1 Introduction

Since it has been shown that Bose-Einstein condensation occur in gases of alkali atoms confined in magnetic traps, understanding the confined Bose systems has become a popular field. In this project we have tried to calculate the ground state energy of various number of trapped bosons using Variational Monte Carlo calculations with a specific trial wavefunction. Firstly using the spherical harmonic oscillator as a trap with no interaction between the particles, then swtiching to an elliptical trap. Where in both cases we calculate both the analytical and numerical solution. Also running the calculations with and without a repulsive potential due to the fact that bosons have a size.

2 Method

2.1 System

A two-body Hamiltonian of this system is on the form

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

where the external potential for the trap is for the spherical and elliptical part given by

$$V_{ext}(\boldsymbol{r}) = \begin{cases} \frac{1}{2} m\omega^2 r^2 & \text{Spherical} \\ \frac{1}{2} m[\omega^2(x^2+y^2) + \gamma^2 z^2] & \text{Elliptical} \end{cases} \tag{2}$$

where ω is the trap potential strength, and in the elliptical trap γ is the strength in z-direction. The internal potential which represents the repulsion when two boson gets close is

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

with a as the hard-core diamater of the bosons. Setting up the system with a trial wavefunction Ψ_T for the ground state of the form

$$\Psi_T(\mathbf{R}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N, \alpha, \beta) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i<j} f(a, |\mathbf{r}_i - \mathbf{r}_j|), \tag{4}$$

using α and β as variational parameters, and

$$g(\alpha, \beta, \boldsymbol{r}_i) = e^{-\alpha(x_i^2 + y_i^2 + \beta z_i^2)}$$

and

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \tag{5}$$

For the simplest cases where we set the boson size $a = 0$ and $\beta = 1$ the trial wavefunction becomes

$$\Psi_T(\boldsymbol{R}) = e^{-\alpha r^2} \tag{6}$$

Calculating the double derivative of the wavefunction returns

$$\nabla^2 \Psi_T = \nabla^2 e^{-\alpha r^2} = \nabla - \alpha 2 r e^{-\alpha r^2} = 2\alpha e^{-\alpha r^2}(2\alpha r^2 - 1) \tag{7}$$

then inserting this result into the expression for the local energy yields

$$E_L(\boldsymbol{R}) = \frac{1}{\Psi_T(\boldsymbol{R})} H\Psi_T(\boldsymbol{R}) = 2\alpha(2\alpha r^2 - 1) \tag{8}$$

which is the analytical experssion for the local energy in a spherical trap without interaction.

2.1.1 Interacting local energy

$$\begin{aligned}
\nabla_k^2 \Psi_T(\mathbf{R}) = \nabla_k^2 \phi_k \prod_{i \neq k} \phi_i \exp\left(\sum_{i<j}\right) \\
+ \nabla_k \phi_k \prod_{i \neq k} \phi_i \sum_{i \neq k} \nabla_k u_{ij} \exp\left(\sum_{i<j} u_i j\right) \\
+ \nabla_k \phi_k \prod_{i \neq k} \phi_i \exp\left(\sum_{i<j}\right) \sum_{i \neq k} \nabla_k u_{ij} \\
+ \prod_i \phi_i \sum_{i<j} u_{ij} \exp\left(\sum_{i<j} u_{ij}\right) \sum_{j \neq k} \nabla_k u_{ij} \\
+ \prod_i \phi_i \exp\left(\sum_{i<j} u_{ij}\right) \sum_{j \neq k} \nabla_k^2 u_{ij}
\end{aligned}$$

dividing by the wavefunction gives us

$$\frac{1}{\Psi_T(\mathbf{R})} \nabla_k^2 \Psi_T(\mathbf{R}) = \frac{\nabla_k^2 \phi_k}{\phi_k} + 2\frac{\nabla_k \phi_k}{\phi_k} \sum_{j \neq k} \nabla_k u_{ij} + \sum_{i \leq j} u_{ij} \sum_{j \neq k} \nabla_k u_{ij} + \sum_{j \neq k} \nabla_k^2 u_{ij}$$

2.2 Numerical method

2.2.1 Brute force Metropolis

For the first part we use a Variational Monte Carlo program with a brute force Metropolis sampling to calculate the ground state energy. The Metropolis algorithm compares

the probability of being in the old position $A_{j\to i}$ against the probability of being in a position in a chosen direction $A_{i\to j}$.

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i T_{i\to j}}{p_j T_{j\to i}} \tag{9}$$

where T is the transition probability and p is the probability distribution. Given that the transition probability is independent of direction we are left with the

$$\frac{A_{j\to i}}{A_{i\to j}} = \frac{p_i}{p_j} \tag{10}$$

The Metropolis choice is to maximize the A values so preferring

$$A_{j\to i} = \min\left(1, \frac{p_i}{p_j}\right) \tag{11}$$

This is done by first placing the system in a random, Gaussian distributed position around 0 with a $\sigma = 1/\sqrt{2}$. Evaluating the wavefunction according to equation 6. Then choosing a random particle and dimension, and moving it with a Gaussian distribution to a new position. Do the same evaluation for the new position and comparing

$$\frac{|\Psi_{New}|^2}{|\Psi_{Old}|^2}$$

Then by taking a random, uniformly distributed number between 0 and 1 and compare it to the ratio between the new and old wavefunction. If the ratio is larger we accept the new step, if not we revert back to the old position. Then lastly we sample the energy for the resulting system.

Listing 1: Brute force Metropolis

```
randomMove        = normalDistribution;
waveFunctionOld = my_waveFunction->evaluate();

my_particles[chosenParticle]->changePosition(chosenDimension, randomMove);

waveFunctionNew = my_waveFunction->evaluate();

waveFunctionsCompared = (waveFunctionNew*waveFunctionNew)/
                (waveFunctionOld*waveFunctionOld);

if (waveFunctionsCompared < uniformDistribution{
  my_particles[chosenParticle]->changePosition(chosenDimension, -randomMove);
  return false;
}
return true;
```

When using this method we get the result which is shown in table 1. This result is in exact correspondence to the analytical solution.

2.2.2 Importance sampling

To make the choice of sampling more relevant we use importance sampling. Here there is a biased direction in the new step which is dependent on the trial wavefunction. The new position is given as the solutions to Langevin equation,

$$y = x + DF(x)\Delta t + \xi\sqrt{\Delta t} \tag{12}$$

Here x is the old position, Δt the step unit, D is the diffusion coefficient, $F(x)$ quantum force and ξ is a normal distributed random variable. The quantum force is given by

$$\mathbf{F} = 2\frac{1}{\Psi_T}\nabla\Psi_T \tag{13}$$

which gives the walker an incentive to go towards areas where the wavefunction is large. The new comparison is now

$$\frac{G(x, y, \Delta t)|\Psi_T(y)|^2}{G(y, x, \Delta t)|\Psi_T(x)|^2} \tag{14}$$

where

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

2.2.3 Interacting

3 Results

3.0.4 Brute force Metropolis

N particles	$< E >$	Variance	Accepted	Time [s]
1	5.000000e-01	0.000000e+00	0.553895	0.018
10	5.000000e+00	0.000000e+00	0.548784	0.024
100	5.000000e+01	0.000000e+00	0.550417	0.075
500	2.500000e+02	0.000000e+00	0.550962	0.312

Table 1: Benchmark of the brute force Metropolis method with analytical calculation of the Laplacian. Using 10^5 cycles and 1.7 as step length

With the brute force Metropolis method and analytical calculation of the Laplacian the results are exactly right, as seen in table for 1 dimension, 10^5 cycles and step length of 1.7. This gives an accpetance rate of about 50%. With lower step length the acceptance rate would go up, but then the energies would be sampled at a more narrow range and we could not be sure if it is the acutal ground state. In table A.1.1 and A.1.2 the runs have been done with the same parameters but with 2 and 3 dimensions. The result is the same with only a slight increase in time for the highest number of particles.

N particles	$< E >$	Variance	Accepted	Time [s]
1	5.000000e-01	-9.436896e-16	0.550639	0.023
10	5.000000e+00	-5.115908e-13	0.548139	0.099
100	5.000000e+01	3.092282e-11	0.551551	2.868
500	2.500000e+02	3.419700e-10	0.550273	61.191

Table 2: Benchmark of the brute force Metropolis method with numerical calculation of the Laplacian. Using 10^5 cycles and a step length of 1.7.

Using numerical derivation to find the Laplacian slows the speed of the program by about 200 times for 500 particles. The precision is the same as with the analytical calculation and so is the acceptance rate. This tells us that using the numerical calculation is a bad choice when the analytical solution is available for the brute force Metropolis. The same results continuos when increasing to 2 and 3 dimensions, as shown in A.1.3 and A.1.4.

3.0.5 Importance sampling

N particles	$< E >$	Variance	Accepted	Time [s]
1	5.000000e-01	0.000000e+00	0.998522	0.03
10	5.000000e+00	0.000000e+00	0.998700	0.038
100	5.000000e+01	0.000000e+00	0.998644	0.111
500	2.500000e+02	0.000000e+00	0.998922	0.451

Table 3: Benchmark of importance sampling Metropolis algorithm with analytical calculated Laplacian. Using 10^5 cycles and a time step of 0.05.

Given the difference in how the step length is implemented in the two different methods it is hard to make any direct comparison of them. The energy and variance have the same values as is to be expected and the time difference is negligeble in the range we have tested here.

N particles	$< E >$	Variance	Accepted	Time [s]
1	4.999999e-01	1.337819e-14	0.998844	0.035
10	4.999999e+00	2.948752e-13	0.998800	0.111
100	4.999999e+01	-2.273737e-11	0.998711	2.907
500	2.500000e+02	1.455192e-10	0.998767	61.364

Table 4: Benchmark of importance sampling Metropolis algorithm with numerical calculation of the Laplacian. Using 10^5 cycles and a time step of 0.05.

The results for the Metropolis importance sampling algorithm has shown to be much the same as for the brute force method, with the obvious difference in acceptance ratio which is not comparable. Also here the CPU-time is almost identical.

3.0.6 Time comparison

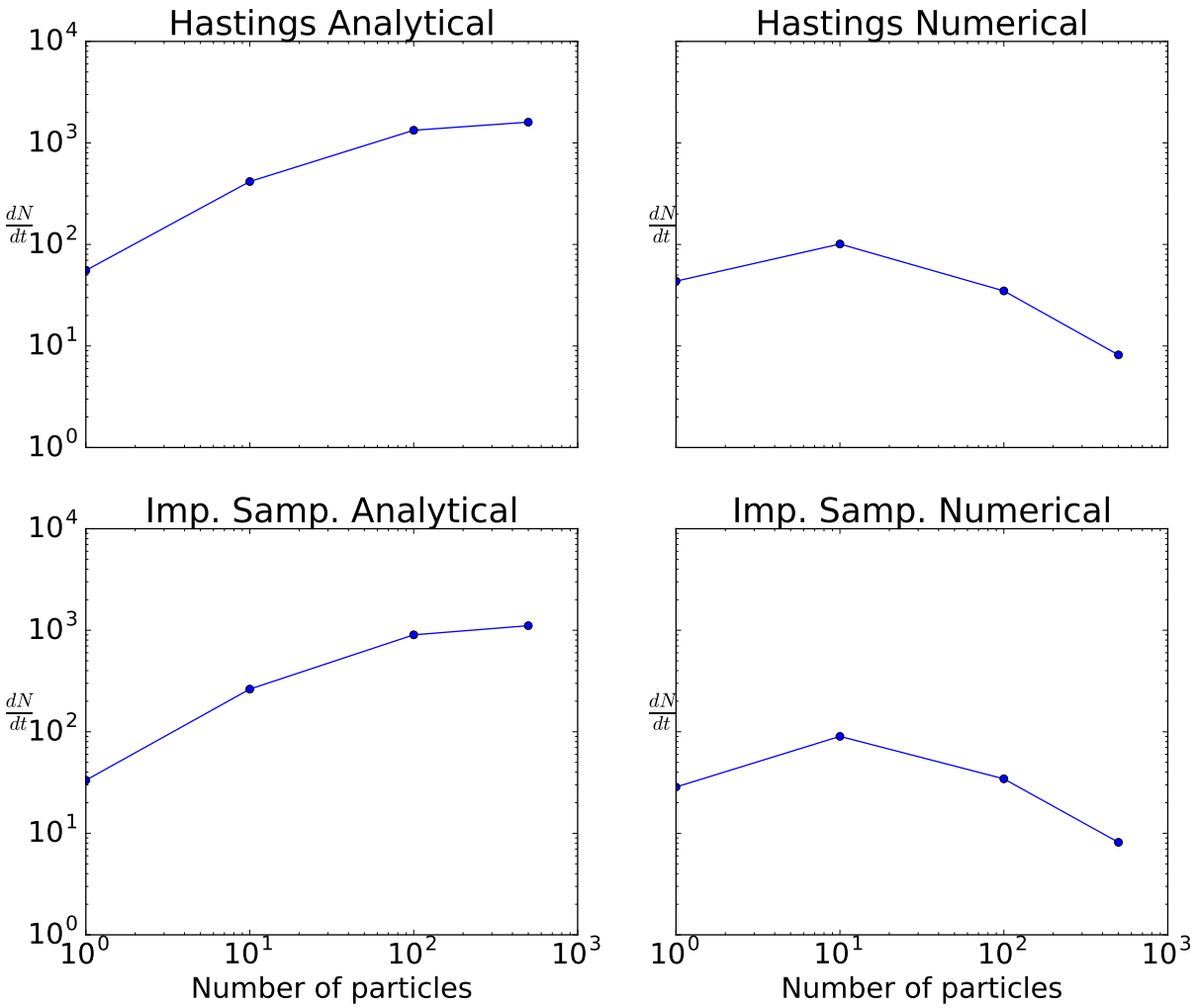

Figure 1: The x-axis is number of particles and y-axis are number of particles per time/step unit. The lower the graph goes, the more time per particle the program spends. This shows that the numerical Laplacians loses effectiveness as the number of particles grows, as for the analytical solutions the methods seem to converge toward a top.

For the analytical solutions the time spent on other operations than calculating the energies is high for lower number of particles, and as we try more particles the time spent on calculating the laplacian dominates and the graph seem to converge to a number around 1000 particles per unit step. With the numerical calculation of the Laplacian the effectiveness of the computation decreases steeply.

A Tables

A.1 Brute force Metropolis algorithm

All runs are with with 10^5 cycles and step length of 1.7.

A.1.1 Analytical 2D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.000000e+00	0.000000e+00	0.548395	0.019
10	1.000000e+01	0.000000e+00	0.549895	0.029
100	1.000000e+02	0.000000e+00	0.551184	0.121
500	5.000000e+02	0.000000e+00	0.551195	0.555

A.1.2 Analytical 3D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.500000e+00	0.000000e+00	0.550362	0.019
10	1.500000e+01	0.000000e+00	0.553095	0.034
100	1.500000e+02	0.000000e+00	0.549951	0.17
500	7.500000e+02	0.000000e+00	0.543150	0.8

A.1.3 Numerical 2D

N particles	$< E >$	Variance	Accepted	Time [s]
1	9.999999e-01	-1.576517e-14	0.549651	0.029
10	9.999999e+00	1.961098e-12	0.551617	0.208
100	9.999999e+01	1.509761e-10	0.552251	10.063
500	5.000000e+02	-5.675247e-09	0.551295	232.441

A.1.4 Numerical 3D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.500000e+00	6.394885e-14	0.553662	0.035
10	1.500000e+01	3.097966e-12	0.552217	0.364
100	1.500000e+02	-6.184564e-11	0.550728	21.808
500	7.499999e+02	-2.328306e-10	0.546217	517.259

A.2 Metropolis algorithm with Importance sampling

All runs are with with 10^5 cycles and step length of 0.05.

A.2.1 Analytical 2D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.000000e+00	0.000000e+00	0.998744	0.03
10	1.000000e+01	0.000000e+00	0.998644	0.044
100	1.000000e+02	0.000000e+00	0.998689	0.181
500	5.000000e+02	0.000000e+00	0.998800	0.817

A.2.2 Analytical 3D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.500000e+00	0.000000e+00	0.998756	0.031
10	1.500000e+01	0.000000e+00	0.998767	0.051
100	1.500000e+02	0.000000e+00	0.998967	0.254
500	7.500000e+02	0.000000e+00	0.998833	1.176

A.2.3 Numerical 2D

N particles	$< E >$	Variance	Accepted	Time [s]
1	9.999999e-01	1.099121e-14	0.998756	0.041
10	9.999999e+00	6.963319e-13	0.998767	0.223
100	9.999999e+01	2.546585e-11	0.998711	10.117
500	4.999999e+02	-5.587935e-09	0.998789	232.673

A.2.4 Numerical 3D

N particles	$< E >$	Variance	Accepted	Time [s]
1	1.500000e+00	1.332268e-15	0.998767	0.046
10	1.500000e+01	2.131628e-12	0.998833	0.38
100	1.500000e+02	-5.456968e-11	0.998611	21.95
500	7.499999e+02	1.012813e-08	0.998667	517.499