FYS4150 - Computational Physics
Project 5

Alex Ho

December 4, 2016

Abstract

Contents

1 Introduction

In project 2, we studied the behaviour of two electrons interacting in a three dimensional harmonic oscillator potential. In that project, we had a look at the so called *Eigenvalue problem* and used Jacobi's method to solve this problem. From that, we noted that Jacobi's method was not an optimal way to solve our quantum mechanical system, as it used a lot of CPU time to finish its calculations.

For this project, we will revisit this three dimensional harmonic oscillator with two electrons. This time, we will use the Monte Carlo method known as the Metropolis algorithm, which we also used in project 4, to solve this quantum mechanical system.

All relevant files used in this project can be found in the this GitHub page: `https://github.com/AHo94/FYS3150_Projects/tree/master/Project5`

2 Method

2.1 Analytical form for the trial wave functions and local energies

For this project, we will use natural units. That is $\hbar = c = e = m_e = 1$. We will consider two electrons in a quantum dot with a frequency $\hbar\omega = 1$. The Hamiltonian for these two electrons is

$$H_0 = -\frac{1}{2}(\nabla_1^2 + \nabla_2^2) + \frac{1}{2}\omega^2(r_1^2 + r_2^2)$$

The wave function for one electron in a harmonic oscillator potential is

$$\phi_{n_x,n_y,n_z}(x,y,z) = \frac{A}{2}H_{n_x}(\sqrt{\omega}x)H_{n_y}(\sqrt{\omega}y)H_{n_z}(\sqrt{\omega}z)e^{-\frac{\omega}{2}(x^2+y^2+z^2)}$$

Where H_{n_x} are Hermite polynomials and A is a normalization constant. n_x, n_y and n_z are some quantum numbers. When the Hamiltonian is acted

on this wave function, we obtain the energy. That is

$$H_0\phi_{n_x,n_y,n_z} = \epsilon_{n_x,n_y,n_z}\phi_{n_x,n_y,n_z}$$

with the energy given as

$$\epsilon_{n_x,n_y,n_z} = \omega\left(n_x + n_y + n_z + \frac{3}{2}\right)$$

For the ground state, $n_x = n_y = n_z = 0$, the energy of the single electron is

$$\epsilon_{0,0,0} = \frac{3}{2}\omega$$

Let us now consider two electrons, so we have $\phi^1_{n_x,n_y,n_z}$ and $\phi^2_{n_x,n_y,n_z}$ (not squared!). Acting the Hamiltonian on both these wave functions gives

$$H_0(\phi^1_{n_x,n_y,n_z} + \phi^2_{n_x,n_y,n_z}) = \epsilon^1_{n_x,n_y,n_z}\phi^1_{n_x,n_y,n_z} + \epsilon^2_{n_x,n_y,n_z}\phi^2_{n_x,n_y,n_z}$$

The total energy for these two electrons, when we consider the ground state, is then

$$\begin{aligned}\epsilon^{\text{tot}}_{0,0,0} &= \epsilon^1_{0,0,0} + \epsilon^2_{0,0,0} \\ &= \omega\left(\frac{3}{2}\right) + \omega\left(\frac{3}{2}\right) \\ &= 3\omega\end{aligned}$$

We will now consider these two trial wave functions given as

$$\psi_{T_1} = C\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right) \tag{1}$$

$$\psi_{T_2} = C\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right)\exp\left(\frac{r_{12}}{2(1+\beta r_{12})}\right) \tag{2}$$

where $r_{12} = \sqrt{r_1 - r_2}$ and α and β are variational parameters. The second exponential in the ψ_{T_2} is known as the Jastrow factor, which is there because it gives the lowest possible energy for the system, while reducing the amount of variational parameters.

Let us find the energy of the first trial function. First, we should rewrite the Hamiltonian in spherical coordinates. Our system does not depend on the radial coordinates, so the ∇ operator, for electron i in spherical coordinates, becomes

$$\nabla_i^2 = -\frac{1}{2}\frac{d^2}{dr_i^2} - \frac{1}{r_i}\frac{d}{dr_i}$$

By acting the Hamiltonian on ψ_{T_1}, we get

$$H_0\psi_{T_1} = \left(-\frac{d^2}{dr_1^2} - \frac{1}{r_1}\frac{d}{dr_1} - \frac{d^2}{dr_2^2} - \frac{1}{r_2}\frac{d}{dr_2} + \frac{1}{2}\omega(r_1^2 + r_2^2)\right)\psi_{T_1}$$

There will be a lot of derivatives to keep track of here, so I will take this step by step. Let us first differentiate with respect to r_1 first. The first derivative is

$$\begin{aligned}\frac{d}{dr_1}\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right) &= (-\alpha\omega r_1)\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right) \\ &= -\alpha\omega r_1\psi_{T_1}\end{aligned}$$

The second derivative then becomes

$$\begin{aligned}\frac{d^2}{dr_1^2}\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right) &= \frac{d}{dr_1}\left[-\alpha\omega r_1\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right)\right] \\ &= (-\alpha\omega + \alpha^2\omega^2 r_1^2)\exp\left(-\frac{\alpha\omega}{2}(r_1^2 + r_2^2)\right) \\ &= (\alpha^2\omega^2 r_1^2 - \alpha\omega)\psi_{T_1}\end{aligned}$$

In short, the kinetic term is

$$\nabla_1^2\psi_{T_1} = -\frac{1}{2}(\alpha^2\omega^2 r_1^2 - 3\alpha\omega)\psi_{T_1} \tag{3}$$

I will skip the calculation for the derivative with respect to r_2, but the results are

$$\begin{aligned}\frac{d}{dr_2}\psi_{T_1} &= -\alpha\omega r_2\psi_{T_1} \\ \frac{d^2}{dr_2^2}\psi_{T_1} &= (\alpha^2\omega^2 r_2^2 - \alpha\omega)\psi_{T_1}\end{aligned}$$

which gives

$$\nabla_1^2\psi_{T_1} = -\frac{1}{2}(\alpha^2\omega^2 r_2^2 - 3\alpha\omega)\psi_{T_1}$$

The local energy, for this trial function, is then

$$\begin{aligned} E_{L_1} &= -\frac{1}{2}(\alpha^2\omega^2 r_1^2 - 3\alpha\omega) - \frac{1}{2}(\alpha^2\omega^2 r_2^2 - 3\alpha\omega) + \frac{1}{2}\omega^2(r_1^2 + r_2^2) \\ &= 3\alpha\omega - \frac{1}{2}\alpha^2\omega^2(r_1^2 + r_2^2) + \frac{1}{2}\omega(r_1^2 + r_2^2) \\ &= 3\alpha\omega + \frac{1}{2}(\omega^2 - \alpha^2\omega^2)(r_1^2 + r_2^2) \\ &= 3\alpha\omega + \frac{1}{2}\omega^2(1 - \alpha^2)(r_1^2 + r_2^2) \end{aligned} \tag{4}$$

which is exactly what we wanted to show. We will skip the derivation of the analytical local energy for the second trial wave function. If we add Coulomb interaction, the Hamiltonian now becomes

$$H_0 = \sum_{i=1}^{N}\left(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}\omega^2 r_i^2\right) + \sum_{i<j}\frac{1}{r_{ij}}$$

and the local energy for the first trial wave function is then

$$E_{L1} = 3\alpha\omega + \frac{1}{2}\omega^2(1 - \alpha^2)(r_1^2 + r_2^2) + \frac{1}{r_{12}} \tag{5}$$

where $r_{12} = |\mathbf{r}_1 - \mathbf{r}_2|$. The analytical expression for the second trial wave function, with Coulomb interaction, is then

$$E_{L2} = E_{L1} + \frac{1}{2(1+\beta r_{12})^2}\left[\alpha\omega r_{12} - \frac{1}{2(1+\beta r_{12})} - \frac{2}{r_{12}} + \frac{2\beta}{1+\beta r_{12}}\right] \tag{6}$$

with E_{L1} given in equation 5.

2.2 The Metropolis Algorithm

We will once again, like in project 4, use the Metropolis algorithm to solve our quantum mechanical system. The algorithm is as follows: for every Monte Carlo cycle we

- **1)** Start the electrons at an arbitrary position. We let the electron start at the position $x, y, z \in [-1, 1]$, which we get from a normal distribution.

- **2)** Give the electrons a new position determined by $\mathbf{R}' = \mathbf{R} + \delta \times s$, where $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N)$, s is a random number generated from a normal distribution such that $s \in [-1, 1]$ and δ is a small step length. Note that $\delta \times r$ is different for every component, e.g

$$\mathbf{r_1}' = (x_1 + \delta \times s_1, y_1 + \delta \times s_2, z_1 + \delta \times s_3)$$

  For this project we will consider $N = 2$ electrons.

- **3)** Calculate $w = P(\mathbf{R}')/P(\mathbf{R}) = |\Psi(\mathbf{R}')|^2/|\Psi(\mathbf{R})|^2$, where Ψ is the wave function we will consider (i.e the trial wave functions ψ_{T1} and ψ_{T2} mentioned previously). Pick a random number $p \in [0, 1]$, also given from a normal distribution function. We now have to consider these two cases

- **Case 1**: If $p <= w$, we accept this new position change and use the calculated $\mathbf{R}'$ to calculate the local energy. The system has now reached a lower energy state.

- **Case 2**: If $p > w$, we do not accept this new position and use the old position $\mathbf{R}$ to calculate the local energy.

- **4)** Use the new or old position (depending on whatever case above hits) to calculate the local energy $E_L = \frac{1}{\Psi} H_0 \Psi$.

By calculating the fraction $|\Psi(\mathbf{R}')|^2/|\Psi(\mathbf{R})|^2$, we eliminate the requirement to compute the integral given in

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

which is similar to what we did in project 4. In project 4, by calculating $p <= e^{-\beta \delta E}$, we could skip the calculation of the partition function Z.

Once again, by running this for N_{mc} Monte Carlo cycles, we find the energy

expectation value as

$$\langle E \rangle = \frac{1}{N_{mc}} \sum_i E_i$$

where E_i is all the energy samples calculated from the Metropolis algorithm. One can also find the expectation value of the mean distance at the energy minimum $r_{12} = |\mathbf{r}_1 - \mathbf{r}_2|$ in the same way.

2.3 Algorithm that determines the step length

We would like to find an algorithm that, for a given value of α, uses an optimal value of the step length δ. This should ideally result to roughly 50% accepted moves in the metropolis algorithm. One can run the program multiple times to determine the step length. However, one can also find an analytical solution, which would automatize the optimal step length.

Since we are looking at accepted moves, we will take a look at the equation

$$p = \frac{|\Psi(\mathbf{R}')|^2}{|\Psi(\mathbf{R})|^2}$$

which was used to determine whether we accept a move or not in the Metropolis algorithm. The length $|\mathbf{r}_1'| = \mathbf{r}_1 + \delta \times s = r_1'$ is now

$$r_1' = \sqrt{(x_1 + \delta s_1)^2 + (y_1 + \delta s_2)^2 + (z_1 + \delta s_3)^2}$$

We let $p = 0.5$, as we want roughly 50% accepted moves. We then plug in the first trial wave function to determine an optimal δ for a given α, which gives

$$\begin{aligned}
0.5 &= \frac{1}{\exp\left[-\alpha\omega(r_1^2 + r_2^2)\right]} \big(\exp\big[-\alpha\omega\big((x_1 + \delta s_1)^2 + (y_1 + \delta s_2)^2 \\
&+ (z_1 + \delta s_3)^2 + (x_2 + \delta s_4)^2 + (y_2 + \delta s_5)^2 + (z_2 + \delta s_6)^2\big)\big]\big) \\
&= \frac{1}{\exp[-\alpha\omega(r_1^2 + r_2^2)]} \big(\exp[-\alpha\omega\big(x_1^2 + y_1^2 + z_1^2 + 2\delta(x_1 s_1 + y_1 s_2 + z_1 s_3) + \\
&x_2^2 + y_2^2 + z_2^2 + 2\delta(x_2 s_4 + y_2 s_5 + z_2 s_6) + \delta^2(s_1^2 + s_2^2 + s_3^2 + s_4^2 + s_5^2 + s_6^2)\big)]\big) \\
&= \exp\big[-\alpha\omega\big(2\delta(x_1 s_1 + y_1 s_2 + z_1 s_3 + x_2 s_4 + y_2 s_5 + z_2 s_6) \\
&+ \delta^2(s_1^2 + s_2^2 + s_3^2 + s_4^2 + s_5^2 + s_6^2)\big)\big]
\end{aligned}$$

where we have in the last line used $x^2 + y^2 + z^2 = r^2$ for both r_1 and r_2, thus cancelling the denominator. Taking the logarithm on both sides gives

$$\begin{aligned}\ln(0.5) = -\alpha\omega\big(&2\delta(x_1s_1 + y_1s_2 + z_1s_3 + x_2s_4 + y_2s_5 + z_2s_6) \\ &+ \delta^2(s_1^2 + s_2^2 + s_3^2 + s_4^2 + s_5^2 + s_6^2)\big) \\ \implies &a\delta^2 + b\delta + c = 0\end{aligned}$$

where we have defined

$$\begin{aligned}a &= s_1^2 + s_2^2 + s_3^2 + s_4^2 + s_5^2 + s_6^2 \\ b &= 2(x_1s_1 + y_1s_2 + z_1s_3 + x_2s_4 + y_2s_5 + z_2s_6) \\ c &= \frac{\ln(0.5)}{\alpha\omega}\end{aligned}$$

This is just a second order equation for δ. We require that δ is always positive, so when we solve this equation for δ, we will have to only consider the positive part. Doing this results to

$$\delta = \frac{1}{2a}(-b + \sqrt{b^2 - 4ac}) \tag{7}$$

Implementing this in the program *should* result to roughly 50% of the configurations being accepted. However, we will see that it is not always the case, but the number of accepted configurations will be relatively close to 50%.

Results from project 2

It is a good idea to note down some relations from project 2, since we will compare a lot of our results with that project. Previously, we found that an electron in a harmonic oscillator potential had the energy

$$E_{nl} = \hbar\omega\left(2n + l + \frac{3}{2}\right)$$

Using natural units we have $\hbar = 1$. We also neglect the angular movements, so $l = 0$ and in the ground state, we let the quantum number $n = 0$. This energy does not take into account center of mass energy E_{cm}. Because of this, we will have to add $E_{cm} = 1.5$ to E_{nl} when we compare our results.

3 Implementation

Like in the previous projects, I will do all the calculations in C++ and plot the results in Python.

For this project, I have object oriented the whole program. The class `Wavefunctions` initializes a so called *functor*. The idea here is to create the wave functions as an object that acts as a (c++) function. This will make the program a lot more flexible, as we can choose whether we want to use the first or the second trial wave function.

I have also made a class called `Metropolis_quantum`. This class contains three different Metropolis solving functions. The first one uses the local energy E_{L1} in the calculation, the second uses the local energy E_{L2} and the final one is made specifically to test the virial problem.

I have also used the class `vec3`, which we developed in project 3. This class will make it easier to keep track of all the components contained within the radius variable $\mathbf{r} = (x, y, z)$, as it contains methods to do certain vector calculations, e.g finding vector length and vector addition.

Testing the Metropolis algorithm

Before we proceed with the results, we should check if the Metropolis algorithm works as it should. When calculating the local energy, we should use a numerical approach to calculate the kinetic term $\nabla^2\psi$. However, we already have the analytical expression of the local energy E_{L1}, given in equation 4. From that, we also know which terms contributes to the kinetic energy. In fact, the kinetic energy is given in equation 3, which we can use to test the Metropolis algorithm.

By implementing this term as its own function `LaplaceAnalytic`, we run the program with the analytical kinetic energy. Using 10^6 Monte Carlo cycles, $\alpha = 1$, $\omega = 1$ and no Coulomb interaction, the result is:

```
Monte Monte Carlo cycles = 1000000
```

```
Kinetic numeric = 3
Variance = 0
Accepted configs (percentage) = 0.62867
Time elapsed: 1.576s
```

So the numerical energy, using the analytical expression of the kinetic energy, is exactly 3! The variance is exactly zero as well, which is what we would expect from this test. Also note the number of accepted configurations. The algorithm for an optimal step length, given in equation 7, gives roughly 63% accepted configurations. Not exactly perfect, but good enough.

We can now also test, with the same values of α and ω, if the numerical derivation of the kinetic term works. The discretized numerical second derivative is given as

$$\left(\frac{d^2f}{dx^2}\right) = \frac{f(x+\Delta x) + f(x-\Delta x) - 2f(x)}{\Delta x^2}$$

The Laplace operator for all three dimensions is thus

$$\nabla^2 f = \sum_{i=1}^{3} \frac{f(x_i+\Delta x) + f(x_i-\Delta x) - 2f(x_i)}{\Delta x^2}$$

where $i = 1, 2, 3$ corresponds to the x, y, z components and Δx is the same for every direction. We let $\Delta x = 10^{-5}$ and running this for the first trial function gives

```
Monte Monte Carlo cycles = 1000000
Kinetic numeric = 3
Variance = 3.30573e-011
Accepted configs (percentage) = 0.628217
Time elapsed: 10.504s
```

The result is incredibly good. The energy is (almost) exactly 3 and the variance is very close to zero as well. We can therefore calculate the kinetic term of the local energy by using numerical derivation for any wave function.

However, we should also note down the time it took for the Metropolis algorithm to compute the energy expectation value. The Metropolis algorithm

was almost 10 times faster if were were to use the exact analytical expression of the kinetic term instead of calculating it numerically. Because of this, we will use the analytical expressions of the local energies to save CPU time. Also, we should really appreciate the computation speed of the Metropolis algorithm. If we look back to project 2, the Jacobi's method often times used more than 5 minutes to calculate the energies.

4 Results

4.1 The first trial wave function

Let us now plot the energy (expectation value) and variance as a function of of the variational parameter α. We can use these determine the optimal value of α, which we can use to compare with our results from project 2. One thing to note is that the optimal variational parameter may be different for different frequencies ω.

Plots of the energy and variance, as a function of α, using $\omega = 1$, can be found in figure 1 and 2 respectively. We can see that both of them has a minimum, but oddly enough at different values of α. We would expect that the energy should be at its minimum when $\alpha = 1$ or smaller. However, the plot in figure 1 says that it is not quite the case. The minimum is shifted to a slightly larger α value, where the minimum is now at $\alpha = 1.08$.

However, if we look at the plot of variance in 2, we see that the minimum is at $\alpha = 1$. Also note that the variance here is exactly zero, which also means that the standard deviation is also zero. Because of this, the calculated energy at $\alpha = 1$ is exact, i.e there are no errors here. On the other hand, if we look at table (4.1) [1], the calculated minimum at $\alpha = 1.08$ has a variance of $\sigma_E^2 = 0.0196079$, or $\sigma_E = 0.14002$, which means that the calculated energy at $\alpha = 1.08$ has a small margin of error which also means that it may not

[1] Full data set, for this case, can be found at `https://github.com/AHo94/FYS3150_Projects/blob/master/Project5/build-Project5_cpp_program-Desktop_Qt_5_7_0_MinGW_32bit-Release/Energy_Alpha_Mdistance_omega_1.00.txt`

actually be the minimum. We can therefore safely say that the optimal value of the variational parameter, using the first trial wave function at $\omega = 1$ is $\alpha = 1$.

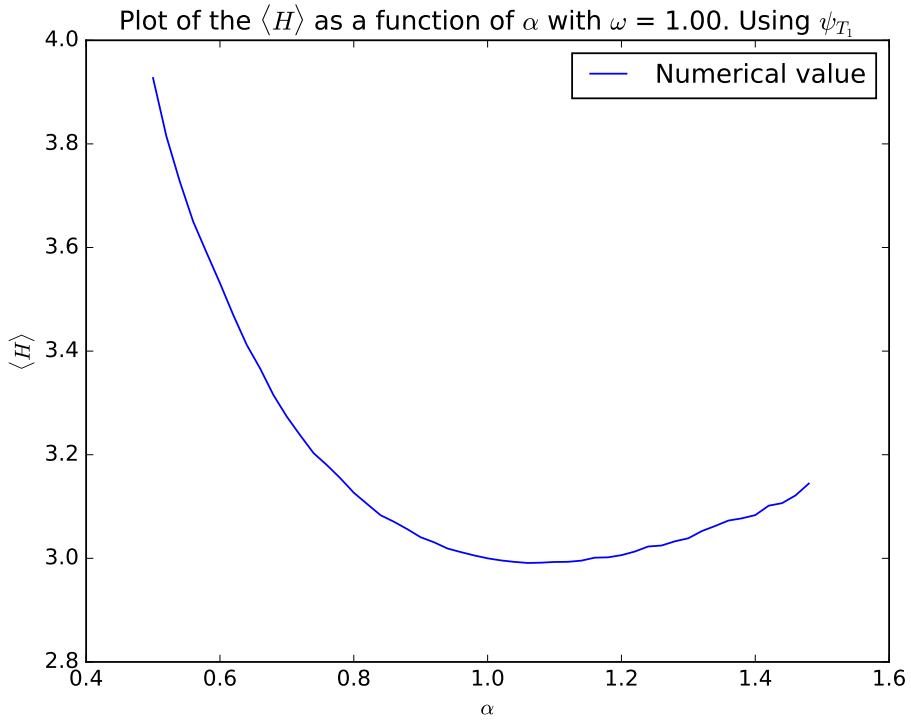

Figure 1: Plot of the Energy as a function of α with $\omega = 1$.

We obtain similar results for the cases with $\omega = 0.5$ and $\omega = 0.01$. The energy, for both cases, are also not at its minimum when $\alpha = 1$, but shifted to a slightly larger α value. However, the variance is once again zero (almost zero for $\omega = 0.01$) when $\alpha = 1$, so we can also safely say that the optimal variational parameter is $\alpha = 1$ for both cases. The plots of these can be found in the appendix.

Now that we have found the optimal variational parameter, we can use it to calculate the mean distance r_{12} between the two electrons at the energy minimum. Using the frequency values $\omega = 0.01, 0.5, 1$, the calculated mean distance is found in the output below.

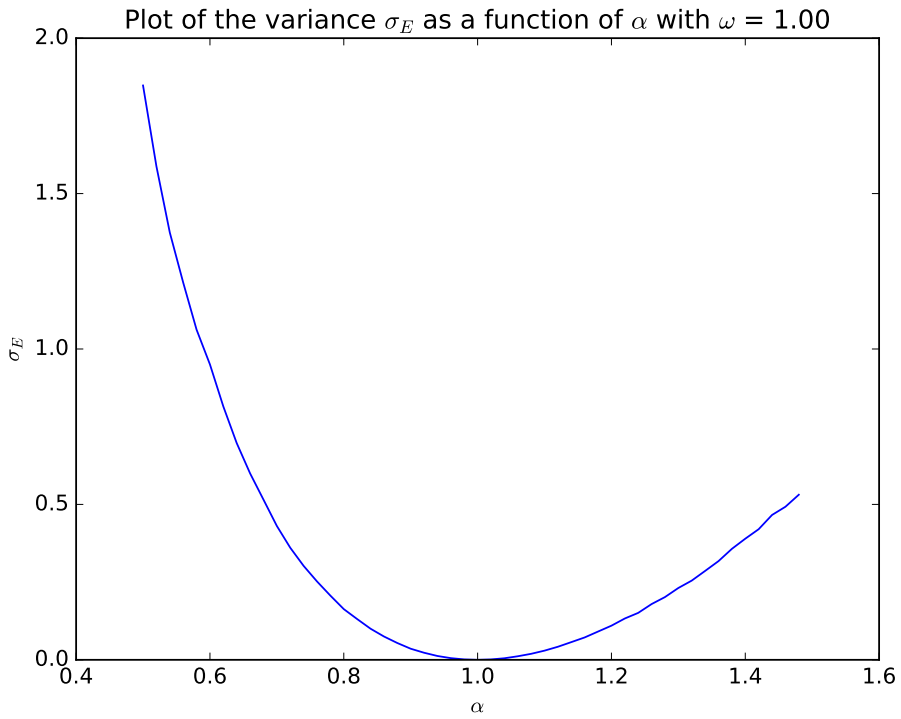

Figure 2: Plot of the variance as a function of α with $\omega = 1$.

```
Omega = 0.01
Mean distance = 16.5251
Omega = 0.5
Mean distance = 2.34729
Omega = 1
Mean distance = 1.65877
```

Let us quickly go back to project 2, where we used the analogy where two electrons are bound by a spring. The electrons now oscillate back and forth due to the oscillation of the spring.

For smaller frequencies, we can imagine that the spring is not very stiff, which means that the electrons can be drawn further apart before they get pulled back in by the spring. Because they can be further apart, the average

α	Energy	Variance
0.98	3.00541	0.0013228
1	3	0
1.02	2.9957	0.00130505
1.04	2.99328	0.00504903
1.06	2.99118	0.0112062
1.08	2.99027	0.0196079
1.1	2.99322	0.0293394
1.12	2.99256	0.0423141
1.14	2.99289	0.0572603
1.16	3.00114	0.0713725

Table 1: The energy, variance and α values near the minimum point.

distance between these two electrons will therefore be larger, which is exactly what we see from the computed results above.

For larger frequencies, we imagine that the spring is a lot stiffer, so the electrons cannot be drawn as far apart. The average distance is thus smaller when we have larger frequencies. This also agrees with the computed value above.

The results of the mean distance gives the same physical interpretation as the one from project 2 of our three dimensional harmonic oscillator. The difference here is that we use the Metropolis algorithm in this project, whereas we used the Jacobi's method in project 2. So far, we have seen that the former is a lot faster than the latter algorithm.

Let us also have a look at the number of accepted configurations for every α value we computed. From the previous section, we noted that the algorithm that is used to determine an optimal step length δ resulted to roughly 60 % accepted configurations. Ideally, we would like to have 50% accepted configurations, but 60% should be good enough. Figure 3 shows a plot of the number of accepted configurations (percentage of the total number of Monte Carlo cycles) as a function of α. We see that the number of accepted configurations is roughly 62.7% for every frequency.

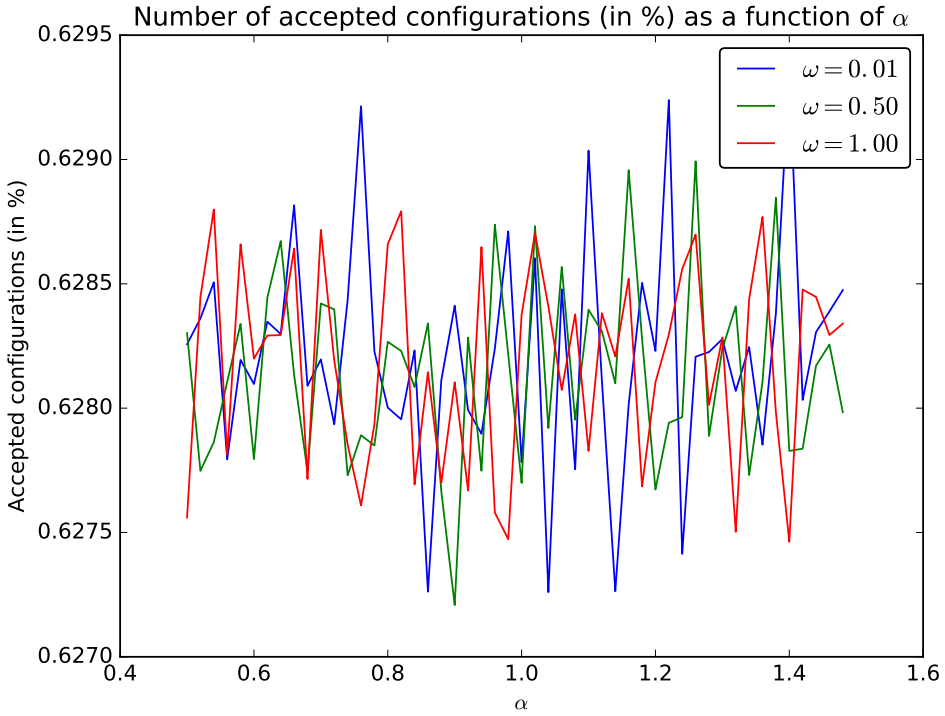

Figure 3: Number of accepted confugrations for each different Metropolis run. The number of accepted configurations is in percentage of the total number of Monte Carlo cycles.

4.2 The second trial wave function

We will now consider the second trial wave function given in 2. Let us first find the optimal variational parameters of both α and β. To do so, we can simply run the Metropolis algorithm as a for loop of different α and β values. It should, again, be noted that the optimal parameters may be different for different frequencies.

Running this for different values of α and β gives a lot of data in the output file. I have therefore used Python to find the optimal value of both the variational parameters. The strategy is to find the energy minimum in the array of energies. With the energy minimum, we can find which array element

that corresponds to this energy minimum. Once we have the array element index, we can use that to find the corresponding α and β values in their respective arrays.

4.3 Testing the virial theorem

We now want to test the virial theorem with the optimal values of α and β that we found in the previous task. The virial theorem states that the expectation of the total kinetic value is proportional to the expectation value of the total potential energy, i.e $\langle T \rangle \propto \langle V \rangle$. If we consider a pure harmonic oscillator potential, the virial theorem states

$$\langle T \rangle = \langle V \rangle$$

We will use our optimal results of the variational parameters, for $\omega = 0.01$, $\omega = 0.5$ and $\omega = 1$, to compute the expectation values of the kinetic and potential energy in the frequency range $\omega \in [0.01, 1]$. [2]

5 Conclusion

6 Appendix

[2] I was not sure if the task was to calculate an optimal α and β for every single frequency in this range or just use the optimal variational parameters to compute our results for $\omega \in [0.01, 1]$. I went ahead and did the latter choice, as calculating an optimal variational parameter for every frequency would take a lot of CPU time.

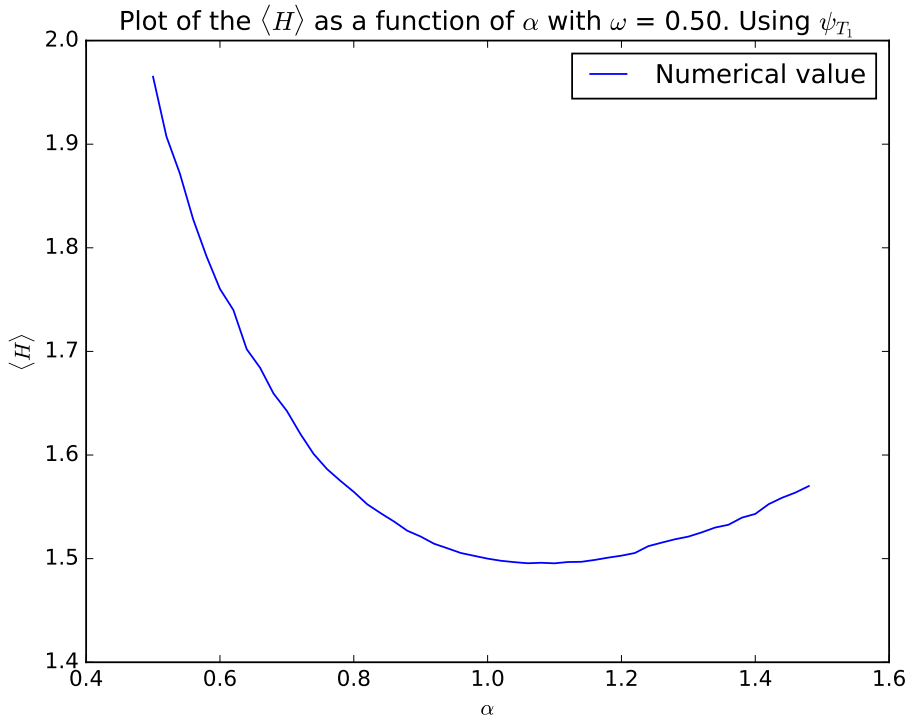

Figure 4: Plot of the Energy as a function of α with $\omega = 0.5$.

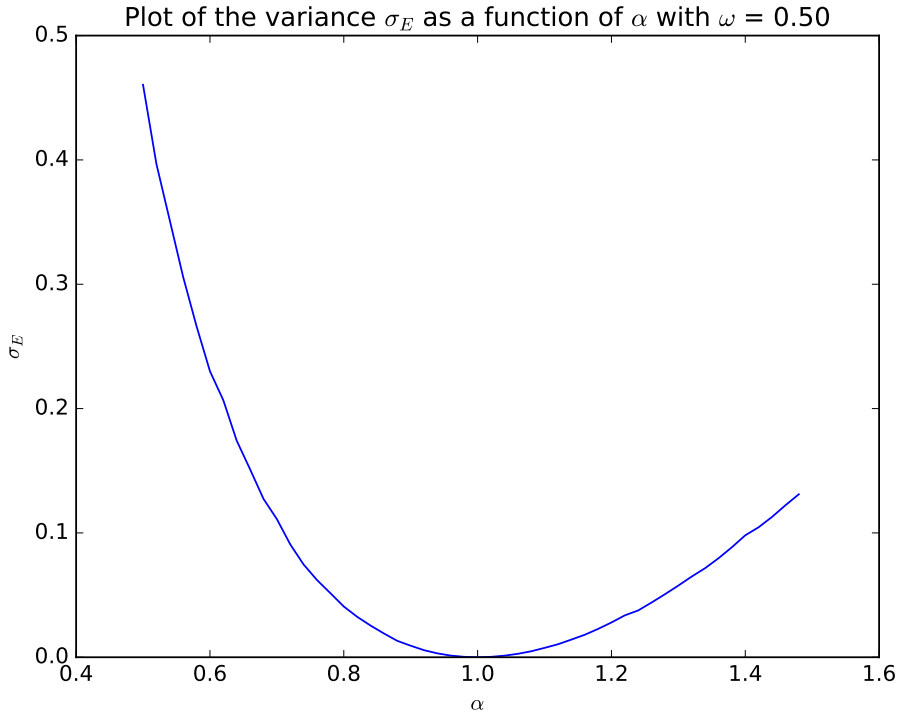

Figure 5: Plot of the variance as a function of α with $\omega = 0.5$.

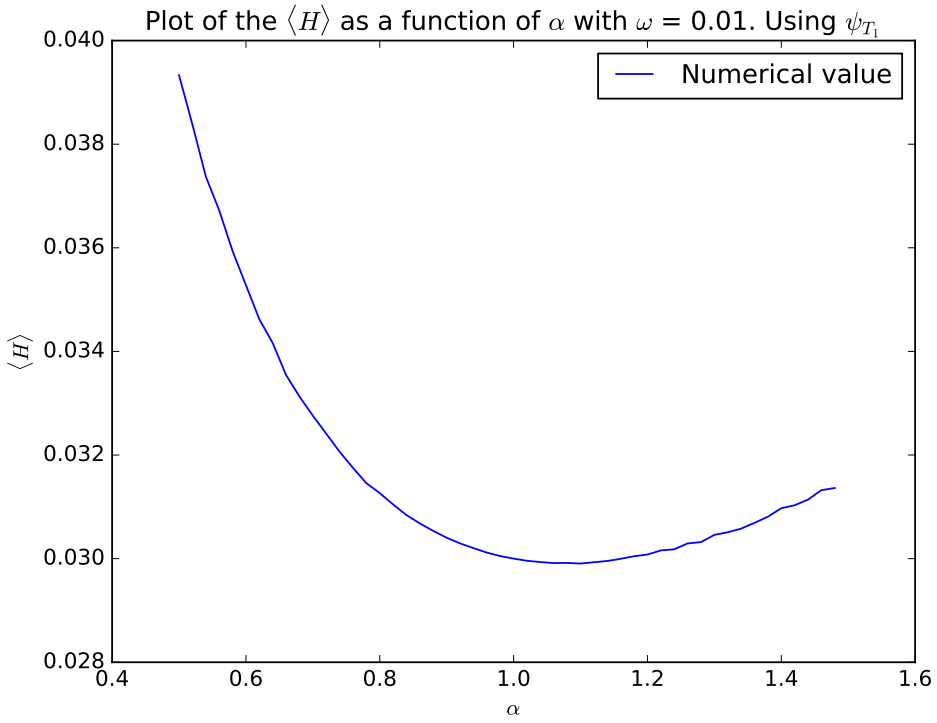

Figure 6: Plot of the Energy as a function of α with $\omega = 0.01$.

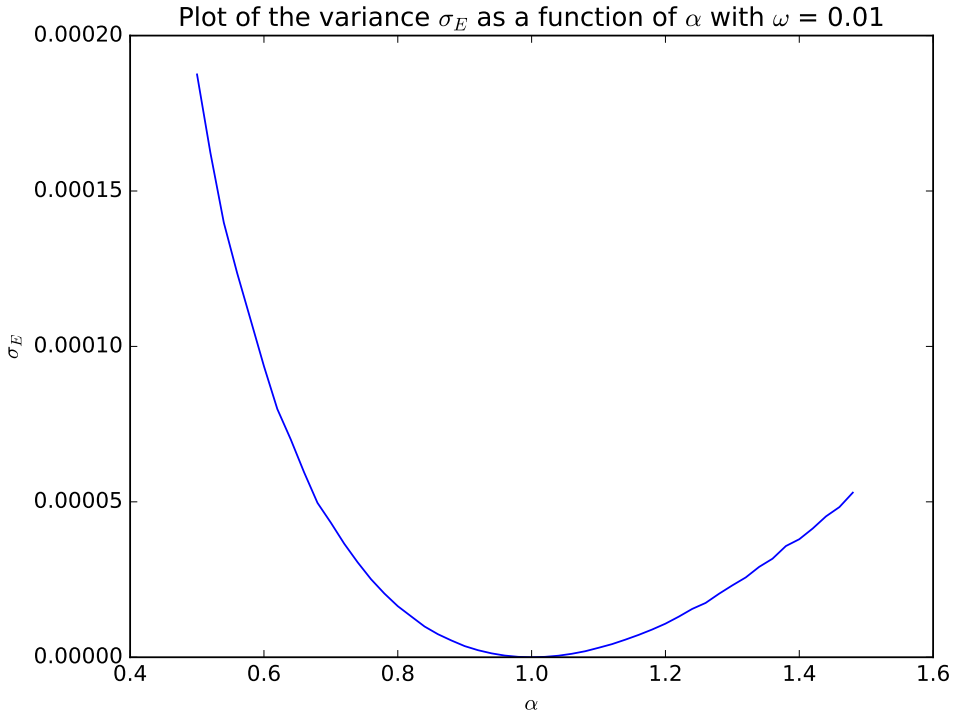

Figure 7: Plot of the variance as a function of α with $\omega = 0.01$.

References

[1] M. Hjorth-Jensen, *Computational Physics*, 2015, 551 pages