School of Physics and Astronomy

Physics Senior Honours Project
Computer Science and Physics

Quantum Harmonic Oscillator on a Computer

Sahaj Porwal
December 2020

Abstract

This project seeks to simulate and study the behaviour of a Quantum Harmonic Oscillator, and to extend this simulation to an An-harmonic Oscillator. This is done using the Metropolis Algorithm which is a version of the Markov Chain Monte Carlo Method of simulation of physical systems. The project successfully modelled the oscillations of a Quantum Harmonic Oscillator about its equilibrium position. It also modelled the double well oscillations of a quartic perturbation Quantum An-harmonic Oscillator, showcasing tunneling between the two potential minima.

Declaration

I declare that this project and report is my own work.

Signature:Sahaj Porwal Date:3/12/2020

Supervisors: Dr. Anthony Kennedy, Dr. Roger Horsley 10 weeks

Contents

1 Introduction

The early twentieth century came with of the most profound theories in the history of Physics: Quantum Mechanics. At the very core of this new area of Physics was the concept that particles(localised objects) and waves(a moving disturbance of quantities that carries energy) are one and the same. It was shown soon after that light can show properties of being a wave(Davisson and Germer Experiment, [1]) or a particle(Einstein's Photoelectric Effect, [2]).

One way to model the behaviour of these wave-particle duals is through the Quantum Harmonic Oscillator. This is a quantum mechanics equivalent of the classical harmonic oscillator that is used to model objects like a spring or a pendulum. The main difference between the two is that while the classical harmonic oscillator can be in a state which has any of a range of energies, for the quantum harmonic oscillator the allowed energy levels are fixed.

This project implements a simulation of the Quantum Harmonic Oscillator to study it's behaviour and properties using a Markov Chain Monte Carlo Method(MCMC) using the Metropolis Algorithm(described in detail in Section 2), programmed in python using the numpy package. All the code for this project can be found at [3]. The Markov Chain Monte Carlo was appropriate to implement this as it is applicable even in situations where it is not possible to directly draw samples, as well as being fairly reliable while being relatively easy to implement. Additionally the problem of the harmonic oscillator, which has a periodic wave-form, and hence satisfies detailed balance, makes an MCMC method ideal for the simulation. In particular the Metropolis algorithm is well suited for implementing a high-dimensional system, which converges to fixed point distributions, and thus is an appropriate choice for the Quantum Harmonic Oscillator and the An-harmonic Oscillator.

2 Method

The simulation is implemented using a Monte Carlo Markov Chain([4], described in subsection 2.1) method.

The Monte Carlo method is a time integration method which uses random numbers to evaluate definite integrals.

A Markov chain is a statistical tool used to model transitions of a system between its states using probabilistic rules.

2.1 Monte Carlo Markov Chain Method

The Monte Carlo Markov Chain Method consists of a combination of two different physical simulation methods:

2.1.1 Monte Carlo Method

A Monte Carlo method([5]) is a method which utilises randomly selected values to give some numerical result. The general process for the kind of method used in this project is:

1. Define some initial properties (eg. for a physical simulation involving particles, this could be positions etc.)
2. Randomly generate an increment for the initial properties
3. Calculate some observable using the properties (Energies,density,etc.)
4. Do the above steps 2 and 3 for multiple increments.
5. Aggregate the results.

For the simulation created for this project, the initial properties that are being used are the initial positions of lattice points. The observable values are the positions themselves, and the squares of the positions.

2.1.2 Markov Chain Method

The second major method used in this project is the Markov Chain([6]). This is a statistical model which revolves around determining changes in a system using probabilistic decisions.

The primary property of a Markov Chain model is that any change in the system is determined only by the current state of the system and not by previous states. In this project this is implemented by setting up a series of lattice points along a chain, and using the Metropolis Algorithm(described below in Section 2.2) to increment the value of position at each lattice point.

The end result is an aggregate of the results from each of the points along the chain.

2.2 Metropolis Method

The motion of the particle was implemented using a Metropolis algorithm(as described in [7]). The purpose of this algorithm is to produce a path for the particle using statistical methods. The process of the algorithm goes as follows:

1. For a max step size of dx, randomly select an increment for the position $\delta x \in (-dx, dx)$
2. Propose a new position $x' = x + \delta x$
3. Calculate the change in the action $\Delta S = S(x') - S(x)$
4. Calculate the metropolis factor $M = e^{-\Delta S}$
5. Generate a uniform random number $r \in (0, 1)$
6. Conditionally update the current x value to the new proposed position from Step 2. The condition for this update is $r < M$. If the condition is not met then the next position is the same as the current position
7. This process is repeated for each successive step for each lattice point.

In the python code([3]) this algorithm was implemented using a separate Metropolis class, which contains two components: a constructor to set values for the parameters needed, and a 'step' function to implement a single step in the Metropolis algorithm.

2.3 How it works

In this project, a 1-dimensional lattice of coupled Quantum Harmonic Oscillators are set up, and the transitions of each of the points in this lattice are simulated using the Metropolis Algorithm(described in subsection 2.3).

To define the process, certain parameters need to be defined. These are listed in Table 1.

In addition to the parameters, some Physical Quantities also need to be defined. The quantum action of a system is a property which measures the energy of a state of a system. In the code used for this project, the action is defined as in [8] as:

$$S = \sum_{i=1}^{N_s} \frac{1}{2}(x_{i+1} - x_i)^2 + \frac{1}{2}\omega^2 x_i^2 \tag{1}$$

Note that only the nearest neighbours have any effect on the change in action for a particular point in the lattice. So this equation was simplified and the change of action for point k on the lattice for a transition from x_i to x_i' was determined to be

$$\delta S = \sum_{i=k-1}^{k} \frac{1}{2}[(x'_{i+1} - x'_i)^2 - (x_{i+1} - x_i)^2 + \omega^2(x_i'^2 - x_i^2)] \tag{2}$$

For both equations, the parameters are as defined in Table 1.

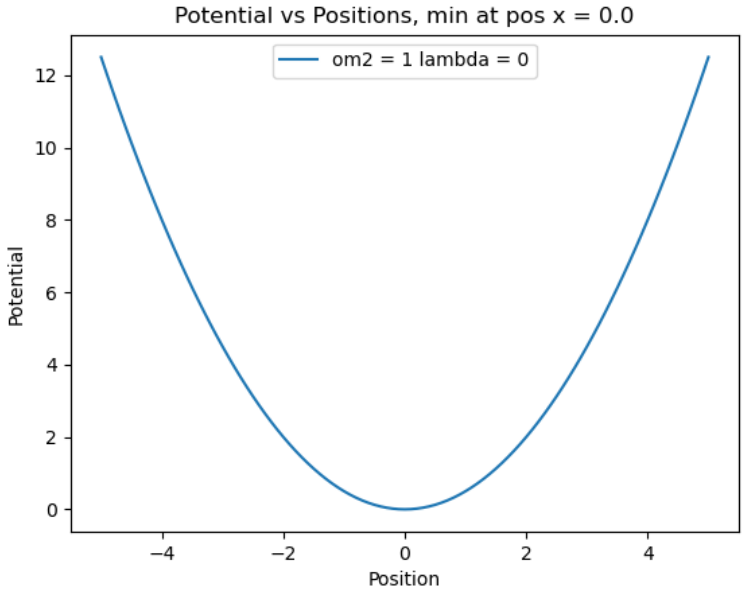

Figure 1: Potential U(x) vs x

The potential at position x for the harmonic oscillator is given by the second term in the sum in the action equation:

$$Potential\ U(x) = \frac{1}{2}\omega^2 x^2 \tag{3}$$

Minimising this equation gives a minimum potential at x = 0.0, so it is expected that the average position over the lattice over time will approach this equilibrium position by the principle of minimum energy. The potential for the harmonic oscillator is visualised in Figure 1.

The auto-correlation function is the correlation of an an observable at any particular time, with it's own value after a certain time difference. In this project the auto-correlation of an observable O for time difference δt can be quantified as ([8]):

$$AC(O) =< O(t)O(t+\delta t) > \tag{4}$$

Here the mean is over the lattice.

The other quantity, called correlation is the correlation of two lattice points at a lattice distance of δs from each other. This is defined as:

$$Corr(\delta s) =< O(s)O(s+\delta s) > \tag{5}$$

Here the mean over the number of time-steps

The general flow of the simulation is:

1. Initialise all positions in the lattice to be at position 0.0

2. Initialise list of positions(to store all positions)

3. Initialise sum of positions(to calculate mean)
4. Initialise sum of squared positions(to calculate variance)
5. Loop over the number of time steps:
   (a) Append the current position to the list of positions
   (b) Add the current position to sum of positions
   (c) Add current position squared to the sum of squared positions
   (d) Loop over the lattice points in a randomised order.
      i. Calculate the next position for the current lattice point using the Metropolis Algorithm and the action defined above.
      ii. Update the positions if Metropolis condition is satisfied
   (e) Update the time step
6. Determine the Mean and Variance from the sum of positions and sum of squared positions.
7. Determine the auto-correlation function and its error for all possible time differences.
8. Determine the correlation function and its error for all possible lattice separations.
9. Code for visualisation

3 Results & Discussion

The methods described above were used to simulate both a harmonic oscillator and an approximation for an an-harmonic oscillator.

3.1 Harmonic Oscillator Results

The harmonic oscillator was simulated using 100 lattice points for 50,000 Metropolis steps. The maximum step-size for this simulation was 0.8 and the ω^2 value was 1.0. The acceptance rate for these parameters was $\sim 73\%$. The mean position of the lattice points was observed at $x \sim -0.006$ and the mean squared position was observed as $< x^2 > \sim 0.450$.

The graphs produced from this simulation are: a representative path for motion of one lattice point(Figure 2), a histogram of positions along with the expected Gaussian distribution for positions(Figure 3).

In addition to these, the auto-correlation function of position x was plotted against time-difference. The values plotted were: position x vs the time difference between positions(Figure 4 for auto-correlation of first 100 time-differences, 5 for log auto-correlation function of x. From these values, the integrated auto-correlation was obtained to be used

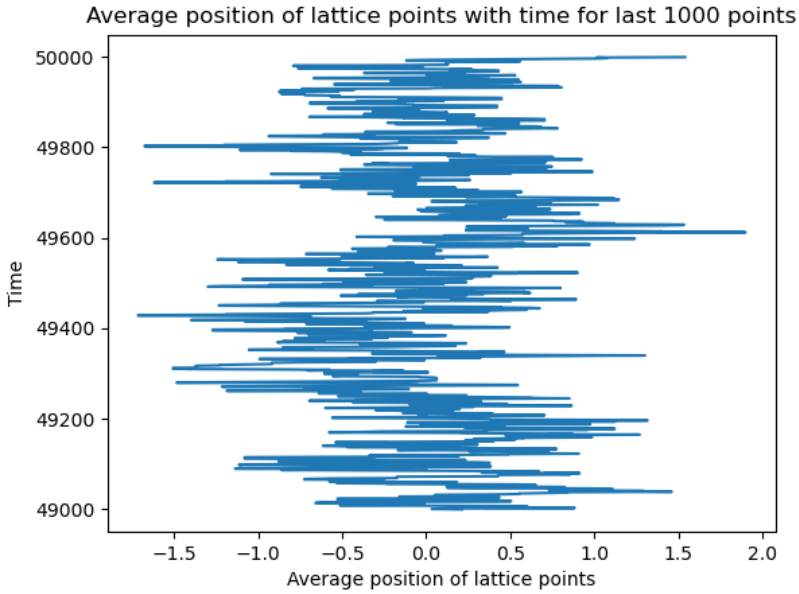

Figure 2: Representative Average Path of Particles

to obtain more reliable error bars by considering the systematic error(error from noise). This was obtained by the Central Riemann sum, which is given by:

$$IntAC = \sum_{i=1}^{N_s-1} AC(\frac{x_i + x_{i+1}}{2}) \tag{6}$$

The correlation function was also plotted which can be found in Figure 6, with the log-correlation function in Figure 7.

The graphs for auto-correlation and correlation both contain errors. For the auto-correlation function, the errors are calculated by using the standard deviation of the product of the observable(positions of all lattice points) at time t and at time $t + \delta t$. Similarly for correlation function except it considers the observable at all times for the product between lattice site s and $s+\delta s$ where the symbols mean the same as in Equations 4 and 5. These errors are then multiplied by a factor of $2 * IntAC + 1$, where IntAC is the Integrated Auto-correlation obtained from the auto-correlation values.

3.2 Harmonic Oscillator Discussion

The values obtained from the simulation were compared with some theoretical values that can be determined by the properties of the Quantum Harmonic Oscillator.

The acceptance rate is in the range which provides reliable results(50-80%). A lower acceptance rate would mean not much change in the position of each lattice point, and a higher acceptance rate would mean that some non-permitted transitions are taking place. The mean position approaches 0.0, which is consistent as the system should oscillate around that point(the potential minimum for the harmonic oscillator, ie. the point of equilibrium). The mean squared position can be derived from the potential and the wave

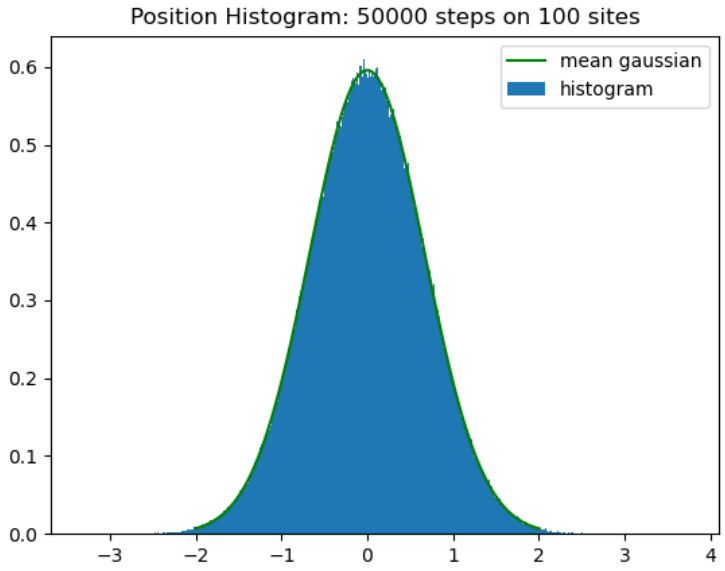

Figure 3: Position Histogram

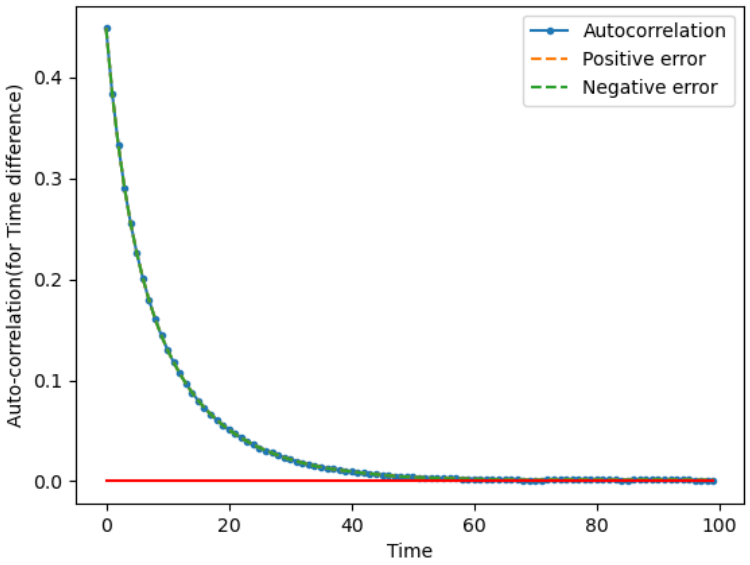

Figure 4: Auto-correlation of x for 100 time-differences

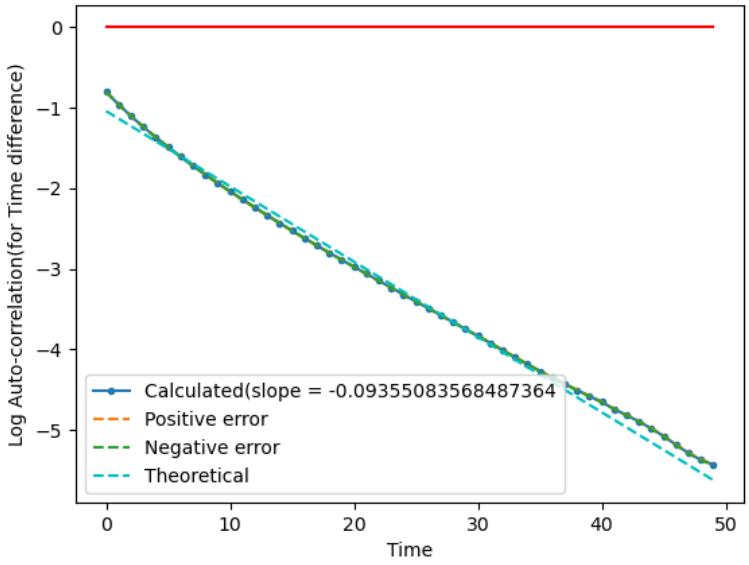

Figure 5: Log auto-correlation of x vs Time difference for first 50 time differences

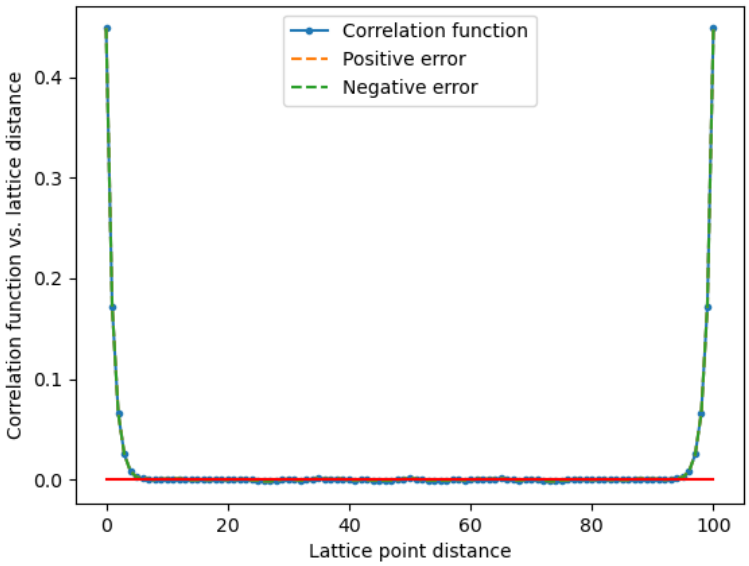

Figure 6: Correlation of x vs Lattice Distance for the whole lattice

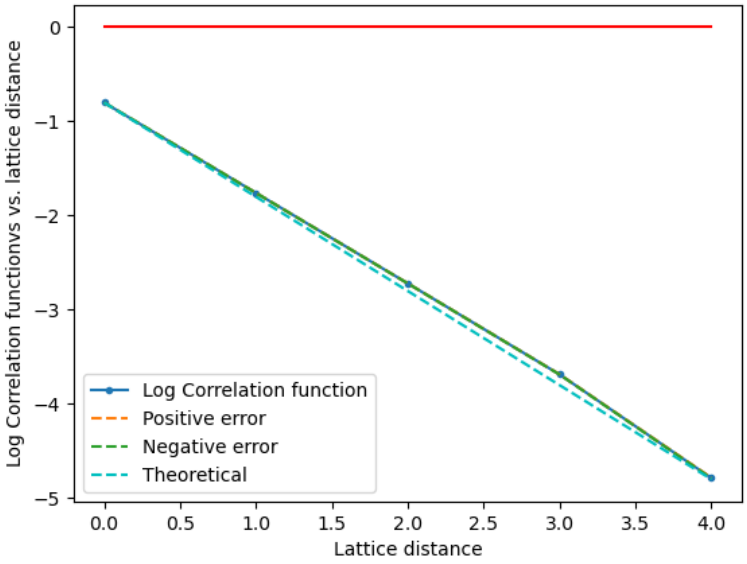

Figure 7: Log correlation of x vs Lattice Spacing for first 5 lattice distances

functions(as shown in [9]) to be :

$$< x^2 >= \frac{1}{2\omega\sqrt{1+\frac{\omega^2}{4}}}(\frac{1+R^{N_\tau}}{1-R^{N_\tau}}) \tag{7}$$

Where N_τ is the number of iterations and the variable R is defined as:

$$R = 1 + \frac{\omega^2}{2} - \omega\sqrt{1+\frac{\omega^2}{4}} \tag{8}$$

For the defined simulation parameters, this theoretical value is determined to be $< x^2 >= 0.447$. The calculated value agrees with the theoretical value(error of $< 1\%$ of the theoretical value).

The position histogram(Figure 3) shows a Gaussian shape, which very closely matches the theoretical Gaussian plotted using the above $< x >$ and $< x^2 >$ values. The auto-correlation function shows the amount of correlation between two time-steps at a certain time-difference from each other, ie. how much the state of a system at one particular time affects the state of a system after a certain number of time-steps. The auto-correlation function vs. time difference shows a rapid decay to zero(at approx. time difference of 40 time-steps) after which point it shows a random oscillation about zero(this is the noise). This matches the expectation that time-steps close to each other will have a stronger effect on each other than time-steps which are much further away. The log auto-correlation function for the first 50 time differences is a line with negative slope indicative of the auto-correlation time for the simulation. The correlation-function shows the amount of correlation between lattice points based on the distance between them on the lattice. The graph shows that only the nearest points have any significant correlation which is consistent with the action involving only the nearest neighbours to determine the probability of transitions.

3.3 An-harmonic Oscillator Results and Discussion

The an-harmonic oscillator is different from the harmonic in that there is an additional term in the potential part of the action equation. The interesting thing about the an-harmonic oscillator is there is no definite analytical solution, however it's study is important to understand real phenomena, since real phenomena don't always have a potential which exactly matches the harmonic case(there are perturbations). This project uses a quartic(Power 4) perturbation to get a new action equation([8]):

$$S = \sum_{i=1}^{N_s} \frac{1}{2}(x_{i+1} - x_i)^2 + \frac{1}{2}\omega^2 x_i^2 + \frac{1}{4}\lambda x_i^4 \tag{9}$$

where λ is the an-harmonic constant which is sometimes referred to as the an-harmonicity, which defines the strength of the an-harmonic perturbation.

For the following results, the simulation was run for 20,000 steps on 100 lattice points, with a step-size of 2.0, an ω^2 value of -2.0 and a λ value of 0.5. The $omega^2$ value is negative here to create a double potential well, with a potential barrier between them, which is characteristic of the an-harmonic potential. The potential for the an-harmonic well contains an additional term scaling with the an-harmonicity:

$$U(x) = \frac{1}{2}x^2 + \frac{1}{4}\lambda x^4 \tag{10}$$

This is a quartic equation which has a turning point(the potential 'bump') at x = 0 and two minima at $x = \pm\frac{\iota\omega}{\sqrt{\lambda}}$, where ι is the imaginary number $\iota = \sqrt{-1}$. A plot of this potential for the defined parameters for this part of the project($\lambda = 1, \omega^2 = -1$), is in Figure 10 with minima at $x = \pm 2$.

The histogram of positions can be found in Figure 8. It can be seen that the graph shows two Gaussian distributions, one each centred at the two minima. Figure 9 shows how this happens. This graph shows the plot of a single lattice point over time. The path shows that the lattice point 'tunnels' from one minima to the other multiple times during the time observed. In a classical setting, this would not occur without providing additional energy to the system from an external source.

Looking at the auto-correlation function of x in Figure 12, it can be seen that the auto-correlation starts with the previous exponential decay(system reaching potential minima), then decreases very slightly over time. This is expected as the positions remain largely the same(ie. close to the potential minima), except for the occasional tunneling between the two minima, which would cause the auto-correlation to be high even for large time-difference.

The correlation function of x in Figure 14 shows an exponential decay with lattice distance till halfway across the lattice(50 lattice points), then an exponential increase till the full lattice length. This can be explained by the fact that the longer the system is in one of the two minima the more likely it is to tunnel to the other minima. This can also be seen visually in Figure 9. This is due to the nearest neighbours influencing the overall evolution of a particular lattice point.

Keeping the minima at the same x(ie. keeping the ratio of ω and $\sqrt{\lambda}$ the same), the potential curves were plotted to see the behaviours of the potential barrier between the two minima. It was found(as can be seen in Figure 11) that with increase in the

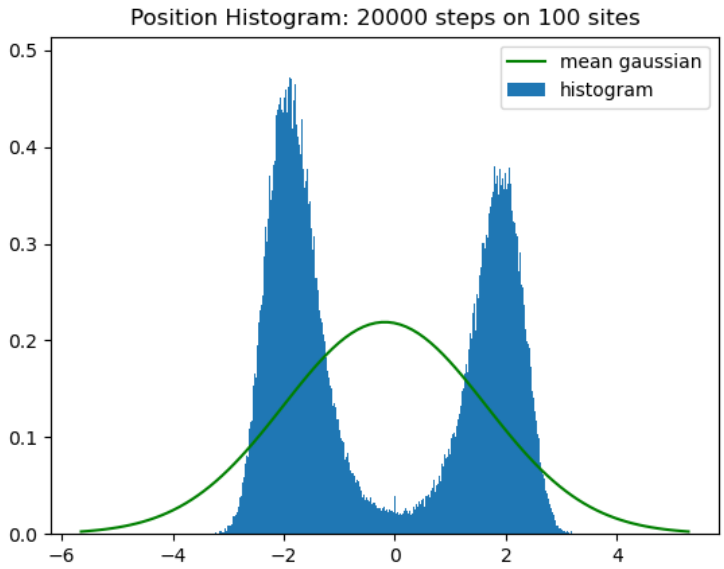

Figure 8: Position Histogram(An-harmonic Oscillator)

scaling factor(a constant multiplying both ω^2 and λ), the bump becomes higher and it would become more difficult for the lattice point to tunnel from one to the other, and vice versa(lower scaling factor, lower bump, easier tunneling).

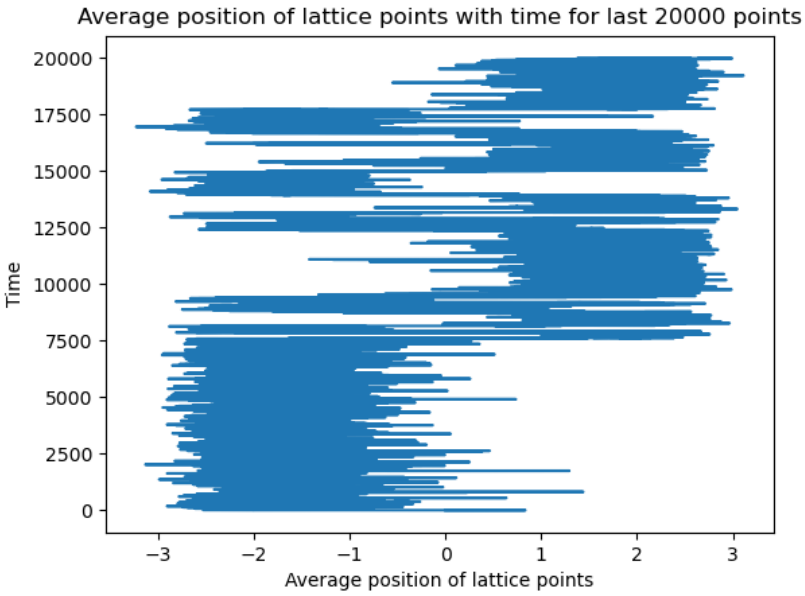

Figure 9: Representative Path(An-harmonic Oscillator)

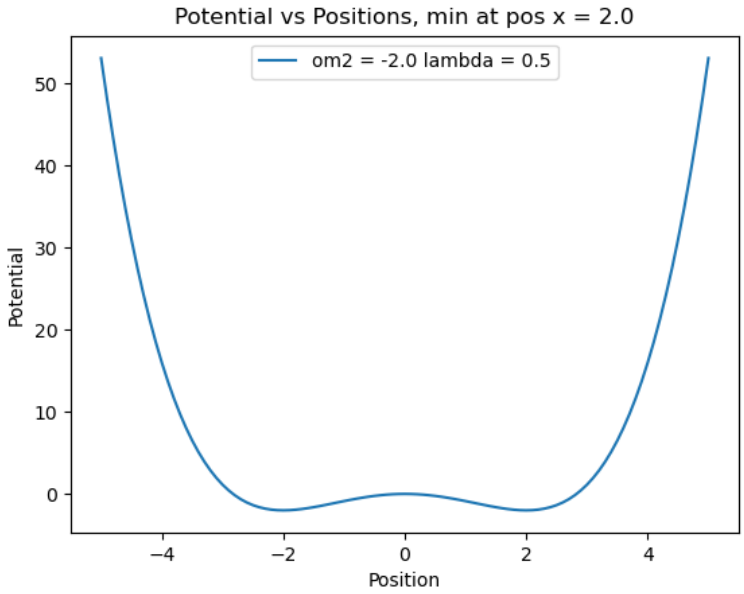

Figure 10: Potential vs Position

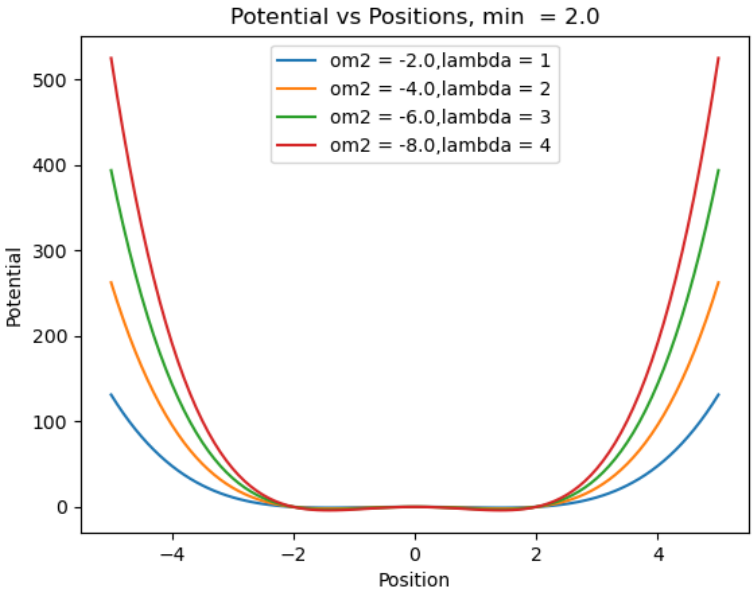

Figure 11: Variation of Potential with scaling

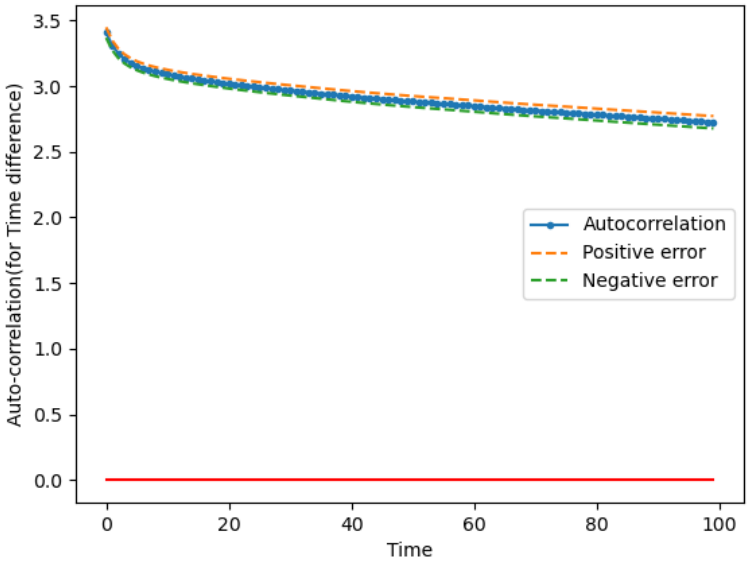

Figure 12: Auto-correlation function(An-harmonic Oscillator)

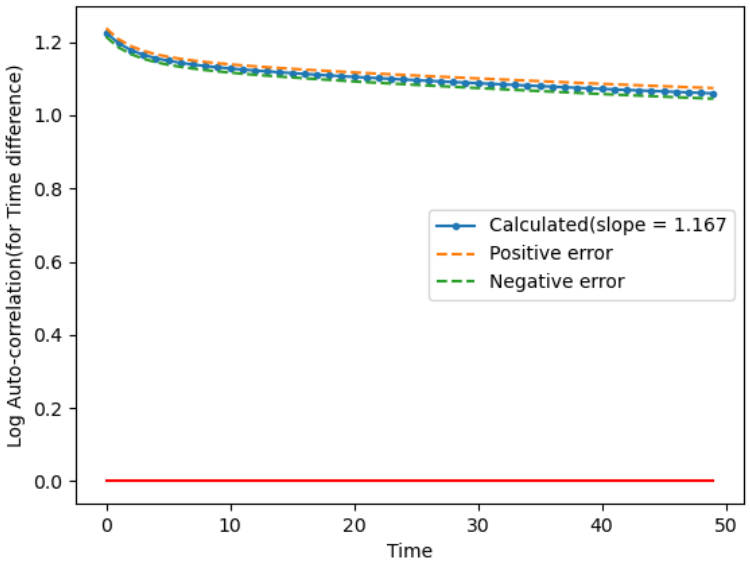

Figure 13: Log Auto-correlation function(An-harmonic Oscillator)

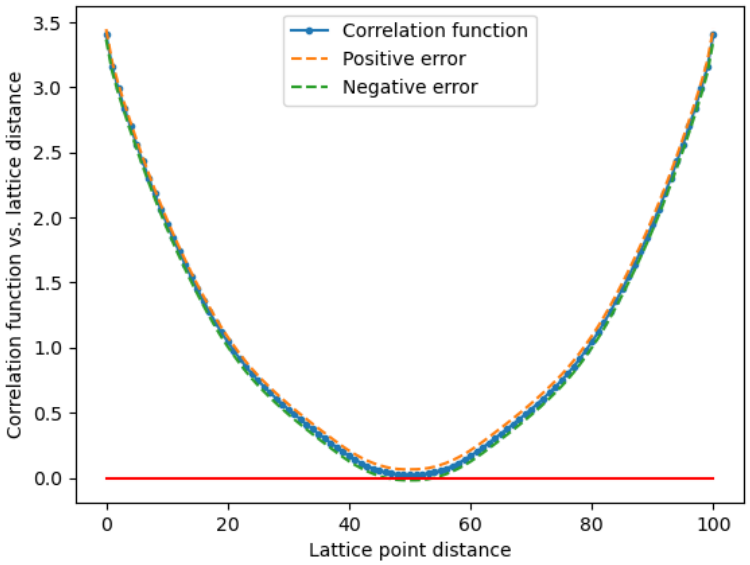

Figure 14: Correlation function(An-harmonic Oscillator)

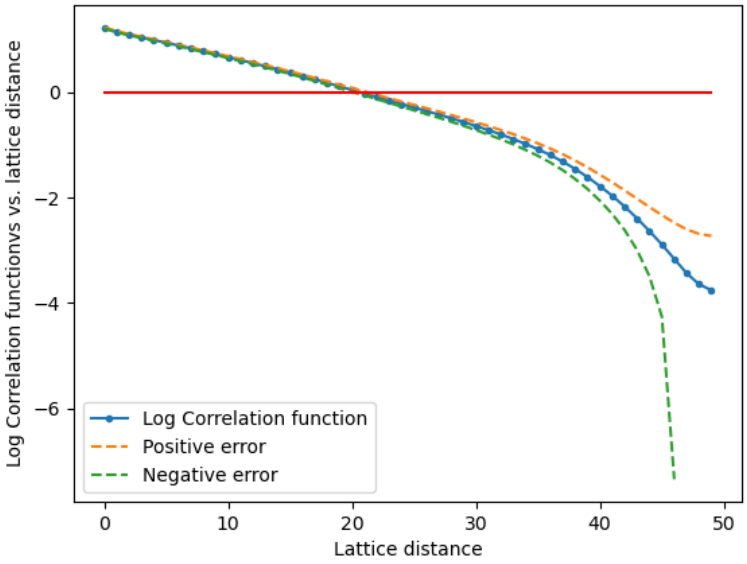

Figure 15: Log Correlation function(An-harmonic Oscillator)

4 Conclusions

The trajectories produced by the simulation successfully reproduced expected results for the harmonic oscillator and gave a fair intuition for how a double potential an-harmonic oscillator would evolve over time with nearest neighbours interactions. This project lays the foundation for a lot of future applications of simulating these systems, and the code can be altered to simulate larger, 2-D and 3-D systems as well as for simulating interactions between separate harmonic/an-harmonic oscillator systems and can be used in areas like: the study of phonons(some of the applications of Phonons are explored in [10]), simulating The Hooke's atom([11]) with the harmonic oscillator, and also in making a representative model for deviations(perturbations) from harmonic behaviour using the an-harmonic oscillator.

5 References

References

[1] C. Davisson and L. H. Germer. Diffraction of electrons by a crystal of nickel. *Phys. Rev.*, 30:705–740, Dec 1927.

[2] A. Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Annalen der Physik*, 322(6):132–148, January 1905.

[3] Porwal Sahaj. Quantum harmonic oscillator. https://github.com/sporwal98/SHP$_{Q}uantumHarmonicOscillator$, 2020.

[4] Masanori Hanada. Markov Chain Monte Carlo for Dummies. *arXiv e-prints*, page arXiv:1808.08490, August 2018.

[5] J. M. Perlado. The monte carlo method. In A. G. Colombo and A. Saiz de Bustamante, editors, *Systems Reliability Assessment*, pages 23–43, Dordrecht, 1990. Springer Netherlands.

[6] Ka Chan, C. Lenard, and Terence Mills. An introduction to markov chains. 12 2012.

[7] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. , 21(6):1087–1092, June 1953.

[8] Marise J. E. Westbroek, Peter R. King, Dimitri D. Vvedensky, and Stephan Dürr. User's guide to Monte Carlo methods for evaluating path integrals. *American Journal of Physics*, 86(4):293–304, April 2018.

[9] M Creutz and B Freedman. A statistical approach to quantum mechanics. *Annals of Physics*, 132(2):427 – 462, 1981.

[10] Atsushi Togo and Isao Tanaka. First principles phonon calculations in materials science. *Scripta Materialia*, 108:1 – 5, 2015.

[11] W. Brown and Ronald White. Perturbation theory of the hooke's law model for the two-electron atom. *J Chem Phys*, 53, 12 1970.

Parameter	Description
ss	Step Size, the maximum increment in position for a single metropolis step
niterations	Number of time steps
ns	N_s, Number of sites, the number of lattice points on the Markov Chain
om2	ω^2, the square of the dimensional parameter(squared as that is what is used in calculation of potential)
anh	an-harmonic constant λ, this defines the contribution of the an-harmonic term in the action($\lambda = 0$) reproduces the harmonic case

Table 1: Parameters for the simulation