Ceren Çağlayan

270201059

CENG 222 Pseudo-Random Number Sampling Report

In homework, we sampled numbers randomly as a simulation (for 50000 times.) with using Inverse Transformation Method(in part a)n and Rejection Method (in part b).

Part a)

Using inverse transform method; we calculated our sample as following algorithm:

1- Generating a random variable [0,1] uniformly. (stage 1)
2- Find inversion of CDF and solve our variable with in it; and return that value.(stage 2)

With this, we have 3 figures (also +2 mean and variance):

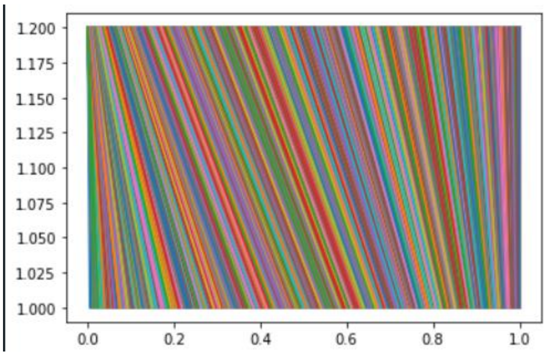
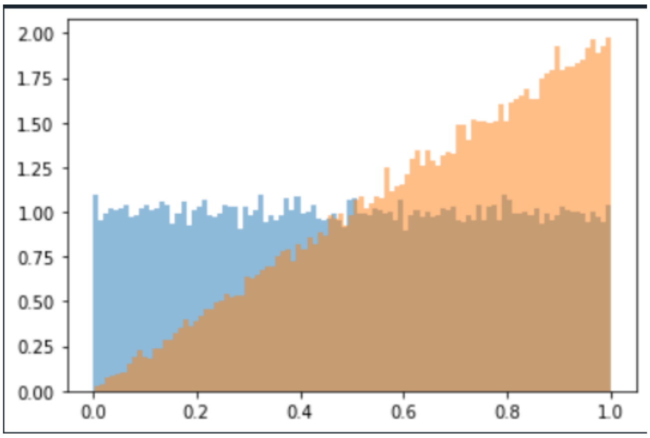


Figure 1:

Explanation:

In code, we plot each U value(values that we find in stage 1 in inverse transform algorithm as generating random variables , u) and Xa value (values that we find in stage 2 in inverse transform algorithm as inverse of cdf's inside : √u)

Figure 2:



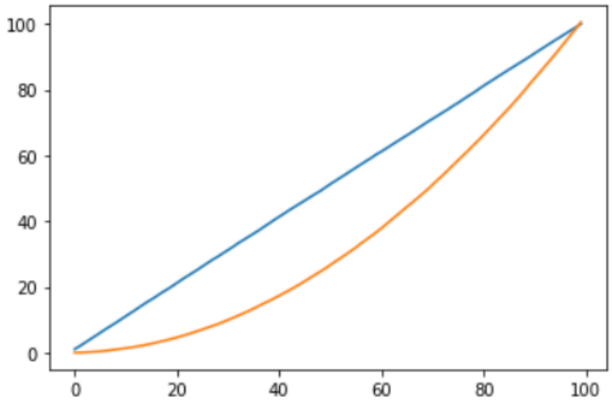
Explanation:

In Figure 2, blue area represents uniformly distributed values.

The orange area is density function f(x) between 0 and 1.

In this homework, given cdf is square of x with interval [0,1]. So that our pdf(orange area) is 2*x (as linear approach) As you can see, when x=1, y = 2.)

Figure 3:



Explanation:

In Figure 3, we are representing cumulative sums.

Orange line is our given cdf in question, which is:

F(x) = x^2

Blue line is F(u):

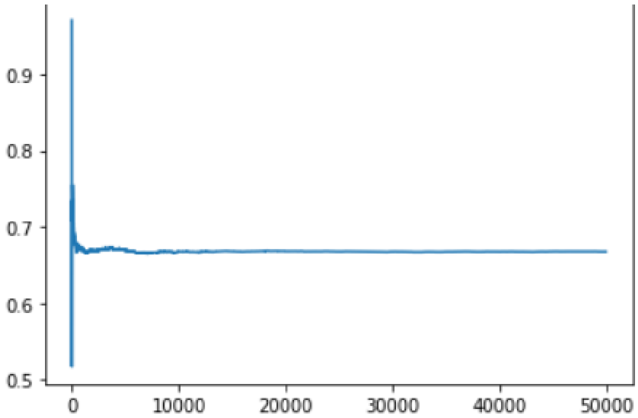
In inverse transformation method, F(u) equals to F(F^-1(u)) , which is :

$(\sqrt{x})^2 \rightarrow$ x.

(As a reminder : We already know in mathematics : fof^{-1} => y=x [unit function])

- After these, we need to calculate mean and variance values for our experiment:

Figure 4:



Explanation: In figure 4, we are represented average(mean) of Xa.

How I calculated in code?

For every 50.000 sample; I calculated 50.000 times mean of Xa.

Because of we do not need any extra conditions in Xa; we can add our mean value each time we change Xa, which means when we added new value in our for loop.Since we doing it for 50.000 times; we can found our value.

In graph , value converges value of 0.6668326610282266 since amount of samples results almost same result.

How I calculated with formulas(manually)?

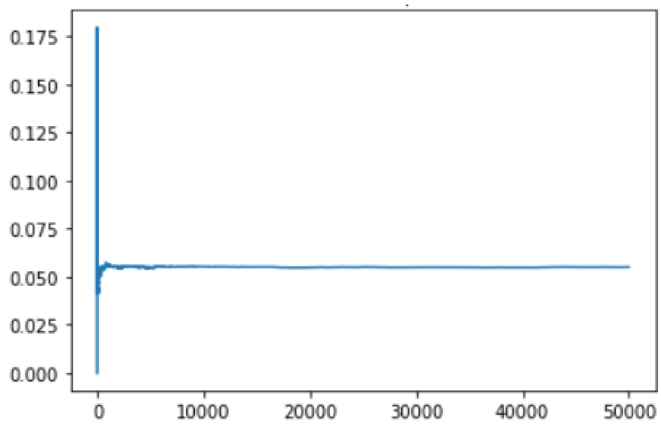
μ=E[X]=∫x*f(x)dx in interval[∞,-∞].

We got that our f(x) as 2x (derivative of our cdf F(x)).

Putting on these : E[X]=$\int_0^1 2\,x^2$dx (in given) → $|2 * \frac{x^3}{3}$

2 * ($(1^3/_3)$ - 0)) = 0.66..7 is the result manually *(which is close to code).* .

Figure 5:



Explanation: In figure 5, we calculated variance of Xa.

How I calculated in code?

For every 50.000 sample; I calculated 50.000 times mean of Xa.

In graph, value converges value of

0.05520597289393212688 since number of samples results almost same result.

How I calculated with formulas(manually)?

Var(X)=E[(X−μ)2]=$\int (x-\mu)^2$ f(x)dx interval[∞,-∞].

We got that our f(x) as 2x and our mean is 0.666..7 .

Putting on these →

$\int_0^1((x-0.66)^2 * 2 * x)$ = 0.0556 - 0 = 0.0556

We get the variance value is 0.0556 manually *(which is close to code).*

Part b)

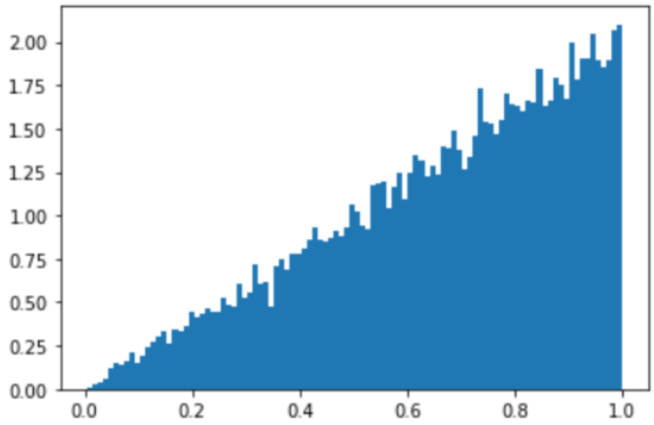
Using rejection method; we calculated our sample as following algorithm:

1. Find such numbers a, b, and c that 0 ≤ f(x) ≤ c for a ≤ x ≤ b.
2. Generate Standard Uniform random variables U and V (in this homework, we use np.random.rand() for both of them.)
3. Define X = a + (b − a)U and Y = cV . Then X has Uniform (a, b) distribution, Y is Uniform(0, c), and the point (X, Y) is Uniformly distributed in the bounding box.

In experiment, I take a = 0, b = 1 and c = 3.

With calculations, we have 3 figures (also +2 mean and variance):

Figure 6:



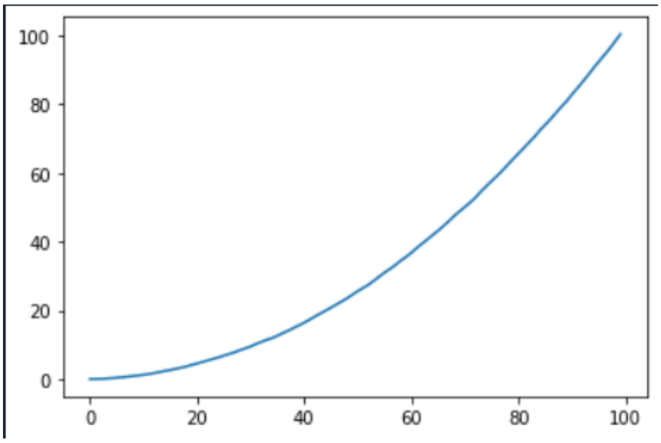
Explanation:

In Figure 6, we are representing our pdf function, (f(x) = 2*x).

Since in the algorithm we added accepted values in our Xb(list in our code), we get histogram like this.

Note: This is the same histogram *in Figure 2 (orange area).* We also got that in separate algorithm here.

Figure 7:



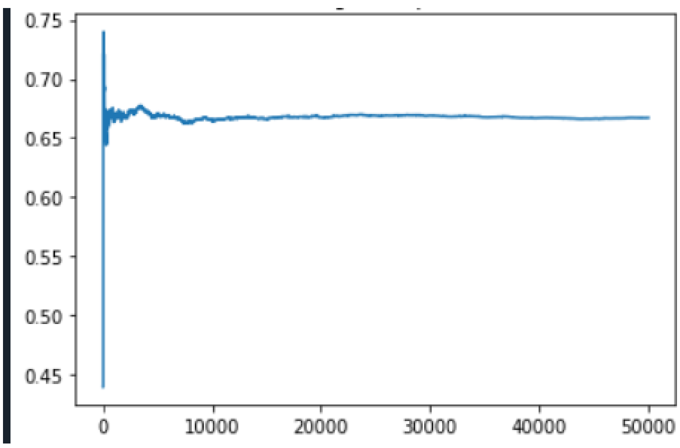
Explanation:

In Figure 7, we are showing our cdf function, as given:

F(x) = x^2

Note: This is the same histogram *in Figure 3 (orange line).* We also got that in separate algorithm here.

Figure 8:



Explanation:

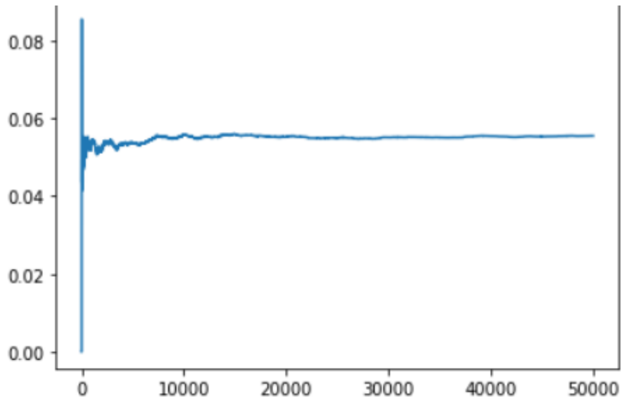
In figure 8, we are representing our mean data.

How I calculated in code?

For every 50.000 sample ; I calculated 50.000 times mean of Xb.I also controlled if Xb is empty or not for wrong calculations(which also means len(Xb) might be less than 50.000 for rejection method since we looking with some conditions).

In graph, value converges 0.6702556943834688 since number of samples results almost same result In *Figure 4*.

Figure 9:



Explanation:

In figure 9, we are representing our variance data.

How I calculated in code?

For every 50.000 sample; I calculated 50.000 times variance of Xb. I also controlled if Xb is empty or not for wrong calculations (which also means len(Xb) might be less than 50.000 for rejection method since we looking with some conditions). (Same as calculating mean value.)

In graph, value converges 0.05495806688608861 since number of samples results almost same result In *Figure 5*.

Comparing part a and part b:

We analysed our samples two separate ways in part a and part b, even though our scope is different for both; our results are almost same:

-In Mean and Variances are almost same at the end of the experiments (for mean values, *Figure 5* and *Figure 9;* for average values *Figure 4* and *Figure 8*).

-We both got our pdf and cdf graphs and functions *(as an example : Figure 6 & Figure 2 and Figure 3(orange line) and Figure 7).*