Phys 512- Problem Set 7

Andrew Lewis

November 18, 2022

Problem 1

Show that when correctly viewed these triple lie along a set of planes
I loaded the random numbers from the text file and plotted them using ax.3d scatterplot, I then rotated the image until planes were visible:

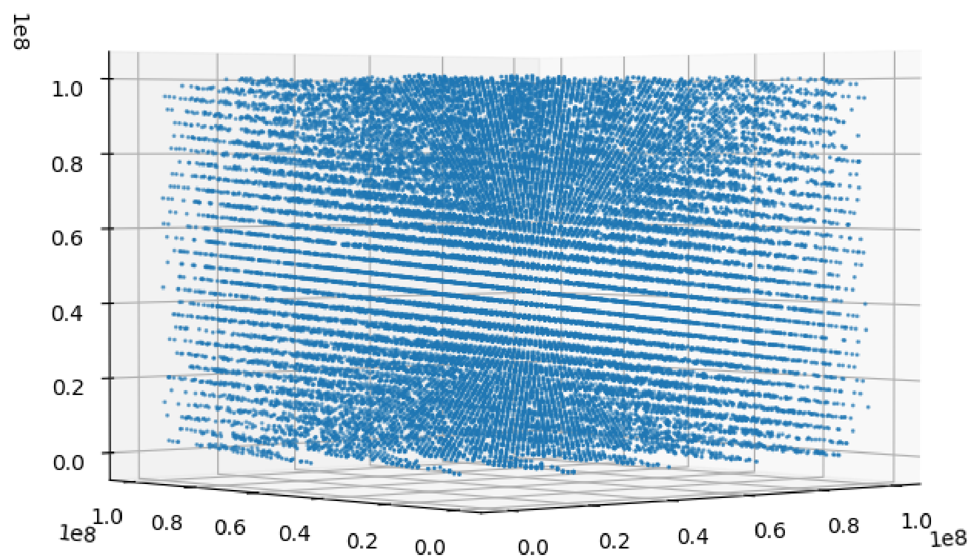

Figure 1: Random number tuples from c's (broken) random number generator

This shows that these supposed 'random' numbers are not random at all and instead have patterns in them. This is obviously a problem if your work requires you to sample a random number that lies between the planes.

Do you see the same effect with python's random number generator?
In python's random number generator I did not see the same effect. Instead of the clear planes seen above, I just got a blob of random numbers. This suggests that Python's random number generator is at least superior in the sense that it doesn't discretize points onto planes. This does not rule out the possibility that there is some pattern to the python random numbers.

Do you see the same effect on your local machine?

I was unable to get this part to work.

Problem 2

Write a rejection method to generate exponential deviates from another distribution
You can find the code to generate exponential deviates, given another distribution, in Q2.py.

Which of Lorentzians, Gaussians and power laws could you use for the bounding distribution?

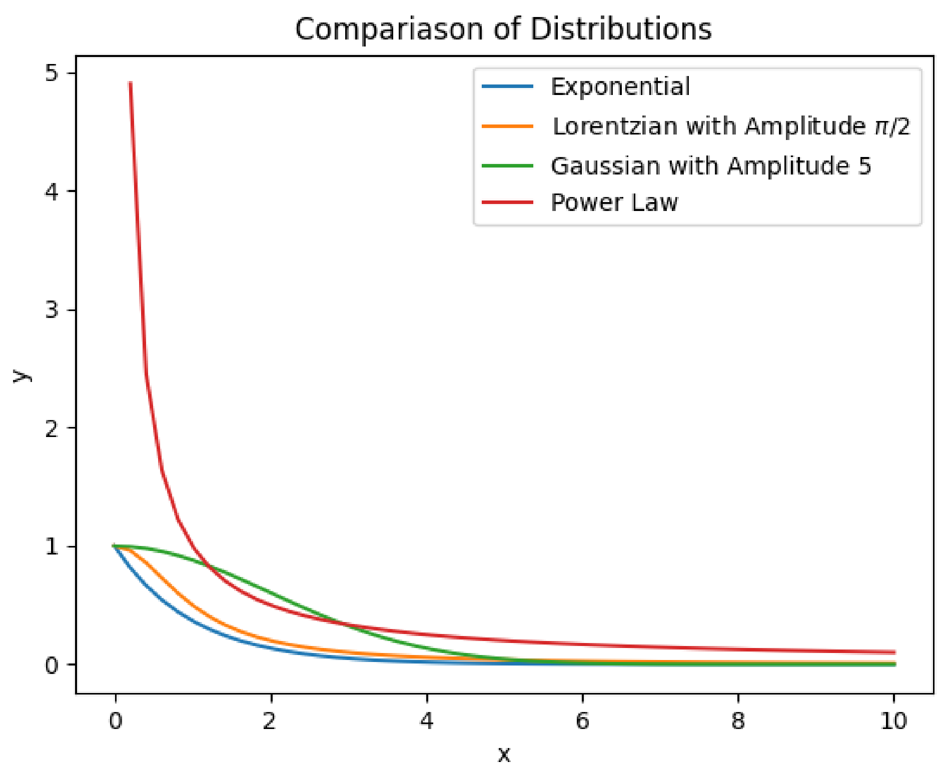

Figure 2: Compariason of the various distributions

While you could use all three as a bounding distribution, it is clear that a lorentzian is with an amplitude of $\pi/2$ is the best option as it most closely matches the the exponential distribution.

Show that a histogram of your deviates matches up with the expected exponential curve

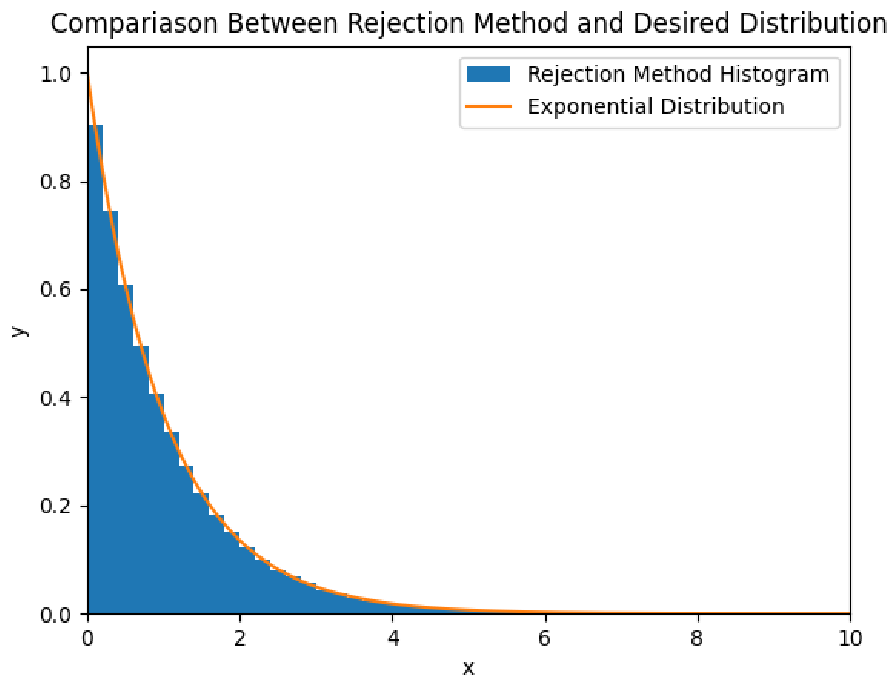

Figure 3: Comparison between the histogram rejection method points and the target exponential distribution

As can be seen above the histogrammed generated samples clearly line up with the target exponential distribution, which is exactly the behavior we were looking for.

How efficient can you make this generator?
In order to properly sample the exponential distribution, the lorentzian distribution must be greater than the exponential distribution at all points that we are sampling. To determine what coefficient we need to put at the front of the lorentzian distribution in order for it to be larger, I simply slowly shrunk the coefficient until the lorentzian was smaller than the exponential at some x (brute force I know). Now that I had a value for the coefficient in front of the lorentzian I can then sample from my distribution.
As it turns out my guess for normalization of $\pi/2$ was the optimal value, this makes sense as we want the peak value of the distributuion to be 1, and integrating the lorentzian from 0 to ∞ gave a value of $\pi/2$

By looking at how many samples were accepted vs. how many samples were generated we can determine an efficiency for the generator. Since we need to generate 2 uniform numbers per sample, and we accept approximately 63% of all steps the fraction of uniform deviates that give rise to an exponential deviate is:

$$\frac{.63 \text{ Deviates per sample}}{2 \text{ Uniforms per sample}}$$
$$= .315 \text{ Exponential Deivates per Uniform}$$

Which isn't bad considering the maximum efficiency we could have would be .5 deviates per uniform variable.

Problem 3

If u goes from 0 to 1, what are your limits on v?
The ratio of uniforms generator works as the following:

1. generate numbers, u,v
2. if u is $\leq \sqrt{p(\frac{u}{v})}$ then
3. return v/u as an acceptable point

We know from the question that:

$$0 \leq u \leq 1$$

Therefore we see that if the maximum value u can take is 1, we can work out the maximum value that v can take.

$$u = \sqrt{p(\frac{v}{u})}$$

this expression is the maximum value u can have and still have a sample be accepted. We don't care about us outside of this range because they would not be accepted anyways, after all we want our distribution to be a close as possible to our target distribution. Substituting in our exponential for our probability distribution

$$u^2 = e^{-\frac{v}{u}}$$

$$2ln(u) = -\frac{v}{u}$$

$$-2uln(u) = v$$

As we all know by now the easiest way to get a maximum value is to take the derivative and set it to zero, doing the following:

$$-\frac{dv}{du} = 2(ln(u) + 1)$$

$$-1 = ln(u)$$

$$u = e^{-1}$$

Therefore we can then plug in our value for u to obtain the maximum value acceptable for v. Using our earlier expression:

$$v = -2uln(u)$$

$$v = -2e^{-1}ln(e^{-1})$$

$$v = 2e^{-1}$$

Therefore the maximum limit on v is:

$$\text{max(v)} = 2e^{-1}$$

How efficient is this generator?
As we get .68 Exponential deviates per sample the efficiency per uniform number is

$$\frac{\text{.68 Exp Deviates/sample}}{\text{2 Uniform Deviates / Sample}}$$

$$\text{.34 Exp DeviatesUniform Number}$$

Which is slightly more efficient than our rejection method

Plot the histogram?
See the following histogram:

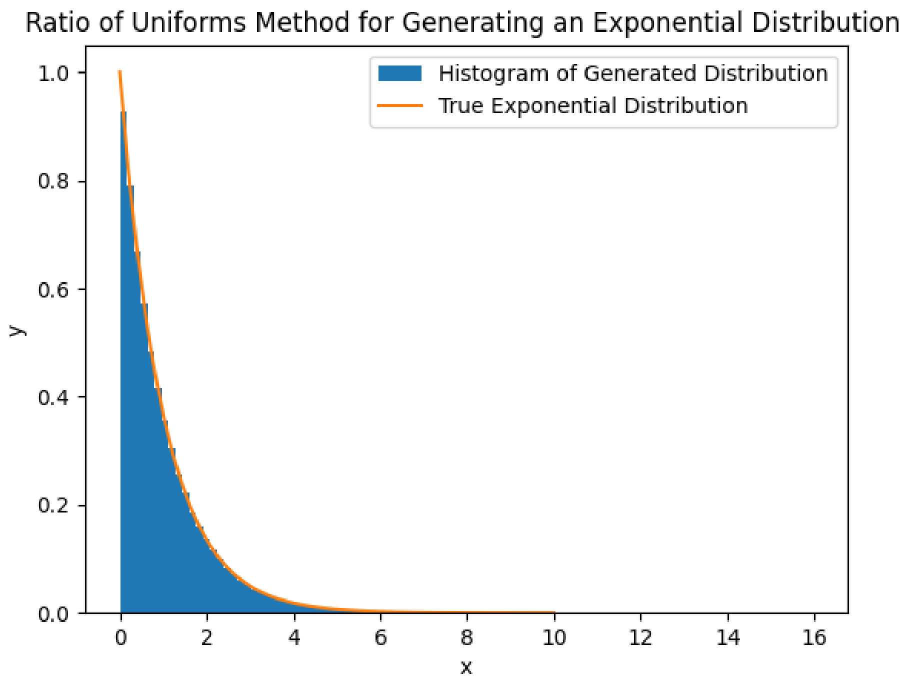

Figure 4: Comparison of the target exponential distribution and the histogramed generated points

As can be seen in the above figure, the histogram once again matches the target distribution and gives the correct answer. This impressive as we were able to squeeze out greater efficiency compared to the rejection method.