Assignment 6
Phys 512 Computational Physics

Hans Hopkins 260919593

Q1:
Plotting the random points from rand_points.txt

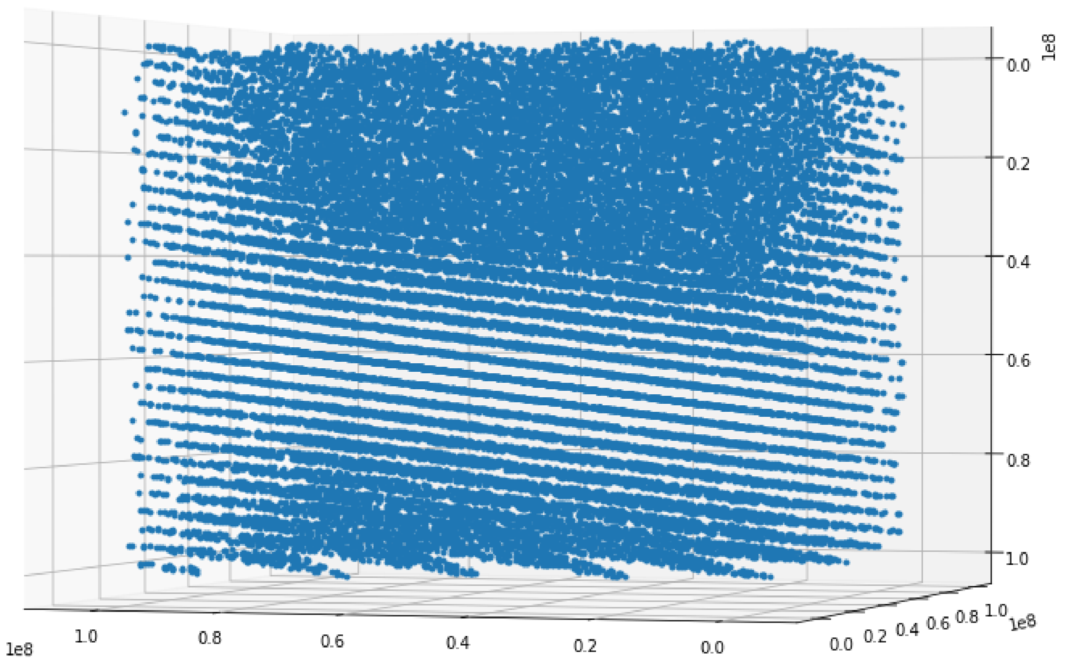

Figure 1: Here you can see gaps in the planes. This was the best angle I could find.

When I used numpy's randint(), I couldn't see the effect anymore.
Plotting the random points from Python:

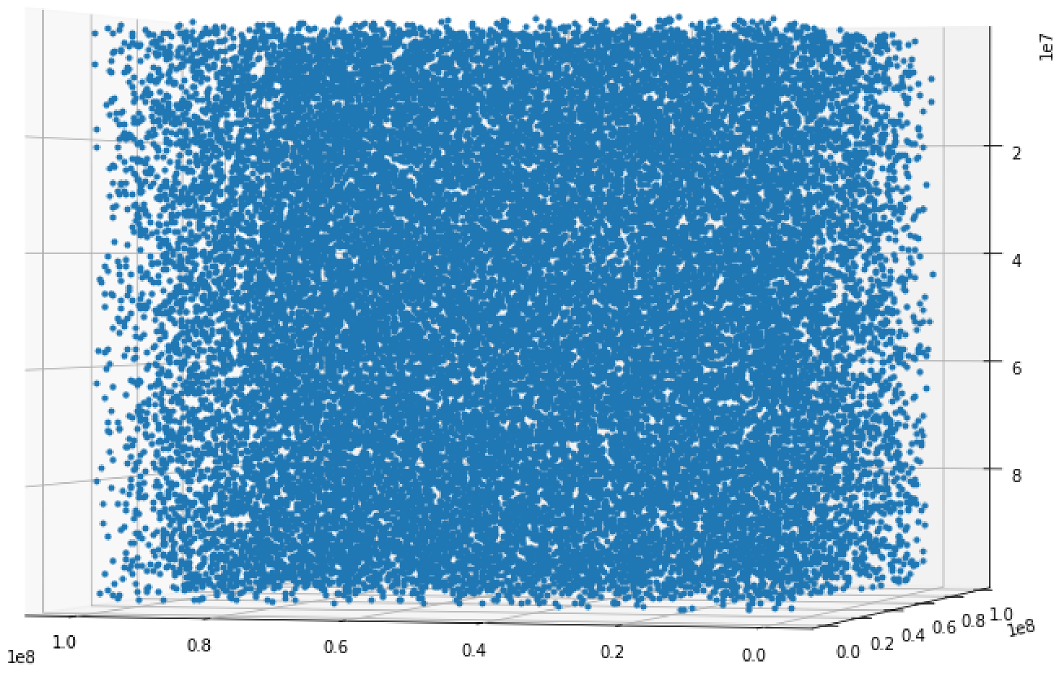

Figure 2: I couldn't see any non-isotropic gaps from any angles. This is about the same angle as the last image.

I couldn't get the it to work locally. None of my c compilers make it obvious which library file is what. Also when I tried to run it from Linux it says that I was using the wrong architecture or something. I know fixing this is going to take me like 3 days so I'd rather skip.

Q2:

A Gaussian wouldn't work since e^{-x^2} would always eventually get smaller than our exponential distribution.

I think both a Lorentzian and a a power law would work, since they can always be greater than an exponential. But a Lorentzian with default parameters is already snug against the exponential curve, so let's use that.

Checking to see that the exponential is indeed always lower:

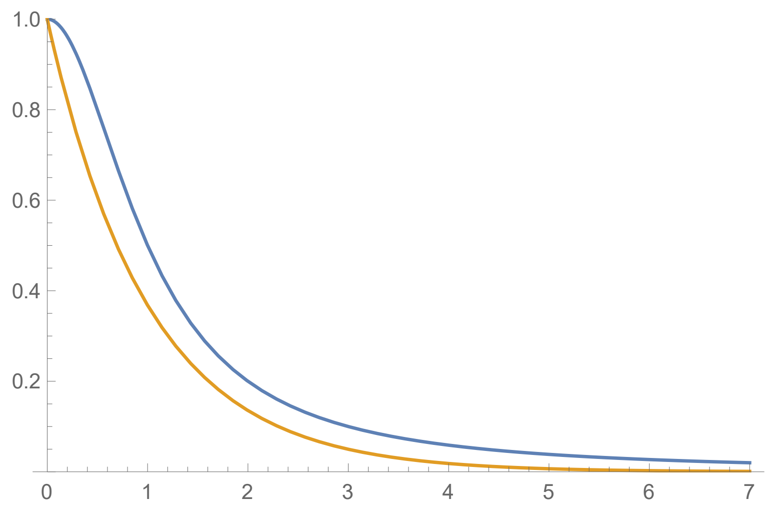

Figure 3: At no point does the orange curve (exponential) go above the blue curve (Lorentzian). Also from here on the Lorentzian acts as a power law, and that will fall slower than an exponential.

Running the code gives this histogram:

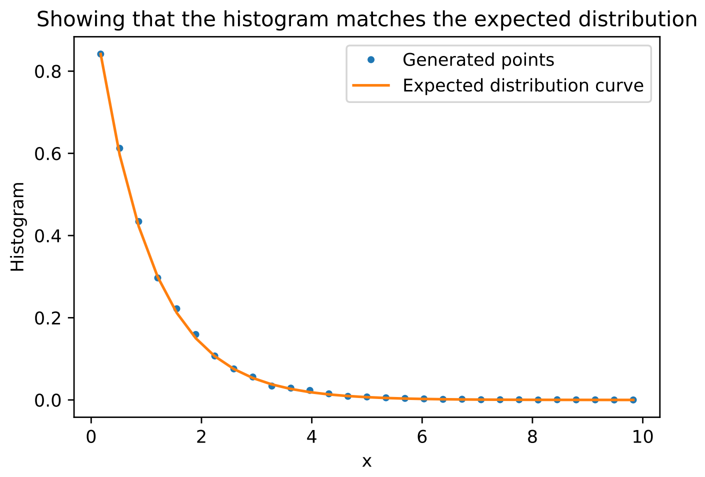

Figure 4: It's normalized so that the first point matches exactly. There's a few points that wiggle around the expected values, but that's also expected since I'm only sampling 10000 times.

Keeping track of how many initial uniform deviates I have to take, I end up using around 63%-64%. But, I have to take another uniform deviate to test against, so technically it's more like 32% efficient by that metric. This doesn't seem too bad. To make this more efficient you would want to find a closer fitting function to the exponential.

Q3: First I want to plot out the intersection between the plane and our probability surface. The equation is

$$\begin{aligned} u &= \sqrt{e^{-v/u}} \\ u^2 &= e^{-v/u} \\ 2\ln(u) &= -\frac{v}{u} \\ v &= -2u\ln(u) \end{aligned}$$

This is what the region looks like:

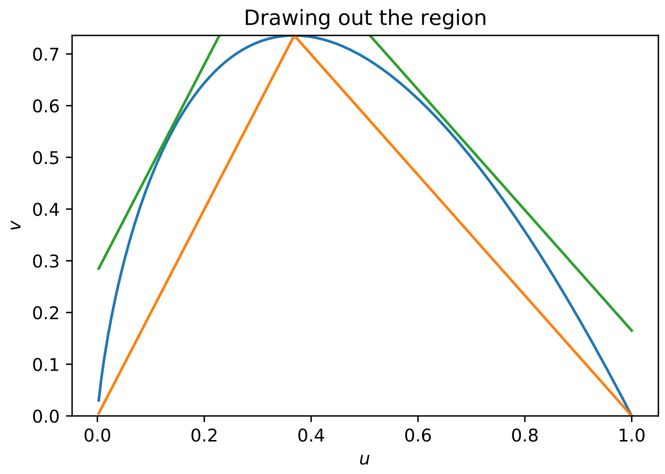

Figure 5: The actual region is blue. The orange is the interior squeeze triangle and the green is the exterior squeeze triangle.

I noticed that the shape of the region looks vaguely triangular to first order, so I figured I could draw triangles on the inside and outside to compare against. Now, the actual curve that we're testing against is already very fast, so is comparing against two triangles actually faster? I don't really know, but it can't hurt too much.

Also the maximum v out of my test points is 0.73576, but that's only actually good to like 3 sig figs, so let's consider v values in (0, 0.736).

The histogram we get from the sampling is this:

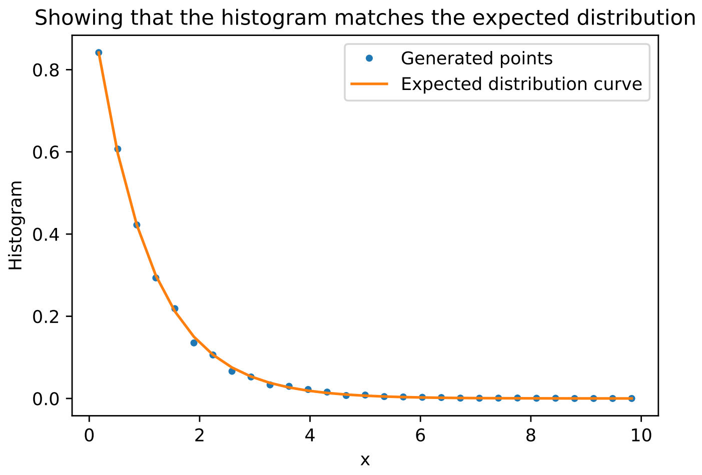

Figure 6: It looks pretty much the same as in question 2, which is good.

Running it a couple times, I keep about 68% of points, or 34% of my uniform deviates. This is only a little bit more efficient than the Q2 method.