Vinod Kancherla, Grant McNamara, Shreya Sutariya, Ben Zeman
PHY 6860: April 6, 2019

Group B: Project 1

1. a)

The first part of the problem is to generate arrays of N = 1,000 and 1,000,000 random numbers between [0, 1]. Then the probability distribution was plotted with a variety of bin numbers. The random numbers in the interval [0, 1] are generated using the command random.random(). To obtain an array of N values, a function is created which takes in the desired size of the array and sets up a while-loop to calculate N random values and place them in an array. The function then returns the array.

In order to graph the probability distribution, another function is created with the arguments being the array of random values,its size, and the number of bins. This function made use of the function plt.hist(). This function was called four times for each array size for bins = 10, 20, 50, and 100. Figures 1. d. and 2. d. are most illuminating as the uniform nature of the distribution is more prominent for N = 1,000,000. However, as the resolution increases via a higher number of bins, the slight uneven nature of the distribution is evident. In the case of N = 1000 values, the array is relatively small and the distribution does not exhibit uniform behavior.

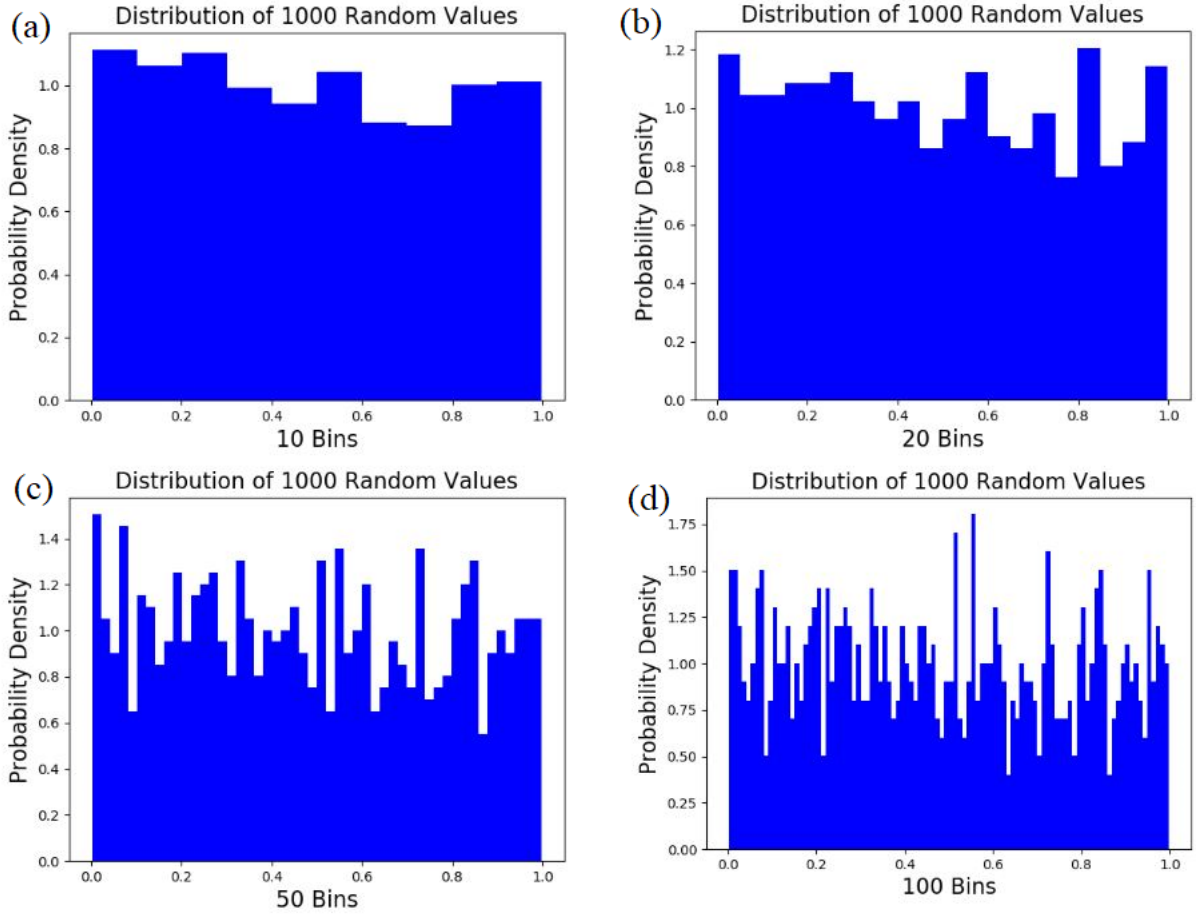


Figure 1: 1,000 random values sorted into an increasing number of bins.

Vinod Kancherla, Grant McNamara, Shreya Sutariya, Ben Zeman
PHY 6860: April 6, 2019

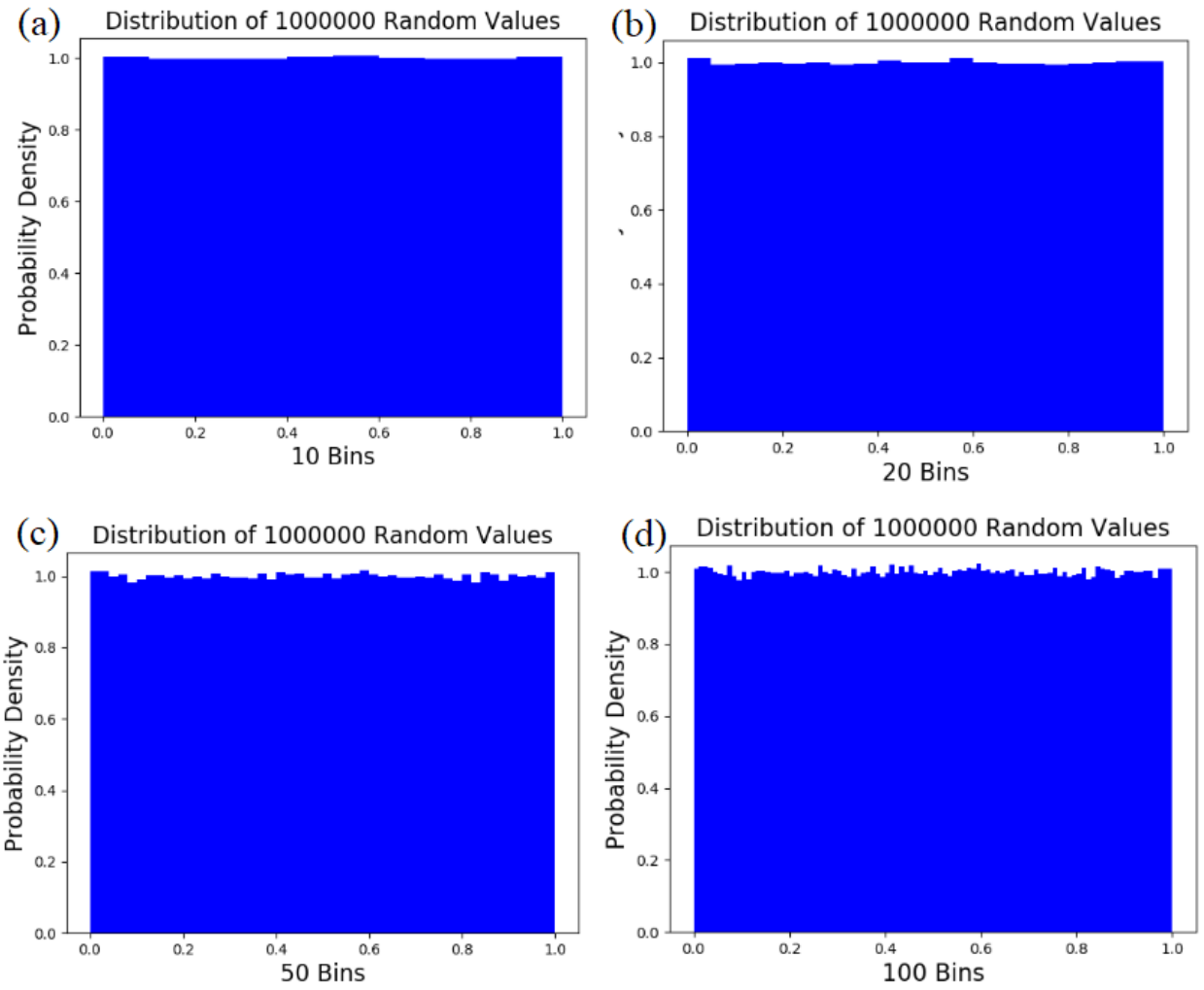


Figure 2: 1,000,000 randomly generated values sorted into an increasing number of bins.

1. b)

The second portion consists of creating an array of random values that are distributed following a Gaussian curve. Again, two arrays of sizes N = 1,000 and N = 1,000,000 are computed. The values are then plotted in a histogram to clearly see the Gaussian distribution of the array.

The array of random values distributed according to a Gaussian curve were calculated using the Rejection Method, also known as the hit-or-miss method. This was achieved using a function which took as an argument the size of the desired array. Then, a y-value was calculated in the interval [0, f_max], where f_max is the amplitude of the Gaussian function that was given. This was done using a simple if-statement which rejected random values generated out of the interval. Then a random x-value is created using the same method as in part (a).

The final step involves using this x-value in a function which calculates P(x). This value is then compared to the y-value in [0, f_max]. If $P(x) \geq y$, then this x-value is added to the

random Gaussian values array. Figures 3 and 4 show the resulting normed histograms. The given Gaussian distribution is shown with a dashed black line.

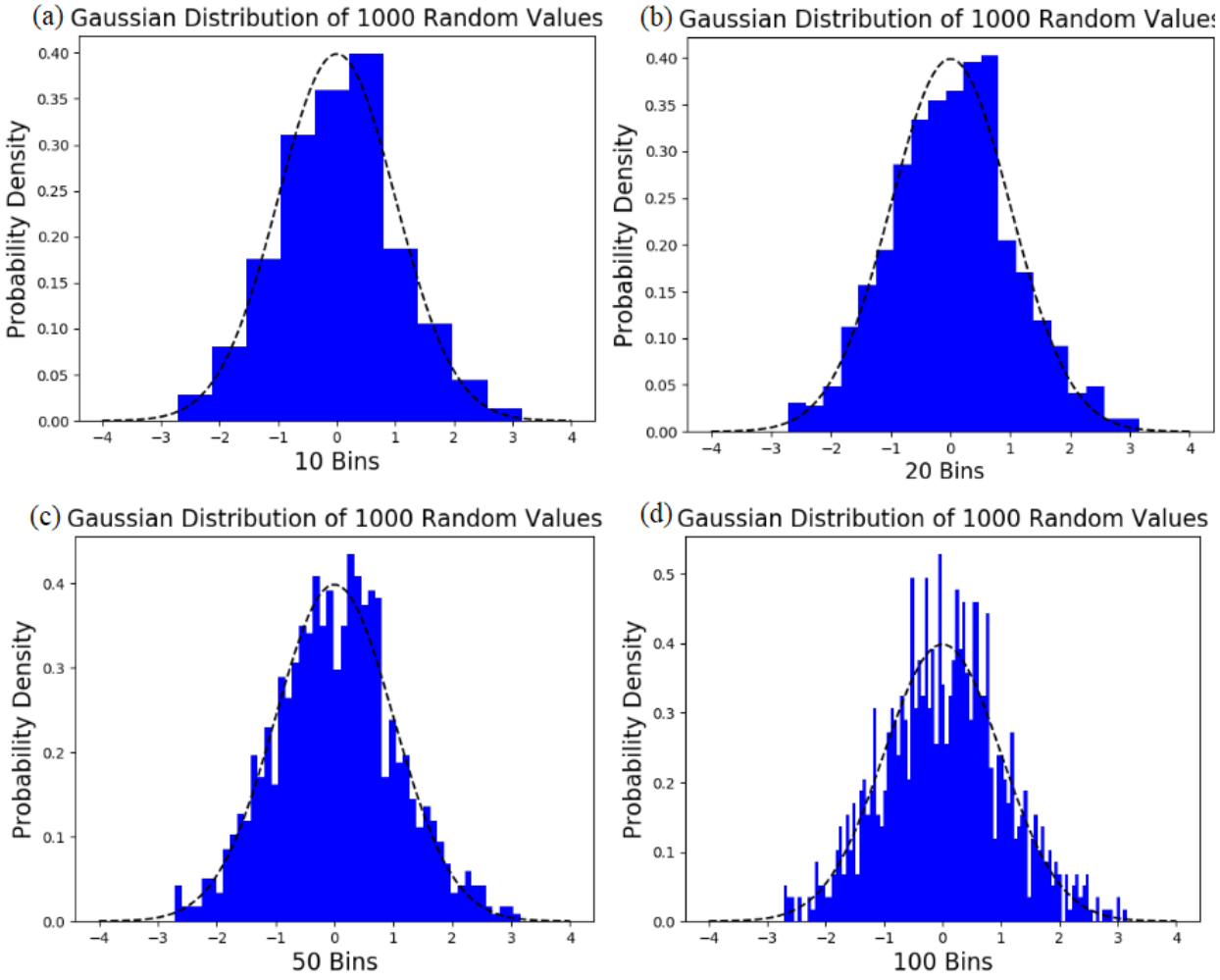


Figure 3: Histograms for 1,000 random Gaussian values. The dashed Gaussian is the original function that was given.

Vinod Kancherla, Grant McNamara, Shreya Sutariya, Ben Zeman
PHY 6860: April 6, 2019

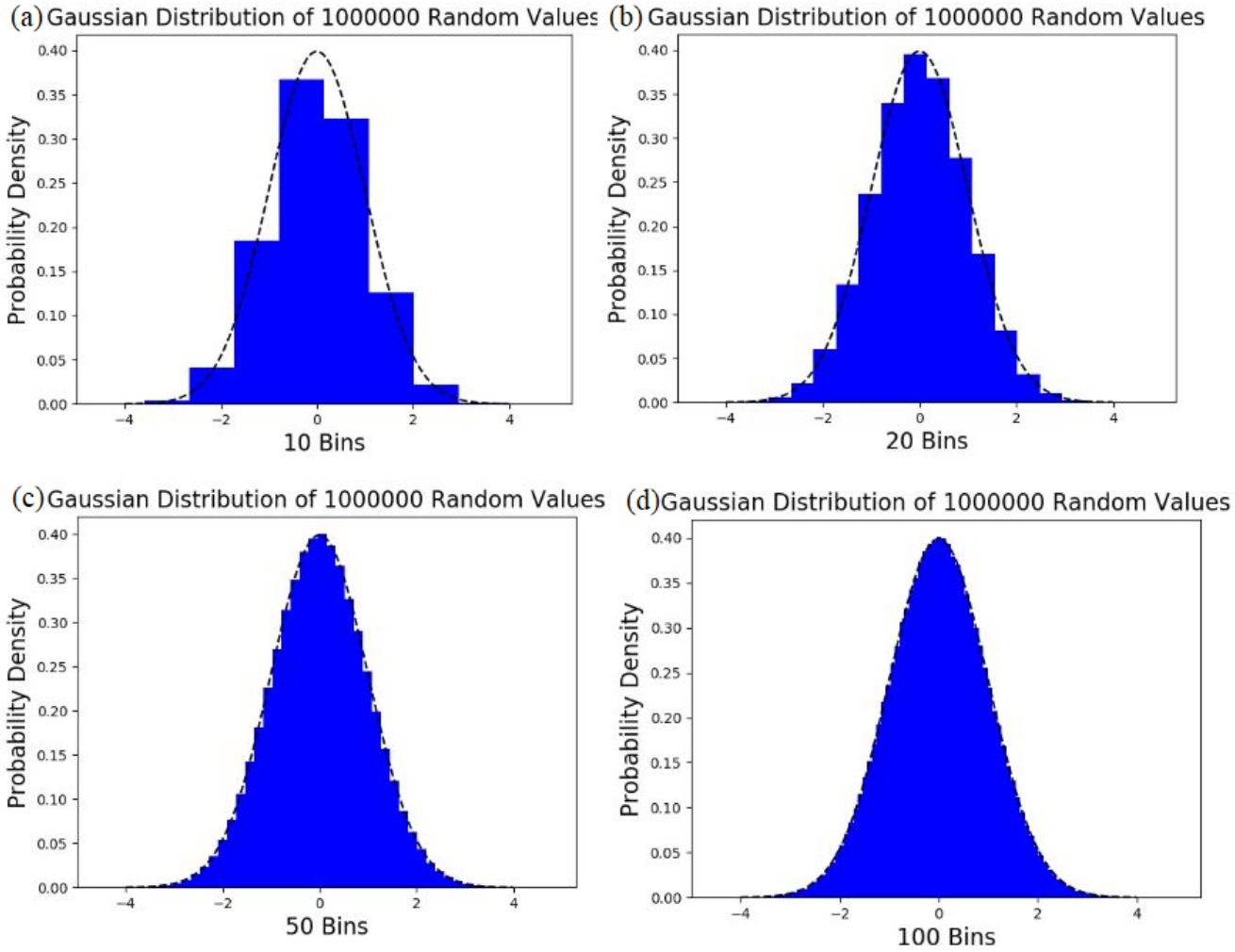


Figure 4: Histograms for 1,000,000 random Gaussian numbers.

2. a). Using the symmetry of the Gaussian function, and the fact that

$$\int_0^\infty x^2 e^{-x^2/\sigma^2} = 2\sqrt{\pi}\left(\frac{\sigma}{2}\right)^3 = (1/4)\sqrt{\pi}\sigma^3,$$

we can find $\int_{-\infty}^{\infty} x^2 e^{-x^2/2\sigma^2} = 2\int_0^\infty x^2 e^{-x^2/2\sigma^2} = \sqrt{2\pi}\sigma^3$

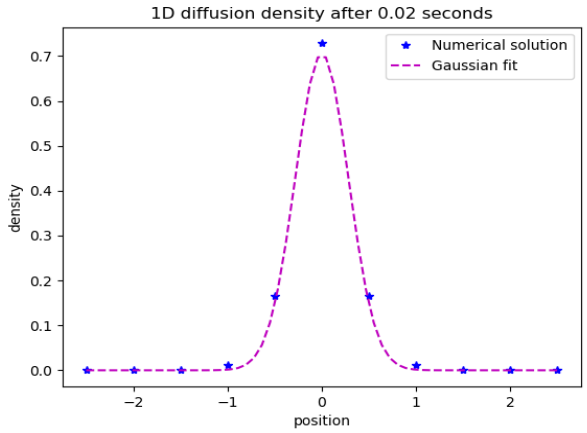
and therefore, $\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} x^2 e^{-x^2/2\sigma^2} = \sqrt{2\pi}\sigma^3/(\sqrt{2\pi}\sigma) = \sigma^2$.

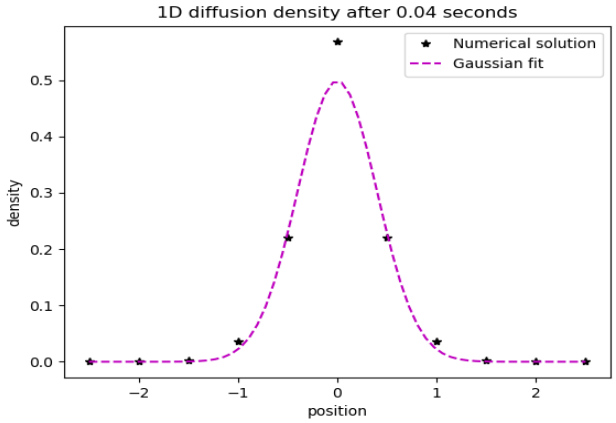
b) For the numerical solution, we used a relaxation method to find the densities at each 1-dimensional position for five-time snapshots. This accomplished with the following steps:

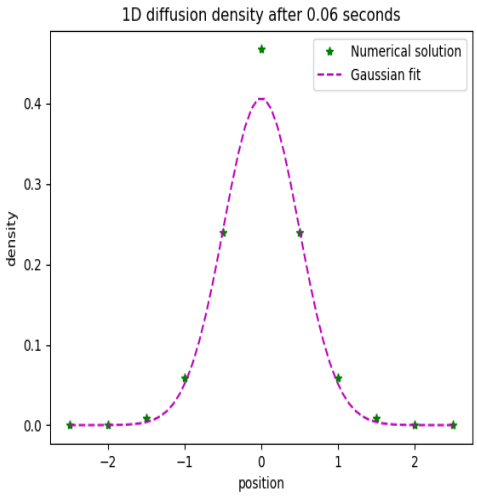
- Initialized variables for relaxation
- Initialized arrays for plotting position vs. density

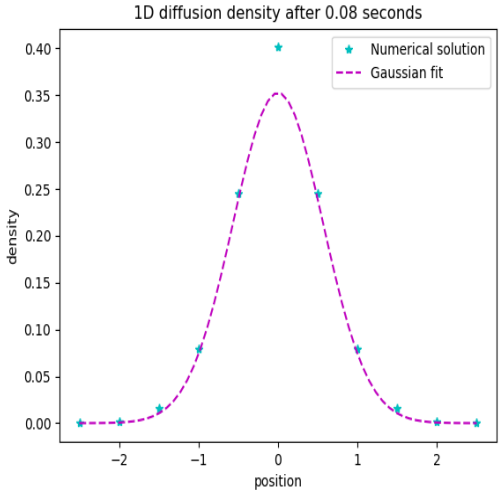
Vinod Kancherla, Grant McNamara, Shreya Sutariya, Ben Zeman
PHY 6860: April 6, 2019

- For time t = 0, we used a Gaussian function and solved it for t = 0.01, which was also our Δt value
- Then, we made for-loops for 11 time-steps and 11 positions.
  - Calculated densities with respect to densities at previous locations at the last time-step and a diffusion constant D=2
- For 5 time snapshots, appended arrays for density profile
  - For each loop iteration at that time, appended position and density arrays specific to that time
- Plotted each specific position array with respect to its corresponding density array
- Plotted Gaussian density profiles for each time snapshot on the same plot as its corresponding numerical density profile

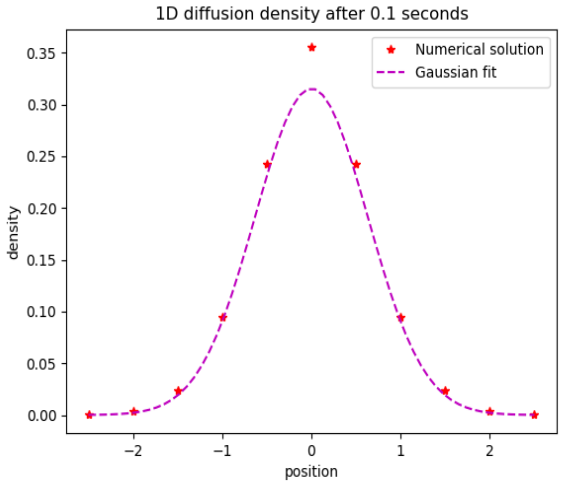








Vinod Kancherla, Grant McNamara, Shreya Sutariya, Ben Zeman
PHY 6860: April 6, 2019



- These plots all show that the numerical distribution is similar to the Gaussian distribution.
- The densities for x = 0 for each numerical distribution is a little off because we used an initial condition which differed from our relaxation method.