Diffusion Monte Carlo Method: an Approach to Ground-State Estimation in Quantum Systems

Student No.: K24090171

13 January 2025

7CCMMS39 Modern Computational Techniques for Theoretical Physicists
King's College, London

Abstract

This project provides an in-depth exploration of Diffusion Monte Carlo (DMC), starting with a discussion on the background and history of both Monte Carlo and Quantum Monte Carlo methods. It then presents the mathematical formulation of DMC derived from path integrals. A Python-based computer program is described and implemented, which allows users to input quantum systems and obtain numerical solutions for their ground state wavefunction and energy using DMC. The examples covered in this project include the harmonic oscillator, the Morse oscillator, the harmonic oscillator with a quartic anharmonic perturbation, the particle in a well (both finite and infinite), the hydrogen atom, and higher-dimensional bosonic systems.

Contents

1 Introduction

The equation of motion in quantum systems is given by the Schrödinger equation (SE). The reason for employing Monte Carlo methods is that in many cases the Schrödinger equation cannot be solved analytically. Solutions to the Schrödinger equation are given by a superposition of solutions of the form

$$\psi(x) \exp\left(\frac{-i}{\hbar} E_n t\right),$$

where $\psi(x)$ are solutions to the time independent Schrödinger equation (TISE), "stationary states", and E_n is the energy associated to the n^{th} energy level of the quantum system, $n = 0, 1, 2, \ldots$.[1].

In the diffusion Monte Carlo method (DMC), one chooses the time to be imaginary by rotating it into the complex plane ($t \to i\tau$). Now, the solutions to the SE are given by a superposition of solutions $\exp\left(\frac{-E_n \tau}{\hbar}\right)$.

The basis of the DMC method is that under the evolution of imaginary time, the ground state energy, $E_0 < E_n$, is that which is associated with the longest lasting solution. In order to determine the ground state energy and hence the ground state wave function, one may allow the state to evolve over a long enough period of imaginary time away from its (arbitrary) initial state [2].

There exist two formulations of the DMC method. The first is based on a comparison of the Schrödinger equation and a generalised diffusion equation (GDE). The kinetic term in the SE corresponds to the diffusion term in the GDE. The potential term in the SE corresponds to the source/sink or reaction term in the GDE. One may solve the diffusion equation with stochastic calculus and the SE may be solved using random walks of particles with their birth/death rates governed by the source/sink terms of the GDE and their probability distribution given by their wave function.

The second formulation is based on the Feynman path integral solution of the time-dependent Schrödinger equation (TDSE). In this method the wave function is expressed (by path integrals) as a multi-dimensional integral. This integral is solvable by the Monte Carlo method. Algorithms to solve each of these two methods lead to the same formulation of the DMC method.

In this project we will focus on the second formulation, as the first formulation is only applicable for systems with wave functions that are positive everywhere, and this would limit the scope of our examples.

2 Background and Theory

2.1 History of Quantum Monte Carlo Methods

Monte Carlo methods may only be loosely defined as a statistical method for solving mathematical systems using pseudo-random numbers to generate a range of trial solutions [3]. However, a universally accepted definition does not exist.
Georges-Louis Leclerc de Buffon is accredited with being the first mathematician to study random trials with his 1733 paper and supplemented by his 1777 book [4]. In these he proposed the needle problem: if one has a surface made of parallel strips and drops a needle on this surface, what is the probability that the needle will lie across a border of two strips? This was a problem in geometric probability, and solved using integral geometry. Even though it was not his intent, one may use the solution to this problem to design a Monte Carlo method for approximating π. However, formal study of Monte Carlo methods was not seen until the 1930s. There was an explosion of advancements in the field in the late 1940s, with the motivation of developing nuclear weapons. A key issue that was not fully understood was neutron diffusion. It was essential that the physicists at Los Alamos Laboratory could predict how much energy a neutron would give off from a collision during detonation, but it seemed impossible to predict with conventional mathematical methods. Ulam suggested the idea of using random operations and worked with Von Neumann among others to implement calculations on the neutron diffusion problem. Metropolios suggested the semi-secretive name of Monte Carlo, borrowed from name of the Casino where Ulam's uncle went to gamble [5].
Quantum Monte Carlo (QMC) methods were first developed by Fermi and Richtmyer at Los Alamos by their work on mean-field particle view of neutron diffusion, which has since been declassified [6]. In 1984 Hetherington developed a reduced matrix model method of ground state approximation contributing hugely to modern quantum Monte Carlo methods. QMC methods are mainly classified by zero-temperature (ground state only) and thermodynamic methods. Diffusion Monte Carlo methods are zero-temperature methods which can be interpreted as a mean-field particle Monte Carlo approximation of Feynman–Kac path integrals [7].

2.2 Foundations and Context for the Path Integral Approach

Goal: Write the solution of the time dependent Schrodinger equation (TDSE) as a series expansion of eigenfunctions of the Hamiltonian. Then perform a Wick rotation: $t \to -i\tau$. The series solution of the Schrödinger equation then decays exponentially in imaginary time. The ground state is identified as the longest-lasting component.

Consider a single particle traveling through a potential $V(x)$ with wavefunction $\Psi(x)$. The TDSE is given by

$$i\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi,$$

and its Hamiltonian given by

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x).$$

Assume that the particle is confined in space, i.e. $V(x) \to \infty$, as $x \to \pm\infty$, then we can write all the solutions of the TDSE as

$$\Psi(x,t) = \sum_{n=0}^{\infty} c_n\phi_n(x)e^{-\frac{i}{\hbar}E_n t},$$

where $\phi_n(x)$ are eigenfunctions of the Hamiltonian (which are square integrable functions for the case of confined particles). The eigenenergies are given by the TISE $\hat{H}\phi_n(x) = E_n\phi_n(x)$ with boundary conditions $\lim_{x\to\pm\infty}\phi_n(x) = 0$. We may order these eigenenergies $E_0 \leq E_1 \leq \ldots$. We may also assume that the eigenfunctions are orthonormal and real and the coefficients c_n are given by

$$c_n = \int_{-\infty}^{\infty} dx\ \phi_n(x)\Psi(x,0),$$

i.e. the overlap between the initial state with the eigenfunctions [1].
The first step in our transformation process is to perform a trivial shift of the energy scale

$V(x) \to V(x) - E_R$ and $E_n \to E_n - E_R$, for some reference energy (guess of energy) E_R. The shifted Schrödinger equation is given by

$$i\hbar \frac{\partial \Psi}{\partial \tau} = -\frac{\hbar^2}{2m}\frac{\partial^2 \Psi}{\partial x^2} + [V(x) - E_R]\Psi,$$

and its series solution given by

$$\Psi(x,t) = \sum_{n=0}^{\infty} c_n \phi_n(x) e^{-i\frac{E_n - E_R}{\hbar}t}.$$

The second step is to perform a Wick rotation by writing these in terms of $\tau = it$. The rotated Schrödinger equation is given by

$$\hbar \frac{\partial \Psi}{\partial \tau} = \frac{\hbar^2}{2m}\frac{\partial^2 \Psi}{\partial x^2} - [V(x) - E_R]\Psi,$$

and its rotated solution given by

$$\Psi(x,\tau) = \sum_{n=0}^{\infty} c_n \phi_n(x) e^{-\frac{E_n - E_R}{\hbar}\tau}.$$

The eigenstates of the Hamiltonian remain 'energy ordered'. We use this to examine the asymptotic behaviour of the system as $\tau \to \infty$. Firstly, if $E_R - E_0 < 0$, then the wave function diverges exponentially. If $E_R - E_0 > 0$, the wave function vanishes exponentially. Finally, if $E_R - E_0 = 0$ the wave function converges to $c_0\phi_0(x)$ regardless of the initial wave function, as long as there is sufficient overlap between it and ϕ_0, i.e. as long as c_0 is not too small. c_0 is non-vanishing for a positive definite initial wave function centered in a region of sufficiently large $\phi_0(x)$.

2.3 Path Integral Formalism and Monte Carlo Formulation

Goal: Using path integrals, reduce the solution to computable definite integrals and a provided initial state wave function.

Given a positive definite ground state wave function, one may use the classical Monte Carlo method to solve the Schrödinger equation to any accuracy. In this case the wave function itself is identified as the probability density We will implement the Monte-Carlo algorithm for calculating our large multi-dimensional integral by alternating diffusive displacements and birth-death processes to a set of imaginary particles in the configuration space. These imaginary particles are known as replicas and their spatial distribution converges to the ground state wave function. We will simulate the diffusive displacements and birth-death processes using random number generators.

We will implement a practical way to integrate the shifted and Wick rotated Schrödinger equation for an arbitrary guess of energy E_R and initial wave function $\Psi(x,0)$.

The solution of the shifted, Wick rotated Schrödinger equation can be written in the form [8]

$$\Psi(x,\tau) = \int_{-\infty}^{\infty} dx_0\ K(x,\tau|x_0,0)\Psi(x_0,0),$$

where $K(x,\tau|x_0,0)$ is a propagator as is expressed in terms of a path integral:

$$\begin{aligned} K(x,\tau|x_0,0) &= \lim_{N\to\infty} \int_{-\infty}^{\infty} dx_1 \cdots \int_{-\infty}^{\infty} dx_{N-1} \left(\frac{m}{2\pi\hbar\Delta\tau}\right)^{\frac{N}{2}} \\ &\times \exp\left\{-\frac{\Delta\tau}{\hbar}\sum_{j=1}^{N}\left[\frac{m}{2\Delta\tau^2}(x_j - x_{j-1})^2 + V(x_j) - E_R\right]\right\}, \end{aligned}$$

where the timestep $\Delta\tau = \frac{\tau}{N}$. Define a probabilty density function

$$P(x_n, x_{n-1}) \equiv \left(\frac{m}{2\pi\hbar\Delta\tau}\right)^{\frac{1}{2}} \exp\left[-\frac{m(x_n - x_{n-1})^2}{2\hbar\Delta\tau}\right].$$

This is is related to the kinetic energy term in the Hamiltonian and is identified as a Gaussian probability density for the random variable x_n (since it is normalised) which has its mean equal to x_{n-1} and variance $\sigma = \sqrt{\frac{\hbar\Delta\tau}{m}}$. Also, define a weight function [2]:

$$W(x_n) \equiv \exp\left[-\frac{[V(x_n) - E_R]\Delta\tau}{\hbar}\right].$$

Then, the wavefunction may be written in the form:

$$\begin{aligned}\Psi(x,\tau) &= \lim_{N\to\infty}\int_{-\infty}^{\infty}\left(\prod_{j=0}^{N-1} dx_j\right)\prod_{n=1}^{N} W(x_n) \\ &\times \; P(x_n, x_{n-1})\Psi(x_0, 0).\end{aligned}$$

The path integral above cannot always be evaluated analytically, but with a sufficiently large value of N, one can evaluate it to any desired accuracy. A large value of N facilitates and encourages the use of Monte Carlo methods..

The Monte Carlo method uses that any N-dimensional convergent integral of the form

$$I = \int_{-\infty}^{\infty}\left(\prod_{j=0}^{N-1} dx_j\right) f(x_0,\ldots,x_{N-1})\mathcal{P}(x_0,\ldots,x_{N-1}),$$

for a probability density $\mathcal{P}$, can be approximated by

$$\mathcal{I} = \frac{1}{\mathcal{N}}\sum_{\substack{i=1 \\ x^{(i)}\in\mathcal{P}}} f\left(x_0^{(i)},\ldots,x_{N-1}^{(i)}\right).$$

The larger $\mathcal{N}$ is, the better approximation $\mathcal{I}$ is to I. According to the central limit theorem, values of $\mathcal{I}$ obtained are normally distributed around I with standard deviation $\frac{1}{\sqrt{\mathcal{N}}}$.

We want to evaluate $\Psi(x,\tau)$ in the path integral, for given x, τ and N. Define

$$\mathcal{P}(x_0,\ldots,x_{N-1}) = \prod_{n=1}^{N} P(x_m, x_{n-1}),$$

and

$$f(x_0,\ldots,x_{N-1}) = \Psi(x_0,0)\prod_{n=1}^{N} W(x_n),$$

such that we can make the above approximation with I and $\mathcal{I}$. Note that $\mathcal{P}$ is a probability distribution. To implement the approximation we must generate sets of coordinate vectors $x^{(i)} \in \mathcal{P}$ (where $x^{(i)} = (x_0^{(i)},\ldots,x_N^{(i)})$ for $i = 1,\ldots,\mathcal{N}$ and where $x_N^{(i)} = x$), which sample the probability density $\mathcal{P}$. To do this, first generate for fixed $x = x_N^{(i)}$ a Gaussian random number $x_{N-1}^{(i)}$ with mean value $x_N^{(i)}$ and variance σ according to the probability density in the path integral $P\left(x_N^{(i)}, x_{N-1}^{(i)}\right)$. Then generate another random number $x_{N-2}^{(i)}$ with mean $x_{N-1}^{(i)}$ and variance σ according to $P\left(x_N^{(i)}, x_{N-2}^{(i)}\right)$. Two consecutive random numbers are related by $x_n^{(i)} = x_{n-1}^{(i)} + \sigma\rho_n^{(i)}$, where $\rho_n^{(i)}$ is a Gaussian random number with mean of zero and a variance of one. These are generated numerically by random number generators. This equation generates the coordinate vectors we wanted, $\{x_{N-1}^{(i)},\ldots,x_0^{(i)}\}$, and one can check through this equation and the definition of P in the path integral, that it gives us the required means and variances for the coordinate vectors. Therefore, they are distributed according to our probability density $\mathcal{P}$. Repeating the sampling of $\mathcal{P}$, $\mathcal{N}$ times, the wave function $\Psi(x,\tau)$ can be determined from the path integral formula and the approximation using $\mathcal{I}$. This only provides a framework for calculating the wave function for a chosen time, and what we want is a description of it for a continuous evolution $\tau \to \infty$.

However, the Monte Carlo algorithm can be written to simultaneously determine E_0 and ϕ_0. We will identify the wave function itself as a probability density (enforcing that it be positive definite). We will sample the initial wave function $\Psi(x,0)$ at $\mathcal{N}_0$ points, and generate random walks evolving according to $\mathcal{P}$ and follow the motion of the ensemble. We will sample the wave function of the system through the position of the random walks and the weights W after each time step. This will allow us to improve the reference energy after each time step. It also allows enough time steps until convergence of the wave function and energy to those of the ground state. This is known as the DMC method. It interprets the path integral formula as

$$W(x_N)P(x_N, X_{N-1})\ldots W(x_2)P(x_2,x_1) \times W(x_1)P(x_1,x_0)\Psi(x_0,0),$$

as stochastic processes 2N to 0. The 0^{th} process describes particles (random walks) distributed according to the initial wave function $\Psi(x_0,0)$, typically taken to be $\Psi(x_0,0) = \delta(x - x_0)$, with x_0 in a region where the ground state is expected to be large. This initial distribution places all particles at initial position x_0. Recall from earlier that successive positions are determined by $x_n = x_{n-1} + \sigma\rho_n$ by generating a series of random numbers ρ_n. This is known as the diffusive displacement.

It is more efficient to replicate particles after each time step with a probability $\propto W(x_n)$ rather than the product of weight factors for each particle. By this method, the wave function is given by a discrete spatial distribution of particles after each time step. Calculating the wave function this way is a simulated diffusion-reaction process for replicas. In this process we replace a particle by m_n particles given by

$$m_n = \min[\text{int}[W(x_n) + u], 3],$$

where u is a random number uniformly distributed in $[0,1]$. If $m = 0$ the particle is deleted and this is known as the 'death' of a particle. This stops the diffusion process. If $m = 1$ the particle is unaffected and the diffusion process continues, if $m_n = 2$ it continues but a new series is born starting at the present location x_n. If $m = 3$ the diffusion process continues but two new series are born. At most 2 new particles can be generated. This limitation is imposed on our process to avoid numerical instabilities which arise particularly at the beginning of the simulation where $|E_R - E_0|$ is large. For sufficiently small time steps $\Delta\tau$ we have [2]

$$W(x) \approx 1 - \frac{V(x) - E_R}{\hbar}\Delta\tau,$$

which has values near one. Therefore the error resulting in $m_n \leq 3$ is small.

2.4 The Diffusion Monte Carlo Method

Goal: Determine the ground state energy and the ground state wave function.

The standard Monte Carlo method will sample the wave function after each time step. The wave function can be approximated at each time step by the spatial distribution of the replicas. If the origin of the energy scale is equal to the ground state energy, this distribution will converge to the ground state energy. We will begin with a reasonable guess of the ground state energy and we will improve this estimate at each time step. This estimate will converge to the ground state energy.

We start with a number $\mathcal{N}_0$ particles ("replicas") placed according to $\Psi(x_0,0)$. The number of replicas varies with the birth-death process. These replicas are characterised by position $x_n^{(j)}$ where n counts diffusive displacements and (j) counts the replicas. $\mathcal{N}_n$ denotes the number of replicas after n displacements. The first step is to assign positions $x_0^{(i)}$ to all be at the same point x_0. To follow all of the diffusive displacements simultaneously, set

$$x_1^{(j)} = x_0^{(j)} + \sqrt{\frac{\hbar\Delta\tau}{m}} \cdot \rho_1^{(j)},$$

generating appropriate random numbers $\rho_1^{(j)}$. After this one-step diffusion process evaluate the weight $W(x_1^{(j)})$ and use this to determine the integers $m_1^{(j)}$. If a replica i has $m_1^i = 0$ it is terminated. If $m_1^i = 1$ it is unaffected, if $m_1^i = 2, 3$ then one or two new replicas i' are added into the system with position $x_1^{i'} = x_1^i$. Then the new number of replicas $\mathcal{N}_1$ is determined. We identify the new distribution of coordinates as $\Psi(x, \Delta\tau)$.

Recall that only $E_R = E_0$ will admit a stable asymptotic distribution of the number of replicas fluctuating around an average $\mathcal{N}_0$. We adjust the value of the reference energy E_R with the intention of stabilising the number of replicas. Averaging over all replicas we have

$$\langle W\rangle_1 \approx 1 - \frac{\langle V\rangle_1 - E_R}{\hbar}\Delta\tau,$$

with

$$\langle V\rangle_1 = \frac{1}{\mathcal{N}_1}\sum_{j=1}^{\mathcal{N}_1} V(x_1^{(j)}).$$

This is specifically after the one-step diffusion process. The goal is that eventually $\langle W\rangle$ is always one to give us a constant number of replicas. This leads us to choose $E_R^{(1)} = \langle V\rangle_1$.
However in large deviations of the number of replica we run into issues. In the case of $\frac{\mathcal{N}_1}{\mathcal{N}_0} < 1$ we want to increase the number of replicas to restore the initial number (choose a bigger E_R) and in the case of $\frac{\mathcal{N}_1}{\mathcal{N}_0} > 1$ we want to decrease the number of replicas (choose a smaller E_R). So in the next step a suitable choice is given by

$$E_R^{(2)} = \langle V\rangle_1 + \frac{\hbar}{\Delta\tau}\left(1 - \frac{\mathcal{N}_1}{\mathcal{N}_0}\right).$$

Note that for sufficiently small $\Delta\tau$ we have

$$\exp[-\Delta\tau\frac{(\langle V\rangle - E_R)}{\hbar}] \approx 1 - \Delta\tau\frac{(\langle V\rangle - E_R)}{\hbar},$$

which leads to this choice of E_R. Our choice of $E_R^{(1)}$ allows us to write

$$E_R^{(2)} = E_R^{(1)} + \frac{\hbar}{\Delta\tau}\left(1 - \frac{\mathcal{N}_1}{\mathcal{N}_0}\right).$$

$\frac{\hbar}{\Delta\tau}$ is the feedback parameter for this DMC problem, and as with all probems of this form, one must empirically choose the parameter so as to reduce as much of the statistical fluctuations in $\mathcal{N}_0$ and also diminish correlations between successive replicas.
The entire process (diffusion, births, deaths and estimates of reference energy) is repeated until the distribution of the number of replicas becomes stationary. The ground state energy is then given by

$$E_0 = \lim_{n\to\infty}\langle V\rangle_n,$$

as

$$\frac{\mathcal{N}_n}{\mathcal{N}_{n-1}} \to 1.$$

The distribution of the replicas is identified as the ground state wave function $\phi_0(x)$.

2.5 Systems with Several Degrees of Freedom

2.5.1 d dimensions

The Hamiltonian of a particle of mass m moving in d dimensions through a potential $V(x_1, \ldots, x_d)$ is

$$\hat{H} = -\frac{\hbar^2}{2m}\sum_{\alpha=1}^{d}\frac{\partial^2}{\partial x_\alpha^2} + V(x_1, \ldots, x_d),$$

where x_α denotes the Cartesian coordinates of the particle. The only difference in the DMC algorithm for this case in comparison to the 1 degree of freedom (DOF) case is that during each time step d random walks must be executed per replica. Therefore the distribution which governs diffusive displacements is now given by [2]

$$P(x_{n,1}, x_{n-1,1}; \ldots; x_{n,d}, x_{n-1,d}) = \prod_{\alpha=1}^{d}\left(\frac{m}{2\pi\hbar\Delta\tau}\right)^{\frac{1}{2}}\exp\left[-\frac{m(x_{n,\alpha} - x_{n-1,\alpha})^2}{2\hbar\Delta\tau}\right].$$

A product of probabilities is described by independent random processes.

2.5.2 S particles

The most general Hamiltonian for S particles with no internal DOF is given

$$\hat{H} = \sum_{j=1}^{S} \left[-\frac{\hbar^2}{2m_j} \sum_{\alpha=1}^{d} \frac{\partial^2}{\partial^2_{j\alpha}} + V(\{x_j \alpha\}) \right],$$

where the potential accounts for interactions between particles. One may rescale the coordinates by setting $x_{j\alpha} = \sqrt{\frac{m}{m_j}} x_{j\alpha}$, $j = 1, \ldots, S$, $\alpha = 1, \ldots, d$ where m is an arbitrary mass. This allows the Hamiltonian to take the form of the single particle Hamiltonian moving in $d' = S \times d$ dimensions.

However in the case of identical particles one must obey a prescribed boson or fermion symmetry of the wave function (symmetric/anti-symmetric under exchange of particles).

2.6 Dimensionless Units

We need to rewrite all equations with dimensionless units before implementing a numerical algorithm. We will relabel everything in terms of 3 indepdent variables: length L, time T and energy $\mathcal{E}$. We will relabel quantities times their dimension to the name of the quantity itself, i.e.$xL \to x, \tau T \to \tau, E\mathcal{E} \to$. Then we will rewrite the TDSE under this relabeling as

$$\frac{\partial \Psi}{\partial \tau} = \frac{\hbar T}{2mL^2} \frac{\partial^2 \Psi}{\partial x^2} - \frac{T\mathcal{E}}{\hbar} [V(x) - E_R] \Psi.$$

For convenience, let us choose our independent variables to give $\frac{\hbar T}{2mL^2} = \frac{1}{2}$ and $\frac{T\mathcal{E}}{\hbar} = 1$. The three independent variables are governed by two independent equations, this means that they form an undetermined system. After this is imposed the Schrödinger equations becomes

$$\frac{\partial \Psi}{\partial \tau} = \frac{1}{2} \frac{\partial^2 \Psi}{\partial x^2} - [V(x) - E_R] \Psi.$$

By the same method we obtain the probability distribution, the weights and as a result the diffusive displacements.

$$P(x_n, x_{n-1}) = \sqrt{\frac{1}{2\pi \Delta\tau}} \exp\left[-\frac{(x_n - x_{n-1})^2}{2\Delta\tau} \right],$$

$$W(x_n) = \exp\{-V(x_n) - E_R]\},$$

$$x_n = x_{n-1} + \sqrt{\Delta\tau} \rho_n.$$

2.7 Limitations of the DMC method

Due to the Pauli exclusion principle, the ground state wavefunction for many-fermion systems contains nodes, for example the Helium atom. DMC methods rely on fixed-node approximations which is not truly reflective of the wavefunction, therefore the presence of nodes may introduce errors in the results.

Monte carlo methods can be computationally costly, despite their simplicity. This is especially true when there is a high runtime per time step. However, this method has a very parallel nature and lends itself to be improved using parallel computing [9].

2.8 Current Research

One up and coming area of research is quantum computing quantum Monte Carlo (QC-QMC). This invloves the trial wavefunction being built with quantum gates in a quantum circuit. Results using QC-QMC can be used to find more accurate solutions than simulations performed on classical computers. Quantum algorithms to exploit this higher-accuracy are currently being researched and used to solve quantum chemistry problems [10].

Monte Carlo simulations are numerical methods therefore they can be easily employed by artificial intelligence to explore larger data sets. For example, this paper ([11]) explored the use of neural networks and Quantum Monte Carlo methods to solve the quantum many-body problem. Specifically, static variational Monte Carlo sampling was used. This is an example of a zero-temperature method.

There are many open source QMC packages available now, allowing researchers across fields to solve problems in quantum systems. There is a large ongoing effort to make these resources available as there are many problems in quantum chemistry, for example, which can be solved with the use of quantum Monte Carlo methods. One example is the QMCPACK for electronic structure calculations [12]. QMCPACK uses highly sophisticated software including the orbital space auxiliary-field quantum Monte Carlo method and Slater–Jastrow type trial wavefunctions. QMCPACK allows this software to be accessible globally.

3 Computer Program

This chapter will be a description of the computer program implemented on Python to obtain the results in this project. Below is a diagramatic scheme for the program followed by explanations of each step, alongside code segments.

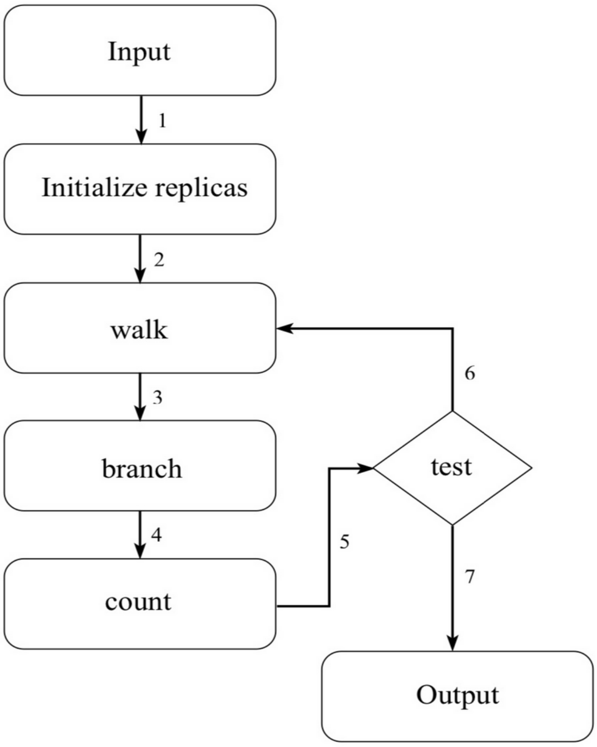

Figure 1: Diagramatic Scheme for the DMC Computer Program [2]

3.1 Input

Beginning with the program's input, this contains all the external data required by the program. This includes the initial and maximum number of replicas $\mathcal{N}_0, \mathcal{N}_{max}$, the number of time steps τ_0, the size of each time step $\Delta\tau$, x_{min} and x_{max} for the spatial sampling of replicas and the value n_b which defines the size of the portions over which the sampling space is divided. To define the quantum system one must also input the potential energy and the dimensionality d' which accounts for all of the degrees of freedom (DOF) of the system. Below is an example of an input library containing dictionaries of inputs for various quantum systems.

```
import numpy as np

def harmonic_oscillator_potential(x):
    return 0.5 * (x ** 2)

def morse_potential(x, w_d=1, L=1, x_e=1):
    # Morse parameters: w_d (well depth), L (width), x_e (equilibrium position)
    return w_d * (1 - np.exp(-L * (x - x_e)))**2


systems = {
    "harmonic_oscillator": {
        "dprime": 1,
        "Nmax": 10000,
        "N0": 200,
        "initial_position": 0.5,
        "delt": 0.1,
        "t0": 30,
        "potential": harmonic_oscillator_potential
        "xmin": -20,
        "xmax": 20,
        "nb": 100 #number of bins
    },
    "morse_potential": {
        "dprime": 1,
        "Nmax": 5000,
        "N0": 150,
        "initial_position": 1.0,
        "delt": 0.05,
        "t0": 50,
        "potential": morse_potential
        "xmin": -20,
        "xmax": 20,
        "nb": 100
    }
}
```

3.2 Initialisation

The initialisation section of the program requires a two-dimensional matrix called “psips”. The first index of the matrix, indicating the row of the relevant entry, is a value of an integer in the interval between 0 and $\mathcal{N}_{max}$. This indicates the replica in question. The second index of the matrix, indicating the column, is an integer between 0 and d’+1. The first column of the matrix psips[:][0] labels the relevant replica with increasing integers. The second column of the psips matrix indicates whether a replica exists. If the entry psips[i][1] = 0 the replica i is dead, and if psips[i][1] = 1 the replica is alive. Columns 2 to d’+1 store the coordinates of the replica throughout the simulation. The psips matrix should be initialised with entries psips[j][0] for j between 1 and $\mathcal{N}_0$ to initialise the existence of the starting replicas. Replicas $j > \mathcal{N}_0$ are not yet born so these have entries psips[j][0] = 0 to start with. Then the coordinates (the rest of the matrix) should be assigned the same values (depending on the system at hand) so that the distribution is given by a delta function. Recall from the theory that the initial value of the reference energy should be chosen as the potential.

Then the program goes through a loop, each iteration being equivalent to a time step $\Delta\tau$. The loop consists of the walk, branch and count steps of the program. Below is a code segment initialising the reference energy and psips matrix for a system of any dimensionality.

```
Er = potential(initial_position)

psips = np.zeros([Nmax, dprime+2])
psips[:, 0] = np.arange(Nmax) #labels rows with integers 0 to Nmax
psips[:N0, 1] = 1
psips[:N0, 2] = Er
```

3.3 Walk

The walk portion performs the diffusive placements by applying $x_n = x_{n-1} + \sqrt{\Delta\tau}\rho_n$ to the alive replicas for a random number ρ taken from a Gaussian distribution with mean zero and standard deviation one. Recall this is the formula for the Brownian diffusion equation.

Below is a code segment defining the walk function on a matrix:

```
def walk(matrix): #takes a row where the existence flag is 1 and adds a random number to the coordinate
    rho = np.sqrt(delt)*np.random.normal(0, 1, matrix.shape[0]) #random array of corrrect dimension
    rows_to_update = matrix[:, 1] == 1 #identifies list of alive replicas
    matrix[rows_to_update, 2] += rho[rows_to_update]
    return matrix
```

3.4 Branch

The branch portion applies the next part of the birth-death process. In this process we replace an alive replica by m_n replicas given by $m_n = \min[\text{int}[W(x_n)+u], 3]$ where where u is a random number uniformly distributed in $[0, 1]$. If $m_n = 0$ then the existence flag in the psips matrix is set to zero. Nothing happens if $m_n = 1$. If $m_n = 2$ then the first inactive replica in psips is set to have existence flag one and coordinates the same as the original replica. If $m_n = 3$, this is done for the first two inactive replicas. This branching factor m_n is limited to integers up to and including three, to avoid a hugely rapid increase in the number of replicas at the beginning of the program. The following code segment defines the weight W, the number m_n and the branching function for a matrix, where the reference energy is given:

```
def weight(x, energy):
    return np.exp(-(potential(x) - energy))

def branch_amount(x, energy):
    u = random.uniform(0, 1)
    w = weight(x, energy)
    mn = min (int(w +u),3)
    return mn

def branch(matrix, energy):
    zero_rows = np.where(matrix[:, 1] == 0)[0]
    for i in range(matrix.shape[0]):
        if matrix[i, 1] == 1:
            mn = branch_amount(matrix[i, 2], energy)

            if mn == 0:
                matrix[i, 1] = 0

            elif mn == 1:
                continue

            elif mn == 2:
                if len(zero_rows) >= 1:  #at least 1 row available
                    matrix[zero_rows[0], 1:] = matrix[i, 1:]  #copy 1
                        coordinate
                    matrix[zero_rows[0], 1] = 1  #set alive
                    zero_rows = zero_rows[1:]  #get rid of this row as
                        identified as zero because now it's not


            elif mn == 3:
                if len(zero_rows) >= 2:  # at least 2 rows available
                    for j in range(2):
                        matrix[zero_rows[j], 1:] = matrix[i, 1:]
                        matrix[zero_rows[j], 1] = 1
                    zero_rows = zero_rows[2:]


    return matrix
```

3.5 Count

The count portion returns the ground state wave function (the distribution of replicas). The spatial interval (x_{min}, x_{max}) is divided into n_b sub-intervals for each DOF. The count starts after τ_0 time steps when the system has reached a stationary state (seen by convergent $\langle V \rangle_\tau$). The count is then performed cumulatively for τ_0 more time steps. Cumulative counting yields better statistics than sampling the replicas at one instant in time. This is because the sample size is effectively multiplied by τ_0. After this process the ground state wave function must be normalised:

$$\phi_0(x_i) \approx \frac{N_i}{\sqrt{\sum_{i=1}^{n_b} N_i^2}}$$

where $i = 1, \ldots, n_b$.
The following code segment creates a distribution of the replicas into bins and then normalises the distribution, which will later be used to return the ground state wavefunction at the end of the process.

```python
def distribution(replicas, subintervals):
    counts, _ = np.histogram(replicas, bins=subintervals)
    return counts

coordinate = stabilised_psips[1]

def normalize_distribution(replica_counts, subintervals):
    bin_width = subintervals[1] - subintervals[0]
    total_replicas = np.sum(replica_counts)

    if total_replicas == 0: #to avoid dividing by zero
        print("No replicas to normalise")
        probability_density = np.zeros_like(replica_counts)
        wave_function = np.zeros_like(replica_counts) #returns a zero wavefunction instead of an error
    else:
        probability_density = replica_counts / (total_replicas * bin_width)
        wave_function = np.sqrt(probability_density)

    return probability_density, wave_function
```

3.6 Test

The test portion ensures that the walk and branch routines are called $2\tau_0$ times, but the count branch is only called τ_0 times for the second half of the calculation.

3.7 Output

The output of the program returns the results. The possible set of results we may obtain are the mean $\langle E_R \rangle \approx E_0$, the standard deviation δE_R, the (imaginary) time evolution of $\langle E_R \rangle$ for the first τ_0 steps (in order to see how fast stationarity is reached) and the normalised distribution of the replicas which is the ground state wave function, this wavefunction is our main focus in this project. Recall, the output of the average reference energy after n times steps is given by

$$\langle E_R(\tau = n\Delta\tau) \rangle = \frac{1}{n} \sum_{i=1}^{n} E_R(i\Delta\tau).$$

4 Examples and Results

4.1 Harmonic Oscillator

The one-dimensional, one particle harmonic oscillator is a great example to discuss as it is simple and also has an exactly solvable solution. The potential in its Hamiltonian describes a restorative force. This has the effect that in random trials walkers remain localised so the method is not vulnerable to divergences causes by delocalised walkers.

Analytical Solution of the Harmonic Oscillator

Consider Schrodinger's equation for the quantum oscillator Hamiltonian:

$$\frac{1}{2m}\left[\hat{p}^2 + (m\omega\hat{x})^2\right]\psi = E\psi.$$

Now, consider creation and annihilation operators

$$\hat{a}_{\pm} = \frac{1}{\sqrt{2m\hbar m\omega}}(\mp i\hat{p} + m\omega\hat{x}).$$

We may then rewrite the Schrodinger equation as

$$\hbar\omega\left(\hat{a}_{\pm}\hat{a}_{\mp} + \frac{1}{2}\right).$$

These are considered ladder operators for the $\mathfrak{su}(2)$ algebra. Therefore there is a highest and lowest weight vector. Let ψ_0 be defined so that $\hat{a}_-\psi_0 = 0$. Working this out using the definition of the annhihilation operator we find that it must be of the form

$$\psi_0(x) = Ae^{-\frac{m\omega}{2\hbar}x^2}.$$

Normalising this we are left with the normalised ground state wavefunction

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

We determine the ground state energy by plugging this into the Schrodinger equation and obtain $E_0 = \frac{1}{2}\hbar\omega$. See a plot of the ground state wavefunction below in figure 3, compared against the DMC solution [1].

Inputs

Each of the replicas are initialised to coordinate 0, since the trial wavefunction in this case is $\psi(x) = \delta(x)$. This trial wavefunctoin is used in cases where the wavefunction is expected to be large at x = 0. The dimension is $d' = 1$. We form the conventional dimensionless quantities: $L = \sqrt{\frac{\hbar}{m\omega}}$, $E = \hbar\omega$ and $T = \frac{1}{\omega}$ we have the dimensionless $\xi = \frac{x}{L}$ and then we can write the dimensionless potential energy $V(\xi) = \frac{1}{2}\xi^2$. From this point on, all potentials mentioned are understood to be dimensionless. Also note that we are using natural units so $\hbar = 1$ etc.

Results for Harmonic Oscillator DMC method

The ground state energy was found to be $E_0 = 5.508$ with standard deviation $\delta E_R = 0.0145$. The potential energy stabilised after 8 time steps. Each time step was 0.1 in this simulation. The simulation was ultimately run over 30 time steps after stationarity was reached. The actual runtime was 17.6 seconds.
See below (fig.2) an example of a stabilisation graph for this Harmonic Oscillator. See below (fig.1) a plot of the DMC solution found for ground state wavefunction the harmonic oscillator compared to the analytic solution. You can see that the solution is quite accurate. There are sum statistical inaccuracies particularly at the top of the wavefunction. These are expected as DMC is a stochastic method, but they appear where the wavefunction is large due to the larger number of walkers in this region.

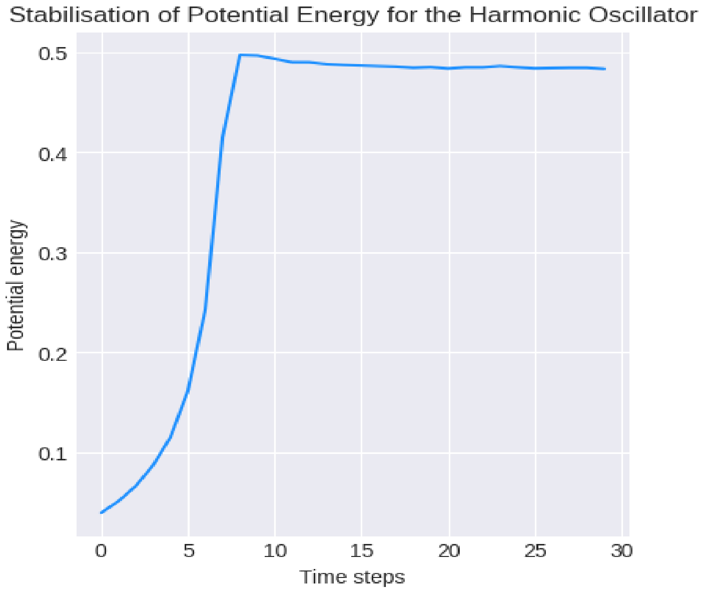

Figure 2: The plot of the Stabilisation of Potential Energy in the DMC Simulation of the 1D Harmonic Oscillator over a Number of Time Steps.

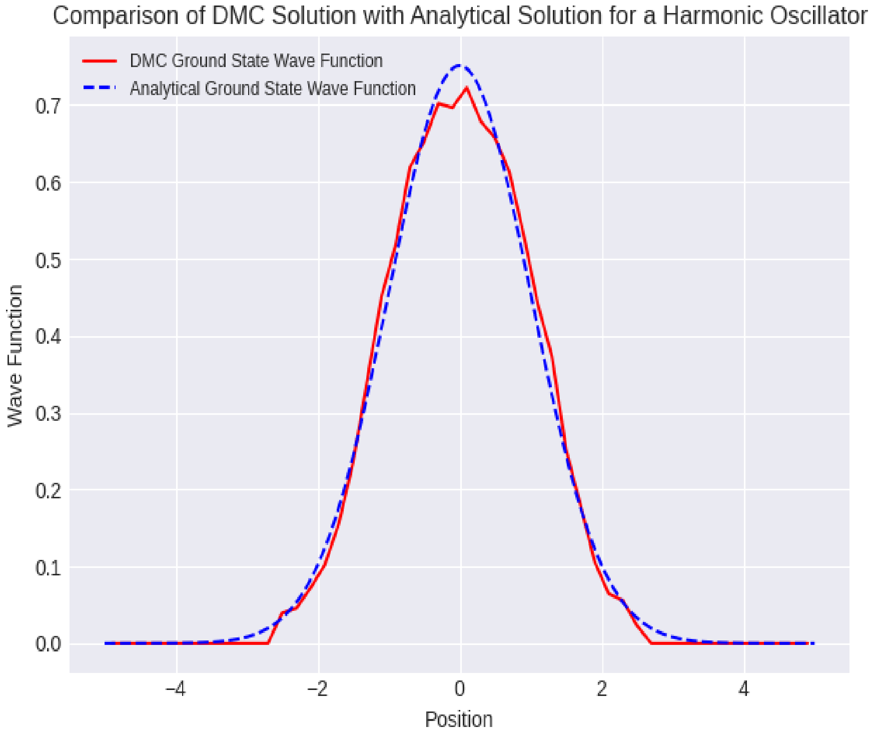

Figure 3: DMC Solution vs Analytical Solution for Ground State of a Harmonic Oscillation

4.2 Morse Oscillator

The morse oscillator can be used to model a 1D anharmonic potential with a dissociative limit it infinity [13]. It can be seen most commonly in spectroscopy calculations and vibrational dynamics [14].

Analytical Solution to the Wavefunction

The potential energy of the Morse oscillator is

$$V(x) = D_e[1 - e^{(1-a(x-x_e)}]^2.$$

Its Hamiltonian has eigenvalues

$$E_n = \hbar\omega_0 \left[(n + \frac{1}{2}) - x_e(n + \frac{1}{2})^2 \right],$$

where $\omega_0 = \frac{4x_e D_e}{\hbar}$ is the fundamental frequency. The wavefunctions for the Morse oscillator are expressed in terms of Laguerre polynomials [15] :

$$\psi_n = N_n e^{-\frac{z}{2}} z^{\frac{b}{2}} \mathcal{L}_n^b(x),$$

where $N_n = \left[\frac{a \cdot b \cdot n!}{\Gamma(k-n)}\right]^{\frac{1}{2}}$, $z = k\exp[-aq]$, $b = k - 2n - 1$ and $k = \frac{4D_e}{\hbar\omega_0}$.

Results for Morse Oscillator

The potential energy of the Morse oscillator is $V(x) = D_e[1 - e^{(1-a(x-x_e)}]^2$ where D_e is the depth of the well, a is the width of the well and x_e is the equilibrium bond distance. For this simulation the Morse parameters were chosen to be D_e = 1.0, a= 1.0 and x_e = 0.0. The trial wavefunction for the Morse oscillator was $\psi(x) = \delta(x - 0.5)$.
The ground state energy was found to be E_0 = -0.119 with standard deviation δE = 0.017. The potential energy converged after 10 time steps. Each time step for the simulation was 0.1. The program was run for 30 time steps after the stationarity was reached. The actual runtime was 19.6 seconds. See below (fig.3) the results of the DMC method for finding the ground state of the wavefunction.

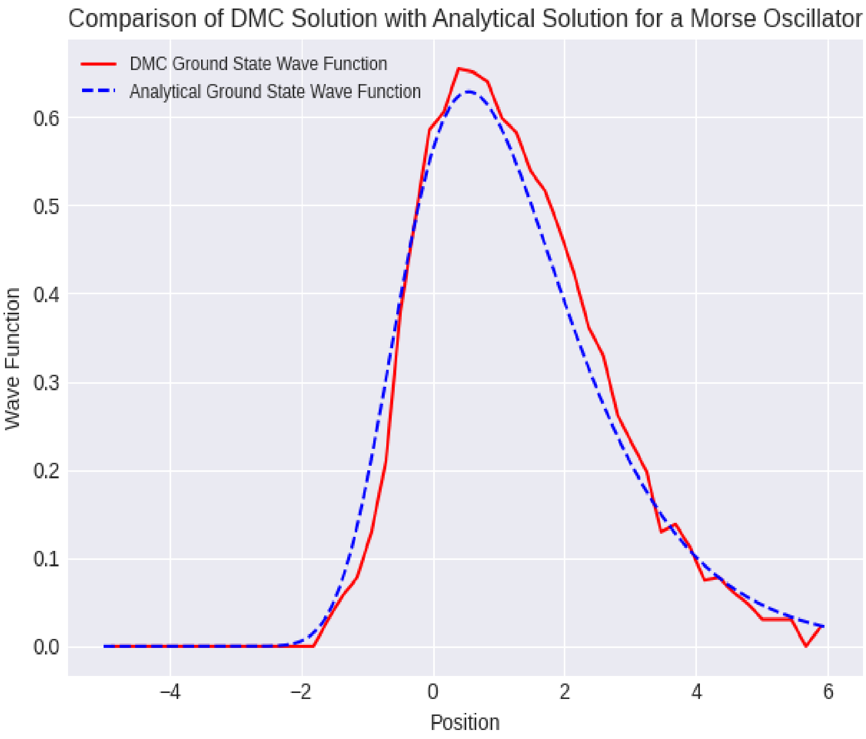

Figure 4: DMC Solution vs Analytical Solution for Ground State of a Morse Oscillation

4.3 Harmonic Oscillator with Quartic Anharmonic Perturbation

We will use this example to study a perturbative system, specifically to find the ground state energy of the system. We begin with a Hamiltonian [1]:

$$H_0 = \frac{1}{2}(\hat{p} + m^2\omega^2x^2) + \lambda g x^4,$$

for some dimensionless parameter λ. We rewrite this to make it dimensionless: $H = H_0 + \lambda g x^4$. The shift in the nth level energy is

$$\Delta E_n = \lambda g \langle \psi_n | x^4 | \psi_n \rangle.$$

Since we have ψ_n, the normalised ground state wavefunctions for a harmonic oscillator already, we can find that

$$\Delta E_0 = \lambda g \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} \int_{-\infty}^{\infty} x^4 e^{\frac{-m\omega x^2}{\hbar}} dx.$$

This is clearly an example of a Gaussian integral, so we obtain the result

$$\Delta E_0 = \frac{3}{16}\left(\frac{\omega\hbar}{2}\right)\lambda.$$

Results for Anharmonic Oscillator

The potential energy for this kind of oscillator is given by $\frac{1}{2}x^2 + \lambda x^4$ for an anharmonicity parameter λ. For this simulation it was chosen to be $\lambda = 0.5$. The trial wavefunction was $\psi(x) = \delta(x)$.

The ground state energy was found to be $E_0 = 0.538$ with standard deviation $\delta E = 0.017$. Using the analytical solution the energy was found to be 5.737. This is outside of 1 standard deviation from the simulation result. This can be improved using more sophisticated methods, for example in importance sampling or in dynamic timestep error control.

The length of each time step was 0.1. The potential energy converged after 9 time steps and after this the program was run for a further 30 time steps. The actual runtime was 22.0 seconds. See below the solution for the ground state wavefunction (fig.5).

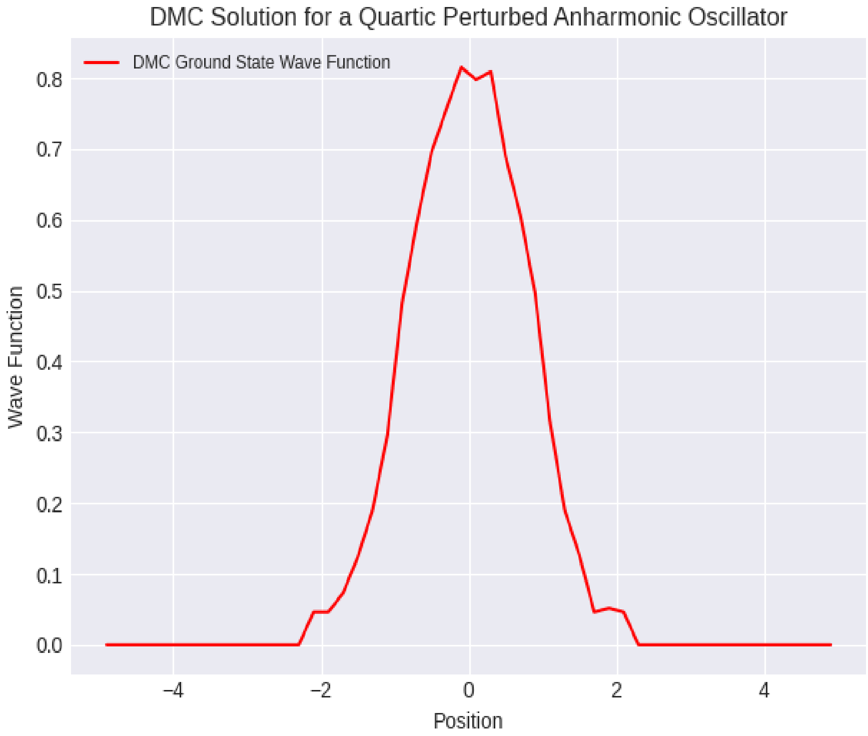

Figure 5: DMC Solution vs Analytical Solution for Ground State of an Anharmonic Oscillator

4.4 Particle in a Well

Systems involving a particle in a well are interesting to explore as they involve discontinuous potentials. The infinite sqaire well potential is commonly used to model molecular orbitals in quantum chemistry, and more recently is being used in nano-technology to model quantum dots [16].

Analytical Solution for the Infinite Square Well

Consider the potential

$$V(x) = \begin{cases} \infty & -a < x \text{ or } x > a \\ 0 & -a < x < a \end{cases}.$$

Outside the well the probability of finding the particle is zero and inside the well we have the Schrodinger equation:

$$\frac{d^2\psi}{dx^2} = -k^2\psi,$$

where $k \equiv \frac{\sqrt{2mE}}{\hbar}$. This is harmonic oscillator solution. One can impose an ansatz solution:

$$\psi(x) = A \sin kx + B \cos kx.$$

One can also impose boundary conditions such that $\psi(-a) = \psi(a) = 0$. After determining the normalisation constant $A = \sqrt{\frac{2}{a}}$ we are left with

$$\psi(x) = \sqrt{\frac{2}{a}} sin\left(\frac{n\pi}{a}x\right),$$

for $n \in \mathbb{N}$ [1]. This solution is plotted in figure 5 alongside the DMC results for the infinite square well and in figure 6 alongside the DMC results for finite square wells.

Results for Infinite and Finite Square Wells

The potential of a particle in an infinite well square is given by $V(x) = \begin{cases} \infty & -a < x \text{ or } x > a \\ 0 & -a < x < a \end{cases}$, where the total length of the box is $L = 2a$. For this simulation the parameter was chosen to be $a = 7$.

The ground state energy was found to be 0.324, with standard deviation 0.080. The potential stabilised after 5 time steps with each time step being of length 0.1. The program was then run for 30 more time steps after stationarity was reached. The actual runtime was 33.8 seconds. Below is a plot (fig.6) of the ground state wavefunction found by DMC in comparison to the analytical solution. One may also examine the case of the finite well where the potential becomes some number V_0 outside of the box instead of ∞. Examine the plot below (fig. 7) which compares the ground state wavefunction of different depths of well against the analytic solution for the infinite well. One can see that even though the finite well system is a lot more difficult to solve, this finite potentials which are quite small lead to a similar shape in wavefunction. However, there is still a clear variation from the infinite well solution.

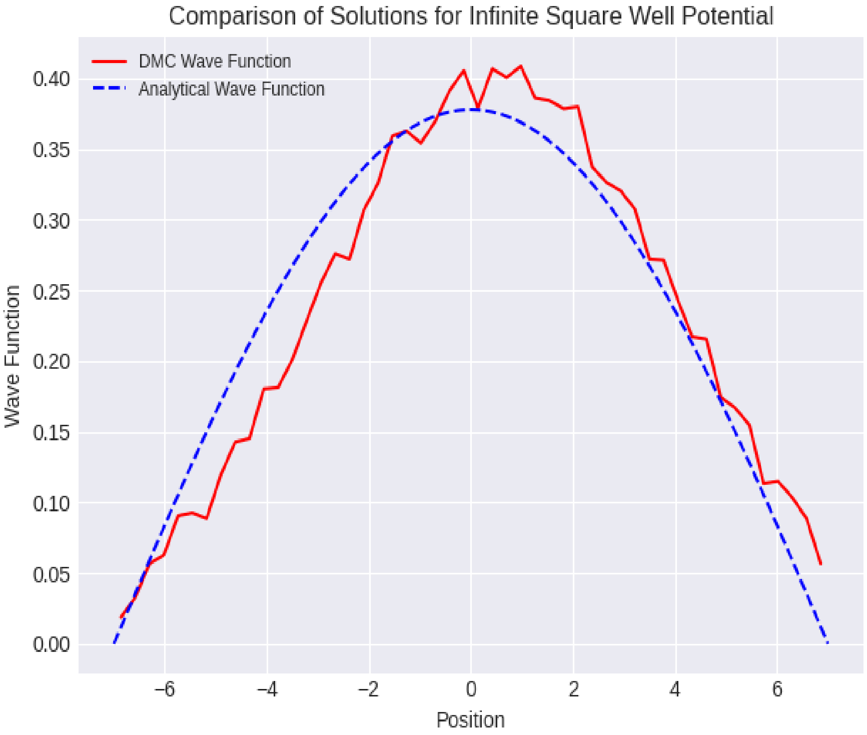

Figure 6: DMC Solution vs Analytical Solution for Ground State of an Infinite Square Well System

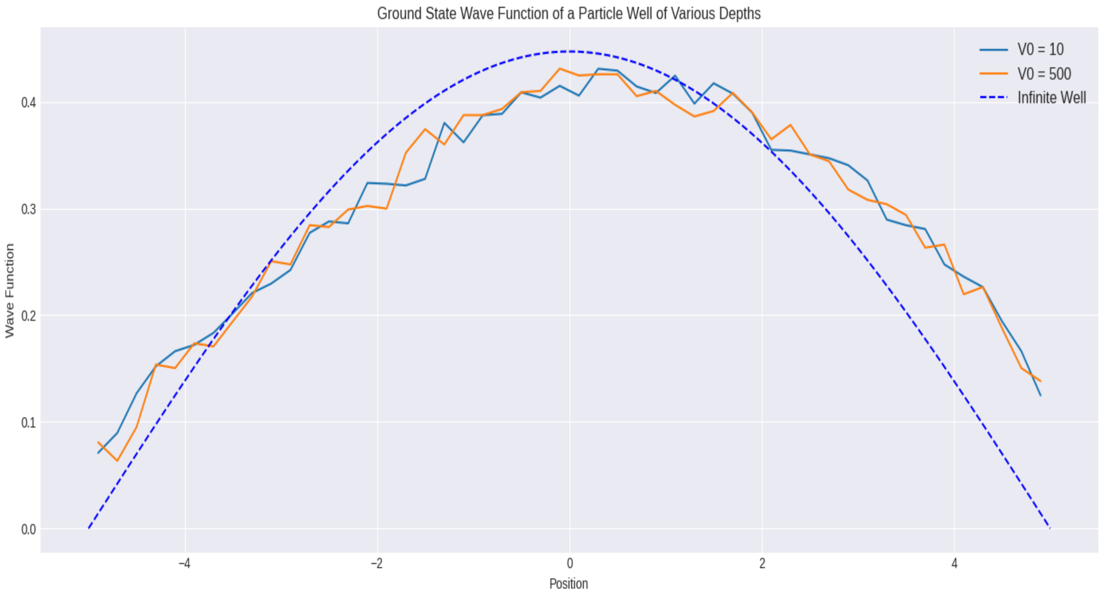

Figure 7: DMC Solution for a Finite Square Well vs Analytical Solution for Ground an Infinite Square Well System

4.5 Hydrogen atom

Analytical Solution of the Hydrogen Atom

The wavefunction of the Hydrogen atom can be written in terms of Laguerre polynomials $L_q(x) \equiv \frac{e^x}{q!}\left(\frac{d}{dx}\right)^q (e^{-x}x^q)$ and spherical harmonics $Y_l^m(\theta,\phi)$ as [1]

$$\psi_{nlm} = \sqrt{\left(\frac{2}{na}\right)\frac{(n-l-1)!}{2n(n+l)!}}e^{-\frac{r}{na}}\left(\frac{2r}{na}\right)^l\left[L_{n-l-1}^{2l+1}(\frac{2r}{na})\right]Y_l^m(\theta,\phi).$$

where $a \equiv \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = 0.529\times10^{-10}m$ in SI units. The radial part of the ground state wavefunction for Hydrogen is then

$$\phi(r) = \frac{1}{\sqrt{\pi a^2}}e^{-\frac{r}{a}}.$$

This can be seen plotted in figure 7 against the DMC solution for the ground state wavefunction of Hydrogen.

Results for the Hydrogen atom

The Hydrogen atom is a 1-dimensional system as it consists of 2 particles (a proton and an electron) under a central potential $V = V(|x|)$. Therefore the there is 1 degree of freedom given by the radial distance between the particles, $r = |x|$. The potential in question is the Coulomb potential so $V(x) = -\frac{1}{|x|}$. The trial wavefunction was given by $\psi(x) = \delta(x)$.
The ground state energy was found to be E_0 = -0.510 with standard deviation δE = 0.092. The length of each time step was 0.1. The potential stabilised after 11 time steps and the program was run for 30 more time steps after stationarity was reached. The actual runtume was 13.8 seconds. See below (fig 8., fig 9.) plots of the DMC and anlytical solutions to the ground state of the radial wavefunction for hydrogen. The results for DMC can be improved for Hydrogen by exploring a larger space since for a hydrogen atom, the wavefunction extends infinitely and finite cut offs can distort the results significantly. It would also be possible to improve the results using extrapolation methods in order to deal with the edge effects.

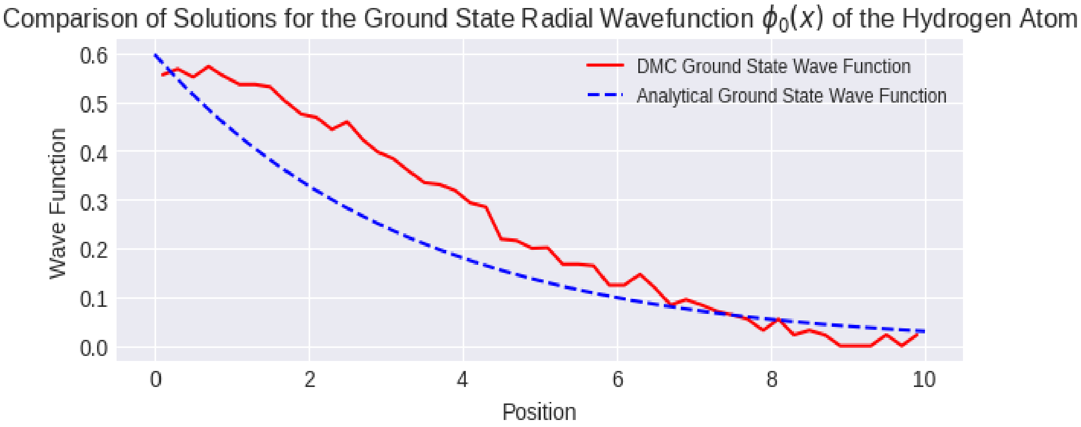

Figure 8: DMC Solution for a Finite Square Well vs Analytical Solution for $\phi_0(r)$

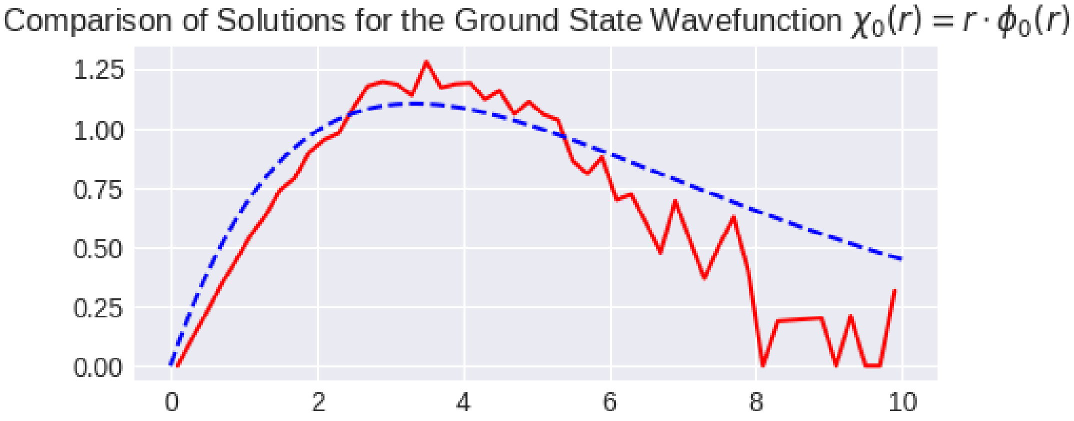

Figure 9: DMC Solution for a Finite Square Well vs Analytical Solution for a $\chi_0(r)$

4.6 Higher dimensional bosonic systems

A higher dimensional bosonic system can be used to model topological phases, lattice models like the Bose-Hubbard model or boson-phonon interactions since phonons obey Bose-Einstein statistics. This is a good example to explore cases which require a higher dimensional matrix.
The Helium atom may seem like a good example to explore DMC also, however many problems arise in fermionic systems with regards to DMC. This is due to the 'sign problem' caused by the Fermi exclusion principle (Fermi-Dirac statistics). This cause weights of configurations in the QMC simulation become negative or even complex. These solutions may be dealt with other kinds of quantum Monte Carlo methods that are designed for fermionic systems, such as the lattice determinantal Quantum Monte Carlo method [17]. However, the sign problem for a general quantum Monte Carlo method is NP-hard therefore a polynomial time solution is ‘almost certainly unattainable’[18].
A potential in this case can be a Coulomb potential, but for a two dimensional system taking two positional arguments. We consider the potential $V(x, y) = -(x^2 + y^2)^{-2}$. In this case the trial wavefunction was given by $\psi(r) = \delta(r)$.
The ground state energy was found to be $E_0 = 0.0356$ with standard deviation $\delta E = 0.004$. See below a plot (fig 9.) of the ground state wavefunction for this system.

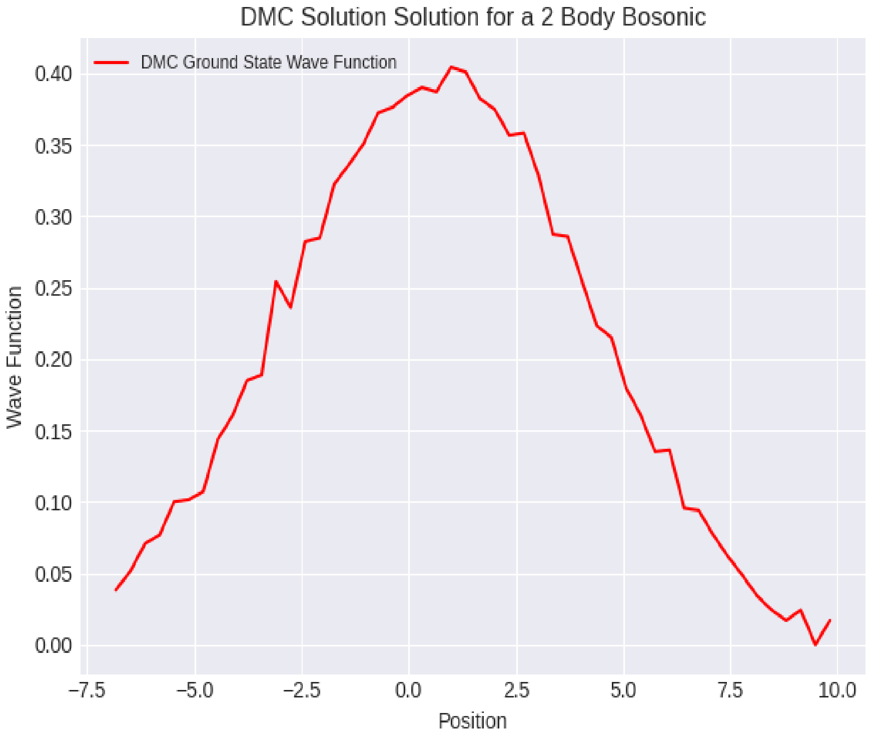

Figure 10: DMC Solution for a 2 Dimensional Bosonic System

References

[1] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. Cambridge University Press, 3 edition, 2018.

[2] Ioan Kosztin, Byron Faber, and Klaus Schulten. Introduction to the diffusion monte carlo method. *arXiv:physics/9702023*, 1997.

[3] Encyclopaedia Britannica. Monte carlo method, 2024. Accessed: 2025-01-02.

[4] G. C. Buffon. Essai d'arithmetique morale. *Supplément à l'histoire naturelle*, 4, 1777. Accessed January 3, 2025.

[5] N. Metropolis. The beginning of the monte carlo method. *Los Alamos Science Special Issue*, 1987. LQS.

[6] R.D Richtmyer E. Fermi. Note on census-taking in monte-carlo calculations. *Los Alamos - 805 Series A*, 1948.

[7] Pierre Del Moral. *Feynman-Kac Formulae*. Springer New York, New York, NY, 2004.

[8] R. P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, New York, 1965.

[9] Ronald W. Shonkwiler and Franklin Mendivil. *Introduction to Monte Carlo Methods*, pages 1–49. Springer New York, New York, NY, 2009.

[10] Shu Kanno, Hajime Nakamura, Takao Kobayashi, Shigeki Gocho, Miho Hatanaka, Naoki Yamamoto, and Qi Gao. Quantum computing quantum monte carlo with hybrid tensor network for electronic structure calculations. *npj Quantum Information*, 10(1):56, 2024.

[11] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[12] Jeongnim Kim, Andrew D Baczewski, Todd D Beaudet, Anouar Benali, M Chandler Bennett, Mark A Berrill, Nick S Blunt, Edgar Josué Landinez Borda, Michele Casula, and David M Ceperley. Qmcpack: an open source ab initio quantum monte carlo package for the electronic structure of atoms, molecules and solids. *Journal of Physics: Condensed Matter*, 30(19), 2018. Open access.

[13] P. M. Morse. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Physical Review*, 34(1):57–64, 1929.

[14] H. L. N. Lefebvre-Brion and R. W. Field. *The Spectra and Dynamics of Diatomic Molecules.* Academic Press, Boston, 2nd edition, 2004.

[15] V. S. Vasan and R. J. Cross. Matrix elements for morse oscillators. *The Journal of Chemical Physics*, 78(6):3869–3871, 1983.

[16] Debasis Bera, Lei Qian, Teng-Kuan Tseng, and Paul H. Holloway. Quantum dots and their multimodal applications: A review. *Materials (Basel)*, 3(4):2260–2345, 2010.

[17] Sergei Iskakov, Mikhail I. Katsnelson, and Alexander I. Lichtenstein. Perturbative solution of fermionic sign problem in quantum monte carlo computations. *npj Computational Materials*, 10:36, 2024. Open access.

[18] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Phys. Rev. Lett.*, 94:170201, May 2005.