ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΜΠ

“QUEUING SYSTEMS”

Ορφανουδάκης Φίλιππος 03113140

<u>2η Ομάδα Ασκήσεων</u>

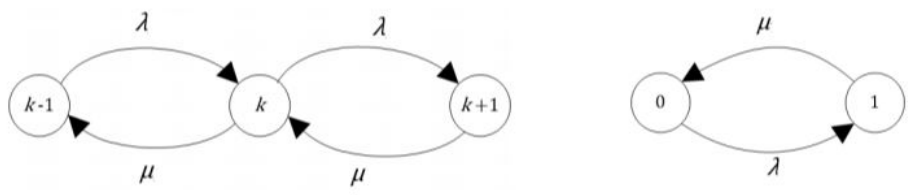
Θεωρητική μελέτη της ουράς M/M/1

A)

- Για την ουρά M/M/1 έχουμε ότι η απαραίτητη συνθήκη για οριακή ισορροπία εργοδικότητα είναι : ρ=u=λ/μ<1 Earlang

Δηλαδή το σύστημα να μην δουλεύει στο 100% αλλά να υπάρχουν χρονικοι περιόδοι που το σύστημα θα μπορέσει να ‘’ξεκουραστεί’’.

- Για ουρά M/M/1 με παραμέτρους λ,μ έχουμε το εξής διάγραμμα μεταβάσεων.



- Οι αντίστοιχες εξισώσεις ισορροπίας είναι :

- $\lambda P_0 = \mu P_1$ 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 $P_k = \rho^k P_0, \;\; k > 0$
- $P_0 + P_1 + \cdots + P_k + \cdots = 1 = P_0(1 + \rho + \rho^2 + \rho^3 + \cdots)$

Συνεπώς έχω ότι Pk=p^k *P0, άρα χρειάζομαι το P0.

Με $0 < \rho < 1$ η άπειρη δυναμοσειρά συγκλίνει, $P_0\left(\frac{1}{1-\rho}\right) = 1 \Rightarrow$

$P_0 = (1-\rho), \;\; P_k = (1-\rho)\rho^k, k > 0$ και $\mathrm{P}\{n(t) > 0\} = 1 - P_0 = \rho$

B)

Μας ζητείται να αποδείξουμε το εξής :

- $$\mathrm{E}[n(t)] = \sum_{k=1}^{+\infty} kP_k = \frac{\rho}{1-\rho} \quad \acute{o}\pi o\upsilon\ \rho = \frac{\lambda}{\mu}$$

$$\sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} k(1-\rho)\rho^k = (1-\rho)\rho \sum_{k=1}^{\infty} k\rho^{k-1} = (1-\rho)\rho \frac{1}{(1-\rho)^2}$$

Επομένως έχω ότι E[n(t)]= $\frac{p}{1-p}$.

- Όταν η ουρά αναμονής βρίσκεται σε κατάσταση ισορροπίας τότε έχω ότι ρ=u⇒λ=γ

  Οπότε απο τύπο Little προκύπτει ότι :

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

C)

Ναι θα υπάρξει τέτοια χρονική στιγμή καθώς για να μεταβούμε στην κατάσταση ισορροπίας πρέπει να έχουμε επαναληπτικές καταστάσεις n(t)=k (απείρως επισκέψιμες).

D) Δεν θα άλλαζε κάτι καθως έχουμε κατανομή Poisson στις αφίξεις , η οποία έχει την ιδιότητα απώλειας μνήμης , δηλαδή το παρελθόν δεν επηρεάζει την μελλοντική κατάσταση της ουράς.

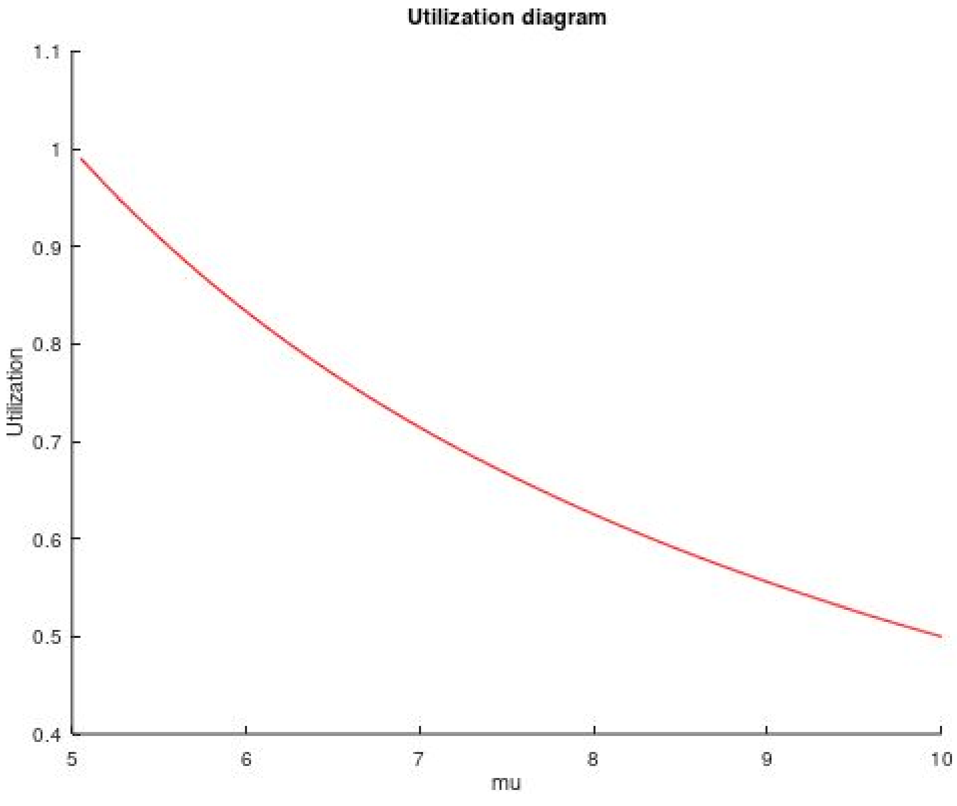
Ανάλυση ουράς M/M/1 με Octave

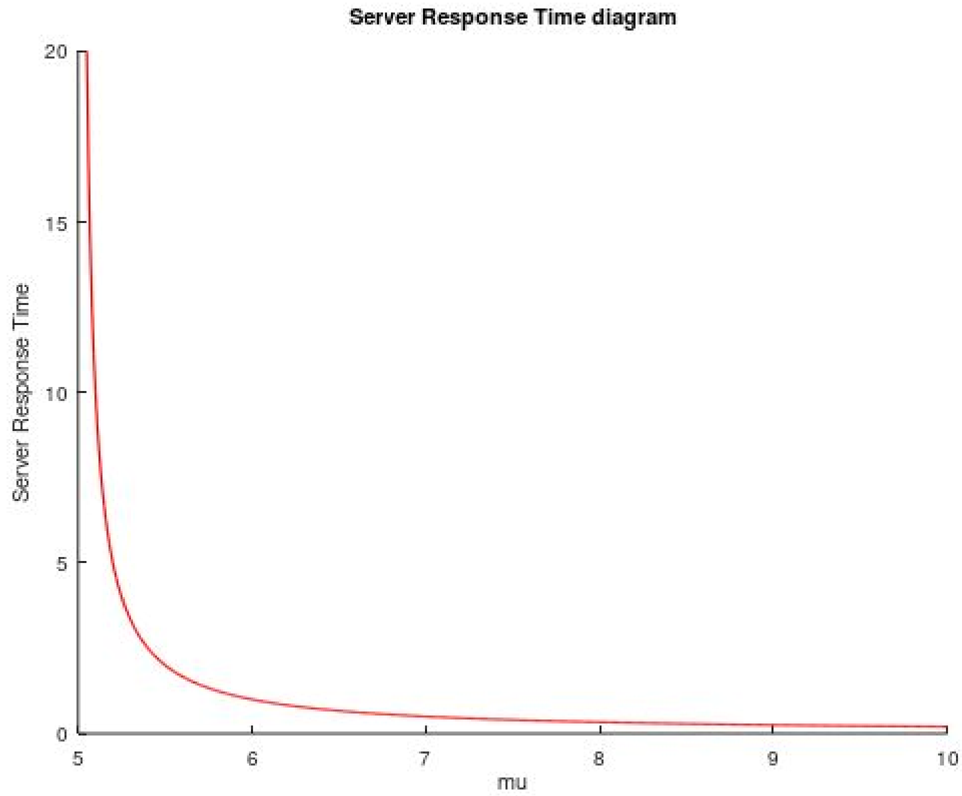
A)

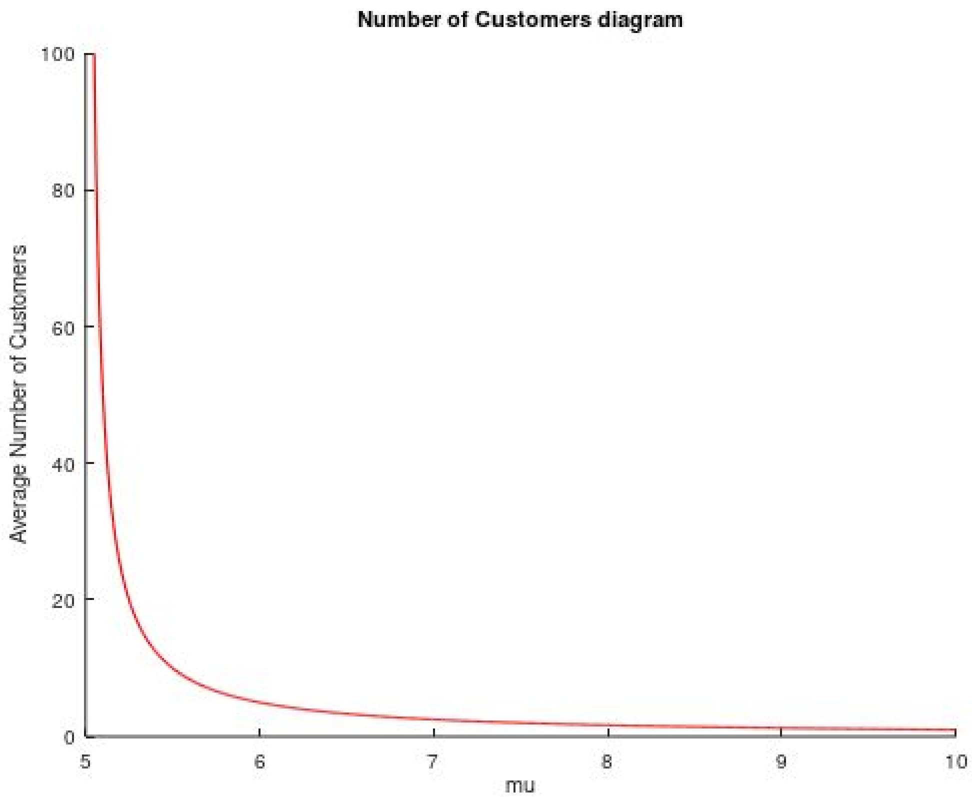
Όταν το σύστημα μας είναι εργοδικό έχουμε σαν προυπόθεση να ισχύει ότι $\rho<1 \Rightarrow \lambda<\mu$

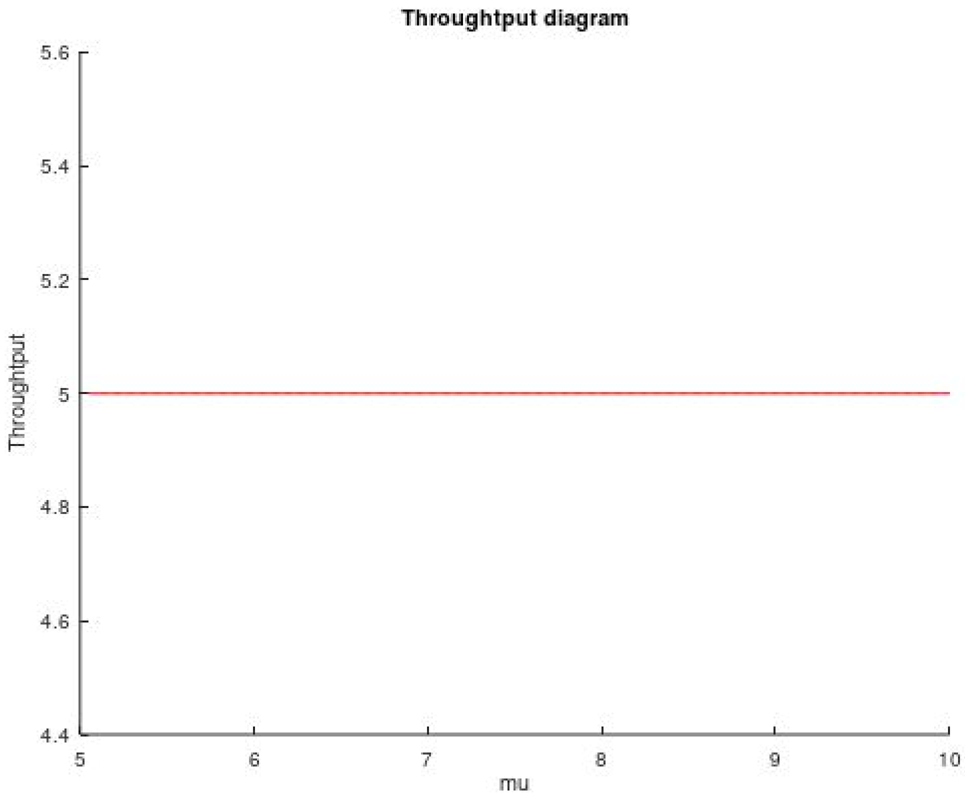
Επομένως $\mu \in (5,10]$

B)




Server Response Time diagram
Server Response Time
20
15
10
5
0
5
6
7
8
9
10
mu


Number of Customers diagram
Average Number of Customers
100
80
60
40
20
0
5
6
7
8
9
10
mu



C) a) Με βάση το διάγραμμα μέσου χρόνου καθυστέρησης ξεκάθαρα η καλύτερη επιλογή είναι θα επιλέξουμε το mu=10 καθώς όσο μεγαλώνει το mu τόσο μικραίνει το μέσος χρόνος καθυστέρησης των πελατών, αν λάβουμε υπόψη και τα υπόλοιπα διαγράμματα , τότε η τιμή 7 που αρχίζει ο κορεσμός ειναι μια καλύτερη επιλογη.

d) Απο τη στιγμή που έχουμε ότι είναι εργοδικό έχουμε το συμπέρασμα ότι throughtput = γ = λ αφού έχω μηδενική πιθανότητα απωλειών, επομένως αναμενόμενο .

Σύγκριση συστημάτων με δύο εξυπηρετητές

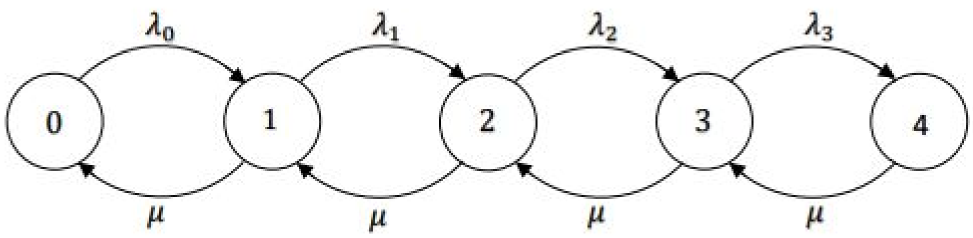
Έχουμε το εξής αποτέλεσμα :

```
Command Window
M/M/2 queue: E(T) = 0.13333
2 M/M/1 queues: E(T) = 0.2
>>
```

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

A)

Αφού έχουμε M/M/1/4 τότε θα έχουμε μέγιστη χωρητικότητα 4 οπότε 4 nodes στο διάγραμμα μεταβάσεων και θα έχουμε επίσης μi=μ και λ0=λ,λ1=λ/2,λ2=λ/3,λ3=λ/4. Επίσης όταν η ουρά γεμίσει τότε θα έχουμε απώλειες , επομένως η πιθανότητα απωλειας είναι P4.
Το το διάγραμμα είναι το εξής:



Επόμενο στάδιο είναι να παρουμε τις εξισώσεις ισορροπίας , όπως είχαμε πάρει και στη περίπτωση M/M/1 μόνο που τώρα δεν έχουμε σταθερο λ , άρα :

λ0 * P0= μ * P1 => P1=P0/2
λ1*P1+μ*P1=λ0*P0 + μ*P2 => P2=P0/8
λ2*P2+μ*P2=λ1*P1 + μ*P3=>P3=P0/48
λ3*P3+μ*P3=λ2*P2 + μ *P4=>P4=P0/384

και επίσης

Κανονικοποίηση Εργοδικών Πιθανοτήτων

$$P_0 + \cdots + P_N = 1$$

Άρα :

P0(½ +⅛ + 1/48 + 1/384)=1=> P0=0,606 ≈0,6

αρα P1=0.3 , P2=0.075 , P3= 0.0125 , P4= 0.00156= Πιθανότητα απωλειών

B)

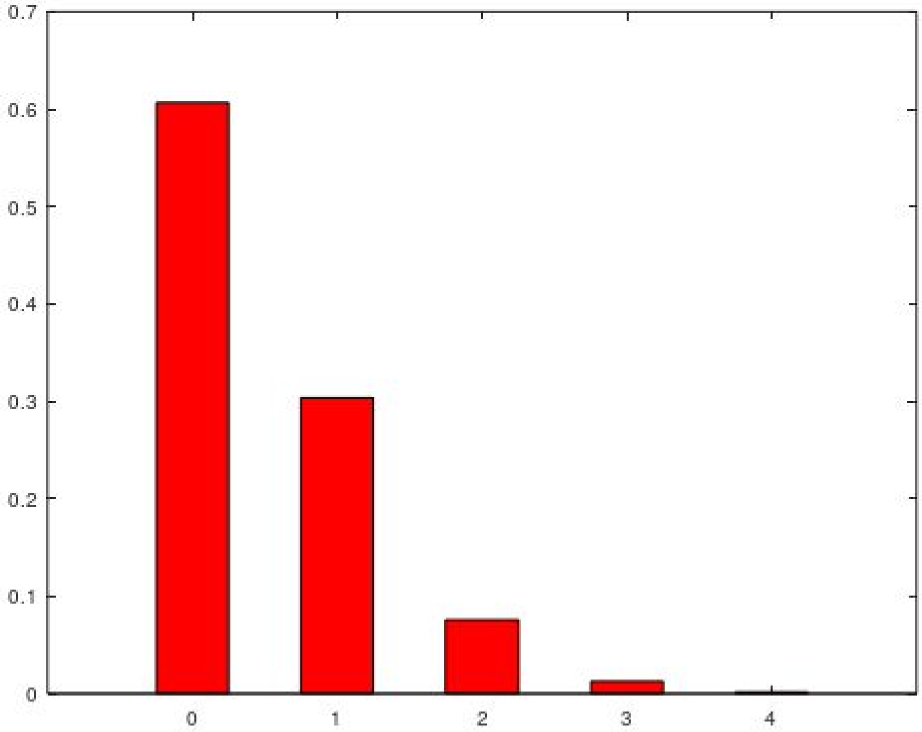
i)

```
transition_matrix =

   -5.00000    5.00000    0.00000    0.00000    0.00000
   10.00000  -12.50000    2.50000    0.00000    0.00000
    0.00000   10.00000  -11.66667    1.66667    0.00000
    0.00000    0.00000   10.00000  -11.25000    1.25000
    0.00000    0.00000    0.00000   10.00000  -10.00000
```

ii)

v)



Απο το οποίο επιβεβαιώνονται οι τιμές που βρήκαμε.

iii)

- Στάσιμος Εργοδικός μέσος όρος πληθυσμού – κατάστασης

$$\mathrm{E}[n(t)] \rightarrow \mathrm{E}(k) = \sum_{k=1}^{N} kP_k = \rho \frac{1-(N+1)\rho^N + N\rho^{N+1}}{(1-\rho)(1-\rho^{N+1})}$$

Επομένως έχω :

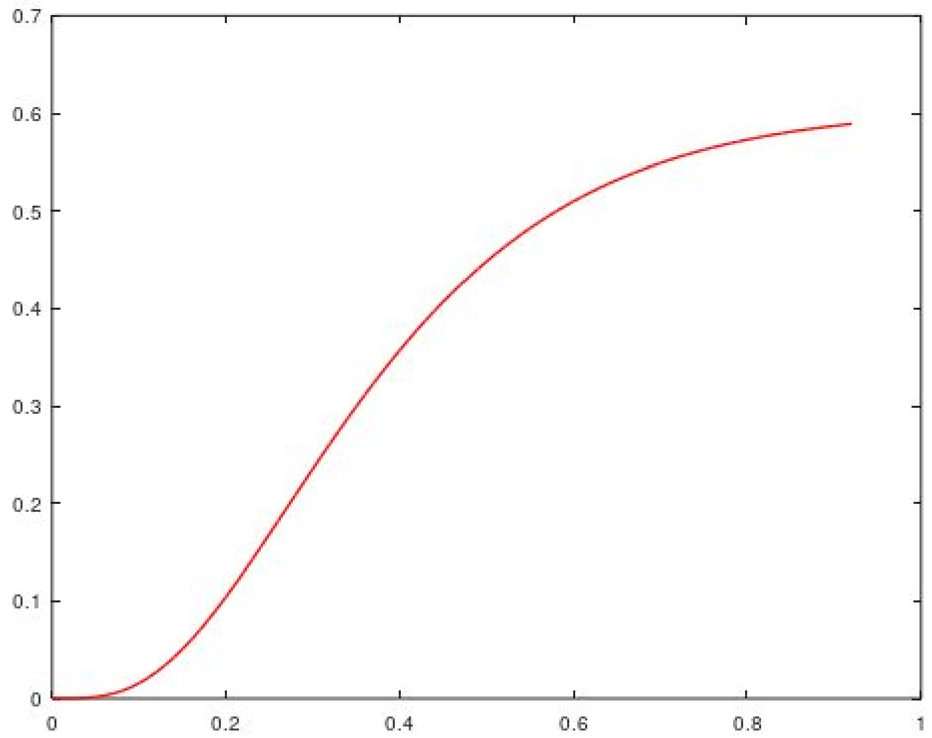
```
E[n(t)] = 0.49921
>> 
```

iv)

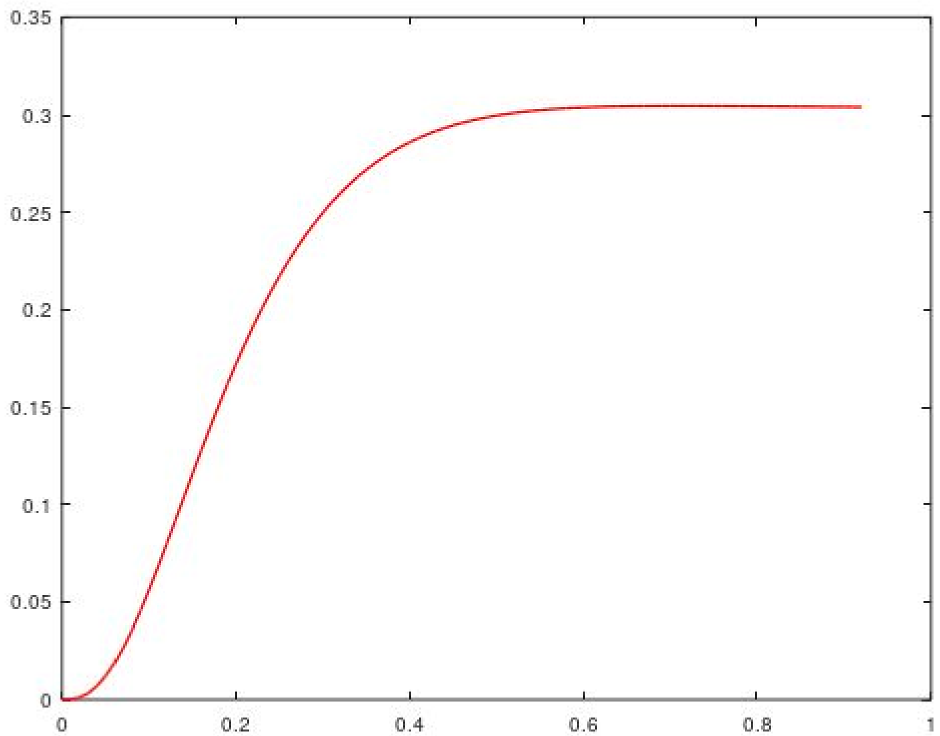
```
P{Blocking} = 0.0015798
>> 
```

v)

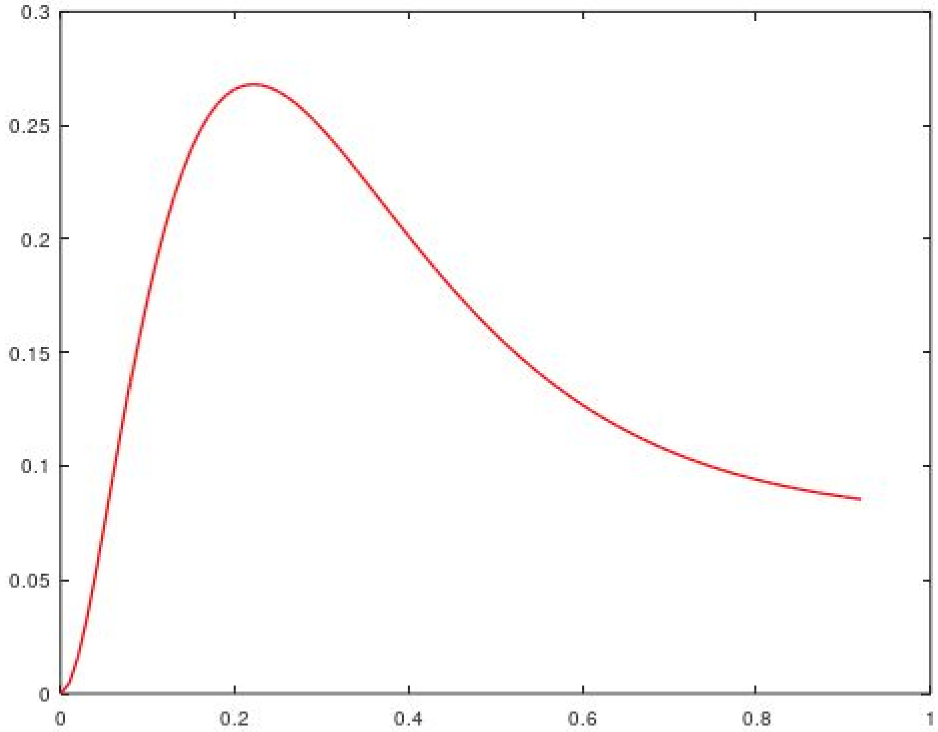
- Κατασταση 0



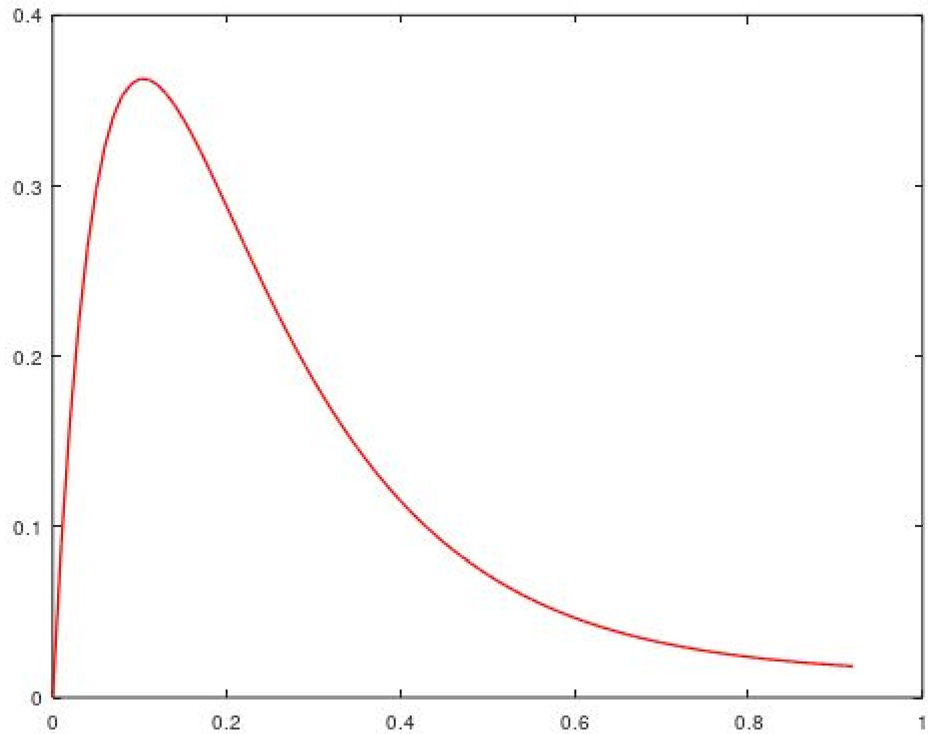
- Κατάσταση 1



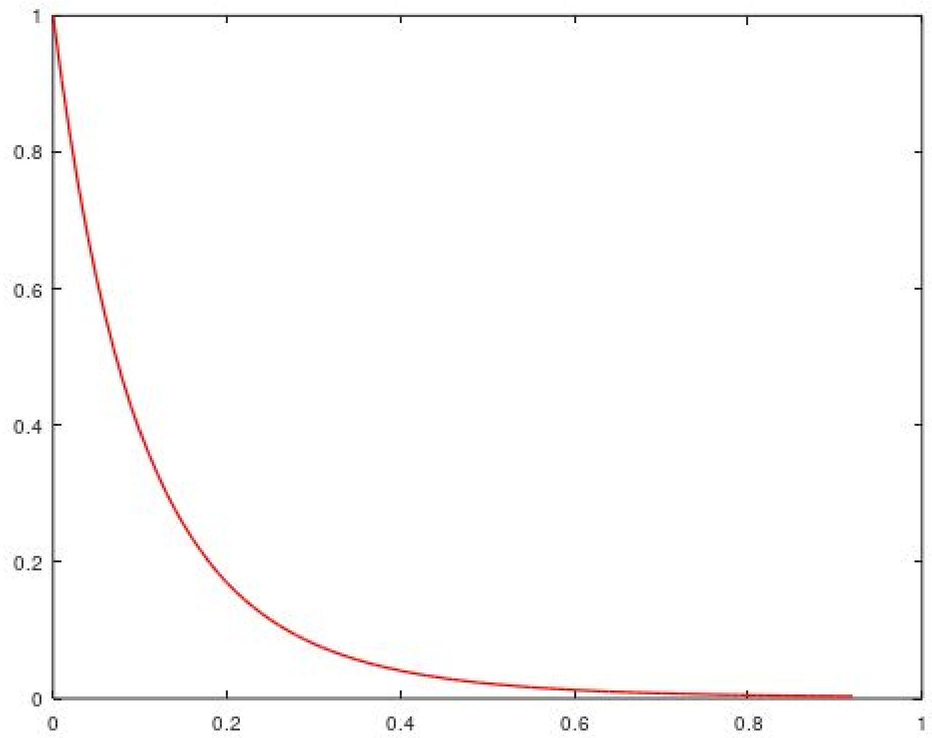
- Κατάσταση 2



- Κατάσταση 3

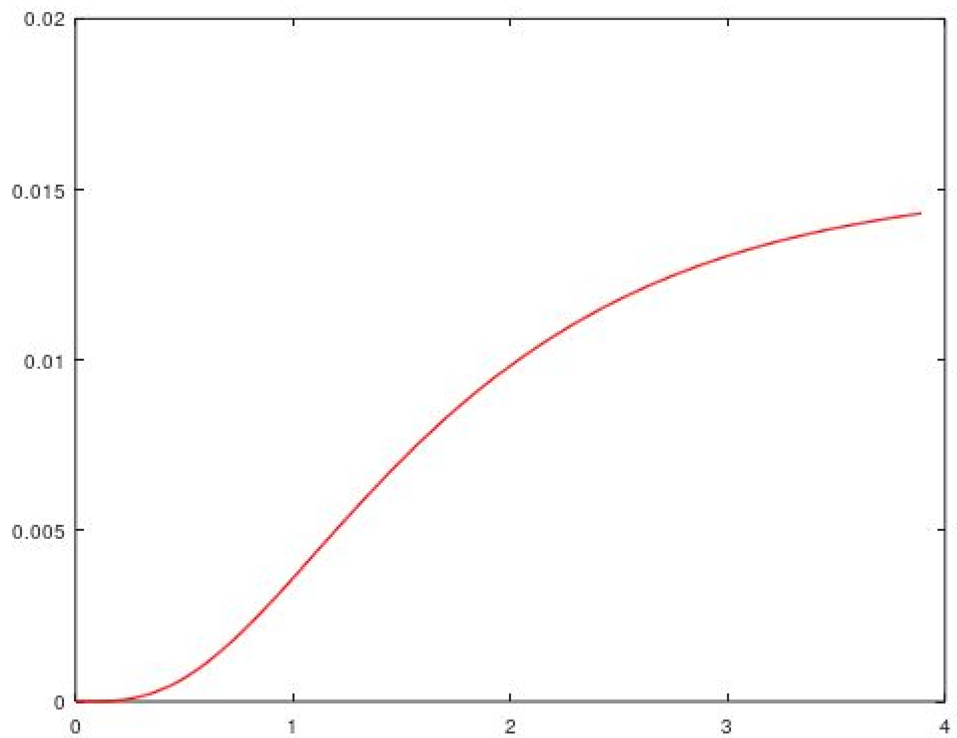


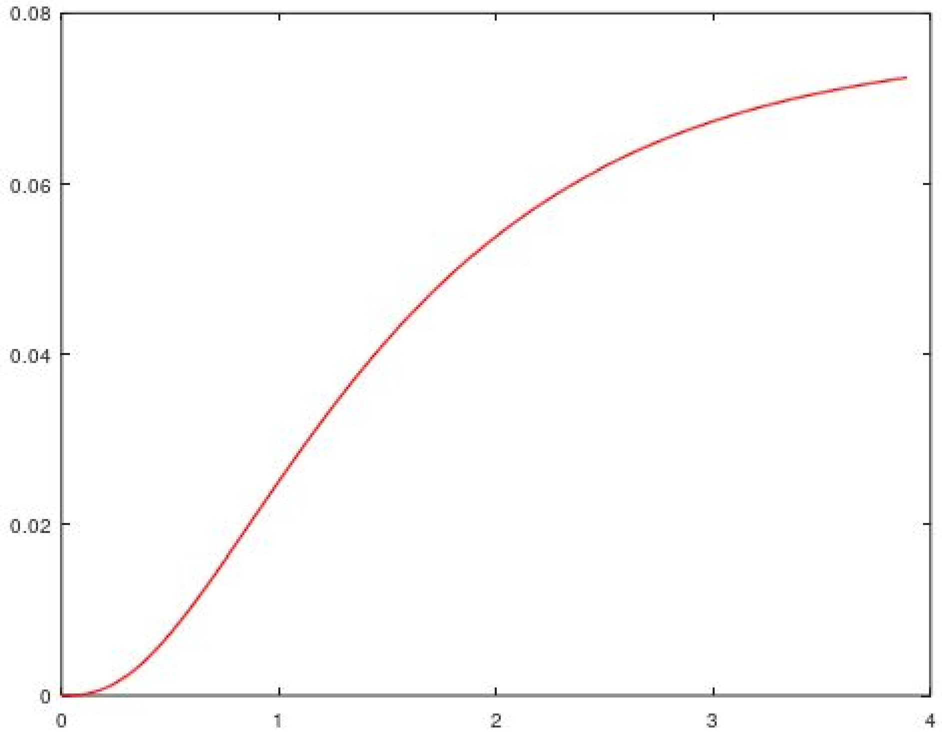
- Κατάσταση 4

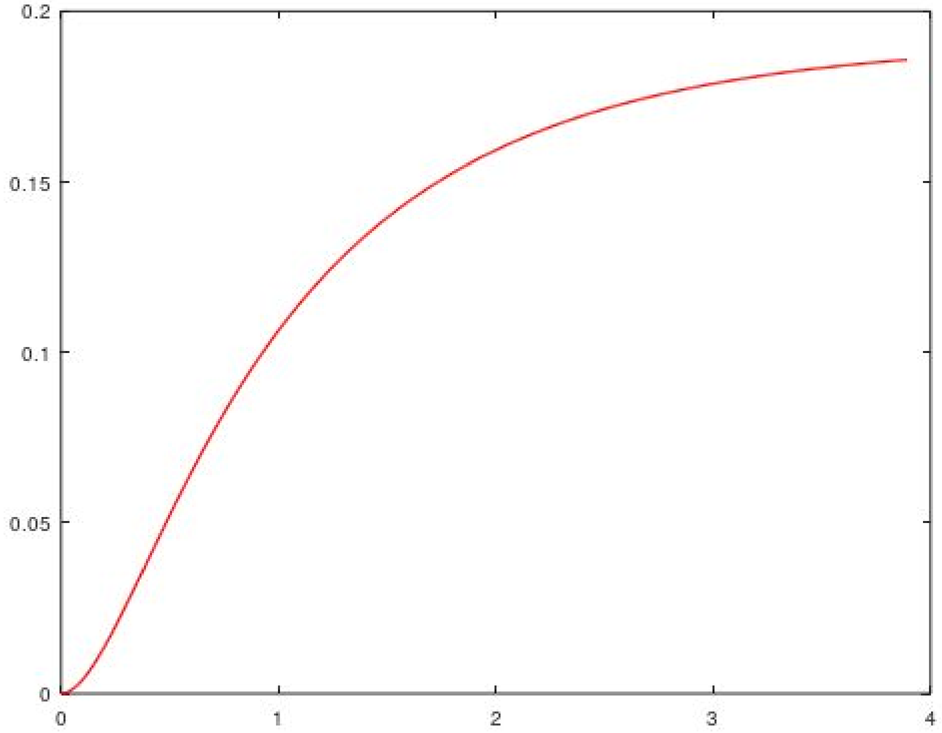


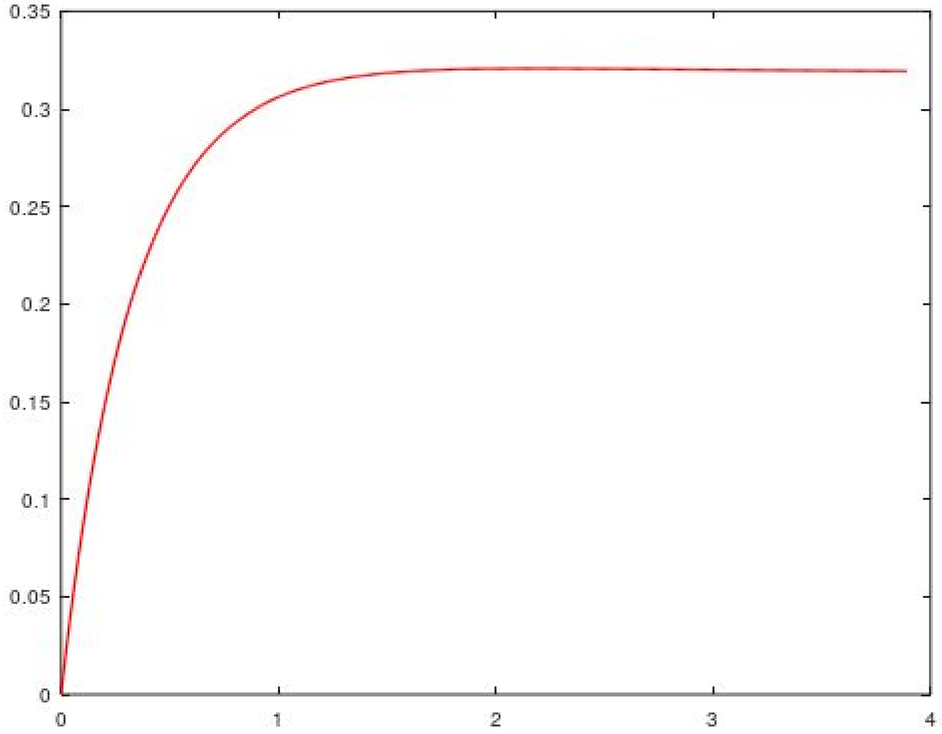
vi)

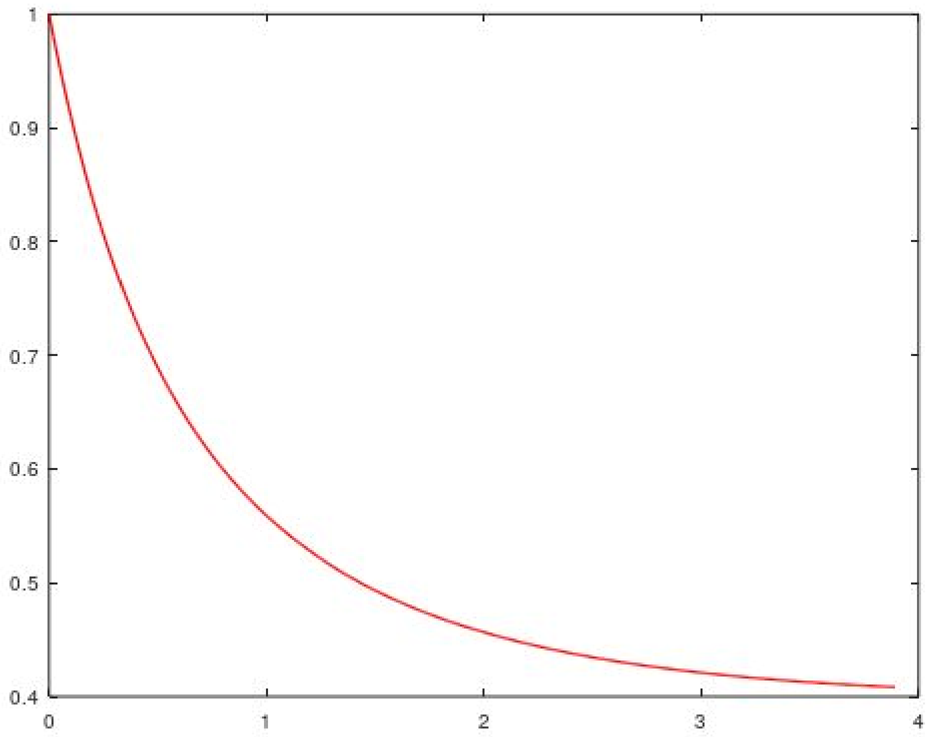
- για λ=5 μ=1



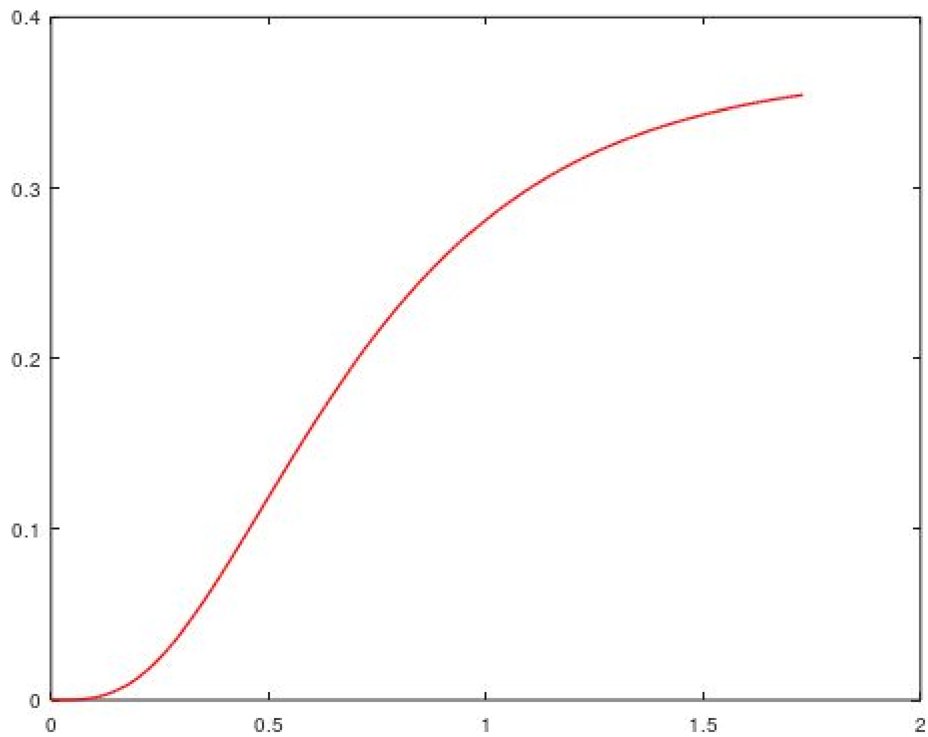

0.08
0.06
0.04
0.02
0
0
1
2
3
4

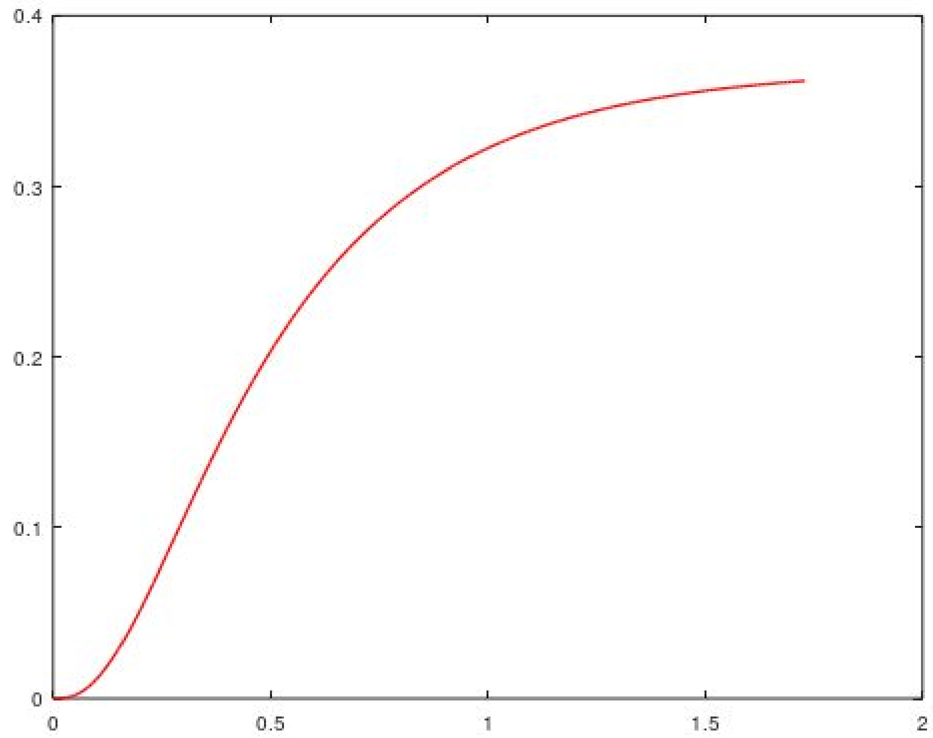

0.2
0.15
0.1
0.05
0
0
1
2
3
4

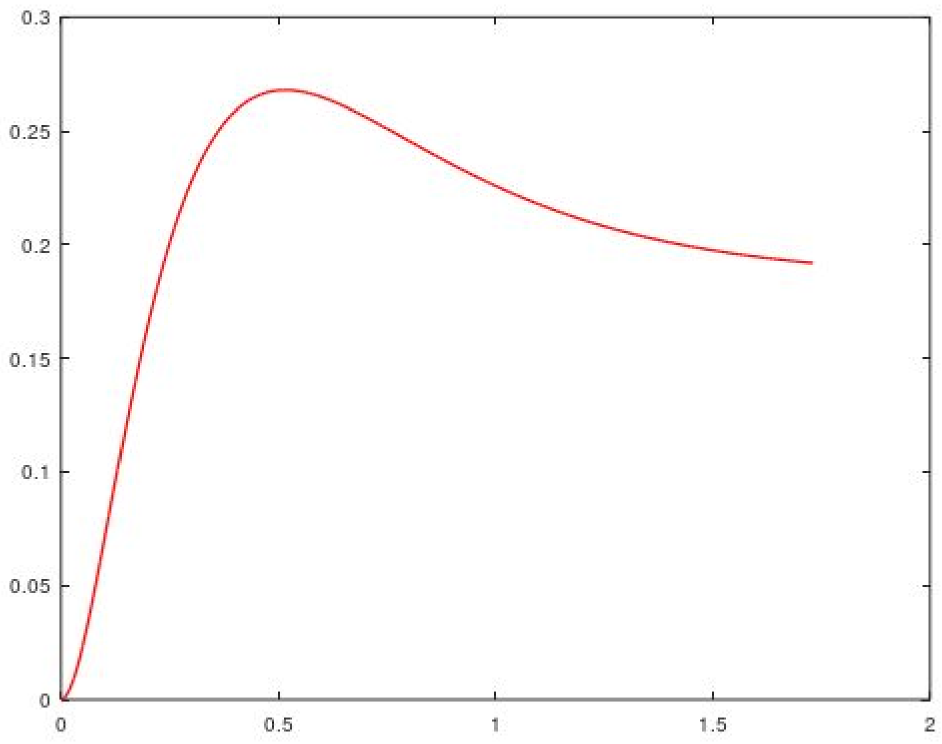

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
4

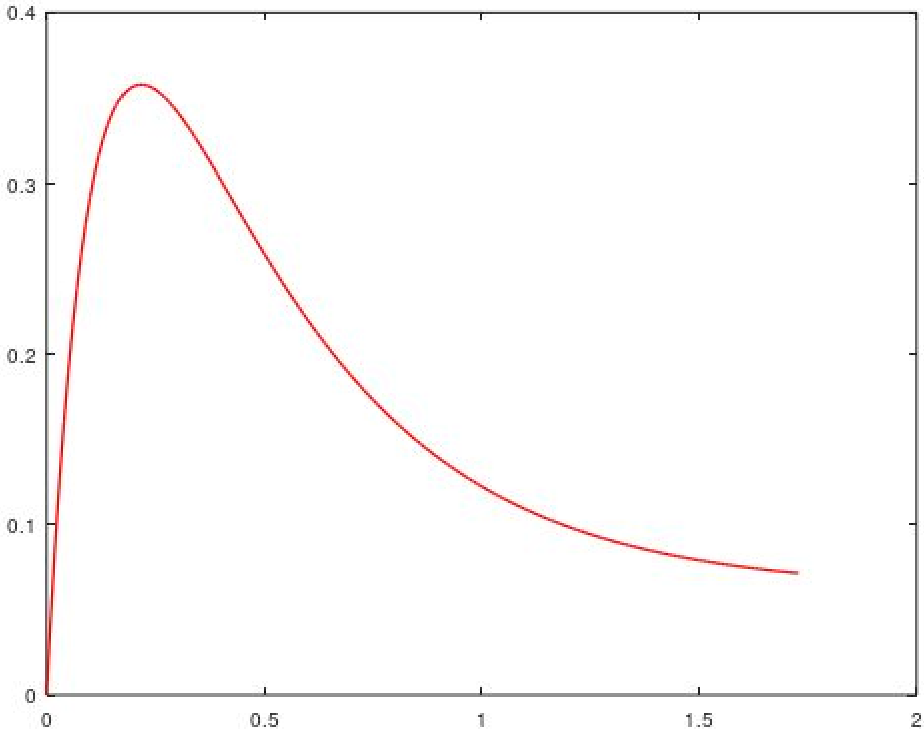


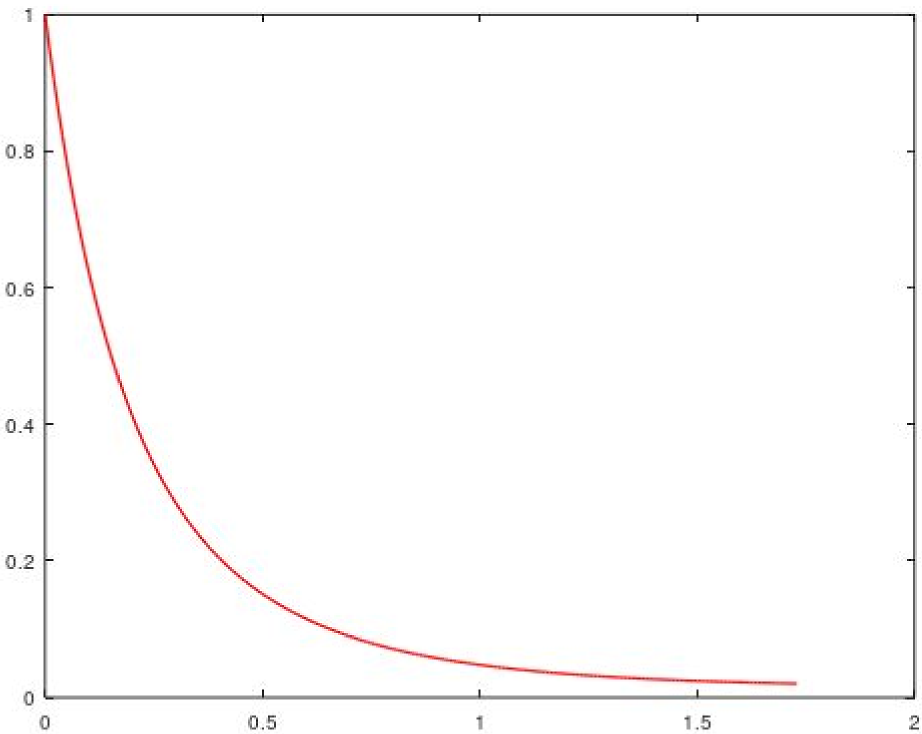
- για λ=5 μ=5



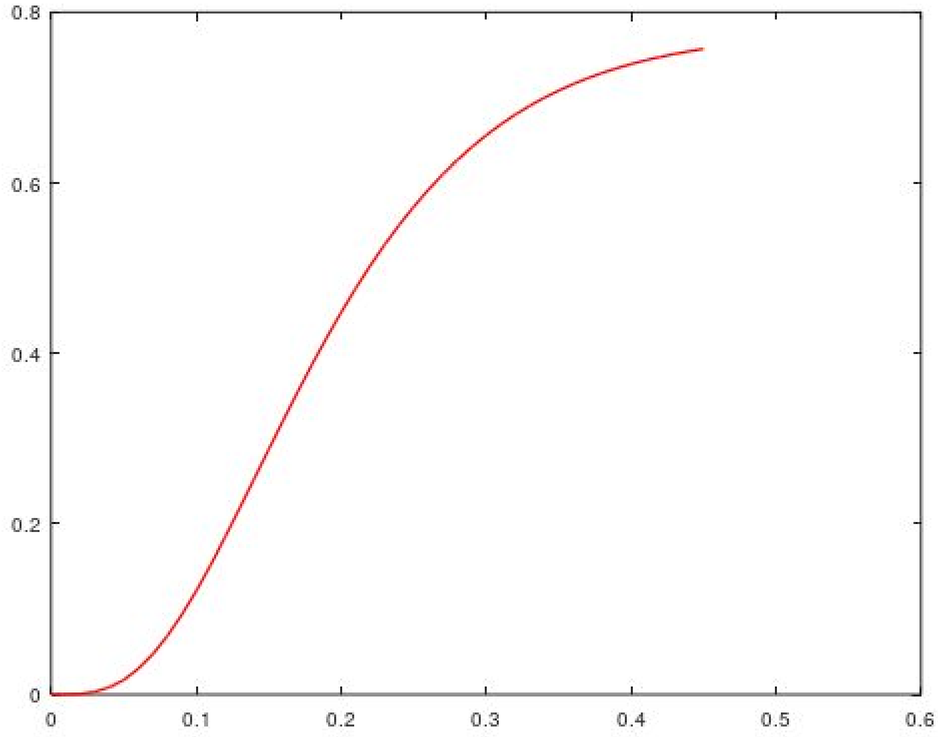

0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2

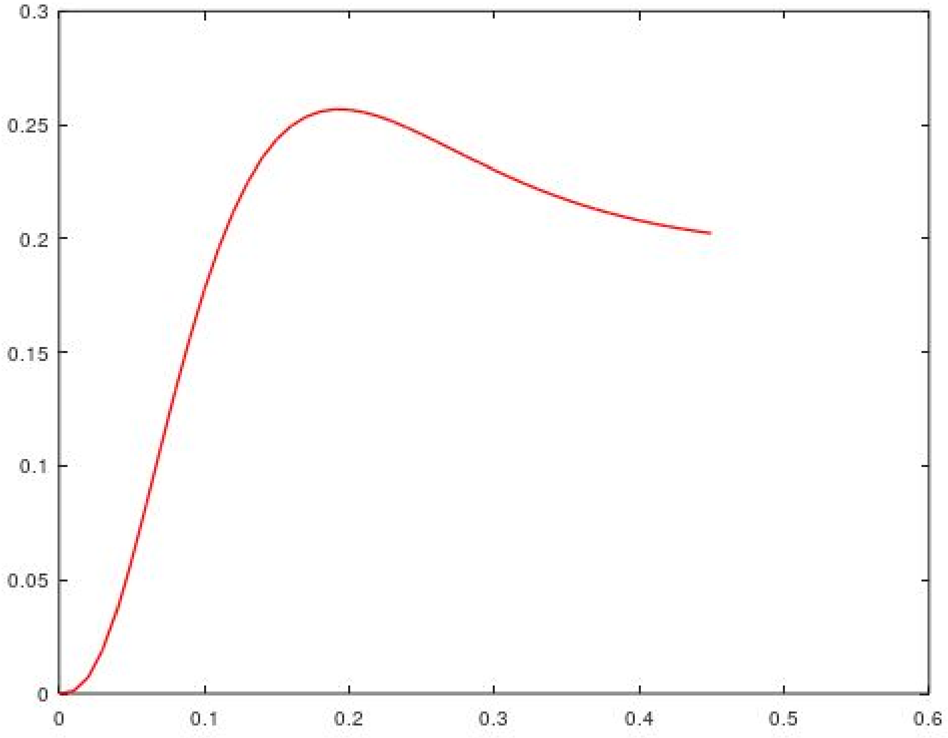

0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2

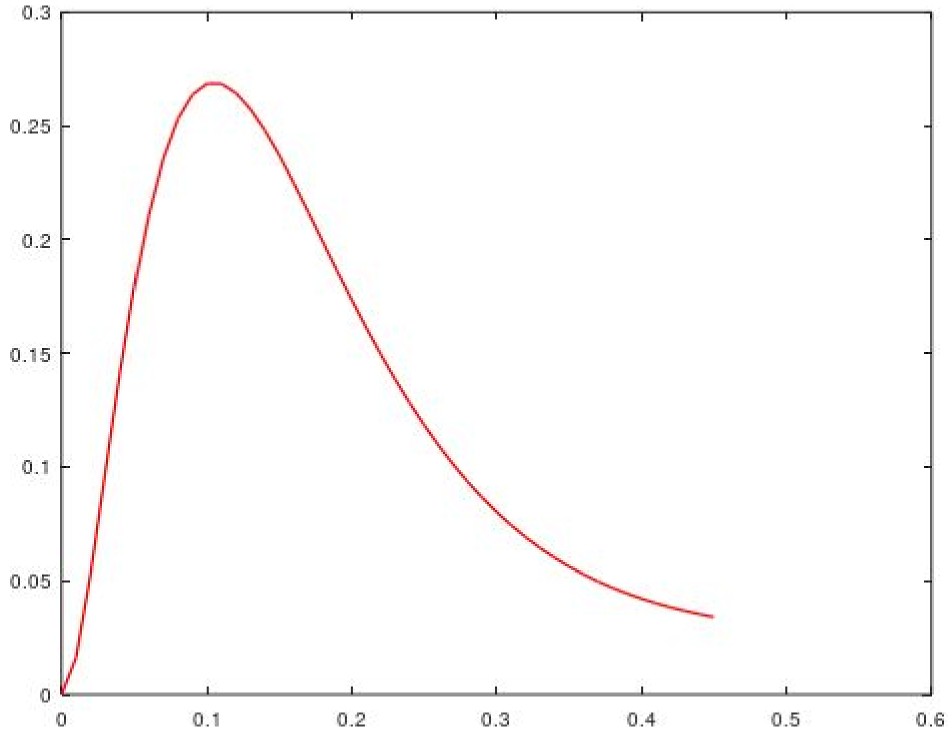


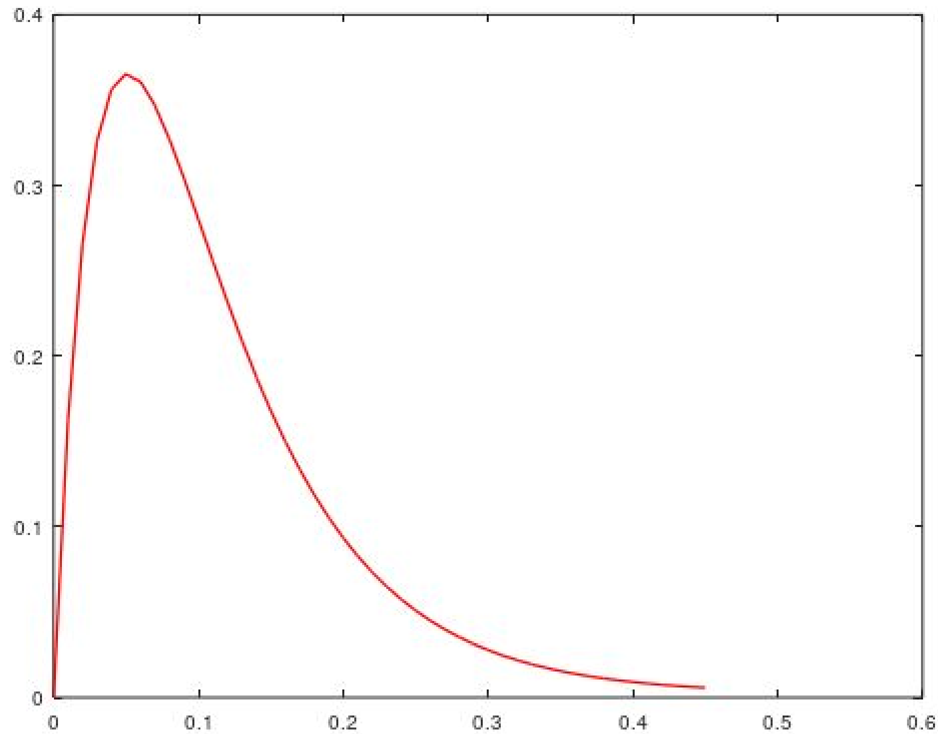


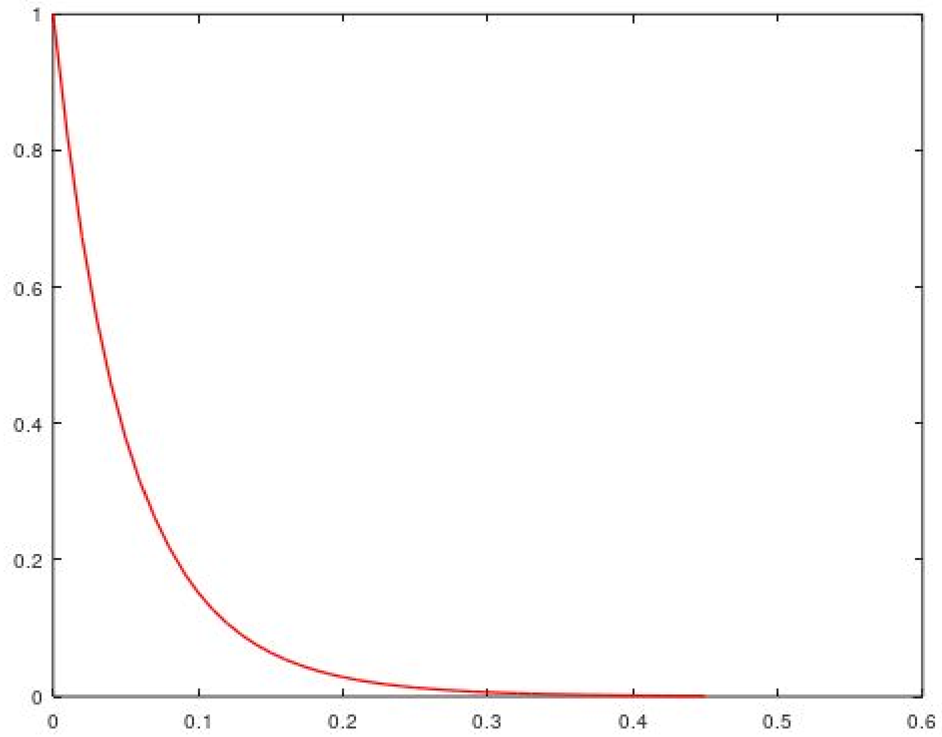
- για λ=5 μ=20


0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6


0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6


0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6


0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6



Στη 1η περίπτωση έχουμε p=5>1 , στη δεύτερη έχουμε p=1, και στη 3η περίπτωση έχουμε p=0.25<1.

Έχουμε ξεκάθαρα πολύ γρηγορότερη σύγκλιση στη 3η περίπτωση που περίπου στα 0.17 sec έχουμε την εργοδική μας κατάσταση, ενω στη 2η περίπτωση την φτάνουμε στα 1.15 sec και στην 1η περίπτωση στα 4.2 sec.

Επίσης βλέπουμε ότι στην 1η περίπτωση οι τιμές των πιθανοτήτων που τελικά θα συγκλίνουν είναι αρκετά μακριά από τις εργοδικές. Το σύστημα θα έχει πολλές απώλειες , όπως φάινεται και απο την μεγάλη πιθανότητα P4 στην οποία το σύστημα θα είναι γεμάτο.

Στην 2η περίπτωση έχουμε μια καλύτερη σύγκλιση στις τιμές μας .

Στην 3η περίπτωση παρατηρούμε ότι στις μεγαλύτερες καταστάσεις η πιθανότητα ειναι χαμηλότερη από την εργοδική και στην κατάσταση 0 είναι μεγαλύτερη , το σύστημα εξυπηρετεί πιο γρήγορα από ότι λαμβάνει πελάτες με αποτέλεσμα το σύστημα να τείνει να αδειάσει.

Παράρτημα , source code :

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

```
#M/M/1
colors = "rbgm";
lambda = 5;
mu = 5.05:0.01:10;

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




figure(1);
hold on;
plot(mu,U,colors(1),"linewidth",1.2);
hold off;
title("Utilization diagram");
xlabel("mu");
ylabel("Utilization");

figure(2);
hold on;
plot(mu,R,colors(1),"linewidth",1.2);
hold off;
title("Server Response Time diagram");
xlabel("mu");
ylabel("Server Response Time");

figure(3);
hold on;
plot(mu,Q,colors(1),"linewidth",1.2);
hold off;
title("Number of Customers diagram");
xlabel("mu");
ylabel("Average Number of Customers");

figure(4);
hold on;
plot(mu,X,colors(1),"linewidth",1.2);
hold off;
title("Throughtput diagram");
xlabel("mu");
ylabel("Throughtput");

# M/M/1 vs M/M/2 comparison
[U2, R2, Q2, X2, p02] = qsmmm(10,10,2);
display(cstrcat("M/M/2 queue: E(T) = ", num2str(R2)));
```

```
[U1, R1, Q1, X1, p01] = qsmm1(5,10);
display(cstrcat("2 M/M/1 queues: E(T) = ", num2str(R1)));

# M/M/1/4 ,  lamda=5, mu=1/ lamda=5,mu=5/lamda=5,mu=20

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 = [0,0,0,0,1];

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

display(transition_matrix);

figure(5);
bar(states,P,"r",0.5);

display(cstrcat("E[n(t)] = ", num2str(sum(P.*[0,1,2,3,4]))));
display(cstrcat("P{Blocking} = ", num2str(P(5))));


index = 0;
for T=0:0.01:50
  index = index + 1;
  P0 = ctmc(transition_matrix,T,initial_state);
  Prob0(index) = P0(1);
  if P0-P < 0.01
    break;
  endif
endfor

T = 0:0.01:T;
figure(6);
plot(T,Prob0,"r","linewidth",1.3);

index = 0;
for T=0:0.01:50
```

```
  index = index + 1;
  P0 = ctmc(transition_matrix,T,initial_state);
  Prob0(index) = P0(2);
  if P0-P < 0.01
    break;
  endif
endfor

T = 0:0.01:T;
figure(7);
plot(T,Prob0,"r","linewidth",1.3);

index = 0;
for T=0:0.01:50
  index = index + 1;
  P0 = ctmc(transition_matrix,T,initial_state);
  Prob0(index) = P0(3);
  if P0-P < 0.01
    break;
  endif
endfor

T = 0:0.01:T;
figure(8);
plot(T,Prob0,"r","linewidth",1.3);

index = 0;
for T=0:0.01:50
  index = index + 1;
  P0 = ctmc(transition_matrix,T,initial_state);
  Prob0(index) = P0(4);
  if P0-P < 0.01
    break;
  endif
endfor

T = 0:0.01:T;
figure(9);
plot(T,Prob0,"r","linewidth",1.3);

index = 0;
for T=0:0.01:50
  index = index + 1;
  P0 = ctmc(transition_matrix,T,initial_state);
  Prob0(index) = P0(5);
  if P0-P < 0.01
```

```
    break;
  endif
endfor

T = 0:0.01:T;
figure(10);
plot(T,Prob0,"r","linewidth",1.3);
```