probability density function and  $E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})} H \Psi_T(\mathbf{R})$  is the local energy. A more detailed explanation of the variational principal and how these quantities are used (especially the local energy) can be found in [1].

### i. The System

The specific configuration of the quantum system for our numerical model is two electrons trapped inside a two-dimensional isotropic harmonic oscillator potential. This allows us to compare our results to the exact solutions provided by Taut [3]. The general expression of the Hamiltonian for a system with  $P$  particles is given by

$$\hat{H} = \underbrace{\sum_{p=1}^P \left( -\frac{1}{2} \nabla_p^2 + \frac{1}{2} \omega^2 r_p^2 \right)}_{\hat{H}_0} + \underbrace{\sum_{p < q} \frac{1}{r_{pq}}}_{\hat{H}_1} \quad (3)$$

where we chose to work with natural units ( $\hbar = c = e = m_e = 1$ ) as it makes things easier and is the natural route when working with numerical approaches. The Hamiltonian consists of two parts, the standard harmonic oscillator denoted by  $\hat{H}_0$  and the repulsive interaction between two particles denoted by  $\hat{H}_1$ .

A particle's coordinates are represented as a vector  $\mathbf{r} = [x, y]$  and its Euclidean norm  $r = \|\mathbf{r}\|_2 = \sqrt{x^2 + y^2}$  is the distance from the center. Meaning,  $r_p$  is the distance of the particle  $p$  from the center of the trap and

$r_{pq} = |\mathbf{r}_p - \mathbf{r}_q|$  is the distance between two particles. As for the repulsive interaction we have used the following convention:

$$\sum_{p < q} \frac{1}{r_{pq}} = \sum_{p=1} \sum_{j=p+1} \frac{1}{r_{pq}} \quad (4)$$

Which is a double sum that runs over every pairwise interaction once.

Throughout our implementation the frequency of the trap  $\omega$  will be set to  $\omega = 1$ . Since, this yields the simplest most direct way to go about things and as we mentioned analytical results are known and provided. As we are merely trying to replicate the results from [3] and [4]; granted Taut provides a much more in-depth results with a plethora of configurations for the quantum system in discussion. The Hamiltonian for a two-particle system is then given by

$$\hat{H} = -\frac{1}{2}\nabla_1^2 + \frac{1}{2}\omega^2 r_1^2 - \frac{1}{2}\nabla_2^2 + \frac{1}{2}\omega^2 r_2^2 + \frac{1}{r_{12}} \quad (5)$$

Working with natural units means that the energies are given in a.u. (atomic units). The energy of the interactive system is 3 a.u. As for the non-interactive case the total energy is 2 a.u. and follows the general expression

$$E = \frac{PD}{2} \quad (6)$$

for  $P$  particles and  $D$  dimensions. The non-interactive case has an analytical solution for the wave function  $\Psi$  given by

$$\Psi = \prod_{p=1}^P \exp\left(-\frac{r_p^2}{2}\right) \quad (7)$$

We note that much of what we showed here follows our work in [1].

## ii. The Restricted Boltzmann Machine

*Everything here is strictly derived from the lecture notes [4] for Boltzmann machines.*

The restricted Boltzmann machine (RBM) is a generative deep learning neural network made up of a visible layer  $\mathbf{x}$  and a hidden layer  $\mathbf{h}$ . It's restricted in the sense that there is no connection between nodes in the same layer; meaning the only viable connections are between a hidden and a visible node. It's generative in the sense that the neural network itself doesn't produce any outputs, instead it tries and learns a probability distribution by providing it with a *trial* probability distribution function (PDF).

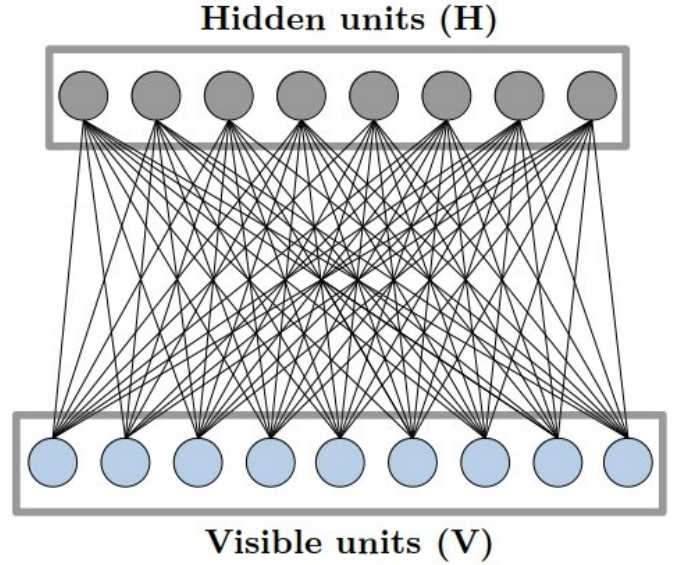


Figure 1. The general structure of a restricted Boltzmann machine, where each node within the visible and hidden layer are directly connected, no intra-layer connection present. Figure from [5].

For our application (as we will dive into later on) the trial PDF inside of the RBM represents either our trial wave function  $\Psi_T$  directly or the squared modulus  $|\Psi_T|^2$ . Within the context of machine learning a wave function representing a quantum state is often referred to as a *neural network quantum state* (NQS), [2]. Following the variational principal and our prior work in [1], we know that the NQS wave function represents a ground state once the energy of the system is minimized and thus the trial wave function is an eigenstate of the Hamiltonian yielding a quantum system variance  $\sigma^2 = 0$ . This fact is used to optimize the biases and weights of the neural network (NN) getting us closer to a trial wave function representing the ground state of the system.

The RBM of our quantum system has the particles positions  $\mathbf{r}$  as inputs, thus for the visible layer  $\mathbf{x} \in \mathbb{R}^M$ , where  $M = P \cdot D$  is the number of visible nodes. A hidden layer  $\mathbf{h} \in \mathbb{R}^N$ , where  $N$  is the number of hidden nodes. Each layer has its own corresponding bias  $\mathbf{a} \in \mathbb{R}^M$  (for the visible) and  $\mathbf{b} \in \mathbb{R}^N$  (for the hidden). Furthermore, a matrix  $\mathbf{W} \in \mathbb{R}^{M \times N}$  containing the weights connecting the two layers. All of  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{W}$  make up the network's parameters to optimize and for the sake of brevity we denote these parameters with  $\alpha$ . Given us a trial function  $\Psi_T(\mathbf{x}; \alpha)$  that is a function of both the particles positions  $\mathbf{x}$  and the RBM parameters  $\alpha$ ;  $\alpha$  being our variational parameter.

## Defining the Trial Wave Function

The RBM is described by a joint Boltzmann distribution of  $\mathbf{x}$  and  $\mathbf{h}$

$$P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp\left(-\frac{1}{T_0} E(\mathbf{x}, \mathbf{h})\right) \quad (8)$$

where  $Z$  is the partition function (a normalization constant) defined as

$$Z = \int \int \exp\left(-\frac{1}{T_0} E(\mathbf{x}, \mathbf{h})\right) d\mathbf{x} d\mathbf{h}$$

and  $T_0$  is the temperature and is ignored by setting it to 1, since temperature isn't a quantity of interest for our quantum system. As for the function  $E(\mathbf{x}, \mathbf{h})$  it's referred to as the *energy* of a configuration of the nodes; not to be confused with the system's local energy  $E_L$ . It gives the specifics of the relation between the hidden and visible nodes; the lower the energy of a configuration the higher the probability of it. Different versions of RBM have a different implementation of  $E(\mathbf{x}, \mathbf{h})$ . In our case the visible nodes' inputs are the coordinates of the particles inside the quantum systems. Naturally this leads us to choose a version of the restricted Boltzmann machine dubbed "Gaussian-binary", since we are working with continuous values as our inputs. The *energy* of the configuration for this particular version of RBM is given by

$$E(\mathbf{x}, \mathbf{h}) = \sum_{i=1}^M \frac{(x_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^N b_j h_j - \sum_{i,j=1}^{M,N} \frac{x_i W_{ij} h_j}{\sigma_i^2} \quad (9)$$

In the case where  $\sigma_i = \sigma$

$$E(\mathbf{x}, \mathbf{h}) = \frac{\|\mathbf{x} - \mathbf{a}\|^2}{2\sigma^2} - \mathbf{b}^T \mathbf{h} - \frac{\mathbf{x}^T \mathbf{W} \mathbf{h}}{\sigma^2}$$

Summing over  $\mathbf{h}$  to obtain the marginal distribution of  $\mathbf{x}$

$$P_{rbm}(\mathbf{x}) = \sum_{\mathbf{h}} P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp(-E(\mathbf{x}, \mathbf{h})) \quad (10)$$

Eq. 10 represents our NQS trial wave function  $\Psi_T(\mathbf{x}; \boldsymbol{\alpha})$

$$\Psi_T(\mathbf{x}; \boldsymbol{\alpha}) = \frac{1}{Z} \exp\left(-\sum_{i=1}^M \frac{(x_i - a_i)^2}{2\sigma^2}\right) \cdot \prod_{j=1}^N \left[1 + \exp\left(b_j + \sum_{i=1}^M \frac{x_i W_{ij}}{\sigma^2}\right)\right] \quad (11)$$

We have now arrived at the most general and straightforward form of our wave function. One caveat is that the above wave function fundamentally changes the probabilistic foundation of the RBM, [4]. This means a lot of the theoretical framework usually used to interpret the model, i.e. graphical models, conditional probabilities and Markov random fields break down.

A different approach and a solution to the problem that arises by setting the NQS equal to the marginal distribution of the RBM, is to let the marginal distribution represent the squared modulus of the wave function which in itself represents a probability density. Meaning letting

$$|\Psi_T(\mathbf{x}; \boldsymbol{\alpha})|^2 = P_{rbm}(\mathbf{x})$$

So the actual trial wave function is now given by

$$\Psi_T(\mathbf{x}; \boldsymbol{\alpha}) = \sqrt{P_{rbm}(\mathbf{x})}$$

yielding the alternative and better (from a quantum mechanics perspective) method of defining the NQS

$$\Psi_T(\mathbf{r}; \boldsymbol{\alpha}) = \frac{1}{\sqrt{Z}} \exp\left(-\sum_i^M \frac{(x_i - a_i)^2}{4\sigma^2}\right) \cdot \prod_j^N \sqrt{1 + \exp\left(b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2}\right)} \quad (12)$$

We will be referring to the solution of the RBM utilizing eq. 11 as the general solution and as for when eq.12 is used we refer to it as the squared solution. An entire derivation of the trial wave function for both eq. 11 and eq. 12 can be found in **VI. Appendix. i & ii** respectively. Following the derivation in the **Appendix** we get that the local energy of the system is:

$$E_L(\mathbf{r}) = \frac{1}{2} \sum_{i=1}^M \left[ -\left(\frac{\partial}{\partial x_i} \ln \Psi_T\right)^2 - \frac{\partial^2}{\partial x_i^2} \ln \Psi_T + \omega^2 x_i^2 \right] + \sum_{p < q} \frac{1}{r_{pq}} \quad (13)$$

Where the derivatives when working with eq. 11 are given by

$$\begin{aligned} \frac{\partial}{\partial x_m} \ln \Psi_T &= -\frac{1}{\sigma^2} (x_m - a_m) \\ &+ \frac{1}{\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp\left(-b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)} \\ \frac{\partial^2}{\partial x_m^2} \ln \Psi_T &= -\frac{1}{\sigma^2} \\ &+ \frac{1}{\sigma^4} \sum_{j=1}^N W_{mj}^2 \frac{\exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)}{\left(1 + \exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)\right)^2} \end{aligned}$$

and as for eq. 12

$$\begin{aligned}\frac{\partial}{\partial x_m} \ln \Psi_T &= -\frac{1}{2\sigma^2}(x_m - a_m) \\ &+ \frac{1}{2\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp\left(-b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)} \\ \frac{\partial^2}{\partial x_m^2} \ln \Psi_T &= -\frac{1}{2\sigma^2} \\ &+ \frac{1}{2\sigma^4} \sum_{j=1}^N W_{mj}^2 \frac{\exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)}{\left(1 + \exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)\right)^2}\end{aligned}$$

A few words on the quantum system with two electrons. When working with two electrons (spin-1/2 particles) the total wave function is ought to be antisymmetric with regards to the interchange of particles. We first define the total two-particle wave function as  $\Psi(x_1, x_2)$ , where  $x_i = (\mathbf{r}_i, m_i)$ ; here  $\mathbf{r}_i$  is particle  $i$ 's spatial coordinates and  $m_i$  is its spin quantum number ( $\pm 1/2$ ). Antisymmetrical wave functions need to satisfy the following condition  $\Psi(x_1, x_2) = -\Psi(x_2, x_1)$  with respect to particle interchange. The Hamiltonian eq. 3 is independent of spin, then we know that the total wave function can be expressed by separation of variables. Meaning, it's written as a product of two wave functions a spatial one and a spin-state one. We choose to express it as such  $\Psi(x_1, x_2) = \Phi(\mathbf{r}_1, \mathbf{r}_2)\chi(m_1, m_2)$ , where  $\Phi(\mathbf{r}_1, \mathbf{r}_2)$  is the spatial wave function and for all intents and purposes corresponds to both trial wave functions eq. 11 and eq. 12. While  $\chi(m_1, m_2)$  is the spin part of the two and it can either be the antisymmetric singlet state or one of the symmetrical states allowed within the triplet combination.

In order to be able to determine which spin-state the quantum system is occupying, we need to take a look at the spatial wave function's symmetry. Looking at eq. 11 and eq. 12 the coordinates of the two particles are expressed as the indexed input  $x_i$ . Meaning the input from the NQS for particle 1 correspond to  $(x_1, x_2)$  and as for particle 2 it's  $(x_3, x_4)$ ; each particle has a pair of inputs assigned to it for the 2 dimensional case. Under interchange of particles this only changes the indices (the weights and biases are interchanged accordingly as well) and does not affect the results outputted by the NQS. This indicates that both trial wave functions are symmetrical and thus  $\chi$  must be antisymmetrical in order for the total wave function to be antisymmetrical. This leaves us with a singlet state as spin state:

$$\chi = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$$

**Note:** I have to thank Øyvind for clearing stuff up for me and taking the time and effort to explain how I should approach this part right here.

### III. METHOD

#### i. Variational Monte Carlo

A thorough description of both variational methods, namely the brute-force Metropolis and the Metropolis-Hastings algorithm, can be found in our prior work in [1]. Thus, for the variational part our choice of parameters is based on our findings and prior analysis.

Since, our prior testing clearly showed that the Metropolis-Hastings algorithm is by far the most versatile of the two, we chose it as our VMC approach alongside the RBM. It's very forgiving when it comes to working with varying parameters regarding the algorithm itself. It yielded adequate results with a relatively low number of Monte Carlo cycles (as low as  $2^{16}$ ) and for a far more complex system, in terms of number of particles and dimensions, than the one we are currently working with. As well as being very forgiving with a varying time step size  $\Delta t$ . Based on our data analysis we conducted that the ideal step size is  $\Delta t = 0.1$ . The reasoning behind our choice of  $\Delta t$  is that lower values resulted in a 100% of the suggested changes being accepted. Logically speaking we can't justify accepting every single suggested move, since a suboptimal suggestion is bound to take place.

The Metropolis-Hastings algorithms makes use of the drift term; the quantum force  $F = 2\frac{\nabla\Psi_T}{\Psi_T}$ . For when working with eq. 11 it is given by

$$\begin{aligned}F &= 2\left(-\frac{1}{\sigma^2}(x_m - a_m) \right. \\ &\left. + \frac{1}{\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp\left(-b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)}\right)\end{aligned}$$

As for eq. 12

$$\begin{aligned}F &= -\frac{1}{\sigma^2}(x_m - a_m) \\ &+ \frac{1}{\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp\left(-b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)}\end{aligned}$$

#### ii. Gradient Descent

We build on our gradient descent work from [1]. As stated in [1] a simple gradient descent search, although it could yield viable results, is by no means adequate enough for a variational problem dependent on many parameters. Thus, we introduce two momentum based gradient descent algorithms. The simplest of the two and appropriately named *Momentum based Gradient Descent* (MGD) along with *ADAM* (ADaptive Moment estimation). We refrain from implementing a *Stochastic Gradient Descent* (SGD) as we simply do not have an abundance of data points to work with. Our quantum system is considerably small in size and thus we draw no benefits from SGD.

We still treat the local energy  $E_L$  as the cost function and differentiate wrt. the variational parameter  $\alpha$ , only

now  $\alpha$  contains the RBM parameters that we need to tune in order to obtain a wave function that is an eigenstate of the ground state energy. Meaning, the premise is still the same but we are working with multiple variational parameters, namely the visible and hidden biases and the weights of the neural network.

$$\bar{E}_\alpha = \nabla \langle E_L(\alpha_i) \rangle = \frac{d \langle E_L(\alpha_i) \rangle}{d\alpha} \quad (14)$$

by the chain rule and the hermiticity of the Hamiltonian the derivative wrt.  $\alpha$  is given by

$$\bar{E}_\alpha = 2 \left( \left\langle \frac{\bar{\Psi}_\alpha}{\Psi_\alpha} E_L(\alpha) \right\rangle - \left\langle \frac{\bar{\Psi}_\alpha}{\Psi_\alpha} \right\rangle \langle E_L(\alpha) \rangle \right) \quad (15)$$

Where

$$\bar{\Psi}_\alpha = \frac{d\Psi_\alpha}{d\alpha} \quad (16)$$

is the derivative of the wave function wrt.  $\alpha$ , see **VI Appendix i & ii**. Following the derivation in the **Appendix** we found that the parameters of the RBM are tuned and updated using

$$\begin{aligned} \frac{\partial}{\partial a_m} \ln \Psi_T &= \frac{1}{\sigma^2} (x_m - a_m) \\ \frac{\partial}{\partial b_n} \ln \Psi_T &= \frac{1}{1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right)} \\ \frac{\partial}{\partial W_{mn}} \ln \Psi_T &= \frac{x_m}{\sigma^2 \left( 1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right) \right)} \end{aligned}$$

for when working with eq. 11 and as for eq. 12 we get the following

$$\begin{aligned} \frac{\partial}{\partial a_m} \ln \Psi_T &= \frac{1}{2\sigma^2} (x_m - a_m) \\ \frac{\partial}{\partial b_n} \ln \Psi_T &= \frac{1}{2 \left( 1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right) \right)} \\ \frac{\partial}{\partial W_{mn}} \ln \Psi_T &= \frac{x_m}{2\sigma^2 \left( 1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right) \right)} \end{aligned}$$

### Momentum based Gradient Descent

Adding a momentum serves as a memory of the direction we are moving in parameter space. The idea is to let the previous iteration depict, to a degree, the change made in the next iteration. This is done by

$$\begin{aligned} v_t &= \gamma v_{t-1} + \eta_t \nabla \langle E_L(\alpha_i) \rangle \\ \theta_{t+1} &= \theta_t - v_t \end{aligned} \quad (17)$$

where  $\gamma \in [0, 1]$  is referred to as the momentum parameter. The advantages behind this modification is that this way the GD scheme converges faster and moves persistently towards a minimum in the case of small gradients even in the presence of stochasticity. Furthermore, it suppresses oscillations in high-curvature directions, [6].

### The ADAM Optimizer

In addition of keeping track of the running average of the first momentum, in ADAM (short for ADAPtive Moment estimation) the same is done for the second moment of the gradient. By appropriating this into our iteration scheme for the GD, we can update our parameter  $\theta$  as follows

$$\begin{aligned} g_t &= \nabla \langle E_L(\alpha_i) \rangle \\ m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ s_t &= \beta_2 s_{t-1} + (1 - \beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{s}_t &= \frac{s_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \eta_t \frac{\hat{m}_t}{\sqrt{\hat{s}_t + \epsilon}} \end{aligned}$$

Where  $m_t = \mathbb{E}[g_t]$  and  $s_t = \mathbb{E}[g_t^2]$  are the running average of both the first and second moment of the gradient.  $\epsilon \sim 10^{-8}$  is a regularization constant to prevent divergence and singularities. Both  $\beta_1$  and  $\beta_2$  set the memory lifetime of the first and second moment and are typically taken to be 0.9 and 0.999, respectively. In addition the learning rate  $\eta_t$  is set to  $10^{-3}$ , usually. This modification of the scheme insures that the learning rate is reduced in parts of the scheme where the norm of the gradient is persistently large over a time period, i.e. when approaching a minimum. Additionally the scheme allows for larger learning rates for flatter/saddle-like parts, which in turn results in a speedier convergence rate, ch. 7 in [6].

### iii. Resampling: Blocking

The theoretical background regarding blocking and that directly relates to our implementation can be found in [1]. It's based on Jonsson's work [7] so a more in-depth and mathematical theory is provided there, in addition to the code. The error estimates and use of blocking here is analogous to what has been done in [1].

## IV. RESULTS & DISCUSSIONS

Following Taut's work [3] and our previous variational Monte Carlo implementation in [1], we know how a system of 2 particles in 2 dimensional space behaves and the value of its ground state energy. Running a VMC simulation of the quantum system we obtain the following values:

$$\langle E \rangle = 2.00 \pm 3.62\text{e-}13 \text{ a.u. (non-interactive)}$$

$$\langle E \rangle = 3.00 \pm 9.09\text{e-}10 \text{ a.u. (interactive)}$$

These will be our benchmarks going forward the error was estimated using blocking similar to what we had done in [1]. The energy values are a result of a 2 particle system inside a spherical harmonic oscillator trap in 2 dimensional space; as is the case here. Simple Gaussian preset for the non-interactive case. Correlated preset for the interactive case with  $\beta = 1$  for a spherical trap type potential.

The baseline for the RBM are the results obtained by the code provided in [4]. Where the results for the non-interactive case hovered closely around 2 a.u. and as for the interactive case the energy was somewhere in between 3.2-3.3 a.u. As you will see we are going to somewhat improve on these results through testing and data analysis.

## i. The General Solution

### Gradient Descent

We start by determining the optimal learning rate  $\eta$  to work with when it comes to both MGD and Adam, in addition how the number of Monte Carlo (MC) cycles affect the quality of the results. To do so we ran both gradient descent schemes for a number of MC-cycles and learning rates. The parameters of the RBM  $\alpha$  were initialized using a uniform distribution, this is to make sure that the initial energy of the system is far from the ground state energy; typically around 6 – 7 a.u. The test was made for a 2 particle system in 2 dimensional space with 2 hidden nodes in the RBM. The goal is to test the performance of both schemes and arriving at the right preset. That is why we chose not to initialize the parameters following a normal distribution as that resulted in initial energies close to that of the ground energy. The following results are generated with a set seed to isolate any variables and with the maximum number of iteration set to 50, with a convergence test

$$|\langle E \rangle_{\text{current}} - \langle E \rangle_{\text{prior}}| \leq \epsilon$$

where  $\epsilon$  is a convergence threshold set to  $1\text{e} - 08$ .

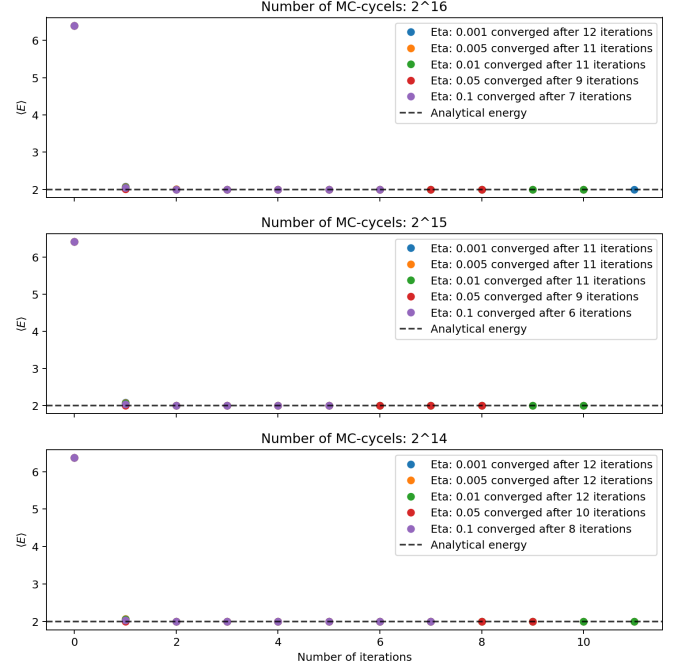


Figure 2. The number of iterations needed for the system to convert to the ground state energy for MGD. The graphs were plotted as a function of the learning rate  $\eta \in [0.001, 0.1]$  for each of the number of MC-cycles:  $[2^{14}, 2^{15}, 2^{16}]$ .

Fig. 2 shows how the value of the learning rate  $\eta$  affects the number of iterations needed to converge towards the ground state energy. Clearly a lower value of  $\eta$  require a higher number of iteration before conversion. The energy expectation value squared off at 2 a.u. for every configuration, no matter  $\eta$  or the number of MC-cycles.

Fig. 3 follows the same analysis as in fig. 2, only the results are for the Adam optimizer. It goes without saying that Adam is more fine-tuned in the way it descends towards the minimal value of any given function. Thus, it's better suited for more complex problems with a higher degree of freedom. This made so none of the configuration converged. Nevertheless, we can draw definitive conclusions from our findings. A low value of  $\eta$  stalls the optimization to a degree where the number of iterations needed to converge is very undesirable, as this noticeably affects our runtime. Since none of our configurations converged with Adam we show their efficacy by observing the quantum variance  $\sigma^2$  which indicates how close we got to the energy minimum (2 a.u.).



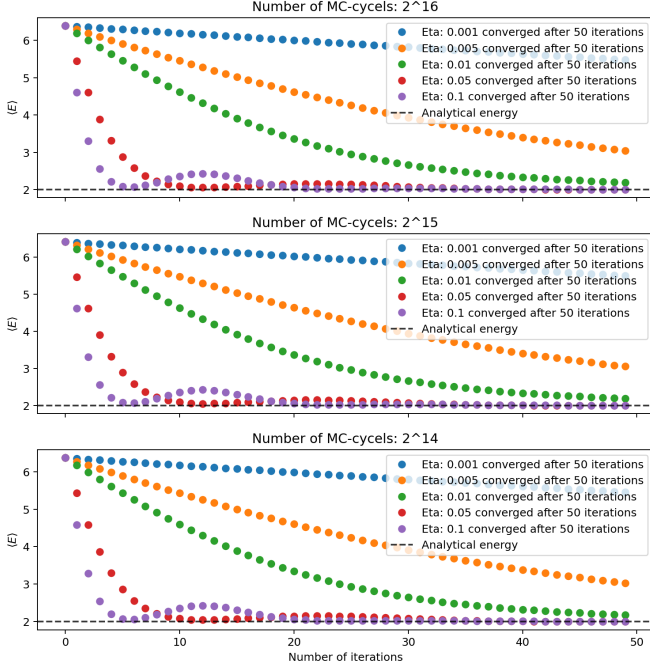


Figure 3. The number of iterations needed for the system to convert to the ground state energy for the Adam optimizer. The graphs were plotted as a function of the learning rate  $\eta \in [0.001, 0.1]$  for each of the number of MC-cycles:  $[2^{14}, 2^{15}, 2^{16}]$ .

Table I. The quantum variance  $\sigma^2$  after 50 iterations using the Adam optimizer. For  $\eta \in [0.001, 0.1]$  and number of MC-cycles:  $[2^{14}, 2^{15}, 2^{16}]$ .

MC-cycles	$\eta$				
	0.001	0.005	0.01	0.05	0.1
$2^{14}$	3.78	1.18	0.21	$9.67e-03$	$5.22e-03$
$2^{15}$	3.76	1.18	0.21	0.01	$5.61e-03$
$2^{16}$	3.86	1.20	0.21	0.01	$6.40e-03$

Looking at fig. 3 and tab. I we can conclude that setting the learning rate to  $\eta = 0.05$  yields the most optimal results. As  $\eta = 0.1$  resulted in an unwanted behavior where the scheme reversed course right before reaching the energy minimum. It's worth noting that reversing course is also true for  $\eta = 0.05$  as one can see in fig. 3. That's why at iteration 20 through 30 the plot for  $\eta = 0.05$  crosses the plot for  $\eta = 0.1$  and its energy values hover over that of  $\eta = 0.1$ . It seems that this is how the RBM behaves with Adam as we had to test with  $\eta = 0.01$  to confirm and the same happened. This time at a much later point where we had set the maximum number of iterations to 100.

Similar results are obtained for the correlated case as can be seen in fig. 5 in VI. **Appendix. iii.** Only this time MGD converges long before reaching the actual minimum energy for the interacting case; which we know to be 3 a.u., [3]. This indicates that relying on MGD to determine the ground state energy is overly optimistic. Thus,

we chose to look at the energy of the system at the last iteration before terminating for both MGD and Adam; the results are shown in tab. III and tab. IV in the **Appendix**. We steered clear from the quantum variance of the system as we found it to be non-indicative and unreliable (in this context) for the correlated system; results tab. V and tab. VI in the **Appendix**. For  $\eta = 0.05$  at its best MGD converged toward  $\approx 3.24$  while Adam yielded better results  $\approx 3.19$ . **Note:** we chose not to display our results with a precision to the second digit (which is what we usually do) in these tables, as we felt the need to keep the results as is. This is to demonstrate to the reader how different configurations behaved, since rounding off to the second digit would make it seem like these configurations behaved very similarly; which is not the case as one could see. This is the main reason as to why these tables are in the **Appendix** and not shown here in the **Results**.

The MGD serves as a good tool when no interaction between particles is present. For the non-correlated case the results with MGD were indifferent for the value of the learning rate  $\eta$  or the number of MC-cycles. As for the correlated case clearly a better learner is needed and that is why Adam yielded better results. A higher number of MC-cycles gave more accurate results and setting the learning rate  $\eta = 0.05$  yielded the best results for Adam.

## Hidden Nodes

We now study how the number of hidden nodes in the RBM effects the overall results and to what degree it correlates with the number of inputs  $M = P \cdot D$ . From the results for gradient descent we know that MGD preformed badly for a quantum system with correlation and that we had no trouble arriving at conclusive results when working with the non-interactive system. Thus, we will solely be focusing on the correlated system to study how the number of hidden nodes effects our results. We initiate the RBM parameters  $\alpha$  following a normal distribution set to  $\mathcal{N}(0, 0.001)$ . Since, we are no longer testing for the efficacy of the GD-schemes and we want our energy, if possible, to converge to a given *minimum*. Thus, we ran our simulation for `NumberHidden` = [2, 4, 8, 16] using the Adam optimizer with  $\eta = 0.05$  and  $2^{16}$  MC-cycles.



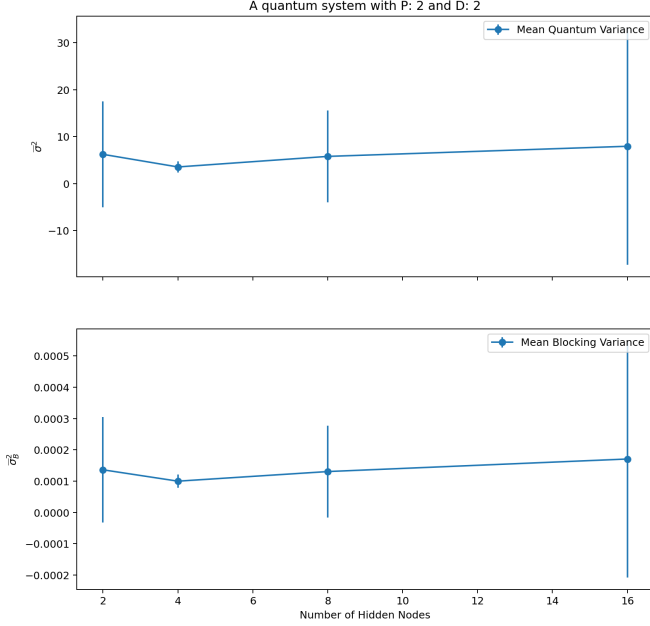


Figure 4. The mean quantum variance  $\bar{\sigma}^2$  (top), mean blocking variance  $\bar{\sigma}_B^2$  (bottom) as a function of the number of hidden nodes, along with their standard deviation as an errorbar. For a system of 2 particles in 2 dimensional space,  $M = 4$ .

We decided to consider the *mean* and *standard deviation* for each of the quantities at hand in fig. 4; namely the quantum variance  $\sigma^2$  and blocking variance  $\sigma_B^2$ . The reason behind it is we want to encompass the behavior of the RBM as a function of the number of hidden nodes. Thus, instead of looking at either the blocking or the quantum variance at the last step, we calculated the *mean* and *standard deviation* over 50 iterations (since Adam never converged) for every configuration of the RBM. Our main idea is that a lower value for the *mean* meant that the scheme propagated faster towards the minimum value and generally a value closer to 3 a.u was obtained. Additionally, a lower *standard deviation* meant that the scheme behaved in an efficient and orderly manner on its way toward a minimum value. So looking at fig. 4 we can definitively conclude that setting the number of hidden nodes equal to the number of inputs  $M = P \cdot D$  yields the best results. Furthermore, fig. 6 in **VI. Appendix. iii** for a system with 1 particle in 2 dimensional space strengthen our findings. It's worth noting that the quantum variance  $\sigma^2$  and blocking variance  $\sigma_B^2$  behave similarly and follow each others trajectory.

## ii. The Squared Solution

Before we start our testing with the squared solution of the NQS. We draw conclusions from our prior analysis, since our goal here is to use the optimal parameters we found throughout our testings and pit the general solution against the squared solution. Our main findings where that when no interaction between particles is in place us-

ing MGD for optimizing the weights gave excellent results and that MGD seemed to be independent of the learning rate and number of MC-cycles. The number of hidden nodes  $N$  must be set to the number of inputs  $M = P \cdot D$  to achieve the best results. For the interactive case we are almost explicitly in need of relying on the Adam optimizer, since MGD proved itself to converge at higher values than we deem adequate. As for the Adam optimizer setting the learning rate to  $\eta = 0.05$  and the number of MC-cycles to  $2^{16}$  was the configuration that yielded energy values closer to 3 a.u. and behaved in a way deemed adequate and fast enough in terms minimizing the energy of the system. Thus, for our testing the RBM parameters  $\alpha$  are initiated following a normal distribution  $\mathcal{N}(0, 0.001)$  with the following parameters:  $\eta = 0.05$ , number of MC-cycles  $2^{16}$ , number of hidden nodes  $N = M$ . In addition, we use MGD for the non-interactive case and Adam for the interactive case.

Rerunning our numerical simulation with this preset and starting with the non-interactive case our testing gave the following results:

$$\langle E \rangle = 2.00 \pm 2.51\text{e-}18 \text{ a.u. (general)} \quad (18)$$

$$\langle E \rangle = 2.49 \pm 3.39\text{e-}04 \text{ a.u. (squared)} \quad (19)$$

For the general solution the MGD converged after 6 iterations while it never converged for the squared solution.

As for the interactive case we obtained the following results:

$$\langle E \rangle = 3.16 \pm 1.04\text{e-}04 \text{ a.u. (general)} \quad (20)$$

$$\langle E \rangle = 3.38 \pm 2.20\text{e-}04 \text{ a.u. (squared)} \quad (21)$$

Adam never converged thus the results shown are from iteration 50.

Looking at the results clearly the squared solution is less accurate and reliable, as it ended up with higher values from the known ground state energies for both the non-interactive and interactive; this is in comparison to the general solution. It's easily observed by looking at the results for the energies and their corresponding error estimate using blocking. The squared solution not only yielded less than adequate results, its error was many orders of magnitudes larger than that of the general solution for the non-interactive case.

Initializing the RBM parameters  $\alpha$  with a normal distribution results in initial energy values close to that of the ground state energy. With this preset the squared solution values always hovered around the initial value oscillating back and forth with nothing conclusive. This doesn't necessarily mean that the gradient descent schemes aren't updating these parameters in a way that minimizes the local energy of the system. We ran a second test this time with  $\alpha$  initialized following a uniform distribution and for both the non-interactive and interactive case the energies started at much higher values and settled at around the same values, shown here in tab. II.

Table II. The energy expectation values for the squared solution at the start and end; with RBM parameters  $\alpha$  initialized following a uniform distribution.

	Non-interactive	Interactive
Start	$12.58 \pm 1.04\text{e-}02$	$13.09 \pm 1.14\text{e-}02$
End	$2.50 \pm 2.76\text{e-}04$	$3.41 \pm 2.41\text{e-}04$

Changing how the wave function is defined wrt. the marginal probability of the RBM, meant the derivatives of the wave function changed accordingly. This apparently had more impact on the validity and precision of the final outcome than we had anticipated. We initially had thought that the weights and biases would still be updated in a way that allows for the minimization of energy in the same manner as for the general solution with similar results if not better. Looking at eq. 19 and eq. 21 this is clearly not the case and frankly we were not able to pinpoint why the results are the way they are. We followed the same procedure shown in [4] and sifted through our code several times to no avail. We tried to look for errors but couldn't find any.

## V. CONCLUSION

### i. The General Solution

We started our data analysis with the general solution as the method to obtain the optimal gradient descent scheme and parameters. We chose it for our analysis since numerical baselines for the *ground state* energy with and without interaction between particles were easily provided to us in [4]. This made it easier to compare the performance of both the MGD and Adam and pit them against each other and test how different configurations of the RBM and values for the parameters affected the viability of our results. Which ultimately resulted in us being able to improve on the estimate for the ground state energy for the quantum system in question.

We managed to determine that MGD preforms excellently for a system with no interaction present between particles, fig. 2. As for an interactive quantum system this particular RBM approach deemed itself to be somewhat unreliable; more on that later on. Adam is by far the superior scheme in this context when it comes to minimizing the energy, tab. III and tab. IV. Generally speaking, we found it to be that setting the learning rate to  $\eta = 0.05$  and working with  $2^{16}$  MC-cycles for estimating the expectation value of the local energy was the configuration that yielded better results and a faster convergence rate toward a *minimum* value. It comes as a general rule that setting the number of hidden nodes  $N$  equal to the number of inputs  $M = P \cdot D$  will yield by far the best results. As opposed to any other configuration of the RBM regarding the number of hidden nodes, fig. 4.

Although our results weren't as conclusive and reliable when it comes to obtaining a wave function that is

an eigenstate of the ground state energy.

### ii. The Squared Solution

In order to compare the results of the general solution with the results produced by the squared solution. We used what we determined to be the optimal preset; obtained from various types of analysis. It seems that setting  $|\Psi_T|^2 = P_{rbm}$  impacted the structural behavior of the RBM; although from a quantum mechanics standpoint this made logical sense. Overall the results for the squared solution were significantly worse in terms of both energy and blocking error estimates. This change for the worse seems to solely stem from how we defined the wave function. As it can't be the gradient descent schemes, since as seen in tab. II the RBM parameters were tuned in a manner that minimized the energy of the system. In addition, it couldn't have been the RBM itself since better results were obtained with the general solution. Which means both parts do indeed work as intended. We cannot arrive at a logical conclusion as to why the RBM behaved this way with the squared solution. It could be something hiding in plain sight but at the time of writing this it just wasn't obvious to us.

### iii. Summary

*For the summary part we are mainly drawing conclusions from the results for the general solution as the squared solution was just not adequate enough for us to consider here.*

Through our data analysis we managed to determine the optimal parameters for the gradient descent schemes and RBM. We studied how different  $\eta$  values and number of MC-cycles affected the convergence speed and reliability of the results respectively. Which led us to concluded which gradient descent scheme served as the best tool for the job with and without interaction. In addition to how the number of hidden nodes affected the validity of the results outputted by the RBM. This was done by studying the mean value and standard deviation of both the blocking variance (our numerical approach error estimate) and quantum mechanical variance. These two quantities behaved in a comparable way, see fig. 4. Which strengthen our findings in that the number of hidden nodes  $N$  must be set equal to the number of inputs  $M = P \cdot D$ , to obtain the most optimal results. With these factors in tandem we managed to improve on the quality of the results present in [4].

The RBM deemed itself to be difficult to manage and our implementation did not yield any improvements when contrasted with VMC. Using the NQS trial wave function imposed by the Gaussian-binary RBM standard form of the *energy* function eq. 9 and its marginal distribution of  $\mathbf{x}$  eq. 10, meant missing out on a *Jastrow factor*-like component in the trial wave function. Given our findings in [1] we know it plays an essential part in governing the behavior of the particles when repulsive interactions are in place. This lead to the RBM being capable of tweaking

its weights and biases to obtain the ground state energy when no interaction was taking place eq. 19 but not when interaction between particles was present eq. 21 (it's **18 and 20 we wanted to highlight but L<sup>A</sup>T<sub>E</sub>X**). But this doesn't necessarily mean that the RBM can't be modified. Although we haven't tested it ourselves (mainly because what we are documenting here is our first implementation) a Jastrow factor could be added to the NQS without affecting the gradients of the weights and biases. The Jastrow factor does not depend on any of the variational parameters ( $a, b$  and  $W$ ) but it does depend on the particles' positions which means the derivatives of the Hamiltonian need to be reevaluated, similar to what have been done in [1].

Finally one might want to consider the option of implementing a full-fledged neural network instead of a Boltzmann machine and test how different structures and configurations affect the viability of the results produced. Given a thought through and well structured piece of code in Python, C++ or Julia, to mention a few, the structure of the neural network could be changed easily and different activation functions between layers could be used.

## REFERENCES

- [1] M. Mahmoud. Variational Monte Carlo for Bosons in a Harmonic Oscillator Trap. *Project Paper*, (2022) <https://github.com/Daheckwith/FYS4411>
- [2] G. Carleo and M. Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, vol. **355**, pp. 602–606 (2017)
- [3] M. Taut. Two electrons in an external oscillator potential: Particular analytic solutions of a Coulomb correlation problem. *Phys. Rev. A* **48**, 3561 (1993)
- [4] M. Hjort-Jensen. Computational Physics II: Lecture notes on Boltzmann Machines and Neural Networks. *Lecture Notes*, (2021). [http://compphysics.github.io/ComputationalPhysics2/doc/LectureNotes/\\_build/html/boltzmannmachines.html#](http://compphysics.github.io/ComputationalPhysics2/doc/LectureNotes/_build/html/boltzmannmachines.html#), (accessed 15.06).
- [5] G. Genovese, A. Nikeghbali, Nicola. Serra & G. Visentin. Universal approximation of credit portfolio losses using Restricted Boltzmann Machines. (2022)
- [6] M. Hjort-Jensen. Applied Data Analysis and Machine Learning, (2021). [https://compphysics.github.io/MachineLearning/doc/LectureNotes/\\_build/html/intro.html](https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html), (accessed 15.06).
- [7] M. Jonsson. Standard error estimation by an automated blocking method. *Phys. Rev. E*, **98**, 043304 (Oct 2018)

## VI. APPENDIX

### i. The General Wave Function

The joint distribution of the Boltzmann machine

$$P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{x}, \mathbf{h}))$$

For the Gaussian-binary RBM the energy of the configuration is given by

$$E(\mathbf{x}, \mathbf{h}) = \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} - \sum_j^N b_j h_j - \sum_{i,j}^{M,N} \frac{x_i W_{ij} h_j}{\sigma^2}$$

Giving us the expression for the joint distribution

$$P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} + \sum_j^N b_j h_j + \sum_{i,j}^{M,N} \frac{x_i W_{ij} h_j}{\sigma^2} \right)$$

The marginal distribution of  $\mathbf{x}$  is then obtained by

$$\begin{aligned} P_{rbm}(\mathbf{x}) &= \sum_{\mathbf{h}} P_{rbm}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} + \sum_j^N b_j h_j + \sum_{i,j}^{M,N} \frac{x_i W_{ij} h_j}{\sigma^2} \right) \\ &= \frac{1}{Z} \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} \right) \prod_j^N \sum_{h_j} \exp \left( \left( b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2} \right) h_j \right) \\ &= \frac{1}{Z} \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} \right) \prod_j^N \left( 1 + \exp \left( b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2} \right) \right) \end{aligned}$$

The marginal distribution of  $\mathbf{x}$  is what we use as our NQS, thus the trial wave function is given by

$$\Psi_T(\mathbf{x}; \boldsymbol{\alpha}) = \frac{1}{Z} \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} \right) \prod_j^N \left( 1 + \exp \left( b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2} \right) \right)$$

To make things simple we divide the wave function into two parts

$$\Psi_A(\mathbf{x}; \boldsymbol{\alpha}) = \exp \left( - \sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2} \right) \quad \Psi_B(\mathbf{x}; \boldsymbol{\alpha}) = \prod_j^N \left( 1 + \exp \left( b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2} \right) \right)$$

Giving us

$$\Psi_T(\mathbf{x}; \boldsymbol{\alpha}) = \frac{1}{Z} \Psi_A(\mathbf{x}; \boldsymbol{\alpha}) \Psi_B(\mathbf{x}; \boldsymbol{\alpha})$$

The Hamiltonian

$$\begin{aligned} \hat{H} &= \sum_{p=1}^P \left( -\frac{1}{2} \nabla_p^2 + \frac{1}{2} \omega^2 r_p^2 \right) + \sum_{p < q} \frac{1}{r_{pq}} \\ &= -\frac{1}{2} \sum_{p=1}^P \nabla_p^2 + \sum_{p=1}^P \frac{1}{2} \omega^2 r_p^2 + \sum_{p < q} \frac{1}{r_{pq}} \end{aligned}$$

For an arbitrary particle  $k$  the Laplacian  $\nabla_k^2 = \sum_{d=1}^D \frac{\partial^2}{\partial x_{kd}^2}$

$$= -\frac{1}{2} \sum_{p=1}^P \sum_{d=1}^D \frac{\partial^2}{\partial x_{pd}^2} + \sum_{p=1}^P \frac{1}{2} \omega^2 r_p^2 + \sum_{p < q} \frac{1}{r_{pq}}$$

Calculating the local energy

$$\begin{aligned}
E_L(\mathbf{r}) &= \frac{1}{\Psi_T(\mathbf{x}; \alpha)} \hat{H} \Psi_T(\mathbf{x}; \alpha) \\
&= \frac{1}{\Psi_T(\mathbf{x}; \alpha)} \left( \sum_{p=1}^P \left( -\frac{1}{2} \nabla_p^2 + \frac{1}{2} \omega^2 r_p^2 \right) + \sum_{p < q} \frac{1}{r_{pq}} \right) \Psi_T(\mathbf{x}; \alpha) \\
&= -\frac{1}{2} \frac{1}{\Psi_T} \sum_{p=1}^P \nabla_p^2 \Psi_T + \sum_{p=1}^P \frac{1}{2} \omega^2 r_p^2 + \sum_{p < q} \frac{1}{r_{pq}} \\
&= -\frac{1}{2} \frac{1}{\Psi_T} \sum_{p=1}^P \sum_{d=1}^D \frac{\partial^2}{\partial x_{pd}^2} \Psi_T + \sum_{p=1}^P \frac{1}{2} \omega^2 r_p^2 + \sum_{p < q} \frac{1}{r_{pq}}
\end{aligned}$$

Taking advantage of the fact that  $\frac{1}{f(x)} \frac{d^2}{dx^2} f(x) = \left( \frac{d}{dx} \ln f(x) \right)^2 + \frac{d^2}{dx^2} \ln f(x)$  we can rewrite the local energy as

$$= -\frac{1}{2} \sum_{p=1}^P \sum_{d=1}^D \left[ \left( \frac{\partial}{\partial x_{pd}} \ln \Psi_T \right)^2 + \frac{\partial^2}{\partial x_{pd}^2} \ln \Psi_T \right] + \sum_{p=1}^P \frac{1}{2} \omega^2 r_p^2 + \sum_{p < q} \frac{1}{r_{pq}}$$

If we were to let each visible node  $x_i$  in the Boltzmann machine represent one coordinate for a single particle

$$= \frac{1}{2} \sum_{i=1}^M \left[ -\left( \frac{\partial}{\partial x_i} \ln \Psi_T \right)^2 - \frac{\partial^2}{\partial x_i^2} \ln \Psi_T + \omega^2 x_i^2 \right] + \sum_{p < q} \frac{1}{r_{pq}}$$

Where  $\ln \Psi_T$

$$\ln \Psi_T = -\ln Z - \sum_i^M \frac{(x_i - a_i)}{2\sigma^2} + \sum_j^N \ln \left( 1 + \exp \left( b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2} \right) \right)$$

as for the derivatives over a particular position  $m$

$$\begin{aligned}
\frac{\partial}{\partial x_m} \ln \Psi_T &= -\frac{1}{\sigma^2} (x_m - a_m) + \frac{1}{\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp \left( -b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij} \right)} \\
\frac{\partial^2}{\partial x_m^2} \ln \Psi_T &= -\frac{1}{\sigma^2} + \frac{1}{\sigma^4} \sum_{j=1}^N W_{mj}^2 \frac{\exp \left( b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij} \right)}{\left( 1 + \exp \left( b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij} \right) \right)^2}
\end{aligned}$$

Setting up for gradient descent

$$\frac{\partial \langle E_L \rangle}{\partial \alpha_i} = 2 \left( \left\langle E_L \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha_i} \right\rangle - \langle E_L \rangle \left\langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha_i} \right\rangle \right)$$

Using  $\frac{1}{f(x)} \frac{d}{dx} f(x) = \frac{d}{dx} \ln f(x)$

$$= 2 \left( \left\langle E_L \frac{\partial \ln \Psi_T}{\partial \alpha_i} \right\rangle - \langle E_L \rangle \left\langle \frac{\partial \ln \Psi_T}{\partial \alpha_i} \right\rangle \right)$$

where  $\alpha_i = \{a_1, \dots, a_M\}, \{b_1, \dots, b_N\}, \{W_{11}, \dots, W_{MN}\}$ . Given us the derivatives

$$\begin{aligned}
\frac{\partial}{\partial a_m} \ln \Psi_T &= \frac{1}{\sigma^2} (x_m - a_m) \\
\frac{\partial}{\partial b_n} \ln \Psi_T &= \frac{1}{1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right)} \\
\frac{\partial}{\partial W_{mn}} \ln \Psi_T &= \frac{x_m}{\sigma^2 \left( 1 + \exp \left( -b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in} \right) \right)}
\end{aligned}$$

## ii. The Squared Wave Function

For the case where the square of the wave function equals the marginal distribution

$$|\Psi_T(\mathbf{x}; \boldsymbol{\alpha})|^2 = P_{rbm}(\mathbf{x})$$

We get

$$\begin{aligned}\Psi_T(\mathbf{x}; \boldsymbol{\alpha}) &= \sqrt{P_{rbm}(\mathbf{x})} \\ &= \frac{1}{\sqrt{Z}} \sqrt{\exp(-E(\mathbf{x}, \mathbf{h}))} \\ &= \frac{1}{\sqrt{Z}} \sqrt{\exp\left(-\sum_i^M \frac{(x_i - a_i)^2}{2\sigma^2}\right) \prod_j^N \left(1 + \exp\left(b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2}\right)\right)} \\ &= \frac{1}{\sqrt{Z}} \exp\left(-\sum_i^M \frac{(x_i - a_i)^2}{4\sigma^2}\right) \prod_j^N \sqrt{1 + \exp\left(b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2}\right)}\end{aligned}$$

Where now  $\ln \Psi_T$  in both the Hamiltonian and gradient descent becomes

$$\ln \Psi_T = -\frac{1}{2} \ln Z - \sum_i^M \frac{(x_i - a_i)^2}{4\sigma^2} + \frac{1}{2} \sum_j^N \ln \left(1 + \exp\left(b_j + \sum_i^M \frac{x_i W_{ij}}{\sigma^2}\right)\right)$$

As for the Hamiltonian its derivatives are now given by

$$\begin{aligned}\frac{\partial}{\partial x_m} \ln \Psi_T &= -\frac{1}{2\sigma^2} (x_m - a_m) + \frac{1}{2\sigma^2} \sum_{j=1}^N \frac{W_{mj}}{1 + \exp\left(-b_j - \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)} \\ \frac{\partial^2}{\partial x_m^2} \ln \Psi_T &= -\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{j=1}^N W_{mj}^2 \frac{\exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)}{\left(1 + \exp\left(b_j + \frac{1}{\sigma^2} \sum_i^M x_i W_{ij}\right)\right)^2}\end{aligned}$$

For the gradient descent

$$\begin{aligned}\frac{\partial}{\partial a_m} \ln \Psi_T &= \frac{1}{2\sigma^2} (x_m - a_m) \\ \frac{\partial}{\partial b_n} \ln \Psi_T &= \frac{1}{2 \left(1 + \exp\left(-b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in}\right)\right)} \\ \frac{\partial}{\partial W_{mn}} \ln \Psi_T &= \frac{x_m}{2\sigma^2 \left(1 + \exp\left(-b_n - \frac{1}{\sigma^2} \sum_i^M x_i W_{in}\right)\right)}\end{aligned}$$



### iii. Figures & Tables

#### The General Wave Function

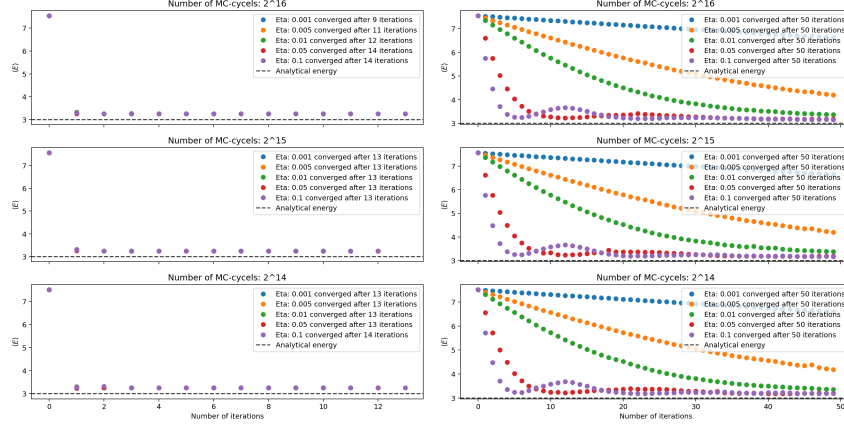


Figure 5. MGD (left) ADAM(right) interaction case

Table III. The energy with interaction after terminating using MGD.

	$\eta$				
MC-cycles	0.001	0.005	0.01	0.05	0.1
$2^{14}$	3.260251	3.259816	3.259287	3.255639	3.252248
$2^{15}$	3.245355	3.245045	3.244671	3.242218	3.240189
$2^{16}$	3.256660	3.256714	3.256785	3.257321	3.257773

Table IV. The energy with interaction after 50 iterations using Adam.

	$\eta$				
MC-cycles	0.001	0.005	0.01	0.05	0.1
$2^{14}$	6.588626	4.184225	3.356255	3.195658	3.194431
$2^{15}$	6.644774	4.201947	3.374639	3.186596	3.169611
$2^{16}$	6.628643	4.196047	3.370846	3.184150	3.164806

Table V. The quantum variance  $\sigma^2$  after terminating with interaction using MGD.

	$\eta$				
MC-cycles	0.001	0.005	0.01	0.05	0.1
$2^{14}$	3.872259	3.811062	3.739015	3.309428	3.013778
$2^{15}$	3.703582	3.641667	3.569693	3.161053	2.901101
$2^{16}$	4.563328	4.585106	4.613990	4.832228	4.967428

Table VI. The quantum variance  $\sigma^2$  with interaction after 50 iterations using Adam.

	$\eta$				
MC-cycles	0.001	0.005	0.01	0.05	0.1
$2^{14}$	6.025641	4.706237	3.832792	4.215334	2.727086
$2^{15}$	6.848933	3.978372	3.583783	2.901720	3.048104
$2^{16}$	7.955976	5.091230	3.764489	3.116812	3.316630

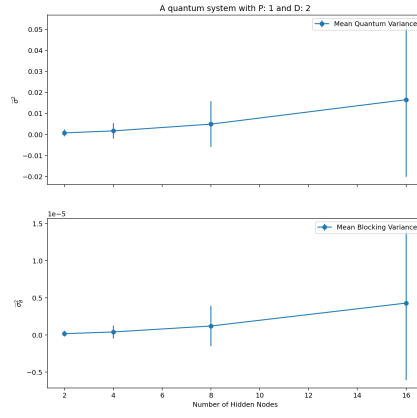


Figure 6. The mean quantum variance  $\bar{\sigma}^2$  (top), mean blocking variance  $\bar{\sigma}_B^2$  (bottom) as a function of the number of hidden nodes, along with their standard deviation as an errorbar. For a system of 1 particle in 2 dimensional space,  $M = 2$ .