

Προσομοίωση συστήματος M/M/1/10:

1) Ακολουθούν οι 3 πίνακες για $\lambda = 1, \lambda = 5, \lambda = 10$:

The debug matrix for tracing the first 30 transitions (transition number, current state, next transition equal to arrival (1) or departure (-1) and total number of arrivals on current state) for Lambda = 1 is:

1	0	1	1
2	1	-1	0
3	0	1	2
4	1	1	1
5	2	-1	0
6	1	-1	1
7	0	1	3
8	1	1	2
9	2	-1	0
10	1	1	3
11	2	-1	0
12	1	-1	3
13	0	1	4
14	1	-1	3
15	0	1	5
16	1	-1	3
17	0	1	6
18	1	-1	3
19	0	1	7
20	1	-1	3
21	0	1	8
22	1	1	4
23	2	-1	0
24	1	-1	4
25	0	1	9
26	1	-1	4
27	0	1	10
28	1	-1	4
29	0	1	11
30	1	1	5

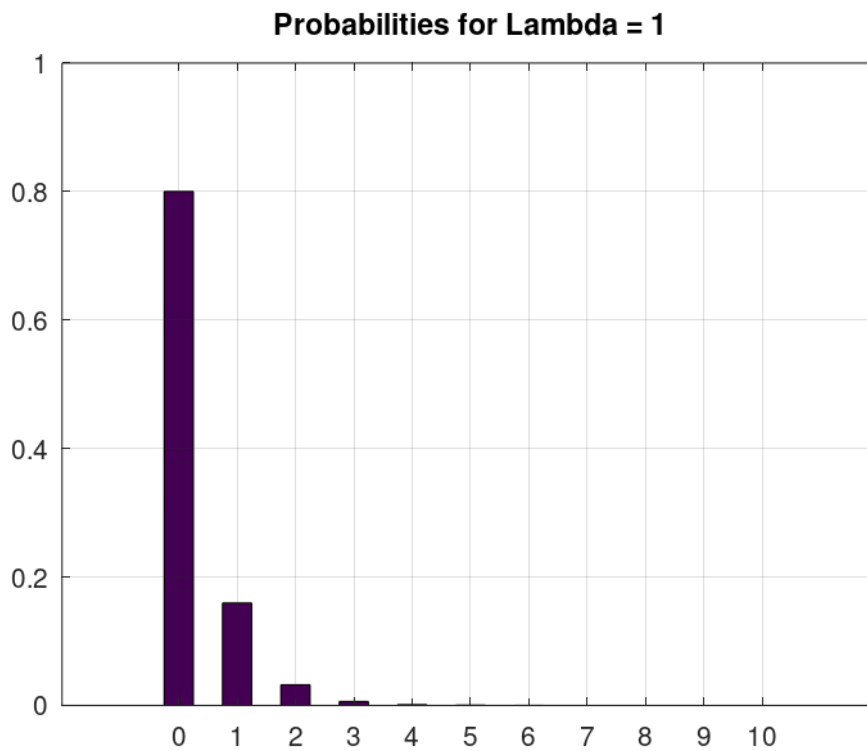
The debug matrix for tracing the first 30 transitions
(transition number, current state, next transition
equal to arrival (1) or departure (-1) and total number
of arrivals on current state) for $\Lambda = 5$ is:

1	0	1	1
2	1	1	1
3	2	1	1
4	3	1	1
5	4	1	1
6	5	-1	0
7	4	1	2
8	5	1	1
9	6	1	1
10	7	1	1
11	8	1	1
12	9	-1	0
13	8	-1	1
14	7	-1	1
15	6	-1	1
16	5	-1	1
17	4	-1	2
18	3	1	2
19	4	-1	2
20	3	1	3
21	4	-1	2
22	3	1	4
23	4	-1	2
24	3	-1	4
25	2	1	2
26	3	-1	4
27	2	1	3
28	3	1	5
29	4	1	3
30	5	1	2

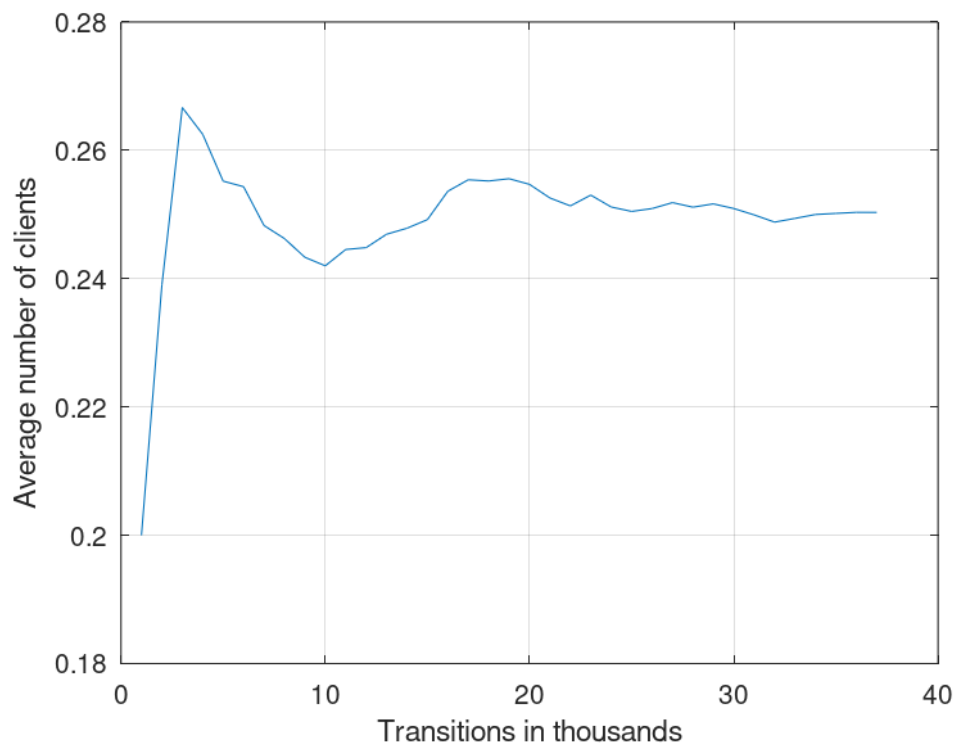
The debug matrix for tracing the first 30 transitions
(transition number, current state, next transition
equal to arrival (1) or departure (-1) and total number
of arrivals on current state) for $\Lambda = 10$ is:

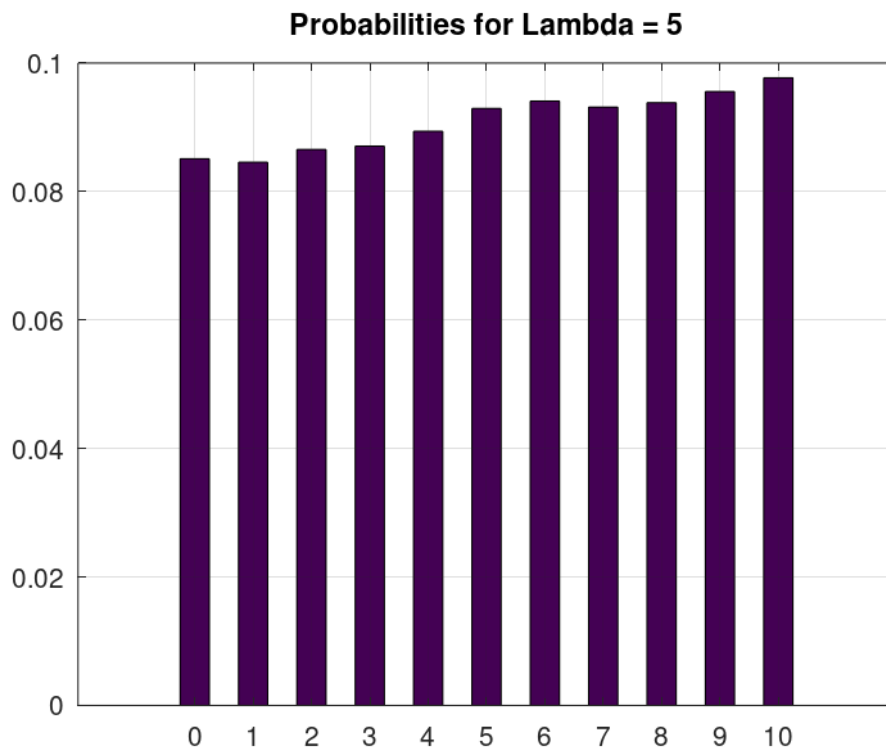
1	0	1	1
2	1	1	1
3	2	1	1
4	3	1	1
5	4	1	1
6	5	1	1
7	6	1	1
8	7	-1	0
9	6	-1	1
10	5	1	2
11	6	-1	1
12	5	1	3
13	6	1	2
14	7	1	1
15	8	1	1
16	9	1	1
17	10	1	1
18	10	1	2
19	10	-1	2
20	9	1	2
21	10	1	3
22	10	1	4
23	10	1	5
24	10	1	6
25	10	1	7
26	10	1	8
27	10	1	9
28	10	1	10
29	10	1	11
30	10	-1	11

2) Ακολουθούν τα γραφήματα για τις εργοδικές πιθανότητες και για την εξέλιξη του μέσου αριθμού πελατών στο σύστημα μέχρι την σύγκλιση για $\lambda = 1, \lambda = 5, \lambda = 10$.

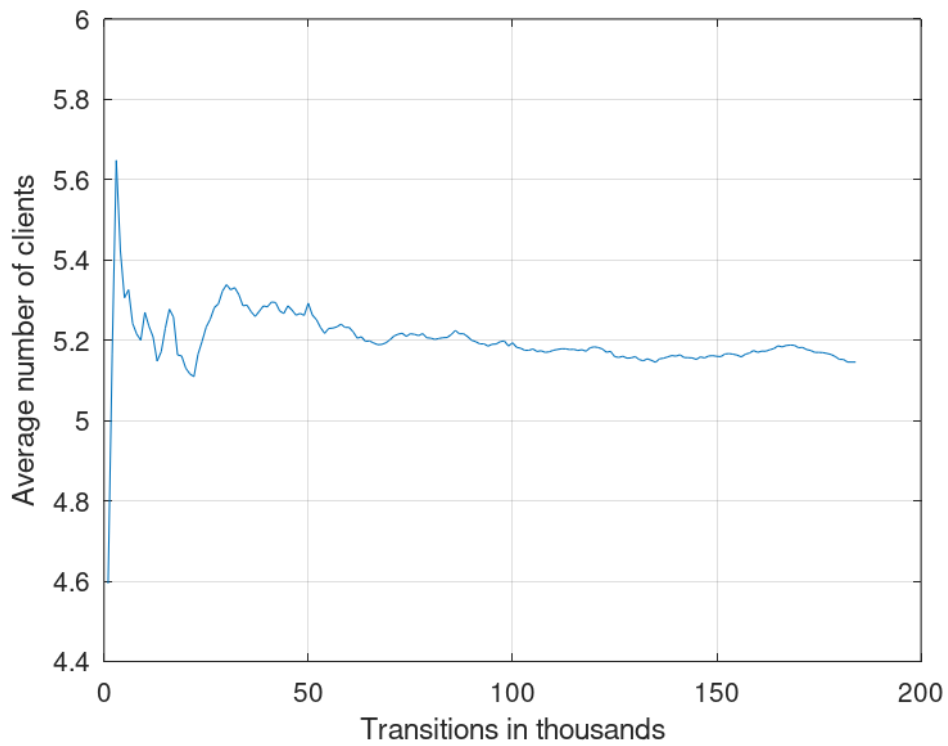


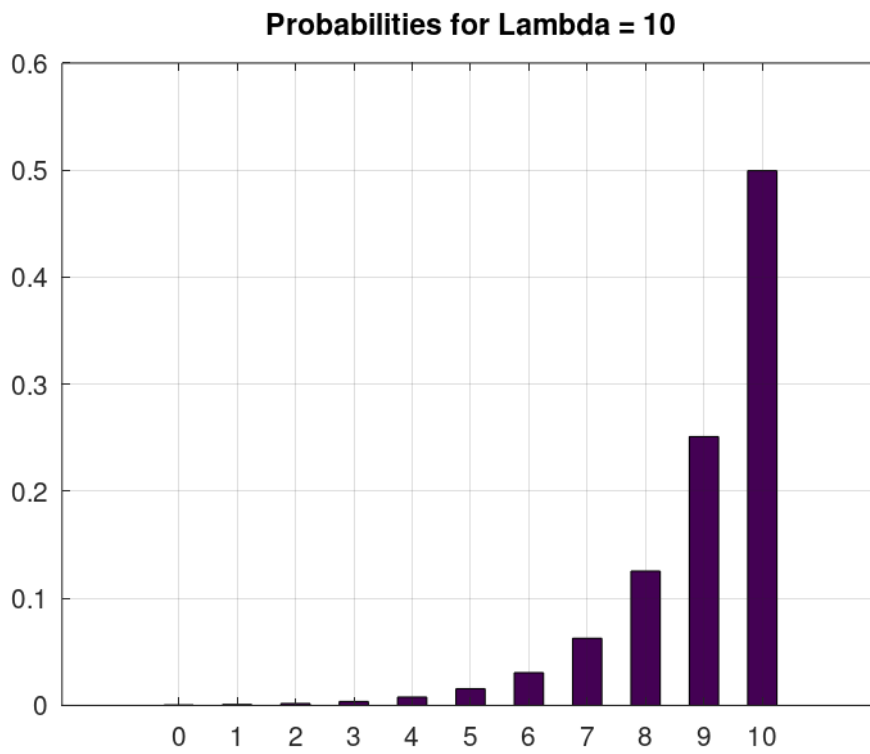
Average number of clients in a M/M/1/10 queue: Convergence for Lambda =



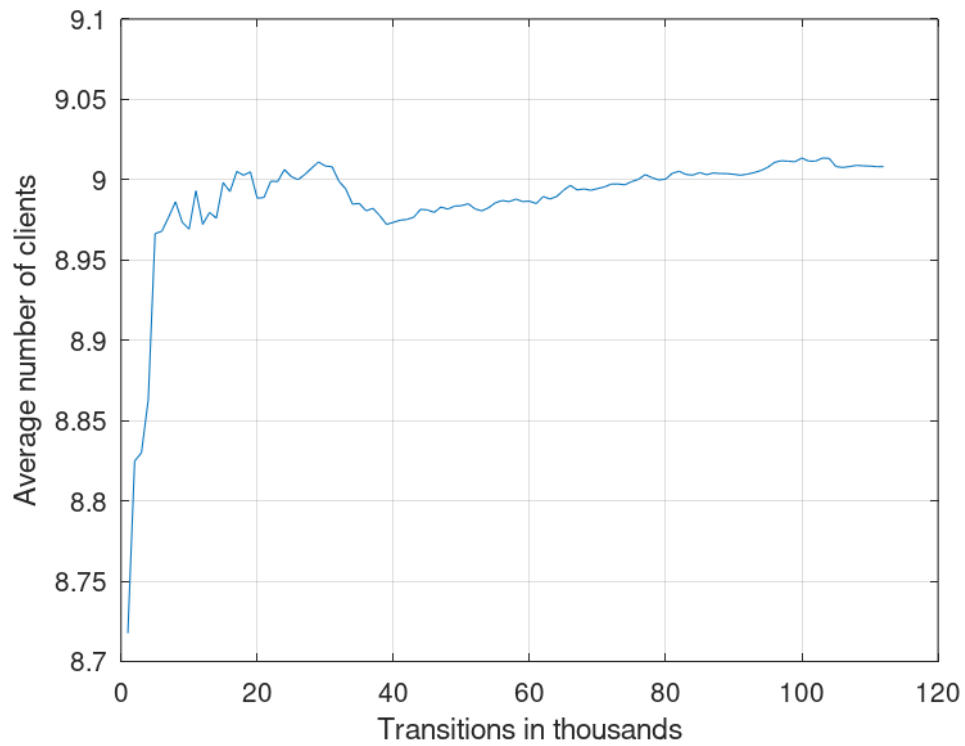


Average number of clients in a M/M/1/10 queue: Convergence for Lambda =





Average number of clients in a M/M/1/10 queue: Convergence for Lambda =



3) Παρατηρούμε ότι όσο αυξάνεται η διαφορά $|\lambda - \mu|$, τόσο πιο γρήγορα συγκλίνει η προσομοίωση μας. Αυτό εξηγείται από το γεγονός ότι όσο πιο κοντά βρίσκεται ο ρυθμός αφίξεων λ με το ρυθμό εξυπηρέτησης μ , τόσο μικρότερη μεταβολή θα παρατηρείται στον μέσο αριθμό πελατών στο σύστημα, καθώς για κάθε πλήθος που εισάγεται θα εξυπηρετούνται περίπου άλλοι τόσοι. Συγκεκριμένα, για $|\lambda - \mu| = 4$ χρειαζόμαστε περίπου 37.000 μεταβάσεις, για $|\lambda - \mu| = 0$ χρειαζόμαστε περίπου 180.000 μεταβάσεις, ενώ για $|\lambda - \mu| = 5$ χρειαζόμαστε περίπου 110.000 μεταβάσεις. Οι αρχικές μεταβάσεις που θα μπορούσαμε να αγνοήσουμε με ασφάλεια εξαρτώνται φυσικά και από την τιμή του λ και κατά επέκταση και στην διαφορά $|\lambda - \mu|$. Όσο μεγαλύτερη αυτή η διαφορά, τόσο λιγότερες αρχικές μεταβάσεις μπορούμε να αγνοήσουμε. Στην περίπτωση μας, με $\lambda = 1, \lambda = 5, \lambda = 10$, περίπου 10.000, 50.000, 40.000 καταστάσεις αντίστοιχα, μπορούν να αγνοηθούν με ασφάλεια.

4) Σε περίπτωση που είχαμε μεταβλητό ρυθμό εξυπηρέτησης, όπως πχ $\mu_i = \mu(i + 1)$, $\mu = 1$ πελάτης/sec και $i = \{1, 2, \dots, 10\}$ η κατάσταση του συστήματος, θα έπρεπε να αλλάζουμε το threshold στον κώδικα μας. Αφού $\mu = 1$, τότε $\mu_i = i + 1$ και επομένως το threshold θα έχει ως εξής: $threshold_i = \frac{\lambda_i}{\lambda_i + current_state + 1}$. Επίσης, η μεταβολή αυτή θα πρέπει να γίνεται μέσα στο while loop, ώστε να ανανεώνεται με κάθε μετάβαση.

Τέλος, παρακάτω ακολουθεί ο κώδικας που χρησιμοποιήθηκε για την παραγωγή των παραπάνω αποτελεσμάτων με τον κώδικα για το debugging του πρώτου ζητούμενου να εμφανίζεται με σχόλια χωρίς κενό, ώστε να ξεχωρίζουν από τα σχόλια επεξήγησης.

% 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;

rand('seed',1);
counter = 1;
lambdas = [1,5,10];

for i = lambdas

    clear to_plot;

    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
%debug_matrix = [];

while transitions >= 0 %&& transitions < 30
    transitions++; % one more transitions step
    %debug_matrix(debug_i,1) = debug_i;
    %debug_matrix(debug_i,2) = current_state;

    if mod(transitions,1000) == 0 % check for convergence every 1000 transitions steps
        index++;
        for j=1:length(arrivals)
            P(j) = arrivals(j)/total_arrivals; % calculate the probability of every state in the system
        endfor

        mean_clients = 0; % calculate the mean number of clients in the system
        for j=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
        total_arrivals++;
        arrivals(current_state + 1)++; % increase the number of arrivals in the current state
    end
end

```



```

%debug_matrix(transitions,3) = 1;
%debug_matrix(transitions,4) = arrivals(current_state + 1);

if(current_state != 10)
    current_state++;
endif

else % departure
    %debug_matrix(transitions,3) = -1;
    %debug_matrix(transitions,4) = arrivals(current_state + 1);

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

%printf(sprintf("The debug matrix for tracing the first 30 transitions\n(transition number, current state, next transition\nequal to arrival (1) or departure (-1) and total number\nof arrivals on current state) for Lambda = %d is :\n", i));
%disp(debug_matrix);

printf(sprintf("The Blocking Probability is equal to: %d\n", P(length(arrivals))));

figure(counter++);
bar(0:1:(length(arrivals)-1),P,0.5);
grid on;
title(sprintf("Probabilities for Lambda = %d", i));

figure(counter++);
plot(to_plot);
grid on;
title(sprintf("Average number of clients in a M/M/1/10 queue: Convergence for Lambda = %d", i));
xlabel("Transitions in thousands");
ylabel("Average number of clients");
endfor

```