Συστήματα Αναμονής
3η Ομάδα Ασκήσεων

Ονοματεπώνυμο: Κουστένης Χρίστος

Α.Μ : el20227

Ακαδημαϊκό έτος: 2022-2023

Προσομοίωση συστήματος M/M/1/10

(1)

Στην αρχική προσομοίωση του συστήματος M/M/1/10 η οποία θα γίνει με την συμπερίληψη του κώδικα του debugging χρησιμοποιούμε ενδεικτικά τις τιμές λ = 5 πελάτες/min και μ = 5 πελάτες/min.

```
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;

P = [0 0 0 0 0 0 0 0 0 0 0]; %Probabilities for each state
arrivals = [0 0 0 0 0 0 0 0 0 0 0]; % arrivals on each state
lambda = 5;
mu = 5;
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

```

Κώδικας 1

Όπως φαίνεται στο πρώτο μέρος του κώδικα κάνουμε αρχικοποίηση των απαραίτητων μεταβλητών. Ειδικότερα, ορίζουμε το διάνυσμα P που σε κάθε θέση του θα περιέχει την πιθανότητα άφιξης στην κατάσταση του συστήματος με πλήθος πελατών ίσο με τον αριθμό της θέσης μείον ένα. Όμοια ορίζουμε το διάνυσμα arrivals που αποθηκεύει το πλήθος των αφίξεων για κάθε state. Επίσης, ορίζουμε τη μεταβλητή threshold = λ/(λ+μ). Αν ο αριθμός που επιστρέφει η συνάρτηση rand(1), ακολουθώντας ομοιόμορφη κατανομή στο [0,1], είναι μικρότερος του threshold τότε έχουμε άφιξη αλλιώς έχουμε αναχώρηση. Η προσομοίωση πρακτικά επιτελείται στο εσωτερικό ενός while βρόχου ο οποίος εκτελείται ώσπου να γίνει break στις γραμμές 37-39 του Κώδικα 2 όταν η σύγκριση ανάμεσα σε δύο τιμές μέσου αριθμού των πελατών έχει διαφορά μικρότερη από 0.001% ή όταν ο αριθμός των συνολικών μεταβάσεων ξεπεράσει το 1000000. Η πρώτη συνθήκη είναι και αυτή που αποτελεί το κριτήριο σύγκλισης της προσομοίωσης μας.

```
21 while transitions >= 0
22   transitions = transitions + 1; % one more transitions step
23
24   if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
25     index = index + 1;
26     for i=1:1:length(arrivals)
27         P(i) = arrivals(i)/total_arrivals; % calcuate the probability of every state in the system
28     endfor
29
30     mean_clients = 0; % calculate the mean number of clients in the system
31     for i=1:1:length(arrivals)
32        mean_clients = mean_clients + (i-1).*P(i);
33     endfor
34
35     to_plot(index) = mean_clients;
36
37     if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000 % convergence test
38       break;
39     endif
40
41     previous_mean_clients = mean_clients;
42
43   endif
44
```

Κώδικας 2

Στη συνέχεια, για κάθε επανάληψη του βρόχου χρησιμοποιούμε τη συνάρτηση rand(1) και το threshold όπως περιγράψαμε παραπάνω για να καθορίσουμε «τυχαία» άφιξη ή αναχώρηση. Επιπρόσθετα για να αποφύγουμε μία αναχώρηση πελάτη από την 0 κατάσταση και την άφιξη πελάτη στον κατάσταση 10, ελέγχουμε την συγκεκριμένη κατάσταση στις γραμμές 46 και 60. Σε περίπτωση που βρισκόμαστε σε ενδιάμεση κατάσταση ανανεώνεται ο μετρητής για την παρούσα κατάσταση current_state και όταν χρειάζεται αυξάνουμε και τον μετρητή των αφίξεων σε κάθε κατάσταση.

```
45   random_number = rand(1); % generate a random number (Uniform distribution)
46   if current_state == 0 || random_number < threshold % arrival
47     if current_state <= 10
48       total_arrivals = total_arrivals + 1;
49       if transitions <= 30  % DEBUGGING
50         display("Current state :");
51         disp(current_state);
52         display("Next transition");
53         disp(transitions);
54         display("and it is");
55         display("ARRIVAL");
56         display("Total arrivals in current state: ");
57         disp(arrivals(current_state+1));
58       endif %DEBUGGING
59       arrivals(current_state+1) = arrivals(current_state+1)+1 ;
60       if current_state < 10
61         current_state = current_state + 1; % if we are not at maximum capacity accept
62       endif                                %the new client and increase current state
63     endif
64   else % departure
65     if current_state != 0 % no departure from an empty system
66       if transitions <= 30  % DEBUGGING
67         display("Current state = ");
68         disp(current_state);
69         display("Next transition");
70         disp(transitions);
71         display("and it is")
72         display("DEPARTURE");
73         display("Total arrivals in current state: ");
74         disp(arrivals(current_state+1));
75       endif %DEBUGGING
76       current_state = current_state - 1;
77     endif
78   endif
79 endwhile
```

Κώδικας 3

Το τμήμα κώδικα που αφορά στο ζητούμενο debugging διακρίνεται με τη χρήση κατάλληλων comments.

Τέλος, εκτυπώνουμε κατάλληλα τις πιθανότητες άφιξης σε κάθε κατάσταση, την πιθανότητα απόρριψης, τον μέσο χρόνο καθυστέρησης κάθε πελάτη, τη γραφική που δείχνει τη σύγκλιση του μέσου αριθμού πελατών στο σύστημα και των πιθανότητων με χρήση της bar().

```
display("State probabilities:")
for i=1:1:length(arrivals)
  display(P(i));
endfor

P_blocking = P(11);
display("P{Blocking} = P{state = 10} = ");
disp(P_blocking);

display("The number of mean_clients when the simulation ends:");
disp(previous_mean_clients);
g = lambda*(1-P_blocking);
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);


figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence with λ=5 clients/min");
xlabel("transitions in thousands");
ylabel("Average number of clients");

figure(2);
state = [0 1 2 3 4 5 6 7 8 9 10];
bar(state,P,'r',0.4);
title("Probabilities of each of the 11 states λ=5 clients/min");
```

Κώδικας 4

Ακολουθεί η έξοδος του προγράμματος με χρήση του Debugging κώδικα.

```
Current state :
0
Next transition
1
and it is
ARRIVAL
Total arrivals in current state:
0
Current state =
1
Next transition
2
and it is
DEPARTURE
Total arrivals in current state:
0
Current state :
0
Next transition
3
and it is
ARRIVAL
Total arrivals in current state:
1
Current state =
1
Next transition
4
and it is
DEPARTURE
Total arrivals in current state:
0
Current state :
0
Next transition
5
and it is
ARRIVAL
Total arrivals in current state:
2
Current state :
1
Next transition
6
and it is
ARRIVAL
Total arrivals in current state:
0
Current state =
2
Next transition
7
and it is
DEPARTURE
Total arrivals in current state:
0
Current state :
1
```

```
Next transition
8
and it is
ARRIVAL
Total arrivals in current state:
1
Current state :
2
Next transition
9
and it is
ARRIVAL
Total arrivals in current state:
0
Current state =
3
Next transition
10
and it is
DEPARTURE
Total arrivals in current state:
0
Current state =
2
Next transition
11
and it is
DEPARTURE
Total arrivals in current state:
1
Current state :
1
Next transition
12
and it is
ARRIVAL
Total arrivals in current state:
2
Current state =
2
Next transition
13
and it is
DEPARTURE
Total arrivals in current state:
1
Current state :
1
```

```
Next transition
14
and it is
ARRIVAL
Total arrivals in current state:
3
Current state :
2
Next transition
15
and it is
ARRIVAL
Total arrivals in current state:
1
Current state :
3
Next transition
16
and it is
ARRIVAL
Total arrivals in current state:
0
Current state =
4
Next transition
17
and it is
DEPARTURE
Total arrivals in current state:
0
Current state :
3
Next transition
18
and it is
ARRIVAL
Total arrivals in current state:
1
Current state =
4
Next transition
19
and it is
DEPARTURE
Total arrivals in current state:
0
Current state =
3
Next transition
20
and it is
DEPARTURE
Total arrivals in current state:
2
Current state :
2
```

```
Next transition
21
and it is
ARRIVAL
Total arrivals in current state:
2
Current state =
3
Next transition
22
and it is
DEPARTURE
Total arrivals in current state:
2
Current state :
2
Next transition
23
and it is
ARRIVAL
Total arrivals in current state:
3
Current state :
3
Next transition
24
and it is
ARRIVAL
Total arrivals in current state:
2
Current state =
4
Next transition
25
and it is
DEPARTURE
Total arrivals in current state:
0
Current state =
3
Next transition
26
and it is
DEPARTURE
Total arrivals in current state:
3
Current state =
2
```

```
Next transition
27
and it is
DEPARTURE
Total arrivals in current state:
4
Current state =
1
Next transition
28
and it is
DEPARTURE
Total arrivals in current state:
4
Current state :
0
Next transition
29
and it is
ARRIVAL
Total arrivals in current state:
3
Current state =
1
Next transition
30
and it is
DEPARTURE
Total arrivals in current state:
4
State probabilities:
0.090065
0.088552
0.088776
0.089632
0.090606
0.091374
0.091105
0.091441
0.092163
0.093047
0.093239
P{Blocking} = P{state = 10} =
0.093239
The number of mean_clients when the simulation ends:
5.0481
Average delay time =
1.1134
>>
```

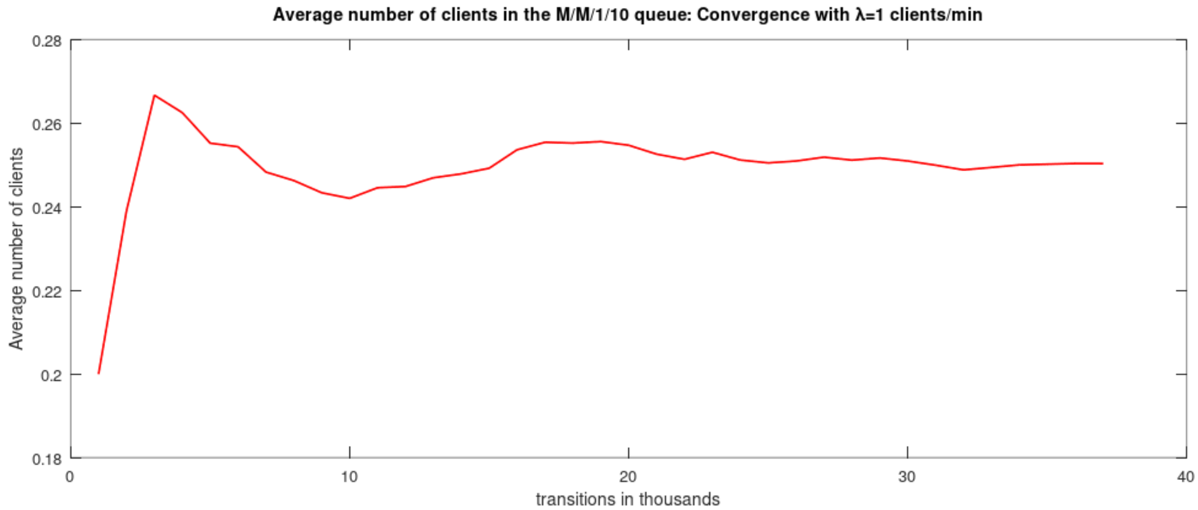
Σημείωση: Οι γραφικές παραλείπονται αφού θα συμπεριληφθούν στο ερώτημα (2) στην περίπτωση του λ = 5 πελάτες/min.

(2)

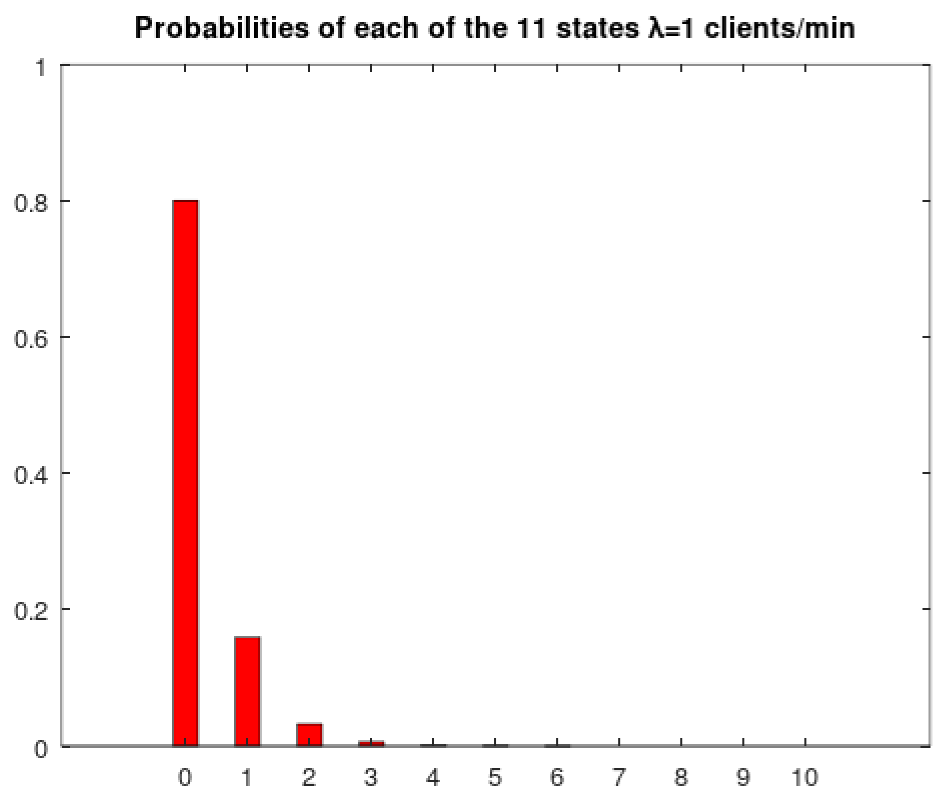
Στο μέρος αυτό παρουσιάζουμε

(α) τις εργοδικές πιθανότητες που υπολογίζει τελικά η προσομοίωσή μας και (β) την εξέλιξη του μέσου αριθμού πελατών στο σύστημα για τις τιμές λ = {1,5,10}.

- $\lambda = 1$

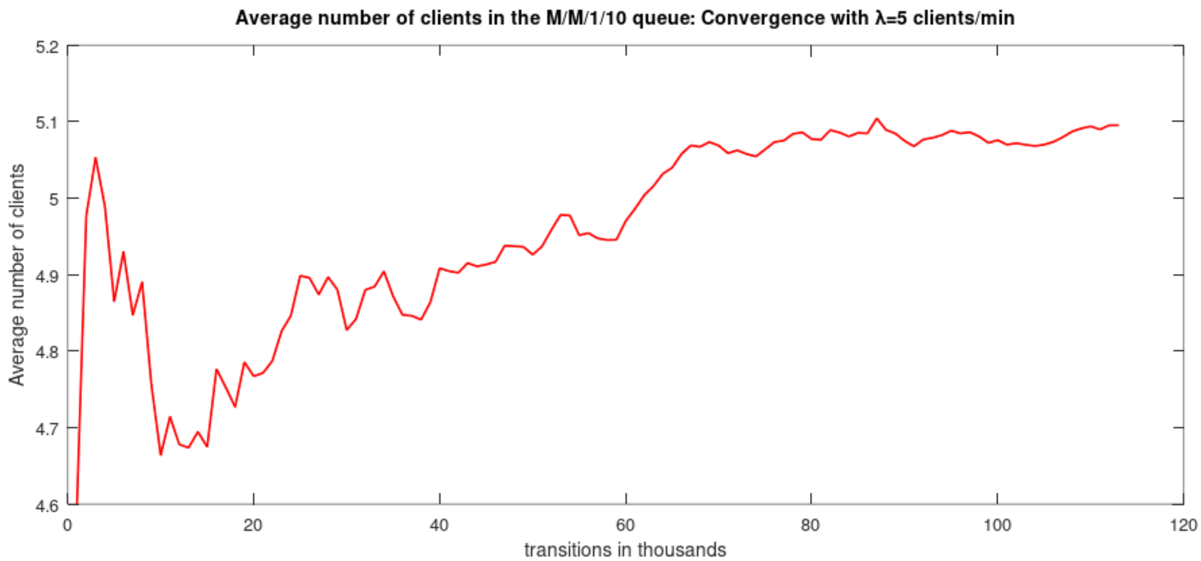


Γραφική 1

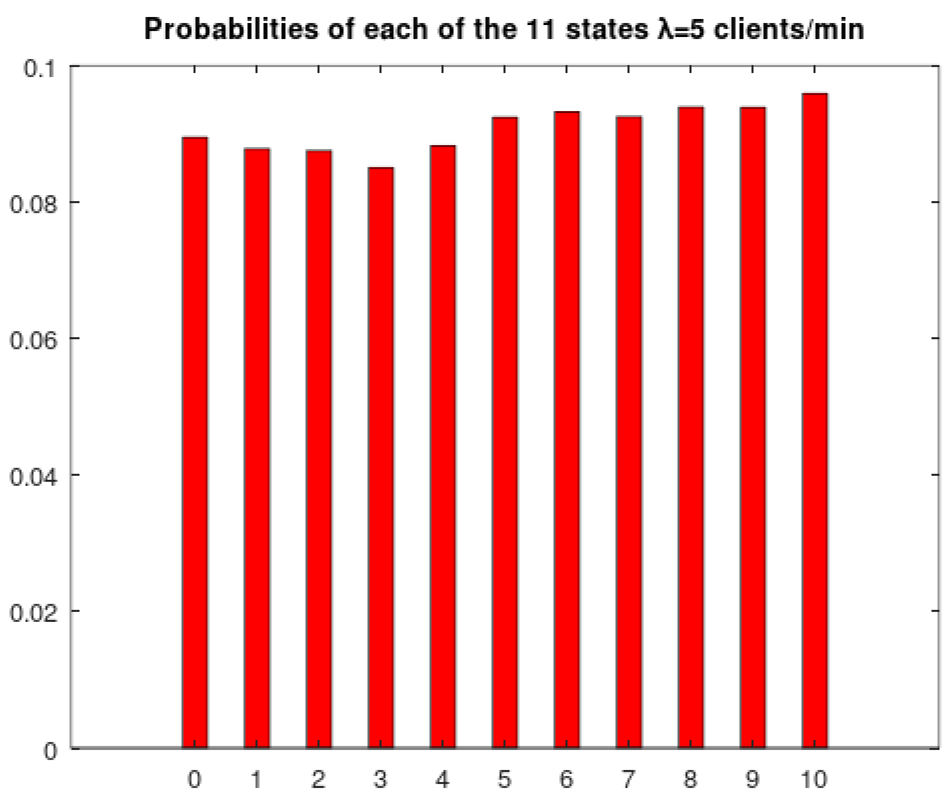


Γραφική 2

- $\lambda = 5$

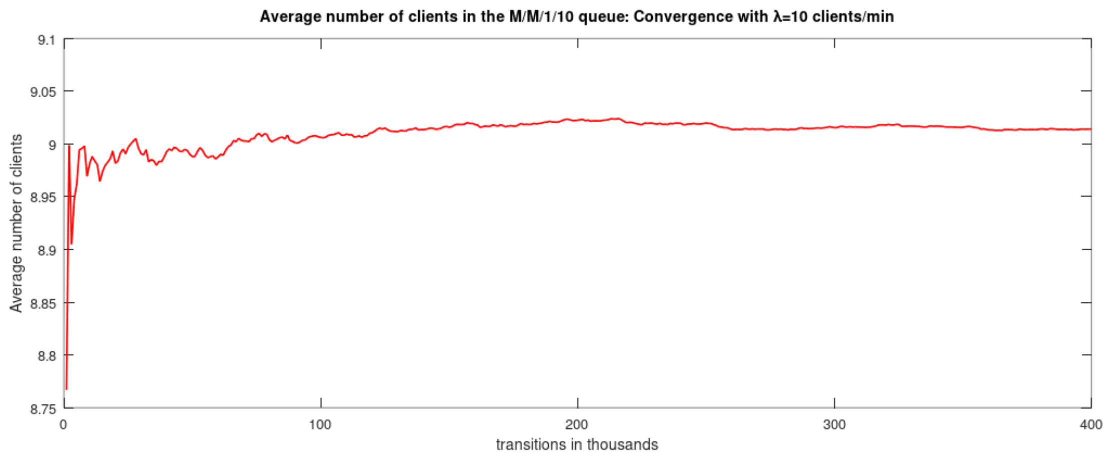


Γραφική 3

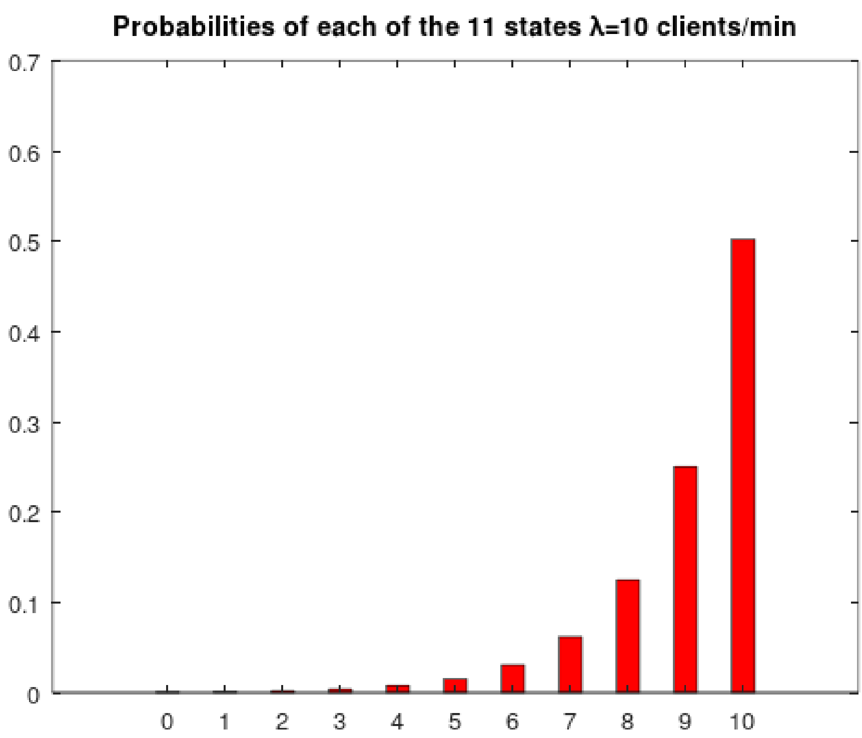


Γραφική 4

- $\lambda = 10$



Γραφική 5



Γραφική 6

Για τα παραπάνω αποτελέσματα χρησιμοποιήσαμε τρία .m αρχεία καθένα απο τα οποία εκτελέστηκε για διαφορετική τιμη του λ. Έτσι, για λ = 1 έχουμε το lab3_2_lambda1.m :

```
% M/M/1/10 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;

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;

P = [0 0 0 0 0 0 0 0 0 0 0]; %Probabilities for each state
arrivals = [0 0 0 0 0 0 0 0 0 0 0]; % arrivals on each state
lambda = 1;
mu = 5;
rand("seed", 1);
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/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 i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    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 current_state <= 10
      total_arrivals = total_arrivals + 1;
      arrivals(current_state+1) = arrivals(current_state+1)+1 ;
      if current_state < 10
        current_state = current_state + 1; % if we are not at maximum capacity accept
      endif                                %the new client and increase current state
    endif
  else % departure
```

Κώδικας 5

```
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile


display("Transitions needed for Convergence");
disp(transitions);

display("State probabilities:")
for i=1:1:length(arrivals)
  display(P(i));
endfor

P_blocking = P(11);
display("P{Blocking} = P{state = 10} = ");
disp(P_blocking);

display("The number of mean_clients when the simulation ends:");
disp(previous_mean_clients);

g = lambda*(1-P_blocking);
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);


figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence with λ=1 clients/min");
xlabel("transitions in thousands");
ylabel("Average number of clients");

figure(2);
state = [0 1 2 3 4 5 6 7 8 9 10];
bar(state,P,'r',0.4);
title("Probabilities of each of the 11 states λ=1 clients/min");
```

Κώδικας 6

Για λ = 5 έχουμε το lab3_2_lambda5.m :

```
% M/M/1/10 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;

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;

P = [0 0 0 0 0 0 0 0 0 0 0]; %Probabilities for each state
arrivals = [0 0 0 0 0 0 0 0 0 0 0]; % arrivals on each state
lambda = 5;
mu = 5;
rand("seed", 1);
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/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 i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    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 current_state <= 10
      total_arrivals = total_arrivals + 1;
      arrivals(current_state+1) = arrivals(current_state+1)+1 ;
      if current_state < 10
        current_state = current_state + 1; % if we are not at maximum capacity accept
      endif                                %the new client and increase current state
    endif
  else % departure
```

Κώδικας 7

```
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile

display("Transistions needed for Convergence");
disp(transitions);


display("State probabilities:")
for i=1:1:length(arrivals)
  display(P(i));
endfor

P_blocking = P(11);
display("P{Blocking} = P{state = 10} = ");
disp(P_blocking);

display("The number of mean_clients when the simulation ends:");
disp(previous_mean_clients);

g = lambda*(1-P_blocking);
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);


figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence with λ=5 clients/min");
xlabel("transitions in thousands");
ylabel("Average number of clients");

figure(2);
state = [0 1 2 3 4 5 6 7 8 9 10];
bar(state,P,'r',0.4);
title("Probabilities of each of the 11 states λ=5 clients/min");
```

Κώδικας 8

και για λ = 10 το lab3_2_lambda10.m :

```
% M/M/1/10 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;

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;

P = [0 0 0 0 0 0 0 0 0 0 0]; %Probabilities for each state
arrivals = [0 0 0 0 0 0 0 0 0 0 0]; % arrivals on each state
lambda = 10;
mu = 5;
rand("seed", 1);
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities

transitions = 0; % holds the transitions of the simulation in transitions steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/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 i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    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 current_state <= 10
      total_arrivals = total_arrivals + 1;
      arrivals(current_state+1) = arrivals(current_state+1)+1 ;
      if current_state < 10
        current_state = current_state + 1; % if we are not at maximum capacity accept
      endif                                %the new client and increase current state
    endif
  else % departure
```

Κώδικας 9

```
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile

display("Transistions needed for Convergence");
disp(transitions);

display("State probabilities:")
for i=1:1:length(arrivals)
  display(P(i));
endfor

P_blocking = P(11);
display("P{Blocking} = P{state = 10} = ");
disp(P_blocking);

display("The number of mean_clients when the simulation ends:");
disp(previous_mean_clients);

g = lambda*(1-P_blocking);
average_delay_time = mean_clients / g;
display("Average delay time =");
disp(average_delay_time);


figure(1);
plot(to_plot,"r","linewidth",1.3);
title("Average number of clients in the M/M/1/10 queue: Convergence with λ=10 clients/min");
xlabel("transitions in thousands");
ylabel("Average number of clients");

figure(2);
state = [0 1 2 3 4 5 6 7 8 9 10];
bar(state,P,'r',0.4);
title("Probabilities of each of the 11 states λ=10 clients/min");
```

Κώδικας 10

(3)

Με αύξηση του λ και διατηρώντας σταθερό το μ και ίσο 5 πελάτες/min γίνεται αισθητά πιο μικρή η ταχύτητα σύγκλισης της προσομοίωσης, δηλαδή ο απαιτούμενος αριθμός μεταβάσεων μέχρι να ικανοποιηθεί το κριτήριο σύγκλισης, αφού με αύξηση του λ παρατηρούμε ότι ο αριθμός μεταβάσεων που απαιτούνται για ικανοποίηση του κριτηρίου σύγκλισης αυξάνεται σημαντικά όπως φαίνεται από το μήνυμα εξόδου που τυπώνεται για κάθε τιμή του λ.

- Για λ = 1: χρειάζονται 37000 μεταβάσεις(Έξοδος 1)
- Για λ = 5: 113000 μεταβάσεις(Έξοδος 2)
- Για λ = 10: 400000 μεταβάσεις(Έξοδος 3)

Να σημειωθεί ότι για να συγκρίνουμε σωστά τις επιμέρους προσομοιώσεις χρησιμοποιήσαμε την εντολή rand(“seed”,1) στην αρχή του προγράμματος.

```
Command Window
Transistions needed for Convergence
37000
State probabilities:
0.8002
0.1596
0.032216
6.2162e-03
1.2973e-03
4.3243e-04
5.4054e-05
0
0
0
0
P{Blocking} = P{state = 10} =
0
The number of mean_clients when the simulation ends:
0.2503
Average delay time =
0.2503
>>
```

Έξοδος 1(λ=1 και μ=5)

```
Transistions needed for Convergence
113000
State probabilities:
0.089492
0.087824
0.087554
0.085027
0.088245
0.092424
0.093216
0.092508
0.093924
0.093873
0.095912
P{Blocking} = P{state = 10} =
0.095912
The number of mean_clients when the simulation ends:
5.0953
Average delay time =
1.1272
```

Έξοδος 2(λ=5 και μ=5)

```
Command Window
Transitions needed for Convergence
400000
State probabilities:
4.7928e-04
8.5372e-04
1.7449e-03
3.8567e-03
7.7284e-03
0.015446
0.030831
0.061939
0.1248
0.2501
0.5022
P{Blocking} = P{state = 10} =
0.5022
The number of mean_clients when the simulation ends:
9.0141
Average delay time =
1.8109
```

Έξοδος 3(λ=10 και μ=5)

Από τις παρατηρήσεις μας διαπιστώνουμε ότι μπορούμε με ασφάλεια να παραλείψουμε:

- Για λ = 1 : 23000 μεταβάσεις(Έξοδος 1)
- Για λ = 5 : 85000 μεταβάσεις(Έξοδος 2)
- Για λ = 10 : 200000 μεταβάσεις(Έξοδος 3).

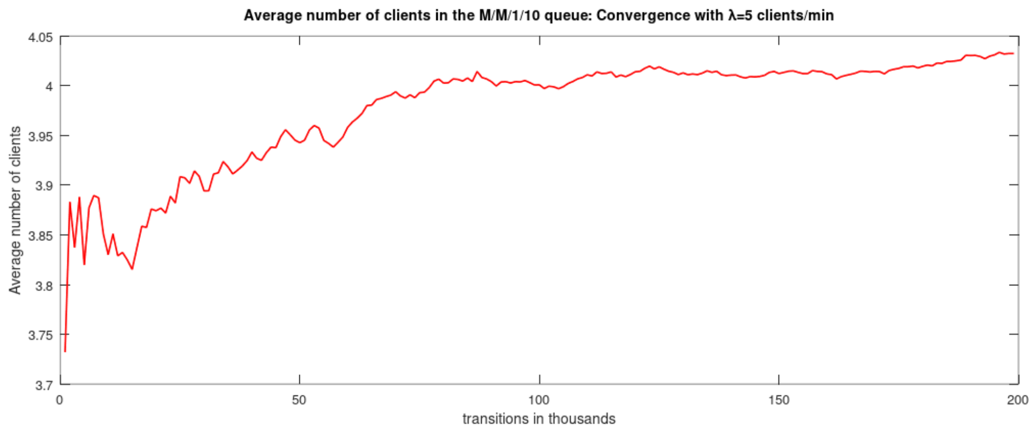
Αξίζει να σημειωθεί ότι σημαντικό ρόλο στο πλήθος των μεταβάσεων που απαιτούνται για σύγκλιση έχει και η διαφορά |λ-μ| όπως φαίνεται εκτελώντας το αρχείο lambda3_3.m. Στον κώδικα αυτού του αρχείου έχουμε θέσει μ = 1 για λ = 5 αυξάνοντας την ποσότητα |λ-μ| με αποτέλεσμα την αύξηση των μεταβάσεων που χρειάστηκαν για σύγκλιση όπως φαίνεται στην Έξοδο 4.

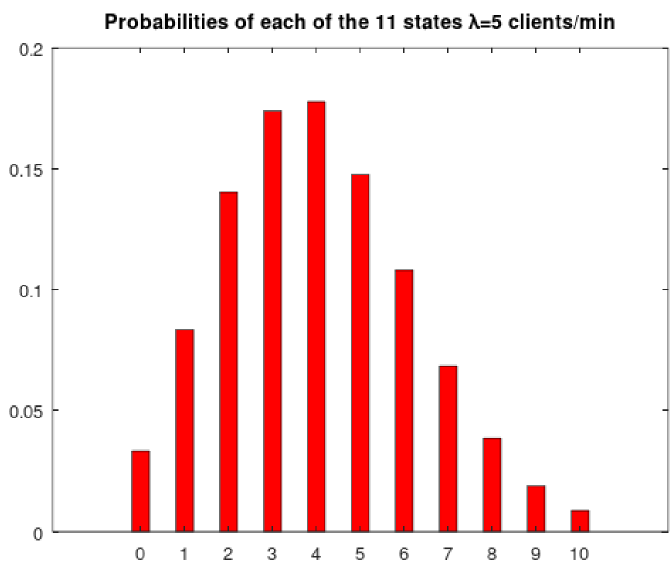
```
Transitions needed for Convergence
199000
State probabilities:
0.033410
0.083629
0.1403
0.1741
0.1779
0.1476
0.1081
0.068580
0.038603
0.019011
8.7752e-03
P{Blocking} = P{state = 10} =
8.7752e-03
The number of mean_clients when the simulation ends:
4.0324
Average delay time =
0.8136
>> |
```

Έξοδος 4(λ=5 και μ=1)

(4)

Εάν το σύστημά μας είχε εκθετικές εξυπηρετήσεις με μεταβλητό μέσο ρυθμό εξυπηρέτησης $\mu_i = \mu * (i+1)$, όπου $\mu = 1$ πελάτης/sec, και i = {1, 2, ..., 10} η κατάσταση του συστήματος μας τότε θα έπρεπε να ορίσουμε μια μεταβλητή m=1 που θα αντικαταστούσε τη mu = 5 του παρόντος προγράμματος, μια μεταβλητή mu = m*(current_state+1) εντός του while βρόχου και το threshold θα έμενε ίδιο. Ενδεικτικά υλοποιήσαμε τον κώδικα για λ = 5 και παρήγαγε τα ακόλουθα αποτελέσματα.





Σημείωση: ο κώδικας περιέχεται στο αρχείο lab3_4.m