Queuing Theory Exercise Series 2

Alexandros Kyriakakis (03112163)

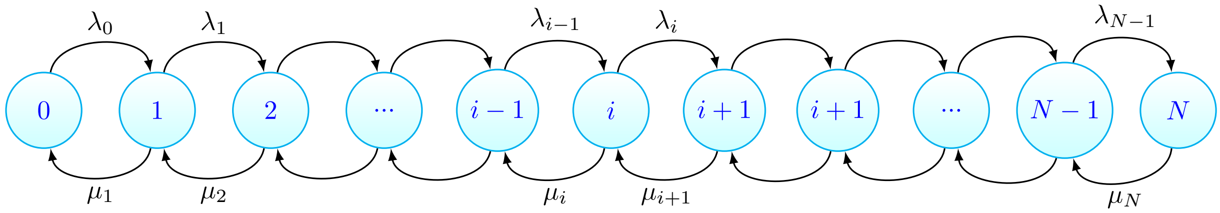
April 2020

1 Theoretical Part of Queue M/M/1

1.1 A

From theory we know that in order $M/M/1$ to be ergodic it must be:

$$\rho = \frac{\lambda}{\mu} < 1 \; Erlang$$



Hence, Arrivals follow Poisson distribution with parameter λ customers per second and the services follow exponential distribution with parameter λ customers per second then:

$$\lambda_0 = \lambda_1 = ... = \lambda_i = ... = \lambda, \; i = 1, 2, 3, ...$$

and

$$\mu_0 = \mu_1 = ... = \mu_i = ... = \mu, \; i = 1, 2, 3, ...$$

Using local and global equations of equilibrium:

$$\lambda P_{i-1} = \mu P_i, \; i = 1, 2, 3, ...$$

and

$$(\lambda_k + \mu_k)P_k = \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}, \; k = 1, 2, 3, ...$$

We also used the equations of ergodic probability normalization $P_0 + ... + P_N$ we have:

$$\lambda P_0 = \mu P_1 \Rightarrow P_1 = \frac{\lambda}{\mu} P_0 = \rho P_0$$

$$(\lambda + \mu)P_1 = \lambda P_0 + \mu P_2 \Rightarrow P_2 = \rho^2 P_0 \Rightarrow P_k = \rho^k P_0$$

$$P_0 \frac{1}{1-\rho} = 1 \Rightarrow P_0 = 1 - \rho \Rightarrow P_k = (1-\rho)\rho^k, \; k > 0 \; and \; P(n(t) > 0) = 1 - P_0 = \rho$$

1.2 B

Using Little's Law we know that the average waiting time of a customer in the system at equilibrium state is $E(T) = \frac{E[n(t)]}{\gamma} = \frac{E[n(t)]}{\lambda} = \frac{1}{\mu(1-\rho)}$.

1.3 Γ

At $k = 57 \Rightarrow P_{57} = (1-\rho)\rho^{57}$ which leads to a positive probability of having 57 customers in the system. We notice that P_i grow is bonded with ρ growth.

1.4 Δ

Above we analyzed the permanent ergodic state of the system, where $t \to \infty$ and there are no transitional states and the initial state has been forgotten. So, if the system had 5 customers at the beginning we would see no difference.

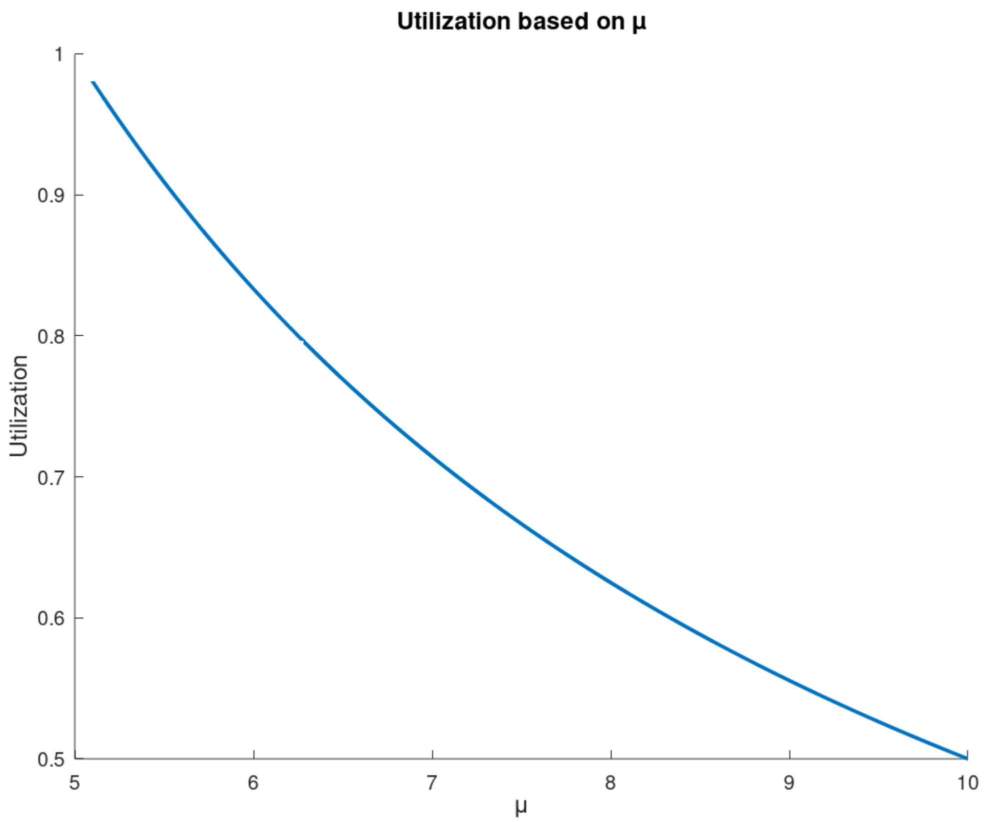
2 Analysis of Queue M/M/1

2.1 α

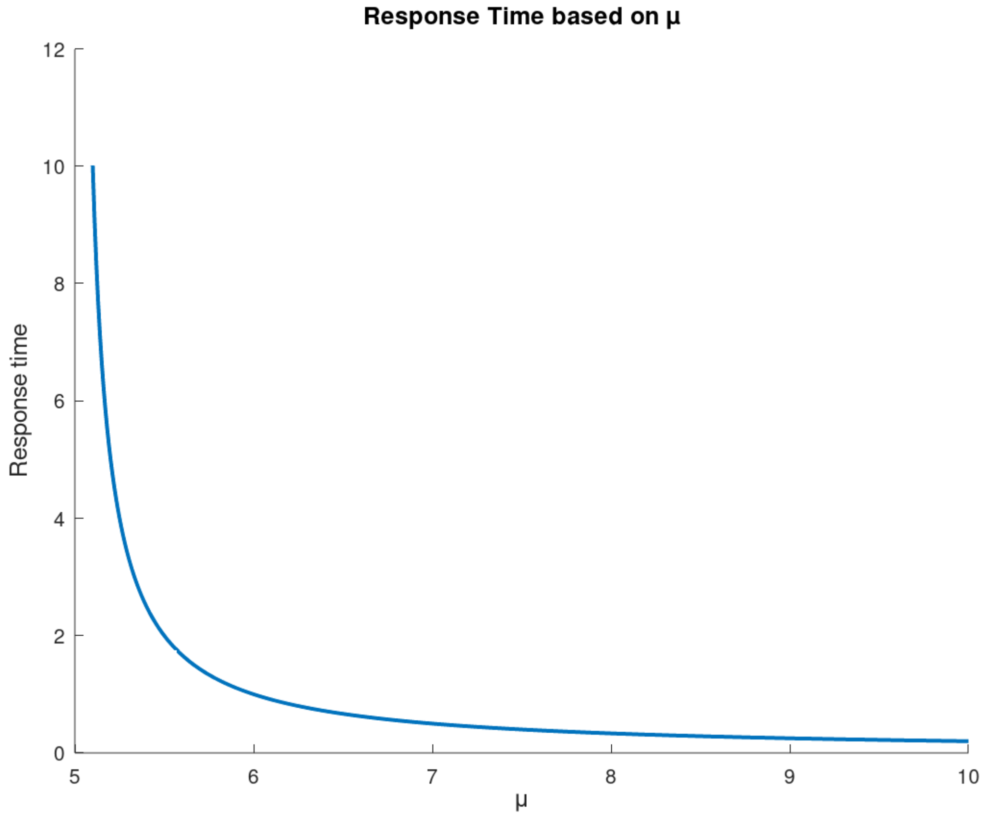
Based on previous theoretical results we know that $\rho = \frac{\lambda}{\mu} > 1 \Rightarrow \mu > \lambda$. So, we choose 50 sample prices from $5.1 \to 10$ customers per minute.

2.2 β

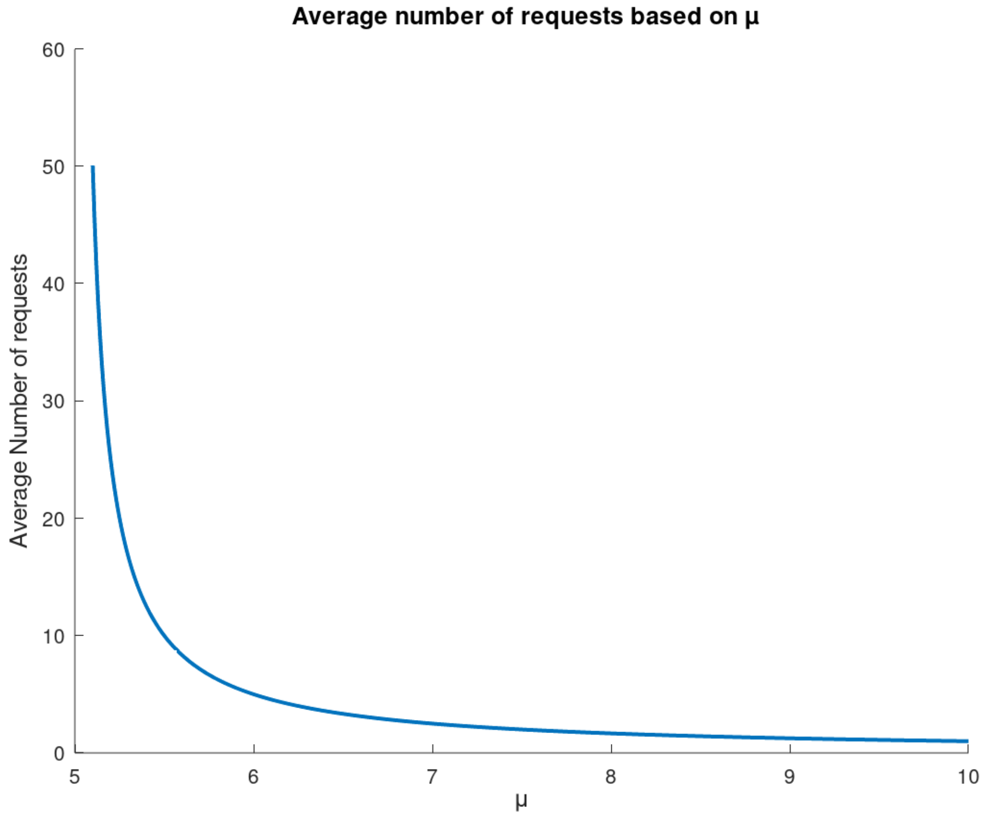
2.2.1 Utilization



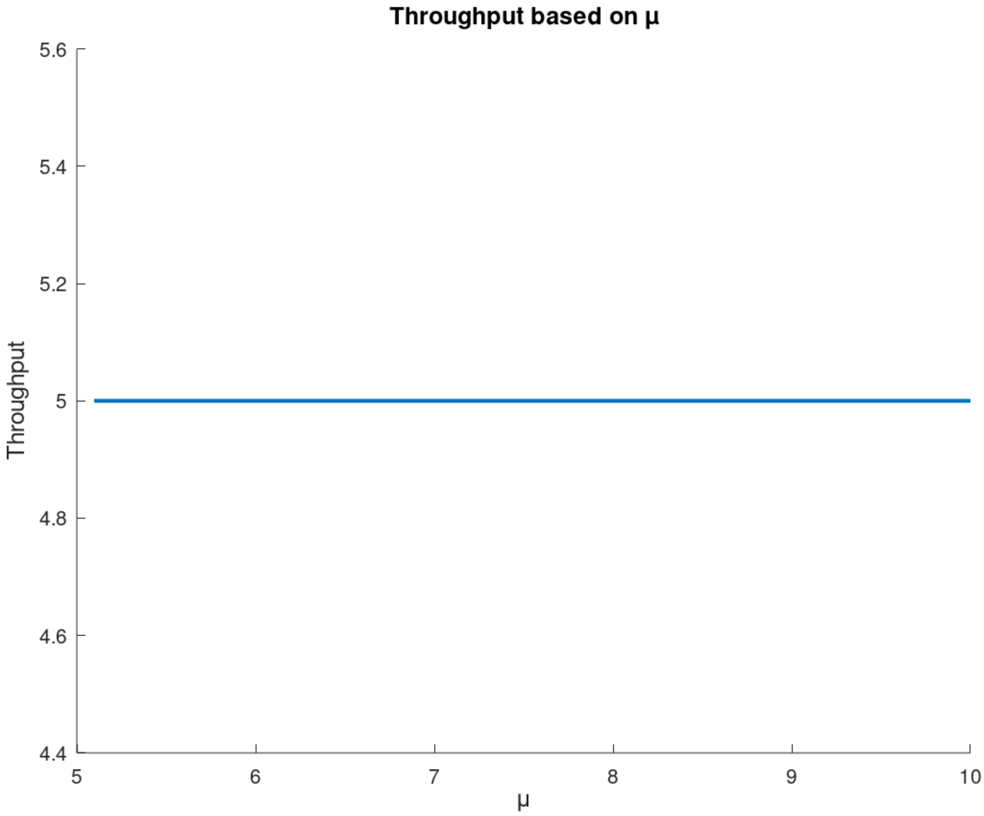
2.2.2 Response Time



2.2.3 Average number of customers



2.2.4 Throughput



2.3 Γ

We notice that the average service time stabilizes when the rate of service goes near $(8 - 10)$ customers per minute. Based on the cost increase which comes with rate increase, we would choose the minimum bound of stabilized state. So near 8 customers per minute would be the best.

2.4 Δ

We know that $\gamma = \lambda(1 - P(blocking))$ where $P(blocking)$ is the probability of losing a customer. In an $M/M/1$ system queue has infinite space so the probability $P(blocking) = 0$ and so throughput is constant. $\gamma = \lambda$.

2.5 Code

```
pkg load statistics
pkg load queueing

clc;
clear all;
```

```
close all;
# M/M/1
# (b)
lambda = 5
U=[0,500]; #utiliaztion
R=[0,500]; #server responce time
Q=[0,500]; #average number of requests
X=[0,500]; #server throughput

mu = [5.1:0.01:10];
display(mu)
for i=1:columns(mu)
    [U(i),R(i),Q(i),X(i)] = qsmm1(lambda, mu(i));
endfor
# Utiliaztion
figure(1);
hold on;
plot(mu,U,"linewidth",2.2);
title("Utilization based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Utilization","fontsize",12);
saveas (1, "figures/figure1.png")
hold off;
# Server responce time
figure(2);
hold on;
plot(mu,R,"linewidth",2.2);
title("Response Time based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Response time","fontsize",12);
saveas (2, "figures/figure2.png")
hold off;
# Average number of requests
figure(3);
hold on;
plot(mu,Q,"linewidth",2.2);
title(" Average number of requests based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Average Number of requests","fontsize",12);
saveas (3, "figures/figure3.png")
hold off;
# Server throughput
figure(4);
hold on;
plot(mu,X,"linewidth",2.2);
title("Throughput based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Throughput","fontsize",12);
saveas (4, "figures/figure4.png")
hold off;

clc;
clear all;
close all;
exit;
```

3 Comparing Systems with two Services

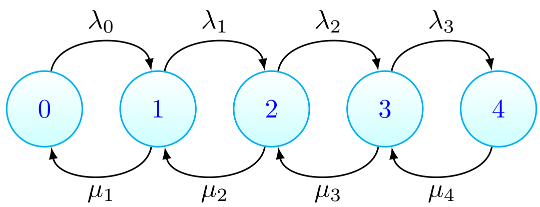
Using Octave we notice that $R_1 = 0.13333\ min$ and $R_2 = 0.2\ min$. We notice that $M/M/2$ is faster than two parallel $M/M/1$. This comes as result to the disruption of the Poisson evolution using Bernoulli random variables $\lambda_1 = p\lambda\ and\ \lambda_2 = q\lambda$. So in this case using $p = 0.5$ we construct two independent $M/M/1$ systems with rate $\lambda_2 = 5$ customers per minute. Also, using the theorem of total average price, we have that the average delay time of a customer is $0.5R_2 + 0.5R_2 = 0.2$ minutes. So we would choose $M/M/2$.

3.1 Code

```
pkg load queueing

clc;
clear all;
close all;

# Sigkrisi me 2 eksipiretites
[U1,R1,Q1,X1] = qsmm1(5,10);

[U2,R2,Q2,X2] = qsmmm(10, 10, 2);
display([R1,R2])

exit;
```

4 Birth-Death Procces with M/M/1/K

4.1 α



We have:

$$\lambda_i = \frac{\lambda}{i+1}\ and\ \mu_i = \mu,\ i = 0, 1, 2, 3$$

So,

$$\lambda_0 P_0 = \mu_1 P_1 \Rightarrow \lambda_{k-1} P_{k-1} = \mu_k P_k,\ k = 1, 2, 3, 4$$

$$P_0 + P_1 + P_2 + P_3 + P_4 = 1$$

$$P_k = \frac{\lambda^k}{k!\mu^k} P_0,\ k = 1, 2, 3, 4 \Rightarrow P_k = \frac{\rho^k}{k!} P_0,\ k = 1, 2, 3, 4$$

Where $\sum P_k = 1$ where ergodic probabilities are

$$P_0 = 0.60664,\ P_1 = 0.30332\ P_2 = 0.075830 P_3 = 0.012638\ P_4 = 0.0015798$$

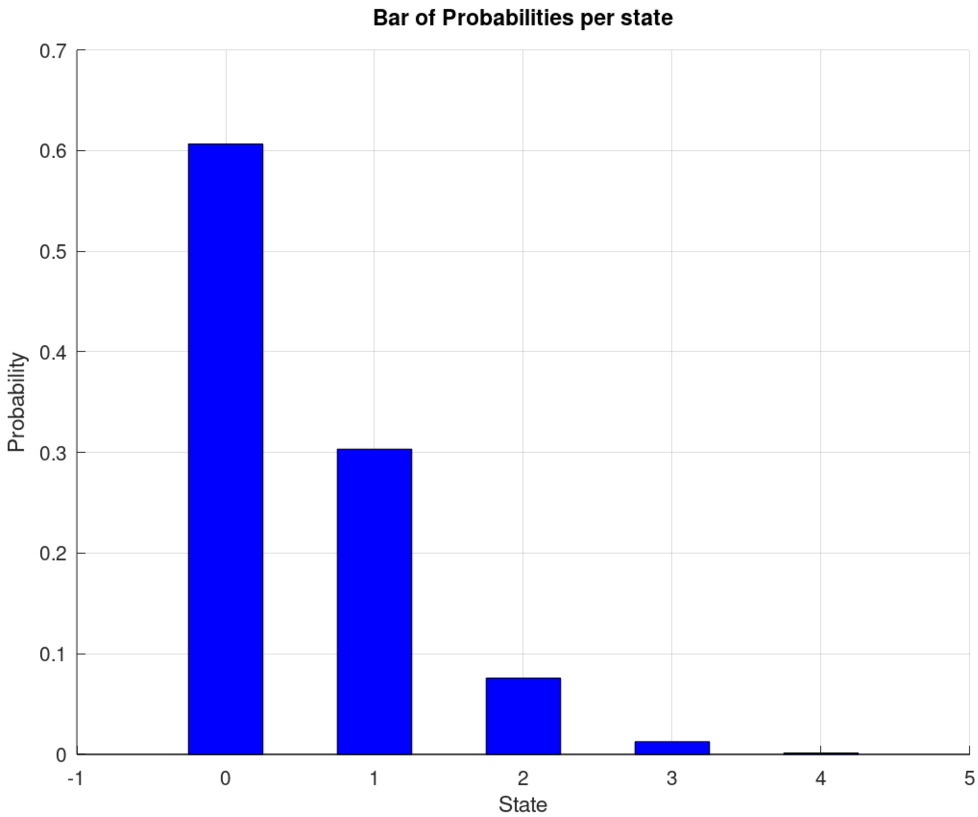
and $P_{loss} = P_4 = 0.0015798$.

4.2 β

4.2.1 i

The transition array is: $\begin{bmatrix} -5 & 5 & 0 & 0 & 0 \\ 10 & -12.5 & 2.5 & 0 & 0 \\ 0 & 10 & -11.667 & 1.6667 & 0 \\ 0 & 0 & 10 & -11.25 & 1.25 \\ 0 & 0 & 0 & 10 & -10 \end{bmatrix}$

4.2.2 ii



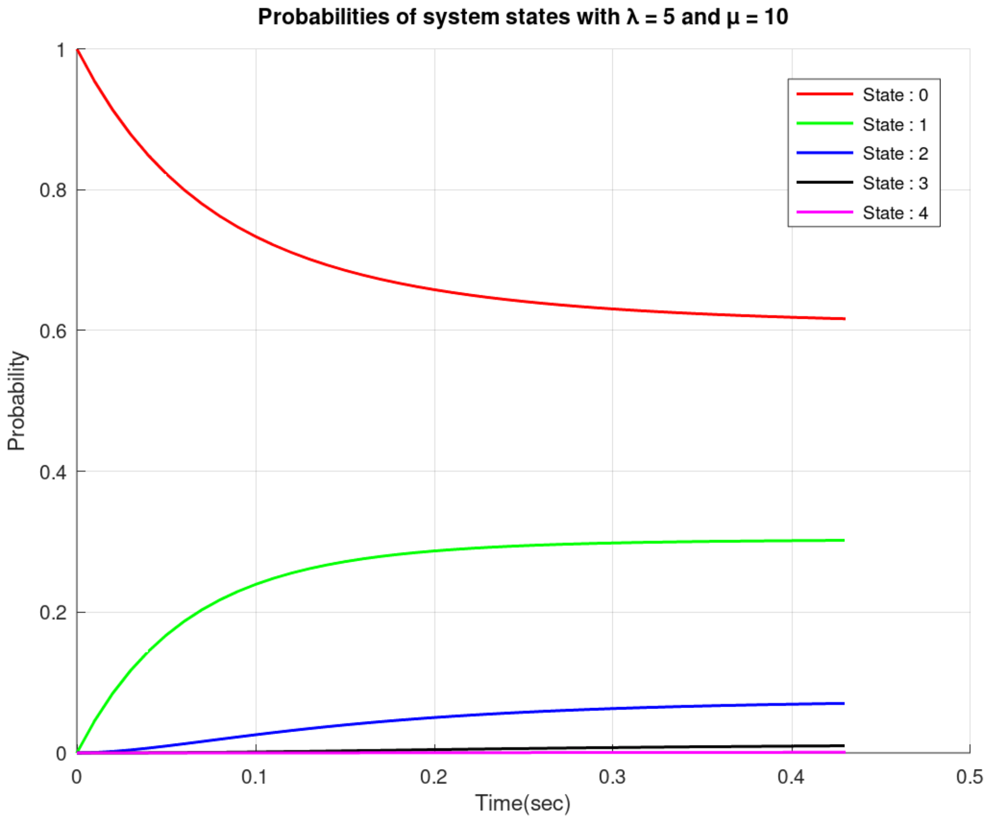
4.2.3 iii

Based on definition $\sum kP_k = 0.49921,\ k = 0, 1, 2, 3, 4.$

4.2.4 iv

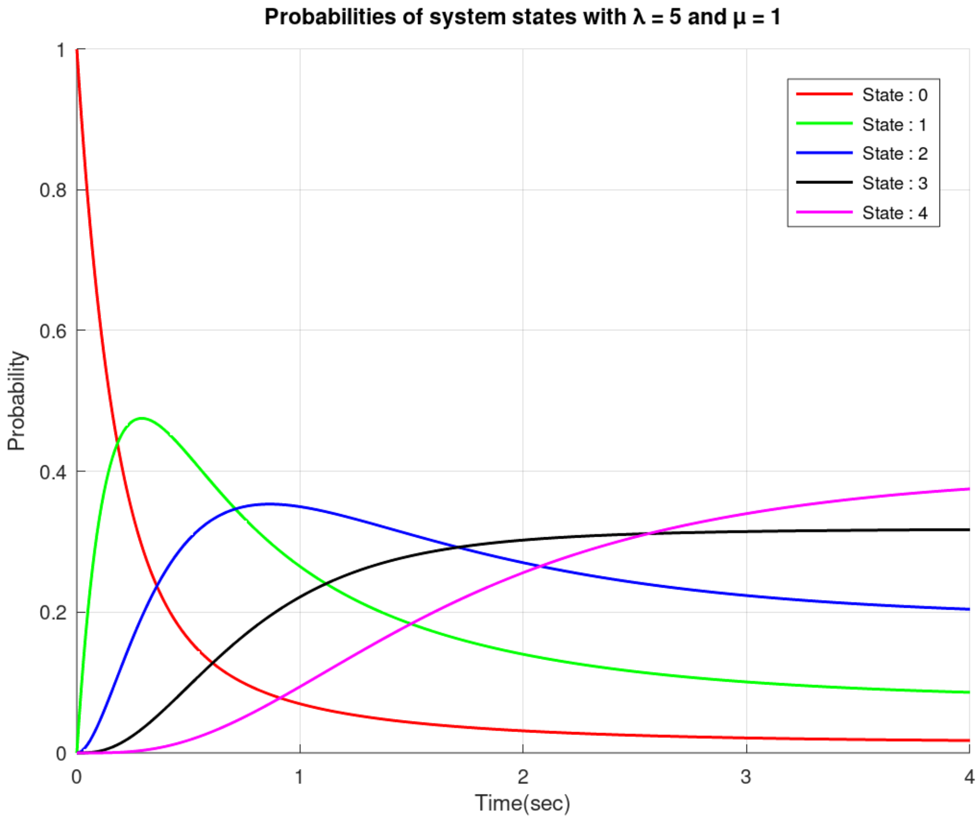
We already calculated $P_blocking = P4 = 0.0015798$.

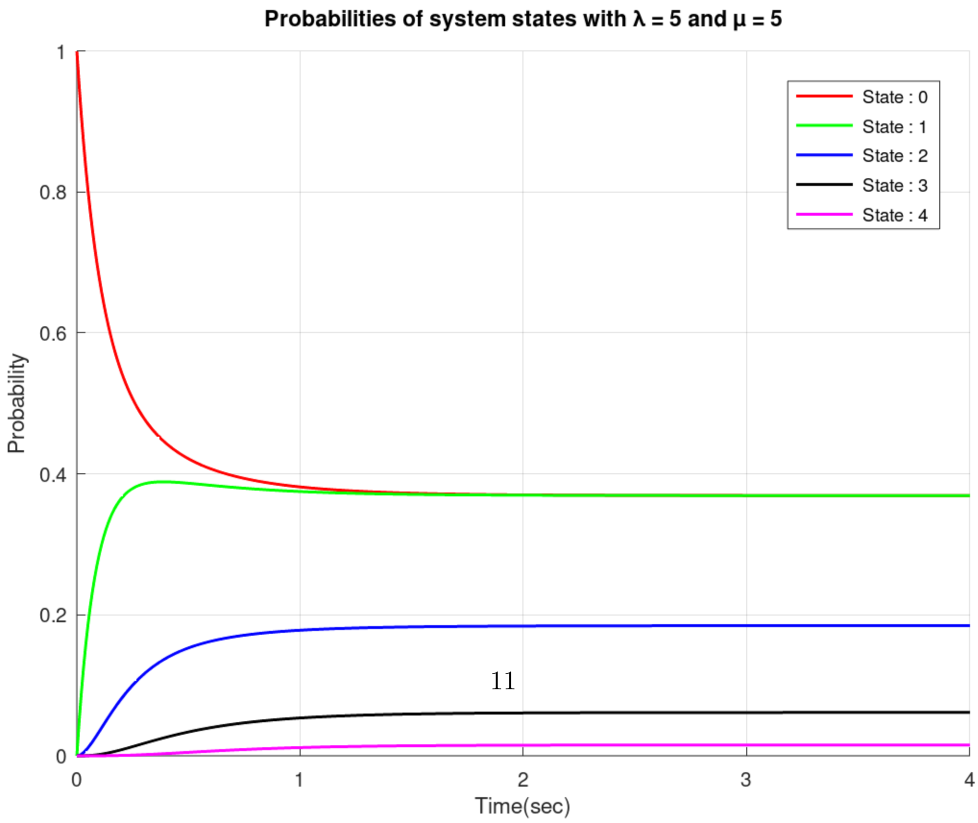
4.2.5 v

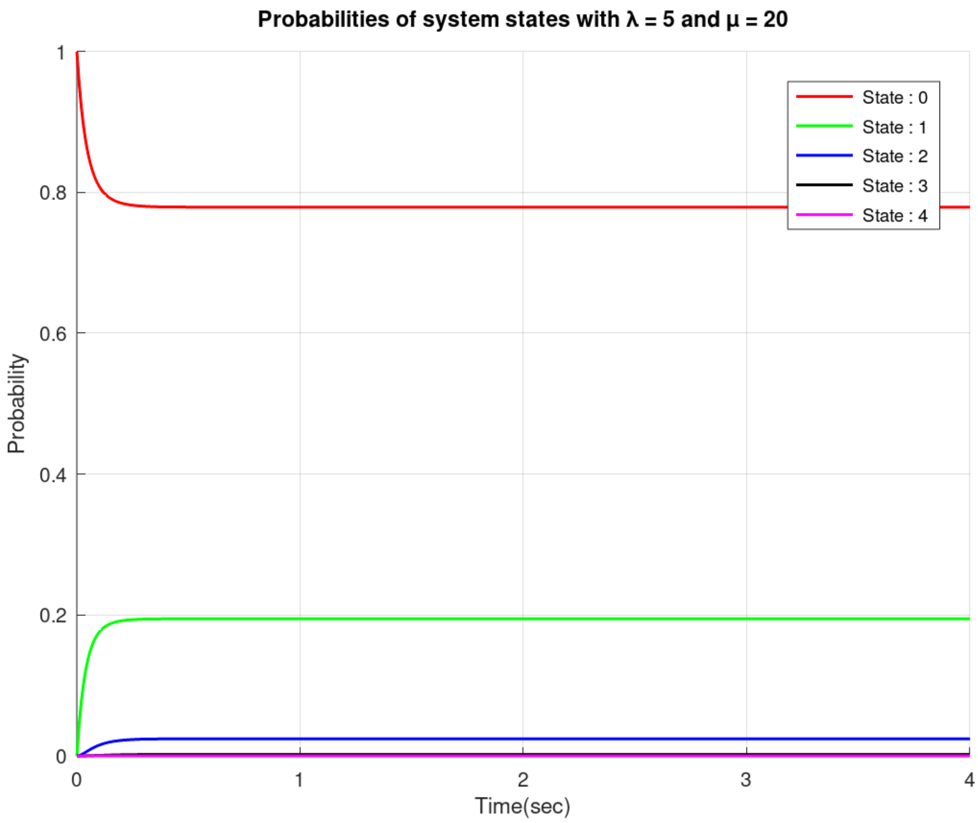


Probabilities of system states with λ = 5 and μ = 10
State : 0
State : 1
State : 2
State : 3
State : 4
Probability
Time(sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.8
1

4.2.6 vi


Probabilities of system states with λ = 5 and μ = 1
Probability
Time(sec)
State : 0
State : 1
State : 2
State : 3
State : 4


Probabilities of system states with λ = 5 and μ = 5
Probability
Time(sec)
State : 0
State : 1
State : 2
State : 3
State : 4



We notice that as $\frac{\lambda}{\mu}$ decrease then the probabilities shrink to the initial state, while the time until probabilities to be less than 1% from the ergodic probabilities also decrease with result the increase of convocation speed.

4.3 Code

```
pkg load statistics
pkg load queueing

% system M/M/1/4
% when there are 3 clients in the system, the capability of the server doubles.

clc;
clear all;
close all;

# (i)

lambda = 5;
mu = 10;
states = [0, 1, 2, 3, 4]; % system with capacity 4 states
% the initial state of the system. The system is initially empty.
initial_state = [1, 0, 0, 0, 0];

```

```
% define the birth and death rates between the states of the system.
births_B = [lambda, lambda/2, lambda/3, lambda/4];
deaths_D = [mu, mu, mu, mu];

% get the transition matrix of the birth-death process
transition_matrix = ctmcbd(births_B, deaths_D);
display (transition_matrix)

# (ii)
% get the ergodic probabilities of the system
P = ctmc(transition_matrix);
display (P);
figure(1);
hold on;
title("Bar of Probabilities per state")
xlabel("State")
ylabel("Probability")
bar(states, P, "b", 0.5);
grid on;
saveas (1, "figures3/figure3_1.png")
hold off;
# (iii)
display( " Average Number of customers in the system : ")
display( sum(P.*[0,1,2,3,4]))

# (iv)
display( " Probability of blocking a customer :")
display( P(5) )

# (v)
% plot the ergodic probabilities (bar for bar chart)
index = 0;
for T = 0 : 0.01 : 50
  index = index + 1;
  P0 = ctmc(transition_matrix, T, initial_state);
  Prob0(index) = P0(1);
  Prob1(index) = P0(2);
  Prob2(index) = P0(3);
  Prob3(index) = P0(4);
  Prob4(index) = P0(5);
  if P0 - P < 0.01
    break;
  endif
endfor

T = 0 : 0.01 : T;
figure(2);
title(strjoin({"Probabilities of system states with \\lambda = ",num2str(lambda),"
    and \\mu = ",num2str(mu)},""))
xlabel("Time(sec)")
ylabel("Probability")
hold on;
plot(T, Prob0, "r", "linewidth", 1.5);
plot(T, Prob1, "g", "linewidth", 1.5);
plot(T, Prob2, "b", "linewidth", 1.5);
plot(T, Prob3, "k", "linewidth", 1.5);
plot(T, Prob4, "m", "linewidth", 1.5);
legend("State : 0","State : 1","State : 2","State : 3","State : 4");
grid on;
saveas (2, strjoin({"figures3/figure3_",num2str(2),".png"},""))
```

```
hold off;
% transient probability of state 0 until convergence to ergodic probability. Convergence takes place P0 and P differ by 0.01
mu = [1,5,20];
for i=1:columns(mu)
  deaths_D = [mu(i), mu(i), mu(i), mu(i)];
  transition_matrix = ctmcbd(births_B, deaths_D);
  index = 0;
  for T = 0 : 0.01 : 4
    index = index + 1;
    P0 = ctmc(transition_matrix, T, initial_state);
    Prob0(index) = P0(1);
    Prob1(index) = P0(2);
    Prob2(index) = P0(3);
    Prob3(index) = P0(4);
    Prob4(index) = P0(5);
    if P0 - P < 0.01
      break;
    endif
  endfor


  T = 0 : 0.01 : T;
  figure(i+2);
  title(strjoin({"Probabilities of system states with \\lambda = ",num2str(lambda)," and \\mu = ",num2str(mu(i))},""))
  xlabel("Time(sec)")
  ylabel("Probability")
  hold on;
  plot(T, Prob0, "r", "linewidth", 1.5);
  plot(T, Prob1, "g", "linewidth", 1.5);
  plot(T, Prob2, "b", "linewidth", 1.5);
  plot(T, Prob3, "k", "linewidth", 1.5);
  plot(T, Prob4, "m", "linewidth", 1.5);
  legend("State : 0","State : 1","State : 2","State : 3","State : 4");
  grid on;
  saveas (i+2, strjoin({"figures3/figure3_",num2str(i+2),".png"},""))
  hold off;
endfor


clc;
clear all;
close all;

exit;
```