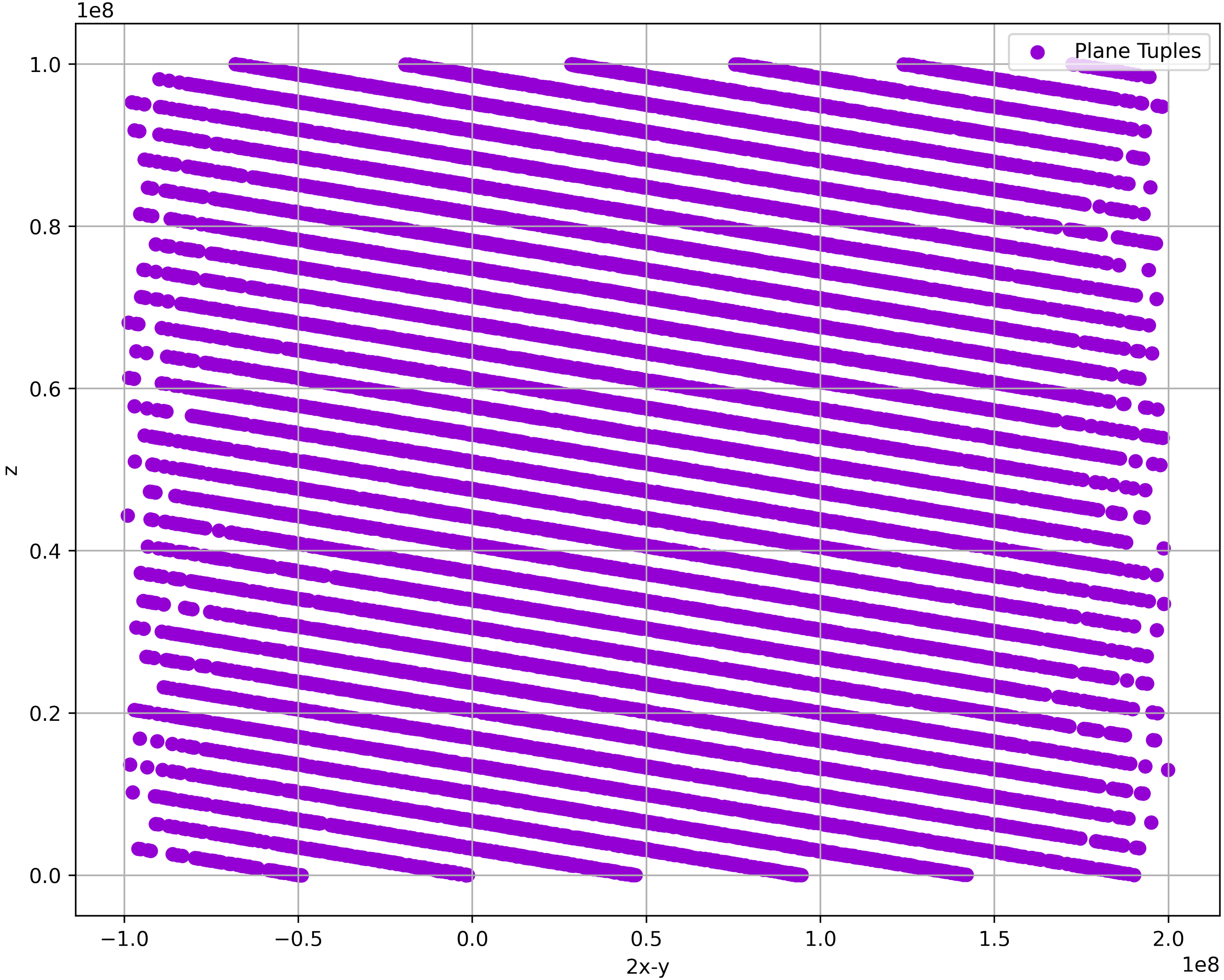
PS7 : Phys 512

1) First, we would like to view the correlation between the random 3-tuples of generated numbers in the form of stacked 2D planes embedded in the 3D ambient space

↳ To do so we will plot the cross section of these stacked planes by restricting our orientation to a fixed plane given by:

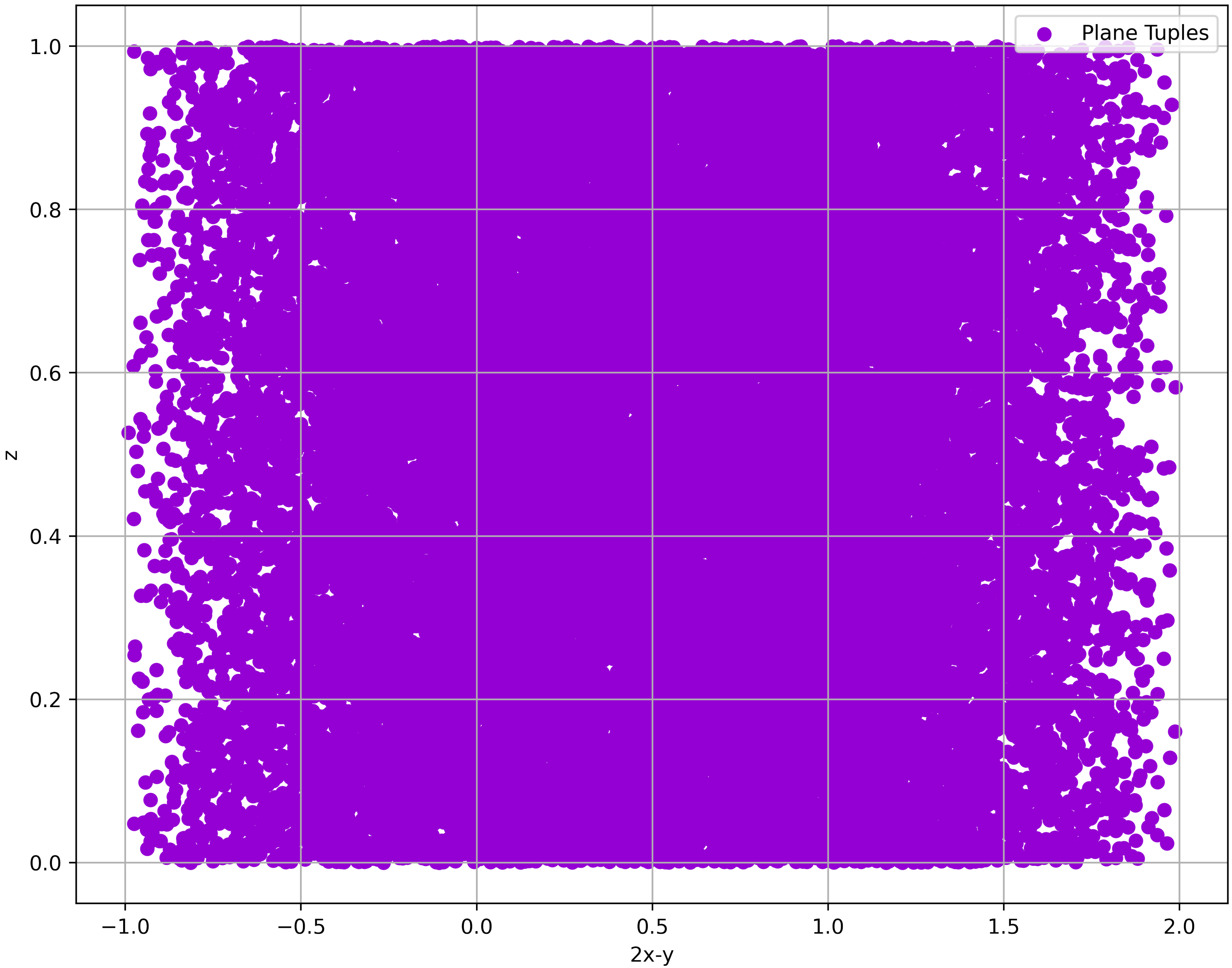
$$z = f(x,y) = ax + by$$

Trial & error gives us $a=2$, $b=-1$ for which the plot $(2x-y, z)$ looks like:


1e8
1.0
0.8
0.6
0.4
0.2
0.0
z
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
1e8
2x-y
Plane Tuples

We can see the structure of the stacked planes through the planar cross section & thus confirm the correlation between the 3-tuples

Does the same happen w/ Python's number generator? We randomly produce 3-tuples & plot the results for the same planar restriction:



Plane Tuples
1.0
0.8
0.6
0.4
0.2
0.0
z
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2x-y

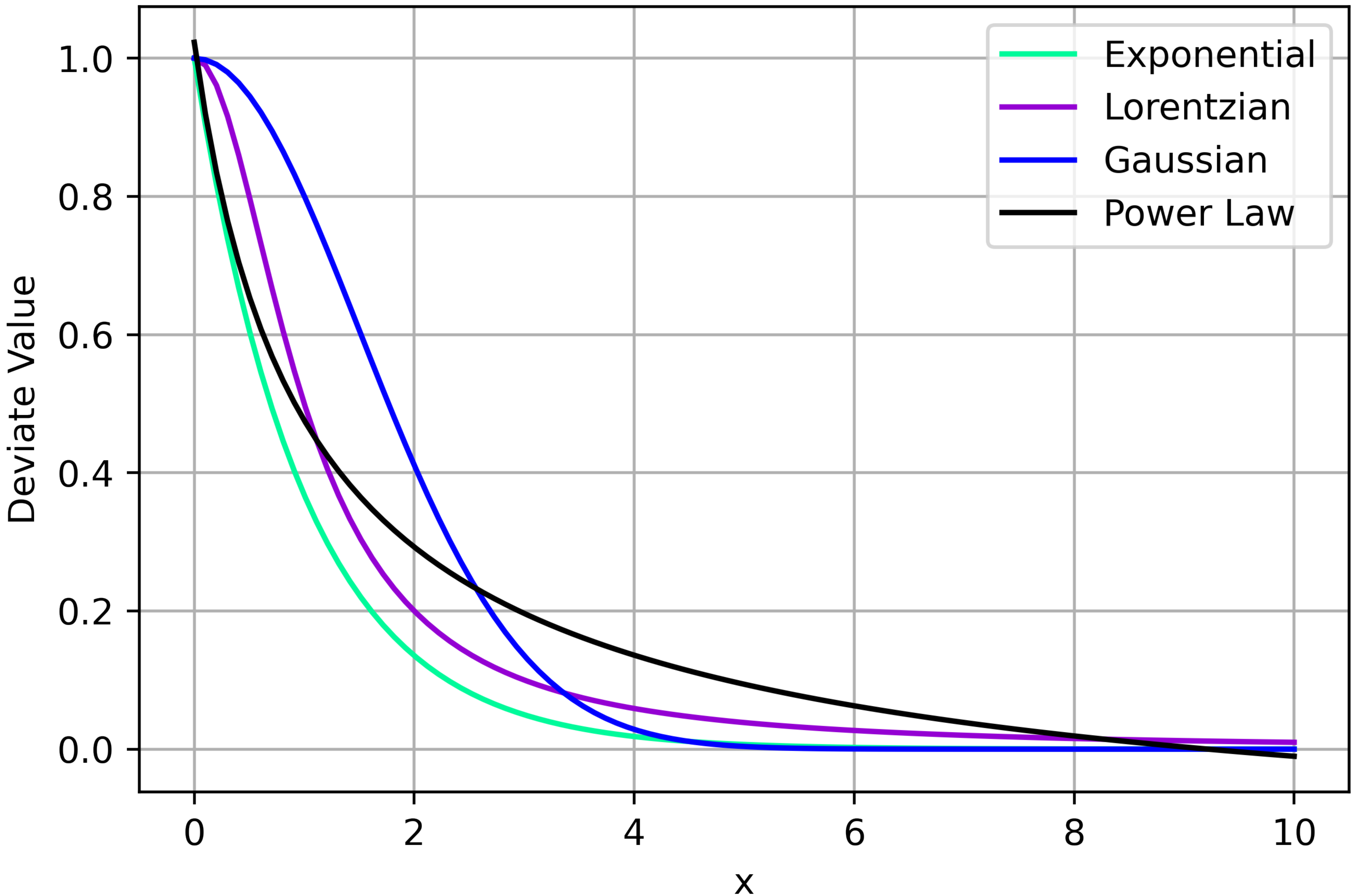
We can see there is no visible correlation between the tuples.

↳ One can argue this might not hold for other 'a' & 'b' values but it can be checked by plotting the tuples in 3D space & seeing no correlation from all angles.

Note: I could not get libc.so or libc.dylib on my Windows machine, unfortunately.

2) Now we would like to randomly sample points lying w/in an exponential function **but** we cannot use an exponential & instead use a Lorentzian/Gaussian/powerlaw

↳ We plot all functions against each other to get:



Exponential
Lorentzian
Gaussian
Power Law
1.0
0.8
0.6
0.4
0.2
0.0
Deviate Value
0
2
4
6
8
10
x

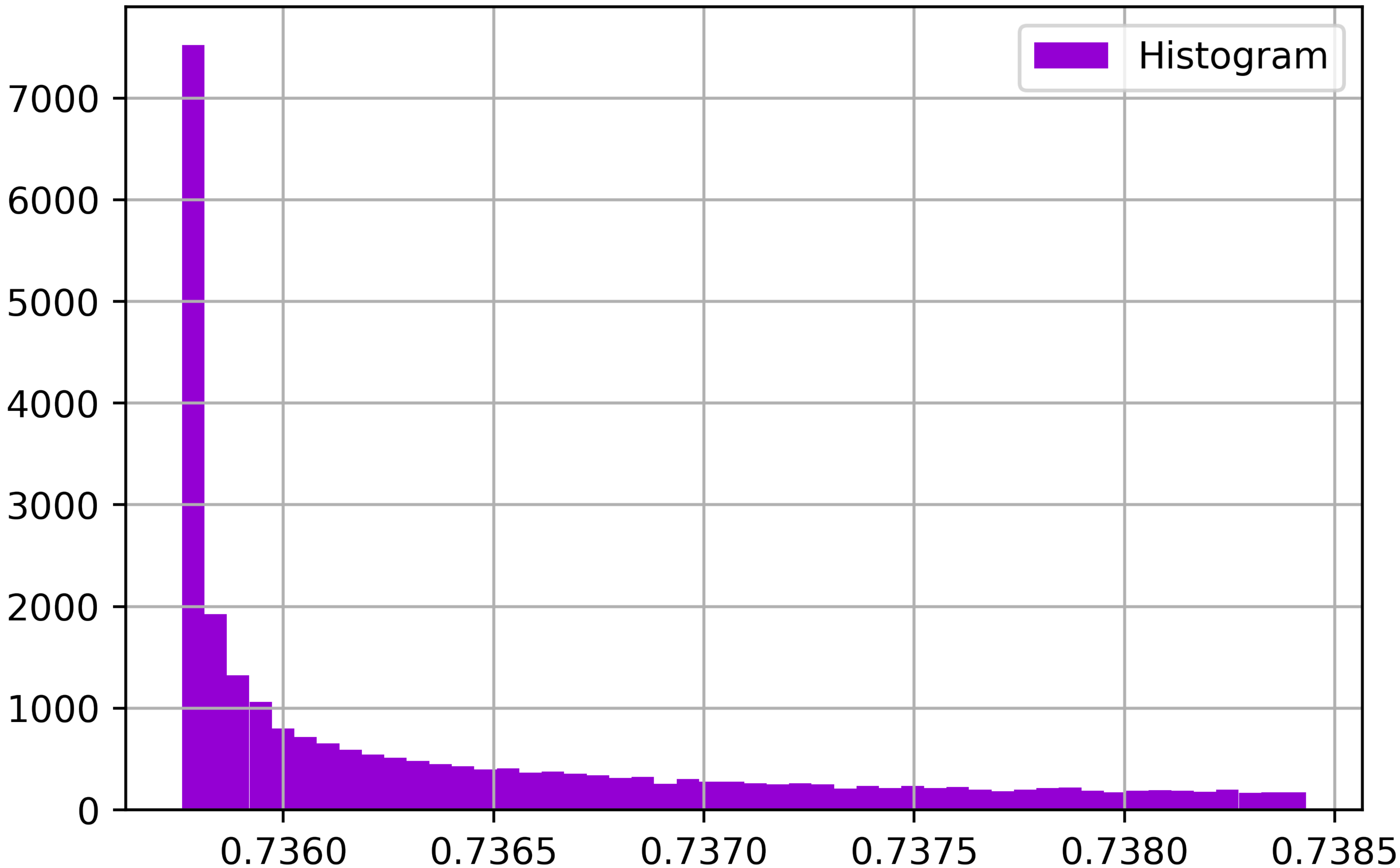
We can see the curvature of the Lorentzian matches most closely to the exponential & is always greater than the exponential

↳ For a Lorentzian $L(x)$ & exponential $E(x)$, our rejection condition will be: $L(x) > E(x)$

$$\frac{E(x)}{L(x)} < x \quad \forall x \in x\text{Range}$$

$$\Rightarrow L(x) > E(x)/x$$

We can use this to generate a ton of points sampled only under the Lorentzian and we can plot the ratio $E(x)/L(x)$ as histograms to get:



Histogram
7000
6000
5000
4000
3000
2000
1000
0
0.7360
0.7365
0.7370
0.7375
0.7380
0.7385

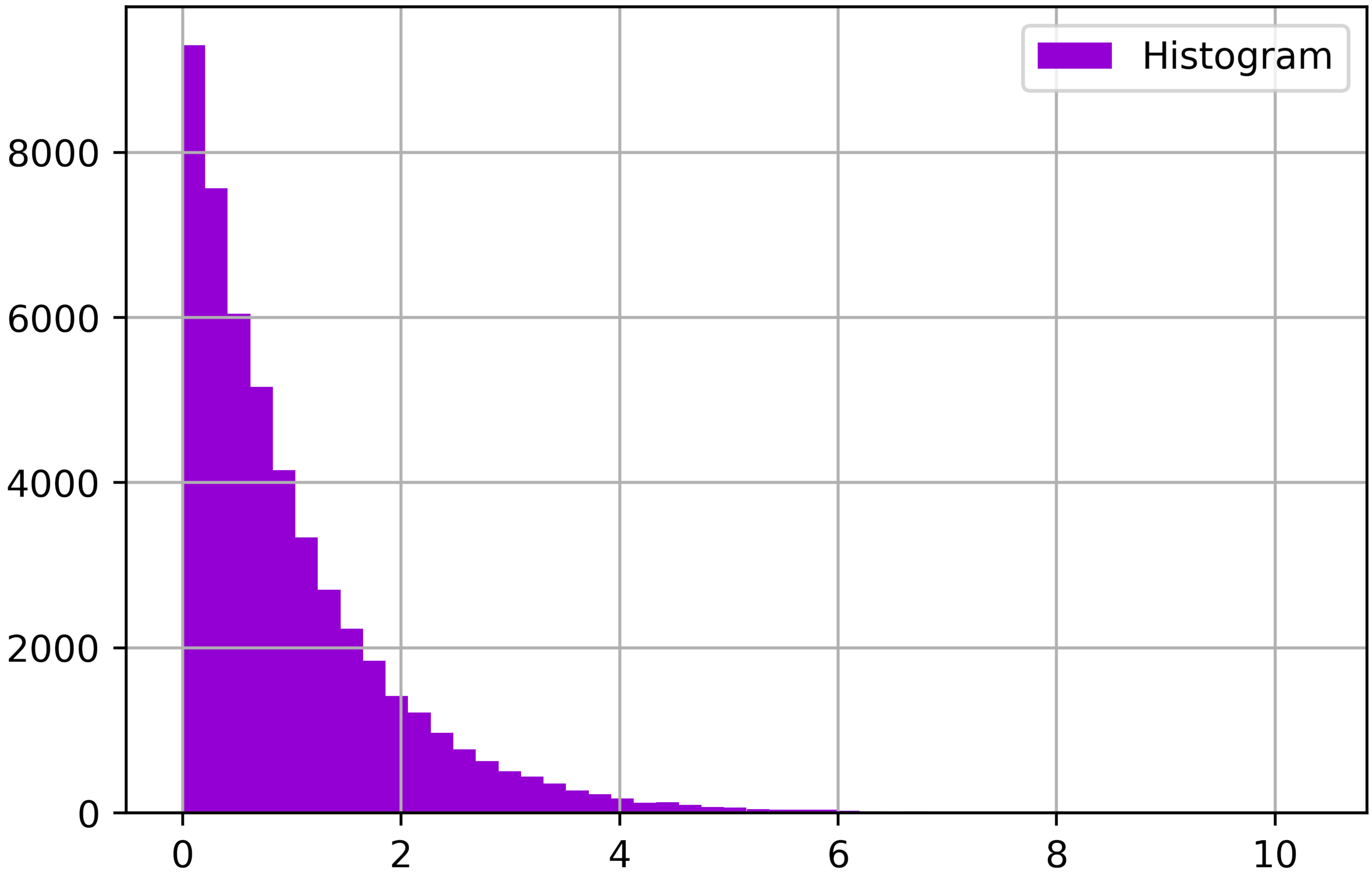
As we can see it clearly follows the exponential condition e^{-x}

We can make this estimate more efficient by restricting the similarity between the Lorentzian & the exponential.

3) Now we repeat the same thing we did in Q2 but now we use a ratio-of-uniforms generator

↳ Since $u \in [0,1]$, $v \in [0,1]$

Using this we get the following plot:



Histogram
8000
6000
4000
2000
0
0
2
4
6
8
10

We can see this is more efficient as the fraction of kept points is 0.50089

↳ We need less random points for a higher fraction

p is thus more efficient.