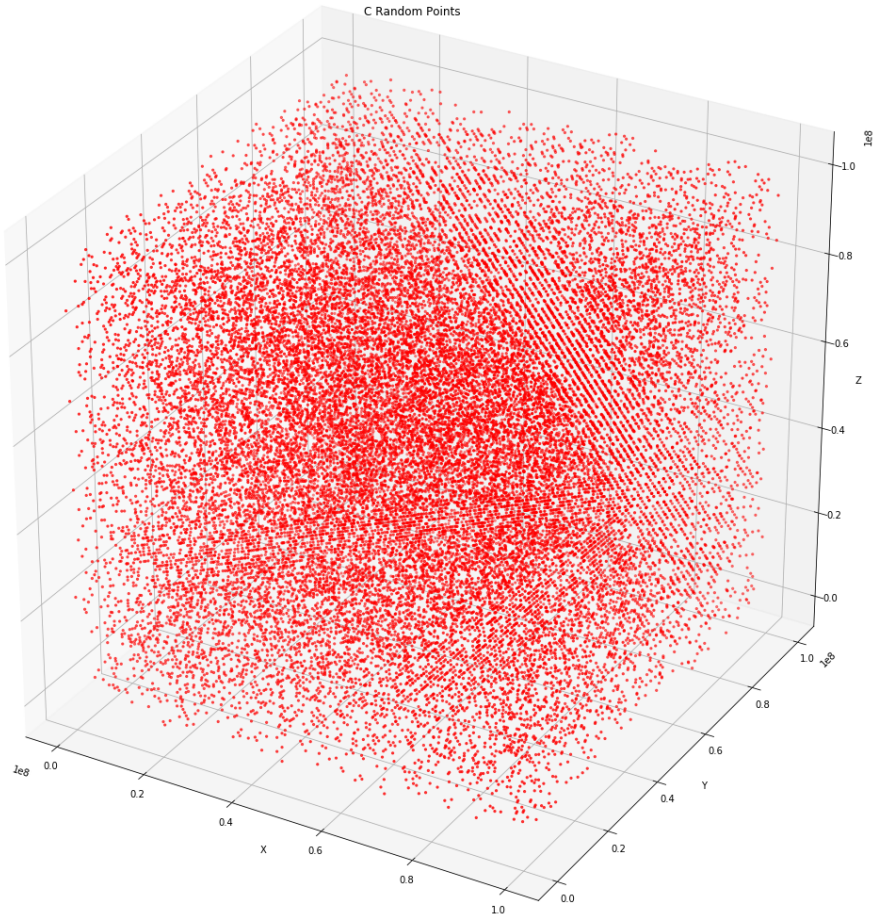
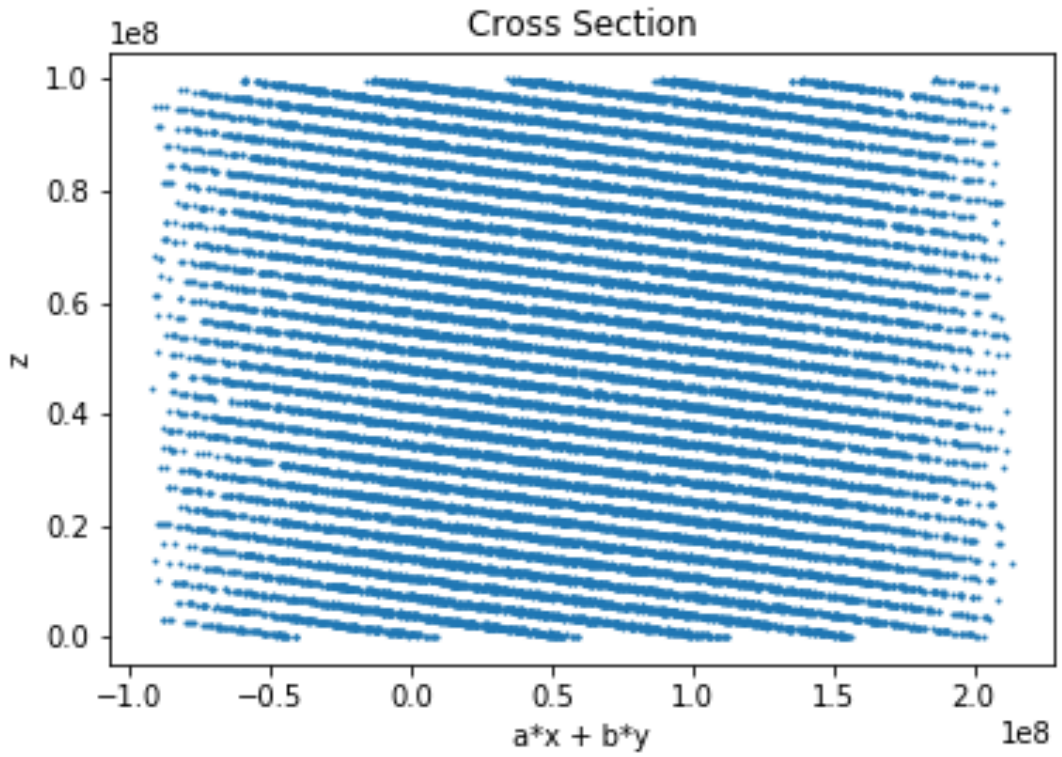
PSET 7 Computational Physics 512

Problem 1)

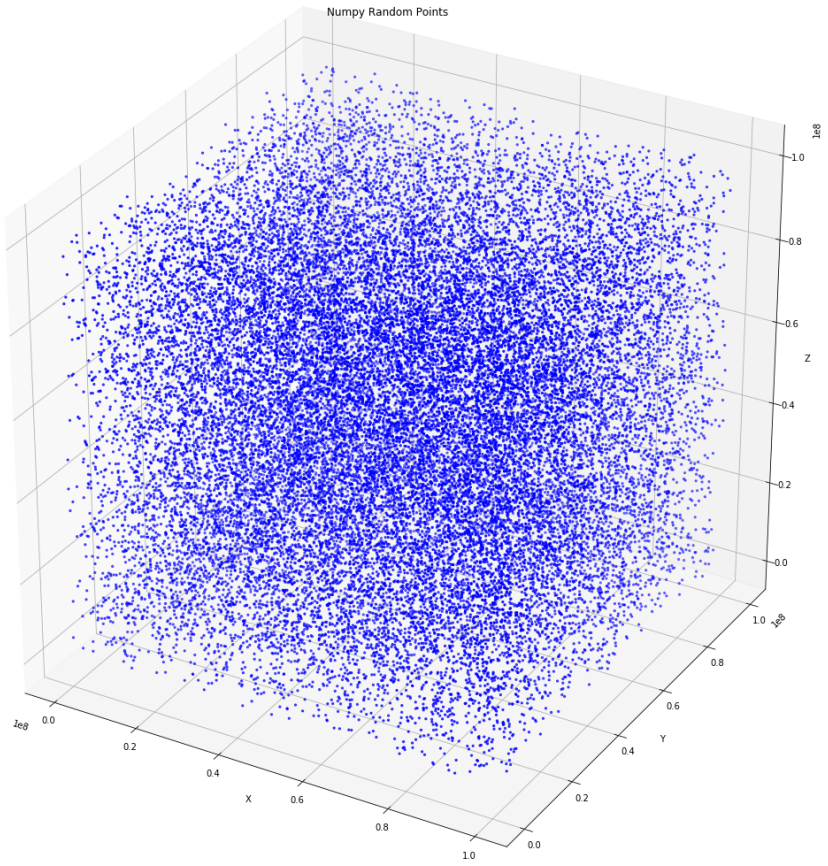


Scattered from the provided ‘random’ data, we can clearly see the evidence of some correlation between our variables as seen in the line segments viewed at different viewing angles. These can be alternated by playing with the elev and azim…



After Trying a multitude of different cross-sections, I picked this one to clearly illustrate the correlation between z and a*x + b*y these values for a and b respectively:

{2.1318438704144382, -0.9297303705485978}

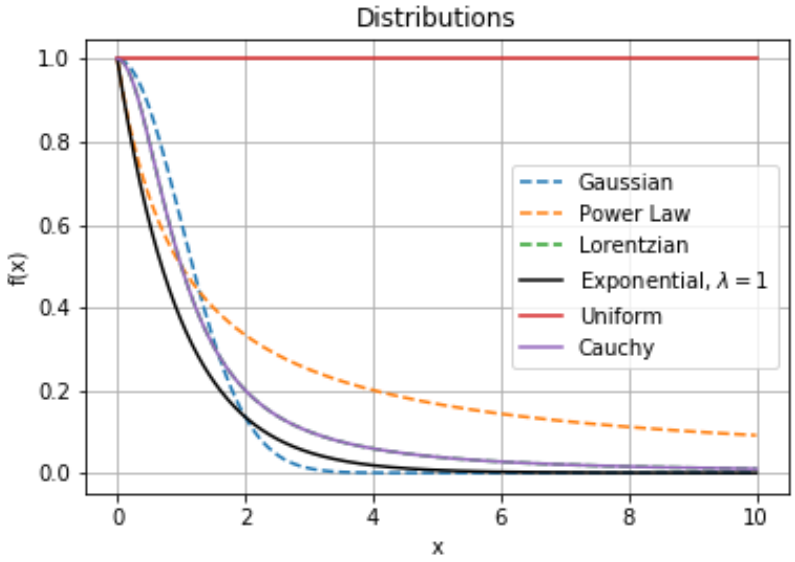


As for the random numbers generated by NumPy's algorithm, no such correlation could be found. Playing with various a and b cross-sectional angles would not generate any evident lines either.

All code for Q1 is in PSET7_Q1.py

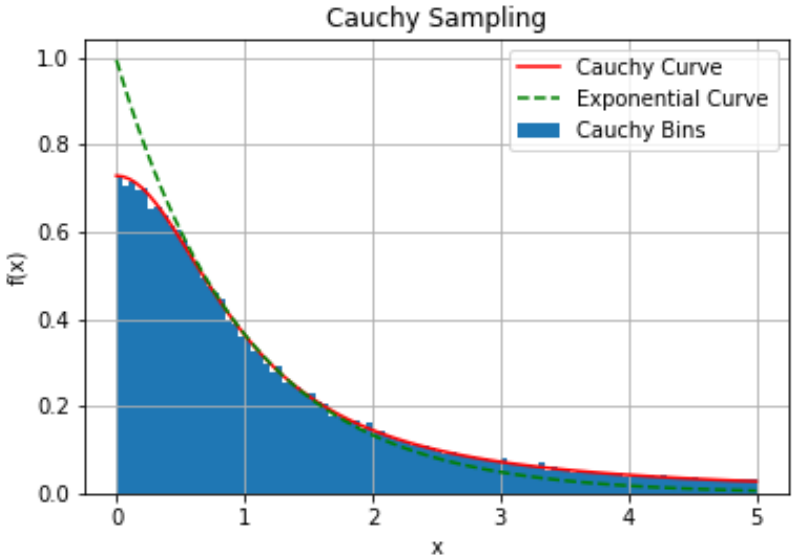
Problem 2)

We shall use the distribution most appropriate; the Cauchy curve is clearly most accustomed to following the exponential over our domain.



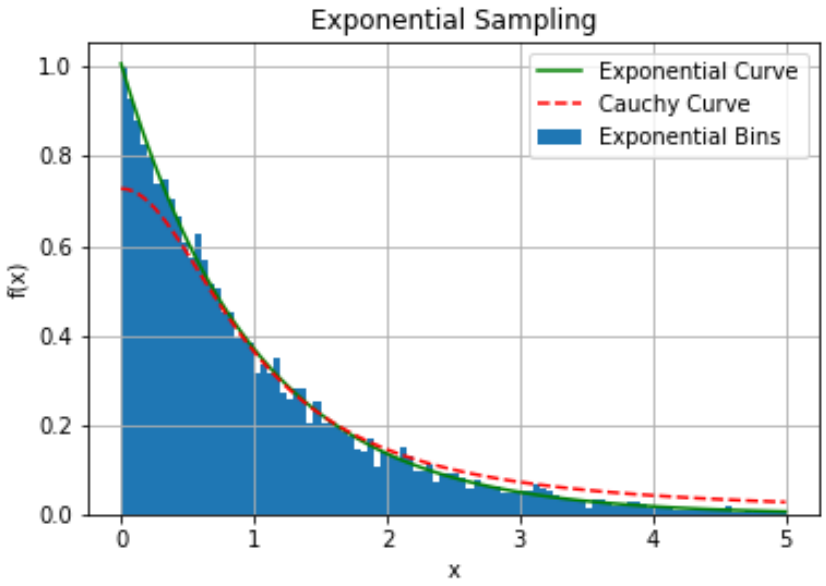
Here, we see a Cauchy distribution (with a 100 percent acceptance rate) under our Cauchy curve…

Acceptance Rate: 1.0



We are then interested in sampling distribution to fit our exponential curve, rather than our Cauchy curve. To do so, we simply pass the earlier Cauchy distribution through a rejection filter and scale to our exponential. This results in a much lower acceptance rate as shown below.

Acceptance Rate: 0.14647



All code for Q2 is in PSET7_Q2+3.py

```
#We must use a rejection method...
def CauchyRando(tail):

    rando = np.random.rand(2)*tail
    rando[0] = rando[0]

    if rando[1] < CauchyCurve(rando[0]) * np.pi/tail:
        return rando[0]
    else:
        return CauchyRando(tail)

def ExpRando(tail):

    randoCauch = np.array([CauchyRando(tail) for i in range(int(1e5))])
    randoNum = np.random.rand(int(1e5))*tail

    approved = randoNum < np.exp(-randoCauch) / CauchyCurve(randoCauch)

    return randoCauch[approved]

```

Problem 3)

For this problem, we must look at a purely exponential distribution with its sampling about e^{-x}

```
47 u = np.random.rand(int(1e5))
48 v = np.random.rand(int(1e5))*2/np.e
49
50 approved = u <= np.sqrt(np.exp(-v/u))
```

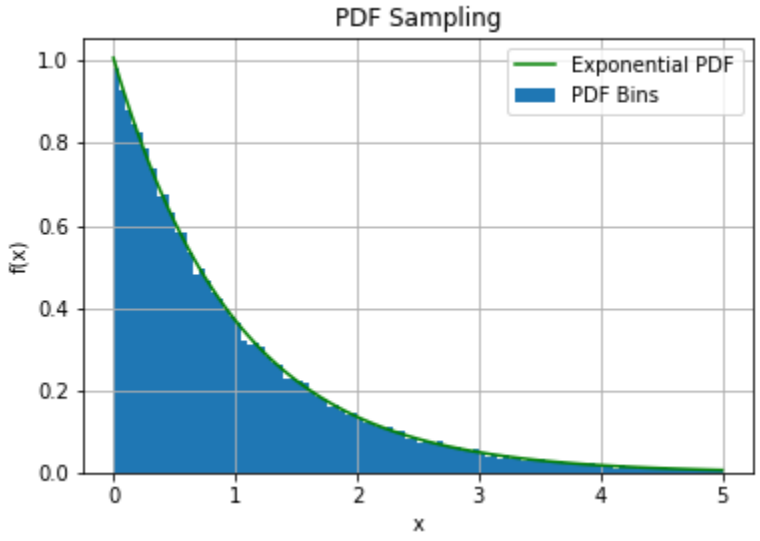
We have that $2\ ln\ (u)\ \leq\ -\frac{v}{u}$ <=> $0\ \leq\ v\ \leq\ -\ u\ ln(u^2)$

=> $u\ =\ 1/e$

Consequently, $v\ \leq\ 2\ /\ e$

This leads to roughly a third of samples being rejected:

Acceptance Rate of PDF: 0.67927



All code for Q3 is in PSET7_Q2+3.py