GROUND STATE ENERGY SIMULATIONS USING MONTE CARLO RANDOM WALK

ELLIOTT ASHBY
PHYSICS AND ASTRONOMY
UNIVERSITY OF SOUTHAMTON

Abstract. Placeholder for abstract.

Contents

Date: March 2025.

1. Introduction

Overview of the challenges in solving the Schrödinger equation numerically The connection between the imaginary-time Schrödinger equation and diffusion processes Historical development of Monte Carlo methods in quantum mechanics Brief comparison with alternative numerical approaches (matrix diagonalization, variational methods) Motivation: why random walks are particularly well-suited for ground state energy calculations

2. Method

2.1. Theoretical Foundation. While the usage of Monte Carlo methods to simulate quantum mechanics may seem strange at first, looking at the time-dependent Schrödinger equation as a diffusion equation in imaginary time reveals the connection that makes it possible. Here is the time-dependent Schrödinger equation for a particle of mass m in a potential $V(x)$:

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x,t) + V(x)\psi(x,t) = \hat{H}\psi(x,t) \tag{2.1}$$

By using $\tau = it$, it becomes:

$$\frac{\partial}{\partial \tau}\psi(x,\tau) = \frac{\hbar}{2m}\frac{\partial^2}{\partial x^2}\psi(x,\tau) - \frac{V(x)}{\hbar}\psi(x,\tau) = -\frac{\hat{H}}{\hbar}\psi(x,\tau) \tag{2.2}$$

This is a diffusion equation with a growth/decay term with a diffusion constant $D = \hbar/2m$ where the growth/decay term determines whether the population density of walkers at a given position increases or decreases.

The general solution to equation 2.2 can be written using the eigenstates and eigenvalues of the Hamiltonian:

$$\psi(x,\tau) = \sum_n c_n\phi_n(x)e^{-E_n\tau/\hbar} \tag{2.3}$$

where $\phi_n(x)$ are the eigenfunctions of the Hamiltonian with corresponding energies E_n, and c_n represents the projection of the initial wavefunction onto each eigenstate. If we consider the case where $\tau \to \infty$, all terms decay exponentially, but the ground state decays the slowest:

$$\lim_{\tau\to\infty}\psi(x,\tau) = c_0\phi_0(x)e^{-E_0\tau/\hbar} \tag{2.4}$$

If we then were able to simulate the diffusion process for a sufficiently long time, then the distribution of our random walkers would approach the ground state wavefunction $\phi_0(x)$, so the decay rate of the walker population would be the ground state energy E_0.

2.2. Implementation Details. In order to not have infinite computation time, we must discretize space and time. To do this we can use a lattice with spatial sites j and temporal sites n. The probability $p(j,n)$ for a walker to be at lattice site j after n time steps satisfies:

$$p(j,n+1) = \frac{1}{2}[1-a(j)][p(j-1,n) + p(j+1,n)] \tag{2.5}$$

and converting to continuous space and time with intervals h_x and h_τ:

$$p(x, \tau + h_\tau) = \frac{1}{2}\left[1 - a(x)\right]\left[p(x - h_x, \tau) + p(x + h_x, \tau)\right] \tag{2.6}$$

Here, $a(x) = h_\tau V(x)$ and represents the probability of a walker being absorbed at position x.

So for $\tau \to \infty$, swapping $\psi(x, \tau)$ for $p(x, \tau)$ and defining $h_\tau = h_x^2$ we get the final form of the diffusion equation:

$$\frac{\partial p}{\partial \tau} = \frac{1}{2}\frac{\partial^2 p}{\partial x^2} - V(x)p \tag{2.7}$$

2.3. Algorithm Variants. Using the diffusion equation 2.7 we can derive a couple different ways to simulate the ground state energy of a quantum system.

The simplest method is the simple absorption method, where walkers are absorbed at the boundaries of the lattice. This method is simple to implement but can be computationally expensive for large systems.

A more efficient method is the diffusion Monte Carlo method, which uses branching to kill and birth new walkers. This method is much more efficient but can be more difficult to implement due to the branching process.

2.3.1. *Simple Absorption Method.* In the case for the simple absorption method, our potential is implemented as a infinite square well potential. The probability of a walker being absorbed at a given position is given by:

$$a(x) = \begin{cases} 0 & \text{if } x \in [-L/2, L/2] \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

where L is the width of the well.

Using this potential, we can implement the diffusion equation as a random walk on a lattice where:

$$P_{survival}(\tau) = \int_{-L/2}^{L/2} p(x, \tau)^2 dx \approx e^{-\lambda \tau} \tag{2.9}$$

The algorithm is then as follows:

Simple Absorption Method

(1) Initialize the walker population at the centre of the lattice.
(2) For each walker, move it to a neighbouring lattice site with equal probability to move to any of the neighbouring sites.
(3) If the walker is at the boundary of the lattice ($x \geq L/2$), kill it and record the number of steps it took to reach the boundary.
(4) Repeat steps 2-3 for a maximum number of steps. If any of the walkers survive, mark them as surviving.
(5) Find the number of walkers that survive at each step and using a linear regression calculate the decay rate of the population λ.
(6) The ground state energy can then be approximated as $E_0 = \lambda(L/2)^2$.

2.3.2. *Diffusion Monte Carlo with Branching.* The diffusion Monte Carlo with branching method is more complex than the simple absorption method and requires some additional theory.

One of the problems with the simple absorption method is that to get reasonable accuracy, a very large number of initial walkers is required. This can be a problem for large systems since the run time will be $O(N)$ where N is the number of walkers since every walker must be simulated individually.

While we cannot lower the run time, we can lower the number of initial walkers and also greater our chances of finding the ground state. If we were to kill walkers that have a high probability of being absorbed and birth new walkers at positions with a low probability of being absorbed, we can attempt to maintain a more constant number of walkers throughout the simulation.

Since this approach requires us to find the probability of a walker being absorbed at some position, it is reasonable to use a potential that is continuous rather than the infinite square well potential. I have chosen to use the harmonic oscillator potential with $m = \omega = 1$:

$$V(x) = \frac{1}{2}x^2 \tag{2.10}$$

The probability of a walker being absorbed at a given position is then:

$$a(x) = h_\tau V(x) = \frac{1}{2}h_\tau x^2 \tag{2.11}$$

The algorithm is then as follows:

Diffusion Monte Carlo with Branching

(1) (a) Initialize N_0 walkers in a Gaussian distribution around the centre of the lattice and record their positions x_i.
    (b) Compute initial reference potential energy $V_{ref} = \sum_i V(x_i)/N_0$.
(2) For each walker:
    (a) Move it to a neighbouring lattice site with equal probability to move to any of the neighbouring sites.
    (b) Compute the potential energy at the new position $V(x)$ and compute the change in potential energy $\Delta V = V(x) - V_{ref}$.
    (c) If $\Delta V > 0$, kill the walker with probability $|\Delta V \times h_\tau|$. Otherwise, birth a new walker at the new position with the same probability.
(3) Re-calculate $V_{ref} \leftarrow \sum_i \frac{V(x_i)}{N_a} - k\left(\frac{N_a - N_b}{N_b \cdot h_\tau}\right)$ where N_a is the number of walkers after step 2 and N_b is the number of walkers before step 2 and k is a constant to tune.
(4) Repeat steps 2-3 for a maximum number of steps or until the number of walkers reaches zero.
(5) The ground state energy can then be approximated as $E_0 = V_{ref}$.

3. Results

3.1. Simple Absorption Method.

3.2. Diffusion Monte Carlo with Branching.

4. Discussion

While extension to higher dimensions is straightforward, focusing on one-dimensional cases allows for easier visualization and comparison with analytical solutions. I opted to spend more time on the one-dimensional case since it allowed for easier debugging, testing and iteration of the code.

4.1. Parameter Optimization.

Effects of lattice spacing, time step, and walker population Balancing computational cost and accuracy Handling statistical fluctuations through averaging techniques

4.2. Method Strengths and Limitations.

Robust performance in higher dimensions Challenges with excited states and nodal surfaces Comparison with traditional numerical methods Applicability to different potential types

4.3. Advanced Applications.

Extensions to field theory with probability

$$a[\phi] = h_\tau \left(v \sum_{j=1}^{N} V(\phi)_j E_s \right) \tag{4.1}$$

Many-body quantum systems Handling sign problems for fermions

5. Conclusion

Summary of key findings and method effectiveness Practical recommendations for implementation Future directions for method improvement Potential applications to more complex quantum systems

Appendix

Detailed derivations of key equations Pseudocode for the main algorithms Benchmark data and supplementary results

Appendix A. Additional Citations

18 Examples
6 for each: random walk, Monte Carlo, random numbers
3 in academia, 3 outside academia

References