ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ
3Η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

Ονοματεπώνυμο: Δημήτριος Βασιλείου
Α.Μ: el19830
Εξάμηνο: 6ο
Σχολή: ΗΜΜΥ

Προσομοίωση συστήματος M/M/1/10:

(1). Παραθέτουμε την έξοδο που παράγεται για debugging, δηλαδή για τις 30 πρώτες μεταβάσεις φαίνονται: η κατάσταση στην οποία βρίσκεται το σύστημα, τι είδους μετάβαση είναι η επόμενη και ο συνολικός αριθμός αφίξεων στην παρούσα κατάσταση:

```
Transition 1:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 1

Transition 2:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 0

Transition 3:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 2

Transition 4:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 0

Transition 5:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 3

Transition 6:
Current system state : 1
Next transition : arrival
Total number of arrivals in current state : 1

Transition 7:
Current system state : 2
Next transition : arrival
Total number of arrivals in current state : 1

Transition 8:
Current system state : 3
Next transition : arrival
Total number of arrivals in current state : 1
```

```
Transition 9:
Current system state : 4
Next transition : departure
Total number of arrivals in current state : 0

Transition 10:
Current system state : 3
Next transition : departure
Total number of arrivals in current state : 1

Transition 11:
Current system state : 2
Next transition : departure
Total number of arrivals in current state : 1

Transition 12:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1

Transition 13:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 4

Transition 14:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1

Transition 15:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 5

Transition 16:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1
```

```
Transition 17:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 6

Transition 18:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1

Transition 19:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 7

Transition 20:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1

Transition 21:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 8

Transition 22:
Current system state : 1
Next transition : departure
Total number of arrivals in current state : 1

Transition 23:
Current system state : 0
Next transition : arrival
Total number of arrivals in current state : 9

Transition 24:
Current system state : 1
Next transition : arrival
Total number of arrivals in current state : 2
```

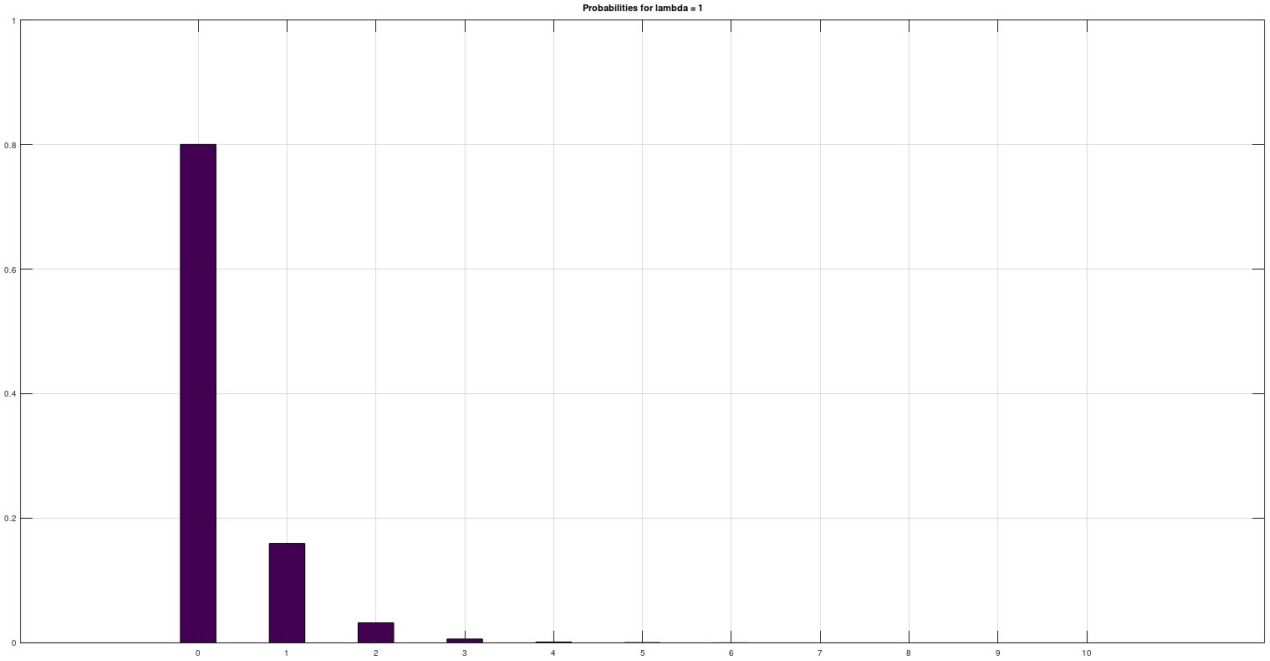
```
Transition 25:
Current system state : 2
Next transition : arrival
Total number of arrivals in current state : 2

Transition 26:
Current system state : 3
Next transition : departure
Total number of arrivals in current state : 1

Transition 27:
Current system state : 2
Next transition : arrival
Total number of arrivals in current state : 3

Transition 28:
Current system state : 3
Next transition : departure
Total number of arrivals in current state : 1

Transition 29:
Current system state : 2
Next transition : departure
Total number of arrivals in current state : 3

Transition 30:
Current system state : 1
Next transition : arrival
Total number of arrivals in current state : 3
```

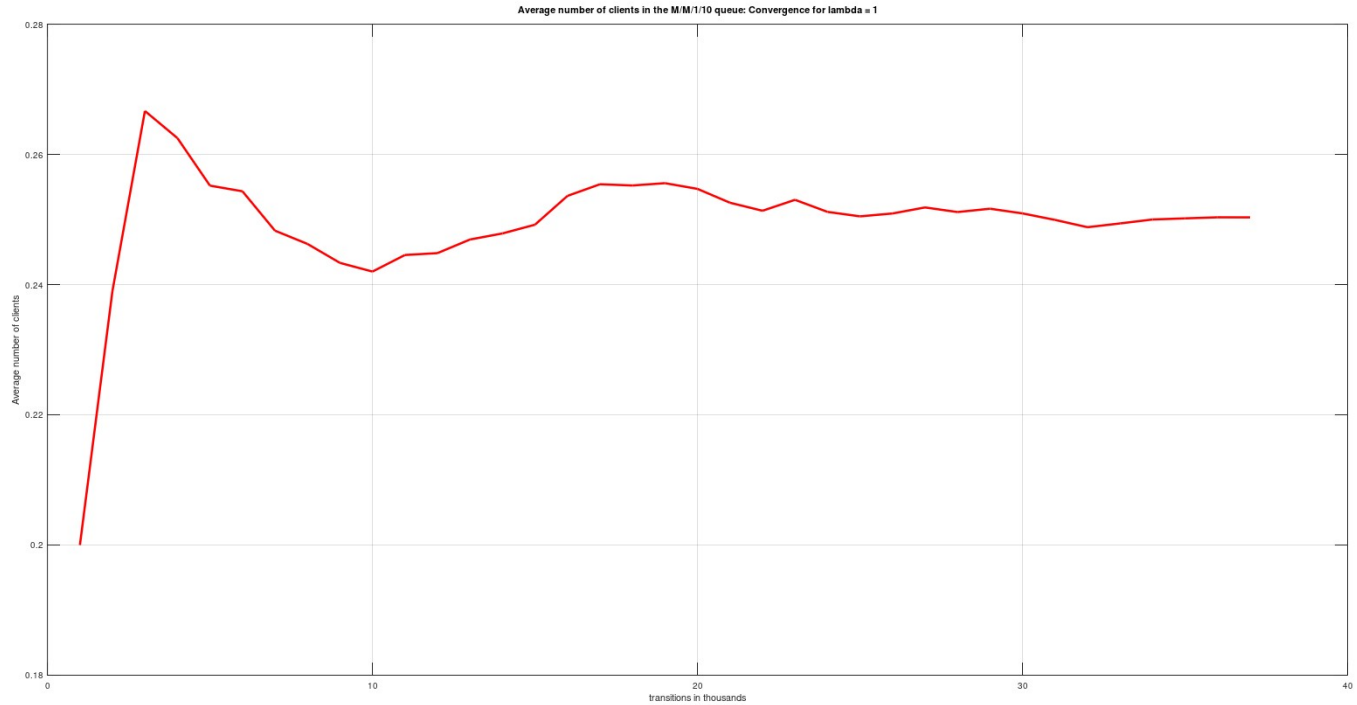
Ο κώδικας που χρησιμοποιήθηκε είναι ο ακόλουθος:

```
#1
clc;
clear all;
close all;

P = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
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;

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

while transitions >= 0 && transitions < 30
  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
     total_arrivals = total_arrivals + 1;
    if current_state < 11
      printf("Transition %d:\n", transitions);
      printf("Current system state : %d\n", current_state);
      printf("Next transition : arrival\n");
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
      printf("Total number of arrivals in current state : %d\n\n", arrivals(current_state + 1));
    endif
    if current_state < 10 %increase current state only if less than 10
      current_state = current_state + 1;
    endif

  else % departure
    if current_state != 0
      printf("Transition %d:\n", transitions);
      printf("Current system state : %d\n", current_state);
      printf("Next transition : departure\n");
      printf("Total number of arrivals in current state : %d\n\n", arrivals(current_state + 1));
    endif
    current_state = current_state - 1;
  endif
endwhile
```

(2). Εκτελούμε την προσομοίωση για λ = {1, 5, 10}. Παραθέτουμε γραφικά τις εργοδικές πιθανότητες που υπολογίζει τελικά η προσομοίωση και την εξέλιξη του μέσου αριθμού πελατών στο σύστημα.

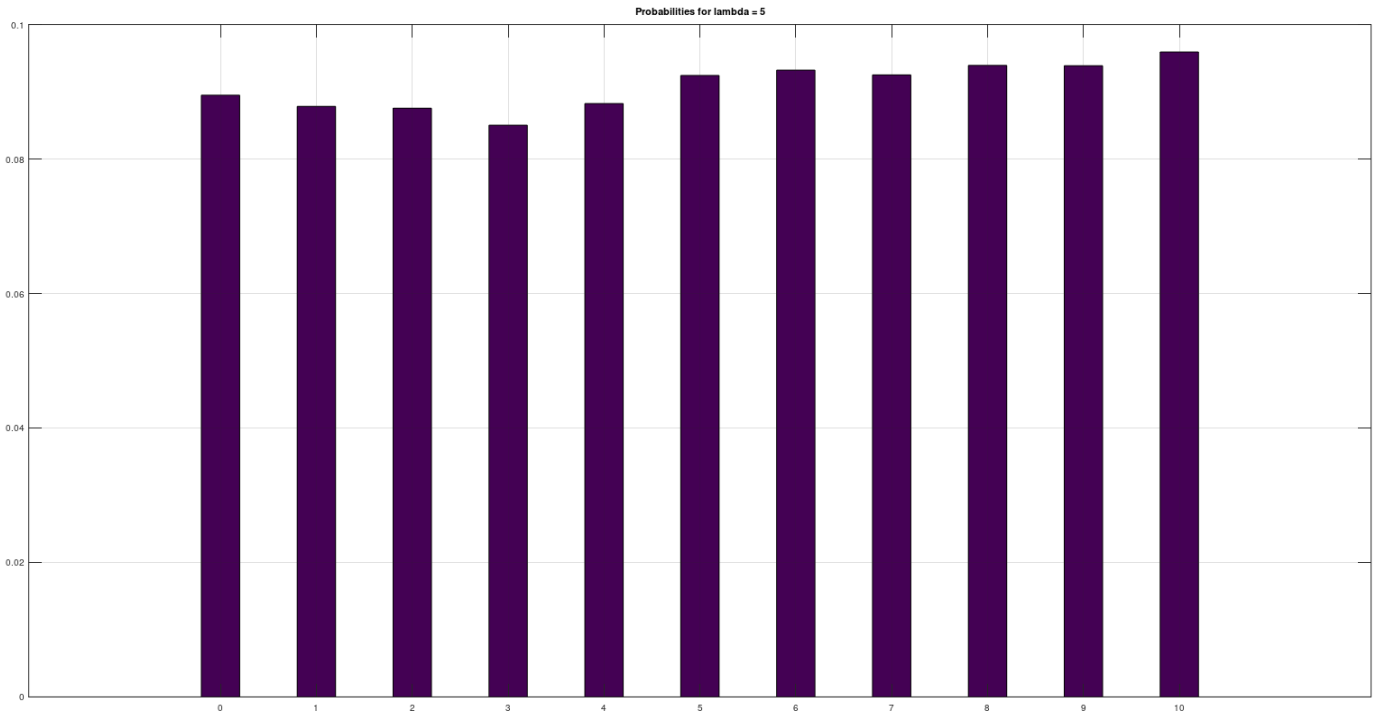
→ Για λ = 1:

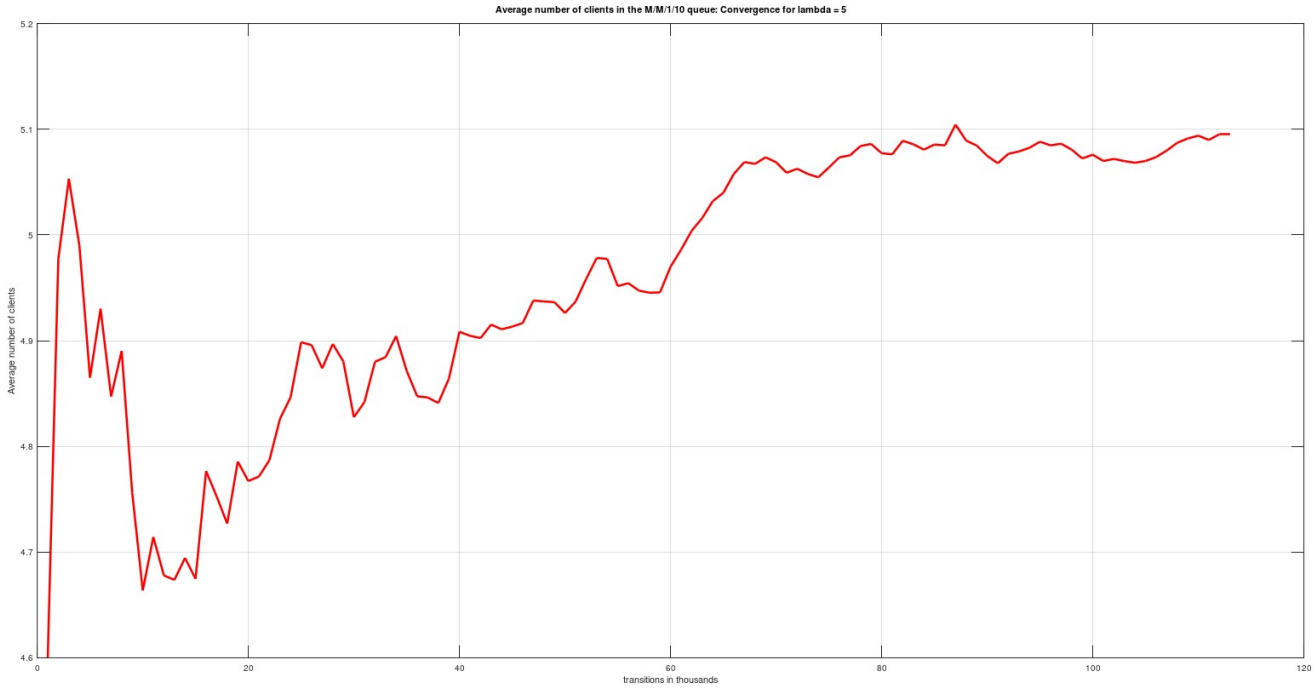




```
Probabilities of each state:
P(0) = 0.800216
P(1) = 0.159568
P(2) = 0.0322162
P(3) = 0.00621622
P(4) = 0.0012973
P(5) = 0.000432432
P(6) = 5.40541e-05
P(7) = 0
P(8) = 0
P(9) = 0
P(10) = 0
P(blocking) = 0
Average delay time : 0.250324
```

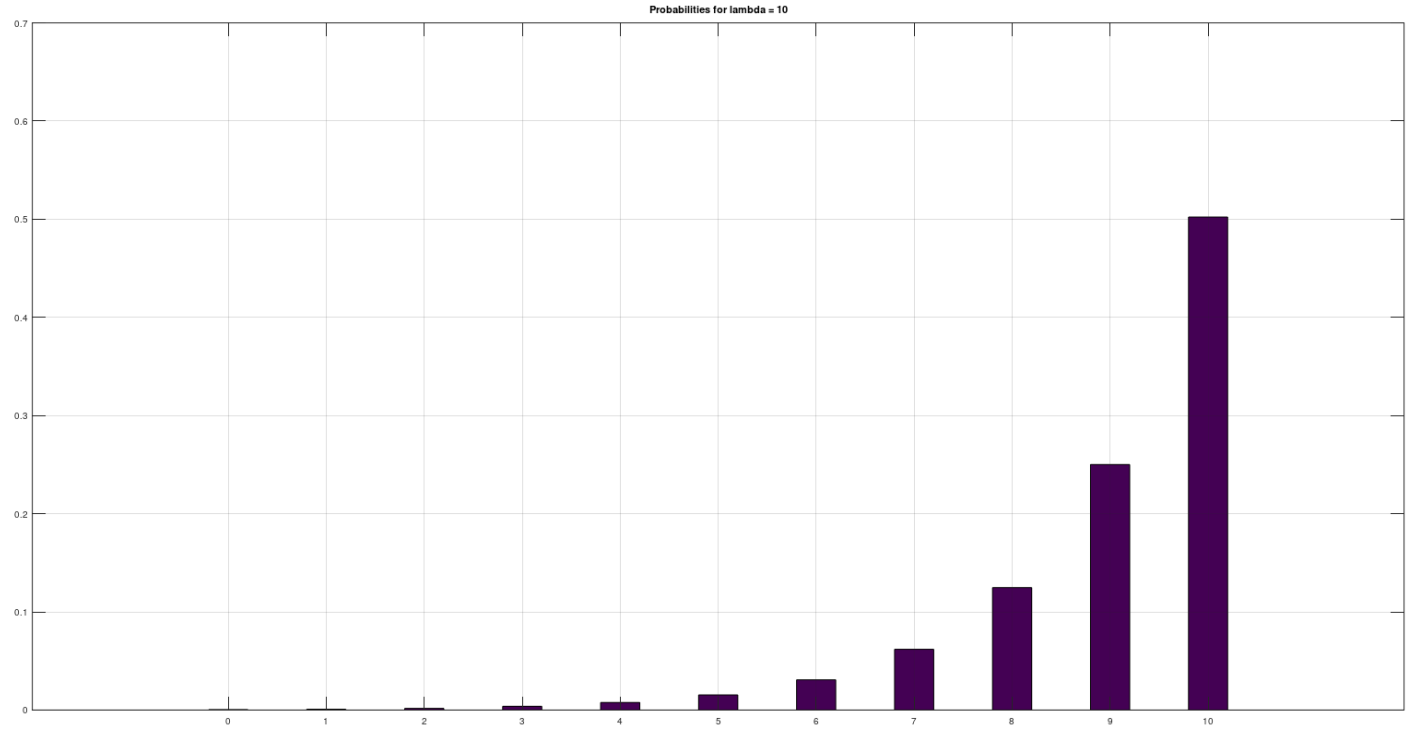
→ **Για λ = 5:**

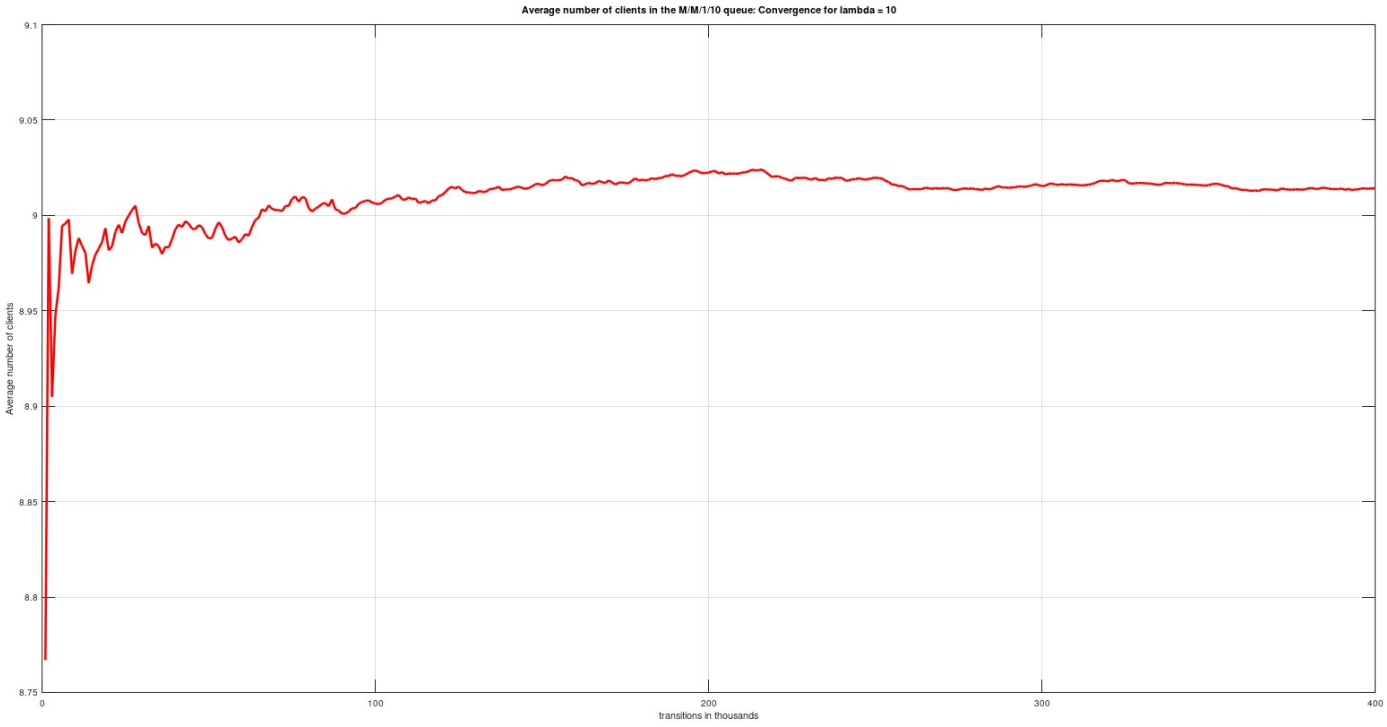





```
Probabilities of each state:
P(0) = 0.0894921
P(1) = 0.0878239
P(2) = 0.0875543
P(3) = 0.0850268
P(4) = 0.0882452
P(5) = 0.0924241
P(6) = 0.0932161
P(7) = 0.0925083
P(8) = 0.0939238
P(9) = 0.0938732
P(10) = 0.0959121
P(blocking) = 0.0959121
Average delay time : 1.12718
```

→ Για λ = 10:


Probabilities for lambda = 10
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


Average number of clients in the M/M/1/10 queue: Convergence for lambda = 10
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 of each state:
P(0)  = 0.000479279
P(1)  = 0.000853715
P(2)  = 0.00174487
P(3)  = 0.0038567
P(4)  = 0.00772837
P(5)  = 0.0154455
P(6)  = 0.0308311
P(7)  = 0.0619393
P(8)  = 0.124785
P(9)  = 0.250112
P(10)  = 0.502224
P(blocking)  = 0.502224
Average delay time  :  1.81088
```

Παραθέτουμε ενδεικτικά τον κώδικα που χρησιμοποιήθηκε για λ = 10, διότι για λ = 1 και λ = 5 αρκεί απλώς να αλλάξει η τιμή της μεταβλητής lambda καθώς και τα αντίστοιχα μηνύματα στους τίτλους.

```
#2, λ = 10
clc;
clear all;
close all;
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];
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 = 10;
mu = 5;
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
    total_arrivals = total_arrivals + 1;
    arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
    if current_state < 10 %increase current state only if less than 10
      current_state = current_state + 1;
    endif

  else % departure
    if current_state != 0
      current_state = current_state - 1;
    endif
  endif
endwhile

display("Probabilities of each state:");
for i=1:1:length(arrivals)
  printf("P(%d) = %d\n", i-1, P(i));
endfor

printf("P(blocking) = %d\n", P(11));

throughput = lambda*(1-P(11));
average_delay_time = mean_clients / throughput; %Little
printf("Average delay time : %d\n", average_delay_time);
```

```
grid on;

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

state = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
figure(2);
bar(state,P ,0.4);
title("Probabilities for lambda = 10")
grid on;
```

(3). Από τα διαγράμματα των μέσων αριθμών πελατών στο σύστημα παρατηρούμε ότι όσο το λ αυξάνεται, χρειάζονται περισσότερες μεταβάσεις ώστε ο μέσος αριθμός πελατών να σταθεροποιηθεί και να τελειώσουν τα μεταβατικά φαινόμενα.
Παρατηρώντας τα διαγράμματα, βλέπουμε πως για να επιταχυνθεί η σύγκλιση της προσομοίωσης μπορούμε προσεγγιστικά να αγνοήσουμε:
→ Για λ = 1, 30000 μεταβάσεις
→ Για λ = 5, 100000 μεταβάσεις
→ Για λ = 10, 300000 μεταβάσεις

(4). Εφόσον πλέον το μ εξαρτάται από την κατάσταση όπου βρίσκεται το σύστημα τότε θα πρέπει με κάθε αλλαγή της μεταβλητής current_state να ενημερώνεται και το μ ως
`mu = 1 * (current_state + 1);`
Επίσης αλλάζει και η τιμή της μεταβλητής threshold οπότε μετά από κάθε αλλαγή το mu στον κώδικα, θα έχουμε στη συνέχεια την εντολή
`threshold = lambda / (lambda + mu);`
Παραθέτουμε τα σημεία όπου ο κώδικας έχει αλλάξει:

```
lambda = 1;
mu = 1 * (current_state + 1); %initialize mu with current_state 0
threshold = lambda/(lambda + mu); % the threshold used to calculate probabilities
```

Πριν το loop αρχικοποιούμε το μ έχοντας ως αρχική την κατάσταση 0 και υπολογίζουμε το threshold.

```
  if current_state == 0 || random_number < threshold % arrival
    total_arrivals = total_arrivals + 1;
    arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
    if current_state < 10 %increase current state only if less than 10
      current_state = current_state + 1;
    endif

  else % departure
    if current_state != 0
      current_state = current_state - 1;
    endif
  endif
  mu = 1 * (current_state + 1); %new value of m
  threshold = lambda/(lambda + mu); %new value of threshold
endwhile
```

Μετά από κάθε αλλαγή του current_state (arrival η departure) ενημερώνεται το μ, καθώς και το threshold.