Diffusion Monte Carlo Introduction and Derivation

Luke Hatcher

University of Washington: Department of Chemistry
McCoy Group, Fall 2019

1 Introduction to Diffusion Monte Carlo

Diffusion Monte Carlo (DMC) is a numerical method for solving the time independent Schrodinger equation. At its core, DMC is computational algorithm. The algorithm approximates the zero point energy (ZPE) of a system as well as produces an accurate representation of Ψ.[1]

1.1 Applications of DMC

DMC is a useful approach for studying the vibrations of molecules that exhibit anharmonic behavior. Molecules with lots of anharmonicity are poor candidates for other methods.

Applications include investigating the effect of adding zero point energy to a system and observing what happens to the wavefuction when the system can access different parts of the potential. This can give insight into the structural configuration of the system.

Furthermore, while the style of DMC detailed in this document limits results to the ground state vibrational energy, further more complicated applications include approximating higher order energies.[2]

2 Derivation

The goal of this section is to motivate Diffusion Monte Carlo (DMC) by providing a well outlined derivation to the best of my understanding.

2.1 Imaginary Time

Although DMC solves the TISE, it starts with the TDSE equation and time dependant quantum mechanics are applied to solve it.

$$i\hbar\frac{\partial}{\partial t}\Psi(r,t) = \hat{H}\Psi(r,t) \tag{1}$$

The solutions as a superposition of eigenstates of the Hamiltonian:

$$\Psi(r,t) = \sum_n c_n e^{\frac{-iE_n t}{\hbar}} \psi(r) \tag{2}$$

Complete a Wick rotation by substituting $\frac{it}{\hbar} = \tau$ to transform to imaginary time.

$$\Psi(r,\tau) = \sum_n c_n e^{-E_n \tau} \psi(r) \tag{3}$$

2.2 Discretize time

In order for this algorithm to work, it is important to note that imaginary time τ must be discretized into time steps of length $\Delta\tau$. A smaller $\Delta\tau$ is more accurate however it is also more computationally expensive. The algorithm is ran for a finite number of time steps where at any point, τ is equivalent to the corresponding timestep.

2.3 Shift energy by V_{ref}

For reasons that will become more apparent later, we shift the energy in equation (3) by a reference amount V_{ref}:

$$V_{ref}(\tau) = \overline{V}(\tau) - \alpha \frac{N_w(\tau) - N_w(\tau_o)}{N_w(\tau_o)},\ \alpha = \frac{1}{2\Delta\tau} \tag{4}$$

Plugging this into (3) results in:

$$\Psi(r,\tau) = \sum_n c_n e^{-(E_n - V_{ref})\tau} \psi(r) \tag{5}$$

Here, V_{ref} is composed of several parts. For now, this equation relies on several details of the algorithm which are explained later in section 3. For now all that needs to be understood is that N_w is the number of "walkers" at that point and $\overline{V}(\tau)$ is the average of each "walkers" potential at time τ. The importance of V_{ref} is shown next in section 2.4 and its application to the algorithm will be outlined in section 3 as well.

2.4 ZPE convergence

The convenience of DMC is that when $E_0 = V_{ref}$,

$$\Psi(r,\tau) = \sum_n c_n e^{-(E_n - V_{ref})\tau} \psi(r) = c_0 \psi_0 \tag{6}$$

Thus assuming that for long enough time τ, V_{ref} converges to E_0:

$$\lim_{\tau\to\infty} \sum_n c_n e^{-(E_n - V_{ref})\tau} \psi(r) = c_0 \psi_0 \tag{7}$$

Therefore the ZPE can be found upon this convergence. Now in order to do so, there needs to be a way to propagate forward in imaginary time.

2.5 Propagation of τ

Since the Hamiltonian is time independent, let's consider the time evolution operator:

$$\hat{U} = e^{\frac{-iEt}{\hbar}} \tag{8}$$

Including the Wick rotation from section 2.1:

$$\hat{U} = e^{-\hat{H}\tau} \tag{9}$$

Normally, $\hat{U}$ as written would now be operated on (6) in order to propagate forward. However an approximation will be used to simplify the operator to in turn simpligy the algorithm. Recalling that $\hat{H} = \hat{T} + \hat{V}$ and Taylor expanding $\hat{U}$ results in:

$$e^{-\hat{H}\Delta\tau} = 1 + (\hat{T} + \hat{V})\Delta\tau - \frac{1}{2}(\hat{T} + \hat{V})^2\Delta\tau^2 + ... \tag{10}$$

recombining terms:

$$= (1-\hat{T}\Delta\tau+\frac{1}{2}\hat{T}^2\Delta\tau^2-...)(1-\hat{V}\Delta\tau+\frac{1}{2}\hat{V}^2\Delta\tau^2-...)+(\hat{T}\hat{V}-\hat{V}\hat{T})\Delta\tau^2+... \tag{11}$$

which is approximately equivalent to:

$$\approx e^{-\hat{T}\Delta\tau}e^{-\hat{V}\Delta\tau} + [\hat{T}, \hat{V}]\Delta\tau^2 + ... \tag{12}$$

here the commutator of $\hat{T}$ and $\hat{V}$ is viewed as a correction value. Due to the $\Delta\tau^2$ term and considering a small enough value of $\Delta\tau$, this term is omitted in the approximation, leaving:

$$\approx e^{-\hat{T}\Delta\tau}e^{-\hat{V}\Delta\tau} \tag{13}$$

furthermore, adding in V_{ref} from section 2.3 results in:

$$\approx e^{-\hat{T}\Delta\tau}e^{-(\hat{V}-V_{ref})\Delta\tau} \tag{14}$$

2.6 Representing Ψ

In Diffusion Monte Carlo, an ensemble of localized functions is used to represent Ψ.

$$\Psi(r, \tau) = \sum_i f(r - r_i(\tau)) \tag{15}$$

here, the sum is indexed by the number of walkers at that point and each walker only has amplitude at position r_i at time τ.

2.7 Operating on Ψ

In order to propagate forward, we will operate our time evolution operator approximation (14) on the representation of Ψ (15).

$$\Psi(r, \tau + \Delta\tau) = e^{-(\hat{V} - V_{ref})\Delta\tau} e^{-\hat{T}\Delta\tau} \sum_i f(r - r_i(\tau)) \tag{16}$$

The operator (14) is composed of a kinetic and a potential portion. It does not matter which portion operates on Ψ first. However, for algorithmic simplicity, the kinetic portion will act first. Note that in the following equations the summation from (16) has been omitted. This is done for simplicity where only one walker i is being worked with.

$$e^{-\hat{T}\Delta\tau} f(r - r_i(\tau)) \tag{17}$$

This operation involves a Fourier transform from position to momentum space and back in order to solve. The details of this are beyond the scope of this paper, however the result is a Gaussian function of width $\sqrt{\frac{\Delta\tau}{m}}$. The operation of the potential energy follows suit:

$$e^{-(\hat{V}((r_i(\tau+\Delta\tau)) - V_{ref})\Delta\tau} f(r - r_i(\tau + \Delta\tau)) \tag{18}$$

here, the walker i located at r_i at time $\tau + \Delta\tau$ has its potential energy evaluated at that point. Furthermore, the reason $\tau + \Delta\tau$ is used here rather than τ is because of the order we operated the two portions of the time evolution operator (16). The Gaussian made in (17) at time τ is not used until time $\tau + \Delta\tau$.

Much of this may be difficult to understand without a better background on the actual algorithm which is being used which will now be explained.

3 Algorithm

The first thing to understand is what exactly a "walker" is. Mathematically, the walkers are localized functions. Algorithmically, a walker is a set of coordinates that correspond to the molecule the DMC simulation is being run on. For example, if DMC is being ran on H_2O, one walker would be composed of two Hydrogen coordinates and one Oxygen coordinate for a total of three coordinates.

Prior to starting the simulation, an amount of initial walkers are chosen and each walker is given the equilibrium coordinates of the molecule of interest.

3.1 Each timestep

Displace each walker by amount drawn randomly from Gaussian of width $\sqrt{\frac{\Delta\tau}{m}}$. Calculate V_{ref} from prior timestep using equation 4. Evaluate each walker at the potential $V(r_i)$. Now using each walkers evaluated potential, it is compared

to the V_{ref}. Each walker is then assigned a probability of either replicating or deleting itself based on criteria listed below.

$$P_b = \begin{cases} \mathrm{e}^{(-(V(r_i)-V_{ref})\Delta\tau)-1}, & \text{if } V(r_i) < V_{ref}. \\ 0, & \text{if } V(r_i) > V_{ref}. \end{cases} \tag{19}$$

$$P_d = \begin{cases} \text{1-e}^{(-(V(r_i)-V_{ref})\Delta\tau)}, & \text{if } V(r_i) > V_{ref}. \\ 0, & \text{if } V(r_i) < V_{ref}. \end{cases} \tag{20}$$

Once probabilities of birth or death have been assigned to a walker, a uniformly random number X is drawn on the range $[0, 1]$. This number is compared to the walkers P_b or P_d. If X is less than P_b or P_d, the walker is birthed or killed respectively. This replication and deletion process leads to V_{ref} converging to E_0.

3.2 Extracting ZPE results

As detailed in section 2.4, for long enough τ, V_{ref} converges to E_0. Thus, once the DMC simulation has ran to completion, taking the average off each V_{ref} collected at each timestep is approximately equal to the ZPE. However due to the nature of convergence, the first $10 - 20\%$ of the simulations V_{ref} values are not used in the average calculation to account for the time the ensemble took to converge.

3.3 Extracting Ψ^2 using descendant weighting

Since we are representing Ψ as an ensemble of localized functions(walkers), then the walker coordinates at the end of the simulation is equal to the wavefunction Ψ. Knowing that Ψ^2 is a probability density, then by assigning weights to each walker, a probability density should be able to be plotted.

Returning to the replication and deletion portion of the algorithm, each time a walker is replicated it gets a weight of $+1$ and each time it is deleted it gets a weight of 0. Collecting this data allows for Ψ^2 to be projected onto the bond length by histogramming along the bond length and using the walkers weights as their level of contribution to each histogram bin. The result is a probaboilty density representing Ψ^2.

4 References

[1] James B. Anderson, Quantum chemistry by random walk. J. Chem. Phys. 65, 4121 (1976); https://doi.org/10.1063/1.432868

[2] Anne B. McCoy, Eric G. Diken, and Mark A. Johnson, Generating Spectra from Ground-State Wave Functions: Unraveling Anharmonic Effects in the OH·H_2O Vibrational Predissociation Spectrum, The Journal of Physical Chemistry A 2009 113 (26), 7346-7352 DOI: 10.1021/jp811352c