FYS4411 - Project 1
Variational Monte Carlo

Daniel Heinesen, Gunnar Lange & Aram Salihi

April 10, 2018

Abstract

We present an investigation into bosons trapped in a harmonic oscillator trap.

Contents

1 Introduction

Confined quantum mechanical systems have received considerable attention in the literature, as they have many possible novel applications, ranging from qubits to energy storage. The existence of Bose-Einstein condensation in gases of certain alkali atoms, has sparked the interest in confined bosonic systems in particular (see e.g. [Nilsen et al., 2005]).

We present an investigation into a system of bosons trapped in a harmonic oscillator well, based on the work by [DuBois and Glyde, 2001]. We model our bosons as spheres with radius a, and investigate systems both with and without a repulsive interaction preventing them from moving through one another. We then use a variational Monte Carlo simulation to estimate the ground-state energy of our system.

We seek to understand how the ground-state energy of the system depends on various parameters such as the number of particles in the system, the shape of the trap and the way we sample in our variational Monte Carlo method. We also investigate metods for quickly determining an approximation to the optimal variational parameter. Finally, we present a brief discussion of onebody densities in our system.

2 Theoretical model

We investigate bosons in a potential trap. The Hamiltonian of such a system for N bosons is in general given by:

$$H = \sum_{i=1}^{N} \left(-\frac{\hbar^2}{2m} \nabla_i^2 + V_{ext}(\boldsymbol{r}_i) \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{1}$$

Where V_{ext} is the external potential, i.e. the potential of the trap, whereas V_{int} is the interal potential, i.e. the interaction potential between the bosons. We wish to approximate the ground state energy of our system described by the Hamiltonian given in equation 1.

2.1 The potentials employed

2.1.1 The external potential

We will focus on two different trap shapes: spherical and elliptical. The spherical trap can be described by a standard harmonic oscillator potential, which is given by:

$$V_{ext}(\boldsymbol{r}) = \frac{1}{2} m \omega_{ho}^2 r^2 \quad \text{(spherical trap)} \tag{2}$$

Where m is the mass of the bosons and ω is a parameter characterizing the strength of the potential (the characteristic frequency of the trap). The elliptical trap, however, is given by:

$$V_{ext}(\boldsymbol{r}) = \frac{1}{2} m [\omega_{ho}^2 (x^2 + y^2) + \omega_z z^2] \quad \text{(elliptical trap)} \tag{3}$$

I.e. we allow the strength of the potential to vary in one direction, which we designate as the z-axis.

2.1.2 The internal potential

We will begin by investigating models where there is no repulsion between the bosons, however, we will later act a repulsive potential that prevents bosons from occupying the same point in space. Here a is a typical diameter of our bosons (the hard-core diameter), which equals the scattering length of the single boson potential. We then define our internal potential as:

$$V_{int}(a, |\boldsymbol{r}_i - \boldsymbol{r}_j|) = \begin{cases} \infty & |\boldsymbol{r}_i - \boldsymbol{r}_j| \leq a \\ 0 & |\boldsymbol{r}_i - \boldsymbol{r}_j| > a \end{cases} \tag{4}$$

This potential thus represents the impossibility of bosons to occupy the same point in space.

2.2 The variational method

2.2.1 The variational principle

To find the ground state energy, E_0 of bosons trapped in a potential, we employ the variational method. The variational principle states that:

$$E_0 \leq \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi | \Psi \rangle} \tag{5}$$

Where H is the Hamiltonian of our system and Ψ_T is any trial wavefunction. We therefore choose a functional form for Ψ_T, with a number of free parameters. By varying these parameteres, we can find an upper bound for the ground state energy. If our chosen functional form is reasonably close to the exact wavefunction, Ψ of the system, then our estimate for E_0 should be reasonably close to the actual ground state energy of our system.

2.2.2 Choosing a trial wavefunction

As noted above, we must choose our trial wavefunction to be reasonably close to the expected form of the actual position wavefunctions of our system. We note first that the external potential represents a harmonic oscillator. For the harmonic oscillator, it is well known from introductory quantum mechanics that the eigenfunctions are exponential functions for each particle seperately. Note that the elliptic oscillator has a preferred direction (the z-axis). We accomodate for this by introducing a parameter β, representing the asymmetry of the elliptical oscillator. We therefore choose our trial wavefunction for this part of the potential to be:

$$h(\boldsymbol{r}_1, ..., \boldsymbol{r}_N, \alpha, \beta) = \prod_{i=1}^{N} g(\alpha, \beta, \boldsymbol{r}_i) = \prod_{i=1}^{N} \exp\left[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)\right] = \exp\left[-\alpha \sum_{i=1}^{N} (x_i^2 + y_i^2 + \beta z_i^2)\right] \tag{6}$$

Where we now have two variational parameters, α and β. Here β tunes the strength of the potential as a whole, whereas β tunes the asymmetry of the spherical oscillator.

For the internal potential, we expect a function that attenuates our wavefunction down to zero if the distance between any pair of particles becomes less than a. We choose this function to be continuous, on physical grounds. A simple choice for such a function is:

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

Putting this together, gives our trial wavefunction as:

$$\Psi_T(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) = \prod_{i} g(\alpha, \beta, \boldsymbol{r}_i) \prod_{i<j} f(a, |\boldsymbol{r}_i - \boldsymbol{r}_j|) \tag{8}$$

This is thus the wavefunction that we will be using in equation 5 to find our upper bound on the ground state energy.

2.3 Analytic results in the simplest case

For $\beta = 1$ (spherical trap) and $a = 0$ (no interaction amongst the particles), our system reduces to a simple harmonic oscillator, for which analytic results are available. For a single particle, the ground-state energy is (according to [Griffiths, 2004]) given by:

$$E = \frac{\hbar \omega d}{2} \tag{9}$$

Where d is the number of dimensions. For N non-interacting particles this turns into:

$$E = \frac{N \hbar \omega d}{2} \tag{10}$$

The exact ground-state wavefunction for the harmonic oscillator is (in dimensionless-form) given by:

$$\exp\left(-\frac{\sum_{i=1}^{d} x_i^2}{2}\right) \tag{11}$$

For N non-interacting particles this turns into:

$$\prod_{i=1}^{N} \exp\left(-\frac{\sum_{i=1}^{d} x_i^2}{2}\right) \tag{12}$$

Comparing this with our trial wavefunction in equation 8, shows that we should expect the ground-state energy to be achieved for:

$$\alpha = 0.5 \tag{13}$$

2.4 Variational Monte Carlo methods

Note that equation 5 involves two integrals over $3N$ dimensions. This is usually not possible to evaluate analytically for large N, and traditional numerical methods such as gaussian quadrature are far too slow in high dimensions to be feasible. We therefore employ variational Monte Carlo methods to help in evaluting these integrals. To do this, we consider these integrals to represents stochastic quantities, and define the probability distribution as:

$$P(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) = \frac{|\Psi_T|^2}{\int d\boldsymbol{r}_1 d\boldsymbol{r}_2 ... d\boldsymbol{r_n} |\Psi_T|^2} \tag{14}$$

Now define a new quantity, the local energy, as:

$$E_L(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) = \frac{1}{\Psi_T} H \Psi_T \tag{15}$$

The expectation value of the Hamiltonian (i.e. the estimate of the energy) now turns into:

$$E[H(\alpha, \beta)] = \int P(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) E_L(\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N, \alpha, \beta) d\boldsymbol{r}_1 d\boldsymbol{r}_2 ... d\boldsymbol{r}_N \approx \frac{1}{M} \sum_{i=1}^{M} E_{L,i} \tag{16}$$

Where now M is the number of Monte Carlo cycles, whereas $E_{L,i}$ is the local energy computed in the i-th Monte Carlo step. Note that the quality of this approximation is contingent upon our ability to sample our space. We employ two different algorithms for sampling our coordinate space.

2.4.1   The Metropolis algorithm

To move from one Monte Carlo step to the other, we will employ the Metropolis algorithm, as described in [Hjorth-Jensen, 2015]. The alogrithm works by sampling the probability distribution, but adding a bias to sample mostly in the regions where the probaility distribution actually is large, to avoid wasting CPU cycles. The algorithm works by proposing a step:

$$\boldsymbol{r}_{p,new} = \boldsymbol{r}_p + r \cdot \boldsymbol{dx} \tag{17}$$

Where r is a random number in $[0, 1]$ and $\boldsymbol{dx}$ is a chosen step vector. We then compute the ratio:

$$w = \frac{P(\boldsymbol{r}_1, ..., \boldsymbol{r}_{p,new}, ..., \boldsymbol{r}_n)}{P(\boldsymbol{r}_1, ..., \boldsymbol{r}_p, ..., \boldsymbol{r}_n)} = \frac{|\Psi_T(\boldsymbol{r}_1, ..., \boldsymbol{r}_{p,new}, ..., \boldsymbol{r}_n)|^2}{|\Psi_T(\boldsymbol{r}_1, ..., \boldsymbol{r}_p, ..., \boldsymbol{r}_n)|^2} \tag{18}$$

This ratio determines whether the new position has a higher or lower value of the PDF than the old position. We then accept the move if and only if $w \geq r$, where $r \in [0, 1]$ is a uniformly distributed random variable. This ensures that some steps that decrease the value of the PDF are accepted (giving a spread of value), but that we mostly sample where the PDF is large (which avoids wasting CPU cycles).

2.4.2   The Metropolis-Hastings algorithm (importance sampling)

The simple Metropolis algorithm described above assigns no preferred direction $\boldsymbol{dx}$ for the proposed step. This can be improved by adding a force term, "pushing" the Monte Carlo loop towards the region where the PDF is large. This is described in further detail in [Hjorth-Jensen, 2015], where it is shown that the expression fur such a quantum force is given, for particle k, by:

$$\boldsymbol{F}_k(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) = 2\frac{1}{\Psi_T}\nabla_k\Psi_T \tag{19}$$

We compute explicitly the expression for $\boldsymbol{F}$, using the trial wavefunction in equation 8, in appendix A. This gives:

$$\boldsymbol{F}_k = -4\alpha\boldsymbol{Q}_k + 2\sum_{j \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}u'(r_{kj}) \tag{20}$$

A more detailed analysis (given in [Hjorth-Jensen, 2015]) shows that the correct way to incorporate such a term into the Metropolis algorithm is by adding a transition probability term given by the Green's functions:

$$G(\boldsymbol{r}_1, ..., \boldsymbol{r}_N, \boldsymbol{r'}_1, ..., \boldsymbol{r'}_N) = \frac{1}{(4\pi D\Delta t)^{3/2}}\sum_{i=1}^{N}\exp\left(-\frac{(\boldsymbol{r}_i - \boldsymbol{r'}_i - D\Delta t\boldsymbol{F}_i(\boldsymbol{r'}_1, ..., \boldsymbol{r'}_N))^2}{4D\Delta t}\right) \tag{21}$$

Where $\boldsymbol{r}_i$ are the coordinates prior to the move and $\boldsymbol{r'}_i$ are the coordinates after the move, D is a diffusion constant (equal to $1/2$) and Δt is a step parameter that we tune. The effect of this term is to modulate the acceptance probability, w. This term is now given by:

$$w(\boldsymbol{r}_1, ..., \boldsymbol{r}_N, \boldsymbol{r'}_1, ..., \boldsymbol{r'}_N) = \frac{G(\boldsymbol{r}_1, ..., \boldsymbol{r}_N, \boldsymbol{r'}_1, ..., \boldsymbol{r'}_N)|\Psi_T(\boldsymbol{r}_1, ..., \boldsymbol{r}_N)|^2}{G(\boldsymbol{r'}_1, ..., \boldsymbol{r'}_N, \boldsymbol{r}_1, ..., \boldsymbol{r}_N)|\Psi_T(\boldsymbol{r'}_1, ..., \boldsymbol{r'}_N)|^2} \tag{22}$$

Which leads to the modified Metropolis algorithm, the Metropolis-Hastings algorithm, where a move is accepted if and only if $w \geq r$ where $r \in [0, 1]$ again is a uniformly distributed random variable. This method is computationally far more expensive than the brute-force Metropolis algorithm, but it leads to a significantly larger number of accepted steps.

2.5 The local energy

Note that in order to estimate the expectation value of the Hamiltonian we must, according to equation 16 compute the local energy, given by equation 15, in every Monte Carlo step. The local energy can be rewritten as:

$$E_L = \frac{1}{\Psi_T} H \Psi_T = \sum_{i=1}^{N} \left(-\frac{\hbar^2}{2m\Psi_T} \nabla_i^2 \Psi_T + V_{\text{ext}}(\boldsymbol{r}_i) \right) + \sum_{i<j}^{N} V_{\text{int}} \tag{23}$$

We compute this numerically, but we also include an analytic computation, using our trial wavefunction Ψ_T defined in equation 8. This is done in detail in appendix A, and gives:

$$\begin{aligned}
E_L = -\frac{\hbar^2}{2m} \sum_k &\left[4\alpha^2 \boldsymbol{Q}_k^2 - 4\alpha \boldsymbol{Q}_k + \sum_{j\neq k} \left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}} u'(r_{kj}) \right) \right. \\
&+ \sum_{i,j\neq k} \left(\frac{\boldsymbol{r} \cdot \boldsymbol{r} + \boldsymbol{r}_j \cdot \boldsymbol{r}_i - \boldsymbol{r}_j \cdot \boldsymbol{r}_k - \boldsymbol{r}_k \cdot \boldsymbol{r}_i}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \right) \\
&+ \sum_{j\neq k} \left. \left(u''(r_{kj} + \frac{2}{r_{kj}} u'(r_{kj})) \right) \right] + \sum_k V_{ext}(\boldsymbol{r}_k) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)
\end{aligned} \tag{24}$$

Where $\boldsymbol{Q}_k = \left(x_k \hat{i} + y_k \hat{j} + \beta z_k \hat{k} \right)$.

2.6 Gradient descent method to find the optimal parameter α

To reduce our search space, we use a minimization method to find an approximation to the optimal variational parameter, α. This minimization method is run with relatively few Monte Carlo steps, and once an approximate optimal parameter is found, we perform the analysis with many Monte Carlo steps. As our parameter space is one-dimensional (α is the only variational parameter), we choose to implement a simple gradient descent algorithm. This is implemented as follows:

$$\alpha_{n+1} = \alpha_n - \gamma_n \frac{\partial E_L}{\partial \alpha}(\alpha_n) \tag{25}$$

We choose an initial α_0 close to the correct α from the non-interacting case, and then iteratively follow wherever the gradient is lower. We use an adaptive step-size, which decreases as the difference between subsequent gradients decrease (i.e. as we get closer to the minimum). We follow [Barzilai and Borwein, 1988] for this, using:

$$\gamma_n = \frac{(\alpha_n - \alpha_{n-1}) + \epsilon}{\left[\frac{\partial E_L}{\partial \alpha}(\alpha_n) - \frac{\partial E_L}{\partial \alpha}(\alpha_{n-1}) \right]} \tag{26}$$

Where we, however, implement add an additional parameter ϵ (compared to [Barzilai and Borwein, 1988]) to ensure that we do not get stuck at the same α values. Furthermore, we also implement an additional test that checks if $\frac{\partial E_L}{\partial \alpha}(\alpha_n) - \frac{\partial E_L}{\partial \alpha}(\alpha_{n-1})$ is less than a treshold, in which case we reset γ to the initial guess γ_0.

To implement this, we require the derivative of the local energy. We show in appendix D that this is given by:

$$\frac{\partial E_L}{\partial \alpha}(\alpha_n) = 2 \left(\langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} E_L[\alpha_n] \rangle - \langle \frac{1}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} \rangle \langle E_L[\alpha_n] \rangle \right) \tag{27}$$

Where we note that both the local energy and the wavefunction depend on α.

2.7 One-body densities

We are interested in investigating the probability of finding a single particle at a distance r from the origin of our system. This is an important quantity, as it can be measured experimentally. We approach this through the concept of the one-body density. The one-body density is defined by:

$$\rho(\boldsymbol{r_1}) = \int d\boldsymbol{r_2}...d\boldsymbol{r_N} |\Psi_T(\boldsymbol{r_1}, \boldsymbol{r_2}, ..., \boldsymbol{r_N})|^2 \tag{28}$$

Where $\boldsymbol{r_1}$ is the position of any particle. The integral is evaluated using Monte Carlo integration, as described in further detail in section 3.7.

3 Methods

3.1 A note on units

The scattering-length for a typical BEC condensate atom is somewhere in the range $85 < a < 140a_0$, where a_0 is the Bohr radius. The characteristic trap length (at 0 K), however, is $a_{ho} = (\hbar/m\omega)^{1/2} \approx 10^4 \text{Å} - 2 \cdot 10^4 \text{Å}$. We introduce length units of a_{ho} and energy units of $\hbar\omega$. We show in appendix B that, using these units, the Hamiltonian can be rewritten as:

$$H = \sum_{i=1}^{N} \left(-\nabla_i^2 + x_i^2 + y_i^2 + \gamma z_i^2 \right) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{29}$$

Where $\gamma = \omega_z/\omega$.

3.2 Implementing the interacting case

To implement the interacting case ($a \neq 0$), we follow the parameters used by [Nilsen et al., 2005]. We therefore let $a = 0.0043$ and $\omega_z = \beta = 2.82843$. We find the optimal α in the interacting case by using the gradient descent method described in section 2.6, using 100 times fewer Monte Carlo cycles in the gradient descent calculations compared to the full simulation.

3.3 Outline of the structure of our program

Our program uses the `Eigen` library in `C++` to keep track of the matrices. We implement three main classes:

main.cpp   This class reads in the parameters from an input file, sets up parallelization in MPI and starts a new simulation.

simulation.cpp   This class contains the gradient descent method, as well as a method that actually runs the main Monte Carlo loop. The onebody densities are also computed in this class.

system.ccp   This class distributes the particles in the system, sets up the grid, calculates the wavefunction, proposes new steps and checks the acceptance of the steps.

3.4   Proposing a new position

To propose a new step, we use a different method for the brute force method versus the importance sampling method. For the brute force method, we propose a new step as:

$$\boldsymbol{r}^i_{\text{new}} = \boldsymbol{r}^i_{\text{old}} + \boldsymbol{\xi}dx \tag{30}$$

Where $\boldsymbol{r}^i_{\text{old}}$ is the previous position of particle i and $\boldsymbol{xi}$ is a normally distributed variable with mean 0 and standard deviation 1. We tune the parameter dx for the brute force simulation, and find that we get good results for $dx = 1.5$.

For importance sampling, however, we propose a different step to keep the statistics consistent (as discussed in [Jonsson, 2018]). The new step is then given by:

$$\boldsymbol{r}^i_{\text{new}} = \boldsymbol{r}^i_{\text{old}} + \boldsymbol{\xi}\sqrt{dt} + D\boldsymbol{F}_idt \tag{31}$$

Where F_i is the quantum force on particle i and D is the diffusion parameter, discussed in section 2.4.2. The parameter dt is the same parameter used in the Greens function (equation 21), and we find it by looking at the energy for the exactly solvable system for different dt.

3.5   Finding the optimal step size

For the brute-force Monte Carlo sampling, we must estimate an optimal step size, dx, with which to move our step. For the Monte Carlo simulation with importance sampling, we must similarly estimate the parameter dt. These step sizes should ensure that:

- The simulation quickly converges to the correct energy

- The variance quickly stabilizes

To find these properties, we look at how the variance and the mean of the local energy (across every Monte Carlo sample) varies as we increase the number of Monte Carlo cycles. We investigate various values of dx (respectively dt). For each of these values, we ensure that we manage to hit an energy minimum at $\alpha = 0.5$ in the noninteracting case.

We measure the variance in the standard way (i.e. not using blocking), as blocking requires the number of steps to be a power of 2. We do this for the noninteracting system, using $\alpha = 0.45$ (as the exact α gives essentially 0 variance).

We then choose the dx (respectively dt) that ensure that the variance and the mean quickly stabilize.

3.6   Statistical treatment of the data

We wish to estimate the statistical errors of our estimate of the local energy. As subsequent values for the local energy are based on the previous values of the local energy, we are dealing with a correlated time series. The normality assumptions usually employed when generating confidence intervals therefore does not hold. Instead, we use blocking to estimate the error in our energy. A detailed mathematical treatment of this method is found in [Jonsson, 2018]. We therefore present only the algorithm. Assume that we have n observations $Q^0 = \{Q^0_1, ..., Q^0_n\}$ of the local energy, and we wish to estimate the variance of the local energy. The uncorrelated estimator for the standard deviation is given by:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Q_i - \overline{Q})^2} \tag{32}$$

Where $\overline{Q}$ is the mean of the local energy estimates. This expression is only valid in the uncorrelated case, however. The idea of blocking is to perform iterative transformations of the data, until the

covariance of the data is approximately zero. Specifically, the transformation that we will be implementing is the following:

$$Q^i \to Q^{i+1} : Q^{i+1} = \left\{ \frac{Q_1^i + Q_2^i}{2}, \frac{Q_3^i + Q_4^i}{2}, ..., \frac{Q_{n-1}^i + Q_n^i}{2} \right\} \tag{33}$$

We choose n to be a power of a, i.e. $n = 2^d$. Thus, we can perform this transformation a maximum of d times. We now define an estimator for the covariance bewteen observations that are distance h apart by:

$$\hat{\gamma^k}(h) = \frac{1}{n} \sum_{i=1}^{n-h} (Q_i^k - \overline{Q}^k)(Q_{i+h}^k - \overline{Q}^k) \tag{34}$$

We are interested in knowing when $\gamma^k(1) \approx 0$. This can be found by hand, simply by realizing that the transformation $Q^i \to Q^{i+1}$ described above preserves the variance of the data. It is shown in [Jonsson, 2018] that $\gamma^k(1) \to 0$ for successive iterations, and therefore a plot of the empirical variance as a function of iterations should eventually stabilize, giving the variance.

We use an alternative, automatic, approach to estimate when the covariance is small. This is based on comparing our results to a χ^2 distribution, and is described further in [Jonsson, 2018].

3.7 Implementing the onebody density

To compute the one-body density defined by equation 28, we must compute the integral. We show that in appendix C that, if we only consider the radial distribution of the onebody density, this integral can be discretized as:

$$\rho(r_i) \approx \frac{N_i}{4\pi r_i^2 \Delta r} \tag{35}$$

Where N_i is the number of particles in the spherical shell with radius r_i to $r_i + \Delta r$. Thus, to compute the onebody density, we run our Monte Carlo simulation and check in which shell the particle that we move at every step is located. We also check numerically that the normalization is correct.

3.8 Parallelizing our code

We note that our algorithms are embarrassingly parallel, as they do not need to communicate between one another to span a large portion of the parameter space. We therefore run 4 cores simultaneously, producing 4 times as much data. This data is simply appended **ASK DANIEL** in the blocking, giving 4 times more data, and thus better estimates of the mean and the standard deviation.

4 Results

4.1 Finding the optimal step size

We plot the mean and the variance for various steps sizes, as discussed in section 3.5. This gives the plot shown in figure 1 for the brute force sampling. As discussed further in section 5.1.1, we choose $dx = 0.5$ as our parameter for the brute-force case.

We produce a similar plot for the case with importance sampling. This gives the plots shown in figure 2. As explained in section 5.1.2, we choose $dt = 0.05$.
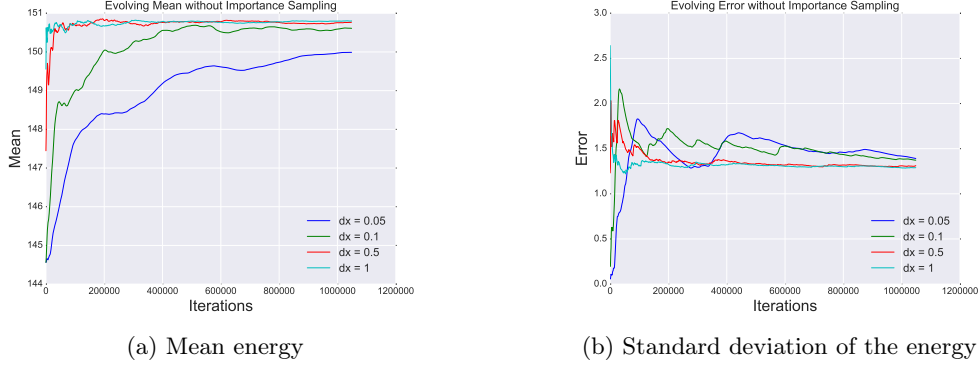
(a) Mean energy

(b) Standard deviation of the energy

Figure 1: Mean energy and standard deviation of the energy as a function of Monte Carlo cycles for various step sizes, dx. This was done without importance sampling in three dimensions, using $N = 100$, $a = 0$ (noninteracting), $\alpha = 0.45$, $\omega_z = 1$, $\beta = 1$,

(a) Mean energy

(b) Standard deviation of the energy

Figure 2: Mean energy and standard deviation of the energy as a function of Monte Carlo cycles for various step sizes, dx. This was done with importance sampling in three dimensions, using $N = 100$, $a = 0$ (noninteracting), $\alpha = 0.45$, $\omega_z = 1$, $\beta = 1$,
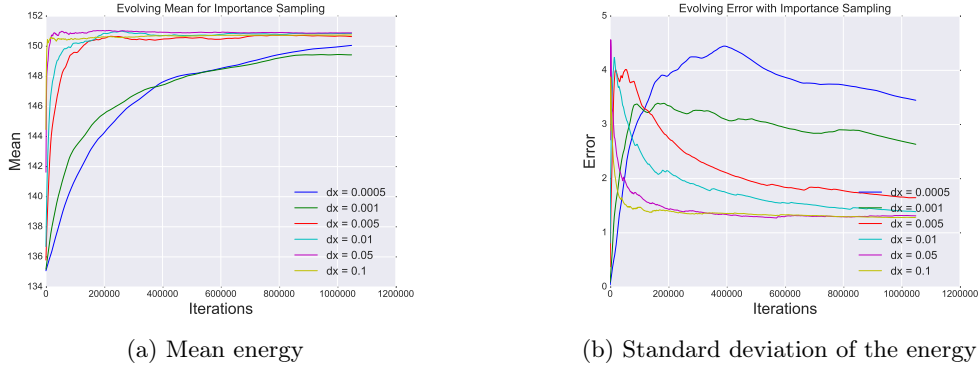
4.2 Testing our code

We test our code by running the simulations for the noninteracting case ($a = 0$) and for the correct value of the variational parameter, $\alpha = 0.5$. We then expect our code to reproduce the exact energy given in equation 10. We test this using both the analytic and the numeric energy, and time the CPU difference between them. We compute the uncertainty using the blocking method, as described in section 3.6. We show the results without importance sampling in table 1 (1D), table 2 (2D) and table 3 (3D). We do a similar analysis with importance sampling. This gives table 4 (1D), table 5 (2D) and table 6 (3D). As discussed further in section 5.3 , we opt to not use importance sampling in the interacting case.

4.3 Energy as a function of α

To investigate how our solution varies with α, we plot the local energy as a function of the variational parameter α. We first do this for the spherical harmonic oscillator ($\beta = 1$) with no interaction ($a = 0$), with $N = 100$ and 2^{20} Monte Carlo cycles per core (running with 4 cores). We also compute the error in each case, using the blocking method. This gives the plot shown in figure 3 (brute-force sampling) and figure 4 (importance sampling). We do a similar analysis for

N	Analytic energy, E_a [$\hbar\omega$]	Numeric energy, E_n [$\hbar\omega$]	Exact energy, E [$\hbar\omega$]	CPU time E_a [s]	CPU time E_n [s]
1	0.5	0.5	0.5	1.41	2.03
10	5	$5.00000 \pm 1 \cdot 10^{-5}$	5	1.47	9.13
100	50	50.000 ± 0.001	50	2.42	446.83
500	250	-	250	51.02	-

Table 1: Benchmarking of our code in the noninteracting case ($a = 0$), in one dimension, showing the analytic energy (computed by means of equation 24), the numeric energy and the exact energy (given by equation 10). This is for the spherical trap ($\omega_z = \beta = 1$) in one dimension. This was obtained using 4 cores, each running 2^{20} Monte Carlo cycles. The simulation with the numeric energy for $N = 500$ took too long to be feasible. Wherever no uncertainty is specified, blocking gave an error of 0. This is with brute force sampling.

N	Analytic energy, E_a [$\hbar\omega$]	Numeric energy, E_n [$\hbar\omega$]	Exact energy, E [$\hbar\omega$]	CPU time E_a [s]	CPU time E_n [s]
1	1	1	1	1.32	2.27
10	10	$10.00000 \pm 3 \cdot 10^{-5}$	10	1.49	15.11
100	100	100.000 ± 0.003	50	2.65	810.2
500	500	-	500	49.2	-

Table 2: Same table as table 1, but in 2 dimensions

N	Analytic energy, E_a [$\hbar\omega$]	Numeric energy, E_n [$\hbar\omega$]	Exact energy, E [$\hbar\omega$]	CPU time E_a [s]	CPU time E_n [s]
1	1.5	1.5	1.5	1.56	2.85
10	15	$15.00000 \pm 6 \cdot 10^{-5}$	15	1.91	29.89
100	150	149.997 ± 0.005	150	3.92	1791.83
500	750	-	750	54.36	-

Table 3: Same table as table 1, but in 3 dimensions

N	Analytic energy, E_a [$\hbar\omega$]	Numeric energy, E_n [$\hbar\omega$]	Exact energy, E [$\hbar\omega$]	CPU time E_a [s]	CPU time E_n [s]
1	0.5	0.5	0.5	1.83	2.36
10	5	$5.00000 \pm 2 \cdot 10^{-5}$	5	2.03	9.59
100	50	49.997 ± 0.001	50	4.92	441.62
500	250	-	250	56.83	-

Table 4: Benchmarking of our code in the noninteracting case ($a = 0$), in one dimension, showing the analytic energy (computed by means of equation 24), the numeric energy and the exact energy (given by equation 10). This is for the spherical trap ($\omega = \omega_z = \beta = 1$) in one dimension. This was obtained using 4 cores, each running 2^{20} Monte Carlo cycles. The simulation with the numeric energy for $N = 500$ took too long to be feasible. Wherever no uncertainty is specified, blocking gave an error of 0. This is with importance sampling.

N	Analytic energy, E_a [$\hbar\omega$]	Numeric energy, E_n [$\hbar\omega$]	Exact energy, E [$\hbar\omega$]	CPU time E_a [s]	CPU time E_n [s]
1	1	1	1	1.87	2.80
10	10	$10.00004 \pm 3 \cdot 10^{-5}$	10	2.16	16.86
100	100	99.998 ± 0.003	100	5.71	808.74
500	500	-	500	58.17	-

Table 5: Same as table 4 but in 2 dimensions.

$N = 10$ in the interacting case[1]. This gives the plot shown in figure 5 for brute-force sampling.

---

[1] Any higher N was far too slow.

N	Analytic energy, E_a $[\hbar\omega]$	Numeric energy, E_n $[\hbar\omega]$	Exact energy, E $[\hbar\omega]$	CPU time E_a [s]	CPU time E_n [s]
1	1.5	1.5	1.5	1.87	3.08
10	15	$14.99999 \pm 6 \cdot 10^{-5}$	15	2.66	26.26
100	150	149.997 ± 0.006	150	6.40	1672.32
500	750	-	750	58.55	-

Table 6: Same as table 4 but in 3 dimensions.
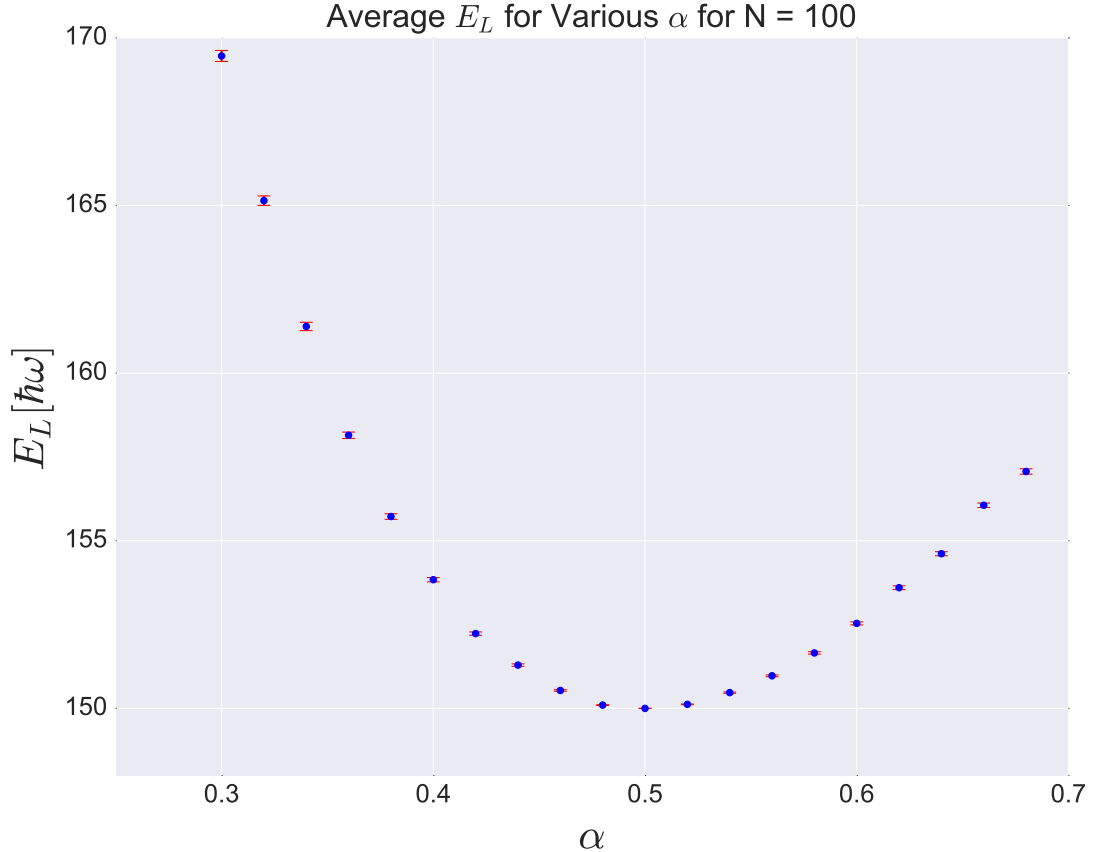
Figure 3: Local energy for various values of α, for $a = 0$, $\beta = 1$, $\omega_z = 1$, $N = 100$, 3 dimensions, 2^{20} Monte Carlo cycles and brute force sampling.

4.4 Finding the optimal α

We run our gradient descent method for various N in the fully interacting case, to find the optimal parameter. We first test our method by running it for the 3D noninteracting system of 10 particles, and ensuring that it finds the correct value at $\alpha = 0.5$. The result of this is shown in figure 6. The method finds $\alpha = 0.5$ in few time steps. We do a similar analysis for the interacting case with $N = 10, 50$ and 100. This is shown in figure 7 (N=10), figure 8 (N=50) and figure 9 (N=100).

4.5 Results from the fully interacting system

We now run the fully interacting system. We choose the parameters used by [Nilsen et al., 2005], and an α value found by the gradient descent. We do not use importance sampling, as it took too long. We use the analytic expression for the local energy, given in equation 24. This gives the
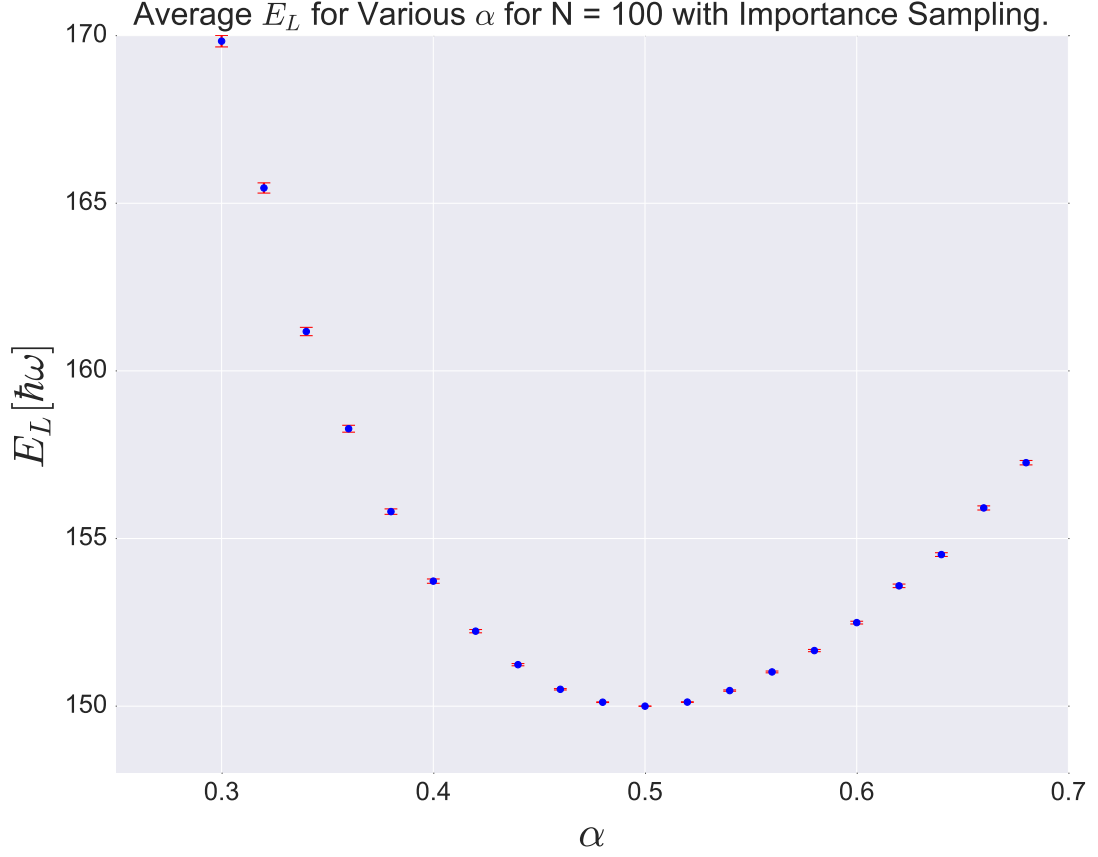
Figure 4: Local energy for various values of α, for $a = 0$, $\beta = 1$, $\omega_z = 1$, $N = 100$, 3 dimensions, 2^{20} Monte Carlo cycles and importance sampling.

N	α
10	0.499037
50	0.494033
100	0.474419

Table 7: Values for the variational parameter α for various N. This was estimated in 3 dimensions, with $a = 0.0043$, $\omega = 1$, $\omega_z = \beta = 2.82843$ and using brute force sampling with $2^{20}/100$ timesteps. We stopped the gradient descent method once it became stable, but after a minimum of 40 iterations.

values shown in table 8.

N	Local energy, E_L $[\hbar\omega]$	CPU time [s]
10	$24.14259 \pm 9 \cdot 10^{-5}$	78.4679
50	123.859 ± 0.001	4338.12
100	253.7 ± 0.1	31734.3

Table 8: Local energy in 3 dimensions for the fully interacting system. We use the parameters used by [Nilsen et al., 2005], i.e. $a = 0.0043$, $\omega_z = \beta = 2.82843$, $\omega = 1$. We use brute force sampling, and the parameters α found by our gradient descent method, shown in table 7.
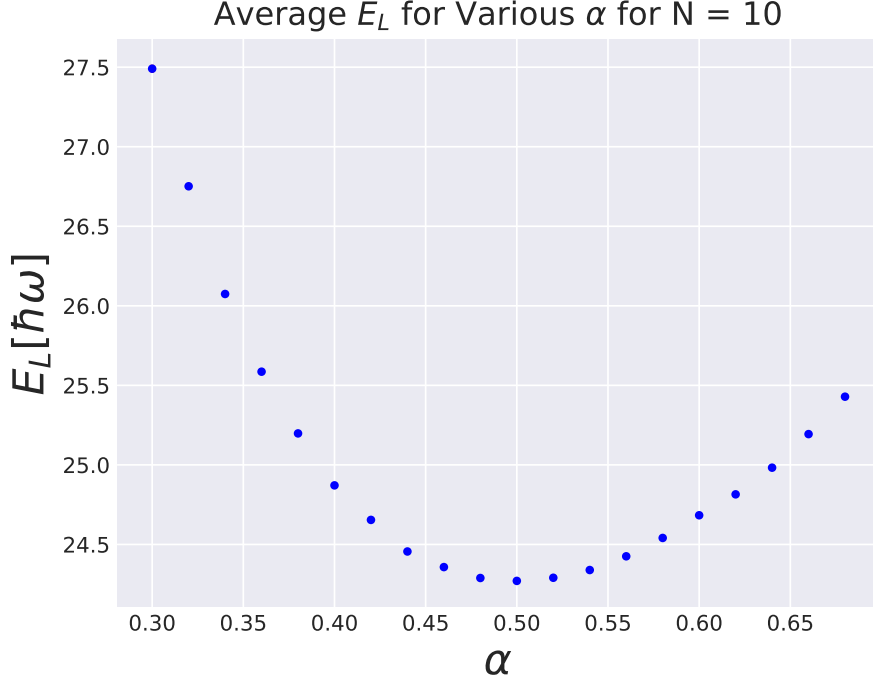
13

Figure 5: Local energy for various values of α, for $a = 0.0043$, $\omega = 1$, $\omega_z = \beta = 2.82843$, $N = 10$, 3 dimensions, 2^{20} Monte Carlo cycles and brute force sampling. The error bars are too small to be shown.

4.6 Onebody densities

We plot the onebody density for the interacting and noninteracting case in figure 10 below.

5 Discussion

5.1 The optimal step size

5.1.1 The brute force sampling

Looking at figure 1, we notice that the error stabilizes quickest for $dx = 0.5$ and $dx = 1$. Notice that this erorr is *not* estimated using blocking. Therefore, the absolute value of the error is misleading, and we only consider the trend. On a related note, we see that the mean also stabilizes equally quickly in both cases. We therefore choose the dx that gives the highest acceptance ratio, as this minimizes the wasted CPU cycles. This is the case for $dx = 0.5$, which explains our choice of step size for the brute-force sampling.

5.1.2 Importance sampling

The exact same argumentation as in the previous section for figure 2 shows that the error (again *not* corrected for correlations), stabilizes quickest for $dt = 0.05$ and $dt = 0.1$, and similarly for the mean. Again, we choose the timestep that gives the highest acceptance ratio, which gives $dt = 0.05$.
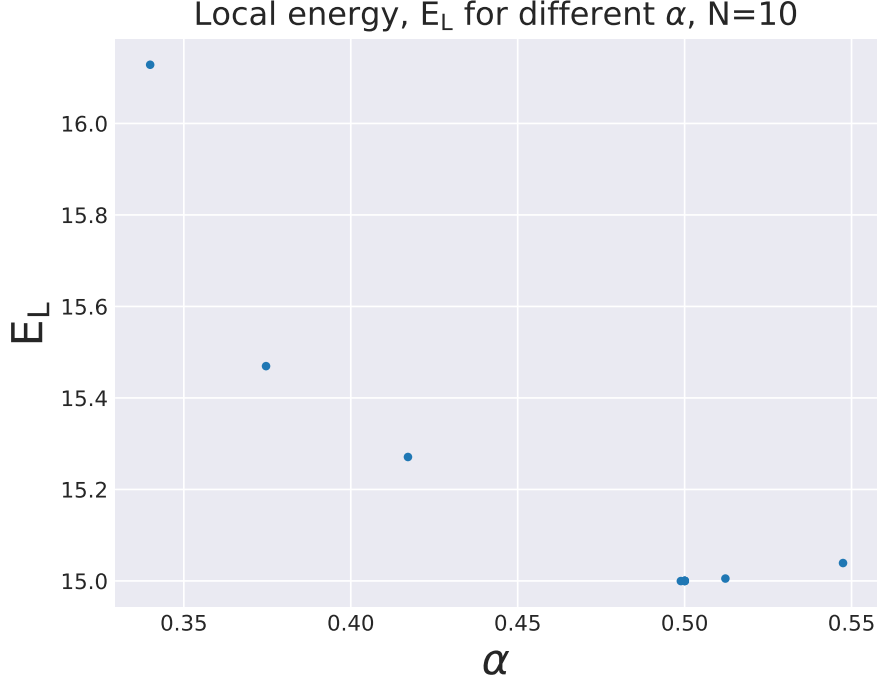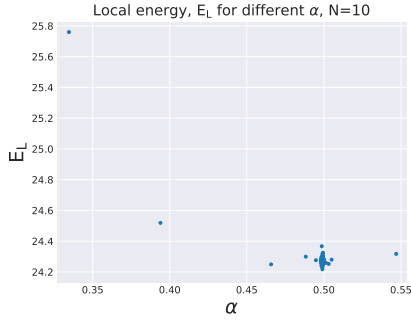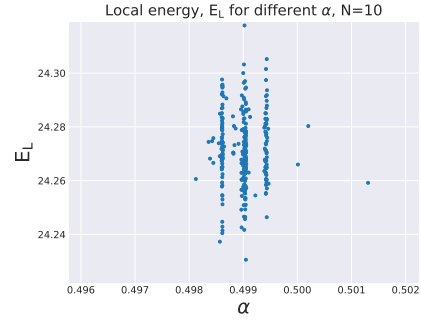
Figure 6: Gradient descent for finding the optimal α in the noninteracting case ($a = 0$), with $N = 10$, $\omega = \omega_z = \beta = 1$, 3 dimensions and brute-force sampling. We use a minimum step size, $\epsilon = 10^{-6}$, and an accept whenever the gradient is less than 10^{-6}.

(a) Gradient descent

(b) Zoomed in version

Figure 7: Gradient descent for finding the optimal α in the interacting case ($a = 0.0043$), with $N = 10$, $\omega = 1$, $\omega_z = \beta = 2.82843$, 3 dimensions and brute-force sampling. We use a minimum step size, $\epsilon = 10^{-6}$, and an accept whenever the gradient is less than 10^{-6}. We show also a zoomed-in version of the same plot.

5.2 Testing of our code

Looking at table 1, 2 and 6, we see that the numeric and the analytic energy both reproduce the exact energy, within the uncertainty. However, the analytic energy runs significantly faster, and has an error of 0 in the minimum. This justifies using the exact energy for the rest of our simulations.
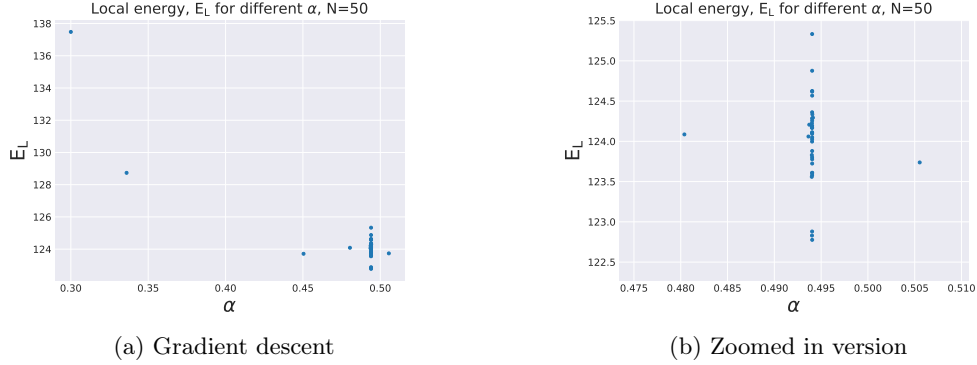
(a) Gradient descent

(b) Zoomed in version

Figure 8: Same as figure 7 but with $N = 50$.
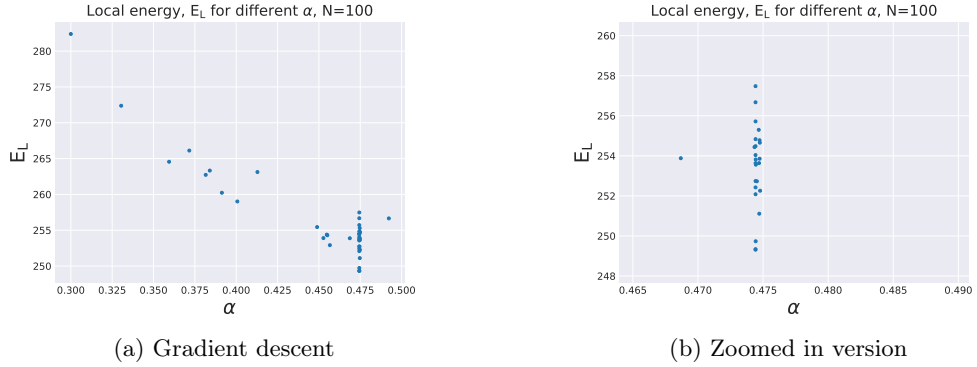

(a) Gradient descent

(b) Zoomed in version

Figure 9: Same as figure 7 but with $N = 100$.

The tables showing the benchmarking with importance sampling (table 4, 5 and 6) tell a similar story, though the importance sampling is generally somewhat slower (at least for large enough N).

We also note that the CPU time used by our algorithm suddenly soars when using $N = 500$. This is most likely due to the fact that the overhead due to the parallelization at this point becomes insignificant, relative to the time used in the code, so that the code scales strongly with N.

5.3 The energy as a function of α

We show how the energy varies as a function of α with brute-force sampling in figure 3 and for importance sampling in figure 4. We note that we manage to hit a minimum at the expected point with both methods, and that this minimum has the correct value in both cases. This suggests that our method is working as expected.

Comparing this with the interacting case, shown for 10 particles in figure 5, we see that we get a similar shape - a minimum near $\alpha = 0.5$, with a steady increase in energy beyond that point. We also note that the error is relatively small. Comparing figure the noninteracting case, we see that importance sampling does not significantly improve the error bound, suggesting that we use sufficiently many steps to attain a stable variance.

We therefore decide to implement brute force sampling in the rest of this project, as it is sig-
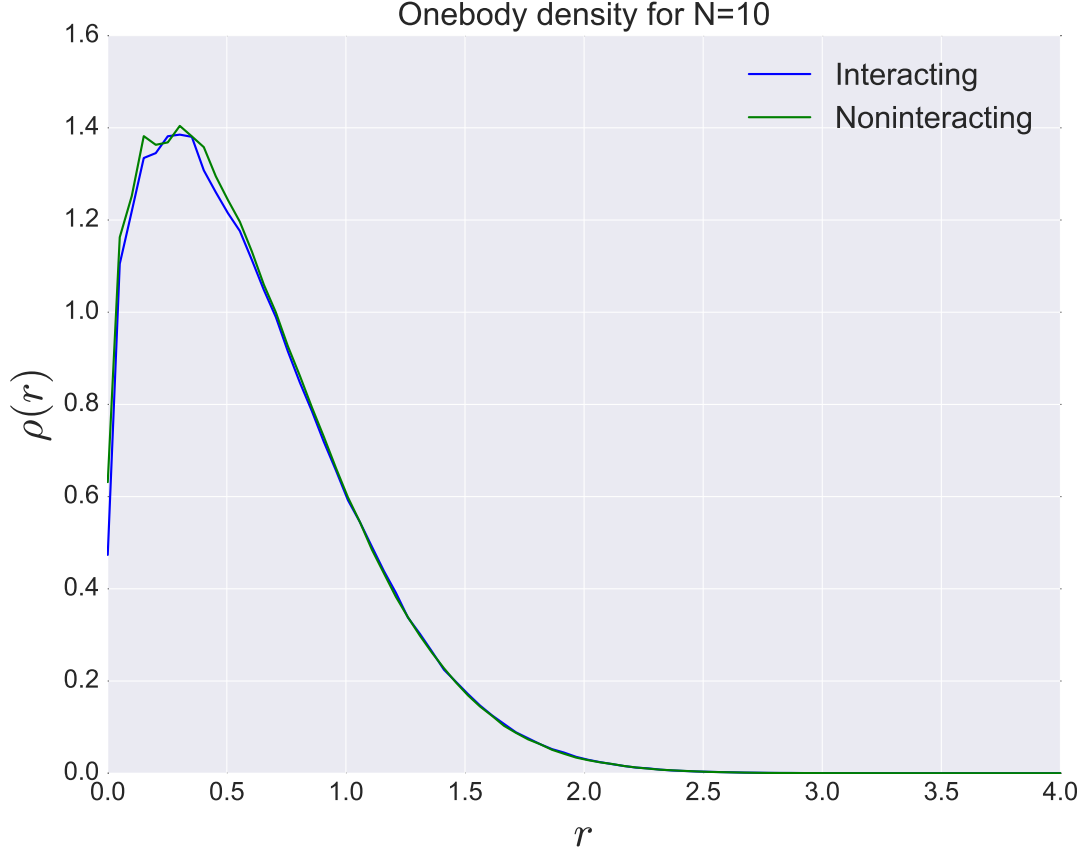
Figure 10: Radial onebody density for noninteracting ($a = 0$) and interacting ($a = 0.0043$), using $\omega_z = \omega = \beta = 1$, $N = 10$ particles in 3 dimensions, using brute force sampling.

nificantly faster, and does not appear to come at a large cost to accuracy. Whilst it would be possible to use importance sampling with fewer steps, we were unable to get satisfactory results using this method. We therefore use exclusively brute-force sampling for the interacting case.

5.4 Results from the fully interacting system

We show how the gradient descent evolves for $N = 10$ in the fully interacting system in figure 7. After 40 iterations, the gradient was less than 10^{-5}, the convergence was fairly stable, and the optimal α did not change much. We therefore adopt $\alpha = 0.499037$, as stated in table 7 as an acceptable value for α.

A similar analysis is done for $N = 50$ in figure 8. Also in this case, the gradient descent method gave a small gradient after 40 iterations (on the order of 10^{-5}), and the convergence was relatively stable. Furthermore, the optimal α did not change much. We therefore adapt the value $\alpha = 0.494033$ quoted in table 7 as our estimate for α.

The convergence for $N = 100$ was very weak. The gradient stayed above 10^{-3} throughout the 40 iterations, and it varied quite wildly between successive iterations. Furthermore, it was highly sensitive to the starting conditions. Starting with an initial guess of $\alpha = 0.3$ gives the $\alpha = 0.474419$ shown in table 7. However, starting with $\alpha = 0.49$, gives (after 40 iterations), $\alpha = 0.493261$.

Running with this α gives an energy of $E_L = 253.0 \pm 0.1\ \hbar\omega$. Similarly, running with the same α as for $N = 10$ (i.e. $\alpha = 0.499037$) gives a local energy of $E_L = 253.12 \pm 0.06\ \hbar\omega$, which thus is slightly lower than for the optimal α found by the gradient descent method. This suggests that the energy is relatively flat close to the minimum. We therefore choose $\alpha = 0.493261$ as our correct value of α, as this has the lowest energy.

5.5 The onebody density

6 Conclusion

6.1 Conclusion

6.2 Outlook

References

[Barzilai and Borwein, 1988] Barzilai, J. and Borwein, J. M. (1988). Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8(1):141–148.

[DuBois and Glyde, 2001] DuBois, J. L. and Glyde, H. R. (2001). Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis. *Physical Review A*, 63(2):023602.

[Griffiths, 2004] Griffiths, D. (2004). *Introduction to Quantum Mechanics*. 2 edition.

[Hjorth-Jensen, 2015] Hjorth-Jensen, M. (2015). *Computational Physics*. Oslo.

[Jonsson, 2018] Jonsson, M. (2018). *LECTURE NOTES FOR BLOCKING METHOD*.

[Nilsen et al., 2005] Nilsen, J. K., Mur-Petit, J., Guilleumas, M., Hjorth-Jensen, M., and Polls, A. (2005). Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region. *Physical Review A*, 71(5):053610.

Appendices

A Analytic expression for the local energy and quantum force

First derivative: For particle k, ∇_k is given by:

$$\nabla_k \Psi_T = \nabla_k \left(\Psi_T(\mathbf{r}) = \nabla_k \prod_{i=1}^{n} \phi(\mathbf{r}_i) \exp\left(\sum_{i<j} u(r_{ij}) \right) \right) \tag{36}$$

Now apply the product rule. For the first term, all terms are unchanged except where $i = k$, giving the first term as:

$$\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right) \tag{37}$$

The second term is trickier. We can rewrite it as follows:

$$\prod_i \phi(\boldsymbol{r}_i) \nabla_k \left[\exp\left(\sum_{i<j} u(r_{ij}) \right) \right] = \prod_i \phi(\boldsymbol{r}_i) \nabla_k \left[\exp\left(\sum_{j=1}^{n} \sum_{i=1}^{j-1} u(r_{ij}) \right) \right] \tag{38}$$

18

This is calculated by using the chain-rule. To evaluate the innerfunction, we must look at $1 \leq j \leq k$, $j = k$ and $j \geq k$. Begining with $1 \leq j \leq k$, the first sum is from 1 up the kth particle, meaning all terms differentiated before k vanishes and we are left with $u(r_{ik})$, thus leaving with us with:

$$\prod_i \phi(\boldsymbol{r}_i)\nabla_k \left[\exp\left(\sum_{j=1}^{k}\sum_{i=1}^{k-1} u(r_{ij})\right)\right] = \prod_i \phi(\boldsymbol{r}_i)\exp\left\{\left(\sum_{i<j} u(r_{ij})\right)\right\}\left[\sum_{i=1}^{k-1} \nabla_k u(r_{ik})\right] \quad (39)$$

For $j = k$, we see that the first sum goes from and end up at kth particle, meaning this sum vanishes, secondly, the second sum is evaluated at $1 \leq i \leq k-1$, leaving with constants up to $k-1$, thus this term becomes 0.

$$\prod_i \phi(\boldsymbol{r}_i)\nabla_k \left[\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{j-1} u(r_{ij})\right)\right] = 0 \quad (40)$$

For $j > k$, the first sum goes from $k+1 \leq j \leq n$ and the second $1 \leq i \leq k$. The second sum vanishes, because all constants are differentiated away except the last term $i = k$. This leaves us with:

$$\prod_i \phi(\boldsymbol{r}_i)\nabla_k \left[\exp\left(\sum_{j=k+1}^{n}\sum_{i=1}^{k} u(r_{ij})\right)\right] = \prod_i \phi(\boldsymbol{r}_i)\exp\left\{\left(\sum_{i<j} u(r_{ij})\right)\right\}\left[\sum_{j=k+1}^{n} \nabla_k u(r_{kj})\right] \quad (41)$$

We can now write the second term as 38):

$$\prod_i \phi(\boldsymbol{r}_i)\exp\left\{\left(\sum_{i<j} u(r_{ij})\right)\right\}\left[\sum_{j=k+1}^{n} \nabla_k u(r_{kj}) + \left[\sum_{i=1}^{k-1} \nabla_k u(r_{ik})\right]\right] \quad (42)$$

This can be rewritten to:

$$\prod_i \phi(\boldsymbol{r}_i)\exp\left\{\left(\sum_{i<j} u(r_{ij})\right)\right\}\left[\sum_{j\neq k} \nabla_k u(r_{kj})\right] \quad (43)$$

Thus the first derivative of the trial wavefunction can be written as:

$$\nabla_k \Psi_T = \nabla_k \phi(\mathbf{r}_k)\left[\prod_{i\neq k} \phi(\mathbf{r}_i)\right]\exp\left(\sum_{i<j} u(r_{ij})\right) + \prod_i \phi(\boldsymbol{r}_i)\exp\left\{\left(\sum_{i<j} u(r_{ij})\right)\right\}\left[\sum_{j\neq k} u(r_{kj})\right] \quad (44)$$

Second derivative: Lastly we will calculate the second derivative of the trial function, where we use the product rule and the chain rule. We will then obtain the following:

$$\nabla_k^2 \Psi_T = \underbrace{\nabla_k^2 \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left(\sum_{i<j} u(r_{ij}) \right)}_{\text{I}} +$$

$$\underbrace{2 \left(\nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp\left\{ \left(\sum_{i<j} u(r_{ij}) \right) \right\} \left[\sum_{j \neq k} \nabla_k u(r_{kj}) \right] \right)}_{\text{II}} +$$

$$\underbrace{\prod_i \phi(\mathbf{r}_i) \exp\left\{ \left(\sum_{i<j} u(r_{ij}) \right) \right\} \left[\sum_{j \neq k} \nabla_k u(r_{kj}) \right]^2}_{\text{III}} +$$

$$\underbrace{\prod_i \phi(\mathbf{r}_i) \exp\left\{ \left(\sum_{i<j} u(r_{ij}) \right) \right\} \left[\sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) \right]}_{\text{IV}}$$

Where the derivatives of the different parts are obtained from previous. We will now divide the second derivative by the trial wavefunction, and obtain:

$$\frac{1}{\Psi_T} \nabla_k^2 \Psi_T = \underbrace{\frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_{\phi(\mathbf{r}_k)})}}_{\text{I}} + \underbrace{2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \nabla_k u(r_{kj})}_{\text{II}} + \underbrace{\left(\sum_{j \neq k} \nabla_k u(r_{kj}) \right)^2}_{\text{III}} + \underbrace{\left[\sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) \right]}_{\text{IV}} \quad (45)$$

We will now carry out the differentiation in term (II), (III) and (IV). Begining with the (II), by using the chain rule we can rewrite (II) as:

$$2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \sum_{j \neq k} \frac{\partial u(r_{kj})}{\partial r_{kj}} \frac{\partial r_{kj}}{\partial \mathbf{r}_k} = 2 \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left(\sum_{j \neq k} \frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} u'(r_{kj}) \right)$$

Where we have used $\nabla_k r_{kj} = (\mathbf{r}_k - \mathbf{r}_j)/r_{kj}$. Looking at (III), we can write out the square into two factors. The second paranthese the summation index is replaced by a dummy index i. Thus will look as:

$$\left(\sum_{j \neq k} \nabla_k u(r_{kj}) \right)^2 = \left(\sum_{j \neq k} \nabla_k u(r_{kj}) \right) \left(\sum_{i \neq k} \nabla_k u(r_{ki}) \right) \quad (46)$$

From (II) we found the derivative of this function. Thus this can be expressed as (combining also the double sum):

$$\left(\sum_{j \neq k} \nabla_k u(r_{kj}) \right)^2 = \sum_{i,j \neq k} \left(\frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \right) \left(\frac{\mathbf{r}_k - \mathbf{r}_i}{r_{ki}} \right) u'(r_{kj}) u'(r_{ki}) \quad (47)$$

The last part (IV), we must use the chain rule and product rule together. We will obtain the following term:

$$\sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) = \sum_{j \neq k} \nabla_k \left(\left(\frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \right)^2 u''(r_{kj}) + u'(r_{kj}) \nabla_k \left(\frac{\mathbf{r}_k - \mathbf{r}_j}{r_{kj}} \right) \right) \quad (48)$$

20

Note that this is a unit vector squared:

$$\left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}\right)^2 = 1 \tag{49}$$

and the last part is found by using the qoutient rule. This is the divergence to the vectors, since we are evaluating for specific particle k, and gradient to scalar (we must look for all combination of k particle):

$$\nabla_k \left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}\right) = \frac{r_{kj} \left(\nabla_k \cdot \boldsymbol{r}_k - \nabla_k \cdot \boldsymbol{r}_j\right) - (\boldsymbol{r}_k - \boldsymbol{r}_j)\nabla_k r_{kj}}{r_{kj}^2} \tag{50}$$

This simply becomes:

$$\nabla_k \left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}\right) = \frac{3r_{kj}^2 - \boldsymbol{r}_k^2 + 2(\boldsymbol{r}_k \cdot \boldsymbol{r}_j) - \boldsymbol{r}_j^2}{r_{kj}^3} \tag{51}$$

Note that $r_{kj} - \boldsymbol{r}_k^2 + 2(\boldsymbol{r}_k \cdot \boldsymbol{r}_j) - \boldsymbol{r}_j^2 = 0$, this leaves us with:

$$\nabla_k \left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}\right) = \frac{2}{r_{kj}} \tag{52}$$

Thus expression (IV) can be expressed as:

$$\sum_{j \neq k} \nabla_k \nabla_k u(r_{kj}) = \sum_{j \neq k} \nabla_k \left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right) \tag{53}$$

Combining (I), (II), (III) and (IV), we we will get that the second derivative of the trial function is:

$$\frac{1}{\Psi_T}\nabla_k^2\Psi_T = \frac{\nabla_k^2\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2\frac{\nabla_k\phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)}\left(\sum_{j \neq k}\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}u'(r_{kj})\right) +$$
$$\sum_{i,j \neq k}\left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}\right)\left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_i}{r_{ki}}\right)u'(r_{kj})u'(r_{ki}) + \sum_{j \neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right) \tag{54}$$

Local energy: Now that we have the second derivative of the trial-function, we can write out the analytical expression for the local energy. Recall that:

$$E_L(\boldsymbol{r}) = \frac{1}{\Psi_T}H\Psi_T = \frac{1}{\Psi_T}\left[\sum_i^N\left(-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\boldsymbol{r}_i)\right) + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j)\right]\psi_T \tag{55}$$

The second derivative written explicitly is:

$$\frac{1}{\Psi_T}\nabla_k^2\Psi_T = 4\alpha\boldsymbol{Q}_k^2 - 4\alpha\boldsymbol{Q}_k\sum_{j \neq k}\left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}}u'(r_{kj})\right) +$$
$$\sum_{i,j \neq k}\frac{\boldsymbol{r}_k \cdot \boldsymbol{r}_k - \boldsymbol{r}_k \cdot \boldsymbol{r}_j - \boldsymbol{r}_k \cdot \boldsymbol{r}_i + \boldsymbol{r}_i \cdot \boldsymbol{r}_j}{r_{kj}r_{ki}}u'(r_{ki})u'(r_{kj}) \tag{56}$$
$$+ \sum_{j \neq k}\left(u''(r_{kj}) + \frac{2}{r_{kj}}u'(r_{kj})\right)$$

Where $\boldsymbol{Q}_k = \left(x_k\hat{i} + y_k\hat{j} + \beta z_k\hat{k}\right)$, and the first and second derivative of $u(r_{kj})$ is:

$$u'(r_{kj}) = \begin{cases} \infty & , r_{kj} \leq a \\ \frac{a}{r_{kj}^2 - ar_{kj}} & , r_{kj} > a \end{cases} \tag{57}$$

and

$$u''(r_{kj}) = \begin{cases} \infty & , r_{kj} \leq a \\ \dfrac{a^2 - 2ar_{kj}}{\left(r_{kj}^2 - ar_{kj}\right)^2} & , r_{kj} > a \end{cases} \tag{58}$$

Thus the analytical local energy for a group of particle is:

$$\begin{aligned} E_L = -\frac{\hbar^2}{2m} \sum_k \Bigg[& 4\alpha^2 \boldsymbol{Q}_k^2 - 4\alpha \boldsymbol{Q}_k + \sum_{j \neq k} \left(\frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}} u'(r_{kj}) \right) \\ & + \sum_{i,j \neq k} \left(\frac{\boldsymbol{r} \cdot \boldsymbol{r} + \boldsymbol{r}_j \cdot \boldsymbol{r}_i - \boldsymbol{r}_j \cdot \boldsymbol{r}_k - \boldsymbol{r}_k \cdot \boldsymbol{r}_i}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) \right) \\ & + \sum_{j \neq k} \left(u''(r_{kj} + \frac{2}{r_{kj}} u'(r_{kj})) \right) \Bigg] + \sum_k V_{ext}(\boldsymbol{r}_k) + \sum_{i<j} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \end{aligned} \tag{59}$$

This expression includes the situation where particles interact with each other. Now if we assume the particles does not interact with each other, we then have:

$$E_L = \sum_k \left(4\alpha^2 \boldsymbol{Q}_k + V_{ext}(\boldsymbol{r}_k) \right) \tag{60}$$

Quantum force/Drift force: The quantum/drift force is defined to be:

$$\boldsymbol{F}_i = 2 \frac{\nabla \Psi_T}{\Psi_T} \tag{61}$$

Using the first derivative we just derived above this expression will look like:

$$\boldsymbol{F}_i = 2 \frac{\nabla_k \phi(\boldsymbol{r}_k)}{\phi(\boldsymbol{r}_k)} + 2 \sum_{j \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}} u'(r_{kj}) = -4\alpha \boldsymbol{Q}_i + 2 \sum_{j \neq k} \frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{r_{kj}} u'(r_{kj}) \tag{62}$$

Where $\boldsymbol{Q}_i = x_i \hat{i} + y_i \hat{j} + \beta z_i \hat{k}$ and $u'(r_{kj})$ is given in (57). This is for a interacting case, since the distance between two particle and the gradient kj'th particle appears. Further notice that the quantum force can be written as $3 \times N$ matrix, where N is number of particles in the system. For a non-interacting case, we have:

$$\boldsymbol{F}_i = -4\alpha \boldsymbol{Q}_i \tag{63}$$

B The dimensionless Hamiltonian

We aim to demonstrate that the Hamiltonian in equation 1 is equivalent to the Hamiltonian shown in equation 29 under the coordinate transformation:

$$\boldsymbol{r} \to \frac{\boldsymbol{r}}{a_{ho}} = \boldsymbol{r}\sqrt{\frac{m\omega}{\hbar}} \quad E \to \frac{E}{\hbar\omega} \tag{64}$$

Where $a_{ho} = (h/m\omega)^{1/2}$. To show this, note first that the gradient operator under this transformation transforms as:

$$\nabla \to \sqrt{\frac{\hbar}{m\omega}} \nabla \tag{65}$$

We now introduce this coordinate transformation into our Hamiltonian. This gives:

$$H \to H = \sum_{i=1}^N \left(-\frac{\hbar^2}{2m} \frac{\omega m}{\hbar} \nabla_i^2 + \frac{1}{2} m\omega^2 \frac{\hbar}{m\omega} (x_i^2 + y_i^2 + \omega_z^2 z_i^2) \right) + \sum_{i<j}^N V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{66}$$

Which can be rewritten as:

$$H = \frac{\hbar\omega}{2} \sum_{i=1}^{N} \left(-\nabla^2 + \frac{\hbar\omega}{2} \left(x_i^2 + y_i^2 + \frac{\omega_z^2}{\omega^2} z_i^2 \right) \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{67}$$

Relabeling the energy and defining $\gamma = \omega_z/\omega$ gives:

$$H = \frac{1}{2} \sum_{i=1}^{N} \left(-\nabla_i^2 + x_i^2 + y_i^2 + \gamma z_i^2 \right) + \sum_{i<j}^{N} V_{int}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{68}$$

Which is exactly equation 29.

C   Deriving an approximate expression for the onebody density

We require that the spatial integral over the onebody density, $\rho(\boldsymbol{r_1})$ equals the total number of particles N. As we are only interested in the radial component of the onebody density, we consider only $\rho(r)$. This requirement can then be written as:

$$N = \int d^3 \boldsymbol{r} \rho(r) \tag{69}$$

Which gives, in spherical coordinates:

$$N = 4\pi \int_0^\infty dr \; r^2 \rho(r) \tag{70}$$

We now discretize this integral radius, choosing to integrate from 0 to $r_{\max}$, where we choose $r_{\max}$ larger than the largest distance any particle deviates from the origin. This interval is divided into M points, with a spacing of Δr. The discretized version of the integral then becomes:

$$N \approx 4\pi \sum_{i=1}^{M} r_i^2 \rho(r_i) \Delta r \tag{71}$$

From which it follows that the onebody density can be computed as:

$$\rho(r_i) \approx \frac{N_i}{4\pi r_i^2 \Delta r} \tag{72}$$

Where N_i is the number of particles in the spherical shell with radius r_i to $r_i + \Delta r$.

D   Finding the derivative of the local energy

The local energy, E_L is given by:

$$E_L = \langle \Psi_T | H | \Psi_T \rangle \tag{73}$$

We now want to find the derivative of the expectation value of the local energy $\langle E_L \rangle$ with respect to the variational parameter α. Before we do that we must normalize the wavefunction:

$$|\Psi_T\rangle = \frac{|\Psi_T\rangle}{\sqrt{\langle \Psi_T | \Psi_T \rangle}} \tag{74}$$

The expectation value of the local energy is then:

$$\langle E_L \rangle = \frac{\langle \Psi_T | \, E_L \, | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \tag{75}$$

Notice that can be written as:

$$\langle E_L \rangle = \frac{\langle \Psi_T | E_L | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} = \frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \Psi_T^* E_L \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{76}$$

Using the definition of E_L and we know that the wavefunction is real, we get that:

$$\langle E_L \rangle = \frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \Psi_T^* \left(\frac{1}{\Psi_T} H \Psi_T \right) \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N = \frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int H \Psi_T \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{77}$$

Thus this can be written as:

$$\langle E_L \rangle = \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} \tag{78}$$

By using the product rule and differentiating this with respect to α, we will get:

$$\frac{\partial \langle E_L \rangle}{\partial \alpha} = \frac{1}{\langle \Psi_T | \Psi_T \rangle} \frac{\partial \langle \Psi_T | H | \Psi_T \rangle}{\partial \alpha} + \langle \Psi_T | H | \Psi_T \rangle \frac{\partial}{\partial \alpha} \left(\frac{1}{\langle \Psi_T | \Psi_T \rangle} \right) \tag{79}$$

The first expression is:

$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \frac{\partial \langle \Psi_T | H | \Psi_T \rangle}{\partial \alpha} = \frac{1}{\langle \Psi_T | \Psi_T \rangle} \frac{\partial}{\partial \alpha} \int .. \int \Psi_T^* H \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{80}$$

Taking the derivative into the integral and apply the product rule, we will obtain:

$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \frac{\partial \Psi_T^*}{\partial \alpha} H \Psi_T + \Psi_T^* \frac{\partial H}{\partial \alpha} \Psi_T + \Psi_T^* H \frac{\partial \Psi_T}{\partial \alpha} \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{81}$$

Since the hamiltonian does not depend on α we have:

$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \underbrace{\frac{\partial \Psi_T^*}{\partial \alpha} H \Psi_T}_{I} + \underbrace{\Psi_T^* H \frac{\partial \Psi_T}{\partial \alpha}}_{II} \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{82}$$

We now wish to include the local energy in the integral, to do this we must do some algebra. Consider the first expression (I):

$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \frac{\partial \Psi_T^*}{\partial \alpha} H \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{83}$$

By dividing and multiplying with Ψ_T^* and Ψ_T on the left and right for the hamiltonian operator, we can write I as:

$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \frac{\Psi_T^*}{\Psi_T^*} \frac{\partial \Psi_T^*}{\partial \alpha} H \Psi_T \frac{\Psi_T}{\Psi_T} \, d\mathbf{r}_1 .. d\mathbf{r}_N =$$
$$\frac{1}{\langle \Psi_T | \Psi_T \rangle} \int .. \int \frac{|\Psi_T|^2}{\Psi_T} \frac{\partial \Psi_T^*}{\partial \alpha} E_L \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{84}$$
$$= \langle \frac{E_L}{\Psi_T} \frac{\partial \Psi_T}{\partial \alpha} \rangle$$

Expression (II) is more trickier to handle, since we know that the hamiltionian operator is hermitian and the wavefunction is real, we know that:

$$\langle a | H | b \rangle^* = \langle b | H | a \rangle \tag{85}$$

We will now write expression (I) and (II) in Dirac notation. Expression (I) will look like:

$$\frac{1}{\langle\Psi_T|\Psi_T\rangle} \int .. \int \frac{\Psi_T^*}{\Psi_T^*} \frac{\partial\Psi_T^*}{\partial\alpha} H\Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N = \langle\Psi_T| \frac{1}{\Psi_T} \frac{\partial\Psi_T}{\partial\alpha} H |\Psi_T\rangle \tag{86}$$

Expression (II) will look like:

$$\frac{1}{\langle\Psi_T|\Psi_T\rangle} \int .. \int \Psi_T^* H \frac{\partial\Psi_T}{\partial\alpha} \frac{\Psi_T}{\Psi_T} \, d\mathbf{r}_1 .. d\mathbf{r}_N = \langle\Psi_T| H \frac{\partial\Psi_T}{\partial\alpha} \frac{1}{\Psi_T} |\Psi_T\rangle \tag{87}$$

Notice that (86) and (87) are just the conjugate of each oher, thus:

$$\langle\Psi_T| H \frac{\partial\Psi_T}{\partial\alpha} \frac{1}{\Psi_T} |\Psi_T\rangle^* = \langle\Psi_T| \frac{1}{\Psi_T} \frac{\partial\Psi_T}{\partial\alpha} H |\Psi_T\rangle \tag{88}$$

Since (86) is equal to (84) then (86) must also equal the result from (84). Thus:

$$\frac{1}{\langle\Psi_T|\Psi_T\rangle} \int .. \int \frac{\partial\Psi_T^*}{\partial\alpha} H\Psi_T + \Psi_T^* H \frac{\partial\Psi_T}{\partial\alpha} \, d\mathbf{r}_1 .. d\mathbf{r}_N = 2\langle \frac{E_L}{\Psi_T} \frac{\partial\Psi_T}{\partial\alpha} \rangle \tag{89}$$

Now differentiating the second expression in (79):

$$\langle\Psi_T| H |\Psi_T\rangle \frac{\partial}{\partial\alpha} \left(\frac{1}{\langle\Psi_T|\Psi_T\rangle} \right) = -\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \frac{\partial\langle\Psi_T|\Psi_T\rangle}{\partial\alpha} \tag{90}$$

Writing the last expression as an integral:

$$-\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \frac{\partial\langle\Psi_T|\Psi_T\rangle}{\partial\alpha} = -\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \int .. \int \frac{\partial\Psi_T^*}{\partial\alpha} \Psi_T + \Psi_T^* \frac{\partial\Psi_T}{\partial\alpha} \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{91}$$

we clearly see that:

$$-\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \int .. \int \frac{\partial\Psi_T^*}{\partial\alpha} \Psi_T + \Psi_T^* \frac{\partial\Psi_T}{\partial\alpha} \, d\mathbf{r}_1 .. d\mathbf{r}_N = -2\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \int .. \int \frac{\partial\Psi_T^*}{\partial\alpha} \Psi_T \, d\mathbf{r}_1 .. d\mathbf{r}_N \tag{92}$$

We will now divide with $\frac{\Psi_T^*}{\Psi_T^*}$, use the definition of the expectation value of the local energy and derivative of the wavefunction with respect to α, This can also be written as:

$$-2\frac{\langle\Psi_T| H |\Psi_T\rangle}{\langle\Psi_T|\Psi_T\rangle^2} \int .. \int \Psi_T^* \Psi_T \left(\frac{1}{\Psi_T} \frac{\partial\Psi_T^*}{\partial\alpha} \right) d\mathbf{r}_1 .. d\mathbf{r}_N = -2\langle E_L\rangle\langle \frac{1}{\Psi_T} \frac{\partial\Psi}{\partial\alpha} \rangle \tag{93}$$

thus:

$$\langle\Psi_T| H |\Psi_T\rangle \frac{\partial}{\partial\alpha} \left(\frac{1}{\langle\Psi_T|\Psi_T\rangle} \right) = -2\langle E_L\rangle\langle \frac{1}{\Psi_T} \frac{\partial\Psi}{\partial\alpha} \rangle \tag{94}$$

Combining the first and second expression the derivative of the local energy can be expressed as:

$$\frac{\partial\langle E_L\rangle}{\partial\alpha} = 2 \left(\langle \frac{E_L}{\Psi_T} \frac{\partial\Psi_T}{\partial\alpha} \rangle - \langle E_L\rangle\langle \frac{1}{\Psi_T} \frac{\partial\Psi}{\partial\alpha} \rangle \right) \tag{95}$$