

# MXB261 Problem-Solving Task

Braydan Newman: n11272031

September 1, 2023

## Contents

<b>1</b>	<b>Part 1 - A Biased Random Walk</b>	<b>2</b>
1.1	Single vs Random Starting Point . . . . .	2
1.2	Increasing the Particles Count . . . . .	3
1.3	Observations on the Distribution of Particles . . . . .	4
<b>2</b>	<b>Part 2 - Sampling from Experimental Data</b>	<b>5</b>
2.1	Similarities in the 2 Data Sets . . . . .	5
2.2	Kullback-Leibler Measurements on the Data Sets . . . . .	5
2.3	The Effect of Changing the Number of Bins . . . . .	5

# 1 Part 1 - A Biased Random Walk

## 1.1 Single vs Random Starting Point

As shown in Figure 1 and 2, having a single starting point vs a random starting point changes the distribution of column heights drastically.

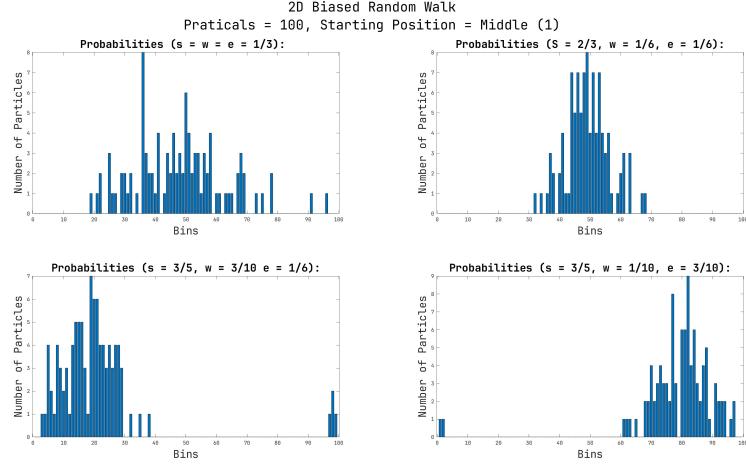


Figure 1: Middle Starting Position - 100 particles

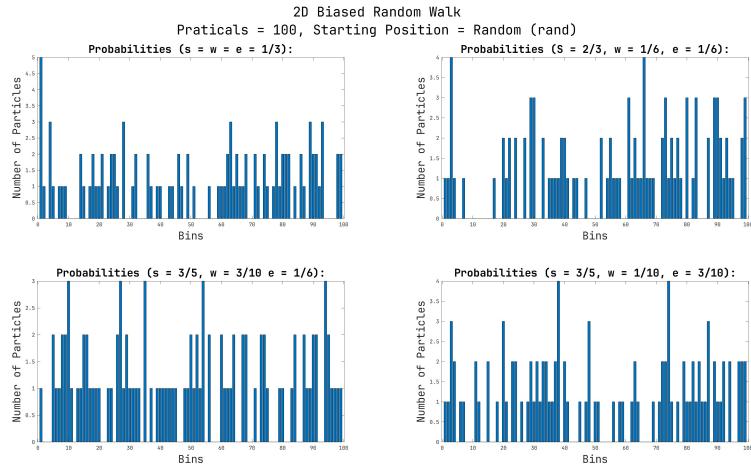


Figure 2: Random Starting Position - 100 particles

Figure 1 shows distinct groupings of particle locations while Figure 2 shows no real discernible groupings of particles, and is spread out evenly with all column heights within 4 particles of each other. This shows that because of the randomness of the start position, the end location will be just as random even with the different biases in the directions. This could change if the simulation area didn't wrap around. The middle start position showed a strong tendency for all the particles to fall together, as the biases in the direction add together over many particles to increase the likelihood of a particle falling in a particular area.

## 1.2 Increasing the Particles Count

Increasing the particle count in the simulations does not change much in terms of the overall shape and distribution. Both the random and middle starting position have the same falling patterns as their respective 100 particle simulations and adding more particles doesn't change this. If the particle count was much smaller for the middle simulation, this would cause the graph to look a lot more random as the small differences between groups, or trivial correlations, would be detected as being statistically significant. However adding more particles makes the groupings more defined as shown in Figure 3. This is because with more data samples this reduces the amount of noise caused by the randomness. This is also shown in the random start position simulation (Figure 4), where the graph becomes much more level and flat, and with more particles this will continue to flatten out and become a uniform height across the whole graph.

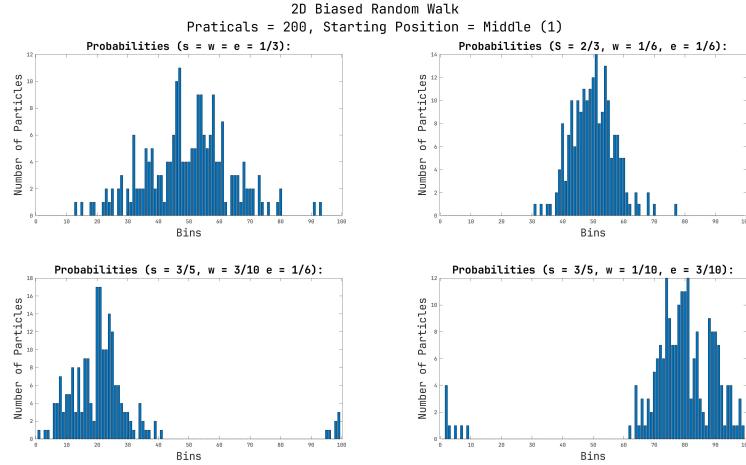


Figure 3: Middle Starting Position - 200 particles

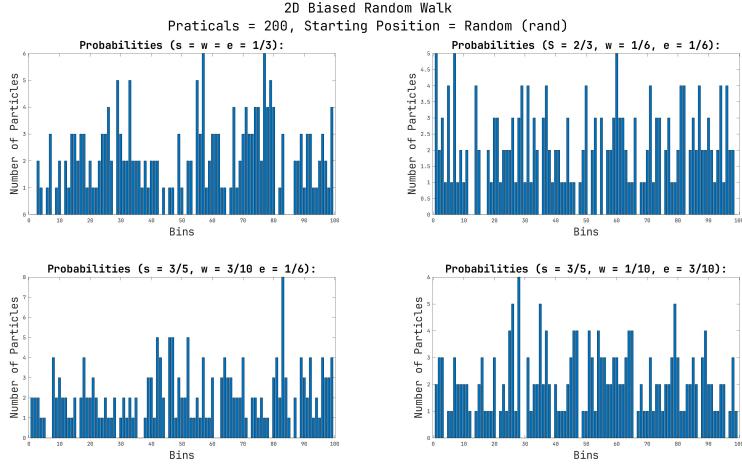


Figure 4: Random Starting Position - 200 particles

### 1.3 Observations on the Distribution of Particles

The middle starting position simulations all approximates a normal distribution, this is shown best by looking at the top left subplot in Figure 3 where all directions have equal probability. While the top left shows a unit normal distribution all the other subplots for all simulations with the middle starting position are normal distribution too. These subplots just have different Skewness and kurtosis values.

This can been see more clearly when looking at the probability of the simulation area. This is a variation of this simulation with 2 possible directions instead of three, with equal probability for simplicity. Notice that the closer to the edges the less likely it is for the particle to make it there while vise versa in the center. The further down it goes the more prevalent it becomes.

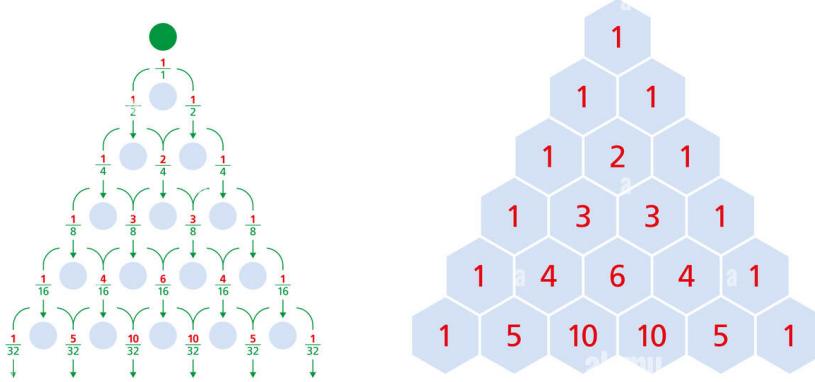


Figure 5: Galton Board Normal Distribution and Pascals Triangle

## 2 Part 2 - Sampling from Experimental Data

### 2.1 Similarities in the 2 Data Sets

The data sets are very similar as seen in the probability distribution of the two dataset seen in Figure 6. They both follow basically the same pattern and this is to be expected. As the CDF used to sample data from to create the 2nd dataset was made with the 1st Dataset. The variation comes from the randomness of the samples collected. If more samples where collected it would converge on the first dataset.

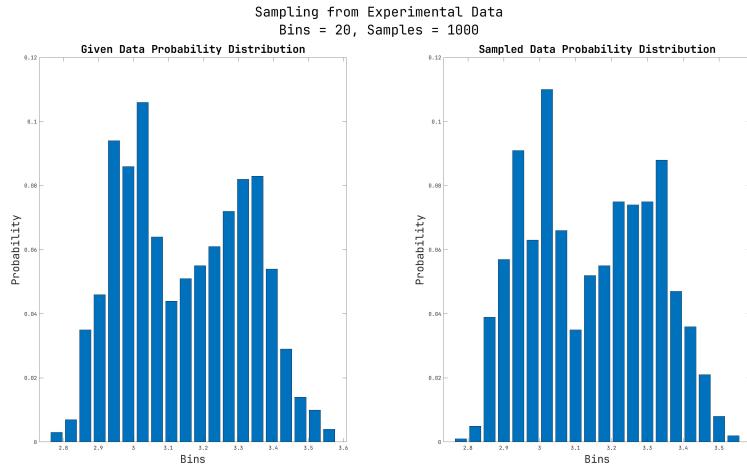


Figure 6: Sampling from Experimental Data - 20 Bins

### 2.2 Kullback-Leibler Measurements on the Data Sets

The Kullback–Leibler divergence, also known as relative entropy, is a measure of how one probability distribution differs from another.

For the Datasets calculated with 20 Bins (Original and generated) the divergence can be different depending on which one you are measuring against, as seen in equation 1.

$$\begin{aligned} D_{KL}(P||Q) &= 0.0134 \\ D_{KL}(Q||P) &= 0.0128 \end{aligned} \tag{1}$$

The K-L divergence of two random variables is an expected value, and so it matters which distribution you're taking the expectation with respect to.

### 2.3 The Effect of Changing the Number of Bins

The effect of changing the number of bins has a drastic effect of the Kullback–Leibler Measurement with a lower amount of bins the chance of having divergent

bins is lower decreasing the Kullback-Leibler Measurement and vise versa for increasing the bin count.

This is shown in Figure 7 and Figure 8 where Figure 7 is the decreased bin count of 10 and Figure 8 has an increased bin count of 40. The amount of variations between the two subplot graphs is far higher in the 40 bin PDF then the far smaller amount of variations in the 10 bin PDF. This is proven in the Kullback-Leibler Measurements for both PDF's comparisons.

(2) = 10 Bins - figure 7

(3) = 40 Bins - figure 8

$$D_{KL}(P||Q) = 0.0047 \quad (2)$$

$$D_{KL}(Q||P) = 0.0045$$

$$D_{KL}(P||Q) = 0.0184 \quad (3)$$

$$D_{KL}(Q||P) = 0.0178$$

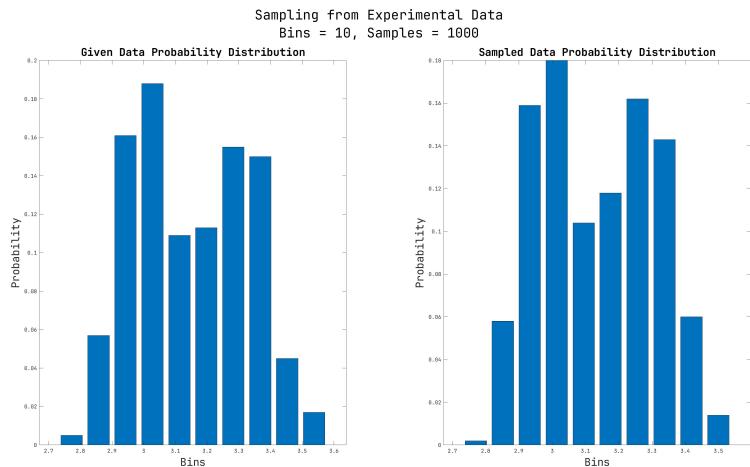


Figure 7: Sampling from Experimental Data - 10 Bins

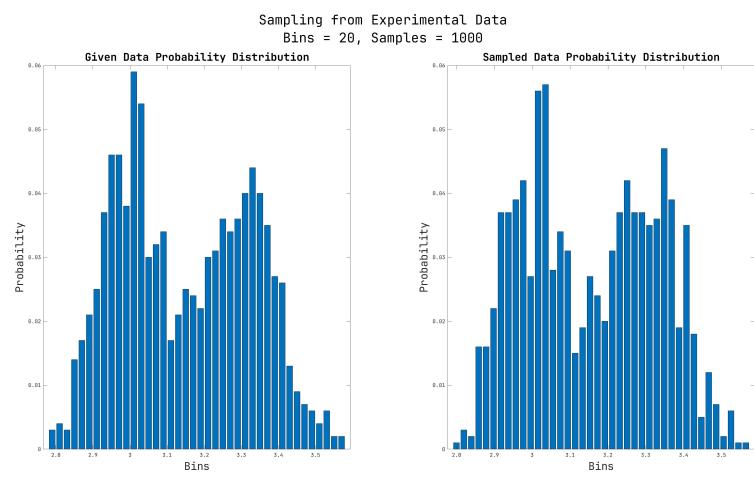


Figure 8: Sampling from Experimental Data - 40 Bins

**GITHUB:** [https://github.com/BraydanNewman/MXB261\\_PST](https://github.com/BraydanNewman/MXB261_PST)

**PART 1 CODE**

**main.m**

```
%% Plot 1
N = 100;
P = "1";

subplot(2,2,1)
[results] = ball_drop(1/3, 1/3, 1/3, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = w = e = 1/3):', 'FontSize', 20)

subplot(2,2,2)
[results] = ball_drop(2/3, 1/6, 1/6, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (S = 2/3, w = 1/6, e = 1/6):', 'FontSize', 20)

subplot(2,2,3)
[results] = ball_drop(3/5, 3/10, 1/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 3/10 e = 1/6):', 'FontSize', 20)

subplot(2,2,4)
[results] = ball_drop(3/5, 1/10, 3/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 1/10, e = 3/10):', 'FontSize', 20)

sgtitle(["2D Biased Random Walk", "Practicals = 100, Starting Position = Middle (1)"], 'FontSize',
25)

%% Plot 2
N = 200;
P = "1";

subplot(2,2,1)
[results] = ball_drop(1/3, 1/3, 1/3, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = w = e = 1/3):', 'FontSize', 20)

subplot(2,2,2)
```

```

[results] = ball_drop(2/3, 1/6, 1/6, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (S = 2/3, w = 1/6, e = 1/6):', 'FontSize', 20)

subplot(2,2,3)
[results] = ball_drop(3/5, 3/10, 1/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 3/10 e = 1/6):', 'FontSize', 20)

subplot(2,2,4)
[results] = ball_drop(3/5, 1/10, 3/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 1/10, e = 3/10):', 'FontSize', 20)

sgtitle(["2D Biased Random Walk", "Practicals = 200, Starting Position = Middle (1)"], 'FontSize',
25)

%% Plot 3
N = 100;
P = "rand";

subplot(2,2,1)
[results] = ball_drop(1/3, 1/3, 1/3, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = w = e = 1/3):', 'FontSize', 20)

subplot(2,2,2)
[results] = ball_drop(2/3, 1/6, 1/6, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (S = 2/3, w = 1/6, e = 1/6):', 'FontSize', 20)

subplot(2,2,3)
[results] = ball_drop(3/5, 3/10, 1/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 3/10 e = 1/6):', 'FontSize', 20)

subplot(2,2,4)
[results] = ball_drop(3/5, 1/10, 3/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)

```

```

xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 1/10, e = 3/10):', 'FontSize', 20)

sgtitle(["2D Biased Random Walk", "Practicals = 100, Starting Position = Random (rand)"],
'FontSize', 25)

%% Plot 4
N = 200;
P = "rand";

subplot(2,2,1)
[results] = ball_drop(1/3, 1/3, 1/3, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = w = e = 1/3):', 'FontSize', 20)

subplot(2,2,2)
[results] = ball_drop(2/3, 1/6, 1/6, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (S = 2/3, w = 1/6, e = 1/6):', 'FontSize', 20)

subplot(2,2,3)
[results] = ball_drop(3/5, 3/10, 1/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 3/10 e = 1/6):', 'FontSize', 20)

subplot(2,2,4)
[results] = ball_drop(3/5, 1/10, 3/10, N, P);
bar(results)
ylabel('Number of Particles', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Probabilities (s = 3/5, w = 1/10, e = 3/10):', 'FontSize', 20)

sgtitle(["2D Biased Random Walk", "Practicals = 200, Starting Position = Random (rand)"],
'FontSize', 25)

```

### **ball\_drop.m**

```

function [results] = ball_drop(s, w, e, N, P)
    board_height = 99;
    board_width = 99;

    game_board = zeros(board_height, board_width);

    for ball_index = 1:N

        if P == "rand"
            start_x = randi([1, board_width]);

```

```

elseif P == "1"
    start_x = (board_width + 1) / 2;
end

current_x = start_x;
current_y = board_height;

next_x = current_x;
next_y = current_y;

stopped = false;

while ~stopped
    u = rand;
    if u < w
%        LEFT
        temp = current_x - 1;
        if temp < 1
            temp = board_width;
        end

        if game_board(temp, current_y) ~= 1
            next_x = temp;
        end
    elseif u < w + e
%        RIGHT
        temp = current_x + 1;
        if temp > board_width
            temp = 1;
        end

        if game_board(temp, current_y) ~= 1
            next_x = temp;
        end
    elseif u < w + e + s
%        DOWN
        next_y = current_y - 1;
        if game_board(next_x, next_y) == 1 || next_y == 1
            game_board(current_x, current_y) = 1;
            stopped = true;
        end
    end
    current_x = next_x;
    current_y = next_y;
end

results = zeros(1, board_width);
for i = 1:board_width
    results(i) = sum(game_board(i,:));
end

```

```
end
```

## PART 2 CODE

### main.m

```
%% Plot 1
bins = 20;
samples = 1000;
[KLD1, KLD2, X, b, X1,b1] = sampling(bins, samples);

KLD1
KLD2

subplot(1, 2, 1)
bar(X, b)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Given Data Probability Distribution', 'FontSize', 20)

subplot(1, 2, 2)
bar(X1, b1)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Sampled Data Probability Distribution', 'FontSize', 20)

sgtitle(["Sampling from Experimental Data", "Bins = 20, Samples = 1000"], 'FontSize', 25)

%% PLOT 2
bins = 10;
samples = 1000;
[KLD1, KLD2, X, b, X1,b1] = sampling(bins, samples);

KLD1
KLD2

subplot(1, 2, 1)
bar(X, b)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Given Data Probability Distribution', 'FontSize', 20)

subplot(1, 2, 2)
bar(X1, b1)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Sampled Data Probability Distribution', 'FontSize', 20)

sgtitle(["Sampling from Experimental Data", "Bins = 10, Samples = 1000"], 'FontSize', 25)

%% Plot 3
bins = 40;
samples = 1000;
```

```

[KLD1, KLD2, X, b, X1,b1] = sampling(bins, samples);

KLD1
KLD2

subplot(1, 2, 1)
bar(X, b)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Given Data Probability Distribution', 'FontSize', 20)

subplot(1, 2, 2)
bar(X1, b1)
ylabel('Probability', 'FontSize', 20)
xlabel('Bins', 'FontSize', 20)
title('Sampled Data Probability Distribution', 'FontSize', 20)

sgtitle(["Sampling from Experimental Data", "Bins = 20, Samples = 1000"], 'FontSize', 25)

```

### **sampling.m**

```

function [KLD1, KLD2, X, b, X1,b1] = sampling(bin, N)
    % Set Up
    load("sampledData2023.mat");

```

```

figure(1)
h = histogram(Data0, bin);
b = h.Values / length(Data0);
c = cumsum(b);
X = h.BinEdges(1:bin)+h.BinWidth/2;

```

```

rng(4)
DataNew = zeros(1, N);
for i = 1:N
    u = rand;
    index = find(c>u,1);
    DataNew(1,i) = X(index);
end

```

```

figure(2)
h1 = histogram(DataNew, bin);

b1 = h1.Values / length(Data0);
X1 = h1.BinEdges(1:bin)+h1.BinWidth/2;

```

```

KLD1 = 0;
for i = 1:bin
    KLD1 = KLD1 + b(i) * log(b(i)/b1(i));
end

```

```

KLD2 = 0;
for i = 1:bin

```

```
KLD2 = KLD2 + b1(i) * log(b1(i)/b(i));  
end  
end
```