Markov Chain Monte Carlo

Nandita Sudha Tiwari
John Batarekh

Dec 1 2023

1 Statement of the Problem

For our simulation, we have come up with an interesting model (not necessarily physical) that we would like to examine. We are going to imagine that we have three particles that are in an anharmonic potential such that the three particles exist in a superposition of their ground state and their first excited state, but no other higher energy levels. We want these particles to be coupled to each other as follows:

$$|\psi\rangle = e^{-\sum_i J_0^{(i)}|\langle 1|\phi_i\rangle|^2}|0\rangle + e^{-\sum_i J_1^{(i)}|\langle 0|\phi_i\rangle|^2}|1\rangle \tag{1}$$

Here, i represents the other particles, J_n^i is the "energy" corresponding to the particle coupling, and the n index is there to acknowledge that energy coupling between ψ and particle in the ground state is different from the coupling energy between ψ and a particle in the first excited state.

In our simulation, we will be examining the likelihood distribution of one of the particles being found in the ground state depending on the likelihood of the other particles also in the ground state. To do this we will take pseudo-measurements of the other particles. To simplify this we will imagine that the particle we are examining is coupled to two other particles but the other two particles are not coupled. So we will simulate that for any state that the particle we are examining is in, the other particles exist in an unknown superposition of the two lowest energy states, and hence $|\langle 0|\phi\rangle|$ is a uniform random number between -1 and 1 for each of the other particles. Therefore, our posterior function will be

$$|\langle 0|\psi\rangle|^2 = \frac{e^{-2(J_0^{(x)}(1-x^2)+J_0^{(y)}(1-y^2))}}{e^{-2(J_0^{(x)}(1-x^2)+J_0^{(y)}(1-y^2))} + e^{-2(J_1^{(x)}x^2+J_1^{(y)}y^2)}} \tag{2}$$

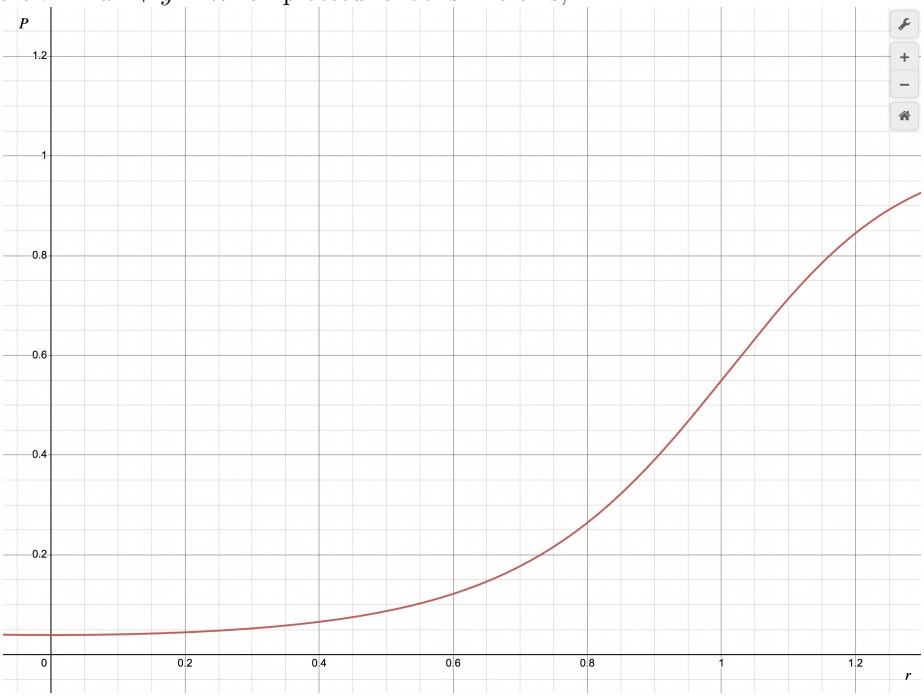
Our Markov chain will work as follows, we take a random step in the x,y plane such that those x and y are still in the bounds $[-1, 1]$. Then we calculate the posterior function using eq.(2). Next, we generate a uniform random number. If that number is less than the posterior function we store the values of x and y as well as the probability. The stored values of x and y become the chain.

2 Initial Expectations

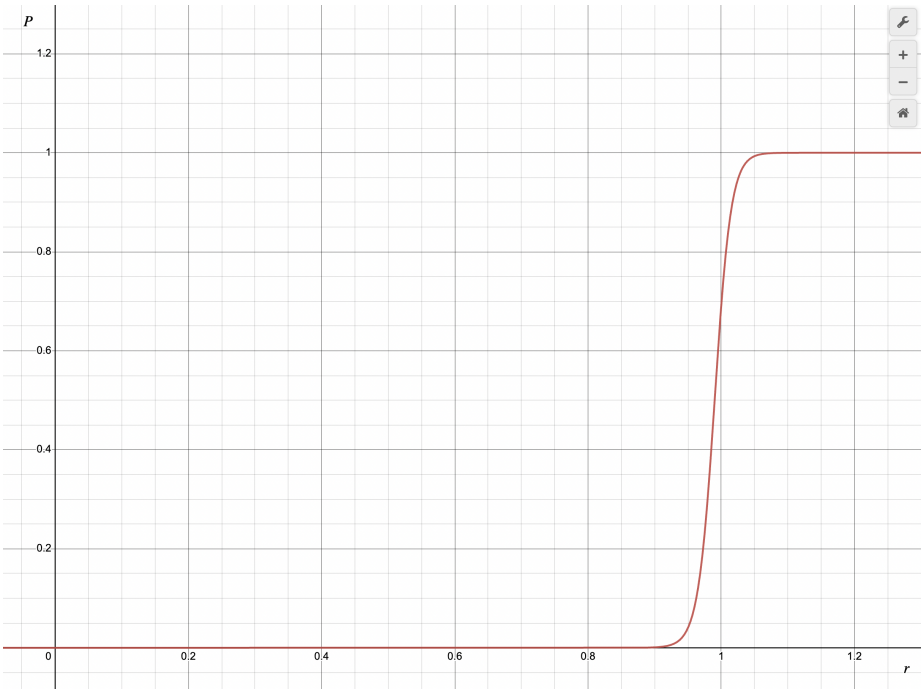
Before we get deeper into the problem, we would like to show an example in 1-D. If we set $J_0^{(x)} = J_0^{(y)}$ and $J_1^{(x)} = J_1^{(y)}$ then equation 1 reduces to:

$$|\langle 0|\psi\rangle|^2 = \frac{e^{-2J_0(2-r^2)}}{e^{-2J_0^{(x)}(2-r^2)} + e^{-2J_1^{(x)}(r^2)}} \tag{3}$$

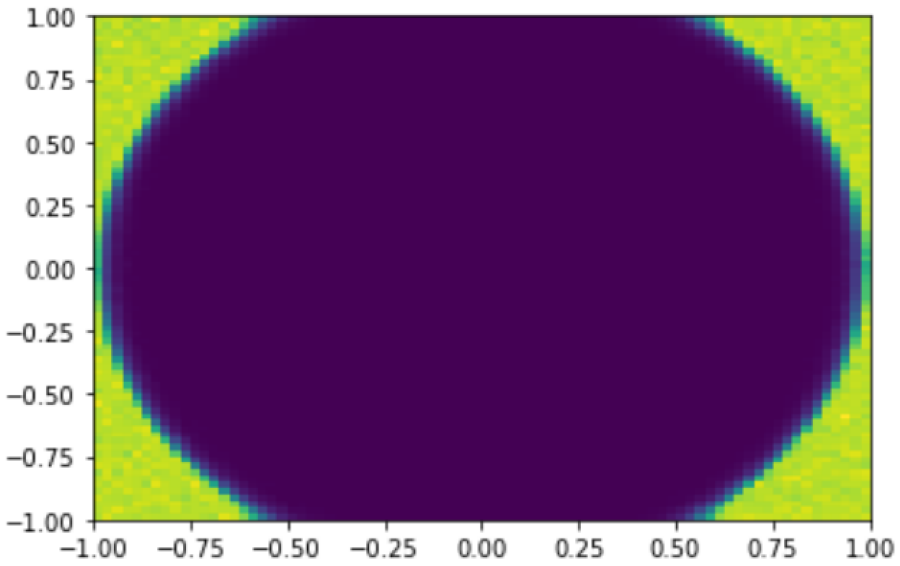
where $r^2 = x^2 + y^2$. When plotted it looks like this,



when $J_0 = .8$ and $J_1 = .9$, and like this,



when $J_0 = 10$ and $J_1 = 10.4$. Notice that as J increases the posterior function looks more and more like a step function. This is very easy to see if we were to simulate it. If we were to plot out the histogram of the accepted points in the $x - y$ plane then we would expect a circular, low region with a radius of about 1, and then a very quick climb to a high region everywhere else. When we ran the simulation, that is exactly what we saw.



Since the theory visually matches the simulation, we believe our simulation is at least working and probably giving the right answer. However, there are other, better methods of confidence.

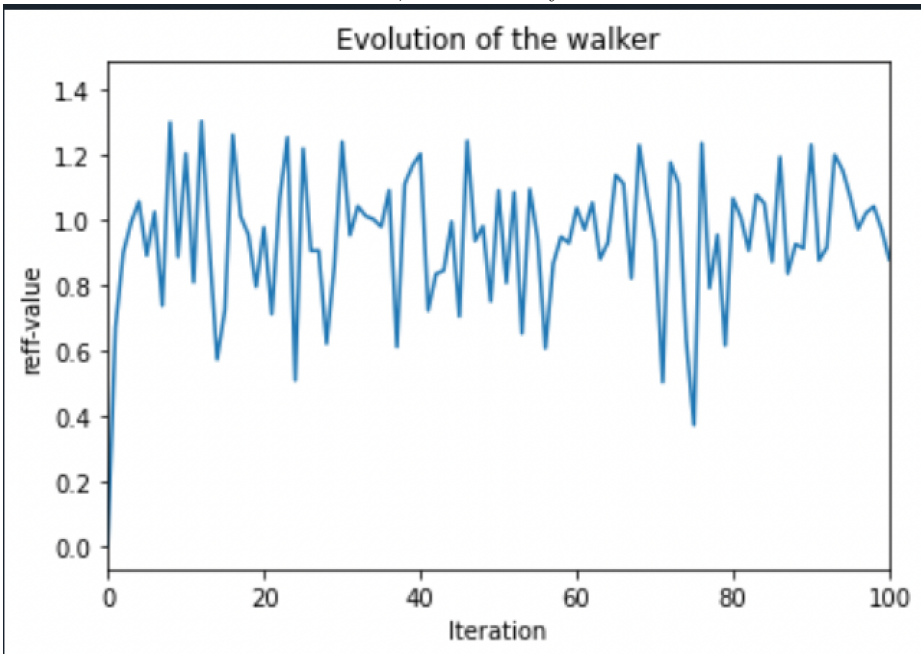
3 Convergence Testing

Convergence is the measure of how trustworthy a Markov chain Monte Carlo simulation is. Although convergence testing can tell us if our simulation is not accurate, it cannot tell us, for certain, if the simulation is completely accurate. For that reason, using several types of convergence testing can increase confidence in simulation results. For this section, and the rest of this paper, we will be using the conditions $J_0^{(x)} = 1$, $J_0^{(y)} = 0.95$, $J_1^{(x)} = 2$, and $J_1^{(y)} = 1.4$ with initial conditions for x and y being 0.

3.1 Naive Approach

A basic approach to test whether or not the chain has converged is to eyeball the graph of the iterated variable(s) and number of iterations. If your initial conditions are far away from the true value, the chain will start moving towards the true value. These iterations are called the burn-in. After some time (possibly infinite) the chain will converge to the correct value. This, however, implies that if we believe that the chain has converged at some finite time we cannot be sure that has actually converged. For our simulation, the values of x and y are not going to visually converge since they can bounce between positive and negative values. So instead, our naive approach will be to take an effective radius $r_{eff} = x^2+y^2$ for each x and y in the chain and see if that converges. Since we know that the particle is less likely to be in the ground state near the origin, a converged

chain should not consistently be in that region. Since our chain has more than four hundred thousand iterations, we will only show the first 100 iterations.



In this graph we see that almost instantly the the walker jumps away from the origin and doesn't return. This is a pretty good sign that the walker converges. Of course, this walker shouldn't converge to one value because the optimal value of r_{eff} will fluctuate based on the differences in the J, but since the walker is oscillating in a region around the optimal values that suggests that the x and y walkers might be converging.

3.2 Auto Correlation Length

3.3 Gelman Rubin Statistic

The Gelman Rubin statistic is a more sophisticated, numeric method of testing convergence. We will spare a lot of the details, but the principle is to check if chains with different initial condition are alike or unalike. The procedure is to generate chains with several different initial conditions then calculate the following:

$$
\begin{aligned}
\bar{x}_j &= 1/L \sum_{i=1}^{L} x_i^{(j)} \\
\bar{x}_\cdot &= 1/J \sum_{j=1}^{J} \bar{x}_j \\
B &= \frac{L}{J-1} \sum_{j=1}^{J} (\bar{x}_j - \bar{x}_\cdot)^2 \\
s_j^2 &= \frac{1}{L-1} \sum_{i=1}^{L} (x_i^{(j)} - \bar{x}_j)^2 \\
W &= \frac{1}{j} \sum_{j=1}^{J} s_j^2 \\
R &= \frac{\frac{L-1}{L} W + \frac{1}{L} B}{W}
\end{aligned}
\tag{4}
$$

J is the number of chains with different starting conditions, j is the j'th chain in a list of chains. L is the length of the chain after discarding a number of entries at the begining to account for the burn in. Notice that B is the variance between chains and W is the average variance of all the chains, thus as R approaches 1, then the chains are alike. Therefore, if R is close to 1 that would suggest that the chain converges.

For our simulation we created a matrix of initial conditions where both variables ranged from -1 to .8 separated by intervals of .2. This created a total of 81 chains. The value for R in our x domain was .999990 and in the y domain was 999992. This suggests that our model is converging, but to reiterate, this is not conclusive.

4 Conclusion

4.1 Comparison of Gelman-Rubin Statistic and Autocorrelation Lengths

The Gelman-Rubin statistic, denoted as R, is a measure of convergence for multiple chains in an MCMC simulation. A value close to 1 suggests that the chains have converged. Here, we have obtained a Gelman-Rubin statistic of approximately 1.23.

Now, let's compare this with the autocorrelation lengths:

- **Autocorrelation Lengths:** $[0.86045635, 0.80177047]$
- **Gelman-Rubin Statistic:** 1.225817482686245

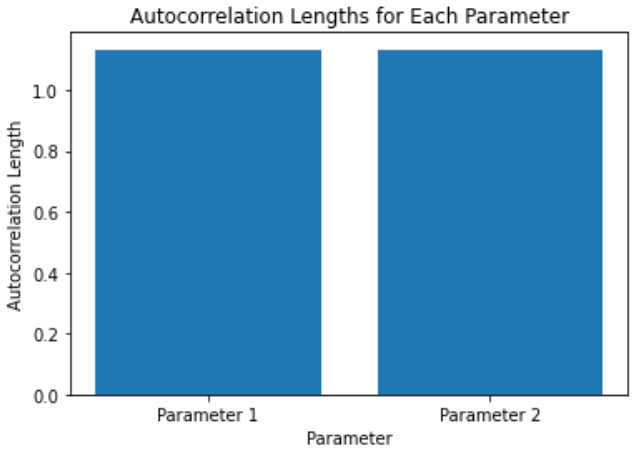


Figure 1: Enter Caption

Interpretation:

1. **Gelman-Rubin Statistic:**
    - The Gelman-Rubin statistic is slightly larger than 1 (1.23). While values close to 1 indicate convergence, a value slightly larger than 1 may suggest that the chains have not fully converged.
2. **Autocorrelation Lengths:**
    - Autocorrelation lengths around 0.8 suggest that, on average, each sample is correlated with its predecessor for about 0.8 iterations.

Possible Explanations:

- The autocorrelation lengths indicate relatively good mixing and efficiency in terms of sample independence.
- The Gelman-Rubin statistic may be slightly elevated due to longer-term correlations or more subtle convergence issues not fully captured by the autocorrelation lengths.

Suggestions:

- Evaluate other convergence diagnostics: Continue to use visual inspection of trace plots and other diagnostics to assess convergence.
- Increase the number of iterations: Running the MCMC sampler for more iterations may help improve convergence and reduce the Gelman-Rubin statistic.

Results: While the autocorrelation lengths suggest reasonably good efficiency and mixing, the slightly elevated Gelman-Rubin statistic indicates a potential need for further investigation. Additional diagnostics and longer simulation runs may help in refining the assessment of convergence and the reliability of parameter estimates.

4.2 Evolution of the Walker using EMCEE

The *Evolution of the Walker* plot for emcee typically shows the progression of the walkers in the parameter space over iterations. The three distinct lines in the plot signify:

- **Horizontal Line at the Bottom (Parallel to x-axis):**

  This horizontal line could indicate that one of the parameters (perhaps the x-coordinate) is not changing or exploring much during the iterations. The walkers might have found a region where the posterior is relatively flat in the direction of this parameter. A horizontal line at the bottom suggests that one of the parameters has reached a **stable region** where it is not changing much. This could indicate **convergence in that parameter.**

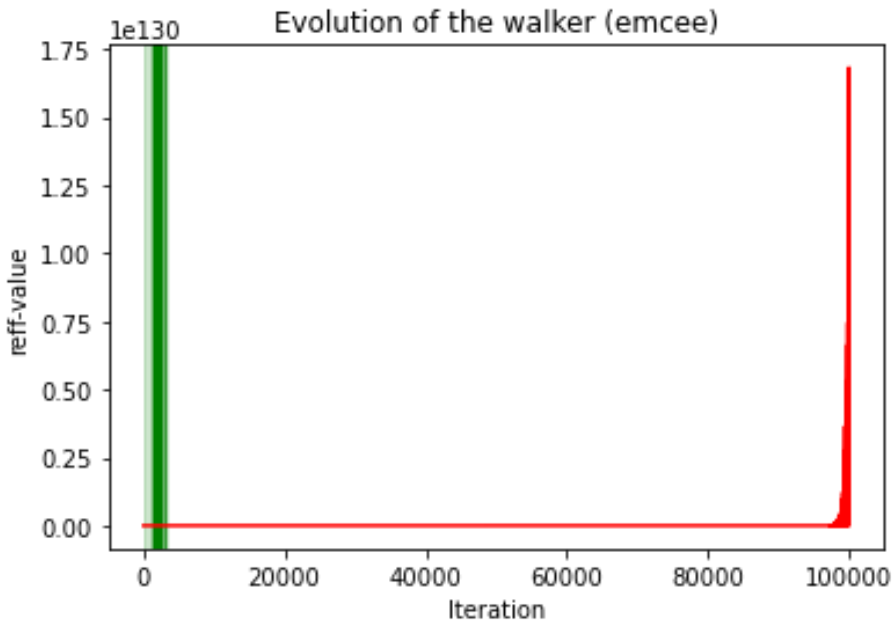


Figure 2: Enter Caption

- **Vertical Line at the Far Right Corner (Red, Parallel to y-axis):**

  A vertical line at the far right could suggest that the walkers are exploring values along one parameter (perhaps the y-coordinate) but not changing much in the other parameter. This could indicate a region where the posterior is relatively flat in the direction of this parameter. While this doesn't necessarily indicate **instability,** it does show that the walkers are spending more time exploring along this parameter.

- **Vertical Line at the Left Side (Green, Parallel to y-axis):**

  Similar to the red line, a vertical line at the left side could suggest exploration along another parameter (perhaps the y-coordinate) but in a different region of the parameter space.

4.3 Trace Plots

The stability of an MCMC sampler is often assessed by examining the convergence of the walkers' traces. In the "Evolution of the Walker" plot for emcee, stability can be inferred by looking for signs of convergence and whether the traces oscillate around a fixed point.

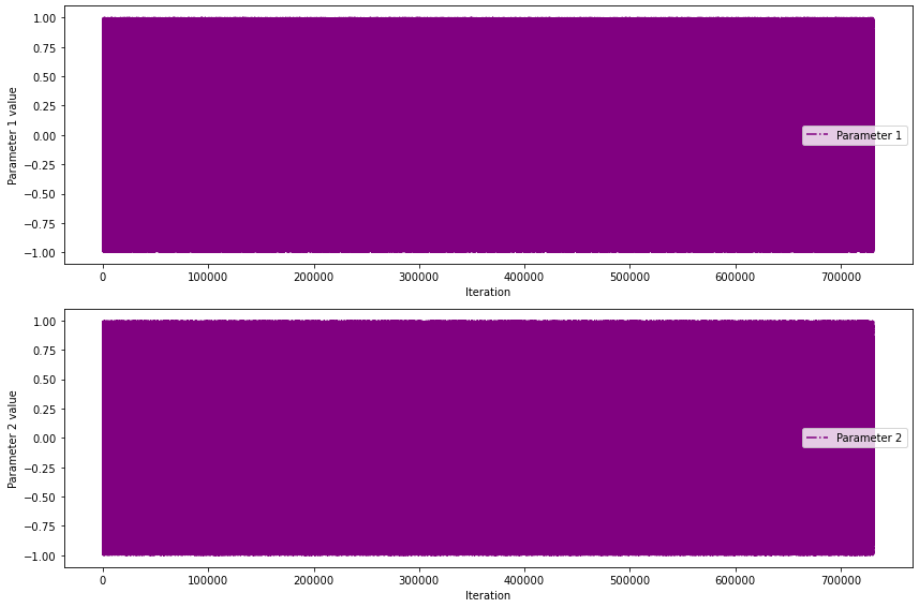


Figure 3: Trace plots for each parameter in the chain.