FYS4411 - Project 2

Geir Tore Ulvik, Kari Eriksen, Robert Solli

June 15, 2018

Abstract

In this work we explore a system of many bosons in a harmonic oscillator trap at temperatures low enough for Bose-Einstein condesation. A variational monte-carlo machinery is implemented as a solution to find the expecation value of the energy as a function of a single variational parameter α. Over this variational parameter the ground-state energies are found by gradient descent both for a non-interactive case and with a repulsive potential. As an additional improvement to the convergence of the algorithm an importance sampling techinque for chosing the proposed next step in the metropolis algorithm is implemented. Lastly we consider the onebody densities for both systems with and without a repulsive interaction and find most particles to be localized for the non-interactive case and distributed according to a bell curve when we introduce interactions.

1 Introduction

The aim of this project is to apply Restricted Boltzmann Machine (RBM) on the quantum many-body-problem. We seek the ground state energy of a system containing N particles, however we will not extend the case of two particles. The reason for this is that for two electrons in a quantum dot with frequency $\hbar\omega = 1$ we have a closed form solutions to the ground state energy. We also have analytic answers for the non-interacting case. This is convenient in that we can compare results from our numerical methods.
Another reason to limit the amount of particles is that we operate with three optimization parameters. This requires a lot of CPU time.
Our wave function is represented by the energy of the RBM, eq. (2). We produce input to feed the RBM using Monte Carlo method with Metropolis Hasting or Importance sampling algorithm for selecting states. With stochastic gradient descent we are able to optimize the parameters, the weights W_{ij} and bias functions $a_i(v_i)$ and $b_j(h_j)$.
We also want to experiment with Gibbs sampling. For this sampling method we can use an easier version of the wave function since we know it to be positive definite. The expressions of both energy and the derivatives for the gradient descent can be found in the Appendix C and B.

2 Theory

2.1 Ansatz for the wavefunction and the hamiltonian

We are studying a system of two electrons confined in a harmonic oscillator trap described by the Hamiltonian

$$\hat{H} = \sum_{i=1}^{N} \left(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}\omega^2 r_i^2 \right) + \sum_{i<j} \frac{1}{r_{ij}} \tag{1}$$

where the first sum is the standard harmonic oscillator part and the last is the interacting part between the electrons and N represent the number of particles. ω is the oscillator frequency of the trap and r_i is the position of particle i, whereas r_{ij} is the distance between the particles and given as $r_{ij} = |\mathbf{r_i} - \mathbf{r_j}|$.
Our wave function is given from the energy of the restricted Boltzmann machine, see (4), which is the joint energy functional between the visible and hidden nodes. From the marginal probability of the joint probability distribution, see (7), we get our wave equation.

$$\Psi(X) = F_{rbm}(X) \tag{2}$$

$$= \frac{1}{Z} \exp\left(-\sum_{i}^{M} \frac{(X_i - a_i)^2}{2\sigma^2} \right) \prod_{j}^{N} \left(1 + \exp\left(b_j + \sum_{i}^{M} \frac{X_i \omega_{ij}}{\sigma^2} \right) \right) \tag{3}$$

Here Z is the partition function, X_i represents the visible nodes running up to M, and a_i and b_j are the biases described in the section below, (2.3), where number of hidden nodes j runs up to N. ω_{ij} is an $M \times N$ matrix holding the weights connecting the visible nodes with the hidden and σ is the standard deviation of the noise in our model.

2.2 Variational Monte Carlo

Monte Carlo simulations are widely used methods in numerical science that employs random walkers. In this project we are taking a closer look at trapped bosons. We are given a trail wave function we assume is as close to the real case as possible, $\Psi_T(\mathbf{R}; \alpha)$ where $\mathbf{R} = (\mathbf{R}_1, ..., \mathbf{R}_N)$ is the position of the different particles. From quantum mechanics we know the probability distribution is given by the wave function.

$$P(\mathbf{R}; \alpha) = \frac{|\Psi_T(\mathbf{R}; \alpha)|^2}{\int |\Psi_T(\mathbf{R}; \alpha)|^2 d\mathbf{R}}$$

Monte Carlo integration allow us to evaluate the integral at hand. The expectation value of the Hamiltonian is given as follows.

$$\langle \widehat{\mathbf{H}} \rangle = \frac{\int d\mathbf{R} \Psi^*(\mathbf{R}) H(\mathbf{R}) \Psi(\mathbf{R})}{\int d\mathbf{R} \Psi^*(\mathbf{R}) \Psi(\mathbf{R})}$$

The variational principle states that the expectation value of the Hamiltonian is an upper-bound for the ground state energy of the Hamiltonian.

$$E_0 \leq \langle H \rangle$$

This is what the Variational Monte Carlo method bases itself on. Given a probability distribution we can evaluate the wave function and look for a local minimum. We define the local energy by

$$\widehat{\mathbf{E}}_L(\mathbf{R}; \alpha) = \frac{1}{\Psi_T(\mathbf{R}; \alpha)} \widehat{\mathbf{H}} \Psi_T(\mathbf{R}; \alpha).$$

Then the expectation value of the local energy is given by

$$\langle \widehat{\mathbf{E}}_L \rangle = \int P(\mathbf{R}) \widehat{\mathbf{E}}_L d\mathbf{R} \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{E_L}(x_i)$$

where N is the number of Monte Carlo cycles. Now we can calculate the probability distribution and the local energy. And for each cycle we propose a new configuration $\mathbf{R}_p$ for the system and hopefully we come closer to the ground state.

$$\mathbf{R}_p = \mathbf{R} + r * \Delta \mathbf{R}$$

2.3 Restricted Boltzmann Machine

Restricted Boltzmann machine is a energy-based generative model which include hidden and visible variables (ref to Metha). It consists of a two-layer network with a two dimensional matrix W_{ij} telling how strong the connections between the hidden and visible nodes are. The energy related to a configuration of the nodes is what forms the basis of our model. It is given in (4).

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i(v_i) - \sum_\mu b_j(h_j) - \sum_{ij} W_{ij} v_i h_j \tag{4}$$

Here $a_i(v_i)$ and $b_j(h_j)$ are bias functions of the visible and hidden layers which we are allowed to choose our self. In our case we want the visible layer to take a continuous form and the hidden layer to be binary, (Gaussian-binary), meaning a_i and b_j takes the following from

$$a_i(v_i) = \frac{v_i^2}{2\sigma_i^2}, \quad b_j(h_j) = b_j h_j.$$

We are working with restricted Boltzmann machine meaning there is no connection between nodes within layers, only between layers. Also this network is a generative one so we want our network to learn a probability distribution. We begin with the joint probability distribution of the visible and hidden nodes is given in (5) where Z is the partition function, see (6).

$$F_{rbm}(\mathbf{X}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{X},\mathbf{h})} \tag{5}$$

$$Z = \int\int \frac{1}{Z} e^{-E(\mathbf{X},\mathbf{h})} d\mathbf{x} d\mathbf{h} \tag{6}$$

From (5) we can marginalize over all the hidden units and get the distribution over the visible units. And as mentioned above this is what we use to represent our wave function.

$$F_{rbm}(\mathbf{X}) = \sum_{\mathbf{h}} F_{rbm}(\mathbf{X}, \mathbf{h}) \tag{7}$$

$$= \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{X},\mathbf{h})} \tag{8}$$

Since we have no data to train we use the technique of reinforcement learning. We feed the network with input using Monte Carlo method and based on the variational principle we find the ground state of the system by seeking the configuration that gives the lowest quantum mechanical energy. According to this principle we change the weights and biases by gradient descent method and hopefully the network will converge towards the correct state.

2.4 The Metropolis Algorithm

How we select the new configurations during the simulation is given by the Metropolis Algorithm. Ideally we would have a transition probability matrix which told us how likely it would be for each particle to move in the configuration space. Since we don't we try and model it. We define the following entities:

$P_i^{(n)} \rightarrow$ probability of finding the system in state i at the n'th step.
$j \rightarrow$ possible new step
$A_{i\rightarrow j} \rightarrow$ probability of acceptance.
$T_{j\rightarrow i} \rightarrow$ probability of making the transition

Now we can say that a transition probability matrix can be constructed by $T_{j\rightarrow i}A_{j\rightarrow i}$. The probability of finding the system in state i at step n is

$$P_i^{(n)} = \sum_j \left[P_j^{(n-1)} T_{j\rightarrow i} A_{j\rightarrow i} + P_j^{(n-1)} T_{i\rightarrow j} (1 - A_{i\rightarrow j}) \right]. \tag{9}$$

We want to push the system towards high density regional space of $P_i^{(n)}$. /We want to select states according to the probability distribution. In that way to converge to the desired stationary distribution p_i.

$$P_i^{(n\to\infty)} \to p_i$$

Using this statement and the fact that the sum over all possible transitional probabilities is one we can rewrite the above equation 9.

$$\sum_j [p_j T_{j\to i} A_{j\to i} - p_i T_{i\to j} A_{i\to j}] = 0$$

In order to stop at the configuration where we have reached equilibrium we us the condition of detailed balance, and it gives us

```
Data: matrix R
Result: float E_L
number of MC-cycles N;
initialize R;
calculate |Ψ_T(R)|^2;
for i in [0, N] do
    R_p = R + r * ΔR;
    calculate ω = |Ψ_T(R_p)|^2 / |Ψ_T(R)|^2;
    if q ≤ ω then
        accept new move;
    else
        reject new move;
    end
    update energy, E_L;
end
```

Algorithm 1: Monte Carlo with Metropolis-Hastings

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

We choose to accept when the acceptance is bigger than 1.

$$A_{j\to i} = \min\left(1, \frac{p_i T_{i\to j}}{p_j T_{j\to i}}\right)$$

What we must calculate is equation 10. And an example of the code is given further down, in Algorithm 1

$$\frac{p_i T_{i\to j}}{p_j T_{j\to i}} = \frac{|\Psi_T(\mathbf{R}_p)|^2}{|\Psi_T(\mathbf{R})|^2} \tag{10}$$

2.5 Importance Sampling

Now we move to an extension of the Metropolis algorithm, the Importance sampling. This method provides us with a better way of suggesting new moves. The expression for the new move, y, is given by the solution of the Langevin equation. It reads

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t},$$

where ξ is a gaussian random variable, D is the diffusion coefficient and is $\frac{1}{2}$ and Δt is the time step which take values between $[0.001, 0.01]$. $F(x)$ is the drift force and is given by the gradient of the wave function.

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

It is the drift force that ensures us we move particles towards regions of configuration space where the trail wave function is large. This method increases the efficiency of our program, since the standard Metropolis can suggest new moves in every direction with same probability. From the solution of the Fokker-Planck equation we get a new transition probability, the Green's function.

$$G(y, x, \Delta t) = \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp(-(y - x - DF(x)\Delta t)^2/4D\Delta t)$$

Now equation 10 from Metropolis algorithm changes to

$$\frac{p_i T_{i\to j}}{p_j T_{j\to i}} = \frac{G(x, y, \Delta t)|\Psi_T(y)|^2}{G(y, x, \Delta t)|\Psi_T(x)|^2},$$

and we must calculate the drift force in order to find new moves and the Green's function to accept/reject these moves.

2.6 Gibbs Sampling

A special case of the Metropolis sampling is when we know the wave function to be positive definite. We can then express the wave function as in (**??**) and since we do not need to worry about a complex wave function we can model the probability density as we wish. Now we sample new configurations from the conditional probabilities directly. This we can do since all nodes in each layer are independent from one another.

$$P(H_j = 1|\mathbf{x}) = \frac{1}{1 + e^{-b_j - \sum_i \frac{x_i w_{ij}}{\sigma^2}}} \tag{11}$$

We first calculate values for the hidden nodes from the current configuration, which in the beginning is initialized with random numbers. Here we recognize the Sigmoid function. From this we begin generating new samples for the visible values. These however are picked from a normal distribution with mean equal $a_i + \sum_j \mathbf{w}_{ij}\mathbf{h}$ and variance equal σ^2.

$$P(X_i|\mathbf{h}) = \mathcal{N}(X_i; a_i + \sum_j \mathbf{w}_{ij}\mathbf{h}, \sigma^2) \tag{12}$$

The acceptance rate is 1, so the configuration is change every time. We then update the energy as before. Accept now we operate with a different wave function so the expression for both the energy and the derivatives in the gradient descent method are different, with a factor 0.5. So this must be accounted for. When evaluating the wave function we take a ratio and this factor vanishes, so changing the wave function itself is unnecessary.

2.7 Gradient Descent

Like most problems in physics the variational quantum problem is one of optimization. In this project we are optimizing the expectation value of the energy over the parameter α. In the naive implementation we simply span a reasonable range of α [1]. For a multi-parameter variational problem this approach is too computationally expensive to implement in most cases. To remedy this problem iterative gradient methods have been developed to make more intelligent choices for the variational parameters. In this project we have opted to implement a simple gradient descent method. The gradient descent is ordinarily presented as

$$\alpha_{i+1} = \alpha_i - \gamma \frac{\partial \langle E_L(\alpha_i) \rangle}{\partial \alpha_i} \tag{13}$$

Where we have opted to update the parameter γ iteratively by the Barzilai-Borwein formula which is a method that uses an iterative approach to the step size γ derived in much the same manner as for quasi-Newton methods[**?**]. The only constraint we have to enforce is that $0 < \alpha$. In the algorithm we have a simple rule that checks the sign of α and makes a correction if it is out-of-bounds. For $i > 0$ the step size γ_i is then given as:

$$\gamma_i = (\alpha_{i+1} - \alpha_i) \cdot \frac{1}{\langle E_L \rangle_{i+1}^{\alpha} - \langle E_L \rangle_i^{\alpha}} \tag{14}$$

We also introduce the notation $\frac{\partial \langle E_L^i \rangle}{\partial \alpha} = \langle E_L \rangle_i^{\alpha}$ for brevity, and the same notation will be used for the tiral wavefunction ψ_T. The partial derivative itself can be found by the equation[2]

2.8 Blocking Method

For the statistical analysis and error estimates we have used the technique of blocking. We are looking for the expectation value of the ground state energy of our system. In order to say something about how accurate these results are we want to look at the standard deviation.

[1] an implementation of which can be seen in the file `main_b.cpp`

[2] The equation was retrieved from the lecture notes

If our samples where uncorrelated we could calculate the standard deviation through the following equation

$$\sigma = \sqrt{\frac{1}{n-1}\left(\langle E_L^2\rangle - \langle E_L\rangle^2\right)}.$$

It can be showed that for correlated samples the equation for the standard deviation is

$$\sigma = \sqrt{\frac{1+2\tau/\Delta t}{n-1}\left(\langle E_L^2\rangle - \langle E_L\rangle^2\right)}$$

where τ is the time between one sample and the next uncorrelated sample and Δt is the time between each sample. If we knew what τ was we could simply find our σ, but since we don't we use blocking to find it. The method is simple enough, we divide our samples of data into blocks and calculating the mean of each block. So if we have an amount of samples of $\langle E_L\rangle$, we divide these into M blocks and calculate the mean. By plotting the standard deviation as function of the block size we can keep blocking and blocking until we see that the std. dev. stops increasing, and this is where the blocks are uncorrelated. Now we have an estimate for τ and thus the standard deviation.

2.9 Numerical differentiation

To evaluate the local energy of the system as defined in the project it necessary to compute the second derivative of the trial wavefunction ψ_T. We've chosen to implement the numerical differentiation as a finite difference approximation. Let $\mathbf{R}$ be the row major $N \times D$ matrix where N is the number of particles and D is their dimension. Then the second derivative can be found by the procedure listed as algorithm 2

Since the derivative involves three function calls for each particle the numerical derivative will obviously be quite computationally expensive. It is noted that the differentiation could be substantially optimized from the version included in the code, but is outside the scope of this project.

Data: matrix $\mathbf{R}$
Result: float $\nabla^2 \psi_T(R)$

$\Delta = 0$
$\mathbf{R_p} = \mathbf{R}$
$\mathbf{R_m} = \mathbf{R}$
for i *in* $[0, N-1]$ **do**
    for j *in* $[0, D-1]$ **do**
        $\mathbf{R_p}(i,j) += h$
        $\mathbf{R_m}(i,j) -= h$
        $\Delta = \psi_T(\mathbf{R_p}) + \psi_T(\mathbf{R_m}) - 2\psi_T(\mathbf{R})$
        $\mathbf{R_p}(i,j) -= h$
        $\mathbf{R_m}(i,j) += h$
    end
end
return $\frac{\Delta}{h^2}$

Algorithm 2: Numerical differentiation of the second order of the trial wavefunction on a system $\mathbf{R}$

Name	Type	Shape
R	vec	M
W	mat	$M \times N$
a	vec	M
b	vec	N
F	vec	M
G	mat	3×1

Table 1: List of different variables in our code and their type and shape. R is the positions of the particles, W is the weights, a and b the bias functions, F the drift force and G the matrix containing the gradients of local energy. N is the number of hidden nodes and M is the number of visible, equal $N_p \cdot N_d$, number of particles times number of dimensions.

3 Method

3.1 Gibbs sampling

Our Gibbs sampler works quite simple updating the hidden nodes according to (11), a listing of the code can be seen below. For each number of hidden unit we calculate the new value hj(j) given the conditional probability.

```
    for(int j = 0; j < N; j++){

        double sum_xi_wij = 0;
        for(int i = 0; i < M; i++){
            sum_xi_wij += R(i)*psi_t.W(i, j);
        }

        double Hj = psi_t.b(j) + (sum_xi_wij/sigma_sq);
        double z = 1/(1 + exp(-Hj));

        hj(j) = z;
    }
```

As for the visible units they are drawn from a normal distribution according to (12) and an illustration is seen below. Here the mean is the variable my and standard deviation is σ.

```
    for(int i = 0; i < M; i++){

        double sum_hj_wij = 0;
        for(int j = 0; j < N; j++){
            sum_hj_wij += psi_t.W(i, j)*hj(j);
        }
        double my = psi_t.a(i) + sum_hj_wij;
        //Creating a normal dist
        uniform_real_distribution<double> dis(my, sigma_sq);
        R(i) = dis(*gen);
    }
```

4 Results

5 Discussion

6 Non interacting bosons

We've observed that for non-interactive bosons we are able to find an analytical minimum for the local energy, that we've used as a reference to assert that the VMC machinery indeed produces the correct result. As noted in task 1b this information is not in general something that is available to us. Methods like gradient descent ad variations thereof must then be considered. We implemented a GD algorithm that converged fast, and with relatively few MC-cycles, to a stable minimum for α. More complex methods were not considered since we knew the problem to be quadratic, and thus well behaved in that the search space only has one minimum. As and additional metric on the system we computed the one-body desities which shows a delta like behaviour in the non-interactive regime. The physical interpretation of this is that the particles are all very close to eachother and the origio. The authors did not find any demonstrations of the onebody density for bosons in the BEC in a HO trap in the litterature so comparison weren't readily available.

We have also shown that there is, unsurprisingly, a large chunk of time lost by doing numerical estimations of the local energy. While trivial to implement and compute it is only useful as a sanity-check for the analytical solution for many body systems where an analytic derivative is available. Both the numeric and analytic simulations were run with the `-O3` optimization flag.

There is also shown to be experimental indications of the hypothesis [3] that the ground state of the ensamble of bosons also is the one that has the lowest variance in the VMC scheme by presenting the relative error in the measurment of $\langle E_L \rangle$

7 Interacting bosons

In the case of the elliptical trap with interaction we observe the expectation value of the local energy to increases by a small amount as we increase the number of particles in the system. This is consistent with what we see in Fig. 2 in ref [**?**]. As the density of the gas increases so should its energy. We are able to find an optimal value for α, not without difficulties though. With an analytic solution with many terms something is bound to go wrong. Also with the amount of time it takes to numerically solve the interactive case it is important know the code produces the correct answers. However we are happy with the results.

The onebody densities of the bosons in the ground state was shown to be distributed according to a normal distribution [4]. Which shows a large effect of the jastrow factor on the density, as the shape is nearly delta-function distributed for the non-interactive case.

[3] theory, assumption?

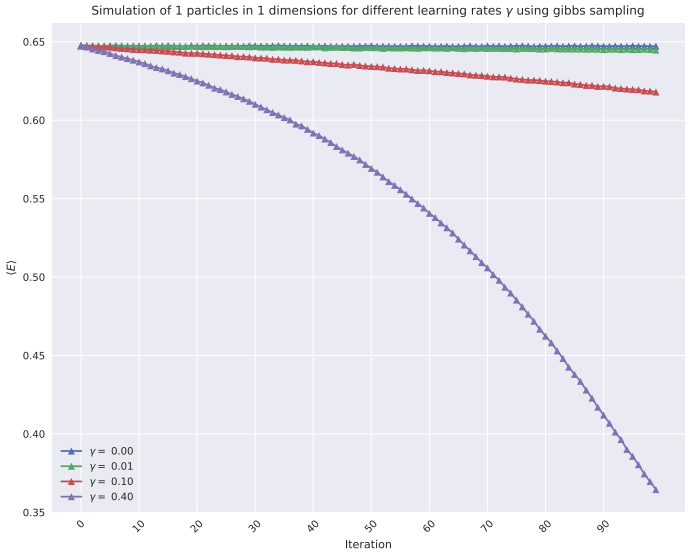
[4] This should probably be rigorously tested, but we didn't

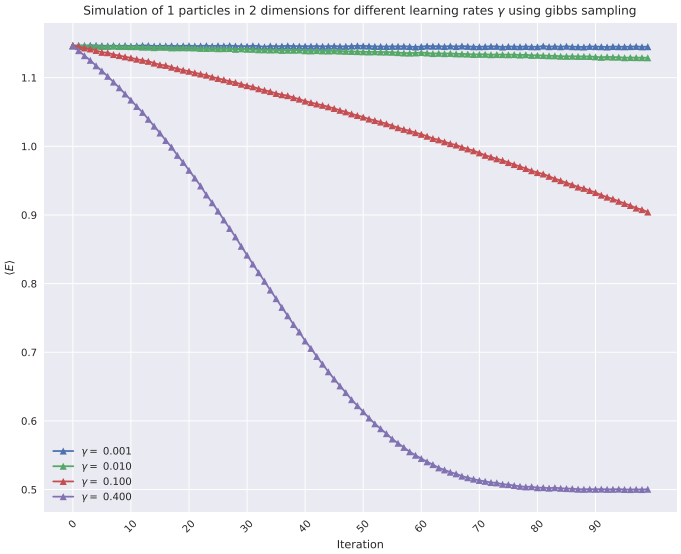
8 Conclusion

Herein lies the conclusions of yonder project, verily I say!

A Figures

A.1 1f)





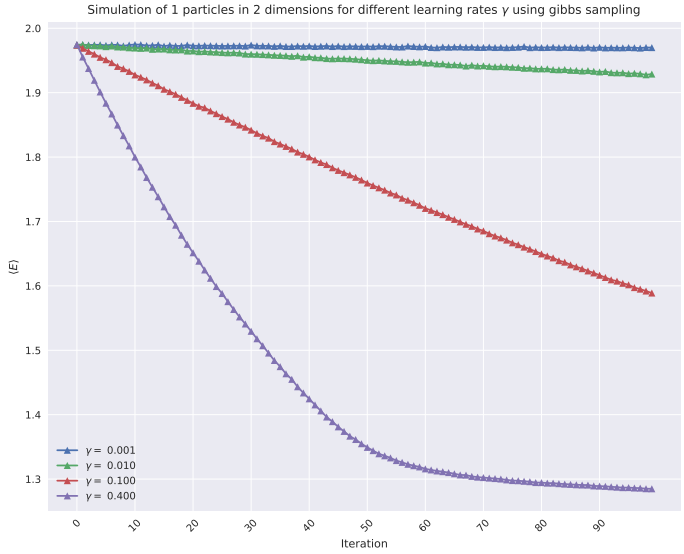
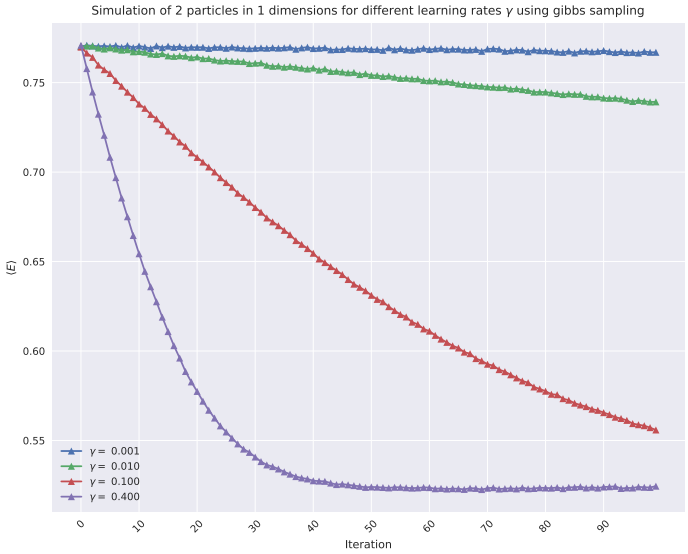


Figure 1: Plots for simulations with 1 particle and 1-3 dimensions using Gibbs sampling with different training rate values. Number of simulations for optimizing parameters are showed in x-axis.(The lowest figure is for 1 particle in 3 dimensions, wrong title.)



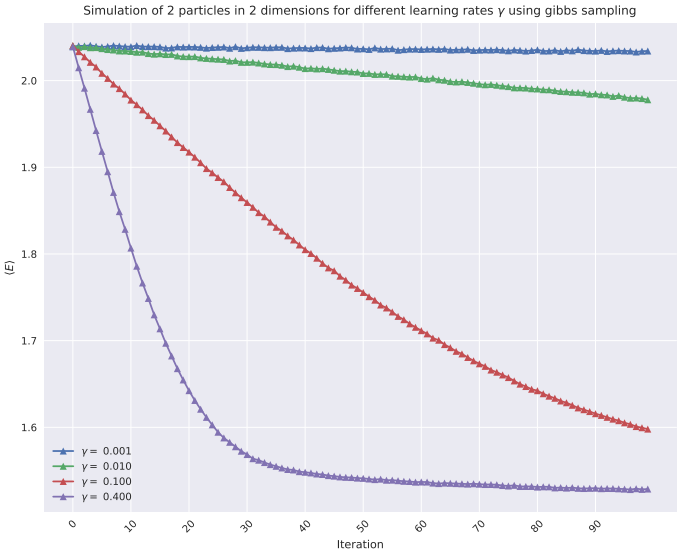


Figure 2: Plots for simulations with 2 particle and 1-3 dimensions using Gibbs sampling with different training rate values. Number of simulations for optimizing parameters are showed in x-axis.

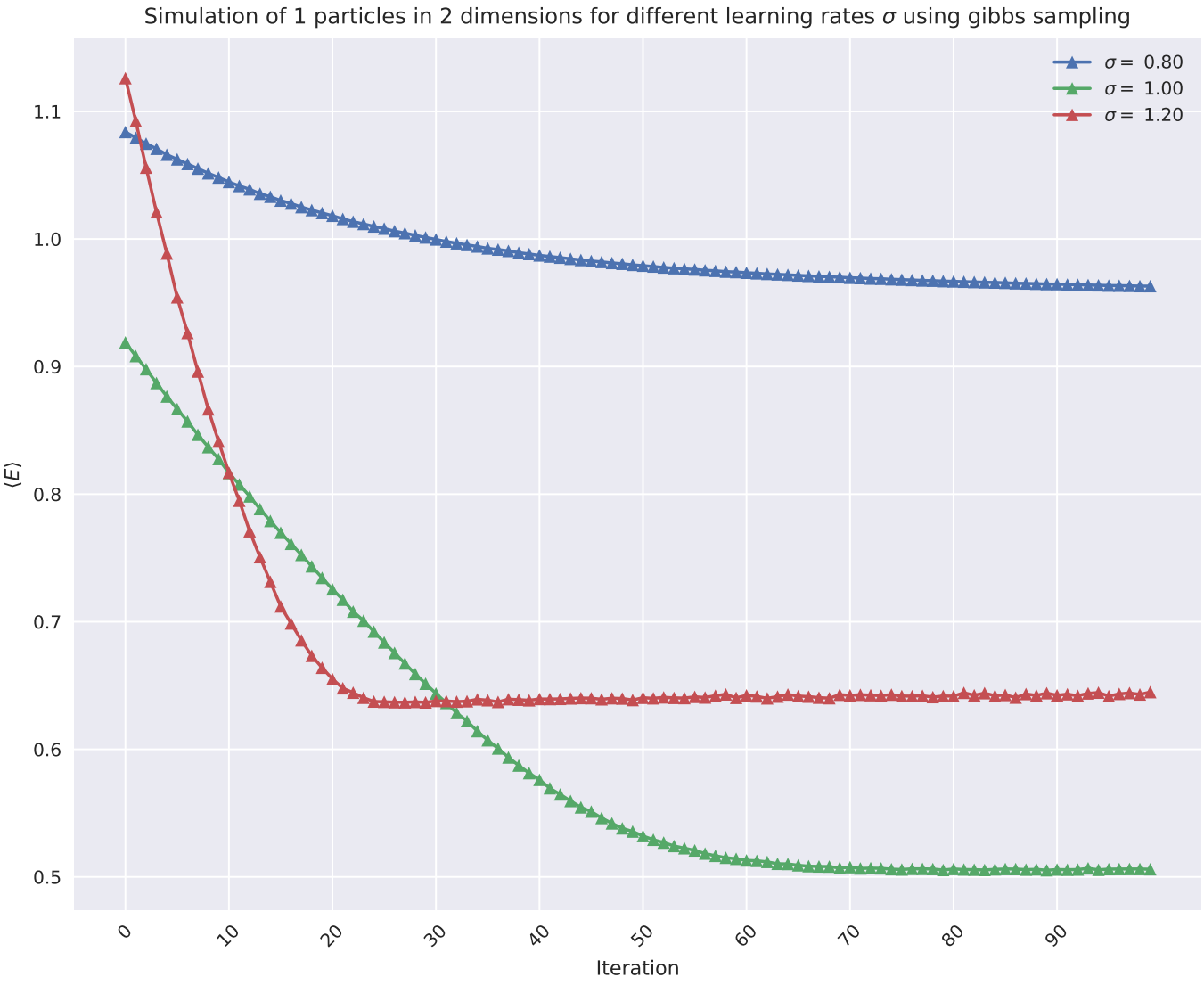


Figure 3: Plot of simulation with 1 particle in 2 dimensions using Gibbs sampling with different standard deviation of the nois of the visible nodes, σ.

B Local energy with $\psi(x) = F_{rbm}$

Given the Hamiltonian in eq. (REF EQ FROM ABOVE) the expression for the local energy is given by the following.

$$\begin{aligned} E_L &= \frac{1}{\Psi}\hat{H}\psi \\ &= \sum_k^P \frac{1}{2}\left(-\frac{1}{\Psi}\nabla_k^2\Psi + \omega^2 r_k^2\right) + \sum_{k<l}\frac{1}{r_{kl}} \end{aligned}$$

We wish to find the Laplacian of the NQS wave function. Since our visible nodes represents the particle positions in Cartesian coordinates the Laplace operator takes the second partial derivative with respect to each independent variable in the vector space. It can therefore be written as

$$\nabla_k^2\Psi = \sum_l^D \frac{\partial^2}{\partial x_{kl}^2}\Psi.$$

$$E_L = \sum_k^P\sum_l^D \frac{1}{2}\left(-\frac{1}{\Psi}\frac{\partial^2}{\partial x_{kl}^2}\Psi + \omega^2 r_k^2\right) + \sum_{k<l}\frac{1}{r_{kl}}$$

$$E_L = \sum_i^M \frac{1}{2}\left(-\frac{1}{\Psi}\frac{\partial^2}{\partial X_i^2}\Psi + \omega^2 X_i^2\right) + \sum_{k<l}\frac{1}{r_{kl}}$$

In the above equation we have used the fact that the number of visible nodes are $M = P \cdot D$. This reduces the number of sums in the first term to one. Also we see that it would be an advantage to solve the derivatives of $\ln\Psi$ since our wave function consist of sums in the exponential and also removes the Ψ in the denominator. Thus the identity below is helpful.

$$\frac{1}{\Psi}\frac{\partial^2}{\partial X_i^2}\Psi = \left(\frac{\partial}{\partial X_i}\ln\Psi\right)^2 + \frac{\partial^2}{\partial X_i^2}\ln\Psi$$

$$E_L = \sum_i^M \frac{1}{2}\left(-\left(\frac{\partial}{\partial X_i}\ln\Psi\right)^2 + \frac{\partial^2}{\partial X_i^2}\ln\Psi + \omega^2 X_i^2\right) + \sum_{k<l}\frac{1}{r_{kl}}$$

Our local energy can be rewritten and we must now solve both the first and second derivative of the wave function. Our wave function is represented by the marginal probability F_{rbm}.

$$
\begin{aligned}
\Psi(X) &= F_{rbm}(X) \\
&= \frac{1}{Z} \exp\left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}\right) \prod_j^N \left(1 + \exp\left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right)\right)
\end{aligned}
$$

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

Since Z is the normalization constant its derivative is zero and can be removed. We end up with the following

$$
\begin{aligned}
\nabla_i \ln \Psi &= \nabla_i \left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2} + \sum_j^N \ln\left(1 + \exp\left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right)\right)\right) \\
&= \frac{\partial}{\partial X_i} \left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2} + \sum_j^N \ln\left(1 + \exp\left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right)\right)\right) \\
&= \frac{-2(X_i - a_i)2\sigma^2}{(2\sigma^2)^2} + \sum_j^N \frac{1}{1 + \exp\left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right)} \left(\frac{\omega_{ij}}{\sigma^2}\right) \exp\left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right) \\
&= \frac{-(X_i - a_i)}{\sigma^2} + \frac{1}{\sigma^2} \sum_j^N \frac{\omega_{ij}}{1 + \exp\left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2}\right)}
\end{aligned}
$$

Given the first derivative we now find the second derivative. And we have solved the analytic case of the local energy.

$$
\begin{aligned}
\nabla_i^2 \ln \Psi &= \nabla_i^2 \left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2} + \sum_j^N \ln \left(1 + \exp \left(b_j + \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right) \right) \right) \\
&= \frac{\partial}{\partial X_i} \left(\frac{-(X_i - a_i)}{\sigma^2} + \frac{1}{\sigma^2} \sum_j^N \frac{\omega_{ij}}{1 + \exp\left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right)} \right) \\
&= -\frac{1}{\sigma^2} + \frac{1}{\sigma^2} \sum_j^N \frac{-\omega_{ij}}{\left(1 + \exp\left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right) \right)^2} \left(\frac{-\omega_{ij}}{\sigma^2} \right) \exp \left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right) \\
&= -\frac{1}{\sigma^2} + \frac{1}{\sigma^4} \sum_j^N \frac{\left(\omega_{ij} \right)^2}{\left(1 + \exp\left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right) \right)^2} \exp \left(-b_j - \sum_i^M \frac{X_i \omega_{ij}}{\sigma^2} \right)
\end{aligned}
$$

C Gradients with respect to RMB parameters, $\psi(x) = F_{rbm}$

To find the local energy minimum of the NQS wavefunction the gradient of the local energy with respect to the variational parameter $\alpha = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ needs to be computed. For each variational parameter $\alpha_i \in \alpha$ the gradient is defined as in equation 15

$$
G_i = \frac{\partial \langle E_L \rangle}{\partial \alpha_i} = 2 \left(\langle E_L \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \rangle - \langle E_L \rangle \langle \frac{1}{\Psi} \frac{\partial \Psi}{\partial \alpha_i} \rangle \right) \tag{15}
$$

While the NQS is defined for the Gaussian Binary RBM as (REPLACE WITH REF TO EQ)

$$
\Psi(\mathbf{X}) = \frac{1}{Z} \exp \left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2} \right) \prod_j^N \left(1 + \exp \left(b_j + \sum_i^M \frac{X_i w_{ij}^{\mathbf{T}}}{\sigma^2} \right) \right)
$$

We now find it useful to use the identity

$$
\frac{d}{dx} \ln f(x) = \frac{1}{f(x)} \frac{d}{dx} f(x) \tag{16}
$$

Taking the logarithm of the NQS wavefunction then gives

$$\ln(\Psi(\mathbf{X})) = \ln\frac{1}{Z} - \sum_{i}^{M}\frac{(X_i - a_i)^2}{2\sigma^2} + \sum_{j}^{N}\ln\left[1 + \exp\left(b_j + \sum_{i}^{M}\frac{X_i w_{ij}^{\mathbf{T}}}{\sigma^2}\right)\right] \tag{17}$$

The final result for the gradient of the components of α_i is then

$$\frac{\partial\ln(\Psi(\mathbf{X}))}{\partial a_k} = \frac{1}{\sigma^2}(X_k - a_k) \tag{18}$$

$$\frac{\partial\ln(\Psi(\mathbf{X}))}{\partial b_n} = \left(1 + \exp\left(-b_n - \sum_{i}^{M}\frac{X_i w_{in}^{\mathbf{T}}}{\sigma^2}\right)\right)^{-1} \tag{19}$$

$$\frac{\partial\ln(\Psi(\mathbf{X}))}{\partial w_{kn}} = \frac{X_k}{\sigma^2}\left(1 + \exp\left(-b_n - \sum_{i}^{M}\frac{X_i w_{in}^{\mathbf{T}}}{\sigma^2}\right)\right)^{-1} \tag{20}$$

D Local energy with $\psi(x) = \sqrt{F_{rbm}}$

In the case of the Gibbs sampling we represent the wave function as

$$\begin{aligned}\Psi(X) &= \sqrt{F_{rbm}(X)} \\ &= \sqrt{\frac{1}{Z}\exp\left(-\sum_i^M \frac{(X_i - a_i)^2}{2\sigma^2}\right)\prod_j^N \left(1 + \exp\left(b_j + \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)\right)}.\end{aligned}$$

Which is simply reduced to a constant in front of the term when taking the natural logarithm of the wave function.

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

Now we can easily see that the derivatives of the logarithm of the nqs wave function becomes,

$$\begin{aligned}\nabla_i \ln\Psi &= \nabla_i\left(-\sum_i^M \frac{(X_i - a_i)^2}{4\sigma^2} + \frac{1}{2}\sum_j^N \ln\left(1 + \exp\left(b_j + \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)\right)\right) \\ &= \frac{-(X_i - a_i)}{2\sigma^2} + \frac{1}{2\sigma^2}\sum_j^N \frac{\omega_{ij}}{1 + \exp\left(-b_j - \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)},\end{aligned}$$

$$\begin{aligned}\nabla_i^2 \ln\Psi &= \nabla_i^2\left(-\sum_i^M \frac{(X_i - a_i)^2}{4\sigma^2} + \frac{1}{2}\sum_j^N \ln\left(1 + \exp\left(b_j + \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)\right)\right) \\ &= -\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_j^N \frac{\left(\omega_{ij}\right)^2}{\left(1 + \exp\left(-b_j - \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)\right)^2}\exp\left(-b_j - \sum_i^M \frac{X_i\omega_{ij}}{\sigma^2}\right)\end{aligned}$$

and are the derivative to be replaced in the expression of the local energy when we employ the Gibbs sampling.

E Gradients with respect to RMB parameters, $\psi(x) = \sqrt{F_{rbm}}$

We also need to update the gradients to be used in the gradient descent method. These are also easily found, a simple constant i front of each term.

$$\frac{\partial \ln(\Psi(\mathbf{X}))}{\partial a_k} = \frac{1}{2\sigma^2}(X_k - a_k) \tag{21}$$

$$\frac{\partial \ln(\Psi(\mathbf{X}))}{\partial b_n} = \frac{1}{2}\left(1 + \exp\left(-b_n - \sum_i^M \frac{X_i w_{in}}{\sigma^2}\right)\right)^{-1} \tag{22}$$

$$\frac{\partial \ln(\Psi(\mathbf{X}))}{\partial w_{kn}} = \frac{X_k}{2\sigma^2}\left(1 + \exp\left(-b_n - \sum_i^M \frac{X_i w_{in}}{\sigma^2}\right)\right)^{-1} \tag{23}$$