Studying Harmonic Measure through Brownian Motion Simulation and Teleportation

Miranda Schaum,
Undergraduate
Oberlin College
309 Edgemeer Place
Oberlin, OH 44074
1-440-574-1575
mschaum@oberlin.edu

Kevin Gerstle, Visiting Assistant Professor of Mathematics
Oberlin College
10. N. Professor St, King 205
Oberlin, OH 44074
1-502-553-0559
kgerstle@oberlin.edu

ABSTRACT

The harmonic measure distribution function, or h-function, of a two-dimensional complex domain gives valuable insight into the geometric properties of the domain such as its curvature and connectedness. In this paper, we will discuss how these functions may be computed through repeated simulation of Brownian motion, and we will discuss how the process of teleportation may be used to greatly speed up these simulations by means of taking large random steps inside of our domains.

1. INTRODUCTION

1.1 Background Information

Given a two-dimensional complex domain Ω and a fixed base point z_0 inside Ω, the harmonic measure distribution function, or h-function, encodes important information about the geometric properties of the domain relative to the point z_0.

While the h-function may be defined analytically to encode information about the harmonic measure of the boundary of the domain, by a result of Kakutani [2], we may equivalently define the h-function h(r) for a fixed base point z_0 in Ω to be the probability a particle undergoing random, two-dimensional Brownian motion starting at the point z_0 will first exit the domain within distance r from z_0. This produces a right-continuous function of r starting at 0 and increasing towards 1. The h-function of any domain will describe properties of Ω and z_0 such as the minimum distance from z_0 to the boundary of Ω and the symmetry of the domain Ω around z_0. An example of a domain and its h-function can be seen in Figures 1 and 2, respectively.

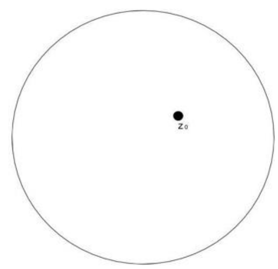

Figure 1. An example of an off-center circle domain $x^2 + y^2 < 1$ where z_0 is (0.5, 0.5)

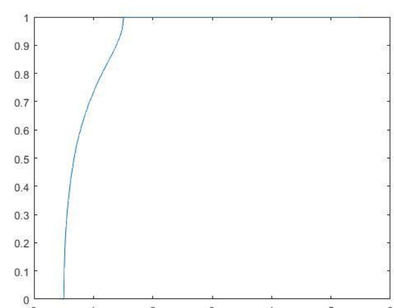

Figure 2. The h-function that corresponds with the off-center circle domain.

Many domains have h-functions that can be explicitly computed by analytical means, including several different types of simply connected domains. However, for h-functions that cannot be computed as easily, simulation of Brownian motion may be used to numerically approximate these functions. By repeatedly simulating Brownian paths starting at z_0 and ending when the particle first intersects the boundary of **Ω**, past studies [1,3] have successfully approximated h-functions for a variety of domains using Matlab. However, these past simulations were computationally inefficient in their simulation of Brownian motion by taking extremely small random steps until the simulated particle hit the boundary.

In our current work, we have implemented a new process to greatly increase the efficiency of simulated Brownian paths. In this process, rather than taking extremely small steps to more accurately represent Brownian motion in **Ω**, we instead took a large disc D centered around z_0 and contained in **Ω**, then took one step in a random direction chosen from a uniform distribution of size d equal to the particle's distance from the boundary of D, making use of the fact that a Brownian particle first exiting **Ω** must first leave the disc D. Since in doing so the particle is equally likely to exit D anywhere on its boundary, such a large step will still preserve the randomness of Brownian motion. This process is then iterated, choosing a new disc D each time still contained inside **Ω**, until the particle gets within some specified tolerance distance ε from the boundary of **Ω**, at which point the particle is considered to have crossed the boundary of **Ω** for the purpose of simulating the h-function as seen in Figure 3. We have called this process of taking large random steps *teleportation*.

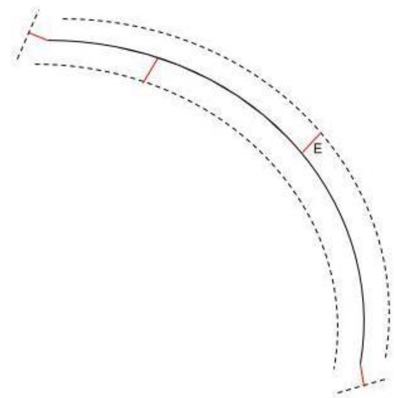

Figure 3. Distance ε from the boundary of Ω.

We have implemented this process of teleportation in the study of a class of domains called circle domains that are of particular importance to the study of h-functions. Circle domains are domains centered at the base point z_0 with arcs of smaller circles excised from the domain; hence the boundary of such a domain consists of the outer circle as well as these inner arcs of varying radii all centered around z_0. For example, see Figure 4 and its corresponding h-function in Figure 5.

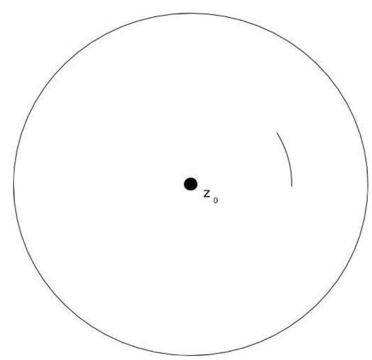

Figure 4. An example of a circle domain.

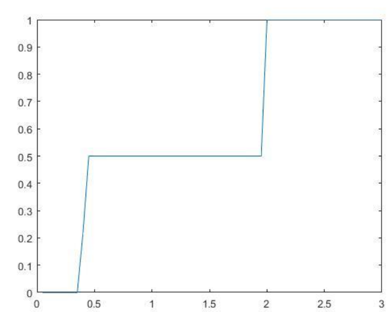

Figure 5. The corresponding h-function for the circle domain in Figure 3.

Circle domains are of particular importance in our study of h-functions in that their h-functions, which are always step functions increasing from 0 to 1, can be used to approximate the h-functions of other domains. In particular, given any function f satisfying necessary properties to be the h-function of some unknown domain, f may be approximated by a sequence of step functions increasing from 0 to 1. By constructing the corresponding circle domains for which these approximated step functions are the h-functions, we can ultimately find examples of domains Ω whose h-functions will be the function f. This process of constructing domains given an h-function is called the *Step Function Program* [6].

We hypothesized that implementing the teleportation process into simulating Brownian motion in these circle domains would vastly improve the computational efficiency of previous programs. In this paper, we will discuss the details of our implementation and the results we have discovered so far. In doing so, we will describe how the computational efficiency of these simulations was improved and compare the simulated data between the new and old programs as well as the corresponding h-functions.

2. PROCEDURE

2.1 Scenarios

To start, we describe all possible scenarios of finding the distance from a particle to any one arc in the boundary of a circle domain as seen in Figure 6.

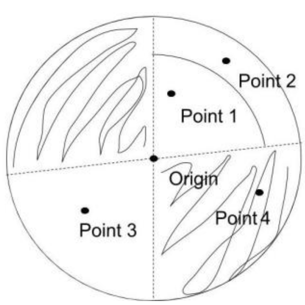

Figure 6. All of the possible scenarios of where a particle could be in a one-arced circle domain.

To determine the distance from the simulated particle to the boundary at any time t, we first found that the distance between the particle and a single boundary arc in each scenario was the minimum of the distance from the particle to either of the two endpoints of the arc and the distance along the line passing through the particle's current location and the origin if this line intersected the arc. We then took the current distance d from the particle to the boundary as a whole to be the minimum of the distances from the particle to all of the boundary arcs and its distance to the outside circle.

To determine to which point the particle would teleport, the program chose a random angle to indicate the direction of teleportation and then teleported the particle in a step size of the distance d to the boundary in that direction.

2.2 Problems and Solutions

Since when taking a step of size equal to the distance from the boundary has probability of actually crossing the boundary of 0, we knew that through teleportation alone, no particle would ever cross the boundary of the domain. To combat this, we assigned a variable, ε, to indicate the tolerance for error in hitting the boundary, meaning that if after teleporting the particle came within distance ε of the boundary, then we considered the particle to have crossed the boundary and ended the current path as seen in Figure 7.

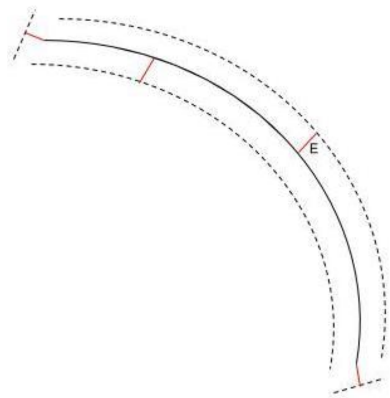

Figure 7. Crossing the boundary using teleportation corresponds to coming within distance ε of the boundary.

3. DATA

3.1 Determining if the Program Runs Successfully

On our first run of the program, we wanted to simply make sure that it created an h-function that correctly represented the information that it was given. To do this, we tested it starting with only one boundary arc of radius 0.4 with a subtended angle of size π/4 radians and an outer circle of radius 2. We told it to run 10,000 paths, and if the number of times the particle teleported within a single path without coming within distance ε of the boundary exceeded 100, that path was regarded as a failure and was excluded from our observations. A visual representation of this circle domain can be seen in Figure 8.

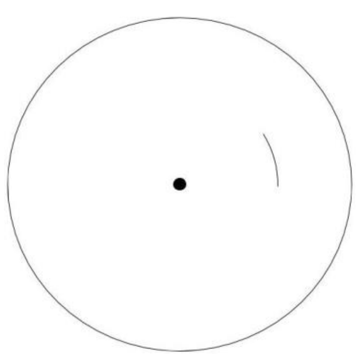

Figure 8. A representation of the circle domain.

The h-function our program created can be seen in Figure 9.

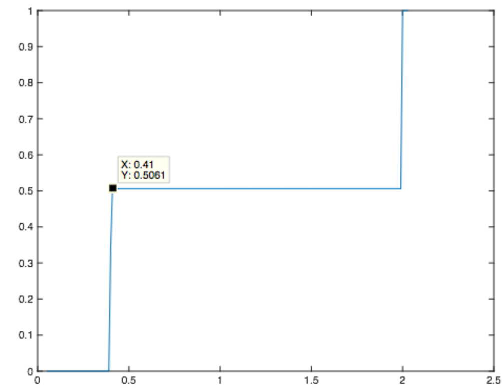

Figure 9. The h-function the program created where the x-axis signifies the distance away from the starting point of the Brownian particle (the center point of the circle) and the y-axis signifies the probability that the particle will cross the boundary at that distance or less from the starting point.

From our knowledge of h-functions of circle domains, we knew that the h-function corresponding to any circle domain must be a discontinuous step function, so the graph our program produced was not surprising. There are two boundary components in this domain, the outer circle and the inner arc, so there should be two steps at radii of 0.4 and 2 corresponding to the distances of these boundary components from the particle's origin. The graph is never decreasing, and the probability h(r) becomes 1 when the radius r becomes 2 because that is the radius of the outer circle. All of this supports that the program is successfully creating the h-function that corresponds to this circle domain.

One thing to note is that since our simulated particles do not actually cross the boundary but rather stop when they come within distance ε of the boundary, the program doesn't automatically jump from one probability to the next exactly at the radii, like an actual step function. The program did not accurately measure probabilities extremely close to values of r corresponding to boundary components due to how we measured crossing the boundary by coming within distance ε of the boundary. Also, while step functions by definition are not continuous at the points r=0.4 and r=2, Matlab automatically connects all the points in our simulated graphs, so while they appear to be continuous, they are not.

3.2 Comparing the New and Old Programs

After determining that our program was running correctly for circle domains with only a single inner boundary arc, we decided we could begin to test our program's simulated data against the old circle domain program that did not use the teleportation process but instead more closely simulated Brownian motion by taking very small random steps. We used the simplest form of the circle domain that was possible at first by only considering circle domains whose inner boundary arcs were all symmetric about the positive x-axis as seen in Figure 10.

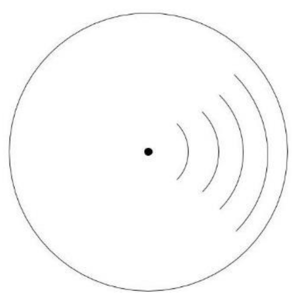

Figure 10. The circle domain Ω that was used.

Table 1. The information that was put into both programs to define the circle domain

Radius for arcs	Starting angle of arc	Ending angle of arc
0.4	$-\pi/32$	$\pi/32$
0.8	$-\pi/16$	$\pi/16$
1.2	$-\pi/8$	$\pi/8$
1.6	$-\pi/4$	$\pi/4$

In this example, the outer circle radius was still 2, and the number of paths the program simulated was 10,000. The new program created a graph seen in Figure 11.

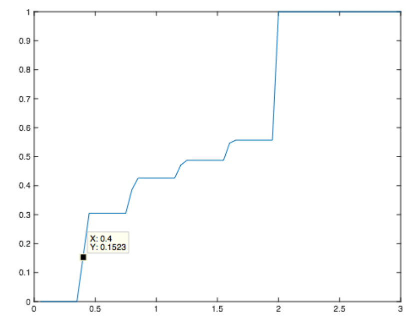

Figure 11. The h-function that the new program created for Table 1.

There are some points of interest on this graph. The points of interest tell the probability of the boundary being hit within that radius away from the starting position of the particle.

Table 2. Points of interest for Figure 11.

Arcs at each radius	Points of interest
1	(0.45, 0.3048)
2	(0.85, 0.4259)
3	(1.25, 0.4875)
4	(1.65, 0.5568)
Outer circle	(2, 1)

After each point of interest, the h-function remained constant until the next point of interest. We note that this all looks generally correct, with the steps corresponding with arc radii, the graph is never decreasing, and at 2, the graph reaches 1.

The old program created this graph seen in Figure 12. It should be noted that the line going diagonally across the graph is simply an artifact of the old program. This line means nothing for the purpose of our information and it should be ignored.

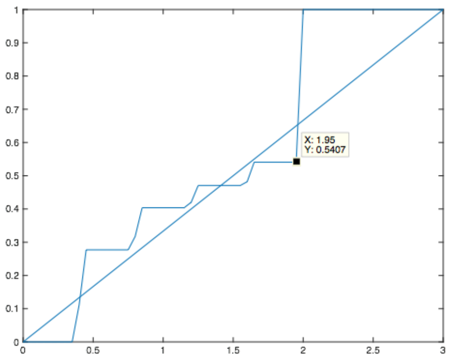

Figure 12. The h-function the old program created for Table 1.

There are also some points of interest for this graph.

Table 3. Points of interest for Figure 12.

Arcs at each radius	Points of interest
1	(0.45, 0.277)
2	(0.85, 0.4038)
3	(1.25, 0.4705)
4	(1.65, 0.5407)
Outer circle	(2, 1)

After each of the points of interest, the h-function remains constant again until the next point of interest. The y-coordinates of the points for both Figure 9 and Figure 10 are extremely similar to one another, proving that the new program produces similar results to the other one.

We found that when using the old program without teleportation, the average number of steps every path took before crossing the boundary in the circle domain from Figure 9 was 11,280. For the new program, the average number of steps it would have to take is about 19.6. Thus, the new program was about 575 times more computationally efficient than the old program in terms of the number of steps took, though it should be noted that the new program required more computation in finding the new minimum distance from the boundary at each step.

3.3 Other Types of Domains

After confirming that the program was running successfully with circle domains with one arc and multiple arcs symmetric about the x-axis, we wanted to see how using different domains would affect the h-functions.

The first such domain we tested was a circle domain where, if all the arcs had the same radius, the arcs would form a perfect circle as seen in Figure 13. This interested us particularly because we knew that if our circle domain's boundary consisted only of the outer circle, the h-function would only have one step up to 1 at the point on the x-axis that corresponded to the circle's radius. We wanted to see how likely it was for the particle to be able to maneuver around these arcs in teleportation before hitting the true outer circle compared to the old program taking very many small steps.

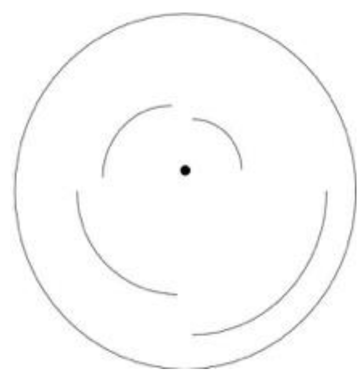

Figure 13. The circle domain.

Table 4. The information that was put into both programs to define Figure 13's circle domain

Radius for arcs	Starting angle of arc	Ending angle of arc
0.4	0	$\pi/2$
0.8	$\pi/2$	π
1.2	π	$3\pi/2$
1.6	$3\pi/2$	2π

In this domain, the outer circle radius is 2, and our program ran 10,000 paths. The new program created a graph seen in Figure 14.

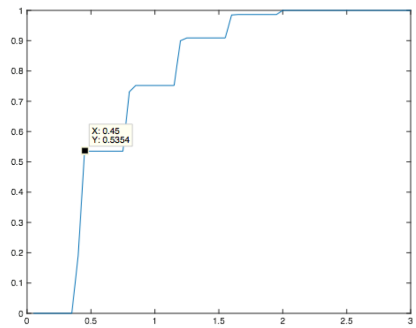

Figure 14. The h-function for the domain shown in Figure 13.

Table 5. Points of interest for Figure 14.

Arcs at each radius	Points of interest
1	(0.45, 0.5354)
2	(0.85, 0.7521)
3	(1.25, 0.9091)
4	(1.65, 0.9863)
Outer circle	(2, 1)

This all seems to make sense, it fits the qualifications for a reasonable step function. What was surprising about this information was that the particle had over a 50% probability of hitting the first arc, but the final arc only had a 7.72% increase of probability of being hit, even though they both covered the same angular distance and the final arc had a much larger arc length. This seems to suggest a strong correlation between probability of hitting a boundary and how close the boundary is to the origin of the particle. In the future, we hope to study this correlation more carefully and analyze the relationship between distance from the boundary and probability of hitting that arc more formally.

The above data indicated that the likelihood of a particle maneuvering around inner boundary arcs to hit the outer circle was very small. The probability of a particle hitting the boundary within the radius of the final arc is 98.63%, leaving only a 1.37% probability that a particle would find a way around all of the arcs, and cross the boundary of the full circle.

We computed the h-function for the domain in Figure 12 using the old program, producing the function seen in Figure 15.

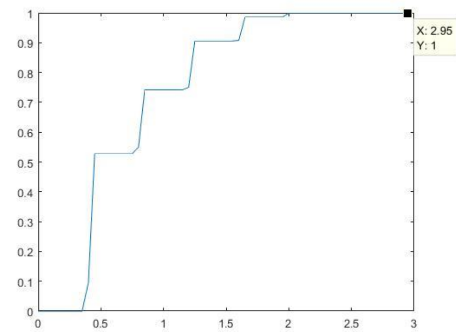

Figure 15. The h-function for the domain shown in Figure 13.

Table 6. Points of interest for Figure 15.

Arcs at each radius	Points of interest
1	(0.45, 0.5279)
2	(0.85, 0.7421)
3	(1.25, 0.9063)
4	(1.65, 0.9859)
Outer circle	(2, 1)

The probabilities for each arc are similar to the other program, again showing that the new program produces similar results to the old program. Also, this set of data supports the same correlation between the probability of hitting a boundary and how close the boundary is to the starting point of the particle. Additionally, based on these probabilities alone, it is slightly easier for a particle to maneuver around the arcs when the program takes small steps compared to the teleportation process, though this probability was still much smaller than we had hypothesized.

It should also be noted that the average number of steps that the old program took in a path before the particle crossed the boundary was 5,326.8, while for the new program, the process only took about 13.54 steps on average. This means that the new program took about 400 times fewer steps than the old program while still producing comparable data.

To further test the correlation between radius of an inner boundary arc and the probability the particle first hits that arc, we tested another circle domain seen in Figure 15. This domain was similar to the previous one in that if all the arcs had the same radius, they would form a circle. The difference in this boundary is that the arcs are generally closer to the origin as seen in Figure 16. We hoped to determine if the probability of hitting the arcs in the first and third quadrant (ie: in the second "ring" of arcs) was larger than the corresponding probabilities found in the previous domain.

Figure 16. The circle domain.

Table 7. The information that was put into both programs to define Figure 16's circle domain

Radius for arcs	Starting angle of arc	Ending angle of arc
0.4	0	$\pi/2$
0.8	$\pi/2$	π
0.4	π	$3\pi/2$
0.8	$3\pi/2$	2π

In this domain, the outer circle radius was 2, and our program simulated 10,000 paths. The new program created the graph seen in Figure 17.

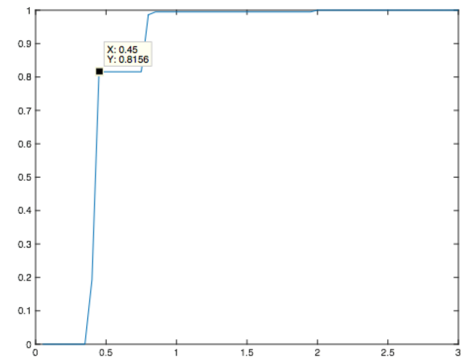

Figure 17. The h-function for the domain shown in Figure 16.

Table 8. Points of interest for Figure 17.

Arcs at each radius	Points of interest
1 and 3	(0.45, 0.8156)
2 and 4	(0.85, 0.9949)
Outer circle	(2, 1)

In the previous domain, the probability of crossing the boundary at the first and third arcs was only 69.24%, while for this domain this probability was 81.56%. This supports our hypothesis that if the boundary is closer to the origin of the particle, the probability of it being crossed is much larger than if the boundary was farther away.

We used the old program next to simulate this same domain, producing the graph seen in Figure 18.

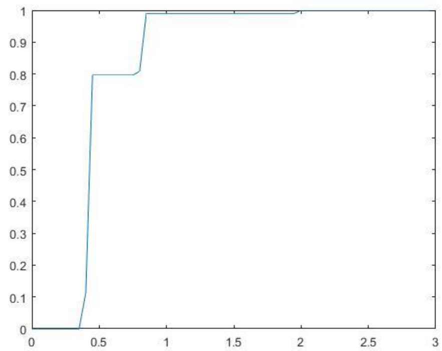

Figure 18. The h-function for the domain shown in Figure 16.

Table 9. Points of interest for Figure 18.

Arcs at each radius	Points of interest
1 and 3	(0.45, 0.7964)
2 and 4	(0.85, 0.9905)
Outer circle	(2, 1)

We found that the probabilities of hitting arcs 1 and 3 was again much higher than for the domain from Figure 12, and so our theorized correlation was once again supported.

We also noted that the average number of steps that the old program had to take before the particle crossed the boundary was approximately 5,289.3, while the new program on average took about 9.17 steps per path. This means that the new program took about 575 times fewer steps on average.

4. Conclusion

4.1 Results

For every domain that we have considered, the new program was consistently much more efficient than the old program in terms of average step count. The true efficiency is likely smaller due to the additional calculations needed to determine distance from the boundary at each step. Finding what the actual efficiency of the new program compared to the old program could be a possible direction in which to take this research. Also, the h-functions computed for each domain appeared extremely similar between the ones computed by the old program and those computed with teleportation. For these two reasons, we believe that the program was successfully updated to perform much more efficiently when considering circle domains.

4.2 Discussion

From our research, it becomes evident that using the method of teleportation is much more efficient than the method of many small steps of a particle. This means that we can apply this method to all of our programs to make them run more efficiently as well. This has progressed the research for h-functions because now we know how to make our programs run much more efficiently, allowing us to simulate the h-functions of more complicated domains much more effectively.

4.3 Evaluation

We encountered several obstacles throughout this work in the simulation process. One major issue was the error in measurement of the probabilities due to having to come within a small distance ε from the boundary rather than actually crossing it. There is not a way to truly fix this as long as we are using this type of program because there is a probability of zero that a particle hits the boundary at a particular point, meaning we need some tolerance $\varepsilon > 0$.

There are many choices for continuing this research. One approach is to further investigate the claim that was made about correlation between the probability of a boundary being hit and how close the boundary is to the particle's origin, especially when considering circle domains. We believe that if the boundary is closer to the origin of the particle when compared to the other boundaries, the probability of it being crossed is much larger.

Another future direction this could go in is to perform a more formal statistical analysis of the probabilities for a domain using both the new and old programs in order to make sure that the new program is producing results similar to the old program. Finally, one other direction for this project is to apply the process of teleportation to study other circle domains and finding possible patterns among h-functions of circle domains and the h-functions of more general types of complex domains as per the Step Function Program.

5. REFERENCES

[1] K. Gerstle, *Harmonic Measure distribution functions in complex domains*, poster (2010), Kenyon College.

[2] S. Kakutani, Two-dimensional Brownian motion and harmonic functions, *Proc. Imp. Acad. Tokyo* **20** (1944), 706-714.

[3] S. Matsumoto, What do random walks tell us about the shapes of regions in the complex plane?, poster (2011), Harvey Mudd College.

[4] M.A. Snipes, *Brownian motion and the shape of a region's boundary: generating step functions from circle domains*, Senior Thesis (1999), Harvey Mudd College.

[5] M.A. Snipes and L.A. Ward, *Harmonic measure distribution functions of planar domains: a survey*, in preparation.

[6] M.A. Snipes and L.A. Ward, *Realizing step functions as harmonic measure distributions of planar domains*, Complex Var. Elliptic Equ. **53** (2008), no. 10, 897-913.