SOLUTİONS:

Question 1:

1.a Binomial distribution

Generate 1000 Binomial(40,0.7) random variables and compare theoretical, simulated results:

In this question first step was generating 1000 random variables in MATLAB. Webcan generate *Binomial(n,p)* by adding n independent *Bernoulli(p)* random variables. Below, there is a MATLAB code about generating binomial random variables.

```
n = 40;
p=0.7;
for i=1:1000
    U=rand(n,1);
   x(i)=sum(U<p);
end
```

Generating Binomial random varible

Then, we plot the histogram of these random variables with their probabilities (simulated result) and we compute the analytical results of these random variables with the formula has given below with MATLAB code and we plot the pmf of (x,p(x)).

```
for i=1:1000
   p2(i)= nchoosek(40,x(i))*power(0.7,x(i))*power(0.3,(40-x(i)));
end
```

Computing analytical results with the formula is: p(x)=C(n,x)*p^x*q^(n-x)

The variance and mean of this binomial distribution have given below.

The mean:

E(x)= p+p+..+p= n*p= 40*0.7=28

Sample mean:

Mean=27.92

The variance:

V(x)= pq+pq+..+pq=n*pq=40*0.7*0.3=8.4

Sample variance:

Variance=8.1838

In Figure 1, there is a result of these two calculation. According to the figure we can say that the analytic and simulated results are similar, there is no big difference between them.

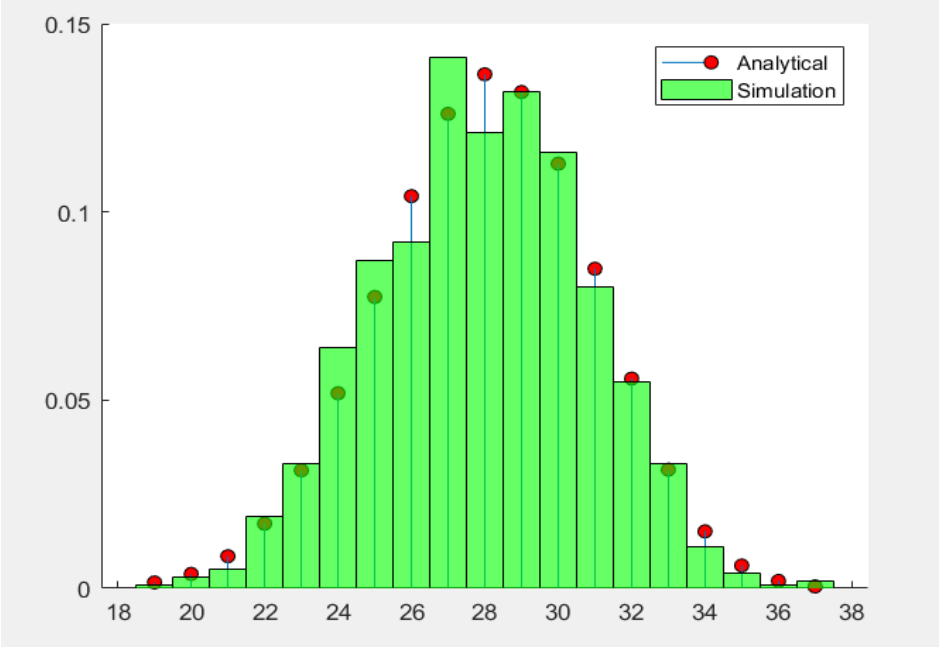


Figure 1: Comparing analytic and simulated results

Generate 1000 Binomial(50,0.5) random variable and compare theoretical, simulated results:

In this problem, the calculations which have described above has done with n=50 and p=0.5 and the results are below. In Figure 2, we can see the differnce betwwen these two results.

The sample mean:
Mean= 24.8630
The sample variance:
Variance=12.1744

The mean and variance values of these binomial distribution

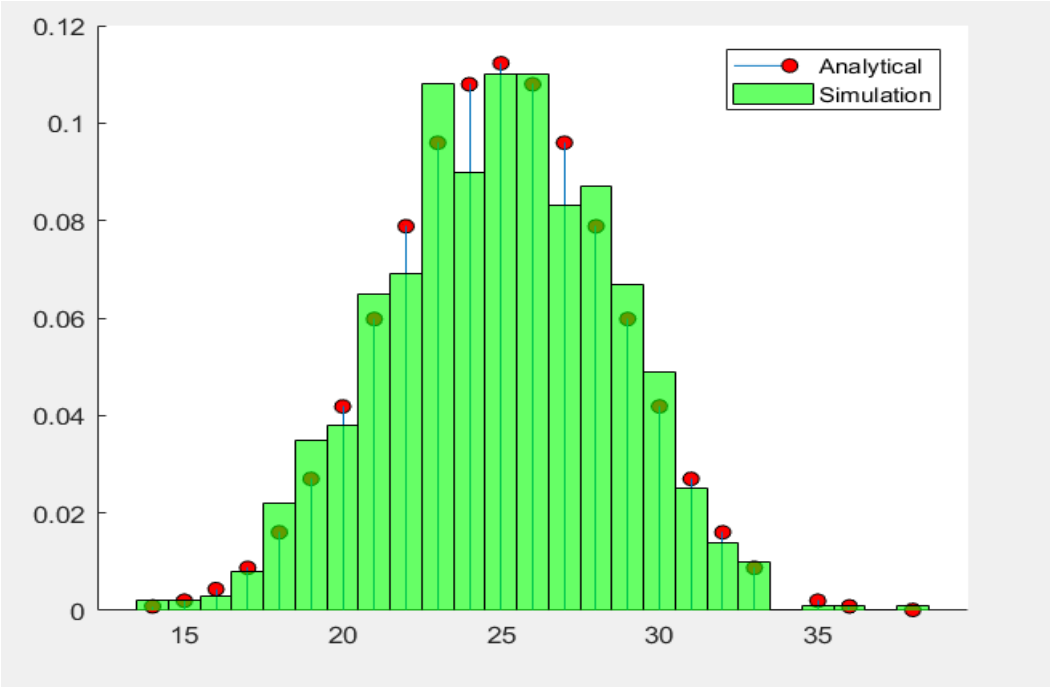
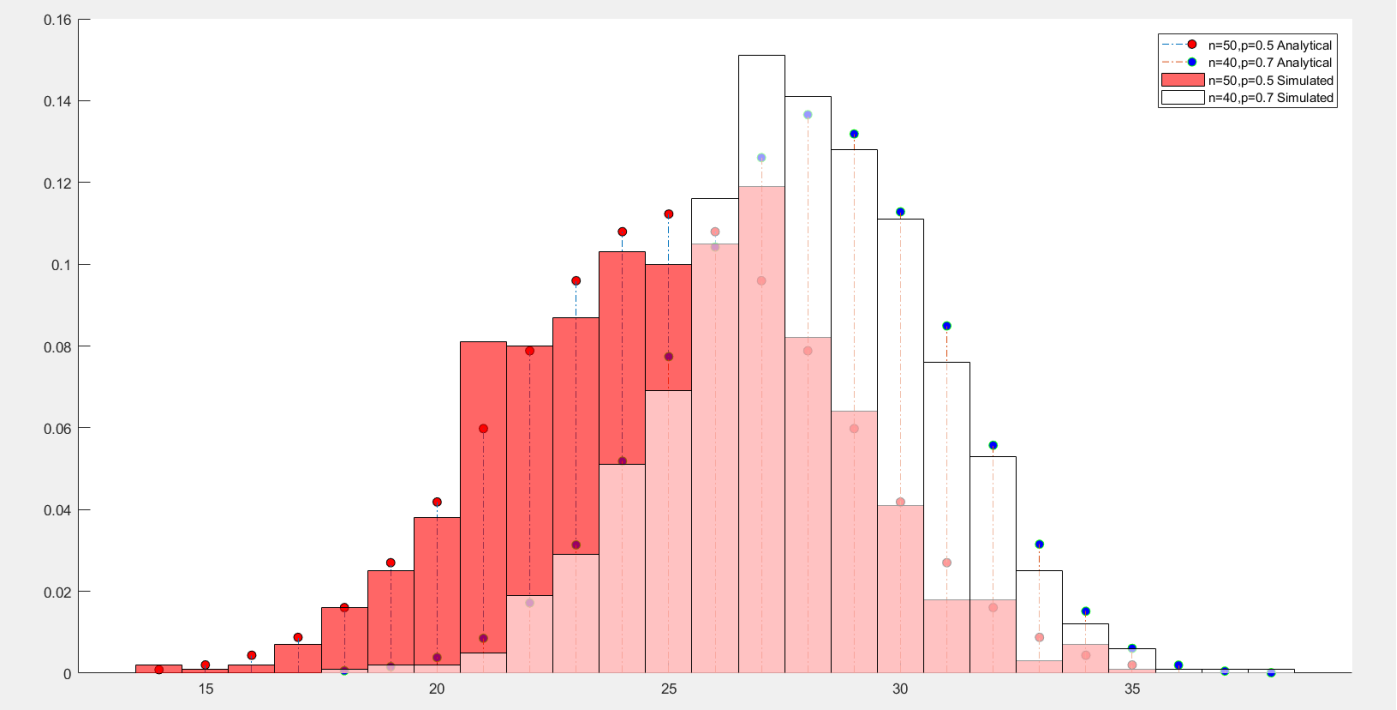


Figure 2: Comparing analytic and simulated results(n=50,p=0.5)

Comparing n=50,p=0.5 and n=40,p=0.7:



1.b Geometric distribution

Generate 1000 Geometric(0.5) random variable and compare theoretical, simulated results:

For generating geometric random variables, there is a loop of Bernoulli trials until the first success occurs. C counts the number of failures plus one success, which is equal to the total number of trials.

```
p_geo=0.5;
for i=1:1000
   C=1;
   while(rand()>p_geo)
      C=C+1;
   end
   geo_1(i)=C;
end
```

Generating geometric random variable

This random variables has used for plotting emprical pdf and for calculating analytical formula of geometric distribution and we plot pmf of (x,p(x)).

```
for i=1:1000
   p_geo(i)=power(0.5,(geo_1(i)-1))*0.5 ;
end
```

Computing analytical results with the formula is: p(x)=p*q^(x-1)

The mean and varince of these geometric distribution has described below.

The mean:
E(x)= 1/p=1/0.5=2
Sample mean:
Mean=2.0830
The variance:
V(x)=q/p^2=0.5/0.5*0.5=2
Sample variance:
Variance=2.2323

Mean and varince values of this geometric distribution

In Figure 3, there is a comporison between theorical and simulated results of the problem with has described above. According to the figure we can say that these two results are similar with each other.

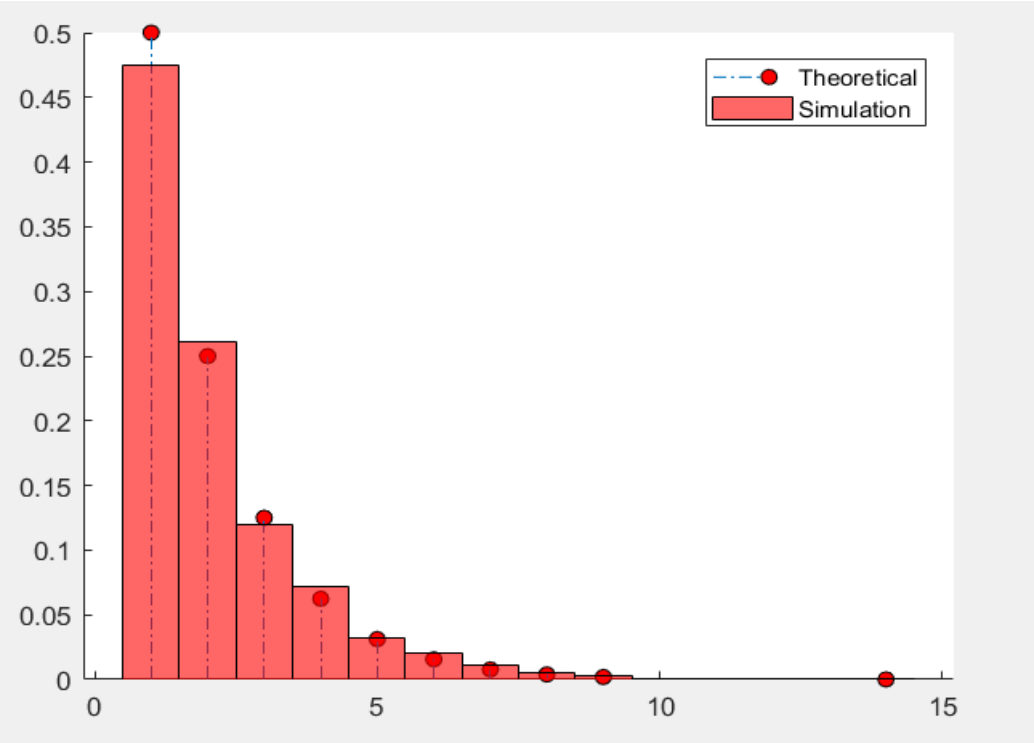


Figure 3: Comparing analytic and simulated results(p=0.5)

Generate 1000 Geometric(0.3) random variable and compare theoretical, simulated results:

The calculations are the same with Geometric(0.5) problem. The mean and variance values are below. In Figure 4, there is a comparison of analytical and simulated results.

Sample mean:
Mean= 3.2680
Sample variance:
Variance= 7.9822

Mean and varince values of this geometric distribution(p=0.3,q=0.7)

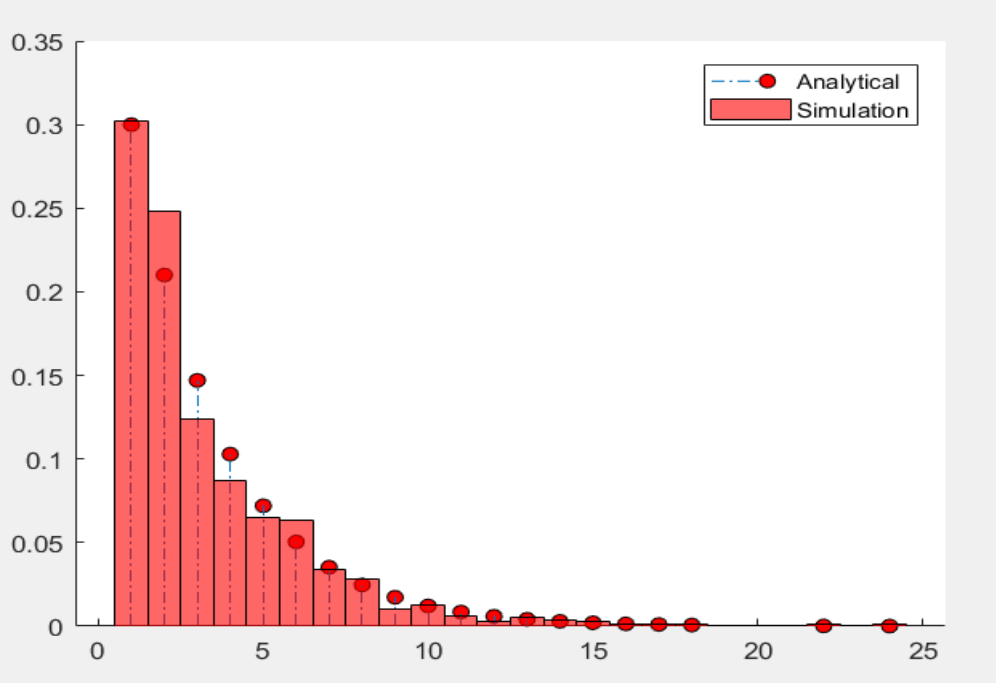
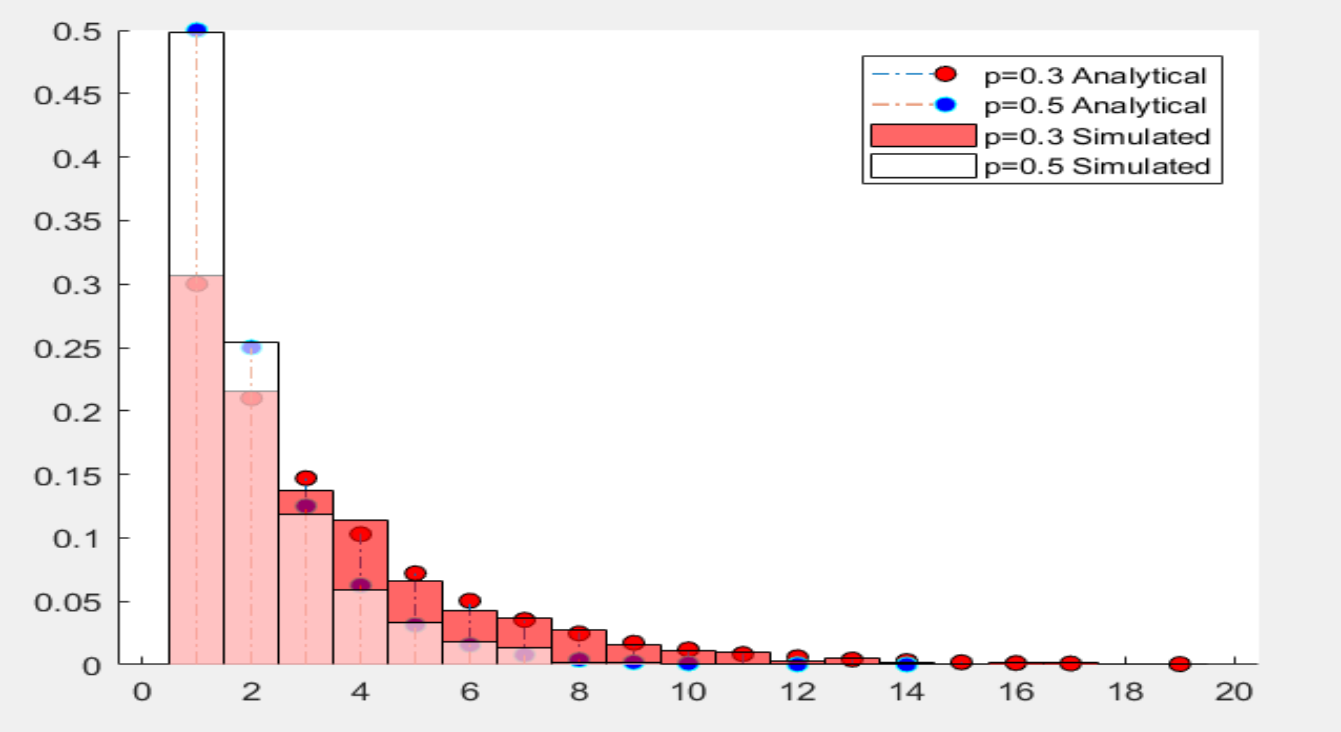


Figure 4: Comparing analytic and simulated results(p=0.3)

Comparing P=0.5 and p=0.3:



1.c Poisson distribution

Generate 1000 Poisson(6) random variable and compare theoretical, simulated results:

Generating poisson random variables has shown below with MATLAB code.

```
for j=1:1000
   i=0;
   u=rand();
   y=-(1/lambda)*log(u);
   sum=y;
   while(sum<=1)
      u=rand();
      y=-(1/lambda)*log(u);
      sum=sum+y;
      i=i+1;
   end
   poisson(j)=i;
end
```

Generating geometric random varibles[1]

These random variables used for emprical pdf and calculating pmf with the formula has described below.

```
for i=1:1000
   p_poiss_1(i)=exp(-lambda)*power(lambda,poisson(i))/factorial(poisson(i));
end
```

Calculation analytical formula in Matlab: P(x)=e^(-alpha)*alpha^x/factorial(x)

Below there is a mean and variance values of this distribution.

The mean and variance:
E(x)=V(x)=alpha=6
Sample mean= 5.9570
Sample variance=6.2574

The mean and variance of the poisson distribution

In Figure 5, there is a comparison of analytical and simulated results of teh poisson distribution with value of alpha is 6.

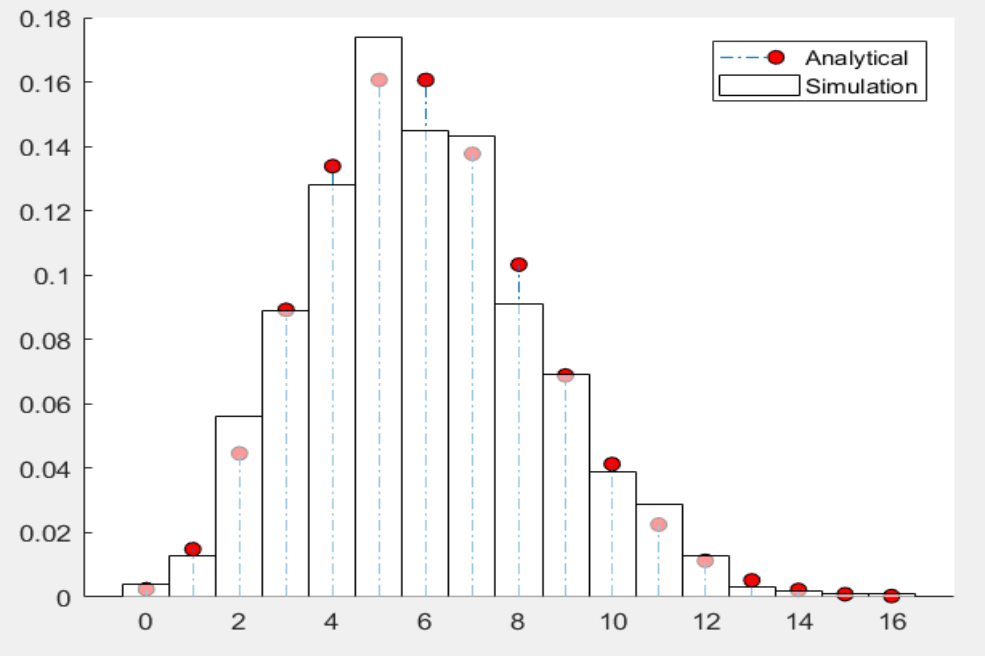


Figure 5: Comparing analytic and simulated results(alpha=6)

Generate 1000 Poisson(4.5) random variable and compare theoretical, simulated results:

Mean and variance values are below. And in Figure 6, there is a comparison between analytical and simulated results of the poisson distribution with alpha=4.5.

Sample mean =4.4720
Sample variance=4.5418

The mean and variance of the poisson distribution

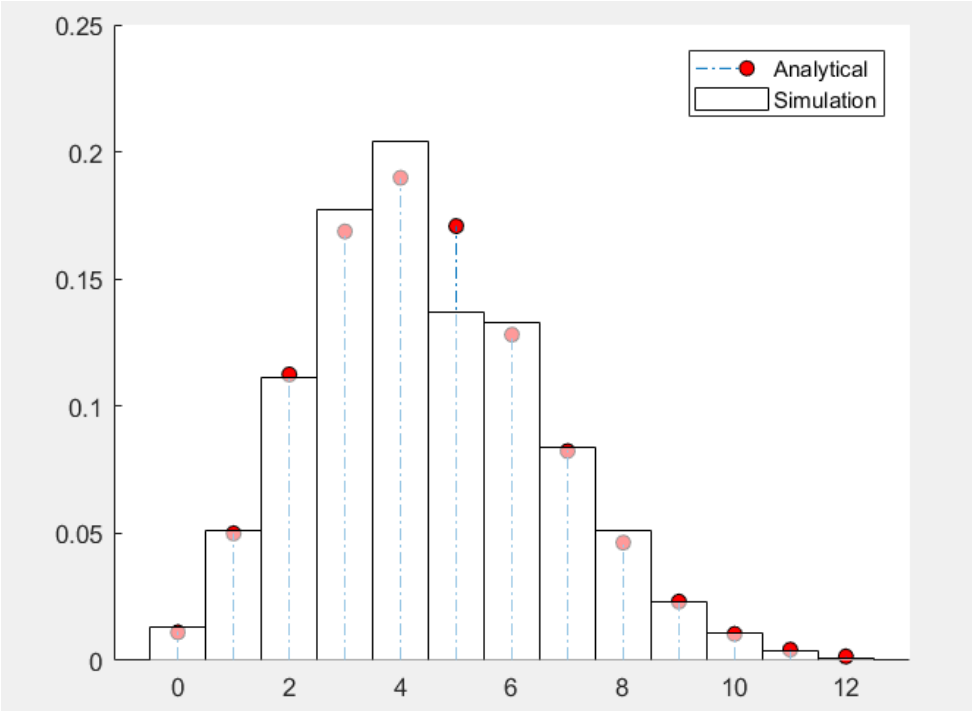
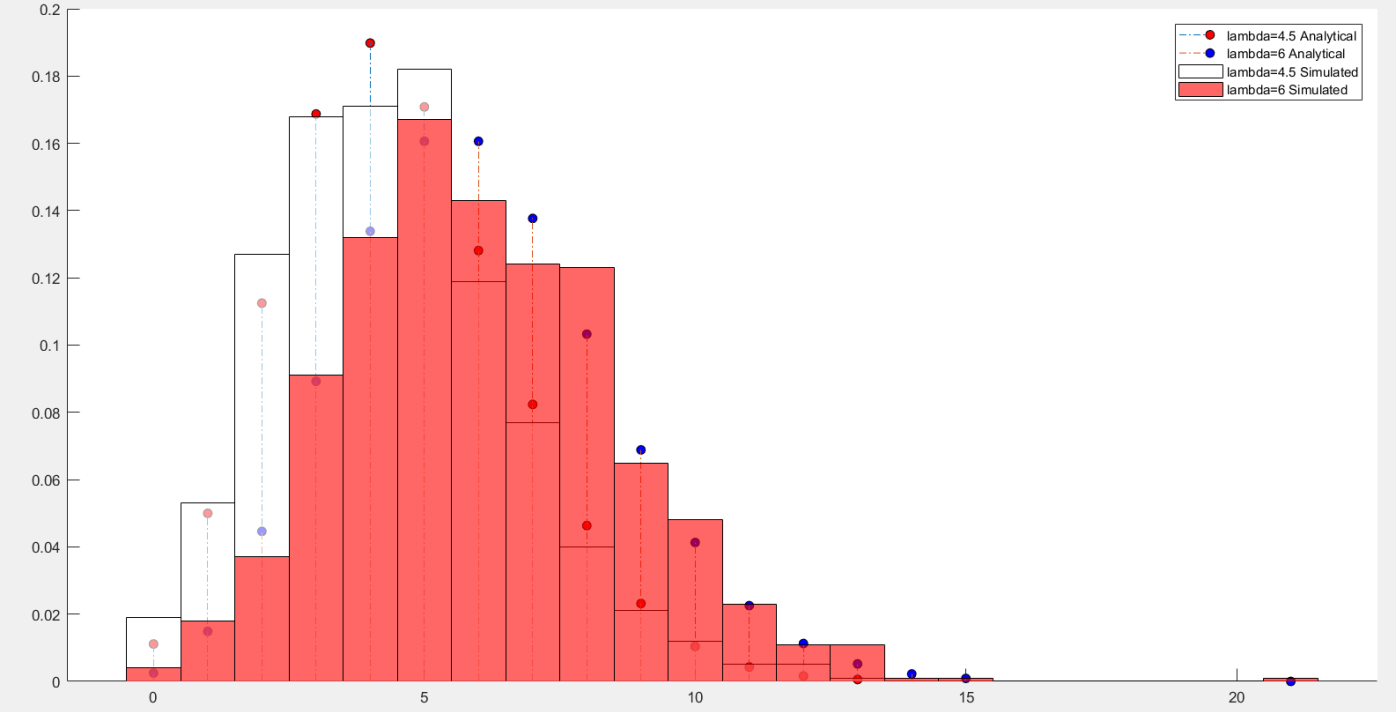


Figure 6: Comparing analytic and simulated results(alpha=4.5)

Comparing alpha=6 and alpha=4.5:



Question 2:

2.a Exponential Distribution

Lambda=1.2

For generating exponential random variable, we have used the formüle which is explained below:

F(x)= 1-e^x
U- Uniform(0,1)
X= F^-1(U) = -ln(1-U)

We can simplify this:
1-U – Uniform(0,1)
U+1-U=1
X=-ln(U)

The Matlab code of this generating exponential random variable is below.

```
lambda_exp=1.2;
for i=1:1000
    u_exp=rand;
    exp_1(i)=-log(u_exp)/lambda_exp;
end
```

Generating exponential random variables

These random variables are used for calculation analytic and simulated cdf's. Below mean and variance values of these sample has shown.

The mean:
E(x)=1/lambda=0,83
Sample mean: 0.82
The Variance:
V(x)=1/lambda^2=0,69
Sample variance: 0.65

For exponential distribution for calculation F(x) cdf below formula has used .

```
for i=1:1000
   p_exp(i)=1-exp(-lambda_exp*exp_1(i));
end
```

In Figure 7, theorical cdf and emprical cdf has shown. These two results are very close to each other.

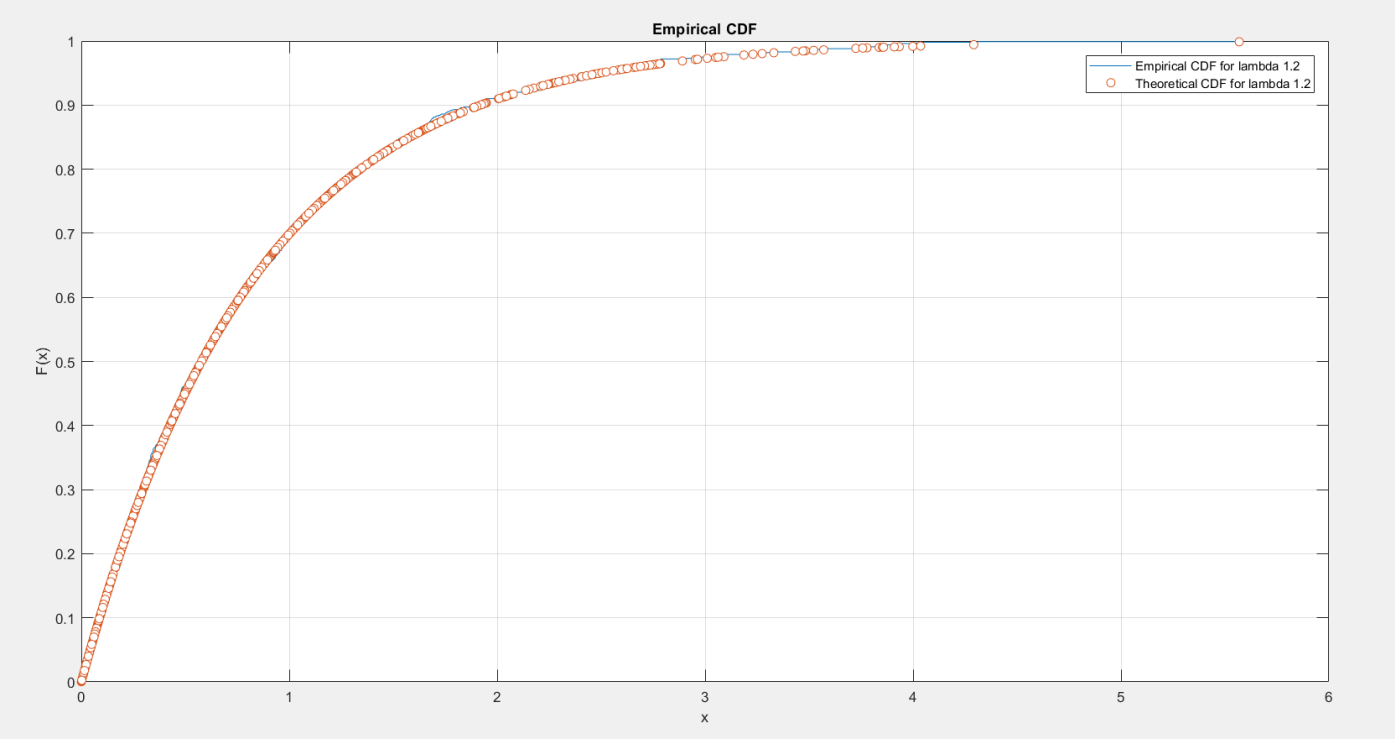


Figure 7: Comparing analytic and simulated results(lambda=1.2)

Lambda 2.1

The mean and variance values are below. In figure 8, comparison of results has shown.

Sample mean: 0.47
Sample variance: 0.20

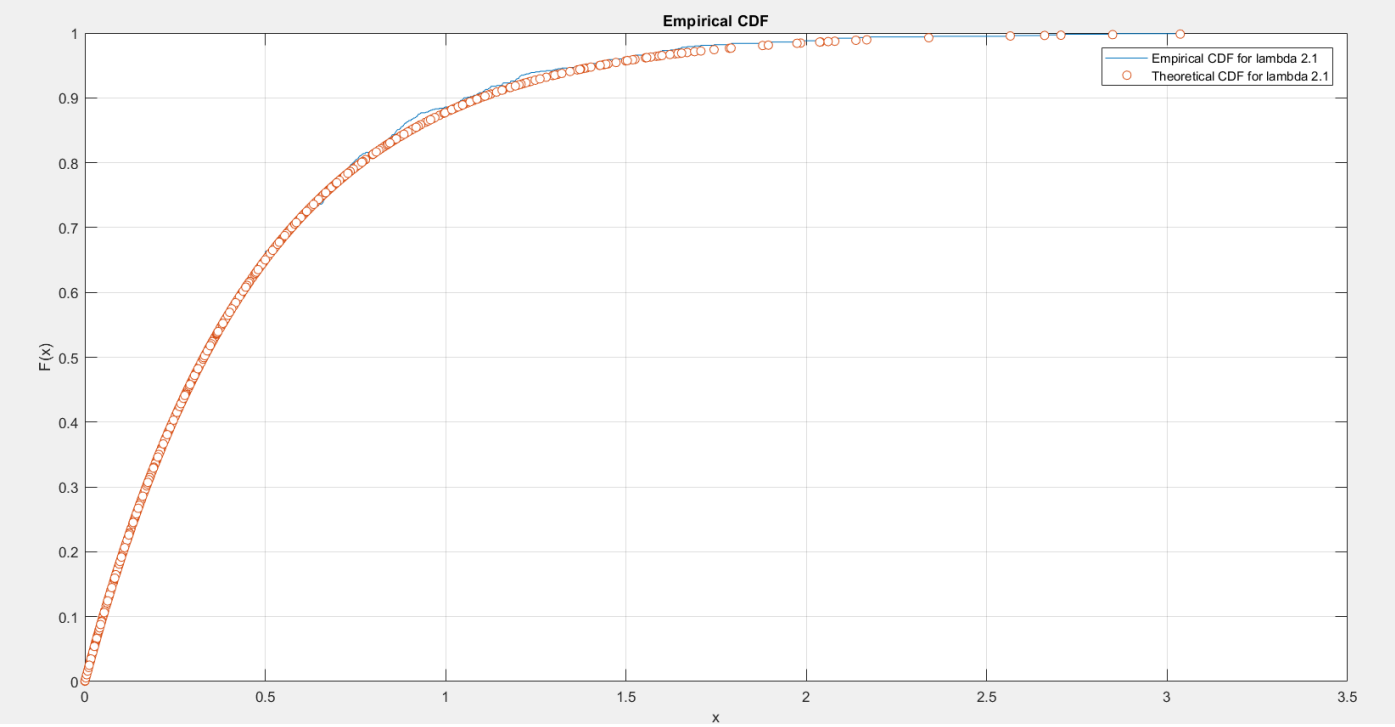
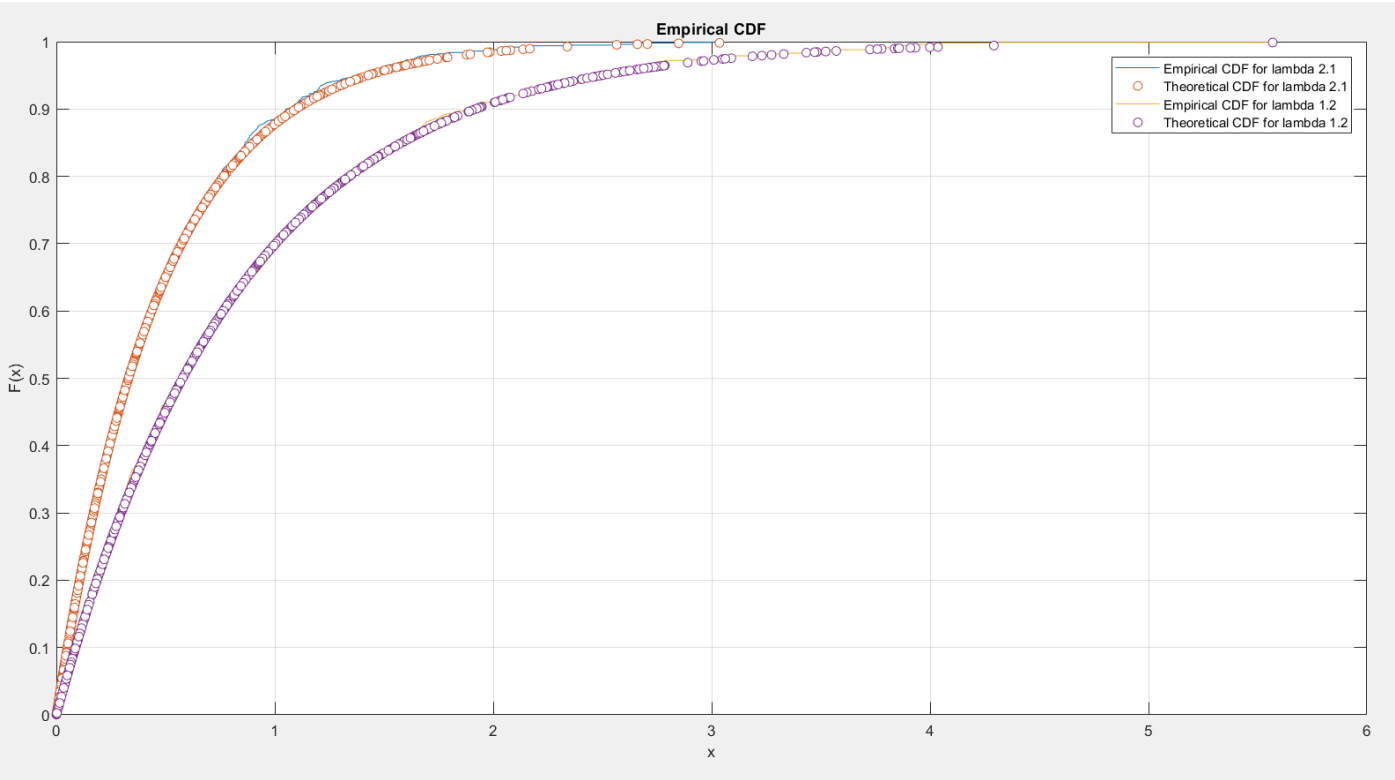


Figure 8: Comparing analytic and simulated results(lambda=2.1)

Lambda=1.2 vs Lambda=2.1: The theorical and emprical results are veru close.



2.b Normal Distribution

Generating normal distribution random variable has shown in below. ‘randn’ creates Uniform(0,1) and with sigma and mean values we can get the normal(nu,sigma^2) distribution.

```
sigma=sqrt(5);
nu=3.1;
norm_r= sigma*randn(1,1000)+nu;
```

For calculation of analytical results, normal distribution cdf formula has been used. The Matlab code in below is doing cdf calculation of normal distribution.

```
fun= @(x)exp((-power(x-nu,2)/(2*^power(sigma,2))));
for i=1:1000
  norm_cdf(i)=(1/(sigma*sqrt(2*pi)))*integral(fun,-inf,norm_r(i));
end;
```

The mean and variance values are below:

Sigma^2=5, nu=3.1	**Sigma^2=2, nu=1**
Mean=3.1 **Mean of the sample=**3.006 **Variance=**5 **Variance of the sample=**4.3	**Mean=**1 **Mean of the sample=** 1.0082 **Variance=**2 **Variance of the sample=**1.94

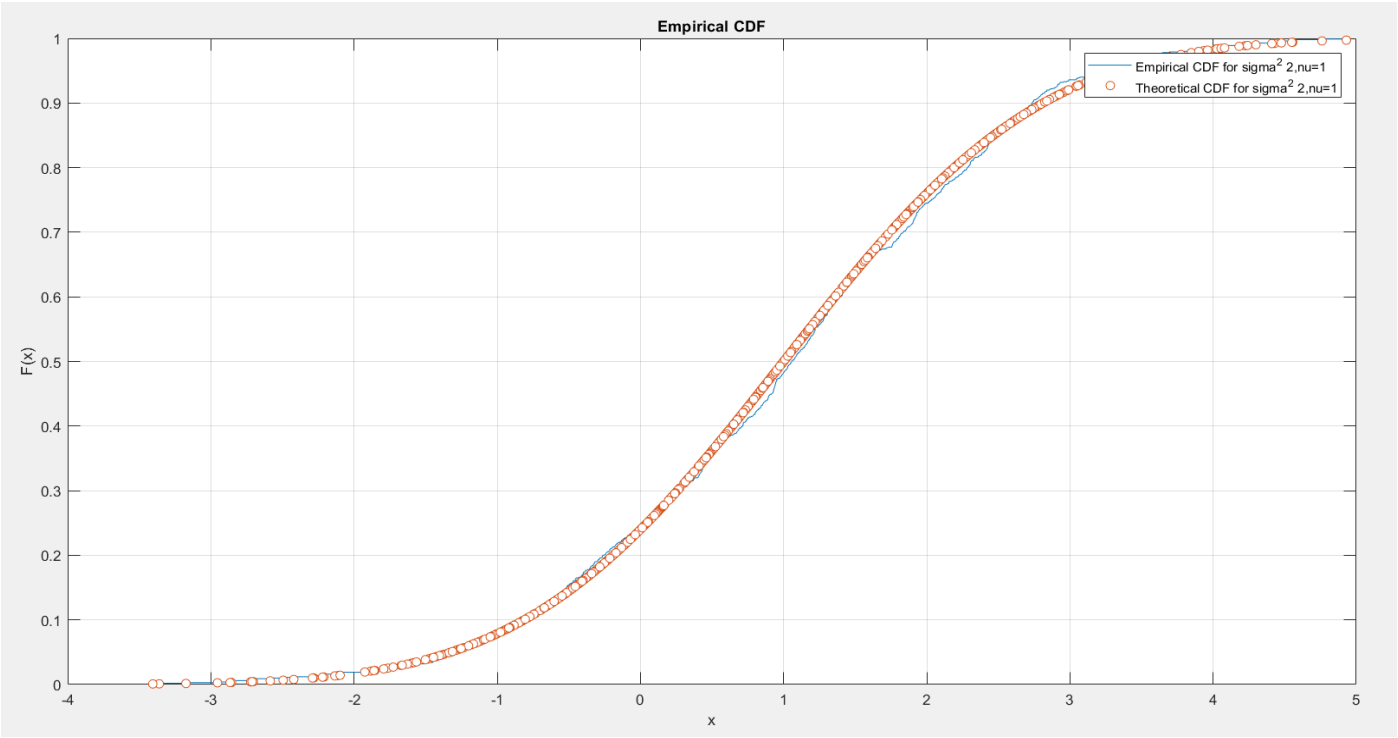


Figure 9: Emprical CDF and calculated CDF(sigma^2=2, nu=1)

In Figure 9 and 10, there is comparison of emprical cdf and calculated cdf for each set variables. According to the figures we can say that, there is almost no difference between emprical cdf and calculated cdf, there are very close to each other.

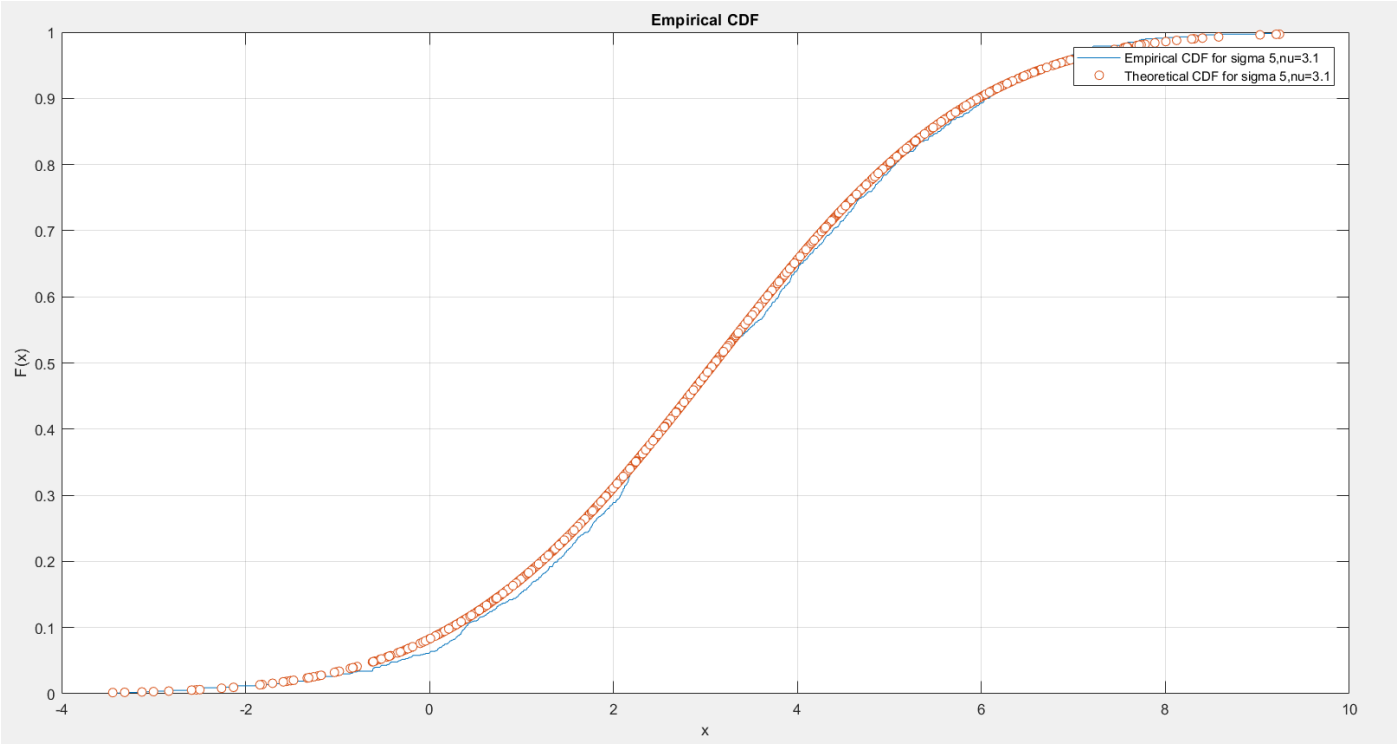
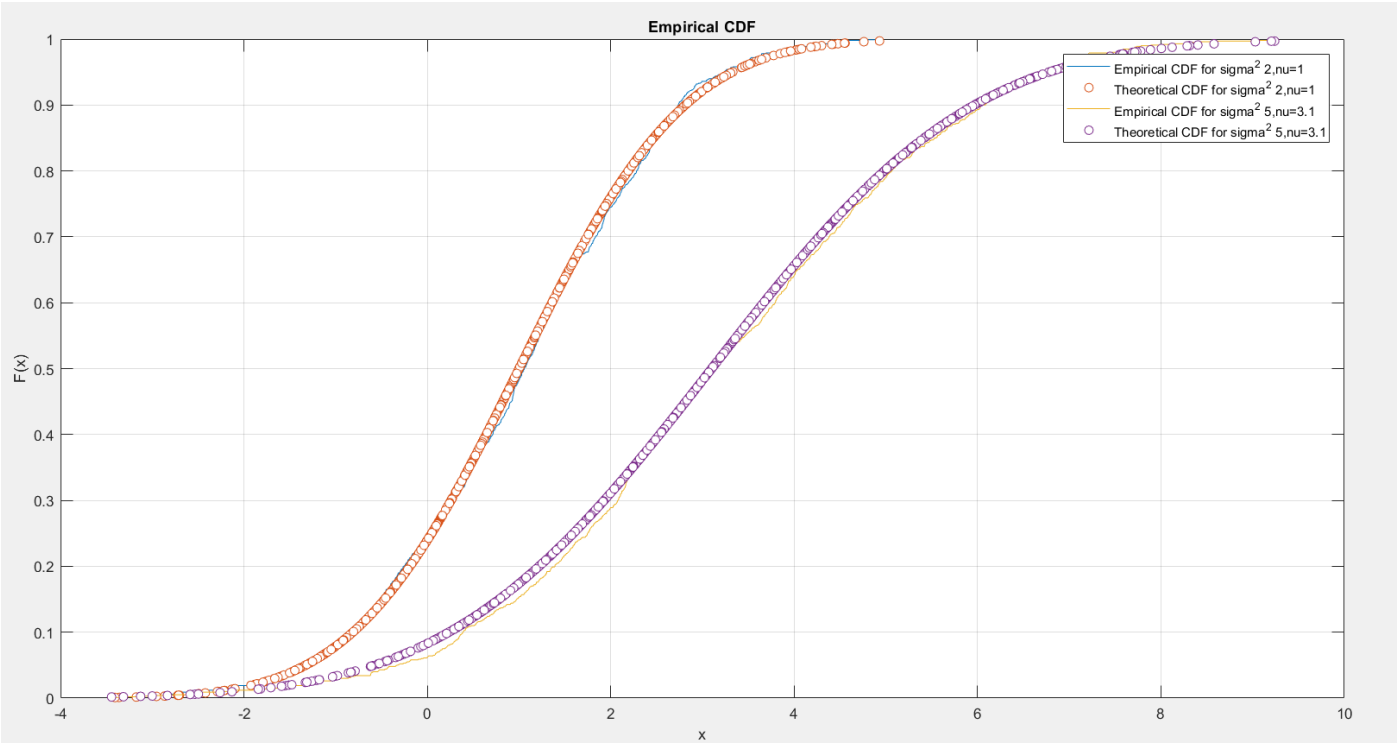


Figure 10: Emprical CDF and calculated CDF(sigma^2=5, nu=3.1)

Below figure, there is comparison between all emprical cdf results and calculated cdf results.



In figure 11, we can see two different results one result is about calculated cdf for each set variables which is left hand side and the other is about emrical cdf of generated random variables which is right hand side.

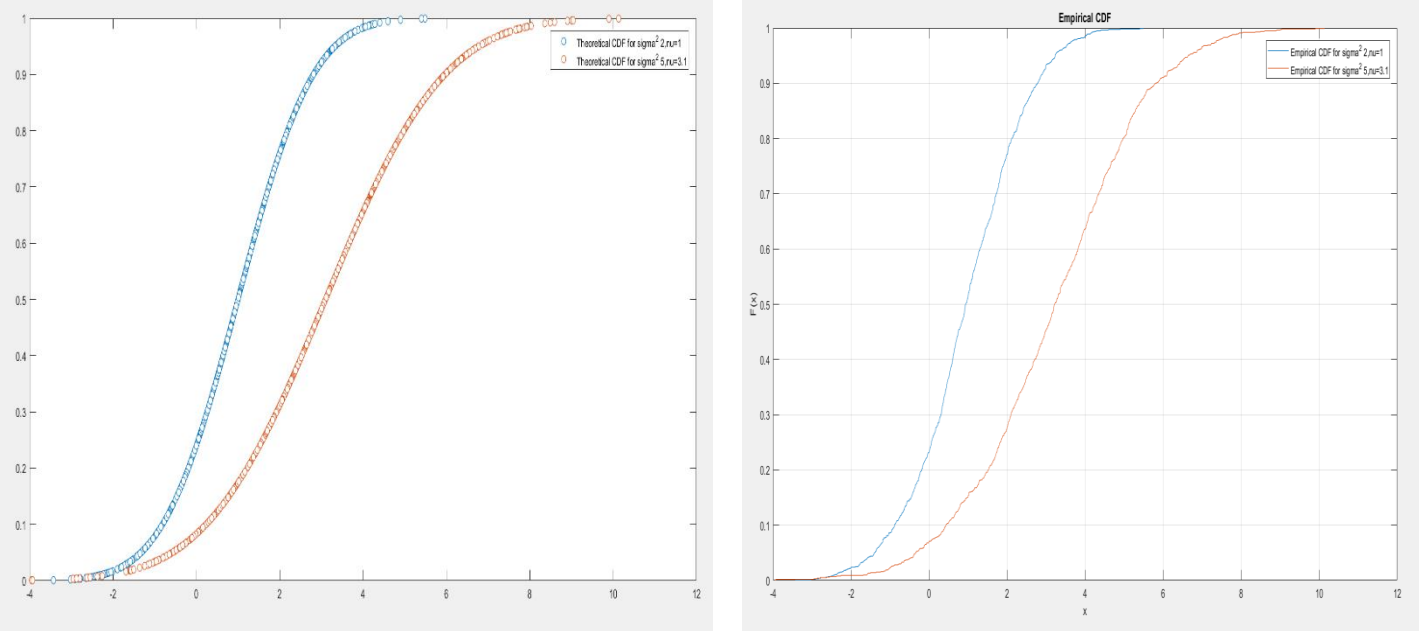


Figure 11: Emprical CDF rigth figure and calculated CDF left figure

Question 3:

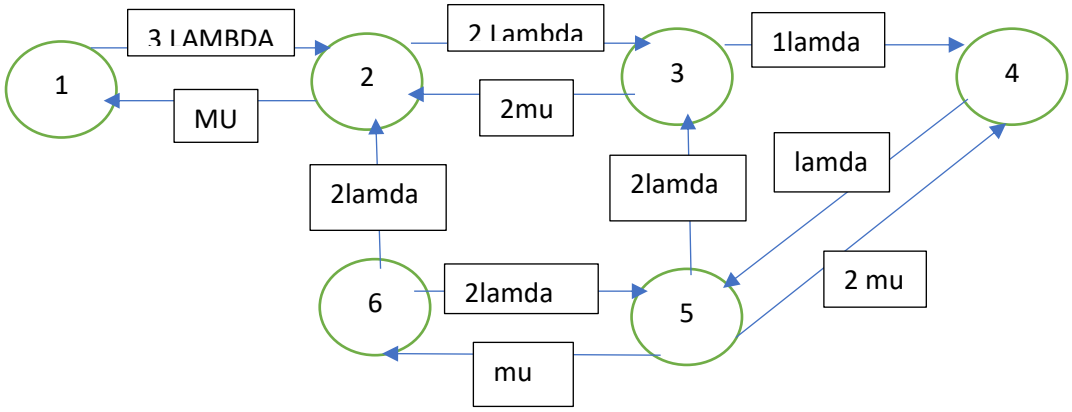
3.a Determining MATLAB code components

Entities:	Two service line, three callers
Attributes of service line:	Number of lines that are using
Attributes of callers:	Patient, impatient, calling(in service)
Activities:	Finishing phone call, making call attempt, block the caller, give a service to caller
Events:	The caller finishes service, impatient caller call again
State variables:	Time spent in state k

3.b Determine and draw state transition diyagram

- State 1: no calls , 3 callers idle
- State 2: 1 calls in progress, 2 callers idle
- State 3: 2 calls in progress, 1 callers idle
- State 4: 2 calls in progress, 1 callers impatient
- State 5: 1 calls in progress, 1 callers impatient
- State 6: 0 calls in progress, 1 callers impatient

The state transition diyagram is below:

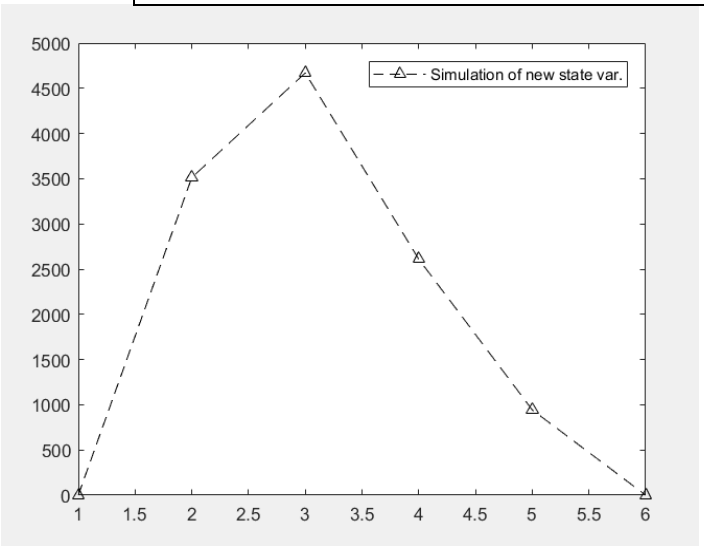


3.c Define new state variable and plot the changing of this variable with respect to time

I add new state variable number of callers which is taking a service in state. That's means, I added the new state variable to determine the number of callers being in service in each state with respect to time.

I added a new variable **numCaller** this variable is changing in each transition of the simulation with respect to **serveLineNum.** Then for finding number of callers which is taking a service in each state I used the below MATLAB code:

```
NumCallerInState = zeros(6,1);
NumCallerInState (initState) = numCallerInEachTrans(1);
for k=1:6
   for i=1:N-1
      if State(i) == k
         NumCallerInState(k) = NumCallerInState (k) + numCallerInEachTrans (i);
      end
   end
end
```



This diyagram shows as number of callers which is taking a service in each state.