

PHYS 512 Problem Set 7

Teophile Lemay, 281081252

1

I was not able to get the C standard PRNG working on my computer, so I used the previously produced random numbers shared in `rand_points.txt`. Figure 1 shows a 3D scatter plot of the subset of pseudo-random points produced by the C standard PRNG. rotating the selected points to view the scatter plot from the correct angle reveals that the random points are actually distributed on multiple separate planes. Clearly, the C standard PRNG does not produce a uniform distribution in 3D space.

To compare the C standard PRNG to a more modern PRNG, I produced a new set of $3 \cdot 10^8$ pseudo random numbers between 0 and 2^{31} using Numpy's `default_rng().integers()` function which uses the PCG-64 algorithm. After truncating the selection to a viewable subset (x, y, z between 0 and 10^8), the scatter plot output was examined by rotating it under multiple (non-exhaustive) combinations of rotations. No distinct planes (or other patterns) such as those found in the C standard PRNG were found.

2

The rejection method for producing random deviates for some distribution from random deviates from some other distribution requires the second distribution to be equal or larger than the first at all points. If my target distribution is the exponential distribution

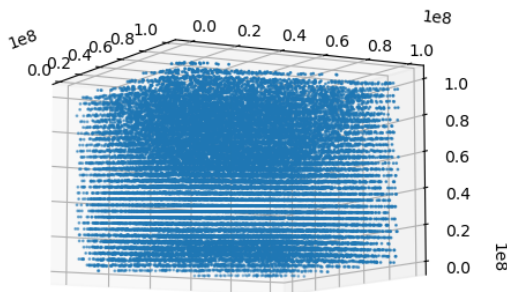
$$p_e(x) = \begin{cases} e^{-x} & x \geq 0 \\ 0 & x < 0 \end{cases} x$$

then I can only choose sampling distributions that are larger at all points. Of the three options (power law, Cauchy, and Gaussian distributions), both a power law distribution and a Cauchy distribution are able to serve as bounding distributions. The Gaussian distribution

$$p_G(x) = e^{-x^2}$$

decays much faster than e^{-x} for large x , even for a Gaussian with a large variance. For the other two distributions, it is possible to prove that they are both equal or greater than the exponential distribution at all points.

Figure 1: Non-uniformity in C standard PRNG



First, the power law distribution $p_p = \frac{1}{x}$. For $x < 0$, this is trivial since $p_e(x) = 0$ and $\frac{1}{x}$ is greater than 0 for all x . For $x > 0$, this can be proven by contradiction: assume

$$e^{-x} > \frac{1}{x} \quad \text{for } x \geq 0$$

then

$$\begin{aligned} e^{-x} &> e^{\ln \frac{1}{x}} = e^{-\ln x} \\ -x &> -\ln x \end{aligned}$$

This is impossible since $-\ln x$ is always larger than $-x$. Thus, $p_p(x)$ is greater than $p_e(x)$ at all points.

Similarly, the Cauchy distribution $p_c(x) = \frac{1}{1+x^2}$ is proved to be greater or equal than p_e . Again, for $x < 0$, this is trivial as $\frac{1}{1+x^2}$ is greater than 0 for all x . For $x \geq 0$, I use a proof by contradiction again. Assume

$$e^{-x} > \frac{1}{1+x^2} \quad \text{for } x \geq 0$$

then

$$\begin{aligned} e^{-x} &> e^{\ln \frac{1}{1+x^2}} = e^{-\ln(1+x^2)} \\ -x &> -\ln(1+x^2) \\ 0 &> x - \ln(1+x^2) \end{aligned}$$

at $x = 0$, $x - \ln(1+x^2) = 0 - \ln(1) = 0$. the derivative of $x - \ln(1+x^2)$ is $1 - \frac{2x}{1+x^2} \geq 0 \forall x > 0$ so $x - \ln(1+x^2) \geq 0 \forall x \geq 0$, in contradiction to the initial assumption.

Both a power law and a Cauchy distribution meet the requirements for a bounding distribution in rejection sampling, but the power law diverges as $x \rightarrow 0$ so I choose to use the Cauchy distribution. On top of not needing to deal with a power law distribution diverging to ∞ , the Cauchy distribution also stays much closer to the exponential distribution than the power law which increases the efficiency of the rejection sampling.

In order to sample the Cauchy distribution in my code to use it for rejection sampling, I need the inverse CDF of the Cauchy distribution:

$$CDF = \int_{-\infty}^x dt \frac{1}{1+t^2} = \arctan t \Big|_{-\infty}^x = \arctan(x) + \pi/2 .$$

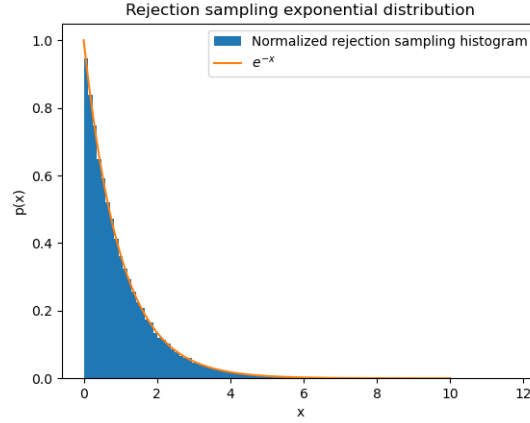
Using this result to generate pseudo-random deviates from the Cauchy distribution, I used rejection sampling to sample the exponential distribution using a Cauchy distribution. Figure 2 shows that the results of using the rejection sampling method to produce $N = 100000$ random deviates. As expected, the normalized histogram of the deviates shows a very close match to an exponential distribution. Using the Cauchy distribution, the rejection method has an efficiency fraction of ≈ 0.318 . This efficiency could be improved to use a larger fraction of the input uniform deviates by finding some other distribution that fits between the Cauchy and the exponential distributions at all points. Code for this question is in `Q2_rejection.py`

3

For the ratio of uniforms method, I define a region in the u, v plane such that $0 \leq u \leq \sqrt{p(v/u)}$, then produce uniform samples in some bounding box that encompasses the region, and return v/u for any u, v pair from the uniform sampling that land within the distribution's region in the plane. For the exponential distribution, the acceptance region is

$$u, v \text{ s.t. } 0 \leq u \leq \sqrt{e^{-\frac{v}{u}}} = e^{\frac{-v}{2u}} .$$

Figure 2: Rejection sampling method for exponential distribution



The bounding box for this region can be found by transforming the equation to get v as a function of u

$$\ln u \leq \frac{-v}{2u}$$

$$0 \leq v \leq -2u \ln u = -u \ln u^2$$

$$v(u) = -u \ln u^2 .$$

$$\lim_{u \rightarrow 0} v(u) = 0 \quad \text{and} \quad v(1) = 0$$

so u is drawn from the uniform distribution on $[0, 1]$. Between 0 and 1, $v(u)$ is positive and has a maximum for $v'(u) = 0$:

$$0 = v'(u) = -\ln u^2 - 2$$

$$\ln u = -1$$

$$u = \frac{1}{e}$$

$$v\left(\frac{1}{e}\right) = \frac{2}{e}$$

so v is drawn from the uniform distribution on $[0, \frac{2}{e}]$. I used these results to produce $N = 100000$ random deviates from the exponential distribution using ROU sampling. Figure 3 shows that the histogram of the samples is a good match to the exponential distribution as expected. The ROU method is also more efficient than rejection sampling with the Cauchy distribution, with an efficiency of ≈ 0.680 , using approximately twice the number of random points as the rejection method. Code for this question is in `Q3.ROU.py`

Figure 3: ROU sampling method for exponential distribution

