Statistical Modeling and Design of Experiments

ASSIGNMENT 3 - QUEUING THEORY

January 12, 2018

Kymry Burwell
David Sarda

Contents

1 R Program Instructions

The included R program is easy to use. First, load the entire program into R studio. You must also install/load the included package. There are three separate functions included in the program that you can run directly from R studio:

serviceTimes()

This function outputs three things. (1) Statistics for 10,000 sample service times of a Weibull distribution. (2) Theoretical statistics for the Weibull distribution. (3) A comparison of the sample and theoretical statistics. It takes a single input, *loadfactor=n*: n=1 is 0.4, n=2 is 0.7, n=3 is 0.85, n=4 is 0.925. For example, run the following command:

$$serviceTimes(loadfactor = 3)$$

simulate()

This function outputs four things. (1) L, LQ, W, WQ for 10 separate simulations of 100,000 clients. (2) The average L, LQ, W, WQ of all 10 simulations. (3) LQ and WQ confidence intervals. (4) A single plot displaying the stabilization of the 10th simulation. We opted to include the last simulation because it would be too messy to display all 10. It takes a single input, *loadfactor=n*: n=1 is 0.4, n=2 is 0.7, n=3 is 0.85, n=4 is 0.925. For example, run the following command:

$$simulate(loadfactor = 3)$$

allenCuneen()

This function outputs one thing. (1) WQ and LQ of all four load factors, using the Allen Cunneen formula. There are no inputs. Simply run the following command:

$$allenCuneen()$$

2 Analysis of Service Time

We will begin with an analysis of service times by comparing theoretical service times to a sampling of 10,000 service times generated in R. A Weibull distribution will be used for the analysis. Please see section ***Analysis of Service Times*** in the attached R file for calculations.

Theoretical Service Times

In order to compute the theoretical service times, we will use the following formulas.

$$b = \frac{E[\tau] * \rho}{\gamma(\frac{a+1}{a})} \tag{1}$$

$$\rho = \frac{E[X]}{E[\tau]} \tag{2}$$

$$E[X] = b * \gamma(\frac{a+1}{a}) \tag{3}$$

$$Var[X] = b^2 * \left(\gamma(\frac{a+2}{a}) - \gamma^2(\frac{a+1}{a})\right) \tag{4}$$

$$std[X] = \frac{\sqrt{Var[X]}}{E[X]} \tag{5}$$

$$CV = \frac{std[X]}{E[X]} \tag{6}$$

Figure 1 displays the theoretical service times of a Weibull distribution with varying load factors (ρ) and a constant shape parameter (a) of 0.3135. It's interesting to note that the median is such a small value compared to the expected value of the random variable, in all cases. Furthermore, we see that the variation is extremely large, and increases as the load factor increases. Additionally, we see that as our load factor increases the expected service time moves toward the expected inter-arrival time (τ) of 79.

Weibull Distribution - Theoretical						
ρ	a	b	$E[X]$	$Var[X]$	CV	Median
0.4	0.3135	4.13	31.6	24,260	4.93	1.28
0.7	0.3135	7.23	55.3	74,297	4.93	2.24
0.85	0.3135	8.77	67.1	109,550	4.93	2.73
0.925	0.3135	9.55	73.1	129,736	4.93	2.97

Figure 1: Theoretical service times

Sample Service Times

Next, we generated 10,000 sample service times using the same Weibull distribution. We did this for each loading factor. Figure 2 displays the results. From a quick scan, we can see that the statistics of the sample data and the theoretical data exhibit similar characteristics. Figure 4 displays histograms of the sample distributions with each load factor. In order to make it easier to view, we set a limit on the x-axis of 400. It's a bit hard to tell from the graphs, but as the load factor increases, the average service times increase as well. We can clearly see that the distribution is long-tailed. Additionally, there is a lot of variability among the service times. It is interesting to note that a large portion of the services times are far smaller than the expected service time.

Weibull Distribution - Sampled						
ρ	a	b	$E[X]$	$Var[X]$	CV	Median
0.4	0.3135	4.13	34.1	26,068	4.73	1.27
0.7	0.3135	7.23	53.17	55,843	4.45	2.35
0.85	0.3135	8.77	70.44	111424	4.73	2.76
0.925	0.3135	9.55	72.54	112,589	4.62	2.93

Figure 2: Sampled service times

Comparison of Service times

In order to compare the theoretical and sample service times we simply used the ratio of the sample statistics against the theoretical statistics. The results are displayed in figure 3. We see that, on average, the results were very similar. The results from the load factor of 0.7 did vary a bit more than the rest, but this is likely due to the seeds chosen more than anything else.

Weibull Distribution - Comparison						
ρ	a	b	$E[X]$	$Var[X]$	CV	Median
0.4	0.3135	4.13	0.93	0.93	1.04	1.01
0.7	0.3135	7.23	1.04	1.33	1.11	0.95
0.85	0.3135	8.77	0.95	0.98	1.04	0.99
0.925	0.3135	9.55	1.01	1.15	1.07	1.01

Figure 3: Theoretical vs Sample service times

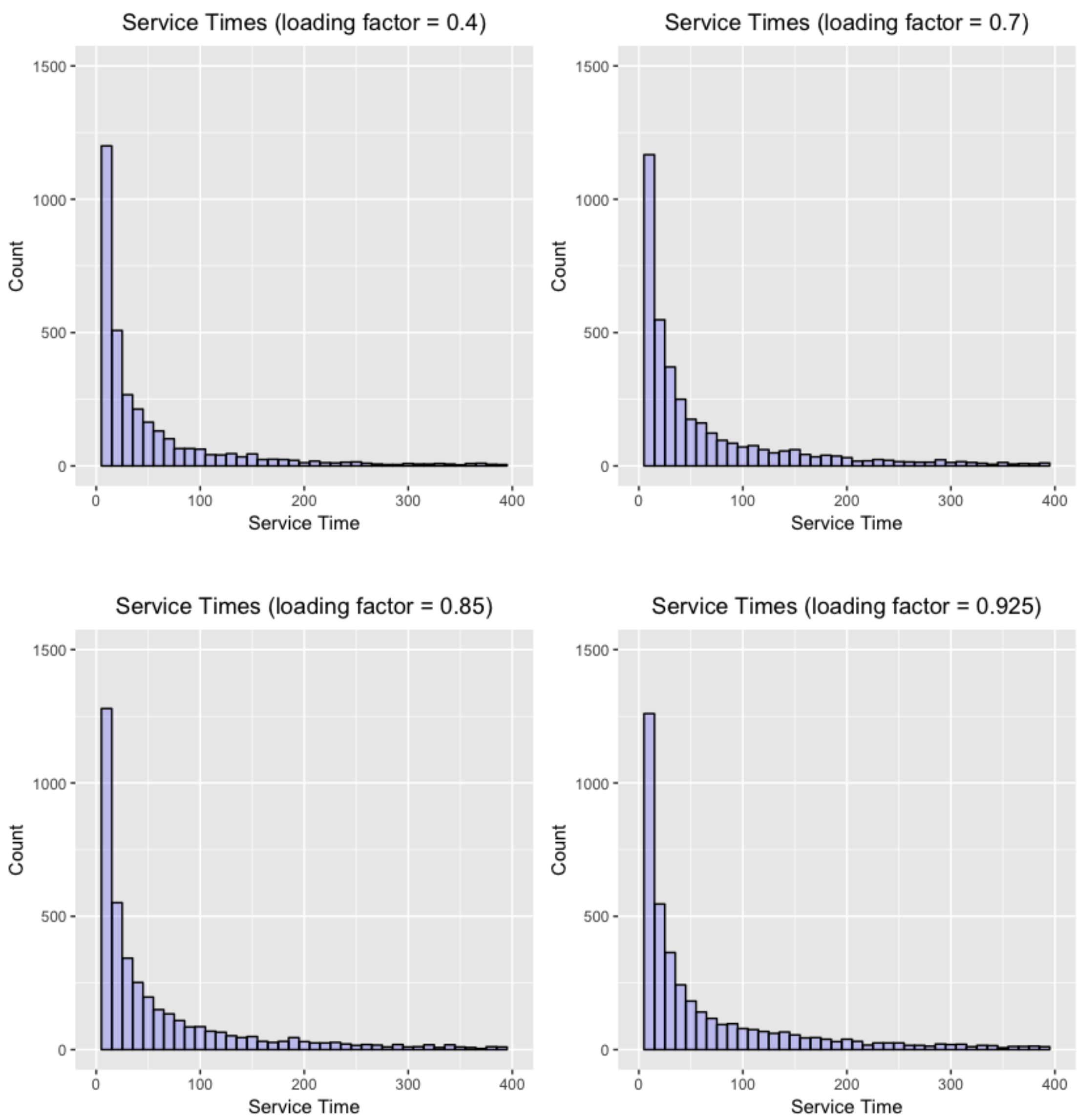

Figure 4: Sample service distributions with varying load factors

3 G/G/1 SIMULATION

Simulation

To simulate a G/G/1 queuing system, we built a program in R. Each execution of the program carries out 10 simulations (of 100,000 clients) using different seeds. For each simulation, it records the average occupancy of the waiting system (L) and the queue (Lq) and the average time spent in the waiting system (W) and the queue (Wq). See figure 8. After all simulations are run, it calculates the average waiting time and queue lengths over all 10 simulations. See figures 7 and 9. Finally, it calculates 95% confidence intervals for the waiting time and queue length. See figure 10.

Steady State

In order to visually check that the simulations reached a steady state, we plotted the average occupancy (L_{T_i}) vs the entrance time instant (t_i). We did this for two simulations (chosen at random) for each load factor. Figures 5 and 6 display the results. It appears that, on average, as the load factor increases, the time it takes to reach steady state increases as well. These 8 graphs don't show this definitively, but we ran many more simulations and it appeared to be the case in general.

Additionally, we see that simulations with the same loading factor eventually reach steady states with roughly equal average occupancies. This occurs even though the average occupancy fluctuates greatly over the simulations. For example, both simulations with a loading factor of 0.7 reached steady states of approximately 20, but we see that the first simulation reached an average occupancy of over 120 in it's early time instances, while the second simulation never surpassed 30.

Looking at the bottom two graphs in figure 6 (load factor = 0.925), we see that steady state isn't quite reached but is very close; it may take a few thousand more clients to reach steady state.

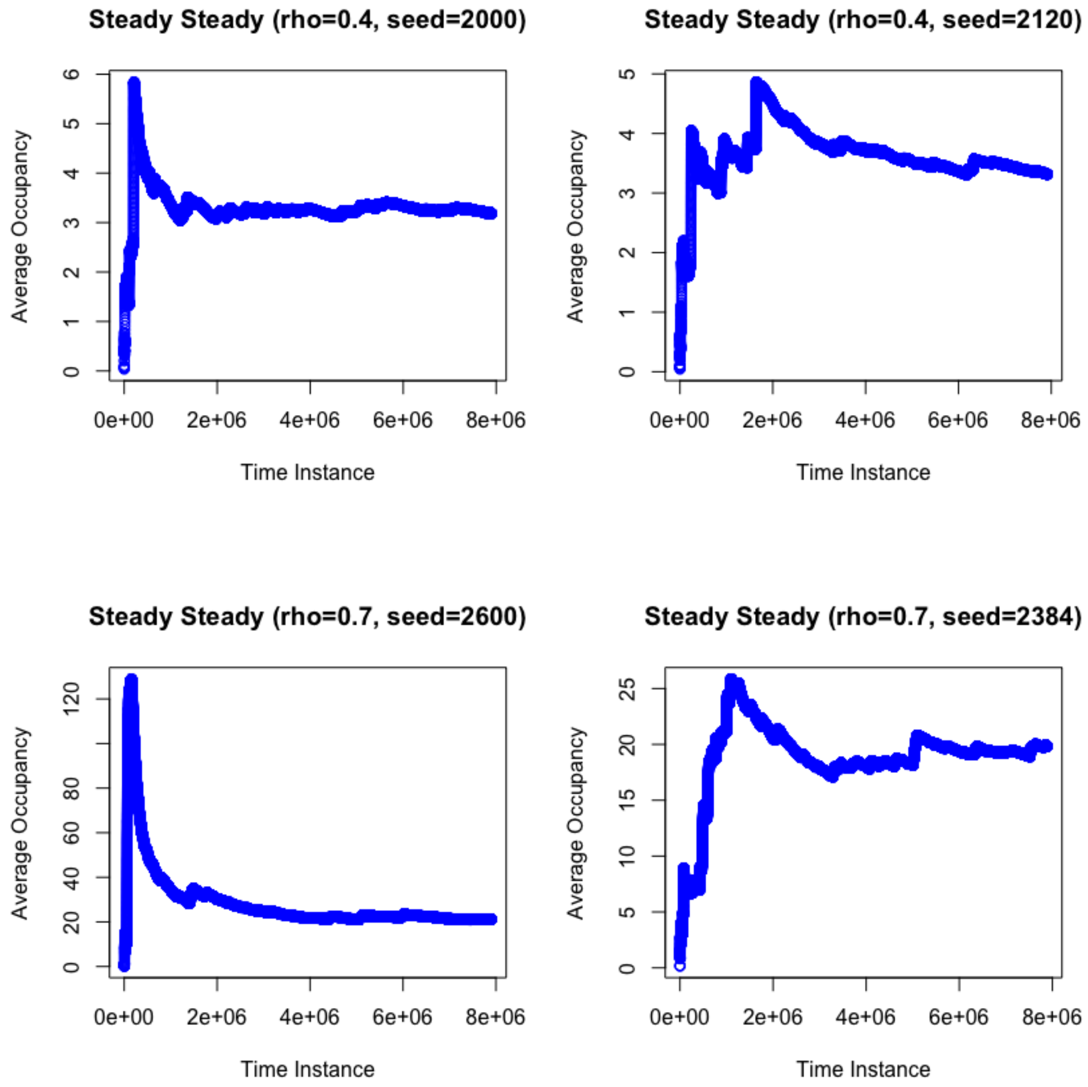

Figure 5: Steady State for Load Factors 0.4 and 0.7

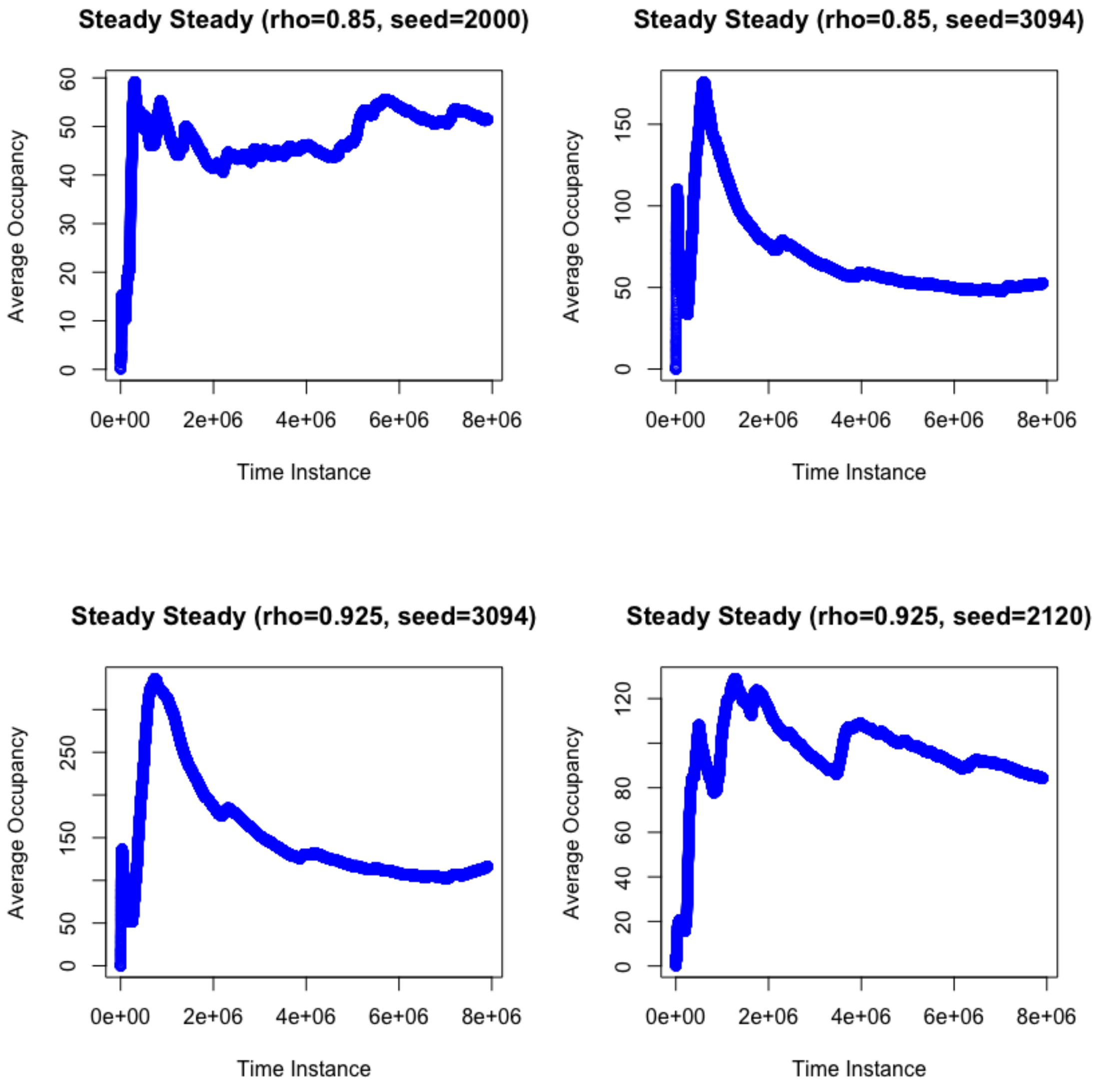

Figure 6: Steady State for Load Factors 0.85 and 0.925

Results

Looking at figures 7 and 9 we can see that when the loading factor is small, the average occupancy and waiting time are also small. With a loading factor of 0.4, we see that the average occupancy of the waiting system is around 3.5 and the average time spent in the waiting system is around 270 time units. However, when the loading factor is

0.925, we see that the average occupancy is roughly 130 and the average waiting time is about 10,000. This is exactly what we expect to see. As the loading factor increases, so too does the occupancy of the system. This makes sense, as the loading factor determines the percentage of the 'server' being used on average. Figure 10 displays 95% confidence intervals. Here we can see that as the loading factor increases, the size of the confidence intervals increases as well. This is understandable because as the loading factor increases, the number of clients in the waiting system at any given time increases as well, introducing more variability into the system. When the loading factor is 0.925, we can see the interval is quite large. Much of this is size is likely due to the long-tailed distribution we are using.

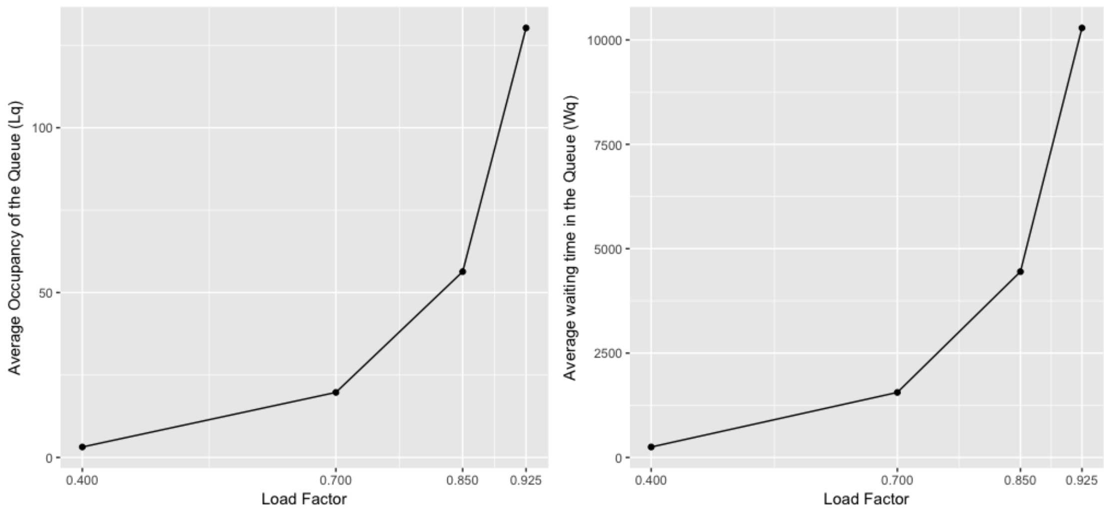

Figure 7: Wq and Lq for each load factor

Simulation	L	Lq	W	Wq
1	3.189	2.785	251.89	219.954
2	3.295	2.898	259.782	228.514
3	3.935	3.53	310.26	278.378
4	3.316	2.923	262.418	231.332
5	3.109	2.719	245.617	214.814
6	3.596	3.199	284.294	252.868
7	4.033	3.641	319.575	288.546
8	3.525	3.123	278.086	246.346
9	4.588	4.183	362.245	330.236
10	3.198	2.806	252.923	221.903

Load factor = 0.4

Simulation	L	Lq	W	Wq
1	51.463	50.604	4065.155	3997.291
2	51.63	50.787	4070.641	4004.197
3	74.428	73.569	5868.718	5800.97
4	45.025	44.19	3563.063	3497.005
5	43.634	42.805	3446.954	3381.498
6	55.13	54.286	4357.943	4291.163
7	58.691	57.859	4650.728	4584.792
8	55.02	54.165	4340.656	4273.209
9	84.396	83.534	6663.163	6595.143
10	52.385	51.551	4143.017	4077.1

Load factor = 0.85

Simulation	L	Lq	W	Wq
1	17.895	17.188	1413.566	1357.679
2	18.491	17.797	1457.873	1403.154
3	24.994	24.287	1970.821	1915.029
4	17.924	17.237	1418.451	1364.05
5	16.953	16.271	1339.265	1285.361
6	21.079	20.384	1666.29	1611.294
7	22.196	21.511	1758.844	1704.544
8	19.854	19.15	1566.308	1510.763
9	27.231	26.522	2149.935	2093.918
10	17.566	16.88	1389.261	1334.975

Load factor = 0.7

Simulation	L	Lq	W	Wq
1	126.268	125.333	9974.164	9900.313
2	137.199	136.282	10817.19	10744.88
3	165.647	164.712	13061.41	12987.68
4	84.197	83.288	6662.975	6591.088
5	93.088	92.186	7353.666	7282.435
6	106.648	105.728	8430.287	8357.614
7	115.362	114.456	9141.382	9069.627
8	142.279	141.348	11224.78	11151.38
9	224.649	223.711	17736.34	17662.32
10	116.302	115.395	9198.068	9126.334

Load factor = 0.925

Figure 8: Simulation Output for all loading factors

load factor	L (mean)	LQ (mean)	W (mean)	WQ (mean)
0.4	3.59	3.18	282.71	251.29
0.7	20.42	19.73	1613.06	1558.08
0.85	57.18	56.34	4517	4450.24
0.925	131.17	130.24	10360.03	10287.37

Figure 9: Averages over 10 simulations

Lq lower	Lq upper	Wq lower	Wq upper
2.825	3.536	223.216	279.362
Load factor = 0.4			
17.119	22.327	1352.8	1763.36
Load factor = 0.7			
46.714	65.956	3692.26	5208.22
Load factor = 0.85			
99.633	160.855	7876.05	12698.7
Load factor = 0.925			

Figure 10: Confidence Intervals for all loading factors

4 Allen-Cunneen Approximation

In this section, we calculated the expected waiting time in the queue (Wq) and occupancy of the queue (Lq) using the Allen-Cunneen approximation formulas for all loading factors. We then compared this with the results of our simulations. The following formulas are used to compute the approximations:

$$Wq \approx \frac{C(S,\theta) * (\lambda^2 \sigma_\tau^2 + \mu^2 \theta_x^2)}{2S\mu(1-\rho)} \tag{7}$$

$$C(S,\theta) = \frac{\dfrac{\theta^S}{S!(1-\rho)}}{\sum_{l=0}^{S-1} \dfrac{\theta^l}{l!} + \dfrac{\theta^2}{S!(10\rho)}} \tag{8}$$

$$\lambda = \frac{1}{E[\tau]} \tag{9}$$

$$\mu = \frac{1}{E[X]} \tag{10}$$

$$\rho = \frac{\lambda}{S\mu} \tag{11}$$

$$\theta = \frac{\lambda}{\mu} \tag{12}$$

$$Lq = Wq * \lambda \tag{13}$$

Results

Figure 11 displays the results of the Allen-Cunneen Approximation over each loading factor. Figure 12 displays the averages of our 10 simulations. Finally, figure 13 displays comparison between our simulations and the Allen-Cunneen approximations. To compute the comparison we simply took the ratio of our simulation averages against the Allen-Cunneen approximations. Overall, we see that the approximations were quite good, all with less than a 7% difference.

Allen-Cunneen Approximation		
ρ	WQ	LQ
0.4	258.54	3.27
0.7	1583.57	20.01
0.85	4669.94	59.11
0.925	11060.81	140.01

Figure 11: Allen-Cunneen Approximation

Simulation Averages		
ρ	WQ	LQ
0.4	251.29	3.18
0.7	1558.08	19.72
0.85	4450.24	56.34
0.925	10287.37	130.24

Figure 12: Simulation Averages

Simulation/Allen-Cunneen Comparison		
ρ	WQ	LQ
0.4	0.972	0.973
0.7	0.984	0.986
0.85	0.953	0.953
0.925	0.930	0.930

Figure 13: Simulation/Allen-Cunneen Comparison