ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΣΙΤΩΝ
ΡΟΗ Δ - ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ (QUEUING SYSTEMS)

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 2ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

ΘΕΩΡΗΤΙΚΉ ΜΕΛΕΤΗ ΤΗΣ ΟΥΡΆΣ M/M/1

Σύμφωνα με την θεωρία η ουρά M/M/1 αντιστοιχίζει σε μία ουρά όπου:
Οι αφίξεις στην ουρά ακολουθούν κατανομή Poisson με παράμετρο λ πελάτες/sec. Οι εξυπηρετήσεις ακολουθούν εκθετική κατανομή με παράμετρο μ πελάτες/sec. Υπάρχει μόνο ενας εξυπηρετητής. Το σύστημα έχει άπειρη χωρητικότητα

Α) Η απαραίτητη συνθήκη ώστε η ουρά M/M/1 να είναι εργοδική είναι η ένταση φορτίου ρ (traffic intensity) του συστήματος να είναι μικροτερη της μονάδας (ρ<1). Η ένταση φορτίου είναι ουσιαστικά η πιθανότητα το σύστημα να μην είναι άδειο. Ισχύει ότι ρ=λ/μ.
Επομένως η συνθήκη Erlang που πρέπει να ικανοποιείται είναι:

$\rho = \frac{\lambda}{\mu} < 1$ Erlang (συνθήκη για οριακή ισορροπία – εργοδικότητα)

Οι εξισώσεις ισορροπίας ενός συστήματος γενικά είναι:

- $(\lambda_k + \mu_k)P_k = \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}, \; k > 1$
- $\lambda_0 P_0 = \mu_1 P_1$
- $P_0 + P_1 + \cdots + P_k + \cdots = 1$

Οι παράμετροι λn και μn είναι οι ρυθμοί γεννήσεων και θανάτων, αντίστοιχα, όταν το σύστημα βρίσκεται στην κατάσταση n. Το Pk είναι η πιθανότητα το σύστημα να βρίσκεται στην κατάσταση k.
Ισχύει ότι το μεταβατικό φαινόμενο παρέρχεται για επαναληπτικές καταστάσεις n(t)=k, έτσι ξεχνιέται η αρχική συνθήκη Pk(0) και οι Pk(t) συγκλινουν στις οριακές πιθανότητες Pk>0:

Για $t \to \infty$, $\frac{dP_k(t)}{dt} = 0$, $P_k(t) \to P_k > 0$: ***Εργοδικές Οριακές Πιθανότητες***

Παρατηρούμε ότι οι παράμετροι λn και μn είναι ανεξάρτητοι του χρόνου. Αυτό σημαίνει ότι η αλυσίδα Markov είναι ομοιογενής.

Για το απλό σύστημα M/M/1 ισχύει ότι οι ρυθμοί γεννήσεων και θανάτων είναι ανεξάρτητοι της κατάστασης:
Αρα λn=λ και μn=μ
Επομένως:

- $\lambda P_0 = \mu P_1$ ή $P_1 = \left(\frac{\lambda}{\mu}\right)P_0 = \rho P_0$
- $(\lambda + \mu)P_1 = \lambda P_0 + \mu P_2$ ή $P_2 = \rho^2 P_0$ και γενικά $P_k = \rho^k P_0, \; k > 0$
- $P_0 + P_1 + \cdots + P_k + \cdots = 1 = P_0(1 + \rho + \rho^2 + \rho^3 + \cdots)$

Ωστόσο ισχύει ότι:

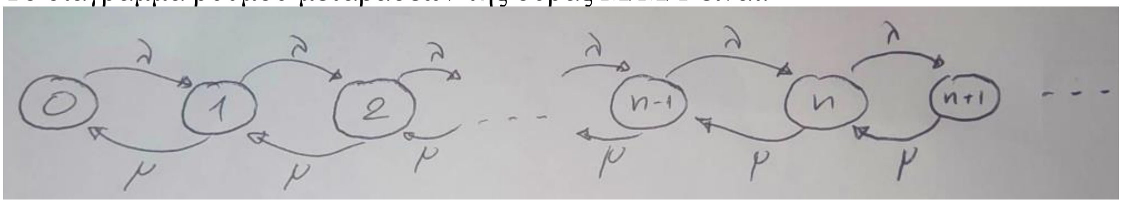
1. $\rho = \frac{\lambda}{\mu} < 1$

2. $\sum_0^n \rho^n = \frac{1}{1-\rho}$

Επομένως καταλήγουμε, επειτα από πράξεις, στο ότι οι εργοδικές πιθανότητες είναι:

$$P_0 = (1-\rho), \quad P_k = (1-\rho)\rho^k, \qquad k > 0$$

Το διάγραμμα ρυθμού μεταβάσεων της ουράς M/M/1 είναι:



Β) Ισχύει ότι

$$E[n(t)] \triangleq \sum_{k=1}^{\infty} kP_k = \frac{\rho}{1-\rho}$$

Ο μέσος χρόνος καθυστέρησης από τύπο Little είναι:

$$\mathrm{E}(T) = \frac{\mathrm{E}[n(t)]}{\gamma} = \frac{\mathrm{E}[n(t)]}{\lambda} = \frac{1/\mu}{1-\rho}$$

Ο μέσος χρόνος παραμονής ενός πελάτη στο σύστημα, όταν η ουρά αναμονής βρίσκεται σε κατάσταση ισορροπίας είναι: Ο μέσος χρόνος καθυστέρησης και ο χρόνος εξυπηρέτησης Δηλαδή T = W+s, όπου s=1/μ

Επομένως:

$$W = T - s = T - \frac{1}{\mu} = \frac{1}{\mu} \cdot \frac{\rho}{1-\rho}$$

Γ) Υπολογίζουμε την πιθανότητα Pn για n=57 για τυχαίο ρ, έστω ρ=0.5

$$Pn = (1-\rho)\rho^n = P_{57} = (1-0.5)0.5^{57} = 3.47 \cdot 10^{-18}$$

Επομένως πράγματι το σύστημα θα βρεθεί με 57 πελάτες με πολύ μικρή πιθανότητα. Παρατηρούμε ότι όσο μεγαλώνει το ρ, δηλαδη οσο μικρότερη διαφορά έχουν τα λ και μ, τόσο μεγαλώνει και η πιθανότητα να βρεθεί στην κατάσταση n=57.

<u>Σχόλιο:</u> Πολλές από τις εικόνες που χρησιμοποιήθηκαν προέρχονται από τις διαφάνειες του μαθήματος.

<u>ΑΝΑΛΥΣΗ ΟΥΡΑΣ M/M/1 ΜΕ OCTAVE</u>

Ισχύει ότι έχουμε ουρά M/M/1 όπου οι αφίξεις ακολοθούν κατανομή Poisson με λ=5 πελάτες/min.

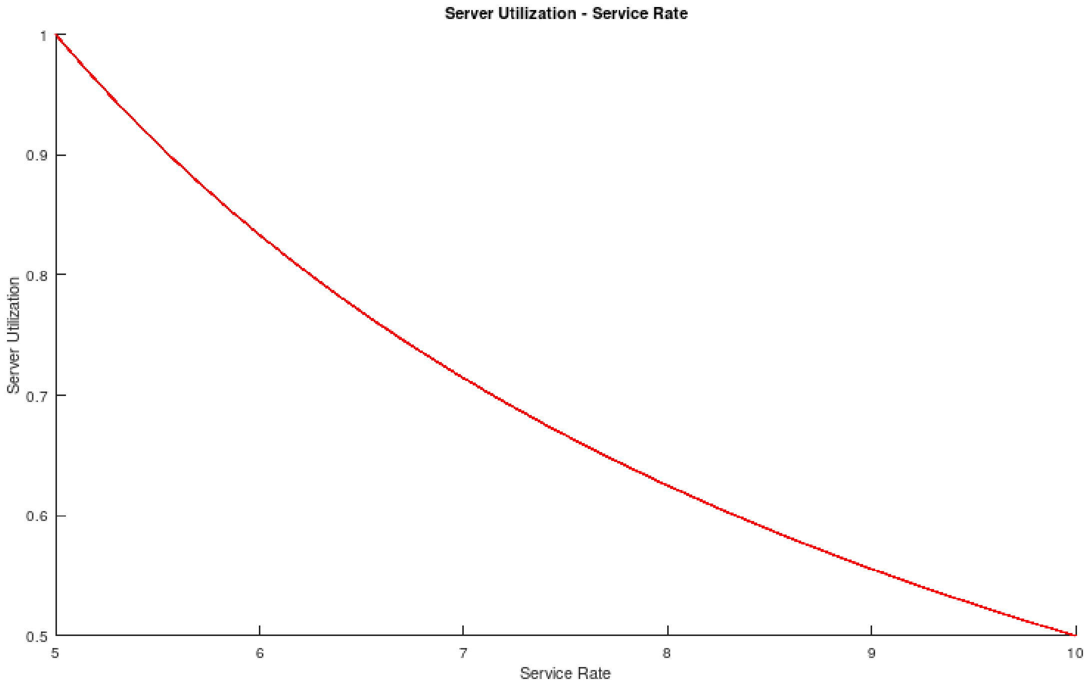
Α) Θέλουμε το σύστημα να παραμείνει εργοδικό. Αυτό σημαίνει ότι πρέπει ρ<1. Αρα:

$$\rho = \frac{\lambda}{\mu} < 1 \Rightarrow \mu > 5\ \frac{\pi\varepsilon\lambda\alpha\tau\varepsilon\varsigma}{min}$$

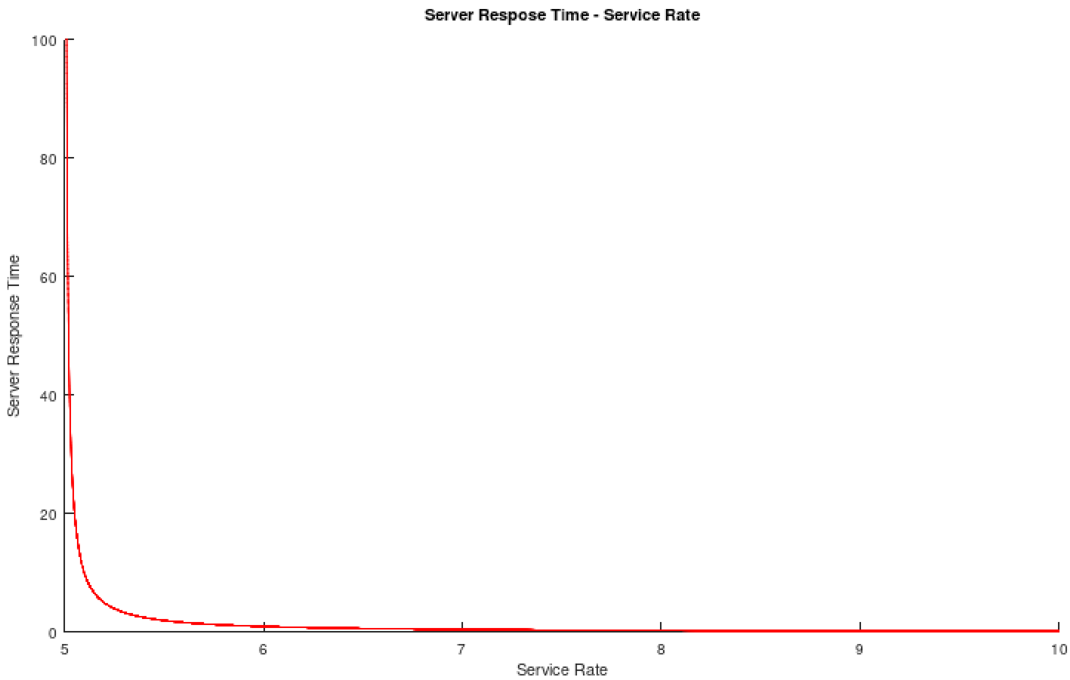
Επομένως πρέπει να ισχύει: 5<μ<10 πελάτες/min

Β) Ορίζουμε τον ρυθμό εξυπηρέτησης στο πεδίο που υπολογίσαμε στο προηγούμενο ερώτημα. Με βάση αυτό προκύπτουν τα εξής διαγράμματα:

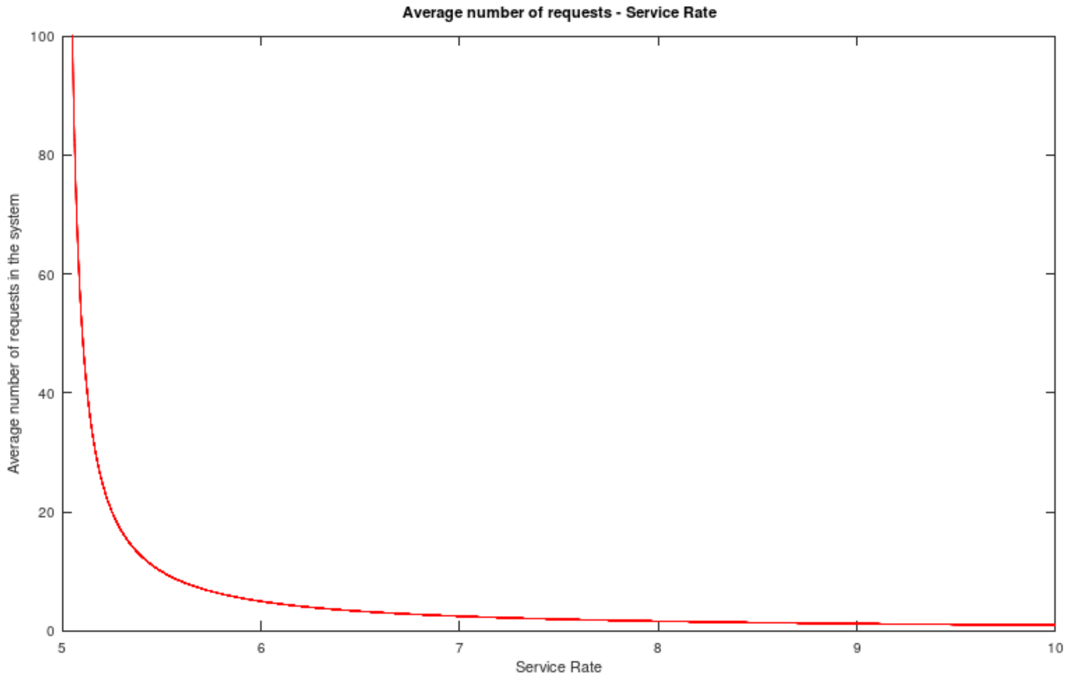
1. Βαθμός χρησιμοποίησης (utilization) ως προς το ρυθμό εξυπηρέτησης:



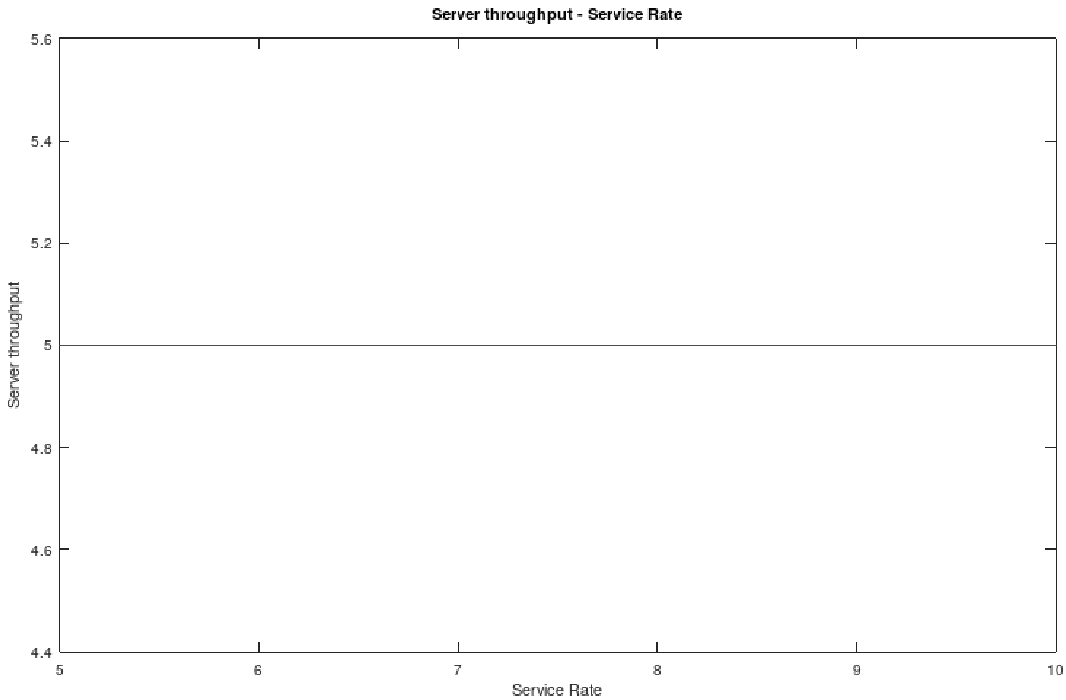
2. Μέσος χρόνος καθυστέρησης του συστήματος E(T) ως προς το ρυθμό εξυπηρέτησης:



3. Μέσος αριθμός πελατών στο σύστημα ως προς το ρυθμό εξυπηρέτησης:



4. Ρυθμαπόδοση (throughput) πελατών ως προς το ρυθμό εξυπηρέτησης:



Παρατηρήσεις:

- Χρησιμοποιήσαμε την εντολή qsmm1 όπως υποδεικνύει το documentation της.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
clc;
close all;
clear all;
pkg load queueing;

#Ερώτημα Β
lambda = 5; # λ=5
mu = 5.0001 : 0.0001 : 10; # Service rate / 5<μ<=10

[U, R, Q, X, p0] = qsmm1 (lambda, mu);
#https://octave.sourceforge.io/queueing/function/qsmm1.html
#Compute utilization, response time, average number of requests and
#throughput for a M/M/1 queue.
# U = Server Utilization
# R = Server Response time
# Q = Average number of requests in the system
# X = Server throughput. If the system is ergodic (mu > lambda),
#     we always have X = lambda
# p0 = Steady-state probability that there are no requests in the system.

#B.1
figure(1);
hold on;
plot(mu,U,"r","linewidth",1.2);
hold off;
title("Server Utilization - Service Rate");
xlabel("Service Rate");
ylabel("Server Utilization");
#B.2
figure(2);
hold on;
plot(mu,R,"r","linewidth",1.2);
axis([5 10 0 100]);
hold off;
title("Server Respose Time - Service Rate");
xlabel("Service Rate");
ylabel("Server Response Time");
#B.3
figure(3);
plot(mu,Q,"r","linewidth",1.2);
axis([5 10 0 100]);
title("Average number of requests - Service Rate");
xlabel("Service Rate");
ylabel("Average number of requests in the system");
#B.4
figure(4);
plot(mu,X,"r","linewidth",1.2);
title("Server throughput - Service Rate");
xlabel("Service Rate");
ylabel("Server throughput");
#Πάντα ίσο με λ=5 αφού λ<μ
```

Γ) Από το διάγραμμα του μέσου χρόνου καθυστέρησης ως προς τον ρυθμό εξυπηρέτησης παρατηρούμε ότι αυτός κινείται αντιστρόφως ανάλογα ως προς τον ρυθμό εξυπηρέτησης. Επομένως, εφόσον θέλουμε χαμηλό χρόνο καθυστέρησης το λογικό είναι να επιλέξουμε μεγάλη τιμή για το μ. Ωστόσο δεν είναι εφικτό να επιλεξουμε πάρα πολύ μεγαλη τιμή (δηλαδή κοντά στο 10) στον πραγματικό κόσμο λόγω μεγάλης πολυπλοκότητας και χρηματικής δαπάνης. Πρέπει, λοιπόν, να επιλέξουμε μια ενδιάμεση τιμή συνδυάζοντας μικρό χρόνο καθυστέρησης και μικρή πολυπλοκότητα και χρηματική δαπάνη. Επομένως τιμές γύρω από το μ=7.5 είναι οι καταλληλότερες.

Δ) Σε μία ουρά M/M/1 όπου μ>λ το throughput ισούται με λ για οποιαδήποτε τιμή του μ. Παρατηρούμε αυτό το συμπέρασμα και από το 4ο διάγραμμα του Β) ερωτήματος. Επιπλέον σύμφωνα με την θεωρία ισχύει ότι:

Διεκπεραίωση πελατών – Ρυθμαπόδοση (Throughput)

γ πελάτες/sec

Σε περίπτωση 1 ουράς, 1 εξυπηρετητή:

$$\gamma = \lambda(1 - \mathrm{P}\{\text{blocking}\}) \leq \lambda, \qquad \gamma < \mu$$

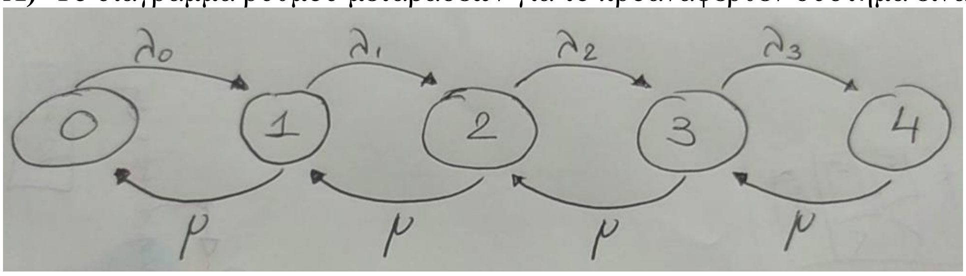
(απο διαφάνειες μαθήματος)

Στην περίπτωση όπου μ>λ ισχύει ότι P[blocking]=0 καθώς δεν έχουμε απόρριψη πελατών
Επομένως ισχύει ότι γ=5.
Διαισθητικά: εφόσον η ουρά είναι σταθερή και ο ρυθμός που οι πελάτες φευγουν από την ουρα είναι μεγαλύτερος από αυτόν που έρχονται ο ρυθμός που εξυπηρετούνται θα είναι ίσος με τον ρυθμό που έρχονται.

ΔΙΑΔΙΚΑΣΙΑ ΓΕΝΝΗΣΕΩΝ ΘΑΝΑΤΩΝ (birth-death process): ΕΦΑΡΜΟΓΗ ΣΕ ΣΥΣΤΗΜΑ Μ/Μ/1/Κ

Πλέον έχουμε ένα σύστημα M/M/1/4, δηλαδή με 1 εξυπηρετητή και μέγιστη χωρητικότητα 4 πελάτες.
Αφίξεις: Poisson με $\lambda_i = \frac{\lambda}{i+1}$
Εξυπηρετήσεις: Εκθετική με $\mu_i = \mu$
Επίσης ισχύει ότι λ = 5 πελάτες/sec και μ = 10 πελάτες/sec
Αρχική κατάσταση: 0 πελάτες στο σύστημα

Α) Το διάγραμμα ρυθμού μεταβάσεων για το προαναφερθέν σύστημα είναι το εξής:



Οι εξισώσεις ισορροπίας που προκύπτουν είναι:

$$\lambda_i P_i = \mu P_{i+1}, \ 0 \leq i \leq 3$$

$P_0 + P_1 + P_2 + P_3 + P_4 = 1$ (Συνθήκη Κανονικοποίησης)

Επίσης ισχύει ότι:
$\rho_i = \frac{\lambda_i}{\mu}$ (Συνθήκη Erlang)
Επομένως υπολογίζουμε τις εργοδικές πιθανότητες:
$P_0 \cdot \left(1 + \frac{P_1}{P_0} + \frac{P_2}{P_0} + \frac{P_3}{P_0} + \frac{P_4}{P_0}\right) = 1 \Rightarrow P_0 \cdot \left(1 + \frac{\lambda_0}{\mu} + \frac{\lambda_1 \cdot \lambda_0}{\mu^2} + \frac{\lambda_2 \cdot \lambda_1 \cdot \lambda_0}{\mu^3} + \frac{\lambda_3 \cdot \lambda_2 \cdot \lambda_1 \cdot \lambda_0}{\mu^4}\right) = 1 \Rightarrow P_0 = 0.607$
Ετσι:

$$P_1 = 0.304$$
$$P_2 = 0.076$$
$$P_3 = 0.013$$
$$P_4 = 0.0016$$

Ισχύει ότι το P4 είναι η πιθανότητα το σύστημα να γεμίσει. Αρα είναι ίση με την πιθανότητα απώλειας πελάτη, δηλαδή ενας επόμενος πελάτης να μην μπορεί να εξυπηρετηθεί. Επομένως P[blocking] = P4=0.0016

B)

i)Στην επόμενη εικόνα φαίνεται η μήτρα ρυθμού μεταβάσεων, δηλαδή ο πίνακας που περιγράφει τους ρυθμούς μετάβασης ανάμεσα στις καταστάσεις του συστήματος:

```
transition_matrix =

   -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
```

Τα στοιχεία του πίνακα που μας ενδιαφέρουν είναι όλα τα υπόλοιπα εκτός των στοιχείων της κυρίας διαγωνίου. Αυτά συνεισφέρουν μόνο στον μηδενισμό κάθε σειράς και πουθενά αλλού. Επίσης τα στοιχεία όπου ισχύει ότι $|i-j| \neq 1$ είναι μηδενικά καθώς οι μεταβάσεις συμβαίνουν μόνο μεταξύ γειτονικών καταστάσεων. Ετσι τελικά μας ενδιαφέρουν μόνο τα στοιχεία των 2 δευτερευόντων διαγωνίων όπου $|i-j| = 1$.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

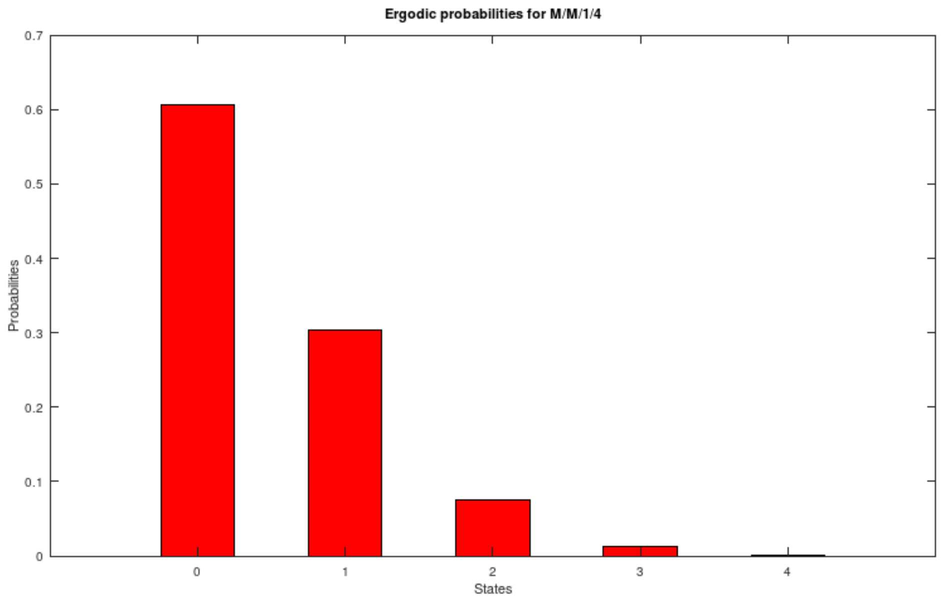
```
#Ερώτημα B
#B.i
lambda = 5;
mu = 10;
states = [0, 1, 2, 3, 4]; % system with capacity 4 states
% the initial state of the system. The system is initially empty.
initial_state = [1, 0, 0, 0, 0];

% define the birth and death rates between the states of the system.
births_B = [lambda, lambda/2, lambda/3, lambda/4];
deaths_D = [mu, mu, mu, mu];

% get the transition matrix of the birth-death process
transition_matrix = ctmcbd(births_B, deaths_D)
#https://octave.sourceforge.io/queueing/function/ctmcbd.html
```

ii) Υπολογίζουμε τις εργοδικές πιθανότητες των καταστάσεων του συστήματος χρησιμοποιώντας την εντολή ctmc του πακέτου queueing του Octave. Το αποτέλεσμα που προκύπτει είναι το εξής:

```
Probabilities:
0.6066
0.3033
0.075829
0.012638
1.5798e-03
```



Παρατηρούμε ότι οι τιμές που υπολογίσαμε με το Octave ταιριάζουν με αυτές που βρήκαμε στο θεωρητικό μέρος, στο ερώτημα Α, με αμελητέες αποκλίσεις λόγω στρογγυλοποίησης.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#B.ii
% get the ergodic probabilities of the system
P = ctmc(transition_matrix);
#https://octave.sourceforge.io/queueing/function/ctmc.html
#Compute stationary or transient state occupancy probabilities for a
# continuous-time Markov chain.
printf("Probabilities: \n")
for i=1:5
 display(P(i));
endfor
% plot the ergodic probabilities (bar for bar chart)
figure(1);
bar(states, P, "r", 0.5);
title("Ergodic probabilities for M/M/1/4");
xlabel("States");
ylabel("Probabilities");
```

iii) Ο μέσος αριθμός πελατών στο σύστημα όταν βρίσκεται σε κατάσταση ισορροπίας είναι:

```
mo_of_clients = 0.4992
```

Αυτός υπολογίζεται από τον τύπο:

$$\mathrm{E}(k) = \sum_{k=1}^{N} k P_k$$

(απο διαφάνειες μαθήματος)

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#B.iii
mo_of_clients = 0;
for i = 1 : 1 : 5
  mo_of_clients = mo_of_clients + P(i)*(i-1);
endfor
display(mo_of_clients);
```

iv) Η πιθανότητα απόρριψης πελάτη (blocking probability) από το σύστημα, όταν εκείνο βρίσκεται σε κατάσταση ισορροπίας είναι ίση με την πιθανότητα το σύστημα να βρεθεί στην κατάσταση 4.
Μέσω του Octave βρήκαμε ότι:

```
P[blocking] =
1.5798e-03
```

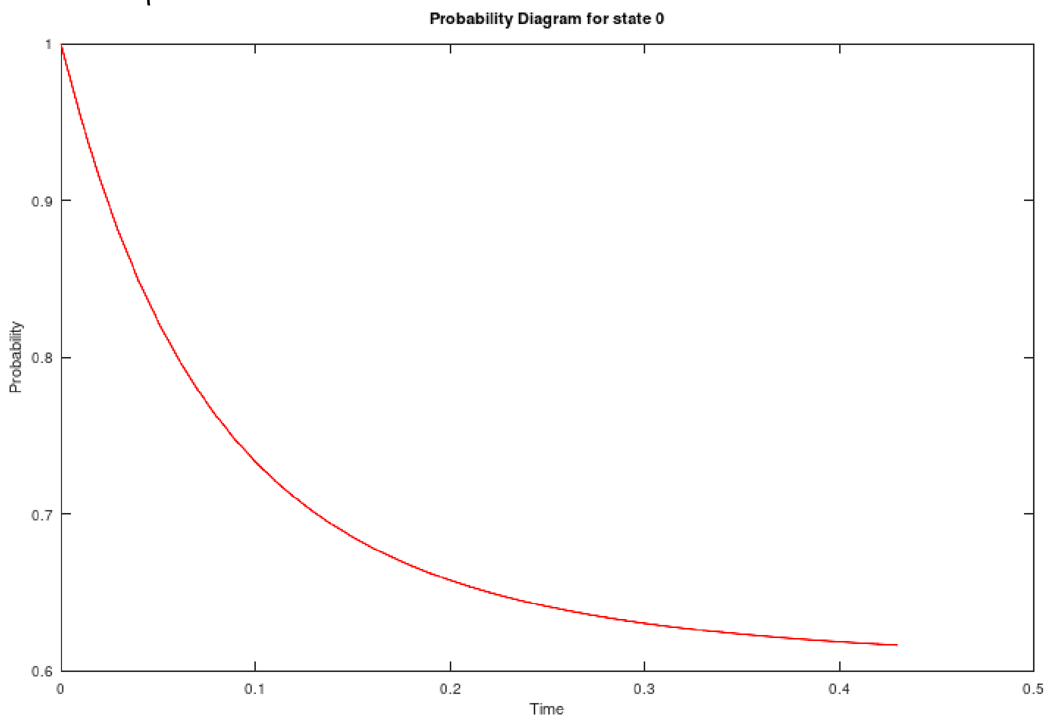
Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#B.iv
display("P[blocking] = ");
display(P(5)); # P(4) όμως το index ξεκιναει απο 1
```

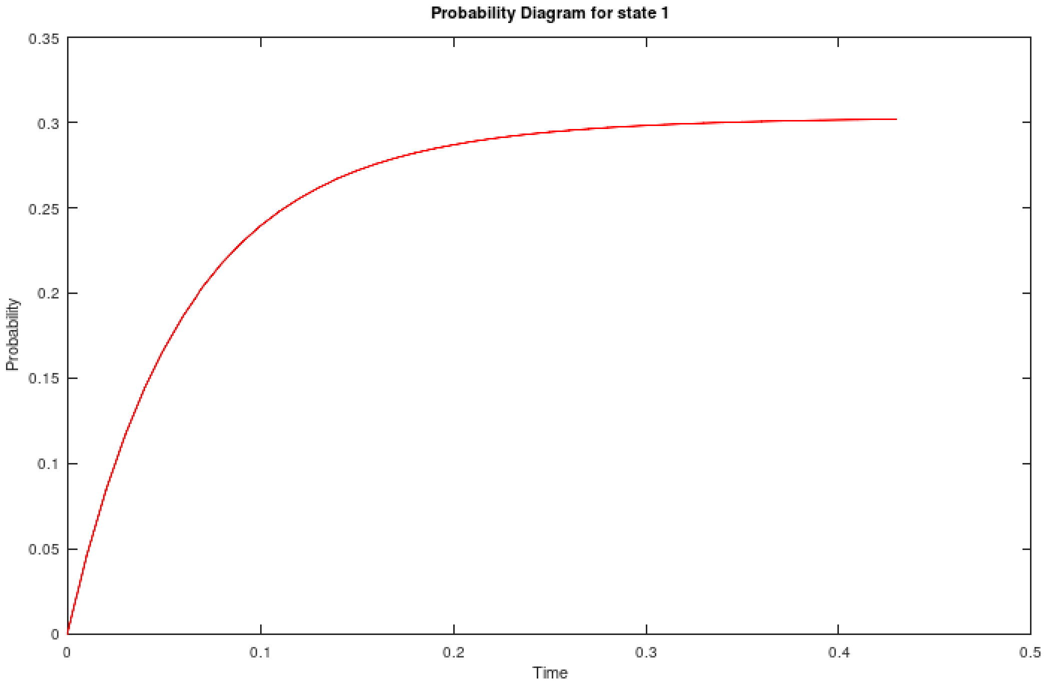
v) Χρησιμοποιώντας το Octave δημιουργούμε τα διαγράμματα των πιθανοτήτων των καταστάσεων του συστήματος σαν συναρτήσεις του χρόνου από την αρχική κατάσταση μέχρι οι πιθανότητες να έχουν απόσταση μικρότερη του 1% από τις εργοδικές πιθανότητες.

Ουσιαστικά αποτυπώνουμε την εξέλιξη των πιθανοτήτων καθώς αυτές προσσεγγίζουν τις εργοδικές.

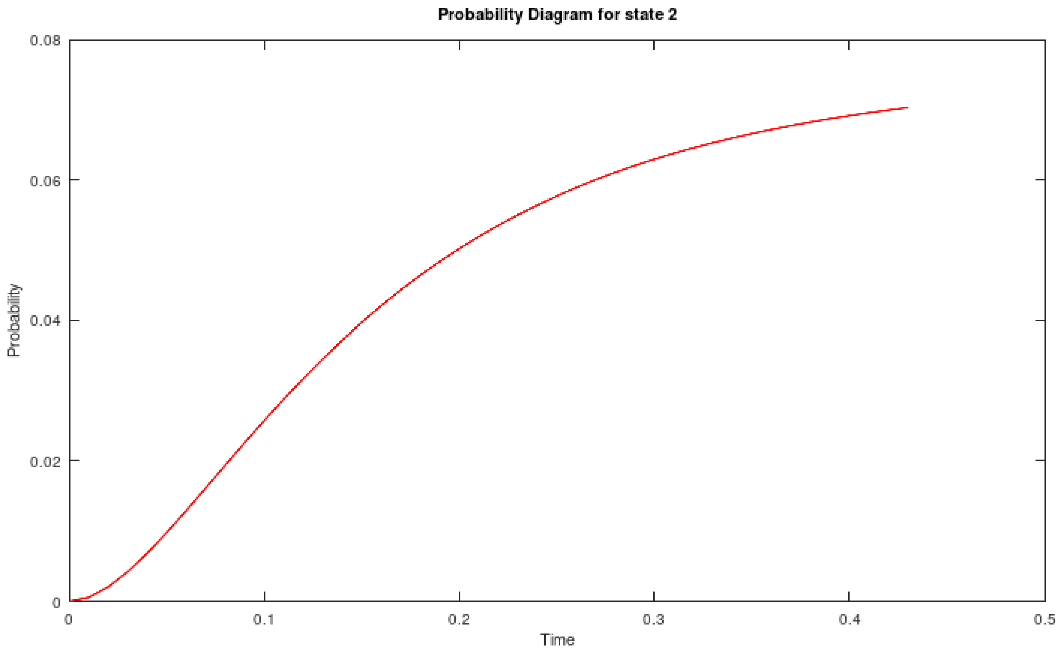
Κατάσταση 0:



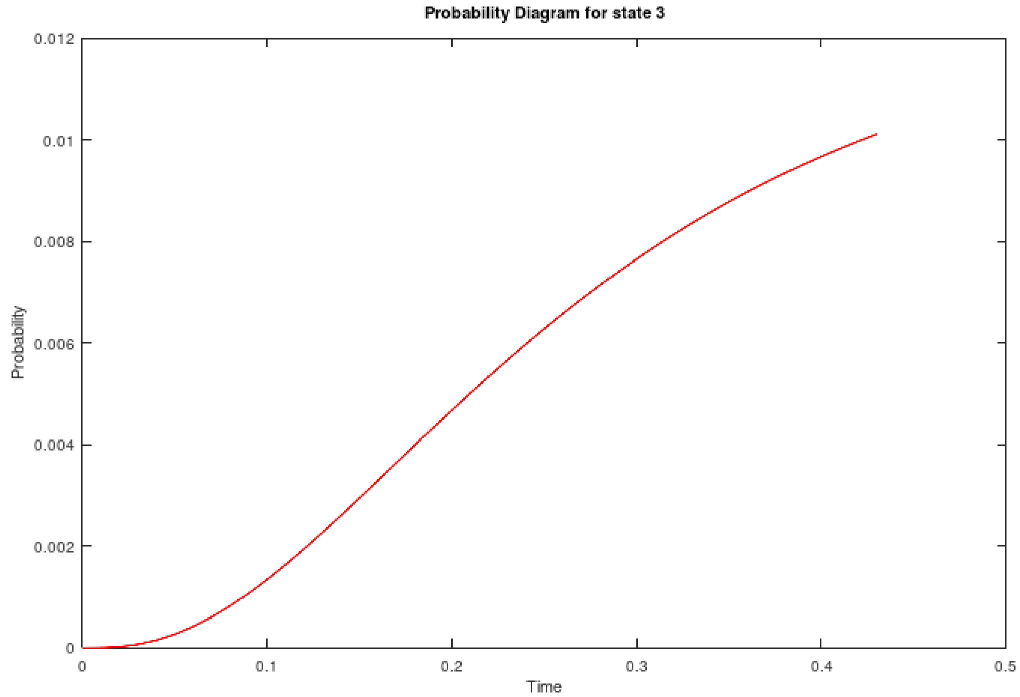
Κατάσταση 1:



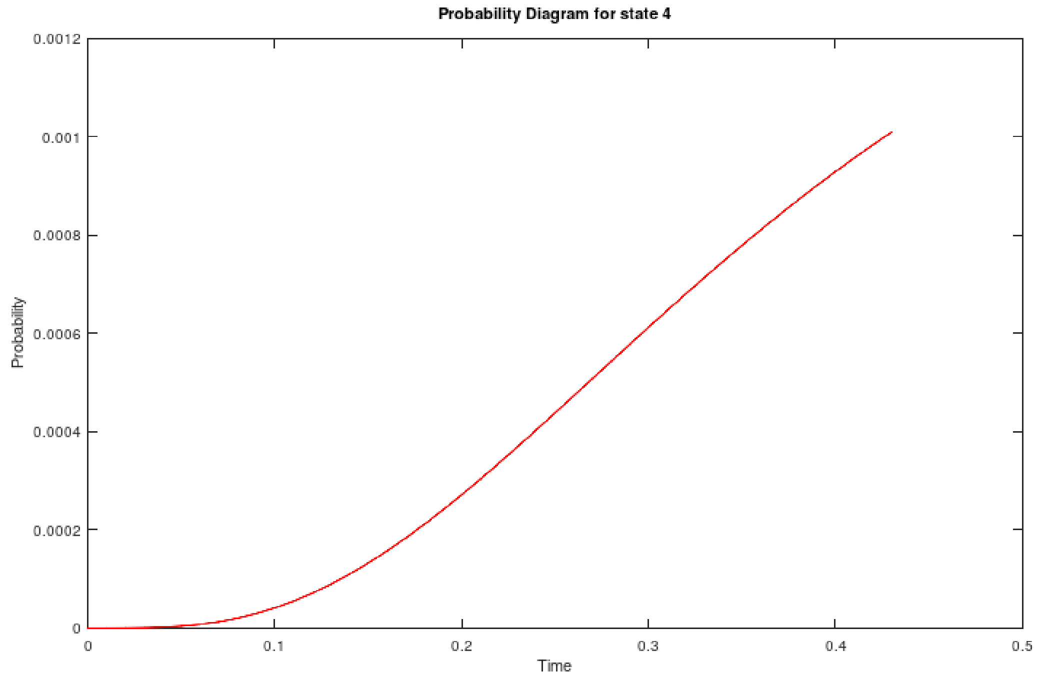
Κατάσταση 2:



Κατάσταση 3:



Κατάσταση 4:



Παρατηρήσεις:

1. Η πιθανότητα της κατάστασης 0 ξεκινάει από P=1 γιατί αυτή είναι η αρχική κατάσταση σύμφωνα με την εκφώνηση. Για τον ίδιο λόγο οι υπόλοιπες καταστάσεις ξεκινάνε από P=0.

Ο κώδικας που χρησιμοποιήσαμε για το ερώτημα αυτό είναι ο εξής:

```
#B.v
#for loop that plots 5 different diagrams
for i = 1:1:5
  index = 0;
  #creating
  for T = 0 : 0.01 :50
    index = index + 1; # keep track of iterations
    P_i= ctmc(transition_matrix, T, initial_state);
    #vector of steady-state probabilities of the Markov chain at a particular time T
    Prob_i(index) = P_i(i);
    if P_i - P < 0.01
      break;
    endif
  endfor
  #plotting
  t = 0 : 0.01 : T; # μεχρι να τελειώσει
  figure(i+1);
  plot(t, Prob_i, "r", "linewidth", 1.3);
  title(sprintf('Probability Diagram for state %d',i-1));
  xlabel("Time");
  ylabel("Probability");
endfor
```

Σχόλια:

- Χρησιμοποιήθηκε το αρχείο demo2.m που μας είχε δωθεί, στο οποίο έγιναν πολλές αλλαγές
- Τα αρχεία με τους κώδικες για κάθε άσκηση βρίσκονται στο zip αρχείο το οποίο παραδόθηκε.