Double Well Potential with MCMC

Bhavya Gupta,* Tiffany Liou,† Nicole Liu,‡ and Brigette Vazquez-Segovia§
University of California San Diego
(Group B)
(Dated: March 22, 2024)

Harmonic oscillator potentials have been foundational in the field of quantum physics. The solutions to these problems are typically found using analytical methods to solve the time-independent Schrodinger wave equation. In our project, we aim to find solutions to the ground state wave function and energy value using computational techniques. We employ Markov Chain Monte Carlo (MCMC) simulations to derive the probability distribution for the ground state wavefunction and its associated energy. We specifically apply the Metropolis-Hastings algorithm to help build our Markov Chain by accepting and rejecting new steps to converge our distribution to the desired value. To verify our results, we compared the output to the power method, based on analytic techniques to solve for energy eigenvalues and wavefunctions. Next, we also decreased our imaginary time values (τ_b) to further investigate the statistical mechanics properties of the ground state wavefunctions for the double potential well. We were able to match our MCMC distribution to the power method by using the cold start approximation where the energy value achieved is $\sim$1.27 eV, which is close to the 1.23 eV [1] determined by the analytic method. Reducing τ_b values results in an uneven probability distribution for the ground state due to its association with higher temperatures.

I. INTRODUCTION

Single and double-well harmonic oscillators are some of the most common potentials studied in the field of physics. Generally, solutions for the energy eigenvalues and the wave functions for the two are found using analytic solutions of the time-independent Schrodinger equation or through ladder operators to obtain the energy. We will instead try to find these solutions through the use of computer and probability algorithms, namely the Markov Chain Monte Carlo method.

Markov Chain Monte Carlo (MCMC) samples from a continuous random variable where the probability distribution is proportional to the known distribution. A series of chains is formed where it begins with arbitrary points. The next set of values is chosen randomly based on higher probability. After a large number of sweeps, the MCMC algorithm converges to the desired value.

In our project, we specifically implemented the Metropolis-Hastings algorithm for MCMC. In this method, the Markov chains are generated using the proposed density functions for the new steps. There is also a method for rejecting some of the moves generated. This is intrinsic to the algorithm for it to converge to the desired value/distribution.

There are three main parts to this project which we explore. First, we evaluate the particle's ground state energy and probability distribution using MCMC with the Metropolis-Hastings algorithm, in the realm of a large upper τ_b limit. Next, we compare the probability distribution from the MCMC method to the expected analytical form [1]. Lastly, we calculate the energy and probability distribution of the double-well from the same simulation code for a smaller value of τ_b and demonstrate that the expected ground state energy will increase. In a statistical mechanics interpretation, this corresponds with a higher temperature, implying some initial kinetic energy.

* bhgupta@ucsd.edu
† tiliou@ucsd.edu
‡ n5liu@ucsd.edu
§ blvazque@ucsd.edu

II. METHODS

A. Setup

We begin by considering the potential equation for a double well.

$$V(x) = \alpha x^4 - 2x^2 + \frac{1}{\alpha} \tag{1}$$

Here α is chosen to be 0.4 and x represents the position of the particle. We also define $m = \hbar = 1$ in order to work in energy units of eV.

For our path integral, we use the following setup:

$$Z = \int \mathcal{D}x(\tau) \exp\left[-\frac{1}{\hbar}\oint_0^{\tau_b} L_E(x(\tau))d\tau\right] \tag{2}$$

and

$$Z = \lim_{\delta\tau\to 0} \int\cdots\int dx_0\cdots dx_{N-1} \left(\frac{2\pi\hbar\delta\tau}{m}\right)^{-\frac{N}{2}} \times \exp\left[-\frac{1}{\hbar}\left(\sum_{i=1}^{N}\frac{m(x_i - x_{i-1})^2}{2\delta\tau} + \delta\tau V\left(\frac{x_{i-1}+x_i}{2}\right)\right)\right] \tag{3}$$

where $L_E(x(\tau)) = \frac{m}{2}\left(\frac{dx}{d\tau}\right)^2 + V(x(\tau))$ is the Euclidean Lagrangian and imaginary time (τ) is discretized with N increments, with $\tau_a = 0$ and $\tau_b = N\delta\tau = \hbar\beta$. The probability of a given path $(x_0, x_1, \ldots, x_{N-1})$ is

$$p(x_0, \ldots, x_{N-1}) \propto \exp\left[-\frac{1}{\hbar}\left(\sum_{i=1}^{N} \frac{m(x_i - x_{i-1})^2}{2\eta} + \eta V\left(\frac{x_{i-1} + x_i}{2}\right)\right)\right] \quad (4)$$

Our constants and parameters are as given in Table II B and Table III A. Constants were given by the limits of our problem, and parameters were chosen by looking at the temperature limits as well as general testing of the MCMC algorithm.

B. Implementation

We defined five functions for this implementation of the MCMC algorithm [2]. Our first function is `potential`, which takes in the point x along the path and returns the double-well potential at that point. Our second function is `vary_path`, which takes in the current point in the x-domain and the hitsize. It returns a new randomly picked point within our hitsize range by adding a random value found using $np.random.random() \times 2 \times hitsize - hitsize$. Our third function is `action`, adding up kinetic and potential energy at some point. It takes in the left limit of the action potential, the right limit of the action potential, and the step size in τ. Kinetic and potential energy are defined inside this function as follows; Kinetic $= \frac{1}{2} \times \frac{((x_{\text{right}} - x_{\text{left}})^2)}{d\tau}$ and Potential $= d\tau \times$ `potential` $\left(\frac{x_{\text{left}} + x_{\text{right}}}{2}\right)$. Where `potential` is the function defined above. Our fourth function is `delta_action`. This function helps us compare the action across a new proposed point, the method by which the MCMC Metropolis-Hastings algorithm operates, and the action across the old point. It takes in the original path, the proposed new point, the index of the old point, and the step size in τ. It returns the difference in the action across the original point and the action across the proposed new point. This difference is then used in our `mcmc` function when accepting or rejecting the new proposed point.

Our final and most important function was our `mcmc` function. This function's main goal was to find the ground state energy, which is proportional to the average energy over the number of steps, as well as the probability of such an occurrence. This function takes in the path across which to apply the MCMC algorithm, an array to save energy values, an array to save probability values, the running total of accepted values, the total imaginary time range across which the algorithm would run, the size of each time step, the hitsize from which to generate new values, and a thinning value in the case that we needed to thin our setup. This function then returns the number of accepted values, to be used again in the next iteration of the MCMC algorithm. We run this MCMC function for a set number of sweeps. With increasing iterations, we expect our average energy to converge to the expected value for the ground state energy for our setup, around 1.23 eV [1].

Constant	Value
α	0.4
x_{low}	-4
x_{high}	4
δx	0.08
n_{xbins}	100

TABLE I. Set constants used in our path integral and problem set up.

Additional parameters such as τ_b, dτ, hitsize, and number of sweeps still need to be determined. Another thing to consider is the path configuration that we start with. Our MCMC simulation can start with either a cold start, pre-determined values, or a hot start, consisting of random values within the physical limits of the problem. Varying these parameters one at a time will help us determine how they affect the problem.

III. RESULTS

The first parameter we can narrow down is τ_b. Table III A demonstrates how we chose a value of τ_b using the relationship of $\tau = \hbar\beta$, and $\beta = \frac{1}{k_b T}$, where k_b is Boltzmann's constant. We expect very little kinetic energy at ground state, corresponding with a very low temperature and a relatively large τ. We start with $\tau_b = 2000$.

A. High τ_b limit

For a high τ_b limit of $\tau_b = 2000$, we expect that the majority of particles will reside in the minima of the double potential well. Thus, the probability should ideally peak at the location of the minima $\left(\pm\sqrt{\frac{1}{\alpha}})\right)$. Figure 1 shows the MCMC energy calculations of the cold and hot start models.

We chose our burn-in steps to be after convergence. The cold start converged quickly, so we chose our burn-in to be 100 over a total of 2000 sweeps. This quick convergence can be attributed to a good initial guess of where the particles reside $\left(\pm\sqrt{\frac{1}{\alpha}})\right)$ and where we expect ground state particles to be. Thus, with each random walk, the probability of the particles at the well is higher. On the other hand, the hot start takes around 15000 to 20000 sweeps to fully converge. Thus, we chose the hot

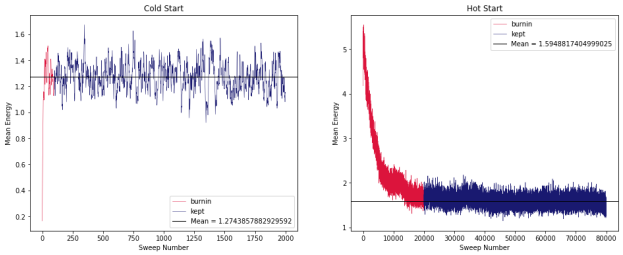

FIG. 1. MCMC simulation (cold vs. hot start) for a high τ_b value of 2000.

burn-in to be 20000 over a total of 80000 sweeps. This model took more iterations to converge due to the randomized initial locations of the particles. In a statistical mechanics sense, the randomization may indicate additional kinetic energies and complicate the accept-reject algorithms. This behavior can be seen in the MCMC energy estimations before convergence. In the end, the cold start gave an energy estimate of $E_0 = 1.27439$ eV while the hot start estimated 1.59488 eV. The cold start is closer to the true value of $E_0 = 1.23447$ eV [1].

The results of the particles' probability distribution throughout the double potential well are in Figure 2. From both the inferred results and the probability distribution, we see that the cold start model performs significantly better than the hot start. The hot start takes longer sweeps to converge and also over-estimates the ground state energy. This is reflected in the skewed probability distribution for the hot start model. Moreover, the probability for the cold start peaks at $\left(\pm\sqrt{\frac{1}{\alpha}}\right)$ and is almost zero everywhere, just as expected.

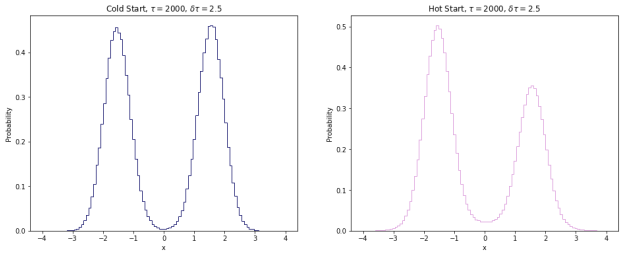

FIG. 2. Probability histograms (cold vs. hot start) for MCMC simulation with $\tau_b = 2000$.

We can also compare our double-well probability to the probability distribution of an anharmonic oscillator made by others in Figure 3 [3], where we see similar results. The probability is zero everywhere except for at the well minima. It is also important to note that as we increase τ_b, our probability distribution becomes more accurate and aligns more closely with Figure 3.

However, we chose to not run our MCMC simulation with larger τ_b values to keep a reasonable runtime. Similar reasoning is used for $\mathrm{d}\tau$. A smaller $\mathrm{d}\tau$ relative to τ_b means more time steps and a longer run time. The chosen value for hitsize also follows related logic. A larger hitsize corresponds with a higher accepted number of values, meaning the algorithm takes "larger" steps toward the supposed true value as the number of iterations increases. At the same time, we do not want the acceptance rate to be too high or we risk overshooting and missing the true value. We also don't want the hitsize and acceptance value to be too low, or else our algorithm takes a very long time to converge. With these in mind, our chosen parameters are listed in Table III A. We also found that setting a thinning value made little impact on our computation time in comparison to the rest of our parameters. Thus we can ignore it.

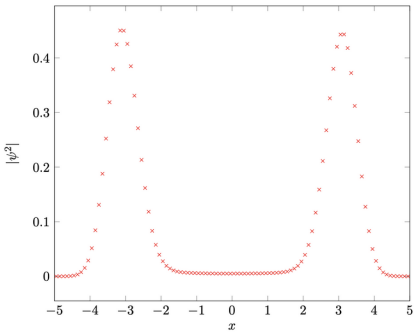

FIG. 3. Probability histogram from Rodgers and Raes 2014 [3]

Parameter	Value
τ_b	2000
$\delta\tau$	2.5
η	2.5
N_τ	800
Hit size	0.8

TABLE II. Parameters and chosen values used in the MCMC implementation.

B. Comparison to analytic results

We now compare our results with analytic methods. Using the power method, we find the ground state energy E_0 and the corresponding eigenvector ϕ_0. This represents the wavefunction of the particle's ground state. To calculate the analytic probability, we take $\psi \cdot \psi$. Figure 4 shows the comparison of the analytic probability versus the MCMC probability.

The analytic probability is also evenly distributed between the two well minima, but there are higher probabilities of particles being at other points in the well, unlike our MCMC guess. Looking by eye, we see that the cold start model still performs better and aligns well with the analytic probability. The probability misalignment in between the peak may be due the way that the

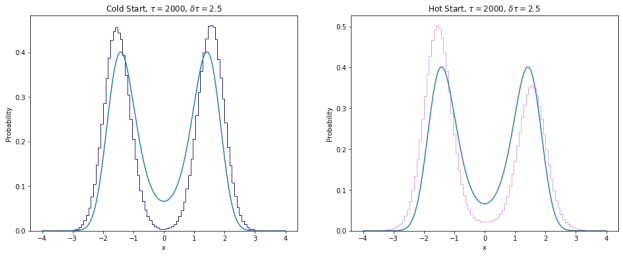

FIG. 4. Comparison of analytic probabilities and simulated probabilities for a cold start and a hot start.

power method is calculated. The power method may account for more of the kinetic energy, whereas we assumed kinetic energy is very small, especially for our cold start model.

C. Low τ_b limit

For our low τ_b limit, we chose $\tau_b = 50$. The stark contrast between $\tau_b = 2000$ and $\tau_b = 50$ is to see the difference in behavior for the energy inference and also the probability distribution 5 6. Physically, a low imaginary time corresponds to a higher temperature. This comes from how $\tau_b = \hbar * \beta$. If the temperature is higher, then there is a more significant amount of kinetic energy in the system, as KE $= \frac{3}{2} k_b T$. Higher kinetic energy implies that there are particles in a potential well that are excited and move to higher energy levels. We then expect the energy guess to be higher than $E_0 = 1.23447$ and that the probability distribution is skewed.

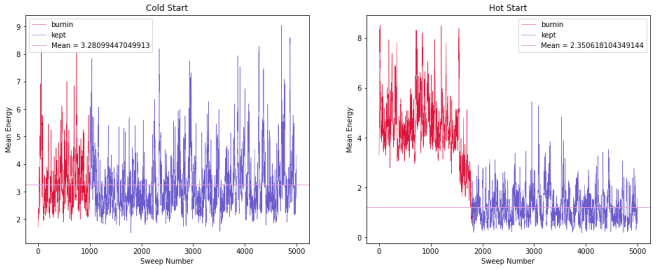

FIG. 5. MCMC simulation for $\tau_b = 50$ with a cold start and a hot start.

Figure 5 shows the energy guesses from both the cold and hot start models and Figure 6 shows the probability distribution of the particles' positions in the well. We can see that there is a skewed and wider spread distribution, as expected of a higher energy state. Moreover, this skew in probability can also indicate the tunneling of particles through the potential barrier. Note that the difference between the two probability distributions is because the hot start takes a different number of sweeps to fully converge. For simple comparison, we chose to stick with a sweep number of 5000 to keep runtime fast.

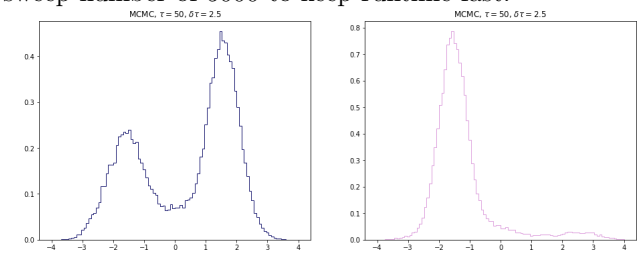

FIG. 6. Probability distributions for $\tau_b = 50$ for a cold start and a hot start.

IV. CONCLUSION

In this project we focused on three main things, the implementation of MCMC and the Metropolis-Hastings algorithm for a large value of an imaginary time τ, comparing our result to an analytical power method probability distribution for a particle in a double-well potential, and analyzing the lower τ limit. We were able to successfully implement the MCMC algorithm and converge both cold and hot starts for a particle. Our preferred value of τ_b was 2000. For our burn-in steps, we chose the values of 100 sweeps for the cold start and 15000 for the hot start. The cold start required a total of 20000 sweeps compared to the hot start which needed 80000. The cold start gave an energy estimation of $E_0 = 1.27439eV$ and the hot start gave an estimate of $E_0 = 1.5949eV$. This led us to prefer the cold start model, as it also more accurately mimics our analytic solution found in Figure 4. When the value for τ_b is decreased by a significant amount, in this case from 2000 to 50, we recognize that the probability distribution for the particle in the cold start prefers one side over the other, however not as drastically as in the hot start. The hot start simulation highly prefers the left side and assumes that almost all the particles lie in one well in this case.

For future work, we would like to continue further testing for different values for our parameters, implement other analytical methods for comparison, and consider using an initial higher temperature setup to also consider non-negligible kinetic energy.

V. CONTRIBUTIONS

For this project, all of us contributed in the development of the code, and the final edits of the code were made by Tiffany Liou and Nicole Liu. Bhavya Gupta and Brigette Vazquez worked on testing different parameters in the MCMC function. We would like to acknowledge Professor Duarte for helping us understand the fundamental setup for the MCMC algorithm and all others who provided helpful commentary.

[1] K. Banerjee and S. P. Bhatnagar, Two-well oscillator, Phys. Rev. D **18**, 4767 (1978).
[2] B. Gupta, T. Liou, N. Liu, and B. Vazquez-Segovia, UCSD PHYS 142 Double-Well Potential Group B (2024).
[3] R. Rodgers and L. Raes, Monte carlo simulations of harmonic and anharmonic oscillators in discrete euclidean time https://www-zeuthen.desy.de/students/2014/reports/RodgersRaes.pdf (2014).