# WXML Final Report: Discrete Hyperbolic Geometry and Random Walks

Xinyue Peng, Haoran Li

Project Mentor: Ryokichi Tanaka

Graduate Mentor: Anthony Sanchez

Winter 2017

## 1 Abstract

In this project, we will conduct the simulation of random walks on hyperbolic plane. There is still unresolved problem related to the question that whether it is uniformly distributed when a random walker escape from the unit disk. We will visualize the movement of random walk on hyperbolic plane and use unit disk as model to present the trajectory of random walks. We are going to perform simulation of Random Walk using computational program - Matlab, compute distribution plot and statistical histogram to help us with this problem.

## 2 Background and Motivation

### 2.1 Hyperbolic Plane

We will consider upper half plane as

$$\mathbf{H}^2 = \{(x, y) \ \in \mathbf{R}^2 : y > 0\}$$

Consider points in $\mathbf{H}^2$ as complex number

$$z = x + iy$$

We will rewrite $\mathbf{H}^2$ as :

$$\mathbf{H}^2 = \{(x, y) \ \in \mathbf{R}^2 : y > 0\}$$

## 2.2 Application and Method

We will consider random walk on hyperbolic plane as following, supppose

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} a, b, c, d \in \mathbf{R}, ad - bc \neq 0$$

Consider the **fractional linear** action on $\mathbf{H}^2$ via

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot z = \frac{az + b}{cz + d}$$

For example

$$\begin{bmatrix} 3 & b \\ 4 & 1 \end{bmatrix} \cdot z = \frac{3z + b}{4z + 1}$$

Random walk proceed faster on hyperbolic plane comparing the walk on Euclidean plane. To learn more about the random walk pattern on hyperbolic plane, we will start with initial point on unit disk, which also regard as our model, and transform the points from unit disk to hyperbolic plane while perform random walk on hyperbolic plane. In order to show the random walk points on hyperbolic plane, we will transform the points from hyperbolic plane to unit disk.

We will consider finite number of steps random walk. In this case, there will be a finite collection of rotational matrices.

$$A_1, A_2, A_3, ...A_k$$

For each matrix, there will be a probability such that determines the steps of the random walker.

$$p_1, p_2, p_3, ...p_k$$

with

$$0 \leq p_i \leq 1 \ and \sum_i p_i = 1$$

In this project, the rotational matrices applied are following

$$A = \begin{bmatrix} \cos\frac{\theta}{a} & \sin\frac{\theta}{a} \\ -\sin\frac{\theta}{a} & \cos\frac{\theta}{a} \end{bmatrix}$$

$$B = \begin{bmatrix} \cos\frac{\theta}{b} & t\sin\frac{\theta}{b} \\ -\frac{1}{t}\sin\frac{\theta}{b} & \cos\frac{\theta}{b} \end{bmatrix}$$

where

$$t + \frac{1}{t} = 2\frac{\cos\frac{\pi}{a}\cos\frac{\pi}{b} + \cos\frac{\pi}{c}}{\sin\frac{\pi}{a}\sin\frac{\pi}{b}}$$

(There are two roots of t and we will take the larger one. The choice of $\theta$ can be varied.)

# 3 Computation Result

We will study different collections of matrices and probabilities, and draw pictures of the paths that we get in $\mathbf{H}^2$.

Below are a simulation plot of Random Walks with 1000 steps and a plot of a single random walk generated by four rotational matrices.

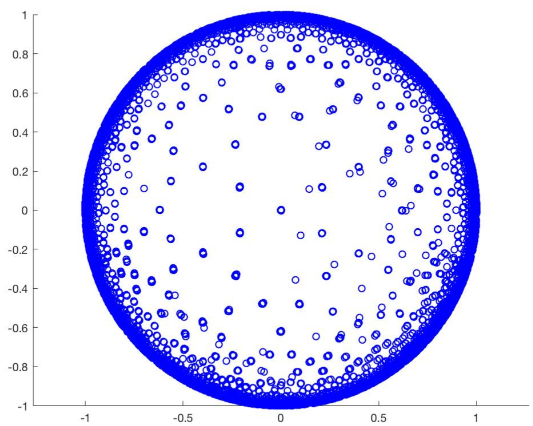

Figure 1: Complete trajectory of random walks

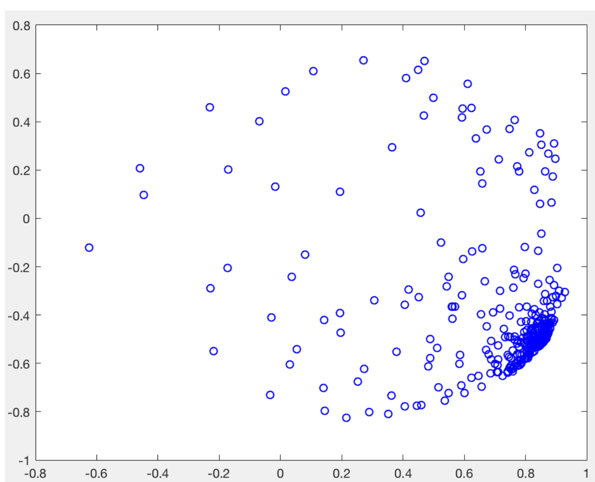

Figure 2: Random Walk generated by $\mathbf{A}, \mathbf{A}^{-1}, \mathbf{B}, \mathbf{B}^{-1}$

Below is a plot of 1000 random walk distributed around radius 0.9 stage.

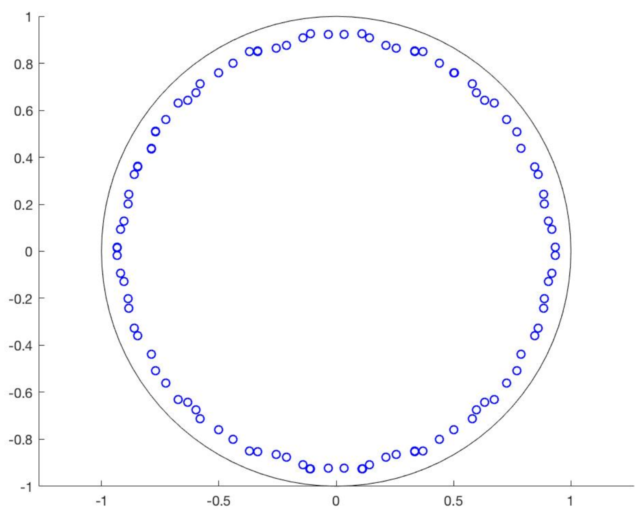


Figure 3: Distribution of random walk around radius 0.9

To be more interesting, if we have a considerable large number of random walks, i.e. 2000 and 3000 random walks, we will understand how the random walk perform for most of the cases, which could help us understand the problem of the distribution of escaping location on a unit disk. In order to see the result virtually, we have used Matlab to simulate the random walk behavior, recorded every escape point and plot the relation of frequency of escaping and locations about the edge of the unit disk.

We computed the histograms of distribution of different number of random walks at the 2500th step, which is also the last step based on our assumption.

On the histograms, x-axis represents locations on the unit disk, from $-\pi$ to $\pi$ and y-axis represents for frequencies of escaping from corresponding locations. For comparison, we have also tested random walk for 4000 random walks and 5000 random walks, if there are preferred locations on the unit disk, we can see there are higher columns on some locations than others.

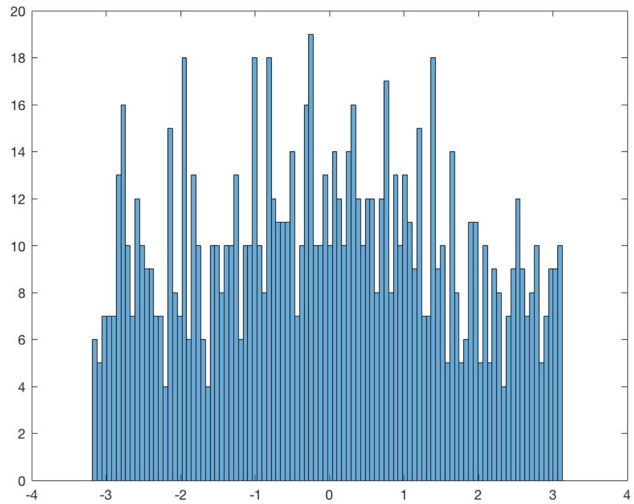


Figure 4: 2000 Steps,Range: $[-\pi, \pi]$

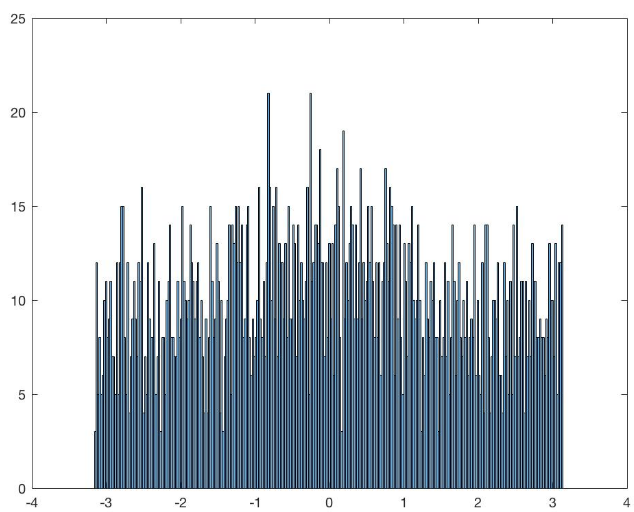


Figure 5: 3000 Steps,Range: $[-\pi, \pi]$

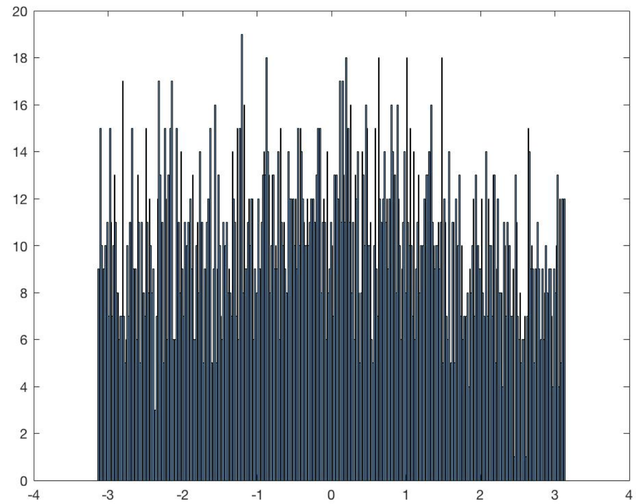


Figure 6: 4000 Steps,Range: $[-\pi, \pi]$

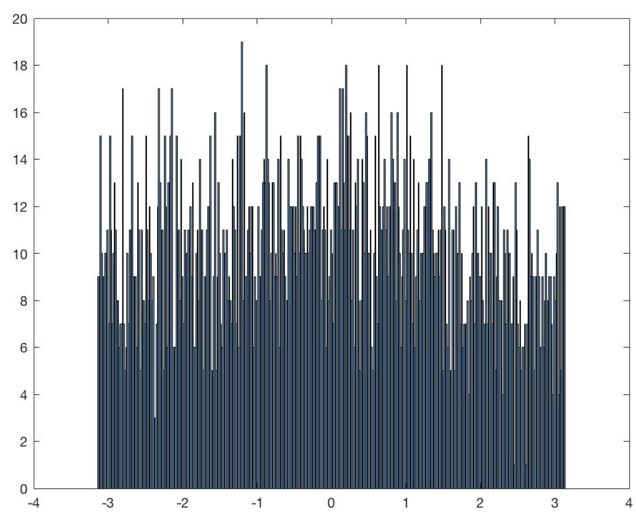


Figure 7: 5000 Steps,Range: $[-\pi, \pi]$

Below are histograms for comparing our random walk histogram using uniform distribution. Each "point" will randomly generate a escaping location so that each location has an equal probability to choose.

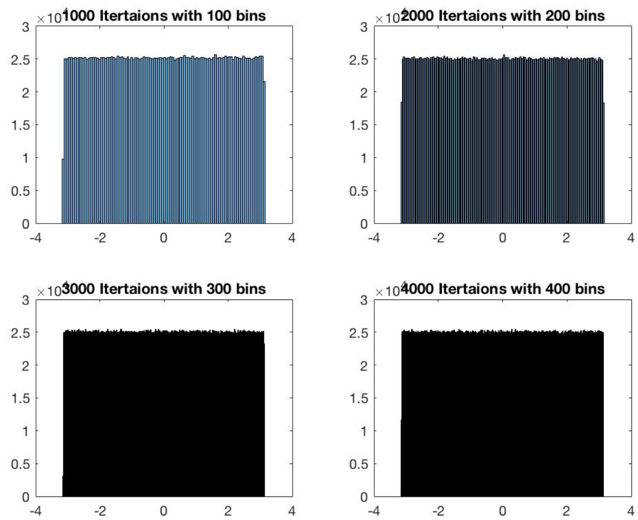


Figure 8: Uniform distribution Histogram

# 4 Conclusion and Summary

From the comparison of Random Walk radiant distribution histograms and uniform distribution histograms, we can conclude that the distribution of Random Walk is not uniformly distributed. There exists a preferred area of the hyperbolic plane that random walk will accumulate after a finite number of steps.

# A Matlab Code for random walk

```
close all;clear all;clc

a = 2;b = 3;c = 7;t = 1.32725;v = 2*((cos(pi/a)*cos(pi/b)+cos(pi/c))/(sin(pi/a)*sin(
x0 = 0+0i ; % initial point at unit disk
x1 = -i*((x0+1)/(x0-1)); %transform the point to upper half plane
l = 50;
theta = pi/16;% radient
A = [cos(pi/a), sin(pi/a);-sin(pi/a),cos(pi/a)];%rotational matrix A
Ainv = inv(A);
B = [cos(pi/b) , t*sin(pi/b);-(1/t)*(sin(pi/b)), cos(pi/b)]; %rotational matrix B
Bsq = B^2;
Binv = inv(B);
N = 2000;D = 0.9;%threshold
P = zeros(l*l,N);

%%
AA = [];
for j = 1:N
    nums = mod(reshape(randperm(l*l), l*l, 1), 3 );%random permutation
    M = eye(2); %identity matrix
    for k = 1:l*l
        if nums(k,1) == 0
            M = M*A;
        elseif nums(k,1) == 1
            M = M*B;
        elseif nums(k,1) == 2
            M = M*Bsq;
```

```
        end
        x2 = getrotateA(x1,M);%random walk step, get a new point on upper half plane
        x3=(x2-i)/(x2+i); %send the points back to unit disk
        P(k,j) = x3;
        AA(k,j) = angle(P(k,j));% store points
    end
end
%% Distribution of radiant of points at 2500th radius
MM = [];
for i = 1:N

    a = angle(P(2500,i));
    MM = [MM a];

end
figure(4)
histogram(MM,500)
```