Joseph Shaffer

Shaffer.567

CSE 6461

Project 1

**(1) (a) Write a computer program to generate random numbers between [0,1]. Such a random**

**number generator simulates the values generated by a uniform random variable U[0,1].**

The computer program uses the random object in java and uses the random.nextDouble() function to return a pseudorandom uniformly distributed number between 0.0 and 1.0;

**(b) Write another program (using the program in (a)) to estimate P(U > x). Plot P(U > x)**

**for values of x ϵ (0.5,1).**

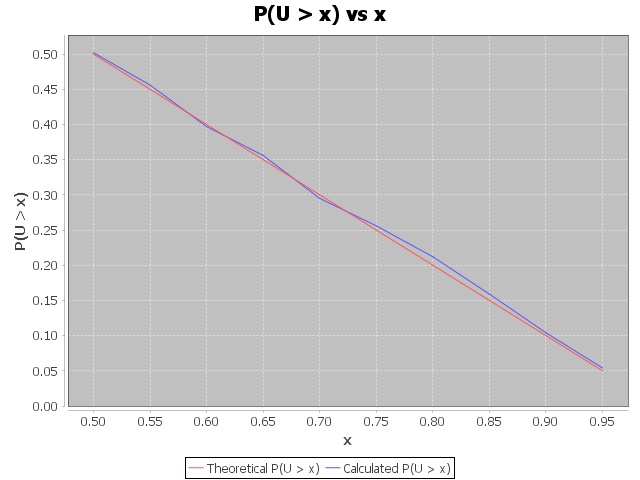


Figure P(U > x)

Figure 1 shows two lines for the probability that a uniform random variable between 0.0 and 1.0 is greater than x from the range of 0.5 to 1.0. The red line is the theoretical probability of P(U > x), which is a straight, linear line. The blue line is the calculated probability by finding a series of uniform random variables and from that series determining how many of the uniform random variables are greater than x out of the total number of random variables in the series. By doing this, the P(U > x) is found through a form of simulation.

**(2) (a) Write a computer program to simulate the values generated by Exponential (X) and**

**Poisson random (Y) variables using the program you developed in (1).**

If X is an exponential random variable with mean 1 / µ, and U is an uniform random variable between 0.0 and 1.0, we can find X by

X = -ln(U) / µ.

This formula is used to calculate an exponential random variable by taking the natural log of the uniform random variable that is generated from the program above and dividing the result by -µ.

To generate a Poisson random variable (Y), we use the idea that the time between arrivals in a Poisson process is exponentially distributed. Therefore, we count how many arrivals there are in an interval by simulating the times between arrivals and adding them up until the time sum spills over the interval.

The Poisson random variable (Y) can be generated by generating independent and identically distributed exponential random variables with mean 1/lambda, stopping as soon as the sum is greater or equal to 1;

**(b) Provide plots for P(X > x) and P(Y > x), for E(Y) = E(X) = 2. It may be necessary**

**to show your result on a plot where the vertical axis is logarithmic.**

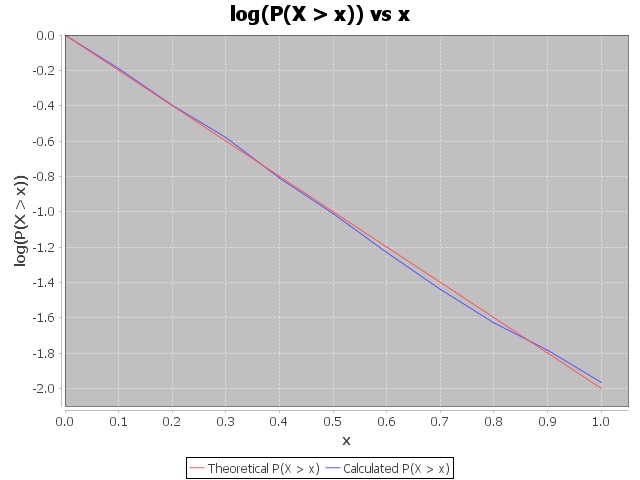


Figure log(P(X > x))

Figure 2 shows the theoretical P(X > x), which is the red line, while the blue line shows the calculated probability of P(X > x), which is determined by generating a series of exponential random variables and finding from the series, how many of the exponential random variables are greater than x. E(X) = 2 for both the theoretical and calculated probability.

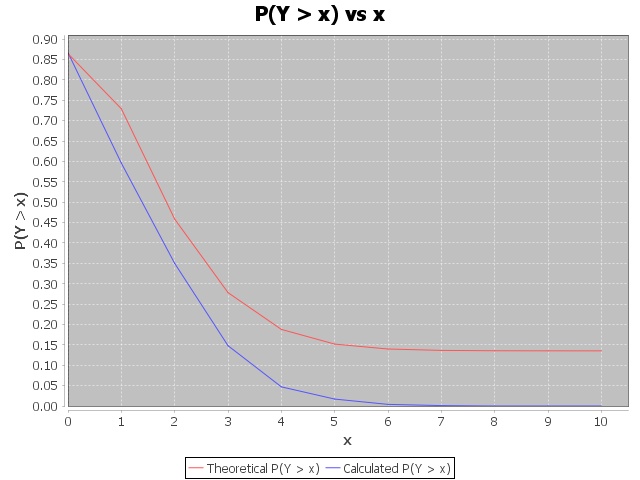


Figure P(Y > x)

Figure 3 shows the theoretical P(Y > x) and the calculated P(Y > x) for E(Y) = 2. The theoretical probability is found by summing P(Y = x) for Y= 0 to Y=x and subtracting the sum from 1, which finds the P(Y > x). Thus, P(Y > x) = 1 – (P(Y= 0) + … +(Y = x)).

The calculated P(Y > x), the blue line, is found by generating a series of Poisson random variables and determining out of all the random variables generated, the fraction of random variables Y that are greater than x.

**(3) (a) Write a computer program that simulates an M/M/1 queue.**

My program simulates an M/M/1 queue with interarrival times being exponentially distributed and the service times being exponentially distributed. The code will be submitted with the report.

**(b) Based on this program, plot Pn against n when λ = 5 and μ = 6.**

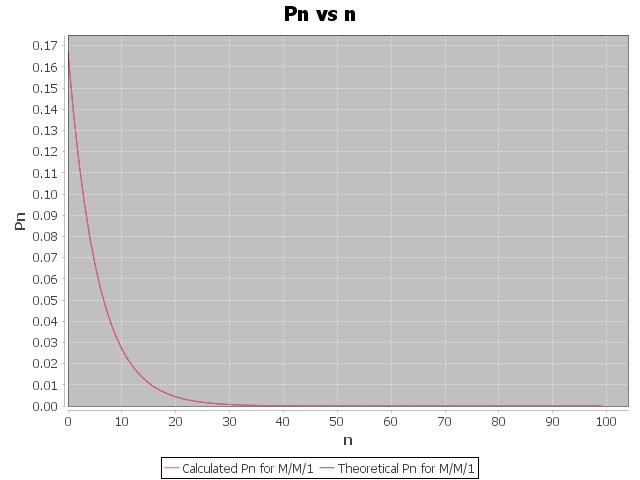


Figure P(n) for MM1

Figure 4 shows P(n) that is found for both my simulated M/M/1 system, which is the red line, and what the theoretical P(n) should be for the M/M/1 system where µ = 6, and λ = 5. The theoretical P(n) is found from the equation ρn \* P0. The calculated and theoretical Pn in the graph actually overlap, if it is not distinguishable.

**(c) Again, from your program, find the expected number and expected delay in your**

**M/M/1queueing system when ρ= 5/6.**

Generated Results from Simulated M/M/1 Queue Program:

Mean Interarrival Time = 1 / λ = 0.200 minutes

Mean Service Time = 1 / µ = 0.1667 minutes

Time of simulation = 1 x 106

Little’s Law to verify the simulation and its results: L = λ \* W

Theoretical Expected Number in M/M/1 System E(N): 5.000000000000002 customers

Expected Number in M/M/1 System E(N): 5.017758494769674 customers

Theoretical Expected Delay in M/M/1 System E(T): 1.0 minutes

Expected Delay in M/M/1 System E(T): 1.0035502160486587 minutes

By Little's Law E(N) = lambda \* E(T): 5.0 \* 1.0035502160486587 = 5.017751080243293

Expected Number = 5.017758494769674, Little’s Law = 5.017751080243293

Little’s Law verifies the results of the simulation as the expected numbers are almost identical.

**(4) (a) Write a computer program that simulates an M/Ek/1 queue. Here, Ek is an Erlang**

**random variable with k phases.**

My program simulates and M/Ek/1 queue with interarrival times that are exponentially distributed with mean arrival time of 1 / λ. The arrival process is still Poisson distributed. The service times are Erlang distributed with an Erlang random variable with k phases used for the service time of a packet. The Erlang random variable is found by

Σ(Xi), where X is an exponential random variable and i ranges from 1 to k.

**(b) Based on this program, plot Pn against n when k = 4, λ = 5 and μ = 6. Also, find**

**the expected number in the system. How do these results compare with your M/M/1**

**results in (3)?**

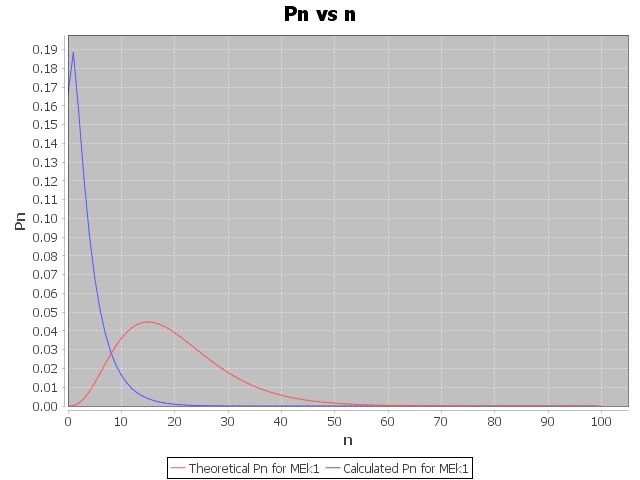


Figure P(n) M/Ek/1

Figure 5, shows the probability of n for an M/Ek/1 queue. The red line is the theoretical P(n) and the blue line is the calculated P(n), which is found from the simulation of the M/Ek/1 queue. Obviously, the calculated P(n) from the simulation shows a greater probability of n, or the number in the system being closer to 0, which makes sense for the simulation because the queue starts originally empty and takes time for the number in the system to increase, thus the probability of the number in the system will be relatively high when n is low for the simulation.

Generated results from M/Ek/1 Queue System:

Mean interarrival time: 0.200 minutes

Mean Service Time = 1 / µ = 0.1667 minutes

k-phases = 4

Time of simulation = 1 x 106

Little’s Law to verify the simulation and its results: L = λ \* W

Theoretical Expected Number in M/Ek/1 System E(N): 3.4375000000000013 customers

Expected Number in M/Ek/1 System E(N): 3.40000652315371 customers

Theoretical Expected Delay in M/Ek/1 System E(T): 0.5208333333333334 minutes

Expected Delay in M/Ek/1 System E(T): 0.6804221394458952 minutes

By Little's Law E(N) = lambda \* E(T): 5.0 \* 0.6804221394458952 = 3.4021106972294763

Expected Number = 3.40000652315371 customers, Little’s Law = 3.4021106972294763

Little’s Law verifies the results of the simulation as the expected numbers are almost identical.

When comparing the results from M/M/1 and M/Ek/1 for P(n), the expected number and expected delay, we can see the values in the M/Ek/1 are smaller.

This can be verified with the Pollaczek-Khinchin Formula (PK -Formula).

The mean queue length for an M/G/1 system, given by the PK-Formula is

E(N) = (rho) / (1 – rho) \* (1 – ((rho) / 2) \* (1 - µ2 \* σ2))

Rho is the utilization and σ2 is the variance of the service time distribution.

For M/M/1 queue system: σ2 = 1 / µ2 and as a result, the equation for E(N) becomes

E(N) = rho / (1 – rho), for the M/M/1 system.

For a M/Ek/1 queue system, the service time distribution is Erlang-k, with k phases, and a mean of 1 /µ, and the variance σ2 = 1 / (k \* µ2). From this and the PK-Formula, the E(N) becomes

E(N) = (rho) / (1 – rho) \* (1 – ((rho / 2) \* (1 – (1 / k))

With 1 > (1 – ((rho / 2) \* (1 – (1 / k)) > 0 for 1 > rho > 0 and k > 1

From these two equations for the expected number E(N) for the M/M/1 and M/Ek/1 queue systems, we can see that for the same rho value and when k > 1, the E(N) is less for the M/Ek/1 system than the M/M/1 system. In the simulations, the utilization is 5/6 and k = 4/

**(c) Plot the expected number in the system for different values of the utilization when**

**k = 40. Also plot the expected number in the system in an M/D/1 queue from the**

**analysis in class and compare the results with your simulation. What does this tell you**

**and why?**

For the M/D/1 system, the expected number in the system can be represented by

E(N) = (rho) / (1 – rho) \* (1 – (rho) / 2), where σ2 = 0.

For the same rho values, the E(N) for the M/Ek/1 system is going to be greater than the E(N) for the M/D/1 system, and the E(N) for the M/Ek/1 system approaches the E(N) for the M/D/1 system as k increases.

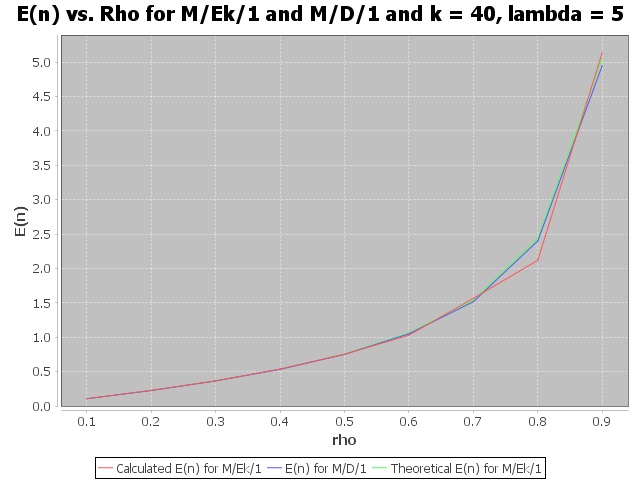


Figure E(N) for M/Ek/1 and M/D/1

Figure 6 shows the expected number when k is constant at 40, and the utilization (rho = lambda / mu) increases. The utilization is found with a constant value of lambda = 5, and mu is lambda / rho. The red line is the calculated E(n) for the different values of rho and k = 40, and is found from the simulation of the M/Ek/1 system. The blue line is the E(n) for the M/D/1 system, and is found using the equation above for the E(n) of the M/D/1. The green line is the theoretical E(n) for the M/Ek/1 system, which is found from the equation above for the E(n) of the M/Ek/1 system. All three lines are close together, showing that the E(n) for k = 4 and different values of rho are similar for the M/Ek/1 simulated and theoretical system, and the M/D/1 system. This suggests that with a constant value for k, the service time is deterministic and the expected number for M/Ek/1 and M/D/1 are equivalent for different values of rho.