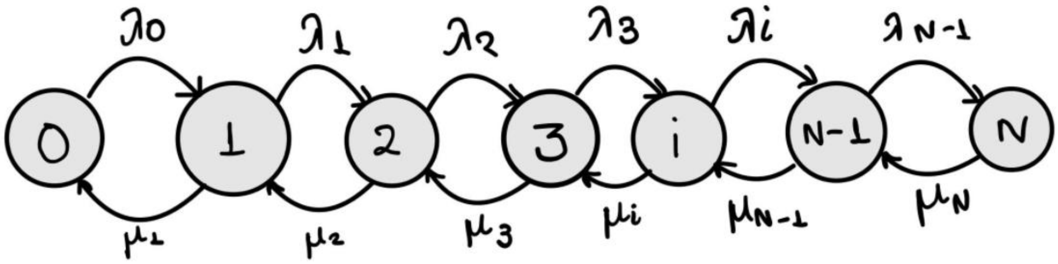
Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Ακ. έτος 2020-2021, 6ο Εξάμηνο: Συστήματα Αναμονής

Παναγιώτα-Μικαέλα Ξυλιά
AM:03118859

<u>2η Ομάδα Ασκήσεων</u>

Θεωρητική μελέτη της ουράς M/M/1

(α) Απαραίτητη συνθήκη για να είναι η ουρά M/M/1 εργοδική είναι να ισχυεί η συνθήκη $\rho < 1$, δηλαδή $\lambda < \mu$, όπου λ οι αφίξεις πελατών ανα sec και μ οι εξυπηρετήσεις πελατών ανα sec.



Ισχύει ότι $\lambda_0=\lambda_1=\lambda_2=\ldots=\lambda_N=\lambda$

$\mu_0=\mu_1=\mu_2=\ldots=\mu_N=\mu$

$\lambda P_{i-1}=\mu P_{i,}$ $(\lambda_\kappa+\mu_\kappa)P\kappa= \lambda_{\kappa-1}P_{\kappa-1} + \mu_{\kappa+1}P_{\kappa+1}$

λP0=μP1 => P1=pP0

(λ+μ)P1 =λP0 +μ P1 => P2 =p^2 P0

Άρα Pk = p^k P0

P0 $\frac{1}{1-p}$ =1 => P0 =1-p => $\text{Pk} = \text{p}^\text{k} (1-\text{p})$

και P(n(t))>0)=1-P0 =p

(β) Έχουμε ως δεδομένο ότι ισχύει η σχέση :

$$E[n(t)] = \sum_{k=1}^{+\infty} kPk = \frac{\rho}{1-\rho}, \quad \rho = \frac{\lambda}{\mu}$$

Επομένως ο μέσος χρόνος καθυστέρησης στην κατάσταση ισορροπίας (δηλαδή όταν ρ=$\frac{\lambda}{\mu}$) θα είναι:

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

(γ) Η πιθανότητα να έχω 57 πελάτες στο συστημά μου είναι: $P_{57} = p^{57}(1\text{-}p)$

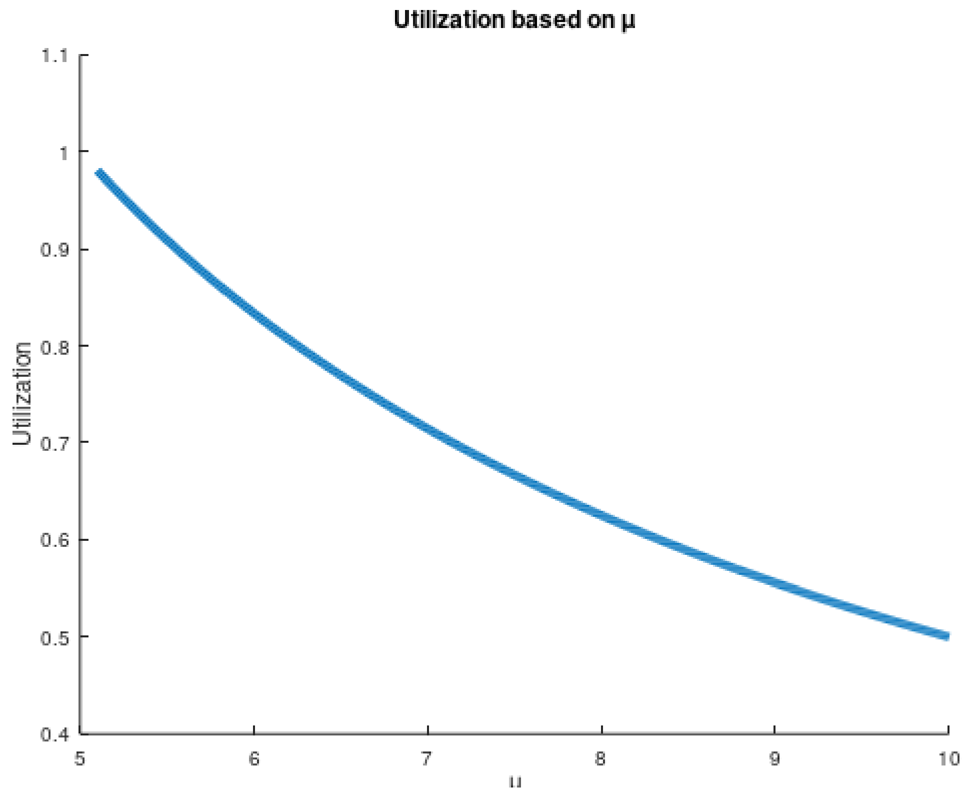
Ανάλυση ουράς M/M/1 σε Octave

(α) λ = 5 πελ/min

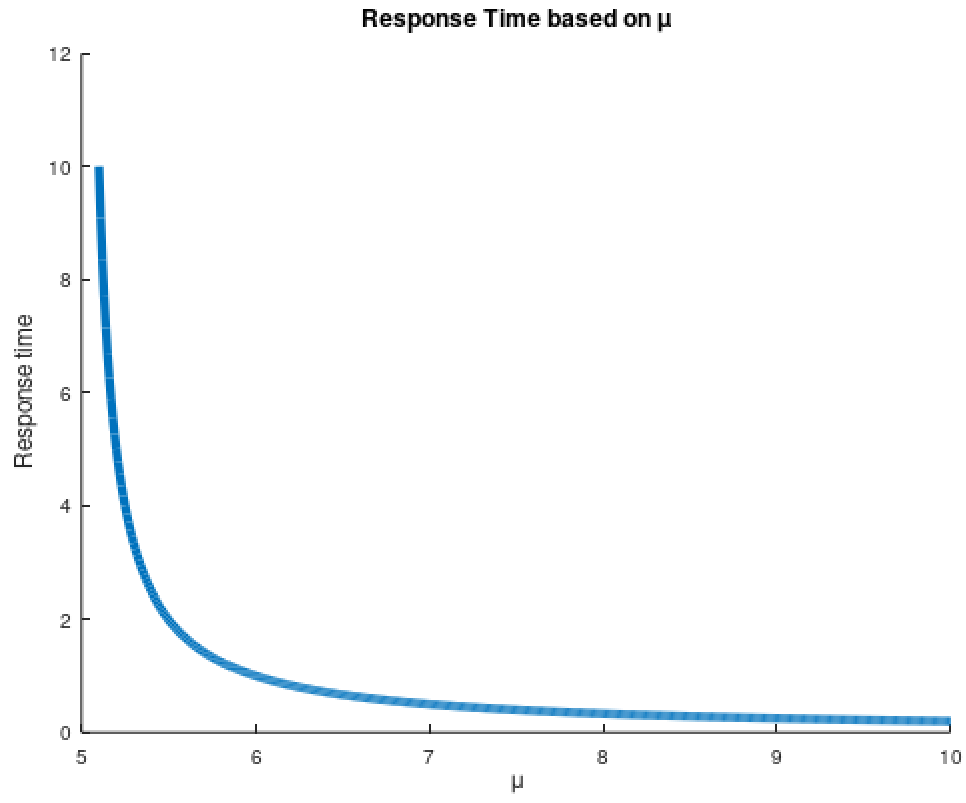
μ= 0 εώς 10 πελ/min

Για να είναι το σύστημα εργοδικό αρκεί p=λ/μ >1 =>λ>μ => 5 < μ <10

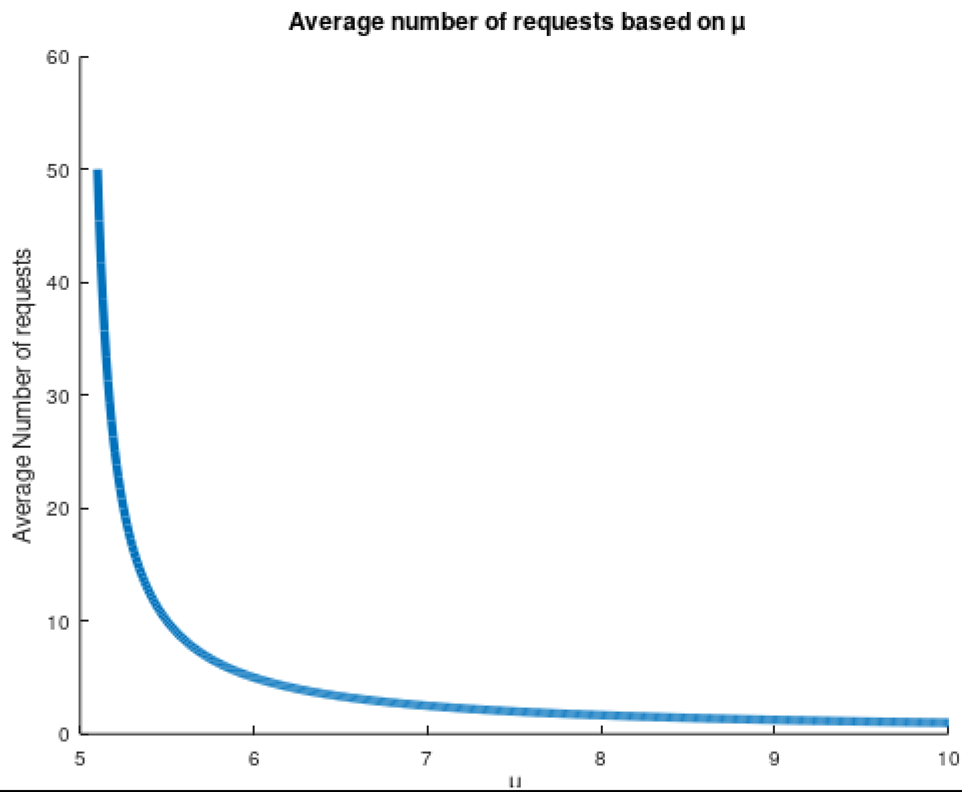
(β) Βαθμός χρησιμοποίησης (utilization) ως προς το ρυθμό εξυπηρέτησης:



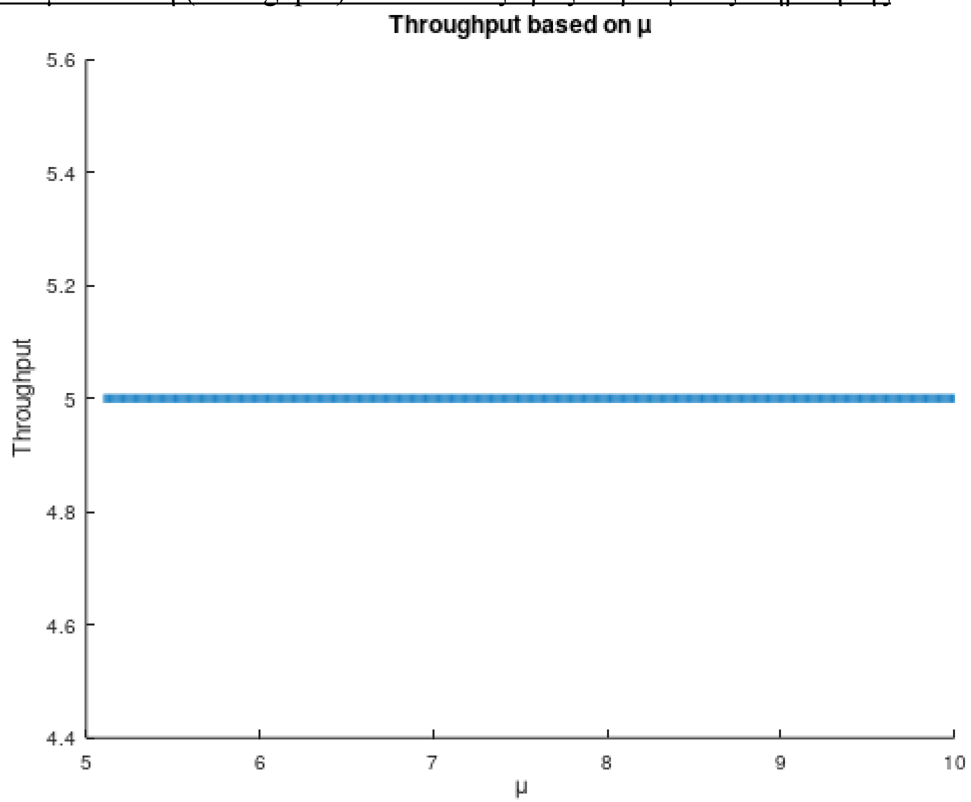
Μέσος χρόνος καθυστέρησης του συστήματος E(T) ως προς το ρυθμό εξυπηρέτησης:



Μέσος αριθμός πελατών στο σύστημα ως προς το ρυθμό εξυπηρέτησης:



Ρυθμαπόδοση (throughput) πελατών ως προς το ρυθμό εξυπηρέτησης:



(γ) Παρατηρούμε ότι το διάγραμμα αρχίζει να σταθεροποιείται για την τιμή μ=8 πελ/ min, οπότε θα επέλεγα αυτή την τιμή, καθώς μεγαλύτερος αριθμός θα έχει ελάχιστη διαφορά στην απόδοση.
(δ) Παρατηρούμε ότι η ρυθμαπόδοση (throughput) παραμένει σταθερή και ίση με 5. Γενικότερα ισχύει ο τύπος **γ= λ(1-P(loss))**, όμως για ουρά M/M/1 έχουμε P(loss)=0 και άρα γ=λ=5.

Κώδικας που χρησιμοποιήθηκε:

```
pkg load statistics
pkg load queueing

clc;
clear all;
close all;
# M/M/1
# (b)
lambda = 5
U=[0,500]; #utiliaztion
R=[0,500]; #server responce time
Q=[0,500]; #average number of requests
X=[0,500]; #server throughput

mu = [5.1:0.01:10];
display(mu)
for i=1:columns(mu)
    [U(i),R(i),Q(i),X(i)] = qsmm1(lambda, mu(i));
endfor

# Utiliaztion
figure(1);
hold on;
plot(mu,U,"linewidth",2.2);
title("Utilization based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Utilization","fontsize",12);

hold off;

# Server responce time
figure(2);
hold on;
plot(mu,R,"linewidth",2.2);
title("Response Time based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Response time","fontsize",12);

hold off;

# Average number of requests
figure(3);
hold on;
plot(mu,Q,"linewidth",2.2);
title(" Average number of requests based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Average Number of requests","fontsize",12);

hold off;

# Server throughput
```

```
figure(4);
hold on;
plot(mu,X,"linewidth",2.2);
title("Throughput based on \\mu","fontsize",12);
xlabel("\\mu","fontsize",12);
ylabel("Throughput","fontsize",12);

hold off;

clc;
clear all;
```

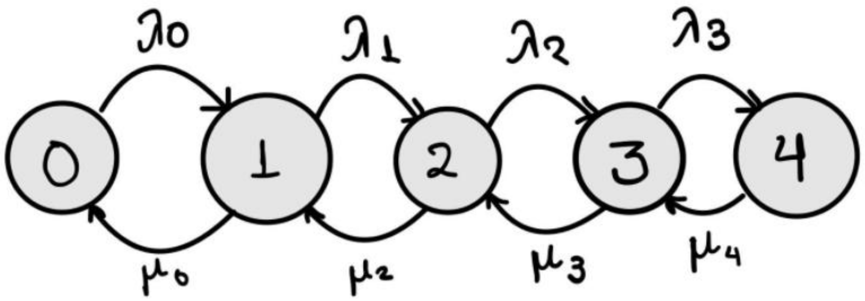
Διαδικασία γεννήσεων θανάτων (birth-death process): εφαρμογή σε σύστημα M/M/1/K

(α) Ισχύει ότι $\lambda i = \frac{\lambda}{i+1}$ και μ_i = μ όπου

λi: μέσος ρυθμός αφίξεων στο σύστημα

μi: μέσος ρυθμός εξυπηρέτησης

i: κατάσταση που βρίσκεται το σύστημα (0,1,2,3,4)



$\lambda_0 P_0 = \mu_1 P_1 \Rightarrow \lambda_{k-1} P_{k-1} = \mu_k P_k$ για $k=1,2,3,4$

$P_0+P_1+P_2+P_3+P_4=1$

$$P_k = \frac{\lambda^k}{k!\,\mu^k} P_0 \overset{\frac{\lambda}{\mu}=\rho}{=} P_k = \frac{\rho^k}{k!} P_0 \quad , k=1,2,3,4$$

Από το τύπο εξάγουμε ότι P0=0,60664, P1=0,30332, P2=0,07583, P3=0,012638, P4=Ploss=0,0015798

(β) i)

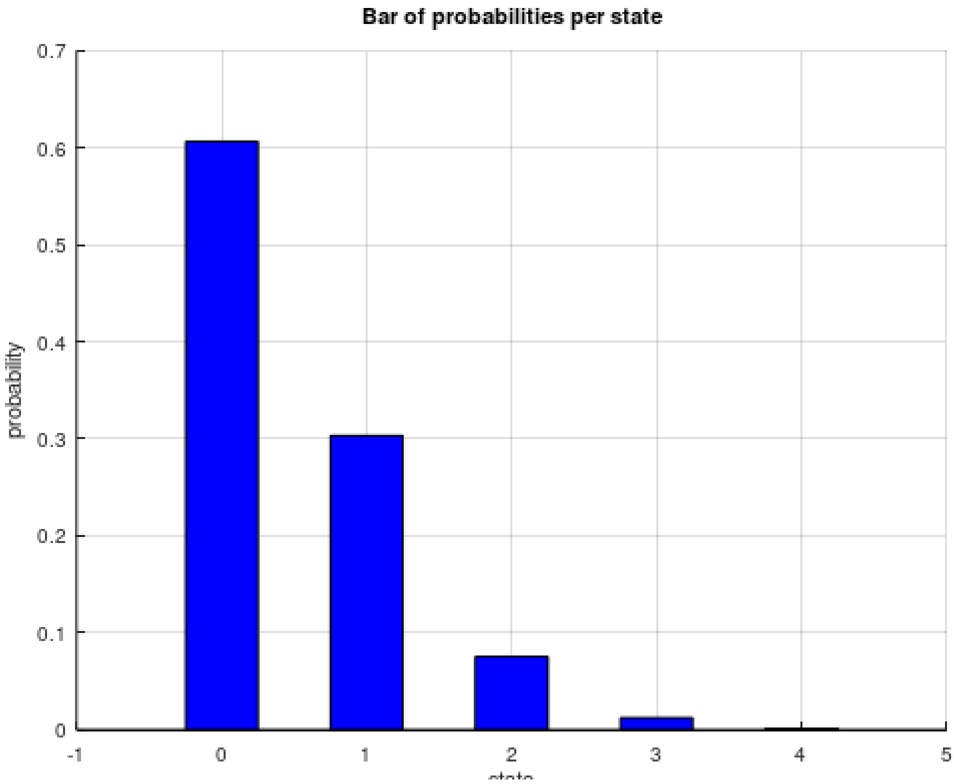
```
transition_matrix =

   -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
```

ii)

```
P =

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

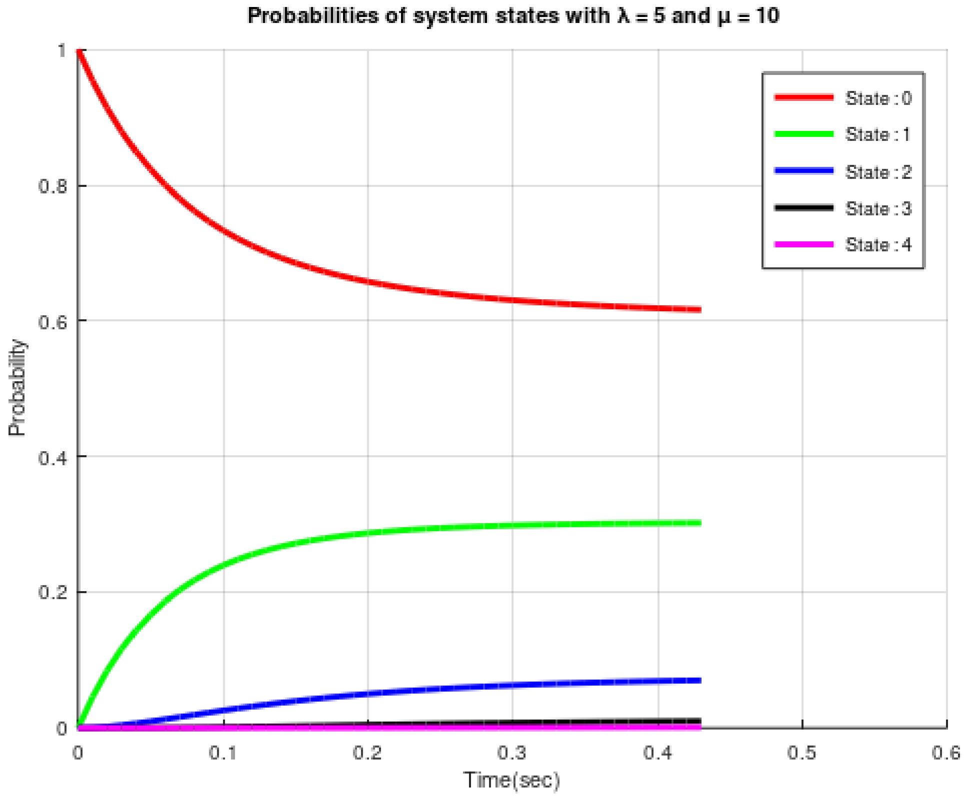


iii)

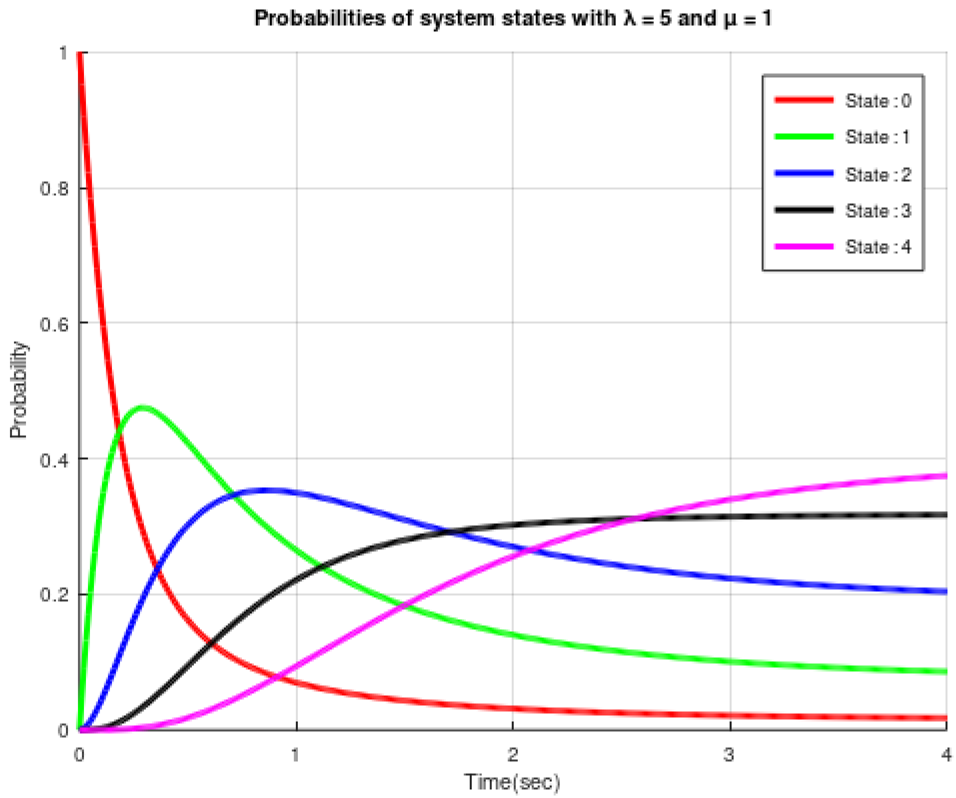
```
 Average Number of customers in the system :
0.4992
```

iv) Όπως φαίνεται στο ερώτημα ii, P4=1.5798e-03

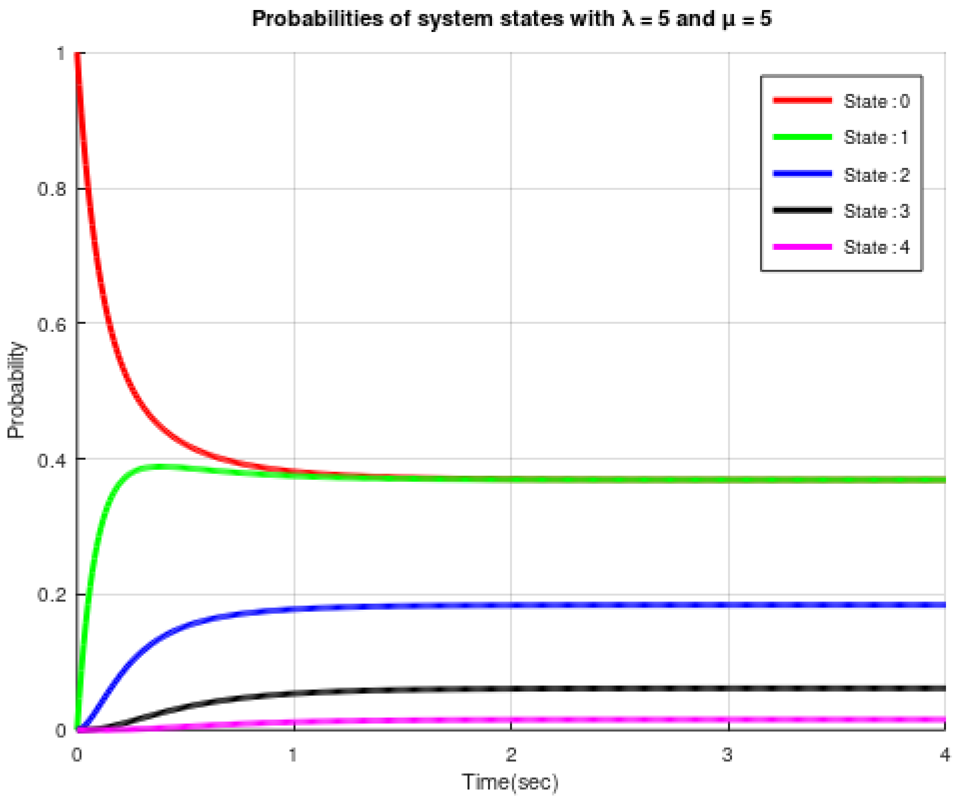
v)


Probabilities of system states with λ = 5 and μ = 10
State : 0
State : 1
State : 2
State : 3
State : 4
Probability
Time(sec)
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5

vi)i)

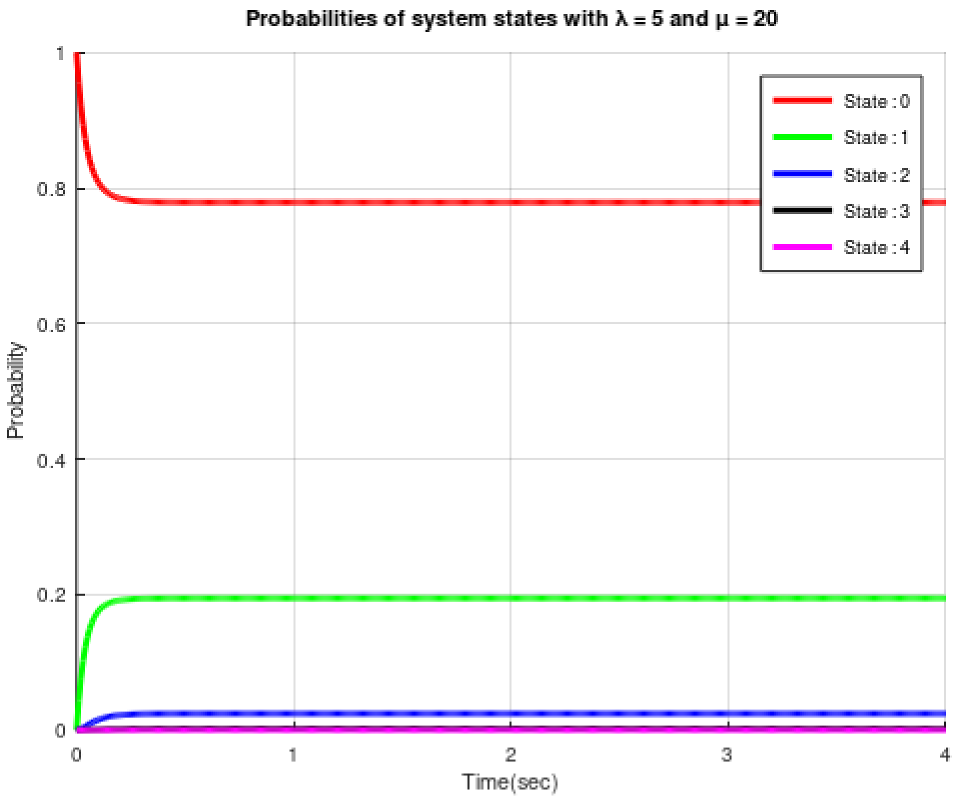

Probabilities of system states with λ = 5 and μ = 1
State : 0
State : 1
State : 2
State : 3
State : 4
Probability
Time(sec)
0
0.2
0.4
0.6
0.8
1
2
3
4

ii)



Probabilities of system states with λ = 5 and μ = 5
Probability
Time(sec)
State : 0
State : 1
State : 2
State : 3
State : 4
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4

iii)



Probabilities of system states with λ = 5 and μ = 20
Probability
Time(sec)
State : 0
State : 1
State : 2
State : 3
State : 4
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4

Κώδικας:

```
% system M/M/1/4
% when there are 3 clients in the system, the capability of the server doubles.

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

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);
display(transition_matrix);
% get the ergodic probabilities of the system
P = ctmc(transition_matrix);
display(P);


% plot the ergodic probabilities (bar for bar chart)
figure(1);
hold on;
title("Bar of probabilities per state");
xlabel("state");
ylabel("probability");
bar(states, P, "b", 0.5);
grid on;

hold off;

display( " Average Number of customers in the system : ");
display( sum(P.*[0,1,2,3,4]));

#blocking probability
display("probability of blocking a customer")
display(P(5))
#PLOT THE ERGODIC PROBABILITY

% transient probability of state 0 until convergence to ergodic probability. Convergence
 takes place P0 and P differ by 0.01
index = 0;
for T = 0 : 0.01 : 50
  index = index + 1;
  P0 = ctmc(transition_matrix, T, initial_state);
  Prob0(index) = P0(1);
  Prob1(index) = P0(2);
  Prob2(index) = P0(3);
  Prob3(index) = P0(4);
  Prob4(index) = P0(5);
  if P0 - P < 0.01
    break;
```

```
  endif
endfor

T = 0 : 0.01 : T;
figure(2);
title(strjoin({"Probabilities of system states with \\lambda = ",num2str(lambda)," and \
\mu = ",num2str(mu)},""))
xlabel("Time(sec)")
ylabel("Probability")
hold on;
plot(T, Prob0, "r", "linewidth", 1.5);
plot(T, Prob1, "g", "linewidth", 1.5);
plot(T, Prob2, "b", "linewidth", 1.5);
plot(T, Prob3, "k", "linewidth", 1.5);
plot(T, Prob4, "m", "linewidth", 1.5);
legend("State : 0","State : 1","State : 2","State : 3","State : 4");
grid on;

hold off;

% transient probability of state 0 until convergence to ergodic probability. Convergence
 takes place P0 and P differ by 0.01
mu = [1,5,20];
for i=1:columns(mu)
  deaths_D = [mu(i), mu(i), mu(i), mu(i)];
  transition_matrix = ctmcbd(births_B, deaths_D);
  index = 0;
  for T = 0 : 0.01 : 4
    index = index + 1;
    P0 = ctmc(transition_matrix, T, initial_state);
    Prob0(index) = P0(1);
    Prob1(index) = P0(2);
    Prob2(index) = P0(3);
    Prob3(index) = P0(4);
    Prob4(index) = P0(5);
    if P0 - P < 0.01
      break;
    endif
  endfor


 T = 0 : 0.01 : T;
  figure(i+2);
  title(strjoin({"Probabilities of system states with \\lambda = ",num2str(lambda)," and
 \\mu = ",num2str(mu(i))},""))
  xlabel("Time(sec)")
  ylabel("Probability")
  hold on;
  plot(T, Prob0, "r", "linewidth", 1.5);
  plot(T, Prob1, "g", "linewidth", 1.5);
  plot(T, Prob2, "b", "linewidth", 1.5);
  plot(T, Prob3, "k", "linewidth", 1.5);
  plot(T, Prob4, "m", "linewidth", 1.5);
  legend("State : 0","State : 1","State : 2","State : 3","State : 4");
  grid on;

  hold off;
endfor
```