ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

3η Ομάδα Ασκήσεων

Νικόλαος Δημητριάδης
03114016
ΗΜΜΥ 8ο

Εισαγωγή

Τα διάφορα μέρη παραδίδονται σε διαφορετικά σε διαφορετικά αρχεία .m και τα επιμέρους ερωτήματά τους χωρίζονται σε sections στον κώδικα για καλύτερη κατανόηση/διόρθωση. Το μέρος 1 βασίστηκε στο demo1.m που αναρτήθηκε στο site του μαθήματος. Χρησιμοποιήθηκε `MATLAB`.

1 Σύγκριση συστημάτων M/M/1 και M/D/1

Ο μέσος αριθμός πελατών σε ένα σύστημα M/D/1 είναι ίσος με:

$$\mathbb{E}[n(t)] = \rho + \frac{1}{2}\frac{\rho^2}{1-\rho} \tag{1}$$

ερώτημα 1

Λόγω άπειρης χωρητικότητας έχουμε $P_{blocking} = 0$. ¨αρα για $\rho < 1$ παρατηρείται εργοδικότητα και $\gamma = \lambda$. Τότε από τύπο του Little έχουμε:

$$Little \implies \mathbb{E}[T] = \frac{\mathbb{E}[n(t)]}{\gamma} = \frac{\mathbb{E}[n(t)]}{\lambda} \overset{(1)}{=} \frac{1}{\mu} + \frac{1}{2}\frac{\frac{\lambda}{\mu^2}}{1-\rho} \tag{2}$$

Ο μέσος χρόνος αναμονής, ο μέσος χρόνος καθυστέρησης και ο μέσος χρόνος εξυπηρέτησης συνδέονται από τη σχέση:

$$\begin{aligned} \mathbb{E}[T] &= \mathbb{E}[W] + \mathbb{E}[S] \\ \implies \mathbb{E}[W] &= \mathbb{E}[T] - \mathbb{E}[S] \\ \implies \mathbb{E}[W] &= \mathbb{E}[T] - \frac{1}{\mu} \\ \implies \mathbb{E}[W] &= \frac{1}{2}\frac{\frac{\lambda}{\mu^2}}{1-\rho} \end{aligned}$$

Τέλος, η συνθήκη για εργοδικότητα είναι $\lambda < \mu$.

ερώτημα 2

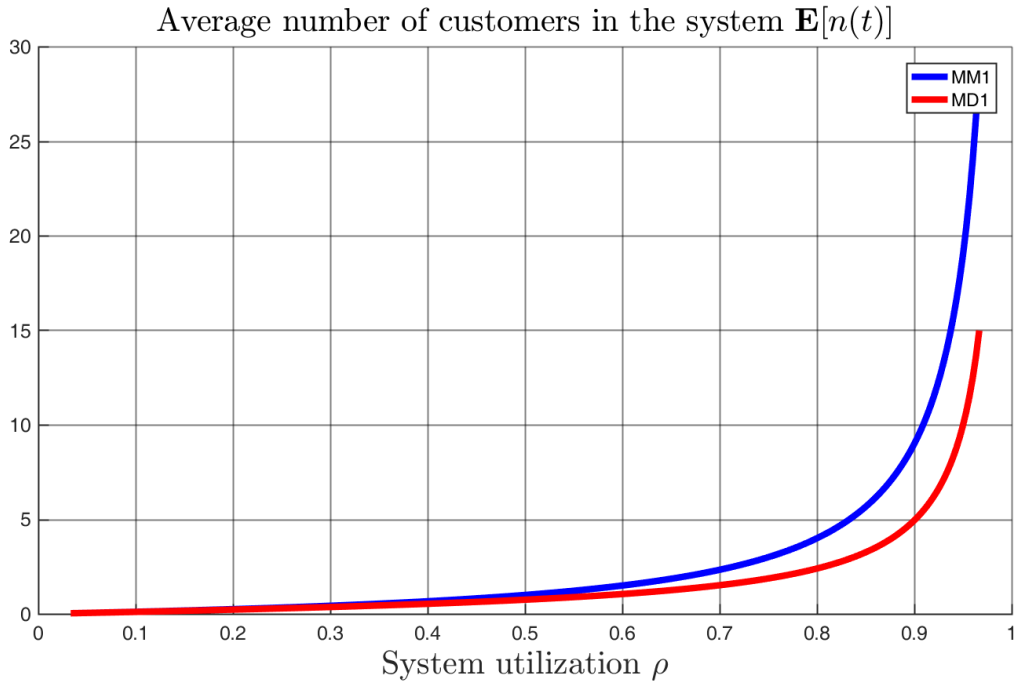
Η συνάρτηση `qsmd1` βασίζεται στην παραπάνω ανάλυση και παρουσιάζεται παρακάτω:

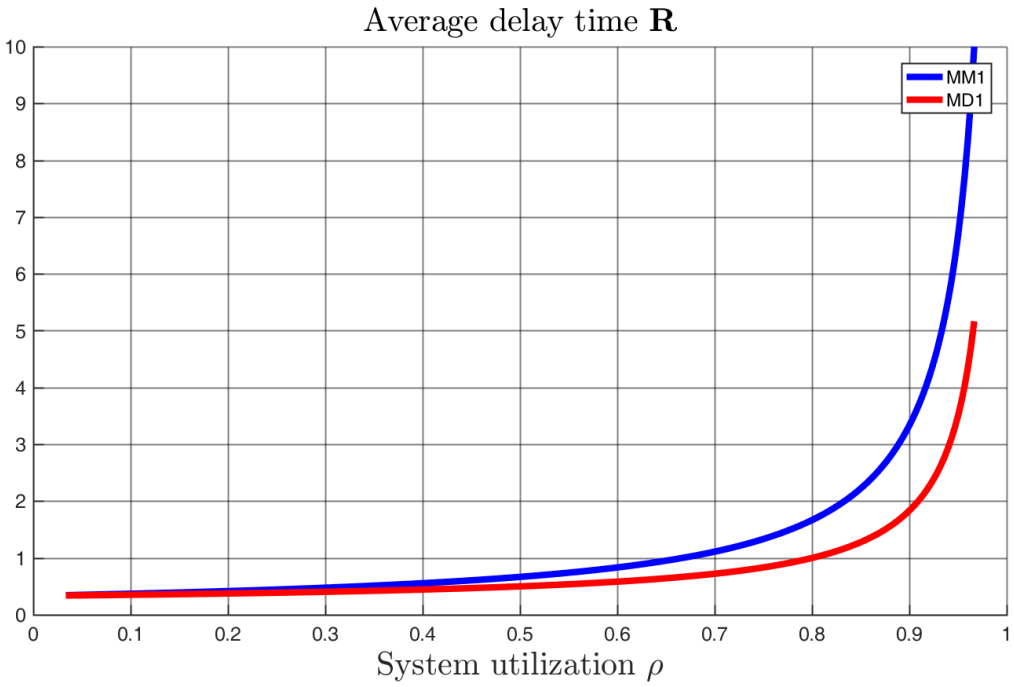
Listing 1: συνάρτηση `qsmd1`

```
function [U, R, Q, X, p0] = qsmd1( lambda, mu )
  lambda = lambda(:)';
  mu = mu(:)';
  U = lambda ./ mu; % utilization
  rho = U;
  p0 = 1-rho;
  Q = rho + (0.5*rho.^2) ./ (1-rho);
  R = Q ./ lambda;
  X = lambda;
end
```

ερώτημα 3

Από τη σύγκριση των δύο ουρών παρήχθησαν οι ακόλουθες δύο γραφικές παραστάσεις:





Από τα παραπάνω δύο γραφήματα φαίνεται ότι το σύστημα M/D/1 παράγει καλύτερα αποτελέσματα, καθώς χαρακτηρίζεται από μικρότερο χρόνο καθυστέρησης.

2 Προσομοίωση συστήματος M/M/1/10

ερώτημα 1

Για το `debugging` βλ. αντίστοιχο κώδικα στο παράρτημα(lines 71-79, 88).

ερώτημα 2

$\lambda = 1$

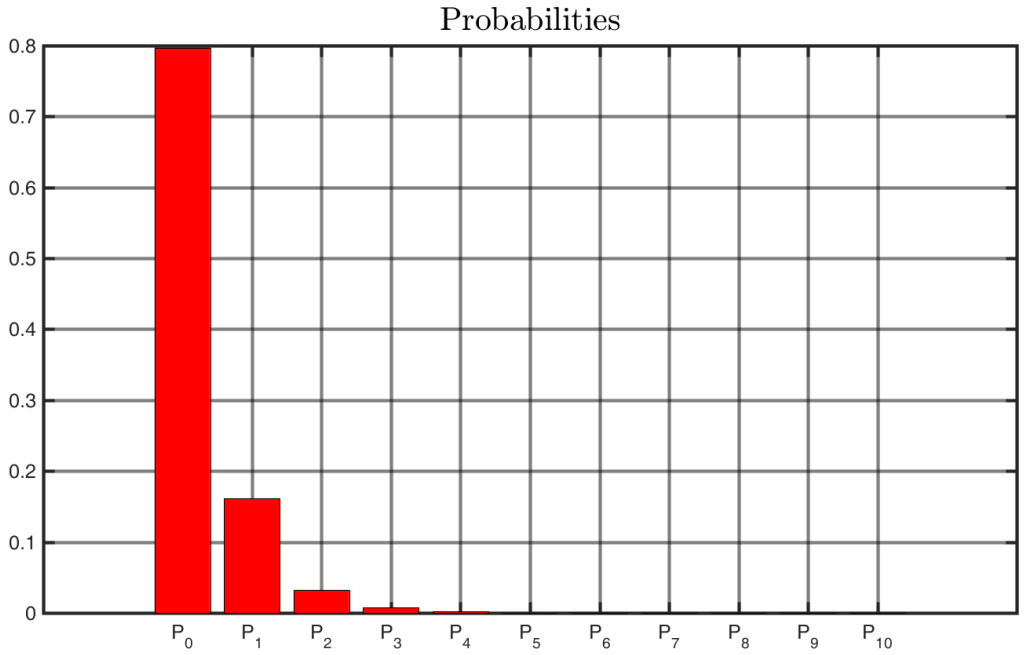


Figure 1: εργοδικές πιθανότητες για $\lambda = 1$

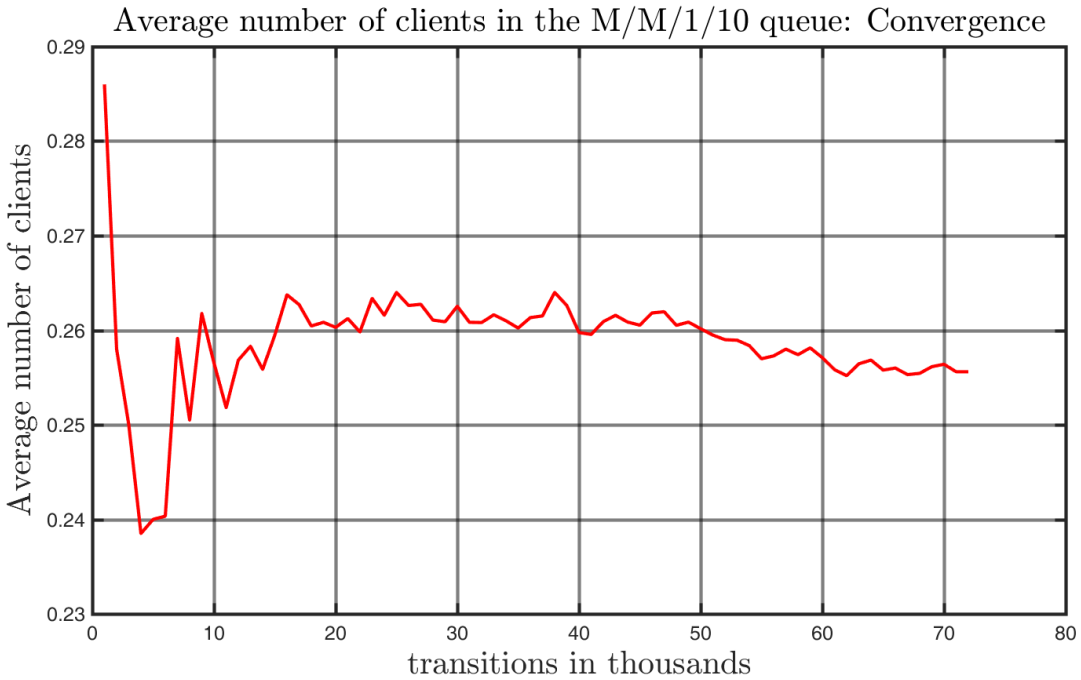


Figure 2: εξέλιξη του μέσου αριθμού πελατών στο σύστημα για $\lambda = 1$

λ = 5

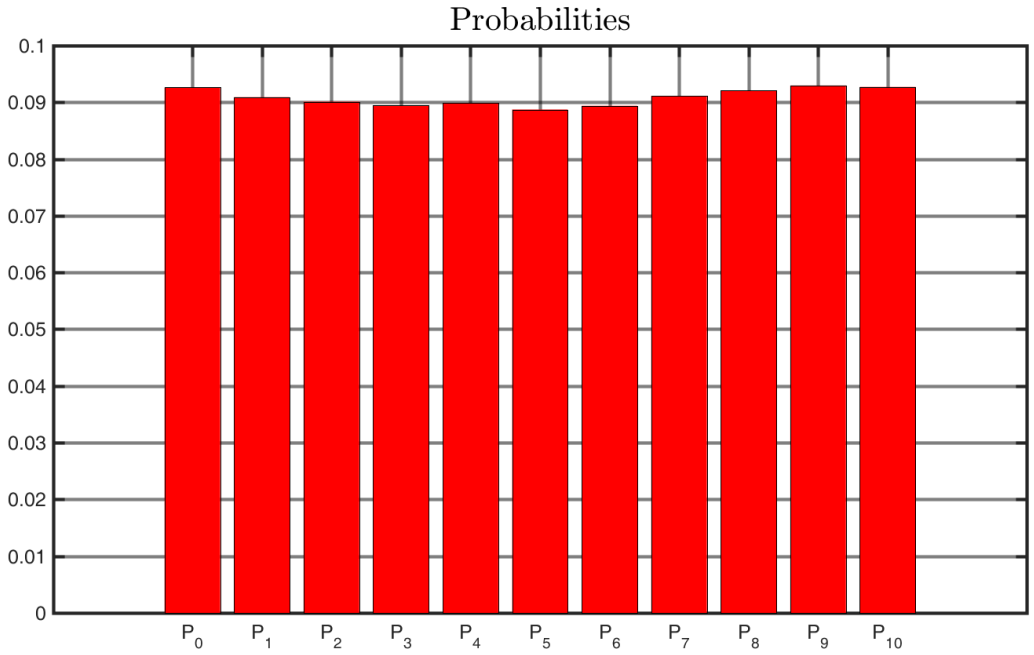


Figure 3: εργοδικές πιθανότητες για $\lambda = 5$

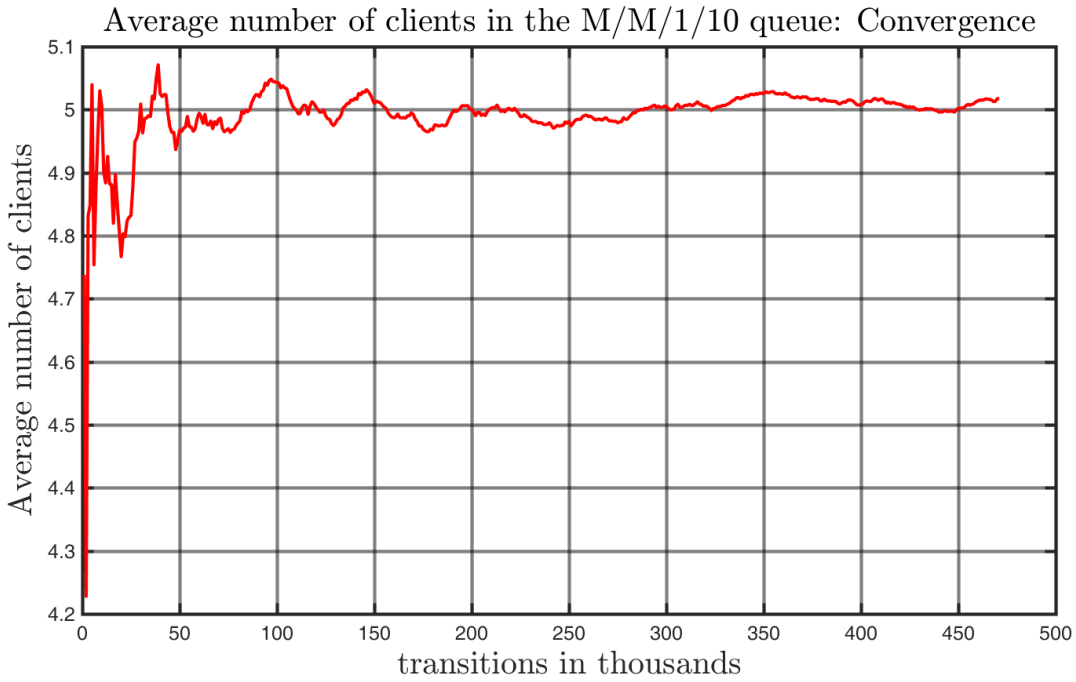


Figure 4: εξέλιξη του μέσου αριθμού πελατών στο σύστημα για $\lambda = 5$

$\lambda = 10$

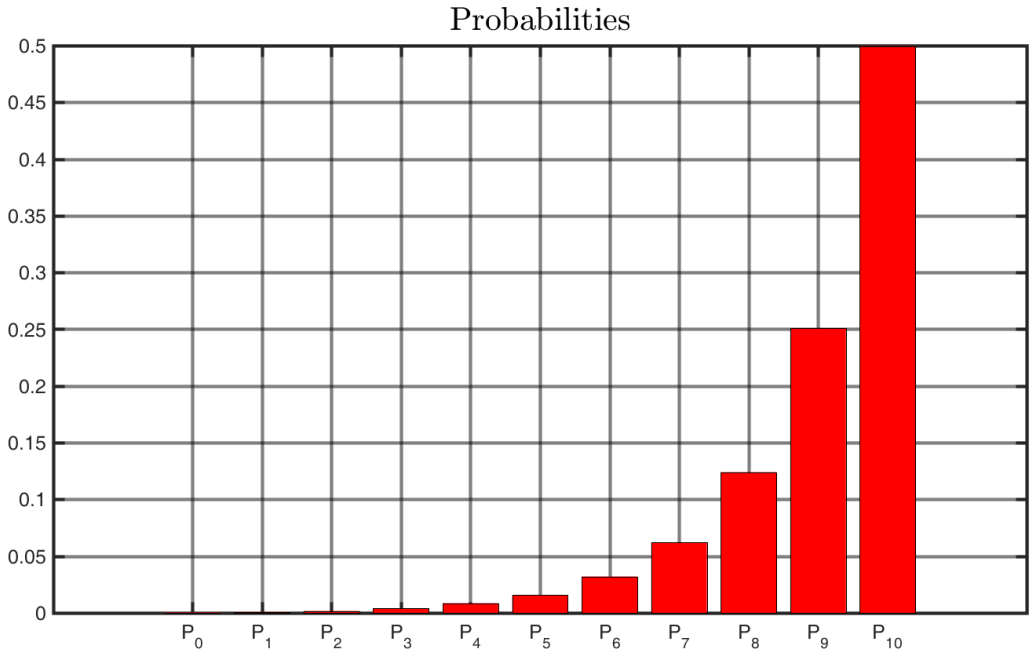


Figure 5: εργοδικές πιθανότητες για $\lambda = 10$

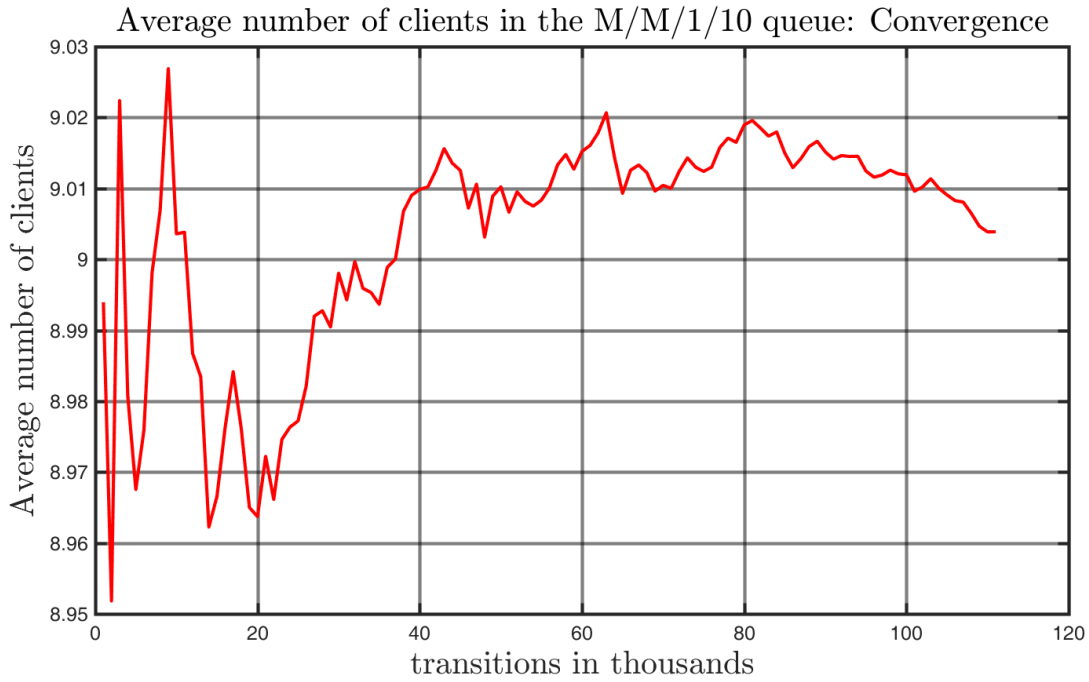


Figure 6: εξέλιξη του μέσου αριθμού πελατών στο σύστημα για $\lambda = 10$

ερώτημα 2: αναλυτικά αποτελέσματα

λ=1

$$P_{blocking} = 0 \qquad \mathbb{E}[n(t)] = 0.2587$$

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \\ P_{10} \end{bmatrix} = \begin{bmatrix} 0.797200077775617 \\ 0.161189966945363 \\ 0.0324157662287159 \\ 0.00733312963528791 \\ 0.00172217438404489 \\ 0.000138885030971362 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

λ=5

$$P_{blocking} = 0.0927118259714596 \qquad \mathbb{E}[n(t)] = 4.9918$$

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \\ P_{10} \end{bmatrix} = \begin{bmatrix} 0.0926591833098466 \\ 0.0909057777345838 \\ 0.0900958906328458 \\ 0.0894398820804380 \\ 0.0898772211153765 \\ 0.0887271814309085 \\ 0.0893872394188250 \\ 0.0911446944295965 \\ 0.0920841634676126 \\ 0.0929669404085071 \\ 0.0927118259714596 \end{bmatrix} \tag{4}$$

λ=10

$$P_{blocking} = 0.499594517881761 \qquad \mathbb{E}[n(t)] = 9.0035$$

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \\ P_{10} \end{bmatrix} = \begin{bmatrix} 0.000337901765198821 \\ 0.00083796377693077 \\ 0.00162192847295434 \\ 0.00401427297056200 \\ 0.00852864055361825 \\ 0.0157732543994810 \\ 0.0320871516232801 \\ 0.0622415051496229 \\ 0.124050496039791 \\ 0.250912334766037 \\ 0.499594517881761 \end{bmatrix} \quad (5)$$

ερώτημα 3

Σε γενικές γραμμές παρατηρούμε ότι καθώς $|\lambda - \mu| \to 0$ η προσομοίωση απαιτεί περισσότερες επαναλήψεις για να ολοκληρωθεί. Αυτό συμβαίνει διότι το σύστημα μπορεί να βρεθέι στην κάθε κατάσταση με ίση πιθανότητα και επομένως είναι δύσκολο να επέλθει η στεθερή κατάσταση. Μπορούμε να αγνοήσουμε τις αρχικές καταστάσεις όπου παρατηρείται το μεταβατικό φαινόμενο.

3 Προσομοίωση συστήματος M/M/1/5 με μεταβλητό μέσο ρυθμό εξυπηρέτησης

ερώτημα 1

Οι εργοδικές πιθανότητες των καταστάσεων του συστήματος υπολογίζονται με τις συναρτήσεις `ctmc` και `ctmcbd` και είναι ίσες με:

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0.162933 \\ 0.244399 \\ 0.244399 \\ 0.183299 \\ 0.109980 \\ 0.054990 \end{bmatrix} \tag{6}$$

Ο μέσος αριθμός πελατών στο σύστημα είναι:

$$\boxed{\mathbb{E}[n(t)] = 1.998} \tag{7}$$

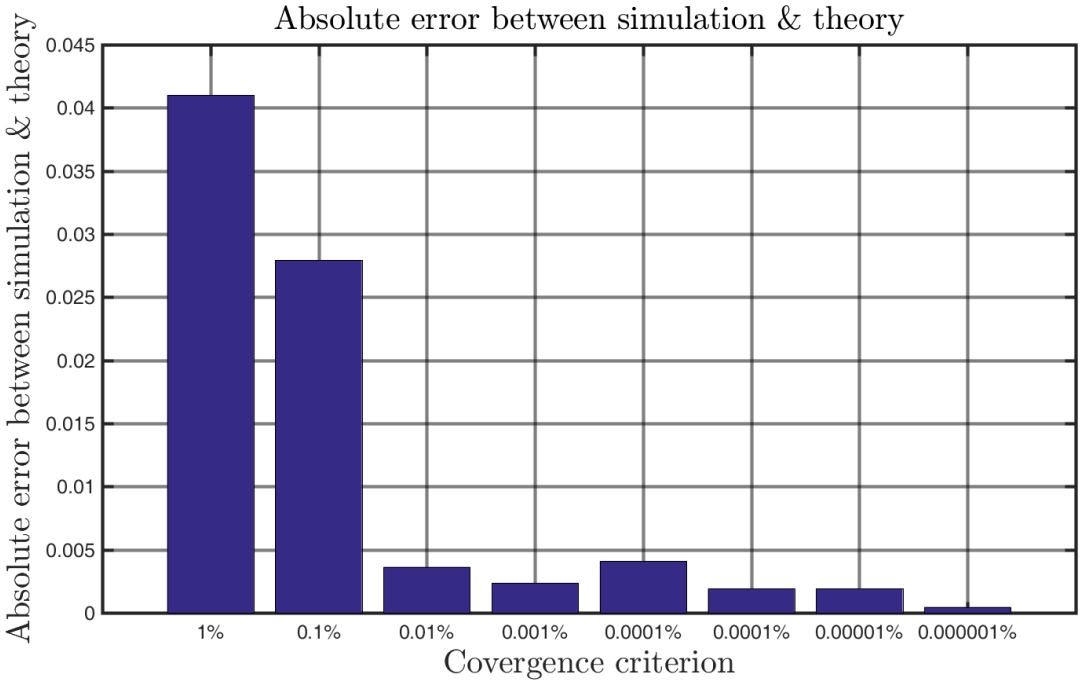
ερώτημα 2

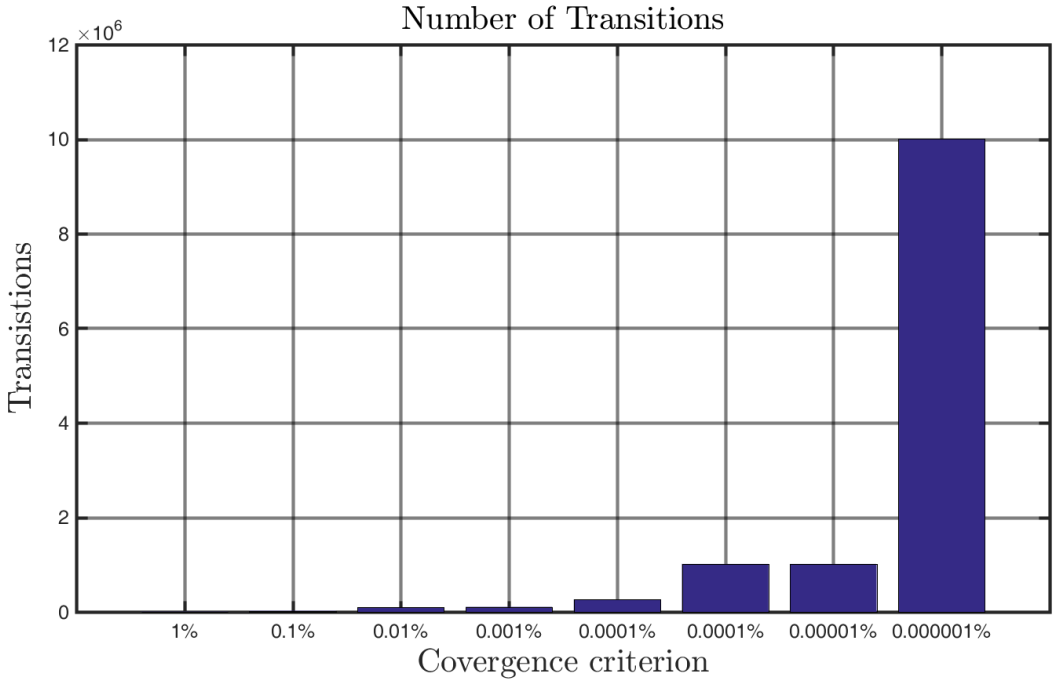
Λαμβάνουμε τα διαγράμματα της επόμενης σελίδας για τις προσομοιώσεις:

Από αυτά θα επέλεγα ως κριτήριο σύγκλισης

$$|e| < 0.001 \tag{8}$$

καθώς για παραπάνω ακρίβεια θυσιάζω υπολογιστικούς πόρους για πολύ μικρή βελτίωση στα αποτελέσματα. Το γεγονός ότι για αυστηρότερο κριτήριο σύγκλισης δεν παρατηρείται πάντα καλύτερη προσέγγιση της θωερητικής τιμής οφείλεται στο γεγονός ότι το κριτήριο σύγκλισης αφορά την απόλυτη διαφορά δύο διαδοχικών instances της προσομοίωσης και όχι τη θεωρητική τιμή. Τέλος, ως εγγύηση ώστε η προσομοίωσή να τερματίζει όταν το κριτήριο σύγκλισης είναι υπερβολικά αυστηρό είναι ο τερματισμός της όταν φτάσει κάποιο συγκεκριμένο αριθμό επαναλήψεων.


Absolute error between simulation & theory
Absolute error between simulation & theory
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1%
0.1%
0.01%
0.001%
0.0001%
0.0001%
0.00001%
0.000001%
Covergence criterion


Number of Transitions
×10^6
Transistions
12
10
8
6
4
2
0
1%
0.1%
0.01%
0.001%
0.0001%
0.0001%
0.00001%
0.000001%
Covergence criterion

4 Appendix

Listing 2: `qsmm1`

```
function [U, R, Q, X, p0] = qsmm1( lambda, mu )
  lambda = lambda(:)';
  mu = mu(:)';
  U = lambda ./ mu; % utilization
  rho = U;
  p0 = 1-rho;
  Q = rho ./ (1-rho);
  R = 1 ./ ( mu .* (1-rho) );
  X = lambda;
end
```

Listing 3: `qsmd1`

```
function [U, R, Q, X, p0] = qsmd1( lambda, mu )
  lambda = lambda(:)';
  mu = mu(:)';
  U = lambda ./ mu; % utilization
  rho = U;
  p0 = 1-rho;
  Q = rho + (0.5*rho.^2) ./ (1-rho);
  R = Q ./ lambda;
  X = lambda;
end
```

Listing 4: `PART 1`

```
clc;
clear all;
close all;


%%

lambda = 0.1:0.01:2.9;
mu = 3;

[U_mm1, R_mm1, Q_mm1, X_mm1, p0_mm1] = qsmm1(lambda, mu);
[U_md1, R_md1, Q_md1, X_md1, p0_md1] = qsmd1(lambda, mu);

%% PLOTS

fontsize = 12;
```

```
lineWidth = 1;
plotLineWidth = 4;
width=1024;
height=568;

fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(lambda/mu,Q_mm1,'Color','b','LineWidth',plotLineWidth);
plot(lambda/mu,Q_md1,'Color','r','LineWidth',plotLineWidth);
temp = gca;

temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off
% ylim([0 50]);

xlabel(['System utilization $\rho$'],'FontSize',20,'Interpreter','latex');
title(['Average number of customers in the system $\mathbf{E}[n(t)]$'],'
   FontSize',20, 'Interpreter','latex');
legend('MM1','MD1');
grid on;
saveas(gcf,'figure1.png');
%%
fig2 = figure();
fig2.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(lambda/mu,R_mm1,'Color','b','LineWidth',plotLineWidth);
plot(lambda/mu,R_md1,'Color','r','LineWidth',plotLineWidth);
temp = gca;

temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off
% ylim([0 50]);

xlabel(['System utilization $\rho$'],'FontSize',20,'Interpreter','latex');
```

```
title(['Average delay time $\mathbf{R}$'],'FontSize',20, 'Interpreter','latex
    ');
legend('MM1','MD1');
grid on;
saveas(gcf,'figure2.png');
```

Listing 5: PART 2

```
clc;
clear all;
close all;
%%
rng(1);     % seed
P = zeros(11,1);
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;
file = [];
covergence_criterion =  0.00001 ;
maxTransitions = 1000000;
lambda = 5;
mu = 5;
N = 10;
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
     end

     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);
     end
```

```
34
35      to_plot(index) = mean_clients;
36
37      if abs(mean_clients - previous_mean_clients) <  covergence_criterion ||
            transitions > maxTransitions % convergence test
38        break;
39      end
40
41      previous_mean_clients = mean_clients;
42
43    end
44
45    random_number = rand(1); % generate a random number (Uniform distribution)
46    if current_state == 0 || random_number < threshold % arrival
47      old_state = current_state;
48      total_arrivals = total_arrivals + 1;
49      try % to catch the exception if variable arrivals(i) is undefined.
            Required only for systems with finite capacity.
50        arrivals(current_state + 1) = arrivals(current_state + 1) + 1; %
              increase the number of arrivals in the current state
51        Y = arrivals(current_state + 1);
52        if current_state ~= N
53          current_state = current_state + 1;
54        end
55      catch
56        arrivals(current_state + 1) = 1;
57        Y = arrivals(current_state + 1);
58        if current_state ~= N
59          current_state = current_state + 1;
60        end
61      end
62    else % departure
63      if current_state ~= 0 % no departure from an empty system
64        current_state = current_state - 1;
65      end
66    end
67
68    % DEBUGGING
69    X = 0;
70    Y = arrivals(old_state+1);
71    if (random_number < threshold)
72        X=1;
73    end
74    if (transitions < 31)
75        newrow = [old_state X Y];
```

```
76        file = [file ; newrow];
77    end
78
79
80   end
81
82  for i=1:1:length(arrivals)
83    display(P(i));
84  end
85
86  %% RESULTS
87  csvwrite(['debug',num2str(lambda),'.csv'],file);
88  P_blocking = P(N+1);
89  i = 0:(N);
90  MeanClients = sum (i.*P'); % = to_plot(length(index)) = mean_clients dld h
        teleutaia timh
91  gamma = lambda*(1-P_blocking);
92  E_T = MeanClients / gamma;
93  %% PLOT 1
94
95  fontsize = 12;
96  lineWidth = 1;
97  plotLineWidth = 4;
98  width=1024;
99  height=568;
100 fig1 = figure();
101 fig1.Color = 'w';
102 set(gcf,'units','pixels','position',[0,0,width,height]) ;
103
104 plot(to_plot,'r','LineWidth',2*lineWidth);
105 temp = gca;
106 temp.Color = 'w';
107 temp.LineWidth = 2*lineWidth;
108 temp.GridColor = 'k';
109 temp.GridAlpha = 0.5;
110 temp.FontSize = fontsize;
111 hold off
112 % ylim([0 50]);
113 xlabel(['transitions in thousands'],'FontSize',20,'Interpreter','latex');
114 ylabel(['Average number of clients'],'FontSize',20,'Interpreter','latex');
115 title(['Average number of clients in the M/M/1/10 queue: Convergence'],'
        FontSize',20, 'Interpreter','latex');
116 grid on;
117 saveas(gcf,['figure2_1_lambda=',num2str(lambda),'.png']);
118
```

```
119
120 %% PLOT 2
121 fig1 = figure();
122 fig1.Color = 'w';
123 set(gcf,'units','pixels','position',[0,0,width,height]) ;
124 bar(0:length(P)-1,P,'r');
125 set(gca,'XTickLabel',{'P_{0}','P_{1}','P_{2}','P_{3}','P_{4}','P_{5}','P_{6}'
    ,'P_{7}','P_{8}','P_{9}','P_{10}'});
126 temp = gca;
127 temp.Color = 'w';
128 temp.LineWidth = 2*lineWidth;
129 temp.GridColor = 'k';
130 temp.GridAlpha = 0.5;
131 temp.FontSize = fontsize;
132 hold off
133 % ylim([0 50]);
134 % xticks([1:10]);
135 % xticklabels({'P_{0}','P_{0}','P_{0}','P_{0}','P_{0}','P_{0}','P_{0}','P_
    {0}','P_{0}','P_{0}','P_{0}'})
136 title(['Probabilities'],'FontSize',20, 'Interpreter','latex');
137 grid on;
138 saveas(gcf,['figure2_2_lambda=',num2str(lambda),'.png']);
```

Listing 6: PART 3

```
1  clc;
2  clear all;
3  close all;
4  %%
5  P = zeros(11,1);
6  total_arrivals = 0; % to measure the total number of arrivals
7  current_state = 0;  % holds the current state of the system
8  previous_mean_clients = 0; % will help in the convergence test
9  index = 0;
10 file = [];
11 covergence_criterion =  0.1 ;
12
13 maxTransitions = 10000000;
14 lambda = [3, 3, 3, 3, 3];
15 mu = 2:6;
16 N = 5;
17 threshold = lambda./(lambda + mu); % the threshold used to calculate
       probabilities
18
19 transitions = 0; % holds the transitions of the simulation in transitions
```

```
   steps

for loop = 1:8
    rng(1);     % seed
    P = zeros(11,1);
    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;
    transitions = 0;
    to_plot=[];
    arrivals = [];
    covergence_criterion = covergence_criterion/10;
    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
        end

        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);
        end

        to_plot(index) = mean_clients;

        if abs(mean_clients - previous_mean_clients) <  covergence_criterion
           || transitions > maxTransitions % convergence test
          break;
        end

        previous_mean_clients = mean_clients;

      end

      random_number = rand(1); % generate a random number (Uniform
         distribution)
      if current_state == 0 || random_number < threshold(current_state) %
         arrival
```

```
58          total_arrivals = total_arrivals + 1;
59          try % to catch the exception if variable arrivals(i) is undefined.
               Required only for systems with finite capacity.
60            arrivals(current_state + 1) = arrivals(current_state + 1) + 1; %
                 increase the number of arrivals in the current state
61            if current_state ~= N
62              current_state = current_state + 1;
63            end
64          catch
65            arrivals(current_state + 1) = 1;
66            if current_state ~= N
67              current_state = current_state + 1;
68            end
69          end
70        else % departure
71          if current_state ~= 0 % no departure from an empty system
72            current_state = current_state - 1;
73          end
74        end
75
76      end
77
78      to_plot_mc(loop) = abs(mean_clients-1.998);
79      to_plot_tr(loop) = transitions;
80      display(loop)
81  end
82
83  %% PLOTS
84
85  fontsize = 12;
86  lineWidth = 1;
87  plotLineWidth = 4;
88  width=1024;
89  height=568;
90  fig1 = figure();
91  fig1.Color = 'w';
92  set(gcf,'units','pixels','position',[0,0,width,height]) ;
93  bar(to_plot_mc);
94  set(gca,'XTickLabel',{'1%','0.1%','0.01%','0.001%','0.0001%','0.0001%','
       0.00001%','0.000001%'});
95  temp = gca;
96  temp.Color = 'w';
97  temp.LineWidth = 2*lineWidth;
98  temp.GridColor = 'k';
99  temp.GridAlpha = 0.5;
```

```
temp.FontSize = fontsize;
hold off
ylabel([' Absolute error between simulation \& theory'],'FontSize',20,'
    Interpreter','latex');
xlabel(['Covergence criterion'],'FontSize',20,'Interpreter','latex');
title([' Absolute error between simulation \& theory'],'FontSize',20,'
    Interpreter','latex');
grid on;
saveas(gcf,['figure3_1.png']);

fig2 = figure();
fig2.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
bar(to_plot_tr);
set(gca,'XTickLabel',{'1%','0.1%','0.01%','0.001%','0.0001%','0.0001%','
    0.00001%','0.000001%'});
temp = gca;
temp.Color = 'w';
temp.LineWidth = 2*lineWidth;
temp.GridColor = 'k';
temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off
ylabel(['Transistions in thousands'],'FontSize',20,'Interpreter','latex');
xlabel(['Covergence criterion'],'FontSize',20,'Interpreter','latex');
title(['Number of Transitions'],'FontSize',20,'Interpreter','latex');

grid on;
saveas(gcf,['figure3_2.png']);
```