ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ

ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

2η Ομάδα Ασκήσεων

Συστήματα Αναμονής (Queuing Systems)

ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΟΥΣΗΣ

03119005

georg.dimi00@gmail.com

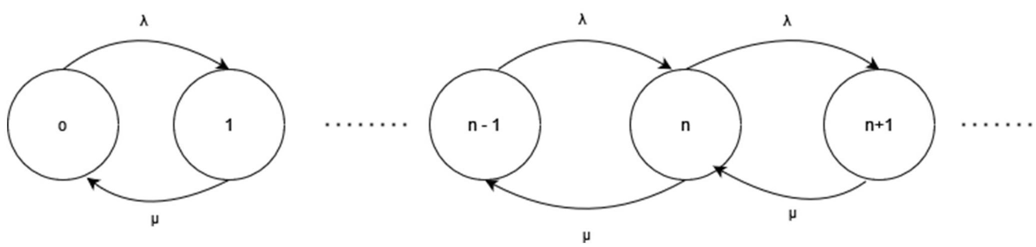
Θεωρητική μελέτη της ουράς M/M/1

Οι αφίξεις ακολουθούν κατανομή Poisson με παράμετρο λ πελάτες/sec και οι εξυπηρετήσεις ακολουθούν εκθετική κατανομή με παράμετρο μ πελάτες/sec.

Στην ουρά M/M/1 οι χρόνοι μεταξύ αφίξεων και οι χρόνοι εξυπηρέτησης ακολουθούν εκθετική κατανομή.

(α) Μια αλυσίδα Markov λέγεται εργοδική, αν όλες της οι καταστάσεις είναι εργοδικές. Μια κατάσταση λέγεται εργοδική αν είναι απεριοδική και θετικά επαναληπτική. Η απεριοδικότητα εξασφαλίζεται από το γεγονός ότι σε κάθε κατάσταση της αλυσίδας Markov μπορούμε ή να πάμε στη προηγούμενή της ή στην επόμενη ή να μείνουμε σε αυτήν (εκτός από τη μηδενική κατάσταση που δεν έχει προηγούμενη).

Για τη δεύτερη προϋπόθεση θέλουμε η αλυσίδα Markov να είναι αμείωτη (από κάθε κατάσταση μπορούμε να πάμε σε κάθε άλλη) και ομοιογενής. Μας αρκεί η κατανομή πιθανότητας που περιγράφει την πιθανότητα να βρεθεί η διαδικασία στη κατάσταση j στο μακρινό μέλλον να συγκλίνει σε μη μηδενικές τιμές.



Ενδιαφερόμαστε για τη μόνιμη κατάσταση της διαδικασίας, στην οποία το σύστημα πρέπει να βρίσκεται σε ισορροπία και η ροή που εξέρχεται από ένα σύνολο καταστάσεων πρέπει να ισούται με τη ροή που εισέρχεται σε αυτό.

Θεωρώντας ως σύνολο καταστάσεων τις {0, 1, …, n – 1} έχω ότι εξέρχεται από αυτές ροή $\lambda_{n-1}p_{n-1}$, όπου λ_{n-1} ο ρυθμός γεννήσεων στη κατάσταση n – 1 και p_{n-1} η πιθανότητα να βρεθούμε σε αυτή τη κατάσταση. Το λέμε ροή, διότι εκφράζει κατά κάποιο τρόπο τη ροή της πιθανότητας μέσα από το σύνολο καταστάσεων.

Η εισερχόμενη ροή είναι: $\mu_n p_n$ κατ' αναλογία με τα παραπάνω και θα έχουμε:

$\begin{cases} \lambda_{n-1}p_{n-1} = \mu_n p_n \\ \lambda_0 p_0 = \mu_1 p_1 \end{cases}$, καθώς και τη σχέση $\sum_{i\geq 0} p_i = 1$, αφού είναι πιθανότητες.

Στη περίπτωση της αλυσίδας του σχήματος (στην οποία έχουμε παραλείψει να δείξουμε τη περίπτωση που μια κατάσταση γυρίζει στον εαυτό της, αυτό όμως δεν επηρεάζει τη παραπάνω ανάλυση) έχουμε ότι το παραπάνω σύστημα γράφεται:

$\begin{cases} \lambda p_{n-1} = \mu p_n \\ \lambda p_0 = \mu p_1 \end{cases}$, καθώς και τη σχέση $\sum_{i\geq 0} p_i = 1$, αφού είναι πιθανότητες.

Θέτουμε ως ρ = λ/μ και λύνουμε την αναδρομή:

$$\begin{cases} p_n = \rho p_{n-1} \\ \rho p_{n-1} = \rho^2 p_{n-2} \\ \dots \\ \rho^{n-2} p_2 = \rho^{n-1} p_1 \\ \rho^{n-1} p_1 = \rho^n p_0 \end{cases} \Rightarrow (\pi\rho\acute{o}\sigma\theta\varepsilon\sigma\eta\ \kappa\alpha\tau\acute{\alpha}\ \mu\acute{\varepsilon}\lambda\eta) p_n = \rho^n p_0$$

$$1 = \sum_{i\geq 0} p_i = p_0 \sum_{i\geq 0} \rho^i$$

$$p_0 = \left(\sum_{i\geq 0} \rho^i \right)^{-1}$$

Θέλουμε οι καταστάσεις του συστήματος να είναι θετικές επαναληπτικές, οπότε σύμφωνα με το θεώρημα 2.2 σχετικά με την μόνιμη κατάσταση του βιβλίου «Ανάλυση Επίδοσης Υπολογιστικών Συστημάτων» (σελίδα 30) έχουμε ότι αφού θέλουμε το p_0 να είναι θετικός πραγματικός αριθμός και το ρ > 0 πρέπει η σειρά $\sum_{i\geq 0} \rho^i$ να συγκλίνει σε θετικό αριθμό άρα πρέπει ρ < 1.

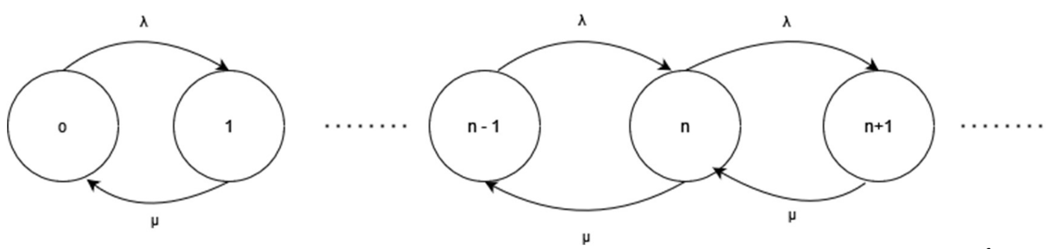
Η συνθήκη που εξασφαλίζει την εργοδικότητα, λοιπόν, είναι λ < μ.

Παίρνουμε, λοιπόν:

$$1 = \sum_{i\geq 0} p_i = p_0 \frac{1}{1-\rho}\ \acute{\alpha}\rho\alpha\ p_0 = 1 - \rho$$

Αναφέρουμε τα παραπάνω πιο συγκεντρωμένα:

Η απαραίτητη συνθήκη ώστε η ουρά M/M/1 να είναι εργοδική είναι λ < μ. Το διάγραμμα του ρυθμού μεταβάσεων είναι:



Οι εργοδικές πιθανότητες των καταστάσεων είναι $p_n = (1-\rho)\rho^n, n = 0, 1, \dots \mu\varepsilon\ \rho = \frac{\lambda}{\mu}$

(β) Οι πιθανότητες p_n εκφράζουν τη πιθανότητα να βρεθούμε στη κατάσταση n της διαδικασίας. Εδώ ασχολούμαστε με πελάτες που εξυπηρετούνται άρα εκφράζουν τη πιθανότητα στο σύστημα να βρίσκονται n πελάτες. Η ουρά αυτή ακολουθεί γεωμετρική κατανομή με μέση τιμή E(n) = ρ/(1 – ρ).

Με βάση τον τύπο του Little ο μέσος χρόνος παραμονής στο σύστημα είναι E(n)/λ, δηλαδή, T = 1/[μ(1 – ρ)].

Ο μέσος χρόνος εξυπηρέτησης (αφού οι χρόνοι εξυπηρέτησης ακολουθούν εκθετική κατανομή) είναι 1/μ.

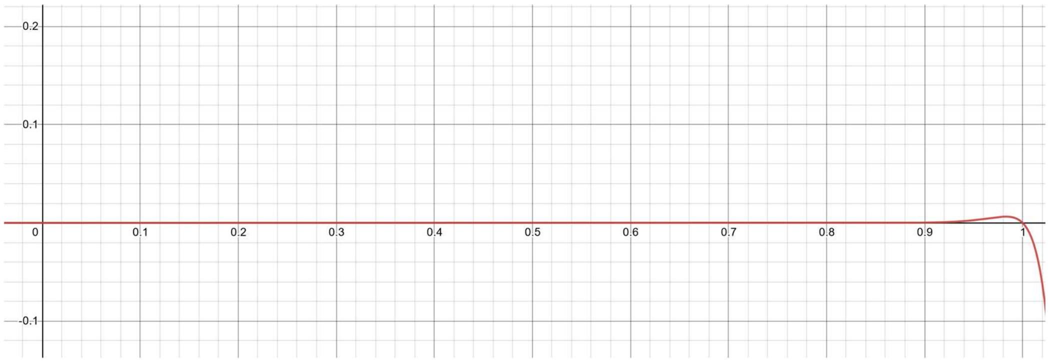
Ο μέσος χρόνος παραμονής είναι το άθροισμα του χρόνου αναμονής και του χρόνο εξυπηρέτησης.

Συνεπώς, ο μέσος χρόνος αναμονής στο σύστημα θα είναι $W = \frac{1}{\mu(1-\rho)} - \frac{1}{\mu} = \frac{1}{\mu}\frac{\rho}{1-\rho}$.

Ως μέσο χρόνο καθυστέρησης ενός πελάτη στο σύστημα εννοούμε τον συνολικό χρόνο που αυτός παραμένει στο σύστημα άρα είναι $T = \frac{1}{\mu(1-\rho)}$.

(γ) Είναι $p_n = (1-\rho)\rho^n \ \mu\varepsilon\ \rho < 1$. Αυτό σημαίνει ότι καθώς το n μεγαλώνει η p_n πάει να γίνει 0.

Μας νοιάζει η $p_{57} = (1-\rho)\rho^{57}$.



Αυτό είναι το γράφημα της p_{57} συναρτήσει του ρ για 0 < ρ < 1. Βλέπουμε ότι η πιθανότητα να βρεθούν 57 πελάτες στο σύστημα είναι σχεδόν μηδενική, άρα είναι πολύ απίθανο να συμβεί. Αποκτά μία κάπως σημαντική τιμή για ρ κοντά στη μονάδα.

Για να ερμηνεύσουμε αυτή τη παρατήρηση. Ας σκεφτούμε το ρ ως ρ = λ/μ.

Τα παραπάνω μας λένε ότι αν έχουμε πολύ μικρότερη συχνότητα άφιξης πελατών από τη συχνότητα που τους εξυπηρετούμε είναι πολύ δύσκολο να βρεθούν στο σύστημα 57 πελάτες. Ωστόσο, αν η συχνότητα άφιξης των πελατών πλησιάζει αυτή με την οποία τους εξυπηρετούμε είναι πιο πιθανό να βρεθεί σημαντικό πλήθος πελατών (όπως 57 πελάτες) ταυτόχρονα στο σύστημα (που περιμένουν εξυπηρέτηση).

Ανάλυση ουράς M/M/1 με Octave

(α) Θέλουμε ρυθμούς εξυπηρέτησης τέτοιους ώστε λ < μ. Συνεπώς, οι επιτρεπτές τιμές είναι:

5 πελάτες/min < μ ≤ 10 πελάτες/min.

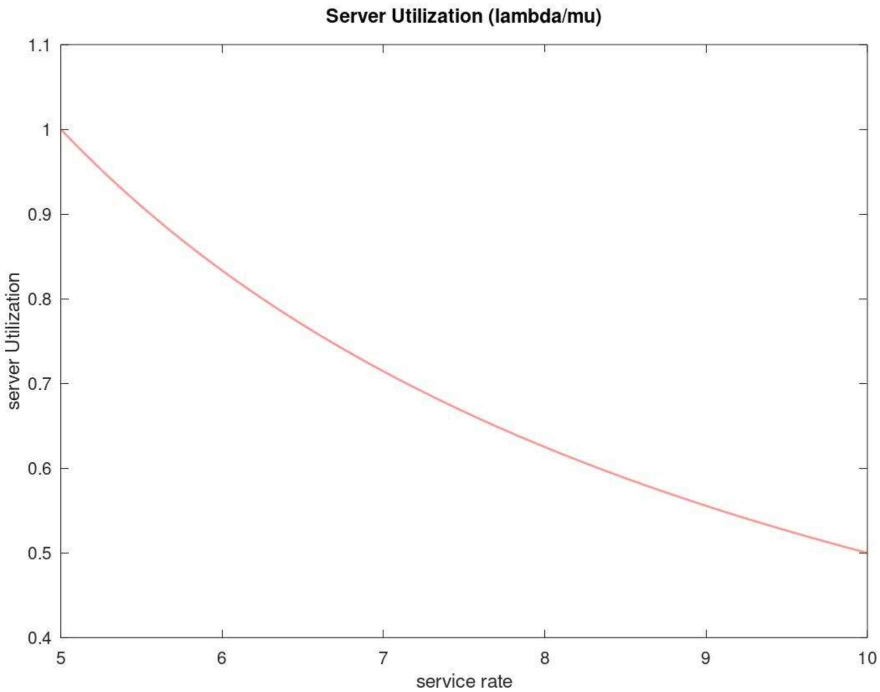
(β)

Για το ερώτημα (β) ο κώδικας είναι ο παρακάτω:

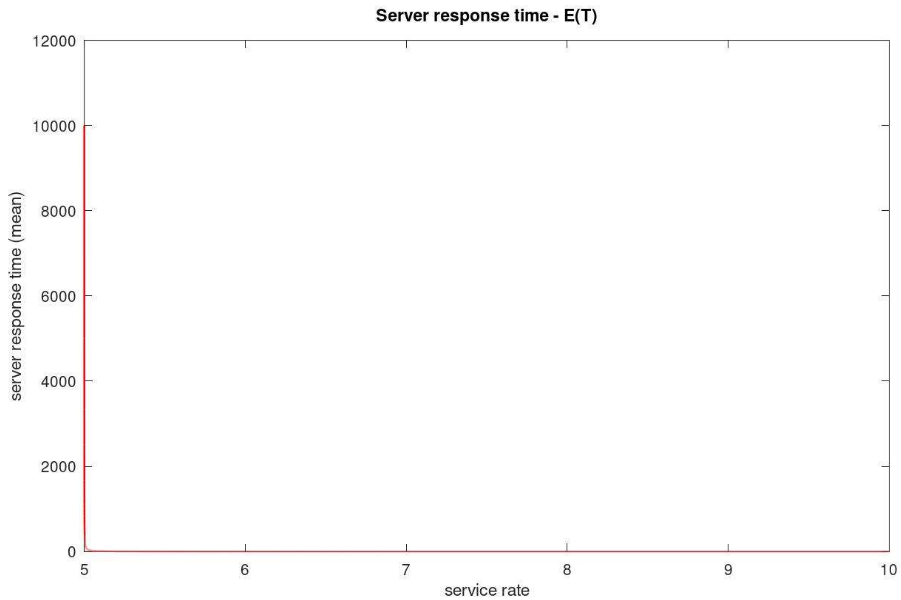
```
clc;
close all;
clear all;

pkg load queueing;

colors = "rbkmg";

%system M/M/1
%[U, R, Q, X, p0] = qsmm1(lambda, mu)
%pk = qsmm1(lambda, mu, k)
%lambda is the arrival rate
%mu is the service rate
%k is the number of requests in the system (including the one being
served)

%U is the server utilization
%R server response time
%Q average number of requests in the system
%X sever throughput. If the system is ergodic (mu > lambda) we always
have
%X = lambda
%p0 steady-state probability that there no requests in the system
%pk steady-state probability that there are k requests in the system
%(including the one being served)

%if the function is called without k, then lambda and mu can be
vectors of the
%same size

lambda = 5;
step = 0.0001;
mu = (lambda+step):step:10;

[U, R, Q, X, p0] = qsmm1(lambda, mu);

for i = 1 : 10
  index=find(mu == 5 + 0.5*i);
  [5 + 0.5*i, R(index)]
endfor
['(service rate)',' (response time ?(?))']
```

```
%diagram of U in relation to mu

figure(1);

plot(mu, U, colors(1), "linewidth", 1.2);

title("Server Utilization");
xlabel("service rate");
ylabel("server Utilization (lambda/service rate)");

%diagram of R in relation to U

figure(2);

plot(mu, R, colors(1), "linewidth", 1.2);

title("Server response time - ?(?)");
xlabel("service rate");
ylabel("server response time (mean)");

figure(3);

plot(mu, R, colors(1), "linewidth", 1.2);
axis([5 10 0 100]);
%values of R for mu close to lambda are too large

title("Server response time - ?(?)");
xlabel("service rate");
ylabel("server response time (mean)");

figure(4);

plot(mu, Q, colors(1), "linewidth", 1.2);

title("Average number of clinets in the system");
xlabel("service rate");
ylabel("average number of clients in the system");

figure(5);

plot(mu, Q, colors(1), "linewidth", 1.2);
axis([5 10 0 100*lambda]);
%values of Q too large for mu close to lambda

title("Average number of clinets in the system");
xlabel("service rate");
ylabel("average number of clients in the system");

figure(6);

plot(mu, X, colors(1), "linewidth", 1.2);

title("Server throughput");
xlabel("service rate");
ylabel("server throughput");
```

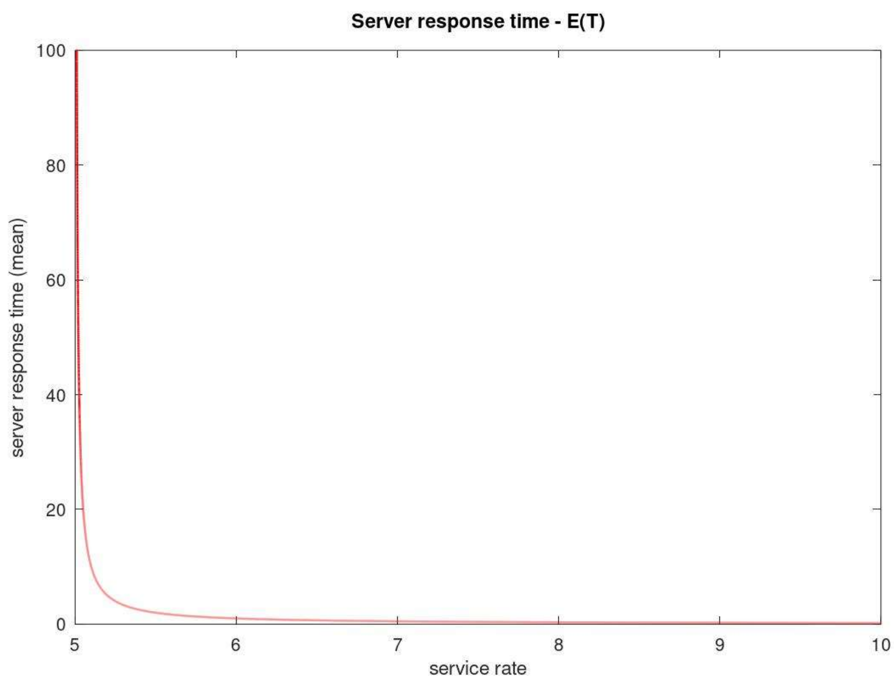
Παρουσιάζουμε και τα σχετικά διαγράμματα:

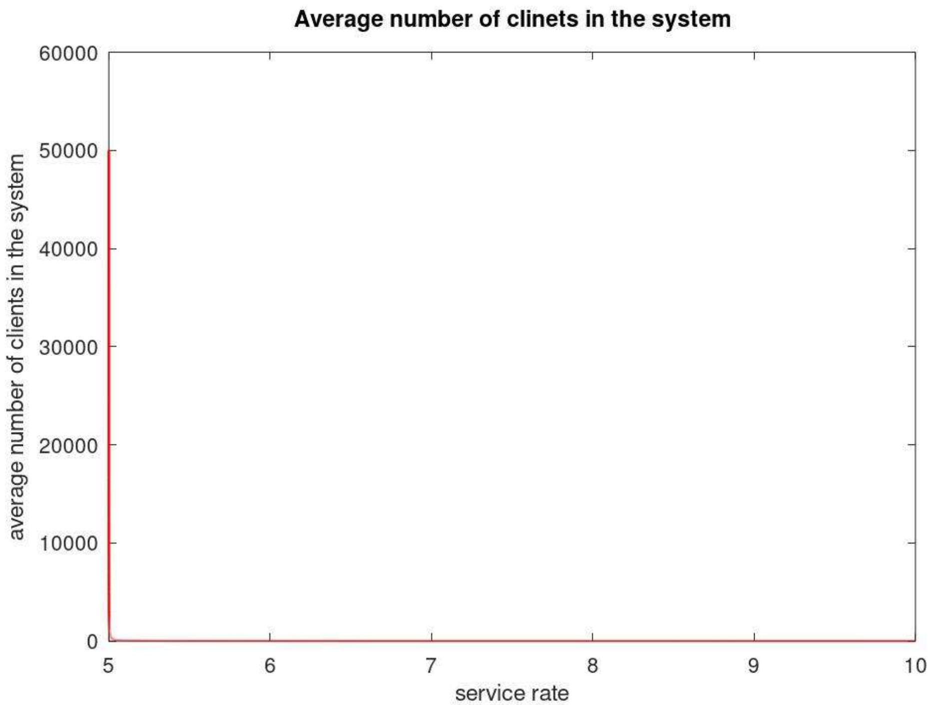


Το ρ = λ/μ που είδαμε στο πρώτο ερώτημα ονομάζεται και ένταση κυκλοφορίας, αλλά και βαθμός χρησιμοποίησης της μονάδας εξυπηρέτησης (επειδή εκφράζει την πιθανότητα να μην είναι άδειο το σύστημα, δηλαδή, να είναι απασχολημένη η μονάδα εξυπηρέτησης, εφ' όσον ρ = 1 – p0, όπου p0 η πιθανότητα το σύστημα να είναι άδειο).

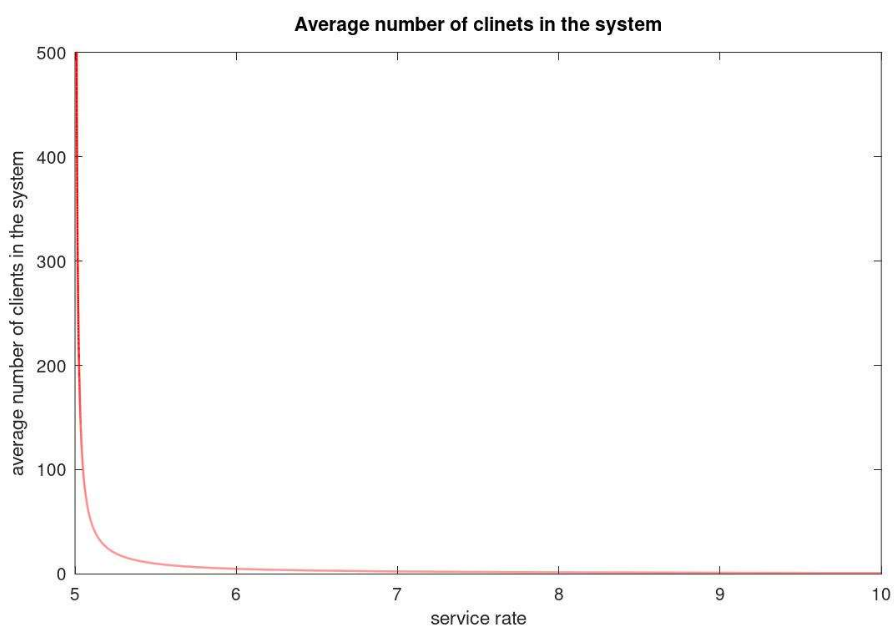


Ξανακάνουμε το ίδιο διάγραμμα βάζοντας όριο στον κατακόρυφο άξονα για να είναι πιο εύκολο να δούμε τι ακριβώς παρουσιάζεται.

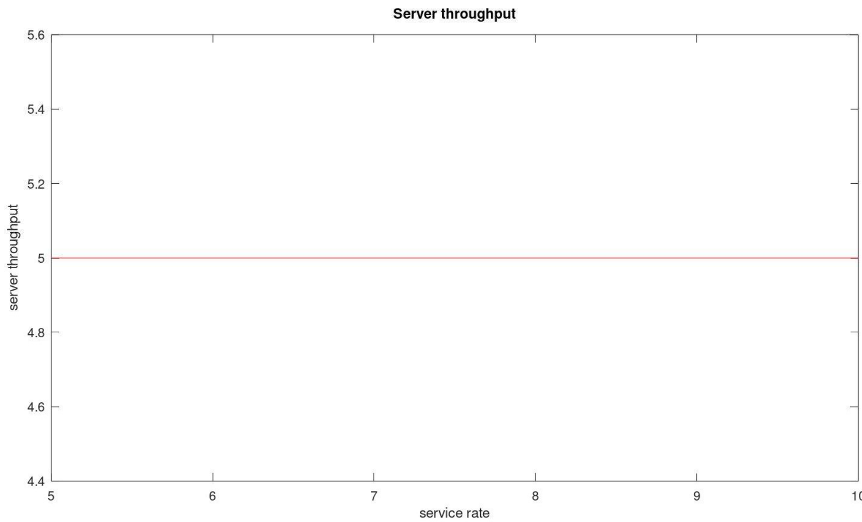




Όπως προηγουμένως εξαιτίας του μεγάλου αριθμού πελατών που θα υπάρχουν στο σύστημα για τιμές του ρυθμού εξυπηρέτησης κοντά στον ρυθμό άφιξης πελατών. Έχουμε βάλει όριο στον κατακόρυφο άξονα και παρουσιάζουμε ξανά το μέσο αριθμό πελατών στο σύστημα ως προς το ρυθμό εξυπηρέτησης παρακάτω.

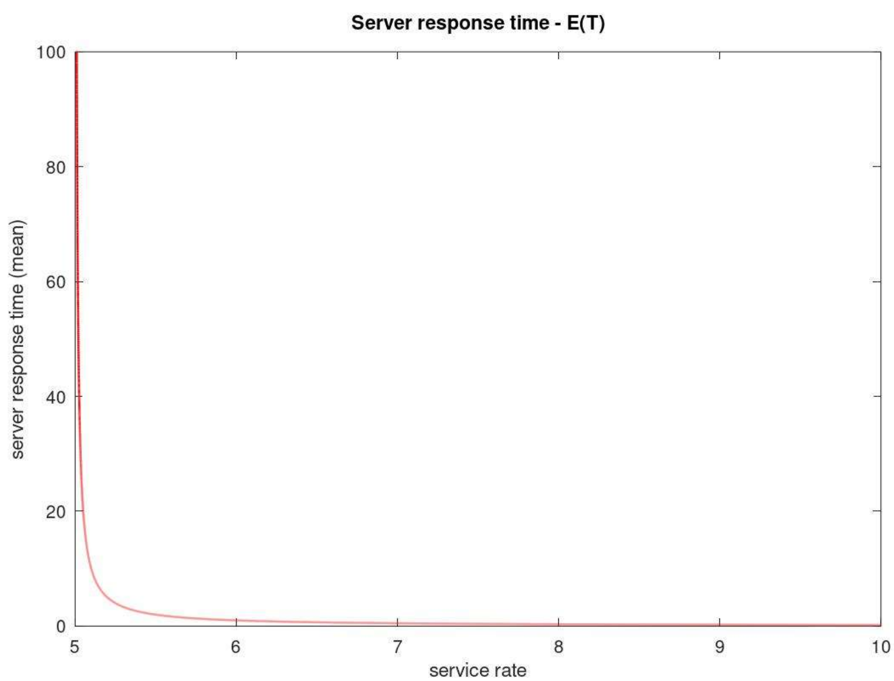


Παρατηρούμε ότι το διάγραμμα αυτό είναι, ουσιαστικά, το πενταπλάσιο του διαγράμματος του μέσου χρόνου καθυστέρησης συστήματος E(T). Η αιτιολογία είναι ότι από τον τύπο του Little που χρησιμοποιήσαμε και στο ερώτημα (α) έχουμε πώς $E(n) = \lambda E(T)$, όπου E(n) είναι ο μέσος αριθμός πελατών στο σύστημα και λ είναι ο ρυθμός αφίξεων που εδώ είναι 5 πελάτες/min.



(σχόλια για τη ρυθμαπόδοση στο (δ))

(γ) Το διάγραμμα που μας δείχνει τον μέσο χρόνο καθυστέρησης E(T) σε σχέση με το ρυθμό εξυπηρέτησης μ είναι το παρακάτω:



Και η συνάρτηση που εκφράζει είναι:

$$E(T) = f(\mu) = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu(1-\frac{\lambda}{\mu})} = \frac{1}{\mu-\lambda} = \frac{1}{\mu-5}$$

Τα ποσά είναι αντιστρόφως ανάλογα, συνεπώς, το μεγαλύτερο δυνατό μ μας δίνει το μικρότερο χρόνο καθυστέρησης.

Παρουσιάζουμε τα παρακάτω δεδομένα από το Octave:

μ (ρυθμός εξυπηρέτησης)	E(T) (χρόνος καθυστέρησης συστήματος)
5.5	2.0000
6	1.0000
6.5	0.6667
7	0.5000
7.5	0.4000
8	0.3333
8.5	0.2857
9	0.2500
9.5	0.2222
10	0.2000

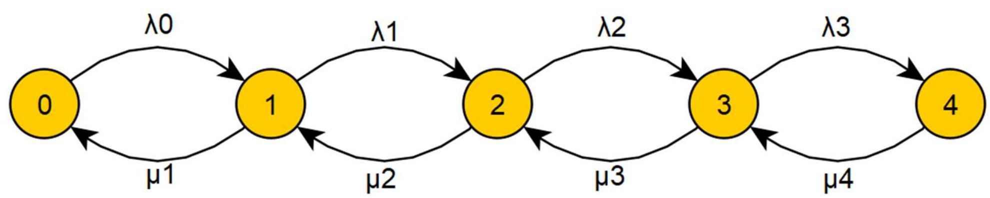
Το μ είναι σε πελάτες/min και το E(T) σε minutes.

Η αύξηση του μέσου ρυθμού εξυπηρέτησης (μ) πιθανώς επιβαρύνει σε πόρους το σύστημα ή έχει κάποια οικονομική επιβάρυνση. Συνεπώς, πήραμε τα παραπάνω στοιχεία και με βάση αυτά μπορούμε να διαλέξουμε να κατασκευάσουμε το σύστημα με ένα μ που δεν είναι το μέγιστο δυνατό, αλλά ταυτόχρονα ο μέσος χρόνος καθυστέρησης του συστήματος να μην μας είναι πολύ επιζήμιος.

(δ) Η ρυθμαπόδοση είναι γ = λ(1 – P(blocking)) = λ = 5 πελάτες/min ανεξαρτήτως του ρυθμού εξυπηρέτησης μ. Αυτό παρατηρείται και στο διάγραμμα που κατασκευάσαμε στο ερώτημα (β). Το σύστημα M/M/1 έχει άπειρη χωρητικότητα. Επίσης, έχουμε εξασφαλίσει ότι μ > λ το οποίο μπορούμε να ερμηνεύσουμε ως αδυναμία πρόκλησης συμφόρησης στο σύστημα (ο ρυθμός που εξυπηρετούμε είναι μεγαλύτερος από το ρυθμό που φτάνουν σε εμάς πελάτες). Συνεπώς, το P(blocking) = 0.

Διαδικασία γεννήσεων θανάτων (birth – death process): εφαρμογή σε σύστημα M/M/1/K

(α) Για το σύστημα M/M/1/4 οι καταστάσεις του συστήματος είναι: {0, 1, 2, 3, 4} πελάτες μέσα στο σύστημα.



Για το παραπάνω σύστημα έχουμε $\begin{cases} \lambda_i = \frac{\lambda}{i+1}, i = 0, 1, 2, 3 \\ \mu_i = \mu, i = 1, 2, 3, 4 \end{cases}$

λ = 5 πελάτες/sec και μ = 10 πελάτες/sec.

Η παραπάνω αλυσίδα Markov είναι αμείωτη, απεριοδική και έχει πεπερασμένο πλήθος καταστάσεων άρα είναι πάντοτε εργοδική.

Λόγω της ισορροπίας που υπάρχει στο σύστημα όταν ασχολούμαστε με τις εργοδικές πιθανότητες της κάθε κατάστασης χρησιμοποιούμε παρόμοια επιχειρήματα με αυτά του (α) ερωτήματος της 1ης άσκησης σχετικά με την εισερχόμενη και εξερχόμενη ροή σε σύνολα καταστάσεων και έχουμε:

$$\lambda_{n-1} p_{n-1} = \mu_n p_n, n = 1, 2, 3, 4 \; και \; \sum_{i=0}^{4} p_i = 1$$

Όλα τα μ_n είναι ίδια μεταξύ τους και θέτουμε ως ρ = λ/μ. Η 1η σχέση γίνεται $p_n = \frac{\rho}{n} p_{n-1}$.

Λύνουμε την αναδρομή:

$$\begin{cases} p_n = \frac{\rho}{n} p_{n-1} \\ \frac{\rho}{n} p_{n-1} = \frac{\rho}{n} \frac{\rho}{n-1} p_{n-2} \\ \ldots \\ \frac{\rho^{n-2}}{n(n-2) \cdot \ldots \cdot 3} p_2 = \frac{\rho^{n-1}}{n(n-1) \cdot \ldots \cdot 2} p_1 \\ \frac{\rho^{n-1}}{n(n-2) \cdot \ldots \cdot 2} p_1 = \frac{\rho^{n}}{n(n-1) \cdot \ldots \cdot 1} p_0 \end{cases} \Longrightarrow (πρόσθεση \; κατά \; μέλη)$$

Και παίρνουμε ότι:

$$p_n = \frac{\rho^n}{n!} p_0, n = 0, 1, 2, 3, 4$$

$$1 = \sum_{i=0}^{4} p_i = p_0 \sum_{i=0}^{4} \frac{\rho^i}{i!} \;\; άρα \; p_0 = \left(\sum_{i=0}^{4} \frac{\rho^i}{i!} \right)^{-1}$$

ρ = λ/μ = 5/10 = 0.5

Συνεπώς, υπολογίζουμε ότι: $p_0 = 0.6066350711$

Χρησιμοποιώντας τώρα τη σχέση $p_n = \frac{\rho}{n} p_{n-1} = \frac{1}{2n} p_{n-1}$ υπολογίζουμε τις υπόλοιπες πιθανότητες:

$$p_0 = 0.6066$$
$$p_1 = 0.3033$$
$$p_2 = 0.0758$$
$$p_3 = 0.0126$$
$$p_4 = 0.0017$$

Από τους τύπους υπολογίσαμε ότι $p_4 = 0.00158$, το οποίο θα έδινε στρογγυλοποίηση 0.0016. Ωστόσο, βάλαμε 0.0017 για να συμφωνεί με τις υπόλοιπες στρογγυλοποιήσεις που κάναμε και να δίνουν άθροισμα μονάδα.

Αν έχουμε 4 πελάτες στο σύστημα, τότε δεν μπορούμε να δεχτούμε άλλους εξαιτίας της εξάντλησης της χωρητικότητας του συστήματος, επομένως, τους χάνουμε και για τον λόγο αυτό η πιθανότητα απώλειας πελάτη είναι $P_{blocking} = p_4 = 0{,}0017$.

(β) Ο κώδικας για το ερώτημα αυτό είναι:

```
clc;
clear all;
close all;

pkg load statistics
pkg load queueing

%system M/M/1/K

lambda = 5;
mu = 10;

states = [0, 1, 2, 3, 4];

for i = 1 : 4
  b(i) = lambda/i;
  d(i) = mu;
endfor

%Q is the transition matrix
Q = ctmcbd (b, d)

%p are the stationary occupancy probabilities
p = ctmc(Q)

% plot the ergodic probabilities (bar for bar chart)
figure(1);
bar(states, p, "r", 0.5);
```

```
title("Ergodic Probabilities of system states");
xlabel("states of the system");
ylabel("Probabilities");

mean_clients = 0;
for i = 1 : 5
  mean_clients = mean_clients + states(i)*p(i);
endfor

mean_clients

display("P(blocking) = ");
display (p(5));

step = 0.01;
p0 = [1, 0, 0, 0, 0];
%At time 0 the probability of the system having 0 clients is 1, thus
all others
%are 0.

lower_limit = 0.99*p;
upper_limit = 1.01*p;

for j = 1 : 5
  index = 0;
  clear Pr;
%so that Pr's size is not set already, this caused some errors
previously
  for T = 0 : step : 2
    index = index + 1;
    Pr_help = ctmc(Q, T, p0);
    Pr(index) = Pr_help(j);
    if Pr_help(j) > lower_limit(j) && Pr_help(j) < upper_limit(j)
      break;
    endif
  endfor
  t = 0 : step : T;
  figure(1+j)
  hold on

  plot(t, Pr, 'm', "linewidth", 1.2);

  title({"Transient probability of state",j-1});
  xlabel("time in seconds");
  ylabel({"Probability of state",j-1});

  line([0 T],[p(j) p(j)],"linestyle","--
","color","k","linewidth",0.3);
  hold off
endfor
```

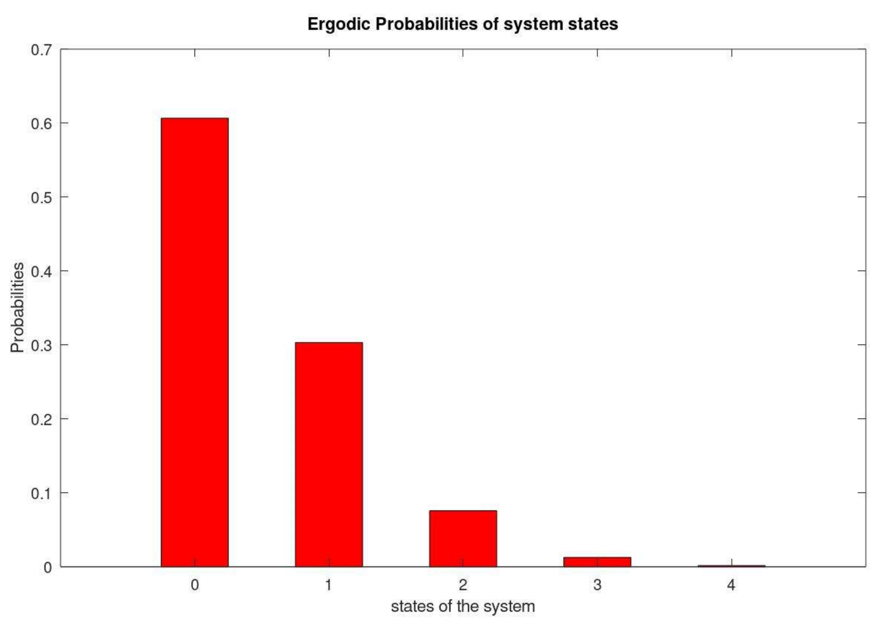
i) Στο μοντέλο αναμονής Γεννήσεων – Θανάτων που προσομοιώνουμε σε αυτό το ερώτημα έχουμε ότι στη μήτρα των ρυθμών μεταβάσεων Q = {qi,j} υπάρχουν μόνο τα στοιχεία της μορφής:

$$\begin{cases} q_{n,n-1} = \mu_n \\ q_{n,n+1} = \lambda_n \\ q_{n,n} = -(\lambda_n + \mu_n) \end{cases}$$

Αφού έχουμε μόνο γεννήσεις και θανάτους. Στον πίνακα Q αναμένουμε, λοιπόν, να υπάρχει η κύρια διαγώνιος, τα στοιχεία που είναι ακριβώς αριστερά (αντιπροσωπεύουν τους ρυθμούς θανάτου) και ακριβώς δεξιά (αντιπροσωπεύουν τους ρυθμούς γέννησης) από αυτήν και όλα τα υπόλοιπα να είναι μηδενικά.

```
Q =

   -5.0000    5.0000         0         0         0
   10.0000  -12.5000    2.5000         0         0
         0   10.0000  -11.6667    1.6667         0
         0         0   10.0000  -11.2500    1.2500
         0         0         0   10.0000  -10.0000
```

ii)




```
p =

   6.0664e-01   3.0332e-01   7.5829e-02   1.2638e-02   1.5798e-03
```

Το octave μας δίνει δηλαδή:

$$p_0 = 0.60664$$
$$p_1 = 0.30332$$
$$p_2 = 0.075829$$
$$p_3 = 0.012638$$
$$p_4 = 0.0015798$$

Οι οποίες είναι ίδιες με αυτές που βρήκαμε στο ερώτημα (α) (οι διαφορές τους οφείλονται στις στρογγυλοποιήσεις).

iii) Για να βρούμε το μέσο αριθμό πελατών στο σύστημα όταν αυτό βρίσκεται σε κατάσταση ισορροπίας χρησιμοποιήσαμε τον τύπο:

$$E(X) = \sum_{k\geq 0} k p_X(k)$$

Ο οποίος στη δική μας περίπτωση είναι:

$$E(n(t)) = \sum_{k=0}^{4} k p_k$$

Και παίρνουμε το παρακάτω αποτέλεσμα:

```
mean_clients = 0.4992
```

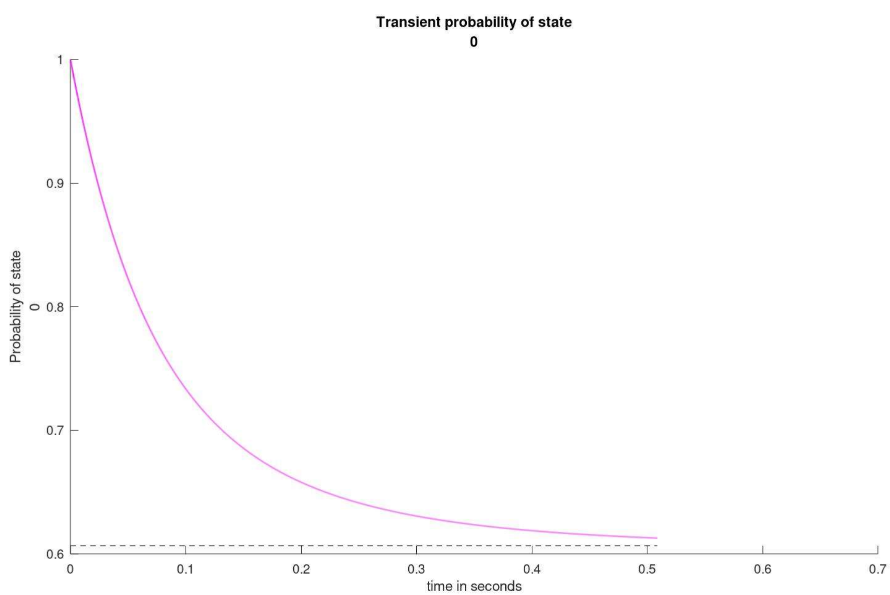
iv) Ως blocking probability τυπώνουμε το p4.

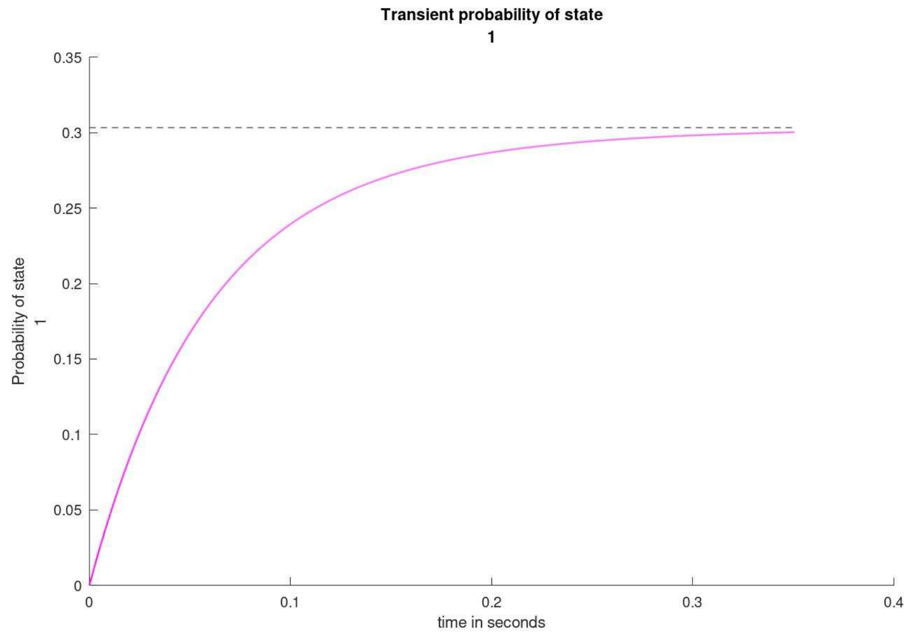
```
P(blocking) =
1.5798e-03
```

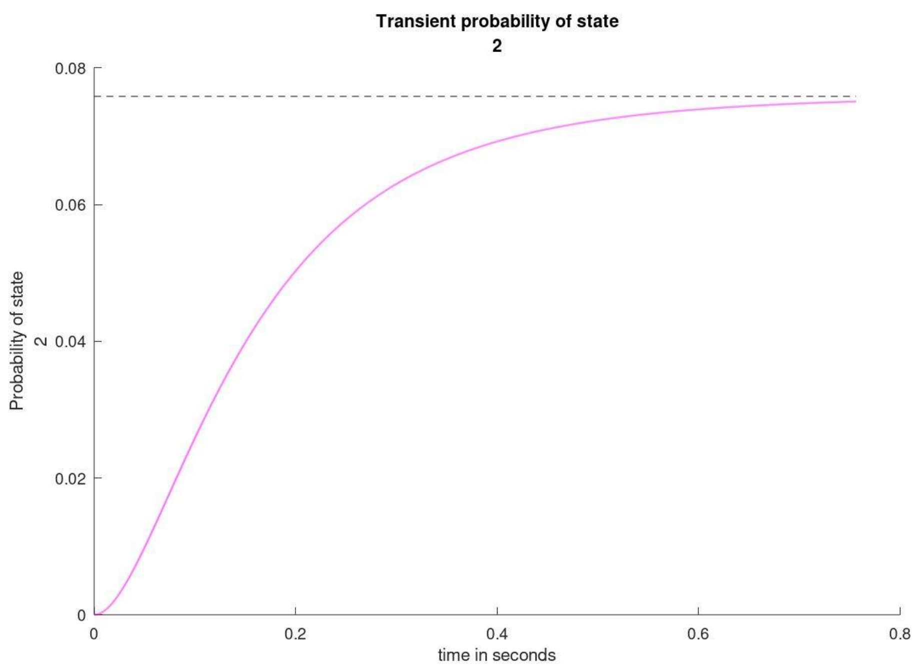
v) Η εκφώνηση αναφέρει «… μέχρι οι πιθανότητες να έχουν απόσταση μικρότερη του 1% από τις εργοδικές πιθανότητες του ερωτήματος (2)». Ερμηνεύουμε αυτή την απαίτηση ως κάθε πιθανότητα να έχει απόσταση μικρότερη του 1% της εργοδικής πιθανότητας που της αντιστοιχεί από αυτή την εργοδική πιθανότητα. Να ικανοποιείται, δηλαδή, η ανισότητα:

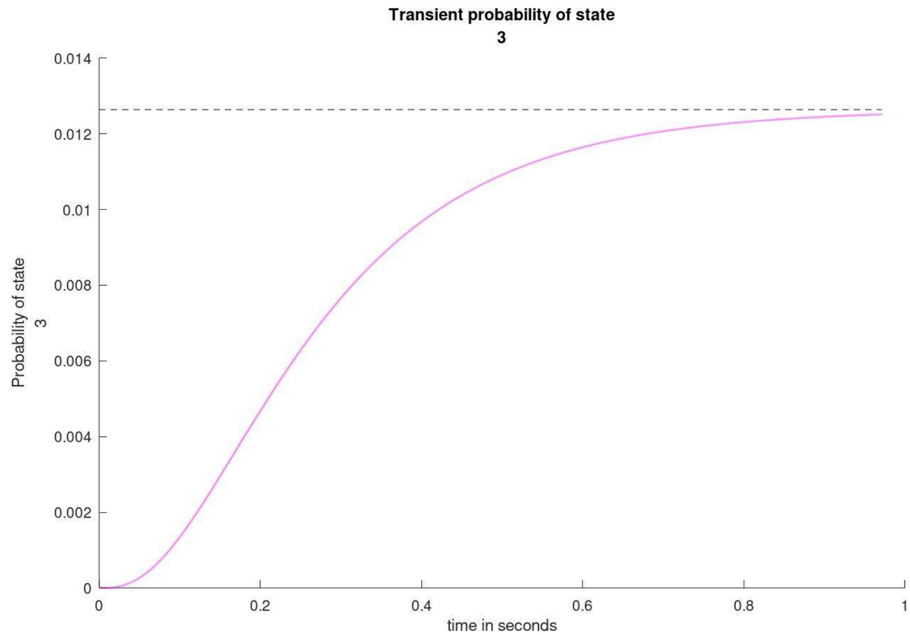
$$0.99p_i \leq p_i(t) \leq 1.01p_i, i = 0, 1, 2, 3, 4$$

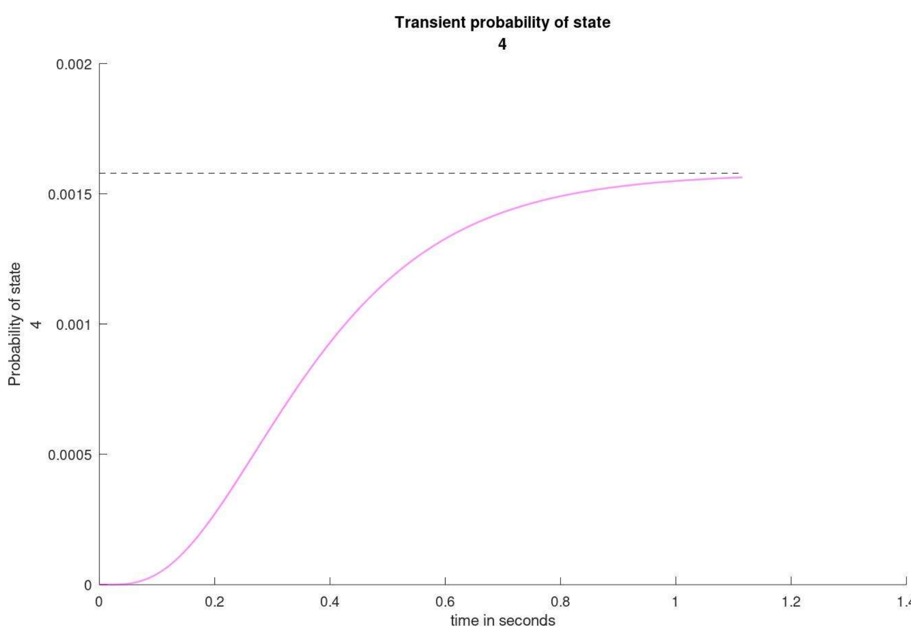
Στα διαγράμματα εμφανίζουμε με μαύρη διακεκομμένη γραμμή την αντίστοιχη εργοδική πιθανότητα.




Transient probability of state
1
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Probability of state
1
0
0.1
0.2
0.3
0.4
time in seconds


Transient probability of state
2
0.08
0.06
0.04
0.02
0
Probability of state
2
0
0.2
0.4
0.6
0.8
time in seconds





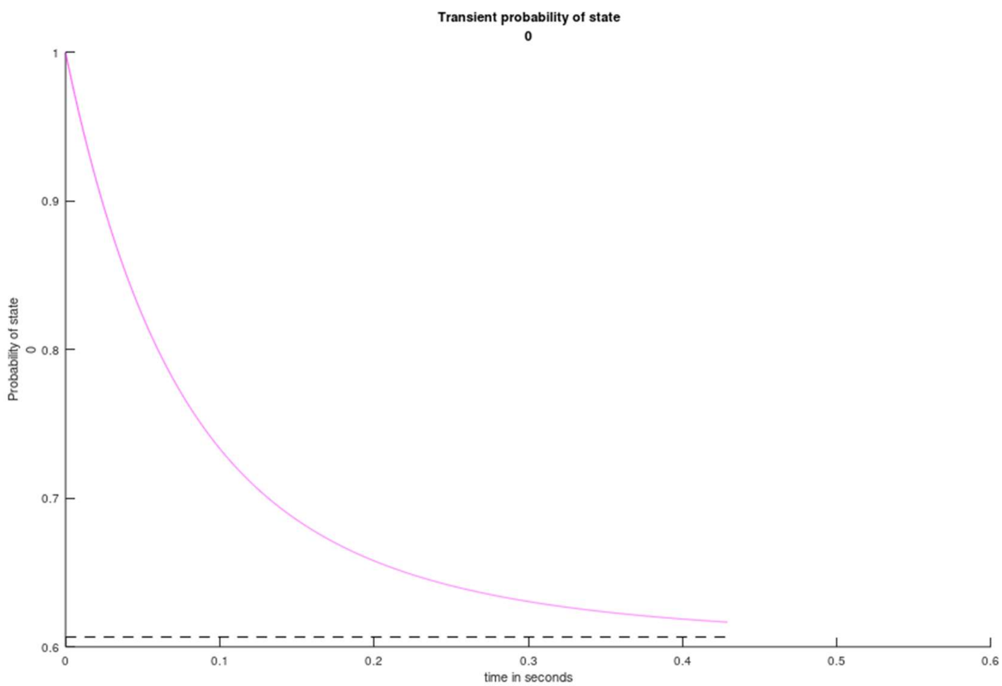
Ως αρχική κατάσταση επιλέχθηκε η:

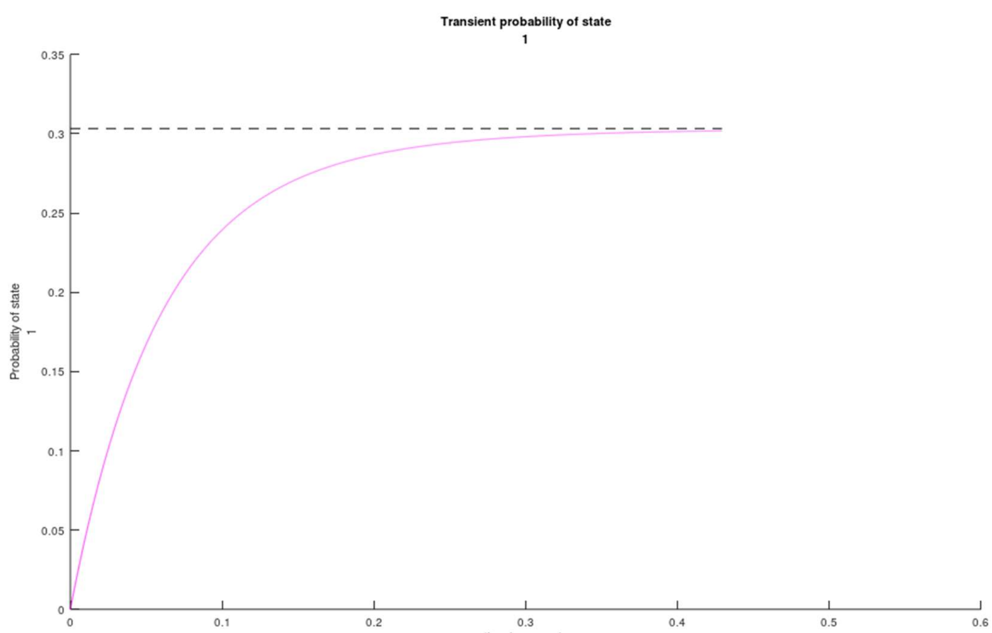
$$p_0(0) = 1$$
$$p_1(0) = 0$$
$$p_2(0) = 0$$
$$p_3(0) = 0$$
$$p_4(0) = 0$$

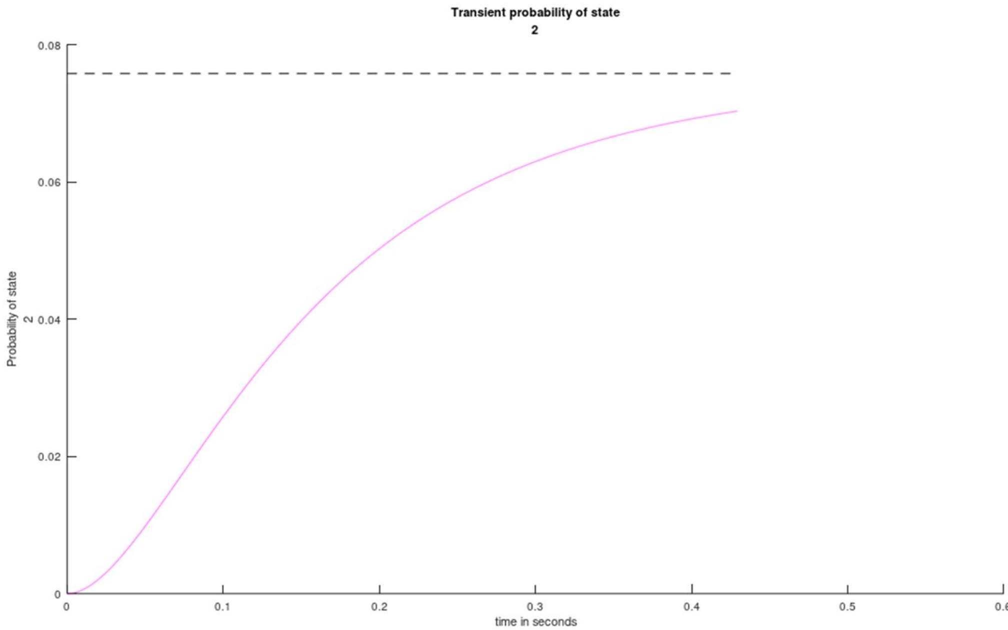
το οποίο θεωρούμε δεδομένο από την εκφώνηση. Στα διαγράμματα βλέπουμε, επίσης, ότι οι διαφορετικές πιθανότητες χρειάστηκαν και διαφορετικό χρόνο για να συγκλίνουν στην εργοδική τους τιμή.

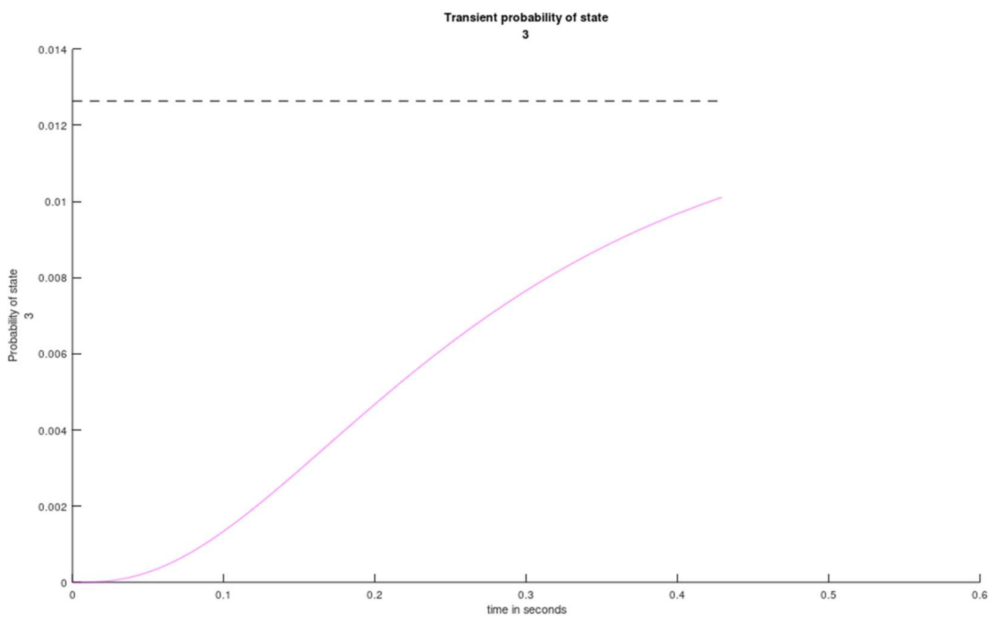
Στο ‘demo2.m’ (δοσμένος κώδικας) για ένα παρόμοιο ερώτημα σύγκλισης θεωρείται ότι πρέπει κάθε πιθανότητα να διαφέρει από την αντίστοιχη εργοδική κατά 0.01. Δείχνουμε τα αποτελέσματα και με αυτή τη λογική:

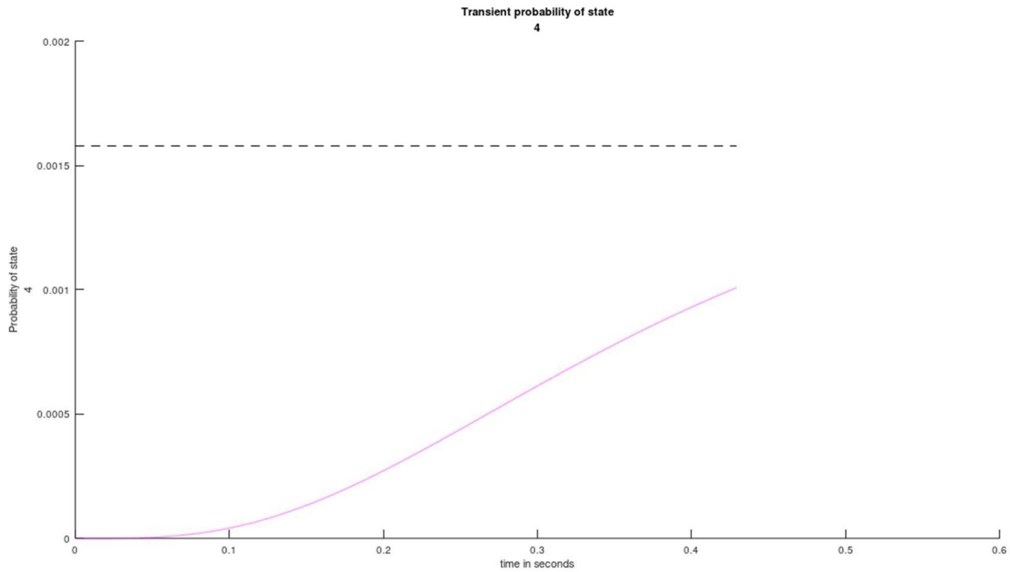
Χρησιμοποιήσαμε τον ίδιο κώδικα με πριν με μερικές αλλαγές στο τέλος (αρχείο ‘mm1k_analysis_version2.m’), ο κώδικας που αλλάξαμε υπάρχει μετά τις γραφικές παραστάσεις.






Transient probability of state
2
Probability of state
2
0.08
0.06
0.04
0.02
0
0
0.1
0.2
0.3
0.4
0.5
0.6
time in seconds


Transient probability of state
3
Probability of state
3
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
0.1
0.2
0.3
0.4
0.5
0.6
time in seconds




```
step = 0.0001;
p0 = [1, 0, 0, 0, 0];
%At time 0 the probability of the system having 0 clients is 1, thus
all others
%are 0.

lower_limit = p - 0.01;
upper_limit = p + 0.01;

for j = 1 : 5
  index = 0;
  clear Pr;
%so that Pr's size is not set already, this caused some errors
previously
  for T = 0 : step : 2
    index = index + 1;
    Pr_help = ctmc(Q, T, p0);
    Pr(index) = Pr_help(j);
%convergence takes place when Pr_help and p differ by 0.01.
    if Pr_help > lower_limit & Pr_help < upper_limit
      break;
    endif
  endfor
  t = 0 : step : T;
  figure(1+j)
  hold on

  plot(t, Pr, 'm', "linewidth", 1.2);

  title({"Transient probability of state",j-1});
  xlabel("time in seconds");
  ylabel({"Probability of state",j-1});

  line([0 T],[p(j) p(j)],"linestyle","--
","color","k","linewidth",0.3);
  hold off
endfor
```