EE 511 - PROJECT 7

Tamoghna Chattopadhyay

1. INTRODUCTION

In this project, we look at generating continuous Random variables using three different methods. We compare the time taken in each method and then output the results.

2. THEORY

(a) Let $Y \sim U[0,1]$
This implies, $P(Y \le y) = y$

Now, if $X = -\ln(1-Y)$
This implies that,

$$
\begin{aligned}
& P(X \le x) \\
= \; & P(-\ln(1-Y) \le x) \\
= \; & P(1-Y \geq e^{-x}) \\
= \; & P(Y \le 1-e^{-x}) \\
= \; & 1-e^{-x}
\end{aligned}
$$

This is the cdf of the standard exponential with $\lambda = 1$.

Hence, if Y is uniformly distributed over $(0,1)$, then $X = -\ln(1-Y)$ will be exponentially distributed with parameter $\lambda = 1$.

- X_i are iid random exponentially distributed random variables with $\lambda = 1$. A random variable S_n is defined by $\sum_{i=0} X_i$. This turns out to be an Erlang distribution (n, λ).

ality pr

The expected value or mean of an exponential variable is given by $\frac{1}{\lambda} = 1$ in our case.

The pdf of an Erlang distribution is given by :

$$f_{S_n}(x) = \frac{(\lambda x)^{n-1} \lambda e^{-\lambda x}}{(n-1)!}$$

(c) A more computationally efficient way to generate samples for S_n uses :

$$S_n = \frac{-\ln\left(\prod_{i=1}^{n} U_i\right)}{\lambda}$$

where $U_i \sim \text{Uniform } (0,1)$

As, $X = -\ln(1-Y)$ $\quad Y \sim U[0,1]$

or $X = -\ln(Y)$

is exponentially distributed

and, $S_n = \sum_{i=1}^{n} X_i$

$$= \sum_{i=1}^{n} -\ln Y$$

$$= \frac{-\ln\left(\prod_{i=1}^{n} U_i\right)}{\lambda}$$

In our case $\lambda = 1$

Hence, this works too.

3. SIMULATION METHODOLOGY

(a) (b) Using the rand function in MATLAB, uniformly distributed random numbers between 0 and 1 were generated ($N = 1000$). An exponential function with $\lambda = 1$ can be generated using uniform distribution with $y = -\frac{1}{\lambda} \ln(x)$.

Here, $y = -\ln x$ as $\lambda = 1$.
Several values of X_i are generated and these values are used to generate a series of samples for

$$S_n = \sum_{i=0}^{n} X_i$$

Analytically,
for $n = 3$, mean $= \frac{n}{\lambda} = 3$

variance $= \frac{n}{\lambda^2} = 3$

In the simulation, values for mean and variance are approximated and determined. Using samples generated, a histogram plot is made for pdf $f_{S_n}(x)$. Using tic toc function, the time taken to generate the samples is measured.

(c) In part c, uniform random variables are generated using rand function in MATLAB and these numbers are used to generate samples for $S_n = (\ln \prod_{i=1}^{n} U_i)/\lambda$. Here too, analytically, for $n = 3$, mean $= 3$ and variance $= 3$. Using samples generated, a histogram plot is made for the pdf. Using tic toc function, the time taken to generate the samples is measured.

- In the Rejection Method,

we want to generate an RV, X with pdf $f_X(x)$. We have an ("efficient) method to generate an RV Y with pdf $g_Y(x)$ which is defined over the same range as X. The steps are:

1. Find (smallest) c, such that $\frac{f_X(x)}{g_Y(x)} \leq c$

   over the range of interest of x

2. Generate $Y \sim g_Y$, generate $U \sim$ uniform $(0,1)$

3. If $U \leq \frac{f_X(Y)}{c\,g_Y(Y)}$, then set $X = Y$ otherwise return to step 2.

In our case, $\lambda = 1$, $n = 3$.

So,

$$f(x) = \frac{x^2 e^{-x}}{2}$$

So,

$$f'(x) = -\frac{x^2 e^{-x}}{2} + x e^{-x}$$

$$= -\frac{1}{2} x e^{-x} (x-2)$$

$$= 0$$

$$\therefore \quad x = 0, 2$$

$f(2) = 2e^{-2} = 0.27067$

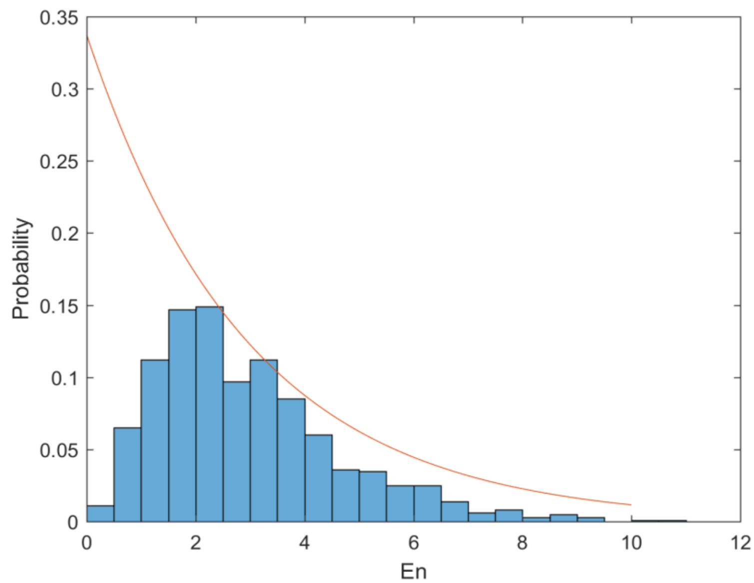
$f(0) = 0$

So, $c = 0.2707$

For part d, I used the algorithm as mentioned in the theory.

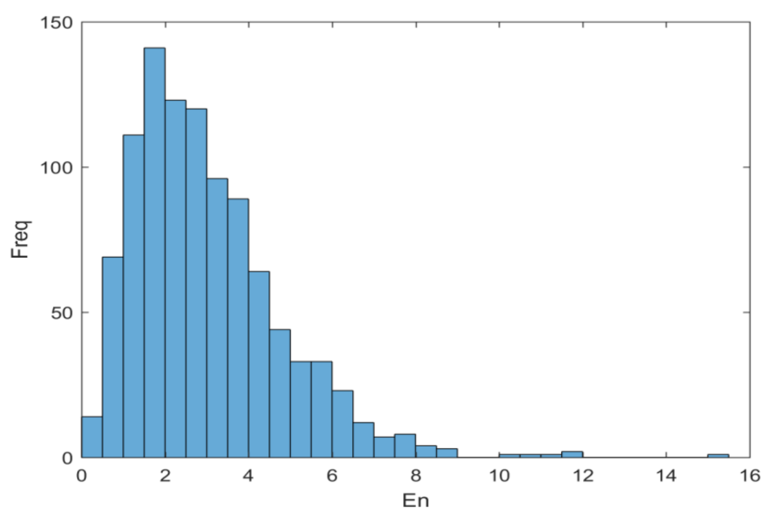
4. <u>Results and Observations</u>

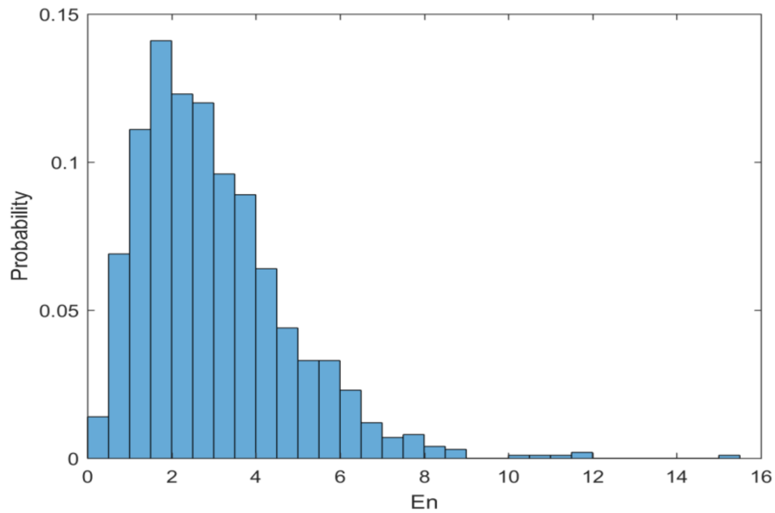
Comparing the empirical distributions in part b,

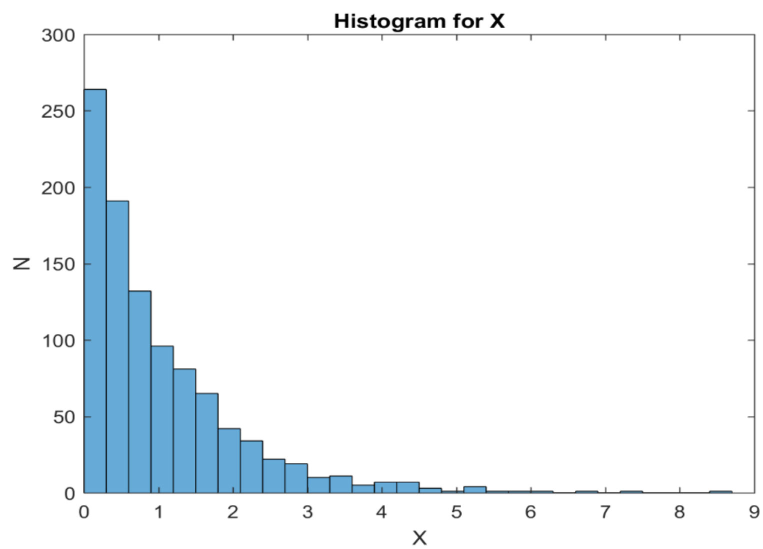


For part c,

n = 3
Expected Value = 2.9985
Variance = 3.1900
Elapsed Time = 0.843689 seconds
While analytically, the mean and variance should have been 3, but after simulation the values are as mentioned above. I expect the values to get closer as n increases. The time is calculated using tic toc function. The histogram shows approximations for the pdf for value n = 3, Erlang-3 distribution.



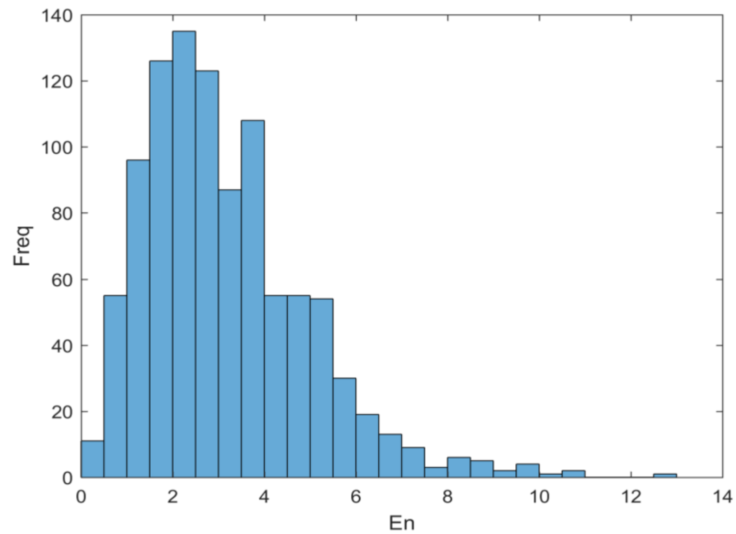




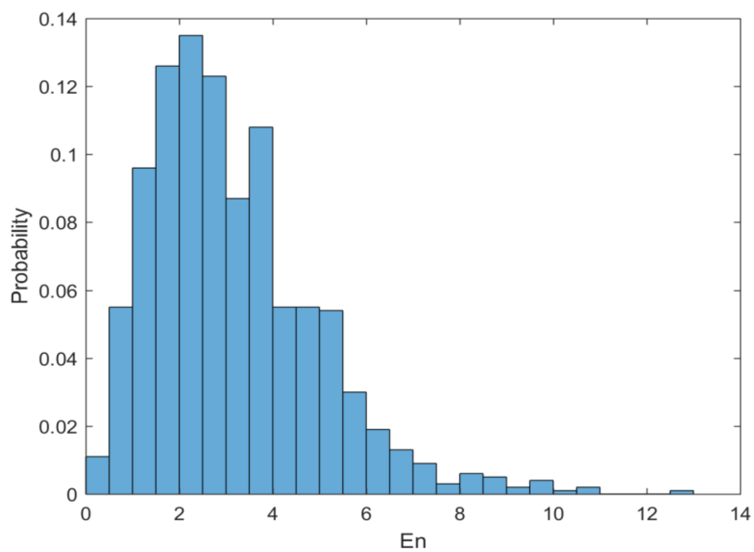
For part d,

n = 3
Expected Value = 3.1508
Variance = 3.1909
Elapsed Time = 0.665127 seconds

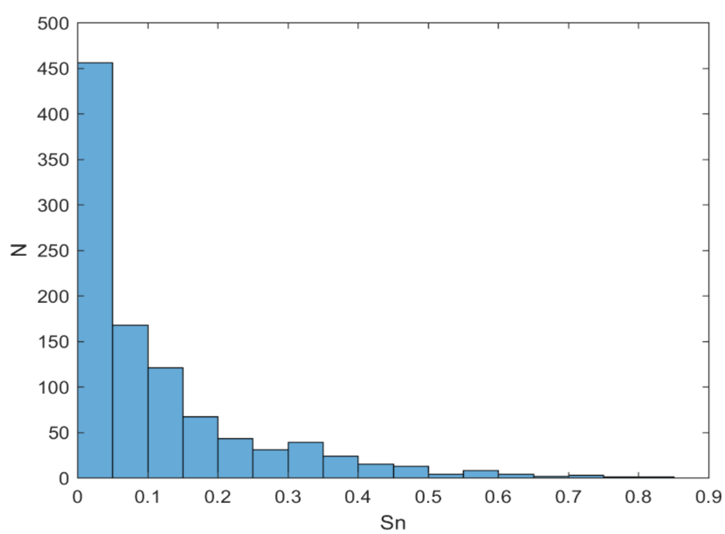
While analytically, the mean and variance should have been 3, but after simulation the values are as mentioned above. I expect the values to get closer as n increases. The time is calculated using tic toc function. The histogram shows approximations for the pdf for value n = 3. It represents Erlang-3 distribution. As we can see, the second method is thus computationally more efficient and takes lesser time to generate the samples.



Freq
En



Probability
En



N
Sn

For part e,

n = 3
Expected Value = 3.0271
Variance = 3.1387
Elapsed Time = 0.459213 seconds

While analytically, the mean and variance should have been 3, but after simulation the values are as mentioned above. I expect the values to get closer as n increases. The time is calculated using tic toc function. As we can see, the third method is thus computationally most efficient and takes least time to generate the samples.

5. Codes

b)

```
clc;
clear all;

n = 3;
l = 1;
E = zeros(1,1000);

for i = 1:n

    Y = rand(1,1000);
    X = -log(Y);
    E = E + X;

end

M = mean(E);
V = var(E);

figure
histogram(E,'normalization','prob')
xlabel('En');
ylabel('Probability');
hold on
x = 0:0.1:10;
y = exppdf(x,M);
plot(x,y);
```

c)

```
clc;
clear all;

tic
n = 3;
l = 1;
```

```
E = zeros(1,1000);
S = zeros(1,1000);

for i = 1:n

    Y = rand(1,1000);
    X = -log(Y);
    E = E + X;
    pdf = (l*E).^(n-1).*exp(-E*l)*l/(factorial(n-1));
end

S = S + (E/i);

figure
histogram(E)
xlabel('En');
ylabel('Freq');
figure
histogram(E,'normalization','prob')
xlabel('En');
ylabel('Probability');
figure
histogram(X)
title('Histogram for X');
xlabel('X');
ylabel('N');

M = mean(E);
V = var(E);

toc
```

d)

```
clc;
clear all;

tic
n = 3;
l = 1;
S = ones(1,1000);

for i = 1:n
    Y = rand(1,1000);
    S = S.*Y;
end

E = -log(S);

figure
histogram(E)
xlabel('En');
ylabel('Freq');
figure
histogram(E,'normalization','prob')
xlabel('En');
```

```
ylabel('Probability');
figure
histogram(S)
xlabel('Sn');
ylabel('N');

M = mean(E);
V = var(E);
toc
```

e)

```
clc;
clear all;

tic
c = 0.27067;
E = zeros(1,1000);
count = 1;

while count<1001

    U = rand;
    Y = rand;
    H = Y.^2.*exp(-Y)/(c*2);

    if (U <= H)
        E(count) = Y;
        count = count+1;
    end
end

figure
histogram(E)
xlabel('En');
ylabel('Freq');

M = mean(E);
V = var(E);

toc
```