

# Συστήματα Αναμονής

## 3η Ομάδα Ασκήσεων

Παναγιώτης Σταματόπουλος

A.M: el20096

### Προσομοίωση του συστήματος M/M/1/10:

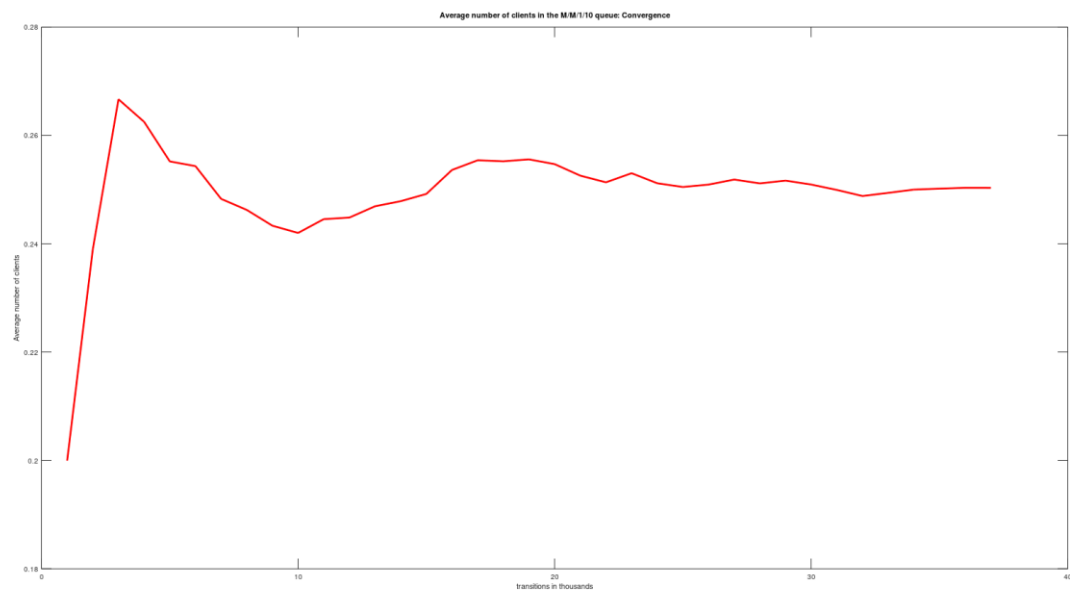
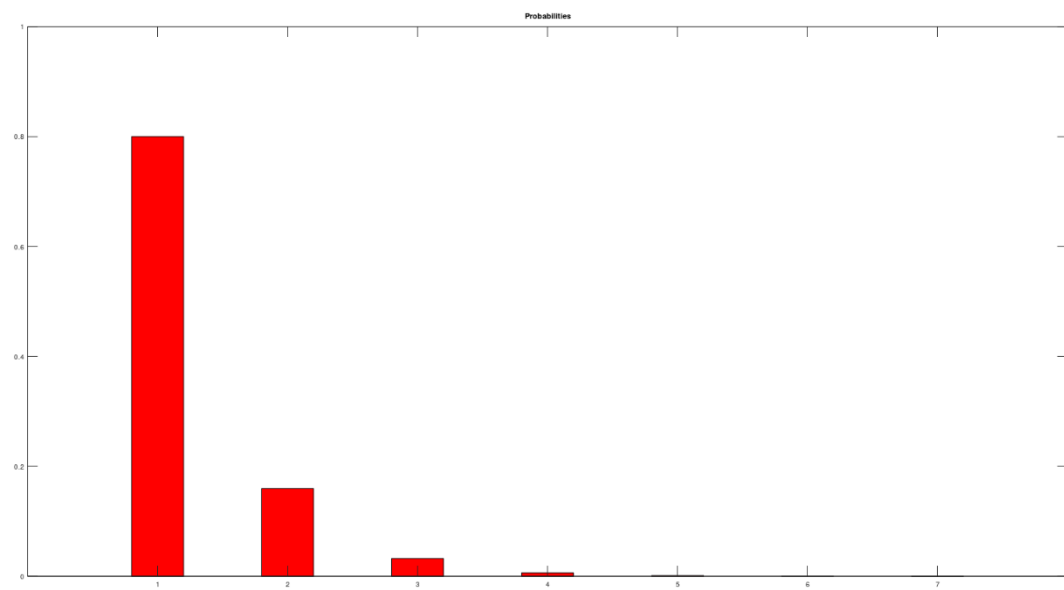
1. Στην προσομοίωση τρέχουμε το εξής commented κομμάτι κώδικα για να βεβαιωθούμε ότι δουλεύει:

```
90 % Debugging:
91
92 %for i=1:1:30
93 % disp(cstrcat("State No: ", num2str(state(i))));
94 % if(next_transition(i+1) ==1)
95 %     display("Type of next transition: Arrival");
96 % else
97 %     display("Type of next transition: Departure");
98 % endif
99 % disp(cstrcat("Total arrivals: ", num2str(t_arrivals(i)), "\n"));
.00 %endfor
.01
```

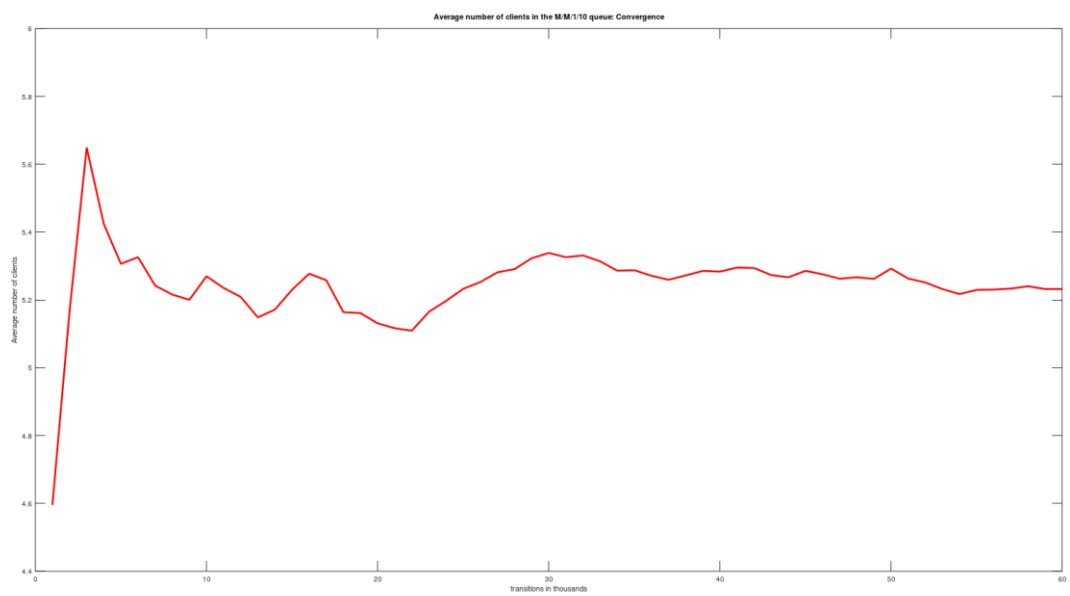
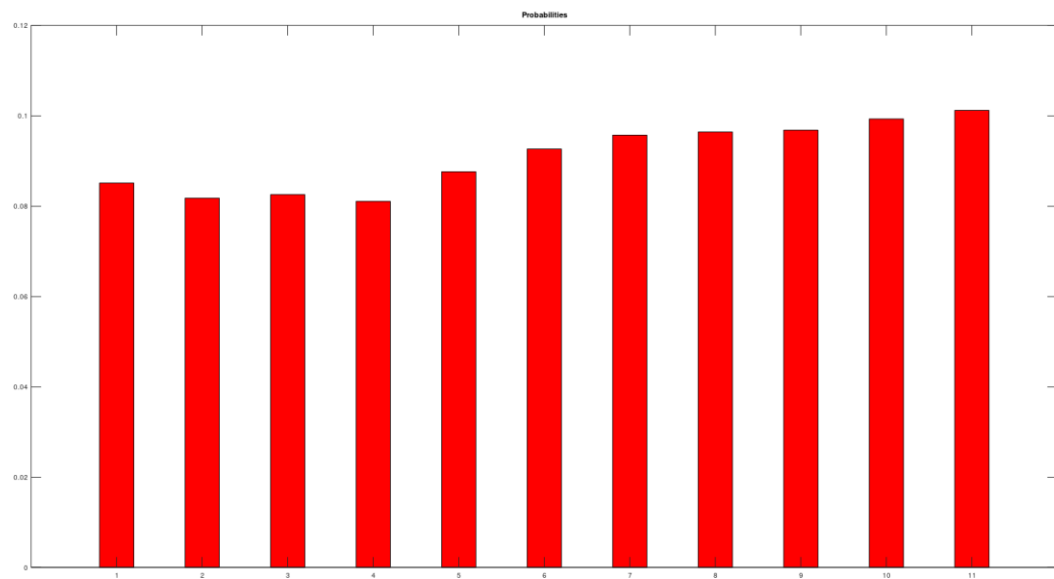
Πράγματι για τα 30 πρώτα states τα αποτελέσματα είναι σωστά, επομένως προχωρούμε στο επόμενο κομμάτι της άσκησης.

2. Εκτελούμε την προσομοίωση για κάθε τιμή του  $\lambda$  και σχεδιάζουμε τις γραφικές παραστάσεις των εργοδικών πιθανοτήτων και της εξέλιξης του μέσου αριθμού πελατών στο σύστημα.

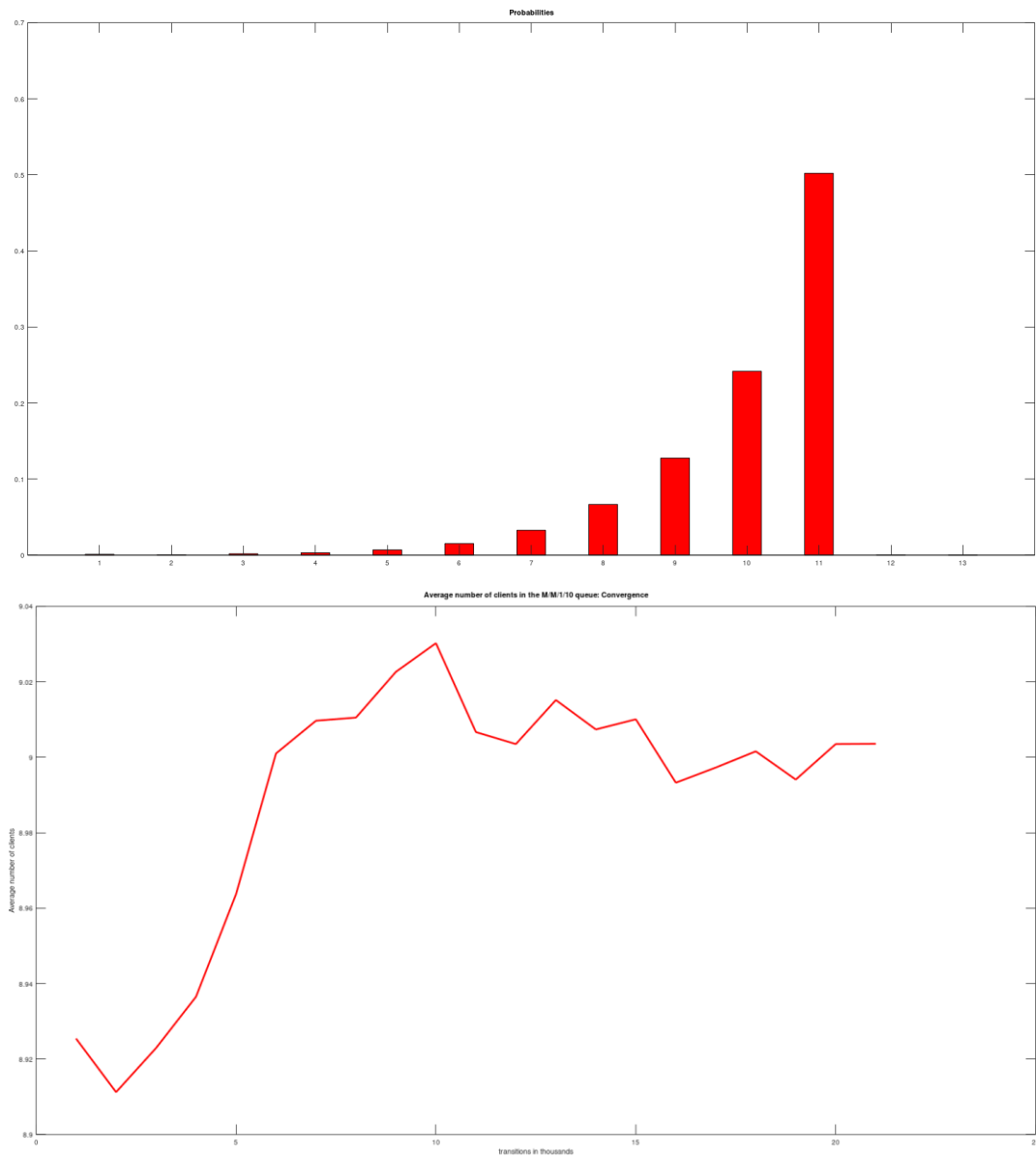
- Για  $\lambda = 1$ :



- Για  $\lambda = 5$ :



- Για  $\lambda = 10$ :



3. Παρατηρούμε ότι η ταχύτητα σύγκλισης εξαρτάται από την παράμετρο  $\lambda$ . Για  $\lambda = 1$  και  $\lambda = 10$  η γραφική παράσταση συγκλίνει γρηγορότερα από ότι για  $\lambda = 5 = \mu$ . Η ταχύτητα σύγκλισης προφανώς επηρεάζει και τον αριθμό των αρχικών μεταβάσεων που μπορούμε να παραλείψουμε με ασφάλεια. Είναι εμφανές ότι για  $\lambda = 1$  και  $\lambda = 10$  μπορούμε να παραλείψουμε λιγότερες αρχικές μεταβάσεις από ότι για  $\lambda = 5$  (περίπου τις μισές).
4. Εάν η παράμετρος  $\mu$  εξαρτάται από την κατάσταση του συστήματος, τότε το threshold αλλάζει, δηλαδή αλλάζει η τιμή της τυχαίας

μεταβλητής που ορίζει αν το συμβάν είναι άφιξη (μικρότερη από το threshold) ή αναχώρηση (μεγαλύτερη από το threshold). Επομένως χρειάζεται να επαναπροσδιορίσουμε την τιμή του threshold σε κάθε κατάσταση.

## Πηγαίος Κώδικας:

```
1 % system M/M/1/10
2 % when there are 3 clients in the system, the capability of the server doubles.
3
4 clc;
5 close all;
6
7 rand("seed",1);
8
9 for l = 1:3
10
11     lambda = [1, 5, 10];
12     mu = 5;
13
14     total_arrivals = 0; % to measure the total number of arrivals
15     current_state = 0; % holds the current state of the system
16     previous_mean_clients = 0; % will help in the convergence test
17     index = 0;
18
19     threshold = lambda./(lambda + mu);
20
21     transitions = 0;
22     total=0;
23     while transitions >= 0 & transitions < 1000000
24
25         transitions = transitions + 1; % one more transitions step
26
27         if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
28             index = index + 1;
29             for i=1:length(arrivals)
30                 P(i) = arrivals(i)/total_arrivals; % calculate the probability of every state in the system
31             endfor
32
33             mean_clients = 0; % calculate the mean number of clients in the system
34             for i=1:length(arrivals)
35                 mean_clients = mean_clients + (i-1).*P(i);
36             endfor
37
38             to_plot(index) = mean_clients;
39
40             if abs(mean_clients - previous_mean_clients) < 0.0001 % convergence test
41                 break;
42             endif
43         end
```

```

44     previous_mean_clients = mean_clients;
45
46 endif
47
48 random_number = rand(1); % generate a random number
49
50 if(current_state == 0 || random_number < threshold(1))
51     total_arrivals = total_arrivals + 1;
52     state(transitions) = current_state + 1;
53     try
54         arrivals(current_state+1) = arrivals(current_state+1) + 1;
55         state(transitions) = current_state;
56         if (current_state < 10)
57             current_state = current_state + 1;
58         endif
59         next_transition(transitions) = 1;
60         total = total + 1;
61     catch
62         arrivals(current_state + 1) = 1;
63         current_state = current_state + 1;
64         next_transition(transitions) = 1;
65         total = total + 1;
66     end
67
68 else % departure
69     state(transitions) = current_state+1;
70     if current_state != 0 % no departure from an empty system
71         current_state = current_state - 1;
72         next_transition(transitions) = 0;
73         total = total - 1;
74     endif
75 endif
76 t_arrivals(transitions) = total;
77
78 endwhile
79
80 figure(1);
81 bar(P,'r',0.4);
82 title("Probabilities");
83 figure(1+3);
84 plot(to_plot,"r","linewidth",1.3);
85 title("Average number of clients in the M/M/1/10 queue: Convergence");
86 xlabel("transitions in thousands");

```

```

87 ylabel("Average number of clients");
88
89 clear all;
90
91 endfor
92
93 % Debugging:
94
95 %for i=1:1:30
96 % disp(cstrcat("State No: ", num2str(state(i))));
97 % if(next_transition(i+1) == 1)
98 %     display("Type of next transition: Arrival");
99 % else
100 %     display("Type of next transition: Departure");
101 % endif
102 % disp(cstrcat("Total arrivals: ", num2str(t_arrivals(i)), "\n"));
103 %endfor
104

```