

Project 1

Vilde Mari Reinertsen

1 Introduction

For this project in FYS4411, Computational Physics II: Quantum Mechanical Systems, we are using Variational Monte Carlo methods to study the ground state of a Bose-Einstein gas. This report includes a short explanation of how we have modelled the system with different approximations and potentials. Furthermore a short description of the different computational methods are included. Here the methods are explained using this specific problem as an example, to grasp the different methods. Finally, in the theoretical part of the report, the statistical aspects of the Variational Monte Carlo method are explored to be able to analyze the results and get a good error estimation.

The results are shown with an escalating amount of complexity. At first the simplest case is presented where we assume no interaction between the particles, we scan blindly over a range of parameters and we have no smart way of sampling the local energies. Gradually we include importance sampling, optimization and interaction while we compare the more complex situations with the starting point. Finally we calculate the one-body density of the Bose-Einstein gas with and without interaction and analyse the results.

2 Theory and method

In this project we want to evaluate the ground state energy of the Bose-Einstein gas. This system is a many-body system, therefore we have to solve the many-body time-independent Schrödinger equation,

$$\hat{H}\Psi = E\Psi, \quad (1)$$

to find the expectation value for the energy. Here Ψ is a function of the position of every particle, $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_N$.

The hamiltonian of the model is given by

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

which includes the kinetic energy, the external potential, V_{ext} , and the potential from interactions, V_{int} . We have assumed that the interactions are dominated and defined by only two-body interactions, i.e. two-body collisions.

In this project we have chosen to model the many-particle wavefunctions as a product of free single-

particle wavefunctions times a function f which represent their interaction,

$$\Psi_T(\mathbf{r}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha) = \left[\prod_k^N g(\mathbf{r}_k, \alpha) \right] \left[\prod_{j<i}^N f(a, |\mathbf{r}_j - \mathbf{r}_i|) \right], \quad (3)$$

where N is the number of particles and

$$g(\mathbf{r}_k, \alpha) = \exp[-\alpha(x_k^2 + y_k^2 + \beta z_k^2)]. \quad (4)$$

The Bose-Einstein gas is created by confining the bosons in a magnetic trap. We model this trap as a harmonic oscillator potential

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \quad (5)$$

which can be either spherical (S) or elliptical (E). Here ω_{ho}^2 is the trap potential strenght and in the elliptical case ω_{ho} is the trap frequency in the x- and y- direction while ω_z is the trap frequency in the z-direction.

The two-body collisions are modelled as a pair-wise, repulsive potential

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (6)$$

where a is the hard-core diameter of the bosons. This interaction potential results in the function

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \quad (7)$$

describing the interaction part of the wave functions in Eq. 3.

To simplify calcaultions we define a new function u to be $u = \ln(f)$, so that Eq. 3 is

$$\Psi_T(\mathbf{r}) = \left[\prod_k^N g(\mathbf{r}_k, \alpha) \right] \exp \left(\sum_{j<i} u(r_{ji}) \right)$$

where we have defined $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$.

2.1 Variational Monte Carlo

We are using the Variational Monte Carlo Method (VMC) to find the ground state energy of the system discribed above. We need to evaluate the expectation value of the energy for a trial wavefunction with the parameter, α , and then vary α to find the parameter that gives us the ground state energy, E_0 . We know that for all wave functions that are not the exact ground state wavefunction, Ψ_0 , the expectation energy will be larger than E_0 . Hence we vary α until we find the minimum energy.

We then have to solve Eq. 1 with regards to the energy, E ,

$$\langle E \rangle = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \int \frac{\Psi^*(\tau) \hat{H} \Psi(\tau)}{\int \Psi^*(\tau') \Psi(\tau') d\tau'} d\tau \quad (8)$$

where $d\tau = d\mathbf{r}_1 d\mathbf{r}_2 \dots d\mathbf{r}_N$. To use the VMC method we want to rewrite Eq. 8 to

$$\langle E(\alpha) \rangle = \int \frac{|\Psi^*(\tau)|^2}{\int |\Psi(\tau')|^2 d\tau'} E d\tau = \int P(\tau, \alpha) E d\tau \quad (9)$$

by introducing

$$P(\tau) = \frac{\Psi^*(\tau) \Psi(\tau)}{\int \Psi^*(\tau') \Psi(\tau') d\tau'} = \frac{|\Psi^*(\tau)|^2}{\int |\Psi(\tau')|^2 d\tau'}$$

which is the probability distribution of the energies. This is useful because, if we assume P to be a normal distribution, we can solve the integral by calculating the local energy, E_L , at random positions for every particle N and calculate the average of these local energies. The expectation value is then

$$\langle E(\alpha) \rangle \approx \frac{1}{M} \sum_{k=1}^M E_L(\tau_k, \alpha)$$

where M is the number of Monte Carlo cycles. This is a use of the VMC method.

2.2 Sampling

In this project sampling is about how to extract the local energy of the system, with the particles in random positions, in an efficient way. We start with a system of particles in random positions and we want to change one particle's positions, make a move, and then calculate the energy in the new situation. In particular, sampling is how we decide if a move is accepted or not. If it is accepted, the local energy (and other parameters) is sampled, if it is not accepted, the local energy in the old position is sampled. We have looked at two different methods to decide if a move is accepted or not, the Metropolis algorithm, the brute force method, and the Metropolis-Hastings method, importance sampling.

Both methods start by looking at a Markow chain

$$P_i(t + \epsilon) = \sum_j W(j \rightarrow i) P_j(t)$$

which is an expression for the probability of being in a state i after a time ϵ . The probability is found by adding the probability of transitioning from a state j which is given by the probability of being in a state j times the probability of making the transition from i to j , $W(i \rightarrow j)$. $W(i \rightarrow j)$ is modelled by separating it into the acceptance of a proposed move times the likelihood of making the transition from i to j , $W(i \rightarrow j) = A(i \rightarrow j) T(i \rightarrow j)$.

2.2.1 Brute force

In the brute force sampling method we assume that $T(i \rightarrow j) = T(j \rightarrow i)$ which leads to a move going to a state with higher probability if

$$\frac{P_j}{P_i} = \frac{W(i \rightarrow j)}{W(j \rightarrow i)} = \frac{A(i \rightarrow j)}{A(j \rightarrow i)} > 1$$

The brute force way of sampling works like this:

- Pick random particle
- Move by random amount (in one, two or three dimensions) times a step length, dl , to new position. The state where one of the particles is moved is the new state j
- Check if the move is accepted. It is accepted if a random number between 0 and 1 is bigger than P_j/P_i , i.e. if $P_j > P_i$ it is always accepted because it will then move to a more likely state. If $P_i > P_j$ it will sometimes make the move depending on the random number.

Depending of the step length, dl , the number of accepted moves will vary, but it is in general not a very effective method. We might waste many Monte Carlo (MC) cycles on moves that are not accepted.

2.2.2 Importance sampling

Importance sampling is a smarter way of sampling where we get more accepted moves. In this case we have $T(i \rightarrow j) \neq T(j \rightarrow i)$ which leads to

$$A(i \rightarrow j) = \min \left(1, \frac{P_i T(i \rightarrow j)}{P_j T(j \rightarrow i)} \right)$$

and we need a model for $T(j \rightarrow i)$.

To model $T(i \rightarrow j) = T(t, \mathbf{x})$ (we imagine that the MC cycles is time elapsing) we use the Fokker-Planck equation

$$\frac{\partial T}{\partial t} = D \frac{\partial^2 T}{\partial x^2} - D \frac{\partial F}{\partial x} T$$

which is the same as the heat equation and Fick's second law with the inclusion of a drift force, F , for one dimension. We are interested in moving towards a stationary state where $T(t, \mathbf{x})$ is not changing i.e. the maximum of the probability distribution. We therefore evaluate

$$\frac{\partial T}{\partial t} = 0 \implies 0 = \sum_i D \frac{\partial}{\partial x_i} \mathbf{e}_i \left(\frac{\partial}{\partial x_i} \mathbf{e}_i - \mathbf{F}_i \right) T$$

for several dimensions. From this we find the drift force to be

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

With importance sampling we use the drift force to move the particles to a position, i.e. a state, which has a higher probability, $P = |\Psi|^2$, and we also use it to change the way we accept the moves.

The importance sampling works like this:

- Pick random particle
- Move the particle according to the drift force, \mathbf{F} , to a position which is more has higher probability

using the expression

$$\mathbf{r}'_k = \mathbf{r}_k + D\Delta t \mathbf{F}(\mathbf{r}_1, \dots, \mathbf{r}_k, \dots, \mathbf{r}_N) + \xi \sqrt{\Delta t}$$

where $F(r_k) = \frac{2}{\Psi} \nabla_k \Psi$, ξ is a random number from a gaussian distribution, D is a constant and Δt is a variable called the time step.

- Check if the move is accepted using

$$A(r_1 \rightarrow r_2) = \min \left(1, \frac{g(r_1 \rightarrow r_2) P_{r_2}}{g(r_2 \rightarrow r_1) P_{r_1}} \right)$$

where g is Green's function

$$g(\mathbf{r}_k \rightarrow \mathbf{r}'_k) = \frac{1}{4D\Delta t^N} \exp \left(\frac{-(\mathbf{r}'_k - \mathbf{r}_k - D\Delta t \mathbf{F}(\mathbf{r}_k))}{4D\Delta t} \right)$$

which also includes the drift force.

With importance sampling more moves are accepted hence it is more effective in MC cycles, but it is less effective for each cycle because we have to calculate the drift force including the derivative of Ψ and the Green's function ratio for every MC cycle.

2.3 Optimization and gradient methods

Optimization is how to find the optimal parameters. In our case it is how to find the α which gives the minimum energy, the ground state energy, E_0 . We want to find the parameter in a more efficient way than just calculating the expectation energy for a range of α s and afterwards evaluate which α that gave the lowest energy.

We want to find the energy minimum, which naturally occur if the derivative of the expectation value with regards to α is zero and the double derivative is positive.

2.3.1 Newton-Raphson's root-finding algorithm

Newton-Raphson's root-finding algorithm is an iterative method of finding the root of an equation. We want to find the ground state wave function i.e. the α that gives the minimum energy. Then we have to find the root of the derivative of the local energy, i.e. the α where the derivative of the expectation value of the local energy is zero.

$$0 = \frac{\partial \langle E_L(\alpha) \rangle}{\partial \alpha}$$

We start with a guess for the root α_k , and extract a new guess from the tangent line of the derivative of the energy with respect to α . The tangent line, y , at α_k is:

$$y = \frac{\partial \langle E_L(\alpha_k) \rangle}{\partial \alpha} + (\alpha_{k+1} - \alpha_k) \frac{\partial^2 \langle E_L(\alpha_k) \rangle}{\partial \alpha^2}$$

The new approximation of the root, α_{k+1} , is found from where the tangent line is zero, $y(\alpha_{k+1}) = 0$

$$\alpha_{k+1} = \alpha_k - \left(\frac{\partial^2 \langle E_L(\alpha_k) \rangle}{\partial \alpha^2} \right)^{-1} \frac{\partial \langle E_L(\alpha_k) \rangle}{\partial \alpha}$$

We need to calculate the double derivative of the expectation value of the local energy, but this expression is difficult to and might involve many integrals which are expensive to calculate numerically. Therefore we used methods that approximate the double derivative.

2.3.2 Simple gradient descent

The simplest gradient descent method use Newton-Raphson's root-finding algorithm, but approximate the inverse of the double derivative with a constant, γ . Then we have the expression

$$\alpha_{k+1} = \alpha_k - \gamma \frac{\partial \langle E_L(\alpha_k) \rangle}{\partial \alpha}.$$

With this method the value for γ is found from trial and error.

2.3.3 Utilizing previous gradients

Another simple method which also include the previously calculated gradient gives this expression for the next approximation for α .

$$\alpha_{k+1} = \alpha_k - \tilde{\gamma}(1 - \beta) \frac{\partial \langle E_L(\alpha_k) \rangle}{\partial \alpha} - \beta \frac{\partial \langle E_L(\alpha_{k-1}) \rangle}{\partial \alpha}$$

Here we can choose how much weight we want to give the previous gradient by adjusting this new constant β . An optimal value of $\tilde{\gamma}$ has to be found just as γ had to be in the simpler version.

2.4 Statistical analysis and resampling

An important part of this project is statistical analysis. We have to know as much as we can about the errors of our calculated results. Since we are using a VMC method, this involves statistical analysis.

2.4.1 Evaluation of statistical errors

There several errors involved in this project. We can group them into systematical error and statistical errors. The systematic errors includes the errors from the acutally model we use to model reality. In our project with VMC this could be related to the step length, dl , in brute force sampling or the time step, Δt , in importance sampling. These errors are hard to investigate.

The other type of errors, the statistical ones, can be evaluated from common tools in statistics which is what we focused on in this project.

2.4.2 Statistical parameters

Table 1: A oversikt over the different statistical terms where M is the number of Monte Carlo cycles, p is the probability distribution and \bar{E}_L is the simple mean. *assume that $p(E_L)$ is a normal distribution and diskretized. For the covariance m is the number of data points in the data set X and n is the number of data points in the data set Y .

Expectation value	$\langle E_L \rangle = \int E_L p(E_L) dE_L$
Expectation value*	$\langle E_L \rangle = \bar{E}_L = \frac{1}{M} \sum_{i=1}^M E_L$
Variance	$\sigma_{E_L}^2 = \langle E_L - \langle E_L \rangle^2 \rangle = \langle E_L \rangle^2 - \langle E_L^2 \rangle$
Variance*	$\sigma_{E_L}^2 = \frac{1}{M} \sum_{i=1}^M (E_L - \bar{E}_L)^2$
Standard deviation	$\sigma_{E_L} = \sqrt{\sigma_{E_L}^2}$
Covariance	$cov(X_i, X_j) = \langle E_{L,i} - \langle E_{L,i} \rangle^2 \rangle \langle E_{L,j} - \langle E_{L,j} \rangle^2 \rangle = \langle E_{L,j} E_{L,i} \rangle - \langle E_{L,j} \rangle \langle E_{L,i} \rangle$
Covariance*	$cov(X, Y) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2} (x_i - x_j) (y_i - y_j)$

Table 1 shows the expression for the statistical terms in this project. All terms except the covariance is pretty much self-explanatory by their names. The covariance is a term that includes the measure of correlation in the dataset. If there is no correlation in the data set, the covariance is zero, just as the variance would be zero if you had found the exact energy.

To calculate the covariance directly, we need to evaluate the double sum in the last expression in Tab. 1, but because VMC gives many data points this is a very expensive task. Therefore, we want to use a method to approximate the covariance.

2.5 Resampling methods

Methods used to approximate the covariance are called resampling techniques. What you do is to take the data set you have from the numerical experiments and resample them, to extract the correlation in the data set.

In this project the correlation in the data set comes from our random number generator, because it is not truly random and the next random number is related to the previous one. The correlation leads to a larger error in the results than the normal variance shows. That is why it is important to approximate the covariance, so that a better estimate of the error is presented.

2.5.1 Blocking

Because VMC gives many data points we have used the blocking method as a resampling method, since it works best for large data sets. In the blocking method, you separate the data set into blocks and calculate the variance of the separate data sets. Afterwards, you calculate the variance of the different variances. Then you separate the previous blocks into smaller blocks and do the same thing with these new and smaller blocks. Eventually the change in the variances of the variances will decrease and this value is the approximation of the variance which includes the error from correlation in the data set.

3 Results and discussion

3.1 Analytical versus numerical evaluation of the double derivative

Table 2: A comparison of the CPU time of calculating the double derivative analytically and numerically. Here N is the number of particles and these calculations were performed in three dimensions. The numbers in the table are an average of 10 runs.

N	CPU time [s]		Ratio num/ana
	Analytical	Numerical	
1	1.6319	2.7882	1.7085
2	2.3090	8.2743	3.5835
4	3.5503	14.5833	4.1076
6	4.7517	29.1024	6.1246
8	6.0642	48.4739	7.9934
10	7.6771	67.4531	8.7863

3.2 Comparison with exact values

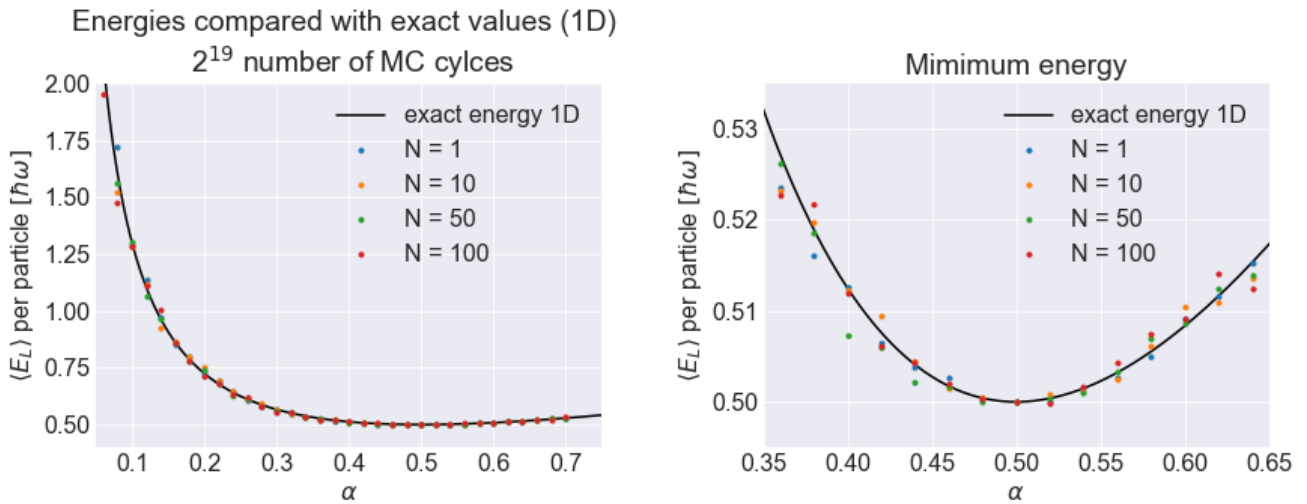


Figure 1: •

3.3 Brute force sampling

Table 3: 1 particle

α :	$\langle E_L \rangle$:	E_{exact}	σ_B	σ
0.35	1.59495	1.59643	0.00564	0.44597
0.40	1.53837	1.53750	0.00317	0.27486
0.45	1.50902	1.50833	0.00139	0.12902
0.50		1.50000		
0.55	1.50574	1.50682	0.00118	0.11680
0.60	1.52408	1.52500	0.00222	0.22306
0.65	1.54782	1.55192	0.00310	0.32800

Table 4: 10 particles

α :	$\langle E_L \rangle$:	E_{exact}	σ_B	σ
0.35	15.96823	15.96429	0.05513	1.45680
0.40	15.31296	15.37500	0.02946	0.86209
0.45	15.06932	15.08333	0.01372	0.40948
0.50		15.00000		
0.55	15.06036	15.06818	0.01129	0.37253
0.60	15.25915	15.25000	0.02057	0.72021
0.65	15.50985	15.51923	0.03065	1.00307

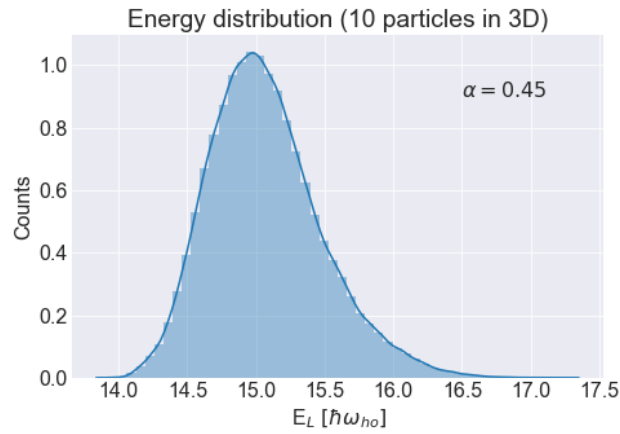


Figure 2: •

Table 5: 50 particles

α :	$\langle E_L \rangle$:	E_{exact}	σ_B	σ
0.35	79.49535	79.82143	0.23481	2.97403
0.40	77.08187	76.87500	0.12685	1.94938
0.45	75.34621	75.41667	0.05503	0.88031
0.50		75.00000		
0.55	75.20971	75.34091	0.05544	0.86965
0.60	76.10958	76.25000	0.09233	1.57130
0.65	77.71489	77.59615	0.13609	2.33112

Table 6: 100 particles

α :	$\langle E_L \rangle$:	E_{exact}	σ_B	σ
0.35	160.41867	159.64286	0.42227	4.47971
0.40	153.99383	153.75000	0.25800	2.78924
0.45	150.84608	150.83333	0.08956	1.23005
0.50		150.00000		
0.55	150.67186	150.68182	0.11373	1.26926
0.60	152.53009	152.50000	0.18938	2.15082
0.65	155.05236	155.19231	0.24124	2.99347

3.4 Including importance sampling

Figure 3 shows that with brute force sampling there seems to be a trade-off between the acceptance and the accuracy of the result. The right plot shows that larger steps, to a certain point, will give a better accuracy, but, as can be observed in plot to the left, the acceptance decreases with larger step lengths, dl . We also see that for smaller step sizes, the brute force sampling's accuracy is very poor, at least for 2^{20} number of MC cycles. With importance sampling, on the other hand, both acceptance and accuracy increases with smaller time steps (Δt). A time step at $\Delta t = 0.005 = 5 \cdot 10^{-3}$ seems to be a good choice with 2^{20} number of MC cycles according to Fig. 3.

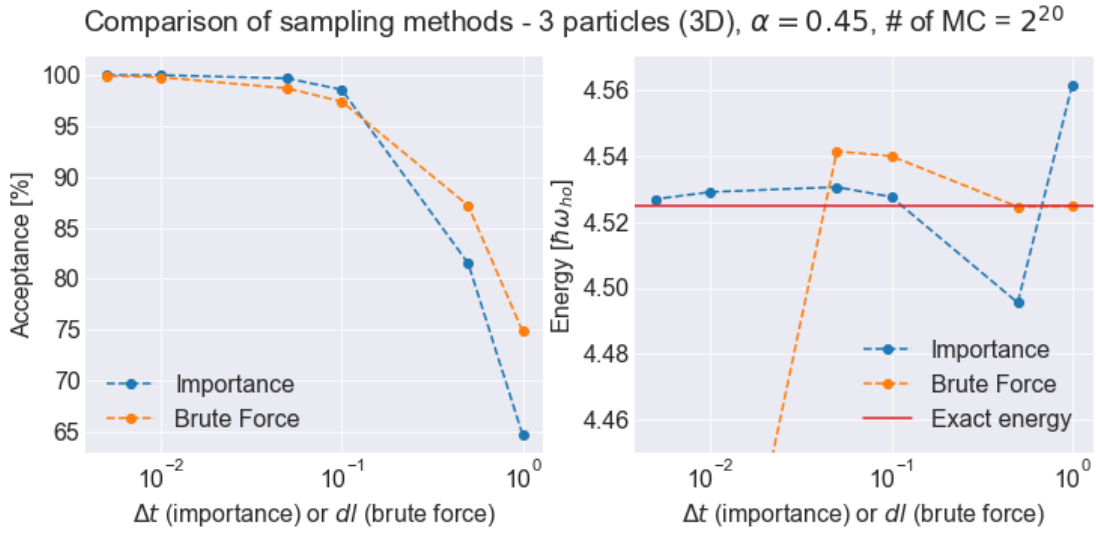


Figure 3: A comparison between brute force sampling and importance sampling. Left: The acceptance percent of suggested moved as a function of step length (dl) or time step (Δt). Right: The expectation value of the energy after 2^{20} steps and $\alpha = 0.45$ compared with the exact energy $\alpha = 0.45$.

$$E_L = -\frac{\hbar^2}{2m} \sum_k^N \left(\frac{\partial^2}{\partial x_k^2} \phi(x_k, y_k, z_k) + \frac{\partial^2}{\partial y_k^2} \phi(x_k, y_k, z_k) + \frac{\partial^2}{\partial z_k^2} \phi(x_k, y_k, z_k) \right) + \frac{1}{2} m \omega^2 \sum_k^N (x_k^2 + y_k^2 + z_k^2) \quad (10)$$

Appendices

A The derivatives and the local energy

In order to find the drift force and the local energy analytically we need to calculate both the derivative and the double derivative of the trial wavefunction.

A.1 The derivative of the trial wave function

We separate the total trial wave function (Eq. 3) into the onebody part and the interaction part,

$$\Psi_T = \Psi_{ob} \Psi_{in}. \quad (11)$$

Using the product rule, the derivative with regards to the particle k is

$$\nabla_k \Psi_T = \Psi_{ob} \nabla_k \Psi_{in} + \nabla_k \Psi_{ob} \Psi_{in}.$$

(Here the operator ∇_k only works on the first function after it.)

So we have to calculate $\nabla_k \Psi_{ob}$ and $\nabla_k \Psi_{in}$ and insert the expressions into the equation above.

We have (if $g(\mathbf{r}_k, \alpha) = \phi(\mathbf{r}_k)$)

$$\nabla_k \Psi_{ob} = \nabla_k \phi(\mathbf{r}_k) \prod_{i \neq k}^N \phi(\mathbf{r}_i) = \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \Psi_{ob} \quad (12)$$

using the chain rule.

The interaction part is a little more complicated. We start with

$$\nabla_k \Psi_{in} = \nabla_k \exp \left(\sum_{j < i} u(r_{ji}) \right) = \exp \left(\sum_{j < i} u(r_{ji}) \right) \sum_{l \neq k}^N u(r_{kl}) \nabla_k u(r_{kl})$$

. Because it is an exponential function we have to multiply the original function with the sum over all the terms in the exponent that are dependant on the particle k . Because $u(r_{kj}) = u(r_{jk})$ this sum is $\sum_{l \neq k}^N u(r_{kl})$. In addition we have to multiply with the derivative of $u(r_{kl})$ because of the chain rule. Since we have a sort of simplified way of showing the derivative (using the operator ∇_k), the expression $u(r_{kj}) \nabla_k u(r_{kj}) = \nabla_k u(r_{kj})$ by using the chain rule the opposite way. We then have the expression

$$\nabla_k \Psi_{in} = \exp \left(\sum_{j < i} u(r_{ji}) \right) \sum_{l \neq k}^N \nabla_k u(r_{kl}) = \sum_{l \neq k}^N \nabla_k u(r_{kl}) \Psi_{in}$$

for the derivative of the interaction part of the wave function.

The total expression of the derivative of the trial wave function is hence

$$\nabla_k \Psi_T = \Psi_{ob} \nabla_k \Psi_{in} + \nabla_k \Psi_{ob} \Psi_{in} \quad (13)$$

$$= \prod_{i \neq k}^N \phi(\mathbf{r}_i) \exp \left(\sum_{j < i} u(r_{ji}) \right) \sum_{l \neq k}^N \nabla_k u(r_{kl}) + \nabla_k \phi(\mathbf{r}_k) \prod_{i \neq k}^N \phi(\mathbf{r}_i) \exp \left(\sum_{j < i} u(r_{ji}) \right) \quad (14)$$

$$= \Psi_{ob} \sum_{l \neq k}^N \nabla_k u(r_{kl}) \Psi_{in} + \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \Psi_{ob} \Psi_{in} \quad (15)$$

$$= \left(\sum_{l \neq k}^N \nabla_k u(r_{kl}) + \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \right) \Psi_T \quad (16)$$

A.2 The double derivative of the trial wave function

The double derivative with regards to particle k is

$$\nabla_k^2 \Psi_T = \Psi_{ob} \nabla_k^2 \Psi_{in} + 2 \nabla_k \Psi_{ob} \nabla_k \Psi_{in} + \Psi_{ob} \nabla_k^2 \Psi_{in}. \quad (17)$$

So we have to calculate $\nabla_k^2 \Psi_{ob}$ and $\nabla_k^2 \Psi_{in}$, in addition to the derivatives from the previous section, and insert the expressions into the equation above.

From Eq. 12 we find

$$\nabla_k^2 \Psi_{ob} = \nabla_k^2 \phi(\mathbf{r}_k) \prod_{i \neq k}^N \phi(\mathbf{r}_i) = \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \Psi_{ob},$$

since $\prod_{k \neq i}^N \phi(\mathbf{r}_i)$ is independent of the particle k .

For the double derivative of Ψ_{in} we use the product rule again

$$\nabla_k^2 \Psi_{in} = \exp \left(\sum_{j < i} u(r_{ji}) \right) \nabla_k \left[\sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] + \nabla_k \left[\exp \left(\sum_{j < i} u(r_{ji}) \right) \right] \sum_{l \neq k}^N \nabla_k u(r_{kl}). \quad (18)$$

Here the square brackets are used to show what ∇_k applies to. So,

$$\nabla_k \left[\exp \left(\sum_{j < i} u(r_{ji}) \right) \right] = \exp \left(\sum_{j < i} u(r_{ji}) \right) \sum_{l' \neq k}^N \nabla_k u(r_{kl'})$$

as in the previous section. We calculate $\nabla_k u(r_{kl'})$ using the chain rule and get

$$\nabla_k u(r_{kl'}) = \frac{\mathbf{r}_k - \mathbf{r}_{l'}}{r_{kl'}} u'(r_{kl'}),$$

where $u'(r_{kl'}) = \frac{d}{dr_{kl'}} u(r_{kl'})$, because

$$\frac{d}{d\mathbf{r}_k} r_{kl'} = \frac{d}{d\mathbf{r}_k} \sqrt{(x_k - x_{l'})^2 + (y_k - y_{l'})^2 + (z_k - z_{l'})^2} = \frac{\mathbf{r}_k - \mathbf{r}_{l'}}{r_{kl'}}$$

and then

$$\nabla_k \left[\exp \left(\sum_{j < i} u(r_{ji}) \right) \right] = \exp \left(\sum_{j < i} u(r_{ji}) \right) \sum_{l' \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_{l'}}{r_{kl'}} u'(r_{kl'}) = \Psi_{in} \sum_{l' \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_{l'}}{r_{kl'}} u'(r_{kl'})$$

Next, we have

$$\nabla_k \left[\sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] = \nabla_k \left[\sum_{l \neq k}^N \Psi_{in} \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \right] = \sum_{l \neq k}^N \nabla_k \left[d \cdot \frac{a}{b} \cdot c \right] \quad (19)$$

where $a = (\mathbf{r}_k - \mathbf{r}_l)$, $b = r_{kl} = |\mathbf{r}_k - \mathbf{r}_l|$, $c = u'(r_{kl})$ and $d = \Psi_{in} = \exp \left(\sum_{i < j}^N u(r_{ij}) \right)$. Furthermore, $a' = \frac{d}{d\mathbf{r}_k} (\mathbf{r}_k - \mathbf{r}_l) = \frac{d}{dx_k} x_k + \frac{d}{dy_k} y_k + \frac{d}{dz_k} z_k = 3$ (the number of dimensions), $b' = \frac{d}{d\mathbf{r}_k} r_{kl} = \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}}$, $c' = \frac{d}{d\mathbf{r}_k} u'(r_{kl}) = u''(r_{kl}) \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}}$ and $d' = \Psi_{in} \sum_{l \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl})$. Inserting all these expressions into Eq. 19 gives (skipping some of the simplifications since the explanation is in the above part)

$$\nabla_k \left[\sum_{l \neq k}^N \nabla_k u(r_{kl}) \right] = \sum_{l \neq k}^N \left(\left(\frac{3r_{kl}}{(\mathbf{r}_k - \mathbf{r}_l)^2} - \frac{1}{r_{kl}} \right) u'(r_{kl}) + u''(r_{kl}) \right) \frac{(\mathbf{r}_k - \mathbf{r}_l)^2}{r_{kl}^2} \Psi_{in} \quad (20)$$

$$= \sum_{l \neq k}^N \left(\frac{2}{r_{kl}} u'(r_{kl}) + u''(r_{kl}) \right) \Psi_{in} \quad (21)$$

When we insert the results into Eq. 18 we get

$$\begin{aligned}\nabla_k^2 \Psi_{in} &= \Psi_{in} \sum_{l \neq k}^N \left(\frac{2}{r_{kl}} u'(r_{kl}) + u''(r_{kl}) \right) \Psi_{in} + \Psi_{in} \sum_{l' \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_{l'}}{r_{kl'}} u'(r_{kl'}) \Psi_{in} \sum_{l \neq k}^N \frac{\mathbf{r}_k - \mathbf{r}_l}{r_{kl}} u'(r_{kl}) \\ \nabla_k^2 \Psi_{in} &= \Psi_{in}^2 \left[\sum_{l \neq k}^N \left(\frac{2}{r_{kl}} u'(r_{kl}) + u''(r_{kl}) \right) + \sum_{l' \neq k}^N \sum_{l \neq k}^N \frac{(\mathbf{r}_k - \mathbf{r}_{l'}) (\mathbf{r}_k - \mathbf{r}_l)}{r_{kl'} r_{kl}} u'(r_{kl}) u'(r_{kl'}) \right]\end{aligned}$$

Inserting it all into Eq. 18 and dividing by the trial wavefunction (as we will do to find the local energy) gives

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

A.3 Local energy

To calculate the kinetic energy part of the local energy we use the last expression in the previous section, sum over all particles k . We use $\phi(\mathbf{r}_k)$ from Eq. 4 and find

$$\sum_k^N \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} = -2\alpha N d + 4\alpha^2 \sum_k^N \mathbf{r}_k^2$$

where d is the number of dimensions and $\mathbf{r}_k^2 = x_k^2 + y_k^2 + \beta z_k^2$. (This is the expression for the kinetic part of the local energy if there is no interaction.) Furthermore,

$$\sum_k^N \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} = -2\alpha \mathbf{r}_k.$$

We also have

$$\begin{aligned}u'(r_{kl}) &= -\frac{a}{ar_{kl} - r_{kl}^2} \text{ and} \\ u''(r_{kl}) &= \frac{a(a - 2r_{kl})}{r_{kl}^2(a - r_{kl})^2}\end{aligned}$$

for $r_{kl} > a$. The other case is not relevant because the local energy is never sampled if $r_{kl} < a$. Then the wave equation is zero. With this the local energy can be calculated analytically.

To get the potential energy part of the local energy the sum over all particles k is made with the relevant expression for the trap from Eq. 5. This is done for both the analytical and the numerical evaluation of the local energy.

References