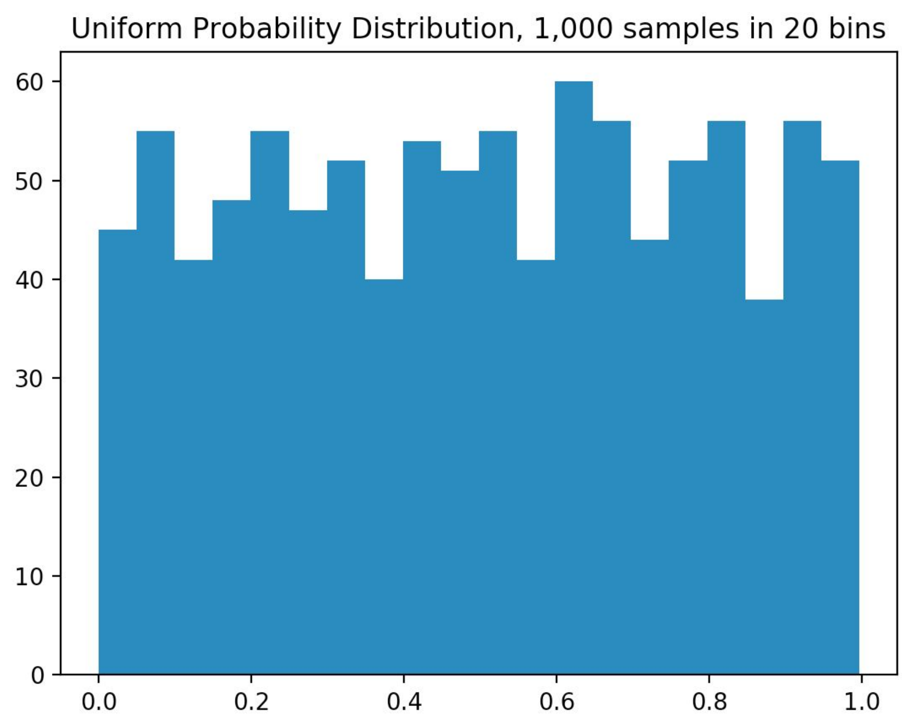
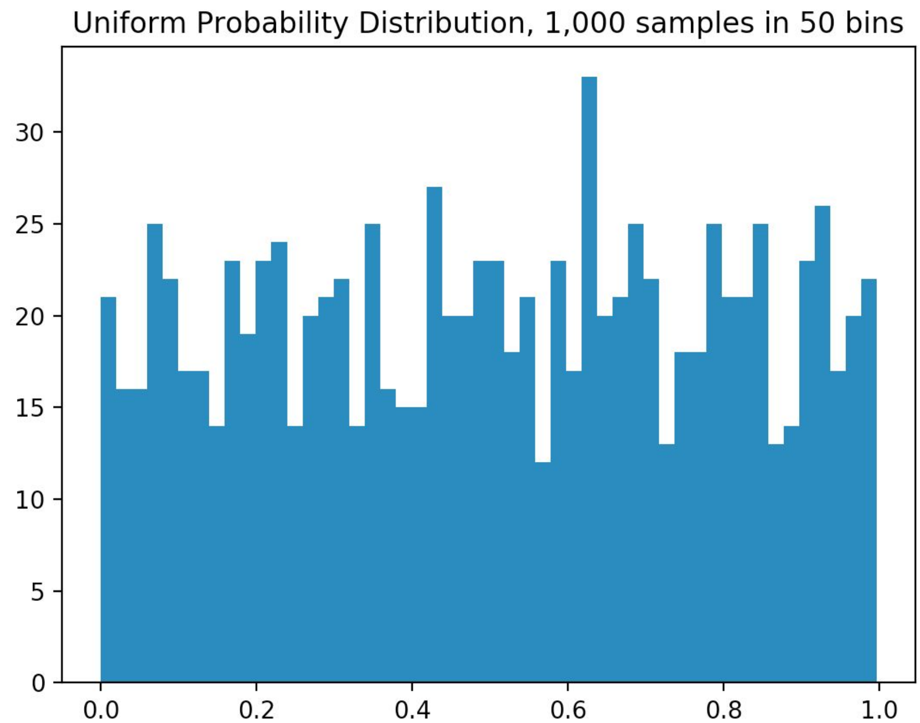
Amelia Doetsch GH3772
William Hanley GJ0637
Michel Kaadi
Alex Kaddis FV6117
PHY 6860

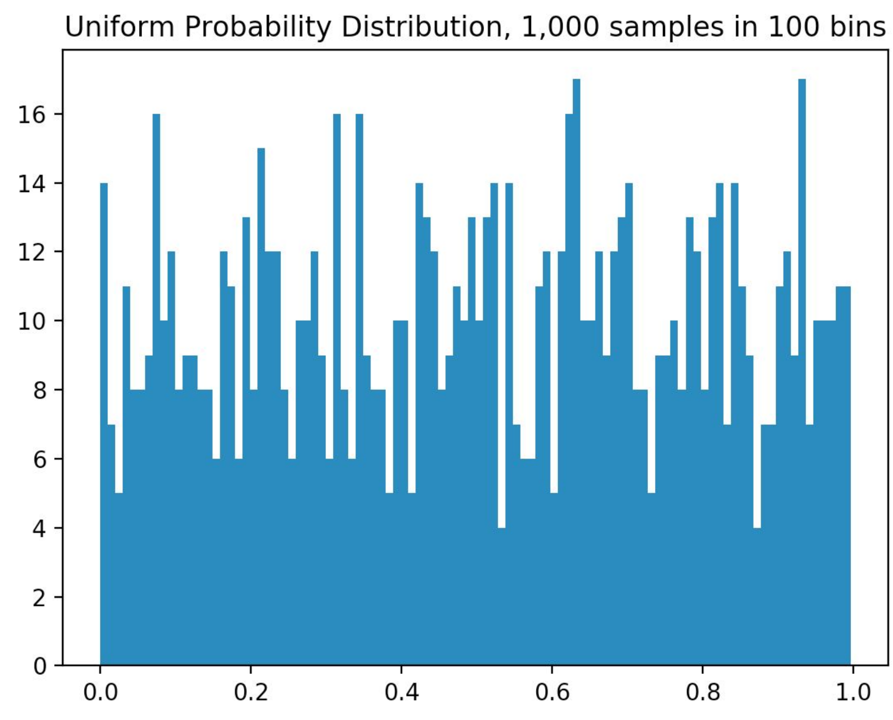
Random Numbers and Random Walk

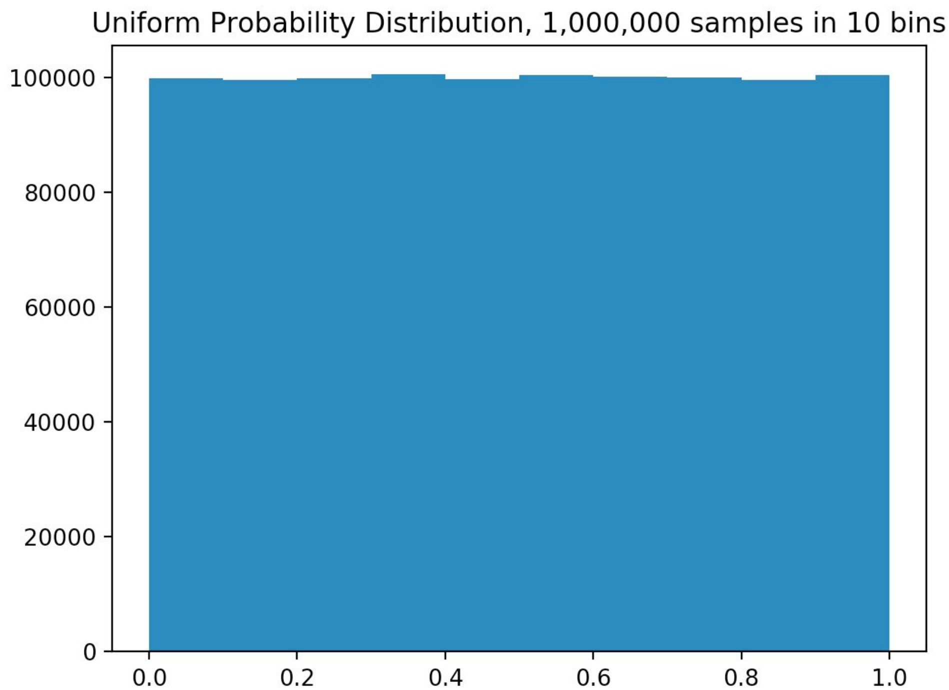
Random Numbers

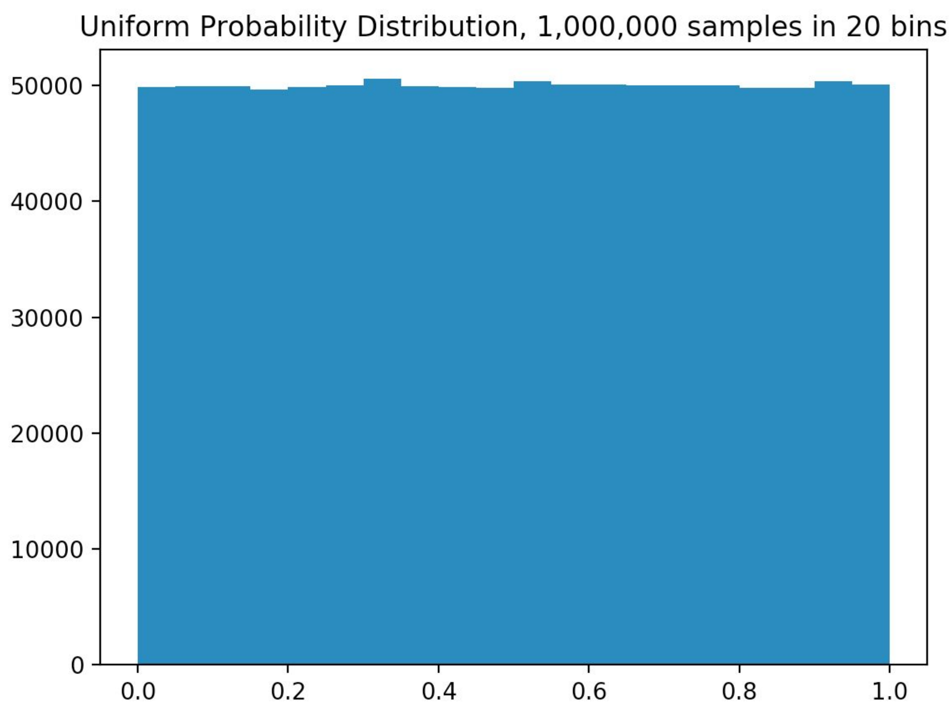
a) The purpose of this problem was to write the code for a random number generator. Generating random numbers is a useful application for modeling many systems in physics such as: the motion interacting of gas molecules, probability of a particles location given some potential, and systems of many particles. While there is no algorithm to generate truly “random” numbers, there are many methods (Monte Carlo, linear congruence) which can be used to generate “pseudo-random” numbers. For this problem in particular, the random.random() python function was used to generate a uniform random distribution of numbers between 0 and 1. These random numbers were stored in a list of size 1,000 and 1,000,000 samples with varying resolution.

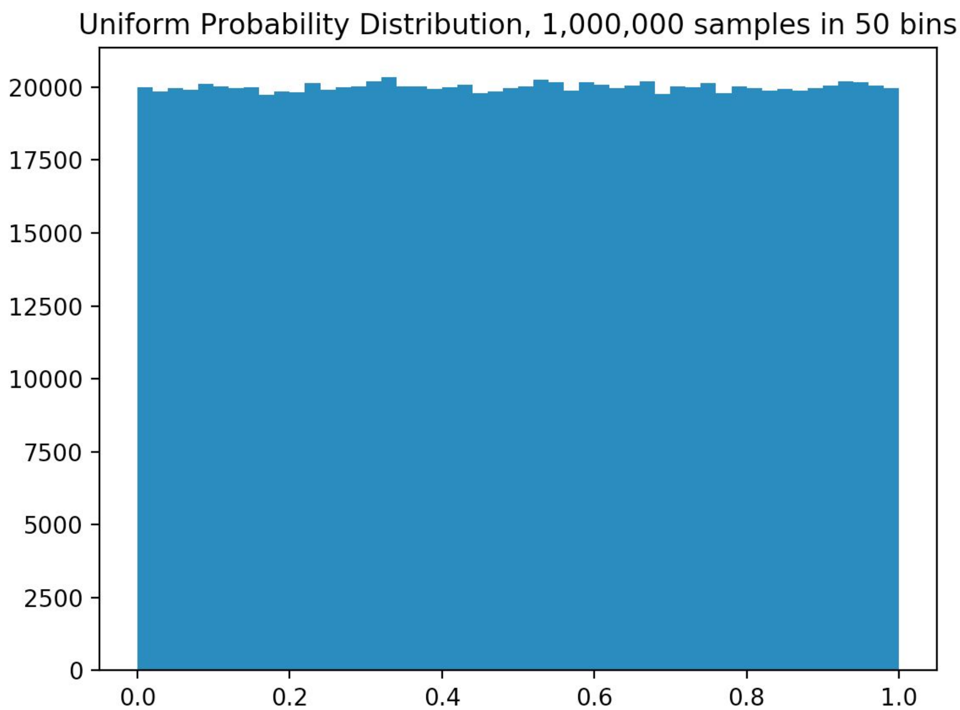


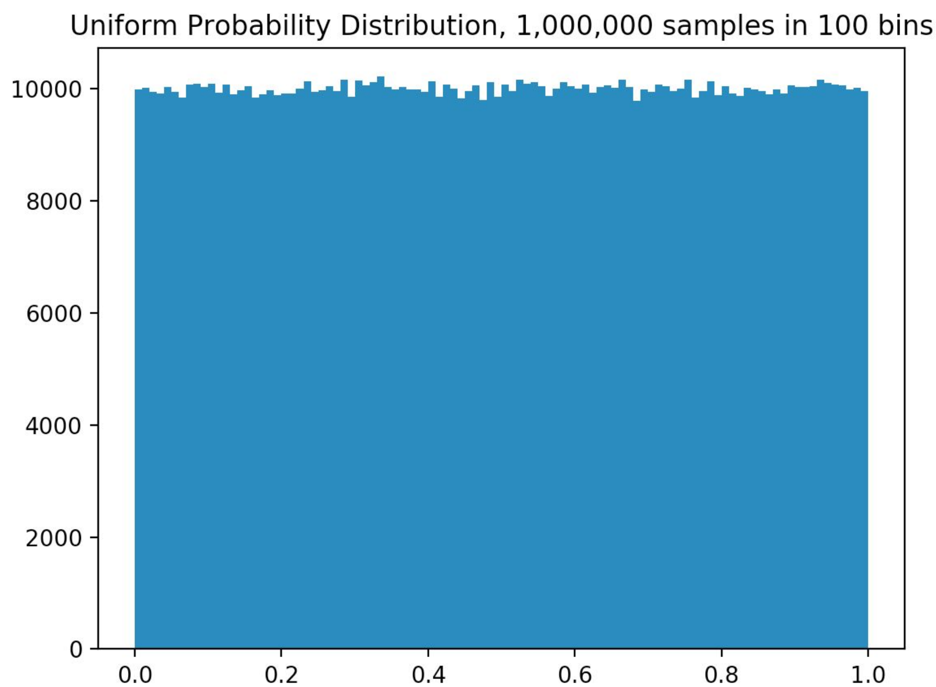

Uniform Probability Distribution, 1,000 samples in 50 bins
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0


Uniform Probability Distribution, 1,000 samples in 100 bins
0
2
4
6
8
10
12
14
16
0.0
0.2
0.4
0.6
0.8
1.0


Uniform Probability Distribution, 1,000,000 samples in 10 bins
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0


Uniform Probability Distribution, 1,000,000 samples in 20 bins
50000
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8
1.0



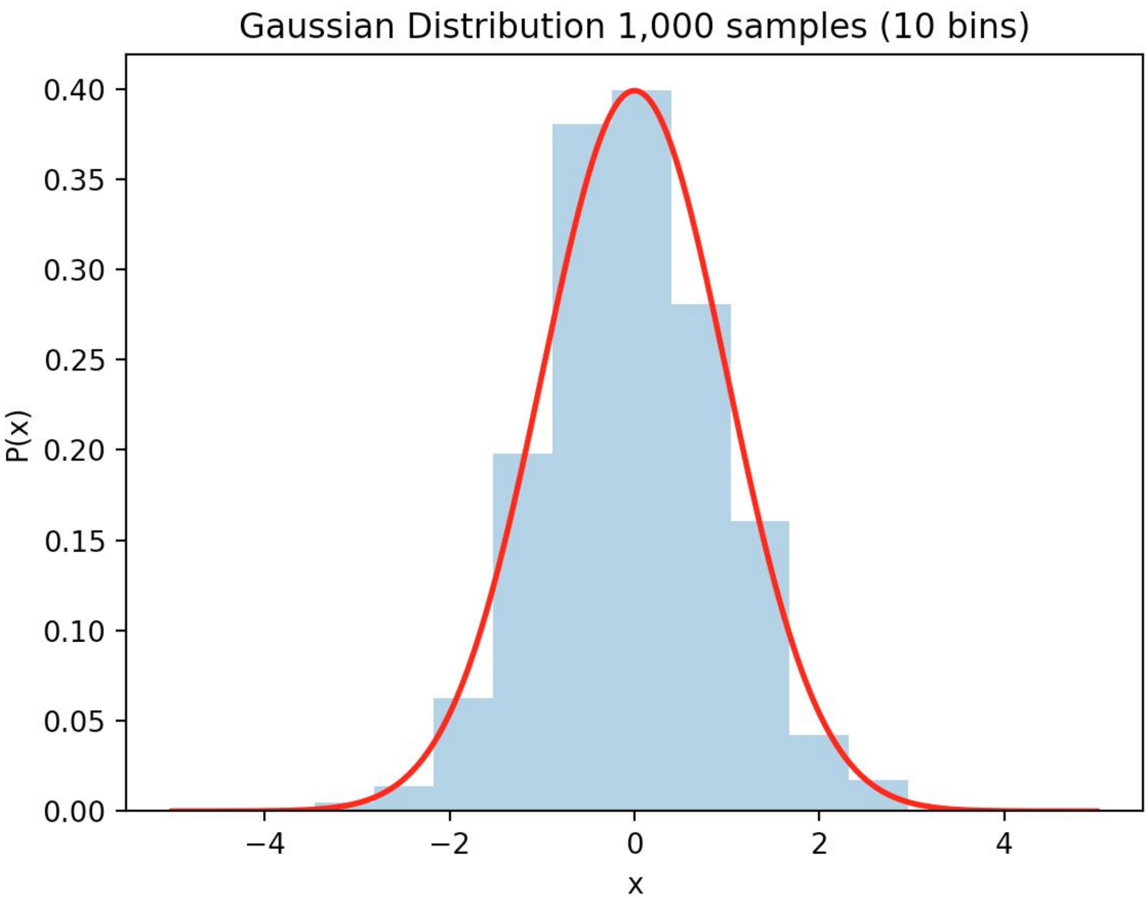


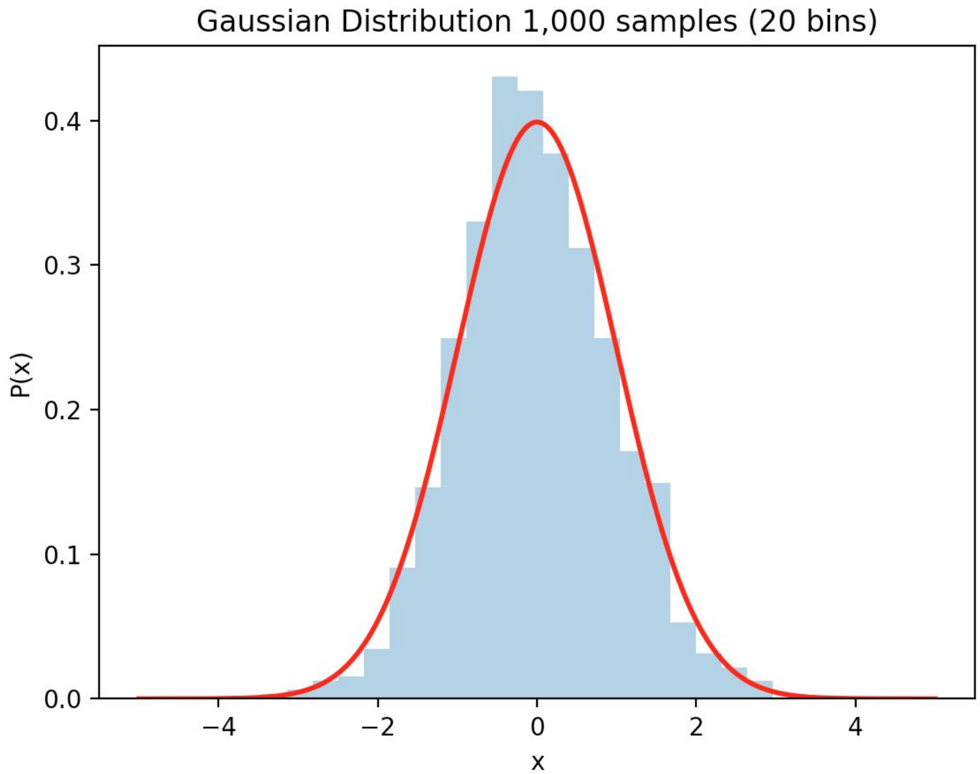
b) The purpose of this problem was to use a random number generator to fit a Gaussian distribution. This could be done using the “Rejection method” which worked by sampling

a selected amount of random numbers between some range and “accepting” the sample by storing it in a list if it falls below the value of a given Gaussian function.
The Gaussian function used was:

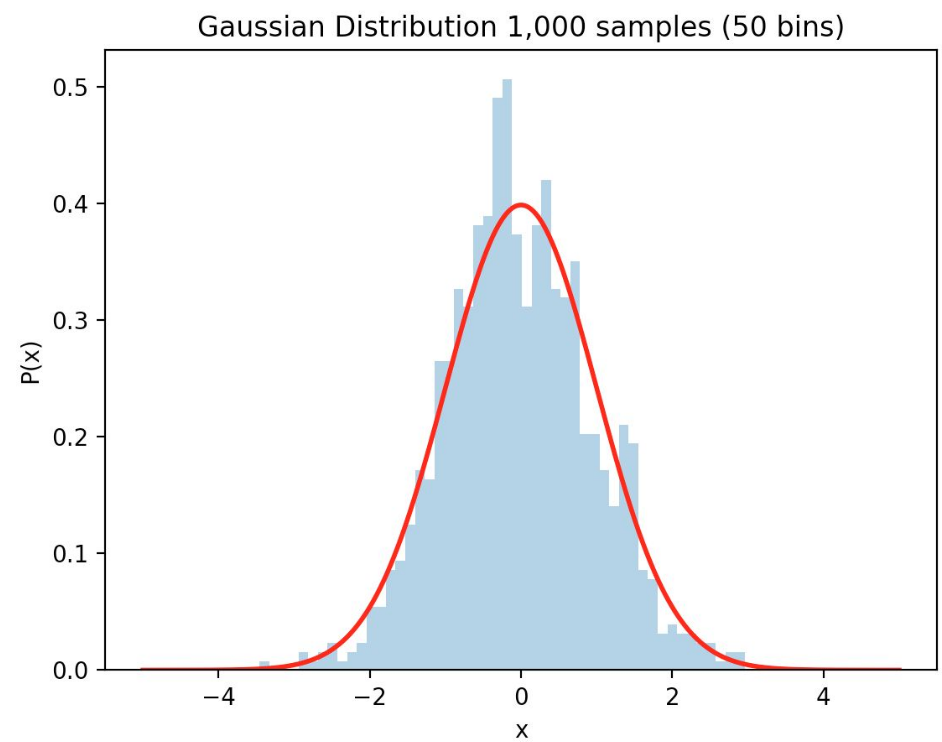
$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{1}{2\sigma^2}x^2)$$

For this method an x value was sampled between -5 and 5 and for that x a y was sampled between 0 and 1. If the x value substituted into the Gaussian function gave a value larger than the y sample then that y sample was accepted, and if not a new x was sampled until we reached the selected number of samples.

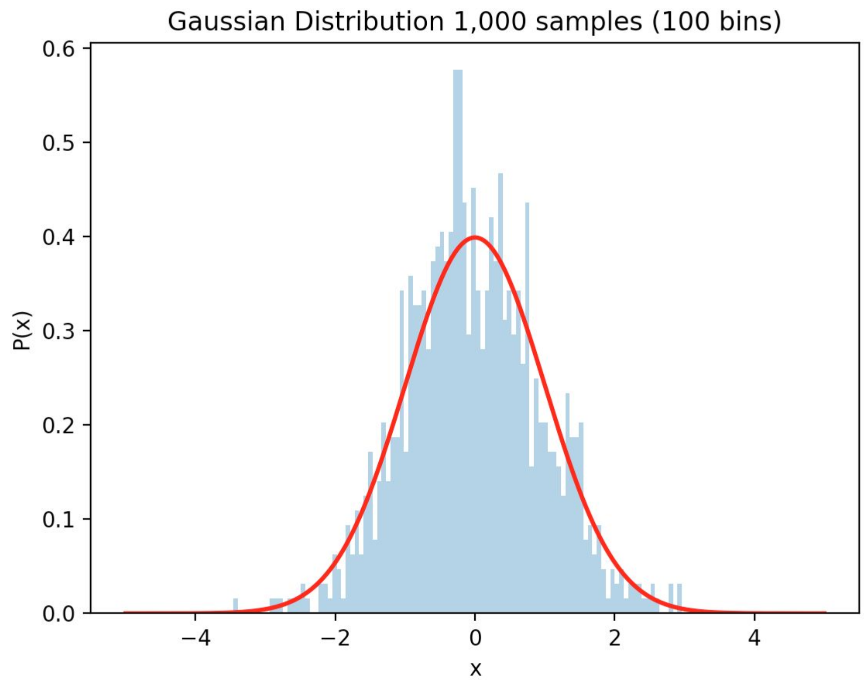




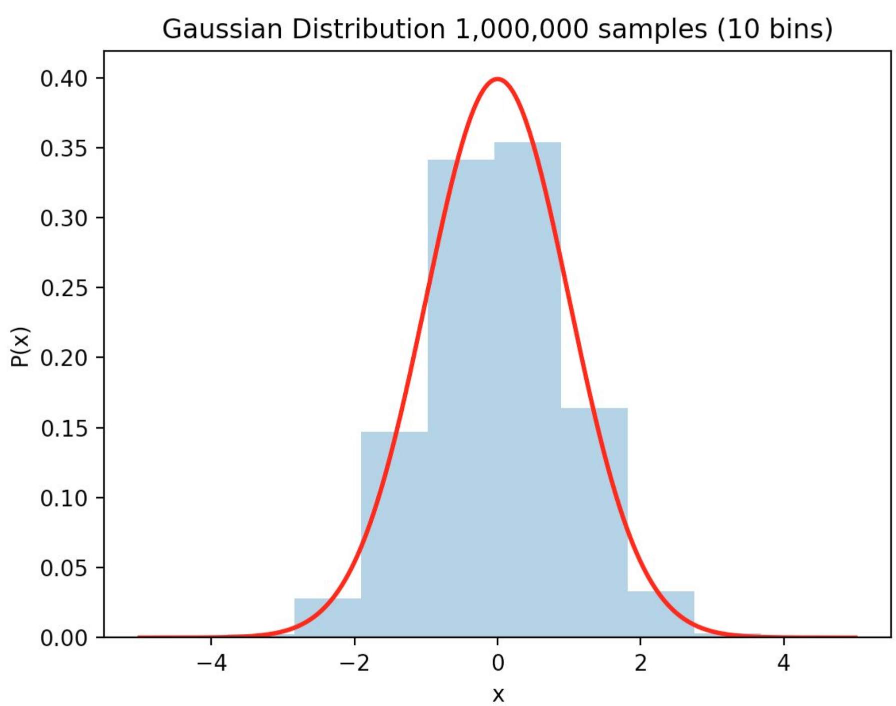
Gaussian Distribution 1,000 samples (20 bins)
P(x)
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
x



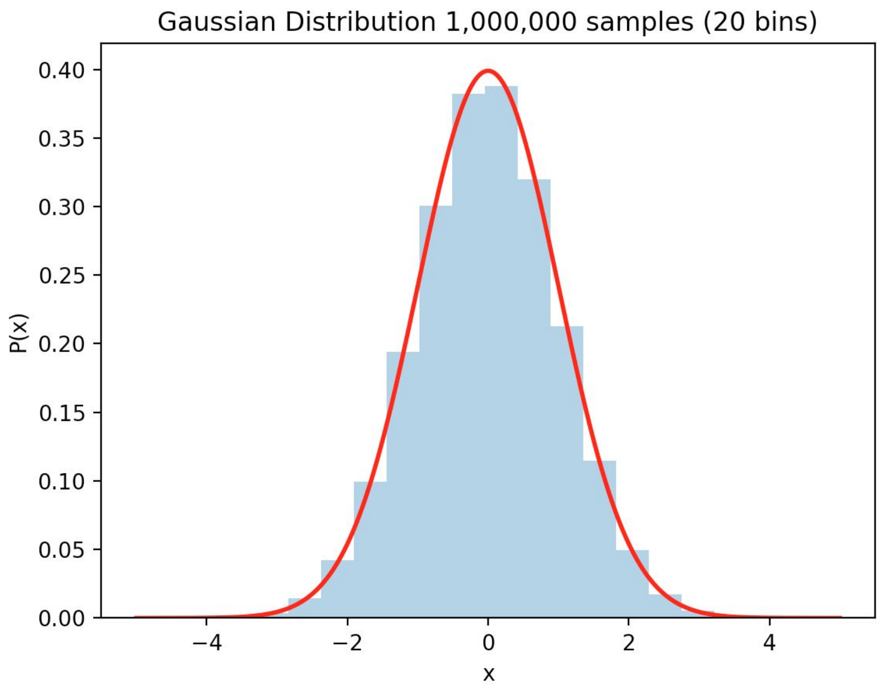
Gaussian Distribution 1,000 samples (50 bins)
P(x)
0.0
0.1
0.2
0.3
0.4
0.5
−4
−2
0
2
4
x



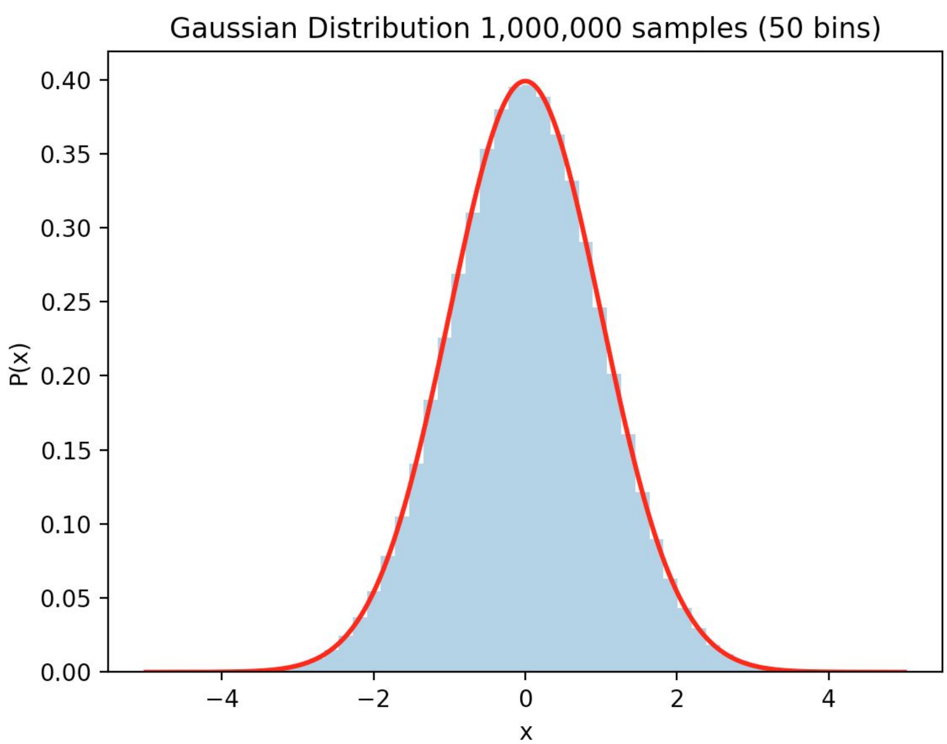
Gaussian Distribution 1,000 samples (100 bins)
P(x)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−4
−2
0
2
4
x



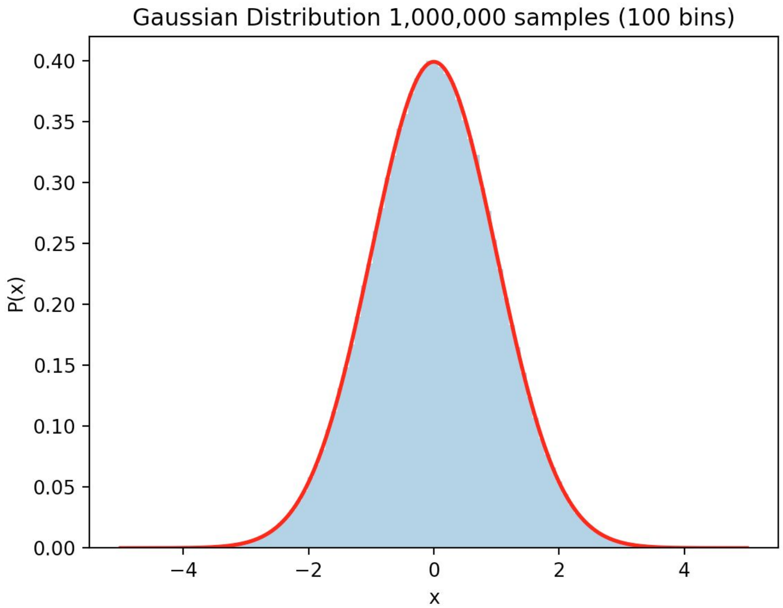
Gaussian Distribution 1,000,000 samples (10 bins)
P(x)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
−4
−2
0
2
4
x



Gaussian Distribution 1,000,000 samples (20 bins)
P(x)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
x

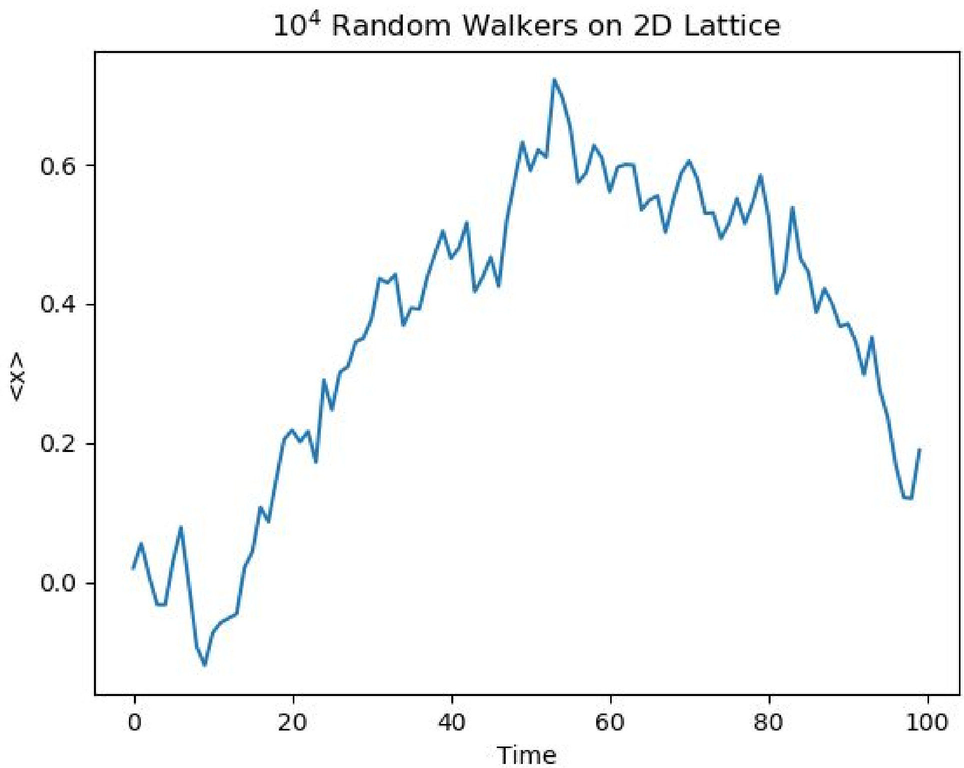


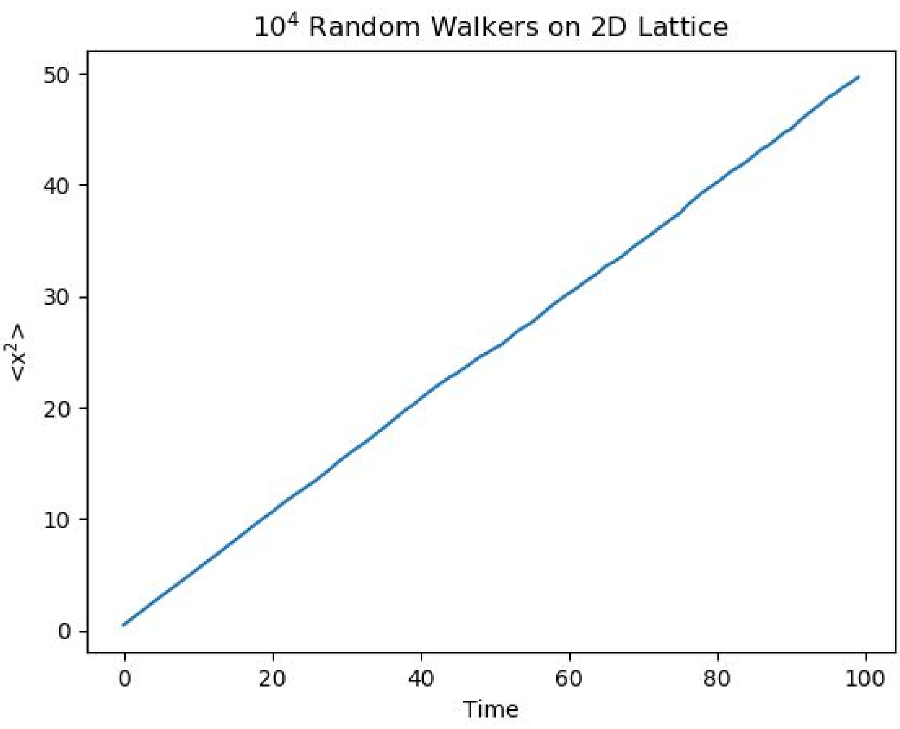
Gaussian Distribution 1,000,000 samples (50 bins)
P(x)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
x



Random Walk

a) The purpose of this problem was to write a program to simulate a 2-D random walker taking steps in $\pm x$ and $\pm y$ and then plot $\langle x \rangle$ and $\langle x^2 \rangle$ vs t (n-steps). To accomplish this we utilized random number generation with values given confined between 0 and 1 to represent the probability of the step going in one of the four directions. Values that fell below 0.25 decreased the x position by 1, values between 0.25 and 0.50 increased the x position by 1, values between 0.50 and 0.75 increased the y position by 1 and all other values decreased the y value by 1. For each step the x position and the square of the x position were divided by 10000 and stored in an array to be plotted.





b) The purpose of this problem was to write additional code to evaluate the mean square distance within the 2-D random walk and see if we could extract the diffusion constant and show the motion to be diffusive. To accomplish this code was added that calculates the radius at each step of the loop and an array that tracks the square of the radius divided by 10000. An additional plot was made show <r^2> vs. t, we can see the similar results and slope of the graph changes by a factor of two, consistent with 4Dt one would find in a two dimensional system versus the 2Dt of a one dimensional system. Thus in this case our diffusion constant D = 0.25.