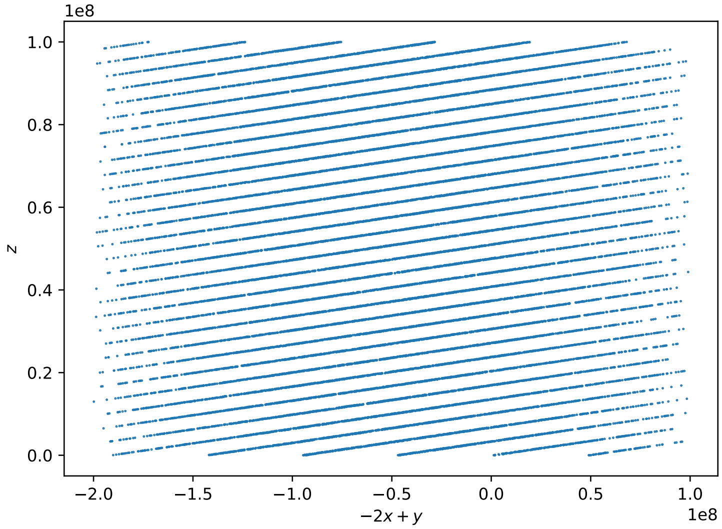

Figure 1: Scatter plot of z as a function of $-2x+y$. We can see very clear set of planes in the scatter plot, indicating a bad random number generator.

1. As shown in Figure 1, we plot z as a function of $-2x+y$ from `rand_points.txt`. We can see very clear set of planes in the figure. We can count a total of around 35 planes. I first verified that there are indeed planes using Octave with a simple

```
d = load('rand_points.txt')
scatter3(d(:,1),d(:,2),d(:,3),'.')
```

   and we can rotate the 3D plot freely to see if there are any planes. The appropriate factor -2 is found by iterating over a lot of linear combinations generated with `np.linspace`. I also generated similar random numbers using Python's `random.randint` and save in `rand_points_py.txt`. There is no such planes when viewed using Octave. The similar effect cannot be replicated on my computer running an up-to-date Arch Linux with GNU C Library version 2.36. The generated random numbers using glibc on my computer is saved in `rand_points_my.txt`.

2. For the Lorentzian (up to some normalization)

$$\frac{1}{1+x^2} \tag{1}$$

   which decays as x^{-2}. This mean for x large, the exponential we want to draw from is always smaller than the Lorentzian. For x near zero, we can choose a large enough prefactor such that the Lorentzian is larger than the exponential. Similarly, for the power law

$$\frac{\alpha-1}{x_{\text{min}}}\left(\frac{x}{x_{\text{min}}}\right)^{-\alpha}$$

   which decays as $x^{-\alpha}$ and for x large, the exponential is always smaller than the power law. Both Lorentzian and the power law can be used. However, for the Gaussian, we can see that

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

   decays as $\exp(-x^2)$. For x large, the Gaussian will always be smaller than the exponential, thus the Gaussian cannot be used here. We also plot these functions as anadditional check, shown in Figure 2.

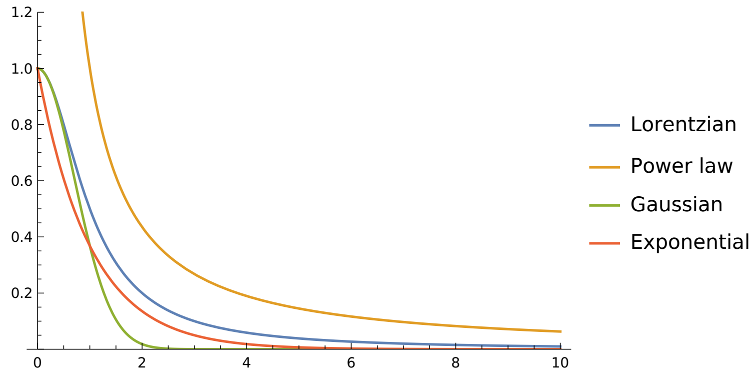

Figure 2: Plot of Lorentzian, power law, Gaussian and exponential. We can see that Gaussian decays faster and cannot be used in rejection method.

We will use a Lorentzian given in Equation 1 in this question, since we know from lecture that its normalized inverse CDF is

$$\arctan\left[\pi\left(x-\frac{1}{2}\right)\right]. \tag{2}$$

Note that the Lorentzian given in Equation 1 equals the exponential at $x = 0$ and is always larger than the exponential for $x > 0$, thus we can draw from Equation 1 directly without additional scaling. In `2.py`, random number in $[0.5, 1)$ is generated by rescaling the random numbers in $[0, 1)$. It is then used to generate Lorentzian for $x > 0$ using Equation 2. We draw the generated Lorentzian with probability and we have a exponential deviates using rejection method. We get an efficiency of

```
Accept probability is 0.637327
```

This is around the maximum efficiency with the Lorentzian, since we cannot reduce the prefactor any further. The histogram of the generated exponential deviates using rejection is shown in Figure 3.

3. Since the exponential is assumed to have $x \geq 0$, we must have $v \geq 0$. To determine the upper bound, we note that we need to take v such that

$$u < \sqrt{\exp\left(-\frac{v}{u}\right)} = \exp\left(-\frac{v}{2u}\right) \;\Rightarrow\; \ln u < -\frac{v}{2u} \;\Rightarrow\; v > -2u\ln u$$

and we have

$$v < \max_{u\in[0,1]} -2u\ln u = \frac{2}{\mathrm{e}}.$$

The accept probability is

```
Accept probability is 0.67874
```

which is a little larger than the method of rejection. The histogram of the generated exponential deviates using ROU is shown in Figure 4.

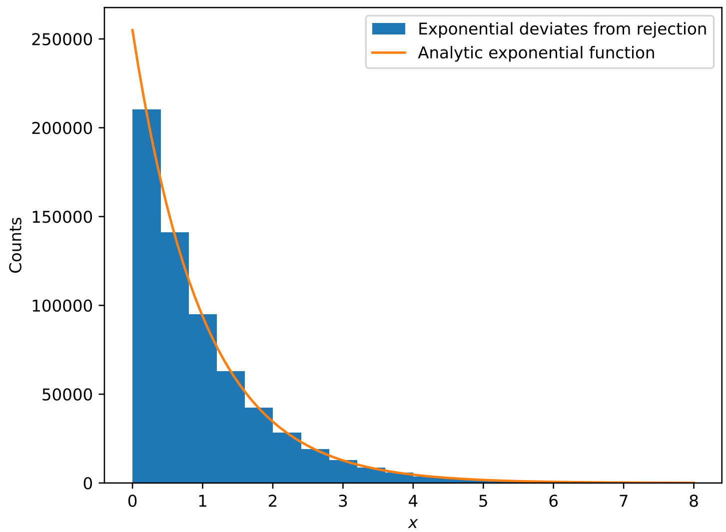

Figure 3: Generated exponential deviates using rejection method from the Lorentzian given by Equation 1. The analytic exponential function is normalized such that the area under the curve is the same as the area of the whole histogram.

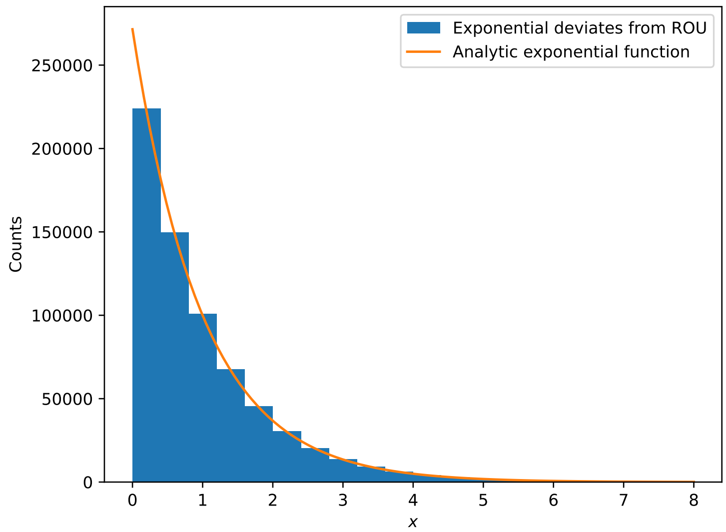

Figure 4: Generated exponential deviates using ROU. The analytic exponential function is normalized such that the area under the curve is the same as the area of the whole histogram.