MCMC Methods for Calculation of Feynman Path Integrals

Rowan d'Auria Supervisor: Dr. Aidin Masouminia

Introduction

In my project I am investigating various Markov Chain Monte Carlo (MCMC) methods, focusing on performance comparisons between historical MCMC methods and newer, more efficient implementations. The motivation for this is their straightforward application to computation of multidimensional integrals used in calculating Feynman Path Integrals.

Theory

Feynman Path Integrals provide an alternate method for computing the time evolution of a quantum state that uses a weighted sum of all possible paths between states (x_a, t_a) and (x_b, t_b). The path weight is defined as the imaginary exponential of classical action along the path in units of $\hbar$. The time evolution of the states is performed by the propagator

$$G(b,a) = \sum_{paths} e^{iS[b,a]/\hbar}$$

where G is the propagator, S is the classical action and b, a are the states (x_b, t_b) and (x_a, t_a) respectively. To make this problem numerically soluble, we discretise the path of the particle so that it moves in steps rather than along a continuous path. This has the effect of limiting the possible paths and thus limiting computation time. The propagator then has the form

$$G(b,a) = A \int e^{-\Delta t E(x_j)} dx_1 ... dx_{N-1}$$

where E is the total energy change along the path, Δt is the time interval between discrete path points and A is a normalisation constant. This is an $N-2$ dimensional integration where N is the number of path points. It is an $N-2$ dimensional integral rather than N because the points x_a and x_b are held fixed, while all intermediate points are integrated across (in theory $-\infty < x_j < \infty$).

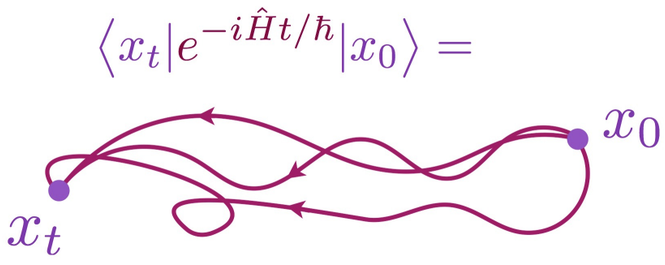

Figure 1: Visual representation of a Feynman path integral. Taken from [1].

An issue with computing these integrals is that they are computationally intensive, even for a relatively coarse lattice of 20 points. However, MCMC algorithms are ideally suited to computing fast approximations of multidimensional integrals. But what are they? Essentially, they are a hybrid of two mathematical tools, Markov Chains and Monte Carlo Sampling:

- Markov chains are a random walk process where the conditional probability of the kth step is solely dependent on the outcome of the k-1th step.
- Monte Carlo is a sampling method where random samples are generated according to a probability distribution.

These two concepts are combined to create a sequenece of events where the kth point is randomly selected according to a probability distribution related to the k-1th outcome.
Summary: MCMC algorithms can be used to find the paths between states with the largest weightings, allowing a good approximation of the multidimensional integral used to calculate the propagator.

Hamiltonian Algorithm

The Hamiltonian approach to MCMC uses a similar iterative structure to the Metropolis Algorithm, but it is more effective at exploring the sample space. The Hamiltonian Algorithm (HMC) uses properties of classical Hamiltonian dynamics to produce distant proposals with a high probability of acceptance. If we introduce a Hamiltonian $H(q,p)$ where q and p are position and momentum, position is our variable of interest and momentum is a construct related to each position, q, that allows Hamiltonian dynamics to operate. In this context, q is a path point. Typically, the Hamiltonian looks like

$$H(q,p) = U(q) + K(p)$$

where $U(q)$, the potential energy, is defined as $U(q) = -log(P(q))$ where $P(q)$ is the probability distribution of the target variable[2].
An essential property of the Hamiltonian for MCMC is that it's conserved through changes of position and momentum. The equations defining how q and p change with time

$$\frac{dq}{dt} = \frac{\delta H}{\delta p} \quad ; \quad \frac{dp}{dt} = -\frac{\delta H}{\delta q}$$

can be discretised to provide finite time progressions, ε, of q and p. Because the Hamiltonian is almost preserved (it would be entirely preserved in the limit $\varepsilon \to 0$), almost all points proposed are accepted. This is because the acceptance criteria are very similar to that of the Metropolis Algorithm. For Metropolis, if $\Delta E = 0$ for all perturbations, acceptance probability is 1 for all proposals.

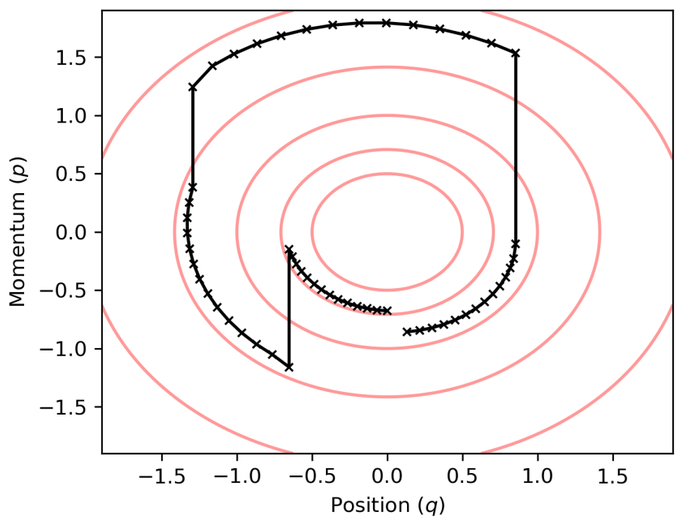

Figure 2: Plot demonstrating q updates that preserve the Hamiltonian. 13 incremental steps were used with $\varepsilon = 0.3$. Red lines indicate paths of constant energy.

Metropolis Algorithm

The Metropolis Algorithm is the simplest approach to MCMC. The algorithm implemented for our use works like this[3]:

1. Perturb point x_i in our path by a random distance ϵ, $x_i' = x_i + \epsilon$ where x_i' is the perturbed point.
2. Calculate the change in energy along the path, $\Delta E = E[x'] - E[x]$ where $E[x']$ is the energy along the unperturbed path, $E[x]$ is the energy along the original path, and ΔE is the energy difference.
3. Accept the perturbed point, x_i', with probability $\min[1, \exp(-\Delta t \Delta E)]$.
4. Move on to point x_{i+1} and repeat to the end of the path.

The above procedure is repeated a large number of times, performing small optimisations to the path each time. Statistical correlation between paths is reduced by recording every jth path. For example, the algorithm can be run 10,000 times, and every 100th is recorded.

Conclusion

So far, I have successfully implemented both a Metropolis and HMC algorithm that can explore a harmonic potential, simulating the ground state wavefunction of a trapped particle. Both algorithms can be easily applied to more complex potentials, or used to simulate higher-dimensional distributions.

Future

Although HMC is demonstrated to outperform Metropolis in a harmonic potential, calculations for higher-dimensional systems increase performance advantages compared to Metropolis. Neal (2011) states that the computing time increases as $d^{5/4}$ and d^2 for HMC and Metropolis respectively, where d is the number of dimensions. Furthermore, path integral approaches can be used to calculate approximations of the ground state energy, as well as excited states. I aim to simulate the eigenfunctions of higher dimensional potentials, and use these to calculate real values of the energy states, comparing them to the literature.

Results

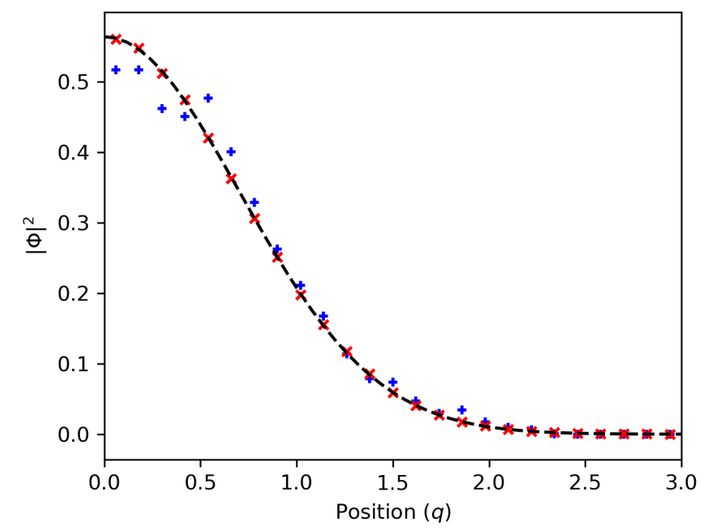

Figure 3: Plots of the numerical approximations of the ground state probability distribution in a harmonic potential, calculated using the Metropolis Algorithm (blue +) and the Hamiltonian Algorithm (red x). The black line is the analytic solution. Only half of the symmetrical distribution is plotted here.

A good fit to the theoretical probability distribution was shown with the HMC Algorithm. However, application of the Metropolis Algorithm resulted in a worse quality of fit to the exact solution. The largest differences arise at the top of the probability distribution. 14,000 proposals were performed with the Metropolis Algorithm. To match computing time, 1000 proposals were performed with the Hamiltonian Algorithm, with 14 steps to find each position update.

References

[1] C. Althorpe Group. http://www-stuart.ch.cam.ac.uk/page0.html, accessed 24/11/19.

[2] R. M. Neal, "Mcmc using hamiltonian dynamics," 2012. Handbook of Markov Chain Monte Carlo, Ch. 5.

[3] G. P. Lepage, "Lattice qcd for novices," 2005. arXiv:hep-lat/0506036.

[4] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," 2017. arXiv:1701.02434.

Contact Information

Email rowan.d'auria@durham.ac.uk