Phys 512 Assignment 7

Ian Hendricksen

November 19, 2021

1. Figure 1 depicts a 3D plot of C standard library random numbers from `rand_points.txt`, and Figure 2 depicts a 3D plot of NumPy random numbers of the same size as that of `rand_points.txt`.

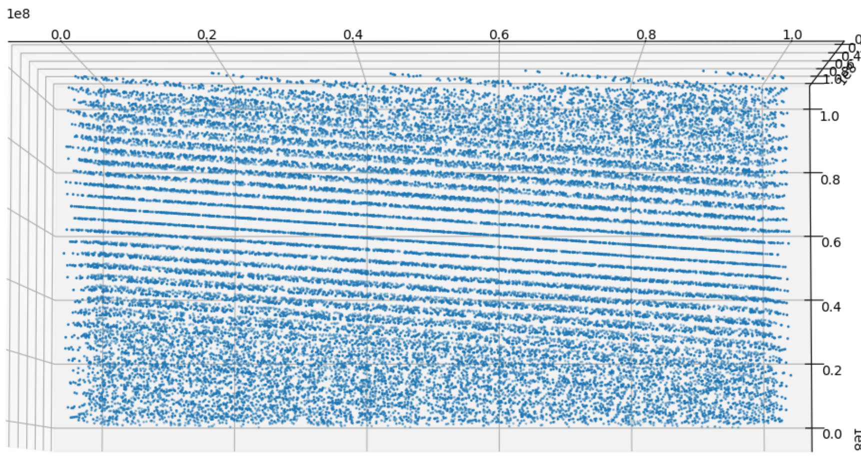


Figure 1: Random numbers generated from the C standard library.

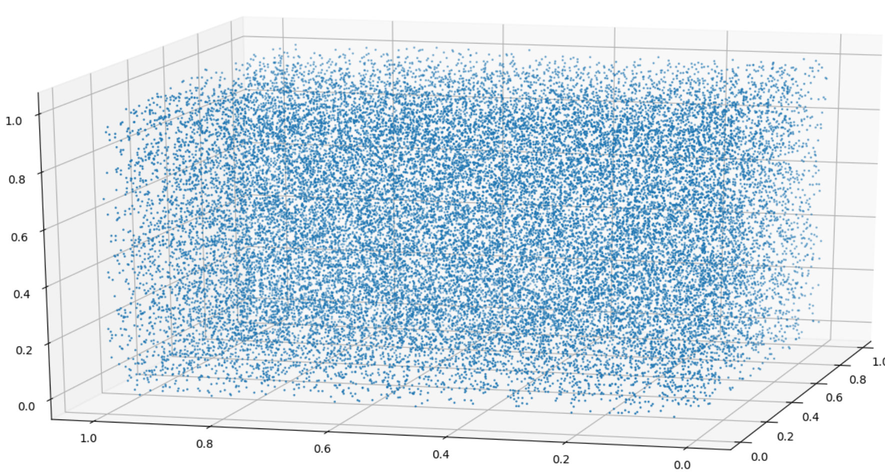


Figure 2: Random numbers generated from NumPy.

It is obvious from Figure 1 that the C standard library does not produce random numbers that are actually random and uncorrelated from other numbers, since the random numbers actually lie upon a set of planes in 3D space (and for n-space since this behavior is not limited to 3 dimensions).

I was unable to get the C standard library to work on my local (Windows) machine.

2. My rejection method has been written to generate exponential deviates from a Lorentzian distribution. I used a bounding function that is close in shape to an exponential decay, and the Lorentzian does a pretty good job of this compared with other functions. A Gaussian ends up dipping below the exponential and requires a relatively large σ to remain above the exponential over the range I selected. A power law goes to infinity at 0, which is undesirable for this application. The Lorentzian is greater than the exponential for all values (at least those I use here) and follows a decaying exponential pretty well.

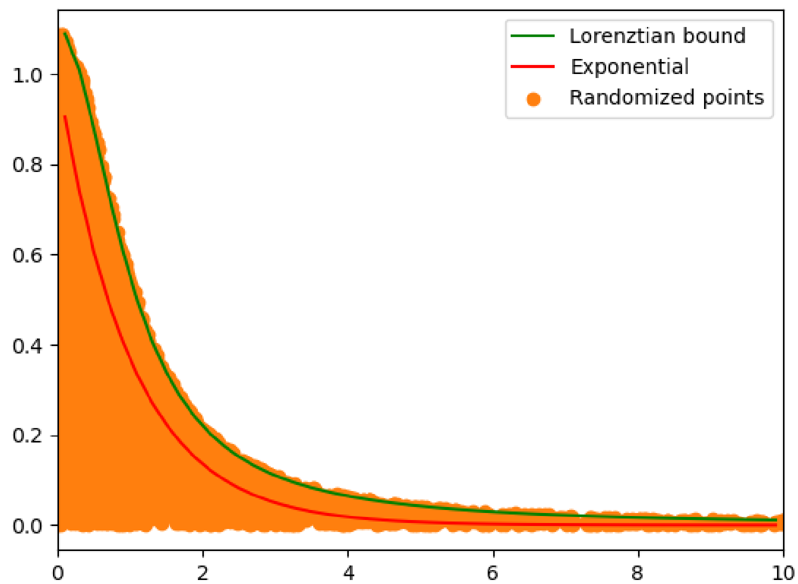
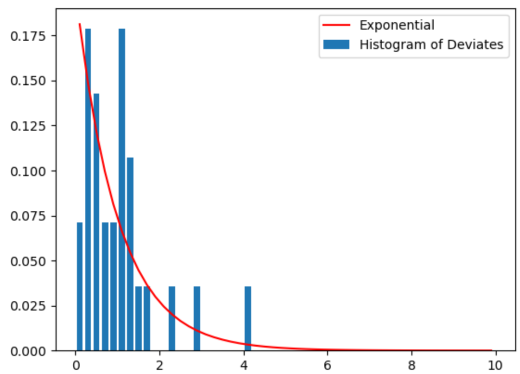


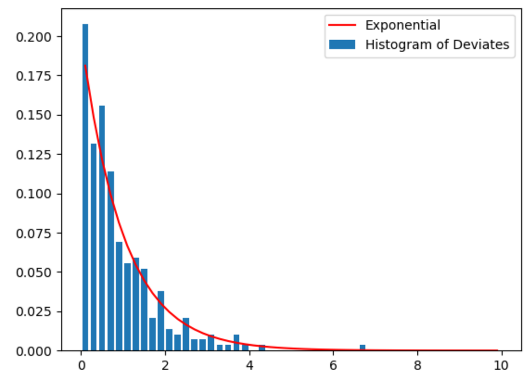
Figure 3: The Lorentzian distribution randomized points.

Figure 3 displays the Lorentzian I have selected (with amplitude 1.1 to keep it above the exponential), as well as the exponential (simply e^{-x}) and the randomized points living "within" (under) the Lorentzian. All points greater than e^{-x} are rejected. Note that the randomized points are all greater than 0. I then gather the randomized points into 51 bins (noting that the exponential has an equivalent number of points, with the bin centers lined up with these points on the x-axis) and plot their histograms compared with the exponential (note that the histograms and exponentials are normalized by their sum).

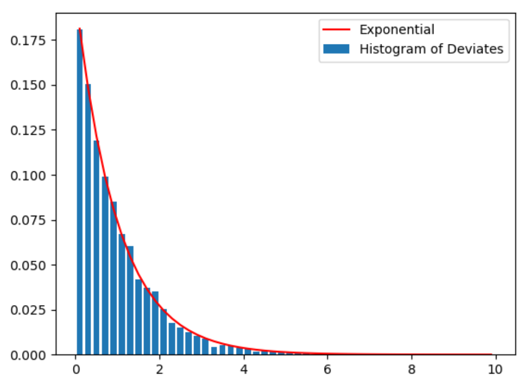
Figure 4 displays these histograms compared to the exponential for varying values of n, where n is the number of randomized points drawn from the Lorentzian distribution. We cannot say that the deviates follow the expected exponential decay for $n = 100$, but we get reasonable agreement for $n = 1000$, and even better for $n = 10000$. Of course, when $n = 100{,}000$, we get the best agreement. To zeroth order, you can get away with the deviates for $n = 1{,}000$, but a safer bet would be $n = 10{,}000$. Efficiency could possibly be further improved by reducing the amplitude of the Lorentzian (so less points are rejected), but it is already quite close to the exponential, so a solid majority of points are conserved as is.



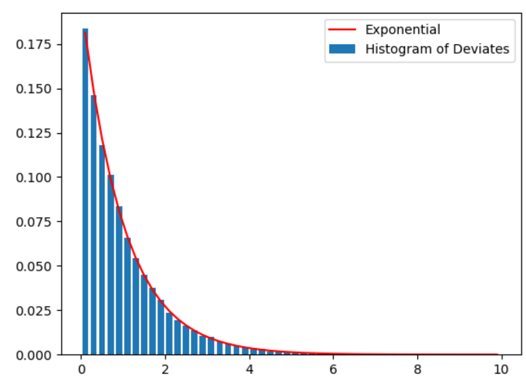
(a) $n = 100$



(b) $n = 1{,}000$



(c) $n = 10{,}000$



(d) $n = 100{,}000$

Figure 4: Histogram of deviates for varying n (number of randomized points) compared with expected exponential decay.

3. Now we repeat the above, but use a ratio-of-uniforms generator, beginning with creating a u, v plane, where $u \in (0, 1]$ (it should be noted that 0 is not included because this later results in division by 0 error). Since we have $x = v/u$, and we want $u < e^{-x}$, we solve for v as

$$u = e^{-x} = e^{-v/u} \quad \rightarrow \quad v = -u\ln(u). \tag{1}$$

The resulting u, v bounding region is displayed in red in Figure 5 (as with the rejection method, we only want positive deviates). The limits on this region are [0.0, 0.36787911641291404]. In this Figure are also displayed the randomly generated points in blue and the accepted points in orange, which are then used to calculate our deviates.

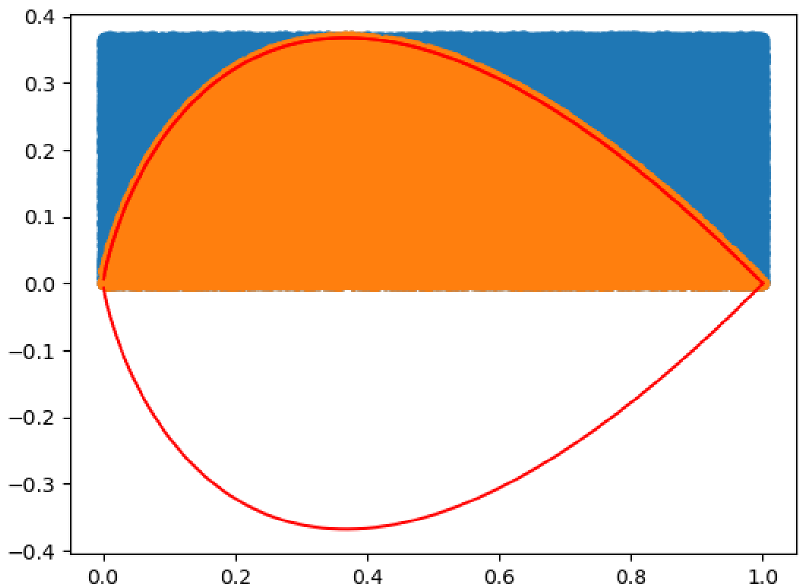
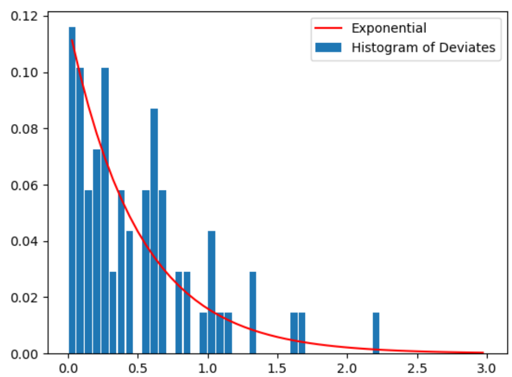


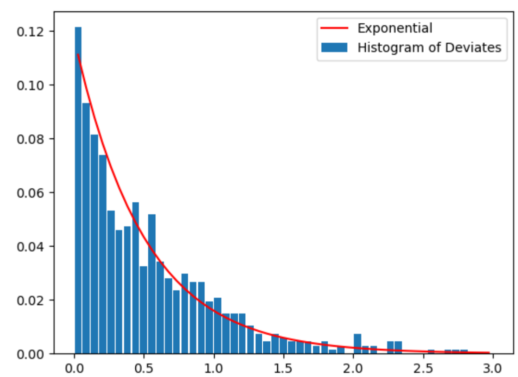
Figure 5: Bounding region in u, v space, along with randomly generated and accepted points.

Figure 6 displays the resulting histograms of exponential deviates and the expected exponential function. Similar to the rejection method, we find that we could get away with $n = 1{,}000$, but a better lower limit on the number of points we need before we get a good agreement between the histogram and the expected distribution is about 10,000.

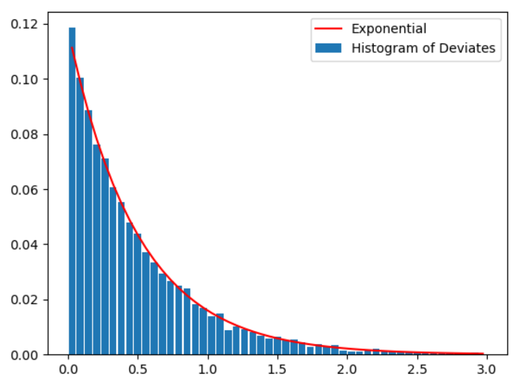
The ratio-of-uniforms method (for e^{-x}) accepts a little over 2/3 of the values (shown in Figure 6), remaining roughly constant for changing n, since the shape of the boundary in u, v space remains the same, and the points are randomly generated in a box constrained by the boundary (i.e. since we are using the same space and randomly generating values, we should get about the same acceptance rate).



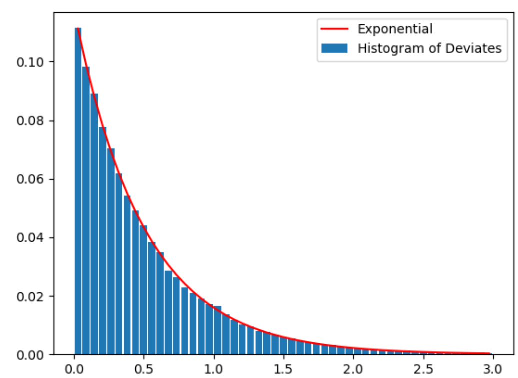
(a) n = 100, accepted 67.0% of values.



(b) n = 1,000, accepted 67.8% of values.



(c) n = 10,000, accepted 68.11% of values.



(d) n = 100,000, accepted 67.825% of values.

Figure 6: Histogram of deviates for varying n (number of randomized points) compared with expected exponential decay. Accepted percentage of exponential deviates also shown.