1. The planes can be seen clearly in planes.png. In planes_py.png, we see the distribution of points from numpy's random number generator. The plane effect is not observed. I was unfortunately not able to get test_broken_libc.py to run on my laptop.

2. A Lorentzian distribution works well in this case because it goes to zero slower than the exponential. For an exponential PDF $p(x) = \alpha e^{-\alpha x}, x > 0$, we can choose a the bounding distribution $p'(x) = \alpha/[1+(\alpha x)^2], x > 0$ which will always be greater than or equal to $p(x)$ (in both cases $y = 0$ for all $x < 0$). I believe this is the most efficient Lorentzian bounding function. Code for my rejection method is shown in p23.py, and the resulting histogram is shown in expdev_rejection.png, where the red line shows the expected curve. My rejection generator keeps about 64% of the points.

3. My ratio-of-uniforms generator can be seen in p23.py. For $0 < u < 1$ and $\alpha = 1$, I find that v goes from 0 to about 0.736. My ratio-of-uniforms generator keeps about 68% of the points.