Using Random Walk Simulations to Calculate Ground State Energies in Quantum Physics

Sai Pandian, ID: 29899923

***Abstract*—In this paper, we present a novel way of calculating ground state energies for a quantum simulation using a Random Walk Monte Carlo simulation. The random walk consists of a walker making steps in random directions, with each step having a probability of causing the walker to die and the random walk to end, with the probability determined by a function of position. The function can be chosen to model specific quantum systems. The rate of death of the walkers is used to calculate the ground state energy. This paper explores the efficacy of such a method in reproducing the known ground state energy of an infinite square well, and the extensibility of the method to more complex systems. Using the infinite square well example, the simulation found the energy proportionality to be 1.2339 ± 0.0008, which is consistent to high precision with the known $\frac{\pi^2}{8}$ theoretical constant. Thus, we found that this method is very good at reproducing the known results to a high precision, and with good reliability, having very small variance in repeats. The method is extensible to higher dimensions, with a linear increase in time-complexity for each added dimension.**

I. Introduction

THE developments in computer power in recent years have opened up the possibility for novel computational solutions to many problems that were otherwise limited. We now have access to computers that can numerically solve problems and run simulations that were previously too computationally intensive to do. One such problem is the calculation of the ground state energies of different quantum systems. Many methods exist for solving these problems, using both classical and quantum computational approaches. Whilst generally it may be true that Quantum computing approaches are more suited to solving such problems [1], these approaches are still in infancy and are not readily accessible. Thus, there exist an abundance of classical computing approaches, such as the Variational-Relaxation algorithm [2]. This is a particularly useful algorithm for two-dimensional systems with non-separable potentials but quickly becomes difficult to implement for more complex systems.

In this investigation, we explore an alternative Monte Carlo method using Random Walk simulations, and demonstrate its efficacy with simple systems. This approach has applications in many areas of Quantum Physics and is values due to its effectiveness for more complicated systems.

II. Theoretical Background

A. Random Walk Simulations

Random walks describe motion as a series of steps in some vector space, with the direction of each step being random, i.e. with there being an assigned likelihood of a step in each possible direction. Consider a simple 1-dimensional line and a walker standing at a point labelled the origin. The walker takes 20 steps, the direction of each step (forwards or backwards on the line) is random. The position of the walker at each step is shown in Figure 1.

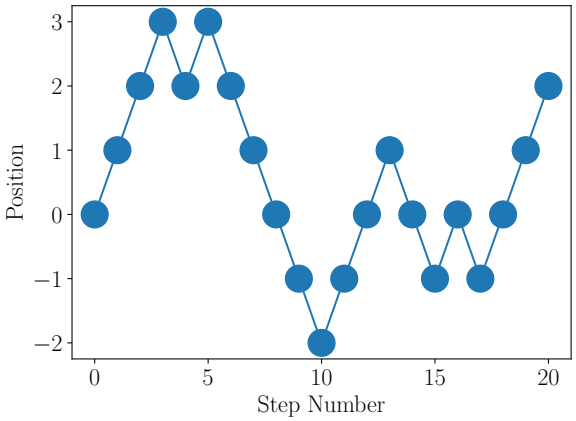


Fig. 1. The position of a walker at each step in a 20-step random walk simulation. The average position of the walker does not necessarily remain at the origin as one might expect. Instead it is normal for the walker to finish the walk at some distance from the origin, as shown here.

As we can see from the figure, it is not uncommon for the average position of the walker to be away from the origin once the walk has concluded. This is because the position of the walker at any step is dependent only on the position at the previous step. If the walker happens to take a number of steps in the same direction, it is highly unlikely for the walker to take an equal number of steps in the opposite direction, in order to return to the origin. Thus, in such a case, it is likely that the walker will conclude the walk in a position away from the origin.

In this example, the walker had an equal likelihood of stepping forward or backwards, but we can add a bias in a direction if we want to model a particular phenomenon. The walk is also easily extensible to multiple dimensions, as for each extra dimension, one extra random number must be drawn. This gives a linear increase in time-complexity, which is convenient for more complex systems.

B. Application to Quantum Systems

This investigation applied Random walk simulations to calculate the ground state energies of Quantum Systems by solving the Schrödinger equation, as shown in [3].

We can extend the simple one-dimensional line model from Figure 1 to a quantum system by imagining it as a one-dimensional simple lattice system. A walker can start at a position labelled the origin, and then proceed in a random walk. The random walk concludes if the walker "dies" on a particular step. The probability that the walker will die is dependent on a defined potential at that point. For example, in an infinite square well system, the potential is 0 within a well, and infinite outside the well. So the walker will have no chance of dying whilst inside the well, but will die if they step outside.

Let us define $p(j, n+1)$ as the probability that the walker will be at position j after $n+1$ steps and $a(j)$ as the probability that the walker will die at position j. Thus, for this to be the case, the walker would have had position $j \pm 1$ after n steps. Thus, we can say:

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

If $a(j)$ is small and n is large, we can use Taylor's expansion to expand each term in this expression and obtain a differential equation. If we then make an ansatz substitution:

$$p(j, n) = q(j) \exp(-\lambda n) \tag{1}$$

where $q(j)$ is an arbitrary function and λ is the rate of death, the differential equation in $p(j, n)$ simplifies to:

$$-\frac{1}{2}\frac{d^2 q(j)}{dj^2} + a(j)q(j) = \lambda q(j) \tag{2}$$

The full derivation is taken from [4] and is shown in Appendix A. Notice that Equation 2 is in the form of the Schrödinger equation:

$$\frac{-\hbar^2}{2m}\frac{d^2\psi}{dr^2} + V(r)\psi(r) = E\psi(r) \tag{3}$$

Comparing Equation 2 and Equation 3, we can see that the probability of death $a(j)$ is related to the potential $V(r)$, and the rate of death λ is related to the Energy E.

We can extend this to an infinite square well by choosing our probability of death such that:

$$a(j) = \begin{cases} 0, & \text{if } j < J \\ \infty, & \text{if } j \geq J \end{cases}$$

where J is a well-defined boundary. So the walker cannot die inside the well but will immediately die when they step outside, which we expect will mostly happen given sufficient time. Inside the well, the potential is simply that of a free particle [5]:

$$\frac{-\hbar^2}{2m}\frac{d^2\psi}{dj^2} = E\psi(j)$$

where x is the position of the walker. By making the substitution $x = J\frac{j}{R}$, such that $j = R$ corresponds to the walker being at the boundary, we find:

$$-\frac{\hbar^2}{2m}\frac{R^2}{J^2}\frac{d^2\psi}{dj^2} = E\psi(x)$$
$$-\frac{1}{2}\frac{d^2\psi}{dj^2} = \frac{mJ^2E}{\hbar^2R^2}\psi(x)$$

If we compare this to Equation 2, allowing $a(j) = 0$ inside the well, we see that:

$$E = \lambda R^2 \frac{\hbar^2}{mJ^2} \tag{4}$$

where only λ is an unknown quantity. Thus, by calculating rate of death λ, we can calculate the ground state energy of the infinite square well. The known ground state energy of an infinite square well is $\frac{\pi^2}{8}\frac{\hbar^2}{mJ^2}$ [6], so we expect to find λR^2 close to $\frac{\pi^2}{8}$.

To find the rate of death λ, we consider Equation 1. With some simple manipulation:

$$p(j, n) = q(j)\exp(-\lambda n)$$
$$\log(p) = \log(q\exp(-\lambda n))$$

$$\log(p) = -\lambda n + log(q) \tag{5}$$

Thus, by plotting the logarithm of the probability of surviving n steps against the number of steps n, we can obtain the death rate λ from the gradient of the line.

III. Method

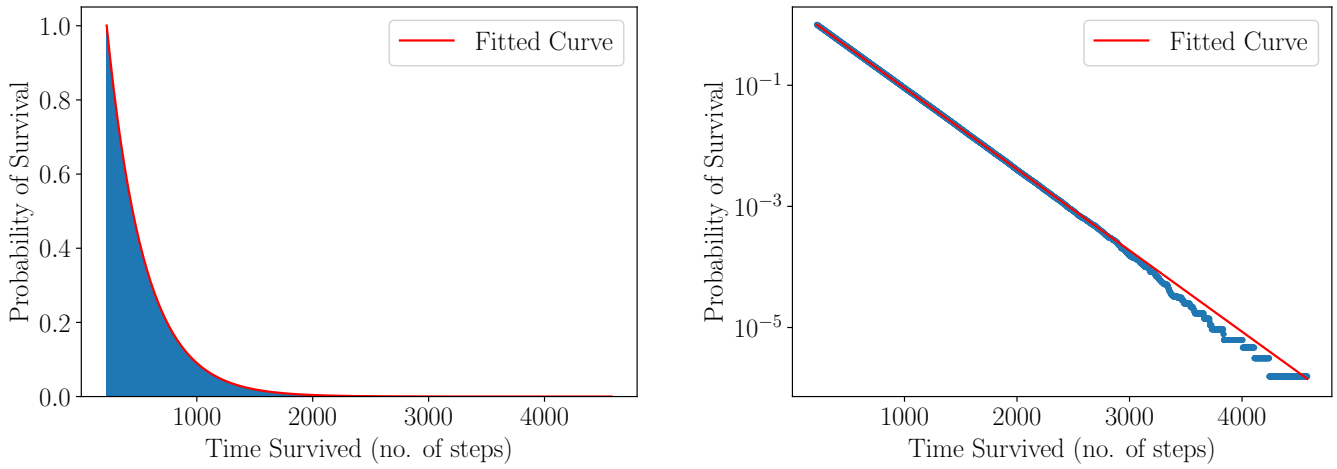
We implemented the Random Walk simulation computationally in C++ in order to optimise the performance of the simulation, and thereby allow for a larger number of iterations and larger potential wells. The source code is available [7]. The simulation is easily extensible to multiple dimensions with a linear increase in time-complexity as the number of dimensions increases, but this investigation focused on a one-dimensional system.

The random direction is selected using a pseudorandom number generator (pRNG) known as the Mersenne Twister. This algorithm was chosen as it is a very fast implementation of a pRNG [8], passes many statistical randomness tests [9], and has a very large period of $2^{19937} - 1$.

We initially ran the simulation for a boundary of $J = 20$, with 1000000 walkers. The step on which each walker died was recorded so that a probability distribution showing the likelihood of dying on a particular step could be constructed.

We then considered the probability that a walker will survive n steps, and use the relationship between the number of walkers surviving n steps and n to determine the rate of death λ, as shown in Equation 5.

One method of reducing the computing time of the simulation is to introduce a maximum number of steps. The walker will die if they reach the boundary of the well, or if they reach this maximum number of steps, whichever happens first. The effect of this method on the result is investigated in this paper.



(a) Linear Scale

(b) Log Scale

Fig. 2. Histograms showing the Probability of surviving to at least a particular step in an infinite square well with $J = 20$. The peak of 1 does not start at the origin but at step 100. This is because these steps were neglected in the plot as prior to this step the exponential shape is less clear. An exponential decay curve provides a good fit, with an R^2 value of 0.9999998. Plotting on a logarithmic scale shows the straight line fit, the gradient of which is λ, giving $\lambda J^2 = 1.2334$, with a negligible error. The fluctuation in the higher step number regions is due to the large variance in the small number of walkers that die at these steps.

IV. Results and Discussion

We first investigated an infinite square well with boundary $J = 20$ and 1000000 walkers. We found the energy constant to be 1.2339 ± 0.0008, calculated from the probability distribution of survival at a particular step. The probability that a walker will survive at a particular step is simply the 1 minus the normalised cumulative sum of walkers dying at each step. Figure 2 presents this probability and confirms the ansatz of the exponential function. The curve has been fitted with an exponential function of the form in Equation 1. The gradient of Figure 2b allowed for calculation of death rate which from Equation 4 gives the calculated energy.

Figure 2 results in a very good fit, with an R^2 value of 0.9999998, indicating the data does not vary much from the curve of best fit. This is likely due to the very large number of walkers used in this experiment, as allowed by the performant C++ implementation. We can see this better in Figure 2b, which is the same exponential plot fitted on a logarithmic scale. The closeness of the data to the line of best fit is evident here. The error in the fitted parameters is negligible, as shown by the R^2 value. The gradient of the fitted line is taken to be λ and thus allows us to calculate $\lambda J^2 = 1.2334$, compared to the theoretical value of $\frac{\pi^2}{8} = 1.2337$, quoting both values to the precision at which they vary. Since the numerical error in the fits is negligible, it cannot be propagated into an error in the final answer. Thus, we get the error in the final answer from repeated runs of the simulation and taking the standard deviation of the many values.

As we can observe in Figure 2a, the peak of the curve does not lie at the origin as we might expect from the fact that all walkers should survive at least the first step. The reason for this is twofold:

- For a well with boundary J, it is not possible for the walker to die in a number of steps less than J
- It is highly unlikely that the walker will die at a number of steps greater than but still close to J since it would require walking in the same direction a disproportionate number of times.

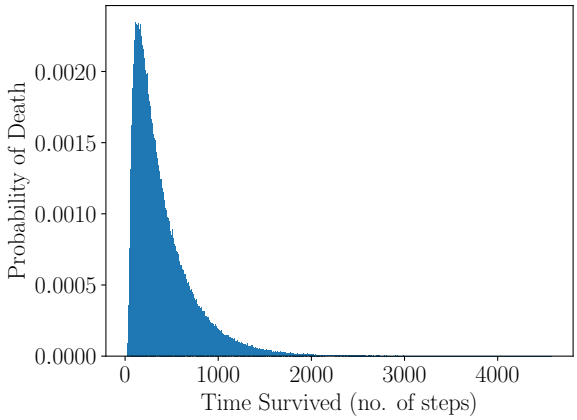


Fig. 3. The probability of dying on a particular step in an infinite square well with $J = 20$. As can be seen, the peak is away from the origin. This is because, for a well with boundary J, the minimum number of steps a walker must take is J. However, it is still unlikely that the walker will die on this minimum number of steps, as then the walker must walk in the same direction every time. Thus, the most probable step for the walker to die on is greater than J. It is for this reason that the exponential decay does not begin at the origin in Figure 2a, meaning some earlier steps must be neglected in order to achieve a better fit.

These effects are evident in Figure 3, which presents the probability that a walker will die at a particular number of steps. We can see that the peak of the curve is very clearly at a step that is greater than 20, which was the J value used in this

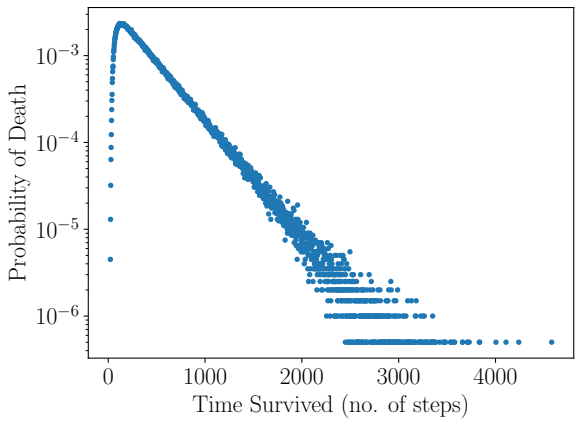


Fig. 4. The probability of dying on a particular step in an infinite square well with $J = 20$ as in Figure 3, plotted on a logarithmic scale. As explained for Figure 3, the peak probability is on a step greater than J. This is clear in this plot. It is also evident that there is significantly more spread in the points at higher steps. This is due to there being significantly fewer walkers dying on these steps, so the relative differences are higher. This explains the fluctuations seen at higher steps in Figure 2b.

simulation. Thus, for low step numbers, the shape of the curve is not exponential. For this reason we used a simple algorithm to identify a suitable number of steps to skip before fitting the exponential curve. It is for this reason that the histogram in Figure 2a does not start at the origin.

We observe some fluctuation in the data in the higher steps in Figure 2b. This is because there are so few walkers surviving to this many steps that there is more significant variance in their relative numbers. We can observe this in Figure 4. At higher step numbers, the spread is significantly larger due to the greater relative variance in the small number of walkers surviving to this many steps. When this data is cumulatively summed to calculate the probability of survival shown in Figure 2b, the spread is present as the fluctuations.

It is possible to remove this variance by introducing a cutoff number of steps, i.e. a maximum number of steps at which the walker will automatically die, regardless of whether they reached a boundary or not. Thus, there will not be such high variance in the number of walkers surviving to high number of steps. This is presented in Figure 5b. While the Figure 5b fit is observed to be better, with a higher R^2 value, some systematic error has been introduced since there will now be a disproportionately high number of walkers dying at the cutoff step as can be seen in Figure 5a. Thus, the final λJ^2 value will be less accurate. On a faster computer, with more efficient code, it is possible to forgo the step cutoff and allow the simulation to run until all the walkers die. This may take much longer but will ensure there will not be such a systematic error introduced. This is the method used in this investigation.

It is also necessary to investigate how varying the boundary J affects the accuracy of the final answer. Figure 6 presents the probability plot from Figure 2b for many different boundary J values. No step cutoff was introduced. The calculated λJ^2 value is shown in the legend. The gradient evidently changes, since the expected highest number of steps would increase as the boundary is larger, as there is more space for the walker to move before reaching a boundary. As shown in the legend, the calculated λJ^2 values get closer and closer to the theoretical value as J increases. This is due to the fact that as J increases, there are more walkers surviving to larger number of steps. Thus, we can say that using a larger value of J will result in a more accurate answer but will also be more computationally intensive and take longer. Thus, we make a trade-off between speed and accuracy and choose $J = 20$ for the final investigation.

Running the simulation 10 times with 1000000 walkers in an infinite square well with $J = 20$ and taking the average λJ^2 with error being standard deviation, we find:

$$\lambda J^2 = 1.2339 \pm 0.0008$$

We quote the precision to the decimal place at which there is variation from the theoretical result of 1.2337. We notice that the error in the calculated answer is small (since there is not much variation between repeated simulations) and the theoretical result is within the range of the calculated answer.

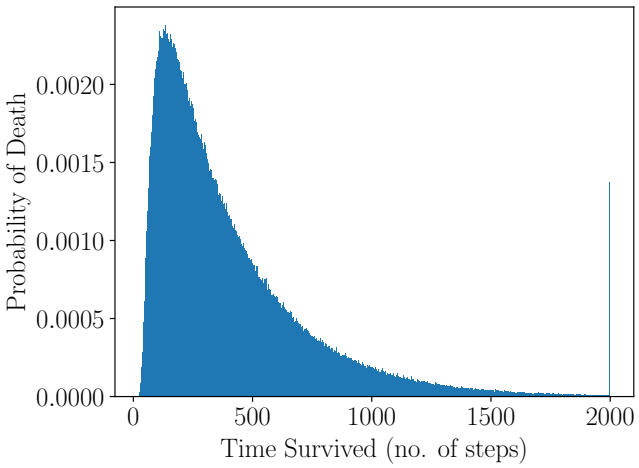
V. Conclusions

In this investigation we demonstrated a method of calculating ground state energies of a quantum system using Random walk computational simulations. The investigation focussed on a one dimensional infinite square well, but can easily be extended to more complex systems such as a circular well in two dimensions.

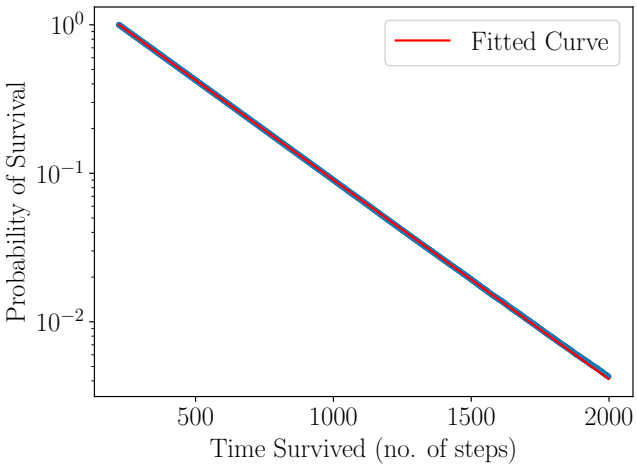
The investigation explored the effects of varying the size of the well in determining the ground state energy of a quantum system, and showed that this Monte Carlo method produces more accurate results for ground state energies of larger wells. This is because for wells with larger boundaries, the walker can take a larger number of steps on average before dying, and thus can populate larger step bins. Thus, this method produces data that is close to the theoretical result for larger infinite square wells.

The numerical result found the ground state energy proportionality constant to be $\lambda J^2 = 1.2339 \pm 0.0008$, which is consistent with the known theoretical value of $\frac{\pi^2}{8}$.

This method can be easily extended to multiple dimensions with only a linear increase in time complexity for each added dimension. Thus, while this method may not be very efficient for low number of dimensions, it quickly becomes effective at a higher number of dimensions.



(a) Probability of dying on a particlar step in Linear scale



(b) Probability of surviving to at least a particular step in a logarithmic scale

Fig. 5. Histograms showing the results of the simulation when a maximum number of steps is introduced. If a walker reaches this number of steps, they automatically die, regardless of position. This results in a disproportionately high number of walkers dying at this maximum number, 2000 in this case, as shown by Figure 5a. Although this presents no fluctuation at higher number of steps, as shown in Figure 5b, some systematic error is introduced through this method.

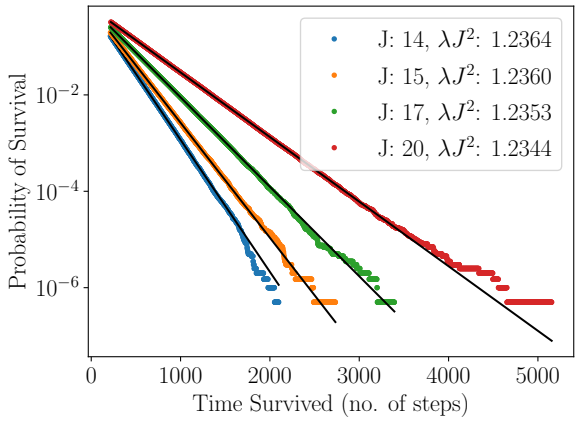


Fig. 6. Histogram plotted on a logarithmic scale showing the Probability of surviving to at least a particular step in different infinite square wells with different boundaries J. The simulations were run with 10000 walkers The λJ^2 values for each are shown. The values get closer to the theoretical value of $\frac{\pi^2}{8} = 1.2337$ as we increase the size of the well. This is likely due to more walkers surviving to a higher number of steps, as the well size is increased.

Appendix A
Random walk Schrödinger Equation Derivation

Consider a one-dimensional lattice upon which a walker is undergoing a random walker. Let us denote the probability that the walker is at position j after n steps as $p(j,n)$ and the probability that the walker will die at position j as $a(j)$. We will denote the position-step number coordinates as $[j,n]$. Thus, for a walker to have coordinates $[j,n+1]$, in the previous step, they must have had coordinates $[j\pm1,n]$. So we can say:

$$p(j,n+1) = \frac{1}{2}(1-a(j))(p(j-1,n)+p(j+1,n)) \quad (6)$$

If $a(j)$ is small and n is large, then we can say $p(j,n)$ is a continuous function and we can use Taylor expansion to expand each of $p(j+1,n)$, $p(j-1,n)$, and $p(j,n+1)$:

$$p(j+1,n) = p(j,n) + \frac{\partial p}{\partial j} + \frac{1}{2}\frac{\partial^2 p}{\partial j^2} + ...$$

$$p(j-1,n) = p(j,n) - \frac{\partial p}{\partial j} + \frac{1}{2}\frac{\partial^2 p}{+\partial j^2} + ...$$

$$p(j,n+1) = p(j,n) + \frac{\partial p}{\partial n} + ...$$

We can now substitute these expansions into Equation 6, we find:

$$p(j,n) + \frac{\partial p}{\partial n} = (1-a(j))\Big(p + \frac{1}{2}\frac{\partial^2 p}{\partial j^2}\Big)$$

But the probability of death $a(j)$ and the second derivative are so small, that their product is neglected. We also let $p(j,n) = q(j)\exp(-\lambda n)$ so that:

$$-\frac{1}{2}\frac{d^2 q(j)}{dj^2} + a(j)q(j) = \lambda q(j)$$

References

[1] G. Mazzola and M. Troyer, "Ground-state statistics from annealing algorithms: quantum versus classical approaches Related content Quantum Monte Carlo annealing with multi-spin dynamics," DOI: 10.1088/1367-2630/11/7/073021. [Online]. Available: http://www.njp.org/.

[2] D. V. Schroeder, "The variational-relaxation algorithm for finding quantum bound states," *American Journal of Physics*, vol. 85, no. 9, pp. 698–704, Jan. 2017. DOI: 10.1119/1.4997165. arXiv: 1701.08934. [Online]. Available: http://arxiv.org/abs/1701.08934%20http://dx.doi.org/10.1119/1.4997165.

[3] D. Ceperley, "The simulation of quantum systems with random walks: A new algorithm for charged systems," *Journal of Computational Physics*, vol. 51, no. 3, pp. 404–422, Sep. 1983, ISSN: 10902716. DOI: 10.1016/0021-9991(83)90161-4.

[4] M. Newton, *Random Walks for Quantum Mechanics Project Description*, 2020.

[5] B. Bransden, C. Joachain, N. York, S. Francisco, H. Kong Tokyo, N. Delhi, C. Town, and M. City, *Physics of Atoms and Molecules 2nd edition*. 2003, p. 157, ISBN: 978-0-582-35691-7.

[6] J. Davies, *The physics of low-dimensional semiconductors: an introduction*, 6th Reprin. 1998, p. 5, ISBN: 978-0-521-48491-6.

[7] S. Pandian, *saiduc/Random-Walk-Quantum-Physics: Using random walk Monte Carlo simulation to solve the Schrodinger Equation*, 2020. [Online]. Available: https://github.com/saiduc/Random-Walk-Quantum-Physics (visited on 04/17/2020).

[8] M. Route, "Radio-flaring Ultracool Dwarf Population Synthesis," Jul. 2017. DOI: 10.3847/1538-4357/aa7ede. arXiv: 1707.02212. [Online]. Available: http://arxiv.org/abs/1707.02212%20http://dx.doi.org/10.3847/1538-4357/aa7ede.

[9] J. Bellamy, "Randomness of D Sequences via Diehard Testing," Dec. 2013. arXiv: 1312.3618. [Online]. Available: http://arxiv.org/abs/1312.3618.