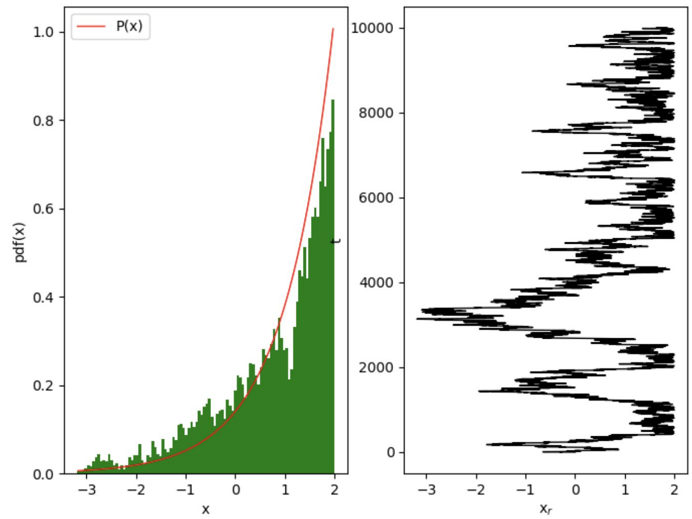
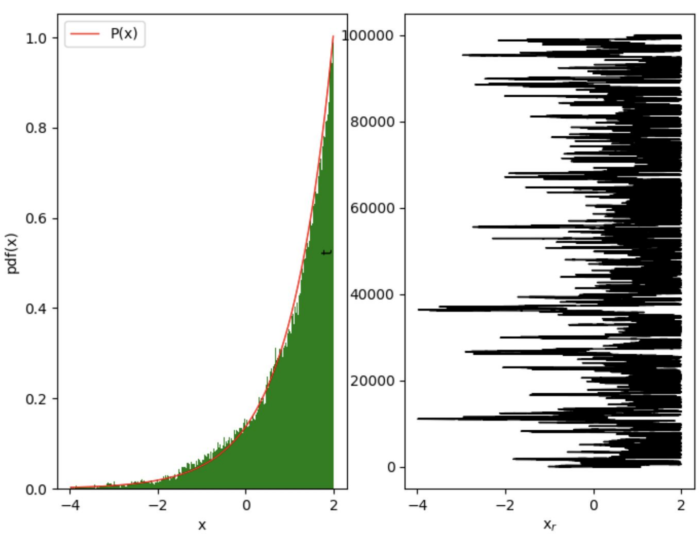
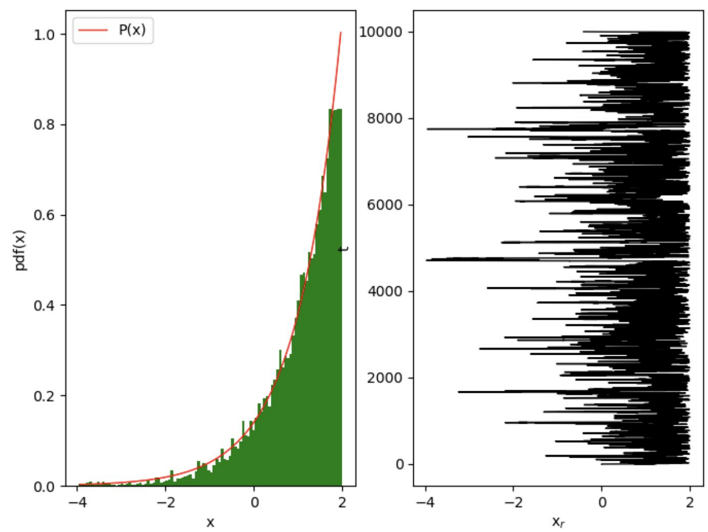
Part 1



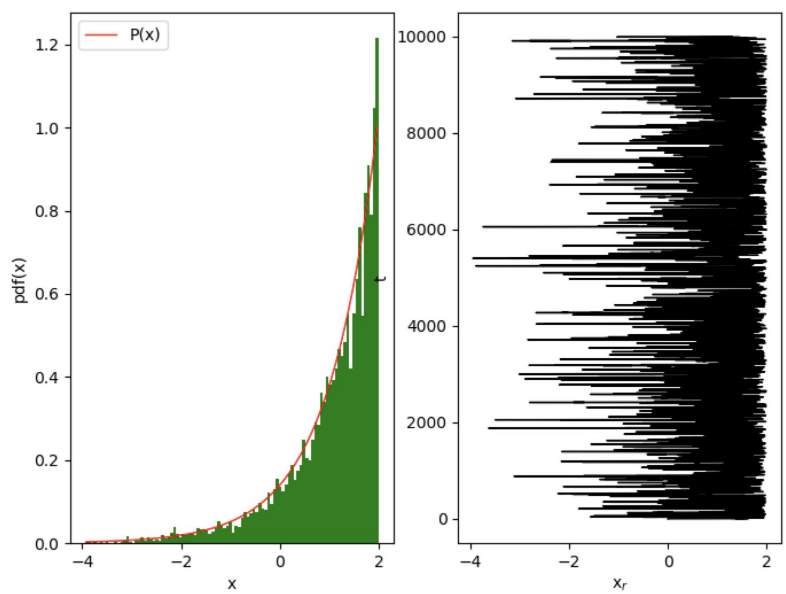
1. Delta = 0.1, R = 10^4: The histogram seems to follow the expected distribution line pretty well. Since the MH algorithm is probabilistic, it will not follow the line perfectly. However, by increasing R we can greatly improve the quality of our analytical solution since we have more points allowing us to explore the entire domain more evenly (Delta = 0.1, R = 10^5):



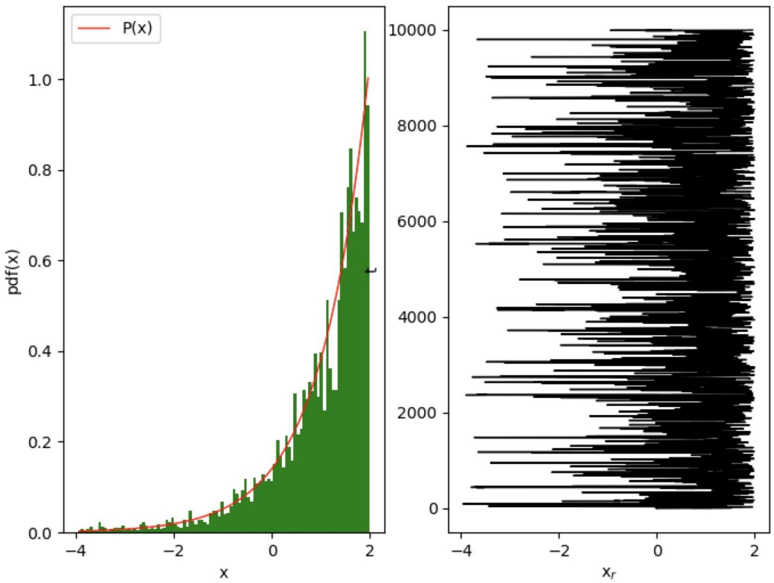
2.



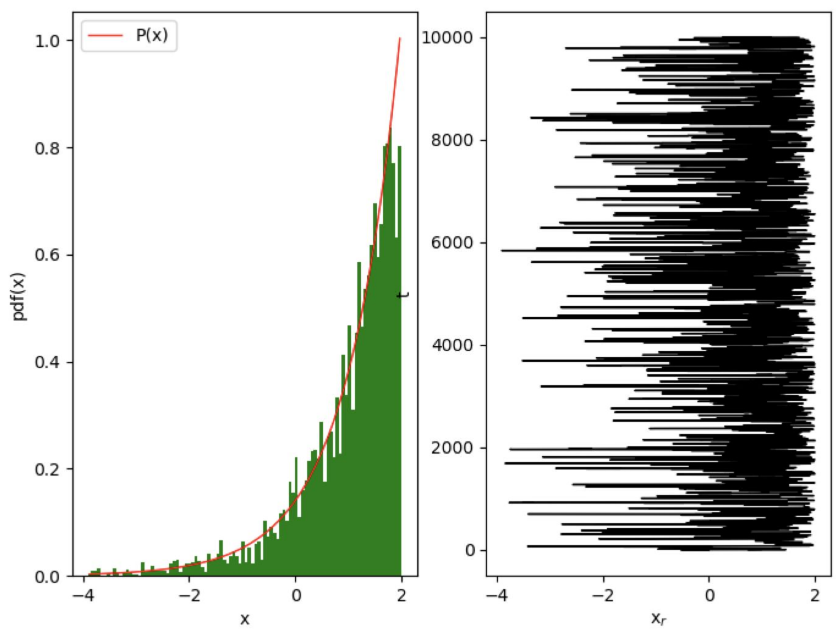
a. Delta = 0.5



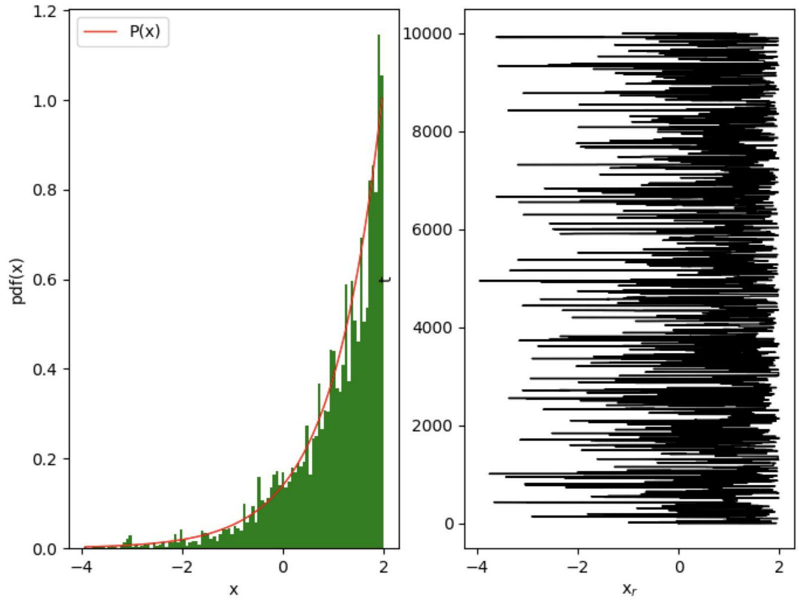
b. Delta = 1.0



c. Delta = 2.0



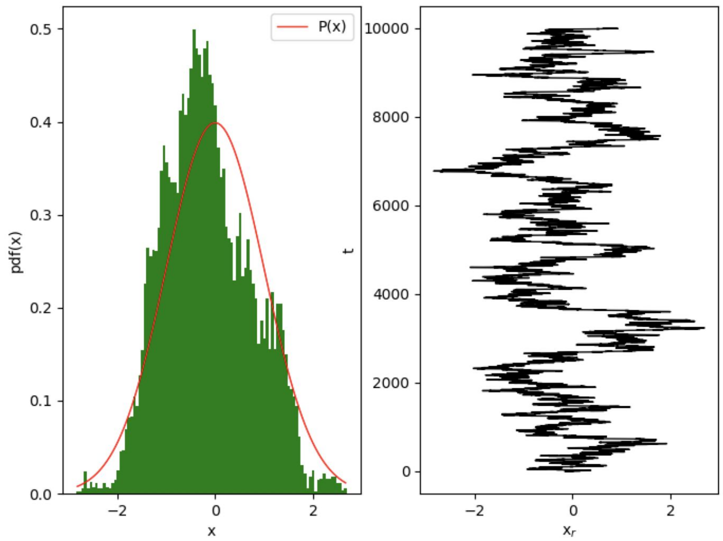
d. Delta = 3.0



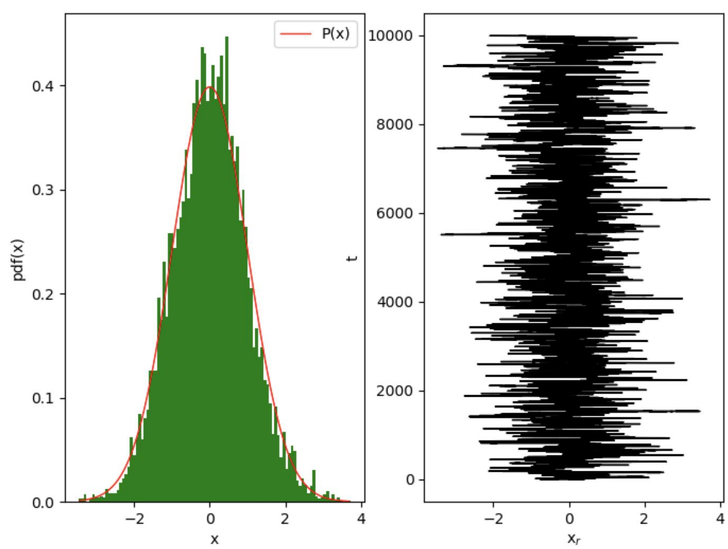
e. Delta = 4.0

f. The right plot varies more dramatically as delta increases. This is because we are taking bigger steps between samples causing us to have a higher variance in the right plot. When delta is smaller, we get stuck exploring certain sections of the domain at a time since we cannot take big steps and easily explore areas far from the current value.

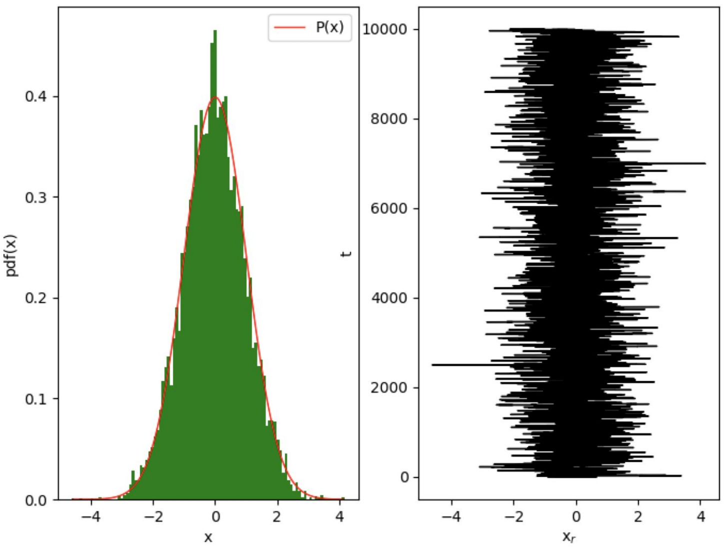
3. Delta = 1.0 seems to be the best delta for the exponential distribution since it is the smallest delta that does not exhibit the symptoms of a delta value that is too small (getting stuck exploring the same section). (It is possible to see to values get stuck in the same section when Delta <1.) However, it is important to note that if we allow R to be larger, then we are able to lower our delta (for example the pictures provided in section 1).
4.
   a. normal



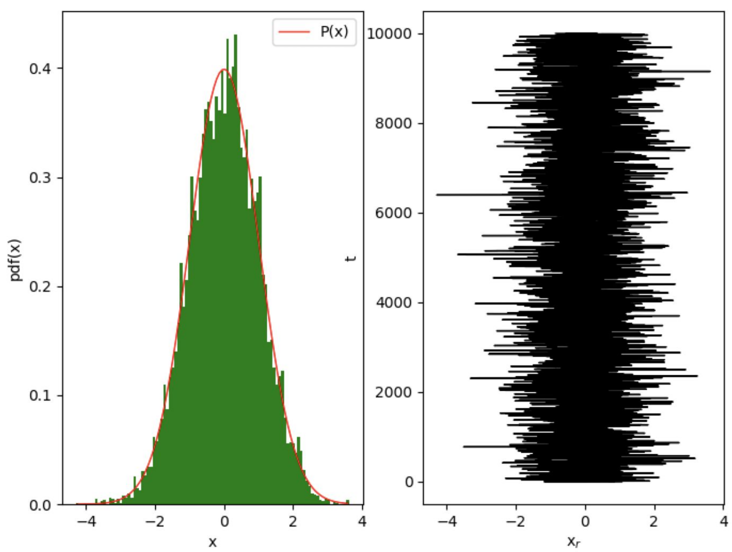
i. Delta = 0.1



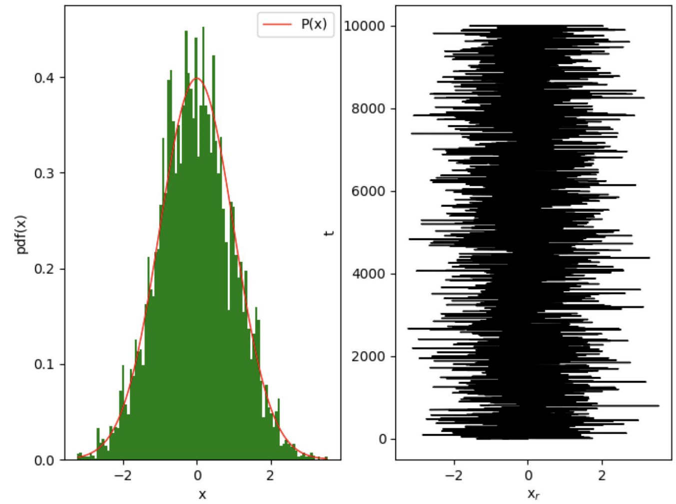
ii. Delta = 0.5



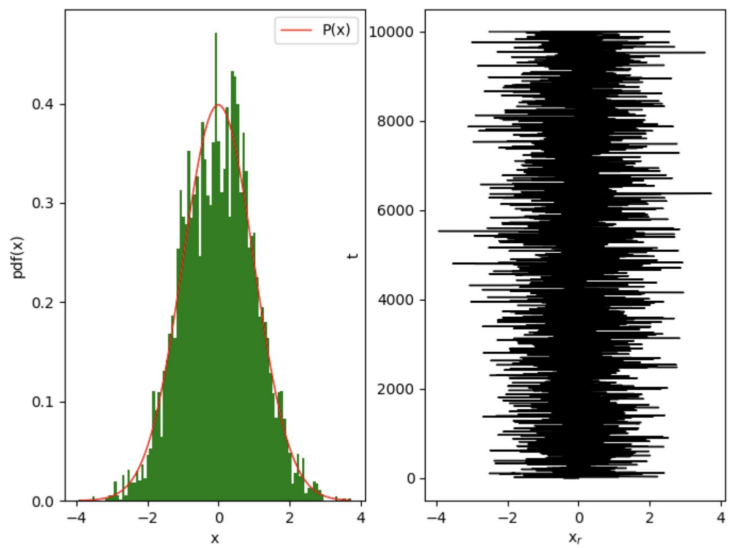
iii. Delta = 1.0



iv. Delta = 2.0



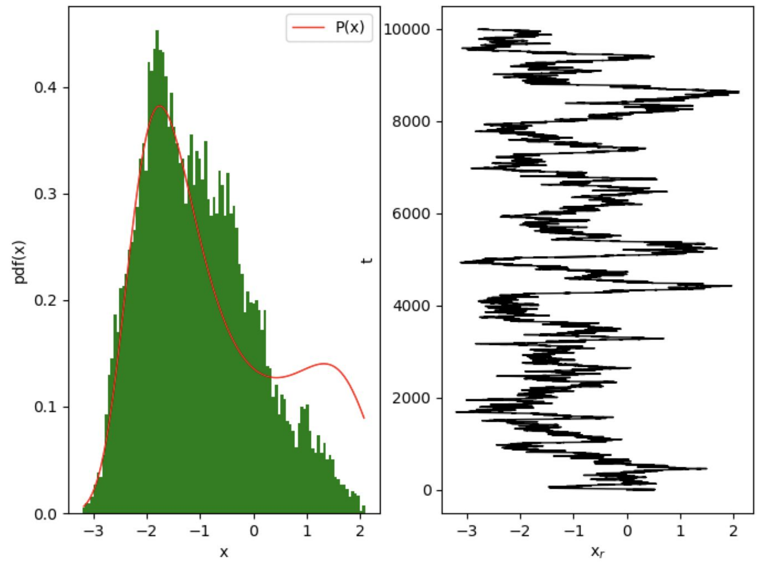
v. Delta = 3.0



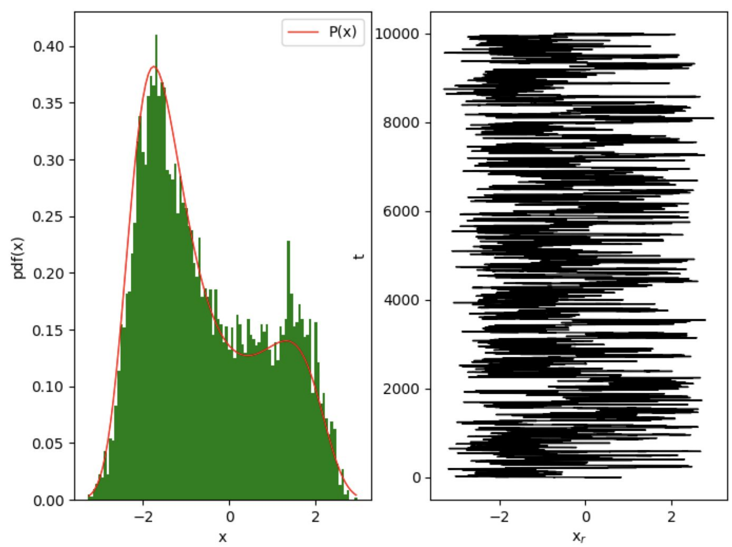
vi. Delta = 4.0

vii. The best Delta value for the normal distribution seems to be Delta = 2.0. This is because when Delta < 2.0, then plots exhibit symptoms of having a Delta too small (explained earlier in section 2f and 3). When Delta > 2, the distributions seem to be worse at accurately recreating the PDF. This is probably because the Delta is too large to accurately show the details of the PDF. When the Delta is too large it is difficult to accurately follow the analytical solutions of PDF’s that have large derivatives.

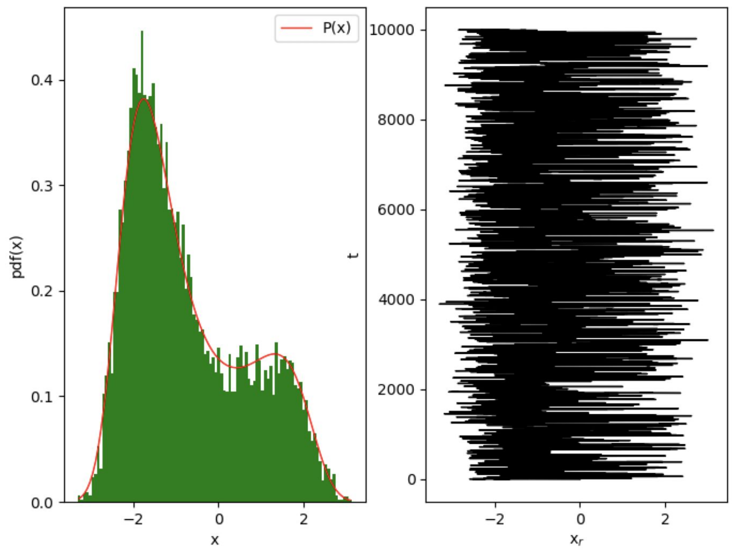
b. crazy



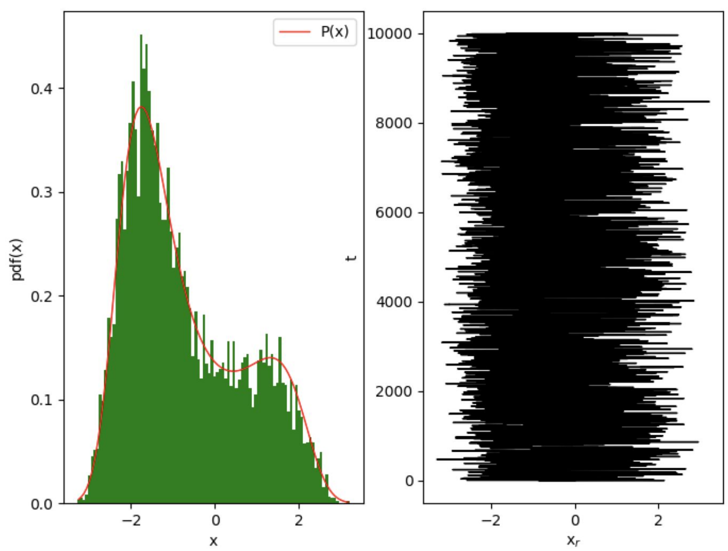
i. Delta = 0.1



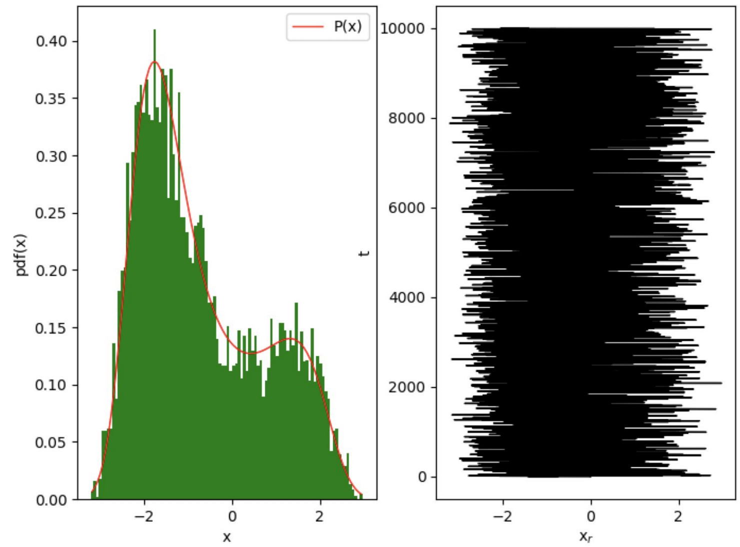
ii. Delta = 0.5



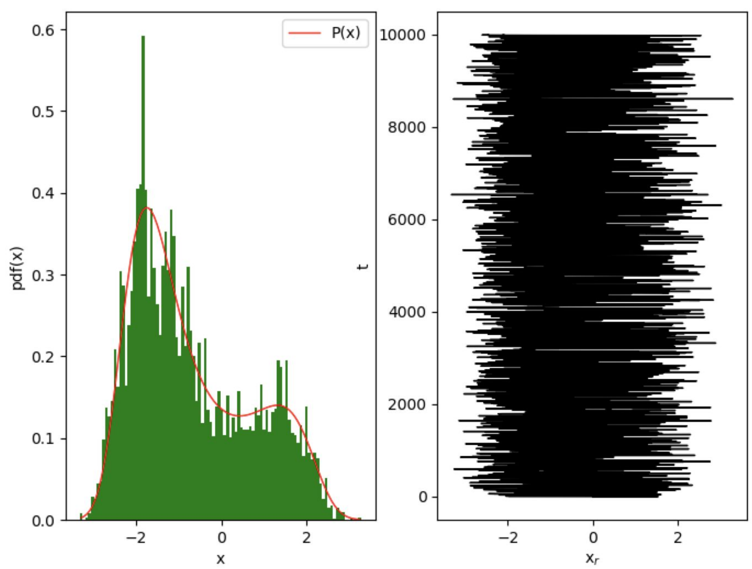
iii. Delta = 1.0



iv. Delta = 2.0



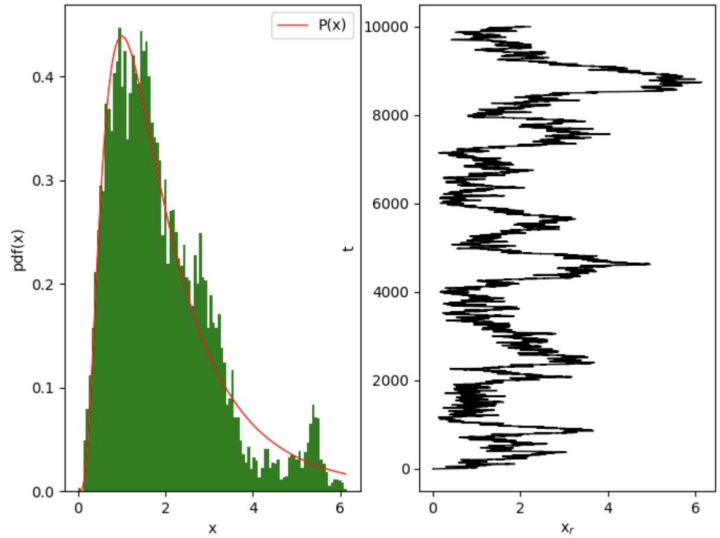
v. Delta = 3.0



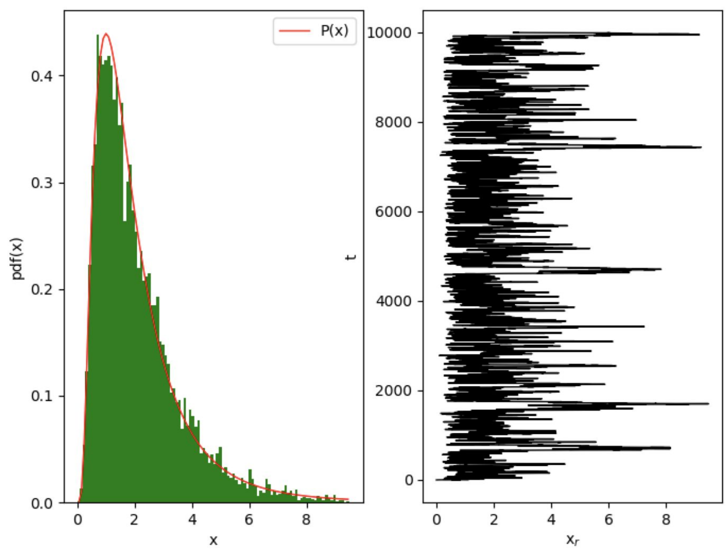
vi. Delta = 4.0

vii. The best Delta for the crazy functions seems to be Delta = 1.0. This is for the exact same reasons given for the normal function in section 4 a vii.

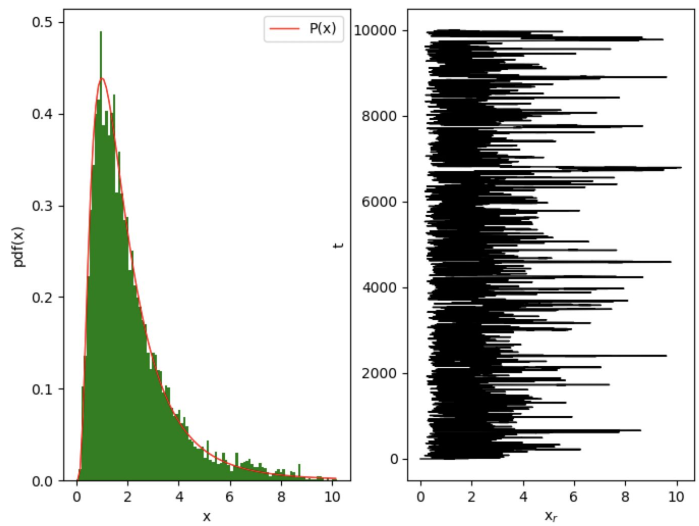
c. lognormal



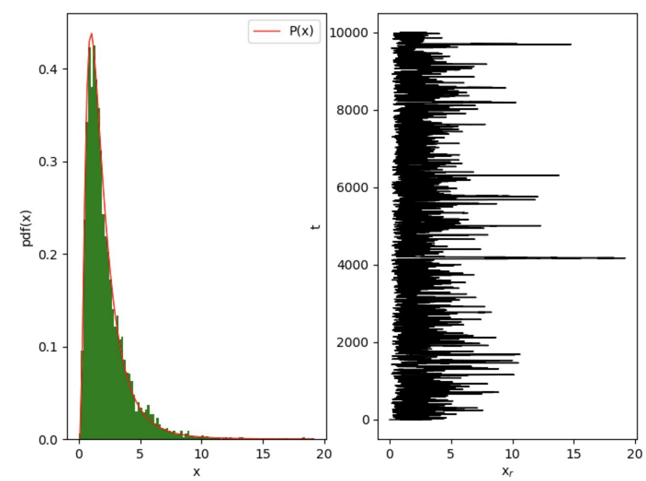
i. Delta = 0.1



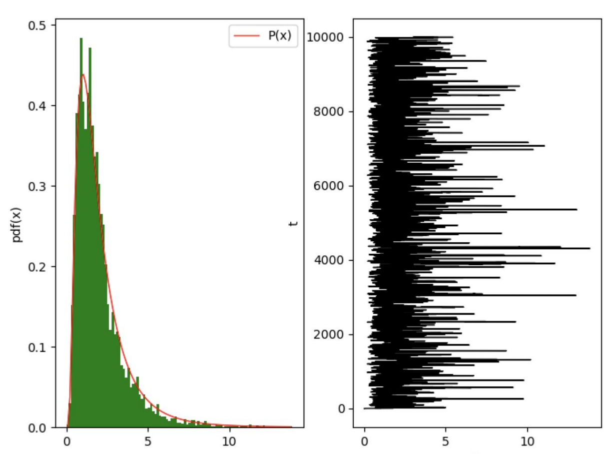
ii. Delta = 0.5



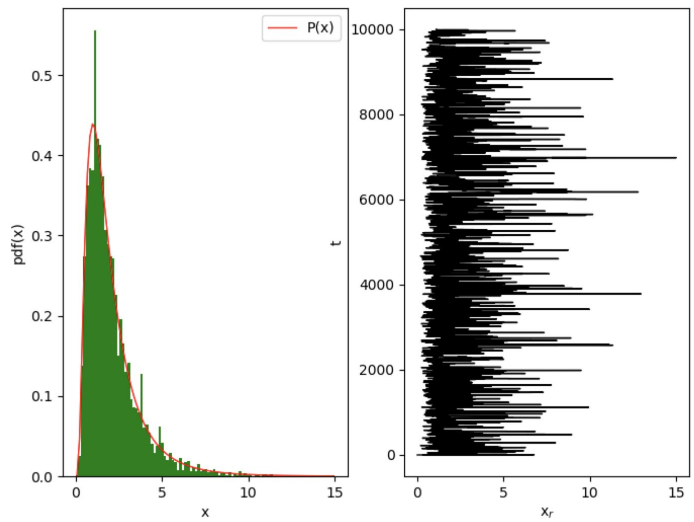
iii. Delta = 1.0



iv. Delta = 2.0



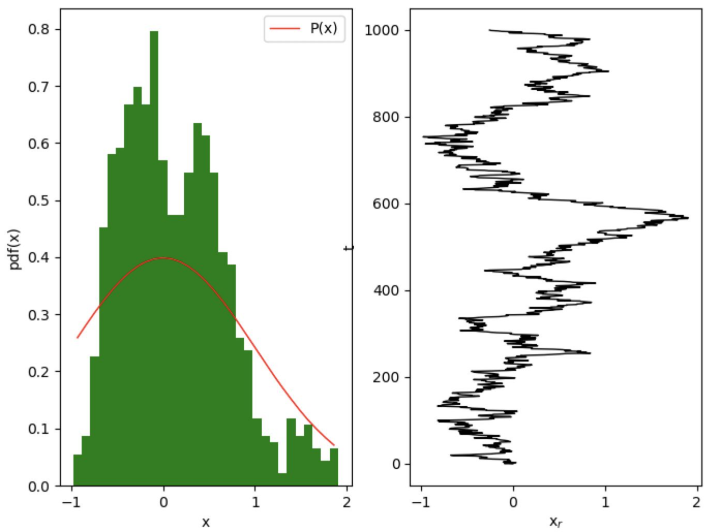
v. Delta = 3.0



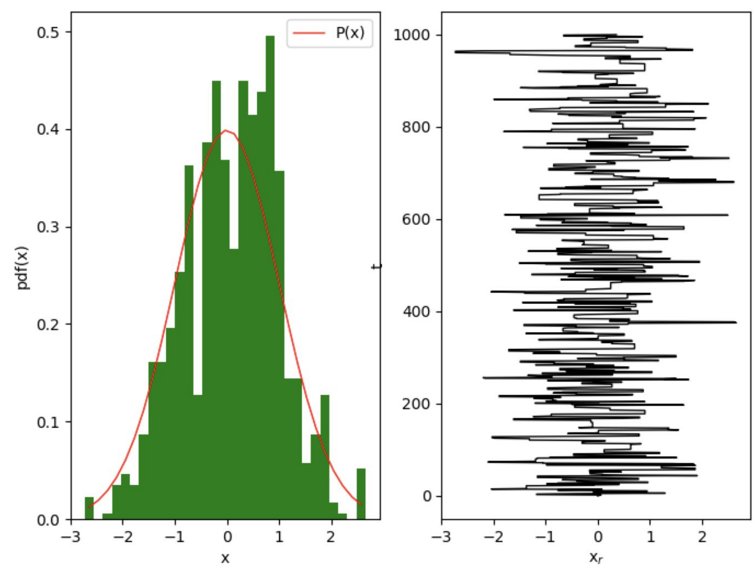
vi. Delta = 4.0

vii. The best Delta value for the lognormal distribution seems to be when Delta = 2.0. However, as Delta increases much larger than 1, it does not seem to have the same negative effect that a large delta had in previous examples. This seems to be due to properties of the lognormal distribution. This causes the right plots to have a relatively similar shape as long as Delta > 1.

5. Delta is the step size, so it determines how the level of detail we are able to obtain. However, if Delta is too small, it will be unable to effectively explore the entire domain, unless we also significantly increase R. Thus, as delta decreases (so we can obtain more detail) R must increase to allow us to effectively sample the domain. An example of this is given in section 1, where Delta = 0.1 is too small, so it is not very accurate, but by increasing R it is able to become much more accurate.
When R is small, it is useful to increase Delta so that we are able to explore the entire domain (since we are taking fewer steps, increasing the step size will help us explore the entire domain). An example of this is R=1000, Delta = 0.1:

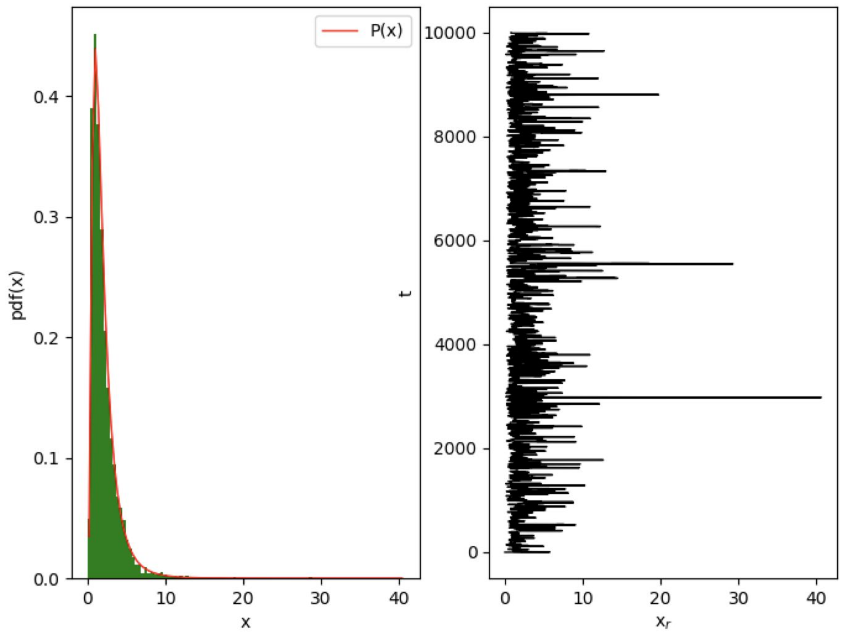


does a very bad job of exploring the domain while (R=1000, Delta = 3):

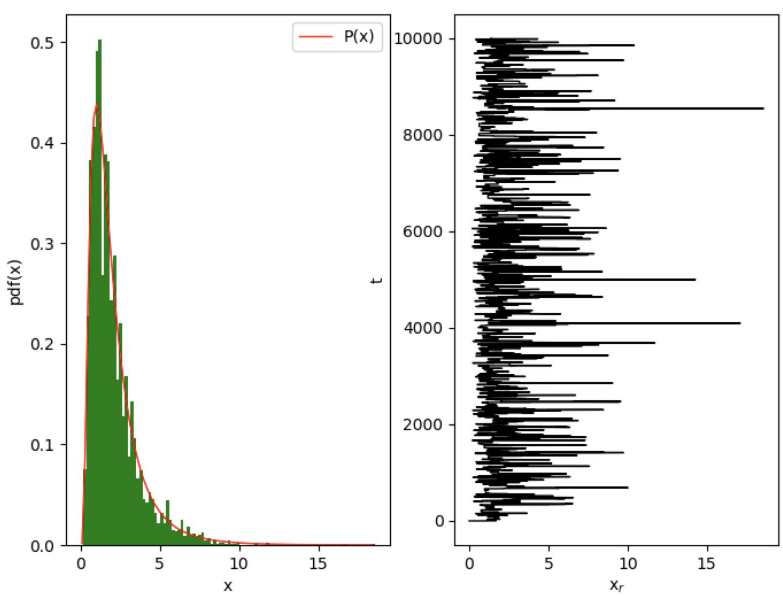


does a much better job. R directly correlates with compute time as it determines for how long the program should run before it terminates.

6. The lognormal distribution seems to have the least constraints on Delta. When Delta < 1, it runs into the problems we expect from a small delta (over accepting from certain sections of the domain). However, when Delta > 1, it does not seem to suffer from problems we expect form large Deltas. For example when R=10000, Delta = 6:



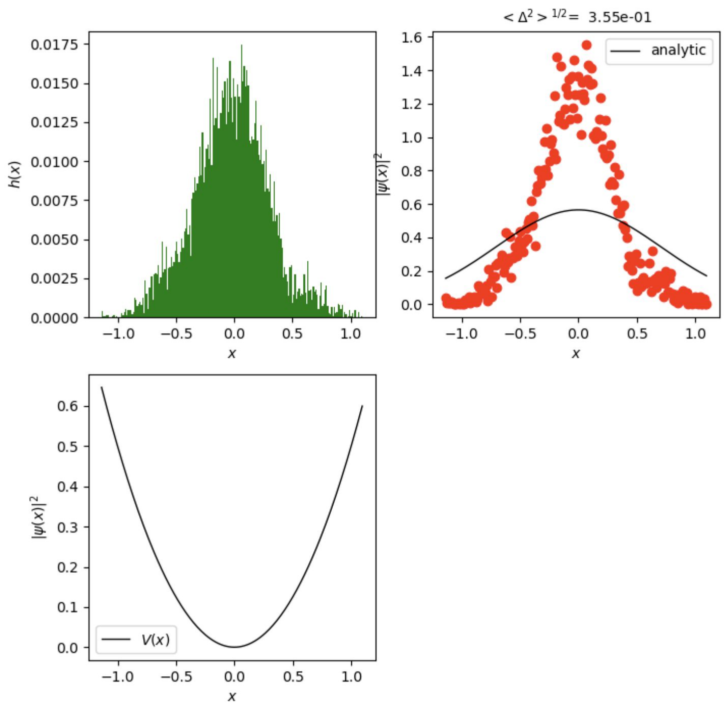
In this example the very high step size creates the possibility for very large x values to be accepted (such as 40 in this case). This creates a situation in which the mean value can be heavily skewed by a few large x values being accepted. This makes it difficult to calculate the mean of the distribution. For example in another run of R = 10000, Delta = 6:



This graph had the exact same parameters, but accepted much fewer large x values.

Part 2

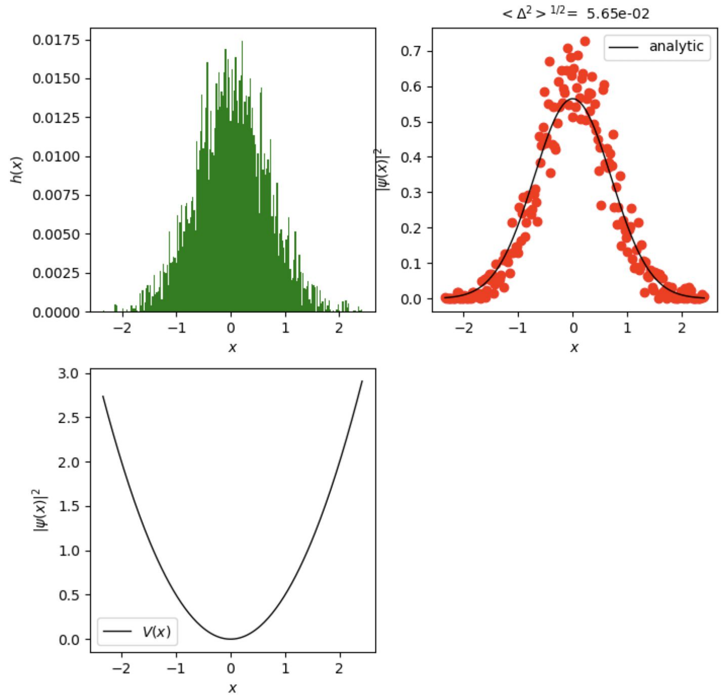
1. Harmonic (N=100, T=10000, Delta = 0.1)



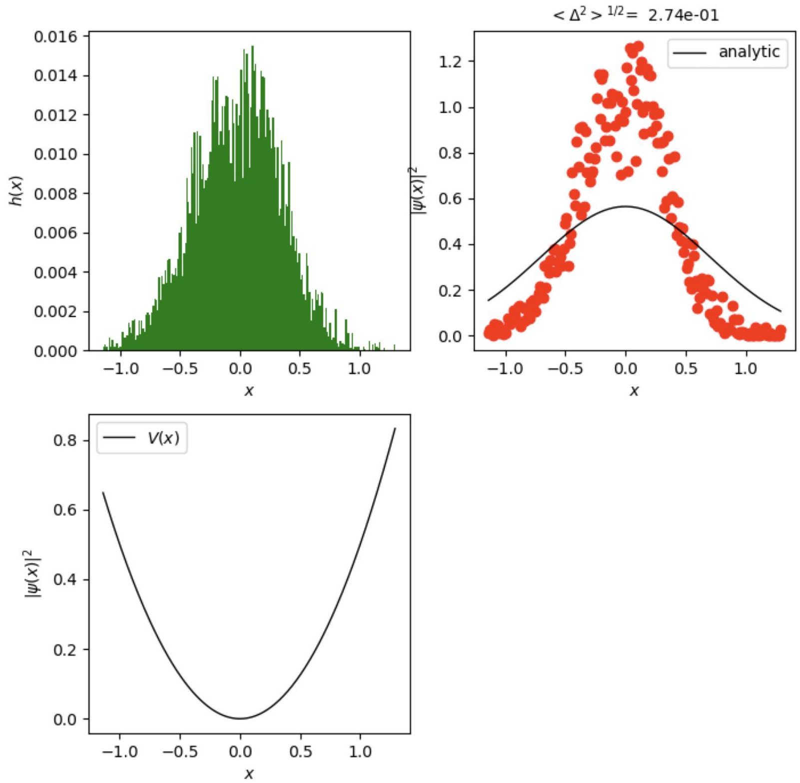
   a. The first plot is a histogram of our accepted points.
   b. The second plot is the analytic solution of the probability of finding a particle at a given x value versus the numerical probability of finding a particle at a given x value. The root mean squared is displayed above this graph.
   c. The third plot is the potential energy at a given x.

2.

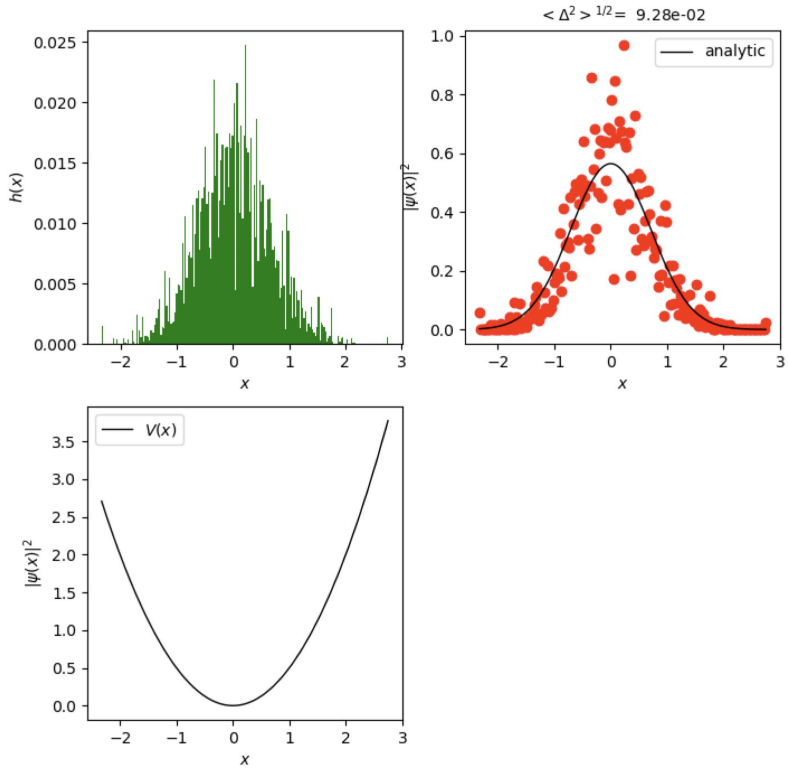
a. Delta = 1 is the most suitable Delta.



When the delta value is at its best, the numerical solution is very similar to the analytic solution and the variance (and root mean squared error) is minimized.

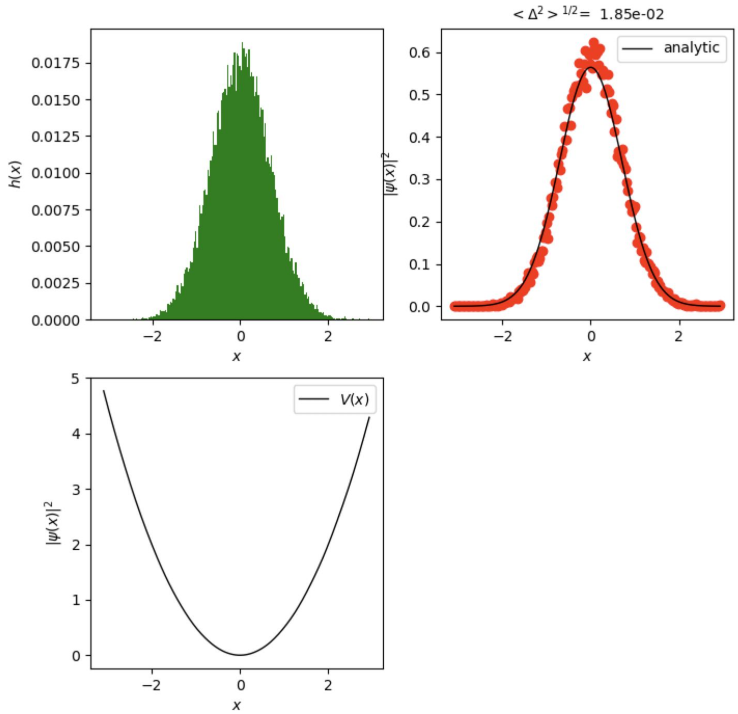


b. Delta = 0.1
When delta is too small, we are unable to explore the entire domain well, so our histogram has a much larger peak around the center than it should.



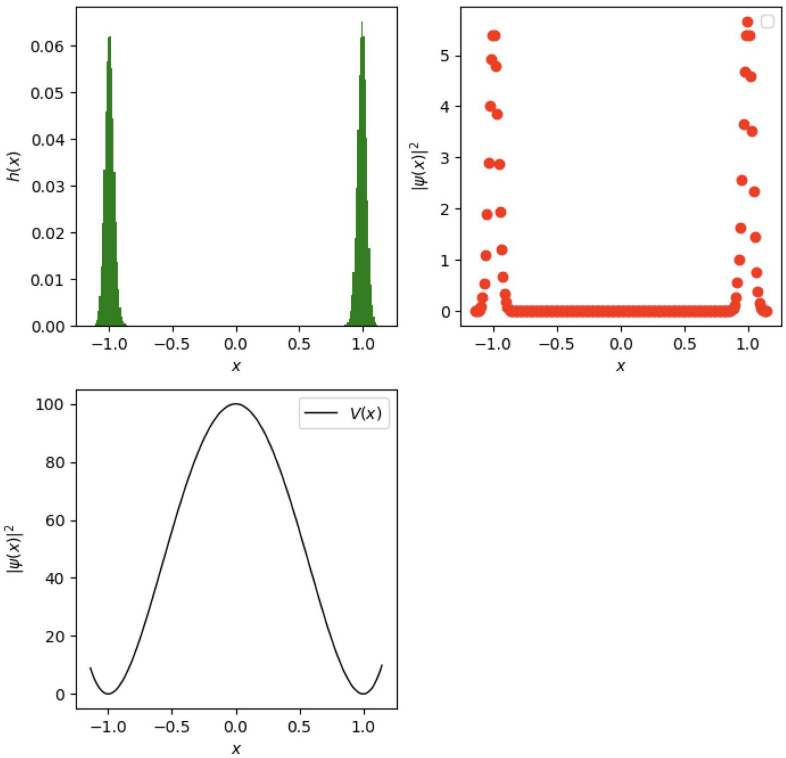
c. Delta = 3 When delta is too large there the Path Integral Monte Carlo is unable to pick up the details of the solution (wave function squared). The numerical solution has a larger variance from the analytic solution (causing a larger root mean squared error). It's harder to connect the dots on the numerical solution to create the analytical solution. (The gaps between x values is larger)

3. T = 100,000

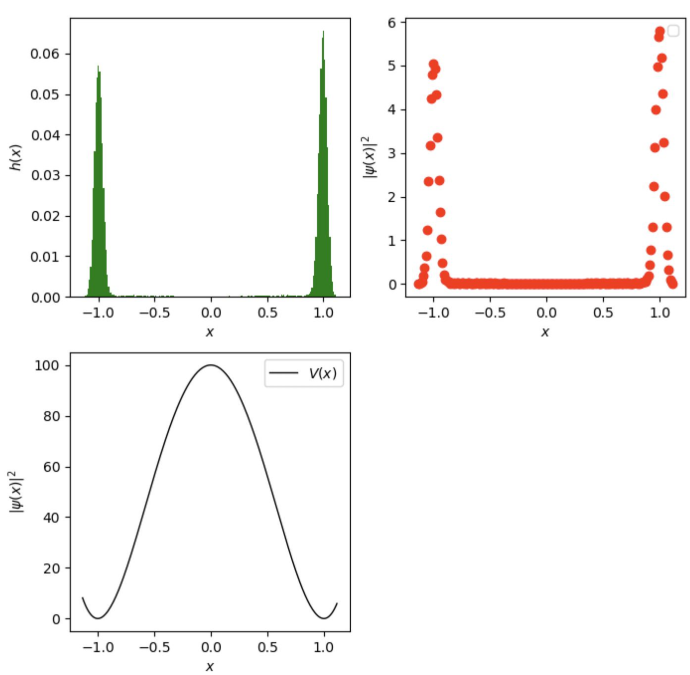


a. The numerical solution seems to do an extremely good job describing the analytical solution. As you increase T, the computation time increases, but the accuracy of the method also increases. This is because there are now more points, and the closer T goes to infinite, the closer we will get to the analytical solution. (The relationship between T and Delta is the same as the relationship between R and Delta from the MH algorithm, so choosing a good Delta value can help this happen quicker, similar to how choosing the correct Delta in the MH algorithm can help.)

4. Instantom
    a. The algorithm can produce quantum tunneling since we allow there to be a probability of a particle existing at an x value where the particle's E is less than the potential at that point. In classical mechanics there is no probability that an object can reach a point in which the potential energy is larger than the object's energy, but this is possible in quantum mechanics. We implemented this by randomly accepting values that should be rejected.
5. Different heights
    a. It seems as though two peaks are at different heights since the method does not have enough time to explore both points equally when T = 100000. If I increase T to 1000000 and keep delta at 0.1 my graphs look like:



while previously they looked like (T=100000, Delta = 0.1):



Since V(x) is

symmetric, I would expect the analytic solution of the probability of the particle at any x value to be symmetrical as well.

6. Box
   a. We know that the wave function must be normalized such that the area under the wave function squared is 1 (since the wave function squared is a PDF, the area under a PDF curve must equal 1). Since the potential (V(x)) is a vertical line, we can infer that it is infinite at the boundaries of the box forcing the probability of finding the particle outside of the box to become 0. This is exactly what we see when we see the solution for the box.