Random Walks on Spheres
End of Term Report Summer 2013

Annie Brunelle, Emily Diana, Jeff Dzugan
Ryan Gallagher, Elisabeth Johnson, David Wegscheid

July 26, 2013

We are examining a classical Kakutani result on the relationship between Brownian motion, a form of random movement, and harmonic functions, which are solutions to Laplace equation. We will use this result to numerically solve Laplace's equation in certain regions with various boundary conditions via the Random Walks on Spheres method. Looking at several regions, we will discuss the distribution of the point of first encounter with the boundary. We will also approximate the solutions using Gaussian quadrature, a method which allows us to have incomplete data concerning the boundary conditions. Lastly we consider the accuracy of our numerical solutions.

Contents

1 Introduction

1.1 Theory

Suppose that you are looking at some region, say a square or a sphere, that is heated solely on the boundary, and you know that the region is in its steady-state (temperature is no longer changing). Now suppose that you want to determine the temperature at any given point on the region–this requires solving Laplace's equation on the region. In other words, we want to find a function u such that $\delta u = 0$ inside the region and $u = g(x)$ along the boundary of the region.

Kakutani's theorem on harmonic functions states that such a function's steady-state solution can be approximated with Brownian, random motion[1]. Consider a particle undergoing Brownian motion that is subject to at least one boundary. Given infinite time, the particle will encounter a boundary with probability one. To find the steady-state solution for the particle's starting point, you allow the particle to move randomly until it hits a boundary and record the value of the boundary. For instance, if you are analyzing a heat dissipation scenario, record the temperature of the boundary. Repeat this process sufficiently many times and average the results to approach the point's true value in the function's steady-state solution.

$$f(x) = \begin{cases} -\Delta u(x) = 0, & x \in D \\ \lim_{y \to x} u(y) = g(x), & x \in \partial D \end{cases}$$

Under nice conditions, that is the existence of a harmonic solution, $u(x)$ is $E[g(X)]$ where $X(x)$ is the first exit point (which is random) for Brownian motion started at x.

1.2 Random Walks on Spheres

In order to get to the exit point $X(x_0)$, we used a method called Random Walks on Spheres (RWOS).

1. Start at a point $\mathbf{x_0}$ in a region R
2. Define r as the shortest distance to the boundary
3. Create a sphere around $\mathbf{x_0}$ using radius r
4. Randomly pick a point $\mathbf{x_1}$ on the sphere and move there
5. If $\mathbf{x_1}$ is sufficiently close to the boundary, stop and record the temperature at appropriate X
6. Otherwise repeat the process with $\mathbf{x_1}$ as the new center

We are motivated by two ideas. The exit point on the sphere ($\mathbf{x_1}$) for a Browinian motion started at the center of the sphere ($\mathbf{x_0}$) is distributed uniformly, so its simulation does not require simulation of the entire path of the particle inside the sphere from $\mathbf{x_0}$ to $\mathbf{x_1}$. The radius in each step is optimally chosen for faster convergence to the boundary of the region.

1.3 Relevance

Random walk simulations are applicable in a variety of fields. In physics and chemistry, random simulations model heat diffusion and the diffusion of solutes in fluids. Such a model may approximate a reaction rate if a process is time dependent or show when a disturbed fluid will return to equilibrium. Likewise, studying the randomness inherent in quantum physics requires the use of these simulations.

Larger scale processes can also be investigated with random walk simulations. Consider animal migration. As using apparatuses for monitoring migration can be difficult and expensive, creating an effective model can greatly increase efficiency. With such a model, researchers could rely more heavily on interpolation between monitors while requiring fewer of them. Also, if population data is coupled with a simulation, then the simulation can potentially model the population in any area.

Thus random walks simulations can assist scientific research in a myriad of ways. Our solutions to the Laplace equation relate directly to physics, chemistry, and biology, rendering processes in each discipline more effective and efficient.

2 Random Walks on Spheres Simulations

2.1 One Dimension

Assume both ends of a one-dimensional rod are kept at different, constant temperatures. After sufficient time passes such that the heat in the rod is no longer in flux, each point is at a constant temperature. We want to determine the temperature at any place along the rod. This problem can be solved analytically, setting the heat equation equal to zero and integrating twice,

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= 0 \\ \frac{\partial u}{\partial x} &= k_1 \\ u(x) &= k_1 x + k_2 \end{aligned} \tag{1}$$

where k_1 and k_2 can be found with initial boundary conditions.

Instead, we used Kakutani's work with random walks on spheres to simulate this solution. Using MATLAB, we ran trials of uniformly distributed random movement. If $rand > .5$, our particle stepped right $(+r)$, and if $rand < .5$, our particle stepped left $(-r)$. When our particle hit the edge of our *sphere*, we stepped again, iterating until we hit the outer boundary. Here we recorded the temperature. Over thousands of trials, we maximized efficiency by strategically choosing the step radius to be the shortest distance to the boundary (thereby minimizing iterations).

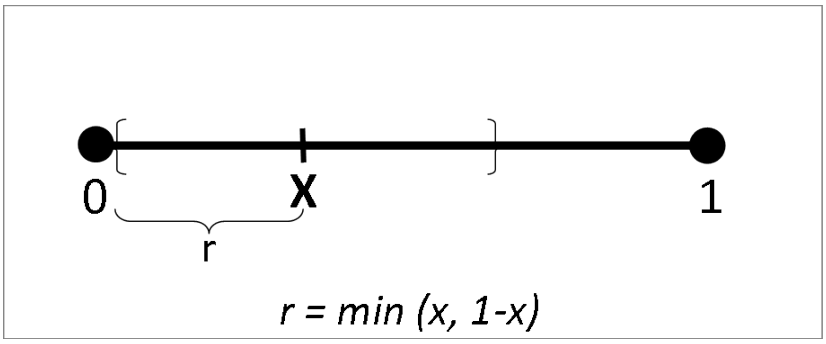

Figure 1: Rod with optimal radius r

We ended with a list of temperatures generated from a set starting point, averaging these to find our estimate for the temperature. The Central Limit Theorem guarantees that with a large number of trials, this average temperature will approach the true temperature at that point. We then ran simulations for starting values of x from 0 to 1 and generated graphs like the one below.

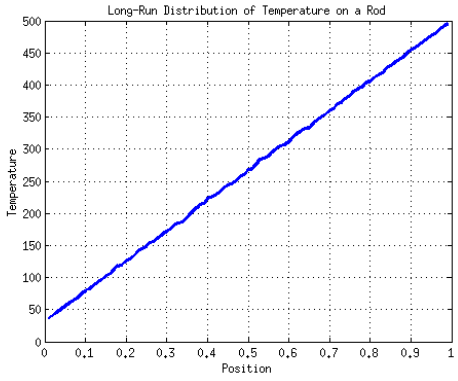

Figure 2: Temperature distribution on a rod at steady-state

From the general solution to the steady-state heat equation (1), we know that the temperature distribution should be linear. Our simulation supports this result as shown by the graph above.

So analytically and by simulation, we come to the same conclusion.

2.2 Two Dimensions

2.2.1 Method

In the two-dimensional case, our RWOS was performed with a circle of optimized radius, r– the closest distance to the boundary. After calculating the radius, a random angle θ is used to move the point to the edge of the circle. As in the one-dimensional case, the simulation ran for many iterations. The algorithm then chose another point inside the shape and repeated the process. The temperature was indicated by either color or height.

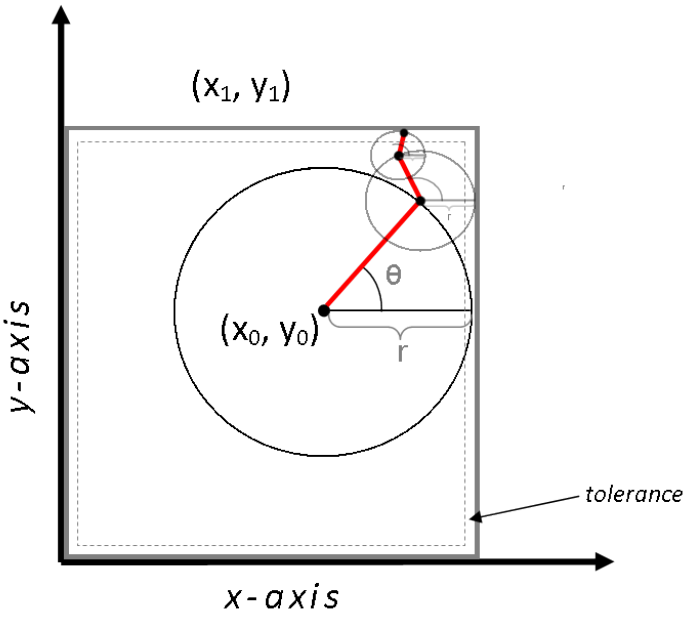

Figure 3: Two-dimensional random walk iterating until hitting bound

2.2.2 Square

The first region we looked at was a square. The following graph shows heat distribution on a square region with side boundaries held at constant temperatures.

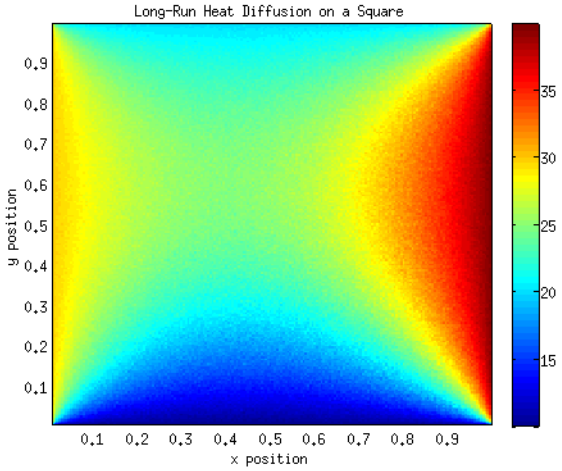

Figure 4: Heat distribution on a steady-state square with constant side bounds

2.2.3 Circle

To experiment with other shapes, we modified our code to polar coordinates and set our region to that of a circle. We set interesting temperature conditions on the circumference of our circle and investigated the resulting distributions. In the following images, both color and height represent the temperature of the circle at the point.

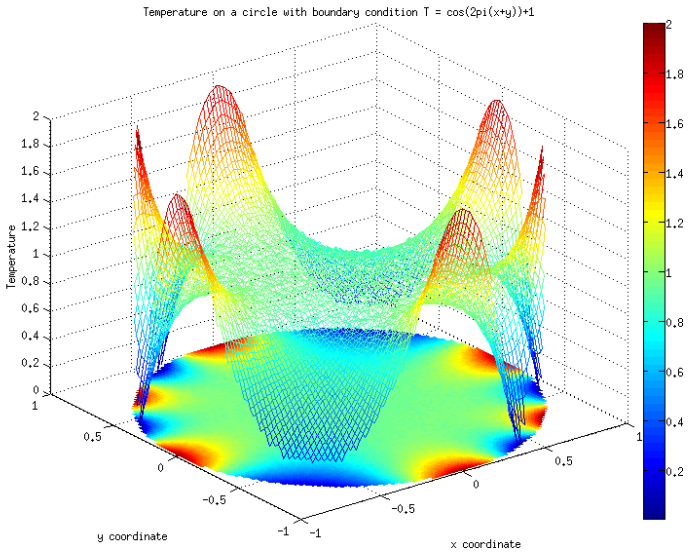

Figure 5: Heat distribution of a steady-state circle with trigonometric bounds

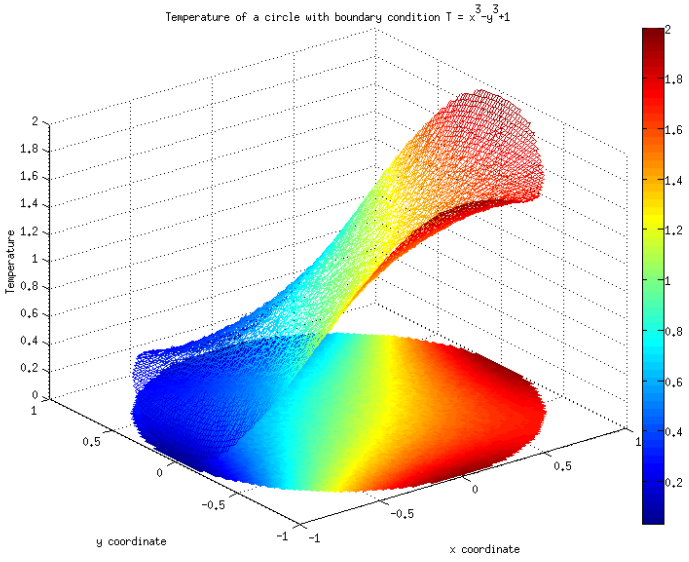

Figure 6: Heat distribution of a steady-state circle with cubic bounds

2.3 Three Dimensions

To adjust our simulation into three-dimensions we rewrote our MATLAB code into spherical coordinates. This allowed us to incorporate two random elements, uniformly distributed theta [0, 2π] and phi [0, π] with our radius being the minimum distance to an object side. Each dot represents a corresponding point inside the object. The color of that dot represents the temperature at that point.

2.3.1 Cube

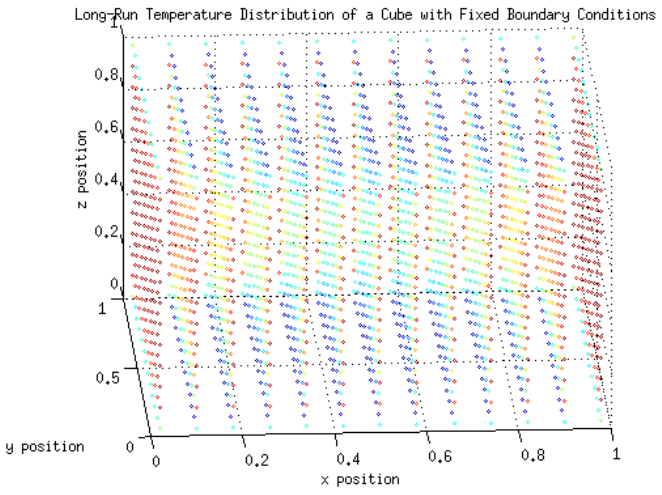

Figure 7: Heat distribution of a steady-state cube

2.3.2 Sphere

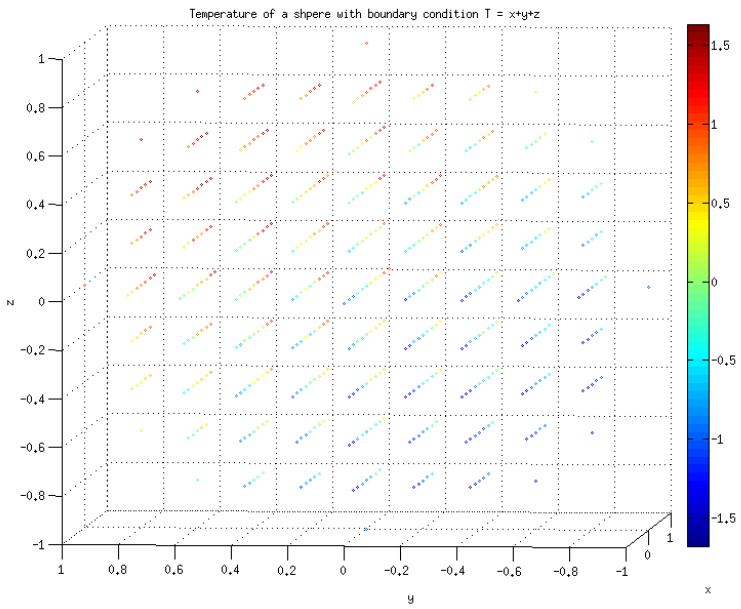

Figure 8: Heat distribution of a steady-state sphere

2.4 Exit Points

Next, instead of recording the temperature, we recorded the position of the particle when it hit the boundary, analyzing the resulting distribution of these *exit points*.

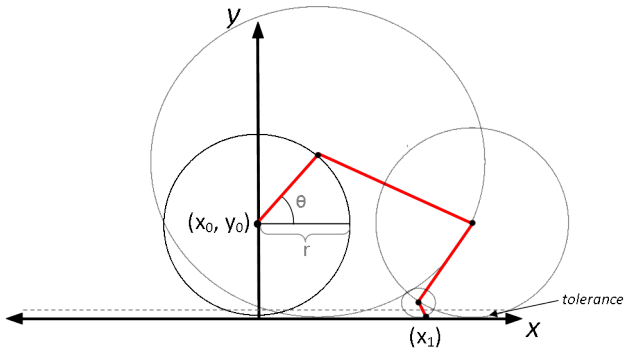

Figure 9: Simulation stepping until particle hits the half-plane boundary (x-axis)

The following figure shows a histogram of the exit points along the half-plane. The Brownian motion simulation particle was reset to $(0, 10)$ for each iteration. The histogram represents the probability distribution function (PDF).

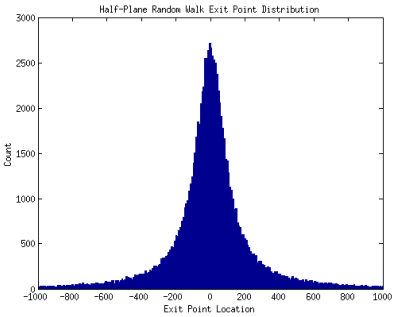

Figure 10: Distribution of exit points on the half-plane

Cumulatively summing the histogram boxes creates the following cumulative distribution function (CDF).

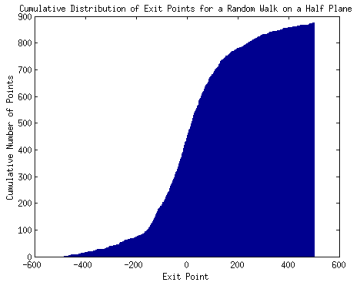

Figure 11: Cumulative distribution of exit points on the half-plane

We also looked at how the distribution changed according to the initial position. The following graph shows how the distribution varies with the changes in height–the closer the point is to the x-axis, the more points will hit directly below it, and the further the point is from the x-axis, the closer to uniform the distribution will be.

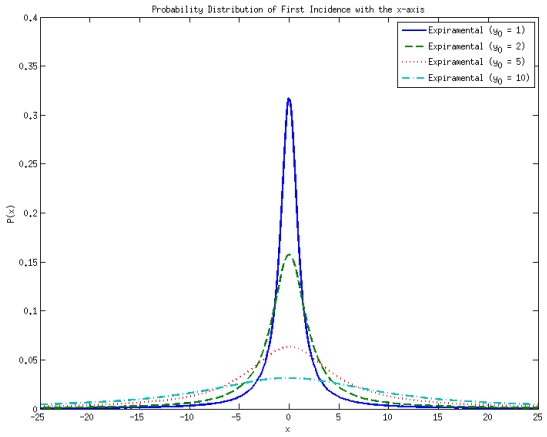

Figure 12: Probability distribution of exit points on the half-plane

3 Analysis of Simulation

3.1 Error Analysis

We investigated the Central Limit Theorem (CLT) by looking at the error of our simulation. According to the CLT, as sample size grows, the amount of error converges to the inverse square root of the sample size, N,

$$Error = \frac{1}{\sqrt{N}}$$

The following plot shows multiple simulations with increasing sample sizes. The larger the sample size, the more the graph fits the true linear temperature distribution.

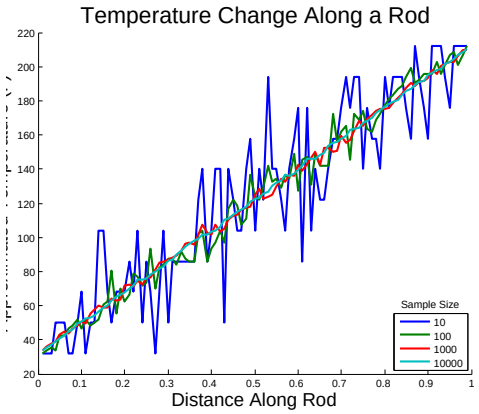

Figure 13: Average temperature along rod with varying sample size

Running the simulation with many different sample sizes allowed us to calculate the average error at a point relative to those sizes. In the graph below, the points approximate the curve $\frac{1}{\sqrt{N}}$, demonstrating the Central Limit Theorem.

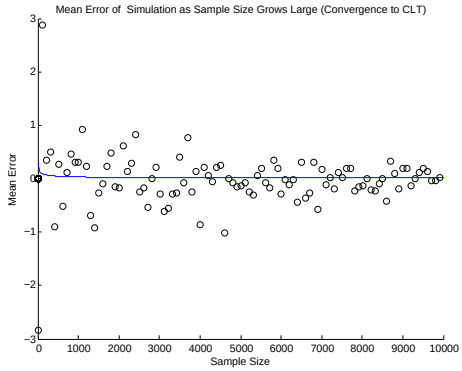

Figure 14: Average error of rod simulations approaches $\frac{1}{\sqrt{N}}$

3.2 Average Step Analysis

We also studied our algorithm's average number of steps before hitting a boundary, comparing the efficacy of an optimized radius to that of a fixed radius. Specifically, the average steps for a set radius depends on the sizes of both the radius and the region. Einstein discovered that the mean distance travelled by Brownian motion is the square root of the time it has traveled.[3]

The diagrams below shows the optimized radius stepping to the boundary on the half-plane compared to a process starting from the same start point stepping with a fixed radius. Both started at $(0, 10)$ and stepped until within a tolerance of .01 of the x-axis. The average number of steps using an optimized r was 12.8069. The average steps using $r = 5$ was 42,656. However, the set radius mean was highly skewed by outliers. The median using the optimized radius was 11 and the median using the set radius was 25, a much smaller gap but still significant.

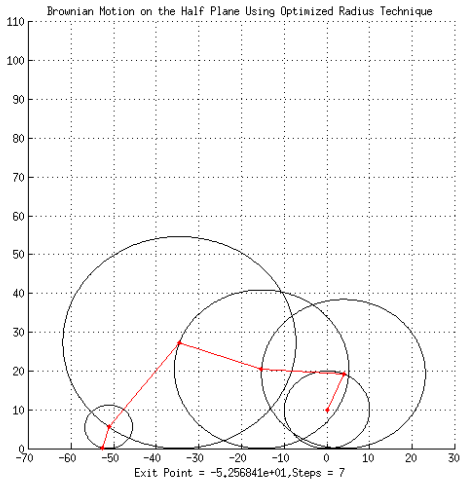

Figure 15: Stepping diagram on the half-plane

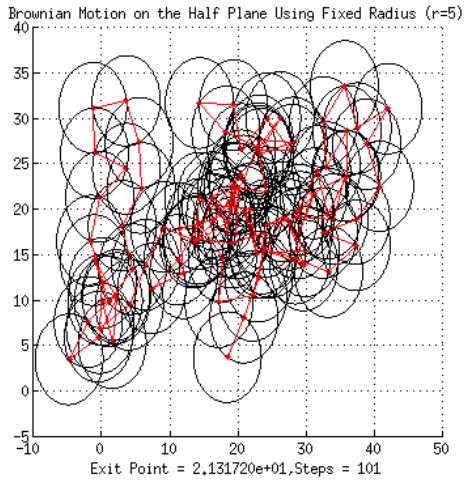

Figure 16: Stepping diagram on the half-plane

If we shift our attention to the upper half-plane, we can get an explicit result for the average speed of convergence of RWOS. Consider (x_0, y_0) an initial point in upper half plane. Let (x_k, y_k) be that point after k RWOS iterations. We want to know how quickly y_k goes to zero. In order to do this, we consider the ratio $y_k/y_{k-1} = 1 + sin(\theta)$ where θ is a uniform random variable on $[0, 2\pi]$. On average $y_k/y_{k-1} = 1$, so why would y_k even go to 0, even though we know it must? Instead of looking at $E\left(y_k/y_{k-1}\right)$, we look at

$$E \ln\left(\frac{y_k}{y_{k-1}}\right) = \frac{1}{2\pi}\int_0^{2\pi} \ln(1+\sin(\theta))\, d\theta$$

Now, since

$$\ln\left(\frac{y_k}{y_0}\right) = \sum_{j=1}^{k} \ln\left(\frac{y_j}{y_{j-1}}\right)$$

and the mean E is a linear operation, we can see that

$$E \ln\left(\frac{y_k}{y_0}\right) = \sum_{j=1}^{k} E \ln\left(\frac{y_j}{y_{j-1}}\right) = k\frac{1}{2\pi}\int_0^{2\pi} \ln(1+\sin(\theta))\, d\theta = -\frac{k}{2}\ln(4)$$

Therefore $E \ln\left(\frac{y_k}{y_0}\right)$ goes to negative infinity (linearly with k), and

$$\exp\left(E \ln\left(\frac{y_k}{y_0}\right)\right) = \exp\left(-\frac{k}{2}\ln(4)\right) = \exp\left(\left(\ln\frac{1}{2}\right)^k\right) = \left(\frac{1}{2}\right)^k$$

So we see that on average, each step of RWOS will move the particle halfway to the x-axis.

4 Laplace's Equation on the Half-Plane

4.1 Motivation

Recall the exit point distribution for the half plane. It was a bell-shaped curve centered around the initial x starting point. One might think it was following a normal distribution, but observe how a similarly placed normal distribution compares.

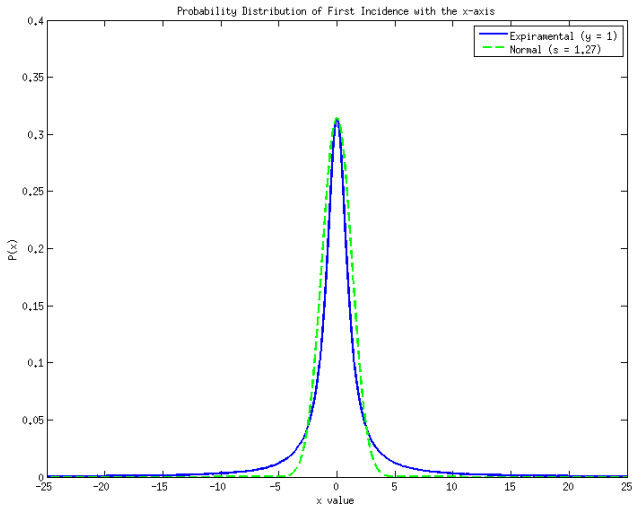

Figure 17: Probability distribution of exit points on the half-plane versus the normal distribution

So what distribution do our exit points follow? We can find this by solving Laplace's equation on the upper half-plane.

4.2 Analytic Solution

Laplace's Equation on the upper half-plane is given by

$$u_{xx} + u_{yy} = 0 \tag{2}$$

$$u(x,0) = u_0(x), \forall\, x \in \mathbb{R} \tag{3}$$

With the condition that the temperature is bounded, i.e.

$$y \geq 0 \text{ and } |u(x,y)| < \infty$$

Now, by taking the Fourier Transform of (2) we get

$$\begin{aligned}\mathcal{F}(u_{xx} + u_{yy}) &= \int_{-\infty}^{\infty} e^{-i\omega x}(u_{xx} + u_{yy})dx \\ &= \int_{-\infty}^{\infty} e^{-i\omega x}u_{xx}dx + \int_{-\infty}^{\infty} e^{-i\omega x}u_{yy}dx\end{aligned}$$

We know from the properties of Fourier Transforms that

$$\mathcal{F}(u_{xx}) = -\omega^2\mathcal{F}(u)$$

$$\mathcal{F}(u_{yy}) = \frac{\partial^2}{\partial y^2}\mathcal{F}(u)$$

Letting $\hat{u}(\omega, y) = \mathcal{F}(u(x,y))$, (2) and (3) are transformed into

$$\frac{\partial^2}{\partial y^2}\hat{u}(\omega,y) = \hat{u}(\omega,y) \tag{4}$$

$$\hat{u}(\omega,0) = \hat{u}_0(\omega) \tag{5}$$

which is a second order ODE.

The general solution to this second order ODE is

$$\hat{u}(\omega,y) = C_1(\omega)e^{\omega y} + C_2(\omega)e^{-\omega y} \quad (\omega > 0)$$

Since $|\hat{u}(\omega,y)| < \infty$, it must be the case that $C_1(\omega) = 0$

$$\implies \hat{u}(\omega,y) = C_2(\omega)e^{-|\omega|y}$$

Furthermore, since $\hat{u}(\omega,0) = \hat{u}_0(\omega)$ and $\hat{u}(\omega,0) = C_2(\omega)$, it is clear that $C_2(\omega) = \hat{u}_0(\omega)$

$$\implies \hat{u}(\omega,y) = \hat{u}_0(\omega)e^{-|\omega|y} \tag{6}$$

In order to solve Laplace's Equation, we must find the inverse transform of (6). First, however, we must find the inverse of $e^{-|\omega|y}$

$$\begin{aligned}\mathcal{F}^{-1}(e^{-|\omega|y}) &:= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}e^{-|\omega|y}\,d\omega \\ &= \frac{1}{2\pi}\left[\int_0^{\infty} e^{ix\omega}e^{-\omega y}\,d\omega + \int_0^{\infty} e^{-ix\omega}e^{-\omega y}\,d\omega\right] \\ &= \frac{1}{2\pi}\left[\int_0^{\infty} e^{-\omega(y-ix)}\,d\omega + \int_0^{\infty} e^{-\omega(y-ix)}\,d\omega\right] \\ &= \frac{1}{2\pi}\left[\frac{1}{y-ix} + \frac{1}{y+ix}\right] = \frac{1}{2\pi}\left[\frac{y+ix+y-ix}{x^2+y^2}\right] \\ &= \frac{1}{\pi}\frac{y}{x^2+y^2}\end{aligned} \tag{7}$$

Another property of Fourier Transforms is that

$$\mathcal{F}^{-1}(\hat{f}(\omega,y)\hat{g}(\omega,y)) = \int_{-\infty}^{\infty} f(t,y)g(x-t,y)\,dt$$

So the inverse transform of (6), and the solution to Laplace's Equation on the upper half-plane, is

$$\begin{aligned} u(x,y) &= \mathcal{F}^{-1}(\hat{u}(\omega,y)) \\ &= \mathcal{F}^{-1}(\hat{u}_0(\omega)e^{-|\omega|y}) \\ &= \int_{-\infty}^{\infty} u_0(t)\frac{1}{\pi}\frac{y}{(x-t)^2+y^2}\,dt \end{aligned}$$

So we have solved Laplace's equation on the half-plane and note that $\frac{1}{\pi}\frac{y}{(x-t)^2+y^2}$ is of the form of a Cauchy distribution, but we still need to verify that this is related to the exit points. Recall, that by Kakutani

$$u(x,y) \;=\; E[u_0(X)] \;=\; \int_{-\infty}^{\infty} u_0(x)f(x)\,dx$$

where $f(x)$ is the probability density function of our random variable X. So, from our definition of expected value and from solving Laplace's equation on the half-plane, we know

$$\begin{aligned} u(x,y) \;&=\; \int_{-\infty}^{\infty} u_0(x)f(x)\,dx \\ u(x,y) \;&=\; \int_{-\infty}^{\infty} u_0(x)\frac{1}{\pi}\frac{y}{(x-t)^2+y^2}\,dt \\ \Rightarrow f(x) \;&=\; \frac{1}{\pi}\frac{y}{(x-t)^2+y^2} \end{aligned}$$

This verifies that the exit points follow a Cauchy distribution on the half-plane.

5 Conformal Mapping and Other Regions

5.1 Conformal Mapping

After we analytically solved Laplace's equation on the half-plane we shifted our attention to other regions. However, instead of starting from scratch, we applied conformal mapping to our Cauchy result. This saved us time and gave us insight into how the space was changed. Conformal mapping is a process that transforms one space or shape into another. Importantly, a conformal map preserves analyticity such that Laplace's equation still holds.

5.2 Quarter-Plane

5.2.1 Analytic Solution

The first region we were interested in looking at was the quarter-plane in the positive quadrant. We used the function $f(z) = z^2$ to map the quarter-plane to the half-plane and allow us find the exit point distribution. Note that we will expect to have two exit point distributions by the end of this since we will be able to exit on the x-axis or y-axis

First, let $h(x,y)$ be the temperature distribution on the region and let $h_0(x)$ and $h_1(y)$ be the boundary conditions of the region. Start by transforming our $u(x,y)$ from the half-plane to obtain these functions.

$$\begin{aligned} h(x,y) &= h(z) = u(f(z)) = u(x^2-y^2, 2xy) \\ h_0(x) &= h(x,0) = u(x^2,0) = u_0(x^2) \\ h_1(y) &= h(0,y) = u(-y^2,0) = u_0(-y^2) \\ \implies u_0(\xi) &= \begin{cases} h_0(\sqrt{\xi})\,, \xi > 0 \\ h_1(\sqrt{-\xi})\,, \xi < 0 \end{cases} \end{aligned}$$

So, recall we know that on the half-plane, the exit points were solved with the following

$$u(x,y) \;=\; \int_{-\infty}^{\infty} u_0(x) \frac{1}{\pi} \frac{y}{(x-t)^2+y^2}\, dt$$

Start by substituting in the new boundary conditions $h_0(x)$ and $h_1(x)$ for $u_0(x)$

$$= \frac{1}{\pi}\int_{-\infty}^{0} \frac{y}{(x-t)^2+y^2} h_1(\sqrt{-t})\, dt + \frac{1}{\pi}\int_{0}^{\infty} \frac{y}{(x-t)^2+y^2} h_0(\sqrt{t})\, dt$$

Now, for substitution, let $t = \tau^2$ for the first integral and let $t = -\tau^2$ for the second integral

$$= \frac{1}{\pi}\int_{0}^{\infty} \frac{2\tau y}{(x+\tau^2)^2+y^2} h_1(\tau)\, d\tau + \frac{1}{\pi}\int_{0}^{\infty} \frac{2\tau y}{(x-\tau^2)^2+y^2} h_0(\tau)\, d\tau$$

Finally, recall from above that $h(x,y) = u(x^2-y^2, 2xy)$. So, we are left with

$$h(x,y) \;=\; \frac{1}{\pi}\int_{0}^{\infty} \underbrace{\frac{2\tau y}{(x+\tau^2)^2+y^2}}_{\text{y-axis PDF}} h_1(\tau)\, d\tau + \frac{1}{\pi}\int_{0}^{\infty} \underbrace{\frac{2\tau y}{(x-\tau^2)^2+y^2}}_{\text{x-axis PDF}} h_0(\tau)\, d\tau$$

Recall how much effort it took to solve Laplace's equation on the half-plane. Rather than solving it again on the quarter-plane and having to go through the entire process, we were able to use conformal maps to make this a much easier problem and reduce the effort in finding the exit point distributions.

5.2.2 Theoretical vs. Simulation

We want to confirm that this theoretical distribution actually follows the exit point distribution of the simulation. Below are two examples comparing the exit point simulation distribution to the theoretical distribution.

These two distributions match up very well and would converge more with a higher sample size during the simulation. Note, though, the disparity at $x = 0$. This is due to the fact that we used a Parzen Window estimation to graph our simulation data, giving us a C^∞ function, which we are comparing to a function with no derivative at zero.

5.3 Circle

5.3.1 Analytic Solution

We also looked at solving Laplace's Equation on the unit disk $D = \{z \in \mathbb{C} : |z| \leq 1\}$. One way of doing this is to find an analytic mapping $f : D \to \mathbb{H}$, where $\mathbb{H} = \{z \in \mathbb{C} : Im(z) \geq 0\}$ is the upper half-plane, and use it to map the points and boundary condition of the disk onto the half-plane.

The Möbius Transformation $f(x,y) = \left(\frac{2x}{1-2y+y^2+x^2}, \frac{1-x^2-y^2}{1-2y+y^2+x^2}\right)$ maps the unit disk onto the upper half-plane and since f is a Möbius Transformation we know that is analytic. Therefore $h = u \circ f$ is still harmonic and solves Laplace's equation on the unit circle.

Using f, we can solve $\Delta h = 0$ with initial condition $h_1(\theta)$ by remembering that $\Delta h = 0 \Leftrightarrow \Delta u \circ f = 0$ (where u satisfies $\Delta u = 0$ on the half-plane with initial condition $u(x,0) = u_0(x)$).

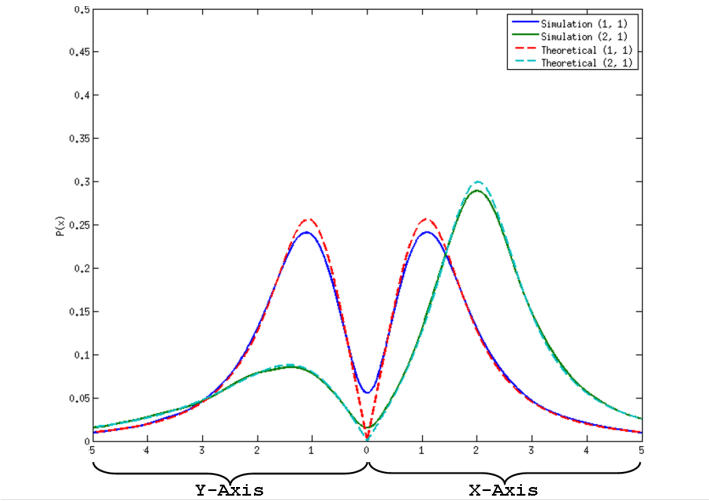

Figure 18: Probability distribution function of exit points from Brownian motion on the quarter-plane

Since f maps the boundary of D onto the boundary of $\mathbb{H}$,

$$u(t,0) = h(f^{-1}(t,0)) = h(\tfrac{2t}{t^2+1}, \tfrac{t^2-1}{t^2+1}) = h_1(tan^{-1}(\tfrac{t^2-1}{2t}))$$

Now, we know for the half-plane that

$$\begin{aligned} u(x,y) &= \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{y}{(x-t)^2+y^2} u_0(t)\,dt \\ &= \frac{1}{\pi}\int_{-\infty}^{0} \frac{y}{(x-t)^2+y^2} h_1(tan^{-1}(\tfrac{t^2-1}{2t}))\,dt \\ &\quad + \frac{1}{\pi}\int_{0}^{\infty} \frac{y}{(x-t)^2+y^2} h_1(tan^{-1}(\tfrac{t^2-1}{2t}))\,dt \end{aligned}$$

Letting $\theta = tan^{-1}(\frac{t^2-1}{2t})$, we see that $t = tan(\theta) + sec(\theta)$ and $dt = (sec^2(\theta) + tan(\theta)sec(\theta))\,d\theta$. By substituting this into what we had above, we get

$$\begin{aligned} u(x,y) &= \frac{1}{\pi}\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \frac{y(tan(\theta)sec(\theta)+sec^2(\theta))}{(x-(sec(\theta)+tan(\theta)))^2+y^2} h_1(\theta)\,d\theta \\ &\quad + \frac{1}{\pi}\int_{\frac{3\pi}{2}}^{\frac{5\pi}{2}} \frac{y(tan(\theta)sec(\theta)+sec^2(\theta))}{(x-(sec(\theta)+tan(\theta)))^2+y^2} h_1(\theta)\,d\theta \\ &= \frac{1}{\pi}\int_{\frac{\pi}{2}}^{\frac{5\pi}{2}} \frac{y(tan(\theta)sec(\theta)+sec^2(\theta))}{(x-sec(\theta)-tan(\theta))^2+y^2} h_1(\theta)\,d\theta \end{aligned}$$

And so finally, we get that

$$\begin{aligned} h(x,y) &= u\left(\tfrac{2x}{1-2y+y^2+x^2}, \tfrac{1-x^2-y^2}{1-2y+y^2+x^2}\right) \\ &= \frac{1}{\pi}\int_{\frac{\pi}{2}}^{\frac{5\pi}{2}} \frac{\left(\frac{1-x^2-y^2}{1-2y+y^2+x^2}\right)(tan(\theta)sec(\theta)+sec^2(\theta))}{\left(\frac{2x}{1-2y+y^2+x^2} - sec(\theta) - tan(\theta)\right)^2 + \left(\frac{1-x^2-y^2}{1-2y+y^2+x^2}\right)^2} h_1(\theta)\,d\theta \end{aligned}$$

So the PDF (in radians) of exit points of Brownian Motion starting from inside a circle at the point (x,y) is

$$\frac{\left(\frac{1-x^2-y^2}{1-2y+y^2+x^2}\right)(tan(\theta)sec(\theta)+sec^2(\theta))}{\left(\frac{2x}{1-2y+y^2+x^2}-sec(\theta)-tan(\theta)\right)^2+\left(\frac{1-x^2-y^2}{1-2y+y^2+x^2}\right)^2}$$

5.3.2 Theoretical vs. Simulation

Again, we graphed our analytically solved theoretical solution against our simulation to compare accuracy. The figure below shows a PDF of the exit points from a circle. Note that the graph shown has a period of 2π and thus loops to the start every 2π. The two starting points shown in the graph are (1/2, 0) and (-1/2, 0). These are equally far from the circle's center and thus are equally tall. They appropriately have exit points weighted heavier at 2π and π respectively.

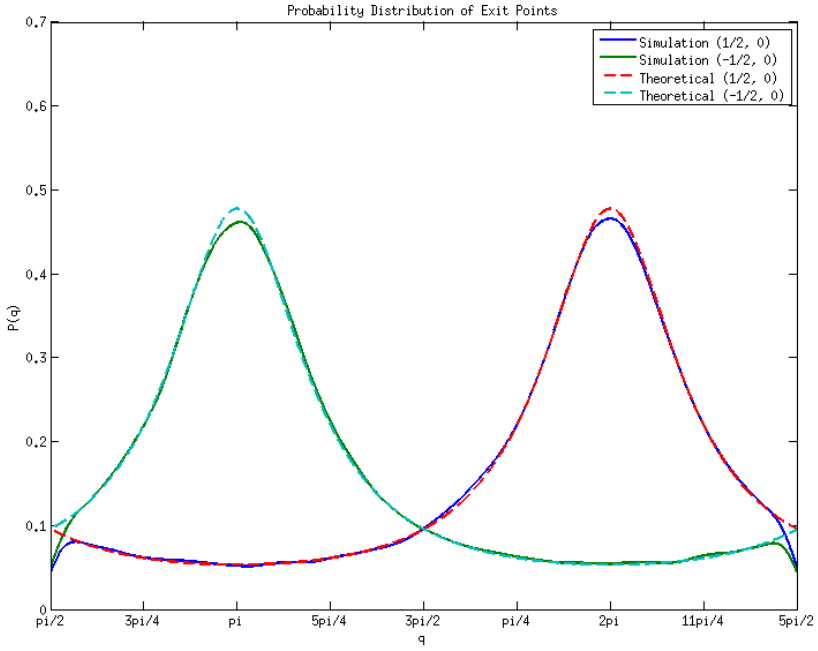

Figure 19: Probability distribution function of exit points from Brownian motion on the quarter-plane

The disparity at the tails of the graph (at $\pi/2$ and $\pi/2$) appears because we are comparing a continuous function (theoretical) to a Parzen Window approximation of a discrete function (simulation). However, the simulation and theoretical graphs are sufficiently similar to assume equivalence.

6 Gaussian Quadrature

6.1 Theory

So we have seen our process works accurately and efficiently, but it has to calculate the temperature at the boundary many times each time it runs. What if the boundary values were expensive to compute? For example, suppose an ecologist wants to study the population of rabbits in an enclosed national forest. The time, equipment, and man power needed to tag the rabbits, trap them along the perimeter of the forest, and count them is expensive. To reduce this cost, the ecologist sets only ten traps around the edge of the area. However, he wants to know the optimal placement of the traps to best capture the population trends.

We want to approximate the temperature at any given point, which is given by the integral

$$u(x_0, y_0) = \int_{\partial R} f(x)u_0(x)\,dx$$

One way we can approximate this integral is through Gaussian quadrature. Gaussian quadrature is particularly useful for the type of problem we are looking at because it uses a small, finite number of points to get a close approximation to the true value of the integral.

$$\int_{\partial R} f(x)u_0(x)\,dx \approx \sum_{i=1}^{n} \underbrace{u_0(F^{-1}(x_i))}_{\text{value}} \underbrace{w_i}_{\text{weight}}$$

Here, $f(x)$ is the probability density function of the exit points, $F^{-1}(x)$ is the inverse cumulative distribution function of that density function. The idea is that we can find points $x_1, x_2, ..., x_n$ along the border of the region using our exit point distribution, and then assign a weight to each point to create our approximation.

6.2 Gaussian Quadrature

The Legendre polynomials are a widely studied set which is often associated with Gaussian quadrature. They are a set of orthogonal polynomials that can be explicitly expressed as

$$P_n(x) = (-1)^n \sum_{k=0}^{n} \binom{n}{k}\binom{n+k}{k}(-x)^k$$

To perform our approximation, we found the roots of the n^{th} degree Legendre Polynomial. It is well known that these roots are often a good choice for using with Gaussian quadrature. However, we transformed the roots according to our exit point distribution so that they were properly spaced along our region.

After having our points spaced properly along the boundary of our region, we needed to find the appropriate weights. There were two ways we found these weights. The first way used a system of equations to solve for the weights of each point.

$$\int_0^1 x^k\,dx \approx \sum_{i=1}^{n} (x_i)^k w_i \tag{8}$$

We plugged $k = 0, 1, ..., n-1$ into our integral approximation formula (8) to form n equations.

$$\begin{pmatrix} x_1^0 & x_2^0 & ... & x_n^0 \\ x_1^1 & x_2^1 & ... & x_n^1 \\ ... & ... & ... & ... \\ x_1^n & x_2^n & ... & x_n^n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ ... \\ w_n \end{pmatrix} = \begin{pmatrix} \int_0^1 x^0\,dx \\ \int_0^1 x^1\,dx \\ ... \\ \int_0^1 x^n\,dx \end{pmatrix}$$

This method works well, though for some large n, this system can become unstable when evaluating it in some software packages. For this reason, we also worked with an explicit formula for finding the weights. In the case of the Legendre polynomials, the weights are given by the following formula

$$w_i = \frac{2}{n} \frac{1}{p_n'(x_i)p_{n-1}(x_i)}$$

where p_n is the n^{th} degree Legendre polynomial.

We found this method to be more efficient when using large n because it bypassed the possibility of unstable solutions resulting from inverting matrices. However, in most of the of the cases that we looked at, there was no need to have a high degree n that would cause the matrix calculation to be unstable.

6.3 Error Analysis

After implementing this procedure in MATLAB, we wanted to look at how well our approximation actually worked. When we first started looking at this, we realized that this heavily depends on what region we are on and what our initial boundary condition is. For example, if we are on the half-plane, we are approximating the following integral

$$u(x_0, y_0) = \int_{-\infty}^{\infty} \frac{y_0}{(x_0 - t)^2 + y_0^2} u_0(t)\, dt$$

where (x_0, y_0) is our starting points and $u_0(x)$ is our boundary condition. The first boundary condition we wanted to try was $u_0(x) = x^2$ because it seemed straightforward to implement. But, note that if the boundary is x^2 then the integral will not converge in the first place and we cannot create an accurate approximation to it. So, we needed to restrict the class of functions we could use for our boundary, depending on what region we were on.

Staying on the half-plane, we looked at boundary conditions that we believed would allow us to create an accurate approximation. We tried four different functions to see how different functions would lead to better or worse convergence for the process. The following table shows the approximation created for the boundary condition using an n^{th} degree Legendre polynomial at the point $(0, 1)$. "True" stands for the rounded true value of the integral with the given boundary function.

Table 1: Half-Plane Approximation

n	$cos(x)$	$\frac{1}{1+x^2}$	$\frac{sin(\pi x)}{\pi x}$	e^{-x^2}
True	0.3678	0.5000	0.3046	0.4276
1	1.000	1.000	1.000	1.000
2	0.2885	0.5112	-0.1910	0.1952
3	-0.0591	0.4995	0.4697	0.4448
4	0.4720	0.5000	0.3612	0.4597
5	0.7022	0.5000	0.2396	0.4183
6	0.2151	0.5000	0.2972	0.4210
7	0.4280	0.5000	0.3644	0.4308
8	0.1763	0.5000	0.2869	0.4292
9	0.3558	0.5000	0.2744	0.4266
10	0.4590	0.5000	0.3171	0.4271
20	0.3880	0.5000	0.3047	0.4276
40	0.3246	0.5000	0.3058	0.4276
70	0.3629	0.5000	0.3056	0.4276

So, notice that even up to the 20^{th} degree polynomial the quadrature is having hard time accurately approximating the true value of the $cos(x)$ integral, but by higher degrees it begins to converge, though somewhat slowly. Part of this is mostly likely from the fact that as cosine goes out towards infinity, it

continues to move periodically so the approximation struggles with handling points very far from the origin.

The rational function $\frac{1}{1+x^2}$ converges much more quickly, and very little computational effort is needed to work towards the true value. We believe this is because it has vanishing derivatives, that the function's derivative goes to 0 very quickly as the function goes out to infinity, so the approximation has less to worry about far away from the origin.

Finally, just notice that $\frac{sin(\pi x)}{\pi x}$ performed moderately, having a convergence somewhere between the cosine function and rational function. The exponential function converged very quickly, which further suggests that how quickly the function's derivative vanishes affects how well the approximation performs.

The following graph shows the percent error for the four boundary conditions over the n^th degree Legendre polynomials from 1 to 70.

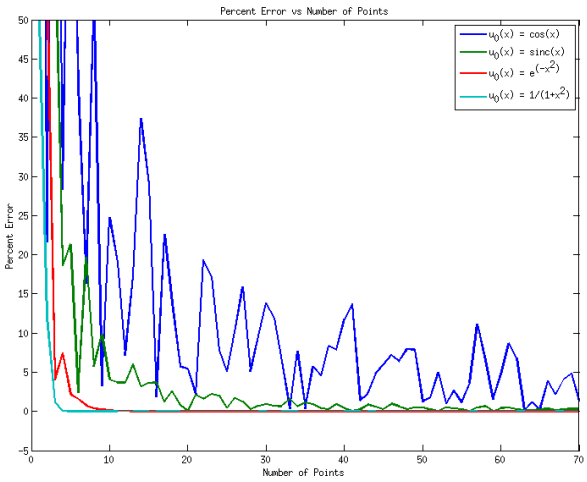

Figure 20: Analysis of error in Gaussian quadrature vs. number of points in approximation

7 Conclusion and Further Work

Looking back, we first utilized Kakutani's Theorem by using random walk simulations to determine the temperature within multiple regions. Then we shifted our attention to exit points. We derived the Cauchy distribution as the probability density function of the exit points on the half-plane, and we supported this through additional random walks. Next we studied the use of conformal mapping to adapt our solution to additional regions, notably the unit disk and the upper half-plane. Lastly we investigated approximating the values inside the region when only incomplete boundary data is given using Gaussian quadrature.

Further work would involve looking deeper into the conditions for accuracy and efficiency in Gaussian quadrature. There are clear trends in the boundary functions that work particularly well. We have noticed those with vanishing derivatives seem to decrease error in our approximation. However, we would like to test a greater variety of boundary functions and possibly prove which types are superior. Lastly, we want to investigate applying Gaussian quadrature to regions with unknown probability density functions. Our entire project up to this point was built upon the assumption that the probability density function of a region was known and managable. We would like to investigate methods of finding these functions for difficult regions, regions that are more realistic in world applications.

8 Acknowledgements

We would like to thank our advisor, Dr. Igor Nazarov, for his insight and guidance throughout this project. Thank you to Justin Droba, Richard Shadrach, and Joseph Roth for their help with MATLAB, Complex Analysis, and PDEs. We also extend our gratitude to Michigan State University, Lyman Briggs College and, more specifically, Dr. Aklilu Zeleke for organizing and hosting the SURIEM program this summer. We would also like to thank our home universities: Michigan State University, University of Connecticut, Belmont University, Yale University, Samford University, and Bryn Mawr College. Lastly, a great thanks to the National Security Agency and National Science Foundation for funding the Mathematics REU.

Project sponsored by the National Science Foundation under Grant Number DMS 1062817.
Project sponsored by the National Security Agency under Grant Number H98230-13-1-0259.

References

[1] Shizuo Kakutani *Two-dimensional Brownian Motion and Harmonic Functions*. Mathematical Institute, Osaka Imperial University, Dec. 12, 1944.

[2] Abramowitz, Milton; Stegun, Irene A., eds. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications, 1972.

[3] Albert Einstein *Investigation on the Theory of the Brownian Movement*. New York: Dover 1956.

[4] Fima C. Klebaner, *Introducation to Stochastic Calculus with Applications*. Imperial College Press, London, 2nd Edition, 2005.