Approximating the Ground State of Helium Using Diffusion Monte Carlo

Alex Hadley

July 21, 2020

1 Introduction

According to quantum mechanics, the behaviors of subatomic particles are governed by solutions to the Shrödinger equation, a partial differential equation. Exact solutions can only be found for particular potential functions, such as a square well or a harmonic oscillator. For cases in which the Shrödinger equation cannot be solved exactly, we must turn to numerical approximations in order to determine the behavior and properties of a quantum system.

One application of quantum mechanics is determining the behavior and properties of atoms and molecules. In an atom, a positively charged nucleus and one or more electrons are bound together by a Coulomb potential. For hydrogen, the Shrödinger equation can be solved exactly due to spherical symmetry; however, in helium, the second electron breaks the symmetry and the system can only be approximated.

For a given potential well, the Shrödinger equation has discrete eigenfunctions whose eigenvalues correspond to discrete energy levels. The lowest possible energy level is known as the ground state. Atoms are commonly in their ground state configuration, so the ground state energy and wave function are important to know and understand.

One common method to approximate the ground state of helium is the Hartree-Fock method, which involves iteratively solving equations that are derived by minimizing the energy of an approximate wave function [6]. However, the Hartree-Fock method does not fully consider interactions between electrons, which can lead to an error in the estimate for the ground state energy [3].

Another class of algorithms to approximate quantum systems use Monte Carlo methods. Unlike the Hartree-Fock method, quantum Monte Carlo methods have the potential to give arbitrarily accurate solutions [5]. In particular, diffusion Monte Carlo (DMC) is a method which cleverly frames the Shrödinger equation as a diffusion equation in imaginary time and uses random walks to find an approximate solution.

For quantum systems with many particles, the wave function describing the system will depend on the three position coordinates of every particle; thus, for a system of N particles, the wave function will be a $3N$ dimensional function. Solving the Shrödinger equation for many particles can therefore become increasingly complex due to the so-called "curse of dimensionality." Monte Carlo methods for computing multi-dimensional integrals scale well as the number of dimensions increases, so one might expect DMC to likewise scale well as the number of particles increases. However, DMC does not necessarily scale well, particularly for systems of fermions (particles, such as electrons, with half-integer spin, which obey the Pauli exclusion principle) [5]. The main advantage of DMC is its potential to give arbitrarily accurate solutions for the ground state and excited state energies and other properties, which can provide a useful check on other calculation procedures [5].

In this project, I will implement the diffusion Monte Carlo algorithm in MATLAB. First, I will test my implementation on a one-dimensional harmonic oscillator potential and the hydrogen atom, quantum systems for which the exact ground state is known. Then I will use the algorithm to estimate the ground state energy and wave function of helium, a system for which there is not exact solution, and compare my result to the empirical energy value.

2 Background on the Shrödinger Equation

Before we derive the diffusion Monte Carlo algorithm, some background in basic quantum mechanics is necessary. As explained in the Introduction, a quantum system is governed by the Shrödinger equation,

$$i\hbar\frac{\partial\Psi}{\partial t} = H\Psi, \tag{2.1}$$

where $\hbar$ is Plank's constant, Ψ is the wave function, and H is the Hamiltonian operator. The wave function gives us the probability density function, $|\Psi|^2$.

For a single particle in three spacial dimensions, the wave function is $\Psi(\mathbf{r}, t)$, where $\mathbf{r} = (x, y, z)$ is the position of the particle and t is time. Then the probability density function is $|\Psi(\mathbf{r}, t)|^2$, which means that the probability of detecting the particle if a measurement is conducted over the region of space between $\mathbf{r}$ and $\mathbf{r} + d\mathbf{r}$ is $|\Psi(\mathbf{r}, t)|^2 d\mathbf{r}$. Assuming that the potential $V(\mathbf{r})$ is constant in time, the Hamiltonian operator is

$$H = -\frac{\hbar}{2m}\nabla^2 + V(\mathbf{r}), \tag{2.2}$$

where m is the particle's mass and $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian.

The Shrödinger equation can easily be generalized to describe a system of multiple particles. The wave function and the potential now depend on the coordinates of all N particles in the system and can be written as $\Psi(\mathbf{r}_1, \dots \mathbf{r}_N, t)$ and $V(\mathbf{r}_1, \dots \mathbf{r}_N)$, where $\mathbf{r}_i = (x_i, y_i, z_i)$ is the position of particle i. The multi-particle Hamiltonian becomes

$$H = -\frac{\hbar}{2m_1}\nabla_1^2 - \cdots - \frac{\hbar}{2m_N}\nabla_N^2 + V(\mathbf{r}_1, \dots \mathbf{r}_N), \tag{2.3}$$

where m_i is the mass of particle i and $\nabla_i^2 = \frac{\partial^2}{\partial x_i^2} + \frac{\partial^2}{\partial y_i^2} + \frac{\partial}{\partial z_i^2}$ is the Laplacian with respect to the coordinates of particle i. We can condense the notation by describing the coordinates of all particles in the system as $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_N) = (x_1, y_1, z_1, \dots, x_N, y_N, z_N)$. Now the wave function is simply $\Psi(\mathbf{R})$ and the potential is $V(\mathbf{R})$. If $m_1 = \cdots = m_N$, which is the case when we are dealing with many electrons in an atom, the multi-particle Hamiltonian becomes

$$H = -\frac{\hbar}{2m}\nabla^2 + V(\mathbf{R}). \tag{2.4}$$

3 Derivation of Diffusion Monte Carlo

The derivation that follows is based on a short article from *Computers in Physics* by Reynolds et al. [5].

As we saw in the Background section above, the Shrödinger equation is

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar}{2m}\nabla^2\Psi + V\Psi. \tag{3.1}$$

This equation has solutions of the form

$$\Psi(\mathbf{R}, t) = \sum_{n=0}^{\infty}\psi_n(\mathbf{R})e^{-iE_nt/\hbar}, \tag{3.2}$$

where ψ_n and E_n are the eigenfunctions and eigenvalues, respectively, of the time-independent Shrödinger equation, $H\Psi = E\Psi$. The ground state wave function is ψ_0 and the ground state energy is E_0.

Now we will shift the Shrödinger equation into imaginary time. Let $\tau = it/\hbar$, and let $D = \hbar^2/2m$. We will also shift the potential by the constant E_0, which does not change the eigenfunctions.

Then the Shrödinger equation becomes

$$\hbar\frac{\partial\Psi}{\partial\tau} = D\nabla^2\Psi - (V - E_0)\Psi, \tag{3.3}$$

which then has solutions of the form

$$\Psi(\mathbf{R}, \tau) = \sum_{n=0}^{\infty}\psi_n(\mathbf{R})e^{-(E_n-E_0)\tau}. \tag{3.4}$$

For the normal Shrödinger equation, solutions oscillate as time progresses; however, for the imaginary time Shrödinger equation, we can see that the ground state is constant in time, and all other solutions (for which $E_n > E_0$) exponentially decay in time. This means that if we can approximate the solution to equation (3.3) and propogate it forward in time, we can see whether $\Psi(\mathbf{R}, \tau)$ is increasing or decreasing and adjust our estimate for E_0 accordingly until we converge at an accurate estimate of the ground state energy.

We can also write the solution to equation (3.3) in terms of a Green's function, which is an operator $G = e^{-\tau[-D\nabla^2+(V-E_0)]}$ such that

$$\Psi(\mathbf{R}, t) = G\Psi(\mathbf{R}, 0). \tag{3.5}$$

According to Reynolds et al.,

$$G(\mathbf{R}, \mathbf{R}', t) = \langle\mathbf{R}'|\, e^{\tau D\nabla^2+\tau(V-E_0)}\, |\mathbf{R}\rangle \tag{3.6}$$

can be interpreted as the probability of moving from $\mathbf{R}$ to position $\mathbf{R}'$ in time τ. I do not have the background to fully understand how the Green's function operator works, but see the article by Reynolds et al. for more information [5]. Over a small time interval $\Delta\tau$, equation (3.6) can be approximated to $\mathcal{O}(\Delta t)$ as

$$\begin{aligned} G(\mathbf{R}, \mathbf{R}', \Delta\tau) &\approx \langle\mathbf{R}'|\, e^{-\Delta\tau V/2}e^{\Delta\tau D\nabla^2}e^{-\Delta\tau V/2}\, |\mathbf{R}\rangle\, e^{\Delta\tau E_0} \\ &= e^{-(\Delta\tau/2)[V(\mathbf{R}')+V(\mathbf{R})]+\Delta\tau E_0}\, \langle\mathbf{R}'|\, e^{\Delta\tau D\nabla^2}\, |\mathbf{R}\rangle \\ &= \omega(\mathbf{R}, \mathbf{R}', \Delta\tau)G_D(\mathbf{R}, \mathbf{R}', \Delta\tau), \end{aligned} \tag{3.7}$$

where

$$\omega(\mathbf{R}, \mathbf{R}', \Delta\tau) = e^{-(\Delta\tau/2)[V(\mathbf{R}')+V(\mathbf{R})]+\Delta\tau E_0} \tag{3.8}$$

is a weight which will be used in the DMC simulation as the rate of decay or birth of random walkers, and

$$G_D(\mathbf{R}, \mathbf{R}', \Delta\tau) = (4\pi D\Delta\tau)^{-3N/2} e^{(\mathbf{R}'-\mathbf{R})^2/4D\Delta\tau}, \tag{3.9}$$

which is the Green's function solution for ordinary diffusion. In equation (3.9), N is the number of particles.

DMC uses random walks to simulate applying the Green's function operator from equation (3.7) to the wave function $\Psi(\mathbf{R}, \tau)$. As we saw in equation (3.5), doing so propogates the wave function forward in time. Along the way, we will adjust our estimate for E_0 depending on whether $\Psi(\mathbf{R}, \tau)$ is generally increasing or decreasing. As we saw in equation (3.4), as our estimate becomes close to E_0, $\Psi(\mathbf{R}, \tau)$ will become constant as we propogate it forward in time and will approach the ground state $\psi_0(\mathbf{R})$. The steps for the algorithm, adapted from Reynolds et al., are outlined below [5].

1. Because we do not know $\Psi(\mathbf{R}, \tau)$, we start out with an random initial distribution of M values of the $3N$-dimensional vector $\mathbf{R}$, each of which specifies a possible configuration of N particles. The algorithm will converge faster if these points are chosen in regions where $\psi(\mathbf{R}, \tau)$ is expected to be reasonably large.

2. For each configuration $\mathbf{R}$ of the M configurations,

   (a) Displace each coordinate of $\mathbf{R}$ by a random amount chosen from a normal distribution with mean 0 and standard deviation $\sqrt{2D\Delta\tau}$ (which corresonds to applying $G_D(\mathbf{R}, \mathbf{R}', \Delta\tau)$ from equation (3.9)) to get $\mathbf{R}'$.

   (b) Calculate the weight $\omega(\mathbf{R}, \mathbf{R}', \Delta\tau)$ from equation (3.8) and create new walkers to account for the weight. This is done by choosing a uniform random number $\xi \in [0, 1)$ and taking the integer part of $\omega + \xi$ as the number of indentical walkers to have in the next iteration. So,

      i. If $\lfloor \omega + \xi \rfloor = 0$, delete the current walker.

      ii. If $\lfloor \omega + \xi \rfloor = 1$, do nothing

      iii. If $\lfloor \omega + \xi \rfloor > 1$ create $\lfloor \omega + \xi \rfloor - 1$ new walkers with identical configurations $\mathbf{R}$. As the simulation progresses, these new walkers will take their own random paths.

3. As Reynolds et al. explain, step 2 is equivalent to performing the integration

$$\Psi(\mathbf{R}', \tau + \Delta\tau) = \int G(\mathbf{R}, \mathbf{R}', \Delta\tau)\Psi(\mathbf{R}, \tau) d\mathbf{R}, \tag{3.10}$$

   which steps the wave function forward in time by $\Delta\tau$. After performing step 2, the distribution of $\mathbf{R}$ represents $\Psi(\mathbf{R}, \tau)$, a solution to equation (3.3). However, we must repeat step 2 thousands of times in order to converge to the ground state.

4. The energy estimate for $E_{0_{est}}$ must be adjusted along the way. In order to keep the simulation stable, $E_{0_{est}}$ is updated infrequently, typically every 100 steps or so. The new estimate is

$$E'_{0_{est}} = E_{0_{est}} - a\frac{M - M_0}{\Delta\tau M_0}, \tag{3.11}$$

where M is the current number of walkers, M_0 is the number of walkers when $E_{0_{est}}$ was last updated, and a is a proportionality factor which can be adjusted if the number of configurations is changing too quickly or slowly.

After many steps, the algorithm above will converge to a ground state energy E_0, and the distribution of configurations $\mathbf{R}$ will be sampled from the ground state wave function $\psi_0(\mathbf{R})$.

4 Guided Diffusion Monte Carlo

The algorithm explained in the previous section works pretty well and can converge to accurate values for E_0 and ψ_0. However, particularly for potentials with singularities (such as the Coulomb potential in helium), the procedure can be inefficient due to the large amount of branching that occurs. One way to improve the DMC algorithm is to incorporate additional knowledge about the unknown wave function by using a guess or "trial" wave function $\Psi_T(\mathbf{R}) \approx \psi_0(\mathbf{R})$ as a guide. Once again, the following derivations are adapted from Reynolds et al. [5].

In order to incorporate the trial wave function, we will introduce a new function $f(\mathbf{R}, \tau) = \Psi_T(\mathbf{R})\Psi(\mathbf{R}, \tau)$. If we then multiply the imaginary time Shrödinger equation, equation (3.3), by $\Psi_T(\mathbf{R})$, we get

$$\frac{\partial f}{\partial t} = D\nabla^2 f - D\nabla\cdot[f\mathbf{F}(\mathbf{R})] + [E_0 - E_L(\mathbf{R})]f, \tag{4.1}$$

where

$$E_L(\mathbf{R}) = \frac{H\Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})} = -D\frac{\nabla^2\Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})} + V(\mathbf{R}) \tag{4.2}$$

is known as the "local energy" [5] and

$$\mathbf{F}(\mathbf{R}) = 2\frac{\nabla\cdot\Psi(\mathbf{R})}{\Psi(\mathbf{R})} \tag{4.3}$$

is known as the "quantum force" [3]. As Reynolds et al. derive in their paper, the new Green's function operator is

$$\widetilde{G}(\mathbf{R}, \mathbf{R}', \Delta\tau) = \widetilde{\omega}(\mathbf{R}, \mathbf{R}', \Delta\tau)\widetilde{G}_D(\mathbf{R}, \mathbf{R}', \Delta\tau), \tag{4.4}$$

where the new branching weight is

$$\widetilde{\omega}(\mathbf{R}, \mathbf{R}', \Delta\tau) = e^{-(\Delta\tau/2)[E_L(\mathbf{R})+E_L(\mathbf{R}')]+\Delta\tau E_0} \tag{4.5}$$

and the new diffusion propogator is

$$\widetilde{G}_D(\mathbf{R}, \mathbf{R}', \Delta\tau) = (4\pi D\Delta\tau)^{-3N/2} e^{[\mathbf{R}'-\mathbf{R}-D\Delta\tau\mathbf{F}(\mathbf{R})]^2/4D\Delta\tau}. \tag{4.6}$$

Reynolds et al. explain that $\widetilde{G}_D$ is the operator for diffusion with a drift from the $\mathbf{F}(\mathbf{R})$ term, which in the DMC algorithm acts to push the random walk away from unimportant regions where $\Psi_T(\mathbf{R}, t)$ is small [5].

We can also improve the algorithm by selecting the initial distribution of $\mathbf{R}$ values based on the trial wave function. One way to do this is to use the Metropolis algorithm for sampling random points from a distribution function [5]. The Metropolis sampling process and the steps for the guided DMC algorithm are outlined below [5].

1. Because we do not know $f(\mathbf{R}, \tau)$, we will start out with M random values of $\mathbf{R}$. Because $f(\mathbf{R}, \tau) = \Psi_T(\mathbf{R})\Psi(\mathbf{R}, \tau)$, if the trial wave function is chosen such that $\Psi_T(\mathbf{R}) \approx \psi_0(\mathbf{R})$, then the algorithm is trying to converge to $f_0(\mathbf{R}) = \Psi_T(\mathbf{R})\psi_0(\mathbf{R}) \approx \Psi_T(\mathbf{R})^2$. Thus, we will sample the initial configurations from the distribution function $\Psi_T(\mathbf{R})^2$ using the Metropolis algorithm, outlined below.

   (a) Select an initial random configuration $\mathbf{R}$ and add it to the list of accepted $\mathbf{R}$ values.

   (b) Until the average local energy (from equation (4.2)) has settled down:

      i. Randomly change each coordinate in the most recently accepted value of $\mathbf{R}$ to get a new configuration $\mathbf{R}'$.

      ii. If $\Psi_T(\mathbf{R}')^2/\Psi_T(\mathbf{R})^2 + \xi > 1$, where $\xi \in [0, 1)$ is a uniform random number, then add $\mathbf{R}'$ to the list of initial configurations; otherwise, reject $\mathbf{R}'$.

   (c) Select M evenly spaced values from the accepted values of $\mathbf{R}$. For example, if we end up with $10M$ values, then select every 10th value of $\mathbf{R}$ to get M initial values. (Selecting every 10th value also helps with bias due to adjacent $\mathbf{R}$ values being close to each other.) These should now be sampled from $\Psi_T(\mathbf{R})^2$.

2. For each configuration $\mathbf{R}$ of the M configurations,

   (a) Displace each coordinate of $\mathbf{R}$ by a random amount chosen from a normal distribution with mean 0 and standard deviation $\sqrt{2D\Delta\tau}$, and impose drift by adding $D\Delta\tau\mathbf{F}(\mathbf{R})$ from equation (4.3). This corresponds to applying $\widetilde{G}_D(\mathbf{R}, \mathbf{R}', \Delta\tau)$ from equation (4.6).

   (b) Calculate the weight $\widetilde{\omega}(\mathbf{R}, \mathbf{R}', \Delta\tau)$ from equation (4.5) and create new walkers to account for the weight using the same process outlined in the previous section.

3. In analogy with before, step 2 performs the integration

$$f(\mathbf{R}', \tau + \Delta\tau) = \int \widetilde{G}(\mathbf{R}, \mathbf{R}', \Delta\tau) f(\mathbf{R}, \tau) d\mathbf{R}, \tag{4.7}$$

   which propogates $f(\mathbf{R}', \tau)$ forward in time. After this point, the algorithm is identical to before. We adjust our estimate for E_0 and our distribution of $\mathbf{R}$'s will converge to the distribution $f_0(\mathbf{R}) = \Psi_T(\mathbf{R})\psi_0(\mathbf{R})$. If $\Psi_T \approx \psi_0(\mathbf{R})$, then $f_0(\mathbf{R}) \approx \psi_0(\mathbf{R})^2$, the probability density function for the ground state.

5 Implementing Diffusion Monte Carlo

I used MATLAB to implement the unguided and guided DMC algorithms from the previous two sections. The commented code can be found at `https://github.com/alexhad6/DiffusionMonteCarlo`. The file `DMC.m` implements the basic DMC algorithm, and the file `DMCguided.m` implements the guided DMC algorithm. In the MATLAB live file `DMCtests.mlx`, I test my DMC implementations on three potentials: the one-dimensional harmonic oscillator, an electron in a hydrogen atom, and two electrons in a helium atom.

In order to implement the algorithms explained earlier, we first need to know what units we are using in order to have specific values for physical constants, and in order to interpret our results. Also, to implement guided DMC, we need to determine a reasonable trial function and to calculate the local energy and quantum force functions. This is done below.

5.1 Units

In the derivations from the previous sections, we use the reduced Plank's constant and the mass of an electron. In SI units, the value of the reduced Plank's constant is $\hbar \approx 1.05 \times 10^{-34}\,\mathrm{J\,s}$ and the mass of an electron is $m_e \approx 9.11 \times 10^{-31}\,\mathrm{kg}$. According to Coulomb's law, the potential between an electron and a proton depends on the distance r between them according to

$$V(r) = -\frac{e^2}{4\pi\epsilon_0 r}, \tag{5.1}$$

where $e \approx 1.60 \times 10^{-19}\,\mathrm{C}$ is the charge of an electron and $\epsilon_0 \approx 8.85 \times 10^{-12}\,\mathrm{F\,m^{-1}}$ is the vacuum permittivity constant. If we were to use SI units in our code, we would be dealing with very small numbers, which is confusing and seems prone to bugs.

Thankfully, there is another option. We can instead use atomic units, a unit system which conveniently sets $\hbar = e = a_0 = m_e = 1$, where a_0 is the Bohr radius, a unit of length on the atomic scale [2]. In atomic units, the diffusion constant for electrons reduces to $D = \hbar/2m_e = 1/2$ and the Coulomb potential from equation (5.1) becomes $V(r) = -1/r$.

In atomic units, $a_0 = 1\,\mathrm{bohr} \approx 5.29 \times 10^{-11}\,\mathrm{m}$ is the unit of length and $E_h = 1\,\mathrm{hartree} \approx 27.2\,\mathrm{eV} \approx 4.36 \times 10^{-18}\,\mathrm{J}$ is the unit of energy. Since we will use atomic units for the physical constants in our DMC simulation, the results will naturally be given in these units.

5.2 One-Dimensional Harmonic Oscillator

The first potential I tested my DMC implementations on was the one-dimensional harmonic oscillator potential. For convenience, I set $\omega = 1\ E_h/\hbar \approx 4.13 \times 10^{16}\,\mathrm{s^{-1}}$. Then in atomic units, the potential becomes

$$V(x) = \frac{1}{2}m_e\omega^2 x^2 = \frac{1}{2}x^2, \tag{5.2}$$

For this potential, the exact ground state solution to the Shrödinger equation (equation (3.1)) is

$$\psi_0(x) = \pi^{-1/4}e^{-x^2/2}, \tag{5.3}$$

with corresponding energy $E_0 = 0.5\,\mathrm{hartree}$. We can use this value to check that the results of the DMC simulations are accurate. Given that this is the true ground state, a good trial wave function to test with is

$$\Psi_T(x) = e^{-\beta x^2}, \tag{5.4}$$

where β is a constant close to the true value, 0.5 [5]. According to equation (4.2), the local energy for this trial wave function is

$$E_L(x) = -D\frac{\frac{d^2}{dx^2}\Psi_T(x)}{\Psi_T(x)} + V(x) = -D(4\beta^2x^2 - 2\beta) + V(x), \tag{5.5}$$

and the quantum force from equation (4.3) is

$$F(x) = 2\frac{\frac{d}{dx}\Psi_T(x)}{\Psi_T(x)} = -4\beta x. \tag{5.6}$$

5.3 Hydrogen

A hydrogen atom can be modeled as an electron in the Coulomb potential of a proton. If the position of the electron is $\mathbf{r} = (x, y, z)$, then the potential (in atomic units) becomes

$$V(\mathbf{r}) = -\frac{e^2}{4\pi\epsilon_0 r} = -\frac{1}{r}, \tag{5.7}$$

where r is the magnitude of $\mathbf{r}$. A few approximations are made when using this potential. First, the true mass that should be used is the reduced mass of the proton and the electron, $\mu = \frac{m_e m_p}{m_e + m_p}$; however, since $m_p >> m_e$, $\mu \approx m_e$. Also, we are not accounting for spin-spin interactions and various other aspects of the system that might affect the potential. These other terms are relatively insignificant for the cases we are studying here.

For the hydrogen atom potential in equation (5.12), the exact ground state solution in atomic units is

$$\psi_0(\mathbf{r}) = \pi^{-1/2} e^{-r}, \tag{5.8}$$

with corresponding energy $E_0 = -0.5\,\mathrm{hartree}$. A good trial wave function for hydrogen is

$$\Psi_T(\mathbf{r}) = e^{-\beta r}, \tag{5.9}$$

where β is a constant close to the true value of 1 [5]. According to equation (4.2), the local energy for this trial wave function is

$$E_L(x) = -D\frac{\nabla^2\Psi_T(\mathbf{r})}{\Psi_T(\mathbf{r})} + V(\mathbf{r}) = -D\left(\beta^2 - \frac{2\beta}{r}\right) + V(\mathbf{r}), \tag{5.10}$$

and the quantum force from equation (4.3) is

$$F(x) = 2\frac{\nabla \cdot \Psi_T(x)}{\Psi_T(x)} = -2\beta\frac{\mathbf{r}}{r}. \tag{5.11}$$

5.4 Helium

For helium, we will make the same approximations that we made for hydrogen, where we will use m_e as the mass and ignore less significant terms in the potential. There are two electrons in helium, so the potential will have three terms: two Coulomb potential terms for the interactions between the two electrons and the nucleus (of charge $2e$), and a third term for the interaction between the two electrons. Thus, in atomic units the potential becomes

$$V(\mathbf{R}) = -\frac{2e^2}{4\pi\epsilon_0 r_1} - \frac{2e^2}{4\pi\epsilon_0 r_2} + \frac{e^2}{4\pi\epsilon_0 r_{12}} = -\frac{2}{r_1} - \frac{2}{r_2} + \frac{1}{r_{12}}, \tag{5.12}$$

where $\mathbf{R} = (\mathbf{r_1}, \mathbf{r_2})$ represents the positions of both electrons, r_1 and r_2 are the magnitudes of $\mathbf{r_1}$ and $\mathbf{r_2}$, and $r_{12} = |\mathbf{r_1} - \mathbf{r_2}|$ is the distance between the two electrons.

For helium, the exact ground state solution is unknown. However, it has been empirically found that the ground state energy is $E_0 \approx 2.903\,386$ hartree, so we can use this as a reference value for the accuracy of our results [4].

A reasonable trial function for helium can be constructed by ignoring the electron-electron repulsion term in the potential, which results in each electron being in a hydrogen ground state [1]. The trial wave function is then

$$\Psi_T(\mathbf{R}) = e^{-\beta(r_1+r_2)}, \tag{5.13}$$

where β is a constant that is chosen so that Ψ_T best approximates ψ_0. In this case, using the charge of the nucleus, $\beta = 2$, is a reasonably good approximation [1].

According to equation (4.2), the local energy for this trial wave function is

$$E_L(\mathbf{R}) = -D\frac{\nabla^2\Psi_T(\mathbf{R})}{\Psi_T(\mathbf{R})} + V(\mathbf{R}) = -2D\left(\beta^2 - \beta\left(\frac{1}{r_1} + \frac{1}{r_2}\right)\right) + V(\mathbf{r}), \tag{5.14}$$

and the quantum force from equation (4.3) is

$$F(x) = 2\frac{\nabla\cdot\Psi_T(x)}{\Psi_T(x)} = -2\beta\left(\frac{\mathbf{r_1}}{r_1}, \frac{\mathbf{r_2}}{r_2}\right). \tag{5.15}$$

5.5 Simulation Parameters

Now that everything has been set up, there are still a few parameters needed in order to run the DMC algorithm. Reynolds et al. write that for a one-dimensional harmonic oscillator, they got reasonable results (5% to 10% accuracy) using a time step of $\Delta\tau = 0.01$, an initial number of walkers $M = 100$, 1000 total steps in the algorithm, an adjustment to $E_{0_{est}}$ every 20 steps, and a proportionality factor of $a = 0.1$, so we will use these as our starting parameter values [5]. When I tried these simulation parameters, I found that the adjustment to E_0 was changing too quickly, so I used $a = 0.01$ instead, which worked much better.

In the DMC algorithm, a few places refer to "convergence." In this cases, I found better results setting a maximum number of steps like Reynolds et al. do [5]. For the algorithm itself, I found accurate results using 1000 steps for all cases that I investigated. Also, for the Metropolis sampling step of the guided DMC algorithm, I found better results by explicitly sampling $10M$ configurations from Ψ_T^2 and then selecting every 10th configuration than by trying to detect when the local energy converged.

5.6 Psuedo Random Numbers

Monte Carlo algorithms involve pseudo random numbers. In order to make my results reproducable, I used the Mersenne Twister generator with a particular seed, 737308, and ran the entire file `DMCtests.mlx`. This seed was also selected randomly so as to prevent bias.

6 Findings

6.1 Initial Test

Using the simulation parameters from the previous section, I first applied the unguided DMC algorithm to the one-dimensional harmonic oscillator. Using the simulation param-

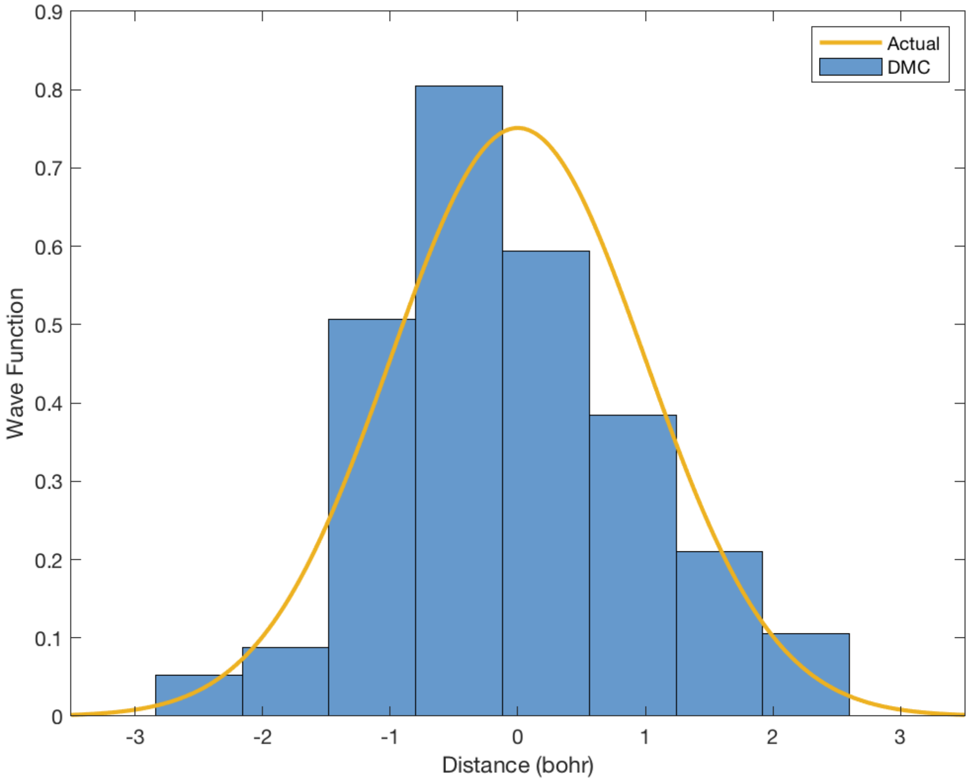

Figure 6.1: Ground state of electron in one-dimensional harmonic oscillator (unguided DMC, $M = 100$)

eters explained above and an initial energy estimate of $E_{0_{est}} = 1$, the algorithm returned a ground state energy of $E_0 = 0.506064898600324$. This is pretty close to the exact value of 0.5, which means that the algorithm is reasonable. The algorithm also returns the final distribution of x values, which should approximate the ground state wave function. In order to visualize this distribution, I created a histogram of the x values and normalized the histogram so that the square of the area is 1 (this is how wave functions are normalized). Then I plotted the true ground state wave function, equation (5.3), resulting in Figure 6.1.

Next, I tried the guided DMC algorithm, sampling from the trial wave function and using the local energy and quantum force equations derived in section 5.2. Using 0.4 as the value for β for the trial wavefunction, the algorithm returned $E_0 = 0.499275606776310$, which is even closer than before. Once again, I plotted the resulting distribution, which can be seen in Figure 6.2. Note that the guided DMC algorithm returns the distribution of $\Psi_T\psi_0 \approx \psi_0^2$, so this time the solution is normalized and plotted against the true value of ψ_0^2.

The estimate for $\psi_0(x)^2$ in Figure 6.2 seems to be pretty accurate. I found that by increasing the number of walkers from $M = 100$ to $M = 1000$ and plotting the results in Figure 6.3, it becomes even more clear that the points truly are being sampled, at least approximately, from $\psi_0(x)^2$.

6.2 Statistical Error

When numerically approximating something, it is important to have an error value so that we know the accuracy of our estimate. For Monte Carlo algorithms, an error can

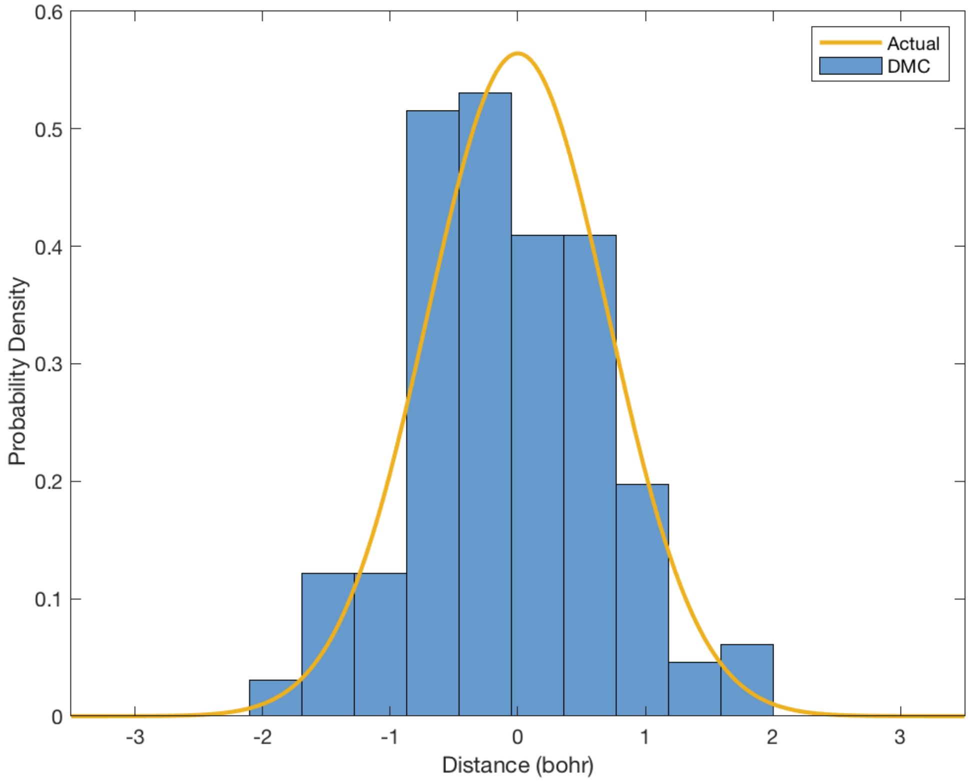

Figure 6.2: Ground state of electron in one-dimensional harmonic oscillator (guided DMC, $M = 100$)

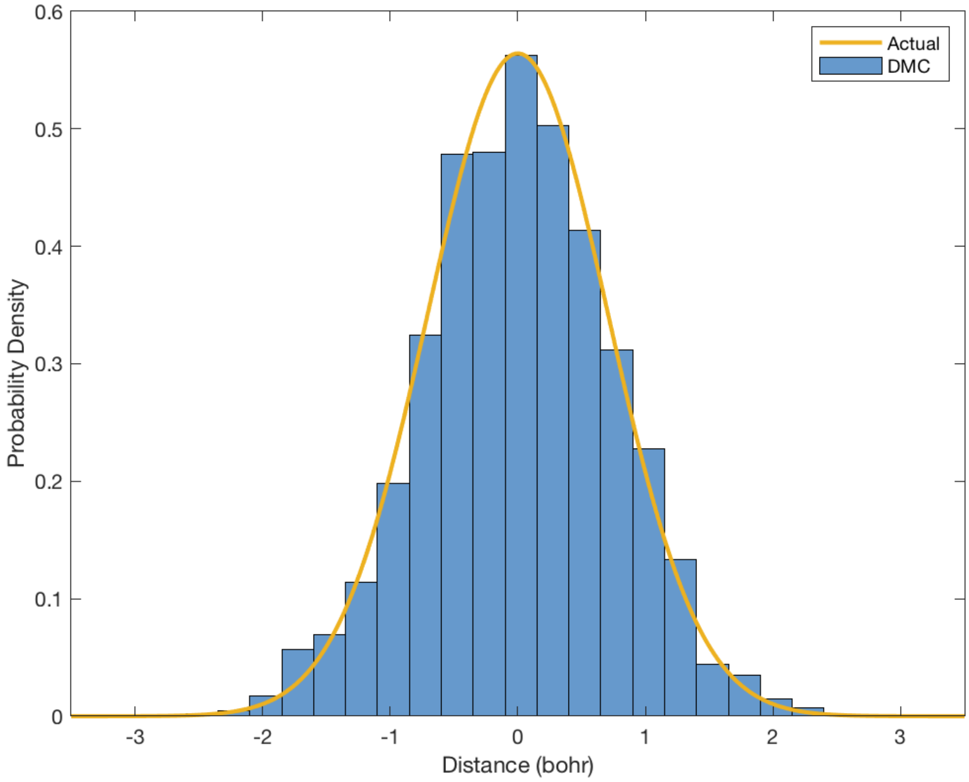

Figure 6.3: Ground state of electron in one-dimensional harmonic oscillator (guided DMC, $M = 1000$)

be estimated by running the same code many times and taking the mean and standard deviation of the results.

As Reynolds et al. explain, the accuracy of using the guided DMC algorithm depends on the accuracy of the trial wave function used [5]. To verify that this is true and as an example of calculating statistical error, we can compare the error using the unguided and guided DMC algorithms on the one-dimensional harmonic oscillator for various values of β. I performed several trials for the one-dimensional harmonic oscillator, with $M = 100$. For each trial, I found the mean and standard deviation (error) of 100 samples. Between trials, I varied the value of β. The results are shown below.

Algorithm	β	**Mean** E_0	**Error in** E_0
unguided	—	0.5018	0.0432
guided	0.2	0.5023	0.0386
guided	0.4	0.4988	0.0189
guided	0.6	0.4973	0.0220
guided	0.8	0.4979	0.0438
guided	1.0	0.5068	0.0646

We can see all means are relatively close to the true value of $E_0 = 0.5$, and that the true value is within the statistical error. Compared to the statistical error for the unguided algorithm, we can see that β values of 0.4 and 0.6 offer a significant reduction in the error. We can also see that indeed, the error is smallest when β is closest to 0.5, which would give the exact wave function. These tests confirm that our choice of trial wave function affects the accuracy we can achieve using guided DMC and that guided DMC has the potential to be more accurate than the unguided algorithm.

6.3 Hydrogen and Helium

Next, I ran 100 trials of the guided and unguided DMC algorithms for all three potentials, calculating the mean value of E_0 and the statistical error. For the guided algorithms, I used the trial wave functions from sections 5.2, 5.3, and 5.4. For the one-dimensional harmonic and hydrogen potentials, I used values of β which did not result in the exact wave function so that the algorithm was truly performing an estimate. For helium, there is no exact value for β, so I used 2, which is a good estimate [1]. Since helium is the result we care most about given that the others can be solved for exactly, I did one final trial for helium with $M = 1000$ initial walkers to further increase the accuracy.

Potential	**Algorithm**	β	M	**Initial** E_0	**Mean** E_0	**Error**	**True** E_0
1D harmonic	unguided	—	100	1	0.5018	0.0473	0.5
1D harmonic	guided	0.4	100	1	0.5015	0.0179	0.5
hydrogen	unguided	—	100	-1	-0.4821	0.1022	-0.5
hydrogen	guided	0.9	100	-1	-0.4978	0.0248	-0.5
helium	unguided	—	100	-3	-2.444	0.5790	-2.903
helium	guided	2.0	100	-3	-2.932	0.0581	-2.903
helium	guided	2.0	1000	-3	-2.925	0.0233	-2.903

Overall, the estimates for E_0 all seem relatively close to the true values. The errors seem to get worse for the helium simulation, perhaps due to additional degrees of freedom since there are now six dimensions (3 spatial dimensions per electron). The unguided helium simulation has the worst estimate for E_0, although it is still within the error of the true value. The errors for the guided helium simulations are much lower, showing that guided DMC really is neccessary for practical applications. For the last trial with $M = 1000$ walkers, the estimate for helium's E_0 is the closest to the true value, although it is still a bit lower. I appears that both of the estimates in the last two trials for helium are too low, although I do not think there is any systematic bias since I did see closer values (closer to -2.90) in other informal trials.

Because the estimates for helium are within the error bounds, they are still accurate estimates. Our previous set of trials suggests that one way to decrease the error is to construct a more accurate trial wave function, which for helium would probably involve constructing a completely new function rather than refining the parameter β.

We can also plot the distribution estimates for the ground state wave functions of hydrogen and helium. In this case, I ran a single simulation with $M = 10\,000$ initial random walkers in order to get higher resolution. In this case, the distributions will actually be three-dimensional. In order to visualize the probability distribution for hydrogen and helium, I calculated the radial distance from the nucleus for each point. I then plotted the one-dimensional radial distribution as a histogram.

For hydrogen, we can plot the square of the exact wave function from equation 5.8. In this case, to convert from a three-dimensional probability density function to the one-dimensional radial probability distribution, we can multiply by the surface area of a sphere, $4\pi r^2$. Because the wave function is spherically symmetric, this is equivalent to integrating over a spherical shell at radius r to get the probability that the particle is a distance r from the nucleus. Thus, the radial probability density function for hydrogen is

$$P(r) = 4\pi r^2 \psi_0(\mathbf{r})^2 = 4\pi \left(r^2 \pi^{-1/2} e^{-r} \right)^2 , \tag{6.1}$$

which is plotted in Figure 6.4.

For helium, we do not know the exact wave function, so I just plotted the estimated radial probability distribution. In this case, there are two electrons, so I plotted the distributions for both electrons in figures 6.5 and 6.6.

For hydrogen, the radial distribution approximately matches the true solution, which shows that the algorithm can reach an accurate solution. The distributions for helium look similar to the one for hydrogen, which makes sense given that the electrons are in the $1s$ orbital in both atoms. It is a good sign that the distributions in figures 6.5 and 6.6 look similar, since the two electrons in helium should be indistinguishable. It also makes sense intuitively that the peak of the plots for helium (the point where the electrons are most likely to be found) is about half the value it was for hydrogen, since the charge of the nucleus has doubled, which would pull the electrons closer.

Also, when I performed the calculations to generate these graphs, I also printed out the corresponding values of E_0. For $M = 10\,000$, the hydrogen simulation gave $E_0 = -0.5027$ and the helium simulation gave $E_0 = -2.9037$. The hydrogen estimate is similar to previous estimates, suggesting that while increasing the number of walkers makes for better plots, it does not necessarily improve the esimates for E_0. However, the estimate for E_0 for helium is the closest yet to the true value of -2.903. Without repeated samples (which would be difficult to conduct since the helium simulation took 206 seconds), we do not know the statistical error on these estimates, but it seems possible that for helium, more walkers significantly improves the estimate.

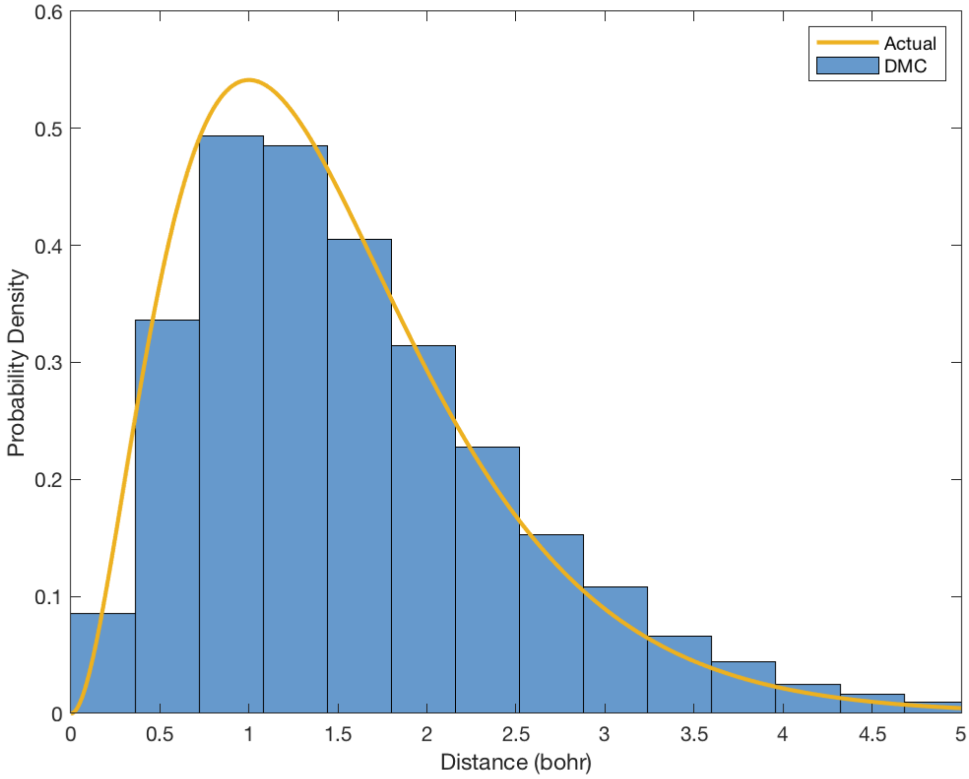

Figure 6.4: Ground state of electron in helium (guided DMC, $M = 10\,000$)

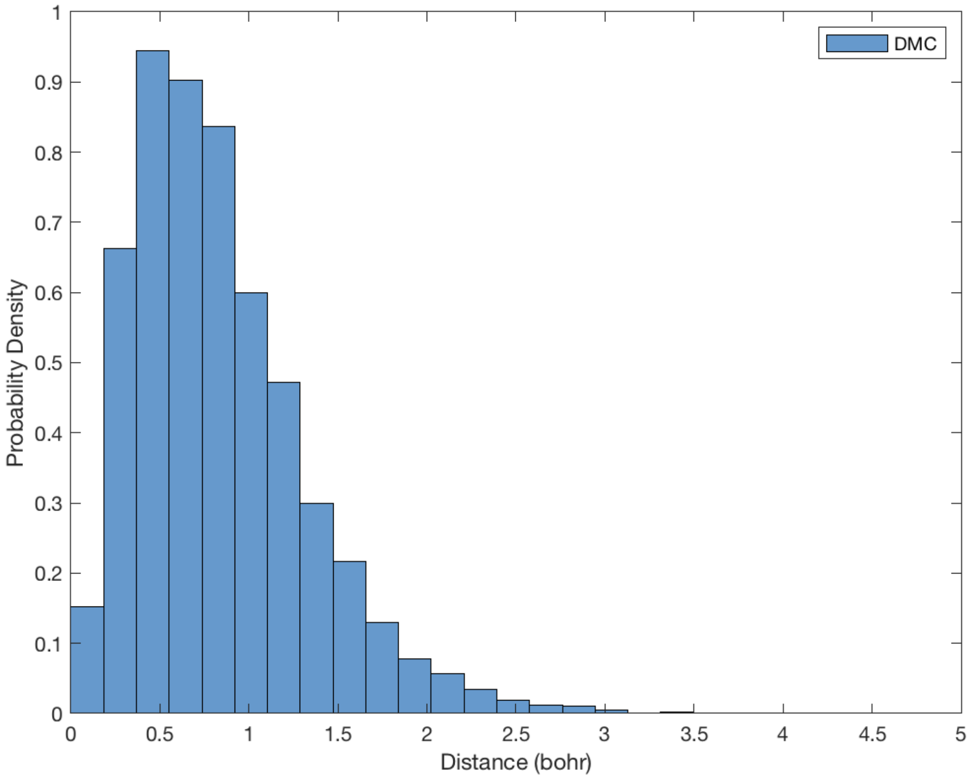

Figure 6.5: Ground state of 1st electron in helium (guided DMC, $M = 10\,000$)

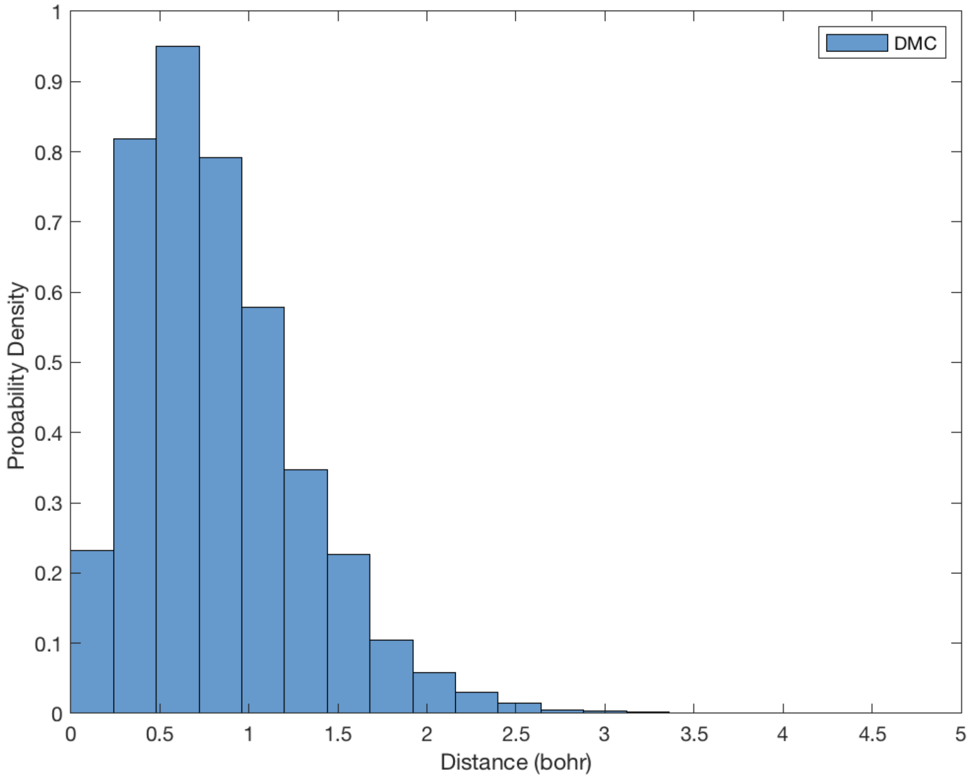

Figure 6.6: Ground state of 2nd electron in helium (guided DMC, $M = 10\,000$)

7 Conclusion

It appears that the diffusion Monte Carlo method has a good potential for calculating the ground states of quantum systems. Especially for more complicated systems, the guided DMC algorithm can greatly increase the accuracy of estimates of the ground state energy. The main challenge seems to lie in finding an accurate trial wave function, which decreases the error.

Based the trials I conducted, it appears that performing the algorithm many times and taking the average of the resulting estimates of E_0 tends to give a fairly accurate estimate and error bound. For estimating E_0, increasing the initial number of walkers M may slightly improve estimates, although it greatly increases the time per trial. However, for estimating the ground state wave function itself, one should increase the number of random walkers M to between 1000 and 10 000 to improve resolution.

In the future, one could look into whether adjusting the simulation parameters (such as $\Delta\tau$, M, a, etc.) could improve accuracy. It could also be possible to write my implementations of the DMC algorithms in more computationally efficient ways, perhaps involving parallelization. There also might be ways to improve the DMC algorithm itself, perhaps by keeping the number of random walkers more constant over time, since currently they have the potential to start multiplying rapidly.

Acknowledgments

I discussed my topic with Prof. Yong, who gave some insight on Monte Carlo methods in general.

References

[1] S. M. Blinder, *Quantum chemistry, chapter 8: Helium atom*, U. Michigan, 2002, available at `http://www.umich.edu/~chem461/QMChap8.pdf`.

[2] D. R. Hartree, *The wave mechanics of an atom with a non-Coulomb central field. Part I. Theory and methods*, Math. Proc. Cambridge Philos. Soc. **24** (1928), no. 1, 89–110, DOI `10.1017/S0305004100011919`.

[3] Paul Kent, *Techniques and applications of quantum Monte Carlo*, Ph.D. thesis, University of Cambridge, 1999, available at `https://web.ornl.gov/~kentpr/thesis/`.

[4] A. Kramida, Yu. Ralchenko, J. Reader, and NIST ASD Team, *NIST atomic spectra database (ver. 5.7.1)*, National Institute of Standards and Technology, 2019, DOI `10.18434/T4W30F`.

[5] Peter J. Reynolds, Jan Tobochnik, and Harvey Gould, *Diffusion quantum Monte Carlo*, Comput. in Phys. **4** (1990), no. 6, 662–668, DOI `10.1063/1.4822960`.

[6] J. C. Slater, *A simplification of the Hartree-Fock method*, Phys. Rev. **81** (1951), no. 3, 385–390, DOI `10.1103/PhysRev.81.385`.