ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

2η Εργαστηριακή Αναφορά: Προσομοίωση συστήματος M/M/1/10

Γεωργία Μπουσμπουκέα- el19059

Προσομοιώνω την διαδικασία αφίξεων και αποχωρήσεων του συστήματος M/M/1/10 (περιορισμένη χωρητικότητα-> πάντα εργοδικό) ως εξής:

Παράγω με τυχαίο τρόπο έναν αριθμό με ομοιόμορφη κατανομή στο διάστημα [0, 1], το οποίο χωρίζω με ένα threshold σε δύο διαστήματα, που αντιστοιχούν στις δύο πιθανές αλλαγές κατάστασης. Επιλέγω αυθαίρετα αν αυτός ο αριθμός είναι μικρότερος από το threshold να θεωρώ ότι έχω άφιξη, αλλιώς αναχώρηση. Άφιξη ακόμη θεωρώ πως έχω όταν είμαι στην κατάσταση 0. Ορίζω ως κατώφλι το λ/(λ+μ), έτσι ώστε το διάστημα [0, threshold] να εκφράζει την πιθανότητα άφιξης.

Σε κάθε άφιξη αυξάνω τον αριθμό των συνολικών αφίξεων στο σύστημα, ενώ συγχρόνως παρακολουθώ πόσες αφίξεις έχω από κάθε κατάσταση και, αν δεν είμαι στην κατάσταση 10, πηγαίνω στην επόμενη κατάσταση. Αντίστροφα όταν έχω αναχώρηση, αρκεί να μην είμαι στην κατάσταση 0 (μετράω τις αφίξεις και στην κατάσταση 10 έτσι ώστε να μπορώ να υπολογίσω την πιθανότητα να βρεθώ σε αυτήν την κατάσταση, όπως εξηγείται παρακάτω). Συγχρόνως, μετράω τον συνολικό αριθμό μεταβάσεων και, μόλις περάσουν 1000, βρίσκω τις πιθανότητες κάθε κατάστασης, οι οποίες υπολογίζονται ως το πηλίκο των αφίξεων από κάθε κατάσταση με τον συνολικό αριθμό αφίξεων, και τον μέσο αριθμό πελατών στο σύστημα:

$$mean\ clients = \sum_{i=0}^{N=10} i \cdot P(i)$$

Αυτήν την διαδικασία την επαναλαμβάνω έως ότου φτάσω τις 1.000.000 μεταβάσεις ή δύο μέσοι όροι πλήθους πελατών, υπολογισμ απέχουν κατά 0.001%.

<u>Η πιθανότητα απόρριψης</u> πελάτη είναι η πιθανότητα να υπάρξει άφιξη σε πλήρες σύστημα, δηλαδή η πιθανότητα να βρεθώ στην κατάσταση 10.

<u>Ο μέσος χρόνος καθυστέρησης</u> στο σύστημα ορίζεται ως το πηλίκο του μέσου αριθμού πελατών με το throughput, όπου $throughput = \gamma = \lambda \cdot \left(1 - P_{blocking}\right)$

1.

Για το debugging της προσομοίωσης παράγω trace των 30 πρώτων μεταβάσεων του συστήματος ενδεικτικά για λ=5. Συγκεκριμένα καταγράφω για αυτές τις μεταβάσεις την τρέχουσα κατάσταση, το είδος τηε επόμενης μετάβασης και τον συνολικό αριθμό αφίξεων μέχρι στιγμής στην τρέχουσα κατάσταση. Προκύπτουν:

```
Current state:
0
Next transition= arrival
Total arrivals in current state:
0
Current state:
1
Next transition= arrival
Total arrivals in current state:
0
Current state:
2
Next transition= departure
Total arrivals in currents state:
0
Current state:
1
Next transition= departure
Total arrivals in currents state:
1
Current state:
0
Next transition= arrival
Total arrivals in current state:
1
Current state:
1
Next transition= departure
Total arrivals in currents state:
1
Current state:
0
Next transition= arrival
Total arrivals in current state:
2
Current state:
1
Next transition= arrival
Total arrivals in current state:
1
Current state:
2
Next transition= departure
Total arrivals in currents state:
0
Current state:
1
Next transition= arrival
Total arrivals in current state:
2
Current state:
2
Next transition= arrival
Total arrivals in current state:
0
```

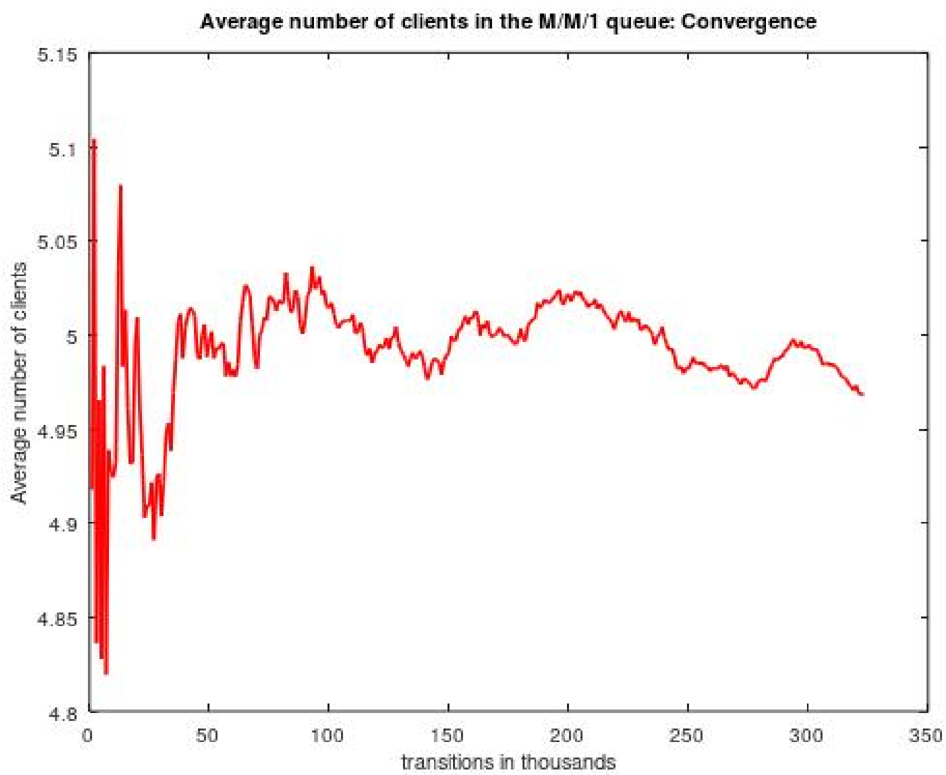
```
Current state:
4
Next transition= departure
Total arrivals in currents state:
0
Current state:
3
Next transition= departure
Total arrivals in currents state:
1
Current state:
2
Next transition= arrival
Total arrivals in current state:
1
Current state:
3
Next transition= departure
Total arrivals in currents state:
1
Current state:
2
Next transition= departure
Total arrivals in currents state:
2
Current state:
1
Next transition= departure
Total arrivals in currents state:
3
Current state:
0
Next transition= arrival
Total arrivals in current state:
3
Current state:
1
Next transition= departure
Total arrivals in currents state:
3
Current state:
0
Next transition= arrival
Total arrivals in current state:
4
Current state:
1
Next transition= arrival
Total arrivals in current state:
3
Current state:
2
Next transition= departure
Total arrivals in currents state:
2
```

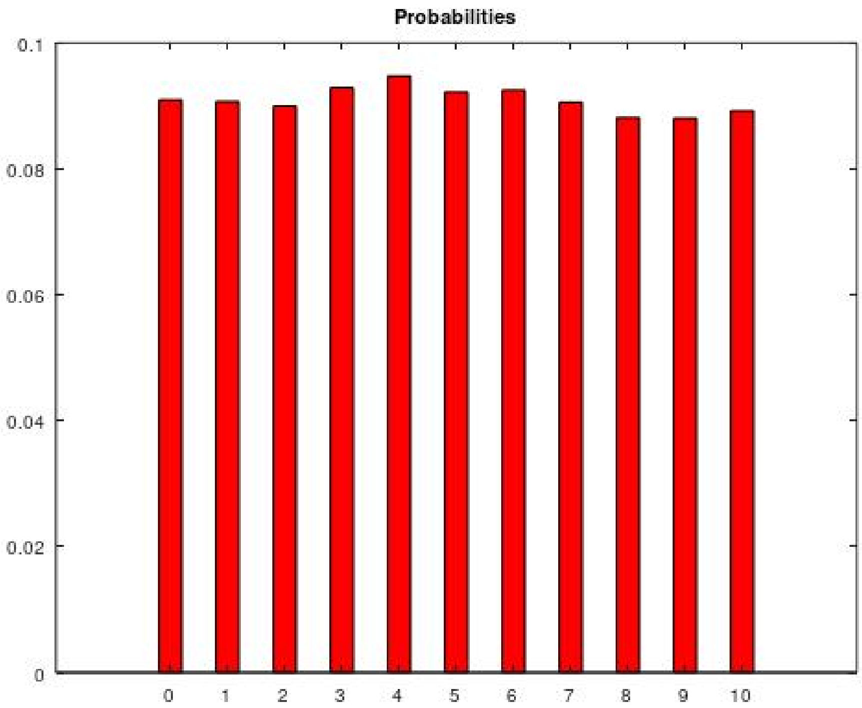
```
Current state:
1
Next transition= departure
Total arrivals in currents state:
4
Current state:
0
Next transition= arrival
Total arrivals in current state:
5
Current state:
1
Next transition= arrival
Total arrivals in current state:
4
Current state:
2
Next transition= arrival
Total arrivals in current state:
2
Current state:
3
Next transition= departure
Total arrivals in currents state:
1
Current state:
2
Next transition= departure
Total arrivals in currents state:
3
Current state:
1
Next transition= arrival
Total arrivals in current state:
5
```

Τα αποτελέσματά μας φαίνονται λογικά.

Τα υπόλοιπα ζητούμενα:

```
-
Probabilities of each state
0.090957
0.090715
0.089970
0.092898
0.094738
0.092212
0.092519
0.090603
0.088124
0.088041
0.089224
Probability that a client is rejected=
0.089224
Average delay of a client in the system=
1.0910
```





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

Κώδικας:

```
%ask1
% Note: Due to ergodicity, every state has a probability >0.

clc;
clear all;
close all;

N=10;
P=[0 0 0 0 0 0 0 0 0 0 0];
arrivals=[0 0 0 0 0 0 0 0 0 0 0]; %arrivals(i)->how many arrivals i have totally
                                   %from state i, i=0- N

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;

lambda = 5;
m = 5;
threshold = lambda/(lambda + m); % 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:N+1
        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:N+1
       mean_clients = mean_clients + (i-1).*P(i);  %mean=Σk*P(k), k=0,..., N, but the enumeration of
                                                   %array starts at one, so P(0)->P(1),..
    endfor

    to_plot(index) = mean_clients;    %keep track of mean value of clients every 1000 transitions

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

```

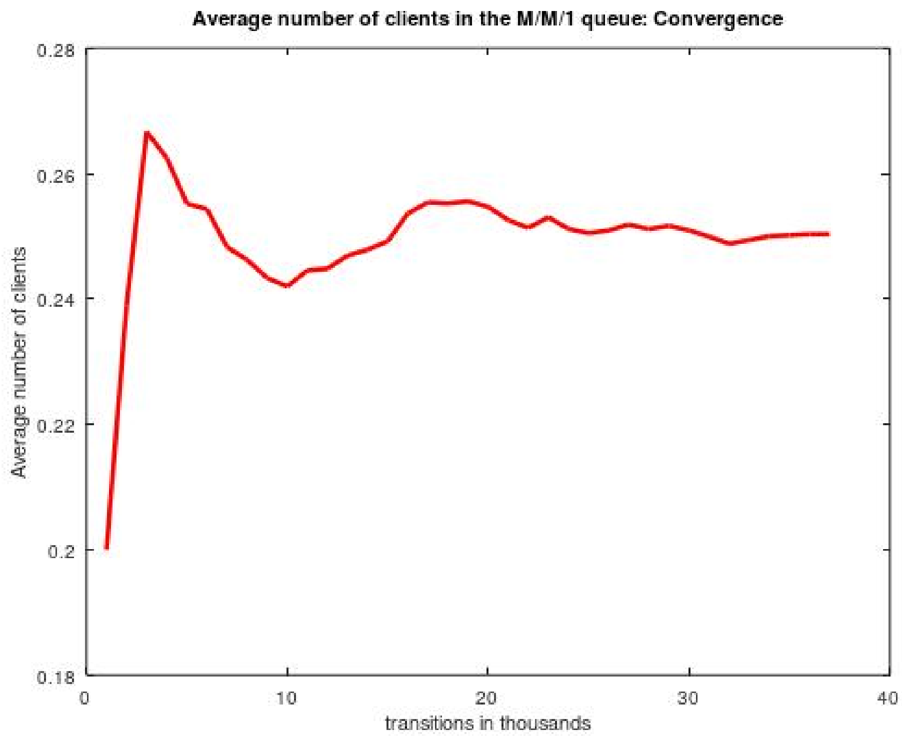
```

    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<11   %i don't count clients that come in a full system
      total_arrivals = total_arrivals + 1;
      if transitions<31    %plot the 30 first transitions
        display("Current state:")
        disp(current_state)
        display("Next transition= arrival")
        display("Total arrivals in current state:")
        disp(arrivals(current_state+1))
      endif
      arrivals(current_state + 1)++;
      if current_state<10
       current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      if transitions<31
        display("Current state:")
        disp(current_state)
        display("Next transition= departure")
        display("Total arrivals in currents state:")
        disp(arrivals(current_state+1))
      endif
      current_state = current_state - 1;
    endif
  endif
endwhile

display("Probabilities of each state");
for i=0:1:N
  display(P(i+1));
endfor

Pblocking=P(N+1);
display("Probability that a client is rejected=")
disp(Pblocking)
```

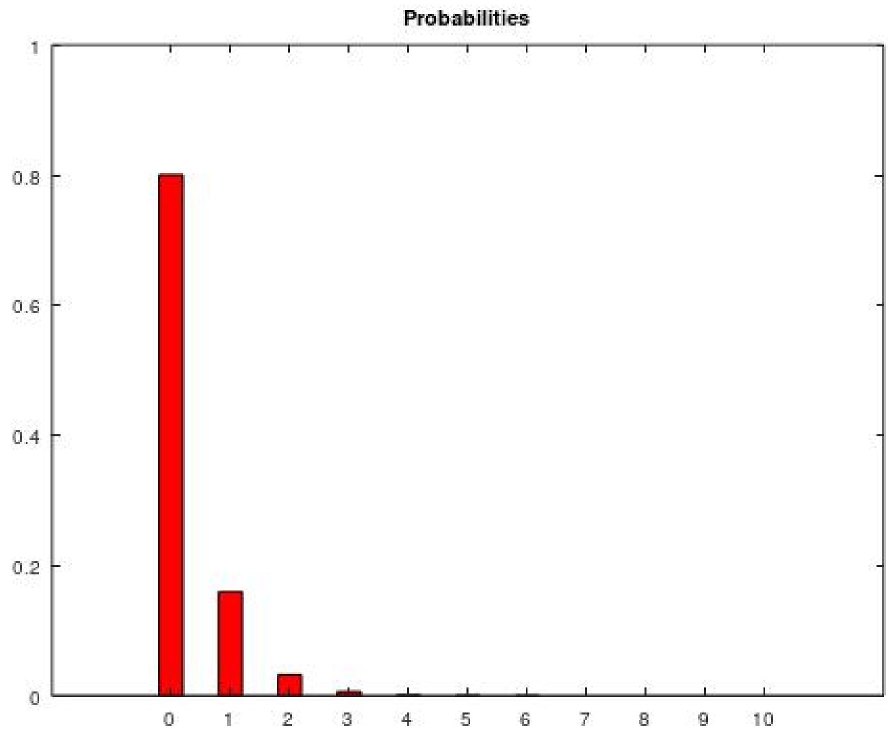
```
g=lambda*(1-P(11));
average_delay=mean_clients/g;
display("Average delay of a client in the system=")
disp(average_delay)

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

figure(2);
x=[0 1 2 3 4 5 6 7 8 9 10];
bar(x, P,'r',0.4);
title("Probabilities")
```

2.

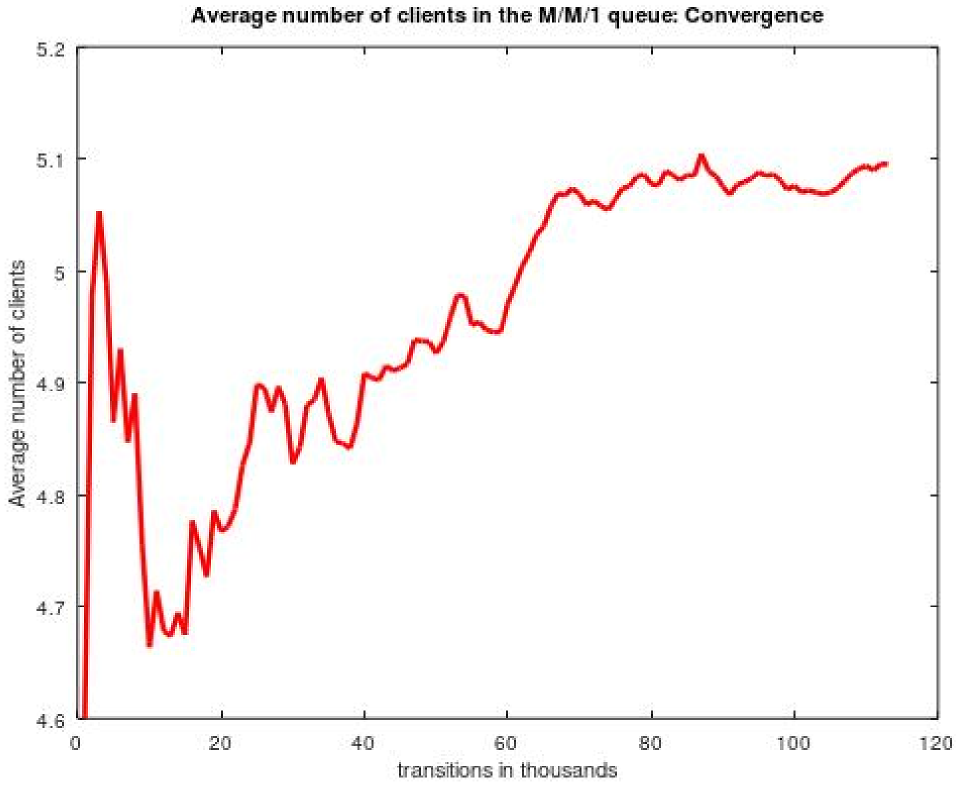
Για λ={1, 5, 10} θα παραστήσω τις εργοδικές πιθανότητες της προσομοίωσης σε ραβδόγραμμα και την εξέλιξη του μέσου αριθμού πελατών στο σύστημα. Προκύπτουν:

- Για λ=1:

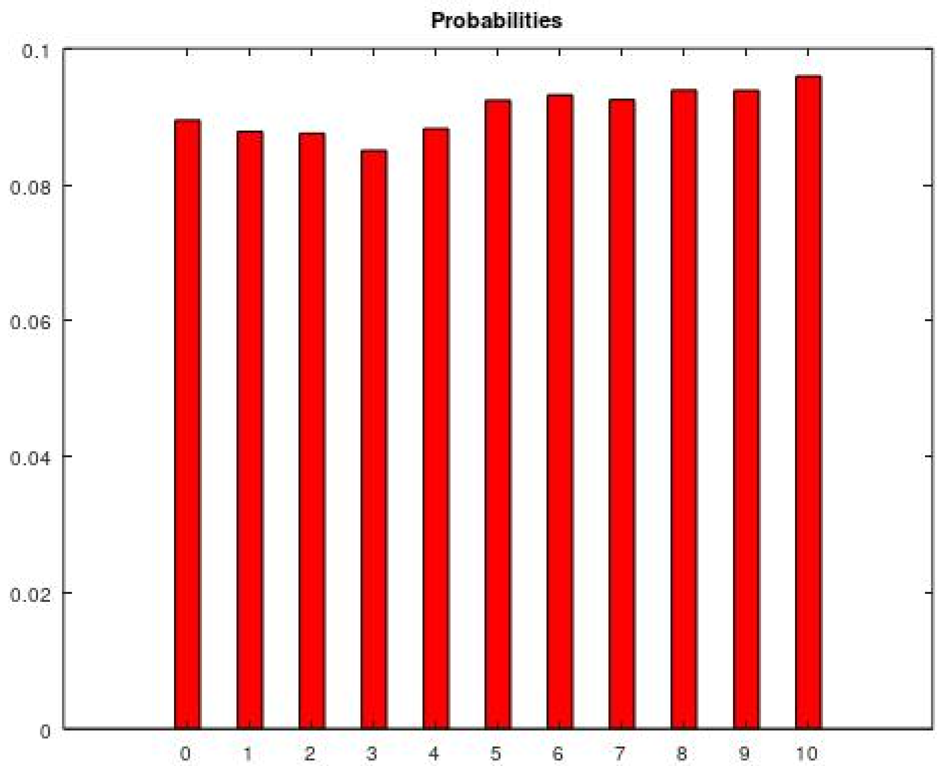




- Για λ=5:

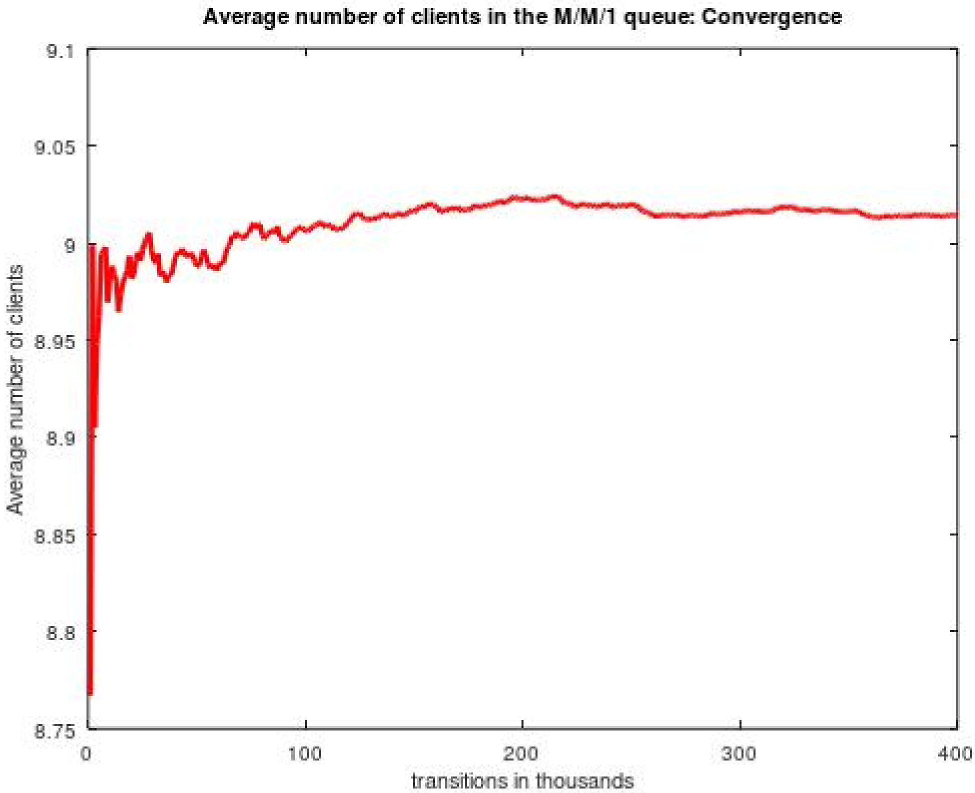


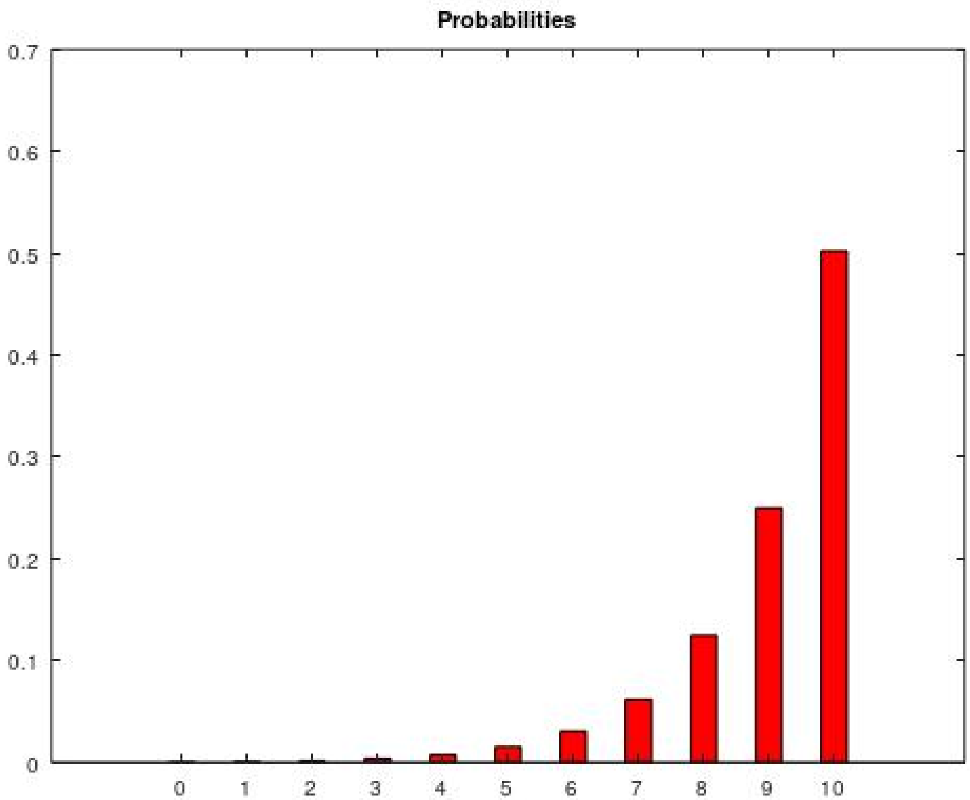
Average number of clients in the M/M/1 queue: Convergence
Average number of clients
transitions in thousands
5.2
5.1
5
4.9
4.8
4.7
4.6
0
20
40
60
80
100
120



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

- Για λ=10:


Average number of clients in the M/M/1 queue: Convergence
Average number of clients
9.1
9.05
9
8.95
8.9
8.85
8.8
8.75
0
100
200
300
400
transitions in thousands


Probabilities
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10

Κώδικας:

```
%ask2
% Note: Due to ergodicity, every state has a probability >0.

clc;
clear all;
close all;

%rand('seed', 1)

N=10;

lambda = [1, 5, 10];
m=5;

for k=1:1:3
rand('seed', 1)
P=[0 0 0 0 0 0 0 0 0 0 0];
arrivals=[0 0 0 0 0 0 0 0 0 0 0]; %arrivals(i)->how many arrivals i have totally
                                  %from state i, i=0- N

clear('to_plot');
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;
threshold(k) = lambda(k)./(lambda(k) + m); % 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:N+1
        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:N+1
       mean_clients = mean_clients + (i-1).*P(i);  %mean=Σk*P(k), k=0,..., N, but the enumeration of
                                                   %array starts at one, so P(0)->P(1),..
    endfor
```

```
    to_plot(k,index) = mean_clients;     %keep track of mean value of clients every 1000 transiti

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 1000000 % convergenc
      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(k) % arrival
    if current_state<11   %i don't count clients that come in a full system
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1)++;
      if current_state<10
       current_state = current_state + 1;
      endif
    endif
  else % departure
    if current_state != 0 % no departure from an empty system
      current_state = current_state - 1;
    endif
  endif
endwhile

display("Probabilities of each state for λ=");
disp(lambda(k));
for i=0:1:N
  display(P(i+1));
endfor


figure(2*k-1);
plot(to_plot(k,:),"r","linewidth",1.3);
title("Average number of clients in the M/M/1 queue: Convergence");
xlabel("transitions in thousands");
ylabel("Average number of clients");
hold off

figure(2*k);
x=[0 1 2 3 4 5 6 7 8 9 10];
bar(x, P,'r',0.4);
title("Probabilities")
endfor
```

3.

Από τα διαγράμματα παρατηρούμε πως όσο αυξάνεται το λ απαιτούνται περισσότερες μεταβάσεις για να ικανοποιηθεί το κριτήριο σύγκλισης. Αυτό συμβαίνει διότι, καθώς αυξάνουμε τον ρυθμό αφίξεων και διατηρούμε σταθερό τον ρυθμό εξυπηρέτησης, αυξάνεται η καθυστέρηση στο σύστημα και αργεί να επέλθει η εργοδική ισορροπία.

Σύμφωνα με τα παραπάνω διαγράμματα, μπορούμε να θεωρήσουμε μια αρχική μεταβατική περίοδο σε κάθε περίπτωση, ώστε να επιταχυνθεί η σύγκλισης της προσομοίωσης. Συγκεκριμένα για λ=1 πελάτες/sec μπορούμε να αγνοήσουμε 20000 αρχικές μεταβάσεις, για λ=5 80000 μεταβάσεις και για λ=10 100000 μεταβάσεις.

4.

Για μεταβλητό ρυθμό εξυπηρέτησης $\mu_i = \mu \cdot (i+1),\ \mu = 1\ \frac{\pi\varepsilon\lambda\alpha\tau\eta\varsigma}{sec}$, όπου i=current state, θα άλλαζε ανάλογα το threshold σε $\frac{\lambda}{\mu_i+\lambda}$. Στον κώδικα, δηλαδή, θα υπολογίζαμε σε κάθε αλλαγή κατάστασης νέο μ και νέο threshold.