Queuing Theory Exercise Series 3

Alexandros Kyriakakis (03112163)

May 2020

1 Simulation of a System M/M/1/10

1.1 Debugging

1.1.1 $\lambda = 1$

Transitions	Current State	Arrivals at Current State	Is Arrival	Is Departure
0	0	0	1	0
2	1	1	1	0
3	2	1	0	1
4	1	1	1	0
5	2	2	1	0
6	3	1	0	1
7	2	2	0	1
8	1	1	0	1
10	1	2	0	1
12	1	3	0	1
14	1	4	1	0
15	2	3	0	1
16	1	4	0	1
18	1	5	1	0
19	2	4	0	1
20	1	5	0	1
22	1	6	0	1
24	1	7	0	1
26	1	8	0	1
28	1	9	1	0
29	2	5	0	1
30	1	9	0	1

1.1.2 $\lambda = 5$

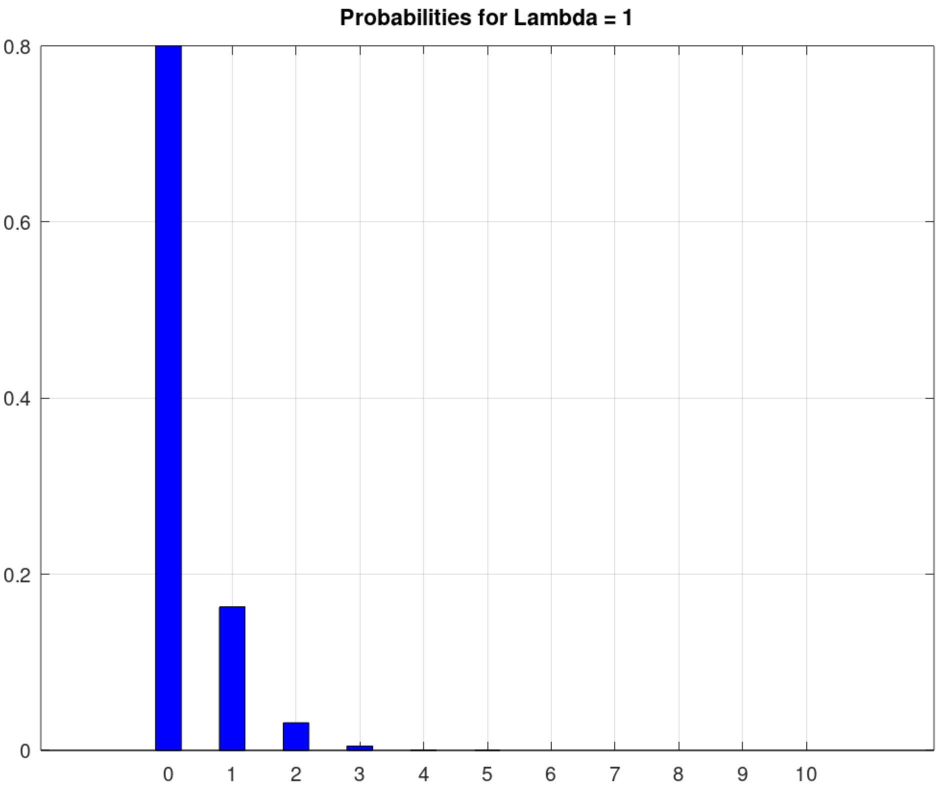
Transitions	Current State	Arrivals at Current State	Is Arrival	Is Departure
0	0	0	1	0
2	1	1	1	0
3	2	1	0	1
4	1	1	1	0
5	2	2	1	0
6	3	1	1	0
7	4	1	0	1
8	3	1	0	1
9	2	2	1	0
10	3	2	0	1
11	2	2	1	0
12	3	3	1	0
13	4	2	1	0
14	5	1	1	0
15	6	1	1	0
16	7	1	1	0
17	8	1	0	1
18	7	1	1	0
19	8	2	1	0
20	9	1	1	0
21	10	1	1	0
22	10	2	1	0
23	10	3	1	0
24	10	4	0	1
25	9	1	1	0
26	10	5	1	0
27	10	6	0	1
28	9	1	1	0
29	10	7	1	0
30	10	8	0	1

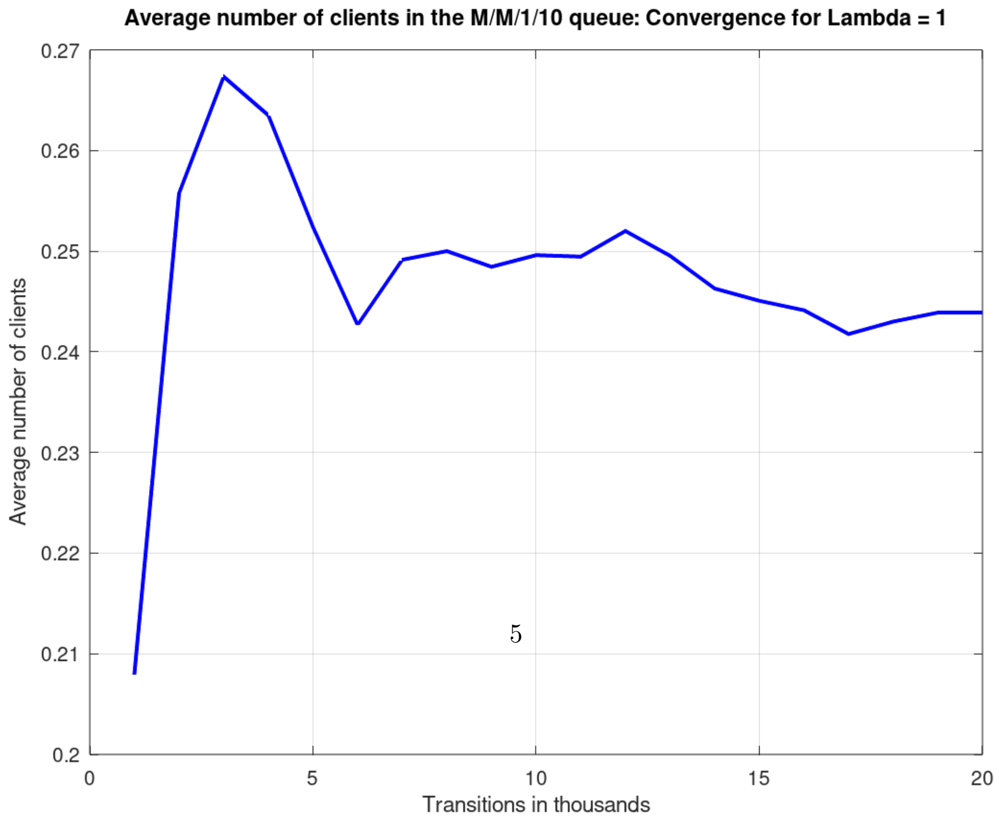
1.1.3 $\lambda = 10$

Transitions	Current State	Arrivals at Current State	Is Arrival	Is Departure
0	0	0	1	0
2	1	1	1	0
3	2	1	0	1
4	1	1	1	0
5	2	2	1	0
6	3	1	1	0
7	4	1	0	1
8	3	1	0	1
9	2	2	1	0
10	3	2	0	1
11	2	2	1	0
12	3	3	1	0
13	4	2	1	0
14	5	1	1	0
15	6	1	1	0
16	7	1	1	0
17	8	1	1	0
18	9	1	1	0
19	10	1	1	0
20	10	2	1	0
21	10	3	1	0
22	10	4	1	0
23	10	5	1	0
24	10	6	0	1
25	9	1	1	0
26	10	7	1	0
27	10	8	0	1
28	9	1	1	0
29	10	9	1	0
30	10	10	0	1

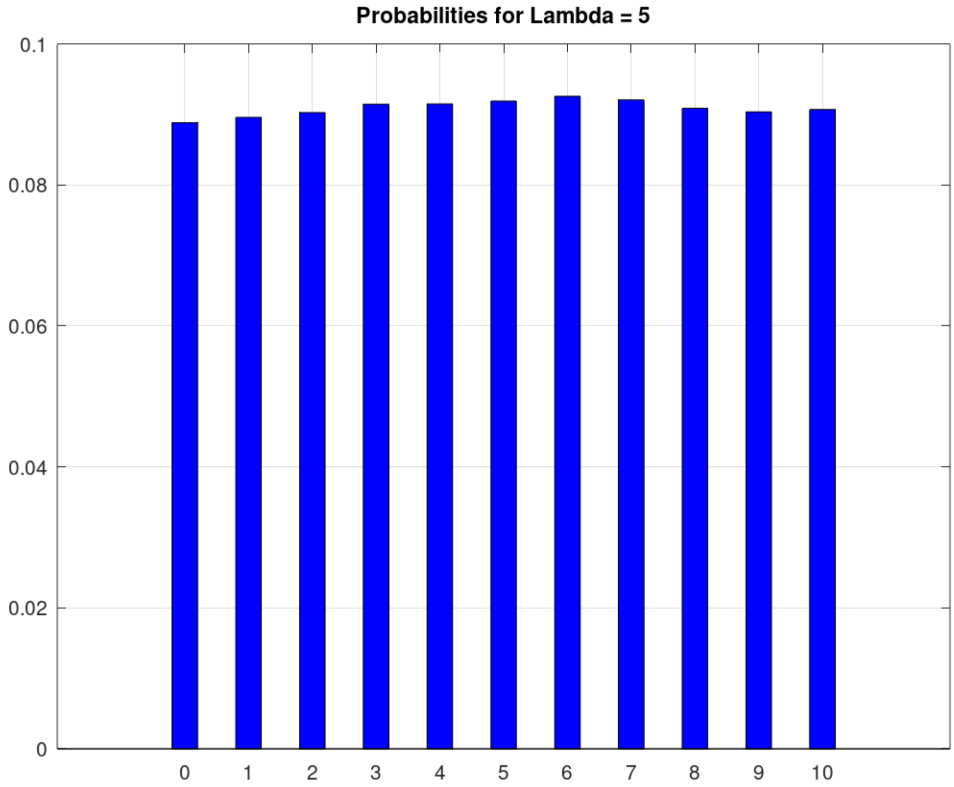
1.2 Results

1.2.1 $\lambda = 1$

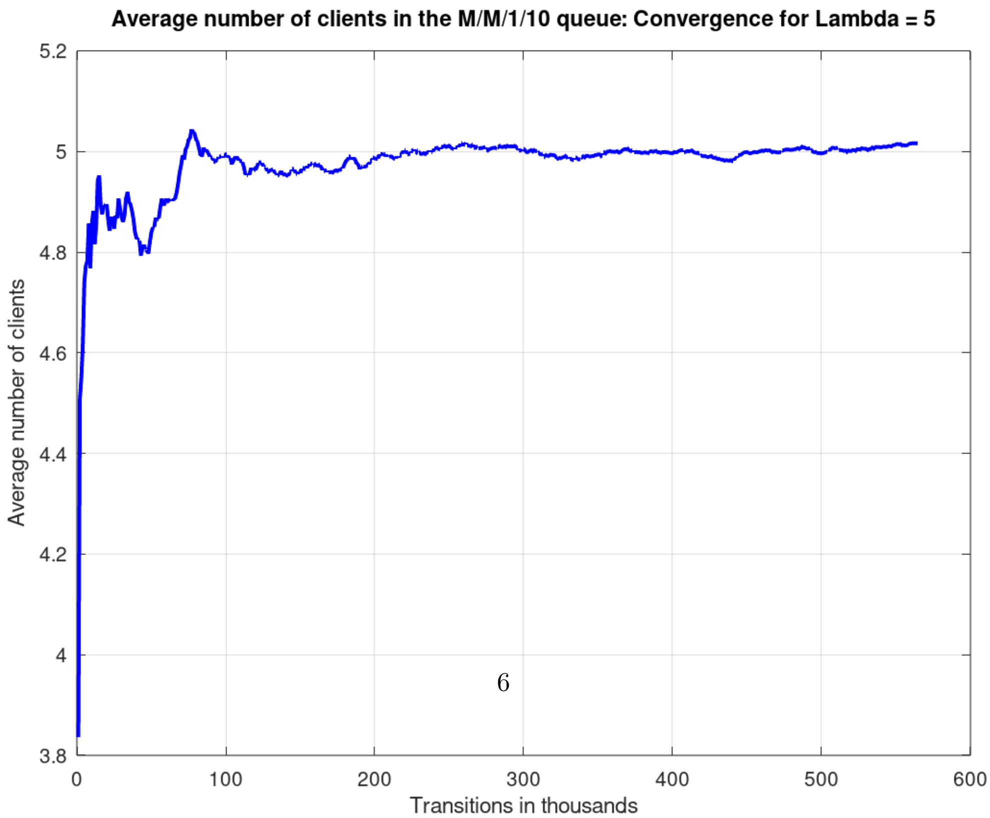




1.2.2 $\lambda = 5$

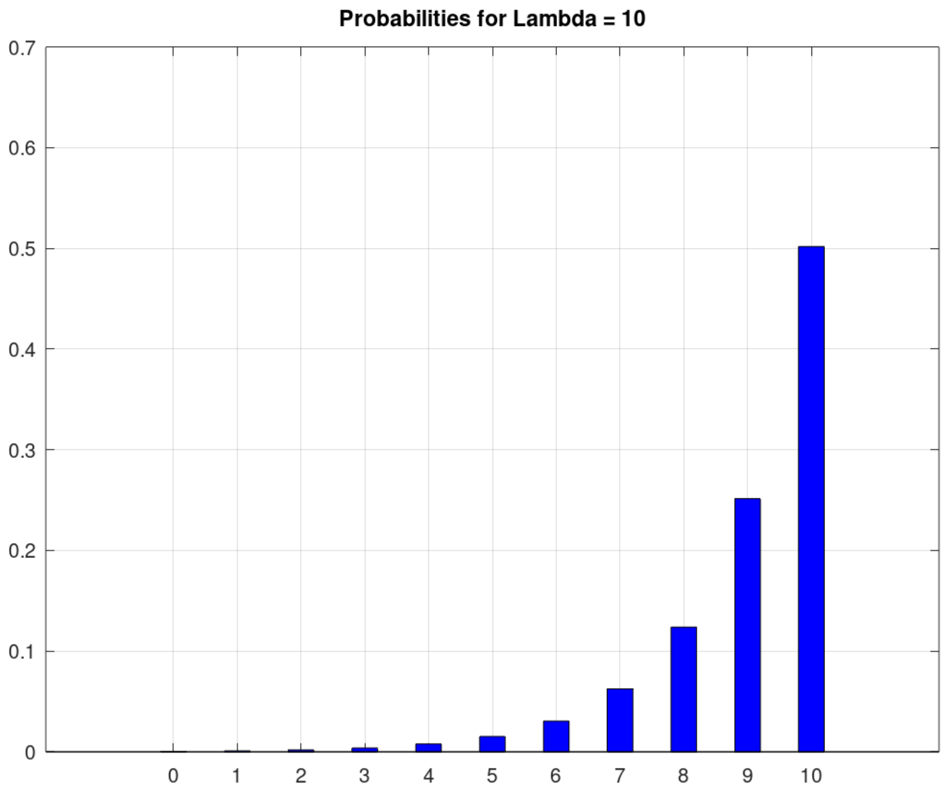


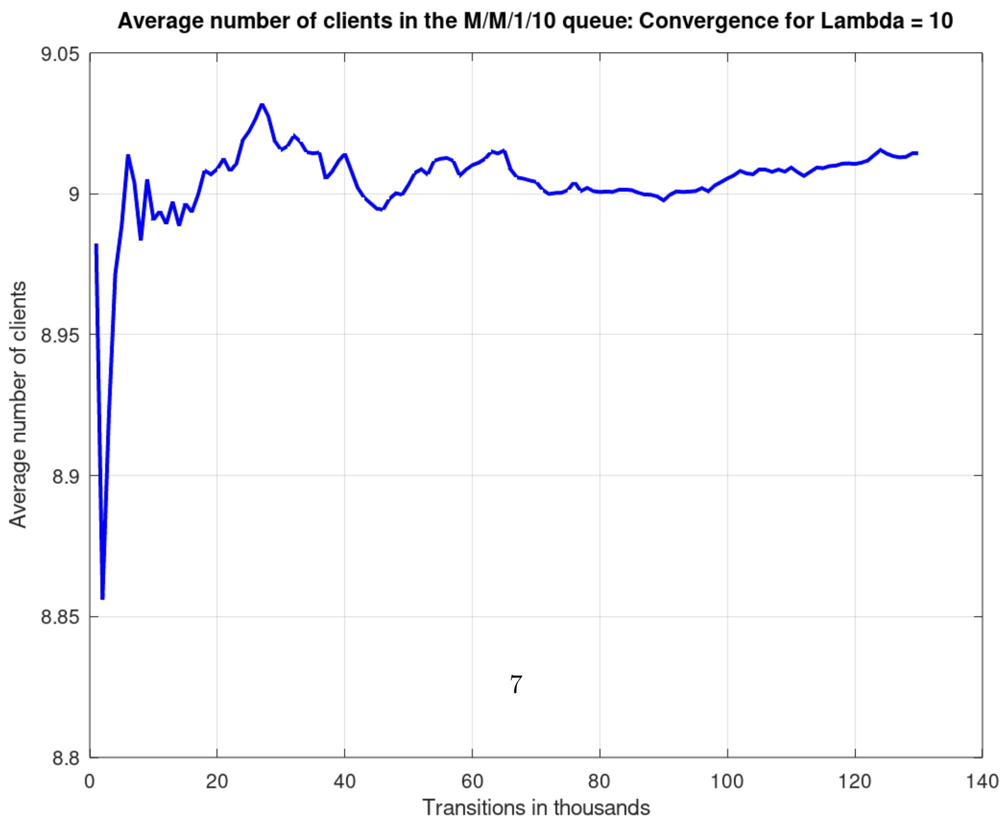
Probabilities for Lambda = 5
0.1
0.08
0.06
0.04
0.02
0
0
1
2
3
4
5
6
7
8
9
10



Average number of clients in the M/M/1/10 queue: Convergence for Lambda = 5
Average number of clients
Transitions in thousands
5.2
5
4.8
4.6
4.4
4.2
4
3.8
0
100
200
300
400
500
600

1.2.3 $\lambda = 10$





1.3 Comments

During the simulation we noticed, that for $0 \leq |\lambda - \mu| \leq \varepsilon$ while ε increases the simulation converges faster and the opposite. The number of transitions that we can ignore depends on λ which leads to ε, hence, for bigger ε we need to ignore smaller number of transitions and the opposite. So in case where $\lambda = \{1, 5, 10\}$, based on the results, ignoring at least $20K - 50K$ transitions would be safe.

1.4 Recommendations

In case, where $\mu_i = \mu \cdot (i + 1)$, $\mu = 1$, $i = \{1, 2, ..., 10\}$ where i is the current state of the system, we would change only the threshold (code-line). Basically this applies for $\lambda_i \in \{1, 5, 10\}$, we have $threshold_i = \frac{\lambda_i}{\lambda_i + current\_state + 1}$. Also we should move the initialization of threshold inside the scope of while in order to update the *current_state* variable.

2 Code

```
% M/M/1 simulation. We will find the probabilities of the first states.
% Note: Due to ergodicity, every state has a probability >0.

clc;
clear all;
close all;
rand('seed',12163);

lambdaAll = [1,5,10];
for i = lambdaAll
    arrivals = [0,0,0,0,0,0,0,0,0,0,0]
    total_arrivals = 0; % to measure the total number of arrivals
    current_state = 0;  % holds the current state of the system
    previous_mean_clients = 0; % will help in the convergence test
    index = 0;
    mu = 5;
    threshold = (i)/((i) + mu); % the threshold used to calculate probabilities

    transitions = 0; % holds the transitions of the simulation in transitions steps
    %my_i = 0
    %traceMatr = [];
    while  transitions >= 0 % && transitions < 30
        transitions = transitions + 1; % one more transitions step
        %{
        my_i = my_i + 1
        if 0 < my_i < 31 && current_state > 0
            traceMatr(my_i,1) = my_i;
            traceMatr(my_i,2) = current_state;
            traceMatr(my_i,3) = arrivals(current_state);
        endif
        %}
        if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
            index = index + 1;
            for j=1:1:length(arrivals)
                P(j) = arrivals(j)/total_arrivals; % calcuate the probability of every state in the system
            endfor

            mean_clients = 0; % calculate the mean number of clients in the system
            for j=1:1:length(arrivals)
```

```
                mean_clients = mean_clients + (j-1).*P(j);
            endfor

            to_plot(index) = mean_clients;

            if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000 % convergence test
            break;
            endif

            previous_mean_clients = mean_clients;

        endif

        random_number = rand(1); % generate a random number (Uniform distribution)
        if current_state == 0 || random_number < threshold % arrival
            %{
            if 0 < my_i < 31
                traceMatr(my_i,4) = 1
                %disp("arrival"), disp(current_state);
            endif
            %}
            total_arrivals = total_arrivals + 1;
            % to catch the exception if variable arrivals(i) is undefined. Required only for systems with finite capacity.
            x = arrivals(current_state + 1) + 1
            arrivals(current_state + 1) = x; % increase the number of arrivals in the current state
            if (current_state != 10)
                current_state = current_state + 1;
            endif
            else % departure
            %{
            if 0 < my_i < 31
                traceMatr(my_i,5) = 1
                %disp("departure"), disp(current_state);
            endif
            %}
            if current_state != 0 % no departure from an empty system
            current_state = current_state - 1;
            endif
        endif
    endwhile
    #{
    for j=1:1:length(arrivals)
        display(P(j));
    endfor
    display(traceMatr);
    pause(5);
    #}
    figure(1);
    plot(to_plot,"b","linewidth",2);
    title(strjoin({"Average number of clients in the M/M/1/10 queue: Convergence for Lambda = ",num2str((i))},""));
    xlabel("Transitions in thousands");
    ylabel("Average number of clients");
    grid on;
    saveas (1, strjoin({"figure_",num2str(1),"_lambda_",num2str((i)),".png"},""))
    figure(2);
    bar(0:1:(length(arrivals)-1),P,'b',0.4);
```

```
    title(strjoin({"Probabilities for Lambda = ",num2str((i))},""));
    grid on;
    saveas (2, strjoin({"figure_",num2str(2),"_lambda_",num2str((i)),".png"},""))

    clc;
    clear all;
    close all;

endfor
exit;
```