School of Physics and Astronomy

Senior Honours Project

The Quantum Harmonic Oscillator on a Computer

James C. Clarke
March 2020

Abstract

A Monte Carlo Markov Chain process is used to compute Path Integrals on a 1D lattice of position values, which is used to investigate the Quantum Harmonic Oscillator and the Anharmonic Double Well potential with the former being used to ensure the system is producing correct results. This was found to be the case for some observables but the issues with the results from the harmonic potential resulted in a lack of confidence in those obtained from the double well potential. Further improvements to the system used are recommended in order to potentially obtain more reliable results.

Declaration

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

Signature: Date:

Supervisors: Professor A D Kennedy, Dr R Horsley 10 Weeks

Contents

1 Introduction

The Quantum Harmonic Oscillator, QHO, is a standard system used widely within quantum mechanics and is discussed in almost every introductory course on the subject. It is the quantum analog of a simple harmonic oscillator and as such shares many of the same characteristics.

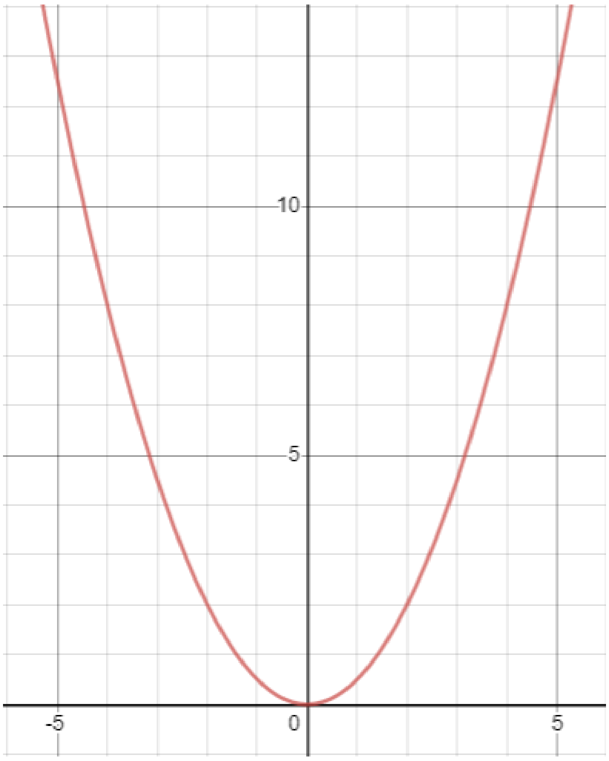


Figure 1: The form of the QHO potential when considering a particle of $m = \omega = 1$.

Modelling this potential (Figure 1) on a computer is quite simple to do in a dynamic manner, as it is not unlike a pendulum. However, by using a different approach to quantum mechanics developed by Richard Feynman, it is possible to use the concept of Path Integrals to describe the motion of a particle in this, and other quantum oscillator potentials. By doing this, it is possible to move beyond considering those potentials that have analytical solutions such as the QHO, and move towards potentials such as Anharmonic potentials that are more difficult to produce expected results for.

This report will first outline in Section 2 the necessary background and theory for both the quantum mechanical processes at play and the computational methods employed. It will then discuss the specific implementation in this project in the following section, before discussing the results in Section 4. An appendix containing the code used can be found at then end of this report.

2 Background and Theory

2.1 Quantum Harmonic Oscillator

The Schrödinger equation is a fundamental tool in the consideration of quantised systems that are found in Quantum Mechanics. It is an initial value problem with solutions satisfying the following:

$$i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi \tag{1}$$

where ψ is the wavefunction of the particle being considered and $\hat{H}$ is the Hamiltonian. The wavefunction is defined as a probability wave, usually relating to matter. The Hamiltonian operator seen in the above equation is the operator form of the Hamiltonian of classical mechanics that describes the total energy of the system. In this form it produces an eigenvalue problem when operating on the wavefunction, with its corresponding eigenvalue solutions giving the energy of different states the system may be in. This can be expressed in two different notations, wavefunction notation and Dirac notation:

$$\hat{H}\Psi(\vec{r},t) = E\Psi(\vec{r},t)$$

$$H\left|\Psi(\vec{r},t)\right\rangle = E\left|\Psi(\vec{r},t)\right\rangle$$

Dirac notation has the benefit of simplifying much of the algebra of quantum mechanics, provided the reader is able to follow the concept of each "bra" and "ket" as being vectors in a space defined by a certain basis. There are many resources, including Dirac's own paper on the matter[2], which can be used to gain insight on this notation but the main points required for the purposes of this project are that quantum systems may be defined in terms of quantum numbers which refer to the state the system is in, with these states often being referred to as energy levels and hence if we consider the relevant basis we may replace the wavefunction with a ket representing the n^{th} energy level of a system:

$$H\left|n\right\rangle = E_n\left|n\right\rangle \tag{2}$$

where E_n is the energy eigenvalue of the state the system is in when the Hamiltonian operator is applied.

The Hamiltonian is defined as:

$$H = K + V$$

where K is the kinetic energy of something, in the case of most quantum mechanics a point like particle, which is moving in some potential V.

In most quantum mechanics, when considering the kinetic energy possessed by a particle the momentum is used as it classical would be. If we consider the one-dimensional case:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \hat{V}(x)$$

When considering a classical undamped harmonic oscillator, the force on the particle oscillating is defined as: $F = -kx^2$ and due to the relation that U the potential energy of the system is given by $\frac{dF}{dx} = U(x)$, we find that the classical potential is: $U(x) = \frac{k}{2}x^2$. The k used above is the spring constant, which comes from Hooke's law. It has the relation that $k = m\omega^2$ where ω is the angular frequency of the oscillations and m is the mass of the particle.

If we take this potential and substitute it into the Hamiltonian we get the result:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{k}{2}\hat{x}^2 = \frac{\hat{p}^2}{2m} + \frac{m\omega^2}{2}\hat{x}^2 \tag{3}$$

thus we have an expression for the Hamiltonian of a particle in a harmonic oscillator potential, which we may use to investigate a quantum harmonic oscillator system.

In order to work out the energy eigenvalues of the system, the ladder operators $\hat{a}$ and $\hat{a}^\dagger$ are used to consider the system in the number basis seen in Equation 2. These operators, developed by Paul Dirac has the following forms:

$$\hat{a} = \sqrt{\frac{m\omega}{2\hbar}}\left(\hat{x} + \frac{i}{2m\omega}\hat{p}^2\right)$$

$$\hat{a}^\dagger = \sqrt{\frac{m\omega}{2\hbar}}\left(\hat{x} - \frac{i}{2m\omega}\hat{p}^2\right)$$

These operators are often referred to as the creation and annihilation operators due to their behaviour whenever they interact with a $|n\rangle$ state being as follows:

$$\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle \quad \hat{a}^\dagger = \sqrt{n+1}\,|n+1\rangle$$

by substituting these operators into Equation 3, it is possible to extract the energy eigenvalues of the hamiltonian in terms of the quantum number n. Many rigorous derivations of these values exist and so they will not be repeated here, the result of this consideration provides energy levels that satisfy:

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right) \tag{4}$$

where $n =$ 0,1,2,3,... and denotes the energy level of the system. By subtracting the $n = 1$ and $n = 0$ energy values it is easy to see that $\Delta E_n = \hbar\omega$.

The ground state of the Quantum Harmonic Oscillator is the state for which $n = 0$ and the corresponding normalised wavefunction is:

$$\psi_0 = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} \exp -\frac{m\omega x^2}{2\hbar} \tag{5}$$

and the corresponding probability density function is given by:

$$\left|{\psi_0}^2\right| = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{2}} \exp -\frac{m\omega x^2}{\hbar} \tag{6}$$

From these, it is easy to see that it has the form of a gaussian centred at the origin. We can now consider the expectation value of the position observable, $\langle n|\,\hat{x}\,|n\rangle = \langle x\rangle$ which in this case will be 0 due to the form of the probability density function.

One more observable that will be considering during this project is $\langle x^2 \rangle$. The expected value for this quantity can be obtained by considering the definition of the position operator, $\hat{x}$, in terms of the ladder operators:

$$\hat{x} = \sqrt{\frac{\hbar}{2m\omega}} \left(\hat{a}^\dagger + \hat{a} \right) \tag{7}$$

thus we can write the expectation value of $\hat{x}^2$ as:

$$\langle \hat{x}^2 \rangle = \frac{\hbar}{2m\omega} \langle n| \left(\hat{a}^\dagger + \hat{a} \right)^2 |n\rangle = \frac{\hbar}{2m\omega}(2n+1) \tag{8}$$

A more challenging potential, and one that does not currently have a complete set of definite analytical solutions, is the Anharmonic Oscillator.

The Anharmonic oscillator's potential, $V(x)$, has the following form:

$$V(x) = \frac{C}{2}\hat{x}^2 + \frac{\gamma}{4}\hat{x}^4 \tag{9}$$

where the constants C and λ may be positive or negative. Different combinations of these constants produce different potentials, some more interesting than others. A "Pure Anharmonic Potential" can be thought of as a perturbation of the Quantum Harmonic Oscillator discussed above, with a positive C and γ. The potentials considered in this project have a negative quadratic term, C, and a positive but small quartic term, γ. This combination produces a potential of the form seen in Figure 2, which shows off the so-called "Double Well" potential. If the signs of the constants were inverted, then the "Inverted Double Well" would be obtained, and this can also be considered a perturbation of the Quantum Harmonic Oscillator but with a negative quartic perturbation.

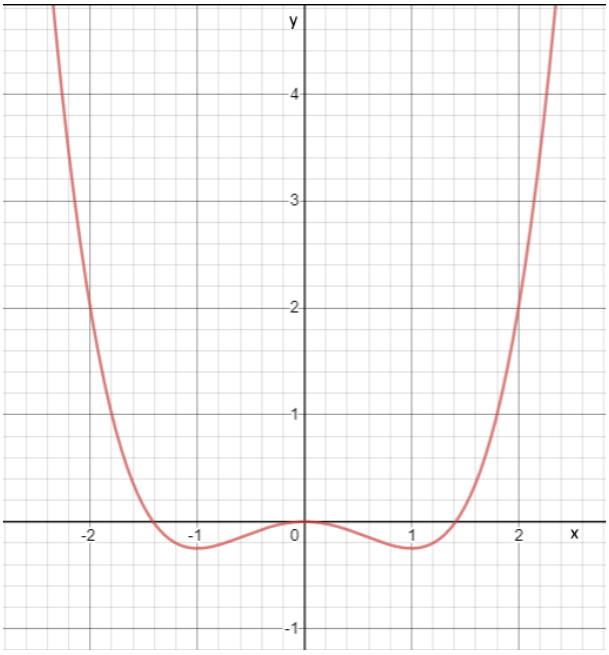


Figure 2: Form of the Anharmonic potential when $C = -1$ and $\gamma = 1$.

The main point of interest in this parameter set is the presence of the two "wells", in which a particle may potentially find itself. The "bump" in between these presents an opportunity to potentially witness states that have a wavefunction which imply the possibility of a particle tunnelling from one well to the other as these are the two most likely positions for it to occupy. The Double Well potential is of particular interest in quantum field theory, and the search for a method to determine explicitly the energy eigenvalues of this potential has resulted in the development of methods such as Feynman Path integrals which are explained in a later section. There have been numerous considerations of this problem with many estimations being made by many different methods[5][7][10], including some by computational approaches as will be attempted in this project[1][11].

2.2 Monte Carlo Methods

Named after the ward of Monaco famous for its extravagance and gambling scene (although more synonymous with Formula 1 racing nowadays), Monte Carlo methods use chance, or rather randomness, in order to solve or at least simulate physical problems.

The general method is to utilise randomly generated numbers to drive a stochastic process that attempts to follow the time dependence of a system that is being "simulated". These methods require a system to be either naturally discrete, or approximated into a discrete form. As a result these methods see considerable use in Statistical Physics, where partition functions make the problems discrete in some manner.

Some examples of applications of these methods include simulating the Ising Model, showing the evolution of Brownian motion, and modelling the separation of two liquids of different densities.

The overall aim is to determine an observable from a Monte Carlo simulation that uses a certain string of random numbers, and then by producing more results from further simulations (or simply generating more results in the same simulation depending on the problem) with different random number generations, it is possible to extract trends or average values based on these banks of results.

An illustrative example of how Monte Carlo methods may be used to find the value of some quantity, is determining the value of π by Monte Carlo methods. This is a common example, but a particularly useful run-through of this example is found in Reference 3 which is also a good resource for learning more about Monte Carlo methods and their uses. The calculation of π may be done by generating a pair of random numbers, each on the interval (0,1). These represent xy coordinates of a particular point on the xy-plane. Calculating the distance of this point from the origin and comparing this to 1 is used to decide whether the point lies within the upper-right corner of a unit circle centred at 0. Without discarding points which do not satisfy this condition, it is possible to estimate the area of the quarter unit circle by counting the fraction of points within the circle. Thus, $\pi \approx 4\frac{N_o}{N}$, where N_o is the number points which lie within the circle and N is the total number of points produced. It can be shown that generating more points will converge the answer to the expected value of π[6].

To some, this example may be rather obviously a method of numerical integration, but a non-deterministic approach to the method. This is a key usage of the Monte Carlo process within Physics, and is particularly useful when attempting to evaluate integrals of higher dimension than 1 as its error scales as $\frac{1}{\sqrt{N}}$ where N is the number random values used to approximate the integrals. In contrast to some deterministic methods of numerical integration, this may be the preferred method for certain problems. Monte

Carlo integration is often paired with a Markov Chain process, this is the topic of or next section.

2.3 Markov Chains and the Metropolis Algorithm

By using the idea of importance sampling averages of values from a particular sample drawn from a distribution may be obtained without needing to have the same distribution as the distribution that is of interest.

This is the core idea behind the establishment of a Markov Chain. Markov Chains are stochastic models that describe a sequence of possible states or events. These states or events will be called configurations for the rest of this section in order to attempt to link this explanation into the content of the project better. Each configuration's probability depends only on the previous state obtained. These are used extensively in various statistical models, but when combined with Monte Carlo methods, the resulting Markov Chain Monte Carlo process (MCMC process) can be used to simulate the generation of values from a probability distribution to enable importance sampling.

In order to combine these two processes, the Metropolis-Hastings algorithm (often referred to as merely the Metropolis algorithm) is employed. The Metropolis algorithm essentially compares a newly proposed configuration to the previously found configuration, deciding which is more likely to fit the distribution being considered. This sets up a Markov Chain.

A more in depth explanation is best seen through an example. As mentioned above, Monte Carlo methods may be used to simulate the Ising Model. This model uses a lattice of N^d spins which are either up or down, where d refers to the dimensions of the lattice. Considering the case where $d = 1$ we have a string of spins which may be up or down.

The spins may be randomly aligned, or deliberately positioned to start, this forms the first configuration in the Markov Chain. To generate the next configuration a random number corresponding to a point on the lattice is generated. At this stage, there are a number of different styles of dynamics which may be employed, we will consider the simplest which is known as Glauber dynamics. In Glauber dynamics, the lattice point accessed has the associated spin flip from up to down or vice versa. This generates the "trial" configuration for the Metropolis algorithm. In order to determine whether this trial state is likely to be part of the different permutations that would be possible, consideration is given to the total energy change between the two configurations. If the energy change is negative, the trial is accepted as the next configuration in the Markov Chain immediately. Otherwise, it is accepted with some probability which is weighted to the change in such a way that a larger positive change produces a lower probability of accepting the trial configuration. Should the trial configuration be rejected, the previous configuration is repeated in the chain. Once this process has gone on for a number of iterations, by considering the average energy and magnetisation it is possible to see that the expected behaviour will have, within error bounds, been observed from the system.

As is clearly evident, it is important to have the Metropolis algorithm be able to determine the validity of the trial configuration using some quantity or function that is related to the distribution of configurations or values that is of interest. In the next section, a relevant quantity for quantum mechanical MCMC processes known as the Classical Action, S, will be discussed in relation to Path Integrals.

It is not imperative, but it will please many readers to note that detailed balance for quantum systems is achieved through the Metropolis-Hastings Algorithm as is noted in both *User's guide to Monte Carlo methods for evaluating path integrals*[11] and *Monte Carlo Statistical Methods*[9], the latter being a potentially useful if expansive text for further understanding Monte Carlo methods in general.

2.4 Path Integrals

In quantum mechanics, if a system moves between the initial eigenstate $|q\rangle$ and the final state $|q'\rangle$ in a time T, it can be shown that the "propagator" that describes this path between initial and final state can be written as $\langle q'| e^{-iHT} |q\rangle$

In 1933, Paul Dirac suggested that the quantum mechanical propogator could be said to correspond to the expression $e^{\frac{iS}{\hbar}}$ in which S is the classical action. The classical action is defined as the path taken between two points and is given by the equation:

$$S = \int_{t_1}^{t_2} L dt \tag{10}$$

where L is the lagrangian of the system such that $L = T - V$ where T is the expression for the kinetic energy and V is the expression for the potential energy. This is classical action is required to be a bounded path in space and time for this consideration.

In 1948 Richard Feynman developed on this idea, and managed to prove that not just the classical path of the particle contributed to the overall propagator, but rather that every possible path contributed its own factor of $e^{\frac{iS}{\hbar}}$ to it. The suggestion was that a quantum particle travels all possible paths between the fixed intitial and final states, and further exploration of this idea resulted in the development of Path Integrals[4].

While a full and more involved derivation is not presented in this report, the reader is encouraged to delve into a series of lectures released on the preprint server *arXiv* entitled *Path Integral Methods and Applications*[8] which is where this rather brief derivation has been taken.

Starting from the expression given before,

$$\langle q'| e^{-iHT} |q\rangle \tag{11}$$

we define $\delta t = \frac{T}{N}$ which denotes a small time division of the total time between the initial and final states T of which there are N such divisions. This results in a factorisation such that

$$\langle q'| e^{-iH\delta t^N} |q\rangle \tag{12}$$

this can then be thought of as the product of a number of small segments of a total path between q and q' which can be thought of as q_0 and q_N respectively.

Now considering just one pair of intermediate points we can show that

$$K_{q_{j+1}q_j} = \langle q_{j+1}| e^{-iH\delta t} |q_j\rangle \tag{13}$$

which can be thought of as a small component of the total propagator. Thus the total propagator can be shown to be equivalent to

$$A = \sum_{paths} A_{path} \quad [8] \tag{14}$$

where

$$A_{path} = \prod_{j=0}^{N-1} K_{qj+1q_j} \tag{15}$$

and when $N \to \infty$, this may be thought of as

$$A = \int \prod_{j=0}^{N-1} dq_j A_{path} \tag{16}$$

One particularly interesting connection to make is that the formulation of the propagator in this manner shows us that it is possible to think of a path integral as a Markov chain in which each small step between the initial and final state is a "link" in that Markov chain as they are necessarily connected and correlated.

Returning to Equation 13, if we now consider a large N such that δt will be sufficiently small we can expand the exponential as a Taylor series. This, combined with considering a standard Hamiltonian ($H = \frac{p^2}{2m} + V(q)$), will result in the discovery that

$$K = \int Dq(t) e^{iS[q(t)]} \quad [8] \tag{17}$$

which can be interpreted as the amplitude of each path integral being an integral over the possible paths determined by their respective classical action. This expression bears a resemblance to the original thought proposed by Dirac, and in the derivation of Equation 17 $\hbar$ was approximated as 1 to improve readability of the calculations. As stated previously, a full derivation may be found in *Path Integral Methods and Applications*[8].

Of use in this project are the forms of the classical action for the 1D Quantum Harmonic Oscillator and the 1D Anharmonic Oscillator discussed previously.

$$S = \int_{t=0}^{T} dt \left(\frac{m}{2} \left(\frac{dx}{dt} \right)^2 + \frac{m\omega^2 x^2}{2} \right) \tag{18}$$

$$S = \int_{t=0}^{T} dt \left(\frac{m}{2} \left(\frac{dx}{dt} \right)^2 - \frac{m\omega^2 x^2}{2} + \frac{\gamma}{4} x^4 \right) \tag{19}$$

where in Equation 19 you may note that the constant C from Equation 9 has been set to be the same combination of mass and frequency seen in the QHO in Equation 3 but made negative to denote the "Double Well" potential used in this project.

2.5 Discrete vs. Continuous

When dealing with MCMC processes, it is necessary to formulate discrete forms of any equations being used. As a result, discrete forms of many equations carry with them inherent error when compared to the continuous forms, either due to some approximations used or the number of discrete steps used. The concept of splitting a time interval into N small time intervals that is utilised in the formulation of the path integral consideration of quantum mechanics is an example of a step towards a discrete consideration.

Expanding on this, it is possible to produce discrete forms of Equations 18, and 19.

$$S = \sum_{i=0}^{N} \left(\frac{m}{2}(x_{i+1} - x_i)^2 + \frac{m\omega^2 {x_i}^2}{2} \right) \tag{20}$$

$$S = \sum_{i=0}^{N} \left(\frac{m}{2}(x_{i+1} - x_i)^2 - \frac{m\omega^2 {x_i}^2}{2} + \frac{\gamma}{4}{x_i}^4 \right) \tag{21}$$

The idea is that as $N \to \infty$ (move towards the continuum limit), these discrete sums converge to the same result as the integral forms in Equations 18 and 19. These forms use forward integration to evaluate the momentum, and as such there will need to be boundary conditions on any set of $\{x_i\}$ that are used with these expressions. This will be discussed further in the following sections on implementation of this theory.

3 Method

The aim of this project is to develop a program that will compute path integrals over a lattice for a given potential, but to build this program in a manner that will allow another potential to be investigated. For the purposes of this project and its time scale, the final goal is be able to provide some results for the anharmonic oscillator. As was explained in the theory section on the anharmonic oscillator, it is a more challenging problem that does not offer exact solutions. The aim is to construct the program and test it on the Quantum Harmonic Oscillator to check that the method works and provides useful results which may be verified with those expected of the system. Once this verification is complete, an analysis of the anharmonic oscillator will attempt to provide good estimates of observables and the energy spectrum for that system.

The program was constructed in Python as this was the language most familiar to the author, and the limited time scale of the project did not give much room to learn another language to the capacity required. The full code can be found at a GitHub repository, the link for which is in Appendix A. Packages such as Numpy were used to improve performance compared to dealing with standard python lists and arrays, and matplotlib was used to generate the majority of the graphs seen in this report.

3.1 MCMC and Metropolis Algorithm

The process of setting up a Markov Chain for the oscillator potentials centres around the production of a number of paths of a particle in the potential being considered. These paths are represented by a 1D list of the particle's position at small time intervals.

A starting path is generated at the start, and successive iterations will apply random changes to each of the position values in the list in order to construct new paths via the Monte Carlo process combined with a Metropolis algorithm. The resulting paths are held in memory and may then be used to calculate the value of various observables.

Each iteration of the Monte Carlo - Metropolis process takes in the previously accepted configuration of the 1D list representing a particle's path. Each element in the list is accessed in turn and a random increase or decrease to this value is proposed based on a preset number, h. This value determines the acceptance rate of the Metropolis algorithm and consideration should be made based on the problem at hand which value is best. This number h is used to determine the bounds for the random increase or decrease applied to the value in the list, and once this trial value is constructed, the Metropolis algorithm will need to decide whether to accept the change or not.

The Metropolis algorithm checks against the condition of the system being considered, in the case of the oscillator potentials this is the classical action of the path. The difference between the classical action of the path before the change is compared to that of the path with the proposed change, and if the classical action is smaller with the change then the trial value is immediately accepted. Otherwise, there is a probability of acceptance based on the magnitude of the difference in classical action between the two paths. Should the value be rejected, the next element in the list is considered with the previous value being repeated in the list.

Once each element in the list has been accessed once, the configuration is saved as the next entry in the Markov Chain, regardless of whether any amendments have been made or not, and the next iteration begins the process all over again until the end condition is met. In the case of this project, the end condition was a large number of iterations.

3.2 Position Array and Dimensionless Parameters

The structure of the program centres around a 1D array of position values, $\{x_i\}$. The initial values can be an array of zeros signifying a "cold" start or a series of random numbers drawn from a distribution such as a gaussian to signify a "hot" start. The resulting initial path is then fed through a Monte Carlo process combined with a Metropolis algorithm to produce a series of new paths and set up a Markov Chain process. The resulting configurations are then used to provide the necessary data to produce ensemble averages of certain observables such a average position and higher moments, as well as estimate energy level differences and produce visualisations of the modulus squared amplitude of the ground state.

Due to the nature of computer simulations, it is necessary to ensure that units are taken into careful consideration. As such, quantities that have real physical meaning such as mass, m, and frequency, ω, are to be scaled so as to be dimensionless. The 1D lattice is considered to be a series of position measurements made at a certain time interval $\delta\tau$, with the number of position measurements present in the array being referred to as "Time Slices" from here on and denoted by N_τ. The time interval $\delta\tau$ is calculated as being $1/N_\tau$ and thus becomes smaller as the array becomes larger. This is analogous to considering a finer lattice of positions, with a cruder lattice being made up of fewer values and thus having a larger time interval between those values. The quantities such as mass and frequency are given in terms of this time interval or lattice spacing, with the following relations defining dimensionless quantities:

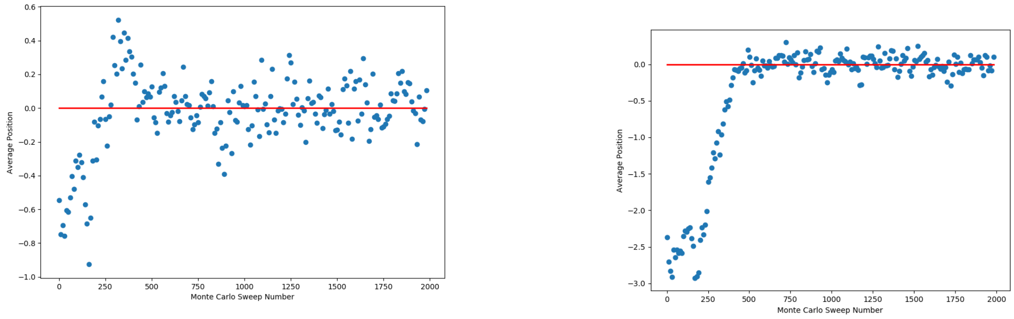
$$\tilde{m} = m\delta\tau, \quad \tilde{\omega} = \omega\delta\tau, \quad \{\tilde{x}_i\} = \{x_i\}/\delta\tau$$

This choice is motivated in *User's guide to Monte Carlo methods for evaluating path integrals*[11], including the reasoning for a further approximation that $\hbar = c = 1$ which results from the ability to scale these away. This results in the implication that units of length and time are equivalent to the inverse units of energy and mass. Thus the relations for the dimensionless variables above are satisfied.

There will be a certain amount of relaxation time present as the initial configuration converges towards the equilibrium state of the potential being investigated. As such, not all configurations generated by the MCMC process should be retained for analysis at the start of the iterations. The specific issue can be visualised well by considering an observable with a known expectation value such as the average position of a particle in the quantum harmonic oscillator. The expectation value of the average position over a discrete lattice is as follows:

$$\langle x \rangle = \frac{1}{N_\tau} \sum^{N_\tau} \tilde{x}_i \tag{22}$$

For the Harmonic Oscillator, this has a value of 0 similarly to most classical oscillators that have a symmetric potential such as a pendulum. In order to determine this for given initial path, the program will compute the entire Markov Chain, and then plot the value of the observables and display these to the user. The point at which the system has reached equilibrium is then judged by eye by the user, with the advice being that cutting too few values from the start of the chain is worse than cutting too many as with the latter at least they are still in the equilibrium state.



(a) Early configurations do not show behaviour consistent with harmonic oscillator, and as such the first 400 configurations ought not to be considered.

(b) A longer period of thermalisation can be seen, suggesting that the first 500 configurations should be ignored.

Figure 3: Plots of Average Position versus Metropolis Sweep Number for two different initial paths. The red horizontal line shows the position of the theoretically expected value of 0.

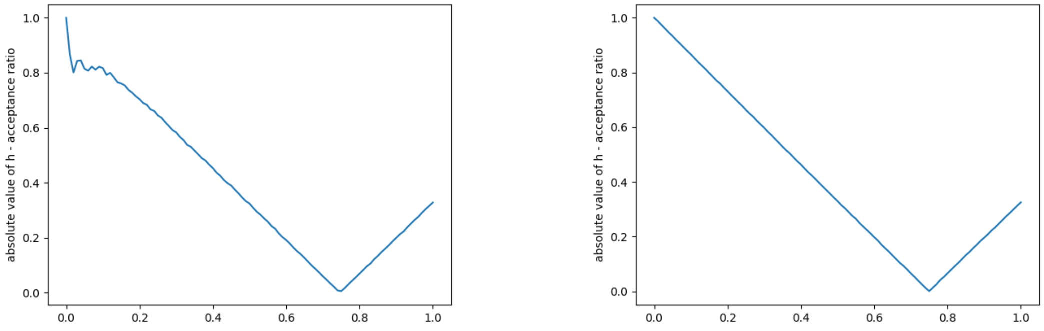
Figures 3a and 3b show results from two simulations using $N_\tau = 40$, $\tilde{m} = \tilde{\omega} = 1.25$, and run for 2000 Metropolis sweeps and calculating average position every 10 sweeps. It is clearly visible that the first 250 configurations do not exemplify the expected behaviour and a further 50-100 would be questionable. This arrangement shows that the time to properly thermalise the system needs to be taken into account when measuring the simulation. This is done by throwing away the configurations produced immediately after initialisation, in the case of Figure 3b it would likely be useful to leave out the first 400 sweeps of the array of position values despite this run using identical parameters bar the randomly generated initial values of the array.

Ultimately the time to thermalise the system will depend on the input parameters and the initially generated path and this is why the system generates the Markov Chain without considering the time to equilibriate and then the non-equilibrium values are removed.

3.3 MCMC and Metropolis

The Monte Carlo process utilised by the program will access each lattice site, and attempt to make a change to each value once per Metropolis sweep. The amount that each position value will attempt to be changed by is given by a uniform random number distribution, specifically for this program utilising the random number distributions present in the Numpy package. This distribution is sampled on the interval $[-h, h]$ where the value of h is fixed at the start of the simulation. This value is changed until the desired acceptance ratio is achieved. The acceptance ratio refers to the ideal ratio of acceptances and rejections made by the Metropolis algorithm.

A value of $h = 0.8$ was suggested in *User's guide to Monte Carlo methods for evaluating path integrals*[11], although it is noted in the paper that a smaller value should be used for the harmonic oscillator. It is said that ultimately the choice of this value does not affect the outcome of the simulation. However, a trial was conducted by considering a full spectrum of values of h against the acceptance ratio achieved during the running of the Metropolis algorithm for the same system parameters.



(a) Hot start

(b) Cold start

Figure 4: Plot of the difference between h and the acceptance ratio for each value of h, it can be seen that the cold start and hot start both agree that a value of 0.75 produces the least difference between desired acceptance and the value of h

As can be seen in Figure 4, a value of $h = 0.75$ appears to be optimal for this particular implementation, and this is in line with the discussion of the value of h in *User's guide to Monte Carlo methods for evaluating path integrals*[11].

Once the trial change has been calculated, the program calculates the change in the classical action that this change would produce, considering only nearest neighbours in the array (those before and after the value being changed) as this will significantly reduce the computations necessary. The full value of the classical action for any given array can still be calculated should it be required, with the discrete expression for the Harmonic Oscillator given in Equation 20. The expression for the nearest neighbour components of the classical action is thus:

$$\tilde{S}_{NN} = \sum_{i=n-1}^{n} \left(\frac{\tilde{m}(\tilde{x}_{i+1} - \tilde{x}_i)^2}{2} + \frac{\tilde{m}\tilde{\omega}^2 \tilde{x}_i^2}{2} \right) \tag{23}$$

Where in the above expression you will notice that we have used the dimensionless classical action for nearest neighbours, $\tilde{S}_{NN}$, due to the nature of the dimensionless parameters. The number n is the index of the position value being changed by the simulation. The value of this expression for the array not including the trial change is taken away from the value of $\tilde{S}$ including the trial change.

$$\delta\tilde{S} \equiv \delta\tilde{S}_{NN} = \tilde{S}_{NN,\text{trial}} - \tilde{S}_{NN,\text{original}} \tag{24}$$

where $\tilde{S}_{NN,\text{trial}}$ is the value given by Equation 23 when the trial value is considered, and $\tilde{S}_{NN,\text{original}}$ is the value of Equation 23 when the original state of the lattice is considered.

Should $\delta\tilde{S} < 0$ the trial change will be immediately accepted by the Metropolis algorithm. Otherwise it is accepted with probability $\min\{1, e^{-\delta\tilde{S}}\}$. If the trial is rejected, the old configuration repeated in the chain and the next trial is attempted. It should be noted that the array is programmed to include periodic boundary conditions such that if the next time slice to be accessed is $N_\tau + 1$ then it accesses the first element of the list and vice versa.

3.4 Correlation

By the nature of the process, the individual paths produced by each metropolis sweep will be heavily correlated with the paths produced immediately before them. This correlation causes issues when attempting to measure observables such as the previously discussed average position and other higher moments as the statistical processes utilised to complete the analysis require independent or uncorrelated data to be useful. This is remedied by making measurements of the current path are at certain intervals of Metropolis sweeps. This interval will vary based on the size of the lattice being considered and a study of its behaviour was undertaken for the Harmonic potential in order to ascertain a consistent ratio that might be employed to avoid the need to enter a unique value for every parameter set. Figure 5 shows the results for a lattice of size $N_\tau = 40$. From these rough results it was determined that while a separation of 10 Metropolis sweeps (and thus a relation of $N_{\text{sep}} = \frac{1}{4}N_\tau$) would be the optimal based on the values of the integrated autocorrelation time, there was the worry that this would lead to longer computation time in order to maintain the same number of results and so the value of 4, resulting in the relation $N_{\text{sep}} = \frac{1}{10}N_\tau$, was chosen instead.

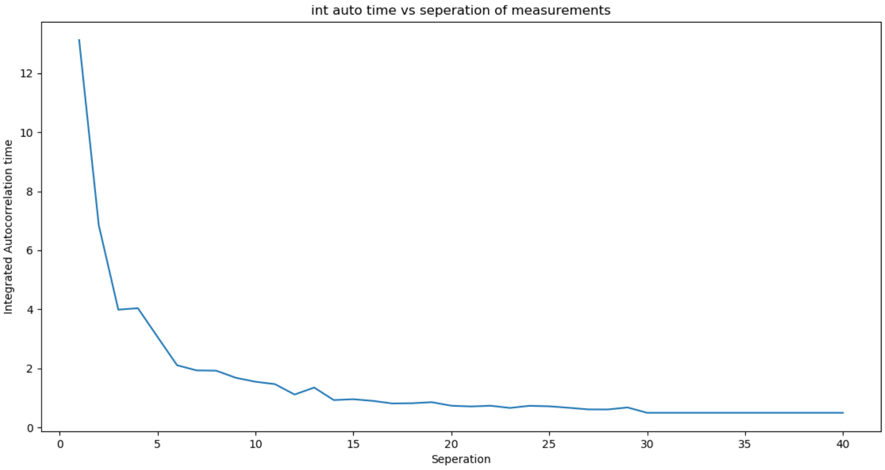


Figure 5: Plot of the integrated autocorrelation time of the average position observable against the separation of the measurements of the observable. Note the lack of any dramatic decrease past a separation of 10 Metropolis sweeps.

The integrated autocorrelation time is also a necessary quantity to calculate as it shows the difference between the sample variance and the ensemble variance. Note that the use of the word “time” in this context is purely convention as it is more accurately a number but gives an indication of the “time” between correlated entries in a chain. The sample variance is the variance about the average value for a given sample of data drawn from a distribution, while the ensemble variance is the variance of the ensemble average which is determined by combining many sample averages which have been computed from various samples from the same distribution. The relation between these two quantities is as follows:

$$\bar{\Omega}^2 = 2\tau_{O,\text{int}} \left\langle \Omega^2 \right\rangle \tag{25}$$

where $\tau_{O,\text{int}}$ is the integrated autocorrelation time seen in Equation 26. This result may be obtained from consideration of the derivation of the autocorrelation time the definition of the naive variance of a quantity[11].

The integrated autocorrelation time for a given observable O can be calculated using the following expression which relates it to the path position in the Markov Chain ν:

$$\tau_{O,\text{int}} = \frac{1}{2} + \sum_{\nu=1}^{N-1} \frac{A_O(\nu)}{A_O(0)} \quad [11] \tag{26}$$

Where N is the number of paths in the Markov Chain, and $A_O(\nu)$ has the form:

$$A_O(\nu) = \frac{1}{N-\nu-1} \sum_{i=1}^{N-\nu} [O_i - \langle O_i \rangle][O_{i+\nu} - \langle O_{i+\nu} \rangle] \quad [11] \tag{27}$$

where the averages $\langle O_i \rangle$ and $\langle O_{i+\nu} \rangle$ are the averages over the first $N-\nu$ paths and then the remaining paths respectively. The value of $A_O(0)$ is taken to be the normal variance of the observable, σ^2.

In order to mitigate this it is possible to use re-sampling methods such as bootstrap or jackknife analysis. As long as the integrated autocorrelation time is low, the deviation of the two variances will be small enough that this re-sampling will account for the correlation that is present. The program does still calculate the integrated autocorrrelation time of each observable in the analysis stage so that the errors produced by these re-sampling can be deemed reliable or not based on the value of the integrated autocorrelation time. These re-sampling methods will be discussed in the following section.

3.5 Observables and Errors

The observables that were calculated from the MCMC process for this project for the Harmonic Oscillator were as follows: $\langle x \rangle$ the average position, $\langle x^2 \rangle$ the average squared position, $E_1 - E_0$ the difference in the ground state energy and the first excited state energy, and the form of $\left|{\psi_0}^2\right|$ the ground state probability density function

The average position and squared position were calculated as ensemble averages over the Markov Chain configurations, with the value for each configuration calculated and then the average of these taken. The probability density function of the ground state may be found by producing a histogram of all of the position values present in every configuration in the Markov Chain. This works particularly well when stored in a numpy array that may be flattened in order to produce a single 1D list of all individual position values. The resulting errors on the histogram bins (which were taken to be of width $\Delta x = 0.1$ as trials with smaller bins produced less accurate results) are taken to be the standard Poisson errors that go as the inverse square root of the number of entries in each bin.

The energy difference between the ground state and the first excited state is calculated by considering the two point correlation function

$$G(\Delta\tau) = \langle x(\tau)x(\tau + \Delta\tau)\rangle - \langle x(\tau)\rangle \langle x(\tau + \Delta\tau)\rangle \tag{28}$$

where $\Delta\tau$ is a fixed time interval between two position values on the lattice. It is possible to consider ladder operators and arrive at an expression which may be used to determine the the energy difference.

$$\tilde{m}G(\Delta\tau) = e^{-(E_1 - E_0)\Delta\tau}G(0) \quad [11] \tag{29}$$

This expression allows the plotting of values of the two point correlation function if the natural logarithm is applied

$$ln(G(\Delta\tau)) = -(E_1 - E_0)\Delta\tau + ln\left(\frac{G(0)}{\tilde{m}}\right) \tag{30}$$

this expression allows a straight line plot of the values of the two point correlation function to provide the energy difference between the ground state and the first excited state as its gradient. A least squares fit of the results was completed in each case, with the errors being given by those of the least squares fit unless there were too few points available. This will be discussed in detail in the results section.

Errors on the average position, average squared position, integrated autocorrelation function, and the two point correlation function were produced using the re-sampling method of jackknife analysis. This is a standard statistical procedure that re-calculates the average of a set of values after removing one of the values used to calculate the original average and then determines the variance of these new values from the original to obtain an error. As was previously stated, these methods account for the correlation in the configurations in the Markov Chain. As will be discussed in the following section, there are more sophisticated methods for dealing with this correlation but due to the brevity and scope of this project they were not advised, nor likely possible to implement[3][11].

4 Results & Discussion

4.1 Harmonic Oscillator Potential

The Harmonic Oscillator potential was tested using a pre-programmed mass of 40, and natural frequency of 40. This would ensure that a lattice of $N_\tau = 40$ would give $\tilde{m} = \tilde{\omega} = 1$. Finer lattices were then considered, with one cruder lattice considered to showcase the behaviour of the system at higher dimensionless mass and frequency.

Table 1

N_τ	N_{sep}	$\tilde{m}$	$\tilde{\omega}$
20	2	2.0	2.0
40	4	1.0	1.0
50	5	0.8	0.8
60	6	$\frac{2}{3}$	$\frac{2}{3}$
80	8	0.5	0.5
100	10	0.4	0.4
120	12	$\frac{1}{3}$	$\frac{1}{3}$
200	20	0.2	0.2

Values of the observables listed in the previous section were calculated along with integrated autocorrelation times for the parameter sets seen in Table 1. These were compared with expected theoretical results in order to determine how effective the program was at reproducing the behaviour of the QHO. For $N_\tau = 100, 120, 200$ a cold start was also used as the cold start is near equilibrium, the hope being that this would allow more measurements from the system and potentially give a clearer picture.

In general, the average position was consistent with the theoretical expectation of 0 throughout the different parameter sets. The only values that lie largely outside the expected value are seen at very finest lattice at $N_\tau = 200$. The error on the hot start for this run is large, but does not account for the variance from the expected value. It is clear that at higher values of dimensionless mass and frequency the system is more accurate when calculating this observable. Figure 6 shows the results in full.

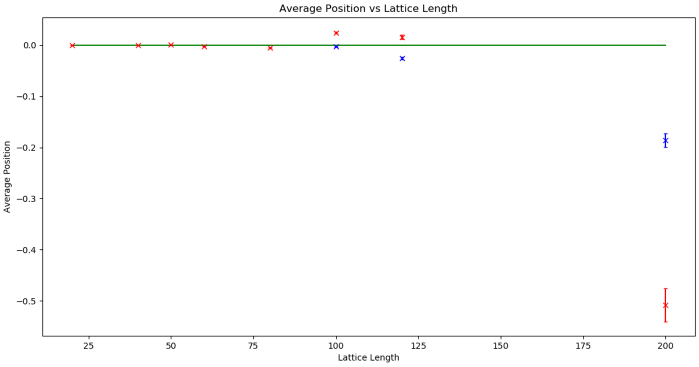


Figure 6: Plot of the observed values of average position throughout the parameter sets detailed in Table 1. The cold starts are denoted by blue markers, and the green line shows the expected value of $\langle x \rangle = 0$. Some error bars are smaller than the markers used to plot the points.

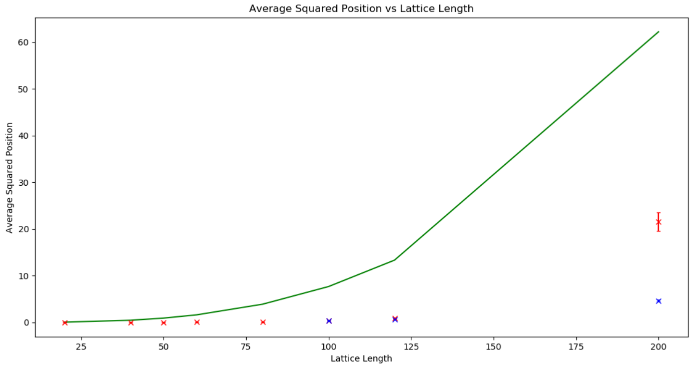


Figure 7: Plot of the average squared position for the various parameter sets detailed in Table 1. The cold starts are denoted by blue markers, and the green line shows the general trend of the expected values for these lattice lengths considered. Some markers cover the respective errors, and the cold and hot start results for $N_\tau = 100$ are so close they overlap.

The average squared position was less consistent with theoretical values, as can be seen in Figure 7. There appears to be an increase in the value at finer lattices, along with an increase in their respective errors, but it does not match the expected increase of the theoretical values. It would appear that the values from the program do not increase quickly enough to match this trend, and thus a correction on these values would need to be deduced if further analysis was to be carried out.

The ground state distributions produced by the program would follow a Gaussian form as long as the values had been taken after the Markov Chain had entered into equilibrium. However, the forms obtained were always at some reduced height compared to the form that would be expected in theory, with this difference in the general height of the two Gaussians seeming to increase with lattice length. This would suggest that the dimensionless mass and/or frequency are involved in a correction factor necessary to reach the theoretical curve.

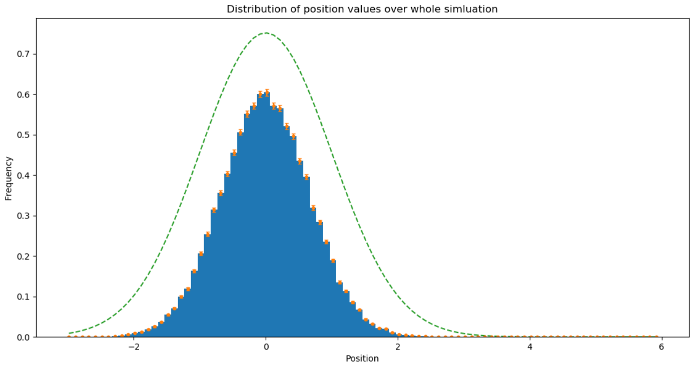
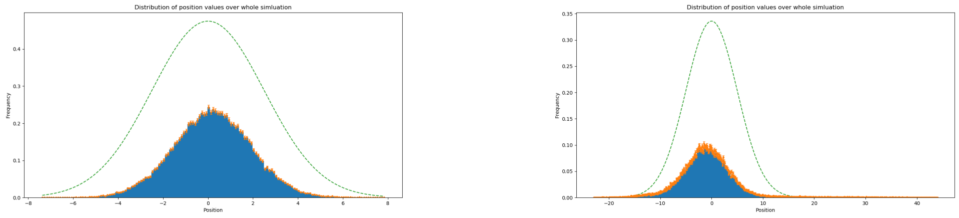


Figure 8: Histogram plot showing the probability density distribution of the ground state when using $N_\tau = 40$. The difference between the theoretical curve and the observed form is significantly larger than the errors on the histogram bins.



(a) Ground state distribution for $N_\tau = 100$ (b) Ground state distribution for $N_\tau = 200$

Figure 9: Histograms showing the correct general form of the ground state distribution, but the dramatic difference between the expected curve and the observed curve is clearly visible.

The disparity between theory and result can be most clearly seen in Figures 9a and 9b, as the results with finer lattices appear to deviate more drastically from the expected results.

The energy level differences were a mixed bag. One issue that became apparent quickly was that the correlation function would descend into noise quickly, resulting in few useful results to use for plotting the linear fit discussed in Section 3.5. This also resulted in the errors for some of these values, few that there were, being too large to warrant any analysis, with some having the same order of magnitude as the value. This is not entirely unexpected due to the small nature of these values, any error on them that is in any way significant will skew the results as a result. However, only the results for $N_\tau = 20$ and 60 were discounted, with the rest having sufficient data to allow removal of some particularly troublesome data points.

However, as can be seen in Figure 10, only the $N_\tau = 80$ result was within one error bar of the expected result. The rest are often just out of range of three error bars, or are only just in range of two or three error bars. Of particular interest, if only due to occurrence being a complete coincidence, taking the cold and hot start results for $N_\tau = 120$ which

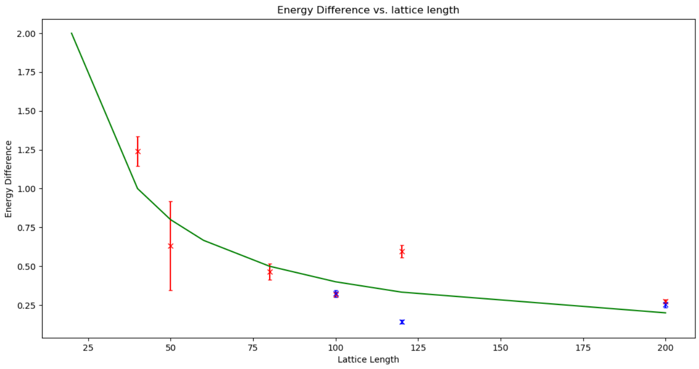


Figure 10: Results of the energy difference analysis vs lattice length used to obtain them. The green curve shows the expected behaviour over the range of lattice lengths, and the blue markers denote cold starts while the red denote hot starts. Some of the error bars are smaller than the points used to plot them.

are either side of the expected value and using these as maximum and minimum slopes, the result comes out to be close to that expected. However this analysis is not accurate and, as stated, may be complete coincidence.

Overall the results for the QHO are not very accurate for some observables, but do tend to follow the correct behaviour in certain ranges. The ground state distributions appear to be the most reliable for providing the correct shape and could be used to accurately describe the expected result if a correction factor could be determined.

4.2 Anharmonic Oscillator Potential

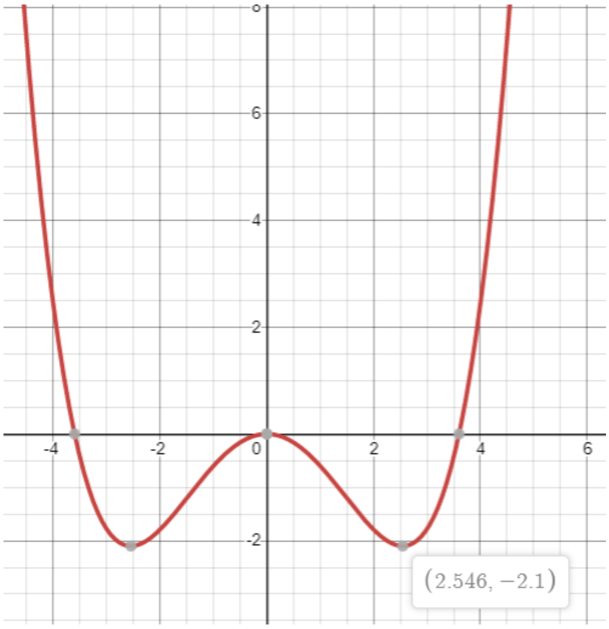


Figure 11: The anharmonic potential being investigated with C $= -\frac{5}{3}$ and $\gamma = 0.2$.

The Anharmonic Oscillator potential considered for this project was the double well potential which can be seen in Figure 11. The parameter sets used can be seen in Table 2 and are designed to ensure that the same double well is being considered so that the results may be checked against each-other at finer and finer lattices. One exception is $N_\tau = 40$ which has a slightly difference position of the two wells, but was chosen as it was challenging to find whole integer values to produce the same potential in the desired range.

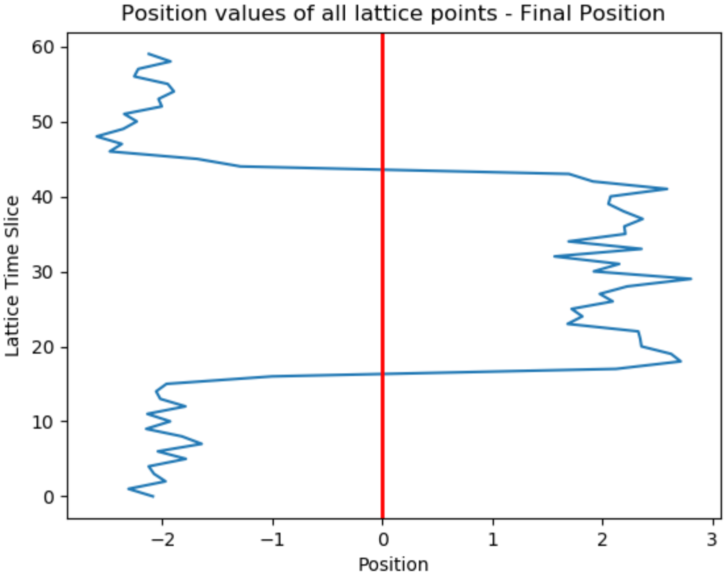


Figure 12: Example of the positon lattice when considering the double well potential. The two wells for this potential are at the symmetric positions noted in Figure 2

Table 2

N_τ	γ	$\tilde{m}$	$\tilde{\omega}$
30	0.2	$\frac{5}{3}$	$\frac{5}{3}$
40	0.2	1.75	1.75
60	0.2	$\frac{5}{3}$	$\frac{5}{3}$
120	0.2	$\frac{5}{3}$	$\frac{5}{3}$

As can be seen in Figure 12, the lattice positions themselves can show the presence of the two wells as the lattice splits between keeping points in one well over another. An animation is optionally produced by the program as the MCMC process progresses and shows the lattice points as a line. The figure shows the final position achieved, but the animation of the evolution of the positions in the lattice showed that one well was more favoured than the other. This can be seen in the ground state distributions which forms two peaks around the positions of the two wells, with a higher density in the well which more points found themselves. This could be seen as a particle spending more time in one well before tunneling to the next.

An example of this can be seen in Figure 13, where one peak can be clearly seen to be the more dense of the two despite the symmetric potential. This occurs due to

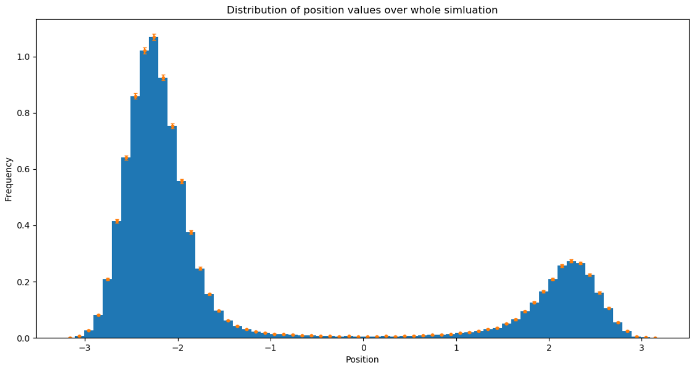


Figure 13: Ground state probability density distribution of the anharmonic double well potential. The two peaks can be seen to exist in the area of the two wells, although slightly closer to the origin due to the "particles" tending to oscillate closer to the origin than the minima of the wells. Particles spend more time in one well than the other, although which well this affects is random.

the random nature of the changes under the MCMC process placing a larger majority of the lattice closer to one well, and then the split is not even. Even using a cold start to position every lattice point at zero does not guarantee an even split between the wells as the random changes due to the Monte Carlo process will not necessarily graduate an even umber of points either side of the "bump" in the middle. It was observed that the opposite was true, that a cold start would not show the tunnelling even at this parameter set which was determined to be the optimum for experiencing this behaviour. A cold start would instead oscillate around the origin for some time and then the entire lattice would begin to slip into one of the wells and stay there for the duration of the remaining iterations. Very occasionally some of the lattice points would creep close to the origin, but even if they crossed the origin they would be dragged back into one of the wells. This behaviour became more common at larger lattice lengths and could potentially provide insight into the position of a particle in one of these potentials, although in a symmetric well you would suspect that a particle would wish to tunnel if possible to maintain itself in both minima equally.

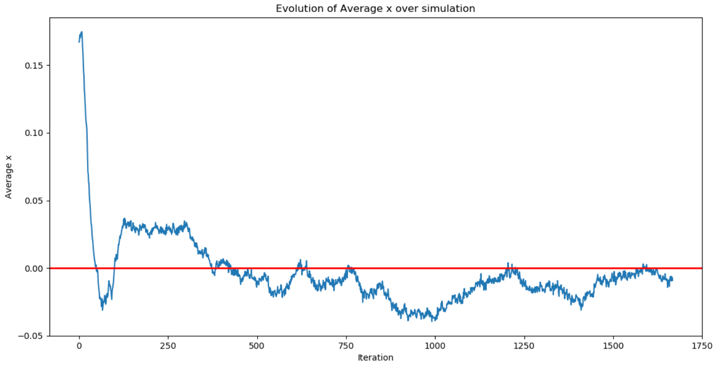


Figure 14: Evolution of average position over the course of the MCMC process for the anharmonic potential. The chnages in how the lattice points were split over the "bump" in the middle may be clearly seen as points where the average position crosses the origin.

As has been stated before, the splitting over the "bump" in the middle of the potential changes over the generation of the Markov Chain, with this most clearly being visualised by plotting the average position. Figure 14 shows this for the case of $N_\tau = 60$ and gives good reasoning as to why the ground state distribution in Figure 13 has a bias towards one of the wells.

Overall it was observed that the average position would be 0 in the case of the double well potential, with the deviations seen from this value tending to be explained by the MCMC process favouring one of the wells over the other later into the simulation. Average squared position was seen to increase with lattice length, but due to the issues regarding the results from the QHO, it was decided that due to time constraints this would not be investigated further.

The energy difference between the first excited state and the ground state was an interesting quantity to examine. As we considered the same potential but under finer and finer lattices, it was possible to see compare the results with each other to see if a common value may be obtained. As a result of time constraints, only the parameters in Table 2 could be investigated which does limit the confidence in this approach. However, since the issue with the correlation function descending into noise quickly did not appear to persist with this potential, it was possible to get a much larger pool of values for the linear regression leading to overall more accurate results. These can be seen in Table 3.

Table 3

N_τ	$E_1 - E_0$
30	0.173(30)
40	0.196(18)
60	0.143(10)
120	0.142(10)

By considering only those points that had the lowest errors, the results seem to show that a value of around 0.14 would be consistent for this parameter set. This occurred at a values of $\tilde{m} = \tilde{\omega} = \frac{5}{3}$ and so there may potentially be some link to these values and potentially a clue to the general expression for this energy change in terms of them.

4.3 Improvements and Future Work

Potential improvements that could be made if this project were to be repeated would be to employ variable bin-size jackknife analysis which would potentially reduce some of the large errors seen in the QHO potential. As a result, better confidence in the values output from the program may be possible and thus better deductions of results from other potentials may be made.

An investigation into the potential correction factors for the ground state and the average squared position would be recommended, as this may also allow better analysis of the anharmonic potential.

Some method of measuring the tunnelling time in the double well potential would be an interesting component to add to a future study of this kind, and this system may also be able to show the critical point at which this tunnelling time goes to infinity (the point at which tunnelling becomes so improbable that it would be essentially impossible).

For both potentials, further moments of position could be investigated, with some analytical expressions existing for the QHO [11].

The issues with the two point correlation function descending too quickly ought to be investigated in order to obtain more values to apply to the linear regression. The method could also be amended to use higher order correlation functions to investigate higher energy differences.

Finally, another potential that would be interesting to investigate with this system would be the inverted double well. This potential would require some extra checks and balances to ensure that the lattice does not go towards positive or negative infinity should it escape the potential but this is rather trivial.

5 Conclusion

A program that uses a MCMC process that may be used to investigate various quantum mechanical potentials by considering the methods of path integrals has been produced. Limited success has been achieved in observing the behaviour of the QHO, but with further work this program could accurately observe the analytically expected behaviour. The anharmonic double well potential was investigated using this system and a potential value for the first level energy difference has been tentatively deduced, although heavy scepticism is advised due to the issues with the program's treatment of the QHO. Splitting of the lattice and tunnelling have been observed in the case of this double well potential, with further work recommended to begin analysing the tunnelling time.

6 References

References

[1] Michael Creutz and B.A. Freedman. "A statistical approach to quantum mechanics* 1". In: *Annals of Physics* 132 (Apr. 1981), pp. 427–462. DOI: 10.1016/0003-4916(81)90074-9.

[2] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), pp. 416–418. DOI: 10.1017/S0305004100021162.

[3] B. Efron and C. Stein. "The Jackknife Estimate of Variance". In: *Ann. Statist.* 9.3 (May 1981), pp. 586–596. DOI: 10.1214/aos/1176345462. URL: https://doi.org/10.1214/aos/1176345462.

[4] R. P. Feynman. "Space-Time Approach to Non-Relativistic Quantum Mechanics". In: *Rev. Mod. Phys.* 20 (2 Apr. 1948), pp. 367–387. DOI: 10.1103/RevModPhys.20.367. URL: https://link.aps.org/doi/10.1103/RevModPhys.20.367.

[5] Eldad Gildener and Adrian Patrascioiu. "Pseudoparticle contributions to the energy spectrum of a one-dimensional system". In: *Phys. Rev. D* 16 (2 July 1977), pp. 423–430. DOI: 10.1103/PhysRevD.16.423. URL: https://link.aps.org/doi/10.1103/PhysRevD.16.423.

[6] David P. Landau and Kurt Binder. *A Guide to Monte Carlo Simulations in Statistical Physics (Third Edition)*. Cambridge University Press, 2009. ISBN: 978-0-521-76848-1.

[7] Jiu-Qing Liang and H. J. W. Müller-Kirsten. "Periodic instantons and quantum-mechanical tunneling at high energy". In: *Phys. Rev. D* 46 (10 Nov. 1992), pp. 4685–4690. DOI: 10.1103/PhysRevD.46.4685. URL: https://link.aps.org/doi/10.1103/PhysRevD.46.4685.

[8] Richard MacKenzie. *Path Integral Methods and Applications*. 2000. arXiv: quant-ph/0004090 [quant-ph].

[9] Christian Roberts and George Casella. *Monte Carlo Statistical Methods*. Springer New York, 2004. ISBN: 978-1-4419-1939-7.

[10] Alexander E. Sitnitsky. "Analytic calculation of ground state splitting in symmetric double well potential". In: 2018.

[11] Marise J. E. Westbroek et al. "User's guide to Monte Carlo methods for evaluating path integrals". In: *American Journal of Physics* 86.4 (Apr. 2018), pp. 293–304. ISSN: 1943-2909. DOI: 10.1119/1.5024926. URL: http://dx.doi.org/10.1119/1.5024926.

A Python Code

The code used for this project is publicly available in a GitHub repository at the URL: https://github.com/jcclarke290/shp_MCMC

The following small programs were used to generate some of the graphs found in this report but were not imperative to the completion of the project's main goals.

```
import numpy as np
import math as m
import matplotlib.pyplot as plt

x_h = np.loadtxt("averageX.txt")
x_s = x_h[0]
x_s_err = x_h[1]

x_cold = np.loadtxt("averageX-cold.txt")
x_s_c = x_cold[0]
x_s_c_err = x_cold[1]

x_2_h = np.loadtxt("averageX_2.txt")
x_2_s = x_2_h[0]
x_2_s_err = x_2_h[1]
x_2_s_exp = x_2_h[2]

x_2_cold = np.loadtxt("averageX_2-cold.txt")
x_2_s_c = x_2_cold[0]
x_2_s_c_err = x_2_cold[1]
x_2_s_c_exp = x_2_cold[2]

E_h = np.loadtxt("energyDifference.txt")
E_s_h = E_h[0]
E_s_h_err = E_h[1]
E_s_h_exp = E_h[2]

E_cold = np.loadtxt("energyDifference-cold.txt")
E_s_c = E_cold[0]
E_s_c_err = E_cold[1]
E_s_c_exp = E_cold[2]


lattices = np.asarray([20,40,50,60,80,100,120,200])

plt.plot(lattices, [0,0,0,0,0,0,0,0], 'g-')
plt.errorbar(lattices, x_s, yerr=x_s_err, capsize=2, fmt='rx')
plt.errorbar(lattices[5:], x_s_c, yerr=x_s_c_err, capsize=2, fmt='bx')
plt.title("Average Position vs Lattice Length")
plt.xlabel("Lattice Length")
plt.ylabel("Average Position")
plt.show()

plt.plot(lattices, x_2_s_exp, 'g-')
plt.errorbar(lattices, x_2_s, yerr=x_2_s_err, capsize=2, fmt='rx')
plt.errorbar(lattices[5:], x_2_s_c, yerr=x_2_s_c_err, capsize=2, fmt='bx')
plt.title("Average Squared Position vs Lattice Length")
plt.xlabel("Lattice Length")
plt.ylabel("Average Squared Position")
plt.show()

plt.plot(lattices, E_s_h_exp, 'g-')
plt.errorbar(lattices, E_s_h, yerr=E_s_h_err, capsize=2, fmt='rx')
plt.errorbar(lattices[5:], E_s_c, yerr=E_s_c_err, capsize=2, fmt='bx')
plt.title("Energy Difference vs. lattice length")
plt.xlabel("Lattice Length")
plt.ylabel("Energy Difference")
plt.show()
```