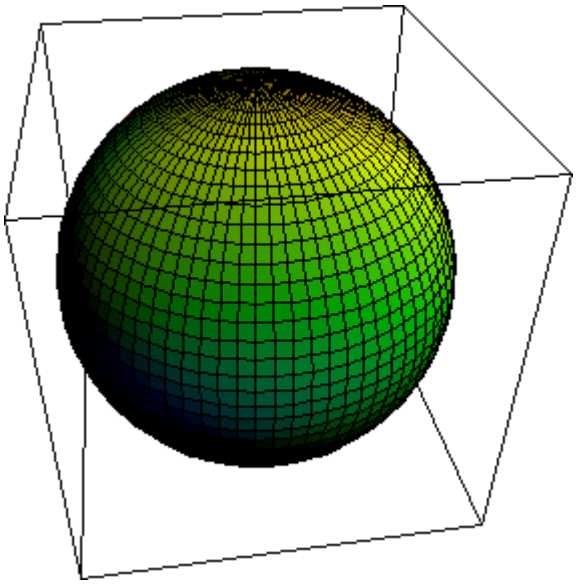
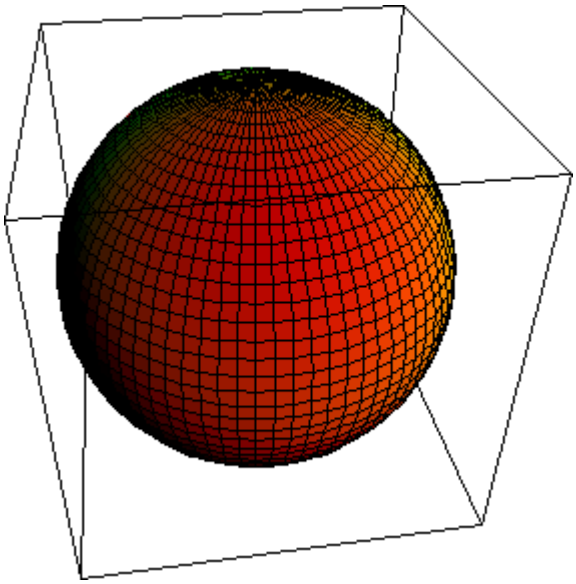
Simulation of a Quantum Random Walk



Chris Melancon

Honors Project

Spring 2005



Abstract

For my project I simulated a quantum random walk in one plus one dimensions with a ratcheting potential applied. The particle begins at the origin and after running the simulation for one hundred time steps the most probable final position was calculated and then graphed according to its initial condition. These results were analyzed to determine the dependence on initial condition of final position.

Introduction

The quantum random walk has been studied by mathematicians, primarily by David Meyer. They have been found to be discrete models for such physical processes as the Dirac equation. [1] In this project we hoped to learn the effect of the initial condition on

a given set of potentials. This simulation was first conducted by David Meyer with potentials that created a Parrondo game [3]. However in his simulation only one initial condition was used. Here I am interested in the affect of the initial condition on the simulation, primarily on the expected position.

Background: Quantum random walk versus classical random walk

Classical Random Walk

In a classical random walk in one plus one dimensions (one position dimension and one time dimension) we deal with a single particle with a velocity in only one dimension. The particle first goes through an advection stage where it moves left or right according to its velocity and then goes through a scattering stage in which the velocity either stays the same or switches according to some respective probabilities. This process is then repeated for a given number of time steps. It should be noted that here both time and position are quantized and velocity $= \pm 1$. A particle can only move an integer number of units left or right and time goes by in units or 'ticks'.

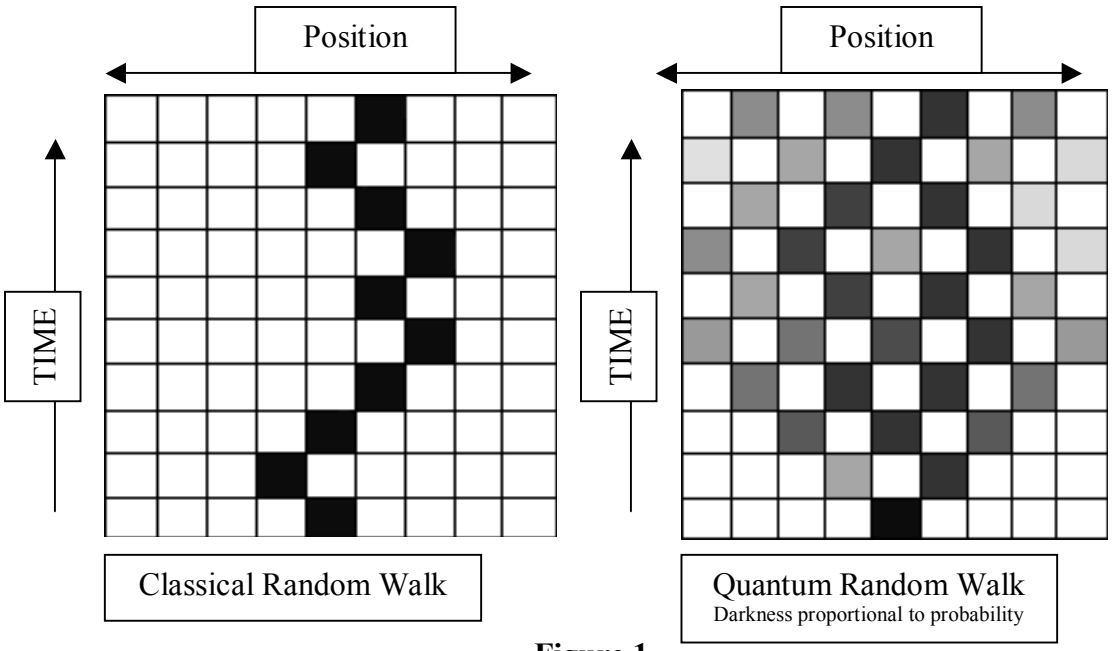


Figure 1

Quantum Random Walk

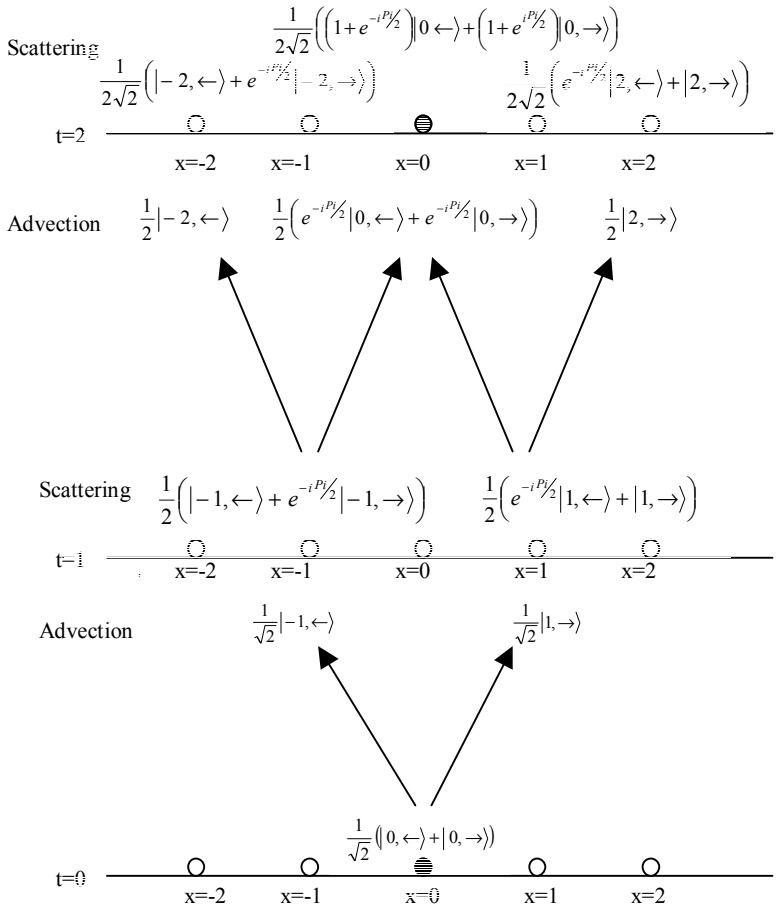
Before we begin discussing a quantum random walk we must introduce some key ideas. In quantum mechanics, particles are represented by state vectors, $\psi(x,t)$. In our simulation the state vectors take the form $\psi(x,t) = \sum_{x=-\infty}^{\infty} \left(a(x,t)\left|x,\rightarrow\right\rangle + b(x,t)\left|x,\leftarrow\right\rangle \right)$. Here $a(x,t)$ and $b(x,t)$ are complex numbers and $\left|x,\rightarrow\right\rangle, \left|x,\leftarrow\right\rangle$ are an orthonormal basis for a Hilbert space where *x* represents the position and the arrows represent the direction of the velocity. The state vectors are normalized such that $\sum_{x=-\infty}^{\infty} \left(\left|a(x,t)\right|^2 + \left|b(x,t)\right|^2 \right) = 1$. The probability for our particle to be at a position x at time t is given by $\left|a(x,t)\right|^2 + \left|b(x,t)\right|^2$. Therefore the expected position of our particle is given by $\sum_{x=-\infty}^{\infty} x * \left(\left|a(x,t)\right|^2 + \left|b(x,t)\right|^2 \right)$. The state vectors contain all the information of our system.

Also unitary matrices are used in our discussion and therefore must be explained. A unitary matrix is one whose hermitian conjugate is its inverse. The general 2x2 unitary matrix has the form $\begin{pmatrix} e^{i\alpha}\cos\theta & e^{i\beta}\sin\theta \\ -e^{i\gamma}\sin\theta & e^{i(\gamma-\beta-\alpha)}\cos\theta \end{pmatrix}$. In quantum mechanics each time step is done by multiplying one or more unitary operators to the state vector. We now have everything we need to begin our discussion of quantum random walks.

In our simulation, the particle starts localized at the origin and therefore the state vector has the form $a(0,0)|0,\rightarrow\rangle + b(0,0)|0,\leftarrow\rangle$. Since $|a(0,0)|^2 + |b(0,0)|^2 = 1$ we see our particle begins with a probability of one to be at the origin at this time. Now each time step consists of three parts which can be expressed as unitary matrices. The first of these is the advection stage. The advection stage is defined by the following action: $a(x,t) = a(x-1,t-1)$, $b(x,t) = b(x+1,t-1)$, which is unitary and preserves the normalization since it simply reassigns the coefficient values without changing them.

Next we have the scattering stage. The scattering stage is represented by a 2x2 unitary matrix where α=0 and $\beta,\gamma = \frac{\pi}{2}$. The reason for this choice is that in this form our unitary matrix is invariant under parity, something that a scattering stage is required to be[2]. In this form the matrix represents a scattering such that the particle has $\cos^2\theta$ probability to keep the same velocity or a $\sin^2\theta$ probability for it's velocity to flip, that is $a(x,t)|x,\rightarrow\rangle + b(x,t)|x,\leftarrow\rangle \Rightarrow b(x,t)|x,\rightarrow\rangle + a(x,t)|x,\leftarrow\rangle$. One important aspect of quantum mechanics that can be seen from the advection and scattering stages is that the overall phase of the state vector does not matter; only the relative phase. So multiplying all $a(x,t)$ and $b(x,t)$ by some $e^{i\phi}$ will not affect any computed probability. Let us look

at a very simple simulation. We will begin with our particle localized at the origin and our scattering matrix will have $\theta = \pi/4$.



An example of simple Quantum Random Walk, to show the necessity of the relatives phases between coefficients

Figure 2

As the above simple simulation shows the relative phases of the state-vector affect the probabilities. If they were eliminated we would lose conservation of probability. For example at t=1 after the scattering stage if we eliminate the phases then our probability to be at x=-1 is ¼ and the probability to be at x=1 is ¼. But it is also clear that if our entire state vector were multiplied by some phase then that would not affect any probability. To

give an example of this we will compute the probabilities of our particle to be -1 and 1 at the second time step and then multiply by a phase and recalculate them.

$$a(-1,1)=\frac{1}{2}\quad |a(-1,1)|^2=\frac{1}{4} \qquad a'(-1,1)=\frac{e^{-i\phi}}{2}\quad |a'(-1,1)|^2=\frac{1}{4}$$

$$b(-1,1)=\frac{e^{-i\pi/2}}{2}\quad |b(-1,1)|^2=\frac{1}{4} \qquad b'(-1,1)=\frac{e^{-i(\pi/2+\phi)}}{2}\quad |b'(-1,1)|^2=\frac{1}{4}$$

$$a(1,1)=\frac{e^{-i\pi/2}}{2}\quad |a(1,1)|^2=\frac{1}{4} \qquad a'(1,1)=\frac{e^{-i(\pi/2+\phi)}}{2}\quad |a'(1,1)|^2=\frac{1}{4}$$

$$b(1,1)=\frac{1}{2}\quad |b(1,1)|^2=\frac{1}{4} \qquad b'(1,1)=\frac{e^{-i\phi}}{2}\quad |b'(1,1)|^2=\frac{1}{4}$$

It is clear from the example that multiplying our entire vector by some overall phase does not affect the probability to be at a given location. This will help us to simplify our simulation as we will see later on.

Now along with the advection and scattering stages at each time step we can also introduce a potential to our walk. A potential in a quantum random walk is a position-dependent phase multiplication. We can see this by solving the Schrödinger equation analytically:

$$i\hbar\frac{\partial\psi}{\partial t}=H\psi+V(x)\psi$$

$$i\hbar\frac{\psi(t+\Delta t)-\psi(t)}{t+\Delta t-t}=H\psi(t)+V(x)\psi(t)$$

$$i\hbar\,\psi(t+\Delta t)-\psi(t)=\Delta t\big(H\psi(t)+V(x)\psi(t)\big)$$

$$\psi(t+\Delta t)=\frac{1}{i\hbar}\big(\Delta t\big(H\psi(t)+V(x)\psi(t)\big)+\psi(t)\big)$$

$$\psi(t+\Delta t)=\psi(t)\left(1-\frac{i}{\hbar}\Delta tV(x)\right)\left(1-\frac{i}{\hbar}\Delta tH\right)\qquad up\ to\ o(\Delta t^2)$$

$$\psi(t+\Delta t)=\psi(t)e^{-iV(x)\Delta t/\hbar}e^{-iH\Delta t/\hbar}$$

The last step comes from the Taylor series expansion of e^{-x} neglecting the $o(\Delta t^2)$ term. We will set $\hbar=1$ and we now have that a potential V(x) is applied by $e^{-iV(x)}$ [2]. This potential will be applied after the scattering stage and is the final part of

each time step. It should be noted that the H term is our advection and scattering stages. Physically a potential represents a force $-\nabla V$ applied to a particle. $F = -V'(x)$.

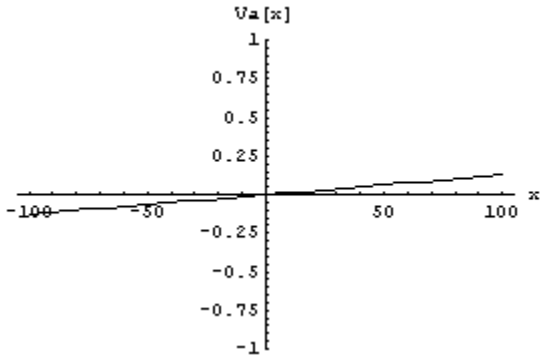
Set-up

To begin, our scattering matrix is fixed for the entire simulation. Our potential is a ratcheting potential (more detail to come). Our simulation is started from many different initial conditions, all of which begin localized at the origin. The initial conditions are different choices of a(0,0) and b(0,0). These can be represented by letting $a(0,0) = e^{i\phi}\cos\theta/2$ *and* $b(0,0) = e^{i\gamma}\sin\theta/2$. The overall phase of our a and b do not matter as we demonstrated above, only the relative phase so we can let them take the form $a(0,0) = \cos\theta/2$ *and* $b(0,0) = e^{i\phi}\sin\theta/2$. They can be represented by positions on a unit sphere where θ ranges from 0 to π and ϕ ranges from 0 to 2π. This unit sphere is called a Bloch sphere. After the simulation is run the most probable final position is calculated and color coded at its appropriate place on the Bloch sphere.

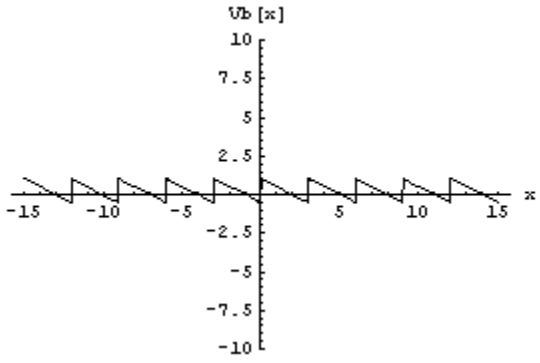
Potentials

There are two different potentials used in this simulation. The first potential, which we will call $V_a(x) = \frac{2\pi}{5000}x$ is a simple linear potential with a positive slope. The second is a saw tooth potential which we will call $V_b(x) = V_a(x) + \frac{\pi}{3} - \frac{\pi}{3}\frac{x \bmod 3}{2}$. These potentials are applied in the repeating order of V_a, V_a, V_a, V_a, V_b on consecutive time steps.[1,2]

The reason for these potentials is that when applied to the initial condition where $\theta = Pi/4$ and $\phi = 0$ these create a quantum Parrondo game [3, 4].



A graph of the potential $V_a(x)$

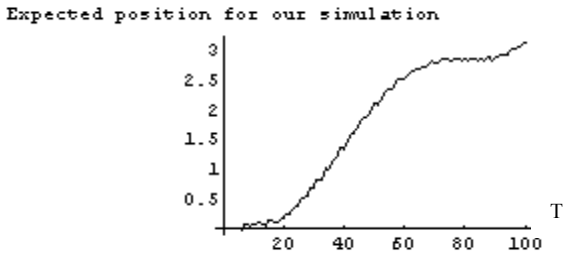


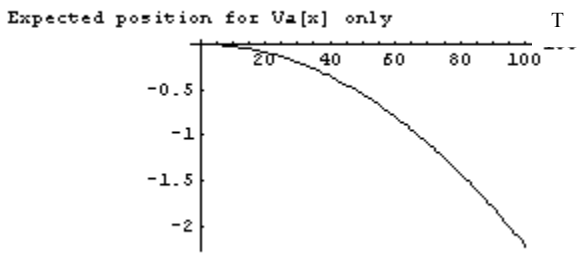
A graph of the potential $V_b(x)$

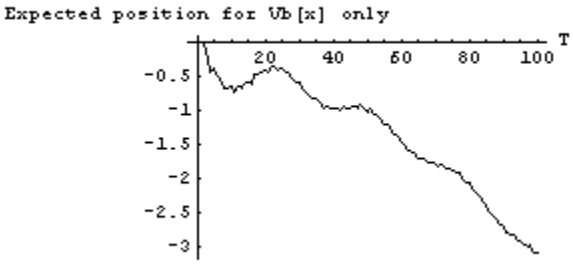
Figure 3

Parrondo Games

A Parrondo game is one in which two losing games are combined to make a winning game. In our case if only potential *a* is applied then the expected position is slightly negative, same for potential *b*. But when the above pattern is applied the expected position is positive. We will see later that our simulation is a Parrondo game for some positions but not all. Below we have three graphs that demonstrate this aspect of our simulation. The graphs represent the expected position as a function of time. As the graphs show the expected position of the pattern V_a, V_a, V_a, V_a, V_b is positive, while the expected position from either potential alone is negative.



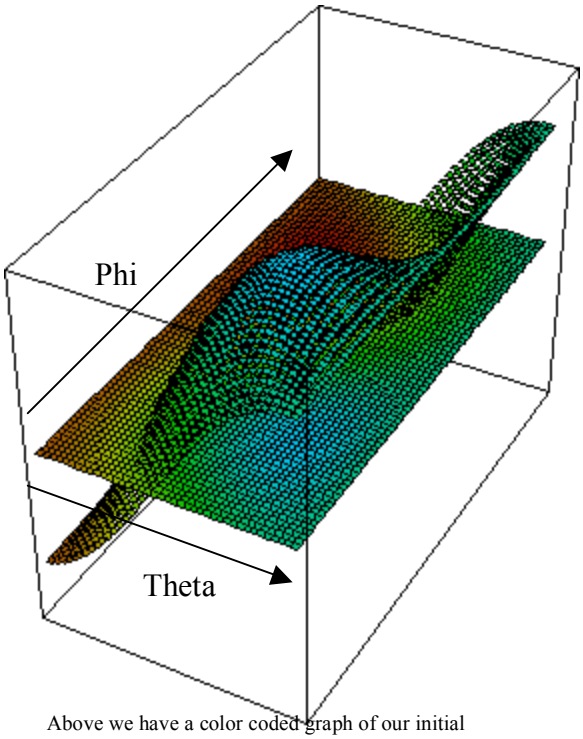




Above we see why our simulation is a Parrondo game for some initial conditions. The Individual potentials both drive the particles expected positions negative while the alternating pattern of them drives the expected position positive

The Simulation

Our simulation was run using the above potentials for a total of one hundred time steps. The θ and ϕ were done in steps of $\pi/80$. This gave us a resolution of the Bloch sphere with 3200 different positions. Below a graph of the results can be seen. The image is color coded such that blue represents the maximum expected position and red represents the minimum expected position. On top of the color coded image is the same image with the z-axis representing the expected position. We can see from the image that the expected position gradually changes from positive to negative as a function of θ and ϕ. The maximum and minimum can be seen in the table below.

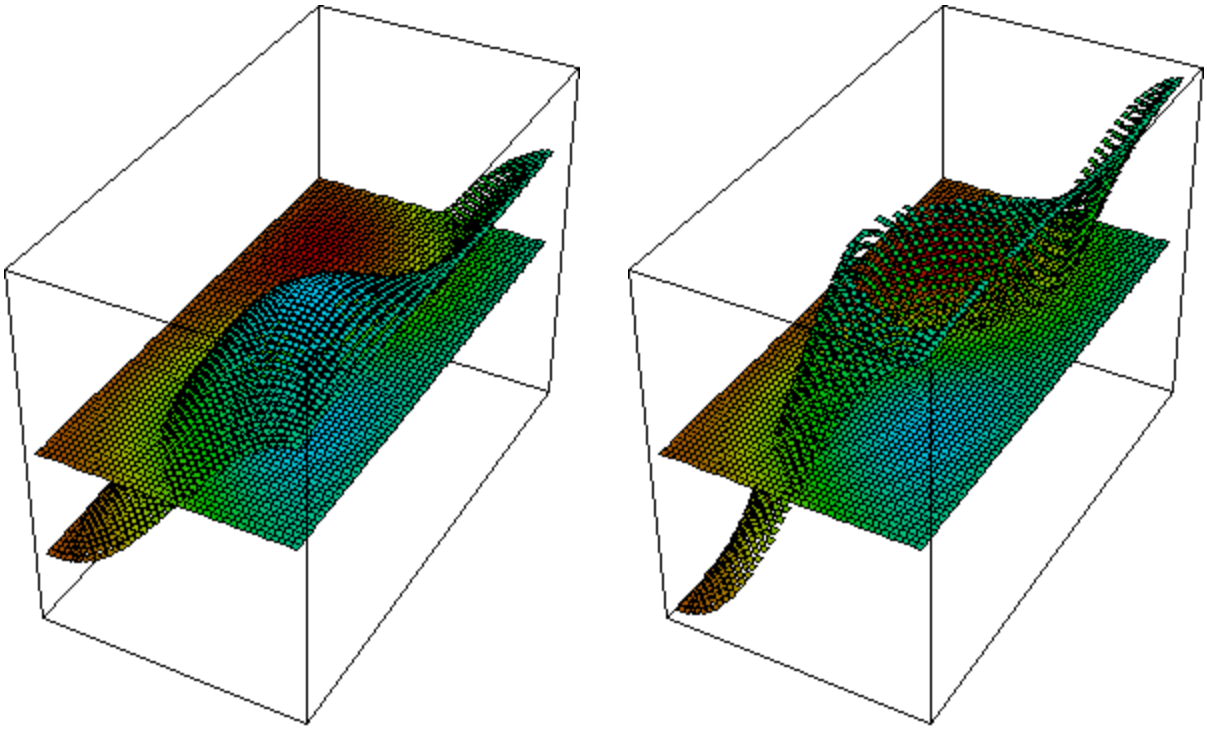


Above we have a color coded graph of our initial conditions versus expected final position

Results from simulation

Maximum Expected Position	ϕ at max	θ at max	Minimum Expected Position	ϕ at min	θ at min
29.4677	37 π /80	59 π/80	-28.3845	117 π/80	21 π/80

Before we begin to analyze our results let's compare them to the simulation where only one of our two potentials is applied. If we only apply $V_a(x) = \frac{2\pi}{5000}x$, we get the results seen at the right. If we only apply $V_b(x) = V_a(x) + \frac{\pi}{3} - \frac{\pi}{3}\frac{x \bmod 3}{2}$ then we get the results on the left.



On the left we have the results of a simulation using only the potential $V_a(x)$ and on the right we have the results from only using $V_b(x)$. The similarity to the simulation using the alternating potentials is quite obvious.

Figure 6

The first thing one will notice is the similarity to our result when the alternating pattern of potentials is applied. The other interesting observation is that our simulation is not a Parrondo game for all initial conditions. In fact for most initial conditions all three simulations had similar expected positions. These graphs are only shown for comparison and for the rest of the paper we will only be working with our original simulation.

Analysis

A few interesting facts can be seen from the maximum and minimum expected positions. First is that our expected position can be both positive and negative. It also appears that

we have an equal number of initial conditions that give us positive expected positions as negative expected positions. It is also seen that that the initial conditions that give us the maximum and minimum expected positions are diametrically opposite. We see that the ϕ where the maximum and minimum expected position occur are out of phase by π and the θs add up to π.

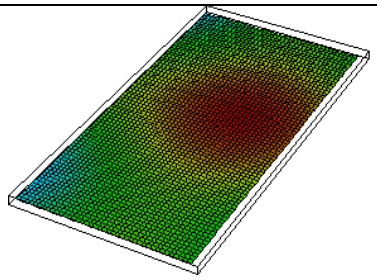
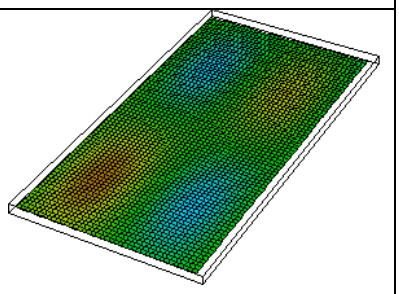
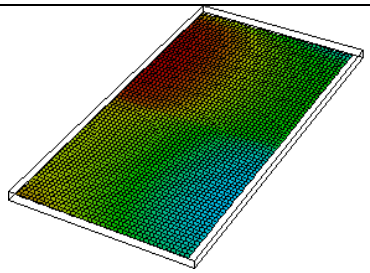
With this sort of symmetry we would expect our maximum and minimum expected position to have equal magnitude but we see that this is not the case. Now this could be attributed to round-off error since our simulation doesn't keep exact answers because of the time and size costs but this is very unlikely because of the accuracy that is still kept by Mathematica. It could also be attributed to the fact that the above positions aren't the exact locations of the absolute maximum and minimum expected positions because only a grid of discrete θ and ϕ values were used. It can be seen that the θs are very close to π/4 and 3 π/4 for example. Therefore a more finely meshed set of data was taken around these locations to see if it would lead to new points and if it would give maximum and minimum expected positions of equal magnitude. We will discuss these results in a following section.

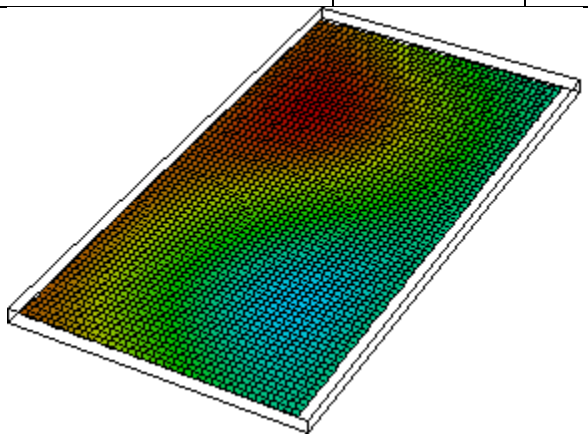
A functional fit to the results would be very beneficial to have in order to infer some relation to the potentials and number of time steps so some significant effort was devoted to finding a fitting function.

A Functional Fit

Visual observation of the shape of the expected position versus initial condition graph suggest a basic function consisting of sines and cosines of θ and ϕ. Therefore several

attempts were made to guess the general functional form of the fit. Two separate approaches were used to find the fit of our results. The first was to use the FindFit function in Mathematica. With this function a general equation is entered as a potential guess and the program finds the best coefficients to fit our guess to the actual results. Our second method was very similar but instead a general equation is entered into the formula for a chi-square test ($\chi^2 = \sum_{\theta}\sum_{\phi} \frac{(E(\theta,\phi) - f(\theta,\phi))^2}{E(\theta,\phi)}$, $E(\theta,\phi)$ is our test equation and $f(\theta,\phi)$ is our results) and then the coefficients that minimized the chi-square value were found using a minimization function built into Mathematica. The FindFit proved to be quicker and produce better results so it was used primarily. However the chi-square values of the fit determined by FindFit were computed along with the graph as a test of the goodness of the fit. A table containing some of these fits can be seen below. It is clear from the table that none of these fits does a good job of fitting the results. Many of the functions only capture one aspect or another of the actual results. This attempt was abandoned for a more systematic approach.

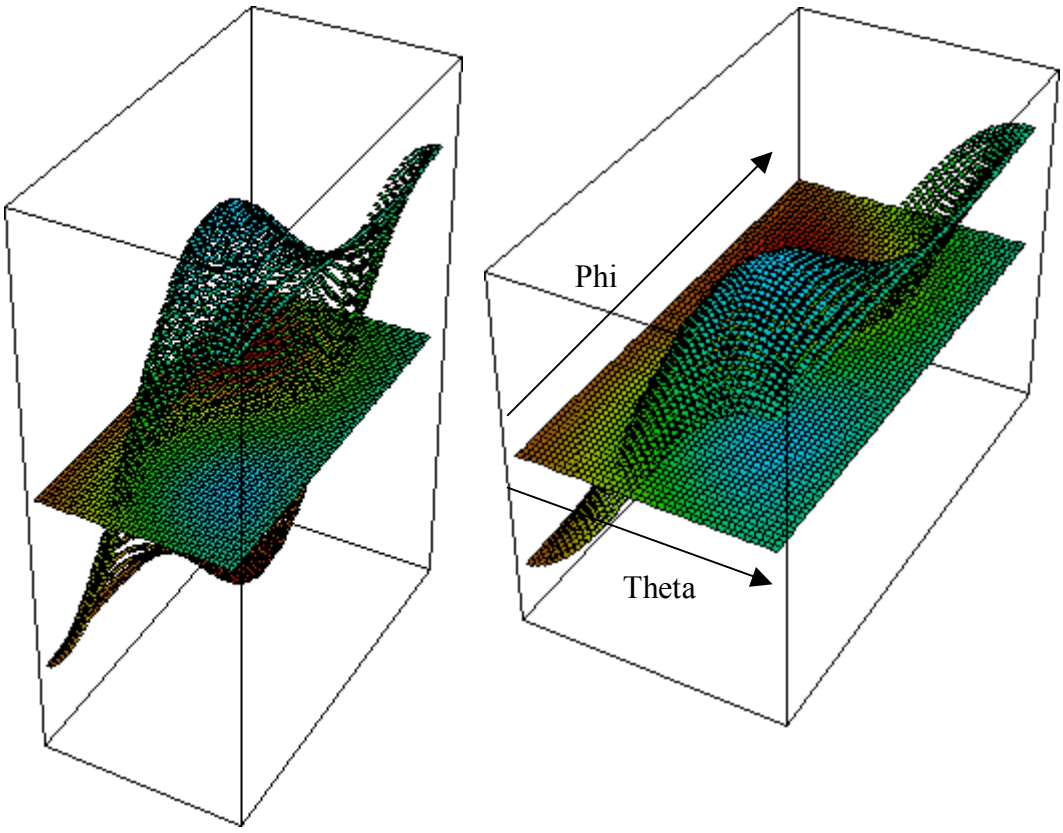
Test Function	Coefficients	Chi-Square Value	Graph
$a*\sin(b*\theta+c)+g*\cos(d*\phi+f)+e$	a=0.9932 b=1.0217 c=1.0023 d=1.0021 e=0.9954 f=1.0142 g=0.9925	- 3.44823*10^13	
$a*\sin(b*\theta+c)*\cos(d*\phi+e)+f*\theta+g*\phi+h$	a=9.2614 b=0.9058 c=3.5775 d=1.9777 e=-.51884 f=0.95003 g=0.37891 h=12.9113	102958	
$g*\sin(b*\theta+c)*\cos(d*\phi+e)+f$	a→1 b→1 c→3.26874 d→1 e→-1.5708 f→0.541595 g→-20.719	-69533.7	



The result of our simulation for comparison

Since it appears that the function would consist primarily of sines and cosines, the discrete Fourier coefficients were computed. It would appear the Fourier series would be the best way to analytically determine such a function but this approach also did not work. The problem with this approach is that the Fourier series is for solutions on a torus. If our θ and ϕ were to both run from 0 to 2 π then this approach would have been correct and hopefully provided some meaningful results but as the case is it is not correct and did not produce meaningful results.

In the spirit of the Fourier series it was decided to find the spherical harmonic series that would approximate our result. This seemed like a good choice since spherical harmonics form an orthonormal basis on a sphere. The following formula was used to determine the coefficients of the series: $c_{l,m} = \sum_{\phi}\sum_{\theta} f(\theta,\phi)Y_m^l(\theta,\phi)^* \Delta A$. In the above formula $f(\theta,\phi)$ is the expected position from our simulation for a given θ and ϕ, Y_m^l is the spherical harmonic and ΔA is the segment of area surrounding the given point. After the coefficients were calculated the following series was constructed: $g(\theta,\phi) = \sum_{l}\sum_{m} c_{l,m}Y_m^l(\theta,\phi)$. This did produce a fit to our simulation results but it was not exact. For this fit we allowed the l and m to range from -10 to 10. Below we have the values of the l=0 row as an example. The chi-square value of this fit was 1743.34. This was much better than our other attempts but still quite large. We are left with no good functional formula for the results of our simulation. Even without a good formula to work with it is still possible to do some further analysis of the results.



The graph of the spherical harmonic fit to our function. This fit captures some key aspects of the results but it is not as close as we would like to the original on the right.

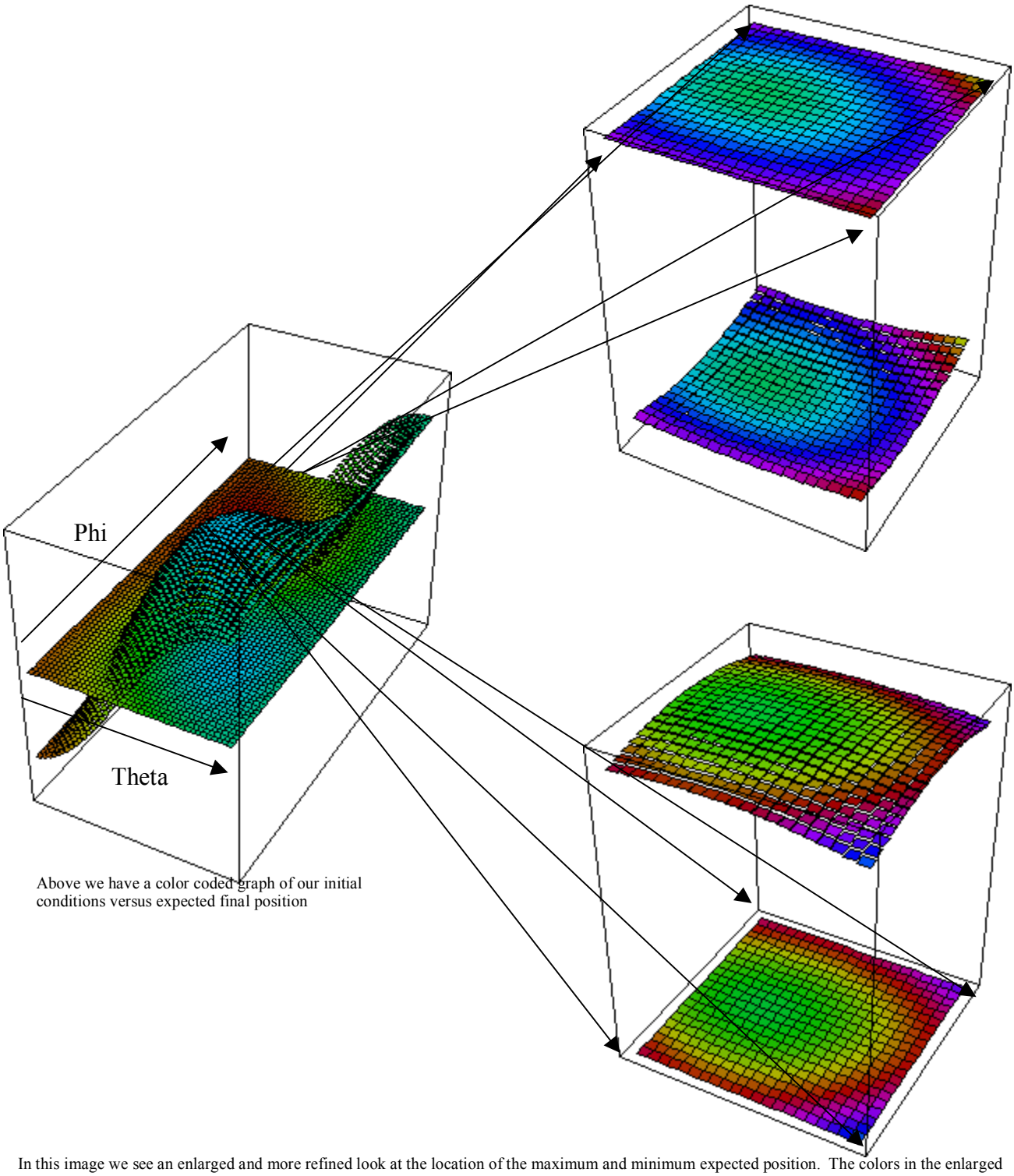
Maximum Expected Position	ϕ at max	θ at max	Minimum Expected Position	ϕ at min	θ at min
58.965	37 π /80	59 π/80	-56.7975	117 π/80	21 π/80

For the l=0 state here are the coefficients from m=-10 to 10:

{-0.0411,-0.1594,-0.0362,-0.3198,-0.0308,-0.853,-0.0244,-5.0903,-20.6827,19.1735,19.1735,-20.6827,-5.0903,-0.0244,-0.853,-0.0308,-0.3198,-0.0362,-0.1594,-0.0411,-0.0924}

Absolute Maximum and Minimum Expected Position

In the graph below we see an enlarged and more refined look at the location of the maximum and minimum expected positions. It turns out that our original guess of the θs being equal to $\pi/4$ and 3 $\pi/4$ were correct. The ϕs did turn out to be near $\frac{3\pi}{2}$ and $\frac{\pi}{2}$ but different. The initial conditions that give the maximum and minimum expected position are still diametrically opposite. In the end the maximum and minimum expected positions did not have equal magnitude but very close to it. It is most likely that it is a real difference. At first glance this result seemed promising since the $\pi/4$ result would seem to imply that the initial condition with equal of a and b is most affected by the potentials. However this is not the case. Because of how our initial conditions are described $\pi/4$ does not represent that at all. $\pi/2$ would represent the initial condition with equal magnitudes since $a(0,0) = e^{i\phi} \cos \theta/2$ *and* $b(0,0) = e^{i\gamma} \sin \theta/2$. So no obvious reason for this position to be the maximum expected position seems to exist. Similarly a $\phi = \pi/2$ was tried since it is very close the where the maximum and minimum ϕ are located but unfortunately they not the real max or min. So we are not able to fully explain the initial condition resulting in the max or min position.



Above we have a color coded graph of our initial conditions versus expected final position

In this image we see an enlarged and more refined look at the location of the maximum and minimum expected position. The colors in the enlarged graphs are reset to help show the features of the areas. The graph runs from, $\frac{15\pi}{100} \le \theta \le \frac{35\pi}{100}, \frac{140\pi}{100} \le \phi \le \frac{160\pi}{100}$ for the top right graph and $\frac{65\pi}{100} \le \theta \le \frac{85\pi}{100}, \frac{40\pi}{100} \le \phi \le \frac{60\pi}{100}$ for the bottom right graph, both in increments of $\frac{\pi}{100}$. The inverted, yet nearly identical look to these enlarged graphs shows the symmetry that is prominent in our simulation.

Figure 7

Results from refined simulation

Maximum Expected Position	Max ϕ	Max θ	Minimum Expected Position	Min ϕ	Min θ
29.4763	23 π/50	3 π/ 4	-28.3931	73 π/50	π/4

Conclusion

This simulation provided some interesting results that turned out to be quite a challenge to analyze. This simulation really shows the unique and complicated behavior that quantum random walks have.

This project has given endless angles that can be explored and many new variations of the simulation that could be tried. Simple things like changing the pattern of the potentials would undoubtedly give completely different results as would simply increasing the number of time steps. More than one particle could have been simulated and different scattering matrices could have been used. More dimensions could have been added or a more refined Bloch sphere used for the initial conditions. Endless possibilities but unfortunately factors such as computing power, computing time and human time are the biggest restraints.

Given more time I would have like to further explore the spherical harmonic series to try to find one that gives a better fit to our results. Also I could have tried to derive the functional form of the results perturbatively if time allowed. I found myself continually finding new things to try even as I wrote this paper. I am disappointed that I was not able to find a good functional form of the results since that would have allowed for a whole new angle of analysis.

I also can not here describe the code used for this simulation, modified from the original written by Professor David Meyer, or the many modules and smaller programs that I wrote and used in the long analysis portion of this project. I feel that as much of the learning process took place not only in learning the math and physics behind the simulation but in creating the code and learning Mathematica to the degree needed to do this project. During the writing process I found myself beginning to really have a firm grip on the basic components of the quantum random walk and the basic quantum mechanics behind it because it is not until you try to explain something to someone else that you really grasp the knowledge.

References

(1) D. A. Meyer, "Quantum mechanics of lattice gas automata: One particle plane waves and potentials",
quant-ph/9611005; Phys. Rev. E 55 (1997) 5261–5269.

(2) David A. Meyer, ``From quantum cellular automata to quantum lattice gases'', *Journal of Statistical Physics* **85** (1996) 551-574;

(3) David A. Meyer and Heather Blumer, ``Parrondo games as lattice gas automata'', *Journal of Statistical Physics* **107** (2002) 225-239.

(4) David A. Meyer, ``Noisy quantum Parrondo games'', in D. Abbott, J. H. Shapiro and Y. Yamamoto, eds., *Fluctuations and Noise in Photonics and Quantum Optics*, *Proceedings of SPIE* **5111** (2003) 344-350.